

**Bapatla Engineering College :: Bapatla
(Autonomous)**
Department of Computer Science & Engineering

DevOps – Lab Record
(Lab code : 20CSL701)
IV/IV B.Tech
VII SEMESTER
2025-2026



Name : Are Suresh Reddy
Regd No. : Y22ACS414

**Bapatla Engineering College :: Bapatla
(Autonomous)**
(Affiliated To Acharya Nagarjuna University)
Bapatla – 522102, A.P
www.becbapatla.ac.in

**Bapatla Engineering College :: Bapatla
(Autonomous)**
Department of Computer Science & Engineering



CERTIFICATE

Register Number: Y22ACS414

Certified and Bonafide Record work done by Are Suresh Reddy (Computer Science Engineering) 7th Semester in “DevOps Lab (20CSL701)” during the academic year 2025 – 2026. Number of experiments recorded is 9

Faculty in charge

Date

Head of the Department

Dept of CSE



Estd.1981

Bapatla Engineering College

(Autonomous)

Mahatmaji Puram, BAPATLA-522102, Guntur (Dt.), A.P., INDIA

Sponsored by the Bapatla Education Society

Approved by AICTE :: Affiliated to Acharya Nagarjuna University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION

To produce Computer Science Engineers with Global Standards who can handle the challenges of the society and industry with their innovations and services.

MISSION

- To impart high quality education with effective teaching and learning processes.
- To provide an environment where the students can handle research problems confidently.
- To prepare the students with latest technologies with fidelity towards industry.
- To inculcate professional ethics and human values in handling engineering challenges.

Program Educational Objectives (PEOs)

The students graduated in Computer Science and Engineering will be able:

- To choose diverse professional careers in software industry, research, academia, engineering, and administrative services.
- To apply the principles of Basic Sciences, Mathematics and Computer Science to solve real world problems using digital computing systems.
- To analyze, design, implement and evaluate robust, scalable, and cost-effective computer-based systems and processes in the industry with sustained self-learning.
- To be aware of professional and ethical practices in the context of social impacts of computing.

Head of Department
Computer Science & Engineering

Professor & Head
Department of Computer Science & Engineering
Bapatla Engineering College
(Autonomous)

INDEX

S.No	Lab Exercise	Date	Page No
1	Create a remote repository, clone it to local git repository, commit files and push changes to remote repository.	14/07/25	1 – 6
2	Demonstrate 1. Fast forward merge 2. Three-way merge	28/07/25	7 – 10
3	Demonstrate 1. Rebase 2. Pull 3. Fetch	25/08/25	11 – 15
4	Launch an EC2 instance, install Java, Maven, and Tomcat, and manually deploy a Maven web application to Tomcat.	01/09/25	16 – 23
5	Create Jenkins job integrated with GitHub and Maven to pull code from GitHub and build the project.	08/09/25	24 – 25
6	Create Jenkins job to pull, build, and automatically deploy the .war file to the Tomcat server.	06/10/25 to 13/10/25	32 – 34
7	Demonstrate containerization using Docker. (Pull a Docker image from Docker Hub and create a running container from it)	03/11/25 to 07/11/25	35 – 38
8	Docker CI/CD Pipeline using Jenkins.	08/11/25	39 – 43
9	Complete Docker CI/CD pipeline using Jenkins and Ansible.	10/11/25 to 17/11/25	44 – 56

Lab Experiment – 1

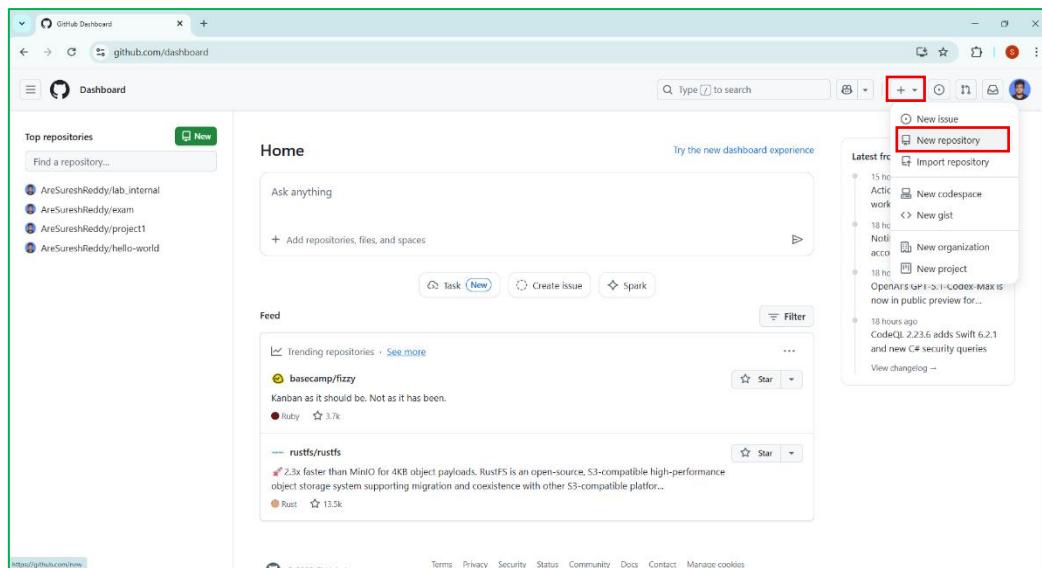
Aim:

Create a remote repository, clone it to local git repository, commit files and push changes to remote repository.

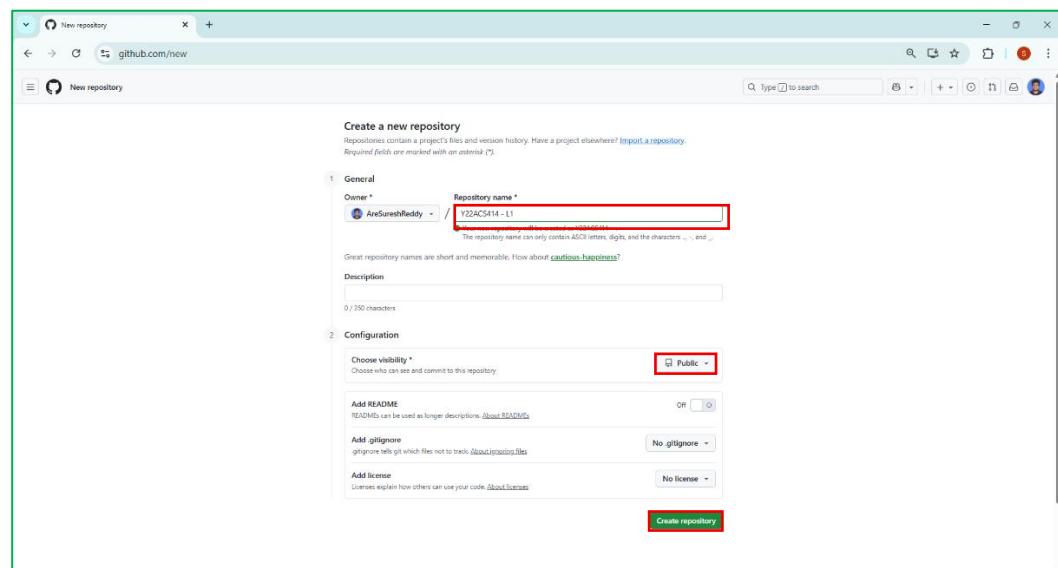
Procedure:

Step 1: Creating a Remote Repository

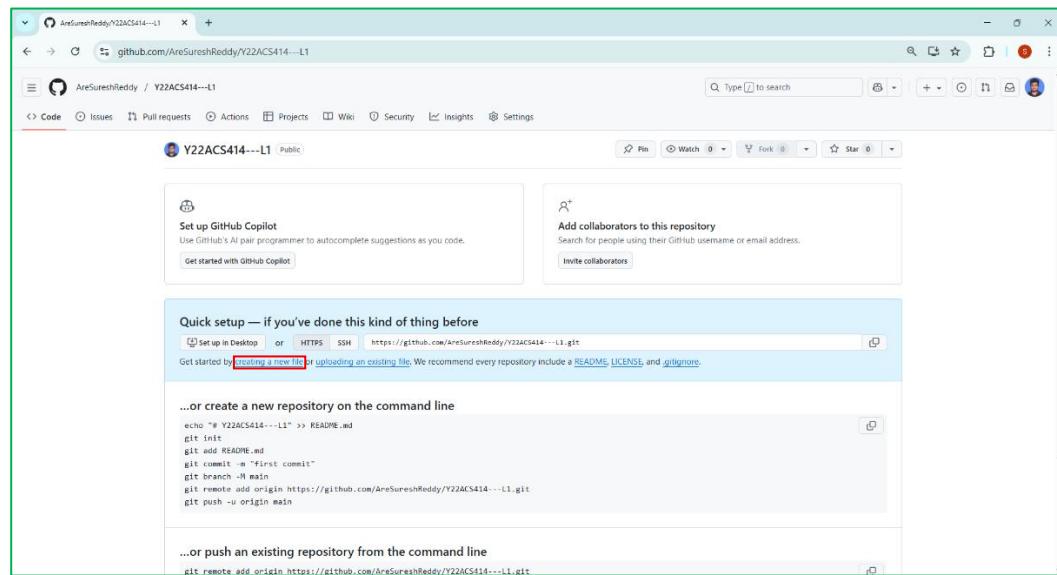
- Open www.github.com in your browser
- Login to your Github account using your credentials.
- Click on “+” icon and then new repository on the top of the website



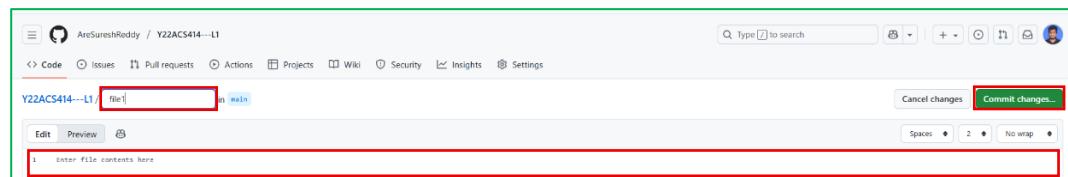
- Provide the Repository name of your choice. Keep the Repository Public and Click on Create Repository, this will create your Remote Repository.



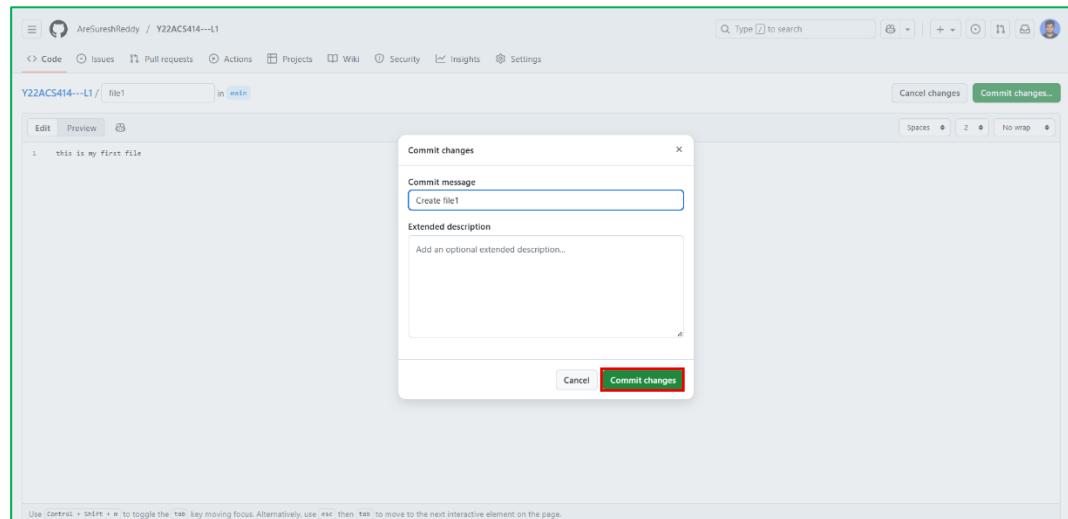
- After creating a new repository click on the hyperlink “creating a new file”.



- Provide the File Contents and a suitable file name and then click on commit message.

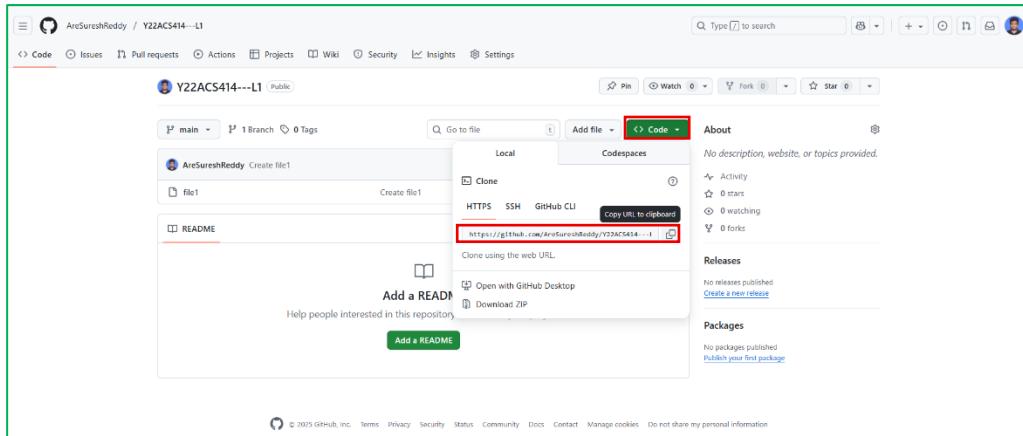


- A pop will appear which will ask to provide a commit message. Please add a suitable commit message and then click on “Commit Changes”.



