**Lesson 9 Demo 5**

**Understanding Container Logs**

**Objective:** Understanding cluster logging architecture

**Tools required:** Kubernetes platform with master, worker1, and worker2 nodes

**Prerequisites:** Kubernetes cluster must be set up with kubeadm, kubectl, and kubelet installed and tested. Some Pods, Containers, Services, etc. should be in place so that troubleshooting can be done on the existing cluster and its contents.

# Steps to be followed:

1. Logging a Docker Container
2. Logging a Kubernetes Container

**Step 1: Logging a Docker Container**

1. Fetching the container ID for all the Docker images using the below command:

**docker ps -a**

Graphical user interface, website

Description automatically generated

1. Listing the specified container’s logs using the following command:

**docker logs -f <Container\_ID>**

A screenshot of a computer

Description automatically generated

**Step 2: Logging a Kubernetes Container**

1. Execute the command to list the Containers of the Pods that are deployed in the default namespace:

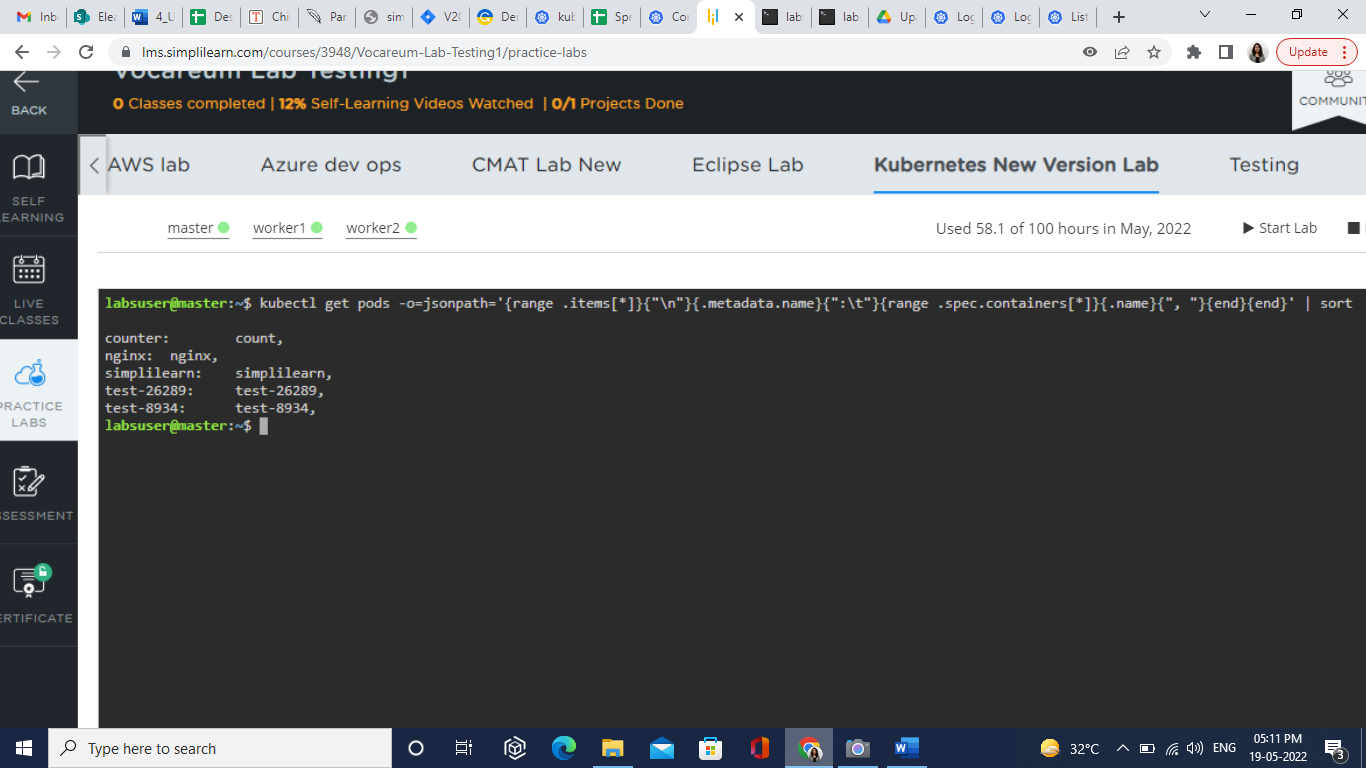
**kubectl get pods -o=jsonpath='{range .items[\*]}{"\n"}{.metadata.name}{":\t"}{range .spec.containers[\*]}{.name}{", "}{end}{end}'**

A screenshot of a computer

Description automatically generated

1. Execute the command to list the Containers of the Pods that are deployed in the default namespace in sorted order:

**kubectl get pods -o=jsonpath='{range .items[\*]}{"\n"}{.metadata.name}{":\t"}{range .spec.containers[\*]}{.name}{", "}{end}{end}' | sort**



1. Execute the command to list the Containers of the Pods that are deployed in all the namespaces in sorted order:

**kubectl get pods --all-namespaces -o=jsonpath='{range .items[\*]}{"\n"}{.metadata.name}{":\t"}{range .spec.containers[\*]}{.name}{", "}{end}{end}' | sort**

