Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

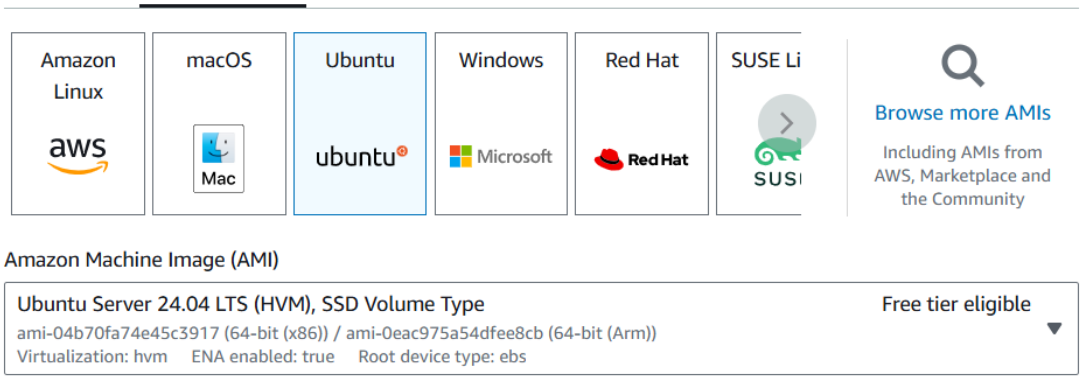
EC2 Link for deployed image on Kubernetes cluster: http://3.228.98.74:31020/hw2/

The following assignment involves launching EC2 instances, then setting up Rancher and Jenkins servers. Both these parts start with the creation of three separate EC2 instances:

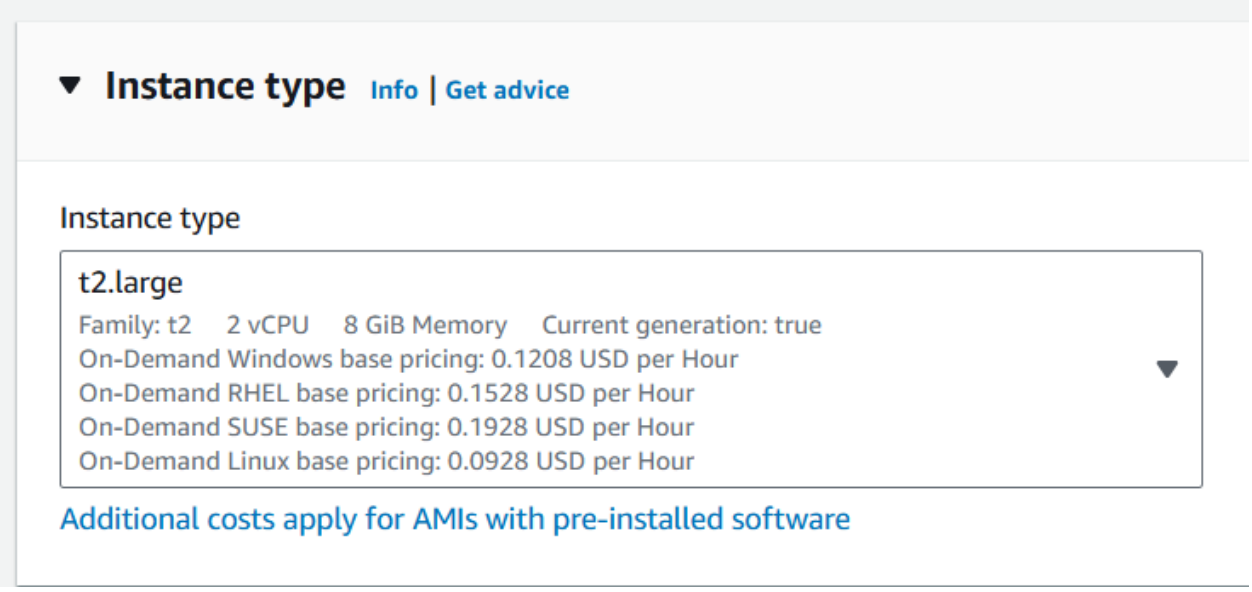
1. For running the **Rancher deployments** (which is where the app will reside).
2. For running the **Rancher server** and performing deployments to (1).
3. For running **Jenkins** that continuously checks a **GitHub repository**, build the war files & containerize the application, then deploys to the cluster.

Part 0: Launching Instances

1. Start by **creating three new EC2 instances**. Launch these instances with the following configurations:
    a. **Name**: <you can choose> (i.e. Rancher Server, Rancher Deployments, Jenkins Server)
    b. **Application and OS Images (Amazon Machine Image)**: Ubuntu Server 24.04

