Октябрь 2017

10 причин [не] использовать k8s

Иван Глушков
@gliush
ivan.glushkov@gmail.com

Немного о себе

- Postmates, Infra команда
- MZ, Infra команда
- Echo, Backend команда
- MCST, Компиляторный проект
- DevZen подкаст, Со-основатель

План на сегодня

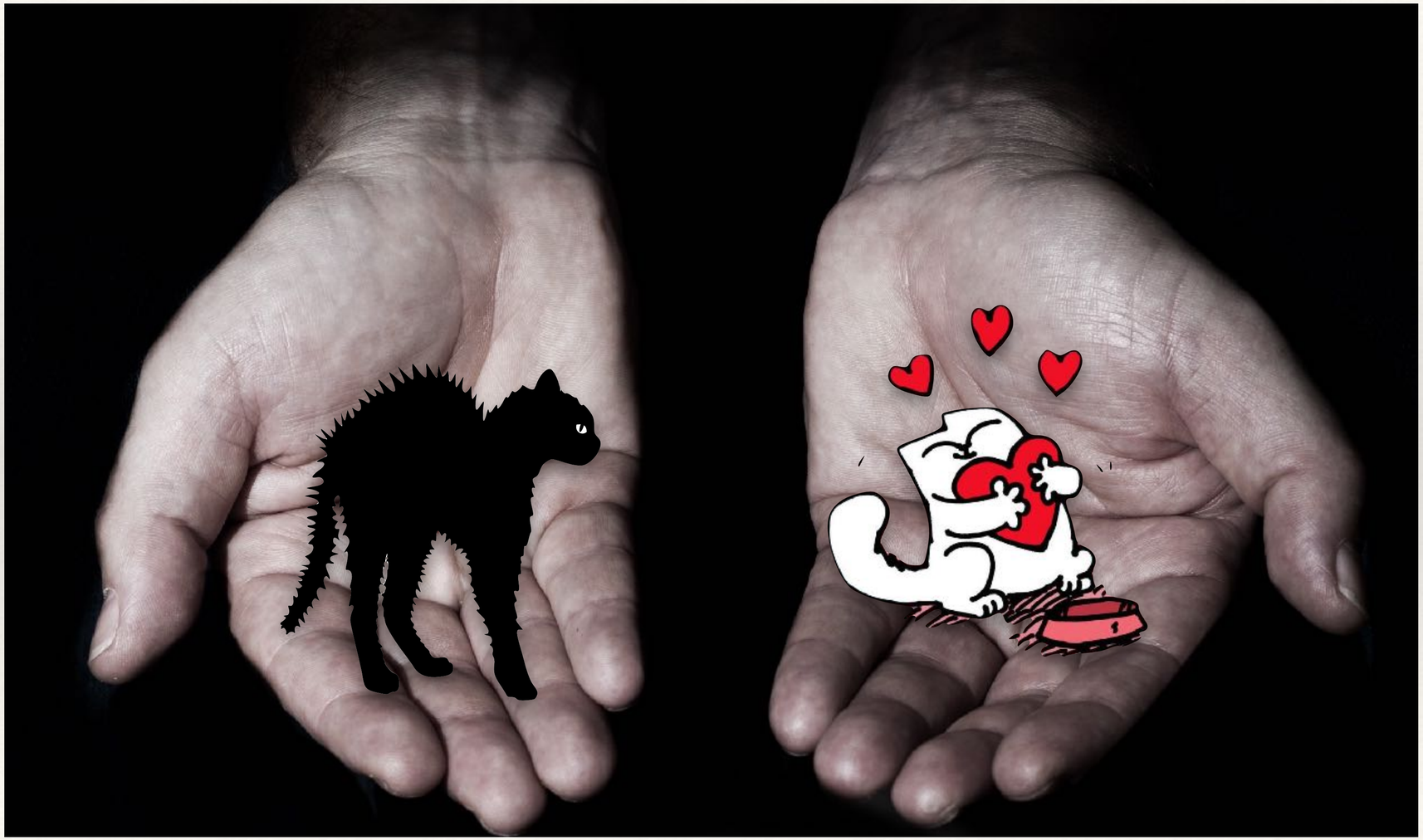
- Краткое введение в область (5 мин)
- Наш опыт и используемые технологии (30 мин)
- Выводы (5 мин)

Слайды можно скачать

In English

На Русском

Сделай свой выбор



Почему это вообще важно

- Почему <X> лучше чем <Y>?
- Puppet/Chef/Ansible/Bash+SSH - уже достаточно хороши
- Критерии:
  - Время от коммита до релиза
  - Автоматизация сборки, тестов, релиза
  - CI/CD [1]
  - Количество прилагаемых усилий

Очень быстрое погружение в k8s

- Masters - Slaves (Nodes)
- API, CLI
- Namespaces
- Pod, Labels, Selectors
- ReplicaSet, Services, Deployments

Важность для разработчиков

- Повторяемость
- Стандартизированное окружение
- Упрощенная процедура тестирования
- CI

Важность для релиз-инженеров

- Иммутабельная инфраструктура
- Инфраструктура как код
- Идемпотентность
- Релиз одной кнопкой (почти)
- Простота и скорость отката релиза
- Интроспекция и анализ проблем
- CD

Заблуждения

- Контейнеры - это облегченные виртуальные машины [2]
- k8s делает ваше приложение более защищенным
- k8s автоматически делает ваше приложение скалируемы
- k8s работат с любой ОС без проблем

Опыт

Опыт: Начало

- Нужен большой дополнительных стек вокруг k8s
- Требуется изучение большого стека технологий
- Нужно беспокоиться об отказоустойчивости многих сервисов этого стека

Experience: Observation

- Нужен большой дополнительных стек вокруг k8s
- Требуется изучение большого стека технологий
- Нужно беспокоиться об отказоустойчивости многих сер[illegible]в [illegible] стека

Experience: Observation

- k8s не ходит в одиночку
- Требуется изучение большого стека технологий
- Нужно беспокоиться об отказоустойчивости многих сервисов этого стека
- Приходящие новички уже будут знать k8s

Experience: Observation

- k8s не ходит в одиночку
- Требуется изучение большого стека технологий
- Нужно беспокоиться об отказоустойчивости многих сервисов стека
- Приходящие новички уже будут знать k8s

Опыт: Helm [10]

