

ООО "Тивун Технологии"

**Программно-вычислительный комплекс
"Тивун Интеллектуальные Виртуальные Машины"**
Руководство пользователя по эксплуатации продукта

Москва – 2021

Содержание

Список сокращений, обозначений и аббревиатур	5
Введение	7
1 Архитектура, основные возможности и технологические характеристики.....	11
1.1 Состав программного обеспечения и организация его эксплуатации	11
1.2 Требования к вычислительным платформам. Программная и информационная совместимость.....	14
1.3 Категории пользователей и управление доступом.....	16
1.4 Функциональная структура	19
1.5 Типовые сценарии применения.....	22
1.5.1 Использование провайдером виртуальных вычислительных ресурсов	22
1.5.1.1 Реализация высокоуровневых средств управления виртуализацией и интерфейса самообслуживания клиентов.....	22
1.5.1.2 Консолидация разнородных виртуальных вычислительных ресурсов и обеспечение централизованного управления ими...	23
1.5.2 Управление виртуализацией на основе вычислительных ресурсов организации.....	23
1.5.2.1 Реализация высокоуровневых средств управления виртуализацией и интерфейса самообслуживания прикладных пользователей..	24
1.5.2.2 Консолидация разнородных виртуальных вычислительных ресурсов, включая привлечение ресурсов внешних облачных сервисов	25
1.6 Варианты формирования облачных вычислительных сред	25
1.7 Доступ к серверу управления виртуализацией.....	30
2 Управление хостами виртуализации и кластерами	31
2.1 Управление хостами виртуализации	31
2.1.1 Регистрация хоста виртуализации.....	31
2.1.2 Отмена регистрации хоста виртуализации.....	32
2.1.3 Получение информации о хосте виртуализации.....	32
2.1.4 Получение списка хостов виртуализации.....	36
2.1.5 Управление ЖЦ хоста виртуализации	37
2.1.6 Очистка хоста виртуализации от VM, запущенных на нем	38
2.1.7 Определение дополнительных признаков хоста виртуализации	39
2.1.8 Обновление драйверов хоста виртуализации.....	40
2.1.9 Импорт VM, запущенных на хосте виртуализации до его регистрации в TIVM	40
2.1.10 Переименование хоста виртуализации	41
2.2 Управление кластерами хостов виртуализации	41
2.2.1 Регистрация кластера.....	42
2.2.2 Переименование кластера	42
2.2.3 Отмена регистрация кластера	42
2.2.4 Включение хоста виртуализации в кластер.....	43
2.2.5 Удаление хоста виртуализации из кластера.....	43
2.2.6 Получение информации о кластере.....	44
2.2.7 Получение списка кластеров	44
2.2.8 Управление связями кластера с виртуальными сетями	46

2.2.9 Управление связями кластера с хранилищами данных.....	46
2.2.10 Изменение шаблона кластера	47
2.3 Настройка планировщика	48
2.3.1 Процедура подбора вычислительных ресурсов для VM.....	48
2.3.2 Пользовательские критерии выбора вычислительных ресурсов	52
2.3.3 Типовые критерии выбора вычислительных ресурсов	53
2.3.4 Файл конфигурации планировщика.....	55
2.3.5 Запуск запланированных операций с VM.....	57
2.4 Управление хранилищами данных	57
2.4.1 Типы хранилищ данных и технологий их реализации.....	57
2.4.2 Регистрация хранилища	60
2.4.3 Переименование хранилища.....	62
2.4.4 Определение прав доступа к хранилищу.....	63
2.4.5 Отмена регистрации хранилища.....	65
2.4.6 Получение информации о хранилище	66
2.4.7 Получение списка хранилищ	67
2.4.8 Блокировка и снятие блокировки хранилища для планировщика.....	68
2.4.9 Изменение шаблона хранилища	69
3 Управление пользователями и группами	70
3.1 Типы пользователей и организация управления правами доступа	70
3.2 Управление пользователями	71
3.3 Управление группами	79
3.4 Управление виртуальными дата-центрами.....	87
3.5 Управление разрешениями на операции.....	89
3.6 Управление квотами.....	98
3.7 Формирование отчетов об использовании вычислительных ресурсов.....	105
3.8 Формирование отчетов о стоимости потребления вычислительных ресурсов виртуальной машиной.....	114
4 Управление виртуальными сетями.....	118
4.1 Виртуальные сети и операции с ними	118
4.1.1 Определение виртуальной сети	118
4.1.2 Операции с виртуальными сетями	120
4.1.3 Управление диапазонами адресов	123
4.1.4 Использование виртуальных сетей	124
4.1.5 Создание пользовательских виртуальных сетей.....	126
4.2 Виртуальные маршрутизаторы	129
4.3 Группы безопасности	134
4.4 Шаблоны виртуальных сетей	140
5 Управление виртуальными машинами	143
5.1 Образы дисков виртуальных машин.....	143
5.1.1 Типы образов дисков виртуальных машин и жизненный цикл образа ...	143
5.1.2 Управление образами	144
5.1.3 Использование образов в виртуальных машинах	151
5.2 Шаблоны виртуальных машин.....	153
5.2.1 Определение виртуальной машины	153
5.2.2 Создание виртуальной машины на основе шаблона	163
5.2.3 Управление шаблонами виртуальных машин.....	166
5.3 Управление экземплярами виртуальных машин.....	170

5.3.1	Жизненный цикл виртуальной машины	170
5.3.2	Получение информации о VM	174
5.3.3	Завершение работы и удаление VM	176
5.3.4	Временное прекращение работы VM	176
5.3.5	Перезагрузка VM	178
5.3.6	Задержка развертывания VM	179
5.3.7	Управление снимками состояния образов дисков VM	180
5.3.8	Подключение и отключение диска у работающей VM	183
5.3.9	Подключение и отключение сетевого интерфейса у работающей VM ...	184
5.3.10	Управление снимками состояния VM	184
5.3.11	Изменение объемных характеристик VM	186
5.3.12	Изменение объема дисков VM	187
5.3.13	Изменение конфигурации VM	188
5.3.14	Копирование VM	190
5.3.14.1	Создание постоянной VM	191
5.3.14.2	Сохранение VM как нового шаблона	191
5.3.15	Планирование операций с VM	192
5.3.15.1	Однократные операции с точно заданным временем выполнения	196
5.3.15.2	Периодические операции с точно заданным временем выполнения	197
5.3.15.3	Операции с относительным временем выполнения	198
5.3.16	Присвоение VM дополнительных атрибутов	199
5.3.17	Управление правами доступа к VM	199
5.3.18	Перепланирование вычислительных ресурсов, выделяемых VM	200
5.3.19	Перенос VM на другие ресурсы	201
5.3.20	Восстановление функционирования VM, ожидающей операции драйвера	202
5.3.21	Доступ к консоли VM через VNC	202
5.4	Особенности управления виртуальными машинами, основанными на виртуализации VMware	204
5.4.1	Импорт шаблонов VM и запущенных в vCenter VM	204
5.4.2	Запуск VM, поддерживаемой vCenter, на основе шаблона	205
5.4.3	Создание постоянной VM	206
5.4.4	Сохранение VM как нового шаблона	207
5.4.5	Особенности выделения вычислительных ресурсов планировщиком	207
5.4.6	Особенности операций с образами дисков	208
5.4.7	Теги и категории, как дополнительные признаки VM	209
6	Настройка виртуальных машин	210
6.1	Контекстуализация виртуальной машины	210
6.1.1	Подготовка образа гостевой ОС	210
6.1.2	Включение данных контекстуализации в шаблон VM	210
6.1.3	Использование средств vCenter для настройки VM	213
6.2	Наполнение облачной вычислительной среды	214
6.2.1	Использование готовых образов дисков с дистрибутивами ПО	215
6.2.2	Наполнение облачной вычислительной среды путем установки ПО и формирования соответствующих образов средствами TIVM	217

Список сокращений, обозначений и аббревиатур

ACPI	Advanced Configuration and Power Interface – усовершенствованный интерфейс управления конфигурацией и питанием. Стандартный интерфейс для обеспечения взаимодействия ОС, BIOS материнской платы и аппаратных средств вычислительного устройства
API	Application Program Interface – прикладной программный интерфейс
AR	Address Range – диапазон адресов
CLI	Command Line Interface – интерфейс командной строки
IaaS	Infrastructure as a Service – инфраструктура как услуга
LVM	Logical Volume Manager – менеджер логических томов
MOR	Managed Object Reference – ссылка на управляемый объект
NAS	Network Attached Storage – сетевое хранилище данных
NIC	Network Interface Controller – сетевой интерфейсный контроллер
OC CI	Open Cloud Computing Interface – открытый интерфейс взаимодействия облачных сред
QoS	Quality of Service – качество обслуживания
RDM	Raw Device Mapping – система сопоставления виртуальных и физических устройств хранения данных
SAN	Storage Area Network – сеть хранения данных
SOAP	Simple Object Access Protocol – протокол обмена сообщениями, использующий язык XML
TIVM	Tivun Intelligent Virtual Manager – сервер управления виртуализацией (менеджер виртуализации) "ТИБМ"
VMDK	Virtual Machine Disk – формат образа диска VM, используемый в технологии VMware
VNC	Virtual Network Computing – система удаленного доступа к рабочему столу компьютера, использующая протокол удаленного кадрового буфера
БД	база данных
ВДЦ	виртуальный дата-центр
ВМ	виртуальная машина
ДЦ	дата-центр
ЖЦ	жизненный цикл
ОС	операционная система
ПАМ	программный агент мониторинга
ПВК	программно-вычислительный комплекс
ПО	программное обеспечение

СТП	служба технической поддержки
СУБД	система управления базами данных
СУД	список управления доступом (Access Control List, ACL)
СХД	система хранения данных

Введение

Программно-вычислительный комплекс "Тивун Интеллектуальные Виртуальные Машины" (ПВК "ТИВМ") предназначен для управления виртуальной вычислительной инфраструктурой, включая формирование облачных вычислительных сред, предоставляемых пользователям по модели IaaS (инфраструктура как услуга). Программный инструментарий "ТИВМ" выступает в качестве информационно-управляющей и координирующей надстройки над средствами виртуализации нижележащего уровня – гипервизорами, функционирующими на хост-серверах. Он формирует интегральное представление виртуальной вычислительной инфраструктуры и реализует простые и эффективные средства для управления и мониторинга ее компонентов.

Программное обеспечение (ПО) "ТИВМ" позволяет взаимодействовать с гипервизором KVM, системой управления виртуальными ресурсами vCenter, менеджером контейнеров LXD (рис. 0.1). На физическом уровне обеспечивается поддержка серверов x86-64, а также IBM Power и OpenPower. В качестве гостевых операционных систем (ОС) могут использоваться различные дистрибутивы Linux (RedHat, SuSe, CentOS, Debian, Ubuntu, ALT Linux) и продукты семейства MS Windows Server (2008, 2012, 2016, 2019).

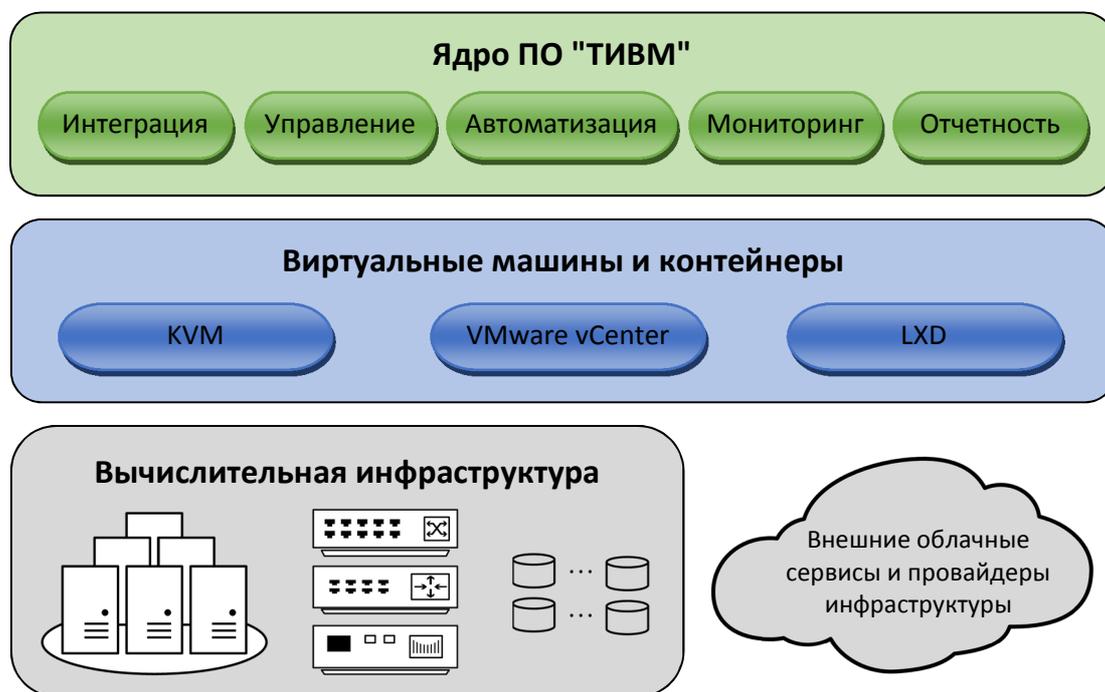


Рис. 0.1 – Упрощенная структура ПВК "ТИВМ"

Конечные пользователи ПВК "ТИВМ" – инженерный персонал, обеспечивающий техническую поддержку облачной среды: системные администраторы виртуальной вычислительной инфраструктуры, администраторы облачной вычислительной среды и облачных сервисов. ПО реализует единый интерфейс управления виртуальной вычислительной инфраструктурой, адаптируемый к зоне ответственности и уровню квалификации пользователей. Это обеспечивает доступность инструментария для персонала, не обладающего глубокими знаниями технологий виртуализации и навыками выполнения соответствующих операций на уровне гипервизоров хост-серверов и команд ОС.

ПВК "ТИВМ" имеет *полнофункциональный графический интерфейс* управления и мониторинга, работающий в среде тонкого web-клиента. Для доступа к нему достаточно типового web-браузера. В системе также предусмотрен интерфейс командной строки.

Инструментарий "ТИВМ" обеспечивает *автоматизацию* всех основных *операций управления* виртуальной вычислительной инфраструктурой:

- регистрация в системе хост-сервера (хоста виртуализации для развертывания виртуальных машин (ВМ));
- управление жизненным циклом (ЖЦ) зарегистрированных хостов виртуализации;
- определение показателей хоста виртуализации, подлежащих мониторингу;
- регистрация кластера (группы хост-серверов);
- управление составом хост-серверов, входящих в кластер;
- создание шаблонов кластеров, определяющих параметры, учитываемые при развертывании на их основе ВМ;
- управление шаблонами кластеров;
- создание виртуальной сети;
- управление виртуальными сетями;
- создание и управление группами безопасности, определяющими правила межсетевых экранов;
- создание шаблонов виртуальных сетей, определяющих их параметры и используемых для упрощения их развертывания;
- создание виртуального маршрутизатора и управление им;
- управление хранилищами данных;
- создание ВМ;
- управление ЖЦ ВМ;
- импорт ВМ, запущенных на хосте виртуализации до его регистрации в "ТИВМ";
- создание и управление образами дисков ВМ;

- создание шаблонов ВМ, определяющих их параметры и используемых для упрощения их развертывания;
- управление шаблонами ВМ;
- управление связями ВМ и кластеров с виртуальными сетями и хранилищами данных;
- управление параметрами планировщика, влияющими на распределение ВМ по хостам виртуализации, использование ими хранилищ данных, приоритеты ВМ;
- планирование операций с ВМ, которые подлежат выполнению по расписанию;
- перенос ВМ на другой хост-сервер;
- перенос всех ВМ, работающих на хосте, на другой хост;
- создание облачных сервисов, состоящих из набора взаимосвязанных ВМ, для развертывания компонентов приложений многоуровневой архитектуры;
- управление ЖЦ облачного сервиса;
- создание и управление шаблонами облачных сервисов;
- создание виртуального дата-центра (ВДЦ), представляющего логический пул вычислительных ресурсов (из одного или разных кластеров), выделяемых определенным группам пользователей;
- управление ВДЦ.

Эффект от внедрения и применения ПВК "ТИВМ" обусловлен следующими факторами:

- созданием дополнительного уровня управления виртуальной вычислительной инфраструктурой, формирующего интегральное представление ее компонентов, абстрагированное от технологий их реализации, и содержащего единый интерфейс мониторинга и управления облачной средой;
- эффективной и гибкой автоматизацией всех типовых операций управления виртуальной вычислительной инфраструктурой;
- консолидацией виртуальных вычислительных ресурсов, базирующихся на разных технологиях виртуализации, расположенных в одном или разных дата-центрах;
- существенным упрощением процедур мониторинга и управления, что позволяет их выполнять специалистам, не обладающим глубокими знаниями технологий виртуализации;
- возможностями построения виртуальной вычислительной среды с использованием ресурсов внешних облачных сервисов (благодаря поддержке базовых технологических интерфейсов).

К основным *преимуществам*, отличающим ПВК "ТИВМ" от продуктов аналогичного назначения, относятся:

- поддержка хост-серверов различной архитектуры (как x86-64, так и IBM Power и OpenPower);
- встроенная реализация поддержки систем хранения данных (СХД) корпоративного (enterprise) уровня IBM Storwize и HP 3par;
- использование высокоэффективной свободно распространяемой СУБД PostgreSQL;
- интеграция с системой мониторинга "Тивун";
- простота развертывания облачной среды в рекомендуемой конфигурации с помощью средств инсталляции, размещенных на дистрибутивном диске;
- высокая надежность виртуальной вычислительной инфраструктуры, организованной средствами ПВК "ТИВМ", работоспособность которой сохраняется даже при выходе из строя или отключении сервера управления виртуализацией;
- эффективные возможности для расширения и интеграции благодаря модульной структуре и реализации взаимодействия на основе типовых и подключаемых интерфейсов и драйверов;
- возможность развертывания виртуальной вычислительной инфраструктуры на базе одного хост-сервера, используемого как для размещения ядра "ТИВМ", СУБД, БД и хранилищ данных, так и для запуска VM.

1 Архитектура, основные возможности и технологические характеристики

1.1 Состав программного обеспечения и организация его эксплуатации

Ядром ПВК "ТИВМ" служит *сервер управления виртуализацией*, называемый *Tivun Intelligent Virtual Manager (TIVM)*. Он может быть установлен как на физическом, так и на виртуальном сервере. TIVM должен иметь сетевое соединение со всеми хост-серверами и СХД, входящими в инфраструктуру, на базе которой формируется виртуальная среда.

ПО TIVM включает компоненты, отвечающие за различные функции формирования, контроля и поддержки виртуальной вычислительной среды:

- базовый модуль управления виртуализацией;
- планировщик вычислительных ресурсов;
- модуль мониторинга;
- web-интерфейс;
- интерфейс командной строки (CLI);
- модуль управления облачными сервисами, состоящих из набора взаимосвязанных ВМ, для функционирования приложений многоуровневой архитектуры (TIVFLOW);
- интерфейс, позволяющий ПО, функционирующему на ВМ, получать данные о конфигурации и состоянии виртуализации из TIVM (TIVGATE);
- и др.

Обобщенная архитектура системы показана на рис. 1.1.

Информация о конфигурации и состоянии виртуальной вычислительной среды, а также данные мониторинга ее компонентов хранятся в *базе данных (БД)*, управляемой СУБД PostgreSQL. СУБД может быть установлена как на том же сервере, что и TIVM, так и на другом сервере.

Взаимодействие с компонентами физической инфраструктуры осуществляется через слой *драйверов*. В частности, драйвера поддерживают операции управления хостами виртуализации и ВМ, миграцию ВМ, взаимодействие с СХД, процедуры аутентификации (в том числе, основанные на взаимодействии с внешними сервисами), обмен данными мониторинга. Модульная реализация драйверов создает эффективные возможности для расширения состава поддерживаемых компонентов.

Для мониторинга вычислительной среды на хостах виртуализации и ВМ устанавливаются *программные агенты мониторинга (ПАМ)*. В соответствии с установленными правилами они направляют данные о состоянии контролируемых ими компонентов модулю мониторинга в составе TIVM. Кроме того, ПАМ используются для получения данных о

вычислительных ресурсах, включаемых в инфраструктуру виртуализации (механизм автоматического обнаружения ресурсов).

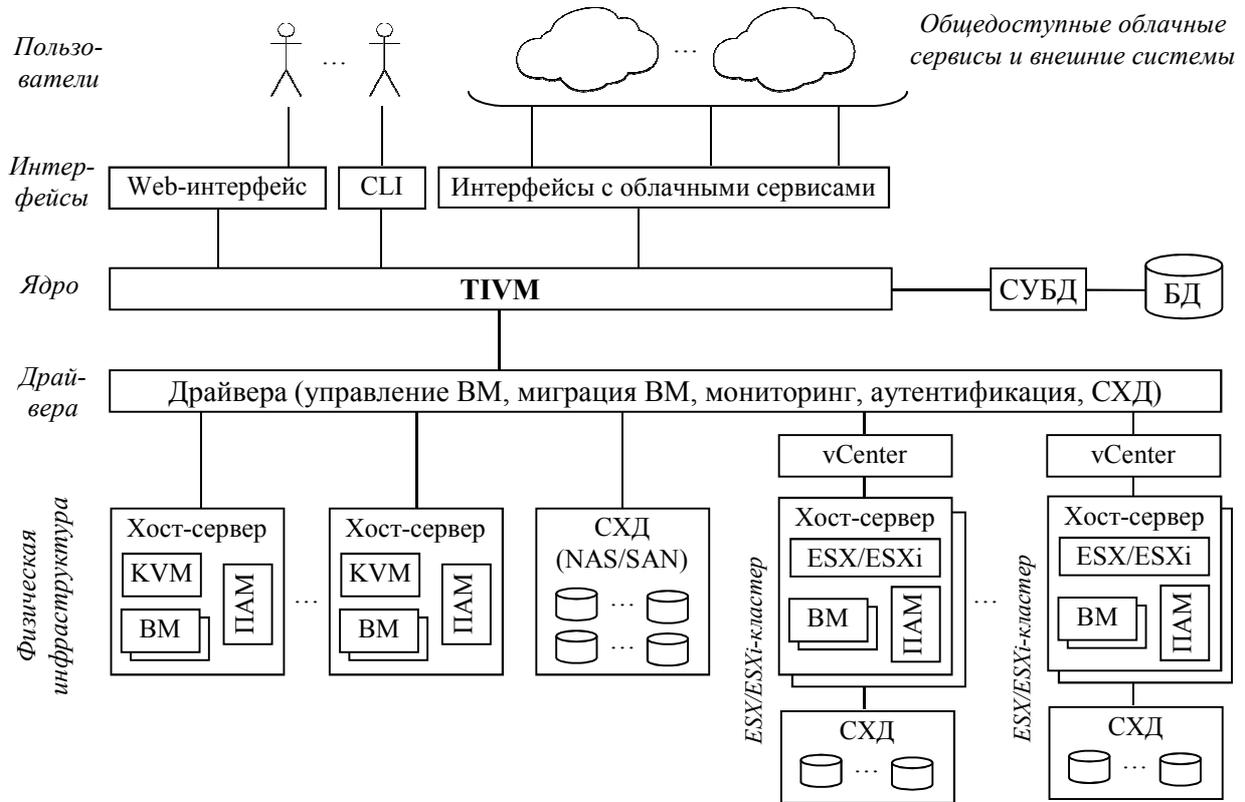


Рис. 1.1 – Обобщенная архитектура ПКВ "ТИВМ"

Модуль мониторинга TIVM способен взаимодействовать с системой мониторинга "Тивун". Однако данная система не входит в дистрибутив ПКВ "ТИВМ" и поставляется отдельно.

Операции с данными ВМ (развертывание, создание образа, миграция, создание снимка (snapshot) и др.) используют понятие *хранилища данных*. Эффективной технической воплощением хранилища служит СХД (в виде SAN- или NAS-реализации).

Для работы TIVM необходимы, как минимум, 2 хранилища данных:

- *хранилище образов*, служащее репозиторием образов дисков ВМ;
- *системное хранилище*, содержащее данные дисков работающих ВМ.

При развертывании ВМ, ее запуске после остановки или присоединении диска к работающей ВМ соответствующий образ копируется из репозитория образов в системное хранилище и становится доступным для данной ВМ. При остановке ВМ или создании снимка ее диска создается образ, сохраняемый в хранилище образов.

Слой *интерфейсов* обеспечивает взаимодействие TIVM с пользователями и внешними системами. Реализация программных интерфейсов дает возможность использования инфраструктуры публичных облачных сервисов, а также сопряжения с другими системами управления виртуальными средами.

TIVM позволяет формировать и управлять *неоднородной облачной инфраструктурой*, сочетающей разные технологии виртуализации. Вычислительная инфраструктура может быть представлена хост-серверами с гипервизорами KVM, контейнеризацией LXD или виртуализацией VMware, а также любой комбинацией этих компонентов. Кроме того, TIVM поддерживает использование как локальных (корпоративных) вычислительных ресурсов, так и внешних облачных сервисов, а также их сочетание (гибридные виртуальные среды).

Компоненты вычислительной инфраструктуры, управляемые TIVM, должны быть связаны *физической служебной сетью* (рис. 1.2). При запуске VM ее сетевые интерфейсы подключаются к интерфейсам гипервизора в соответствии с конфигурацией виртуальных сетей, определенной в TIVM. Это обеспечивает требуемую сетевую связанность VM.

TIVM позволяет поддерживать множество *виртуальных сетей*, функционирующих поверх служебной сети. Конфигурация виртуальной сети включает имя, тип реализации, с которым связан сетевой драйвер, логическое адресное пространство, контекстные атрибуты, передаваемые VM при ее запуске (адреса DNS и шлюза, сетевая маска), и другие параметры.

Для маршрутизации в виртуальных служат *виртуальные маршрутизаторы*. Реализацией такого маршрутизатора служит VM, создаваемая на основе образа, который содержится в хранилище образов, поставляемом с TIVM.

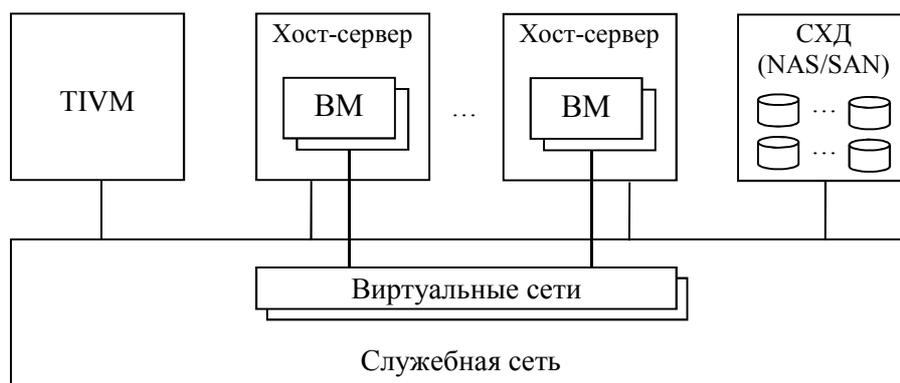


Рис. 1.2 – Сетевая связанность компонентов виртуальной инфраструктуры

Дистрибутив ПО "TIVM" предоставляется на установочном диске.

Сервер управления виртуализацией TIVM устанавливается на физический или виртуальный сервер с ОС Debian версии 10.x. Существует возможность формирования сборки ПО, рассчитанной на другие дистрибутивы Linux, по заказу пользователей.

Для работы системы также требуется СУБД PostgreSQL версии 11 или выше. Она может быть установлена на том же сервере, что и TIVM, или на отдельном сервере. Простейший вариант развертывания ПВК "ТИВМ" предусматривает установку всех его компонентов на один сервер и использование его ресурсов для запуска VM.

Подробные инструкции по установке ПО "ТИВМ" приведены в отдельном руководстве.

1.2 Требования к вычислительным платформам. Программная и информационная совместимость

Реализация ПВК "ТИВМ" является платфо́рмо-независимой. Управляющий сервер TIVM рассчитан на работу под управлением ОС семейства Linux (рекомендуемый вариант – ОС Debian 10.x; существует возможность формирования сборки ПО по заказу пользователей, рассчитанной на другие дистрибутивы Linux). Возможность запуска TIVM на VM позволяет использовать для него любые серверные вычислительные ресурсы.

Для упрощения инсталляции и настройки ПО "ТИВМ" поставляется в виде *дистрибутива*, включающего все необходимые компоненты для установки на сервер с ОС Debian версии 10.x.

ПВК "ТИВМ" может применяться в сопряжении с гипервизором KVM, системой управления виртуальными ресурсами vCenter, менеджером контейнеров LXD. Система позволяет управлять облачными вычислительными средами со множеством различных гипервизоров (сочетаний хост-серверов, использующих KVM, vCenter и LXD).

Так как операции с физической инфраструктурой осуществляются через слой драйверов, ПВК "ТИВМ" не имеет принципиальных ограничений по ее поддержке. Возможность использования в виртуальной среде того или иного вычислительного ресурса достигается включением в конфигурацию системы соответствующего драйвера. Данный механизм создает условия для расширяемости TIVM в части задействуемых внутренних (корпоративных) вычислительных ресурсов. В свою очередь, возможности использования внешних облачных сервисов, а также логической интеграции и межсистемного взаимодействия обеспечивает реализация базовых технологических интерфейсов (см. рис. 1.1).

Минимальные вычислительные ресурсы для функционирования управляющего сервера TIVM:

- ядра процессора – 4;
- оперативная память – 8 Гб;
- дисковое пространство – 64 Гб (из них 10 Гб – ПО ОС, TIVM и СУБД, 24 Гб – БД);
- сетевые интерфейсы – 2.

Показатели производительности операций управления и мониторинга виртуальной вычислительной среды зависят от числа ВМ, интенсивности запросов мониторинга, направляемых ПАМ к TIVM, а также объема физических ресурсов, выделенных TIVM и ВМ. Критическое влияние на производительность оказывают характеристики СХД. Во многих случаях именно СХД является узким местом при расширении виртуальной среды.

Управляющий сервер TIVM в минимальной конфигурации, которая приведена выше, позволяет управлять виртуальной инфраструктурой VMware, включающей 4 экземпляра vCenter, 160 ESX/ESXi-серверов (до 40 на каждый экземпляр vCenter) и 1000 ВМ (до 250 на каждый экземпляр vCenter).

В любом случае рекомендуется, чтобы на один экземпляр TIVM приходилось не более 2500 управляемых хост-серверов и 10000 ВМ. При необходимости управления большим числом хост-серверов и (или) ВМ следует разворачивать несколько экземпляров TIVM, осуществляющих репликацию данных и совместно использующих данные конфигурации и пользователей.

При планировании вычислительных ресурсов, представленных хост-серверами, необходимо иметь в виду, что функционирование гипервизоров KVM и ESX/ESXi требует около 10% памяти хоста.

Каждый хост-сервер должен иметь, как минимум, 3 сетевых интерфейса. Еще один сетевой интерфейс на сервере TIVM и каждом хост-сервере требуется при использовании сети хранения данных (SAN).

Информация мониторинга вычислительной инфраструктуры сохраняется в БД управляющего сервера TIVM. Рекомендуется разместить эту БД в СХД. В зависимости от масштаба инфраструктуры и настроек мониторинга эти данные могут иметь значительный объем, который должен быть зарезервирован.

Размер записи о состоянии хост-сервера – около 2 Кб, размер записи о состоянии ВМ – около 4 Кб. Частота фиксации данных определяется параметром `MONITORING_INTERVAL_HOST`. Его значение по умолчанию – 180 сек. Времена хранения данных о состоянии хост-сервера и ВМ фиксируют параметры `HOST_MONITORING_EXPIRATION_TIME` и `VM_MONITORING_EXPIRATION_TIME`. Их значения по умолчанию – 12 и 4 часа соответственно.

Продукт поддерживает следующие хост-платформы:

- для виртуализации на основе KVM – Red Hat Enterprise Linux (v. 7, 8), CentOS (v. 7, 8), Ubuntu Server (v. 16.04, 18.04, 20.04), Debian (v. 9, 10, 11);
- для виртуализации на основе LXD – Ubuntu Server (v. 16.04, 18.04, 20.04), Debian (v. 10, 11);
- виртуализация на основе vCenter – v. 6.0, 6.5, 6.7, 7.0.

1.3 Категории пользователей и управление доступом

Пользователи ПВК "ТИВМ" – инженерный персонал, обеспечивающий техническую поддержку облачной среды. В зависимости от решаемых задач и уровня владения технологиями виртуализации пользователей можно разделить на следующие 3 *категории*, с каждой из которых соотносятся функциональные роли, определяющие права доступа и применяемые представления web-интерфейса TIVM.

1. Системные администраторы виртуальной вычислительной инфраструктуры.

В данную категорию входят специалисты, хорошо знакомые с технологиями виртуализации, отвечающие за управление физической вычислительной инфраструктурой, контроль и поддержку ее состояния, подготовку базовых средств виртуализации, распределение вычислительных ресурсов между потребителями (виртуальными средами и группами пользователей), управление пользователями "ТИВМ". Системные администраторы имеют полные права доступа и максимально подробные представления TIVM. К числу решаемых ими задач относятся:

- настройка, контроль состояния и техническая поддержка компонентов вычислительной инфраструктуры, используемых в виртуальных средах (хост-серверов, СХД, телекоммуникационного оборудования, системного ПО);
- регистрация в TIVM и управление инфраструктурными ресурсами (хост-сервера, их кластеры, хранилища данных, виртуальные сети);
- создание и управление образами ВМ;
- создание и управление шаблонами виртуальных ресурсов (ВМ, кластеры, виртуальные сети);
- настройка параметров мониторинга;
- управление пользователями и их группами;
- создание и управление ВДЦ;
- создание и управление облачными сервисами (наборами взаимосвязанных ВМ);

- настройка параметров планировщика (определение политики распределения вычислительных ресурсов по ВМ, использования хранилищ данных, приоритеты ВМ);
- управление квотами на вычислительные ресурсы для пользователей и групп;
- организация взаимодействия TIVM с внешними системами (включая публичные облачные сервисы);
- настройка функционирования кластера TIVM с репликацией данных конфигурации и пользователей.

В организационном плане системные администраторы виртуальной вычислительной инфраструктуры могут представлять как компанию-пользователя, так и провайдера вычислительных ресурсов (например, дата-центр (ДЦ), сдающий в аренду виртуальные сервера).

2. Администраторы облачной вычислительной среды.

Это специалисты, обеспечивающие формирование и поддержку виртуальных вычислительных ресурсов на уровне организации-пользователя. Они также хорошо знакомы с технологиями виртуализации. TIVM предоставляет им те же возможности, что и системным администраторам, за исключением операций с физической вычислительной инфраструктурой. Кроме того, права доступа данной категории ограничены ресурсами и квотами, выделенными их группе. Основные решаемые задачи:

- создание (запуск) ВМ;
- все операции управления ЖЦ ВМ;
- создание и управление шаблонами ВМ;
- создание и управление образами ВМ;
- создание и управление виртуальными сетями и виртуальными маршрутизаторами;
- создание и управление шаблонам виртуальных сетей;
- создание и управление облачными сервисами (наборами взаимосвязанных ВМ).

3. Администраторы облачных сервисов.

В данную категорию входят пользователи виртуальных вычислительных ресурсов (администраторы виртуальных серверов). От этих специалистов не требуются глубокие знания технологий виртуализации, достаточно иметь общие представления о них.

Для администраторов облачных сервисов в TIVM предусмотрены упрощенные интерфейсные формы, реализующие обращения к базовым функциям управления ВМ на основе доступных шаблонов и образов. Создание и редактирование шаблонов, формирование образов ВМ с нуля и изменение настроек компонентов виртуальной инфраструктуры не

предусматриваются. Права доступа ограничены ресурсами и квотами, выделенными соответствующей группе пользователей.

Основные задачи из области ответственности администраторов облачных сервисов:

- создание (запуск) ВМ на основе шаблона и образа, сформированных системным администратором или администратором облачной вычислительной среды;
- базовые операции управления ВМ (выключение и включение, перезагрузка, удаление, создание снимка, сохранение образа запущенной ВМ);
- создание (запуск) ВМ на основе ранее сохраненного образа;
- получение информации о конфигурации и состоянии ВМ;
- удаление сохраненных ранее образов и снимков ВМ.

Три категории пользователей, охарактеризованные выше, соотносятся с четырьмя функциональными ролями TIVM (табл. 1.1).

Табл. 1.1 – Категории пользователей и функциональные роли TIVM

Категория пользователей	Функциональная роль
Системный администратор виртуальной вычислительной инфраструктуры	Системный администратор
Администратор облачной вычислительной среды	Администратор облака
Администратор облачных сервисов	Пользователь
	Администратор группы

Роли "Пользователь" и "Администратор группы" соответствуют одной и той же категории. Администратору группы дополнительно предоставляются возможности:

- управления пользователями группы;
- управления ВМ, их образами и снимками, созданными пользователями группы.

Таким образом, управление пользователями TIVM распределено по двум уровням: системные администраторы регистрируют группы и их администраторов, а администраторы групп – пользователей внутри них. Группа может иметь несколько администраторов.

Возможности администратора группы управлять виртуальными ресурсами всех ее членов исключают проблему блокировки доступа к ресурсам из-за монопольного владения ими конкретными пользователями.

1.4 Функциональная структура

Определение функций ПКВ "ТИВМ" базируется на нескольких признаках (рис. 1.3):

- целевом назначении;
- потребителе;
- реализующем компоненте ПО.

По назначению выделяются 10 групп функций, состав которых приведен в табл. 1.2.

Функции ПКВ "ТИВМ"

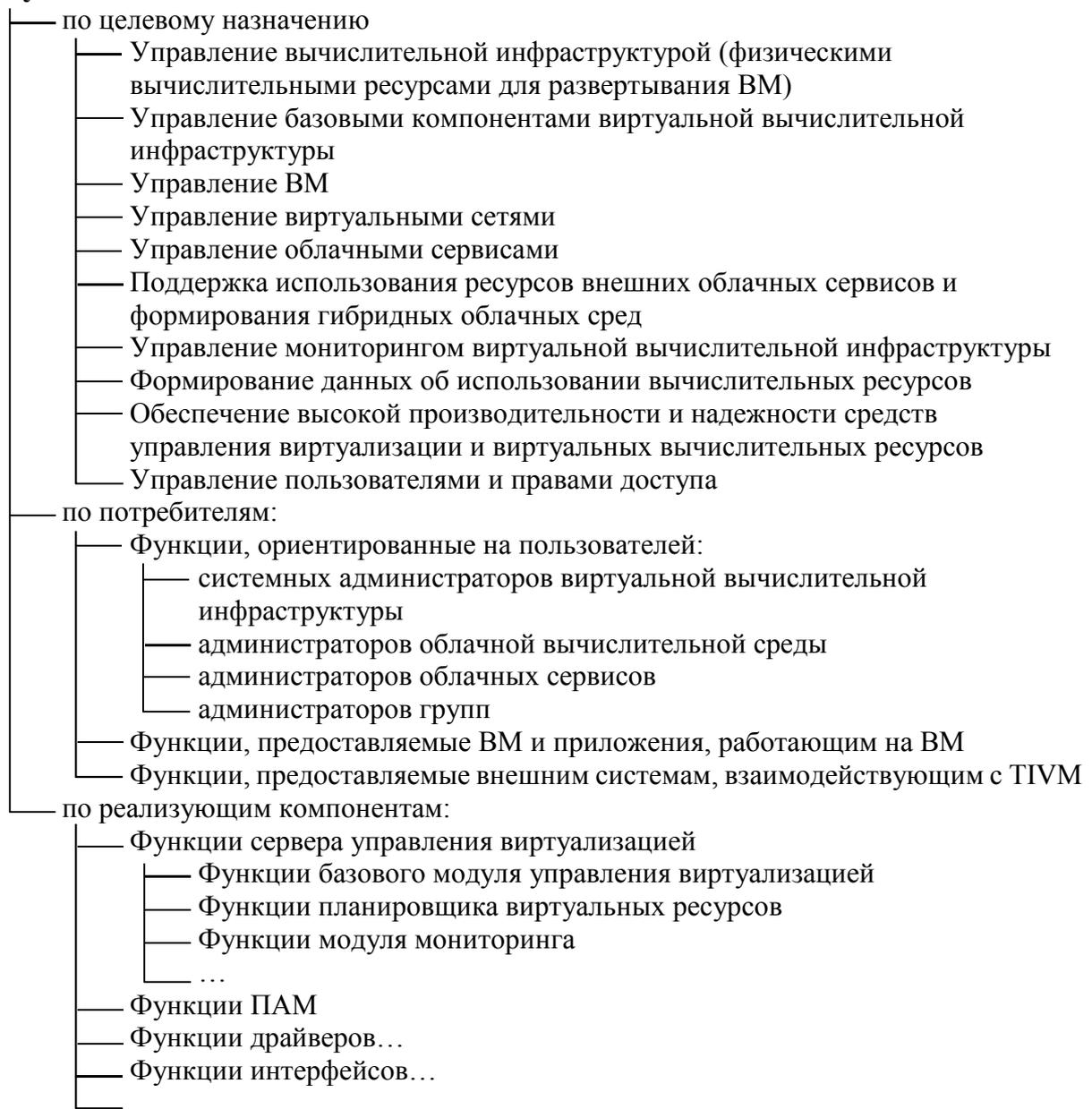


Рис. 1.3 – Функциональная структура "ТИВМ"

Потребитель – это пользователь определенной категории или техническая система за рамками "ТИВМ", в интересах которой выполняется функция.

Реализацию функций обеспечивают компоненты архитектуры "ТИВМ" (см. разд. 1.1): сервер управления виртуализацией, ПАМ, драйвера и интерфейсы.

Табл. 1.2 – Основные функции ПКВ "ТИВМ" по их назначению

Группа функций	Функция
Управление вычислительной инфраструктурой (физическими вычислительным и ресурсами для развертывания VM)	<ul style="list-style-type: none"> • регистрация в системе сервера (хоста виртуализации); • управление ЖЦ хостов виртуализации; • регистрация кластера (группы хост-серверов); • управление составом хост-серверов, входящих в кластер; • управление хранилищами данных;
Управление базовыми компонентами виртуальной вычислительной инфраструктуры	<ul style="list-style-type: none"> • создание и управление образами дисков VM; • создание и управление шаблонами VM; • создание и управление шаблонами кластеров; • управление каталогами шаблонов и образов VM (с обеспечением совместного доступа к ним для разных экземпляров TIVM); • управление параметрами планировщика, влияющими на распределение VM по хостам виртуализации, использование ими хранилищ данных, приоритеты VM; • создание и управление ВДЦ; • управление вычислительными ресурсами, выделяемыми пользователям, их группам и ВДЦ (включая управление квотами на вычислительные ресурсы); • управление связями VM и кластеров с виртуальными сетями и хранилищами данных;
Управление VM	<ul style="list-style-type: none"> • создание VM на основе шаблона и образов дисков, представленных в хранилище; • передача пользовательских параметров конфигурации VM при ее создании; • управление ЖЦ VM (выключение, включение, перезагрузка, удаление, создание снимка, сохранение образа запущенной VM); • изменение вычислительных ресурсов, выделенных запущенной VM; • планирование операций с VM, которые подлежат выполнению по расписанию; • импорт VM, запущенных на хосте виртуализации до его регистрации в системе; • перенос VM на другой хост-сервер; • перенос всех VM, работающих на хосте, на другой хост;

Группа функций	Функция
Управление виртуальными сетями	<ul style="list-style-type: none"> • передача параметров ВМ работающим в ней приложениям; • создание и управление виртуальными сетями; • создание и управление виртуальными маршрутизаторами; • создание и управление группами безопасности, определяющими правила межсетевых экранов; • создание и управлениями шаблонами виртуальных сетей;
Управление облачными сервисами	<ul style="list-style-type: none"> • создание облачного сервиса, состоящих из набора взаимосвязанных ВМ, для развертывания компонентов приложений многоуровневой архитектуры; • управление ЖЦ облачного сервиса; • создание и управление шаблонами облачных сервисов; • управление правилами развертывания ВМ и автоматического масштабирования вычислительных ресурсов для облачного сервиса;
Поддержка использования ресурсов внешних облачных сервисов и формирования гибридных облачных сред	<ul style="list-style-type: none"> • настройка взаимодействия системы с внешними облачными сервисами; • управление виртуальными вычислительными ресурсами, предоставляемыми внешними облачными сервисами;
Управление мониторингом виртуальной вычислительной инфраструктуры	<ul style="list-style-type: none"> • определение характеристик хоста виртуализации, хранилища данных, ВМ, подлежащих мониторингу; • сбор, хранение и представление данных мониторинга функционирования физических и виртуальных вычислительных ресурсов; • контроль событий по данным мониторинга; • получение данных о вычислительных ресурсах, включаемых в инфраструктуру виртуализации (автоматическое обнаружение ресурсов); • получение данных мониторинга из внешних систем;
Формирование данных об использовании вычислительных ресурсов	<ul style="list-style-type: none"> • сбор, хранение и выгрузка записей истории функционирования ВМ, а также данных об использовании ими вычислительных ресурсов; • представление отчетных данных и диаграмм в пользовательском интерфейсе; • определение удельной стоимости использования вычислительных ресурсов и расчет стоимости их потребления с формированием соответствующих отчетов и возможностью интеграции с платежными системами;
Обеспечение высокой производительности и надежности средств	<ul style="list-style-type: none"> • настройка средств управления виртуализацией в виде отказоустойчивого кластера, поддерживающего согласованность данных о вычислительной среде;

Группа функций	Функция
управления виртуализации и виртуальных вычислительных ресурсов	<ul style="list-style-type: none"> • репликация данных о вычислительной среде между серверами управления виртуализацией, входящими в кластер; • поддержка совместного использования серверами управления виртуализацией, входящими в кластер, данных о конфигурации, пользователях, их группах и правах доступа; • автоматический контроль отказов ВМ и физических ресурсов с запуском процедур их восстановления (перераспределения нагрузки);
Управление пользователями и правами доступа	<ul style="list-style-type: none"> • управление пользователями, их группами и правами доступа; • аутентификация пользователей (с помощью внутренних или внешних средств); • изоляция и разделение вычислительных ресурсов, предоставляемых пользователям и группам;

1.5 Типовые сценарии применения

1.5.1 Использование провайдером виртуальных вычислительных ресурсов

Организация-пользователь – провайдер виртуальных вычислительных ресурсов, предоставляющий услуги по аренде ВМ и СХД. В распоряжении пользователя имеется парк хост-серверов и СХД, связанных физической сетью и подключенных к Интернет-каналу. Пользователю требуется эффективный и удобный инструмент управления виртуализацией, позволяющий интегрировать ее базовые средства (гипервизоры нижнего уровня), обеспечить централизованное оперирование ими, включая выделение ресурсов клиентам, распределение прав их доступа, учет потребления ресурсов, контроль функционирования физической инфраструктуры и основанных на ней виртуальных сред. Инструментарий также должен содержать простые средства самообслуживания клиентов, реализующий базовые операции управления предоставленными им виртуальными ресурсами.

Возможны 2 варианта данного сценария. В первом у провайдера нет никаких других средств централизованного управления виртуализацией уровня "ТИБМ". Во втором провайдер использует такие средства (в частности, VMware vCenter), но при этом актуальны задачи обеспечения интеграции и централизованного управления виртуальными ресурсами, а также предоставления клиентам удобного интерфейса самообслуживания.

1.5.1.1 Реализация высокоуровневых средств управления виртуализацией и интерфейса самообслуживания клиентов

Хост-сервера провайдера используют гипервизоры KVM и LXU. ПО "ТИБМ" взаимодействует с ними, формируя интегральный интерфейс

управления виртуальной инфраструктурой, абстрагированный от технологий реализации виртуальных вычислительных ресурсов. Инженерный персонал провайдера применяет TIVM для конфигурирования и мониторинга виртуальной инфраструктуры, подготовки ее базовых компонентов, создания и управления ВДЦ, учета их владельцев (клиентов), распределения по ним виртуальных ресурсов, сбора данных о потреблении ресурсов.

Клиенты провайдера регистрируются в TIVM в качестве администраторов облачных сервисов. Предоставляемый им web-интерфейс TIVM реализует функции самообслуживания, позволяющие выполнять базовые операции управления VM, контролировать их работу, фиксировать потребление вычислительных ресурсов.

1.5.1.2 Консолидация разнородных виртуальных вычислительных ресурсов и обеспечение централизованного управления ими

Провайдер использует средства виртуализации VMware (гипервизоры ESX/ESXi, систему управления vCenter). Также на его серверах могут быть установлены и другие гипервизоры (KVM, LXU). Вычислительные ресурсы провайдера могут располагаться в разных ДЦ.

Применение ПО "TIVM" обеспечивает:

- консолидацию виртуальных вычислительных ресурсов, базирующихся на разных технологиях виртуализации, расположенных в одном или разных ДЦ;
- совместное использование всеми средствами управления виртуализацией данных о конфигурации виртуальной инфраструктуры, пользователях, их группах и правах доступа;
- централизованное управление и мониторинг виртуальной вычислительной инфраструктуры.

Как и в предыдущем сценарии, TIVM предоставляет инженерному персоналу провайдера развитые функции управления и мониторинга. В свою очередь, клиентам провайдера предоставляется простой web-интерфейс самообслуживания.

1.5.2 Управление виртуализацией на основе вычислительных ресурсов организации

Организация-пользователь владеет собственными вычислительными ресурсами (серверами, СХД, сетью с выходом в Интернет), на базе которых необходимо развернуть виртуальную инфраструктуру. Организации требуется эффективный и удобный инструмент управления виртуализацией, позволяющий интегрировать ее базовые средства

(гипервизоры нижнего уровня), обеспечить централизованное оперирование ими, включая выделение ресурсов внутренним (прикладным) пользователям (подразделениям и их сотрудникам), распределение прав их доступа, учет потребления ресурсов, контроль функционирования физической инфраструктуры и основанных на ней виртуальных сред. Инструментарий также должен содержать простые средства самообслуживания прикладных пользователей, реализующий базовые операции управления предоставленными им виртуальными ресурсами.

Возможны 2 варианта данного сценария. В первом у организации, владеющей вычислительной инфраструктурой, нет никаких других средств централизованного управления виртуализацией уровня "ТИВМ", а имеющаяся вычислительная инфраструктура размещена в одном ДЦ и достаточна для удовлетворения бизнес-потребностей прикладных пользователей. Во втором организация уже применяет высокоуровневые средства управления виртуализацией (VMware vCenter), но нуждается в интеграции ресурсов, размещенных в разных ДЦ или базирующихся на разных технологиях виртуализации, либо испытывает недостаток вычислительных ресурсов, который планирует закрыть за счет внешних облачных сервисов.

1.5.2.1 Реализация высокоуровневых средств управления виртуализацией и интерфейса самообслуживания прикладных пользователей

Хост-сервера организации используют гипервизоры KVM и LXDM. ПО "ТИВМ" взаимодействует с ними, формируя интегральный интерфейс управления виртуальной инфраструктурой, абстрагированный от технологий реализации виртуальных вычислительных ресурсов. Системные администраторы организации применяют TIVM для конфигурирования и мониторинга виртуальной вычислительной инфраструктуры, подготовки ее базовых компонентов, управления прикладными пользователями и их группами, распределения по ним виртуальных ресурсов, сбора данных о потреблении ресурсов.

Прикладные пользователи организации регистрируются в TIVM в качестве администраторов облачных сервисов. Предоставляемый им web-интерфейс TIVM реализует функции самообслуживания, позволяющие выполнять базовые операции управления VM, контролировать их работу, фиксировать потребление вычислительных ресурсов.

1.5.2.2 Консолидация разнородных виртуальных вычислительных ресурсов, включая привлечение ресурсов внешних облачных сервисов

Парк серверов организации использует различные технологии виртуализации: VMware (ESX/ESXi, vCenter), KVM, LXD. Вычислительные ресурсы могут располагаться в разных ДЦ. Недостаток собственных вычислительных ресурсов планируется компенсировать за счет привлечения ресурсов внешних облачных сервисов.

Применение ПО "ТИВМ" обеспечивает:

- консолидацию виртуальных вычислительных ресурсов, базирующихся на разных технологиях виртуализации, расположенных в одном или разных ДЦ;
- совместное использование всеми средствами управления виртуализацией данных о конфигурации виртуальной инфраструктуры, прикладных пользователях, их группах и правах доступа;
- централизованное управление и мониторинг виртуальной вычислительной инфраструктуры.

Как и в предыдущем сценарии, TIVM предоставляет системным администраторам организации развитые функции управления и мониторинга. В свою очередь, прикладным пользователям организации предоставляется простой web-интерфейс самообслуживания.

1.6 Варианты формирования облачных вычислительных сред

Архитектура "ТИВМ" предусматривает множество вариантов организации облачных вычислительных сред, отличающихся:

- применяемыми базовыми технологиями виртуализации (KVM, VMware, LXD);
- размещением используемых вычислительных ресурсов (ресурсы в ДЦ организации-владельца, ресурсы внешних облачных сервисов, гибридные облачные среды);
- размещением вычислительных ресурсов в одном или множестве ДЦ, управляемых одним сервером TIVM;
- использованием одного или множества экземпляров сервера управления виртуализацией TIVM, формирующих кластер с единым интерфейсом управления.

Для организации, располагающей собственными вычислительными ресурсами, достаточными для поддержки ее бизнес-процессов, базовым решением является *архитектура с локальным размещением ресурсов и средств управления виртуализацией "ТИВМ"*. Соответствующие схемы для технологий виртуализации KVM и VMware приведены на рис. 1.4 и 1.5.

При использовании технологии виртуализации VMware сервер TIVM управляет хост-серверами не напрямую, а через vCenter. Кластер ESX/ESXi-хостов, управляемый vCenter, рассматривается TIVM как агрегированный гипервизор. По сравнению с виртуализацией на основе KVM и LXDM такое решение проще в настройке, поскольку и ESX/ESXi-хосты, и узел vCenter не требуют установки дополнительных программных компонентов. Все запросы TIVM направляются через SOAP API и драйвер vCenter. При этом TIVM имеет прямой доступ к хостам виртуализации и запущенным на них VM через средства VNC и VRMC.

Если имеющихся в организации вычислительных ресурсов недостаточно, и вместо наращивания собственного парка серверов и СХД выгодней арендовать их во внешних облачных сервисах, то TIVM позволяет сформировать гибридную облачную среду с централизованным управлением виртуальной инфраструктурой независимо от физического размещения ее компонентов (рис. 1.6). Локальные вычислительные ресурсы, установленные в ДЦ организации, могут использовать разные технологии виртуализации (KVM, LXDM, VMware). Взаимодействие с внешними облачными сервисами поддерживают соответствующие интерфейсы.

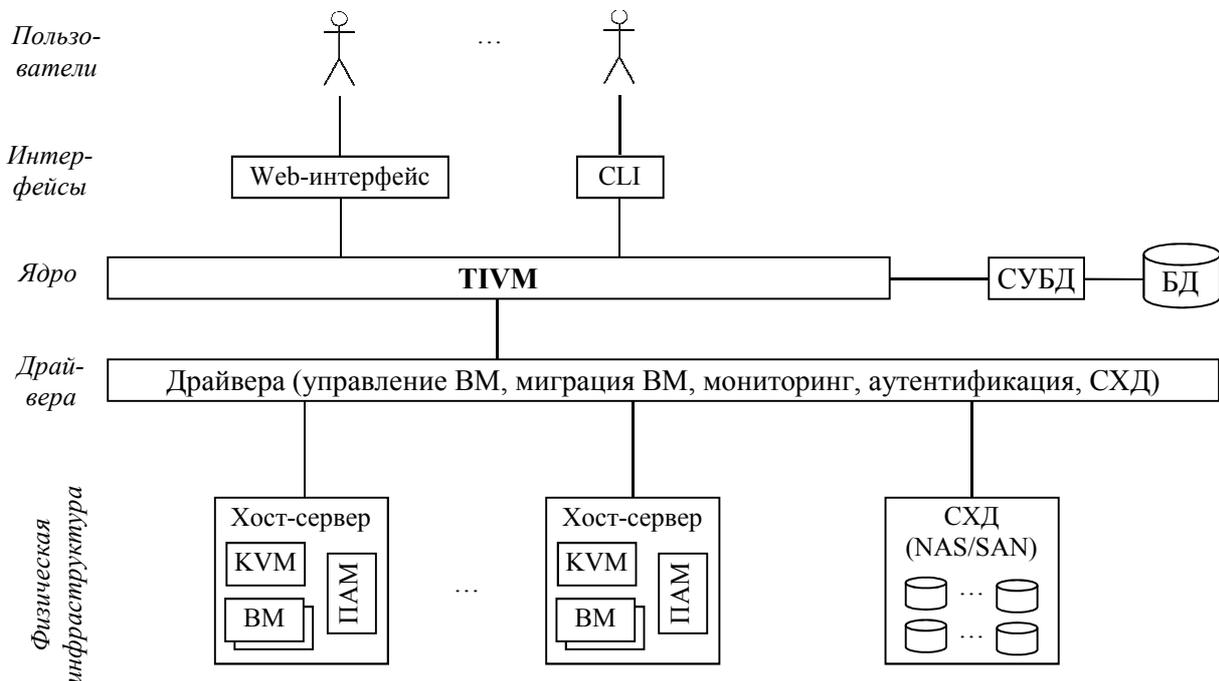


Рис. 1.4 – Облачная среда с локальным размещением вычислительных ресурсов, использованием гипервизоров KVM и одним сервером TIVM

Для организации, использующей технологию виртуализации VMware, данный вариант архитектуры обеспечивает расширяемость облачной среды за счет подключения компонентов, основанных на других (в первую очередь, открытых) технологиях. При этом сохраняются все ранее внедренные решения VMware.

Физическая вычислительная инфраструктура организации может быть размещена как в одном, так и в *нескольких ДЦ* (рис. 1.7). С технологической точки зрения этот вариант повторяет предыдущий:

- сервер TIVM обеспечивает централизованное управление формируемой облачной средой независимо от физического размещения ее компонентов;
- виртуальные вычислительные ресурсы могут базироваться на разных технологиях виртуализации (KVM, LXD, VMware);
- взаимодействие с ними TIVM обеспечивают облачные интерфейсы.

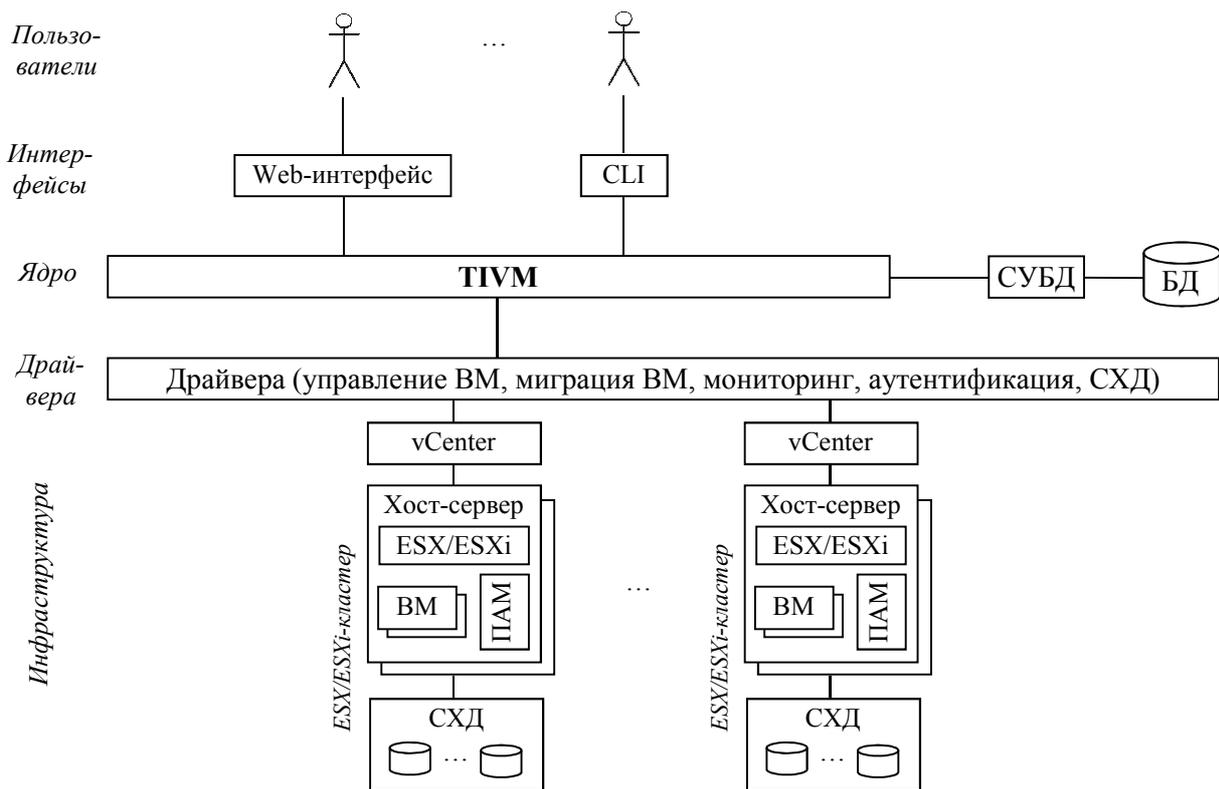


Рис. 1.5 – Облачная среда с локальным размещением вычислительных ресурсов, использованием средств виртуализации VMware и одним сервером TIVM

Организационные различия обусловлены принадлежностью вычислительных ресурсов, формирующих облако. Управление собственными (локальными) и арендованными (внешними) ресурсами базируется на одних и тех же интерфейсных решениях.

Количество хост-серверов и ВМ, которыми может управлять один экземпляр TIVM, зависит от ресурсов, выделенных TIVM и СУБД, а также производительности СХД. Ограничения масштабирования виртуальной инфраструктуры с одним сервером TIVM приведены в разд. 1.2. Выйти за их границы позволяет архитектура со множеством TIVM-серверов, образующих *отказоустойчивый кластер* (рис. 1.8).

Каждый экземпляр TIVM управляет своим пулом виртуальных ресурсов. Этот сегмент облачной среды называется *зоной*.

