Final Exam - Jenkins Outputs

Name: Hanil Zarbailov

Cloning Yaniv’s repo and checking out to the relevant branch:

```
ubuntu@ip-172-31-4-198:~$ git clone https://github.com/yanivomc/devopshift-welcome.git
Cloning into 'devopshift-welcome'...
remote: Enumerating objects: 18127, done.
remote: Total 18127 (delta 0), reused 0 (delta 0), pack-reused 18127 (from 1)
Receiving objects: 100% (18127/18127), 138.08 MiB | 27.09 MiB/s, done.
Resolving deltas: 100% (8808/8808), done.
ubuntu@ip-172-31-4-198:~$ cd devopshift-welcome/
ubuntu@ip-172-31-4-198:~/devopshift-welcome$ git checkout jenkins-workshop
Branch 'jenkins-workshop' set up to track remote branch 'jenkins-workshop' from 'origin'.
Switched to a new branch 'jenkins-workshop'
```

Docker-compose up to get Jenkins:

```
ubuntu@ip-172-31-4-198:~/devopshift-welcome$ cd welcome/
ubuntu@ip-172-31-4-198:~/devopshift-welcome/welcome$ docker-compose up -d
[+] Running 2/2
 ! jenkins-slave Warning pull access denied for jenkins-jb/slave, repository does not exist or
may require 'doc...                    0.1s
 ! jenkins Warning      pull access denied for jenkins-jb/master, repository does not exist or
may require 'docker l...                0.1s
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 64.3s (22/22) FINISHED
docker:default
 => [jenkins internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 945B
0.0s
 => [jenkins internal] load metadata for docker.io/jenkins/jenkins:2.452.3
0.4s
 => [jenkins internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [jenkins internal] load build context
0.0s
```

```
 => => transferring context: 596B
0.0s
 => [jenkins 1/4] FROM
docker.io/jenkins/jenkins:2.452.3@sha256:50b22a852fa690a195453231f649ee171ee6b15a
57c3b49bab4130dd97612c34         11.3s
 => => resolve
docker.io/jenkins/jenkins:2.452.3@sha256:50b22a852fa690a195453231f649ee171ee6b15a
57c3b49bab4130dd97612c34                0.0s
 => =>
sha256:7fef0d6ded53ed2c0bb64b8e07763090d7ae8cb509b12c3eb5d4bef89a02a704
12.63kB / 12.63kB                                    0.0s
 => => sha256:941a647ff357b9178f9d3d8260ed25d844ff821a22d4dafed9d0af2b3c1ca0af
61.36MB / 61.36MB                                    0.9s
 => => sha256:f0f251032519904eab2d3a254804dcdba946388780fba4623cfb235f36cbb9dd
8.85MB / 8.85MB                                      0.5s
 => => sha256:9afe56375fb720d32e496bb88a8fc7e93f019fd8461fdd693b6ae0c17a8a89a0
2.58kB / 2.58kB                                      0.0s
 => => sha256:e9aef93137af6e967e7242f3b3c8ecd8e6f571d1e6fdd9e72db0befeeae3cf13
49.55MB / 49.55MB                                    0.7s
 => =>
sha256:50b22a852fa690a195453231f649ee171ee6b15a57c3b49bab4130dd97612c34
3.12kB / 3.12kB                                      0.0s
 => =>
sha256:d4c6328162883aab3027931f99aa96b07a981739d217a68f4e4130b0242ded76
1.24kB / 1.24kB                                      0.6s
 => => sha256:995009f196ec387953e3a6f67b44dfcf7365e0074eace3faf36a9254630aea55
93.21MB / 93.21MB                                    2.3s
 => =>
sha256:07b0d8ca8ab78e59d86d891327b173de35ca5b98f150244d7d0ade086a330bcd
184B / 184B                                          0.7s
 => => extracting
sha256:e9aef93137af6e967e7242f3b3c8ecd8e6f571d1e6fdd9e72db0befeeae3cf13
4.1s
 => => sha256:25f6587a42fbde5287f33d61704a652e354efb6f355f2eadd0e02d708df041ef
191B / 191B                                          1.0s
 => =>
sha256:aa636159e94bd1c2d5444e45776b94da48da06bd396265952180ae4999bc6feb
63.22MB / 63.22MB                                    2.0s
 => => sha256:a559a5c3d54950547150cc5235c054941ff6120844d132711cf53472271cfa61
6.24MB / 6.24MB                                      1.2s
 => =>
sha256:862717e1a01c3b1e214cad199a3e3cfd8771669fa846c5be4b7c8bd9dc41c1ca
1.92kB / 1.92kB                                      1.4s
 => => sha256:6778d605d2d135446cea0c67fa05a5a68a90e323ffff50be51bef57e94c5e890
1.23kB / 1.23kB                                      1.5s
 => =>
sha256:86a41c428de5b2a3904a423112954de2d86bff6f0d7f8714aad1a35454851430 391B
/ 391B                                       1.6s
```