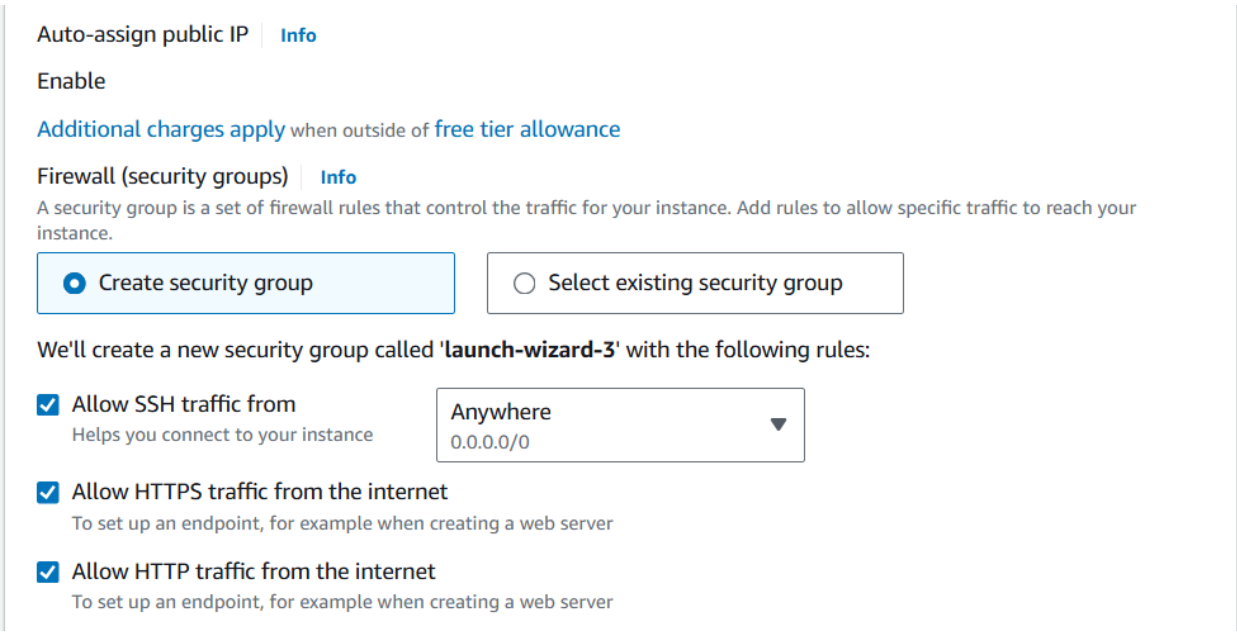


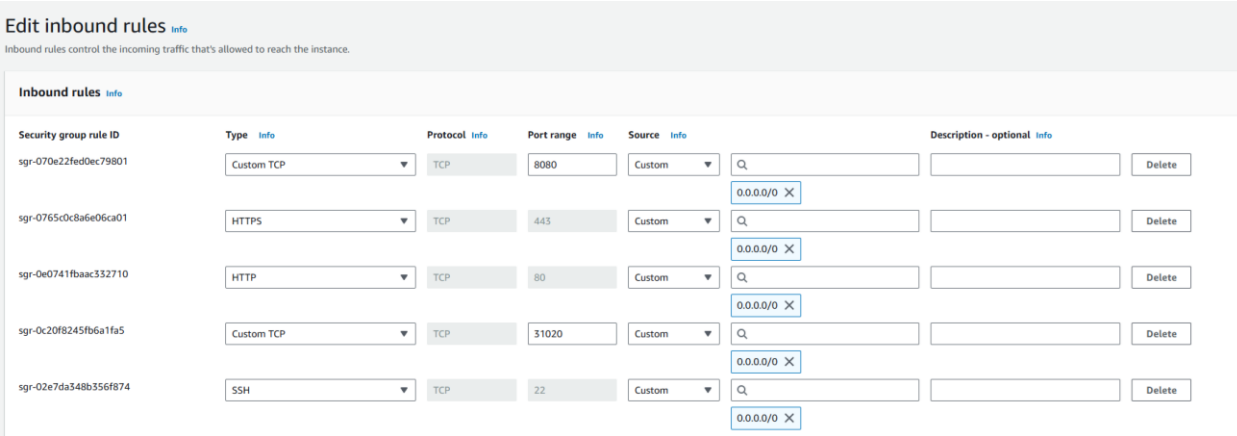
    c. **Instance Type**: t2.large

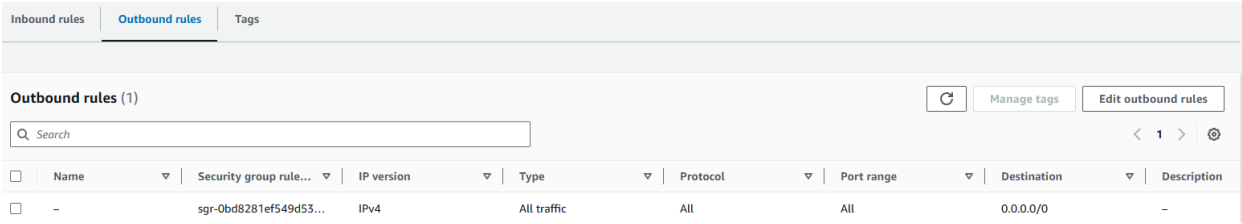


Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

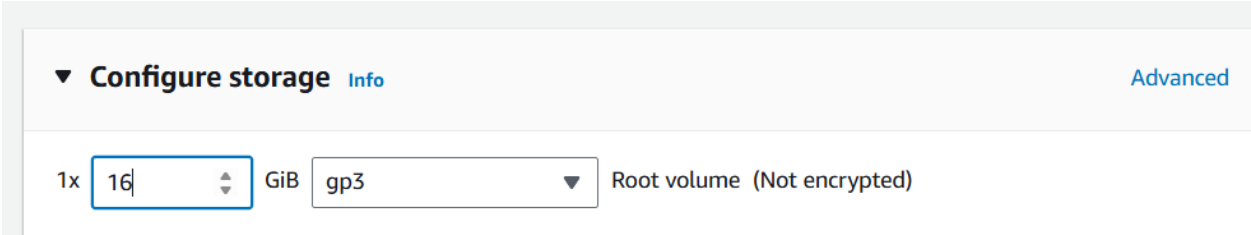
d. **Key pair (login)**: Create a new pair.
    i. Key pair name: <you can choose> (i.e. rancher_kp)
    ii. Private key format: .ppk
e. **Network settings**: Create a new **security group** with **inbound** rules allowing traffic from 8080, 80, 443, 22, and outbound rule to All traffic.





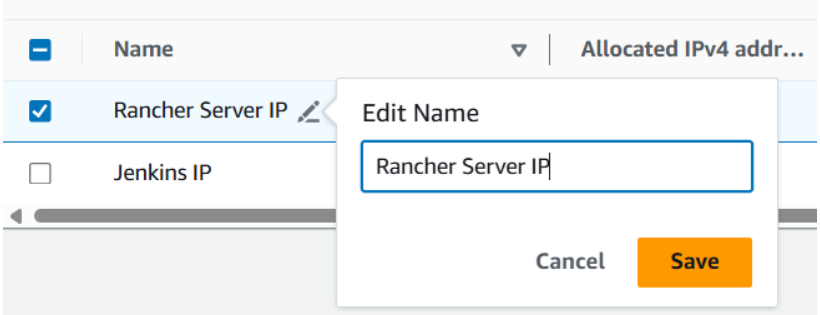


f. **Configure storage**: 16GB

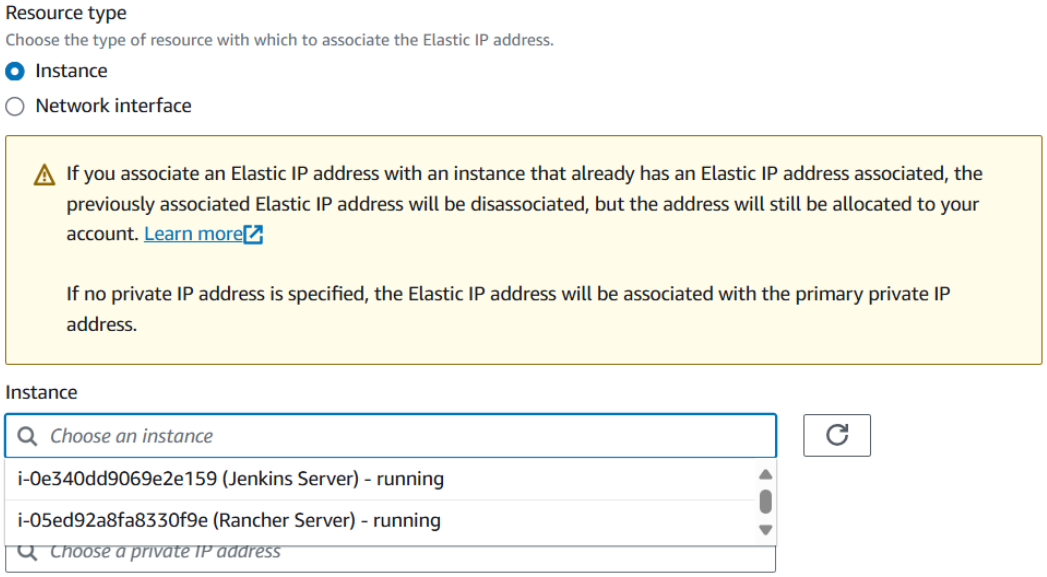


g. Finally, click **Launch Instance**

2. Set up an **Elastic IP Addresses** so that you don't lose the public IP address overtime for these:
   a. Lookup "**Elastic IP Addresses**" under VPC features
   b. Click **Allocate Elastic IP Address**
   c. Leave all default configurations and click **Allocate.**
   d. Click the pencil icon next to the new IP and assign a name to it.



   e. Click the new IP address link and click **Associate Elastic IP Address**
   f. Under Instance find the instance you created and select it.