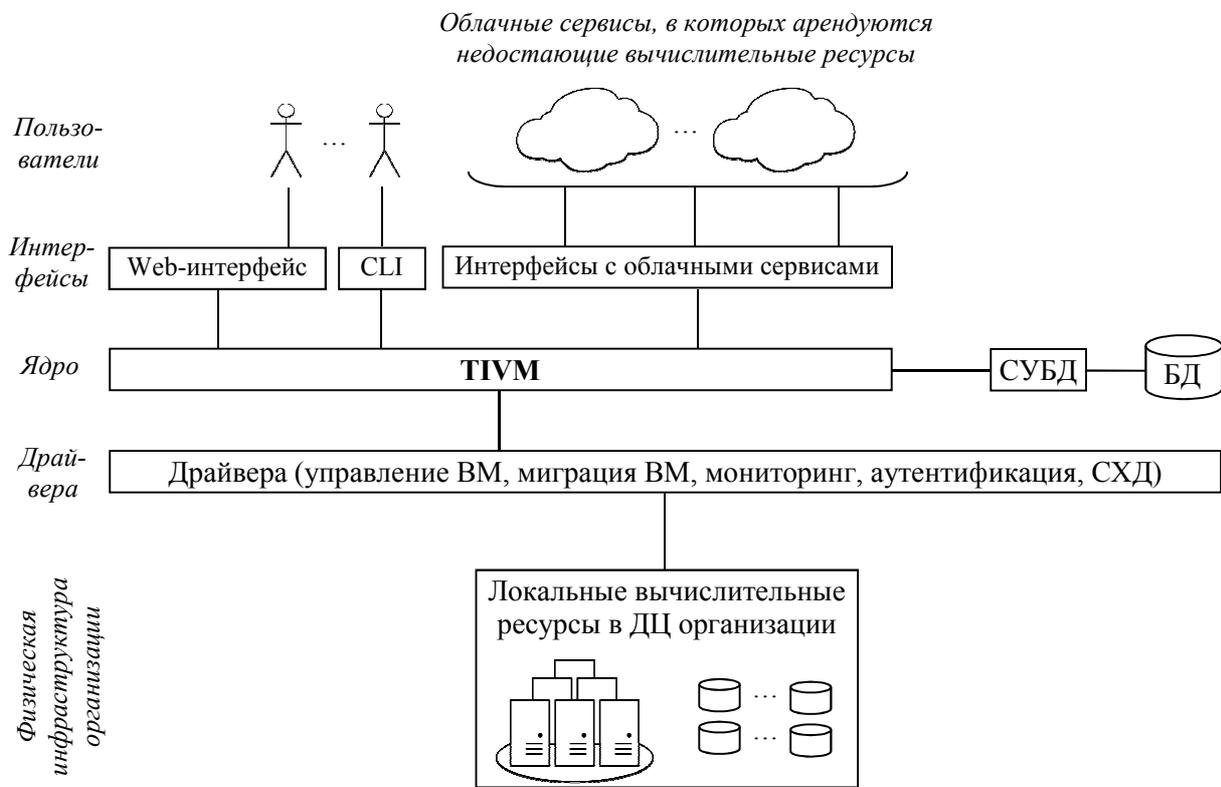
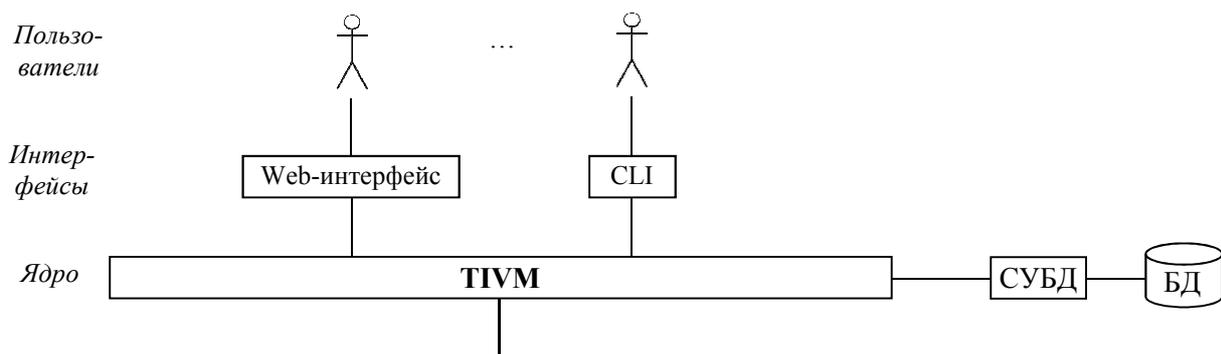


Рис. 1.6 – Гибридная облачная среда с одним сервером TIVM



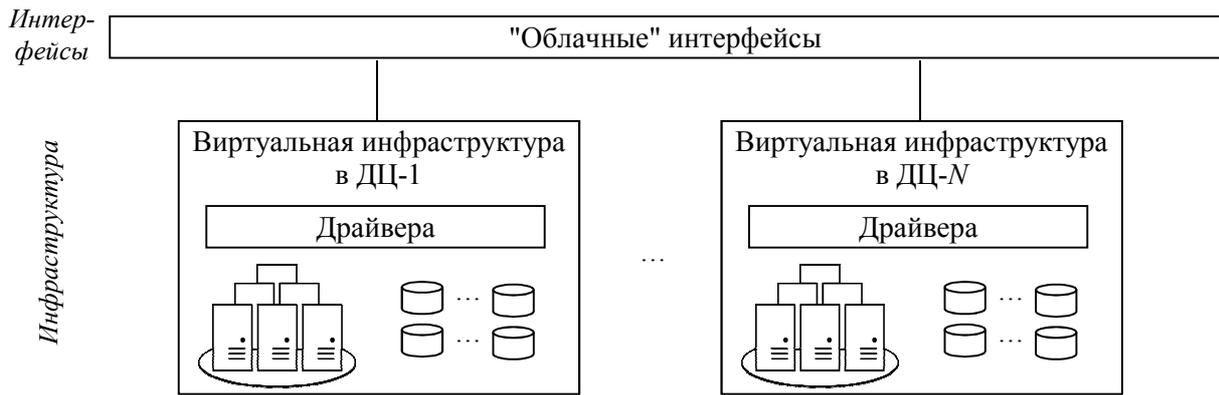


Рис. 1.7 – Облачная среда с размещением вычислительных ресурсов в разных ДЦ и одним сервером TIVM

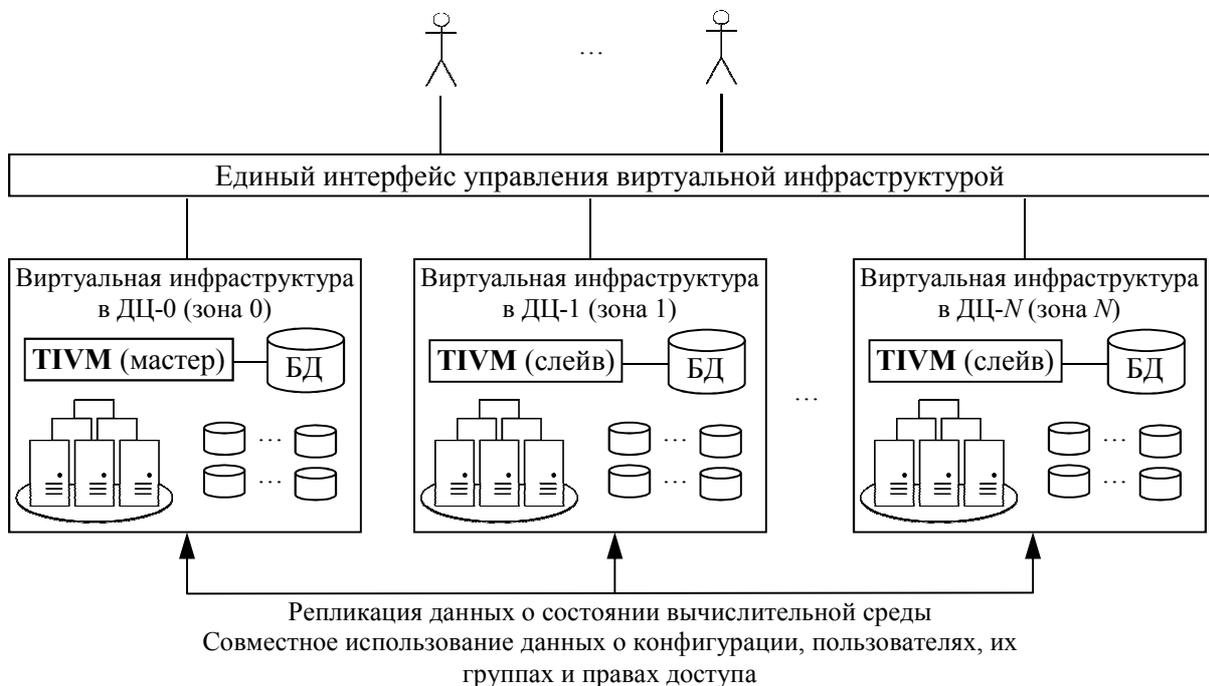


Рис. 1.8 – Облачная среда на основе кластера TIVM

В рамках кластера осуществляется репликация данных о вычислительной среде (ее конфигурации). Один из TIVM-серверов выступает в качестве мастера (ведущего), прочие – слейвов (ведомых). Все изменения в данных проводятся через мастера, фиксируются в журнале и передаются слейвам. Согласованность реплицируемых данных обеспечивается реализацией алгоритма консенсуса Raft в сети распределенных хранилищ. Выход из строя или временная недоступность какого-либо из TIVM-серверов не приводит к нарушению согласованности данных. В случае недоступности мастер-узла его роль возьмет на себя какой-либо из слейв-серверов.

Зоны формируют интегральную среду управления виртуализацией. Они используют общие данные о конфигурации, пользователях, их группах и правах доступа. Web-интерфейсы зон поддерживают общую авторизацию с простым переключением между зонами (с учетом прав доступа пользователя).

Виртуальная вычислительная инфраструктура, поддерживаемая ПВК "ТИВМ", может быть создана на базе одного хост-сервера, который обеспечивает работу TIVM, СУБД PostgreSQL и одновременно используется для запуска VM. Данный вариант развертывания виртуальной среды удобно применять для проверки функциональных возможностей системы.

1.7 Доступ к серверу управления виртуализацией

В ПВК "ТИВМ" реализованы 2 инструмента взаимодействия пользователей с сервером управления виртуализацией TIVM: CLI и web-интерфейс.

Для использования CLI необходимо авторизоваться на сервере, на котором установлен TIVM. Вводимые команды CLI выполняются от имени авторизованного пользователя. При первом использовании CLI следует авторизоваться под служебной учетной записью *tivadmin*, создаваемой при установке ПО и имеющей права суперадминистратора.

Web-интерфейс TIVM работает в среде тонкого web-клиента. Для доступа к нему достаточно типового web-браузера. При работе на сервере, где установлен TIVM, для входа в web-интерфейс в адресной строке браузера следует ввести `http://localhost:10180`. Будет открыта страница авторизации, на которой нужно указать логин *tivadmin* и пароль. Для входа в web-интерфейс с другого компьютера, вместо `localhost` в адресной строке указывается адрес TIVM-сервера (или отображаемое в него доменное имя).

Пароль для учетной записи *tivadmin* по умолчанию – *tivadmin*. Системный администратор может поменять этот пароль, а также создать учетные записи других пользователей (через CLI или web-интерфейсе). Соответствующие операции рассмотрены в разд. 3.

2 Управление хостами виртуализации и кластерами

2.1 Управление хостами виртуализации

Хост виртуализации – это физический сервер, управляемый TIVM, на котором могут запускаться ВМ. ПВК "ТИВМ" обеспечивает управление хост-серверами, использующими разные дистрибутивы Linux и технологии виртуализации.

Группа команд управления хостами виртуализации в CLI – **tivhost**:

```
tivhost command [args] [options]
```

2.1.1 Регистрация хоста виртуализации

Чтобы использовать физический сервер в виртуальной вычислительной среде, его надо зарегистрировать в TIVM, как хост виртуализации. Для регистрации достаточно указать имя и тип хоста.

Для *регистрации хоста* служит команда **tivhost create**:

```
tivhost create hostname [options]
```

Аргумент *hostname* – уникальное имя хоста.

Допустимые параметры (секция *[options]*):

-c, --cluster *id* | *name* – идентификатор или имя кластера, в который включается хост;

-i, --im – использовать драйвер, управляющий получением данных о хосте (Information Manager). За параметром через пробел указывается имя драйвера;

-t, --type – зарегистрировать хост, предоставляемый внешним облачным провайдером. За параметром через пробел указывается имя провайдера;

-v, --vm – использовать драйвер управления ВМ (Virtual Machine Manager). За параметром через пробел указывается имя драйвера.

Для регистрации KVM- или LXD-хоста необходимо, чтобы он был доступен по SSH без ручного ввода пароля.

Пример:

```
tivhost create host1 -i kvm -v kvm
```

Данная команда регистрирует хост виртуализации с именем *host1*, использующий гипервизор KVM.

Команда `tivhost create` возвращает идентификатор зарегистрированного хоста виртуализации. Дальнейшие операции с зарегистрированным в TIVM хостом могут выполняться по его имени или идентификатору.

После регистрации хост становится доступен для мониторинга, а его ресурсы могут использовать планировщиком TIVM.

2.1.2 Отмена регистрации хоста виртуализации

Для отмены регистрации хоста виртуализации служит команда **tivhost delete**:

```
tivhost delete rangeHostId | hostIDlist
```

Аргумент команды: *rangeHostId* – диапазон идентификаторов хостов виртуализации в форме *n..m,k* – либо *hostIDlist* – список идентификаторов или имен хостов, перечисляемых через запятую. Список может состоять из одного элемента.

Примеры:

```
tivhost delete host1  
tivhost delete 3,5  
tivhost delete 5..7,12
```

Первая команда отменяет регистрацию хоста виртуализации с именем *host1*, вторая делает то же самое по отношению к хостам с идентификаторами 3 и 5, третья отменяет регистрацию хостов с идентификаторами 5, 6, 7 и 12.

2.1.3 Получение информации о хосте виртуализации

Для получения информации о хосте виртуализации служит команда **tivhost show**:

```
tivhost show hostId | hostName [options]
```

Аргумент команды:

- *hostId* | *hostName* – идентификатор или имя хоста виртуализации.

Допустимые параметры (секция *[options]*):

--decrypt – получить дешифрованные атрибуты;

-x, --xml – представить информацию в XML-формате.

Состав возвращаемой информации о хосте виртуализации:

- общие атрибуты хоста (раздел "Information" – имя, идентификатор, используемые драйверы, время получения последних данных мониторинга);

- данные об основных вычислительных ресурсах (раздел "Host Shares") – памяти, процессорах, а также общем числе запущенных на хосте ВМ;

- данные о локальном хранилище данных (раздел "Local System Datastore"), если оно используется хостом;

- основные характеристики хоста, получаемые модулем мониторинга (раздел "Monitoring Information");

- список ВМ, запущенных на хосте до его регистрации в TIVM, т.е. не управляемых TIVM (раздел "Wild Virtual Machines");

- список ВМ, запущенных на хосте и управляемых TIVM (раздел "Virtual Machines").

Пример:

```
tivhost show host1
```

Результат:

```
HOST 3 INFORMATION
ID                : 3
NAME              : host1
CLUSTER           : cluster1
STATE             : MONITORED
IM_MAD            : kvm
VM_MAD            : kvm
LAST MONITORING TIME : 12/16 10:22:55

HOST SHARES
TOTAL MEM         : 8.1G
USED MEM (REAL)  : 4.5G
USED MEM (ALLOCATED) : 1024M
TOTAL CPU        : 200
USED CPU (REAL)  : 64
USED CPU (ALLOCATED) : 100
TOTAL VMS        : 1

LOCAL SYSTEM DATASTORE #0 CAPACITY
TOTAL:           : 780.1G
USED:            : 255.4G
FREE:           : 524.7G

MONITORING INFORMATION
ARCH="x86_64"
CPUSPEED="1599"
HOSTNAME="host1"
HYPERVISOR="kvm"
IM_MAD="kvm"
MODELNAME="Intel(R) Core(TM) i7-4650U CPU @ 1.70GHz"
NETRX="0"
NETTX="0"
RESERVED_CPU=""
RESERVED_MEM=""
VERSION="5.00.0"
VM_MAD="kvm"

WILD VIRTUAL MACHINES
NAME                                IMPORT_ID  CPU    MEMORY

VIRTUAL MACHINES
ID  USER      GROUP    NAME      STAT  UCPU  UMEM  HOST    TIME
9   tivadmin  tivadmin kvm1-9    runn  0.0   1024M server  2d 16h29
```

Основные данные хоста виртуализации, возвращаемые модулем мониторинга, приведены в табл. 2.1.

Таблица 2.1 – Состав основных данных мониторинга хоста виртуализации, возвращаемых командой `tivhost show`

Параметр	Описание
HYPERVISOR	Имя гипервизора, установленного на хосте
ARCH	Архитектура процессоров, например, "x86_64"
MODELNAME	Модель процессоров, например, " Intel(R) Core(TM) i7-2620M CPU @ 2.70GHz"
CPUSPEED	Частота процессоров в МГц
HOSTNAME	Имя хоста (зарегистрированное командой <code>tivhost create</code>)
VERSION	Версия драйвера мониторинга и ПАМ
MAX_CPU	Число ядер процессоров, умноженное на 100. Например, если сервер оснащен 16 ядрами, то <code>MAX_CPU = 1600</code> .
RESERVED_CPU	Число зарезервированных ядер процессоров, отводимых для функционирования гипервизора, умноженное на 100. Значение <code>MAX_CPU – RESERVED_CPU</code> возвращается командой <code>tivhost show</code> в параметре "Total CPU" в разделе "Host Share"
TOTAL_CPU	Процессорные ресурсы, выделяемые для функционирования ВМ. Значение равно <code>MAX_CPU – RESERVED_CPU</code>
MAX_MEM	Максимальный объем памяти, который может использоваться ВМ, развернутыми на хосте. При планировании распределения вычислительных ресурсов рекомендуется исключать из максимального объема память, отводимую для гипервизора (параметр <code>RESERVED_MEM</code>)
RESERVED_MEM	Объем памяти, отводимый для функционирования гипервизора. Значение <code>MAX_MEM – RESERVED_MEM</code> возвращается командой <code>tivhost show</code> в параметре "Total mem" в разделе "Host Share"
TOTAL_MEM	Объем памяти, выделяемый для функционирования ВМ. Значение равно <code>MAX_MEM – RESERVED_MEM</code>
MAX_DISK	Общий объем хранилища данных в Мб, размещенного на хосте виртуализации
USED_CPU	Процент используемых ресурсов процессоров (относительно <code>TOTAL_CPU</code>), умноженный на число ядер. Значение выводится в разделе "Host Share" в параметре "Used CPU (real)". Например, если хост оснащен одним процессором с 4 ядрами, и процессор загружен на 50%, то <code>USED_CPU = 200</code>
USED_MEM	Объем используемой памяти в Кб. Значение выводится в разделе "Host Share" в параметре "Used mem (real)"
USED_DISK	Объем занятого пространства хранилища данных в Мб
FREE_CPU	Процент простаивающих (не используемых) ресурсов процессоров (относительно <code>TOTAL_CPU</code>), умноженный на число ядер
FREE_MEM	Свободная память, доступная для ВМ, в Кб
FREE_DISK	Объем свободного пространства хранилища данных в Мб

Параметр	Описание
CPU_USAGE	Общее число ядер процессоров, выделенных запущенным на хосте ВМ, умноженное на 100. Значение выводится в разделе "Host Share" в параметре "Used CPU (allocated)"
MEM_USAGE	Общий объем памяти, выделенной запущенным на хосте ВМ, в Кб. Значение выводится в разделе "Host Share" в параметре "Used mem (allocated)"
DISK_USAGE	Общий объем пространства в хранилища данных в Мб, выделенного запущенным на хосте ВМ для образов их дисков
NETRX	Объем данных в байтах, полученных хостом по сети
NETTX	Объем данных в байтах, переданным хостом по сети
RUNNING_VMS	Число ВМ, запущенных на хосте
WILD	Список имен ВМ (разделенных запятыми), которые были запущены на хосте виртуализации до его регистрации в TIVM и не управляемых им
ZOMBIES	Список имен ВМ (разделенных запятыми), которые были запущены на хосте виртуализации с помощью TIVM, но в данный момент не управляются им

Модуль мониторинга обновляет данные, собираемые ПАМ, с периодичностью, установленной в его настройках. Для *внеочередного обновления данных мониторинга* служит команда **tivhost forceupdate**:

```
tivhost forceupdate rangeHostId | hostIDlist [option]
```

Аргумент этой команды: *rangeHostId* – диапазон идентификаторов хостов виртуализации в форме *n..m,k* – либо *hostIDlist* – список идентификаторов или имен хостов, перечисляемых через запятую. Список может состоять из одного элемента.

Единственный необязательный параметр команды:

-c, --cluster id | name – идентификатор или имя кластера, включающие хосты, для которых запрашивается обновление данных мониторинга.

Примеры:

```
tivhost forceupdate host1,host6
tivhost forceupdate 3..5,8
```

Последовательность значений данного показателя мониторинга, зафиксированных на хосте виртуализации за определенный интервал времени, возвращает команда tivhost monitoring:

```
tivhost monitoring hostId | hostName attr options
```

Аргументы команды:

- hostId | hostName – идентификатор или имя хоста виртуализации;

- *attr* – имя показателя мониторинга, значения которого запрашиваются.

Допустимые параметры (секция [*options*]):

--start *date* – дата и время начала интервала;
--end *date* – дата и время конца интервала;
--n_elements *n* – число запрашиваемых значений;
--csv – выводить информацию в CSV-формате;
--table – выводить информацию табличной форме;
--unit *name* – имя единицы измерения показателя.

2.1.4 Получение списка хостов виртуализации

Для получения списка хостов виртуализации служит команда **tivhost list**:

```
tivhost list [options]
```

Допустимые параметры (секция [*options*]):

--adjust *col1, col2, ...* – выравнять размеры возвращаемой таблицы, чтобы содержимое столбцов с указанными номерами (*col1, col2, ...*) не обрезалось;

--csv – представить информацию в CSV-формате;

--csv-del *delimiter* – в CSV-формате использовать в качестве разделителя символ *delimiter*;

--describe – выводить описания столбцов;

--expand [*name1=prop1, name2=prop2, ...*] – расширять возвращаемую таблицу, чтобы она занимала всю ширину терминала. Параметры *name1* и *name2* – имена столбцов, *prop1* и *prop2* – их относительная ширина в долях от единицы (например, 0.4 и 0.6). Если относительная ширина столбцов не указана, то таблица расширяется пропорционально их содержимому. Если столбцы не указаны, то расширяются все столбцы таблицы;

--extended – выводить дополнительную информацию (при использовании XML-формата);

-f, --filter *cond1, cond2, ...* – фильтровать выводимую информацию по условиям *cond1, cond2, ...*. Каждое условие задает тройка <имя столбца, предикат, значение>. Набор предикатов: =, != (не равно), <, >, <=, >=, ~ (начинается с). Условия соединяются оператором, указанным в параметре *operator*;

-k, --kilobytes – выводить значения в килобайтах;

-l, --list *col1, col2, ...* – выводить только столбцы с указанными номерами (*col1, col2, ...*);

--listconf *conf* – выводить только столбцы, заданные в заранее определенном списке *conf*;
 --no-expand – запретить расширение столбцов таблицы;
 --no-header – не выводить заголовков таблицы;
 --no-pager – не разбивать таблицу на страницы;
 -n, --numeric – не преобразовывать идентификаторы пользователя и группы (выводить их в цифровом виде);
 --operator *oper* – логический оператор, связывающий условия фильтра, заданного параметром *filter*. Допустимые значения *oper* – AND, OR. Значение по умолчанию – AND;
 -s, --size *name1=size1, name1=size1, ...* – установить ширину столбцов (в символах) с указанными именами;
 -x, --xml – представить информацию в XML-формате.

Пример:

```
tivhost list
```

Результат:

ID	NAME	CLUSTER	RVM	ALLOCATED_CPU	ALLOCATED_MEM	STAT
0	host5	cluster1	1	100 / 400 (25%)	1024M / 7.3G (13%)	on
1	host1	cluster1	0	-	-	off
2	host2	cluster1	0	-	-	off

В столбце "RVM" выводится число запущенных ВМ, в столбце "STAT" – обозначение режима работы хоста.

Вывод списка хостов виртуализации в режиме периодического обновления обеспечивает команда **tivhost top**. Она совпадает с **tivhost list** за исключением того, что список обновляется через установленный параметром **--delay** интервал времени. Параметры команды те же, что у **tivhost list** кроме **--describe**. Также есть специфичный параметр:

-d, --delay *x* – период обновления данных мониторинга – *x* секунд.

2.1.5 Управление ЖЦ хоста виртуализации

С точки зрения TIVM хост виртуализации может находиться в одном из 6 режимов работы, приведенных в табл. 2.2.

Для перевода хоста виртуализации в режимы **disabled**, **offline** и **enabled** предназначены команды **tivhost disable**, **tivhost offline** и **tivhost enable** соответственно:

```
tivhost disable rangeHostId | hostIDlist
tivhost offline rangeHostId | hostIDlist
tivhost enable rangeHostId | hostIDlist
```

Аргумент этих команд: *rangeHostId* – диапазон идентификаторов хостов виртуализации в форме *n..m,k* – либо *hostIDlist* – список

идентификаторов или имен хостов, перечисляемых через запятую. Список может состоять из одного элемента.

Пример:

```
tivhost disable 3
```

Таблица 2.2 – Режимы работы хоста виртуализации с точки зрения TIVM

Режим работы	Обозначение режима	Характеристика режима работы	Доступность хоста для мониторинга	Возможности управления хостом	
				командами администратора	планировщиком TIVM
enabled	on	Хост в эксплуатации и полностью доступен TIVM для развертывания VM	+	+	+
update	update	Характеристики хоста фиксируются модулем мониторинга	+	+	+
disabled	dsbl	Хост недоступен для планировщика TIVM, например, на период проведения его технического обслуживания. Возможности управление хостом в ручном режиме сохраняются	+	+	–
offline	off	Хост в автономном режиме. Он полностью недоступен для TIVM	–	–	–
error	err	Ошибка в процессе мониторинга хоста. Для ее детализации следует использовать команду <code>tivhost show</code>	+	+	–
retry	retry	Мониторинг хоста в состоянии ошибки	+	+	–

2.1.6 Очистка хоста виртуализации от VM, запущенных на нем

Для *переноса всех VM с хоста на другой хост (миграции VM)* служит команда `tivhost flush`:

```
tivhost flush rangeHostId | hostIDlist
```

Аргумент этой команды: `rangeHostId` – диапазон идентификаторов хостов виртуализации в форме `n..m,k` – либо `hostIDlist` – список идентификаторов или имен хостов, перечисляемых через запятую. Список может состоять из одного элемента.

Выполнение команды `tivhost flush` включает:

- перенос всех ВМ, запущенных на хостах, идентификаторы которых принадлежат диапазону `rangeHostId` или списку `hostIDlist`, на другие хосты виртуализации, располагающие достаточными для этих ВМ вычислительными ресурсами;

- перевод хостов, идентификаторы которых принадлежат диапазону `rangeHostId` или списку `hostIDlist`, в режим `disabled`.

Данный сценарий может быть изменен на альтернативный, при котором все ВМ хоста перезапускаются. Сценарий для команды `tivhost flush` задается в конфигурационном файле `/etc/tiv/cli/tivhost.yaml` через параметр `default_actions/flush`, для которого указывается значение `resched` (перенести ВМ на другие хосты) или `delete-recreate` (перезапустить ВМ), например:

```
:default_actions:  
- :flush: delete-recreate
```

2.1.7 Определение дополнительных признаков хоста виртуализации

TIVM получает параметры хоста виртуализации от модуля мониторинга. Если необходимо приписать хосту дополнительные атрибуты (признаки), то это можно сделать как с помощью средств мониторинга, настроив возврат этих атрибутов через ПАМ, так и установив нужные атрибуты с помощью команды `tivhost update` с параметром `--append`:

```
tivhost update hostId | hostName [file] --append property
```

Команда `tivhost update` предназначена для *обновления шаблона хоста виртуализации*. Ее аргументы:

- `hostId | hostName` – идентификатор или имя хоста виртуализации;

- `file` – имя файла шаблона хоста.

Шаблон формируется в виде текстового файла. Его строка определяет атрибут в виде пары: `name = value`, например:

```
CACHE = 1024  
LOAD = "high"
```

Если имя шаблона не указано, то TIVM запустит редактор для изменения текущего содержимого шаблона.

Параметр `--append (-a)` позволяет задать признак `property`, добавляемый в шаблон. Форма его определения: `name = value`. Например, можно присвоить хосту атрибут "TYPE" со значением "production":

```
tivhost update 3 --append TYPE="production"
```

В дальнейшем можно использовать данное значение, как условие для отбора хостов для развертывания ВМ. Например, ВМ для рабочих приложений развертывать на хостах типа "production", а тестовые приложения – на хостах типа "test".

2.1.8 Обновление драйверов хоста виртуализации

Драйвера и параметры ПАМ, используемые TIVM для мониторинга хоста виртуализации, копируются с него в папку /var/tmp/tiv сервера управления виртуализацией. Если файлы драйверов и компонентов ПАМ в этой папке заменены на новые версии, то они должны быть также скопированы на управляемый хост. Это действие выполняет команда **tivhost sync**:

```
tivhost sync [rangeHostId | hostIDlist] [options]
```

Необязательный аргумент команды: *rangeHostId* – диапазон идентификаторов хостов виртуализации в форме *n..m,k* – либо *hostIDlist* – список идентификаторов или имен хостов, перечисляемых через запятую. Список может состоять из одного элемента. Если аргумент не указан, то команда относится ко всем хостам виртуализации.

Обновление драйверов и компонентов ПАМ на хосте виртуализации осуществляется, если их версия меньше версии, указанной в файле /var/lib/tiv/remotes/VERSION. Версию драйверов и ПАМ, установленных на хосте виртуализации, возвращает команда **tivhost show** (в секции "Monitoring Information", см. п. 2.1.3). Например, если **tivhost show** вернула версию "5.0.11", а в /var/lib/tiv/remotes/VERSION указана версия "5.0.13", то обновление будет проведено.

Допустимые параметры **tivhost sync** (секция [options]):

-c, --cluster id | name – идентификатор или имя кластера, включающие хосты, которые подлежат обновлению;
--force – выполнить обновление без проверки версий;
--rsync – удалять с хоста файлы ПАМ, которых нет в папке /var/tmp/tiv сервера управления виртуализацией.

Пример:

```
tivhost sync 3 --rsync
```

2.1.9 Импорт ВМ, запущенных на хосте виртуализации до его регистрации в TIVM

Модуль мониторинга позволяет получать информацию о всех ВМ, функционирующих на хосте виртуализации: как управляемых TIVM, так и запущенных другими средствами (и не управляемых TIVM). *Список ВМ, не*

управляемых TIVM, содержится в разделе "Wild Virtual Machines" результата, возвращаемого командой `tivhost show` (см. п. 2.1.3).

Импорт ВМ создает условия для их управления средствами TIVM. Для импорта служит команда **tivhost importvm**:

```
tivhost importvm hostId | hostName name
```

Аргументы команды:

- *hostId* | *hostName* – идентификатор или имя хоста виртуализации;

- *name* – имя импортируемой ВМ.

Пример:

```
tivhost importvm 3 vm24
```

Возможности управления импортированными ВМ несколько ограничены по сравнению с управлением ВМ, запущенными средствами TIVM. В частности, по отношению к импортированным ВМ не могут выполняться команды `tivvm poweroff`, `tivvm undeploy`, `tivvm migrate` и `tivvm recover` с параметром `--recreate` (см. разд. 5).

2.1.10 Переименование хоста виртуализации

Для *изменения имени* хоста виртуализации служит команда **tivhost rename**:

```
tivhost rename hostId | hostName name
```

Аргументы команды:

- *hostId* | *hostName* – идентификатор или имя хоста виртуализации;

- *name* – новое имя хоста.

Пример:

```
tivhost rename 3 host3
```

2.2 Управление кластерами хостов виртуализации

Кластер – это группа хостов виртуализации (физических серверов), определяемая для упрощения управления ими. С кластерами могут связываться хранилища данных и виртуальные сети.

Группа команд управления кластерами хостов виртуализации в CLI – **tivcluster**:

```
tivcluster command [args] [options]
```

2.2.1 Регистрация кластера

Для регистрации кластера хостов виртуализации служит команда **tivcluster create**:

```
tivcluster create clusterName
```

Аргумент *clusterName* – уникальное имя кластера.

Пример:

```
tivcluster create cluster1
```

Команда `tivcluster create` возвращает идентификатор зарегистрированного кластера хостов виртуализации. Дальнейшие операции с зарегистрированным в TIVM кластером могут выполняться по его имени или идентификатору.

Кластер после его создания является пустым, т.е. не имеет хостов. Они могут быть включены в кластер командой `tivcluster addhost` (см. п. 2.2.4) или быть привязаны к нему при их регистрации (п. 2.1.1, см. параметр `--cluster`).

2.2.2 Переименование кластера

Имя зарегистрированного кластера может быть *изменено* с помощью команды **tivcluster rename**:

```
tivcluster rename clusterIdOrName newName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *newName* – новое имя кластера.

Пример:

```
tivcluster rename 1 cluster1prod
```

2.2.3 Отмена регистрация кластера

Для отмены регистрации кластера служит команда **tivcluster delete**:

```
tivcluster delete rangeClusterId | ClusterIDlist
```

Аргумент команды: *rangeClusterId* – диапазон идентификаторов кластеров в форме *n..m,k* – либо *ClusterIDlist* – список идентификаторов или имен кластеров, перечисляемых через запятую. Список может состоять из одного элемента.

Примеры:

```
tivcluster delete cluster1prod  
tivcluster delete 1,4
```

```
tivcluster delete 1..3,5
```

Первая команда отменяет регистрацию кластера с именем `cluster1prod`, вторая делает то же самое по отношению к кластерам с идентификаторами 1 и 4, третья отменяет регистрацию кластеров с идентификаторами 1, 2, 3 и 5.

Отмена регистрации кластера не отменяет регистрацию хостов, включенных в них командой `tivcluster addhost` (см. п. 2.2.4). Отмену регистрации хостов реализует команда `tivhost delete` (п. 2.1.2).

2.2.4 Включение хоста виртуализации в кластер

Включение в кластер хоста виртуализации обеспечивает команда **`tivcluster addhost`**:

```
tivcluster addhost clusterIdOrName hostIdOrName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *hostIdOrName* – идентификатор или имя хоста виртуализации.

Примеры:

```
tivcluster addhost 1 3  
tivcluster addhost 1 host3  
tivcluster addhost cluster1prod 3  
tivcluster addhost cluster1prod host3
```

Хост виртуализации может входить только в один кластер.

2.2.5 Удаление хоста виртуализации из кластера

Удаление хоста виртуализации из кластера выполняет команда **`tivcluster delhost`**:

```
tivcluster delhost clusterIdOrName hostIdOrName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *hostIdOrName* – идентификатор или имя хоста виртуализации.

Примеры:

```
tivcluster delhost 1 3  
tivcluster delhost 1 host3  
tivcluster delhost cluster1prod 3  
tivcluster delhost cluster1prod host3
```

Удаление из кластера хоста виртуализации не отменяет регистрацию последнего. Отмену регистрации хостов реализует команда `tivhost delete` (п. 2.1.2).

2.2.6 Получение информации о кластере

Для получения информации о кластере служит команда **tivcluster show**:

```
tivcluster show clusterIdOrName [options]
```

Аргумент команды:

- *clusterIdOrName* – идентификатор или имя кластера.

Допустимые параметры (секция [options]):

--decrypt – получить дешифрованные атрибуты;

-x, --xml – представить информацию в XML-формате.

Состав возвращаемой информации о кластере:

- имя и идентификатор кластера;

- атрибуты из шаблона кластера (раздел "Cluster Template");

- список хостов виртуализации, входящих в кластер;

- списки виртуальных сетей и хранилищ данных, связанных с кластером.

Пример:

```
tivhost show cluster1prod
```

Результат:

```
CLUSTER 1 INFORMATION
ID           : 1
NAME        : cluster1prod

CLUSTER TEMPLATE

HOSTS
1
3

VNETS
1

DATASTORES
4
```

2.2.7 Получение списка кластеров

Для получения списка кластеров служит команда **tivcluster list**:

```
tivcluster list [options]
```

Допустимые параметры (секция [options]):

--adjust *col1, col2, ...* – выравнять размеры возвращаемой таблицы, чтобы содержимое столбцов с указанными номерами (*col1, col2, ...*) не обрезалось;

--csv – представить информацию в CSV-формате;

--csv-del *delimiter* – в CSV-формате использовать в качестве разделителя символ *delimiter*;
 --describe – выводить описания столбцов;
 --expand [*name1=prop1, name2=prop2, ...*] – расширять возвращаемую таблицу, чтобы она занимала всю ширину терминала. Параметры *name1* и *name2* – имена столбцов, *prop1* и *prop2* – их относительная ширина в долях от единицы (например, 0.4 и 0.6). Если относительная ширина столбцов не указана, то таблица расширяется пропорционально их содержимому. Если столбцы не указаны, то расширяются все столбцы таблицы;
 -f, --filter *cond1, cond2, ...* – фильтровать выводимую информацию по условиям *cond1, cond2, ...*. Каждое условие задает тройка <имя столбца, предикат, значение>. Набор предикатов: =, != (не равно), <, >, <=, >=, ~ (начинается с). Условия соединяются оператором, указанным в параметре *operator*;
 -l, --list *col1, col2, ...* – выводить только столбцы с указанными номерами (*col1, col2, ...*);
 -c, --listconf *conf* – выводить только столбцы, заданные в заранее определенном списке *conf*;
 --no-expand – запретить расширение столбцов таблицы;
 --no-header – не выводить заголовков таблицы;
 --no-pager – не разбивать таблицу на страницы;
 -n, --numeric – не преобразовывать идентификаторы пользователя и группы (выводить их в цифровом виде);
 --operator *oper* – логический оператор, связывающий условия фильтра, заданного параметром *filter*. Допустимые значения *oper* – AND, OR. Значение по умолчанию – AND;
 -s, --size *name1=size1, name2=size2, ...* – установить ширину столбцов (в символах) с указанными именами;
 -x, --xml – представить информацию в XML-формате.

Пример:

```
tivcluster list
```

Результат:

ID	NAME	HOSTS	NETS	DATASTORES
100	cluster1prod	2	1	1

В списке для каждого кластера приводятся число хостов, входящих в него, а также количество виртуальных сетей и хранилищ данных, связанных с ним.

2.2.8 Управление связями кластера с виртуальными сетями

С кластером может быть связано множество виртуальных сетей. В свою очередь, виртуальная сеть может быть связана со множеством кластеров. Связь кластера и виртуальной сети означает, что настройки всех хостов кластера обеспечивают возможности использования запущенными на них VM ресурсов данной виртуальной сети (например, аренды IP-адресов, доступа к ее DNS-серверу и проч.).

Для *установки связи кластера и виртуальной сети* служит команда **tivcluster addvnet:**

```
tivcluster addvnet clusterIdOrName vnetIdOrName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *vnetIdOrName* – идентификатор или имя виртуальной сети.

Примеры:

```
tivcluster addvnet 1 12  
tivcluster addvnet cluster1prod vn4
```

Удаление связи кластера и виртуальной сети обеспечивает команда **tivcluster delvnet:**

```
tivcluster delvnet clusterIdOrName vnetIdOrName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *vnetIdOrName* – идентификатор или имя виртуальной сети.

Примеры:

```
tivcluster delvnet 1 vn4  
tivcluster addvnet cluster1prod 4
```

Удаление связи кластера и виртуальной сети не отменяет их регистрации в TIVM.

2.2.9 Управление связями кластера с хранилищами данных

С кластером может быть связано множество хранилищ данных. В свою очередь, хранилище может быть связано со множеством кластеров. Связь кластера и хранилища данных означает, что настройки всех хостов кластера обеспечивают возможности использования запущенными на них VM ресурсов этого хранилища. Другими словами, все хосты кластера должны иметь доступ к хранилищам данных этого кластера.

Для *установки связи кластера и хранилища данных* служит команда **tivcluster adddatastore:**

```
tivcluster adddatastore clusterIdOrName dsIdOrName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *dsIdOrName* – идентификатор или имя хранилища данных.

Примеры:

```
tivcluster adddatastore 1 12
tivcluster adddatastore cluster1prod isci12
```

Удаление связи кластера и хранилища данных обеспечивает команда **tivcluster deldatastore:**

```
tivcluster deldatastore clusterIdOrName dsIdOrName
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *dsIdOrName* – идентификатор или имя хранилища данных.

Примеры:

```
tivcluster deldatastore 1 isci12
tivcluster deldatastore cluster1prod 12
```

Удаление связи кластера и хранилища данных не отменяет их регистрации в TIVM.

Для того, чтобы на хостах кластера планировщик TIVM мог запускать ВМ, необходимо, чтобы к кластеру было привязано хотя бы одно системное хранилище данных. Для повышения производительности ВМ могут использоваться несколько системных хранилищ разных типов.

2.2.10 Изменение шаблона кластера

Кластер имеет шаблон, в котором могут быть представлены его атрибуты, используемые планировщиком TIVM при запуске ВМ. Атрибуты, определенные на уровне ядра системы, приведены в табл. 2.3.

Таблица 2.3 – Общие атрибуты кластера, представляемые в его шаблоне

Параметр	Описание
RESERVED_CPU	Выражается в процентах и применяется ко всем хостам кластера. Значение вычитается из параметра "Total CPU" (п. 2.1.3, табл. 2.1)
RESERVED_MEM	Выражается в Кб и применяется ко всем хостам кластера. Значение вычитается из параметра "Total mem" (п. 2.1.3, табл. 2.1).

Атрибуты, содержащиеся в шаблоне кластера, могут быть изменены с помощью команды **tivcluster update**. Также эта команда позволяет включить в шаблон другие атрибуты (параметр `--append`). Формат команды:

```
tivcluster update clusterIdOrName [file] [options]
```

Аргументы команды:

- *clusterIdOrName* – идентификатор или имя кластера;
- *file* – имя файла шаблона хоста.

Команда загружает атрибуты из текстового файла *file*. Его строка определяет атрибут в виде пары: `name = value`, например:

```
RESERVED_CPU = 5
LEVEL = "A0"
```

Если имя файла шаблона не указано, то TIVM запустит редактор для изменения текущего содержимого шаблона.

Параметр `--append (-a)` позволяет задать признак *property*, добавляемый в шаблон. Форма его определения: `name = value`. Например, можно присвоить кластеру атрибут "MODE" со значением "normal":

```
tivcluster update 1 --append MODE="normal"
```

В дальнейшем можно использовать данное значение, как условие для планировщика при отборе хостов для развертывания VM.

2.3 Настройка планировщика

Планировщик вычислительных ресурсов – модуль TIVM, функционирующий параллельно с базовым модулем управления виртуализацией (как отдельный процесс) и отвечающий за распределение VM по хостам, выделение им пространств в хранилищах данных, сетевых интерфейсов и прочих вычислительных ресурсов. Планировщик реализует *политику распределения ресурсов, основанную на рангах* и направленную на определение приоритетов ресурсов с точки зрения потребностей запрашивающей их VM.

2.3.1 Процедура подбора вычислительных ресурсов для VM

Выполняемая планировщиком *процедура подбора вычислительных ресурсов для VM* состоит из следующих 9 шагов.

Шаг 1. Если объем какой-либо из дисков VM превышает доступный объем в зарегистрированных хранилищах образов, то VM не может быть запущена. Конец процедуры.

Шаг 2. Отбор хостов виртуализации, подходящих для размещения VM. Из списка хостов исключаются те, что не обладают достаточными свободными ресурсами (процессорными ядрами и памятью), а также те, что не удовлетворяют условию `SCHED_REQUIREMENTS`, заданному в шаблоне VM. Если список подходящих хостов пуст, то VM не может быть запущена (конец процедуры).

Шаг 3. Отбор системных хранилищ данных, подходящих для VM. Из списка хранилищ исключаются те, что не обладают достаточным свободным объемом, а также те, что не удовлетворяют условию

SCHED_DS_REQUIREMENTS, заданному в шаблоне VM. Если список подходящих системных хранилищ данных пуст, то VM не может быть запущена (конец процедуры).

Шаг 4. Если в шаблоне VM предусмотрен автоматический выбор виртуальной сети, то отбор виртуальных сетей, подходящих для *сетевых интерфейсов* VM (NIC). Из списка виртуальных сетей исключаются те, в которых нет свободных адресов, а также те, что не удовлетворяют условию SCHED_REQUIREMENTS, заданному в шаблоне VM. Проверка выполняется для каждого NIC, указанного в шаблоне VM. Если список подходящих виртуальных сетей пуст хотя бы для одного NIC VM, то VM не может быть запущена (конец процедуры).

Шаг 5. Выбор хоста на основе политики, установленной в шаблоне VM, или *политики по умолчанию* для всех VM, заданной в *файле конфигурации планировщика* /etc/tiv/sched.conf (п. 2.3.4).

Параметр PLACEMENT/SCHED_RANK шаблона VM определяет пользовательский критерий выбора хоста (способ вычисления его ранга). Файл конфигурации планировщика позволяет применять как типовые критерии, так и пользовательский критерий. Параметр PLACEMENT/SCHED_RANK в шаблоне VM имеет приоритет над политикой по умолчанию.

Пользовательский критерий описывается по правилам, приведенным в п. 2.3.2. В файле конфигурации планировщика он указывается в параметре DEFAULT_SCHED/RANK и применяется, если параметр DEFAULT_SCHED/POLICY равен 3.

Ранг вычисляется на основе параметров хоста виртуализации. В их число могут входить как типовые параметры, заложенные в ядро TIVM (табл. 2.1), так и те, что добавлены в шаблон хоста (п. 2.1.7) или кластера (п. 2.2.10) пользователем, а также те, что включены в состав параметров, возвращаемых модулем мониторинга. Хост, имеющий *максимальный ранг*, считается наиболее подходящим для размещения VM.

Если в шаблоне VM нет параметра PLACEMENT/SCHED_RANK, и в файле конфигурации планировщика параметр DEFAULT_SCHED/POLICY равен 0, 1, 2 или 4, то хост выбирается с помощью одного из *типовых критериев* (п. 2.3.3).

Шаг 6. Из списка хранилищ данных, сформированного на шаге 3, выбираются те, что связаны с кластером, к которому относится хост, выбранный на шаге 5. Если список подходящих хранилищ пуст, то VM не может быть запущена (конец процедуры).

Шаг 7. Выбор системного хранилища данных на основе политики, установленной в шаблоне VM, или *политики по умолчанию* для всех VM, заданной в файле конфигурации планировщика. Хранилище выбирается из списка, сформированного на предыдущем шаге.

Параметр `PLACEMENT/SCHED_DS_RANK` шаблона ВМ определяет пользовательский критерий выбора хранилища (способ вычисления его ранга). Файл конфигурации планировщика позволяет применять как типовые критерии, так и пользовательский критерий. Параметр `PLACEMENT/SCHED_DS_RANK` в шаблоне ВМ имеет приоритет над политикой по умолчанию.

Пользовательский критерий описывается по правилам, приведенным в п. 2.3.2. В файле конфигурации планировщика он указывается в параметре `DEFAULT_DS_SCHED/RANK` и применяется, если параметр `DEFAULT_DS_SCHED/POLICY` равен 2.

Ранг вычисляется на основе параметров хранилища данных. В их число могут входить как типовые параметры, заложенные в ядро TIVM, так и те, что добавлены в шаблон хранилища пользователем, а также те, что включены в состав параметров, возвращаемых модулем мониторинга. Хранилище, имеющее *максимальный ранг*, считается наиболее подходящим для размещения ВМ.

Если в шаблоне ВМ нет параметра `PLACEMENT/SCHED_DS_RANK`, и в файле конфигурации планировщика параметр `DEFAULT_DS_SCHED/POLICY` равен 0, 1 или 3, то хранилище выбирается с помощью одного из типовых критериев (п. 2.3.3).

Шаг 8. Из сформированных на шаге 4 списков виртуальных сетей для каждого сетевого интерфейса ВМ выбираются сети, связанные с кластером, к которому относится хост, выбранный на шаге 5. Если хотя бы один из списков пуст, то ВМ не может быть запущена (конец процедуры).

Если шаблон ВМ предусматривает подключение ее сетевого интерфейса к конкретной виртуальной сети, то проверяется ее связь с кластером, к которому относится хост, выбранный на шаге 5. Если данная не установлена, то ВМ не может быть запущена (конец процедуры).

Шаг 9. Выбор виртуальных сетей из списков, сформированных на предыдущем шаге, для NIC, предусматривающих автоматический выбор сети, в соответствии с политикой, установленной в шаблоне ВМ, или *политикой по умолчанию* для всех ВМ, заданной в файле конфигурации планировщика.

Параметр `NIC/SCHED_RANK` шаблона ВМ определяет пользовательский критерий выбора виртуальной сети (способ вычисления ее ранга). Файл конфигурации планировщика позволяет применять как типовые критерии, так и пользовательский критерий. Параметр `NIC/SCHED_RANK` в шаблоне ВМ имеет приоритет над политикой по умолчанию.

Пользовательский критерий описывается по правилам, приведенным в п. 2.3.2. В файле конфигурации планировщика он указывается в параметре

DEFAULT_NIC_SCHED/RANK и применяется, если параметр DEFAULT_NIC_SCHED/POLICY равен 2.

Ранг вычисляется на основе параметров виртуальной сети. В их число могут входить как типовые параметры, заложенные в ядро TIVM, так и те, что добавлены в шаблон виртуальной сети пользователем, а также те, что включены в состав параметров, возвращаемых модулем мониторинга. Сеть, имеющая *максимальный ранг*, считается наиболее подходящей для использования VM.

Если в шаблоне VM нет параметра NIC/SCHED_RANK, и в файле конфигурации планировщика параметр DEFAULT_NIC_SCHED/POLICY равен 0, 1 или 3, то виртуальная сеть выбирается с помощью одного из типовых критериев (п. 2.3.3).

Конец процедуры.

Описанная процедура выполняется при создании VM или ее запуске (включении после отключения). Также она может быть выполнена по отношению к работающей VM. В последнем случае результатом *перепланирования* вычислительных ресурсов может быть решение о переносе VM на другой хост.

Политика выбора вычислительных ресурсов для VM на основе рангов может задаваться как индивидуально для VM (с помощью параметров PLACEMENT/SCHED_RANK, PLACEMENT/SCHED_DS_RANK и NIC/SCHED_RANK в шаблоне VM), так и для всех VM, у которых соответствующий параметр в шаблоне не указан. Политика по умолчанию определяется в файле конфигурации планировщика (п. 2.3.4).

Задания по подбору вычислительных ресурсов для VM формируют очередь и выполняются планировщиком в соответствующем порядке (по дисциплине FIFO). Для конкретных VM данный порядок может быть изменен. Для этого в шаблоне VM задается атрибут USER_PRIORITY с целочисленным значением. VM, имеющие такой атрибут, обрабатываются планировщиком в первую очередь в порядке убывания значения USER_PRIORITY.

При оценке достаточности ресурсов хоста для размещения VM на шаге 2 требуемое число процессорных ядер и объем памяти, указанные в шаблоне VM, сравниваются с соответствующими показателями свободных ресурсов хоста FREE_CPU и FREE_MEM, полученными из модуля мониторинга. При этом часть ресурсов хоста может быть зарезервирована для поддержки гипервизора. Это обеспечивают параметры RESERVED_CPU и RESERVED_MEM (см. табл. 2.1, 2.3). Их значения вычитаются из максимальных объемов процессорных ресурсов и памяти хоста, в результате чего определяются объемы ресурсов, которые могут выделяться VM. Параметры RESERVED_CPU и RESERVED_MEM могут задаваться как в шаблоне хоста, так и в шаблоне

кластера (п. 2.2.10). Во втором случае они применяются для всех хостов кластера, в шаблонах которых эти параметры отсутствуют.

2.3.2 Пользовательские критерии выбора вычислительных ресурсов

Пользовательский критерий – это способ вычисления ранга хоста, системного хранилища данных или виртуальной сети, заданный в шаблоне VM или в файле конфигурации планировщика, как политика по умолчанию. В шаблоне VM пользовательские критерии указываются в параметрах:

- PLACEMENT/SCHED_RANK – для хостов виртуализации;
- PLACEMENT/SCHED_DS_RANK – для хранилищ данных;
- NIC/SCHED_RANK – для виртуальных сетей.

В файле конфигурации планировщика пользовательские критерии задают параметры:

- DEFAULT_SCHED/RANK – для хостов виртуализации. Критерий применяется, если соответствующий параметр DEFAULT_SCHED/POLICY равен 3;

- DEFAULT_DS_SCHED/RANK – для хранилищ данных. Критерий применяется, если соответствующий параметр DEFAULT_DS_SCHED/POLICY равен 2;

- DEFAULT_NIC_SCHED/RANK – для виртуальных сетей. Критерий применяется, если соответствующий параметр DEFAULT_NIC_SCHED/POLICY равен 2.

Значением перечисленных параметров служит строка, содержащая формулу вычисления ранга:

```
formula ::= expression
expression ::= number | attr
              | expression + expression
              | expression - expression
              | expression * expression
              | expression / expression
              | -expression
              | (expression)
```

В определении формулы:

expression – выражение, имеющие числовой результат;

number – число;

attr – атрибут хоста, хранилища данных или виртуальной сети, имеющий числовое значение, определенный в соответствующем шаблоне или возвращаемый модулем мониторинга.

К выражениям могут применяться арифметические действия. При вычислении ранга используются операции с плавающей запятой, а итоговый результат округляется до целого значения.

Примеры пользовательских критериев в шаблоне VM:

```
PLACEMENT = [  
  SCHED_RANK = "FREE_CPU * 100 - TEMPERATURE",  
  SCHED_DS_RANK = "FREE_MB"  
]  
NIC = [  
  NETWORK_MODE = "auto",  
  SCHED_RANK = "-USED_LEASES"  
]
```

Примеры пользовательских критериев в файле конфигурации планировщика:

```
DEFAULT_SCHED = [  
  policy = 3,  
  RANK = "FREE_CPU * 100 - TEMPERATURE"  
]  
DEFAULT_DS_SCHED = [  
  policy = 2,  
  rank = "FREE_MB"  
]  
DEFAULT_NIC_SCHED = [  
  policy = 2,  
  rank = "-USED_LEASES"  
]
```

Критерий выбора ресурса – максимальное значение ранга. Критерий, установленный для VM, имеет приоритет над критерием в составе политики по умолчанию.

2.3.3 Типовые критерии выбора вычислительных ресурсов

Типовые критерии выбора хоста виртуализации, системного хранилища данных и виртуальной сети, в наибольшей степени подходящих для обеспечения функционирования VM, приведены в табл. 2.4. Критерий выбора ресурса – максимальное значение ранга.

Таблица 2.4 – Типовые критерии выбора вычислительных ресурсов, предусматриваемые политикой их распределения по умолчанию

Название	Описание	Значение параметра policy	Способ вычисления ранга
Типовые критерии выбора хоста виртуализации			
Уплотнение	Цель – минимизировать число используемых хостов, уменьшая фрагментацию распределения по ним VM. Для размещения VM приоритет имеют хосты с наибольшим числом запущенных VM	DEFAULT_SCHED/ policy = 0	RUNNING_VMS
Равномерное распределение	Цель – максимизировать ресурсы хостов, доступные VM. Для размещения VM приоритет имеют хосты с наименьшим числом запущенных VM	DEFAULT_SCHED/ policy = 1	-RUNNING_VMS
Равномерное распределение по нагрузке на процессоры	Цель – максимизировать процессорные ресурсы хостов, доступные VM. Для размещения VM приоритет имеют хосты с наибольшим значением показателя FREE_CPU (процент не используемых ресурсов процессоров)	DEFAULT_SCHED/ policy = 2	FREE_CPU
Выбор по приоритетам, заданным вручную	Приоритеты хостов задаются вручную в их шаблонах с помощью параметра PRIORITY. Для размещения VM выбирается хост с наибольшим значением приоритета	DEFAULT_SCHED/ policy = 4	PRIORITY
Типовые критерии выбора системного хранилища данных			
Уплотнение	Цель – минимизировать число хранилищ данных для уменьшения фрагментации данных VM. Для использования VM приоритет имеют хранилища с наименьшим свободным пространством	DEFAULT_DS_SCHED/ policy = 0	-FREE_MB
Равномерное распределение	Цель – максимизировать объем данных в хранилище, доступный VM, и ускорить операции обмена данными. Для использования VM приоритет имеют хранилища с наибольшим объемом свободного пространства	DEFAULT_DS_SCHED/ policy = 1	FREE_MB
Выбор по приоритетам, заданным вручную	Приоритеты хранилищ данных задаются вручную в их шаблонах с помощью параметра PRIORITY. Для использования VM выбирается хранилище с наибольшим значением приоритета	DEFAULT_DS_SCHED/ policy = 3	PRIORITY

Название	Описание	Значение параметра policy	Способ вычисления ранга
Типовые критерии выбора виртуальной сети			
Уплотнение	Цель – минимизировать число виртуальных сетей. Для использования VM приоритет имеют сети с наименьшим числом свободных адресов	DEFAULT_NIC_SCHED/policy = 0	USED_LEASES-SIZE
Равномерное распределение	Цель – максимизировать доступные VM сетевые ресурсы и ускорить операции сетевого обмена данными. Для использования VM приоритет имеют сети с наибольшим число свободных адресов	DEFAULT_NIC_SCHED/policy = 1	SIZE-USED_LEASES
Выбор по приоритетам, заданным вручную	Приоритеты виртуальных сетей задаются вручную в их шаблонах с помощью параметра PRIORITY. Для использования VM выбирается сеть с наибольшим значением приоритета	DEFAULT_NIC_SCHED/policy = 3	PRIORITY

2.3.4 Файл конфигурации планировщика

Параметры планировщика задаются в файле конфигурации /etc/tiv/sched.conf. Файл имеет текстовый формат. Параметры описываются в виде пар name = value по одному на строке. Помимо одиночных параметров предусмотрены группы:

```
groupName = [
    parameter1Name = value,
    parameter2Name = value,
    ...
]
```

Регистр символов в именах параметров и групп не имеет значения. Строковые значения приводятся в двойных кавычках. Регистр символов значений является существенным.

Состав параметров в файле конфигурации приведен в табл. 2.5.

Таблица 2.5 – Параметры планировщика, задаваемые в файле его конфигурации

Имя	Описание
MESSAGE_SIZE	Размер буфера (в байтах) для представления ответов базового модуля управления виртуализацией
TIV_XMLRPC	Сетевой адрес для взаимодействия с базовым модулем управления виртуализацией TIVM (по протоколу XML-RPC). Значение по умолчанию – "http://localhost:2633/RPC2" (планировщик функционирует на том же сервере, что и базовый модуль)

Имя	Описание
SCHED_INTERVAL	Интервал (в секундах) между циклами работы планировщика. Значение по умолчанию – 30
MAX_VM	Максимальное число VM, обрабатываемых в одном цикле планировщика. Значение по умолчанию – 5000. Ноль означает, что в цикле будут обрабатываться все задания VM, поставленных в очередь
MAX_DISPATCH	Максимальное число VM, фактически размещенных на хостах виртуализации в одном цикле планировщика. Значение по умолчанию – 30
MAX_HOST	Максимальное число VM, размещенных на данном хосте виртуализации в одном цикле планировщика. Значение по умолчанию – 1
LIVE_RESCHED	Признак режима миграции VM. Допустимые значения: 1 ("живая" миграция: VM переносится на другой хост без ее выключения), 0 ("холодная миграция": образы дисков VM сохраняются, VM выключается и затем запускается с использованием новых вычислительных ресурсов)
MEMORY_SYSTEM_DS_SCALE	Показатель учета расходов системного хранилища данных на хранение файлов контрольных точек (содержащих образы памяти VM). Требуемый VM объем системного хранилища увеличивается на величину запрошенной памяти, умноженной на данный показатель. Если показатель не равен нулю, то планировщик будет выполнять указанную корректировку при подборе хранилища данных для VM
DIFFERENT_VNETS	Признак подключения сетевых интерфейсов VM к одной или разным виртуальным сетям. Если указано значение YES, то разные NIC должны относиться к разным сетям. В случае NO это не обязательно
<i>Политика по умолчанию при выборе хостов виртуализации</i>	
DEFAULT_SCHED	Группа параметров
POLICY	0, 1, 2 или 4 – применение типового критерия (см. табл. 2.4), параметр RANK игнорируется; 3 – применение пользовательского критерия, заданного параметром RANK
RANK	Пользовательский критерий (см. п. 2.3.2)
<i>Политика по умолчанию при выборе хранилищ данных</i>	
DEFAULT_DS_SCHED	Группа параметров
POLICY	0, 1, или 3 – применение типового критерия (см. табл. 2.4), параметр RANK игнорируется; 2 – применение пользовательского критерия, заданного параметром RANK
RANK	Пользовательский критерий (см. п. 2.3.2)
<i>Политика по умолчанию при выборе виртуальных сетей</i>	
DEFAULT_NIC_SCHED	Группа параметров
POLICY	0, 1, или 3 – применение типового критерия (см. табл. 2.4), параметр RANK игнорируется; 2 – применение пользовательского критерия, заданного параметром RANK
RANK	Пользовательский критерий (см. п. 2.3.2)

Пример файла конфигурации планировщика:

```
MESSAGE_SIZE = 1073741824
TIV_XMLRPC = "http://localhost:2633/RPC2"
SCHED_INTERVAL = 30
MAX_VM = 5000
MAX_DISPATCH = 30
MAX_HOST = 1
LIVE_RESCHEDS = 0
MEMORY_SYSTEM_DS_SCALE = 0
DIFFERENT_VNETS = YES
DEFAULT_SCHED = [
    policy = 3,
    rank = "-RUNNING_VMS*50 + FREE_CPU"
]
DEFAULT_DS_SCHED = [
    policy = 1
]
DEFAULT_NIC_SCHED = [
    policy = 2,
    rank = "-USED_LEASES"
]
```

2.3.5 Запуск запланированных операций с ВМ

Планировщик обеспечивает запуск операций с ВМ в соответствии с расписанием, определенным в ее шаблоне. Планируемые операции с ВМ вводятся с помощью параметра `--schedule` команды `tivvm`. Редактирование расписания обеспечивает команда `tivvm update`. Соответствующие возможности описаны в п. 5.3.15.

2.4 Управление хранилищами данных

2.4.1 Типы хранилищ данных и технологий их реализации

TIVM поддерживает 3 типа хранилищ данных:

- хранилища образов;
- системные хранилища;
- хранилища файлов.

Для работы TIVM необходимы, как минимум, одно хранилище образов и одно системное хранилище данных. Эти хранилища могут использовать разные *технологии хранения* и режимы передачи данных между хранилищами:

- файловая система (образы дисков ВМ хранятся в виде файлов);
- менеджер логических томов (Logical Volume Manager – LVM);
- программно-реализуемая распределенная система хранения данных

Ceph;

- система сопоставления виртуальных и физических устройств хранения данных (Raw Device Mapping – RDM);

- система хранения iSCSI-Libvirt.

TIVM реализует единый набор команд для управления хранилищами данных независимо от их типа и используемой технологии. Однако для каждого сочетания типа и технологии в командах и шаблонах фигурируют специфичные атрибуты, а взаимодействие с устройствами хранения данных поддерживают соответствующие драйверы.

В перечисленных выше технологиях хранения могут применяться разные *режимы передачи данных* между хранилищем данных и системным хранилищем (табл. 2.6).

Таблица 2.6 – Типы хранилищ данных по используемым технологиям и поддерживаемые ими режимы передачи данных

Технология	Описание
Файловая система	<i>shared</i> – хранилище образов доступно всем хостам кластера в составе разделяемой файловой системы. При запуске ВМ образ диска копируется или привязывается ссылкой к системному хранилищу
	<i>ssh</i> – доступ к хранилищу образов осуществляется по протоколу SSH. При запуске ВМ образ диска должен быть скопирован по SSH в локальное системное хранилище, выделенное целевому хосту
	<i>qcow2</i> – аналогичен режиму <i>shared</i> , но использует формат образов дисков <i>Qcow2</i>
Ceph	<i>ceph</i> – образы дисков в хранилище образов и диски запущенных ВМ размещаются в общем пуле Ceph
	<i>ssh</i> – при запуске ВМ образ диска экспортируется в виде файла блочного устройства (<i>rbd</i> -файла), копируется по SSH в локальное системное хранилище, выделенное целевому хосту
LVM	<i>fs_lvm</i> – хранилище образов доступно всем хостам кластера в составе разделяемой файловой системы. При создании ВМ на основе образа диска в локальном системном хранилище целевого хоста формируется логический том, из которого запускается ВМ
RDM	<i>dev</i> – технология используется в хранилище образов, обеспечивая RDM-доступ к образам, как к блочным устройствам, заранее подготовленным и доступным гипервизорам. RDM-хранилище образов применяется для регистрации этих блочных устройств на целевых хостах. Совместно с таким хранилищем может использоваться системное хранилище, основанное на файловой системе и действующее в режимах <i>shared</i> или <i>ssh</i>
iSCSI-Libvirt	<i>iscsi</i> – технология используется в хранилище образов, обеспечивая доступ к образам, как к iSCSI-целям (<i>iSCSI-targets</i>), заранее подготовленным и доступным гипервизорам. Данный тип хранилища образов применяется для регистрации iSCSI-томов на хостах, выбранных для развертывания ВМ. Совместно с таким хранилищем может использоваться системное хранилище, основанное на файловой системе и действующее в режимах <i>shared</i> или <i>ssh</i>

Помимо хранилища образов и системного хранилища данных, которые обязательно должны быть в составе облачной среды, TIVM позволяет включать в нее *хранилища файлов*. В подобном хранилище могут размещаться, например, файлы ядра ВМ и RAM-диски. Реализация хранилища данного типа базируется на стандартных средствах работы с файловой системой, имеющихся в ОС.

Набор общих атрибутов хранилища данных представлен в табл. 2.7.

Таблица 2.7 – Общие атрибуты хранилища данных

Имя	Описание
NAME	Имя хранилища
TYPE	Тип (системное хранилище – SYSTEM_DS, хранилище образов – IMAGE_DS, хранилище файлов – FILE_DS)
TM_MAD	Драйвер менеджера передачи данных
DS_MAD	Драйвер хранилища данных
RESTRICTED_DIRS	Список путей, которые нельзя использовать для сохранения образов дисков, регистрируемых в системе. Пути разделяются пробелами
SAFE_DIRS	Список путей, исключаемых из RESTRICTED_DIRS. Пути разделяются пробелами
NO_DECOMPRESS	YES – не разворачивать (разархивировать) файл образа диска при его регистрации в системе. По умолчанию (значение NO) файл образа при регистрации распаковывается
LIMIT_TRANSFER_BW	Максимальная скорость передачи данных при загрузке образа при обращении к хранилищу по URL. По умолчанию значение выражается в байтах в секунду. В его конце может быть указана буква К, М или G, обозначающая кило-, мега- и гигабайты в секунду
DATASTORE_CAPACITY_CHECK	YES – свободный объем хранилища проверяется перед созданием в нем образа диска ВМ. NO – указанная проверка не выполняется
LIMIT_MB	Максимальный объем данных в хранилище в мегабайтах
BRIDGE_LIST	Список разделенных пробелами имен хостов виртуализации (мостов хранилища), которые могут добавлять в хранилище образы дисков
STAGING_DIR	Путь на хосте, выступающем в качестве моста хранилища, к папке для копирования образов дисков перед их перемещением в целевое местоположение. Значение по умолчанию – /var/tmp.
DRIVER	Драйвер, используемый для отображения образов дисков. Данное значение переопределяет драйвер, указанный в атрибутах образа или шаблоне ВМ
COMPATIBLE_SYS_DS	Список идентификаторов системных хранилищ данных, совместимых с данным хранилищем образов. Указывается только для хранилища образов

Для управления хранилищами данных предназначена группа команд CLI – **tivdatastore**:

```
tivdatastore command [args] [options]
```

2.4.2 Регистрация хранилища

Для регистрации хранилища данных служит команда **tivdatastore create**:

```
tivdatastore create file [options]
```

Аргумент *file* – имя файла шаблона хранилища.

Допустимый параметр (секция [options]):

-c, --cluster *id* | *name* – идентификатор или имя кластера, с которым связывается хранилище.

Связи хранилища и кластера могут определяться с помощью команд `tivcluster adddatastore` и `tivcluster deldatastore` (см. п. 2.2.9).

Команда `tivdatastore create` возвращает идентификатор зарегистрированного хранилища данных. Дальнейшие операции с зарегистрированным в TIVM хранилищем могут выполняться по его имени или идентификатору.

Файл шаблона хранилища определяет его параметры. Он имеет текстовый формат. Параметры описываются в виде пар `name = value` по одному на строке. Регистр символов в именах параметров не имеет значения. Состав и допустимые значения параметров зависят от типа хранилища, технологии хранения и режима передачи данных между хранилищами (см. табл. 2.6). Минимально необходимые наборы параметров для разных видов хранилищ представлены в табл. 2.8.

Таблица 2.8 – Минимально необходимые наборы параметров хранилища данных, указываемые в шаблоне при его создании

Технология	Тип хранилища данных	
	Системное хранилище	Хранилище образов
Файловая система	<ul style="list-style-type: none"> • NAME – имя хранилища; • TYPE = SYSTEM_DS – тип хранилища; • TM_MAD – драйвер менеджера передачи данных. Значение определяет режим передачи данных. Одно из значений: <i>shared</i>, <i>ssh</i> или <i>qcow2</i> 	<ul style="list-style-type: none"> • NAME – имя хранилища; • TYPE = IMAGE_DS – тип хранилища; • TM_MAD – драйвер менеджера передачи данных. Значение определяет режим передачи данных. Одно из значений: <i>shared</i>, <i>ssh</i> или <i>qcow2</i>; • DS_MAD = <i>fs</i> – драйвер хранилища
	<p><i>Примечание.</i> Режим передачи данных (значение параметра TM_MAD) должно совпадать для используемых совместно системного хранилища и хранилища образов. Возможная конфигурация: хранилище образов в режиме <i>shared</i> и два системных хранилища: одно в режиме <i>shared</i>, второе – <i>ssh</i>.</p>	
Ceph	<ul style="list-style-type: none"> • NAME – имя хранилища; 	<ul style="list-style-type: none"> • NAME – имя хранилища;

Технология	Тип хранилища данных	
	Системное хранилище	Хранилище образов
	<ul style="list-style-type: none"> • TYPE = SYSTEM_DS - тип хранилища; • TM_MAD - драйвер менеджера передачи данных. Значение определяет режим передачи данных. Одно из значений: ceph или ssh; • POOL_NAME - имя пула Ceph; • CEPH_HOST - разделенный пробелами список имен хостов, служащих мониторами в пуле Ceph (например, "host1 host2:port2"); • BRIDGE_LIST - разделенный пробелами список имен хостов (мостов хранилища), которые имеют доступ к нему, но относятся к кластеру, не связанному с хранилищем. Мосты позволяют переносить данные между кластерами; • CEPH_USER - имя пользователя Ceph; • CEPH_SECRET - ключ доступа для управления Ceph 	<ul style="list-style-type: none"> • TYPE = IMAGE_DS - тип хранилища; • TM_MAD = ceph - драйвер менеджера передачи данных; • DS_MAD = ceph - драйвер хранилища; • DISK_TYPE = RBD - тип дискового устройства, через которое экспортируется образ; • POOL_NAME - имя пула Ceph; • CEPH_HOST - разделенный пробелами список имен хостов, служащих мониторами в пуле Ceph (например, "host1 host2:port2"); • BRIDGE_LIST - разделенный пробелами список имен хостов (мостов хранилища), которые имеют доступ к нему, но относятся к кластеру, не связанному с хранилищем. Мосты позволяют переносить данные между кластерами; • CEPH_USER - имя пользователя Ceph; • CEPH_SECRET - ключ доступа для управления Ceph
	<p><i>Примечание.</i> Значения параметров POOL_NAME, CEPH_HOST, BRIDGE_LIST, CEPH_USER и CEPH_SECRET должны совпадать для используемых совместно системного хранилища и хранилища образов</p>	
LVM	<ul style="list-style-type: none"> • NAME - имя хранилища; • TYPE = SYSTEM_DS - тип хранилища; • TM_MAD = fs_lvm - драйвер менеджера передачи данных; • BRIDGE_LIST - разделенный пробелами список имен хостов, имеющих доступ к LVM-томам для их мониторинга 	<ul style="list-style-type: none"> • NAME - имя хранилища; • TYPE = IMAGE_DS - тип хранилища; • TM_MAD = fs_lvm - драйвер менеджера передачи данных; • DS_MAD = fs - драйвер хранилища; • DISK_TYPE = BLOCK - тип дискового устройства, через которое экспортируется образ
RDM	<p>Для эмуляции энергозависимых дисков и контекстных устройств может быть определено системное хранилище данных, основанное на</p>	<ul style="list-style-type: none"> • NAME - имя хранилища; • TYPE = IMAGE_DS - тип хранилища; • TM_MAD = dev - драйвер менеджера передачи данных;

Технология	Тип хранилища данных	
	Системное хранилище	Хранилище образов
	файловой системе и использующее режим <i>shared</i> или <i>ssh</i>	<ul style="list-style-type: none"> • DS_MAD = dev – драйвер хранилища; • DISK_TYPE = BLOCK – тип экспортируемого дискового устройства
iSCSI-Libvirt	--""--	<ul style="list-style-type: none"> • NAME – имя хранилища; • TYPE = IMAGE_DS – тип хранилища; • TM_MAD = iscsi_libvirt – драйвер менеджера передачи данных; • DS_MAD = iscsi_libvirt – драйвер хранилища; • DISK_TYPE = ISCSI – тип экспортируемого дискового устройства; • ISCSI_HOST – имя или адрес хоста, выполняющего функции iSCSI-цели

Помимо параметров, указанных в табл. 2.8, в файл шаблона хранилища данных могут включаться другие параметры (как общие, перечисленные в табл. 2.7, так и специфичные для типа и технологии).

Пример файла шаблона хранилища образов (*ds_img_tmpl*), основанного на файловой системе:

```
NAME = ds_image1
TYPE = IMAGE_DS
TM_MAD = shared
DS_MAD = fs
```

Пример команды регистрации данного хранилища:

```
tivdatastore create ds_img_tmpl -cluster cluster1
```

Для регистрации *хранилища файлов* в шаблоне должны быть представлены следующие параметры:

- NAME – имя хранилища;
- TYPE = FILE_DS – тип хранилища;
- TM_MAD = ssh – драйвер менеджера передачи данных;
- DS_MAD = fs – драйвер хранилища.

2.4.3 Переименование хранилища

Имя зарегистрированного хранилища может быть *изменено* с помощью команды **tivdatastore rename**:

```
tivdatastore rename dsIdOrName newName
```

Аргументы команды:

- *dsIdOrName* – идентификатор или имя хранилища данных;
- *newName* – новое имя кластера.

Пример:

```
tivdatastore rename 101 ds sys 101new
```

2.4.4 Определение прав доступа к хранилищу

Права доступа к хранилищу данных определяют:

- его владелец (пользователь TIVM);
- группа пользователей TIVM, связанная с хранилищем;
- разрешения на операции с хранилищем.

Владельца и группу пользователей определяет команда **tivdatastore**

chown:

```
tivdatastore chown rangeDsId | dsIdList userIdOrName  
[groupIdOrName]
```

Аргументы команды:

- *rangeDsId* – диапазон идентификаторов хранилищ данных в форме *n..m, k*;
- *dsIdList* – список идентификаторов или имен хранилищ данных, перечисляемых через запятую. Список может состоять из одного элемента;
- *userIdOrName* – идентификатор или имя пользователя-владельца хранилища;
- *groupIdOrName* – идентификатор или имя группы пользователей.

Примеры:

```
tivdatastore chown 101 user5 userGroup1  
tivdatastore chown 101..103,106 user5 userGroup1  
tivdatastore chown 101, 102 user5  
tivdatastore chown ds_sys_101, ds_img_101 user5 userGroup1
```

В команде **tivdatastore chown** аргумент *groupIdOrName*, представляющий группу пользователей, может отсутствовать, а указание пользователя-владельца обязательно. Для определения только *группы*, связанной с хранилищем, предназначена команда **tivdatastore chgrp:**

```
tivdatastore chgrp rangeDsId | dsIdList groupIdOrName
```

Аргументы команды:

- *rangeDsId* – диапазон идентификаторов хранилищ данных в форме *n..m, k*;
- *dsIdList* – список идентификаторов или имен хранилищ данных, перечисляемых через запятую. Список может состоять из одного элемента;

- *groupIdOrName* – идентификатор или имя группы пользователей.

Разрешения на операции используются в TIVM для управления правами доступа к вычислительным ресурсам различных типов (шаблонам, ВМ, виртуальным сетям и др.), включая хранилища данных. Разрешения на операции с ресурсом определяет набор признаков, отражающих доступность 3 классов операций (использование, управление и администрирование) для 3 категорий пользователей (владелец, группа, прочие пользователи).

Класс операций *использования* ресурса включает все операции, в которых он применяется, но не может быть модифицирован. Например, для образа диска ВМ это операции его копирования, получения атрибутов, запуска ВМ с его использованием, присоединения к работающей ВМ.

К *управлению* ресурсом относятся операции, в которых в ходе его использования он может быть модифицирован. Например, для образа диска ВМ это операции изменения его состояния, имени, типа, создания и удаления снимка, обновления шаблона, установки и снятия блокировки, удаления и др.

Администрирование – особый класс операций управления, меняющих базовые атрибуты ресурса и оказывающих существенное влияющих на его возможности и доступность для пользователей. Эти операции предоставляются администраторам системы. Как правило, к ним относятся управление пользователями и правами доступа.

Разрешения задает 3-разрядное 8-ричное число: *ugo*. Первый разряд (считая слева) представляет разрешения для владельца ресурса (*u*), второй разряд – разрешения для группы пользователей ресурса (*g*), третий – разрешения для прочих пользователей (*o*). Разряды двоичного представления чисел *u*, *g* и *o* кодируют права доступа:

0-й бит (считая справа) – администрирование;

1-й бит – управление;

2-й бит – использование.

Таким образом, 8-ричный разряд представляет следующие варианты разрешений на операции с ресурсом:

0 – доступ запрещен;

1 – администрирование;

2 – управление;

3 – управление и администрирование;

4 – использование;

5 – использование и администрирование;

6 – использование и управление;

7 – использование, управление и администрирование.

Примеры:

- 754 – все операции доступны владельцу, использование и администрирование – пользователям группы ресурса, использование – прочим пользователям;

- 664 – использование и управление доступны владельцу и пользователям группы, только использование – прочим пользователям;

- 640 – использование и управление доступны владельцу, использование – пользователям группы;

- 777 – все пользователи имеют доступ ко всем операциям;

- 000 – ресурс недоступен пользователям кроме владельца и администратора системы, которые могут менять разрешения.

Для установки разрешений на операции с хранилищем данных служит команда **tivdatastore chmod**:

```
tivdatastore chmod rangeDsId | dsIdList permissions
```

Аргументы команды:

- *rangeDsId* – диапазон идентификаторов хранилищ данных в форме *n..m, k*;

- *dsIdList* – список идентификаторов или имен хранилищ данных, перечисляемых через запятую. Список может состоять из одного элемента;

- *permissions* – разрешения, представляемые 3-разрядным 8-ричным кодом.

Примеры:

```
tivdatastore chmod 101 754
tivdatastore chmod 101..103,106 664
tivdatastore chmod ds sys 101, ds img 101 660
```

2.4.5 Отмена регистрации хранилища

Для *отмены регистрации хранилища данных* служит команда **tivdatastore delete**:

```
tivdatastore delete rangeDsId | dsIdList
```

Аргумент команды: *rangeDsId* – диапазон идентификаторов хранилищ в форме *n..m, k* – либо *dsIdList* – список идентификаторов или имен хранилищ, перечисляемых через запятую. Список может состоять из одного элемента.

Примеры:

```
tivdatastore delete ds_image1
tivdatastore delete 101,104
tivdatastore delete 101..103,105
```

Первая команда отменяет регистрацию хранилища данных с именем *ds_image1*, вторая делает то же самое по отношению к хранилищам с

идентификаторами 101 и 104, третья отменяет регистрацию хранилищ с идентификаторами 101, 102, 103 и 105.

2.4.6 Получение информации о хранилище

Для получения информации о хранилище данных предназначена команда **tivdatastore show**:

```
tivdatastore show dsIdOrName [options]
```

Аргумент команды:

- *dsIdOrName* – идентификатор или имя хранилища.

Допустимые параметры (секция [options]):

--decrypt – получить дешифрованные атрибуты;

-x, --xml – представить информацию в XML-формате.

Пример:

```
tivdatastore show 101
```

Результат:

```
DATASTORE 101 INFORMATION
ID          : 101
NAME        : sw7000VDI-system
USER        : user5
GROUP       : userGroup1
CLUSTERS    : 0
TYPE        : SYSTEM
DS_MAD      : -
TM_MAD      : qcow2
BASE_PATH   : /var/lib/tiv/datastores/101
DISK_TYPE   : FILE
STATE       : READY

DATASTORE CAPACITY
TOTAL:      : 399.8G
FREE:       : 384.6G
USED:       : 15.2G
LIMIT:      : -

PERMISSIONS
OWNER       : um-
GROUP      : u--
OTHER      : ---

DATASTORE TEMPLATE
ALLOW_ORPHANS="NO"
DISK_TYPE="FILE"
DS_MIGRATE="YES"
RESTRICTED_DIRS="/"
SAFE_DIRS="/var/tmp"
SHARED="YES"
TM_MAD="qcow2"
```

```
TYPE="SYSTEM_DS"
```

2.4.7 Получение списка хранилищ

Список зарегистрированных хранилищ данных возвращает команда **tivdatastore list**:

```
tivdatastore list [options]
```

Допустимые параметры (секция [options]):

--adjust *col1, col2, ...* – выравнивать размеры возвращаемой таблицы, чтобы содержимое столбцов с указанными номерами (*col1, col2, ...*) не обрезалось;

--csv – представить информацию в CSV-формате;

--csv-del *delimiter* – в CSV-формате использовать в качестве разделителя символ *delimiter*;

--describe – выводить описания столбцов;

--expand [*name1=prop1, name2=prop2, ...*] – расширять возвращаемую таблицу, чтобы она занимала всю ширину терминала. Параметры *name1* и *name2* – имена столбцов, *prop1* и *prop2* – их относительная ширина в долях от единицы (например, 0.4 и 0.6). Если относительная ширина столбцов не указана, то таблица расширяется пропорционально их содержимому. Если столбцы не указаны, то расширяются все столбцы таблицы;

-f, --filter *cond1, cond2, ...* – фильтровать выводимую информацию по условиям *cond1, cond2, ...*. Каждое условие задает тройка <имя столбца, предикат, значение>. Набор предикатов: =, != (не равно), <, >, <=, >=, ~ (начинается с). Условия соединяются оператором, указанным в параметре *operator*;

-l, --list *col1, col2, ...* – выводить только столбцы с указанными номерами (*col1, col2, ...*);

-c, --listconf *conf* – выводить только столбцы, заданные в заранее определенном списке *conf*;

--no-expand – запретить расширение столбцов таблицы;

--no-header – не выводить заголовков таблицы;

--no-pager – не разбивать таблицу на страницы;

-n, --numeric – не преобразовывать идентификаторы пользователя и группы (выводить их в цифровом виде);

--operator *oper* – логический оператор, связывающий условия фильтра, заданного параметром *filter*. Допустимые значения *oper* – AND, OR. Значение по умолчанию – AND;

-s, --size *name1=size1, name2=size2, ...* – установить ширину столбцов (в символах) с указанными именами;

-x, --xml – представить информацию в XML-формате.

Пример:

```
tivdatastore list
```

Результат:

```
tivdatastore list
ID      NAME      SIZE  AVAIL  CLUSTERS  IMAGES  TYPE  DS      TM      STAT
0      ds_sys1   200G  52%    0          0      sys  -      shared  on
1      ds_img1   150G  64%    0          3      img  fs     shared  on
2      ds_files1 100G  13%    0          0      fil  fs     ssh     on
```

В списке для каждого хранилища приводятся его идентификатор, имя, общий объем, процент свободного объема, идентификаторы кластеров, с которым связано хранилище, число хранимых образов, тип, имена драйвера хранилища и драйвера передачи данных, а также состояние.

2.4.8 Блокировка и снятие блокировки хранилища для планировщика

Системное хранилище данных может быть заблокировано для предотвращения размещения в нем образов дисков запускаемых ВМ. Команда блокировки – **tivdatastore disable**:

```
tivdatastore disable rangeDsId | dsIdList
```

Аргумент команды: *rangeDsId* – диапазон идентификаторов хранилищ в форме *n..m,k* – либо *dsIdList* – список идентификаторов или имен хранилищ, перечисляемых через запятую. Список может состоять из одного элемента.

В списке хранилищ, возвращаемых командой **tivdatastore list**, заблокированное хранилище имеет состояние "выключено" (в столбце **STAT** указано **off**). Однако блокировка не означает отключения хранилища, это только запрет его применения для новых ВМ. Заблокированные хранилища доступны для мониторинга, а использующие их ВМ продолжают работать.

Для отмены блокировки предназначена команда **tivdatastore enable**:

```
tivdatastore enable rangeDsId | dsIdList
```

Она имеет те же аргументы, что **tivdatastore disable**.

Пример:

```
tivdatastore disable 101
tivdatastore list
ID      NAME      SIZE  AVA  CLUSTERS  IMAGES  TYPE  DS      TM      STAT
102     ds_system1 97.9G 76%    0          0      sys  -      shared  on
101  sw7000VDI-system 399.8G 96%    0          0      sys  -      qcow2  off
100     sw7000VDI 399.8G 96%    0          2      img  fs     qcow2  on
2       files     97.9G 76%    0          0      fil  fs     ssh     on
1       default   97.9G 76%    0          6      img  fs     ssh     on
```

0	system	-	-	0	0	sys	-	ssh	on
tivdatastore enable 101									
tivdatastore list									
ID	NAME	SIZE	AVA	CLUSTERS	IMAGES	TYPE	DS	TM	STAT
102	ds_system1	97.9G	76%	0	0	sys	-	shared	on
101	sw7000VDI-system	399.8G	96%	0	0	sys	-	qcow2	on
100	sw7000VDI	399.8G	96%	0	2	img	fs	qcow2	on
2	files	97.9G	76%	0	0	fil	fs	ssh	on
1	default	97.9G	76%	0	6	img	fs	ssh	on
0	system	-	-	0	0	sys	-	ssh	on

2.4.9 Изменение шаблона хранилища

Атрибуты, содержащиеся в шаблоне хранилища, могут быть изменены с помощью команды **tivdatastore update**. Также эта команда позволяет включить в шаблон другие атрибуты (параметр `--append`).
Формат команды:

```
tivdatastore update dsIdOrName [file] [options]
```

Аргументы команды:

- *dsIdOrName* – идентификатор или имя хранилища данных;
- *file* – имя файла шаблона хранилища.

Команда загружает атрибуты из текстового файла *file*. Его формат описан в п. 2.4.2.

Если имя файла шаблона не указано, то TIVM запустит редактор для изменения текущего содержимого шаблона.

Параметр `--append` (`-a`) позволяет задать признак *property*, добавляемый в шаблон. Форма его определения: `name = value`. Например, можно присвоить хранилищу атрибут "SPEED" со значением "fast":

```
tivdatastore update 101 --append SPEED="fast"
```

В дальнейшем можно использовать данное значение, как условие для планировщика.

3 Управление пользователями и группами

3.1 Типы пользователей и организация управления правами доступа

TIVM реализует средства управления пользователями и их группами. Учетные записи пользователей делятся на *четыре группы*:

- **администраторы** (входят в группу администраторов и могут выполнять операции управления системой в целом);

- **обычные пользователи**, имеющие доступ к большинству функций TIVM;

- **публичные пользователи**, имеющие доступ только к базовым функциям и общедоступным интерфейсам;

- **служебные пользователи** (учетные записи, используемые сервисами TIVM для обработки запросов аутентификации).

Доступ пользователей к функциям и ресурсам TIVM управляется *системой разрешений*, которая похожа на разрешения операций в ОС UNIX. По умолчанию только владелец ресурса (например, VM или образа ее диска) может оперировать им. Пользователи могут делиться ресурсами, предоставляя права на операции с ними другим пользователям в своей группе или любому другому пользователю системы.

Для группы пользователей можно назначать ее администраторов. По умолчанию *администратор группы* может создавать ее пользователей и управлять ее ресурсами с помощью CLI или специального представления пользовательского интерфейса TIVM. Группу пользователей также можно отнести к ВДЦ, который представляет пул физических ресурсов TIVM (хосты виртуализации, хранилища данных и виртуальные сети).

Наряду с пользователями и группами подсистема аутентификации TIVM отвечает за аутентификацию и авторизацию пользовательских запросов.

Любой интерфейс, обслуживающий запросы к системе (CLI, web-интерфейс TIVM, OCCI или иной поддерживаемый API), взаимодействует с ядром с помощью вызовов XML-RPC, содержащих строку сессии пользователя, которая аутентифицируется ядром TIVM. Каждая операция генерирует запрос авторизации, который проверяется на соответствие зарегистрированным правилам *списков управления доступом* (СУД). Затем ядро может предоставить разрешение или отклонить запрос.

TIVM поставляется с набором правил СУД по умолчанию, который обеспечивает его типовое применение. При необходимости администраторы TIVM могут корректировать правила СУД для более полного отражения условий применения системы.

По умолчанию аутентификация и авторизация обрабатываются ядром TIVM. Эти функции могут быть делегированы внешним программным средствам.

3.2 Управление пользователями

Пользователь в TIVM имеет уникальное *имя* и *пароль*. Для пользователей не требуются отдельные учетные записи ОС на сервере управления виртуализацией. Пользователи проходят аутентификацию с использованием строки сессии, входящей в состав каждой операции, которая проверяется ядром TIVM.

После регистрации пользователя ему присваивается *уникальный идентификатор*. Кроме того, пользователь принадлежит к определенной *группе*.

После установки ПВК "ТИВМ" в системе имеются две учетные записи администратора (`tivadmin` и `serveradmin`) и две группы по умолчанию. Проверить наличие этих записей можно с помощью команд `tivuser list` и `tivgroup list`.

В TIVM существуют различные типы пользователей.

Администратор облака – учетная запись `tivadmin`, создаваемая при первом запуске TIVM с использованием данных `TIV_AUTH`. Пользователь `tivadmin` имеет права выполнения любой операции над любым объектом. Любой другой пользователь в группе `tivadmin` имеет те же права, что и `tivadmin`.

Пользователи вычислительной инфраструктуры имеют доступ к большинству функций, предлагаемых TIVM для управления ресурсами.

Администраторы групп управляют ограниченным набором ресурсов и пользователей, принадлежащих группе.

Обычные пользователи имеют доступ к упрощенному представлению TIVM с ограниченными действиями по созданию ВМ и выполнению основных операций их ЖЦ.

Пользователь `serveradmin` – учетная запись также создается при первом запуске TIVM. Его пароль генерируется случайным образом. Данная учетная запись используется web-интерфейсом TIVM для взаимодействия с другими компонентами TIVM.

Имя пользователя TIVM и пароль не могут содержать символов: горизонтальная табуляция (десятичный код символа – 9), новая строка (10), вертикальная табуляция (11), прогон страницы (12), возврат каретки (13) и пробел (32). Имя пользователя также не может включать двоеточие (58).

На сервере, на котором установлен TIVM, следует определить следующие *переменные окружения*.

- `TIV_XMLRPC` – адрес, через который демон TIVM получает запросы. Если он не установлен, то CLI будет использовать по умолчанию адрес `http://localhost:2633/RPC2`.
- `TIV_XMLRPC_TIMEOUT` – время ожидания в секундах до истечения времени ожидания запроса `xmlrpc`.
- `TIV_AUTH` – должен указывать на файл, содержащий действительный ключ аутентификации. Это может быть:
 - файл пароля с одной строкой `username:password`;
 - файл токена с одной строкой `username:token`, где токен является

действительным токеном, созданным с помощью команды `tivuser login` или вызова API.

Если значение `TIV_AUTH` не определено, то вместо него будет использовано `"$HOME/.tiv/tiv_auth"`. Если по данному пути нет файла аутентификации, то TIVM не может работать должным образом, так как это необходимо для ядра, CLI и облачных компонентов.

Наряду с параметрами `TIV_XMLRPC` и `TIV_AUTH` существует альтернативный метод указания учетных данных пользователя и адреса доступа к TIVM. Он заключается в использовании параметров командной строки. Большинство команд TIVM могут понимать следующие параметры:

- user name – имя пользователя, используемое для подключения к TIVM;
- password pass – пароль пользователя;
- endpoint URL – адрес TIVM XML-RPC.

Если задано имя пользователя, но не указан пароль, то пароль будет запрошен при подключении. Пример:

```
tivvm list --user my_user -endpoint http://tiv.host.com:2633/RPC2
Password:
...
```

- `TIV_POOL_PAGE_SIZE` - по умолчанию API-интерфейс TIVM Cloud (его используют CLI и web-интерфейс TIVM) разбивает на страницы данные результатов запросов некоторых пулов. По умолчанию размер страницы равен 300, но его можно изменить с помощью этой переменной. Числовое значение больше 2 определяет размер страницы. Чтобы отключить разбиение на страницы, следует указать не числовое значение. Примеры:

```
# Установка размера страницы, равного 5000
export TIV_POOL_PAGE_SIZE=5000

# Отключение разбиения на страницы
export TIV_POOL_PAGE_SIZE=disabled
```

- `TIV_PAGER` – команды получения списков обрабатывают свои результаты через процесс пагинации в интерактивной оболочке ОС. По умолчанию процесс пагинации использует команду `less`, но с помощью данной переменной ее можно изменить на любую другую программу.
- `TIV_LISTCONF` – позволяет использовать шаблон вывода данных списков. Шаблоны определены в `/etc/tiv/cli/tivvm.yaml`.
Пример:

```
tivvm list
ID USER      GROUP      NAME      STAT UCPU  UMEM  HOST      TIME
20 tivadmin   tivadmin   tty-20    fail  0      0K    localhost 0d 00h32
21 tivadmin   tivadmin   tty-21    fail  0      0K    localhost 0d 00h23
22 tivadmin   tivadmin   tty-22    runn  0.0    104.7M localhost 0d 00h22
```

```
export TIV_LISTCONF=user
tivvm list
```

```
ID NAME      IP          STAT UCPU  UMEM  HOST      TIME
20 tty-20     10.3.4.20  fail  0      0K    localhost 0d 00h32
21 tty-21     10.3.4.21  fail  0      0K    localhost 0d 00h23
22 tty-22     10.3.4.22  runn  0.0    104.7M localhost 0d 00h23
```

- `TIV_CERT_DIR` и `TIV_DISABLE_SSL_VERIFY` – если конечная точка XML-RPC TIVM находится за SSL-прокси, то с помощью `TIV_CERT_DIR` можно указать дополнительный каталог доверенных сертификатов. Сертификат должен иметь имя `<hash>.0`. Хэш сертификата можно получить с помощью команды:

```
openssl x509 -in <certificate.pem> -hash
```

Чтобы отключить проверку сертификата, переменной среды `TIV_DISABLE_SSL_VERIFY` следует присвоить любое значение. Этот параметр должен применяться только для тестирования, так как он делает соединение небезопасным.

- `TIV_TIVM` – URL-адрес web-интерфейса TIVM. Если значение не указано, то оно будет получено из `TIV_XMLRPC` путем изменения порта на 10180. Если переменная `TIV_XMLRPC` также не определена, то по умолчанию `TIV_TIVM` присваивается адрес `http://localhost:10180`.
- `TIVFLOW_URL`, `TIVFLOW_USER` и `TIVFLOW_PASSWORD` – переменные используются инструментами CLI TIVFLOW. Если их значения не заданы, то по умолчанию URL-адрес TIVFLOW будет `http://localhost:2474`, а пользователь и пароль будут взяты из файла `TIV_AUTH`.

- TIV_LOCATION – папка на сервере TIVM, куда установлено ПО. Переменная должна быть определена, если TIVM инсталлирован в автономном режиме (в отдельную папку).
- PATH – путь к исполняемым файлам TIVM на сервере управления виртуализацией. Переменная должна быть определена, если TIVM инсталлирован в автономном режиме (в отдельную папку). Пример команды добавления пути к значению переменной PATH:

```
export PATH=$TIV_LOCATION/bin:$PATH
```

Учетные записи пользователей TIVM управляются администратором `tivadmin` с помощью команд `tivuser create` и `tivuser delete`.

Учетная запись администратора создается командой:

```
tivuser create AdminUserLogin AdminUserPassword
ID: 5
```

Команда возвращает идентификатор созданной учетной записи. В данном примере это 5.

Далее необходимо включить созданную учетную запись в группу `tivadmin`:

```
tivuser chgrp AdminUserLogin tivadmin
```

Список зарегистрированных пользователей возвращает команда `tivuser list`. Пример ее выполнения:

```
tivuser list
ID GROUP      NAME          AUTH          PASSWORD
0  tivadmin    tivadmin      core          5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
1  tivadmin    serveradmin   server_c      1224ff12545a2e5dfeda4eddacdc682d719c26d5
. . .
5  tivadmin    AdminUserLogin core          5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
```

Информацию о данной учетной записи возвращает команда `tivuser show`. Пример ее выполнения:

```
tivuser show AdminUserLogin
USER 5 INFORMATION
ID           : 5
NAME        : AdminUserLogin
GROUP       : 0
PASSWORD    : 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
AUTH_DRIVER : core
ENABLED     : Yes
USER TEMPLATE
. . .
```

Создание учетной записи обычного пользователя осуществляется аналогично:

```
tivuser create UserLogin UserPassword
ID: 6
```

Учетная запись *публичного пользователя* отличается тем, что она использует специальный метод аутентификации `public`. Такая запись *создается*, как любая другая:

```
tivuser create PublicUserLogin PublicUserPassword
ID: 7
```

Затем для нее необходимо *сменить метод аутентификации*:

```
tivuser chauth PublicUserLogin public
```

Учетные записи служебных пользователей предназначены для авторизации сервисов TIVM. Пользователи-физические лица не применяют их непосредственно. Учетные записи, в которых задействованы методы аутентификации `server_cipher` и `server_x509`, являются служебными.

Регистрация служебной учетной записи включает ее создание с помощью `tivuser create` и смену метода аутентификации с помощью `tivuser chauth`:

```
tivuser create ServerUserLogin ServerUserPassword
ID: 8
tivuser chauth ServerUserLogin server_cipher
```

Для аутентификации в TIVM необходимо предоставить системе *пароль* или *токен аутентификации*. Способ аутентификации зависит от драйвера `AUTH_DRIVER`, установленного для данного пользователя.

По умолчанию созданной учетной записи пользователя присваивается драйвер аутентификации `core`, реализующий *механизм аутентификации по логину и паролю*. При регистрации пользователя в файл `$HOME/.tiv/tiv_auth` добавляется строка с логином и паролем в формате: `UserLogin:UserPassword`. TIVM хранит не исходные пароли, а сформированное по нему хеш-значение. Эти значения выводятся командами `tivuser list` и `tivuser show`, как было показано в примерах выше.

Механизм аутентификации с помощью токенов повышает уровень информационной безопасности. TIVM генерирует токены с установленным временем жизни, которые затем передаются пользователями для их аутентификации. По умолчанию токены также сохраняются в файле `$HOME/.tiv/tiv_auth`.

Если пользователь принадлежит нескольким группам, то токен может быть связан только с одной из них. При работе с данным токеном система будет считать, что пользователь относится только к группе этого токена.

Команда создания токена:

```
tivuser token-create
```

```
File /var/lib/tiv/.tiv/tiv_auth exists, use --force to overwrite.
Authentication Token is:
username:b61010c8ef7a1e815ec2836ea7691e92c4d3f316
```

Время жизни токена по умолчанию составляет 10 часов. Параметр `--time <seconds>` команды `tivuser token-create` позволяет установить время жизни токена. Значение `-1` устанавливает время жизни бесконечным.

При создании токена он *может быть связан с одной из групп* пользователя. Для указания группы служит параметр `--group <IdGroup>` команды `tivuser token-create`. При аутентификации по данному токену пользователь будет считаться принадлежащим только соответствующей группе, и ему будут доступны ресурсы только этой группы. Ресурс, созданный пользователем, будет приписан данной группе.

Если группа при создании токена не указана, то по умолчанию он обеспечивает доступ пользователя к ресурсам всех групп, в которые данный пользователь входит.

Список токенов пользователя входит в состав информации, возвращаемой командой `tivuser show`:

```
tivuser show UserLogin
. . .
TOKENS
      ID EGID  EGROUP      EXPIRATION
3ea673b 100   groupB      2020-09-13 23:51:52
c33ff10 100   groupB      expired
f836893 *1    users      forever
```

Звездочка в столбце `EGID` означает, что основной группой пользователя является 1, а токен не относится к конкретной группе. Значение `forever` в столбце `EXPIRATION` выводится, когда время жизни токена не установлено.

Для включения токена нужно сначала получить строку файла `tiv_auth` с информацией о нем:

```
tivuser token-set --token b6
```

Параметр `--token` задает строку, с которой должен начинаться идентификатор токена (в данном примере он начинается с "b6"). Далее строку, возвращенную командой `tivuser token-set`, следует передать команде ОС `export`:

```
export TIV_AUTH=/var/lib/tiv/.tiv/5ad20d96-964a-4e09-b550-
9c29855e6457.token; export TIV_EGID=-1
```

Команда удаления токена:

```
tivuser token-delete b6
```

В состав TIVM входит файл /usr/share/tiv/tivtoken.sh, в котором есть две вспомогательные функции: tivtokencreate и tivtokenset, позволяющие оперировать токенами аутентификации. Функция tivtokencreate поддерживает те же параметры, что и tivuser token-create: --time и --group.

Пример использования данных команд:

```
source /usr/share/tiv/tivtoken.sh
tivtokencreate
Password:
File /var/lib/tiv/.tiv/tiv_auth exists, use --force to overwrite.
Authentication Token is:
testuser:f65c77250cfd375dd83873ad68598edc6593a39e
Token loaded.
cat $TIV_AUTH
testuser:f65c77250cfd375dd83873ad68598edc6593a39e%
tivuser show
[...]
TOKENS
      ID EGID  EGROUP      EXPIRATION
3ea673b 100   groupB      2020-11-30 13:58:11
c33ff10 100   groupB      expired
f65c772 *1     users       2020-11-30 14:10:22
[...]
tivtokenset
Token loaded.
cat $TIV_AUTH
testuser:3ea673b90d318e4f5e67a83c220f57cd33618421
```

Пользователю могут быть присвоены любые дополнительные атрибуты. Это обеспечивает команда **tivuser update**, предназначенная для изменения шаблона пользователя. Например, пользователю с идентификатором 12 присваиваются атрибуты DEPARTMENT и EMAIL:

```
tivuser update 12 --append DEPARTMENT="IT"
tivuser update 12 --append EMAIL=abc@def.com
tivuser show 12
USER 12 INFORMATION
ID           : 12
NAME        : user1
GROUP       : 1
PASSWORD    : 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
AUTH_DRIVER : core
ENABLED     : Yes

USER TEMPLATE
DEPARTMENT=IT
EMAIL=abc@def.com
```

Дополнительные атрибуты могут быть использованы при контекстуализации ВМ.

Обычному пользователю доступна информация своей учетной записи. Команда `tivuser show` без параметров возвращает информацию о текущем авторизованном пользователе. В ее состав входит идентификатор пользователя, например:

```
tivuser show
USER 12 INFORMATION
ID           : 12
NAME        : user1
GROUP       : 1
PASSWORD    : 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
AUTH_DRIVER : core
ENABLED     : Yes

USER TEMPLATE
DEPARTMENT=IT
EMAIL=abc@def.com
```

Команда `tivuser passwd` позволяет установить пароль, например:

```
tivuser passwd 12 new password
```

Информация учетной записи пользователя (*профиль пользователя*) доступна в web-интерфейсе TIVM (рис. 3.1). Данная страница открывается через пункт меню пользователя "Настройки".

Настройки

Информация | Квоты | Квоты группы | Отчетность | Потребление ресурсов | Аутентификация

Информация

ID	2	
Название	tivm_user	
Порядок таблицы	-	✎
Язык интерфейса	-	✎
Представление	-	✎
Default Zone Endpoint	-	✎

Атрибуты

TOKEN_PASSWORD	7dfeb209e22dc030a3626dd7bc2656e3f9fecf3146900e7fa7c92457adfb7ebe	✎ 🗑
		+

Рис. 3.1

Операции управления пользователями также доступны в web-интерфейсе. Список пользователей, которыми может управлять данный пользователь, отображается на странице, открываемой через пункт меню Система/Пользователи (рис. 3.2).

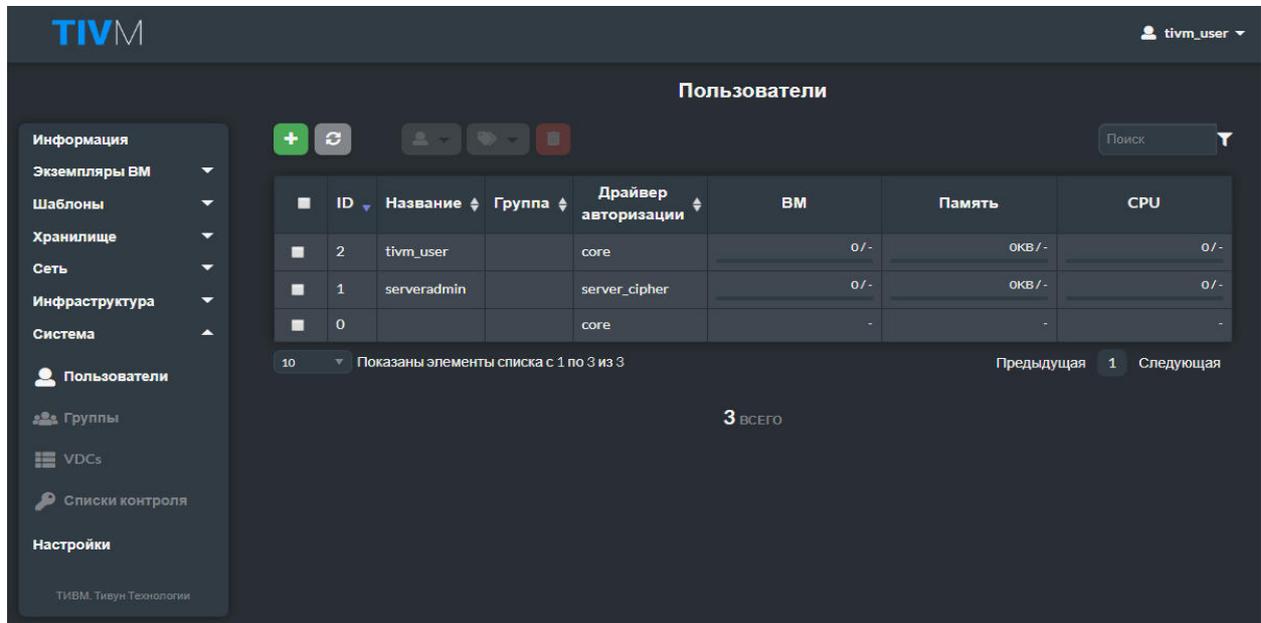


Рис. 3.2

Для смены пароля суперадминистратора *tivadmin* необходимо ввести команды:

```
tivuser passwd 0 new_password
echo 'tivadmin:new_password' > /var/lib/tiv/.tiv/tiv_auth
```

После этого необходимо перезапустить TIVM.

Для смены пароля пользователя *serveradmin* необходимо ввести команды:

```
tivuser passwd 1 --sha256 new_password
echo 'serverdmin:new_password' > /var/lib/tiv/.tiv/tivflow_auth
echo 'serverdmin:new_password' > /var/lib/tiv/.tiv/ec2_auth
echo 'serverdmin:new_password' > /var/lib/tiv/.tiv/tivgate_auth
echo 'serverdmin:new_password' > /var/lib/tiv/.tiv/occi_auth
echo 'serverdmin:new_password' > /var/lib/tiv/.tiv/tivm_auth
```

После этого необходимо перезапустить web-интерфейс TIVM.

3.3 Управление группами

Группа в TIVM позволяет изолировать пользователей и вычислительные ресурсы. Пользователь может видеть и использовать общие ресурсы других пользователей.

Группа является границей авторизации для пользователей. TIVM позволяет разделить облачную инфраструктуру и определить, какие ресурсы доступны для каждой группы, используя ВДЦ.

По умолчанию в TIVM определены две *специальные группы*. Группа `tivadmin` позволяет любому пользователю выполнять любые операции, позволяя различным пользователям действовать с теми же правами, что и пользователь `tivadmin`. Группа `users` – это группа по умолчанию, в которой создаются новые пользователи.

Для управления группами в TIVM можно использовать команду CLI `tivgroup`. Команды создания новой группы:

```
tivgroup list
  ID NAME
  0 tivadmin
  1 users
tivgroup create "new group"
ID: 100
```

Новая группа имеет ID 100, чтобы отличать специальные группы от пользовательских.

При создании новой группы также создается правило СУД, обеспечивающее поведение по умолчанию, позволяющее пользователям создавать основные ресурсы.

Для назначения пользователей в группы служит команда `tivuser chgrp`:

```
tivuser chgrp -v regularuser "new group"
USER 1: Group changed
tivgroup show 100
GROUP 100 INFORMATION
ID           : 100
NAME         : new group

USERS
ID           NAME
1            regularuser
```

Чтобы удалить пользователя из группы, его следует переместить в группу `users` по умолчанию.

После создания группы можно определить специальную учетную запись администратора. Этот пользователь с правами администратора будет иметь права администратора только для новой группы, а не для всех ресурсов в облаке TIVM, как у пользователей группы `tivadmin`.

Другим аспектом, который можно контролировать во время создания, является тип ресурсов, которые могут создавать пользователи группы.

Это может управляться как в web-интерфейсе TIVM, так и через CLI. В последнем случае детали группы передаются команде `tivgroup create` в качестве аргументов. В табл. 3.1 ниже приведено описание указанных аргументов.

Таблица 3.1 – Аргументы команды `tivgroup create`

Аргумент	Обязательность	Значение	Описание
<code>-n, --name</code>	Обязательный	Любая строка	Имя для новой группы
<code>-u, --admin_user</code>	Необязательный	Любая строка	Создает администратора для группы с указанным именем.
<code>-p, --admin_password</code>	Необязательный	Любая строка	Пароль для администратора пользователя группы
<code>-d, --admin_driver</code>	Необязательный	Любая строка	Драйвер аутентификации для пользователя группы с правами администратора
<code>-r, --resources</code>	Необязательный	Список, разделенный символом "+"	Какие ресурсы могут быть созданы групповыми пользователями (по умолчанию: "VM+IMAGE+TEMPLATE")

Пример:

```
tivgroup create --name groupA
--admin_user admin_userA --admin_password somepassword
--resources TEMPLATE+VM
```

Страница создания группы в web-интерфейсе показана на рис. 3.3.

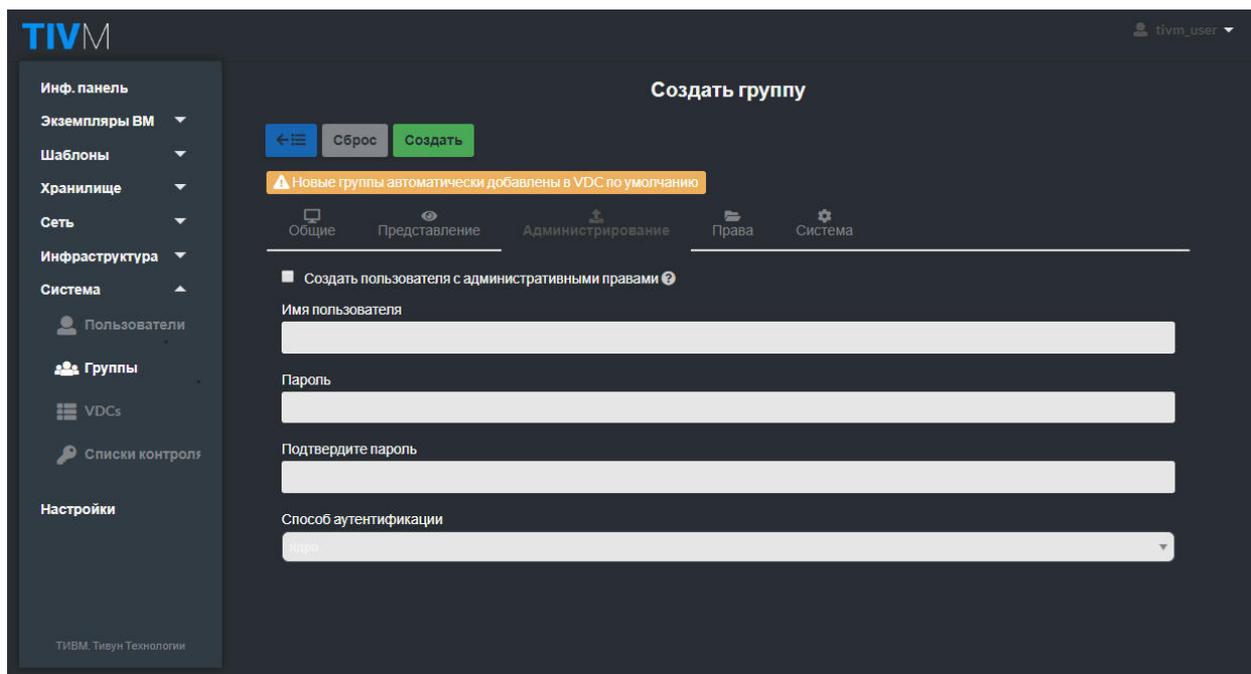


Рис. 3.3

Любой пользователь может быть назначен администратором группы с помощью команд `tivgroup addadmin` и `tivgroup deladmin`. Эти же операции доступны в web-интерфейсе (рис. 3.4).

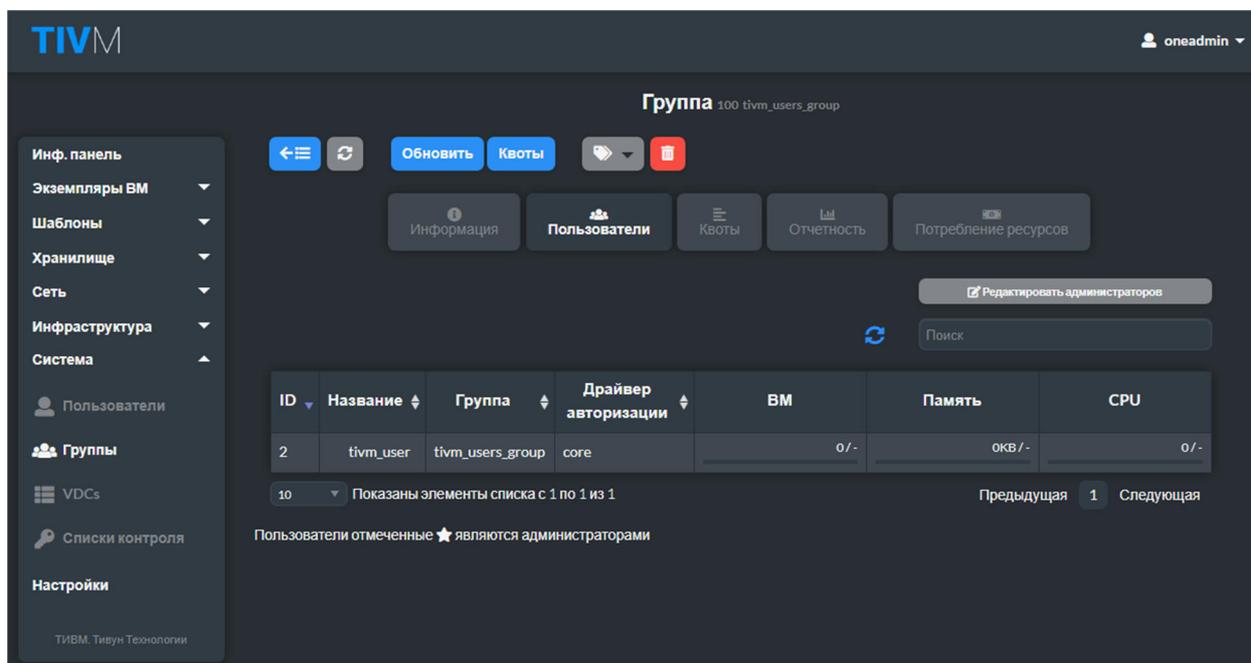


Рис. 3.4

Следующие виртуальные ресурсы могут быть определены доступными для группы пользователей:

- шаблоны VM;

- сервисные шаблоны;
- образы дисков;
- файлы и ядра.

Чтобы сделать виртуальный ресурс, принадлежащий `tivadmin`, доступным для пользователей новой группы, есть два варианта.

1. Изменить группу ресурса и выделить ему права `GROUP USE`. Это сделает ресурс доступным только пользователям в этой группе. Рекомендуемая практика назначения "золотых" ресурсов группам – сначала клонировать ресурс, а затем назначить его желаемой группе для потребления пользователями.

2. Оставить ресурс в группе `tivadmin` и дать ему права `OTHER USE`. Это сделает ресурс доступным для каждого пользователя `TIVM`.

Шаблоны ВМ и обслуживания видны пользователям группы, когда они хотят создать новую ВМ или службу. Образы (включая образы файлов), используемые этими шаблонами, не видны пользователям, но должны быть также доступны, в противном случае создание ВМ не будет завершено с сообщением об ошибке:

```
[TemplateInstantiate] User [6] : Not authorized to perform USE IMAGE [0].
```

Когда создается новая группа, администратор облака может определить, будут ли пользователи этого представления иметь доступ к ВМ и службам других пользователей в той же группе. Если эта опция отмечена, будет создано новое правило СУД для предоставления пользователям в этой группе доступа к `VMS` и службам в той же группе (рис. 3.5). Пользователи не смогут управлять этими ресурсами, но они будут включены в список представлений каждого ресурса.

С помощью команд `tivuser addgroup` и `tivuser delgroup` администратор может добавлять или удалять вторичные группы. Пользователи, назначенные более чем на одну группу, будут видеть ресурсы из всех своих групп. Например, пользователь в группах тестирования и производства будет видеть ВМ из обеих групп.

Группа с `chgrp` является основной группой, и ресурсы (образы, ВМ и т.д.), созданные пользователем, будут принадлежать этой основной группе. Пользователи могут изменить свою основную группу на любую из своих вторичных групп без вмешательства администратора, используя `chgrp` снова.

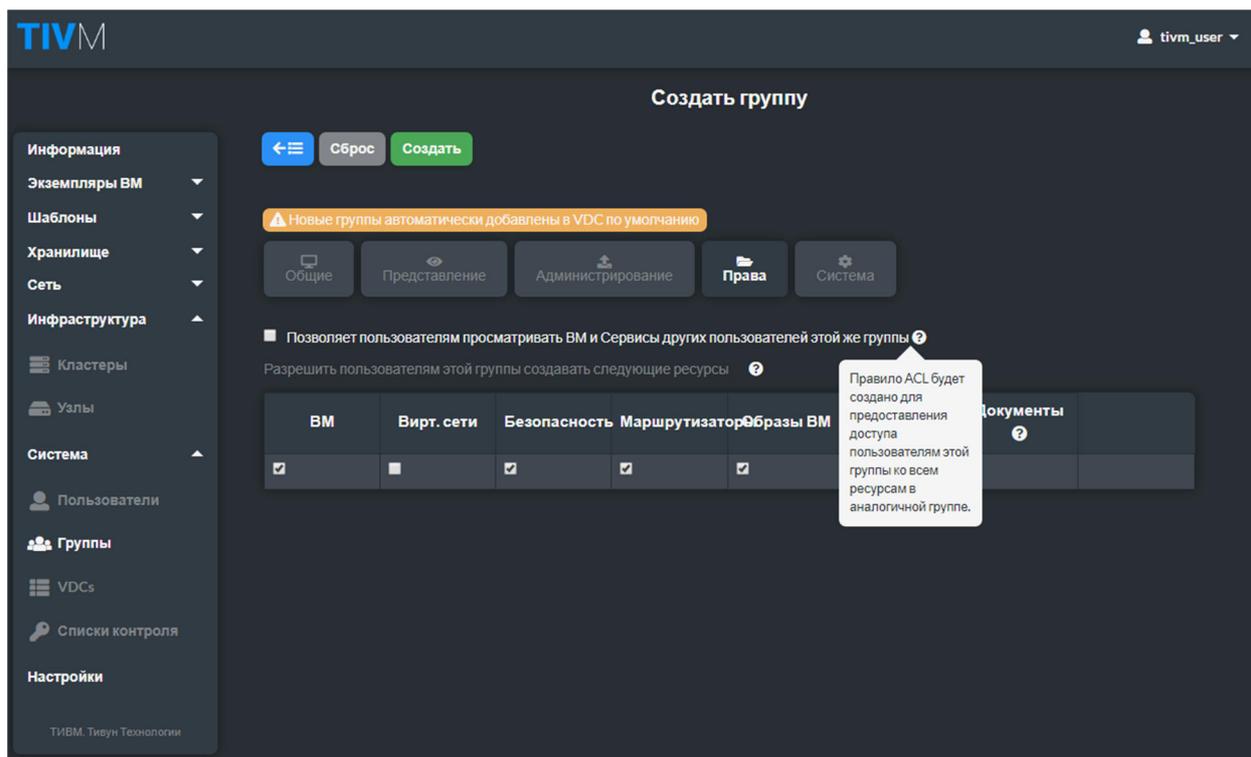


Рис. 3.5

Когда группа создана, можно определить конфигурацию для ее пользователей. Она включает следующие атрибуты:

- TIVM. Предоставление пользователям и администраторам групп доступа к определенным представлениям. Атрибуты конфигурации хранятся в шаблоне группы в атрибуте TIVM (табл. 3.2).

Таблица 3.2 – Атрибуты конфигурации группы, подчиненные атрибуту TIVM

Атрибут	Описание
DEFAULT_VIEW	Стандартный вид TIVM для обычных пользователей
VIEWS	Список доступных просмотров для обычных пользователей
GROUP_ADMIN_DEFAULT_VIEW	Представление TIVM по умолчанию для пользователей группы администраторов
GROUP_ADMIN_VIEWS	Список доступных просмотров для администраторов группы

Представления определяются разделенным запятыми списком имен групп. По умолчанию определены следующие представления: *groupadmin*,

cloud, admin, user, admin_vcenter, cloud_vcenter, groupadmin_vcenter.
 Пример:

```
TIVM = [
  DEFAULT_VIEW = "cloud",
  VIEWS        = "cloud",
  GROUP_ADMIN_DEFAULT_VIEW = "groupadmin",
  GROUP_ADMIN_VIEWS       = "groupadmin,cloud"
]
```

- TIV Manager Core. Задаёт атрибуты для управления некоторыми операциями. Атрибуты конфигурации хранятся в шаблоне группы в атрибуте TIVMANAGER (табл. 3.3).

Таблица 3.3 – Атрибуты конфигурации группы, подчиненные атрибуту TIVMANAGER

Атрибут	Описание
DEFAULT_IMAGE_PERSISTENT	Задаёт значение по умолчанию для атрибута PERSISTENT при создании образа (клонирование и сохранение на диске)
DEFAULT_IMAGE_PERSISTENT_NEW	Задаёт значение по умолчанию для атрибута PERSISTENT при создании образа (только новые образы).
API_LIST_ORDER	Устанавливает порядок (по идентификатору) элементов в вызовах API списка (например, список <code>tivvm list</code>). Значения: ASC (в порядке возрастания) или DESC (в порядке убывания)

Эти значения могут быть перезаписаны для каждого пользователя, поместив нужные значения в шаблон пользователя.

Если значения не установлены, используются значения по умолчанию, определённые в `tiv.conf`. Пример:

```
TIVMANAGER = [
  DEFAULT_IMAGE_PERSISTENT      = "YES",
  DEFAULT_IMAGE_PERSISTENT_NEW = "NO"
]
```

Все описанные функции доступны в web-интерфейсе TIVM (рис. 3.6).

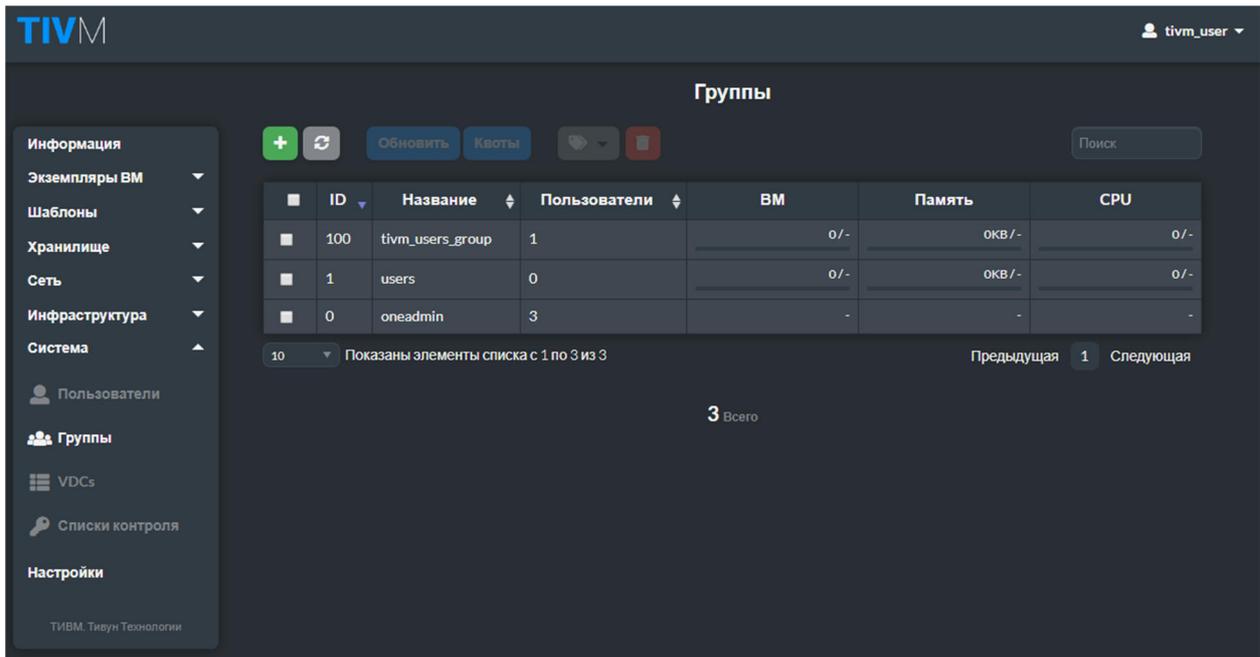


Рис. 3.6

Существует возможность фильтрации всех системных ресурсов по группе, к которой они относятся. В меню пользователя отображаются группы этого пользователя. Есть опция All для просмотра всех системных ресурсов. При фильтрации по группам можно изменить применяемую группу пользователя (рис. 3.7). Это позволяет более комфортно работать над проектами, изолируя ресурсы, принадлежащие одной группе, от других.

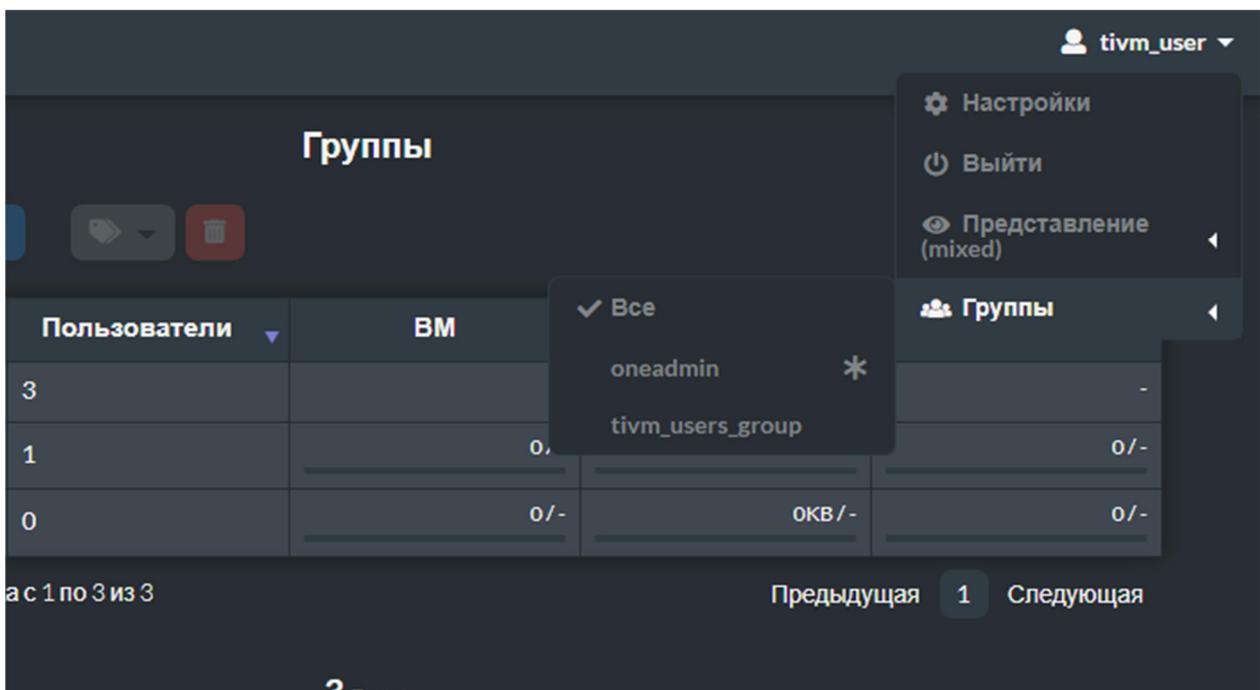


Рис. 3.7

3.4 Управление виртуальными дата-центрами

ВДЦ определяет назначение одной или нескольких групп для пула физических ресурсов. Этот пул физических ресурсов состоит из ресурсов одного или нескольких кластеров, которые могут принадлежать разным зонам или публичным внешним облакам для гибридных облачных вычислений.

При инсталляции TIVM в конфигурации создается ВДЦ по умолчанию (*default VDC*), который позволяет использовать все физические ресурсы.

Группа `users` принадлежит к этой группе ВДЦ, и любая новая группа автоматически добавляется в `default VDC`. Пользователь может изменить физические ресурсы ВДЦ, но их нельзя удалить.

Перед добавлением новой группы к определенному ВДЦ можно удалить ее из группы `default`, так как она позволяет использовать все физические ресурсы.

Для управления ВДЦ предназначена команда `tivvdc`.

Создание ВДЦ:

```
$ tivvdc list
ID NAME
0 default

$ tivvdc create "high-performance"
ID: 100
```

Новый ВДЦ имеет ID 100, чтобы отличать ВДЦ по умолчанию от заданных пользователем.

По умолчанию группа не принадлежит ни одному ВДЦ, поэтому пользователи не будут иметь права использовать какой-либо ресурс до тех пор, пока он не будет явно добавлен к ВДЦ.

Добавление группы к ВДЦ:

```
$ tivvdc addgroup <vdc_id> <group_id>
```

Физические ресурсы (узлы, виртуальные сети и хранилища данных) могут быть добавлены в ВДЦ. Внутри ВДЦ будут созданы правила СУД, которые позволяют группам ВДЦ использовать этот пул ресурсов.

Обычно необходимо добавить кластеры в ВДЦ. Например, кластер 7 из зоны 0:

```
$ tivvdc addcluster <vdc_id> 0 7
```

Также можно добавить отдельные узлы, виртуальные сети и хранилища данных:

```
$ tivvdc addhost <vdc_id> 0 3
$ tivvdc addvnet <vdc_id> 0 9
$ tivvdc adddatastore <vdc_id> 0 102
```

Специальный идентификатор ALL можно использовать для добавления всех кластеров/узлов/сети/данных из данной зоны:

```
$ tivvdc addcluser <group_id> 0 ALL
```

Для удаления физических ресурсов из ВДЦ предназначены обратные команды `tivvdc delcluster`, `tivvdc delhost`, `tivvdc delvnet`, `tivvdc deldatastore`.

При назначении физических ресурсов для ВДЦ пользователи в группах этих ВДЦ смогут использовать хранилища данных и виртуальные сети. Планировщик также развернет ВМ из этой группы в любой из узлов ВДЦ.

Создаваемые правила СУД можно посмотреть с помощью команды `tivacl list`. Эти правила добавляются автоматически, и их не следует редактировать вручную. Они будут удалены командами `tivvdc del*`. Стандартные разрешения для этих правил можно настроить на `tiv.conf`.

В рамках небольшой вычислительной инфраструктуры управление ВДЦ может не потребоваться для назначения физических ресурсов каждой группе. Но наличие независимого объекта ВДЦ позволяет ему иметь более одной группы, и в то же время группа может быть частью более одного ВДЦ. С практической точки зрения это означает, что при организации пользователей в группы, а физических ресурсов в ВДЦ можно легко назначать больше или меньше ресурсов этим группам. Администратор облачной инфраструктуры может добавить ресурсы в тот или иной ВДЦ, если их рабочая нагрузка увеличится, а затем удалить эти ресурсы через несколько дней.

ВДЦ может иметь более одного физического ресурса каждого типа (кластер, хосты виртуализации, виртуальные сети, хранилища данных), а физический ресурс может находиться в нескольких ВДЦ. Это означает, что вы можно создать ВДЦ, который включает в себя ресурсы, которые могут не входить в один и тот же физический кластер.

Например, ВДЦ, называемый «высокопроизводительным», может содержать хосты из двух несовместимых кластеров – "kvm-ceph" и "kvm-qcow2". Эти хосты могут быть частью одного и того же ВДЦ, но с точки зрения развертывания важным фактором является их кластер. Планировщик

будет определять цель развертывания на основе каждого кластера хоста, и это гарантирует, что ВМ всегда развернуты на совместимом хосте.

Поскольку ВДЦ может содержать отдельные хосты, виртуальные сети и хранилища данных, можно использовать ВДЦ для разделения кластера на «подкластеры», которые содержат несколько хостов.

Например, существует большой кластер "kvm-серв". ВДЦ с одним или двумя хостами может быть создан, чтобы изолировать небольшую часть кластера. В этом случае следует добавить необходимые хранилища данных и виртуальные сети в ВДЦ, иначе пользователи не смогут создавать экземпляры шаблонов ВМ.

Пусть имеются две группы с одинаковой рабочей нагрузкой. Если необходимо изолировать пользователей и виртуальные ресурсы этих групп, то обе группы могут быть добавлены к одному и тому же ВДЦ. Аналогично, физический ресурс (например, хост виртуализации) может быть частью двух разных ВДЦ. Группы будут совместно использовать физические ресурсы, не зная об этом. Если физические ресурсы не предназначены исключительно для данной группы, можно установить квоты использования.

Описанные функции управления ВДЦ доступны на соответствующей странице web-интерфейса TIVM (рис. 3.8).

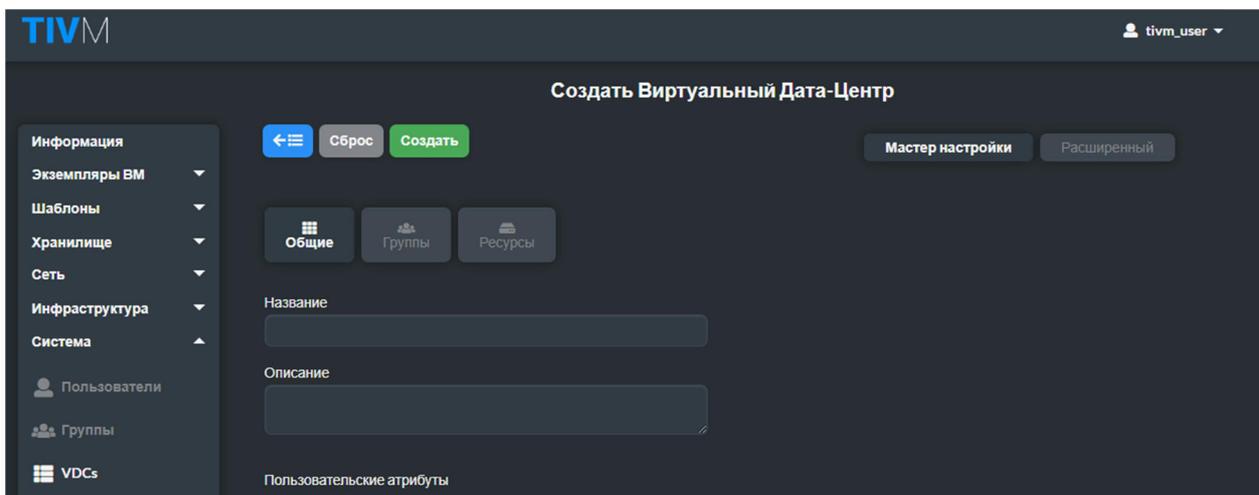


Рис. 3.8

3.5 Управление разрешениями на операции

Большинство ресурсов TIVM имеют соответствующие разрешения на выполнение операций с ними для владельца, пользователей в своей группе и прочих пользователей. Для каждой из этих групп можно установить три разрешения, соответствующие категориям операций: **USE** (использовать), **MANAGE** (управлять) и **ADMIN** (администрировать). Логика этих разрешений похожа на разрешения для файловой системы UNIX.

Ресурсы, операции с которыми связаны с разрешениями: это шаблоны, ВМ, образы и виртуальные сети. Исключение составляют пользователи, группы и узлы.

Установленные разрешения отображаются в составе шаблона вычислительного ресурса. Например, разрешения на операции с шаблоном ВМ:

