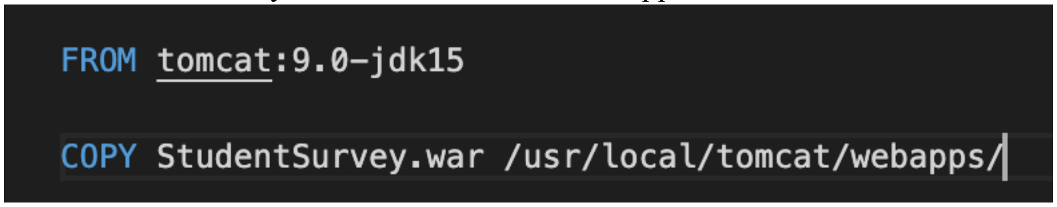
Emile Issaelkhoury
SWE645 – HW2 – Installation and Setup Instruction

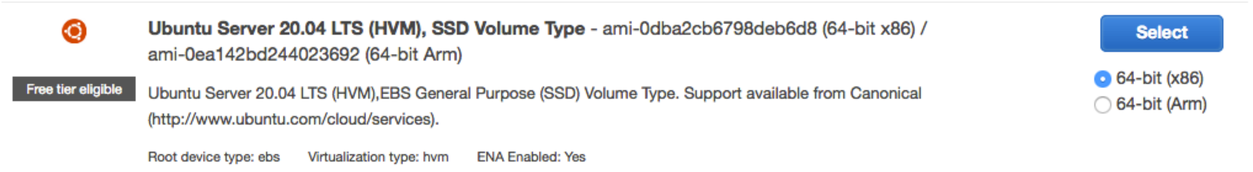
1. Setting up BitBucket
Push the files that you create in HW1 to a git repo (GitHub or BitBucket). First create an empty repository and then use Eclipse Teams' commands to push the code a new repository.

2. Creating a Docker image and pushing it to DockerHub
a. Make sure you have docker installed on your machine. I am using Docker Desktop for Mac. You will also need to create an account on https://hub.docker.com/
b. In Eclipse, create a file called Dockerfile. Docker requires the file to be called 'Dockerfile'
c. Build the HW1 project on eclipse and put the war file in the same folder as the Dockerfile. Note that 'StudentSurvey' is the display name of my Tomcat application.
d. In the DockerFile, use the FROM command to get the initial image for the build. We want to run the war file in Tomcat so we should use the tomcat image:
FROM tomcat:9.0-jdk15
e. Next, we need to drop the war file in the webapps folder:
COPY StudentSurvey.war /usr/local/tomcat/webapps/

```
FROM tomcat:9.0-jdk15

COPY StudentSurvey.war /usr/local/tomcat/webapps/
```
f. On the command line, use this command: 'docker build --tag studentSurvey645:0.1 .' You can use whatever name and tag you want.
g. Verify that the image is properly working by running 'docker run -it -p 8182:8080 studentsurvey645:0.1' and open a browser at http://localhost:8182/StudentSurvey
h. On the command line, login to docker using 'docker login -u <your username>'
i. Change the name of you image to be <your username on dockerhub>/<name of the app>:<image tag> using the docker tag command. In my case it is: 'docker tag studentsurvey645:0.1 hekme5/studentsurvey645:0.1'
j. Verify that your image is on Docker Hub. Your image is accessible from the internet

2. Installing Rancher on AWS
a. Log in to the AWS console https://aws.amazon.com/ and create an account. You will need to have a credit card. You can also use an AWS Educate but it has limited capabilities www.awseducate.com
b. I am running Rancher on an Ubuntu EC2 but all you need is an instance that has docker running.

c. In the AWS Console, click on Services then EC2 and click Instance and 'Launch instance'. Using the Ubuntu AMI from the list



Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0dba2cb6798deb6d8 (64-bit x86) / ami-0ea142bd244023692 (64-bit Arm)

Free tier eligible

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

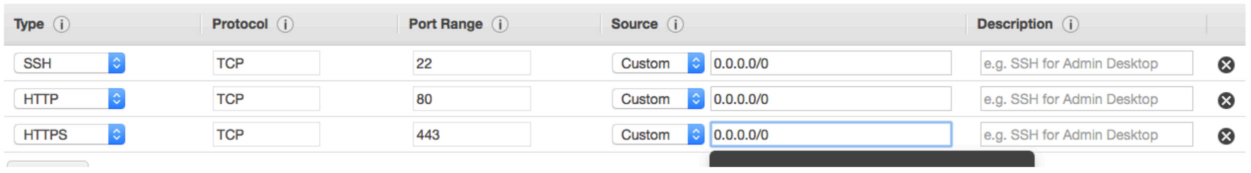
Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Select

