

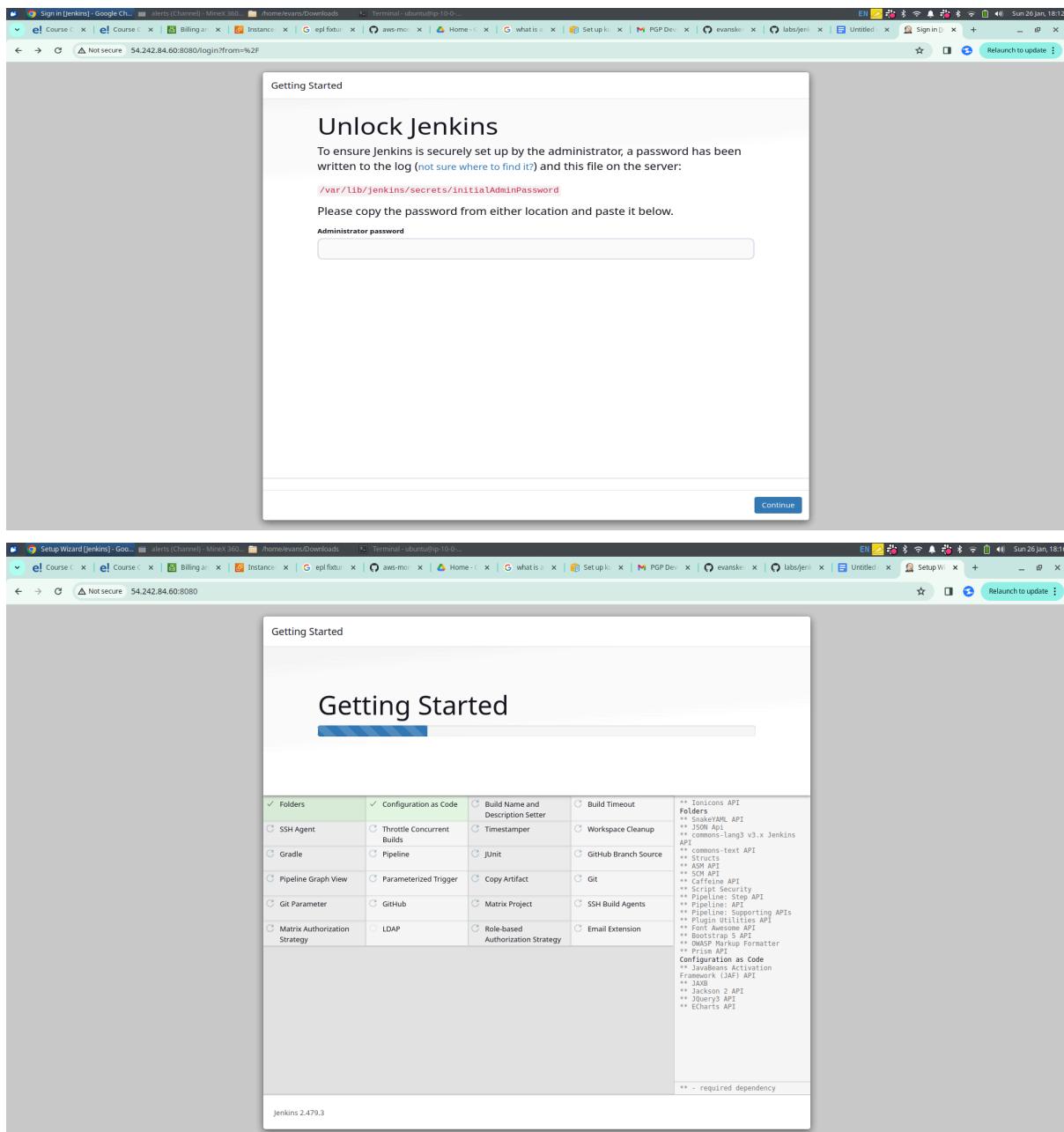
PGP Final Project

Source Code.

My github link: <https://github.com/evanskenney/final-devops-project>. My dockerfile and playbook used can be found at the link provided above.

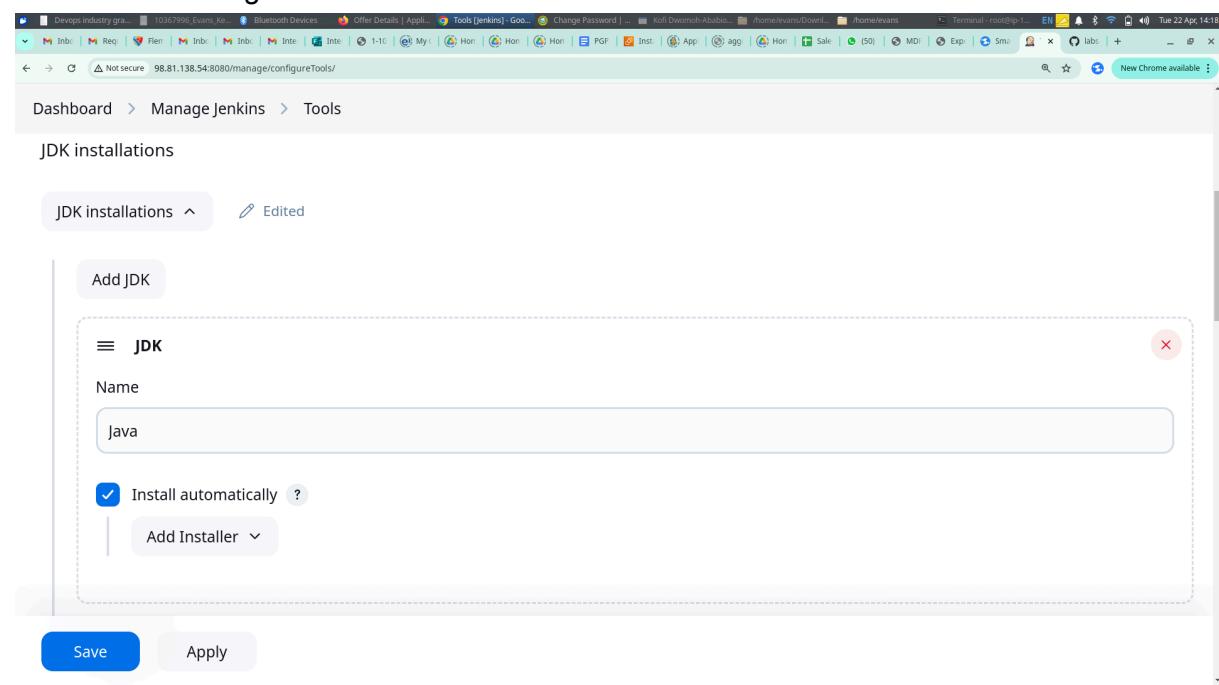
Setting up jenkins.

I followed <https://github.com/evanskenney/labs/tree/master/jenkins/Install> to get jenkins installed.



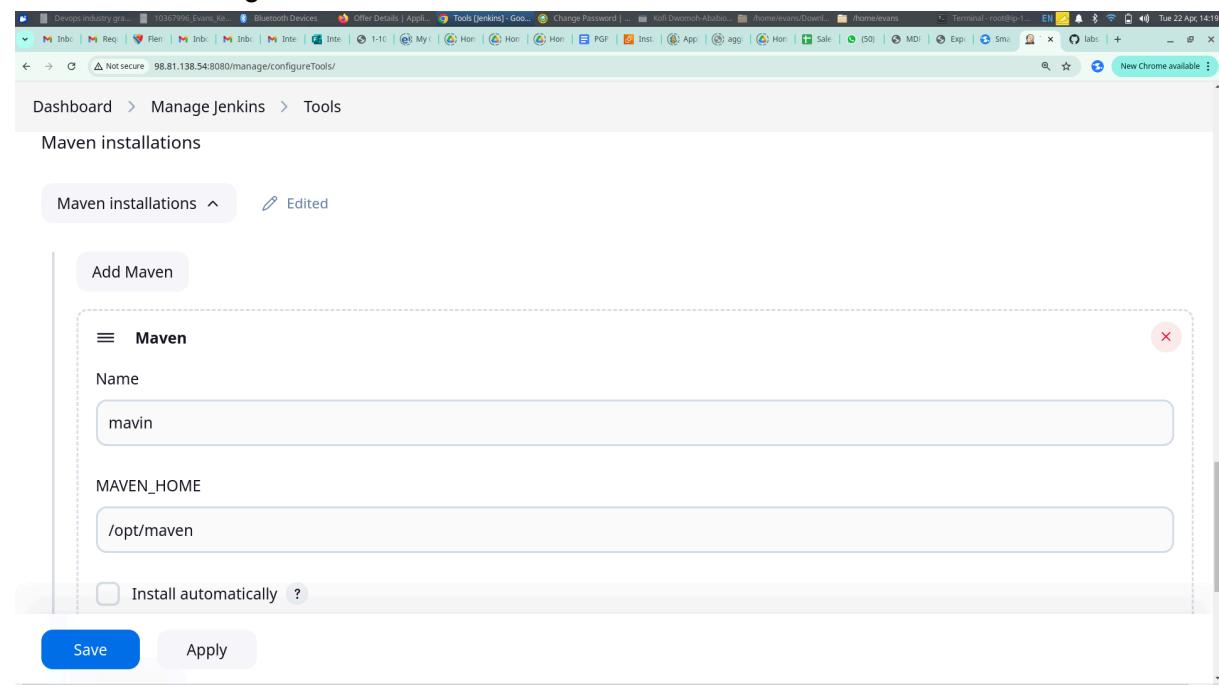
Setting up Java and Maven Home

Tool location configuration for Java



The screenshot shows the 'JDK installations' configuration page in Jenkins. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins > Tools. Below that, it says 'JDK installations'. A sub-breadcrumb 'JDK installations ^' is shown with an 'Edited' status indicator. A large 'Add JDK' button is visible. A modal dialog box titled 'JDK' is open, containing fields for 'Name' (set to 'Java') and a checkbox for 'Install automatically' which is checked. Below the dialog are 'Save' and 'Apply' buttons.

Tool location configuration for Maven



The screenshot shows the 'Maven installations' configuration page in Jenkins. The breadcrumb navigation is identical to the previous page: Dashboard > Manage Jenkins > Tools. Below that, it says 'Maven installations'. A sub-breadcrumb 'Maven installations ^' is shown with an 'Edited' status indicator. A large 'Add Maven' button is visible. A modal dialog box titled 'Maven' is open, containing fields for 'Name' (set to 'mavin'), 'MAVEN_HOME' (set to '/opt/maven'), and a checkbox for 'Install automatically' which is unchecked. Below the dialog are 'Save' and 'Apply' buttons.

Completed Setup for Jenkins. Free style jobs were created for testing purposes. The final-project is a pipeline job which has all the configurations for every stage.

The screenshot shows the Jenkins dashboard with several build items listed:

S	W	Name	Last Success	Last Failure	Last Duration
✓	cloud	code compile	2 mo 22 days #9	2 mo 23 days #5	4 sec
✓	cloud	containerize	2 mo 21 days #7	2 mo 21 days #5	2.8 sec
✓	sun	final-project	24 days #35	N/A	37 sec
✓	sun	package	2 mo 21 days #3	N/A	6.5 sec
✓	sun	PWD	2 mo 21 days #1	N/A	14 ms
✓	cloud	Unit Test	2 mo 22 days #4	2 mo 22 days #2	9.2 sec

Build Queue: No builds in the queue.

Build Executor Status: Built-In Node (ubuntu) (offline)

