Task 1: Setup a very simple Jenkins as a container and 1 or more Jenkins slaves (as containers again), all in the laptop itself preferably and show the following:

a) Create dummy task, spawn the job multiple times and show that the task can run on any of the Jenkins nodes (thus demonstrating horizontal scalability, without auto-scaling)

b) Using Docker or Kubernetes, Show how auto-scaling can be achieved - where the Jenkins nodes (or slaves) get created dynamically and destroyed after job completion.

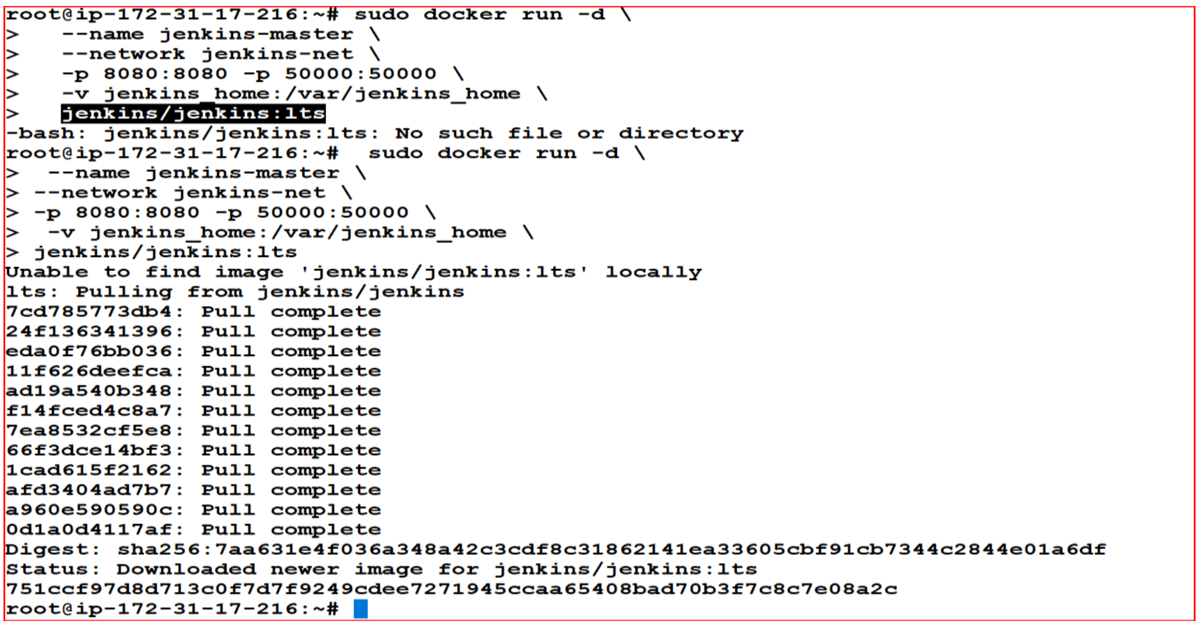
Created the EC2 instance with ubuntu 24.04 & install the Docker:

And Run the below commans:

```
# sudo docker network create jenkins-net
```

```
#  sudo docker run -d \
>   --name jenkins-master \
>   --network jenkins-net \
>   -p 8080:8080 -p 50000:50000 \
>   -v jenkins_home:/var/jenkins_home \
>   jenkins/jenkins:lts
```

You will get the output like this:



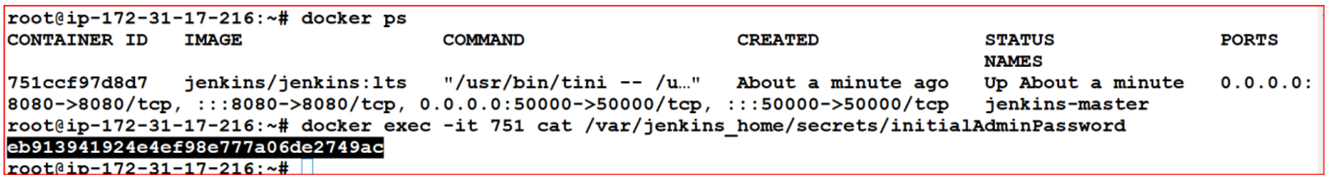
```
root@ip-172-31-17-216:~# sudo docker run -d \
>    --name jenkins-master \
>    --network jenkins-net \
>    -p 8080:8080 -p 50000:50000 \
>    -v jenkins_home:/var/jenkins_home \
>    jenkins/jenkins:lts
-bash: jenkins/jenkins:lts: No such file or directory
root@ip-172-31-17-216:~#  sudo docker run -d \
>   --name jenkins-master \
> --network jenkins-net \
> -p 8080:8080 -p 50000:50000 \
>   -v jenkins_home:/var/jenkins_home \
> jenkins/jenkins:lts
Unable to find image 'jenkins/jenkins:lts' locally
lts: Pulling from jenkins/jenkins
7cd785773db4: Pull complete
24f136341396: Pull complete
eda0f76bb036: Pull complete
11f626deefca: Pull complete
ad19a540b348: Pull complete
f14fced4c8a7: Pull complete
7ea8532cf5e8: Pull complete
66f3dce14bf3: Pull complete
1cad615f2162: Pull complete
afd3404ad7b7: Pull complete
a960e590590c: Pull complete
0d1a0d4117af: Pull complete
Digest: sha256:7aa631e4f036a348a42c3cdf8c31862141ea33605cbf91cb7344c2844e01a6df
Status: Downloaded newer image for jenkins/jenkins:lts
751ccf97d8d713c0f7d7f9249cdee7271945ccaa65408bad70b3f7c8c7e08a2c
root@ip-172-31-17-216:~#
```

Jenkins initial setup is required. An admin user has been created and a password generated.

