ООО «ТИОНИКС»

УТВЕРЖДЕН
RU.НРФЛ.00004 – 01 32 01-ЛУ

Эксплуатационная документация
Инструкция по развертыванию

облачной платформы «TIONIX»

(Tionix Cloud Platform)

Версия 2.x

RU.НРФЛ.00004-01 32 01

Листов 51

Москва, 2020

Содержание

1 Системные требования

Установка, обновление и удаление продуктов TIONIX (Cloud Platform) выполняется инженером по внедрению или системным администратором ОП, имеющим необходимый уровень подготовки и определенную квалификацию.

Допускается комбинированное взаимодействие между инженером по внедрению и системным администратором, предполагающее ассистирование с передачей основных навыков, которые потребуются в дальнейшем – на этапе пробной эксплуатации и/или после перевода облачной инфраструктуры в непрерывную эксплуатацию (т.н. производство или продакшен).

Требования, предъявляемые к выбору системного администратора, подробно изложены в документе Описание применения ОП TIONIX.

1.1. Аппаратные требования

Аппаратные средства размещаются в территориально-распределенном ЦОД. Для облачной платформы непринципиально, какие именно СВТ применяются, так как на уровне функциональных подсистем используются решения OpenStack[1].

На площадке ЦОД, используемой для строительства облачной инфраструктуры, должно быть выделено одно обособленное СВТ, предназначенное для размещения инфраструктурного сервера.

Используемое для сетевого подключения инфраструктурных узлов коммутационное оборудование должно обеспечивать высокую пропускную способность между сервером и другими СВТ. Для повышения надежности системы передачи данных может быть применено резервирование физических каналов.

При подключении СХД (SAN) должны использоваться только рекомендованные (совместимые) сетевые коммутаторы или специализированные решения, гарантирующие высоконадежный и высокоскоростной сетевой обмен с управляющими/вычислительными узлами.

Каждый инфраструктурный узел, размещенный в ЦОД, укомплектован несколькими сетевыми адаптерами различной пропускной способности и функционального назначения (LACP, IPMI, 10GE).

[1] https://ru.wikipedia.org/wiki/OpenStack

1.2. Среда функционирования

Среда функционирования ПО ТИОНИКС (далее – системное окружение) должна включать:

- рабочее окружение Python2;
- платформу OpenStack Queens3;
- OpenStack Dashboard4;
- реляционная СУБД (MySQL или PostgreSQL).

Вышеперечисленное ПО должно быть установлено на управляющий узел (контроллер).

Рекомендуемая ОС, обеспечивающая функционирование ПО – CentOS 7.8.

Установка продуктов TIONIX и вышеперечисленного ПО производится посредством системы управления пакетами, встроенной в ОС (зависит от дистрибутива Linux).

Примечание.

Использование хоста с репозиторием, в котором хранятся установочные файлы пакетов (*.rpm), может потребовать организации сетевой доступности.

1.3. Стороннее ПО

При проведении ПНР инженер может в разной мере применять стороннее ПО, не относящееся к составным частям платформы.

Ansible – система управления конфигурациями, написанная на языке Python, с использованием декларативного языка разметки для описания конфигураций. Используется для автоматизации настройки и развертывания ПО.

Cobbler – инфраструктурный сервер, используемый в автоматизации развертывания, обеспечивая массовую загрузку ОС в СВТ – управляющие или вычислительные узлы облачной платформы, – сихронизацию времени, централизованное хранение репозитория с ПО и обновлениями.

Docker – свободное ПО, используемое для автоматизации развертывания системы мониторинга облачной платформы.

---

[2] https://www.python.org
[3] https://docs.openstack.org/install-guide/InstallGuide.pdf
[4]https://docs.openstack.org/horizon/latest/doc-horizon.pdf

Prometheus – сервер системы инфраструктурного мониторинга.

Grafana – система графической визуализации метрик, собираемых сервером мониторинга (Prometheus).

1.4. Инфраструктурные компоненты

Облачная инфраструктура состоит из:

- сетевой инфраструктуры;
- инфраструктурного сервера;
- инфраструктурных узлов (control-/compute-);
- АРМ администратора (Ansible);
- система мониторинга (служб TIONIX).

Конфигурация *сетевой инфраструктуры* определена так называемой референсной (эталонной) архитектурой, содержащей указания типового способа физической и логической коммутации.

Перед установкой ПО OpenStack прежде всего должна быть настроена физическая сетевая инфраструктура, обеспечивающая поддержку сетевых потребностей работающих облачных служб. Физическая сетевая инфраструктура должна быть настроена для поддержки OpenStack Networking [5].

В некоторых случаях может применяться программно-определяемая сеть, построенная на технологии Open vSwitch [6]. Её применение наиболее востребовано при организации высокоскоростных виртуальных сетевых каналов, обеспечивающих сетевое взаимодействие между виртуальными узлами/машинами, происходящее внутри отдельно взятых вычислительных или управляющих нод.

Инфраструктурный сервер необходим для того, чтобы было возможным выполнение операций по развертыванию облачной платформы TIONIX Cloud Platform (установке ПО OpenStack/TIONIX[7]).

Под *инфраструктурными узлами* далее будут пониматься СВТ – хост-компьютеры, для которых выполняется управляемое с помощью Ansible или вручную развертывание

[5] https://docs.openstack.org/mitaka/networking-guide/intro.html
[6] https://docs.openvswitch.org/en/latest/intro/what-is-ovs, https://habr.com/ru/post/242741/
[7] https://docs.openstack.org/queens/deploy, https://tionix.ru/services/tionix-cloud-platform

Каждый из узлов, выделенных для использования в рамках облачной инфраструктуры, сразу после установки ОС настраивается на автономное выполнение команд[8].

Примечания.

При установке операционных систем с помощью Cobbler подготовка к Ansible-взаимодействию производится автоматически.

В определенных случаях может использоваться Docker[9].

Мониторингу на доступность, объем и производительность (метрики) должны подвергаться хосты, сети и хранилища данных, контролируемые узлами управления. Эти компоненты могут быть как физическими, так и виртуальными.

Существующая система мониторинга, устанавливаемая из репозитория ТИОНИКС, реализована с помощью контейнеров Docker Compose, а потому не требует развертывания системы виртуализации. Такая система может быть установлена как на АРМ администратора (Ansible), так и на инфраструктурный сервер.

В одном случае, система мониторинга будет полезной для инженеров, при проведении пусконаладочных работ (ПНР) и передаче облачной инфраструктуры в эксплуатацию. В другом случае, она может быть использована после ПНР, в качестве «бюджетной» и легковесной альтернативы внешним системам мониторинга.

1.5. Организационное обеспечение

Обеспечение физической сетевой инфраструктуры находится в зоне ответственности администратора сети и может включать в себя необходимость настройки и поддержания в исправном состоянии:

- физических коммутаторов;
- межсетевых экранов или маршрутизаторов;
- сетевых интерфейсов на серверных компьютерах (инфраструктурных узлах).

Следует провести определенные оргмероприятия по обеспечению безопасности доступа: к рабочему месту (АРМ) администратора, к инфраструктурным узлам и инфраструктурному серверу.

---

[8] https://docs.openstack.org/openstack-ansible/latest,
https://www.openstack.org/software/releases/queens/components/openstack-ansible
[9] https://docs.docker.com/get-started/overview

При активной поддержке политики импортозамещения на выделенное СВТ (ПК или ТК) рекомендуется установить сертифицированную ОС, класс защищенности которой определяется специалистом по информационной безопасности эксплуатирующей организации (оператором облачной платформы).

Для того, чтобы ПО TIONIX могло полноценно функционировать, необходимо позаботиться о получении и установке актуальных лицензий на его использование.

1.6. АРМ администратора

АРМ администратора может быть как отдельно-стоящим (стационарным или мобильным), так и виртуализованным. Виртуализованное АРМ дает некоторые эксплуатационные преимущества, в частности – может быть прозрачно или явно резервировано.

При использовании виртуализованного АРМ администратора, функционирующего на инфраструктурном сервере, достаточно оснастить рабочее место средствами ввода и отображения информации и высокопроизводительным тонким клиентом.

Для гарантии доступа из АРМ администратора к средствам управления облачной платформой необходимо обеспечить бесперебойность электропитания, а также высокую пропускную способность сетевого подключения (не ниже 1Гбит/с). Для повышения надежности или пропускной способности может быть использовано аггрегирование – связывание двух сетевых интерфейсов в один [10].

[10] https://itproffi.ru/obedinenie-setevyh-interfejsov-v-linux-nastrojka-bonding

2 Подготовительные действия

Ниже, в обычной последовательности изложения, указаны подготовительные действия, выполняемые при установке облачной платформы – TIONIX Cloud Platform. Любые действия, связанные с установкой и настройкой ПО ТИОНИКС, далее будут подразумеваться под термином «развертывание ПО».

Подготовка к развертыванию ПО на выделенной площадке выполняется в ручном режиме, при этом осуществляются следующие (пусконаладочные) операции:

- подготовка АРМ администратора;
- установка ОС и начальное конфигурирование инфраструктурного сервера;
- скачивание плейбуков и занесение исходных данных в конфигурационные файлы;
- настройка доступа к лицензиям.

