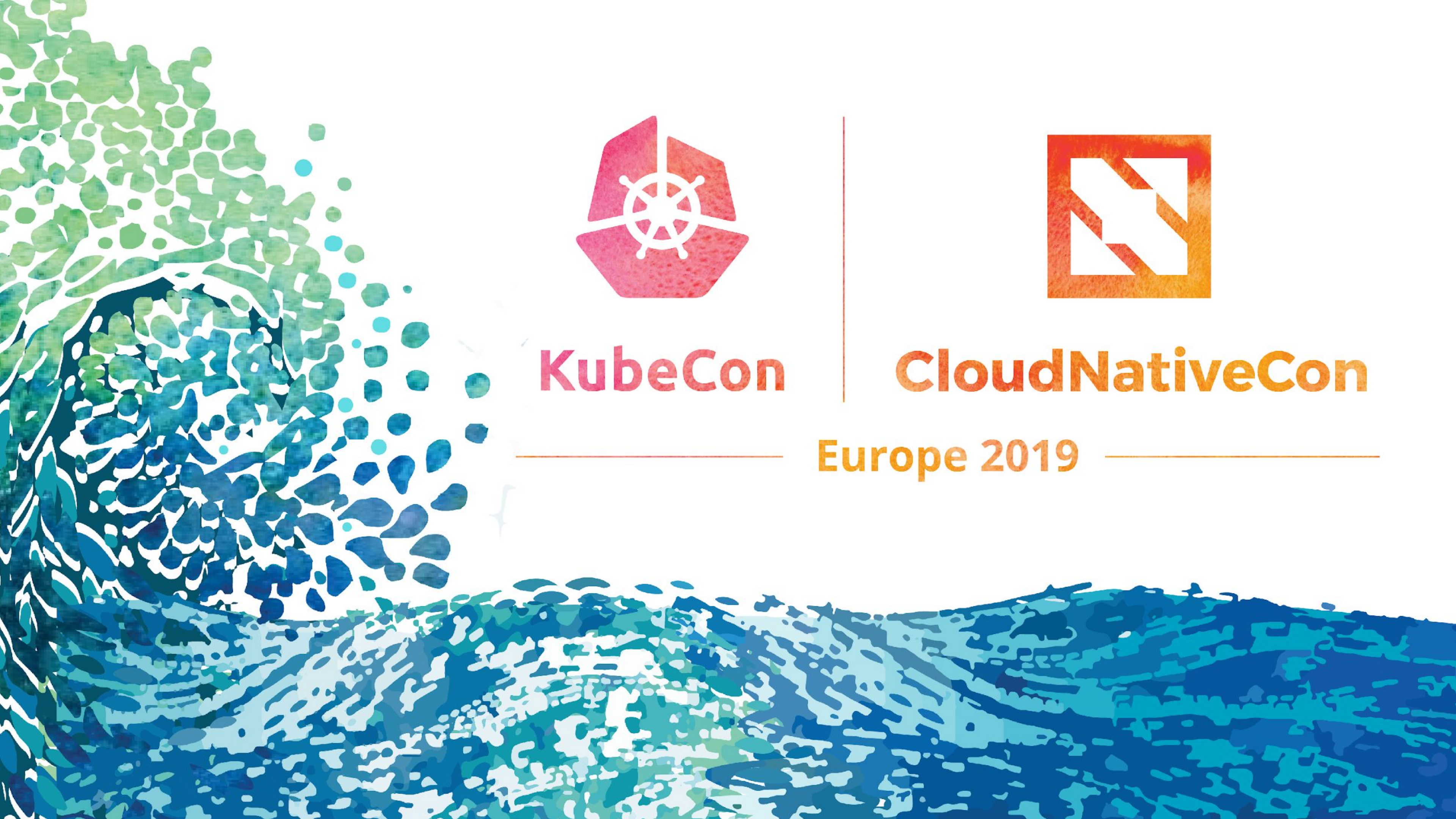


KubeCon | CloudNativeCon

Europe 2019

KubeCon | CloudNativeCon

Europe 2019

Infrastructure as Software

@paulwilljones

What This Talk Is About

Evolving the paradigm for infrastructure provisioning

Agenda

Declarative vs Imperative Infrastructure Composition

Infrastructure as Software - Programmatically creating cloud resources

Pulumi Deep Dive

AWS CDK Deep Dive

Infrastructure Provisioning

Idempotency

Source of truth

Desired vs observed state

Reconciler pattern

Versioned

Auditable

Testable

Infrastructure Provisioning

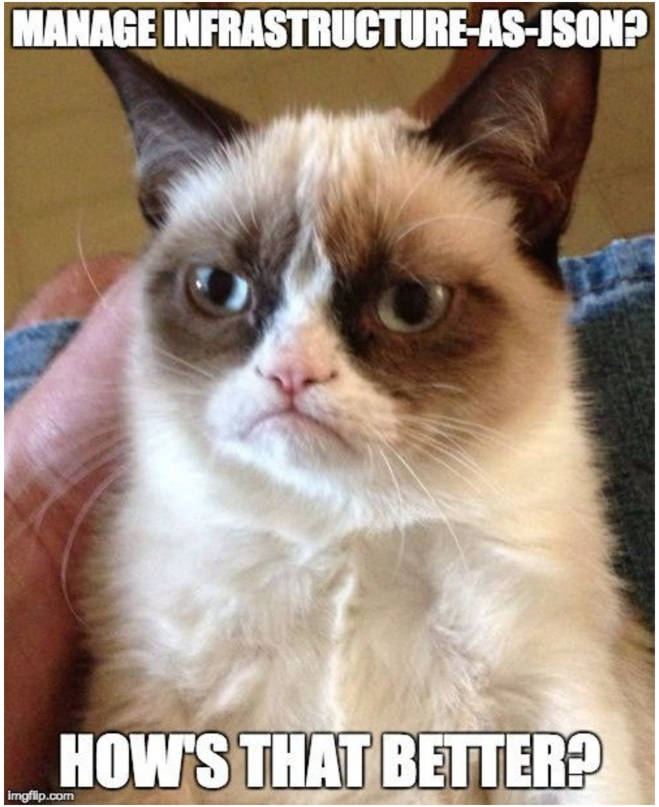
`aws cloudformation`

The first-party tool for Desired State Configuration management from Amazon. CloudFormation templates use YAML to describe all the infrastructure resources of AWS.

`terraform`

An open source tool to define infrastructure in declarative configuration files. It has a pluggable architecture, so the tool supports all major clouds and even hybrid scenarios.

But...


MANAGE INFRASTRUCTURE-AS-JSON?
HOW'S THAT BETTER?
imgflip.com

YAML & DSLs

Verbose

Cognitive overhead

Lack of features

Declarative vs Imperative

Declarative

Saying **what** you want

YAML/JSON/DSLs

Limited on features

Imperative

Saying **how** to do it

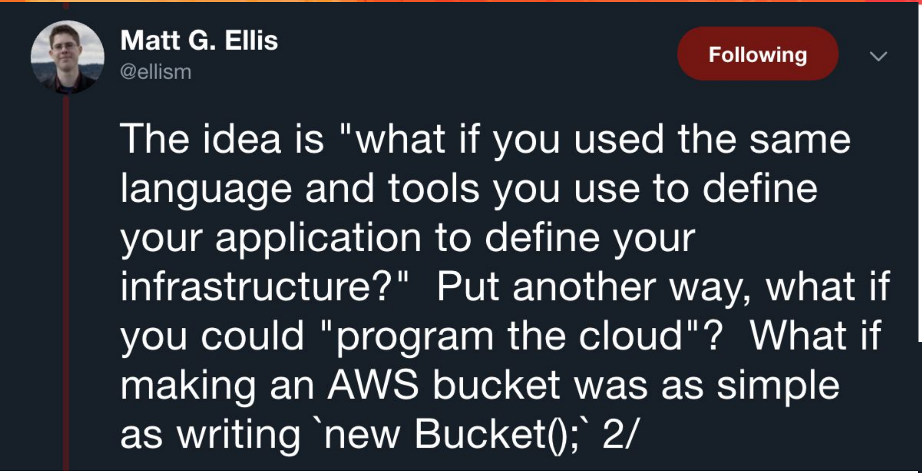
CDK, Troposhere, GoFormation, Pulumi, Cloud SDKs

Added flexibility through feature rich language experiences

Static Analysis

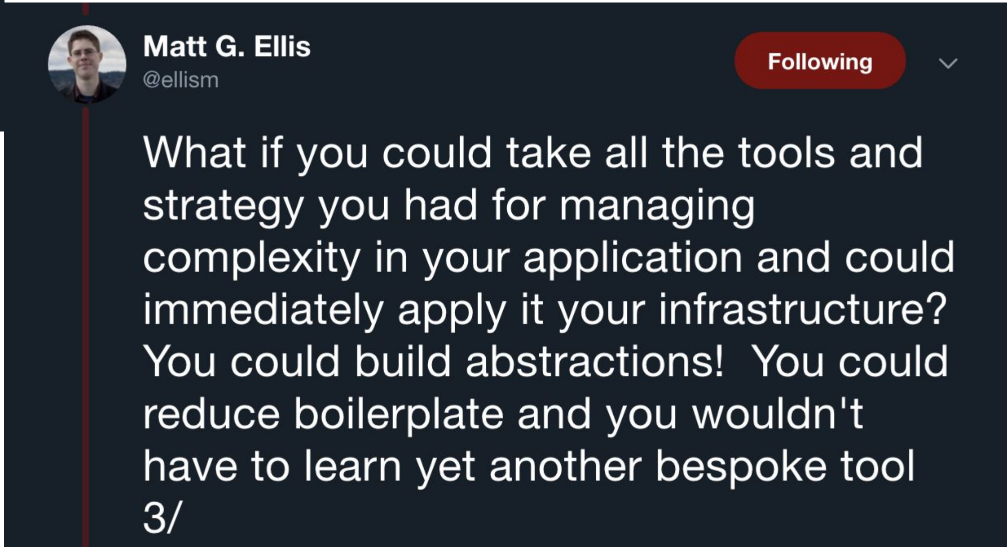
Testability

Evolving the paradigm



Matt G. Ellis
@ellism

The idea is "what if you used the same language and tools you use to define your application to define your infrastructure?" Put another way, what if you could "program the cloud"? What if making an AWS bucket was as simple as writing `new Bucket();` 2/



Matt G. Ellis
@ellism

What if you could take all the tools and strategy you had for managing complexity in your application and could immediately apply it your infrastructure? You could build abstractions! You could reduce boilerplate and you wouldn't have to learn yet another bespoke tool 3/

Infrastructure as Software

Programmatically defining infrastructure using modern programming languages

Infrastructure as Software

Leverages software principles in infrastructure composition

Facilitates more robust governance of infrastructure code

Tighter cohesion between infrastructure and application code

Testable infrastructure code

Infrastructure as Software

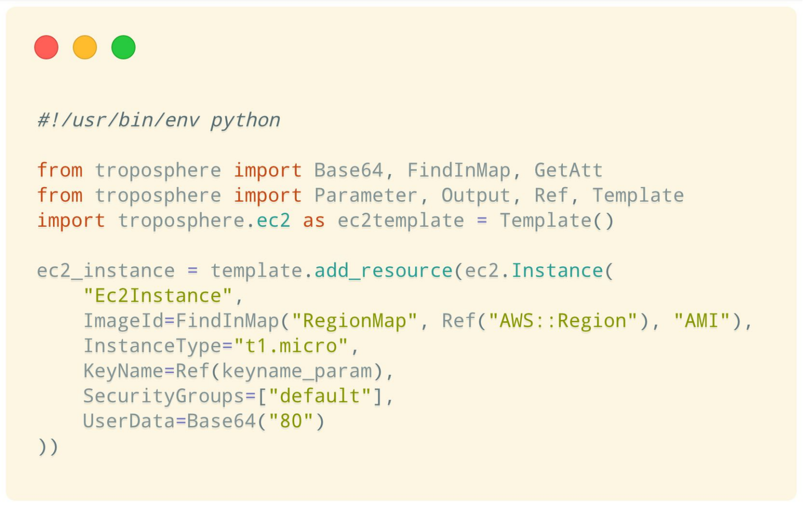
Moving developers down the stack

Moving operations up the stack

Existing solutions

- Troposphere

The Troposphere library allows for easier creation of AWS CloudFormation by writing Python code to describe the AWS resources.



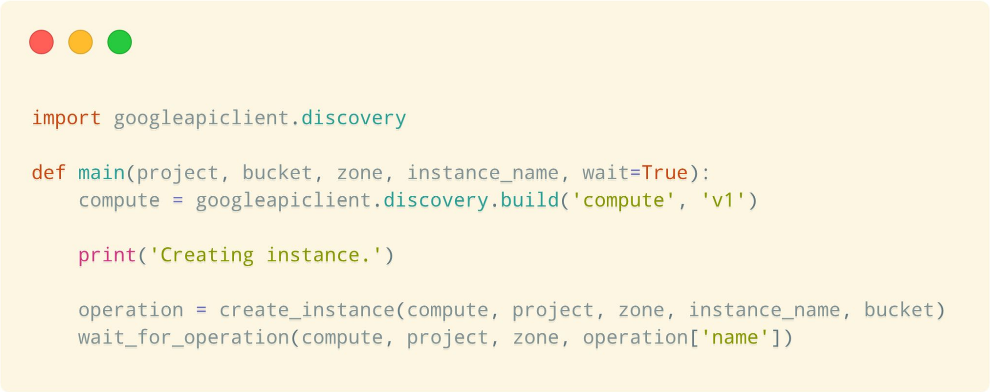
```
#!/usr/bin/env python

from troposphere import Base64, FindInMap, GetAtt
from troposphere import Parameter, Output, Ref, Template
import troposphere.ec2 as ec2template = Template()

ec2_instance = template.add_resource(ec2.Instance(
    "Ec2Instance",
    ImageId=FindInMap("RegionMap", Ref("AWS::Region"), "AMI"),
    InstanceType="t1.micro",
    KeyName=Ref(keyname_param),
    SecurityGroups=["default"],
    UserData=Base64("80")
))
```

- goformation

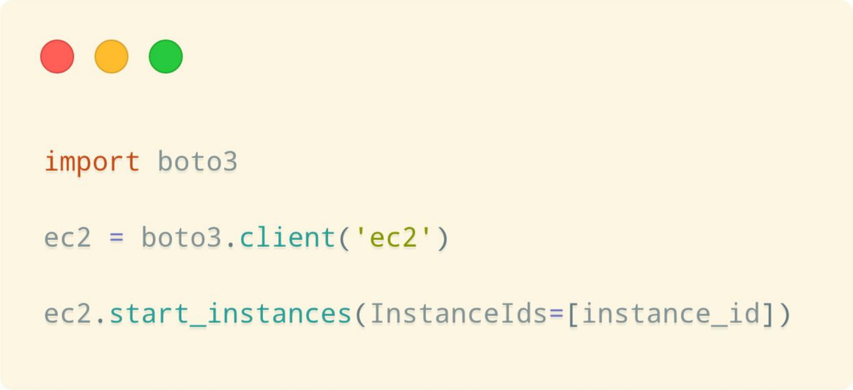
GoFormation is a Go library for working with AWS CloudFormation / AWS Serverless Application Model (SAM) templates.

Existing solutions

Cloud SDKs



```python
import googleapiclient.discovery

def main(project, bucket, zone, instance_name, wait=True):
    compute = googleapiclient.discovery.build('compute', 'v1')

    print('Creating instance.')

    operation = create_instance(compute, project, zone, instance_name, bucket)
    wait_for_operation(compute, project, zone, operation['name'])
```



```python
import boto3

ec2 = boto3.client('ec2')

ec2.start_instances(InstanceIds=[instance_id])
```

pulumi

Pulumi

Delivering Cloud Native Infrastructure as Code

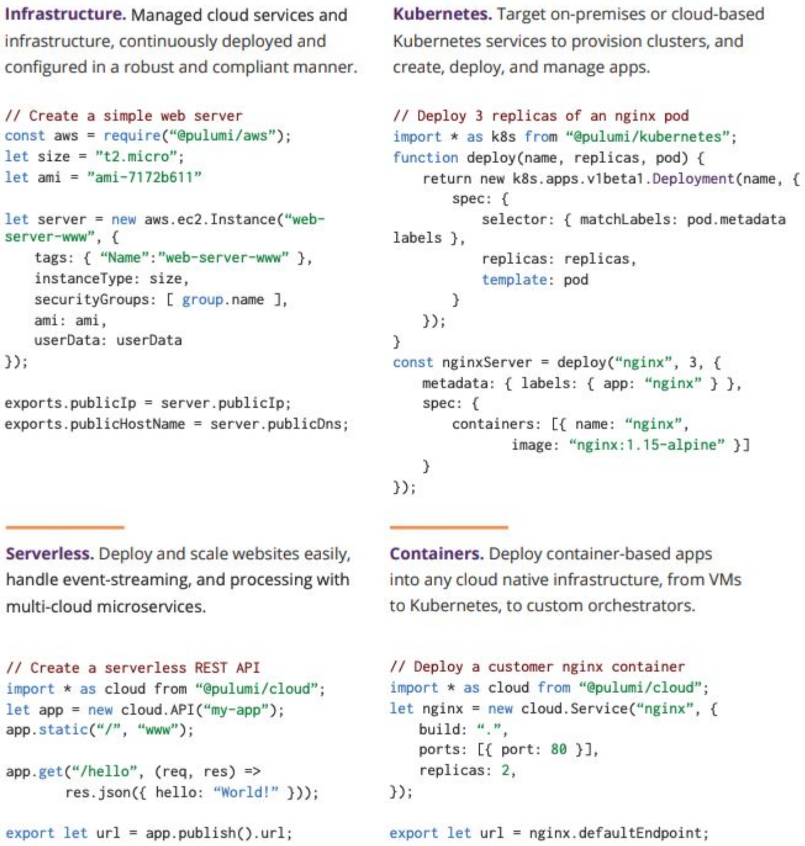
Pulumi is a platform for building and deploying cloud infrastructure and applications in your favourite language on any cloud

Multi-Language Runtime Multi-Cloud Multi-Technology Scope

Abstraction and reuse

TS JS GO

Pulumi



Infrastructure. Managed cloud services and infrastructure, continuously deployed and configured in a robust and compliant manner.

```
// Create a simple web server
const aws = require(“@pulumi/aws”);
let size = ”t2.micro”;
let ami = ”ami-7172b611”

let server = new aws.ec2.Instance(“web-
server-www”, {
    tags: { “Name”:”web-server-www” },
    instanceType: size,
    securityGroups: [ group.name ],
    ami: ami,
    userData: userData
});

exports.publicIp = server.publicIp;
exports.publicHostName = server.publicDns;
```

Kubernetes. Target on-premises or cloud-based Kubernetes services to provision clusters, and create, deploy, and manage apps.

```
// Deploy 3 replicas of an nginx pod
import * as k8s from “@pulumi/kubernetes”;
function deploy(name, replicas, pod) {
    return new k8s.apps.v1beta1.Deployment(name, {
        spec: {
            selector: { matchLabels: pod.metadata
labels },
            replicas: replicas,
            template: pod
        }
    });
}
const nginxServer = deploy(“nginx”, 3, {
    metadata: { labels: { app: “nginx” } },
    spec: {
        containers: [{ name: “nginx”,
                image: “nginx:1.15-alpine” }]
    }
});
```

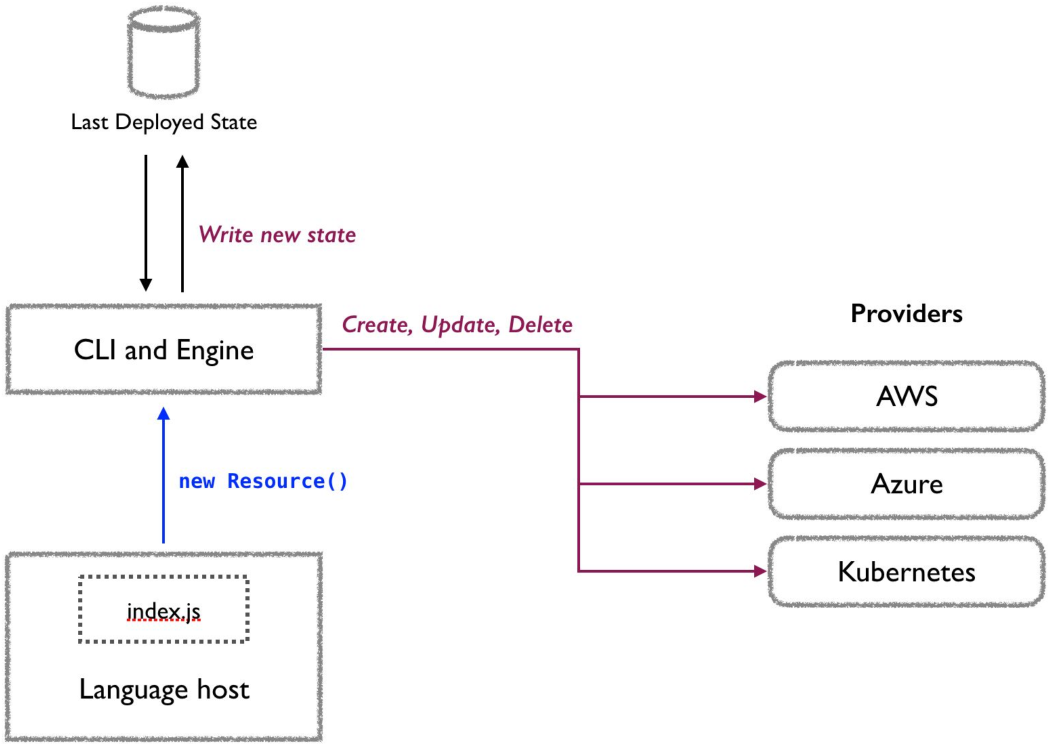
Serverless. Deploy and scale websites easily, handle event-streaming, and processing with multi-cloud microservices.

