Terraform vs Helm

Битва за инфраструктуру

Дмитрий Губенко, КУРС

Имя

Дмитрий Губенко, КУРС

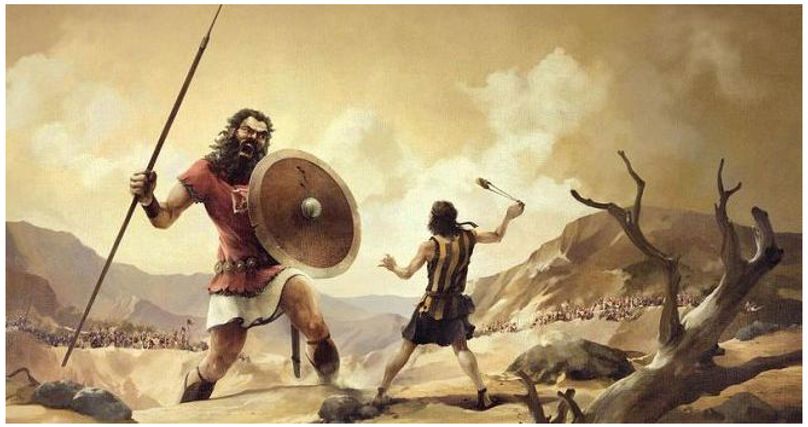
Почта

dmeatriy@gmail.com

Telegram

@Dmeatriy

Когда каждый может обидеть



Что имеем на входе?

Что имеем на входе?

- Облачный провайдер - DigitalOcean

Что имеем на входе?

- Облачный провайдер - DigitalOcean
- Terraform
  - ~1000 строк провайдер-ориентированного кода
  - ~100 виртуальных машин
  - ~20 dns-записей

Что имеем на входе?

- Облачный провайдер - DigitalOcean
- Terraform
  - ~1000 строк провайдер-ориентированного кода
  - ~100 виртуальных машин
  - ~20 dns-записей
- Ansible
  - ~5000 строк кода
  - ~20 плейбуков с пачкой ролей

Что болит?

Что болит?

- Нет быстрого способа переехать к другому провайдеру.

Что болит?

- Нет быстрого способа переехать к другому провайдеру.
- МНОГО кода для деплоя

Что болит?

- Нет быстрого способа переехать к другому провайдеру.
- МНОГО кода для деплоя
- Отвратительное масштабирование системы.

Что болит?

- Нет быстрого способа переехать к другому провайдеру.
- МНОГО кода для деплоя
- Отвратительное масштабирование системы.
- Деплой тоже требует дебага.

Что болит?

- Нет быстрого способа переехать к другому провайдеру.
- МНОГО кода для деплоя
- Отвратительное масштабирование системы.
- Деплой тоже требует дебага.
- Кто ходил на мою ноду?

Оставь меня, я задержу их.

```
resource "digitalocean_droplet" "pgdb-dev" {
    image = "31754481"
    name = "pgdb-dev"
    region = "ams2"
    size = "s-2vcpu-2gb"
    private_networking = "true"
    resize_disk = "false"
    ssh_keys = ["${var.ssh_keys}"]
    user_data = "${file("minimal.conf")}"
}
```

Оставь меня, я задержу их.

```
resource "digitalocean_droplet" "pgdb-dev" {
    image = "31754481"
    name = "pgdb-dev"
    region = "ams2"
    size = "s-2vcpu-2gb"
    private_networking = "true"
    resize_disk = "false"
    ssh_keys = ["${var.ssh_keys}"]
    user_data = "${file("minimal.conf")}"
}
```

- Ах, если бы только ноды.

Посыпьте их пеплом

```
variable "eps_names" { default = ["epsilon400",
                                  "epsilon401",
                                  "epsilon402",
                                  "epsilon403",
                                  "epsilon404",
                                  "epsilon405"] }
```

Посыпьте их пеплом

```
variable "eps_names" { default = ["epsilon400",
                                  "epsilon401",
                                  "epsilon402",
                                  "epsilon403",
                                  "epsilon404",
                                  "epsilon405"] }
```

- Когда в твою инфраструктуру попадает цикл.

Ansible-fu?

Ansible-fu?

- Управление inventory для ansible

Ansible-fu?

- Управление inventory для ansible
- Соблюдение идемпотентности.

Ansible-fu?

- Управление inventory для ansible
- Соблюдение идемпотентности.
- Конфигурационный ад.

Ansible-fu?

- Управление inventory для ansible
- Соблюдение идемпотентности.
- Конфигурационный ад.
- Управление сервисами при обновлении.

Ansible-fu?

- Управление inventory для ansible
- Соблюдение идемпотентности.
- Конфигурационный ад.
- Управление сервисами при обновлении.
- Зависимость от операционной системы.

Ansible-fu?

- Управление inventory для ansible
- Соблюдение идемпотентности.
- Конфигурационный ад.
- Управление сервисами при обновлении.
- Зависимость от операционной системы.
- Создание внутренней сети с настройкой ip-tables

Когда каждый ansible-playbook - произведение искусства.

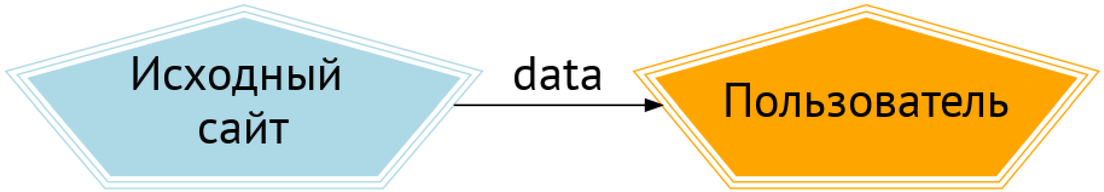
```
---
- name: configure backoffice server
... 10 more lines
  roles:
    - role: do_hostname
... 20 more lines
    - role: backoffice
  tasks:
... 50 more lines
    - name: Reload nginx
      service:
        name: nginx
        state: reloaded
```

Когда каждый ansible-playbook - произведение искусства.

