

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Building a CI/CD Pipeline for a Retail Company - ABC Technologies.

Goals:

- Highly Available
- Highly scalable
- Highly performant
- Easily built and maintained
- Developed and deployed quickly
- Lower production bugs
- Frequent releases

Prerequisites:

- Java
- Maven
- Git
- Jenkins
- Docker
- Ansible
- Kubernetes
- Grafana
- Prometheus

My Approach:

Steps:

- Cloned the project SRC from GitHub Link to Local Machine
- Created a new GitHub repo: link https://github.com/iambowcreek/abc_technologies.git
- Initiated Git in Local Machine
- Pushed the SRC to the New Repo Created on GitHub
- Servers provisioned - 3 Servers for the following integration
 - Server A: Jenkins Master, Ansible Master, Docker, Maven
 - Server B: K8s Master, Ansible Node, Docker, Prometheus and Grafana
 - Server C: K8s Node
- Server A: Configured Jenkins and installed all necessary plugins including Maven for the pipeline integration, Installed Ansible – configured the node for communication in server B, installed docker and integrated it with Jenkins, integrated ansible with docker and K8S.
- Server B: Installed K8S, joined the node in server C. Created Ansible user and configured SSH connection with master node is server A. Installed Docker.
- Server C: K8S Node 1 configured.
- Created my CI/CD Pipeline in Jenkins and configured My Jenkins Configurations with Maven Tool.
- Created my Jenkinsfile for my CI/CD Pipeline
- Executed the Maven commands using Jenkinsfile – compile, test and package.

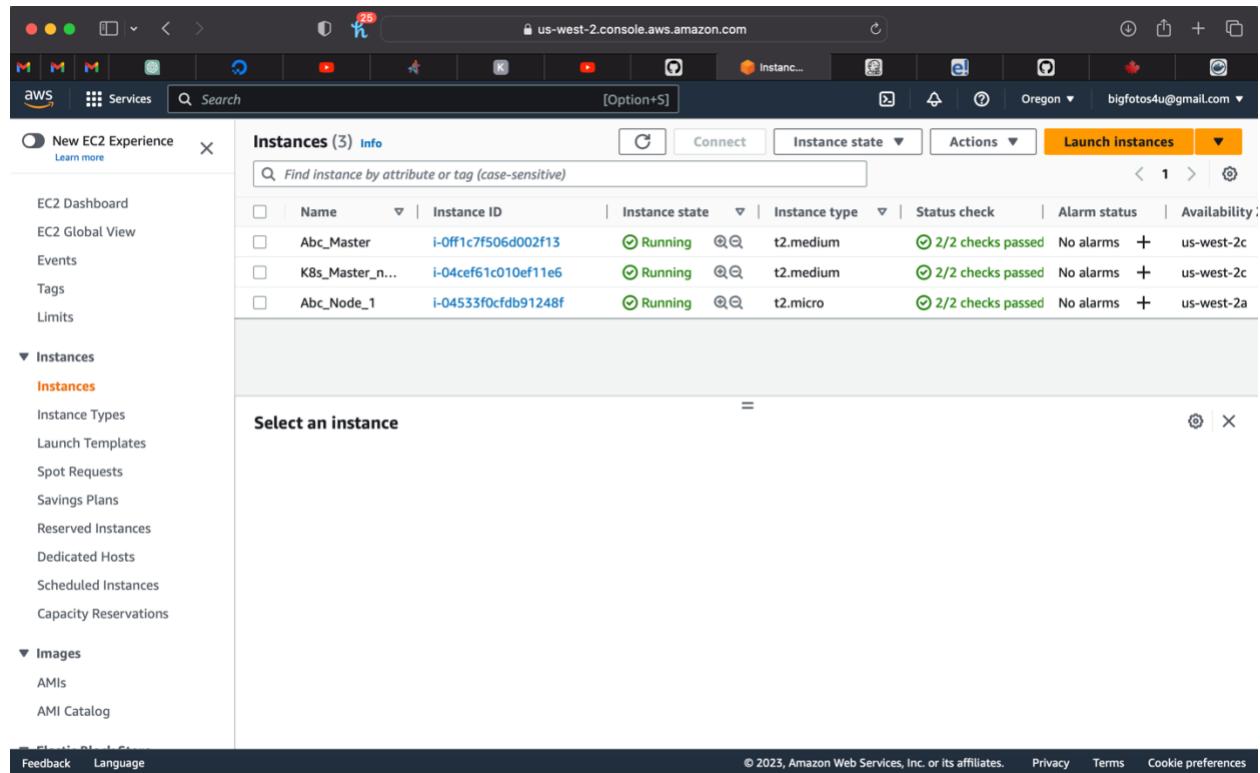
Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

- Configured my Pipeline Build triggers and GitHub Webhook.
- Configured Docker and Ansible in Jenkins Global Tool Configuration.
- Created a new Docker repo for my image in Dockerhub -
https://hub.docker.com/repository/docker/iambowcreek/abc_technologies/general
- Created my Dockerfile on my local machine using VSCode and pushed to my GitHub repo to trigger build.
- Updated my Jenkinsfile to build Docker images with Tomcat as the base image and push to Dockerhub
- Configured K8S cloud on Jenkins
- Created my Ansible Playbook to deploy ABC Technologies Artifacts to K8S – deployment and service
- Updated my Jenkinsfile to execute Ansible Playbook to deploy my containerised application to K8S server.
- Installed Prometheus and Grafana on Server B to monitor the Kubernetes cluster

Snapshots:

Provisioned my instances for the project using AWS – On the ABC_Master VM I used Ubuntu. I installed java, Jenkins, Docker, Ansible master and Maven. On K8s_Master_node VM, i used amazon linux and installed K8s control plane, Docker and used it as my Ansible node. Then on Abc_Node_1 VM, i used amazon linux as well. I used the vm as my K8s slave node.



The screenshot shows the AWS EC2 Instances page with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Abc_Master	i-0ff1c7f506d002f13	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
K8s_Master_n...	i-04cef61c010ef11e6	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
Abc_Node_1	i-04533fcfdb91248f	Running	t2.micro	2/2 checks passed	No alarms	us-west-2a

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Pushing my code – I already have my src on my local machine after downloading it from my LMS. I created a GitHub repo and pushed my src to the repo. Below is the history from my local machine.

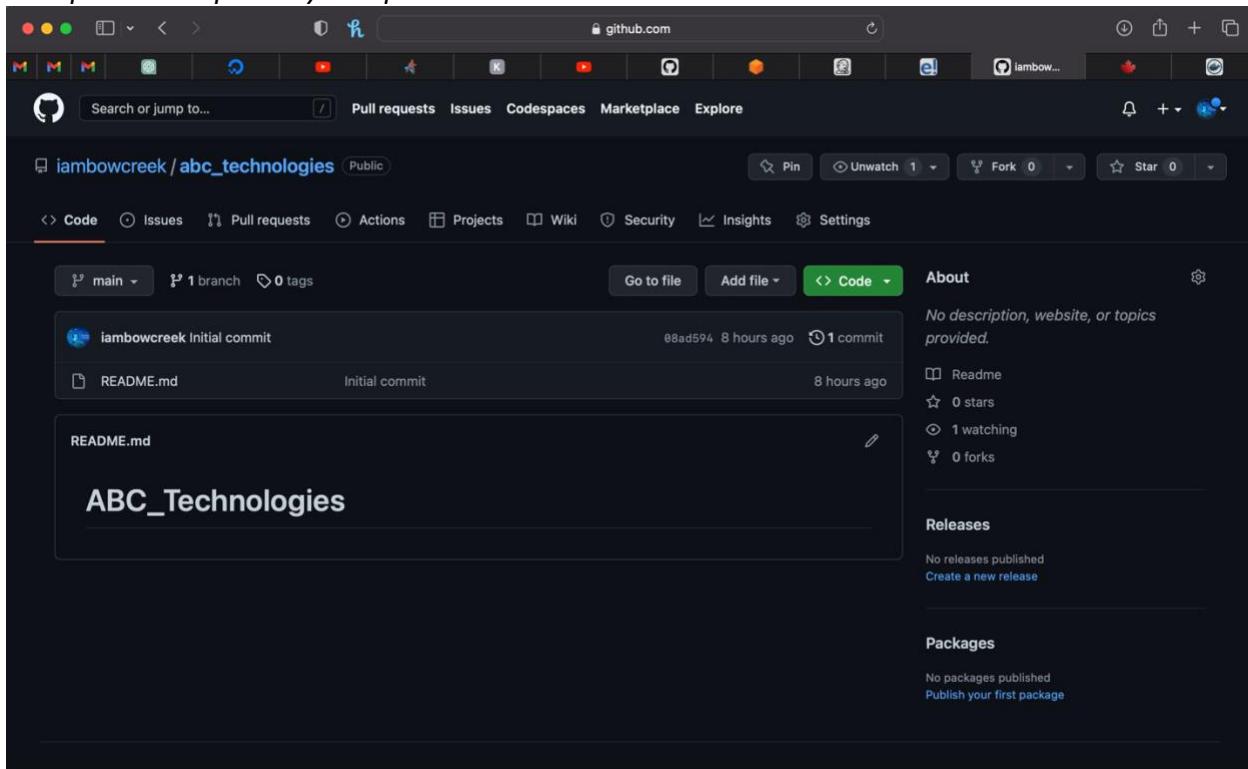


The screenshot shows a terminal window with the following content:

```
PROBLEMS (3) OUTPUT DEBUG CONSOLE TERMINAL COMMENTS + ... ^ X
Delta compression using up to 8 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (51/51), 8.85 KiB | 2.95 MiB/s, done.
Total 51 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/iambowcreek/abc_technologies.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
● bowcreek@Ofotos-MBP abc_technologies % history
 1 echo "# abc_technologies" >> README.md
 2 git init
 3 git add README.md
 4 git status
 5 git add .
 6 git commit -m "src first commit"
 7 git branch -M main
 8 git remote add origin https://github.com/iambowcreek/abc_technologies.git
 9 git push -u origin main
○ bowcreek@Ofotos-MBP abc_technologies %
```

Ln 38, Col 61 Spaces: 4 UTF-8 LF Groovy ⌂ ⌂

GitHub Repo – This is the repo I created on GitHub before pushing my src. Set up the Git repository and push the source code.



Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

The screenshot shows a GitHub repository page for 'iambowcreek/abc_technologies'. The repository is public and contains one branch ('main') and no tags. A single commit from 'iambowcreek' is listed, titled 'src first commit', made at 'now' by 'Sad7012'. The commit details show files like '.settings', 'src', 'target', '.DS_Store', '.classpath', '.project', 'README.md', 'pom.xml', and 'pom.xml.bak' all updated to 'src first commit' at 'now'. On the right side, there's an 'About' section with a note about no description, website, or topics provided, and sections for 'Readme', 'Releases', 'Packages', and 'Languages'.

Then, log in to Jenkins.