- Это пакетный менеджер для Kubernetes
- Client (helm) + Server (Tiller: etcd)
- Chart (yaml) + templates (yaml) + values (yaml)
- Позволяет скачивать (из репы), устанавливать, откатывать пакеты
- Работает с плагинами
- Мы запускаем один Tiller на namespace

Опыт: Helm [10]

- Это пакетный менеджер для Kubernetes
- Client (helm) + Server (Tiller: etcd)
- Chart (yaml) + templates (yaml) + values (yaml)
- Позволяет скачивать (из репы), устанавливать, откатывать пакеты
- Работает с плагинами
- Мы запускаем один Tiller на namespace

Примеры конфигураций

- puppet/chef/ansible -> BOOMer
- Сравним конфигурацию boomer, k8s и helm

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

где (железо)

что (код)

как (подготовка кода)

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

где (железо)

что (код)

как (подготовка кода)

наследование

права доступа

настройка сетей (@ops)

планирование нагрузки

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

где (железо)

что (код)

как (подготовка кода)

наследование

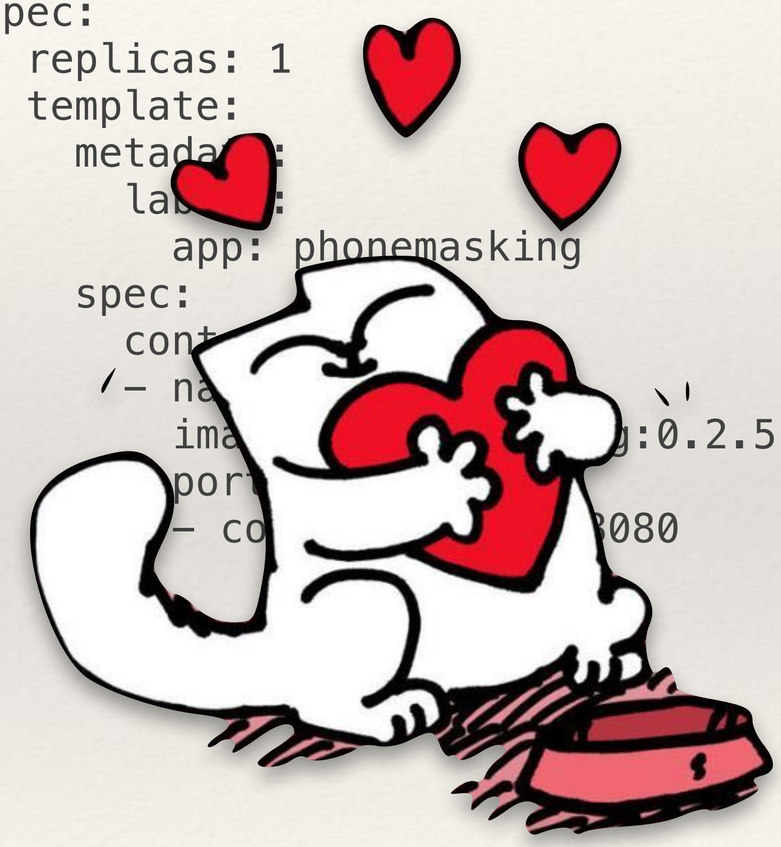
права доступа

настройка сетей (@ops)

планирование нагрузки

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metada
      lab
        app: phonemasking
    spec:
      cont
      - na
        ima                g:0.2.5
        port
        - co               3080
```



Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

```
apiVersion: v1
kind: Service
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 8080
    targetPort: 8080
  selector:
    app: phonemasking
```

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

```
apiVersion: v1
kind: Service
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 8080
    targetPort: 8080
  selector:
    app: phonemasking
```

ConfigMap ??

LB ??

Secrets ??

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mountPoint: /usr/local/src/phonemasking
        url: git@github.com:postmates/phonemasking
        ref: master
    exec: bash provision.sh
  launch:
    tags:
      Service: phonemasking

phonemasking-prod:
  inherit: [common]
  launch:
    iamInstanceProfile: app-base-prod
    environment: prod
    instanceType: m3.large
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

```
apiVersion: v1
kind: Service
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 8080
    targetPort: 8080
  selector:
    app: phonemasking
```

ConfigMap ??

LB ??

Secrets ??

Stage/Prod/Dev ??

Boomer + tf (опущено)

```
common:
  name: phonemasking
  region: us-west-1
  build:
    depends:
      url: git@github.com:postmates/pi-elixir.git
      name: elixir-us-west-1
    extraRepositories:
      - mount[illegible]r/[illegible]/src/phonemasking
        ur[illegible]ates/phonemasking
        r[illegible]
    exec:[illegible]
  launch:[illegible]
    tags:[illegible]
      Se[illegible]

phonema[illegible]ing[illegible]
  inher[illegible]: [illegible]n]
  laun[illegible]n:
    iamInsta[illegible]ofile: [illegible]-base-prod
    environme[illegible]: [illegible]d
    instanceT[illegible]: m3.larg[illegible]
    loadBalancerNames: [tf-phonemasking-prod-ext]
    subnetIDs: [subnet-bb123456,subnet-bb123457]
    securityGroups: [sg-aa123456, sg-aa123457]
    minSize: 2
    maxSize: 3
    desiredCapacity: 2
```

Kubernetes

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: phonemasking
    spec:
      containers:
      - name: server
        image: phonemasking:0.2.5
        ports:
        - containerPort: 8080
```

```
apiVersion: v1
kind: Service
metadata:
  name: phonemasking
  labels:
    app: phonemasking
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 8080
    targetPort: 8080
  selector:
    app: phonemasking
```

ConfigMap ??

LB ??

Secrets ??

Stage/Prod/Dev ??

