Practice for Lesson 7: Jenkins Integration with Docker

Practices for Lesson 7

Overview

In these practices, you will learn how to setup the environment for Docker container job creation on Jenkins instance. Further, create the Pipeline Job for Docker on Jenkins instance using a sample example and then deploying the Docker Container with the Jenkins Pipeline using the GitHub repository

Practice 7-1: Setup Docker Plugins on Jenkins

Overview

In this practice, you will learn how to setup the environment for Docker container job creation on Jenkins instance.

Assumptions

You should have completed the Practice of Lesson 6.

Tasks

1. Sign in to **Jenkins Instance Dashboard**.
   a. In a browser on your local machine, enter the **Public IP** address of the EC2 instance followed by the IP address to sign in to the Jenkins Dashboard (for example, **<Public-IP>:8080**).

   b. Enter the user name and password provided.
   c. You will have access to the Jenkins Dashboard.

2. Install Docker on the **Linux instance**.
   a. Connect to putty and execute the below shown command to clone the Git repository.
   URL: https://github.com/oralabuser/Pipeline-using-Docker.git

```
[ec2-user@ip-172-31-35-174 ~]$ git clone https://github.com/oralabuser/Pipeline-using-Docker.git
```

   b. Execute the maven package command as shown below.

```
[ec2-user@ip-172-31-35-174 ~]$ cd Pipeline-using-Docker/
[ec2-user@ip-172-31-35-174 Pipeline-using-Docker]$
[ec2-user@ip-172-31-35-174 Pipeline-using-Docker]$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building LoginWebApp Maven Webapp 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ LoginWebApp --
-
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e.
build is platform dependent!
[INFO] skip non existing resourceDirectory /home/ec2-user/Pipeline-using-Docker/src/
main/resources
[INFO]
```

c. The maven BUILDs the Git repository packages successfully as shown below.

```
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.4:war (default-war) @ LoginWebApp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [LoginWebApp] in [/home/ec2-user/Pipeline-using-Docker/targ
et/LoginWebApp-1]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ec2-user/Pipeline-using-Docker/src/main/webap
p]
[INFO] Webapp assembled in [91 msecs]
[INFO] Building war: /home/ec2-user/Pipeline-using-Docker/target/LoginWebApp-1.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.974s
[INFO] Finished at: Thu Jul 08 06:39:08 UTC 2021
[INFO] Final Memory: 7M/20M
[INFO] ------------------------------------------------------------------------
[ec2-user@ip-172-31-35-174 Pipeline-using-Docker]$ cd ..
[ec2-user@ip-172-31-35-174 ~]$
```

d. The Linux Instance local repositories should be updated, run the **yum update** command as shown below.

```
[ec2-user@ip-172-31-35-174 ~]$
[ec2-user@ip-172-31-35-174 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                          | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package amazon-ssm-agent.x86_64 0:3.0.529.0-1.amzn2 will be updated
---> Package amazon-ssm-agent.x86_64 0:3.0.1124.0-1.amzn2 will be an update
---> Package bind-export-libs.x86_64 32:9.11.4-26.P2.amzn2.5 will be updated
---> Package bind-export-libs.x86_64 32:9.11.4-26.P2.amzn2.5.2 will be an update
---> Package bind-libs.x86_64 32:9.11.4-26.P2.amzn2.5 will be updated
---> Package bind-libs.x86_64 32:9.11.4-26.P2.amzn2.5.2 will be an update
---> Package bind-libs-lite.x86_64 32:9.11.4-26.P2.amzn2.5 will be updated
---> Package bind-libs-lite.x86_64 32:9.11.4-26.P2.amzn2.5.2 will be an update
---> Package bind-license.noarch 32:9.11.4-26.P2.amzn2.5 will be updated
---> Package bind-license.noarch 32:9.11.4-26.P2.amzn2.5.2 will be an update
```

```
  nss-tools.x86_64 0:3.53.1-7.amzn2
  openldap.x86_64 0:2.4.44-23.amzn2.0.1
  perl-Git.noarch 0:2.32.0-1.amzn2.0.1
  python.x86_64 0:2.7.18-1.amzn2.0.4
  python-devel.x86_64 0:2.7.18-1.amzn2.0.4
  python-libs.x86_64 0:2.7.18-1.amzn2.0.4
  python-lxml.x86_64 0:3.2.1-4.amzn2.0.3
  python-urllib3.noarch 0:1.25.9-1.amzn2.0.1
  python3.x86_64 0:3.7.10-1.amzn2.0.1
  python3-libs.x86_64 0:3.7.10-1.amzn2.0.1
  python3-pip.noarch 0:20.2.2-1.amzn2.0.3
  systemd.x86_64 0:219-78.amzn2.0.14
  systemd-libs.x86_64 0:219-78.amzn2.0.14
  systemd-sysv.x86_64 0:219-78.amzn2.0.14
  systemtap-runtime.x86_64 0:4.4-1.amzn2.0.1
  tzdata.noarch 0:2021a-1.amzn2
  tzdata-java.noarch 0:2021a-1.amzn2
  update-motd.noarch 0:1.1.2-2.amzn2.0.2

Complete!
[ec2-user@ip-172-31-35-174 ~]$
```

