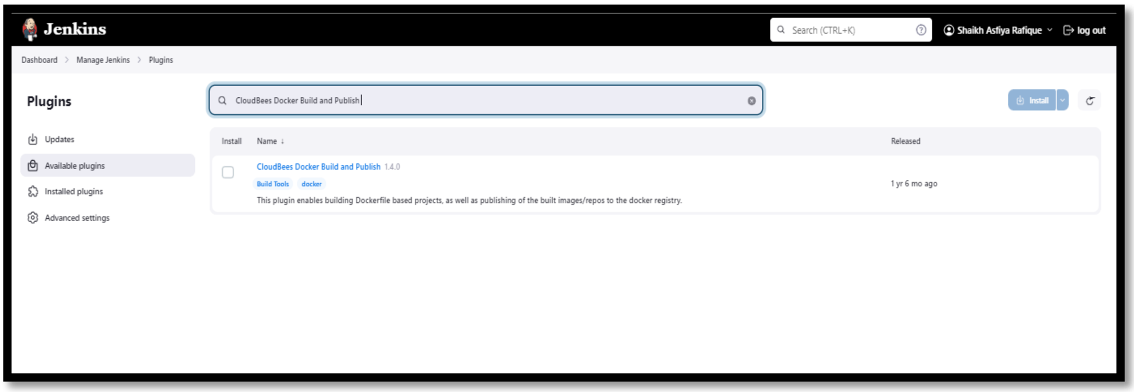
ASSIGNMENT NO:02

==

Q1. Create a Jenkins file to construct and upload Docker images to your Docker-hub registries. Ensure that when the branch is 'dev', the image is constructed and uploaded to the DEV Docker-hub registry. Similarly, when the branch is 'QA', it should be sent to the QA Docker-hub registry.

SOLUTION:

INSTALL PLUGIN IN JENKINS

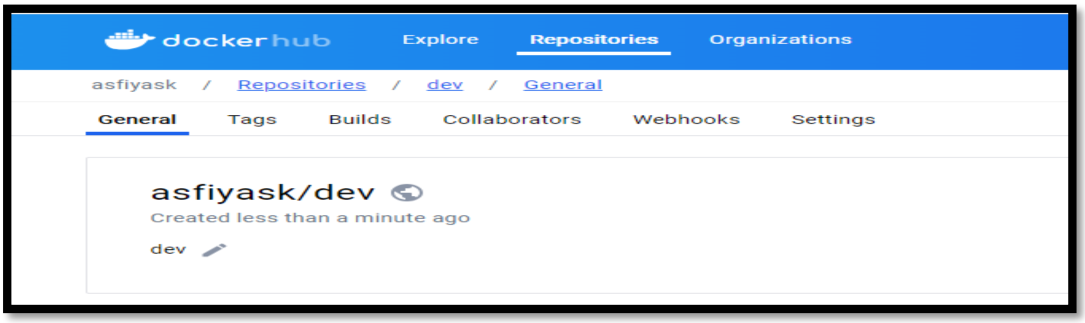


CloudBees Docker Build and Publish

Install	Name ↓	Released
□	CloudBees Docker Build and Publish 1.4.0 Build Tools docker This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.	1 yr 6 mo ago

CREATE DOCKER REPOSITORY

DEV REPOSITORY

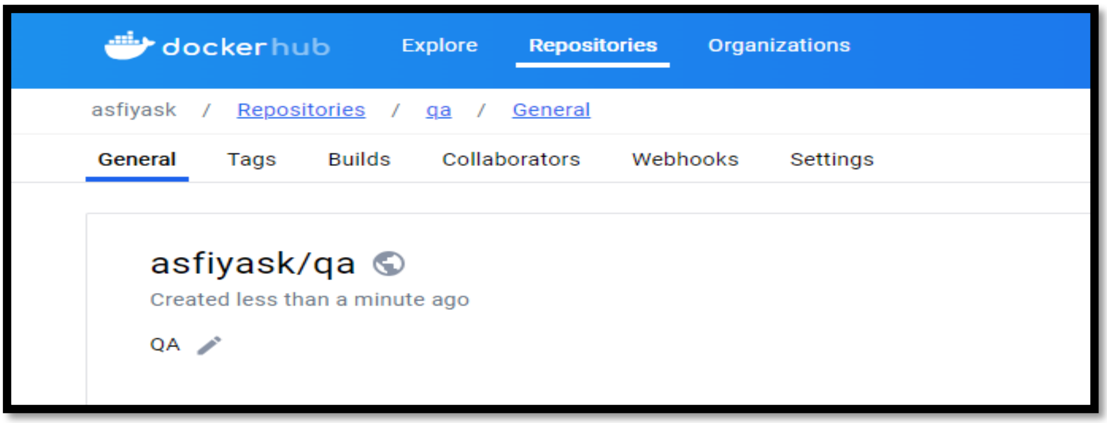


asfiyask/dev

Created less than a minute ago

dev

QA REPOSITORY

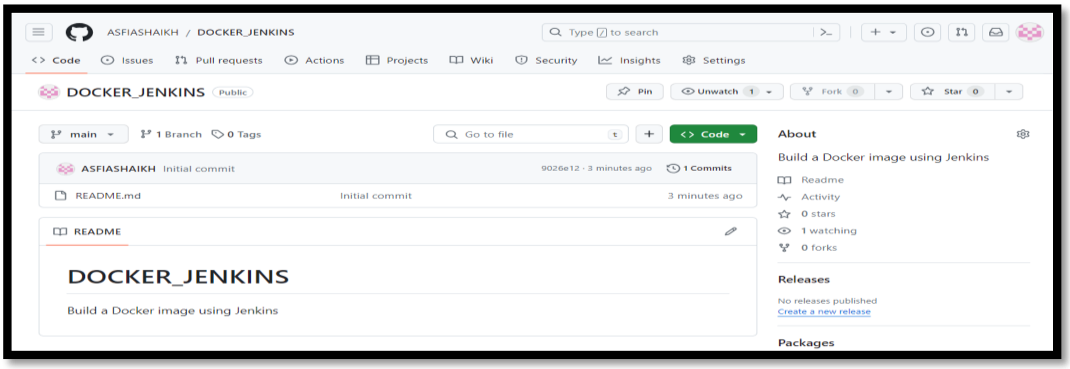


asfiyask/qa

Created less than a minute ago

QA

CREATE GIT REPOSITORY



ASFIASHAIKH Initial commit 9026e12 · 3 minutes ago 1 Commits

README.md Initial commit 3 minutes ago

DOCKER_JENKINS

Build a Docker image using Jenkins

About: Build a Docker image using Jenkins

No releases published
Create a new release

```
root@DESKTOP-8OOG2HF:JENKINS# git clone git@github.com:ASFIASHAIKH/DOCKER_JENKINS.git
Cloning into 'DOCKER_JENKINS'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

