Установка KUMA с отказоустойчивым ядром

Отказоустойчивость KUMA обеспечивается путем внедрения ядра KUMA в кластер Kubernetes, развернутый установщиком KUMA. В качестве распределённого блочного хранилища для кластера используется Longhorn. Схема:

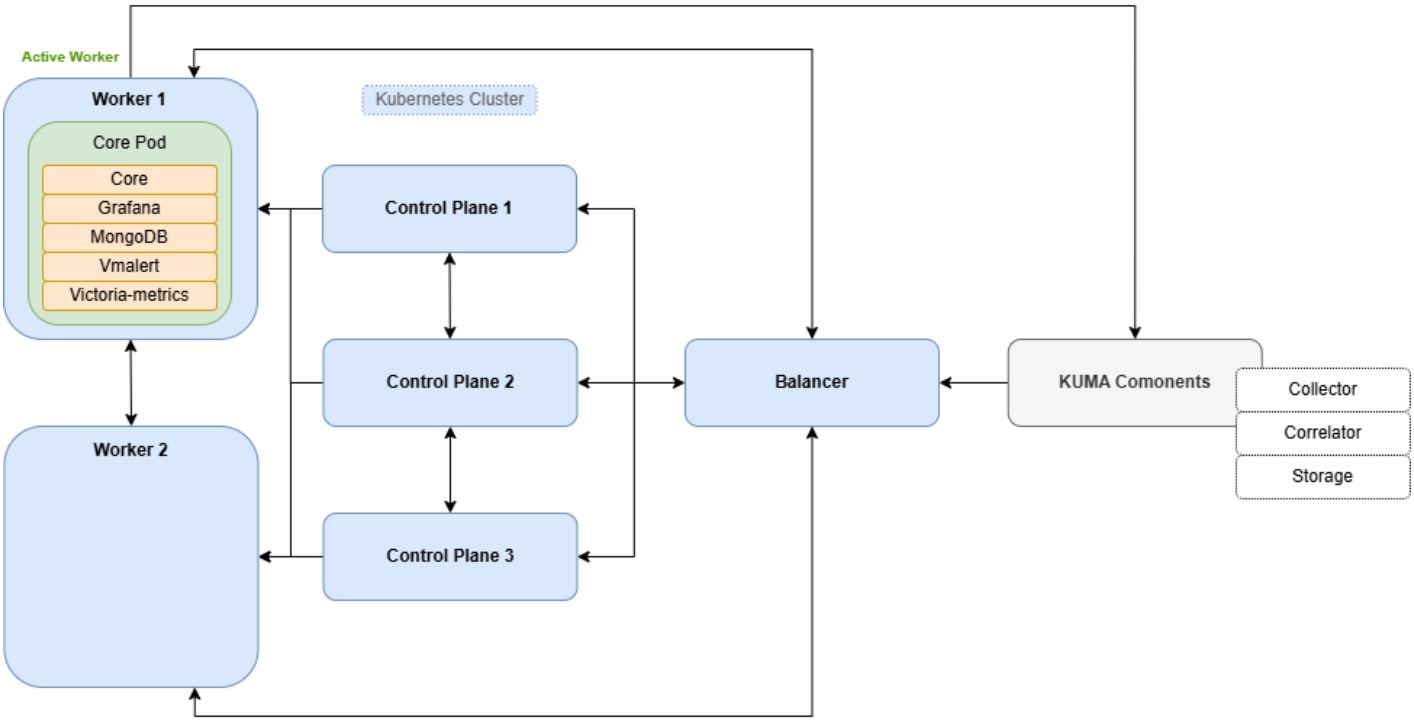
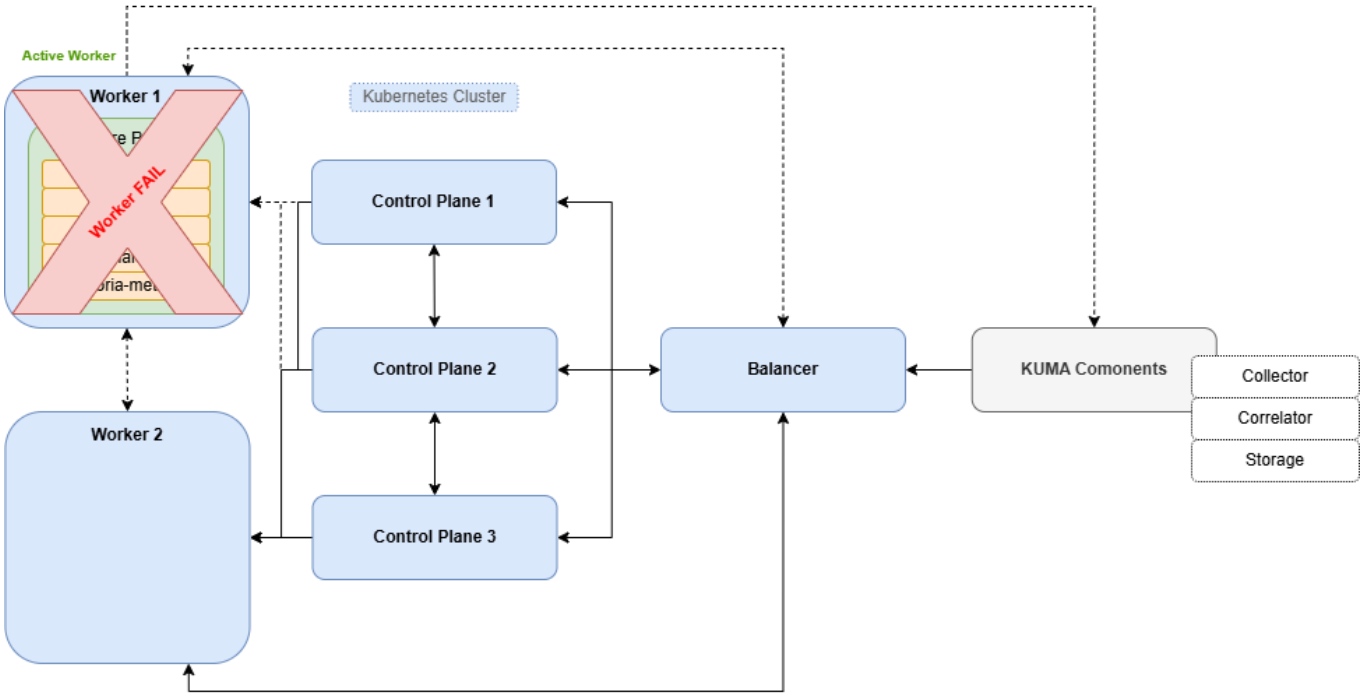
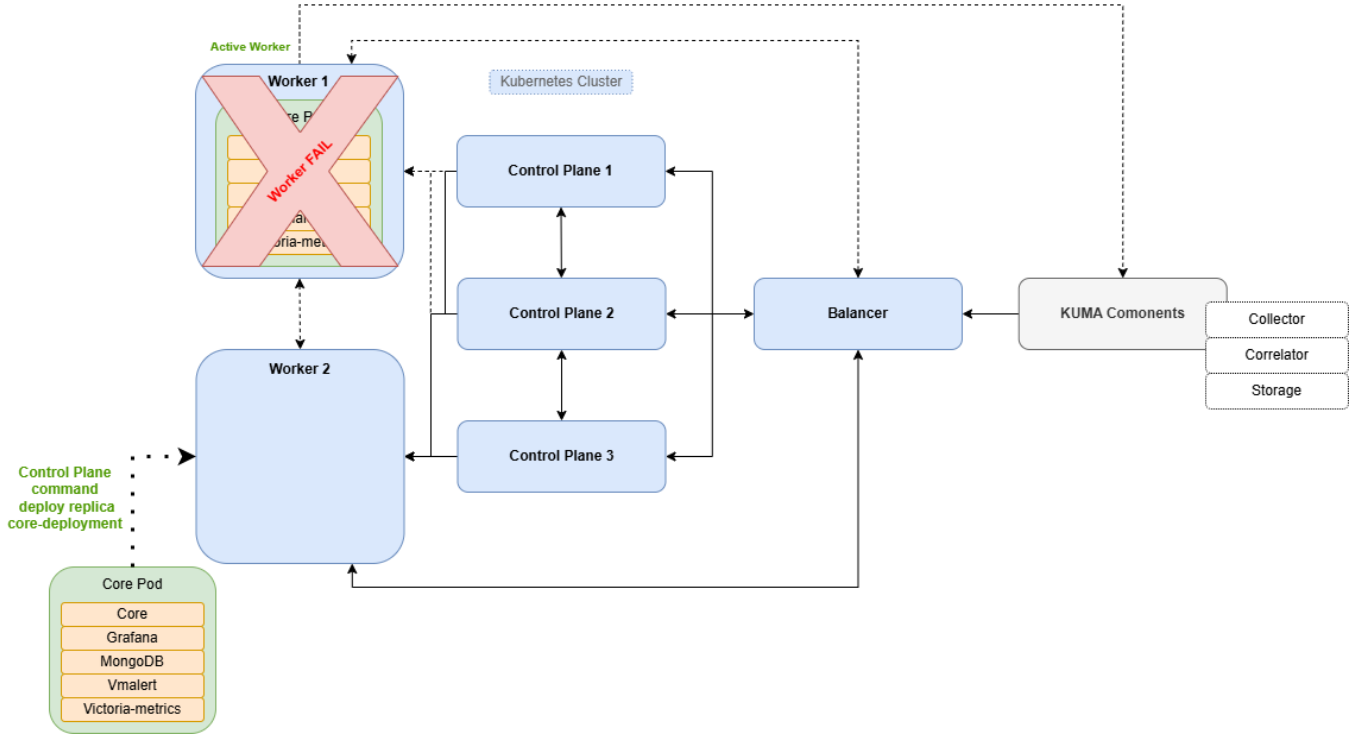