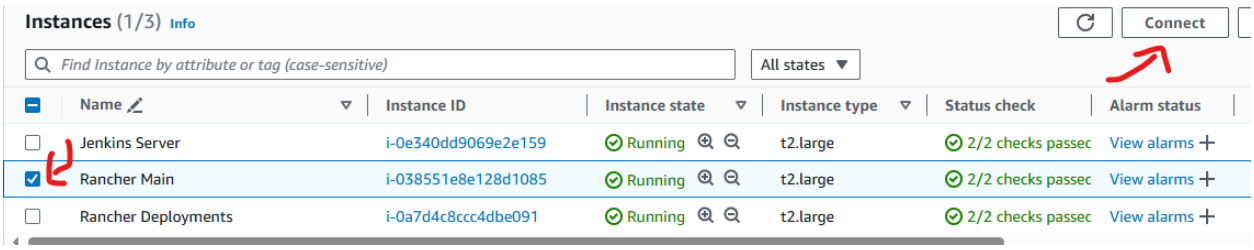
   g. Click **Associate.** Perform these same steps for all three new instances.

Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

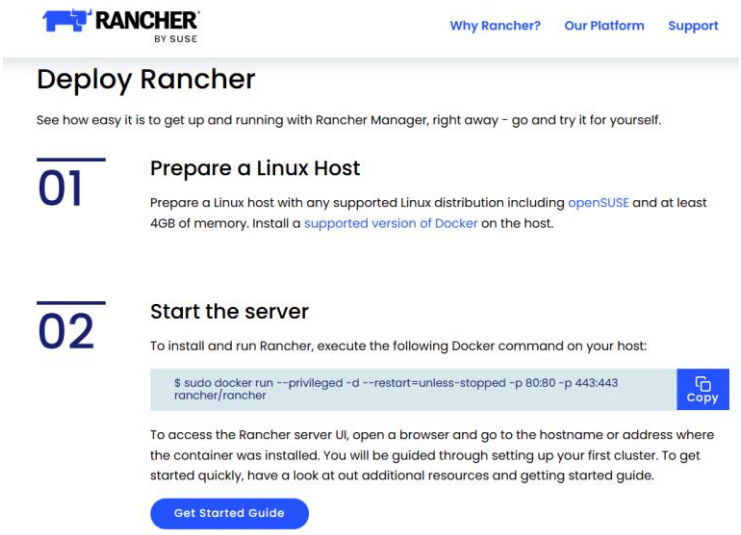
Part 1: Setting Up Rancher Server & Deployments Server

Now we'll set up Rancher for two of the instances—one to act as the main server, and another as the deployment's server

1. Start by **connecting to each of the new EC2 instances** by clicking one of the instances and using the **Connect** button.

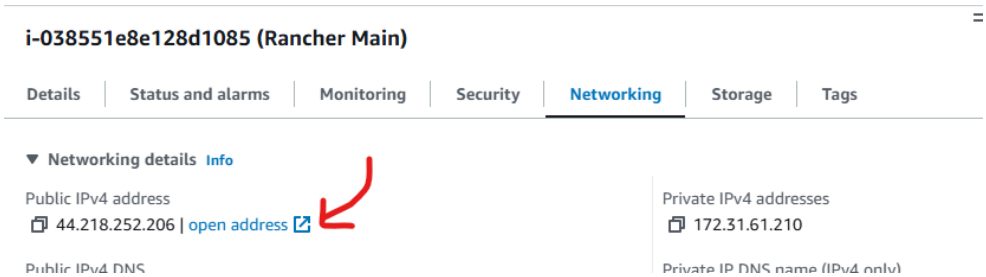


2. Leave everything as is on the next page (keep the ubuntu user), and just click **Connect** again.
3. Run the following commands **to each of the two instances:**
   a. **sudo su-**
      i. Allows you to skip typing **sudo** before each command
   b. **apt-get update**
   c. **apt install docker.io** (type Y when prompted)
4. We'll focus on the **main Rancher server setup** first.
   a. Start by going to **Rancher.com** and try to **copy** the **download script** from the website. It should be something like the following:
      i. **docker run –privileged=true -d --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher**
      ii. Note: In the moment of writing this assignment, it can be found under this link https://www.rancher.com/quick-start, in the **Start the server** section.

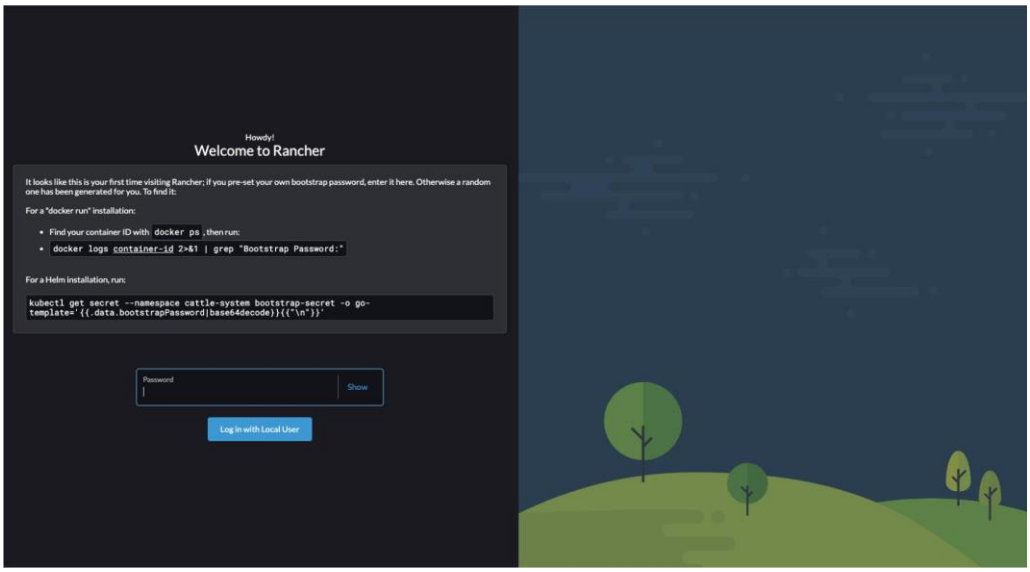


   b. Copy this script and paste it to the terminal of one of the two instances so it can download Rancher.

c. Once finished, open the server on another tab by clicking its **public IP address** as specified in **Networking** section of the EC2 configuration for that instance. Just proceed with the unsecured network warning.



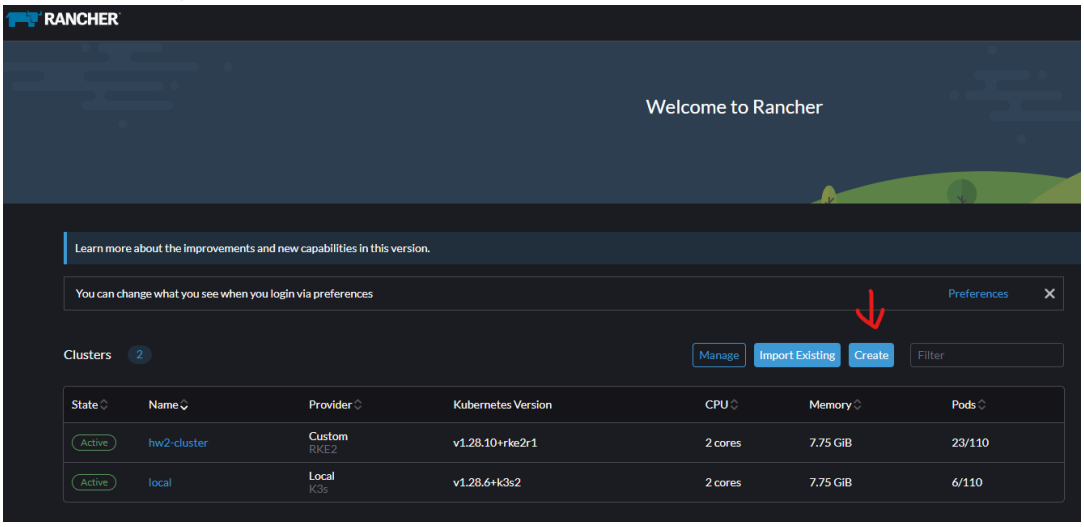
d. It should then present the following page giving instructions on how to set up the password.