```
root@DESKTOP-8OOG2HF:JENKINS# ll
drwxrwxrwx 1 asfiya asfiya    512 Mar 15 11:18 DOCKER_JENKINS/
```

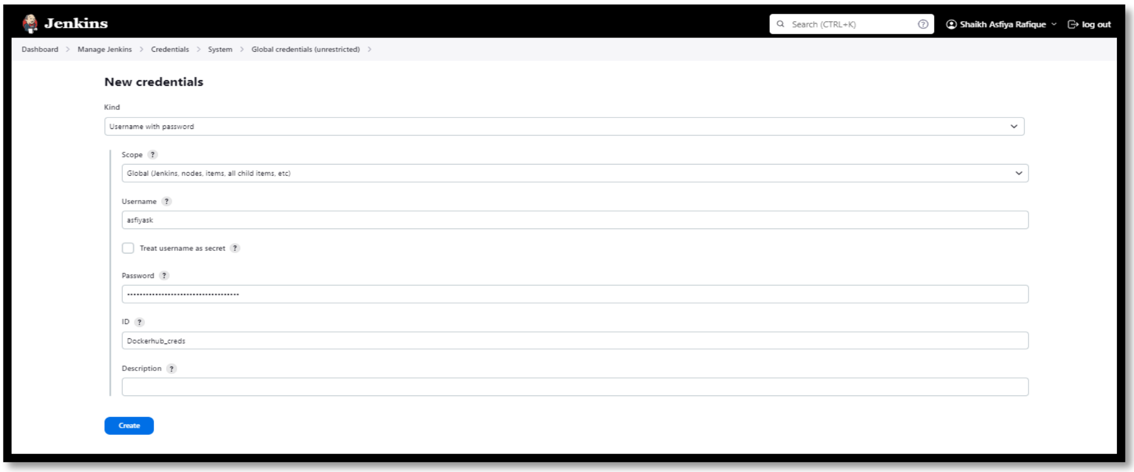
```
root@DESKTOP-8OOG2HF:JENKINS# cd DOCKER_JENKINS/
```

```
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# ll
drwxrwxrwx 1 asfiya asfiya 512 Mar 15 11:18 .git/
-rwxrwxrwx 1 asfiya asfiya  52 Mar 15 11:18 README.md*
```

```
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# vim Dockerfile
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# cat Dockerfile
FROM nginx:alpine

