Cinimex Microservice Platform

Руководство по установке

Версия платформы: v1.0.0

Москва 2019г.

Оглавление

1 Глоссарий ... 3

2 Введение ... 3

2.1 Цели документа ... 3

2.2 Описание ПО ... 3

2.3 Целевая аудитория и уровень подготовки ... 4

2.4 Связанные документы ... 4

3 Системные требования ... 5

3.1 Серверы ... 5

3.2 Рабочая станция администратора ... 6

3.3 Сетевые настройки ... 6

4 Описание инсталлятора ... 7

4.1 Порядок загрузки дистрибутива ... 7

4.2 Директории ... 7

4.3 Файлы ... 7

4.4 Файлы ответов Ansible ... 7

5 Установка ... 8

5.1 Подготовка рабочей станции администратора ... 8

5.2 Подготовка файлов ответов ... 9

5.2.1 Настройка списка серверов ... 9

5.2.2 Настройка доступа к серверам ... 10

5.2.3 Общие настройки для платформы ... 10

5.2.4 Настройки компонент платформы ... 10

5.2.5 Индивидуальные настройки серверов ... 10

5.3 Подготовка серверов ... 11

5.4 Запуск установки ... 12

6 Проверка установки ... 12

1 Глоссарий

ПО	Программное обеспечение
ОС	Операционная система
АС	Автоматизированная система
CI&CD	Continues Integration, Continues Delivery and Continues Deployment - процессы непрерывной интеграции, доставки и развертывания ПО
Под	Представляет собой запрос на запуск одного или нескольких контейнеров на одном узле, а также совокупность контейнеров, которые запускаются на платформе в ответ на такой запрос.
Узел	Сервер, управляемый платформой Kubernetes

2 Введение

2.1 Цели документа

Настоящий документ представляет из себя руководство по установке платформы Cinimex Microservice Platform.

2.2 Описание ПО

Полное наименование программы для ЭВМ: Cinimex Microservice Platform.

Разработчик: Акционерное общество «СИНИМЕКС-ИНФОРМАТИКА»

ОГРН 1027700553781

ИНН / КПП 773 414 8839 / 770 501 001

115184, г. Москва, улица Большая Татарская, д. 35, стр. 3

+7 495 955-7-955

tender@cinimex.ru

Область назначения: Программа предназначена для создания и управления надежной и масштабируемой среды исполнения автоматизированных систем и приложений (далее АС).

Область применения: Программа применяется при автоматизации процессов разработки, сборки, тестирования и развёртывания АС и для создания среды их исполнения.

Функциональные возможности программы:

1) Создание среды исполнения для АС с использованием средств легковесной виртуализации;
2) Автоматизация процессов разработки, сборки, тестирования, развёртывания и деинсталляции АС;
3) Управление процессами настройки АС и среды их исполнения;
4) Обеспечение контроля и управления доступом к АС;
5) Мониторинг здоровья и производительности АС и среды окружения;
6) Сбор, агрегация, доставка и хранение событий аудита;
7) Автоматическое восстановление работоспособности АС;
8) Автоматизация процессов масштабирования АС.

Язык программирования: yaml

2.3 Целевая аудитория и уровень подготовки

Документ предназначен для администраторов платформы.

Предполагается, что администратор платформы владеет как минимум следующим набором базовых знаний и навыков:

- Уверенное использование ОС Linux;
- Знание основных принципов, подходов и технологий системной интеграции;
- Умение работать с системой контроля версий git;
- Умение работать с системой управления конфигурацией Ansible
- Умение работать с Docker.
- Понимание основ работы платформы Kubernetes.

2.4 Связанные документы

Помимо настоящего руководства, администратору необходимо ознакомиться с инструкцией по администрированию платформы

Cinimex Microservice Platform. Руководство администратора.