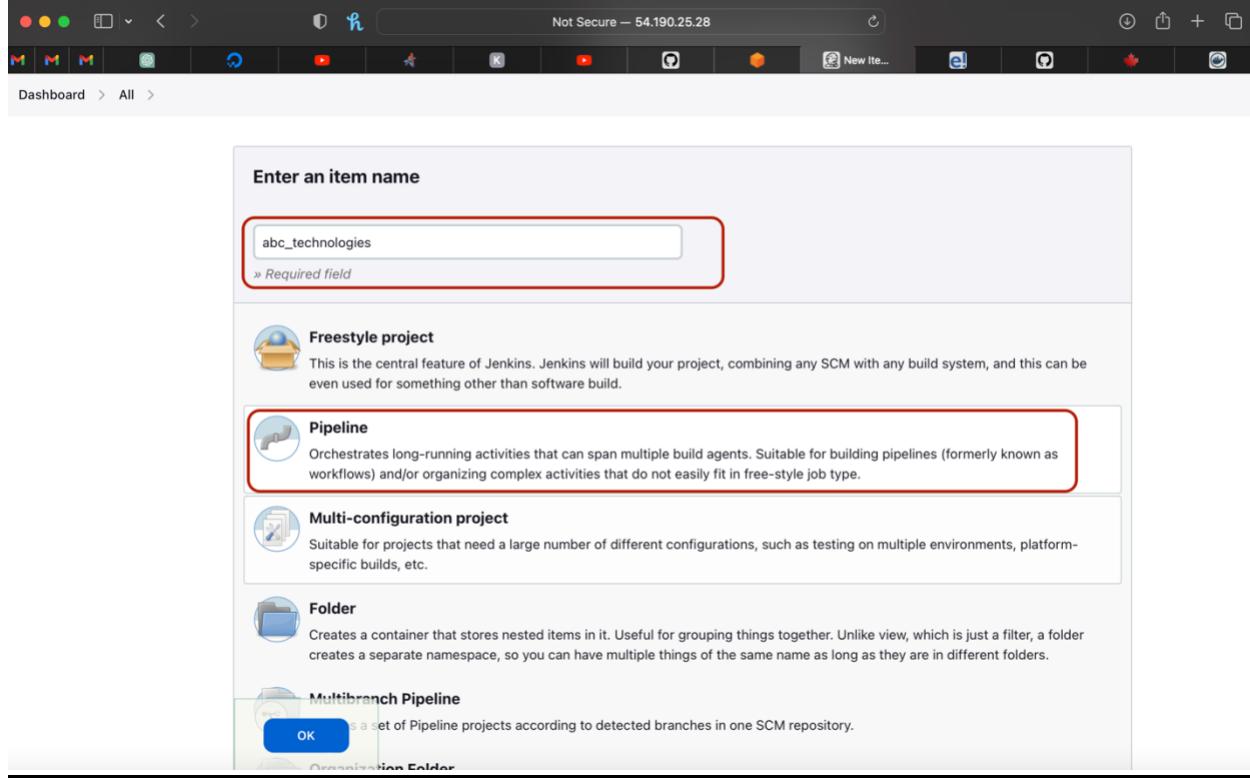
Created a build pipeline containing a job for each: compiling, testing and package

The screenshot shows the Jenkins login page. It features the Jenkins logo (a smiling figure holding a coffee cup) and the text 'Welcome to Jenkins!'. There are two input fields: one for 'Username' containing 'NelsonDevops' and another for 'Password' containing '*****'. Below the password field is a checkbox for 'Keep me signed in'. At the bottom is a blue 'Sign in' button.

Post Graduate Certification Program in DevOps

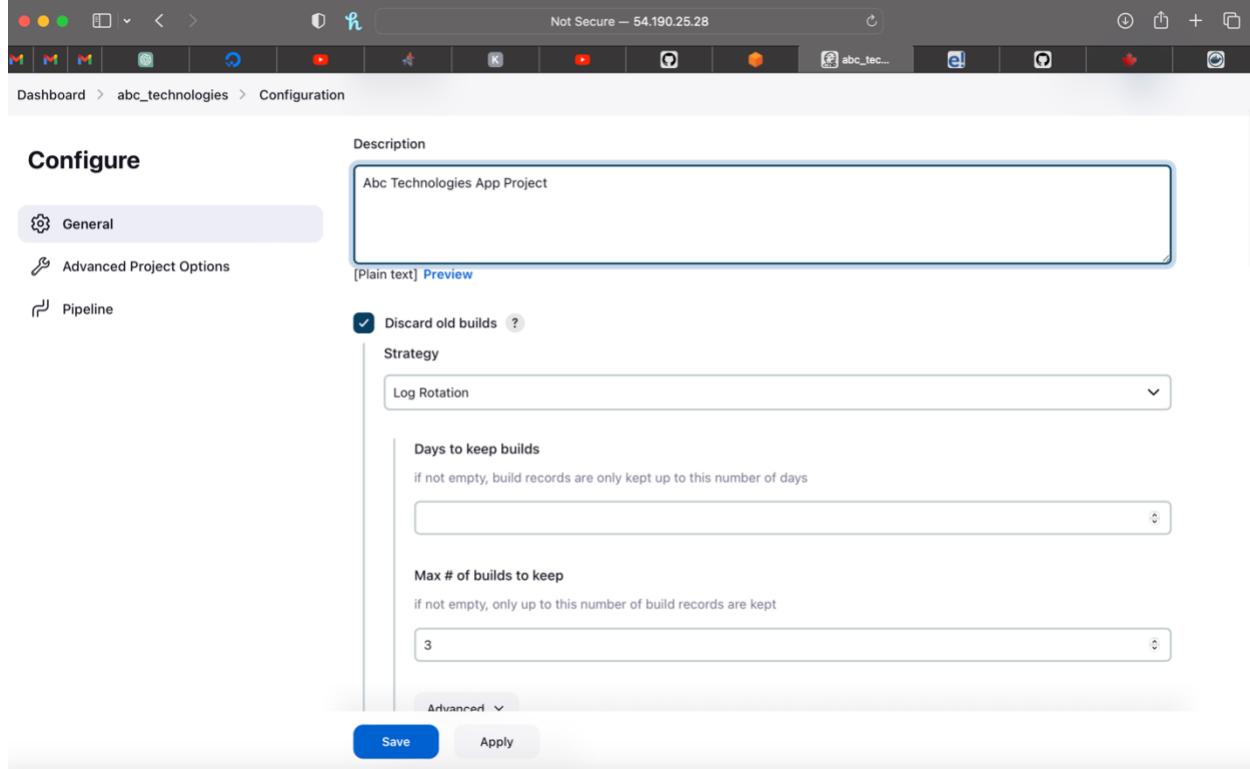
ABC-Technologies Edureka Project 1

Setting up my project pipeline in Jenkins using pipeline



The screenshot shows a web browser window with the URL 'Not Secure - 54.190.25.28'. The page title is 'Dashboard > All > Enter an item name'. A red box highlights the input field where 'abc_technologies' is typed. Another red box highlights the 'Pipeline' project type option, which is described as 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' Other options shown include 'Freestyle project', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'.

Configuring my pipeline before build

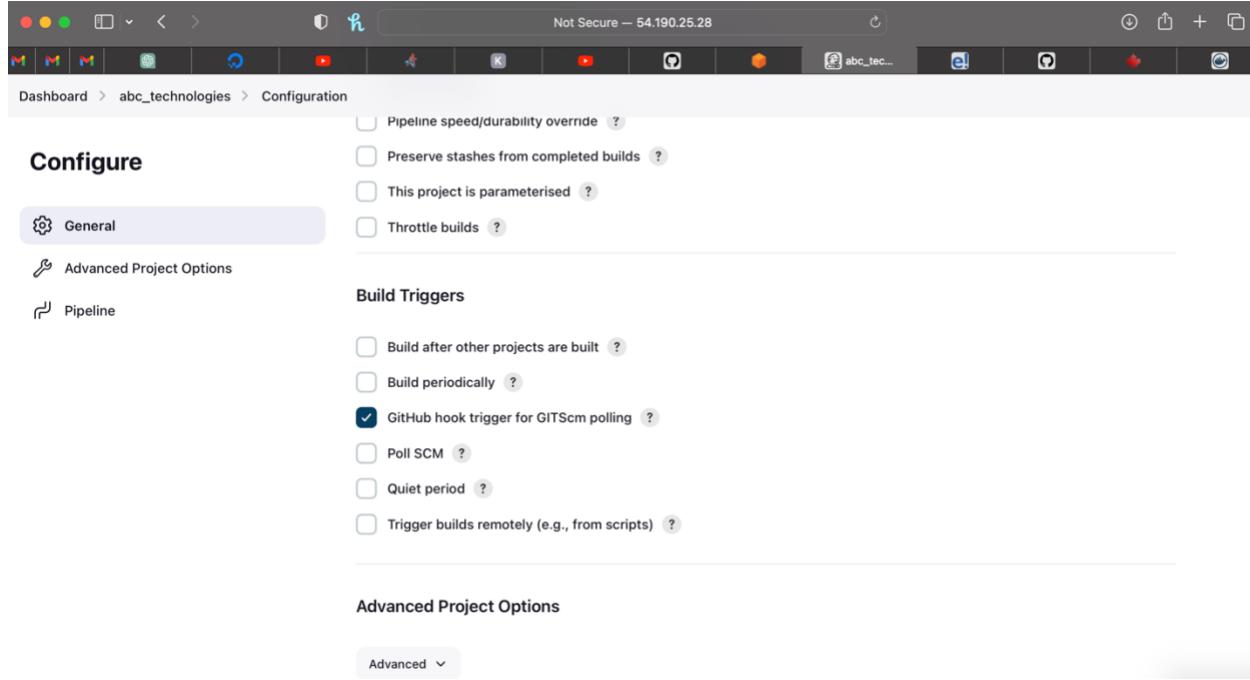


The screenshot shows a web browser window with the URL 'Not Secure - 54.190.25.28'. The page title is 'Dashboard > abc_technologies > Configuration'. On the left, there are tabs: 'Configure' (selected), 'General' (highlighted with a red box), 'Advanced Project Options', and 'Pipeline'. The 'General' tab has a 'Description' field containing 'Abc Technologies App Project'. Below it are 'Plain text' and 'Preview' buttons. The 'Pipeline' tab has a 'Discard old builds' checkbox (highlighted with a red box). Under 'Strategy', 'Log Rotation' is selected. In the 'Days to keep builds' section, there is a note: 'if not empty, build records are only kept up to this number of days' and an empty input field. In the 'Max # of builds to keep' section, there is a note: 'if not empty, only up to this number of build records are kept' and an input field containing '3'. At the bottom are 'Save' and 'Apply' buttons.

Post Graduate Certification Program in DevOps

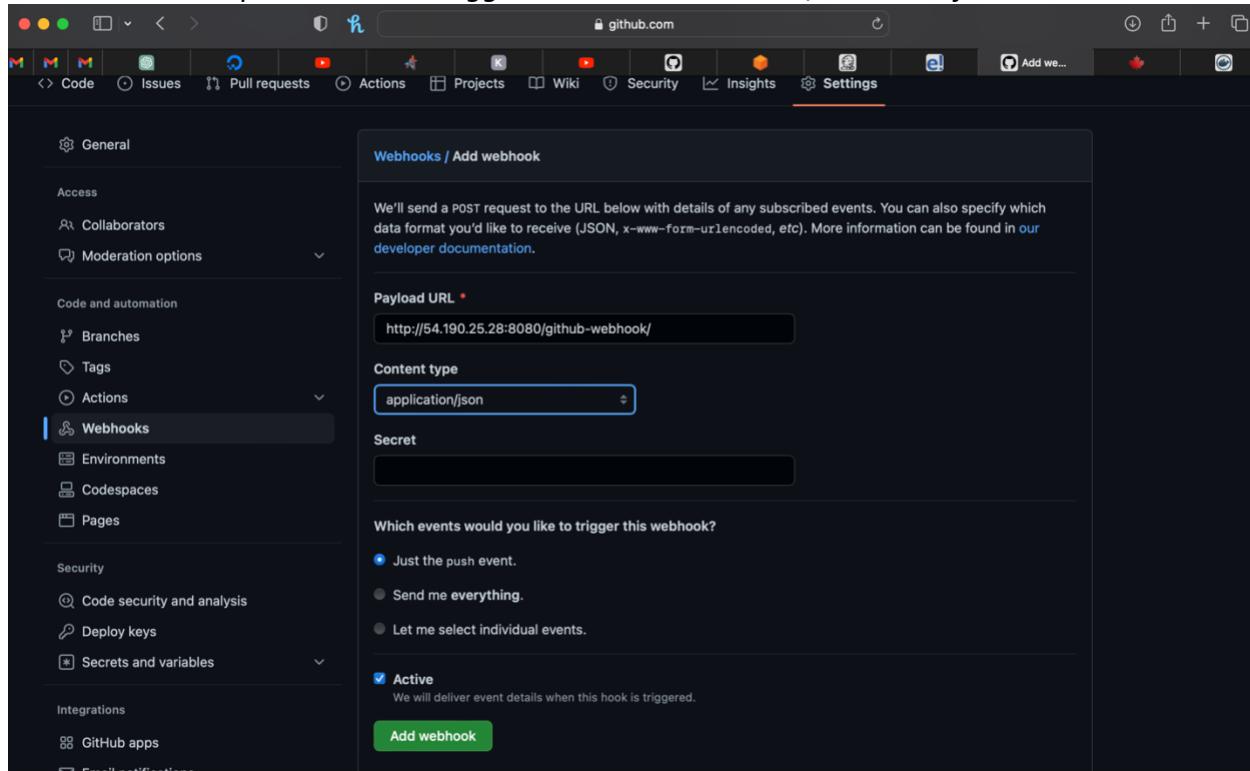
ABC-Technologies Edureka Project 1

Git webhook set up on Jenkins to trigger build once new codes/code modifications are done.



