

# Документация по установке и эксплуатации «Nodus»

## Оглавление

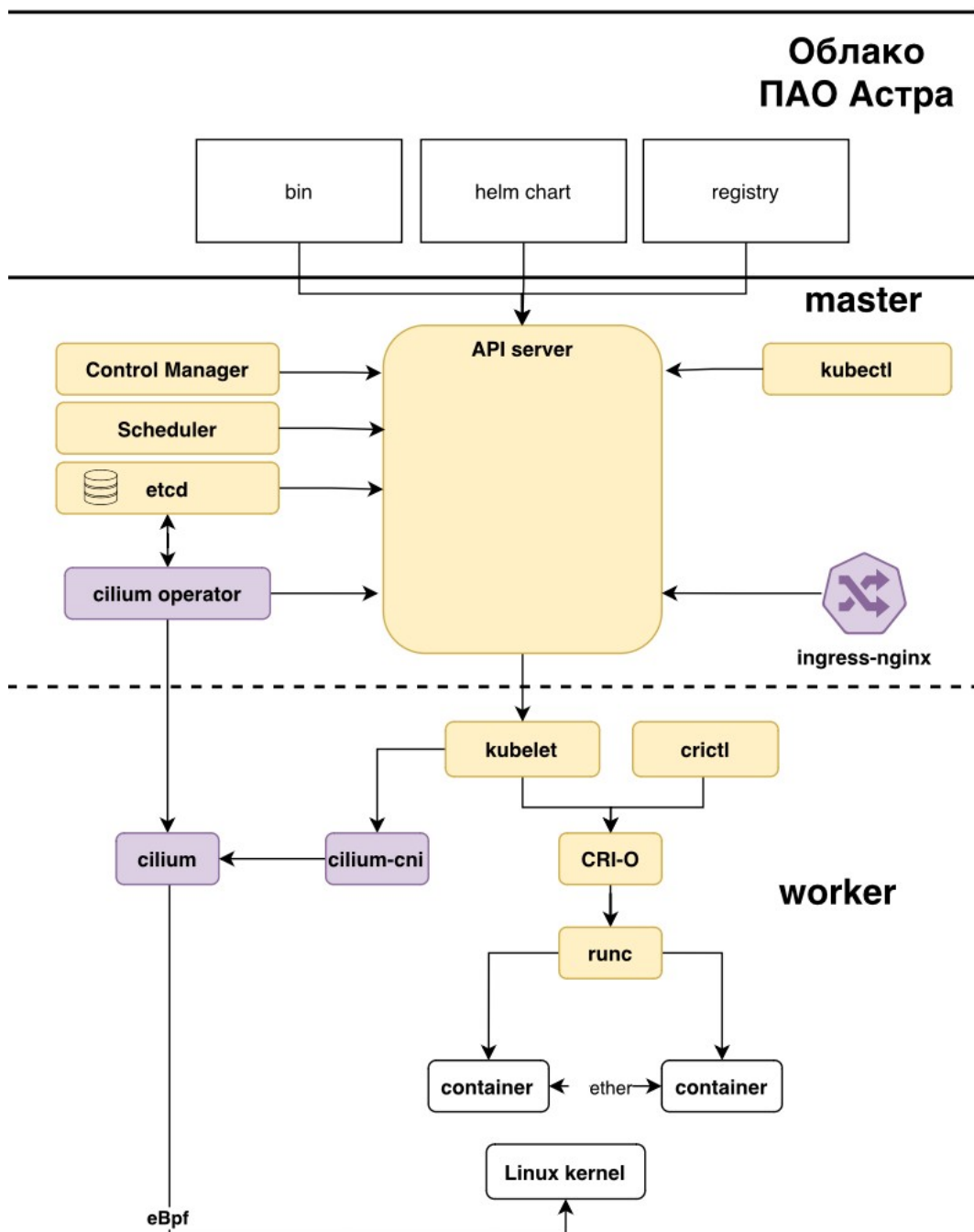
<b>1</b>	<b>Обзор платформы.....</b>	<b>3</b>
1.1	Описание.....	3
1.2	Архитектура продукта.....	3
1.3	Описание компонентов.....	4
1.4	Версии системных компонентов.....	5
<b>2</b>	<b>Процесс установки кластера.....</b>	<b>6</b>
2.1	Описание процесса.....	6
2.2	Схема разворачивания кластера.....	6
2.3	Предварительная настройка кластера Kubernetes.....	6
<b>3</b>	<b>Масштабирование кластера.....</b>	<b>9</b>
3.1	Основные шаги масштабирования кластера с использованием Kubespray:.....	9

