Join us for KubeCon + CloudNativeCon Virtual

Event dates: August 17-20, 2020
Schedule: Now available!
Cost: $75

Register now!

Production-ready Services with Kubernetes and Serverless

With Jay Smith and Mike Metral

Jason (Jay) Smith

@thejaysmith

Mike Metral

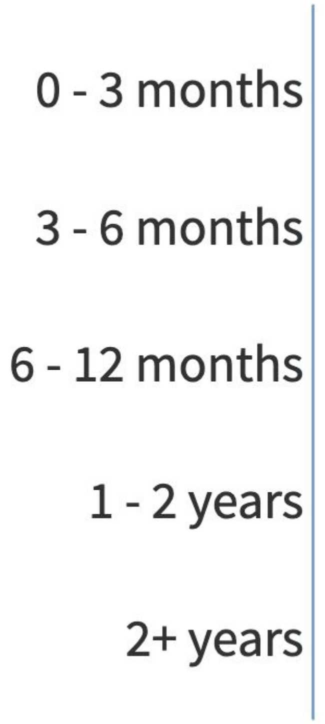
@mikemetral

Outline

- Pulumi Overview
- Managing Kubernetes with programming languages
- Knative Overview
- Serverless apps

Respond at **PollEv.com/mikemetral075**

How long have you worked with Kubernetes?



Poll Everywhere

Respond at **PollEv.com/mikemetral075**

How long have you worked with some form of Serverless?

Poll Everywhere

Modern Infrastructure as Code

Cloud Native infrastructure using your favorite languages

- ☁️ Any AWS, Azure, or GCP service
- 🏔️ Share best practices using package managers
- 📝 Preview changes before they happen
- 📚 Full audit of who changed what and when
- 🔐 Easy secrets management
- 🔴 Test your infrastructure
- 👥 Open source SDK, SaaS available for teams

```go
package main

import (
  "github.com/pulumi/pulumi-gcp/compute"
  "github.com/pulumi/pulumi/sdk/go/pulumi"
)

func main() {
  pulumi.Run(func(ctx *pulumi.Context) {
    // Create a network and firewall rules.
    firewall := compute.NewFirewall(
      ctx, "firewall", &compute.FirewallArgs{
        Allows: []int{22, 80},
      },
    )

    // Create a web server and export its IP.
    inst := compute.NewInstance(
      ctx, "instance", &compute.InstanceArgs{
        MachineType: "f1-micro",
        BootDisk: "debian-9-stretch-v20181210",
      },
    )
    ctx.Export("instanceIP", inst.Nics[0].NatIp)
  })
}
```

Create a GCP Firewall and VM using Go

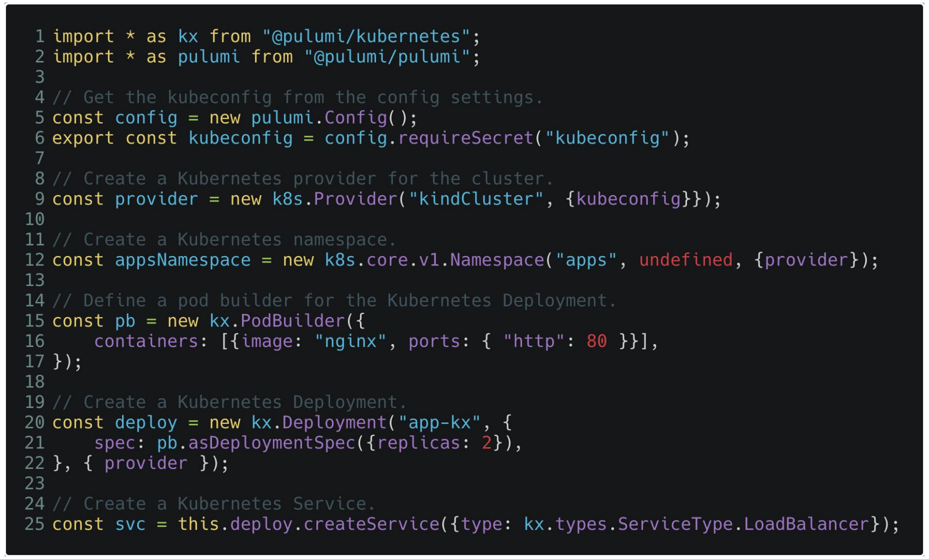
Any Cloud

Real Languages

Kubernetes Superpowers

No YAML, JSON, or DSLs
Use your favorite languages

- ⚙️ Declarative infrastructure as code
- 🏃‍♀️ More productivity, less copy and paste
- 👀 Preview changes before they happen
- ⏳ Rich deployment status updates
- ☸ Deploy Helm charts
- 💉 Inject sidecars for Envoy, Istio, others
- 🚃 Built-in continuous delivery integrations



```typescript
import * as kx from "@pulumi/kubernetes";
import * as pulumi from "@pulumi/pulumi";

// Get the kubeconfig from the config settings.
const config = new pulumi.Config();
export const kubeconfig = config.requireSecret("kubeconfig");

// Create a Kubernetes provider for the cluster.
const provider = new k8s.Provider("kindCluster", {kubeconfig});

// Create a Kubernetes namespace.
const appsNamespace = new k8s.core.v1.Namespace("apps", undefined, {provider});

// Define a pod builder for the Kubernetes Deployment.
const pb = new kx.PodBuilder({
    containers: [{image: "nginx", ports: { "http": 80 }}],
});

// Create a Kubernetes Deployment.
const deploy = new kx.Deployment("app-kx", {
    spec: pb.asDeploymentSpec({replicas: 2}),
}, { provider });

// Create a Kubernetes Service.
const svc = this.deploy.createService({type: kx.types.ServiceType.LoadBalancer});
```

Create a Kubernetes Deployment and Service using TypeScript

Get started today:
https://pulumi.com

pulumi
@PulumiCorp

A Quick Demo

1. **Deploy a GKE cluster**
2. **Deploy an app**
3. **Deploy a Helm chart**

Deploy Code From/To Anywhere

SOURCE CODE

- GitHub
- ATLASSIAN
- GitLab
- Visual Studio Team Services

LANGUAGES

Infrastructure as Code

- JS
- TS
- python™
- GO
- .NET Core

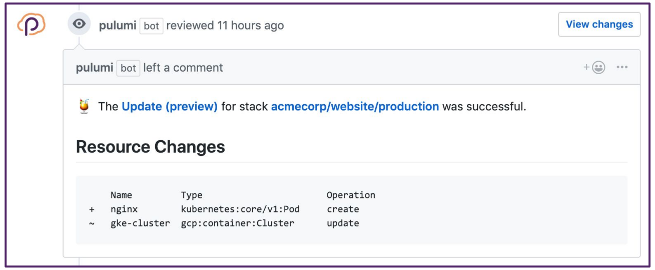
Applications
Any Language

TOOLS

- npm
- Spinnaker
- Jenkins
- circleci
- Travis CI
- codefresh
- GitHub

ENVIRONMENTS

- kubernetes
- aws
- Google Cloud Platform
- Azure
- vmware vSphere
- openstack.
- DigitalOcean


pulumi bot reviewed 11 hours ago

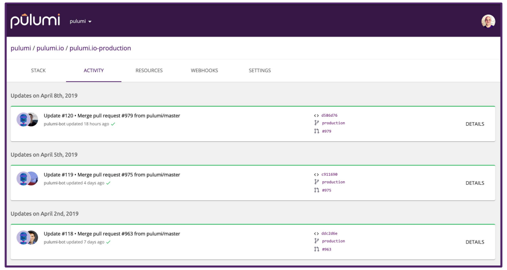
View changes

pulumi bot left a comment

The Update (preview) for stack acmecorp/website/production was successful.