Please use the following password to proceed to installation:

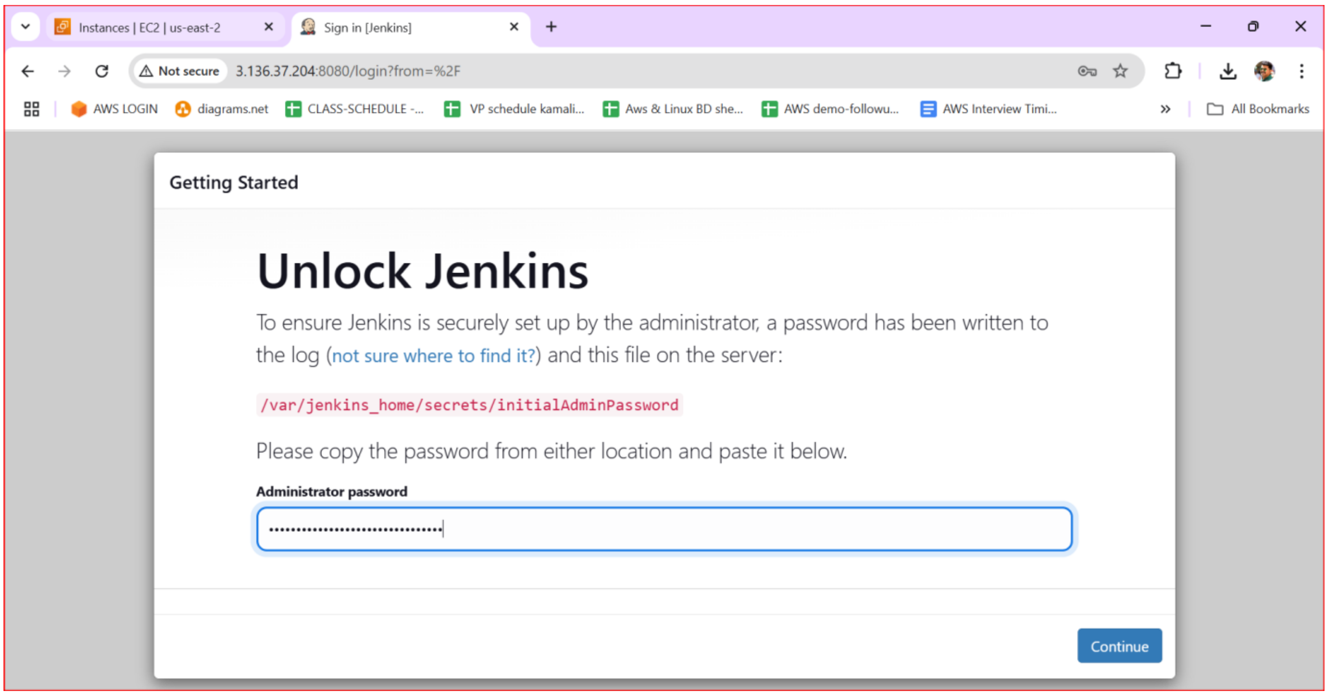
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

docker exec -it 751 cat /var/jenkins_home/secrets/initialAdminPassword



```
root@ip-172-31-17-216:~# docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED              STATUS              PORTS
                                                                                   NAMES
751ccf97d8d7   jenkins/jenkins:lts   "/usr/bin/tini -- /u…"   About a minute ago   Up About a minute   0.0.0.0:
8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp   jenkins-master
root@ip-172-31-17-216:~# docker exec -it 751 cat /var/jenkins_home/secrets/initialAdminPassword
eb913941924e4ef98e777a06de2749ac
root@ip-172-31-17-216:~#
```

Now we can access the Jenkins from browser:



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/jenkins_home/secrets/initialAdminPassword

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

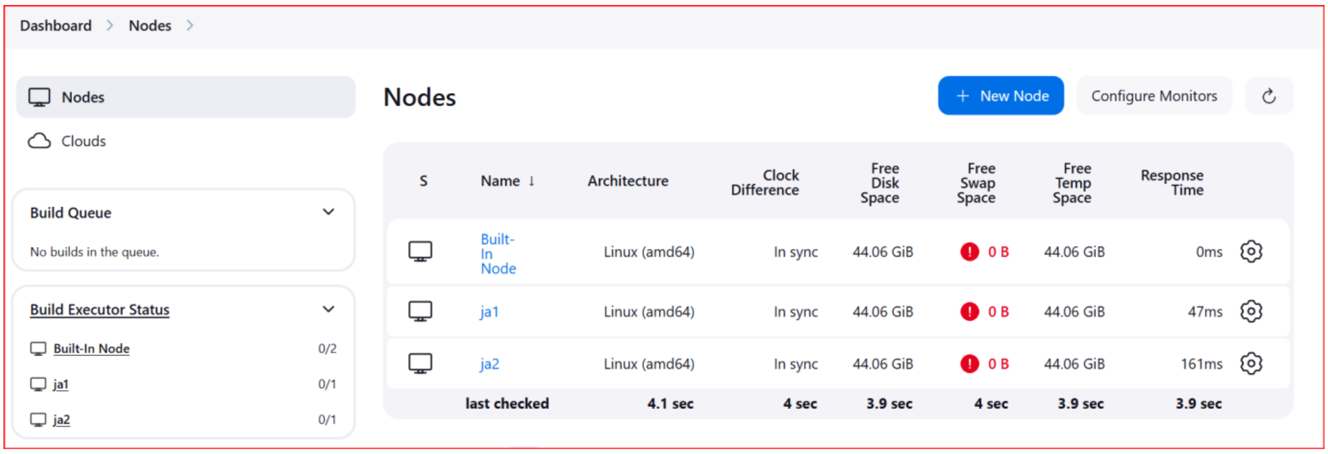
Administrator password

································

Continue

After login to the Jenkins need to create 2 nodes:

Manage Jenkins -> node -> create node

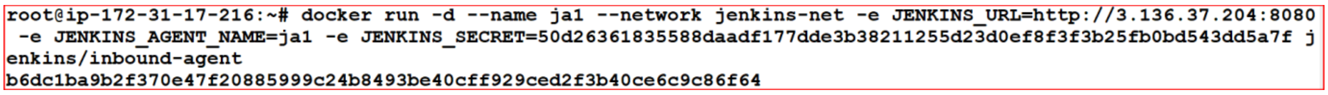


Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	44.06 GiB	0 B	44.06 GiB	0ms
	ja1	Linux (amd64)	In sync	44.06 GiB	0 B	44.06 GiB	47ms
	ja2	Linux (amd64)	In sync	44.06 GiB	0 B	44.06 GiB	161ms
	last checked	4.1 sec	4 sec	3.9 sec	4 sec	3.9 sec	3.9 sec

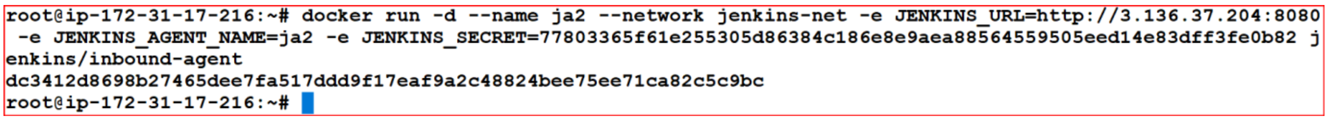
Get the secrets from the created nodes and create a container using the commands to node online:

docker run -d --name ja1 --network jenkins-net -e JENKINS_URL=http://3.136.37.204:8080 -e JENKINS_AGENT_NAME=ja1 -e JENKINS_SECRET=50d26361835588daadf177dde3b38211255d23d0ef8f3f3b25fb0bd543dd5a7f jenkins/inbound-agent



```
root@ip-172-31-17-216:~# docker run -d --name ja1 --network jenkins-net -e JENKINS_URL=http://3.136.37.204:8080
 -e JENKINS_AGENT_NAME=ja1 -e JENKINS_SECRET=50d26361835588daadf177dde3b38211255d23d0ef8f3f3b25fb0bd543dd5a7f j
enkins/inbound-agent
b6dc1ba9b2f370e47f20885999c24b8493be40cff929ced2f3b40ce6c9c86f64
```

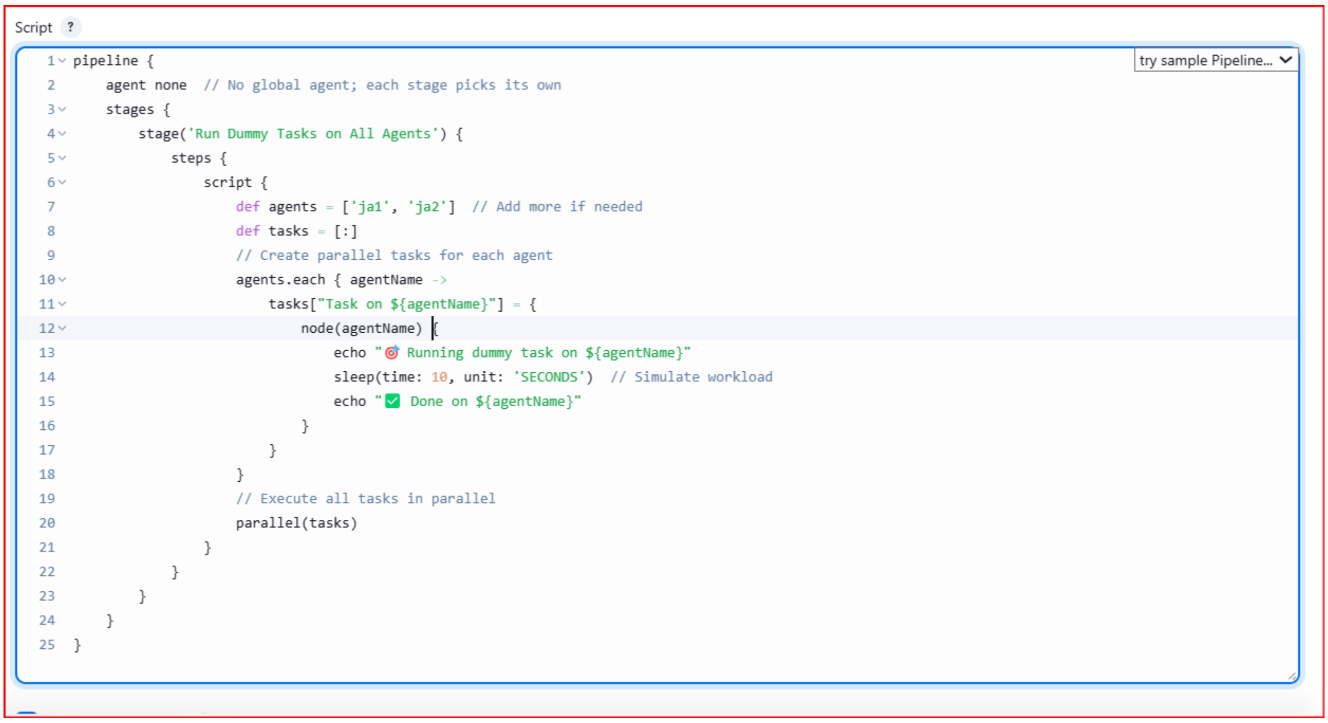
docker run -d --name ja2 --network jenkins-net -e JENKINS_URL=http://3.136.37.204:8080 -e JENKINS_AGENT_NAME=ja2 -e JENKINS_SECRET=77803365f61e255305d86384c186e8e9aea88564559505eed14e83dff3fe0b82 jenkins/inbound-agent



```
root@ip-172-31-17-216:~# docker run -d --name ja2 --network jenkins-net -e JENKINS_URL=http://3.136.37.204:8080
 -e JENKINS_AGENT_NAME=ja2 -e JENKINS_SECRET=77803365f61e255305d86384c186e8e9aea88564559505eed14e83dff3fe0b82 j
enkins/inbound-agent
dc3412d8698b27465dee7fa517ddd9f17eaf9a2c48824bee75ee71ca82c5c9bc
root@ip-172-31-17-216:~#
```

