Jenkins lab 2

Created	@July 23, 2022 9:06 PM
Class	DevOps
Materials	Dockerfile lab2.txt
Type	Lab
Reviewed	☑
Date	@July 23, 2022
By	Ahmed Mohamed Ahmed Rizk

Q1: configure Jenkins image to run docker commands on your host docker daemon

- what we want to install docker inside Jenkins, we will install docker client and then we will make our Jenkins container user our local ”my pc” docker demon via docker volume

- Create Dockerfile for this task

```
FROM jenkins/jenkins:lts
USER root
# Install docker client
RUN apt-get update -qq
# Install dependencies
RUN apt-get install -qq apt-transport-https ca-certificates curl gnupg2 software-properties-common
# Add Docker’s GPG Key (remember it is debian not linux ) / don't set it as sudo !
RUN curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -
# Install the Docker Repository "remember it is debian not linux)"
RUN add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/debian \
    $(lsb_release -cs) \
    stable"
# Update Repositories  &  Install Latest Version of Docker
RUN apt-get update -qq \
    && apt-get install docker-ce -y
# Add the user jenkins to the group docker on the system
RUN usermod -aG docker jenkins
```

- We build a new image based on it

```
docker build -t jenkins-with-docker .
```

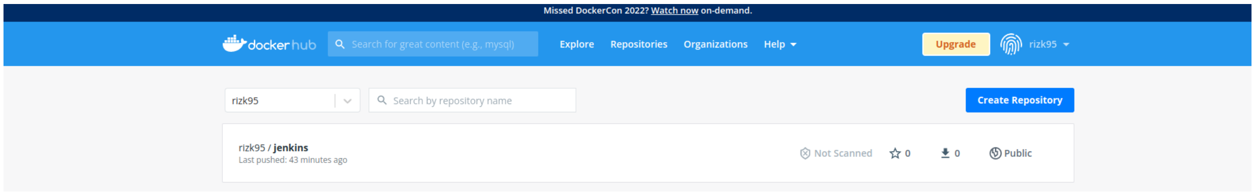
- Run a container based on this image **(set volume of my local docker daemon to reflect on Jenkins docker daemon)**

```
docker run -d -p 8089:8080 -v /var/run/docker.sock:/var/run/docker.sock --name JenkinsWithDocker jenkins-with-docker
```

Q2: create CI/CD for this repo https://github.com/mahmoud254/jenkins_nodejs_example.git

First: you have to create a repo inside dockerhub.com

- log-in inside dockerhub “ we will need username and password for your credientials”
- create public repo “Jenkins”



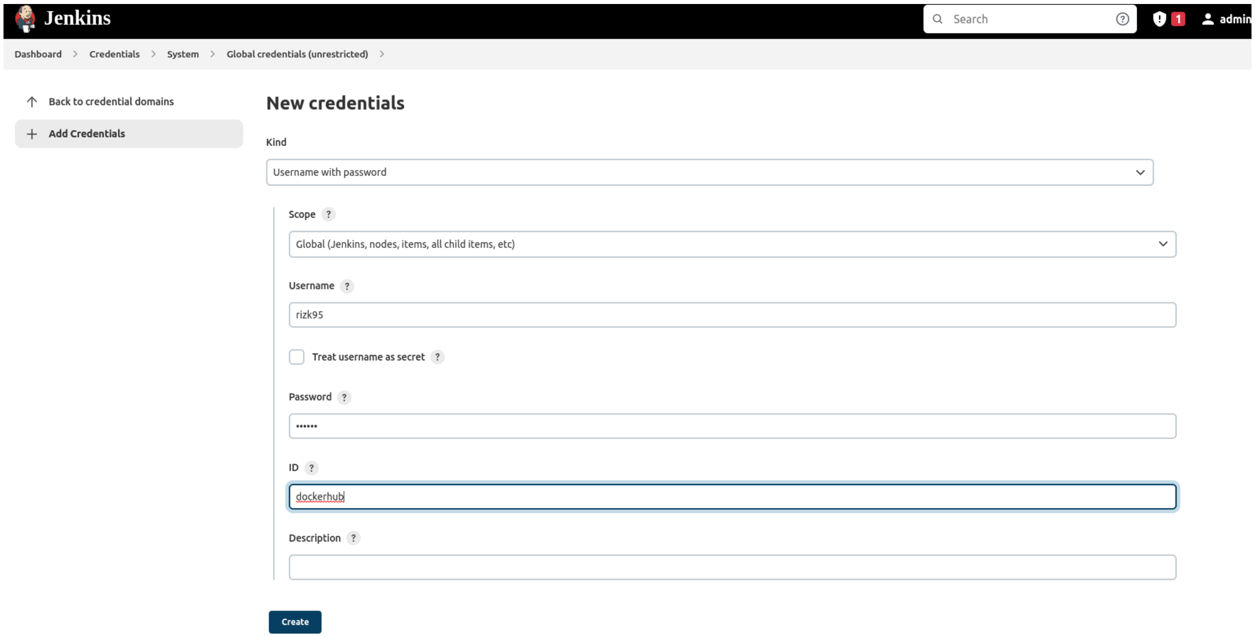
Missed DockerCon 2022? Watch now on-demand.

rizk95 / jenkins
Last pushed: 43 minutes ago

Not Scanned | 0 | 0 | Public

Second: based on Q1, we will use a Jenkins with docker installed and its daemon pointed to my local docker daemon

- set my dockerhub credentials inside Jenkins using Manage Credentials
- set username, password and ID in order to use it in the stage



Jenkins
Search
1
admin
Dashboard > Credentials > System > Global credentials (unrestricted) >
Back to credential domains
Add Credentials

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: rizk95

☐ Treat username as secret

Password: ••••••

ID: dockerhub

Description:

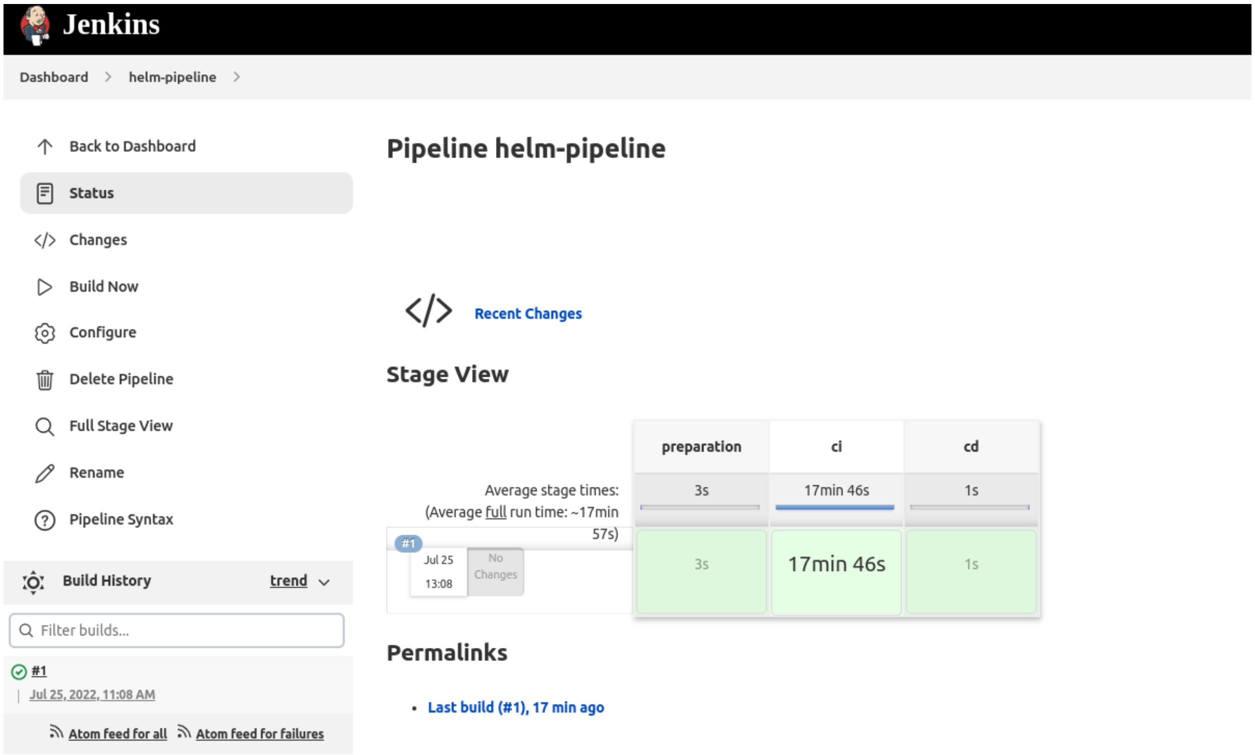
Third: Create a pipeline for the application, set the code and build it