Для выполнения операций, предшествующих развертыванию ПО, требуется подготовить АРМ администратора (см. ниже).

> Примечание.
>
> Практическим вопросам администрирования облачной платформы, построенной на основе ПО TIONIX (VDC), посвящена серия книг, объединенная в документ Руководство администратора.

2.1. Подготовка АРМ администратора

В качестве АРМ администратора следует использовать персональный компьютер (ноутбук) с установленной операционной системой Linux и ПО Ansible[11].

На АРМ должен быть установлен веб-браузер, поддерживаемый операционной системой Linux (дистрибутивы Ubuntu 16+, CentOS 7, ALT Linux 7 и т.п.). Это обеспечит использование веб-интерфейса, предоставляемого модулем TIONIX.Dashboard (см. документ Руководство по эксплуатации. ПО TIONIX VDC).

Рекомендуемый к установке веб-браузер:

– Google Chrome не ниже версии 43;
– Firefox не ниже версии 45.

[11] https://www.ansible.com/overview/how-ansible-works

Для управления выполнением сценариев автоматизированного развертывания потребуется установить программную утилиту ansible[12]. С её помощью выполняются команды на удаленном инфраструктурном узле [13].

Для работы со сценариями развертывания, скачиваемыми из репозитория, может потребоваться ПО Git[14]. Пакет **git**, содержащий это ПО, устанавливается из стандартного репозитория ОС (используемого дистрибутива Linux).

Ниже, сверху вниз, перечислены подготовительные действия, выполняемые с помощью АРМ администратора перед началом проведения ПНР. Установка виртуального сервера не является обязательной и выполняется на усмотрение инженера по внедрению облачной платформы TIONIX.

2.1.1. Настройка подключения к инфраструктурному серверу

Необходимо, прежде всего, обеспечить подключение к инфраструктурному серверу.

Для этого следует сгенерировать новый ssh-ключ, выполнив команду:

```
$ ssh-keygen
```

После готовности инфраструктурного сервера, обеспечить подключение к его консоли (по SSH-ключу):

```
$ ssh-copy-id tionix@адрес_сервера
```

где:

tionix – пользователь, созданный по требованиям п.1.4 п.п.(3);

адрес_сервера – адрес инфраструктурного сервера (Cobbler) в сети PXE.

2.1.2. Установка средств управления конфигурацией

Установите в систему следующие пакеты:

- ansible 2.7.9 [15];
- python-openstackclient 3.14.0;
- python-shade 1.7.0-2;

[12] https://docs.ansible.com/ansible/latest/installation_guide/index.html
[13] https://docs.openstack.org/project-deploy-guide/openstack-ansible/latest/deploymenthost.html
[14] https://www.digitalocean.com/community/tutorials/how-to-install-git-on-centos-7
[15] https://xakep.ru/2018/09/12/ansible-deploy

- python-openstacksdk >=0.12.0;
- ipmitool 1.8.18 [16].

Примечание.

В случае установки из PIP добавить путь к переменной среды PATH:

```
export PATH=$PATH:/usr/local/bin/
```

2.1.3. Получение файлов с настройками типовой конфигурации

Создайте рабочую директорию проекта и распакуйте в неё плэйбук (ansible).

Актуальные плейбуки хранятся в репозитории TIONIX и доступны через Интернет, по ссылке (URL):

https://git.tionix.ru/rest/api/latest/projects/ES/repos/ansible_tionix_vdi_v2/archive?format=zip

Внимание. Плейбуки должны быть настроены для конкретной площадки с учетом референсной архитектуры (на основании частного ТЗ).

ВАЖНО! Релиз должен быть актуальным на дату проведения работ, способ получения релиза в данном документе не рассматривается. Архив предоставляется только по запросу.

Смените рабочую директорию на ~/ansible_tionix_vdi/<дистрибутив_Linux>.

2.1.4. Настройка инфраструктурного окружения

Выполните экспорт переменных окружения:

```
export http_proxy=адрес_инфраструктурного_сервера:8888
export https_proxy=адрес_инфраструктурного_сервера:8888
```

2.1.5. Настройка конфигурационного файла Ansible

Проконтролируйте содержание конфигурационного файла (ansible.cfg), в случае необходимости – отредактируйте его. Структура данного файла подробно рассмотрена в Приложении 1.

[16] https://ru.bmstu.wiki/index.php?title=OpenStack

2.1.6. Виртуальный сервер инженерного персонала

Для повышения эффективности по выполнению инженерным персоналом ПНР на различных площадках допускается использование заранее подготовленного *виртуального сервера*, который может быть скачан по ссылке:

http://storage.tionix.ru/pub/deploy_vm_appliance_ansible/<формат_образа>

где <формат_образа> может принимать значения:

appliance_kvm.img – образ диска для системы виртуализации KVM;

appliance_vbox.vdi – образ диска для системы виртуализации VirtualBox;

appliance_vmware.vmdk – образ диска для системы виртуализации VMware.

Скачанный образ виртуального сервера импортируется в существующую систему виртуализации (АРМ администратора) и запускается в работу при помощи виртуальной машины. Потребуются скачивание и установка соответствующего ПО (KVM, VirtualBox, VMware), в зависимости от образа.

Примечания.

Для повышения производительности виртуального сервера допускается использование аппаратных расширений виртуализации (настройка BIOS).

Система виртуализации может быть развернута как в АРМ администратора, так и на инфраструктурном сервере.

Внимание. Одновременное функционирование нескольких систем виртуализации в одном компьютере не рекомендуется.

Гипервизор KVM необязателен для скачивания, т.к. поставляется в виде пакетов репозитория операционной системы CentOS 7. Скачивание других систем виртуализации персонального пользования доступно по ссылкам:

https://www.virtualbox.org/wiki/Downloads

https://www.vmware.com/

После того как виртуальная машина с виртуальным сервером подготовлена (подключен образ диска, указано распределение выделяемых аппаратных ресурсов), необходимо настроить сетевой интерфейс и обеспечить маршрут к сетевой коммутации облачной инфраструктуры.

2.2. Инфраструктурное окружение

Т.к. платформа OpenStack основана на использовании стека протоколов TCP/IP, требуется наличие подготовленной физической и сетевой инфраструктуры, обеспечивающей непрерывное взаимодействие управляющих и вычислительных узлов.

Перед развертыванием платформы должны быть заранее подготовлены и постоянно доступны следующие инфраструктурные сервисы:

- NTP – сервер синхронизации времени [17];
- DNS – служба доменных имен, содержащая необходимые разрешения IP-адресов;
- сетевой доступ к репозиториям по протоколам HTTP/HTTPS.

Для инвентаризации и мониторинга узлов, управления сбросом питания и решения других «хозяйственных» задач используется IPMI[18].

2.2.1. Сетевая инфраструктура

Сетевая инфраструктура (Network), в обобщенном виде представленная на Рисунок 1, является важной составляющей облака, несущей транспортную функцию системы передачи данных. Она должна быть подготовлена раньше прочих – управляющей (Control) или вычислительной (Compute) инфраструктур.

К транспорту (системе данных) относится как физическая, так и логическая сетевая коммутация сетевой инфраструктуры виртуального ЦОД.

Физическая коммутация представлена сетевыми устройствами (коммутаторы, маршрутизаторы и т.п.), объединяющими инфраструктурные узлы в ЛВС, а также – структурированной кабельной системой ЦОД, независимо от способа распределения.

[17] https://www.altlinux.org/Ipmitool
[18] https://ru.wikipedia.org/wiki/Intelligent_Platform_Management_Interface

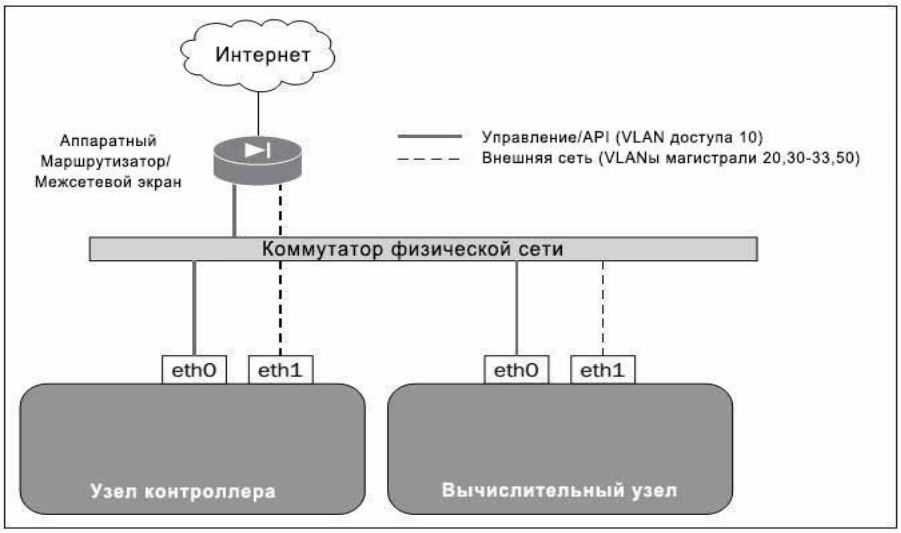

Рисунок 1 – Принцип коммутации физической сети

Логическая коммутация (Рисунок 2) определяет способы взаимодействия облачных сервисов и решается при помощи разграничений на каналы (VLAN) и сетевых интерфейсных адаптеров[19], которыми конфигурируются ноды – серверные узлы (CONTROL, COMPUTE) и система хранения данных (STORAGE).

