|  |  |
| --- | --- |
| **DOCUMENT RULES:** | |
| **Task Number / Name:** | **DevOPS Scenarios** |
| **Task name & column name should be written:** | **Bold (CTRL+B)** |
| **Commands should be written in the after # sign:** | *Italic (CTRL+I) #hostname* |
| **Output photo should be cropped or compressed:**  **Photo could be more than one:**  **If you need extra lines, add the line next after it:** | ***Description photo should be with title bar (CTRL + I + B)*** |
| **All other text should be written:** | Standard |
| **Font name and text size:** | Calibri and 9 |
| **Group name:** | Dev\_ops\_ |
| **Student name and surname:** | Murad Abbaszade |
| **E-mail:** | [muradabbaszade6@gmail.com](mailto:muradabbaszade6@gmail.com) |
| **WhatsApp number:** | **+994703664205** |

|  |  |
| --- | --- |
| 1.Git-Test |  |
| Clone the below git repo https://github.com/sakha-devops/git-assignment.git  user\_name: sakha-devops  password: ghp\_Esdl2wsgxzmwcQt0YWS9qTKcWo8aHb26Y9lw  git clone https://github.com/sakha-devops/git-assignment.git |  |
| Create new gitbranch  git branch mybranch |  |
| Delete branch called mybranch  git checkout main  git branch –delete mybranch |  |
| Backup mybranch to mybackupbrach  git checkout –b mybackupmybranchbackup |  |
| git clone <https://github.com/sakha-devops/git-assignment.git>  cpapp.py ./git-assignment  git push origin main |  |
| 2.Docker |  |
| docker run -it -d --name=mycontainer alpine:3.16.0 |  |
| cp application.py /home  cp application.py /root/filesystem |  |
| docker run -it -d -p 5200:5200 --name=test1 alpine:latest |  |
| docker run -itd --name=test1 -p 8300:8300 node:alpine3.15 |  |
| docker run -itd --name=new-docker2 alpine3.16.0 |  |
| Echo new-docker2 > file.txt  Echo new-docker2 > file2.txt |  |
| Docker commit –m “commited-image” new-docker2 |  |
| 3.Docker with Nginx |  |
| 4.Docker-Nginx-mysql |  |
| 5.Jenkins |  |
| We will give GITHUB repo access, They are Dockerfile and Python Application. Dockerfile is used to create a Docker image  Pre-requisite:   1. Jenkins server(installed with port 8080) 2. Docker also installed on the same Jenkins server. 3. GIT installed on same server. |  |
| Clone the repo url https://github.com/sakha-devops/jenkins.git | Git clone https://github.com/sakha-devops/jenkins.git |
| Github: username: sakha-devops  password: ghp\_Esdl2wsgxzmwcQt0YWS9qTKcWo8aHb26Y9lw |  |
| 6.Kubernetes basics |  |
| Create a pod called helloworld |  |
| Nano helloworld.yaml  **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: nginx  **spec**:  **containers**:  - **name**: nginx  **image**: nginx:1.14.2  **ports**:  - **containerPort**: 80 |  |
| Kubectl create –f helloworld.yaml |  |
| Delete the pod called helloworld  Kubectl delete pod helloworld |  |
| 7.Kubernetes-expose-app |  |
| You can use the kubectl command or Write a YAML file to deploy the application  Use any one below sample docker images  Sample docker Images   1. nginx:latest 2. gcr.io/google-samples/hello-app:1.0 |  |
| Next, expose the deployed pod to the outside as a Kubernetes Service  specifying a static port where it will be accessible with type=NodePort and | nginx image use port 80  gcr image use port 8080 |
| 8.Kubernetes-expose-react-app |  |
| Kubernetes challenge  Deploy this application in kubernetes and customise the environment variable to display your name.  Output Should be get like this:  Hello {YOUR NAME!} Instructions: Fork this repo  GIR URL: https://github.com/sakha-devops/k8s-react.git |  |
| Build the Docker image  tag docker image and push the Docker image to a Docker registry NOTE: docker images should tag like this sakhadevopsdocker/application:{anything}  Write yaml files for a deployment, service yaml files  port number = 4000 |  |