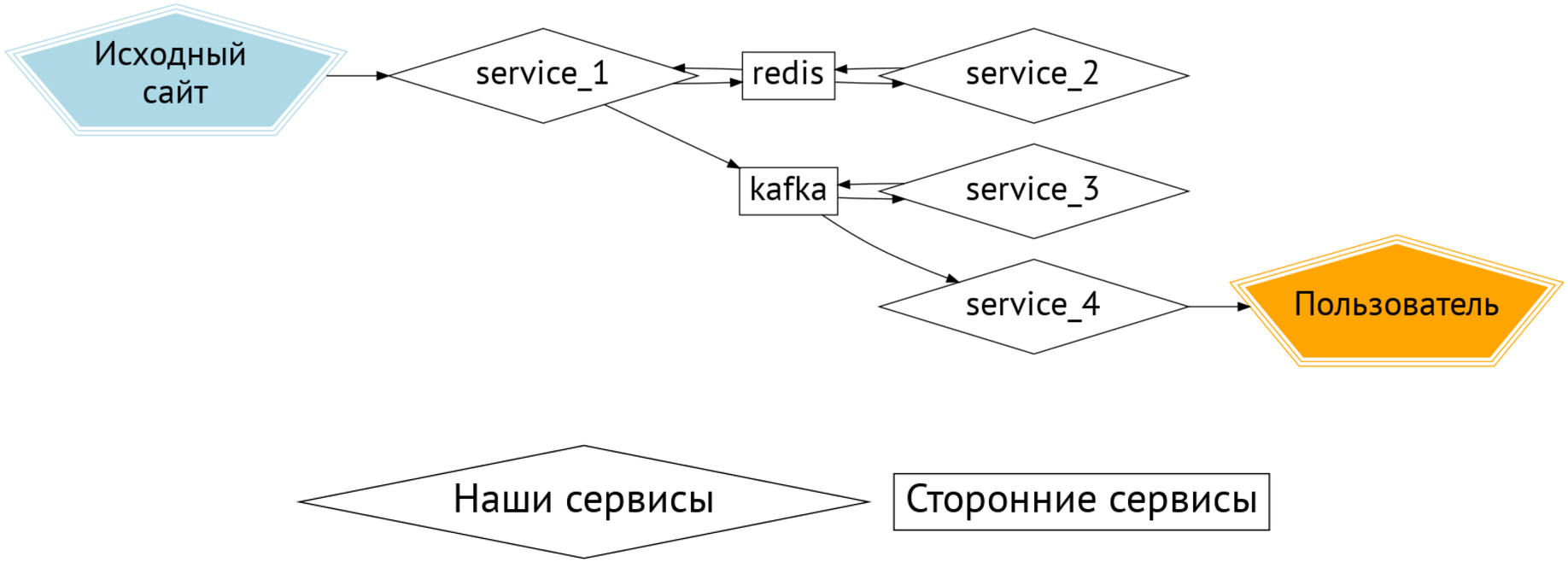
```
---
- name: configure backoffice server
... 10 more lines
  roles:
    - role: do_hostname
... 20 more lines
    - role: backoffice
  tasks:
... 50 more lines
    - name: Reload nginx
      service:
        name: nginx
        state: reloaded
```

- Долго, дорого, будет цениться после смерти автора(нет).

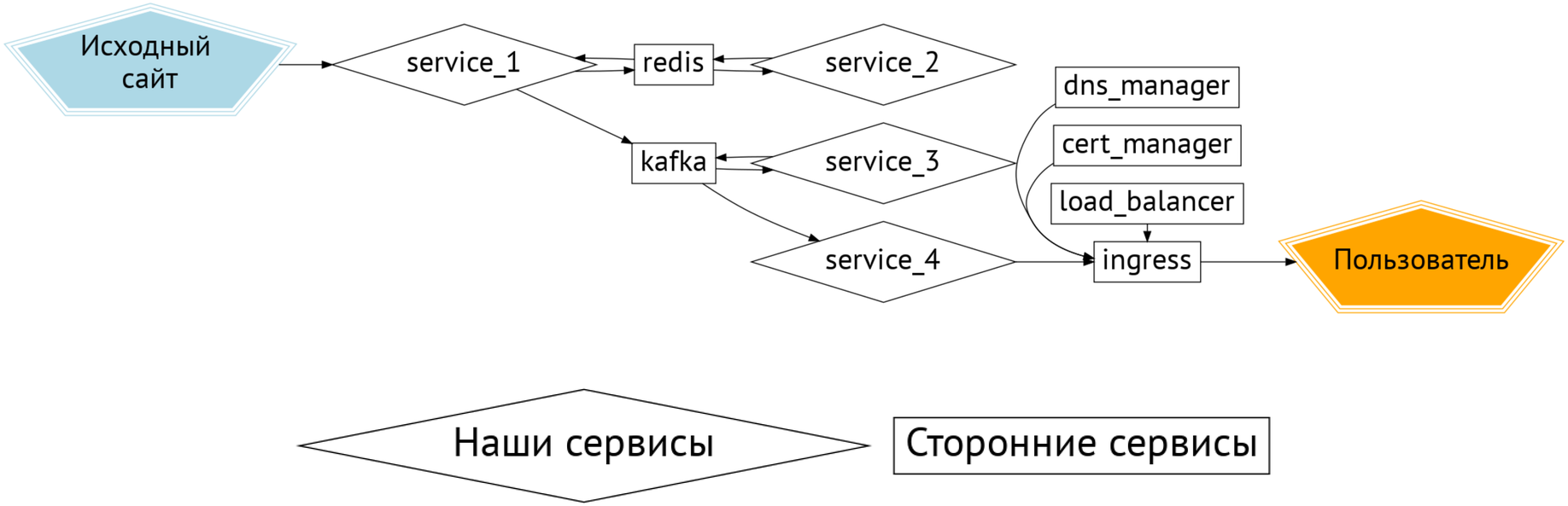
А что, собственно, деплоим?