Resource Changes

```
    Name         Type                    Operation
+   nginx        kubernetes:core/v1:Pod  create
~   gke-cluster  gcp:container:Cluster   update
```


pulumi / pulumi.io / pulumi.io-production

Updates on April 8th, 2019

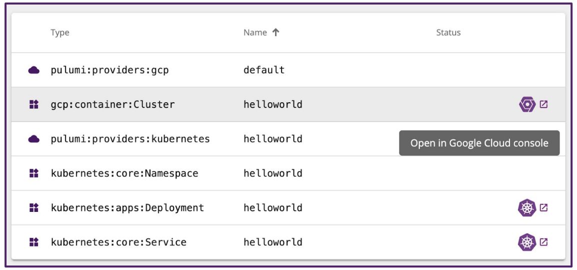
Update #120 • Merge pull request #979 from pulumi/master
pulumi-bot updated 18 hours ago ✓
d586d76 · production · #979 — DETAILS

Updates on April 5th, 2019

Update #119 • Merge pull request #975 from pulumi/master
pulumi-bot updated 4 days ago ✓
c911690 · production · #975 — DETAILS

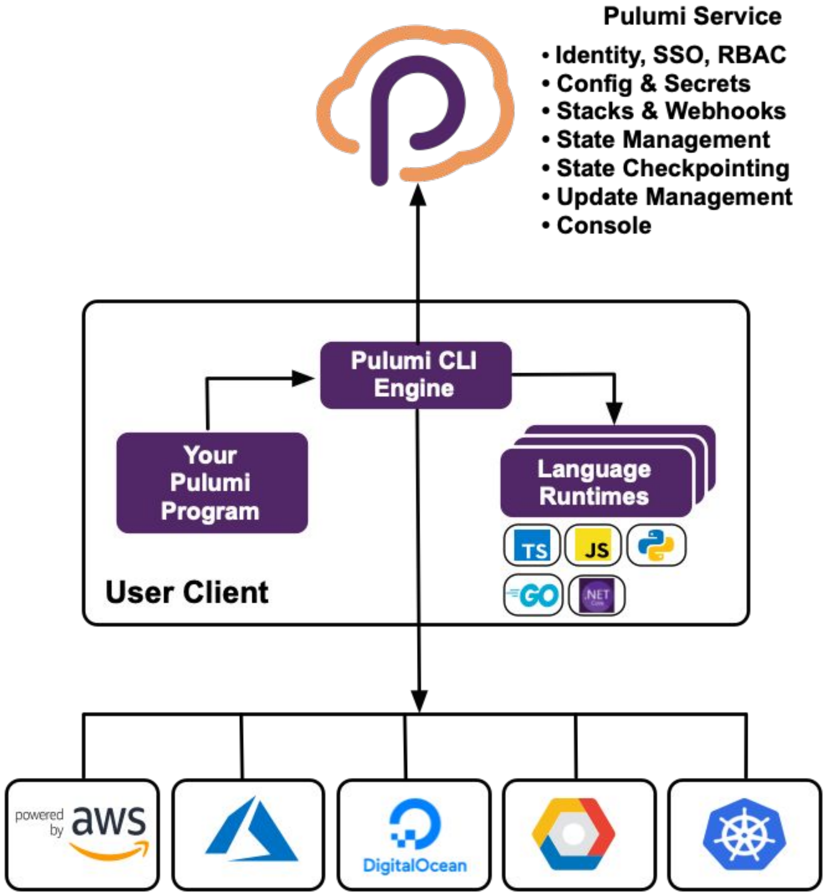
Updates on April 2nd, 2019

Update #118 • Merge pull request #963 from pulumi/master
pulumi-bot updated 7 days ago ✓
ddc2d6e · production · #963 — DETAILS

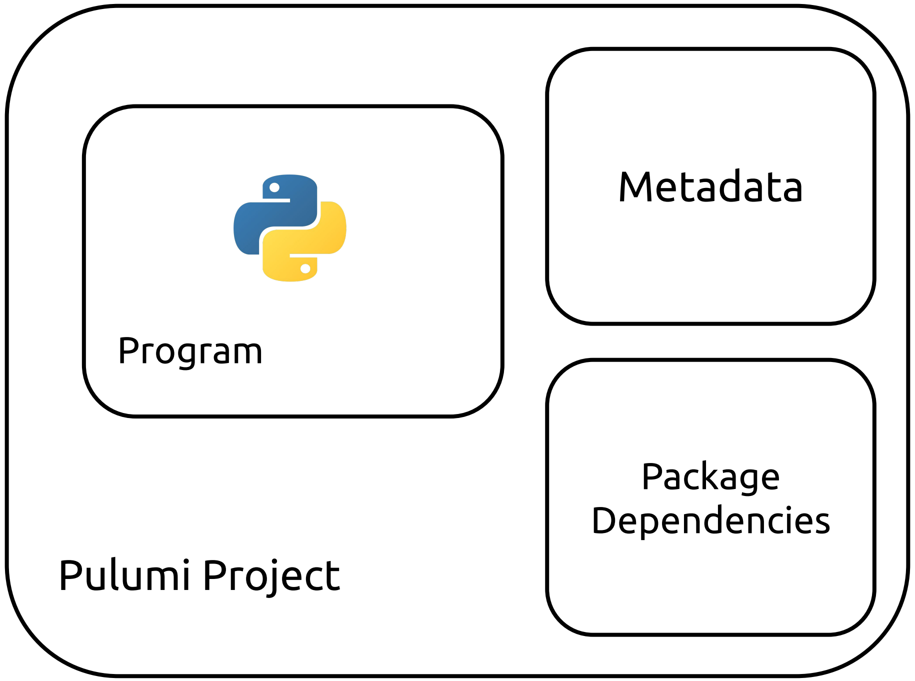


Type	Name ↑	Status
pulumi:providers:gcp	default	
gcp:container:Cluster	helloworld	
pulumi:providers:kubernetes	helloworld	
kubernetes:core:Namespace	helloworld	
kubernetes:apps:Deployment	helloworld	
kubernetes:core:Service	helloworld	

