ООО «АИДАТех»
119021, Г.Москва, ул Льва Толстого, д. 2/22 стр. 6
ИНН 9704261020
КПП 770401001

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «KAGECORE ML PLATFORM»**

# Инструкция по установке экземпляра

Листов 79

# Содержание

# Обозначения и сокращения

В настоящем документе применяют следующие сокращения и обозначения:

| | | |
|---|---|---|
| API | - | Application Programming Interface, программный интерфейс взаимодействия |
| BFD | - | Bidirectional Forwarding Detection, протокол, работающий на уровне интерфейса и протокола маршрутизации и предназначенный для быстрого обнаружения сбоев между двумя соседними маршрутизаторами, включая интерфейсы, каналы передачи данных и механизмы пересылки |
| BGP | - | Border Gateway Protocol, протокол динамической маршрутизации |
| BIOS | - | Basic Input/Output System, базовая система ввода-вывода |
| CA | - | Certificate Authority, доверенный орган, который выдает цифровые сертификаты |
| CHAP | - | Challenge Handshake Authentication Protocol, протокол аутентификации с косвенным согласованием |
| CIFS | - | Common Internet File System, сетевой протокол, который позволяет компьютерам получать доступ к файлам и ресурсам на других компьютерах по сети |
| CIMOM | - | CIM Object Manager, программный компонент, который реализует и управляет Common Information Model (CIM) |
| CPU | - | Central Processing Unit, процессор |
| DHCP | - | Dynamic Host Configuration Protocol, протокол динамической настройки узла |
| DNS | - | Domain Name System, система доменных имен |
| FQDN | - | Fully Qualified Domain Name, полное доменное имя |
| GB | - | Gigabyte, гигабайт |
| GPU | - | Graphics Processing Unit, графический процессор |
| GS | - | Gluster Storage, POSIX-совместимая файловая система |
| HA | - | High Availability, высокая доступность |
| HTTP | - | HyperText Transfer Protocol, протокол передачи гипертекста |
| HTTPS | - | HyperText Transfer Protocol Secure, расширение протокола HTTP для поддержки шифрования |

| | | |
|---|---|---|
| ICMP | - | Internet Control Message Protocol, протокол межсетевых управляющих сообщений |
| ID | - | Identifier, идентификатор |
| IOMMU | - | Input/Output Memory Management Unit, блок управления памятью для операций ввода-вывода |
| IOPS | - | Input/Output Operations Per Second, количество операций ввода-вывода в секунду |
| IP | - | Internet Protocol, межсетевой протокол |
| IPAM | - | IP Address Management, служба управления IP-адресами |
| IPMI | - | Intelligent Platform Management Interface, интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ |
| LUN | - | Logical Unit Number, логический (виртуальный) том |
| LVM | - | Logical Volume Manager, менеджер логических томов |
| MAC-адрес | - | Media Access Control address, уникальный идентификатор, присваиваемый каждой единице сетевого оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet |
| ML | - | Machine Learning, машинное обучение |
| MTU | - | Maximum Transmission Unit, максимальная единица передачи |
| NAS | - | Network Attached Storage, сервер для хранения данных на файловом уровне |
| NFS | - | Network File System, протокол сетевого доступа к файловым системам |
| NTP | - | Network Time Protocol, протокол сетевого времени |
| OVN | - | Open Virtual Network, виртуализация сети, отделяющая физическую топологию сети от логической |
| PCI | - | Peripheral Component Interconnect, стандарт компьютерной шины, используемый для подключения периферийных устройств к материнской плате компьютера |

| | | |
|---|---|---|
| POSIX | - | Portable Operating System Interface, набор стандартов, определяющих интерфейсы между операционной системой и прикладными программами, а также между библиотеками и командами |
| RAM | - | Random Access Memory, оперативная память |
| RBAC | - | Role-Based Access Control, управление доступом на основе ролей |
| RDP | - | Remote Desktop Protocol, протокол удалённого рабочего стола |
| RHDS | - | Red Hat Directory Server |
| SAN | - | Storage Area Network, сеть хранения данных |
| SDN | - | Software-Defined Networking, программно-определяемая сеть |
| SNMP | - | Simple Network Management Protocol, простой протокол сетевого управления |
| SPICE | - | Simple Protocol for Independent Computing Environments, Простой протокол для независимой вычислительной среды |
| SSD | - | Solid State Drive, твердотельный накопитель |
| SSH | - | Secure Shell, безопасная оболочка |
| SSL | - | Secure Sockets Layer, уровень защищенных сокетов |
| TCP | - | Transmission Control Protocol, протокол управления передачей |
| UDP | - | User Datagram Protocol, протокол пользовательских датаграмм |
| URL | - | Uniform Resource Locator, унифицированный указатель ресурса |
| VDSM | - | Virtual Desktop and Server Manager, агент, написанный на Python по средствам которого Engine взаимодействует с вычислительными узлами |
| VLAN | - | Virtual Local Area Network, виртуальная локальная сеть |
| VM | - | Virtual Machine, виртуальная машина |
| VNC | - | Virtual Network Computing, протокол удаленного доступа к рабочему столу компьютера |
| VXLAN | - | Virtual Extensible LAN, технология сетевой виртуализации, созданной для решения проблем масштабируемости в больших системах облачных вычислений |
| АРМ | - | автоматизированное рабочее место |
| БД | - | база данных |

| | | |
|---|---|---|
| ВМ | - | виртуальная машина |
| ГБ | - | гигабайт |
| ИТ | - | информационные технологии |
| ОС | - | операционная система |
| ПК | - | персональный компьютер |
| ПО | - | программное обеспечение |
| СХД | - | система хранения данных |
| ТБ | - | терабайт |
| ЦП | - | центральный процессор |

# Термины и определения

В настоящем документе применяют следующие термины с соответствующими определениями:

| | | |
|---|---|---|
| кластер | - | это логическая группа хостов с общими доменами хранения и ЦП одного типа (Intel или AMD). Если модели ЦП хостов относятся к разным поколениям, то используются только те функции, которые присутствуют во всех моделях. Виртуальные машины динамически распределяются между хостами кластера и могут перемещаться между ними в соответствии с политиками, заданными в кластере, и настройками виртуальных машин. Кластер является самым высоким уровнем, на котором могут определяться политики электропитания и разделения нагрузки |
| контейнер | - | легковесные запускаемые образы, в состав которых входит некоторое ПО и его зависимости. Поскольку в контейнерах виртуализируется операционная система, вы можете запускать контейнеры одинаково в любом совместимом окружении |

# Введение

Платформа KageCore ML Platform, включая модули KageCore ML Platform. Модуль тарификации, KageCore ML Platform. Модуль витрины сервисов и KageCore ML Platform. Модуль пользовательского мониторинга (далее – ПО, Система), предназначена для предоставления высокопроизводительных вычислительных ресурсов и совокупности сервисов (IaaS/PaaS/SaaS) в интересах одной или нескольких организаций и проектов, позволяя эффективнее обучать и эксплуатировать модели искусственного интеллекта, оптимизировать производственные и научно-исследовательские процессы и тем самым способствовать ускоренному развитию технологий ИИ.

ПО предназначено для решения следующих задач:

– формирование пула аппаратных средств (серверы с CPU и GPU, высокоскоростные сети, системы хранения), доступного пользователям из единого веб-интерфейса на основе квотирования, ролевой модели и механизма биллинга;

– автоматизация процесса заказа ресурсов (vCPU, RAM, GPU, объём дисков) через портал самообслуживания и программный интерфейс (API);

– использование готовых шаблонов (маркетплейса) с преднастроенными библиотеками и фреймворками (Keras, PyTorch, ONNX, Scikit-learn, TensorFlow), а также средствами разработки JupyterLab, VSCode;

– использование механизмов контейнеризации и виртуализации для гибкой оркестрации и оперативного масштабирования ML-заданий;

– предоставление инфраструктурных сервисов (виртуальные машины) и платформенных (среды разработки, БД, аналитические инструменты) в унифицированном виде;

– разграничение прав и ресурсов на уровне отдельных организаций, проектов и групп, что позволяет параллельно вести несколько сценариев инференса и обучения;

– внедрение роли и квот (RBAC), позволяющих ограничивать доступ и лимитировать объём ресурсов (CPU, GPU, память, хранилище);

– учёт и детальный биллинг (включая CPU, GPU, хранение данных), обеспечивающие прозрачность и справедливое распределение затрат между участниками;

– поддержка экспериментов и трассировки (логирование метрик, параметров, артефактов) с целью воспроизводимости и контроля качества обучаемых моделей;

– предоставление пользователям возможности оперативного выбора и автоконфигурации необходимых сервисов из унифицированного каталога (marketplace), содержащего преднастроенные модули, которые охватывают базовые и

прикладные сервисы, а также автоматизированного развертывания в контейнерной или виртуальной среде без ручной настройки;

– обеспечение администраторов и пользователей платформы инструментами наблюдения за компонентами платформы. Предоставление возможности сбора доступных метрик и их отображение, а также сбора анализа журналов приложений, операционных систем, при необходимости.

# 1 Компоненты Системы

## 1.1 Основные компоненты Системы

ПО KageCore ML Platform состоит из следующих основных составных частей:

– система управления виртуализацией (далее – СУВ);

– система контейнеризации и оркестрации (далее – СКО);

– система управление вычислительными ресурсами и обеспечивающая масштабирование уровня IaaS/PaaS/SaaS (далее – СУВР);

– инфраструктурный мониторинг;

– система машинного обучения.

В состав ПО также входят самостоятельные модули:

1) KageCore ML Platform. Модуль витрины сервисов.
2) KageCore ML Platform. Модуль тарификации.
3) KageCore ML Platform. Модуль пользовательского мониторинга.

# 2 Требования к инфраструктуре

## 2.1 Требования к вычислительным ресурсам

Минимальный требования к оборудованию для инсталляции Системы:

1) Сервер, с характеристиками 2-х CPU 24c/2.2GHz/RAM 256GB/2x SSD 480GB SATA3/4x 10/25GbE.

2) Сервер с GPU, с характеристиками 2-х CPU 16C/2.2GHz/RAM 256GB/2x SSD 480GB SATA3/2x GPU (определяется клиентом) /4x 10/25GbE.

3) Система хранения с файловым (NFS) или блочным доступом (iSCSI, FC), доступная серверу управления и серверу виртуализации с размеченными томами:

– 100 Гбайт (для установки менеджера управления СУВ).

– 7 Тбайт (хранилище для управляющих ВМ).

– 10 Тбайт (хранилище для создаваемых продуктов в виде ВМ).

4) Коммутаторы Ethernet 1/10/25 GbE с организованными сетями:

– Управления, подключаются интерфейсы управления серверов (BMC).

– Сети хранения данных, к которой подключены 2x 10/25GbE интерфейсы серверов и СХД.

– Сети обмена данными к которой подключены 2x 10/25GbE интерфейсы серверов.

Все процессоры должны поддерживать расширения Intel® 64 или AMD64 CPU, а также расширения аппаратной виртуализации AMD-V™ или Intel VT®. Дополнительно требуется поддержка флага No eXecute (NX).

Виртуализация должна быть включена в BIOS.

Для проверприменения изменений, выключите и перезагрузите хост после внесенных изменений.

Проверить какие расширения процессора доступны в системе можно следующей командой:

```
grep flags /proc/cpuinfo|head -n1|grep -Eo '(vmx|svm|nx)'
```

Если в выводе есть расширения – процессор поддерживает аппаратную виртуализацию. Если в выводе ничего нет, возможно, процессор поддерживает аппаратную виртуализацию, но в некоторых случаях производители отключают расширения виртуализации в BIOS. Если вы считаете, что аппаратная виртуализация выключена, обратитесь к руководству производителя материнской платы и BIOS.