- pipeline code
- you will use **withCredentials** plugin in order to login to dockerhub and push the image
- we have 3 stages, 1- preparation: to git the code, 2- CI: build a docker image and login and push this image to my dockerhub repo, 3- CD: run the container

```
pipeline {
    agent any

    stages {

        stage('preparation') {
            steps {
                // Get some code from GitHub repository
                git 'https://github.com/mahmoud254/jenkins_nodejs_example.git'
            }
        }

        stage('ci') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'dockerhub', passwordVariable: 'PASSWORD', usernameVariable: 'USERNAME')])
                    sh """
                    docker build . -f dockerfile -t rizk95/jenkins
                    docker login -u ${USERNAME} -p ${PASSWORD}
                    docker push rizk95/jenkins
                    """
                }
            }
        }

        stage('cd') {
            steps {
                sh "docker run -d -p 3000:3000 rizk95/jenkins"
            }
        }
    }
}
```

Finally Build your pipeline (It will take a lot of time to push to your dockerhub as per your internet speed connection but it will work at the end)

RESULTS

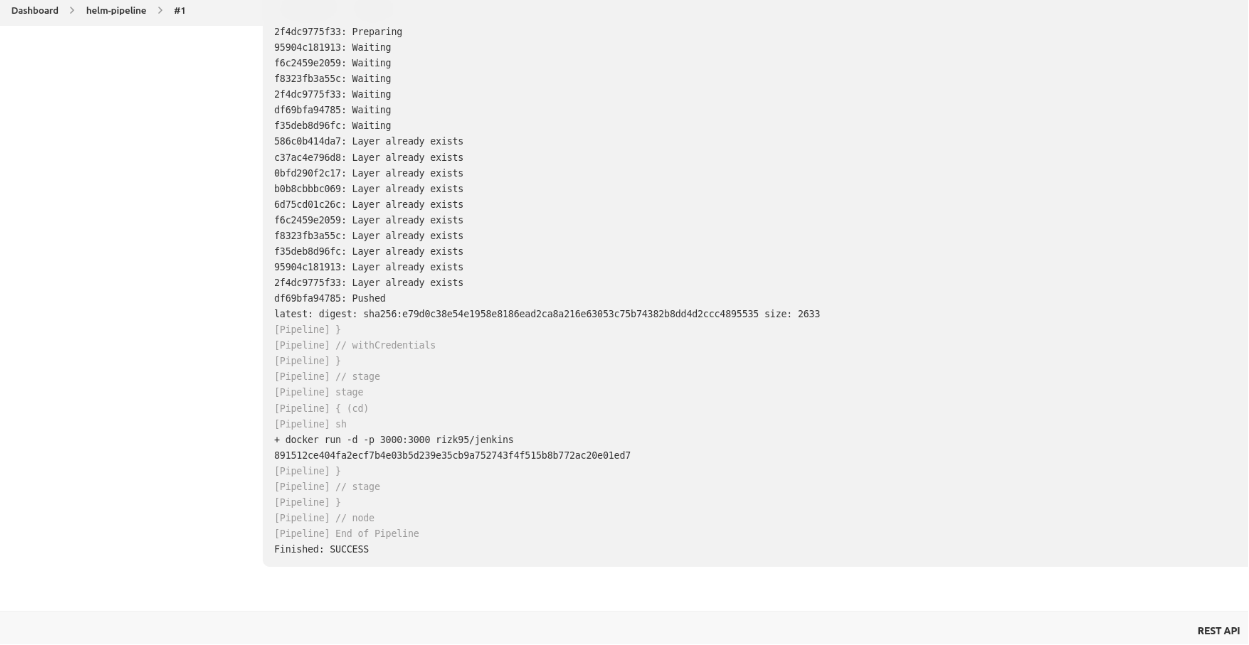


Pipeline helm-pipeline

Stage View

	preparation	ci	cd
Average stage times: (Average full run time: ~17min 57s)	3s	17min 46s	1s
#1 Jul 25 13:08 No Changes	3s	17min 46s	1s

Permalinks



```
2f4dc9775f33: Preparing
95904c181913: Waiting
f6c2459e2059: Waiting
f8323fb3a55c: Waiting
2f4dc9775f33: Waiting
df69bfa94785: Waiting
f35deb8d96fc: Waiting
586c0b414da7: Layer already exists
c37ac4e796d8: Layer already exists
0bfd290f2c17: Layer already exists
b0b8cbbbc069: Layer already exists
6d75cd01c26c: Layer already exists
f6c2459e2059: Layer already exists
f8323fb3a55c: Layer already exists
f35deb8d96fc: Layer already exists
95904c181913: Layer already exists
2f4dc9775f33: Layer already exists
df69bfa94785: Pushed
latest: digest: sha256:e79d0c38e54e1958e8186ead2ca8a216e63053c75b74382b8dd4d2ccc4895535 size: 2633
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (cd)
[Pipeline] sh
+ docker run -d -p 3000:3000 rizk95/jenkins
891512ce404fa2ecf7b4e03b5d239e35cb9a752743f4f515b8b772ac20e01ed7
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Q3: create docker file to build image for jenkins slave

- i generated a key called “docker_rsa” for the purpose of the lab and assign and reallocate it in docker container

```
FROM ubuntu
USER root

################ FIRST CONFIGURE IMAGE TO BE SLAVE #################
# create a directory to set all jenkins projects
RUN mkdir -p jenkins_home
RUN chmod 777 jenkins_home
# setup the required packages for creating a jenkins slave:
# openjdk ,openssh: for allowing ssh
RUN apt-get update -qq
RUN apt-get install openjdk-11-jdk -qq
RUN apt-get install openssh-server -qq
# create a user called jenkins
RUN useradd -ms /bin/bash jenkins

################ SETUP DOCKER CLIENT INSIDE  SLAVE #################
# Install docker client
RUN apt-get update -qq
# Install dependencies
RUN apt-get install -qq apt-transport-https ca-certificates curl gnupg2 software-properties-common
# Add Docker’s GPG Key (remember it is ubuntu based ) / don't set it as sudo !
RUN curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
# Install the Docker Repository "remember it is ubuntu based)"
RUN add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
# Update Repositories  &  Install Latest Version of Docker
RUN apt-get update -qq \
    && apt-get install docker-ce -y
# Add the user jenkins to the group docker on the system
RUN usermod -aG docker jenkins

# Create shh folder in home directory and set the access for it
RUN mkdir -p /home/jenkins/.ssh
COPY docker_rsa.pub /home/jenkins/.ssh/authorized_keys
RUN chown -R jenkins:jenkins /home/jenkins/.ssh
RUN chmod 700 /home/jenkins/.ssh
RUN chmod 644 /home/jenkins/.ssh/authorized_keys
```

```
# log-in as jenkins user
USER jenkins
# cd jenkins_home
WORKDIR /home/jenkins/jenkins_home

