Управление системными конфигурациями

Для управления конфигурациями узлов в Nova Container Platform SE используется специальный сервис настройки ПО Nova Configuration Manager. В данном разделе представлена информация о назначении данного сервиса, задачах, которые он решает, а также о его архитектуре.

1. Configuration Manager

Nova Configuration Manager является одним из критически важных компонентов Nova Container Platform SE и реализует механизмы автоматизированной конфигурации узлов платформы, управления узлами, установки и настройки необходимого ПО. В основе Configuration Manager используются компоненты платформы управления конфигурациями Puppet, однако решения для классификации и управления узлами платформы уникальны для Nova Container Platform SE. В сервере управления Nova Universe используется упрощенная версия Configuration Manager, адаптированная для работы в закрытом сетевом окружении.

1.1. Описание архитектуры

1.1.1. Используемые практики и подходы

В Configuration Manager применяются подходы IaaC и GitOps для управления конфигурациями Nova Container Platform SE. Для подготовки конечных кластеров используется инфраструктурный код, который содержит различные сценарии развертывания и обновления платформы, установку необходимых пакетов, среды Kubernetes, а также различных платформенных сервисов. Инфраструктурный код декларативен, определяет конечное желаемое состояние всех компонентов, а процесс его выполнения идемпотентен, то есть может выполняться множество раз с одним и тем же результатом.

1.1.2. Компоненты и роли

Configuration Manager имеет клиент-серверную архитектуру, где выполняет роль сервера. В роли клиента выступает узел Nova Container Platform SE, на котором установлен хостовый агент Nova Host Agent. Сервер синхронизирует инфраструктурный код необходимой версии с внешним Git-репозиторием, подготавливает уникальные сценарии для узла платформы, а агент выполняет данные сценарии локально, транслируя инфраструктурный код в необходимые команды. Результаты выполнения сценария передаются агентом на сервер в виде отчета.

На диаграмме ниже представлена схема работы Configuration Manager.

Рисунок 1. Схема работы Configuration Manager

1. При каждом запуске Nova Host Agent отправляет серверу Configuration Manager факты о себе (конфигурацию ОС, сетевые параметры, список доступных устройств и т.п.).
2. В ответ сервер Configuration Manager направляет агенту сценарий настройки (каталог), который агент должен выполнить, чтобы привести состояние узла к желаемому. При этом агент пропускает те задачи, которые уже были выполнены ранее.
3. После выполнения сценария настройки агент направляет серверу отчет, который содержит информацию о результатах выполнения операций и их статус. Данный отчет может использоваться для диагностики проблем, возникших в ходе операций с платформой.

> Отдельный веб-интерфейс для просмотра отчетов Configuration Manager на текущий момент доступен только при установке платформы в закрытом сетевом контуре с помощью сервера Nova Universe.

Каждый узел Nova Container Platform SE получает свой уникальный сценарий настройки и выполняет его параллельно с другими узлами. За счет этого достигается высокая скорость установки Nova Container Platform SE, которая практически не зависит от количества узлов в кластере Kubernetes.

1.2. Особенности использования

Работа Configuration Manager в Nova Container Platform SE выполняется прозрачно для пользователя и не требует погружения в сценарии установки. Пользователь использует только утилиту `nova-ctl`, которая взаимодействует напрямую с Configuration Manager и узлами платформы.

Лог работы Configuration Manager и Host Agent доступны пользователю при установке Nova Container Platform SE и в других сервисных операциях, выполняемых с помощью `nova-ctl`.

Недоступность Configuration Manager для `nova-ctl` и узлов платформы приводит к невозможности выполнения сервисных операций. При этом, сами кластера Kubernetes и сервисы, развернутые в них, продолжают работать и поддерживают все операции.

Configuration Manager хранит важную для платформы информацию: состояния узлов, результаты сценариев, которые на них выполнялись, параметры взаимодействия узлов с Configuration Manager. Потеря данной информации может привести к невозможности выполнения операций с имеющимся кластером Nova Container Platform SE.

1.3. Управляемые компоненты в Kubernetes

Кроме настройки узлов и кластера Kubernetes, Configuration Manager также устанавливает в Kubernetes определенный набор системных компонентов, необходимых для инициализации базового функционала платформы. Перечень основных компонентов представлен далее в таблице.

