**SWE 645**

**Component-based Software Development**

**ASSIGNMENT-2**

**Team Members**

**Saranya Chakravarthy Korrapati – G01471987**

**Akhil Nakka – G01411969**

**Samuel Archit Sowpati – G01448337**

* First, create three EC2 instances in the AWS Academy Leaner Lab with the configurations given in the instructions. One EC2 instance is for hosting the Kubernetes cluster, one is for hosting Rancher and the third one is for hosting Jenkins.
* Then we connected the EC2 instance and installed docker in it by using the following commands
  + sudo apt-get update
  + sudo apt install docker.io

**Setting up the GitHub Repository:**

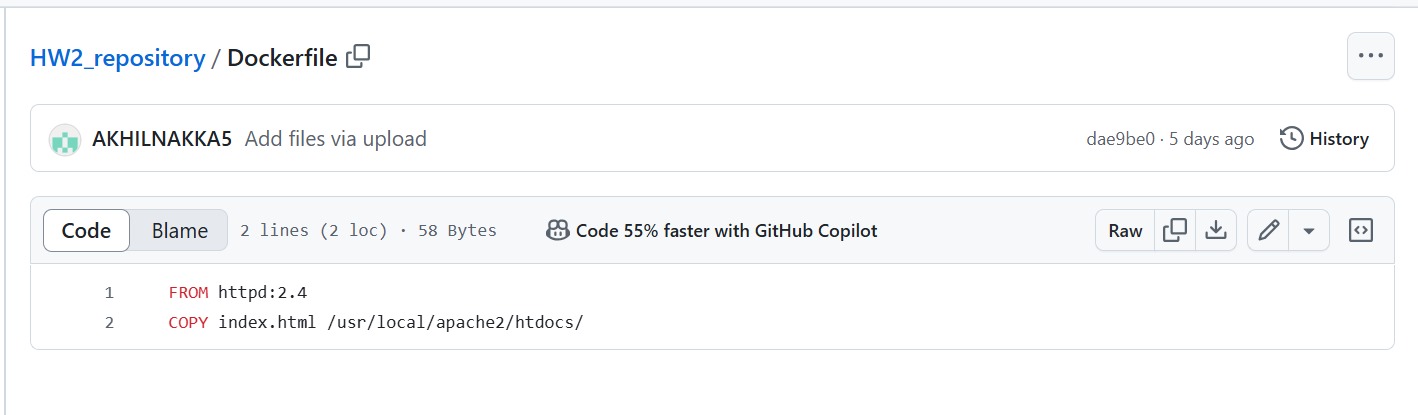
* We created the HW2\_repository in GitHub.
* Push index.html to this repository.

A screenshot of a computer

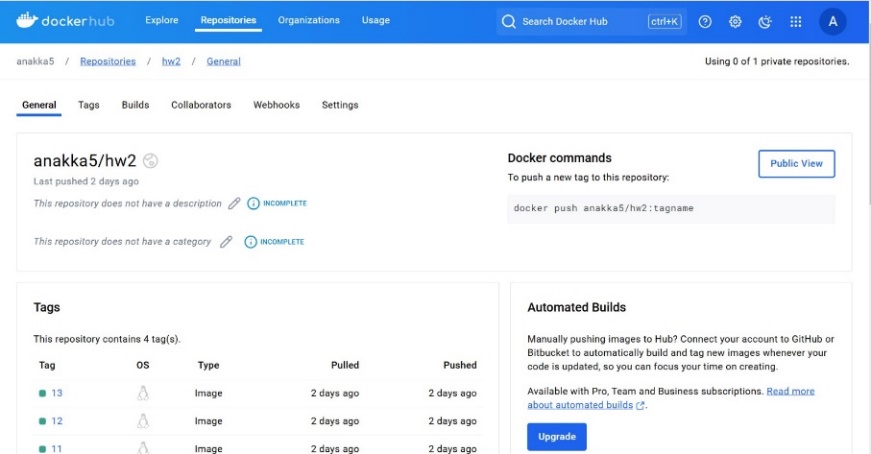
Description automatically generated

**Creating a Docker Image and pushing it to DockerHub:**

* We installed docker as shown above.
* Verify docker installation by using
  + $ sudo docker -v
* To grand user permissions to perform docker commands by running the command
  + $ sudo usermod -aG docker ubuntu
* Log out and log back in for the changes to take place.
* Create an account on <https://hub.docker.com/>
* We now create the dockerfile which docker uses to build the image.