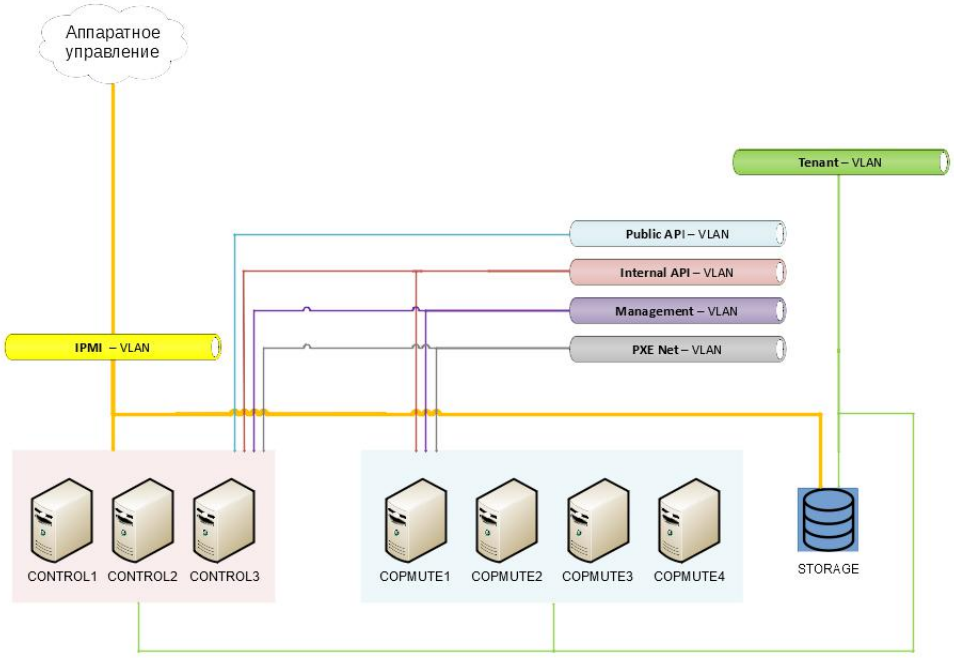

Рисунок 2 – Принцип логической коммутации сети (L3 Network)

[19] https://habr.com/ru/post/319080

2.2.2. Инфраструктурный сервер

В качестве инфраструктурного сервера (сокр. – ИС) используется компьютер с установленной ОС Linux и указанными ниже пакетами (СПО).

На инфраструктурный сервер должно быть установлено следующее СПО:

- Ansible==2.9.4;
- python-openstackclient [20];
- python-shade;
- python-openstacksdk >=0.29.

Примечание. Рекомендуется установка пакетов из rpm-репозитория (ТИОНИКС).

Рекомендованная операционная система – CentOS 7. Загрузочный образ с инсталлятором ОС доступен для скачивания по ссылке:

https://repo.tionix.ru/centos/centos7/<версия_ОС>/isos/

При разметке дискового пространства необходимо предусмотреть объем, предназначенный для хранения локальных репозиториев (более 30ГБ) в разделе файловой системы /var.

2.2.3. Системный пользователь

Должен быть создан системный пользователь tionix, наделенный:

- беспарольного способа выполнения команд (sudo);
- SSH – доступом (по ключу).

Ключевая пара должна быть сгенерирована на одном из узлов и скопирована на все остальные. Таким образом, у системного пользователя будет доступ ко всем узлам, организованный с помощью одинаковых ключей.

Также может быть произвольно сгенерирована ключевая пара, после чего открытая часть (публичный ключ) должна быть выложена на каждый из узлов, в директорию ~/tionix/.ssh/authorized_keys. Затем следует предоставить закрытую часть ключа, необходимую для выполнения работ с плейбуками (сценариями развертывания).

2.2.4. Плейбуки и репозиторий

Развернуть в рабочей директории плэйбук ansible, распространяемый в виде tar.gz-архива. Для этого должен быть доступен репозиторий, содержащий директорию playbooks:

https://git.tionix.ru/projects/AR/repos/repoadd/browse

[20] https://docs.openstack.org/operations-guide/ops-lay-of-the-land.html#command-line-tools

Потребуется выполнить следующие команды:

```
cd $HOME && wget

https://repo.tionix.ru/centos/playbooks/ansible_tionix_vdi_v2.tar.gz tar xvf
ansible_~/tionix_vdi_v2.tar.gz cd ansible_tionix_vdi_v2/playbooks
```

После распаковки следует проконтролировать содержание плейбука, с учетом референсной архитектуры (Раздел 3.2).

В случае необходимости, следует отредактировать конфигурационный файл в соответствии с содержанием типового конфигурационного файла (см. Приложения):

```
vim ansible.cfg
```

или

```
nano ansible.cfg
```

Установка ОС CentOS (7.8) на остальные узлы платформы выполняется либо вручную, для каждого узла, либо может быть выполнена с помощью средства автоматизации Cobbler (см. ниже).

Примечание.

Работа на сервере обеспечена из АРМ администратора, для этого может быть использовано безопасное подключение (по протоколу SSH).

2.2.5. Установка Cobbler (опционально)

Для подготовки ИС к использованию массовой загрузки ОС на узлы инфраструктуры при помощи ПО Cobbler необходимо выполнить следующие шаги:

1. Установить на инфраструктурный сервер ОС – CentOS (версия 7.x).

   Процесс установки не автоматизирован, поэтому установку следует выполнить самостоятельно. При разметке дискового пространства необходимо предусмотреть объем (ок. 30ГБ), предназначенный для локальных репозиториев (раздел /var).
2. Сконфигурировать сетевые интерфейсы в соответствии со схемами:

– схема размещения оборудования;

– схема структурная облачной платформы;

– план распределения IP- адресов.

3. Создать сервисного пользователя tionix (с паролем tionix) и обеспечить выполнение sudo без пароля, выполнив команду:

```
visudo /etc/sudoers
```

4. Добавить в конфигурационный файл (sudoers) строку:

```
tionix ALL=(ALL) NOPASSWD: ALL
```

ВАЖНО: Определить в настройках сетевого окружения максимальный размер MTU, применимый к конкретной сети, при необходимости изменить конфигурацию сетевых интерфейсов инфраструктурного сервера (Cobbler).

2.3. Настройка доступа к лицензиям

Доступ к лицензиям настраивается в следующей последовательности:

1. Получить файл лицензий на модули TIONIX (в виде RPM-пакета или tar.gz-архива).
2. Файл лицензий копируется на контроллере (управляющем узле).
3. Прежняя (просроченная) лицензия удаляется.
4. Устанавливается новая лицензия.
5. Выполняется перезапуск сервиса httpd и модулей TIONIX.

2.3.1. Установка лицензий

Для установки (новых) лицензий выполните одну из следующих команд, в зависимости от способа их получения (пакет/архив):

1. Лицензии в RPM – пакете:

```
yum install <полный_путь_к_файлу>/licensing.rpm
```

2. Лицензии в tar.gz – архиве:

```
pip install <полный_путь_к_файлу>/tionix-licensing-<версия.релиз>.tar.gz
```

где

<версия.релиз> – текущая версия/релиз ПО ТИОНИКС.

2.3.2. Удаление лицензий

Для удаления (просроченных) лицензий необходимо выполнить одну из следующих команд:

1. Для лицензий, установленных из RPM-пакета:

```
pip install <полный_путь_к_файлу>/tionix-licensing-<версия.релиз>.tar.gz
```

2. Для лицензий, установленных из архива:

```
pip uninstall tionix-licensing
```

3 Автоматизированное развертывание

Предполагается, что в исходном состоянии оборудование целевых компьютеров, таких как управляющие узлы — control, или вычислительные узлы — compute, а также инфраструктурного сервера, не имеет операционных систем (ОС).

Процесс автоматизированного развертывания ПО состоит из трех фаз:

1. Установка ОС и ПО на инфраструктурный сервер.

После установки операционной системы на инфраструктурный сервер должно быть установлено ПО Ansible[21]. Если будет использовано ПО Cobbler[22], то о нем необходимо ввести соответствующие инвентарные сведения[23].

2. Установка операционных систем на целевые компьютеры (узлы инфраструктуры).

На устройства хранения должна быть произведена установка или переустановка операционных систем, обеспечивающих нормальное функционирование узлов, выделенных на площадке (ЦОД) согласно *плана развертывания*. План должен быть разработан заранее, с учетом референсной архитектуры.

3. Установка ПО OpenStack и модулей TIONIX.

Установка ПО OpenStack и модулей TIONIX на целевые компьютеры осуществляется поверх ОС, посредством АРМ администратора (совместно с Ansible). При этом используются средства удаленного доступа – SSH.