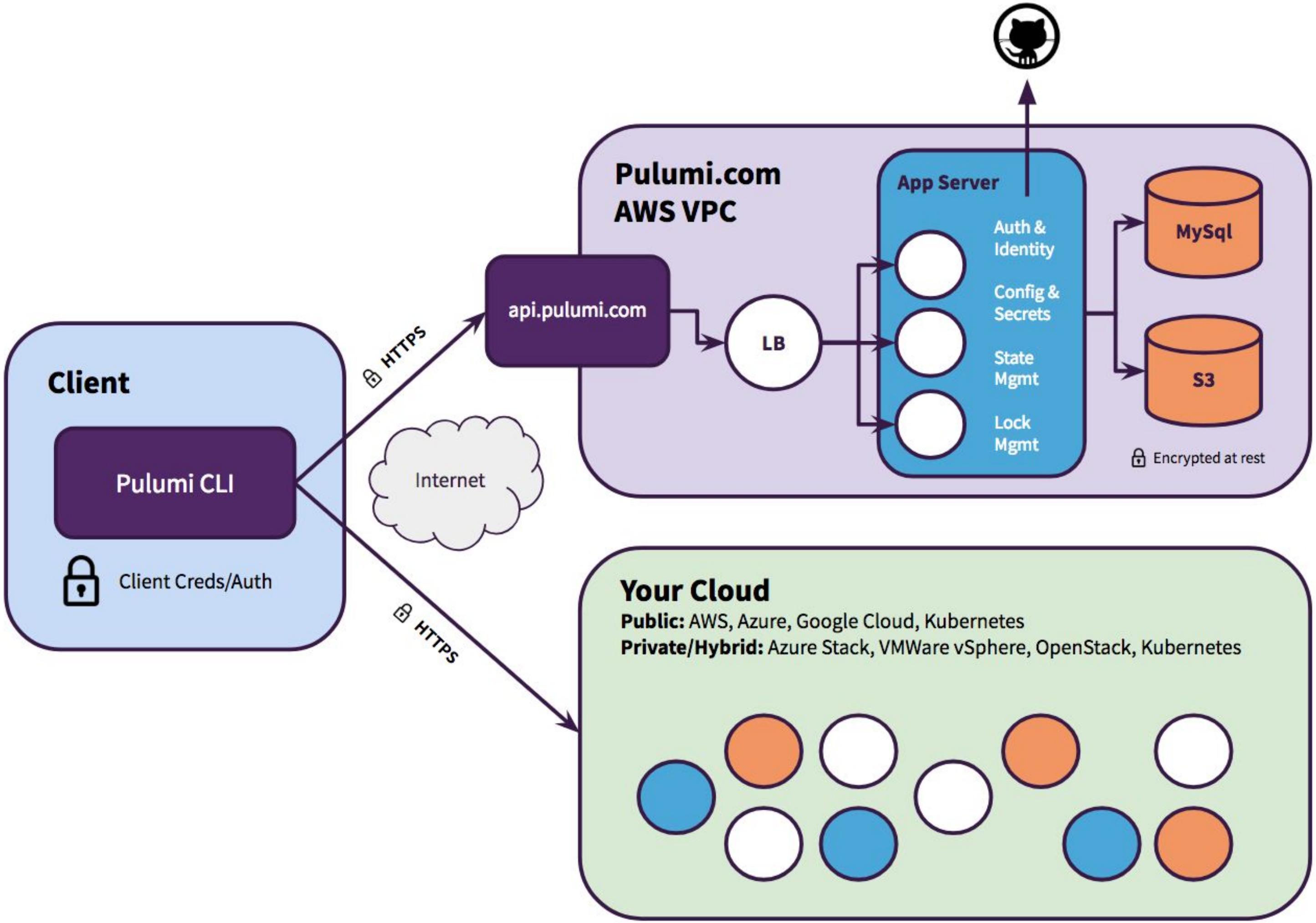
```
// Create a serverless REST API
import * as cloud from “@pulumi/cloud”;
let app = new cloud.API(“my-app”);
app.static(“/”, “www”);

app.get(“/hello”, (req, res) =>
       res.json({ hello: “World!” }));

export let url = app.publish().url;
```

Containers. Deploy container-based apps into any cloud native infrastructure, from VMs to Kubernetes, to custom orchestrators.

```
// Deploy a customer nginx container
import * as cloud from “@pulumi/cloud”;
let nginx = new cloud.Service(“nginx”, {
    build: “.”,
    ports: [{ port: 80 }],
    replicas: 2,
});

export let url = nginx.defaultEndpoint;
```

Pulumi - Architecture




Client
Pulumi CLI
Client Creds/Auth
HTTPS
HTTPS
Internet
api.pulumi.com
Pulumi.com
AWS VPC
LB
App Server
Auth &
Identity
Config &
Secrets
State
Mgmt
Lock
Mgmt
MySql
S3
Encrypted at rest
Your Cloud
Public: AWS, Azure, Google Cloud, Kubernetes
Private/Hybrid: Azure Stack, VMWare vSphere, OpenStack, Kubernetes

Pulumi comparisons

Terraform

- Language features
- Kubernetes native support

CloudFormation

- Multi-cloud
- Language features

Cloud SDKs

- Orchestration of provisioning and state management
- Reconciler pattern
- Concurrency management

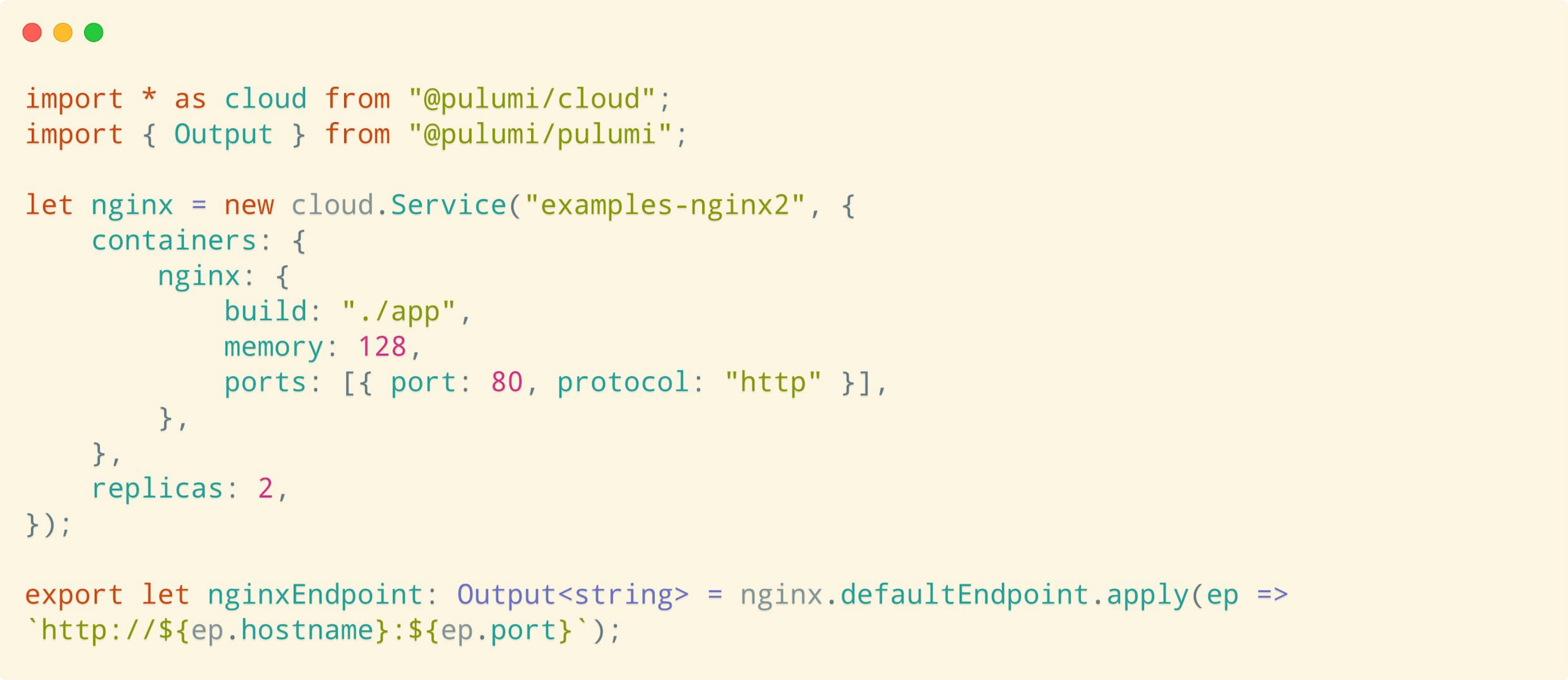
Pulumi Cloud

Pulumi's multi-cloud framework for building modern container and serverless cloud applications

Offers interoperability for cloud resource composition

aws

Pulumi Cloud



```
import * as cloud from "@pulumi/cloud";
import { Output } from "@pulumi/pulumi";

let nginx = new cloud.Service("examples-nginx2", {
    containers: {
        nginx: {
            build: "./app",
            memory: 128,
            ports: [{ port: 80, protocol: "http" }],
        },
    },
    replicas: 2,
});

export let nginxEndpoint: Output<string> = nginx.defaultEndpoint.apply(ep =>
`http://${ep.hostname}:${ep.port}`);
```

Pulumi Kubernetes

Pulumi exposes a Kubernetes SDK to compose K8s deployments in general purpose programming languages

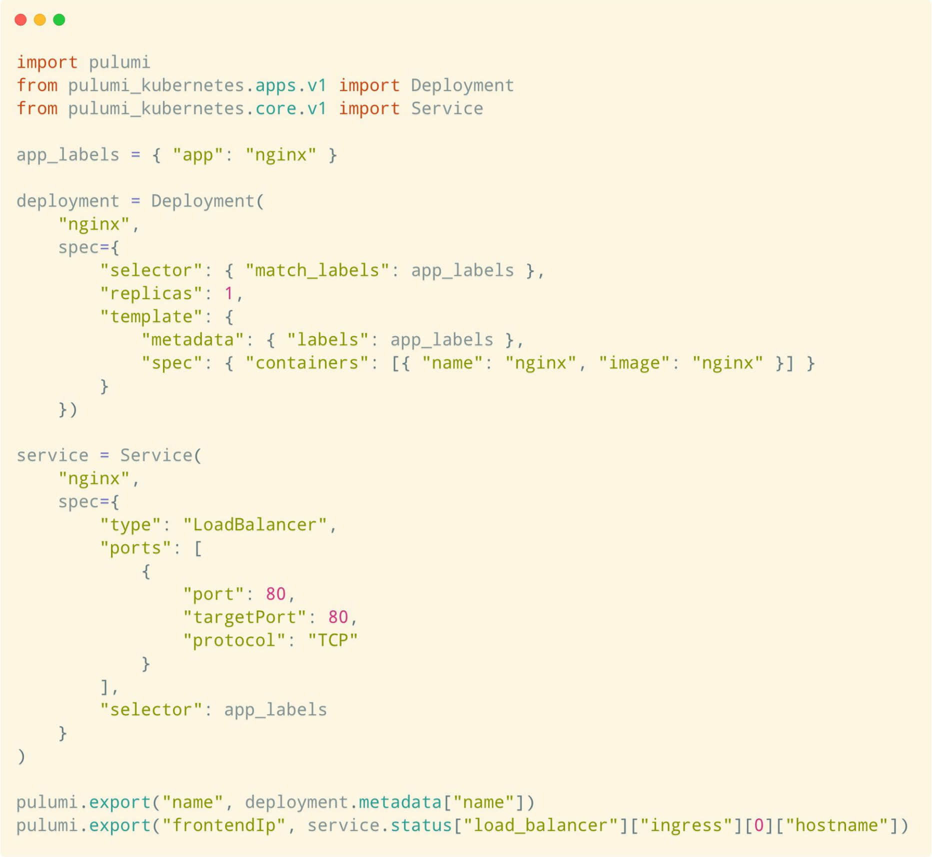
API-compatible with Kubernetes

Interoperable with kubectl

Integration with managed Kubernetes offerings

Compatible with Kubernetes YAML and Helm charts

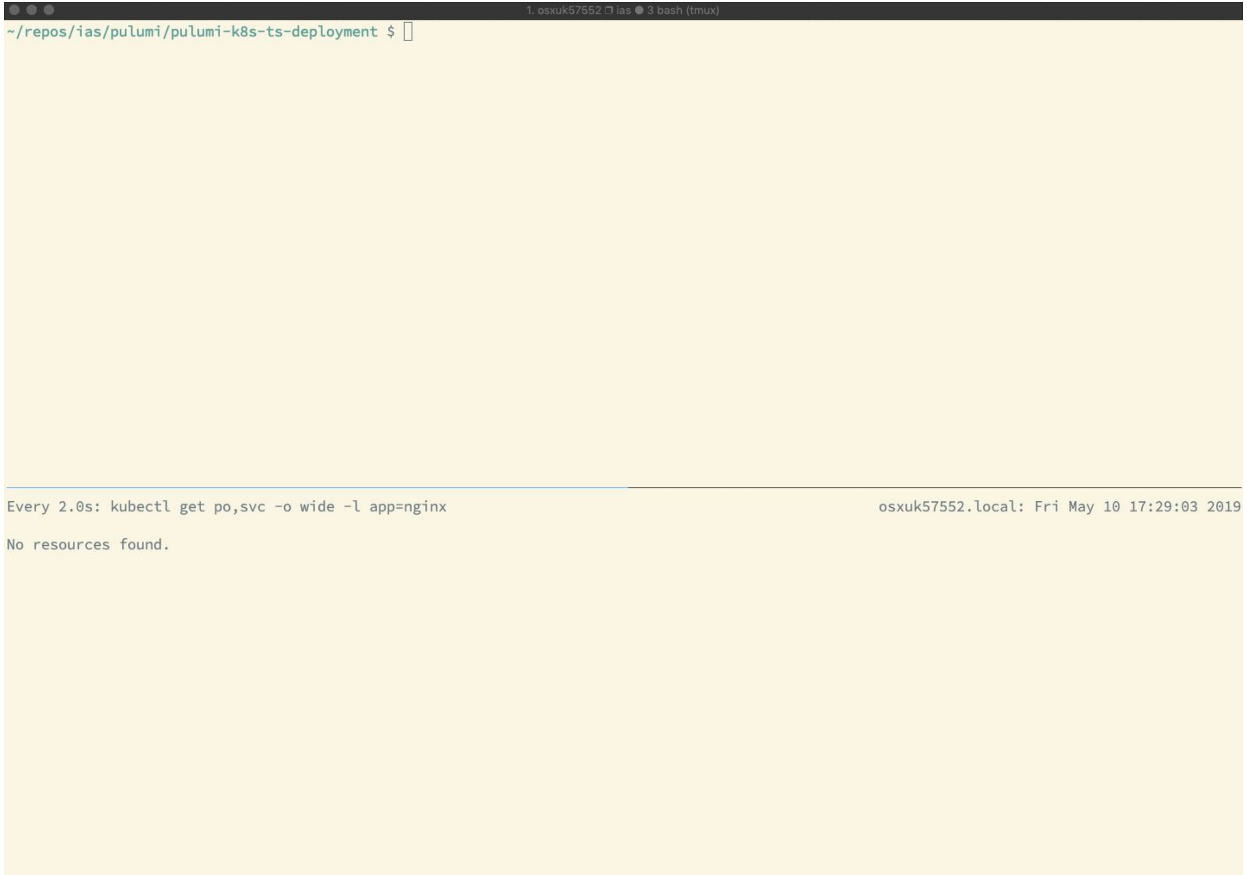
Pulumi Kubernetes Deployments



```
import pulumi
from pulumi_kubernetes.apps.v1 import Deployment
from pulumi_kubernetes.core.v1 import Service

app_labels = { "app": "nginx" }

deployment = Deployment(
    "nginx",
    spec={
        "selector": { "match_labels": app_labels },
        "replicas": 1,
        "template": {
            "metadata": { "labels": app_labels },
            "spec": { "containers": [{ "name": "nginx", "image": "nginx" }] }
        }
    })

service = Service(
    "nginx",
    spec={
        "type": "LoadBalancer",
        "ports": [
            {
                "port": 80,
                "targetPort": 80,
                "protocol": "TCP"
            }
        ],
        "selector": app_labels
    }
)

pulumi.export("name", deployment.metadata["name"])
pulumi.export("frontendIp", service.status["load_balancer"]["ingress"][0]["hostname"])
```

Pulumi Kubernetes Deployments

KubeCon CloudNativeCon Europe 2019

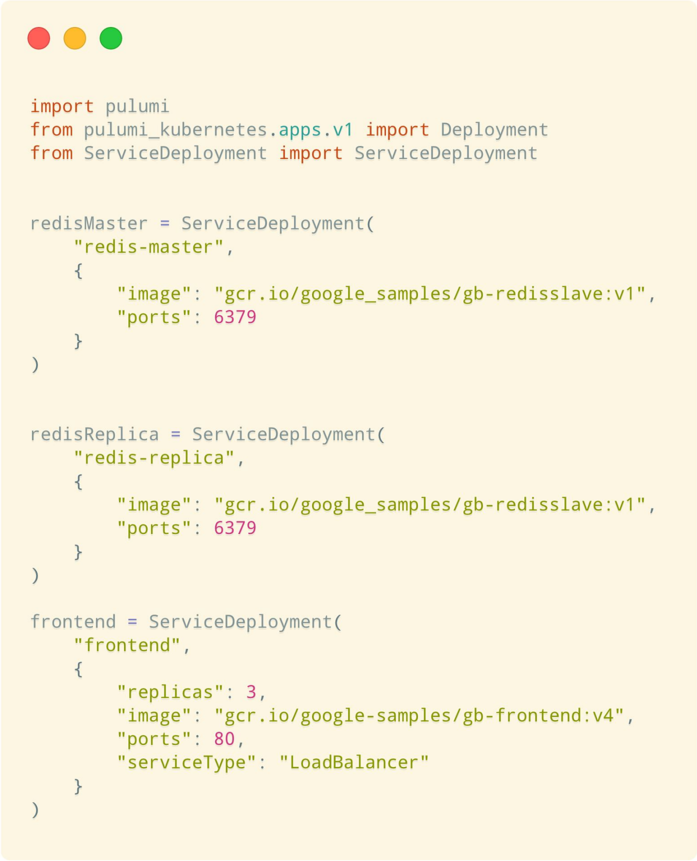


```
~/repos/ias/pulumi/pulumi-k8s-ts-deployment $ 
```

```
Every 2.0s: kubectl get po,svc -o wide -l app=nginx                                           osxuk57552.local: Fri May 10 17:29:03 2019

No resources found.
```

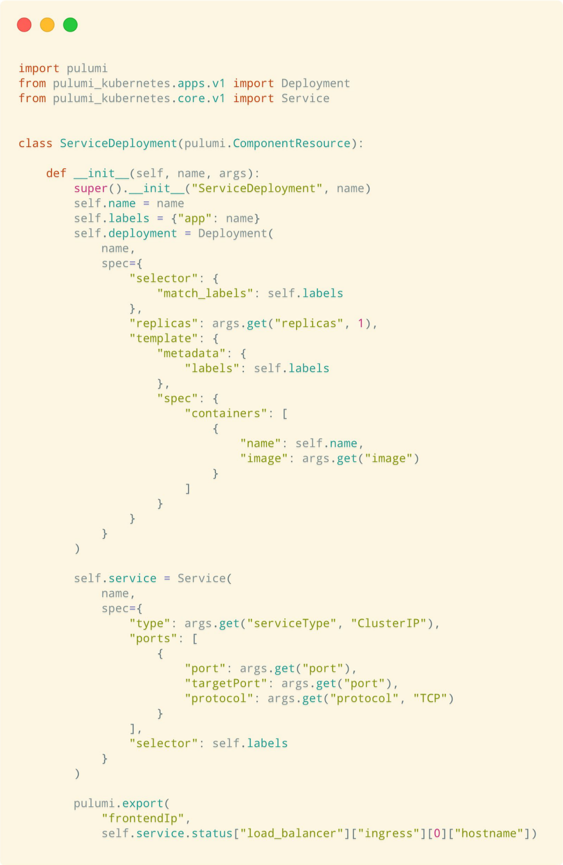
@paulwilljones

Pulumi Kubernetes Abstractions



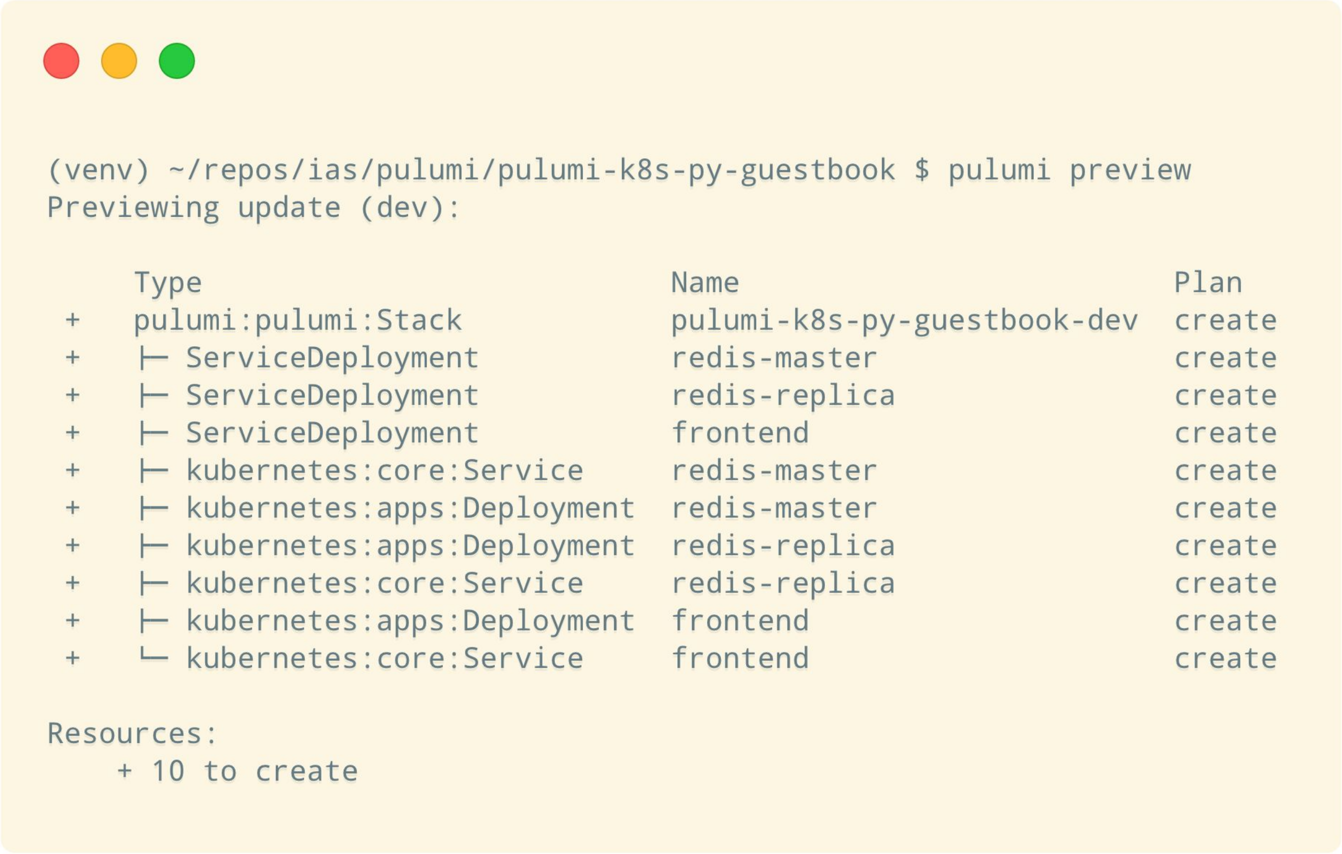
```
import pulumi
from pulumi_kubernetes.apps.v1 import Deployment
from ServiceDeployment import ServiceDeployment


redisMaster = ServiceDeployment(
    "redis-master",
    {
        "image": "gcr.io/google_samples/gb-redisslave:v1",
        "ports": 6379
    }
)


redisReplica = ServiceDeployment(
    "redis-replica",
    {
        "image": "gcr.io/google_samples/gb-redisslave:v1",
        "ports": 6379
    }
)

frontend = ServiceDeployment(
    "frontend",
    {
        "replicas": 3,
        "image": "gcr.io/google-samples/gb-frontend:v4",
        "ports": 80,
        "serviceType": "LoadBalancer"
    }
)
```

Pulumi Kubernetes Abstractions



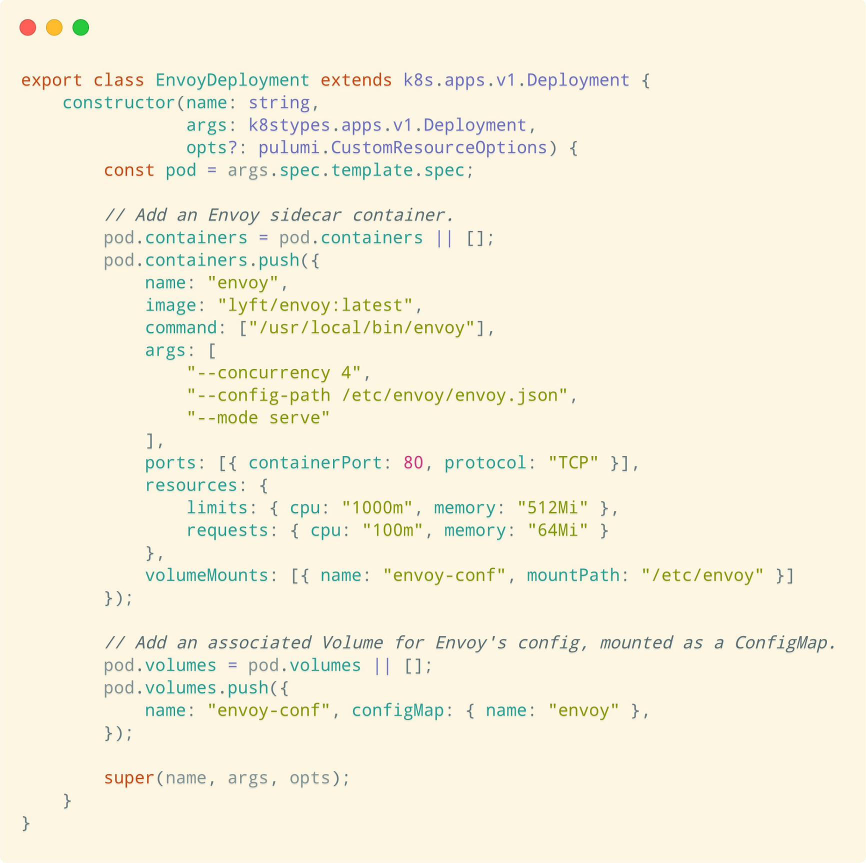
```python
import pulumi
from pulumi_kubernetes.apps.v1 import Deployment
from pulumi_kubernetes.core.v1 import Service


class ServiceDeployment(pulumi.ComponentResource):

    def __init__(self, name, args):
        super().__init__("ServiceDeployment", name)
        self.name = name
        self.labels = {"app": name}
        self.deployment = Deployment(
            name,
            spec={
                "selector": {
                    "match_labels": self.labels
                },
                "replicas": args.get("replicas", 1),
                "template": {
                    "metadata": {
                        "labels": self.labels
                    },
                    "spec": {
                        "containers": [
                            {
                                "name": self.name,
                                "image": args.get("image")
                            }
                        ]
                    }
                }
            }
        )

        self.service = Service(
            name,
            spec={
                "type": args.get("serviceType", "ClusterIP"),
                "ports": [
                    {
                        "port": args.get("port"),
                        "targetPort": args.get("port"),
                        "protocol": args.get("protocol", "TCP")
                    }
                ],
                "selector": self.labels
            }
        )

        pulumi.export(
            "frontendIp",
            self.service.status["load_balancer"]["ingress"][0]["hostname"])
```

