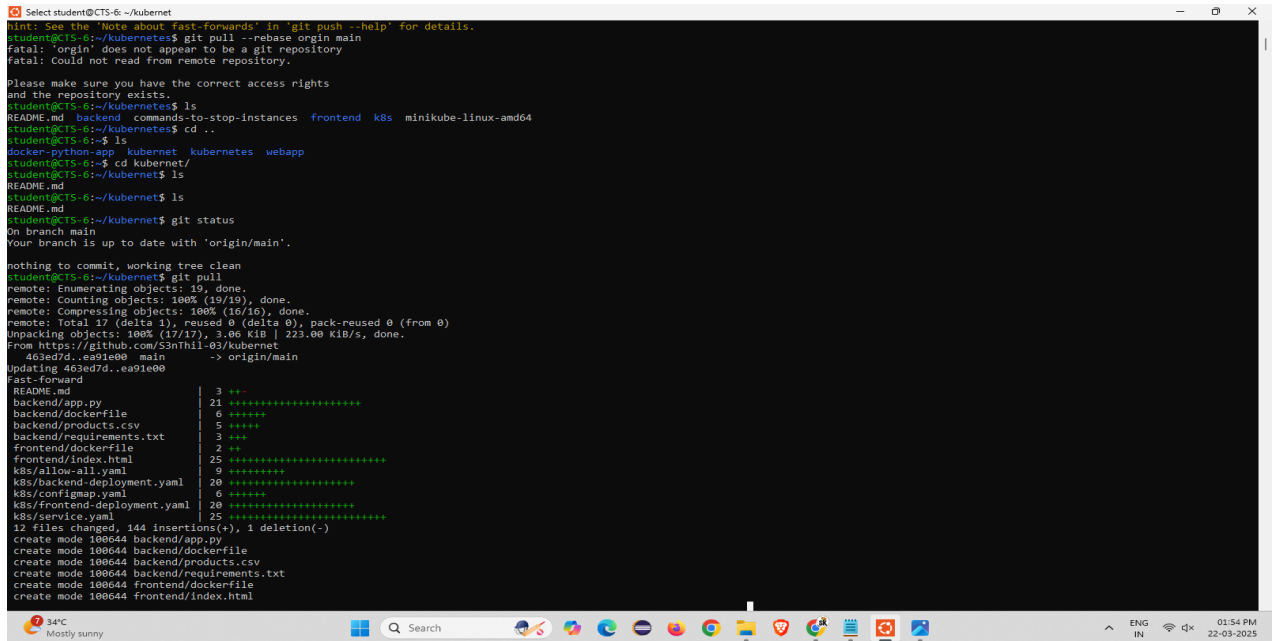


DEVOPS TRAINING

DAY 5 CONFIGURING PIPELINE

Step 1: Create github repository kubernetes and push the cd kubernetes files as frontend ,backend and k8s

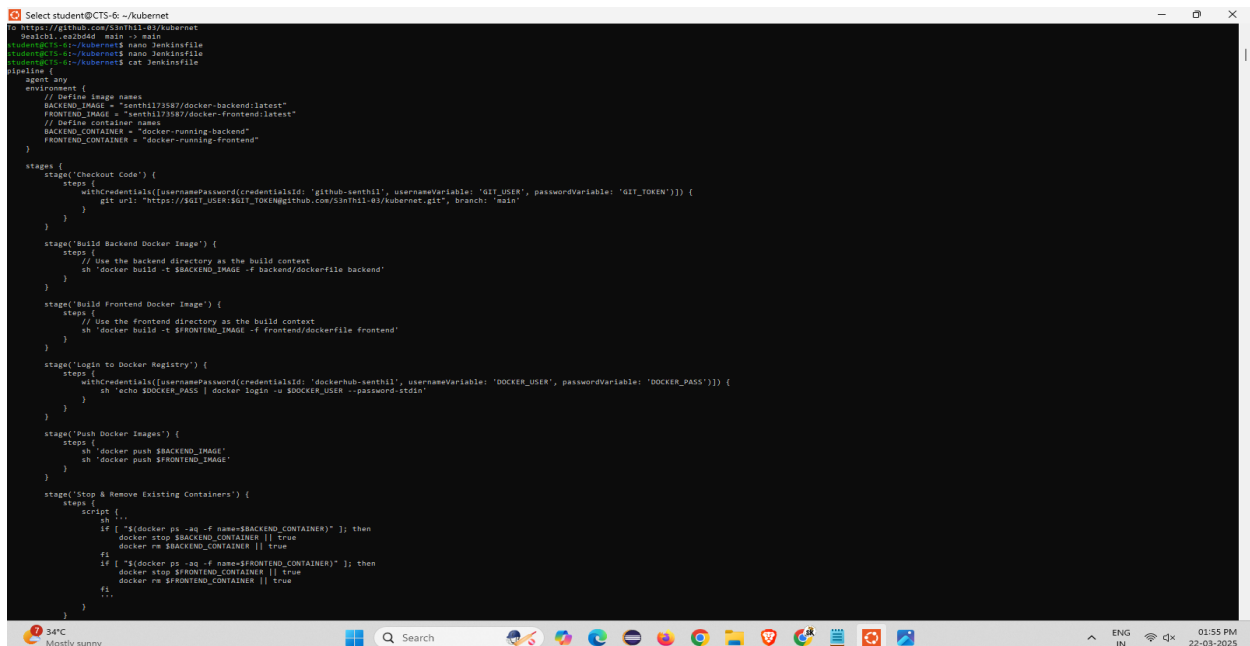


```
Select student@CTS-6: ~/kubernetes
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
student@CTS-6:~/kubernetes$ git pull --rebase origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