[✓ 64-bit (x86)] 64-bit (Arm)

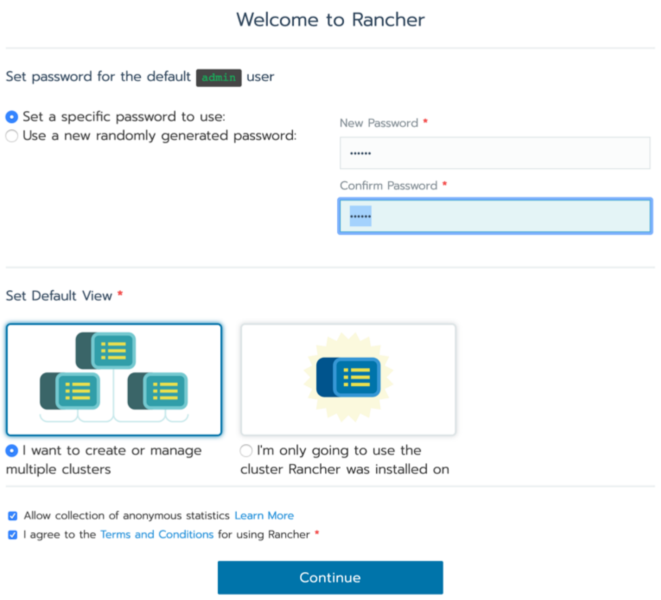
d. Under Instance Type choose t2.medium. I tried to run it on the free tiers but Rancher wouldn't start.

Click Next until you get to the 'Security Group'. You need to add ports 443 and 80 to 0.0.0.0/0



Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

e. Click 'Review and Launch' and then Launch. Make sure you save your pem key.
f. Once the instance is up and running, get the public IP address and public DNS. Ssh into it by running the following command from the terminal (or Putty for windows): ssh -i <YOUR KEY LOCATION> ubuntu@<YOUR PUBLIC IP>. You might have to change the permission on your key to be read-only.
g. Once you are in the instance, updated it by running: 'sudo apt-get update'
h. Install docker: 'sudo apt install docker.io'
i. Verify that docker is installed by running 'docker -v'
j. Run this docker command using 'sudo docker run --privileged=true -d --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher'
k. Wait a minute or so then open your browser and copy/paste your public DNS from step f. After some warnings, you should get this screen:



Welcome to Rancher

Set password for the default admin user

[✓ Set a specific password to use:]
Use a new randomly generated password:

New Password *
••••••

Confirm Password *
••••••

Set Default View *

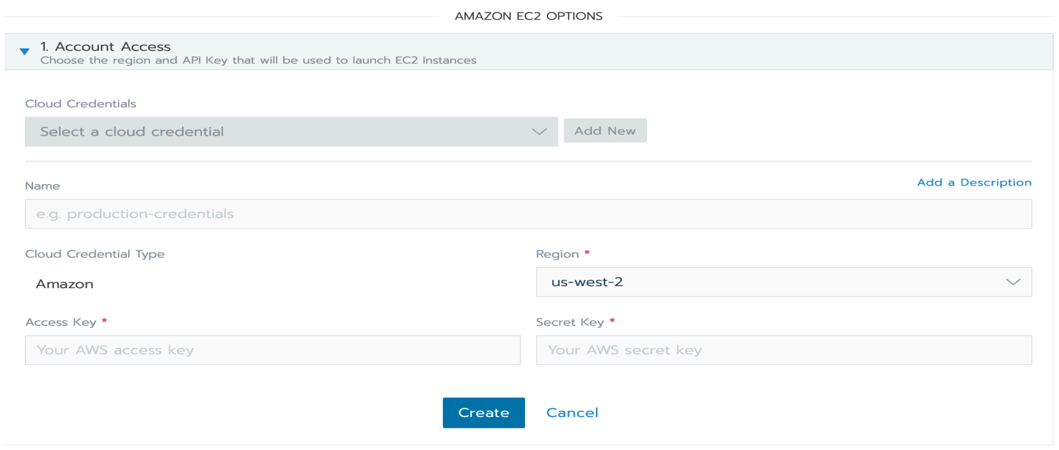
[✓ I want to create or manage multiple clusters]
I'm only going to use the cluster Rancher was installed on