# 1 Обзор платформы

## 1.1 Описание

Nodus позволяет управлять контейнеризированными приложениями в масштабируемой среде с минимальными затратами на настройку. Он поддерживает как государственные облака, так и частные центры обработки данных.

## 1.2 Архитектура продукта



## 1.3 Описание компонентов

Платформа состоит из следующих основных компонентов:

**Kubelet:** агент, который работает на каждом узле кластера и отвечает за обеспечение работоспособности контейнеров в Pod'ах.

**Kubectl:** CLI инструмент, предназначенный для взаимодействия с сервером API Kubernetes. Он позволяет создавать, обновлять и удалять компоненты кластера Kubernetes.

**Kube-apiserver:** главный компонент управления кластером Kubernetes, предназначенный для обработки REST операций и обновления объектов в кластере Kubernetes.

**Kube-proxy:** компонент, который обеспечивает реализацию сетевого прокси и балансировщика нагрузки в Kubernetes. Отвечает за маршрутизацию трафика к подходящим контейнерам.

**Kube-controller-manager:** компонента, которая запускает контроллеры кластера Kubernetes, такие как контроллеры узлов, репликаций и другие.

**Kube-scheduler:** отвечает за распределение Pod'ов по узлам кластера в зависимости от различных метрик, таких как ресурсы узла и требования по предпочтениям местонахождения.

**ETCD:** база данных типа ключ-значение, используемая для хранения всех данных кластера, она обеспечивает сохранность и доступность данных.

**Cilium:** решение для сетевых политик и маршрутизации в Kubernetes на основе технологии eBPF.

**CRI-O:** Это реализация интерфейса контейнеров для Open Containers Initiative (OCI), позволяющая запускать любые контейнеры, соответствующие стандарту OCI.

**Nginx-ingress:** прокси и балансировщик нагрузки для входящего трафика в Kubernetes, который использует nginx в своей основе.

**Runc:** бинарный файл, который используется в качестве рантайма OCI для запуска контейнеров.

**Kubeadm:** инструмент, предназначенный для инициализации кластера Kubernetes. Обеспечивает минимальную функциональность для запуска поставщика на основе Kubernetes.

## 1.4 Версии системных компонентов

Наименование	Версия
ОС	Astra 1.7.4 , 1.7.5
Kubeadm	1.26.5
Kubelet	1.26.5
Kubectl	1.26.5
Kube-apiserver	1.26.5
Kube-proxy	1.26.5
Kube-controller-manager	1.26.
Kube-scheduler	1.26.5
ETCD	v3.5.6
Calico	v3.25.1
CRI-O	1.26.3
Kube-nginx	nginx:1.23.2-alpine
K8s-dns-node-cache	1.22.18
Cluster-proportional-autoscaler	v1.8.8
Nginx-ingress	3.1.1
Runc	1.1.7

## 2 Процесс установки кластера

### 2.1 Описание процесса



Используемая версия кастомизации набора ролей Kubespray не зафиксирована и проходит этапы тестирования.

Для инсталляции платформы контейнеризации "**Nodus**" используется кастомизированный набор ролей [kubespray](#), с сохранением основных функциональностей:

- Конфигурирование окружения виртуальных машин;
- Кастомизация CRI-O;
- Кастомизация Cilium;
- Кастомизация Etcd.

Основные кейсы с использованием набора ролей:

- Раскатка кластера;
- Масштабирование control plane / node;
- Обновление кластера и его компонентов;
- Удаление ноды.