The screenshot shows the Jenkins 'Configure' screen for a project named 'abc_technologies'. Under the 'Build Triggers' section, the 'GitHub hook trigger for GITScm polling' option is checked. Other options like 'Build after other projects are built', 'Build periodically', 'Poll SCM', 'Quiet period', and 'Trigger builds remotely' are unchecked. The 'Advanced Project Options' tab is also visible.

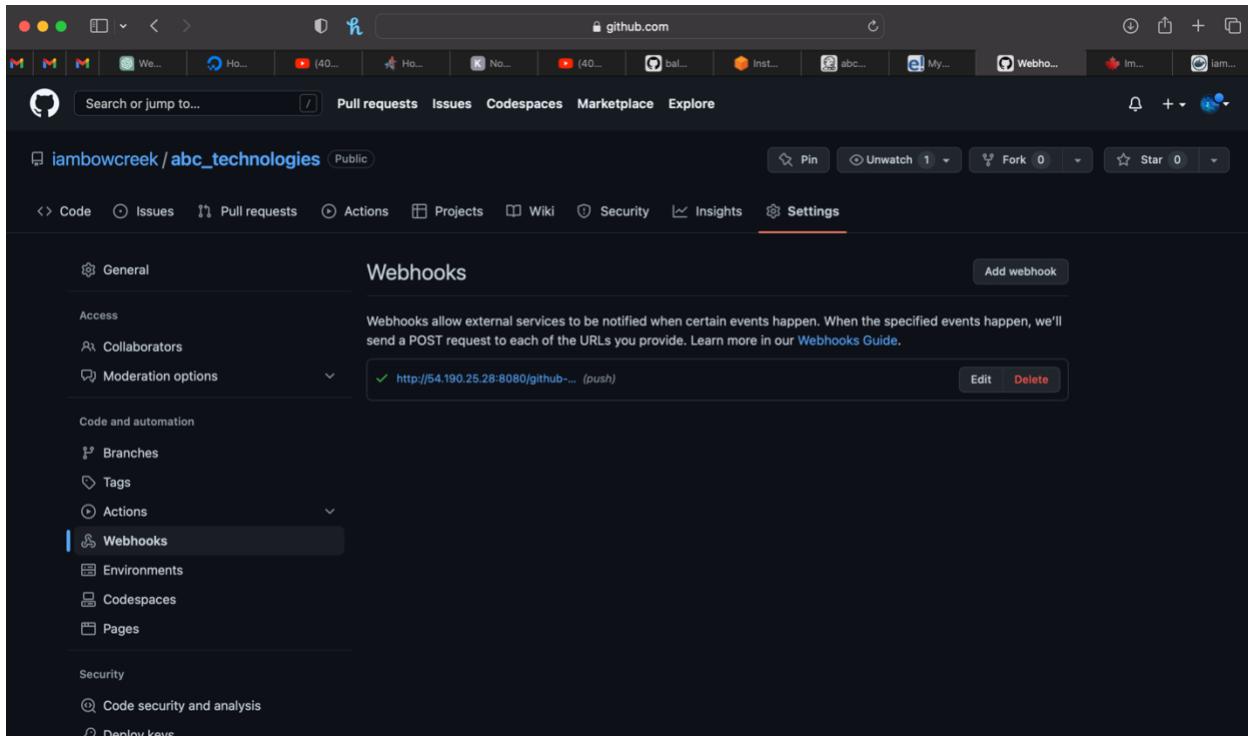
Git webhook set up on GitHub to trigger build once new codes/code modifications are done.



The screenshot shows the GitHub 'Webhooks / Add webhook' configuration page. The 'Payload URL' is set to 'http://54.190.25.28:8080/github-webhook/'. The 'Content type' is set to 'application/json'. The 'Secret' field is empty. Under 'Which events would you like to trigger this webhook?', the 'Just the push event' radio button is selected. The 'Active' checkbox is checked. A green 'Add webhook' button is at the bottom right.

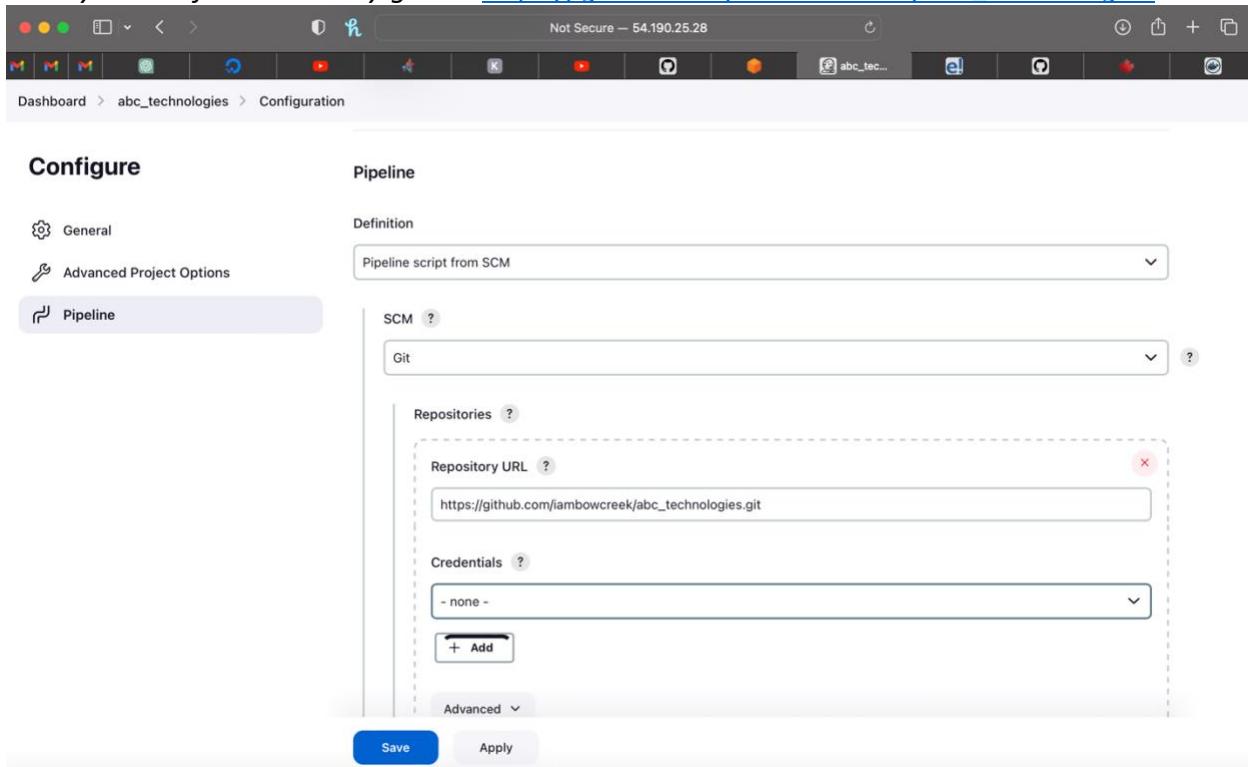
Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1



The screenshot shows the GitHub settings page for the repository `iambowcreek/abc_technologies`. The left sidebar is collapsed, and the main area is titled "Webhooks". A single webhook is listed with the URL `http://54.190.25.28:8080/github-webhook/push`. There are "Edit" and "Delete" buttons next to it. The right sidebar contains links for General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Actions, Webhooks), Environments, Codespaces, Pages, Security, Code security and analysis, and Deploy keys.

For my Jenkinsfile – Go to my github https://github.com/iambowcreek/abc_technologies



The screenshot shows the Jenkins Pipeline configuration page for the project `abc_technologies`. The left sidebar has tabs for "Configure" and "Pipeline", with "Pipeline" selected. The main area is titled "Definition" and shows "Pipeline script from SCM". Under the "SCM" tab, "Git" is selected. The "Repositories" section shows a single repository with the URL `https://github.com/iambowcreek/abc_technologies.git`. The "Credentials" dropdown is set to "- none -". At the bottom are "Save" and "Apply" buttons.

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Project Pipeline Dashboard ready

The screenshot shows the Jenkins Pipeline abc_technologies dashboard. On the left, there's a sidebar with various options: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, and GitHub Hook Log. The main area is titled "Pipeline abc_technologies" and "Abc Technologies App Project". It features a "Stage View" section with a message "No data available. This Pipeline has not yet run." and a "Permalinks" section. Below these is a "Build History" section with a dropdown menu set to "trend" and a search bar. The message "No builds" is displayed.

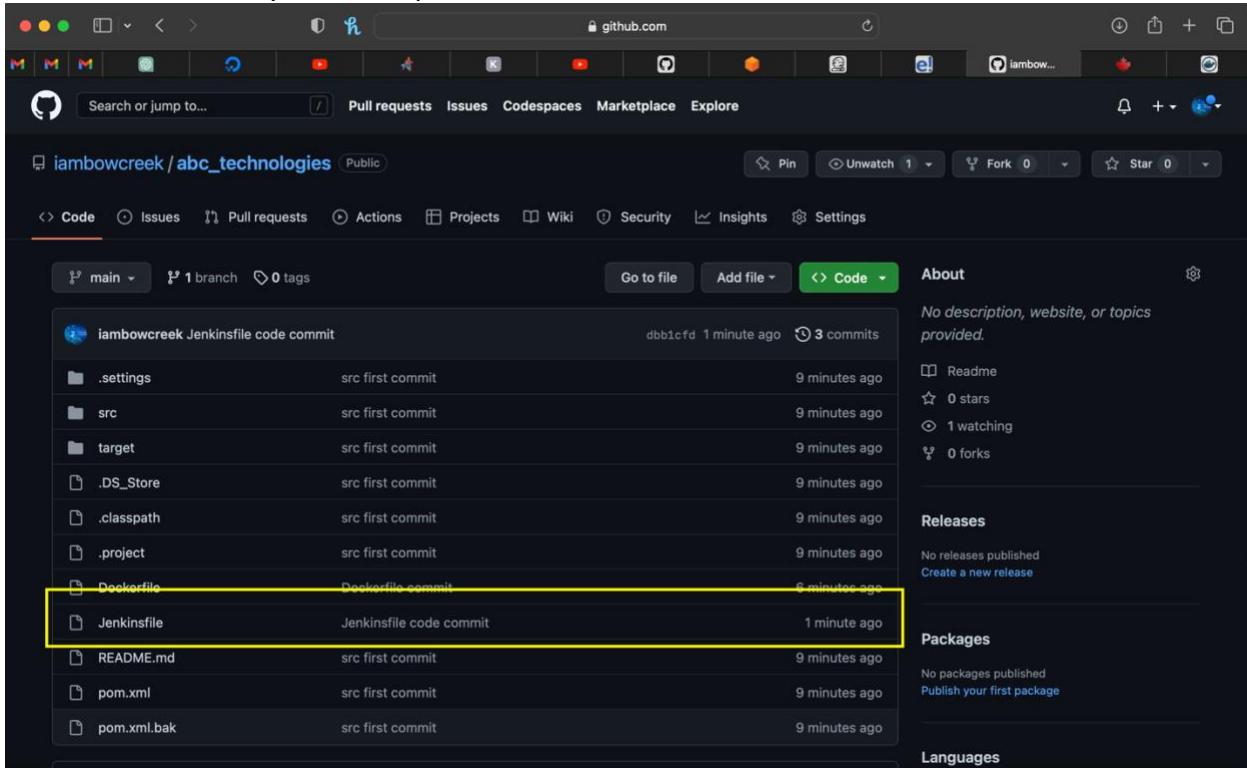
Maven Installation using Jenkins Global Tool Configuration

The screenshot shows the Jenkins Global Tool Configuration page for Maven. The URL is "Not Secure – 18.237.26.133". The sidebar shows "Dashboard > Manage Jenkins > Global Tool Configuration". The main content is titled "Maven" and shows "Maven installations". A sub-section titled "List of Maven installations on this system" contains a "Add Maven" button. Below it, there's a configuration for a Maven installation named "maven3.8.6". The configuration includes a checked checkbox for "Install automatically" and a dropdown menu for "Install from Apache" with "Version" set to "3.8.6". At the bottom are "Save" and "Apply" buttons.

