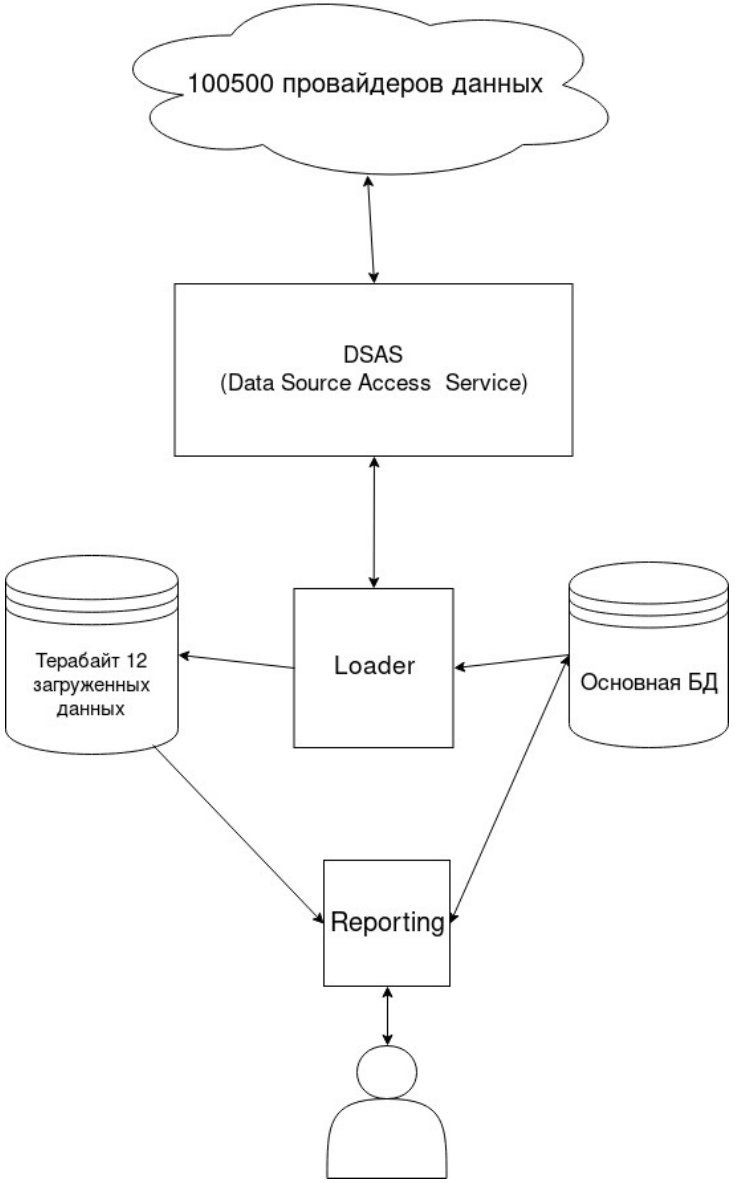
Секта свидетелей CNCF: от деплоя через git pull до CI/CD с Kubernetes

Андрей Инишев для DevPro“19

Кто мы?

- Стартап с головным офисом в Сан-Франциско
- ETL и визуализация для 150+ источников данных
- Филиалы в Москве и Томске

Сильно упрощенная архитектура



Преамбула

• Изначально проект в спешке создавался под нескольких крупных клиентов.

• Деплой через git pull + рестарт сервисов под одним пользователем на всех. В лучшем случае это завернуто в shell-скрипт.

• Либо юнит-тестов нет, либо они сломаны — CI не было. Оказывается, все надежды были на то, что перед коммитом тесты будут дергаться разрабочиком самостоятельно. Ну вы поняли.

Нет, это еще не все

- Сервера-«снежинки»
- AWS — это глобально и надежно. Виртуалки там никогда не падают, несколько машин с лоудбалансером перед ними — для слабаков, готовая встать на подхват реплика бд — для хипстеров
- Быстрые коммиты в мастер фикса и фикса фикса как альтернатива откату релиза
- Тем временем от монолита начали почковаться сервисы поменьше

Хватит это терпеть!

Начнем с CI

- Оптимальным выбором оказался Jenkins — есть плагины на любые случаи жизни. Немного больно, но это ради искусства.
- Под это же дело можно сделать контейнеризацию. Одно и то же окружение локально и при прогоне тестов — это же замечательно! Да и потом пригодится...
- Но сначала фикс нескольких сотен юнит-тестов!
- Выкат по-прежнему руками. Ну, точнее, ими надо запустить у себя ансиблоплейбук.