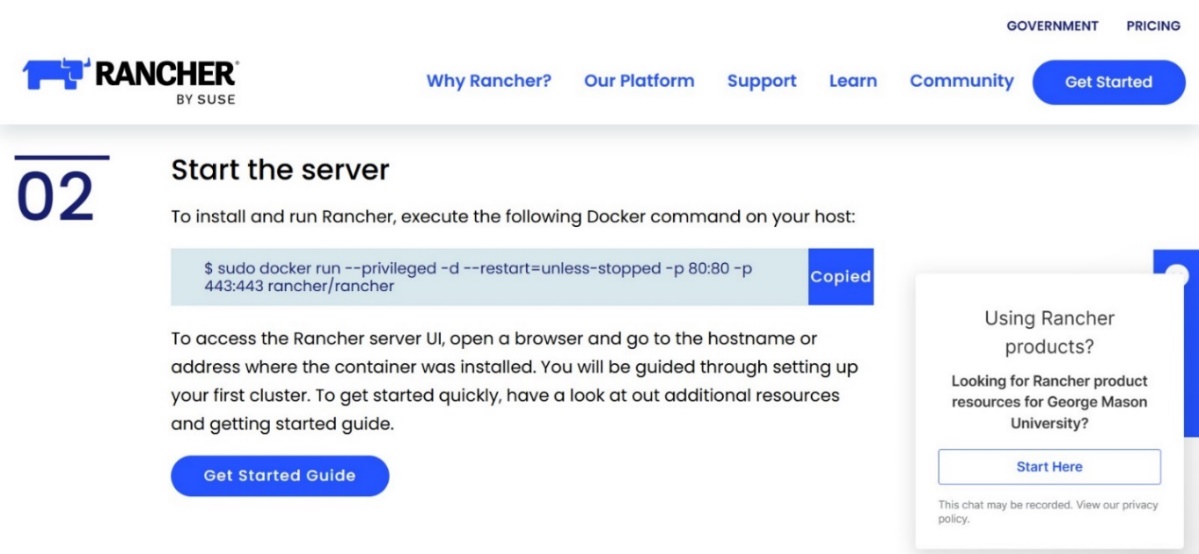


* From httpd:2.4: Using the httpd:2.4(version) base image, Docker starts building the image.
* We then copy the index.html file into the following path: /usr/local/apache2/htdocs/index.html
* This actions transfers our index.html file from the local file system to a specified directory(i.e., /usr/local/apache2/htdocs/ directory) inside the Docker image.Push the dockerfile to the HW2\_repository in Github.
* On the command line, use this command:
  + docker build --tag anakka5/hw2:first .
* To verify the images created, use the following command
  + docker images
* Login to Docker by using the following command
  + docker login
* It will ask us for the credentials, we will enter the username and password.
* Now we will push this image to the docker hub
  + docker push anakka5/hw2



**Setting up Rancher:**

* Go to Rancher.io, Click on Get Started and click on Deploy Rancher.
* In the Start Server, there will be a URL, copy it to the first instance.



* To get the container ID, use the command
  + $ sudo docker ps
* Use the public IPv4 address od the EC2 instance to open the rancher
* We will retrieve the bootstrap password to login into Rancher
  + $ sudo docker logs container-id 2>&1 | grep “Bootstrap password:”
* Login to the rancher using that password.