```
tivtemplate show 0
TEMPLATE 0 INFORMATION
ID           : 0
NAME        : vm-example
USER        : tivuser1
GROUP       : users
REGISTER TIME : 01/13 05:40:28

PERMISSIONS
OWNER       : um-
GROUP      : u--
OTHER      : ---

[...]
```

В этом примере пользователь `tivuser1` имеет разрешения на операции с шаблоном, имеющим идентификатор `0`, из категорий `USE` и `MANAGE`. Пользователи в группе `users` имеют права `USE`, а пользователи, не являющиеся владельцами или входящие в группу `users`, не имеют никаких разрешений на операции с этим шаблоном.

Категории `USE`, `MANAGE` и `ADMIN` включают следующие операции.

USE: Операции, которые не изменяют ресурс, такие как его размещение или использование (например, с помощью образа или виртуальной сети). Обычно пользователь предоставляет разрешения `USE` на доступ к ресурсам для других пользователей группы или прочих пользователей.

MANAGE: Операции, которые изменяют ресурс, такие как остановка ВМ, изменение постоянного атрибута образа или снятие аренды с сети. Обычно пользователь предоставляет разрешения `MANAGE` для пользователей, которые будут управлять его ресурсами.

ADMIN: Специальные операции, которые обычно выполняются администраторами, такие как обновление данных хоста или удаление группы пользователей. Обычно разрешения `ADMIN` предоставляются пользователям, у которых есть роль администратора.

Разрешения могут быть установлены с помощью команды `chmod`. Эта команда принимает октет в качестве параметра, следуя восьмеричной нотации команды Unix `chmod`. Октет должен быть трехразрядным числом. Каждый разряд со значением от 0 до 7 представляет права для владельца,

группы и прочих пользователей соответственно. Права представлены следующими значениями:

- бит USE добавляет 4 к его общей сумме (в двоичном представлении – 100);

- бит MANAGE добавляет 2 к общей сумме (в двоичном представлении – 010);

- бит ADMIN добавляет 1 к его общему значению (в двоичном представлении – 001).

Примеры:

```
tivtemplate show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---
tivtemplate chmod 0 664 -v
VMTEMPLATE 0: Permissions changed
tivtemplate show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : um-
OTHER          : u--
tivtemplate chmod 0 644 -v
VMTEMPLATE 0: Permissions changed
tivtemplate show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : u--
tivtemplate chmod 0 607 -v
VMTEMPLATE 0: Permissions changed
tivtemplate show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : uma
```

Разрешения по умолчанию, предоставленные вновь созданным ресурсам, могут быть установлены:

- глобально через атрибут DEFAULT_UMASK в *tiv.conf*;

- индивидуально для каждого пользователя с помощью команды *tivuser umask*.

Атрибуты маски работают аналогично команде Unix *umask*. Значение – трехразрядное число. Каждый разряд представляет

маску, которая отключает права доступа для владельца, группы и прочих пользователей соответственно. В табл. 3.4 приведены примеры.

Таблица 3.4 – Примеры масок, задающие права доступа к ресурсам

<i>umask</i>	Разрешения (в 8-ричное коде)	Разрешения
177	600	um- --- ---
137	640	um- u-- ---
113	664	um- um- u--

Web-интерфейс TIVM предлагает удобный способ управления разрешениями. Это можно сделать, выбрав ресурсы из представления (например, представление шаблонов) и нажав на кнопку "Обновить свойства". Диалоговое окно обновления позволяет пользователю установить права доступа к ресурсу (рис. 3.9).

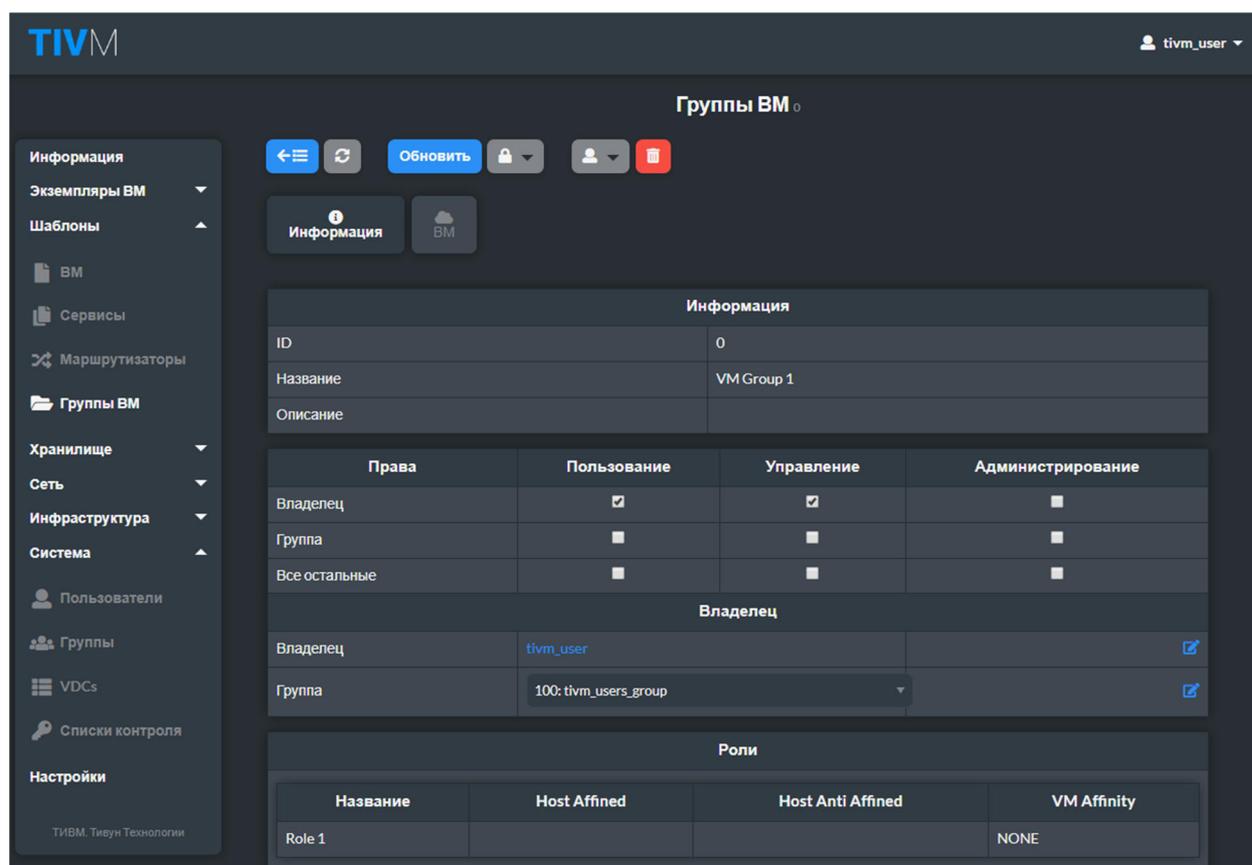


Рис. 3.9

TIVM может *блокировать операции* с ресурсом для предотвращения непреднамеренных операций, например, для предотвращения случайного удаления VM, путем указания уровня доступа, необходимого для операции:

- USE: блокирует все возможные действия;
- MANAGE: блокирует операции управления и администрирования;
- ADMIN: блокирует действия администратора.

Следующие ресурсы могут быть заблокированы:

- VM;
- виртуальные сети;
- образы дисков;
- шаблоны;
- документы;
- виртуальные маршрутизаторы.

Пример:

```
tivimage lock 2
tivimage delete 2
[tiv.image.delete] User [4] : Not authorized to perform MANAGE IMAGE
[2].
```

```
tivimage unlock 2
```

Пользователь *tivadmin* не может быть заблокирован и может выполнить операцию в любом случае.

Другим средством управления правами доступа к ресурсам, наряду с разрешениями на операции, служат *списки управления доступом СУД (Access Control List)*. Система авторизации СУД позволяет точно настроить разрешенные операции для любого пользователя или группы пользователей. Каждая операция генерирует запрос авторизации, который проверяется по зарегистрированному набору правил СУД. Затем ядро может предоставить разрешение или отклонить запрос.

Это позволяет администраторам настраивать роли пользователей в соответствии с потребностями их инфраструктуры. Например, используя правила СУД, можно создать группу пользователей, которые могут видеть и использовать существующие виртуальные ресурсы, но не создавать никаких новых. Или можно предоставить разрешения пользователю для управления виртуальными сетями для некоторых из существующих групп, но не для выполнения каких-либо других операций в облачной среде.

Следует отметить, что пользовательские правила СУД – это продвинутый механизм управления доступом к вычислительным ресурсам. В большинстве случаев достаточно разрешений на операции и правил СУД, создаваемых автоматически при создании группы и при добавлении физических ресурсов в ВДЦ.

Пример правила СУД:

```
#5 IMAGE+TEMPLATE/@103 USE+MANAGE #0
```

Данное правило предоставляет пользователю с идентификатором 5 право выполнять операции USE и MANAGE над всеми образами и шаблонами в группе с идентификатором 103. Правило состоит из четырех компонентов, разделенных пробелами:

- идентификатор пользователя (в данном примере – 5);
- вычислительные ресурсы – список типов ресурсов, записанных через символ "+" (в данном примере – образы и шаблоны), слэш ("/") и идентификатор группы (в данном примере – 103);
- права – список операций, разделенных символом "+" (в данном примере – использование и управление);
- зона – идентификатор зоны, к которой применяется правило (в данном примере – 0). Этот компонент является необязательным и может быть опущен, если TIVM не настроен на работу в федерации.

Определение идентификатора пользователя, как правило, записывается так:

- #<id> – для индивидуальных идентификаторов;
- @<id> – для идентификатора группы;
- * – для всех.

Определение идентификатора ресурса имеет тот же синтаксис, что и для пользователей, но с добавлением:

- %<id> – для идентификаторов кластера.

Следующее правило позволяет всем пользователям в группе 105 создавать новые виртуальные ресурсы.

```
@105 VM+NET+IMAGE+TEMPLATE/* CREATE
```

Следующее позволяет всем пользователям в группе 106 использовать виртуальную сеть 47. Это означает, что они могут создавать экземпляры шаблонов ВМ, которые используют эту сеть.

```
@ 106 NET / # 47 USE
```

Следующее правило позволяет пользователям в группе 106 развертывать виртуальные машины в узлах, назначенных кластеру 100.

```
@106 HOST/%100 MANAGE
```

Для управления правилами СУД служит группа команд `tivacl`.

Команда `tivacl list` возвращает список правил:

```
tivacl list
ID      USER RES_VHNIUTGDCOZSvRMA  RID OPE_UMAC  ZONE
0       @1    V--I-T---O-S----      *   ---c      *
1       *    -----Z-----      *   u---      *
2       *    -----MA          *   u---      *
3       @1    -H-----          *   -m--      #0
4       @1    --N---D-----      *   u---      #0
5       @106  ---I-----      #31  u---      #0
```

Показанные правила соответствуют следующим:

```
@1      VM+IMAGE+TEMPLATE+DOCUMENT+SECGROUP/*  CREATE  *
*       ZONE/*                               USE      *
*       MARKETPLACE+MARKETPLACEAPP/*     USE      *
@1      HOST/*                               MANAGE   #0
@1      NET+DATASTORE/*                     USE      #0
@106    IMAGE/#31                           USE      #0
```

Первые пять были созданы по умолчанию при установке TIVM, а последнее – с помощью `tivacl`:

```
tivacl create "@106 IMAGE/#31 USE"
ID: 5
```

В списке, выводимом `tivacl list`, колонка "ID" представляет идентификатор правила. Он необходим для удаления правил с помощью `tivacl delete`.

Следующий столбец – "USER" – может быть индивидуальным (#) или групповым (@) идентификатором пользователя. Либо он соответствует всем пользователям (*).

В столбце ресурсов указаны обозначения типов ресурсов, на которые распространяется правило:

- V – VM;
- H – хост виртуализации;
- N – виртуальная сеть;
- I – образ диска;
- U – пользователь;
- T – шаблон;
- G – группа;
- D – хранилище данных;
- C – кластер;
- D – документ;
- Z – зона;
- S – группа безопасности;
- v – ВДЦ;

- R – виртуальный маршрутизатор.

В столбце "RID" указан идентификатор ресурса. Это может быть отдельный объект (#), группа (@), идентификатор кластера (%) или все (*) объекты.

В следующем столбце перечислены обозначения разрешенных типов операций:

- U – использование;
- M – управление;
- A – администрирование;
- C – создание.

В столбце "Зона" указана зона, в которой применяется правило. Это может быть отдельный ID зоны (#) или вся (*) зона.

Страница управления СУД в web-интерфейсе TIVM показана на рис. 3.10. Она открывается через пункт меню Система/Списки контроля.

ID	Применено к	Затрагиваемые ресурсы	№ ресурса / Принадлежит	Разрешенные действия	Зона
9	Группа tivm_users_group	Вирт. машины, Образы VM, Шаблоны VM, Документы, Безопасность, Маршрутизаторы	Все	create	Все
8	Группа tivm_users_group	Хранилища	Все	use	ТИВМ
7	Группа tivm_users_group	Вирт. сети	Все	use	ТИВМ
6	Группа tivm_users_group	Узлы	Все	manage	ТИВМ
5	Группа users	Хранилища	Все	use	ТИВМ
4	Группа users	Вирт. сети	Все	use	ТИВМ
3	Группа users	Узлы	Все	manage	ТИВМ
2	Все	Магазин приложений, Приложения из магазина приложений	Все	use	Все
1	Все	Зоны	Все	use	Все
0	Группа users	Вирт. машины, Образы VM, Шаблоны VM, Документы, Безопасность, Группы VM	Все	create	Все

Рис. 3.10

Это представление позволяет понять назначение каждого правила СУД. Для создания СУД следует нажать кнопку New (плюс) вверху. Появится диалоговое окно, показанное на рис. 3.11.

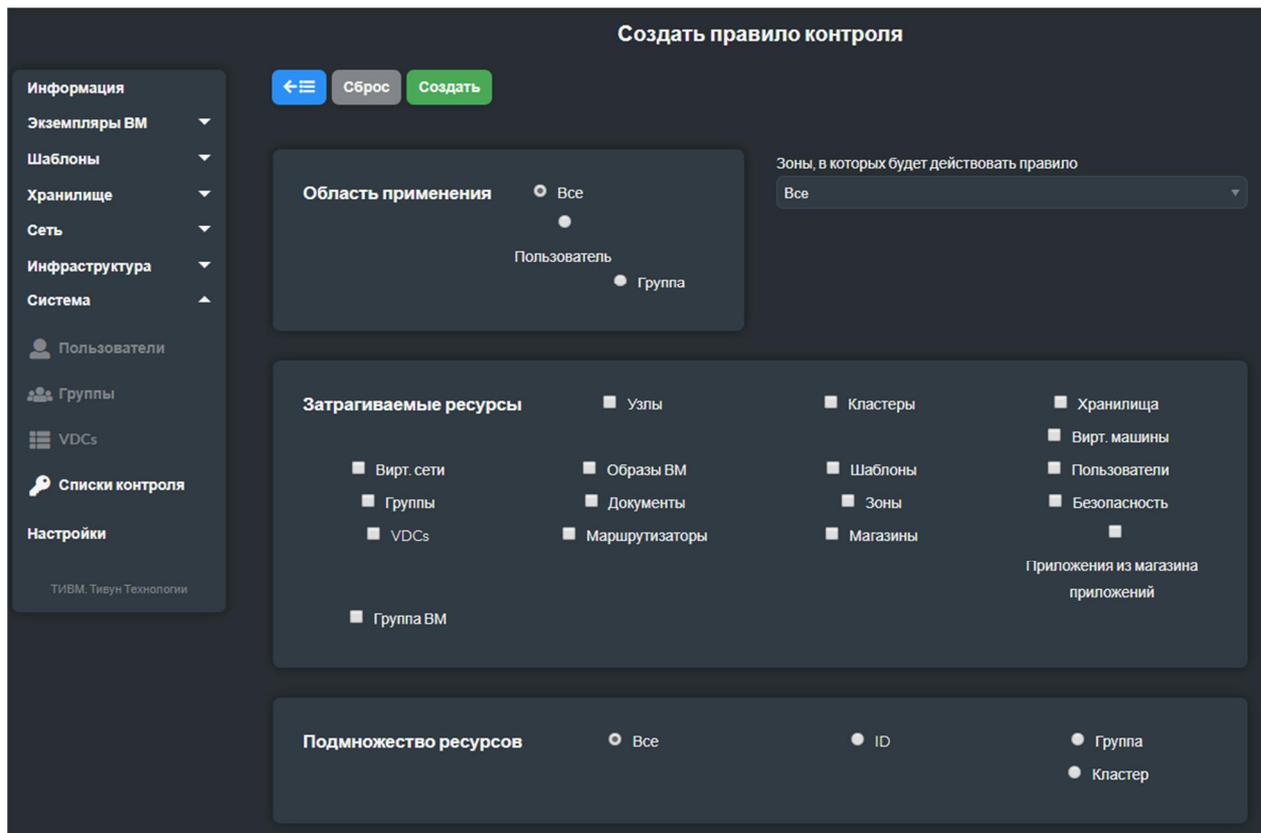


Рис. 3.11

В диалоговом окне создания правила указываются ресурсы, на которые оно распространяется, и разрешения, которые на них выдаются.

Для внутренней авторизации в TIVM существует неявное правило: пользователь *tivadmin* или пользователи в группе *tivadmin* имеют право выполнять любые операции.

Если ресурс относится к одному из типов – VM, виртуальная сеть, образ, шаблон, документ, то проверяются разрешения на операции с ним. Например, разрешения на операции с образом выводятся командой `tivimage show`:

```
tivimage show 2
IMAGE 2 INFORMATION
ID           : 2
[...]

PERMISSIONS
OWNER       : um-
GROUP      : u--
OTHER      : ---
```

Как видно из примера, владелец образа имеет разрешения на операции категорий `USE` и `MANAGE`.

Если ни одно из указанных выше условий не выполняется, то набор правил СУД повторяется до тех пор, пока одно из правил не разрешит операцию.

Важной концепцией набора СУД является то, что каждое правило добавляет новые разрешения, и они не могут ограничивать существующие: если какое-либо правило дает разрешение, операция разрешается.

Например, пользователь с идентификатором 7 находится в группе @108 со следующим существующим правилом:

```
@108 IMAGE/#45 USE+MANAGE
```

Тогда следующее правило не будет иметь никакого эффекта:

```
#7 IMAGE/#45 USE
```

Существует специальный подтип виртуальной сети: резервирование. Для этих виртуальных сетей система СУД делает следующие исключения:

- правила СУД, которые применяются ко всем (*), игнорируются;
- правила СУД, которые применяются к кластеру (%), игнорируются.

Применяются другие правила СУД: индивидуальный (#) и групповой (@). Разрешения, относящиеся к виртуальной сети, также применяются, как обычно.

3.6 Управление квотами

Система квотирования отслеживает использование пользователями и группами вычислительных ресурсов и позволяет системному администратору устанавливать ограничения на использование этих ресурсов. Можно установить лимиты квот для:

- пользователей, чтобы индивидуально ограничить потребление ресурсов, сделанное данным пользователем;
- групп, чтобы ограничить общее потребление, осуществляемое всеми пользователями данной группы. Это может представлять особый интерес при работе с зонами и ВДЦ.

Система квотирования позволяет отслеживать и ограничивать использование:

- хранилищ данных, чтобы контролировать объем дисковой памяти, выделенный каждому пользователю или группе для каждого хранилища данных;
- вычислительных ресурсов (оперативной памяти, процессорных ядер, экземпляров VM);

- сетевых ресурсов, чтобы ограничить количество IP-адресов, которые пользователь или группа может получить от данной сети. Это особенно актуально для сетей с публичными IP-адресами, которые обычно являются ограниченным ресурсом;

- образов, чтобы ограничить количество экземпляров ВМ для данного пользователя или группы, задействующих данный образ. Этот вид квотирования актуален, если образ представляет лимитируемые ресурсы (например, лицензии на ПО).

Для хранилищ данных, реализуемых на основе технологии Ceph, может быть ограничен только общий объем хранилища. Такие технологии используют одно и то же пространство для системных хранилищ и хранилищ образов, поэтому TIVM не может знать, какое пространство используется в каждом случае.

Квоты на использование вычислительных ресурсов задаются в их шаблонах (в виде простого текста или в формате XML). В табл. 3.5–3.8 приведены атрибуты шаблонов, необходимые для установки каждой квоты.

Таблица 3.5 – Квоты для хранилищ данных. Имя атрибута: DATASTORE

Атрибут DATASTORE	Описание
ID	ID хранилища данных для установки квоты
SIZE	Максимальный размер в мегабайтах, который можно использовать в хранилище данных
IMAGE	Максимальное количество изображений, которое можно создать в хранилище данных

Таблица 3.6 – Квоты для вычислительных ресурсов. Имя атрибута: VM

Атрибут VM	Описание
VMS	Максимальное количество ВМ, которое может быть создано
MEMORY	Максимальный объем памяти в мегабайтах, который может быть запрошен виртуальными машинами пользователей или группы
CPU	Максимальная емкость процессоров, которую могут запросить виртуальные машины пользователя или группы

Атрибут VM	Описание
RUNNING VMS	Максимальное количество VM которые могут быть запущены
RUNNING MEMORY	Максимальный объем памяти в мегабайтах, который может быть запущен с помощью VM пользователя или группы.
RUNNING CPU	Максимальная емкость процессоров, которая может быть запущена виртуальными машинами пользователя или группы
SYSTEM_DISK_SIZE	Максимальный размер (в мегабайтах) системных дисков, который может быть запрошен VM пользователя или группы

Таблица 3.7 – Квоты для сетевых ресурсов. Имя атрибута: NETWORK

Атрибут NETWORK	Описание
ID	ID сети, для которой устанавливается квота
LEASES	Максимальное количество IP-адресов, которые можно арендовать у сети

Таблица 3.8 – Квоты на образы. Имя атрибута: IMAGE

Атрибут IMAGE	Описание
ID	ID изображения для установки квоты.
RVMS	Максимальное количество виртуальных машин, которые могут одновременно использовать этот образ.

Текущие квоты будут увеличены или уменьшены в зависимости от состояния VM. Состояния, в которых машина считается работающей, являются: ACTIVE, HOLD, PENDING и CLONING.

Для каждой квоты есть два *особых значения*:

- 1 – квота по умолчанию;
- 2 – отсутствие ограничений.

Каждая квота имеет счетчик использования, связанный с именем <QUOTA_NAME>_USED. Например, MEMORY_USED означает общую память, используемую VM пользователей и групп, и связанную с ней квоту MEMORY.

Ниже приведен пример шаблона с квотами для пользователя. Они ограничивают:

- общее использование хранилища данных с идентификатором 1: до 20 Гб (для неограниченного количества изображений);
- количество ВМ, которые могут быть созданы: до 4 с максимальной памятью до 2 Гб и 5 процессорами;
- количество IP-адресов, которые могут быть получены из сети: до 4;
- количество ВМ, которые могут одновременно использовать образ с идентификатором 1: до 3.

```
DATASTORE=[
  ID="1",
  IMAGES="-2",
  SIZE="20480"
]

VM=[
  CPU="5",
  MEMORY="2048",
  VMS="4",
  SYSTEM_DISK_SIZE="-1"
]

NETWORK=[
  ID="1",
  LEASES="4"
]

IMAGE=[
  ID="1",
  RVMS="3"
]

IMAGE=[
  ID="2",
  RVMS="-2"
]
```

Всякий раз, когда используется сеть, образ, хранилище данных или ВМ, создаются соответствующие счетчики квот для пользователя с неограниченным значением. Это позволяет отслеживать использование ресурсов каждым пользователем или группой, даже если квоты фактически не установлены.

Квоты для пользователей и групп можно настроить с помощью CLI или web-интерфейс TIVM. Для определения квот *нужны разрешения*:

- MANAGE для установки квоты пользователя;
- ADMIN для установки квоты группы.

Таким образом, по умолчанию только администратор системы может устанавливать квоты для групп. При наличии администратора группы он может управлять квотами для пользователей в своей группе (распределяя ресурсы по мере необходимости). Эту логику можно изменить, установив соответствующие правила СУД.

Для определения квоты пользователя служит команда `tivuser quota`:

```
tivuser quota user_name
```

По этой команде откроется сессия текстового редактора для изменения шаблона квоты для пользователя `user_name`. В шаблон можно добавить столько квот ресурсов, сколько необходимо, даже если они не были автоматически инициализированы.

Аналогичным образом, можно установить квоты для группы:

```
tivgroup quota group_name
```

Команды `tivuser batchquota` и `tivgroup batchquota` позволяют устанавливать одинаковые квоты для нескольких пользователей или групп соответственно:

```
tivuser batchquota userA, userB, 35
tivgroup batchquota 100..104
```

Аргументами этих команд служат имена или идентификторы пользователей или групп (следующие через запятую), а также интервалы идентификаторов (через "..").

Для установки квот в web-интерфейсе TIVM предназначены вкладки "Квоты" на страницах редактирования описания пользователя или группы (рис. 3.12, 3.13). Эти страницы открываются через пункты меню Система/Пользователи и Система/Группы.

Если квоты для пользователя или группы не определены, то TIVM использует для них шаблоны квот по умолчанию. Эти шаблоны применимы ко всем пользователям и группам, если только у них нет индивидуального набора ограничений.

Для изменения шаблона квот по умолчанию предназначены команды:

- `tivuser defaultquota;`
- `tivgroup defaultquota.`

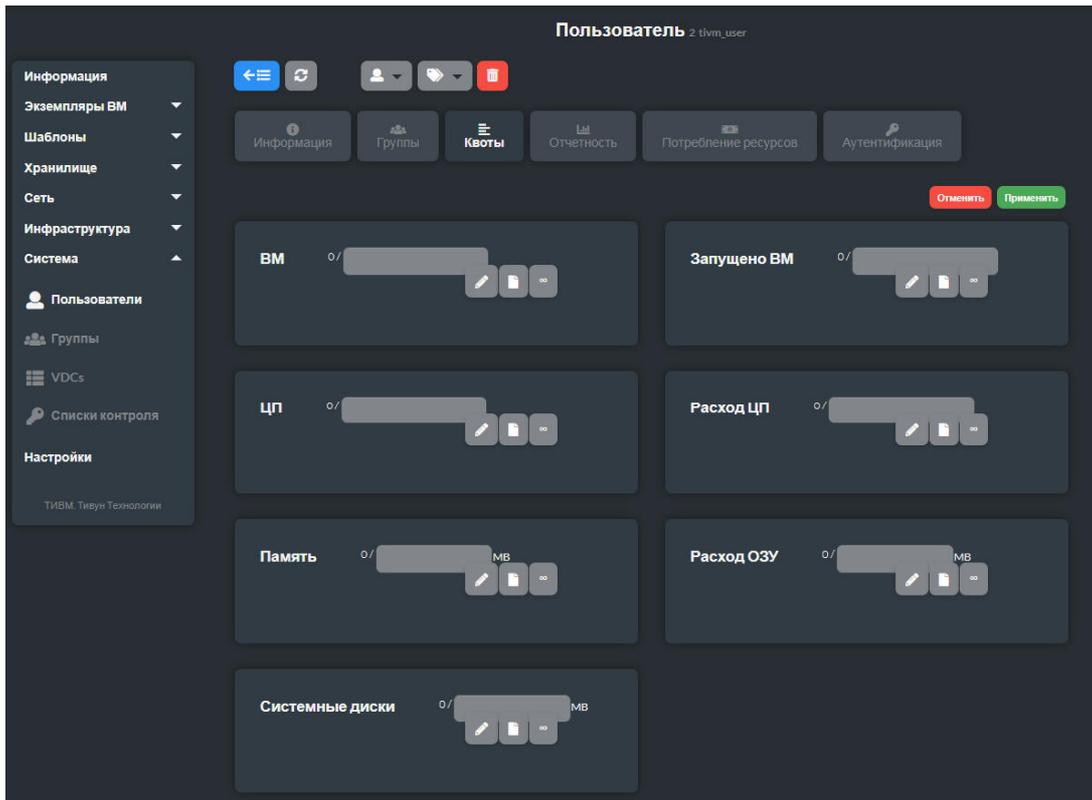


Рис. 3.12

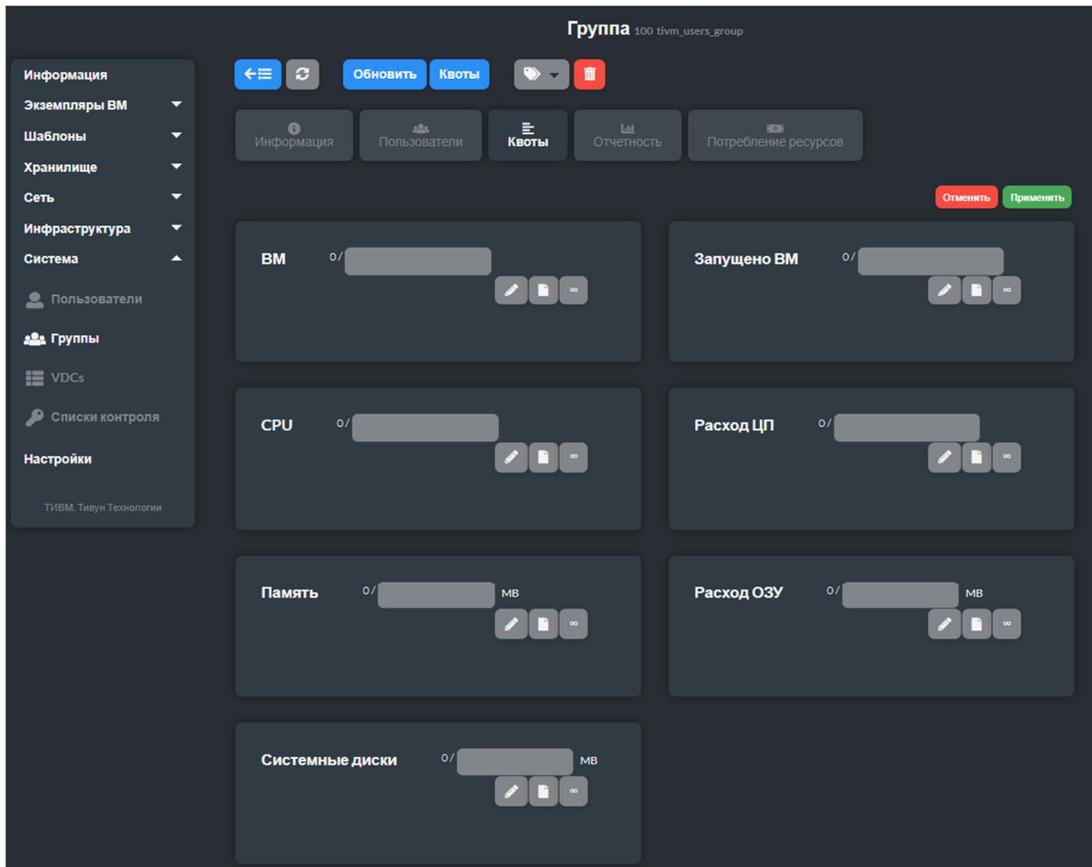


Рис. 3.13

Информация о квотах и фактическом использовании ресурсов пользователем или группой возвращается командами `tivuser show` и `tivgroup show`.

Пример для пользователя `userA`:

```
tivuser show userA
USER 2 INFORMATION
ID           : 2
NAME        : userA
GROUP       : groupA
PASSWORD    : a9993e364706816aba3e25717850c26c9cd0d89d
AUTH_DRIVER : core
ENABLED     : Yes

USER TEMPLATE

VMS USAGE & QUOTAS

          VMS                MEMORY                CPU
SYSTEM_DISK_SIZE
  1 /      4          1M /      -          2.00 /      -          0M /
-

VMS USAGE & QUOTAS - RUNNING

          RUNNING VMS          RUNNING MEMORY          RUNNING CPU
  1 /      -          1M /      2M          2.00 /      -

DATASTORE USAGE & QUOTAS

NETWORK USAGE & QUOTAS

IMAGE USAGE & QUOTAS
```

Пример для группы:

```
tivgroup show groupA
GROUP 100 INFORMATION
ID           : 100
NAME        : groupA

USERS
ID
2
3

VMS USAGE & QUOTAS

          VMS                MEMORY                CPU
SYSTEM_DISK_SIZE
```

```

1 /      4      1M /      -      2.00 /      -      0M /
-

VMS USAGE & QUOTAS - RUNNING

      RUNNING VMS      RUNNING MEMORY      RUNNING CPU
1 /      -      1M /      2M      2.00 /      -

DATASTORE USAGE & QUOTAS

NETWORK USAGE & QUOTAS

IMAGE USAGE & QUOTAS

```

Эта информация также доступна в web-интерфейсе TIVM, как часть информации о пользователе или группе.

3.7 Формирование отчетов об использовании вычислительных ресурсов

TIVM включает набор инструментов для учета потребления виртуальных вычислительных ресурсов и построения соответствующих отчетов. Отчеты формируются на основе данных, возвращаемых гипервизорами и модулем мониторинга.

Для получения данных о ВМ служит команда `tivacct`:

```
tivacct [options]
```

Ее параметры (секция `[options]`):

- s, --start *time* – дата и время начала отчетного периода;
 - e, --end *time* – дата и время окончания отчетного периода;
 - u, --userfilter *user* – имя или идентификатор пользователя (фильтр для отбора данных);
 - g, --group *group* – имя или идентификатор группы (фильтр для отбора данных);
 - h, --host *host* – имя или идентификатор хоста (фильтр для отбора данных);
 - xpath *expression* – выражение Xpath для отбора результатов.
- Например, `--xpath 'HISTORY[ETIME>0]'`;
- x, --xml – представить информацию в XML-формате;
 - j, --json – представить информацию в JSON-формате;
 - split – разделить выходные данные по ВМ;
 - v, --verbose – подробный режим;
 - describe – выводить описания столбцов;
 - l, --list *col1, col2, ...* – выводить только столбцы с указанными номерами (*col1, col2, ...*);

--csv – выводить информацию в CSV-формате;
--user *name* – имя пользователя для авторизации;
--password *password* – пароль для авторизации;
--endpoint *endpoint* – адрес сервиса TIVM, обрабатывающего запрос.

Дата и время в параметрах *-s* и *-e* указываются в формате `month/day/year hour:minute:second`. Компоненты времени (секунды, минуты, часы) могут быть опущены.

Команда `tivacct` представляет данных, основанные на записях истории ЖЦ ВМ. Эти записи создаются при событиях запуска, оставки, приостановки, миграции ВМ. Изменение размера диска и включение/отключение диска или сетевого адаптера также создаст новую запись.

Каждая запись содержит полную информацию о ВМ, включая информацию о ее мониторинге. По умолчанию сообщается только о потреблении сети.

При фильтрации данных с помощью параметров *-s* и *-e*, учитываются все записи истории, которые были активны в течение этого интервала времени. Однако они могут начинаться или заканчиваться за его пределами.

Например, пусть ВМ работала с 1 мая по 1 июня. Пример запроса данных за этот период и полученного результата:

```
tivacct -s 05/01 -e 06/01
Showing active history records from 2020-05-01 00:00:00 +0300 to 2030-06-02
00:00:00 +0300

# User 0

  VID HOSTNAME ACTION   REAS START_TIME      END_TIME      MEMORY CPU NETRX
NETTX  DISK
  28 host01   terminate user 05/27 16:40:47 05/27 17:09:20 1024M  0.1 0K
0K     10.4G
  29 host02   none      none 05/27 17:09:28 -          256M  1   2.4M
1.3K   10G
```

Результат представляет полную историю ВМ и общее потребление сети. В ней не будет отражено потребление, произведенное только в течение мая месяца.

Необходимо учитывать, что активные записи истории ВМ, не имеющие времени завершения (столбец `END_TIME` содержит прочерк), обновляют свою информацию мониторинга каждый раз, когда осуществляется мониторинг ВМ. После того, как ВМ выключена, перенесена или остановлена, устанавливается `END_TIME`, и сохраненная информация мониторинга замораживается. Окончательные значения отражают общее количество для накопительных атрибутов, таких как `NETRX/NETTX`.

Пример получения всей доступной учетной информации.

```
tivacct
# User 0

  VID HOSTNAME      ACTION      REAS      START_TIME      END_TIME
MEMORY CPU  NETRX  NETTX  DISK
-----
  13 host01          nic-attach  user 05/17 17:10:57 05/17 17:12:48
256M 0.1  19.2K 15.4K   8G
  13 host01          nic-detach  user 05/17 17:12:48 05/17 17:13:48
256M 0.1  36.9K 25K     8G
  13 host01          nic-attach  user 05/17 17:13:48 05/17 17:14:54
256M 0.1  51.2K 36.4K   8G
  13 host01          nic-detach  user 05/17 17:14:54 05/17 17:17:19
256M 0.1  79.8K 61.7K   8G
  13 host01          nic-attach  user 05/17 17:17:19 05/17 17:17:27
256M 0.1  79.8K 61.7K   8G
  13 host01          terminate-hard user 05/17 17:17:27 05/17 17:37:52
256M 0.1 124.6K 85.9K   8G
  14 host02          nic-attach  user 05/17 17:38:16 05/17 17:40:00
256M 0.1  16.5K 13.2K   8G
  14 host02          poweroff    user 05/17 17:40:00 05/17 17:53:40
256M 0.1  38.3K 18.8K   8G
  14 host02          terminate-hard user 05/17 17:55:55 05/18 14:54:19
256M 0.1    1M 27.3K   8G
```

Значения столбцов представлены в табл. 3.9.

Таблица 3.9 – Столбцы таблицы учетной информации, возвращаемой командой `tivacct`

Столбец	Значение
VID	Идентификатор VM
HOSTNAME	Имя хоста
ACTION	Действие VM, которое создало новую запись истории
REASON	Причина изменения состояния VM: - none: виртуальная машина работает; - error: действие VM закончилось ошибкой; - user: действие VM, запущенное пользователем.
START_TIME	Время запуска
END_TIME	Время окончания
MEMORY	Назначена память. Это запрошенная память, а не контролируемое потребление памяти

Столбец	Значение
CPU	Количество процессоров. Это запрошенное количество ресурсов центрального процессора узла, а не контролируемое использование процессора
NETRX	Данные, полученные из сети
NETTX	Данные отправлены в сеть

Для получения данных, относящихся к пользователю, служит параметр `-u`. В примере ниже результаты разделяются по ВМ:

```
tivacct -u 0 --split
# User 0

  VID HOSTNAME      ACTION      REAS      START_TIME      END_TIME
MEMORY CPU NETRX NETTX  DISK
  12 host01         none       user 05/09 19:20:42 05/09 19:35:23
1024M  1  29.8M 638.8K   0K

  VID HOSTNAME      ACTION      REAS      START_TIME      END_TIME
MEMORY CPU NETRX NETTX  DISK
  13 host01         nic-attach user 05/17 17:10:57 05/17 17:12:48
256M  0.1  19.2K 15.4K   8G
  13 host01         nic-detach user 05/17 17:12:48 05/17 17:13:48
256M  0.1  36.9K 25K     8G
  13 host01         nic-attach user 05/17 17:13:48 05/17 17:14:54
256M  0.1  51.2K 36.4K   8G
  13 host01         nic-detach user 05/17 17:14:54 05/17 17:17:19
256M  0.1  79.8K 61.7K   8G
  13 host01         nic-attach user 05/17 17:17:19 05/17 17:17:27
256M  0.1  79.8K 61.7K   8G
  13 host01         terminate-hard user 05/17 17:17:27 05/17 17:37:52
256M  0.1 124.6K 85.9K   8G

  VID HOSTNAME      ACTION      REAS      START_TIME      END_TIME
MEMORY CPU NETRX NETTX  DISK
  14 host02         nic-attach user 05/17 17:38:16 05/17 17:40:00
256M  0.1  16.5K 13.2K   8G
  14 host02         poweroff   user 05/17 17:40:00 05/17 17:53:40
256M  0.1  38.3K 18.8K   8G
  14 host02         terminate-hard user 05/17 17:55:55 05/18 14:54:19
256M  0.1    1M 27.3K   8G

  VID HOSTNAME      ACTION      REAS      START_TIME      END_TIME
MEMORY CPU NETRX NETTX  DISK
  29 host02         none       none 05/27 17:09:28      -
256M  1    2.4M 1.3K   10G
```

При запросе данных в формате CSV (`--csv`) первая строка будет представлять имена столбцов, а последующие – значения атрибутов, разделенные запятыми.

```
tivacct --csv
```

```

UID,VID,HOSTNAME,ACTION,REASON,START_TIME,END_TIME,MEMORY,CPU,NETRX,NETTX,DISK
0,12,host01,none,user,05/09 19:20:42,05/09 19:35:23,1024M,1,29.8M,638.8K,0K
0,13,host01,nic-attach,user,05/17 17:10:57,05/17
17:12:48,256M,0.1,19.2K,15.4K,8G
0,13,host01,nic-detach,user,05/17 17:12:48,05/17
17:13:48,256M,0.1,36.9K,25K,8G
0,13,host01,nic-attach,user,05/17 17:13:48,05/17
17:14:54,256M,0.1,51.2K,36.4K,8G
0,13,host01,nic-detach,user,05/17 17:14:54,05/17
17:17:19,256M,0.1,79.8K,61.7K,8G
0,13,host01,nic-attach,user,05/17 17:17:19,05/17
17:17:27,256M,0.1,79.8K,61.7K,8G
0,13,host01,terminate-hard,user,05/17 17:17:27,05/17
17:37:52,256M,0.1,124.6K,85.9K,8G
0,14,host02,nic-attach,user,05/17 17:38:16,05/17
17:40:00,256M,0.1,16.5K,13.2K,8G
0,14,host01,poweroff,user,05/17 17:40:00,05/17
17:53:40,256M,0.1,38.3K,18.8K,8G
0,14,host02,terminate-hard,user,05/17 17:55:55,05/18
14:54:19,256M,0.1,1M,27.3K,8G
0,29,host02,none,none,05/27 17:09:28,-,256M,1,2.4M,1.3K,10G

```

При запросе результатов в формате XML (параметр `-x`) данные формируются в соответствии со следующей XML-схемой.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://tivmanager.org/XMLSchema"
xmlns="http://tivmanager.org/XMLSchema">

<xs:element name="HISTORY_RECORDS">
  <xs:complexType>
    <xs:sequence maxOccurs="1" minOccurs="1">
      <xs:element ref="HISTORY" maxOccurs="unbounded" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="HISTORY">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="OID" type="xs:integer"/>
      <xs:element name="SEQ" type="xs:integer"/>
      <xs:element name="HOSTNAME" type="xs:string"/>
      <xs:element name="HID" type="xs:integer"/>
      <xs:element name="CID" type="xs:integer"/>
      <xs:element name="STIME" type="xs:integer"/>
      <xs:element name="ETIME" type="xs:integer"/>
      <xs:element name="VM_MAD" type="xs:string"/>
      <xs:element name="TM_MAD" type="xs:string"/>
      <xs:element name="DS_ID" type="xs:integer"/>
      <xs:element name="PSTIME" type="xs:integer"/>
      <xs:element name="PETIME" type="xs:integer"/>
      <xs:element name="RSTIME" type="xs:integer"/>
      <xs:element name="RETIME" type="xs:integer"/>
      <xs:element name="ESTIME" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="EETIME" type="xs:integer"/>

<!-- REASON values:
NONE = 0 History record is not closed yet
ERROR = 1 History record was closed because of an error
USER = 2 History record was closed because of a user action
-->
<xs:element name="REASON" type="xs:integer"/>

<!-- ACTION values:
NONE_ACTION = 0
MIGRATE_ACTION = 1
LIVE MIGRATE ACTION = 2
SHUTDOWN_ACTION = 3
SHUTDOWN_HARD_ACTION = 4
UNDEPLOY_ACTION = 5
UNDEPLOY_HARD_ACTION = 6
HOLD_ACTION = 7
RELEASE_ACTION = 8
STOP_ACTION = 9
SUSPEND_ACTION = 10
RESUME_ACTION = 11
BOOT_ACTION = 12
DELETE_ACTION = 13
DELETE_RECREATE_ACTION = 14
REBOOT_ACTION = 15
REBOOT_HARD_ACTION = 16
RESCHED_ACTION = 17
UNRESCHED_ACTION = 18
POWEROFF_ACTION = 19
POWEROFF_HARD_ACTION = 20
DISK_ATTACH_ACTION = 21
DISK_DETACH_ACTION = 22
NIC_ATTACH_ACTION = 23
NIC_DETACH_ACTION = 24
DISK_SNAPSHOT_CREATE_ACTION = 25
DISK_SNAPSHOT_DELETE_ACTION = 26
TERMINATE_ACTION = 27
TERMINATE_HARD_ACTION = 28
-->
<xs:element name="ACTION" type="xs:integer"/>

<xs:element name="VM">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ID" type="xs:integer"/>
      <xs:element name="UID" type="xs:integer"/>
      <xs:element name="GID" type="xs:integer"/>
      <xs:element name="UNAME" type="xs:string"/>
      <xs:element name="GNAME" type="xs:string"/>
      <xs:element name="NAME" type="xs:string"/>
      <xs:element name="PERMISSIONS" minOccurs="0"
maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="OWNER_U" type="xs:integer"/>
            <xs:element name="OWNER_M" type="xs:integer"/>
            <xs:element name="OWNER_A" type="xs:integer"/>
            <xs:element name="GROUP_U" type="xs:integer"/>
            <xs:element name="GROUP_M" type="xs:integer"/>

```

```

        <xs:element name="GROUP_A" type="xs:integer"/>
        <xs:element name="OTHER_U" type="xs:integer"/>
        <xs:element name="OTHER_M" type="xs:integer"/>
        <xs:element name="OTHER_A" type="xs:integer"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="LAST_POLL" type="xs:integer"/>

    <!-- STATE values,
    see
http://docs.tivmanagtr.org/stable/user/references/vm_states.html
-->
    <xs:element name="STATE" type="xs:integer"/>

    <!-- LCM_STATE values, this sub-state is relevant
only when STATE is
        ACTIVE (4)
    see
http://docs.tivmanagerorg/stable/user/references/vm_states.html
-->
    <xs:element name="LCM_STATE" type="xs:integer"/>
    <xs:element name="PREV_STATE" type="xs:integer"/>
    <xs:element name="PREV_LCM_STATE"
type="xs:integer"/>
    <xs:element name="RESCHEDED" type="xs:integer"/>
    <xs:element name="STIME" type="xs:integer"/>
    <xs:element name="ETIME" type="xs:integer"/>
    <xs:element name="DEPLOY_ID" type="xs:string"/>
    <xs:element name="MONITORING">
    <!--
        <xs:complexType>
            <xs:all>
                <- Percentage of 1 CPU consumed
(two fully consumed cpu is 200) ->
                <xs:element name="CPU"
type="xs:decimal" minOccurs="0" maxOccurs="1"/>
                <- MEMORY consumption in kilobytes
->
                <xs:element name="MEMORY"
type="xs:integer" minOccurs="0" maxOccurs="1"/>
                <- NETTX: Sent bytes to the network
->
                <xs:element name="NETTX"
type="xs:integer" minOccurs="0" maxOccurs="1"/>
                <- NETRX: Received bytes from the
network ->
                <xs:element name="NETRX"
type="xs:integer" minOccurs="0" maxOccurs="1"/>
            </xs:all>
        </xs:complexType>
    -->
</xs:element>
<xs:element name="TEMPLATE" type="xs:anyType"/>
<xs:element name="USER_TEMPLATE"
type="xs:anyType"/>
<xs:element name="HISTORY_RECORDS">

```

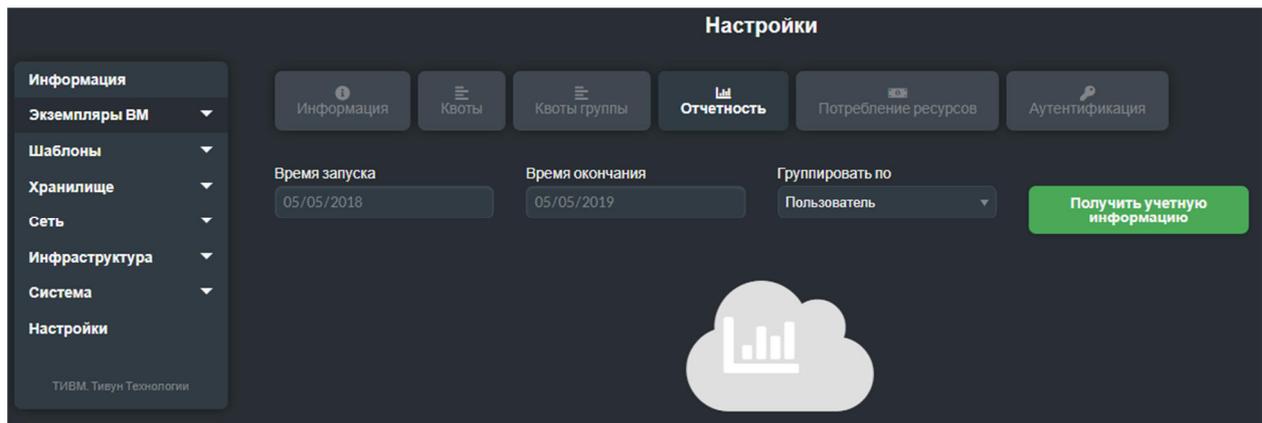



Рис. 3.14

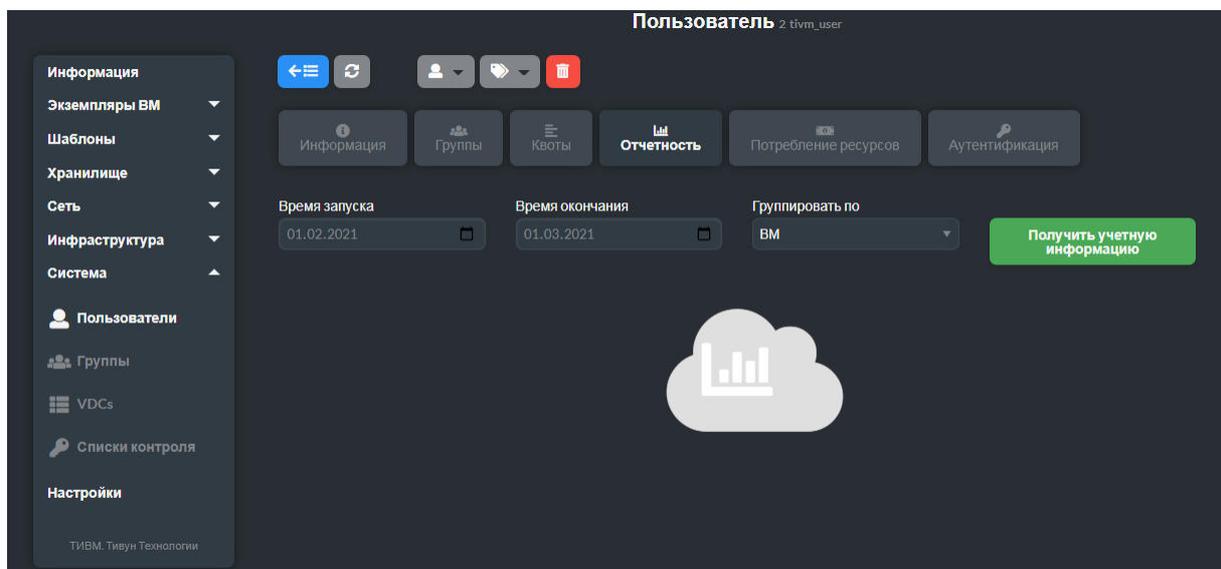


Рис. 3.15

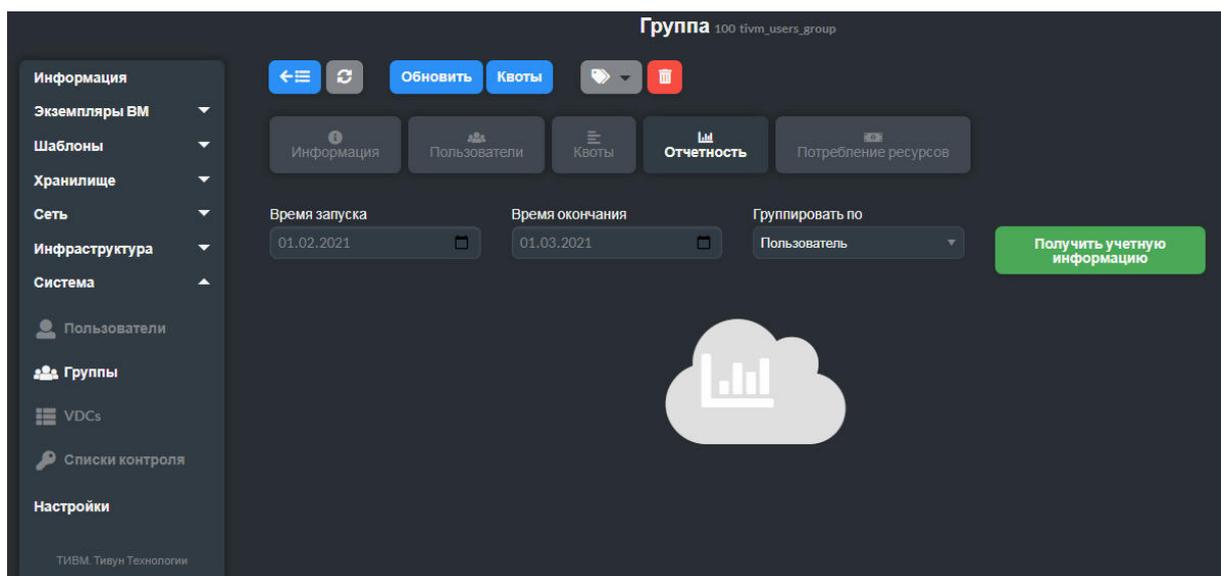


Рис. 3.16

3.8 Формирование отчетов о стоимости потребления вычислительных ресурсов виртуальной машиной

TIVM позволяет рассчитывать стоимость потребления вычислительных ресурсов ВМ и формировать соответствующие отчеты. Эти инструменты могут интегрироваться с внешними платежными системами.

В шаблоне ВМ можно задать параметры стоимости ее использования (рис. 3.17). *Общая стоимость* складывается из следующих составляющих:

- стоимость одного процессора в час;
- стоимость одного мегабайта памяти в час;
- стоимости одного гигабайта в хранилище данных в час.

Единицы затрат являются абстрактными, и их эквивалент денежным или другим стоимостным показателям должен определяться при каждом развертывании.

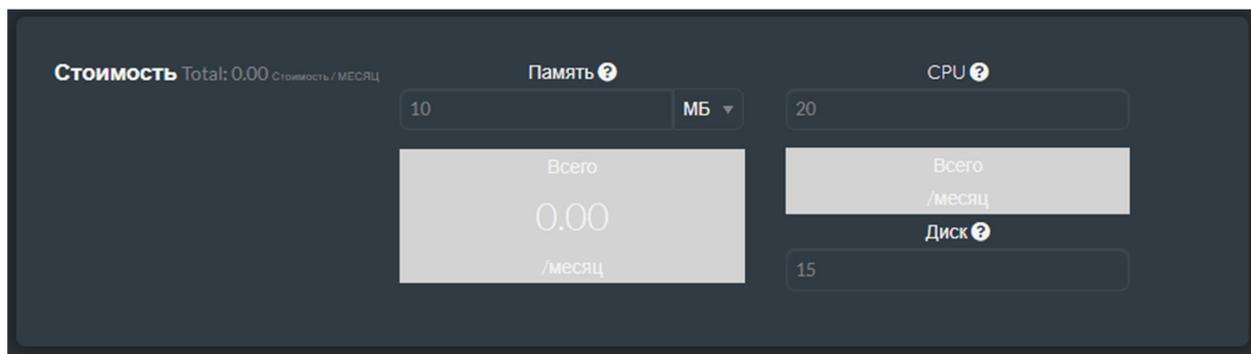


Рис. 3.17

Использование этой стоимостной схемы позволяет пользователям устанавливать объемные характеристики ВМ при их развертывании, исходя из общей стоимости их эксплуатации в единицу времени (рис. 3.18).

В конфигурации TIVM определена стоимость по умолчанию, которая будет применяться к шаблонам ВМ без определения стоимости. Соответствующие параметры задаются в файле `tiv.conf`.

Если пользователям доступно представление web-интерфейса TIVM «пользователь», важно установить стоимость по умолчанию. Обычные пользователи могут управлять своими шаблонами, для которых не будет назначена конкретная стоимость.

Для предоставления отчетов о стоимости пользователям администратор должен определить их в системе. Для *создания ежемесячного отчета* о стоимости вычислительных ресурсов служит команда `tivshowback calculate`:

```
tivshowback calculate [options]
```

Ее параметры (секция [options]):

- s, --start time – месяц начала отчетного периода;
- e, --end time – месяц окончания отчетного периода.

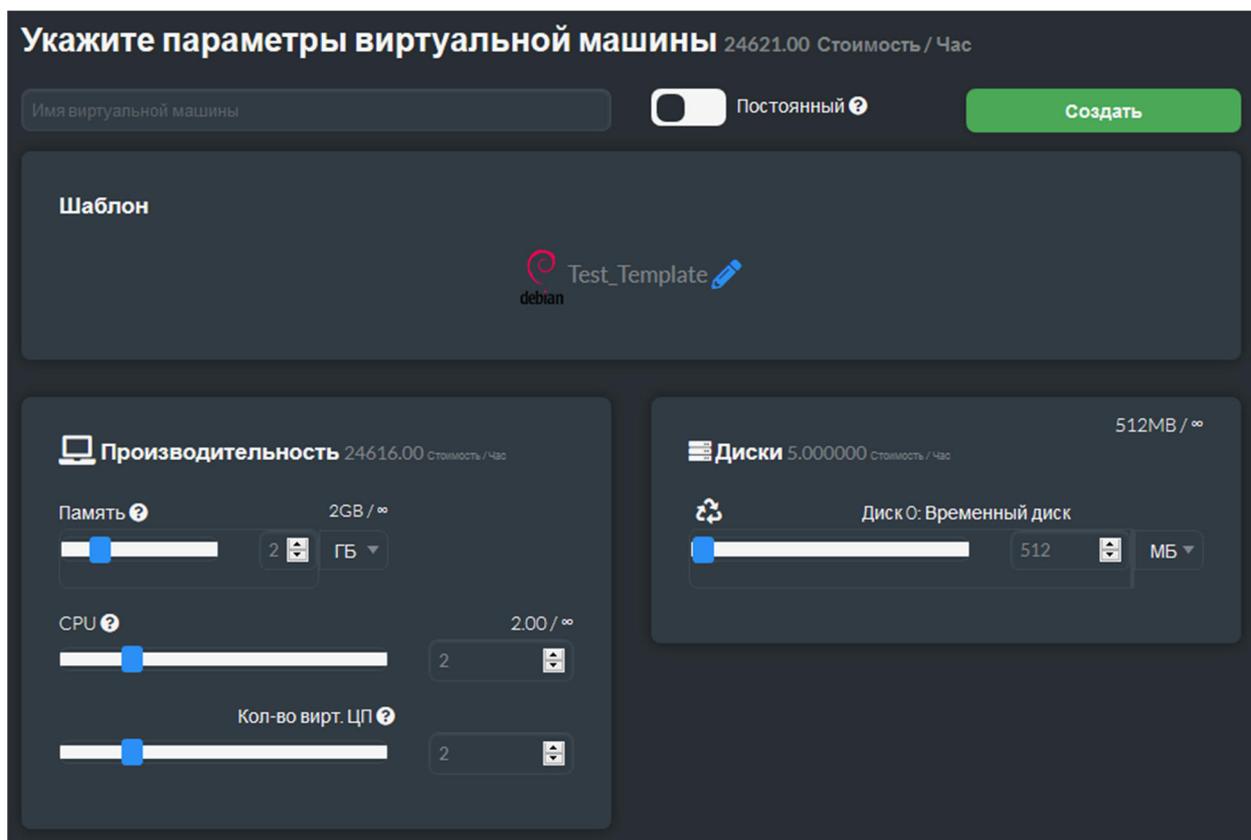


Рис. 3.18

При выполнении данной команды TIVM получает записи о функционировании VM за данный период и рассчитывает общую сумму расходов за каждый месяц. Формируемые записи включают общую стоимость за месяц, а также основную информацию о VM и ее владельце. Эта информация заносится в БД и используется командой которая будет использована командой `tivshowback list`.

Ежемесячная стоимость VM рассчитывается как сумма трех компонентов:

- CPU_COST*CPU*HOURS;
- MEMORY_COST*MEMORY*HOURS;
- DISK_COST*DISK_SIZE*HOURS.

Параметры CPU_COST, MEMORY_COST и DISK_COST – составляющие стоимости VM из ее шаблона. Параметры CPU, MEMORY и DISK – фактические объемные характеристики VM. Количество часов HOURS рассчитывается как общее количество часов, в течение которых VM

находилась в состоянии `running`. Нахождения ВМ в других состояниях, таких как `pending`, `poweroff` или `stopped` не учитываются в стоимости.

Если временной диапазон включает текущий месяц, TIVM рассчитает стоимость до сегодняшней даты. Администраторы должны оставить текущий месяц вне записей, обновлять его ежедневно или ежечасно. В любом случае важно пересчитать его по окончании месяца. Эта операция может быть легко автоматизирована с помощью задания системного планировщика.

Команда `tivshowback calculate` может быть выполнена только пользователем `tivadmin`.

Пример вызова `tivshowback calculate` для расчета данных о стоимости с августа по сегодняшний день:

```
tivshowback calculate --start "08/2020"
```

Пример вызова `tivshowback calculate` для расчета данных о стоимости с августа по сентябрь:

```
tivshowback calculate --start "09/2020" --end "09/2020"
```

Для получения записей о стоимости эксплуатации ВМ, которых были сформированы с помощью `tivshowback calculate`, служит команда `tivshowback list`:

```
tivshowback list [options]
```

Ее параметры (секция `[options]`):

- `-s, --start time` – дата и время начала отчетного периода;
- `-e, --end time` – дата и время окончания отчетного периода;
- `-u, --userfilter user` – имя или идентификатор пользователя (фильтр для отбора данных);
- `-g, --group group` – имя или идентификатор группы (фильтр для отбора данных);
- `--describe` – выводить описания столбцов;
- `--csv` – выводить информацию в CSV-формате;
- `-x, --xml` – представить информацию в XML-формате;
- `-j, --json` – представить информацию в JSON-формате;
- `-v, --verbose` – подробный режим;
- `-l, --list col1, col2, ...` – выводить только столбцы с указанными номерами (`col1, col2, ...`);
- `--user name` – имя пользователя для авторизации;
- `--password password` – пароль для авторизации;
- `--endpoint endpoint` – адрес сервиса TIVM, обрабатывающего запрос.

Для получения отчетов о стоимости эксплуатации ВМ в web-интерфейсе TIVM предназначена вкладка "Потребление" ресурсов на странице настроек (рис. 3.19).

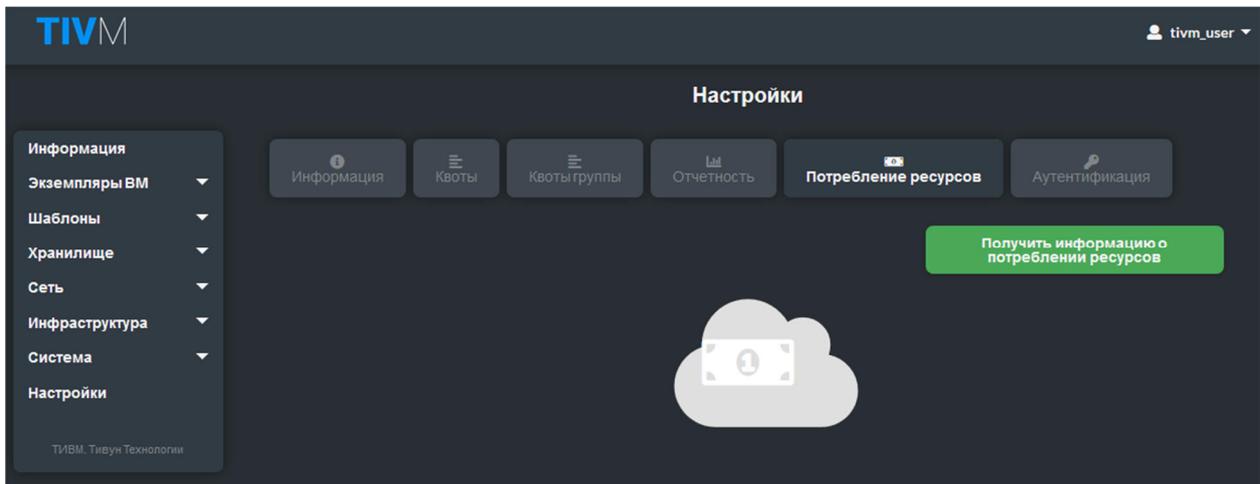


Рис. 3.19

Доступ к отчетам о стоимости эксплуатации ВМ может быть отключен в любом из представлений web-интерфейса TIVM путем изменения уml-файла в /etc/tiv/TIVM-views (параметру showback должно быть присвоено значение false).