Лиха беда начало

- Надо деплоить собранные CI образы на стейджинг
- Надо деплоить собранные CI образы на прод
- Конечно, по кнопке из дженкинса
- И мониторинг всего этого хозяйства
- И логи начать наконец-то собирать
- И отказоустойчивость с быстрым масштабированием
- И с ума не сойти этим всем управлять
- А еще откаты\обновления без простоев

Время проектов CNCF

- CNCF (Cloud Native Computing Foundation) — фонд при The Linux Foundation, курирующий ряд инфраструктурных проектов
- Kubernetes — система оркестрации контейнеров
- Prometheus — система мониторинга
- Etcd — высоконадежное хранилище типа «ключ-значение». В нем Kubernetes хранит свое состояние.
- CoreDNS — расширяемый DNS-сервер — используется для service discovery внутри кластера.
- Helm — менеджер пакетов для Kubernetes. Сейчас активно стараемся от него избавиться.

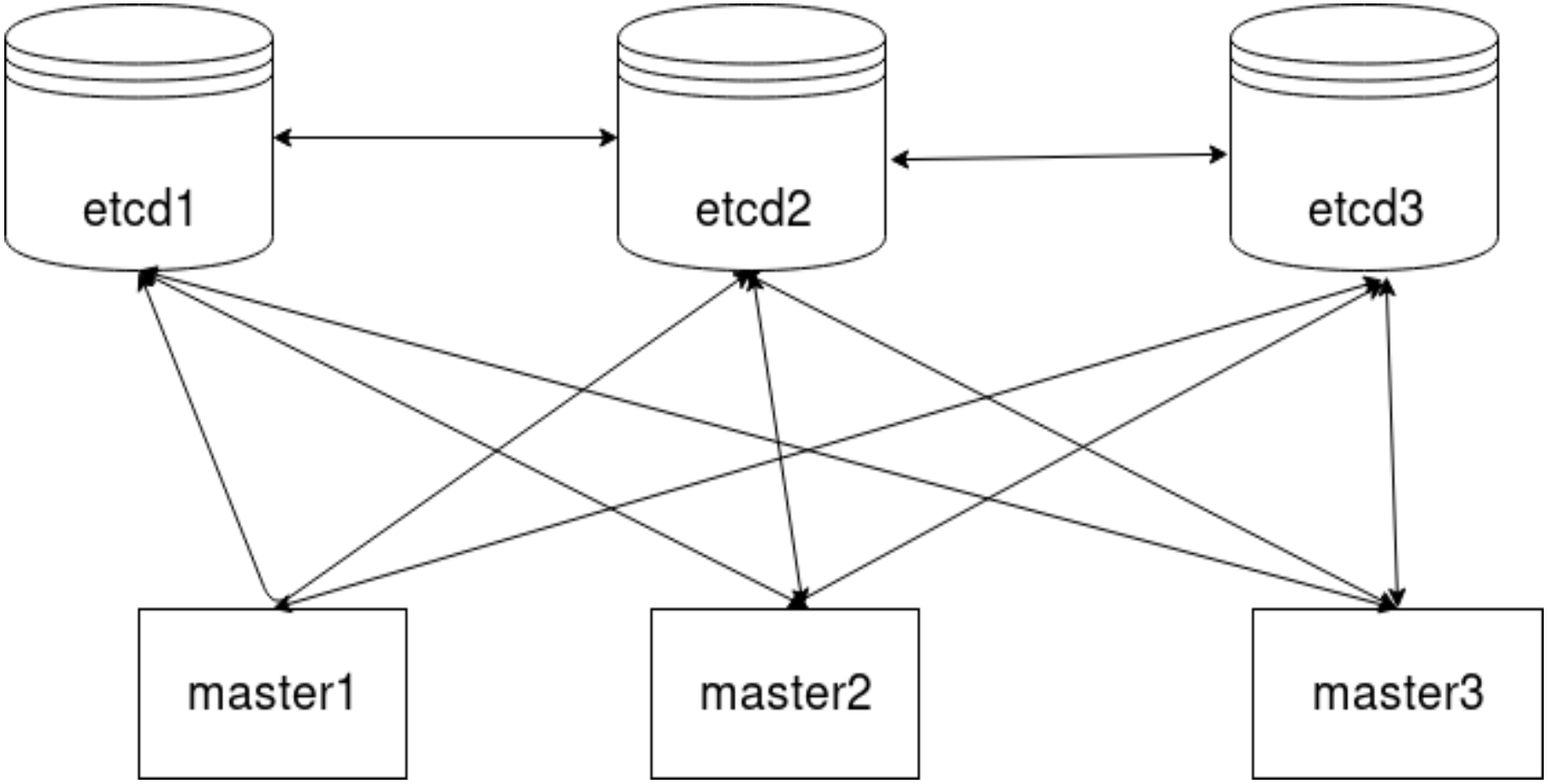
Kubernetes

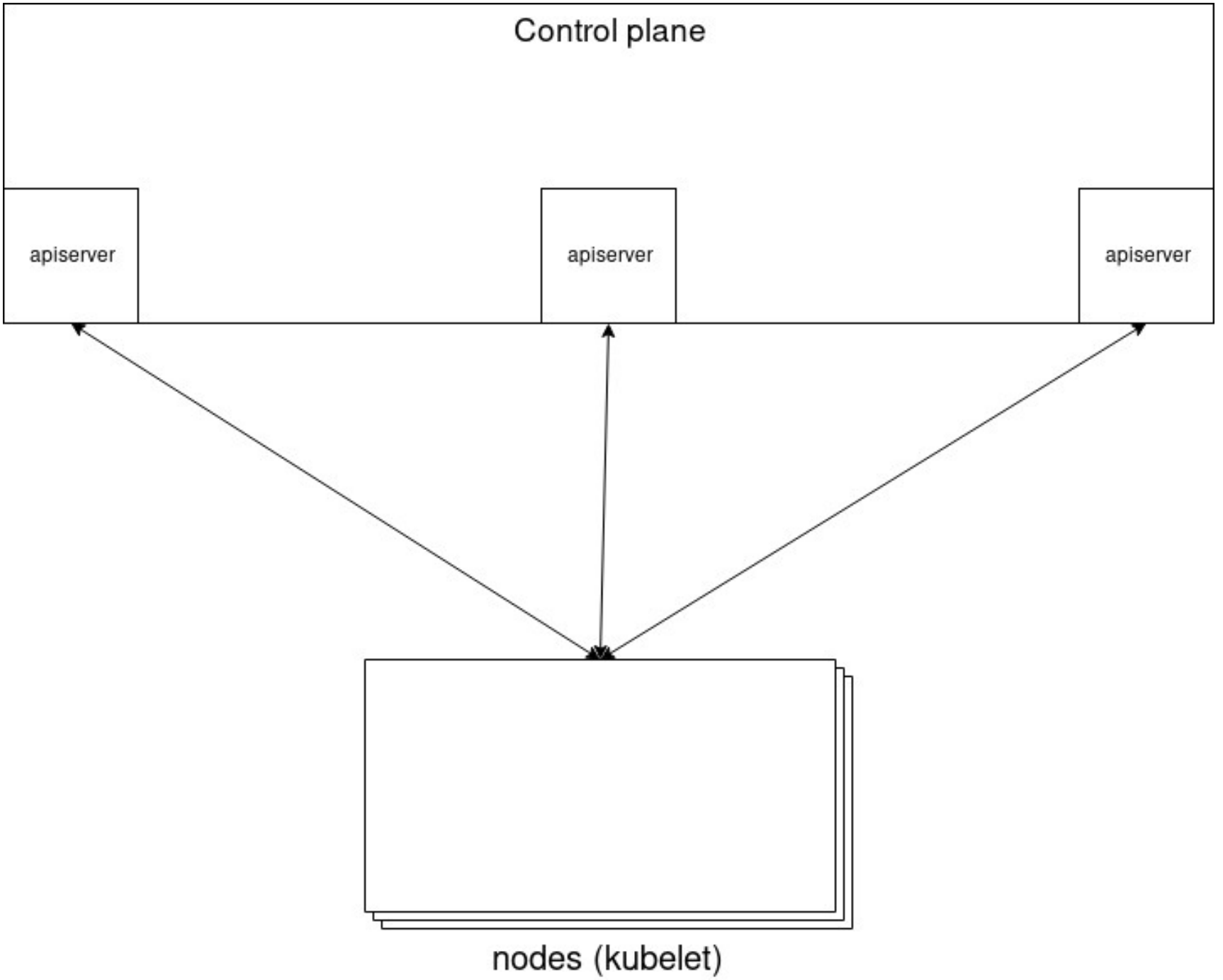
- Декларативное API
- Минимальная рабочая единица — под: один или несколько контейнеров, запущенных на одной ноде с общим сетевым неймспейсом и volumes
- Сам по себе под может быть удален или потерян вместе с нодой. Поэтому поверх спецификации пода строятся ресурсы, которыми управляют контроллеры