Icon: S M L

REST API Jenkins 2.479.3

Task 1: Clone the project from the GitHub link

Console Output

Cloning

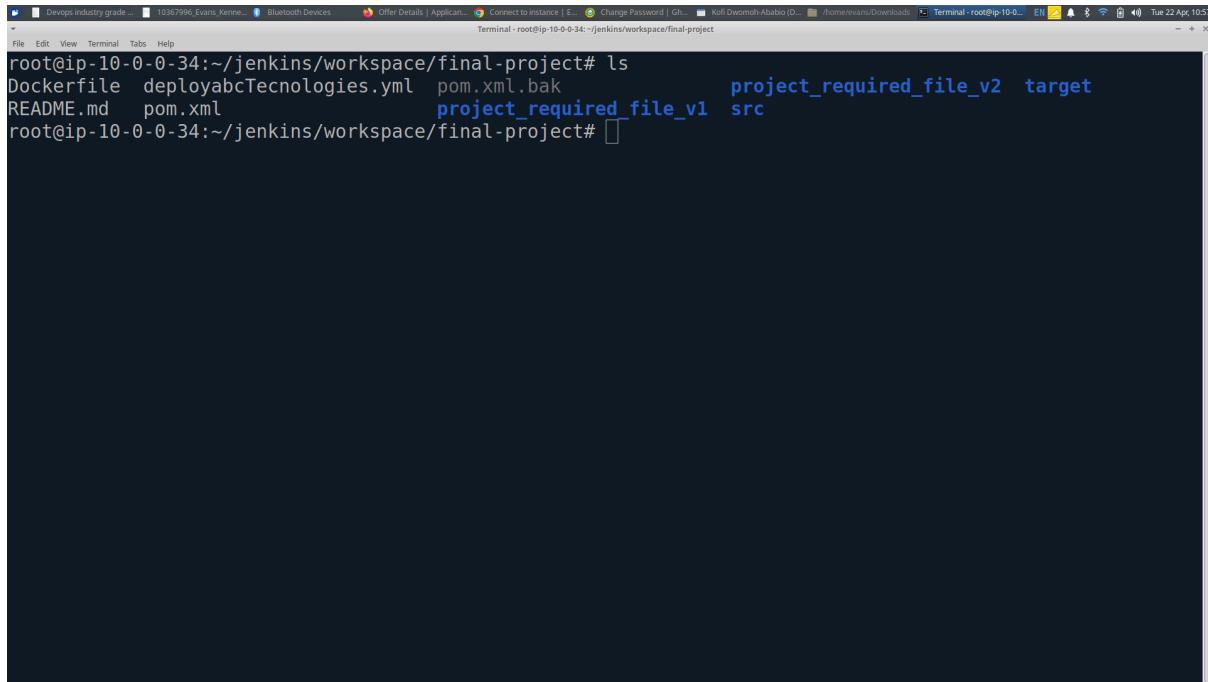
The screenshot shows the Jenkins Pipeline console output for build #35. The output logs show the cloning of a GitHub repository:

```

Started by user unknown or anonymous
[Pipeline] Start of Pipeline
[Pipeline] node
Running on ubuntu in /root/jenkins/workspace/final-project
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Compile)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
  compiling...
[Pipeline] git
  The recommended git tool is: NONE
  No credentials specified
  Fetching changes from the remote Git repository
  Checking out Revision cd15bd456e77a4a700b505a1369d53alea46599 (refs/remotes/origin/main)
  Commit message: "updating playbook"
  > git rev-parse --resolve-git-dir /root/jenkins/workspace/final-project/.git # timeout=10
  > git config remote.origin.url https://github.com/evanskenney/final-devops-project/ # timeout=10
  Fetching upstream changes from https://github.com/evanskenney/final-devops-project/
  > git -> version # 'git version 2.43.0'
  > git fetch --tags --progress -- https://github.com/evanskenney/final-devops-project/ +refs/heads/*:refs/remotes/origin/* # timeout=10
  > git config core.sparsecheckout # timeout=10
  > git checkout -f cd15bd456e77a4a700b505a1369d53alea46599 # timeout=10
  > git branch -a -v --no-abbr # timeout=10
  > git branch -D main # timeout=10
  > git push --force-with-new-sha main # timeout=10

```

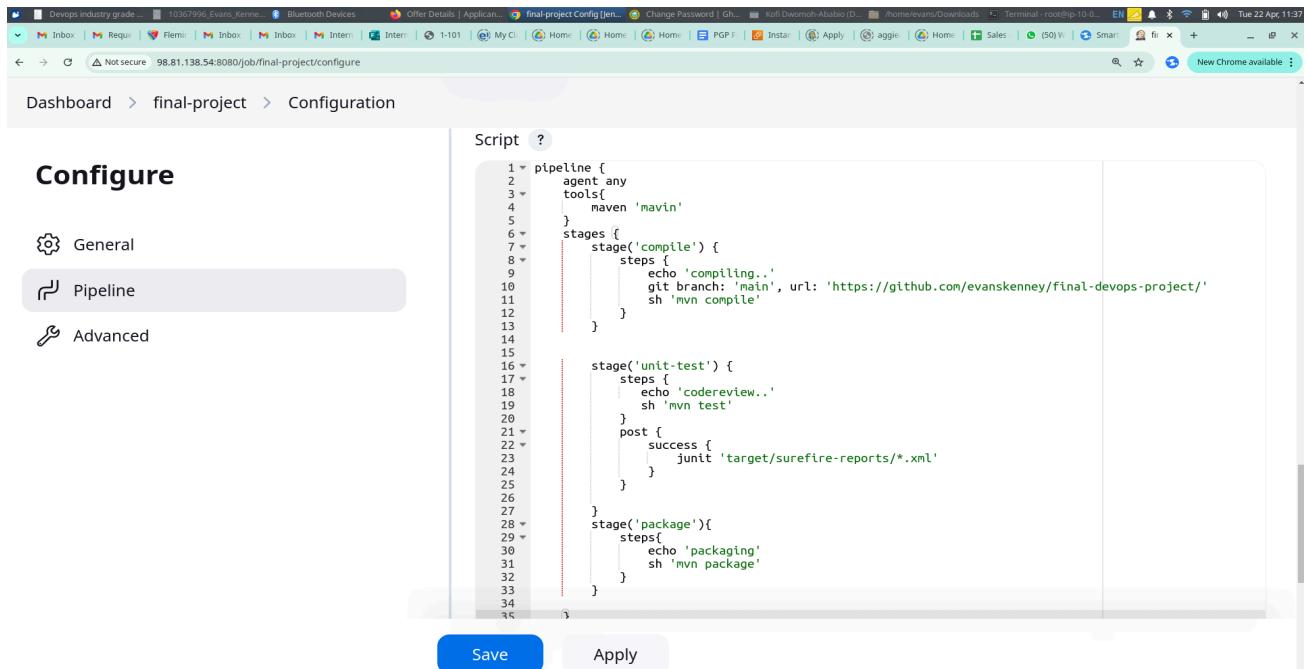
Project cloned on the jenkins worker node.



```
root@ip-10-0-0-34:~/jenkins/workspace/final-project# ls
Dockerfile  deployabcTecnologies.yml  pom.xml.bak  project_required_file_v2  target
README.md   pom.xml                  project_required_file_v1  src
root@ip-10-0-0-34:~/jenkins/workspace/final-project#
```

Task 2: Create a continuous integration pipeline using Jenkins to compile, test, and package the code present in GitHub.

Step 1: Create a build pipeline containing a job for each compiling, testing and packaging code



The screenshot shows the Jenkins Pipeline configuration page for a job named "final-project". The left sidebar has tabs for General, Pipeline (which is selected), and Advanced. The main area contains a "Script" section with a code editor showing Groovy script for the pipeline. The script defines a pipeline with stages for compilation, unit testing, and packaging.

```
1 pipeline {
2   agent any
3   tools{
4     maven 'maven'
5   }
6   stages {
7     stage('compile') {
8       steps {
9         echo 'compiling..'
10        git branch: 'main', url: 'https://github.com/evanskenney/final-devops-project/'
11      }
12    }
13    stage('unit-test') {
14      steps {
15        echo 'codereview..'
16        sh 'mvn test'
17      }
18      post {
19        success {
20          junit 'target/surefire-reports/*.xml'
21        }
22      }
23    }
24    stage('package'){
25      steps{
26        echo 'packaging'
27        sh 'mvn package'
28      }
29    }
30  }
31 }
32 }
33 }
34 }
35 }
```

At the bottom of the script editor are two buttons: "Save" and "Apply".

Step 2: Running the pipeline to perform the above actions of Maven Compile, Test and Package

Console Output: Images have been arranged in an a sequential order from Maven compile to Maven Package

Maven Compile

The screenshot shows a Jenkins job console for build #35. The left sidebar contains links like 'Console Output' (which is selected), 'Edit Build Information', 'Delete build', 'Timings', 'Git Build Data', 'Test Result', 'Pipeline Overview', 'Pipeline Console', 'Restart from Stage', 'Replay', and 'Pipeline Steps'. The main area displays the Jenkins pipeline script and its execution logs.

```
</> Changes
Started by user unknown or anonymous
[Pipeline] Start of Pipeline
[Pipeline] node
Running on ubuntu in /root/jenkins/workspace/final-project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (compile)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
```

The screenshot shows a Jenkins job console for build #35. The left sidebar contains links like 'Workspaces' (which is selected). The main area displays the Jenkins pipeline script and its execution logs, specifically focusing on the git cloning process.

```
compiling..
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision cd15bd456e77a4a700b58b5a1369d53a1ea46599
(refs/remotes/origin/main)
Commit message: "updating playbook"
> git rev-parse --resolve-git-dir /root/jenkins/workspace/final-project/.git #
timeout=10
> git config remote.origin.url https://github.com/evanskenney/final-devops-
project/ # timeout=10
Fetching upstream changes from https://github.com/evanskenney/final-devops-
project/
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/evanskenney/final-
devops-project/ +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
> git config core.sparsecheckout # timeout=10
> git checkout -f cd15bd456e77a4a700b58b5a1369d53a1ea46599 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
```

```
> git branch -a -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main cd15bd456e77a4a700b58b5a1369d53a1ea46599 # timeout=10
> git rev-list --no-walk 070a6d14a10ea23a0957ac6ac8999ae0f3689612 # timeout=10
[Pipeline] sh
+ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.abc:ABCtechnologies >-----
[INFO] Building RetailModule 1.0
[INFO]   from pom.xml
[INFO] -----[ war ]-----
[INFO]
[INFO] --- jacoco:0.8.6:prepare-agent (jacoco-initialize) @ ABCtechnologies ---
[INFO] argLine set to -
javaagent:/root/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/root/jenkins/workspace/final-project/target/jacoco.exec
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ABCtechnologies ---
[INFO] skip non existing resourceDirectory /root/jenkins/workspace/final-project/src/main/resources
```

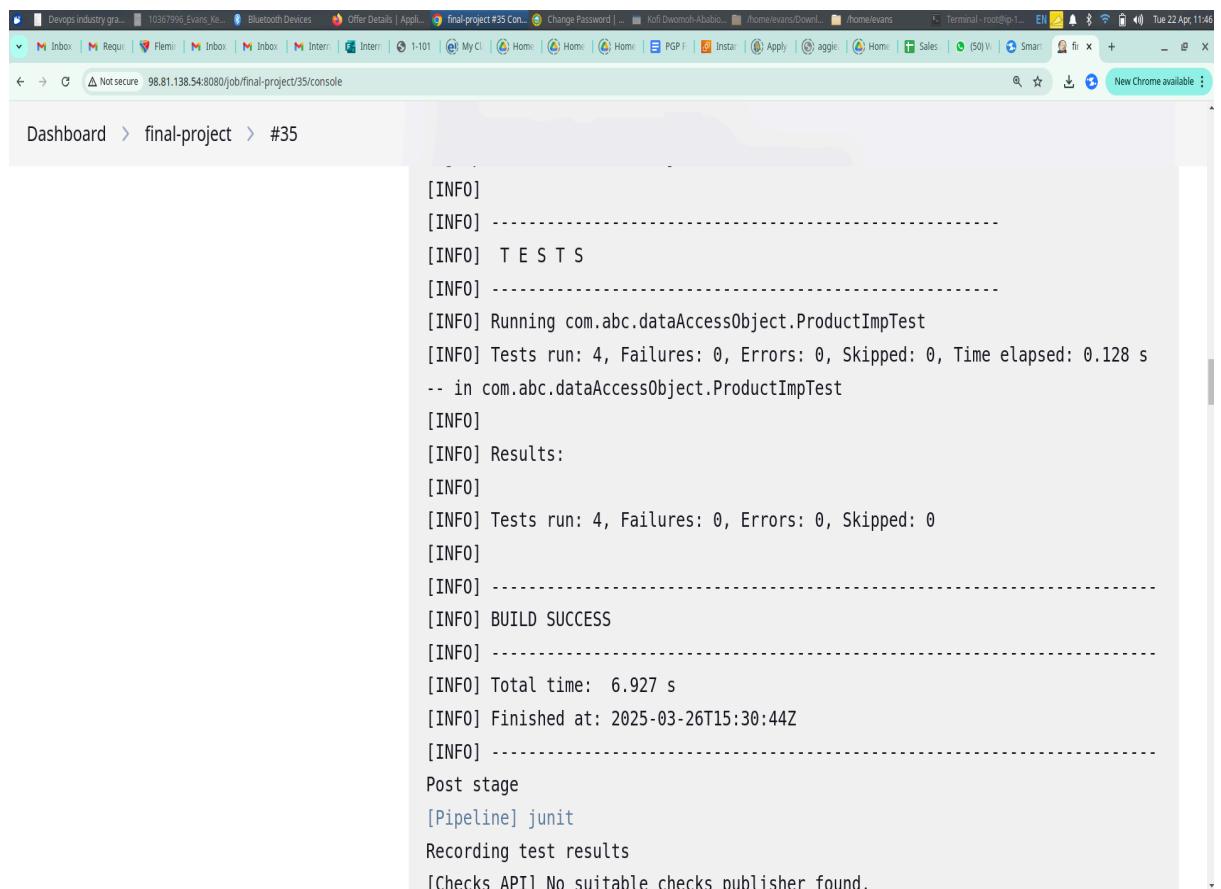
```
project/src/main/resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ ABCtechnologies ---
[INFO] Changes detected - recompiling the module! :input tree
[INFO] Compiling 3 source files with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.232 s
[INFO] Finished at: 2025-03-26T15:30:34Z
[INFO]
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (unit-test)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] }
```

Maven Test

```
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
codereview..
[Pipeline] sh
+ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.abc:ABCtechnologies >-----
[INFO] Building RetailModule 1.0
[INFO]   from pom.xml
[INFO] -----[ war ]-----
[INFO]
[INFO] --- jacoco:0.8.6:prepare-agent (jacoco-initialize) @ ABCtechnologies ---
[INFO] argline set to -
javaagent:/root/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/root/jenkins/workspace/final-project/target/jacoco.exec
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ABCtechnologies ---
[INFO] skip non existing resourceDirectory /root/jenkins/workspace/final-project/src/main/resources
```

```
Dashboard > final-project > #35

project/src/main/resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ ABCtechnologies ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ABCtechnologies ---
[INFO] skip non existing resourceDirectory /root/jenkins/workspace/final-project/src/test/resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ ABCtechnologies ---
[INFO]
[INFO] Changes detected - recompiling the module! :input tree
[INFO] Compiling 1 source file with javac [debug target 1.8] to target/test-classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- surefire:3.2.2:test (default-test) @ ABCtechnologies ---
[INFO] Using auto detected provider
org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -----
```