Create a dummy pipeline job:

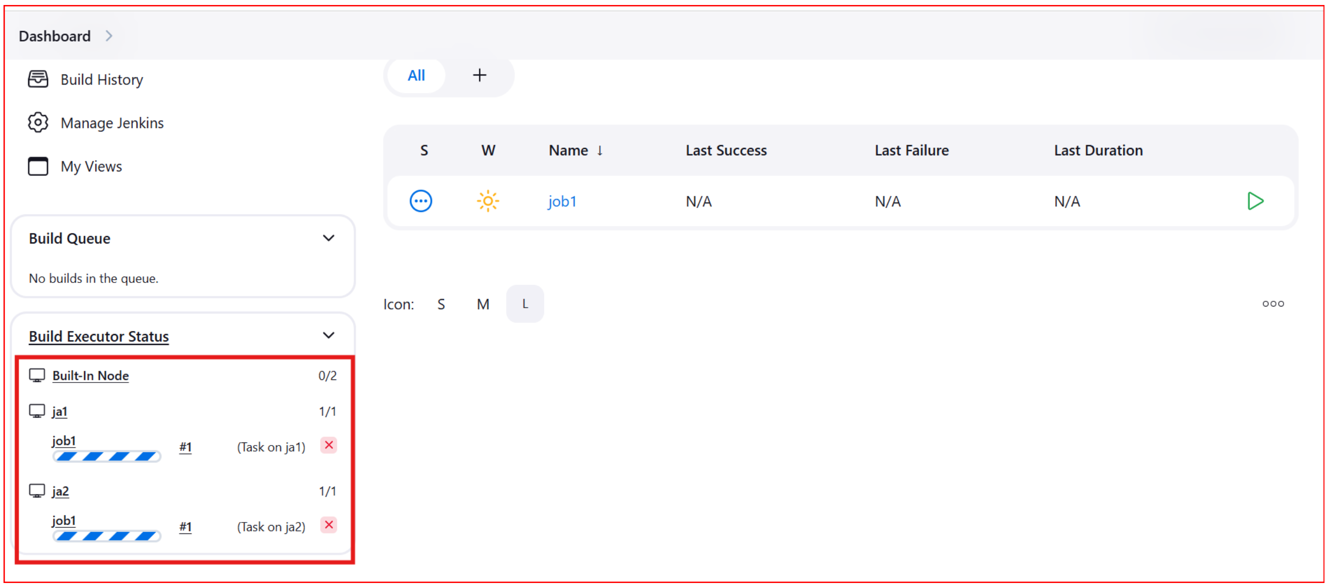
Using the below sample pipeline script:



```
pipeline {
    agent none  // No global agent; each stage picks its own
    stages {
        stage('Run Dummy Tasks on All Agents') {
            steps {
                script {
                    def agents = ['ja1', 'ja2']  // Add more if needed
                    def tasks = [:]
                    // Create parallel tasks for each agent
                    agents.each { agentName ->
                        tasks["Task on ${agentName}"] = {
                            node(agentName) {
                                echo "🎯 Running dummy task on ${agentName}"
                                sleep(time: 10, unit: 'SECONDS')  // Simulate workload
                                echo "✅ Done on ${agentName}"
                            }
                        }
                    }
                    // Execute all tasks in parallel
                    parallel(tasks)
                }
            }
        }
    }
}
```

Task A output completed.