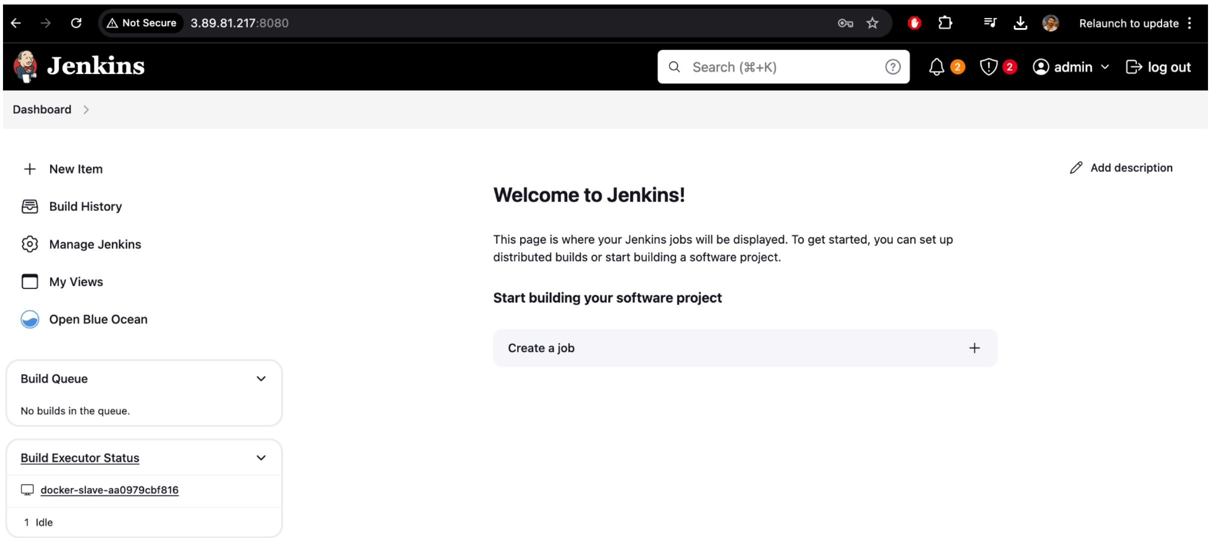
```
 => => extracting
sha256:941a647ff357b9178f9d3d8260ed25d844ff821a22d4dafed9d0af2b3c1ca0af
3.1s
 => => extracting
sha256:f0f251032519904eab2d3a254804dcdba946388780fba4623cfb235f36cbb9dd
0.2s
 => => extracting
sha256:d4c6328162883aab3027931f99aa96b07a981739d217a68f4e4130b0242ded76
0.0s
 => => extracting
sha256:07b0d8ca8ab78e59d86d891327b173de35ca5b98f150244d7d0ade086a330bcd
0.0s
 => => extracting
sha256:995009f196ec387953e3a6f67b44dfcf7365e0074eace3faf36a9254630aea55
0.9s
 => => extracting
sha256:25f6587a42fbde5287f33d61704a652e354efb6f355f2eadd0e02d708df041ef
0.0s
 => => extracting
sha256:a559a5c3d54950547150cc5235c054941ff6120844d132711cf53472271cfa61
0.1s
 => => extracting
sha256:aa636159e94bd1c2d5444e45776b94da48da06bd396265952180ae4999bc6feb
1.2s
 => => extracting
sha256:862717e1a01c3b1e214cad199a3e3cfd8771669fa846c5be4b7c8bd9dc41c1ca
0.0s
 => => extracting
sha256:6778d605d2d135446cea0c67fa05a5a68a90e323ffff50be51bef57e94c5e890
0.0s
 => => extracting
sha256:86a41c428de5b2a3904a423112954de2d86bff6f0d7f8714aad1a35454851430
0.0s
 => [jenkins 2/4] RUN /bin/jenkins-plugin-cli --plugins git matrix-auth credentials-binding
workflow-aggregator docker-workflow blueoce  15.1s
 => [jenkins 3/4] COPY executors.groovy /usr/share/jenkins/ref/init.groovy.d/
0.0s
 => [jenkins 4/4] COPY default-user.groovy /usr/share/jenkins/ref/init.groovy.d/
0.0s
 => [jenkins] exporting to image
0.8s
 => => exporting layers
0.8s
 => => writing image
sha256:1ea6ab0b034a04ff7a1bcf68c24b38f43697bb7b77c9d59c5483c4af7fd0263e
0.0s
 => => naming to docker.io/jenkins-jb/master:1.0
0.0s
```

```
 => [jenkins] resolving provenance for metadata file
0.0s
 => [jenkins-slave internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 773B
0.0s
 => [jenkins-slave internal] load metadata for docker.io/yanivomc/jenkins-slave:baseline-1.0
0.2s
 => [jenkins-slave internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [jenkins-slave 1/6] FROM
docker.io/yanivomc/jenkins-slave:baseline-1.0@sha256:a33eb6ae100e0e111ac90db4d1e5b
b737b7c49e6a3ba2a36c577d  31.1s
 => => resolve
docker.io/yanivomc/jenkins-slave:baseline-1.0@sha256:a33eb6ae100e0e111ac90db4d1e5b
b737b7c49e6a3ba2a36c577d76b0fa247a9      0.0s
 => =>
sha256:d19f32bd9e4106d487f1a703fc2f09c8edadd92db4405d477978e8e466ab290d
30.43MB / 30.43MB                                  0.5s
 => =>
sha256:32284c1b394eedb00472a01718ca8298646c143b7e6a2dd655fde353ecb42a32
26.81MB / 26.81MB                                  0.6s
 => =>
sha256:a33eb6ae100e0e111ac90db4d1e5bb737b7c49e6a3ba2a36c577d76b0fa247a9
2.22kB / 2.22kB                                  0.0s
 => => sha256:df3aee5eea3790fa2edb4bf0f103b91fc9d8eb5874510d4d60858ae05e110ac5
2.86kB / 2.86kB                                  0.0s