EXPOSE 22
# run it
CMD ["/bin/bash"]
```

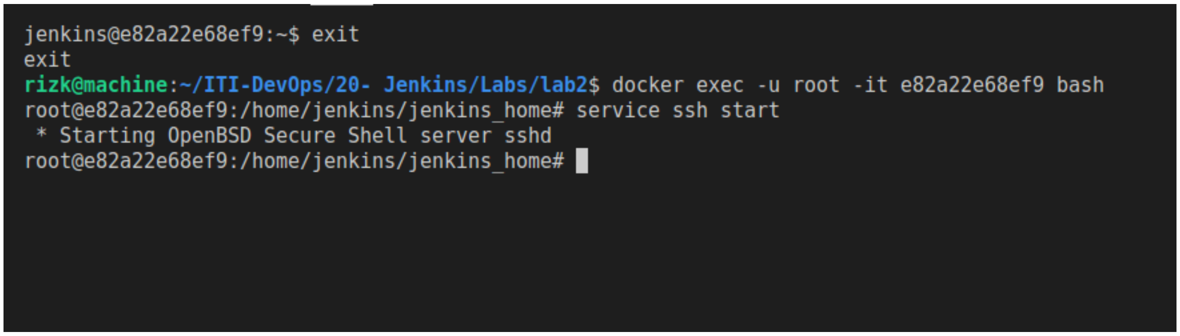
```
docker build . -f slave_dockerfile -t jenkins-slave
```

Q4: create container from this image and configure ssh and from jenkins master create new node with the slave container

- From the previous question
- remember to assign docker demon from your local pc via docker volume in /var/run/docker.sock

```
docker run -d -it -v /var/run/docker.sock:/var/run/docker.sock --name r-slave jenkins-slave
```

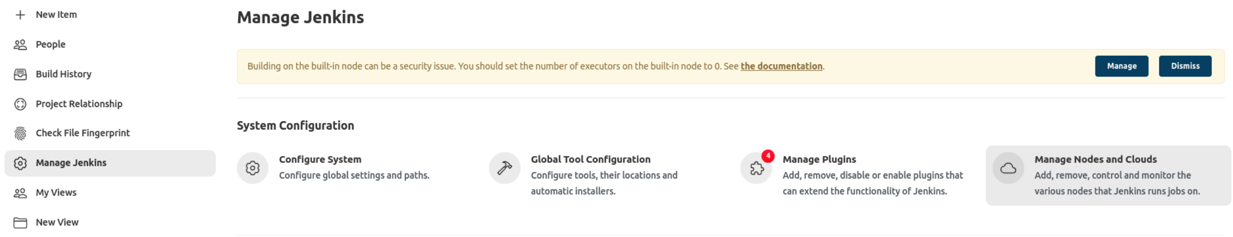
- Very Important (for our dockerfile we use Ubuntu image which its openssh service is not automatically opened) , you have to log in as root user and start the service



```
jenkins@e82a22e68ef9:~$ exit
exit
rizk@machine:~/ITI-DevOps/20- Jenkins/Labs/lab2$ docker exec -u root -it e82a22e68ef9 bash
root@e82a22e68ef9:/home/jenkins/jenkins_home# service ssh start
 * Starting OpenBSD Secure Shell server sshd
root@e82a22e68ef9:/home/jenkins/jenkins_home# 
```

To create New node inside Jenkins

- we first go to **Manage nodes and clouds** → **New Node**



Manage Jenkins

Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See the documentation.

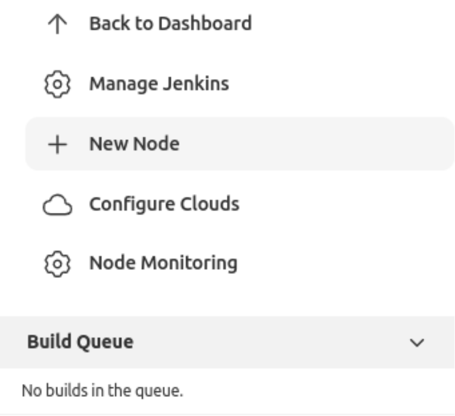
System Configuration

Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

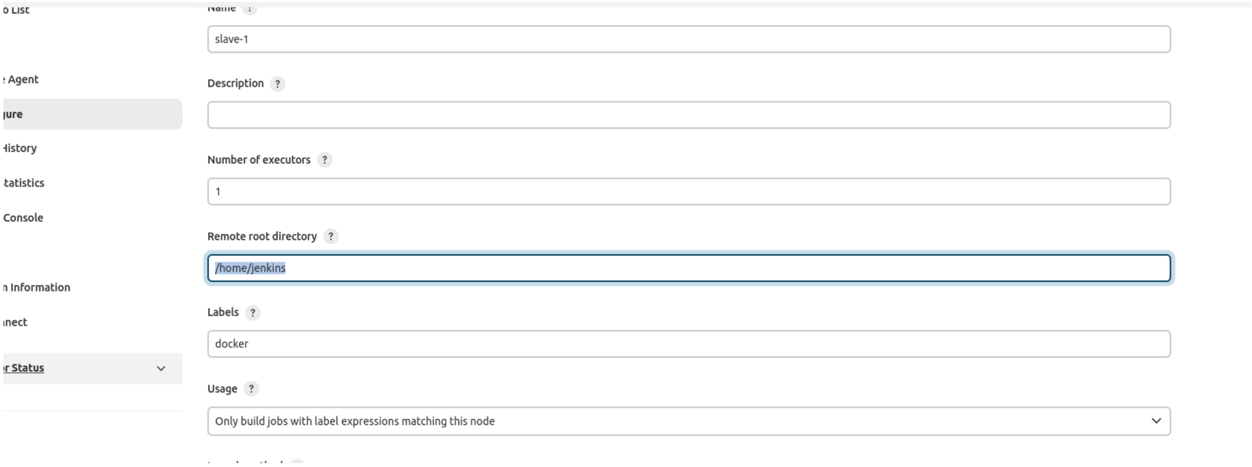
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



No builds in the queue.

- set remote root directory / set labels
- **Usage:** only build



Name: slave-1

Description:

Number of executors: 1

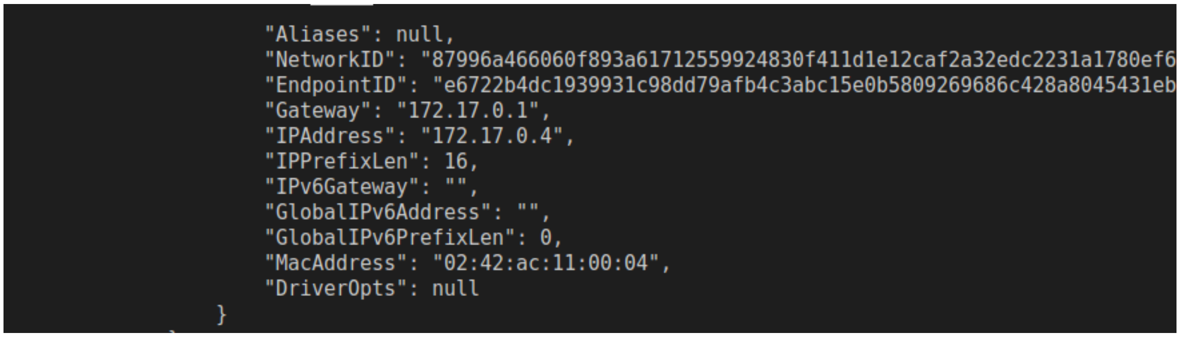
Remote root directory: /home/jenkins

Labels: docker

Usage: Only build jobs with label expressions matching this node

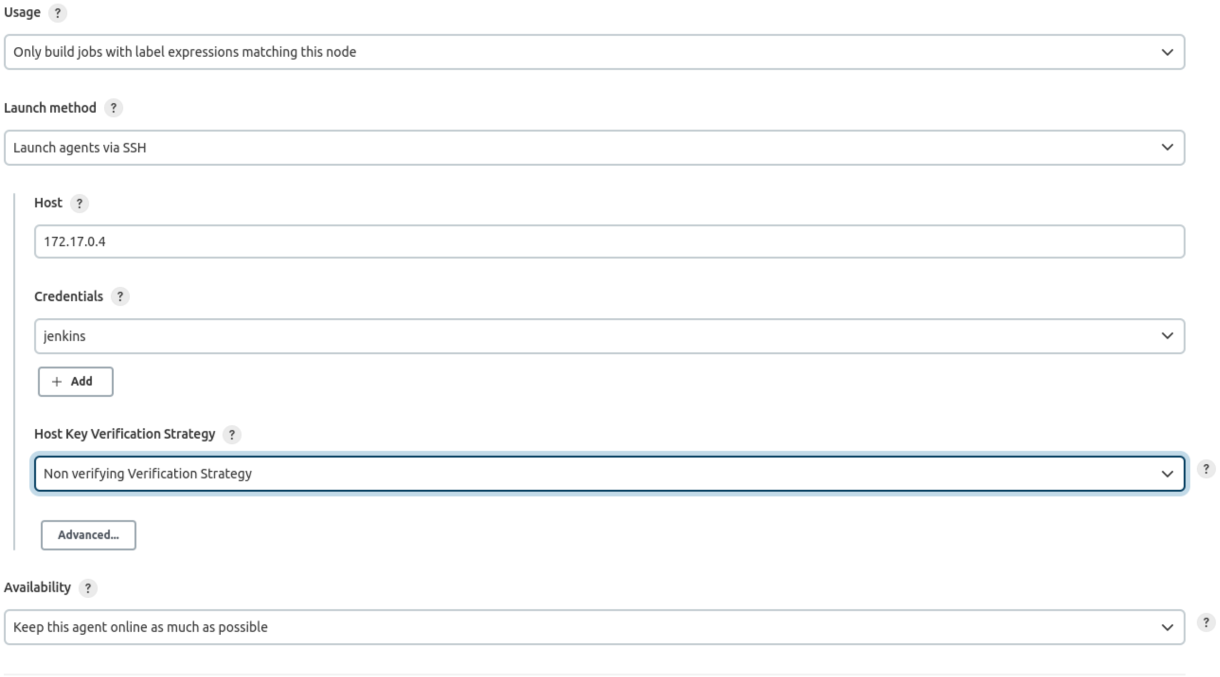
- get container IP Address