If we build the job it will run on Jenkins agent as a container:


Dashboard >

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node			0/2
ja1			1/1
job1	#1	(Task on ja1)	
ja2			1/1
job1	#1	(Task on ja2)	

All +

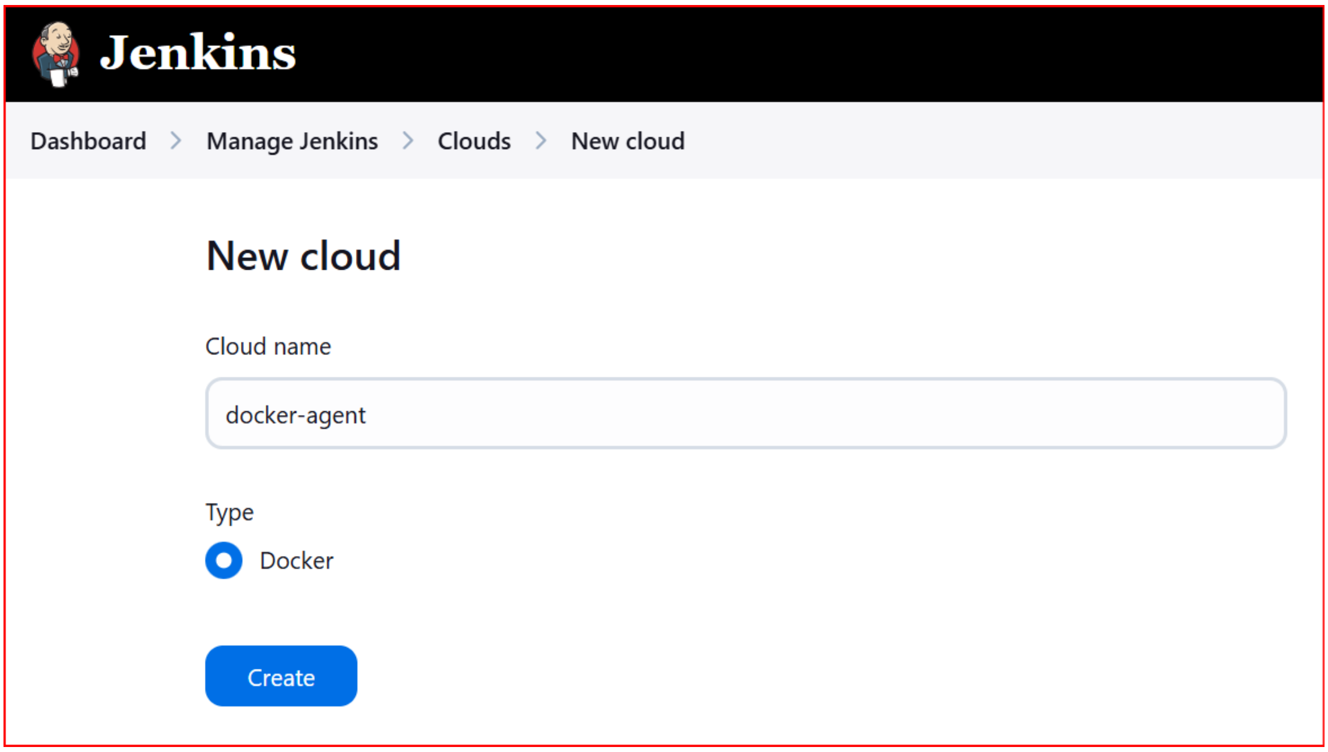
S	W	Name ↓	Last Success	Last Failure	Last Duration
		job1	N/A	N/A	N/A

Icon: S M L

B) Using Docker or Kubernetes, Show how auto-scaling can be achieved - where the Jenkins nodes (or slaves) get created dynamically and destroyed after job completion

Step1:

1. **Install the docker plugin in Jenkins**
2. **Manage jenkins -> cloud -> new cloud**


Jenkins

Dashboard > Manage Jenkins > Clouds > New cloud

New cloud

Cloud name

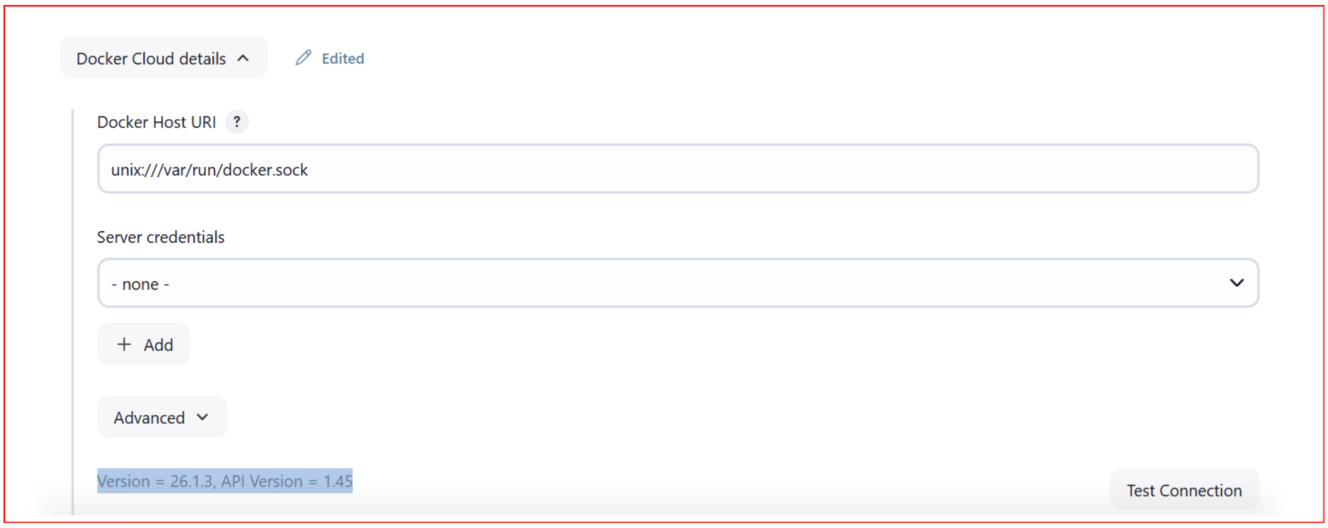
docker-agent

Type

[✓ Docker]