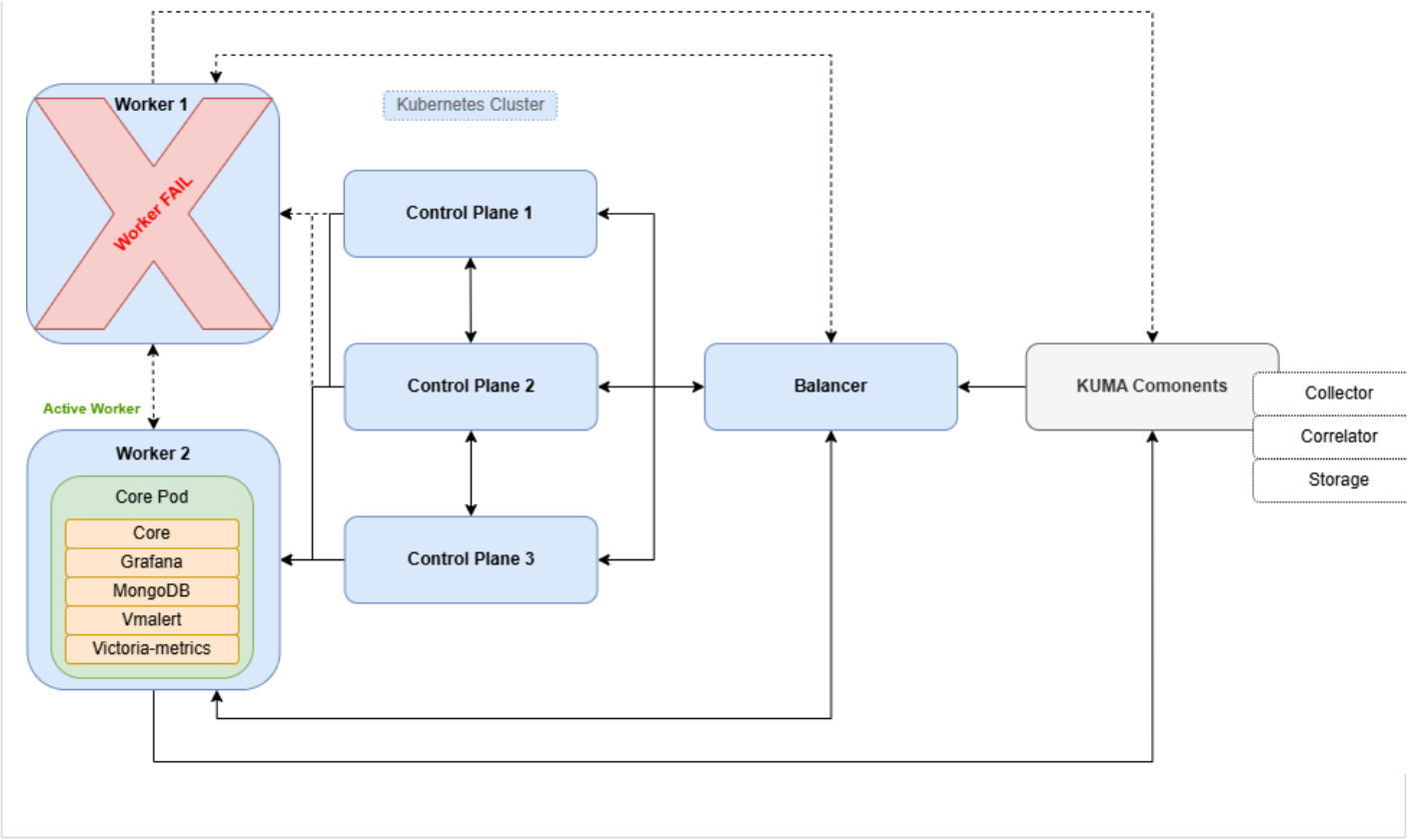
Схема работы в случае отказа оного из воркеров ядра