Post Graduate Certification Program in DevOps

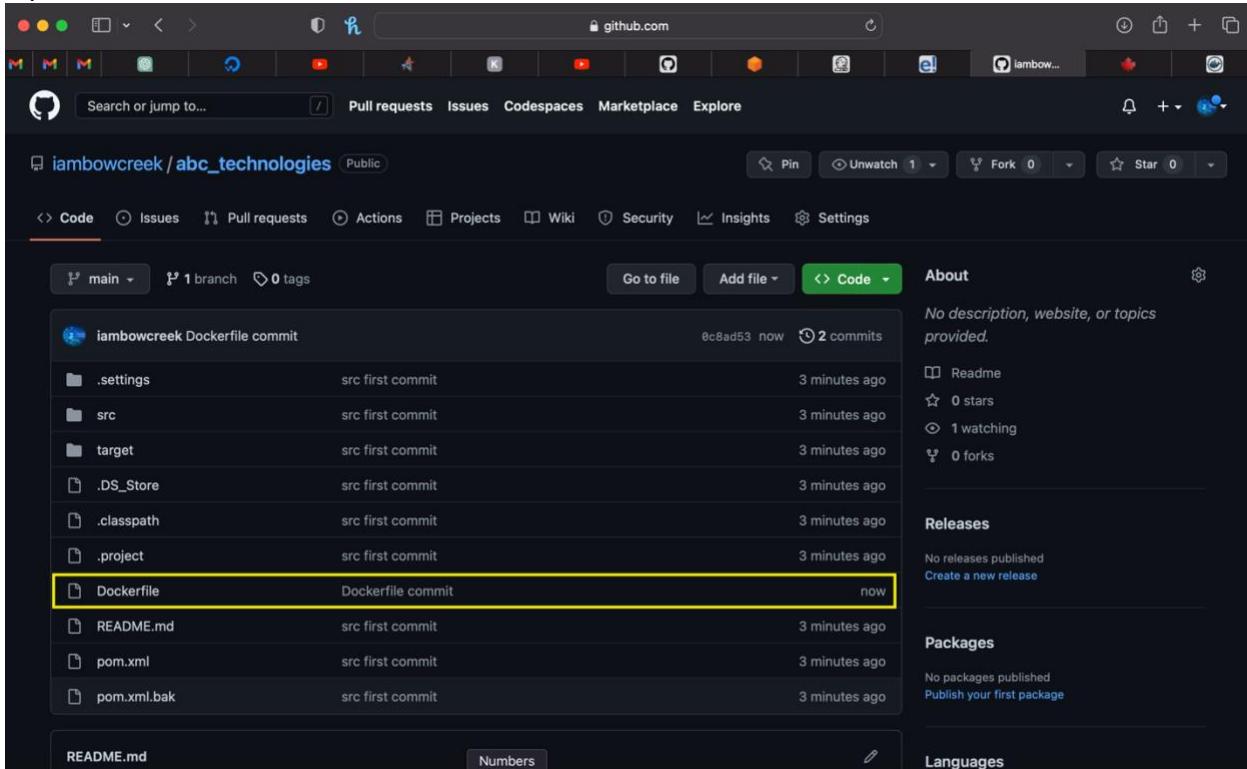
ABC-Technologies Edureka Project 1

Jenkinsfile to execute the maven commands: compile the src, test the src and package my src. Also committed in my GitHub repo.



A screenshot of a GitHub repository page for 'iambowcreek / abc_technologies'. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A yellow box highlights the 'Jenkinsfile' commit, which was made 1 minute ago by 'iambowcreek'. The commit message is 'Jenkinsfile code commit'. Other files listed include '.settings', 'src', 'target', '.DS_Store', '.classpath', '.project', 'Dockerfile', 'README.md', 'pom.xml', and 'pom.xml.bak', all with 'src first commit' messages and 9 minutes ago timestamps. The repository has 1 watch, 0 forks, and 0 stars. It also shows no releases or packages published.

Dockerfile to create an image of my artifact with tomcat image. Also committed in my GitHub repo.



A screenshot of the same GitHub repository page for 'iambowcreek / abc_technologies'. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A yellow box highlights the 'Dockerfile' commit, which was made now by 'iambowcreek'. The commit message is 'Dockerfile commit'. Other files listed include '.settings', 'src', 'target', '.DS_Store', '.classpath', '.project', 'README.md', 'pom.xml', and 'pom.xml.bak', all with 'src first commit' messages and 3 minutes ago timestamps. The repository has 1 watch, 0 forks, and 0 stars. It shows no releases or packages published.

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Docker repo created to store my application image when pushed.

The screenshot shows the Docker Hub interface for the repository `iambowcreek/abc_technologies`. The repository details page is displayed, showing a description of "Edureka Project 1" and a timestamp of "Last pushed: a few seconds ago". A "Docker commands" section contains the command `docker push iambowcreek/abc_technologies:tagname`. Below this, there are sections for "Tags" (empty) and "Automated Builds" (disabled). A "Public View" button is visible in the top right corner.

Project Pipeline Build – git clone, mvn compile, mvn test, mvn package, docker image build and image pushed to dockerhub

The screenshot shows the Jenkins Pipeline status page for the project `abc_technologies`. The pipeline stages are listed as follows:

Stage	Description	Time
Declarative: Checkout SCM	835ms	835ms
Declarative: Tool Install	178ms	178ms
1. Git Clone	684ms	684ms
2. Compile SRC	4s	4s
3. Test SRC	4s	4s
4. Build Package	6s	6s
5. Docker Build Package	2s	2s
6. Push To Dockerhub	8s	8s

Average stage times: (Average full run time: ~33s)

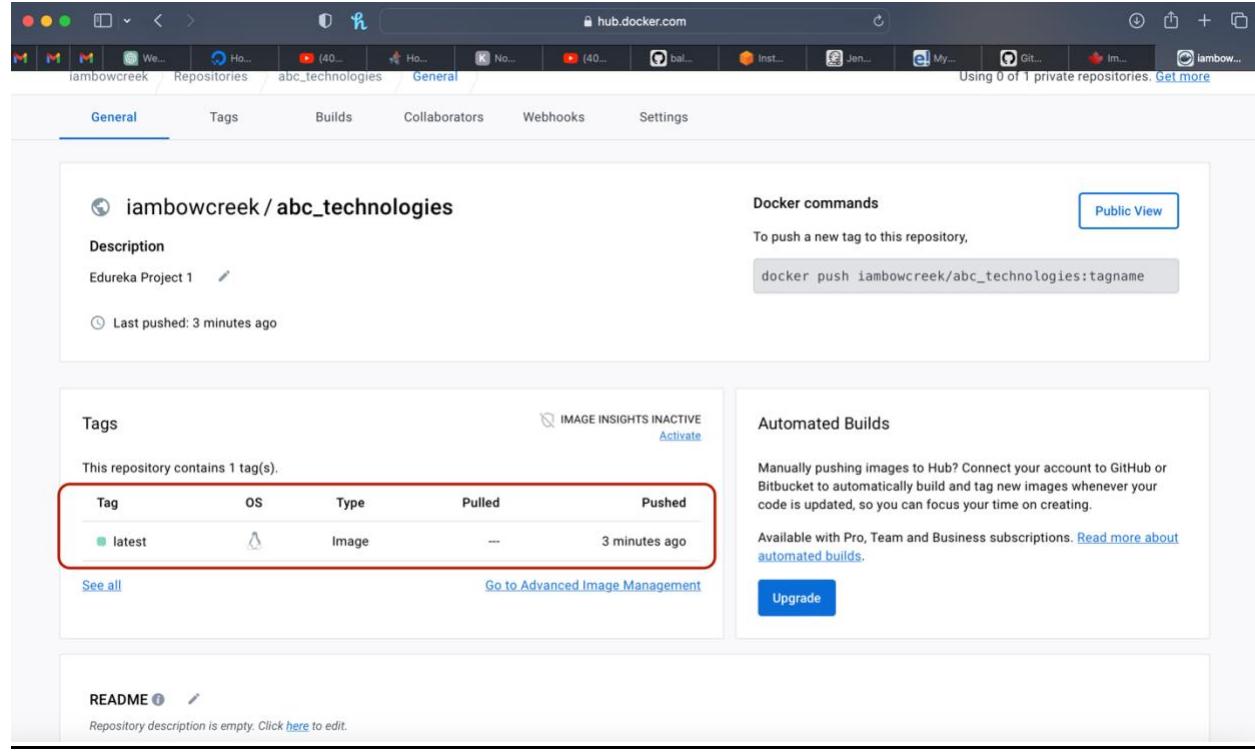
Build History: trend, Mar 19, 08:29, No Changes

Permalinks: Atom feed for all, Atom feed for failures

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

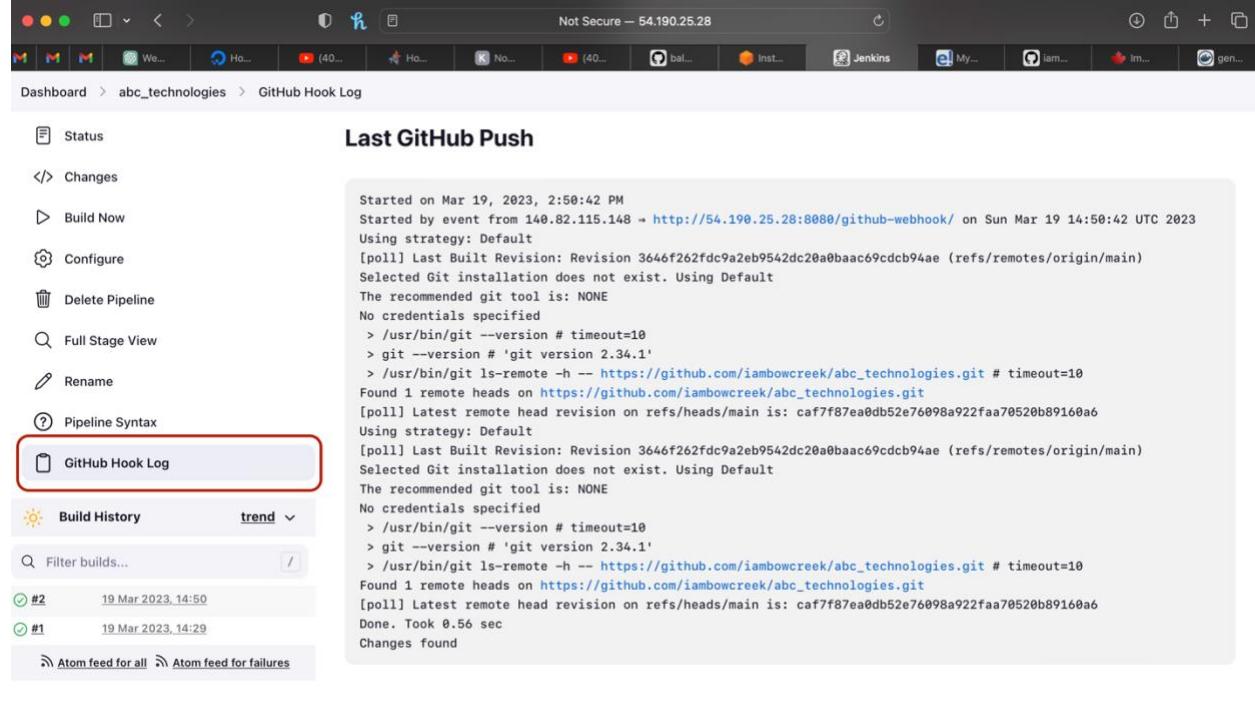
Application image pushed to dockerhub with tag:latest