Pulumi Kubernetes Abstractions



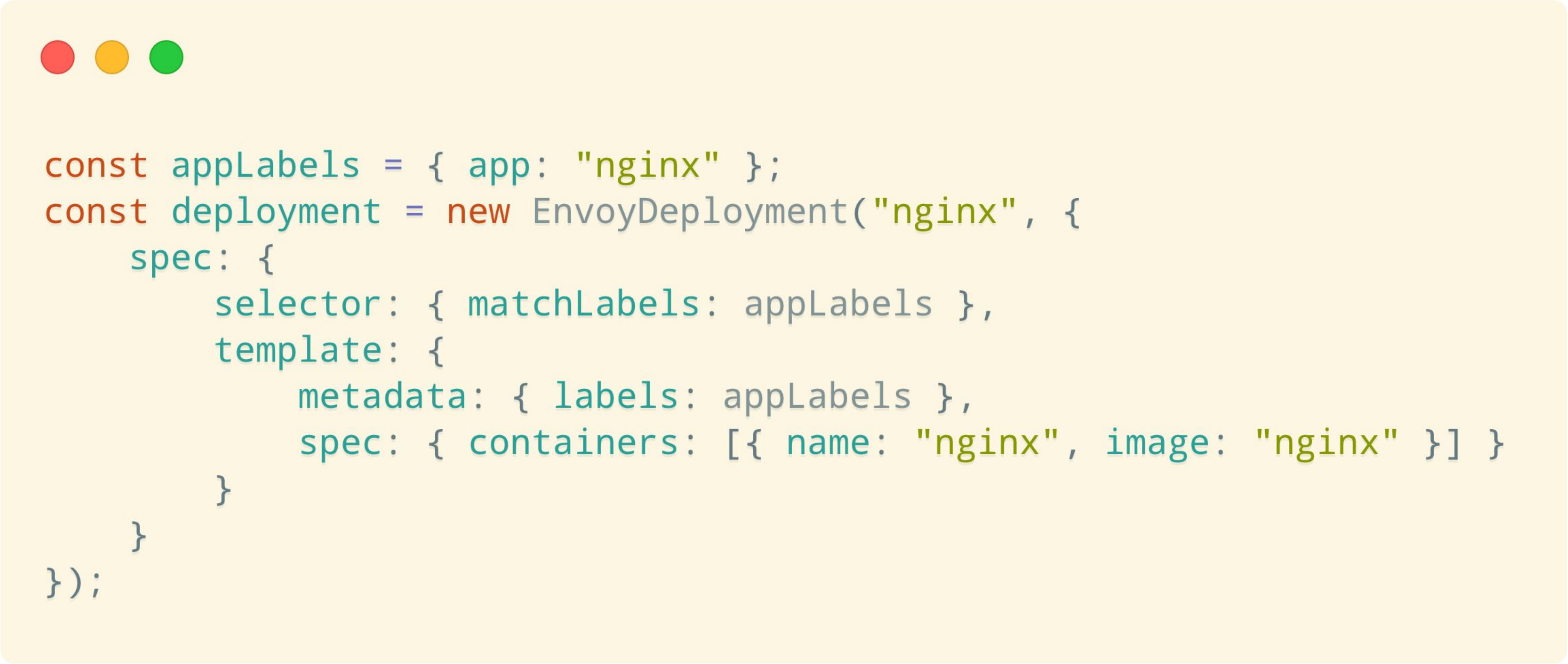
```
(venv) ~/repos/ias/pulumi/pulumi-k8s-py-guestbook $ pulumi preview
Previewing update (dev):

     Type                           Name                         Plan
 +   pulumi:pulumi:Stack            pulumi-k8s-py-guestbook-dev  create
 +   ├─ ServiceDeployment           redis-master                 create
 +   ├─ ServiceDeployment           redis-replica                create
 +   ├─ ServiceDeployment           frontend                     create
 +   ├─ kubernetes:core:Service     redis-master                 create
 +   ├─ kubernetes:apps:Deployment  redis-master                 create
 +   ├─ kubernetes:apps:Deployment  redis-replica                create
 +   ├─ kubernetes:core:Service     redis-replica                create
 +   ├─ kubernetes:apps:Deployment  frontend                     create
 +   └─ kubernetes:core:Service     frontend                     create

Resources:
    + 10 to create
```

Pulumi Kubernetes Abstractions



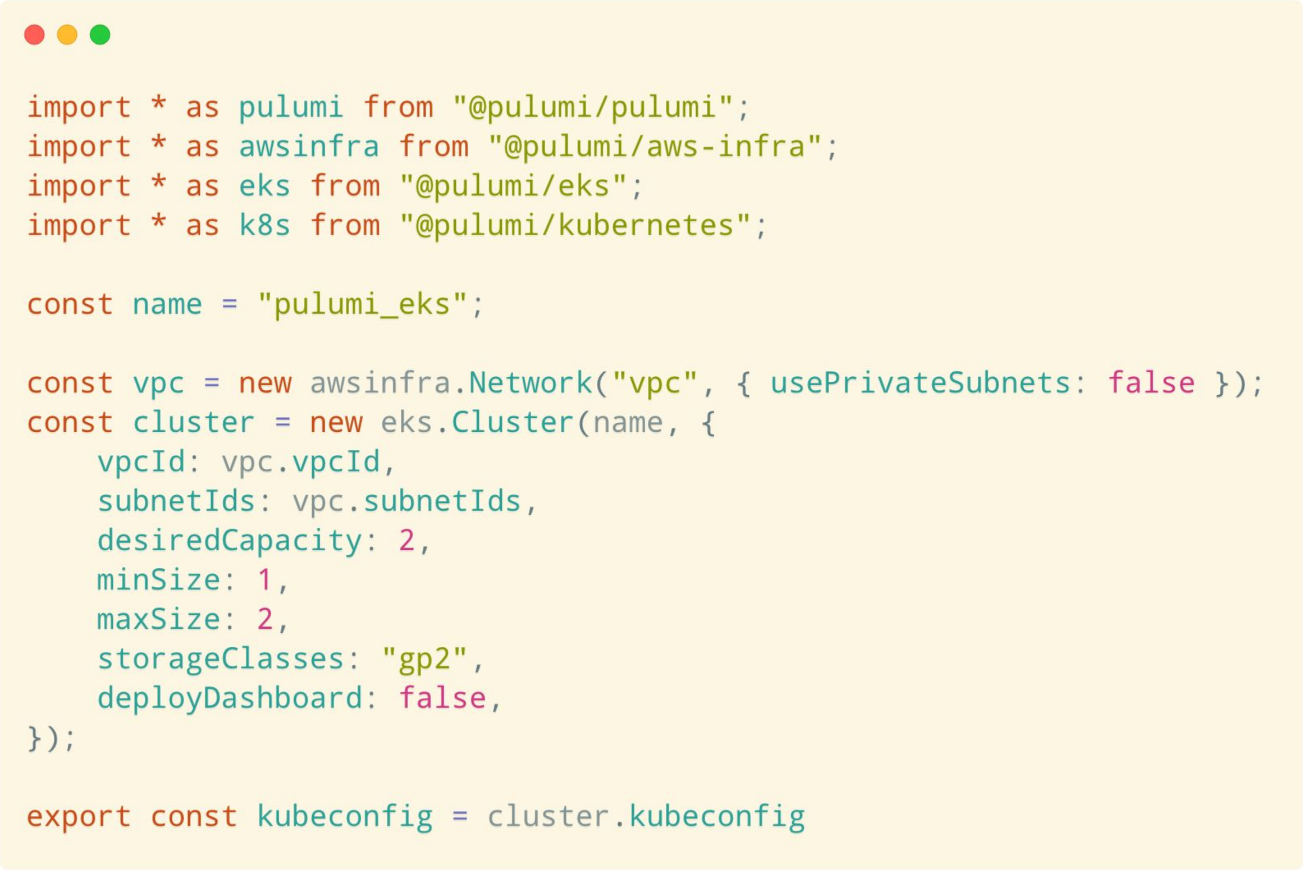
```
export class EnvoyDeployment extends k8s.apps.v1.Deployment {
    constructor(name: string,
                args: k8stypes.apps.v1.Deployment,
                opts?: pulumi.CustomResourceOptions) {
        const pod = args.spec.template.spec;

        // Add an Envoy sidecar container.
        pod.containers = pod.containers || [];
        pod.containers.push({
            name: "envoy",
            image: "lyft/envoy:latest",
            command: ["/usr/local/bin/envoy"],
            args: [
                "--concurrency 4",
                "--config-path /etc/envoy/envoy.json",
                "--mode serve"
            ],
            ports: [{ containerPort: 80, protocol: "TCP" }],
            resources: {
                limits: { cpu: "1000m", memory: "512Mi" },
                requests: { cpu: "100m", memory: "64Mi" }
            },
            volumeMounts: [{ name: "envoy-conf", mountPath: "/etc/envoy" }]
        });

        // Add an associated Volume for Envoy's config, mounted as a ConfigMap.
        pod.volumes = pod.volumes || [];
        pod.volumes.push({
            name: "envoy-conf", configMap: { name: "envoy" },
        });

        super(name, args, opts);
    }
}
```

Pulumi Kubernetes Abstractions

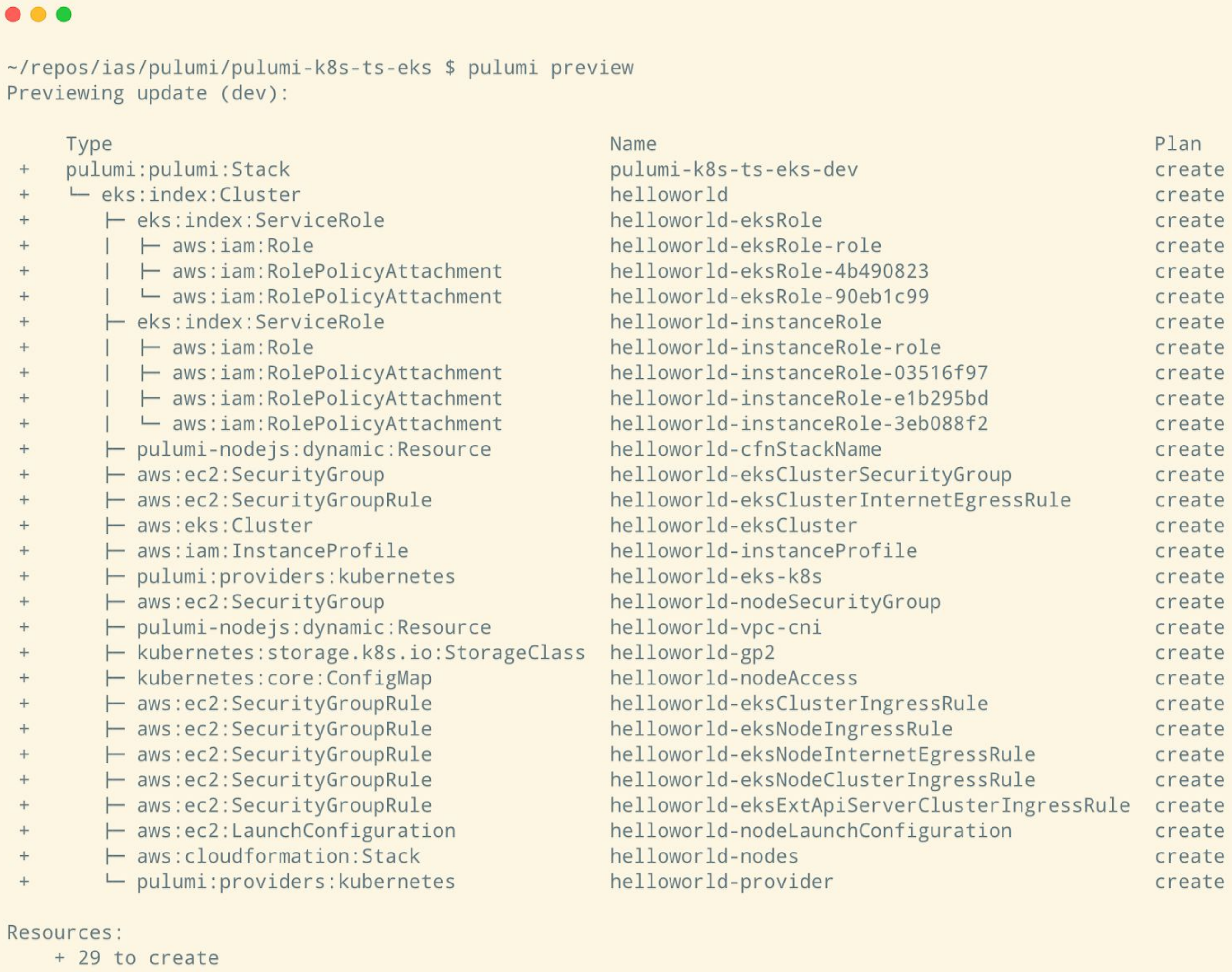


```
const appLabels = { app: "nginx" };
const deployment = new EnvoyDeployment("nginx", {
    spec: {
        selector: { matchLabels: appLabels },
        template: {
            metadata: { labels: appLabels },
            spec: { containers: [{ name: "nginx", image: "nginx" }] }
        }
    }
});
```

Pulumi Kubernetes Clusters - EKS

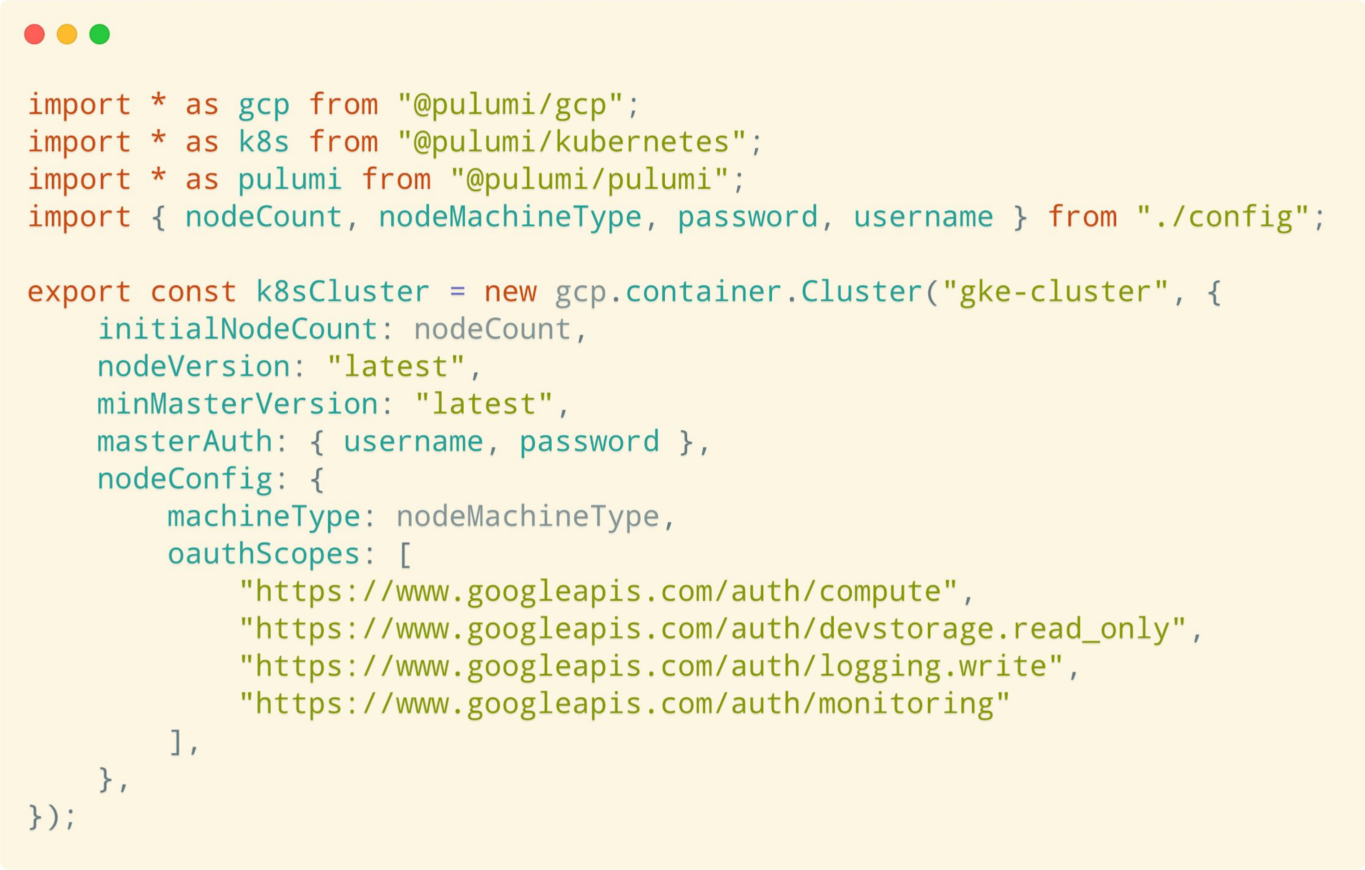


```
import * as pulumi from "@pulumi/pulumi";
import * as awsinfra from "@pulumi/aws-infra";
import * as eks from "@pulumi/eks";
import * as k8s from "@pulumi/kubernetes";

const name = "pulumi_eks";

const vpc = new awsinfra.Network("vpc", { usePrivateSubnets: false });
const cluster = new eks.Cluster(name, {
    vpcId: vpc.vpcId,
    subnetIds: vpc.subnetIds,
    desiredCapacity: 2,
    minSize: 1,
    maxSize: 2,
    storageClasses: "gp2",
    deployDashboard: false,
});

export const kubeconfig = cluster.kubeconfig
```



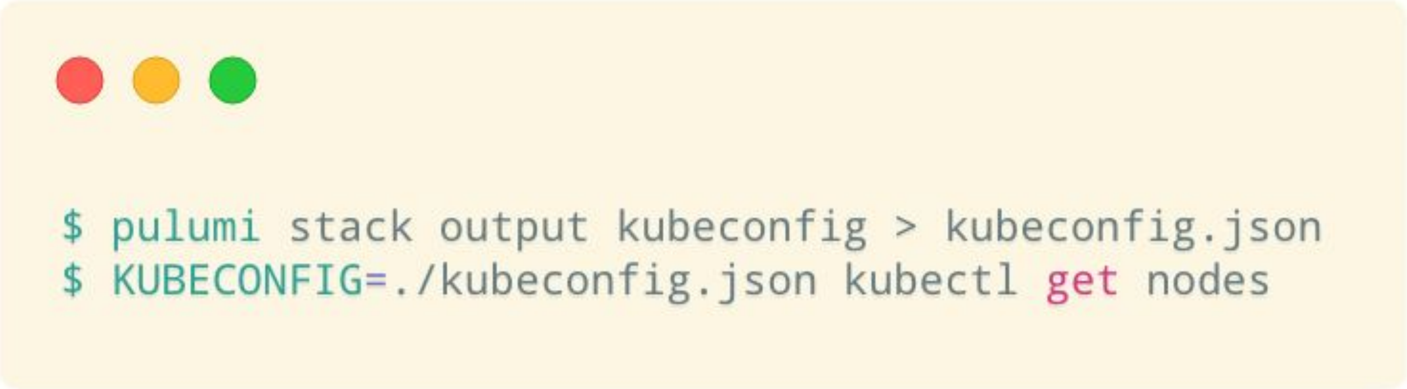
```
~/repos/ias/pulumi/pulumi-k8s-ts-eks $ pulumi preview
Previewing update (dev):

     Type                                      Name                                          Plan
 +   pulumi:pulumi:Stack                       pulumi-k8s-ts-eks-dev                         create
 +   └─ eks:index:Cluster                      helloworld                                    create
 +      ├─ eks:index:ServiceRole               helloworld-eksRole                            create
 +      |  ├─ aws:iam:Role                     helloworld-eksRole-role                       create
 +      |  ├─ aws:iam:RolePolicyAttachment     helloworld-eksRole-4b490823                   create
 +      |  └─ aws:iam:RolePolicyAttachment     helloworld-eksRole-90eb1c99                   create
 +      ├─ eks:index:ServiceRole               helloworld-instanceRole                       create
 +      |  ├─ aws:iam:Role                     helloworld-instanceRole-role                  create
 +      |  ├─ aws:iam:RolePolicyAttachment     helloworld-instanceRole-03516f97              create
 +      |  ├─ aws:iam:RolePolicyAttachment     helloworld-instanceRole-e1b295bd              create
 +      |  └─ aws:iam:RolePolicyAttachment     helloworld-instanceRole-3eb088f2              create
 +      ├─ pulumi-nodejs:dynamic:Resource      helloworld-cfnStackName                       create
 +      ├─ aws:ec2:SecurityGroup               helloworld-eksClusterSecurityGroup            create
 +      ├─ aws:ec2:SecurityGroupRule           helloworld-eksClusterInternetEgressRule       create
 +      ├─ aws:eks:Cluster                     helloworld-eksCluster                         create
 +      ├─ aws:iam:InstanceProfile             helloworld-instanceProfile                    create
 +      ├─ pulumi:providers:kubernetes         helloworld-eks-k8s                            create
 +      ├─ aws:ec2:SecurityGroup               helloworld-nodeSecurityGroup                  create
 +      ├─ pulumi-nodejs:dynamic:Resource      helloworld-vpc-cni                            create
 +      ├─ kubernetes:storage.k8s.io:StorageClass  helloworld-gp2                            create
 +      ├─ kubernetes:core:ConfigMap           helloworld-nodeAccess                         create
 +      ├─ aws:ec2:SecurityGroupRule           helloworld-eksClusterIngressRule              create
 +      ├─ aws:ec2:SecurityGroupRule           helloworld-eksNodeIngressRule                 create
 +      ├─ aws:ec2:SecurityGroupRule           helloworld-eksNodeInternetEgressRule          create
 +      ├─ aws:ec2:SecurityGroupRule           helloworld-eksNodeClusterIngressRule          create
 +      ├─ aws:ec2:SecurityGroupRule           helloworld-eksExtApiServerClusterIngressRule  create
 +      ├─ aws:ec2:LaunchConfiguration         helloworld-nodeLaunchConfiguration            create
 +      ├─ aws:cloudformation:Stack            helloworld-nodes                              create
 +      └─ pulumi:providers:kubernetes         helloworld-provider                           create

Resources:
    + 29 to create
```