```
docker inspect <container ID>
```



```
                    "Aliases": null,
                    "NetworkID": "87996a466060f893a61712559924830f411d1e12caf2a32edc2231a1780ef6
                    "EndpointID": "e6722b4dc1939931c98dd79afb4c3abc15e0b5809269686c428a8045431eb
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.4",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:11:00:04",
                    "DriverOpts": null
                }
```

- set your IP host



Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agents via SSH

Host ?

172.17.0.4

Credentials ?

jenkins

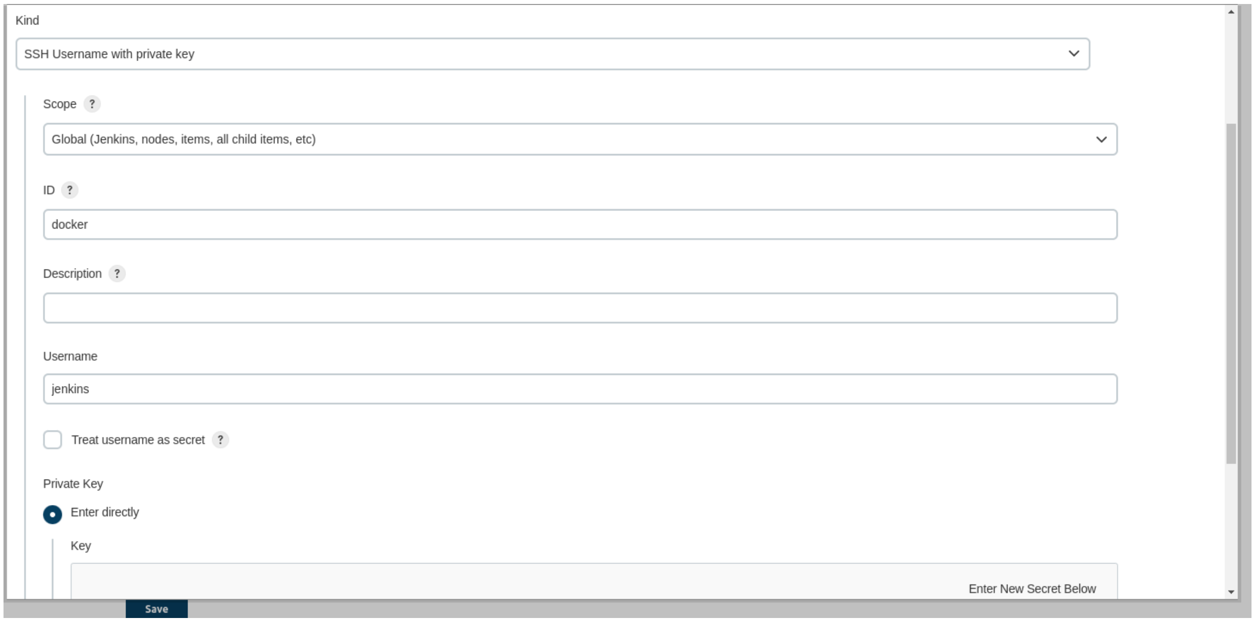
Host Key Verification Strategy ?

Non verifying Verification Strategy

Availability ?

Keep this agent online as much as possible

- set credentials:
  - use as SSH
  - id: set the id that you will use in pipeline in the future
  - [Very important]username: the user inside the container “we created a jenkins user”
  - Private Key: copy and paste your created private key that you generated before
  - save and save your node



Kind

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

docker

Description ?

Username

jenkins

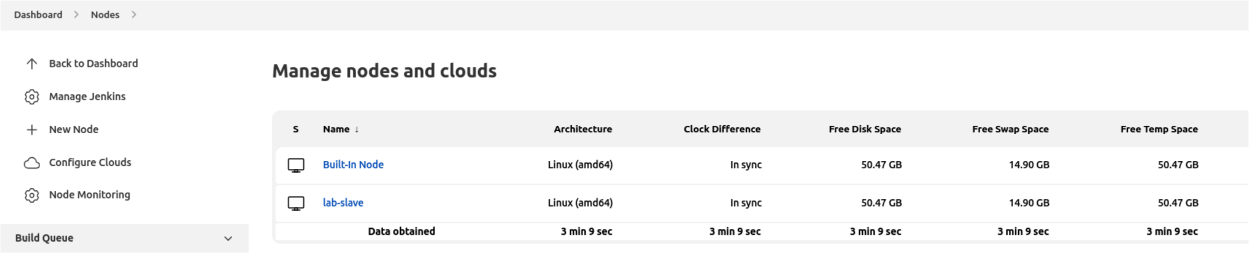
□ Treat username as secret ?

Private Key

● Enter directly

Key

Enter New Secret Below

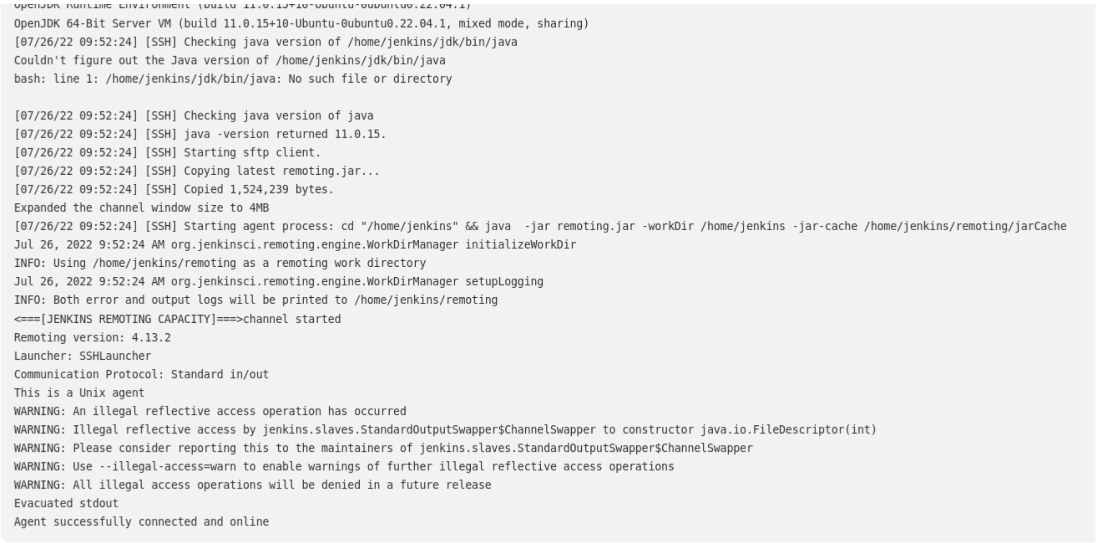


Manage nodes and clouds

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space
🖥	Built-In Node	Linux (amd64)	In sync	50.47 GB	14.90 GB	50.47 GB
🖥	lab-slave	Linux (amd64)	In sync	50.47 GB	14.90 GB	50.47 GB
	Data obtained	3 min 9 sec	3 min 9 sec	3 min 9 sec	3 min 9 sec	3 min 9 sec

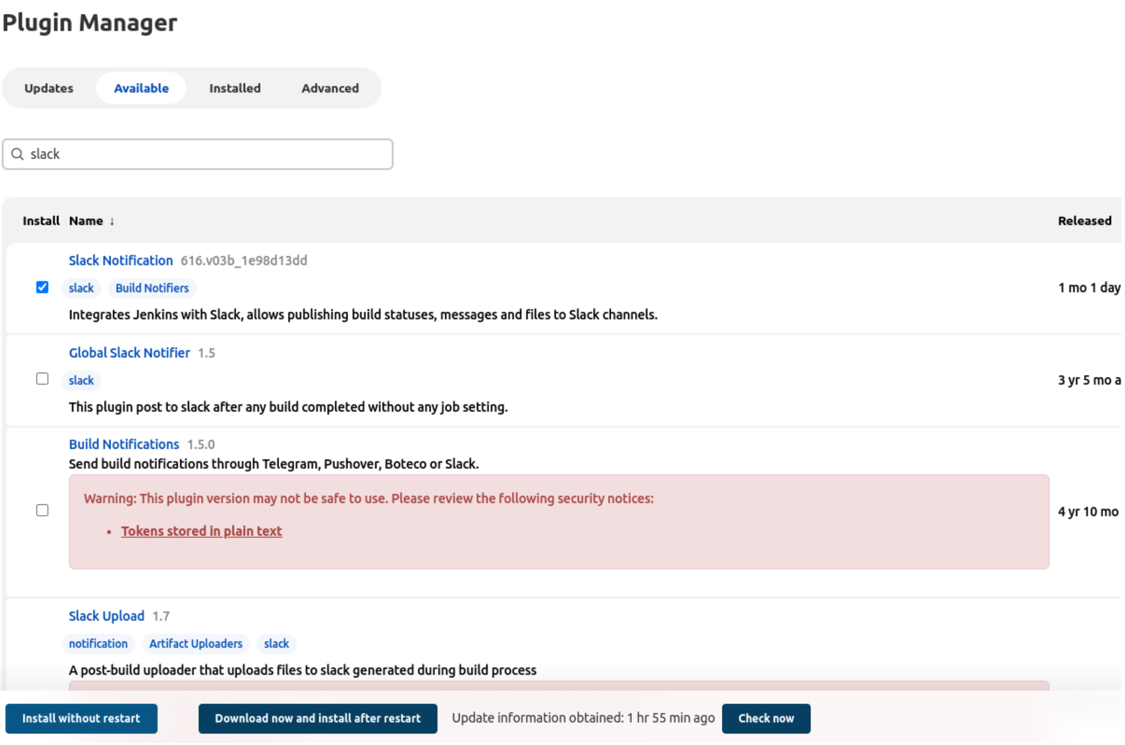
RESULTS

- get inside the node logs