Helm шаблон

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: {{ template "fullname" . }}
  labels:
    app: {{ template "name" . }}
    chart: {{ .Chart.Name }}-{{ .Chart.Version | replace "+" "_" }}
spec:
  replicas: {{ .Values.server.replicaCount }}
  template:
    metadata:
      labels:
        app: {{ template "name" . }}
    spec:
      imagePullSecrets:
        - name: postmates-k8s-deployer-pull-secret
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.server.image.repository }}:{{ .Values.server.image.tag }}"
          imagePullPolicy: {{ .Values.server.image.pullPolicy }}
          ports:
            - containerPort: {{ .Values.server.service.internalPort }}
```

Helm template

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: {{ template "fullname" . }}
  labels:
    app: {{ template "name" . }}
    chart: {{ .Chart.Name }}-{{ .Chart.Version | replace "+" "_" }}
spec:
  replicas: {{ .Values.server.replicaCount }}
  template:
    metadata:
      labels:
        app: {{ template "name" . }}
    spec:
      imagePullSecrets:
        - name: postmates-k8s-deployer-pull-secret
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.server.image.repository }}:{{ .Values.server.image.tag }}"
          imagePullPolicy: {{ .Values.server.image.pullPolicy }}
          ports:
            - containerPort: {{ .Values.server.service.internalPort }}
```

Helm values

```
server:
  replicaCount: 1
  image:
    repository: <phonemasking-image>
    tag: 8.8.2
    pullPolicy: IfNotPresent
  service:
    name: phonemasking
    type: LoadBalancer
    externalPort: 80
    internalPort: 8080
  resources: {}
```

Helm template

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: {{ template "fullname" . }}
  labels:
    app: {{ template "name" . }}
    chart: {{ .Chart.Name }}-{{ .Chart.Version | replace "+" "_" }}
spec:
  replicas: {{ .Values.server.replicaCount }}
  template:
    metadata:
      labels:
        app: {{ template "name" . }}
    spec:
      imagePullSecrets:
        - name: postmates-k8s-deployer-pull-secret
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.server.image.repository }}:{{ .Values.server.image.tag }}"
          imagePullPolicy: {{ .Values.server.image.pullPolicy }}
          ports:
            - containerPort: {{ .Values.server.service.internalPort }}
```

Helm values

```
server:
  replicaCount: 1
  image:
    repository: <phonemasking-image>
    tag: 8.8.2
    pullPolicy: IfNotPresent
  service:
    name: phonemasking
    type: LoadBalancer
    externalPort: 80
    internalPort: 8080
  resources: {}
```

Helm helpers

Helm chart

Helm notes

Helm template

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: {{ template "fullname" . }}
  labels:
    app: {{ ter[illegible]me"
    chart: [illegible]t.Version | replace "+" "_" }}
spec:
  replica[illegible]count }}
  templa[illegible]
    meta[illegible]
      la[illegible]
      [illegible] . }}
    spe[illegible]
      [illegible]ets:
      - na[illegible]stmates[illegible]-deployer-pull-secret
      [illegible]contai[illegible]
        - na[illegible].Chart.[illegible]e }}
          ima[illegible]Value[illegible]erver.image.repository }}:{{ .Values.server.image.tag }}"
          ima[illegible]ullPolicy: [illegible]Values.server.image.pullPolicy }}
          ports:
            - containerPort: {{ .Values.server.service.internalPort }}
```

Helm values

```
server:
  replicaCount: 1
  image:
    repository: <phonemasking-image>
    tag: 8.8.2
    pullPolicy: IfNotPresent
  service:
    name: phonemasking
    type: LoadBalancer
    externalPort: 80
    internalPort: 8080
  resources: {}
```

Helm helpers

Helm chart

Helm notes

Опыт: погружаемся глубже

Кластера

- e2e - для тестирования самого k8s
- prod, stage (полностью повторяет prod)
- admin:
  - CI (похоже, всегда будет самым большим),
  - консолидированный VPN (добавим вскоре),
  - interVPC конфигурация (добавим вскоре)
  - …

Опыт: CoreOS

- CloudFormation или TF
- Нет интроспекции текущего состояния кластера
- Возможные проблемы: при обновлении кластера, порядок действий
- -> Риски стабильности

Опыт: CoreOS

- CloudFormation или TF
- Нет интроспекции текущего состояния кластера
- Возможные проблемы: при обновлении кластера, порядок действий
- -> Риски стабильности

Опыт: Внутренний репозиторий образов

- Внутренний репозиторий (Docker Distribution) [3]
- Высокая цена поддержки:
  - Требуется отказоустойчивость (поломки ssd, сети, самой машины)
  - GC (масса устаревших и/или больших образов)
  - Высокая нагрузка (32 k8s нод, на каждой нужно закачать 2GB образ)

Опыт: Внешний репозиторий образов

- Quay как внешний репозиторий образов [4]
- Чаще всего непубличные образы: Secrets -> imagePullSecrets

```
$ cat secrets.yaml
…
kind: Secret
metadata:
  name: quay-pull-secret
data:
  .dockerconfigjson: ….
```

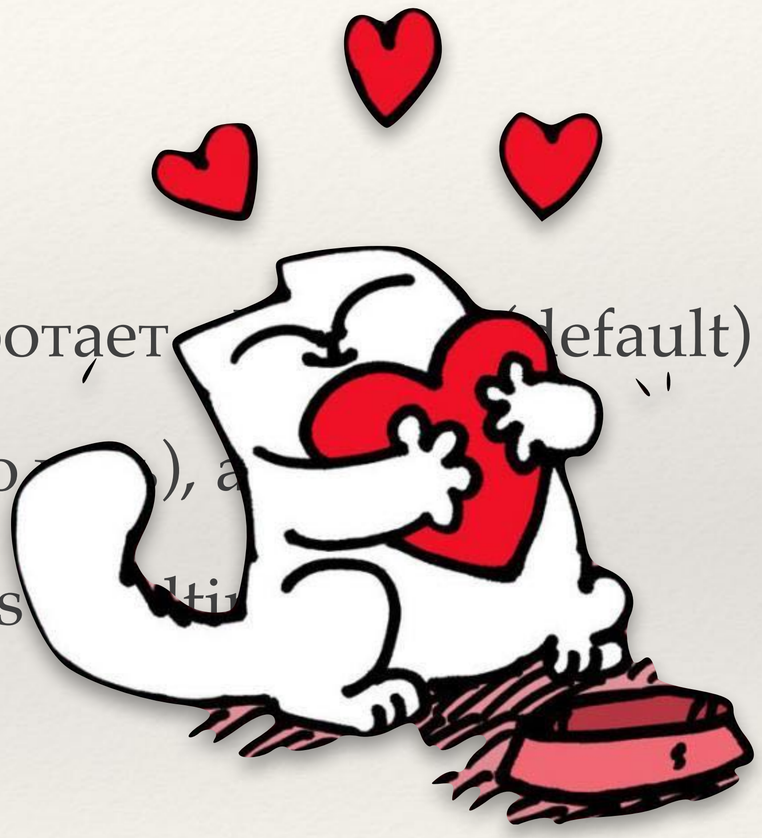
```
$ cat deployment.yaml
…
kind: Pod
spec:
  container:
      …
      imagePullSecrets:
      - name: quay-pull-secret
```

