**DevOps**

**Day 2**

**Activity 3: Containers with Docker**

Table of Contents

[Table of Contents 2](#_Toc472717081)

[Activity: Container with Dockers 3](#_Toc472717082)

[Sign in Docker public Registry 4](#_Toc472717083)

[Add Docker Credentials to Jenkin 4](#_Toc472717084)

[Trigger Docker Pipeline 5](#_Toc472717085)

[Explore the Pipeline 6](#_Toc472717086)

[View Image at Docker Registry 7](#_Toc472717087)

[Troubleshooting 7](#_Toc472717088)

Activity: Container with Dockers

The list of steps to be performed in this activity is given below:

|  | **Task Description** | **Target Date for Completion** | **Supporting Documents/Personnel** |
| --- | --- | --- | --- |
|  | Sign in Docker public Registry | Day 2 | Participants |
|  | Add Docker Credentials | Day 2 | Participants |
|  | Trigger Docker Pipeline | Day 2 | Participants |
|  | Explore the pipeline | Day 2 | Participants |
|  | View Image at Docker Registry | Day 2 | Participants |

**Activity Duration:** 30 minutes

**Activity Description:**

Activity Type: Individual

* Purpose: In this module, the activity aims in pushing your built Docker images to your public Docker registry and pull it again in container by triggering Docker pipeline execution.
* You will need to provide Jenkins with the credentials of your Docker Hub account

.

**Technical Overview:** In this activity, you will trigger Docker pipeline execution to push the image to the Docker hub registry.

**Activity Inputs/Templates:** NA

**Activity Instructions:**

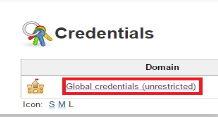
Sign in Docker public Registry

1. You are required to sign up to Docker Hub at <https://hub.docker.com/>
2. While signup, please provide Accenture mail id as an email address. (Confirmation link will be sent).
3. Remember Docker Id and password which you will be using for configuration.

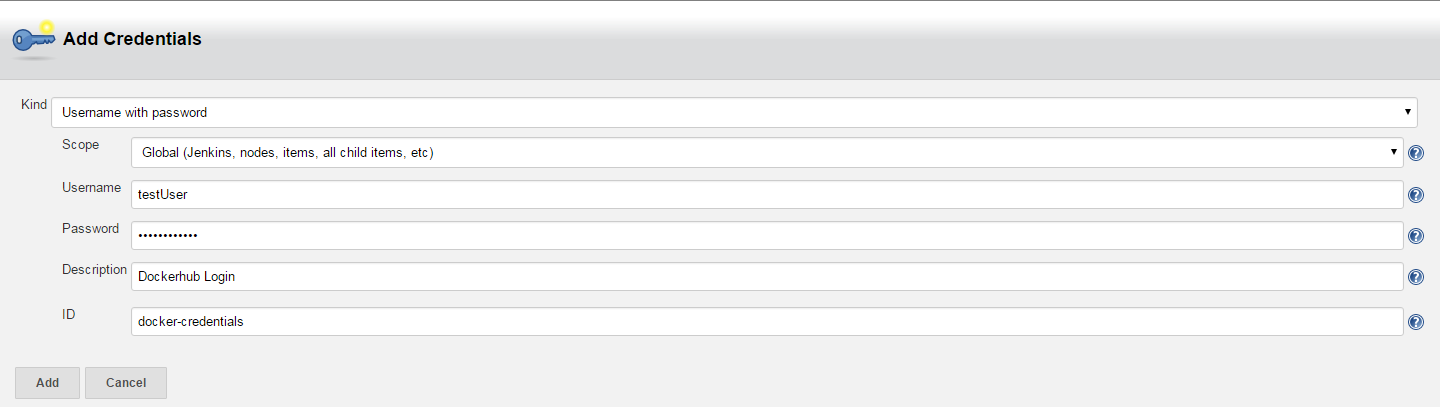
Add Docker Credentials to Jenkin

1. Open a browser and got to at http://<public IP>/jenkins
2. Click on "Credentials" link
3. Navigate to the Stores scoped to "Jenkins" as highlighted below and click on Jenkins
4. Then click on "Global Credentials" -> "Add Credentials "





1. Ensure that the "Kind" of credentials is selected as **"Username with password".**
2. Add your Docker username and password (Type Docker Id and password created during sign up in Docker registry)
3. The description is an optional field to help you identify your credentials easily
4. Add the ID as "**docker-credentials**" as shown in the below image.  *The Module 6 jobs are configured to look for that ID by default*