 => =>
sha256:53b0a1566aad76d8d1f613d3fae4c1321476571232374425 85b1c4141118570f
46.18MB / 46.18MB                                  1.7s
 => => extracting
sha256:d19f32bd9e4106d487f1a703fc2f09c8edadd92db4405d477978e8e466ab290d
4.4s
 => => sha256:3b5dc1bb247f6fffdcce87acdf0012c9b320bf117025c5860e15bbf909e7b9dd
3.03kB / 3.03kB                                  0.6s
 => =>
sha256:8863737a01ca625bc67faee64cc48c78a0212b6dc4a04f0300283787a37e8989
190.95kB / 190.95kB                                  0.7s
 => =>
sha256:c2755d3340c1e5dcd9645da114124a36432a1bc2f0cad00c58485a27ff371090
160.87MB / 160.87MB                                   3.7s
 => => sha256:a372301c893215850cd168686bdac8d2b96f5851ff1a0fd87e5ac85452fbf58a
419.75kB / 419.75kB                                  0.8s
 => =>
sha256:6ba9ed0ab9112b4a5edd5ad71c7a5715d3df905c357bfc96a72d48c29b86e341
1.33MB / 1.33MB                                  1.0s
```

```
 => => sha256:541ef992d7b4398bf1f281f4ba5a4a8739f7b7e0836dbbb9ec727783999020f2
444.48MB / 444.48MB                                7.7s
 => => extracting
sha256:32284c1b394eedb00472a01718ca8298646c143b7e6a2dd655fde353ecb42a32
1.2s
 => => extracting
sha256:53b0a1566aad76d8d1f613d3fae4c132147657123237442585b1c4141118570f
3.3s
 => => extracting
sha256:3b5dc1bb247f6fffdcce87acdf0012c9b320bf117025c5860e15bbf909e7b9dd
0.0s
 => => extracting
sha256:8863737a01ca625bc67faee64cc48c78a0212b6dc4a04f0300283787a37e8989
0.0s
 => => extracting
sha256:c2755d3340c1e5dcd9645da114124a36432a1bc2f0cad00c58485a27ff371090
6.0s
 => => extracting
sha256:a372301c893215850cd168686bdac8d2b96f5851ff1a0fd87e5ac85452fbf58a
0.0s
 => => extracting
sha256:6ba9ed0ab9112b4a5edd5ad71c7a5715d3df905c357bfc96a72d48c29b86e341
0.6s
 => => extracting
sha256:541ef992d7b4398bf1f281f4ba5a4a8739f7b7e0836dbbb9ec727783999020f2
12.8s
 => [jenkins-slave internal] load build context
0.0s
 => => transferring context: 3.20kB
0.0s
 => [jenkins-slave 2/6] RUN pip install jenkins-webapi
4.1s
 => [jenkins-slave 3/6] RUN mkdir -p /home/jenkins
0.3s
 => [jenkins-slave 4/6] RUN mkdir -p /var/lib/jenkins
0.2s
 => [jenkins-slave 5/6] ADD slave.py /var/lib/jenkins/slave.py
0.1s
 => [jenkins-slave 6/6] WORKDIR /home/jenkins
0.0s
 => [jenkins-slave] exporting to image
0.1s
 => => exporting layers
0.1s
 => => writing image
sha256:b36294a155d4d8e580929709ef98fd9434da9142615ee38321f68ca4ad318fe9
0.0s
```

```
 => => naming to docker.io/jenkins-jb/slave:1.0
0.0s
 => [jenkins-slave] resolving provenance for metadata file
0.0s
[+] Running 5/5
 ✔ jenkins                      Built
0.0s
 ✔ jenkins-slave                Built
0.0s
 ✔ Network welcome_jb           Created
0.1s
 ✔ Container jenkins            Started
0.8s
 ✔ Container welcome-jenkins-slave-1  Started
1.2s
```

