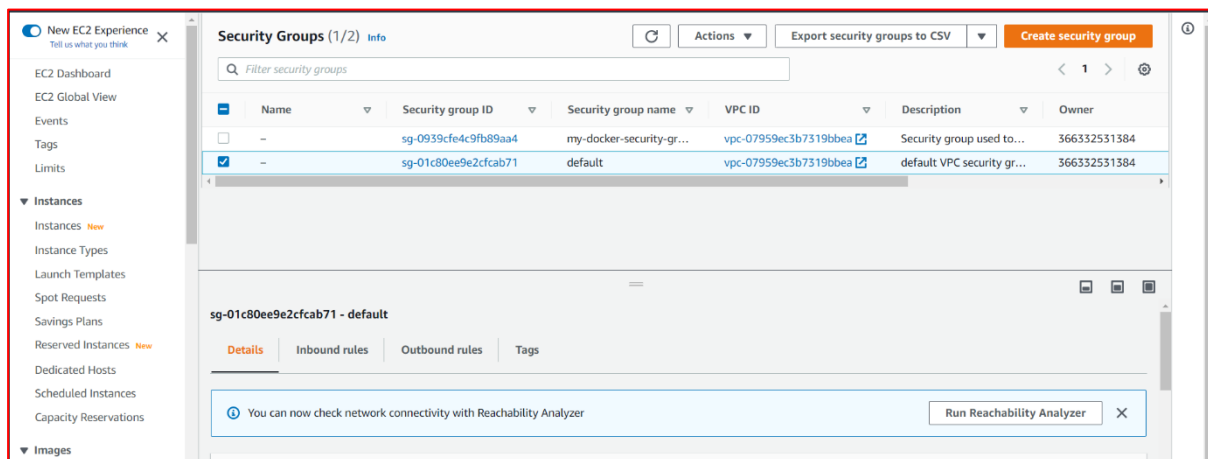


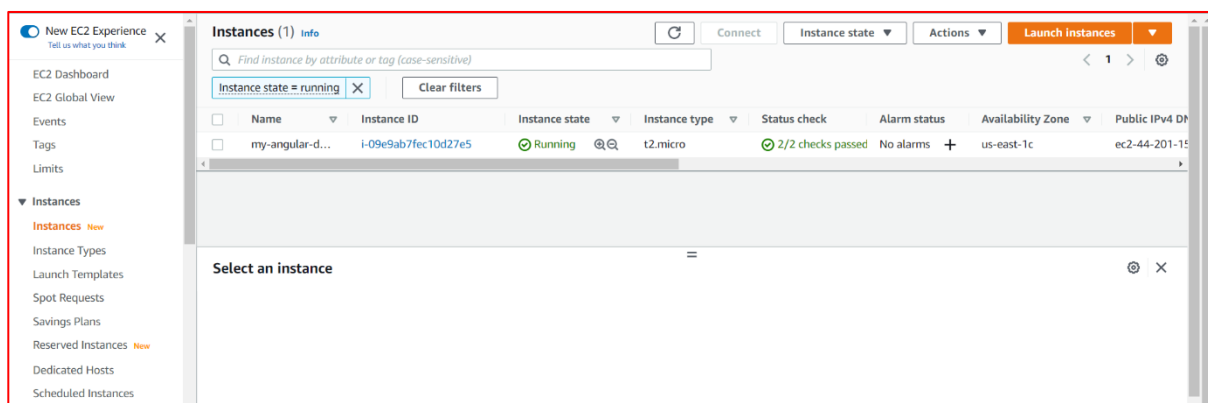
Jenkins Pipeline to Deploy Docker Swarm(Screen Shots)

```
C:\Users\Admin\Desktop\Saveri\Java\programs\Java Phase 5\My images\my image to run angular\angular-docker>docker tag my-angular123 saverisusar1a/my-angular123:1.0
C:\Users\Admin\Desktop\Saveri\Java\programs\Java Phase 5\My images\my image to run angular\angular-docker>docker push saverisusar1a/my-angular123:1.0
The push refers to repository [docker.io/saverisusar1a/my-angular123]
1c428057acde: Pushed
d6a3537fc36a: Mounted from library/nginx
819eb3ad5632: Mounted from library/nginx
5eda6fa69be4: Mounted from library/nginx
6f4f3ce1dca0: Mounted from library/nginx
58a06a0d345c: Mounted from library/nginx
fe7b1e9bf792: Mounted from library/nginx
1.0: digest: sha256:fe816a6c4d7d94a908bb708f061d6eb4e5b56e41ac1cf12b7d1c06bbcb1ab12e size: 1779
C:\Users\Admin\Desktop\Saveri\Java\programs\Java Phase 5\My images\my image to run angular\angular-docker>
```



The screenshot shows the AWS Management Console 'Security Groups' page. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The main content area is titled 'Security Groups (1/2) Info' and includes a search bar, a table of security groups, and a detailed view for the selected 'sg-01c80ee9e2cfcab71 - default' group. The table lists two security groups: 'my-docker-security-gr...' and 'default'. The detailed view for the 'default' group shows tabs for Details, Inbound rules, Outbound rules, and Tags. A notification banner at the bottom suggests using the Reachability Analyzer.

