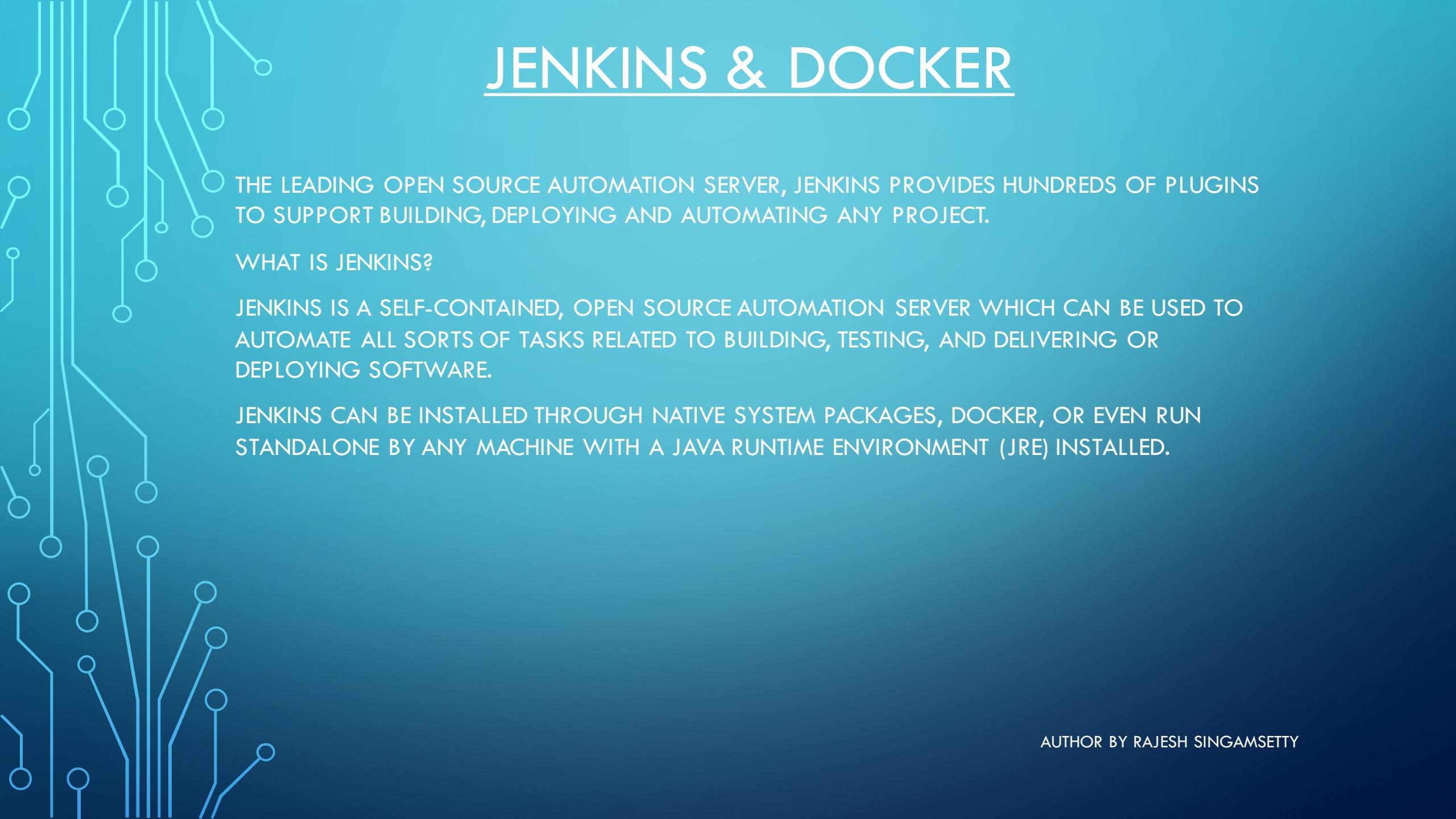


# JENKINS & DOCKER

A faint, light blue circuit board pattern serves as a background for the entire slide.