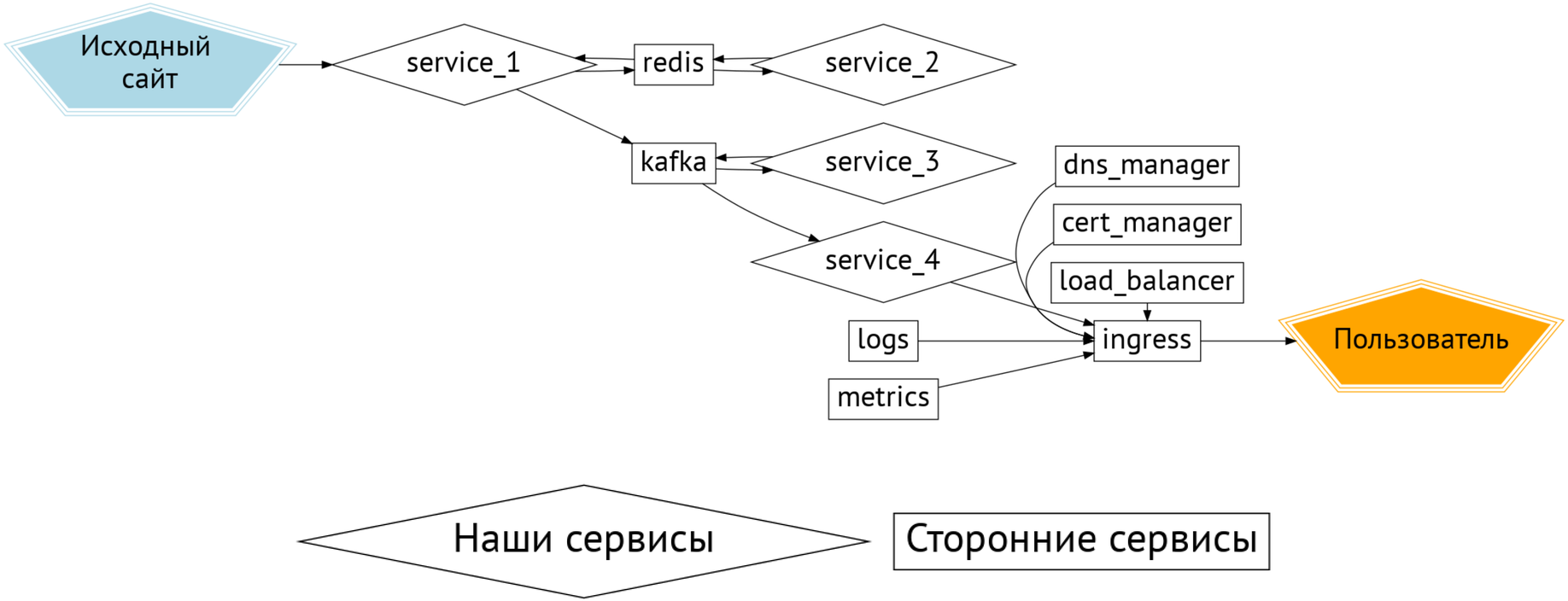
А что, собственно, деплоим?



А что, собственно, деплоим?

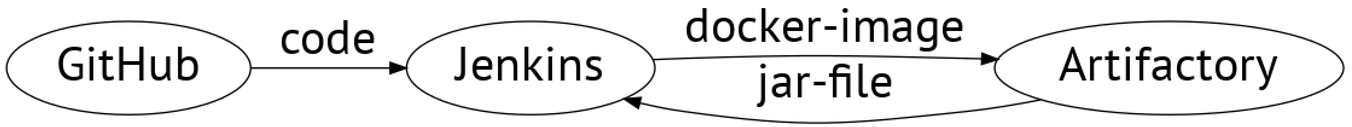


А что, собственно, деплоим?



Сборка jar-file

Сборка docker-образа



Я рассажу вас по стульчикам

До

После

Почему docker/kubernetes?

Почему docker/kubernetes?

- Урезать рацион

Почему docker/kubernetes?

- Урезать рацион
- Легко масштабировать

Почему docker/kubernetes?

- Урезать рацион
- Легко масштабировать
- Упал - перезапустился

Почему docker/kubernetes?

- Урезать рацион
- Легко масштабировать
- Упал - перезапустился
- Изоляция окружения

Почему docker/kubernetes?

- Урезать рацион
- Легко масштабировать
- Упал - перезапустился
- Изоляция окружения
- Собственная межсервисная сеть и dns

Почему ОБЛАЧНЫЙ kubernetes?

Почему ОБЛАЧНЫЙ kubernetes?

- Легко свернуть-развернуть

Почему ОБЛАЧНЫЙ kubernetes?

- Легко свернуть-развернуть
- K8saaS - это недорого

Почему ОБЛАЧНЫЙ kubernetes?

- Легко свернуть-развернуть
- K8saaS - это недорого
- У Terraform есть провайдеры, поэтому легко жить с IaC

Упаковка helm-чарта

Почему helm?