3 Системные требования

3.1 Серверы

Для установки платформы должны быть подготовлены виртуальные или физические серверы на базе ОС Centos 7.6

Серверы имеют различные системные требования в зависимости от исполняемой роли в кластере. Выделяются 3 вида серверов (узлов кластера):

1) Мастер-узлы, предназначенные для запуска управляющей подсистемы кластера и хранилища состояния кластера (на базе etcd). Отказоустойчивость кластера достигается при наличии 3-х узлов, для таких узлов важны низкие задержки дисковой подсистемы и сети между такими узлами для синхронизации данных кластера etcd
2) Инфраструктурные узлы, предназначенные для развертывания различных компонент платформы, реализующих сервисы логирования, мониторинга, ci/cd, трассировки, обработки сообщений, баз данных. Для данных узлов важно обеспечить достаточный объем хранения данных для систем мониторинга и логирования, а также достаточный вычислительные ресурсы.
3) Рабочие узлы, предназначены для выполнения приложений, требования к узлам зависят от нагрузки.

В таблице приведены требования к узлам:

Характеристика		Роль		
		Мастер-узел	Инфраструктурный узел	Рабочий узел
Количество серверов	минимальное	3	3	1
	рекомендуемое	3 или 5	7	2 и более, в зависимости от нагрузки
CPU		2 ядра (4, если в кластере более 12 узлов)	4 ядра	4 ядра
RAM		8 Gb (16, если в кластере более 12 узлов)	16 Gb	16 Gb

Характеристика		Роль		
		Мастер-узел	Инфраструктурный узел	Рабочий узел
Storage	Системные разделы	40Gb (рекомендуется SSD для данных etcd)	40Gb	40Gb
	Хранилище для Docker	40Gb	120Gb	120Gb
	Хранилище для сервисов	Не требуется	От 100Gb	
Net		10GB/s	10GB/s	10GB/c
NFS хранилище		От 100Гб для использования в инфраструктурных сервисах, дополнительный объем в зависимости от требований, развернутых на рабочих узлах сервисов.		

3.2 Рабочая станция администратора

Для выполнения установки и администрирования платформы должен быть выделен компьютер (или виртуальная машина) на базе операционной системы Centos 7.6 (рекомендуется объем памяти не менее 2Гб, свободный объем диска от 40Гб, количество процессорных ядер 2 или более). Этот компьютер должен иметь доступ к серверам на которых планируется установить платформу по протоколу ssh.

3.3 Сетевые настройки

Для успешной установки требуется наличие доступа к Интернет на всех серверах и рабочей станции администратора, так как в процессе установки загружаются пакеты и образы.

Всем серверам должен быть назначен статический IPv4 адрес и разрешен IPv4 forwarding.

Должно быть настроено 2 wild-card доменных имени для публикации инфраструктурных и рабочих сервисов. Эти доменные имена должны указывать на набор IP-адресов, соответствующих инфраструктурным узлам кластера. Одно из доменных имен предназначено для переадресации на инфраструктурные вэб-сервисы, а другое – для переадресации на сервисы пользователей

4 Описание инсталлятора

4.1 Порядок загрузки дистрибутива

Дистрибутив с ПО Cinimex Microservice Platform поставляется в электронном виде в формате tar.gz-архива cinimex-microservice-platform.v1.0.0.tar.gz, содержащего в себе исходные коды и документацию.

Загрузка дистрибутива осуществляется с официального сайта компании изготовителя: https://www.cinimex.ru/solutions/microservice-platform/

Так же возможен вариант передачи дистрибутива заказчику напрямую. Ключ лицензии и пароль к архиву с дистрибутивом всегда передаются отдельно.

Перед выполнением установки необходимо распаковать содержимое архива, например, в директорию /opt/cmxas/install

4.2 Директории

components	Инфраструктурные компоненты кластера
environment	Директория с описанием стендов (окружений) через файлы ответов ansible
kubespray	Пакет kubespray
playbooks	Плейбуки Ansible
resources	Файлы с yaml-декларациями ресурсов кластера для использования в скриптах и плейбуках
roles	Роли ansible
util	Различные вспомогательные скрипты

4.3 Файлы

Makefile	все команды инсталлятора для запуска с помощью утилиты make
.env	Файл настройки на окружение, должен быть создан на базе прилагаемого .env.example файла.
requirements.txt	Файл зависимостей для установки модулей python через pip install
ansible.cfg	Настройки системы Ansible
.env.example	Пример файла с настройками окружения

4.4 Файлы ответов Ansible

Инсталлятор построен на базе системы управления конфигурацией Ansible 2.7. Параметры установки определяются в файлах-ответов в формате yaml и ini. В инсталляторе имеется пример файлов ответов, на базе которого нужно создать директорию в каталоге

environments/<имя окружения>, где <имя окружения> – условное название кластера, например stage, prod, test.

