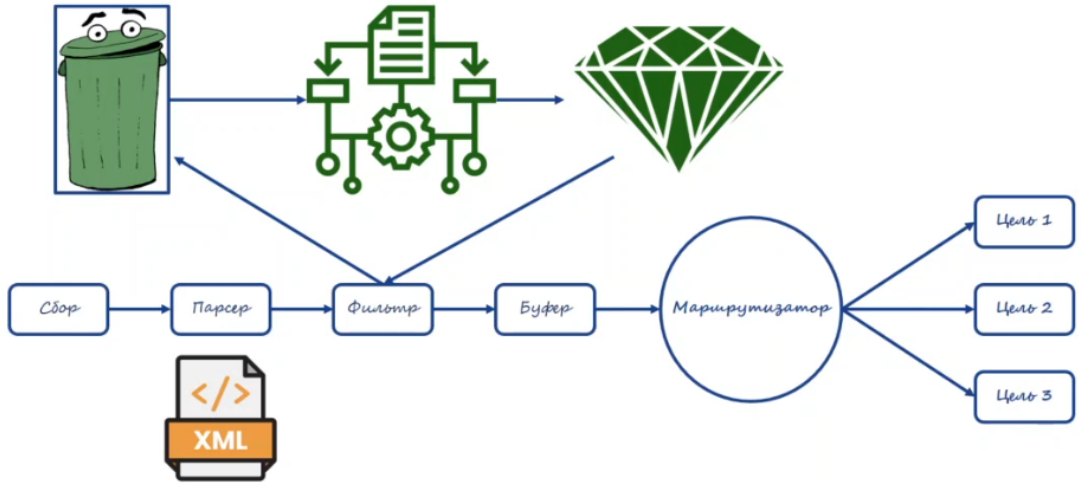
Модуль 5, урок 2

Сбор и анализ логов — тема интересная, а когда мы говорим про Kubernetes, она становится еще и сложной. Ведь K8s — это большой слоеный пирог из целого ряда различных решений. Каждое из них генерирует нечеловеческие объемы информации в виде событий и сообщений, которые необходимо собирать, концентрировать и обрабатывать.

Для сбора логов мы предлагаем использовать Fluent Bit. Это инструмент с открытым исходным кодом, который позволяет собирать логи со всех уровней вложенности K8s, обогащать их фильтрами и отправлять необходимым агрегаторам. Он простой, легковесный, но при этом еще и максимально производительный.

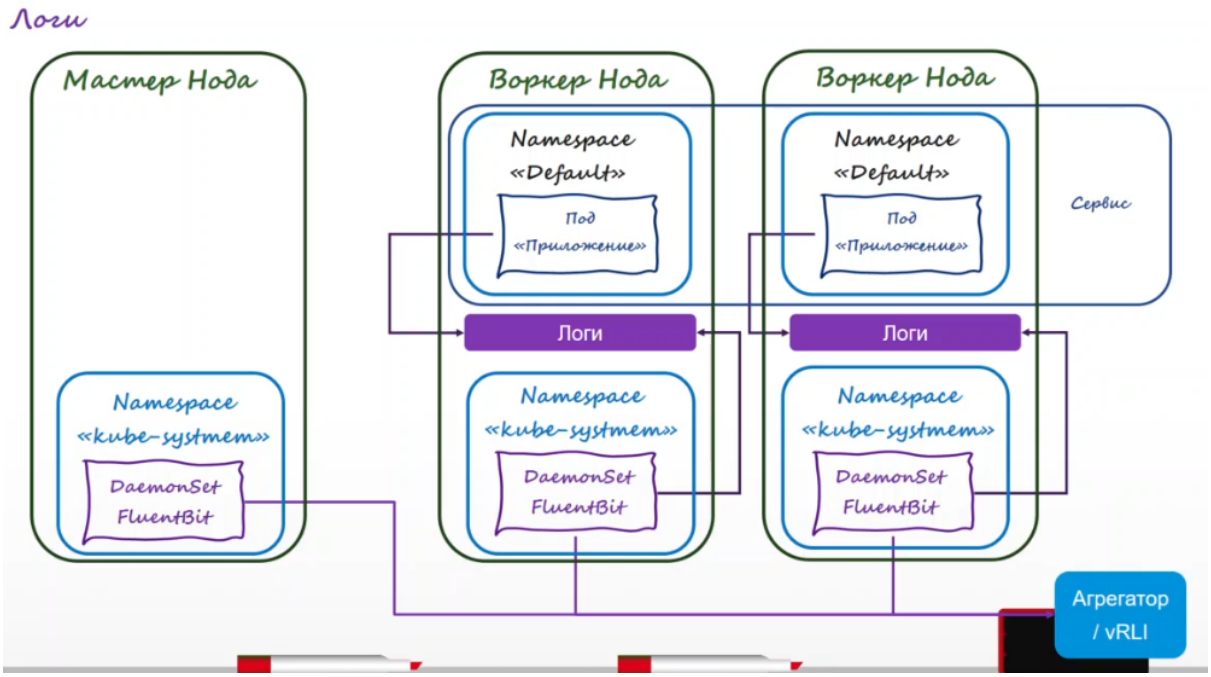
Архитектура Fluent Bit модульная и разделена на конкретные зоны ответственности разных модулей.