Массовое первичное конфигурирование узлов может быть осуществлено посредством инфраструктурного сервера и технологии PXE.

> Внимание. Если на устройствах хранения управляющих или вычислительных узлов хранится какая-либо информация, то она будет удалена.

Прежде чем использовать плейбук(и) Ansible, необходимо его/их скачать из репозитория в рабочую директорию – <playbooks> – АРМ администратора или виртуального сервера.

В особых случаях допускается использование ручной установки.

---

[21] https://docs.openstack.org/openstack-ansible/latest
[22] https://cobbler.github.io/about.html
[23] https://docs.ansible.com/ansible/latest/user_guide/intro_dynamic_inventory.html

Планирование инсталляции основывается на следующих документах, подготовленных специально для выбранной площадки:

а) схема размещения оборудования;

б) план распределения IP адресов и VLAN;

в) схема сетевой связанности.

3.1. Подсистемы автоматизации

Перед выполнением операций по развертыванию (облачной платформы) необходимо убедиться, что выполнены все необходимые и достаточные подготовительные действия (Раздел 2).

Ansible – система управления конфигурациями[24], позволяющая выполнять:

- рутинные операции по настройке ОС (взаимодействие с Cobbler);
- установку необходимого ПО (с помощью плейбуков).

Установка операционной системы на узел выполняется при помощи IMPI. Однако, на начальном этапе массового развертывания может использоваться инфраструктурный сервер (сокр. – ИС), оснащенный ПО Cobbler[25]. Cobbler выполняет функции сервера сетевой установки, обеспечивая быстрое построение необходимой среды развертывания и управления процессом инсталляции ОС Linux на компьютеры или виртуальные машины.

Примечания.

Использование Cobbler требует настройки BIOS на всех узлах (УУ, ВУ), в части поддержки сетевой загрузки через PXE.

Факт установки ОС на узел фиксируеся, чтобы после этого (PXE-)загрузка не производилась повторно и узел начал работать самостоятельно.

После завершения основной фазы развертывания – раздачи загрузочных образов ОС по сети на все узлы площадки и отработки плейбука Ansible – ИС будет использован для служебных целей. Например, в процессе эксплуатации время от времени может требоваться переустановка операционных систем на узлы инфраструктуры (т.н. «сопровождение облачного ЦОД»).

[24] https://go.redhat.com/delivery-with-ansible-20181012
[25] https://cobbler.github.io/quickstart/

Дополнительно, ИС может выполнять такие служебные функции, как синхронизация времени по сети (NTP-сервер) или обеспечивать разрешение локальных имен в IP-адреса (DNS), предоставлять пакеты ПО OpenStack и TIONIX и их обновленные версии через rpm-репозиторий.

3.2. Референсная архитектура

Под референсной архитектурой будет пониматься совокупность шаблонов (решений), используемых при построении облачной инфраструктуры на базе программных продуктов ТИОНИКС.

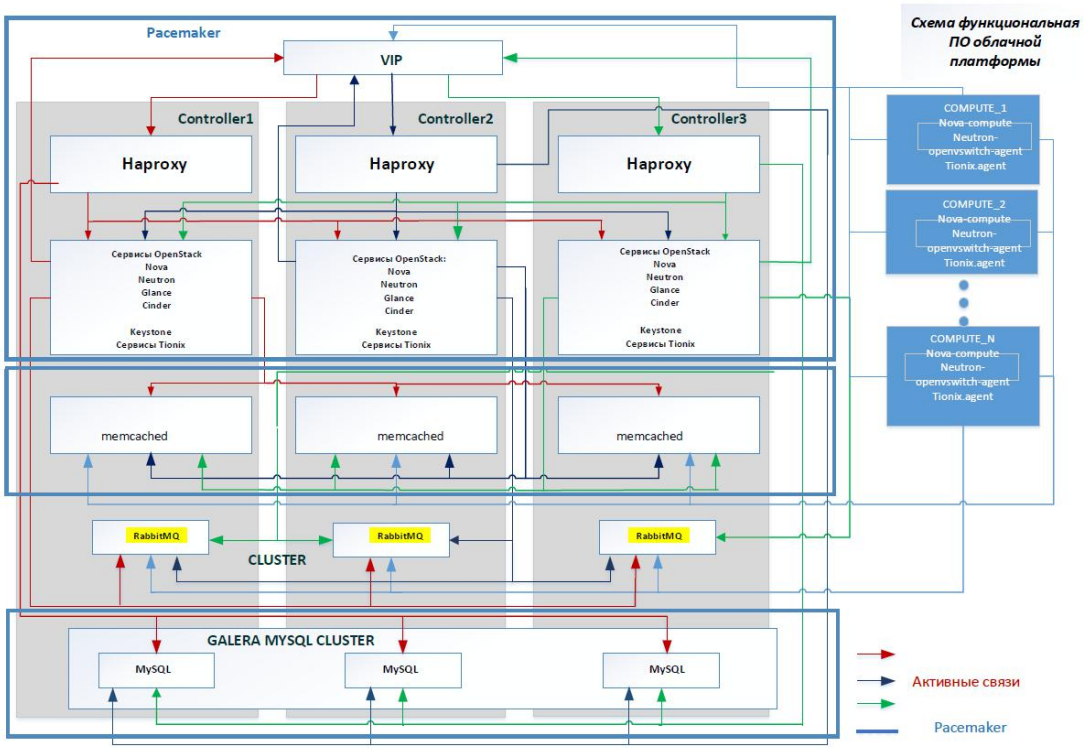

Рисунок 3 – Функциональная схема инфраструктуры TIONIX Cloud

Сокращенные обозначения на функциональной схеме (*Рисунок 3*):

- Controller1, Controller2, Controller3 – управляющие узлы (УУ);
- COMPUTE_1, COMPUTE_2, COMPUTE_N – вычислительные узлы (ВУ);
- HAProxy – прокси-сервер для повышения производительности серверной среды;

- memcached – ПО, реализующее сервис кэширования данных в оперативной памяти (на основе хеш-таблицы);
- MYSQL – реляционная система управления базами данных (с открытым исходным кодом);
- RabbitMQ – программный брокер сообщений;
- Virtual IP – виртуальный IP-адрес, через который осуществляется взаимодействие с облаком.

Все внешние запросы к инфраструктуре приходят на Virtual IP (сокр. VIP) – виртуальный IP-ресурс, создаваемый средствами Pacemaker[26]. Виртуальный адрес назначен одному из контроллеров (управляющих узлов). Если контроллер становится недоступным, то VIP переезжает на другой, доступный в рамках организованного кластера.

Типовой блок Haproxy обеспечивает отказоустойчивость сервисов OpenStack и баланс нагрузки[27].

На каждом (управляющем) узле создан один экземпляр memcached[28]. С помощью клиентской библиотеки memcached позволяет кэшировать данные в оперативной памяти. Отказоустойчивость обеспечена одновременным использованием трех экземпляров и конфигурируется на этапе развертывания.

RabbitMQ – программный брокер сообщений на основе стандарта AMQP. Он позволяет масштабировать систему и кластеризуется за счёт собственных средств построения кластера[29].

GALERA MYSQL CLUSTER – сервер баз данных MySQL/MariaDB, кластеризованный средствами GALERA[30].

3.3. Плейбуки Ansible и поток инсталляции

В плейбуках Ansible секции, в отличие от конфигурационного файла Ansible (Конфигурационный файл Ansible), имеют условный характер и вместе с параметрами образуют единый поток настроек (Описание инфраструктуры). Этот поток используется как сценарий,

[26] https://clusterlabs.org/pacemaker/doc/
[27] https://ru.wikipedia.org/wiki/HAProxy
[28] https://docs.openstack.org/ocata/ru/install-guide-rdo/environment-memcached.html
[29] https://habr.com/ru/post/193332, https://www.tech-notes.net/configure-rabbitmq-cluster
[30] https://galeracluster.com, https://ru.bmstu.wiki/Galera_Cluster

необходимый для автоматизированного выполнения операций при инсталляции облачной инфраструктуры IaaS-типа.