...

```
features:  
  showback: false
```

4 Управление виртуальными сетями

4.1 Виртуальные сети и операции с ними

4.1.1 Определение виртуальной сети

Хост виртуализации подключен к одной или нескольким вычислительным сетям, которые доступны для ВМ через соответствующие мосты. TIVM позволяет создавать виртуальные сети, функционирующие поверх физических.

Определение виртуальной сети состоит из следующих компонентов:

- *базовая инфраструктура физической сети*, которая будет ее поддерживать, включая сетевой драйвер;

- *логическое адресное пространство*. Адреса, связанные с виртуальной сетью, могут быть IPv4, IPv6, IPv4-IPv6 с двумя стеками или Ethernet;

- *атрибуты гостевой конфигурации* для настройки сети ВМ, которые могут включать, например, сетевые маски, DNS-серверы или шлюзы;

- *группы безопасности*, действующие в сети (см. разд. 4.3).

Чтобы определить виртуальную сеть, необходимо сформировать ее шаблон, в котором указать *атрибуты базовой инфраструктуры*:

- имя сети (атрибут NAME);

- драйвер для реализации (атрибут VN_MAD);

- атрибуты качества обслуживания (Quality of Service, QoS) для каждого NIC, связываемого с сетью. Эти атрибуты позволяют ограничить входящую, исходящую, среднюю и пиковую пропускной способности, а также размера пакетных данных, которые могут передаваться на пиковой скорости.

К числу атрибутов QoS относятся:

- INBOUND_AVG_BW – средняя скорость передачи данных в Кб/с для входящего траффика;

- INBOUND_PEAK_BW – максимальная скорость передачи данных в Кб/с для входящего траффика;

- INBOUND_PEAK_KB – объем данных входящего траффика, который может передаваться на максимальной скорости в Кб;

- OUTBOUND_AVG_BW – средняя скорость передачи данных в Кб/с для исходящего траффика;

- OUTBOUND_PEAK_BW – максимальная скорость передачи данных в Кб/с для исходящего траффика;

- OUTBOUND_PEAK_KB – объем данных исходящего траффика, который может передаваться на максимальной скорости в Кб.

Определение атрибутов QoS не является обязательным.

В зависимости от драйвера может потребоваться указать дополнительные атрибуты, в частности: имя физической сети (атрибут PHYDEV), имя моста (атрибут BRIDGE) и др.

Например, шаблон, определяющий виртуальную сеть 802.1Q, должен содержать:

```
NAME      = "Private Network"
VN_MAD    = "802.1Q"
PHYDEV    = "eth0"
```

```
OUTBOUND_AVG_BW = "1000"
OUTBOUND_PEAK_BW = "1500"
OUTBOUND_PEAK_KB = "2048"
```

Адреса, доступные в виртуальной сети, определяются одним или несколькими *диапазонами адресов* (address range, AR). Каждый AR определяет непрерывный диапазон адресов и, необязательно, атрибуты конфигурации, которые будут переопределять атрибуты первого уровня, определенные в виртуальной сети. Существуют следующие типы AR:

- IPv4 – непрерывный набор адресов IPv4 (бесклассовый);
- IPv6 для определения глобальных и ULA IPv6 сетей;
- IPv6 без SLAAC для определения фиксированного 128-битного адреса IPv6;
- двойной стек – каждый NIC в сети получит как IPv4, так и IPv6-адрес (SLAAC или no-SLAAC);
- Ethernet – только MAC-адреса, генерируемые для VM. Этот тип AR используется, когда IP-адреса предоставляет внешняя служба (например, DHCP-сервер).

Например, для определения диапазона адресов IPv4 10.0.0.150 – 10.0.0.200 в шаблоне должно быть указано:

```
AR = [
  TYPE    = "IP4" ,
  IP      = "10.0.0.150" ,
  SIZE    = "51" ,
]
```

Для настройки *гостевой сети VM* виртуальная сеть может содержать дополнительную информацию, которая будет введена в виртуальную машину во время загрузки. Эти атрибуты *контекстуализации* могут включать, сетевые маски, DNS-серверы и шлюзы. Например, чтобы определить шлюз и DNS-сервер для VM в виртуальной сети, в шаблоне должно быть указано:

```
DNS      = "10.0.0.23"
GATEWAY  = "10.0.0.1"
```

Эти атрибуты автоматически добавляются в ВМ и обрабатываются пакетами контекстуализации. Для этого шаблон ВМ должен содержать:

```
CONTEXT = [  
    NETWORK="yes"  
]
```

Пример определения виртуальной сети:

```
NAME      = "Private"  
VN_MAD    = "802.1Q"  
PHYDEV    = "eth0"
```

```
AR=[  
    TYPE = "IP4",  
    IP   = "10.0.0.150",  
    SIZE = "51"  
]
```

```
DNS       = "10.0.0.23"  
GATEWAY   = "10.0.0.1"
```

```
DESCRIPTION = "A private network for VM inter-communication"
```

На основе этого файла шаблона будет создана сеть IPv4 с использованием тегов VLAN (идентификатор VLAN в этом случае назначается TIVM). Сеть будет предоставлять в аренду IP-адреса в диапазоне 10.0.0.150 – 10.0.0.200. ВМ в этой сети получают IP-адреса в этом диапазоне и настроят DNS-серверы на 10.0.0.23 и 10.0.0.1 в качестве шлюза по умолчанию.

4.1.2 Операции с виртуальными сетями

В CLI операции с виртуальными сетями реализует команда `tivvnet`. Они также доступны в web-интерфейсе TIVM на странице "Виртуальные сети", открываемой через меню Сеть/Виртуальные сети (рис. 4.1).

Чтобы создать виртуальную сеть, нужно подготовить ее шаблон, как описано в разд. 4.1.1, сохранить его в файле, а затем выполнить команду `tivvnet create`, указав имя шаблона в качестве аргумента. Например:

```
$ tivvnet create vnet_template.net  
ID: 4
```

Команда `tivvnet create` возвращает идентификатор созданной виртуальной сети. Он может указываться в других командах, оперирующих виртуальными сетями.

Для удаления виртуальной сети служит команда `tivvnet delete`. Ее аргумент – идентификатор или имя сети, подлежащей удалению. Примеры:

```
$ tivvnet delete 4
```

```
$ tivvnet delete "Private"
```

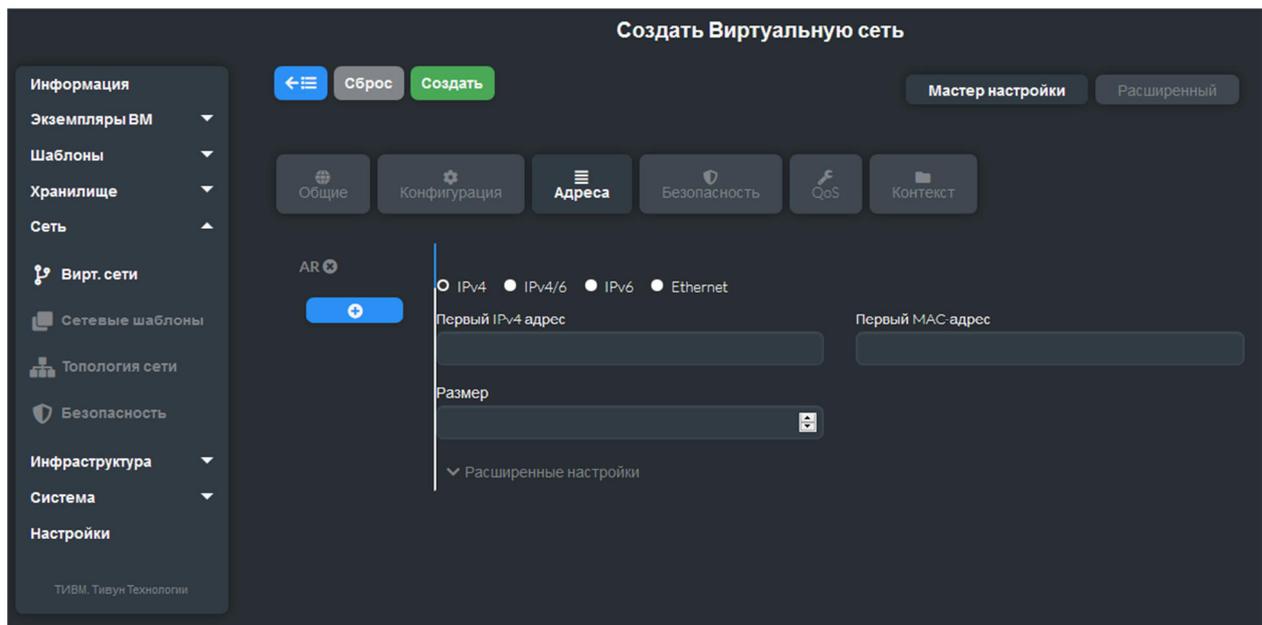


Рис. 4.1

Для получения списка виртуальных сетей, зарегистрированных в системе, служит команда `tivvnet list`:

```
$ tivvnet list
ID USER      GROUP      NAME      CLUSTER  BRIDGE  LEASES
0  admin     tivadmin   Private   0,100    tivbr.10  0
1  admin     tivadmin   Public    0,101    vbr0      0
```

Она формирует таблицу, в которой столбец `USER` представляет владельца сети, `GROUP` – группу, `LEASES` – число IP-адресов, присвоенных VM или зарезервированных.

Информацию о данной виртуальной сети возвращает команда `tivvnet show`. Ее аргументом служит идентификатор или имя виртуальной сети

```
$ tivvnet show 1
VIRTUAL NETWORK 4 INFORMATION
ID              : 4
NAME            : Private
USER            : user25
GROUP           : tivadmin
CLUSTERS        : 0
BRIDGE          : br4
VN_MAD          : 802.1Q
PHYSICAL DEVICE: eth0
VLAN ID         : 6
USED LEASES     : 0

PERMISSIONS
```

```
OWNER           : um-
GROUP           : ---
OTHER           : ---
```

VIRTUAL NETWORK TEMPLATE

```
BRIDGE="br4"
DESCRIPTION="A private network for VM inter-communication"
DNS="10.0.0.23"
GATEWAY="10.0.0.1"
PHYDEV="eth0"
SECURITY_GROUPS="0"
VN_MAD="802.1Q"
```

ADDRESS RANGE POOL

```
AR 0
SIZE           : 51
LEASES        : 0
```

RANGE	FIRST	LAST
MAC	02:00:0a:00:00:96	02:00:0a:00:00:c8
IP	10.0.0.150	10.0.0.200

Если какие-то IP-адреса в сети используются для внутренних потребностей, они могут быть зарезервированы, чтобы исключить их присвоение VM. В шаблоне виртуальной сети можно указать несколько атрибутов AR, чтобы определить диапазоны адресов, не включающие зарезервированные адреса.

AR может иметь размер, равный 1, для определения схемы аренды отдельных адресов.

AR не обязательно должны быть одного типа или принадлежать к одной IP-сети. В рамках этого варианта использования можно перезаписать атрибуты контекста в AR, например, добавив такие атрибуты, как NETWORK_MASK или DNS, в определение AR.

В суперсетях можно комбинировать AR, перезаписывая физические атрибуты, например, BRIDGE или VLAN_ID. В этом случае виртуальная сеть может представлять собой логическую суперсеть, например, DMZ, которая может быть реализована через несколько виртуальных сетей, каждая из которых использует отдельный мост гипервизора.

Нет необходимости сразу фиксировать весь план назначения IP-адресов заранее. AR могут быть добавлены и изменены после создания виртуальной сети (см. п. 4.1.3).

Команда `tivvnet update` позволяет *менять атрибуты виртуальной сети*, обновляя ее шаблон. Могут быть изменены:

- атрибуты, соответствующие контексту и описанию сети;
- атрибуты физической конфигурации сети (например, PHYDEV и VLAN_ID);
- любой прикладной атрибут, присутствующий в шаблоне.

Аргументы команды:

- имя или идентификатор виртуальной сети;
- имя файла шаблона, содержащего новые значения атрибутов (при отсутствии этого файла будет открыт редактор текущего шаблона).

Имя виртуальной сети может быть *изменено* с помощью команды `tivvnet rename`. Аргументы команды:

- имя или идентификатор виртуальной сети;
- новое имя виртуальной сети.

4.1.3 Управление диапазонами адресов

Адреса виртуальной сети структурированы в AR. Диапазоны адресов можно динамически добавлять или удалять из виртуальной сети. Это позволяет добавлять адреса в существующую виртуальную сеть, если текущие адреса исчерпаны.

AR может быть *добавлен* с использованием параметров определения, описанных в п. 4.1.1. Например, следующая команда добавит AR из 20 IP-адресов в виртуальную сеть с именем "Private":

```
tivvnet addar Private --ip 10.0.0.200 --size 20
```

Команда `tivvnet rmar` предназначена для *удаления AR* из шаблона виртуальной сети. Ее аргументы:

- имя или идентификатор сети;
- внутренний идентификатор AR в шаблоне сети (номер AR по порядку следования в шаблоне, начиная с 0).

Например, следующая команда удаляет второй AR из виртуальной сети с именем "Private":

```
tivvnet rmar Private 2
```

Команда `tivvnet updatear` позволяет *обновлять* следующие *атрибуты AR*:

- размер диапазона (`SIZE`) – назначенные адреса не могут выходить за его пределы;
- префикс IPv6 (`GLOBAL_PREFIX` и `ULA_PREFIX`);
- любой прикладной атрибут, который может переопределить значения по умолчанию для виртуальной сети.

Аргументы команды:

- имя или идентификатор сети;
- внутренний идентификатор AR в шаблоне сети;
- имя файла шаблона, содержащего новые значения атрибутов AR (при отсутствии этого файла будет открыт редактор текущего шаблона).

IP-адреса виртуальной сети могут быть временно помечены как *зарезервированные* (имеющие признак `hold`). Они по-прежнему являются частью сети, но они не будут назначены ни одной VM.

Для резервирования и освобождения IP-адресов AR предназначены команды `tivvnet hold` и `tivevnet release`. Их аргументы:

- имя или идентификатор виртуальной сети;
- IP-адрес (резервируемый или освобождаемый).

По умолчанию адрес будет удерживаться во всех AR, содержащих его. Если необходимо резервировать IP-адрес определенного AR, то диапазон должен быть указан с помощью параметра `-a`, после которого указывается внутренний идентификатор AR.

Пример. Команда резервирования адреса 10.0.0.120 во всех AR виртуальной сети "Private Network":

```
$ tivvnet hold "Private Network" 10.0.0.120
```

Пример. Команда резервирования адреса 10.0.0.123 в диапазоне AR с идентификатором 0 виртуальной сети "Private Network":

```
$ tivvnet hold 0 10.0.0.123 -a 0
```

В списке арендованных IP-адресов, возвращаемом командой `tivvnet show`, зарезервированные адреса отображаются как используемые VM с идентификатором -1.

4.1.4 Использование виртуальных сетей

После настройки виртуальных сетей их можно сделать доступными для пользователей на основе прав доступа и владения. Предпочтительный способ сделать это – через абстракцию ВДЦ. По умолчанию все виртуальные сети автоматически доступны для группы `users`.

Доступ VM к виртуальной сети обеспечивают 2 механизма:

1) описание в шаблоне VM связей NIC с определенной виртуальной сетью;

2) автоматический выбор используемой сети (виртуальная сеть выделяется VM планировщиком, как другие разделяемые ресурсы).

Чтобы подключить VM к виртуальной сети с помощью первого механизма, нужно указать ее имя или идентификатор в атрибуте `NIC`. Например, чтобы определить VM с сетевым интерфейсом, подключенным к виртуальной сети с именем "Private", нужно включить в шаблон VM ссылку на сеть:

```
NIC = [ NETWORK = "Private" ]
```

Вместо имени можно указать идентификатор сети:

```
NIC = [ NETWORK_ID = 0 ]
```

ВМ получит адрес из любого AR сети.

Можно запросить конкретный адрес путем добавления параметра IP или MAC в NIC. Например, чтобы разместить ВМ в сети "Private" и запросить адрес 10.0.0.153, нужно указать:

```
NIC = [ NETWORK = "Network", IP = 10.0.0.153 ]
```

Если TIVM не может получить запрошенный адрес у сети, связь ВМ с ней не устанавливается. Пользователь может использовать виртуальную сеть в своей ВМ, если ему предоставлены разрешения на операции USE с этой сетью.

Режим *автоматического выбора виртуальной сети для ВМ* включается путем установки значения атрибута NETWORK_MODE = "auto" в шаблоне ВМ:

```
NIC = [ NETWORK_MODE = "auto" ]
```

В этом режиме при развертывании ВМ планировщик выбирает виртуальную сеть из числа доступных сетей на хосте, где запускается ВМ. Данный механизм полезен при формировании типовых шаблонов ВМ, которые могут использоваться для запуска ВМ в разных кластерах облачной среды.

В атрибут NIC в данном режиме могут быть также включены атрибуты SCHED_REQUIREMENTS и SCHED_RANK. Это позволит определить требования к сетям, связываемым с конкретными NIC (SCHED_REQUIREMENTS) и предпочтения (SCHED_RANK) для выбора сети среди подходящих.

Пример:

```
NIC = [ NETWORK_MODE = "auto",  
        SCHED_REQUIREMENTS = "TRAFFIC_TYPE = \"public\" &  
        INBOUND_AVG_BW<1500",  
        SCHED_RANK = "-USED_LEASES" ]
```

В этом примере планировщик отберет виртуальные сети в данном кластере с прикладным атрибутом TRAFFIC_TYPE, имеющим значение public, и средней скорости передачи входящих данных, меньшей 1500 Кб/с. Из этих сетей будет использована та, что имеет наибольшее число свободных IP-адресов.

Для присоединения к ВМ псевдонима NIC можно сослаться на родительский NIC по его имени:

```
NIC = [ NETWORK = "public", NAME = "test" ]
```

Другой способ – использовать атрибут NIC_ALIAS:

```
NIC_ALIAS = [ NETWORK = "private", PARENT = "test" ]
```

Если атрибут NAME сети пуст, он генерируется автоматически по идентификатору сети в форме NIC\${NIC_ID}. Такое имя также может использоваться для получения псевдонима:

```
NIC_ALIAS = [ NETWORK = "private", PARENT = "NIC0" ]
```

Атрибут NIC_ALIAS может включать любой атрибут виртуальной сети, указываемый в составе атрибута NIC кроме NETWORK_MODE. Сети, определенные через псевдоним, не могут автоматически выбираться планировщиком.

Гипервизоры будут устанавливать MAC-адрес для сетевой карты ВМ, но не IP-адрес. Настройка IP-адресов в гостевой системе выполняется процессом контекстуализации (см. разд. 6). В качестве альтернативы пользовательская внешняя служба может настраивать сеть ВМ (например, собственный DHCP-сервер в отдельной ВМ).

4.1.5 Создание пользовательских виртуальных сетей

TVM предоставляет пользователям облачной среды 2 способа создания собственных виртуальных сетей:

- 1) бронирование;
- 2) развертывание на основе шаблона виртуальной сети.

В рамках первого способа виртуальная сеть пользователя формируется из частей существующей виртуальной сети. Выбор каждой части называется *бронированием*. Данный механизм состоит из следующих основных шагов.

- Определение виртуальной сети с желаемыми AR и атрибутами. Эти атрибуты будут унаследованы при бронировании, поэтому конечным пользователям не нужно иметь дело с сетевыми деталями низкого уровня.
- Настройка доступа. Чтобы организовать бронирование, пользователям нужны права USE в виртуальной сети, так же, как если бы они напрямую использовали ее и получали бы IP-адреса.

- Собственно бронирование. Пользователи могут запросить конкретные адреса или любые несколько адресов из сети. Забронированные адреса размещаются в виртуальной сети пользователя.
- Использование виртуальной сети, сформированной на основе бронирования. Такая виртуальная сеть реализует типовой интерфейс, поэтому для ее использования подходят те же методы, что и для любой виртуальной сети, созданной администратором (см. п. 4.1.4). Количество IP-адресов и статистика использования отображаются, как обычно.

Для бронирования адресов существующей виртуальной сети в создаваемой пользовательской сети необходимо:

- выбрать сеть-источник;
- определить имя пользовательской сети и число бронируемых адресов;
- вызвать команду `tivvnet reserve`:

```
$ tivvnet reserve Private -n MyVNET -s 10
Reservation VNET ID: 7
```

В данном примере бронируются 10 адресов из сети "Private", которые включаются в сеть "MyVNET". Команда `tivvnet list` показывает новую сеть в списке виртуальных сетей:

```
$ tivvnet list
ID USER      GROUP      NAME      CLUSTER  BRIDGE  LEASES
0  admin      tivadmin   Private   -        vbr1    10
7  user25     users     MyVNET    -        vbr1    0
```

Как видно из примера, в сети "Private" 10 адресов переданы в аренду (они забронированы для виртуальной сети с идентификатором 7). Важно, что обе сети используют одну и ту же конфигурацию, например, BRIDGE.

При бронировании могут быть заданы дополнительные параметры, в частности:

- AR, запрашиваемый для бронирования в исходной виртуальной сети;
- Начальный IP-адрес или MAC-адрес для бронирования.

Сеть, созданную с помощью бронирования, можно удалить как обычную виртуальную сеть:

```
$ tivvnet delete MyVNET
```

Сеть, созданную путем бронирования, можно использовать, как любую другую виртуальную сеть. Она предоставляет типовой интерфейс доступа. Например, чтобы присоединить ВМ к сети "MyVNET", сформированной с помощью бронирования, достаточно указать в шаблоне ВМ:

```
NIC = [ NETWORK = "MyVNET" ]
```

Сеть, созданную путем бронирования, можно расширить новыми IP-адресами. Другими словами, к существующему бронированию можно добавить новое. Таким образом, пользователь может обращаться к своей сети с контролируемым и детерминированным адресным пространством.

Увеличение числа бронированных IP-адресов увеличивает счетчики аренды у пользователя и группы. Расход этих ресурсов может быть ограничен с помощью квот.

Средства бронирования доступны в web-интерфейсе TIVM. Диалог бронирования (рис. 4.2) вызывается со страницы виртуальной сети (открываемой через пункт меню Сеть/Виртуальные сети), из которой будут запрошены IP-адреса. Для запуска диалога следует нажать кнопку "+" и выбрать "Забронировать".

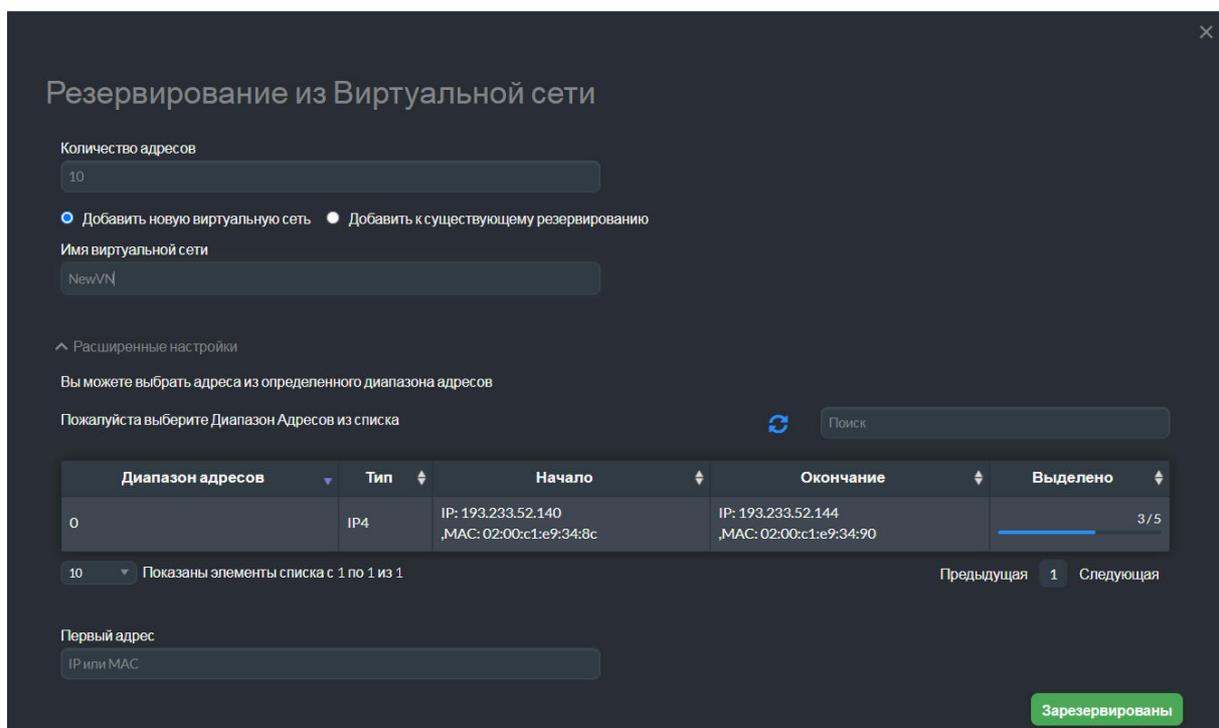


Рис. 4.2

Второй способ создания виртуальной сети пользователя основан на формировании ее шаблона (или применении готового шаблона). Он

позволяет сформировать сеть без обладания знаниями о лежащей в ее основе инфраструктуре. Работа с шаблонами виртуальных сетей описана в разд. 4.4.

4.2 Виртуальные маршрутизаторы

Виртуальные маршрутизаторы обеспечивают связь виртуальных сетей. Администраторы могут работать с виртуальными маршрутизаторами с помощью web-интерфейса TIVM и CLI.

Виртуальный маршрутизатор реализуется программно в рамках VM, разворачиваемой из шаблона, входящего в комплект поставки ПКВ "ТИВМ". Образ VM, реализующей виртуальный маршрутизатор, подготовлен для работы в режиме высокой готовности и обработки контекстной информации из TIVM. Таким образом, его базовые возможности могут быть расширены.

Для *создания виртуального маршрутизатора* в web-интерфейсе TIVM следует:

- открыть страницу управления маршрутизаторами через пункт меню Экземпляры VM/Маршрутизаторы;
- нажать кнопку "+";
- на странице диалога создания виртуального маршрутизатора (рис. 4.3) ввести необходимые атрибуты маршрутизатора и запускаемой VM;
- выбрать шаблон VM (TIVM предлагает реализацию "Alpine-KVM");
- выбрать связываемые виртуальные сети и ввести атрибуты сетевых интерфейсов;
- нажать кнопку "Создать".

Соединение виртуальных сетей через маршрутизатор вступает в силу, когда содержащая его VM разворачивается с сетевым интерфейсом, подключенным к каждой виртуальной сети.

Для каждой *виртуальной сети*, подключаемой к маршрутизатору, могут быть определены следующие *параметры*:

- принудительно устанавливаемый адрес IPv4 и IPv6 для NIC. Если маршрутизатор не настроен в режиме высокой доступности, это будет IP-адрес, запрашиваемый для устройства VM;
- признак плавающего IP-адреса. Используется только в условиях высокой доступности;
- признак интерфейса управления. Если флажок установлен, то сетевой интерфейс будет интерфейсом управления виртуальным маршрутизатором. Трафик не будет направляться на него;
- группы безопасности, применяемые для сетевого интерфейса.

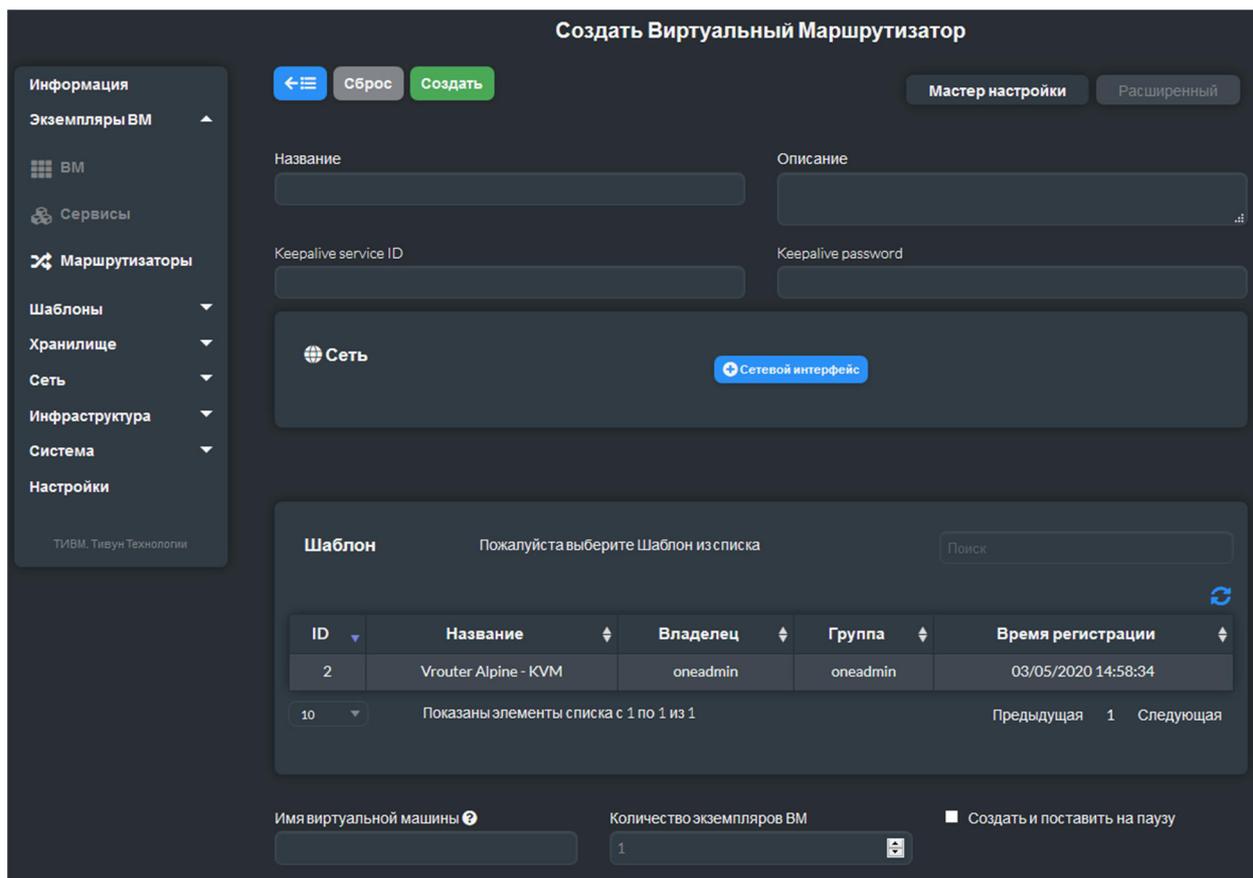


Рис. 4.3

Для управления виртуальными маршрутизаторами в CLI служит команда `tivvrouter`.

Чтобы *создать* виртуальный маршрутизатор в CLI, сначала необходимо сформировать файл его шаблона, указав следующие атрибуты:

- имя маршрутизатора (NAME);
- NIC связываемых виртуальных сетей.

Пример шаблона:

```
NAME = my-vr
NIC = [
  NETWORK="blue-net",
  IP="192.168.30.5" ]
NIC = [
  NETWORK="red-net" ]
```

Для создания маршрутизатора на основе шаблона предназначена команда `tivvrouter create`. Ее аргумент – имя файла шаблона. Пример:

```
$ tivvrouter create myvr.txt
ID: 1
```

Команда возвращает идентификатор зарегистрированного маршрутизатора. На этом шаге он еще не запущен, т.к. не имеет реализующей его VM.

Далее необходимо запустить VM для маршрутизатора. Это делает команда `tivvrouter instantiate`. Ее аргументы:

- имя или идентификатор виртуального маршрутизатора;
- имя или идентификатор шаблона VM;
- необязательный файл шаблона VM для переопределения атрибутов из исходного шаблона.

Пример:

```
$ tivvrouter instantiate 1 7
```

Команда `tivvrouter list` выводит список зарегистрированных виртуальных маршрутизаторов, команда `tivvrouter show` – информацию о данном маршрутизаторе (владелец, группа, список виртуальных сетей, соединенных этим маршрутизатором, VM, которые фактически обеспечивают маршрутизацию).

Эта же информация может быть получена в web-интерфейсе. Пункт меню *Экземпляры VM/Маршрутизаторы* открывает страницу списка виртуальных маршрутизаторов (рис. 4.4), на которой можно выбрать маршрутизатор и открыть страницу его описания (рис. 4.5).

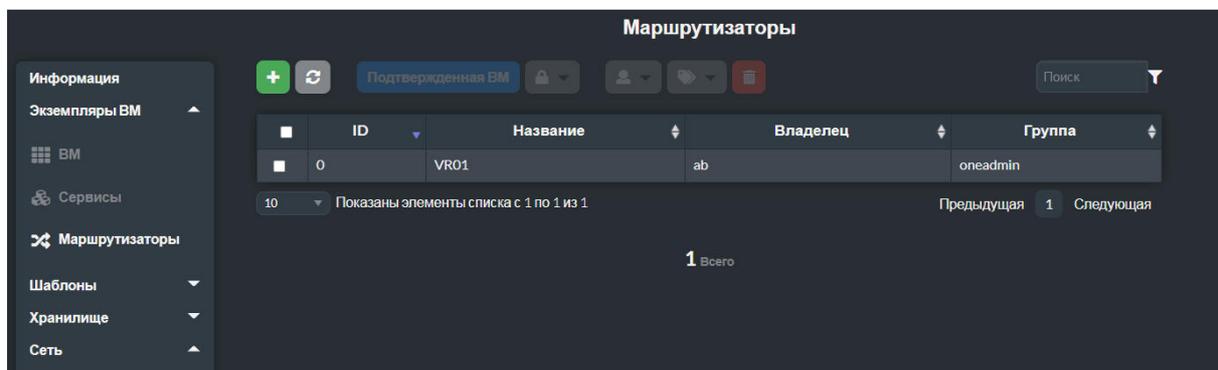


Рис. 4.4

Топология виртуальных сетей отображается на странице, открываемой через пункт меню *Сеть/Топология сети* (рис. 4.6).

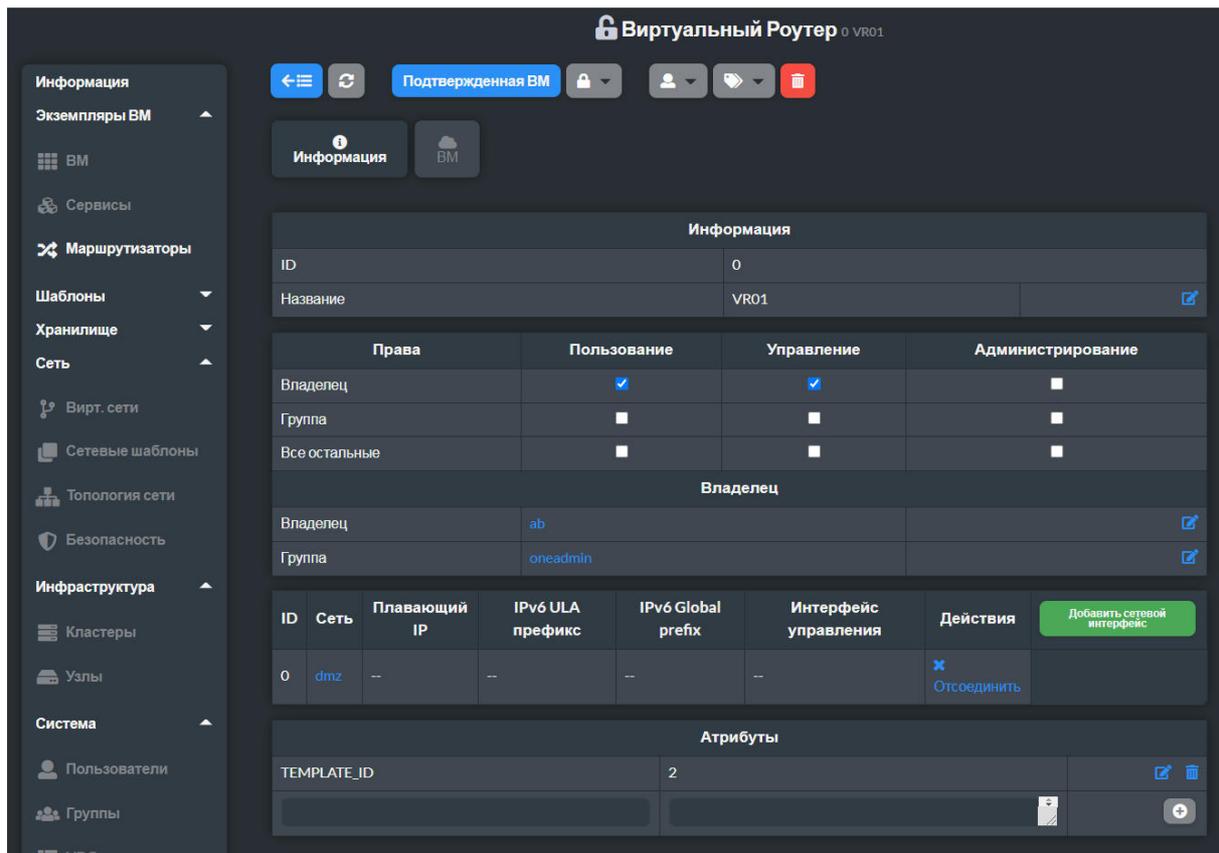


Рис. 4.5

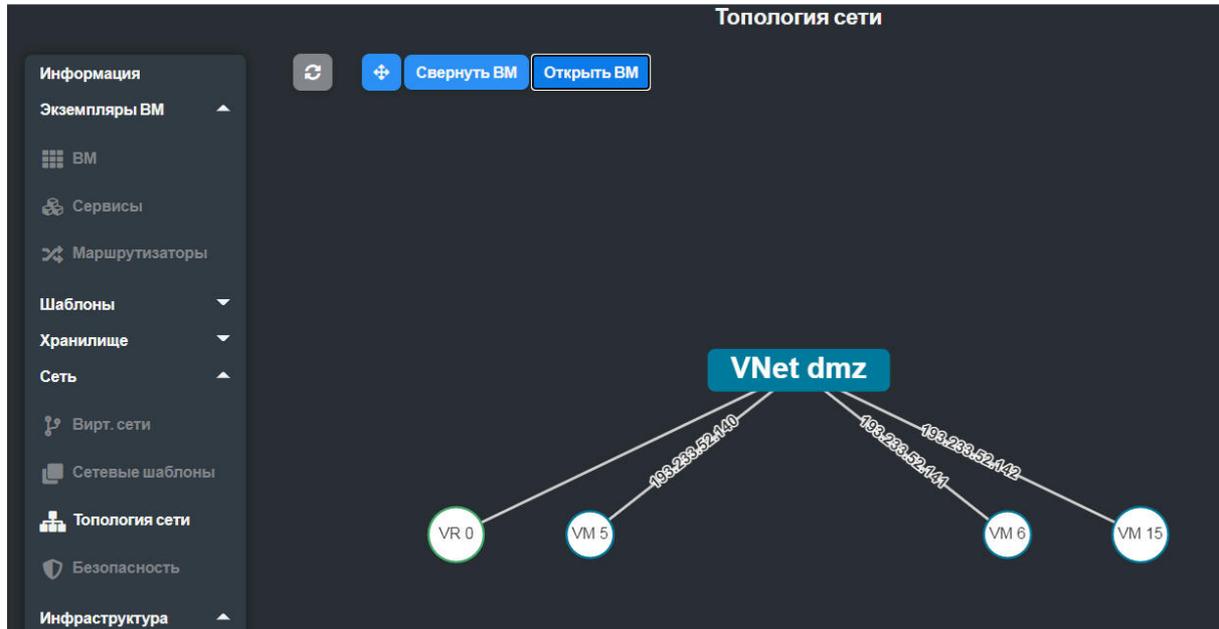


Рис. 4.6

Виртуальные сети могут быть подключены или отключены от работающего виртуального маршрутизатора. В web-интерфейсе TIVM эти возможности предоставляет страница виртуального маршрутизатора

(рис. 4.5). Кнопка "Отсоединить" в таблице сетей отключает виртуальную сеть. Кнопка "Добавить сетевой интерфейс" открывает диалог ввода его параметров.

Эти операции в CLI реализуют команды:

- `tivvrouter nic-attach` – подключение NIC виртуальной сети к маршрутизатору;

- `tivvrouter nic-detach` – отключение NIC виртуальной сети от маршрутизатора.

После подключения или отключения NIC виртуальной сети от виртуального маршрутизатора его реализация автоматически перенастраивается, чтобы начать маршрутизацию с учетом изменений. Перезагрузки VM маршрутизатора для этого не требуется.

С VM, реализующими виртуальные маршрутизаторы, могут выполняться все операции, предусмотренные TIVM, кроме `tivvm nic-attach` и `tivvm nic-detach`. Эти VM могут быть отключены и удалены. Вместо них к существующему виртуальному маршрутизатору могут быть добавлены другие VM.

Все VM, реализующие виртуальный маршрутизатор, автоматически удаляются при его удалении. Однако каждая такая VM может быть остановлена отдельно в любое время.

Для создания VM для реализации виртуального маршрутизатора следует использовать команду `tivvrouter instantiate` или диалог создания VM в web-интерфейсе TIVM.

Для обеспечения работы виртуального маршрутизатора в *режиме высокой доступности* его реализация может задействовать несколько VM. В этом случае TIVM назначит плавающий IP-адрес группе VM, которая будет координировать управление трафиком, направленным на этот IP-адрес.

Чтобы включить сценарий высокой доступности, необходимо выбрать 2 или более экземпляра VM для реализации виртуального маршрутизатора. В CLI количество экземпляров виртуальных машин задается с помощью параметра `-m` команды `tivvrouter instantiate`.

В этом сценарии становятся доступны следующие параметры виртуального маршрутизатора:

- признак сохраняемого идентификатора. При установке данного флага параметр конфигурации `virtual_router_id` сохраняется при перезапуске маршрутизатора;

- признак сохранения пароля. При установке данного флага параметр конфигурации `authentication/auth_pass` сохраняется при перезапуске маршрутизатора.

Для каждого виртуального NIC в режиме высокой доступности должен быть установлен признак плавающего IP.

Назначение плавающего IP-адреса управляется аналогично обычным виртуальным IP-адресам. Виртуальная сеть, связанная с маршрутизатором, будет фиксировать аренду адреса, назначенного виртуальному маршрутизатору (не VM). Помимо плавающего IP-адреса каждая VM получит индивидуальный IP-адрес.

Другие VM в сети будут использовать плавающий IP для связи с VM виртуального маршрутизатора. В любой момент времени только одна VM маршрутизатора использует этот плавающий IP-адрес. Если активная VM дает сбой, другие VM будут согласовывать назначение плавающего IP-адреса новой VM.

В настройках *службы поддержки маршрутизатора, запущенной на реализующей его VM*, можно указать два дополнительных параметра:

- VROUTER_KEEPAALIVED_PASSWORD – пароль, используемый для защиты службы от пакетов мошеннических машин. По умолчанию сервис настроен без пароля;

- VROUTER_KEEPAALIVED_ID – идентификатор сервиса (0–255). Указание идентификатора полезно, когда в одной сети есть несколько виртуальных маршрутизаторов или других сервисов их поддержки. По умолчанию идентификатор генерируется из идентификатора виртуального маршрутизатора, но при необходимости можно задать его вручную.

Эти параметры могут быть введены в диалоге создания виртуального маршрутизатора web-интерфейса TIVM.

4.3 Группы безопасности

Группы безопасности определяют правила брандмауэра, применяемые для VM.

Группы безопасности не поддерживаются для сетей OpenvSwitch и vCenter.

Группа безопасности состоит из множества *правил*. Каждое правило определяется атрибутами, приведенными в табл. 4.1.

Таблица 4.1 – Атрибуты правил брандмауэра для группы безопасности

Атрибут	Обязательность	Описание	Значения
PROTOCOL	Обязательный	Протокол правила	ВСЕ, TCP, UDP, ICMP, IPSEC
RULE_TYPE	Обязательный	Направление траффика для правила (входящий или исходящий)	INBOUND, OUTBOUND

Атрибут	Обязательность	Описание	Значения
IP	Необязательный	Задается, если правило применяется только к конкретной сети. Это первый IP из последовательного набора адресов. Должен использоваться с SIZE.	Действительный IP
SIZE	Необязательный	Задается, если правило относится только к конкретной сети. Определяет общее количество последовательных IP-адресов сети. Используйте всегда с IP.	Целое число, больше или равно 1
RANGE	Необязательный	Диапазон портов для их фильтрации определенных портов. Работает только с TCP и UDP.	Список в синтаксисе iptables. Пример: 22, 53, 80:90, 110, 1024:65535
ICMP_TYPE	Необязательный	Определенный тип ICMP-правила. Если тип имеет несколько кодов, он включает все коды внутри. Может использоваться с ICMP. Если пропустить это правило, оно повлияет на весь протокол ICMP.	0,3,4,5,8,9,10, 11,12,13,14, 17,18
NETWORK_ID	Необязательный	Идентификатор сети, к которой будет применяться эта группа безопасности.	Действительный сетевой идентификатор

Список зарегистрированных групп безопасности отображается в веб-интерфейсе TIVM: страница открывается через пункт меню Сеть/Безопасность (рис. 4.7). Выбор группы в списке открывает страницу ее редактирования (рис. 4.8), а кнопка "+" вызывает диалог создания группы (рис. 4.9).

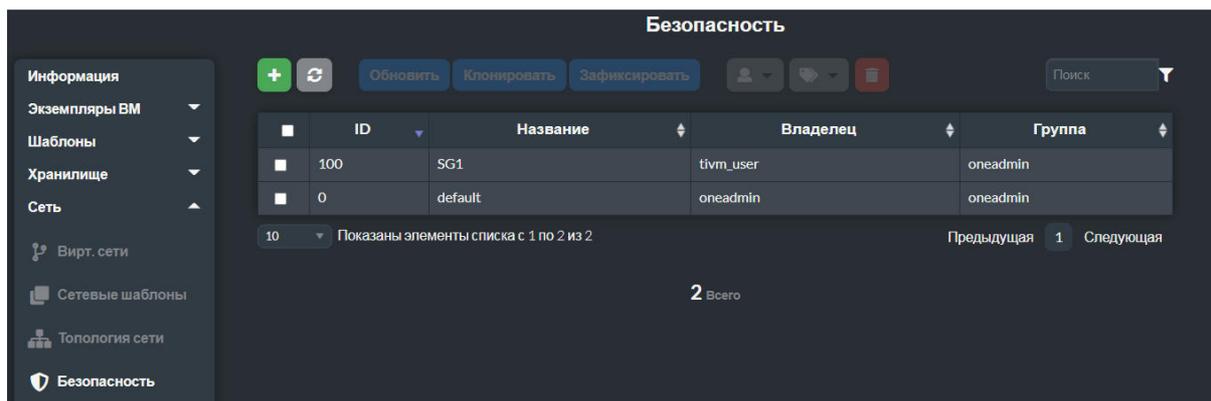


Рис. 4.7

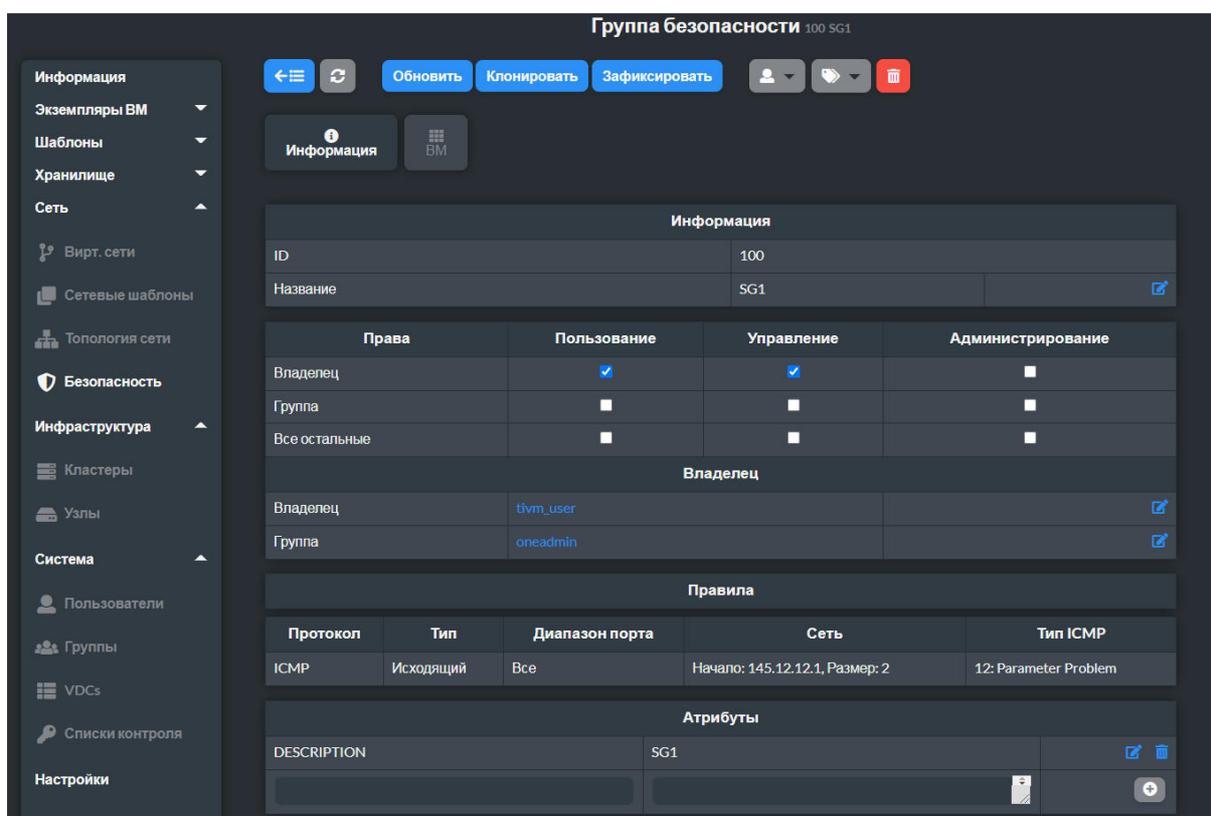


Рис. 4.8

Создание и управление группами безопасности возможно и с помощью CLI. Для определения группы необходимо сформировать файл ее шаблона. Пример:

```
NAME = test

RULE = [
    PROTOCOL = TCP,
    RULE_TYPE = inbound,
    RANGE = 1000:2000
]
```

```
RULE = [
    PROTOCOL= TCP,
    RULE_TYPE = outbound,
    RANGE = 1000:2000
]
```

```
RULE = [
    PROTOCOL = ICMP,
    RULE_TYPE = inbound,
    NETWORK_ID = 0
]
```

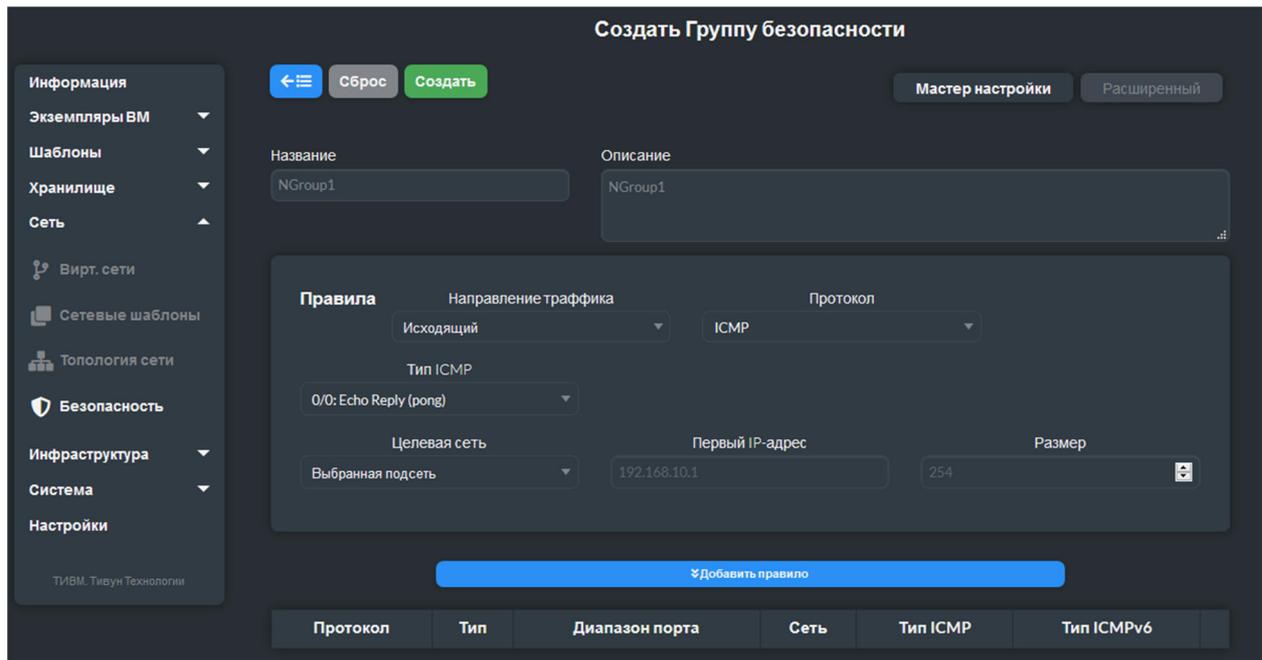


Рис. 4.9

Команда создания группы безопасности по шаблону:

```
$ tivsecgroup create ./sg.txt
ID: 102
```

Аргумент команды – имя файла шаблона. Команда возвращает идентификатор созданной группы безопасности.

Чтобы группа безопасности *применялась* к VM, она должна быть приписана виртуальным сетям, с которыми связана VM.

При создании VM *правила из групп безопасности копируются в ресурс VM*. Список правил отображается в web-интерфейсе TIVM на вкладке "Сеть" страницы описания VM (рис. 4.10). Они также выводятся командой `tivvm show` (см. пример ниже).

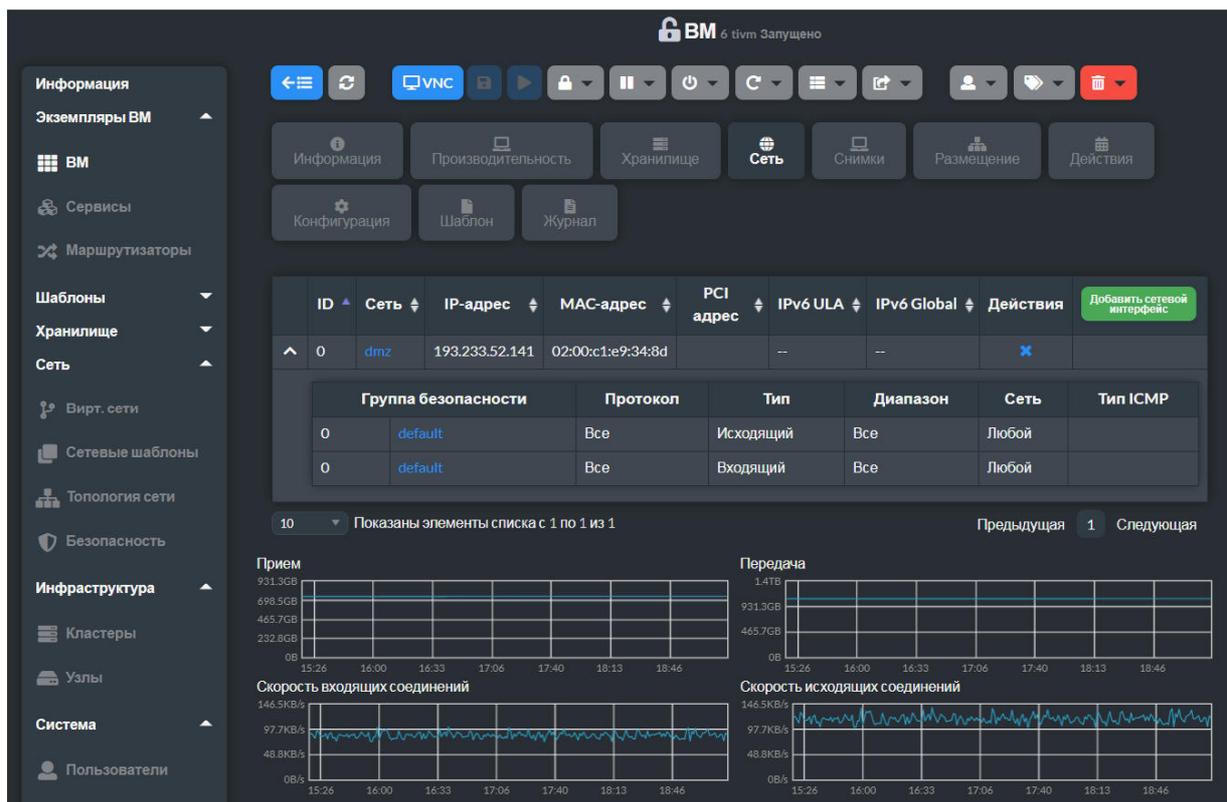


Рис. 4.10

Пример представления правил безопасности в составе информации о VM, возвращенной командой `tivvm show`:

```
$ tivvm show 6
VIRTUAL MACHINE 6 INFORMATION
ID                : 6
NAME              : tivm
USER              : tivadmin
GROUP            : tivadmin
STATE            : ACTIVE
LCM_STATE        : RUNNING
LOCK              : None
RESCHED          : No
HOST             : 127.0.0.1
. . .

VIRTUAL MACHINE MONITORING
. . .

VM DISKS
. . .

VM NICs
  ID NETWORK      BRIDGE          IP                MAC                PCI_ID
  0 dmz           tivbr.62        193.233.52.141  02:00:c1:e9:34:8d

SECURITY
```

```

NIC_ID NETWORK                                SECURITY_GROUPS
   0 dmz                                        0

SECURITY_GROUP  TYPE      PROTOCOL NETWORK RANGE
  ID NAME
   0 default    OUTBOUND ALL   VNET START  SIZE
   0 default    INBOUND  ALL
. . .

```

Группы безопасности могут быть назначены как для виртуальной сети в целом, так и для AR внутри нее.

Каждый NIC шаблона VM может определять список групп безопасности:

```

NIC = [
  NETWORK = "private-net",
  NETWORK_UNAME = "tivadmin",
  SECURITY_GROUPS = "103, 125"
]

```

Если для AR или NIC определены группы безопасности, то их идентификаторы будут добавлены к идентификаторам, заданным для виртуальной сети. Все идентификаторы группы безопасности объединяются и применяются к экземпляру VM.

Существует специальная *группа безопасности по умолчанию* с именем `default` и идентификатором `0`. Эта группа разрешает весь внешний трафик и весь внутренний трафик.

Всякий раз, когда создается виртуальная сеть, группа безопасности `default` добавляется в нее. Это означает, что необходимо отредактировать каждую вновь созданную сеть и удалить группу безопасности `default` из нее. В противном случае добавление иных групп безопасности не будет иметь эффекта, т.к. `default` разрешает весь трафик и переопределяет правила других групп.

Администратор системы может удалить правила, включенные в группу безопасности `default`. Это приведет к необходимости пользователям создавать собственные группы безопасности (в противном случае они не будут иметь подключения к VM).

Группы безопасности могут *редактироваться*. Эти изменения после их подтверждения (команда `tivsecgroup commit`) распространяются на все VM, к которым относится группа безопасности, поэтому может потребоваться некоторое время, чтобы изменения вступили в силу. Статус VM можно проверить в свойствах группы безопасности, где перечислены VM, использующие устаревшие и актуальные правила безопасности.

Команда применения изменений в группе безопасности на ВМ: `tivsecgroup commit`. Ее аргумент – имя или идентификатор группы безопасности.

4.4 Шаблоны виртуальных сетей

Шаблон виртуальной сети содержит формализованное описание ее конфигурации. Обычно администратор системы на этапе ее настройки готовит шаблоны с типовыми конфигурациями виртуальных сетей, включая в них инфраструктурные атрибуты. Конечные пользователи TIVM применяют данные шаблоны, дополняя их атрибутами, определяемыми прикладными требованиями (необходимыми AR, шлюзами, DNS-серверами и проч.). Это позволяет пользователям облачной среды управлять виртуальными сетями, абстрагируясь от деталей их физической реализации.

Шаблоны виртуальных сетей – разделяемый ресурс системы. На основе одного шаблона может создано множество виртуальных сетей.

При создании виртуальной сети на основе шаблона она привязывается к *кластеру по умолчанию*. Для явного указания кластеров, в которых будет доступна виртуальная сеть, служит атрибут `CLUSTER_IDS`. Ниже приведен простой пример шаблона виртуальной сети.

```
NAME=vntemplate
VN_MAD="bridge"
AR=[
IP="10.0.0.1",
SIZE="10",
TYPE="IP4" ]
CLUSTER_IDS="1,100"
```

По умолчанию создавать шаблоны виртуальной сети может только пользователь *tivadmin*. *Права на создание шаблонов* другими пользователями должны быть делегированы правилами СУД.

Права доступа к существующим шаблонам управляются с помощью соответствующих разрешений. Так, для создания виртуальной сети на основе данного шаблона, необходимо разрешение на операции USE с ним.

В CLI операции с шаблонами виртуальных сетей обеспечивает группа команд `tivvntemplate`. Для шаблонов виртуальных сетей доступны команды:

- `tivvntemplate chgrp` – определение группы шаблона;
- `tivvntemplate chmod` – определение разрешений на операции с шаблоном;
- `tivvntemplate chown` – определение владельца и группы шаблона;
- `tivvntemplate clone` – создание шаблона путем копирования данного шаблона;

- tivvntemplate create – создание шаблона;
- tivvntemplate delete – удаление шаблона;
- tivvntemplate instantiate – создание виртуальной сети на основе данного шаблона;
- tivvntemplate list – получение списка зарегистрированных шаблонов;
- tivvntemplate lock – заблокировать операции определенного уровня с виртуальной сетью, основанной на данном шаблоне. Уровень задается параметрами --admin (операции администрирования), --manage (операции управления), --use (операции использования) и --all (все операции);
- tivvntemplate rename – переименование шаблона;
- tivvntemplate show – получение описания шаблона;
- tivvntemplate unlock – заблокировать операции с виртуальной сетью, основанной на данном шаблоне, которые ранее были заблокированы командой tivvntemplate lock;
- tivvntemplate update – обновление шаблона.

Комплексный пример.

1. Системный администратор сформировал файл шаблона виртуальной сети vn_template1.txt:

```
NAME=vntemplate
VN_MAD="bridge"
BRIDGE="virbr0"
```

2. Создание шаблона:

```
$ tivvntemplate create vn_template1.txt
ID: 1
```

Команда возвратила идентификатор созданного шаблона (1).

3. Установка разрешений на использование шаблона для всех пользователей:

```
$ tivvntemplate chmod 1 604
```

4. Проверка шаблона:

```
$ tivvntemplate show 1
TEMPLATE 1 INFORMATION
ID           : 1
NAME         : vntemplate
USER         : tivadmin
GROUP        : tivadmin
```

```
LOCK : None
REGISTER TIME : 11/28 14:44:21
```

```
PERMISSIONS
OWNER : um-
GROUP : ---
OTHER : u--
TEMPLATE CONTENTS
BRIDGE="virbr0"
VN_MAD="bridge"
```

5. Создание виртуальной сети на основе шаблона:

```
$ tivvntemplate instantiate 1 --user user25 --name private
VN ID: 1
```

Команда возвратила идентификатор созданной сети (1).

6. Проверка присутствия созданной сети в списке виртуальных сетей:

```
$ tivvnet list
ID USER          GROUP          NAME          CLUSTERS BRIDGE  LEASES
0  tivadmin       tivadmin       public         0        virbr0  1
1  user25         users          private        0        virbr0  0
```

Виртуальная сеть с идентификатором 1, созданная пользователем user25, присутствует в списке. Она может использоваться в ВМ.

5 Управление виртуальными машинами

5.1 Образы дисков виртуальных машин

5.1.1 Типы образов дисков виртуальных машин и жизненный цикл образа

Существует 3 типа образов дисков VM, которые могут использоваться в *хранилищах образов* и *системных хранилищах* данных:

- OS – ОС, образ загрузочного диска. Шаблон VM должен определять один диск, ссылающийся на образ этого типа;

- CDROM – образ, содержащий данные, доступные только для чтения. Шаблон VM можно использовать только один образ этого типа;

- DATABLOCK – блок данных (хранилище данных). Эти образы могут быть созданы из существующих данных или представлять пустой диск.

В *хранилищах файлов* могут регистрироваться следующие 3 типа образов:

- KERNEL – простой файл, предназначенный для использования в качестве ядра ОС (атрибут VM OS/KERNEL_DS).

- RAMDISK – простой файл, предназначенный для использования, как виртуальный диск (атрибут VM OS/INITRD_DS).

- CONTEXT – простой файл, предназначенный для включения в контекстный CD-ROM (атрибут VM CONTEXT/FILES_DS).

Команда `tivimage chtype` позволяет изменить тип образа.

ЖЦ образа представлен в табл. 5.1.