Поток данных



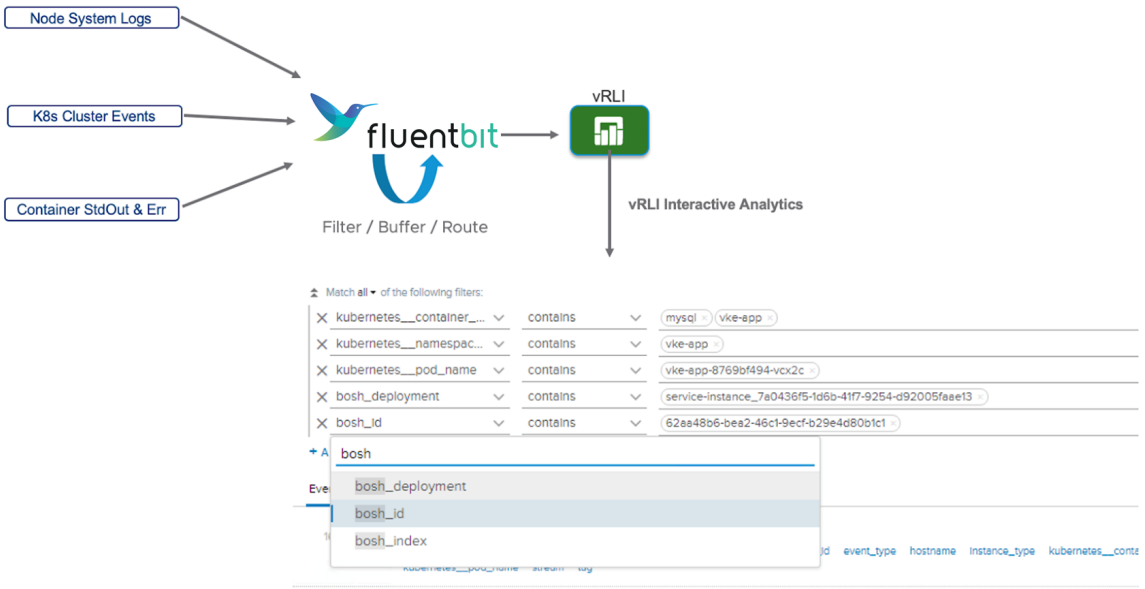
- За сбор данных отвечают отдельные плагины. Из коробки доступно огромное количество таких плагинов, но важно учитывать, что при загрузке плагин запускает свой инстанс и процесс. У каждого процесса — своя независимая конфигурация. Это позволяет очень точечно настраивать, как и откуда собирать данные.
- Логи — это плохо структурированный набор данных. Работать с ними без структуры — настоящая боль. Парсер позволяет анализировать входящие строки данных и превращать их в удобный XML-формат.
- В продуктиве хотелось бы не только данные собирать и структурировать, но и контролировать. Фильтрация как раз и позволяет изменить данные перед их доставкой и отсеять мусорную информацию, обогатить нужные нам данные метаданными, придав им ценности.
- Fluent Bit предлагает механизм буферизации данных, который нужен для своего рода отказоустойчивости. Например, чтобы избежать потери данных в случае сбоев со стороны системы. Данные можно хранить на уровне файловой системы до их отправки в целевые агрегаторы.
- Механизм маршрутизации необходим для отправки логов в целевую систему. Сам механизм прост и изящен: данные тегируются при сборе, маршрутизатор анализирует тег и перенаправляет в нужную целевую систему согласно вашим предпочтениям.
- Отправка данных в целевые системы тоже построена на модульной архитектуре для поддержки большего количества различных протоколов.