The screenshot shows a terminal window with a light gray background and a dark gray header bar. The header bar contains various icons and text, including 'Offer Details | Appl...', 'final-project #35 Con...', 'Change Password | ...', 'Terminal - root@ip-1...', 'Tue 22 Apr, 11:46', and 'New Chrome available'. Below the header, the URL 'Not secure 98.81.138.54:8080/job/final-project/35/console' is visible. The main content area displays the following Maven build output:

```
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.abc.dataAccessObject.ProductImpTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.128 s
-- in com.abc.dataAccessObject.ProductImpTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.927 s
[INFO] Finished at: 2025-03-26T15:30:44Z
[INFO] -----
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
```

Maven Package

```
[INFO]   from pom.xml
[INFO] -----[ war ]-----
[INFO]
[INFO] --- jacoco:0.8.6:prepare-agent (jacoco-initialize) @ ABCtechnologies ---
[INFO] argLine set to -
javaagent:/root/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/root/jenkins/workspace/final-project/target/jacoco.exec
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ ABCtechnologies ---
[INFO] skip non existing resourceDirectory /root/jenkins/workspace/final-project/src/main/resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ ABCtechnologies ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ ABCtechnologies ---
[INFO] skip non existing resourceDirectory /root/jenkins/workspace/final-project/src/test/resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ ABCtechnologies
```

```
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (package)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
[Pipeline] packaging
[Pipeline] sh
+ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.abc:ABCtechnologies >-----
[INFO] Building RetailModule 1.0
[INFO]   from pom.xml
[INFO] -----[ war ]-----
```

```
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ ABCtechnologies
---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- surefire:3.2.2:test (default-test) @ ABCtechnologies ---
[INFO] Using auto detected provider
org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.abc.dataAccessObject.ProductImpTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.091 s
-- in com.abc.dataAccessObject.ProductImpTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- war:3.2.2:war (default-war) @ ABCtechnologies ---
[INFO] Packaging webapp
```

```
[INFO]
[INFO] --- war:3.2.2:war (default-war) @ ABCtechnologies ---
[INFO] Packaging webapp
[INFO] Assembling webapp [ABCtechnologies] in [/root/jenkins/workspace/final-project/target/ABCtechnologies-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/root/jenkins/workspace/final-project/src/main/webapp]
[INFO] Webapp assembled in [110 msecs]
[INFO] Building war: /root/jenkins/workspace/final-project/target/ABCtechnologies-1.0.war
[INFO]
[INFO] --- jacoco:0.8.6:report (jacoco-site) @ ABCtechnologies ---
[INFO] Loading execution data file /root/jenkins/workspace/final-project/target/jacoco.exec
[INFO] Analyzed bundle 'RetailModule' with 2 classes
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  6.131 s
[INFO] Finished at: 2025-03-26T15:30:54Z
```

Task 2, Step 3 Set up a master-slave node to distribute the tasks in the pipeline.

An EC2 instance was used to create the worker node.

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a navigation sidebar with sections like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. Below that is the Elastic Block Store section with Volumes, Snapshots, and Lifecycle Manager. Network & Security includes Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. Load Balancing lists Load Balancers. At the bottom are CloudShell and Feedback links. The main area displays a histogram titled 'jenkinsworker' with a single data series. The x-axis represents time from 2023-09-18T00:00:00Z to 2023-09-18T01:00:00Z. The y-axis represents count from 0 to 10. The histogram shows a single bar at 10 for the first bin. A legend indicates the color mapping for different metrics: blue for 'jenkinsworker', green for 'devopsProject', red for 'pgp-abcTechnologie-k8-master', and orange for 'pgp-abcTechnologie-k8-worker'. The overall interface is light gray with dark blue header and sidebar elements.

I connected remotely and installed Java and Maven using the manual in the screenshot below. Git was installed out of the box on the Ubuntu ec2 instance used.

```
1  # On Jenkins Agent VM install the following
2
3  ## Setup Jenkin Agent Home Directory
4
5  mkdir -p /home/jagenthome
6  chmod 777 /home/jagenthome
7
8
9  ## Install Git
10
11 sudo su -
12 sudo apt-get install software-properties-common
13 sudo add-apt-repository ppa:git-core/ppa -y
14 sudo apt-get update
15 sudo apt-get install git -y
16 git --version ( if it writes the version then the installation is successful )
17
18 ## Install JAVA
19
20 sudo add-apt-repository ppa:openjdk-r/ppa
21 sudo apt-get update
22 sudo apt-get install -y openjdk-8-jdk
23
24 ## Install Maven
25
26 cd /tmp ; sudo wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.tar.gz
27 cd /tmp ; sudo tar -xzf apache-maven-3.8.6-bin.tar.gz -C /opt/
28 mv /opt/apache-maven-3.8.6 /opt/maven
29 sudo echo "MAVEN_HOME=/opt/maven" >> /etc/profile
30 sudo echo "PATH=\$MAVEN_HOME/bin:\$PATH" >> /etc/profile
31 source /etc/profile
32
33 ## Download agent.jar given on Jenkins Master Console
```

```

Status
Delete Agent
Configure
Build History
Load Statistics
Log
Build Executor Status
0/4
ubuntu
(offline)

Run from agent command line: (Unix)
curl -s0 http://10.0.0.199:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://10.0.0.199:8080/ -secret b45f90777256b6f7acb13924bf5abbd7eabe7e7fc8c1c89b84b96b38d671839a -name ubuntu -WebSocket -workDir "/root/jenkins"

Run from agent command line: (Windows)
curl.exe -s0 http://10.0.0.199:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://10.0.0.199:8080/ -secret b45f90777256b6f7acb13924bf5abbd7eabe7e7fc8c1c89b84b96b38d671839a -name ubuntu -WebSocket -workDir "/root/jenkins"

Or run from agent command line, with the secret stored in a file: (Unix)
echo b45f90777256b6f7acb13924bf5abbd7eabe7e7fc8c1c89b84b96b38d671839a > secret-file
curl -s0 http://10.0.0.199:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://10.0.0.199:8080/ -secret @secret-file -name ubuntu -WebSocket -workDir "/root/jenkins"

Or run from agent command line, with the secret stored in a file: (Windows)
echo b45f90777256b6f7acb13924bf5abbd7eabe7e7fc8c1c89b84b96b38d671839a > secret-file
curl.exe -s0 http://10.0.0.199:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://10.0.0.199:8080/ -secret @secret-file -name ubuntu -WebSocket -workDir "/root/jenkins"

If you prefer to use TCP instead of WebSockets, remove the -WebSocket option. Run java -jar agent.jar -help for more.
The TCP port is disabled so TCP agents may not be connected. You may still use WebSocket agents. Go to the security configuration screen and change it.
Note: PowerShell users must use curl.exe instead of curl because curl is a default PowerShell cmdlet alias for Invoke-WebRequest.

Monitoring Data

```

```

root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~# sudo su
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~# ls
agent.jar jenkins secret-file snap
root@ip-10-0-0-34:~# java -jar agent.jar -url http://10.0.0.199:8080/ -secret @secret-file -name ubuntu -WebSocket -workDir "/root/jenkins"

```

Connecting agent with jenkins master

```

root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~# sudo su
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~#
root@ip-10-0-0-34:~# ls
agent.jar jenkins secret-file snap
root@ip-10-0-0-34:~# java -jar agent.jar -url http://10.0.0.199:8080/ -secret @secret-file -name ubuntu -
WebSocket -workDir "/root/jenkins"
Apr 22, 2025 1:03:03 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /root/jenkins/remoting as a remoting work directory
Apr 22, 2025 1:03:04 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /root/jenkins/remoting
Apr 22, 2025 1:03:04 PM hudson.remoting.Launcher createEngine
INFO: Setting up agent: ubuntu
Apr 22, 2025 1:03:04 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3261.v9c670a_4748a_9
Apr 22, 2025 1:03:04 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /root/jenkins/remoting as a remoting work directory
Apr 22, 2025 1:03:04 PM hudson.remoting.Launcher$CuiListener status
INFO: WebSocket connection open
Apr 22, 2025 1:03:04 PM hudson.remoting.Launcher$CuiListener status
INFO: Connected

```

Console Output showing that agent is the entity executing the pipeline script. Root directory was set to /root/jenkins on the jenkins server. This directory was also created on the agent. The purpose is to hold the output of completed jobs.

Dashboard > final-project > #35

</> Changes

- [Console Output](#)
- [Edit Build Information](#)
- [Delete build '#35'](#)
- [Timings](#)
- [Git Build Data](#)
- [Test Result](#)
- [Pipeline Overview](#)
- [Pipeline Console](#)
- [Restart from Stage](#)
- [Replay](#)
- [Pipeline Steps](#)

Started by user unknown or anonymous

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [ubuntu](#) in /root/jenkins/workspace/final-project

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Tool Install)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (compile)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] echo

The screenshot shows a Jenkins agent configuration interface. At the top, there's a navigation bar with links like 'Dashboard', 'Nodes', 'ubuntu', and 'Configure'. Below this, a sidebar on the left lists options: 'Delete Agent', 'Configure' (which is selected and highlighted in grey), 'Build History', 'Load Statistics', 'Script Console', 'Log', 'System Information', and 'Disconnect'. The main content area has several input fields and dropdowns. One field is 'Description' with a placeholder 'Plain text' and a 'Preview' link. Another is 'Number of executors' set to '4'. A third is 'Remote root directory' set to '/root/jenkins'. At the bottom is a large blue 'Save' button.

Agent connection view from jenkins

Content of /root/jenkins after job completion

```

root@ip-10-0-0-34:~# ls
agent.jar jenkins secret-file snap
root@ip-10-0-0-34:~# cd jenkins/
root@ip-10-0-0-34:~/jenkins# cd workspace/
root@ip-10-0-0-34:~/jenkins/workspace# cd final-project
root@ip-10-0-0-34:~/jenkins/workspace/final-project# ls
Dockerfile deployabcTecnologies.yml pom.xml.bak    project_required_file_v2 target
README.md pom.xml          project_required_file_v1 src
root@ip-10-0-0-34:~/jenkins/workspace/final-project#

```

Task 3: Write a Docker file.

```

#FROM docker.io/library/ubuntu:18.04
#RUN apt-get -y update && apt-get -y upgrade
#RUN apt-get -y install openjdk-8-jdk wget
#RUN mkdir /usr/local/tomcat
#ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.tar.gz /tmp/apache-tomcat-9.0.95.tar.gz
#RUN cd /tmp && tar xvfz apache-tomcat-9.0.95.tar.gz
#RUN cp -Rv /tmp/apache-tomcat-9.0.95/* /usr/local/tomcat/
#ADD */*.war /usr/local/tomcat/webapps
#EXPOSE 8080
#CMD /usr/local/tomcat/bin/catalina.sh run

FROM ubuntu:18.04

RUN apt-get -y update
RUN apt-get -y install openjdk-8-jdk wget

RUN wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.55/bin/apache-tomcat-8.5.55.tar.gz -O /tmp/tomcat.tar.gz
RUN cd /tmp && tar xvfz tomcat.tar.gz
RUN mv /tmp/apache-tomcat-8.5.55 /opt/tomcat

COPY target/ABCtechnologies-1.0.war /opt/tomcat/webapps/

EXPOSE 8080

```

1,1 Top

Pipeline Script to build and push image to remote registry

```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