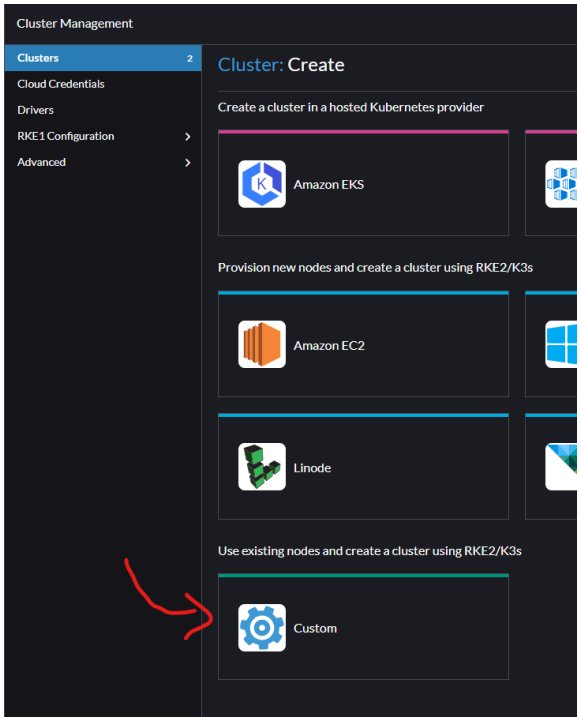
To set up the login credentials for Rancher (as specified in that page):

i. Type **docker ps** on the instance's terminal, and copy the **CONTAINER ID**.
ii. Type **docker logs <CONTAINER ID> 2>&1 | grep "Bootstrap Password:"**
iii. Copy the **Bootstrap Password:** output and paste it to the field in the server's website.
iv. You can choose to keep the randomly generated password or specify your own. Agree to the Terms and Conditions, then click **Continue**

5. Now we'll focus on **creating a new cluster and** setting up our **deployment server** (which is the other instance).
    a. On top of the **Clusters** section click **Create**

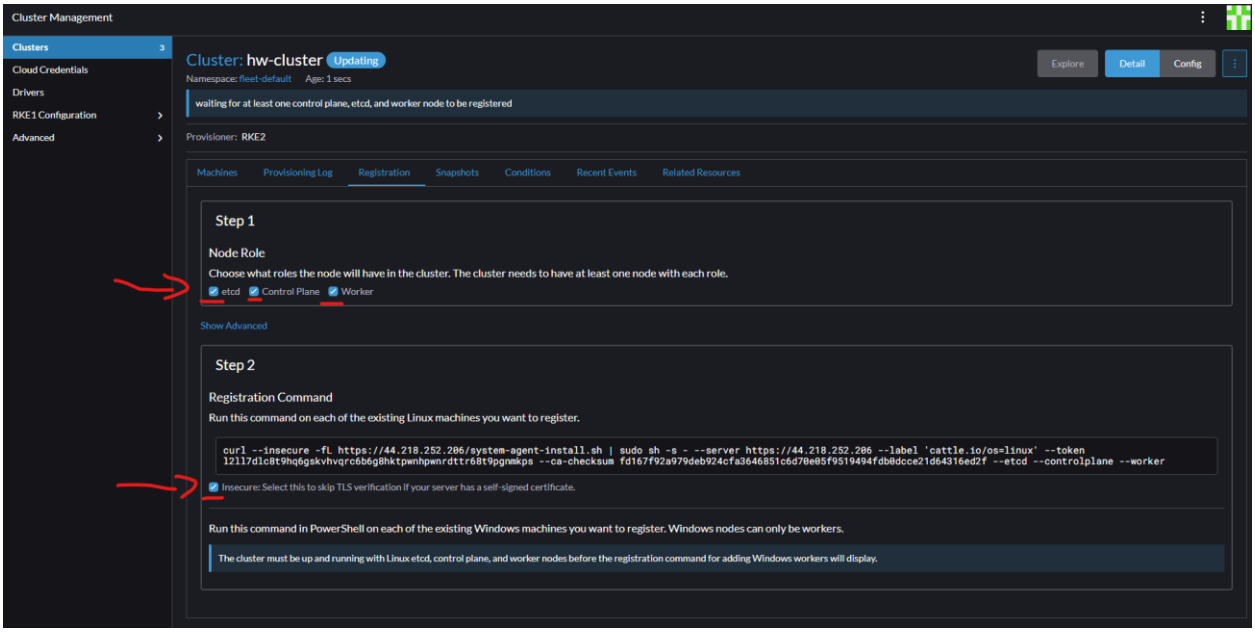


    b. Select **Custom**



    c. Specify a **Cluster Name** (i.e. hw2-cluster).
    d. Leave all fields in their default values and click **Create**.

e. You'll see a page that says Step 1 Node Role, and Step 2 Registration Command. Check off all three checkboxes (**etcd, Control Plane, Worker)** in Step 1, and check off the **Insecure** checkbox in Step 2.



f. Copy the **curl** command and paste it to the **terminal <u>of the second EC2 instance</u>** that will be the deployments server.

g. Go back the **Clusters** section on the top left under **Cluster Management** and you'll see the **State** of your new cluster go from a blue **Updating** to green **Active**.

h. Your deployments server is all set up! We'll go back again to this after creating our Docker image.

Part 2: Setting Up Dockerfile

1. You first need to create a **war file (.war)** of your application as that will be the deployable version of it. *If you already have one, you can skip this step.*
   a. Make sure all your **js**, **css**, **index.html** files are <u>in the same folder</u>.
   b. Add a **WEB-INF folder** with a **web.xml** file inside containing the following:

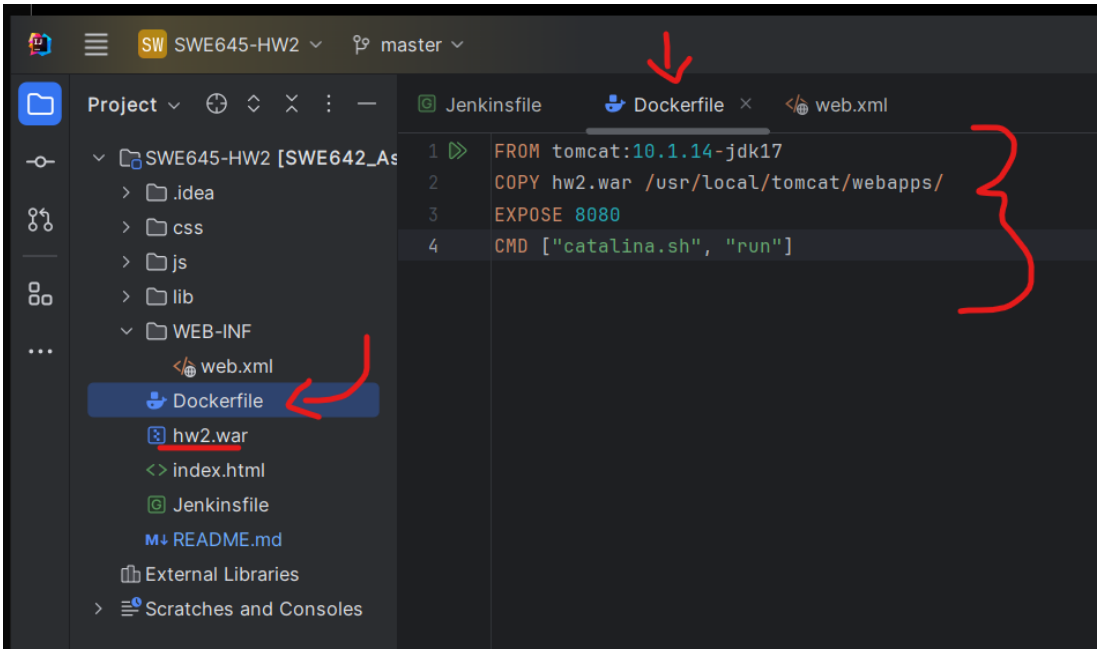
```
Jenkinsfile   Dockerfile   web.xml
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
         version="3.1">

    <display-name>HW 2 - Student Survery Form</display-name>
</web-app>
```

