Общество с ограниченной ответственностью «Зетра»

ОГРН 1237700278155 ИНН 7707488152 КПП 770701001
Юр. адрес: 127473, г. Москва, вн.тер.г. муниципальный округ Тверской, пер 1-й Волконский, д. 15, помещение 1/3
тел.: +7 (909) 909-17-73,
e-mail: Zetrasoft@mail.ru

Общее руководство пользования программным обеспечением «Система оркестровки контейнеризированных приложений ZETRAKUBER» («ZETRAKUBER»)

Москва
2023 г.

Наименование

Полное наименование системы: «Система оркестровки контейнеризированных приложений ZETRAKUBER».

Сокращенное наименование системы: «*ZETRAKUBER*» (или Система).

Назначение системы

ZETRAKUBER - это программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями, с расширенными функциями безопасности.

Областью применения Системы является автоматизация развёртывание, масштабирование и координация контейнеризированных приложений в условиях кластера. Системой поддерживаются основные технологий контейнеризации, включая Docker, rkt, а также поддержка технологий аппаратной виртуализации.

ZETRAKUBER реализован на базе программного обеспечения с открытым исходным кодом Kubernetes. ZETRAKUBER обладает следующими основными преимуществами по отношению к стандартному Kubernetes, а именно:

- поддержка отечественных стандартов шифрования (ГОСТ 34.10-2018, ГОСТ 34.11-2018, ГОСТ Р 34.12-2015, ГОСТ Р 34.13-2015).

- повышенный уровень надежности (за счет исправления программных ошибок и оптимизации открытого исходного кода);

ZETRAKUBER включает в себя существенно переработанные компоненты Kubernetes, такие как:

- kubeadm
- kubectl
- kubelet
- kube-apiserver
- kube-controller-manager
- kube-proxy
- kube-scheduler

Вышеперечисленные компоненты поддерживают отечественные стандарты шифрования поддержка отечественных стандартов шифрования (ГОСТ 34.10-2018 - электронная цифровая подпись, ГОСТ 34.11-2018 - хеш-функция, ГОСТ Р 34.12-2015 — симметричные блочные шифры, ГОСТ Р 34.13-

2015 — режимы применения симметричных блочных шифров) при установлении соединения и последующим взаимодействии между собой.

Основные функции Системы

Защищенное соединение между компонентами Системы по отечественным стандартам шифрования

В Системе реализованы отечественные ГОСТы шифрования ГОСТ 34.10-2018. Информационная технология. Криптографическая защита информации.; ГОСТ 34.11-2018 «Информационная технология. Криптографическая защита информации. Функция хэширования», ГОСТ Р 34.12-2015 "Информационная технология. Криптографическая защита информации. Блочные шифры", ГОСТ Р 34.13-2015". Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров".

Мониторинг сервисов и распределение нагрузки

Система может обнаружить контейнер, используя имя DNS или собственный IP-адрес. Если трафик в контейнере высокий, Система может сбалансировать нагрузку и распределить сетевой трафик, чтобы развертывание было стабильным.

Оркестрация хранилища

Система позволяет вам автоматически смонтировать систему хранения по вашему выбору, такую как локальное хранилище, провайдеры общедоступного облака и многое другое.

Автоматическое развертывание и откаты

Используя Система можно описать желаемое состояние развернутых контейнеров и изменить фактическое состояние на желаемое. Например, вы можете автоматизировать Систему на создание новых контейнеров для развертывания, удаления существующих контейнеров и распределения всех их ресурсов в новый контейнер.

Автоматическое распределение нагрузки

Вы предоставляете Системе кластер узлов, который она может использовать для запуска контейнерных задач. Вы указываете Системе, сколько ЦП и памяти (ОЗУ) требуется каждому контейнеру. Система может разместить контейнеры на ваших узлах так, чтобы наиболее эффективно использовать ресурсы.

Самоконтроль

Система перезапускает отказавшие контейнеры, заменяет и завершает работу контейнеров, которые не проходят определенную пользователем проверку работоспособности, и не показывает их клиентам, пока они не будут готовы к обслуживанию.

Управление конфиденциальной информацией и конфигурацией

Система может хранить и управлять конфиденциальной информацией, такой как пароли, OAuth-токены и ключи SSH. Вы можете развертывать и обновлять конфиденциальную информацию и конфигурацию приложения без изменений образов контейнеров и не раскрывая конфиденциальную информацию в конфигурации стека.

Архитектура решения

Архитектура Системы реализована как кластерная архитектура, включающая в себя следующие функциональные компоненты:

1. Узлы.
2. Связь между плоскостью управления и узлом.
3. Контроллеры.
4. Диспетчер облачных контроллеров.
5. Container Runtime Interface (CRI).
6. Сборщик мусора.