Почему helm?

- Пакетный менеджер для сервисов k8s

Почему helm?

- Пакетный менеджер для сервисов k8s
- Cloud Native Computing Foundation

Почему helm?

- Пакетный менеджер для сервисов k8s
- Cloud Native Computing Foundation
- Большое количество готовых пакетов для классического web

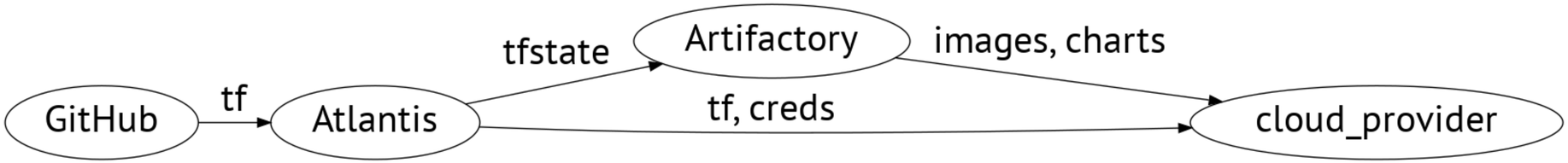
Почему helm?

- Пакетный менеджер для сервисов k8s
- Cloud Native Computing Foundation
- Большое количество готовых пакетов для классического web
- Скоро будет Helm 3 без детских болезней(уже есть альфа)

Почему helm?

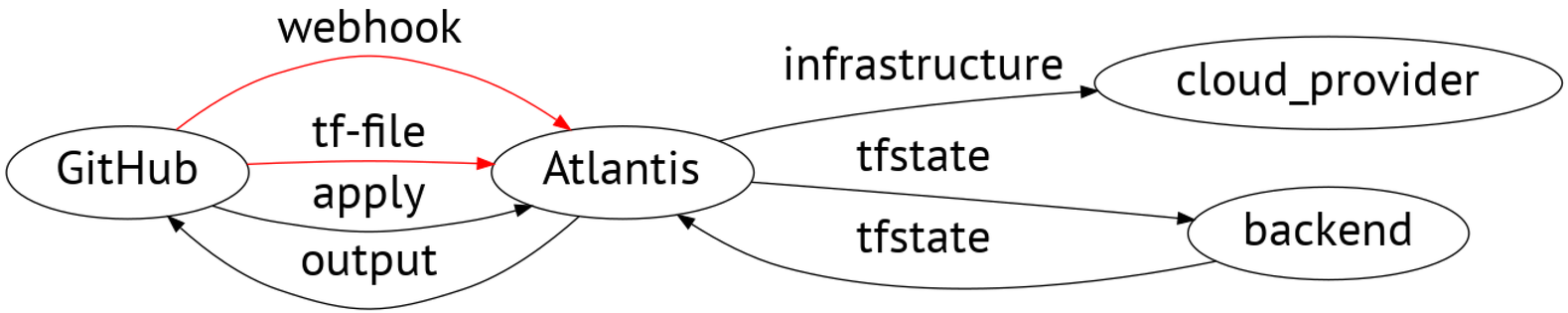
- Пакетный менеджер для сервисов k8s
- Cloud Native Computing Foundation
- Большое количество готовых пакетов для классического web
- Скоро будет Helm 3 без детских болезней(уже есть альфа)
- Достаточно выбрать другой namespace, и будет dev, test, staging, prod

