

Seneca

Assignment2: Create K8s cluster, deploy containerized stateless applications using K8s manifests, expose the applications as NodePort services, roll out an updated version of the application

Submission Instructions	To be submitted via Blackboard. Refer to Blackboard for submission instructions
Value	15% of final grade
Due Date	July 09, 2023

Learning Outcomes Covered in Assignment
<p>Evaluate the applicability of containerization approach and viability of publicly/privately hosted containers orchestration platform for the business needs of the organization.</p> <p>Design, implement and deploy containerized applications to address cost optimization, high availability, and scalability requirements of business applications</p> <p>Evaluate and recommend networking, persistent storage, and IAM (Identity and Access Management) solutions to achieve the desired level of infrastructure and applications security.</p>

Assignment Outline

The objective of this Assignment is to host our first containerized application in the locally simulated single-node K8s cluster. You can use kind tool to create the single node K8s cluster as we did in the class .

We will continue working with the application used in Assignment 1:

https://github.com/ladunuthala/clo835_summer2023_assignment1

At this point you already know how to create docker images for this application and you have published the images in Amazon ECR.

In this assignment, you will host the containerized application on kind cluster running on Amazon Linux-based EC2 instance in AWS environment.

The Assignment flow includes the steps below:

1. Deploy Amazon Linux based EC2 with sufficient capacity to run kind cluster and host our containerized application
2. Install all the pre-requisites on the Amazon EC2 needed to host the containerized application on K8s cluster created by kind (kind, kubectl).
3. Create K8s cluster using kind tool.
4. Deploy containerized application using pod, replicaset , deployment and service manifests.
5. Expose web application using Service of type Nodeport
6. Expose MySQL using Service of type ClusterIP
7. Update the applications and deploy the new version of the application

You will be using the services and tools below:

- Amazon ECR to securely store your container images
- Cloud 9 IDE or your local environment to develop your application and build container images
- Amazon EC2 to host your K8s cluster
- Kind to deploy local K8s cluster
- Kubectl to communicate with K8s API Server
- AWS EC2 to host your containerized application
- AWS IAM to grant EC2 instance access to Amazon ECR repo
- Terraform to deploy the infrastructure (optional)

Evaluation

Please see the evaluation breakdown on the last page.

Submission

Your submission should include the following:

GitHub repo link with all the relevant K8s manifests for MySQL and web applications:

- Pod
- ReplicaSet
- Deployment
- Service

Recording that captures application's deployment and update as per the script below defined in the "Recording" section. No need to include K8s cluster deployment in the video. The recording should not exceed 15 minutes.

Report that specifies the challenges that you faced while implementing the assignment and the ways you resolved them.

All the answers to the questions posed in the assignment should be answered in the report.

Important Notes – please read carefully:

- **Make sure there are no credentials pushed to GitHub repo at any stage!**
- All the commits should have dates before the Assignment 2 due date
- There should be a sequence of commits in your GitHub repo that reflects the progression of your assignment. Submissions with a small number of commits will raise authenticity questions.

- Add meaningful messages to your commits that reflect the added functionality or the fixes you made.
- Make sure your recording is 15 minutes long.
- **Submitting the report with your authentic recount of the challenges faced during the assignment is mandatory. Assignments without the report or with the inauthentic report will be considered not satisfactory.**
- Reference sources used in implementing the assignment, such as specific blog posts, videos or any other sources.
- You cannot use the work/ideas of other students in your individual submission

Submission Requirements Description

GitHub link with K8s manifests

Task	Submission Requirements Description
Pod manifests	Pod manifests for MySQL and web applications. Make sure the manifests specify the port the application is listening on.
ReplicaSet manifests	ReplicaSet manifests for MySQL and web applications.
Deployment manifests	Deployment manifests for MySQL and web applications.
Service manifests	Service manifests for MySQL and web applications.