The screenshot shows the DockerHub interface for the repository `iambowcreek/abc_technologies`. The General tab is selected. The repository details include:

- Description: Edureka Project 1
- Last pushed: 3 minutes ago
- Docker commands: `docker push iambowcreek/abc_technologies:tagname`
- Public View button

The Tags section shows one tag, `latest`, which was pushed 3 minutes ago. The Automated Builds section indicates that manually pushing images to Hub? is inactive. The README section notes that the repository description is empty.



The screenshot shows the Jenkins GitHub Hook Log for the project `abc_technologies`. The Last GitHub Push log is displayed, showing the following output:

```
Started on Mar 19, 2023, 2:50:42 PM
Started by event from 140.82.115.148 → http://54.190.25.28:8080/github-webhook/ on Sun Mar 19 14:50:42 UTC 2023
Using strategy: Default
[poll] Last Built Revision: Revision 3646f262fdc9a2eb9542dc20a0baac69cdcb94ae (refs/remotes/origin/main)
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> /usr/bin/git --version # timeout=10
> git --version # 'git' version 2.34.1'
> /usr/bin/git ls-remote -h -- https://github.com/iambowcreek/abc_technologies.git # timeout=10
Found 1 remote heads on https://github.com/iambowcreek/abc_technologies.git
[poll] Latest remote head revision on refs/heads/main is: caf7f87ea0db52e76098a922faa70520b89160a6
Using strategy: Default
[poll] Last Built Revision: Revision 3646f262fdc9a2eb9542dc20a0baac69cdcb94ae (refs/remotes/origin/main)
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> /usr/bin/git --version # timeout=10
> git --version # 'git' version 2.34.1'
> /usr/bin/git ls-remote -h -- https://github.com/iambowcreek/abc_technologies.git # timeout=10
Found 1 remote heads on https://github.com/iambowcreek/abc_technologies.git
[poll] Latest remote head revision on refs/heads/main is: caf7f87ea0db52e76098a922faa70520b89160a6
Done. Took 0.56 sec
Changes found
```

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Full console output for my CI/CD Pipeline:

mvn compile

The screenshot shows the Jenkins interface with the URL "Not Secure - 52.35.251.190". The navigation bar includes "Dashboard", "abc_technologies", "#4", "Shell Script", and "Console Output". The main area is titled "Console Output" with a green checkmark icon. A red box highlights the command "+ mvn compile". The log output shows the Maven build process, starting with "[INFO] Scanning for projects..." and ending with "[INFO] BUILD SUCCESS". The entire log concludes with "[INFO] Total time: 2.236 s" and "[INFO] Finished at: 2023-03-19T15:01:44Z".

```
+ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.abc:ABCtechnologies >-----
[INFO] Building RetailModule 1.0
[INFO] -----[ war ]-----
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:prepare-agent (jacoco-initialize) @ ABCtechnologies ---
[INFO] argline set to -
javaagent:/var/lib/jenkins/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-
runtime.jar=destfile=/var/lib/jenkins/workspace/abc_technologies@2/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ABCtechnologies ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/abc_technologies@2/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ABCtechnologies ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to /var/lib/jenkins/workspace/abc_technologies@2/target/classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
```

The screenshot shows the Jenkins interface with the URL "Not Secure - 52.35.251.190". The navigation bar includes "abc_tec...", "Stage Logs (2. Compile SRC)", and "x". The main area is titled "Stage Logs (2. Compile SRC)" with a red box highlighting the "Shell Script -- mvn compile" step. The log output is identical to the one in the previous screenshot, showing the Maven build process and concluding with "[INFO] BUILD SUCCESS". The log ends with "[INFO] Total time: 2.236 s" and "[INFO] Finished at: 2023-03-19T15:01:44Z". At the bottom, there are links for "Atom feed for all" and "Atom feed for failures", and a timestamp "Mar 19 09:01". A timeline at the bottom shows various stages: commit (516ms), 71ms, 484ms, 3s, 4s, 5s, 1s, and 7s.

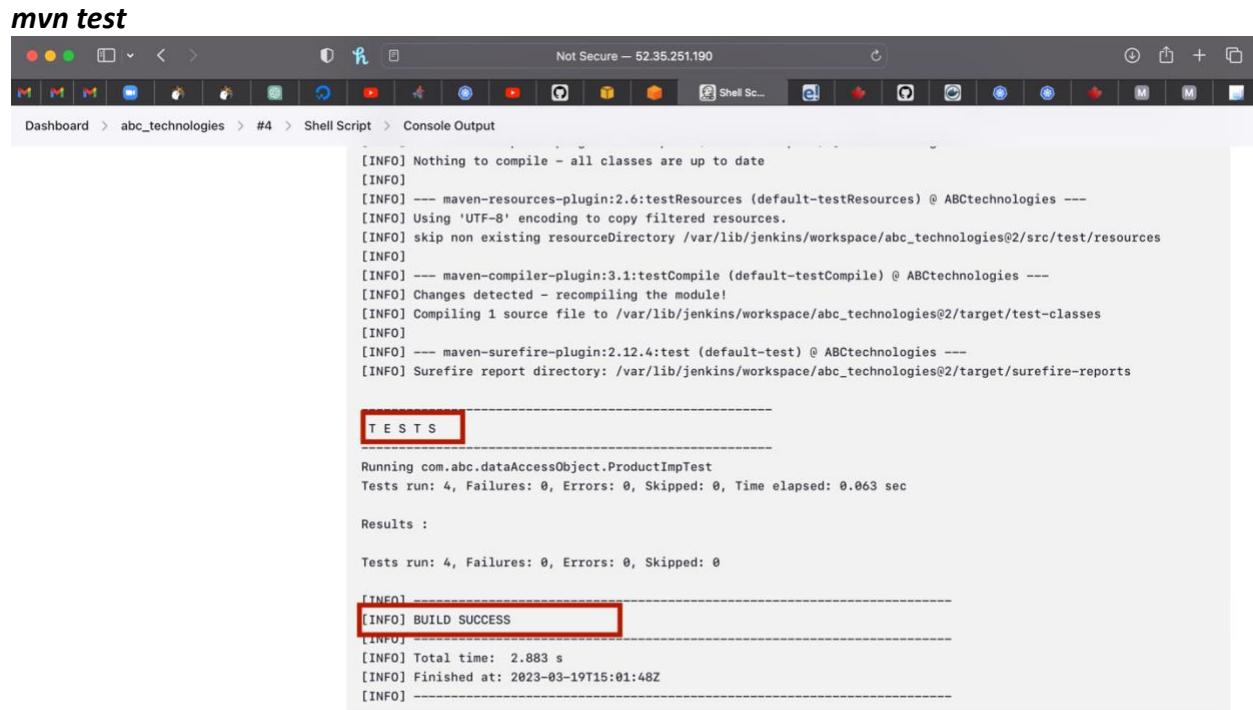
```
Stage Logs (2. Compile SRC)
x
@ Use a tool from a predefined Tool Installation -- maven3.8.6 (self time 24ms)
@ Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 29ms)
@ Shell Script -- mvn compile (self time 3s)

[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:prepare-agent (jacoco-initialize) @ ABCtechnologies ---
[INFO] argLine set to -javaagent:/var/lib/jenkins/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/var/lib/jenkins/workspace/abc_technologies@2/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ABCtechnologies ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/abc_technologies@2/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ABCtechnologies ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to /var/lib/jenkins/workspace/abc_technologies@2/target/classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
```

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

mvn test

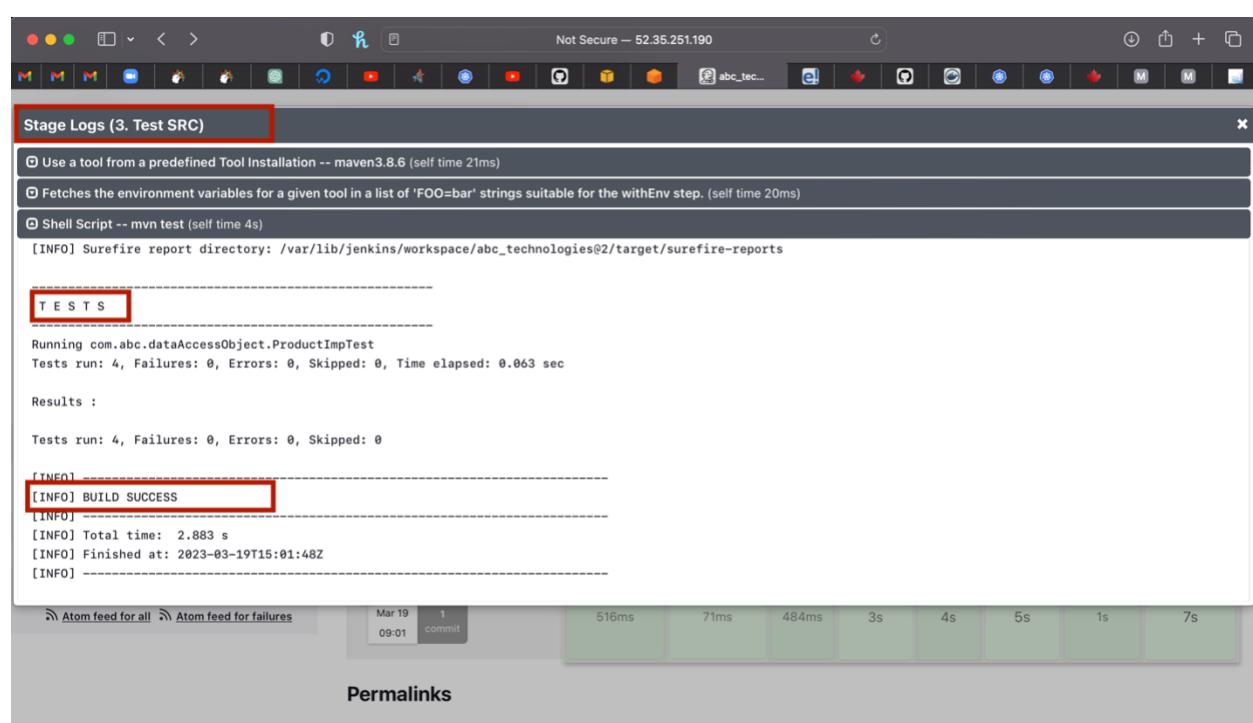


The screenshot shows the Jenkins console output for a Maven test run. The output indicates that there was nothing to compile, and the build was successful. The test results show 4 tests run, 0 failures, 0 errors, and 0 skipped. The total time taken was 2.883 seconds.

```
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ABCtechnologies ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/abc_technologies@2/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ABCtechnologies ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/abc_technologies@2/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ABCtechnologies ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/abc_technologies@2/target/surefire-reports

-----  
TESTS  
-----  
Running com.abc.dataAccessObject.ProductImpTest  
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.063 sec  
  
Results :  
  
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0  
  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 2.883 s  
[INFO] Finished at: 2023-03-19T15:01:48Z  
[INFO] -----
```

