Способы установки платформы

1. О способах установки

Установка Nova Container Platform поддерживает различное количество конфигураций и сценариев, которые рассмотрены в данном разделе документации.

1.1. Инструменты установки

В качестве инструмента установки платформы используется консольная утилита nova-ctl, которую можно использовать в следующих режимах:

- **Интерактивный**: с помощью `nova-ctl` пользователь в интерактивном режиме может получить шаблоны установочных манифестов, подтвердить планируемые изменения в инфраструктуре (например, при автоматическом создании ВМ в виртуальных средах), выполнить развертывание платформы.
- **Автоматический**: пользователь может заранее подготовить необходимые манифесты для установки платформы и использовать `nova-ctl` в режиме автоматического подтверждения всех операций. Данный режим может быть полезен, если требуется автоматизация процесса по установке платформы.

1.2. Инфраструктура для установки

В зависимости от доступной инфраструктуры установка платформы может быть выполнена тремя методами:

- **Installer-provisioned infrastructure (IPI)**: Автоматизированный метод развертывания в инфраструктуре, подготовленной узлом `nova-ctl` для управления платформой. Данный метод может применяться в средах виртуализации и облачных средах. За счет взаимодействия `nova-ctl` с API провайдера инфраструктуры необходимые узлы платформы (виртуальные машины) могут быть подготовлены автоматически.
- **User-provisioned infrastructure (UPI)**: Автоматизированный метод развертывания в инфраструктуре, подготовленной пользователем. Данный метод обеспечивает полный контроль и кастомизацию инфраструктурного слоя. Перед установкой платформы пользователь самостоятельно подготавливает необходимые узлы платформы согласно представленным в документации требованиям. Данный метод подходит для развертывания в средах, где взаимодействие узла `nova-ctl` для управления

платформой с API провайдера инфраструктуры недоступно, а также в случае развертывания платформы на узлах без использования средств виртуализации.

- **Гибридный**: Метод используется, если планируется размещать все или часть рабочих узлов кластера в отдельных средах виртуализации, облаках или на физических серверах.

> **Информация**
>
> Более подробную информацию по провайдерам инфраструктуры вы можете получить в разделе документации Провайдеры инфраструктуры.

1.3. Сетевое окружение

Установка платформы может быть выполнена с учетом требований к сетевому окружению в используемой инфраструктуре:

- **Онлайн**: Когда все узлы кластера имеют прямой доступ в сеть Интернет. Данный метод является самым простым и быстрым, используются публичные репозитории Nova Container Platform.
- **Онлайн через HTTP-прокси**: Установка может быть выполнена с использованием на узлах кластера HTTP-прокси организации для доступа к публичным репозиториям Nova Container Platform.
- **Офлайн**: Установка платформы выполняется в полностью закрытом сетевом окружении. Для установки используется предварительно настроенный сервер управления Nova Universe, который предоставляет все необходимые репозитории. Дальнейшее обновление Nova Container Platform выполняется также с использованием Nova Universe без доступа к сети Интернет.

1.4. Тип кластера

В зависимости от требований к количеству ресурсов и отказоуйстойчивости компонентов, вы можете развертывать кластера следующих типов:

- **Минимальный**: Кластер для целей тестирования и разработки. Поддерживается установка минимум на 3 узла (1 мастер-узел, 1 инфраструктурный и 1 рабочий узлы).
- **Минимальный с выделенным балансировщиком**: Данный кластер аналогичен минимальному, однако балансировщик нагрузки (сервис-прокси) Ingress Nginx размещается на отдельных узлах. Вы можете размещать данный узел в отдельном сетевом сегменте для повышения безопасности платформы.
- **Высокодоступный**: Кластер для продуктивных окружений. Поддерживается установка минимум на 8 узлов (3 мастер-узла, 3 инфраструктурных и 2 рабочих узла). Компоненты

Kubernetes и платформенные сервисы Nova Container Platform разворачиваются в нескольких репликах.

- **Высокодоступный с выделенным балансировщиком**: Данный кластер аналогичен высокодоступному, однако балансировщики нагрузки (сервис-прокси) Ingress Nginx размещаются на отдельных узлах. Вы можете размещать данные узлы в отдельном сетевом сегменте для повышения безопасности платформы.

> **Информация**
>
> В разделе документации Выбор метода установки вы можете найти рекомендации по использованию различных конфигураций Nova Container Platform. Ознакомьтесь с данным разделом перед установкой платформы.

1.5. Операционные системы и среды

Nova Container Platform поддерживает различные среды для установки и операционные системы. Подробную информацию вы можете найти в разделе Перечень матриц совместимости и протестированных интеграций.