Step 2: Cloning The Remote Repository

- To clone a repository we need the url of the remote repository. To get the url click on the code option and copy the link.



- Clone the repository using git clone command this will create a new Repository with the name of Remote Repository.

```
$ git clone https://github.com/AreSureshReddy/Y22ACS414---L1.git
```

This will create a repository in your working directory with the name of the remote repository.

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414---L1
$ ls
Y22ACS414---L1/
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414 (master)
$ cd Y22ACS414---L1/
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ ls
file1
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$
```

- we can also know the commits we made in the remote repository using the command git log.

```
$ git log
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git log
commit f9294442295045b219b542f6ba8bf26ad53304b (HEAD -> main, origin/main, origin/HEAD)
Author: Aresureshreddy <aresureshreddy589@gmail.com>
Date:   Fri Dec 5 21:12:40 2025 +0530

    Create file1

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$
```

- Check the status of the Repository using the git status command.

```
$ git status
```

This will report the status of the repository weather the constituents of the repository are untracked or in the staging area or they are already committed.

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Step 3: Committing files and pushing changes to remote repository.

- Create new files in the repository using any of the file creating commands.

```
$ cat >> file2
```

```
$ touch file3
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ ls
file1

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ cat >> file2
this is file2

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ touch file3

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ ls
file1  file2  file3
```

- Check the status of the repository now.

```
$ git status
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git status
on branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file2
    file3

nothing added to commit but untracked files present (use "git add" to track)
```

- Adding the untracked files to the Staging Area
- To add the files to the staging area we can use the git add command.

```
$ git add file2 (To add a particular file to staging area)
```

```
$ git add . (To add all the untracked files to the staging area)
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git add file2
warning: in the working copy of 'file2', LF will be replaced by CRLF the next time Git touches it

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git add .

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file2
    new file:   file3
```

Now as we can see the files are added to the staging area yet to be commit.

- Committing the files in the staging area.
- To commit the files in the staging area we use the command.

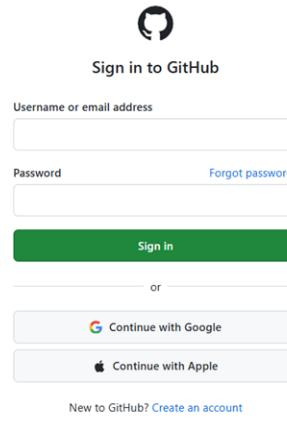
```
$ git commit -m "Commit_Message"
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git commit -m "Commit_Message"
[main 2b23f16] "Commit_Message"
2 files changed, 1 insertion(+)
create mode 100644 file2
create mode 100644 file3
```

- Pushing the Commits to the Remote Repository
- Now push the commits to the remote repository using git push command.

```
$ git push origin main
```

When you start executing this command you will be asked to provide authorization to the remote repository.



Click on Sign. If you already have your account signed up in your browser the commits will automatically pushed or else, you need to sign in using your username and password.

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/Y22ACS414---L1 (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 333 bytes | 333.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AreSureshReddy/Y22ACS414---L1.git
  f929444..2b23f16  main -> main
```

- As we can observe our remote repository have the updated files and the commits.

Before Pushing Changes

After Pushing Changes

Lab Experiment – 2

Aim:

Demonstrate 1. Fast forward merge 2. Three-way merge

Procedure:

Step 1: Fast forward merge

- Initialize a Git repository

```
$ git init
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414 (master)
$ git init fastforward-demo
cd fastforward-demo
Initialized empty Git repository in C:/Users/aresu/OneDrive/Desktop/Y22ACS414/fastforward-demo/.git/
```

- Create or modify files and commit them on the main branch

```
$ echo "line 1" > file.txt
```

```
$ git add file.txt
```

```
$ git commit -m "Initial commit"
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (master)
$ echo "line 1" > file.txt
git add file.txt
git commit -m "Initial commit"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) c73b96b] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
```

- Create a new branch from the main branch

```
$ git checkout -b feature
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (master)
$ git checkout -b feature
Switched to a new branch 'feature'

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (feature)
```

- Make changes in the new branch and commit them

```
$ cat >> file.txt
```

```
$ git add .
```

```
$ git commit -m "Work done in feature branch"
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (feature)
$ cat >> file.txt
feature branch

aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (feature)
$ git add .
git commit -m "work done in feature branch"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[feature 0de76ad] work done in feature branch
 1 file changed, 1 insertion(+)
```

- Switch back to the master branch

```
$ git checkout master
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (feature)
$ git checkout master
Switched to branch 'master'
```

Note : Ensure no new commits were added to main (This condition makes fast-forward merge possible.)

- Merge the feature branch into the main branch

```
$ git merge feature
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (master)
$ git merge feature
Updating c73b96b..0de76ad
Fast-forward
  file.txt | 1 +
  1 file changed, 1 insertion(+)
```

- Verify the commit history

```
$ git log
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/fastforward-demo (master)
$ git log
commit 0de76ad26a165cb1da420fb30ab6ed412fc09690 (HEAD -> master, feature)
Author: AreSureshReddy <aresureshreddy589@gmail.com>
Date:   Sat Dec 6 02:40:06 2025 +0530

    work done in feature branch

commit c73b96bdbe5f73f0be3cac46486571d40e3e61e6
Author: AreSureshReddy <aresureshreddy589@gmail.com>
Date:   Sat Dec 6 02:37:16 2025 +0530

    Initial commit
```

Step 2: Three-way merge

- Initialize a Git repository

```
$ git init threeway-demo
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414 (master)
$ git init threeway-demo
Initialized empty Git repository in C:/Users/aresu/OneDrive/Desktop/Y22ACS414/threeway-demo/.git/
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414 (master)
$ cd threeway-demo/
```

- Create initial commit on main branch

```
$ echo "line 1" > file.txt
```

```
$ git add file.txt
```

```
$ git commit -m "Initial commit"
```

```
aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (master)
$ echo "Line 1" > file.txt
git add file.txt
git commit -m "Initial commit"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) 1b5127d] Initial commit
 1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

- Create a new branch

```
$ git checkout -b feature
```

```
aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (master)
$ git checkout -b feature
Switched to a new branch 'feature'
```

- Make changes and commit in feature branch

```
$ cat >> file.txt
```

```
$ git add .
```

```
$ git commit -m "Work done in feature branch"
```

```
aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (feature)
$ cat >> file.txt
git add .
git commit -m "Work done in feature branch"
line 2 in feacher branch

warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[feature dc4030e] Work done in feature branch
 1 file changed, 1 insertion(+)
```

- Switch back to master branch

```
$ git checkout master
```

```
aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (feature)
$ git checkout master
Switched to branch 'master'

aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (master)
$
```

- Make new commits in main branch also (This creates divergence.)

```
$ cat >> file.txt
```

```
$ git add .
```

```
$ git commit -m "Main branch update"
```

```
aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (master)
$ cat >> file.txt
master changes

aresu@sureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (master)
$ git add .
git commit -m "Main branch update"
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
[master a21407d] Main branch update
 1 file changed, 1 insertion(+)
```

- Merge the feature branch into main

```
$ git merge feature
```

```
aresu@SureshReddy MINGW64 ~/OneDrive/Desktop/Y22ACS414/threeway-demo (master)
$ git merge feature
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Note: a **merge conflict** is **normal** in a three-way merge. Open the file showing the merge conflict and edit it manually.

- Verify the commit history

```
$ git log --oneline --graph
```

```
aresu@SureshReddy MINGW64 ~/oneDrive/Desktop/Y22ACS414/threeway-demo (master|MERGING)
$ nano file1.txt

aresu@SureshReddy MINGW64 ~/oneDrive/Desktop/Y22ACS414/threeway-demo (master|MERGING)
$ git add .

aresu@SureshReddy MINGW64 ~/oneDrive/Desktop/Y22ACS414/threeway-demo (master|MERGING)
$ git commit -m "last commit"
[master 44adf3c] last commit

aresu@SureshReddy MINGW64 ~/oneDrive/Desktop/Y22ACS414/threeway-demo (master)
$ git log --oneline --graph
* 44adf3c (HEAD -> master) last commit
|\ \
| * 8d85002 (feature) commit
| * 9e700ec main commit
| * 1e5248a Merge branch 'feature'
|\ \
| * 5520121 feature commit
| * 0ac02fd master commit
|/
|/
```

Lab Experiment – 3

Aim:

Demonstrate 1. Rebase 2. Pull 3. Fetch

Procedure:

1. Rebase

Step 1: Create a new repository

```
$ mkdir rebase-demo  
$ cd rebase-demo  
$ git init
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414  
$ mkdir rebase-demo  
  
aresu@SureshReddy MINGW64 /d/Devops/y22acs414  
$ cd rebase-demo  
  
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo  
$ git init  
Initialized empty Git repository in D:/Devops/y22acs414/rebase-demo/.git/
```

Step 2: Create first commit on master.

```
$ echo "Line 1" > file.txt  
$ git add file.txt  
$ git commit -m "Initial commit on master"  
"
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)  
$ echo "Line 1" > file.txt  
  
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)  
$ git add file.txt  
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time  
  Git touches it  
  
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)  
$ git commit -m "Initial commit on master"  
[master (root-commit) 823d084] Initial commit on master  
  1 file changed, 1 insertion(+)  
  create mode 100644 file.txt
```

Step 3: Create a second commit on master.

```
$ echo "Line 2" >> file.txt  
$ git add file.txt  
$ git commit -m "Second commit on master"
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ echo "Line 2" > file.txt

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time
  Git touches it

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git commit -m "Second Commit on master"
[master 0e545ee] Second Commit on master
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Now the **master** branch history is:

```
$ git log --oneline --graph
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git log --oneline --graph
* 0e545ee (HEAD -> master) Second Commit on master
* 823d084 Initial commit on master
```

Step 4: Create a new feature branch.

```
$ git checkout -b feature
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git checkout -b feature
Switched to a new branch 'feature'
```

Step 5: Make commits on feature branch.

```
$ echo "Feature work 1" >> file.txt
```

```
$ git add file.txt
```

```
$ git commit -m "Feature commit 1"
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ echo "feature work 1" >> file.txt

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time
  Git touches it

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git commit -m "feature commit 1"
[feature 9691277] feature commit 1
 1 file changed, 1 insertion(+)
```

```
$ echo "Feature work 2" >> file.txt
```

```
$ git add file.txt
```

```
$ git commit -m "Feature commit 2"
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ echo "feature work 2" >> file.txt

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time G
it touches it

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git commit -m "feature commit 2"
[feature d36edd8] feature commit 2
 1 file changed, 1 insertion(+)
```

Now the **feature** branch history is:

```
$ git log --oneline --graph
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git log --oneline --graph
* d36edd8 (HEAD -> feature) feature commit 2
* 9691277 feature commit 1
* 0e545ee (master) Second Commit on master
* 823d084 Initial commit on master
```

Step 6: Add new commits to master (simulate team updates)

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git checkout master
Switched to branch 'master'
```

```
$ echo "Main branch update" >> file.txt
```

```
$ git add file.txt
```

```
$ git commit -m "Main commit after feature started"
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ echo "main branch update" >> file.txt

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time
ouches it

aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git commit -m "main commit after feature started"
[master 4f15ec2] main commit after feature started
 1 file changed, 1 insertion(+)
```

Now the **master** branch history is:

```
$ git log --oneline --graph
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git log --oneline --graph
* 4f15ec2 (HEAD -> master) main commit after feature started
* 0e545ee Second Commit on master
* 823d084 Initial commit on master
```

While the **feature** branch history is:

```
$ git log --oneline --graph
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git log --oneline --graph
* d36edd8 (HEAD -> feature) feature commit 2
* 9691277 feature commit 1
* 0e545ee (master) Second Commit on master
```

Step 7: Rebase feature onto master.

Switch to feature branch:

```
$ git checkout feature
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (master)
$ git checkout -b feature
Switched to a new branch 'feature'
```

Run rebase:

```
$ git rebase main
```

```
[detached HEAD c44546d] Feature commit 1
 1 file changed, 2 insertions(+)
Successfully rebased and updated refs/heads/feature.
```

Now the **feature** branch history is:

```
$ git log --oneline --graph
```

```
aresu@SureshReddy MINGW64 /d/Devops/y22acs414/rebase-demo (feature)
$ git log --oneline --graph
* d36edd8 (HEAD -> feature) feature commit 2
* 9691277 feature commit 1
* 0e545ee (master) Second Commit on master
* 823d084 Initial commit on master
```

2. Pull

Purpose:

Pull downloads new changes from a remote repository and merges them into your current branch.