Инфраструктура тоже здесь

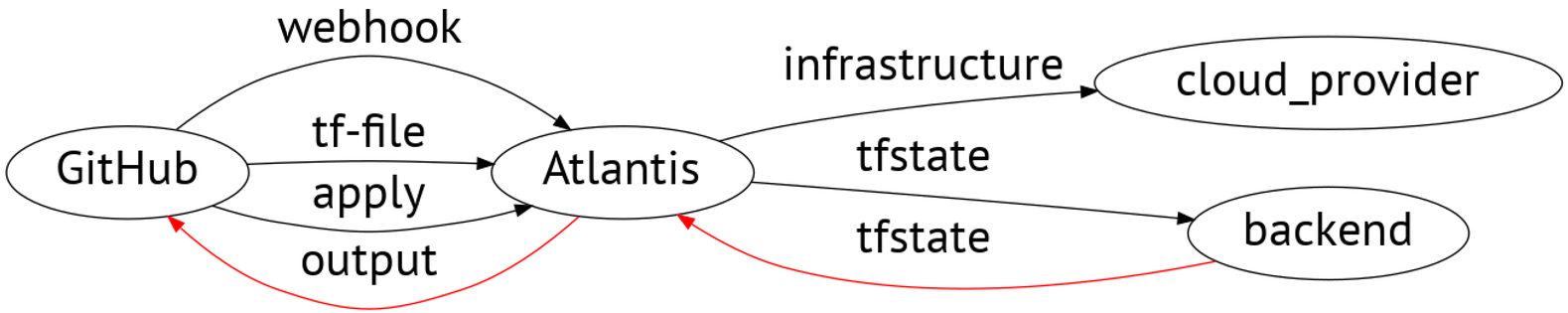


Atlantis - IaC автоматизация прямо из VCS

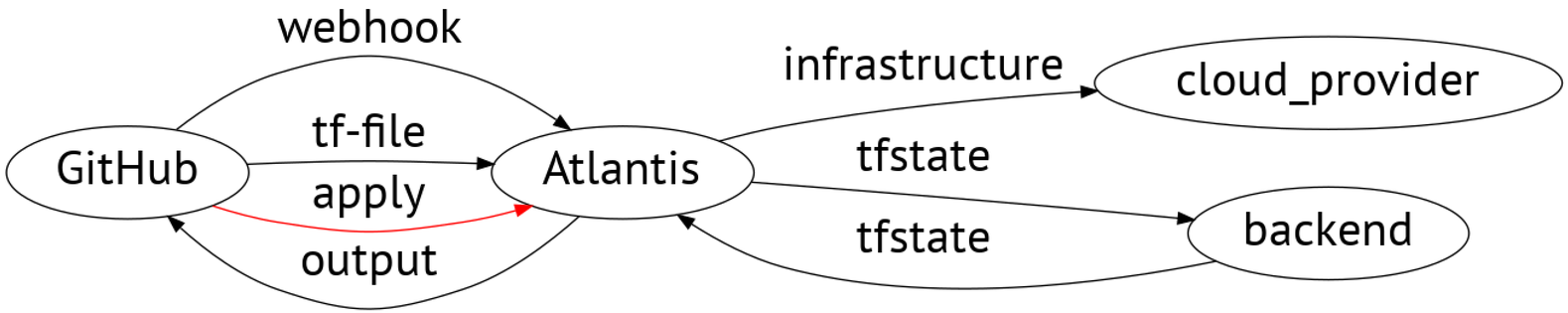
Некосмический корабль



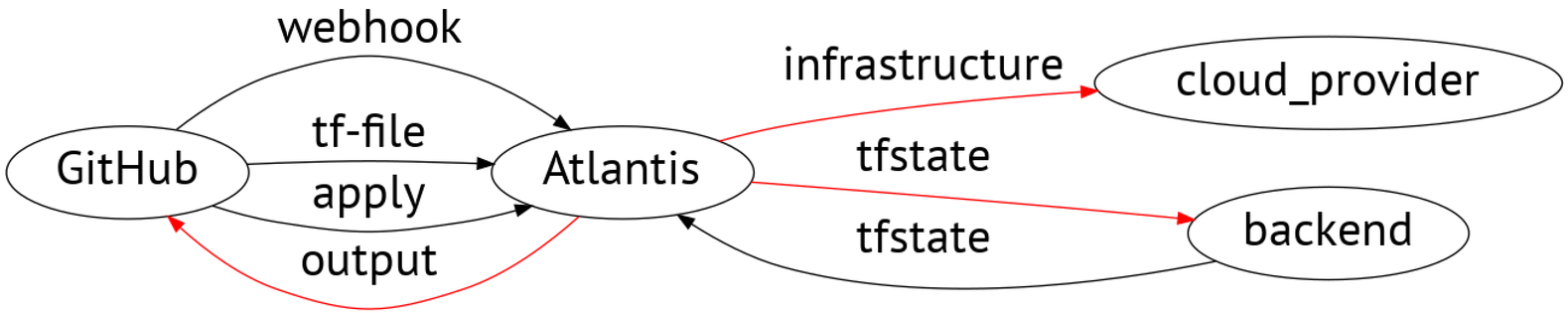
Некосмический корабль



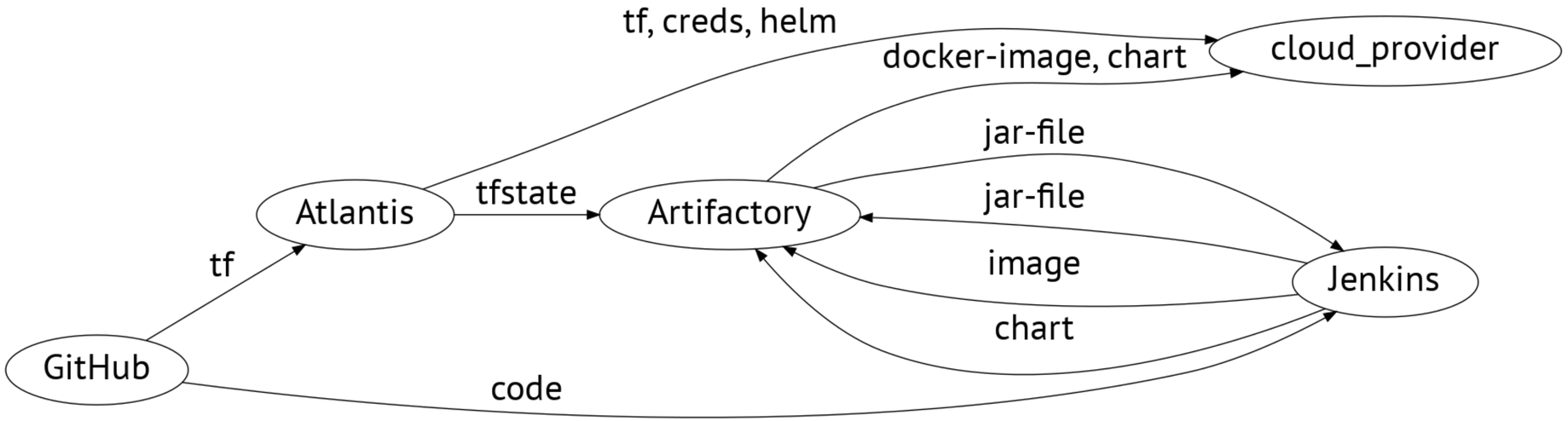
Некосмический корабль



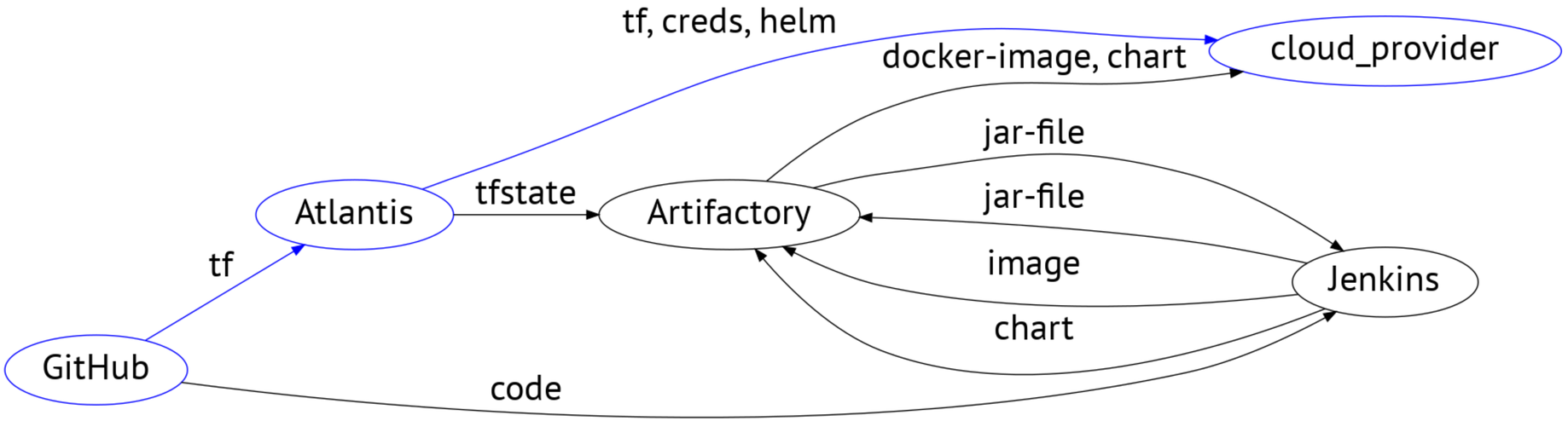
Некосмический корабль



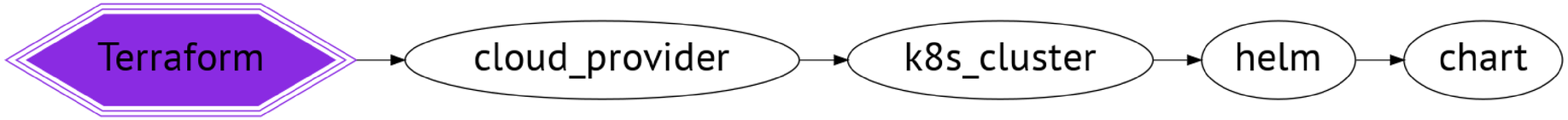
Не будем отвлекаться

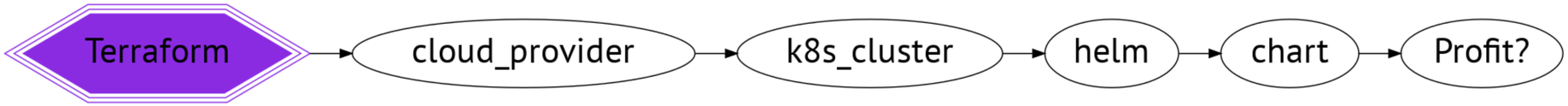


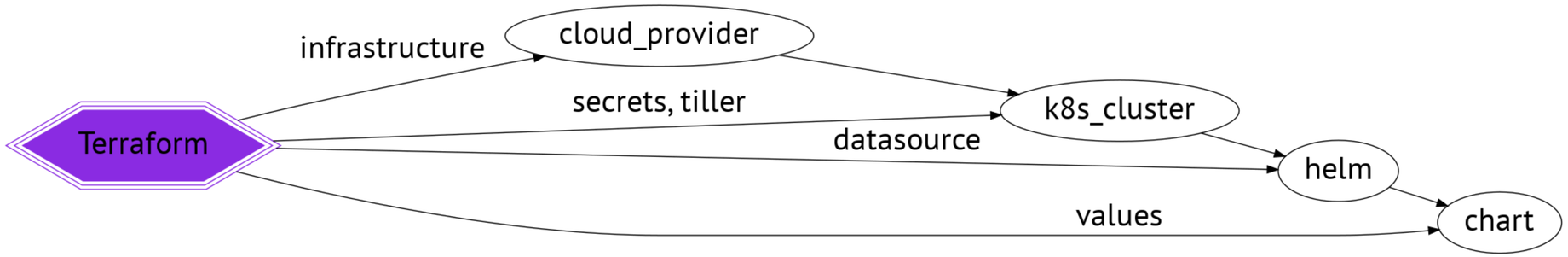
Не будем отвлекаться




Terraform


Terraform
cloud_provider
k8s_cluster
helm
chart


Terraform
cloud_provider
k8s_cluster
helm
chart
Profit?


Terraform
infrastructure
cloud_provider
secrets, tiller
k8s_cluster
datasource
helm
values
chart

Всё хорошо. Почти.

Всё хорошо. Почти.

- Правая рука не знает, что делает левая.

Всё хорошо. Почти.

- Правая рука не знает, что делает левая.
- Для каждого изменения - терраформ.

Terraform - это хорошо.

Из пушки по воробьям?

Из пушки по воробьям?

- Добавить/убавить ноду в пул.

Из пушки по воробьям?

- Добавить/убавить ноду в пул.
- Добавить/убавить поду в сервис.

Из пушки по воробьям?

- Добавить/убавить ноду в пул.
- Добавить/убавить поду в сервис.
- Пробросить dns-запись.

Из пушки по воробьям?

- Добавить/убавить ноду в пул.
- Добавить/убавить поду в сервис.
- Пробросить dns-запись.
- Настроить load-balancing.

Из пушки по воробьям?

