Непрерывное развертывание и доставка компонентов

Подходы IaaC и GitOps для управления конфигурациями в Nova Container Platform применяются и к ресурсам Kubernetes, входящим в состав модулей платформы. Непрерывная доставка модулей осуществляется с помощью службы (системы) FluxCD.

1. GitOps в Nova Container Platform

GitOps является одним из способов управления инфраструктурой или приложениями в инфраструктуре, когда вся их конфигурация декларативно описана, а версия конфигураций хранится и контролируется в Git-репозитории. Развертывание данной инфраструктуры является автоматизированным процессом, при котором состояние приложений в инфраструктуре приводится к состоянию, описанному в конфигурациях Git-репозитория.

Nova Container Platform использует основные принципы GitOps в управлении модулями платформы, а именно:

- Хранит всю конфигурацию модулей в локальном Git-репозитории Gitea.
- Git-репозиторий содержит необходимые ветки и теги, соответствующие версиям платформы.
- Git-репозиторий неизменяемый и поддерживает только одностороннюю синхронизацию.
- Служба FluxCD постоянно проверяет Git-репозиторий на наличие новых версий конфигураций.
- Служба FluxCD постоянно отслеживает изменения компонентов в кластере Kubernetes и поддерживает их состояние в соответствии с конфигурацией в Git-репозитории.

2. Репозитории

Репозиторий является единым источником всех конфигураций, которые описывают желаемое состояние объектов в Kubernetes. В системе FluxCD поддерживается несколько типов репозиториев:

- `GitRepository` : Git-репозитории, в которых содержатся конфигурации и манифесты для развертывания объектов в Kubernetes.
- `OCIRepository` : OCI-репозитории артефактов, частно используются для хранения образов контейнеров, Helm-чартов, а также пакетов.

- `HelmRepository` : Репозитории Helm-чартов для развертывания в Kubernetes.
- `Bucket` : Бакеты в S3-хранилище, в которых содержатся конфигурации и манифесты для развертывания объектов в Kubernetes.

Nova Container Platform использует Git-репозиторий (`GitRepository`), размещаемый в локальном хранилище Gitea. Gitea устанавливается в кластер Kubernetes на этапе развертывания платформы и автоматически настраивается. На схеме ниже представлен процесс взаимодействия с Git-репозиторием.

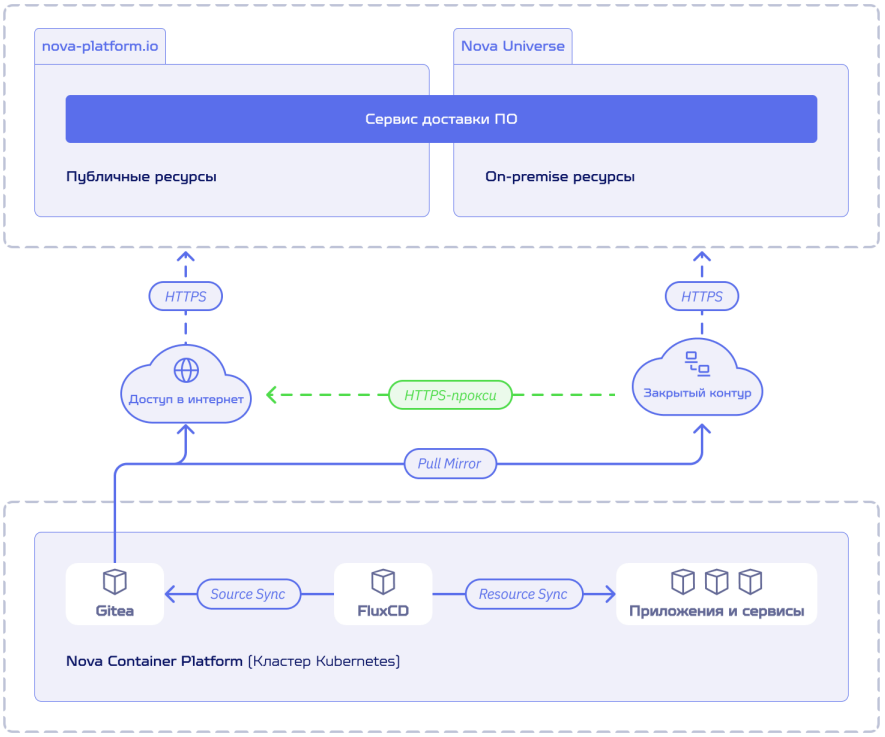

Рисунок 1. Взаимодействие с Git-репозиторием в Nova Container Platform

При каждом запуске хранилище Gitea настраивается автоматически:

- Выполняется настройка учетной записи администратора через интеграцию с StarVault.
- Выполняется настройка учетной записи для FluxCD через интеграцию с StarVault
- Выполняется настройка параметров входа в Gitea по протоколу OIDC через интеграцию с StarVault.
- Выполняется настройка внутренней организации.
- Выполняется настройка зеркалирования релизного Git-репозитория: при установке платформы через Интернет зеркалируется репозиторий из хранилища `code.nova-platform.io` , при установке платформы в закрытом контуре - из хранилища Nova Universe.