```

stage('package'){
    steps{
        echo 'packaging'
        sh 'mvn package'
    }
}
stage('dockerize'){
    steps{
        echo 'containerizing application'
        sh 'docker build -t docker.io/evansken1/abctechnologies:$BUILD_NUMBER .'
    }
}
stage('push docker image') {
    steps {
        withDockerRegistry(credentialsId: 'DOCKER_HUB_LOGIN', url: 'https://index.docker.io/v1/')
        sh 'docker push docker.io/evansken1/abctechnologies:$BUILD_NUMBER'
    }
}

```

[Save](#) [Apply](#)

I had to set up my docker hub credentials in jenkins to enable it to login and push the image.
Configuration in the image below

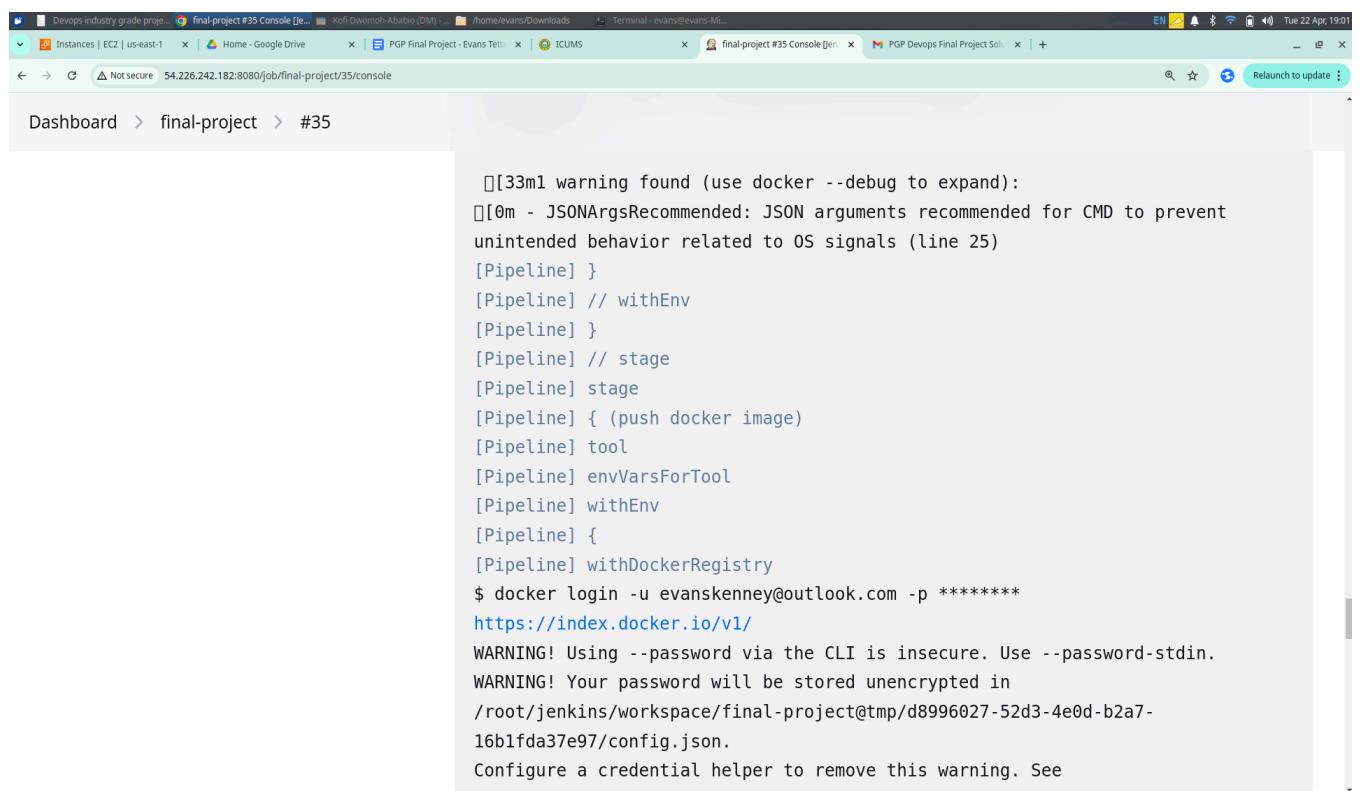
The screenshot shows the Jenkins 'Update credentials' interface. At the top, there are buttons for 'Update', 'Delete', and 'Move'. Below these, the 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Username' field contains 'evanskenney@outlook.com'. There is an unchecked checkbox for 'Treat username as secret'. The 'Password' field is marked as 'Concealed' and has a 'Change Password' button next to it. The 'ID' field is set to 'DOCKER_HUB_LOGIN'. The 'Description' field contains 'DOCKER_HUB_LOGIN'. At the bottom is a 'Save' button.

Console output while running this aspect of the script

```
[INFO] -----
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (dockerize)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
[Pipeline] containerizing application
[Pipeline] sh
+ docker build -t docker.io/evansken1/abctechnologies:35 .
#0 building with "default" instance using docker driver

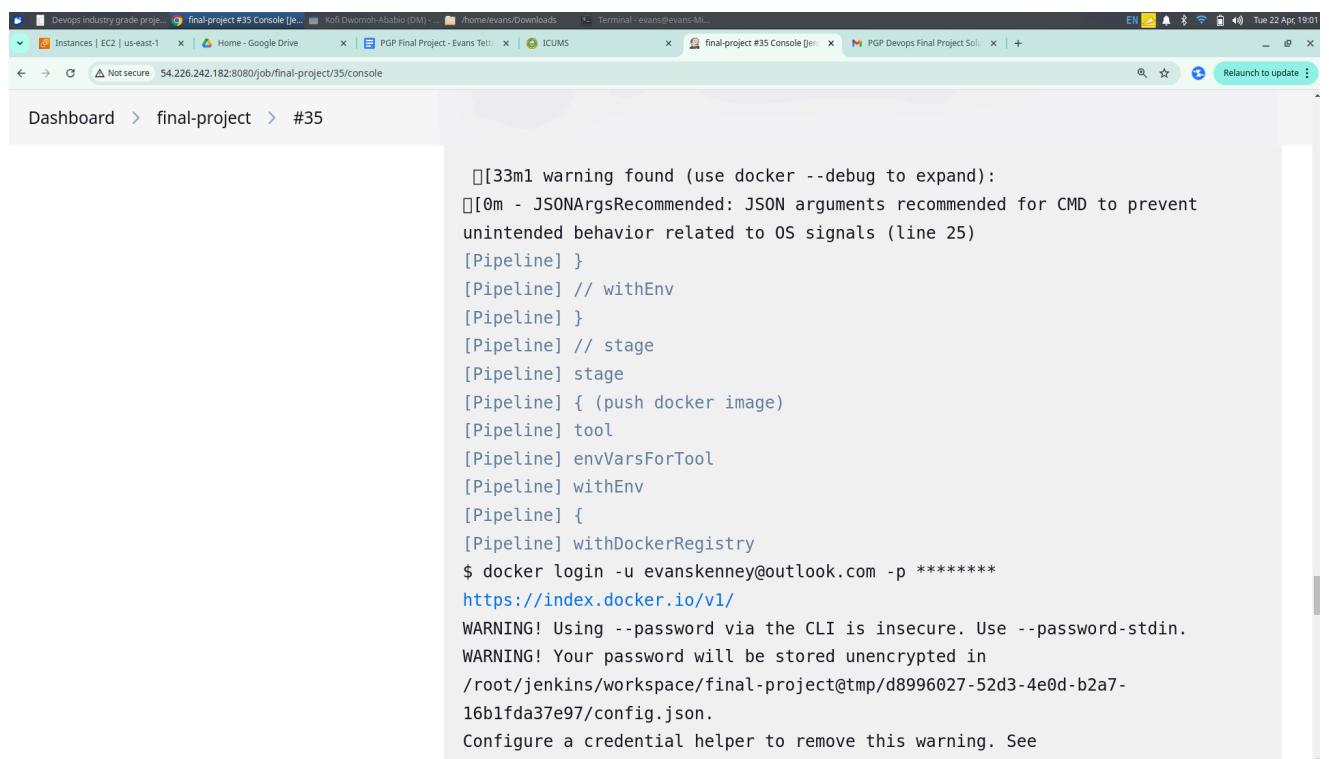
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 918B done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/ubuntu:18.04
```



Dashboard > final-project > #35

```
□[33m1 warning found (use docker --debug to expand):  
□[0m - JSONArgsRecommended: JSON arguments recommended for CMD to prevent  
unintended behavior related to OS signals (line 25)  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (push docker image)  
[Pipeline] tool  
[Pipeline] envVarsForTool  
[Pipeline] withEnv  
[Pipeline] {  
[Pipeline] withDockerRegistry  
$ docker login -u evanskenney@outlook.com -p *****  
https://index.docker.io/v1/  
WARNING! Using --password via the CLI is insecure. Use --password-stdin.  
WARNING! Your password will be stored unencrypted in  
/root/jenkins/workspace/final-project@tmp/d8996027-52d3-4e0d-b2a7-  
16b1fda37e97/config.json.  
Configure a credential helper to remove this warning. See
```



Dashboard > final-project > #35

```
□[33m1 warning found (use docker --debug to expand):  
□[0m - JSONArgsRecommended: JSON arguments recommended for CMD to prevent  
unintended behavior related to OS signals (line 25)  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (push docker image)  
[Pipeline] tool  
[Pipeline] envVarsForTool  
[Pipeline] withEnv  
[Pipeline] {  
[Pipeline] withDockerRegistry  
$ docker login -u evanskenney@outlook.com -p *****  
https://index.docker.io/v1/  
WARNING! Using --password via the CLI is insecure. Use --password-stdin.  
WARNING! Your password will be stored unencrypted in  
/root/jenkins/workspace/final-project@tmp/d8996027-52d3-4e0d-b2a7-  
16b1fda37e97/config.json.  
Configure a credential helper to remove this warning. See
```

The terminal window shows a session titled 'final-project #35 Console [jen]' with the following output:

```
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
[Pipeline] {
[Pipeline] sh
+ docker push docker.io/evansken1/abctechnologies:35
The push refers to repository [docker.io/evansken1/abctechnologies]
14fdb9101703: Preparing
0bf66f7f710a: Preparing
90224bef7710: Preparing
48b5c5376db2: Preparing
3dbd8fc2b2d6a: Preparing
771d65021a48: Preparing
548a79621a42: Preparing
771d65021a48: Waiting
548a79621a42: Waiting
0bf66f7f710a: Layer already exists
48b5c5376db2: Layer already exists
90224bef7710: Layer already exists
3dbd8fc2b2d6a: Layer already exists
771d65021a48: Layer already exists
```

After the push command, the session continues with:

```
[Pipeline] sh
+ docker push docker.io/evansken1/abctechnologies:35
The push refers to repository [docker.io/evansken1/abctechnologies]
14fdb9101703: Preparing
0bf66f7f710a: Preparing
90224bef7710: Preparing
48b5c5376db2: Preparing
3dbd8fc2b2d6a: Preparing
771d65021a48: Preparing
548a79621a42: Preparing
771d65021a48: Waiting
548a79621a42: Waiting
0bf66f7f710a: Layer already exists
48b5c5376db2: Layer already exists
90224bef7710: Layer already exists
3dbd8fc2b2d6a: Layer already exists
771d65021a48: Layer already exists
548a79621a42: Layer already exists
14fdb9101703: Pushed
35: digest:
sha256:6df80e1e98f218a4999e94a251cb3f9a0cfaa4d21732d74f2e75e1bdd493aa82 size:
1801
```

Verifying Image pushed to Dockerhub

The screenshot shows a Docker Hub repository page for the user 'evansken1'. The repository name is 'abctechnologies/general'. The page displays a table of tags:

Tag	OS	Type	Pulled	Pushed
35		Image	27 days	27 days

Below the table, there is a section titled 'Repository overview' with a status of 'INCOMPLETE'. A button labeled 'Add overview' is visible.

On the right side of the page, there is an advertisement for 'buildcloud' and a 'Docker commands' section with a command input field containing 'docker push evansken1/abctechnologies:tagname'.

Task 4: Integrate the Docker host with Ansible. Write an Ansible playbook to create an image and create a continuer. Integrate Ansible with Jenkins. Deploy Ansible-playbook. CI/CD job to build code on ansible and deploy it on docker container

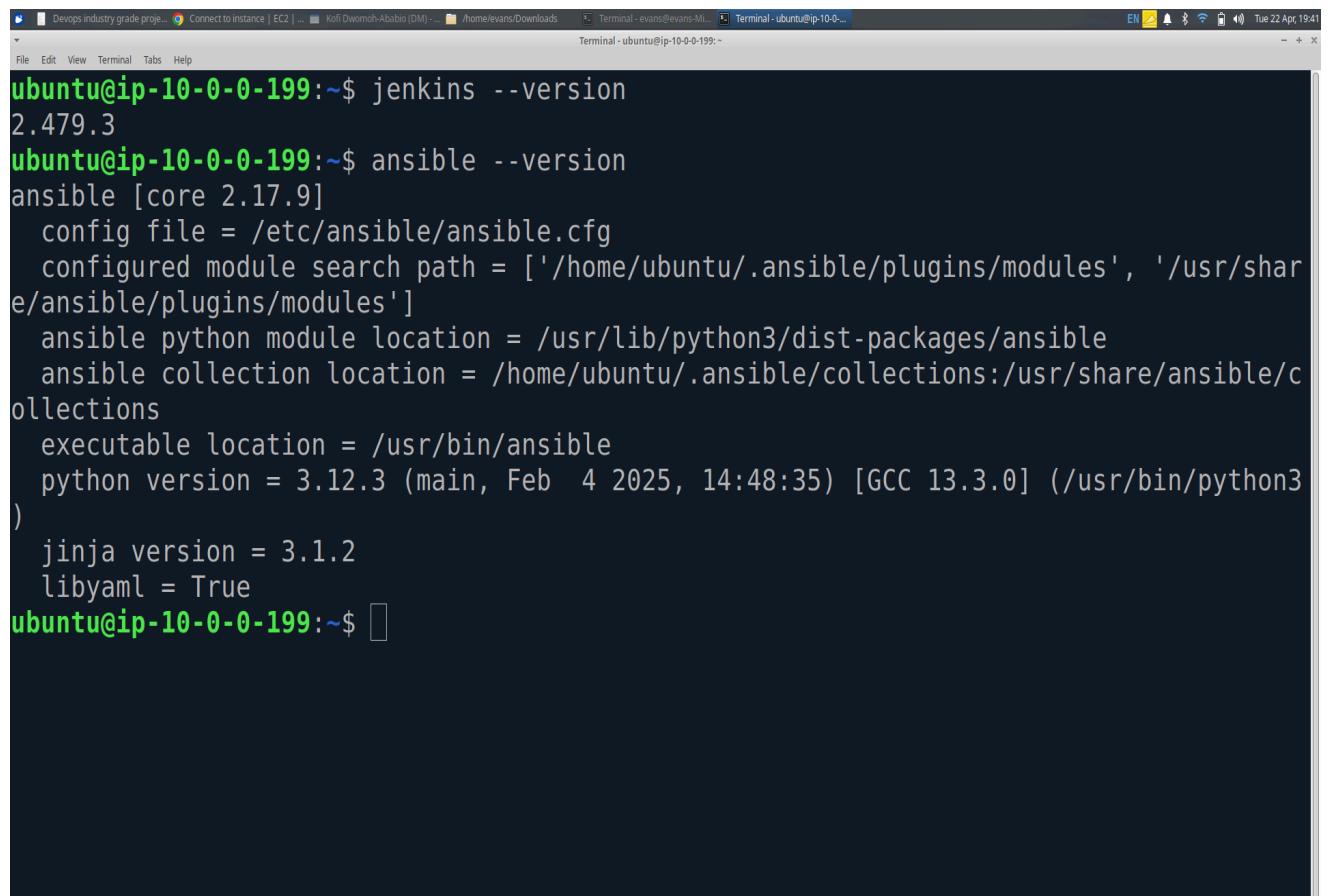
Docker image has already been created in the previous step so I'll proceed to use ansible to deploy to kubernetes.

Step 1: Install Ansible on jenkins server

I used the script below to complete this step

```
sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installAnsible.sh -P /tmp
sudo chmod 755 /tmp/installAnsible.sh
sudo bash /tmp/installAnsible.sh
```

Verify ansible installation from jenkins server



The screenshot shows a terminal window with two tabs. The active tab is titled 'Terminal - ubuntu@ip-10-0-0-199:~\$'. It displays the output of running Jenkins and Ansible commands. The Jenkins command 'jenkins --version' shows version 2.479.3. The Ansible command 'ansible --version' provides detailed information about the configuration, module search path, Python module location, collection location, executable location, Python version, Jinja version, and libyaml status.