☑ Allow collection of anonymous statistics Learn More
☑ I agree to the Terms and Conditions for using Rancher *

Continue

l. Create a password to the admin user. Installing Rancher installs Kubernetes

3. Starting a Rancher cluster
   a. In the global Rancher page, click on 'Add Cluster'
   b. You can choose any type of cloud provider you want; I chose Amazon EC2
   c. Give your cluster a Name
   d. Next you will need to create a template for your master/etc/worker nodes. I used the same template for all the nodes types. Click on 'Add node template'


AMAZON EC2 OPTIONS

1. Account Access
Choose the region and API Key that will be used to launch EC2 Instances

Cloud Credentials: Select a cloud credential | Add New

Name: e.g. production-credentials | Add a Description

Cloud Credential Type: Amazon

Region *: us-west-2

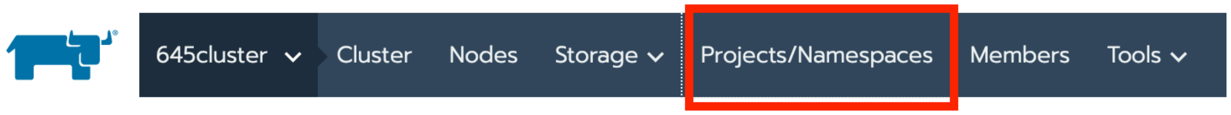
Access Key *: Your AWS access key

Secret Key *: Your AWS secret key

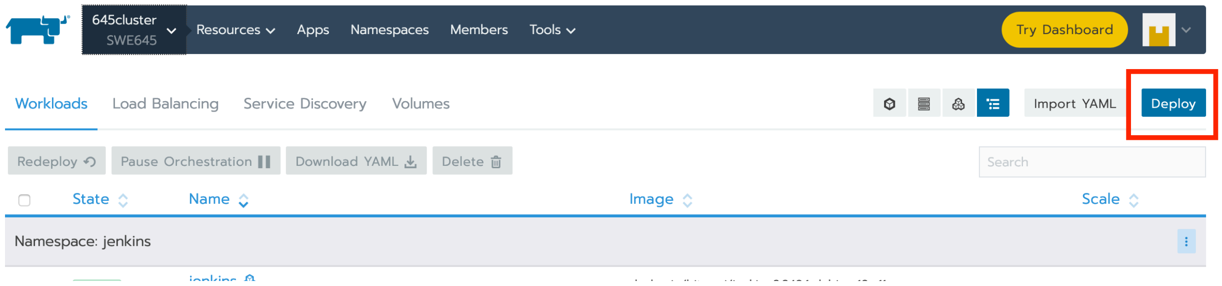
Create | Cancel

   e. You will need to create a user from your AWS account that can create and manage EC2 instances. For that you will need to go back to the AWS console and click on Service and IAM
   f. Click on 'Add User', give it a name and click on the 'Programmatic access' to get the Access key and Secret key.
   g. Click Next into permission and click on 'Attach Existing Policy directly'. Add the following permission 'AdministratorAccess'.
   h. Click Next until you Reach 'Create User'. Make sure you save the Access Key and Secret Key
   i. Copy both keys to the screen on step d. and choose the appropriate region (us-east-1)
   j. Click Create and don't change your VPC or security group
   k. Choose the type of instance you want to use. You will also have to choose an AMI from the AMI list that is provided by Rancher.
   l. Under 'IAM Instance Profile Name', put the name of the IAM user that you created.
   m. Give a name to the template and click on Create.
   n. Your template will show under template. It is advisable to have one instance dedicated to runs etc and the 'Control pane' and have the workers running by itself.
   o. Leave all the rest as default and click Create. Rancher will take some time to provision your cluster.
   p. Once the cluster is ready, you will be able to see the health worker nodes under your cluste. You will also be able to see the new instances created on the AWS console.