# 3 Установка KageCore ML Platform

## 3.1 Система управления виртуализацией

Среда управления виртуализации (далее – СУВ) может быть развернута как в режиме Hosted Engine, так и в режиме Standalone. Рекомендуемый вариант развертывания для развёртывания Системы – Hosted Engine, при котором менеджер управления работает внутри ВМ, запущенной на хостах (серверах управления, серверах виртуализации), управляемых этой службой управления. ВМ и менеджер управления создаются и настраиваются при первоначальной установке кластера.

Основное преимущество режима Hosted Engine состоит в том, что отсутствует необходимость в отдельном хосте с ролью менеджера управления. Кроме того, Hosted Engine может работать в режиме высокой доступности.

Для доступа к Порталу администрирования и Пользовательскому порталу СУВ можно использовать браузеры на основе Google Chrome или Mozilla Firefox.

Использование различных расширений браузера может оказывать влияние на отображение элементов на портале администрирования.

Доступ к консолям виртуальных машин возможен с помощью клиента Remote Viewer (virt-viewer) на Linux и Windows. Для установки virt-viewer требуются права администратора.

Можно получить доступ к консолям виртуальных машин с помощью протоколов SPICE, VNC или RDP (только для Windows). Можно установить графический драйвер QXLDOD в гостевой операционной системе для улучшения функционала протокола SPICE. В настоящее время SPICE поддерживает максимальное разрешение 2560x1600 пикселей.

Менеджер управления должен быть развернут на хосте с установленной средой исполнения СУВ Node.

Не устанавливайте дополнительные пакеты после базовой установки, так как они могут вызвать проблемы с зависимостями при попытке менеджером управления установить пакеты из репозитория подсистемы управления виртуализацией.

Не включайте дополнительные репозитории, кроме тех, которые предоставляются с дистрибутивом подсистемы управления виртуализацией Node.

## 3.2 Требования к хранилищу

Хостам требуется хранилище для хранения конфигурации, журналов, дампов ядра и для использования в качестве пространства подкачки.

Минимальные требования и рекомендуемая схема разбиения хранилища на хостах:

- /(root) – 55 ГБ;
- /home – 1 ГБ;
- /tmp – 1 ГБ;
- /boot – 1 ГБ;
- /var – 15 ГБ;
- /var/crash – 10 ГБ;
- /var/log – 8 ГБ;
- /var/log/audit – 2 ГБ;
- swap – 1 ГБ.

## 3.3 Требования к устройствам PCI

Хосты должны иметь как минимум один сетевой интерфейс с минимальной пропускной способностью – 1 Гбит/с. Рекомендуется, чтобы у каждого хоста было минимум два сетевых интерфейса, один из которых предназначен для поддержки интенсивных сетевых действий, таких как миграция виртуальных машин. Производительность таких операций ограничена доступной пропускной способностью.

## 3.4 Требования к проброcу устройств

Для того, чтобы создаваемые СУВР ВМ могли использовать определенное PCI-устройство хоста (например, GPU), убедитесь, что следующие требования выполнены:

- процессор должен поддерживать IOMMU (например, VT-d или AMD-Vi);
- встроенное ПО должно поддерживать IOMMU;
- используемые корневые порты CPU должны поддерживать ACS или эквивалентную ACS возможность;
- устройства PCI должны поддерживать ACS или эквивалентную ACS возможность;
- все коммутаторы и мосты PCIe между устройством PCI и корневым портом должны поддерживать ACS. Например, если коммутатор не поддерживает ACS, все

устройства за этим коммутатором будут иметь общую группу IOMMU и могут быть назначены только одной виртуальной машине.

Проверьте спецификацию и технические характеристики производителя, чтобы убедиться, что ваше оборудование соответствует этим требованиям. Команда lspci -v может быть использована для получения информации о PCI-устройствах.

## 3.5 Требования к vGPU

Чтобы виртуальные машины могли использовать vGPU, хост должен отвечать следующим требованиям:

– vGPU-совместимый GPU от NVIDIA[1];

– ядро хоста с поддержкой GPU;

– установленный GPU с корректными драйверами, поддерживающими vGPU;

– предопределенный тип mdev_type, соответствующий одному из типов mdev, поддерживаемых устройством;

– операционная система виртуальной машины с поддержкой vGPU и установленными драйверами vGPU.

## 3.6 Сетевые требования

При планировании инфраструктуры учитывайте следующие требования:

1) Для менеджера управления необходимо, чтобы IPv6 оставался включенным на физическом хосте или виртуальной машине, в зависимости от того, где запущен менеджер управления.

2) Не отключайте IPv6 на ВМ HostedEngine и хостах, даже если в вашей сети его не используют.

3) Сетевой адаптер: рекомендуется использовать сетевые интерфейсы с пропускной способностью не менее 1 Гбит/с, предпочтительно — 10 Гбит/с или более. Допускается объединение интерфейсов в агрегированные каналы для повышения надежности и общей производительности.

[1] KageCore ML Platfrom, не включает в себя платные лицензии. Для активации vGPU, полльзователь должен самостоятельно приобрести необходимые лицензии NVIDIA.

4) Размер MTU: максимальный размер передаваемого блока данных должен быть установлен не менее 1500 байт. В случае использования хостов виртуализации, развернутых за пределами широковещательного домена управления, заданный размер должен поддерживаться на всем пути прохождения пакета управления от менеджера управления до хоста виртуализации. В противном случае могут возникнуть проблемы прохождения пакетов команд управления, содержащих флаг DontFragment. В случае использования 25, 40 и более гигабитных адаптеров рекомендуется использовать MTU 9000. В случае применения логических сетей SDN следует установить размер MTU сети управления (по умолчанию ovirtmgmt) как минимум в 1600 байт.

5) Задержки прохождения пакета: в общем случае задержка прохождения пакета от менеджера управления до хоста виртуализации не должна превышать 100мс. Ограничение вызвано механизмами работы служб высокой доступности виртуальных машин. Требование ужесточается в случае применения отдельных комбинаций функциональных возможностей подсистемы управления виртуализацией:

– Максимально допустимая задержка для сети обмена трафика SDS – не более 5 мс.

– Максимально допустимая задержка прохождения пакета от менеджера управления типа Hosted Engine к хосту при использовании служб обеспечения высокой доступности виртуальных машин – не более 7 мс.

– Максимально допустимая задержка при применении SDN между хостами виртуализации – не более 100 мс. Превышение данного порога может негативно повлиять на работу протокола BFD, используемого для определения доступности хоста, выбранного в качестве основного для работы логического маршрутизатора.

При планировании и настройке сетей в СУВ настоятельно рекомендуется ознакомиться с концепциями сетей и их использованием. Прочитайте руководства производителя сетевого оборудования для получения дополнительной информации об управлении сетями.

Логические сети могут поддерживаться с помощью физических устройств, таких как сетевые карты, или логических устройств, таких как bond. Bonding улучшает высокую доступность и обеспечивает повышенную отказоустойчивость, поскольку все объединенные сетевые карты должны выйти из строя, чтобы сам bond вышел из строя. Режимы объединении 1, 2, 3 и 4 могут использоваться для сетей виртуальных машин. Режимы 0, 5 и 6 не предназначены для сети виртуальных машин. СУВ по умолчанию использует режим 4.

Нет необходимости иметь одно устройство для каждой логической сети, поскольку несколько логических сетей могут совместно использовать одно устройство с помощью тегов виртуальных локальных сетей (VLAN) для изоляции сетевого трафика. Чтобы использовать эту функцию, тегирование VLAN должно поддерживаться на уровне коммутатора.

Ограничения на количество логических сетей в подсистемы управления виртуализацией:

– Количество логических сетей, подключенных к хосту, ограничено количеством доступных сетевых устройств в сочетании с максимальным количеством виртуальных локальных сетей (VLAN), которое составляет 4096.

– Количество сетей, которые могут быть присоединены к хосту за одну операцию, в настоящее время ограничено 50.

– Количество логических сетей в кластере ограничено количеством логических сетей, которые могут быть присоединены к хосту, поскольку сетевое взаимодействие должно быть одинаковым для всех хостов в кластере.

– Количество логических сетей в центре данных ограничено только количеством содержащихся в нем кластеров в сочетании с количеством логических сетей, разрешенных для каждого кластера.

В процессе развертывания подсистемы управления виртуализацией в варианте Hosted Engine временно используется локальная сеть из диапазона 192.168.0.0/16. По умолчанию используется адрес из сети 192.168.222.0/24, если эта подсеть используется, система проверяет другие сети 192.168, до тех пор, пока не найдет свободную подсеть. Если система не найдет свободную подсеть в указанном диапазоне - установка завершится ошибкой.

С помощью командной строки можно настроить сценарий развертывания на использование альтернативного диапазона сети /24 с помощью опции --ansible-extra-vars=he_ipv4_subnet_prefix=PREFIX, где PREFIX - префикс для диапазона по умолчанию. Например:

```
hosted-engine --deploy --ansible-extra-vars=he_ipv4_subnet_prefix=192.168.222
```

СУВ не создает сервер DNS или NTP, поэтому межсетевому экрану не нужно иметь открытые порты для входящего трафика.

По умолчанию СУВ разрешает исходящий трафик к DNS и NTP на любой адрес назначения. Если запрещается исходящий трафик, необходимо определить исключения для запросов на серверы DNS и NTP.

Менеджер управления СУВ и все хосты должны иметь подготовленные полные доменные имена, а также прямое и обратное разрешение имен.

Служба DNS не должна находиться внутри среды виртуализации. Все службы DNS, используемые средой виртуализации, должны быть размещены за пределами среды виртуализации.

Рекомендуется использовать сервер DNS вместо файла /etc/hosts для разрешения имен.

Для IPMI (Intelligent Platform Management Interface) и других механизмов Fencing (ограждения) межсетевому экрану не нужно иметь открытые порты для входящего трафика.

По умолчанию подсистемы управления виртуализацией разрешает исходящий трафик IPMI на порты с любым адресом назначения. Если вы запрещаете исходящий трафик, сделайте исключения для запросов IPMI или Fencing.

Каждый хост в кластере должен иметь возможность подключаться к устройствам ограждения всех остальных хостов в кластере. Если хосты в кластере получают ошибку (например: сетевая ошибка, ошибка хранилища…) и не могут функционировать как гипервизор, они должны иметь возможность подключения к другим хостам в центре данных.

## 3.7 Требования к межсетевому экрану менеджера управления

Менеджеру управления требуется, чтобы были открыты порты, указанные в таблице 1, чтобы пропускать сетевой трафик через межсетевой экран системы.

Сценарий установки автоматически настроит firewalld во время развертывания, но перезапишет старую конфигурацию, если вы используете iptables. Если вы хотите использовать iptables и оставить существующую конфигурацию, вы должны настроить его самостоятельно. Описанная здесь конфигурация межсетевого экрана предполагает конфигурацию по умолчанию.