Для хранилища Gitea не требуется и не используется персистентное хранилище. Каждый перезапуск Gitea приводит к переинициализации и синхронизации релизного репозитория. Релизный репозиторий доступен в Gitea только на чтение учетной записи FluxCD.

Для подключения репозитория Gitea к FluxCD используется конфигурация `GitRepository`, в которой определяется URL Git-репозитория, версия релиза, набор устанавливаемых модулей, интервал синхронизации и параметры учетной записи.

Релизный Git-репозиторий в Gitea доступен только на чтение учетной записи FluxCD. Изменение какой-либо информации в репозитории невозможно.

FluxCD выполняет постоянную синхронизацию Git-репозитория каждые 10 минут и кеширует полученную информацию. После того, как Git-репозиторий успешно синхронизирован в FluxCD, Configuration Manager создает ресурсы `Kustomization` для каждого из устанавливаемых компонентов. В данных ресурсах описываются следующие параметры:

- Путь к манифестам в репозитории Gitea
- Зависимости от других устанавливаемых компонентов
- Параметры и переменные для генерации финального манифеста компонента
- Имя служебного аккаунта (*Service Account*) для развертывания компонента
- Перечень ресурсов для проверки доступности компонента

Далее FluxCD обрабатывает ресурсы `Kustomization` и выполняет их реконсиляцию - устанавливает компоненты и приводит их действительное состояние в описанное в Git-репозитории. По умолчанию интервал реконсиляции составляет 10 минут.

При обновлении Nova Container Platform для установки новой версии модулей выполняется переключение FluxCD на новую версию релизного репозитория.

3. Kustomize и Kustomization

Наряду с использованием ресурсов *Kustomization* в API `kustomization.kustomize.toolkit.fluxcd.io`, в Nova Container Platform также используется API `kustomization.kustomize.config.k8s.io` для управления параметрами Kustomize.

Kustomize – это инструмент нативного управления конфигурациями в Kubernetes, позволяющий настраивать простые YAML-манифесты без использования шаблонов. При этом, оригинальные YAML-манифесты остаются без изменений, а конечные YAML-манифесты генерируются в отдельный слой с использованием пользовательских параметров. Kustomize существует как отдельная утилита, но также встроен в инструментарий FluxCD.

С помощью Kustomize в Nova Container Platform решаются следующие задачи:

- Контролируется перечень конфигураций (YAML-манифестов), которые могут быть установлены для компонента модуля.
- Устанавливаются общие метки и аннотации на ресурсы Kubernetes.
- Применяются strategic merge и JSON6902 патчи для адаптации конфигурации компонента под тип и метод установки платформы.

Параметры Kustomize, как правило, указываются в YAML-манифестах, хранимых в Git-репозиториях, в то время как манифесты Kustomization являются ресурсами Kubernetes и в первую очередь обрабатываются контроллерами FluxCD. При этом некоторые из параметров Kustomize доступны в рамках спецификации Kustomization.

Вы можете получить подробную информацию о Kustomize и Kustomization по ссылкам ниже:

- Kustomization
- Kustomize

После добавления слоя *Kustomize* для YAML-манифестов какого-либо компонента добавляется еще один дополнительный *PostBuild*-слой. В данном слое выполняется замена переменных, обозначенных в YAML-манифестах на значения ключей из существующих в Kubernetes объектов *ConfigMap* или *Secret*.

В Nova Container Platform в пространстве имен `nova-gitops` хранятся объекты *ConfigMap*, задающие базовые параметры для развертывания модулей платформы. Пример одного из общих ConfigMap представлен ниже:

```yaml
apiVersion: v1
data:
  clusterId: a836c40c-1f77-4c81-a43b-3e69269442d2
  controlPlaneTopology: SingleReplica
  dnsBaseDomain: apps.mycompany.local
  imageRepository: hub.universe.mycompany.local/nova-universe
  infraNodesCount: '"1"'
  k8sDefaultDnsZone: cluster.local
  mirrorRepoType: gitea
kind: ConfigMap
metadata:
  name: nova-gitops-common-substitute-config
  namespace: nova-gitops
```

Таким образом, в Git-репозитории хранятся универсальные YAML-манифесты без чувствительной информации, которые описывают базовый сценарий установки. FluxCD “на лету” адаптирует манифесты и устанавливает их в Kubernetes, в дальнейшем постоянно поддерживая их консистентность.

