

1. Login page of Jenkins



Sign in to Jenkins

Username

Password

☐ Keep me signed in

Sign in

2. Configuration Setting in Jenkins

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/sanjaykumarm2001/devops-build/

Credentials ?

DOCKERHUB_CREDENTIALS

+ Add

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

build/Jenkinsfile

3.Executed step command

```
pipeline {
  agent any

  stages {
    stage('Build and Push Docker Image') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'DOCKERHUB_CREDENTIALS', usernameVariable:
        script {
          // List files in the build directory to confirm location
          sh 'ls -la build'

          // Use relative path to deploy.sh and ensure it's executable
          sh 'chmod +x build/deploy.sh'
          sh './build/deploy.sh'
        }
      }
    }
  }
}
```

✓ Console Output

[Download](#)[Copy](#)[View as plain text](#)

```
Started by user admin
Obtained build/Jenkinsfile from git https://github.com/sanjaykumarm2001/devops-build/
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Guvi-Project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential DOCKERHUB_CREDENTIALS
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Guvi-Project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sanjaykumarm2001/devops-build/ # timeout=10
Fetching upstream changes from https://github.com/sanjaykumarm2001/devops-build/
> git --version # timeout=10
> git --version # 'git version 2.43.0'
using GIT ASKPASS to set credentials DOCKERHUB_CREDENTIALS
> git fetch --tags --force --progress -- https://github.com/sanjaykumarm2001/devops-build/
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/dev^{commit} # timeout=10
Checking out Revision 4932d2f8b90e73563f0b512ffa7edfa31887093a (refs/remotes/origin/dev)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4932d2f8b90e73563f0b512ffa7edfa31887093a # timeout=10
Commit message: "chnages"
> git rev-list --no-walk 4932d2f8b90e73563f0b512ffa7edfa31887093a # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build and Push Docker Image)
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKER_USERNAME or $DOCKER_PASSWORD
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ ls -la build
total 84
drwxr-xr-x 4 jenkins jenkins 4096 Aug 19 18:50 .
```

```
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 388B 0.0s done
#1 DONE 0.0s

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

#3 [internal] load metadata for docker.io/library/nginx:latest
#3 DONE 0.3s

#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [1/3] FROM
docker.io/library/nginx:latest@sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 1.66kB 0.0s done
#6 DONE 0.1s

#7 [2/3] COPY . /usr/share/nginx/html/
```

```
#7 CACHED
```

```
#8 [3/3] RUN ls -la /usr/share/nginx/html/
#8 CACHED
```

```
#9 exporting to image
#9 exporting layers done
#9 writing image sha256:58dc4967eefcadec3389151112c9b845dc53d0892f2cea41dc0e0f1a1046901d done
#9 naming to docker.io/library/react-app 0.0s done
#9 DONE 0.0s
```

WARNING! Using --password via the CLI is insecure. Use --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/#credential-stores>

Login Succeeded

Using default tag: latest

The push refers to repository [docker.io/****/dev]