Таблица 5.1 – ЖЦ образа диска VM

Краткое обозначение состояния	Обозначение состояние	Значение
lock	LOCKED	Файл образа копируется или создается в хранилище данных.
lock	LOCKED_USED	Файл образа копируется или создается в хранилище данных, а VM ожидают завершения операции.
lock	LOCKED_USED_PERS	То же, что LOCKED_USED, но для постоянных образов.
rdy	READY	Образ готов к использованию.
used	USED	Непостоянный образ, используемый, как минимум, одной

Краткое обозначение состояния	Обозначение состояние	Значение
		ВМ. Он может использоваться и другими ВМ.
used	USED_PERS	Постоянный образ, используемый ВМ. Он не может использоваться новыми ВМ.
disa	DISABLED	Образ отключен его владельцем, он не может использоваться новыми ВМ.
err	ERROR	Состояние ошибки, операция с образом завершилась неудачно. Сообщение об ошибке может быть получено с помощью команды <code>tivimage show</code> .
dele	DELETE	Образ удаляется из хранилища данных.
clon	CLONE	Образ копируется.

Образ может быть постоянным (*persistent*) и непостоянным (*non-persistent*). *Постоянный* образ связывается с экземпляром использующей его ВМ. При создании ВМ формируется копия этого образа, которую использует ВМ. Изменения ее данных сохраняются в хранилище после выключения ВМ.

Диаграмма состояний постоянного образа показана на рис. 5.1. Аналогичная диаграмма для непостоянного образа приведена на рис. 5.2.

5.1.2 Управление образами

Для управления образами предназначена группа команд `tivimage`. Средства управления образами также предусмотрены в web-интерфейсе TIVM (пункт меню Хранилище/Образы ВМ). По умолчанию эта страница доступна только в представлении *admin*.

Образы типов OS, CD-ROM и DATABLOCK могут быть *созданы* на основе существующего файла. Для блока данных есть возможность создать пустой образ, задав его размер.

Для создания образа загрузочного диска (ОС) нужно подготовить контекстную ВМ и извлечь ее диск. Файл с образом загрузочного диска загружается на сервер TIVM (рис. 5.3).

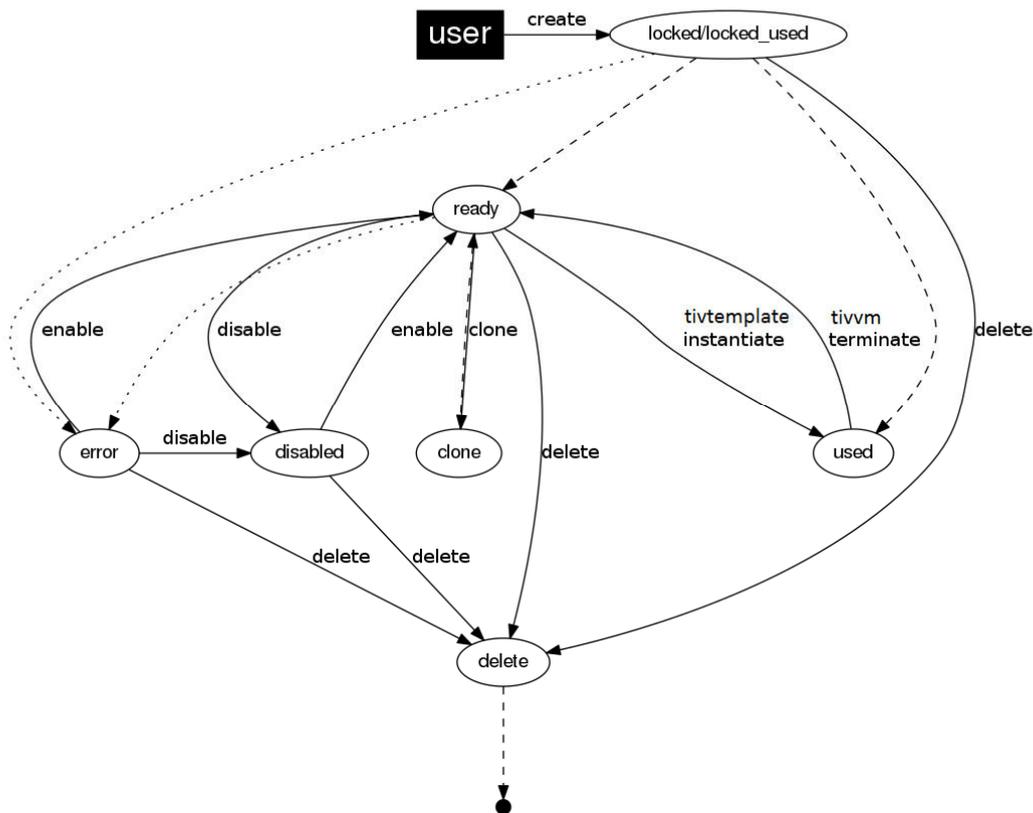


Рис. 5.1 – Диаграмма состояний постоянного образа диска VM

Для регистрации образа в CLI необходимо создать файл шаблона образа и выполнить команду **tivimage create**, передав ей имя данного файла, как аргумент.

Пример файла шаблона образа:

```

NAME           = "Ubuntu"
PATH           = "/home/cloud/images/ubuntu-desktop/disk.0"
TYPE          = "OS"
DESCRIPTION    = "Ubuntu desktop"
  
```

Образ создается в определенном хранилище данных. Для получения списка доступных хранилищ предназначена команда **tivdatastore list**:

```

tivdatastore list
ID NAME      SIZE AVAIL CLUSTERS IMAGES TYPE DS  TM      STAT
0 system    145.2G 56% 0         0   sys -  shared on
1 default   145.2G 56% 0         3   img fs  shared on
  
```



Рис. 5.2 – Диаграмма состояний непостоянного образа диска VM

По умолчанию в процессе установки TIVM создается одно хранилище образов (тип `img`).

Команда создания образа на основе файла шаблона в хранилище данных с именем `default`:

```
tivimage create ubuntu_img.txt --datastore default
ID: 0
```

Команда `tivimage create` позволяет создавать образы без передачи ей файла шаблона, задавая их атрибуты с помощью параметров. Перечень этих параметров команды приведен в табл. 5.2.

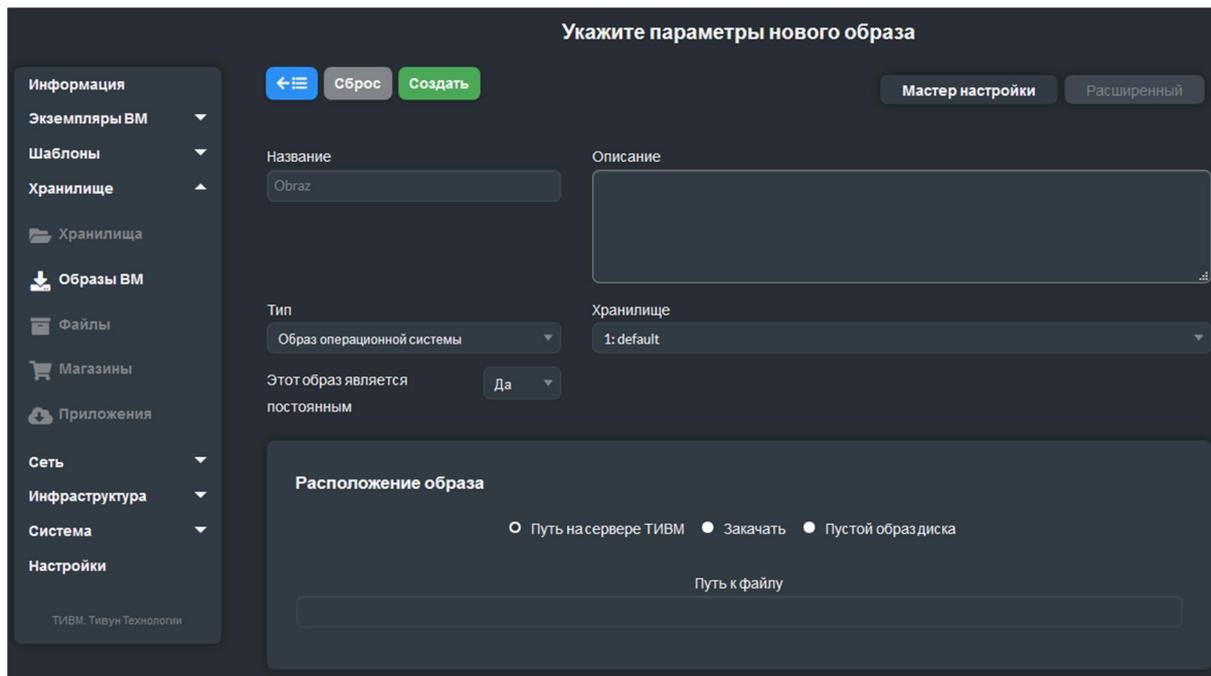


Рис. 5.3 – Диалог создания образа диска VM

Таблица 5.2 – Параметры команды `tivimage create`, определяющие атрибуты создаваемого образа диска VM

Параметр	Описание
<code>--name</code>	Имя образа
<code>--description</code>	Описание образа
<code>--type</code>	Тип образа: OS, CDROM, DATABLOCK, KERNEL, RAMDISK, CONTEXT
<code>--persistent</code>	Признак постоянного образа
<code>--prefix</code>	Префикс устройства для диска (например, hd, sd, xvd или vd)
<code>--target</code>	Устройство, к которому будет прикреплен диск
<code>--path</code>	Путь к файлу образа
<code>--driver</code>	Используемый драйвер (raw, qcow2, tap: aio: ...)
<code>--disk_type</code>	Тип образа диска (BLOCK, CDROM, FILE)
<code>--source</code>	Используемый источник данных (адрес). Указывается, если образ создается не из файла

Параметр	Описание
--size	Размер в мегабайтах. Используется для типа DATABLOCK

Создание образа из предыдущего примера обеспечивает команда:

```
tivimage create --datastore default --name Ubuntu --path
/home/cloud/images/ubuntu-desktop/disk.0 --description "Ubuntu
desktop"
```

Web-интерфейс TIVM поддерживает загрузку файлов образов, сжатых архиватором gz.

Процедура загрузки файла образа на сервер TIVM из web-интерфейса состоит из следующих шагов.

Шаг 1. Клиент загружает файл образа на сервер TIVM. Он размещается в папке `tpmdir`, указанной в конфигурации.

Шаг 2. TIVM регистрирует образ, устанавливая путь к этому временному файлу.

Шаг 3. TIVM копирует файл образа в хранилище данных.

Шаг 4. Временный файл образа удаляется из `tpmdir`, и запрос пользователя успешно завершается (в web-интерфейсе появляется сообщение о том, что образ был загружен).

Для обработки больших файлов образов (более 1 ГБ) может потребоваться много времени на шагах 1 и 3. Время выполнения процедуры может превысить тайм-аут, установленный на сервере TIVM, что вызовет ошибку. Администратору системы следует увеличить тайм-аут, чтобы избежать такого исхода.

Операция *клонирования образа* полезна для создания его резервной копии перед модификацией или для получения частной постоянной копии, к которой будет иметь доступ определенный пользователь. Постоянные образы со снимками (snapshots) не могут быть клонированы. Команда клонирования – **tivimage clone**. Ее аргументы:

- имя или идентификатор образа;
- имя создаваемой копии образа.

Пример:

```
tivimage clone ImageName NewImageName
```

При клонировании создаваемая копия образа может быть помещена в другое хранилище данных:

```
tivimage clone ImageName NewImageName --datastore new_img_ds
```

Новое хранилище данных должно быть совместимо с текущим, то есть иметь те же драйверы `DS_MAD`.

Web-интерфейс TIVM содержит диалог операции клонирования (рис. 5.4).

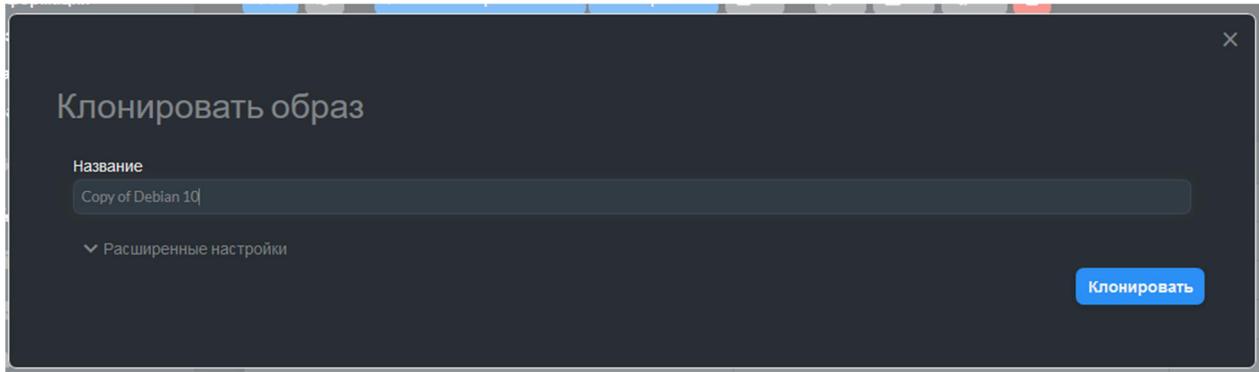


Рис. 5.4 – Диалог клонирования образа диска VM

Список зарегистрированных образов дисков возвращает команда **tivimage list**:

```
tivimage list
ID USER      GROUP      NAME                DATASTORE  SIZE  TYPE  PER  STAT  RVMS
0 tivadmin   tivadmin   ttylinux-vd         default     200M OS    No   used   8
1 user1     users      my-ubuntu-disk     default     200M OS    Yes  used   1
2 user4     testgroup  customized-ubun    default     200M OS    Yes  used   1
```

Описание образа возвращает команда **tivimage show**. Ее аргумент – имя или идентификатор образа. Пример:

```
tivimage show 4
IMAGE 4 INFORMATION
ID                : 4
NAME              : CentOS 8
USER              : tivadmin
GROUP             : tivadmin
LOCK              : None
DATASTORE         : default
TYPE              : OS
REGISTER TIME    : 05/17 10:26:29
PERSISTENT       : No
SOURCE            :
/var/lib/tiv//datastores/1/4716058d9f0af496be245af05d1a3647
PATH              : https://store.host.systems//appliance/fab2e5fc-
1113-11ea-b160-f0def1753696/download/0
SIZE              : 4G
STATE             : rdy
RUNNING_VMS      : 0

PERMISSIONS
OWNER             : um-
GROUP             : ---
OTHER             : ---
```

```
IMAGE TEMPLATE
DEV_PREFIX="vd"
DRIVER="qcow2"
FORMAT="qcow2"
FROM_APP="26"
FROM_APP_MD5="fbeccedfe02caf5574c5f43fe4b04602"
FROM_APP_NAME="CentOS 8"
```

*Список идентификаторов образов, которые не используются ни в одном из шаблонов VM, выводится командой **tivimage orphans**.*

Образы дисков VM могут быть разделяемым ресурсом облачной среды. Владелец образа может предоставить другим пользователям своей группы или всем пользователям системы *права на его использование*, установив для них разрешение на группу операций USE.

В следующем примере с помощью команды **tivimage chmod** устанавливается разрешение на операции USE с образом с идентификатором 0 для пользователей из группы его владельца:

```
tivimage show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---
tivimage chmod 0 640
tivimage show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---
```

Следующая команда позволяет пользователям в той же группе и использовать (USE), и управлять (MANAGE) образом, а прочим пользователям только – только использовать:

```
tivimage chmod 0 664
tivimage show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : um-
OTHER          : u--
```

Для установки типа образа – постоянный или непостоянный – предназначены команды **tivimage persistent** и **tivimage nonpersistent** соответственно. Аргумент команд – диапазон

идентификаторов образов в форме `n.m,k` либо список идентификаторов или имен образов, перечисленных через запятую. Список может состоять из одного элемента.

Пример:

```
tivimage list
  ID USER      GROUP      NAME      DATASTORE      SIZE TYPE PER STAT  RVMS
  0  tivadmin  tivadmin  Ubuntu    default         10G  OS  No  rdy   0

tivimage persistent Ubuntu
tivimage list
  ID USER      GROUP      NAME      DATASTORE      SIZE TYPE PER STAT  RVMS
  0  tivadmin  tivadmin  Ubuntu    default         10G  OS  Yes rdy   0

tivimage nonpersistent 0
tivimage list
  ID USER      GROUP      NAME      DATASTORE      SIZE TYPE PER STAT  RVMS
  0  tivadmin  tivadmin  Ubuntu    default         10G  OS  No  rdy   0
```

Постоянные образы со снимками (snapshots) не могут переведены в непостоянные. Чтобы выполнить такое преобразование, необходимо либо удалить снимки (команда `tivimage snapshot-delete`), либо перевести образ к одному из снимков с последующим удалением всех снимков (команда `tivimage snapshot-flatten`, см. ниже).

Постоянный образ может иметь связанные *снимки состояния* (snapshots), формируемые по командам пользователя в течение ЖЦ ВМ, использующей данный образ. Следующие команды предназначены для *управления снимками*:

- **tivimage snapshot-revert** *image_id snapshot_id* – активное состояние образа *image_id* заменяется снимком *snapshot_id*. Операция удаляет все несохраненные данные о состоянии диска;

- **tivimage snapshot-delete** *image_id snapshot_id* – удаляет снимок *snapshot_id* из образа *image_id*. Операция разрешена, только если снимок *snapshot_id* не является активным для данного образа, и у него нет дочерних элементов;

- **tivimage snapshot-flatten** *image_id snapshot_id* – преобразует образ *image_id* в образ без снимков, заменяя состояние образа снимком *snapshot_id*. Операция аналогична запуску `snapshot-revert` с последующим удалением всех снимков.

5.1.3 Использование образов в виртуальных машинах

Образ диска можно использовать в ВМ, указав ссылку на него в шаблоне ВМ в атрибуте DISK. При развертывании этой ВМ диск будет взят из хранилища образов.

Ссылка на образ может быть указана в атрибуте DISK двумя способами:

- в атрибуте `IMAGE_ID` со значением, равным идентификатору образа;
- в атрибуте `IMAGE` со значением, равным имени образа. В этом случае имя относится к одному из образов, принадлежащих пользователю (имена образов должны быть уникальным для одного пользователя). Если необходимо сослаться на образ, принадлежащий другому пользователю, это можно сделать, добавив атрибут `IMAGE_UID` (уникальный идентификатор владельца) или `IMAGE_UNAME` (имя владельца).

Пример шаблона ВМ:

```
CPU = 1
MEMORY = 3.08

DISK = [ IMAGE_ID = 7 ]

DISK = [ IMAGE = "Ubuntu",
        IMAGE_UNAME = "tivadmin" ]

DISK = [ type = swap,
        size = 1024 ]

NIC = [ NETWORK_ID = 1 ]
NIC = [ NETWORK_ID = 0 ]

GRAPHICS = [
    type      = "vnc",
    listen    = "1.2.3.4",
    port      = "5902" ]
```

Изменения в данных диска ВМ, внесенные в ходе ее работы, могут быть сохранены двумя способами:

- как снимки состояния диска. Снимок сохраняется, и в дальнейшем можно вернуть диск к этому состоянию с помощью команды `tivimage snapshot-revert`;

- с помощью операции "Сохранить диск как" (`save_as`) – на основе диска создается новый образ, сохраняемый хранилище данных. Далее этот диск может быть использован в другой ВМ. При выполнении операции диск должен быть в согласованном состоянии.

Для использования в ВМ файловых образов типов `KERNEL` и `RAMDISK` они должны быть указаны в атрибутах `OS/KERNEL_DS` и `OS/INITRD_DS` шаблона ВМ. Ссылки на образы задаются по их именам или идентификаторам.

Пример:

```
OS = [ KERNEL_DS = "$FILE[IMAGE=kernel3.6]",
      INITRD_DS  = "$FILE[IMAGE_ID=23]",
      ROOT       = "sda1",
```

```
KERNEL_CMD = "ro console=tty1" ]
```

Файловый образ типа CONTEXT позволяет определить данные для контекстуализации CD-ROM. Ссылка на образ указывается в атрибуте CONTEXT шаблона VM:

```
CONTEXT = [  
  FILES_DS = "$FILE[IMAGE_ID=34] $FILE[IMAGE=kernel]",  
]
```

5.2 Шаблоны виртуальных машин

Шаблон VM содержит формализованное описание ее конфигурации. Администратор системы на этапе ее настройки готовит шаблоны с типовыми конфигурациями VM. Конечные пользователи TIVM применяют данные шаблоны для запуска VM. При запуске они могут дополнять конфигурации VM контекстными параметрами, отражающими прикладные требования и условия применения.

Шаблоны VM – разделяемый ресурс системы. На основе одного шаблона может создано множество виртуальных сетей.

5.2.1 Определение виртуальной машины

Набор характеристик VM, определяемых в шаблоне, включает:

- *общие атрибуты* (имя, тип гипервизора, группа VM);
- *объемные характеристики*, представляемые размером оперативной памяти и числом процессоров;
- *множество NIC*, подключенных к одной или нескольким виртуальным сетям;
- набор образов *дисков*;
- дополнительные атрибуты – устройства ввода-вывода, устройства PCI, последовательность загрузки, контекстная информация и др.

Страница списка шаблонов VM в web-интерфейсе TIV открывается через пункт меню Шаблоны/VM (рис. 5.5).

Страница создания шаблона открывается при нажатии на кнопку "+". Она содержит набор вкладок, соответствующих группам характеристик VM. На рис. 5.6 показана вкладка *общих и объемных характеристик*.

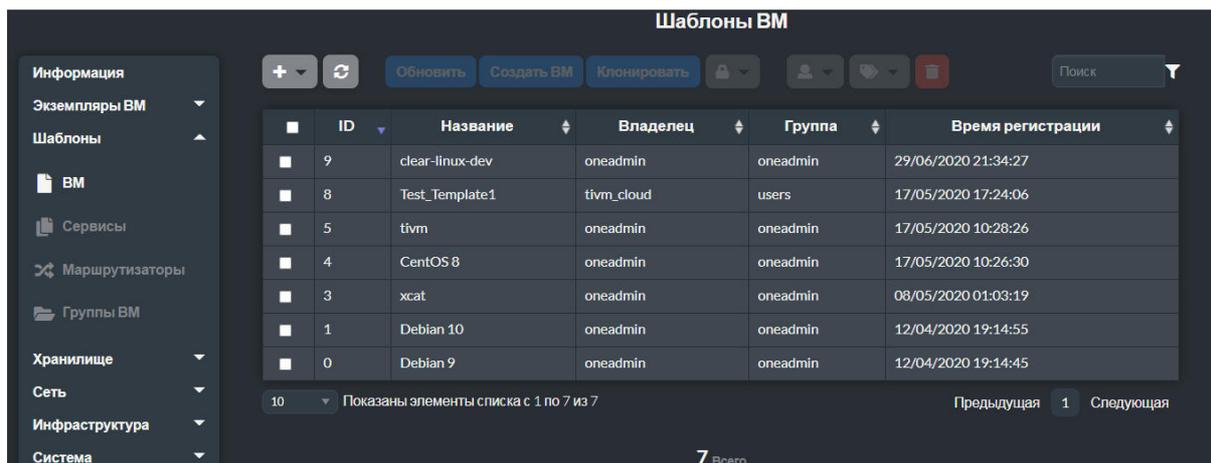


Рис. 5.5 – Страница списка шаблонов VM

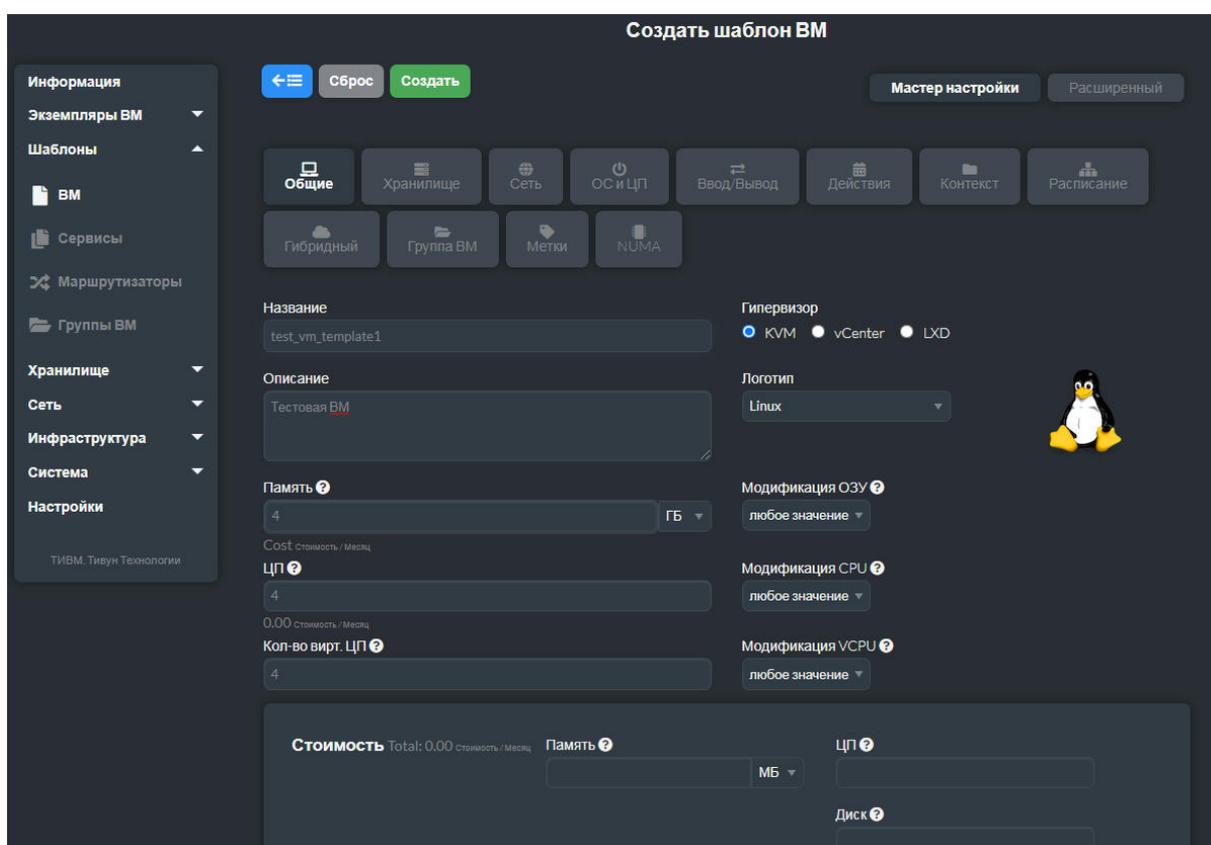


Рис. 5.6 – Страница создания шаблона VM. Вкладка общих и объемных характеристик

Число процессоров (CPU) выражается числом с плавающей запятой, для число виртуальных процессоров (VCPU) и объем памяти – целочисленными значениями.

Каждый диск VM определяется атрибутом DISK шаблона. VM может использовать три типа дисков:

- на основе *постоянного* образа (изменения в данные диска сохраняются после завершения работы VM);

- на основе *непостоянного* образа (VM использует копию исходного образа, и изменения в данных диска после отключения VM машины будут потеряны);

- *временный* (volatile) – диск создается на лету на целевом хосте. После завершения работы VM диск удаляется.

Параметры, относящиеся к атрибуту DISK, показаны на рис. 5.7.

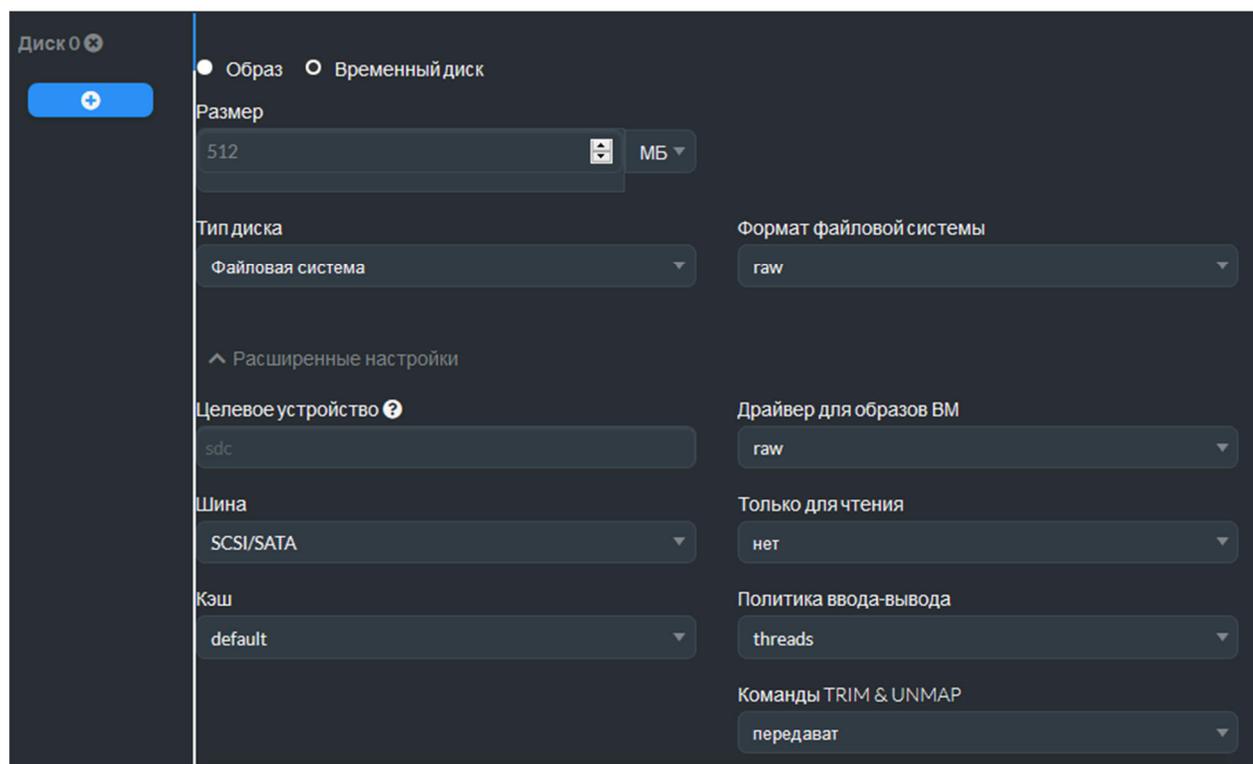


Рис. 5.7 – Страница создания шаблона VM. Вкладка характеристик дисков

Сетевые интерфейсы VM могут быть определены двумя способами:

1) *ручной выбор* – NIC связывается с данной виртуальной сетью. Этот способ может потребовать создания множества шаблонов, рассчитанных на виртуальные сети, созданные в каждом кластере;

2) *автоматический выбор* – виртуальная сеть выбирается при запуске VM планировщиком, как любой другой вычислительный ресурс. В шаблоне задаются требования к виртуальной сети и критерии выбора.

Названные способы описаны в п. 5.1.4. Диалог определения NIC в редакторе шаблона VM показан на рис. 5.8.

Для работы с шаблонами VM в CLI предназначена группа команд **tivtemplate**. Команда **tivtemplate create** создает шаблон VM на основе файла его описания. Ее аргумент – имя файла.

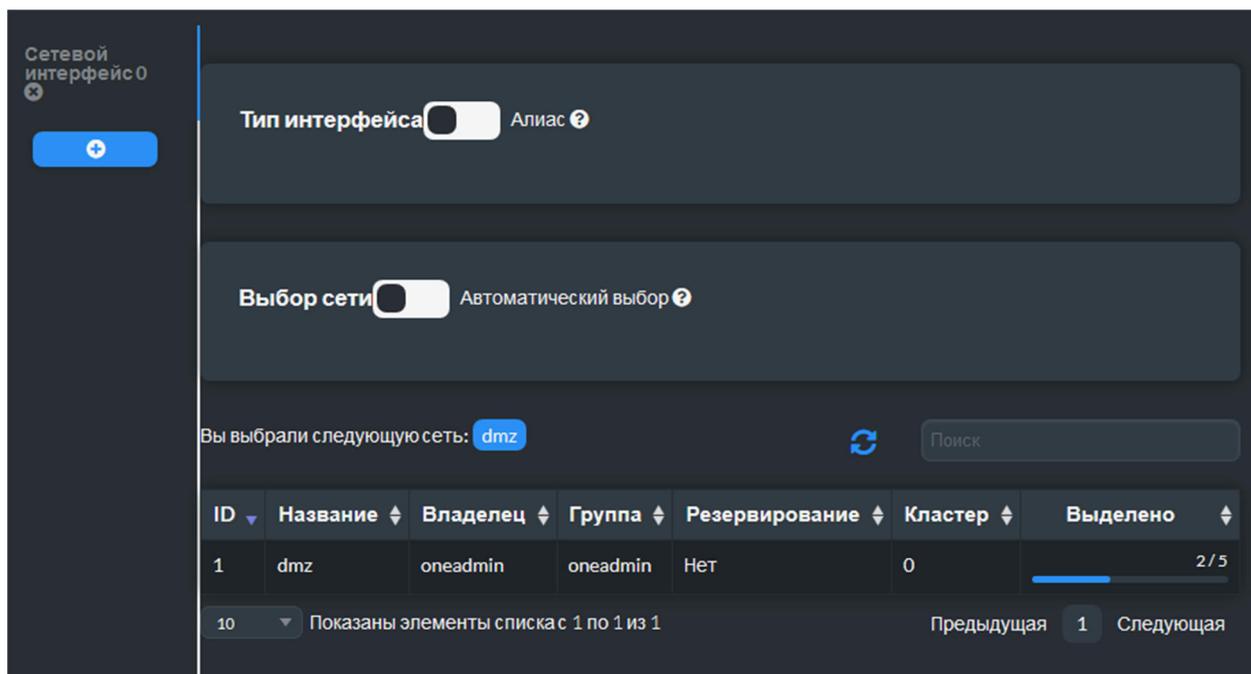


Рис. 5.8 – Страница создания шаблона VM. Вкладка определения сетевых интерфейсов

Пример простого шаблона VM в виде описания в текстовом файле:

```
NAME = test-vm
MEMORY = 128
CPU = 1

DISK = [IMAGE = "Arch Linux"]
DISK = [TYPE = swap,
        SIZE = 1024]

NIC = [NETWORK = "Public", NETWORK_UNAME = "tivadmin"]
NIC = [NETWORK = "Private", NETWORK_UNAME = "private_net"]
NIC_ALIAS = [NETWORK = "Public", PARENT = "private_net"]

GRAPHICS = [
    TYPE = "VNC",
    LISTEN = "0.0.0.0"]
```

Команда `tivtemplate create` позволяет создавать шаблоны без передачи их содержимого из файла, а указывая атрибуты с помощью параметров. Пример:

```
tivtemplate create --name test-vm --memory 128 --cpu 1
--disk "Arch Linux" --nic Public
```

Помимо предопределенных атрибутов модели шаблона VM, в него могут быть включены любые дополнительные прикладные атрибуты,

необходимые пользователям. Эти атрибуты вводятся на вкладке "Метки" (рис. 5.9).

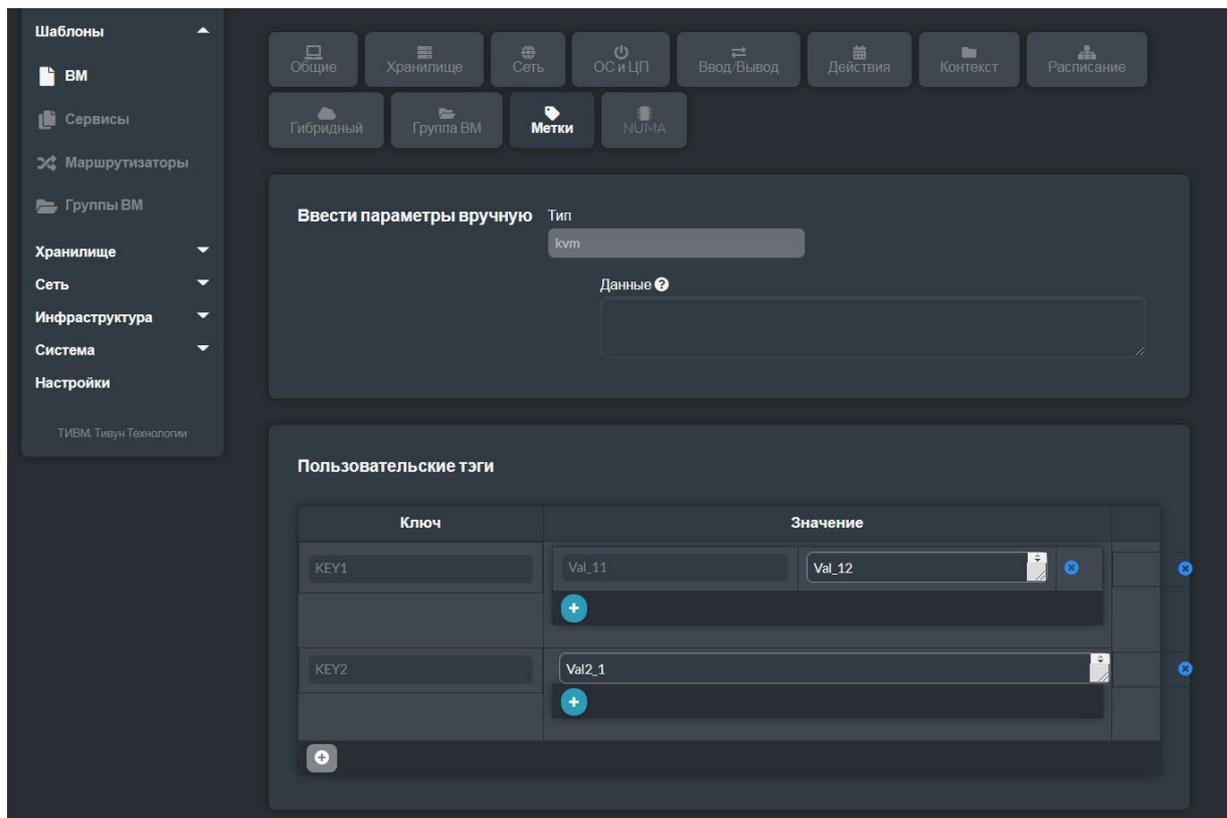


Рис. 5.9 – Страница создания шаблона ВМ. Вкладка прикладных атрибутов

Объемные характеристики ВМ (число процессоров, размер памяти, число виртуальных процессоров) *могут быть изменены пользователем при создании им экземпляра ВМ на основе данного шаблона*. Владелец шаблона может решить, можно ли менять эти атрибуты, и если да, то каким образом (рис. 5.10).

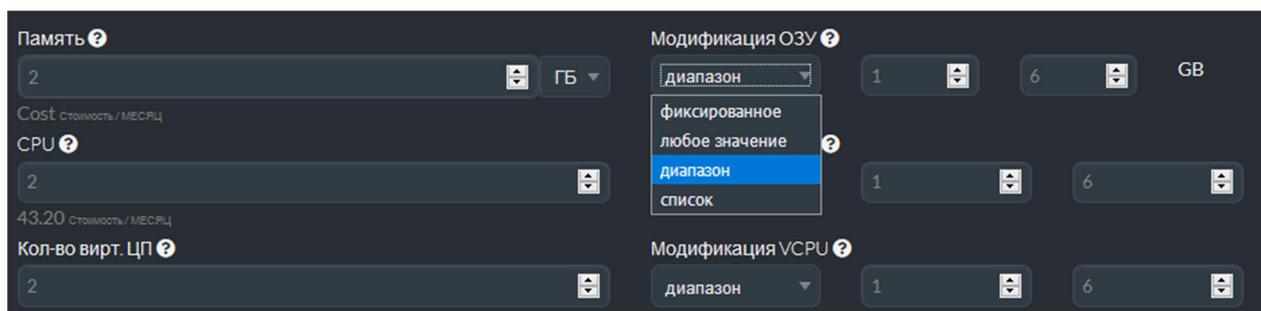


Рис. 5.10 – Параметры, определяющие возможность изменения объемных характеристик ВМ, указанных в шаблоне

Параметр *возможности модификации* характеристики ВМ имеют следующий набор значений:

- *фиксированное* – значение не может быть изменено;
- *любое значение* – пользователем, создающим экземпляр шаблона, может быть задано любое значение;
- *диапазон* – устанавливается левая и правая граница допустимых значений;
- *список* – пользователь может выбрать значений из заданного списка.

В файле описания шаблона, передаваемого команде `tivtemplate create`, возможности модификации объемных характеристик задаются параметрами вводимых пользователями атрибутов `USER_INPUTS` (см. ниже).

Пример:

```
CPU = "1"
MEMORY = "2048"
VCPU = "2"
USER_INPUTS = [
  CPU = "M | list || 0.5,1,2,4 | 1" ,
  MEMORY = "M | range || 512..8192 | 2048 " ]
```

Возможности модификации характеристики VM пользователем при ее запуске, предусмотренные в шаблоне, могут быть отключены в представлении *cloud* web-интерфейса TIVM.

TIVM предусматривает возможность *получения данных контекста* при запуске VM. Эти данные вводятся пользователем, запустившим VM, в специальном диалоге TIVM и передаются VM. Состав параметров, запрашиваемых у пользователя, определяется в шаблоне (вкладка "Контекст", раздел "Пользовательские переменные", рис. 5.11).

Элемент данных, запрашиваемый у пользователя, может иметь один из следующих типов:

- *текст* – символьное значение;
- *текст (base64)* – символьное значение, которое перед передачей VM будет представлено в кодировке base64;
- *пароль* – символьное значение. При его вводе оно не отображается, но сохраняется, как простой текст;
- *число* – целое число;
- *число (плавающее)* – число с плавающей запятой;
- *диапазон* – целое число в пределах определенного диапазона;
- *диапазон (плавающий)* – число с плавающей запятой в пределах определенного диапазона;
- *список* – значение из заданного списка;
- *список (несколько)* – одно или несколько значений из заданного списка;
- *boolean* – логическое значение.

Элемент данных имеет атрибут обязательности, определяющим, что значение должно быть введено пользователем или может быть опущено. Для числовых, списковых и логических атрибутов также может быть установлено значение по умолчанию, принимаемое, если пользователь не ввел данные.

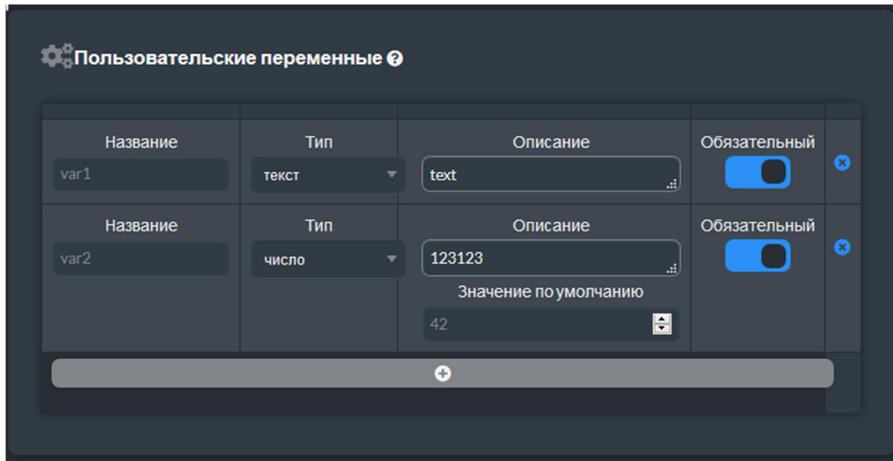


Рис. 5.11 – Редактор пользовательских параметров контекста

Пример диалога для уточнения объемных характеристик запускаемой ВМ и ввода пользовательских параметров показан на рис. 5.12.

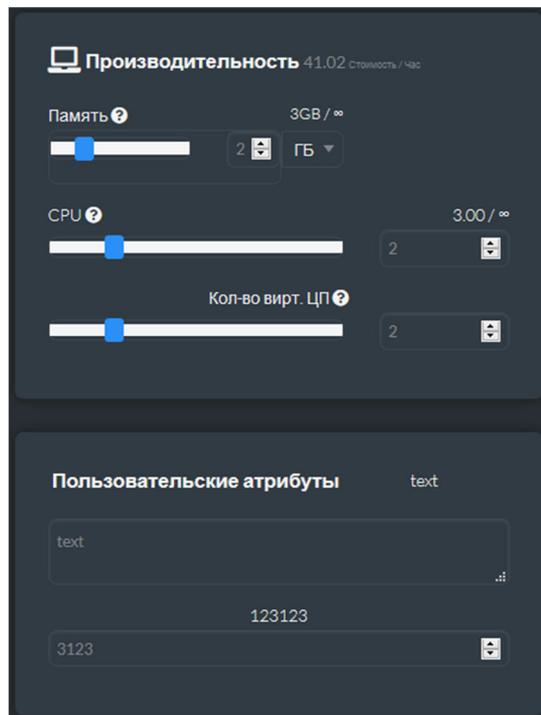


Рис. 5.12 – Пример диалога для уточнения объемных характеристик запускаемой ВМ и ввода пользовательских параметров

Для ввода в шаблон описания параметров, запрашиваемых у пользователя при запуске VM, служит параметр `--user-inputs` команды `tivtemplate create`.

В шаблон VM может быть включен *план действий*, которые должна выполнить VM после ее запуска. Редактор плана отображается на вкладке "Действия" (рис. 5.13).

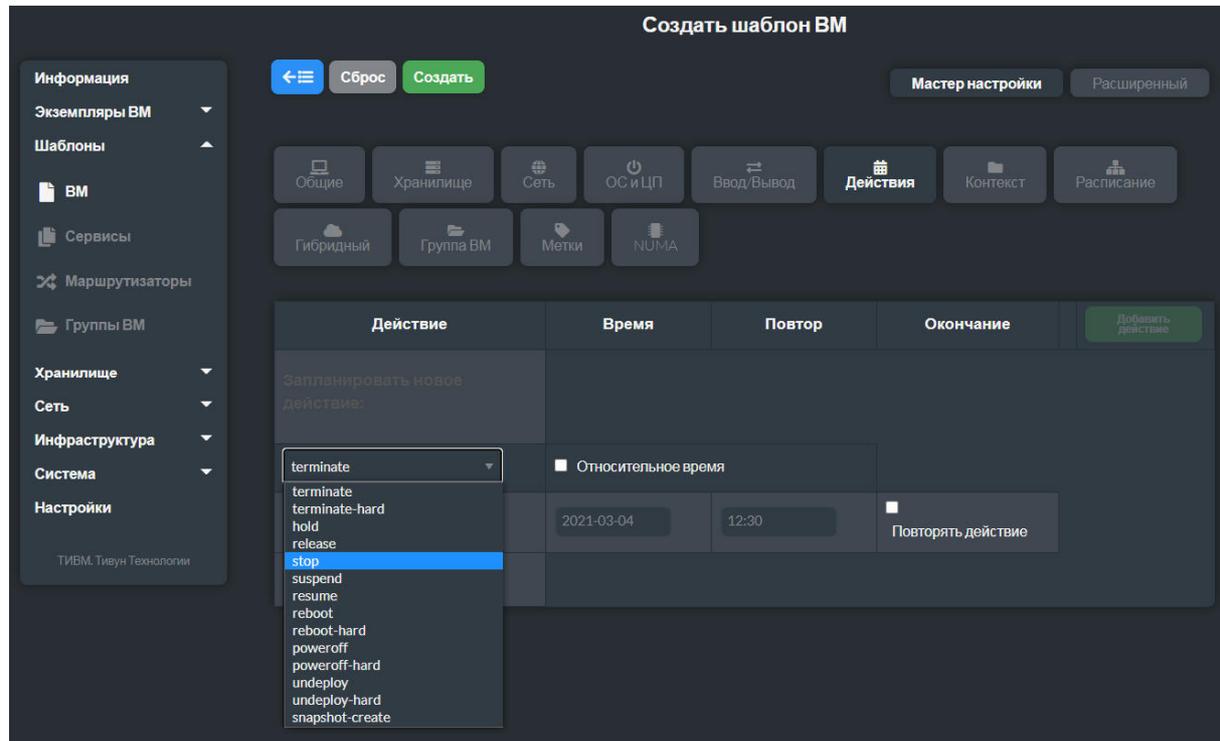


Рис. 5.13 – Редактор плана действий VM, задаваемого в ее шаблоне

В шаблоне может быть установлена *стоимость потребления вычислительных ресурсов* VM в час (1 процессор, 1 мегабайт оперативной памяти, 1 гигабайт хранилища данных). Эти параметры вводятся на вкладке общих атрибутов шаблона VM (рис. 5.6). Использование параметров стоимости описано в разд. 3.8.

В представлении *cloud* web-интерфейса TIVM, рассчитанном на конечного пользователя, предусмотрены возможности, активируемые при их указании в шаблоне VM. К их числу относятся:

- отображение логотипа и описания шаблона;
- доступ к консоли VM через VNC;
- загрузка открытого ssh-ключа пользователя. Для этого требуется, чтобы гостевая ОС, определенная в шаблоне, была контекстуализирована (т.е. настроена на получение данных при запуске VM и выполнение установленных действий), а шаблон должен иметь включенную контекстуализацию ssh (рис. 5.14).

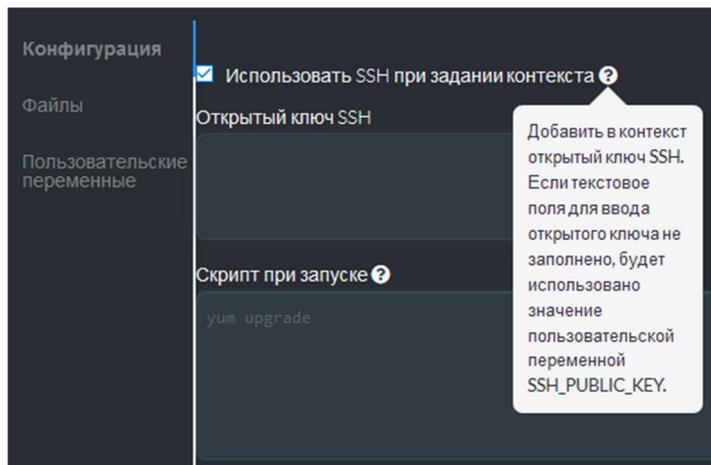


Рис. 5.14 – Параметр загрузки открытого ssh-ключа пользователя

Если шаблон предназначен для применения различными конечными пользователями, то задействуемые в нем образы *не должны быть постоянными*. Постоянный образ может использоваться только на одной ВМ, т.к. все изменения в нем, сделанные данной ВМ, будут доступны другим ВМ, создаваемым на основе этого образа.

Конечные пользователи могут *выбирать NIC при запуске ВМ*. Шаблон может не содержать определения NIC или устанавливать модель сетевого интерфейса по умолчанию. Даже если в шаблоне указан NIC, пользователи по-прежнему могут удалить его и выбрать другой NIC.

Поскольку пользователи будут добавлять сетевые интерфейсы, в шаблоне нужно определить модель NIC по умолчанию на тот случай, если гостевой ОС требуется конкретный интерфейс. Это можно сделать в файле описания шаблона с помощью атрибута NIC_DEFAULT, например:

```
NIC_DEFAULT = [ MODEL = "virtio" ]
```

Также модель NIC по умолчанию может указать в редакторе шаблона (вкладка "Сеть", параметр "Модель оборудования для эмуляции всех сетевых карт"), рис. 5.15. Альтернативный способ – задать значение по умолчанию для всех ВМ в файле конфигурации драйвера виртуализации (для данного типа гипервизора).

Возможность изменения сетевых настроек конечным пользователем может быть отключена в шаблоне. При установке параметра "Не разрешать изменять конфигурацию сети" (рис. 5.16) на вкладке "Общие" редактора шаблона пользователи, запускающие ВМ на его основе, не смогут добавлять, удалять или настраивать NIC, установленные владельцем шаблона.

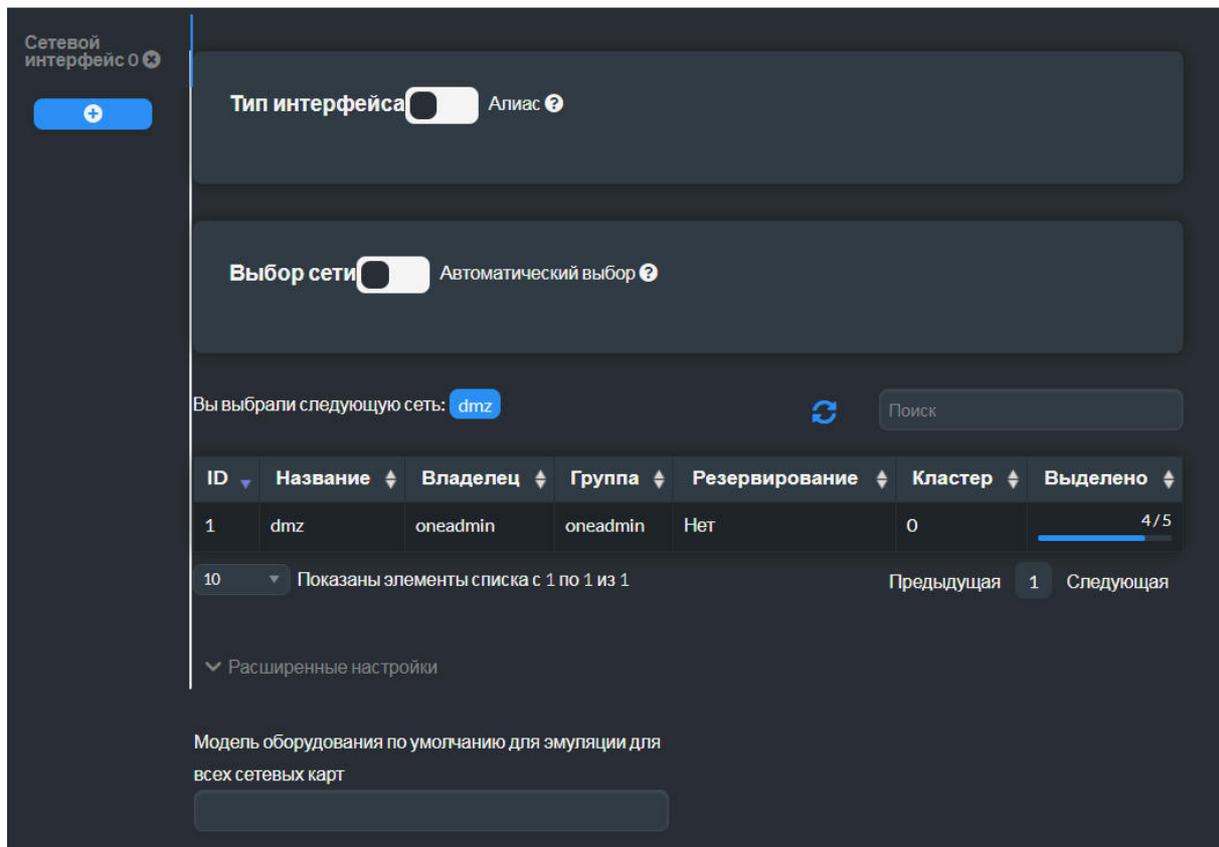


Рис. 5.15 – Параметр определения сетевого интерфейса по умолчанию

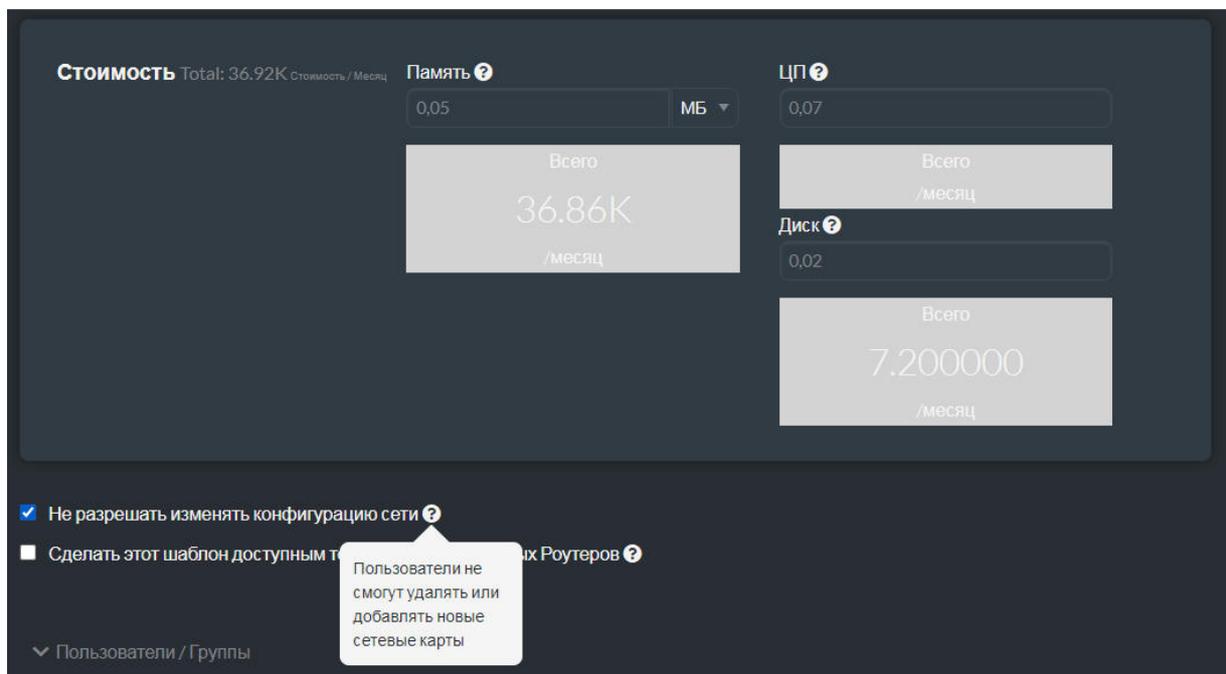


Рис. 5.16 – Параметр, запрещающие изменение сетевых настроек

5.2.2 Создание виртуальной машины на основе шаблона

VM на основе шаблона может быть создана как в web-интерфейсе TIVM, так и с помощью CLI.

В первом случае для создания VM необходимо:

- в списке доступных шаблонов (меню Шаблоны/VM) выбрать нужный;
- на странице его описания нажать кнопку "Создать VM" (рис. 5.17);
- на странице создания VM (рис. 5.18) ввести ее параметры и нажать кнопку "Создать VM".

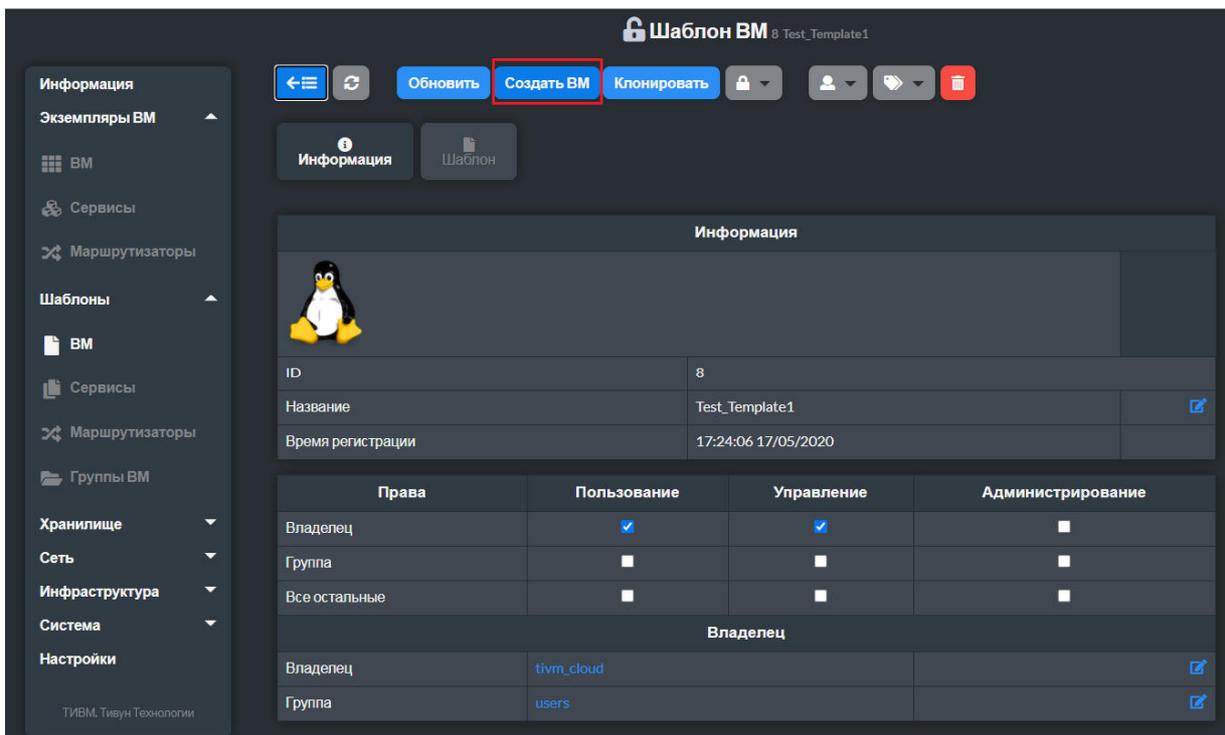


Рис. 5.17 – Кнопка создания VM на основе шаблона

В CLI для создания VM на основе шаблона предназначена команда **tivtemplate instantiate**:

```
tivtemplate instantiate templateIDorName [file] [options]
```

Аргументы команды:

- *templateIDorName* – имя или идентификатор зарегистрированного шаблона;
- *file* – необязательное имя файла шаблона, параметры которого дополняют или переопределяют выбранный шаблон.

Необязательные параметры (секция [*options*]) позволяют задавать атрибуты VM, дополняющие и переопределяющие выбранный шаблон. Их можно использовать вместо дополнительного файла, на который ссылается аргумент *file*.

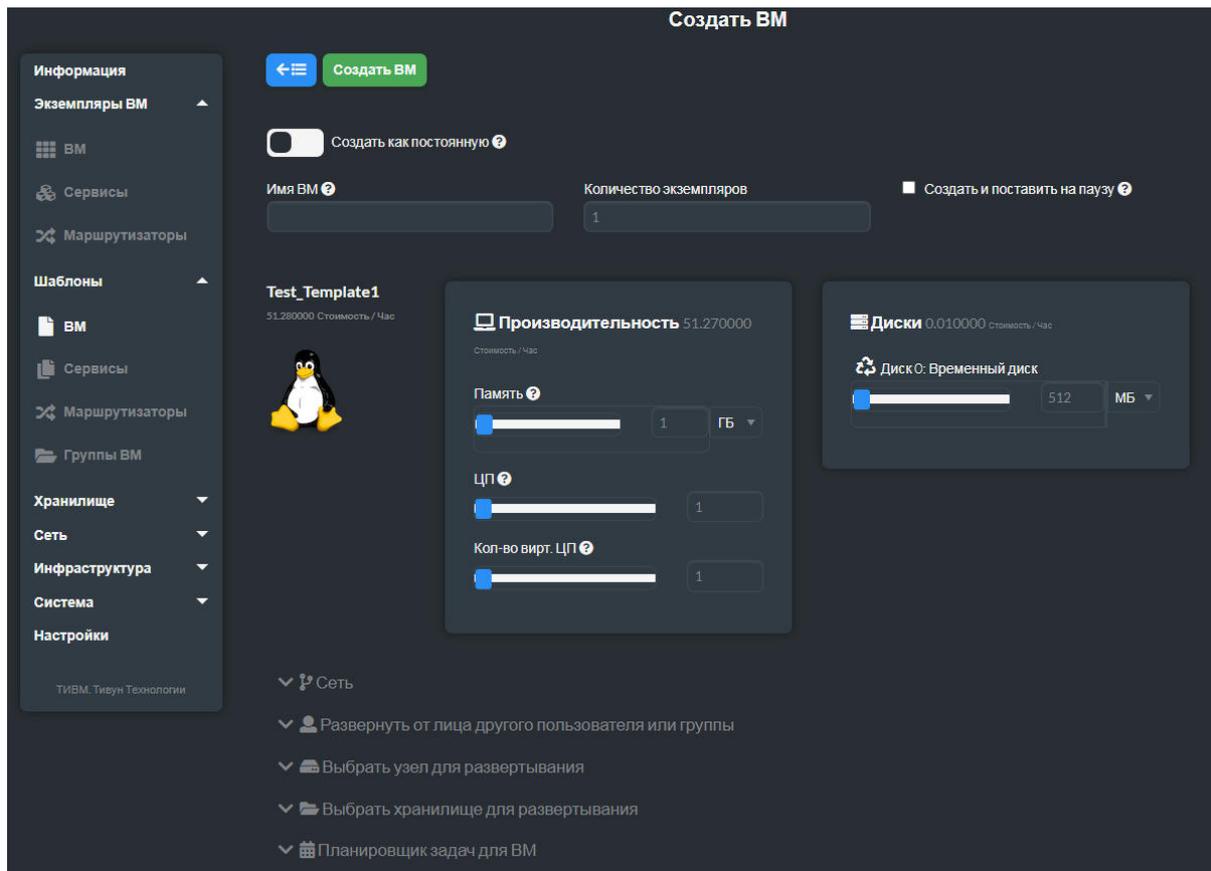


Рис. 5.18 – Страница создания VM

Параметр `--multiple` определяет число экземпляров создаваемых VM.

Созданные VM переходят в состояние ожидания (`pending`). Далее они управляются командой `tivvm` (см. разд. 5.3).