```
ubuntu@ip-10-0-0-199:~$ jenkins --version
2.479.3
ubuntu@ip-10-0-0-199:~$ ansible --version
ansible [core 2.17.9]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb 4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-10-0-0-199:~$
```

Kubeadm was used to set up a two node cluster. Amazon EC2 nodes were used
I followed this manual:

<https://github.com/lerndevops/kubernetes/blob/master/1-intall/install-kubernetes-v1.24-ubuntu-debian.md>

to complete the cluster creation

The screenshot shows the AWS CloudWatch Metrics Dashboard. On the left, there's a navigation sidebar with links like EC2, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main area displays two selected instances: 'ppg-abcTechnologie-k8-master' and 'ppg-abcTechnologie-k8-worker'. Below the instances, there are several metrics visualized as line charts:

- CPU utilization (%)**: Shows utilization over time from 19:00 to 19:45. The chart indicates 'No unit' and 'No data available'.
- Network in (bytes)**: Shows incoming network traffic over the same period. It also indicates 'No unit' and 'No data available'.
- Network out (bytes)**: Shows outgoing network traffic over the same period. It also indicates 'No unit' and 'No data available'.
- Network packets in (count)**: Shows incoming network packets over the same period. It also indicates 'No unit' and 'No data available'.
- Network packets out (count)**: Shows outgoing network packets over the same period. It also indicates 'No unit' and 'No data available'.
- Metadata no token (count)**: Shows metadata tokens over the same period. It also indicates 'No unit' and 'No data available'.
- CPU credit usage (count)**: Shows CPU credit usage over the same period. It also indicates 'No unit' and 'No data available'.
- CPU credit balance (count)**: Shows CPU credit balance over the same period. It also indicates 'No unit' and 'No data available'.

At the bottom right of the dashboard, there are copyright notices: "© 2025, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Ansible host file located at /etc/ansible/hosts is populated with kubernetes master node ip and user credential (devops user)

```
ubuntu@ip-10-0-0-199:~$ vi /etc/ansible/hosts
```

The terminal window shows the command `vi /etc/ansible/hosts` being run. The file contains the following content:

```
[k8s-master]
10.0.0.199 devops
```

```

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110
#
## [kubernetes]
10.0.0.160 ansible_user=devops ansible_ssh_pass=today@1234

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

```

26,1 28%

I followed

<https://github.com/evanskenney/labs/blob/master/ansible/install/ssh-keys-setup.md>

to set up 2 identical users (devops) on both jenkins server and kubernetes master node. This would allow ssh connectivity to the k8 master node from jenkins server. I also added jenkins user to the sudoers file to allow it to execute commands with elevated privileges without requiring any password. `jenkins ALL=(ALL) NOPASSWD: ALL` is being used in the sudoers file.

Verifying jenkins server connection to kubernetes master node via ansible. The ping module was used

```

ubuntu@ip-10-0-0-199:~$ ansible kubernetes -m ping
[WARNING]: Platform linux on host 10.0.0.160 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
10.0.0.160 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-10-0-0-199:~$ 

```

Ansible playbook to deploy kubernetes artefacts (deployment, pod and service).

I added the playbook to the project source code so that when I run the pipeline, it will be available on agent machine which can then be used against the kubernetes master node

Playbook (deployabcTecnologies.yml) in remote repo

The screenshot shows a GitHub repository page for 'evanskenney / final-devops-project'. The repository has 9 commits ahead of the main branch. The 'Code' tab is selected, showing the contents of the 'deployabcTecnologies.yml' file. The file contains an Ansible playbook with tasks for deploying pods and services. The repository has 0 stars and 0 forks. It includes sections for About, Releases, Packages, and Languages.

```
evanskenney / final-devops-project · Public
forked from salwad-basha-shaik/final-devops-project

Code Pull requests Actions Projects Security Insights

This branch is 9 commits ahead of salwad-basha-shaik/final-devops-project:main.

(evans Kenney) replace sampleapp with abctechnologies 8f5f4ac · last month 36 Commits
.github/workflows Update maven.xml 8 months ago
.settings Initial Upload 3 years ago
project_required_file_v1 renamed the folders 3 years ago
project_required_file_v2 added hosts file in version2 3 years ago
src Initial Upload 3 years ago
target Added final submission files 3 years ago
.classpath Initial Upload 3 years ago
.project Initial Upload 3 years ago
Dockerfile revise dockerfile 3 months ago
README.md updating readme file 3 months ago
deployabcTecnologies.yml replace sampleapp with abctechnologies last month
pom.xml Initial Upload 3 years ago
pom.xml.bak Initial Upload 3 years ago

About
PGP Final Project
Readme
Activity
0 stars
0 watching
0 forks
Report repository

Releases
No releases published

Packages
No packages published

Languages
JavaScript 52.7% HTML 34.2%
Shell 5.0% Java 3.7% CSS 3.7%
Dockerfile 0.7%
```

Content of playbook

The screenshot shows the 'Code' tab of the GitHub repository page for 'evanskenney / final-devops-project'. The file 'deployabcTecnologies.yml' is selected. The code in the file is an Ansible playbook with the following content:

```
evanskenney / final-devops-project · Public
forked from salwad-basha-shaik/final-devops-project

Code Pull requests Actions Projects Security Insights

final-devops-project / deployabcTecnologies.yml · 30 lines (29 loc) · 1.81 KB
(evans Kenney) replace sampleapp with abctechnologies 8f5f4ac · last month History

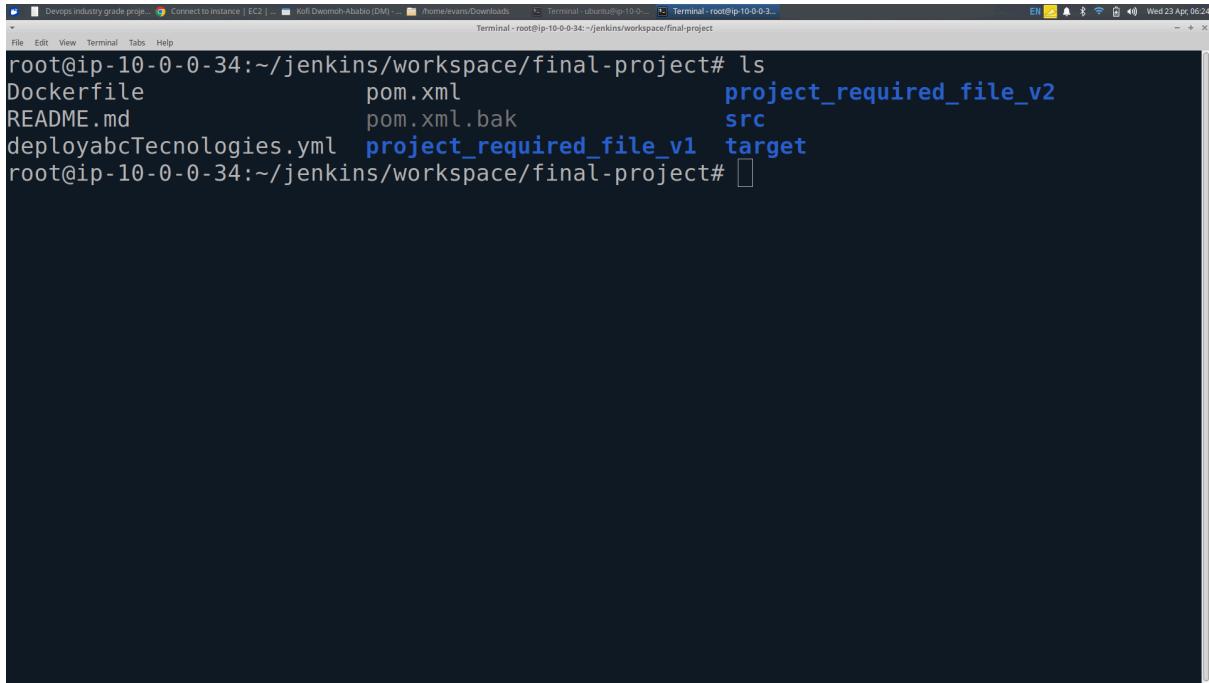
1 - hosts: "({ env })"
2 become: yes
3 tasks:
4 - name: Deploying Application pods...
5 shell:
6   if [ `kubectl --kubeconfig=/home/devops/.kube/config get deployment | grep -v NAME | awk '{print $1}' | grep abctechnologies | wc -l` -gt 0 ]; then
7     echo "Deleting previous application deployment"
8     kubectl --kubeconfig=/home/devops/.kube/config delete deployment `kubectl get deployment | grep -v NAME | awk '{print $1}' | grep abctechnologies`
9     echo "Creating new application deployment"
10    kubectl --kubeconfig=/home/devops/.kube/config create deployment abctechnologies --image=evansken1/abctechnologies:{{ build }}
11  else
12    echo "Deploying ABCtechnologies Application"
13    kubectl --kubeconfig=/home/devops/.kube/config create deployment abctechnologies --image=evansken1/abctechnologies:{{ build }}
14  fi
15 - name: deploying service
16 shell:
17   if [ `kubectl --kubeconfig=/home/devops/.kube/config get svc | grep abctechnologies | awk '{print $1}' | wc -l` -gt 0 ]; then
18     echo "App service found, No actions taken"
19     #kubectl delete svc `kubectl get svc | grep abctechnologies | awk '{print $1}'`
20   else
21     echo "Creating App Services"
22     kubectl --kubeconfig=/home/devops/.kube/config expose deployment abctechnologies --name abctechnologies --type NodePort --port 80 --target-port 8080
23   fi
24 - name: increase replicas
25 shell: kubectl --kubeconfig=/home/devops/.kube/config scale deploy abctechnologies --replicas=2
```

```

- hosts: "{{ env }}"
  become: yes
  tasks:
    - name: Deploying Application pods...
      shell: |
        if [ `kubectl --kubeconfig=/home/devops/.kube/config get deployment | grep -v NAME | awk '{print $1}' | grep abctechnologies | wc -l` -gt 0 ]; then
          echo "deleteing previous application deployment"
          kubectl --kubeconfig=/home/devops/.kube/config delete deployment `kubectl get deployment | grep -v NAME | awk '{print $1}' | grep abctechnologies`
          echo "creating new application deployment"
          kubectl --kubeconfig=/home/devops/.kube/config create deployment abctechnologies --image=evansken1/abctechnologies:{{ build }}
        else
          echo "Deploying ABCtechnologies Application"
          kubectl --kubeconfig=/home/devops/.kube/config create deployment abctechnologies --image=evansken1/abctechnologies:{{ build }}
        fi
    - name: deploying service
      shell: |
        if [ `kubectl --kubeconfig=/home/devops/.kube/config get svc | grep abctechnologies | awk '{print $1}' | wc -l` -gt 0 ]; then
          echo "app service found, No actions taken"
          #kubectl delete svc `kubectl get svc | grep abctechnologies | awk '{print $1}'``
        else
          echo "Creating App Services"
          kubectl --kubeconfig=/home/devops/.kube/config expose deployment abctechnologies --name abctechnologies --type NodePort --port 80 --target-port 8080
        fi
    - name: increase replicas
      shell: kubectl --kubeconfig=/home/devops/.kube/config scale deploy abctechnologies --replicas=2

```

deployabcTecnologies.yml file present on agent node after running pipeline which always clones the repo.