Step 1: Open a directory having git initialized already

```
$ cd Y22ACS414_L1
```

```
$ ls
```

```
aresu@SureshReddy MINGW64 /d/Devops
$ cd Y22ACS414---L1/

aresu@SureshReddy MINGW64 /d/Devops/Y22ACS414---L1 (main)
$ ls
file1 file2 file3
```

Step 2: Pull the commits from the main branch in the remote repository.

```
$ git pull origin main
```

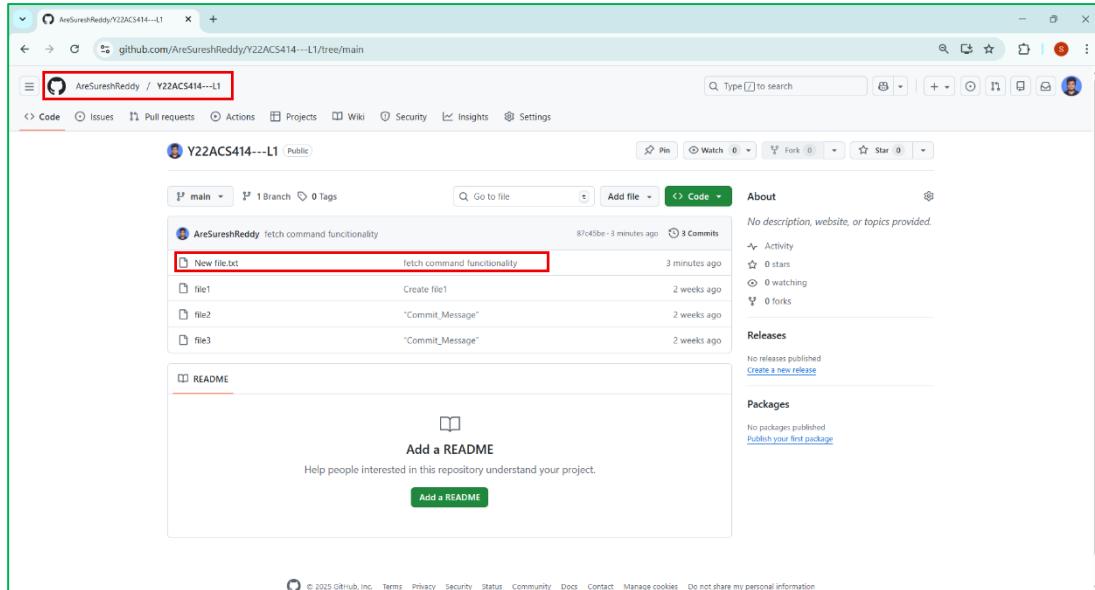
```
aresu@SureshReddy MINGW64 /d/Devops/Y22ACS414---L1 (main)
$ git pull origin main
From https://github.com/AreSureshReddy/Y22ACS414---L1
 * branch            main      -> FETCH_HEAD
Updating 2b23f16..87c45be
Fast forward
```

3. Fetch

Purpose:

Fetch downloads commits, branches, and files from the remote repository **without merging** them into your current branch.

Step 1: Add new commits into the remote repository.



Step 2: fetch the updates to the local repository.

```
$ git fetch origin
```

```
aresu@SureshReddy MINGW64 /d/Devops/Y22ACS414---L1 (main)
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1012 bytes | 63.00 KiB/s, done.
From https://github.com/AreSureshReddy/Y22ACS414---L1
 2b23f16..87c45be main      -> origin/main
```

Note:

The above command

- Downloads all updates
- Stores them in origin/main (or other remote-tracking branches)
- **But your local branch stays unchanged**

```
$ ls
```

```
aresu@SureshReddy MINGW64 /d/Devops/Y22ACS414---L1 (main)
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1012 bytes | 63.00 KiB/s, done.
From https://github.com/AreSureshReddy/Y22ACS414---L1
  2b23f16..87c45be  main       -> origin/main

aresu@SureshReddy MINGW64 /d/Devops/Y22ACS414---L1 (main)
$ ls
file1  file2  file3
```

Lab Experiment – 4

Aim:

Launch an EC2 instance, install Java, Maven, and Tomcat, and manually deploy a Maven web application to Tomcat.

Procedure:

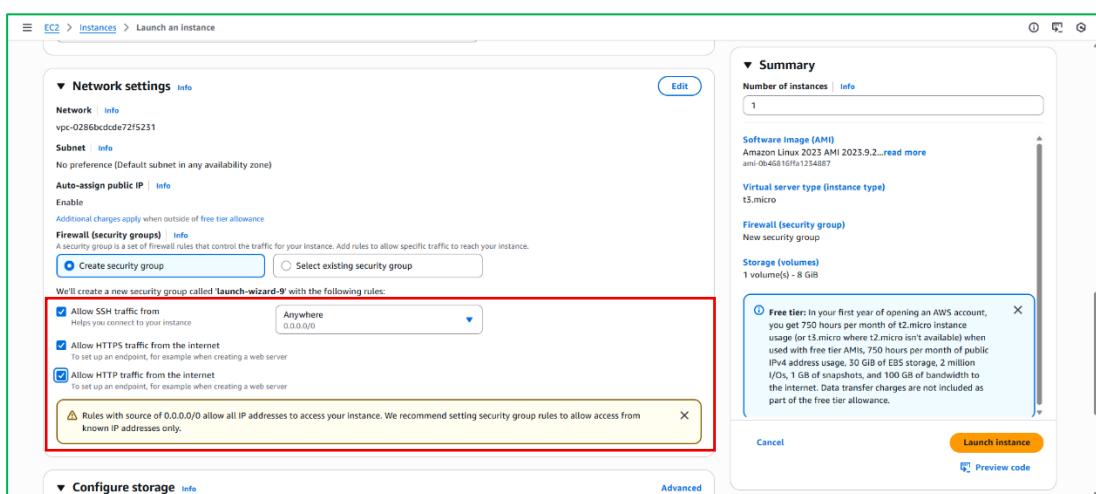
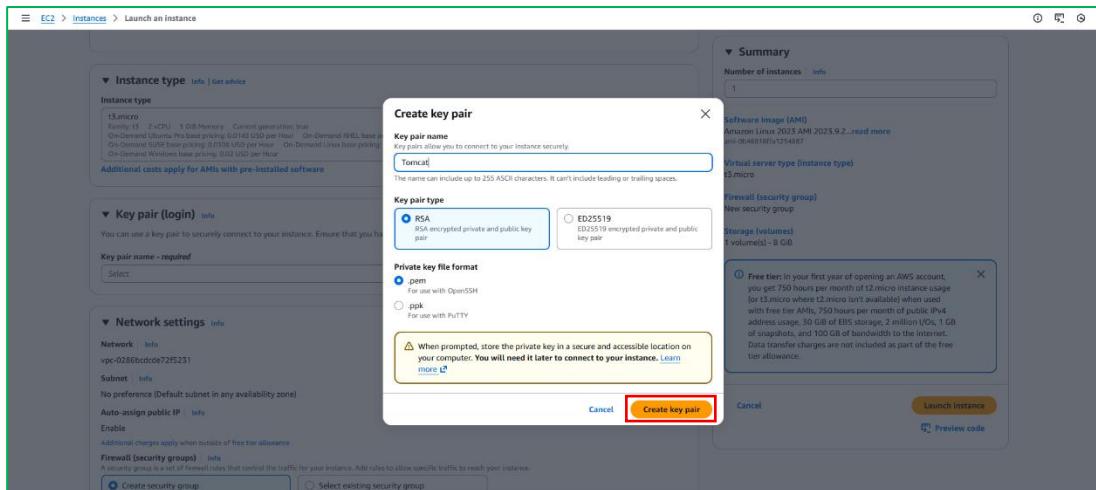
Step 1: Launching an EC2 instance in aws.

- Click on Launch instance and provide a name to it as follows.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation links like Dashboard, AWS Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table titled 'Instances (5) info' with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, and Elastic IP. Below the table, a section titled 'Select an instance' is visible. At the top right of the main area, there are several buttons: Connect, Instance state, Actions, and Launch instances, with the 'Launch instances' button being highlighted with a red box.

The screenshot shows the 'Launch an instance' wizard. Step 1: Set instance details. It has sections for 'Name and tags' (Name: Tomcat), 'Application and OS Images (Amazon Machine Image)' (Search bar: Search our full catalog including 1000s of application and OS images, Recents: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian, Quick Start: Mac, ubuntu, Microsoft, RedHat, SUSE, debian), and 'Summary' (Number of instances: 1). Other sections include 'Software Image (AMI)' (Amazon Linux 2023 AMI 2023.9.2...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), 'Storage (volumes)' (1 volume(s) - 8 GiB), and a note about the free tier: 'Free tier: In your first year of opening an AWS account...'. Buttons at the bottom include 'Cancel', 'Preview code', and a large orange 'Launch instance' button.

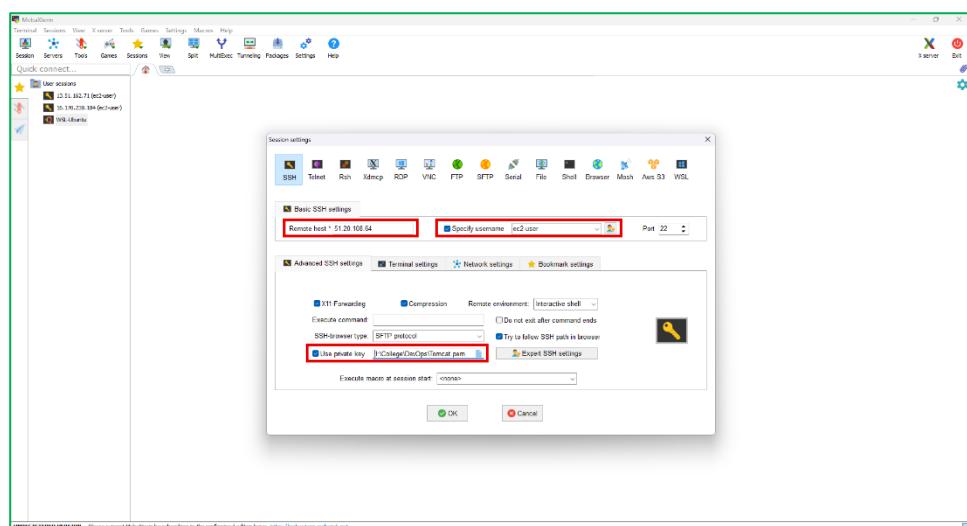
- Create a new Key pair or if already exists keep it there.
- Select the required Protocols by default.



➤ Click on launch instance.



Step 2: Install Java in Tomcat instance through MobaTerm. Specify ip address and created key pair.



```

      ,      #
      ~\_ ####_      Amazon Linux 2023
      ~~ \_#####\
      ~~  \###|
      ~~   \#/ ____ https://aws.amazon.com/linux/amazon-linux-2023
      ~~    \~' '-'>
      ~~~   /
      ~~.~.~/_/
      _/m/.'

Last login: Sat Dec 13 15:39:32 2025 from 152.59.203.173
[ec2-user@Y22ACS414Tomcat ~]$ sudo su -
[root@Y22ACS414Tomcat ~]# sudo yum install java-21
Amazon Linux 2023 Kernel Livepatch repository
242 kB/s | 29 kB 00:00
Dependencies resolved.
=====
=====
  Package
Architecture          Version
Repository           Size
=====
=====
Installing:
  java-21-amazon-corretto          x86_64
  1:21.0.9+11-1.amzn2023.1        amazonlinux
  218 k
Transaction Summary
=====
=====
Install 35 Packages

Total download size: 105 M
Installed size: 283 M
Is this ok [y/N]: y
Downloading Packages:
Complete!

```

Step 3: Installing Maven in Tomcat instance.

```
[root@Y22ACS414Tomcat opt]# sudo yum install maven
Complete!
```

Last metadata expiration check: 0:29:54 ago on Sat Dec 13 15:42:50 2025.				
Dependencies resolved.				
Package	Architecture	Version	Repository	Size
Installing:				
maven	noarch	1:3.8.4-3.amzn2023.0.5	amazonlinux	18 k
installing dependencies:	noarch	1.5.0-3.amzn2023.0.3	amazonlinux	76 k

Step 4: Installing the Tomcat9 in aws for that we need to follow few steps.

- Search in google for tomcat9 download goto the first result you found.
- Copy the link address of it and paste it in mobxterm.
- In Mobxterm we will be having a zip folder of tomcat9.
- To unzip those file we will write a command as shown.
- Move that unzipped file into a new folder “Tomcat”

```
[root@Y22ACS414Tomcat ~]# cd /opt
```

```
[root@Y22ACS414Tomcat opt]# ls
```

```
[root@Y22ACS414Tomcat opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.113/bin/apache-tomcat-9.0.113.tar.gz
```

```
--2025-12-13 15:48:08-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.113/bin/apache-tomcat-9.0.113.tar.gz
```

```
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13049663 (12M) [application/x-gzip]
Saving to: ‘apache-tomcat-9.0.113.tar.gz’
```

```
apache-tomcat-9.0.113.tar.gz
100%[=====] 12.44M --.-KB/s in 0.05s
```

```
2025-12-13 15:48:08 (227 MB/s) - ‘apache-tomcat-9.0.113.tar.gz’ saved [13049663/13049663]
```

```
[root@Y22ACS414Tomcat opt]# ls
apache-tomcat-9.0.113.tar.gz aws
```

```
[root@Y22ACS414Tomcat opt]# tar -xvf apache-tomcat-9.0.113.tar.gz
```

```
[root@Y22ACS414Tomcat opt]# mv apache-tomcat-9.0.113 tomcat
```

```
[root@Y22ACS414Tomcat opt]# ls
apache-tomcat-9.0.113.tar.gz aws tomcat
```

```
[root@Y22ACS414Tomcat opt]# find -name context.xml
./tomcat/conf/context.xml
./tomcat/webapps/docs/META-INF/context.xml
./tomcat/webapps/examples/META-INF/context.xml
./tomcat/webapps/host-manager/META-INF/context.xml
./tomcat/webapps/manager/META-INF/context.xml
[root@Y22ACS414Tomcat opt]# vi ./tomcat/webapps/host-manager/META-INF/context.xml
```

```
[root@Y22ACS414Tomcat opt]# vi ./tomcat/webapps/manager/META-INF/context.xml
```

```
[root@Y22ACS414Tomcat opt]# vi ./tomcat/webapps/manager/META-INF/context.xml
```

```
[root@Y22ACS414Tomcat opt]# find -name tomcat-users.xml
./tomcat/conf/tomcat-users.xml
```

```
[root@Y22ACS414Tomcat opt]# vi ./tomcat/conf/tomcat-users.xml
```

```
[root@Y22ACS414Tomcat opt]# ln -s /opt/tomcat/bin/startup.sh tomcatup
```

```
[root@Y22ACS414Tomcat opt]# ln -s /opt/tomcat/bin/shutdown.sh tomcatdown
```

```
[root@Y22ACS414Tomcat opt]# ./tomcatup
```

```
Using CATALINA_BASE: /opt/tomcat
Using CATALINA_HOME: /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:      /usr
Using CLASSPATH:     /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements. See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to you under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License. You may obtain a copy of the License at
      http://www.apache.org/licenses/LICENSE-2.0
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<Context antResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sessionCookieName="strict" />
  <!--<Valve className="org.apache.catalina.valves.RemoteC�RValve"
    allow="127.0.0.0/8::1/128" />-->
  <Manager sessionAttributeValueClassNameFilter="java.lang.Integer|Long|Number|String|org.apache.catalina.filters.CsrfPreventionFilter\$LruCache(?:\$1)?|java.util.(?:Linked)?HashMap"/>
</Context>
<!--
-->
```