e. Next, install the **Docker** in the Linux instance as shown below.

```
[ec2-user@ip-172-31-35-174 ~]$ sudo yum install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                        | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package docker.x86_64 0:20.10.4-1.amzn2 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.4-1.amzn2.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.4-1.amzn2.x86
_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.4-1.amzn2.x86_64
--> Processing Dependency: pigz for package: docker-20.10.4-1.amzn2.x86_64
--> Running transaction check
---> Package containerd.x86_64 0:1.4.4-1.amzn2 will be installed
---> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
---> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
---> Package runc.x86_64 0:1.0.0-0.3.20210225.gitl2644e6.amzn2 will be installed
--> Finished Dependency Resolution
```

f. Type ‘Y’ to proceed the Docker installation process as shown below.

```
Transaction Summary
================================================================
Install  1 Package (+4 Dependent packages)

Total download size: 59 M
Installed size: 243 M
Is this ok [y/d/N]: y
```

```
Transaction test succeeded
Running transaction
  Installing : runc-1.0.0-0.3.20210225.gitl2644e6.amzn2.x86_64                  1/5
  Installing : containerd-1.4.4-1.amzn2.x86_64                                  2/5
  Installing : libcgroup-0.41-21.amzn2.x86_64                                   3/5
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64                                    4/5
  Installing : docker-20.10.4-1.amzn2.x86_64                                    5/5
  Verifying  : containerd-1.4.4-1.amzn2.x86_64                                  1/5
  Verifying  : docker-20.10.4-1.amzn2.x86_64                                    2/5
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64                                    3/5
  Verifying  : runc-1.0.0-0.3.20210225.gitl2644e6.amzn2.x86_64                  4/5
  Verifying  : libcgroup-0.41-21.amzn2.x86_64                                   5/5

Installed:
  docker.x86_64 0:20.10.4-1.amzn2

Dependency Installed:
  containerd.x86_64 0:1.4.4-1.amzn2     libcgroup.x86_64 0:0.41-21.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1       runc.x86_64 0:1.0.0-0.3.20210225.gitl2644e6.amzn2

Complete!
[ec2-user@ip-172-31-35-174 ~]$
```

g. Run the Docker command to start the Docker and check its status as shown below.

