Troubleshooting a Broken Kubernetes Application

⌛00:52:33 Exit Lab ✓ Complete Lab

⏱ 1 hour duration Practitioner Rate this lab

VIDEOS GUIDE

Troubleshooting a Broken Kubernetes Application

Introduction

Kubernetes administrators need to be able to fix issues with applications running in a cluster. This lab will allow you to test your skills when it comes to fixing broken Kubernetes applications. You will be presented with a cluster running a broken application and asked to identify and correct the problem.

Solution

Log in to the server by using the credentials provided: `ssh cloud_user@<PUBLIC_IP_ADDRESS>` .

Identify What is Wrong with the Application

1. Examine the `web-consumer` deployment, which resides in the `web` namespace, and its Pods by using `kubectl get deployment -n web web-consumer` .
2. Get more information by using `kubectl describe deployment -n web web-consumer` , including container specifications and labels that the Pods are using.
3. Look more closely at the Pods by using `kubectl get pods -n web` to see if both Pods are up and running.
4. Get more information about the Pods by using `kubectl describe pod -n web <POD_NAME>` , and evaluate any warning messages that may come up.

5. Look at the logs associated with the container `busybox` by using `kubectl logs -n web <POD_NAME> -c busybox`.

6. Determine what may be going wrong by reading the output from the container logs.

7. Take a closer look at the pod itself by using `kubectl get pod -n web <POD_NAME> -o yaml` to get the data in the yaml format.

8. Determine which command is causing the errors (in this case, the `while true; do curl auth-db; sleep 5; done` command).

Fix the Problem

1. Take a closer look at the service by using `kubectl get svc -n web auth-db`.

2. Locate where the service is by using `kubectl get namespaces` and finding the one other non-default namespace called `data`.

3. Check `kubectl get svc -n data` and find the `auth-db` service in this namespace, rather than the `web` namespace.

4. Start resolving the issue by using `kubectl edit deployment -n web web-consumer`.

5. In the `spec` section, scroll down to find the pod template and locate the `while true; do curl auth-db; sleep 5; done` command.

6. Change the command to `while true; do curl auth-db.data.svc.cluster.local; sleep 5; done` to give the fully qualified domain name of that service. This will allow the `web-consumer` deployment's Pods to communicate with the service successfully.

7. Save the file and exit by pressing the ESC key and using `:wq`.

8. Check `kubectl get pods -n web` to ensure that the old pods have terminated and the new pods are running successfully.

9. Check the log of one of the new pods by using `kubectl logs -n web <POD-NAME> -c busybox`. This time the pod should be able to communicate successfully with the service.

Conclusion

Tools

Lab Diagram Instant Terminal

Credentials

How do I connect?

Cloud Server Control Plane Node

Username

cloud_user

Password

_*rop3P8

Control Plane Node Private IP

10.0.1.101

Control Plane Node Public IP

52.207.183.89

Launch Instant Terminal

How do I connect?

Additional Resources

Your company, BeeBox, is building some applications for Kubernetes. Your developers have recently deployed an application to your cluster, but it is having some issues.

A set of Pods managed by the `web-consumer` deployment regularly make requests to a service that provides authentication data. Your developers are reporting that the containers are not behaving as expected.

Your task is to look at the application in question, determine the problem, and fix it.

Learning Objectives

0 of 2 completed

- ☐ **Identify What is Wrong with the Application**
- ☐ **Fix the Problem**