➤ The below edited file of Manager.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements. See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to you under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License. You may obtain a copy of the License at
      http://www.apache.org/licenses/LICENSE-2.0
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<Context antResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sessionCookieName="strict" />
  <!--<Valve className="org.apache.catalina.valves.RemoteC�RValve"
    allow="127.0.0.0/8::1/128" />-->
  <Manager sessionAttributeValueClassNameFilter="java.lang.Integer|Long|Number|String|org.apache.catalina.filters.CsrfPreventionFilter\$LruCache(?:\$1)?|java.util.(?:Linked)?HashMap"/>
</Context>
<!--
-->
```

➤ The edited file below is of conf/tomcat-users.xml

```
<!--
  All users need to set the password to something appropriate.
-->
<!--
  <!--
    <user username="admin" password="<must-be-changed>" roles="manager-gui"/>
    <user username="robot" password="<must-be-changed>" roles="manager-script"/>
  -->
<!--
  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!-- ... --> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <!--
    <role rolename="tomcat"/>
    <role rolename="role1"/>
    <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
    <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
    <user username="role1" password="<must-be-changed>" roles="role1"/>
  -->
  <!--<role rolename="admin"/>
  <!--<role rolename="tomcat"/>
  <!--<role rolename="user"/>
  <!--<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
  <!--<user username="tomcat" password="tomcat" roles="manager-script"/>
  <!--<user username="user" password="userpass" roles="manager-gui,manager-status,manager-script"/>
-->
</tomcat-users>
-- INSERT --
```

➤ Add a New security group port number be “8080”

The first screenshot shows the EC2 instance details page for i-0c04fac818faaab4 (Tomcat). It highlights the 'Security groups' section, which lists sg-0f31b217087c536ea (launch-wizard-9) with a red box.

The second screenshot shows the security group sg-0f31b217087c536ea - launch-wizard-9. It displays the 'Inbound rules' tab, listing three existing rules and one new rule being added (Custom TCP, Port 8080, Anywhere). A red box highlights the 'Edit inbound rules' button.

The third screenshot shows the 'Edit inbound rules' dialog. It lists four rules, with the fourth one (Custom TCP, Port 8080, Anywhere) highlighted by a red box. An 'Add rule' button is visible at the bottom left.

➤ Go to google and type IP address:8080 to get the tomcat on browser.

The screenshot shows a browser window for Apache Tomcat 9.0.113. The URL bar shows 51.20.108.64:8080. The page content includes a success message: "If you're seeing this, you've successfully installed Tomcat. Congratulations!", a cartoon cat logo, and links for documentation, examples, and developer quick start.

➤ Now go to Manager App use “admin” for both user and password.

Step 5: Creating a manual Maven web application to deploy on tomcat.

```
[root@Y22ACS414Tomcat opt]# mvn archetype:generate \
-DarchetypeGroupId=org.apache.maven.archetypes \
-DarchetypeArtifactId=maven-archetype-webapp \
-DarchetypeVersion=1.4 \
-DgroupId=com.helloworld \
-DartifactId=helloworld-web \
-DinteractiveMode=false \
-U

[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.4.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] Parameter: packageInPathFormat, Value: com/helloworld
[INFO] Parameter: package, Value: com.helloworld
[INFO] Parameter: groupId, Value: com.helloworld
[INFO] Parameter: artifactId, Value: helloworld-web
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: /opt/helloworld-web
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.171 s
[INFO] Finished at: 2025-12-13T16:23:33Z
[INFO] -----
```

```
[root@Y22ACS414Tomcat opt]# cd helloworld-web/
```

```
[root@Y22ACS414Tomcat helloworld-web]# mvn clean package
```

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.helloworld:helloworld-web >-----
[INFO] Building helloworld-web Maven Webapp 1.0-SNAPSHOT
[INFO] -----[ war ]-----
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-war-plugin/3.2.2/[INFO] Packaging webapp
[INFO] Assembling webapp [helloworld-web] in [/opt/helloworld-web/target/helloworld-web]
[INFO] Processing war project
[INFO] Copying webapp resources [/opt/helloworld-web/src/main/webapp]
[INFO] Webapp assembled in [21 msec]
[INFO] Building war: /opt/helloworld-web/target/helloworld-web.war
[INFO] -----
[INFO] BUILD SUCCESS
```

```
[INFO] -----  
[INFO] Total time: 4.374 s  
[INFO] Finished at: 2025-12-13T16:25:56Z  
[INFO] -----
```

```
[root@Y22ACS414Tomcat helloworld-web]# cp target/helloworld-web.war /opt/tomcat/webapps/
```

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle > [30] minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle > [30] minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle > [30] minutes
/helloworld-web	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle > [30] minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle > [30] minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle > [30] minutes

Hello World!

Lab Experiment – 5

Aim:

Create Jenkins job integrated with GitHub and Maven to pull code from GitHub and build the project.

Procedure:

Step 1: Create an instance named as “Jenkins” and connect it to Mobxterm.

- Install Java-21 and Jenkins on Jenkins instance.
- Search on web “Jenkins download” and download the LTS version of the Jenkins.

```
,      #
~\_ #####_      Amazon Linux 2023
~~ \#####\
~~ \|##|
~~  \#/   https://aws.amazon.com/linux/amazon-linux-2023
~~    \|~'`->
~~~   /
~~~.~. /`/
~~~.~. /`/
~/m/'
```

Last login: Sat Dec 13 18:28:25 2025 from 152.59.203.173

```
[ec2-user@Y22ACS414Jenkins ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-21-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
--2025-12-13 18:31:32-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.174.133, 2a04:4e42:6b::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.174.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 192
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo
100%[=====] 192 --.-KB/s  in 0s

2025-12-13 18:31:32 (19.0 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [192/192]
```

Installed:

jenkins-2.528.3-1.1.noarch

Complete!

```
[ec2-user@Y22ACS414Jenkins ~]$ sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service →
/usr/lib/systemd/system/jenkins.service.
```

```
[ec2-user@Y22ACS414Jenkins ~]$ sudo systemctl start jenkins
```

```
[ec2-user@Y22ACS414Jenkins ~]$ sudo systemctl status jenkins
```

- jenkins.service - Jenkins Continuous Integration Server

Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)

Active: active (running) since Sat 2025-12-13 18:34:03 UTC; 4s ago

Main PID: 26892 (java)

Tasks: 39 (limit: 1067)

Memory: 390.6M

CPU: 18.644s

CGroup: /system.slice/jenkins.service

└─26892 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

The screenshot shows the Jenkins home page. At the top, there's a search bar and a 'Add description' button. Below that, the 'Welcome to Jenkins!' message is displayed, followed by instructions to 'Start building your software project'. There are several buttons for 'Create a job', 'Set up a distributed build', 'Set up an agent', and 'Configure a cloud'. On the left, there are sections for 'Build Queue' (No builds in the queue), 'Build Executor Status' (0/2), and 'Built-in Node' (offline). At the bottom right, there are links for 'REST API' and 'Jenkins 2.528.2'.

➤ Configure Monitors in 'Manage Jenkins->Nodes->Configure Monitors'

The screenshot shows the 'Manage Jenkins' page. At the top, there are buttons for 'Search settings', 'Set up agent', 'Set up cloud', and 'Delete'. Below that, there are sections for 'System Configuration' (System, Tools, Plugins, Clouds, Appearance), 'Security' (Security, Users), and 'Nodes'. An arrow points from the text 'Configure Monitors in 'Manage Jenkins->Nodes->Configure Monitors'' to the 'Nodes' section.

The screenshot shows the 'Nodes' configuration page. It lists a single node named 'Built-in Node' with details like Architecture (Linux (x86_64)), Clock Difference (In sync), Free Disk Space (5.86 GB), Free Swap Space (0 B), and Free Temp Space (453.01 MB). A 'Configure Monitors' button is highlighted with a red box. A legend at the bottom indicates icons for S (Small), M (Medium), and L (Large).

- Keep remaining values to zero.

Step 2: Install maven manually from google.

- Search for maven download and do the remaining process same as tomcat.

```
[root@Y22ACS414Jenkins opt]# wget https://dlcdn.apache.org/maven/maven-3/3.9.11/binaries/apache-maven-3.9.11-bin.tar.gz
```

```
--2025-12-13 18:44:18-- https://dlcdn.apache.org/maven/maven-3/3.9.11/binaries/apache-maven-3.9.11-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9160848 (8.7M) [application/x-gzip]
Saving to: ‘apache-maven-3.9.11-bin.tar.gz’
apache-maven-3.9.11-bin.tar.gz
100%[=====] 8.74M --.-KB/s in 0.1s
Installed:
git-2.50.1-1.amzn2023.0.1.x86_64      git-core-2.50.1-1.amzn2023.0.1.x86_64      git-core-
doc-2.50.1-1.amzn2023.0.1.noarch      perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.7.noarch  perl-Git-2.50.1-1.amzn2023.0.1.noarch      perl-
TermReadKey-2.38-9.amzn2023.0.2.x86_64    perl-lib-0.65-477.amzn2023.0.7.x86_64
```

Complete!

```
[root@Y22ACS414Jenkins opt]# tar -xvf apache-maven-3.9.11-bin.tar.gz
apache-maven-3.9.11/README.txt
apache-maven-3.9.11/LICENSE
apache-maven-3.9.11/NOTICE
apache-maven-3.9.11/lib/
apache-maven-3.9.11/lib/aopalliance.license
apache-maven-3.9.11/lib/asm.license
apache-maven-3.9.11/lib/wagon-http-shared-3.5.3.jar
....
```

```
[root@Y22ACS414Jenkins opt]# mv apache-maven-3.9.11 maven
```

```
[root@Y22ACS414Jenkins opt]# ls
apache-maven-3.9.11-bin.tar.gz aws maven
```

```
[root@Y22ACS414Jenkins jvm]# vi ~/.bash_profile
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
# Java configuration
export JAVA_HOME=/usr/lib/jvm/java-21-amazon-corretto.x86_64

# Maven configuration
export MAVEN_HOME=/opt/maven

# Path configuration
export PATH=$JAVA_HOME/bin:$MAVEN_HOME/bin:$PATH

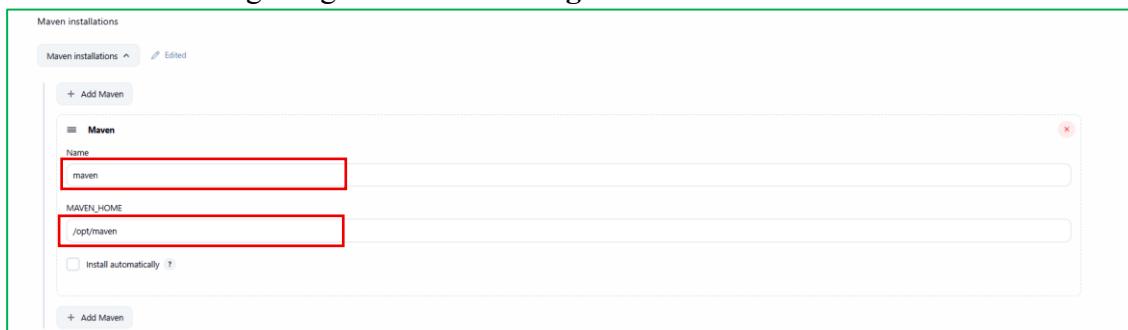
# User specific environment and startup programs
~
```

