

Name: Bharanidharan B [24MCR010]

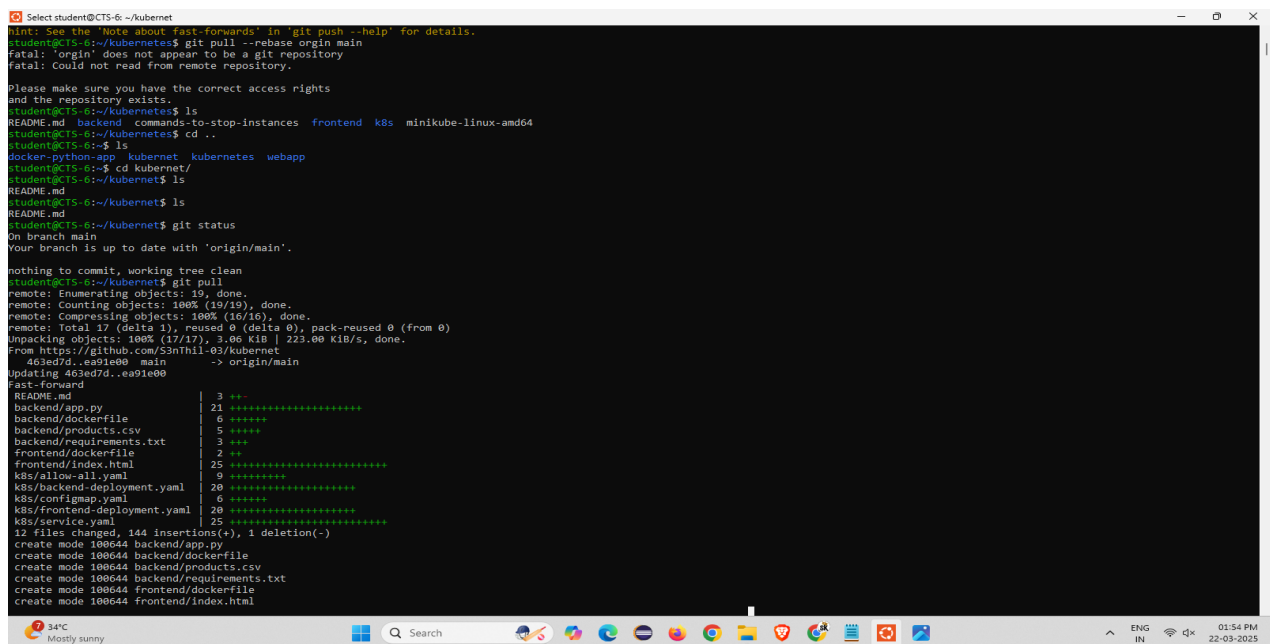
Class: I – MCA – “A”

# DEVOPS TRAINING

## DAY 5 CONFIGURING PIPELINE

Step 1:

Create github repository kubernetes and push the cd kubernetes files as frontend ,backend and k8s



```
Select student@CTS-6: ~/kubernetes
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
student@CTS-6:~/kubernetes$ git pull --rebase origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
student@CTS-6:~/kubernetes$ ls
README.md  backend  commands-to-stop-instances  frontend  k8s  minikube-linux-amd64
student@CTS-6:~/kubernetes$ cd ..
student@CTS-6:~$ ls
docker-python-app  kubernetes  kubernet  webapp
student@CTS-6:~$ cd kubernetes/
student@CTS-6:~/kubernetes$ ls
README.md
student@CTS-6:~/kubernetes$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
student@CTS-6:~/kubernetes$ git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 17 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (17/17), 3.06 KiB | 223.00 KiB/s, done.
From https://github.com/53nThil-03/kubernetes
 463ed7d..ea91e00  main    -> origin/main
Updating 463ed7d..ea91e00
Fast-forward
 README.md          | 3 +++
 backend/app.py     | 21 ++++++
 backend/dockerfile | 6 +++++
 backend/products.csv | 5 +++++
 backend/requirements.txt | 3 +++
 frontend/dockerfile | 2 ++
 frontend/index.html | 25 ++++++
 k8s/allow-all.yaml | 9 ++++++
 k8s/backend-deployment.yaml | 20 ++++++
 k8s/configmap.yaml | 6 ++++++
 k8s/frontend-deployment.yaml | 20 ++++++
 k8s/service.yaml | 25 ++++++
 12 files changed, 144 insertions(+), 1 deletion(-)
create mode 100644 backend/app.py
create mode 100644 backend/dockerfile
create mode 100644 backend/products.csv
create mode 100644 backend/requirements.txt
create mode 100644 frontend/dockerfile
create mode 100644 frontend/index.html
```