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$ ls
README.md
student@CTS-6:~/kubernetes$ git status
On branch main
Your branch is up to date with 'origin/main'.

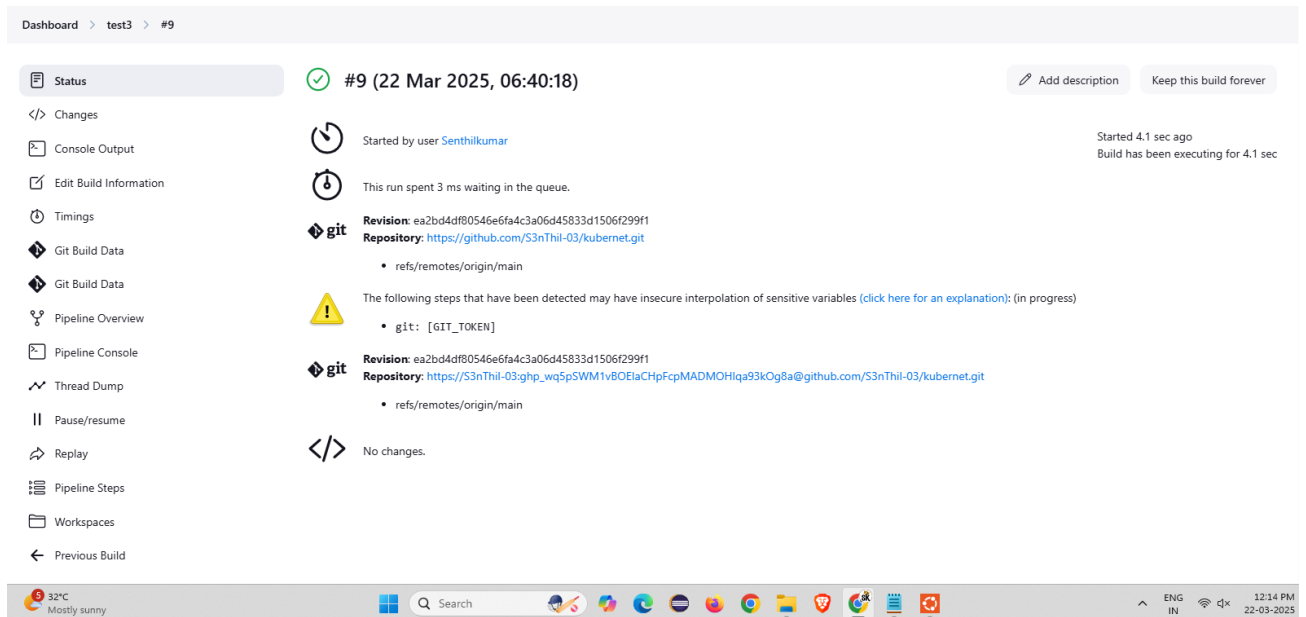
nothing to commit, working tree clean
student@CTS-6:~/kubernetes$ git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 17 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (17/17), 3.06 KiB | 223.00 KiB/s, done.