Accessed Jenkins via instance public ip and port 8080:



No builds in the queue.

docker-slave-aa0979cbf816

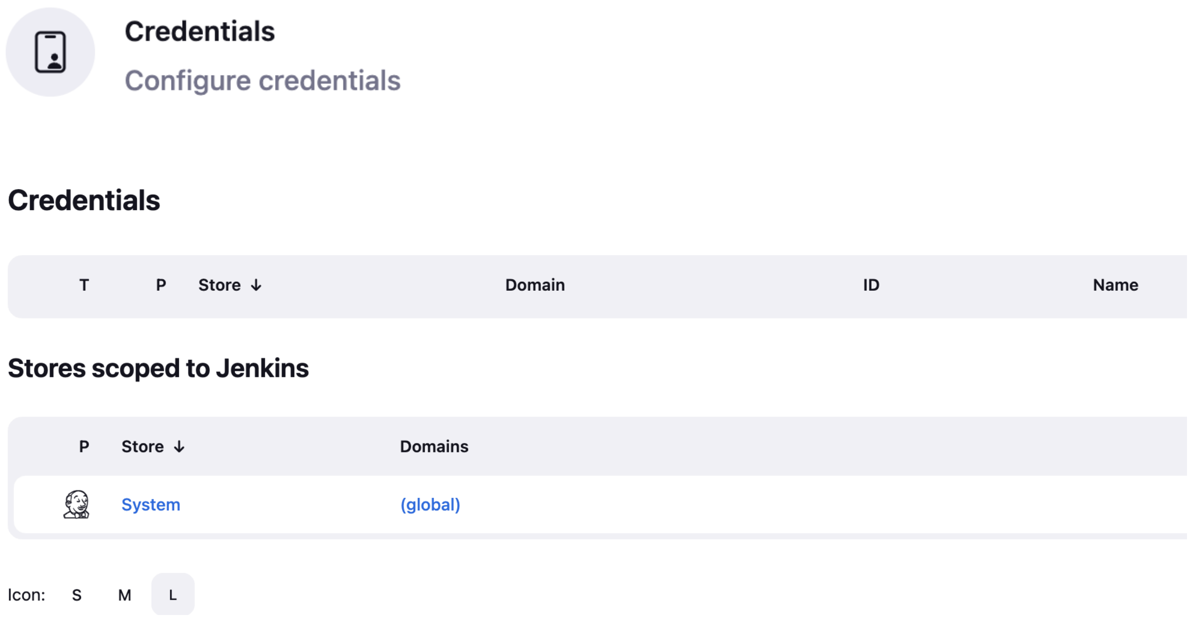
1 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Configuring Dockerhub Credentials:



Credentials

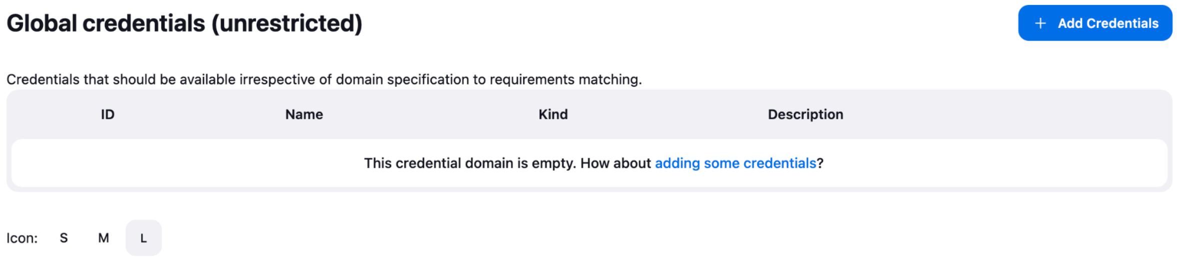
Configure credentials

Credentials

T	P	Store ↓	Domain	ID	Name

Stores scoped to Jenkins

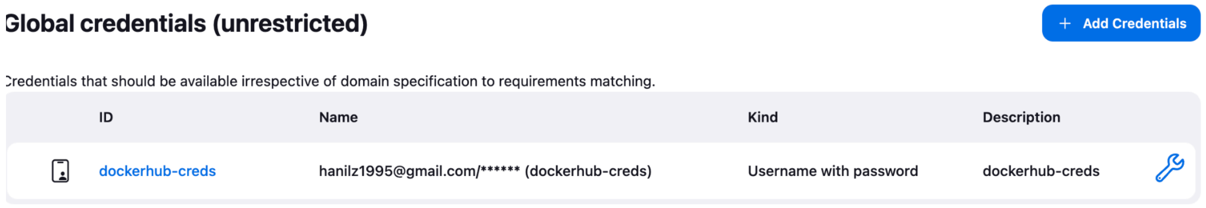
P	Store ↓	Domains
	System	(global)



Global credentials (unrestricted)

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

ID	Name	Kind	Description
This credential domain is empty. How about adding some credentials?			



