Configure Jenkins to build Docker Images based on a Dockerfile:

1. Create Jenkins VM and install all the suggested plugins.

2. Under Available Manage Plugins section, search for docker, there are multiple Docker plugins, docker compose, docker build plugins. Please install all docker plugins.

3. Select **Configure System** to access the main Jenkins settings.

4. At the bottom, there is a dropdown called **Add a new cloud**. Select **Docker** from the list.

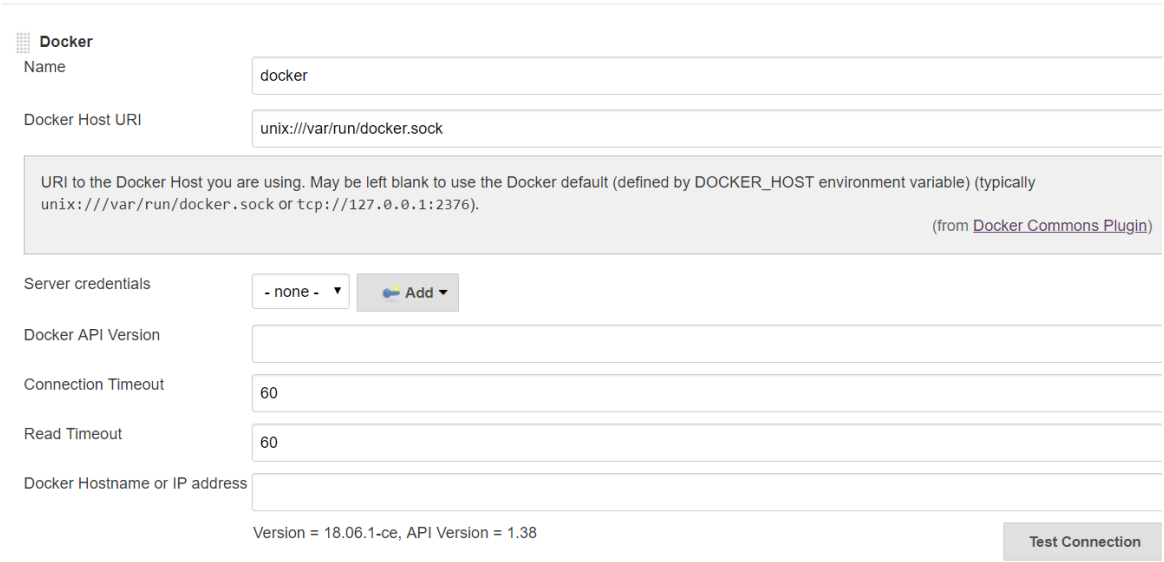
   Now we need docker host uri, The `docker.sock` file is owned by root and does not allow write permissions by other. You have to make it such that `jenkins` can read/write to that socket file when mounted. (That's a bad idea though security wise)

   To enable the docker host URL you need to change permission on your jenkins ec2 instance.

   chmod 777 /var/run/docker.sock

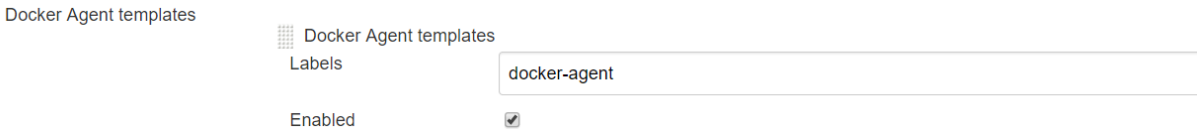
   under Docker Host URI put **unix://var/run/docker.sock**

   and click on Test Connection, it will display installed docker and API version



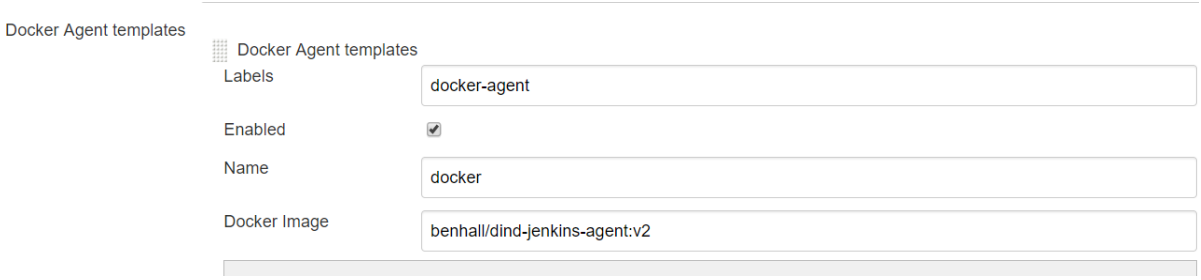
5. Click **Docker Agent templates** and then **Add Docker Template**. You can now configure the container options.