Через ~ 5 минут



Через ~ 1-2 минуты



Для установки KUMA в отказоустойчивом исполнении используется установщик kuma-ansible-installer-**ha**-2.1. X.tar.gz. Конфигурация кластера Kubernetes задается в файле инвентаря `k0s.inventory.yml`. Требования к устройствам для установки KUMA в Kubernetes - https://support.kaspersky.com/help/KUMA/3.0.2/ru-RU/217889.htm

Порты доступа для кластера ядра

Трафик KUMA core в отказоустойчивой конфигурации (трафик, в котором и источником и получателем выступают внешние сервисы KUMA здесь не рассматривается)

В таблице указаны инициатор соединения (источник) и назначение. Номер порта на инициаторе может быть динамическим. Обратный трафик в рамках установленного соединения не должен блокироваться

Источник	Назначение	Порт назначения	Тип
Внешние сервисы KUMA	Балансировщик нагрузки	7209	tcp
Внешние сервисы KUMA	Балансировщик нагрузки	7210	tcp
Внешние сервисы KUMA	Балансировщик нагрузки	7220	tcp
Внешние сервисы KUMA	Балансировщик нагрузки	7222	tcp
Внешние сервисы KUMA	Балансировщик нагрузки	7223	tcp

Рабочий узел	Балансировщик нагрузки	6443	tcp
Рабочий узел	Балансировщик нагрузки	8132	tcp
Управляющий узел	Балансировщик нагрузки	6443	tcp
Управляющий узел	Балансировщик нагрузки	8132	tcp
Управляющий узел	Балансировщик нагрузки	9443	tcp
Рабочий узел	Внешние сервисы KUMA	в зависимости от настроек при создании сервиса	tcp
Балансировщик нагрузки	Рабочий узел	7209	tcp
Балансировщик нагрузки	Рабочий узел	7210	tcp
Балансировщик нагрузки	Рабочий узел	7220	tcp
Балансировщик нагрузки	Рабочий узел	7222	tcp
Балансировщик нагрузки	Рабочий узел	7223	tcp
Внешние сервисы KUMA	Рабочий узел	7209	tcp
Внешние сервисы KUMA	Рабочий узел	7210	tcp
Внешние сервисы KUMA	Рабочий узел	7220	tcp
Внешние сервисы KUMA	Рабочий узел	7222	tcp
Внешние сервисы KUMA	Рабочий узел	7223	tcp
Рабочий узел	Рабочий узел	179	tcp
Рабочий узел	Рабочий узел	9500	tcp
Рабочий узел	Рабочий узел	10250	tcp
Рабочий узел	Рабочий узел	51820	udp
Рабочий узел	Рабочий узел	51821	udp
Управляющий узел	Рабочий узел	10250	tcp
Балансировщик нагрузки	Управляющий узел	6443	tcp
Балансировщик нагрузки	Управляющий узел	8132	tcp
Балансировщик нагрузки	Управляющий узел	9443	tcp
Рабочий узел	Управляющий узел	6443	tcp
Рабочий узел	Управляющий узел	8132	tcp
Рабочий узел	Управляющий узел	10250	tcp

Управляющий узел	Управляющий узел	2380	tcp
Управляющий узел	Управляющий узел	6443	tcp
Управляющий узел	Управляющий узел	9443	tcp
Управляющий узел	Управляющий узел	10250	tcp
Консоль управления кластером (CLI)	Балансировщик нагрузки	6443	tcp
Консоль управления кластером (CLI)	Управляющий узел	6443	tcp

Минимально кластер должен включать:

- один контроллер (выделенный или совмещенный с рабочим узлом);
- один рабочий узел (выделенный, или совмещенный с контроллером);
- 0 и более выделенных рабочих узлов.

> Минимальная конфигурация, на которую можно произвести установку - один контроллер, совмещенный с рабочим узлом. Данная конфигурация не обеспечивает отказоустойчивости core и служит для демонстрации возможностей/проверки программной среды.

> Для реализации отказоустойчивости необходим выделенный контроллер кластера и минимум 2 рабочих узла. Если контроллер кластера содержит рабочую нагрузку и под (pod) с Core размещается на нем, то его отключение приведет к полной потере доступа к Core.