Таблица 1 – Требования к межсетевому экрану менеджера управления**

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| M1 | - | ICMP | Сервер управления, сервер виртуализации | Менеджер управления | Необязательно. Может помочь при диагностике. | Нет |
| M2 | 22 | TCP | Система(ы), используемая для обслуживания менеджера управления | Менеджер управления | Безопасный доступ Secure Shell (SSH). Необязательно. | Да |
| M3 | 2222 | TCP | Клиенты, получающие доступ к консолям ВМ | Менеджер управления | Доступ через Secure Shell (SSH) для подключения к консолям виртуальной машины. | Да |
| M4 | 80, 443 | TCP | Клиент портала администрирования. Клиенты пользовательского портала, хосты виртуализации, клиенты | Менеджер управления | Предоставляет HTTP (порт 80, не зашифрованный) и HTTPS (порт 443, зашифрованный) доступ к менеджеру | Да |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| | | | REST API | | подсистемы управления виртуализацией. HTTP перенаправляет соединения на HTTPS. | |
| M5 | 6100 | TCP | Клиенты портала администрирования. Пользовательский портал | Менеджер управления | Предоставляет доступ через прокси-сервер websocket для веб-консольного клиента noVNC, когда прокси-сервер websocket работает на менеджере подсистемы управления виртуализацией. Однако, если прокси-сервер websocket работает на другом хосте, этот порт не используется. | Нет |
| M6 | 7410 | UDP | Сервер управления, сервер виртуализации | Менеджер управления | Если Kdump включен на хостах, откройте порт для fence_kdump на | Нет |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| | | | | | менеджере управления подсистемы управления виртуализацией. Fence_kdump не поддерживает шифрованное соединение. Вы можете вручную настроить этот порт, чтобы заблокировать доступ от хостов, которые не соответствуют требованиям. | |
| M7 | 54323 | TCP | Клиенты портала администрирования | Менеджер управления (прокси-сервер ImageIO) | Требуется для связи с ImageIO Proxy (ovirtimageioproxy). | Да |
| M8 | 6442 | TCP | Сервер управления, сервер виртуализации | Open Virtual Network (OVN) | Требуется для подключения к базе данных Open Virtual Network (OVN). | Да |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| M9 | 9696 | TCP | Клиенты провайдера внешней сети для OVN | Внешний сетевой провайдер для OVN | OpenStack Networking API | Да, с конфигурацией, сгенерированной при установке менеджера. |
| M10 | 35357 | TCP | Клиенты провайдера внешней сети для OVN | Внешний сетевой провайдер для OVN | OpenStack Identity API | Да, с конфигурацией, сгенерированной при установке менеджера. |
| M11 | 53 | TCP, UDP | Менеджер управления СУВ | DNS - сервер | DNS-запросы поиска от портов с номерами более, чем 1023 к порту 53 и ответы на них. Открыты по умолчанию. | Нет |
| M12 | 123 | UDP | Менеджер управления СУВ | NTP - сервер | NTP-запросы от портов с номерами более, чем 1023 к порту 123 и ответы на них. Открыты по умолчанию. | Нет |

## 3.8 Требования к межсетевому экрану хоста виртуализации

Хостам виртуализации необходимо, чтобы номера портов, перечисленные в таблице 2, были открыты, чтобы пропускать сетевой трафик через межсетевой экран системы.

Таблица 2 – Требования к межсетевому экрану хоста виртуализации

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| X1 | 22 | TCP | Менеджер управления СУВ | Сервер управления, сервер виртуализации | Secure Shell (SSH). Необязательно | Да |
| X2 | 2223 | TCP | Менеджер управления СУВ | Сервер управления, сервер виртуализации | Доступ через Secure Shell (SSH) для подключения к консолям виртуальной машины | Да |
| X3 | 161 | UDP | Сервер управления, сервер виртуализации | Менеджер управления СУВ | Простой протокол управления сетью (SNMP). Требуется только в том случае, если вы хотите, чтобы прерывания Simple Network Management Protocol отправлялись с хоста одному или нескольким внешним SNMP-менеджерам. Необязательно | Да (опционально). Поддерживается шифрование при использовании SNMPv3. Подробнее см. в разделе Параметры уведомлений о событиях в ovirt-engine-notifier.conf |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| X4 | 5900 - 6923 | TCP | Портал администрирования. Пользовательский портал | Сервер управления, сервер виртуализации | Удаленный доступ к гостевой консоли через VNC и SPICE. Эти порты должны быть открыты для обеспечения доступа клиентов к виртуальным машинам | Да (опционально) |
| X5 | 5989 | TCP, UDP | Менеджер объектов общей информационной модели (CIMOM) | Сервер управления, сервер виртуализации | Используется менеджерами объектов общей информационной модели (CIMOM) для мониторинга виртуальных машин, работающих на хосте. Требуется только в том случае, если вы хотите использовать CIMOM для мониторинга виртуальных машин в вашей среде виртуализации | Нет |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| X6 | 9090 | TCP | Менеджер управления подсистемы управления виртуализацией. Клиентские машины | Сервер управления, сервер виртуализации | Требуется для доступа к веб-интерфейсу Cockpit, если он установлен | Да |
| X7 | 16514 | TCP | Сервер управления, сервер виртуализации | Сервер управления, сервер виртуализации | Миграция виртуальных машин с использованием libvirt | Да |
| X8 | 49152 - 49215 | TCP | Сервер управления, сервер виртуализации | Сервер управления, сервер виртуализации | Миграция и ограждение (fencing) виртуальных машин с использованием VDSM. Эти порты должны быть открыты для облегчения как автоматической, так и ручной миграции виртуальных машин | Да. В зависимости от агента для ограждения, миграция осуществляется через libvirt |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| X9 | 54321 | TCP | Менеджер управления СУВ | Сервер управления, сервер виртуализации | Связь VDSM с менеджером подсистемы управления виртуализацией и другими хостами виртуализации | Да |
| X10 | 54322 | TCP | Менеджер управления СУВ (прокси-сервер ImageIO) | Сервер управления, сервер виртуализации | Требуется для связи с демоном ImageIO | Да |
| X11 | 6081 | UDP | Сервер управления, сервер виртуализации | Сервер управления, сервер виртуализации | Требуется, когда в качестве сетевого поставщика используется открытая виртуальная сеть (OVN), чтобы OVN мог создавать туннели между хостами | Нет |
| X12 | 53 | TCP, UDP | Сервер управления, сервер виртуализации | DNS | DNS-запросы поиска от портов с номерами более, чем 1023 к порту 53 и ответы на них. Открыты по умолчанию | Нет |

| Номер правила | Порт | Протокол | Источник | Назначение | Применение | Шифрование по умолчанию |
|---|---|---|---|---|---|---|
| X13 | 123 | UDP | Сервер управления, сервер виртуализации | NTP | NTP-запросы от портов с номерами более, чем 1023 к порту 123 и ответы на них. Открыт по умолчанию | Нет |

## 3.9 Хосты

Все хосты в кластере должны иметь одинаковый тип процессора. Процессоры Intel и AMD не могут находится в одном кластере.

## 3.10 Центр данных

Каждый центр данных должен иметь как минимум один домен хранения данных. Также центр данных может иметь не более одного домена хранения Экспорт. Домены типа Экспорт и ISO устарели, но при необходимости их можно создать.

Домен хранения может состоять либо из блочных устройств (iSCSI или Fibre Channel), либо из файловой системы (POSIX).

По умолчанию домены GlusterFS и локальные домены хранения поддерживают размер блока 4K. Размер блока 4K может обеспечить более высокую производительность, особенно при использовании больших файлов, а также необходим при использовании инструментов, требующих совместимости с 4K, таких как VDO.

В настоящее время подсистемы управления виртуализацией не поддерживает блочное хранилище с размером блока 4K. Вы должны настроить блочное хранилище в режиме (512b block).

Типы хранилищ, описанные в следующих разделах, поддерживаются для использования в качестве доменов хранения данных. Домены хранения «Экспорт» и «ISO» поддерживают только файловые типы хранения. Домен «ISO» поддерживает локальное хранение при использовании в локальном центре данных.

### 3.10.1 NFS

СУВ поддерживает NFS версий 3 и 4. Для производственных рабочих нагрузок требуется сервер NFS корпоративного уровня, если NFS не используется только в качестве домена хранения «ISO». Когда корпоративная NFS развернута на 10GbE, разделена с помощью VLAN, а отдельные службы настроены на использование определенных портов, она является одновременно быстрой и безопасной.

При расширении NFS хранилища СУВ распознает изменение размера хранилища данных. Дополнительной настройки на хостах или менеджере управления не требуется. Это дает NFS небольшое преимущество перед блочным хранилищем с точки зрения масштабирования и эксплуатации.

### 3.10.2 iSCSI

Для производственных рабочих нагрузок требуется сервер iSCSI корпоративного уровня. Если корпоративный iSCSI развернут на 10GbE, разделен на виртуальные локальные сети и использует аутентификацию CHAP, он является одновременно быстрым и безопасным. iSCSI также может использовать многоканальность (multipathing) для повышения высокой доступности.

СУВ поддерживает 1800 логических томов на блочный домен хранения. Разрешается использовать не более 300 LUN.

### 3.10.3 Fibre Channel

Fibre Channel является одновременно быстрым и безопасным, и его следует использовать, если он уже используется в целевом центре данных. Его преимущество заключается в низкой нагрузке на процессор по сравнению с iSCSI и NFS. Fibre Channel также может использовать многоканальность (multipathing) для повышения высокой доступности.

СУВ поддерживает 1500 логических томов на блочный домен хранения. Разрешается использовать не более 300 LUN.

### 3.10.4 Fibre Channel over Ethernet

Чтобы использовать Fibre Channel over Ethernet (FCoE) необходимо включить ключ fcoe в менеджере управления и установить пакет vdsm-hook-fcoe на хосты.

СУВ поддерживает 1500 логических томов на блочный домен хранения. Разрешается использовать не более 300 LUN.

3.10.4.1 Gluster Storage

Gluster Storage (GS) – это POSIX-совместимая файловая система с открытым исходным кодом. Три или более серверов конфигурируются как кластер Gluster Storage, вместо сетевых устройств хранения данных (NAS) или массива сети хранения данных (SAN).

Gluster Storage следует использовать по 10GbE и разделять с помощью виртуальных локальных сетей.

СУВ поддерживает гиперконвергентный вариант развертывания с помощью Gluster. Вместо того чтобы подключать подсистемы управления виртуализацией к внешнему хранилищу Gluster, существует возможность объединить подсистемы управления виртуализацией и Gluster в одной инфраструктуре, что позволяет снизить эксплуатационные расходы и накладные расходы.

### 3.10.5 POSIX-совместимые файловые системы

Другие POSIX-совместимые файловые системы могут использоваться в качестве доменов хранения в СУВ, если они являются кластерными файловыми системами, такими как Global File System 2 (GFS2), и поддерживают разреженные файлы и прямой ввод-вывод. Файловая система Common Internet File System (CIFS), например, не поддерживает прямой ввод/вывод, что делает ее несовместимой с СУВ.

### 3.10.6 Локальное хранилище

Локальное хранилище создается на отдельном хосте, используя собственные ресурсы хоста. Когда вы настраиваете хост на использование локального хранилища, он автоматически добавляется в новый локальный центр данных и кластер, в который не могут быть добавлены другие хосты. Виртуальные машины, созданные в кластере с одним хостом, не могут быть перемещены, ограждены или запланированы.

Для хостов локальное хранилище всегда должно быть определено в файловой системе, отдельной от / (root). Используйте отдельный логический том или диск.

## 3.11 Поддержка серверов каталога

Во время установки менеджер СУВ по умолчанию создает пользователя «**admin**» в домене «**internal**». Эта учетная запись предназначена для использования при первоначальной настройке среды и для устранения неполадок. Вы можете создать дополнительных пользователей во внутреннем домене с помощью утилиты ovirt-aaa-jdbc-tool. Учетные записи пользователей, созданные в локальных доменах, называются локальными пользователями.

Вы также можете подключить внешний сервер каталогов к СУВ и использовать его в качестве внешнего домена. Учетные записи пользователей, созданные во внешних доменах, называются пользователями каталога.

Поддерживается использование более одного сервера каталогов.

Для использования с подсистемы управления виртуализацией поддерживаются следующие серверы каталогов:

- Active Directory.
- Identity Management (IdM - на основе IPA).
- Red Hat Directory Server 9 (RHDS 9 - основан на 389DS).
- OpenLDAP.
- IBM Security (Tivoli) Directory Server.

Пользователь с правами на чтение всех пользователей и групп должен быть создан в сервере каталогов специально для использования в качестве сервисной учётной записи для менеджера управления подсистемы управления виртуализацией. Не используйте пользователя с правами администратора на сервере каталогов в качестве сервисной учётной записи для менеджера управления подсистемы управления виртуализацией.

## 3.12 Описание процесса установки

Установка включает в себя следующие шаги:

- установка СУВ на сервер управления и серверы виртуализации;
- развертывание и настройка менеджера управления.