Опыт: kops

- Мы ранние пользователи
- Не поддерживает CoreOS до конца, неплохо работает с Debian8 (default)
- HA мастер-ноды, идемпотентность (state-sync модель), addons, …
- Нулевое время простоя во время rolling upgrades (multimaster)
- Поддержка режимов multi-cloud / multi-region

Опыт: kops

- Мы ранние пользователи
- Не поддерживает CoreOS до конца, неплохо работает [illegible] (default)
- HA мастер-ноды, идемпотентность (state-sync мо[illegible]), а[illegible]
- Нулевое время простоя во время rolling upgrades [illegible]
- Поддержка режимов multi-cloud / multi-region



Опыт: Сети

- Много вариантов [23]
- kops поддерживает только часть из них (Calico, Weave, Flannel, …)
- Можно использовать другие CNI провайдеры
- Лучше использовать самые популярные: лучше протестированы

Опыт: Calico

- Хорошо работает с минимальным количеством проблем
- Использует BGP, работает быстрее, чем через инкапсуляцию
- IP-in-IP для multi-AZ или multi-cloud [12]
- IP-in-IP: выключена, включена, или только для cross-subnet
- Calico поддерживает интеграцию с k8s сетью (изолировать трафик по меткам)

Опыт: Calico

- Хорошо работает с минимальным количеством проб
- Использует BGP, работает быстрее, чем через инкапсуля
- IP-in-IP для multi-AZ или multi-cloud [12]
- IP-in-IP: выключена, включена, или только для cr subn
- Calico поддерживает интеграцию с k8s сетью (изолиро меткам)

Опыт: HA

- 3 Master-ноды, синхронизация по etcd. Можно 5/7/9.
- Из-за баги, мы не переходили с etcd.v2 (обновимся на etcd.v3 вскоре)

Опыт: Восстановление после сбоев

- Специальный кластер для экспериментов и тестирования обновления самого k8s
- Делаем снепшоты всех дисков перед рискованными операциями

Опыт: Аутентификация и Авторизация

Опыт: Доступ до k8s

- RBAC (Role-Based Access Control) намного лучше чем ABAC
- Dex - провайдер OpenID для интеграции с источником данных [6]
- Логин-скрипт из консоли
- Или идти в to Dex UI, логиниться и ^C-^V в ~/.kube/config

Опыт: Управление Пользователями

- Active Directory как источник информации (по историческим причинам)
- LDAP для OpenVPN, Dex и Grafana для аутентификации
- Информация о группе транслируется в роли в k8s

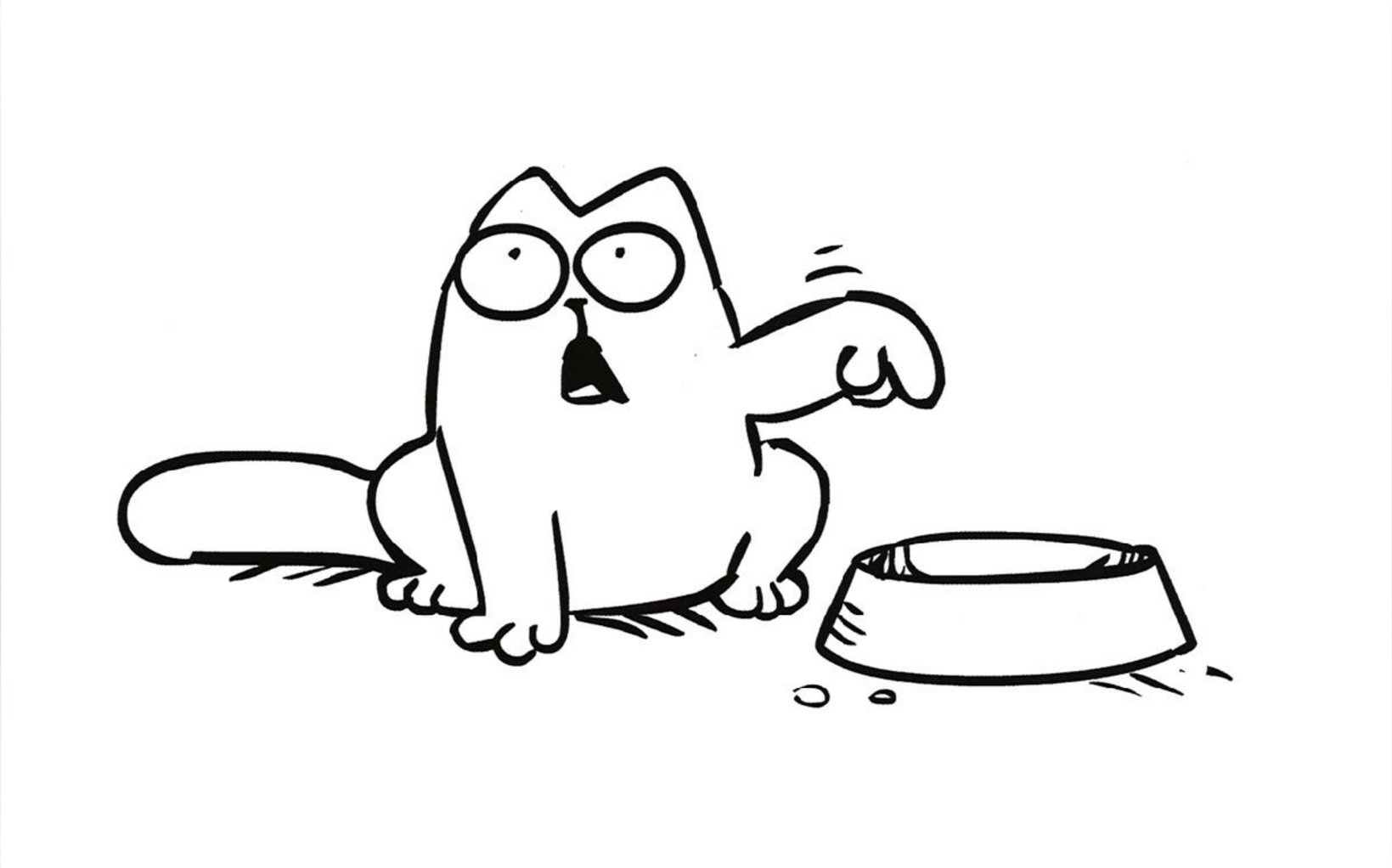
Опыт: Доступ в AWS

- Только для Infra команды (остальным - только если ОЧЕНЬ нужно)
- Нет долгоживущих админских ключей, используются временные, роли
- Ротация ключей
- Awsudo [7]