Create

Need to mention the docker host url as below



Docker Host URI

unix:///var/run/docker.sock

Server credentials

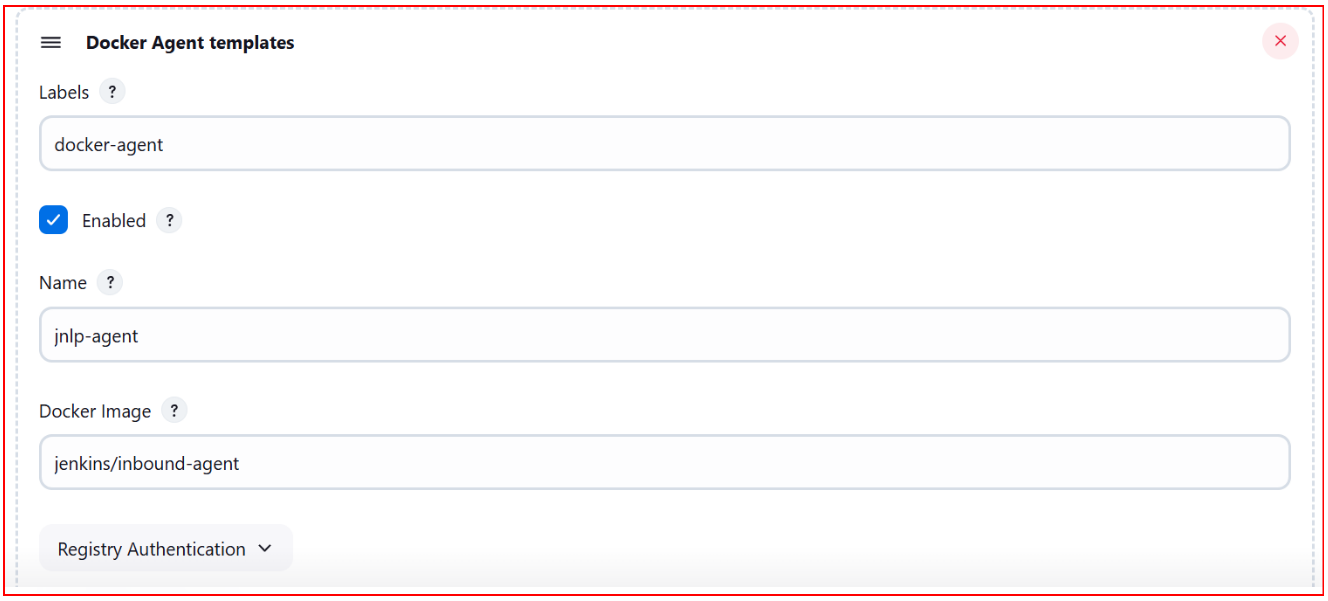
- none -

Version = 26.1.3, API Version = 1.45

If facing error with testing the conatainer need to delete the jenkins master and run the below command to create a new jenkins master with volme

```
# docker run -d \
  --name jenkins \
  -p 8080:8080 -p 50000:50000 \
  -v jenkins_home:/var/jenkins_home \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -e "JAVA_OPTS=-Dhudson.security.csrf.GlobalCrumbIssuerConfiguration.DISABLE_CSRF_PROTECTION=true" \
  --user root \
  jenkins/jenkins:lts
```

Add the agent template



Labels

docker-agent

Enabled ✓

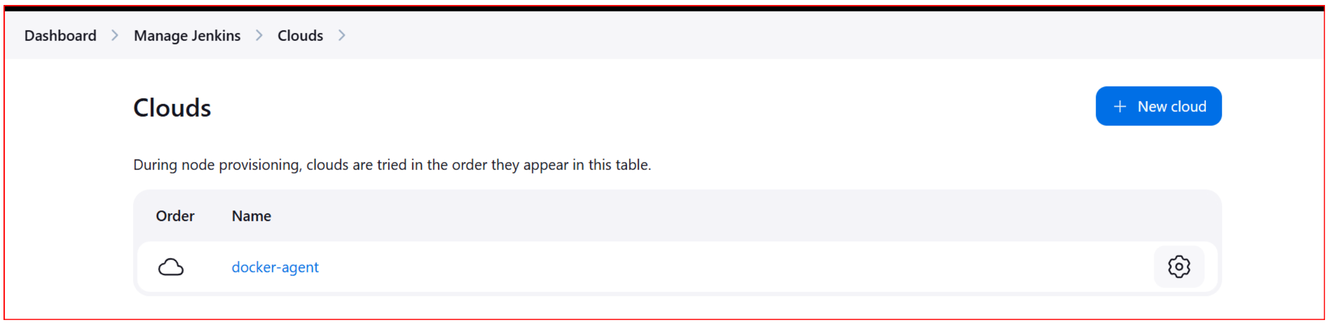
Name

jnlp-agent

Docker Image

jenkins/inbound-agent

Cloud docker agent got created:

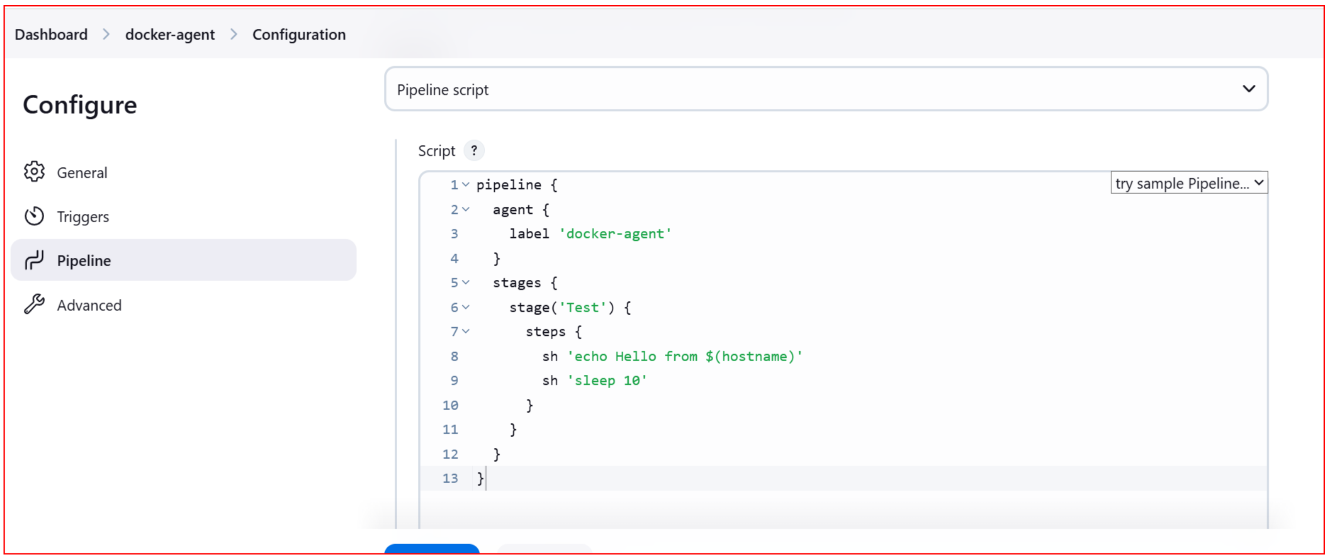

Dashboard > Manage Jenkins > Clouds >

Clouds

During node provisioning, clouds are tried in the order they appear in this table.

Order	Name
	docker-agent

Create a new docker-agent job with simple pipeline:


Dashboard > docker-agent > Configuration

Configure

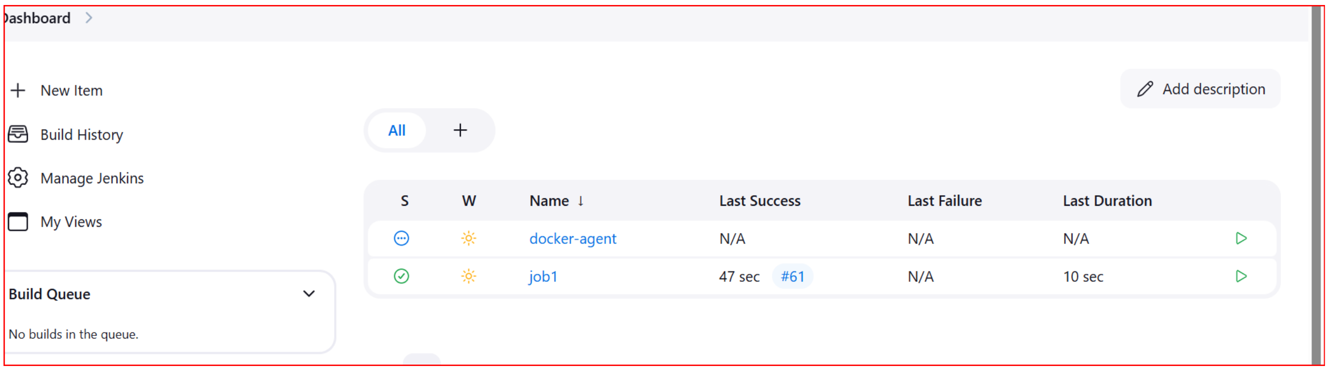
General
Triggers
Pipeline
Advanced

Pipeline script

Script

```
pipeline {
  agent {
    label 'docker-agent'
  }
  stages {
    stage('Test') {
      steps {
        sh 'echo Hello from $(hostname)'
        sh 'sleep 10'
      }
    }
  }
}
```

Job created if the job is running means it will use the docker cloud agent to run the job after job execution the container will automatically getting deleted..


Dashboard >

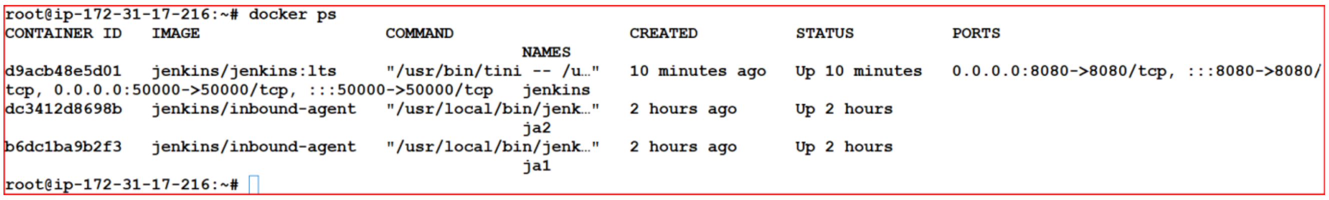
New Item
Build History
Manage Jenkins
My Views