THE LEADING OPEN SOURCE AUTOMATION SERVER, JENKINS PROVIDES HUNDREDS OF PLUGINS TO SUPPORT BUILDING, DEPLOYING AND AUTOMATING ANY PROJECT.

## WHAT IS JENKINS?

JENKINS IS A SELF-CONTAINED, OPEN SOURCE AUTOMATION SERVER WHICH CAN BE USED TO AUTOMATE ALL SORTS OF TASKS RELATED TO BUILDING, TESTING, AND DELIVERING OR DEPLOYING SOFTWARE.

JENKINS CAN BE INSTALLED THROUGH NATIVE SYSTEM PACKAGES, DOCKER, OR EVEN RUN STANDALONE BY ANY MACHINE WITH A JAVA RUNTIME ENVIRONMENT (JRE) INSTALLED.

# INSTALLING DOCKER USING

- Go to aws and launch redhat instance allow ports as we need..
  - Login ur account as a root user
  - yum update -y
  - dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
  - [root@docker-ce ~]# dnf list docker-ce Available Packages docker-ce.x86\_64 3:19.03.5-3.el7 docker-ce-stable [root@docker-ce
  - To install the latest version: dnf install docker-ce --nobest -y
  - systemctl start docker
  - systemctl enable docker
- 
- [root@ip-172-31-13-145 ~]# docker --version

Docker version 19.03.11, build 42e35e61f3

- To check docker correctly installed or not using
- docker run hello-world

Refrence url: <https://www.linuxtechi.com/install-docker-ce-centos-8-rhel-8/>

```
[root@docker-ce ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:4df8ca8a7e309c256d60d7971ea14c27672fc0d10c5f303856d7bc48f8cc17f
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

```
[root@docker-ce ~]#
```

# INSTALL DOCKER COMPOSE

- Note: Docker should be start and enabled.(below steps must be u install with sudo commands otherwise compose not install)  
`dnf install curl`
- `sudo curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
- To give permission to docker compose file
- `sudo chmod +x /usr/local/bin/docker-compose`
- `sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose`
- `$ docker-compose --version`

`docker-compose version 1.26.0, build 1110ad01`

For Reference:

<https://docs.docker.com/compose/install/>

# DOWNLOAD JENKINS WITH DOCKER IMAGE

- docker pull jenkins/jenkins
- To check how many images are present in docker:
- docker images
- To check where docker is located : docker info | grep -i root
- cd /home/
- mkdir jenkins
- mkdir jenkins-data
- cd jenkins-data
- pwd (/home/jenkins/jenkins-data)

# DOCKER COMPOSE FILE

- cd jenkins-data : vi docker-compose.yml
- vi docker-composer.yml

```
version: '3'
services:
  jenkins:
    image: jenkins/jenkins
  ports:
    - 8080:8080
    volumes:
      - $PWD/jenkins_home:/var/jenkins_home
    networks:
      - net
  networks:
    net:
```

## CONTINUE

```
mkdir jenkins_home
```

```
echo $pwd
```

```
echo $PWD/jenkins_home/
```

```
sudo chown 1000:1000 jenkins_home -R
```

(to run the docker file)- docker-compose up -d