A screenshot of a login screen

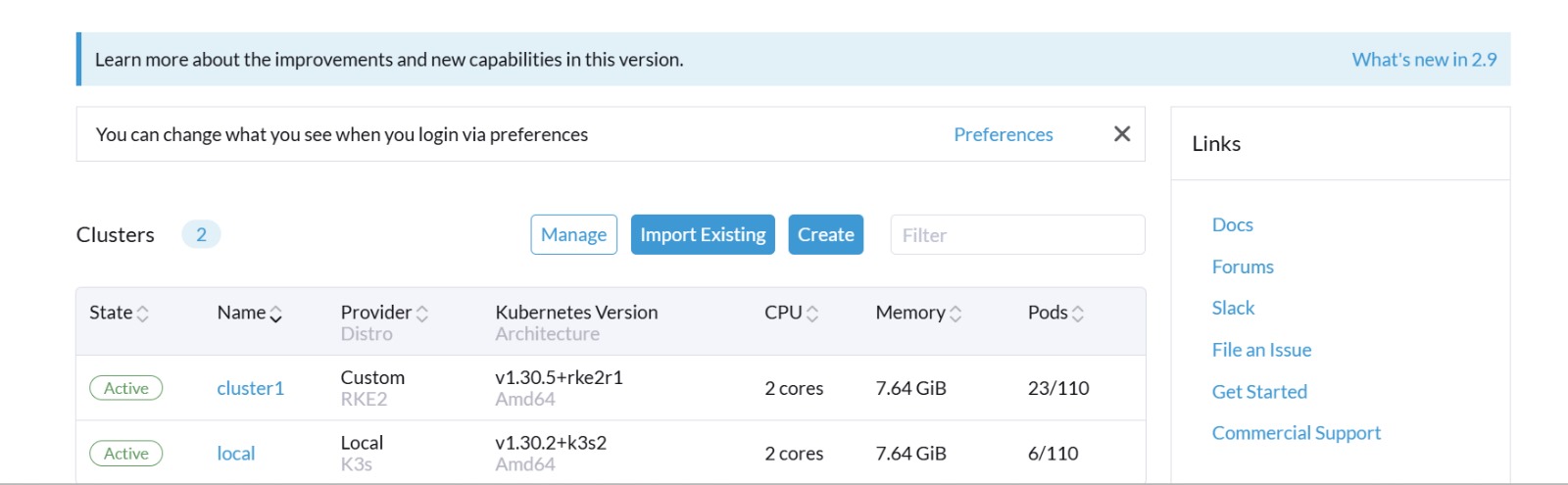
Description automatically generated

* Now, create the cluster using the Create button
* Select Custom, and enter the cluster name (Cluster1)