> На всех компонентах ядра должно быть единое время, настройте на всех машинах NTP

> На контроллерах кластера должен быть уникальный machine-id, это значит, что не рекомендуется клонирование машин с этой ролью, либо необходимо изменить ID на машинах до установки. **Внимание!** Допустимость данной операции должен определять администратор хоста с учётом возможного использования machine-id другими сервисами! `rm /etc/machine-id /var/lib/dbus/machine-id && dbus-uuidgen --ensure=/etc/machine-id && dbus-uuidgen --ensure && reboot`

1. В нашем случае мы будем использовать установку All-In-One хост kuma-1.local, один узел контроллера (хост kuma-2.local) и два рабочих узла (хост kuma-3.local и kuma-4.local), пример файла инвентаря: https://box.kaspersky.com/f/bf06497b5b004dc3b1e5/

Другие примеры инвентарей: https://box.kaspersky.com/d/b397490dc08048acb671/

2. В распределенной установке kuma в секции инвентаря kuma_core нужно указать хост, который есть в роли worker (один из двух)
3. ВАЖНО! Для успешной установки должны быть соблюдены следующие требования:
   - все машины кластера должны быть добавлены в `/etc/hosts`;
   - установлены пакеты в соответствии с: https://support.kaspersky.com/help/KUMA/2.1/ru-RU/244399.htm;
   - На Astra Linux на машине балансировщика нужно установить в дополнение пакету nginx еще один пакет **libnginx-mod-stream**
   - в `/var/lib/` должно быть не менее 32GB свободного места;
4. Значение переменных в инвентаре ansible:
   - `need_transfer` – установка KUMA 2.1 происходит поверх предыдущей версии?;
   - `airgap` - значение неважно, может отсутствовать;
   - `low_resources` – использовать минимальные ресурсы для разворачивания? Отсутсвует по умолчанию. (Достаточно ресурсов: 2 CPU 4 RAM, **НО при этом создается том хранения 4 Гб**, без этого параметра том создается 512 Гб)
   - для части инвентаря kuma_k0s и переменных ansible_host важно указывать IP адреса
   - `kuma_managed_lb: false` - если используется собственный (балансировщик организации, не KUMA) балансировщик, при этом указать FQDN этого балансировщика (для корректного формирования сертификата коры)
   - `no_firewall_actions: false` - инсталлятор будет пытаться открыть необходимые порты на МЭ данного хоста
5. Создайте резервную копию ресурсов и сертификатов, см. советующий раздел в этой инструкции.
6. Распакуйте архив (операции выполняются на ядре системы KUMA): `tar -xvf kuma-ansible-installer-(ВЕРСИЯ).tar.gz`
7. Перейдите в распакованную папку: `cd kuma-ansible-installer`
8. Добавить файл лицензии в папку kuma-ansible-installer/roles/kuma/files и переименовать на license.key: `cp ПУТЬ_ДО_КЛЮЧА*.key roles/kuma/files/license.key`
9. Выполните команду копирования шаблона (пример заполненного файла в п. 0): `cp k0s.inventory.yml.template k0s.inventory.yml`
10. ВАЖНО! Регистр написания хостнеймов в inventory должен совпадать с выводом значения на хостах команды `hostname -f`
11. ВАЖНО! Хостнейм при команде `hostname -f` должен содержать хотя бы одну точку, пример: kuma.local
12. Входим в ОС из-под суперпользователя (root), если это не было сделано ранее: `sudo -i`
13. Запустите установку: `./install.sh k0s.inventory.yml`
14. Зайдите на веб интерфейс ядра KUMA по одному из адресов рабочих узлов или балансировщика, например, в нашем случае это - https://192.168.0.153:7220 Учетные данные для входа по умолчанию: `admin / mustB3Ch@ng3d!`
15. Для начального администрирования кластера воспользуйтесь командами **этого раздела**.

В случае, если при установке произошел сбой (НЕ обновлении), перед последующей установкой рекомендуется выполнить uninstall.sh и перезагрузить все узлы кластера. Если uninstall выполнить нельзя (идет миграция существующей установки в кластер), то перед повторной попыткой установки нужно вручную выполнить команду **sudo k0s reset** (если долгий reset, то **rm -rf /var/lib/k0s/containerd**, затем **k0s reset -d**) на всех узлах кластера и перезагрузить их

Перестроение между воркерами в кластере Kubernetes происходит с таймаутом ~ 5 мин

Отказоустойчивость балансировщиков, см. **тут**

Для работы с кластером можно использовать команды и инструменты **отсюда**

Видео установки в конфигурации AiO-1LB-1CP-2W **тут**

Revision #31
Created 11 August 2023 08:36:02 by Boris Rzr
Updated 16 January 2025 11:58:40 by Boris Rzr