Complete any one of the project to get the certification

Project 1 or Project 2

Steps to complete the Course end Project 1:

To provide the solution with screenshots

1. Provide the screenshot of your code repository in GIThub

Github – you will using the repository : <https://github.com/Sonal0409/DevOpsCodeDemo.git>

1. You will provide screenshot of Building the code using pipeline as Code in Jenkins

Create a pipeline with maven, java in Jenkins. You will create stages:

* Clone the repo
* Compile
* Test
* Package

1. You will also provide us screenshots of the console output of package job

And target folder with addressbook.war file

1. You will create a new job in Jenkins-> Name it as deploy and add the script to write dockerfile

And build the dockerfile into an image and run the image to create a container

You will provide us screenshots of Docker image & docker container up and running

You can also share screenshot of application container accessed through browser.

|  |
| --- |
|  |
| 3. Log into jenkins --> go to package job --go to console and copy the path of .war file |
|  |  |
|  | 4. create a new job -->Deploy |
|  |  |
|  | 5. Go to build -->shell command |
|  |  |
|  | rm -rf mydockerfile |
|  | mkdir mydockerfile |
|  | cd mydockerfile |
|  | cp /var/lib/jenkins/workspace/Package/target/addressbook.war . |
|  | touch dockerfile |
|  | cat <<EOT>> dockerfile |
|  | From tomcat |
|  | ADD addressbook.war /usr/local/tomcat/webapps |
|  | EXPOSE 8080 |
|  | CMD ["catalina.sh", "run"] |
|  | EOT |
|  | sudo docker build -t myimage:$BUILD\_NUMBER . |
|  | sudo docker run -itd -P myimage:$BUILD\_NUMBER |
|  |  |
|  | Save the job |
|  |  |
|  | Go to instance to give jenkins permission to execute docker commands. As of now we are logged in as admin in jenkins and it doesnt have permission to run docker commands. |
|  |  |
|  | So go to |
|  | vim /etc/sudoers |
|  | I |
|  | add under root |
|  | Jenkins ALL=NOPASSWD: ALL |
|  | :wq! |
|  |  |
|  | Go back to jenkins and build now. |
|  |  |
|  | ==> New image and container will be created |
|  |  |
|  | # docker images |
|  |  |
|  | # docker ps -a |
|  |  |
|  | Access from browser |
|  |  |
|  | http://18.221.206.57:32768/addressbook/ |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |

[Optional]

1. Go to Ansible Controller machine which is connect to the host

* Execute the ping module and give screenshot to show us it is connected over ssh to Hosts
* Write a playbook :
  + Install git & docker on the host machine
  + Clone the repo : <https://github.com/Sonal0409/AnsibleDockerdemo.git>
  + Build the docker file
  + Run the container
* Share screenshot of execution of playbook and container creation.

|  |
| --- |
| --- |
|  | - hosts: webservers |
|  | become: true |
|  | become\_user: root |
|  | tasks: |
|  | - name: install tomcat |
|  | yum: name=tomcat state=present |
|  | - name: start tomcat |
|  | service: name=tomcat state=started |
|  | - name: install docker |
|  | yum: name=docker state=present |
|  | - name: start docker |
|  | service: name=docker state=started |
|  | - name: install git |
|  | yum: name=git state=present |
|  | - name: clone a repo |
|  | git: repo=https://github.com/Sonal0409/AnsibleDockerdemo.git dest=/tmp/mygitrepo1 |
|  | - name: build docker file |
|  | command: chdir=/tmp/mygitrepo1 docker build -t myadd:ansible2 . |
|  | - name: create container |
|  | command: docker run -itd -P myadd:ansible2 |

OR

1. You will Connect Jenkins to you Ansible Controller Machine(use Master & slave)
2. You can create a job, using Execute shell you will write the script

to change the directory /etc/ansible

And give the command to execute the playbook

The playbook should have steps :

* Install git & docker on the host machine
* Clone the repo : <https://github.com/Sonal0409/AnsibleDockerdemo.git>
* Build the docker file
* Run the container

1. You will give screenshot of playbook , Console output of jenkins and container created on host machine

Steps for execute project 2:

Execute the same steps till docker as mentioned above

Once image is available, you can push the image to dockerhub

Then you will take the Kubernetes cluster (GKE or Lab)

You will create replicas of the image that was created in docker step

* You will screenshot of pods created

If you want you can also implement HPA to monitor and autoscale your pods

You will share screenshots of :

* HPA YAML
* Autoscaling pod

You will then deploy Prometheus POD and give screenshot with some Prometheus queries

You will also set up Grafana and give us the screenshot of Kubernetes dashboard