COPY default.conf /etc/nginx/conf.d/
COPY index.html /usr/share/nginx/html/
```

```
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# vim default.conf
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# cat default.conf
server {
    listen       80;
    server_name  localhost;
    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
    ssi on;
}
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# vim index.html
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# cat index.html
<html>
<body>
        <h1>Host: <!--#echo var="HOSTNAME" --></h1>
        Version: 1.1
</body>
</html>
```

Create Jenkins Credentials for Docker hub


Jenkins
Search (CTRL+K)
Shaikh Asfiya Rafique
log out
Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >
Create

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

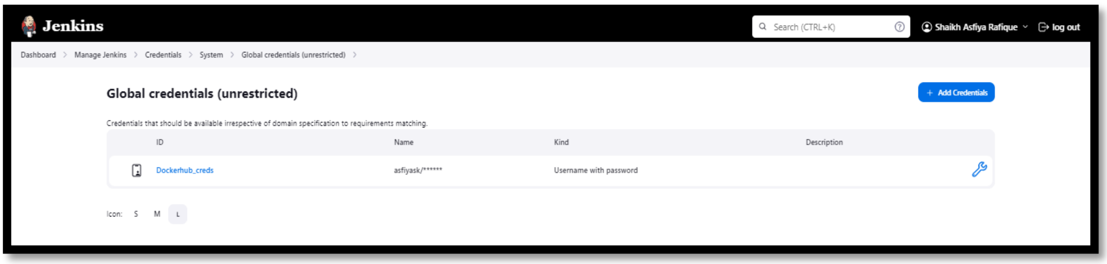
Username: asfiyask

□ Treat username as secret

Password: ••••••••••••••••••••••••••••••••••••

ID: Dockerhub_creds

Description:

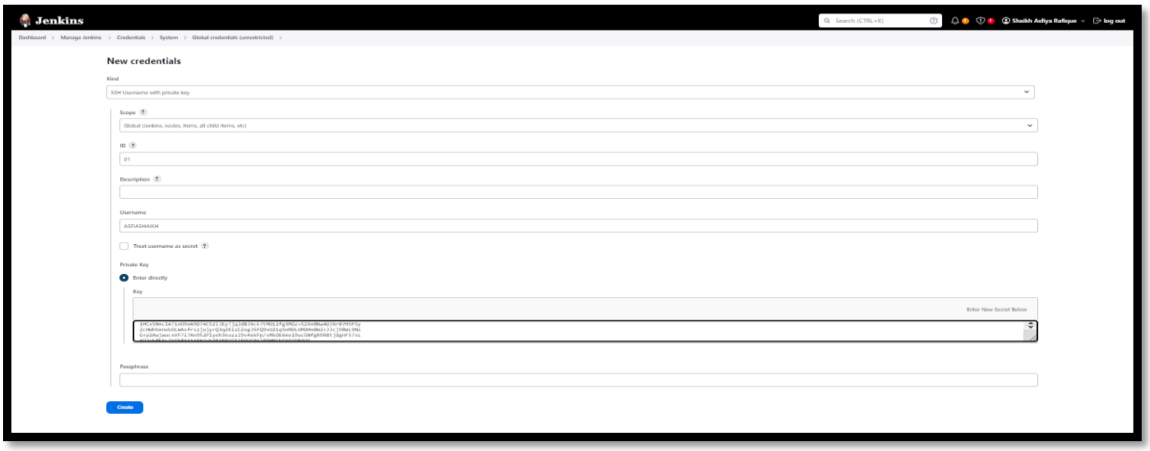

Jenkins
Search (CTRL+K)
Shaikh Asfiya Rafique
log out
Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >
+ Add Credentials
Icon: S M L

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
Dockerhub_creds	asfiyask/******	Username with password	

Create new Credentials for SSH key and Private key


Jenkins
Search (CTRL+K)
Shaikh Asfiya Rafique
log out
Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >
Create

New credentials

