Deploying an Application to Your EKS Cluster

Craig Golightly
Senior Software Consultant

@seethatgo   www.seethatgo.com

Overview

- Application repo setup
  - GitHub actions
    - Build Docker image
    - Push to ECR
- Terraform module for application
  - Deployment, ingress, DNS
- Horizontal pod autoscaler (HPA)
- Managing change
  - Application
  - Infrastructure

Amazon Elastic Container Registry (ECR)

Managed Docker container registry

- Docker Hub
- Manage own registry

Push images to registry

EKS pull from registry

Deploy into cluster

Demo

Set up sample application repository

GitHub actions

- Build container from source code
- Push to ECR

Demo

Deploy sample app into EKS cluster

Define infrastructure code

- TF Cloud workspace
- Container registry
- Image and tag

Ingress

- https

Custom DNS

Access app in your cluster

Automatically scale resources up and down

Cluster autoscaler

- Add and remove nodes

Horizontal pod autoscaler (HPA)

- Add and remove pods
- Relies on metrics server

Balance cost and performance

- Variable workloads

Demo

Install Kubernetes metrics server

Add HPA config to application

Run simple load test

- Observe pods and nodes scale

Demo

Framework helps you manage change

Change to application

- Re-deploy to cluster

Change to cluster config

- Node type

Operations Cost

NAT Gateway

- Each Availability Zone (AZ)

Load balancer(s)

EC2 instances

EKS cluster

CloudWatch

KMS

Route53 hosted zones

Delete infrastructure

- Not running actual application
- VPC or region changes

Terraform Destory

- Removes items managed by Terraform

Queue destroy plan in Terraform Cloud

```
kubernetes-ops-staging-sample-app
kubernetes-ops-staging-helm-grafana-loki-stack
kubernetes-ops-staging-helm-kube-prometheus-stack
kubernetes-ops-staging-helm-external-dns
kubernetes-ops-staging-helm-ingress-nginx
kubernetes-ops-staging-helm-cert-manager
kubernetes-ops-staging-25-eks-cluster-autoscaler
kubernetes-ops-staging-5-route53-hostedzone
kubernetes-ops-staging-20-eks
kubernetes-ops-staging-10-vpc
```

Order to Destroy Workspaces

Reverse order of creation

Use Terraform Destroy plans vs. manual deletion to ensure all items are deleted

Demo

Terraform cloud

Queue destroy plans in order

Remove infrastructure from AWS account

Summary

Create and manage EKS using IaC

Terraform modules

- Supporting tools (DNS, certs, monitor)

GitHub Actions

- Automatically deploy changes

Deploy and maintain applications

Update cluster configuration

Delete cluster