Компонент	Порядок управления	Владелец
Secrets Webhook	Первоначально устанавливается в кластер через Configuration Manager. В дальнейшем им контролируется и обслуживается.	Configuration Manager
Secrets Store CSI	Первоначально устанавливается в кластер через Configuration Manager. В дальнейшем им контролируется и обслуживается.	Configuration Manager
FluxCD	Первоначально устанавливается в кластер через Configuration Manager. В дальнейшем им контролируется и обслуживается.	Configuration Manager
Gitea	Первоначально устанавливается в кластер через Configuration Manager. Далее передается под управление службе непрерывной доставки FluxCD.	FluxCD

1.4. Взаимодействие со службой непрерывной доставки FluxCD

Для запуска процесса непрерывной доставки модулей в кластер Kubernetes Configuration Manager настраивает и устанавливает следующие ресурсы FluxCD:

- Ресурс *GitRepository*, описывающий параметры подключения к хранилищу Gitea, версию и набор устанавливаемых модулей.
- Ресурсы *Kustomization*, описывающие такие параметры, как:
    - Путь к манифестам компонента 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

После установки данных ресурсов Configuration Manager ожидает успешное завершение установки модулей платформы и завершает работу.

Непрерывное развертывание и доставка

Подходы IaaC и GitOps для управления конфигурациями в Nova Container Platform SE 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 SE

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 SE 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 SE 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.

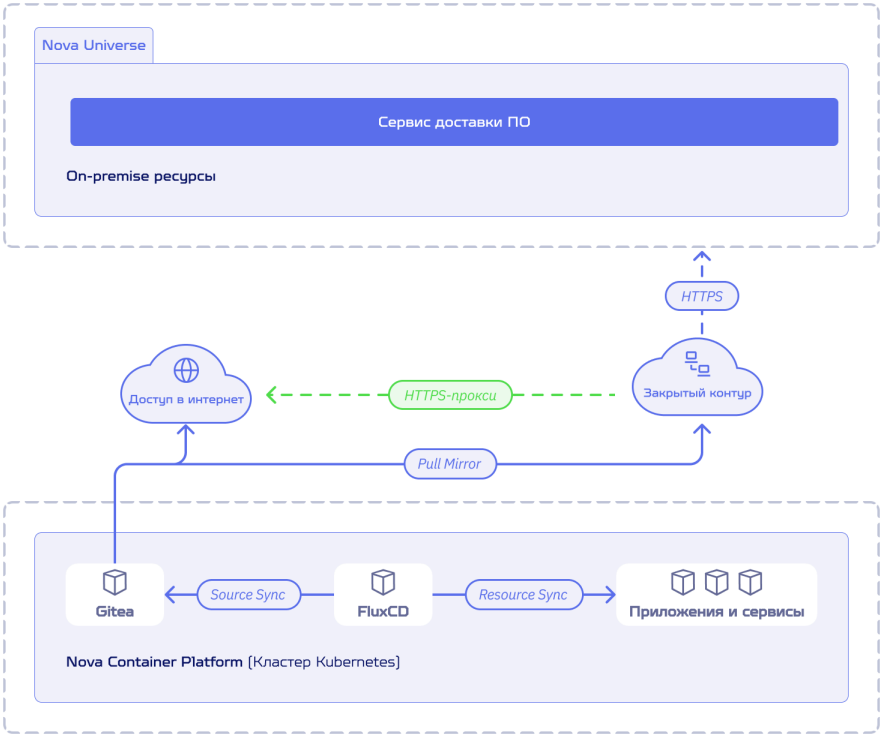

Рисунок 1. Взаимодействие с Git-репозиторием в Nova Container Platform SE

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

YAML

```yaml
apiVersion: v1
data:
  clusterId: a836c40c-1f77-4c81-a43b-3e69269442d2
  controlPlaneTopology: SingleReplica
  dnsBaseDomain: apps.nova.internal
  imageRepository: hub.universe.nova.internal/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 SE 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 SE 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 SE v3.0.0 и выше.