--	--

Recording

Submission Requirements Description
<p>The recording should clearly demonstrate the points below:</p> <ol style="list-style-type: none">1. The local K8s cluster is running on your Amazon EC2 instance. Demonstrate that this is a single node cluster and that all the basic K8s components are running successfully.<ol style="list-style-type: none">a. What is the IP of the K8s API server in your cluster? (Answer in the report)2. Deploy MySQL and web applications as pods in their respective namespaces.<ol style="list-style-type: none">a. Can both applications listen on the same port inside the container? Explain your answer. (Answer in the report)b. Connect to the server running in the application pod and get a valid response.c. Examine the logs of the invoked application to demonstrate the response from the server was reflected in the log file3. Deploy ReplicaSets of the web application with 3 replicas using ReplicaSet manifest. Use the “app:employees” label to create ReplicaSets for web application. Is the pod created in step 2 governed by the ReplicaSet you created. Explain. Use the “app:mysql” label to create ReplicaSets for MySQL application.4. Create deployments of MySQL and web applications using deployment manifests.

- a. Use the labels from step 3 as selectors in the deployment manifest.
 - b. Is the replicaset created in step 3 part of this deployment?
Explain. (Answer in the report)
5. Expose web application on NodePort **30000** using service manifest. Demonstrate that you can reach the application from your Amazon EC2 instance using `curl` and from the browser.
 6. Update the image version in the deployment manifest and deploy a new version of web application. Demonstrate that the new version is running in the cluster using `kubectl`.
 7. Explain the reason we are using different service types for the web and MySQL applications.

Plagiarism:

Plagiarized assignments will receive a mark of zero on the assignment and a failing grade on the course. You may also receive a permanent note of plagiarism on your academic record.

Integrity Pledge:

By submitting my Assignment, I affirm that I will not give or receive any unauthorized help in this submission and that all work provided will be my own. I agree to abide by Seneca's Academic Integrity Policy, and I understand that any violation of academic integrity will be subject to the penalties outlined in the policy. Click on this link to learn more about Seneca's Academic Integrity Policy: [Academic Integrity Policy](#)

Assignment Grade Breakdown

Task	Points
<p>1. Deployment of local single node cluster</p> <p>a. What is the IP of the K8s API server in your cluster? (Answer in the report)</p>	5
<p>2. Deployment of MySQL and web application pods.</p> <p>b. Can both applications listen on the same port inside the container? Explain your answer. (Answer in the report)</p> <p>c. Connect to the server running web application pod and get a valid response.</p> <p>d. Examine the logs of the invoked application to demonstrate the response from the server was reflected in the log file</p>	20
<p>3. Deploy ReplicaSets of the applications with 3 replicas using ReplicaSet manifest. Use the “app:employees” and “app:mysql” labels respectively to create ReplicaSets for MySQL and web applications. Is the pod created in step 2 governed by the ReplicaSet you created. Explain.</p>	15
<p>4. Create deployments of the MySQL and web applications using deployment manifests.</p> <p>e. Use the labels from step 3 as selectors in the deployment manifest.</p> <p>f. Is the replicaset created in step 3 part of this deployment? Explain. (Answer in the report)</p>	20

5. Expose web application on NodePort 30000 using service manifest. Demonstrate that you can reach the application from your Amazon EC2 instance using <code>curl</code> and from the browser.	20
6. Update the image version in the deployment manifest and deploy a new version of the web application. Demonstrate that the new version is running in the cluster.	15
a. Explain the reason we are using different service types for the web and MySQL applications (Answer in the report)	5
Total	100

Appendix

Recommended implementation flow

Deployment of Infrastructure with Terraform

- One Amazon Ec2 instance in the public subnet of default VPC
- Amazon ECR repositories for images

Important Notes:

Terraform code should not be included in your submission of this Assignment

Deploy web application and MySQL container images to a local K8s cluster

1. Create and Amazon EC2, install all the required tools to work with K8s cluster
2. Create K8s cluster, make sure all the components are healthy
3. Create pod manifests, deploy applications, connect to the ports running in the containers using "kubectl port-forward"
4. Deploy replicaset using K8s manifest
5. Deploy deployments using K8s manifest
6. Deploy service for MySQL and web applications
7. Update the application's image and deploy an updated version of the application using the deployment manifest.

References:

<https://catalog.us-east-1.prod.workshops.aws/workshops/8c9036a7-7564-434c-b558-3588754e21f5/en-US/>

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html>