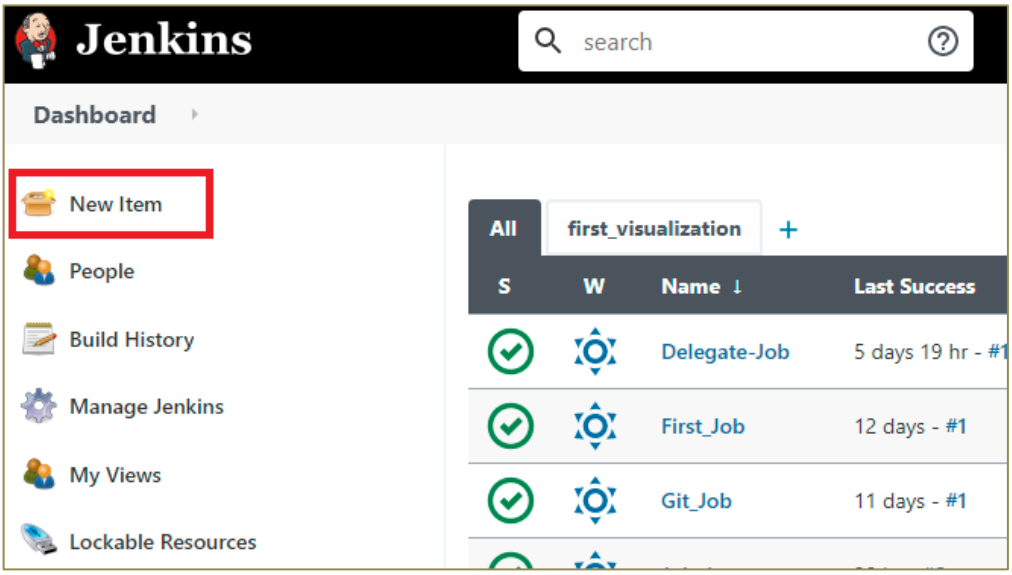
```
[ec2-user@ip-172-31-35-174 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-35-174 ~]$
[ec2-user@ip-172-31-35-174 ~]$ sudo service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-07-08 07:58:27 UTC; 11s ago
     Docs: https://docs.docker.com
  Process: 13764 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, stat
us=0/SUCCESS)
  Process: 13754 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 13771 (dockerd)
    Tasks: 7
   Memory: 49.4M
   CGroup: /system.slice/docker.service
           └─13771 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock ...

Jul 08 07:58:26 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:26 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:26 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:26 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:27 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:27 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:27 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:27 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Jul 08 07:58:27 ip-172-31-35-174.ap-south-1.compute.internal systemd[1]: Started Docker Ap...
Jul 08 07:58:27 ip-172-31-35-174.ap-south-1.compute.internal dockerd[13771]: time="2021-07...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-35-174 ~]$
```

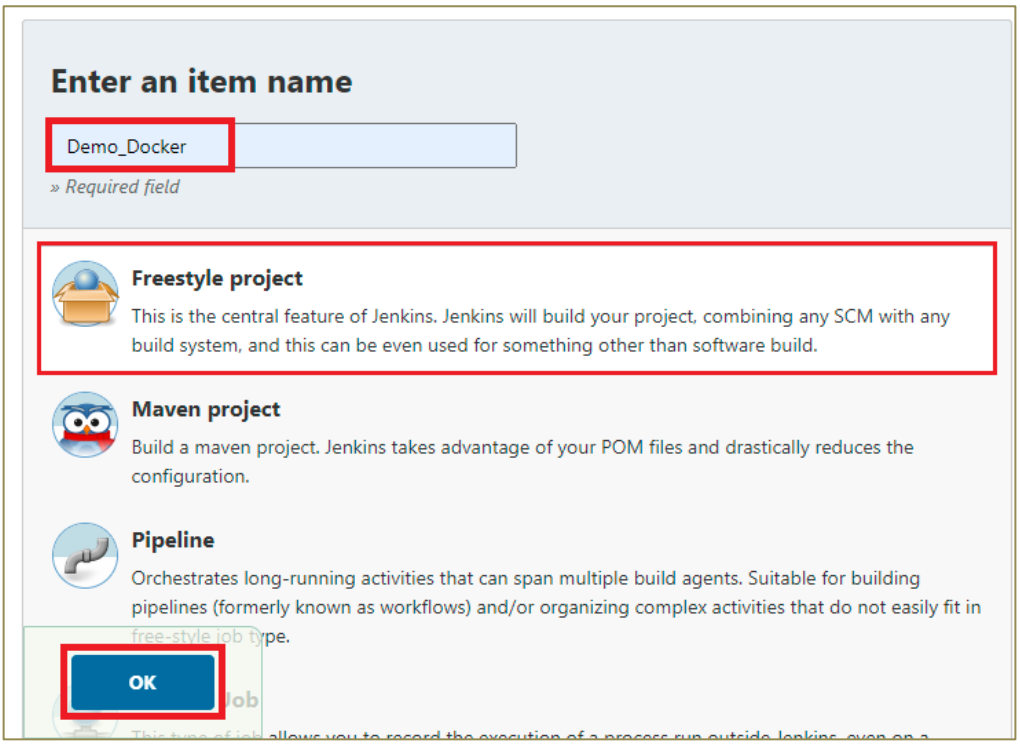
h. Provide the permission to the docker.sock file and restart the Docker as shown below

```
[ec2-user@ip-172-31-35-174 ~]$ sudo chmod 666 /var/run/docker.sock
[ec2-user@ip-172-31-35-174 ~]$
[ec2-user@ip-172-31-35-174 ~]$ sudo service docker restart
Redirecting to /bin/systemctl restart docker.service
[ec2-user@ip-172-31-35-174 ~]$
```