   c. Run the following command to generate your war file:
      i. **jar -cvf <a war file name>.war ***

Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

2. Create a **Dockerfile** in the same location as your war file with the following code:



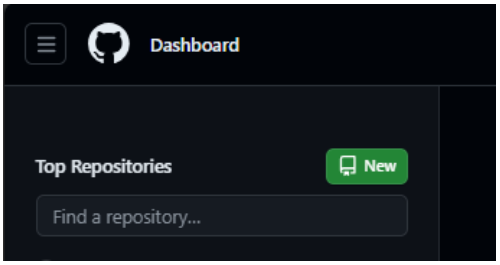
The following downloads a base **Tomcat** image, puts the war file inside the **webapps/** folder of Tomcat, exposes 8080 and runs the Tomcat server.

This new **Dockerfile** will be used by the next part to create an image of your application and push it to Docker Hub to be used by Rancher.

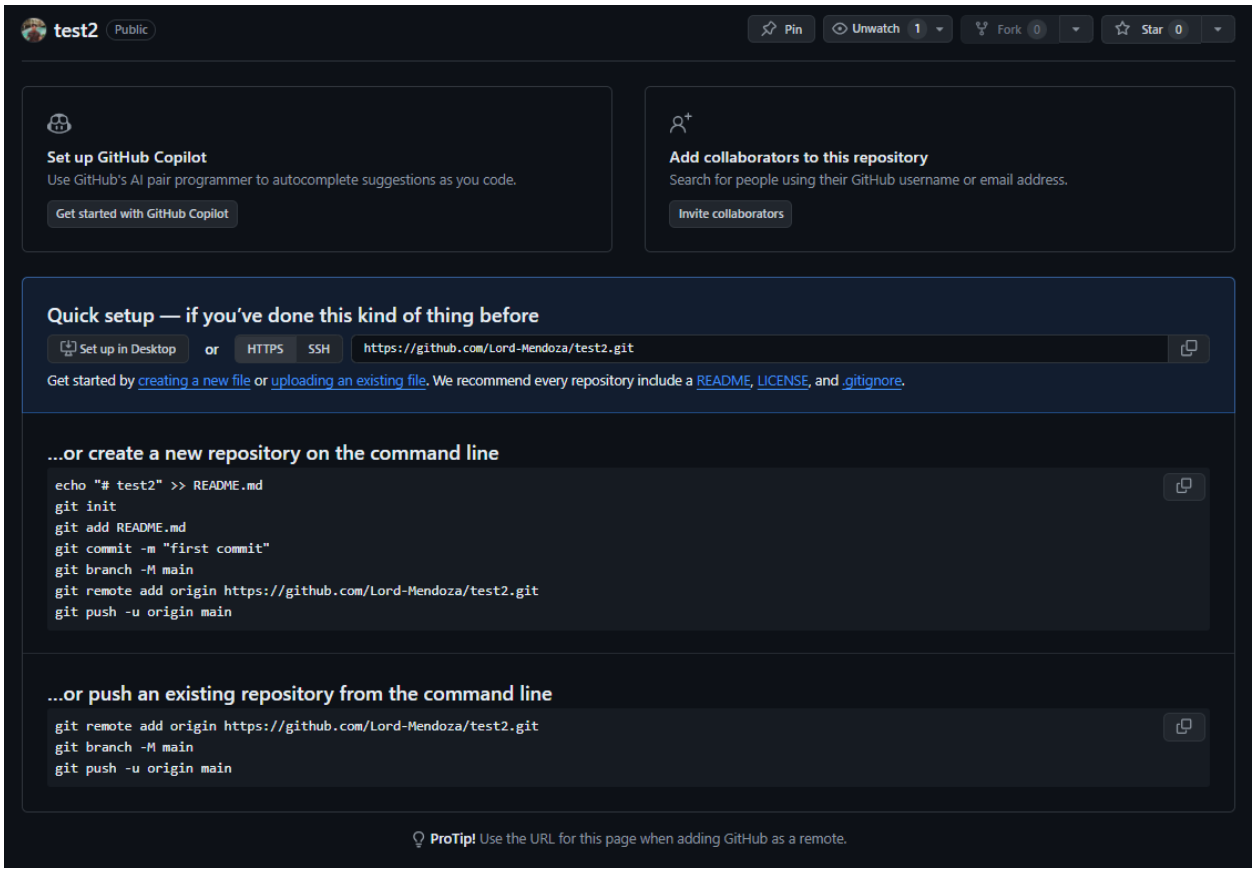
Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

Part 3: Uploading Project to GitHub

1. Before you start, make sure you already uploaded all your code and project to GitHub. *If you already have, skip to Part 4.*
    a. With a logged in account click **New**

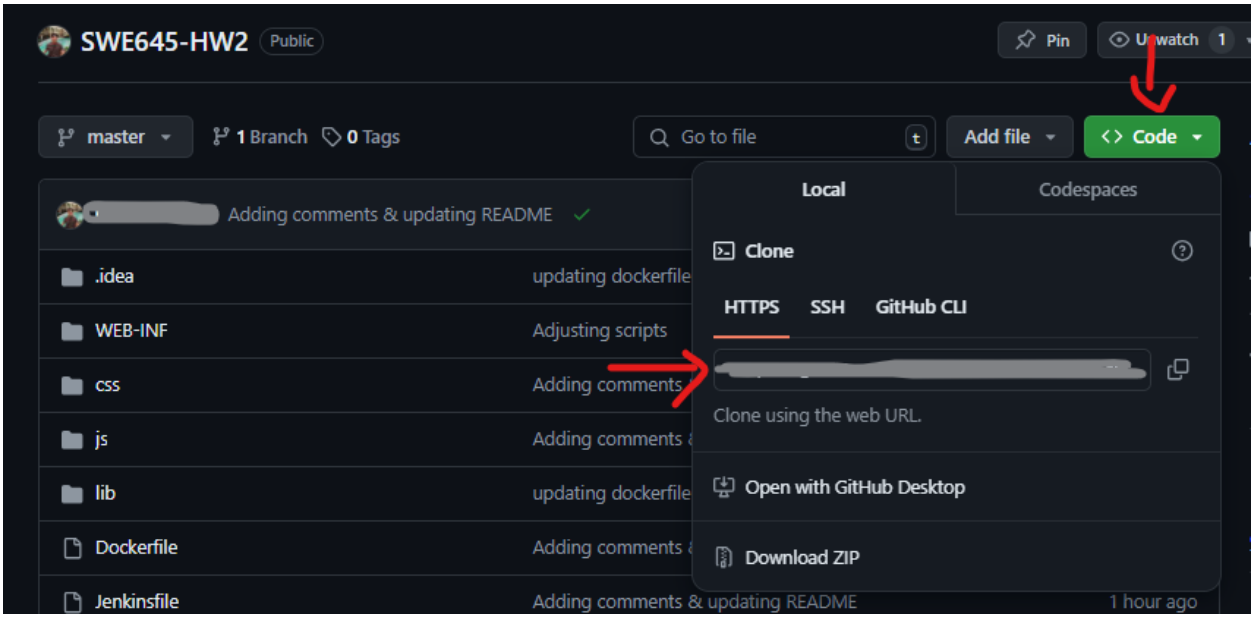


    b. Type a **Repository name** and click **Create Repository.**
    c. Use the scripts provided on the next page to push your code.



Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

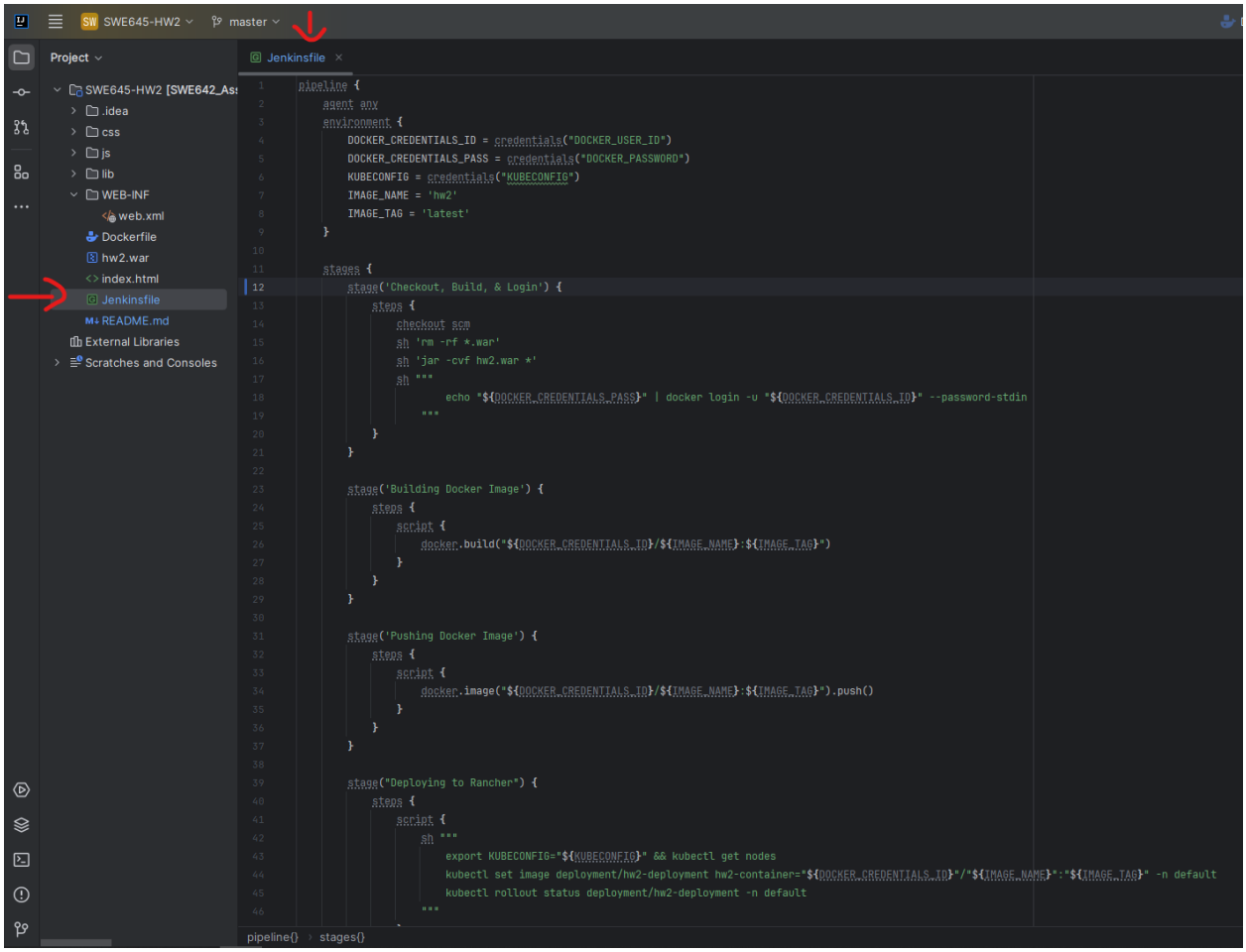
d. Once pushed, just copy the **HTTPS** URL so Jenkins can access this later.



Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

Part 4: Setting Up Jenkinsfile

1. Create a **Jenkinsfile** in the root of your project with the following script:



Explanations:

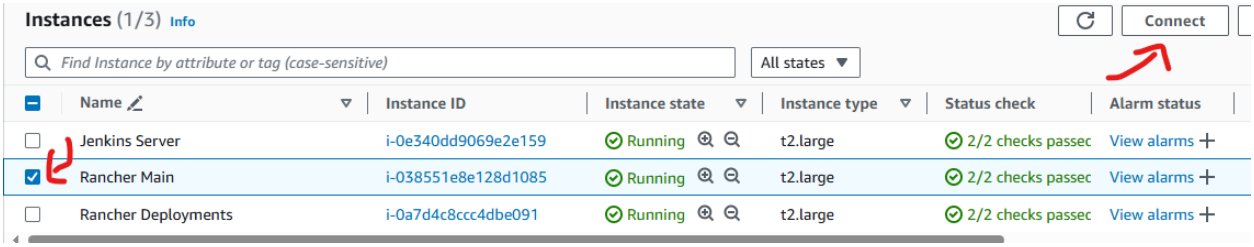
a. The **environment** section sets up all variables to be used by the script. The **credentials** call retrieves the secret values stored in the Jenkins server and assigns them to each of the variables.
b. The **stages** set up each of the stage:
   i. The first step checks out the latest code from GitHub for the target repository, deletes any existing project war file and creates it again. Then logs into docker.
   ii. The second step builds the docker image, which is same as calling **docker build.**
   iii. The third step pushes the docker image, which is same as calling **docker push**.
   iv. The last step calls the Rancher server and tells it to deploy the latest image version of your application.

2. Push your changes to GitHub.

Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

Part 5: Setting Up CI/CD with Jenkins Server

1. Connect to the EC2 instance you want the Jenkins to run on by clicking one of the instances and using the **Connect** button.



2. Leave everything as is on the next page (keep the ubuntu user), and just click **Connect** again.
3. Run the following commands**:**
   a. **sudo su-**
      i. Allows you to skip typing **sudo** before each command
   b. **apt-get update**
   c. **apt install docker.io** (type Y when prompted)
4. Go to **Jenkins.io/download** website, choose **Ubuntu/Debian** and run each of the scripts as specified.
   a. Note: At the time of writing this assignment it can be found in https://pkg.jenkins.io/debian-stable/.