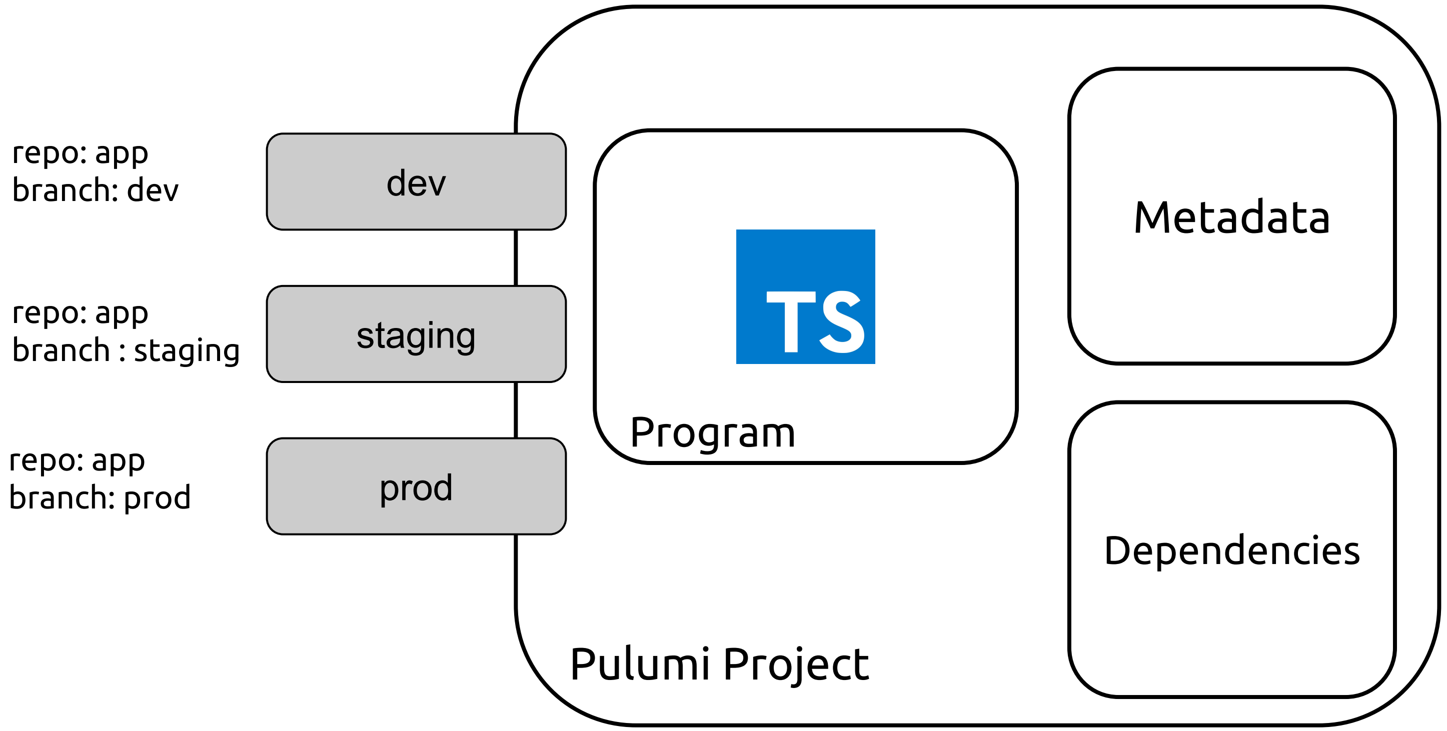
Pulumi CLI and Service



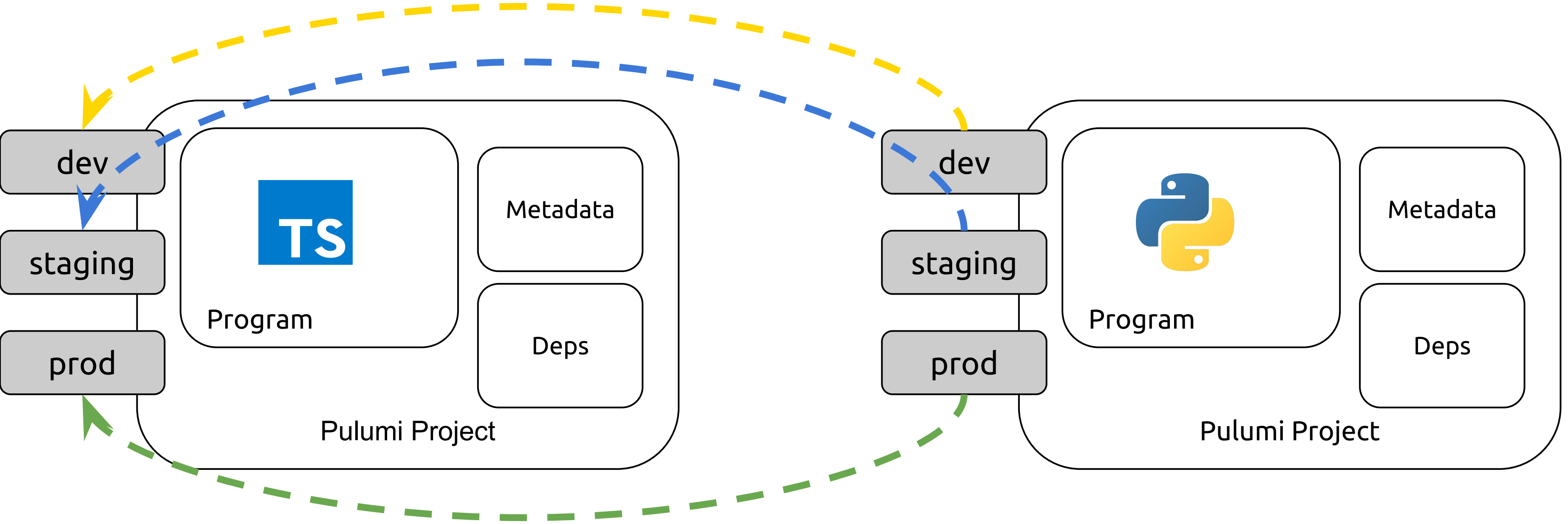
Projects and Programs



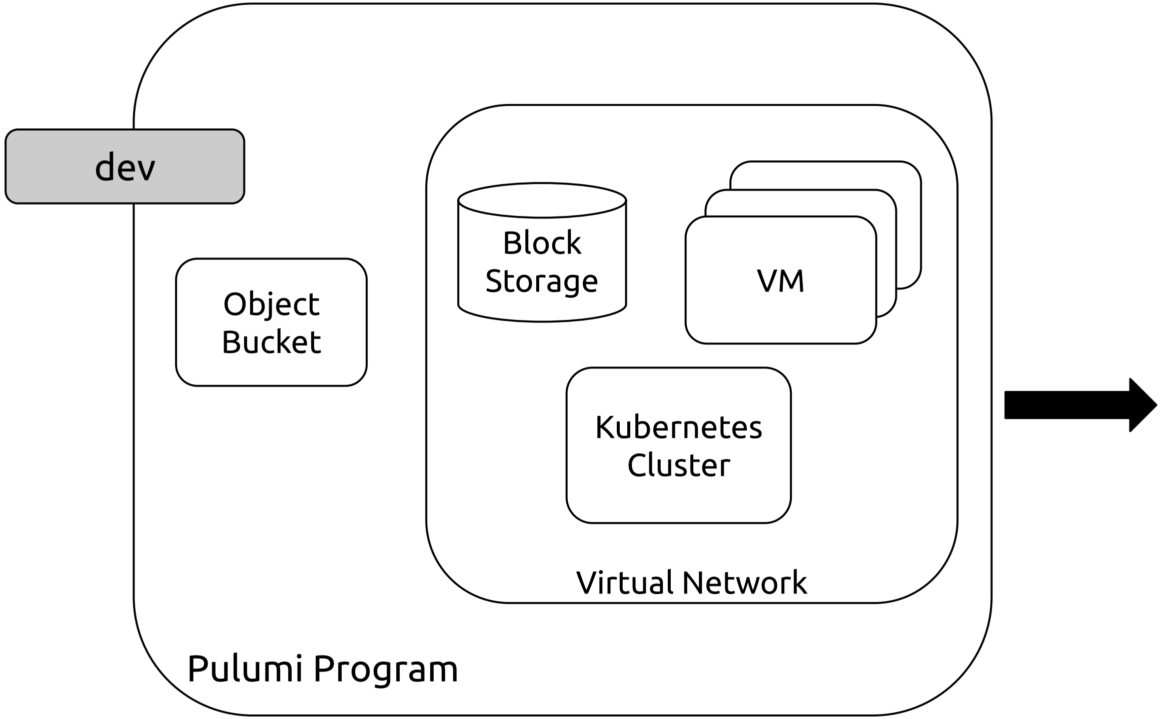
Stacks

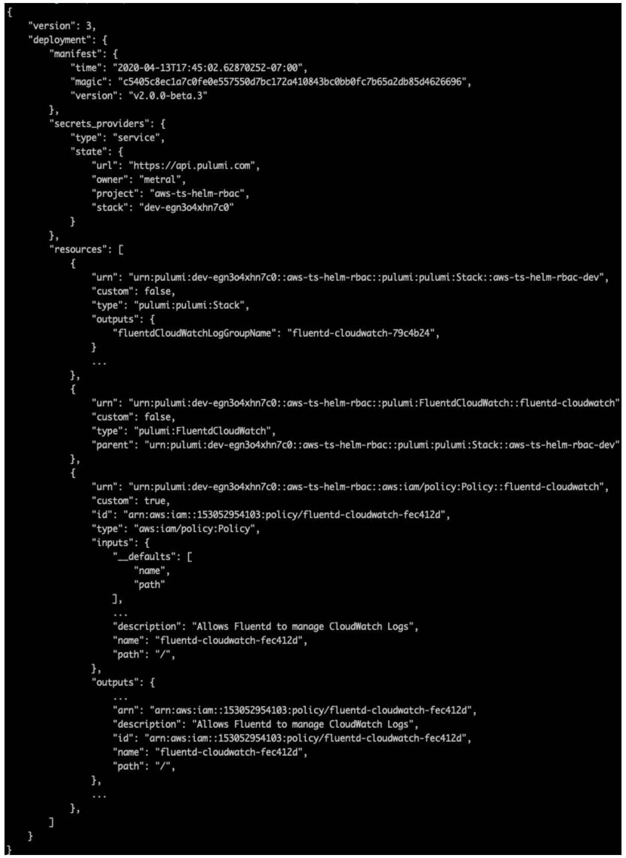

repo: app
branch: dev
dev
repo: app
branch : staging
staging
repo: app
branch: prod
prod
TS
Program
Metadata
Dependencies
Pulumi Project