Опыт: Доступ в AWS

- Только для Infra команды (остальным - только если ОЧЕНЬ нужно)
- Нет долгоживущих админских ключей, используются [illegible] роли
- Ротация ключей
- Awsudo [7]

Опыт: Управление Ресурсами



Опыт: Применение Квот

- RBAC, ограничение доступа - по namespace, для каждой команды
- Минимальный набор квот для каждой команды
  - сейчас - только общее число объектов
  - позже - ЦПУ и Память

Опыт: Применение Квот

- RBAC, ограничение доступа - по namespace, для каждой команды
- Минимальный набор квот для каждой команды
  - сейчас - только общее число объектов
  - позже - ЦПУ и Память

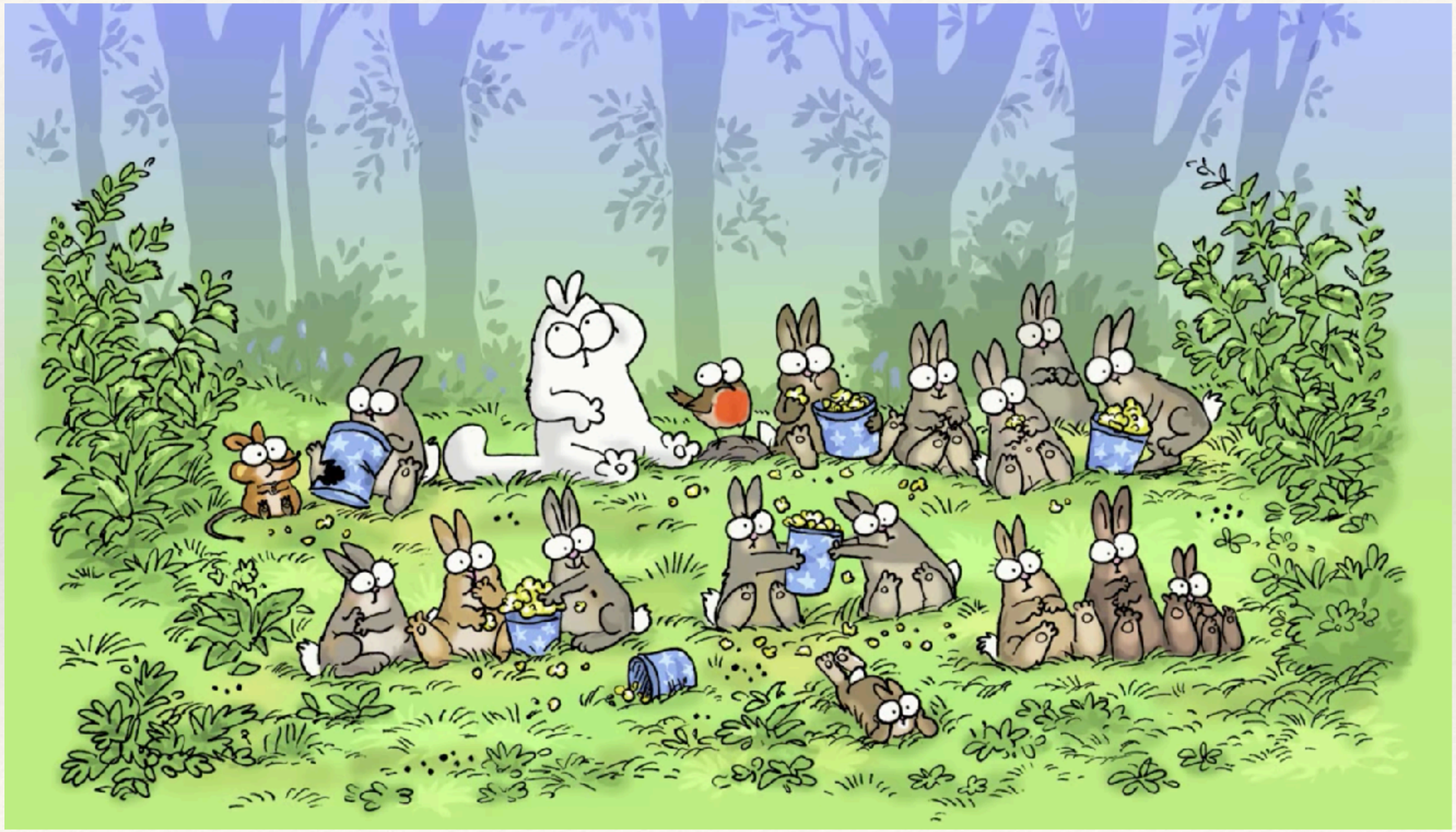
Опыт: Автоматическое Масштабирование

- Плагин: AWS Autoscaler (вертикально и горизонтально) [8]
- Использование AWS Spot инстансов может быть добавлено
- (Не путать с масштабированием Pod внутри k8s)

Опыт: Автоматическое Масштабирование

- Плагин: AWS Autoscaler (вертикально и горизонтально)
- Использование AWS Spot инстансов может быть управл
- (Не путать с масштабированием Pod внутри k8s)

Опыт: Телеметрия

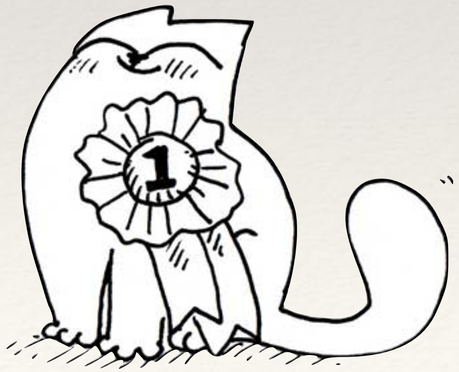


Опыт: Фреймворки для Телеметрии

- Sensu не очень подошёл, нужны более метрико-ориентированные [13]
- Prometheus имеет хороший язык запросов, алерты [16]
- InfluxDB + весь остальной TICK стек (Deis). UI хорош, SQL-подобный язык - ок, мало функциональности [14]