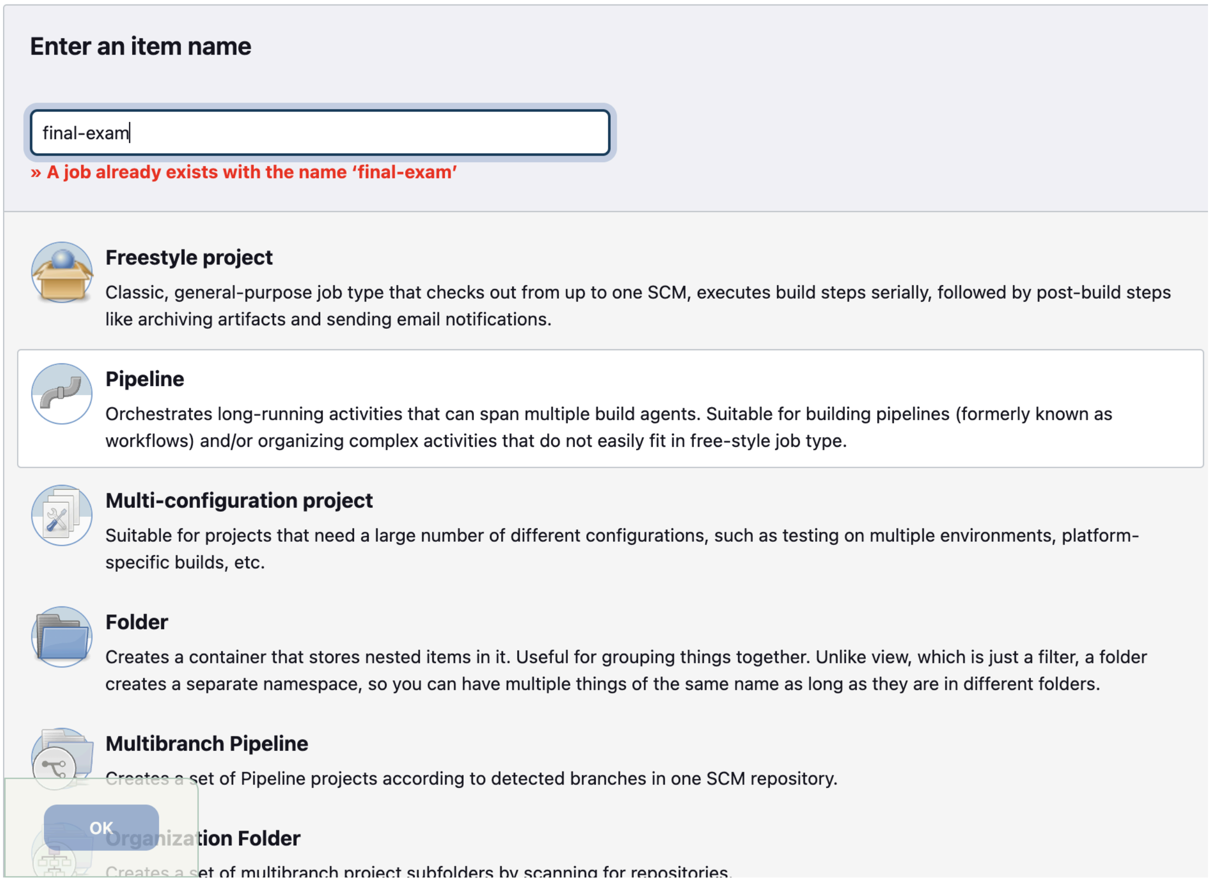
Global credentials (unrestricted)

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

ID	Name	Kind	Description
dockerhub-creds	hanilz1995@gmail.com/****** (dockerhub-creds)	Username with password	dockerhub-creds

NOTE: At this stage I’ve created a repository on Dockerhub but forgot to screenshot the steps for it.

Creating the Pipeline:



Enter an item name

final-exam

» A job already exists with the name 'final-exam'

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.

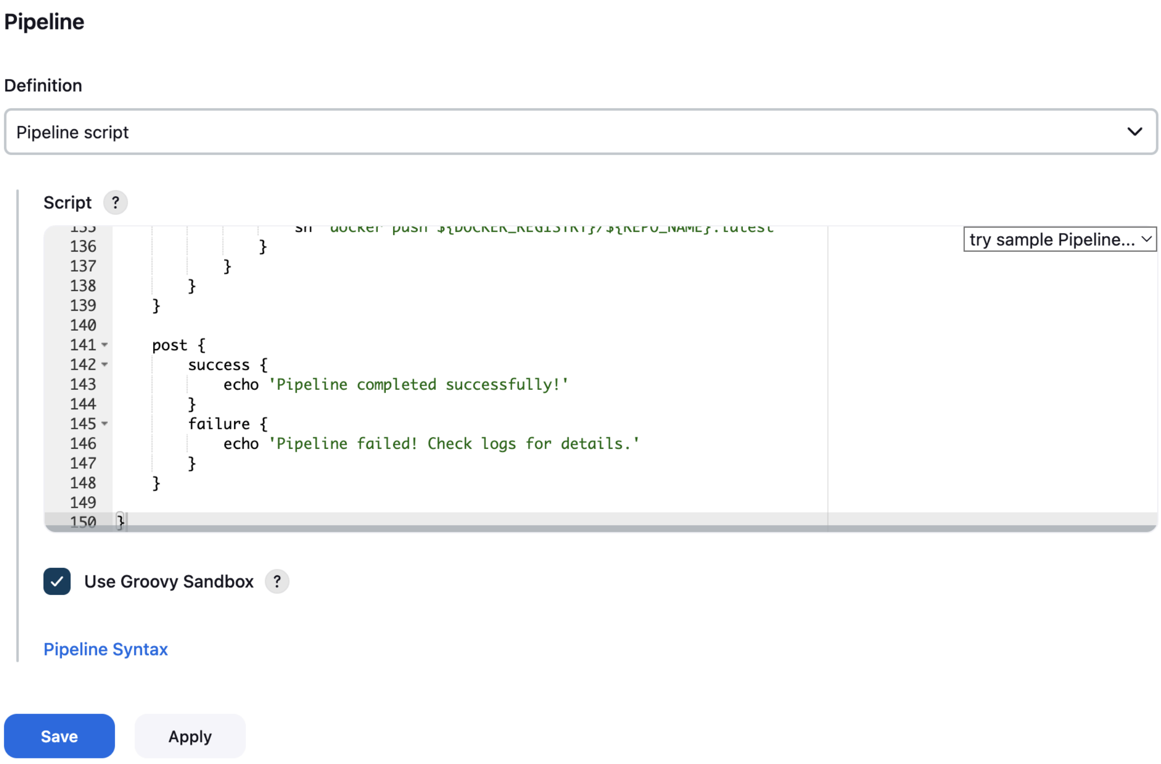
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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.