From https://github.com/S3nTh1l-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



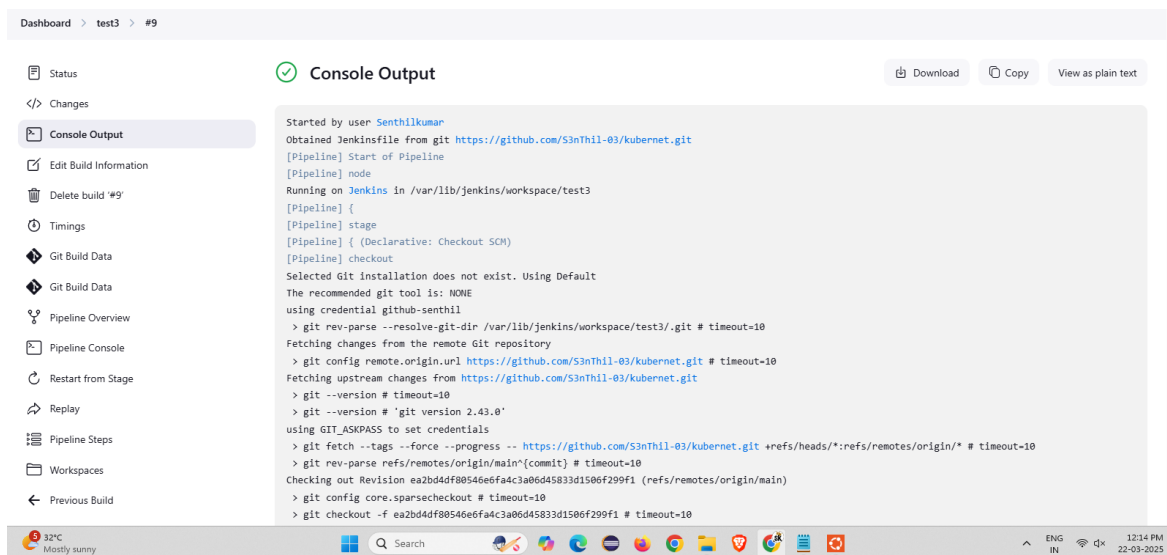
```
Select student@CTS-6: ~/kubernetes
In https://github.com/S3nTh1l-03/kubernetes
9ea1cbl..ea2b0d4  main -> main
student@CTS-6:~/kubernetes$ nano Jenkinsfile
student@CTS-6:~/kubernetes$ cat Jenkinsfile
pipeline {
  agent any
  environment {
    // Define image names
    BACKEND_IMAGE = "senthil73587/docker-backend:latest"
    FRONTEND_IMAGE = "senthil73587/docker-frontend:latest"
    // Define container names
    BACKEND_CONTAINER = "docker-running-backend"
    FRONTEND_CONTAINER = "docker-running-frontend"
  }
  stages {
    stage('Checkout Code') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'github-senthil', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {
          git url: 'https://$GIT_USER:$GIT_TOKEN@github.com:S3nTh1l-03/kubernetes.git', branch: 'main'
        }
      }
    }
    stage('Build Backend Docker Image') {
      steps {
        // Use the backend directory as the build context
        sh "docker build -t $BACKEND_IMAGE -f backend/dockerfile backend"
      }
    }
    stage('Build Frontend Docker Image') {
      steps {
        // Use the frontend directory as the build context
        sh "docker build -t $FRONTEND_IMAGE -f frontend/dockerfile frontend"
      }
    }
    stage('Login to Docker Registry') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'dockerhub-senthil', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
          sh "echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin"
        }
      }
    }
    stage('Push Docker Images') {
      steps {
        sh "docker push $BACKEND_IMAGE"
        sh "docker push $FRONTEND_IMAGE"
      }
    }
    stage('Stop & Remove Existing Containers') {
      steps {
        script {
          sh "if [ $(docker ps -q -f name=$BACKEND_CONTAINER) ]; then
            docker stop $BACKEND_CONTAINER || true
            docker rm $BACKEND_CONTAINER || true
          fi
          sh "if [ $(docker ps -q -f name=$FRONTEND_CONTAINER) ]; then
            docker stop $FRONTEND_CONTAINER || true
            docker rm $FRONTEND_CONTAINER || true
          fi
        }
      }
    }
  }
}
```

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



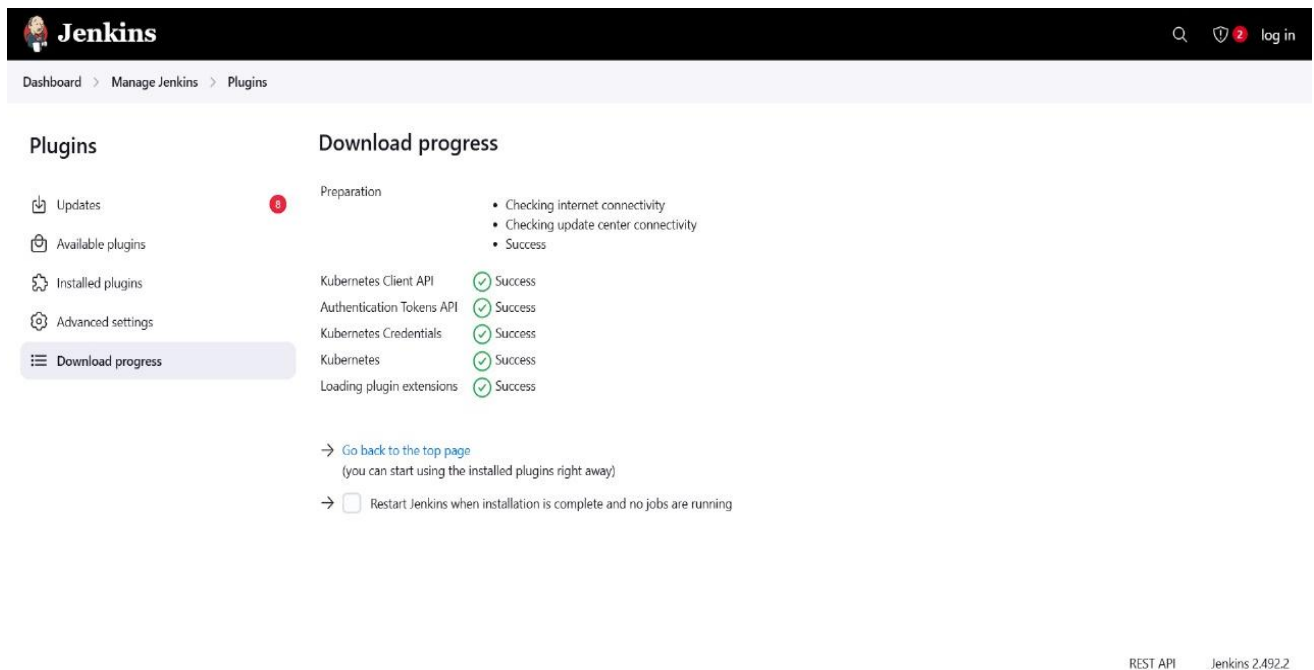