Stack References

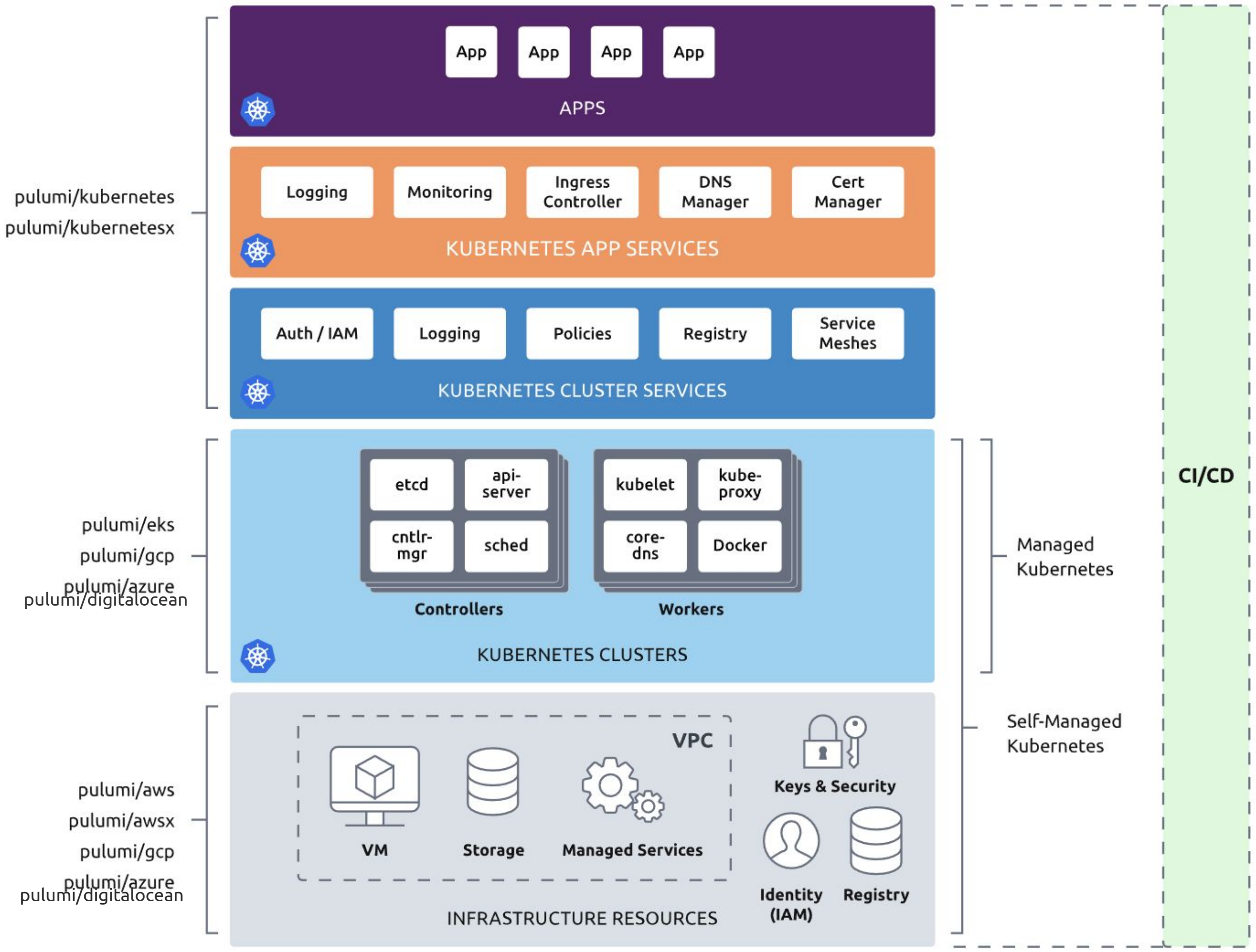


Pulumi Programs for Infrastructure





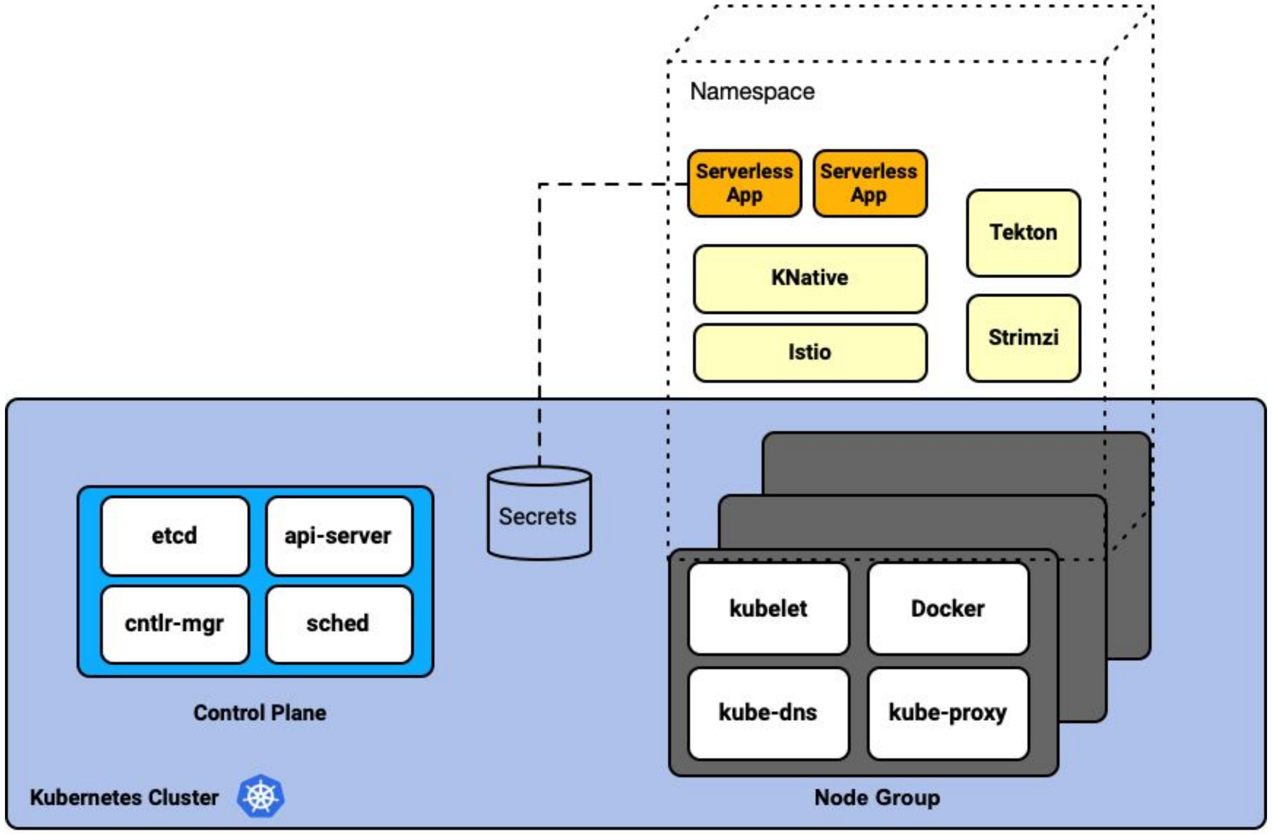
```
{
    "version": 3,
    "deployment": {
        "manifest": {
            "time": "2020-04-13T17:45:02.62870252-07:00",
            "magic": "c5405c8ec1a7c0fe0e557550d7bc172a410843bc0bb0fc7b65a2db85d4626696",
            "version": "v2.0.0-beta.3"
        },
        "secrets_providers": {
            "type": "service",
            "state": {
                "url": "https://api.pulumi.com",
                "owner": "metral",
                "project": "aws-ts-helm-rbac",
                "stack": "dev-egn3o4xhn7c0"
            }
        },
        "resources": [
            {
                "urn": "urn:pulumi:dev-egn3o4xhn7c0::aws-ts-helm-rbac::pulumi:pulumi:Stack::aws-ts-helm-rbac-dev",
                "custom": false,
                "type": "pulumi:pulumi:Stack",
                "outputs": {
                    "fluentdCloudWatchLogGroupName": "fluentd-cloudwatch-79c4b24",
                }
                ...
            },
            {
                "urn": "urn:pulumi:dev-egn3o4xhn7c0::aws-ts-helm-rbac::pulumi:FluentdCloudWatch::fluentd-cloudwatch"
                "custom": false,
                "type": "pulumi:FluentdCloudWatch",
                "parent": "urn:pulumi:dev-egn3o4xhn7c0::aws-ts-helm-rbac::pulumi:pulumi:Stack::aws-ts-helm-rbac-dev"
            },
            {
                "urn": "urn:pulumi:dev-egn3o4xhn7c0::aws-ts-helm-rbac::aws:iam/policy:Policy::fluentd-cloudwatch",
                "custom": true,
                "id": "arn:aws:iam::153052954103:policy/fluentd-cloudwatch-fec412d",
                "type": "aws:iam/policy:Policy",
                "inputs": {
                    "__defaults": [