Stage Logs (3. Test SRC)



The screenshot shows the Jenkins stage logs for the '3. Test SRC' step. It lists three steps: 'Use a tool from a predefined Tool Installation -- maven3.8.6', 'Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.', and 'Shell Script -- mvn test'. The 'Shell Script' step shows the same Maven test results as the previous screenshot.

```
[INFO] Surefire report directory: /var/lib/jenkins/workspace/abc_technologies@2/target/surefire-reports

-----  
TESTS  
-----  
Running com.abc.dataAccessObject.ProductImpTest  
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.063 sec  
  
Results :  
  
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0  
  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 2.883 s  
[INFO] Finished at: 2023-03-19T15:01:48Z  
[INFO] -----
```

Mar 19 09:01 1 commit

516ms 71ms 484ms 3s 4s 5s 1s 7s

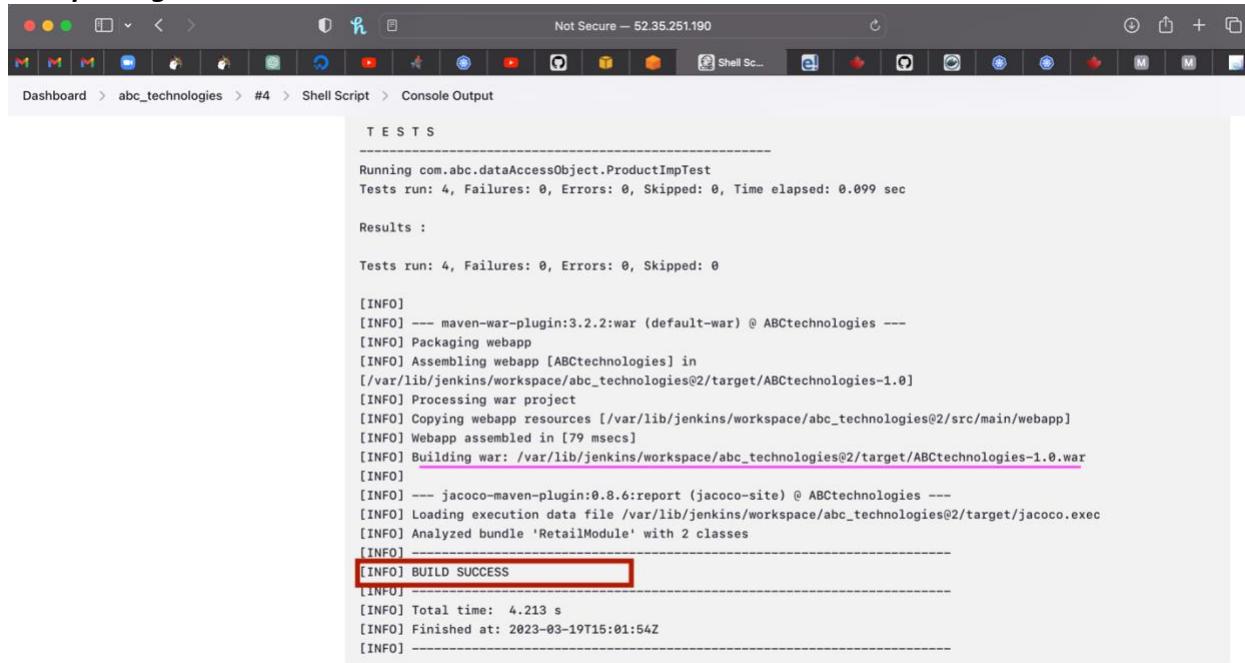
Atom feed for all Atom feed for failures

Permalinks

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

mvn package:



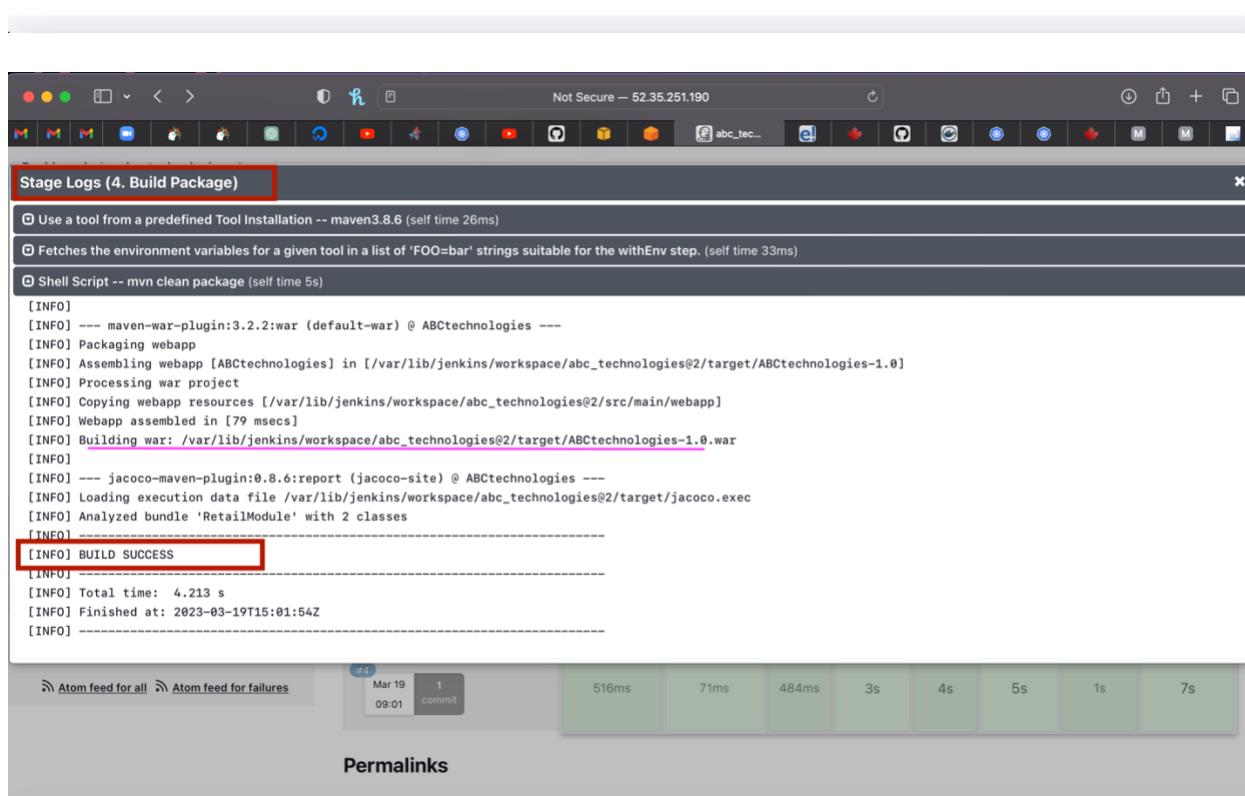
The screenshot shows a Jenkins job console output for a build step named "Shell Script". The output displays the results of a Maven package command. It includes test results, build logs, and a prominent "BUILD SUCCESS" message highlighted with a red box.

```
TESTS
-----
Running com.abc.dataAccessObject.ProductImpTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.099 sec

Results :

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ ABCtechnologies ---
[INFO] Packaging webapp
[INFO] Assembling webapp [ABCtechnologies] in [/var/lib/jenkins/workspace/abc_technologies@2/target/ABCtechnologies-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/abc_technologies@2/src/main/webapp]
[INFO] Webapp assembled in [79 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/abc_technologies@2/target/ABCtechnologies-1.0.war
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:report (jacoco-site) @ ABCtechnologies ---
[INFO] Loading execution data file /var/lib/jenkins/workspace/abc_technologies@2/target/jacoco.exec
[INFO] Analyzed bundle 'RetailModule' with 2 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.213 s
[INFO] Finished at: 2023-03-19T15:01:54Z
[INFO]
```



The screenshot shows the "Stage Logs" section for the "Build Package" step of a Jenkins job. It lists the steps taken: tool installation, environment variable fetching, and the execution of a shell script containing the Maven package command. The "BUILD SUCCESS" message is again highlighted with a red box.

Stage Logs (4. Build Package)

- Use a tool from a predefined Tool Installation -- maven3.8.6 (self time 26ms)
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 33ms)
- Shell Script -- mvn clean package (self time 5s)

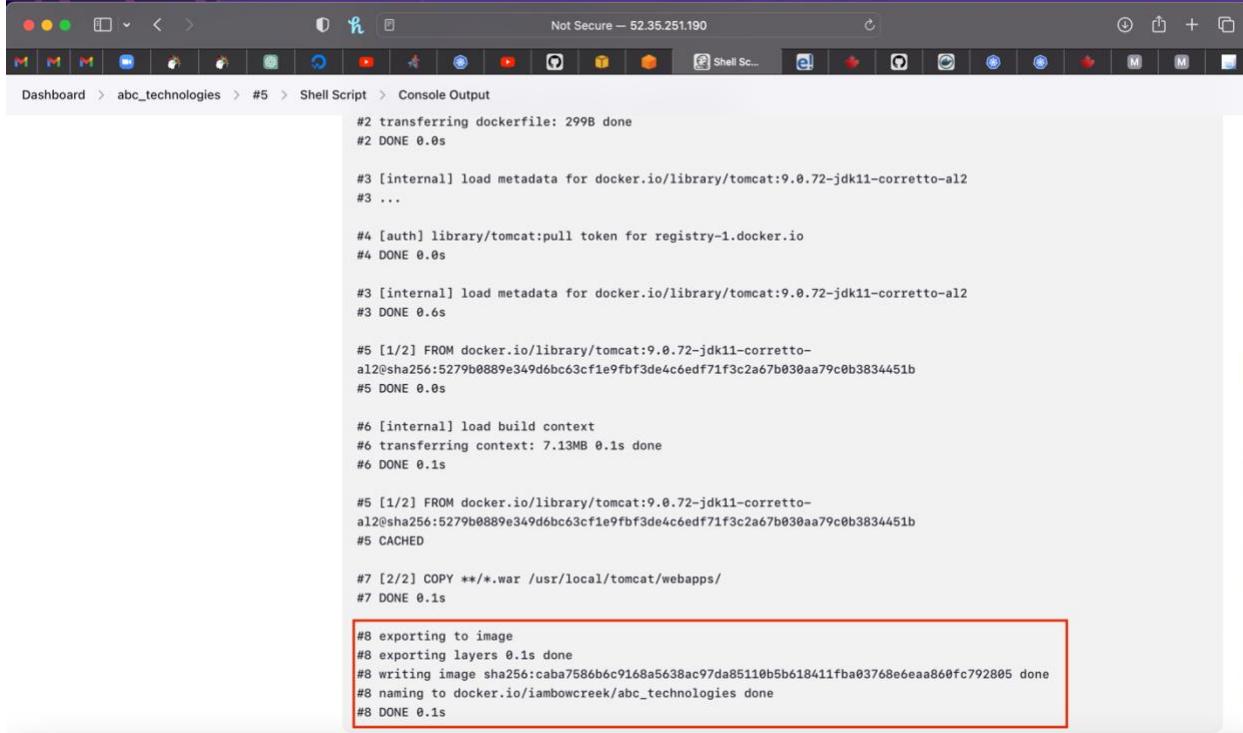
```
[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ ABCtechnologies ---
[INFO] Packaging webapp
[INFO] Assembling webapp [ABCtechnologies] in [/var/lib/jenkins/workspace/abc_technologies@2/target/ABCtechnologies-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/abc_technologies@2/src/main/webapp]
[INFO] Webapp assembled in [79 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/abc_technologies@2/target/ABCtechnologies-1.0.war
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:report (jacoco-site) @ ABCtechnologies ---
[INFO] Loading execution data file /var/lib/jenkins/workspace/abc_technologies@2/target/jacoco.exec
[INFO] Analyzed bundle 'RetailModule' with 2 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.213 s
[INFO] Finished at: 2023-03-19T15:01:54Z
[INFO]
```