4. Контроллеры FluxCD

Служба FluxCD состоит из шести основных контроллеров. Описание данных контроллеров и обслуживаемые ими ресурсы CRD представлены в таблице ниже.

Наименование	Назначение	Обслуживаемые CRD
Source Controller	Контроллер, выполняющий задачи управления артефактами из различных репозиториев.	GitRepository CRD OCIRepository CRD HelmRepository CRD HelmChart CRD Bucket CRD
Kustomize Controller	Контроллер, обеспечивающий непрерывную доставку и развертывание ресурсов в Kubernetes. В качестве источника данных использует репозитории под управлением Source Controller.	Kustomization CRD
Helm Controller	Контроллер, обеспечивающий непрерывную доставку и развертывание Helm-чартов в Kubernetes. В качестве источника данных использует репозитории под управлением Source Controller.	HelmRelease CRD
Notification Controller	Контроллер, выполняющий задачи обработки входящих и исходящих событий в FluxCD и отправки нотификаций во внешние системы.	Provider CRD Alert CRD Receiver CRD
Image Automation Controller	Контроллер, работающий в паре с Image Reflector Controller. Выполняет задачи обновления версий образов в YAML-манифестах в Git-репозитории.	ImagePolicy CRD ImageUpdateAutomation CRD
Image Reflector Controller	Контроллер, работающий в паре с Image Automation Controller. Сканирует хранилища образов контейнеров с целью поиска новых версий.	ImageRepository CRD

5. Мультитенантность

В Nova Container Platform служба FluxCD используется не только для нужд самой платформы, но также может применяться и конечными пользователями для настройки собственных процессов непрерывной доставки приложений и сервисов. FluxCD поддерживает работу множества пользователей или команд в пределах одного кластера Kubernetes путем сегментации и изоляции ресурсов на уровне пространств имен и RBAC.[1]

5.1. Использование RBAC

В Nova Container Platform контроллеры FluxCD работают в режиме Multi-tenancy lockdown, полностью опираясь на политики Kubernetes RBAC. Контроллеры вносят изменения в среду

Kubernetes (развертывают и изменяют ресурсы и приложения), имперсонируя служебный аккаунт (ServiceAccount), указанный в спецификациях *Kustomization* или *HelmRelease*.

По умолчанию, все системные объекты Kustomization запускаются от служебного аккаунта `kustomize-controller` в пространстве имен `nova-gitops`, который имеет роль `cluster-admin` в Kubernetes.

Для того, чтобы развернуть ресурсы с помощью FluxCD в других пространствах имен, необходимо иметь в данных пространствах имен отдельный служебный аккаунт и соответствующие привилегии RBAC.

5.2. Роли пользователей

Глобально при работе с FluxCD роли пользователей можно разделить на две группы:

- Администраторы платформы Nova Container Platform
- Команды (тенанты) FluxCD

5.2.1. Администраторы

Администраторы платформы, как правило, имеют неограниченный доступ к Kubernetes API. В их зоне ответственности могут находиться такие задачи как, например:

- Подключение Git, Helm, OCI репозиториев к FluxCD, в том числе репозиториев команд.
- Установка CRD в кластер Kubernetes.
- Установка и настройка дополнительных пространств имен.
- Настройка RBAC для служебных пользователей команд.

5.2.2. Команды

Пользователи команд обычно имеют ограниченный доступ к Kubernetes API, контролируемый с помощью RBAC администраторами платформы. Примеры операций, которые могут выполняться командами, представлены далее:

- Регистрация собственных репозиториев (*GitRepositories*, *HelmRepositories*, *Buckets*).
- Развертывание собственных сервисов и приложений через FluxCD с помощью *Kustomizations* и *HelmReleases* с использованием согласованного служебного аккаунта.
- Настройка автоматизации обновления приложений с помощью *ImageRepositories*, *ImagePolicies*, *ImageUpdateAutomations*.
- Настройка веб-хуков и оповещений через FluxCD (*Receivers*, *Alerts*).

1. Функционал мультитенантности доступен с версии Nova Container Platform v3.0.0 и выше.