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: 01

Description:

Username: ASFIASHAIKH

□ Treat username as secret

Private Key

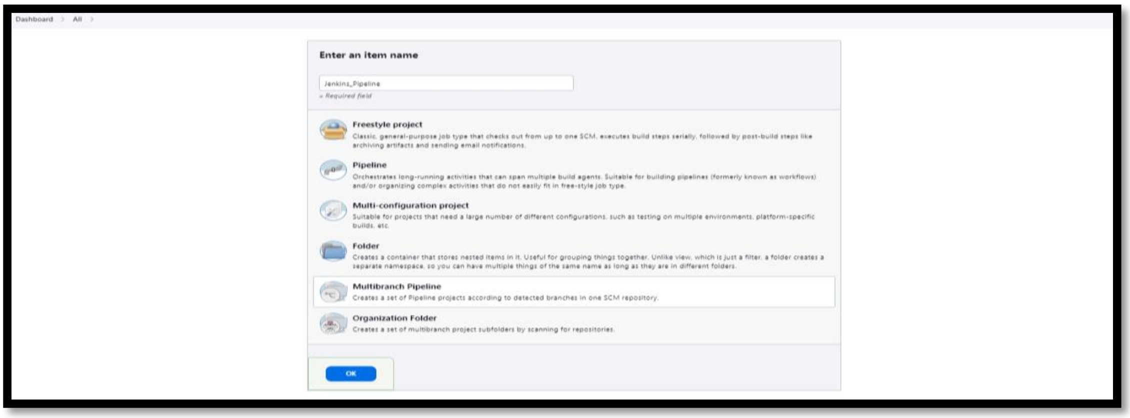
● Enter directly

Key: Enter New Secret Below

[illegible]

Passphrase:

Create Multibranch Pipeline



Enter an item name

Jenkins_Pipeline

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

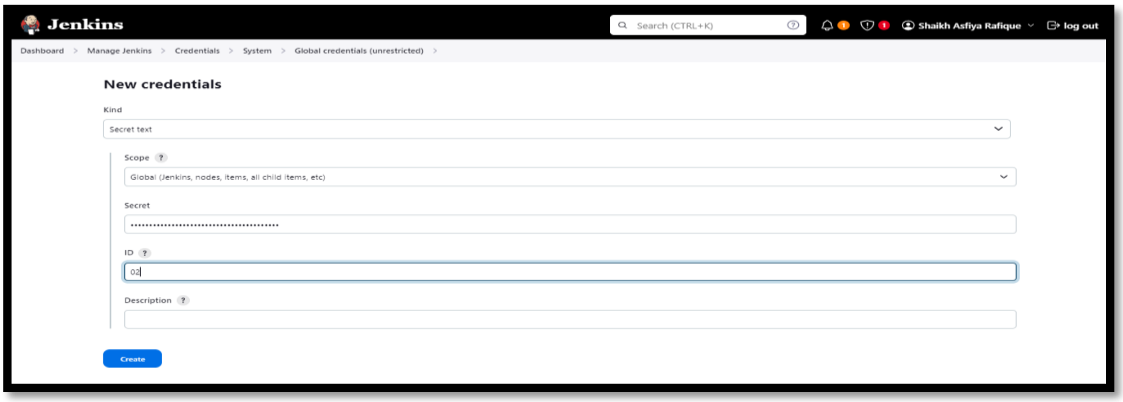
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

Create secret text to be used as your GitHub token password


Jenkins
Search (CTRL+K)
Shaikh Asfiya Rafique
log out
Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >
Create

New credentials

Kind: Secret text

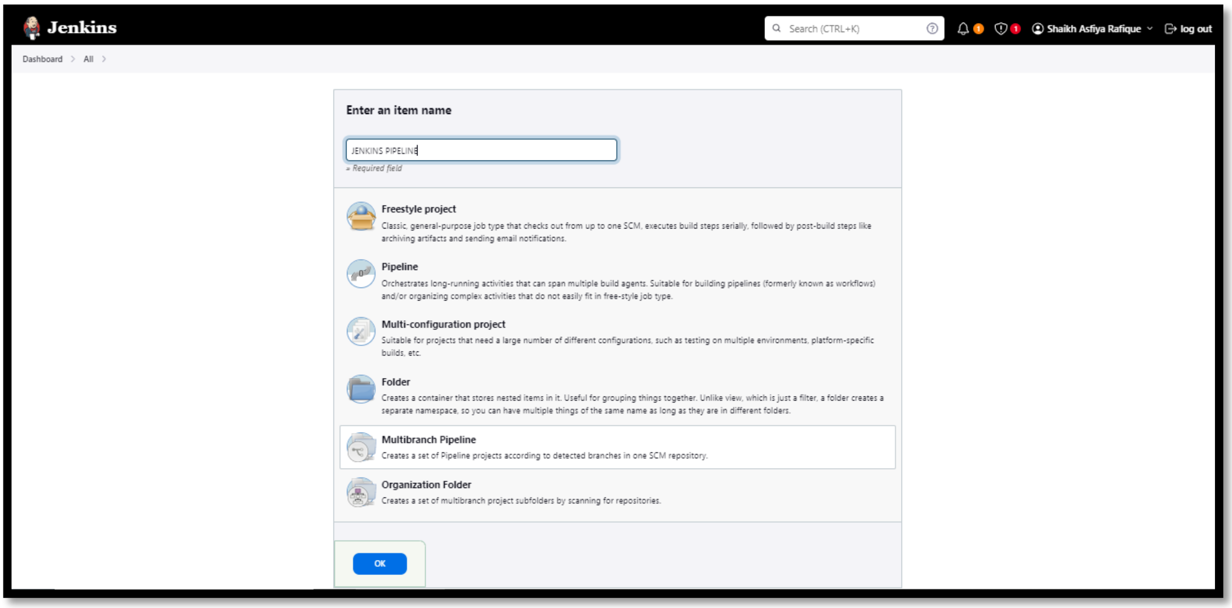
Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret: ••

ID: 02

Description:

Create Multi-branch Pipeline



Enter an item name

JENKINS PIPELINE

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

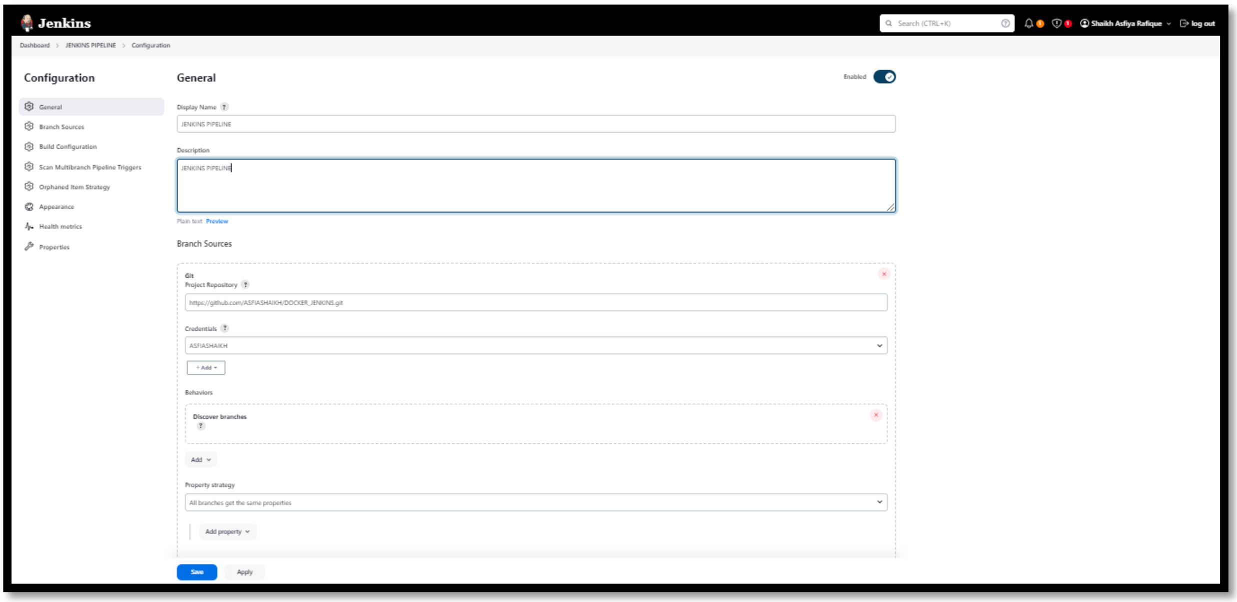
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK



General

Enabled

Display Name: JENKINS PIPELINE

Description: JENKINS PIPELINE

Plain text Preview

Branch Sources

Git

Project Repository: https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git

Credentials: ASFIASHAIKH

+ Add

Behaviors

Discover branches

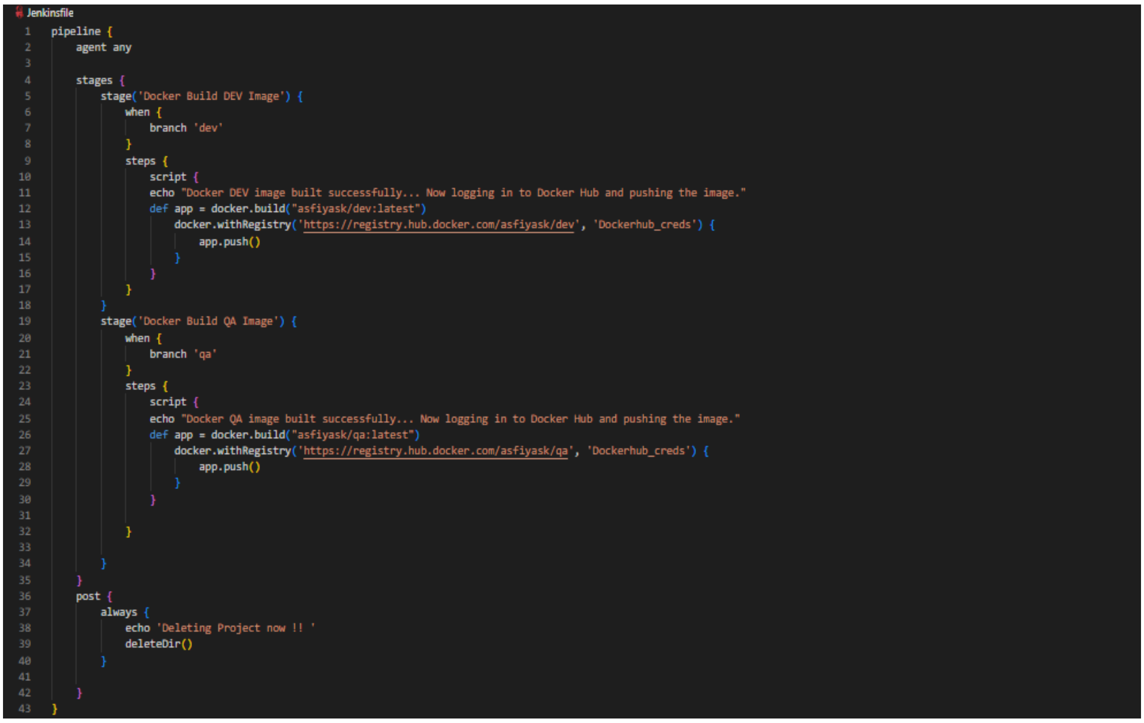
Add

Property strategy: All branches get the same properties

Add property

Save Apply

root@DESKTOP-8OOG2HF:DOCKER_JENKINS# cat Jenkinsfile

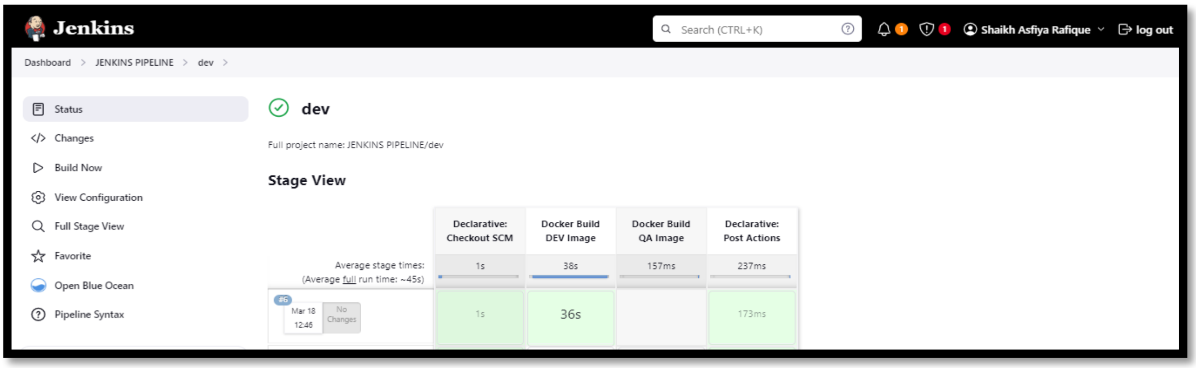