Jenkins Debian Packages

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system (for the Weekly Release Line):

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

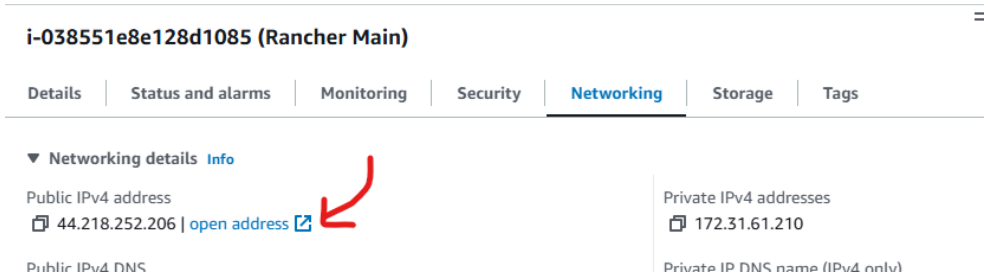
```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

Update your local package index, then finally install Jenkins:

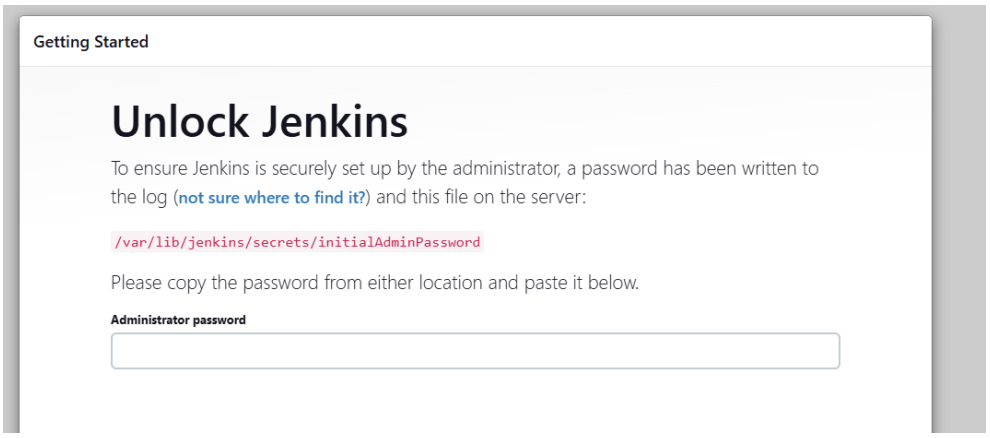
```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

   b. Also run the following commands to add support for Docker and Rancher commands:
      i. **apt install snapd**
      ii. **snap install kubectl --classic**
      iii. **usermod -aG docker jenkins**
   c. Reboot the instance. You can check the status of the Jenkins server using
      i. **systemctl status jenkins**

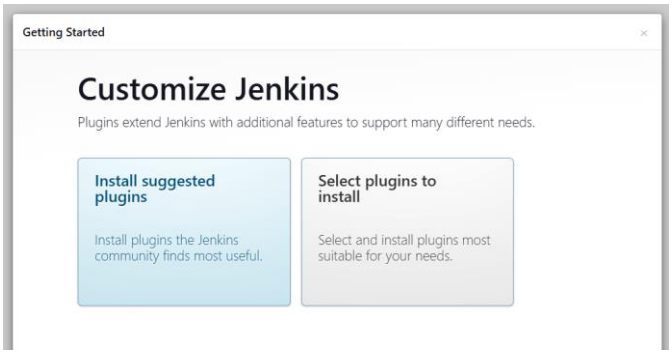
5. Once restarted, open the server on another tab by clicking its **public IP address** as specified in **Networking** section of the EC2 configuration for that instance.



   a. **<u>Make sure you're not on HTTPS and add port 8080 to your URL.</u>**
      i. ex. **http://**<public IPv4 address>**:8080**
6. You'll be taken to a login page that prompts you to retrieve your **initialAdminPassword**.



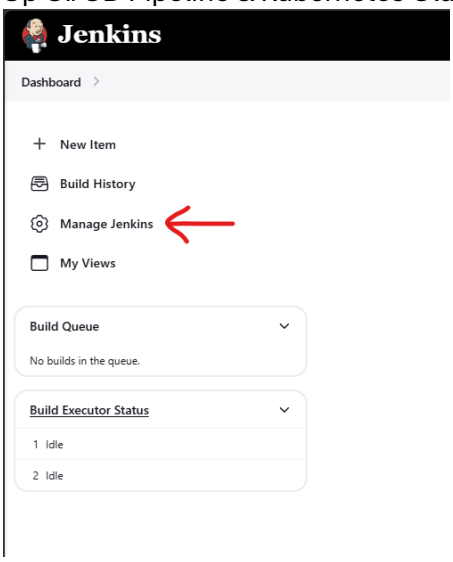
   a. To get your initial password use the following command on your instance's terminal.
      i. **sudo cat /var/lib/jenkins/secrets/initialAdminPassword**
   b. Paste the password to this field.
7. Choose **Install suggested plugins**.



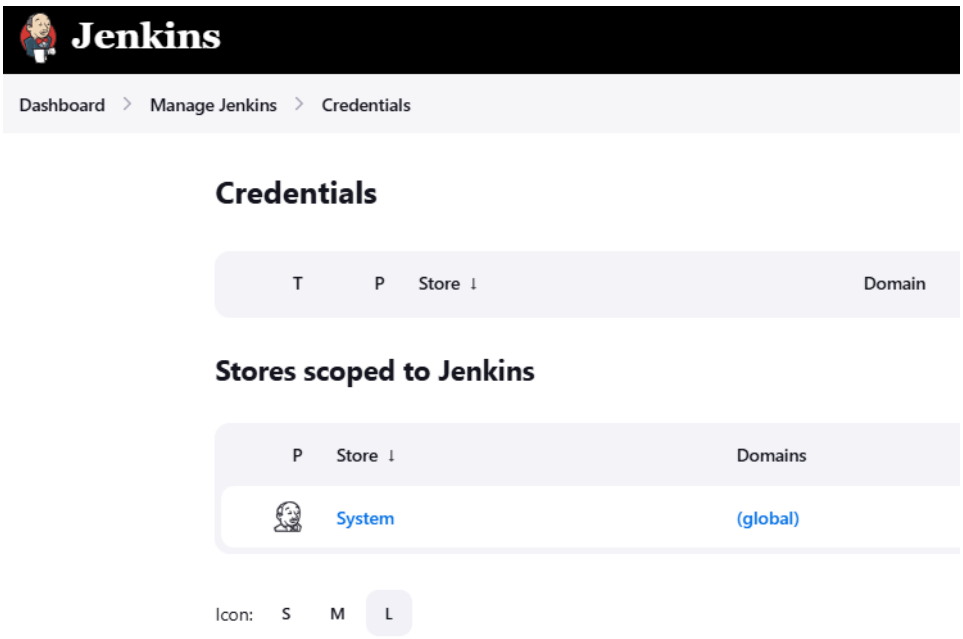
8. Set up your **username** and **password**, or you can **skip and stick with the admin setup** (although you'd have to keep the long initial password).
9. It will then give you the **Jenkins server URL** which you can save/bookmark, then click **Save**.