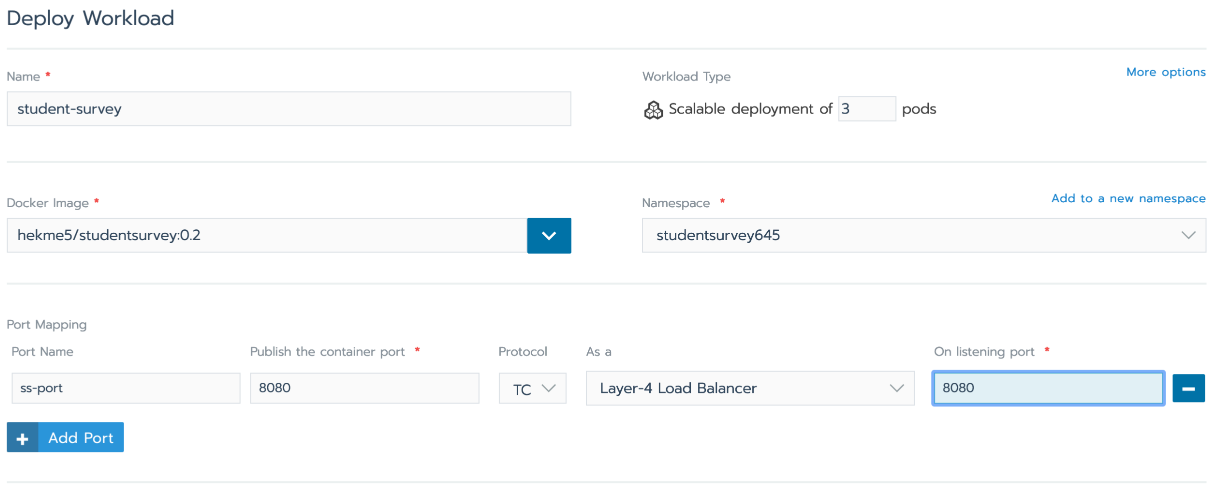
4. Deploying the docker image to rancher
   a. In Rancher, click on the top left side and choose your cluster.
   b. Click on Projects/Namespaces. Create a project and a namespace to make it easier to locate your running pods



c. Click on your cluster on the upper left and choose your project
d. Click on deploy in the upper right



e. Fill out the deploy form. As a docker images, use the image that you pushed to Docker Hub in step 2-j. Choose the appropriate namespace. We need to have three pods running so we choose a load balancer as a Service.


Deploy Workload

Name * student-survey

Workload Type: Scalable deployment of 3 pods

Docker Image * hekme5/studentsurvey:0.2

Namespace * studentsurvey645

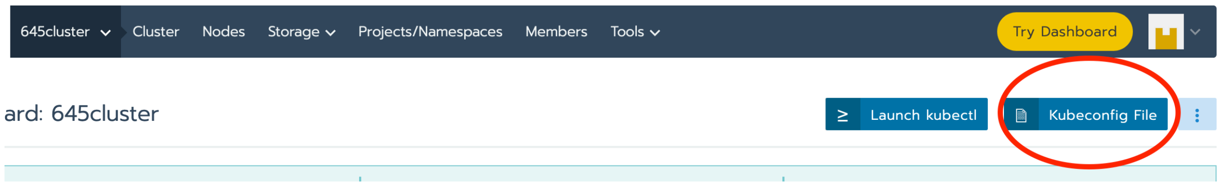
Port Mapping

Port Name	Publish the container port *	Protocol	As a	On listening port *
ss-port	8080	TC	Layer-4 Load Balancer	8080

f. Click Launch and wait a couple of minute for AWS to setup the load balancer. You can check the status by clicking on the ‘Load Balancing’ tab.
g. Under your pods, a ‘8080/tcp’ link will appear. Click on the url and add the display name: /StudentSurvey in my case. The application should load and should see the survey.