### 3.12.1 Установка СУВ на сервер управления и серверы виртуализации

1) Запишите ISO-образ СУВ на USB, CD или DVD.
2) Запустите сервер управления и сервер виртуализации, на которых выполняется установка СУВ, загрузившись с подготовленного установочного носителя.
3) В меню загрузки выберите «Установить» и нажмите Enter.
4) Выберите язык по умолчанию и нажмите «Continue».
5) Выберите часовой пояс в разделе «Time & Date» и нажмите «Done».
6) Выберите раскладку клавиатуры в разделе «Keyboard» и нажмите «Done».

7) Выберите устройство, на которое нужно установить СУВ, в разделе «Installation Destination». При желании включите шифрование. В разделе «Storage configuration» выберите «Custom» и разметьте диск с помощью автоматического сценария, кликнув на «Click here to create them automatically». После автоматической разметки можете произвести необходимые изменения (изменить пространство для lvm разделов, не затрагивая атрибуты диска), учитывая требования раздела Требования к хранилищу.

8) Выберите сетевой интерфейс из раздела «Network & Host Name» - по кнопке «Configure» перейдите во вкладку конфигурации сети, настройте сеть и нажмите «Save». Активируйте настроенный интерфейс.

9) В поле «Host Name» введите имя хоста и нажмите «Apply». Переключите тумблер в состояние 1, находящийся рядом с данными об интерфейсе.

10) При необходимости настройте политику безопасности и Kdump.

11) Установите пароль пользователя root в разделе «Root password» и нажмите «Done».

12) Нажмите «Begin Installation».

13) После успешной установки перезагрузите хост, нажав «Reboot».

### 3.12.2 Установка СУВ на сервер управления и серверы виртуализации

Перенести установочный файл менеджера управления СУВ на сервер управления удобным способом в папку /root и установить пакет средствами ОС. Дополнительно необходимо отключить все имеющиеся репозитории:

```
dnf config-manager --disable '*'
```

Запустите процесс развертывания менеджера управления СУВ:

```
hosted-engine –deploy
```

Следуйте сообщениями в консоли. После успешного развёртывания менеджера управления виртуализации будет выведено сообщение:

```
[ INFO ] Hosted Engine successfully deployed
```

### 3.12.3 Проверка работоспособности

Зайдите в веб-интерфейс по адресу, который был указан для менеджера управления, нажмите Портал администрирования и аутентифицируйтесь с учетной записью администратора. При успешной аутентификации откроется веб-интерфейс управления СУВ.

### 3.12.4 Добавление хостов

Добавьте серверы виртуализации с GPU в кластер, через «Ресурсы» - «Хосты» - «Новый». Для обеспечения доступности менеджера управления в режиме Hosted Engine, в случае выхода из строя первого хоста, при добавлении дополнительных хостов выберите «Да» на вкладке «Hosted Engine».

### 3.12.5 Добавление хостов

Чтобы добавить новый домен хранения перейдите в «Хранилище» - «Домены» - «Новый домен» и добавьте домены хранения.

# 4 Система контейнеризации и оркестрации

## 4.1 Подготовка вычислительных ресурсов

СКО может быть установлена как в закрытом сетевом окружении без доступа к сети Интернет, так и в открытом.

Далее будут приведены минимальные рекомендации конфигурации для установки СКО. Фактические требования могут увеличиваться в зависимости от планируемой нагрузки.

Для установки СКО необходимо подготовить сервер для загрузки инсталляционных пакетов. Минимальная конфигурация приведена в таблице 3.

Таблица 3 – Минимальная конфигурация для установки

| **Наименование узла** | **Количество ВМ** | **vCPU** | **RAM, GB** | **Диск, GB** | **IOPS** |
|---|---|---|---|---|---|
| Хранилище артефактов | 1 | 4 | 16 | 128 | 300+ |

ВМ сервера хранения артефактов запускается в СУВ из образа. После загрузки ВМ необходимо выполнить вход в интерфейс сервера хранения артефактов с помощью учётной записи по умолчанию.

На главной странице интерфейса сервера хранения артефактов отображается статус системных сервисов СКО. При первом запуске общий системный статус «Not initialized» и статус сервисов «Not ready» являются нормой, поскольку инициализация сервера управления еще не выполнена.

Выполнить настройки:

– сменить пароль учётной записи администратора по умолчанию;

– установить параметры сетевого интерфейса. Для настройки сети используется текстовая версия утилиты Network Manager (TUI). С помощью Network Manager вы можете установить имя сервера, используя опцию Set system hostname, отредактировать существующее подключение, используя опцию Edit a connection, активировать и деактивировать существующее подключение, используя опцию Activate a connection. После инициализации сервера хранения артефактов изменить сетевые настройки невозможно. Для их изменения потребуется сброс к заводским настройкам и переустановка платформы, если она уже была развернута;

– выполнить настройку базового имени сервера хранения артефактов. Для этого выберите «Configure» на главной странице, перейдите в раздел «System settings»

и выберите опцию «Configure DNS base domain» и укажите базовое DNS-имя. Базовое DNS-имя используется для публикации внутренних компонентов, таких как сервис доставки ПО, сервис настройки ПО, хранилище образов и т.д. Для удобства использования и предотвращения конфликтов разрешения имен рекомендуется использовать поддомен в вашем корпоративном домене, например, application.mycompany.local. «dnsBaseDomain» не должен совпадать с «dnsBaseDomain» для СКО. Если имена будут одинаковыми, сервисы сервера хранения артефактов станут недоступны из СКО. После инициализации сервера хранения артефактов изменить базовое DNS-имя невозможно. Для его изменения потребуется сброс к заводским настройкам и переустановка СКО;

– выполнить инициализацию сервера хранения артефактов. После настройки параметров сетевого интерфейса и базового DNS-имени необходимо выполнить первичную инициализацию. Процесс инициализации сервера хранения артефактов может занять до 30 минут;

### 4.1.1 Требования к установке k8s-ctl

Для запуска процесса установки и дальнейшего управления платформой используется отдельная ВМ в СУВ или дополнительный физический сервер с установленной ОС AstraLinux SE не ниже 1.7.5 со следующими характеристиками, указанными в таблице 4.

Таблица 4 – Характеристика узла управления платформой

| **Наименование узла** | **Количество ВМ** | **vCPU** | **RAM, GB** | **Диск, GB** | **IOPS** |
|---|---|---|---|---|---|
| Узел управления платформой | 1 (либо локальная машина) | 2 | 4 | 8 | 300+ |

### 4.1.2 Конфигурация кластера (без дополнительных модулей)

Минимальная конфигурация кластера указана в таблице 5.

Таблица 5 – Минимальная конфигурация кластера

| **Наименование узла** | **Количество ВМ** | **vCPU** | **RAM, GB** | **Диск, GB** | **IOPS** |
|---|---|---|---|---|---|
| Master | 1 | 4 | 8 | 32 | 300+ |
| Infra | 1 | 8 | 16 | 128 | 1000+ |
| Worker | 1 | 2 | 4 | 32 | 300+ |
| Итого | 3 | 14 | 28 | 192 | |

Рекомендуемая конфигурация кластера (без дополнительных модулей) указана в таблице 6.

Таблица 6 – Рекомендуемая конфигурация кластера без дополнительных модулей

| **Наименование узла** | **Количество ВМ** | **vCPU** | **RAM, GB** | **Диск, GB** | **IOPS** |
|---|---|---|---|---|---|
| Master | 3 | 4 | 8 | 32 | 300+ |
| Infra | 3 | 6 | 12 | 128 | 1000+ |
| Ingress | 2 и более | 2 | 4 | 32 | 300+ |
| Worker | 2 и более | 2 | 4 | 32 | 300+ |
| Итого | 10 и более | 38 | 76 | 608 | |

Узлы Master, Infra, Ingress устанавливаются в СУВ в ВМ с установленной ОС AstraLinux SE 1.7.5.

Worker устанавливается на сервере контейнеризации с установленной ОС AstraLinux SE 1.7.5. Допускается устанавливать Worker в СУВ в виде отдельной ВМ.

Для корректной установки платформы на подготовленной ВМ или физическом сервере необходимо наличие пакета curl.

СКО чувствительна к производительности диска, поэтому рекомендуется использовать более быстрое хранилище, особенно для хранилища etcd на мастер-узлах.

В стандартной установке ОС при настройке разделов и точек монтирования не используйте отдельное пространство для файлов подкачки (SWAP), поскольку использование SWAP-пространства не поддерживается.

### 4.1.3 Требования для установки дополнительных модулей

Для использования модуля OpenSearch в СКО необходимо добавить следующее количество ресурсов, указанное в таблице 7.

Таблица 7 – Количество ресурсов для использования OpenSearch

| Наименование узла | vCPU | RAM, GB |
|---|---|---|
| Master | 1 | 1 |
| Infra | 3 | 4 |
| Ingress | 1 | 1 |
| Worker | 1 | 1 |
| Итого | 6 | 7 |

Для использования модуля Velero в СКО необходимо добавить следующее количество ресурсов, указанное в таблице 8.

Таблица 8 – Количество ресурсов для использования Velero

| Наименование узла | vCPU | RAM, GB |
|---|---|---|
| Master | 1 | 1 |
| Infra | 1 | 1 |
| Ingress | 1 | 1 |
| Worker | 1 | 1 |
| Итого | 4 | 4 |

### 4.1.4 Подготовка сетевого окружения

4.1.4.1 DNS записи для встроенных сервисов

Подготовка DNS-записей должна быть выполнена до установки СКО. Во избежание задержек при развертывании кластера, необходимо заранее зарезервировать и настроить DNS-имена, указывающие на infra-узлы кластера. Эти DNS-записи будут использоваться встроенными компонентами платформы.

Рекомендуемые DNS-записи для базовой установки:

```
k8s-cilium-hubble.k8s.mycompany.local
k8s-release-git-main.k8s.mycompany.local
k8s-console.k8s.mycompany.local
k8s-oauth.k8s.mycompany.local
k8s-alertmanager-main.k8s.mycompany.local
k8s-grafana-main.k8s.mycompany.local
k8s-prometheus-main.k8s.mycompany.local
```

Дополнительные записи при использовании модулей OpenSearch и NeuVector:

```
k8s-logs-main.k8s.mycompany.local
```

Не забудьте заменить k8s.mycompany.local на dnsBaseDomain, указанный в манифесте k8s-deployment-conf.yaml.

4.1.4.2 Сетевое взаимодействие

Приложение поддерживает использование IP-адреса, настроенного как с помощью DHCP, так и заданного статически. Рекомендуется размещать сервер управления в отдельной от кластеров Kubernetes сети.

Если вы используете DHCP-сервер для настройки сетевого интерфейса сервера, необходимо настроить его на предоставление постоянного IP-адреса и сведений о DNS-серверах.

На текущий момент не поддерживает IPv6.

Для установки СКО требуется внутренний DNS-сервер, при этом создание записей в нем является обязательным условием. При использовании внешнего DNS-сервера, например 8.8.8.8, невозможно установить платформу

Правила доступа из сетей узлов Kubernetes приведены в таблице 9.

Таблица 9 – Правила доступа к сети

| **Ресурс** | **DNS-имя** | **Порт** | **IP-адрес** |
|---|---|---|---|
| Хранилище образов | hub.*DNS-имя* | https/443 | IP-адрес |
| Сервис доставки ПО | hub.*DNS-имя* | https/443 | IP-адрес |
| Сервис настройки ПО | sun.*DNS-имя* | https/443 | IP-адрес |

| Ресурс | DNS-имя | Порт | IP-адрес |
|---|---|---|---|
| Репозиторий пакетов | repo.*DNS-имя* | https/443 | IP-адрес |
| Сервис загрузки обновлений | uploads.*DNS-имя* | https/443 | IP-адрес |

4.1.4.3 Требования к межсетевому экранированию

Для корректной установки и функционирования СКО убедитесь, что в пределах сетевого сегмента (сегментов), в котором располагаются узлы платформы, перечень сетевых правил, либо ограничения по сетевому взаимодействию узлов отсутствуют.