Архитектура DNS в Nova Container Platform

Nova Container Platform предлагает несколько возможных конфигураций системы разрешения имен (DNS) в кластере Kubernetes. В данном разделе представлена детальная информация о конфигурациях и рекомендации по их выбору.

1. О DNS в Kubernetes

Система DNS является неотъемлемой частью среды Kubernetes и предназначена для управления DNS-записями объектов *Service* и *Pod*. Пользователь Kubernetes может обратиться к данным объектам, используя постоянные DNS-имена вместо внутренних IP-адресов. Для обслуживания DNS-зон и процессов Service Discovery применяется решение CoreDNS.

Kubernetes автоматически управляет системой DNS и публикует информацию об объектах *Pod* и *Service*. Компонент *Kubelet* вносит настройки DNS в *Pod*, после чего контейнеры внутри *Pod* могут разрешать запросы как в пределах кластера Kubernetes, так и за его пределами.

Всем объектам *Service* в кластере Kubernetes присваивается постоянное DNS-имя, которое разрешается либо во внутренний IP-адрес самого объекта *Service*, либо в IP-адреса объектов *Pod*, которые составляют данный сервис. По умолчанию, домены поиска каждого объекта Pod содержат свое собственное пространство имен, а также кластерный DNS-домен по умолчанию.

Подробную информацию об устройстве системы DNS в Kubernetes вы можете получить в разделе официальной документации DNS for Services and Pods.

2. О Nova DNS

В Nova Container Platform по умолчанию используется дополнительный компонент Nova DNS, который расширяет систему разрешения имен и решает ряд конфигурационных задач в различных аспектах.

В основе Nova DNS также используется решение CoreDNS. Компонент тесно интегрирован с DNS-службой Kubernetes и решает следующие задачи:

- Обслуживает DNS-зону базового DNS-домена.

- Перенаправляет запросы к записям зоны базового DNS-домена, когда зона обслуживается инфраструктурными (пользовательскими) DNS-серверами.
- Принимает запросы от инфраструктурных (пользовательских) DNS-серверов и разрешает записи зоны базового DNS-домена.

Таким образом, компонент Nova DNS позволяет использовать следующие подходы к организации системы разрешения имен:

- Полностью обслуживать DNS-зону базового DNS-домена средствами кластера Kubernetes.
- Обеспечить интеграцию с инфраструктурными (пользовательскими) DNS-серверами через организацию перенаправления запросов к базовому DNS-домену.
- Обеспечить интеграцию с инфраструктурными (пользовательскими) DNS-серверами через организацию приема запросов к базовому DNS-домену от инфраструктурных (пользовательских) DNS-серверов.

3. Зона DNS по умолчанию

DNS-зона по умолчанию (или базовый домен) необходимы в Nova Container Platform для разрешения имен платформенных сервисов. DNS-зона по умолчанию обязательна, однако ее размещение и характер взаимодействия с ней могут быть изменены пользователем на этапе установки платформы.

Перед установкой платформы пользователь указывает в конфигурационном манифесте параметр dnsBaseDomain, который определяет базовый домен и соответствующую wildcard-запись для разрешения имен платформенных сервисов.

Пример имен платформенных сервисов, доступных после установки платформы, используя параметр `dnsBaseDomain` со значением `mycompany.local`:

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

В примере, представленном выше, все имена платформенных сервисов находятся в DNS-зоне `mycompany.local`, a wildcard-запись, с помощью которой разрешаются данные имена, имеет вид `*.mycompany.local`.

DNS-имена платформенных сервисов предопределены в Nova Container Platform и не могут быть изменены.

Обслуживание DNS-зоны по умолчанию осуществляется компонентом Nova DNS. Поскольку доступ к платформенным сервисам осуществляется через балансировщики нагрузки, расположенные на инфраструктурных узлах, то запись `*.mycompany.local` должна разрешаться в IP-адреса инфраструктурных узлов платформы.

4. Режимы работы DNS

Nova Container Platform поддерживает три режима работы компонента Nova DNS.

В диаграммах, представленных далее, IP-адреса и конфигурации DNS-серверов приведены в качестве примера и могут отличаться в зависимости от вашей инфраструктуры и используемого ПО.

4.1. Внутренний режим

Внутренний режим используется, когда в пользовательской инфраструктуре полностью отсутствует какая-либо служба DNS либо доступ к ней невозможен. Если вам не требуется интеграция с вашими DNS-серверами, а обслуживание DNS-зоны достаточно осуществлять только в пределах кластера Kubernetes, то используйте данный режим.