```
pipeline {
    agent any

    stages {
        stage('Docker Build DEV Image') {
            when {
                branch 'dev'
            }
            steps {
                script {
                echo "Docker DEV image built successfully... Now logging in to Docker Hub and pushing the image."
                def app = docker.build("asfiyask/dev:latest")
                    docker.withRegistry('https://registry.hub.docker.com/asfiyask/dev', 'Dockerhub_creds') {
                        app.push()
                    }
                }
            }
        }
        stage('Docker Build QA Image') {
            when {
                branch 'qa'
            }
            steps {
                script {
                echo "Docker QA image built successfully... Now logging in to Docker Hub and pushing the image."
                def app = docker.build("asfiyask/qa:latest")
                    docker.withRegistry('https://registry.hub.docker.com/asfiyask/qa', 'Dockerhub_creds') {
                        app.push()
                    }
                }

            }

        }
    }
    post {
        always {
            echo 'Deleting Project now !! '
            deleteDir()
        }

    }
}
```

Restart The Docker Container

```
root@DESKTOP-8OOG2HF:DOCKER_JENKINS# docker run \
  --name jenkins-docker \
  --rm \
  --detach \
  --privileged \
  --network jenkins \
  --network-alias docker \
  --env DOCKER_TLS_CERTDIR=/certs \
  --volume jenkins-docker-certs:/certs/client \
  --volume jenkins-data:/var/jenkins_home \
  --publish 2376:2376 \
  docker:dind \
  --storage-driver overlay2
d7978cc2a10087fba8e2a853eb3a654c88b98f7924055f47f04203554631f091
```

Now Build DEV Image



dev

Full project name: JENKINS PIPELINE/dev

Stage View

	Declarative: Checkout SCM	Docker Build DEV Image	Docker Build QA Image	Declarative: Post Actions
Average stage times: (Average full run time: ~45s)	1s	38s	157ms	237ms
#6 Mar 18 12:46 No Changes	1s	36s		173ms

Console Output

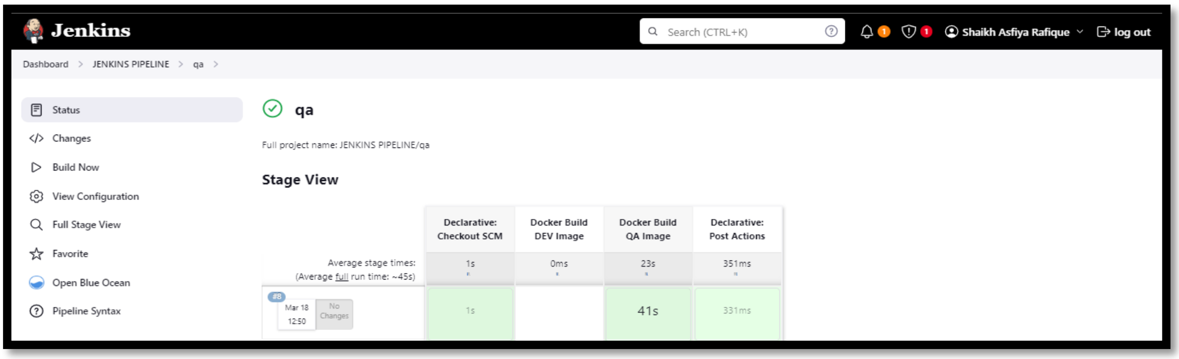
```
Started by user Shaikh Asfiya Rafique
 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-95a0dff2ac1165826c8419c1c9431c10/.git # timeout=10
Setting origin to https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
 > git config remote.origin.url https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
 > git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/dev
Seen branch in repository origin/main
Seen branch in repository origin/qa
Seen 3 remote branches
Obtained Jenkinsfile from de126bc016830bafb0ae4e572c917b0147fface8
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/JENKINS_PIPELINE_dev
[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 01
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
 > git init /var/jenkins_home/workspace/JENKINS_PIPELINE_dev # timeout=10
Fetching upstream changes from https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials
Verifying host key using known hosts file
```

```
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please
go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key
verification.
 > git fetch --no-tags --force --progress -- https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision de126bc016830bafb0ae4e572c917b0147fface8 (dev)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f de126bc016830bafb0ae4e572c917b0147fface8 # timeout=10
Commit message: "Files have been added"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Docker Build DEV Image)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
Docker DEV image built successfully... Now logging in to Docker Hub and pushing the image.
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t asfiyask/dev:latest .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile:
#1 transferring dockerfile: 132B 0.0s done
#1 DONE 0.2s