10. On the home page click **Manage Jenkins**

Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

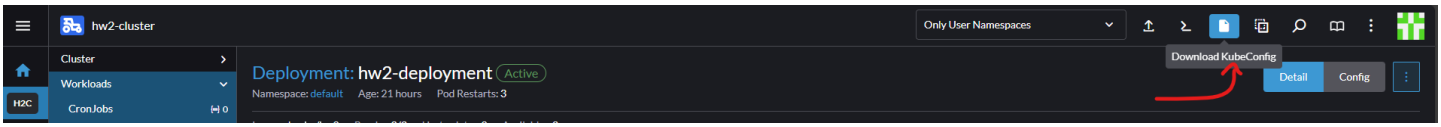


11. We'll be setting up the **Credentials** to keep the secret files/text and downloading the **Docker** plugin so the Jenkinsfile we create later will have them available.
    a. Go to **Credentials** and under **Stores scoped to Jenkins** click **(global)** under **Domains** column



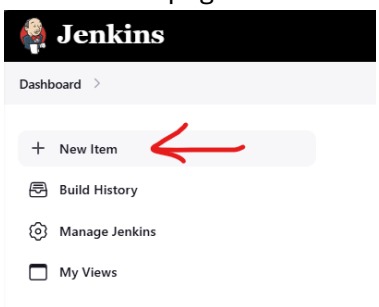
    b. Click **Add Credentials** and add the following:
        i. (For GitHub) **Kind:** Username with password. Paste the GitHub session key you generate from https://github.com/settings/tokens to the **Password** field, then click **Save**.
        ii. (For Docker ID) **Kind**: Secret text, **ID**: <You can choose> (i.e. DOCKER_ID), **Secret**: <your Docker ID>
        iii. (For Docker Password) **Kind**: Secret text, **ID**: <You can choose> (i.e. DOCKER_P), **Secret**: <your Docker Password>

iv. (For Rancher): **Kind**: Secret file, **ID**: <You can choose> (i.e. Kubeconfig), **Secret File**: <your KUBECONFIG file> from your Rancher which you can download from the header.

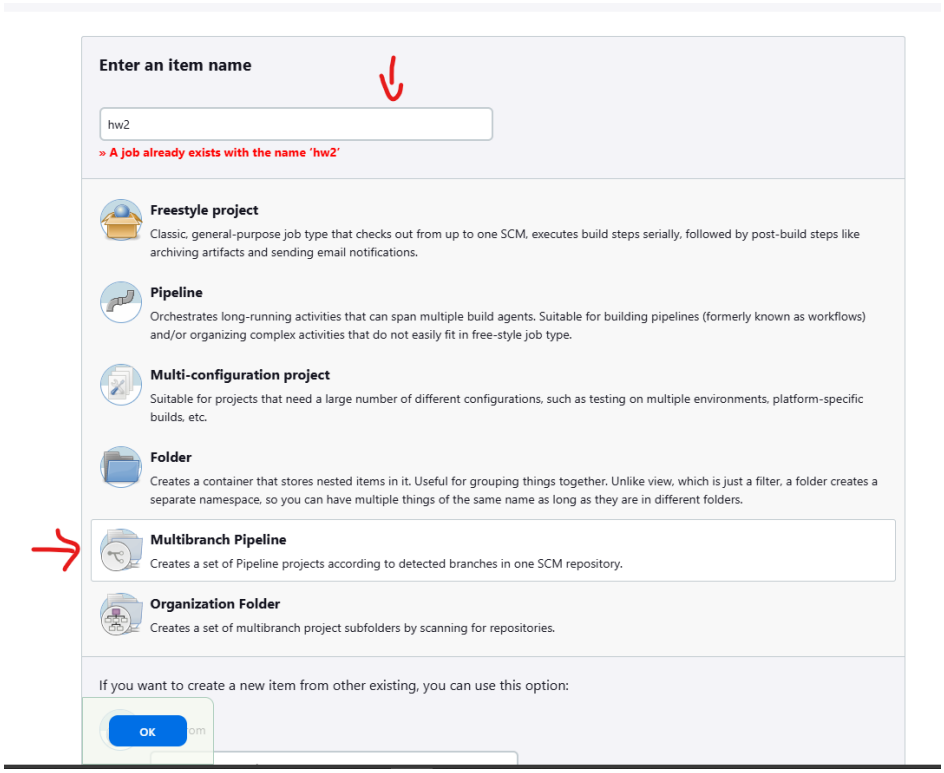


c. Go back to Manage Jenkins and click **Plugins**. Go to **Available Plugins** and install **Docker Pipeline.** You can check the check box for auto-restarting the server upon download.

12. Next, we'll set up a build pipeline:
    a. On the homepage click **New Item**



b. Fill in **Item Name** (i.e. hw2) and choose **Multibranch Pipeline**.



c. Provide the following:
    i. **Display Name** (i.e. hw2)

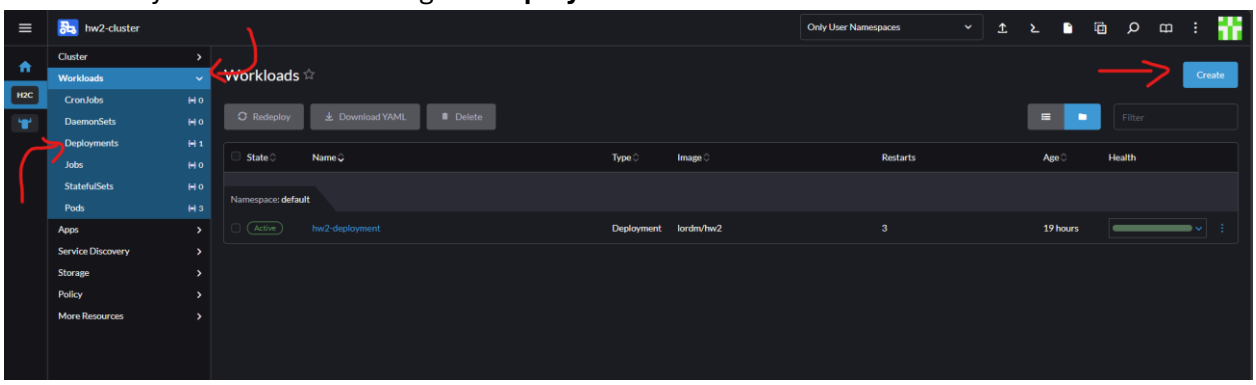
ii. **Branch Sources**: GitHub
    1. **Credentials: <**Select the GitHub token you added earlier>
    2. **Repository:** <your repository URL from GitHub>
iii. **Build Configuration**
    1. Mode: by Jenkinsfile
        a. Script Path: Jenkinsfile
iv. **Scan Repository Triggers**
    1. **Periodically if not otherwise run**
        a. **Interval**: 1 hour

d. Click **Save**

13. The pipeline should pull your latest code and find the **Jenkinsfile** you created in Part 4 and use that as the build script. It will create the war file, create the image and push it to Docker repository, and try to push it to Rancher but will fail. **After you complete Part 5** and do a manual build/wait for the next build in an hour it will build successfully.

Part 6: Finalizing Application Deployment

1. Click on your new cluster and go to **Deployments** under the **Workloads** section.



a. Click **Create** on the top right and specific the following on the **Deployment: Create** form:
    i. **Namespace**: default
    ii. **Name**: <You can choose> (i.e. hw2-deployment)
    iii. **Replicas**: 3
    iv. **Container Name**: <You can leave as default, or specify> (i.e. hw2-container)
    v. **Container Image**: <Your container image name> (i.e. <docker userid>**/**<image name>**:**<image tag>
    vi. **Networking**: Add Port or Service
        1. Service Type: Node Port
        2. Name: <You can choose> (i.e. hw2-port)
        3. Private Container Port: 8080
        4. Protocol: TCP
        5. Listening Port: <Leave blank>

Lord Mharthelle Mendoza
SWE 645 – HW2
Setting Up CI/CD Pipeline & Kubernetes Cluster for Student Survey

vii. Click **Create** again on the bottom.