При использовании внутреннего режима Nova DNS для доступа к веб-интерфейсам платформенных сервисов необходимо использовать статические записи *hosts* в вашей ОС.

Ниже на схеме представлен принцип работы DNS во внутреннем режиме.

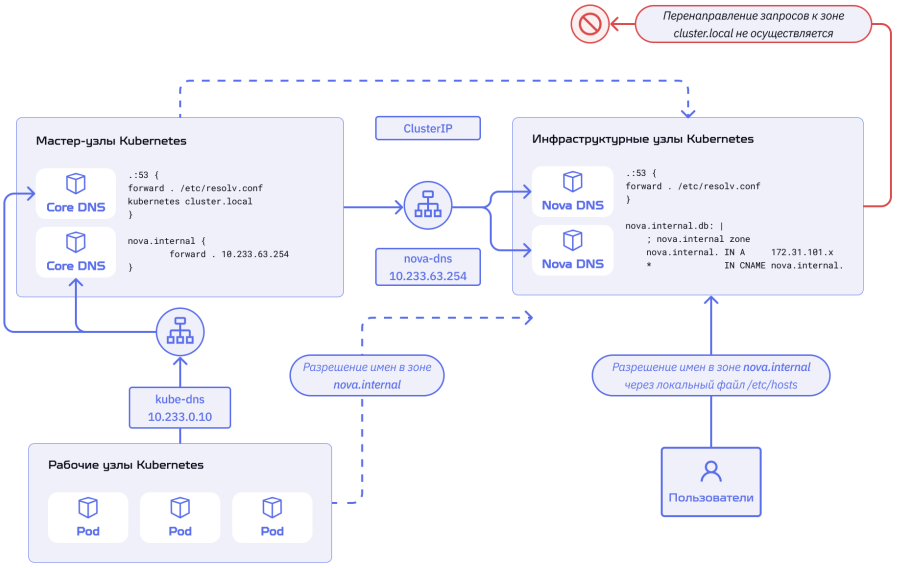

Рисунок 1. Внутренний режим работы Nova DNS в Nova Container Platform

1. Пользовательские сервисы (объекты *Pod*) настроены на использование DNS-сервера по умолчанию - `kube-dns`. Объект *Service* с типом *ClusterIP* предоставляет единый IP-адрес для балансировки запросов к объектам *CoreDNS Pod*.
2. Объекты *CoreDNS Pod* обслуживают основную зону Kubernetes `kubernetes.local`, все запросы к зоне `mycompany.local` направляют в IP-адрес сервиса `nova-dns`, а остальные запросы направляют в хостовые DNS-серверы, указанные в файле `/etc/resolv.conf`.
3. Объекты *Nova DNS Pod* принимают и обрабатывают запросы к зоне `mycompany.local`.
4. Для разрешения имен платформенных сервисов пользователи добавляют в локальный файл `/etc/hosts` необходимые записи, указывающие на IP-адреса инфраструктурных узлов.

Внутренний режим Nova DNS устанавливается по умолчанию, если блок конфигурации extraOptions не заполнен в конфигурационном манифесте.

4.2. Внешний режим

Внешний режим используется, когда DNS-зона и другие записи обслуживаются только инфраструктурными (пользовательскими) серверами DNS. Если вы планируете управлять DNS-зоной самостоятельно, то воспользуйтесь данным режимом. В данном режиме компонент Nova DNS не разворачивается в кластере Kubernetes.

Ниже на схеме представлен принцип работы DNS во внешнем режиме.

