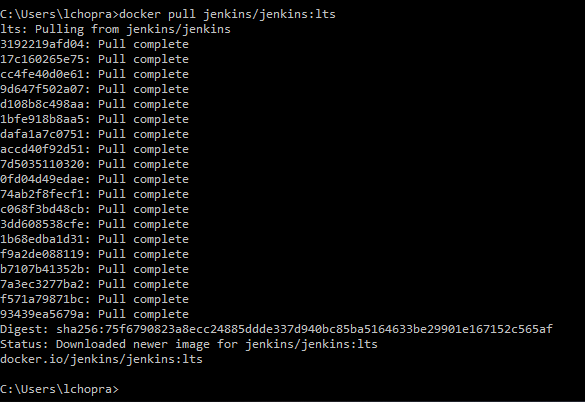
**1. Jenkins (CICD Platform) Installation on Docker Containers**

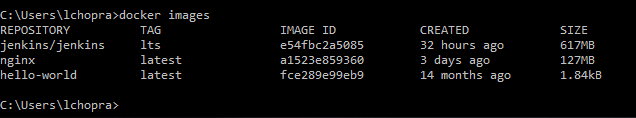
1. Pull the Jenkins docker image from docker hub

> **docker pull jenkins/jenkins:lts**



List the docker images

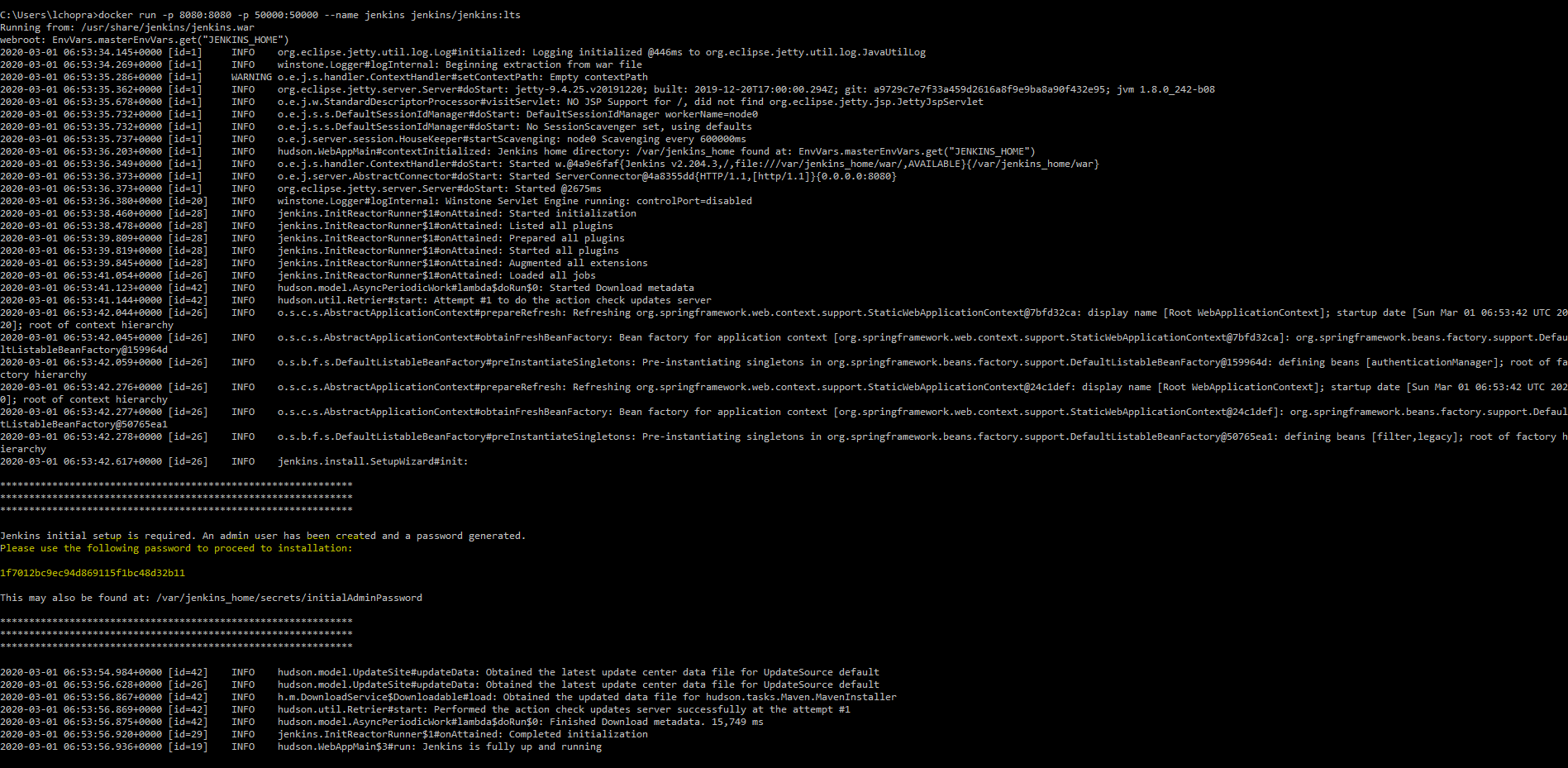
> **docker images**



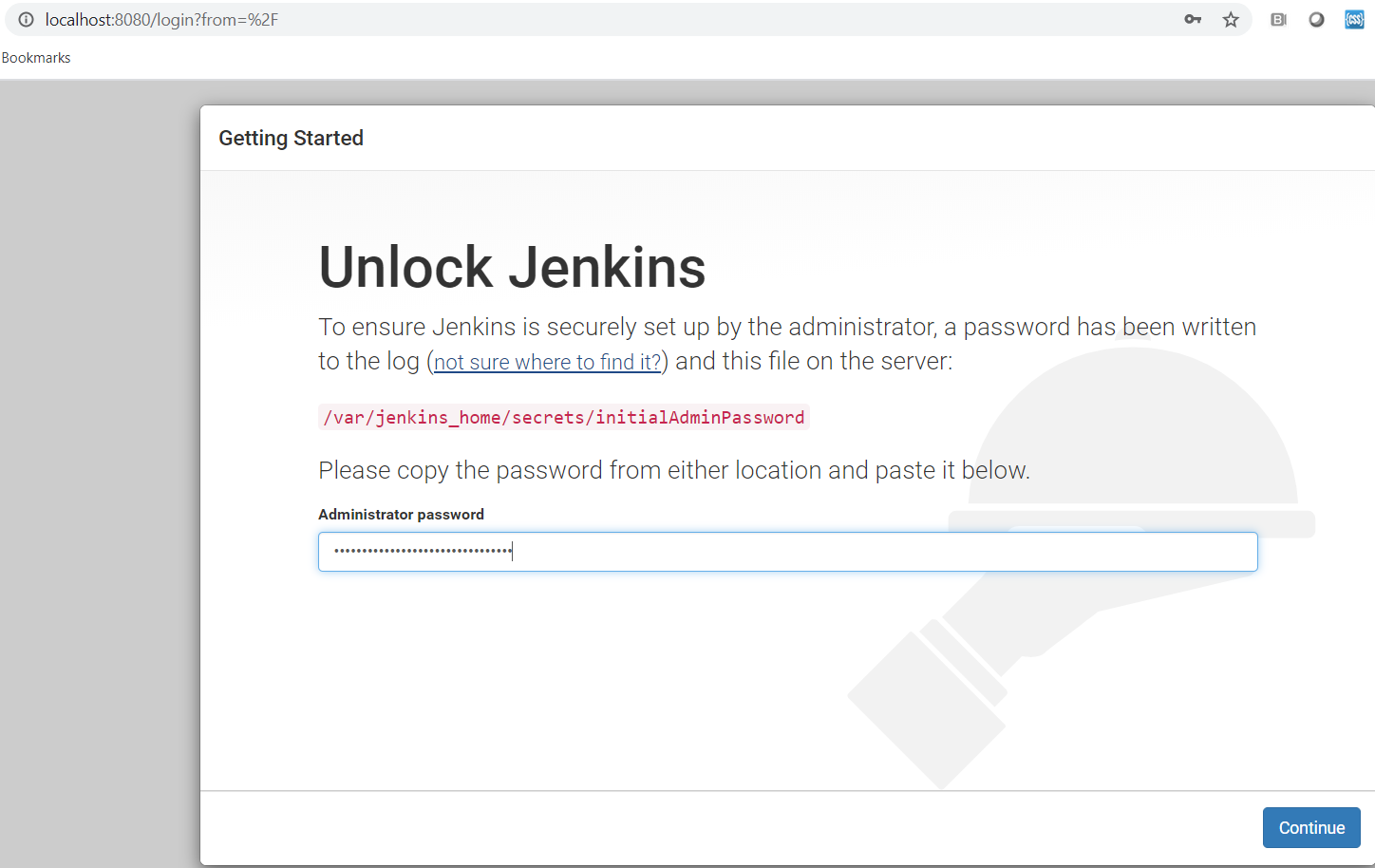
1. Run a Dockerized jenkins instance that we name, jenkins.