Список основных файлов ответов в директории environments/<имя окружения>

Название файла	**Описание**
inventory.ini	список серверов, с разбивкой по ролям
group_vars/all/ansible.yml	В файле задаются переменные, настраивающие доступ ansible к хостам кластера
group_vars/all/cluster.yml	В файле задаются переменные для настройки кластера, например, доменные имена для публикации сервисов
group_vars/all/local-storage.yml	В файле описывается разметка диска для создания хранилища, подключаемого к подам инфраструктурных компонент кластера.
host_vars/<имя хоста>.yml	Файл должен быть создан для каждого сервера, в файле указываются относящиеся к конкретному серверу данные, например, метки узлов или название дискового устройства устройства для настройки хранилища docker

5 Установка

5.1 Подготовка рабочей станции администратора

1) Установить git:

$ yum install git

2) Распаковать инсталлятор, например, в директорию /opt/cmxas

$ mkdir –p /opt/cmxas
$ tar –zxvf cinimex-microservice-platform.v1.0.0.tar.gz -C /opt/cmxas

3) Перейти в папку с инсталлятором:

$ cd /opt/cmxas/install

4) Создать новое окружение на базе поставляемого образца из директории environments/example

$ cp –R environment/example environment/<имя окружения>

5) Скопировавть .env.example в .env:

$ cp .env.example .env

Изменить переменные в .env:

env=<имя окружения>

6) Установить ansible:

$ sudo make ansible

7) Сгенерировать rsa ключ (или использовать имеющийся)

$ ssh-keygen

8) установить утилиты kubectl и helm

$ make utils

9) Инициализировать helm

$ helm init --client-only
$ helm repo add gitlab https://charts.gitlab.io/

5.2 Подготовка файлов ответов

Параметры работы инсталлятора задаются в файлах ответов Ansible, расположенных внутри директории environment/< имя окружения>

Прилагаемые образцы файлов ответов (из директории environment/example) содержат комментарии с описанием переменных, которые нужно настраивать.

5.2.1 Настройка списка серверов

Список серверов задается в файле ответов inventory.ini. Файл имеет формат ini и содержит набор секций (групп) в которые необходимо записать в соответствии с ролью имена серверов:

Название группы	**Назначение**
[all]	Перечислить все серверы
[kube-master]	Серверы, должны выполнять роль мастеров-узлов кластера
[etcd]	Серверы, на которых необходимо установить кластер etcd (рекомендуется использовать те же серверы что и для мастер-узлов)

Название группы	Назначение
[kube-node]	все остальные серверы который войдут в кластер, как инфраструктурные, так и рабочие узлы, признак роли узла потом задается в переменной node_labels
[nfs]	Узел, на которых нужно установить NFS сервер для создания хранилища данных приложений в кластере. Эта группа – опциональна, можно использовать уже готовый NFS сервер, на котором нужно экспортировать необходимые директории, аналогично файлу ответов group_vars/nfs.yml.

5.2.2 Настройка доступа к серверам

В файле ответов group_vars/all/ansible.yml необходимо указать опции ansible для доступа к серверам. По умолчанию настроен доступ от пользователя root с ключом пользователя, запускающего плейбуки.

5.2.3 Общие настройки для платформы

В файле ответов group_vars/all/cluster.yml настраиваются общие для всех серверов параметры. Обязательно нужно установить свои значения DNS имен в переменных cluster_infra_domain, cluster_devops_domain, cluster_apps_domain

В переменной cluster_nfs_server необходимо указать имя NFS сервера

5.2.4 Настройки компонент платформы

В файлах ответов group_vars/all/(logging|monitoring|gitlab|activemq).yml настраиваются параметры установки соответствующих компонентов платформы.