#2 [internal] load metadata for docker.io/library/nginx:alpine
#2 DONE 2.8s

#3 [internal] load .dockerignore
#3 transferring context: 2B done
#3 DONE 0.1s

#4 [internal] load build context
#4 transferring context: 469B done
#4 DONE 0.1s

#5 [1/3] FROM
docker.io/library/nginx:alpine@sha256:02d8d94023878cedf3e3acc55372932a9ba1478b6e2f3357786d916c2af7
43ba
#5 resolve
docker.io/library/nginx:alpine@sha256:02d8d94023878cedf3e3acc55372932a9ba1478b6e2f3357786d916c2af7
43ba 0.1s done
#5 sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8 0B / 3.40MB 0.1s
#5 sha256:e289a478ace02cd72f0a71a5b2ec0594495e1fae85faa10aae3b0da530812608 11.74kB / 11.74kB done
#5 sha256:ed3e62e73b33c9cfa4b253060771e4a9eebb751ab438052f197e847b4553a9ac 0B / 1.90MB 0.1s
#5 sha256:5126dce06df729f9a22956013e160f8b581d47095beec332d647a5c1119b2411 0B / 629B 0.1s
#5 sha256:02d8d94023878cedf3e3acc55372932a9ba1478b6e2f3357786d916c2af743ba 8.71kB / 8.71kB done
#5 sha256:cb3218c8a053881bd00f4fa93e9f87e2c6204761e391b3aafc942f104362e538 2.50kB / 2.50kB done
#5 sha256:5126dce06df729f9a22956013e160f8b581d47095beec332d647a5c1119b2411 629B / 629B 0.3s done
#5 sha256:1d0dd2dc2265a581798226f7c79d134ac797f42db3f934dd4af1d38a6b89ce5c 0B / 957B 0.6s
```

```
#5 sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8 1.05MB / 3.40MB 0.7s
#5 sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8 3.40MB / 3.40MB 0.9s
#5 sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8 3.40MB / 3.40MB 0.9s done
#5 sha256:1d0dd2dc2265a581798226f7c79d134ac797f42db3f934dd4af1d38a6b89ce5c 957B / 957B 1.0s
#5 extracting sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8
#5 sha256:2b1ab92f023179da00446365a60daa60d72a1edeb697fb81811e086eba2e0170 0B / 392B 1.0s
#5 sha256:ed3e62e73b33c9cfa4b253060771e4a9eebb751ab438052f197e847b4553a9ac 1.05MB / 1.90MB 1.1s
#5 sha256:1d0dd2dc2265a581798226f7c79d134ac797f42db3f934dd4af1d38a6b89ce5c 957B / 957B 1.0s done
#5 sha256:6eba808ac059320c42179a6590b021f8695d3f12c2afa8745e219f635acf19d4 0B / 1.21kB 1.1s
#5 sha256:ed3e62e73b33c9cfa4b253060771e4a9eebb751ab438052f197e847b4553a9ac 1.90MB / 1.90MB 1.2s done
#5 sha256:2b1ab92f023179da00446365a60daa60d72a1edeb697fb81811e086eba2e0170 392B / 392B 1.3s done
#5 sha256:6eba808ac059320c42179a6590b021f8695d3f12c2afa8745e219f635acf19d4 1.21kB / 1.21kB 1.3s
#5 sha256:57038e85fbb88e96e34a84b125e568f540437561adb363fa791ff9e94e153dc1 0B / 1.40kB 1.3s
#5 extracting sha256:619be1103602d98e1963557998c954c892b3872986c27365e9f651f5bc27cab8 0.5s done
#5 sha256:6eba808ac059320c42179a6590b021f8695d3f12c2afa8745e219f635acf19d4 1.21kB / 1.21kB 1.4s done
#5 sha256:57038e85fbb88e96e34a84b125e568f540437561adb363fa791ff9e94e153dc1 1.40kB / 1.40kB 1.5s done
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 0B / 12.65MB 1.5s
#5 extracting sha256:ed3e62e73b33c9cfa4b253060771e4a9eebb751ab438052f197e847b4553a9ac
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 1.05MB / 12.65MB 1.8s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 2.10MB / 12.65MB 1.9s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 4.19MB / 12.65MB 2.1s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 5.24MB / 12.65MB 2.3s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 6.73MB / 12.65MB 2.5s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 9.44MB / 12.65MB 2.7s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 10.49MB / 12.65MB 2.9s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 11.65MB / 12.65MB 3.0s
#5 sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 12.65MB / 12.65MB 3.1s done
#5 extracting sha256:ed3e62e73b33c9cfa4b253060771e4a9eebb751ab438052f197e847b4553a9ac 1.4s done
#5 extracting sha256:5126dce06df729f9a22956013e160f8b581d47095beec332d647a5c1119b2411
#5 extracting sha256:5126dce06df729f9a22956013e160f8b581d47095beec332d647a5c1119b2411 done
#5 extracting sha256:1d0dd2dc2265a581798226f7c79d134ac797f42db3f934dd4af1d38a6b89ce5c done
#5 extracting sha256:2b1ab92f023179da00446365a60daa60d72a1edeb697fb81811e086eba2e0170
#5 extracting sha256:2b1ab92f023179da00446365a60daa60d72a1edeb697fb81811e086eba2e0170 done
#5 extracting sha256:6eba808ac059320c42179a6590b021f8695d3f12c2afa8745e219f635acf19d4 done
#5 extracting sha256:57038e85fbb88e96e34a84b125e568f540437561adb363fa791ff9e94e153dc1
#5 extracting sha256:57038e85fbb88e96e34a84b125e568f540437561adb363fa791ff9e94e153dc1 done
#5 extracting sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7
#5 extracting sha256:eec94c9845c062c9f3495ce861d2bd9507bd7a13710de6b89a195a6b089ea8a7 1.2s done
#5 DONE 6.1s