### 2.2 Схема разворачивания кластера

Схема разворачивания предполагается несколькими этапами:

- Формирование инвентарного файла;
- Создание виртуальных машин, в зависимости от заданных параметров:
  - количества VM,
  - параметрах ресурсов,
  - адресации, используемого образа ОС,
  - окружения ОС
- Формирование роли виртуальных машин, т.е. формирование инвентарного файла, где присутствует разграничение на узел плоскости управления и рабочий узел;
- "Раскатка роли" – выполнение ansible-кода, получение кластера на выходе;
- "Стягивание" образов с внешних репозиториев.

### 2.3 Предварительная настройка кластера Kubernetes

Имеется возможность сконфигурировать кластер Kubernetes:

- Настройки ОС
- CRI-O
- Cilium
- Etcd
- k8s addons

На выходе мы получим готовый к работе кластер. Если нам требуется установить его с нуля. Установка кластера происходит по следующему алгоритму:

1. Установите Docker: Убедитесь, что на вашей системе установлен Docker. Можно загрузить и установить Docker с официального сайта [Docker](#).
2. Производить конфигурацию нужно из под root пользователя. Выполняем команду ниже для переключения на root:

```
sudo su
```

3. Создайте docker volume, для того, что бы иметь прямой доступ к каталогу с kubescape inventory из вашей хостовой системы:

```
docker volume create kubescape
```

4. Запустите контейнер с подготовленной средой для установки кластера:

Эта команда запускает контейнер Docker, внутри которого будет выполняться установка Kubernetes с использованием Kubescape. Ключ -v монтирует каталог с вашим приватным ключом для доступа к хостам по ssh. До двоеточия указывается путь к ключу на хосте, после указываем в какой каталог хотим смонтировать каталог с хоста.

Если доступ к хостам осуществляется по связке "пользователь:пароль" параметр "-v ~/.ssh/id\_rsa:/root/.ssh/id\_rsa" нужно убрать

```
docker run --name kubescape -d -i -v kubescape:/kubescape -v
~/.ssh/id_rsa:/root/.ssh/id_rsa
registry.astralinux.ru/rubik/kubescape:v0.0.1a
```

5. Создайте инвентарный файл. Скопируйте шаблон инвентарного файла Kubescape, выполнив следующую команду:

```
cp -rpf /var/lib/docker/volumes/kubescape/_data/inventory/sample
/var/lib/docker/volumes/kubescape/_data/inventory/mycluster
```

6. Доступные настройки:

- o *inventory.ini*: Основной файл инвентаря, в котором определяются группы хостов и их параметры. Можно указать IP-адреса или DNS-имена узлов, а также их роли (мастеры, рабочие узлы и т. д.).
- o *group\_vars/all/all.yml*: Файл с общими переменными, применяемыми ко всем хостам. Здесь устанавливаются глобальные настройки, такие как версия Kubernetes, сетевые параметры и т. д.
- o *group\_vars/k8s-cluster/k8s-cluster.yml*: Файл с переменными для кластера Kubernetes. Здесь можно настроить параметры, связанные с мастерами, рабочими узлами, сетью и другими компонентами Kubernetes.



**Без глубокого понимания работы кластера k8s и его зависимостей, рекомендуется ограничиться редактированием только файла *inventory.ini*.**

7. Редактируйте инвентарный файл. Отредактируйте инвентарный файл *inventory/mycluster/inventory.ini*, чтобы определить свою конфигурацию кластера. Например, укажите IP-адреса или DNS-имена хостов, роли хостов (мастера, рабочего узла и т.д.) и другие параметры, если требуется:

```
vi
/var/lib/docker/volumes/kubescape/_data/inventory/mycluster/inve
ntory.ini
```

Кроме IP адресов или DNS имен хостов, при необходимости можно указать следующие параметры:

```
`ansible_user`: Логин пользователя на удаленных узлах.  
`ansible_ssh_private_key_file`: Путь к закрытому ключу SSH.  
`kube_master_name`: Имя главного узла Kubernetes. `kube_node_name`:  
Шаблон имен для рабочих узлов Kubernetes.  
`kube_version`: Версия Kubernetes для установки.
```