5.2.5 Индивидуальные настройки серверов

Для каждого сервера нужно создать файл host_vars/<имя хоста>.yml и в переменной node_labels указать назначение меток которые будут указывать как платформе следует использовать эти узлы

as.cmx.ru/node-role: infra as.cmx.ru/infra: "true"	признак инфраструктурного узла (в отличие от мастеров и рабочих узлов)
as.cmx.ru/monitoring: "true"	признак узла, на который будет установлен компонент мониторинга, минимум один узел
as.cmx.ru/logging: "true"	признак узла, на который будет установлен компонент логирования, минимум 3 узла

as.cmx.ru/devops: "true"	признак узла, на котором будет установлен gitlab и другие devops инструменты
as.cmx.ru/activemq: "true"	Признак узла для установки ActiveMQ
as.cmx.ru/work: "true"	Признак рабочего узла

Для серверов, предназначенных для инфраструктурных узлов требуется настроить параметры разметки дисков для создания томов, монтируемых к подам инфраструктурных компонент кластера. Эти параметры задаются в файле ответов group_vars/all/local-storage.yml:

- в переменную local_storage_nodes указать на каком узле какой том должен быть размещен, имена узлов использовать короткие (без доменного имени)
- размеры дисков задаются в local_storage_volumes и используются при развертывании инфраструктурных компонент платформы
- достаточность размера не контролируется автоматически, если суммарный объем превысит размер диска, то установка завершится с ошибкой

доступные варианты являются ключами словаря в переменной local_storage_volumes:

elasticdata	хранилище для данных логирования (elasticsearch) должно быть 3 всего от 20Гб
elasticmaster	хранилище для мастер-узлов логирования (elasticsearch) должно быть 3 всего от 5Гб
logstash	хранилище для logstash, должно совпадать с количеством реплик
kibana	хранилище для kibana - 1 шт.
prometheus	хранилище для компоненты мониторинга - prometheus 2 шт., от 20Гб
alertmanager	хранилище для компоненты мониторинга - alertmanager 3 шт., от 5Гб

5.3 Подготовка серверов

1) Распространить ssh ключ на все узлы, будет запрос пароля для каждого хоста.

./util/install-key.sh

2) Проверить, что DNS имена настроены корректно, командой вида

dig cluster.example.com +short

убедиться, что все доменные имена, указанные в cluster.yml разрешаются на IP адреса инфраструктурных узлов кластера

5.4 Запуск установки

Полная установка кластера и всех компонентов выполняется командой

$ make install

Для установки не требуются права суперпользователя на рабочей станции администратора (кроме установки утилит и сертификатов). Самая последняя задача установки - копирование сертификатов, может запросить пароль sudo.

6 Проверка установки

Для того, чтобы убедиться в том, что все компоненты работают, необходимо выполнить команду

$ kubectl get pods --field-selector=status.phase!=Running --all-namespaces

которая выведет список всех Подов, которые не смогли запуститься. Запуск всех компонент после установки может занять некоторое времени (обычно от 5 до 30 минут), в течение которого можно контролировать с помощью указанной команды процесс запуска подов.

Как только список окажется пуст – платформа готова к работе.

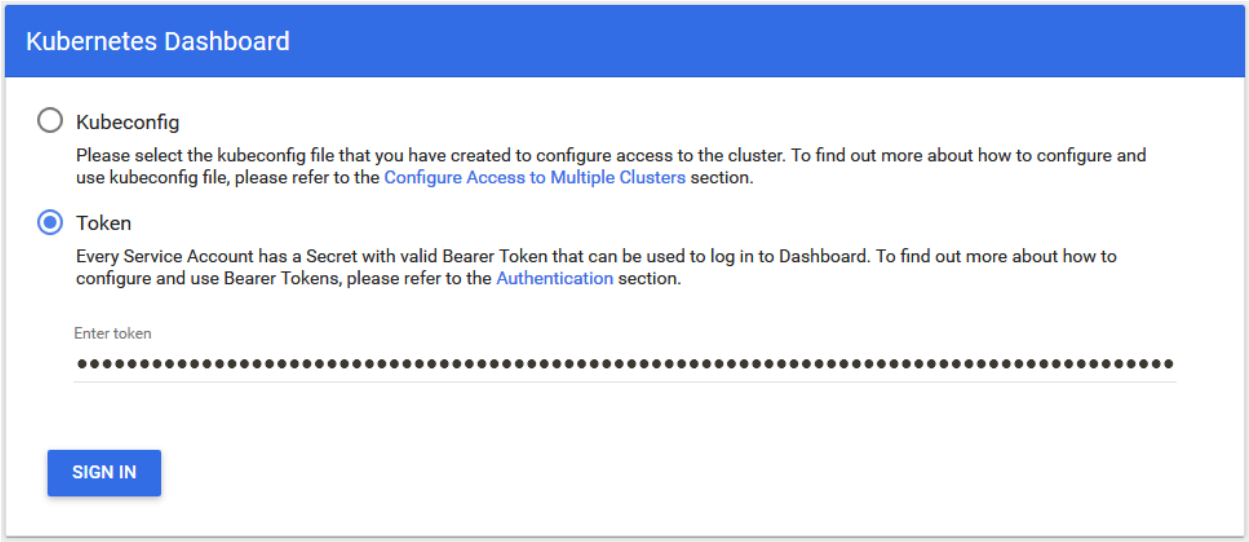
После завершения установки необходимо получить пароли и токены для доступа к панели управления кластера и инфраструктурным сервисам платформы.