> **docker run -p 8080:8080 -p 50000:50000 --name jenkins jenkins/jenkins:lts**

docker run -d -v /usr/jenkins\_home:/var/jenkins\_home -p 8080:8080 -p 50000:50000 jenkins/jenkins

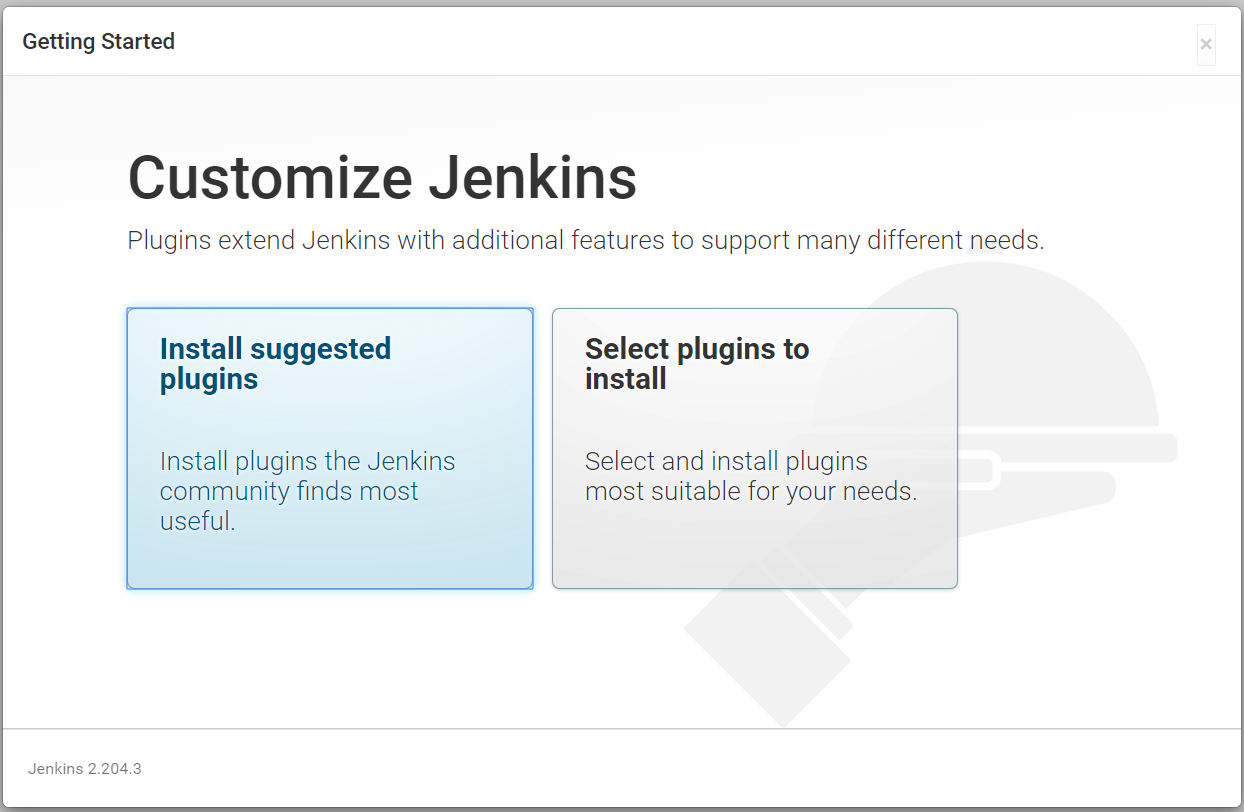


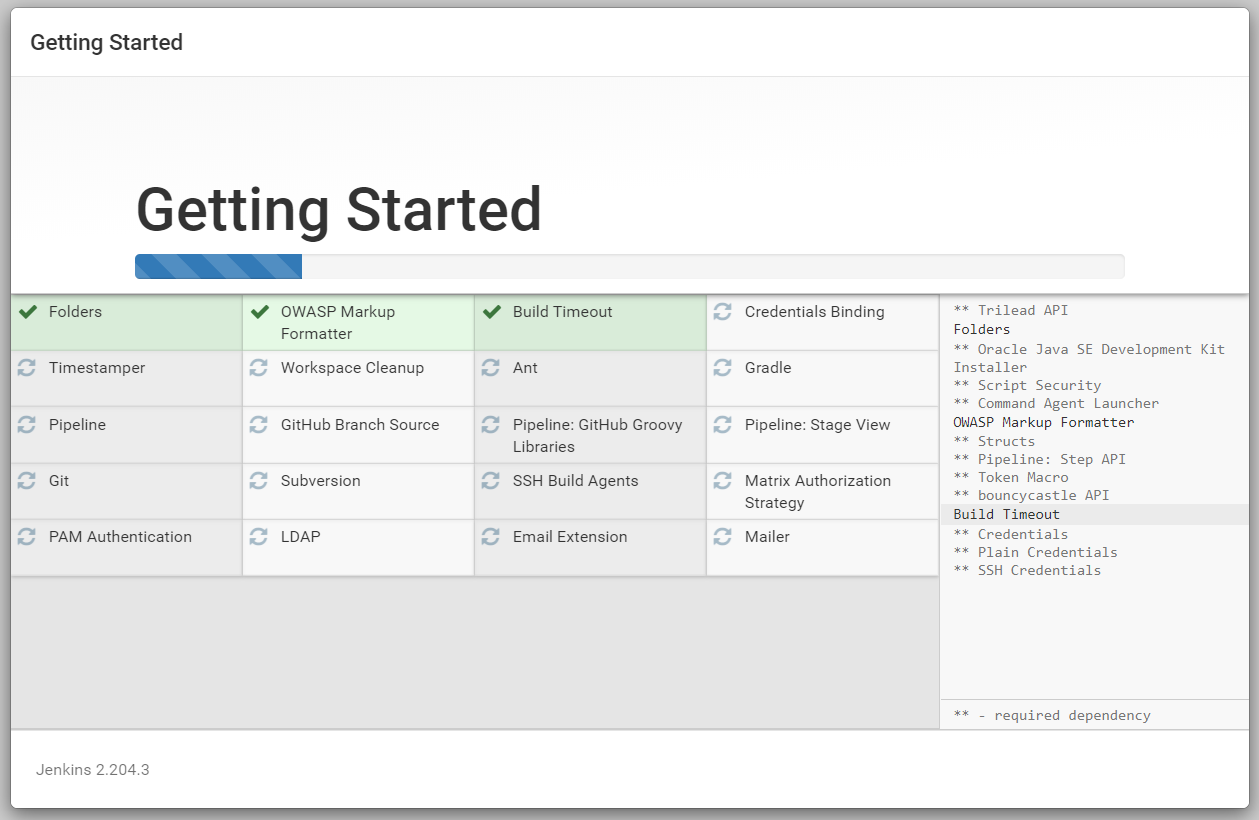
1. Point your web browser at http://localhost:8080 to display the jenkins start page.

