

How To - Elastic Ephemeral Build Jenkins Slaves

1. Plugin installation in Jenkins -

There are few plugins need to install in Jenkins to use Docker within Jenkins jobs and Pipeline. Go to Manage Jenkins >> Manage Plugins >> Available tab >> search for Docker then install Docker Plugin, Docker Pipeline and Docker Slave Plugin.

2. Setting Docker Cloud Setting in Jenkins -

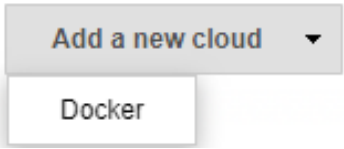
To use the docker within Jenkins jobs and pipeline; need to configure docker in Jenkins as other tools.

Go to configure security settings "Enable Agent Master Access Control" and check this.

Agent → Master Security


☒ Enable Agent → Master Access Control

Go to Manage Jenkins >> Configure System >> Search for an "Add a new cloud" >> Docker.



The screenshot shows the 'Add a new cloud' dropdown menu in Jenkins. The 'Docker' option is selected and highlighted.

Cloud

 Docker	
Name	<input type="text" value="docker"/>
Docker Host URI	<input type="text" value="unix://0.0.0.0:2376"/>
Server credentials	<input type="text" value="- none -"/> <input type="button" value="Add"/>

Mentioned any Name to save this configuration. However Docker Host URI needs to mention, there are three types of Docker Host URI can be given like fd://, unix:// and tcp:// along with IP address (can be given 0.0.0.0) and port numbers (2375, 2376 and 4243). Then click on "Test Connection", you should be able to see text like "Version = 19.03.2, API Version = 1.40" if connection is successful.

If any error is coming then need to investigate further, however one probable reason could be to assign the permission of "/var/run/docker.sock" in host machine.

3. Docker Slave Container Settings -

Need to configure the slave container under "Docker Agent Templates".

Docker Agent templates

Docker Agent templates

Labels

slave1

Enabled

☒

Name

docker-slave

Docker Image

vik/jenkinslave

Registry Authentication...

Container settings...

Instance Capacity

Remote File System Root

/home/jenkins/

Usage

Only build jobs with label expressions matching this node

Idle timeout

10

Connect method

Attach Docker container

Prerequisites:

- Docker image must have [Java](#) installed.
- Entrypoint must be able to accept jenkins slave connection parameters. See [jenkins/docker-inlp-slave](#) as an example.

User

Remove volumes

☐

Pull strategy

Never pull

Pull timeout

300

Node Properties

Add Node Property

Delete Docker Template

- **Labels:** How you will identify this image in your pipelines or job.
- **Docker image:** The name of the image related to this template.
- **Registry Authentication:** The credentials used to log in to your registry account(DockerHub, ECR).
- **Remote Filing System Root:** The home folder for the user we have created in the image, in this case, /home/jenkins.
- **Usage:** Few options are available, however we have selected "Only build jobs with label expression matching this mode" for our use.
- **Connect Method:** Few options are available "Attach Docker Container", "Connect with SSH" and "Connect with JNLP" like Select "Attach Docker Container" which our images were built to be accessed with.
- **User:** Name of the user used to connect, so, jenkins.
- **Pull Strategy:** Options are available as "Never Pull", "Pull all images every time", "Pull once and update latest". So "Pull once and update latest" option is most suitable to choose.

4. Jenkins Job Execution -

4.1 Free Style Job -

When working for Jenkins Freestyle job then need to provide the option where we want to execute the job like below-

☒ Restrict where this project can be run

Label Expression

slave1

[Label slave1](#) is serviced by no nodes and 1 cloud. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

4.2 Through Jenkins File -

When Jenkinsfile is written then we need to mention the agent like below-

```

pipeline {
    agent {label 'slave1'}
    stages {
        stage("POLL SCM") {
            steps {
                echo "Poll SCM"
            }
        }
        stage("Build") {
            steps {
                echo "Build"
            }
        }
        stage("Test") {
            steps {
                echo "Test"
            }
        }
        stage("Deploy") {
            steps {
                echo "Deploy"
            }
        }
    }
}

```

```

pipeline {
    stages {
        stage("Build") {
            agent { label 'slave1' }
            //...
        }

        stage("Test") {
            agent { label 'slave2' }
            //...
        }
    }
}

```

5. Dockerfile -

Dockerfile used for Jenkins slave, which has capabilities for R and Python code base execution within the slave container.

```
FROM jenkinsci/slave:3.35-3
USER root
RUN apt-get update && \
    apt-get install --no-install-recommends -y \
    r-base \
    libopenblas-base \
    && rm -rf /tmp/downloaded_packages/ /tmp/*.rds \
    && rm -rf /var/lib/apt/lists/*
USER jenkins
```