Как Fluent Bit собирает логи



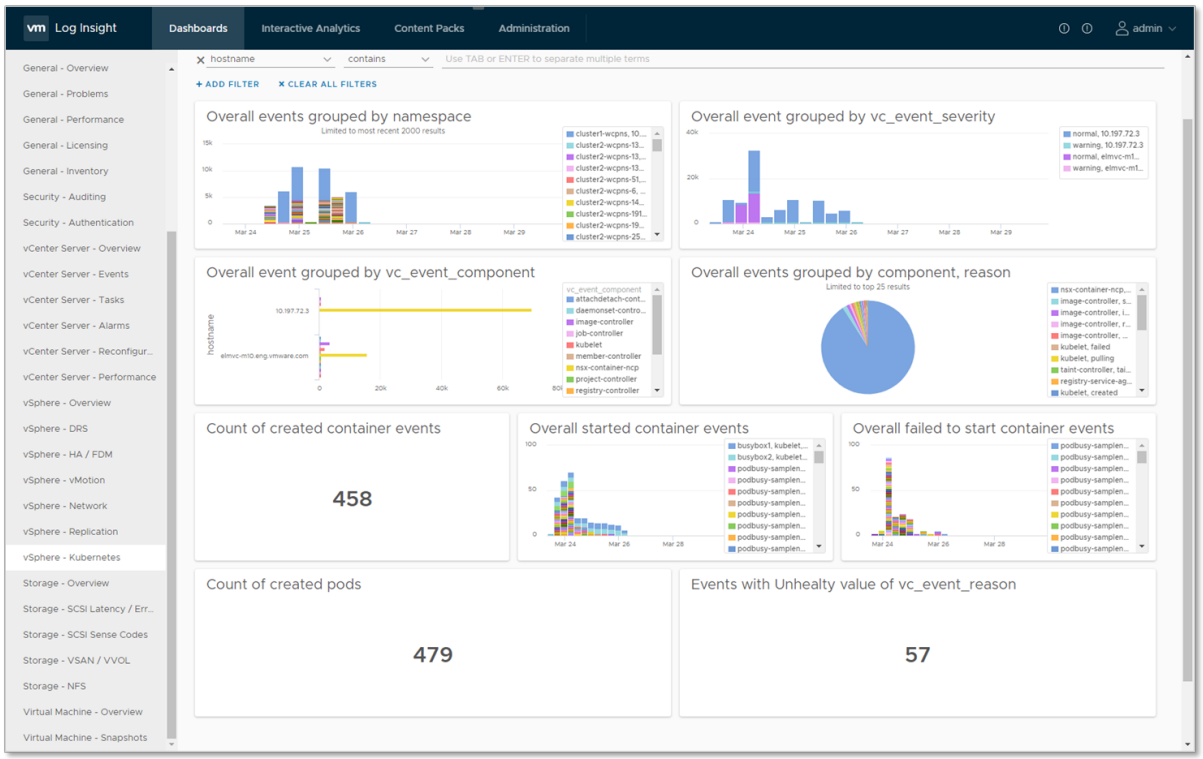
1. Контейнерный движок, на котором работают поды, автоматически перенаправляет потоки данных в единое хранилище. Разные движки делают это по-разному. Например, Docker по умолчанию пишет файлами в формате JSON.
2. В отличие от Prometheus, у Fluent Bit нет выделенного сервера. Вместо этого агент запускается в виде DaemonSet на каждой ноде кластера.
3. FluentBit парсит эти хранилища, собирая данные, делает первичную обработку, добавляя ряд метаданных, чтобы облегчить дальнейший анализ логов.
4. Fluent Bit посылает данные в некую целевую систему. В нашем случае — в vRealize Log Insight, но можно использовать любой доступный плагин и систему.