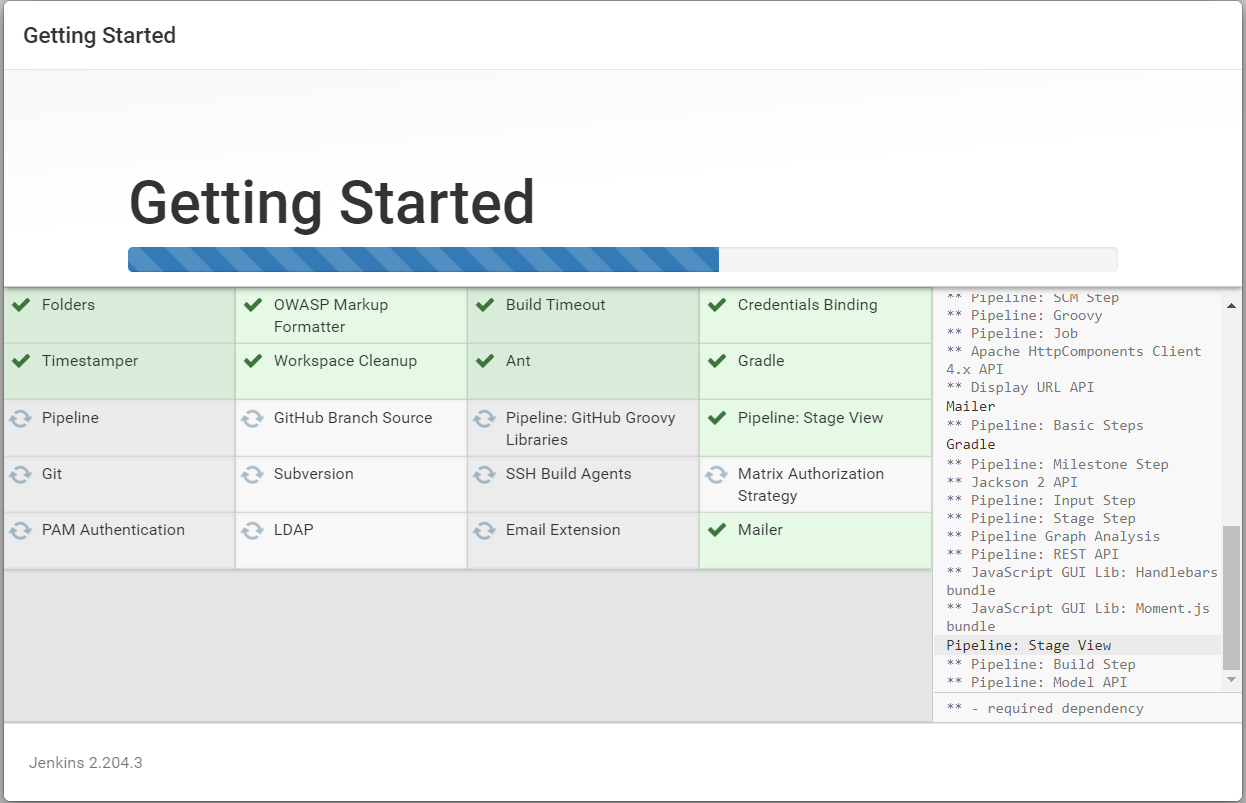


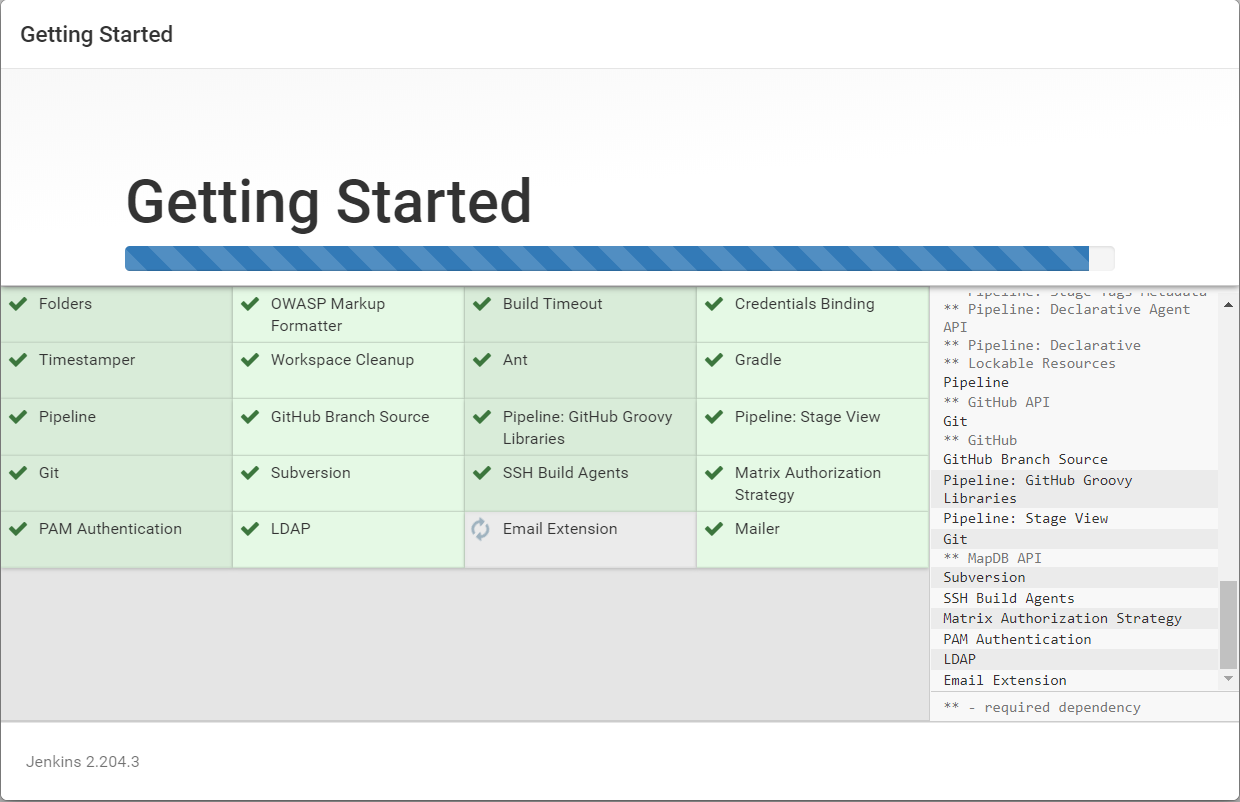
1. Enter the administrator password as in Step#2 above. Click Continue and proceed with the Setup Wizard.

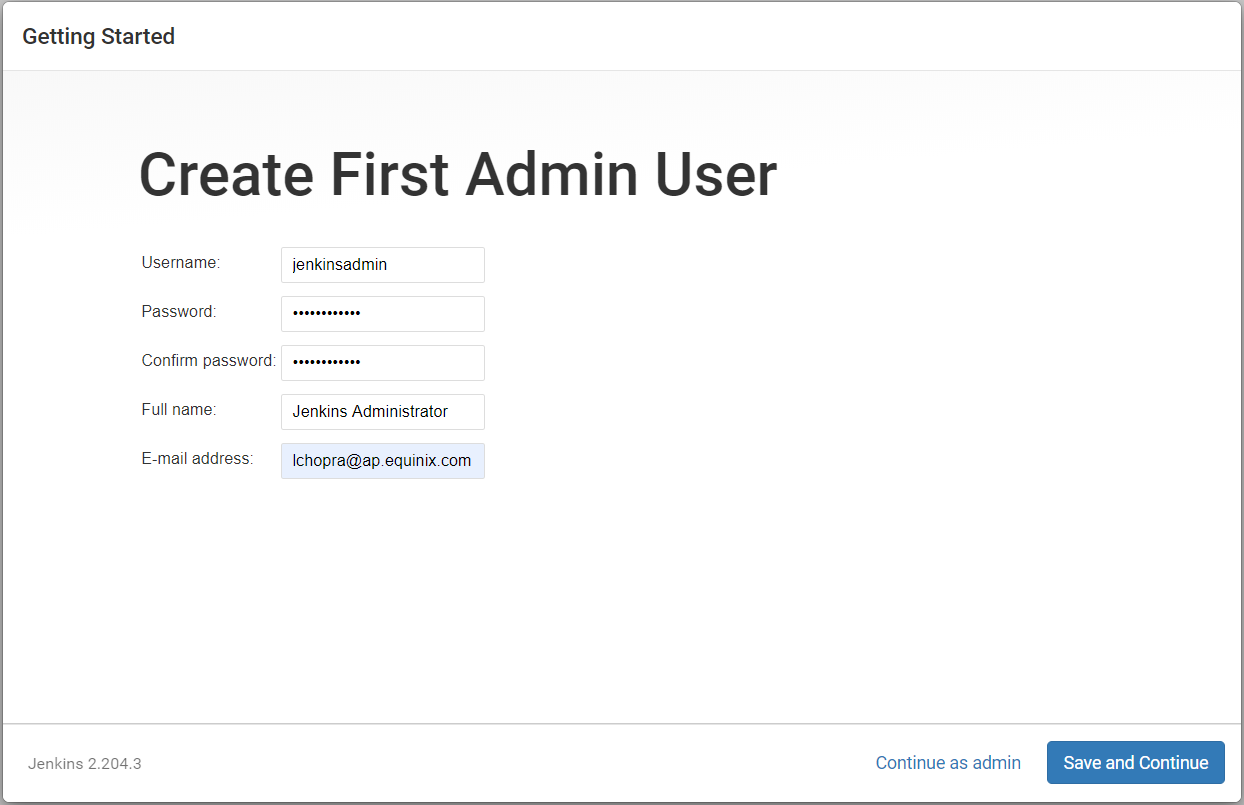
Click on “Install Suggested Plugins”