Ниже показан поток инсталляции компонентов OpenStack, выполняемый на основе определенных Ansible-ролей (*Рисунок 4*):

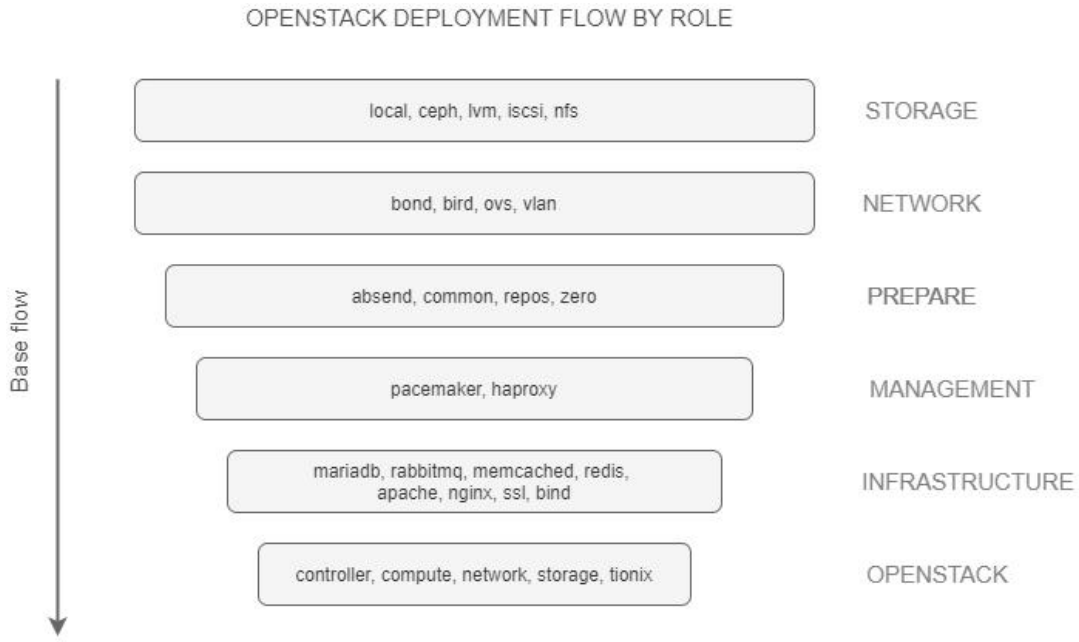

Рисунок 4 – Поток инсталляции компонентов OpenStack

Нормальная последовательность операций охватывает следующие стадии инсталляции:

- STORAGE: подготовительные операции, связанные с устройствами хранения;
- NETWORK: подготовительные операции, связанные с устройствами коммутации;
- PREPARE: операции по подготовке «нулевой» конфигурации;
- MANAGEMENT: средства наблюдения за состоянием узлов;
- INFRASTRUCTURE: создание инфраструктурной обвязки;
- OPENSTACK: надстройка компонентов верхнего уровня.

Обзор установки и настройки служб контроллера (УУ) OpenStack приведен на веб-странице документации OpenStack.

Автоматизация инсталляции обеспечивается плейбуками[31], «проигрываемыми» для Ansible-ролей.

Роли назначаются для определенных IP-адресов узлов, участвующих в развертывании.

Структура плейбука состоит, укрупненно, из нескольких групп конфигурационных параметров, каждая из которых начинается с трех черточек (прочерков, знаков минуса). Принято называть такие группы описаниями:

- HOSTS – описание хостов;
- VARIABLES – описание переменных;
- TASKS – описание задач;
- HANDLERS – описание обработчиков.

Описания хостов и переменных принято складывать в одну директорию, например – <playbooks>/inventory/*.yaml. Директория <playbooks> содержит обычные плейбуки.

3.4. Подготовка к использованию Cobbler

Использование инфраструктурного сервера при установке операционных систем – не является обязательным. Его использование следует согласовывать с конечным сценарием установки, учитывающим референсную архитектуру (см. выше).

Перед установкой Cobbler инфраструктурный сервер должен быть подготовлен должным образом. Также должен быть заполнен файл inventory/hosts.yaml (см. hosts.exapmle).

Для установки Cobbler с помощью Ansible в консоли АРМ администратора необходимо выполнить команды:

```
ansible -m ping allansible-playbook cobbler.yml -vv
```

Запустится процесс установки ПО (сервиса cobbler) на ИС, который может занять до получаса времени.

Примечание.

Более подробная информация по использованию Cobbler вынесена в Приложение 2 (Использование Cobbler).

[31] https://galeracluster.com, https://ru.bmstu.wiki/Galera_Cluster

После установки Cobbler в разделе, доступном из веб-меню «Configuration>> Systems», должны быть перечислены все узлы из групп control и nodes, указанных в файле inventory/vars.yml.

Отчет о текущих настройках сервера Cobbler может быть получен при помощи команды:

```
cobbler setting report
```

3.5. Развертывание ПО (с помощью плейбуков)

Последняя фаза представляет собой автоматизированную установку ПО OpenStack и TIONIX (VDC) на целевые компьютеры (узлы compute и control) посредством АРМ администратора. Установка ПО осуществляется с помощью ПО Ansible и плейбуков – типовых сценариев.

Плейбуки должны быть заранее распакованы из скачанного архива в рабочую директорию.

Прежде чем запускать плейбук(и), в файл /etc/hosts (АРМ администратора) необходимо добавить соответствие:

```
cobbler setting report
```

3.5.1. Проверка доступности (целевых) узлов

Для проверки доступности целевых узлов и корректности заполнения файла hosts.yml нужно провести проверку эхо-отклика:

```
ansible -m ping all
```

При нормальном отклике для каждого из опрашиваемых УУ будет выведен ответ в следующем формате:

```
control<порядковый_номер_УУ> | SUCCESS => {
«ansible_facts»: {
«discovered_interpreter_python»: «/usr/bin/python»
},
«changed»: false,
«ping»: «pong»
```

3.5.2. Выполнение сценариев развертывания

Для запуска автоматизированного развертывания спланированной облачной инфраструктуры в консоли или окне Терминала (АРМ администратора) выполнить команду:

```
ansible-playbook repoadd.yml deploy.yml -vv
```

Обычное время выполнения плейбука(ов) – от полутора до двух часов; необходимо дождаться завершения процесса развертывания ПО.

Примечание.

Учетные данные для доступа к платформе OpenStack содержатся в файле /root/admin-openrc.sh на каждом УУ.

3.5.3. Авторизация (с правами администратора)

По завершении развертывания необходимо выполнить подключение к веб-интерфейсу (используя VIP – _vip_address, назначенный из плейбука changeme.yml). В адресной строке браузера следует ввести URL (обслуживаемый модулем TIONIX.Dashboard) и, используя свои учетные данные, авторизоваться (см. Руководство администратора).

3.6. Настройка окружения OpenStack

На устанавливаемых в облачную инфраструктуру контроллерах (УУ) должно быть настроено окружение OpenStack. Для этого используется shell-скрипт – /root/admin-openrc.sh.

В АРМ администратора открыть консоль и в командной строке (окне Терминала) выполнить команду для входа в контроллер (по SSH):

```
ssh user@<VIP>
```

В ответ на запрос ввести пароль учетной записи (user).

Выполнить команду для вывода списка пользователей в текущем проекте можно следующим образом:

```
source /root/admin-openrc.shopenstack user list
```

Примечание. Команды, выполняемые с помощью утилиты клиента (openstack), могут запускаться как из АРМ администратора, так и из

любого другого узла инфраструктуры, на который установлено ПО клиента[32].

Типичное содержание скрипта (admin-openrc.sh) показано ниже:

```
#!/bin/bas
export OS_PROJECT_DOMAIN_NAME=default
export OS_USER_DOMAIN_NAME=default
export OS_PROJECT_NAME=admin
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=<пароль>
export OS_AUTH_URL=http://manage.tionix.loc:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
export OS_AUTH_TYPE=password
export GNOCCHI_ENDPOINT=http://manage.tionix.loc:8041
```

Окружение может быть просмотрено в Dashboard (с ролью admin). Для этого необходимо выбрать пункт веб-меню «Доступ к API» (см. Руководство по эксплуатации).

3.7. Проверка работоспособности

После завершения развертывания рекомендуется выполнить проверку работоспособности облачной платформы (см. Руководство по эксплуатации ОП TIONIX).

Методики функциональной диагностики и нагрузочного тестирования СУ ОП изложены в документе Методики испытаний инфраструктуры TIONIX.

Если на УУ установлена служба телеметрии, то её рекомендуется также проверить, как описано в документации OpenStack:

OpenStack (Verify operation).

Для выявления работоспособности определенных служб, функционирующих на управляющих или вычислительных узлах, может быть применена система мониторинга.

[32] https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html

4 Установка системы мониторинга

Система мониторинга на основе сервера Prometheus является независимой «надстройкой» над облачной платформой и может быть развёрнута, в зависимости от планируемого характера использования, практически на любом узле[33].

Примечание.

Описание использования системы мониторинга содержится в документе Руководство по эксплуатации ОП TIONIX.

Для упрощения манипуляций по развертыванию и снижения издержек может быть использована контейнерная реализация Grafana[34].

4.1. Системные требования и рекомендации

Минимальным требованием является наличие в операционной системе установленного ПО Ansible, обеспечивающего автоматизированное развертывание системы мониторинга (Раздел 2.1).

Если используется готовое инженерное решение, интегрированное с ПО TIONIX, то потребуется установить Docker и Docker-Compose, а также подготовить docker-compose к запуску.

Для корректного выполнения автоматизированного развертывания потребуется сеть Интернет, обеспечивающая доступ к git-репозиторию. Клонирование репозитория с плейбуками выполняется командой:

```
git clone ssh://git@git.tionix.ru/~f.vagapov/prometheus-d-c.git
```

Может потребоваться однократное добавление SSH-ключа в настройки доступа к репозиторию.

Примечание. Ключ должен быть заранее сгенерирован на рабочем месте (Раздел 2.1).

Пакет prometheus, содержащий сервер мониторинга, следует развернуть на отдельной виртуальной машине или на отдельной ноде. Например, может быть использован инфраструктурный сервер (Раздел 2.2.2).