Name	Security group ID	Security group name	VPC ID	Description	Owner
--	sg-0939cfe4c9fb89aa4	my-docker-security-gr...	vpc-07959ec3b7319bbea	Security group used to...	366332531384
--	sg-01c80ee9e2cfcab71	default	vpc-07959ec3b7319bbea	default VPC security gr...	366332531384



The screenshot shows the AWS Management Console 'Instances' page. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The main content area is titled 'Instances (1) Info' and includes a search bar, a table of instances, and a detailed view for the selected 'my-angular-d...' instance. The table lists one instance: 'my-angular-d...' with ID 'i-09eab7fec10d27e5', state 'Running', type 't2.micro', and status '2/2 checks passed'. The detailed view for the instance shows a 'Select an instance' dialog box.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP v4 D...
my-angular-d...	i-09eab7fec10d27e5	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-44-201-15

```
actice-Project-Phase-5/non-assisted/Docker swarm (main)
$ ssh -i "angular-docker123.pem" ec2-user@ec2-44-201-155-226.compute-1.amazonaws.com
The authenticity of host 'ec2-44-201-155-226.compute-1.amazonaws.com (44.201.155.226)' can't be established.
ED25519 key fingerprint is SHA256:KA0fKkMK37LjA9UyBj+bjXDQeQWITWhe3n09juTyzBs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-201-155-226.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

  _ | _ | _ )
 _ | ( _ | /
 _ | \ _ | _ |
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-81-30 ~]$
```

```
ec2-user@ip-172-31-81-30:~$ sudo amazon-linux-extras install java-openjdk11
Installing java-11-openjdk
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-java-openjdk11 amzn2extra-kernel-5.10 jenkins
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-java-openjdk11
amzn2extra-kernel-5.10
jenkins
(1/10): amzn2-core/2/x86_64/updateinfo
(2/10): amzn2-core/2/x86_64/group_gz
(3/10): amzn2extra-java-openjdk11/2/x86_64/primary_db
(4/10): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(5/10): amzn2extra-docker/2/x86_64/updateinfo
(6/10): amzn2extra-docker/2/x86_64/primary_db
(7/10): amzn2extra-java-openjdk11/2/x86_64/updateinfo
(8/10): amzn2extra-kernel-5.10/2/x86_64/primary_db
(9/10): jenkins/primary_db
(10/10): amzn2-core/2/x86_64/primary_db
```

```
ec2-user@ip-172-31-81-30:~$ sudo yum install jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.372-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
jenkins noarch 2.372-1.1 jenkins 89 M
Transaction Summary
-----
Install 1 Package

Total download size: 89 M
Installed size: 89 M
Is this ok [y/d/N]: y
Downloading packages:
jenkins-2.372-1.1.noarch.rpm | 89 MB 00:00:26
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : jenkins-2.372-1.1.noarch 1/1
Verifying : jenkins-2.372-1.1.noarch 1/1

Installed:
jenkins.noarch 0:2.372-1.1

Complete!
[ec2-user@ip-172-31-81-30 ~]$
```

```
ec2-user@ip-172-31-81-30:~  
[ec2-user@ip-172-31-81-30 ~]$ sudo service jenkins start  
Starting jenkins (via systemctl): [ OK ]  
[ec2-user@ip-172-31-81-30 ~]$
```

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

```
ec2-user@ip-172-31-81-30:~  
[ec2-user@ip-172-31-81-30 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
7f78327b98a846269d6e10e5d59e3e65  
[ec2-user@ip-172-31-81-30 ~]$
```

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<div><div>STATUS</div><div><div>** Token Macro</div><div>Build Timeout</div><div>** Credentials</div><div>** Trilead API</div><div>** SSH Credentials</div><div>** Pipeline: Step API</div><div>** Plain Credentials</div><div>Credentials Binding</div><div>** SCH API</div><div>** Pipeline: API</div></div></div> <div>Timestamp:</div> <div><div>** Caffeine API</div><div>** Script Security</div><div>Icons API</div><div>** JAXB</div><div>** SnakeYAML API</div><div>** Jackson 2 API</div><div>** commons-lang3 v3.x Jenkins API</div><div>** commons-text API</div><div>** Plugin Utilities API</div><div>** Font Awesome API</div><div>** Popper.js 2 API</div><div>** Bootstrap 5 API</div><div>** - required dependency</div></div>
✓ Timestamp	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

Jenkins 2.372



Jenkins

Search (CTRL+K)



Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

Build Queue



No builds in the queue.