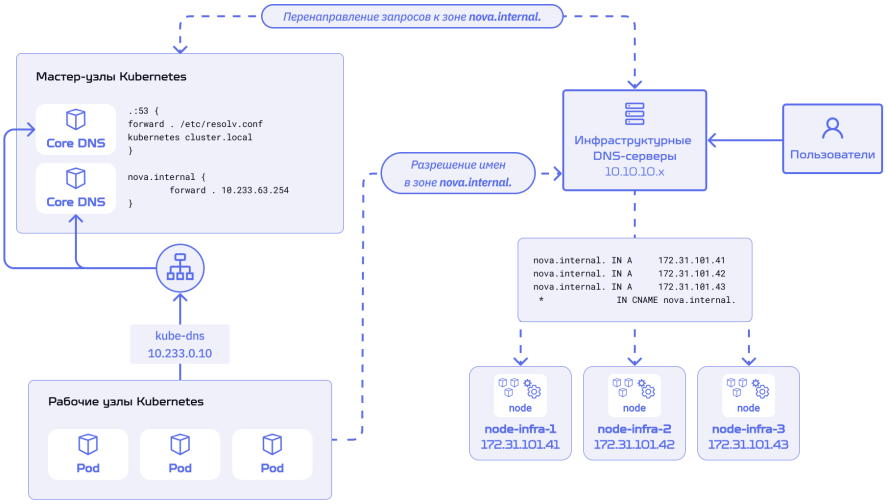

Рисунок 2. Внешний режим работы Nova DNS в Nova Container Platform

1. Пользовательские сервисы (объекты *Pod*) настроены на использование DNS-сервера по умолчанию - `kube-dns`. Объект *Service* с типом *ClusterIP* предоставляет единый IP-адрес для балансировки запросов к объектам *CoreDNS Pod*.
2. Объекты *CoreDNS Pod* обслуживают основную зону Kubernetes `kubernetes.local`, все запросы к зоне `mycompany.local` направляют в IP-адреса инфраструктурных DNS-серверов, а остальные запросы направляют в хостовые DNS-серверы, указанные в файле `/etc/resolv.conf`.
3. Для разрешения имен платформенных сервисов пользователи используют стандартные настройки DNS в собственной инфраструктуре.

Для настройки внешнего режима Nova DNS на этапе установки платформы вы можете использовать пример конфигурационного манифеста ниже.

YAML

```yaml
apiVersion: "config.nova-platform.io/v1alpha4"
kind: "Infrastructure"
metadata:
  name: "cluster"
spec:
  clusterConfiguration:
    extraOptions:
      dns:
        customerDns:
          enable: true
          forwardZones:
          - name: mycompany.local
            server: 10.0.0.1
          - name: acme.corp
            server: 10.0.0.2
```

Если в списке DNS-зон *forwardZones* присутствует базовая DNS-зона, то компонент Nova DNS в кластере не устанавливается.

Пример выше описывает конфигурацию, в результате которой системный компонент *CoreDNS* будет иметь две дополнительные зоны `mycompany.local` и `acme.corp` с перенаправлением запросов на пользовательские серверы `10.0.0.1` и `10.0.0.2` соответственно.

4.3. Гибридный режим

Гибридный режим используется, когда вам необходимо сохранить обслуживание базовой DNS-зоны компонентом Nova DNS в пределах кластера Kubernetes, но инфраструктурные (пользовательские) серверы DNS должны перенаправлять DNS-запросы на серверы Nova DNS.

Доступ к DNS-серверам Nova DNS осуществляется по стандартным портам и протоколам `tcp/53` и `udp/53` через инфраструктурные узлы. Публикация данных портов осуществляется с помощью TCP/UDP балансировки компонента Ingress Controller.

Информация

Для перехода из внутреннего режима в гибридный вам достаточно на собственных DNS-серверах настроить перенаправление запросов к DNS-зоне по умолчанию на инфраструктурные узлы Nova Container Platform. Переход из внутреннего или гибридного режима во внешний не поддерживается.

Ниже на схеме представлен принцип работы DNS в гибридном режиме.