Пример:

```
tivtemplate instantiate 5
VM ID: 0
tivvm list
ID USER      GROUP  NAME  STAT  CPU  MEM  HOSTNAME  TIME
0  tivuser1  users  vm-0  pend  0    0K           00:01:26
```

Возможности пользователя задавать параметры VM, *переопределяющие* те, что установлены в шаблоне, на основе которого она создается, могут быть ограничены путем указания в файле `tiv.conf` *фиксированных параметров*. Они задаются с помощью атрибута `VM_RESTRICTED_ATTR`. Например, пусть `tiv.conf` содержит:

```
VM_RESTRICTED_ATTR = "CPU"
VM_RESTRICTED_ATTR = "VPU"
VM_RESTRICTED_ATTR = "NIC"
```

Также пусть имеется следующий шаблон, зарегистрированный в системе с идентификатором 1:

```
CPU = "1"  
VCPU = "1"  
ПАМЯТЬ = "512"  
DISK = [  
    IMAGE_ID = "0"]  
NIC = [  
    NETWORK_ID = "0"]
```

Пользователи могут создавать ВМ по этому шаблону, устанавливая значения любых параметров кроме CPU, VCPU и NIC. Например, команда создания ВМ, содержащая объем памяти и ссылку на образ диска:

```
tivtemplate instantiate 1 --memory 1G --disk "Ubuntu 17.1"
```

Следует иметь в виду, что параметры ВМ, заданные пользователем при создании ВМ, полностью замещают соответствующие параметры из шаблона. В приведенном выше примере шаблон ссылается на образ диска с идентификатором 0. Этот образ не будет включен в ВМ, т.к. образы переопределены в `tivtemplate` параметром `--disk`. Чтобы создать ВМ с двумя дисками (образом Ubuntu 17.1 и образом, указанным в шаблоне), необходимо сослаться на оба образа в `tivtemplate`:

```
tivtemplate instantiate 1 --memory 1G --disk 0, "Ubuntu 17.1"
```

Планировщик TIVM автоматически развернет созданную ВМ на одном из доступных хостов, удовлетворяющих требованиям, определенным в шаблоне. *Развертывание* может быть вызвано принудительно командой `tivvm deploy`.

Для выключения и удаления ВМ служит команда `tivvm terminate`.

Подробное описание команд управления ВМ приведено в разд. 5.3.

По умолчанию создаваемая на основе шаблона ВМ принадлежит пользователю, который ввел соответствующую команду, и его группе. Система позволяет создать ВМ, *устанавливая в качестве владельца другого пользователя и группу*. На странице запуска ВМ для этого служит группа параметров "Развернуть от лица другого пользователя или группы" (рис. 5.19). Соответствующие диалоговые средства позволяют выбрать пользователя-владельца и группу.

Команда `tivtemplate instantiate` имеет параметры `--as_uid` и `--as_gid`, позволяющие указывать идентификаторы пользователя-владельца и группы. Например:

```
tivtemplate instantiate 5 --as_uid 2 --as_gid 1  
VM ID: 0  
tivvm list
```

ID	USER	GROUP	NAME	STAT	CPU	MEM	HOSTNAME	TIME
0	test_user	users	vm-0	pend	0	0K		00:01:26

Сеть

Развернуть от лица другого пользователя или группы

Вы выбрали следующего пользователя: tivm_user Поиск

ID	Название	Группа	Драйвер авторизации	VM	Память	ЦП
4	ab	oneadmin	core	1/-	128MB / -	1/-
3	tivm_cloud	users	core	3/10	3GB / 4GB	3/6
2	tivm_user	users	core	0/-	0KB / -	0/-
1	serveradmin	oneadmin	server_cipher	0/-	0KB / -	0/-
0	oneadmin	oneadmin	core	-	-	-

10 Показаны элементы списка с 1 по 5 из 5 Предыдущая 1 Следующая

Вы выбрали следующую Группу: tivm_users_group Поиск

ID	Название	Пользователи	VM	Память	ЦП
100	tivm_users_group	2	0/-	0KB / -	0/-
1	users	2	3/-	3GB / -	3/-
0	oneadmin	3	-	-	-

10 Показаны элементы списка с 1 по 3 из 3 Предыдущая 1 Следующая

Рис. 5.19 – Страница создания VM: средства выбора пользователя-владельца и группы

5.2.3 Управление шаблонами виртуальных машин

Пользователи могут управлять шаблонами VM с помощью команды `tivtemplate CLI` и средств web-интерфейса TIVM. Множество шаблонов, доступных конкретному пользователю, определяется их принадлежностью (пользователями и группами, которым они приписаны) и разрешениями на операции с ними.

Для *создания* шаблонов предназначена команда `tivtemplate create`. Она рассмотрена в п. 5.2.1.

Пользователь может создавать шаблоны как для себя, так и множества других пользователей. Администратор системы обычно заранее формирует набор шаблонов в расчете на потребности конечных пользователей.

Для *удаления* шаблонов ВМ предназначена команда `tivtemplate delete`. Удалить шаблон может либо его владелец, либо администратор системы.

Часто бывает необходимо создать шаблон на основе имеющегося, чтобы затем внести в него уточнения, не меняя исходный шаблон. Для реализации данной потребности служит команда *копирования шаблона* `tivtemplate clone`:

```
tivtemplate clone templateIDorName newTemplateName
[--recursive]
```

Ее аргументы:

- *templateIDorName* – имя или идентификатор исходного шаблона;
- *newTemplateName* – имя создаваемой копии шаблона.

Команда возвращает идентификатор созданного шаблона. *Например:*

```
tivtemplate clone 6 new_template
ID: 7
```

При указании параметра `--recursive` команда также копирует каждый образ, используемый в шаблоне. Создаваемые копии образов получают признак постоянных, и в созданном шаблоне атрибуты `DISK/IMAGE_ID` указывают на них. В web-интерфейсе TIVM данный вариант копирования выполняется при нажатии кнопки "Клонировать из образа" (рис. 5.20).

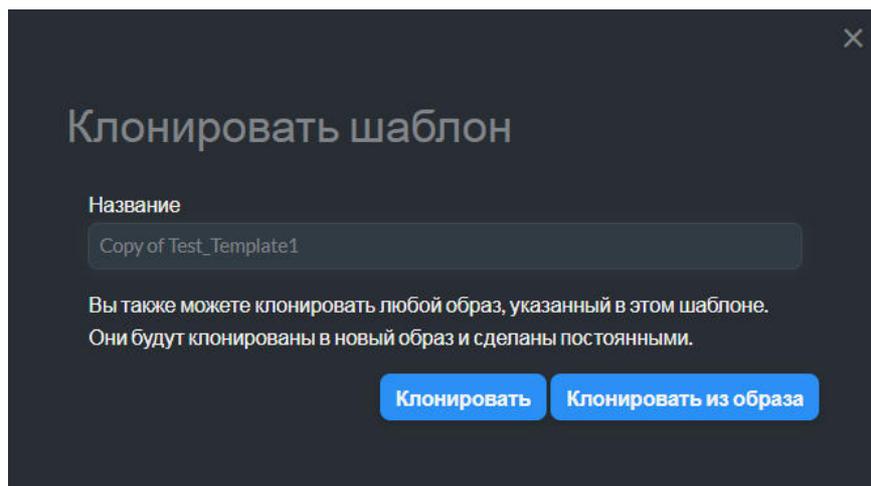


Рис. 5.20 – Диалог копирования шаблона ВМ

Для *изменения* шаблона предназначена команда `tivtemplate update`:

```
tivtemplate update templateIDorName [file] [--append]
```

Ее аргументы:

- *templateIDorName* – имя или идентификатор исходного шаблона;
- *file* – имя файла описания шаблона, содержащего устанавливаемые атрибуты. Если аргумент отсутствует, то TIVM открывает редактор текущего шаблона.

Параметр `--append` добавляет в шаблон указанный после него атрибут со значением.

Редактор шаблона в web-интерфейсе TIVM открывается при нажатии кнопки "Обновить" страницы шаблона (рис. 5.21).

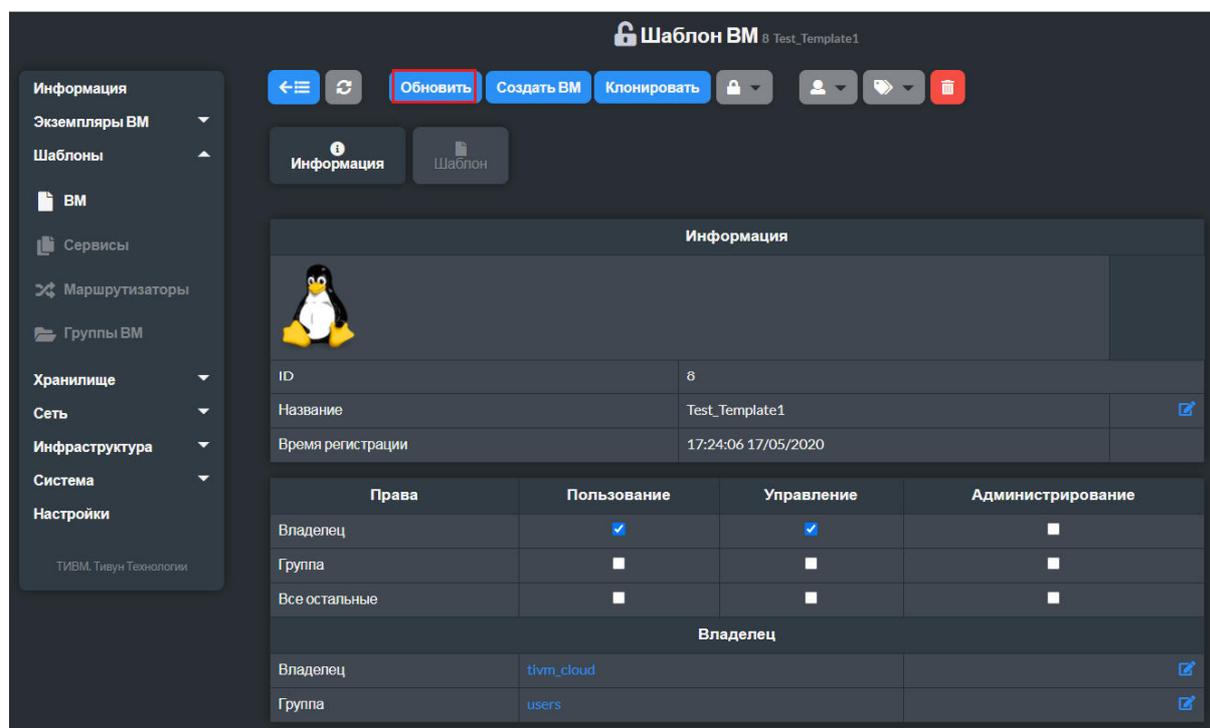


Рис. 5.21 – Кнопка открытия редактора шаблона

Пользователь может объявлять свои шаблоны, как разделяемые ресурсы, предоставляя *разрешения на операции* с ними для других пользователей. Для установки разрешений предназначена команда `tivtemplate chmod`.

Пример. Имеется шаблон с идентификатором 1, и необходимо предоставить права на операции категории USE пользователям группы, в которую входит владелец шаблона. Это обеспечивает октет 640 (4 в 2-м разряде означает разрешение операций USE для группы шаблона):

```
tivtemplate show 1  
...  
PERMISSIONS
```

```
OWNER          : um-
GROUP          : ---
OTHER          : ---
tivtemplate chmod 1 640
tivtemplate show 1
...
PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---
```

Следующая команда предоставляет пользователям, относящимся к группе шаблона, разрешения на операции USE и MANAGE, а прочим пользователям – разрешения на операции USE:

```
tivtemplate chmod 1 664
tivtemplate show 1
...
PERMISSIONS
OWNER          : um-
GROUP          : um-
OTHER          : u--
```

Параметр `--recursive` распространяет действие команды `tivtemplate chmod` также на каждый из образов, используемых в шаблоне.

В web-интерфейсе TIVM на странице описания шаблона предусмотрено меню для управления правами доступа к нему. Пункт "Дать доступ" (рис. 5.22) реализует вызов команды `tivtemplate chmod --recursive 640`.

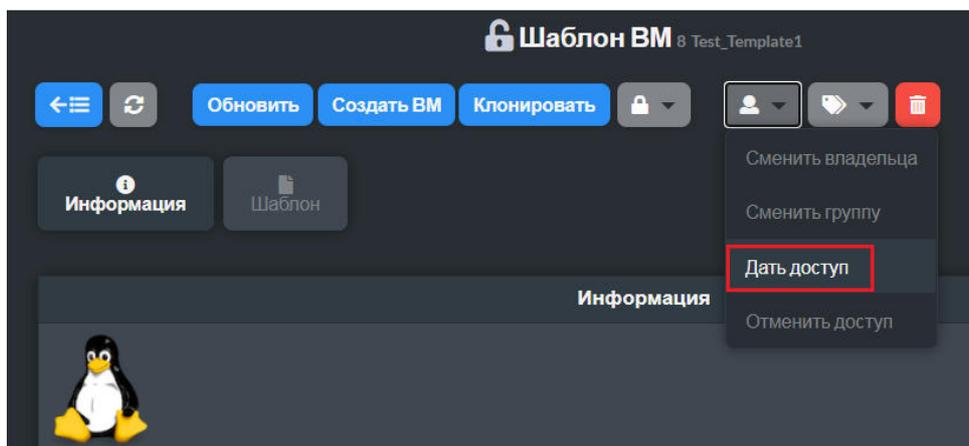


Рис. 5.22 – Команда установки разрешений на операции с шаблоном, вызываемая из web-интерфейса TIVM

5.3 Управление экземплярами виртуальных машин

После создания VM на основе шаблона (как описано в п. 5.2.2) TIVM предоставляет средства для управления ею. В CLI для управления VM служит группа команд `tivvm`.

5.3.1 Жизненный цикл виртуальной машины

Этапы ЖЦ VM представлены в табл. 5.3.

Таблица 5.3 – ЖЦ VM

Краткое обозначение состояния	Обозначение состояние	Значение
pend	Pending	Ожидание выделения ресурсов для развертывания. По умолчанию VM переходит к это состояние после ее создания на основе шаблона, ожидая выделения необходимых вычислительных ресурсов. VM будет оставаться в этом состоянии до тех пор, пока она не будет развернута по решению планировщика или команде пользователя <code>tivvm deploy</code> .
hold	Hold	Задержка развертывания VM. Владелец удерживает VM машину от развертывания. Она не может быть развернута планировщиком, но ее можно развернуть вручную.
clon	Cloning	Копирование. VM ожидает завершения исходного копирования образов дисков в хранилище данных (в ходе копирования эти образы находятся в состоянии <code>lock</code> , т.е. заблокированы).
prol	Prolog	Подготовка развертывания. Система передает файлы VM (образы дисков и файл восстановления) на хост, на котором она разворачивается.
boot	Boot	Загрузка. Гипервизор запускает VM.
runn	Running	VM работает. Данный этап включает, в том числе, загрузку ОС и выключение VM. В этом состоянии драйвер виртуализации периодически отслеживает функционирование VM.
migr	Migrate	VM переносится с одного ресурса на другой. Это может быть миграция работающей VM или "холодная миграция" (образы дисков VM сохраняются, VM выключается, затем файлы VM переносятся на другой хост, где она запускается).
hotp	Hotplug	Выполняется процесс присоединения или отсоединения диска либо NIC.
snap	Snapshot	Создается снимок VM.
save	Save	Система сохраняет файлы VM в рамках операций миграции, остановки или приостановки работы.

Краткое обозначение состояния	Обозначение состояния	Значение
epil	Epilog	На этом этапе система очищает узел, используемый для виртуализации ВМ. Кроме того, образы дисков, которые необходимо сохранить, копируются обратно в системное хранилище данных.
shut	Shutdown	ВМ выключается. TIVM отправил ВМ сигнал ACPI выключения и ждет завершения процесса выключения. Если по истечении периода ожидания ВМ не отключится, то TIVM будет считать, что гостевая ОС проигнорировала сигнал ACPI, и состояние ВМ будет изменено на <code>runn</code> , а не <code>done</code> .
stop	Stopped	ВМ остановлена. Состояние ВМ было сохранено и передано вместе с образами дисков в системное хранилище данных.
susp	Suspended	ВМ приостановлена. То же, что остановлено (<code>stop</code>), но файлы образов остаются на хосте, где развернута ВМ, для последующего быстрого возобновления ее работы (т.е. для при последующем запуске ВМ нет необходимости обращаться к планировщику для выделения для нее вычислительных ресурсов).
poff	PowerOff	ВМ выключена. То же, что приостановлено (<code>susp</code>), но файл контрольных точек не создается. Файлы ВМ остаются на хосте, где она развернута, для последующего быстрого возобновления ее работы. После завершения работы гостевой ОС TIVM переводит ВМ в это состояние.
unde	Undeployed	ВМ отключена, ее развертывание ВМ отменено (выделенные вычислительные ресурсы возвращены планировщику). Диски ВМ переданы в системное хранилище данных. Работа ВМ может быть возобновлена, но это потребует ее нового развертывания.
fail	Failed	Состояние ошибки ВМ.
unkn	Unknown	Не удалось связаться с ВМ, она находится в неизвестном состоянии.
clea	Cleanup-resubmit	ВМ ожидает, когда драйверы очистят хост после команды <code>tivvm recover --recreate</code> .
done	Done	ЖЦ ВМ завершен. ВМ в этом состоянии не включаются в список, возвращаемый командой <code>tivvm list</code> , но информация о них сохраняется в БД для целей учета. Информация о ВМ может быть получена с помощью команды <code>tivvm show</code> .

Диаграмма состояний ВМ представлена на рис. 5.23.

В п. 5.2.2 представлен основной способ создания ВМ – формирование ее экземпляра на основе шаблона с помощью команды `tivtemplate instantiate`. Альтернативный вариант создания ВМ предоставляет команда `tivvm create`:

```
tivvm create [file] [options]
```

Ее единственный необязательный аргумент *file* задает путь к файлу шаблона. Если он не указан, то данные для создания ВМ берутся из параметров команды (секция *options*):

- `--name nameVM` – имя ВМ;
- `--memory value` – объем памяти. По умолчанию значение выражается в мегабайтах. Для указания гигабайтов значение должно заканчиваться на "g";
- `--cpu value` – число процессоров;
- `--disk value` – имя образа диска. При использовании множества дисков указывается несколько значений через запятую. После имени образа через символ ":" могут быть приведены его параметры, применяемые в данной ВМ. Они выражаются в формате `key=value` и разделяются двоеточиями, например: `--disk image1:size=1000g:target=vda,image2`. Ссылка на образ, принадлежащий другому пользователю, указывается в формате `user_name[disk]`;
- `--nic value` – имя сетевого интерфейса. При использовании множества NIC указывается несколько значений через запятую. Формат представления атрибутов NIC аналогичен параметру `--disk`. Ссылка на NIC, принадлежащий другому пользователю, указывается в формате `user_name[nic]`;
- `--hold` – создать ВМ в состоянии `hold` (по умолчанию ВМ получает состояние `pend`);
- `--arch value` – архитектура ВМ (`i386`, `x86_64`);
- `--as_gid gid` – идентификатор группы, приписываемой ВМ;
- `--as_uid uid` – идентификатор пользователя-владельца ВМ;
- `--boot deviceList` – список имен загрузочных устройств ВМ, разделенных запятыми (например, `--boot disk1,disk2,disk3`);
- `-m`, `--multiple n` – число создаваемых ВМ;
- `--context line` – строка, добавляемая в контекст ВМ. Значение может представлять несколько строк, разделяемых запятыми;
- `--init script` – скрипт для запуска в контексте ВМ. Значение может представлять несколько строк, разделяемых запятыми;
- `--raw string_value` – добавить в шаблон произвольную строку `string_value`;
- `--dry` – вывести представление шаблона на экран (без выполнения иных действий).

5.3.2 Получение информации о ВМ

Пусть в системе зарегистрирован шаблон ВМ с именем `vm-example`. Создание на его основе экземпляра ВМ обеспечивает команда `tivtemplate instantiate`:

```
tivtemplate instantiate vm-example --name my_vm
VM ID: 2
```

В данном примере создана ВМ с именем `my_vm`, получившая идентификатор 2.

Если в шаблоне определены параметры, запрашиваемые у пользователя, то TIVM выведет запрос для их получения:

```
tivtemplate instantiate vm-example --name my_vm
There are some parameters that require user input.
* (TITLE) Title: value
* (DB PASSWORD) Database Password:
VM ID: 2
```

Команда `tivvm list` возвращает список зарегистрированных ВМ. Экземпляр ВМ включается в данный список после его создания.

```
tivvm list
ID USER      GROUP  NAME    STAT  CPU  MEM  HOSTNAME  TIME
2 tivadmin  tivadmin my_vm   pend  0    OK                   00 00:00:05
```

Представление на экране обновляемого списка зарегистрированных ВМ обеспечивает команда `tivvm top`.

После того, как планировщик выделит ВМ необходимые вычислительные ресурсы, она будет автоматически развернута.

При необходимости срочного (внеочередного) развертывания ВМ данная операция вызывается администратором с помощью команды `tivvm deploy`:

```
tivvm deploy 2 1
tivvm list
  ID USER      GROUP  NAME    STAT  CPU  MEM  HOSTNAME  TIME
  2 tivadmin  tivadmin my_vm   runn  0    OK    testhost 00 00:12:21
```

Команда `tivvm deploy` имеет 3 аргумента. Первый (обязательный) – диапазон или список идентификаторов разворачиваемых ВМ, второй (обязательный) – имя или идентификатор хоста виртуализации, на котором разворачивается ВМ, третий (необязательный) – имя или идентификатор используемого хранилища данных. Команда применима по отношению к ВМ в состояниях `pend`, `hold`, `stop` и `unde`.

Для получения информации о ВМ служит команда `tivvm show`:

```

tivvm show 0
VIRTUAL MACHINE 2 INFORMATION
ID                : 2
NAME              : my_vm
USER              : tivadmin
GROUP             : tivadmin
STATE             : ACTIVE
LCM_STATE         : RUNNING
START TIME        : 04/27 12:02:29
END TIME          : -
DEPLOY ID:        : tiv-2
PERMISSIONS
OWNER             : um-
GROUP             : ---
OTHER             : ---

VIRTUAL MACHINE MONITORING
NET_TX            : 13.05
NET_RX            : 0
USED MEMORY       : 512
USED CPU          : 0

VIRTUAL MACHINE TEMPLATE
...

VIRTUAL MACHINE HISTORY
SEQ      HOSTNAME  REASON          START      TIME      PTIME
  0      testhost  none           04/27 12:44:01 00 00:07:23 00 00:00:00

```

Команда `tivvm list` позволяет проводить *поиск* зарегистрированных ВМ. Возможности поиска полезны в больших виртуальных средах, содержащих множество экземпляров ВМ.

Для проведения поиска служит параметр `--search`, за которым указывается массив условий (*cond1, cond2, ...*). Каждое условие задает тройка <имя атрибута ВМ, предикат, значение>. Набор атрибутов: UNAME, GNAME, NAME, LAST_POLL, PREV_STATE, PREV_LCM_STATE, RESCHED, STIME, ETIME, DEPLOY_ID. Набор предикатов: =, != (не равно), <, >, <=, >=, ~ (начинается с). Условия соединяются оператором, указанным в параметре `--operator`. Набор операторов: AND, OR. Оператор по умолчанию – AND.

Пример. Поиск ВМ по имени "test-vm":

```

tivvm list --search NAME=test-vm
ID   USER      GROUP      NAME      STAT  UCPU  UMEM  HOST  TIME
21005 tivadmin  tivadmin  test-vm  pend  0     0K    1d 23h13
2100  tivadmin  tivadmin  test-vm  pend  0     0K    12d 17h59

```

5.3.3 Завершение работы и удаление VM

Для *прекращения работы VM и ее удаления* предназначена команда `tivvm terminate`. Завершение работы может быть запрошено по отношению к VM в любом состоянии. Выполнение команды влечет отключение VM, ее удаление и освобождение выделенных ей вычислительных ресурсов (образов дисков, виртуальных сетей и др.). VM переводится в терминальное состояние `done`.

По умолчанию при выполнении команды TIVM направляет VM сигнал выключения ACPI, который должен быть обработан гостевой ОС. После выключения VM хост, на котором она была развернута, очищается, а содержимое ее дисков, основанных на постоянных образах, и отложенные снимки их состояния, переносятся в соответствующее хранилище данных. Если по истечении определенного времени VM все еще работает (например, гостевая ОС игнорирует сигнал ACPI), TIVM вернет ее в состояние `runn`.

Команда `tivvm terminate` с параметром `--hard` вызывает "жесткое" отключение VM. В этом случае TIVM не ждет выключения VM и сразу освобождает ее вычислительные ресурсы. Этот вариант следует использовать, если VM не поддерживает ACPI.

Пример. Прекращение работы и удаление VM с идентификатором 2:

```
tivvm terminate 2
```

Вызов команд прекращения работы и удаления VM из веб-интерфейса TIVM показан на рис. 5.24.

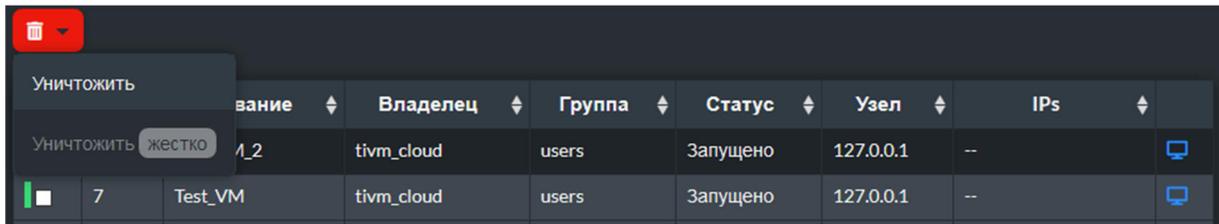


Рис. 5.24 – Вызов команды прекращения работы и удаления VM

5.3.4 Временное прекращение работы VM

Временное прекращение работы VM предполагает, что потребность в VM остается, и через какой-то промежуток времени она должна быть вновь запущена, для чего необходимо обеспечить соответствующие условия. К последним относятся:

- сохранение состояния VM;
- фиксация вычислительных ресурсов, выделенных VM;
- сохранение данных дисков VM.

Чем больше ресурсов, необходимых VM, сохраняется в готовности к быстрому использованию, тем меньше времени потребуется для возобновления работы VM.

Для *прекращения работы VM на относительно короткий период* служат команды:

- **ti`vm` suspend** – приостановка работы VM. Ее состояние сохраняется на хосте, где она развернута. Также фиксируются выделенные VM вычислительные ресурсы. VM переходит в состояние `susp`. При возврате VM в работу, она оперативно запускается на том же хосте с восстановлением ее состояния;

- **ti`vm` poweroff** – выключение VM (путем отправки ей соответствующего сигнала ACPI). За VM закрепляются выделенные ей вычислительные ресурсы, но состояние VM не сохраняется. VM переходит в состояние `poff`. При возврате VM в работу, она оперативно запускается на том же хосте со стартом процедуры загрузки (состояние `boot`).

Команда `tivm poweroff` с параметром `--hard` вызывает "жесткое" выключение VM, при котором TIVM не ждет обработки VM сигнала ACPI. Этот вариант следует использовать, если VM не поддерживает ACPI.

Вызов команд прекращения работы VM на короткий период из web-интерфейса TIVM показан на рис. 5.25. Пункт "Приостановить работу" соответствует команде `tivm suspend`, пункт "Остановить" – команде `tivm poweroff`.

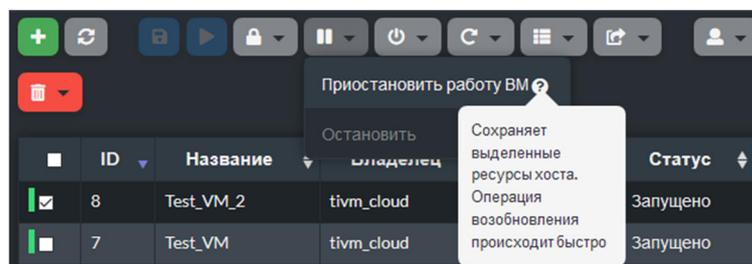


Рис. 5.25 – Вызов команд прекращения работы VM на короткий период

Прекращение работы VM на длительный период предусматривает освобождение выделенных ей вычислительных ресурсов. В рамках данной операции ресурсы хоста, на котором развернута VM, освобождаются, узел очищается, все постоянные диски VM сохраняются в системном хранилище данных. Возврат VM в работу после такой команды вызывает обращение к планировщику (состояние `pend`) для выделения ресурсов и развертывания VM.

Команды прекращения работы VM на длительный период:

- **ti`vm` stop** – остановка VM (путем отправки ей сигнала выключения ACPI), сохранение ее данных и состояния, освобождение выделенных

вычислительных ресурсов. VM переходит в состояние `stop`. При возобновлении работы VM после ее развертывания восстанавливается ее сохраненное состояние;

- `tivvm undeploy` – то же самое, но без сохранения состояния VM. VM переходит в состояние `unde`.

Команда `tivm undeploy` с параметром `--hard` вызывает "жесткую" остановку VM, при которой TIVM не ждет обработки VM сигнала ACPI. Этот вариант следует использовать, если VM не поддерживает ACPI.

Вызов команд прекращения работы VM на длительный период из web-интерфейса TIVM показан на рис. 5.26. Пункт "Отключить питание" соответствует команде `tivm stop`, пункт "Отключить питание жестко" – команде `tivm stop --hard`, пункт "Отменить размещение" – команде `tivm undeploy`, пункт "Отменить размещение жестко" – команде `tivm undeploy --hard`.

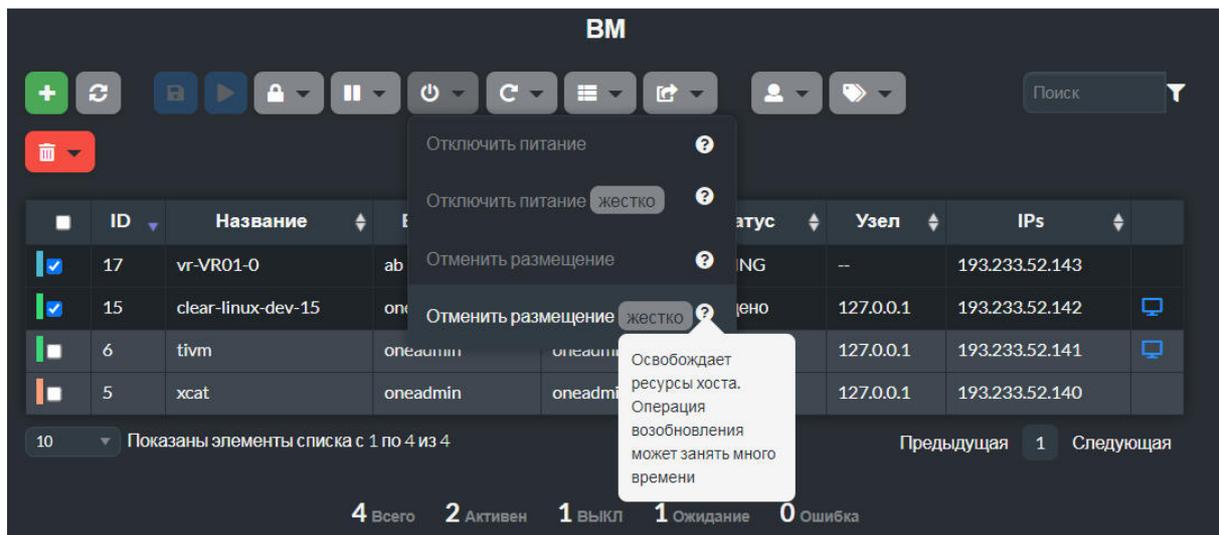


Рис. 5.26 – Вызов команд прекращения работы VM на длительный период

Остановленная VM, находящаяся в состоянии `susp`, `poff`, `stop` или `unde`, может быть возвращена в работу с помощью команды `tivvm resume`. Порядок выполнения данной команды зависит от ее исходного состояния, как описано выше.

5.3.5 Перезагрузка VM

Перезагрузку работающей VM (находящейся в состоянии `runn`) обеспечивает команда `tivvm reboot`.

По умолчанию эта команда направляет VM сигнал перезагрузки ACPI. Вызов `tivvm reboot` с параметром `--hard` инициирует "жесткую" перезагрузку, при которой TIVM не ждет обработки VM сигнала ACPI. Этот вариант следует использовать, если VM не поддерживает ACPI.

Вызов команд перезагрузки VM из web-интерфейса TIVM показан на рис. 5.27. Пункт "Перезагрузить" соответствует команде `tivm reboot`, пункт "Перезагрузить жестко" – команде `tivm reboot --hard`.

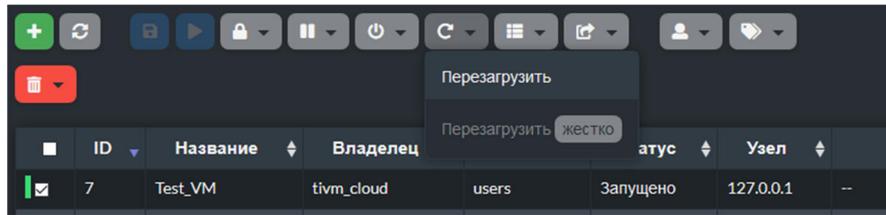


Рис. 5.27 – Вызов команд перезагрузки VM

5.3.6 Задержка развертывания VM

VM в состоянии `pend` ожидают выделения вычислительных ресурсов планировщиком и развертывания. Для *блокировки автоматического развертывания* VM может быть переведена в состояние `hold` с помощью команды `tivvm hold`. Команда применима к одной или множеству VM.

Для возврата VM из состояния `hold` в состояние ожидания автоматического развертывания предназначена команда `tivvm release`.

VM в состоянии `hold` исключаются из автоматической обработки планировщиком. Однако такие VM могут быть развернуты по команде `tivvm deploy`, введенной вручную администратором.

Вызов команд перевода VM между состояниями ожидания автоматического развертывания и задержки развертывания из web-интерфейса TIVM показан на рис. 5.28. Пункт "Запретить развертывание" соответствует команде `tivm hold`, пункт "Разрешить размещение" – команде `tivm release`, пункт "Разместить на узле" – команде `tivm deploy`.

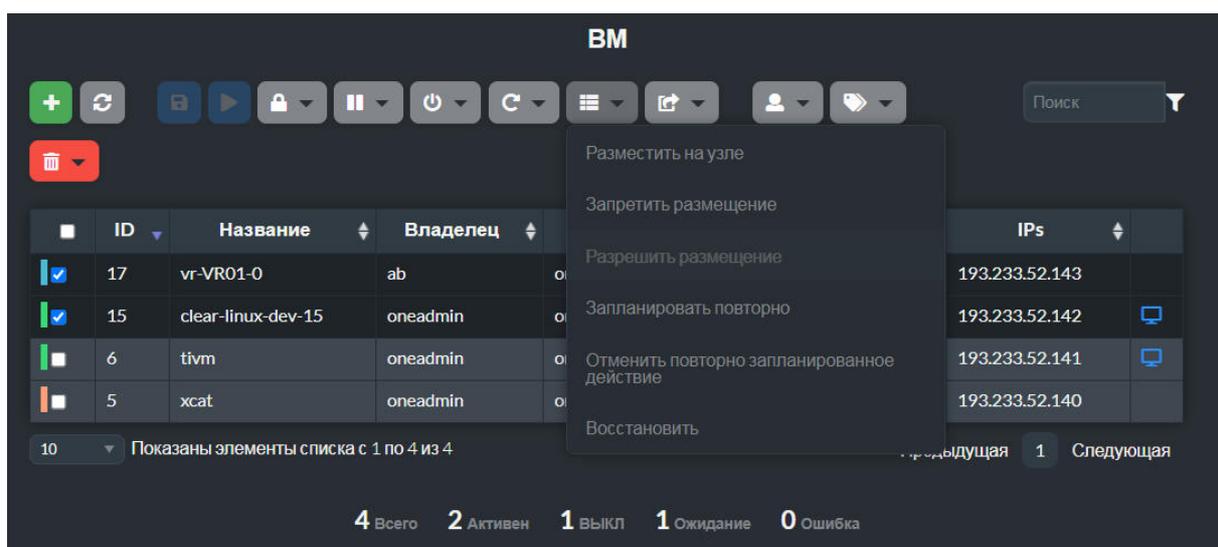


Рис. 5.28 – Вызов команд задержки развертывания

ВМ может быть изначально создана в режиме задержки развертывания `hold`. Эту возможность обеспечивают:

- команда `tivtemplate instantiate` с параметром `--hold`;
- команда `tivvm create` с параметром `--hold`.

5.3.7 Управление снимками состояния образов дисков ВМ

Для управления снимками состояния образов дисков ВМ предназначены команды `tivvm disk-snapshot-create`, `tivvm disk-snapshot-revert`, `tivvm disk-snapshot-delete`, `tivvm disk-snapshot-rename` и `tivvm disk-saveas`. Операции со снимками состояния образов дисков не поддерживаются в vCenter.

Снимок текущего состояния образа диска ВМ, находящейся в состоянии `runn`, `poff` или `susp`, *создает* команда **`tivvm disk-snapshot-create`**:

```
tivvm disk-snapshot-create VmId DiskId Name
```

Аргументы команды:

- *VmId* – имя или идентификатор ВМ;
- *DiskId* – идентификатор диска ВМ;
- *Name* – имя создаваемого снимка.

Создание снимка для ВМ в состоянии `runn` возможно, если эту операцию поддерживают драйверы. Такую поддержку обеспечивают:

- гипервизор KVM (`VM_MAD=kvm`) в сочетании с хранилищами данных Qcow2 (`TM_MAD=qcow2`);
- гипервизор KVM (`VM_MAD=kvm`) в сочетании с хранилищами данных Ceph (`TM_MAD=ceph`).

Если возможности создания снимков состояния образов дисков работающей ВМ отсутствует, то перед их снятием необходимо приостановить функционирование ВМ (перевести ее в состояние `susp` или `poff`).

Пользователь может *вернуть образ к ранее созданному снимку* с помощью команды **`tivvm disk-snapshot-revert`**:

```
tivvm disk-snapshot-revert VmId DiskId SnapshotId
```

Аргумент *SnapshotId* задает идентификатор снимка, к которому следует вернуться. Возврат образа диска ВМ к данному снимку состояния возможен, если ВМ находится в состоянии `susp` или `poff`.

Снимки образуют иерархическую структуру. Каждый снимок имеет родительский за исключением самого первого. Текущий *активный* снимок (тот, к которому был возвращен образ диска, или последний снятый снимок) становится родительским для следующего снимка.

Пример. ВМ с идентификатором 15 имеет диск с идентификатором 0. Для него созданы 2 снимка состояния:

```
tivvm disk-snapshot-create 15 0 sshot15-0-1
tivvm disk-snapshot-create 15 0 sshot15-0-2
```

Информация о созданных снимках возвращается командой `tivvm show`:

```
tivvm show
VIRTUAL MACHINE 15 INFORMATION
ID                : 15
NAME              : clear-linux-dev-15
USER              : tivadmin
GROUP            : tivadmin
STATE            : ACTIVE
LCM_STATE        : RUNNING
LOCK              : None
RESCHED          : No
HOST              : 127.0.0.1
CLUSTER ID       : 0
CLUSTER          : default
START TIME       : 06/29 21:47:21
END TIME         : -
DEPLOY ID        : tiv-15

. . .

VM DISKS
  ID DATASTORE  TARGET IMAGE                SIZE      TYPE  SAVE
  0 sw7000VDI    vda   clear-linux-build    13.9G/128 file  YES
  1 -           hda   CONTEXT              1M/-     -    -

VM DISK SNAPSHOTS
AC  ID DISK PARENT          DATE SIZE           NAME
=>  0  0  -1  04/27 18:53:48 13.9G/128G  sshot15-0-1
    1  0  0  04/27 19:04:39 -/128G     sshot15-0-2

VM NICs
. . .

SECURITY
. . .
```

Страница web-интерфейса TIVM, представляющая снимки состояния образов дисков ВМ, и средства для работы с ними, показана на рис. 5.29.

Команда `tivvm disk-snapshot-rename` предназначена для переименования снимка образа диска ВМ:

```
tivvm disk-snapshot-rename VmId DiskId SnapshotId NewName
```

Аргумент `NewName` задает новое имя снимка с идентификатором `SnapshotId`.

Снимки, не являющиеся активными и не имеющие дочерних, могут быть удалены с помощью команды `tivvm disk-snapshot-delete`:

```
tivvm disk-snapshot-delete VmId DiskId SnapshotId
```

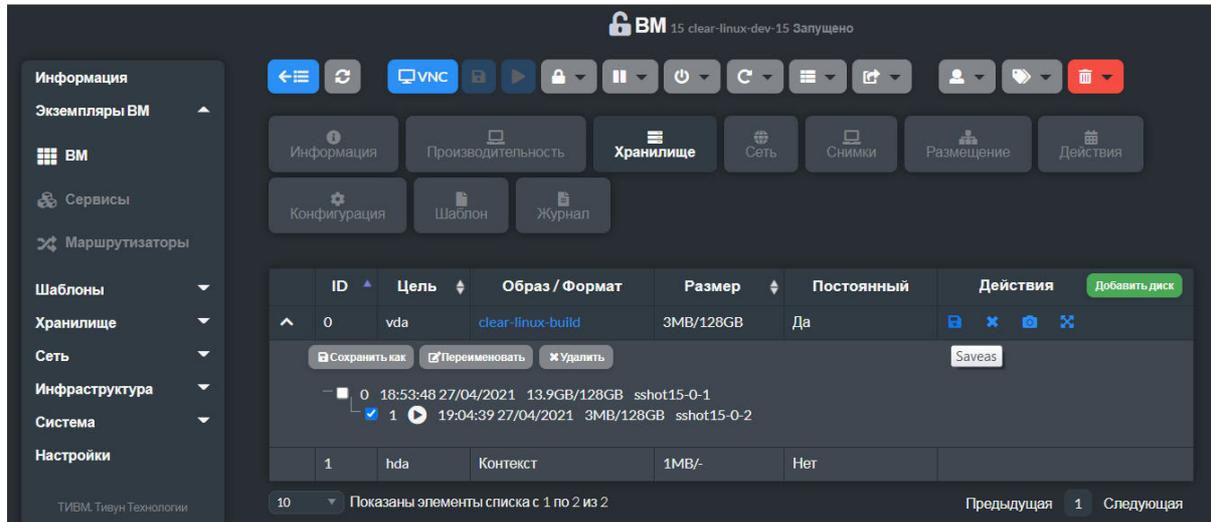


Рис. 5.29 – Страница представления снимков состояния образов дисков VM

Операции создания, удаления и переименования снимка состояния, а также возврата образа к ранее созданному снимку применимы как к постоянным, так и к непостоянным образам. Снимки состояния постоянного образа сохраняются после завершения работы VM и могут использоваться другими VM, в которых будет задействован данный образ.

Операции со снимками состояния образов дисков реализуются по-разному в зависимости от технологии хранения данных (табл. 5.4).

Таблица 5.4 – Особенности реализации операций со снимками состояния образов дисков VM

Операция	Технология хранения данных (драйвер менеджера передачи данных TM MAD)			
	ceph	shared, ssh	qcow2	fs_lvm, dev
Создание снимка	Создание защищенного снимка	Копирование файла	Создание нового образа Qcow2 с предыдущим состоянием диска в качестве резервного файла	Не поддерживается
Создание снимка работающей VM	Создание защищенного снимка с приостановкой	Не поддерживается	В случае использования гипервизора KVM – запуск	Не поддерживается

Операция	Технология хранения данных (драйвер менеджера передачи данных ТМ МАД)			
	ceph	shared, ssh	qcow2	fs lvm, dev
	работы гостевой ОС		команды virsh snapshot-create	
Возврат образа к ранее созданному снимку	Перезапись диска путем создания снимка на основе данного защищенного снимка	Перезапись файла из ранее скопированного файла	Создание нового образа Qcow2 на основе данного снимка, сохраненного как резервный файл	Не поддерживается
Удаление снимка	Удаление защищенного снимка	Удаление файла	Удаление данного снимка qcow2	Не поддерживается

Содержимое диска ВМ, находящейся в состоянии `runn`, `susp` или `poff`, может быть экспортировано в виде образа с помощью команды **tivvm disk-saveas**:

```
tivvm disk-saveas VmId DiskId Name
```

Аргумент *Name* задает имя создаваемого образа.

По умолчанию в виде образа сохраняется текущее состояние диска. Параметр `--snapshot`, за которым указан идентификатор снимка состояния, позволяет выбрать снимок, используемый в качестве источника данных вместо текущего состояния.

В vCenter операция экспорта образа диска может быть выполнена только для ВМ в состоянии `poff`.

5.3.8 Подключение и отключение диска у работающей ВМ

Для подключения к ВМ, находящейся в состоянии `runn`, `susp` или `poff`, нового диска предназначена команда **tivvm disk-attach**. Например, следующая команда подключает к ВМ с именем `vm-15` диск на основе образа с именем `storage1`:

```
tivvm disk-attach vm-15 --image storage1
```

Единственный аргумент команды – имя или идентификатор ВМ. Параметр `--image` задает имя образа диска.

Для отключения диска служит команда **tivvm disk-detach**. Она имеет 2 аргумента:

- имя или идентификатор ВМ;
- идентификатор отключаемого диска.

Команда отключения диска применима к ВМ, находящимся в состояниях `runn` и `poff`. *Пример:*

```
tivvm disk-detach vm-15 1
```

Операции подключения и отключения диска могут быть вызваны из web-интерфейса TIVM со страницы управления ВМ, вкладки "Хранилище" (рис. 5.29).

5.3.9 Подключение и отключение сетевого интерфейса у работающей ВМ

Для подключения NIC к ВМ, находящейся в состоянии `runn`, `susp` или `poff`, предназначена команда `tivvm nic-attach`. Ее единственный аргумент – имя или идентификатор ВМ. Подключаемый NIC определяют параметры:

`--network NameId` – подключить NIC виртуальной сети с именем или идентификатором `NameId`;

`--nic_name Name` – присвоить подключенному NIC имя `Name`;

`--ip IPadr` – использовать IP-адрес `IPadr`;

`--alias AliasName` – подключить NIC через псевдоним `AliasName`.

Например, следующая команда подключает ВМ с идентификатором 15 к виртуальной сети с именем `net172`:

```
tivvm nic-attach 15 --network net172
```

Для отключения NIC служит команда `tivvm nic-detach`. Она имеет 2 аргумента:

- имя или идентификатор ВМ;

- идентификатор отключаемого NIC.

Команда отключения NIC применима к ВМ, находящимся в состояниях `runn` и `poff`. *Пример:*

```
tivvm nic-detach 15 2
```

Операции подключения и отключения NIC могут быть вызваны из web-интерфейса TIVM со страницы управления ВМ, вкладки "Сеть" (рис. 5.30).

5.3.10 Управление снимками состояния ВМ

Для ВМ, поддерживаемых гипервизором KVM, с дисками, использующими драйвер `Qcow2`, имеются возможности оперировать снимками состояния. *Снимок состояния ВМ* включает снимки текущего состояния ее дисков и памяти.

Для создания снимка состояния ВМ служит команда `tivvm snapshot-create`. Она имеет 2 аргумента. Первый (обязательный) – имя

или идентификатор VM. Второй (необязательный) – имя создаваемого снимка.

Например, следующая команда создает снимок состояния VM с идентификатором 15 и присваивает ему имя `current_snapshot1`:

```
tivvm snapshot-create 15 current-snapshot1
```

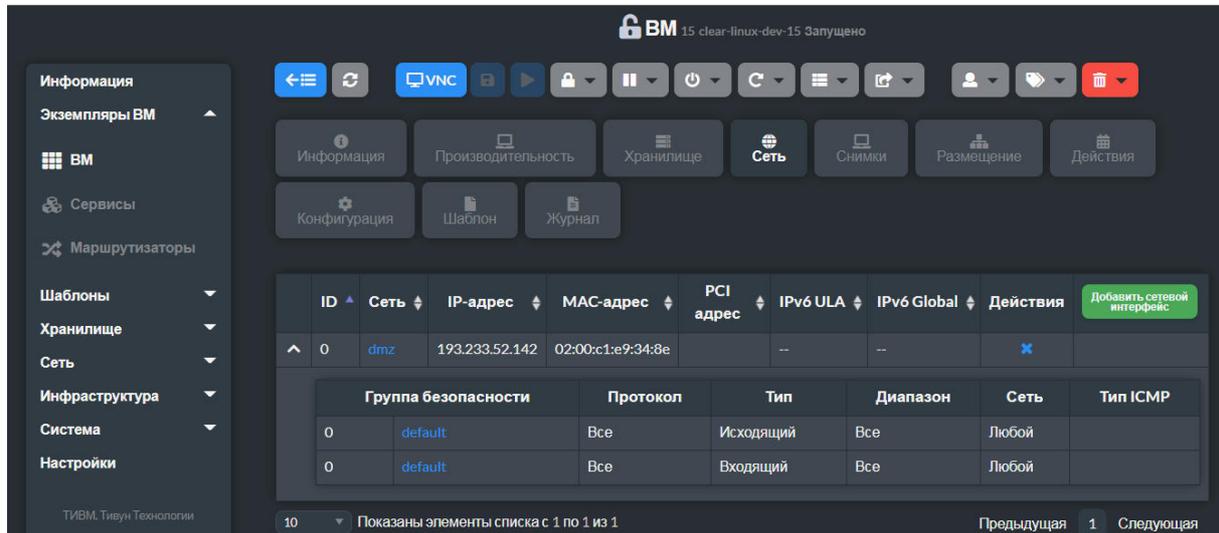


Рис. 5.30 – Страница управления сетевыми интерфейсами VM

Список имеющихся снимков состояния VM входит в состав информации, возвращаемой командой `tivvm show`:

```
tivvm show 15
VIRTUAL MACHINE 15 INFORMATION
ID                : 15
NAME              : clear-linux-dev-15
USER              : tivadmin
GROUP             : tivadmin
STATE             : ACTIVE
LCM_STATE         : HOTPLUG_SNAPSHOT
LOCK              : None
RESCHED           : No
HOST              : 127.0.0.1
CLUSTER ID       : 0
CLUSTER           : default
START TIME        : 06/29 21:47:21
END TIME          : -
DEPLOY ID        : tiv-15
. . .
SNAPSHOTS
  ID      TIME  NAME                HYPERVISOR_ID
  0  04/29 17:58 current-snapshot1
. . .
```

Пользователь может вернуть состояние VM к тому, что было зафиксировано в ранее снятом снимке с помощью команды `tivvm snapshot-revert`:

```
tivvm snapshot-revert VmId SnapshotId
```

VmId – имя или идентификатор VM, *SnapshotId* – идентификатор снимка, к которому следует вернуться. Этот идентификатор можно узнать из списка, возвращенного командой `tivvm show`.

Для удаления снимка состояния VM служит команда `tivvm snapshot-delete`. Она имеет те же аргументы, что и `tivvm snapshot-revert`.

Все созданные снимки состояния VM уничтожаются при выполнении любой операции, меняющей ее состояние.

Операции со снимками состояния VM доступны из web-интерфейса TIVM: страница управления VM, вкладка "Снимки" (рис. 5.31).

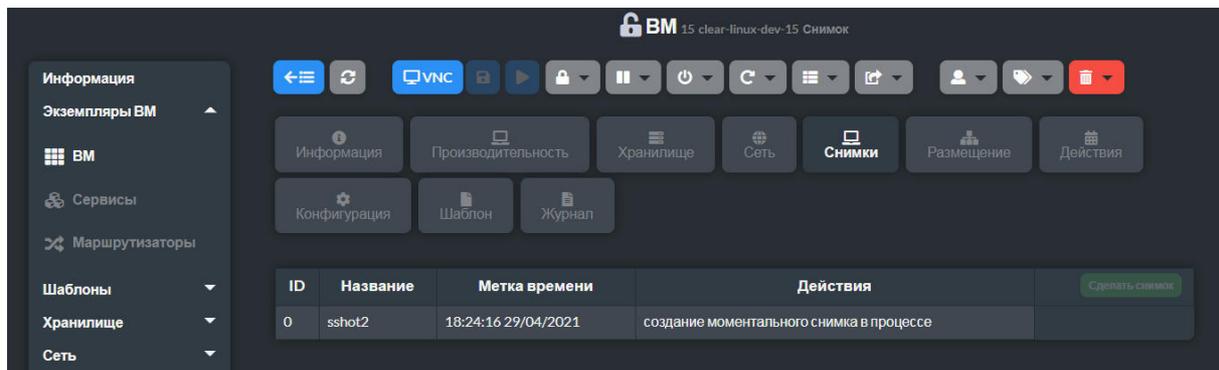


Рис. 5.31 – Страница управления снимками состояния VM

5.3.11 Изменение объемных характеристик VM

Изменение объемных характеристик VM – числа CPU, VCPU и объема памяти – возможно в состояниях `poff` и `unde`. Операцию изменения реализует команда `tivvm resize`:

```
tivvm resize VmId --parameter value
```

Здесь:

- *VmId* – имя или идентификатор VM;
- *parameter* – имя характеристики (`cpu`, `vcpu`, `memory`);
- *value* – значение характеристики.

CPU выражается числом с плавающей запятой, VCPU и объем памяти – целочисленными значениями. Память по умолчанию указывается в мегабайтах. Для задания ее объема в гигабайтах в конце значения следует указать символ "g".

Пример: изменение объема памяти и числа VCPU для VM с идентификатором 17. Последовательно вызываются команды выключения VM, изменения ее объемных характеристик и возобновления работы.

```
tivm poweroff 17
tivm resize 17 --vcpu 2 --memory 4g
tivm resume 17
```

Объемные характеристики VM могут быть изменены из web-интерфейса TIVM со страницы управления VM, вкладка "Производительность" (рис. 5.32).

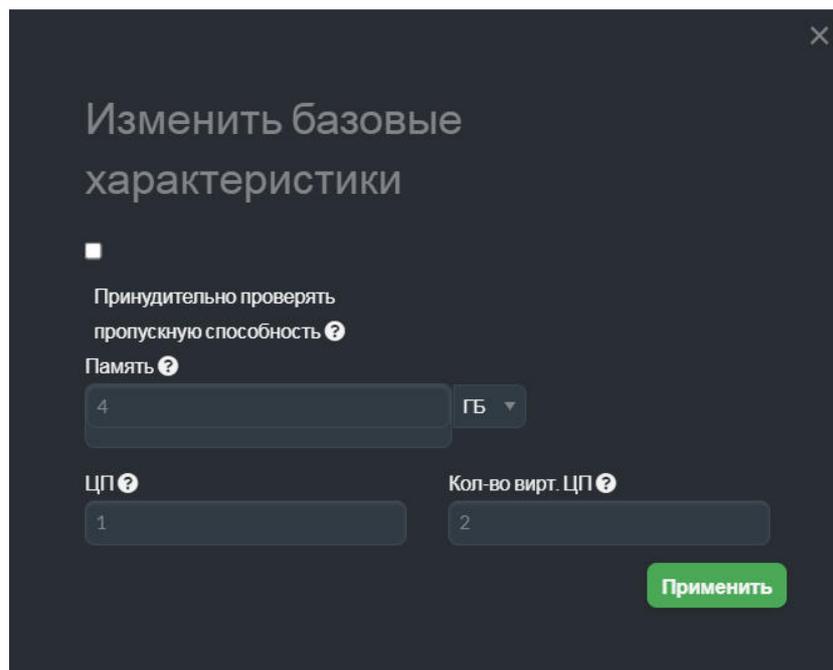


Рис. 5.32 – Диалог изменения объемных характеристик VM

5.3.12 Изменение объема дисков VM

Объем диска VM может быть увеличен по сравнению с тем, что имеет его образ. Это может быть достигнуто:

- на этапе создания VM (команда `tivtemplate instantiate`);
- на этапе загрузки VM (состояние `boot`) благодаря использованию механизма контекстуализации (разд. 6);
- для существующей VM, находящейся в состоянии `runn` или `poft`, с помощью команды `tivvm disk-resize`;

Для увеличения объема диска при создании VM в команде `tivtemplate instantiate` необходимо указать параметр `--disk`, за которым привести обозначение образа и нужный размер, например:

```
tivtemplate instantiate 5 --disk image0:size=4000
```

Эта команда создает VM на основе шаблона с идентификатором 5, у которой объем диска на основе образа `image0` увеличивается до 4 Гб.

Альтернативное решение: указать в `tivtemplate instantiate` ссылку на файл шаблона, переопределяющего размер диска, установленный в исходном шаблоне.

Диалог создания VM в web-интерфейсе TIVM показан на рис. 5.18.

Команда увеличения объема диска существующей VM:

```
tivm disk-resize VmId DiskID Size
```

Все 3 ее аргумента являются обязательными:

- `VmId` – имя или идентификатор VM;
- `DiskID` – идентификатор диска;
- `Size` – увеличенный объем диска.

После объема может быть указан символ, обозначающий единицу измерения: "М" – мегабайты, "G" – гигабайты, "T" – терабайты. По умолчанию объем выражается в мегабайтах.

Объем диска существующей VM может быть увеличен из web-интерфейса TIVM со страницы управления VM, вкладка "Хранилище" (рис. 5.33).

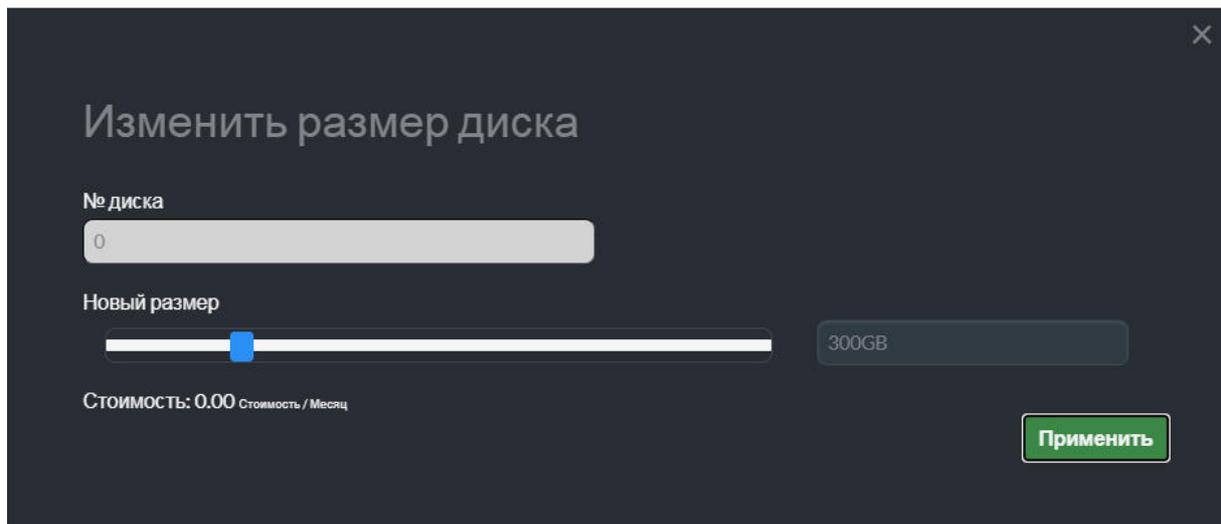


Рис. 5.33 – Диалог увеличения объема диска VM

5.3.13 Изменение конфигурации VM

Некоторые атрибуты конфигурации VM, определенные в ее шаблоне, могут быть изменены после создания VM. Для этого предназначена команда `tivvm updateconf`:

```
tivvm updateconf VmId [file]
```

Команда имеет 2 аргумента

- `VmId` (обязательный) – имя или идентификатор VM;

- *file* (необязательный) – путь к файлу шаблона, переопределяющему текущие атрибуты конфигурации.

Если второй аргумент не указан, то будет открыт редактор для ввода данных.

Атрибуты, которые можно менять с помощью данной команды, приведены в табл. 5.5.

Табл. 5.5 – Атрибуты конфигурации ВМ, которые могут быть изменены после ее создания

Атрибут	Подчиненные атрибуты
OS	ARCH, MACHINE, KERNEL, INITRD, BOOTLOADER, BOOT, SD_DISK_BUS
FEATURES	ACPI, PAE, APIC, LOCALTIME, HYPERV, GUEST_AGENT
INPUT	TYPE, BUS
GRAPHICS	TYPE, LISTEN, PASSWD, KEYMAP
RAW	DATA, DATA_VMX, TYPE
CONTEXT	Любые атрибуты

Изменение атрибутов конфигурации допустимо, если ВМ находится в состоянии *pend*, *fail*, *pooff*, *unde*, *hold* или *clon*. После изменения атрибутов *GRAPHICS* необходим перезапуск ВМ.

Возможности изменения конфигурации ВМ предусмотрены и в web-интерфейсе TIVM. Для этого нужно открыть страницу управления ВМ, выбрать вкладку "Конфигурация" и нажать кнопку "Обновить конфигурацию" (рис. 5.34). Страница редактора конфигурации показана на рис. 5.35. Атрибуты ВМ распределены по вкладкам "ОС и ЦП", "Ввод/вывод", "Контекст" и "Метки" (прикладные пользовательские атрибуты).

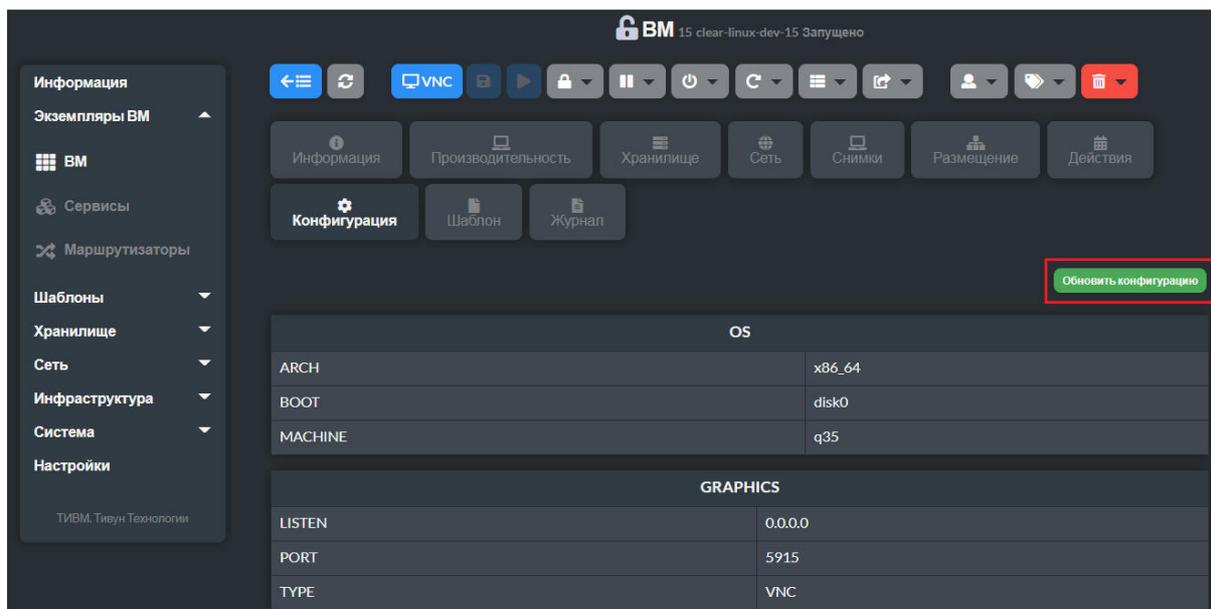


Рис. 5.34 – Страница конфигурации VM

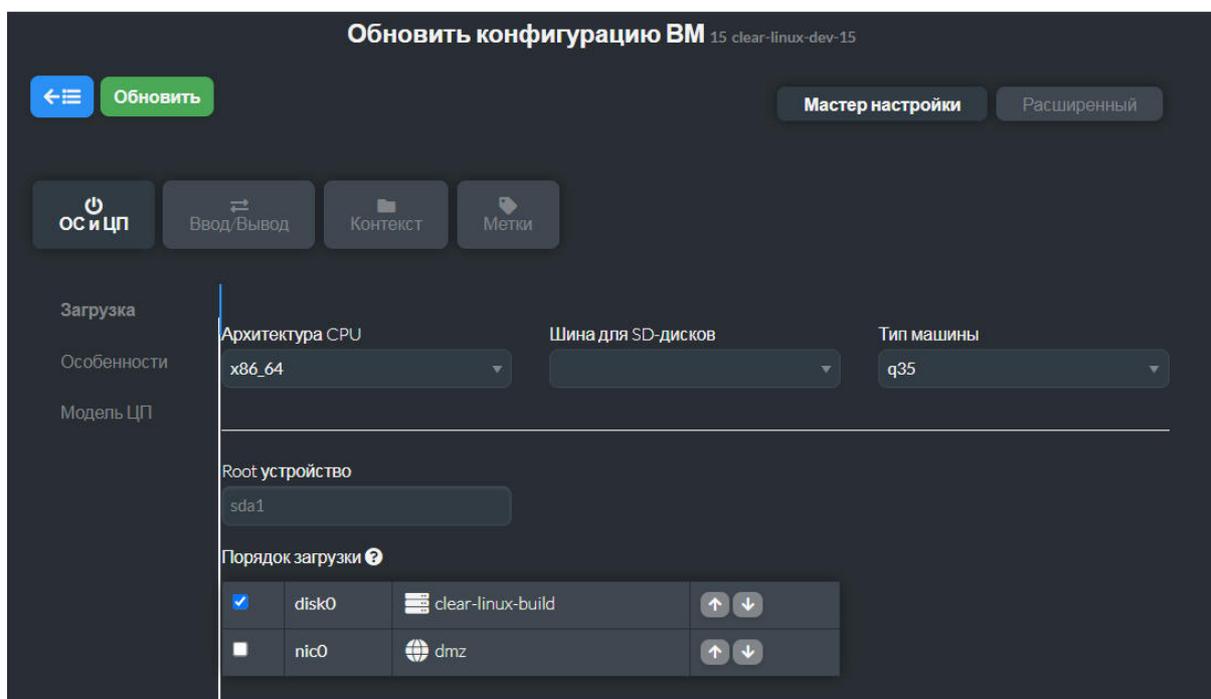


Рис. 5.35 – Диалог изменения конфигурации VM

5.3.14 Копирование VM

VM или ее шаблон могут быть скопированы в шаблон новой VM. Копия будет сохраняться после завершения работы и удаления исходной VM. Формируемый в результате копирования шаблон является частным и доступен только создавшему его пользователю.

Существуют 2 способа создания постоянной частной копии VM:

- 1) создание постоянной (persistent) VM на основе исходного шаблона командой `tivtemplate instantiate`;
- 2) сохранение VM как нового шаблона с помощью команды `tivvm save`.

5.3.14.1 Создание постоянной VM

Параметр `--persistent` команды `tivtemplate instantiate` позволяет создать *постоянную VM*: для каждого образа диска, используемого в исходном шаблоне, создается постоянная копия, которая связывается с новым экземпляром VM.

Пример:

```
tivtemplate instantiate 5 --persistent --name new vm
```

Данная команда создает постоянную VM с именем `new_vm` на основе шаблона с идентификатором 5. Помимо новой VM в результате выполнения команды будет сформирован ее шаблон, а также постоянные образы дисков, используемые в нем.