Build Executor Status



1 Idle

2 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

Create a job



Set up a distributed build

Set up an agent



Configure a cloud



Learn more about distributed builds



```
ec2-user@ip-172-31-81-30:~$ sudo yum install docker
--> Running transaction check
--> Package containerd.x86_64 0:1.6.6-1.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 0:1.1.3-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package Arch Version Repository Size
Installing:
docker x86_64 20.10.17-1.amzn2 amzn2extra-docker 39 M
Installing for dependencies:
containerd x86_64 1.6.6-1.amzn2 amzn2extra-docker 27 M
libcgroup x86_64 0.41-21.amzn2 amzn2-core 66 k
pigz x86_64 2.3.4-1.amzn2.0.1 amzn2-core 81 k
runc x86_64 1.1.3-1.amzn2 amzn2extra-docker 2.9 M


Transaction Summary
Install 1 Package (+4 Dependent packages)

Total download size: 69 M
Installed size: 260 M
Is this ok [y/d/n]: y
Downloading packages:
(1/5): libcgroup-0.41-21.amzn2.x86_64.rpm | 66 kB 00:00:00
(2/5): pigz-2.3.4-1.amzn2.0.1.x86_64.rpm | 81 kB 00:00:00
(3/5): containerd-1.6.6-1.amzn2.x86_64.rpm | 27 MB 00:00:00
(4/5): docker-20.10.17-1.amzn2.x86_64.rpm | 39 MB 00:00:00
(5/5): runc-1.1.3-1.amzn2.x86_64.rpm | 2.9 MB 00:00:00
Total 69 MB/s | 69 MB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : runc-1.1.3-1.amzn2.x86_64 1/3
Installing : containerd-1.6.6-1.amzn2.x86_64 2/3
Installing : libcgroup-0.41-21.amzn2.x86_64 3/3
Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/3
Installing : docker-20.10.17-1.amzn2.x86_64 5/3
Verifying : docker-20.10.17-1.amzn2.x86_64 1/3
Verifying : runc-1.1.3-1.amzn2.x86_64 2/3
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64 3/3
Verifying : containerd-1.6.6-1.amzn2.x86_64 4/3
Verifying : libcgroup-0.41-21.amzn2.x86_64 5/3


Installed:
docker.x86_64 0:20.10.17-1.amzn2


Dependency Installed:
containerd.x86_64 0:1.6.6-1.amzn2 libcgroup.x86_64 0:0.41-21.amzn2 pigz.x86_64 0:2.3.4-1.amzn2.0.1 runc.x86_64 0:1.1.3-1.amzn2


Complete!
[ec2-user@ip-172-31-81-30 ~]$
```


 View as plain text


☒ Edit Build Information


 Delete build '#6'

 Restart from Stage

 Replay

 Pipeline Steps

 Workspaces

 Previous Build

[Pipeline] node

Running on [Jenkins](#) in /var/lib/jenkins/workspace/pipelineJob

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Check Docker version)

[Pipeline] sh

+ docker --version

Docker version 20.10.17, build 100c701

[Pipeline] sh

+ docker pull saverisusarla/my-angular123:1.0

1.0: Pulling from saverisusarla/my-angular123

Digest: sha256:fe816a6c4d7d94a908bb708f061d6ebde5b56e41ac1cf12b7d1c06bbcb1ab12e

Status: Image is up to date for saverisusarla/my-angular123:1.0

docker.io/saverisusarla/my-angular123:1.0

[Pipeline] sh

+ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

saverisusarla/my-angular123 1.0 e8893e2e3c9d 5 hours ago 142MB

[Pipeline] sh

+ docker swarm init

Swarm initialized: current node (655ml4q7ho3avh32dbtrxozy1) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-10c0w7j3jcmis986u955koxsm13s5csfzqs749vi742zjtkdp-f02oeurhsxkp2lpnoqsmymy8ng 172.31.81.30:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

```
Dashboard > pipelineJob > #6

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

[Pipeline] sh
+ docker node ls
ID                                HOSTNAME                                STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
655nl4q7ho3avh32dtrxozy1 *      ip-172-31-81-30.ec2.internal          Ready   Active         Leader           20.10.17
[Pipeline] sh
+ docker service create -p 80:80 --name myService saverisusarla/my-angular123:1.0
ixilvrd0qfzi6eke8zob4f1o
overall progress: 0 out of 1 tasks
1/1:
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 0 out of 1 tasks
overall progress: 1 out of 1 tasks
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 4 seconds to verify that tasks are stable...
verify: Waiting 4 seconds to verify that tasks are stable...
verify: Waiting 4 seconds to verify that tasks are stable...
```

```
Dashboard > pipelineJob > #6

verify: Waiting 3 seconds to verify that tasks are stable...
verify: Waiting 3 seconds to verify that tasks are stable...
verify: Waiting 3 seconds to verify that tasks are stable...
verify: Waiting 3 seconds to verify that tasks are stable...
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Waiting 2 seconds to verify that tasks are stable...
verify: Waiting 1 seconds to verify that tasks are stable...
verify: Waiting 1 seconds to verify that tasks are stable...
verify: Waiting 1 seconds to verify that tasks are stable...
verify: Waiting 1 seconds to verify that tasks are stable...
verify: Waiting 1 seconds to verify that tasks are stable...
verify: Service converged
[Pipeline] sh
+ docker service ls
ID            NAME      MODE     REPLICAS  IMAGE                                  PORTS
ixilvrd0qfz  myService replicated 1/1        saverisusarla/my-angular123:1.0      *:80->80/tcp
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```