Pulumi Kubernetes Clusters - GKE



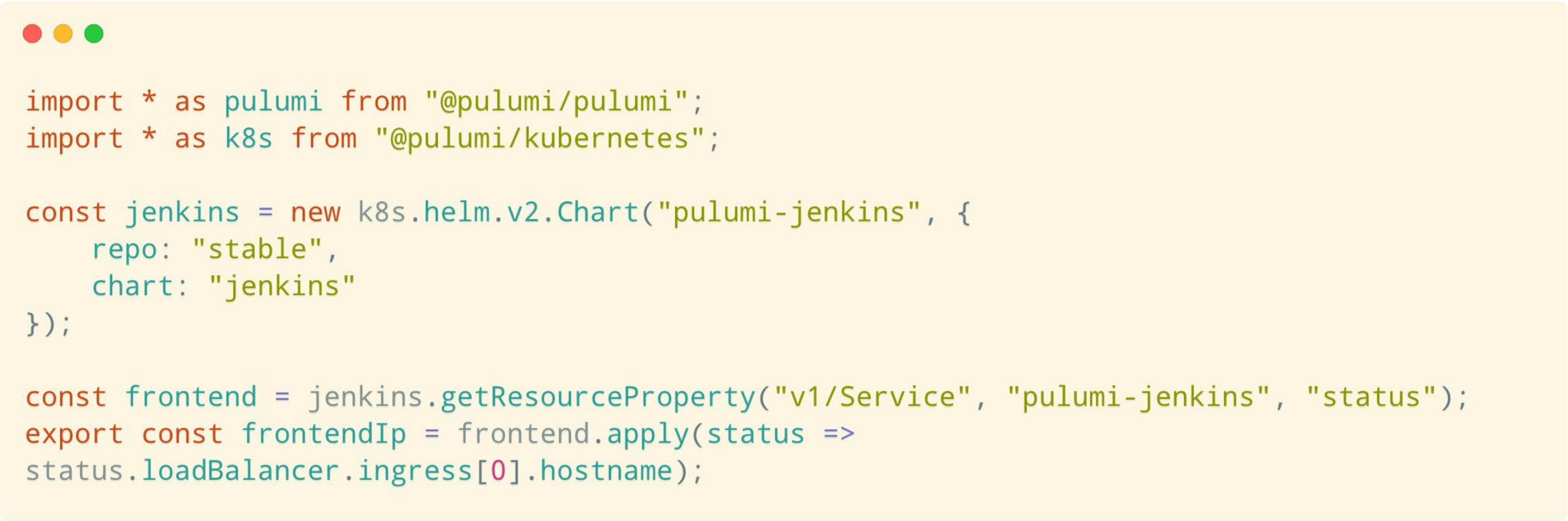
```
import * as gcp from "@pulumi/gcp";
import * as k8s from "@pulumi/kubernetes";
import * as pulumi from "@pulumi/pulumi";
import { nodeCount, nodeMachineType, password, username } from "./config";

export const k8sCluster = new gcp.container.Cluster("gke-cluster", {
    initialNodeCount: nodeCount,
    nodeVersion: "latest",
    minMasterVersion: "latest",
    masterAuth: { username, password },
    nodeConfig: {
        machineType: nodeMachineType,
        oauthScopes: [
            "https://www.googleapis.com/auth/compute",
            "https://www.googleapis.com/auth/devstorage.read_only",
            "https://www.googleapis.com/auth/logging.write",
            "https://www.googleapis.com/auth/monitoring"
        ],
    },
});
```

Pulumi Kubernetes Clusters



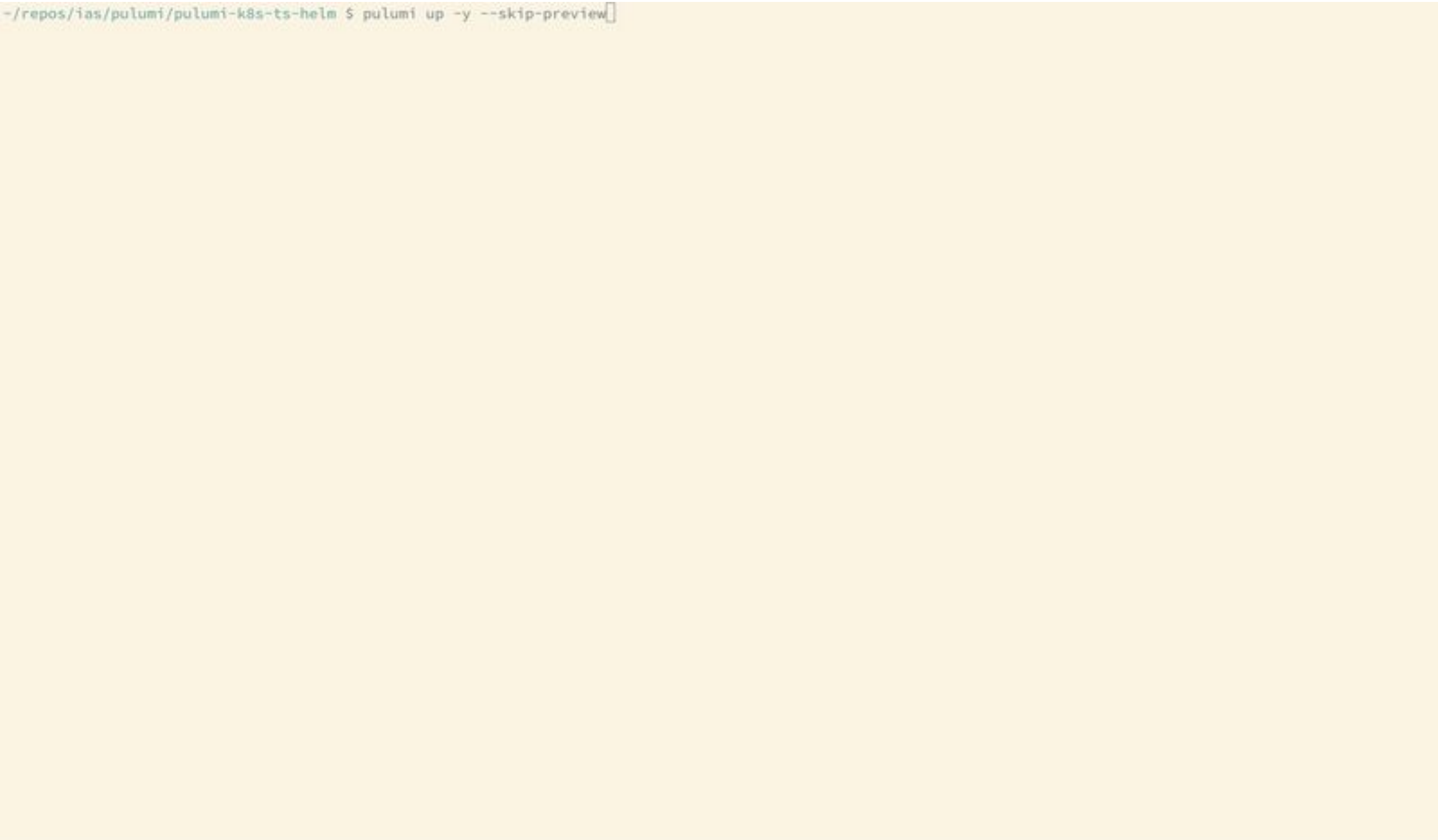
```
$ pulumi stack output kubeconfig > kubeconfig.json
$ KUBECONFIG=./kubeconfig.json kubectl get nodes
```

Pulumi Helm



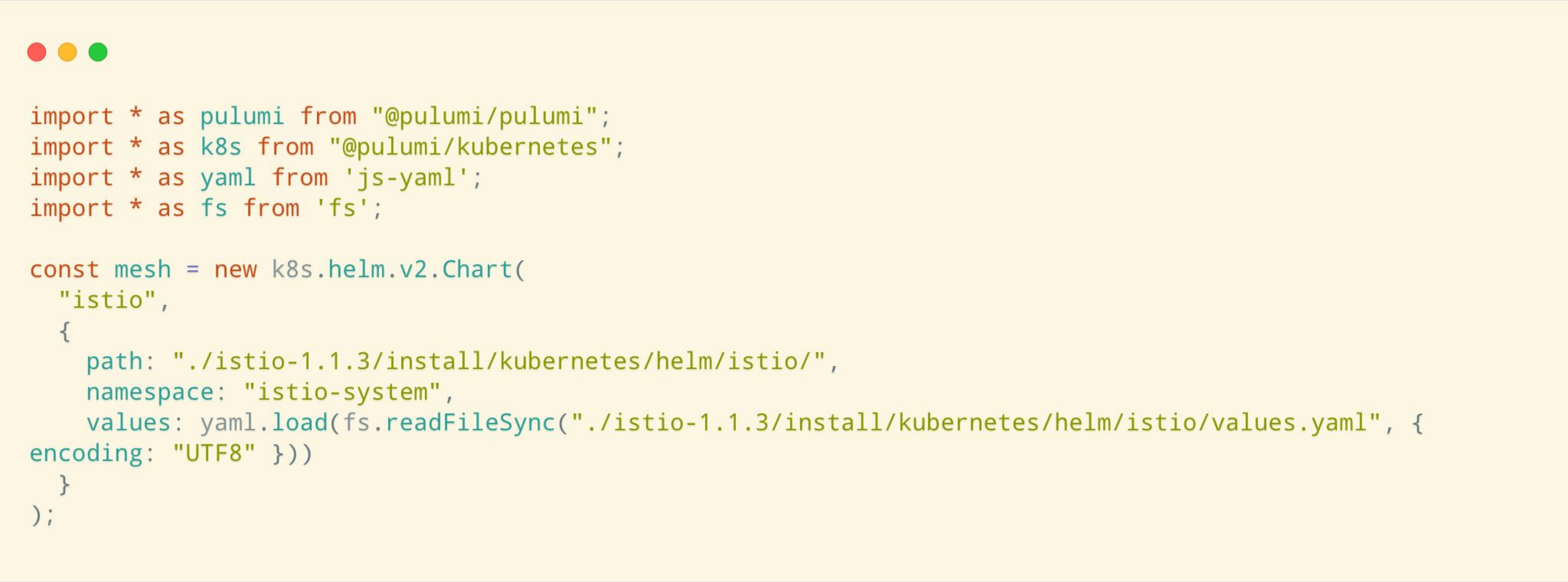
```
import * as pulumi from "@pulumi/pulumi";
import * as k8s from "@pulumi/kubernetes";

const jenkins = new k8s.helm.v2.Chart("pulumi-jenkins", {
    repo: "stable",
    chart: "jenkins"
});

const frontend = jenkins.getResourceProperty("v1/Service", "pulumi-jenkins", "status");
export const frontendIp = frontend.apply(status =>
status.loadBalancer.ingress[0].hostname);
```

Pulumi Helm

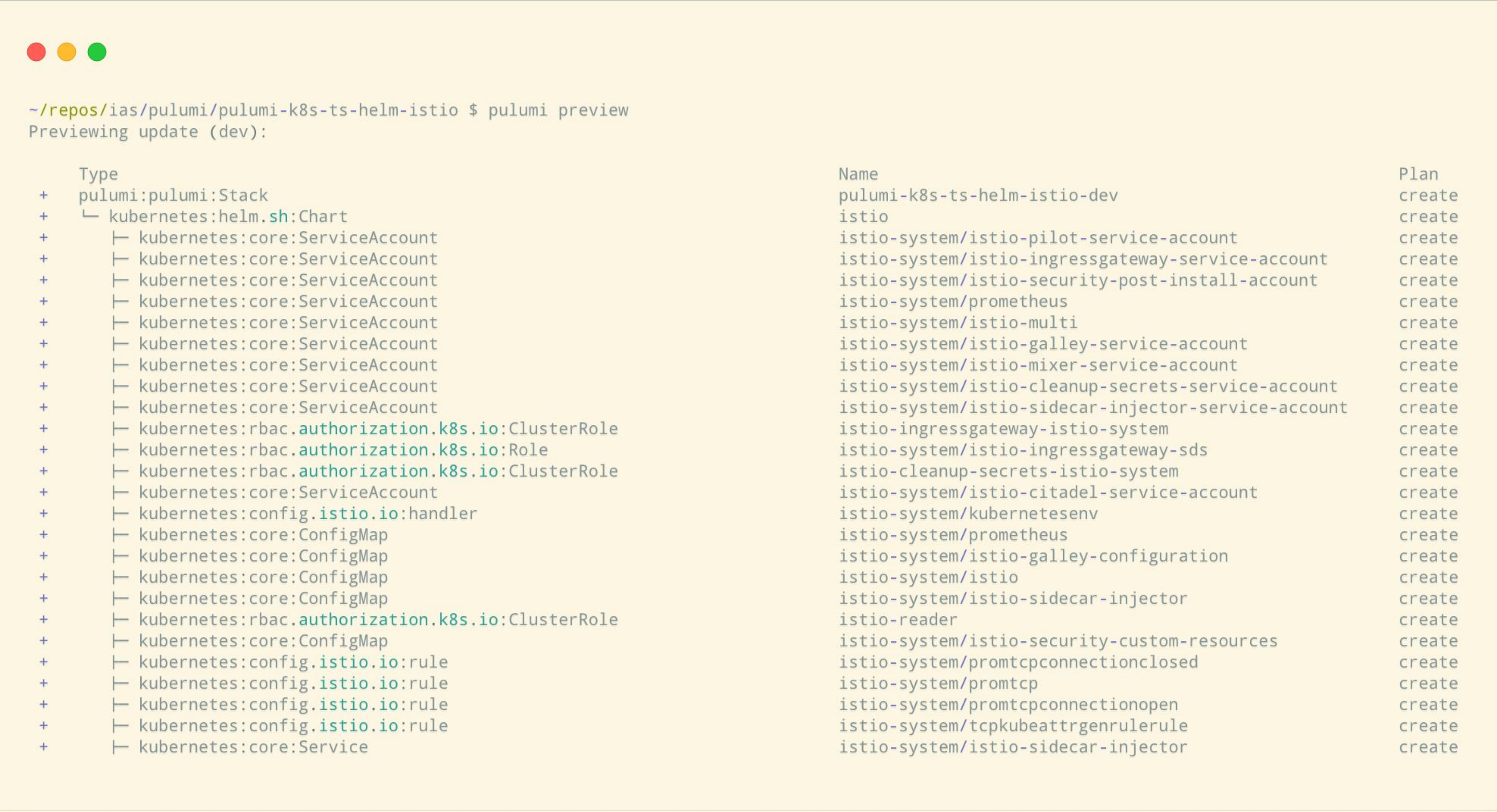


```
~/repos/ias/pulumi/pulumi-k8s-ts-helm $ pulumi up -y --skip-preview
```

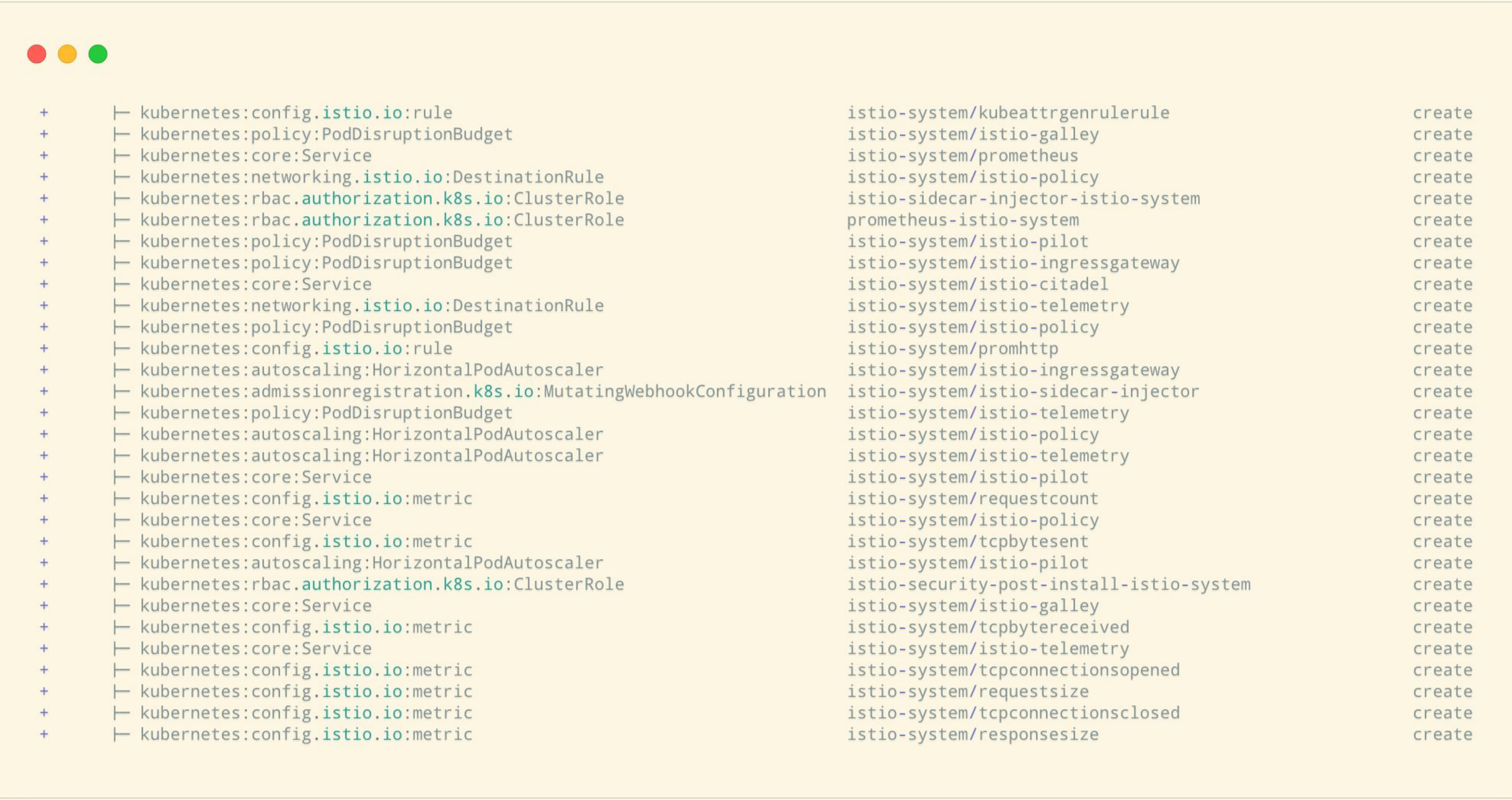
Pulumi Istio



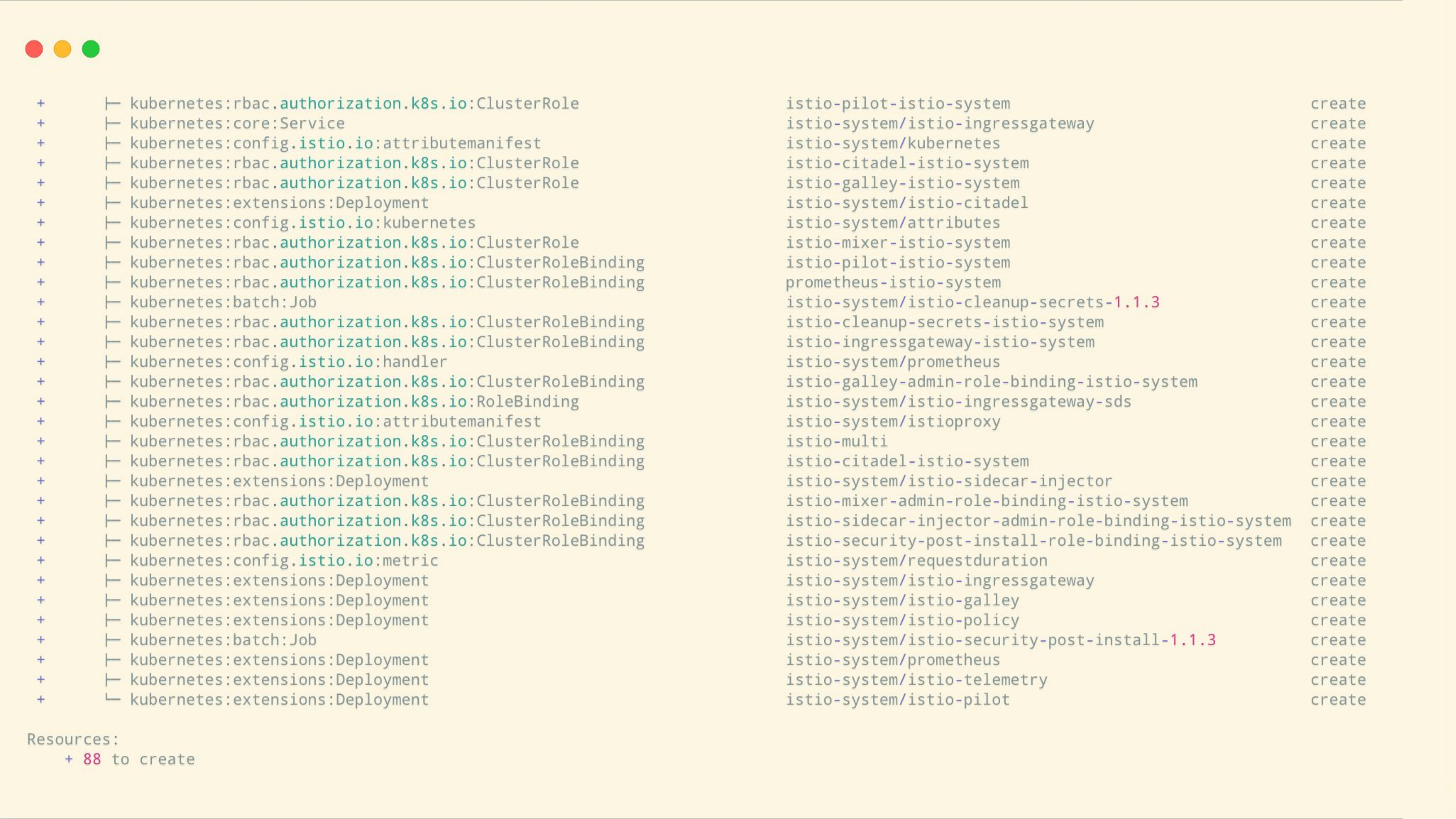
```
import * as pulumi from "@pulumi/pulumi";
import * as k8s from "@pulumi/kubernetes";
import * as yaml from 'js-yaml';
import * as fs from 'fs';

const mesh = new k8s.helm.v2.Chart(
  "istio",
  {
    path: "./istio-1.1.3/install/kubernetes/helm/istio/",
    namespace: "istio-system",
    values: yaml.load(fs.readFileSync("./istio-1.1.3/install/kubernetes/helm/istio/values.yaml", {
encoding: "UTF8" }))
  }
);
```



```
~/repos/ias/pulumi/pulumi-k8s-ts-helm-istio $ pulumi preview
Previewing update (dev):

     Type                                                                 Name                                                   Plan
 +   pulumi:pulumi:Stack                                                  pulumi-k8s-ts-helm-istio-dev                           create
 +   └─ kubernetes:helm.sh:Chart                                          istio                                                  create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-pilot-service-account               create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-ingressgateway-service-account      create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-security-post-install-account       create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/prometheus                                create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-multi                               create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-galley-service-account              create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-mixer-service-account               create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-cleanup-secrets-service-account     create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-sidecar-injector-service-account    create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole               istio-ingressgateway-istio-system                      create
 +      ├─ kubernetes:rbac.authorization.k8s.io:Role                      istio-system/istio-ingressgateway-sds                  create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole               istio-cleanup-secrets-istio-system                     create
 +      ├─ kubernetes:core:ServiceAccount                                 istio-system/istio-citadel-service-account             create
 +      ├─ kubernetes:config.istio.io:handler                             istio-system/kubernetesenv                             create
 +      ├─ kubernetes:core:ConfigMap                                      istio-system/prometheus                                create
 +      ├─ kubernetes:core:ConfigMap                                      istio-system/istio-galley-configuration                create
 +      ├─ kubernetes:core:ConfigMap                                      istio-system/istio                                     create
 +      ├─ kubernetes:core:ConfigMap                                      istio-system/istio-sidecar-injector                    create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole               istio-reader                                           create
 +      ├─ kubernetes:core:ConfigMap                                      istio-system/istio-security-custom-resources           create
 +      ├─ kubernetes:config.istio.io:rule                                istio-system/promtcpconnectionclosed                   create
 +      ├─ kubernetes:config.istio.io:rule                                istio-system/promtcp                                   create
 +      ├─ kubernetes:config.istio.io:rule                                istio-system/promtcpconnectionopen                     create
 +      ├─ kubernetes:config.istio.io:rule                                istio-system/tcpkubeattrgenrulerule                    create
 +      ├─ kubernetes:core:Service                                        istio-system/istio-sidecar-injector                    create
```



```
+       ├─ kubernetes:config.istio.io:rule                                       istio-system/kubeattrgenrulerule                        create
+       ├─ kubernetes:policy:PodDisruptionBudget                                 istio-system/istio-galley                               create
+       ├─ kubernetes:core:Service                                               istio-system/prometheus                                 create
+       ├─ kubernetes:networking.istio.io:DestinationRule                        istio-system/istio-policy                               create
+       ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      istio-sidecar-injector-istio-system                     create
+       ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      prometheus-istio-system                                 create
+       ├─ kubernetes:policy:PodDisruptionBudget                                 istio-system/istio-pilot                                create
+       ├─ kubernetes:policy:PodDisruptionBudget                                 istio-system/istio-ingressgateway                       create
+       ├─ kubernetes:core:Service                                               istio-system/istio-citadel                              create
+       ├─ kubernetes:networking.istio.io:DestinationRule                        istio-system/istio-telemetry                            create
+       ├─ kubernetes:policy:PodDisruptionBudget                                 istio-system/istio-policy                               create
+       ├─ kubernetes:config.istio.io:rule                                       istio-system/promhttp                                   create
+       ├─ kubernetes:autoscaling:HorizontalPodAutoscaler                        istio-system/istio-ingressgateway                       create
+       ├─ kubernetes:admissionregistration.k8s.io:MutatingWebhookConfiguration  istio-system/istio-sidecar-injector                     create
+       ├─ kubernetes:policy:PodDisruptionBudget                                 istio-system/istio-telemetry                            create
+       ├─ kubernetes:autoscaling:HorizontalPodAutoscaler                        istio-system/istio-policy                               create
+       ├─ kubernetes:autoscaling:HorizontalPodAutoscaler                        istio-system/istio-telemetry                            create
+       ├─ kubernetes:core:Service                                               istio-system/istio-pilot                                create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/requestcount                               create
+       ├─ kubernetes:core:Service                                               istio-system/istio-policy                               create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/tcpbytesent                                create
+       ├─ kubernetes:autoscaling:HorizontalPodAutoscaler                        istio-system/istio-pilot                                create
+       ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      istio-security-post-install-istio-system                create
+       ├─ kubernetes:core:Service                                               istio-system/istio-galley                               create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/tcpbytereceived                            create
+       ├─ kubernetes:core:Service                                               istio-system/istio-telemetry                            create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/tcpconnectionsopened                       create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/requestsize                                create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/tcpconnectionsclosed                       create
+       ├─ kubernetes:config.istio.io:metric                                     istio-system/responsesize                               create
```



```
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      istio-pilot-istio-system                                create
 +      ├─ kubernetes:core:Service                                               istio-system/istio-ingressgateway                       create
 +      ├─ kubernetes:config.istio.io:attributemanifest                          istio-system/kubernetes                                 create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      istio-citadel-istio-system                              create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      istio-galley-istio-system                               create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/istio-citadel                              create
 +      ├─ kubernetes:config.istio.io:kubernetes                                 istio-system/attributes                                 create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRole                      istio-mixer-istio-system                                create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-pilot-istio-system                                create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               prometheus-istio-system                                 create
 +      ├─ kubernetes:batch:Job                                                  istio-system/istio-cleanup-secrets-1.1.3                create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-cleanup-secrets-istio-system                      create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-ingressgateway-istio-system                       create
 +      ├─ kubernetes:config.istio.io:handler                                    istio-system/prometheus                                 create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-galley-admin-role-binding-istio-system            create
 +      ├─ kubernetes:rbac.authorization.k8s.io:RoleBinding                      istio-system/istio-ingressgateway-sds                   create
 +      ├─ kubernetes:config.istio.io:attributemanifest                          istio-system/istioproxy                                 create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-multi                                             create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-citadel-istio-system                              create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/istio-sidecar-injector                     create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-mixer-admin-role-binding-istio-system             create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-sidecar-injector-admin-role-binding-istio-system  create
 +      ├─ kubernetes:rbac.authorization.k8s.io:ClusterRoleBinding               istio-security-post-install-role-binding-istio-system   create
 +      ├─ kubernetes:config.istio.io:metric                                     istio-system/requestduration                            create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/istio-ingressgateway                       create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/istio-galley                               create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/istio-policy                               create
 +      ├─ kubernetes:batch:Job                                                  istio-system/istio-security-post-install-1.1.3          create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/prometheus                                 create
 +      ├─ kubernetes:extensions:Deployment                                      istio-system/istio-telemetry                            create
 +      └─ kubernetes:extensions:Deployment                                      istio-system/istio-pilot                                create

Resources:
    + 88 to create
```