1.6. Кастомизация кластера на этапе установки

Вы можете установить как кластер с настройками по умолчанию, так и указать дополнительные настройки кластера в различных контекстах. Это регулируется обязательными и опциональными параметрами API, используемом в конфигурационном манифесте `nova-deployment-conf.yaml`.

- Для любого метода установки вы можете предварительно определить группы узлов кластера и указать для узлов настройки Kubernetes `Labels`, `Annotations`, `Taints`.
- В контексте провайдера инфраструктуры при развертывании кластера методом IPI вы можете указать отдельные настройки для групп узлов кластера, например, шаблоны BM, сетевые настройки, количество ресурсов BM, хранилища данных.
- В контексте кластера Kubernetes вы можете предварительно определить дополнительные роли узлов, указать CIDR подсетей Kubernetes, сконфигурировать службы DNS.
- В контексте установки базового модуля ПО вы можете указать параметры развертывания инфраструктуры PKI и параметры DNS-зоны для размещения служебных веб-сервисов платформы.

2. О узле nova-ctl для управления платформой

Узел nova-ctl выполняет основные задачи для развертывания и обслуживания кластеров Nova Container Platform, а именно:

- развертывает виртуальные машин кластера (при использовании метода развертывания IPI)
- настраивает внутренние службы PKI
- управляет хостовыми агентами Nova Host Agent
- координирует процессы установки, обновления и масштабирования платформы

Для установки платформы `nova-ctl` использует конфигурационный манифест объекта `cluster` с типом `Infrastructure` в API-группе `config.nova-platform.io`. Пользователь может получить шаблон манифеста с помощью команды `nova-ctl init`, далее заполнить манифест в соответствии со спецификацией API, затем выполнить установку платформы с помощью команды `nova-ctl bootstrap`.

Выполнив команду `nova-ctl init` пользователь получает в рабочей директории новую директорию `nova-configs`, в которой находится шаблон конфигурационного манифеста `nova-deployment-conf.yaml`. В данной директории также могут находиться другие подготовленные пользователем конфигурационные манифесты (например, для установки платформы с использованием HTTP-прокси сервера).

Процесс развертывания платформы состоит из следующих этапов:

1. Утилита `nova-ctl` использует подготовленные пользователем манифесты и определяет метод развертывания платформы. В случае использования метода *IPI*, `nova-ctl` взаимодействует с API провайдера инфраструктуры для создания и настройки ВМ.
2. Далее `nova-ctl` проверяет доступность созданных ВМ и по мере их готовности устанавливает в ОС минимальный набор необходимых пакетов и служб.
3. На мастер-узлах платформы `nova-ctl` выполняет установку и настройку *StarVault*: настраивает необходимые политики доступа, секреты, инфраструктуру PKI, провайдера OIDC, роли аутентификации Kubernetes.
4. На всех узлах платформы `nova-ctl` запускает агенты Nova Host Agent, которые подключаются к Configuration Manager для получения сценария установки и применяют его локально.
5. После установки основных компонентов кластера Kubernetes и проверки их работоспособности выполняется установку в кластер базового модуля ПО.
6. По завершении установки компонентов базового модуля ПО `nova-ctl` передает пользователю учетные данные и информацию о доступе к развернутому кластеру.

Архитектура узлов в Nova Container Platform

1. Об узлах платформы

Узлом в Nova Container Platform является виртуальная машина или физический сервер, на котором размещаются компоненты платформы и среды Kubernetes, а также пользовательские приложения. Для стабильной работы ваших приложений важно следить за состоянием узлов, их метриками, своевременно реагировать на возникающие ошибки.

В Nova Container Platform вы можете получить информацию об узле, его конфигурации и событиях через объект *Node* в Kubernetes. Для этого можно использовать как утилиту `kubectl`, так и веб-консоль Nova.

Следующие компоненты каждого узла непосредственно обеспечивают работу *Pod* в среде Kubernetes:

Среда исполнения контейнеров (Container Runtime): обеспечивает работу контейнеров в ОС. В Nova Container Platform используется среда Containerd, однако, существуют и альтернативные решения, например, cri-o, Docker.

Kubelet: *Kubelet* работает на каждом узле платформы, выполняет роль агента и промежуточного звена между Kubernetes и службами ОС, обрабатывает запросы на запуск, удаление или изменение контейнеров в составе *Pod*, контролирует состояние контейнеров, обслуживает задачи на настройке сетевых политик и форвардинга портов. *Kubelet* управляет только теми контейнерами, создание которых было выполнено через Kubernetes.