Pipeline

Definition

Pipeline script

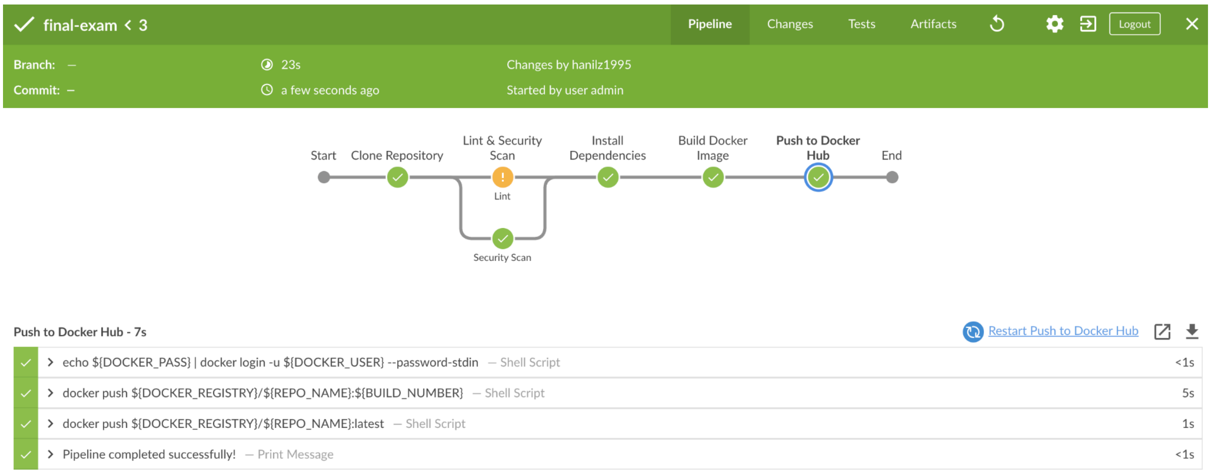
Script ?

```
                    sh 'docker push ${DOCKER_REGISTRY}/${REPO_NAME}:latest
                }
            }
        }
    }

    post {
        success {
            echo 'Pipeline completed successfully!'
        }
        failure {
            echo 'Pipeline failed! Check logs for details.'
        }
    }
}
```

☑ Use Groovy Sandbox ?