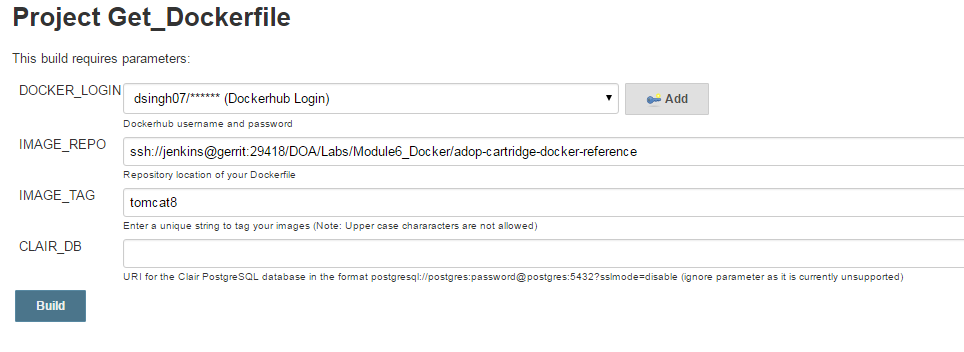


Trigger Docker Pipeline

1. Access the 'Module 6 - Docker CI' view on your Jenkins instance

http://<public-IP>/jenkins/job/DOA/job/Labs/job/Module6\_Docker/

1. Click on the "Get\_Dockerfile" job so that we can trigger the pipeline
2. Click on "Build with Parameters"
   1. For the DOCKER\_LOGIN parameter select your newly added credentials
   2. Leave the IMAGE\_REPO field to the default
   3. The IMAGE\_TAG can be changed to something more meaningful to you (e.g. name.surname\_tomcat8)
   4. Leave the CLAIR\_DB field as the default blank value



1. Once you have verified all the entered credentials are correct, click on "Build" to trigger the pipeline

Explore the Pipeline

1. Access the pipeline view at

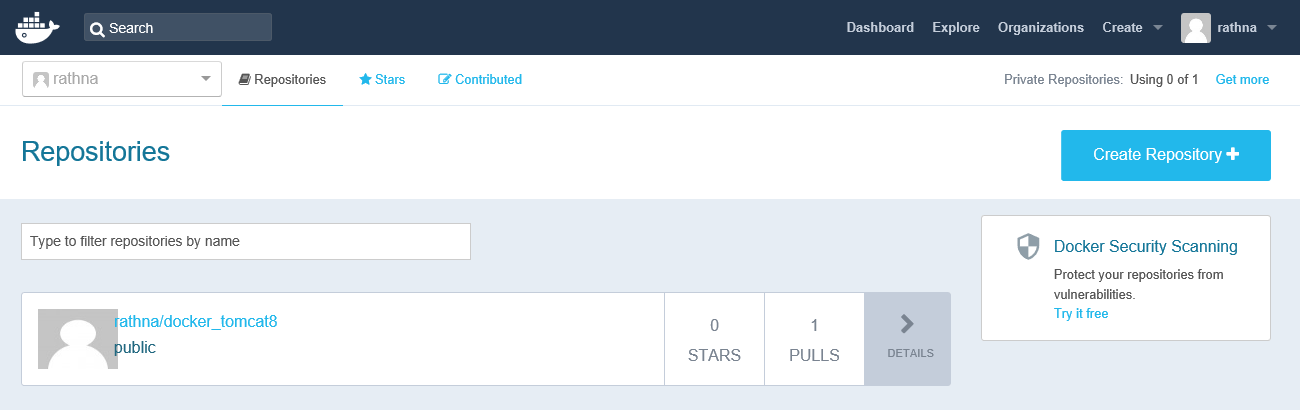
http://<PublicIP>/jenkins/job/DOA/job/Labs/job/Module6\_Docker/view/Sample\_Docker\_CI/

1. Following steps are executed during pipeline execution
   1. Static code analysis on the raw Dockerfile using an external tool called Dockerlint
   2. Image\_Build job will build the Tomcat image from scratch
   3. It's possible that this issue might occur in the Container\_Test job as well
   4. The Vulnerability\_Scan job will currently do nothing
   5. The Image\_Test and Container\_Test jobs will build a testing image and run some standard tests against it, defined in our Dockerfile repository
   6. Once we have passed all stages in the pipeline up to now, we will push our built and tested image to Docker Hub using the credentials provided in the Image\_Push job
   7. Finally, we pull the image we just pushed from Docker Hub and run it in a container in the Container\_Deploy job
   8. The Container\_Deploy job will pull the above pushed job and run it as a container
   9. The final job (optional) in the pipeline, Container\_Cleanup, is used to delete the containers we just deployed

View Image at Docker Registry

1. Once the Image\_Push job has completed, log onto Docker Hub using your credentials
2. You should be able to see a list of "Repositories" where you will see an image with a tag in the form: dockerhub\_username/tag\_that\_you\_provided.

***Note***: *Pushed Image will be reflected in Docker Hub Registry after 15-20 minutes.*



1. When you click on "Details", you will be able to see further information about the image.

Troubleshooting

1. If an “image build” job fails with the following message “Some index files failed to download”, then simply rebuild the pipeline by triggering the Get\_Dockerfile job.
2. To reuse the Docker Image from Docker hub registry, follow the below steps:
   1. Install Docker Toolbox
   2. Open Kitematic application
   3. Pull the newly created image using Kitematic application
   4. Open tomcat homepage using browser