Компоненты Системы

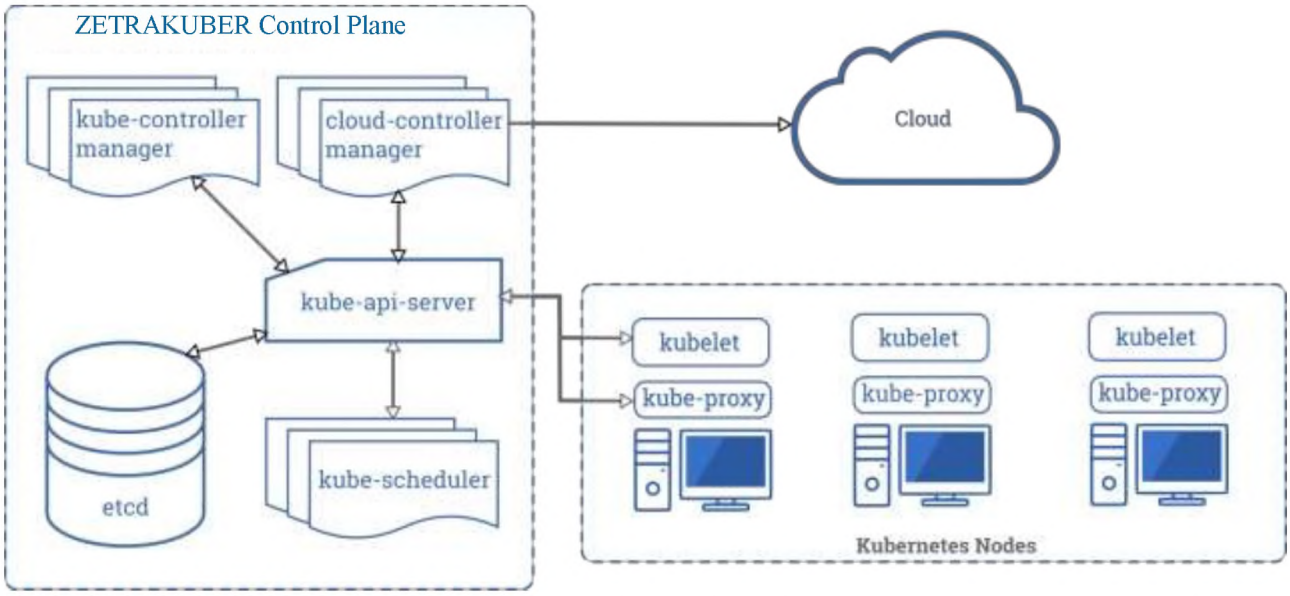
При развёртывании Системы основная работа выполняется с кластером.

Кластер состоит из набор машин, так называемые узлы, которые запускают контейнеризированные приложения. Кластер имеет как минимум один рабочий узел.

В рабочих узлах размещены поды, являющиеся компонентами приложения. Плоскость управления управляет рабочими узлами и подами в кластере. В промышленных средах плоскость управления обычно запускается на нескольких компьютерах, а кластер, как правило, развёртывается на нескольких узлах, гарантируя отказоустойчивость и высокую надёжность.

На этой странице в общих чертах описывается различные компоненты, необходимые для работы кластера Системы.

Ниже показана диаграмма кластера Системы со всеми связанными компонентами.



Плоскость управления компонентами

Компоненты панели управления отвечают за основные операции кластера (например, планирование), а также обрабатывают события кластера (например, запускают новый «под», когда поле replicas развертывания не соответствует требуемому количеству реплик).

Компоненты панели управления могут быть запущены на любой машине в кластере. Однако для простоты сценарии настройки обычно запускают все компоненты панели управления на одном компьютере и в то же время не позволяют запускать пользовательские контейнеры на этом компьютере.

kube-apiserver

Сервер API — компонент Системы панели управления, который представляет API ZETRAKUBER. API-сервер — это клиентская часть панели управления ZETRAKUBER. Основной реализацией API-сервера ZETRAKUBER является kube-apiserver. kube-apiserver предназначен для горизонтального масштабирования, то есть развёртывание на несколько экземпляров. Вы можете запустить несколько экземпляров kube-apiserver и сбалансировать трафик между этими экземплярами.

etcd

Распределённое и высоконадёжное хранилище данных в формате "ключ-значение", которое используется как основное хранилище всех данных кластера в ZETRAKUBER.

kube-scheduler

Компонент плоскости управления, который отслеживает созданные поды без привязанного узла и выбирает узел, на котором они должны работать.