63dfba7f44b8: Preparing

9604fe824b89: Preparing

5f0272c6e96d: Preparing

f4f00eaedec7: Preparing

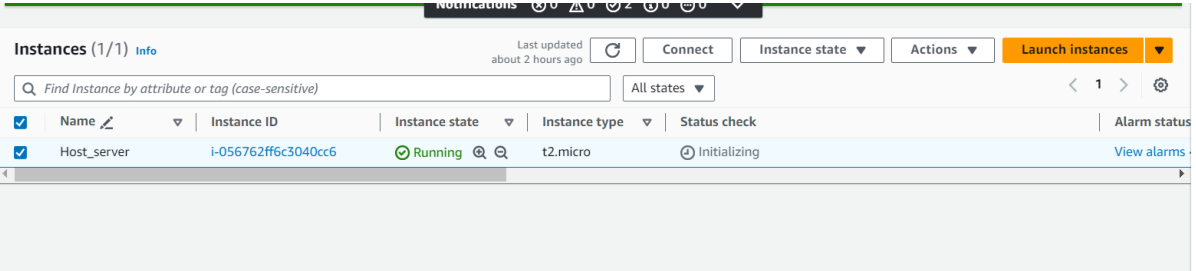
55e54df86207: Preparing

```

ec1a2ca4ac87: Preparing
8b87c0c66524: Preparing
72db5db515fd: Preparing
9853575bc4f9: Preparing
ec1a2ca4ac87: Waiting
8b87c0c66524: Waiting
72db5db515fd: Waiting
9853575bc4f9: Waiting
55e54df86207: Layer already exists
f4f00eaedec7: Layer already exists
9604fe824b89: Layer already exists
63dfba7f44b8: Layer already exists
5f0272c6e96d: Layer already exists
ec1a2ca4ac87: Layer already exists
9853575bc4f9: Layer already exists
72db5db515fd: Layer already exists
8b87c0c66524: Layer already exists
latest: digest: sha256:d6b84843176954dd66b3b0a21bec9a1d79903dcd0ae32d15b83aa4d85d1c461a size: 2194
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

4. EC2 Console and SG Configs



The screenshot shows the AWS Management Console interface for EC2 Instances. At the top, there's a navigation bar with 'Instances (1/1)' and 'Info' tabs. Below this, there's a search bar and a table of instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. One instance is listed: 'Host_server' with ID 'i-056762ff6c3040cc6', state 'Running', type 't2.micro', and status check 'Initializing'. There are also buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Host_server	i-056762ff6c3040cc6	Running	t2.micro	Initializing	View alarms

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-016b69083974d9a00	SSH ▼	TCP	22	Cus... ▼	<input type="text" value="Q"/> <input data-bbox="914 331 1067 365" type="text" value="183.82.179.67/32"/>	<input type="text"/> <input type="button" value="Delete"/>
sgr-04b38ae4f20e81a3c	HTTPS ▼	TCP	443	Cus... ▼	<input type="text" value="Q"/> <input data-bbox="914 416 1010 450" type="text" value="0.0.0.0/0"/>	<input type="text"/> <input type="button" value="Delete"/>
sgr-0d4f2451fcb31b6e6	Custom TCP ▼	TCP	8080	Cus... ▼	<input type="text" value="Q"/> <input data-bbox="914 501 1010 535" type="text" value="0.0.0.0/0"/>	for jenkins <input type="text"/> <input type="button" value="Delete"/>
sgr-0b464e40f01dabcf	HTTP ▼	TCP	80	Cus... ▼	<input type="text" value="Q"/> <input data-bbox="914 586 1010 620" type="text" value="0.0.0.0/0"/>	<input type="text"/> <input type="button" value="Delete"/>

5. Docker Hub Repo with Image Tags

sanjaykumar2001 / [Repositories](#) / [dev](#) / [General](#)
Using 1 of 1 private repositories.

[General](#)
[Tags](#)
[Builds](#)
[Collaborators](#)
[Webhooks](#)
[Settings](#)

sanjaykumar2001/dev
Updated about 2 hours ago

This repository does not have a description [INCOMPLETE](#)

This repository does not have a category [INCOMPLETE](#)

Docker commands [Public View](#)

To push a new tag to this repository:

```
docker push sanjaykumar2001/dev:tagname
```

Tags

This repository contains 1 tag(s).

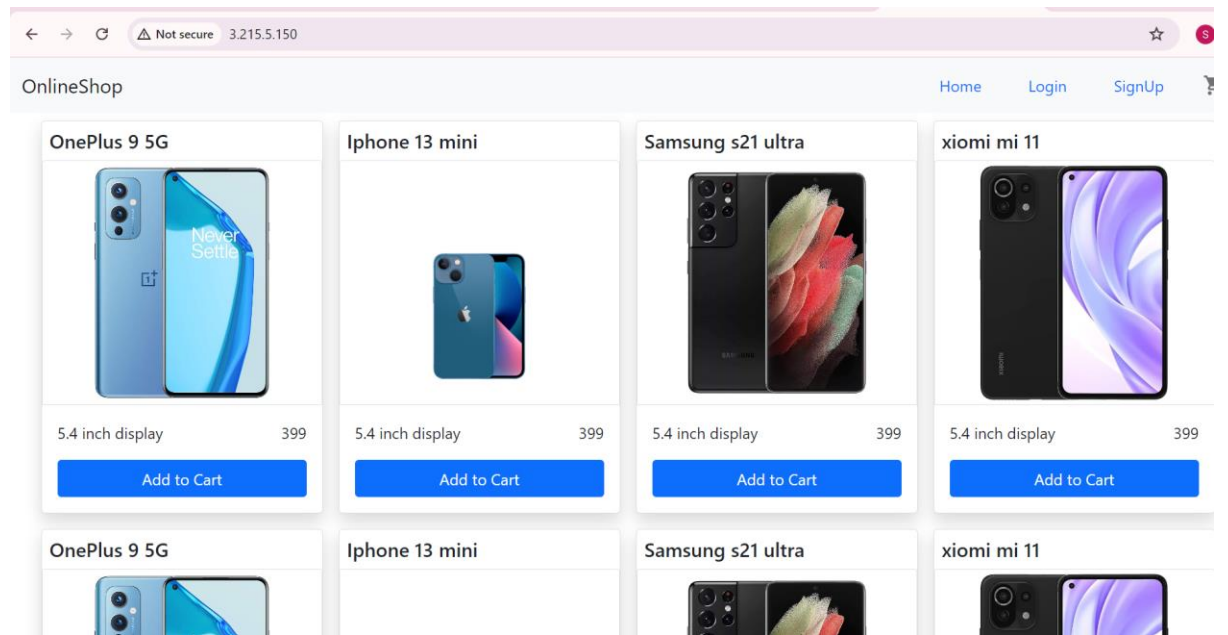
Tag	OS	Type	Pulled	Pushed
latest		Image	an hour ago	2 hours ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

6. Deployed site page



7. Monitoring Health check status