[33] https://grafana.com/docs/grafana/latest/getting-started/getting-started-prometheus
[34] https://hub.docker.com/r/grafana/grafana

Альтернативным выбором будет один из вычислительных узлов, обслуживающих виртуализацию серверного ПО, либо виртуальная машина, развернутая в VDI проекте. Последнее позволит довольно быстро получить доступ к функционалу (WebGUI), однако, в случае каких-либо неполадок в подсистемах ВУ/гипервизора доступ к визуализации и аларминг пропадут.

> Внимание. Не рекомендуется использовать узлы контроллера OpenStack, так как это не только «утяжеляет» загрузку серверной ноды, но и может затронуть кластеризацию, от настроек веб-сервера до средств управления инфраструктурой (TIONIX.Dashboard).

Предпочтительным может оказаться выбор инфраструктурного сервера, т.к. принципиально ПО мониторинга интегрируется с некоторыми инфрастуктурными сервисами (NTP и т.п.) и архитектурно выполнено в парадигме «клиент-сервер», с применением ПО веб-сервера.

4.2. Автоматизированное развертывание

В файле hosts необходимо указать адреса сервера мониторинга и объектов мониторинга. Доступны два ansible-плейбуков, содержащих сценарии развертывания:

cn.yaml – развертывание экспортеров на вычислительных узлах;

server.yaml – развертывание сервера мониторинга.

Запуск плейбуков осуществляется командами:

```
ansible-playbook cn.yaml -i hosts  --ask-pass
ansible-playbook server.yaml -i hosts  --ask-pass
```

После корректного деплоя по адресу http://<server>:3000 должен быть доступен графический интерфейс системы визуализации Grafana (Раздел 4.3).

4.3. Проверка работоспособности

Проверка работоспособности системы мониторинга может быть выполнена входом через веб-интерфейс сервера Grafana и просмотром метрик любого доступного гипервизора. Потребуется лишь уточнить IP-адрес ВУ, на который установлено ПО гипервизора.

Вход в веб-интерфейс Grafana осуществляется по протоколу HTTP, через сетевой порт 3000. В веб-браузере следует ввести URL:

http://<IP-сервера-Grafana>:3000/

где:

IP-адрес_Grafana – адрес, который был указан в файле hosts (при автоматизированном развертывании).

Логин и пароль (по умолчанию): admin/tionix.

Откроется веб-диалог «Welcome to Grafana» с приглашением ко входу.

Выполните переход к информации, поставляемой от источника (datasource=default), узла (Job=node) с IP-адресом (ввести вместо localhost) и нажмите <Enter> для перечитывания информации (*Рисунок 5*).

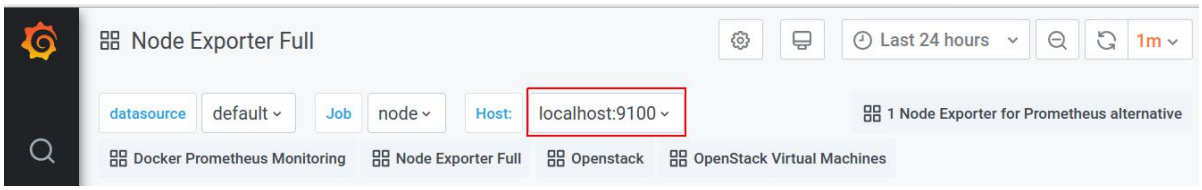

Рисунок 5 – Выбор хоста для мониторинга (Host:)

4.4. Использование контейнера Docker

Склонируйте репозиторий, содержащий плейбуки (Раздел 3.3). После успешного клонирования выполните сборку и запуск контейнера с помощью docker-compose[35]:

```
docker-compose build
docker-compose run
```

4.5. Ручное развертывание (ПО Grafana)

При установке ПО Grafana вручную потребуется выполнить обновление из репозитория[36]. При включении автоматического обновления следует воспользоваться инструкцией (ниже).

4.5.1. Установка произвольной версии Grafana

На странице загрузки Grafana download page необходимо выполнить ряд действий:

[35] https://habr.com/ru/post/309556/, https://habr.com/ru/company/ruvds/blog/450312/
[36] https://grafana.com/docs/grafana/latest/installation/rpm

1. Выбрать версию ПО (*Grafana version*).

   По умолчанию выбирается наиболее свежая версия ПО. Поле Version отображает только завершенные выпуски.

2. Выбрать выпуск/редакцию (an Edition).

   Выпуск Enterprise – рекомендуется к загрузке. Функциональная совместимость с «открытой» версией ПО Grafana сохранена, однако данный выпуск включает дополнительные возможности, раскрываемые при помощи лицензии.

   Выпуск Open Source – та же функциональность как у Enterprise выпуска, но с некоторыми ограничениями в функциональности.

3. Кликнуть: Linux или ARM.

Конечный выбор зависит от того, какая ОС используется в качестве среды функционирования. Скопируйте и вставьте код со страницы установки в командную строку и запустите.

Используйте типовые команды скачивания и установки, как показано ниже:

```
wget <rpm package url>
sudo yum localinstall <local rpm package>
```

Прямая установка rpm-пакета с помощью системы управления пакетами (YUM):

```
sudo yum install <URL_rpm-пакета>
```

4.5.2. Первичная настройка

Когда все компоненты системы мониторинга установлены, их надо настроить на взаимодействие (друг с другом).

Примечание. Есть 2 файла: environment (хосты) login (для прозрачности входа).

В первую очередь настраивается сервер Prometheus. Для этого потребуется отредактировать файл /etc/prometheus/prometheus.yml. Пример содержимого указан ниже:

```
global:
  scrape_interval:     15s
  evaluation_interval: 15s
```

```
# Alertmanager configurationalerting:
  alertmanagers:
  - static_configs:
    - targets:
      - localhost:9093


scrape_configs:
  - job_name: 'node'
    static_configs:
      - targets:
        - localhost:9100
      # - anotherhost:9100
```

Обратитесь к странице Configuration за подробным описанием опций, используемых при настройке окружения, логирования, БД и т.д.

4.5.3. Запуск сервера Grafana

Grafana-server выполняется в системе с правами пользователя Grafana, который создается при установке пакета в систему. Команды инициализации системы (systemd) в большинстве случаев работоспособны, однако некоторые устаревшие системы Linux могут потребовать использование init.d. Инсталлятор подскажет правильные команды.

Допускается два сценария запуска сервера Grafana, изложенных ниже. Выбор сценария зависит от особенностей реализации ОС (Linux).

4.5.3.1. Запуск сервера (systemd)

Для запуска (системной) службы и проверки статуса используйте команды:

```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl status grafana-server
```

Для настройки автозапуска сервера (Grafana) на этапе инициализации ОС необходимо выполнить команду:

```
sudo systemctl enable grafana-server
```

4.5.3.2. Запуск сервера (init.d)

Для запуска (системной) службы и проверки статуса используйте команды:

```
sudo service grafana-server start
sudo service grafana-server status
```

Для настройки автозапуска сервера (Grafana) на этапе инициализации ОС необходимо выполнить команду:

```
sudo /sbin/chkconfig --add grafana-server
```

4.6. Удаленный доступ к Grafana

Веб-интерфейс обслуживается веб-сервером и доступен по адресу http://localhost:3000/. Для доступа к веб-интерфейсу извне вместо localhost необходимо использовать IP-адрес или доменное имя. Для этого потребуется «открыть» порт 3000.[37]

Если в ОС работает системная служба *firewalld*[38], достаточно выполнить команды добавления правила и перезагрузки службы:

```
# firewall-cmd --zone=public --add-port=3000/tcp –permanent
# systemctl reload firewalld
```

Если в ОС работает системная служба *iptables* (устаревший Linux), то следует открыть файл конфигурации любым доступным текстовым редактором:

```
# nano /etc/sysconfig/iptables
```

или

```
# mcedit /etc/sysconfig/iptables
```

В разделе ‘OUTPUT ACCEPT‘ добавить строку:

```
| -A INPUT -p tcp -m tcp --dport 3000 -m state --state NEW -j ACCEPT
```

и перезапустить iptables:

[37] https://hamsterden.ru/grafana-server/#open_3000
[38] https://www.dmosk.ru/miniinstruktions.php?mini=firewalld-centos

```
# sudo systemctl restart iptables
```

Примечание.

При использовании аппаратного файрволла системные службы ОС Linux (firewalld или iptables) могут быть отключены. Окончательное решение принимается в зависимости от референсной архитектуры или ЧТЗ.

5 Обновление модулей и лицензий

Если установка модулей Tionix произведена из RPM-репозитория (yum install tionix-*.rpm), то для обновления модулей следует использовать ту же утилиту – yum (с ключем upgrade).

В случае, если ПО устанавливалось из python-репозитория, рекомендуется выполнить обновление как описано на сайте документации[39], выбрав определенный выпуск к релизу (2.6.x, 2.7.x и так далее).

5.1. Обновление модулей

Обновление ПО (OpenStack/TIONIX) из rpm-репозитория позволяет эффективно и с наименьшими трудозатратами выполнить многочисленные рутинные операции, используя механизм зависимостей.