При планировании развёртывания подов на узлах учитываются множество факторов, включая требования к ресурсам, ограничения, связанные с аппаратными/программными политиками, принадлежности (affinity) и непринадлежности (anti-affinity) узлов/подов, местонахождения данных, предельных сроков.

kube-controller-manager

Компонент Control Plane запускает процессы контроллера.

Каждый контроллер в свою очередь представляет собой отдельный процесс, и для упрощения все такие процессы скомпилированы в один двоичный файл и выполняются в одном процессе.

Эти контроллеры включают:

1. Контроллер узла (Node Controller): уведомляет и реагирует на сбои узла.
2. Контроллер репликации (Replication Controller): поддерживает правильное количество подов для каждого объекта контроллера репликации в системе.
3. Контроллер конечных точек (Endpoints Controller): заполняет объект конечных точек (Endpoints), то есть связывает сервисы (Services) и поды (Pods).
4. Контроллеры учетных записей и токенов (Account & Token Controllers): создают стандартные учетные записи и токены доступа API для новых пространств имен.

cloud-controller-manager

Cloud-controller-manager запускает контроллеры, которые взаимодействуют с основными облачными провайдерами.

Cloud-controller-manager запускает только циклы контроллера, относящиеся к облачному провайдеру. Вам нужно отключить эти циклы контроллера в kube-controller-manager. Вы можете отключить циклы контроллера, установив флаг —cloud-provider со значением external при запуске kube-controller-manager.

С помощью cloud-controller-manager код как облачных провайдеров, так и самого ZETRAKUBER может разрабатываться независимо друг от друга.

Следующие контроллеры зависят от облачных провайдеров:

1. Контроллер узла (Node Controller): проверяет облачный провайдер, чтобы определить, был ли удален узел в облаке после того, как он перестал работать
2. Контроллер маршрутов (Route Controller): настраивает маршруты в основной инфраструктуре облака
3. Контроллер сервисов (Service Controller): создаёт, обновляет и удаляет балансировщики нагрузки облачного провайдера.
4. Контроллер тома (Volume Controller): создаёт, присоединяет и монтирует тома, а также взаимодействует с облачным провайдером для оркестрации томов.

Компоненты узла

Компоненты узла работают на каждом узле, поддерживая работу подов и среды выполнения ZETRAKUBER.

kubelet

Агент, работающий на каждом узле в кластере. Он следит за тем, чтобы контейнеры были запущены в поде.

Утилита kubelet принимает набор PodSpecs, и гарантирует работоспособность и исправность определённых в них контейнеров. Агент kubelet не отвечает за контейнеры, не созданные Системой.

kube-proxy

Kube-proxy — сетевой прокси, работающий на каждом узле в кластере, и реализующий часть концепции сервис. kube-proxy конфигурирует правила сети на узлах. При помощи них разрешаются сетевые подключения к вашими подам изнутри и снаружи кластера. kube-proxy использует уровень фильтрации пакетов в операционной системы, если он доступен. В противном случае, kube-proxy сам обрабатывает передачу сетевого трафика.

Среда выполнения контейнера

Иполняемая среда контейнера — это программа, предназначенная для запуска контейнера в ZETRAKUBER. ZETRAKUBER поддерживает различные среды для запуска контейнеров: Docker, containerd, CRI-O, и любые реализации ZETRAKUBER CRI (Container Runtime Interface).

Дополнения

Дополнения используют ресурсы ZETRAKUBER (DaemonSet, Deployment и т.д.) для расширения функциональности кластера. Поскольку дополнения охватывают весь кластер, ресурсы относятся к пространству имен kube-system.

DNS

Хотя прочие дополнения не являются строго обязательными, однако при этом у всех ZETRAKUBER-кластеров должен быть кластерный DNS, так как многие примеры предполагают его наличие. Кластерный DNS — это DNS-сервер наряду с другими DNS-серверами в вашем окружении, который обновляет DNS-записи для сервисов ZETRAKUBER. Контейнеры, запущенные посредством ZETRAKUBER, автоматически включают этот DNS-сервер в свои DNS.

Веб-интерфейс (Dashboard)

Dashboard — это универсальный веб-интерфейс для кластеров ZETRAKUBER. С помощью этой панели, пользователи могут управлять и устранять неполадки кластера и приложений, работающих в кластере.

Мониторинг ресурсов контейнера

Мониторинг ресурсов контейнера записывает общие метрики о контейнерах в виде временных рядов в центральной базе данных и предлагает пользовательский интерфейс для просмотра этих данных.

Логирование кластера

Механизм логирования кластера отвечает за сохранение логов контейнера в централизованном хранилище логов с возможностью их поиска/просмотра.