```
root@ip-10-0-0-34:~/jenkins/workspace/final-project# ls
Dockerfile           pom.xml          project_required_file_v2
README.md            pom.xml.bak      src
deployabcTecnologies.yml  project_required_file_v1  target
root@ip-10-0-0-34:~/jenkins/workspace/final-project#
```

From here I was able to add another stage in the pipeline script to handle the deployment.

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script

```
55
54 +
55 +     }
56 +     stage('dockerize'){
57 +         steps{
58 +             echo 'containerizing application'
59 +             sh 'docker build -t docker.io/evansken1/abctechnologies:$BUILD_NUMBER .'
60 +         }
61 +     }
62 +     stage('push docker image'){
63 +         steps {
64 +             withDockerRegistry(credentialsId: 'DOCKER_HUB_LOGIN', url: 'https://index.docker.io/v1/') {
65 +                 sh 'docker push docker.io/evansken1/abctechnologies:$BUILD_NUMBER'
66 +             }
67 +         }
68 +     }
69 +     stage('Deploy to kubernetes'){
70 +         steps{
71 +             sh script: 'sudo ansible-playbook /root/jenkins/workspace/final-project/deployabcTecnologies.yml --extra-vars "env=kubernetes build=$BUILD_NUMBER"'
72 +         }
73 +     }
74 + }
```

Use Groovy Sandbox

Pipeline Syntax

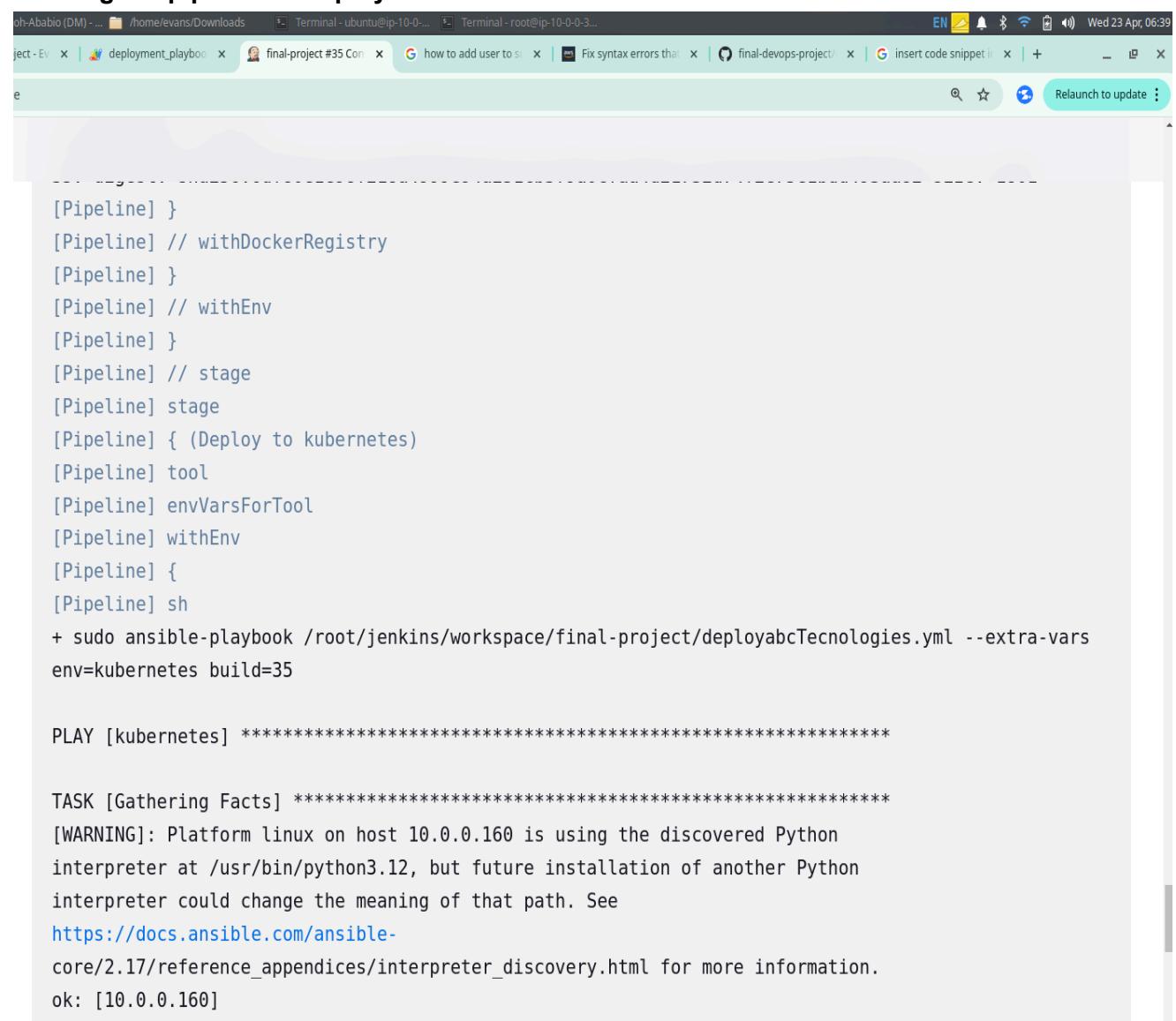
Advanced

```

stage('Deploy to kubernetes'){
    steps{
        sh script: 'sudo ansible-playbook
/root/jenkins/workspace/final-project/deployabcTechnologies.yml
--extra-vars "env=kubernetes build=$BUILD_NUMBER"'
    }
}

```

Running the pipeline to deploy to kubernetes



The screenshot shows a terminal window with multiple tabs open at the top. The tabs include:

- object - Ev...
- deployment_playboo...
- final-project #35 Con...
- how to add user to s...
- Fix syntax errors tha...
- final-devops-project/...
- insert code snippet i...

The main terminal area displays the Jenkins pipeline logs for a build step. The logs show the execution of an Ansible playbook to deploy to Kubernetes. Key parts of the log output include:

```

[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to kubernetes)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ sudo ansible-playbook /root/jenkins/workspace/final-project/deployabcTechnologies.yml --extra-vars
env=kubernetes build=35

PLAY [kubernetes] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 10.0.0.160 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference\_appendices/interpreter\_discovery.html for more information.
ok: [10.0.0.160]

```

```

joh-Ababio (DM) - ... /home/evans/Downloads Terminal - ubuntu@ip-10-0-0-2... Terminal - root@ip-10-0-0-3...
object-Ev deployment_playbo final-project #35 Com G how to add user to s Fix syntax errors tha final-devops-project/ insert code snippet Relaunch to update

le

ok: [10.0.0.160]

TASK [Deploying Application pods...] *****
changed: [10.0.0.160]

TASK [deploying service] *****
changed: [10.0.0.160]

TASK [increase replicas] *****
changed: [10.0.0.160]

PLAY RECAP *****
10.0.0.160 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Pipeline Overview

Not secure 3.80.35.195:8080/job/final-project/35/pipeline-graph/ Relaunch to update

The Jenkins Pipeline Overview for Build #35 displays a successful pipeline execution. The pipeline consists of the following stages:

- Start
- Tool Install (green checkmark)
- compile (green checkmark)
- unit-test (green checkmark)
- package (green checkmark)
- dockerize (green checkmark)
- push docker im... (green checkmark)
- Deploy to kube... (green checkmark)
- End

Details of the build:

- Started 27 days ago
- Queued 2 ms
- Took 37 sec

Actions available: Rebuild, Console, Configure.

Verifying Deployments on kubernetes

```
Devops industry grade project Connect to instance | EC2 | Kofi Dwomoh-Ababio (DM) home/evans/Downloads Terminal - ubuntu@ip-10-0-0-2 Terminal - root@ip-10-0-0-3 Terminal - devops@ip-10-0-0-160... Terminal - devops@ip-10-0-0-160... File Edit View Terminal Tab Help Terminal - devops@ip-10-0-0-160:~$ kubectl get pods NAME READY STATUS RESTARTS AGE abctechnologies-7db8c7cd96-rcp7l 1/1 Running 12 (14d ago) 27d abctechnologies-7db8c7cd96-vxtpf 1/1 Running 12 (14d ago) 27d devops@ip-10-0-0-160:~$ kubectl get deployment NAME READY UP-TO-DATE AVAILABLE AGE abctechnologies 2/2 2 2 27d devops@ip-10-0-0-160:~$ devops@ip-10-0-0-160:~$ kubectl get svc NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE abctechnologies NodePort 10.108.223.132 <none> 80:31380/TCP 27d kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 30d devops@ip-10-0-0-160:~$
```

Accessing ABCTechnologies on Browser

Security group for aws node has been updated to allow port range 30000 - 32003
Service for abctechnologies can be accessed on port 31380.

<http://54.89.158.236:31380>

The screenshot shows the Apache Tomcat 8.5.55 homepage. The main message is "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this, there's a cartoon cat logo and a list of recommended readings: Security Considerations How-To, Manager Application How-To, and Clustering/Session Replication How-To. On the right side, there are buttons for Server Status, Manager App, and Host Manager. The navigation bar includes Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. The footer contains links for Other Downloads, Other Documentation, Get Involved, Miscellaneous, and Apache Software Foundation.

<http://54.89.158.236:31380/ABCtechnologies-1.0/>



Task 5: Prometheus and Grafana Monitoring

I followed <https://prometheus.io/download/> to get the binaries for prometheus server and node exporter

Adding configs to prometheus.yml file

```
# my global config
global:
  scrape_interval: 10s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
```

1,6

Top

```
- targets:
  # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
        # The label name is added as a label `label_name=<label_value>` to any timeseries scraped from this config.
        labels:
          app: "prometheus"
  - job_name: "node-exporter"
    static_configs:
      - targets: [localhost:9100]
```

35,6

Bot

Accessing targets interface from browser

The screenshot shows the Prometheus Targets interface. At the top, there are three input fields: "Select scrape pool", "Filter by target health", and "Filter by endpoint or labels". Below these are two sections, each with a green header bar.

node-exporter section:

Endpoint	Labels	Last scrape	State
http://localhost:9100/metrics	instance="localhost:9100", job="node-exporter"	952ms ago, 15ms	UP

prometheus section:

Endpoint	Labels	Last scrape	State
http://localhost:9090/metrics	app="prometheus", instance="localhost:9090", job="prometheus"	2.367s ago, 4ms	UP

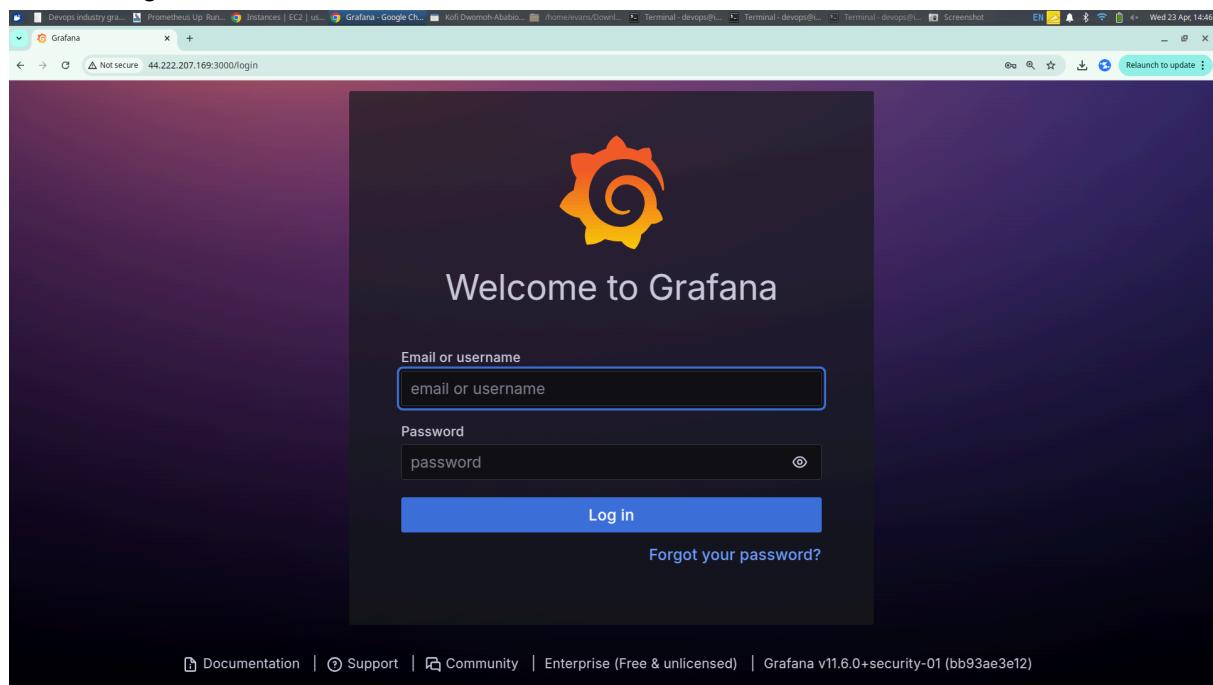
Prometheus query interface

The screenshot shows the Prometheus Query interface. At the top, there are three navigation tabs: "Query" (selected), "Alerts", and "Status". Below the tabs is a search bar with placeholder text "Enter expression (press Shift+Enter for newlines)". To the right of the search bar is a blue "Execute" button. Underneath the search bar is a dropdown menu with options "Table" (selected), "Graph", and "Explain".