Опыт: Cernan [15]

- Много интерфейсов для получения данных (sources)
- Много интерфейсов для вывода данных (sinks)
- Возможность изменять данные на лету, e.g. log <-> metrics (filters)
- Написан на Rust



Опыт: Метрики

- Нужно плавно перейти с текущего стека (Statsd/Cernan/Wavefront)
- Prometheus будет забирать данные из Cernan
- Позже: каждый сервис будет делать endpoint для метрик
- Текущий способ запуска Cernan: DaemonSet / Sidecar

Опыт: Логирование

- Требование: отсутствие дополнительных действий от пользователя, печать в stdout/stderr
- Cernan -> ElasticSearch
- k8s события (events) -> ElasticSearch с помощью Heapster [17]
- Kibana
- Смотреть логи через Stern!! [18]

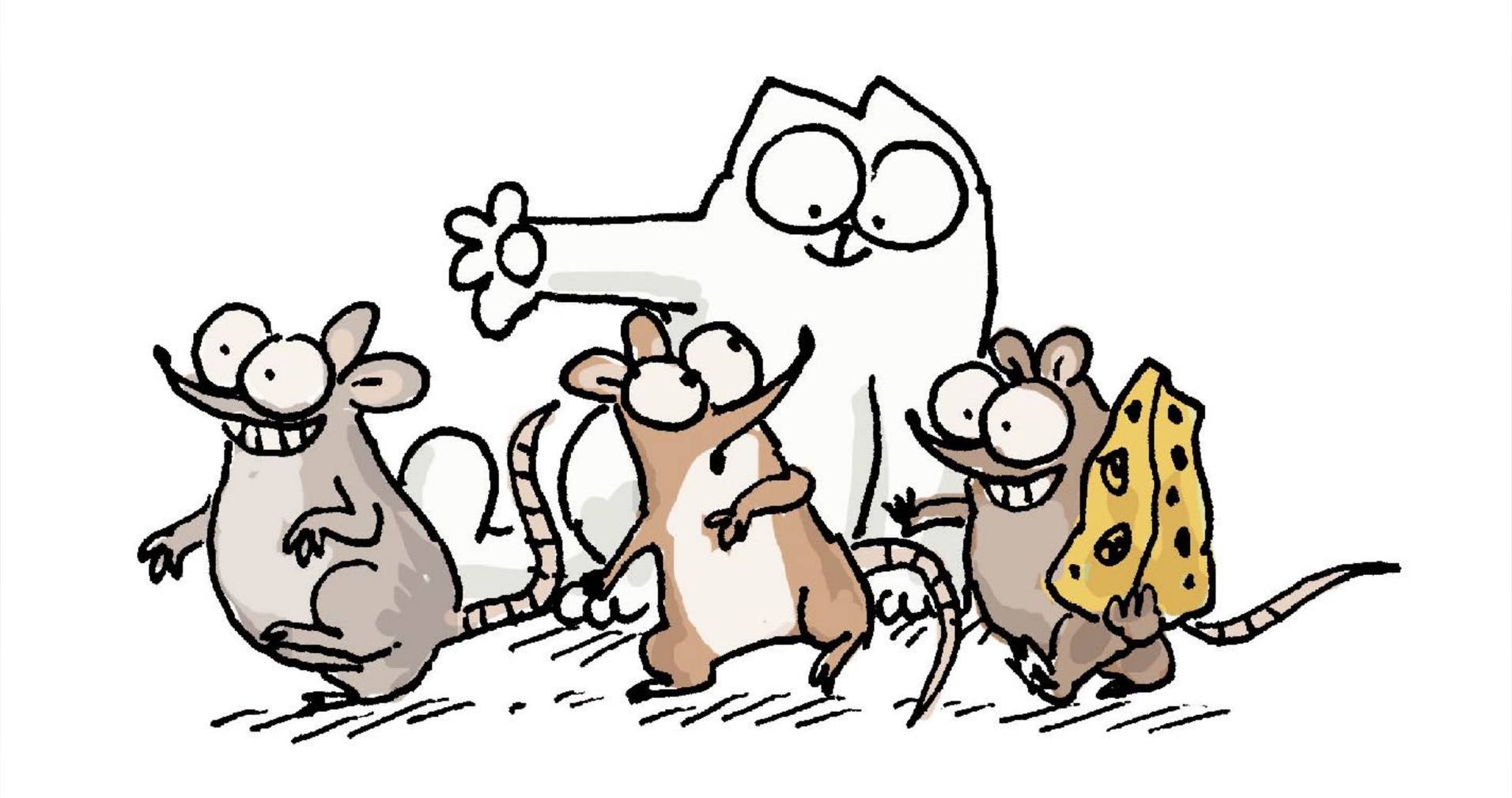
Опыт: Логирование

- Требование: отсутствие дополнительных действий от пользователя, печать в stdout/stderr
- Cernan -> ElasticSearch
- k8s события (events) -> ElasticSearch с помощью Heapster
- Kibana
- Смотреть логи через Stern!! [18]

Опыт: Секреты

- Система ведения секретов досталась в наследство (S3 + KMS)
- K8s имеет поддержку секретов в состоянии “alpha” в 1.7
- Vault?

Опыт: Разработка в k8s



Опыт: Ад разработчика

- “Docker-way”
- Компромиссы:
  - Создавать образ в CI Pipeline <-> Обновлять образ FS на лету
  - Запускать тесты локально <-> Делать коммит, чтобы CI проверил
  - Запускать окружение <-> Эмулировать (mock) окружение

Опыт: Ад разработчика

- “Docker-way”
- Компромиссы:
  - Создавать образ в CI Pipeline <-> Обновлять
  - Запускать тесты локально <-> Делать коммит,
  - Запускать окружение <-> Эмулировать (mock)

Опыт: Инструменты разработчика

- Deis: Очень прост в использовании (Heroku-style), не сможет поддержать все более сложные случаи
- Helm Charts: простой, понятный. Нужны дополнительные инструменты поверх (ksonnet?)
- Нужна хорошая документация + организованная структура процесса (HowTo-s, FAQ-s, “быстрое введение”, etc)
- CLI инструменты: kubectl / helm / drone / docker / stern / minikube

Опыт: Minikube

- Позволяет запускать k8s локально
- VirtualBox или xhyve [20] для MacOS для запуска docker
- Баги, например не работает DNS для xhyve (dnsmasq?) [24], мы рекомендуем VirtualBox
- Нелепа сама идея запуска виртуального окружения для запуска докера для запуска маленького образа!