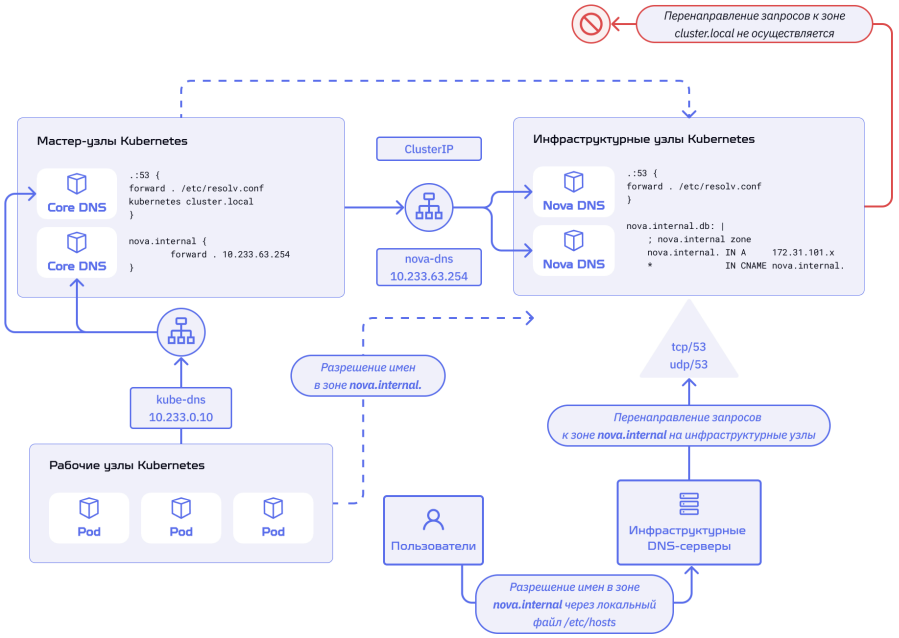

Рисунок 3. Гибридный режим работы Nova DNS в Nova Container Platform

1. Пользовательские сервисы (объекты *Pod*) настроены на использование DNS-сервера по умолчанию - `kube-dns`. Объект *Service* с типом *ClusterIP* предоставляет единый IP-адрес для балансировки запросов к объектам *CoreDNS Pod*.
2. Объекты *CoreDNS Pod* обслуживают основную зону Kubernetes `kubernetes.local`, все запросы к зоне `mycompany.local` направляют в IP-адрес сервиса `nova-dns`, а остальные запросы направляют в хостовые DNS-серверы, указанные в файле `/etc/resolv.conf`.
3. Объекты *Nova DNS Pod* принимают и обрабатывают запросы к зоне `mycompany.local`.
4. Инфраструктурные (пользовательские) DNS-серверы перенаправляют запросы пользователей к зоне `mycompany.local` на инфраструктурные узлы платформы (сервисы Nova DNS).
5. Для разрешения имен платформенных сервисов пользователи используют стандартные настройки DNS в собственной инфраструктуре.

Гибридный режим Nova DNS аналогичен внутреннему и устанавливается по умолчанию, если блок конфигурации extraOptions не заполнен в конфигурационном манифесте.

5. Дополнительные параметры DNS

Кроме выбора режимов работы DNS в Nova Container Platform вы также можете указать дополнительные параметры и DNS-имена некоторых служебных компонентов на этапе установки платформы:

- DNS-зона кластера Kubernetes
- DNS-имя сервера Kubernetes API

- Дополнительные DNS-имена и IP-адреса Kubernetes API
- Пользовательские DNS-серверы

5.1. DNS-зона кластера Kubernetes

По умолчанию, в Nova Container Platform используется DNS-зона `cluster.local`. Однако, при необходимости, вы можете изменить ее, используя параметр k8sDefaultDnsZone

5.2. Kubernetes API

По умолчанию, в Nova Container Platform не используется отдельное публичное DNS-имя для API-сервера Kubernetes. В веб-консоли Nova, а также в конфигурациях `kubeconfig` вам будет доступен IP-адрес первого мастер-узла кластера Kubernetes.

Однако, для удобства вы можете установить DNS-имя по умолчанию для Kubernetes API, используя параметр k8sAPIDefaultFqdn на этапе установки платформы.

Если вам необходимо добавить дополнительные DNS-имена и IP-адреса Kubernetes API, например, для доступа по альтернативным именам или с сетевых балансировщиков, то воспользуйтесь параметром k8sAPIAdditionalSANs.

5.3. Пользовательские серверы DNS по умолчанию

Если вам необходимо добавить список собственных DNS-серверов, которые должны использоваться по умолчанию в среде Kubernetes, то воспользуйтесь параметром servers. В данном случае служба CoreDNS будет перенаправлять все запросы именно на ваш список серверов вместо записей в хостовом файле `/etc/resolv.conf`.

Провайдеры инфраструктуры

Nova Container Platform поддерживает различные провайдеры инфраструктуры. Для взаимодействия с ними узел `nova-ctl` использует *Terraform* и его плагины (инфраструктурные провайдеры).

В плагине *Terraform* реализованы механизмы управления инфраструктурными объектами через API. Утилита `nova-ctl` имеет в составе все поддерживаемые инраструктурные провайдеры, а для их работы не требуется иметь доступ в Интернет. Все операции с инфраструктурными провайдерами выполняются прозрачно для пользователя.

Провайдеры инфраструктуры *Terraform* разрабатываются сообществом и ОРИОН для совместимости с Nova Container Platform.

1. Поддерживаемые интеграции

Вы можете получить актуальный перечень поддерживаемых провайдеров инфраструктуры в разделе Перечень матриц совместимости и протестированных интеграций.

2. Использование в Nova Container Platform

Перед установкой платформы пользователь может получить предзаполненные установочные манифесты (шаблоны конфигураций) с помощью `nova-ctl init`. Далее nova-ctl в интерактивном режиме запрашивает у пользователя тип инфраструктуры, в которой планируется установка, и предоставляет шаблон с блоком конфигурации необходимого провайдера инфраструктуры.