4.1.4.3.1 Узел k8s-ctl для управления платформой

Таблица 10 – Узел k8s-ctl

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Узел k8s-ctl | Мастер-узлы | Входящий | 6443/tcp | Kubernetes API |
| Узел k8s-ctl | Мастер-узлы | Входящий | 8200/tcp | StarVault API |
| Узел k8s-ctl | Все узлы | Входящий | 22/tcp | SSH |
| Узел k8s-ctl | Инфраструктурные узлы | Входящий | 80/tcp | Ingress Nginx Controller HTTP |
| Узел k8s-ctl | Инфраструктурные узлы | Входящий | 443/tcp | Ingress Nginx Controller HTTPS |

4.1.4.3.2 Мастер-узлы кластера Kubernetes

Таблица 11 – Мастер-узлы кластера Kubernetes

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Все узлы, АРМ пользователей платформы | Мастер-узлы | Входящий | 6443/tcp | Kubernetes API |
| Все узлы | Мастер-узлы | Входящий | 2379/tcp | Etcd Client Requests |

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Инфраструктурные узлы | Мастер-узлы | Входящий | 10259/TCP | Kubernetes Scheduler metrics |
| Инфраструктурные узлы | Мастер-узлы | Входящий | 10257/TCP | Kubernetes Controller Manager metrics |
| Мастер-узлы | Мастер-узлы | Двунаправленный | 2380/tcp | Etcd Peer Requests |
| Мастер-узлы | Все узлы | Исходящий | 10250/TCP | Kubelet |

4.1.4.3.3 Инфраструктурные узлы кластера Kubernetes

Таблица 12 – Инфраструктурные узлы кластера Kubernetes

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Все узлы, АРМ пользователей платформы | Инфраструктурные узлы | Входящий | 80/tcp | Ingress Nginx Controller HTTP |
| Все узлы, АРМ пользователей платформы | Инфраструктурные узлы | Входящий | 443/tcp | Ingress Nginx Controller HTTPS |
| Все узлы | Инфраструктурные узлы | Входящий | 8443/tcp | Ingress Nginx Controller Validating webhook |
| Инфраструктурные узлы | Все узлы | Исходящий | 9100/tcp | Prometheus Node Exporter |

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Инфраструктурные узлы | Все узлы | Исходящий | 9962/tcp | Cilium Agent metrics |
| Инфраструктурные узлы | Все узлы | Исходящий | 9963/tcp | Cilium Operator metrics |
| Инфраструктурные узлы | Мастер-узлы | Исходящий | 10259/TCP | Kubernetes Scheduler metrics |
| Инфраструктурные узлы | Мастер-узлы | Исходящий | 10257/TCP | Kubernetes Controller Manager metrics |
| Инфраструктурные узлы | Все узлы | Исходящий | 10250/TCP | Kubelet |
| Инфраструктурные узлы | Инфраструктурные узлы | Исходящий | 10249/tcp | Kube Proxy metrics |

4.1.4.3.4 Узлы балансировки входящих запросов кластера Kubernetes (Ingress-узлы)

Таблица 13 – Узлы балансировки входящих запросов кластера Kubernetes

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Все узлы, АРМ пользователей платформы | Ingress-узлы | Входящий | 80/tcp | Ingress Nginx Controller HTTP |
| Все узлы, АРМ пользователей платформы | Ingress-узлы | Входящий | 443/tcp | Ingress Nginx Controller HTTPS |

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Все узлы | Ingress-узлы | Входящий | 8443/tcp | Ingress Nginx Controller Validating webhook |
| Инфраструктурные узлы | Ingress-узлы | Входящий | 10254/TCP | Ingress Nginx Controller metrics |

Если в конфигурации кластера СКО не используются выделенные Ingress-узлы, то все правила для Ingress-узлов необходимо применить к рабочим узлам (Worker).

4.1.4.3.5 Все узлы кластера Kubernetes

Таблица 14 – Все узлы кластера Kubernetes

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Все узлы | Все узлы | Двунаправленный | 179/tcp | BGP |
| Все узлы | Все узлы | Двунаправленный | 4789/udp | VXLAN Overlay |
| Все узлы | Все узлы | Двунаправленный | 8472/udp | VXLAN Overlay |
| Все узлы | Все узлы | Двунаправленный | IPIP (4) | IP in IP Protocol |
| Все узлы | Все узлы | Двунаправленный | 4240/tcp | Cilium Health Check |
| Все узлы | Все узлы | Двунаправленный | ICMP (8/0) | Cilium Health Check |
| Все узлы | Все узлы | Двунаправленный | 4244/tcp | Cilium Hubble server |
| Все узлы | Все узлы | Двунаправленный | 4245/tcp | Cilium Hubble relay |

| Источник | Адресат | Тип трафика | Порт/Протокол | Описание |
|---|---|---|---|---|
| Инфраструктурные узлы | Все узлы | Входящий | 9100/tcp | Prometheus Node Exporter |
| Инфраструктурные узлы | Все узлы | Входящий | 9962/tcp | Cilium Agent metrics |
| Инфраструктурные узлы | Все узлы | Входящий | 9963/tcp | Cilium Operator metrics |
| Инфраструктурные узлы | Все узлы | Входящий | 10249/tcp | Kube Proxy metrics |
| Инфраструктурные узлы и мастер-узлы | Все узлы | Входящий | 10250/tcp | Kubelet |
| Все узлы | Мастер-узлы | Исходящий | 2379/tcp | Etcd Client Requests |
| Все узлы | Инфраструктурные узлы | Исходящий | 80/tcp | Ingress Nginx Controller HTTP (Internal) |
| Все узлы | Инфраструктурные узлы | Исходящий | 443/tcp | Ingress Nginx Controller HTTPS (Internal) |
| Все узлы | Инфраструктурные узлы, Ingress-узлы | Исходящий | 8443/tcp | Ingress Nginx Controller Validating webhook |
| Все узлы | Мастер-узлы | Исходящий | 6443/tcp | Kubernetes API |

Если установку система оркестрации и контейнеризации планируется выполнять с использованием HTTP-прокси, необходимо добавить в список разрешающих правил доступ (исходящий трафик) к HTTP-прокси со всех узлов платформы.

### 4.1.5 Подготовка узла k8s-ctl для управления СКО

4.1.5.1 Требования по установке

Для установки платформы потребуется отдельный узел (например, локальная машина), которому необходимо минимальное количество ресурсов: 2 CPU и 4 RAM.

Для установки также необходимо установить приложение для запуска контейнеров, например, Docker или Podman.

4.1.5.2 Установка k8s-ctl

Предварительные условия для k8s-ctl:

– установлен и настроен сервера хранения артефактов;

– имеется корневой сертификат сервера хранения артефактов.

Подготовьте локальное хранилище на узле установки платформы для файлов конфигурации, например k8s-install-dir:

```
mkdir ~/k8s-install-dir
cd ~/ k8s-install-dir
```

Скопируйте файл с ключом для доступа на узлы платформы в текущую директорию, затем настройте права доступа на файл:

```
chmod 600 <имя_файла_с_ключом>
```

Запустите контейнер с k8s-ctl.

```
docker run --rm -it -v $PWD:/opt/k8s hub..mycompany.local/k8s /k8s-ctl:v6.0.1
```

Получите шаблон конфигурационного манифеста с помощью k8s-ctl. Для этого выполните команду:

```
k8s-ctl init
```

В результате инициализации в директории k8s-configs появятся следующие файлы:

– k8s-deployment-conf.yaml: основной конфигурационный манифест платформы;

– proxy.yaml: конфигурационный манифест для установки платформы через HTTP-прокси.

Заполните манифест для установки и запустите процесс установки кластера:

```
k8s-ctl bootstrap --ssh-user <имя_пользователя> --ssh-key <закрытый ключ SSH>
```

Команда завершится успешно, когда все компоненты СКО будут установлены и доступны. После успешного завершения установки кластера k8s-ctl сохраняет в рабочей директории файлы с информацией, необходимой для доступа к кластеру и его компонентам:

– kubeadmin.conf - конфигурационный файл с учетной записью администратора кластера для утилиты kubectl, необходимой для работы с Kubernetes.

Учетная запись имеет роль cluster-admin в Kubernetes;

– oauth-credentials.txt - учетная запись администратора кластера для авторизации по протоколу OAuth;

– secrets-store-tokens.txt - токены для доступа к системе управления секретами StarVault;

– k8s-root-ca.pem - корневой TLS-сертификат Kubernetes. Данным сертификатом подписаны все последующие подчиненные центры сертификации.

Не забудьте сохранить учетные данные в надежном месте и удалить их с локальной машины.

# 5 Инфраструктурный мониторинг

## 5.1 Требования к инфраструктуре

### 5.1.1 Требования к вычислительным ресурсам

Инсталляция мониторинга включает в себя 6 компонентов, расположенных на 5 серверах. Перед началом инсталляции необходимо создать 5 серверов. Рекомендуемые имена и параметры серверов для установки компонентов указаны в таблице 15.

Таблица 15 – Рекомендуемые имена и параметры серверов для установки компонентов

| **Сервер** | **Назначение** | **Компоненты** | **vCPU** | **RAM Gb** | **Storage Gb** | **IOPS (не менее)** | **OS** |
|---|---|---|---|---|---|---|---|
| mon-prom | Метрики и алерты | Prometheus, AlertManager | 4 | 8 | 100 | 500 | Ubuntu 24.02 или выше / astra 1.7.5 или выше |
| mon-tsdb | Хранилище метрик | VictoriaMetrics | 8 | 8 | 3 | 1500 | Ubuntu 24.02 или выше / astra 1.7.5 или выше |

| Сервер | Назначение | Компоненты | vCPU | RAM Gb | Storage Gb | IOPS (не менее) | OS |
|---|---|---|---|---|---|---|---|
| mon-logagg | Обработка логов | Vector Agent | 4 | 4 | 30 | 500 | Ubuntu 24.02 или выше / astra 1.7.5 или выше |
| mon-log-engine | Хранение логов | Loki | 4 | 4 | 1000 | 2000 | Ubuntu 24.02 или выше / astra 1.7.5 или выше |
| mon-visual | Визуализация и интерфейсы | Grafana, Karma | 6 | 4 | 50 | 300 | Ubuntu 24.02 или выше / astra 1.7.5 или выше |
| Ansibe host | Установка компонентов | ansible | 6 | 6 | 40 | 300 | Ubuntu 24.02 или выше / astra 1.7.5 или выше |

Для предотвращения переполнения системного раздела и повышения отказоустойчивости рекомендуется выделить отдельные диски/разделы под хранение данных компонентов:

- mon-tsdb (VictoriaMetrics):
  - Точка монтирования: /var/lib/victoria-metrics-data.
  - Рекомендуемый размер (см. в таблице выше).
- mon-log-engine (Loki):
  - Точка монтирования: /var/lib/loki.
  - Рекомендуемый размер (см. в таблице выше).

При подготовке серверов необходимо создать и смонтировать выделенные диски в соответствующие каталоги, добавить их в /etc/fstab для автоматического монтирования при перезагрузке.

Разверните в СУВ 5 ВМ, и если у вас не имеется узла для ансибле, дополнительную вм для управления развертыванием.

### 5.1.2 Установка и настройка модуля мониторинга

Необходимо подготовить сервер, с которого будет осуществляться запуск Ansible. Установите Ansible с помощью команды:

```
sudo apt update
sudo apt install -y ansible
```

Проверьте наличие необходимых утилит и установите их при необходимости:

```
sudo apt install -y sshpass git curl wget python3-venv
```

Сгенерируйте ключ от имени пользователя, от которого будет запускаться Ansible (например, ubuntu, alexandra и т.п.). Этот ключ будет использоваться для подключения по SSH к целевым ВМ от имени пользователя ansible:

```
ssh-keygen -t rsa -b 4096
```

После этого склонируйте репозиторий с Ansible-плейбуками на машину.