Permalinks

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Docker build app image:



```
#2 transferring dockerfile: 299B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/tomcat:9.0.72-jdk11-corretto-a12
#3 ...

#4 [auth] library/tomcat:pull token for registry-1.docker.io
#4 DONE 0.0s

#3 [internal] load metadata for docker.io/library/tomcat:9.0.72-jdk11-corretto-a12
#3 DONE 0.6s

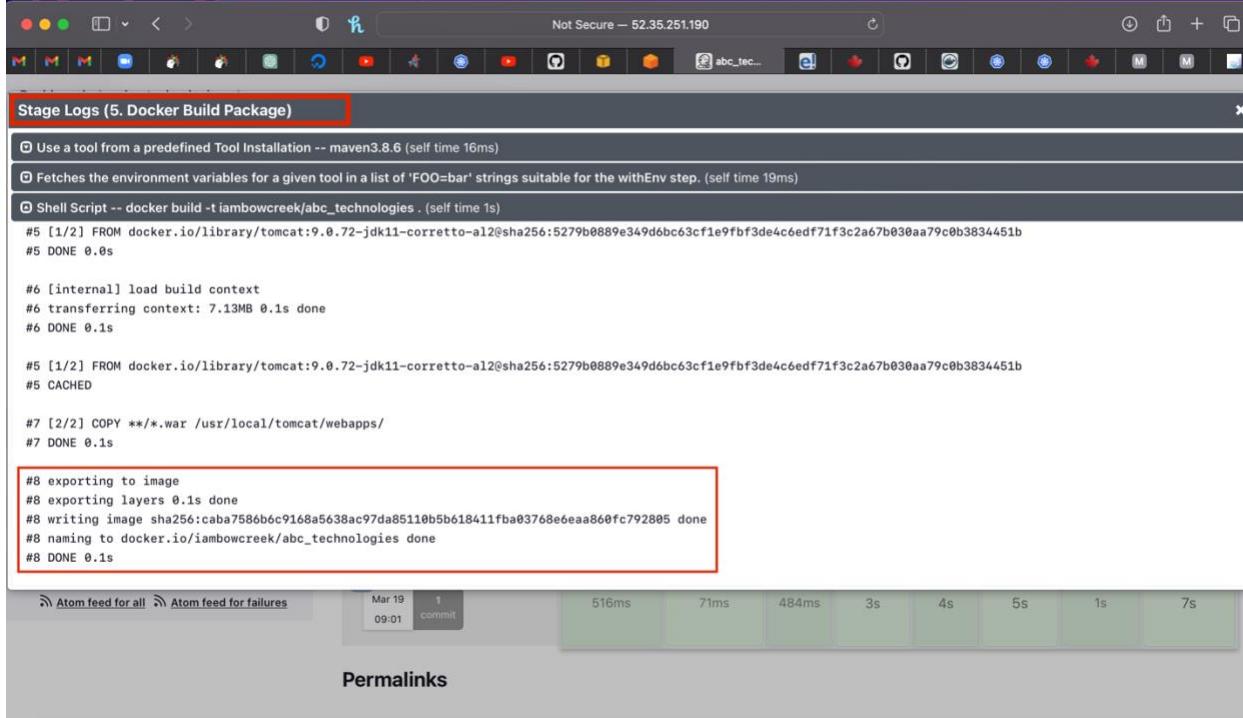
#5 [1/2] FROM docker.io/library/tomcat:9.0.72-jdk11-corretto-
a12@sha256:5279b0889e349d6bc63cf1e9fbf3de4c6edf71f3c2a67b030aa79c0b3834451b
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 7.13MB 0.1s done
#6 DONE 0.1s

#5 [1/2] FROM docker.io/library/tomcat:9.0.72-jdk11-corretto-
a12@sha256:5279b0889e349d6bc63cf1e9fbf3de4c6edf71f3c2a67b030aa79c0b3834451b
#5 CACHED

#7 [2/2] COPY **/*.war /usr/local/tomcat/webapps/
#7 DONE 0.1s

#8 exporting to image
#8 exporting layers 0.1s done
#8 writing image sha256:caba7586b6c9168a5638ac97da85110b5b618411fba03768e6eaa860fc792805 done
#8 naming to docker.io/iambowcreek/abc_technologies done
#8 DONE 0.1s
```



Stage Logs (5. Docker Build Package)

```
Use a tool from a predefined Tool Installation -- maven3.8.6 (self time 16ms)
Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 19ms)
Shell Script -- docker build -t iambowcreek/abc_technologies . (self time 1s)
#5 [1/2] FROM docker.io/library/tomcat:9.0.72-jdk11-corretto-a12@sha256:5279b0889e349d6bc63cf1e9fbf3de4c6edf71f3c2a67b030aa79c0b3834451b
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 7.13MB 0.1s done
#6 DONE 0.1s

#5 [1/2] FROM docker.io/library/tomcat:9.0.72-jdk11-corretto-a12@sha256:5279b0889e349d6bc63cf1e9fbf3de4c6edf71f3c2a67b030aa79c0b3834451b
#5 CACHED

#7 [2/2] COPY **/*.war /usr/local/tomcat/webapps/
#7 DONE 0.1s

#8 exporting to image
#8 exporting layers 0.1s done
#8 writing image sha256:caba7586b6c9168a5638ac97da85110b5b618411fba03768e6eaa860fc792805 done
#8 naming to docker.io/iambowcreek/abc_technologies done
#8 DONE 0.1s
```

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Docker image push to Dockerhub:

The screenshot shows a Jenkins job log titled "Stage Logs (6. Push To Dockerhub)". It displays several log entries:

- Use a tool from a predefined Tool Installation -- maven3.8.6 (self time 21ms)
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 29ms)
- Shell Script -- echo \${DOCKERHUB_CREDENTIALS_PSW} | docker login -u iambowcreek --password-stdin (self time 826ms)

Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.
Affected argument(s) used the following variable(s): [DOCKERHUB_CREDENTIALS_PSW]
See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

```
+ echo ****  
+ docker login -u iambowcreek --password-stdin  
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.  
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

Login Succeeded

Shell Script -- docker push iambowcreek/abc_technologies:latest (self time 6s)

Permalinks

The screenshot shows a Jenkins job log titled "Stage Logs (6. Push To Dockerhub)". It displays several log entries:

- Use a tool from a predefined Tool Installation -- maven3.8.6 (self time 21ms)
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 29ms)
- Shell Script -- echo \${DOCKERHUB_CREDENTIALS_PSW} | docker login -u iambowcreek --password-stdin (self time 826ms)
- Shell Script -- docker push iambowcreek/abc_technologies:latest (self time 6s)

```
+ docker push iambowcreek/abc_technologies:latest  
The push refers to repository [docker.io/iambowcreek/abc_technologies]  
eb4e9e6b3ca9: Preparing  
89cd10063f49: Preparing  
61b10eee8b9a6: Preparing  
7d8ea14a9eb4: Preparing  
d67c000345c6: Preparing  
941b8b0a5987: Preparing  
941b8b0a5987: Waiting  
7d8ea14a9eb4: Layer already exists  
61b10eee8b9a6: Layer already exists  
d67c000345c6: Layer already exists  
89cd10063f49: Layer already exists  
941b8b0a5987: Layer already exists  
eb4e9e6b3ca9: Pushed  
latest: digest: sha256:c9e7b989f52d279dc04bee66b7e92ea2f934ac1e3ebcd972828a64313002120e size: 1579
```

Steps in the configuration of server A and server B to enable ansible deploy the application into the Kubernetes cluster:

Step 1: I Integrated Ansible with Docker by installing Docker SDK on Server A. This will enable interaction with Docker API using Python. I executed the following commands:

- *sudo apt-get install python3-pip*
- *pip install docker*

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

To verify the installation:

- `python -c "import docker; print(docker.from_env().version())"`

Step 2: I integrated Ansible with Kubernetes (Server B) so that Ansible can deploy the application into the K8S cluster.

From the Ansible controller, I configured Ansible to use the Kubernetes API by setting the following environment variables:

- `export K8S_AUTH_API_KEY=<API key>`
- `export K8S_AUTH_HOST=<Kubernetes API server hostname>`
- `export K8S_AUTH_VERIFY_SSL=false`

To obtain the API key and hostname from the K8S control master node, use the following commands:

API key:

- `kubectl get secrets -n kube-system`
- `kubectl get secret service-account-controller-token-zb8qh -n kube-system -o jsonpath='{.data.token}' | base64 -d`

Hostname:

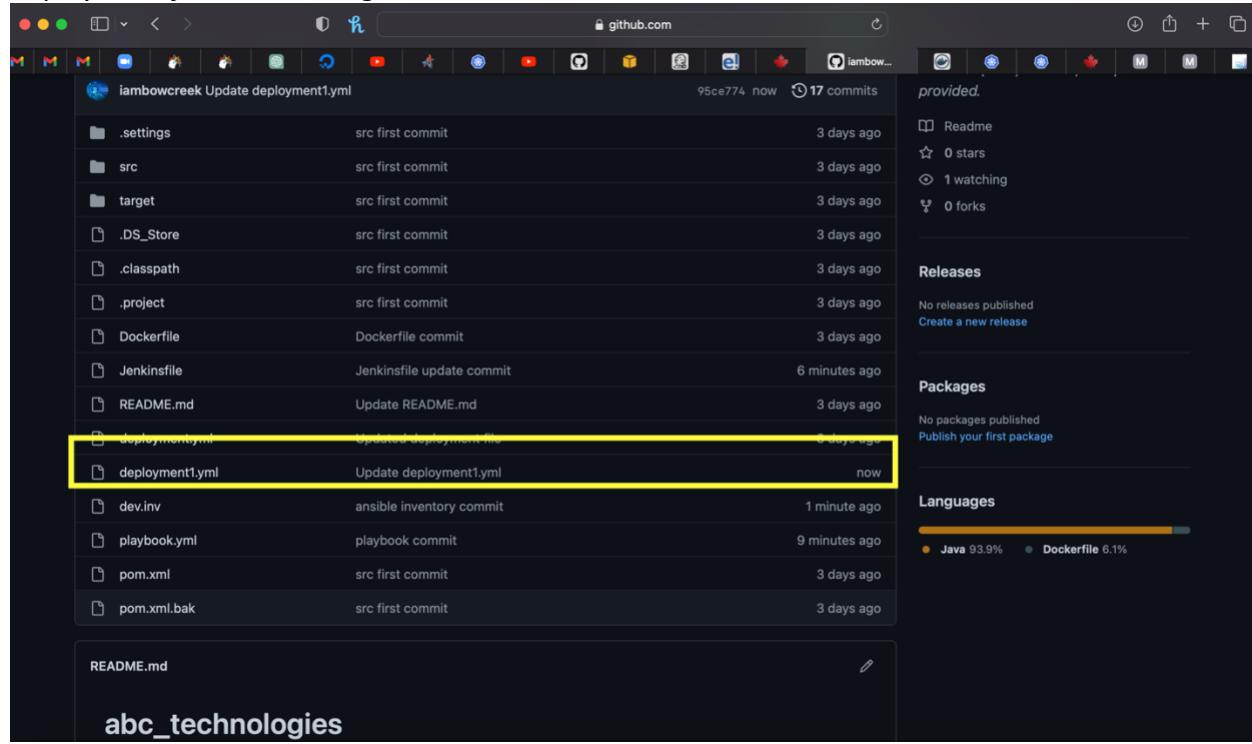
- `kubectl config view --minify --flatten -o jsonpath='{.clusters[].cluster.server}'`

Step 3: I copied the deployment file to the Kubernetes node by adding a task to the Ansible playbook to copy the deployment file to the Kubernetes node using the "copy" module and then deployed the deployment file in Kubernetes cluster by adding a task to the Ansible playbook to deploy the deployment file in Kubernetes using the "k8s" module. ([See deployment file in github](#)) After the deployment in k8s, the was running on both k8s master node and slave node. See screenshots below:

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Deployment file commit to github



I also updated my Jenkinsfile to call my Playbook to copy and deploy my deployment file in k8s:

The screenshot shows a Java project structure in an IDE. The left sidebar lists files: .settings, src, target, .classpath, .project, Dockerfile, Jenkinsfile (highlighted with a yellow box), pom.xml, pom.xml.bak, and README.md. The right pane displays the Jenkinsfile content:

```
1 pipeline{
2     agent any
3     tools {
4         maven "maven3.8.6"
5     }
6     environment{
7         DOCKERHUB_CREDENTIALS = credentials("DOCKERHUB_CRED")
8     }
9     stages {
10        stage ("1. Git Clone") {
11            steps {
12                git branch: 'main', url: 'https://github.com/iambowcreek/edu_abc_tech.git'
13            }
14        }
15    }
16}
```

The terminal window shows the output of a git push command:

```
create mode 100644 target/classes/com/abc/dataAccessObject/RetailDataImp.class
create mode 100644 target/test-classes/.DS_Store
create mode 100644 target/test-classes/com/.DS_Store
create mode 100644 target/test-classes/com/abc/dataAccessObject/ProductImpTest.class
bowcreek@Ofotos-MBP abc_technologies % git branch -M main
bowcreek@Ofotos-MBP abc_technologies % git remote add origin https://github.com/iambowcreek/abc_technologies.git
bowcreek@Ofotos-MBP abc_technologies % git push -u origin main
Enumerating objects: 51, done.
Counting objects: 100% (51/51), done.
Delta compression using up to 8 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (51/51), 8.85 KiB | 2.95 MiB/s, done.
Total 51 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/iambowcreek/abc_technologies.git
 * [new branch]      main    -> main
branch 'main' set up to track 'origin/main'.
bowcreek@Ofotos-MBP abc_technologies %
```

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

A screenshot of a GitHub repository page for 'iambowcreek'. The repository has 1 branch and 0 tags. The 'Code' tab is selected. A yellow box highlights the 'Jenkinsfile' commit, which was updated at 'now' and contains the commit message 'Jenkinsfile update commit'. Other commits listed include '.settings', 'src', 'target', '.DS_Store', '.classpath', '.project', 'Dockerfile', 'README.md', 'deployment.yml', 'deployment1.yml', 'playbook.yml', 'pom.xml', and 'pom.xml.bak', all of which were first committed 3 days ago.

A screenshot of a Jenkinsfile in a GitHub repository. The file content is as follows:

```
12     git branch: 'main', url: 'https://github.com/iambowcreek/abc_technologies.git'
13   }
14   stage('2. Compile SRC'){
15     steps{
16       sh "mvn compile"
17     }
18   }
19   stage('3. Test SRC'){
20     steps{
21       sh "mvn test"
22     }
23   }
24   stage('4. Build Package'){
25     steps{
26       sh "mvn clean package"
27     }
28   }
29   stage('5. Docker Build Package'){
30     steps{
31       sh "docker build -t iambowcreek/abc_technologies ."
32     }
33   }
34   stage("6. Push To Dockerhub"){
35     steps{
36       sh "$DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin"
37       sh "docker push iambowcreek/abc_technologies:latest"
38     }
39   }
40   stage("7. Execute to Ansible"){
41     steps{
42       ansiblePlaybook credentialsId: 'ansible_sshopen', installation: 'ansible', inventory: 'dev.inv', playbook: 'playbook.yml'
43     }
44   }
45 }
46 }
47 }
```

A yellow box highlights the 'Execute to Ansible' stage (lines 40-47), which uses the 'ansiblePlaybook' step to run an Ansible playbook named 'playbook.yml'.

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

A screenshot of a GitHub repository page for 'abc_technologies'. The repository has 95 commits and 17 pushes. The 'deployment1.yml' file is highlighted with a yellow box. Other files shown include '.settings', 'src', 'target', '.DS_Store', '.classpath', '.project', 'Dockerfile', 'Jenkinsfile', 'README.md', 'deployment.yml', 'dev.inv', 'playbook.yml', 'pom.xml', and 'pom.xml.bak'. The 'playbook.yml' file was committed 9 minutes ago. The repository has 0 stars, 1 watching, and 0 forks.

Playbook.yml file

A screenshot of the GitHub code editor for the 'playbook.yml' file. The file contains 12 lines of YAML code. The code defines a task to copy a Kubernetes deployment file from a Jenkins workspace to a temporary directory, and then applies it using kubectl. The file was committed by 'iambowcreek' last month.

```
1 ---  
2 - hosts: localhost  
3   gather_facts: false  
4   tasks:  
5     - name: Copy Kubernetes deployment file  
6       copy:  
7         src: /var/lib/jenkins/workspace/abc_technologies/deployment1.yml  
8         dest: /tmp/  
9     - name: Apply Kubernetes deployment  
10       become: true  
11       become_user: root  
12       command: kubectl apply -f /tmp/deployment1.yml
```

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

K8S Master Node Port

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store. The main area shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Abc_Master	i-0ff1c7f506d002f13	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
K8s_Master_n...	i-04cef61c010ef11e6	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
Abc_Node_1	i-04533f0cdb91248f	Running	t2.micro	2/2 checks passed	No alarms	us-west-2a

Below the table, the details for the selected instance (i-04cef61c010ef11e6) are shown:

Instance: i-04cef61c010ef11e6 (K8s_Master_node)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID i-04cef61c010ef11e6 (K8s_Master_node)	Public IPv4 address 35.91.250.8 open address	Private IPv4 addresses 172.31.2.125
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-35-91-250-8.us-west-2.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-2-125.us-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-2-125.us-west-2.compute.internal	Elastic IP addresses
Answer private resource DNS name	Instance type	

K8S Master Node Port App run

The screenshot shows a web browser window. The address bar says "Not Secure — 35.91.250.8". The page content is:

Welcome to ABC technologies

This is retail portal

[Add Product](#) [View Product](#)

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Node Port IP address

The screenshot shows the AWS EC2 Instances page. There are three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Abc_Master	i-0ff1c7f506d002f13	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
K8s_Master_n...	i-04cef61c010ef11e6	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
Abc_Node_1	i-04533f0cfdb91248f	Running	t2.micro	2/2 checks passed	No alarms	us-west-2a

On the right, the details for the selected instance (Abc_Node_1) are shown:

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance summary						
Instance ID i-04533f0cfdb91248f (Abc_Node_1)	Public IPv4 address 54.188.28.234 open address	Private IPv4 addresses 172.31.31.245				
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-188-28-234.us-west-2.compute.amazonaws.com open address				
Hostname type IP name: ip-172-31-31-245.us-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-31-245.us-west-2.compute.internal	Instance type	Elastic IP addresses			
Answer private resource DNS name						

Node Port App Run

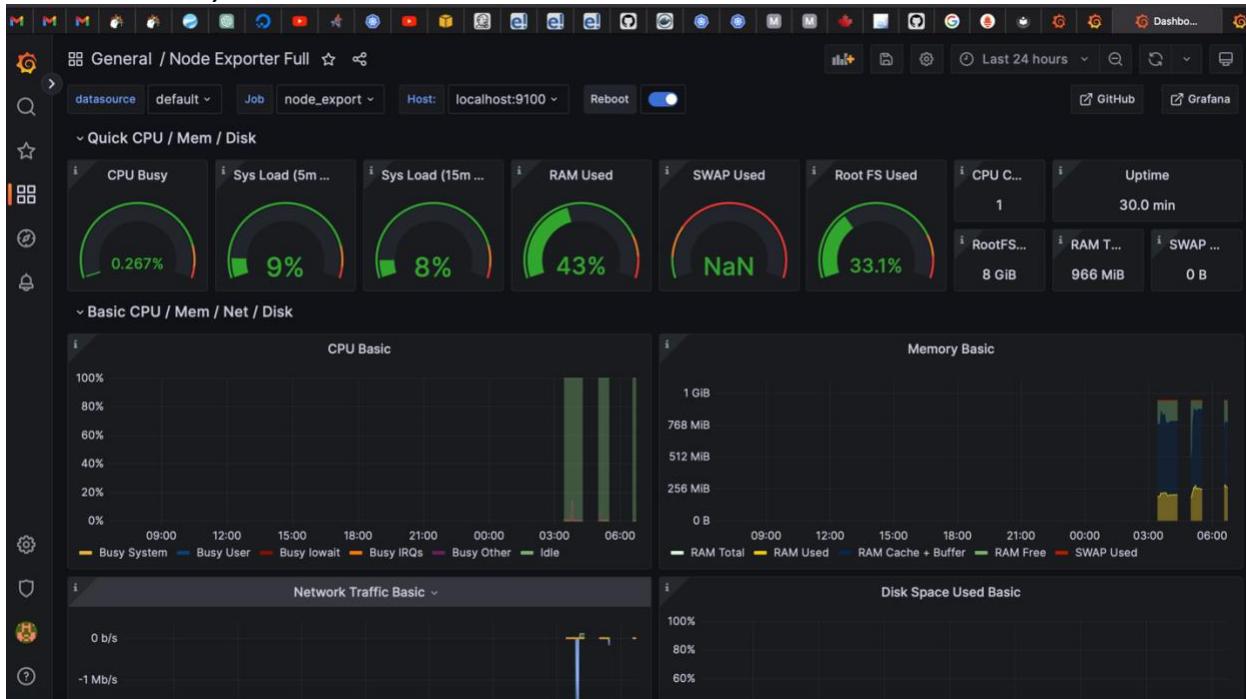
The screenshot shows a web browser window. The address bar displays "Not Secure — 54.188.28.234". The page title is "Welcome to ABC technologies". The main content area contains the text "This is retail portal".

Post Graduate Certification Program in DevOps

ABC-Technologies Edureka Project 1

Step 4: I used the following steps below to install Prometheus and Grafana on my K8s Master node to monitor my Kubernetes cluster (see manifest files on github)

1. Update the package manager on your Amazon Linux instance using the command `sudo yum update`.
2. Install the Prometheus package by running the command `sudo yum install prometheus`.
3. After the installation completes, you can start the Prometheus service using the command `sudo systemctl start prometheus`.
4. To enable Prometheus to start automatically when the server boots up, run the command `sudo systemctl enable prometheus`.
5. Verify that Prometheus is running by accessing its web interface at `http://<your-server-ip>:9090` using a web browser.
6. Now, you can install Grafana by adding the Grafana repository to the package manager using the command `sudo rpm -Uvh https://dl.grafana.com/oss/release/grafana-7.5.10-1.x86_64.rpm`.
7. After adding the repository, install Grafana using the command `sudo yum install grafana`.
8. Start the Grafana service by running the command `sudo systemctl start grafana-server`.
9. To enable Grafana to start automatically when the server boots up, run the command `sudo systemctl enable grafana-server`.
10. Verify that Grafana is running by accessing its web interface at `http://<your-server-ip>:3000` using a web browser.
11. Configure Grafana to use Prometheus as a data source by adding a new data source in the Grafana web interface and specifying the URL for Prometheus, which should be `http://localhost:9090`.
12. Once the data source is configured, you can create dashboards in Grafana to visualize the data collected by Prometheus.



Post Graduate Certification Program in DevOps
ABC-Technologies Edureka Project 1