Module Checklist

Kubernetes on AWS

By Techworld with Nana

Video Overview

- ★ Introduction to Container Services on AWS
- ★ Create EKS cluster with Node Group - Part 1
- ★ Create EKS cluster with Node Group - Part 2 (Autoscaling)
- ★ Create EKS cluster with Fargate
- ★ Create EKS cluster with eksctl
- ★ Deploy to EKS cluster from Jenkins Pipeline
- ★ Bonus Video - Deploy to LKE cluster from Jenkins Pipeline
- ★ Note on Best Practices - Credentials for different services in Jenkins
- ★ Complete CI/CD Pipeline with DockerHub
- ★ Complete CI/CD Pipeline with AWS ECR

Demo Projects	
Java Maven App	https://gitlab.com/nanuchi/java-maven-app

Check your progress... 1/6

Introduction to Container Services on AWS

- ❏ Watched video

Create EKS cluster with Node Group - Part 1

- ❏ Watched video
- ❏ **Demo executed - Create EKS cluster Part 1:**
  - ❏ Created EKS Role
  - ❏ Created VPC with Cloudformation Template
  - ❏ Created EKS cluster
  - ❏ Connected to EKS cluster with kubectl locally
  - ❏ Created Node Group Role
  - ❏ Created Node Group (EC2 Instances - Worker Nodes)
  - ❏ Configure Auto-Scaling - Deployed cluster-autoscaler fPod
    - Created new Policy for Auto-Scaling Permission
    - Attached new Policy to existing Node Group Role
    - Deployed Auscaler Component in EKS cluster

Useful Links:

- Create EKS with AWS Management Console UI: https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html
- EKS VPC CloudFormation template: https://docs.aws.amazon.com/codebuild/latest/userguide/cloudformation-vpc-template.html
- Create VPC for EKS: https://docs.aws.amazon.com/eks/latest/userguide/create-public-private-vpc.html

Check your progress... 2/6

Create EKS cluster with Node Group - Part 2 (Autoscaling)

- ❏ Watched video
- ❏ **Demo executed - Create EKS cluster Part 2:**
  - ❏ Configure Auto-Scaling - Deployed cluster-autoscaler Pod
    - Created new Policy for Auto-Scaling Permission
    - Attached new Policy to existing Node Group Role
    - Deployed Auscaler Component in EKS cluster
  - ❏ Deployed Example application
    - Deployed nginx Pod
    - Deployed nginx Service
    - Started 20 Pods - see autoscaling in action

Useful Links:

- Cluster Auto-Scaler User Guide: https://docs.aws.amazon.com/eks/latest/userguide/cluster-autoscaler.html
- Gitlab Project Repo: https://gitlab.com/nanuchi/bootcamp-kubernetes/-/tree/master/eks-cluster-autoscaler
- Autoscaling Yaml file: https://raw.githubusercontent.com/kubernetes/autoscaler/master/cluster-autoscaler/cloudprovider/aws/examples/cluster-autoscaler-autodiscover.yaml

Create EKS cluster with Fargate

- ❏ **Watched video**
- ❏ **Demo executed - EKS with Fargate:**
  - ❏ Created Role for Fargate
  - ❏ Created Fargate Profile
  - ❏ Deployed Pod through Fargate

Check your progress... 3/6

Create EKS cluster with eksctl

- ❏ Watched videos
- ❏ **Demo executed - Create EKS cluster with eksctl:**
  - ❏ Installed eksctl
  - ❏ Configured AWS credentials to connect eksctl with your AWS account
  - ❏ Created EKS cluster

Useful Links:

- Installation Guides for eksctl: https://github.com/weaveworks/eksctl#installation

Check your progress... 4/6

EKS & Jenkins

Deploy to EKS cluster from Jenkins Pipeline

- ❏ Watched video
- ❏ **Demo executed - Create Pipeline that deploys to EKS cluster:**
  - ❏ Installed kubectl inside Jenkins Container
  - ❏ Installed aws-iam-authenticator inside Jenkins Container
  - ❏ Created ./kube/config and copied inside the Jenkins Container
  - ❏ Created Jenkins Credential
  - ❏ Created simple Jenkinsfile that deploys to EKS cluster

Useful Links:

- User guide - Cluster authentication: https://docs.aws.amazon.com/eks/latest/userguide/managing-auth.html
- Install aws-iam-authenticator: https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html
- Create Kubeconfig file: https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html
- Jenkinsfile Project Repo: https://gitlab.com/nanuchi/java-maven-app/-/tree/deploy-on-k8s

Bonus Video - Deploy to LKE cluster from Jenkins Pipeline

- ❏ **Watched video**
- ❏ **BONUS: Demo executed - Create Pipeline that deploys to LKE cluster:**
  - ❏ Created LKE cluster
  - ❏ Installed Kubernetes CLI Plugin on Jenkins
  - ❏ Created Jenkins Credential with kubeconfig file
  - ❏ Created simple Jenkinsfile that deploys to LKE cluster

Useful Links:

- Jenkinsfile Project Repo: https://gitlab.com/nanuchi/java-maven-app/-/tree/deploy-to-lke

Check your progress... 5/6

Note on Best Practices - Credentials for different services in Jenkins

- ❏ Watched video

Complete CI/CD Pipeline with DockerHub

- ❏ Watched video
- ❏ **Demo executed - Complete CI/CD Pipeline with DockerHub:**
  - ❏ Created Deployment and Service for App deployment
  - ❏ Adjust Jenkinsfile to set environment variables with *envsubst*
  - ❏ Installed “gettext-base” tool inside Jenkins Container on DigitalOcean Server to have envsubst available
  - ❏ Created Secret for DockerHub Registry in EKS cluster (connect to EKS cluster if not already) and added reference to Deployment file
  - ❏ Executed Jenkins Pipeline

Useful Links:

- Jenkinsfile Project Repo: https://gitlab.com/nanuchi/java-maven-app/-/tree/feature/k8s
- Envsubst: https://www.gnu.org/software/gettext/manual/html_node/envsubst-Invocation.html

Check your progress... 6/6

Complete CI/CD Pipeline with AWS ECR

- ❏ Watched video
- ❏ **Demo executed - Complete CI/CD Pipeline with AWS ECR:**
  - ❏ Created ECR Repository
  - ❏ Created Credential for ECR repository in Jenkins
  - ❏ Created Secret for AWS ECR Registry in EKS cluster and adjusted reference in Deployment file
  - ❏ Updated Jenkinsfile
  - ❏ Executed Jenkins Pipeline

Useful Links:

- Jenkinsfile Project Repo: https://gitlab.com/nanuchi/java-maven-app/-/tree/complete-pipeline-ecr-eks

More Resources...

Best practices

- Create VPC with private and public subnet
- **Security**: Use AWS Key Management Service (KMS) keys to provide envelope encryption of Kubernetes secrets (https://aws.amazon.com/about-aws/whats-new/2020/03/amazon-eks-adds-envelope-encryption-for-secrets-with-aws-kms/)
- Create Jenkins User for Services Jenkins needs access to e.g. Jenkins System User on AWS

Official AWS Best practices for EKS:

- https://aws.github.io/aws-eks-best-practices/ - Each topic and recommendation is based on best practices implemented in production by AWS customers and validated by Kubernetes specialists and the Kubernetes engineering team at AWS.