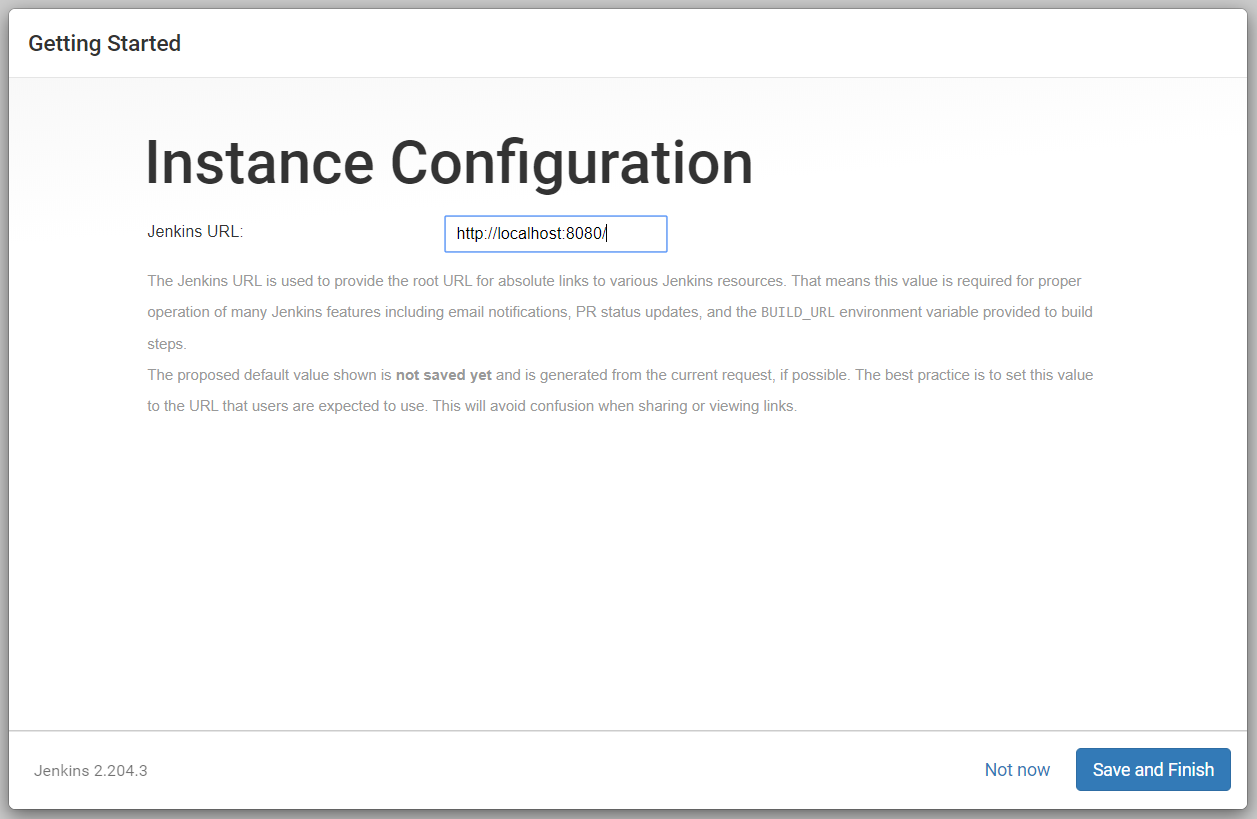


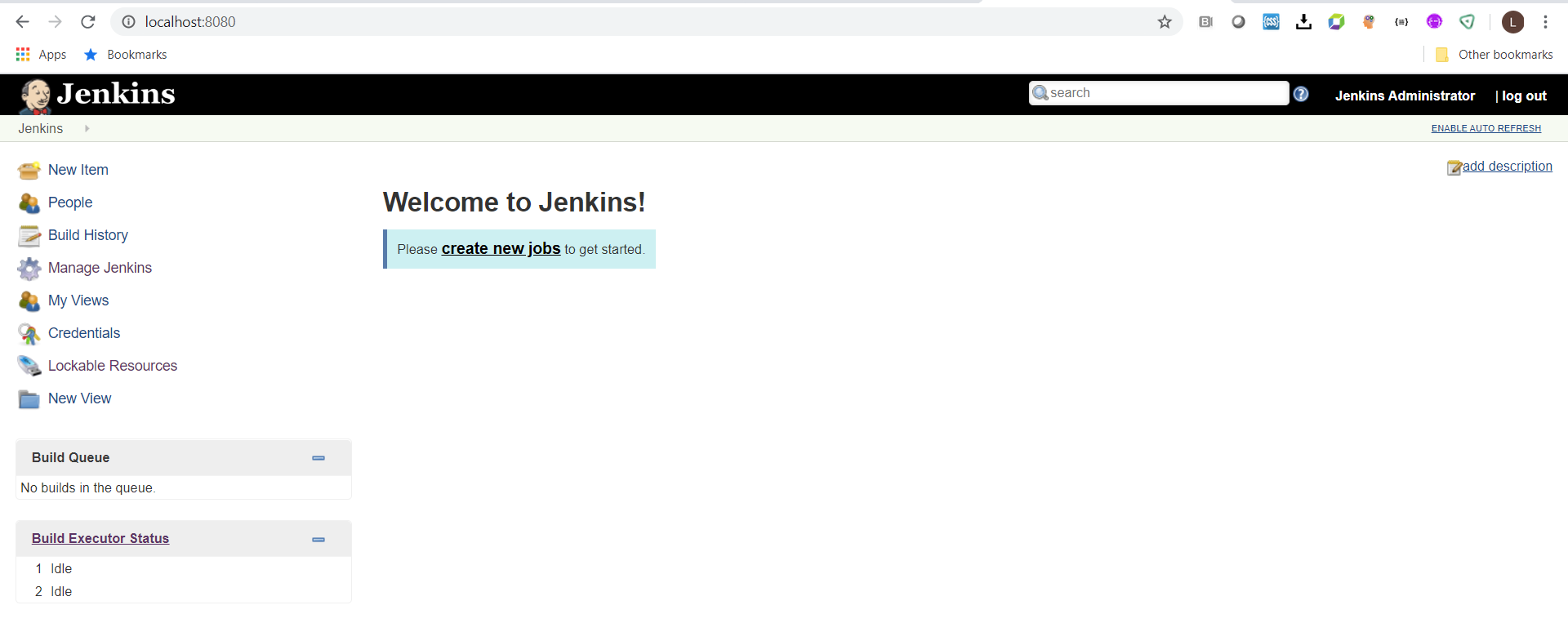








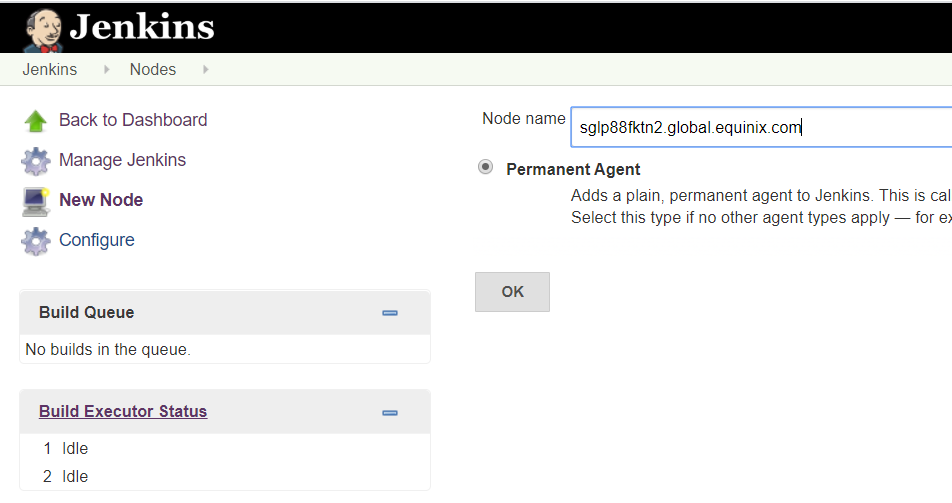


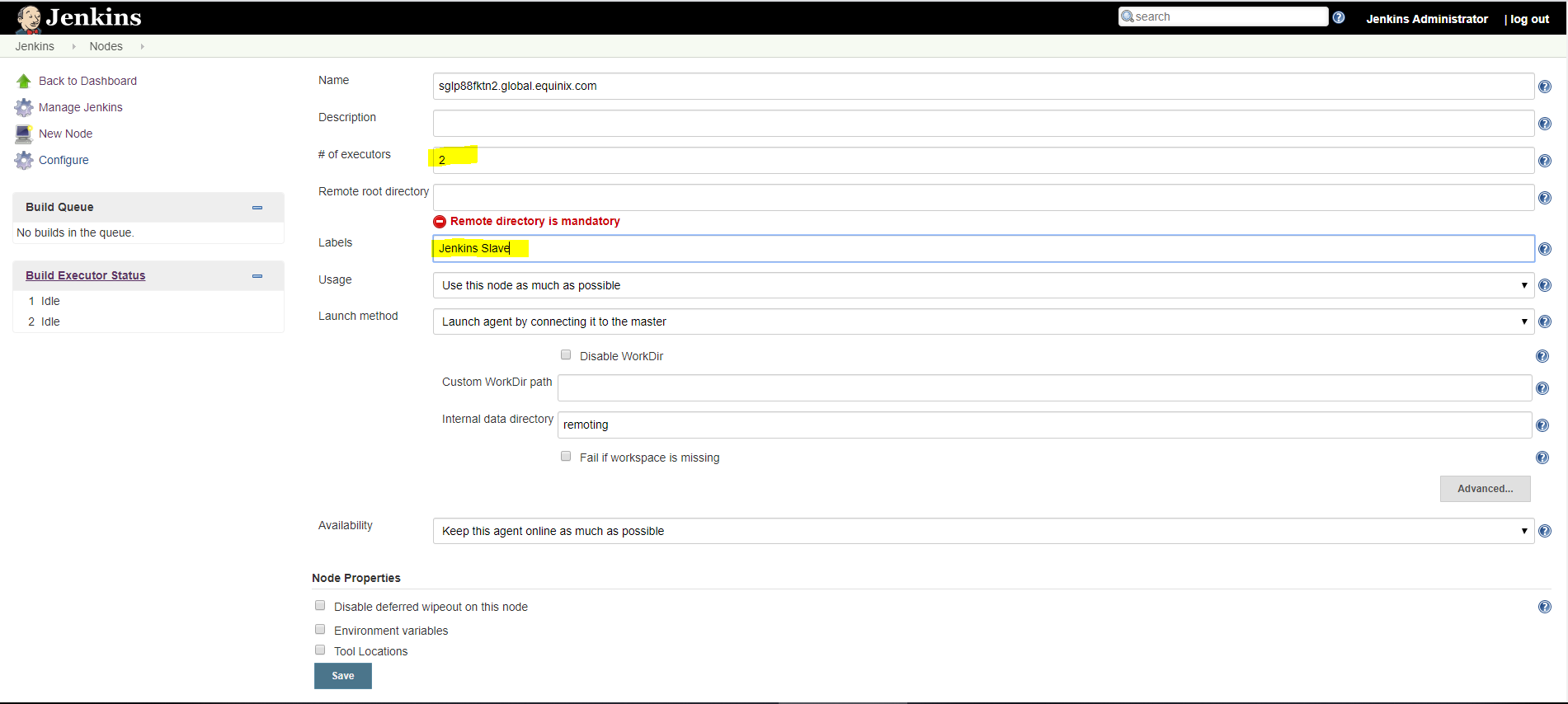


1. Add Jenkins Slave to run jobs.

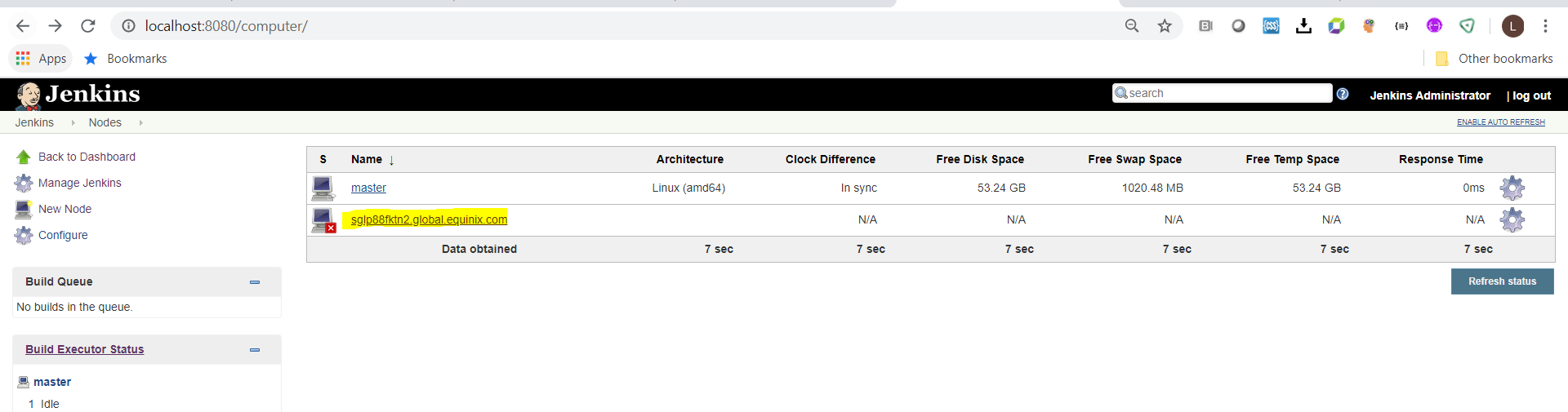
Manage Jenkins > Manage nodes > New Node

Add the node name as your windows machine name.

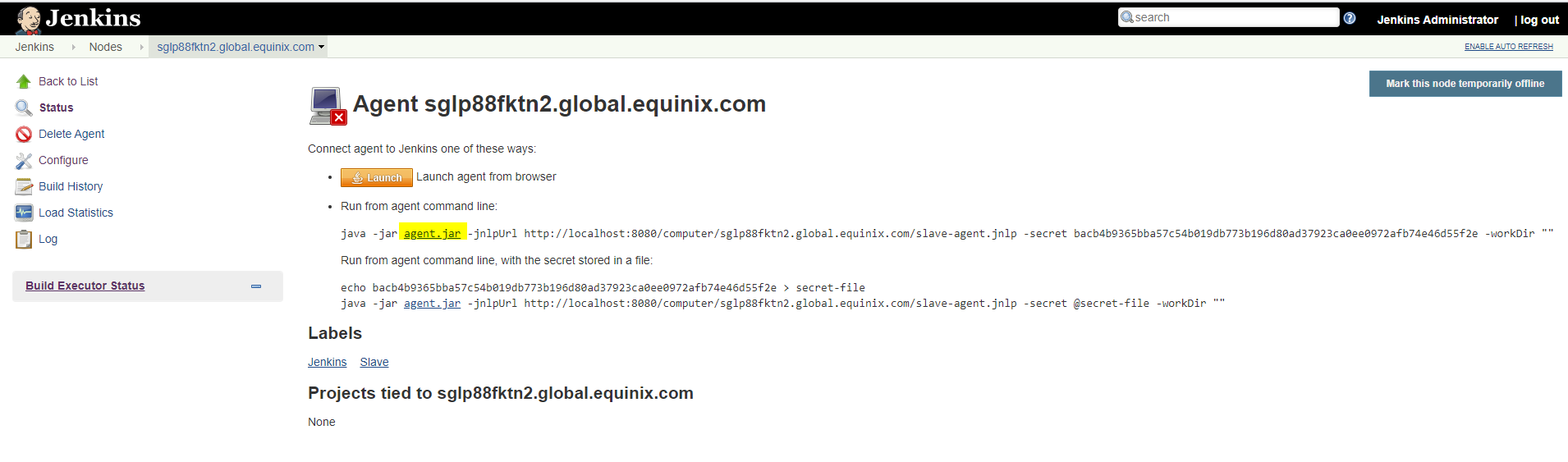