```
OpenJDK Runtime Environment (build 11.0.15+10-Ubuntu-0ubuntu0.22.04.1)
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.22.04.1, mixed mode, sharing)
[07/26/22 09:52:24] [SSH] Checking java version of /home/jenkins/jdk/bin/java
Couldn't figure out the Java version of /home/jenkins/jdk/bin/java
bash: line 1: /home/jenkins/jdk/bin/java: No such file or directory

[07/26/22 09:52:24] [SSH] Checking java version of java
[07/26/22 09:52:24] [SSH] java -version returned 11.0.15.
[07/26/22 09:52:24] [SSH] Starting sftp client.
[07/26/22 09:52:24] [SSH] Copying latest remoting.jar...
[07/26/22 09:52:24] [SSH] Copied 1,524,239 bytes.
Expanded the channel window size to 4MB
[07/26/22 09:52:24] [SSH] Starting agent process: cd "/home/jenkins" && java  -jar remoting.jar -workDir /home/jenkins -jar-cache /home/jenkins/remoting/jarCache
Jul 26, 2022 9:52:24 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins/remoting as a remoting work directory
Jul 26, 2022 9:52:24 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins/remoting
<===[JENKINS REMOTING CAPACITY]===>channel started
Remoting version: 4.13.2
Launcher: SSHLauncher
Communication Protocol: Standard in/out
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSwapper$ChannelSwapper to constructor java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSwapper$ChannelSwapper
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

Q5: Integrate slack with Jenkins and send slack message when stage in your pipeline is successful

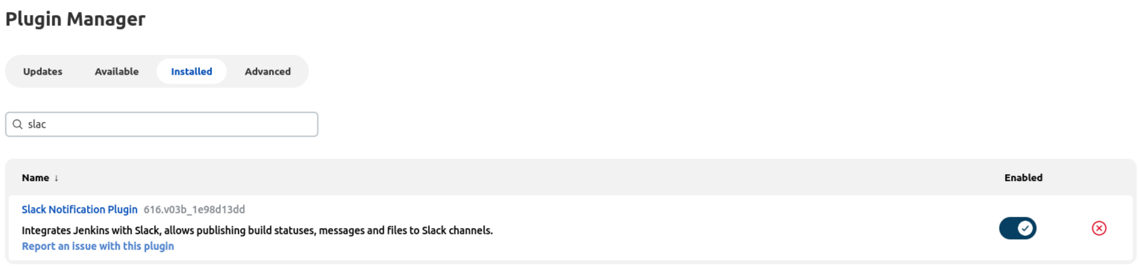


Plugin Manager

slack

Install	Name ↓	Released
☑	Slack Notification 616.v03b_1e98d13dd slack · Build Notifiers Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels.	1 mo 1 day
□	Global Slack Notifier 1.5 slack This plugin post to slack after any build completed without any job setting.	3 yr 5 mo a
□	Build Notifications 1.5.0 Send build notifications through Telegram, Pushover, Boteco or Slack. Warning: This plugin version may not be safe to use. Please review the following security notices: • Tokens stored in plain text	4 yr 10 mo
	Slack Upload 1.7 notification · Artifact Uploaders · slack A post-build uploader that uploads files to slack generated during build process	

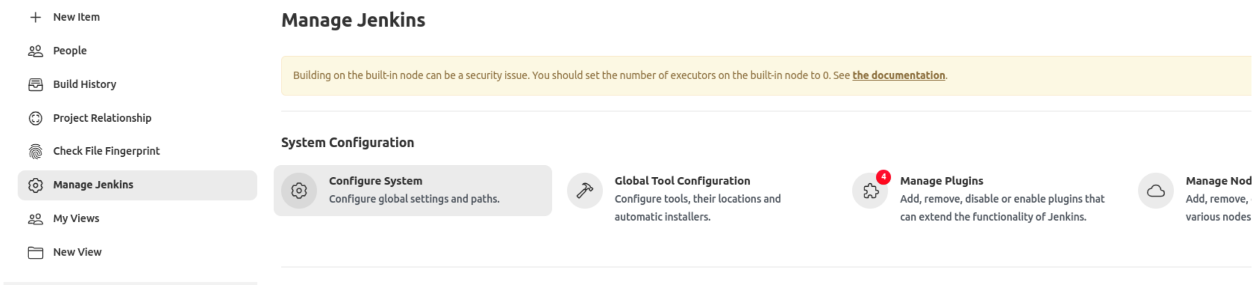
Update information obtained: 1 hr 55 min ago



Plugin Manager

slac

Name ↓	Enabled
Slack Notification Plugin 616.v03b_1e98d13dd Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels. Report an issue with this plugin	✓



Manage Jenkins

Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See the documentation.