Следует иметь в виду, что временные диски, используемые в исходном шаблоне, не могут быть переведены в постоянные. Если такие диски определены в шаблоне, то в создаваемой VM они также будут временными.

Для создания постоянной VM в web-интерфейсе TIVM на соответствующей странице необходимо установить признак "Создать как постоянную" (рис. 5.18).

5.3.14.2 Сохранение VM как нового шаблона

VM, которая была создана как непостоянная, может быть сохранена, как новый шаблон VM. После удаления исходной VM созданный шаблон сохраняется.

Операцию реализует команда `tivvm save`:

```
tivvm save vmid name [--persistent]
```

Ее обязательные аргументы:

- `vmid` – имя или идентификатор исходной VM;
- `name` – имя создаваемого шаблона VM.

Команда применима к VM в состоянии `poff`. Таким образом, перед сохранением нового шаблона исходная VM должна быть выключена. Например, следующие команды обеспечивают выключение VM с идентификатором 5 и создание на ее основе шаблона новой VM с именем `new_vm`:

```
tivvm poweroff 5
```

```
tivvm save 5 new_vm
Template ID: 21
```

Команда `tivvm save` формирует копию шаблона исходной ВМ с заменой в нем образов дисков снимками состояния этих дисков из исходной ВМ. Она возвращает идентификатор созданного шаблона.

Изменения в шаблоне исходной ВМ, сделанные после ее запуска, а также корректировки ее объемных характеристик (`tivvm resize`), состава и размеров дисков (`tivvm disk-attach`, `tivvm disk-detach`, `tivvm disk-resize`), сетевых интерфейсов (`tivvm nic-attach`, `tivvm nic-detach`) применяются при создании нового шаблона (сохраняются в нем).

При указании параметра `--persistent` образы дисков в создаваемом шаблоне становятся постоянными. Это действие не применимо по отношению к временным дискам. Если такие диски есть в исходном шаблоне, то в новом шаблоне они также будут временными, а в запускаемой на его основе ВМ они будут пустыми.

Команда сохранения ВМ может быть вызвана из web-интерфейса TIVM со страницы управления ВМ. Необходимо выбрать ВМ в состоянии `poweroff`, нажать кнопку , в открывшемся диалоге ввести имя шаблона и нажать кнопку "Сохранить как шаблон" (рис. 5.36).

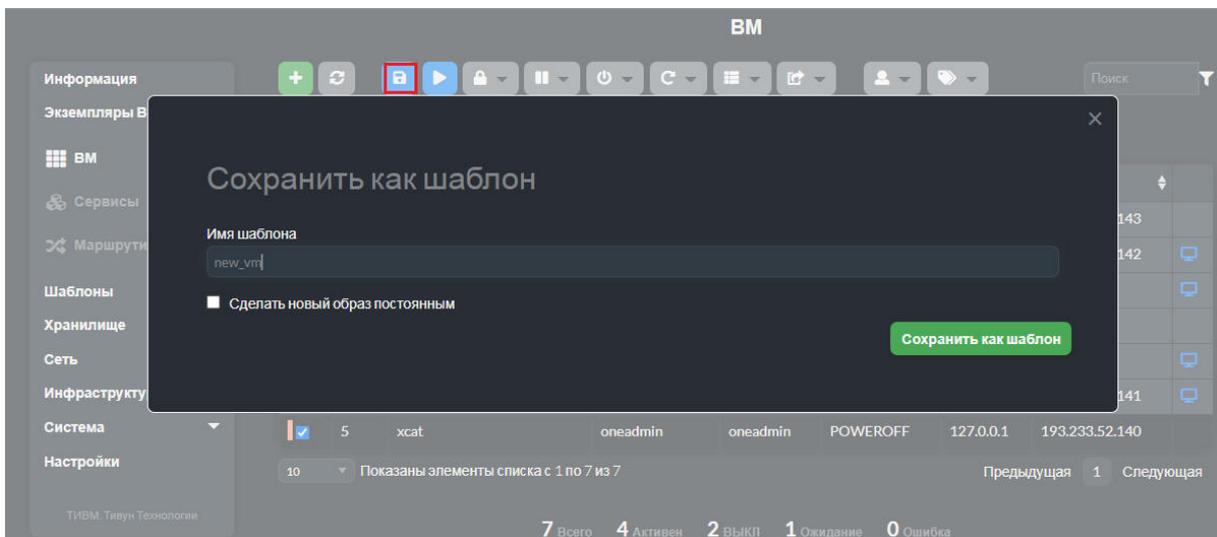


Рис. 5.36 – Сохранение ВМ как нового шаблона

5.3.15 Планирование операций с ВМ

Операции с ВМ могут автоматически выполняться планировщиком TIVM на основе заранее составленного *расписания*. Планирование операций с ВМ заключается в формировании этого расписания.

Момент планируемого выполнения операции с ВМ может быть определен *точно* (операция назначена на конкретные дату и время) или *относительно*, как время с момента запуска ВМ. Операции с точным моментом выполнения могут быть запланированы как *однократные* или *периодические* (например, подлежащие выполнению в определенное время каждый день, в установленные дни недели или месяца).

Операции с ВМ, которые можно планировать, соответствуют следующим командам из группы `tivvm`:

- `terminate;`
- `undeploy;`
- `hold;`
- `release;`
- `stop;`
- `suspend;`
- `resume;`
- `delete;`
- `delete-recreate;`
- `reboot;`
- `poweroff;`
- `snapshot-create;`
- `snapshot-revert;`
- `snapshot-delete;`
- `disk-snapshot-create;`
- `disk-snapshot-revert;`
- `disk-snapshot-delete.`

Если команда из приведенного перечня вызвана с параметром `--schedule`, то ее выполнение откладывается до планируемого момента времени, а в *расписание включается запись* о ее планируемом вызове.

Расписание операций сохраняется в составе пользовательских атрибутов шаблона ВМ. Оно выводится командой `tivvm show` в разделе "SCHEDULED ACTIONS". Для *изменения расписания* служит команда корректировки пользовательских атрибутов шаблона ВМ `tivvm update`.

Пример:

```
tivvm show 17
VIRTUAL MACHINE 17 INFORMATION
ID                : 17
NAME              : vr-VR01-0
USER              : tivadmin
GROUP             : tivadmin
STATE             : PENDING
LCM_STATE         : LCM_INIT
LOCK              : None
RESCHED           : No
START TIME        : 03/02 17:15:00
END TIME          : -
DEPLOY ID         : -
```

```

VIRTUAL ROUTER ID      : 0
. . .
SCHEDULED ACTIONS
ID ACTION      SCHEDULED REPEAT      END      DONE MESSAGE
0 stop        06/30 12:30      On 06/30/21 -
. . .
tivvm update 17
SCHEM_ACTION=[
  ACTION="stop",
  END_TYPE="2",
  END_VALUE="1625045400",
  ID="0",
  TIME="1625045400" ]
. . .

```

В данном примере сначала приведен фрагмент шаблона ВМ с идентификатором 17. Расписание операций содержит единственную запись – остановку ВМ, запланированную на 30.06.2021 12:30.

Далее приведен фрагмент интерфейса редактирования пользовательских атрибутов шаблона ВМ, открытого командой `tivvm update`. Данная команда имеет один обязательный аргумент – имя или идентификатор ВМ. Второй аргумент – имя файла шаблона – является не обязательным. Если он отсутствует (как в данном примере), то команда открывает редактор шаблона. Запись расписания представляет атрибут `SCHEM_ACTION`, включающий ряд подчиненных атрибутов.

Состав атрибутов записи `SCHEM_ACTION` приведен в табл. 5.6.

Таблица 5.6 – Атрибуты записи расписания, представляющей планируемую операцию с ВМ

Атрибут	Описание
ACTION	Операция, соответствующая команде из группы <code>tivvm</code> .
TIME	Планируемое время выполнения однократной операции или время планирования периодической операции. Значение в секундах (Unix time)
REPEAT	Интервал времени, для которого задается частота повторения. Указывается только для периодической операции. Набор значений: - WEEKLY (или 0) – неделя; - MONTHLY (или 1) – месяц; - YEARLY (или 2) – год; - HOURLY (или 3) – часы.
DAYS	Частота. Указывается только для периодической операции. Одно или несколько значений, разделяемых запятыми. Их интерпретация зависит от атрибута REPEAT: - WEEKLY – номер дня недели от 0 до 6; - MONTHLY – день месяца от 1 до 31; - YEARLY – номер дня года от 0 до 365; - HOURLY – число часов в пределах недели (от 0 до 168), через которые должна выполняться операция.

END_TYPE	Тип определения условия завершения операции. Набор значений: - NEVER (или 0) – условие не задается; - NUMBER OF REPETITIONS (или 1) – число попыток выполнения; - DATE (или 2) – время, к которому операция должна быть завершена, или прекращено выполнение периодической операции.
END_VALUE	Условие завершения операции. Тип и интерпретация значения зависят от атрибута END_TYPE: - NUMBER OF REPETITIONS – число повторов (целое число, большее 0); - DATE – время (значение в секундах, Unix time).
ID	Идентификатор записи расписания.
ARGS	Параметры, передаваемые команде, выполнение которой планируется.

Расписание операций с ВМ отображается в web-интерфейсе TIVM на странице управления ВМ, вкладка "Действия" (рис. 5.37). Для включения записи в расписание служит кнопка "Добавить действие", открывающая диалог планирования операции (рис. 5.38).

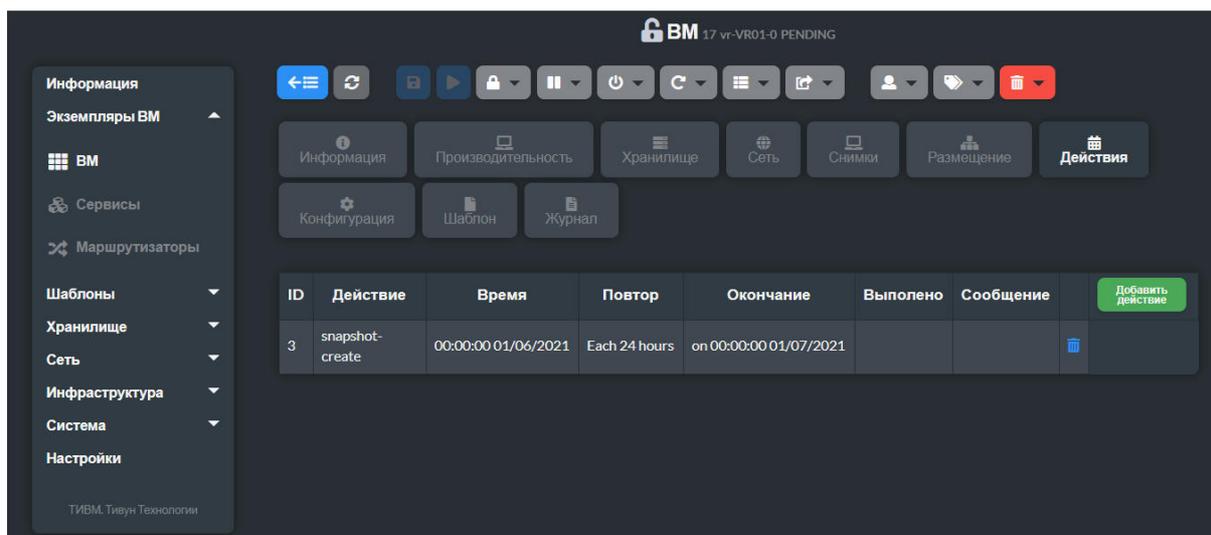


Рис. 5.37 – Расписание операций с ВМ

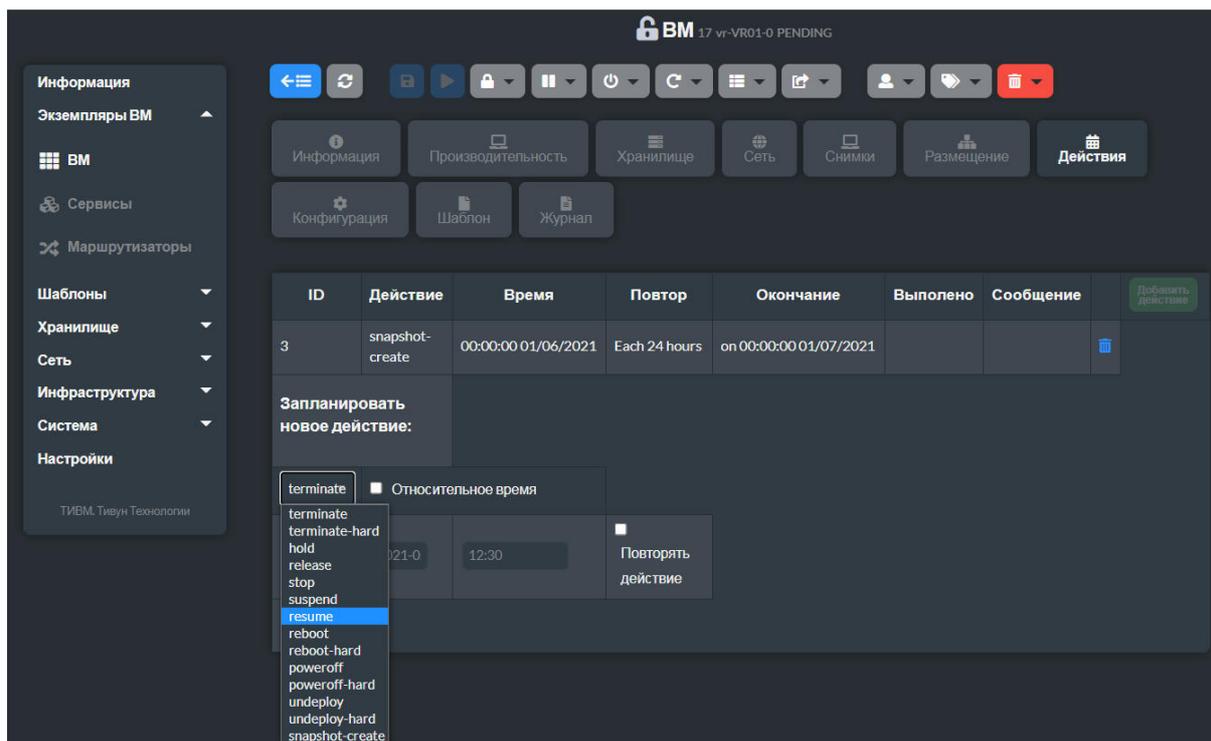


Рис. 5.38 – Диалог планирования операции с VM

5.3.15.1 Однократные операции с точно заданным временем выполнения

Включение записи в расписание обеспечивает параметр `--schedule`, указанный в команде из перечня, приведенного в п. 5.3.15. Аргумент параметра – планируемые дата и время выполнения – приводится в кавычках в формате ISO 8601. Если в аргументе отсутствует год, то принимается текущий год, если отсутствуют секунды, то используется нулевое значение секунд, если отсутствует временная зона, то принимается текущая временная зона сервера TIVM. Аргумент также может быть выражен числом секунд (Unix time).

Пример:

```
tivvm stop 17 --schedule "11/30 05:55"
tivvm resume 17 --schedule "2021-12-30 05:50"
tivvm show 17
VIRTUAL MACHINE 17 INFORMATION
ID           : 17
NAME        : vr-VR01-0
USER        : tivadmin
GROUP       : tivadmin
. . .
SCHEDULED ACTIONS
ID ACTION      SCHEDULED  REPEAT      END          DONE MESSAGE
0 stop        11/30 05:50
1 resume     12/30 05:50
. . .
```

```

tivvm update 17
SCHEM_ACTION=[
  ACTION="stop",
  ID="0",
  TIME="1638240600" ]
SCHEM_ACTION=[
  ACTION="resume",
  ID="1",
  TIME="1640832600" ]
. . .

```

Редактирование записей расписания, представляющих однократные операции с точно заданным временем выполнения, сводятся к корректировке 3 атрибутов: ACTION, TIME и ID. При этом время выполнения в шаблоне выражается числом секунд, как Unix time.

5.3.15.2 Периодические операции с точно заданным временем выполнения

Для включения записи в расписание в команде из перечня, приведенного в п. 5.3.15, помимо параметра `--schedule` необходимо указать параметр *периодичности* (`--weekly`, `--monthly`, `--yearly` или `--hourly`) и параметр *условия завершения операции* (`--end`). Типы аргументов этих параметров приведены в табл. 5.6.

Значения времени в аргументах параметров вводятся и обрабатываются так, как написано в предыдущем пункте. Значения периодичности, разделяемые запятыми, приводятся в кавычках. Одиночные целочисленные значения указываются без кавычек.

Пример:

```

tivvm snapshot-create 17 --schedule "2021-06-01" --hourly 24
--end "2021-07-01"
tivvm show 17
VIRTUAL MACHINE 17 INFORMATION
ID                : 17
NAME              : vr-VR01-0
USER              : tivadmin
GROUP            : tivadmin
. . .
SCHEDULED ACTIONS
ID ACTION          SCHEDULED REPEAT          END          DONE MESS.
 3 snapshot-create 06/01 00:00 Each 24 hours On 07/01/21 -
. . .
tivvm update 17
SCHEM_ACTION=[
  ACTION="snapshot-create",
  DAYS="24",
  END_TYPE="2",
  END_VALUE="1625086800",
  ID="3",

```

```
REPEAT="3",
TIME="1622494800" ]
. . .
```

8.3.15.3 Операции с относительным временем выполнения

Для таких операций в расписании указывается *время с момента запуска ВМ*. Соответствующие записи расписания, сохраненные в шаблоне ВМ, начинают действовать после создания ВМ на основе шаблона. Их включение в шаблон обеспечивает команда его редактирования `tivtemplate update` (см. п. 5.2.3).

Например, следующая запись в шаблоне планирует операцию выключения ВМ с идентификатором 1 после 24 часов ее работы:

```
tivtemplate update 1
SCHEM_ACTION=[
  ACTION="poweroff",
  ID="0",
  TIME="+86400" ]
. . .
```

Некоторые из команд, приведенные в п. 5.3.15, имеют обязательные аргументы. При вызове команды с параметром `--schedule` эти аргументы включаются в него. В случае ввода записи расписания в шаблон для их указания служит атрибут `ARGS` (см. табл. 5.6). Команды с обязательными аргументами приведены в табл. 5.7.

В следующем *примере* запись `SCHEM_ACTION` планирует периодическое создание снимка состояния образа диска ВМ по 1-м и 5-м дням недели. Идентификатор диска (0) и имя снимка (`disk_snapshot_example`) определены с помощью атрибута `ARGS`. Общее число выполнений команды равно 5.

```
SCHEM_ACTION=[
  ACTION="disk-snapshot-create",
  ARGS="0, disksnap_example",
  DAYS="1,5",
  END_TYPE="1",
  END_VALUE="5",
  ID="0",
  REPEAT="0",
  TIME="1625086800" ]
```

Таблица 5.7 – Команды с обязательными аргументами, реализующие операции с ВМ, которые можно планировать

Команда	Аргумент	Описание
<code>snapshot-create</code>	<i>Name</i>	Имя снимка
<code>snapshot-revert</code>	<i>SnapshotId</i>	Идентификатор снимка
<code>snapshot-delete</code>	<i>SnapshotId</i>	Идентификатор снимка
<code>disk-snapshot-create</code>	<i>DiskID</i>	Идентификатор диска

Команда	Аргумент	Описание
	<i>Name</i>	Имя снимка
disk-snapshot-revert	<i>DiskID</i>	Идентификатор диска
	<i>Name</i>	Имя снимка
disk-snapshot-delete	<i>DiskID</i>	Идентификатор диска
	<i>Name</i>	Имя снимка

5.3.16 Присвоение ВМ дополнительных атрибутов

ВМ могут быть присвоены *дополнительные (пользовательские) атрибуты*. Они сохраняются в шаблоне ВМ в разделе "USER TEMPLATE".

Ввод и редактирование пользовательских атрибутов обеспечивает команда **tivvm update**. Она имеет один обязательный аргумент – имя или идентификатор ВМ. Второй аргумент – имя файла шаблона – является не обязательным. Если он отсутствует, то команда открывает редактор шаблона.

Пользовательский атрибут представляет строка шаблона, содержащая его имя и значение в формате `name = "value"`.

Пример. ВМ с идентификатором 17 присвоен атрибут `user_property1` со значением `value1`. Данный атрибут возвращается командой `tivvm show`.

```

tivvm update 17
user_property1="value1"
tivvm show 17
VIRTUAL MACHINE 17 INFORMATION
ID                : 17
NAME              : vr-VR01-0
USER              : tivadmin
GROUP            : tivadmin
. . .

USER TEMPLATE
USER_PROPERTY1="value1"

VIRTUAL MACHINE TEMPLATE
. . .

```

5.3.17 Управление правами доступа к ВМ

Пользователь, создавший ВМ, становится ее *владельцем*. Для ВМ, как и других ресурсов, действует система *разрешений* на операции с ними (см. разд. 3.5). По умолчанию при создании ВМ для нее устанавливаются разрешения, выражаемые октетом 600: владельцу доступны операции категорий USE и MANAGE, для прочих пользователей доступа к ВМ нет.

Пользователь, не имеющий доступа к ВМ, не получает информацию о ней в списке, возвращаемую командами `tivvm list` и `tivvm show`.

Владелец ВМ может *управлять доступом к операциям* с ней пользователям его группы и прочим пользователям, изменив разрешения с помощью команды **tivvm chmod**:

```
tivvm chmod rangeVmId | VmIdList permissions
```

Аргументы команды:

- *rangeVmId* – диапазон идентификаторов ВМ в форме *n..m, k*;
- *VmIdList* – список идентификаторов или имен ВМ, перечисляемых через запятую. Список может состоять из одного элемента;
- *permissions* – разрешения, представляемые 3-разрядным 8-ричным кодом (октет).

Примеры:

```
tivvm chmod 15 660
tivvm chmod 15..17,19 664
tivvm chmod vm12, vm sys 2 660
```

Администратор системы может *менять владельца и группу ВМ* с помощью команд **tivvm chgrp** и **tivvm chown**:

```
tivvm chgrp rangeVmId | VmIdList groupId
tivvm chgrp rangeVmId | VmIdList userId [groupId]
```

В этих командах:

- *groupId* – имя или идентификатор группы;
- *userId* – имя или идентификатор пользователя-владельца.

Доступ к ВМ также регулируют правила СУД (см. разд. 3.5). Они формируются администраторами.

5.3.18 Перепланирование вычислительных ресурсов, выделяемых ВМ

После создания ВМ планировщик TIVM получает задание на ее планирование, т.е. выделение вычислительных ресурсов, наиболее подходящих для данной ВМ, и ее развертывание на основе этих ресурсов. *Перепланирование* – это повторное выполнение данной процедуры для уже развернутой ВМ. Оно может привести к переносу ВМ на другой хост виртуализации.

Необходимость в перепланировании может возникнуть при модификации требований ВМ к вычислительным ресурсам или критериев их выбора, а также изменении нагрузки на вычислительную инфраструктуру в целом.

Планировщик работает циклично, обрабатывая очередь заданий. ВМ имеют *логический признак запроса на перепланирование* RESCHED. Наличие истинного значения данного признака означает, что ВМ ожидает

перепланирования, т.е. связанное с ней задание находится в очереди планировщика.

Текущее значение признака **RESHED** входит в состав информации, возвращаемой командой `tivvm show`. Для его установки и снятия предназначены команды **tivvm resched** и **tivvm unresched**:

```
tivvm resched rangeVmId | VmIdList
tivvm unresched rangeVmId | VmIdList
```

Первая команда устанавливает признак, вторая – снимает. В качестве аргумента они принимают диапазон идентификаторов ВМ (*rangeVmId*) или список идентификаторов или имен ВМ (*VmIdList*). Команды применимы к ВМ в состояниях `runn` и `poff`.

5.3.19 Перенос ВМ на другие ресурсы

Перенос работающей ВМ (находящейся в состоянии `runn`) на другие вычислительные ресурсы (хост виртуализации, хранилище данных) реализует команда **tivvm migrate**:

```
tivvm migrate rangeVmId | VmIdList hostId [dsId] [options]
```

Ее аргументы:

- диапазон идентификаторов ВМ (*rangeVmId*) или список идентификаторов или имен ВМ (*VmIdList*), подлежащих переносу;
- *hostId* – имя или идентификатор хоста, на который переносятся ВМ;
- *dsId* (необязательный) – имя или идентификатор хранилища данных, которое должно использоваться после переноса.

Допустимые параметры (секция [*options*]):

- `poff` – "холодная миграция": ВМ останавливается, переносится на целевой хост и запускается на нем;
- `poff-hard` – "холодная миграция" с жестким выключением ВМ (TIVM не ждет выполнения ВМ сигнала отключения ACPI, см. п. 8.3.4);
- `live` – "живая миграция": ВМ переносится на другой хост без выключения (пауза в работе ВМ, вызванная переносом будет минимальной). Данный режим переноса возможен, если хранилище данных использует разделяемую файловую систему;
- `e`, --`enforce` – переносить ВМ, даже если целевой хост не обладает достаточными ресурсами для их размещения.

Операция переноса на другие ресурсы по команде TIVM не поддерживается для ВМ, основанных на виртуализации VMware. Она должна выполняться непосредственно в vCenter.

5.3.20 Восстановление функционирования ВМ, ожидающей операции драйвера

ВМ может "зависнуть" в каком-либо состоянии в ожидании операции драйвера, которая не выполнена из-за сбоя или ошибки. Для разрешения подобной ситуации и восстановления работы ВМ служит команда `tivvm recover`. Она позволяет:

- симитировать получение успешного или ошибочного результата запрошенной операции драйвера;

- запросить повторное выполнение операции драйвера;

- пересоздать ВМ;

- удалить ВМ.

Формат команды:

```
tivvm recover rangeVmId | VmIdList action
```

Ее аргумент – диапазон идентификаторов ВМ (*rangeVmId*) или список идентификаторов или имен ВМ (*VmIdList*), к которым относится действие, задаваемое параметром *action*. Последний может иметь следующие значения:

- success – имитирует получение успешного результата запрошенной операции драйвера. Действие применимо к ВМ в любом активном состоянии;

- failrue – имитирует получение ошибки выполнения запрошенной операции драйвера;

- retry – запрашивает повторное выполнение операции драйвера;

- recreate – удаление ВМ и перевод ее в состояние `pend` для последующего развертывания. Действие применимо к ВМ в любом состоянии кроме `stop` и `unde`;

- delete – удаление ВМ. Она переходит в состояние `done`;

- delete-db – удаление ВМ. Информация о ней удаляется из БД.

5.3.21 Доступ к консоли ВМ через VNC

Web-интерфейс TIVM позволяет подключиться к консоли ВМ, находящейся в состоянии `runn` и поддерживающей доступ через VNC. Такие ВМ на странице их списка (рис. 5.39) имеют пиктограмму . Щелчок по ней открывает окно консоли (рис. 5.40).

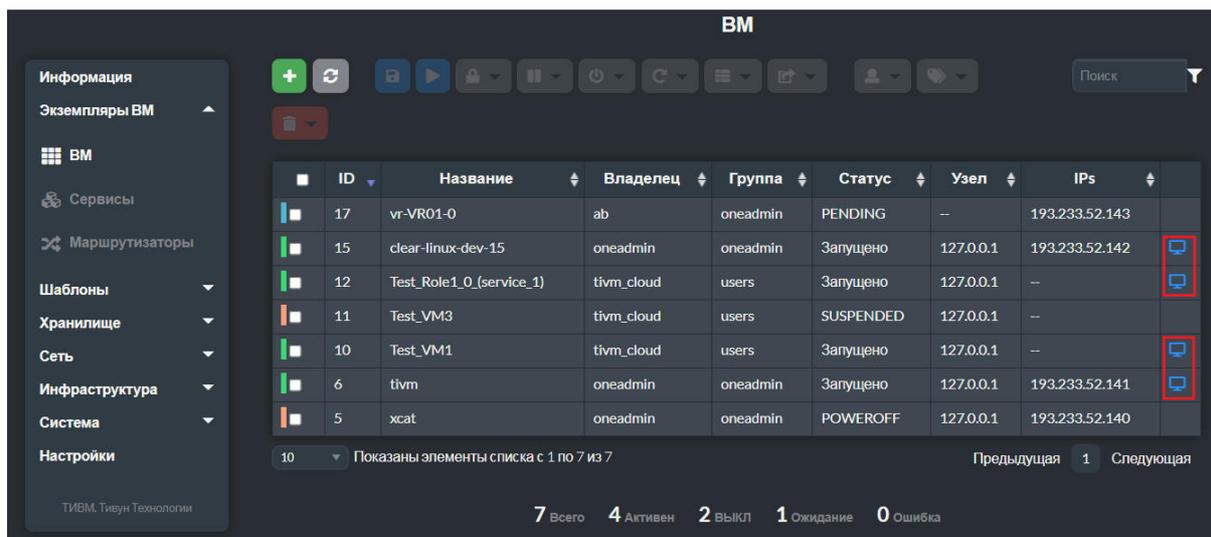


Рис. 5.39 – Пиктограммы доступа к консоли VM на странице их списка

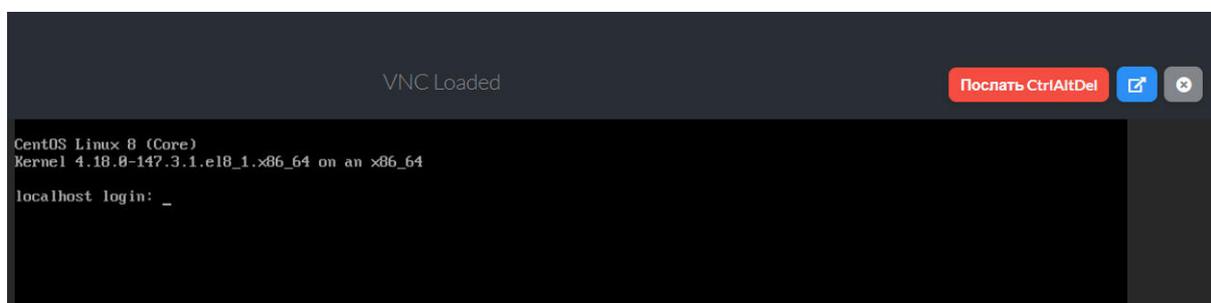


Рис. 5.40 – Консоль VM, открытая в браузере из web-интерфейса TIVM

VNC-доступ к консоли VM, использующих менеджер контейнеров LXD, реализуется через команду `lxc exec containerName --<command>`, запускающую команду `<command>` внутри контейнера (VM) с именем `containerName`. По умолчанию запускается команда `/bin/login`. Она может быть переопределена путем редактирования файла `/var/tmp/tiv/etc/vmm/lxd/lxdrc`, размещенного на LXD-хосте виртуализации.

Команда, запускаемая на VM при доступе к ее консоли, протокол передачи данных удаленного рабочего стола (VNC, SPICE, SDL) и ряд других параметров консоли могут быть заданы для каждой VM в ее шаблоне в разделе `GRAPHICS`. Они могут быть установлены и в web-интерфейсе TIVM: страница управления VM, вкладка "Конфигурация", кнопка "Обновить конфигурацию" (рис. 5.34), диалог изменения конфигурации (рис. 5.35), вкладка "Ввод-вывод" (рис. 5.41).

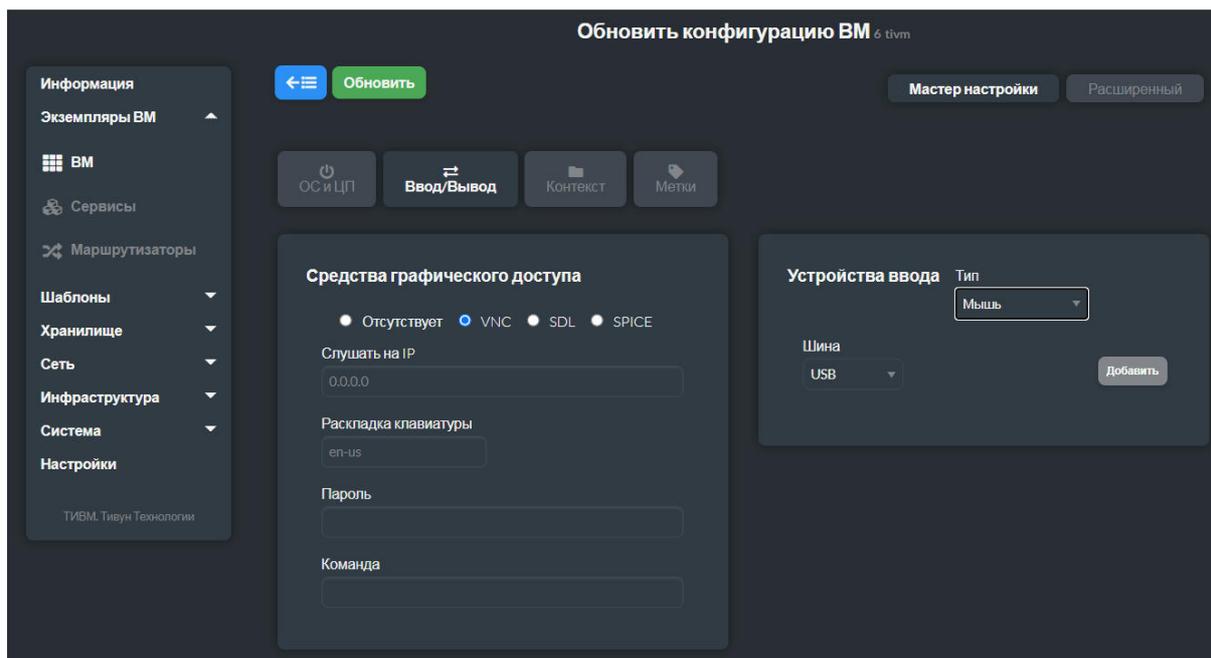


Рис. 5.41 – Настройка параметров консоли VM

5.4 Особенности управления виртуальными машинами, основанными на виртуализации VMware

Особенности управления VM, основанных на виртуализации VMware, обусловлены тем, что TIVM оперирует ими не напрямую, а через систему управления vCenter. Операции с объектами vCenter (VM, кластерами, хранилищами данных, образами дисков, шаблонами) в CLI реализуются через группу команд `tivvcenter`.

5.4.1 Импорт шаблонов VM и запущенных в vCenter VM

В TIVM предусмотрены средства *импорта* шаблонов VM, созданных в vCenter, а также VM, запущенных в vCenter ("Wild VM"). В CLI доступ к этим средствам обеспечивает команда `tivvcenter import`. Тип импортируемого объекта задает параметр `object`.

В ходе импорта TIVM создает образы дисков и регистрирует виртуальные сети, используемые в шаблонах. Поэтому данная процедура может занять длительное время.

В табл. 5.8 приведены особенности определения атрибутов VM, представляемых в их шаблонах, импортируемых из vCenter.

Таблица 5.8 – Определение атрибутов VM в шаблоне, импортированном из vCenter

Атрибут	Комментарий
CPU	Число физических процессов, выделенных VM. Значение не обязательно должно соответствовать тому, что установлено в

Атрибут	Комментарий
	исходном шаблоне vCenter. При импорте в TIVM оно пересчитывается.
MEMORY	Объем оперативной памяти, выделенной VM. Значение не обязательно должно соответствовать тому, что установлено в исходном шаблоне vCenter. При импорте в TIVM оно пересчитывается.
HYPERVISOR	Значение "vcenter"
NIC	Один или множество экземпляров атрибута NIC с подчиненными атрибутами, представляющими характеристики сетевого интерфейса. Допустимые значения подчиненного атрибута MODEL: virtuale1000, virtuale1000e, virtualpcnet32, virtualsriovethernetcard, virtualvmxnetm, virtualvmxnet2, virtualvmxnet3.
GRAPHICS	В vCenter поддерживается только протокол VNC (подчиненный атрибут TYPE = "vnc").
CONTEXT	См. ограничения, приведенные в табл. 9.1.
VCENTER_RESOURCE_POOL	Пул вычислительных ресурсов, используемый для развертывания VM. Если атрибут не указан, то VM разворачивается в пуле вычислительных ресурсов по умолчанию.
VCENTER_VM_FOLDER	Папка, в которой размещается регистрационная информация VM в vCenter. Папки используются для группировки ресурсов и упрощения их выбора. Атрибут может отсутствовать.
VCENTER_TEMPLATE_REF	Идентификатор хранилища данных vCenter, используемого в VM. Такой идентификатор имеет унифицированный формат datastore-XXXX и называется ссылкой на управляемый объект (Managed Object Reference, MOR).
VCENTER_TEMPLATE_NAME	Имя VM (ее шаблона) в vCenter.
VCENTER_CCR_REF	Идентификатор кластера vCenter, к которому относится шаблон. Такой идентификатор имеет унифицированный формат MOR domain-XXXX.
VCENTER_INSTANCE_ID	Идентификатор экземпляра VM (ее шаблона) в vCenter. Такой идентификатор имеет унифицированный формат MOR vm-XXXX.

После запуска VM, основанной на импортированном шаблоне vCenter, или импорта запущенной в vCenter VM сервер TIVM позволяет управлять ее ЖЦ с помощью операций, представленных в разд. 5.3. Исключения составляют операции переноса VM на другие ресурсы (см. п. 5.3.19), которые не поддерживаются.

5.4.2 Запуск VM, поддерживаемой vCenter, на основе шаблона

Запуск VM на основе виртуализации VMware по команде TIVM реализует *процедура копирования шаблона VM*, выполняемая vCenter.

Другими словами, перед запуском ВМ система управления vCenter создает копию ее шаблона.

Копирование шаблона запускаемой ВМ включает копирование всех используемых в нем образов дисков. Если для исходных образов были сохранены снимки состояния, то они не включаются в копии (копируются последние сохраненные состояния образов). Для сформированных копий шаблона и образов дисков отменяется совместный доступ (права делегируются только владельцу запускаемой ВМ).

Система vCenter разворачивает ВМ в хранилище данных, определяемом планировщиком TIVM на основе политики, указанной в шаблоне ВМ, или политики по умолчанию для всех ВМ, заданной в файле конфигурации планировщика `/etc/tiv/sched.conf` (см. разд. 2.3). Для того, чтобы хранилища данных vCenter могли использоваться планировщиком TIVM, необходимо:

- импортировать их в TIVM;
- обеспечить совместный доступ к ним для всех ESX/ESXi-хостов кластера.

ВМ разворачивается в пуле вычислительных ресурсов vCenter, заданном атрибутом ее шаблона `VCENTER_RESOURCE_POOL`. Если последний не указан, то используется пул по умолчанию.

В web-интерфейсе vCenter запущенная ВМ по умолчанию помещается в ту же папку, что и ее исходный шаблон. Данная папка может быть переопределена с помощью атрибута шаблона `VCENTER_VM_FOLDER`.

5.4.3 Создание постоянной ВМ

Процедура копирования шаблона при запуске ВМ, описанная в предыдущем пункте, не означает, что эта ВМ становится постоянной. Запуск постоянной ВМ обеспечивает команда `tivtemplate instantiate` с параметром `--persistent`.

Различия между непостоянной и постоянной ВМ проявляются при завершении их работы (`tivvm terminate`). Для непостоянной ВМ все изменения в ее шаблоне и содержимом непостоянных дисков не сохраняются. При завершении работы постоянной ВМ в vCenter создается экземпляр шаблона, содержащий ее текущую конфигурацию, а в шаблоне данной ВМ в TIVM указывается ссылка на него. Содержимое постоянных дисков также сохраняется. Благодаря этому в дальнейшем может быть запущена ВМ с конфигурацией и дисками постоянной ВМ, ЖЦ который был завершен ранее.

Порядок создания постоянной ВМ, поддерживаемой vCenter, в целом аналогичен тому, что описан в п. 5.3.14.1. Специфика vCenter касается следующих двух обстоятельств.

Во-первых, в шаблоне VM в разделе DISK может быть указан атрибут `TIV_MANAGED` со значением `YES` или `NO`. Значение `YES` служит признаком того, что диск был импортирован в TIVM при импорте шаблона или ранее запущенной VM. Вариант `NO` фиксирует, что диск не подлежит импорту в TIVM и не управляется им. Такой диск не может быть объявлен постоянным. При запуске постоянной VM его образ копируется, как непостоянный.

Во-вторых, в запущенной постоянной VM не следует отключать диски или менять их размер, т.к. при завершении работы такой VM эти изменения конфигурации сохраняются в создаваемом шаблоне vCenter, но не в шаблоне TIVM, созданном при ее запуске. Расхождение этих шаблонов может привести к ошибкам при создании VM на их основе в дальнейшем.

5.4.4 Сохранение VM как нового шаблона

Процедура в целом совпадает с той, что описана в п. 5.3.14.2. Сформированный ею шаблон VM содержит элементы `DISK`, ссылающиеся на созданные копии образов дисков, и элементы `NIC`, ссылающиеся на виртуальные сети, используемые в исходной VM. Созданный шаблон также содержит ссылку на шаблон vCenter исходной VM в формате MOR. Развертывание VM по нему вызовет копирование шаблона vCenter с заменой в нем дисков на их копии, созданные ранее.

5.4.5 Особенности выделения вычислительных ресурсов планировщиком

Кластер вычислительных ресурсов vCenter состоит из набора хостов с гипервизорами ESX/ESXi. Для обеспечения работы планировщика TIVM в него должны быть *импортированы кластеры vCenter* (команда `tivvcenter hosts`). При таком импорте в TIVM регистрируется хост виртуализации. Он относится к определенному кластеру TIVM или кластеру по умолчанию, если конкретный кластер не задан.

При развертывании VM планировщик TIVM принимает решения на уровне зарегистрированных в нем хостов виртуализации. Для VM, основанных на виртуализации VMware, выбирается хост, представляющий кластер vCenter, который входит в кластер TIVM, содержащий используемые в VM образы дисков и виртуальные сети. Решение по размещению VM на конкретном ESX/ESXi-хосте в рамках выбранного кластера vCenter принимает vCenter.

Таким образом, если в шаблоне VM, использующем vCenter, не определены диски или виртуальные сети, то планировщик TIVM не сможет развернуть ее из-за отсутствия данных для выбора хоста (кластера vCenter).

5.4.6 Особенности операций с образами дисков

В технологии VMware используется формат образов дисков VMDK. Такие образы могут быть зарегистрированы в TIVM путем:

- загрузки VMDK-файла из локальной файловой системы или удаленного ресурса;
- создания учетной записи VMDK-образа, содержащегося в хранилище данных;
- создания пустого образа блока данных (тип – "Generic storage datablock" или "Обобщенный блок данных хранилища").

Для *импорта образов дисков vCenter* в TIVM в CLI предназначена команда `tivvcenter import` с параметром `--object images`.

При импорте из vCenter шаблона VM каждому импортированному вместе с ним образу диска присваивается сгенерированное имя, состоящее из имен VMDK-файла, хранилища данных и шаблона VM, созданного в TIVM.

Управление образами дисков, используемых в VM на базе vCenter, осуществляется с помощью средств, описанных в разд. 5.1. Особенность VMDK-образов – отсутствие поддержки операций со снимками состояния.

В табл. 5.9 представлены особенности определения атрибутов VMDK-образов дисков, представляемых в их шаблонах.

Таблица 5.9 – Особенности атрибутов VMDK-дисков в их шаблонах

Атрибут	Комментарий
PATH	Значение может быть: - локальный путь к VMDK-файлу; - путь к VMDK-файлу в хранилище данных vCenter (должен начинаться с префикса "vcenter://"); - URL сетевого ресурса.
VCENTER_ADAPTER_TYPE	Тип адаптера для подключения виртуальных дисков к VM. Если параметр не задан, то используется значение, определенное на уровне хранилища данных. Допустимые значения: <code>siLogic</code> , <code>ide</code> , <code>busLogic</code> .
VCENTER_DISK_TYPE	Тип диска (значение влияет на производительность и объем, занимаемый данными). Варианты значений: <code>delta</code> , <code>eagerZeroedThick</code> , <code>flatMonolithic</code> , <code>preallocated</code> , <code>raw</code> , <code>rdm</code> , <code>rdmp</code> , <code>seSparse</code> , <code>sparse2Gb</code> , <code>sparseMonolithic</code> , <code>thick</code> , <code>thick2Gb</code> , <code>thin</code> .
VCENTER_IMPORTED	Для образов, импортированных в TIVM в рамках импорта шаблона VM или VM, запущенной в vCenter, устанавливается значение <code>YES</code> . Это защищает VMDK-файл от удаления из vCenter по команде TIVM. В случае отсутствия атрибута или при значении <code>NO</code> VMDK-файл будет удален при удалении образа из TIVM.

5.4.7 Теги и категории, как дополнительные признаки ВМ

Система управления vCenter позволяет приписывать ВМ дополнительные признаки, выражаемые тегами и категориями. Эти признаки сохраняются в импортированных шаблонах ВМ. Их использование упрощает учет, поиск и систематизацию ВМ.

Тег задается следующим атрибутом в составе шаблона ВМ:

```
VCENTER_TAG = [  
  NAME = "Tag name",  
  DESCRIPTION = "Tag description",  
  CATEGORY = "Category name" ]
```

Имя категории указывается подчиненным атрибутом CATEGORY.

Шаблон может содержать множество блоков VCENTER_TAG.

6 Настройка виртуальных машин

6.1 Контекстуализация виртуальной машины

Контекстуализация VM – это процесс передачи VM при ее запуске служебной информации и ее использование для настройки VM. Обычно контекстуализация связана с передачей VM сетевой конфигурации (IP-адресов, сетевых масок, шлюзов, DNS-серверов), учетных данных пользователей и параметров виртуальных устройств (например, CD-ROM). Более глубокое применение данного механизма – запуск после загрузки VM скрипта, выполняющего ее подготовку к требуемым условиям применения.

Для обеспечения контекстуализации необходимо:

- 1) подготовить образ гостевой ОС, содержащей необходимые пакеты контекстуализации;
- 2) включить в шаблон VM данные контекстуализации.

Порядок решение этих задач несколько различается в зависимости от используемой технологии виртуализации (KVM и LXD, с одной стороны, и vCenter, с другой).

6.1.1 Подготовка образа гостевой ОС

Необходимо выполнить следующие действия.

1. С помощью TIVM запустить VM с гостевой ОС.
2. В этой VM загрузить из сетевых репозиториях ПО пакеты поддержки контекстуализации для данной гостевой ОС¹.
3. Установить загруженные пакеты.
4. Выключить VM (например, с помощью команды `tivm poweroff`, см. п. 5.3.4).
5. Сохранить образ загрузочного диска гостевой ОС, подготовленной для контекстуализации.

5.1. При использовании гипервизоров KVM и LXD это действие обеспечивает команда `tivvm disk-saveas` (п. 5.3.7).

5.2. При использовании виртуализации VMware:

- создание шаблона по VM осуществляется средствами vCenter;
- шаблон импортируется в TIVM с помощью команды `tivvcenter import` с параметром `--object templates`.

6.1.2 Включение данных контекстуализации в шаблон VM

Шаблон VM может включать раздел "CONTEXT", предназначенный для определения данных контекстуализации. Атрибут контекстуализации,

¹ Данные пакеты входят в состав поставки ПВК "ТИВМ".

передаваемый VM, представляет строка, содержащая выражение вида `name="value"`, где `name` – имя атрибута, а `value` – его значение.

Данные контекстуализации передаются VM через специальный логический раздел диска, называемый *контекстным устройством*.

Состав основных атрибутов контекстуализации приведен в табл. 6.1.

Таблица 6.1 – Основные атрибуты контекстуализации

Атрибут	Описание	Ограничения применимости
FILES	Список путей к файлам, включаемым в контекстное устройство. Пути разделяются пробелами.	
FILES_DS	Список файловых образов, включаемых в контекстное устройство. Имена разделяются пробелами.	Не поддерживается в vCenter
INIT_SCRIPTS	По умолчанию при использовании пакетов контекстуализации после загрузки VM на ней запускается скрипт инициализации <code>init.sh</code> . Если этот скрипт имеет другое имя, или необходимо запустить несколько скриптов инициализации, то их имена указываются в этом параметре через пробел.	
START_SCRIPT	Текст скрипта, который должен быть выполнен после запуска VM. Значением может быть не только скрипт оболочки ОС, но и вызов программных модулей, установленных на VM.	
START_SCRIPT_BASE64	То же, что <code>START_SCRIPT</code> , но в кодировке <code>base64</code> .	
TARGET	Имя устройства, подключаемое к контекстному устройству.	Не поддерживается в vCenter
DEV_PREFIX	Префикс для контекстного устройства (<code>sd</code> или <code>hd</code>).	--""--
TOKEN	YES – создать файл <code>token.txt</code> , который используется интерфейсом TIVGATE, позволяющим ПО, функционирующему на VM, получать из TIVM данные конфигурации и состояния средств виртуализации.	
TIVGATE_ENDPOINT	Адрес интерфейса TIVGATE.	
NETWORK	YES – автоматически заполнять сетевые параметры NIC, используемых в пакетах контекстуализации.	
SET_HOSTNAME	Имя хоста VM.	
DNS_HOSTNAME	YES – установить имя хоста VM на обратное имя DNS (с первого IP-адреса).	

Атрибут	Описание	Ограничения применимости
EC2_HOSTNAME	YES – установить имя хоста VM на основе главного IP-адреса в формате ip-A.B.C.D.	Только для ОС семейства Linux
GATEWAY_IFACE	Номер сетевого интерфейса, конфигурация которого контекстуализируется. Атрибут указывается, когда VM использует несколько виртуальных сетей, и необходимо выбрать одну из них, для которой задаются параметры настройки.	--""--
DNS	DNS-сервер для VM.	--""--
ETHx_IP	Адрес IPv4 для NIC.	
ETHx_IPV6	Адрес IPv6 для NIC.	Только для ОС семейства Linux
ETHx_NETWORK	Сетевой адрес интерфейса.	
ETHx_MASK	Сетевая маска.	
ETHx_GATEWAY	Шлюз IPv4 по умолчанию для NIC.	
ETHx_GATEWAY6	Шлюз IPv6 по умолчанию для NIC.	Только для ОС семейства Linux
ETHx_DNS	DNS-сервер для виртуальной сети.	
USERNAME	Учетная запись пользователя, создаваемая в гостевой ОС.	
PASSWORD	Пароль для пользователя USERNAME. Если атрибут USERNAME не задан, то пароль устанавливается для пользователя root.	
PASSWORD_BASE64	Пароль для пользователя USERNAME в кодировке base64. Если атрибут USERNAME не задан, то пароль устанавливается для пользователя root.	
CRYPTED_PASSWORD	Зашифрованный пароль для пользователя USERNAME. Если атрибут USERNAME не задан, то пароль устанавливается для пользователя root.	Только для ОС семейства Linux
CRYPTED_PASSWORD_BASE64	Зашифрованный пароль для пользователя USERNAME в кодировке base64. Если атрибут USERNAME не задан, то пароль устанавливается для пользователя root.	--""--
SSH_PUBLIC_KEY	SSH-ключ, добавляемый в файл authorized_keys для пользователя USERNAME (или файл root, если атрибут USERNAME не задан).	--""--
TIMEZONE	Временная зона, устанавливаемая в гостевой ОС. В Linux значение должно соответствовать имени в файле временных зон (usr/share/zoneinfo), в Windows – имени поддерживаемой временной зоны,	

Атрибут	Описание	Ограничения применимости
	список которых возвращает утилитой <code>tzutil /l</code> .	

Использование атрибутов контекстуализации позволяет формировать конфигурацию ВМ при ее запуске.

Таки, для передачи ВМ *сетевой конфигурации* необходимо указать атрибут `NETWORK="YES"`. В этом случае каждому NIC будет автоматически назначен IP-адрес. Дополнительные параметры (DNS-сервер, шлюз, сетевая маска и др.) также могут быть получены из контекста.

Передача ВМ *учетных данных пользователя* позволяет осуществлять его авторизацию после запуска ВМ. В Linux рекомендуется организовывать аутентификацию на основе открытого SSH-ключа. Для этого необходимо:

- создать такой ключ;
- добавить его в учетную запись пользователя (с помощью команды `tivuser update` ввести в шаблон пользователя атрибут `SSH_PUBLIC_KEY="<key>"`, см. разд. 6.2);
- добавить в раздел "CONTEXT" шаблона ВМ атрибут `SSH_PUBLIC_KEY="$USER[SSH_PUBLIC]KEY"`.

В результате в конфигурацию ВМ будет добавлен открытый SSH-ключ пользователя, ее создавшего.

Для поддержки доступа ПО, работающего на ВМ, к интерфейсу TIVGATE, в раздел "CONTEXT" должны быть добавлены атрибуты `TOKEN` и `TIVGATE_ENDPOINT`, *пример*:

```
CONTEXT = [
  TOKEN = "YES",
  TIVGATE_ENDPOINT = "http://192.168.0.1:5030"
]
```

Для обеспечения запуска ВМ *скриптов инициализации* служат атрибуты `START_SCRIPT`, `START_SCRIPT_BASE64` и `INIT_SCRIPTS`. В ОС семейства Windows могут использоваться скрипты на языке PowerShell. В Linux это может быть скрипт на любом языке, интерпретатор которого установлен в ОС и указан в шебанге.

6.1.3 Использование средств vCenter для настройки ВМ

Гипервизор vCenter реализует собственные средства контекстуализации гостевой ОС, называемые средствами настройки ВМ (vCenter customization). Они не совместимы с контекстуализацией, обеспечиваемой TIVM, и переопределяют сетевые настройки,

сформированные скриптами контекстуализации, заданными на уровне TIVM.

Контекстуализация vCenter предполагает выбор и применение *типового варианта настройки VM* при ее запуске, рассчитанного на определенное семейство гостевых ОС. Собственно параметры настройки формируются в vCenter. Состав типовых вариантов настройки VM, доступных при размещении ее на хосте виртуализации, управляемом vCenter, возвращается командой `tivhost show`.

TIVM связывает шаблон VM, функционирующей на базе vCenter, с типовым вариантом ее настройки путем указания имени данного варианта, как значения атрибута `VCENTER_CUSTOMIZATION_SPEC` в разделе "USER_TEMPLATE", например:

```
USER_TEMPLATE = [  
  VCENTER_CUSTOMIZATION_SPEC = "LinuxCustomization"  
]
```

Эта же связь может быть задана в шаблоне VM в web-интерфейсе TIVM (рис. 6.1).

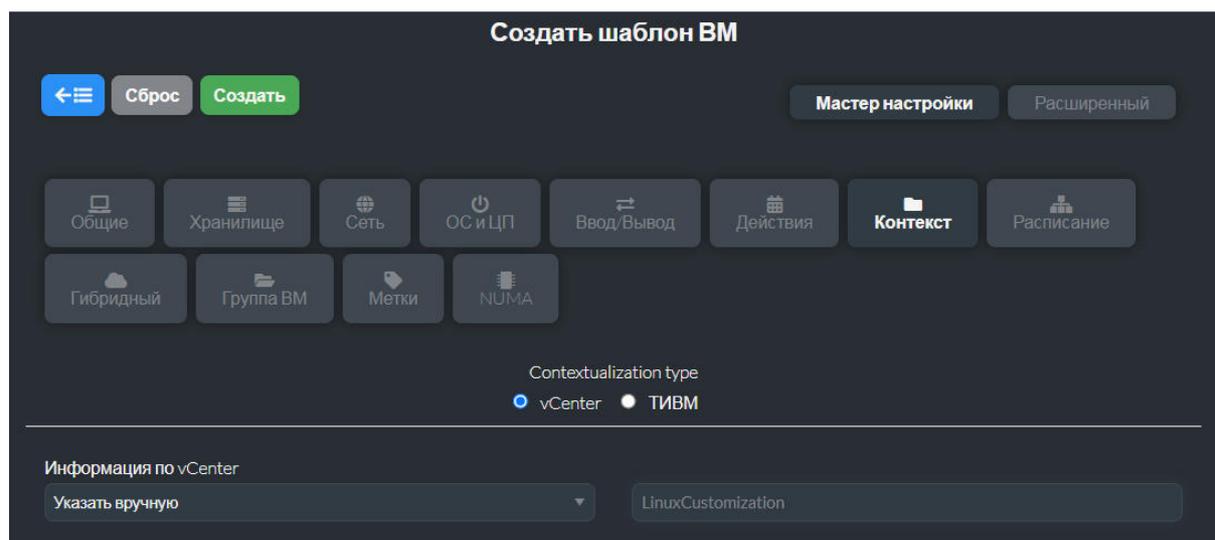


Рис. 6.1 – Настройка параметров консоли VM

При использовании средств настройки VM, реализуемых vCenter, сетевая конфигурация, применяемая VM, должна быть определена вне TIVM.

6.2 Наполнение облачной вычислительной среды

Возможности применения облачной вычислительной среды зависят от наличия в ней образов VM, соответствующих запросам конечных пользователей. Образы должны позволять создавать VM с гостевыми ОС,

конфигурациями и прикладным ПО, рассчитанными на бизнес-потребности пользователей.

Под *наполнением* облачной вычислительной среды понимается процесс подготовки образов дисков для ВМ и шаблонов ВМ, покрывающих запросы прикладных пользователей. Эту задачу решают администраторы облачной вычислительной среды.

Набор образов дисков ВМ для облачной среды, управляемой TIVM, может формироваться:

- 1) на основе готовых образов дисков с дистрибутивами ПО (полученных из внешних источников);
- 2) путем установки ПО и формирования соответствующих образов средствами TIVM.

6.2.1 Использование готовых образов дисков с дистрибутивами ПО

В данном варианте используются образы дисков ВМ с установленным ПО, подготовленные поставщиками этого ПО. Подобные образы создаются в форматах, рассчитанных на различные системы управления виртуализацией. Для применения их в TIVM они должны быть соответствующим образом преобразованы. При этом в образ могут быть добавлены настройки, адаптирующие его к условиям применения.

Для преобразования и адаптации образов дисков ВМ, основанных на виртуализации KVM, рекомендуется использовать открытое ПО *libguestfs* (<https://libguestfs.org/>). В сетевых репозиториях Linux представлены его пакеты для дистрибутивов RHEL, CentOS, Fedora, Debian, Ubuntu и Arch Linux.

Процедура подготовки образа диска ВМ состоит из следующих шагов.

1. Установить пакет *libguestfs-tools*. Например, в RHEL, CentOS и Fedora команда установки:

```
yum install libguestfs-tools
```

2. Загрузить из внешнего источника образ диска ВМ с установленным ПО, которое планируется использовать в облачной среде.

3. Загрузить из сетевых репозиториях ПО пакеты поддержки контекстуализации для гостевой ОС формируемого образа диска ВМ (см. п. 9.1.2). Эти пакеты будут включены в данный образ.

4. Создать образ дистрибутивного CD-ROM для установки пакетов поддержки контекстуализации. В данном примере образу будет приписан ярлык `PACKAGES` для упрощения его монтирования. Действие реализует утилита `genisoimage`:

```
genisoimage -o packages.iso -R -J -V PACKAGES packages-dir/
```

Ее параметры задают:

- `packages.iso` – имя файла образа дистрибутивного CD-ROM;
- `PACKAGES` – ярлык, приписанный образу;
- `packages-dir` – папка, в которой располагаются файлы пакетов поддержки контекстуализации.

5. Сформировать скрипт, выполняющий дополнительные действия по настройке образа диска VM и адаптации его к условиям применения. Этот скрипт будет выполняться в корневой файловой системе модифицируемого образа. Его содержимое будет разным в зависимости от исходного образа, представленного в нем ПО и гостевой ОС. Однако он должен обеспечивать установку пакетов поддержки контекстуализации в гостевой ОС (внутри образа).

Пример скрипта настройки для ОС Centos 7:

```
# Создание папки для монтирования дистрибутивного CD-ROM с
# пакетами поддержки контекстуализации
mkdir /tmp/mount

# Монтирование этого CD-ROM
mount LABEL=PACKAGES /tmp/mount

# Установка пакета, от которого зависят пакеты поддержки
# контекстуализации
yum install -y epel-release

# Удаление NetworkManager
yum remove -y NetworkManager

# Установка пакетов поддержки контекстуализации
yum install -y /tmp/mount/<имя пакета для целевой ОС>

# Корректировка конфигурации загрузчика ОС для предотвращения
# его зависания
sed -i --follow-symlinks 's/console=ttyS[^\ "]*//g'
/etc/default/grub /etc/grub2.cfg
```

6. Создать копию образа диска VM для исключения модификации исходного образа, полученного на шаге 2. Действие реализует утилита `qemu-img`:

```
qemu-img create -f qcow2 -b <имя файла исходного образа>
modified.qcow2
```

В данном примере создается образ в формате Qcow2 с именем `modified.qcow2`.

7. Адаптировать образ к условиям применения. Модификация состоит во включении в образ установленных пакетов поддержки контекстуализации и выполнении скрипта настройки, сформированного на

шаге 5. Действие реализует утилита `virt-customize` из пакета `libguestfs-tools`:

```
virt-customize -v --attach packages.iso --format qcow2
-a modified.qcow2 --run script.sh --root-password disabled
```

В данном примере ее параметры задают:

- a `modified.qcow2` – имя обрабатываемого файла образа;
- format `qcow2` – выходной формат образа – `Qcow2`;
- attach `packages.iso` – включение в образ дистрибутивного CD-ROM с пакетами поддержки контекстуализации, представленного образом `packages.iso`, созданным на шаге 4;
- run `script.sh` – запуск скрипта настройки с именем `script.sh`, сформированного на шаге 5;
- root-password `disabled` – отключение запроса корневого пароля;
- v – работа в режиме вывода диагностических сообщений.

8. Конвертировать образ в целевой формат, поддерживаемый средствами импорта TIVM. Конвертация необходима, даже если выходной формат `virt-customize` совпадает с требуемым входным форматом TIVM. Действие реализует утилита `qemu-img`.

Пример 1: конвертация образа, полученного на предыдущем шаге, в формат `Qcow2` (файл `final.qcow2`):

```
qemu-img convert -O qcow2 modified.qcow2 final.qcow2
```

Пример 2: конвертация образа, полученного на предыдущем шаге, в формат `VMDK` (файл `final.vmdk`):

```
qemu-img convert -O vmdk modified.qcow2 final.vmdk
```

9. Импортировать образ в хранилище образов TIVM. Действие может быть выполнено через web-интерфейс TIVM или с помощью CLI:

```
tivimage create --name imageName1 --path /var/tmp/final.qcow2
--driver qcow2 --prefix vd --datastore imageDs1
```

В данном примере в результате импорта файла `final.qcow2` в хранилище `imageDs1` создается образ диска VM с именем `imageName1` в формате `Qcow2`. Файл образа предварительно должен быть скопирован в папку, доступную средствам импорта (по умолчанию – `/var/tmp`).

6.2.2 Наполнение облачной вычислительной среды путем установки ПО и формирования соответствующих образов средствами TIVM

Данный вариант подходит в случае использования виртуализации KVM. Процедура состоит из следующих шагов.

1. Создать образ дистрибутивного диска с ПО. Например, следующая команда создает образ CD-ROM с дистрибутивом ОС Centos 7:

```
tivimage create --name centos7-install --path
http://buildlogs.centos.org/rolling/7/isos/x86_64/CentOS-7-
x86_64-DVD.iso --type CDRROM --datastore default
```

В данном примере исходный ISO-образ дистрибутивного диска загружается из сетевого репозитория (адрес указан в параметре `--path`) и помещается в хранилище `default`.

2. Создать образ диска VM для установки ПО. На данном шаге формируется пустой постоянный образ типа `DATABLOCK` (блок данных). Размер диска не следует объявлять чересчур большим, т.к. при создании VM он может быть увеличен. Действие может быть выполнено в web-интерфейсе TIVM (рис. 9.2) или с помощью команды:

```
tivimage create --name centos7 --description "Disk image for
Centos 7 installation" --type DATABLOCK --persistent
--prefix vd --driver qcow2 --size 10240 --datastore default
```

3. Создать шаблон VM с двумя дисками: первый – на основе пустого постоянного образа типа `DATABLOCK` (сформированного на шаге 2), второй – на основе образа дистрибутивного CD-ROM (сформированного на шаге 1). При загрузке диски должны использоваться последовательно. При первом запуске VM будет проведена установка ОС с CD-ROM, при последующих запусках – ОС будет загружаться с первого диска. VM должна иметь NIC для обеспечения загрузки пакетов поддержки контекстуализации. Также следует предусмотреть доступ к консоли VM через VNC для управления процессом инсталляции ОС.

Описанную конфигурацию VM представляет шаблон, создаваемый командой:

```
tivtemplate create --name centos7-vm --cpu 1 --memory 2G
--disk centos7,centos7-install --nic network
--boot disk0,disk1 --vnc --raw "INPUT=[TYPE=tablet,BUS=usb]"
```

4. Создать VM на основе шаблона, сформированного на предыдущем шаге (с помощью web-интерфейса TIVM или команды `tivtemplate instantiate`).

5. В консоли VM:

- выполнить установку ОС с диска, эмулирующего дистрибутивный CD-ROM;

- ввести конфигурацию NIC.

6. Перезагрузить VM (см. п. 8.3.5). Установленная на предыдущем шаге ОС будет загружена с первого диска.

7. Загрузить и установить пакеты поддержки контекстуализации.

Укажите параметры нового образа

← Сброс Создать
Мастер настройки Расширенный

Название:
 Описание:

Тип:
 Хранилище:

Этот образ является постоянным: Да

Расположение образа

Путь на сервере TIVM
 Закачать
 Пустой образ диска

Размер:

^ Расширенные настройки

Шина:
 Целевое устройство:

Драйвер для образов VM:

Рис. 6.2 – Создание пустого постоянного образа диска типа DATABLOCK

8. Выключить VM средствами ОС. После отключения VM она будет переведена TIVM в терминальное состояние `done`. Однако постоянный образ первого диска будет сохранен в хранилище.

9. Обеспечить доступ к образу диска с установленным ПО пользователям TIVM.

9.1. Сделать образ непостоянным:

```
tivimage nonpersistent centos7
```

9.2. Установить разрешения на операции с образом:

```
tivimage chmod centos7 744
```

В данном примере владелец образа имеет доступ ко всем операциям с ним, а прочие пользователи – к операциям категорий USE и MANAGE.