Далее выполните следующие системные настройки на каждой ВМ из стенда мониторинга. Проверьте наличие Python 3 на целевых хостах. Если его нет, установите:

```
sudo apt install -y python3 python3-apt python3-pip
```

Создайте пользователя для Ansible:

```
sudo useradd -m -s /bin/bash ansible
sudo passwd ansible
sudo usermod -aG sudo ansible
```

После этого, перейдя на машину с Ansible, следует скопировать публичный ключ:

```
ssh-copy-id ansible@< ip>
```

Необходимо проверить подключение к созданным машинам:

```
ansible all -m ping
```

Если подключение прошло успешно, можно переходить к выполнению плейбука. Для этого требуется перейти в директорию, куда был склонирован репозиторий с Ansible-плейбуком. Структура репозитория содержит:

- collections/ansible_collections/kagecore
- inventories
- kagecore/common
- playbooks
- Dockerfile
- README.md
- ansible.cfg
- requirements.txt

Далее необходимо перейти в папку inventories, затем в папку run. В папке run будет следующая структура:

- group_vars
- grafana_dashboards
- scrape_configs
- hosts.ini

В первую очередь необходимо изменить IP-адреса и доменные имена серверов, используемых в стенде мониторинга. Далее, по аналогии с примером, имеющимся в репозитории, следует занести имена в группы, соответствующие устанавливаемому ПО. Например:

```
[prometheus]
test-mon-prom
```

После этого следует выйти из данного файла и перейти в папку group_vars. (Предварительно в Nexus-репозиторий должны быть перенесены образы устанавливаемых компонентов.) Далее следует перейти в файл all.yml в папке group_vars. Здесь необходимо изменить пути к каждому из компонентов. (Если изменялось только доменное имя Nexus, то следует изменить только его.) Например:

```
nexus_url_global: "http://nexus.KageCore.tech:8081/repository/KageCore-files"
image_paths:
node_exporter_download_url: "{{ nexus_url_global }}/monitoring/node_exporter-1.8.2.linux-amd64.tar.gz"
```

Поменять на:

```
nexus_url_global: "http://nexus.goodcompany.tech:8081/repository/goodcompany-files"
image_paths:
```

```
node_exporter_download_url: "{{ nexus_url_global }}/monitoring/node_exporter-1.8.2.linux-amd64.tar.gz"
```

Также следует изменить учетные данные для входа в Nexus:

– nexus_username: "admin"
– nexus_password: "password"

После внесения всех корректировок можно запускать плейбук. При запуске плейбуков для установки мониторинга требуется передавать параметр --ask-vault-pass, так как чувствительные параметры в инвентаре шифруются с помощью Ansible Vault и хранятся в соответствующей переменной. Команда для шифрования чувствительных данных:

```
ansible-vault encrypt_string 'very_secure_var' --name 'var'
```

И тогда сама команда запуска:

```
ansible-playbook -i inventories/test/hosts.ini playbooks/monitoring_simple.yml --ask-vault-pass
```

При запуске данной команды начинается выполнение плейбука, который перенаправляет на плейбук, находящийся в collections/ansible_collections/KageCore/monitoring/playbooks. Внутри него находится запуск ролей. Каждая роль отвечает за свое ПО стенда мониторинга.

После завершения работы плейбука необходимо проверить доступность каждого из установленных сервисов:

– Prometheus: http://your-prometheus-host:9090.
– Grafana: http://your-grafana-host:3000.
– Alertmanager: http://your-alertmanager-host:9093.
– Loki: http://your-loki-host:3100.
– Karma: http://your-karma-host:8080.

Следует убедиться, что все службы работают и доступны. В случае необходимости, большинство параметров каждой из ролей можно настроить. Роли запускаются в очередности:

– Node exporter.
– Loki.
– VictoriaMetrics.
– Prometheus.
– Alertmanager.
– Vector aggregator.
– Grafana.
– Vector.
– Karma.

Ниже представлено описание каждой роли по отдельности.

5.1.2.1 node_exporter

Эта Ansible-роль предназначена для установки и настройки Node Exporter — утилиты, собирающей метрики системы для **Prometheus**.

Структура файлов роли:

- **defaults/main.yml**:
- **handlers/main.yml**:
- **tasks/**
  - **main.yml**:
  - **confprometheus.yml**
- **templates/etc/default**
  - **node_exporter.j2**
  - **node_exporter_auth.yml.j2**
  - **node_exporter_scrape.j2**
- **templates/etc/systemd/system**
  - **node_exporter.service.j2**

Переменные:

В файле defaults/main.yml задаются следующие переменные:

- **node_exporter_version**: Версия Node Exporter (по умолчанию 1.8.2).
- **node_exporter_arch**: Архитектура системы (по умолчанию определяется автоматически на основе архитектуры машины).
- **node_exporter_download_url**: URL для скачивания архивов с Node Exporter.
- **node_exporter_group_name**: Имя группы для пользователя Node Exporter (по умолчанию "node_exporter").
- **node_exporter_user_name**: Имя пользователя для Node Exporter (по умолчанию "node_exporter").
- **node_exporter_dir**: Директория установки Node Exporter (по умолчанию "/opt/node_exporter").
- **node_exporter_log_dir**: Директория для логов Node Exporter (по умолчанию "/var/log/node_exporter").
- **node_exporter_binary_dir**: Директория для бинарных файлов Node Exporter (по умолчанию "/usr/local/bin").
- **node_exporter_web_listen_address**: Адрес, на котором Node Exporter будет слушать запросы (по умолчанию "0.0.0.0").
- **node_exporter_web_listen_port**: Порт для Node Exporter (по умолчанию "9100").

– **node_exporter_web_telemetry_path**: Путь к меткам для экспорта данных (по умолчанию "/metrics").

– **node_exporter_log_level**: Уровень логирования (по умолчанию "info").

– **node_exporter_log_format**: Формат логов (по умолчанию "json").

– **node_exporter_log_to_file**: определяет, записывать ли логи в файл (по умолчанию false).

– **node_exporter_enabled_collectors**: Список включенных коллекторов (по умолчанию "systemd", "mountstats", "ethtool").

– **node_exporter_disabled_collectors**: Список отключенных коллекторов (по умолчанию пуст).

– **scrape_configs_path**: Путь для конфигурации сбора данных (по умолчанию "{{ inventory_dir }}/scrape_configs").

– **scrape_config_file**: Имя файла конфигурации сбора данных (по умолчанию "node_exporter.yml").

Задачи:

1) **Создание группы и пользователя для Node Exporter**:

– Проверка и создание группы и пользователя для запуска Node Exporter.

2) **Создание директорий**:

– Создание необходимых директорий для установки и хранения файлов Node Exporter.

3) **Загрузка архива с Node Exporter**:

– Загрузка архива с бинарным файлом Node Exporter с заданного URL.

4) **Распаковка архива**:

– Извлечение содержимого архива в соответствующую директорию.

5) **Создание символьной ссылки на бинарный файл**:

– Создание символьной ссылки для удобства вызова бинарного файла node_exporter из системной директории.

6) **Настройка systemd-сервиса**:

– Конфигурация и активация службы Node Exporter с использованием systemd.

7) **Настройка переменных по умолчанию**:

– Конфигурация файла /etc/default/node_exporter для настройки переменных окружения.

8) **Настройка авторизации**:

– Конфигурация файла для аутентификации Node Exporter.

9) **Включение службы Node Exporter**:

– Активация и перезагрузка systemd-сервиса для обеспечения автоматического старта Node Exporter.

10) **Конфигурация для Prometheus**:

– Включение конфигурации для Prometheus с помощью дополнительного файла конфигурации.

5.1.2.2 alert_manager

Эта Ansible-роль предназначена для **автоматизированной установки и настройки Alertmanager** — компонента мониторинга Prometheus, который отвечает за **обработку, группировку и отправку уведомлений об алертах**.

Структура файлов роли:

– **defaults/main.yml** – Определяет переменные по умолчанию.

– **files/am-templates/telegram.tmpl** – Шаблон для интеграции Alertmanager с Telegram.

– **tasks/install.yml** – Устанавливает и настраивает Alertmanager.

– **tasks/main.yml** – Основной файл задач (вызывает prepare.yml и install.yml).

– **tasks/prepare.yml** – Подготавливает окружение (создает пользователей и директории).

– **templates/alertmanager.service.j2** – Шаблон systemd-юнита для управления Alertmanager.

– **templates/alertmanager.yml.j2** – Шаблон конфигурационного файла Alertmanager.

Переменные роли:

Основные переменные (defaults/main.yml):

– **alert_manager_version** – Версия Alertmanager, которая будет установлена (по умолчанию 0.28.0).

– **alert_manager_arch** – Архитектура системы (определяется автоматически, используется amd64 или arm64).

– **alert_manager_download_url** – URL для скачивания архива с Alertmanager.

– **alert_manager_data_dir** – Директория для хранения данных Alertmanager (/var/lib/alertmanager).

– **alert_manager_log_dir** – Директория для логов Alertmanager (/var/log/alertmanager).

– **alert_manager_conf_dir** – Директория для конфигурационных файлов Alertmanager (/etc/alertmanager).

– **alert_manager_binary_dir** – Директория для бинарных файлов Alertmanager (/usr/local/bin).

– **alert_manager_systemd_dir** – Директория для systemd-юнитов Alertmanager (/etc/systemd/system).

– **alert_manager_templates_dir** – Директория для шаблонов Alertmanager (/etc/alertmanager/templates).

– **alert_manager_group_name** – Имя группы пользователя Alertmanager (alertmanager).

– **alert_manager_user_name** – Имя пользователя для Alertmanager (alertmanager).

– **alert_manager_retention_period** – Время хранения алертов (72h).

– **alert_manager_listen_port** – Порт, на котором Alertmanager будет слушать (9093).

– **alert_manager_log_level** – Уровень логирования (info).

– **alert_manager_log_format** – Формат логов (json).

– **alert_manager_log_to_file** – Определяет, записывать ли логи в файл (false).

– **alert_manager_config** – Основная конфигурация Alertmanager:

  – Включает настройки таймаутов, маршрутизации уведомлений и обработчиков алертов.

Основные задачи роли (tasks/main.yml):

1) **Подготовка пользователя и директорий**:

Создаются необходимые пользователи, группы и директории для Alertmanager.

2) **Установка и настройка Alertmanager**:

– Скачивается и распаковывается архив с бинарными файлами.

– Копируются шаблоны конфигурационных файлов.

– Устанавливается systemd-юнит для автоматического управления Alertmanager.

3) **Запуск и перезапуск службы**:

После установки Alertmanager регистрируется в systemd, включается в автозапуск и перезапускается.

5.1.2.3 victoria_metrics

Эта Ansible-роль предназначена для **автоматизированной установки и настройки VictoriaMetrics** — СУБД для временных рядов, с опциональной поддержкой **vmalert** для управления алертами.

Структура роли:

– **defaults/main.yml** – Определяет переменные по умолчанию.

– **files/rules/vm-health.rules** – Правила для мониторинга состояния VictoriaMetrics.

– **files/rules/vmalert.rules** – Правила для vmalert.

– **handlers/main.yml** – Обработчики событий (например, перезапуск сервиса).

– **tasks/configure.yml** – Настройка VictoriaMetrics.

– **tasks/install.yml** – Установка VictoriaMetrics.

– **tasks/main.yml** – Основной файл задач (вызывает prepare.yml, install.yml, configure.yml и vmalert.yml).

– **tasks/prepare.yml** – Подготовка окружения (создание пользователя и директорий).

– **tasks/vmalert.yml** – Установка и настройка vmalert (если включен).

– **templates/prometheus.yml.j2** – Шаблон конфигурации Prometheus для VictoriaMetrics.

– **templates/victoria-metrics.service.j2** – Шаблон systemd-юнита для VictoriaMetrics.