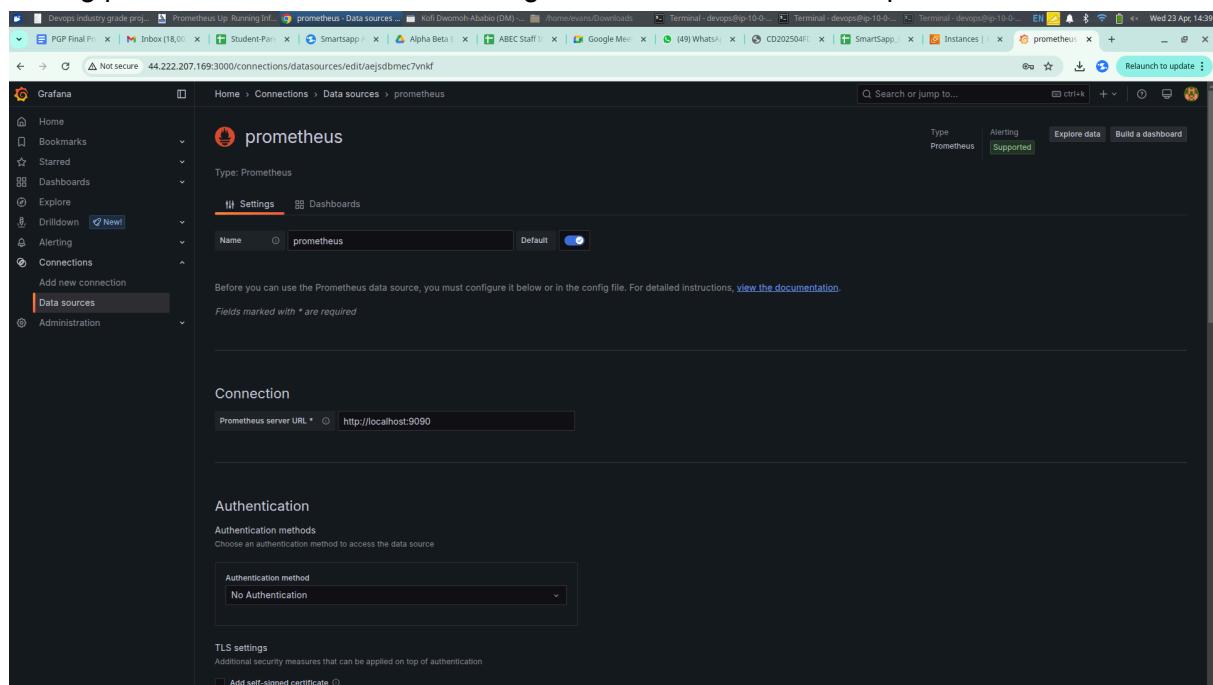
A horizontal slider allows adjusting the "Evaluation time". Below the slider, the message "No data queried yet" is displayed. At the bottom left is a blue "Add query" button.

I followed <https://grafana.com/grafana/download> to download and install grafana on k8 master node

Grafana login interface



Adding prometheus as a data source in grafana; Prometheus url: http://localhost:9090

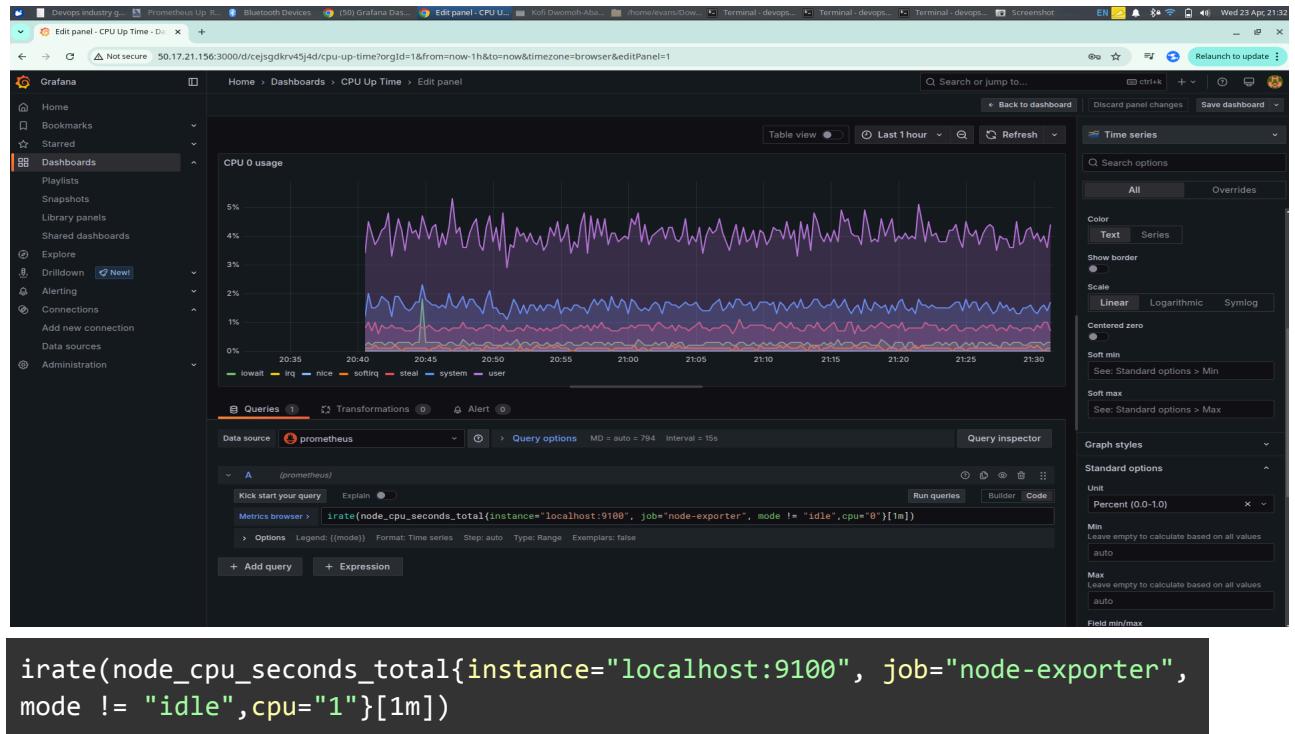


The screenshot shows the 'Data sources' configuration page in Grafana. The URL in the address bar is `http://44.222.207.169:3000/connections/datasources/edit/aej5dbmec7vnkf`. The left sidebar has 'Data sources' selected. The main area shows 'TLS settings' with 'Skip TLS certificate validation' checked. Below it is the 'Advanced settings' section, which includes 'HTTP headers', 'Alerting' (with a toggle switch), 'Interval behaviour' (with 'Scrape interval' set to 15s and 'Query timeout' set to 60s), and 'Query editor' (with 'Default editor' set to 'Builder'). A note at the bottom says 'Successfully queried the Prometheus API.'

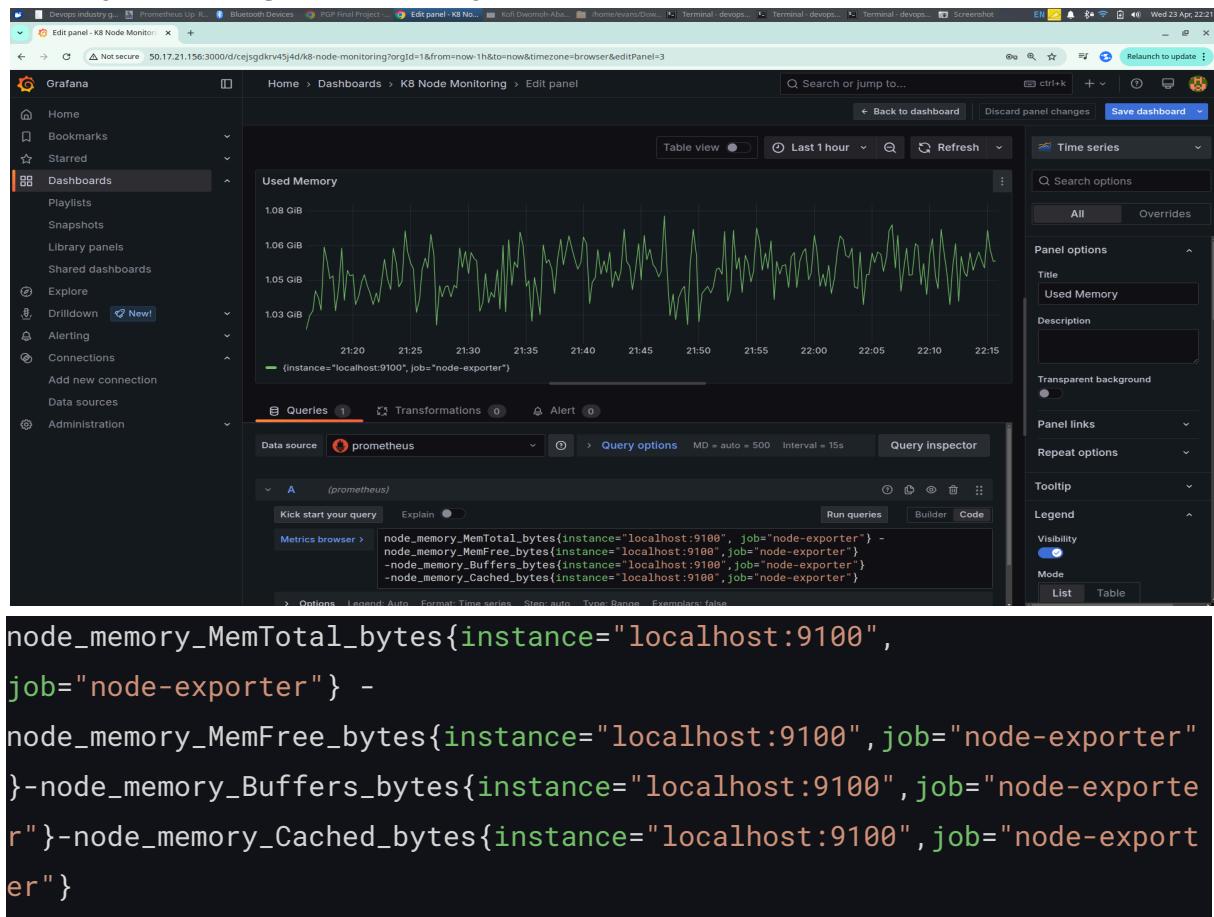
Saved and successfully tested connection to prometheus url: `http://localhost:9090`

This screenshot shows the same 'Data sources' configuration page as the previous one, but with different settings. The 'Default editor' is now set to 'Builder'. The 'Performance' section includes 'Prometheus type' (set to 'Prometheus'), 'Prometheus version' (set to 'Please select'), 'Cache level' (set to 'Low'), 'Incremental querying (beta)' (unchecked), and 'Disable recording rules (beta)' (unchecked). The 'Other' section includes 'Custom query parameters' (set to 'Example: max_source_resolution=5m&timeout') and 'HTTP method' (set to 'POST'). The 'Exemplars' section has a '+ Add' button. A green success message at the bottom states: 'Successfully queried the Prometheus API. Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#)'. At the bottom are 'Delete' and 'Save & test' buttons.