                        "name",
                        "path"
                    ],
                    ...
                    "description": "Allows Fluentd to manage CloudWatch Logs",
                    "name": "fluentd-cloudwatch-fec412d",
                    "path": "/",
                },
                "outputs": {
                    ...
                    "arn": "arn:aws:iam::153052954103:policy/fluentd-cloudwatch-fec412d",
                    "description": "Allows Fluentd to manage CloudWatch Logs",
                    "id": "arn:aws:iam::153052954103:policy/fluentd-cloudwatch-fec412d",
                    "name": "fluentd-cloudwatch-fec412d",
                    "path": "/",
                },
                ...
            },
        ]
    }
}
```


App
App
App
App
APPS
Logging
Monitoring
Ingress Controller
DNS Manager
Cert Manager
KUBERNETES APP SERVICES
Auth / IAM
Logging
Policies
Registry
Service Meshes
KUBERNETES CLUSTER SERVICES
etcd
api-server
cntlr-mgr
sched
Controllers
kubelet
kube-proxy
core-dns
Docker
Workers
KUBERNETES CLUSTERS
VPC
VM
Storage
Managed Services
Keys & Security
Identity (IAM)
Registry
INFRASTRUCTURE RESOURCES
pulumi/kubernetes
pulumi/kubernetesx
pulumi/eks
pulumi/gcp
pulumi/azure
pulumi/digitalocean
pulumi/aws
pulumi/awsx
pulumi/gcp
pulumi/azure
pulumi/digitalocean
Managed Kubernetes
Self-Managed Kubernetes
CI/CD

Cluster Setup for Serverless Apps

1. **GKE** Kubernetes Cluster
2. **Istio** (required by Knative) - Service Mesh
3. **Knative Serving** - Run Serverless apps
4. **Knative Eventing** - Run Serverless eventing
5. **Berglas** - Manage GCP SecretManagers secrets
6. **Strimzi** - Kafka Operator
7. **Tekton** - CI/CD

Deploying the Infrastructure



Demo: Setting up the Cluster for Serverless Apps

Do developers want to use Kubernetes directly?

