Advanced K8S deployments

Bringing Cloud Native to the Enterprise

BANZAICLOUD

Application deployment challenges

- **Resource management**
  - Scheduling priorities
  - Resource change needs
- **Application/deployment management**
  - Handle dependencies
  - Lifecycle management
  - Orchestration, ordering (init containers)

- **Not a new problem**
  - Jobs, resources, dependencies were handled by Puppet, Chef, Ansible, Terraform, etc

A typical application stack

- **Backend**
- **Frontend**
- **Database**
- **Cache**
- **Management - monitoring, cron, workers, etc**

Translated to K8S

- **Deployments**
- **Services**
- **StatefulSets, ReplicaSets, Headless, PV (claims), etc**
- **Ingress**

Available solutions

- **CoreOS KPM**
  - → Not maintained anymore
- **Kubernetes Helm**
  - → Became the standard
- **Mirantis AppController**
  - → Extends on Helm Rudder
- **ATT Armada**
  - → Very complex scenarios
- **Eneco Landscaper**
  - → Flow driven
- **Ksonnet**
  - → Gains traction
- **Helm 3.0**
  - → Ksonnet + Kompose + Kedge + Kubecfg

Helm

- **Pros**
  - Simple
  - Support dependencies between charts
  - Create, deletes, rollbacks
  - Standard
- **Cons**
  - Go template
  - Dependency management - init containers

- **Components**
  - Helm - client
  - Tiller - server (CRUD on K8S objects)
    - Rudder - gRPC interface, CRUD is dispatched there
    - Same as with Tiller, no changes

Armada

- **Openstack on K8S management**
- **Extremely complex**
- **Simple dependency/ordering management**
- **Python**

Landscaper

- **Blueprint describes the landscape**
- **Git flow based**
  - Agent based
  - Keeps track of who, what, when and why
- **K8S state paradigm**
  - Git repo contains the desired state, reconciles
- **API + CLI**
- **Uses Helm/Tiller**
- **Tries to be more like a CI/CD system**

AppController

- **Started by Mirantis**
- **Basic concept**
  - Rudder based
  - Resource definition
    - N/A - part of deployment graph
  - Dependencies
    - Based on deployment graph
    - Topology aware deployment on the graph
- **Not maintained anymore**

Ksonnet

- **Kubernetes Application Deployment Framework**
  - → Yet another one
- **JSonnet Library**
  - → JSON templating language preferred by the ~~K8S~~ ~~Helm~~ whatever community (Heptio?)
- **Not a new API**
  - → Just better primitives
- **Composability**
  - → Template what you want, not what you don’t
- **Part of Helm 3.0**
  - → Developed by Heptio, Bitnami, ~~Deis~~ (Microsoft), CoreOS

Advanced K8S deployments

Ksonnet Core concepts

Jsonnet

```
local nginxContainer =
  container.new("nginx", "nginx:1.13.0") +
  container.ports(containerPort.newNamed("http", 80));

deployment.new("nginx", 5, nginxContainer, {app: "nginx"})
```

YAML

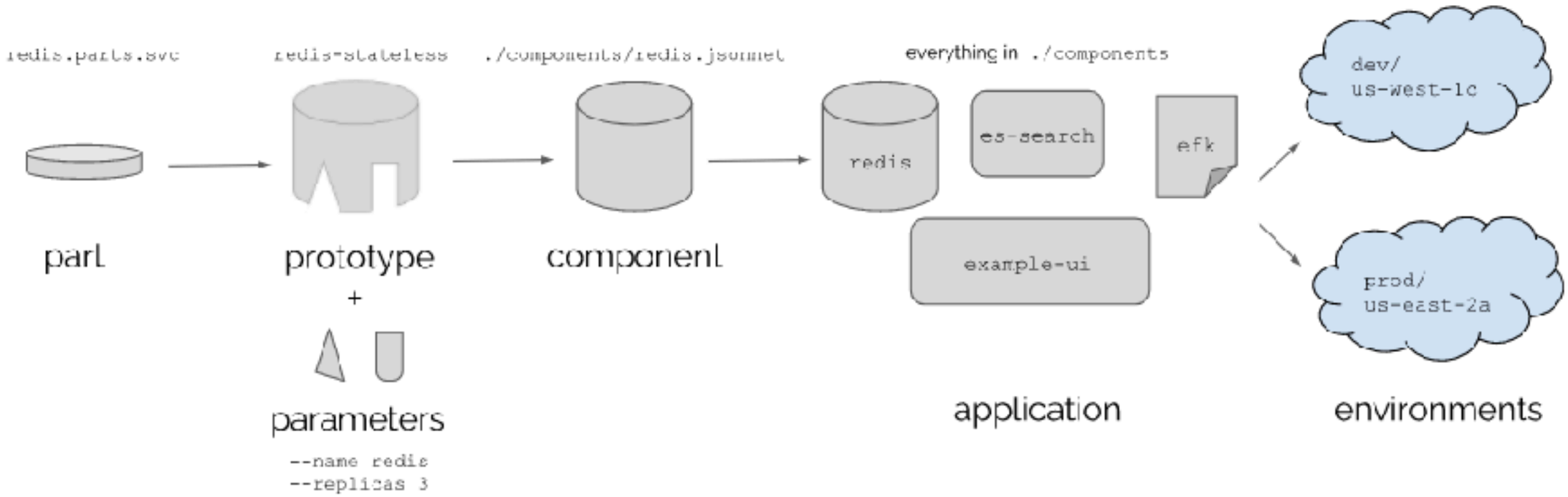
```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 5
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - image: 'nginx:1.13.0'
          name: nginx
          ports:
            - containerPort: 80
              name: http
```

+ **Closer to the Kubernetes API**

+ **Manifest hard to refactor, change (kubectl patch)**

BANZAICLOUD

Ksonnet Core concepts



BANZAICLOUD

Demo

BANZAICLOUD