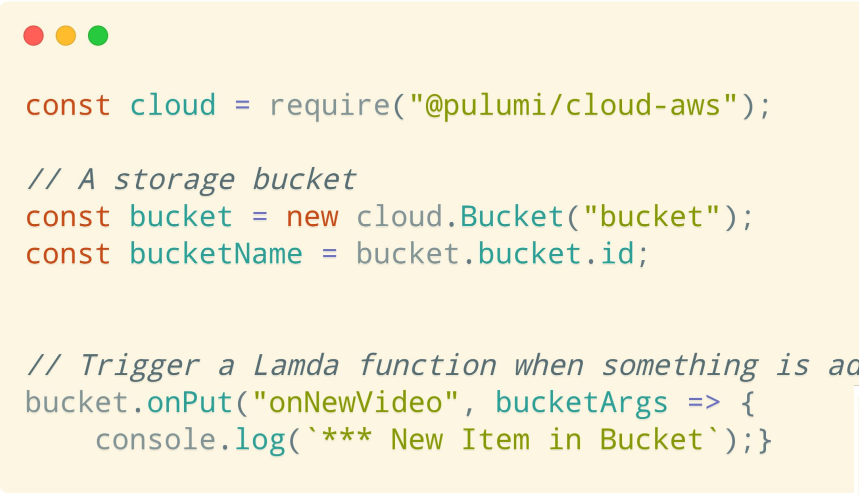
Pulumi Serverless

Adopting real languages for infrastructure code facilitates a path to a coexistence of infrastructure and application code

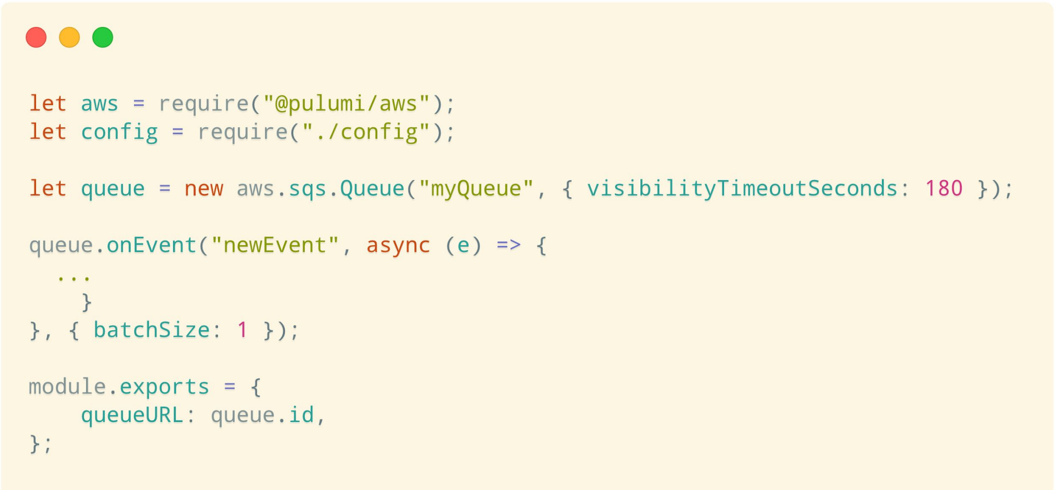
Serverless platform integrations can just be written in real languages, offering a flexible and simple path to serverless

We can create resources, and then wire up event handlers, just like normal event-driven programming

Pulumi Serverless



```
const cloud = require("@pulumi/cloud-aws");

// A storage bucket
const bucket = new cloud.Bucket("bucket");
const bucketName = bucket.bucket.id;


// Trigger a Lamda function when something is added
bucket.onPut("onNewVideo", bucketArgs => {
    console.log(`*** New Item in Bucket`);}
```



```
let aws = require("@pulumi/aws");
let config = require("./config");

let queue = new aws.sqs.Queue("myQueue", { visibilityTimeoutSeconds: 180 });

queue.onEvent("newEvent", async (e) => {
  ...
    }
}, { batchSize: 1 });

module.exports = {
    queueURL: queue.id,
};
```

Infrastructure Testing

Prove that infrastructure works as intended

Prove the infrastructure is functioning correctly between changes

Prove that infrastructure conforms to predetermined specifications

Infrastructure Testing

Unit testing

`moto`

Linting / Static Analysis

`cfn_nag / cfn-lint / cfripper / Terrascan / TFLint`

Mocking

`localstack`

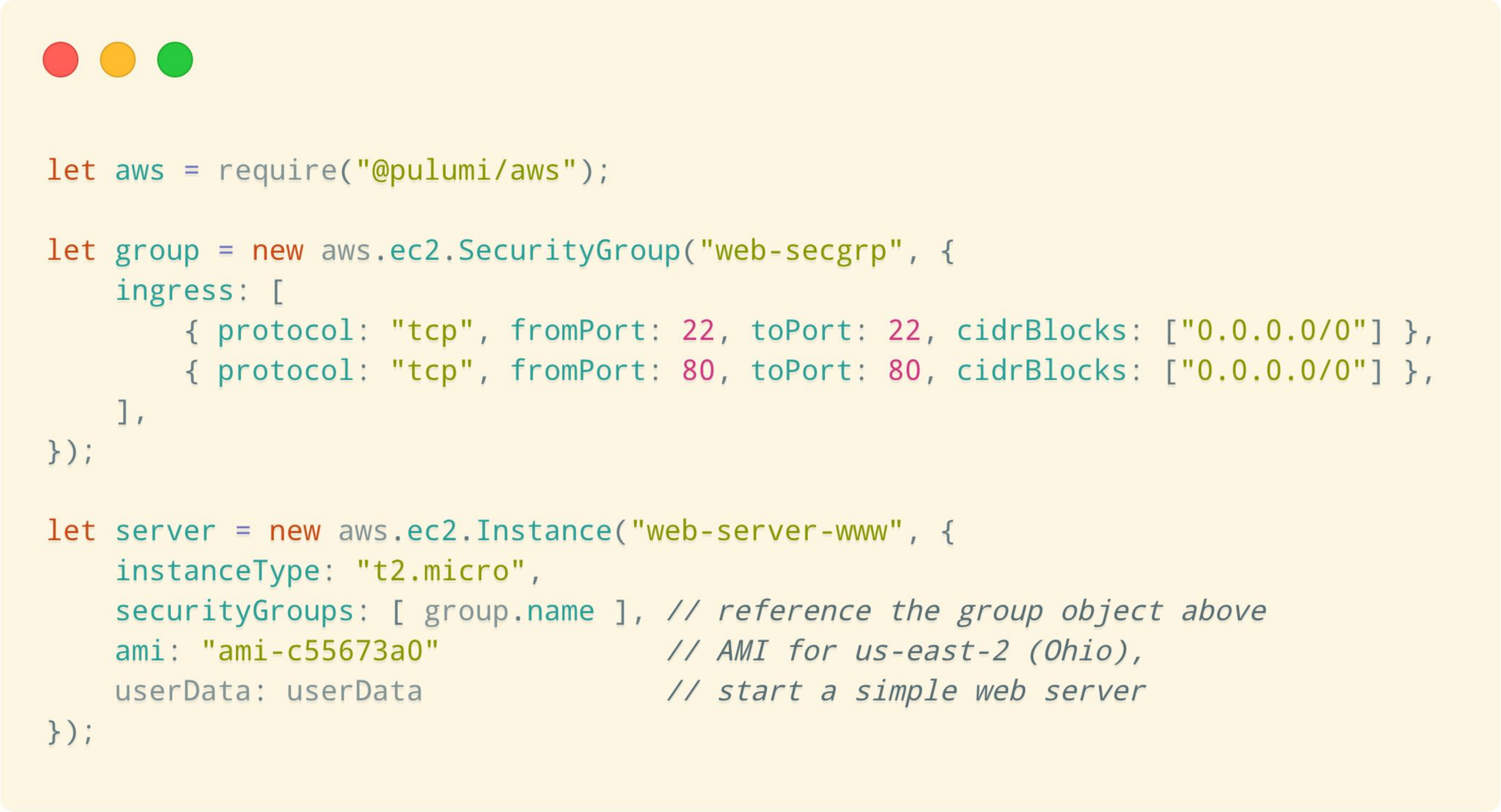
Infrastructure Assertions

`bats`

`awspec / serverspec`

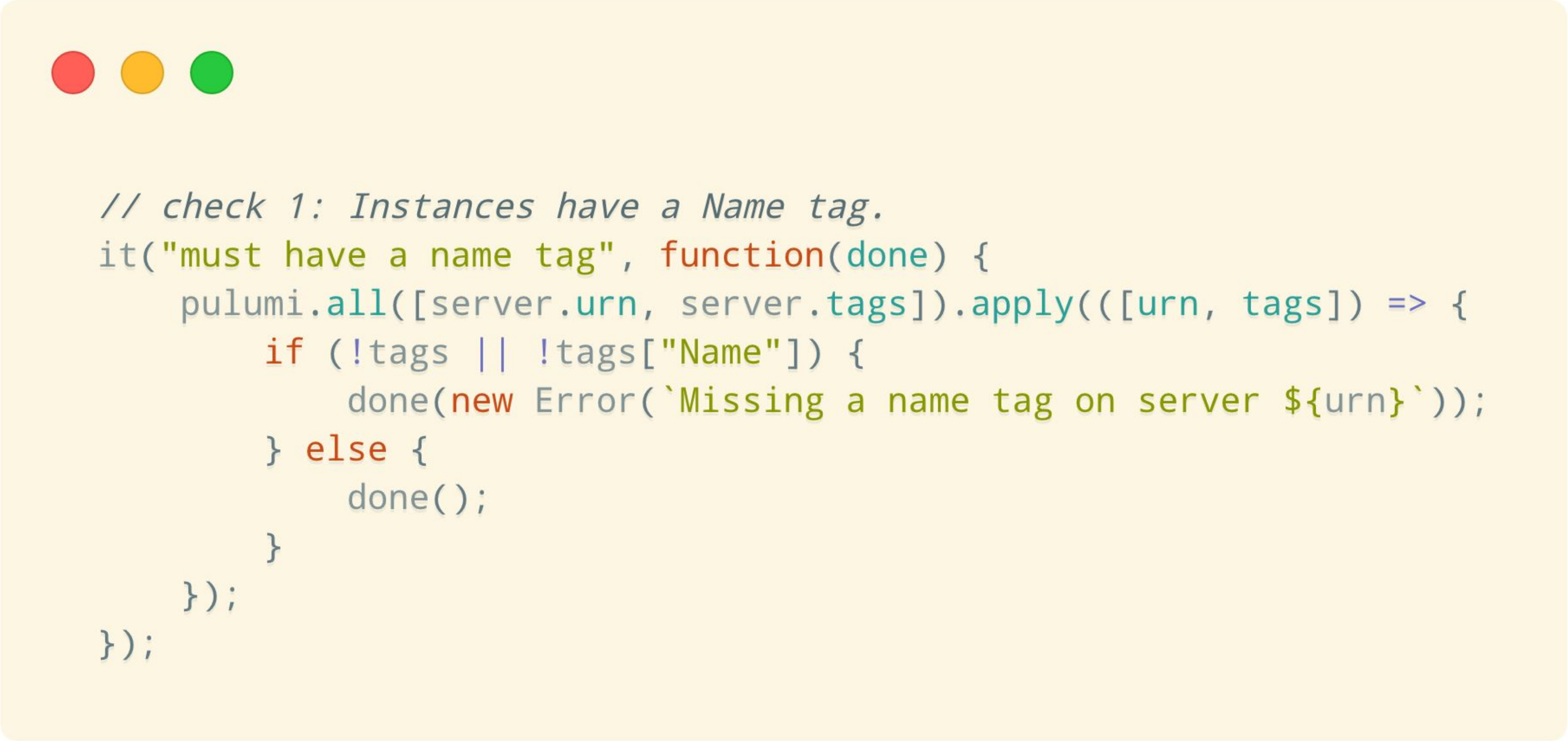
`goss`

Pulumi Testing



```
let aws = require("@pulumi/aws");

let group = new aws.ec2.SecurityGroup("web-secgrp", {
    ingress: [
        { protocol: "tcp", fromPort: 22, toPort: 22, cidrBlocks: ["0.0.0.0/0"] },
        { protocol: "tcp", fromPort: 80, toPort: 80, cidrBlocks: ["0.0.0.0/0"] },
    ],
});

let server = new aws.ec2.Instance("web-server-www", {
    instanceType: "t2.micro",
    securityGroups: [ group.name ], // reference the group object above
    ami: "ami-c55673a0"             // AMI for us-east-2 (Ohio),
    userData: userData              // start a simple web server
});
```

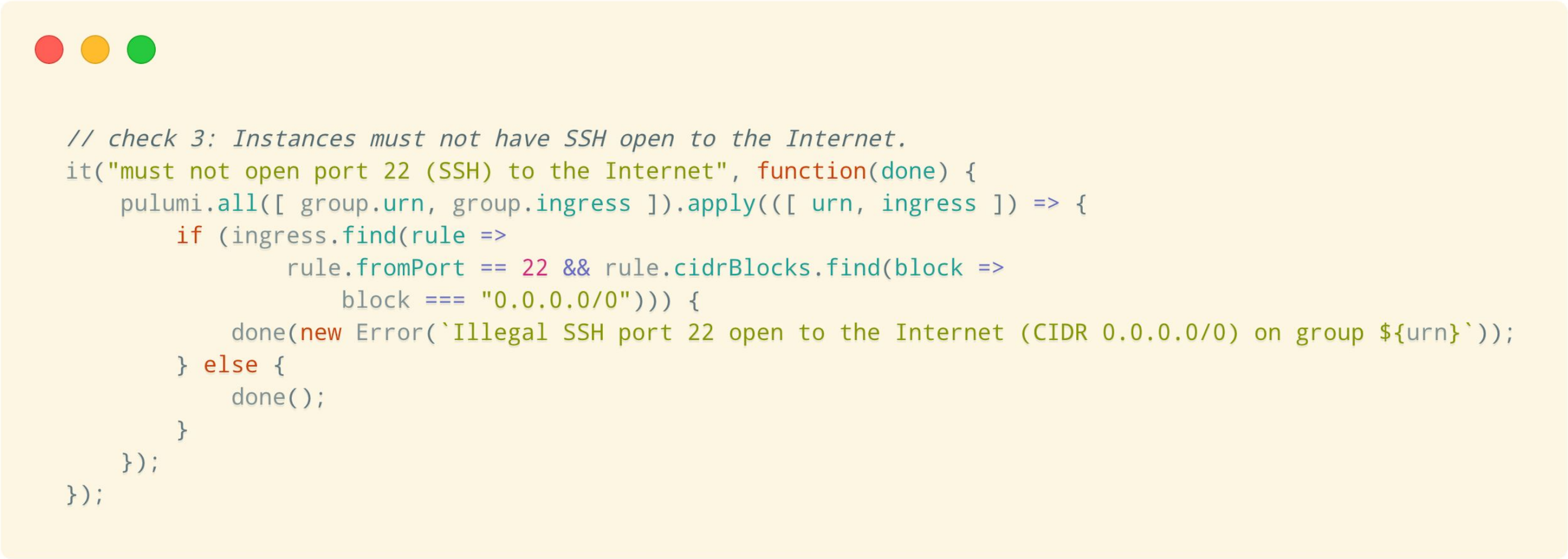
Pulumi Testing



```
// check 1: Instances have a Name tag.
it("must have a name tag", function(done) {
    pulumi.all([server.urn, server.tags]).apply(([urn, tags]) => {
        if (!tags || !tags["Name"]) {
            done(new Error(`Missing a name tag on server ${urn}`));
        } else {
            done();
        }
    });
});
```

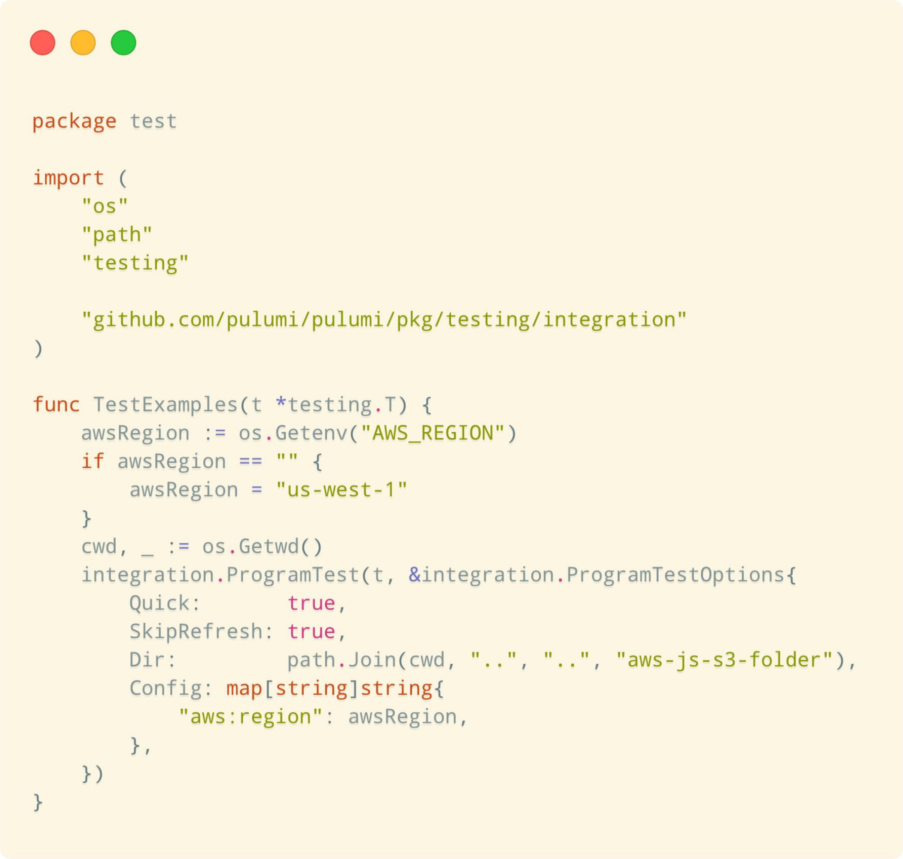
es

Pulumi Testing

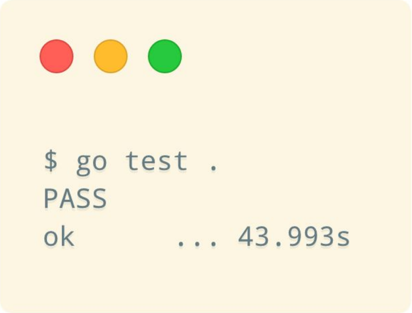


```
// check 3: Instances must not have SSH open to the Internet.
it("must not open port 22 (SSH) to the Internet", function(done) {
    pulumi.all([ group.urn, group.ingress ]).apply(([ urn, ingress ]) => {
        if (ingress.find(rule =>
                rule.fromPort == 22 && rule.cidrBlocks.find(block =>
                    block === "0.0.0.0/0"))) {
            done(new Error(`Illegal SSH port 22 open to the Internet (CIDR 0.0.0.0/0) on group ${urn}`));
        } else {
            done();
        }
    });
});
```

Pulumi Testing



```go
package test

import (
    "os"
    "path"
    "testing"

    "github.com/pulumi/pulumi/pkg/testing/integration"
)

func TestExamples(t *testing.T) {
    awsRegion := os.Getenv("AWS_REGION")
    if awsRegion == "" {
        awsRegion = "us-west-1"
    }
    cwd, _ := os.Getwd()
    integration.ProgramTest(t, &integration.ProgramTestOptions{
        Quick:       true,
        SkipRefresh: true,
        Dir:         path.Join(cwd, "..", "..", "aws-js-s3-folder"),
        Config: map[string]string{
            "aws:region": awsRegion,
        },
    })
}
```



```
$ go test .
PASS
ok      ... 43.993s
```