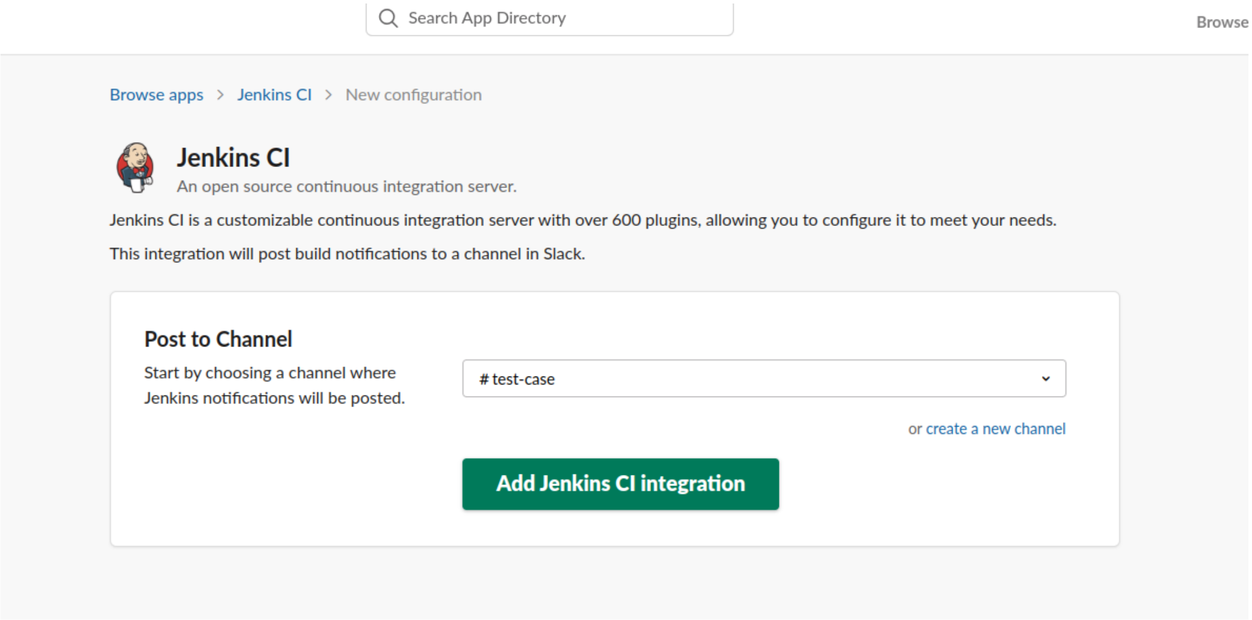
System Configuration

Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins 4
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nod
Add, remove,
various nodes



Browse apps > Jenkins CI > New configuration

Jenkins CI

An open source continuous integration server.

Jenkins CI is a customizable continuous integration server with over 600 plugins, allowing you to configure it to meet your needs.

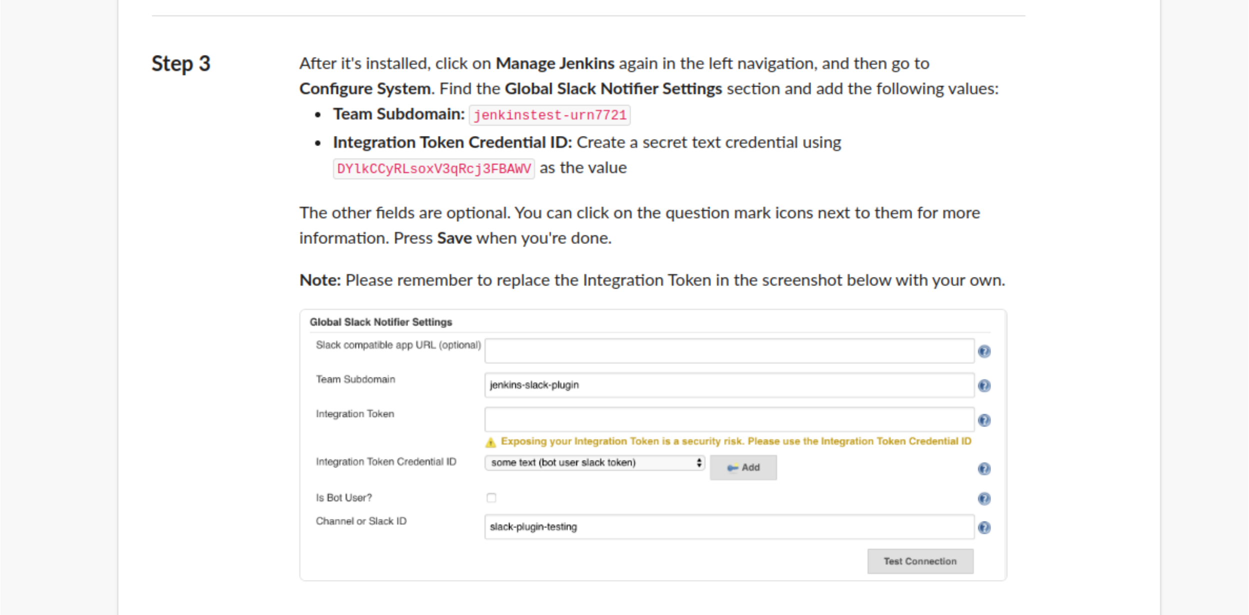
This integration will post build notifications to a channel in Slack.

Post to Channel

Start by choosing a channel where Jenkins notifications will be posted.

test-case

or create a new channel



Step 3

After it's installed, click on **Manage Jenkins** again in the left navigation, and then go to **Configure System**. Find the **Global Slack Notifier Settings** section and add the following values:

- **Team Subdomain:** `jenkinstest-urn7721`
- **Integration Token Credential ID:** Create a secret text credential using `DYlkCCyRLsoxV3qRcj3FBAWV` as the value

The other fields are optional. You can click on the question mark icons next to them for more information. Press **Save** when you're done.

Note: Please remember to replace the Integration Token in the screenshot below with your own.

Global Slack Notifier Settings

Slack compatible app URL (optional)