5. Installing Jenkins
   a. Follow the steps from 2-b to 2-i by creating another ubuntu instance. I installed Docker on Jenkins because I will be using it in the jenkinsfile later
   b. Install JDK 11: ‘sudo apt install openjdk-11-jdk’
   c. Add Jenkins repo using these commands:
   ‘wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add – ‘
   ‘sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list' ‘

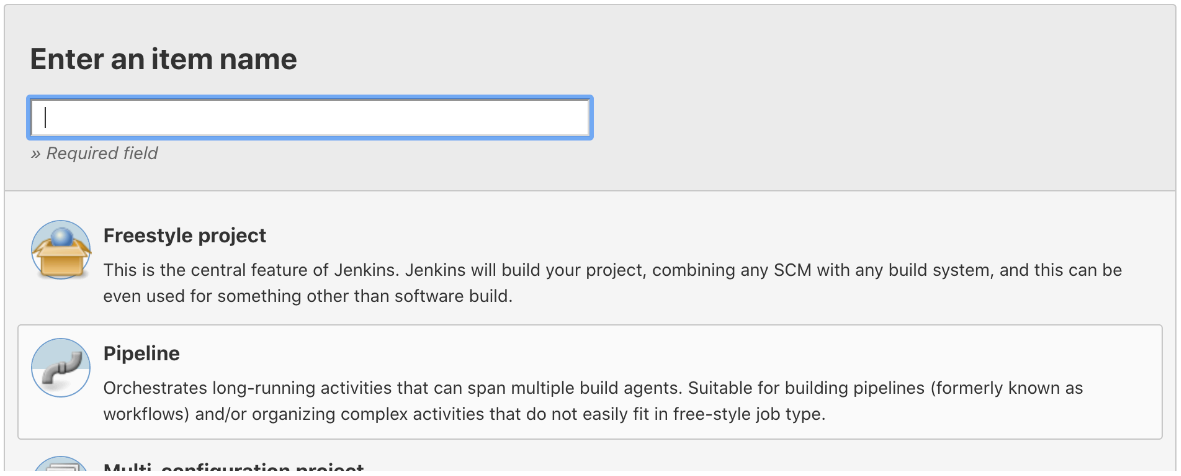
d. Install Jenkins by running:
   'sudo apt update' and then 'sudo apt install jenkins'
e. Check the status of Jenkins install by running 'systemctl status jenkins' and opening a browser on <public IP of the instance>:8080
f. I also installed kubectl on the Jenkins instance since I will be using kubectl commands in the jenkinsFile: 'sudo apt install snapd' and 'sudo snap install kubectl --classic '
g. Login as the Jenkins user: 'sudo su jenkins'
h. In Rancher click on your Cluster and then on 'Kubeconfig File'



i. Copy the content of the Kubeconfig file and paste it in /home/Jenkins/.kube/config file. You will need to create the folder '.kube' in Jenkins home.
j. Verify that the kubectl is working by running 'kubectl config current-context' and getting the name of your cluster

6. Setting up Jenkins
We need to set up a pipeline that checks any changes to BitBucket and build using the Jenkinsfile

a. In the Jenkins UI, click on 'New Items', enter a name and then click on Pipeline



Enter an item name

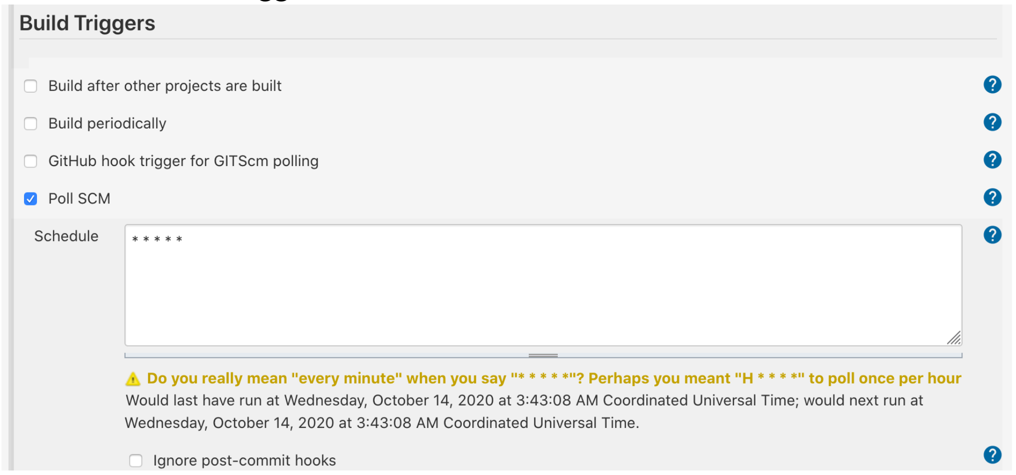
» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project

b. Since we need Jenkins to check for every changes, we set up a cron job that checks every minute as a build trigger