```
Creating network "jenkins-data_net" with the default driver
```

```
Creating jenkins-data_jenkins_1 ... Done
```

Output will be above way will be there.

Now type docker ps u will see an image like jenkins/jenkins

# RUN JENKINS WITH PORTNO:8080

- Before going to start and enter portno 8080:
- We need jenkins admin password to start the jenkins
- `cd /home/jenkins/jenkins-data/jenkins_home/secrets`
- `cat initialAdminPassword` (copy the password and paste in ip:8080) url:
- Click on install plugin go and create username and password
- Start using jenkins
- Welcome to jenkins (admin/admin@123)

# DOCKER COMPOSE COMMANDS (JENKINS-DATA)

- docker-compose stop (to stop container ex: jenkins will stop)
- docker-compose start (to start container ex: jenkins will start)
- docker-compose restart jenkins (to restart container ex: jenkins will start)
- docker-compose down (it will delete all the complete containers 4 m docker)
- docker-compose up -d (to run the composer containers)

# JENKINS RHEL8 AWS INSTALLATION

- `yum update -y`
- `sudo yum install java-1.8.0-openjdk-devel`
- `which java`
- `cd /bin`
- `ls`
- `which java`
- `ls -la | grep java`
- `cd`
- `java -version`
- `curl --silent --location http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo | sudo tee /etc/yum.repos.d/jenkins.repo`
- `sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key`
- `yum -y install jenkins`
- `systemctl status jenkins`
- `systemctl start jenkins`
- `systemctl status jenkins`
- `systemctl enable jenkins`
- `cat /var/lib/jenkins/secrets/initialAdminPassword`
- `.`

# JENKINS DASHBOARD

- Create a job → new item → free style project(give name) → ok → build (Execute shell) echo hello world → apply/save.-→ build now.→success console output

Hello world will be displayed.

Go to putty run below commands to login docker container

```
. docker run -it -d jenkins/jenkins (image name)
```

It will give long id paste that id in below command

```
. docker exec -it
```

```
5534ed52b63f22c41b8120e7335d677d2a53b70843929f4787c5e07d38214cb5
```

```
bash
```

## CONTINUE-

- Login into shell script of container of docker
- jenkins@5534ed52b63f:/\$ date

Mon Jun 8 20:41:43 UTC 2020

Jenkins execute shell:

```
name=rajesh
```

```
echo "hello , $name. data & time is $(date)"
```

Build now (success go to console output).

Redirects to job output

```
name=rajesh
```

```
echo "hello , $name. data & time is $(date)" > /tmp/info
```

AUTHOR BY RAJESH SINGAMSETTY

# JOB OUTPUT REDIRECT

- In putty run ur container and see (redirect job output with in a container )

```
jenkins@4c4dc4c4b01e:$ cat /tmp/info
```

hello , rajesh. data & time is Tue Jun 9 21:02:25 UTC 2020

Learn execute bash script from jenkins

vi script.sh

```
#!/bin/bash  
  
NAME=$1  
  
LAST=$2  
  
echo "hello ,$NAME $LAST"
```

To execute : chmod +x filename and after run : ./script.sh raj singam

# ADD PARAMETER TO JOB

- Configure of job general this project is parametrized → add parameter->string (name : firstname , value: rajesh) → another sting paramter-> (name : secondname , value: singamsetty).
- Build → execute shell (echo "heello ,\$firstname .\$secondname ") + save and build choice parameter we have select any option to build.

Go to putty script.sh file

- if [ "\$SHOW" = "true" ]; then
- echo "hello ,\$NAME \$LAST"
- else
- echo -e "please check the mark"
- fi

To run script file : ./script.sh raj singam true (above putty what we did do in jenkins)

# JENKINS BOOLEN PARAMETER

- Along with first name and last name take boolean ->name :show & default value will in checkmode.
- Build : /tmp/script.sh \$firstname singam true
- Build now.
- Putty inside container must be
- To copy from docker to container
- docker cp script.sh jenkins:/tmp/script.sh

## JENKINS & DOCKER (PART2)

- cd /home/jenkins/jenkins-data/
- mkdir centos7 and cd centos7
- vi Dockerfile (paste the below code)

FROM centos

```
RUN yum -y install openssh-server (to run ssh server)
RUN useradd raj && \
    (create a user add password)
    echo "1234" | passwd raj --stdin && \
    mkdir /home/raj/.ssh && \
    (creating home directory)
    chmod 700 /home/raj/.ssh    (giving permissions 2 home)
```

:wq (for the saving a file)

# CONTINUE

- ssh-keygen -f remote\_key (after press 3 times enter)
- ll or ls to see the what are the files are created in centos folder.
- Now public key and private key both were created
- Now we are going to make connection using ssh (public key)
- vi Dockerfile

```
RUN yum -y install openssh-server  
RUN useradd raj && \  
    echo "1234" | passwd raj --stdin && \  
    mkdir /home/raj/.ssh && \  
    chmod 700 /home/raj/.ssh
```

Copy