– **templates/vmalert.service.j2** – Шаблон systemd-юнита для vmalert.

Переменные роли:

Основные переменные (defaults/main.yml):

– **victoria_metrics_version** – Версия VictoriaMetrics, которая будет установлена (по умолчанию 1.110.0).

– **victoria_metrics_arch** – Архитектура системы (amd64 или arm64, определяется автоматически).

– **victoria_metrics_default_download_url** – URL для скачивания архива с VictoriaMetrics.

– **victoria_metrics_vmutils_download_url** – URL для скачивания утилит VictoriaMetrics (vmutils).

– **victoria_metrics_group_name** – Имя группы пользователя VictoriaMetrics (victoriametrics).

– **victoria_metrics_user_name** – Имя пользователя для VictoriaMetrics (victoriametrics).

– **victoria_metrics_data_dir** – Директория для хранения данных VictoriaMetrics (/opt/data/victoria-metrics/lib/).

– **victoria_metrics_logs_dir** – Директория для логов (/opt/data/victoria-metrics/log/).

– **victoria_metrics_conf_dir** – Директория для конфигурационных файлов (/etc/victoria-metrics).

– **victoria_metrics_binary_dir** – Директория для бинарных файлов (/usr/local/bin).

– **victoria_metrics_systemd_location** – Директория для systemd-юнитов (/lib/systemd/system).

– **victoria_metrics_retention_period** – Время хранения метрик (7d).

– **victoria_metrics_config_check_interval** – Интервал проверки конфигурации (30s).

– **victoria_metrics_max_scrape_size** – Максимальный размер запроса сбора (33554432).

– **victoria_metrics_log_level** – Уровень логирования (INFO).

– **victoria_metrics_log_format** – Формат логов (json).

– **victoria_metrics_log_to_file** – Определяет, записывать ли логи в файл (false).

– **victoria_metrics_port** – Порт для API VictoriaMetrics (8428).

– **victoria_metrics_alert.enabled** – Включение модуля vmalert (false).

– **victoria_metrics_alert.config_check_interval** – Интервал проверки правил (30s).

– **victoria_metrics_alert.datasource_address** – Адрес источника данных (http://localhost:8428).

– **victoria_metrics_alert.alertmanager_address** – Адрес Alertmanager (http://localhost:9093).

– **victoria_metrics_alert.remote_write_address** – Адрес удаленной записи (http://localhost:8428).

– **victoria_metrics_alert.remote_read_address** – Адрес удаленного чтения (http://localhost:8428).

– **victoria_metrics_scrape_configs** – Конфигурация сбора метрик:

  – Имя задания (victoria_metrics).

  – Путь к метрикам (/metrics).

  – Статический таргет (localhost:8428/metrics).

Основные задачи роли (tasks/main.yml):

1) **Создание пользователя и директорий**:

Создаются необходимые пользователи, группы и директории для VictoriaMetrics.

2) **Установка VictoriaMetrics**:

Скачивается и распаковывается архив с бинарными файлами VictoriaMetrics и vmutils. Файлы размещаются в нужных директориях.

3) **Настройка VictoriaMetrics**:

Копируются файлы конфигурации. Применяются параметры хранения данных и логирования.

4) **Запуск и перезапуск службы**:

После установки VictoriaMetrics регистрируется в systemd, включается в автозапуск и запускается.

5) **Установка vmalert (при включенной опции)**:

Если в конфигурации включен vmalert, он скачивается, устанавливается и настраивается для работы с Alertmanager.

5.1.2.4 Loki

Эта Ansible-роль предназначена для автоматизированной установки, настройки и удаления системы логирования **Loki**.

Структура файлов:

– **defaults/main.yml** – содержит переменные по умолчанию, такие как версия Loki, пути установки, настройки сервера и параметры хранения логов.

– **handlers/main.yml** – определяет обработчики событий (например, перезапуск сервиса Loki после изменения конфигурации).

– **tasks/main.yml** – основной файл задач, который включает в себя другие файлы с задачами.

– **tasks/deploy.yml** – выполняет установку и настройку Loki.

– **tasks/uninstall.yml** – содержит задачи для удаления Loki.

– **tasks/setup-Astra Linux.yml**, **setup-Debian.yml**, **setup-RedHat.yml** – файлы с задачами для установки Loki на соответствующие операционные системы.

– **templates/config.yml.j2** – шаблон конфигурационного файла Loki.

– **templates/rules.yml.j2** – шаблон файла с правилами оповещений.

– **vars/Astra Linux.yml**, **vars/Debian.yml**, **vars/RedHat.yml** – файлы переменных для различных операционных систем.

Основные переменные**:**

– loki_version – версия Loki, устанавливаемая по умолчанию.

– loki_http_listen_port – порт, на котором Loki будет принимать HTTP-запросы (3100 по умолчанию).

– loki_expose_port – флаг, указывающий, следует ли открывать порт для Loki.

– loki_download_url_rpm, loki_download_url_deb – ссылки на установочные пакеты для RPM и DEB-дистрибутивов.

– loki_working_path – путь, где Loki будет хранить свои данные.

– loki_ruler_alert_path – путь к файлам с правилами оповещений.

Установка и удаление:

– **Установка**: выполняется с помощью tasks/deploy.yml, где загружаются и устанавливаются необходимые пакеты, создаются файлы конфигурации и запускается сервис Loki.

– **Удаление**: осуществляется через tasks/uninstall.yml, который удаляет файлы и сервис Loki.

Детали развертывания (tasks/deploy.yml):

Определение версии Loki:

Если Loki уже установлен, проверяется текущая версия.

Установка в зависимости от ОС.

Выполняется подключение соответствующего файла задач для установки Loki в зависимости от семейства операционной системы:

– setup-RedHat.yml – для RedHat и Rocky;
– setup-Debian.yml – для Debian и Ubuntu;
– setup-Debian.yml – для Astra Linux.

Настройка конфигурации:

1) Проверяется наличие стандартного каталога Loki /tmp/loki/boltdb-shipper-active. Если он существует, выполняется очистка.
2) Создается рабочая директория Loki (loki_working_path).
3) Применяется шаблон конфигурационного файла (config.yml.j2) и проверяется его корректность с помощью loki --verify-config.

Настройка правил оповещений:

1) Если задан путь для правил (loki_ruler_alert_path), он создается.
2) Применяется шаблон файла правил (rules.yml.j2).

Настройка firewall:

– Если firewalld активен, настраиваются правила для открытия или закрытия порта Loki (loki_http_listen_port).

Запуск и проверка сервиса:

1) Обработчики изменений (restart loki) применяются после конфигурации.
2) Запускается сервис Loki (loki.service).
3) Проверяется доступность Loki по HTTP-запросу к /ready.

#### 5.1.2.5 Vector / Vector aggregator

Данная роль запусмкается в двух состояниях - **Vector (agent)** — это легковесный агент, установленный на конечных узлах (серверы, виртуалки, контейнеры), где собираются логи; **Vector aggregator** — это инстанс Vector, который работает **централизовано,** принимает логи от агентов и выполняет дополнительную обработку. В зависимости от ключа переданного роли выбирается вариант установки.

Структура файлов:

– **defaults/main.yml** – содержит переменные по умолчанию для установки Vector.

– **tasks/main.yml** – основной файл задач, включающий подготовку, установку и запуск сервиса.

– **tasks/prepare.yml** – подготавливает пользователя, группы и директории для Vector.

– **tasks/install.yml** – выполняет загрузку, установку и настройку Vector.

– **templates/vector.j2** – шаблон конфигурационного файла Vector.

– **templates/vector.service.j2** – шаблон systemd-сервиса для Vector.

– **templates/vector.yaml.j2** – шаблон основного конфигурационного файла Vector.

Основные переменные (defaults/main.yml):

– vector_version: версия Vector (по умолчанию 0.44.0).

– vector_install_type: тип установки (archive или package).

– vector_download_url: URL для скачивания архива Vector.

– vector_download_url_deb: URL для скачивания DEB-пакета.

– vector_arch: архитектура процессора (x86_64, aarch64, amd64, arm64).

– vector_binary_dir: каталог, в который будет установлен исполняемый файл Vector (/usr/bin).

– vector_config_dir: каталог для хранения конфигурации Vector (/etc/vector).

– vector_systemd_dir: каталог для systemd-юнитов (/etc/systemd/system).

– vector_data_dir: каталог для хранения временных данных Vector (/var/lib/vector).

– vector_env_dir: каталог для хранения переменных окружения (/etc/default).

– vector_group_name: группа пользователя Vector (vector).

– vector_user_name: имя пользователя Vector (vector).

– vector_additional_groups: дополнительные группы (systemd-journal).

– vector_log_format: формат логов (json).

– vector_api: настройки API Vector (по умолчанию отключен, порт 8686).

– vector_sources: источники логов (по умолчанию пусто, пример с dummy_logs).

– vector_transforms: трансформации логов (по умолчанию пусто, пример с parse_logs).

– vector_sinks: получатели логов (по умолчанию пусто, пример с print).

Основные задачи (tasks/main.yml):

1) Подготовка окружения (tasks/prepare.yml):

– Создает пользователя и группу Vector.

– Создает необходимые каталоги (vector_config_dir, vector_data_dir, vector_env_dir).

2) Установка Vector (tasks/install.yml):

– Определяет способ установки (archive или package).

– Загружает и устанавливает соответствующий пакет Vector.

– Копирует шаблоны конфигурации (vector.yaml.j2) в vector_config_dir.

– Настраивает systemd-юнит (vector.service.j2) и запускает сервис Vector.

3) Запуск и перезапуск сервиса:

– Перезапускает systemd и включает сервис Vector.

– Следит за изменениями конфигурации и перезапускает Vector при их наличии.

5.1.2.6 Grafana

Эта Ansible-роль предназначена для **автоматизированной установки, настройки и управления мониторинговой системой Grafana** с поддержкой провиженинга, плагинов, источников данных, дашбордов, API-ключей и уведомлений.

Структура роли:

Роль включает следующие файлы и каталоги:

– **defaults/main.yml** – Определяет значения переменных по умолчанию (версия Grafana, настройки логов, параметры безопасности и т. д.).

– **files/** – Хранит сертификаты SSL (grafana.crt, grafana.key).

– **handlers/main.yml** – Определяет обработчики (restart grafana и другие).

– **meta/main.yml** – Описание роли, зависимости и поддерживаемые платформы.

– **tasks/** – Основные задачи установки и настройки:

  – install.yml – Установка Grafana.

  – configure.yml – Базовая конфигурация.

  – datasources.yml – Настройка источников данных.

  – dashboards.yml – Загрузка дашбордов.

  – api_keys.yml – Управление API-ключами.

  – notifications.yml – Настройка уведомлений.

  – plugins.yml – Установка и управление плагинами.

  – preflight.yml – Проверка окружения перед установкой.

  – setup-Astra Linux.yml, setup-Debian.yml – Специфические настройки для дистрибутивов.

– **templates/** – Jinja2-шаблоны конфигурационных файлов (grafana.ini.j2, ldap.toml.j2).

– **vars/** – Переменные для различных дистрибутивов (astra linux.yml, debian.yml, redhat.yml, suse.yml).

1) **Файл: defaults/main.yml**:

Этот файл содержит переменные по умолчанию для роли Grafana.

**Основные переменные и их назначение:**

– grafana_version: 11.4.0 — версия Grafana, которая будет установлена.

– grafana_arch: "{{ 'arm64' if ansible_facts['architecture'] == 'aarch64' else 'amd64' }}"

– Определяет архитектуру системы (ARM64 или AMD64) и устанавливает соответствующую версию Grafana.

– grafana_source: "package" — способ установки Grafana (из пакета или архива).

– grafana_download_url_deb: "http://nexus.tech:8081/repository/tech-files/monitoring/grafana_{{ grafana_version }}_{{ grafana_arch }}.deb"

– URL для скачивания пакета Grafana в формате .deb.