Build Triggers

- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ GitHub hook trigger for GITScm polling
- ☑ Poll SCM

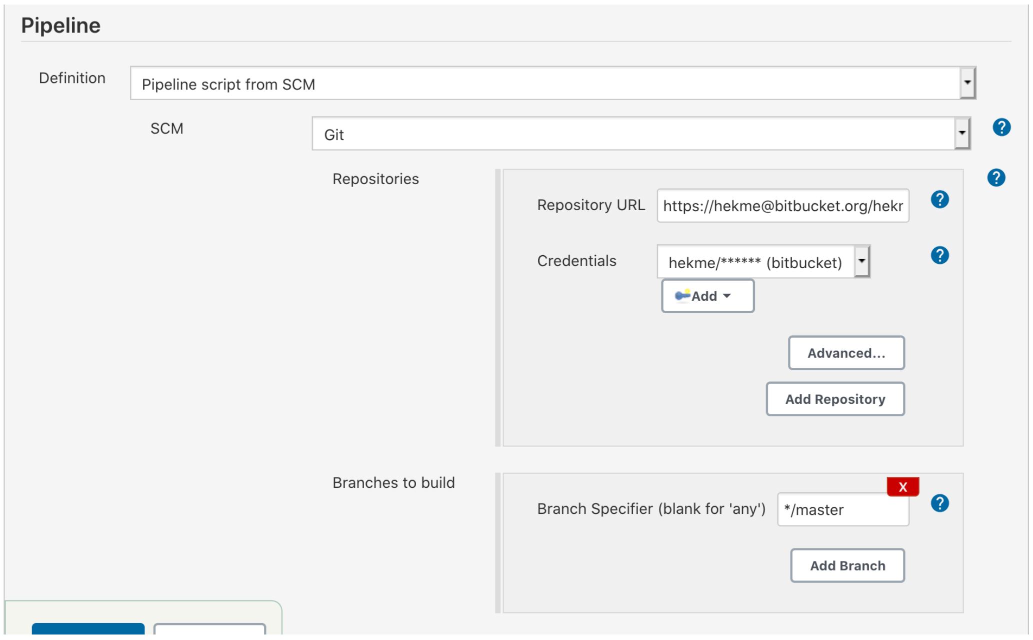
Schedule: `* * * * *`

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Wednesday, October 14, 2020 at 3:43:08 AM Coordinated Universal Time; would next run at Wednesday, October 14, 2020 at 3:43:08 AM Coordinated Universal Time.

- ☐ Ignore post-commit hooks

c. Next, we connect to our BitBucket repository by providing git url and credentials as a cloud credentials



Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

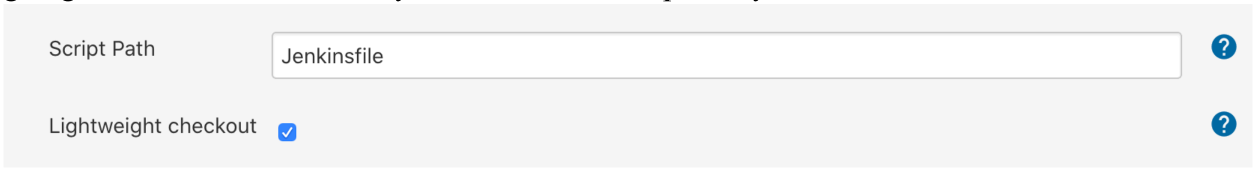
Repository URL: https://hekme@bitbucket.org/hekr

Credentials: hekme/****** (bitbucket)

Branches to build

Branch Specifier (blank for 'any'): */master

d. We also need to let Jenkins know where we will put the jenkinsfile. In our case, it is going to be in the root directory of the BitBucket repository as ‘Jenkinsfile’.