A screenshot of a computer

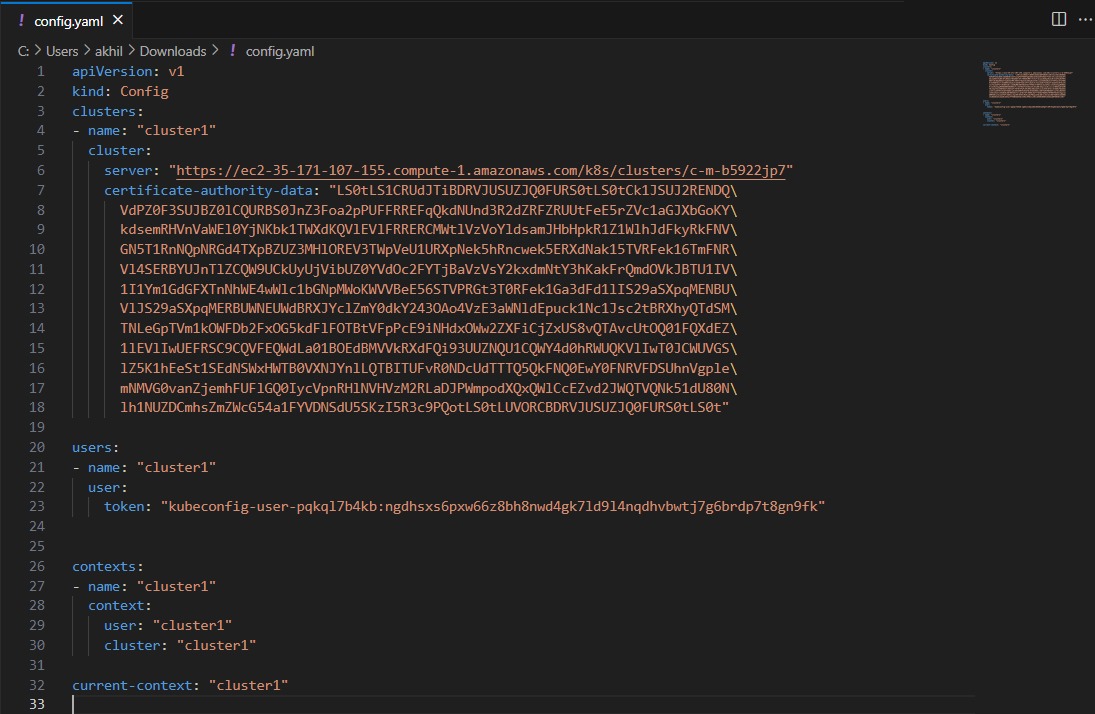
Description automatically generated

* Run the given command in the second instance and the instance will be active.