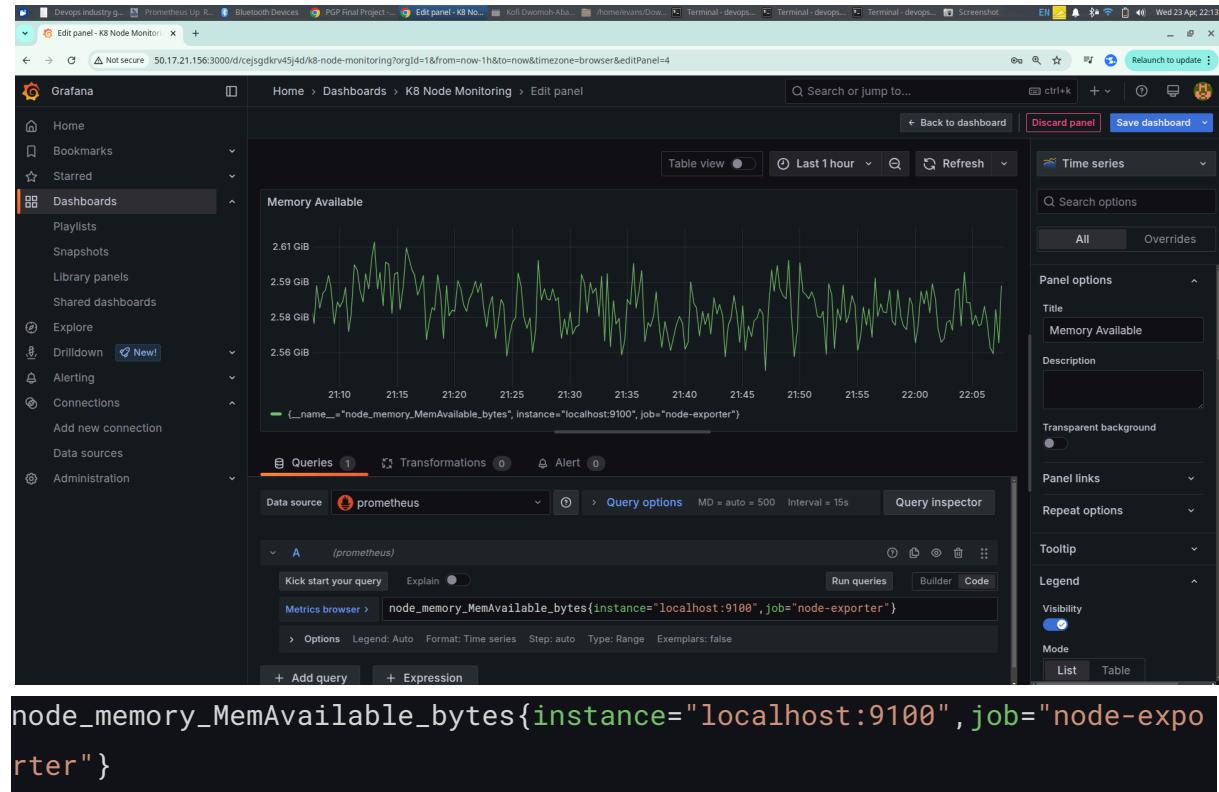
CPU Monitoring



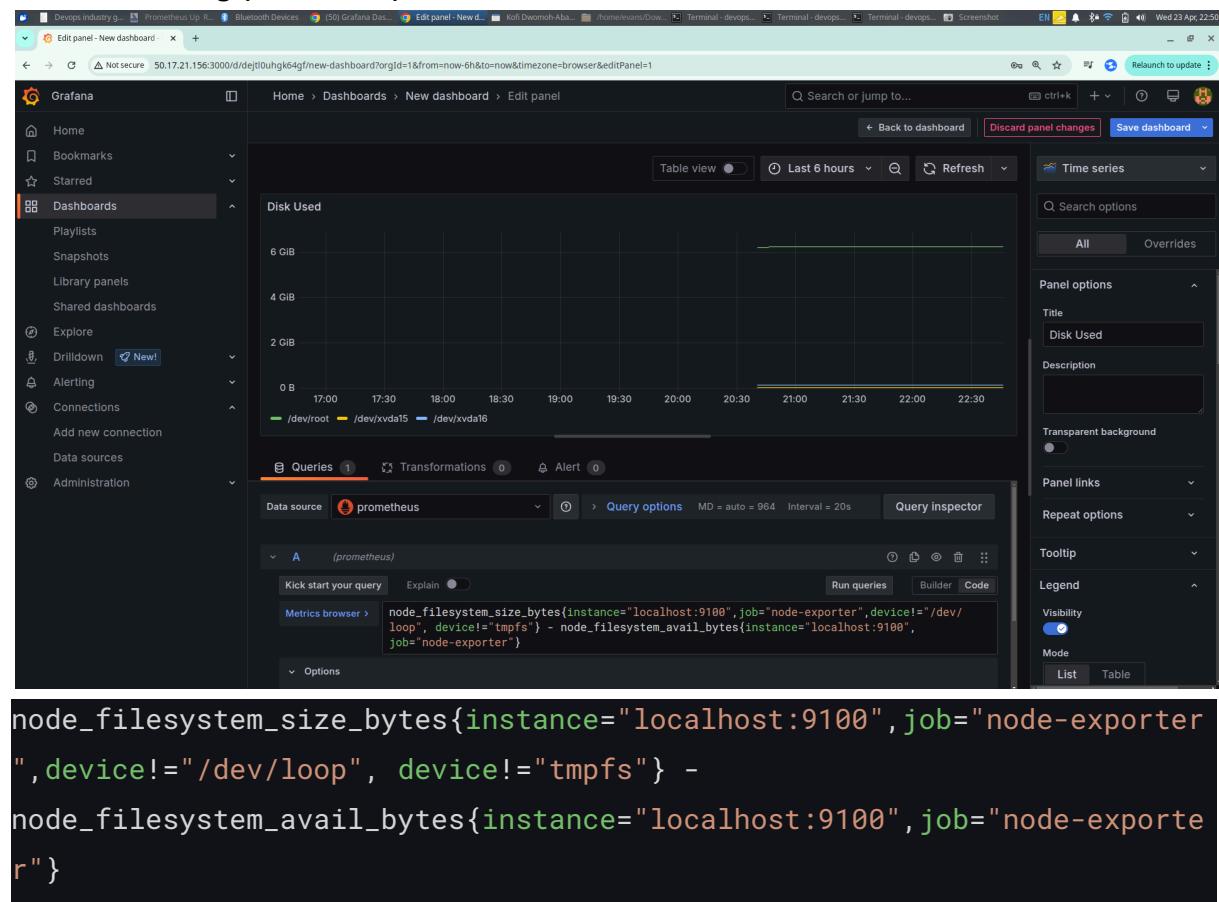
Memory Monitoring (Used Memory on node)



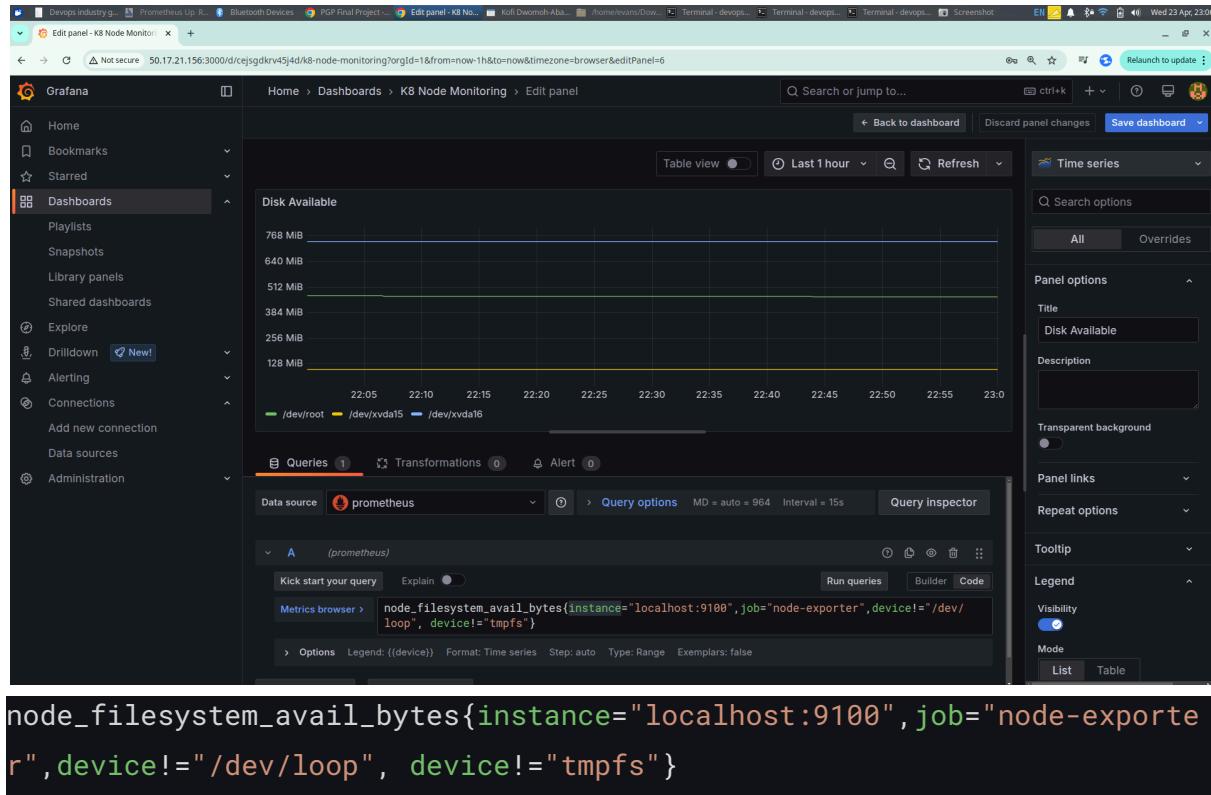
Memory monitoring (Memory Available)



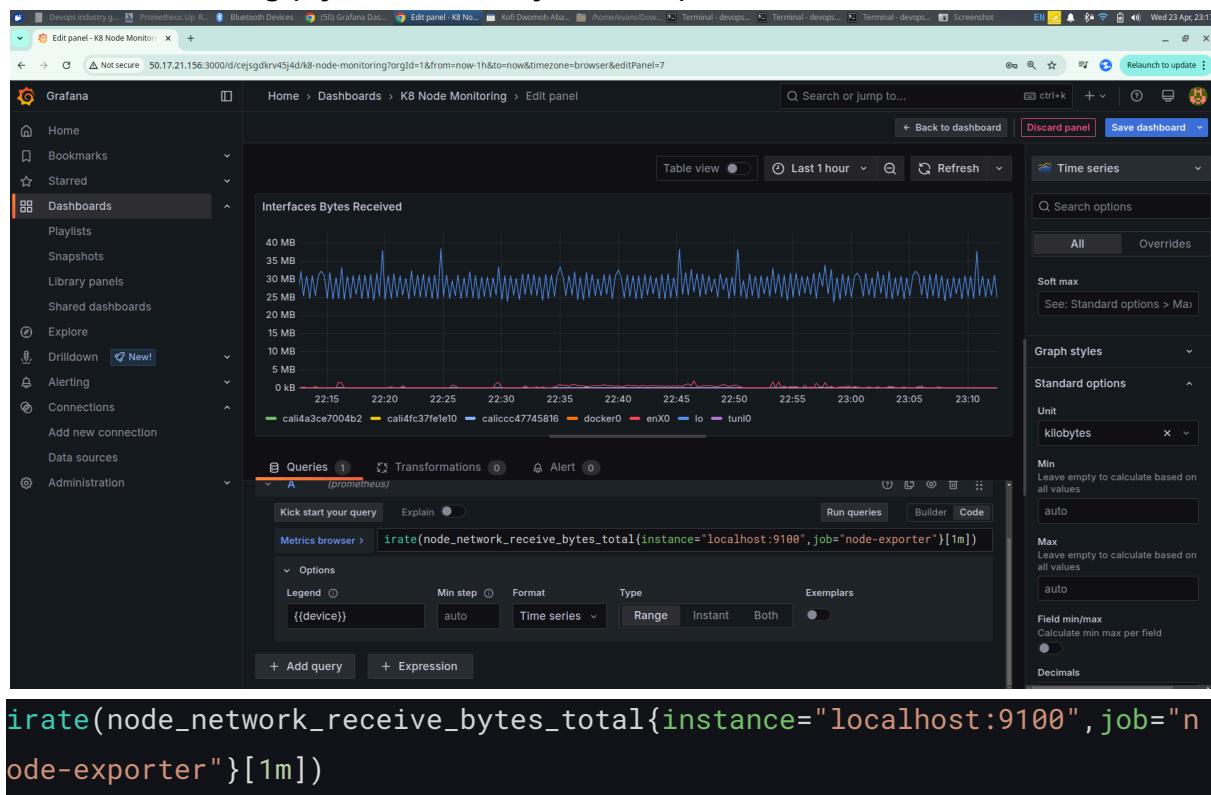
Disk Monitoring (Disk Used)



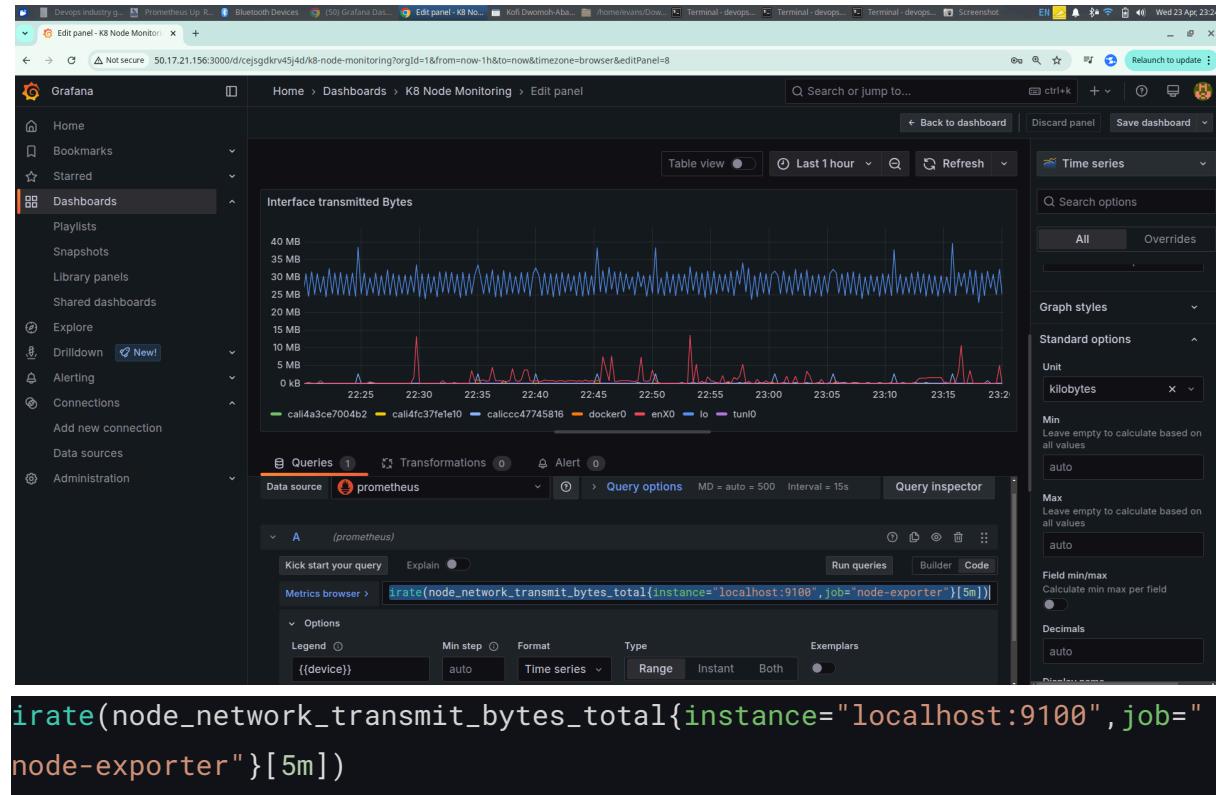
Disk monitoring (Disk Available)



Network Monitoring (Bytes received by interface)



Network Monitoring (Bytes transmitted by Interfaces)



Summary of dashboards in one view



Thank You.