Например

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

Control plane




Control plane
apiserver
apiserver
apiserver
nodes (kubelet)

В итоге получаем

1)Запуск билда в Jenkins

2)Сборка образа

3)Прогон тестов

4)Выкат стейджинга

5)Деплой по кнопке на продакшен

Helm

- Шаблонизатор, который рендерит yaml используя go-template и предоставленные значения переменных — в совокупности чарт (helm chart)
- При helm install результат рендеринга применяется к аписерверу и создается релиз — задеплоенный экземпляр чарта с конкретными значениями переменных
- Обновление\откат с помощью helm upgrade\ rollback
- Отказываемся в пользу операторов

Операторы

- Контроллеры для кастомных ресурсов
- Вместо шаблонов и значений — несколько объектов API
- Отслеживание состояния в реальном времени
- Кроме деплоя — выполнение административных задач\обновление конфигурации
- Бесконечная гибкость при решении задач
- Для несложных операторов подойдет Metacontroller

Мы используем

- https://github.com/tekliner/rabbitmq-operator - наш оператор для RabbitMQ. HA-инсталляция RabbitMQ с кластеризацией и декларативным описанием пользователей\политик
- https://github.com/coreos/prometheus-operator - оператор для Prometheus — декларативное объявление правил service discovery, инстансов прометея, alertmanager, правил для алертинга; оператор сам соберет конечный конфиг для каждого объявленного инстанса
- https://github.com/upmc-enterprises/elasticsearch-operator - elasticsearch + kibana
- Операторы для наших собственных сервисов

Пример с продакшена

```
apiVersion: rabbitmq.improvado.io/v1
kind: Rabbitmq
metadata:
  name: dts-master-rabbitmq-ha
  namespace: production
spec:
  cluster_formation.node_cleanup.only_log_warning: true
  cluster_node_cleanup_interval: 10
  cluster_partition_handling: autoheal
  default_vhost: dts
  hipe_compile: false
  image:
    name: rabbitmq
    tag: 3.7.14-alpine
  k8s_addrtype: hostname
  k8s_host: kubernetes.default.svc.cluster.imp
  k8s_service_discovery: svc.cluster.imp
  k8s_serviceaccount: rabbitmq-instance
  loopback_users.guest: false
  memory_high_watermark: 256M
  policies:
  - apply-to: all
    definition:
      ha-mode: exactly
      ha-params: 3
      ha-sync-mode: automatic
    name: ha-three
    pattern: .*
    priority: 0
    vhost: dts
  prometheus_exporter_port: 9090
  replicas: 3
  secret_service_account: dts-master-rabbitmq-secret
  volume_size: 10Gi
```

Prometheus

- Умеет в service discovery через kubernetes из коробки
- Может управлять автоскейлингом через экспортер в metrics server api
- Kubelet отдает кучу необходимых метрик в его формате
- Много готовых дашбордов для grafana + kubernetes
- Один из первых инструментов, который нужно настроить сразу после разворачивания кластера

Сбор логов в ES — через filebeat

- Дружит с kube-apiserver
- Хорошая документация
- Почти все работает из коробки и сразу
- Довольно нетребователен к ресурсам

Иммутабельность

- Образы мастеров и нод kubernetes собираются с помощью packer и ansible
- Инфраструктура описана в виде кода с использованием terraform
- Откат\обновление кластера — смена версии образа в коде + terraform apply
- Разумеется, сначала на тестовой лабе

Организационные грабли

- Даже не надо и надеяться, что разработчики проявят инициативу в чем-либо, что касается инфраструктурных знаний
- По возможности — объяснять логику работы, а не давать готовые ответы
- По возможности, участвовать в митингах
- Или хотя бы читать их результаты — они могут быть неожиданными
- Терпение — все наладится

Спасибо за внимание