Список токенов и паролей выводится командой: make password

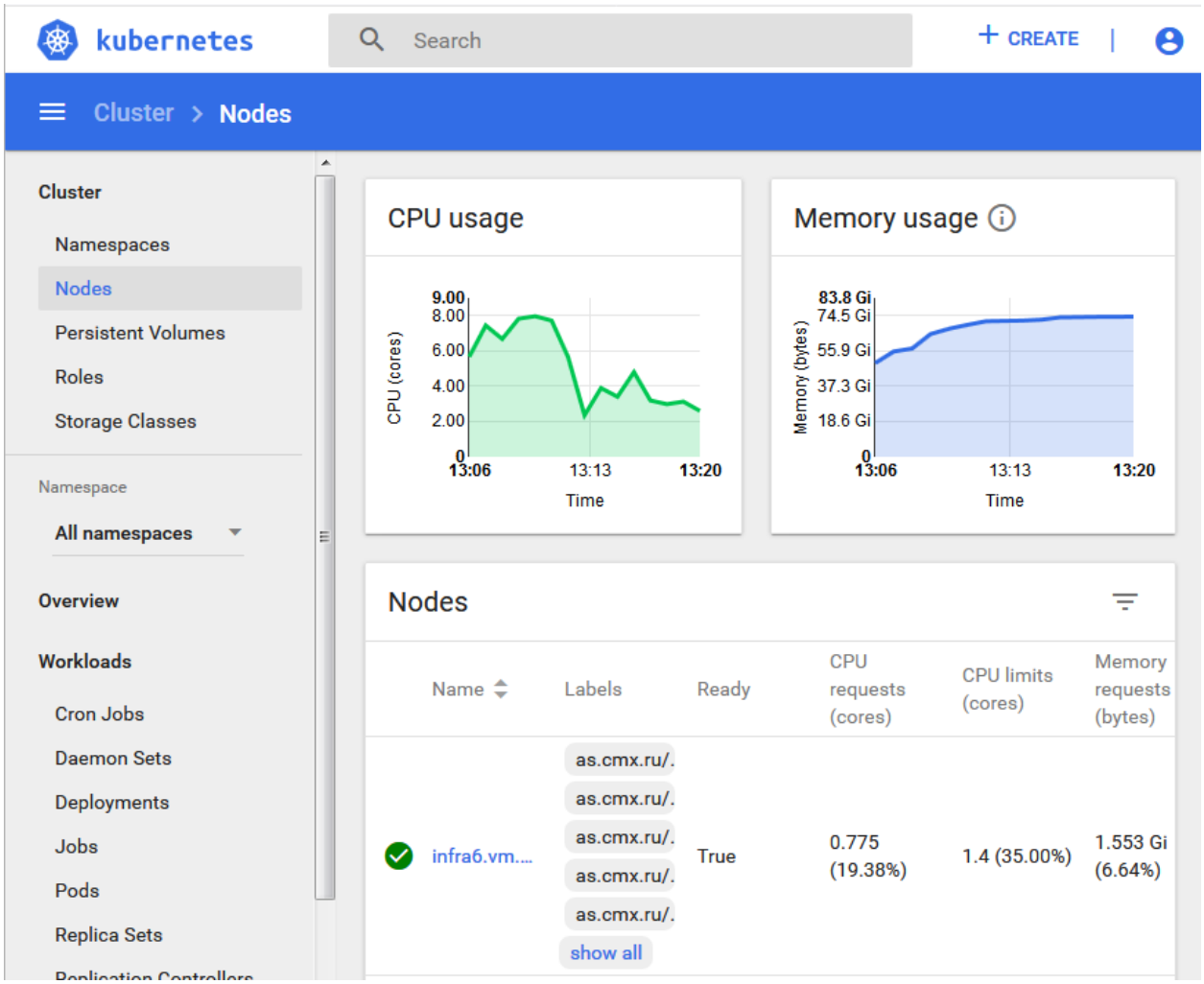
```
[root@prm installer-temp]# make password
scp -q root@m1demo.vm.cmx.ru:/root/.kube/config ~/.kube/config
Copied kube config from master to home
========
Token for Dashboard:

eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb
dC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi1c2VyLXRva2
WI4N2xxIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLml
NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiJlYmJmYjIwZi05ZWYzLTExZTktYWFjYy0wMDUwNTZiYzk0YjEiLCJzdWIiOiJzeXN0
6c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.sH8ywHjmIcHW3k4nodgmj51zDrmy2voHfDvEdF7jy_fqupU0d0JZJregoe7S
FpRcwvBO9CHCP1M_Fo1vXP_Fm_CT5GCWfyKP5_VRQfVZ1guN3LFFnk4mhIEbc_7n9mZg2qUG0M-yPIGpNuUNJM2EJveafUntP_B1w5qij38RVcwIa_
DvAi7S3kqZG-wPR0iGrtGXTW9bfoa8ggIUbPN8ADhUv2t0hsiK3c5n7zP4VqcMiDAT06x4EeAcZnaOqbvo7neaPMrpLQaoNxmovT7W6g15wunS1Gm3
qy0BVK78bUUIzKAOucukU7K8e1wWoDgPy5i4rvD8AhAmQ
========
Gitlab initial root password:

WshIdqTY8Cm1GQtElistoWUAO9MNN38ZnK1rU4vhlndBHqroTcE0EcDBYGfW0ohr
========
Grafana admin password:

prom-operator
```