```
[root@Y22ACS414Jenkins jvm]# source ~/.bash_profile
```

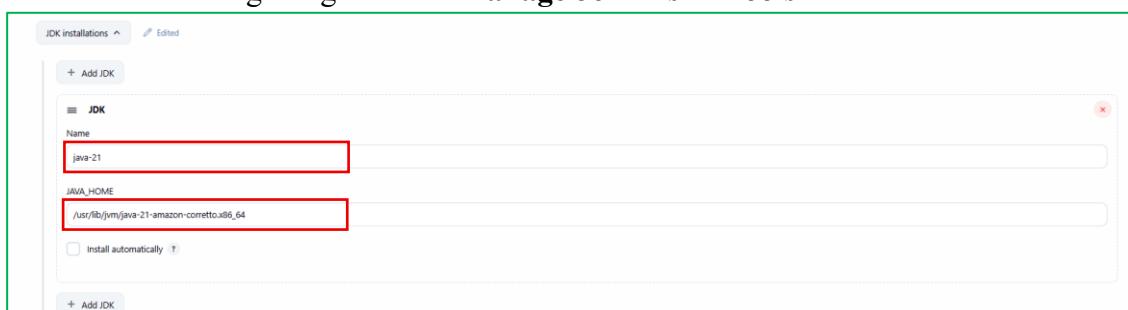
```
[root@Y22ACS414Jenkins jvm]# sudo yum install git
```

Step 3: Pulling the code from GitHub through Jenkins. Before doing it we need to integrate them with Jenkins.

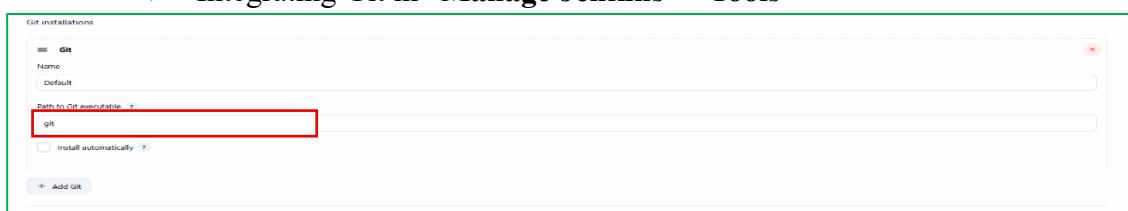
➤ Integrating maven in “Manage Jenkins -> Tools”



➤ Integrating Java in “Manage Jenkins -> Tools”



➤ Integrating Git in “Manage Jenkins -> Tools”



- Install required plugins maven integration and container. Add credentials.
- Create a new job in Jenkins.

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Plugins' section. A red box highlights the 'Available plugins' tab. In the main area, a list of available plugins is shown, with 'Maven Integration 3.27' selected. A red box highlights the 'Install' button at the top right of the list.

Plugin Name	Released	Health
Maven Integration 3.27	4 mo 3 days ago	OK
git 3.8.1	11 days ago	OK
Github 1.43.0	3 mo 7 days ago	OK
GitHub Integration 0.7.3	2 days 16 hr ago	OK
gitlab 1.17	7 mo 4 days ago	OK
Deploy to Container 1.17	1 mo 16 days ago	OK
Timestampers 1.30		OK
Build Warnings		OK
Script Security 1.885		OK

The screenshot shows the Jenkins 'New Item' creation page. A red box highlights the 'Name' field where 'Maven_build' has been entered. Another red box highlights the 'Maven project' option under 'Select an item type'. The 'Create' button at the bottom is also highlighted with a red box.

The screenshot shows the Jenkins 'Source Code Management' configuration page for a 'Git' repository. A red box highlights the 'Repository URL' field containing 'https://github.com/learnwithhsp1/Maven_Project.git'. The 'Save' and 'Apply' buttons at the bottom are also highlighted with red boxes.

The screenshot shows the Jenkins configuration page for a job named "Maven.build". The "Build" section is selected. In the "Goals and options" field, the value "clean install" is highlighted with a red box.

The screenshot shows the Jenkins configuration page for a job named "Maven_Build". The "Post-build Actions" section is selected. Under the "Deploy war/ear to a container" section, the "Containers" field is highlighted with a red box, showing the "+ Add Container" button.

The screenshot shows the "Deploy war/ear to a container" dialog. It displays a credential provider configuration for "Jenkins". The "Username" field contains "admin" and the "Password" field contains "****", both of which are highlighted with red boxes. The "Add" button at the bottom right is also highlighted with a red box.

Jenkins / Maven_build / Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions**

Deploy war/ear to a container

WAR/EAR files: **/*.war

Context path: /

The context path that the container should use to publish the WAR/EAR, note that this may get overridden if the WAR/EAR to deploy has container-specific configuration embedded inside.
(from Deploy to container Plugin)

Containers

Tomcat 9.x Remote

Credentials: admin*****

Tomcat URL: http://13.51.176.242:8080/

Advanced

+ Add Container

Deploy on failure

+ Add post-build action

Save **Apply**

- Give ip address of tomcat in above and apply and save.
- Click on Build now.
- Then check on to the tomcat at manager apps.

Jenkins / Maven_Build

- Status
- Changes
- Workspace
- Build Now**
- Configure
- Delete Maven project
- Modules
- Rename
- Credentials

Maven_Build

Maven

Permalinks

List Applications

Applications		
Path	Version	Display Name
/	None specified	Welcome to Tomcat
/docs	None specified	Tomcat Documentation
/examples	None specified	Servlet and JSP Examples
/host-manager	None specified	Tomcat Host Manager Application
/manager	None specified	Tomcat Manager Application
/maven_project	None specified	Archetype Created Web Application
/sample	None specified	Archetype Created Web Application

HTML Manager Help

The screenshot shows a simple website layout. At the top is a blue header bar with the text "Welcome to My Website" and a subtitle "Your one-stop destination for learning HTML!". Below the header is a grey navigation bar containing links for "Home", "About", and "Contact". The main content area is white and contains three sections: "Home", "About", and "Contact". The "Home" section includes a paragraph about the homepage. The "About" section states that the website is built using HTML and CSS. The "Contact" section provides an email address for reaching out. At the bottom of the page is a dark grey footer bar with the copyright notice "© 2025 My Simple Webpage. All rights reserved."

Welcome to My Website

Your one-stop destination for learning HTML!

Home About Contact

Home

This is the homepage of my simple website. Here, you can find various resources and information.

About

This website is built using HTML and CSS to demonstrate clean and responsive design practices.

Contact

Feel free to reach out via email at example@example.com.

© 2025 My Simple Webpage. All rights reserved.

Lab Experiment – 6

Aim:

Create Jenkins job to pull, build, and automatically deploy the .war file to the Tomcat server.

Procedure:

Step 1: Start the instance of Jenkins and create an ssh in mobxterm with the Jenkins IP.

Step 2: Create a Jenkins job for automatic running and deploy.

The screenshot shows the Jenkins 'New Item' creation interface. At the top, it says 'Jenkins / All / New Item'. Below that is a form with 'Enter an item name' containing 'Maven_Update'. Under 'Select an item type', there are five options: 'Freestyle project', 'Maven project' (which is selected and highlighted with a mouse cursor), 'Pipeline', 'Multi-configuration project', and 'Folder'. At the bottom right is a blue 'OK' button.

The screenshot shows the 'Poll SCM' configuration screen. It lists several triggers: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM' (which is checked). Below this is a 'Schedule' field containing '* * * * *'. A note below says 'No schedules so will only run due to SCM changes if triggered by a post-commit hook'. At the bottom is an unchecked checkbox for 'Ignore post-commit hooks'.

The screenshot shows the Jenkins configuration interface for a job named 'Maven_Build'. The left sidebar lists various configuration sections: General, Source Code Management, Triggers, Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The 'Post-build Actions' section is currently selected and highlighted with a grey background. On the right, under the heading 'Deploy war/ear to a container', there is a form to specify 'WAR/EAR files' (set to '**/*.war) and 'Context path' (empty). Below this is a 'Containers' section with a '+ Add Container' button, which has a hand cursor icon hovering over it. There is also a checkbox for 'Deploy on failure' and a link to '+ Add post-build action'.

The screenshot shows the Jenkins configuration interface for a job named 'Maven_Build'. The left sidebar lists various configuration sections. The 'Tomcat 9.x Remote' section is currently selected and highlighted with a grey background. It contains fields for 'Credentials' (set to 'admin/******** (Maven)'), 'Tomcat URL' (set to 'http://3.110.177.222:8080/'), and an 'Advanced' dropdown menu. Below these are buttons for '+ Add Container' (with a cursor icon hovering over it), 'Deploy on failure' (unchecked), and '+ Add post-build action'. At the bottom are 'Save' and 'Apply' buttons, with 'Save' being highlighted in blue.

- After clicking on apply and save then we need to build the project, so click on build.
- After successfully building, check in Tomcat whether the project is automatically deploying or not.
- We can also change the code in the git repository (index.jsp) and check the output.

List Applications		HTML Manager Help
Applications		
Path	Version	Display Name
/	None specified	Welcome to Tomcat
/docs	None specified	Tomcat Documentation
/examples	None specified	Servlet and JSP Examples
/host-manager	None specified	Tomcat Host Manager Application
/manager	None specified	Tomcat Manager Application
/maven_project	None specified	Archetype Created Web Application
/sample	None specified	Archetype Created Web Application

Welcome to My Website

Your one-stop destination for learning HTML!

Home About Contact

Home

This is the homepage of my simple website. Here, you can find various resources and information.

About

This website is built using HTML and CSS to demonstrate clean and responsive design practices.

Contact

Feel free to reach out via email at example@example.com.

© 2025 My Simple Webpage. All rights reserved.

HTML Forms

Username:

Password:

Lab Experiment – 7

Aim:

Demonstrate containerization using Docker.(Pull a Docker image from Docker Hub and create a running container from it).

Procedure:

Step 1: Create a Docker instance in aws and add two inbound rules.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0729810852556579a	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-06e93c61fd794d8c0	HTTPS	TCP	443	Custom	0.0.0.0/0
sgr-0662a340c18ee69f1	SSH	TCP	22	Custom	0.0.0.0/0
(New Rule)	Custom TCP	TCP	8080 - 9090	Anywhere...	0.0.0.0/0

➤ After that install the docker



Last login: Sun Dec 14 07:45:49 2025 from 152.59.203.181
[ec2-user@Y22ACS414Docker ~]\$ sudo su -

```
[root@Y22ACS414Docker ~]# yum install docker
Amazon Linux 2023 Kernel Livepatch repository 247 kB/s | 29 kB 00:00
Last metadata expiration check: 0:00:01 ago on Sun Dec 14 07:48:33 2025.
Dependencies resolved.
```

Installed:

```
container-selinux-4:2.242.0-1.amzn2023.noarch  containerd-2.1.5-1.amzn2023.0.1.x86_64
docker-25.0.13-1.amzn2023.0.2.x86_64  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  libcgroup-3.0-1.amzn2023.0.1.x86_64
libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64  libnfnetwork-1.0.1-19.amzn2023.0.2.x86_64
libnltnl-1.2.2-2.amzn2023.0.2.x86_64  pigz-2.5-1.amzn2023.0.3.x86_64  runc-1.3.3-
2.amzn2023.0.1.x86_64
```

Complete!

➤ Now we need to start the docker.

```
[root@Y22ACS414Docker ~]# systemctl start docker
```

```
[root@Y22ACS414Docker ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
```

```
[root@Y22ACS414Docker ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

```
[root@Y22ACS414Docker ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

➤ Now pull image from DockerHub.

```
[root@Y22ACS414Docker ~]# docker pull tomcat
```

```
Using default tag: latest
latest: Pulling from library/tomcat
20043066d3d5: Pull complete
627c55a201a9: Pull complete
378e3a6f165e: Pull complete
4f4fb700ef54: Pull complete
901b8cfcd7a7: Pull complete
45e2e3388eeb: Pull complete
841c6b2bc7ed: Pull complete
Digest: sha256:689475f36b76708f19cc939d2d18689de23dd41abdb9e1693c60410ccb7361d6
Status: Downloaded newer image for tomcat:latest
docker.io/library/tomcat:latest
```

```
[root@Y22ACS414Docker ~]# docker images
```

```
REPOSITORY TAG IMAGE ID CREATED SIZE
tomcat latest 557466b563c2 5 days ago 412MB
```

```
[root@Y22ACS414Docker ~]# docker ps
```

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

```
[root@Y22ACS414Docker ~]# docker run -d --name tc1 -p 8081:8080 tomcat:latest
a5619770e148a6671fb547bb7f18dbab194b3ee7971ebd009fc5a4396722f4e4
```

```
[root@Y22ACS414Docker ~]# docker ps
```

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a5619770e148 tomcat:latest "catalina.sh run" 22 seconds ago Up 21 seconds 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp tc1
```

```
[root@Y22ACS414Docker ~]# docker exec -it tc1 /bin/bash
```

```
root@a5619770e148:/usr/local/tomcat# ls
bin BUILDING.txt conf CONTRIBUTING.md filtered-KEYS lib LICENSE logs native-jni-lib
NOTICE README.md RELEASE-NOTES RUNNING.txt temp upstream-KEYS webapps
webapps.dist work
```

```
root@a5619770e148:/usr/local/tomcat# cd webapps.dist/
```

```
root@a5619770e148:/usr/local/tomcat/webapps.dist# ls
docs examples host-manager manager ROOT
```

```
root@a5619770e148:/usr/local/tomcat/webapps.dist# cp -R * ../webapps/
```

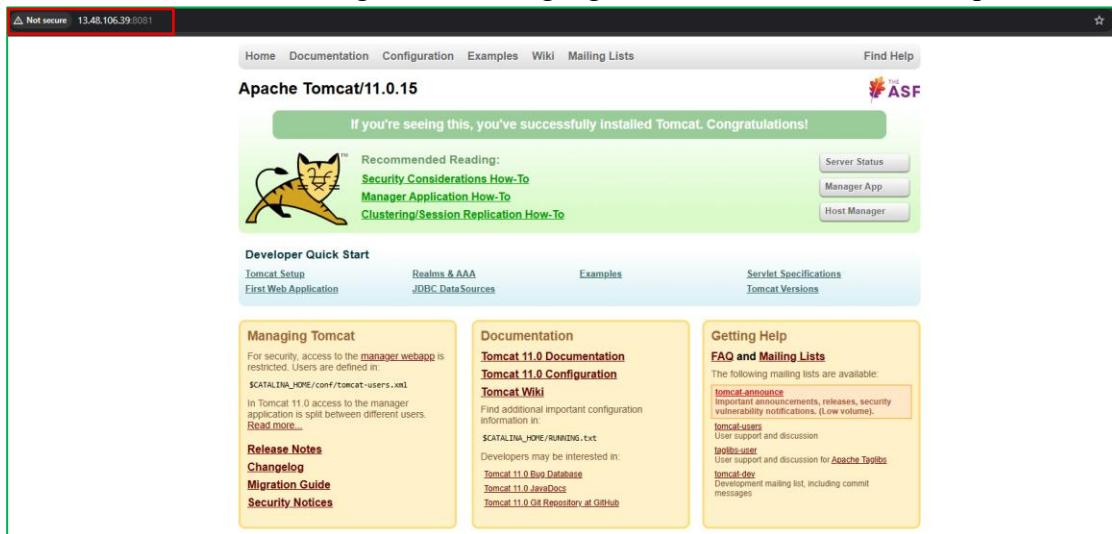
```
root@a5619770e148:/usr/local/tomcat/webapps.dist# cd ..
```

```
root@a5619770e148:/usr/local/tomcat# cd webapps
```

```
root@a5619770e148:/usr/local/tomcat/webapps# ls
docs examples host-manager manager ROOT
```

```
root@a5619770e148:/usr/local/tomcat/webapps# exit
exit
```