Kube-proxy: основной задачей компонента является отслеживание изменений объектов *Service* и *Endpoints* в Kubernetes API и трансляция изменений в сетевые правила ОС. В Nova Container Platform компонент *Kube-proxy* работает в режиме *IPVS*.

На диаграмме ниже схематично отображены компоненты Kubelet и Kube-proxy.

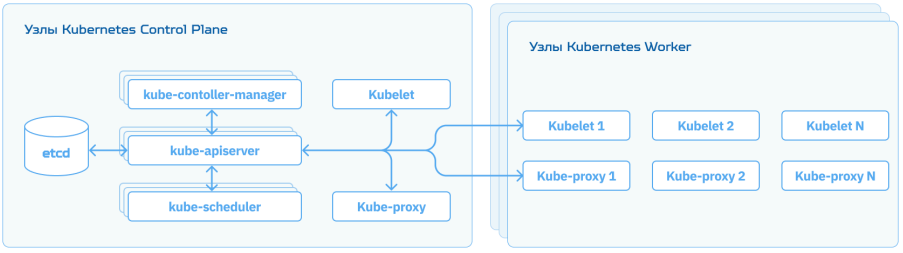

Рисунок 1. Компоненты Kubelet и Kube-proxy

2. Роли узлов

В Nova Container Platform узлы платформы могут иметь разные логические роли, а также объединяться в группы узлов.

Поскольку в Kubernetes роль определяется стандартной меткой *(Label)* узла, например, `node-role.kubernetes.io/control-plane`, то вы можете указать дополнительные метки, чтобы сгруппировать узлы кластера по определенному признаку.

При формировании групп узлов на этапе установки платформы вы также можете указать дополнительные параметры Labels, Taints и Tolerations.

> Дополнительная информация по формированию групп узлов на этапе установки платформы представлена в разделе Описание процессов установки и обновления.

По умолчанию в Nova Container Platform используется несколько преднастроенных ролей:

- `control-plane`: Мастер-узлы. Содержат ключевые компоненты Nova Container Platform и Kubernetes.
- `infra`: Инфраструктурные узлы. Содержат служебные компоненты Nova Container Platform.
- `ingress`: Выделенные узлы для балансировки входящего трафика. На узлах размещается дополнительный контроллер Ingress Nginx.
- `worker`: Рабочие узлы для размещения пользовательских сервисов, приложений и различных нагрузок.

По имени роли определяется, какой набор базовых компонентов будет установлен на узел кластера. Ниже представлена обобщенная схема с указанием ролей узлов и их основными компонентами.

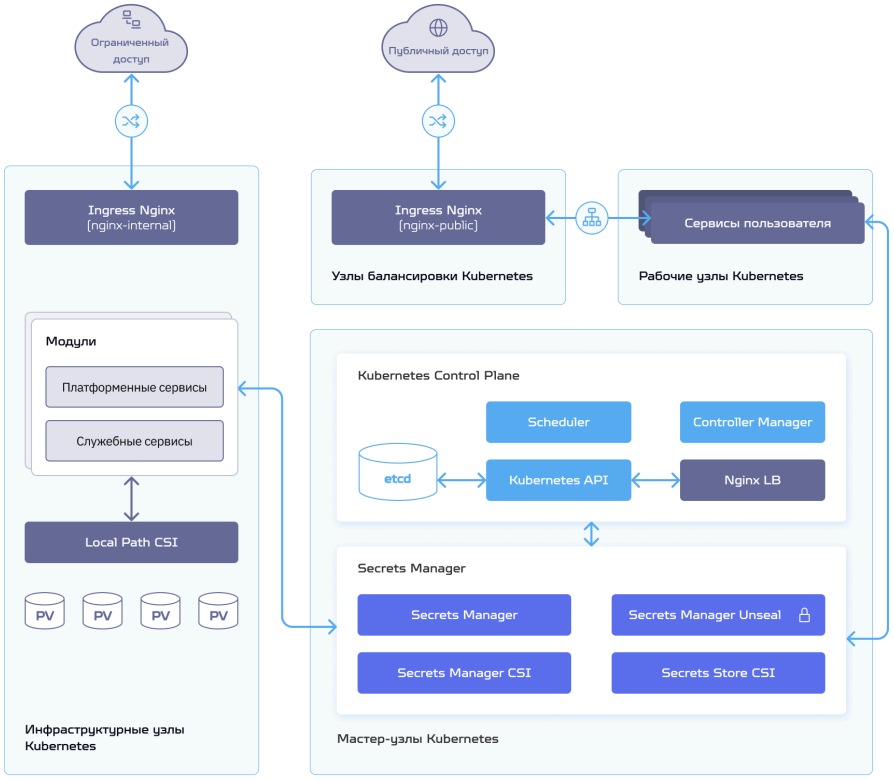

Рисунок 2. Роли узлов в Nova Container Platform

2.1. Мастер-узлы

В кластере Kubernetes мастер-узлы имеют специальную метку `node-role.kubernetes.io/control-plane`. Мастер узлы обеспечивают работу ключевых компонентов среды Kubernetes, таких как Kubernetes API, Etcd, Kubernetes controller manager, Kubernetes Scheduler и другие.