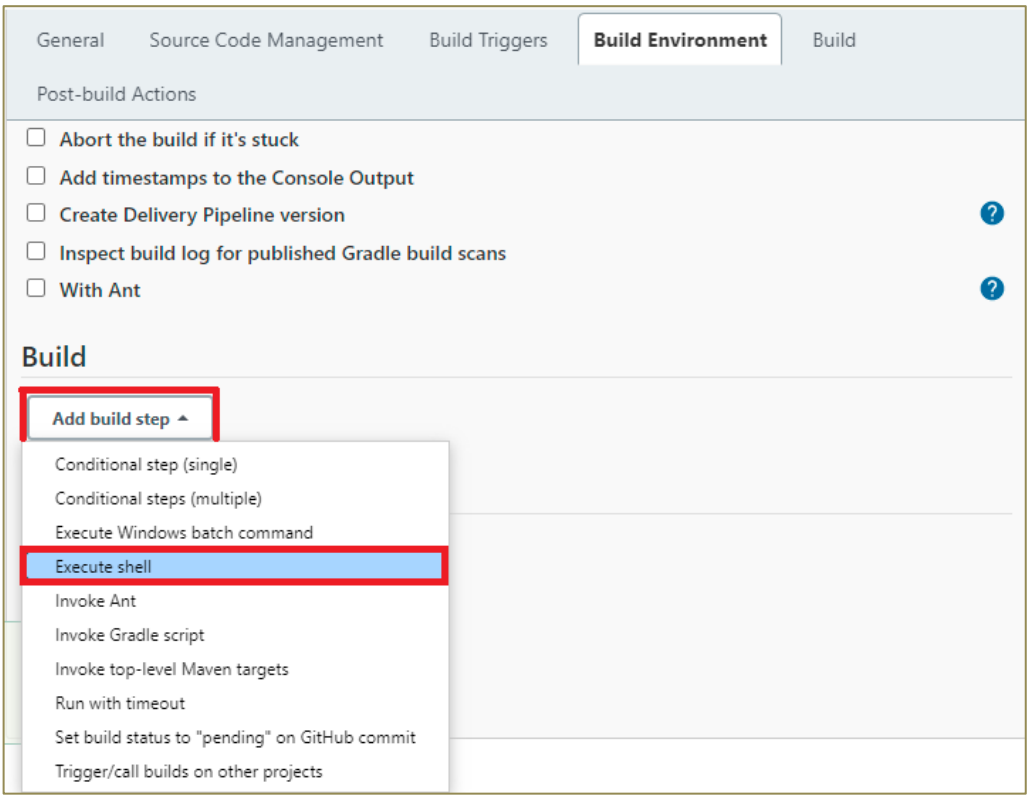
3. Create a sample Jenkins job on Docker.
   a. Open the Jenkins Dashboard, navigate to main menu and select **New Item** to create a sample Docker Job as shown below.