Click Save



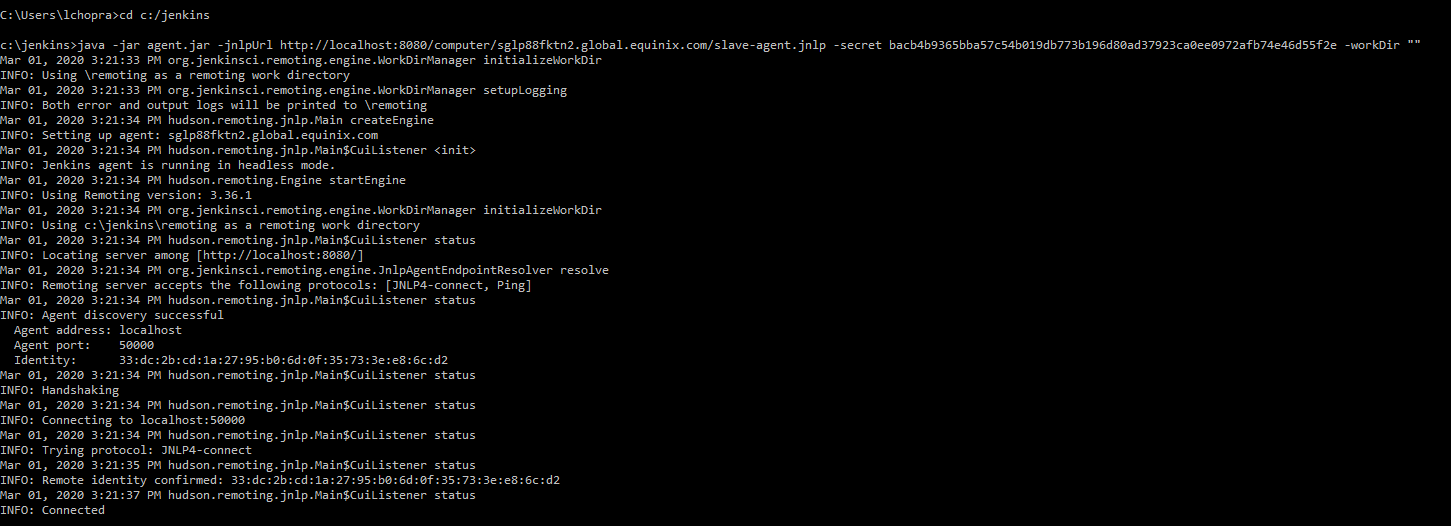
Click on the windows name as highlighted



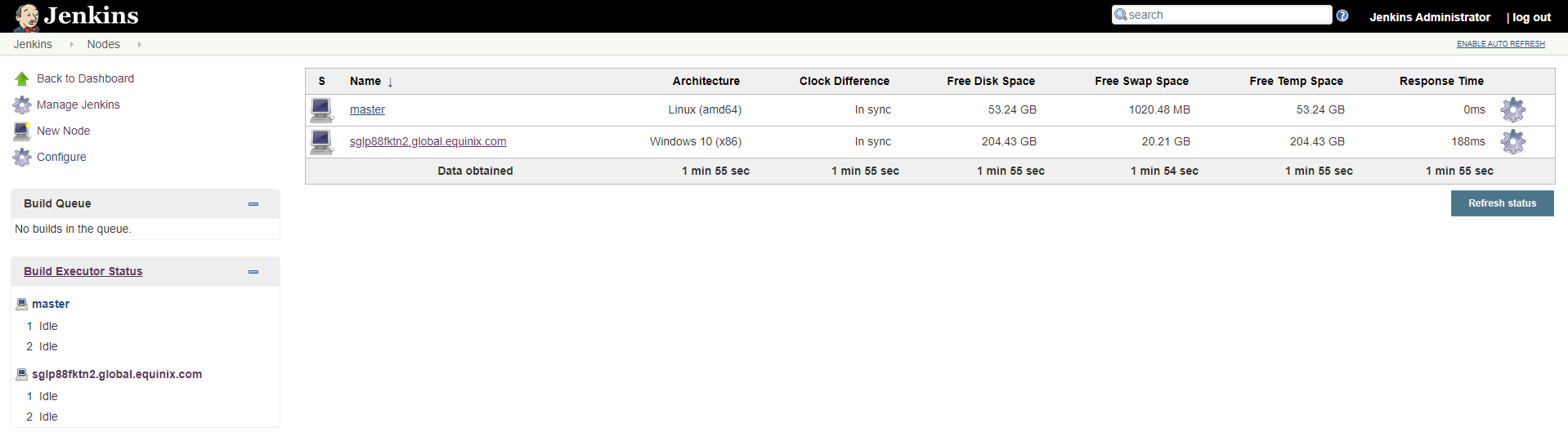
Download the agent.jar as highlighted above and copy to your windows drive e.g. c:\jenkins

On the command prompt, go to c:\jenkins and run the command under “Run from agent command line” as in previous step.

