

Software Architecture

Assignment 2 - Technology Demonstration



Report Due: Thursday, 10/12/2023 11:59PM
Demonstrations: Week of 10/16/2023

I. Goals

Demonstration of the ability to setup a prototype implementation of a relevant technology stack including:

1. Multiple virtual machines
2. Container orchestration
3. Software containers

II. Specifics

You will be building a prototype Kubernetes cluster to run a simple layered architecture.

1. Your cluster will be implemented on two Linux virtual machines. The choice of virtualization tools is up to the student but if you're new to this, VirtualBox is probably the simplest to implement. The specific Linux distribution will also be up to the student, but I have found that both Ubuntu and Fedora are straightforward.
2. Implement Kubernetes on the two virtual machines as a single cluster. Either full Kubernetes (K8S) or the K3S distribution may be used. MiniKube is NOT an alternative as it will only work on a single node.
3. Implement the Kubernetes Dashboard in your cluster
4. Implement a simple two container (application and database) application as at least two pods within your cluster. I will provide a simple Python/MySQL application you can use if you wish (you can do something else if you want).

III. Deliverables

There are two deliverables from this assignment:

1. A written report describing:
 - a. Your process for approaching the problem. E.g. I got the virtual machines working including....
 - b. Issues/challenges encountered in the process
 - c. Your evaluation, based on this experience, of the technology
2. A demonstration (via zoom) of your prototype to the instructor. Each student will sign up for a 15-minute block (via a spreadsheet that will be available in slack) to show their prototype as well as answer questions. Only the student and instructor will be online. You will need to be able to show and explain:
 - a. The state of your cluster using the Kubernetes Dashboard.
 - b. The operation of your sample application

IV. Notes

Here are some links and observations

- a. Get your application containers running outside of the cluster first. Docker (or Podman) compose is a good way to get this part going first.

- b. If you use Podman as opposed to Docker for containerization it has the capability to generate Kubernetes pod definition files from running containers, this can save you some time.
- c. I used Fedora Linux for my virtual machines. This worked well on the Arm based Mac I have. On an Intel machine Ubuntu works just as well
- d. To reduce the number of really odd errors I recommend on the Linux VM's to:
 - a. Disable the firewall (firewalld or ufw)
 - b. Disable SELinux (more of a RedHat thing than Ubuntu)
<https://www.redhat.com/sysadmin/container-permission-denied-errors>
 - c. Install avahi. This will allow using names for the servers rather than ip addresses
<https://bbs.archlinux.org/viewtopic.php?id=244754>
 - d.
- e. Some links that may be helpful
 - a. <https://buildvirtual.net/deploy-a-kubernetes-cluster-using-ansible/>
 - b. <https://stackoverflow.com/questions/65681045/adding-insecure-registry-in-containerd>
 - c. <https://medium.com/@raj10x/configure-local-kubectl-to-access-remote-kubernetes-cluster-ee78feff2d6d>