Build Queue

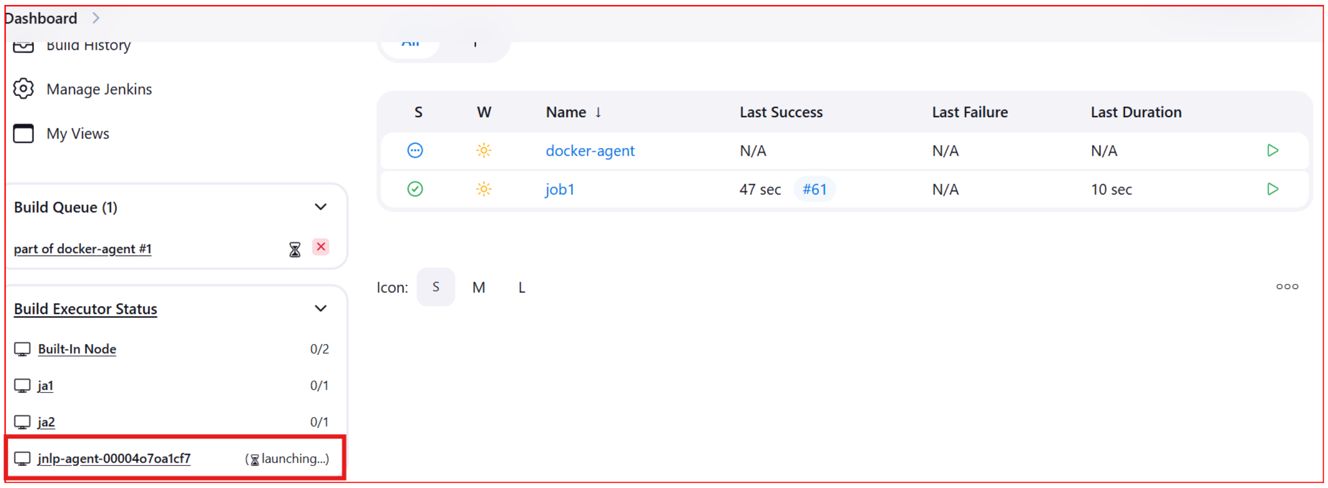
No builds in the queue.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		docker-agent	N/A	N/A	N/A
✓		job1	47 sec #61	N/A	10 sec

As of now I have container as given below


```
root@ip-172-31-17-216:~# docker ps
CONTAINER ID   IMAGE                      COMMAND                  CREATED          STATUS          PORTS
                                                  NAMES
d9acb48e5d01   jenkins/jenkins:lts        "/usr/bin/tini -- /u…"   10 minutes ago   Up 10 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/
tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp   jenkins
dc3412d8698b   jenkins/inbound-agent      "/usr/local/bin/jenk…"   2 hours ago      Up 2 hours
                                                  ja2
b6dc1ba9b2f3   jenkins/inbound-agent      "/usr/local/bin/jenk…"   2 hours ago      Up 2 hours
                                                  ja1
root@ip-172-31-17-216:~#
```

Job is running and agent is getting creating



Dashboard >

Build History

Manage Jenkins

My Views

Build Queue (1)

part of docker-agent #1

Build Executor Status

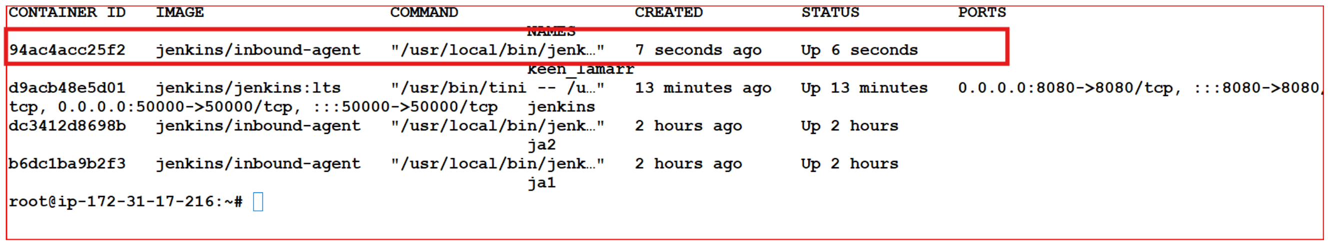
Built-In Node	0/2
ja1	0/1
ja2	0/1
jnlp-agent-00004o7oa1cf7	(launching...)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
⊙	☼	docker-agent	N/A	N/A	N/A	▷
✓	☼	job1	47 sec #61	N/A	10 sec	▷

Icon: S M L

New containe getting created and after the job completion its getting deleted successfully

Task A output completed.



```
CONTAINER ID   IMAGE                   COMMAND                  CREATED          STATUS          PORTS
                                                            NAMES
94ac4acc25f2   jenkins/inbound-agent   "/usr/local/bin/jenk…"   7 seconds ago    Up 6 seconds
                                                            keen_lamarr
d9acb48e5d01   jenkins/jenkins:lts     "/usr/bin/tini -- /u…"   13 minutes ago   Up 13 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/
tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp   jenkins
dc3412d8698b   jenkins/inbound-agent   "/usr/local/bin/jenk…"   2 hours ago      Up 2 hours
                                                            ja2
b6dc1ba9b2f3   jenkins/inbound-agent   "/usr/local/bin/jenk…"   2 hours ago      Up 2 hours
                                                            ja1
root@ip-172-31-17-216:~# 
```

Using Docker where the Jenkins nodes (or slaves) get created dynamically and destroyed after job completion.