- Добавить/убавить ноду в пул.
- Добавить/убавить поду в сервис.
- Пробросить dns-запись.
- Настроить load-balancing.
- Добавить https.

Helm как способ управления инфраструктурой

Какие есть инструменты?

Какие есть инструменты?

- Cluster-autoscaler

Какие есть инструменты?

- Cluster-autoscaler
- Cluster-overprovisioner

Какие есть инструменты?

- Cluster-autoscaler
- Cluster-overprovisioner
- External-dns

Какие есть инструменты?

- Cluster-autoscaler
- Cluster-overprovisioner
- External-dns
- Nginx-ingress

Какие есть инструменты?

- Cluster-autoscaler
- Cluster-overprovisioner
- External-dns
- Nginx-ingress
- Cert-manager

Какие есть инструменты?

- Cluster-autoscaler
- Cluster-overprovisioner
- External-dns
- Nginx-ingress
- Cert-manager
- HPA

Cluster-autoscaler

Cluster-autoscaler

- Manages autoscaling groups

Cluster-autoscaler

- Manages autoscaling groups
- AWS, GCE, Azure AKS

Cluster-autoscaler

- Manages autoscaling groups
- AWS, GCE, Azure AKS
- Autodiscovery by ASG tags

Cluster-autoscaler

- Manages autoscaling groups
- AWS, GCE, Azure AKS
- Autodiscovery by ASG tags
- Manually specify ASG(aws only!)

Cluster-autoscaler - IAM permissions

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "autoscaling:DescribeAutoScalingGroups",
              "autoscaling:DescribeAutoScalingInstances",
              "autoscaling:DescribeLaunchConfigurations",
              "autoscaling:SetDesiredCapacity",
              "autoscaling:TerminateInstanceInAutoScalingGroup"
          ],
          "Resource": "*"
      }
  ]
}
```

Cluster-autoscaler - values.yml

```
cluster-autoscaler:
  autoscalingGroups:
    - name: asg1
      maxSize: 1
      minSize: 10
  awsRegion: us-east-1
```

Cluster-autoscaler - values.yml

```
cluster-autoscaler:
  autoscalingGroups:
    - name: asg1
      maxSize: 1
      minSize: 10
  awsRegion: us-east-1
```

- Можно организовать autoDiscovery, но придется тогда создавать asg через терраформ.

Cluster-overprovisioner

Cluster-overprovisioner

- Holds resources for service pods

Cluster-overprovisioner

- Holds resources for service pods
- HOT autoscaling

Cluster-overprovisioner

- Holds resources for service pods
- HOT autoscaling
- Low PriorityClass

Cluster-overprovisioner

```
helm install stable/cluster-overprovisioner
```

External-dns

External-dns

- Make services discoverable

External-dns

- Make services discoverable
- AWS Route 53, GC DNS stable

External-dns

- Make services discoverable
- AWS Route 53, GC DNS stable
- AWS SD, AzureDNS, CloudFlare beta

External-dns

- Make services discoverable
- AWS Route 53, GC DNS stable
- AWS SD, AzureDNS, CloudFlare beta
- 15+ DNS alpha

External-dns IAM Permissions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:ListHostedZones",
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "*"
    }
  ]
}
```

External-dns IAM Permissions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:ListHostedZones",
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "*"
    }
  ]
}
```

- Можно отталкиваться от минимальных прав, создавая hosted-zone через terraform.

External-dns nginx-ingress values.yaml

```
controller:
    ingressClass: elb
    service:
      annotations:
        external-dns.alpha.kubernetes.io/hostname: service1.example.com
```

Nginx-ingress

Nginx-ingress

- Load-balancing by Kubernetes API

Nginx-ingress

- Load-balancing by Kubernetes API
- Providers
  - AWS ELB
  - DO
  - Google Cloud

Nginx-ingress

- Load-balancing by Kubernetes API
- Providers
  - AWS ELB
  - DO
  - Google Cloud
- Has default-backend

Cert-manager

Cert-manager

- Provision and manage TLS certificates

Cert-manager

- Provision and manage TLS certificates
- Maintainer - Jetstack

Cert-manager

- Provision and manage TLS certificates
- Maintainer - Jetstack
- Pre-1.0

Cert-manager

- Provision and manage TLS certificates
- Maintainer - Jetstack
- Pre-1.0
- Issuers
  - CA
  - Self-Signed
  - ACME
  - Vault
  - Venafi

Cert-manager installation

```
kubectl apply --validate=false -f https://raw.githubusercontent.com/jetstack
kubectl create namespace cert-manager
helm repo add jetstack https://charts.jetstack.io
helm repo update
helm install \
  --name cert-manager \
  --namespace cert-manager \
  --version v0.11.0 \
  jetstack/cert-manager
```

Chart annotations for ingress/tls:

```
grafana:
  ingress:
    enabled: true
    hosts:
      - grafana.example.com
    annotations:
      kubernetes.io/ingress.class: elb
      ingress.kubernetes.io/ssl-redirect: "true"
      kubernetes.io/tls-acme: "true"
      certmanager.k8s.io/issuer: letsencrypt-staging
    tls:
      - secretName: grafana-tls
        hosts:
          - grafana.example.com
```

Issuer for cert-manager

```
apiVersion: cert-manager.io/v1alpha2
kind: ClusterIssuer
metadata:
  name: letsencrypt-staging
spec:
  acme:
    email: user@example.com
    server: https://acme-staging-v02.api.letsencrypt.org/directory
    privateKeySecretRef:
      name: example-issuer-account-key
    solvers:
    - http01:
        ingress:
          class: elb
```

Certificate for cert-manager