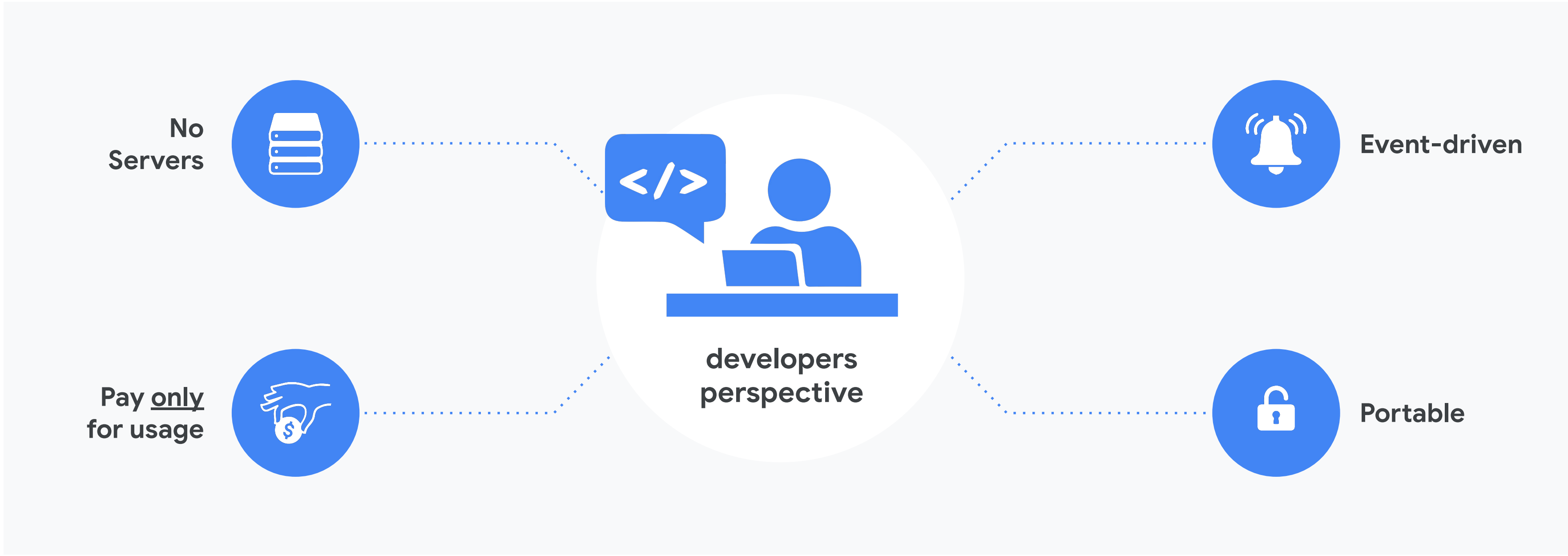
Have to do	Want to do
Write code	Write code
Build docker image	
Upload image to registry	
Deploy service	
Expose to the internet	
Set up monitoring	
Set up autoscaling	

Google Cloud

Developers

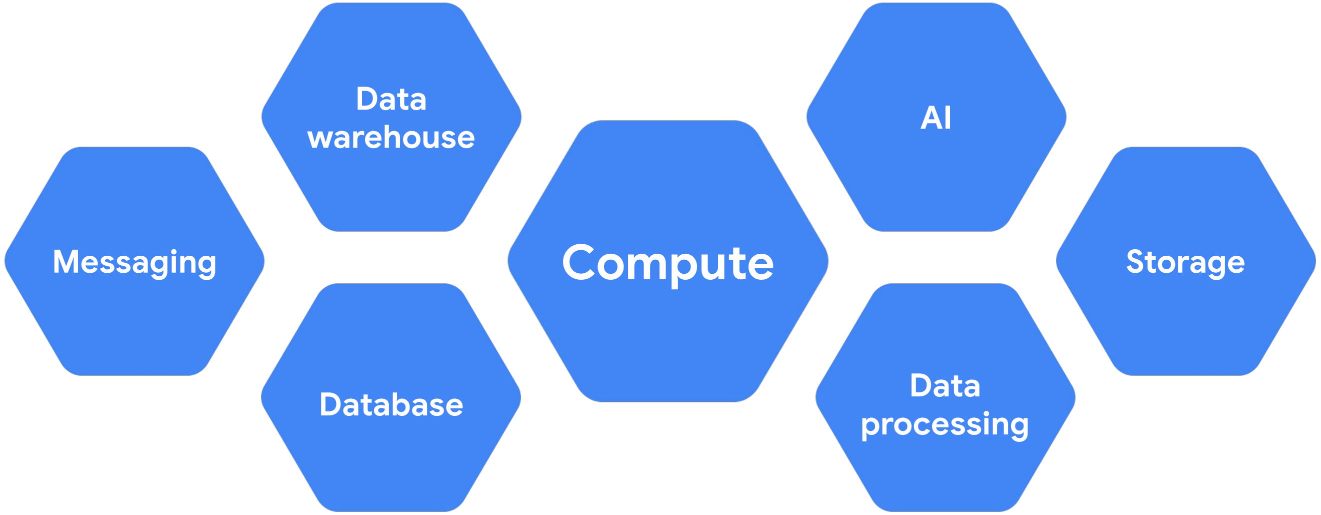
... just want to run their code.
... want to use their **favorite languages** and dependencies.
... don't want to manage the infrastructure.

Serverless usage models



Google Cloud

Serverless is more than snippets of code



Build

Deploy

Consume

Google Cloud

TEKTON

Tekton aims to improve the security, velocity and reliability of software delivery for everyone by creating a set of standard CI/CD components based on cloud native technologies.

- **Open Source and governed by the new Continuous Delivery Foundation (cd.foundation)**
- Kubernetes-native components that are declarative, reproducible and composable
- Uses Pipelines to declare sets of tasks
- Catalog of reusable Tasks and Pipelines
- Integrated with other projects such as Jenkins X, Knative and more!

https://cloud.google.com/tekton/
https://github.com/tektoncd

Google Cloud

Knative

Building blocks for serverless workloads on Kubernetes

What Knative is

- An open source project
- Set of building blocks to construct your own FaaS/PaaS
    - abstracts common tasks through custom Kubernetes API objects
- An abstraction on top of Kubernetes
    - **It's still Kubernetes:** Runs containers at the end of the day.

What Knative is not

- It's not a Google product.
- It's not FaaS.

What can you do with Knative?

- [Developers] Use it directly to deploy stuff (not easy, but works fine)
- [Operators] Put a level of abstraction between your devs and Kubernetes.
- [Platform Architects] Use it to build your own serverless platform.
    - e.g. DIY Heroku or GCF/Lambda.