Сбор и анализ логов Kubernetes



vRealize Log Insight — это инструмент для сбора и анализа логов на базе машинного обучения. Он позволяет строить простые и понятные графики, делать глубокую аналитику логов, используя внутренний движок и свой собственный язык запросов. Этот движок, анализируя лог, разбирает его на составные части, понимая, что есть поле лога, а что — значение этого поля. Это позволяет легко оперировать логами и искать нужные записи, используя встроенный язык запросов.

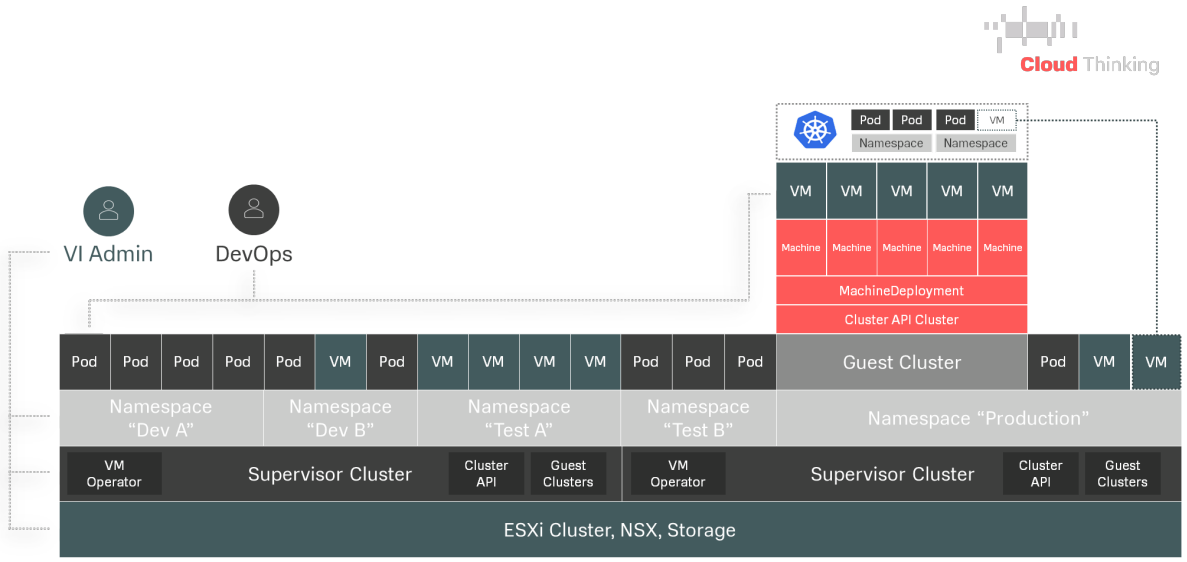
Log Insight собирает логи не только с K8s, но и со всех нужных вам систем. Как мы уже говорили, он позволяет строить простые дашборды и находить корреляции между событиями. Например, вы можете собирать логи о создании и ошибках создания подов и логи с инфраструктуры и каким-то образом находить корреляции между этими событиями.