Опыт: Minikube

- Позволяет запускать k8s локально
- VirtualBox или xhyve [20] для MacOS для запуска docker
- Баги, например не работает DNS для xhyve (dnsmasq?) [24], мы рекомендуем VirtualBox
- Нелепа сама идея запуска виртуального окружения для запуска докера для запуска маленького образа!

CI

- Concourse: богатая функциональность, нестабилен, плохая документация, сложности при переходе на новые версии.
- Drone.io: маленький, простой, отлично подошел [21]
- Каждая стадия CI пайплайна работает в Docker
- Не всегда хватает документации, но есть же код!
- У нас уже работает выкатка в admin/stage кластера
- Система плагинов

Drone config

```
pipeline:
  build-release:
    image: *base_image_versioned
    commands:
      - make rel
      - mv rel/phonemasking/releases/*/phonemasking.tar.gz .

  test-docker-image:
    image: plugins/docker
    registry: quay.io
    dockerfile: Dockerfile.test
    repo: *test_image
    tags:
      - ${DRONE_BRANCH}-${DRONE_BUILD_NUMBER}
    secrets:
      - docker_username
      - docker_password, docker_email

  unit-test:
    image: *test_image_versioned
    commands:
      - bash -x ci/unittest-in-docker.sh

  integration-test:
    image: *test_image_versioned
    commands:
      - bash ci/behave-in-docker.sh

  prod-docker-image:
    image: plugins/docker
    registry: quay.io
    when:
      branch: [master]
      event: [push]
    repo: *prod_image
    tags:
      - ${DRONE_BRANCH}
    secrets:
      - docker_username
      - docker_password, docker_email
```

Drone config

```
pipeline:
  build-release:
    image: *base_image_versioned
    commands:
      - make rel
      - mv rel/phonemasking/releases/*/phonemasking.tar.gz .

  test-docker-image:
    image: plugins/docker
    registry: quay.io
    dockerfile: Dockerfile.test
    repo: *test_image
    tags:
      - ${DRONE_BRANCH}-${DRONE_BUILD_NUMBER}
    secrets:
      - docker_username
      - docker_password, docker_email

  unit-test:
    image: *test_image_versioned
    commands:
      - bash -x ci/unittest-in-docker.sh
```

build code

build test docker image

run unit tests

```
  integration-test:
    image: *test_image_versioned
    commands:
      - bash ci/behave-in-docker.sh

  prod-docker-image:
    image: plugins/docker
    registry: quay.io
    when:
      branch: [master]
      event: [push]
    repo: *prod_image
    tags:
      - ${DRONE_BRANCH}
    secrets:
      - docker_username
      - docker_password, docker_email
```

run integration tests

build production image

Drone config

```
pipeline:
  build-release:
    image: *base_image_versioned
    commands:
      - make rel
      - mv rel/phonemasking/releases/*/phonemasking.tar.gz .

  test-docker-image:
    image: plugins/docker
    registry: quay.io
    dockerfile: Dockerfile.test
    repo: *test_image
    tags:
      - ${DRONE_BRANCH}-${DRONE_BUILD_NUMBER}
    secrets:
      - docker_username
      - docker_password, docker_email

  unit-test:
    image: *test_image_versioned
    commands:
      - bash -x ci/unittest-in-docker.sh
```

build code

build test docker image

run unit tests

```
  integration-test:
    image: *test_image_versioned
    commands:
      - bash ci/behave-in-docker.sh

  prod-docker-image:
    image: plugins/docker
    registry: quay.io
    when:
      branch: [master]
      event: [push]
    repo: *prod_image
    tags:
      - ${DRONE_BRANCH}
    secrets:
      - docker_username
      - docker_password, docker_email
```

run integration tests

build production image

More features:

- matrix-conditions
- volumes
- managed secrets
- Vault support
- Github / Email / Slack / Hipchat integration

Drone config

```
pipeline:
  build-release:
    image: *base_image_versioned
    commands:
      - make rel
      - mv rel/phonemasking/releases/*/phonemasking.tar.gz .

  test-docker-image:
    image: plugins/docker
    registry: quay.io
    dockerfile: Dockerfile.test
    repo: *test_image
    tags:
      - ${DRONE_BRANCH}-${DRONE_BUILD_NUMBER}
    secrets:
      - docker_username
      - docker_password, docker_email

  unit-test:
    image: *test_image_versioned
    commands:
      - bash -x ci/unittest-in-docker.sh
```

build code

build test docker image

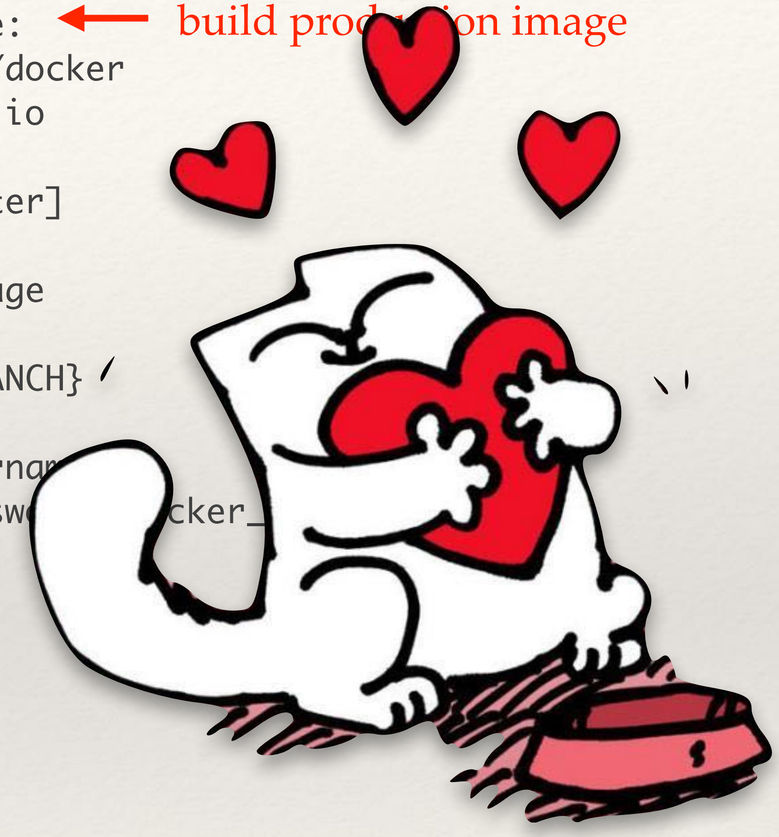
run unit tests

```
  integration-test:
    image: *test_image_versioned
    commands:
      - bash ci/behave-in-docker.sh

  prod-docker-image:
    image: plugins/docker
    registry: quay.io
    when:
      branch: [master]
      event: [push]
    repo: *prod_image
    tags:
      - ${DRONE_BRANCH}
    secrets:
      - docker_usernar
      - docker_passw      cker_
```

run integration tests

build prod ion image



More features:

- matrix-conditions
- volumes
- managed secrets
- Vault support
- Github / Email / Slack / Hipchat integration

Оценка Google

- “Best Practices” (Лучшие практики)
- Несколько кластеров - правильный выбор в данный момент
- Идея “один большой кластер” [22] пока неприменима
- Нужна поддержка Geo location в кластере (роутинг, LB, глобальный раутер, etc)
- Service Mesh может помочь

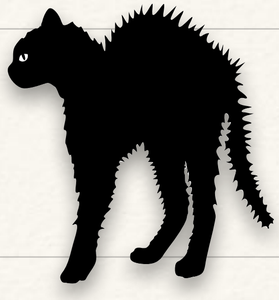
Опыт: Geodesic [11]

- Open-source версия нашего подхода
- Очень похожий дизайн и выбор концепций
- Очень хорошая и ясная имплементация

Текущие проблемы

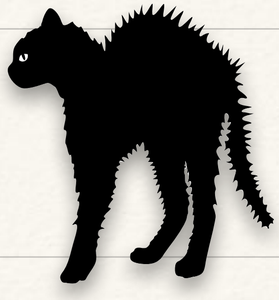
- У нас были проблемы с локальными .k8s (/.tld) именами
- Большие образы - медленная загрузка
- Большие образы - медленная работа (FS?)
- У нас уже три разных способа установки разных расширений k8s :)

Итого: Минусы, часть 1



- Очень большой объем информации на изучение (не только новые технологии, но и новые концепции и привычки)
- Kubernetes будет лишь [маленькой] частью вашей инфраструктуры
- Некоторые приложения в принципе сложно запускать на k8s, чем в “классическом” варианте
- Очень многословные конфигурационные файлы (k8s/helm)
- Все текущие решения на рынке - сырые

Итого: Минусы, часть 2



- Все “менеджеры пакетов” далеки от совершенства
- K8s + Docker-way полностью поменяет ваши привычки разработки
- Очень много компромиссов, делать выбор сложно, это создает негатив и отторжение
- Чтобы отладить работу на k8s локально придется хорошо постараться

Итого: Плюсы 1

- С течением времени все меньше придется учить новичков k8s
- Короткий цикл релиза. Commit -> CI -> Image -> Нажал кнопку для релиза
- Разделение инфраструктурного и бизнес кода
- Большое и очень активное сообщество, много изменений (kops, helm, stern, geodesic, rbac, autoscaler, … - за 2 последних года)
- Логи приложения и k8s событий в одном месте (Kibana)
- Нет прямого доступа на ноды

Итого: Плюсы 2

- Не требуется поддержка или помощь @infra команды для релизов или решения проблем в продакшене
- @infra команда перестает быть узким местом
- Легко добавить Error Budget (у нас это уже работает)
- @infra команда перестает быть “Полицией Нравов”, у всех общая цель
- Сильно уменьшает напряженность отношений в комании

Ссылки: часть 1

1. Continuous Delivery by Jez Humble, David Farley.
   https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912
2. Containers are not VMs.
   https://blog.docker.com/2016/03/containers-are-not-vms/
3. Docker Distribution (image registry).
   https://github.com/docker/distribution
4. Quay - Image Registry as a Service.
   https://quay.io
5. etcd-operator - Manager of etcd cluster atop Kubernetes.
   https://github.com/coreos/etcd-operator
6. Dex - OpenID Connect Identity (OIDC) and OAuth 2.0 Provider with Pluggable Connectors.
   https://github.com/coreos/dex
7. awsudo - sudo-like utility to manage AWS credentials.
   https://github.com/makethunder/awsudo
8. Autoscaling-related components for Kubernetes.
   https://github.com/kubernetes/autoscaler/
9. Simon's cat
   https://simonscat.com/
10. Helm: Kubernetes package manager
    https://helm.sh/
11. Geodesic: framework to create your own cloud platform
    https://github.com/cloudposse/geodesic
12. Calico: Configuring IP-in-IP
    https://docs.projectcalico.org/v2.2/usage/configuration/ip-in-ip

Ссылки: часть 2

13. Sensu: Open-core monitoring platform
    https://github.com/sensu/sensu
14. InfluxDB: Scalable datastore for metrics, events, and real-time analytics
    https://github.com/influxdata/influxdb
15. Cernan: Telemetry and logging aggregation server
    https://github.com/postmates/cernan
16. Prometheus: Open Source monitoring solution
    https://prometheus.io/
17. Heapster: Compute Resource Usage Analysis and Monitoring of Container Clusters
    https://github.com/kubernetes/heapster
18. Stern: Multi pod and container log tailing for Kubernetes
    https://github.com/wercker/stern
19. Minikube: tool to run Kubernetes locally
    https://github.com/kubernetes/minikube
20. Docker machine driver fox xhyve native OS X Hypervisor
    https://github.com/zchee/docker-machine-driver-xhyve
21. Drone: Continuous Delivery platform built on Docker, written in Go
    https://github.com/drone/drone
22. Borg, Omega and Kubernetes
    http://queue.acm.org/detail.cfm?id=2898444
23. Container-Native Networking - Comparison
    https://docs.google.com/spreadsheets/d/1polIS2pvjOxCZ7hpXbra68CluwOZybsP1IYfr-HrAXc/edit#gid=0
24. Bug in minikube when working with xhyve driver
    https://github.com/kubernetes/minikube/issues/1442

Вопросы?

- Иван Глушков
- ivan.glushkov@gmail.com
- @gliush