Team Subdomain: jenkins-slack-plugin

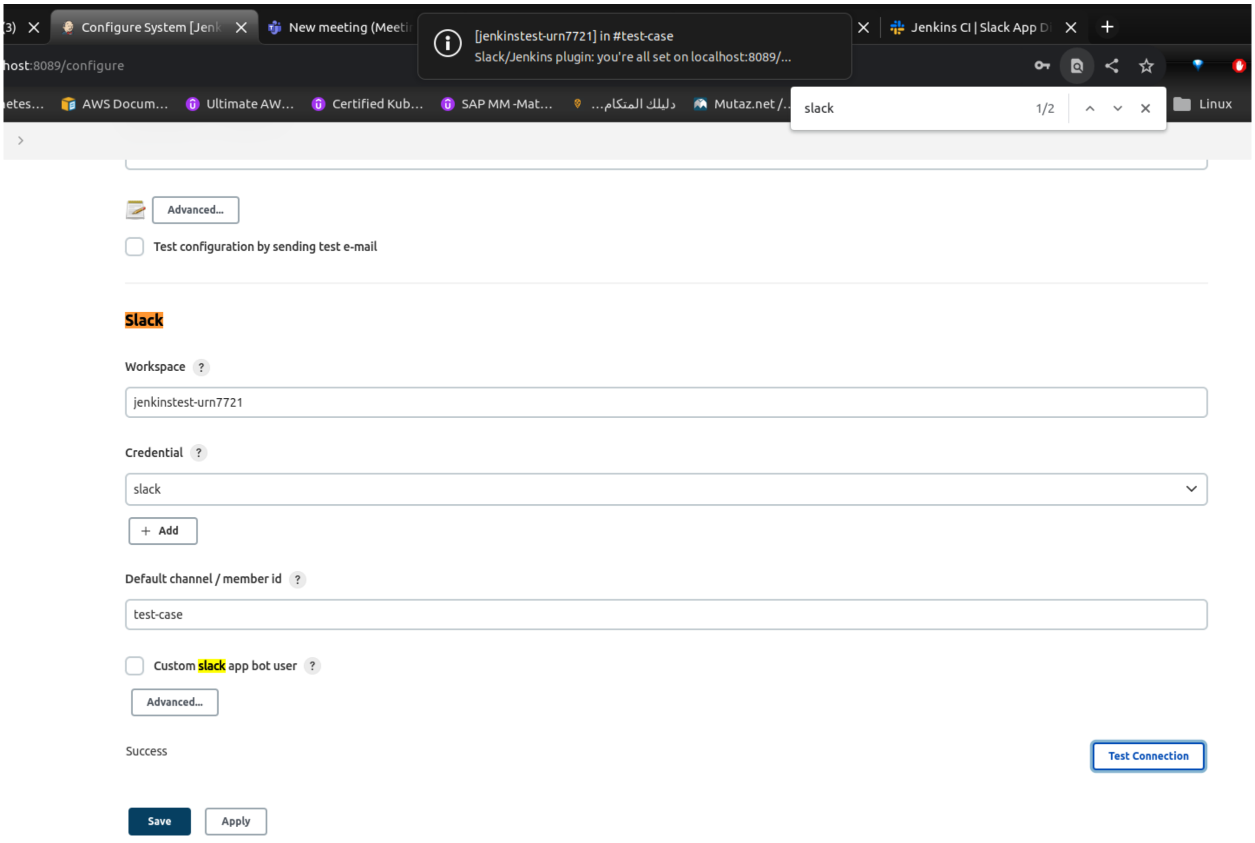
Integration Token

⚠ Exposing your Integration Token is a security risk. Please use the Integration Token Credential ID

Integration Token Credential ID: some text (bot user slack token)

Is Bot User? □

Channel or Slack ID: slack-plugin-testing



[jenkinstest-urn7721] in #test-case
Slack/Jenkins plugin: you're all set on localhost:8089/...

Slack

Workspace ?

jenkinstest-urn7721

Credential ?

slack

Default channel / member id ?

test-case

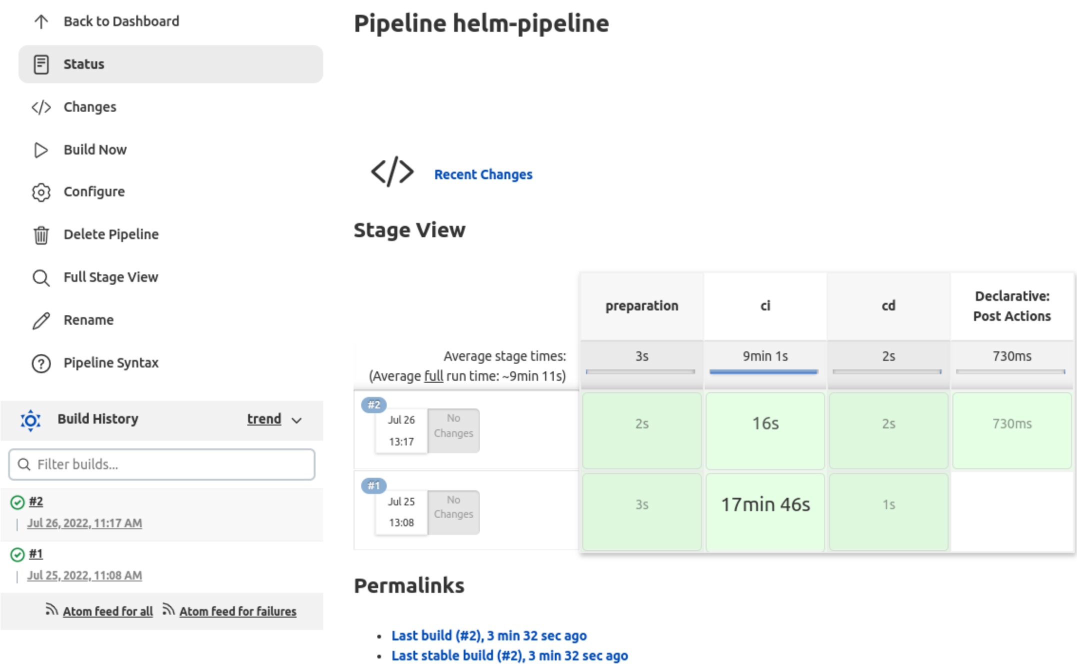
□ Custom slack app bot user ?

Success

-
- I used the previous pipeline to test slack integration using post plugin

```
pipeline {
    agent any

    stages {

        stage('preparation') {
            steps {
                // Get some code from GitHub repository
                git 'https://github.com/mahmoud254/jenkins_nodejs_example.git'
            }
        }

        stage('ci') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'dockerhub', passwordVariable: 'PASSWORD', usernameVariable: 'USERNAME')])
                    sh """
                    docker build . -f dockerfile -t rizk95/jenkins
                    docker login -u ${USERNAME} -p ${PASSWORD}
                    docker push rizk95/jenkins
                    """
                }
            }
        }

        stage('cd') {
            steps {
                sh "docker run -d -p 3000:3000 rizk95/jenkins"
            }
        }
    }
    # for slack notification
     post {
            success {
                slackSend (message:"Build deployed successfully - ${env.JOB_NAME} ${env.BUILD_NUMBER} (<${env.BUILD_URL}|Open>)")
            }
            failure {
                slackSend (message:"Build failed  - ${env.JOB_NAME} ${env.BUILD_NUMBER} (<${env.BUILD_URL}|Open>)")
            }
        }
}
```

RESULTS



- Back to Dashboard
- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Rename
- Pipeline Syntax

Build History trend

Filter builds...