В таблице ниже представлен перечень компонентов Kubernetes на мастер-узлах.

Компонент	**Описание**
Kubernetes API	Компонент, который предоставляет интерфейс взаимодействия (API) остальным компонентам кластера и пользователям.
Etcd	Основное хранилище данных Kubernetes. Множество компонентов кластера следят за состояниями объектов в Etcd через Kubernetes API и приводят данное состояние к описанному (желаемому).
Kubernetes controller manager	Компонент, обеспечивающий основные циклы управления Kubernetes. Controller Manager отслеживает конфигурации в Etcd через Kubernetes API и вносит необходимые изменения для достижения указанного состояния какого-либо компонента.
Kubernetes scheduler	Компонент, задача которого состоит в определении подходящих узлов для вновь создаваемых `Pod`.

Кроме компонентов Kubernetes Control Plane на мастер-узлах располагается хранилище секретов StarVault, локальный балансировщик Nginx, сервис автоматической разблокировки StarVault Unseal. Данные сервисы запускаются в ОС как службы systemd, в виде статических `Pod` и `Deployment` в Kubernetes.

В таблице ниже представлен перечень дополнительных компонентов Nova Container Platform на мастер-узлах.

Компонент	**Тип**	**Описание**
StarVault	Служба Systemd	Компонент Nova Container Platform для организации внешнего хранилища секретов, инфраструктуры PKI и OAuth-провайдера.
StarVault Unseal	Служба Systemd	Служба для автоматической разблокировки (распечатывания) хранилища секретов StarVault в случае его полного перезапуска или блокировки.
Nginx	Static Pod	Внутренний балансировщик нагрузки для обеспечения отказоустойчивого доступа к серверам Kubernetes API.
Secrets Store CSI	Kubernetes Deployment	Компонент Nova Container Platform, который позволяет переносить из StarVault ключи, секреты или сертификаты в кластер Kubernetes, сохранять их в объектах `Secret` или `ConfigMap` и монтировать в `Pod` в виде тома.
StarVault CSI	Kubernetes Deployment	Компонент кластера, который позволяет использовать StarVault в качестве провайдера секретов для Secrets Store CSI.

В Nova Container Platform поддерживается использование одного либо трех мастер-узлов. Для эксплуатации Nova Container Platform в продуктивных средах рекомендуется использовать три мастер-узла.

2.2. Инфраструктурные узлы

Отдельные инфраструктурные узлы в кластере Kubernetes предназначены для размещения платформенных сервисов Nova Container Platform, а также служебных сервисов пользователя. Это позволяет обеспечить работоспособность высоконагруженных сервисов без влияния на пользовательские сервисы и сервисы Control Plane.

Поскольку для платформенных и служебных сервисов может требоваться постоянное хранилище, на инфраструктурных узлах доступен компонент Local Path CSI, который обеспечивает работу *Persistent Volumes* в Kubernetes.

Данные томов (*Persistent Volumes*), предоставленных с помощью Local Path CSI на инфраструктурных узлах, хранятся на узлах локально и не защищаются репликацией. При размещении собственных служебных сервисов на инфраструктурных узлах убедитесь, что ваше ПО (сервис) поддерживает репликацию данных встроенными средствами.

Кроме этого, инфраструктурные узлы содержат отдельный балансировщик входящего трафика (сервис-прокси) Ingress Nginx, доступный через объект *IngressClass* `nginx-internal`. Данный подход позволяет полностью разделить потоки продуктивного и служебного трафика на уровне Ingress-контроллеров.

В кластере Kubernetes инфраструктурные узлы имеют специальную метку `node-role.kubernetes.io/infra`. Для ограничения запуска произвольных сервисов на данных узлах также установлены параметры *Taints* с значением `node-role.kubernetes.io/infra:NoSchedule`.

Большинство компонентов базового модуля Nova Container Platform, а также все дополнительные модули размещаются на инфраструктурных узлах платформы. Веб-интерфейсы платформенных сервисов доступны только через объект *IngressClass* `nginx-internal`.