Knative Serving

Benefits

- Seamlessly scale up and down
- Built-in traffic splitting between revisions
- Integrates networking and service mesh automatically
- Easy to reason about object model

Pluggable

- Connect to your own logging and monitoring platform, or use the built-in system
- Auto-scaler can be tuned or swapped out for custom code

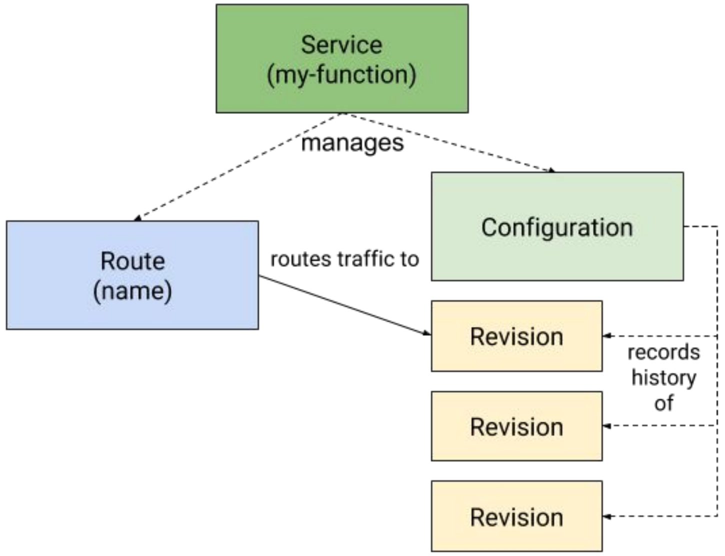
Knative Serving

Primitives with clear separation of concerns:

Configuration
Current/desired state of an application
Code & configuration separated (a la 12-factor)

Revision
Point in time snapshots for your code and configuration

Route
Maps traffic to a revisions
Supports fractional, named routing



Knative enhances Kubernetes

Autoscaling

Kubernetes	Memory/CPU based autoscaling (slow)
Knative	Rapid, request-oriented autoscaling Handles traffic spikes

More: https://ahmet.im/blog/knative-better-kubernetes-networking/

Knative enhances Kubernetes

Scale to zero

Kubernetes	N/A
Knative	Scale application to 0, if no requests coming Activate (0⟶1) on the next request

Google Cloud

More: https://ahmet.im/blog/knative-better-kubernetes-networking/

Knative enhances Kubernetes

Load Balancing

Kubernetes	Connection-based load balancing
Knative	**Per-request** load balancing **Traffic splitting** (blue/green deployments)

Google Cloud

More: https://ahmet.im/blog/knative-better-kubernetes-networking/

Kubernetes Deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello
      tier: web
  template:
    metadata:
      labels:
        app: hello
        tier: web
    spec:
      containers:
      - name: main
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  labels:
    app: hello
    tier: web
spec:
  type: ClusterIP
  selector:
    app: hello
    tier: web
  ports:
  - port: 80
    targetPort: 8080
```

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
~~  replicas: 1~~
  selector:
    matchLabels:
      app: hello
      tier: web
  template:
    metadata:
      labels:
        app: hello
        tier: web
    spec:
      containers:
      - name: main
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  labels:
    app: hello
    tier: web
spec:
  type: ClusterIP
  selector:
    app: hello
    tier: web
  ports:
  - port: 80
    targetPort: 8080
```

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  ~~replicas: 1~~
  selector:
    ~~matchLabels:~~
      ~~app: hello~~
      ~~tier: web~~
  template:
    ~~metadata:~~
      ~~labels:~~
        ~~app: hello~~
        ~~tier: web~~
    spec:
      containers:
      - ~~name: main~~
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  labels:
    app: hello
    tier: web
spec:
  type: ClusterIP
  selector:
    app: hello
    tier: web
  ports:
  - port: 80
    targetPort: 8080
```

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  ~~replicas: 1~~
  selector:
    ~~matchLabels:~~
      ~~app: hello~~
      ~~tier: web~~
  template:
    ~~metadata:~~
      ~~labels:~~
        ~~app: hello~~
        ~~tier: web~~
    spec:
      containers:
      - ~~name: main~~
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  ~~labels:~~
    ~~app: hello~~
    ~~tier: web~~
spec:
  type: ClusterIP
  selector:
    app: hello
    tier: web
  ports:
  - port: 80
    targetPort: 8080
```

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  ~~replicas: 1~~
  selector:
    ~~matchLabels:~~
      ~~app: hello~~
      ~~tier: web~~
  template:
    ~~metadata:~~
      ~~labels:~~
        ~~app: hello~~
        ~~tier: web~~
    spec:
      containers:
      - ~~name: main~~
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  ~~labels:~~
    ~~app: hello~~
    ~~tier: web~~
spec:
  ~~type: ClusterIP~~
  ~~selector:~~
    ~~app: hello~~
    ~~tier: web~~
  ports:
  - port: 80
    targetPort: 8080
```

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  ~~replicas: 1~~
  selector:
    ~~matchLabels:~~
      ~~app: hello~~
      ~~tier: web~~
  template:
    ~~metadata:~~
      ~~labels:~~
        ~~app: hello~~
        ~~tier: web~~
    spec:
      containers:
      - ~~name: main~~
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  ~~labels:~~
    ~~app: hello~~
    ~~tier: web~~
spec:
  ~~type: ClusterIP~~
  ~~selector:~~
    ~~app: hello~~
    ~~tier: web~~
  ~~ports:~~
    ~~port: 80~~
    ~~targetPort: 8080~~
```

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  ~~replicas: 1~~
  selector:
    ~~matchLabels:~~
      ~~app: hello~~
      ~~tier: web~~
  template:
    ~~metadata:~~
      ~~labels:~~
        ~~app: hello~~
        ~~tier: web~~
    spec:
      containers:
      - ~~name: main~~
        image: gcr.io/google-samples/hello-app:1.0
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  name: hello-web
  ~~labels:~~
    ~~app: hello~~
    ~~tier: web~~
spec:
  ~~type: ClusterIP~~
  ~~selector:~~
    ~~app: hello~~
    ~~tier: web~~
  ~~ports:~~
    ~~port: 80~~
    ~~targetPort: 8080~~
```

Knative Service = Kubernetes Deployment + Kubernetes Service

```yaml
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: hello-web
spec:
  template:
    spec:
      containers:
      - image: gcr.io/[...]
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello
      tier: web
  template:
    metadata:
      labels:
        app: hello
        tier: web
    spec:
      containers:
      - name: main
        image: gcr.io/[...]
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