Обновление модулей TIONIX осуществляется из того же rpm-репозитория, из которого была выполнена установка. Актуальные версии rpm-пакетов содержатся в репозитории, доступном по ссылке:

https://repo.tionix.ru/centos/centos7/tionix/queens/

Рекомендуется предварительно проверить, какие именно пакеты установлены. Для этого на УУ следует выполнить команду:

```
rpm -qa | grep tionix | sort
```

Вывод зависит от релиза, а также определенного набора модулей TIONIX. Например, для релиза 2.6.0 будет введен следующий список установленных пакетов:

```
python-tionix_client-2.6.1-1.el7.noarch
python-tionix_dashboard-2.6.4-1.el7.noarch
python-tionix_dashboard_theme-2.6.0-4.el7.noarch
python-tionix_licensing-2.0.1-20200128.el7.x86_64
python-tionix_monitor-2.6.0-4.el7.noarch
python-tionix_node_control-2.6.0-4.el7.noarch
```

[39] https://docs.tionix.ru

```
python-tionix_scheduler-2.6.0-4.el7.noarch
python-tionix_vdi_server-2.6.0-4.el7.noarch
```

Просмотр списка пакетов, доступных из текущего состояния репозитория, осуществляется командой:

```
yum list | grep python-tionix
```

Для выполнения обновления пакетов достаточно подключиться к управляющему узлу и выполнить сначала команду обновление репозитория (tionix-modules), а затем – команду обновления с указанием на репозиторий, содержащий модули TIONIX.

```
yum clean all
yum update --disablerepo=* --enablerepo=tionix-modules
```

Примечания.

Вышеуказанные команды должны выполняться с правами суперпользователя или sudo.

После выполнения обновления rpm-пакетов рекомендуется проверить список пакетов.

Кроме общих действий, выполняющих обновление ПО (на уровне rpm-пакета), для каждого модуля определен индивидуальный подход к обновлению. Он выражен в так называемой первичной настройке, связанной с особенностями взаимодействия модулей между собой, а также почти во всех случаях затрагивает модель данных OpenStack (обновление БД). Кроме того, поведение некоторых модулей зависит от сторонних системных служб, таких как веб-сервер.

Например, для перенастройки обновленного модуля TIONIX.NodeControl в частности, потребуется выполнить следующие команды:

```
openstack tnx configure -n tnx_node_control tnx_client
openstack tnx db migrate -n tnx_node_control
systemctl restart tionix-node-control-*
```

Для получения более подробной информации об обновлении или возврате к исходному состоянию (восстановлении работоспособности модулей TIONIX) необходимо перейти к веб-сайту официальной документации: https://docs.tionix.ru/.

На веб-сайте для каждого из модулей имеется раздел Администрирование, содержащей полное описание *процедуры обновления* и *плана восстановления*.

Для перехода к документации с учетом нюансов конкретной реализации необходимо осуществить переход по ссылке в формате:

`https://docs.tionix.ru/<номер_релиза>`

5.2. Обновление лицензий

Для обновления лицензий на модули ТИОНИКС необходимо выполнить следующие действия:

1. Подготовить rpm-пакет с новыми лицензиями на модули TIONIX.
2. Подключиться к контроллеру (по SSH) и установить лицензионный пакет.

```
yum reinstall licence.rpm
```

3. Перезапустить сервисы httpd и memcached, выполнив команды:

```
systemctl restart httpd
systemctl restart memcached
```

4. Перезапустить модули TIONIX:

```
systemctl restart tionix*
```

Примечание.

В конфигурации из трех контроллеров, повторить пункты 2-4 на каждом контроллере (если используется кластер).

6 Настройка конфигурации

Конфигурационные файлы, используемые при настройке продуктов TIONIX, делятся на два типа:

- общий файл конфигурации;
- модульный файл конфигурации.

Начало эксплуатации определено использованием общего файла конфигурации. После выполнения общей настройки считываются индивидуальные настройки, определенные в модульном файле.

Конфигурационные файлы могут содержать одинаковые секции и параметры, в таком случае (при расхождении параметров) используются настройки модульного файла конфигурации.

В случае отсутствия файлов конфигурации будут использованы параметры по умолчанию (из файлов yaml.example).

При установке любого модуля настраиваются оба вида конфигурационных файлов: модульный и общий (с расширением .yaml.example). И те, и другие файлы размещаются в директории /etc/tionix/.

6.1. Общий файл конфигурации

Общий файл конфигурации используется всеми модулями, что позволяет настроить сразу все модули в одном файле. Общий файл конфигурации – tionix.yaml – создается сразу же после завершения установки модуля TIONIX.Client.

В конфигурационном файле (tionix.yaml.example) перечислены секции и их опции.

> Примечание. Если в описании параметров не указано иное, то значения параметров чувствительны к регистру.

6.2. Модульный файл конфигурации

Каждому из модулей TIONIX может быть сопоставлен файл конфигурации (Перечень конфигурационных файлов инфраструктуры), название которого содержит коренное название модуля, установленного из репозитория (Раздел 2).

Название всех модульных файлов содержит суффикс (расширение):

- dashboard.yaml: конфигурация модуля TIONIX.Dashboard;
- monitor.yaml: конфигурация модуля TIONIX.Monitor;
- node_control.yaml: конфигурация модуля TIONIX.NodeControl;
- scheduler.yaml: конфигурация модуля TIONIX.Dashboard;
- vdi_server.yaml: конфигурация модуля TIONIX.VDIserver.

Примечание.

Полный перечень модулей приведен в документе Описание применения.

Пример содержимого модульного файла конфигурации для TIONIX.Dashboard:

```
DB:
  ENGINE: 'django.db.backends.mysql'
  NAME: 'tionix_dash'

NEUTRON_VERSION: 2

KEYSTONE:
  network_service_name: 'network'
  identity_service_name: 'identity'

SENTRY:
  ENABLED: False
  LOG_LEVEL: ERROR
```

Приложение 1. Конфигурационный файл Ansible

Конфигурационный файл ansible.conf[40] – текст, который состоит из логических секций, названия которых заключены в квадратные скобки:

- [defaults];
- [privilege_escalation];
- [ssh];
- [inventory].

В каждой секции построчно прописаны ключевые пары, в формате:

<ключ> = <значение>

Некоторые ключевые пары могут быть закомментированы, для этого в начало текстовой строки, содержащей ключевую пару, вписан символ решетки (#).

Секция [defaults]

```
callback_whitelist = profile_tasks

inventory = ./inventory

#inventory = hosts.yml

private_key_file = ~/.ssh/my_own_vagrant

host_key_checking = false

remote_user = vagrant

timeout = 60

become_user= root

gathering = smart

strategy_plugins = plugins/mitogen/ansible_mitogen/plugins/strategy
```

Секция [privilege_escalation]

```
#options for you version of "sudo"

become_flags = -H -S
```

[40] https://docs.ansible.com/ansible/latest/installation_guide/intro_configuration.html

Секция [ssh]

```
#Performance optimization

pipelining = true
```

Секция [inventory]

```
cache=true

cache_plugin=jsonfile
```

Приложение 2. Использование Cobbler

Для автоматизации управления инсталляциями ОС может быть использовано СПО Cobbler[41]. Cobbler, как правило, используется в массовой установке ОС на узлы инфраструктуры и требует соответствующей подготовки BIOS всех узлов, которые будут получать загрузочный образ ОС по сети. PXE-загрузка должна быть включена.

Более подробная информация о принципе хранения настроек Cobbler, а также их использования может быть найдена в Интернет [42].

Сценарий установки (скрипт cobbler.yaml) предполагает точное указание MAC-адресов сетевых интерфейсов, через которые осуществляется PXE-загрузка [43].

Установка пакетов Cobbler в Linux (Ubuntu) вместе с веб-интерфейсом может быть выполнена вручную:

```
$ sudo apt install cobbler cobbler-web
```

cobbler-web – необязательный компонент и необходим только в том случае, если планируется использование веб-интерфейса.

Веб-интерфейс

По окончании процесса установки из АРМ администратора следует проверить доступность и корректность настройки (инфраструктурного сервера), переходом из окна веб-браузера по ссылке:

https://<IP-инфраструктурного-сервера>/cobbler_web

Для входа в веб-интерфейс потребуется ввести логин/пароль: **cobbler/cobbler**.

После входа в веб-интерфейс в Меню (область главной страницы слева) будут представлены следующие группы:

- Configuration;
- Resources;
- Actions;
- Cobbler.

[41] https://developer.ibm.com/technologies/linux/articles/l-cobbler
[42] https://cobbler.readthedocs.io/en/latest/_modules/cobbler/settings.html
[43] https://xakep.ru/2016/06/15/cobbler

Рекомендуется перейти выполнить проверку настроек сервера Cobbler, перейдя к следующим пунктам веб-меню (ссылкам):

- Configuration >> Settings;
- Cobbler >> Check.

Примечание.