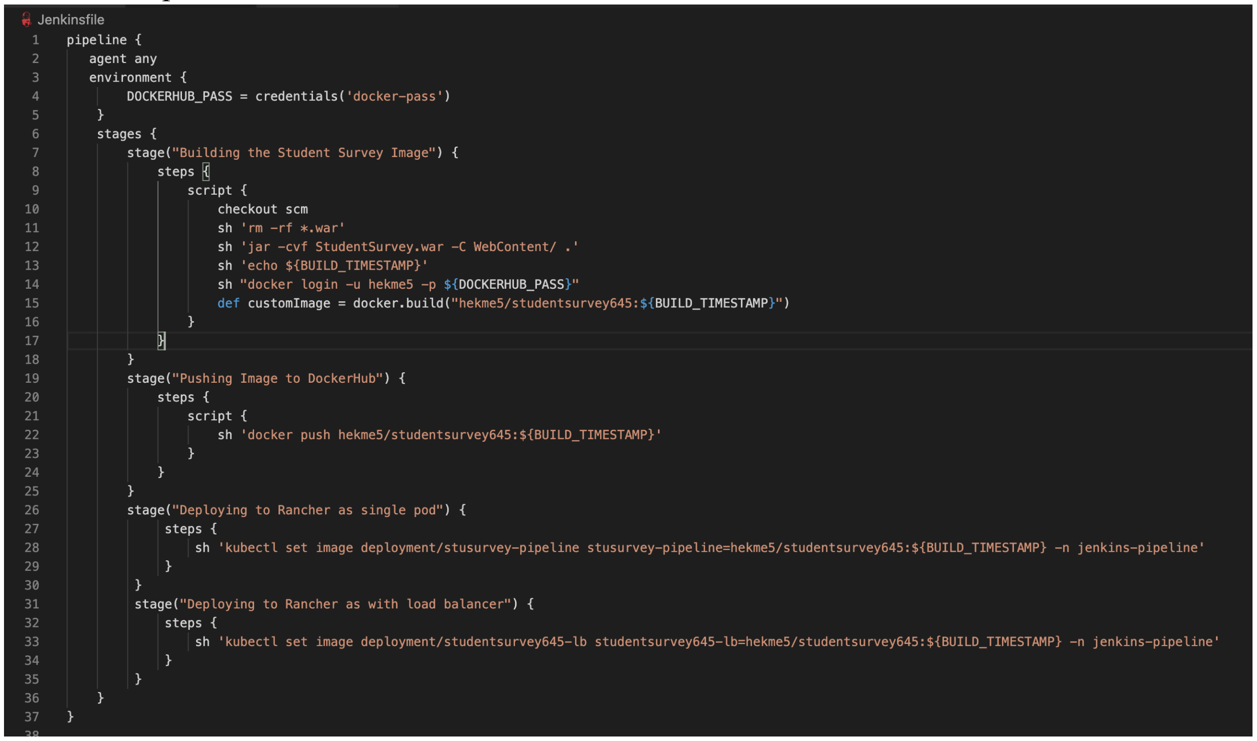
Script Path: Jenkinsfile

Lightweight checkout ☑

e. Leave all the rest of the settings as default and Save.

7. Setting up the jenkinsFile
A jenkinsFile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control.

a. In Jenkins, click on configure and manage plugins. Install the BitBucket plugin, docker plugins and the build time stamp plugin.
b. The DockerHub password will be passed as an environmental variable to the jenkinsFile and so will the timestamp. These are configured under 'Manage Jenkins'
c. Back to Eclipse, create a Jenkins file.