- After running ‘Docker image’ goto browser and search the ‘ipv4:8081’



- Create a file name “Dockerfile” and write the following code.

```
[root@Y22ACS414Docker ~]# vi Dockerfile
```

```
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps/
COPY ./*.war .usr/local/tomcat/webapps/
```

- Build Docker image using “Dockerfile”.

```
[root@Y22ACS414Docker ~]# docker build -t sample_image .
[+] Building 1.2s (8/8) FINISHED
docker:default
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 228B 0.0s
=> [internal] load metadata for docker.io/library/tomcat:latest 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/3] FROM docker.io/library/tomcat:latest 0.2s
=> [internal] load build context 0.1s
=> => transferring context: 2B 0.0s
=> [2/3] RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps/ 0.3s
=> [3/3] COPY ./*.war .usr/local/tomcat/webapps/ 0.0s
=> exporting to image 0.5s
=> => exporting layers 0.5s
=> => writing image sha256:64c8f7c97b1c1948805fddd538d025795f25da042a7d5b4b2ba391d4c78b6aaa 0.0s
```

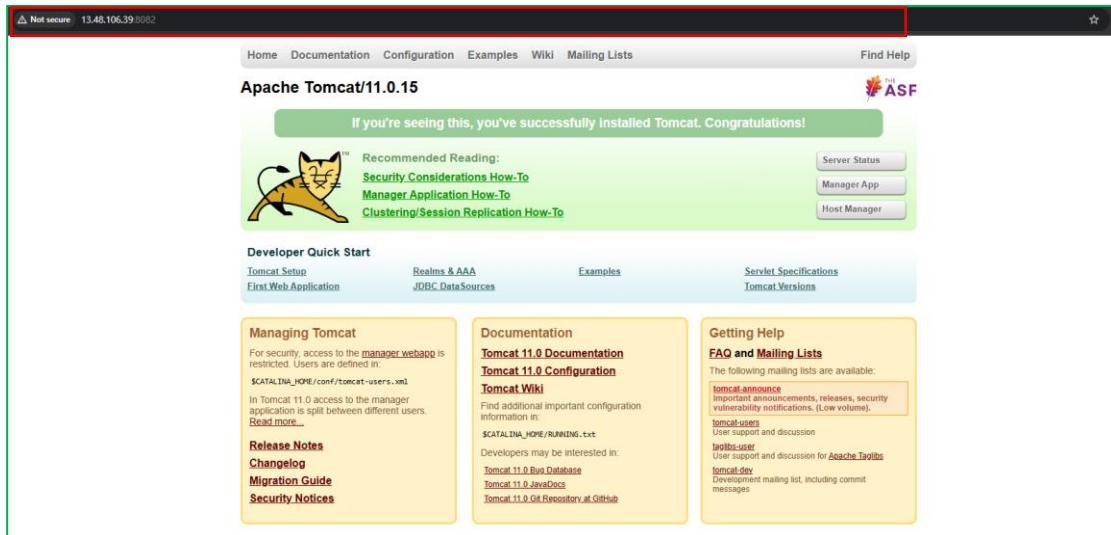
=> => naming to docker.io/library/sample_image 0.0s

```
[root@Y22ACS414Docker ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
sample_image latest 64c8f7c97b1c 7 seconds ago 418MB
tomcat latest 557466b563c2 5 days ago 412MB
```

```
[root@Y22ACS414Docker ~]# docker run -d --name tc2 -p 8082:8080 sample_image
0ae4553e2b844bc0c624116277fa2a16484731fc700baab8f4b83b006f79081a
```

```
[root@Y22ACS414Docker ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
0ae4553e2b84 sample_image "catalina.sh run" 9 seconds ago Up 8 seconds 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp tc2
a5619770e148 tomcat:latest "catalina.sh run" 6 minutes ago Up 6 minutes 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp tc1
```

➤ After running ‘Docker image’ goto browser and search the ‘ipv4:8082’



Lab Experiment – 8

Aim:

Docker CI/CD Pipeline using Jenkins.

Procedure:

Step 1: Create a new user name as docker admin and provide password authentication.

```

      #
~\_ #####
~~ \##### Amazon Linux 2023
~~  \###|
~~   \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~    \~' .-'>
~~     /
~~-. /-
~/m/
[ec2-user@Y22ACS414Docker ~]$ sudo su -
Last login: Sun Dec 14 07:48:16 UTC 2025 on pts/1

```

```
[root@Y22ACS414Docker ~]# useradd dockeradmin
```

```
[root@Y22ACS414Docker ~]# passwd dockeradmin
Changing password for user dockeradmin.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
[root@Y22ACS414Docker ~]# cd /etc/ssh/
```

```
[root@Y22ACS414Docker ssh]# ls
moduli ssh_config ssh_config.d ssh_host_ecdsa_key ssh_host_ecdsa_key.pub
ssh_host_ed25519_key ssh_host_ed25519_key.pub sshd_config sshd_config.d
```

```
[root@Y22ACS414Docker ssh]# vi sshd_config
```

```

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords yes

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
#KerberosAuthentication no

```

➤ Now login to ‘dockeradmin’ user.

```
[root@Y22ACS414Docker ssh]# cd /opt
```

```
[root@Y22ACS414Docker opt]# ls
aws containerd
```

```
[root@Y22ACS414Docker opt]# sudo mkdir docker
```

```
[root@Y22ACS414Docker opt]# ls
aws containerd docker
```

```
[root@Y22ACS414Docker opt]# sudo cp -R /root/Dockerfile /opt/docker
```

```
[root@Y22ACS414Docker opt]# cd docker/
```

```
[root@Y22ACS414Docker docker]# ls
Dockerfile
```

```
[root@Y22ACS414Docker docker]# cat Dockerfile
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps/
COPY ./*.war .usr/local/tomcat/webapps/
```

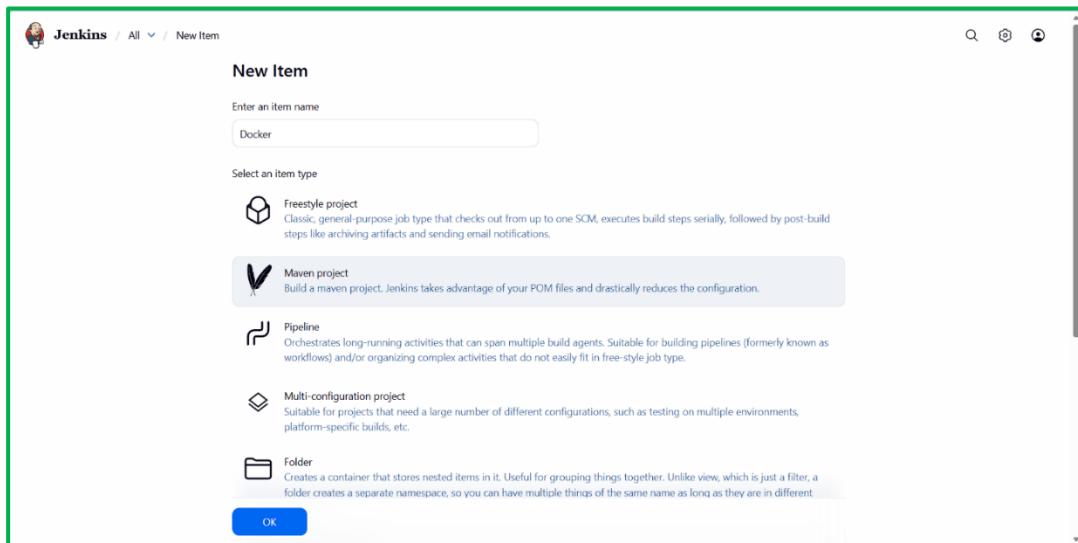
```
[root@Y22ACS414Docker docker]# cd ~
```

```
[root@Y22ACS414Docker ~]# chown -R dockeradmin:dockeradmin /opt/docker
```

```
[root@Y22ACS414Docker ~]# cd /opt/docker/
```

```
[root@Y22ACS414Docker docker]# sudo usermod -aG docker dockeradmin
```

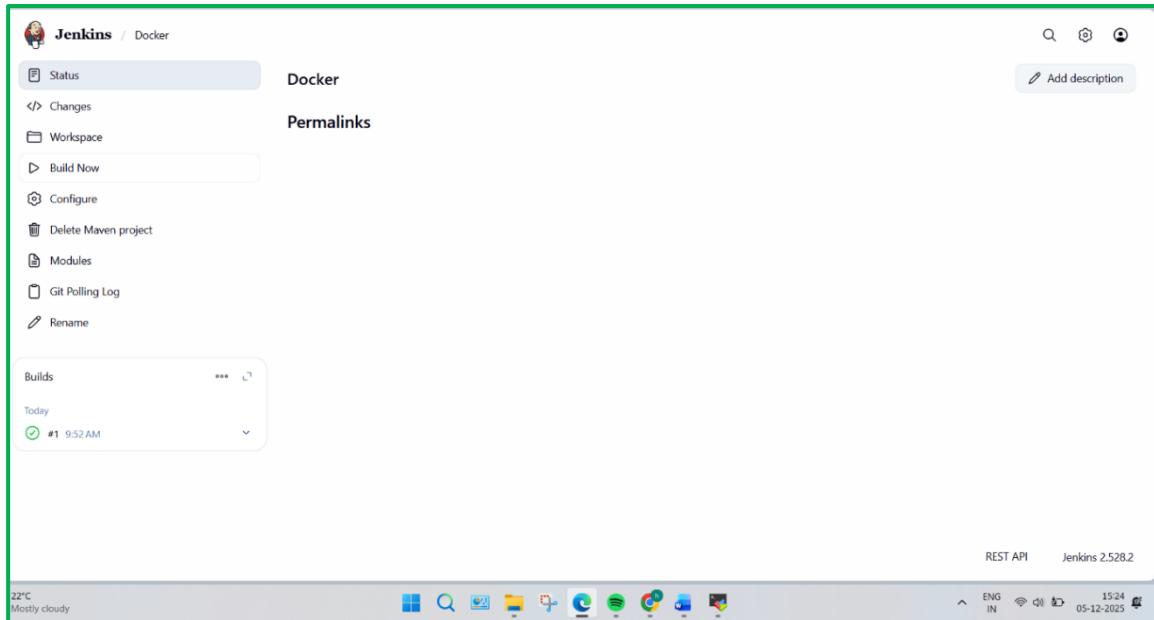
➤ Now start the instance of Jenkins and create a new job



The screenshot shows the Jenkins configuration interface for a project named 'Maven_Project'. The 'Triggers' section is highlighted with a red box. It contains several trigger options: 'Build whenever a SNAPSHOT dependency is built' (checked), 'Schedule build when some upstream has no successful builds', 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub Branches', 'GitHub Pull Requests', and 'GitHub hook trigger for GITScm polling'. Below these is a section titled 'Poll SCM' (checked) with a 'Schedule' field containing '*****'. A note says 'No schedules will only run due to SCM changes if triggered by a post-commit hook'. There is also an unchecked option 'Ignore post-commit hooks'. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

- Add a post build artifact over ssh.
- After typing the below apply and save it.

The screenshot shows the Jenkins configuration interface for a project named 'Docker'. The 'Post-build Actions' section is highlighted with a red box. It contains a 'Transfers' section with the following fields: 'Source files' (target/*.war), 'Remove prefix' (target), 'Remote directory' (/opt/docker), and 'Exec command' (cd /opt/docker; docker stop tc4; docker rm tc4; docker build -t autoupdate; docker run -d --name tc4 -p 8087:8080 autoupdate). A note at the bottom states 'All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables'. At the bottom of the configuration page are 'Save' and 'Apply' buttons.



➤ Output before changing.

A screenshot of a simple website. The header is blue with the text "Welcome to My Website" and "Your one-stop destination for learning HTML!". Below the header is a navigation bar with "Home", "About", and "Contact" links. The main content area has a heading "Home" and a paragraph: "This is the homepage of my simple website. Here, you can find various resources and information.". The "About" section contains: "This website is built using HTML and CSS to demonstrate clean and responsive design practices." The "Contact" section contains: "Feel free to reach out via email at example@example.com". At the bottom is a dark footer bar with the copyright notice: "© 2025 My Simple Webpage. All rights reserved."

➤ Output after changing in github.

Jenkins / Docker

Status

Docker

Permalinks

Add description

Builds

Pending

#2 Finished waiting

Today #1 9:52 AM

REST API Jenkins 2.528.2

Jenkins / Docker

Status

Docker

Permalinks

Add description

Builds

Today

#2 9:56 AM

#1 9:52 AM

REST API Jenkins 2.528.2

Registration Form

Full Name

Email

Phone Number

Gender

Address

Submit

Lab Experiment – 9

Aim:

Complete Docker CI/CD pipeline using Jenkins and Ansible.