> **java -jar agent.jar -jnlpUrl http://localhost:8080/computer/sglp88fktn2.global.equinix.com/slave-agent.jnlp -secret bacb4b9365bba57c54b019db773b196d80ad37923ca0ee0972afb74e46d55f2e -workDir ""**



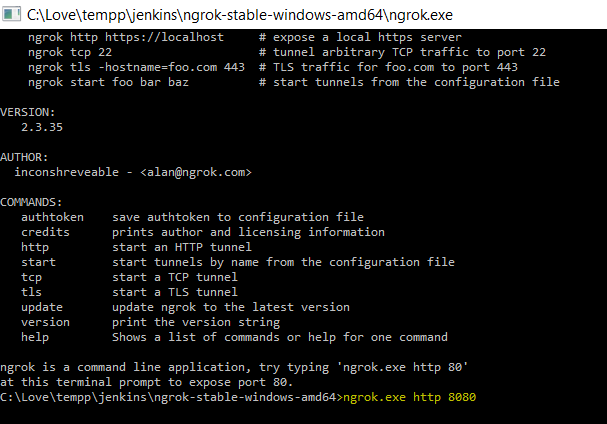
Jenkins slave is now attached to the Jenkins master instance

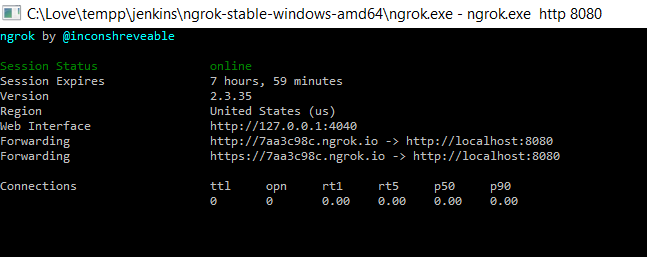


1. Enabling Jenkins Port forwarding via ngrok

Click on the ngrok executable

> **ngrok.exe http 8080 (Jenkins port)**

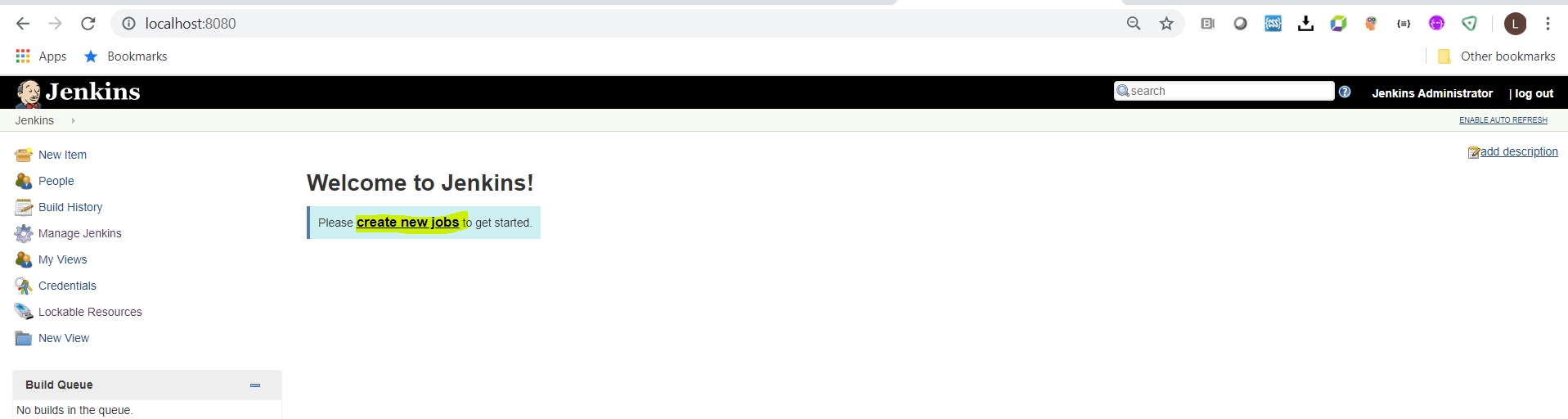


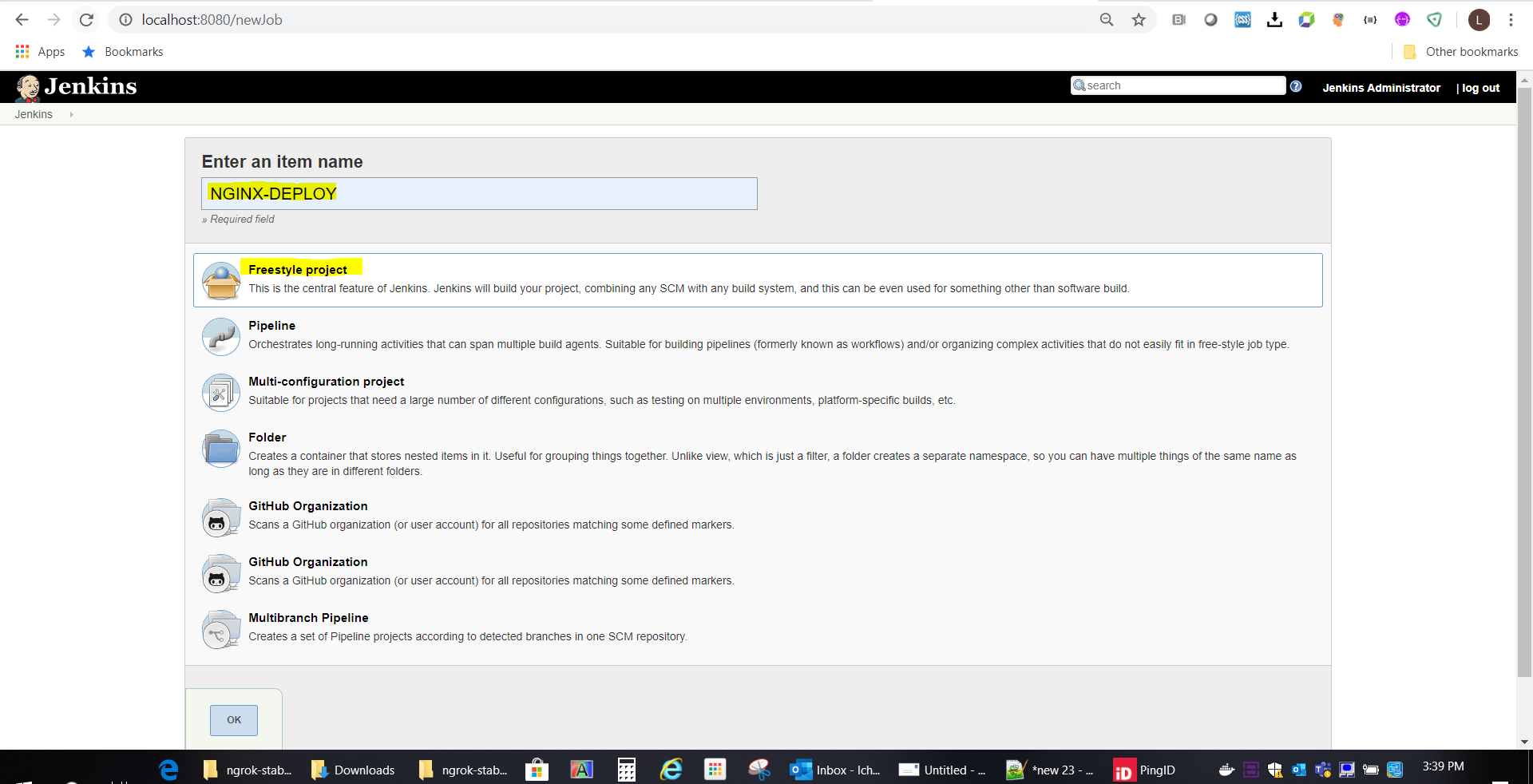


1. On command prompt, execute the below command

> **git config --global http.sslVerify false**

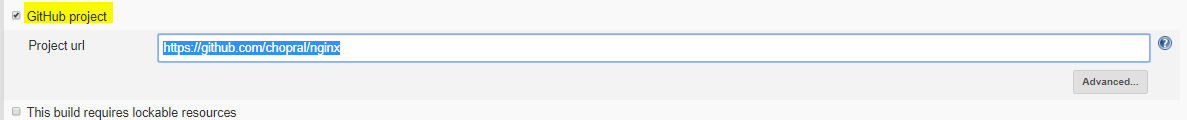
1. **Creating your First Jenkins job to deploy nginx webserver on Github commit**
2. Create free style new job and name it as “NGNIX-DEPLOY”





1. Fork the demo github project

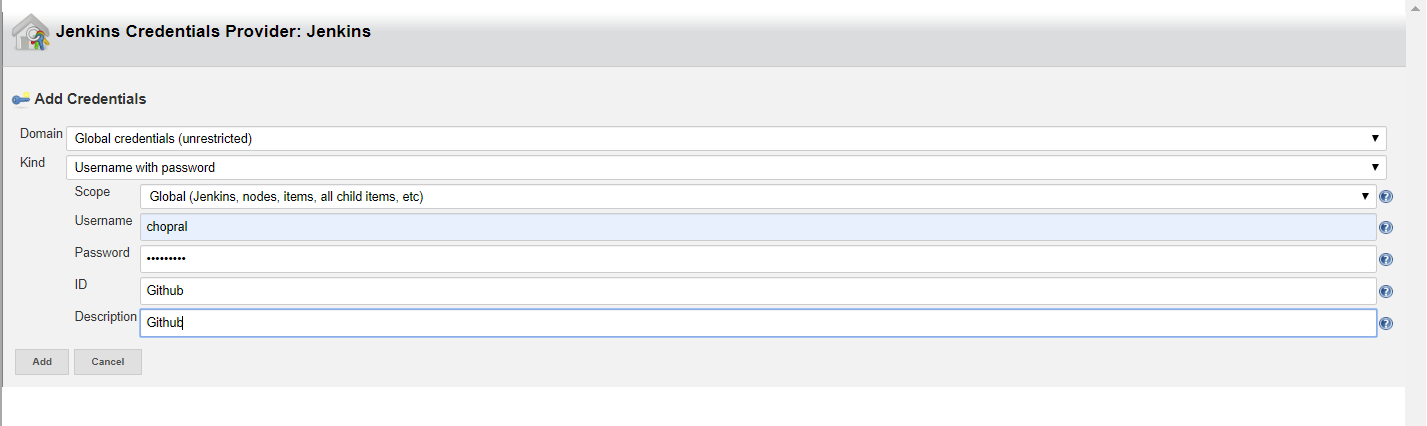
Project url : https://github.com/equinix/devops-demo/