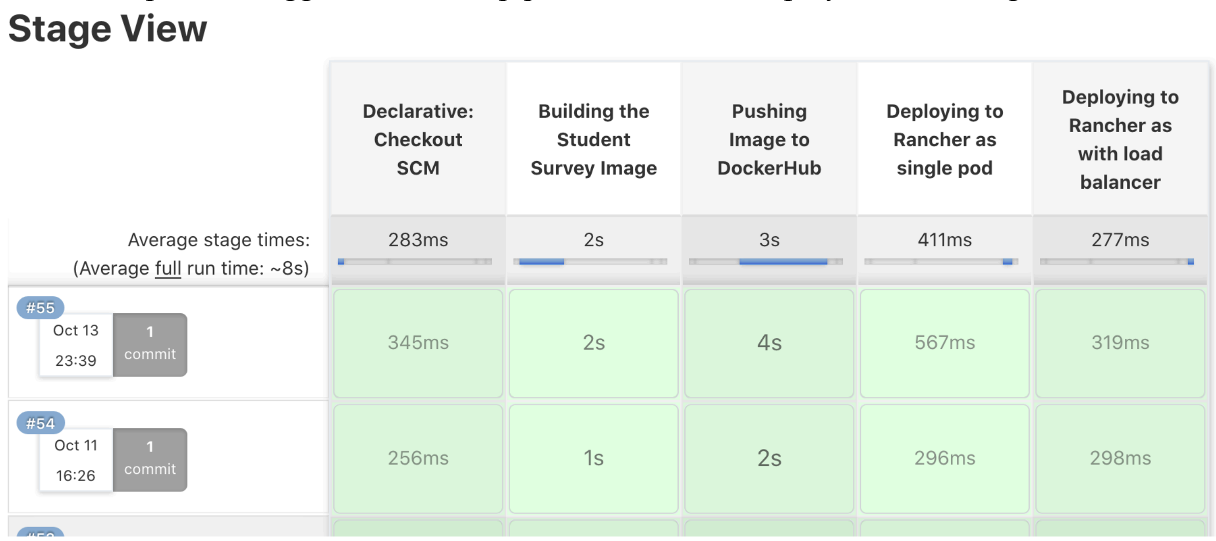
Jenkinsfile

```
pipeline {
   agent any
   environment {
        DOCKERHUB_PASS = credentials('docker-pass')
    }
    stages {
        stage("Building the Student Survey Image") {
            steps {
                script {
                    checkout scm
                    sh 'rm -rf *.war'
                    sh 'jar -cvf StudentSurvey.war -C WebContent/ .'
                    sh 'echo ${BUILD_TIMESTAMP}'
                    sh "docker login -u hekme5 -p ${DOCKERHUB_PASS}"
                    def customImage = docker.build("hekme5/studentsurvey645:${BUILD_TIMESTAMP}")
                }
            }
        }
        stage("Pushing Image to DockerHub") {
            steps {
                script {
                    sh 'docker push hekme5/studentsurvey645:${BUILD_TIMESTAMP}'
                }
            }
        }
        stage("Deploying to Rancher as single pod") {
             steps {
                sh 'kubectl set image deployment/stusurvey-pipeline stusurvey-pipeline=hekme5/studentsurvey645:${BUILD_TIMESTAMP} -n jenkins-pipeline'
             }
         }
         stage("Deploying to Rancher as with load balancer") {
             steps {
                sh 'kubectl set image deployment/studentsurvey645-lb studentsurvey645-lb=hekme5/studentsurvey645:${BUILD_TIMESTAMP} -n jenkins-pipeline'
             }
         }
    }
}
```

d. The first step is getting the pull on Bitbucket repository, cleaning up and running the command 'jar' to create a war file. It will then login to dockerHub and re-tag (rename) the image with the time stamp of YYYYMMDD-HMMSS.
e. The next step, it will push the image to docker hub with its new time stamp. You can verify that the image made it to docker Hub
f. In next two stages, we are using the kubectl command to reset the images of my deployments to the new images. It is using the 'kubectl set image' command.

8. Running the jenkinsFile
We are ready to run the pipeline in Jenkins to deploy in Rancher. Making a quick change to the

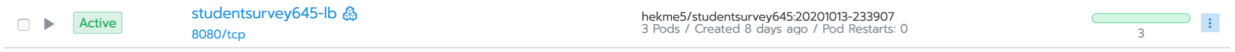
html on Eclipse will trigger the Jenkins pipeline which will deploy the new image to Rancher.



Stage View

	Declarative: Checkout SCM	Building the Student Survey Image	Pushing Image to DockerHub	Deploying to Rancher as single pod	Deploying to Rancher as with load balancer
Average stage times: (Average full run time: ~8s)	283ms	2s	3s	411ms	277ms
#55 Oct 13 23:39 1 commit	345ms	2s	4s	567ms	319ms
#54 Oct 11 16:26 1 commit	256ms	1s	2s	296ms	298ms

In Rancher, we can see that the image have been updated on the deployment.



Active	studentsurvey645-lb 8080/tcp	hekme5/studentsurvey645:20201013-233907 3 Pods / Created 8 days ago / Pod Restarts: 0	3	⋮

As you can see, the build date in Jenkins matches the tag name in Rancher.

References:
https://medium.com/@pra4mesh/deploy-war-in-docker-tomcat-container-b52a3baea448
https://linuxize.com/post/how-to-install-jenkins-on-ubuntu-20-04/
https://www.jenkins.io/doc/book/pipeline/docker/