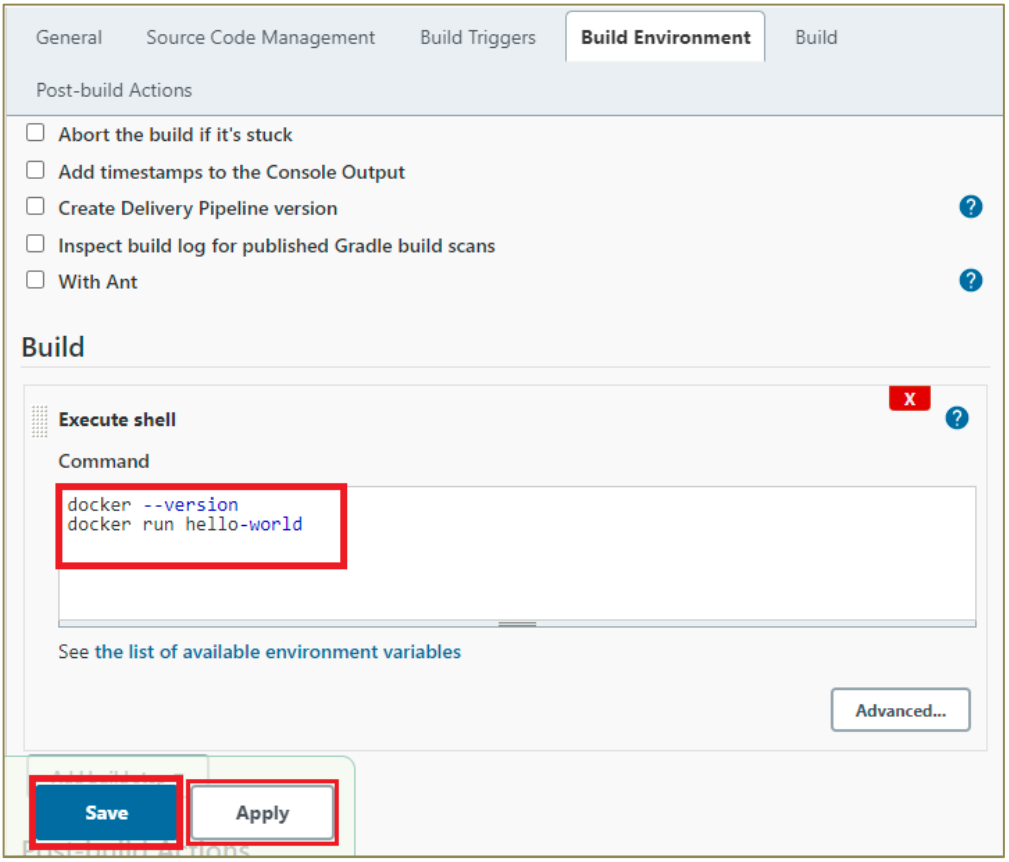
   b. Provide the name for Job, select **Freestyle project** and click **OK.**



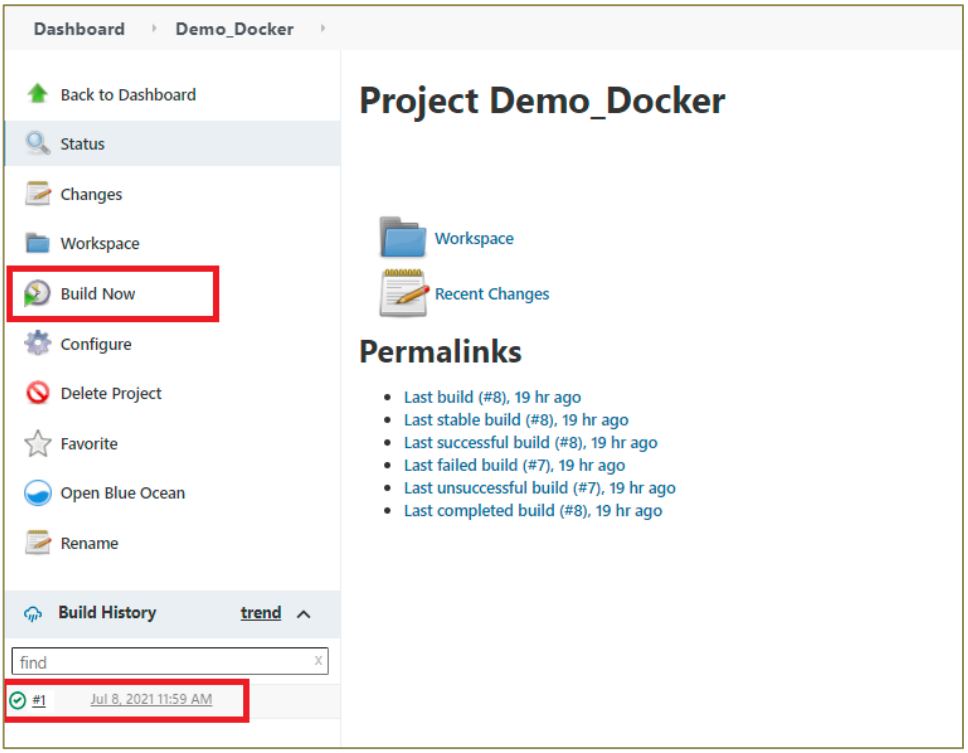
c. Navigate to **Build Environment** and select **Execute Shell** under **Add build step** as shown below.