Step 2:

create Jenkinsfile and add the following code and push into the github repo kubernetes

```
Select student@CTS-6: ~/Kubernetes
id https://github.com/S3nThil-03/kubernetes
$ curl -s https://raw.githubusercontent.com/S3nThil-03/kubernetes/main/.devcontainer/devcontainer.json | jq -r '.pipeline'
{
  "name": "Kubernetes",
  "type": "pipeline",
  "steps": [
    {
      "name": "Checkout Code",
      "steps": [
        {
          "name": "Checkout Code",
          "command": "git clone https://github.com/S3nThil-03/kubernetes.git"
        }
      ]
    },
    {
      "name": "Build Backend Docker Image",
      "steps": [
        {
          "name": "Build Backend Docker Image",
          "command": "docker build -t s3nthil03/docker-backend:latest -f backend/Dockerfile ."
        }
      ]
    },
    {
      "name": "Build Frontend Docker Image",
      "steps": [
        {
          "name": "Build Frontend Docker Image",
          "command": "docker build -t s3nthil03/docker-frontend:latest -f frontend/Dockerfile ."
        }
      ]
    },
    {
      "name": "Login to Docker Registry",
      "steps": [
        {
          "name": "Login to Docker Registry",
          "command": "docker login --username s3nthil03 --password $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 32 | tr -d '\n' | xargs echo)"
        }
      ]
    },
    {
      "name": "Push Docker Images",
      "steps": [
        {
          "name": "Push Docker Images",
          "command": "docker push s3nthil03/docker-backend:latest"
        },
        {
          "name": "Push Docker Images",
          "command": "docker push s3nthil03/docker-frontend:latest"
        }
      ]
    },
    {
      "name": "Stop & Remove Existing Containers",
      "steps": [
        {
          "name": "Stop & Remove Existing Containers",
          "command": "docker rm -f $(docker ps -q)"
        }
      ]
    }
  ]
}
```

### Step 3:

Open Jenkins create a item in pipeline and click ok and go to configure add the repo url and credentials and click build

The screenshot shows the Jenkins job configuration page for a job named 'test3'. The 'Pipeline' tab is selected, showing a pipeline script. The 'Build' button is visible. The 'Pipeline' section shows the following steps:

- git** (Revision: ea2bd4df80546e6fa4c3a06d45833d1506f299f1, Repository: https://github.com/S3nThil-03/kubernetes.git)
- git** (Revision: ea2bd4df80546e6fa4c3a06d45833d1506f299f1, Repository: https://github.com/S3nThil-03/kubernetes.git)

The 'Pipeline' section also shows a warning icon and the message: "The following steps that have been detected may have insecure interpolation of sensitive variables (click here for an explanation): (in progress)".

### Step 4 :

Open console output and check build is complete or not.

The screenshot shows the Jenkins 'Console Output' for a pipeline named 'test3' (build #9). The output details the process of obtaining the Jenkinsfile from a GitHub repository, checking out the code, and running a series of Git commands to fetch and checkout the latest version of the Kubernetes installation script. The pipeline status is 'Completed' with a green checkmark.

```
Started by user Senthilkumar
Obtained Jenkinsfile from git https://github.com/S3nThil-03/kubernetes.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/test3
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-senthil
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/test3/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/S3nThil-03/kubernetes.git # timeout=10
Fetching upstream changes from https://github.com/S3nThil-03/kubernetes.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/S3nThil-03/kubernetes.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision ea2bd4df80546e6fa4c3a06d45833d1506f299f1 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f ea2bd4df80546e6fa4c3a06d45833d1506f299f1 # timeout=10
```

## Step 5 :

Go to dashboard > manage Jenkins > plugins and install the Kubernetes once it all download success will shown.

The screenshot shows the 'Manage Jenkins > Plugins' page. On the left, the 'Download progress' tab is selected. The main area shows a list of plugins being installed, all with a 'Success' status. A red notification bubble indicates that updates are available for some plugins.

Plugin	Status
Kubernetes Client API	Success
Authentication Tokens API	Success
Kubernetes Credentials	Success
Kubernetes	Success
Loading plugin extensions	Success

→ [Go back to the top page](#)  
(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running