Отчет о текущих настройках сервера Cobbler может быть также получен при помощи команды:

```
cobbler setting report
```

Проверка работоспособности системной службы

Выполните подключение к инфраструктурному узлу по SSH с правами суперпользователя:

```
ssh root@<IP-инфраструктурного-сервера>
```

После успешного входа в (удаленную) консоль следует выполнить команду:

```
service cobblerd status || service cobblerd start
```

Если служба не была запущена, то она должа запуститься. Если она уже выполняется, то будет выведен её статус:

```
| cobblerd.service - Cobbler Helper Daemon
|    Loaded: loaded (/lib/systemd/system/cobblerd.service; enabled)
|    Active: active (running) since ...
|  Main PID: 1234 (cobblerd)
|  CGroup: name=systemd:/system/cobblerd.service
|           \- 1234 /usr/bin/python /usr/bin/cobblerd -F
```

Чтобы служба выполнялась на постоянной основе (запускалась автоматически), необходимо выполнить команду:

```
systemctl enable cobblerd.service
```

Проверка доступности (веб-интерфейса)

Если по каким-либо причинам веб-интерфейс недоступен, то из АРМ администратора необходимо подключиться к (виртуальному) узлу, на котором установлено ПО Cobbler и выполнить проверку работы веб-сервера:

```
systemctl status httpd || systemctl start httpd
```

Если сервер выполняется, то следует проверить настройки доступа к веб-контенту (индексному файлу).

ВАЖНО!

ОС CentOS по умолчанию работает в Enforcing-режиме. Уточнить текущий режим работы можно с помощью команды sestatus. Чтобы был выполнен вход в интерфейс пользователя Cobbler, потребуется перевести систему защиты (SELinux) из режима Enforcing в режим Permissive:

```
setenforce 0
```

Репозитории и синхронизация

На веб-странице https://<IP-инфраструктурного-сервера>/cobbler_web/repo/list отображается список всех добавленных репозиториев.

Для синхронизации репозиториев необходимо выбрать пункт «reposync» или выполнить команду:

```
# cobbler reposync
```

После внесения изменений в конфигурационные файлы и шаблоны следует синхронизировать данные [44], выполнив команду:

```
# cobbler sync
```

[44] https://www.tux.in.ua/articles/2143

Приложение 3. Конфигурационный файл службы Nova

Ниже представлен типовой конфигурационный файл службы Nova (Compute Service). Формат файла – INI – состоит из секций, заключенных в квадратные скобки, и параметров, настраиваемых парами: <параметр> = <значение>.

Файл содержит следующие секции:

- DEFAULT;
- api_database;
- database;
- api;
- vnc;
- glance;
- cinder;
- oslo_concurrency;
- cache;
- neutron;
- scheduler;
- placement;
- oslo_messaging_notifications.

```
[DEFAULT]
osapi_compute_listen = <IP-адрес>
metadata_listen = <IP-адрес>
my_ip = <IP-адрес>
state_path = /var/lib/nova
lock_path = $state_path/tmp
#rpc_backend = rabbit
enabled_apis = osapi_compute,metadata
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
#Restrict ephemeral disks
```

```
#max_local_block_devices = 0
#disk_allocation_ratio = 1.0
#ram_allocation_ratio = 0.9
#cpu_allocation_ratio = 5.0
transport_url=rabbit://openstack:123456@controller:5672

[api_database]
connection = mysql+pymysql://nova:123456@controller/nova_api

[database]
connection = mysql+pymysql://nova:123456@controller/nova

[api]
auth_strategy = keystone

[keystone_authtoken]
www_authenticate_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = 123456

[vnc]
server_listen = $my_ip
server_proxyclient_address = $my_ip
novncproxy_base_url = http://<IP-адрес>:6080/vnc_auto.html
novncproxy_host = <IP-адрес>
```

```
enabled = True

[glance]
api_servers = http://controller:9292
insecure = True

[cinder]
os_region_name = RegionOne
[oslo_concurrency]
lock_path = /var/lib/nova/tmp

[cache]
backend = dogpile.cache.memcached
enabled = True
memcache_servers = controller:11211

[neutron]
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = 123456
service_metadata_proxy = True
metadata_proxy_shared_secret = 123456
ovs_bridge = br-int

[scheduler]
```

```
discover_hosts_in_cells_interval = 300

[placement]
url = http://controller:8778
region_name = RegionOne
project_domain_name = default
project_name = service
auth_type = password
user_domain_name = default
auth_url = http://controller:35357
username = placement
password = 123456

[oslo_messaging_notifications]
driver = messagingv2
topics = notifications
```

Термины, сокращения и определения

Термин	Определение
АРМ	автоматизированное рабочее место (администратора инфраструктуры).
ВУ	вычислительный узел – СВТ определенной процессорной архитектуры; как правило, аппаратура ВУ имеет серверную конфигурацию: много процессорных ядер, большая ёмкость оперативной памяти, содержит высокопроизводительный сетевой интерфейс.
ИС	инфраструктурный сервер, позволяющий выполнять операции по развертыванию ОСПО (ОС, Python, OpenStack) на вычислительных и управляющих узлах облачной платформы.
ЛВС	локальная вычислительная сеть.
ОС	операционная система – системное ПО, обеспечивающее для ПО (ОСПО) среду функционирования и доступ к ресурсам аппаратного или виртуального узла (оперативной памяти, файловым системам, сетевым интерфейсам, системным библиотекам и системам управления репозиториями).
ОСПО	общесистемное ПО, с которому могут быть отнесены ОС, используемые как для хостинга, так и в виртуализации, в качестве гостевых ОС.
ПК	персональный компьютер (англ. сокр. – PC, personal computer).
ПО	программное обеспечение.
ПНР	пуско-наладочные работы.
СВТ	средство вычислительной техники.
СПО	свободное программное обеспечение.
СУ	система управления.
СУБД	система управления базами данных.
СХД	система хранения данных.
ТК	тонкий клиент (легковесное исполнение ПК, как правило, на базе микрокомпьютера).

УУ	узел управления (контроллер), специальным образом настроенный компьютер, позволяющий контролировать любые модификации в платформе (с помощью модели данных).
ЦОД	центр обработки данных, дата-центр.
ЧТЗ	частное техническое задание (применяется, как правило, к референсной архитектуре).
AMQP	(англ. Advanced Message Queuing Protocol) открытый протокол для передачи сообщений между компонентами системы. Отдельные подсистемы (или независимые приложения) могут обмениваться произвольным образом сообщениями через посредника – AMQP-брокер.
API	программный интерфейс приложения.
ARM	(англ. Advanced RISC Machine) микропроцессорная архитектура с сокращённым набором команд (RISC).
DNS	(англ. Domain Name System) компьютерная распределённая система для получения информации о доменах. Часто используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах (для работы сетевых протоколов в домене).
HTTP/HTTPS	(англ. HyperText Transfer Protocol) протокол передачи гипертекста, в т.ч. – с безопасным слоем SSL (HTTPS).
IaaS	(англ. Infrastructure-as-a-Service) «инфраструктура как услуга» – разновидность модели облачных вычислений, основанная на предоставлении по подписке информационно-технологических ресурсов, размещенных в ЦОД.
IPMI	(англ. Intelligent Platform Management Interface) – интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ.
LACP	(англ. Link Aggregation Control Protocol) протокол канального уровня в сетях передачи данных, использующих технологии Ethernet. Предоставляет стандартизированные средства контроля обмена информацией между двумя коммуникационными устройствами с целью динамического объединения нескольких физических каналов в один

	логический.
L3	уровень сетевой модели OSI или сетевого стека протоколов TCP/IP. OSI определяет способы взаимодействия друг с другом различных сетевых устройств, а сама реализация взаимодействия выполняется сетевыми протоколами. Сетевой уровень (англ. Network layer) предназначается для определения пути передачи данных. На этом уровне работает такое сетевое устройство, как маршрутизатор.
PXE	(англ. Preboot eXecution Environment) среда загрузки компьютера с помощью сетевой карты; позволяет исключить использование локальных носителей данных. При загрузке используются сетевые протоколы IP, UDP, BOOTP и TFTP.
SAN	(англ. Storage Area Network) архитектурное решение для подключения внешних устройств хранения данных, таких как дисковые массивы, ленточные библиотеки, оптические приводы к серверам. Подключение организовано таким способом, при котором операционная система распознает подключённые ресурсы как локальные.
SSH	(англ. Secure SHell) «безопасная оболочка» – сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (трафики при передаче файлов или команд шифруется).
SSL	(англ. Secure Socket Layer) криптографический протокол, который обеспечивает установление безопасного соединения между клиентом и сервером
URL	(англ. Universal Resource Locator) универсальный локатор ресурсов; синоним – веб-ссылка.
VIP	(англ. Virtual IP) виртуальный IP.
VLAN	(англ. Virtual Local Area Network) «виртуальная» локальная компьютерная сеть, которая представляет собой группу хостов с общим набором требований, взаимодействующих так, как если бы они были подключены к широковещательному домену, независимо от их физического местоположения.
YAML	(англ. Yet Another Markup Language) человекочитаемый формат сериализации данных, концептуально близкий к языкам разметки, но ориентированный на удобство ввода-вывода типичных структур данных (в языках программирования).

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					