В Nova Container Platform поддерживается использование одного и более инфраструктурных узлов. Для эксплуатации в продуктивных средах рекомендуется использовать три инфраструктурных узла. Платформенные сервисы в Nova Container Platform развертываются в кластерном (высокодоступном) режиме, когда используется три и более инфраструктурных узла. Дальнейшее увеличение количества узлов не влияет на конфигурацию платформенных сервисов.

2.3. Узлы балансировки

В Nova Container Platform опционально могут быть использованы отдельные узлы для балансировки входящего трафика. На данных узлах размещаются только компоненты контроллера Ingress Nginx, который доступен через объект *IngressClass* `nginx-public`.

Вы можете использовать данные узлы в случаях:

- Когда вам необходимо вынести узлы, принимающие трафик от внешних клиентов, в отдельный сегмент (DMZ).
- Когда вы хотите повысить устойчивость инфраструктуры к каким-либо воздействиям, распределив обработку трафика и работу собственных сервисов на разные узлы.

В кластере Kubernetes узлы балансировки имеют специальную метку `node-role.kubernetes.io/ingress`. Для ограничения запуска произвольных сервисов на данных узлах также установлены параметры *Taints* с значением `node-role.kubernetes.io/ingress:NoSchedule`.

В кластерах Kubernetes с использованием трех узлов либо установленных без узлов с ролью `ingress`, роль таких балансировщиков принимают рабочие узлы для пользовательских нагрузок. Таким образом, независимо от размера кластера, продуктивный трафик остается всегда отделенным от служебного.

2.4. Рабочие узлы

Рабочие узлы предназначены для размещения пользовательских нагрузок. Данные узлы не имеют ограничительных меток и допускают запуск любых сервисов и приложений.

В кластере Kubernetes рабочие узлы имеют специальную метку `node-role.kubernetes.io/worker`. В кластерах с использованием трех узлов либо установленных без выделенных узлов с ролью `ingress`, рабочие узлы имеют две метки `node-role.kubernetes.io/worker` и `node-role.kubernetes.io/ingress`.

3. Доступные операции

Администратор платформы может выполнять различные операции с узлами кластера Kubernetes:

- Получить список узлов кластера Kubernetes.
- Добавить или обновить собственные метки узлов для контроля размещения Pod в кластере.
- Перевести узел в режим обслуживания с целью его выключения, перезапуска или освобождения от нагрузки.
- Добавить новые узлы в кластер Kubernetes при необходимости горизонтального масштабирования кластера Kubernetes.
- Удалить узлы из кластера Kubernetes.
- Добавить или удалить вычислительные ресурсы для узла кластера Kubernetes.

Резервирование ресурсов

1. Общая информация

Начиная с версии 5.2.0, **Nova Container Platform** поддерживает автоматический расчет и выделение ресурсов, резервируемых под системные нужды. Это обеспечивает более стабильную и предсказуемую работу кластера при различных нагрузках. Механизм резервирования применяется как для **мастер-узлов**, так и для **рабочих узлов**.

2. Резервирование CPU

Расчет резервирования выполняется по формулам:

- **Мастер-узлы**: *резерв = базовое значение + инкремент * (cpus - 1) + дополнительное резервирование*
- **Рабочие узлы**: *резерв = базовое значение + инкремент * (cpus - 1)*

- **Базовое значение** резервируемого объема CPU — 60 миллиядер (по умолчанию);
- **Инкремент** — дополнительный объем для каждого ядра, начиная со второго (по умолчанию: 12 миллиядер);
- **Дополнительное резервирование** — выделяется только на мастер-узлах для корректной работы управляющих компонентов (по умолчанию: 1300 миллиядер; например, etcd, kube-apiserver, controller-manager и др.).

3. Резервирование памяти

Объем памяти, резервируемый для системных нужд, рассчитывается по следующей схеме:

- 50% от первых 4 ГБ;
- минус 20% от объема между 4 и 8 ГБ;
- минус 10% от объема между 8 и 16 ГБ;
- минус 6% от объема между 16 и 128 ГБ;
- минус 2% от общего объема, если памяти больше 128 ГБ.

4. Пересчет резервов при изменении ресурсов

Если после установки кластера объем ресурсов на узле был увеличен (например, через интерфейс провайдера или вручную), **резервирование будет пересчитано автоматически при следующей перезагрузке узла**.

Переинициализация или ручной перезапуск системных сервисов не требуется.

5. Значения для конфигураций по умолчанию

Ниже приведены значения резервируемых ресурсов, которые используются системой в базовой конфигурации (без дополнительной нагрузки):

▸ **Мастер-узлы**

▸ **Инфраструктурные узлы**