8. Передаем команды для установки кластера в контейнер:

С помощью команды ниже можно просмотреть текущую конфигурацию инвентаря и убедиться, что все хосты и группы определены корректно.

```
docker exec kubespray bash -c "ansible-playbook -i  
inventory/mycluster/inventory.ini --list"
```

Эта команда запускает плейбук Ansible cluster.yml, который настроит и установит кластер Kubernetes на указанных хостах, используя инвентарный файл inventory/mycluster/inventory.ini:  
Доп. Ключи:

- o -k - для указания пароля подключения по ssh (используется в случае отсутствия ключевой пары для подключения по ssh )
- o -K - для указания пароля пользователя, под которым подключаемся к удаленным хостам. (Используется для запуска из под sudo при наличии ключа -b)
- o -u - для указания пользователя с помощью которого будет производиться подключение к удаленным хостам

```
docker exec kubespray bash -c "ansible-playbook -i  
inventory/mycluster/inventory.ini cluster.yml -b"
```

9. Дождитесь завершения установки: После запуска плейбука Ansible дождитесь его завершения. Kubespray будет автоматически устанавливать и настраивать каждый компонент кластера Kubernetes на указанных хостах.



Текущий алгоритм указан только для ознакомления.



## 3 Масштабирование кластера



Масштабирование кластера Kubernetes может быть сложным процессом, и требуется аккуратность при обновлении конфигурации и запуске Kubespray.

Рекомендуется сначала ознакомиться с документацией Kubespray и выполнить тестовое масштабирование на отдельной среде перед применением изменений на рабочем кластере.

Чтобы масштабировать кластер, используя Kubespray, вам понадобится настроенная инсталляция Kubespray на сервере управления кластером. Если у вас уже есть установленный Kubespray, вы можете перейти к следующим шагам. Если нет, вам потребуется установить его, следуя официальной документации.

### 3.1 Основные шаги масштабирования кластера с использованием Kubespray:

#### 1. Подготовка инвентарного файла:

Создайте новый инвентарный файл или измените существующий, чтобы указать новые рабочие узлы, которые вы хотите добавить или удалить из кластера. Инвентаризационный файл определяет конфигурацию вашего кластера Kubernetes, включая список рабочих узлов, конфигурацию сети, параметры аутентификации и другие настройки.

#### 2. Обновление конфигурации:

Используя инвентарный файл, вы должны обновить конфигурацию Kubespray, чтобы отразить новые рабочие узлы. Обычно это включает в себя добавление новых IP-адресов и имен узлов, а также обновление параметров конфигурации сети.

#### 3. Запуск Kubespray:

Запустите Kubespray с обновленной конфигурацией, используя команду `ansible-playbook`. Например, для запуска масштабирования кластера с использованием инвентарного файла `inventory.ini` вы можете выполнить следующую команду:

```
ansible-playbook -i inventory.ini -u <username> cluster.yml
```

Здесь `<username>` - ваше имя пользователя на рабочих узлах кластера, а `cluster.yml` - основной плейбук Kubespray, который выполняет масштабирование и настройку кластера.

#### 4. Дождитесь завершения выполнения:

Kubespray автоматически масштабирует кластер, добавляя или удаляя рабочие узлы в соответствии с вашей конфигурацией. Дождитесь завершения выполнения плейбука Kubespray и проверьте результаты.

#### 5. Проверка кластера:

После завершения масштабирования проверьте состояние кластера, чтобы убедиться, что новые рабочие узлы корректно добавлены и работают. Вы можете использовать команды `kubectl get nodes` или `kubectl get pods` для проверки состояния кластера и его компонент.

**Удаление рабочих узлов из кластера выполняется аналогичным образом.** Вам нужно будет обновить инвентарный файл, указав узлы для удаления, и запустить Kubespray с обновленной конфигурацией.