# Deploying to Elastic Kubernetes Service (EKS)



Craig Golightly
SENIOR SOFTWARE CONSULTANT

@seethatgo www.seethatgo.com



## Overview



### Open an AWS account

## Deploy and update demo app on EKS

- Step-by-step
- Commands

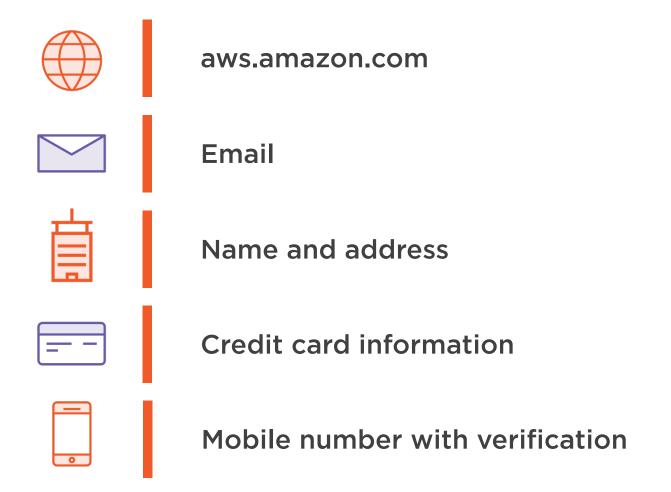
#### **AWS Console**

- Monitoring
- Billing

**Delete cluster** 

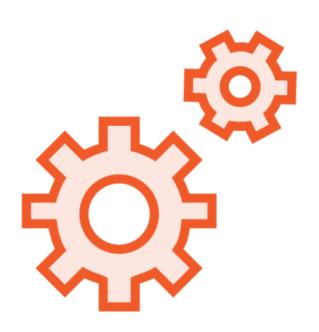


# Opening an Amazon Web Services Account





# Initialize Tooling



## Install and configure AWS CLI

- Python
- Access Key
- Secret

aws-iam-authenticator

#### eksctl

- CloudFormation



```
aws ecr create-repository --repository-name demo

aws ecr get-login --region us-east-1 --no-include-email
```

# Create Repository

**Create repository** 

Login to repository

- Output is a docker login command



```
<registryId>.dkr.ecr..amazonaws.com/<image-name>:<tag>
docker build -t 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:1.0 .

docker tag 8e2036e2586 \
481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:1.0

docker push 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:1.0
```

- <registryId> Id returned from create-repository
- <region> region code for repository
- <image-name> name of the image
- <tag> tag for the image



```
eksctl create cluster
--name
--region
--zones
eksctl create cluster --name demo-cluster --region us-east-1 \
--zones us-east-1a, us-east-1b, us-east-1d
```

## Create Cluster

Cluster name

Region to host cluster

Which availability zones to use for nodes

- UnsupportedAvailabilityZoneException



```
kubectl create deployment demo-app \
--image=481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:1.0
kubectl expose deployment demo-app \
--type=LoadBalancer --port 5000 --target-port 5000
```

# Create Deployment and Service

kubectl to create and expose deployment



```
kubectl scale deployment demo-app --replicas=3
eksctl scale nodegroup --cluster=demo-cluster --nodes=5, --name=ng-e56250ca
```

# Scale Pods and Nodes kubectl to scale pods eksctl to scale nodes



```
docker build -t 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:2.0

docker push 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:2.0

kubectl set image deployment/demo-app \
demo=481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:2.0
```

# Update Application docker build, tag, and push new image kubectl set image





## Serverless Pods with AWS Fargate

- No EC2 nodepool to manage
- Pods run on Fargate resources
- Automatically scale

Create cluster with Fargate nodepool

Deploy and scale demo app



## AWS Web Console



Cluster

Registry

**Monitoring** 

- Cloudwatch

**Billing** 



```
kubectl delete service demo-app
eksctl delete cluster --name demo-cluster
aws ecr list-images --repository-name demo
aws ecr batch-delete-image --repository-name demo --image-ids <image-id>
aws ecr delete-repository --repository-name demo --force
```

# Cleanup

Delete service - kubectl

Delete cluster - eksctl

Delete images – aws ecr



## Summary



#### Use EKS service

## **Entire application lifecycle**

- Create
- Scale
- Update
- Delete

Try it with your app

Check out all 3 cloud options