Заполнение установочного манифеста выполняется строго в соответствии с параметрами объекта *Infrastructure* в API-группе *config.nova-platform.io*. `nova-ctl` обрабатывает полученные из манифеста данные и автоматически подготавливает конфигурационные файлы Terraform. Пользователь также дополнительно уведомляется об изменениях, которые будут внесены в инфраструктуру, например, создание необходимого количества ВМ, дисков, сетевых интерфейсов и т.п.

Процесс использования провайдеров инфраструктуры в Nova Container Platform на этапе развертывания представлен на схеме ниже.

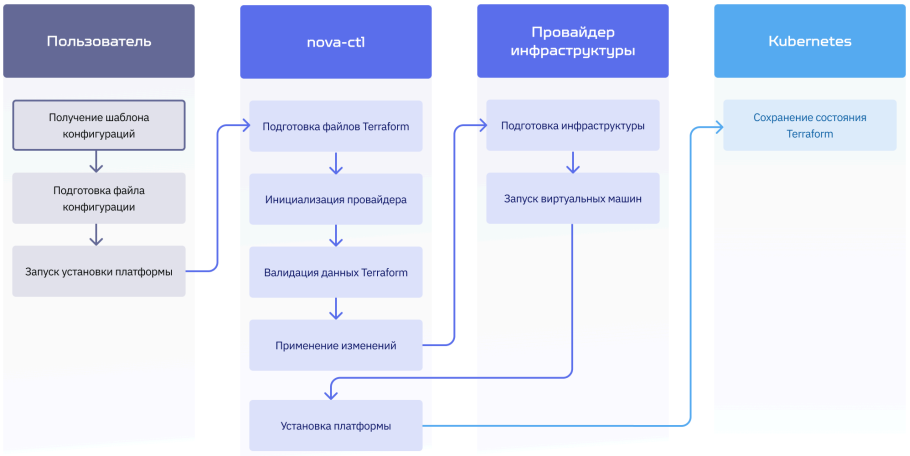

Рисунок 1. Процесс использования провайдеров инфраструктуры в Nova Container Platform

До инициализации кластера Kubernetes `nova-ctl` хранит состояние объектов *Terraform* локально, а после его инициализации сохраняет состояние в кластер Kubernetes в объект *ConfigMap*. При масштабировании кластера Kubernetes `nova-ctl` также работает с блокировками Terraform, которые управляются автоматически объектом *TerraformLock* в API-группе *config.nova-platform.io* в Kubernetes.

Схема, представленная ниже, демонстрирует процесс масштабирования кластера в контексте взаимодействия `nova-ctl` и пользователя с кластером Kubernetes.

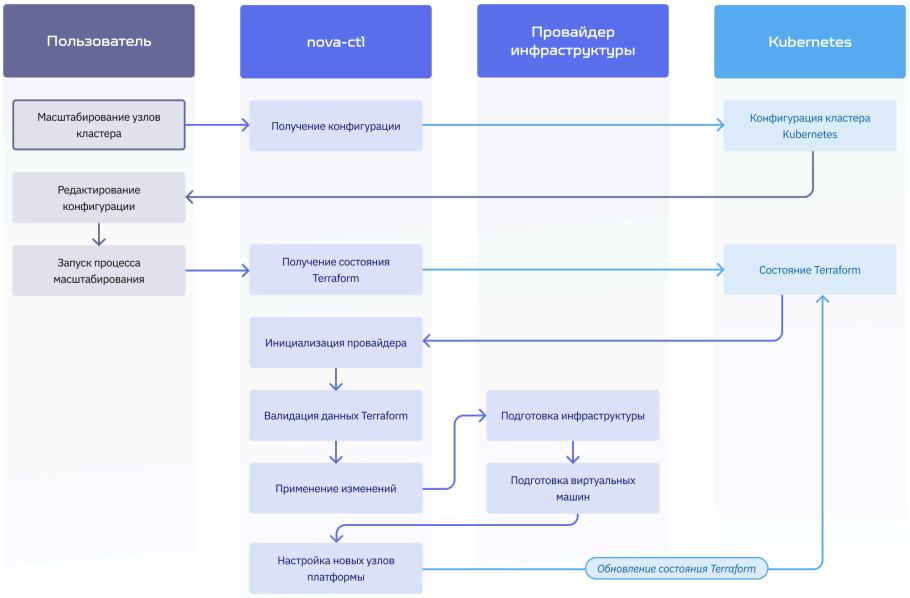

Рисунок 2. Процесс масштабирования кластера Nova Container Platform