Pulumi Findings

Cloud agnosticism provides flexibility and portability

Significant boilerplate reduction

Abstraction aids standardisation

SDK feature parity

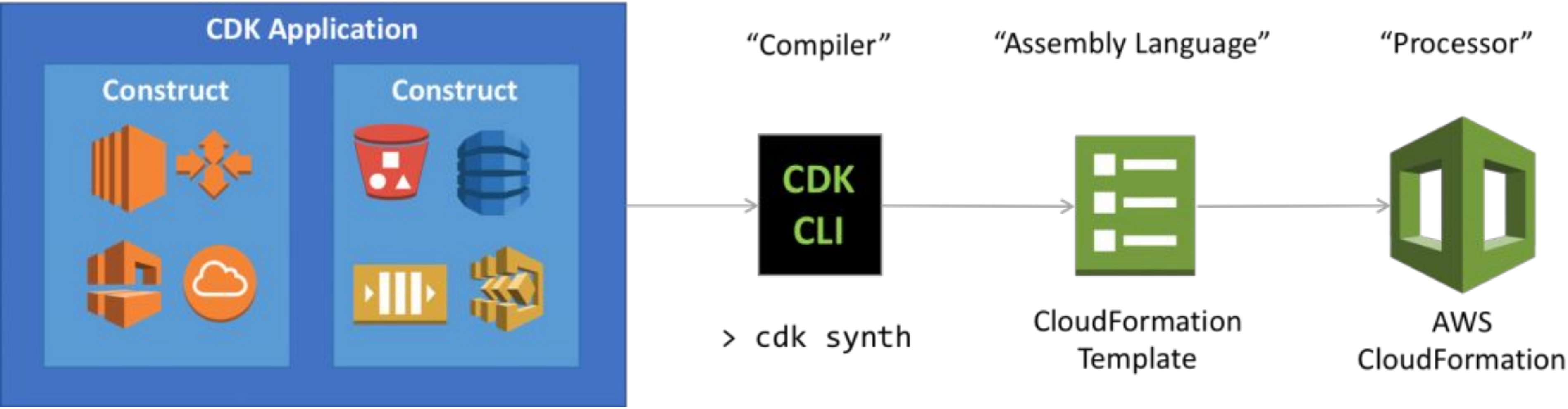
AWS CDK

AWS CDK

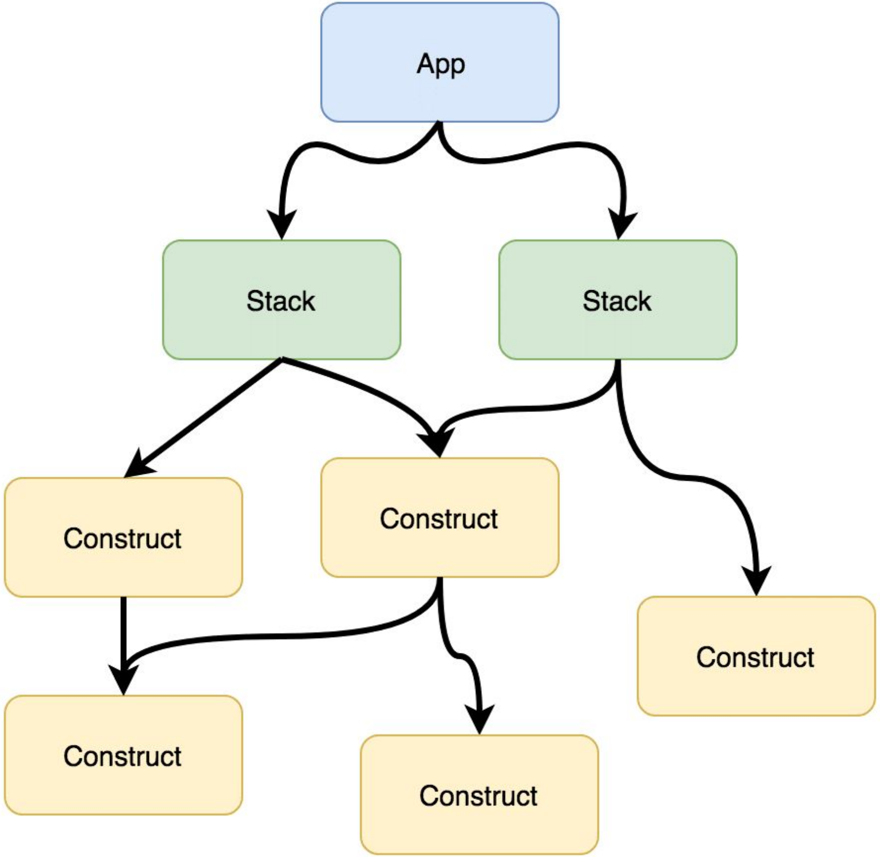
The AWS CDK Toolkit is a command-line tool for interacting with CDK apps. It allows developers to synthesize stacks into CloudFormation Templates, then deploy stacks to development AWS accounts and "diff" against a deployed stack to understand the impact of a code change.

The AWS Construct Library includes a module for each AWS service with constructs that offer rich APIs that encapsulate the details of how to use AWS. The AWS Construct Library aims to reduce the complexity and glue-logic required when integrating various AWS services to achieve your goals on AWS.

AWS CDK

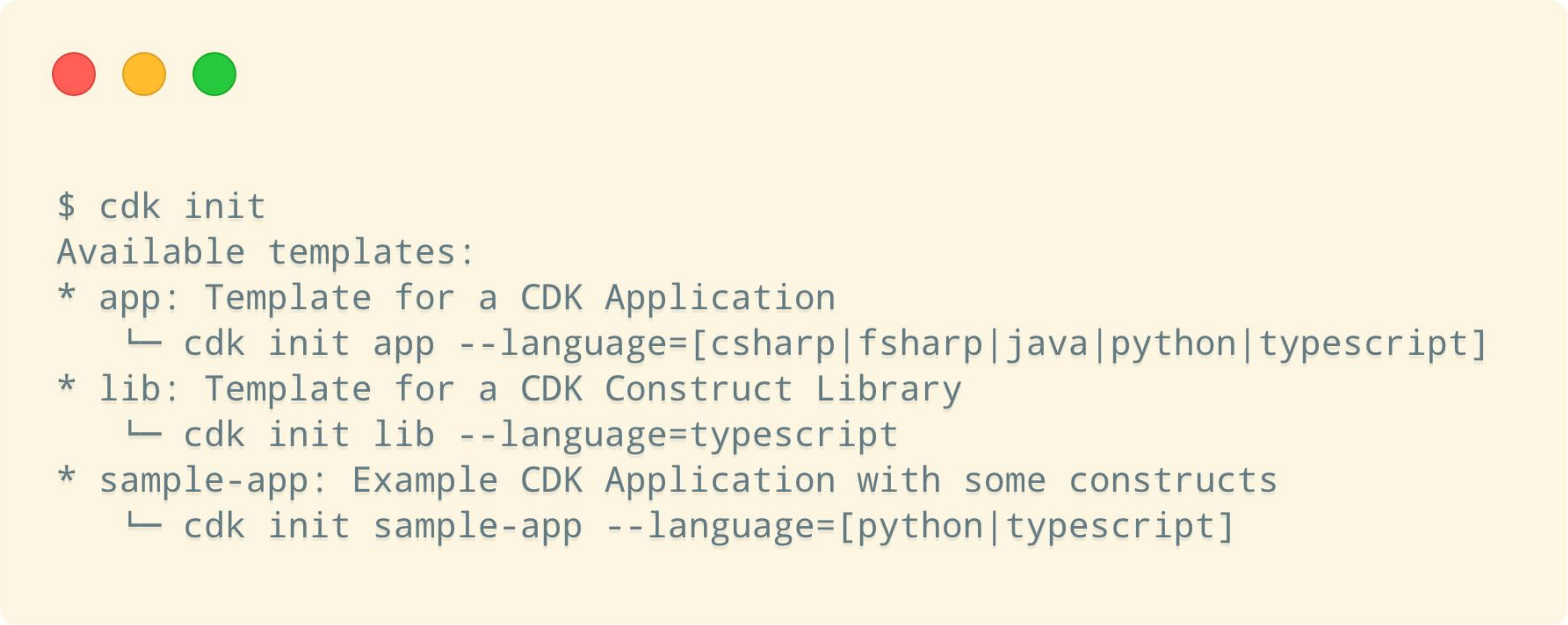

CDK Application
Construct
Construct
"Compiler"
"Assembly Language"
"Processor"
CDK CLI
> cdk synth
CloudFormation Template
AWS CloudFormation

AWS CDK



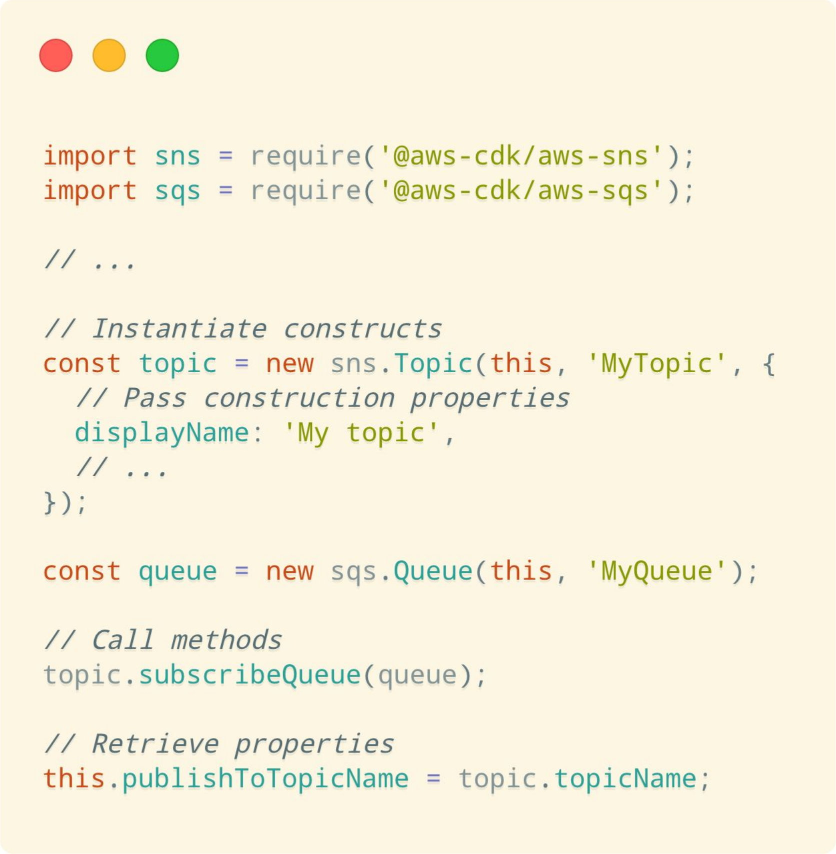
@paulwilljones

AWS CDK - Multi Language Support



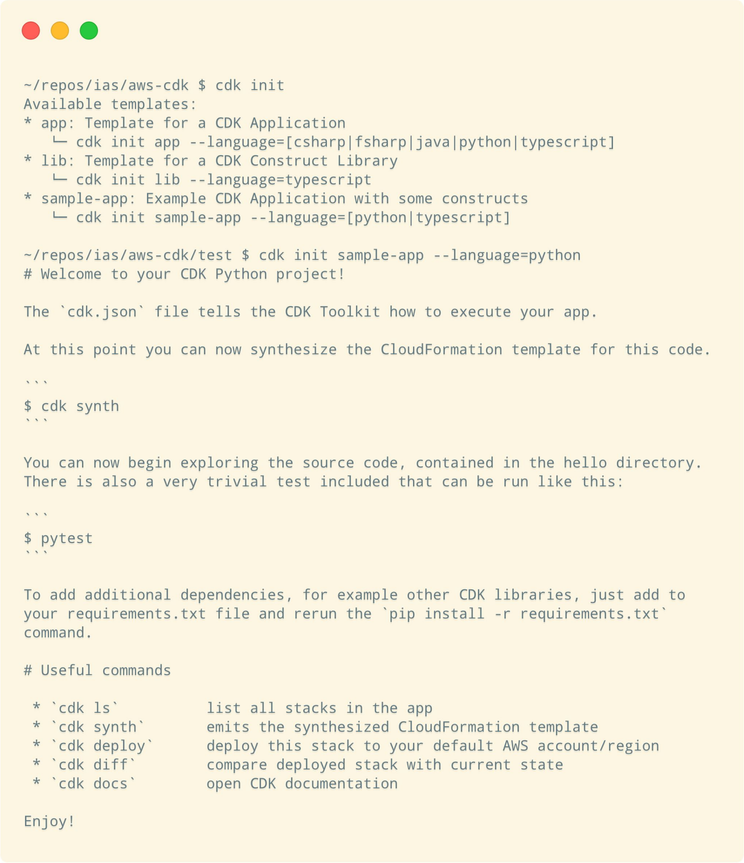
```
$ cdk init
Available templates:
* app: Template for a CDK Application
   └─ cdk init app --language=[csharp|fsharp|java|python|typescript]
* lib: Template for a CDK Construct Library
   └─ cdk init lib --language=typescript
* sample-app: Example CDK Application with some constructs
   └─ cdk init sample-app --language=[python|typescript]
```

AWS CDK



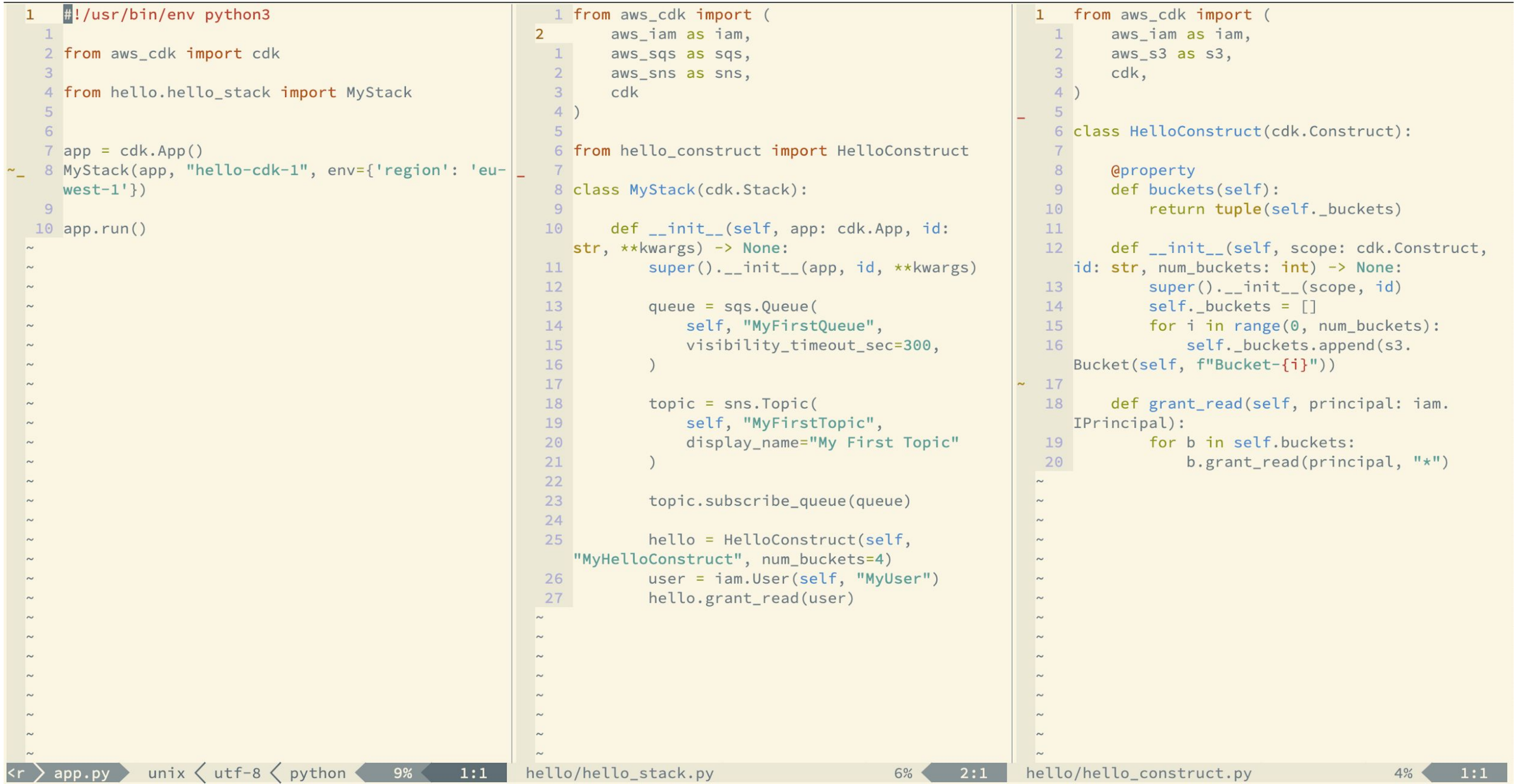
```
import sns = require('@aws-cdk/aws-sns');
import sqs = require('@aws-cdk/aws-sqs');

// ...

// Instantiate constructs
const topic = new sns.Topic(this, 'MyTopic', {
  // Pass construction properties
  displayName: 'My topic',
  // ...
});

const queue = new sqs.Queue(this, 'MyQueue');

// Call methods
topic.subscribeQueue(queue);

// Retrieve properties
this.publishToTopicName = topic.topicName;
```

AWS CDK



```
~/repos/ias/aws-cdk $ cdk init
Available templates:
* app: Template for a CDK Application
   └─ cdk init app --language=[csharp|fsharp|java|python|typescript]
* lib: Template for a CDK Construct Library
   └─ cdk init lib --language=typescript
* sample-app: Example CDK Application with some constructs
   └─ cdk init sample-app --language=[python|typescript]

~/repos/ias/aws-cdk/test $ cdk init sample-app --language=python
# Welcome to your CDK Python project!

The `cdk.json` file tells the CDK Toolkit how to execute your app.

At this point you can now synthesize the CloudFormation template for this code.

```
$ cdk synth
```

You can now begin exploring the source code, contained in the hello directory.
There is also a very trivial test included that can be run like this:

```
$ pytest
```

To add additional dependencies, for example other CDK libraries, just add to
your requirements.txt file and rerun the `pip install -r requirements.txt`
command.

Useful commands

 * `cdk ls`   list all stacks in the app
 * `cdk synth`   emits the synthesized CloudFormation template
 * `cdk deploy`   deploy this stack to your default AWS account/region
 * `cdk diff`   compare deployed stack with current state
 * `cdk docs`   open CDK documentation

Enjoy!
```

AWS CDK



```
#!/usr/bin/env python3

from aws_cdk import cdk

from hello.hello_stack import MyStack


app = cdk.App()
MyStack(app, "hello-cdk-1", env={'region': 'eu-west-1'})

app.run()
```

```
<r > app.py   unix < utf-8 < python   9%   1:1
```

```
from aws_cdk import (
    aws_iam as iam,
    aws_sqs as sqs,
    aws_sns as sns,
    cdk
)

from hello_construct import HelloConstruct

class MyStack(cdk.Stack):

    def __init__(self, app: cdk.App, id: str, **kwargs) -> None:
        super().__init__(app, id, **kwargs)

        queue = sqs.Queue(
            self, "MyFirstQueue",
            visibility_timeout_sec=300,
        )

        topic = sns.Topic(
            self, "MyFirstTopic",
            display_name="My First Topic"
        )

        topic.subscribe_queue(queue)

        hello = HelloConstruct(self, "MyHelloConstruct", num_buckets=4)
        user = iam.User(self, "MyUser")
        hello.grant_read(user)
```

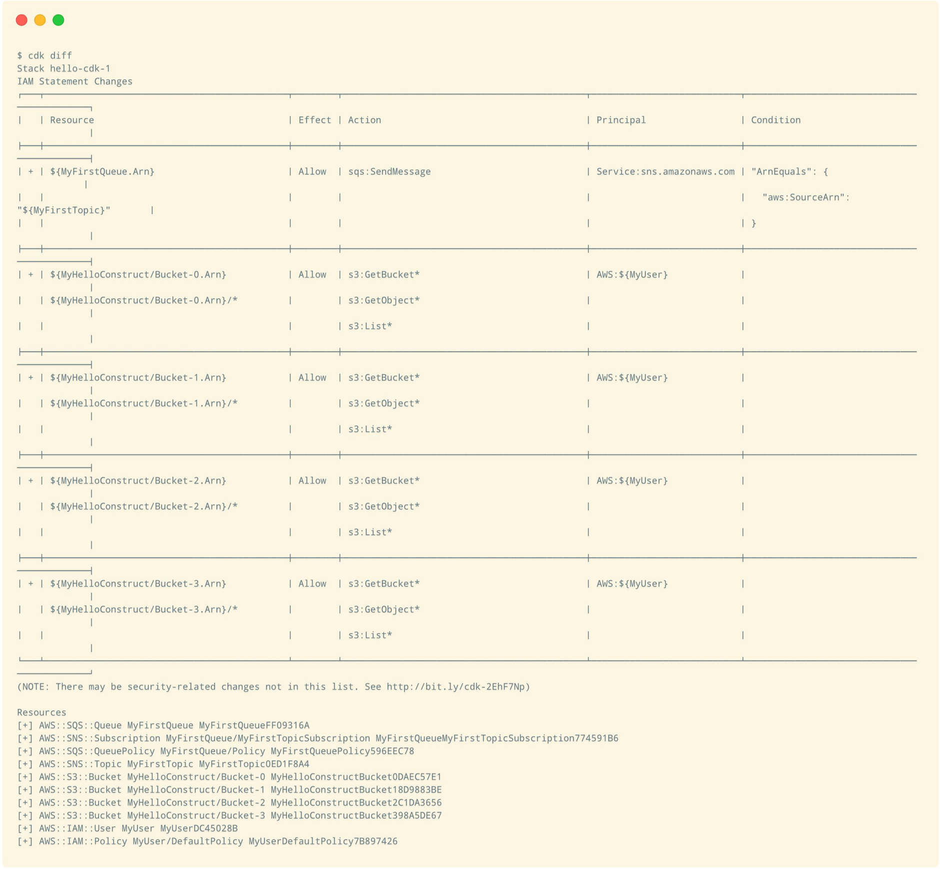
```
hello/hello_stack.py   6%   2:1
```

```
from aws_cdk import (
    aws_iam as iam,
    aws_s3 as s3,
    cdk,
)

class HelloConstruct(cdk.Construct):

    @property
    def buckets(self):
        return tuple(self._buckets)

    def __init__(self, scope: cdk.Construct, id: str, num_buckets: int) -> None:
        super().__init__(scope, id)
        self._buckets = []
        for i in range(0, num_buckets):
            self._buckets.append(s3.Bucket(self, f"Bucket-{i}"))

    def grant_read(self, principal: iam.IPrincipal):
        for b in self.buckets:
            b.grant_read(principal, "*")
```

```
hello/hello_construct.py   4%   1:1
```