Procedure:

Step 1: Setup the docker instance by creating a new user named as “ansadmin”.

- Allocate sudo permissions to it.

```
[root@Y22ACS414Docker ~]# useradd ansadmin
```

```
[root@Y22ACS414Docker ~]# passwd ansadmin
```

Changing password for user ansadmin.

New password:

BAD PASSWORD: The password contains the user name in some form

Retype new password:

passwd: all authentication tokens updated successfully.

```
[root@Y22ACS414Docker ~]# visudo
```

```
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
## Allow ansadmin to run any commands anywhere
ansadmin  ALL=(ALL)  ALL
```

Step 2: Create a new instance for Ansible and set the security groups same as docker.

- Now connect it to Mobxterm using the key pair.

```
[root@Y22ACS414ANSIBLE ~]# useradd ansadmin
```

```
[root@Y22ACS414ANSIBLE ~]# passwd ansadmin
```

Changing password for user ansadmin.

New password:

BAD PASSWORD: The password contains the user name in some form

Retype new password:

passwd: all authentication tokens updated successfully.

```
[root@Y22ACS414ANSIBLE ~]# cd /etc/
```

```
[root@Y22ACS414ANSIBLE etc]# cd ssh/
```

```
[root@Y22ACS414ANSIBLE ssh]# ls
```

moduli ssh_config ssh config.d ssh_host_ecdsa_key ssh_host_ecdsa_key.pub ssh_host_ed25519_key
ssh_host_ed25519_key.pub sshd_config

```
[root@Y22ACS414ANSIBLE ssh]# vi sshd_config
```

```
[root@Y22ACS414-ANSIBLE ssh]# visudo
```

```
[root@Y22ACS414ANSIBLE ssh]# cd ...
```

```
[root@Y22ACS414ANSIBLE etc]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
```

```
##           user      MACHINE=COMMANDS
##           user      MACHINE=COMMANDS
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root      ALL=(ALL)      ALL
ansadmin  ALL=(ALL)      ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys  ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansadmin/.ssh/id_rsa):
Created directory '/home/ansadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansadmin/.ssh/id_rsa
Your public key has been saved in /home/ansadmin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:mJsHqWk87/yNdxZm1iueTYAy+G4ezfcwJ09Wn7Cq39E ansadmin@Y22ACS415Ansible
The key's randomart image is:
+---[RSA 3072]----+
|          .          |
|          .          |
|          =          |
|         * S o o .   |
| . o = = +o+       |
| * o + o+=++E.     |
| . + o.+ .o*o...   |
| .++=o+B=o...     |
+---[SHA256]-----+
```

```
[root@Y22ACS414ANSIBLE ~]# sudo su - ansadmin
Last login: Sat Dec  6 17:07:58 UTC 2025 on pts/2
```

```
[ansadmin@Y22ACS414ANSIBLE ~]$ sudo vi /etc/ansible/hosts
[sudo] password for ansadmin:
```

```
[ansadmin@Y22ACS414ANSIBLE ~]$ ssh-copy-id 172.31.11.211
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.11.211 (172.31.11.211)' can't be established.
ED25519 key fingerprint is SHA256:/fLInIU3FY/TBWKu+6qVbSI0tgc1ncyKSNC8wojkU5w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the
new keys
ansadmin@172.31.11.211's password:
Permission denied, please try again.
ansadmin@172.31.11.211's password:
Permission denied, please try again.
ansadmin@172.31.11.211's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.11.211'"
 and check to make sure that only the key(s) you wanted were added.

```
[ansadmin@Y22ACS414ANSIBLE ~]$ ansible all -m ping
[WARNING]: Platform linux on host 172.31.11.211 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.11.211 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
```

```
[ansadmin@Y22ACS414ANSIBLE ~]$ cd /opt
```

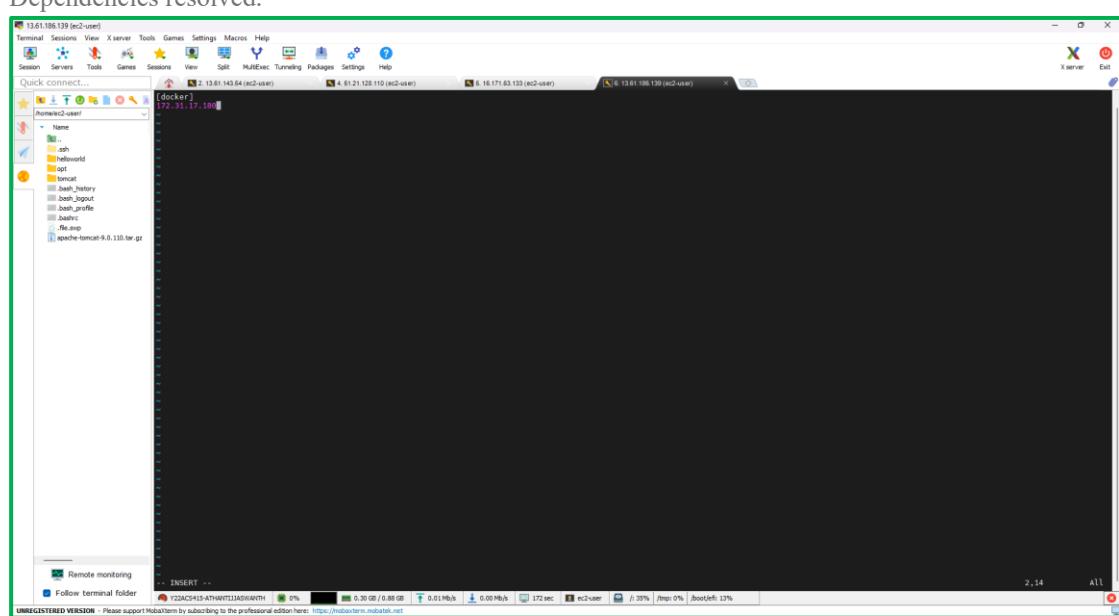
```
[ansadmin@Y22ACS414ANSIBLE opt]$ ls
aws
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ sudo mkdir docker
[sudo] password for ansadmin:
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ ls
aws docker
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ sudo chown -R ansadmin:ansadmin /opt/docker
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ sudo yum install docker -y
Last metadata expiration check: 0:11:37 ago on Sat Dec 6 17:09:50 2025.
Dependencies resolved.
```



```
[ansadmin@Y22ACS414ANSIBLE opt]$ sudo usermod -aG docker ansadmin
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ service docker start
```

```
Redirecting to /bin/systemctl start docker.service
```

```
Failed to start docker.service: Access denied
```

```
See system logs and 'systemctl status docker.service' for details.
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ sudo service docker start
```

```
Redirecting to /bin/systemctl start docker.service
```

```
[ansadmin@Y22ACS414ANSIBLE opt]$ cd docker/
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ ls
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ vi Dockerfile
```

```
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps/
COPY ./*.war /usr/local/tomcat/webapps/
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ ls
```

```
Dockerfile
```

Step 3: Start Jenkins instance.

- Add another new ssh in manager Jenkins. Create a job in it.

The screenshot shows two Jenkins configuration pages. The top page is titled 'SSH Server' and contains fields for 'Name' (Ansible), 'Hostname' (172.31.0.238), 'Username' (ansadmin), and 'Remote Directory'. It includes an 'Advanced' section with a checked checkbox for password authentication and a 'Save' button. The bottom page shows a 'Proxy password' field with a redacted value and a 'Success' message, with a 'Test Configuration' button.

➤ A new job to push maven_project.

The screenshot shows the Jenkins 'New Item' creation interface. A green box highlights the 'Maven project' option under the 'Select an item type' section. The 'Maven project' description states: 'Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.' Other options shown include 'Freestyle project', 'Pipeline', 'Multi-configuration project', 'Folder', 'Multibranch Pipeline', and 'Organization Folder'. A 'Copy from' dropdown and an 'OK' button are also visible.

The screenshot shows the 'SSH Publishers' configuration page for the 'Ansible' job. A green box highlights the 'Transfer Set' section. It includes fields for 'Source files' (set to 'target/*.war'), 'Remove prefix' (set to 'target'), 'Remote directory' (set to '//opt/docker'), and 'Exec command'. Buttons for 'Save' and 'Apply' are at the bottom.

The screenshot shows the Jenkins job details page for 'Ansible_Docker'. A green box highlights the left sidebar with options like 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Maven project', 'Modules', and 'Rename'. The 'Builds' section shows a single build entry from today at 5:43PM. The bottom right corner displays 'REST API' and 'Jenkins 2.528.2'.

Step 4: Build an image and a running container.

```
[ansadmin@Y22ACS414ANSIBLE docker]$ ls
Dockerfile
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ docker images
```

```
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock:
Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect:
permission denied
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ sudo docker build -t image .
```

```
[sudo] password for ansadmin:
[+] Building 14.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 221B
               docker:default          0.0s
                               0.0s
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ sudo docker images
```

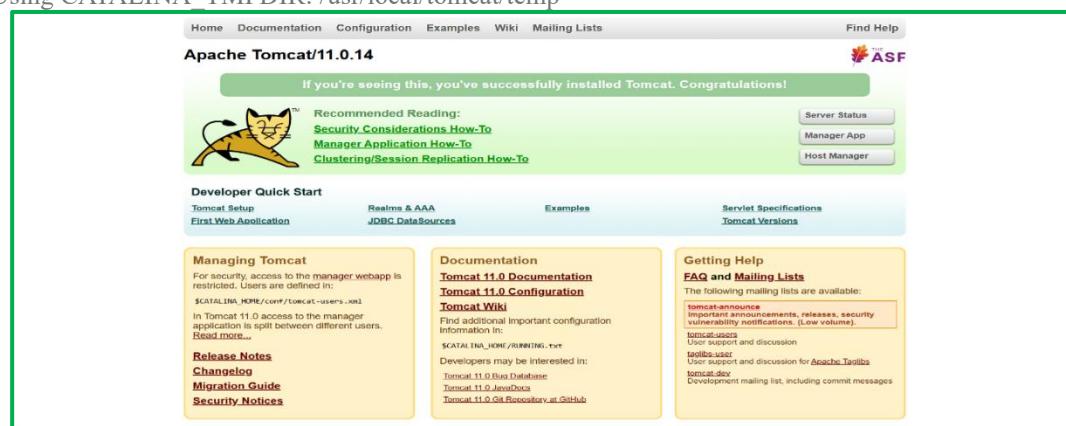
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
image	latest	6b91c8dcc5eb	18 seconds ago	418MB

```
[ansadmin@Y22ACS414ANSIBLE docker]$ sudo docker run -t --name tc1 -p 8081:8080 image
```

Using CATALINA_BASE: /usr/local/tomcat

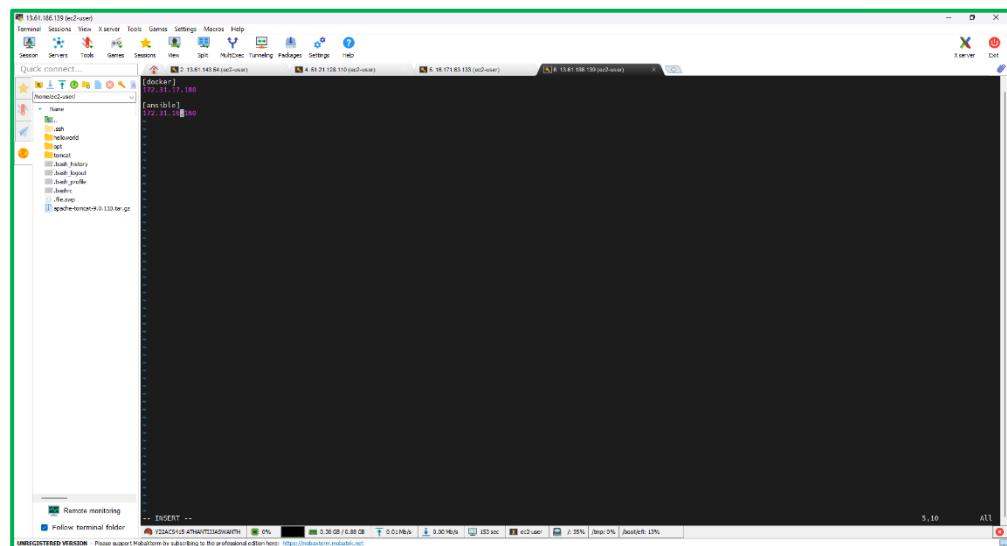
Using CATALINA_HOME: /usr/local/tomcat

Using CATALINA_TMPDIR: /usr/local/tomcat/temp



Step 5: Adding ansible private ip to hosts file.

```
[ansadmin@Y22ACS414ANSIBLE ~]$ sudo vi /etc/ansible/hosts
[sudo] password for ansadmin:
```



```
[ansadmin@Y22ACS414ANSIBLE ~]$ ssh-copy-id 172.31.0.238
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.0.238 (172.31.0.238)' can't be established.
ED25519 key fingerprint is SHA256:pSmcAUaoGWawQBw/3/FtCaVaqeNsLiSjqjLLMk+7AZw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the
new keys
ansadmin@172.31.0.238's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.0.238'"
and check to make sure that only the key(s) you wanted were added.