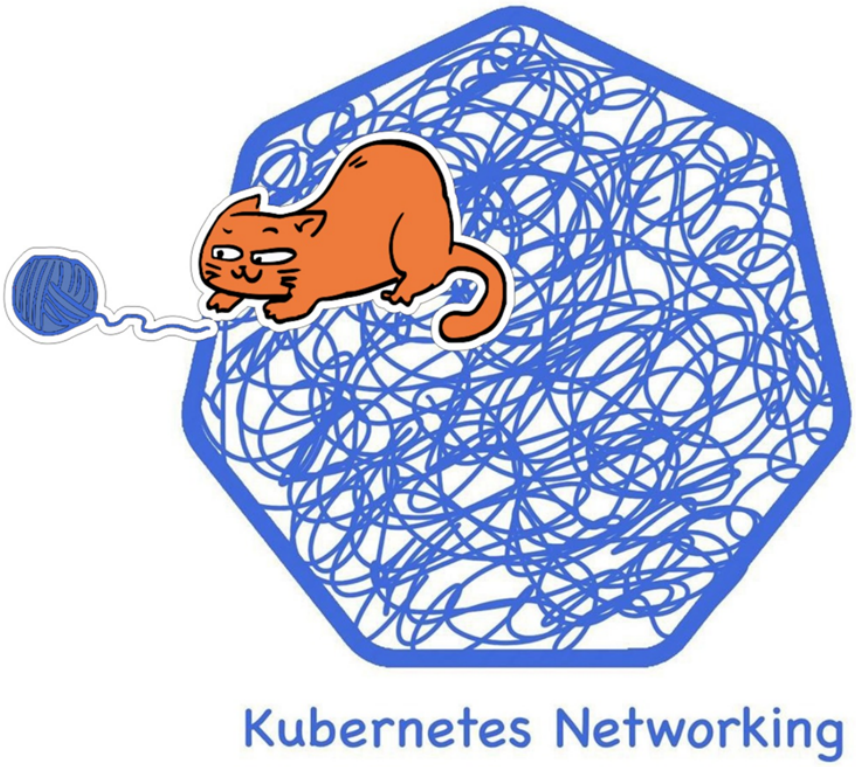
Мониторинг сети и визуализация

Прежде чем мы перейдем к мониторингу сети, разберемся, зачем это вообще нужно с точки зрения микросервисной архитектуры.

В современной распределенной инфраструктуре даже в рамках одного кластера могут быть смешаны различные технологии и приложения. У вас могут быть и традиционные виртуальные машины, и современные микросервисные решения. Вариантов таких решений тоже может быть много: например, vSphere-поды, которые работают в виде маленьких ВМ, но тоже подами, поверх ESXi-хостов. Или вложенные кластеры, внутри которых работают тоже поды и контейнеры.

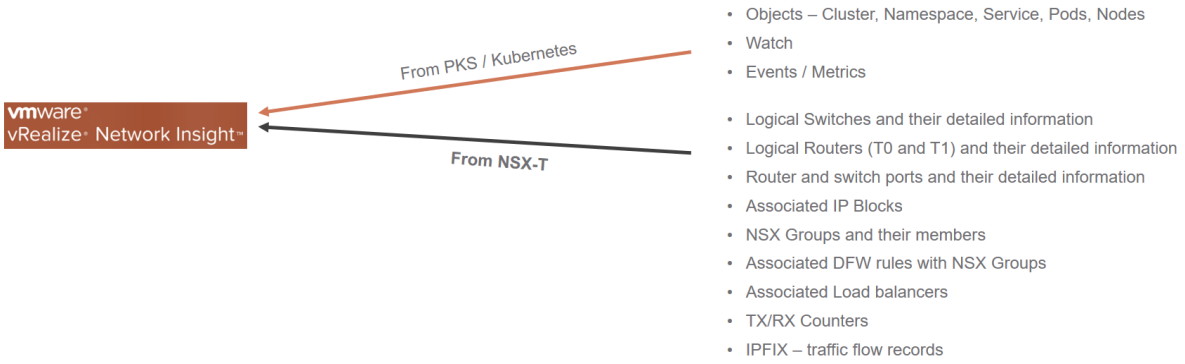


На верхнем уровне такая сеть похожа на большой и очень запутанный клубок ниток. Как обеспечить сетевую взаимосвязь между всеми этими компонентами, а также сегментацию и безопасность трафика? Ведь хочется иметь возможность на простых и понятных графиках отследить все сетевые потоки и коммуникации сервисов на уровнях от кластера до конкретного порта конкретного контейнера.



Для этого существует подходящий инструмент — vRealize Network Insight. Для его работы с микросервисной архитектурой нужен NSX и NSX Container Plugin. NSX в связке с vRNI позволяет собирать данные обо всех сетевых

потоках в инфраструктуре, а на базе связки с Kube API — собирать данные об элементах в K8s и сопоставлять их с существующими сетевыми потоками, по факту отрисовывая на графе понятное взаимодействие между всеми компонентами.



vRealize Network Insight позволяет строить понятные сетевые топологии ваших микросервисов, чтобы этот спутанный клубок приобрел понятную и четкую форму. Более того, анализируя, он предлагает различные варианты защиты трафика, позволяя даже выгрузить готовые YAML-файлы для применения на уровне K8s-инфраструктур.