```
apiVersion: cert-manager.io/v1alpha2
kind: Certificate
metadata:
  name: example-com
  namespace: default
spec:
  secretName: example-com-tls
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  commonName: example.com
  dnsNames:
  - example.com
  - www.example.com
  uriSANs:
  - spiffe://cluster.local/ns/sandbox/sa/example
  issuerRef:
    name: letsencrypt-staging
    kind: ClusterIssuer
```

Terraform как способ инициализации.

В Terraform остаются:

В Terraform остаются:

- Заказ кластера и (опционально) node-pool

В Terraform остаются:

- Заказ кластера и (опционально) node-pool
- Secrets для Kubernetes(credentials для Artifactory)

В Terraform остаются:

- Заказ кластера и (опционально) node-pool
- Secrets для Kubernetes(credentials для Artifactory)
- Инициализация Helm-tiller

В Terraform остаются:

- Заказ кластера и (опционально) node-pool
- Secrets для Kubernetes(credentials для Artifactory)
- Инициализация Helm-tiller
- Первичная установка Helm-chart с указанием namespaces

```
resource "digitalocean_kubernetes_cluster" "melbet2" {
  name    = "melbet"
  region  = "lon1"
  version = "1.14.2-do.0"

  node_pool {
    name       = "puppetheatre-pool"
    size       = "s-4vcpu-8gb"
    node_count = 10
    tags       = ["puppetheatre"]
  }

}
```

Итого

144 строки для DigitalOcean

```
resource "google_container_cluster" "melbet" {
  name                     = "melbet-cluster"
  location                   = "europe-north1-a"
  remove_default_node_pool = true
  initial_node_count = 1
  enable_legacy_abac = true
  provisioner "local-exec" {
    when    = "destroy"
    command = "sleep 90"
  }
}
```

Итого

144 строки для GCP

При переезде из DO в GCP меняется ~20 строк

```
resource "aws_eks_cluster" "tf_eks" {
  name            = "${var.eks_cluster-name}"
  role_arn        = "${aws_iam_role.tf-eks-master.arn}"
  version         = "1.14"

  vpc_config {
    security_group_ids = ["${module.security_groups.tf-eks-master-id}"]
    subnet_ids         = ["${module.subnet_id1}", "${module.subnet_id2}"]
  }

  depends_on = [
    "aws_iam_role_policy_attachment.tf-cluster-AmazonEKSClusterPolicy",
    "aws_iam_role_policy_attachment.tf-cluster-AmazonEKSServicePolicy",
  ]
}
```

Итого

510 строк для AWS(вместе с кастомными IAM)

А как насчет Bare Metal?

А как насчет Bare Metal?

- Гипотетически - Rancher2 имеет свой провайдер в Terraform

При переезде в AWS пришлось потрудиться, но на то он и AWS

Проблемы

Helm tiller

Helm tiller

- Всегда неприятно иметь cluster-admin на своём кластере

Helm tiller

- Всегда неприятно иметь cluster-admin на своём кластере
- Можно поднимать локально, тогда будет пользоваться вашими правами

Helm tiller

- Всегда неприятно иметь cluster-admin на своём кластере
- Можно поднимать локально, тогда будет пользоваться вашими правами
- https://habr.com/ru/company/oleg-bunin/blog/462665 - статья по безопасности Helm, которая может слегка смягчить боль

Helm tiller

- Всегда неприятно иметь cluster-admin на своём кластере
- Можно поднимать локально, тогда будет пользоваться вашими правами
- https://habr.com/ru/company/oleg-bunin/blog/462665 - статья по безопасности Helm, которая может слегка смягчить боль
- С выходом Helm3 станет неактуальным.

Persistence

Persistence

- Грамотный persistence management просто необходим

Persistence

- Грамотный persistence management просто необходим
- Неактуальным не станет

Kafka

Kafka

- Оригинальный helm-chart от Confluent - требует квалификации для варения.

Kafka

- Оригинальный helm-chart от Confluent - требует квалификации для варения.
- Современные библиотеки клиентов - с трудом это переживают.

Kafka

- Оригинальный helm-chart от Confluent - требует квалификации для варения.
- Современные библиотеки клиентов - с трудом это переживают.
- Можно выложить за отдельный load-balancer

Kafka

- Оригинальный helm-chart от Confluent - требует квалификации для варения.
- Современные библиотеки клиентов - с трудом это переживают.
- Можно выложить за отдельный load-balancer
- Купить сервис и забыть.

И вот что мы получим

И вот что мы получим

- Легко переехать к другому провайдеру

И вот что мы получим

- Легко переехать к другому провайдеру
- Мало инфраструктурного кода

И вот что мы получим

- Легко переехать к другому провайдеру
- Мало инфраструктурного кода
- Легко масштабироваться

И вот что мы получим

- Легко переехать к другому провайдеру
- Мало инфраструктурного кода
- Легко масштабироваться
- Легко поддерживать несколько контуров

Что почитать/посмотреть?

Что почитать/посмотреть?

- Флант блог https://habr.com/ru/company/flant

Что почитать/посмотреть?

- Флант блог https://habr.com/ru/company/flant
- Atlantis https://www.runatlantis.io

Что почитать/посмотреть?

- Флант блог https://habr.com/ru/company/flant
- Atlantis https://www.runatlantis.io
- Google блог https://cloud.google.com/blog/products/devops-sre

Выводы

Разгружайте ваш Terraform

Минималистичная инфраструктура с гибким подходом к переезду.

Нагружайте ваш Helm

Пакетный менеджер с легкостью справляется с задачами инфраструктуры в пределах сервиса.

На ваши вопросы ответит

Имя

Дмитрий Губенко, КУРС

Почта

dmeatriy@gmail.com

Telegram

@Dmeatriy