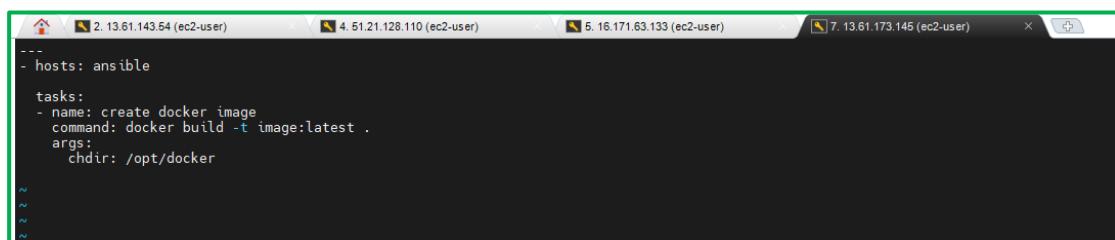
```
[ansadmin@Y22ACS414ANSIBLE ~]$ ansible all -m ping
[WARNING]: Platform linux on host 172.31.11.211 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.11.211 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.0.238 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.0.238 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[ansadmin@Y22ACS414ANSIBLE ~]$ cd .ssh
```

```
[ansadmin@Y22ACS414ANSIBLE .ssh]$ ls
authorized_keys id_rsa id_rsa.pub known_hosts known_hosts.old
```

Step 6: Creating a playbook named as mavenbuild.yml.

```
[ansadmin@Y22ACS414ANSIBLE .ssh]$ cd /opt/docker/
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ ls
Dockerfile maven_project.war
[ansadmin@Y22ACS414ANSIBLE docker]$ vi mavenbuild.yml
```



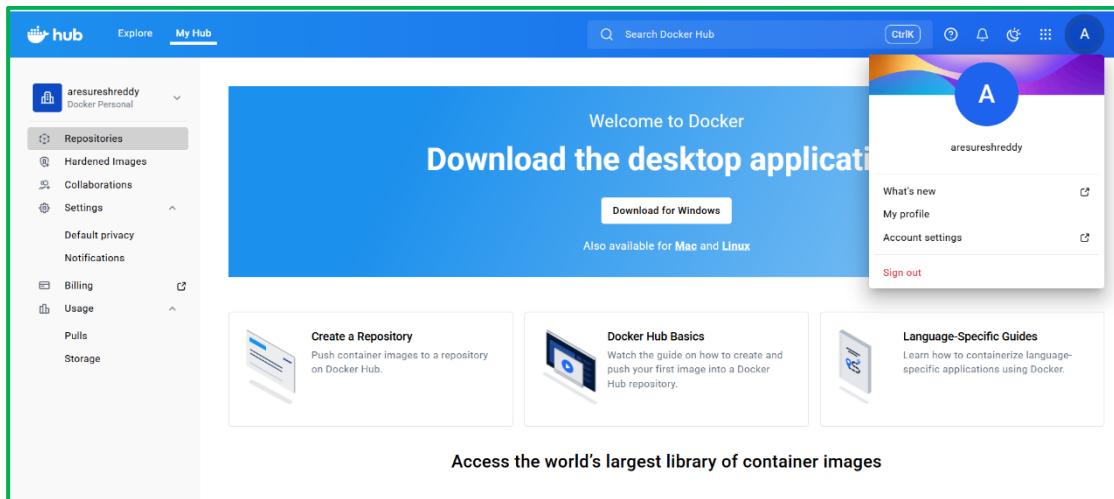
```
hosts: ansible
tasks:
- name: create docker image
  command: docker build -t image:latest .
  args:
    chdir: /opt/docker
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ ansible-playbook mavenbuild.yml
PLAY [ansible]
*****
TASK [Gathering Facts]
*****
[WARNING]: Platform linux on host 172.31.0.238 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.0.238]

TASK [create docker image]
*****
changed: [172.31.0.238]
PLAY RECAP
*****
172.31.0.238      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Step 7: Copy the image onto Docker Hub.

➤ Create a docker hub account.



```
[ansadmin@Y22ACS414ANSIBLE docker]$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
image1 latest 6b91c8dcc5eb 32 minutes ago 418MB
image latest 6b91c8dcc5eb 32 minutes ago 418MB
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ docker login
```

Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com/> to create one.

You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at <https://docs.docker.com/go/access-tokens/>

Username: athantijaswanth

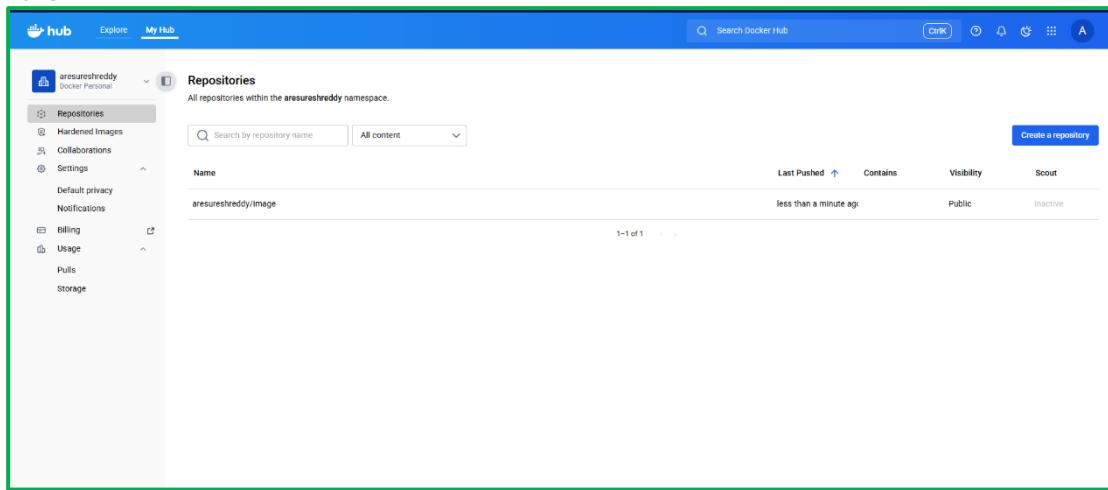
Password:

WARNING! Your password will be stored unencrypted in /home/ansadmin/.docker/config.json. Configure a credential helper to remove this warning. See <https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

```
[ansadmin@Y22ACS414ANSIBLE docker]$ docker tag 6b91c8dcc5eb athantijaswanth/image
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ docker push athantijaswanth/image
Using default tag: latest
The push refers to repository [docker.io/y22acs539/image]
fa685c57fff5: Pushed
0d2cf7735641: Pushed
5f70bf18a086: Pushed
1dea0526eeac: Pushed
2c019e378af7: Pushed
448e27e229f0: Pushed
f9e5f9899ee3: Pushed
818ff17bf41c: Pushed
e8bce0aabd68: Pushed
latest: digest: sha256:66ed96f02a0818b18fbff3abb5d74a8b63d43b5c5084f376e9570d3fdbd68605a size: 2618
```



Step 8: Updating the playbook code for tag building.

```
[ansadmin@Y22ACS414ANSIBLE docker]$ vi mavenbuild.yml
```

```
[ansadmin@Y22ACS414ANSIBLE docker]$ ansible-playbook mavenbuild.yml --check
PLAY [ansible]
*****
TASK [Gathering Facts]
*****
[WARNING]: Platform linux on host 172.31.0.238 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.0.238]
skipping: [172.31.0.238]
TASK [create tag to push onto dockerhub]
*****
skipping: [172.31.0.238]
TASK [Push docker image]
*****
skipping: [172.31.0.238]
PLAY RECAP
*****
172.31.0.238 : ok=1    changed=0   unreachable=0   failed=0   skipped=3   rescued=0   ignored=0
```

Step 9: Now executing from direct Jenkins job.

Top Left Screenshot: A terminal window titled "65.0.179.66 (ec2-user)" showing an Ansible playbook named "hosts: ansible". The playbook contains tasks to build a Docker image, push it to DockerHub, and push it to a specific repository.

```

- hosts: ansible
  tasks:
    - name: create docker image
      command: docker build -t image2:latest .
      args:
        chdir: /opt/docker
    - name: create tag to push onto dockerhub
      command: docker tag image2:latest AresureshReddy/image2:latest
    - name: Push docker image
      command: docker push AresureshReddy/image2:latest
  
```

Top Right Screenshot: A Jenkins configuration screen for an Ansible job. It shows the "Transfers" section where source files "target/*.war" are being transferred to a remote directory "//opt/docker" with a remove prefix "target". The "Exec command" field contains the command "ansible-playbook /opt/docker/mavenbuild.yml".

Bottom Left Screenshot: A Jenkins build history for the "Ansible_Docker" job. It shows four builds: #1 (stable), #2 (stable), #3 (stable), and #4 (stable). All builds were successful and occurred within the last 2 minutes.

Bottom Right Screenshot: A screenshot of the Docker Hub interface showing three repositories under the "aresureshreddy" namespace: "aresureshreddy/image2", "aresureshreddy/image1", and "aresureshreddy/image".

➤ Till now we have connected Ansible to Docker Hub.

Step 10: Connecting docker hub to docker.

```

hosts: docker
tasks:
- name: stop container
  command: docker stop container
  ignore_errors: yes
- name: remove
  command: docker rm container
  ignore_errors: yes
- name: remove image
  command: docker rmi AreSureshReddy/image2:latest
  ignore_errors: yes
- name: Create Container
  command: docker run -d --name container -p 8082:8080 AreSureshReddy/image2:latest

```

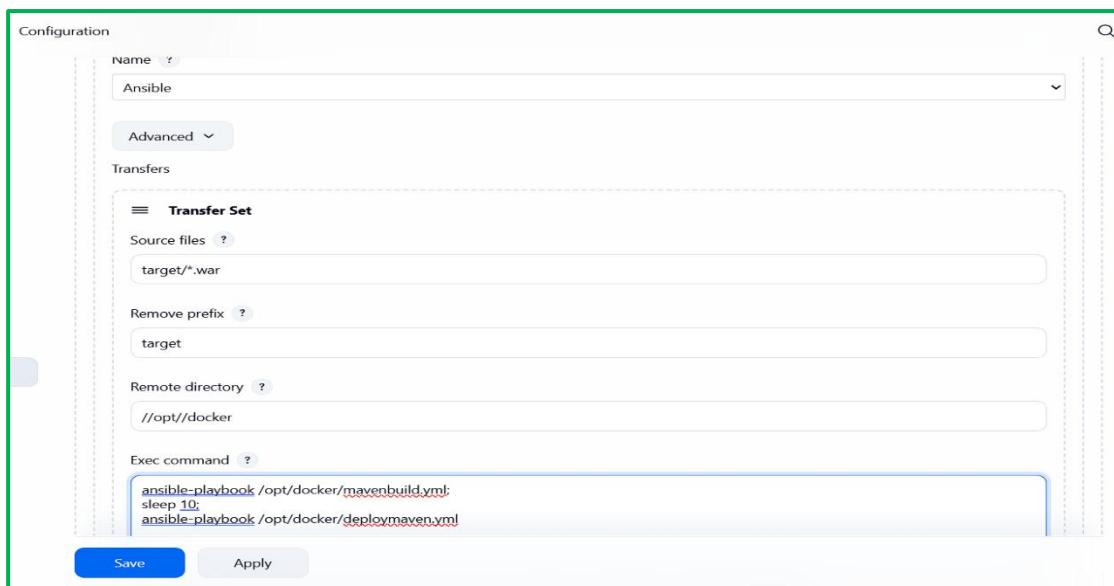
```

[ansadmin@Y22ACS414ANSIBLE docker]$ ansible-playbook deploymaven.yml
PLAY [docker]
*****
TASK [Gathering Facts]
*****
[WARNING]: Platform linux on host 172.31.11.211 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.11.211]
TASK [Create Container]
*****
changed: [172.31.11.211]
PLAY RECAP
*****
172.31.11.211 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

```

➤ Run this sudo chmod 777 /var/run/docker.sock

Step 11: Deploy container using playbook.



Jenkins / Ansible_Docker

Status **Ansible_Docker** **Add description**

Permalinks

- Last build (#2), 10 hr ago
- Last stable build (#2), 10 hr ago
- Last successful build (#2), 10 hr ago
- Last completed build (#2), 10 hr ago

Builds **Filter**

Today

- #3 5:14 AM
- December 6, 2025
- #2 6:46 PM
- #1 5:43 PM

hub Explore My Hub

Repositories All repositories within the **areshreddy** namespace.

Create a repository

Name	Last Pushed	Contains	Visibility	Scout
areshreddy/image2	less than a minute ago		Public	Inactive
areshreddy/image1	less than a minute ago		Public	Inactive
areshreddy/image	10 minutes ago		Public	Inactive

Jenkins / Ansible_Docker

Status **Ansible_Docker** **Add description**

Permalinks

- Last build (#3), 9 min 57 sec ago
- Last stable build (#3), 9 min 57 sec ago
- Last successful build (#3), 9 min 57 sec ago
- Last completed build (#3), 9 min 57 sec ago

Builds **Filter**

Pending

- #4 Executor slot already in use

Today

- #3 5:14 AM
- December 6, 2025

The screenshot shows a web-based application interface for a project named "MyProject". At the top, there's a navigation bar with links for "Features", "Gallery", "Contact", and a prominent blue "Get Started" button. Below the navigation, a main heading reads "Deploy. Preview. Repeat. ⚡ Fast." followed by a brief description: "Simple Playbook + Docker workflow to build and run your app. This starter page includes hero, features, gallery and contact ☀️ all in a single file." There are two primary feature sections: "Automate" and "Containerize". The "Automate" section describes using Ansible or simple scripts for deployments. The "Containerize" section describes packaging the app with Docker for predictable runtime and scaling. To the right, there's a preview area titled "App Preview" showing a placeholder "Coming Soon" message, with sub-labels "v1.0.0 Live Preview" and "Public". At the bottom left, a "Gallery" section is visible.