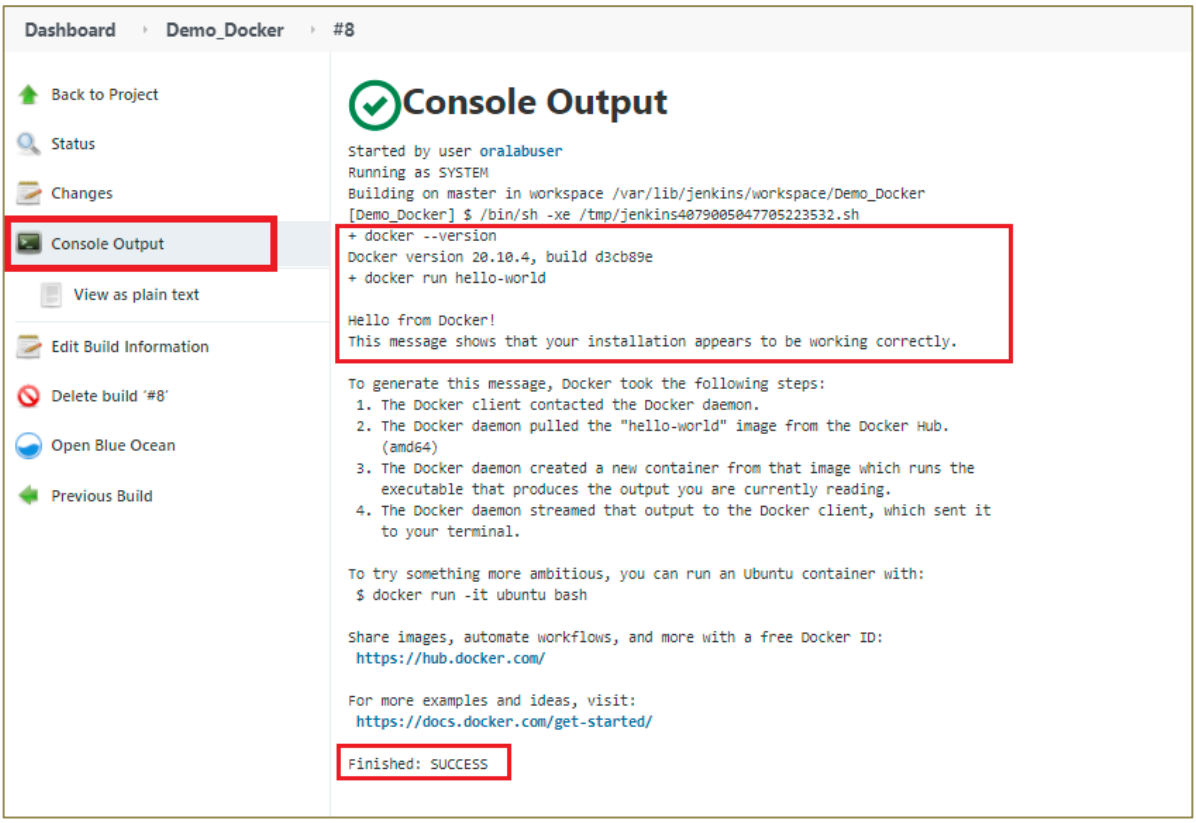
d. Provide the code in **Execute Shell** to execute the Docker commands as shown below. Click **Apply** and **Save.**



e. Job is created successfully, click **Build Now** to execute the job. Click on build created under **Build History** as shown below.



f. In the **Build** page, click **Console Output** to view the output of the job. The Docker commands are executed successfully as shown below.



4. Keep the Jenkins Dashboard, Linux-instance terminal and the AWS Management Console open for the next practice.