Для проверки работы панели управления открыть в браузере страницу с любым из IP адресов инфраструктурных узлов и портом 30443

Для авторизации необходимо будет указать токен из вывода предыдущей команды



После логина в панель управления можно просмотреть состояние различных компонент платформы:



Для вывода списка всех опубликованных сервисов выполнить команду kubectl get ingress --all-namespaces

В колонке HOSTS будут выведены все адреса для использования в браузере.

В колонке PORTS будут указан порты

```
[root@prm installer-temp]# kubectl get ingress --all-namespaces
NAMESPACE             NAME                                HOSTS                                      ADDRESS

     PORTS     AGE
cmxas-activemq        activemq-ui                         activemq.infra.k8sdemo.vm.cmx.ru       192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80        4d22h
cmxas-devops          gitlab-minio                        minio.infra.k8sdemo.vm.cmx.ru          192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-devops          gitlab-registry                     registry.infra.k8sdemo.vm.cmx.ru       192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-devops          gitlab-unicorn                      gitlab.infra.k8sdemo.vm.cmx.ru         192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-infra-logging   cerebro                             cerebro.infra.k8sdemo.vm.cmx.ru        192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-infra-logging   infra-elk-elasticsearch-client      elastic.infra.k8sdemo.vm.cmx.ru        192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-infra-logging   infra-elk-kibana                    kibana.infra.k8sdemo.vm.cmx.ru         192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-infra-mon       prometheus-operator-alertmanager    alertmanager.infra.k8sdemo.vm.cmx.ru   192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-infra-mon       prometheus-operator-grafana         grafana.infra.k8sdemo.vm.cmx.ru        192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
cmxas-infra-mon       prometheus-operator-prometheus      prometheus.infra.k8sdemo.vm.cmx.ru     192.168.17.141,192
.168.17.142,192.168.17.143,192.168.17.144,192.168.17.145,192.168.17.146,192.168.17.147,192.168.17.148,192.168.17.1
49   80, 443   4d22h
```