6. Set the label of the agent to `docker-agent`. This is used by the Jenkins builds to indicate it should be built via the Docker Agent we're defining and **Enabled** it

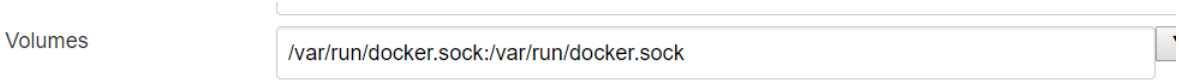


7. For the Docker Image, use `benhall/dind-jenkins-agent:v2`

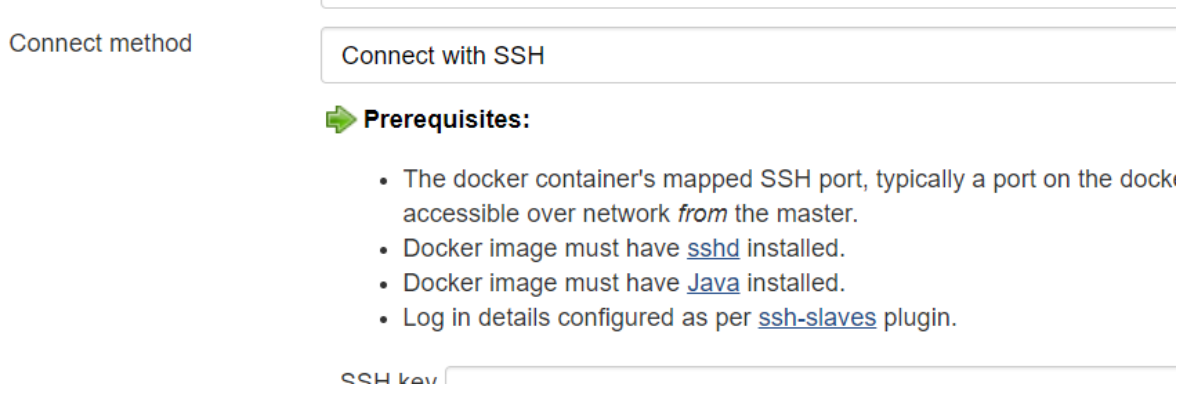
   image will be used as jenkins slave agent and click Enabled



8. Under **Container Settings**, In the "Volumes" text box enter `/var/run/docker.sock:/var/run/docker.sock`. This allows our build container to communicate with the host.



9. For **Connect Method** select **Connect with SSH**. The image is based on the Jenkins SSH Slave image meaning the default Inject SSH key will handle the authentication.



10. Click Save.

11. On the Jenkins dashboard, select **Create new jobs**

12. Give the job a name such as Jenkins Demo, select **Freestyle project** then click OK.
13. The build will depend on having access to Docker. Using the "Restrict where this project can be run" we can define the label we set of our configured Docker agent. The set "Label Expression" to docker-agent. You should have a configuration of "Label is serviced by no nodes and 1 cloud".

https://github.com/prakashk0301/docker-jenkins-demo

14. We can now add a new build step using the **Add Build Step** dropdown. Select **Execute Shell**.
15. Because the logical of how to build is specified in our Dockerfile, Jenkins only needs to call build and specify a friendly name.

In demo, use the following commands.

```
ls
docker info
docker build -t prakash/jenkins-demo:${BUILD_NUMBER} .
docker tag prakash/jenkins-demo:${BUILD_NUMBER} prakash/jenkins-demo:latest
docker images
docker run -d -p 12454:80 prakash/jenkins-demo:latest
```

save and build:

You should see a build scheduled with a message "(pending—Waiting for next available executor)".

In the background, Jenkins is launching the container and connecting to it via SSH. Sometimes this can take a while to configure the Docker Agent. The error "(pending—Jenkins doesn't have label docker-agent)" is while Jenkins waits for the Docker Agent to start.

Login to your jenkins Instance and verify docker images and container.

Once build done you can access sample http <VM>:12454

(If you re-run make sure you change port)