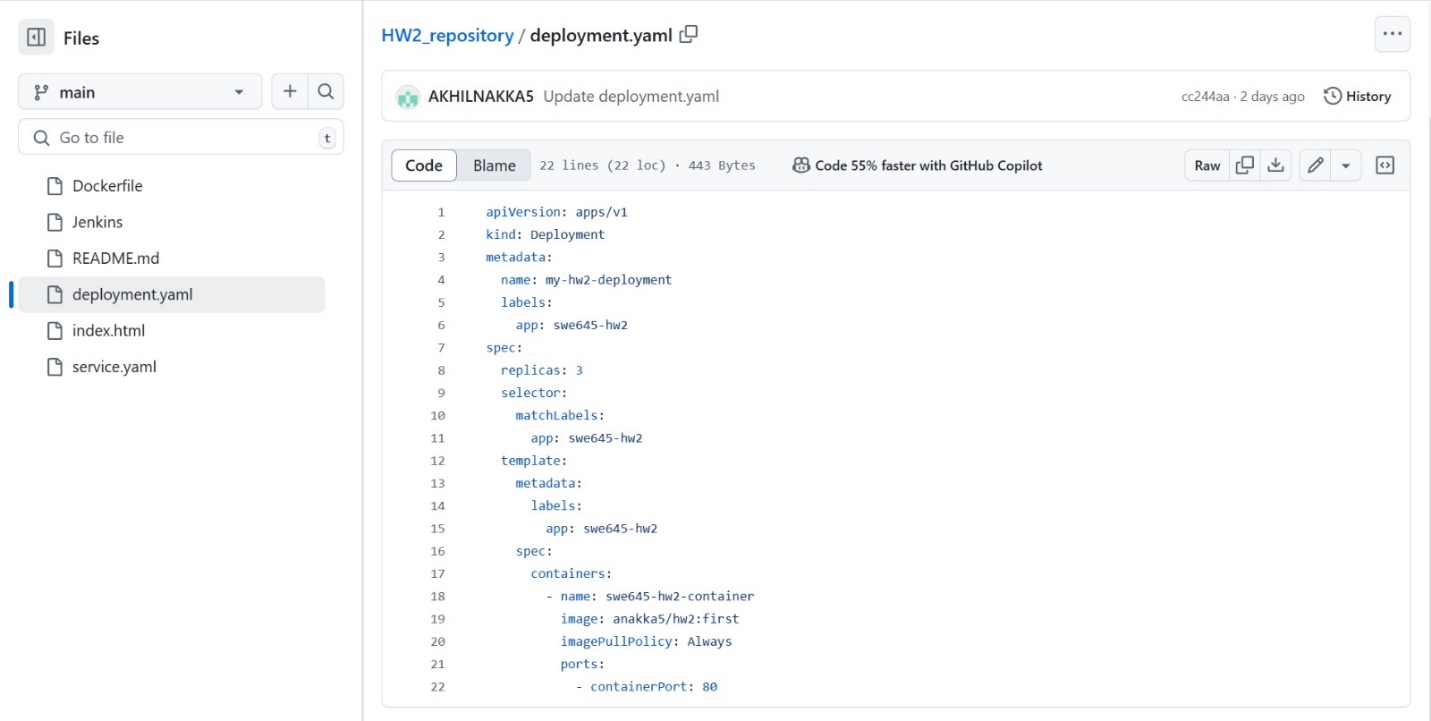


**Connecting to the K8S cluster using the Command line:**

* Install the Kubectl
  + %snap install kubectl --classic
* Create a directory .kube to store the config file, we can download it from the rancher
  + sudo mkdir -p ~/.kube



* Now, create the deployment.yaml and service.yaml and apply these to the Kubernetes cluster



A screenshot of a computer

Description automatically generated

* Then by using these commands, we applied the objects to the cluster.

○ kubectl apply–f deployment.yaml

○ kubectl apply–f service.yaml

* We mapped port 8080 to Nodeport 31416, in service.yaml
* Also, we allowed the traffic in port 31416 in the EC2 instance
* check if we can access the index.html by using the public EC2 instance in port 31416

**http://ec2-52-44-247-46.compute-1.amazonaws.com:31416/**

**Installing Jenkins:**

* Install the docker and Kubectl in the third instance, same as above steps
* Install Java 17
  + sudo apt update && sudo apt upgrade -y
  + sudo apt install openjdk-17-jdk -y
* Install Jenkins and start by using the commands
  + sudo apt install jenkins -y
  + sudo systemctl start jenkins
* Check the status of the Jenkins
  + systemctl status Jenkins
* open the browser on the public IPv4 address in port 8080

**Setting Up Jenkins:**

• A pipeline that generates the build using the Jenkins file and verifies any modifications to GitHub must be set up.   
• In the Jenkins user interface, select "New Items," type a name, and then select "Pipeline."   
• Every five minutes, a cron job will be put up to check our git repository for changes. If any are found, a new build will be started.

****

• Save with all other settings left at their default settings.   
• Select Configure and Manage Plugins from the Manage Jenkins section. Put the Kubernetes, Docker, and Git plugins on your computer.   
• Select credentials from the Manage Jenkins section. Enter the DockerHub username and password to add DockerHub credentials. In the same manner as before, add the git credentials. Choose the secret file option to add the Kubernetes configuration file.   
•To build the pipeline, configure the Jenkins file.



A close-up of a computer code

Description automatically generated

* The Jenkins file receives the Kubernetes configuration file and the Docker Hub credentials as environment variables.
* Consequently, it generates the Docker image, tags it with the Jenkins build number, and fetches our Github repository first.
* It will then push the image after logging into DockerHub with the mentioned credentials.
* To deploy to the Kubernetes cluster, we apply the deployment and service yaml files defined in the root directory of the git hub repository using kubectl commands after giving the build number to the deployment yaml in the next step.
* We made some modifications to index.html in the repository and committed them to produce the build.
* The Jenkins pipeline will now use Docker to create an image, which will subsequently be pushed to Docker Hub.
* After that, it will deploy the image to the Kubernetes cluster after retrieving it from Docker Hub.

A screenshot of a computer

Description automatically generated

**URL for containerized web application deployed in Kubernetes:**

[**http://ec2-52-44-247-46.compute-1.amazonaws.com:31416/**](http://ec2-52-44-247-46.compute-1.amazonaws.com:31416/)

**Jenkins login page:**

[**http://ec2-44-195-154-125.compute-1.amazonaws.com:8080/**](http://ec2-44-195-154-125.compute-1.amazonaws.com:8080/)

**Contributions**:

Saranya – Setting up EC2 instances and creating a docker image

Akhil – Setting up rancher and Kubernetes cluster

Samuel – Setting up the Jenkins pipeline