– grafana_use_provisioning: true — разрешает использование встроенного механизма провиженинга (автонастройки) Grafana.

– grafana_provisioning_synced: false — если true, удаляет старые объекты при обновлении провиженинга.

– grafana_cap_net_bind_service: true

– Позволяет Grafana работать на портах ниже 1024 без повышения привилегий.

– grafana_log_level: info — уровень логирования (debug, info, warn, error).

– grafana_log_format: json — формат логов (текст или JSON).

**Конфигурация grafana.ini (инициализация сервера):**

Переменная grafana_ini_default содержит параметры конфигурации Grafana:

– instance_name — имя экземпляра Grafana (используется FQDN или имя хоста).

– paths.logs, paths.data — пути для хранения логов и данных.

– server.http_addr, server.http_port — адрес и порт веб-сервера Grafana.

– server.protocol: https — использование HTTPS.

– server.cert_key, server.cert_file — пути до SSL-сертификатов.

– security.admin_user: admin — имя администратора Grafana.

– users.allow_sign_up: false — отключение регистрации новых пользователей.

– auth.anonymous.enabled: true — включение анонимного входа.

2) **Файл: tasks/configure.yml**:

Этот файл содержит задачи для настройки Grafana после установки.

**Основные задачи:**

**Создание директорий для конфигурации и данных:**

– Устанавливает права доступа и владельца директорий /etc/grafana, /var/lib/grafana, /var/log/grafana.

**Копирование конфигурационных файлов**:

– Использует шаблоны Jinja (grafana.ini.j2, ldap.toml.j2, tmpfiles.j2) и копирует их в /etc/grafana/.

– Устанавливает владельца файлов (grafana:grafana) и права доступа.

**Перезапуск службы Grafana после изменений в конфигурации**:

– После обновления конфигурации вызывает хэндлер Restart Grafana, который перезапускает сервис.

3) **Файл: tasks/main.yml:**

Этот файл является основным и вызывает другие задачи для установки и настройки Grafana.

**Основные задачи:**

**Включает задачи установки:**

– include_tasks: install.yml — установка Grafana.

**Включает задачи предварительной проверки:**

– include_tasks: preflight.yml — проверка системы перед установкой.

**Конфигурация сервиса:**

– include_tasks: configure.yml — настройка Grafana.

**Добавление источников данных:**

– include_tasks: datasources.yml — настройка подключения к базам данных.

**Настройка дашбордов:**

– include_tasks: dashboards.yml — загрузка и настройка дашбордов.

**Настройка плагинов:**

– include_tasks: plugins.yml — установка необходимых плагинов.

**Настройка уведомлений:**

– include_tasks: notifications.yml — настройка уведомлений в Grafana.

**Создание API-ключей:**

– include_tasks: api_keys.yml — генерация API-ключей для доступа к Grafana.

**Финальная задача:**

– Перезапуск сервиса Grafana, если были внесены изменения в конфигурацию.

**Дополнительная информация:**

При необходимости изменить сертификаты, заменяем:

– **files/** – (grafana.crt, grafana.key) на нужные.

5.1.2.7 Karma

Роль предназначена для **автоматизированной установки и настройки веб-интерфейса Karma**, используемого для отображения и управления алертами из Alertmanager.

Структура роли:

Роль включает следующие файлы и каталоги:

– defaults/main.yml – Определяет значения переменных по умолчанию (версия Karma, настройки логов, адрес и порт для подключения и другие параметры).

– handlers/main.yml – Определяет обработчики для перезапуска и управления сервисом Karma.

– tasks/ – Основные задачи установки и настройки:

– install.yml – Скачивание и установка бинарных файлов Karma, настройка конфигурации и сервисов.

– main.yml – Основные задачи, выполняющиеся при запуске роли, такие как настройка конфигурации и запуск сервисов.

– prepare.yml – Подготовка системы для установки Karma, создание нужных директорий и прав доступа.

– templates/ – Jinja2-шаблоны конфигурационных файлов (karma.yaml.j2 для конфигурации и karma.service.j2 для systemd-сервиса).

1) Файл **defaults/main.yml**:

Этот файл содержит переменные по умолчанию, которые будут использоваться в других частях роли.

Содержание:

– **karma_version**: Версия Karma, которая будет установлена. Установлена версия по умолчанию "0.120".

– **karma_arch**: Архитектура, на которой будет работать Karma. Используется проверка архитектуры машины, чтобы выбрать arm64 для архитектуры aarch64 или amd64 для других архитектур.

– **karma_download_url**: URL для загрузки архива с бинарным файлом Karma. URL зависит от архитектуры.

– **karma_data_dir**: Каталог, в который будут загружены файлы Karma.

– **karma_log_dir**: Каталог, в который будут записываться логи Karma.

– **karma_conf_dir**: Каталог, в котором будет храниться конфигурационный файл Karma.

– **karma_binary_dir**: Каталог, в который будет скопирован бинарный файл Karma.

– **karma_systemd_dir**: Каталог для системных сервисов, в который будет

скопирован файл сервиса для systemd.

– **karma_group_name** и **karma_user_name**: Группа и пользователь, от имени которых будет работать сервис Karma.

– **karma_bind_address**: Адрес, на который Karma будет привязан. По умолчанию это 0.0.0.0.

– **karma_port**: Порт, на котором будет работать Karma. По умолчанию используется порт 8080.

– **karma_prefix**: Префикс для пути в URL, с которым будет работать Karma.

– **karma_log_level**: Уровень логирования для Karma (по умолчанию info).

– **karma_log_format**: Формат логирования (по умолчанию json).

– **karma_log_to_file**: Указывает, будет ли логирование выводиться в файл (по умолчанию false).

– **karma_config**: Конфигурация для Karma, включая параметры для подключения к Alertmanager и настройки интервала.

2) Файл **tasks/install.yml**:

Этот файл содержит задачи для установки и настройки Karma на удаленном хосте.

Содержание:

1) **Загрузка архива с Karma**:

– Используется модуль get_url для скачивания архива с бинарным файлом Karma с указанного URL.

– Архив сохраняется в директории для данных, /var/lib/karma/archive/.

2) **Извлечение архива**:

– Модуль unarchive используется для распаковки скачанного архива в каталог /var/lib/karma/source/.

– Указана группа и владелец, которые будут назначены извлеченным файлам.

3) **Копирование бинарного файла**:

– Модуль copy используется для копирования бинарного файла Karma в каталог /usr/local/bin/ для его использования.

– Установлены права доступа, чтобы файл мог быть выполнен.

4) **Копирование конфигурационного файла**:

– Модуль template используется для копирования и рендеринга конфигурационного файла karma.yaml из шаблона karma.yaml.j2.

5) **Копирование файла сервиса systemd**:

– Модуль template также используется для создания и копирования файла сервиса systemd karma.service из шаблона karma.service.j2.

6) **Запуск Karma**:

– Модуль systemd используется для перезагрузки демона и запуска сервиса Karma. Также сервис включается для автоматического запуска при старте системы.

7) **Перезапуск Karma**:

– Если в процессе установки были изменены бинарные файлы, конфигурация или сервис, роль выполняет перезапуск Karma.

# 6 Установка подсистемы управления вычислительными ресурсами, совместно с модулями[2]

## 6.1 Требования к вычислительным ресурсам.

Manager-хост – хост, с которого производится установка СУВР.

Target-хост – хост на котором будет запущен портал управления СУВР.

В СУВ подготовить ВМ для manager и target с характеристиками, указанными в таблицах 16 и 17, и установленной ОС AstraLinux SE 1.8.2

### 6.1.1 Минимальные системные требования

Требования к Single instance указаны в таблице 16.

Таблица 16 – Требования к Single instance

| Тип хоста | CPU | RAM | Диск SSD (GB) |
|---|---|---|---|
| Manager | 4 | 8 | 64 |
| Target | 16 | 72 | 600 |

Требования к High Availability (HA) указаны в таблице 17.

Таблица 17 – Требования к High Availability

| Тип хоста | CPU | RAM | Диск SSD (GB) |
|---|---|---|---|
| Manager | 4 | 8 | 64 |
| Server (x3) | 2 | 4 | 64 |
| Agent (x3) | 16 | 72 | 600 |

[2] **Требования к исходным данным и процессу установки идентичны**

| Тип хоста | CPU | RAM | Диск SSD (GB) |
|---|---|---|---|
| LB(x2) | 2 | 4 | 32 |

### 6.1.2 Настройка DNS

В DNS необходимо выделить зону, в рамках которой будут размещаться сервисы портала, например: example.app

Далее можно либо завести wildcard запись, которая будет указывать на target-хост, куда устанавливается СУВР (например *.example.app и *.nexus.example.app), либо завести каждую запись отдельно.

Для Single instance все записи должны указывать на IP-адрес target-сервера.

Для High Availability все записи должны указывать на IP-адрес VIP (указывается в файле inventories/app/hosts, см. документацию ниже).

### 6.1.3 Требования для Manager

На BM manager дополнительно должны быть установлены следующие пакеты:

- Docker.
- Образ deploy-manager.

### 6.1.4 Требования для Target

На BM target дополнительно должны быть установлены следующие пакеты:

- Python не ниже 3.7.
- Доступ к репозиториям ОС.
- Необходимо включить IPv6 на Target-хосте Требования по Firewall, SELinux.

Для работы в режиме High Availability на target-хостах рекомендуется отключить Selinux и Firewall. Эта рекомендация распространяется на работу в режиме Single Instance.

### 6.1.5 Требования к портам

Для работы в режиме High Availability должна быть следующая сетевая доступность, указанная в таблице 18.

Таблица 18 – Требования к портам

| Протокол | Порт | Source | Destination | Описание |
|---|---|---|---|---|
| TCP | 6443 | Все ноды | Все ноды | k3s |
| TCP | 2379-2380 | Server | Server | etcd |
| TCP | 10250 | Все ноды | Все ноды | kubelet metrics |
| UDP | 8472 | Все ноды | Все ноды | flannel |
| TCP | 80,443 | | LB | haproxy |

### 6.1.6 Установка образа manager и настройка config

Команды выполняются на manager-хосте.

1) Скачайте дистрибутив в архиве с расширением `tar.gz` и распакуйте его.

– В случае установки с помощью manage-tool пропустите этот шаг, иначе установите образ deploy-manager, который находится в распакованном архиве. Пример установки версии 1.5.0:

```
docker load < deploy-manager-1.5.0.tar
```

2) Скопируйте пример inventory:

```
cp -R inventories/sample inventories/app
```

3) Измените данные подключения к порталу и в файле hosts, расположенном в каталоге inventories/app/. Описание настроек находится в комментариях к этому файлу.

4) Создайте SSH-ключ и скопируйте его на target-хост(ы):

– Создайте ключ, если он отсутствует:

```
ssh-keygen
```

5) Скопируйте SSH-ключ на target-хост:

```
ssh-copy-id [id пользователя]
```

6) Заполните файл hosts:

Файл должен находиться по пути inventories/apps/hosts.

### 6.1.7 Настройки файла config.yml

В файле config.yml заполнить поля для подключения СУВ. Для этого нужно раскомментировать строки платформы виртуализации и указать параметры подключения. Заполнить файл config.yml можно как вручную, так и через пользовательский интерфейс, запустив команду:

```
manage-tool --ui
```

Дополнительно поддерживаются следующие типы платформ: zVirt, vSphere, Openstack, vCloud, Hyper V, Yandex Cloud. Название этих платформ нужно указать в параметре type.

### 6.1.8 Запуск развертывания

1) Если используются параметры по умолчанию, выполните команду:

```
./PATH/manage-tool --install
```

2) Если используются другие пути до инвентаря и ssh-ключа, выполните команду:

```
./PATH/manage-tool.sh --inventory inventories/myinventory/hosts --ssh-private-key ~/.ssh/mykeys/dev –install
```

Подробно ознакомиться с утилитой `manage-tool.sh` можно с помощью команды `manage-tool --help`.