Knative Service = Kubernetes Deployment + Kubernetes Service

```yaml
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: hello-web
spec:
  template:
    spec:
      containers:
      - image: gcr.io/[...]
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

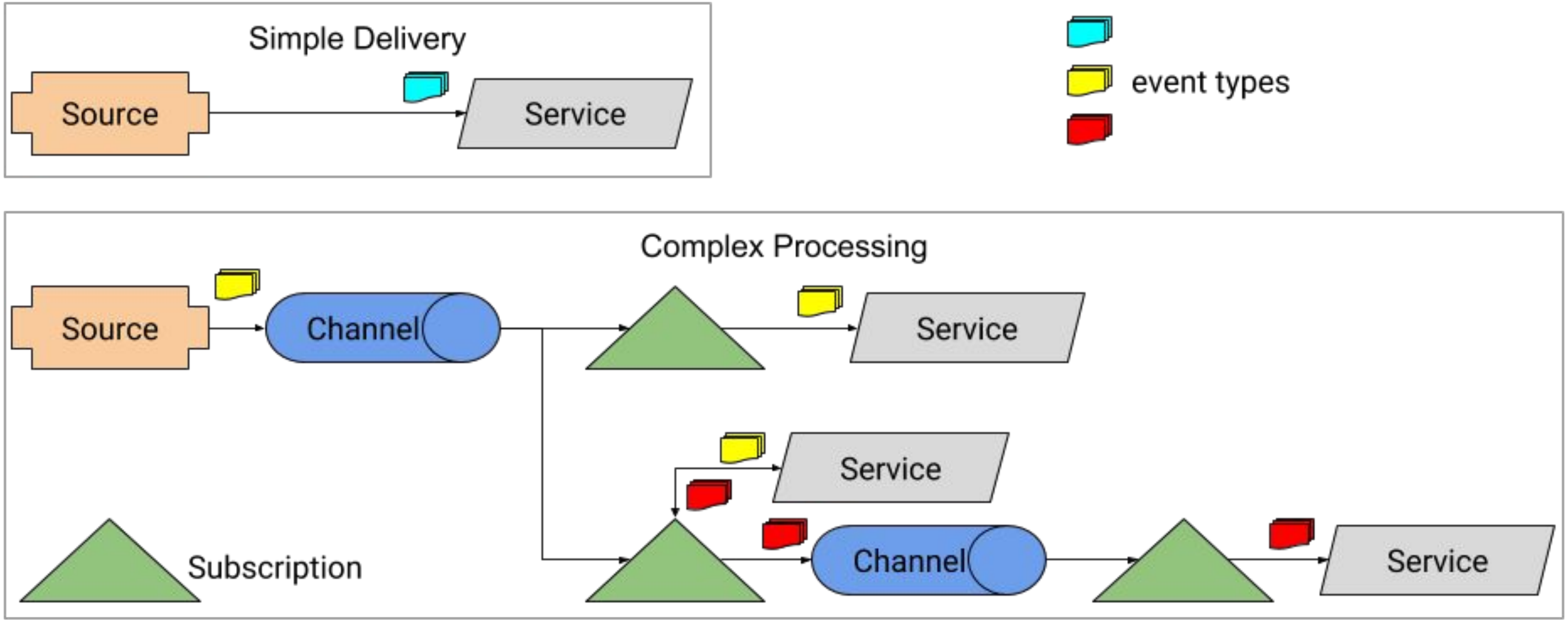
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-web
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello
      tier: web
  template:
    metadata:
      labels:
        app: hello
        tier: web
    spec:
      containers:
      - name: main
        image: gcr.io/[...]
        resources:
          limits:
            cpu: 100m
            memory: 256Mi
```

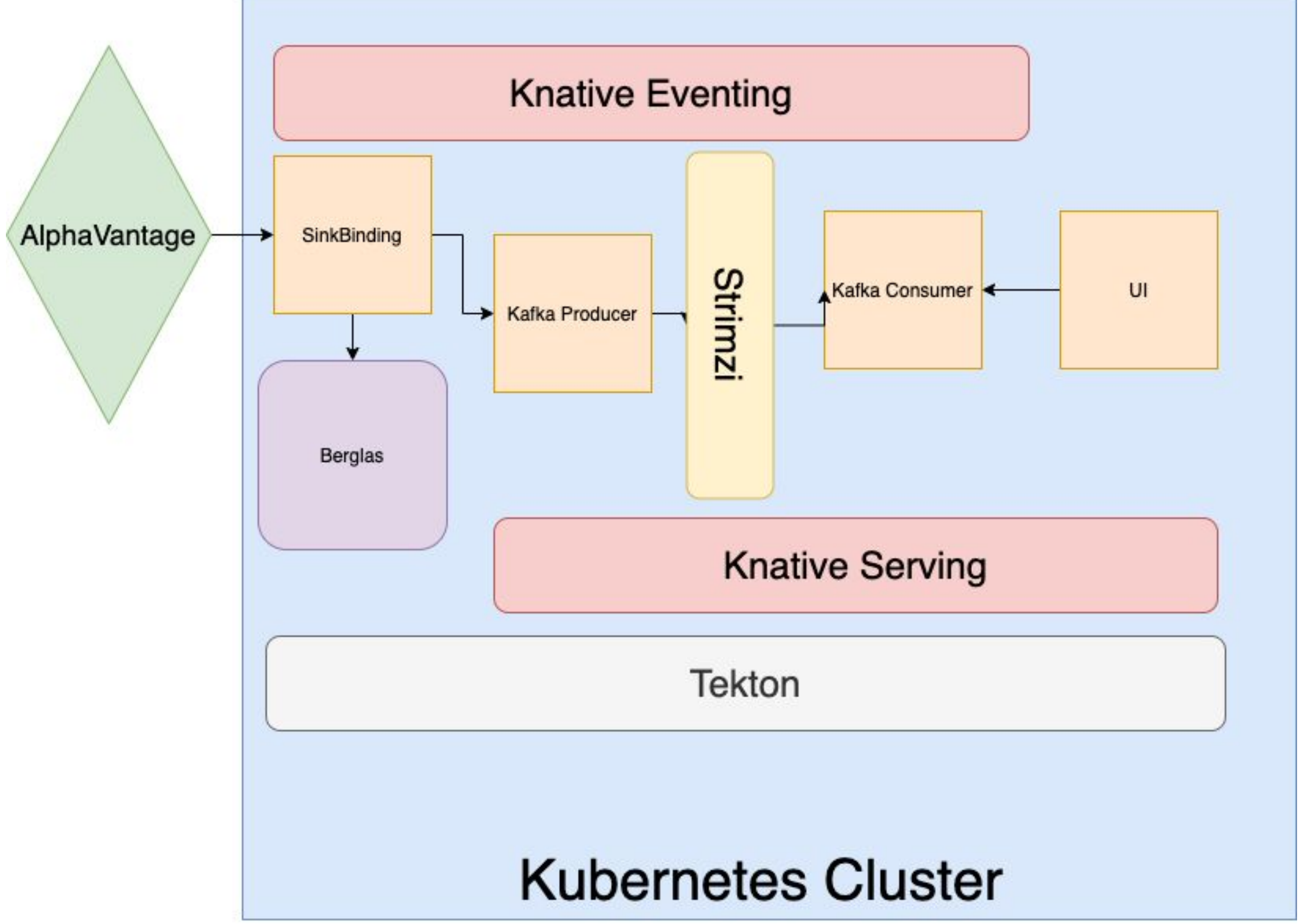
Knative eventing

Benefits

- Declaratively bind between event producers and deployed services
- Scales from just few events to live streams
- Custom event pipelines to connect with your own existing systems

Knative Eventing




Knative Eventing
AlphaVantage
SinkBinding
Kafka Producer
Strimzi
Kafka Consumer
UI
Berglas
Knative Serving
Tekton
Kubernetes Cluster

DEMO

Google Cloud

Q&A

Get started using Kubernetes with Infrastructure as Code:
pulumi.com/kubernetes

Start using KNative with hands-on labs:
forms.gle/APZehmc4WECqid2J6

More questions after the session? Join #kubernetes on
slack.pulumi.com

Infrastructure repo: https://github.com/metral/cncf-gke-pulumi

Serverless repo: https://github.com/TheJaySmith/cncf-streaming-app

@thejaysmith @mikemetral