1. Under source code management, select GIT

Repository url : https://github.com/<your\_name>/devops-demo.git

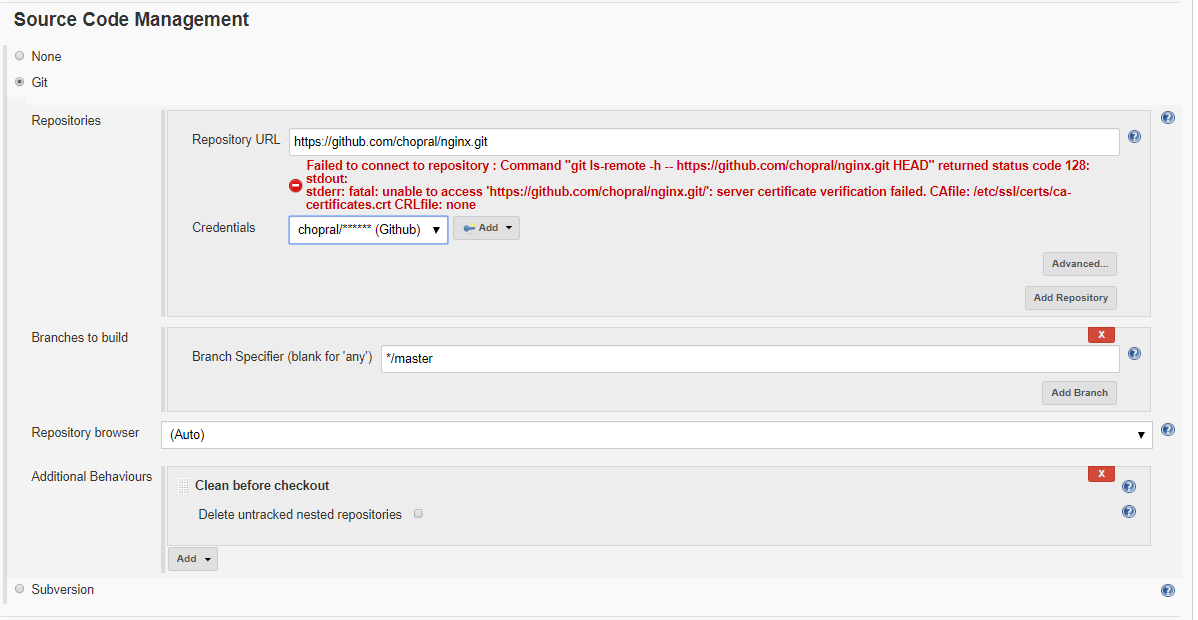
Optionally you can set Credentials > Add > Jenkins credential provider



Credentials : github (one created in above step)

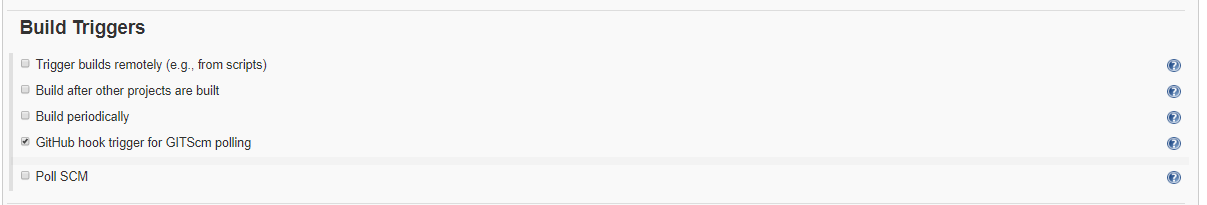
Branch : master

Additional behavior: clean before checkout

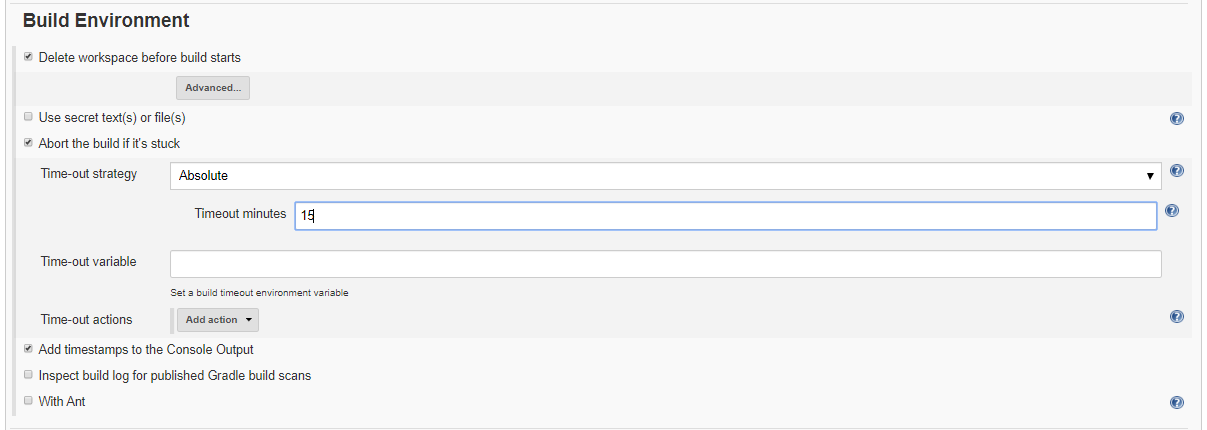


1. Under Build triggers,

Check GitHub hook trigger for GITScm polling



1. Under Build environment,



1. Under Build, Add Build Steps > execute windows batch command

Add the below commands and save

echo "Build Started"

# to Stop the Existing running Container

docker stop nginx\_web\_server

# to Remove and clean the Existing Image

docker rmi -f nginx\_web\_server

pwd

ls -al

# CD into the project Folder

cd nginx\_demo

# To Build the Docker image

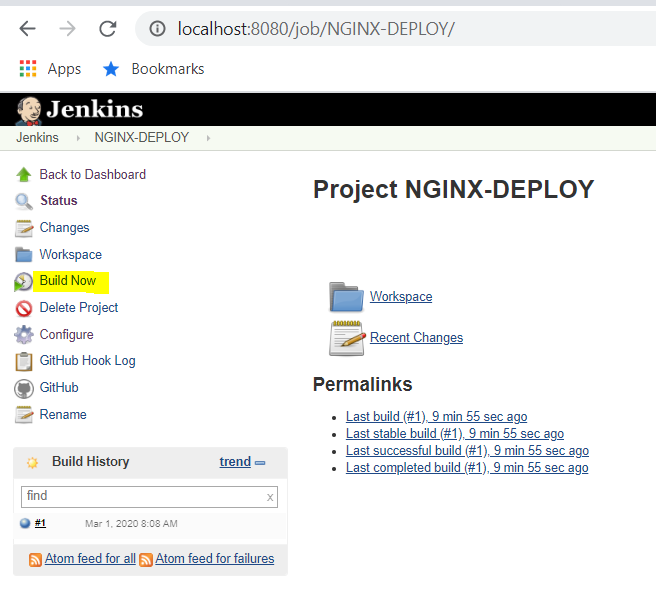
docker build --rm -f "dockerfile" -t nginx\_web\_server .

#To Run the Docker Image

# -d or --detach -p or --publish

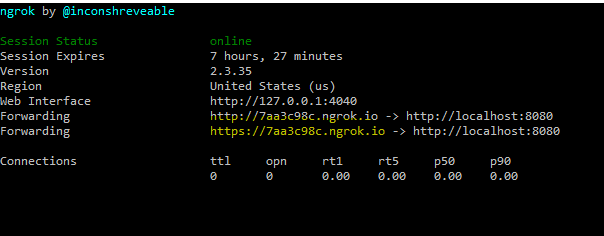
docker run -d --name nginx\_web\_server -p 80:80 -p 443:443 nginx\_web\_server

1. Your first job is ready to be executed via Jenkins (Click on Build Now)

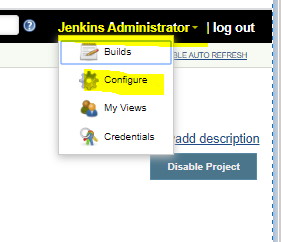


1. Let us integrate it to be executed when the code is checked into Github repo.

* Go to Github repo
  + https://github.com/<your\_neme>/devops-demo.git
* Click on settings > webhook
* Add webhook
* Payload url :  [https://7aa3c98c.ngrok.io/github-webhook/](https://0a254eb6.ngrok.io/github-webhook/)