Pipeline Syntax

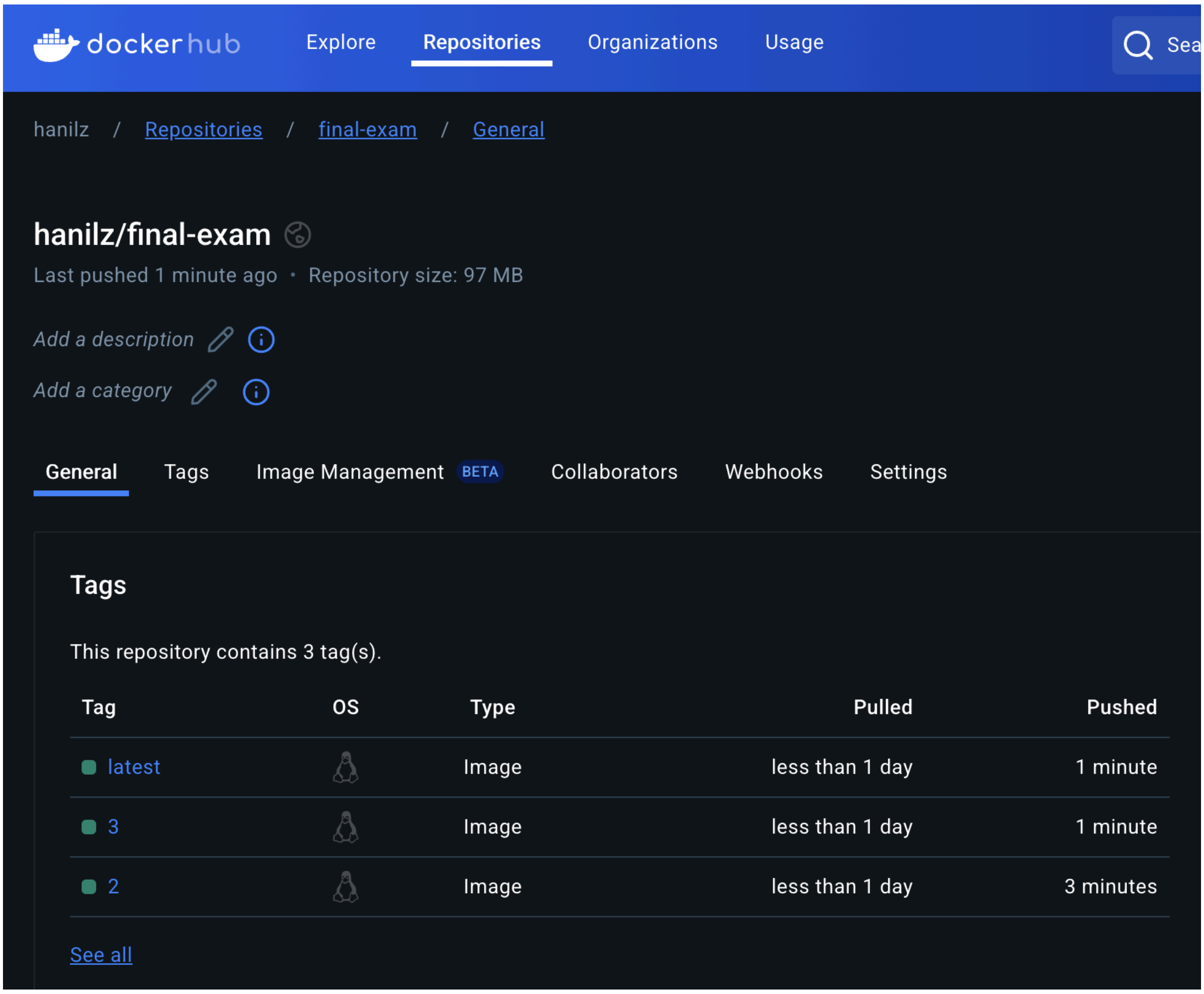


Branch: –
Commit: –
23s
a few seconds ago
Changes by hanilz1995
Started by user admin

Push to Docker Hub - 7s

✓	echo ${DOCKER_PASS} | docker login -u ${DOCKER_USER} --password-stdin — Shell Script	<1s
✓	docker push ${DOCKER_REGISTRY}/${REPO_NAME}:${BUILD_NUMBER} — Shell Script	5s
✓	docker push ${DOCKER_REGISTRY}/${REPO_NAME}:latest — Shell Script	1s
✓	Pipeline completed successfully! — Print Message	<1s

Checking If the Images Were Pushed Correctly To the Dockerhub Repository:



hanilz / Repositories / final-exam / General

hanilz/final-exam

Last pushed 1 minute ago • Repository size: 97 MB

Add a description

Add a category

Tags

This repository contains 3 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	1 minute
3		Image	less than 1 day	1 minute
2		Image	less than 1 day	3 minutes

See all

Great Success!

You can access the repository using the following link:
https://hub.docker.com/repository/docker/hanilz/final-exam/general