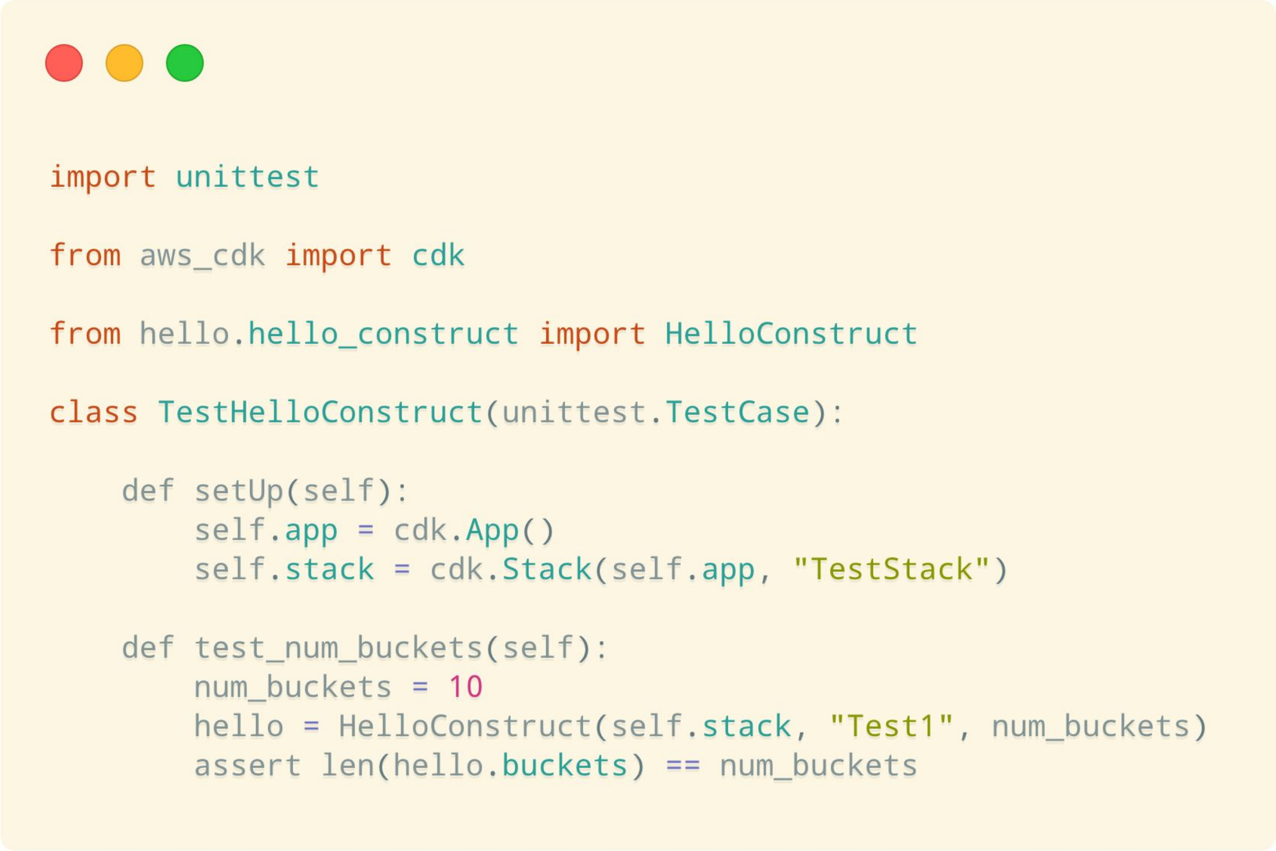
ljones

AWS CDK



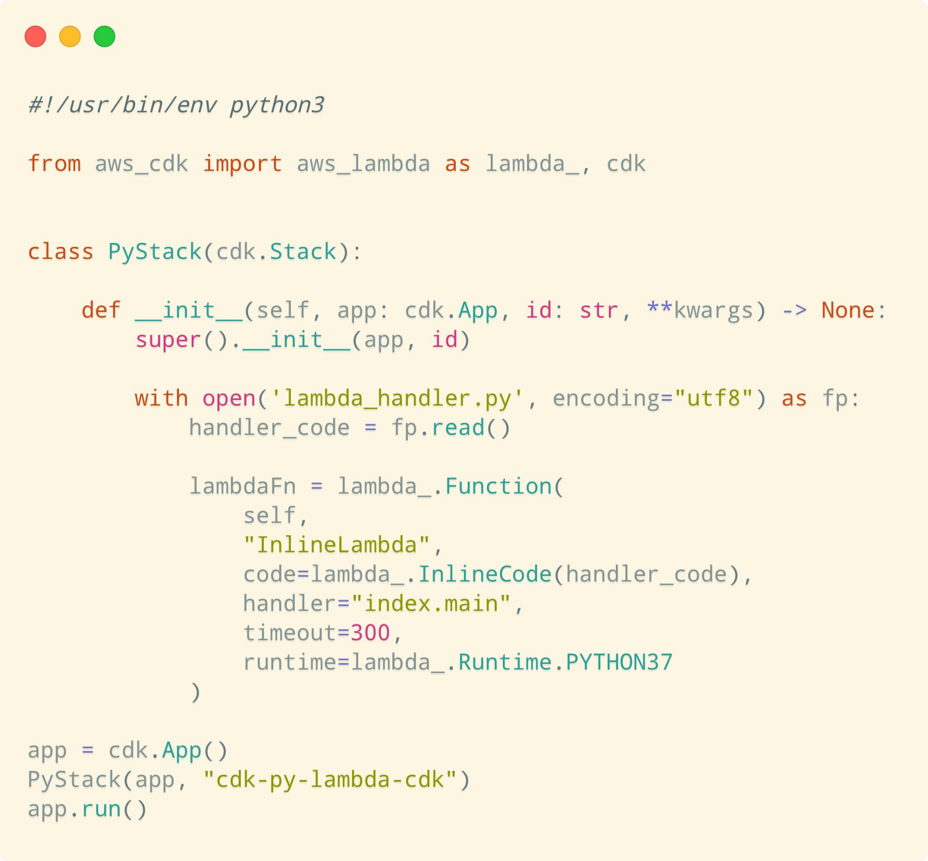
```
$ cdk diff
Stack hello-cdk-1
IAM Statement Changes
┌───┬─────────────────────────────────────────────┬────────┬────────────────────────────────────────────┬───────────────────────────┬────────────────────────────────
─────────────┐
│   │ Resource                                    │ Effect │ Action                                     │ Principal                 │ Condition
             │
├───┼─────────────────────────────────────────────┼────────┼────────────────────────────────────────────┼───────────────────────────┼────────────────────────────────
─────────────┤
│ + │ ${MyFirstQueue.Arn}                         │ Allow  │ sqs:SendMessage                            │ Service:sns.amazonaws.com │ "ArnEquals": {
            │
│   │                                             │        │                                            │                           │   "aws:SourceArn":
"${MyFirstTopic}"       │
│   │                                             │        │                                            │                           │ }
             │
├───┼─────────────────────────────────────────────┼────────┼────────────────────────────────────────────┼───────────────────────────┼────────────────────────────────
─────────────┤
│ + │ ${MyHelloConstruct/Bucket-0.Arn}            │ Allow  │ s3:GetBucket*                              │ AWS:${MyUser}             │
             │
│   │ ${MyHelloConstruct/Bucket-0.Arn}/*          │        │ s3:GetObject*                              │                           │
             │
│   │                                             │        │ s3:List*                                   │                           │
             │
├───┼─────────────────────────────────────────────┼────────┼────────────────────────────────────────────┼───────────────────────────┼────────────────────────────────
─────────────┤
│ + │ ${MyHelloConstruct/Bucket-1.Arn}            │ Allow  │ s3:GetBucket*                              │ AWS:${MyUser}             │
             │
│   │ ${MyHelloConstruct/Bucket-1.Arn}/*          │        │ s3:GetObject*                              │                           │
             │
│   │                                             │        │ s3:List*                                   │                           │
             │
├───┼─────────────────────────────────────────────┼────────┼────────────────────────────────────────────┼───────────────────────────┼────────────────────────────────
─────────────┤
│ + │ ${MyHelloConstruct/Bucket-2.Arn}            │ Allow  │ s3:GetBucket*                              │ AWS:${MyUser}             │
             │
│   │ ${MyHelloConstruct/Bucket-2.Arn}/*          │        │ s3:GetObject*                              │                           │
             │
│   │                                             │        │ s3:List*                                   │                           │
             │
├───┼─────────────────────────────────────────────┼────────┼────────────────────────────────────────────┼───────────────────────────┼────────────────────────────────
─────────────┤
│ + │ ${MyHelloConstruct/Bucket-3.Arn}            │ Allow  │ s3:GetBucket*                              │ AWS:${MyUser}             │
             │
│   │ ${MyHelloConstruct/Bucket-3.Arn}/*          │        │ s3:GetObject*                              │                           │
             │
│   │                                             │        │ s3:List*                                   │                           │
             │
└───┴─────────────────────────────────────────────┴────────┴────────────────────────────────────────────┴───────────────────────────┴────────────────────────────────
─────────────┘
(NOTE: There may be security-related changes not in this list. See http://bit.ly/cdk-2EhF7Np)

Resources
[+] AWS::SQS::Queue MyFirstQueue MyFirstQueueFF09316A
[+] AWS::SNS::Subscription MyFirstQueue/MyFirstTopicSubscription MyFirstQueueMyFirstTopicSubscription774591B6
[+] AWS::SQS::QueuePolicy MyFirstQueue/Policy MyFirstQueuePolicy596EEC78
[+] AWS::SNS::Topic MyFirstTopic MyFirstTopic0ED1F8A4
[+] AWS::S3::Bucket MyHelloConstruct/Bucket-0 MyHelloConstructBucket0DAEC57E1
[+] AWS::S3::Bucket MyHelloConstruct/Bucket-1 MyHelloConstructBucket18D9883BE
[+] AWS::S3::Bucket MyHelloConstruct/Bucket-2 MyHelloConstructBucket2C1DA3656
[+] AWS::S3::Bucket MyHelloConstruct/Bucket-3 MyHelloConstructBucket398A5DE67
[+] AWS::IAM::User MyUser MyUserDC45028B
[+] AWS::IAM::Policy MyUser/DefaultPolicy MyUserDefaultPolicy7B897426
```

AWS CDK Testing



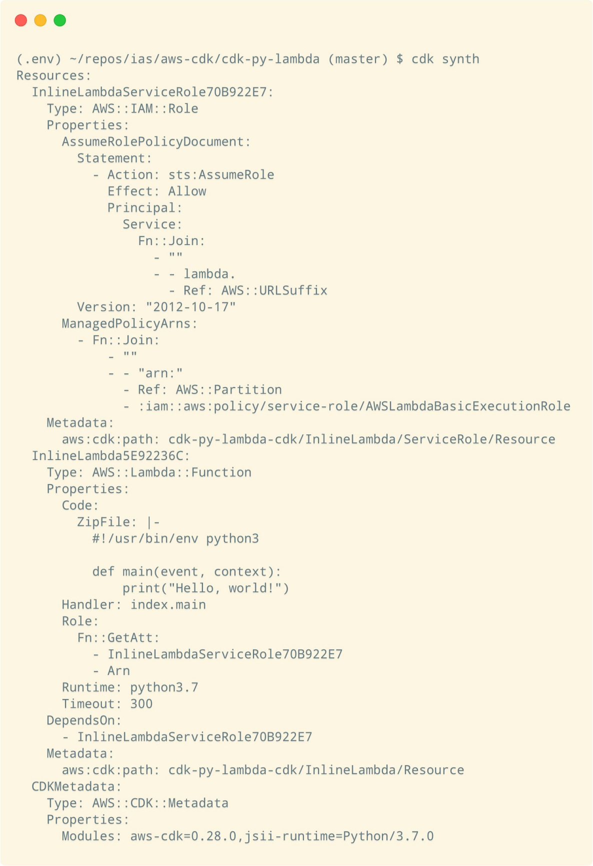
```python
import unittest

from aws_cdk import cdk

from hello.hello_construct import HelloConstruct

class TestHelloConstruct(unittest.TestCase):

    def setUp(self):
        self.app = cdk.App()
        self.stack = cdk.Stack(self.app, "TestStack")

    def test_num_buckets(self):
        num_buckets = 10
        hello = HelloConstruct(self.stack, "Test1", num_buckets)
        assert len(hello.buckets) == num_buckets
```

AWS CDK Serverless



```python
#!/usr/bin/env python3

from aws_cdk import aws_lambda as lambda_, cdk


class PyStack(cdk.Stack):

    def __init__(self, app: cdk.App, id: str, **kwargs) -> None:
        super().__init__(app, id)

        with open('lambda_handler.py', encoding="utf8") as fp:
            handler_code = fp.read()

            lambdaFn = lambda_.Function(
                self,
                "InlineLambda",
                code=lambda_.InlineCode(handler_code),
                handler="index.main",
                timeout=300,
                runtime=lambda_.Runtime.PYTHON37
            )

app = cdk.App()
PyStack(app, "cdk-py-lambda-cdk")
app.run()
```

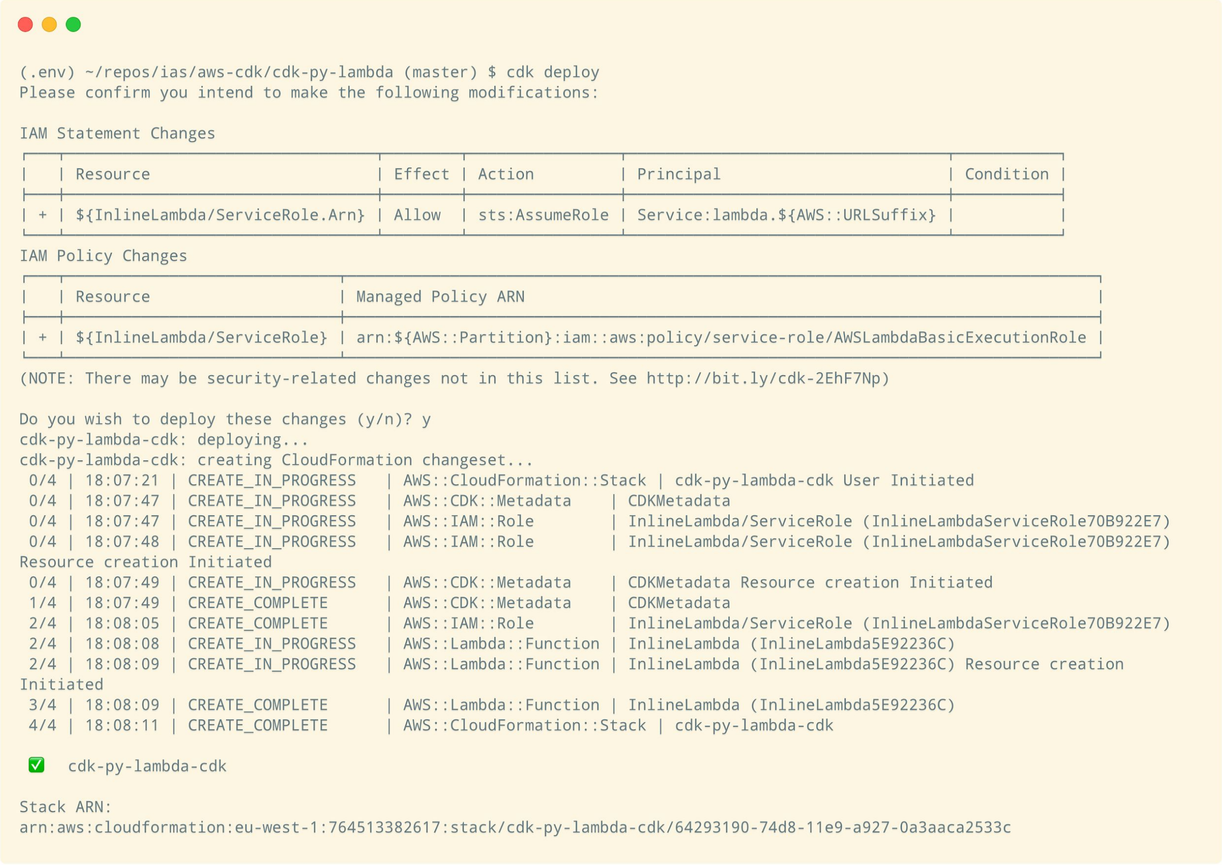
AWS CDK Serverless



```
(.env) ~/repos/ias/aws-cdk/cdk-py-lambda (master) $ cdk synth
Resources:
  InlineLambdaServiceRole70B922E7:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Action: sts:AssumeRole
            Effect: Allow
            Principal:
              Service:
                Fn::Join:
                  - ""
                  - - lambda.
                    - Ref: AWS::URLSuffix
        Version: "2012-10-17"
      ManagedPolicyArns:
        - Fn::Join:
            - ""
            - - "arn:"
              - Ref: AWS::Partition
              - :iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
    Metadata:
      aws:cdk:path: cdk-py-lambda-cdk/InlineLambda/ServiceRole/Resource
  InlineLambda5E92236C:
    Type: AWS::Lambda::Function
    Properties:
      Code:
        ZipFile: |-
          #!/usr/bin/env python3

          def main(event, context):
              print("Hello, world!")
      Handler: index.main
      Role:
        Fn::GetAtt:
          - InlineLambdaServiceRole70B922E7
          - Arn
      Runtime: python3.7
      Timeout: 300
    DependsOn:
      - InlineLambdaServiceRole70B922E7
    Metadata:
      aws:cdk:path: cdk-py-lambda-cdk/InlineLambda/Resource
  CDKMetadata:
    Type: AWS::CDK::Metadata
    Properties:
      Modules: aws-cdk=0.28.0,jsii-runtime=Python/3.7.0
```

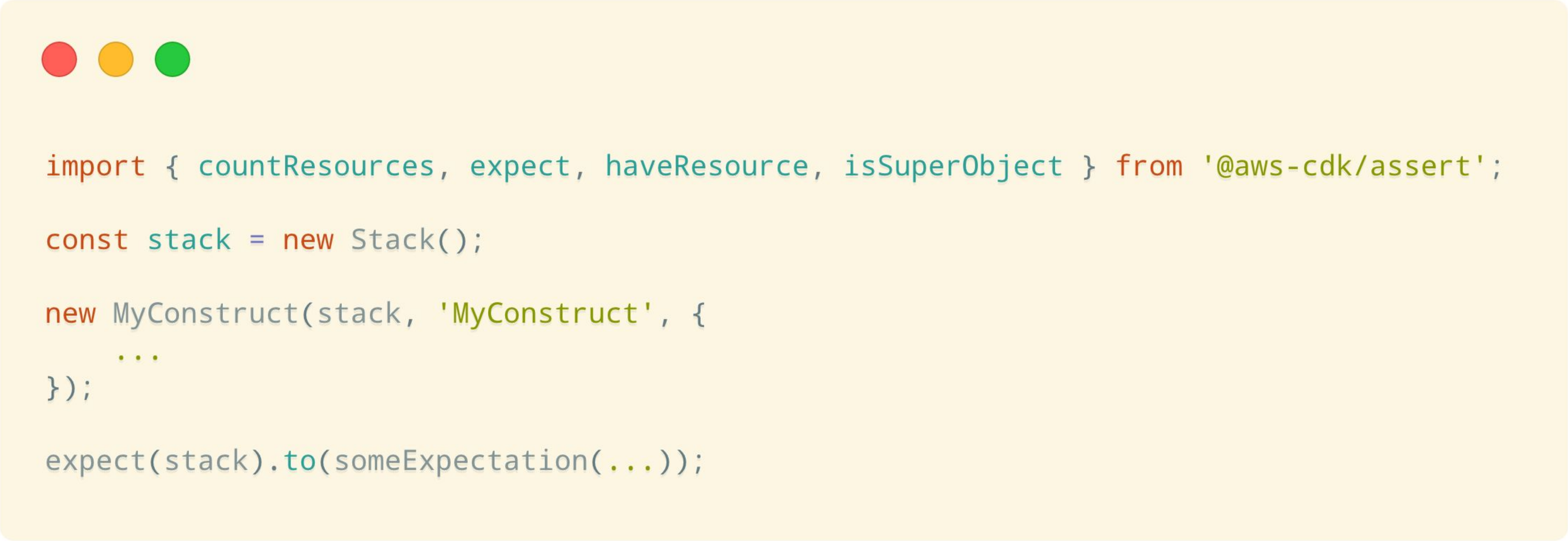
@paulwilljones

AWS CDK Serverless



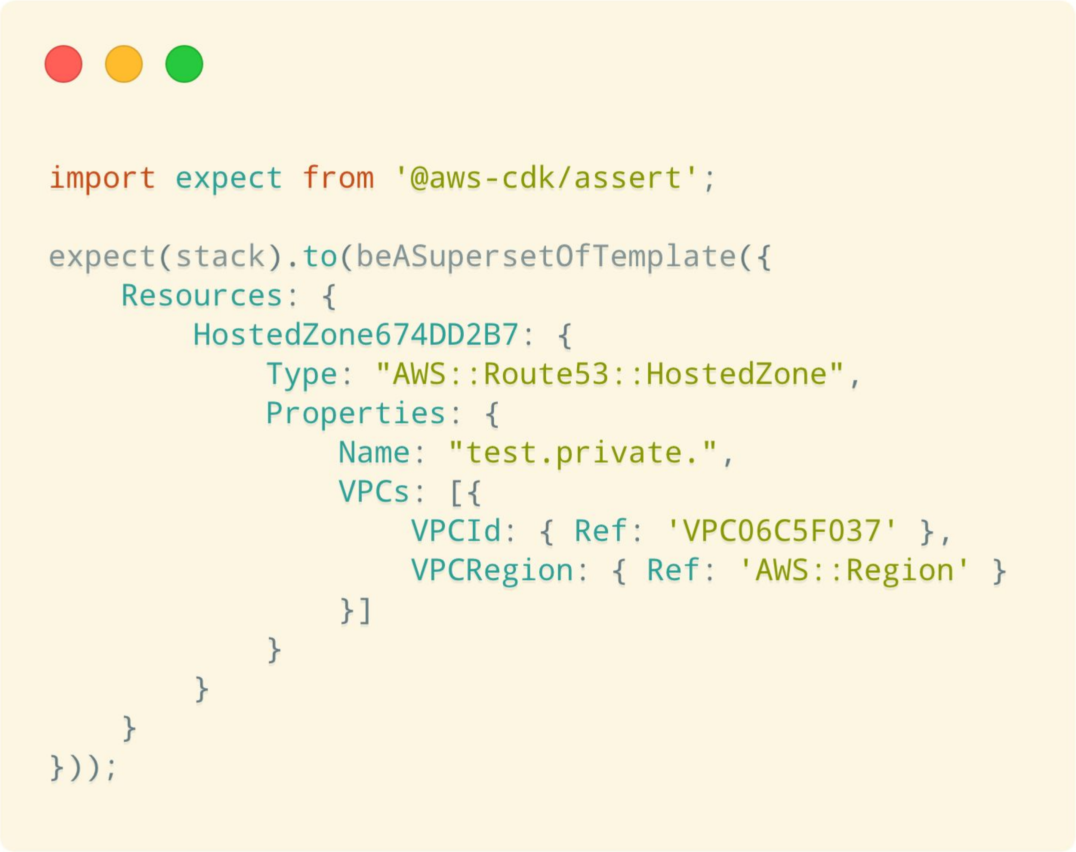
```
(.env) ~/repos/ias/aws-cdk/cdk-py-lambda (master) $ cdk deploy
Please confirm you intend to make the following modifications:

IAM Statement Changes
┌───┬─────────────────────────────────┬────────┬────────────────┬──────────────────────────────────┬───────────┐
│   │ Resource                        │ Effect │ Action         │ Principal                        │ Condition │
├───┼─────────────────────────────────┼────────┼────────────────┼──────────────────────────────────┼───────────┤
│ + │ ${InlineLambda/ServiceRole.Arn} │ Allow  │ sts:AssumeRole │ Service:lambda.${AWS::URLSuffix} │           │
└───┴─────────────────────────────────┴────────┴────────────────┴──────────────────────────────────┴───────────┘
IAM Policy Changes
┌───┬─────────────────────────────┬────────────────────────────────────────────────────────────────────────────────┐
│   │ Resource                    │ Managed Policy ARN                                                             │
├───┼─────────────────────────────┼────────────────────────────────────────────────────────────────────────────────┤
│ + │ ${InlineLambda/ServiceRole} │ arn:${AWS::Partition}:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole │
└───┴─────────────────────────────┴────────────────────────────────────────────────────────────────────────────────┘
(NOTE: There may be security-related changes not in this list. See http://bit.ly/cdk-2EhF7Np)

Do you wish to deploy these changes (y/n)? y
cdk-py-lambda-cdk: deploying...
cdk-py-lambda-cdk: creating CloudFormation changeset...
 0/4 | 18:07:21 | CREATE_IN_PROGRESS   | AWS::CloudFormation::Stack | cdk-py-lambda-cdk User Initiated
 0/4 | 18:07:47 | CREATE_IN_PROGRESS   | AWS::CDK::Metadata     | CDKMetadata
 0/4 | 18:07:47 | CREATE_IN_PROGRESS   | AWS::IAM::Role         | InlineLambda/ServiceRole (InlineLambdaServiceRole70B922E7)
 0/4 | 18:07:48 | CREATE_IN_PROGRESS   | AWS::IAM::Role         | InlineLambda/ServiceRole (InlineLambdaServiceRole70B922E7)
Resource creation Initiated
 0/4 | 18:07:49 | CREATE_IN_PROGRESS   | AWS::CDK::Metadata     | CDKMetadata Resource creation Initiated
 1/4 | 18:07:49 | CREATE_COMPLETE      | AWS::CDK::Metadata     | CDKMetadata
 2/4 | 18:08:05 | CREATE_COMPLETE      | AWS::IAM::Role         | InlineLambda/ServiceRole (InlineLambdaServiceRole70B922E7)
 2/4 | 18:08:08 | CREATE_IN_PROGRESS   | AWS::Lambda::Function | InlineLambda (InlineLambda5E92236C)
 2/4 | 18:08:09 | CREATE_IN_PROGRESS   | AWS::Lambda::Function | InlineLambda (InlineLambda5E92236C) Resource creation
Initiated
 3/4 | 18:08:09 | CREATE_COMPLETE      | AWS::Lambda::Function | InlineLambda (InlineLambda5E92236C)
 4/4 | 18:08:11 | CREATE_COMPLETE      | AWS::CloudFormation::Stack | cdk-py-lambda-cdk

 ✅  cdk-py-lambda-cdk

Stack ARN:
arn:aws:cloudformation:eu-west-1:764513382617:stack/cdk-py-lambda-cdk/64293190-74d8-11e9-a927-0a3aaca2533c
```