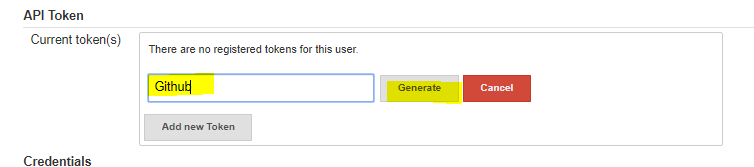


* Content type: application/json
* Secret : Generate one from Jenkins UI

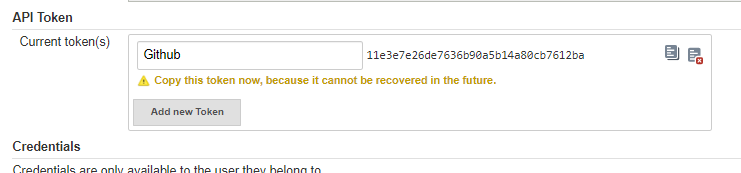


Add New token

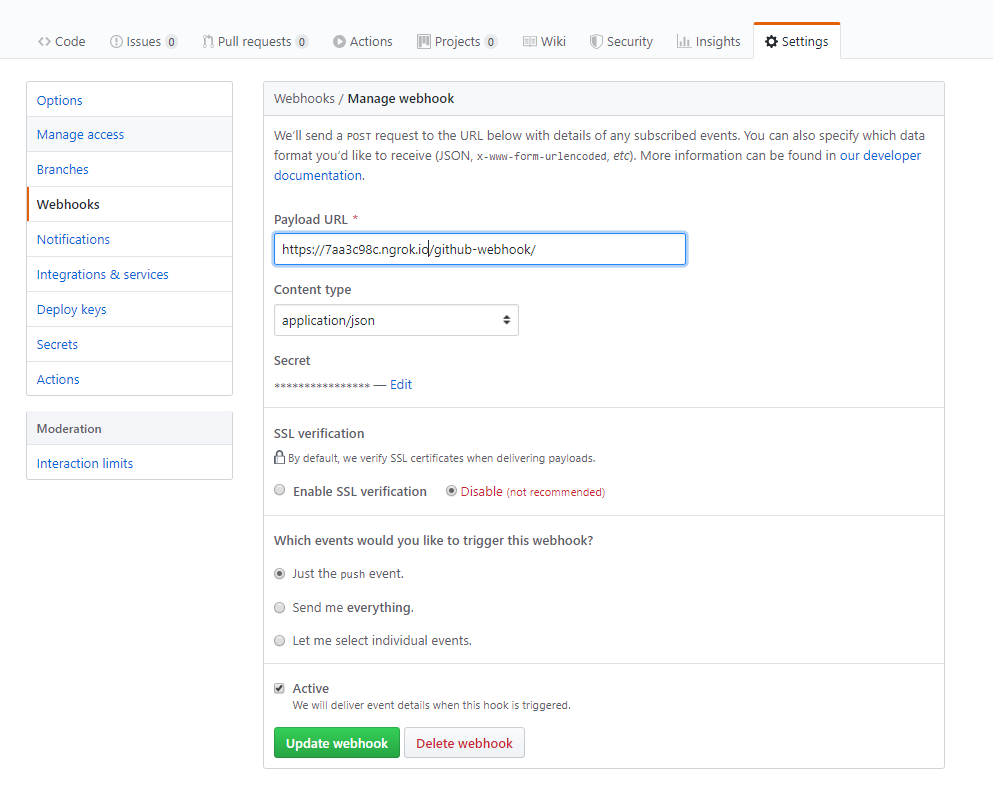




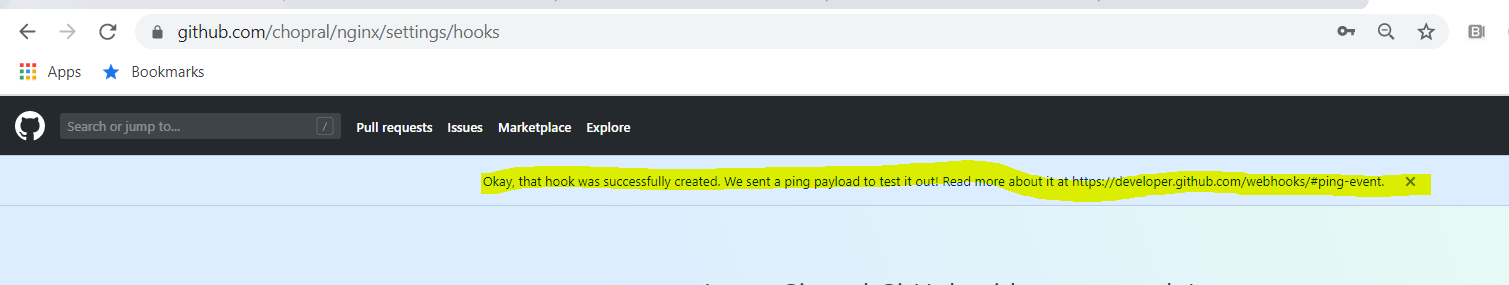
Click Generate and copy the token



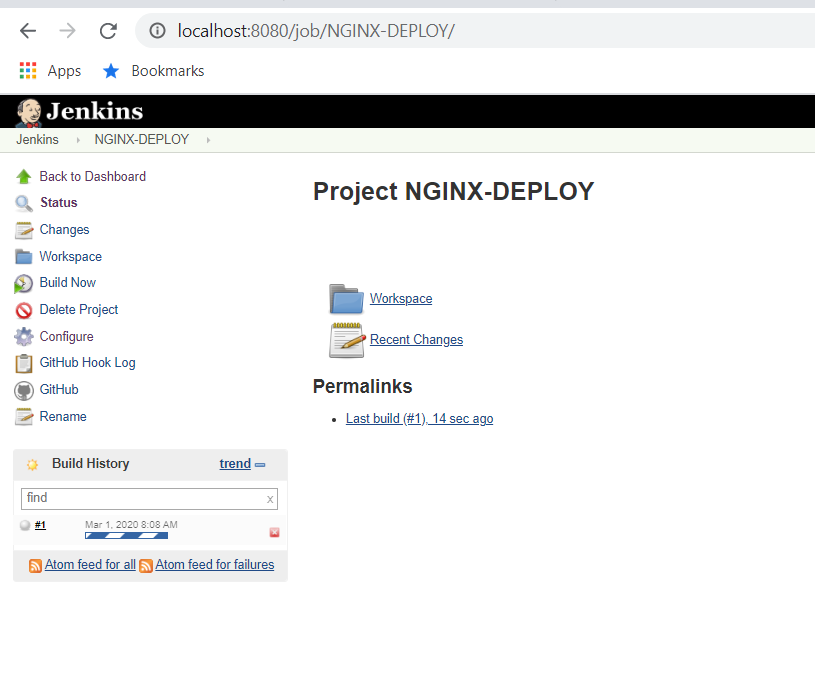
* Under Events, check just the push event
* Check Active
* Add Webbook

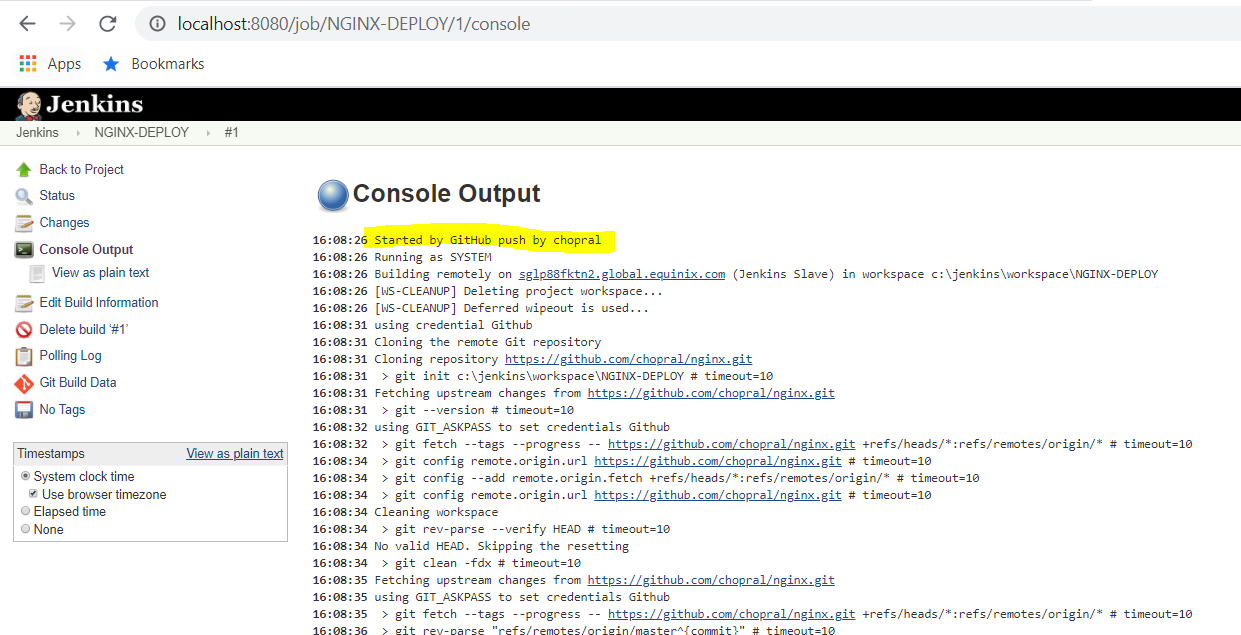


* You will get below screen to confirm the webhook activation



* Update the code in Github repo, which will trigger the Jenkins build automatically.





* You should now be able to access nginx on your windows machine.

<http://localhost/>