The screenshot shows the Jenkins build status page for build #9. The left sidebar contains navigation links: Status, Changes, Console Output, Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, Replay, Pipeline Steps, Workspaces, and Previous Build. The main area displays build details for #9 (22 Mar 2025, 06:40:18). It indicates the build started by user Senthilkumar, waited 3 ms in the queue, and started 4.1 sec ago. The build is currently in progress. The repository is https://github.com/S3nThil-03/kubernetes.git, revision ea2bd4df80546e6fa4c3a06d45833d1506f299f1. A warning icon indicates insecure interpolation of sensitive variables (GIT_TOKEN) in progress. The console output shows 'No changes.'

Step 4 : open console output and check build is complete or not.



The screenshot shows the Jenkins build console output for build #9. The left sidebar is the same as in the previous screenshot. The main area displays the console output, which shows the build process details: Started by user Senthilkumar, Obtained Jenkinsfile from 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/{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 Jenkins web interface. At the top is a black header with the Jenkins logo and name on the left, and search, shield, and 'log in' links on the right. Below the header is a breadcrumb trail: 'Dashboard > Manage Jenkins > Plugins'. The main content area is divided into two columns. The left column, titled 'Plugins', contains a sidebar with links: 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (which is highlighted). The right column, titled 'Download progress', shows the status of the plugin installation. It starts with a 'Preparation' section with a red circle icon and a list of steps: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this is a list of plugins being installed, each with a green checkmark and the word 'Success': 'Kubernetes Client API', 'Authentication Tokens API', 'Kubernetes Credentials', 'Kubernetes', and 'Loading plugin extensions'. At the bottom of the right column, there are two links: 'Go back to the top page (you can start using the installed plugins right away)' and 'Restart Jenkins when installation is complete and no jobs are running' (with an unchecked checkbox).

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress**

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

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

REST API Jenkins 2.492.2