#6 [2/3] COPY default.conf /etc/nginx/conf.d/
#6 DONE 0.2s

#7 [3/3] COPY index.html /usr/share/nginx/html/
#7 DONE 0.1s

#8 exporting to image
```

```
#8 exporting layers 0.1s done
#8 writing image sha256:53ce06d4cf9be36daeab180cf2128cb593116824e60b20953bb3e43976087f39 done
#8 naming to docker.io/asfiyask/dev:latest done
#8 DONE 0.1s
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u asfiyask -p ******** https://registry.hub.docker.com/asfiyask/dev
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in
/var/jenkins_home/workspace/JENKINS_PIPELINE_dev@tmp/ae837847-be3c-4404-88be-
966d77edd86e/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag asfiyask/dev:latest registry.hub.docker.com/asfiyask/dev:latest
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push registry.hub.docker.com/asfiyask/dev:latest
The push refers to repository [registry.hub.docker.com/asfiyask/dev]
4094e9efae23: Preparing
7de67799d542: Preparing
13c52683b537: Preparing
337b7d64083b: Preparing
cdd311f34c29: Preparing
3e8ad8bcb0ac: Preparing
74b4ff8dbbd1: Preparing
c018a48a857c: Preparing
0f73163669d4: Preparing
aedc3bda2944: Preparing
74b4ff8dbbd1: Waiting
0f73163669d4: Waiting
c018a48a857c: Waiting
aedc3bda2944: Waiting
3e8ad8bcb0ac: Waiting
4094e9efae23: Pushed
7de67799d542: Pushed
cdd311f34c29: Pushed
3e8ad8bcb0ac: Pushed
74b4ff8dbbd1: Pushed
337b7d64083b: Pushed
c018a48a857c: Pushed
13c52683b537: Pushed
0f73163669d4: Pushed
aedc3bda2944: Pushed
latest: digest: sha256:01d16589afc0a87f8a3e54ffb7722bd09eb0ff1467d891e886b4c1d5c71d7020 size: 2403
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker Build QA Image)
Stage "Docker Build QA Image" skipped due to when conditional
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deleting Project now !!
[Pipeline] deleteDir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Now Build QA Image



qa

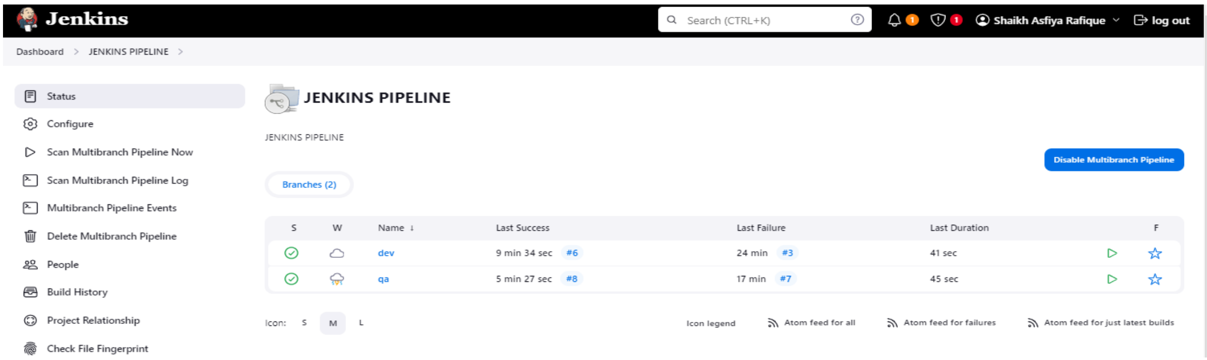
Full project name: JENKINS PIPELINE/qa

Stage View

	Declarative: Checkout SCM	Docker Build DEV Image	Docker Build QA Image	Declarative: Post Actions
Average stage times: (Average full run time: ~45s)	1s	0ms	23s	351ms
#8 Mar 18 12:50 No Changes	1s		41s	331ms

```
Started by user Shaikh Asfiya Rafique
 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-95a0dff2ac1165826c8419c1c9431c10/.git # timeout=10
Setting origin to https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
 > git config remote.origin.url https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
 > git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/dev
Seen branch in repository origin/main
Seen branch in repository origin/qa
Seen 3 remote branches
Obtained Jenkinsfile from 5178120baf8563da5b0bab1802d395ce76faccd1
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/JENKINS_PIPELINE_qa
[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 01
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
 > git init /var/jenkins_home/workspace/JENKINS_PIPELINE_qa # timeout=10
Fetching upstream changes from https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please
go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key
verification.
 > git fetch --no-tags --force --progress -- https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/ASFIASHAIKH/DOCKER_JENKINS.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 5178120baf8563da5b0bab1802d395ce76faccd1 (qa)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 5178120baf8563da5b0bab1802d395ce76faccd1 # timeout=10
Commit message: "Files have been added"
 > git rev-list --no-walk 5178120baf8563da5b0bab1802d395ce76faccd1 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Docker Build DEV Image)
Stage "Docker Build DEV Image" skipped due to when conditional
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker Build QA Image)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
Docker QA image built successfully... Now logging in to Docker Hub and pushing the image.
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t asfiyask/qa:latest .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile:
```

```
#1 transferring dockerfile: 132B done
#1 DONE 0.1s

#2 [internal] load metadata for docker.io/library/nginx:alpine
#2 DONE 1.4s

#3 [internal] load .dockerignore
#3 transferring context: 2B done
#3 DONE 0.1s

#4 [1/3] FROM
docker.io/library/nginx:alpine@sha256:02d8d94023878cedf3e3acc55372932a9ba1478b6e2f3357786d916c2af7
43ba
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 469B done
#5 DONE 0.1s

#6 [2/3] COPY default.conf /etc/nginx/conf.d/
#6 CACHED

#7 [3/3] COPY index.html /usr/share/nginx/html/
#7 CACHED

#8 exporting to image
#8 exporting layers done
#8 writing image sha256:53ce06d4cf9be36daeab180cf2128cb593116824e60b20953bb3e43976087f39 done
#8 naming to docker.io/asfiyask/qa:latest done
#8 DONE 0.0s
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u asfiyask -p ******** https://registry.hub.docker.com/asfiyask/qa
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in
/var/jenkins_home/workspace/JENKINS_PIPELINE_qa@tmp/4ccdf179-c289-4876-b458-
38692a4a2078/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag asfiyask/qa:latest registry.hub.docker.com/asfiyask/qa:latest
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push registry.hub.docker.com/asfiyask/qa:latest
The push refers to repository [registry.hub.docker.com/asfiyask/qa]
4094e9efae23: Preparing
7de67799d542: Preparing
13c52683b537: Preparing
337b7d64083b: Preparing
cdd311f34c29: Preparing
3e8ad8bcb0ac: Preparing
74b4ff8dbbd1: Preparing
c018a48a857c: Preparing
0f73163669d4: Preparing
aedc3bda2944: Preparing
74b4ff8dbbd1: Waiting
c018a48a857c: Waiting
0f73163669d4: Waiting
aedc3bda2944: Waiting
3e8ad8bcb0ac: Waiting
337b7d64083b: Mounted from asfiyask/dev
4094e9efae23: Mounted from asfiyask/dev
13c52683b537: Mounted from asfiyask/dev
7de67799d542: Mounted from asfiyask/dev
cdd311f34c29: Mounted from asfiyask/dev
0f73163669d4: Mounted from asfiyask/dev
c018a48a857c: Mounted from asfiyask/dev
3e8ad8bcb0ac: Mounted from asfiyask/dev
74b4ff8dbbd1: Mounted from asfiyask/dev
aedc3bda2944: Mounted from asfiyask/dev
latest: digest: sha256:01d16589afc0a87f8a3e54ffb7722bd09eb0ff1467d891e886b4c1d5c71d7020 size: 2403
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deleting Project now !!
[Pipeline] deleteDir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

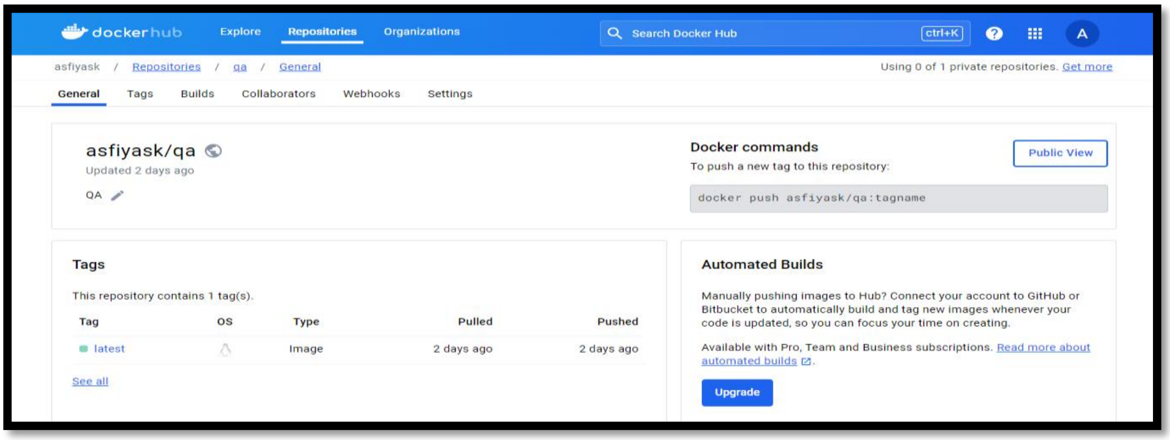

Jenkins
Search (CTRL+K)
Shaikh Asfiya Rafique
log out
Dashboard > JENKINS PIPELINE >
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
People
Build History
Project Relationship
Check File Fingerprint
Disable Multibranch Pipeline
Branches (2)
Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

JENKINS PIPELINE

JENKINS PIPELINE

S	W	Name ↓	Last Success	Last Failure	Last Duration		F
✓	☁	dev	9 min 34 sec #6	24 min #3	41 sec	▷	☆
✓	☂	qa	5 min 27 sec #8	17 min #7	45 sec	▷	☆

OUTPUT



Using 0 of 1 private repositories. Get more

asfiyask/qa

Updated 2 days ago

QA

Docker commands

To push a new tag to this repository:

```
docker push asfiyask/qa:tagname
```

Tags

This repository contains 1 tag(s).

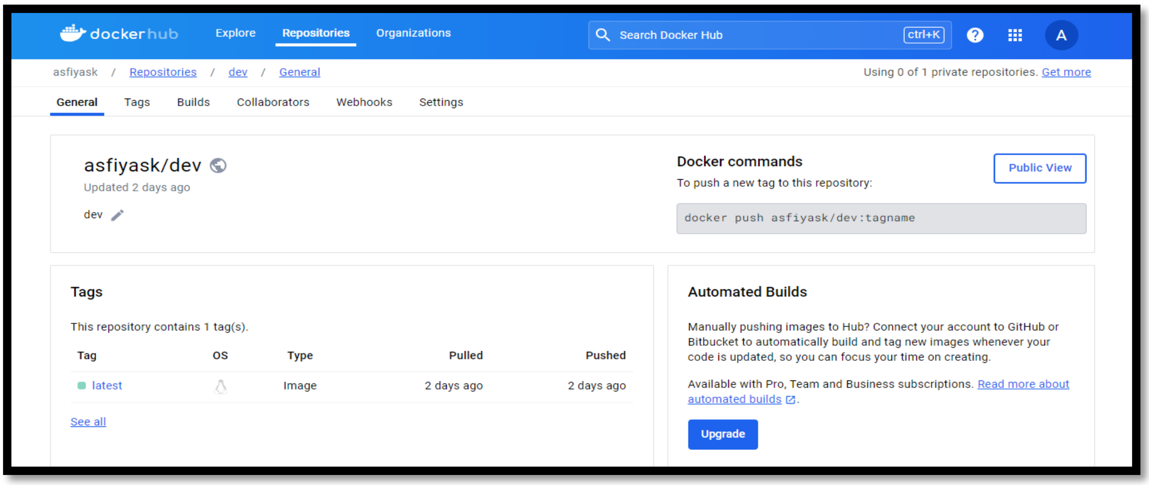
Tag	OS	Type	Pulled	Pushed
🟩 latest		Image	2 days ago	2 days ago

See all

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.



Using 0 of 1 private repositories. Get more

asfiyask/dev

Updated 2 days ago

dev

Docker commands

To push a new tag to this repository:

```
docker push asfiyask/dev:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
🟩 latest		Image	2 days ago	2 days ago

See all

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.

ROUGH WORK:

- **Docker Build and Publish**

Git Project -------------------→ DOCKER_JENKINS **-----------------------------→ DONE**

Branch Name ----------------→ dev OR qa **---→ DONE**

Dockerfile --→ DONE

Jenkins Plugin --------------→ CloudBees Docker Build and Publish ------------→ **DONE**

Docker Hub --→ DONE

Username ------------------------→ asfiyask

Registry / Repository -----------→ asfiyask/qa OR asfiyask/dev

DH Credentials --→ DONE

docker login –u asfiyask

dckr_pat_YMiADcIFx71ODQx8fAIp51oP7RRU

Create Jenkins Credes for Docker Hub ---→ DONE

Dockerhub_creds --→ DONE