AWS CDK Testing



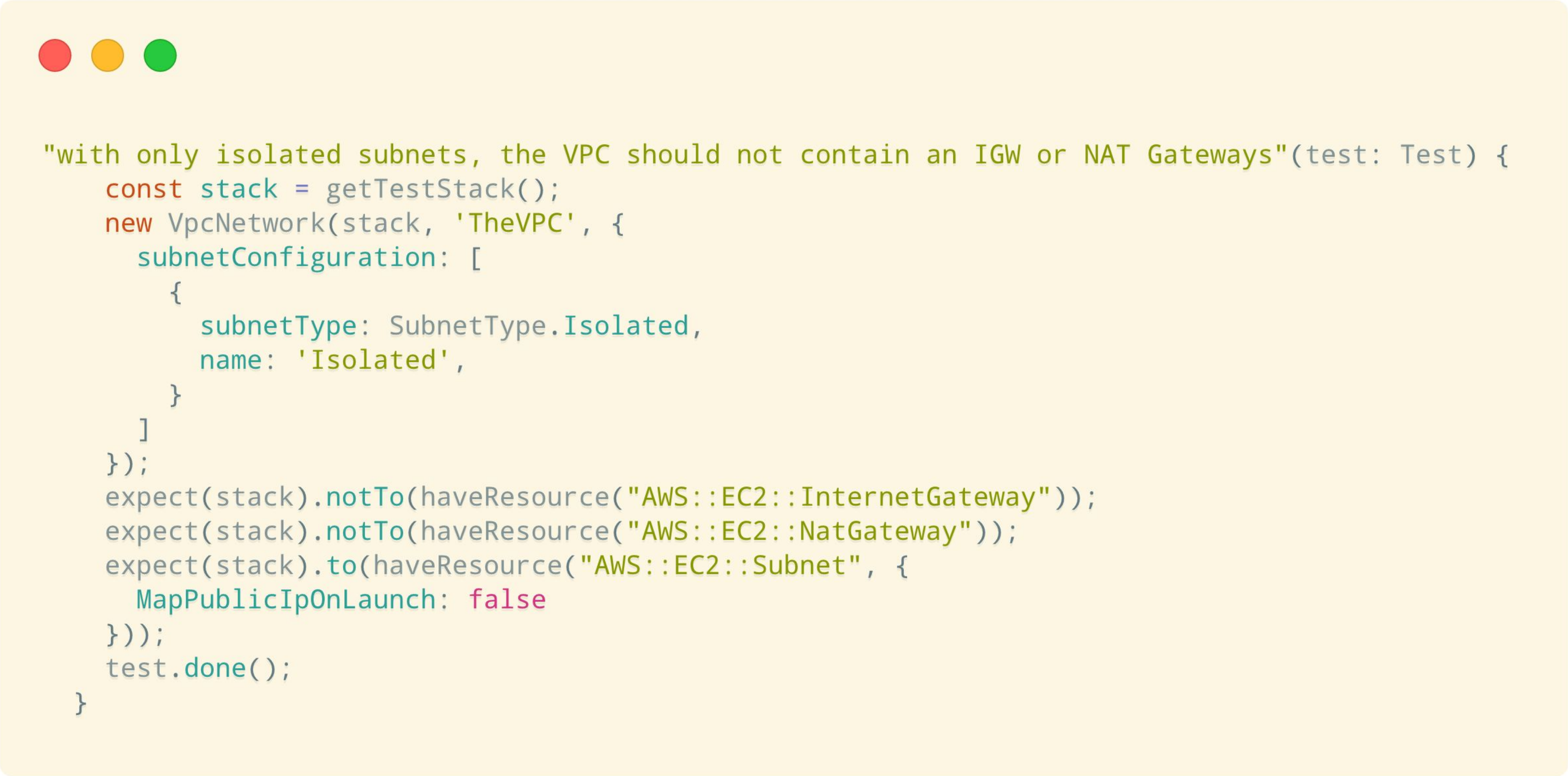
```
import { countResources, expect, haveResource, isSuperObject } from '@aws-cdk/assert';

const stack = new Stack();

new MyConstruct(stack, 'MyConstruct', {
    ...
});

expect(stack).to(someExpectation(...));
```

AWS CDK Testing



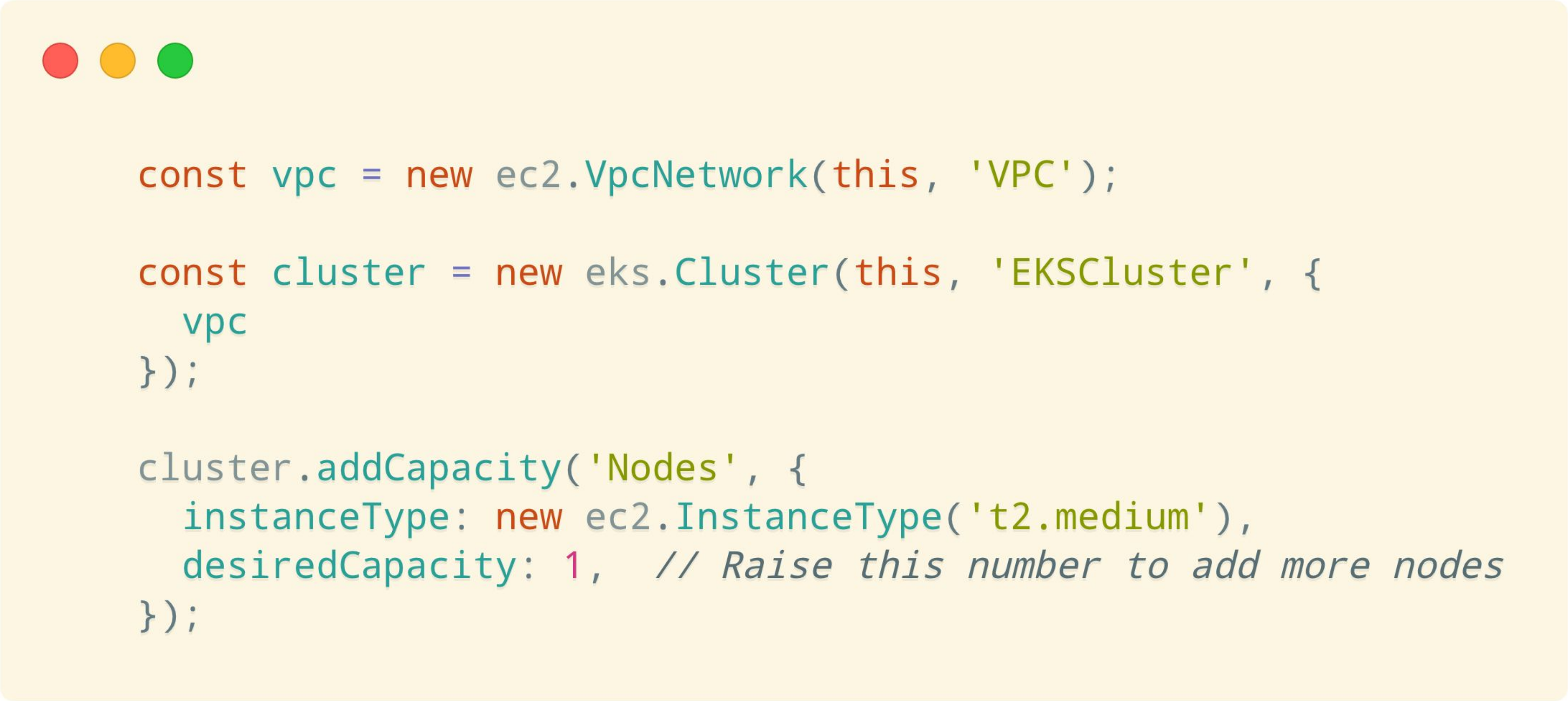
```
import expect from '@aws-cdk/assert';

expect(stack).to(beASupersetOfTemplate({
    Resources: {
        HostedZone674DD2B7: {
            Type: "AWS::Route53::HostedZone",
            Properties: {
                Name: "test.private.",
                VPCs: [{
                    VPCId: { Ref: 'VPC06C5F037' },
                    VPCRegion: { Ref: 'AWS::Region' }
                }]
            }
        }
    }
}));
```

AWS CDK Testing



```
"with only isolated subnets, the VPC should not contain an IGW or NAT Gateways"(test: Test) {
    const stack = getTestStack();
    new VpcNetwork(stack, 'TheVPC', {
      subnetConfiguration: [
        {
          subnetType: SubnetType.Isolated,
          name: 'Isolated',
        }
      ]
    });
    expect(stack).notTo(haveResource("AWS::EC2::InternetGateway"));
    expect(stack).notTo(haveResource("AWS::EC2::NatGateway"));
    expect(stack).to(haveResource("AWS::EC2::Subnet", {
      MapPublicIpOnLaunch: false
    }));
    test.done();
  }
```

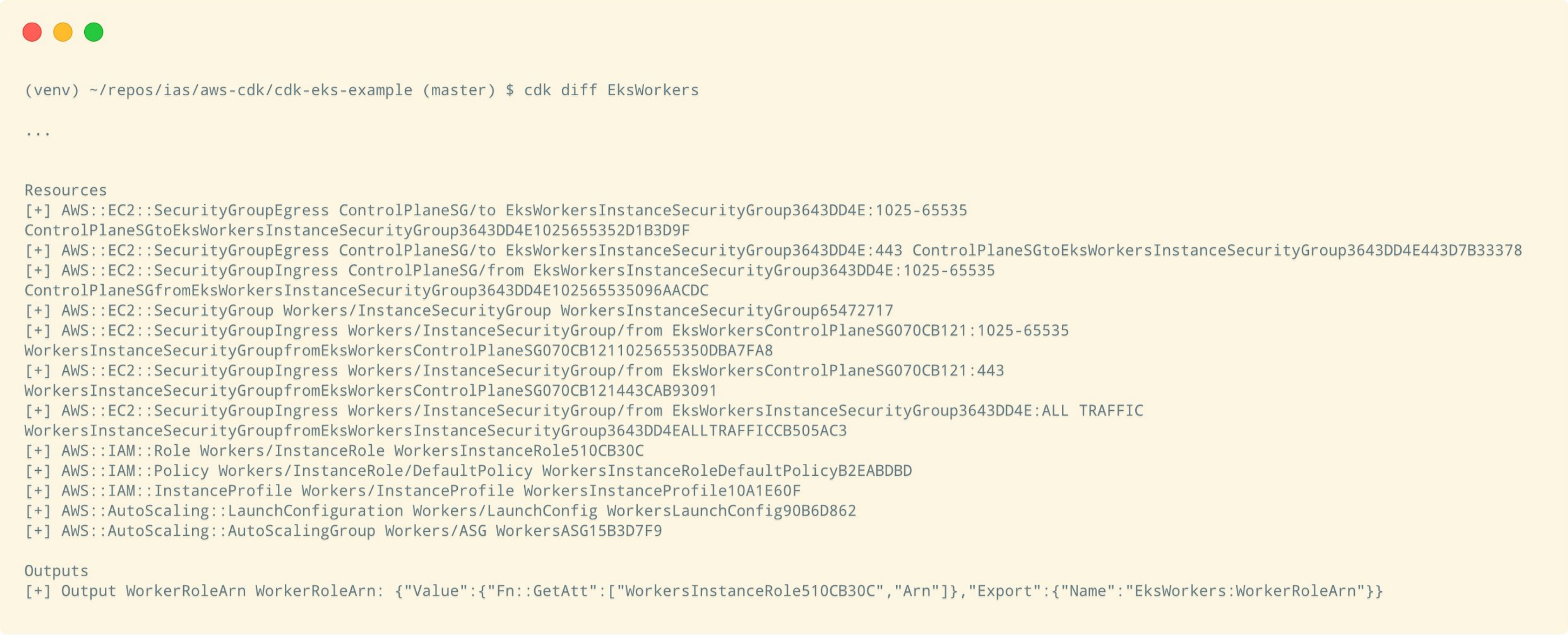
AWS CDK EKS - Control Plane



```
const vpc = new ec2.VpcNetwork(this, 'VPC');

const cluster = new eks.Cluster(this, 'EKSCluster', {
  vpc
});

cluster.addCapacity('Nodes', {
  instanceType: new ec2.InstanceType('t2.medium'),
  desiredCapacity: 1,  // Raise this number to add more nodes
});
```

AWS CDK EKS - Worker Nodes

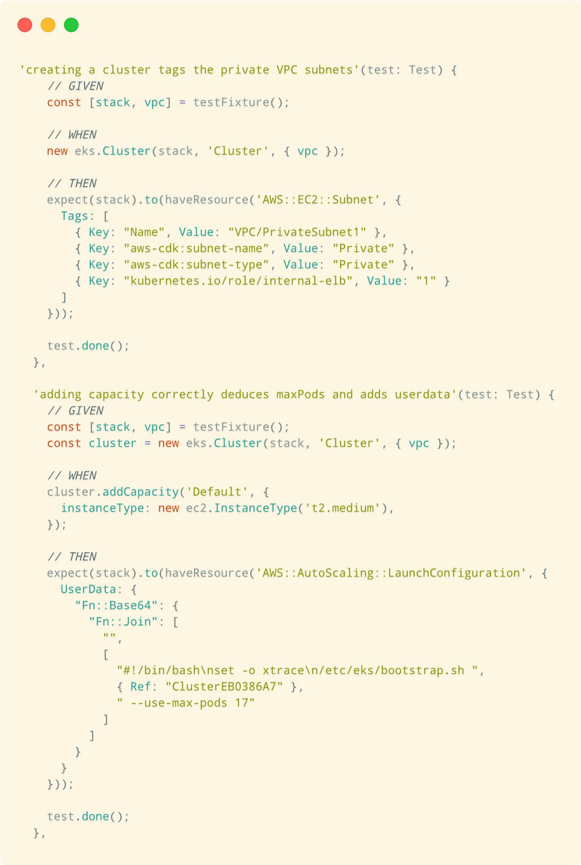
KubeCon CloudNativeCon Europe 2019



```
(venv) ~/repos/ias/aws-cdk/cdk-eks-example (master) $ cdk diff EksWorkers

...


Resources
[+] AWS::EC2::SecurityGroupEgress ControlPlaneSG/to EksWorkersInstanceSecurityGroup3643DD4E:1025-65535
ControlPlaneSGtoEksWorkersInstanceSecurityGroup3643DD4E1025655352D1B3D9F
[+] AWS::EC2::SecurityGroupEgress ControlPlaneSG/to EksWorkersInstanceSecurityGroup3643DD4E:443 ControlPlaneSGtoEksWorkersInstanceSecurityGroup3643DD4E443D7B33378
[+] AWS::EC2::SecurityGroupIngress ControlPlaneSG/from EksWorkersInstanceSecurityGroup3643DD4E:1025-65535
ControlPlaneSGfromEksWorkersInstanceSecurityGroup3643DD4E102565535096AACDC
[+] AWS::EC2::SecurityGroup Workers/InstanceSecurityGroup WorkersInstanceSecurityGroup65472717
[+] AWS::EC2::SecurityGroupIngress Workers/InstanceSecurityGroup/from EksWorkersControlPlaneSG070CB121:1025-65535
WorkersInstanceSecurityGroupfromEksWorkersControlPlaneSG070CB1211025655350DBA7FA8
[+] AWS::EC2::SecurityGroupIngress Workers/InstanceSecurityGroup/from EksWorkersControlPlaneSG070CB121:443
WorkersInstanceSecurityGroupfromEksWorkersControlPlaneSG070CB121443CAB93091
[+] AWS::EC2::SecurityGroupIngress Workers/InstanceSecurityGroup/from EksWorkersInstanceSecurityGroup3643DD4E:ALL TRAFFIC
WorkersInstanceSecurityGroupfromEksWorkersInstanceSecurityGroup3643DD4EALLTRAFFICCB505AC3
[+] AWS::IAM::Role Workers/InstanceRole WorkersInstanceRole510CB30C
[+] AWS::IAM::Policy Workers/InstanceRole/DefaultPolicy WorkersInstanceRoleDefaultPolicyB2EABDBD
[+] AWS::IAM::InstanceProfile Workers/InstanceProfile WorkersInstanceProfile10A1E60F
[+] AWS::AutoScaling::LaunchConfiguration Workers/LaunchConfig WorkersLaunchConfig90B6D862
[+] AWS::AutoScaling::AutoScalingGroup Workers/ASG WorkersASG15B3D7F9

Outputs
[+] Output WorkerRoleArn WorkerRoleArn: {"Value":{"Fn::GetAtt":["WorkersInstanceRole510CB30C","Arn"]},"Export":{"Name":"EksWorkers:WorkerRoleArn"}}
```

@paulwilljones

AWS CDK EKS Testing



```
'creating a cluster tags the private VPC subnets'(test: Test) {
    // GIVEN
    const [stack, vpc] = testFixture();

    // WHEN
    new eks.Cluster(stack, 'Cluster', { vpc });

    // THEN
    expect(stack).to(haveResource('AWS::EC2::Subnet', {
      Tags: [
        { Key: "Name", Value: "VPC/PrivateSubnet1" },
        { Key: "aws-cdk:subnet-name", Value: "Private" },
        { Key: "aws-cdk:subnet-type", Value: "Private" },
        { Key: "kubernetes.io/role/internal-elb", Value: "1" }
      ]
    }));

    test.done();
  },

  'adding capacity correctly deduces maxPods and adds userdata'(test: Test) {
    // GIVEN
    const [stack, vpc] = testFixture();
    const cluster = new eks.Cluster(stack, 'Cluster', { vpc });

    // WHEN
    cluster.addCapacity('Default', {
      instanceType: new ec2.InstanceType('t2.medium'),
    });

    // THEN
    expect(stack).to(haveResource('AWS::AutoScaling::LaunchConfiguration', {
      UserData: {
        "Fn::Base64": {
          "Fn::Join": [
            "",
            [
              "#!/bin/bash\nset -o xtrace\n/etc/eks/bootstrap.sh ",
              { Ref: "ClusterEB0386A7" },
              " --use-max-pods 17"
            ]
          ]
        }
      }
    }));

    test.done();
  },
```

AWS CDK - Recap

Multi language AWS infrastructure composition

Reduce boilerplate through Construct Library

Build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure.

State handled via CloudFormation and Change Sets

AWS CDK - Links

- https://cdkworkshop.com
- https://docs.aws.amazon.com/cdk/api/latest/
- https://github.com/awslabs/aws-cdk
- https://gitter.im/awslabs/aws-cdk

DEV372

Infrastructure Is Code
with the AWS Cloud Development Kit

Elad Ben-Israel
Principal Engineer
AWS Developer Tools

Jason Fulghum
Development Manager
AWS Developer Tools

re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

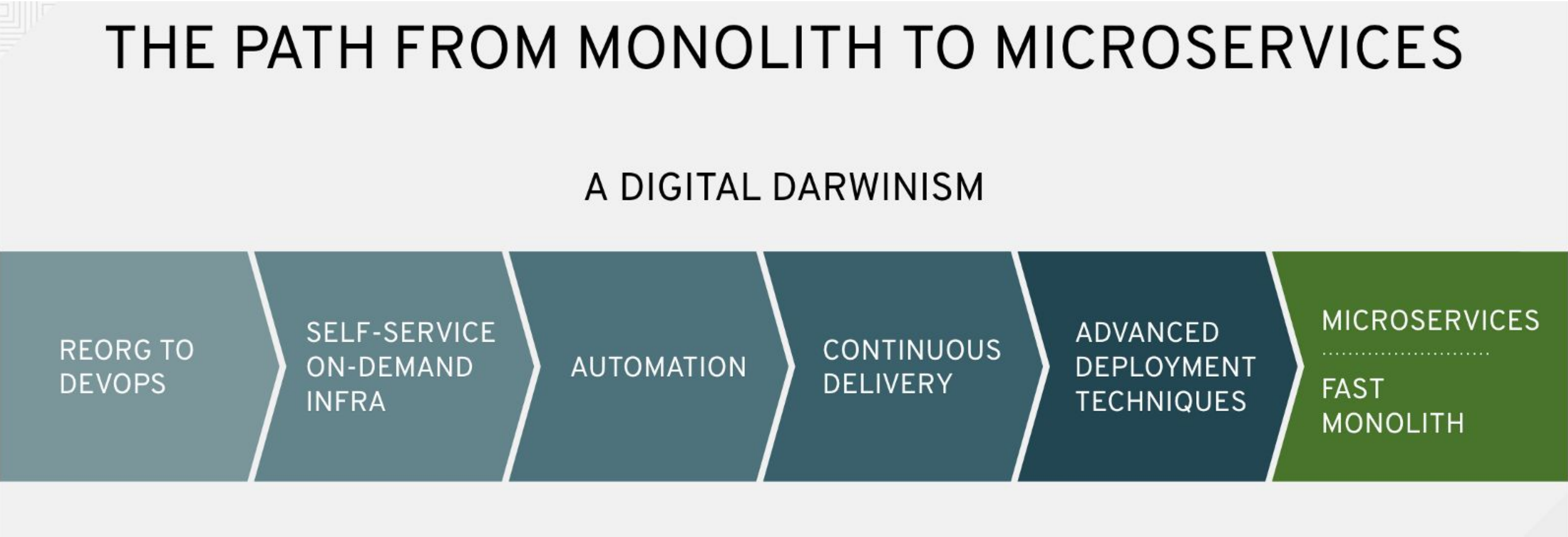
aws

Cultural Readiness

Is your organisation ready to adopt the codification of infrastructure?

Consolidating infrastructure and application concerns may be incompatible with discipline silos

Cultural Readiness



THE PATH FROM MONOLITH TO MICROSERVICES

A DIGITAL DARWINISM

REORG TO DEVOPS

SELF-SERVICE ON-DEMAND INFRA

AUTOMATION

CONTINUOUS DELIVERY

ADVANCED DEPLOYMENT TECHNIQUES

MICROSERVICES

FAST MONOLITH

Wrap Up

Evolve the paradigm of infrastructure composition

Facilitate a coexistence of application and infrastructure code

Develop more testable infrastructure code

Reduce the cognitive overhead of YAML/DSL development

Leverage modern language features by programmatically defining cloud resources

References

IaS
https://medium.com/p/from-yaml-to-typescript-a-developers-view-on-cloud-automation-bba5365439f4
https://www.infoq.com/articles/cloud-native-infrastructure
https://blog.kylegalbraith.com/2018/12/21/how-pulumi-compares-to-terraform-for-infrastructure-as-code/
https://cdn2.hubspot.net/hubfs/4429525/Content/Pulumi-Delivering-CNI-as-Code.pdf
https://cdn2.hubspot.net/hubfs/4429525/Content/AWS-Ebook.pdf

Pulumi
https://www.infoq.com/articles/metaparticle-pulumi-ballerina
https://cdn2.hubspot.net/hubfs/4429525/Content/AWS-Ebook.pdf
https://cdn2.hubspot.net/hubfs/4429525/Content/Pulumi-Delivering-CNI-as-Code.pdf
https://blog.pulumi.com/program-kubernetes-with-11-cloud-native-pulumi-pearls
https://pulumi.io/reference/programming-model.html
https://pulumi.io/reference/stack.html
https://blog.pulumi.com/infrastructure-as-code-from-terraform-to-general-purpose-languages-with-pulumi
https://pulumi.io/reference/vs/cloud_templates.html
https://pulumi.io/reference/vs/terraform.html
http://leebriggs.co.uk/blog/2018/09/20/using-pulumi-for-k8s-config-mgmt.html
https://pulumi.io/reference/how.html
https://pulumi.io/reference/state.html
https://aws.amazon.com/blogs/apn/how-to-easily-deploy-an-amazon-eks-cluster-with-pulumi/
https://blog.pulumi.com/easily-create-and-manage-aws-eks-kubernetes-clusters-with-pulumi
https://blog.pulumi.com/lambdas-as-lambdas-the-magic-of-simple-serverless-functions
https://blog.pulumi.com/easy-serverless-apps-and-infrastructure-real-events-real-code
https://pulumi.io/reference/serializing-functions.html
https://blog.pulumi.com/testing-your-infrastructure-as-code-with-pulumi

References

AWS CDK
https://github.com/awslabs/aws-cdk
https://docs.aws.amazon.com/CDK/latest/userguide/what-is.html
https://dev.to/kayis/the-aws-cloud-development-kit-5c9n
https://medium.com/allermedia-techblog/aws-re-invent-2018-best-of-show-cloud-development-kit-cdk-ad1755561ade
https://aws.amazon.com/blogs/developer/aws-cdk-developer-preview/
https://www.cloudreach.com/blog/deploying-reusable-higher-level-resources-with-aws-cdk/
https://aws.amazon.com/blogs/aws/boost-your-infrastructure-with-cdk/
https://rboyd.dev/b3a9137a-53c9-40a0-a70b-bc3752b75184