#2 Jul 26, 2022, 11:17 AM

#1 Jul 25, 2022, 11:08 AM

Atom feed for all Atom feed for failures

Pipeline helm-pipeline

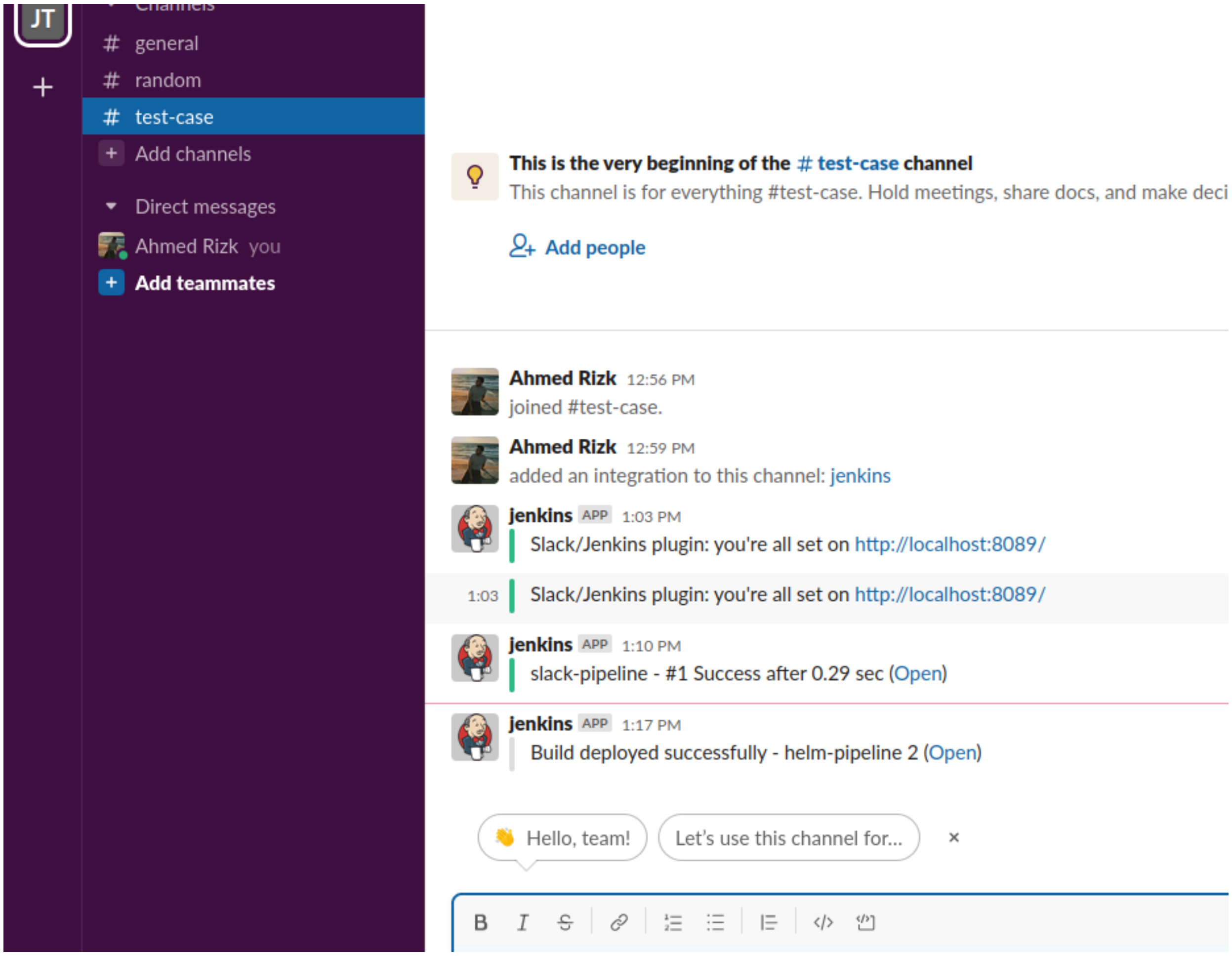
Recent Changes

Stage View

	preparation	ci	cd	Declarative: Post Actions
Average stage times: (Average full run time: ~9min 11s)	3s	9min 1s	2s	730ms
#2 Jul 26 13:17 No Changes	2s	16s	2s	730ms
#1 Jul 25 13:08 No Changes	3s	17min 46s	1s	

Permalinks

- Last build (#2), 3 min 32 sec ago
- Last stable build (#2), 3 min 32 sec ago



This is the very beginning of the # test-case channel
This channel is for everything #test-case. Hold meetings, share docs, and make deci

Ahmed Rizk 12:56 PM
joined #test-case.

Ahmed Rizk 12:59 PM
added an integration to this channel: jenkins

jenkins APP 1:03 PM
Slack/Jenkins plugin: you're all set on http://localhost:8089/

1:03 Slack/Jenkins plugin: you're all set on http://localhost:8089/

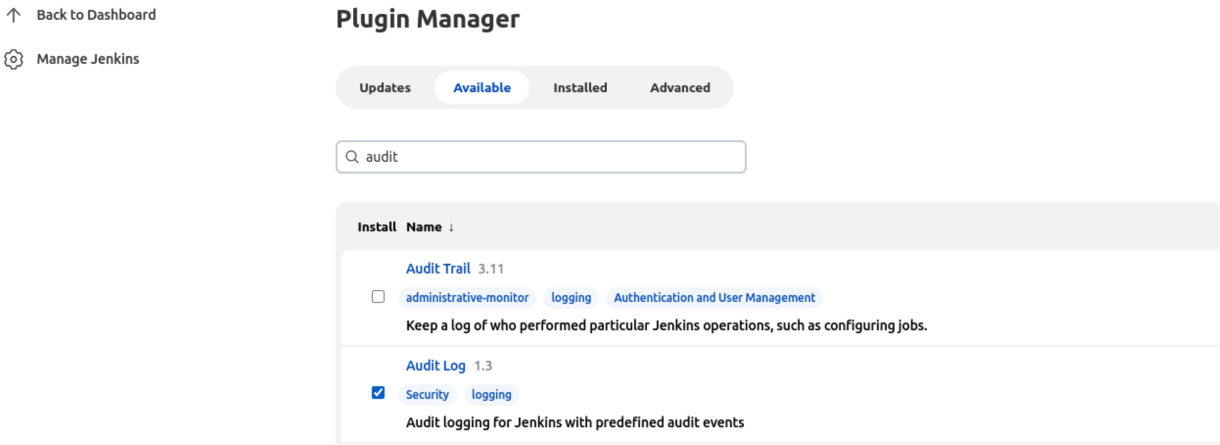
jenkins APP 1:10 PM
slack-pipeline - #1 Success after 0.29 sec (Open)

jenkins APP 1:17 PM
Build deployed successfully - helm-pipeline 2 (Open)

Q5: install audit logs plugin and test it

Manage plugins → audit log → install → restart the docker container

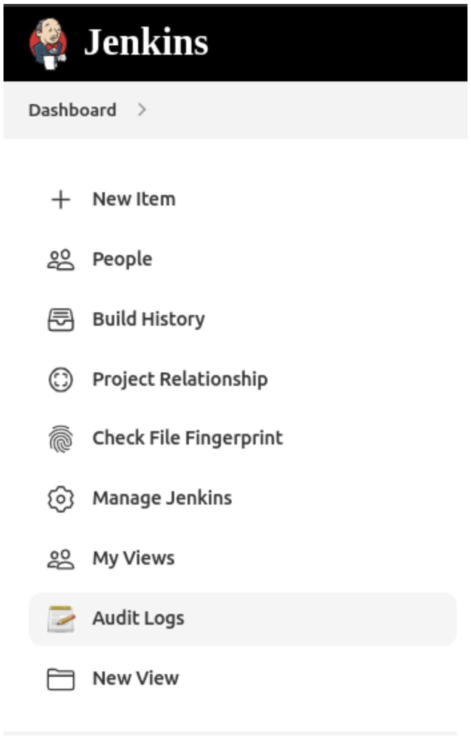
RESULTS

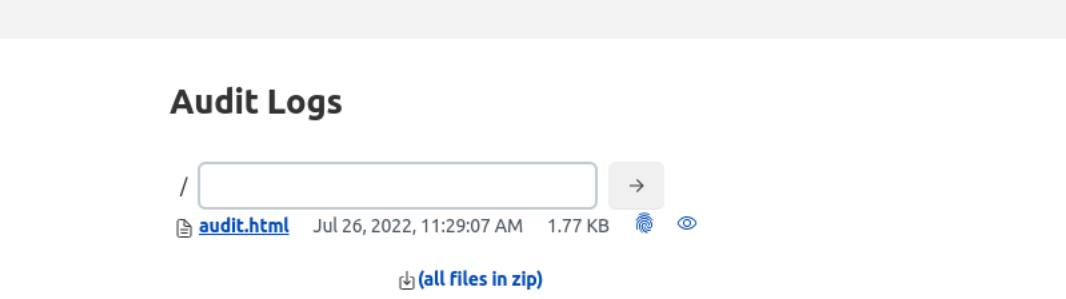


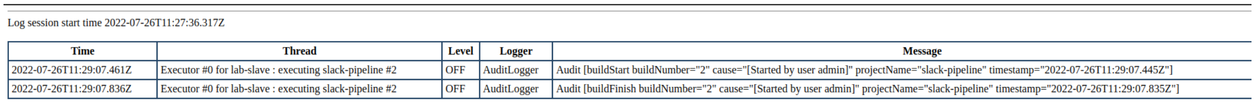
Plugin Manager

audit

Install	Name ↓
☐	Audit Trail 3.11 administrative-monitor logging Authentication and User Management Keep a log of who performed particular Jenkins operations, such as configuring jobs.
☑	Audit Log 1.3 Security logging Audit logging for Jenkins with predefined audit events







Log session start time 2022-07-26T11:27:36.317Z

Time	Thread	Level	Logger	Message
2022-07-26T11:29:07.461Z	Executor #0 for lab-slave : executing slack-pipeline #2	OFF	AuditLogger	Audit [buildStart buildNumber="2" cause="[Started by user admin]" projectName="slack-pipeline" timestamp="2022-07-26T11:29:07.445Z"]
2022-07-26T11:29:07.836Z	Executor #0 for lab-slave : executing slack-pipeline #2	OFF	AuditLogger	Audit [buildFinish buildNumber="2" cause="[Started by user admin]" projectName="slack-pipeline" timestamp="2022-07-26T11:29:07.835Z"]