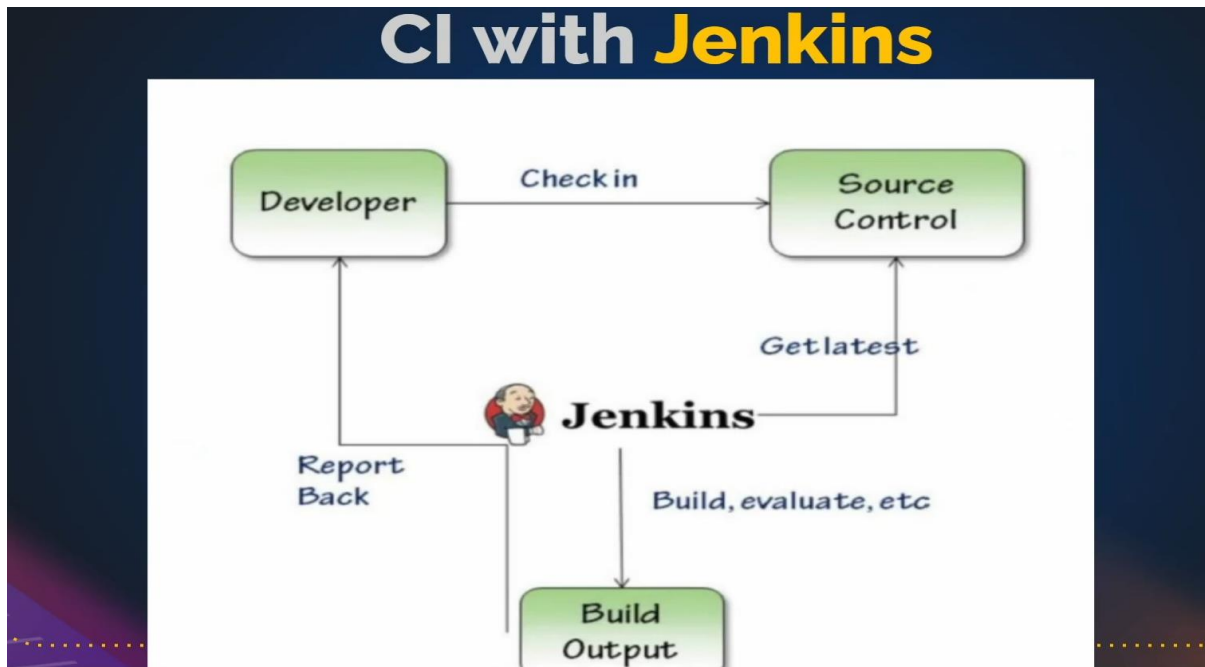


Continuous Integration with Jenkins

This project scope explains the process of setting up jenkins server, pulling source code from git hub, configure maven in jenkins to generate build. The build can be generated with the parameter as well.



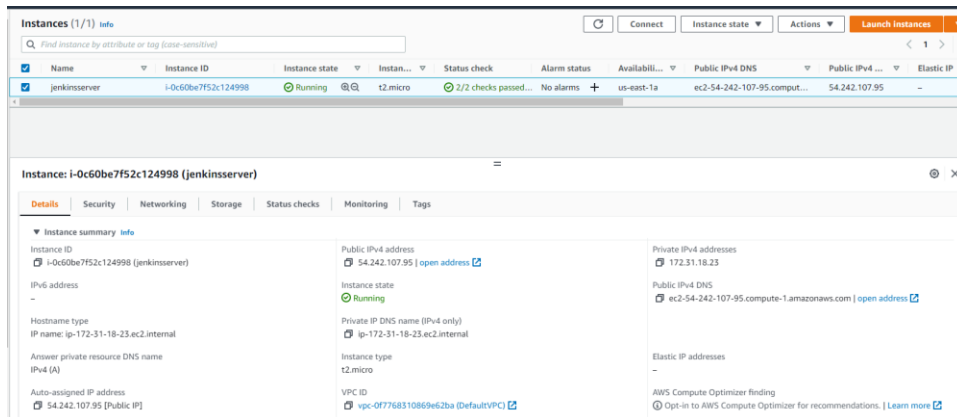
1. Create a security group which allows ssh from local, and allows on port 8080

The screenshot shows the AWS IAM console's 'Inbound rules' tab for a security group. It displays two rules: one for Custom TCP on port 8080 and one for SSH on port 22, both allowing traffic from the source IP 106.212.17.126/32.

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-048f8dc2fbd5794e2	IPv4	Custom TCP	TCP	8080	106.212.17.126/32	-
<input type="checkbox"/>	-	sgr-03eaa4e604ad7cac2	IPv4	SSH	TCP	22	106.212.17.126/32	-

And create access key for login EC2 instance.

2. Create a EC2 instance with the below user data,

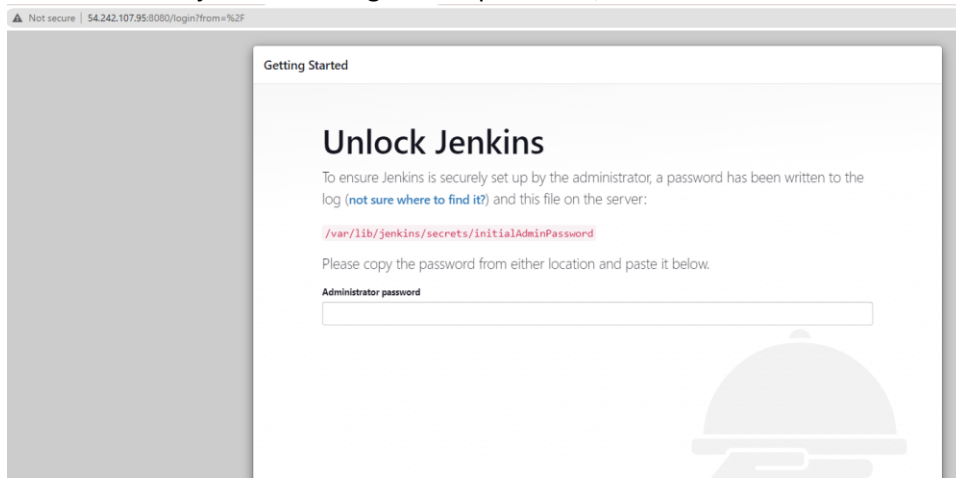


The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there's a table listing instances, with 'jenkinsserver' (ID: i-0c60be7f52c124998) in a 'Running' state. Below this, the 'Details' tab for the instance is selected, showing various attributes:

Category	Attribute	Value
Instance summary	Instance ID	i-0c60be7f52c124998 (jenkinsserver)
	IPv4 address	-
	Hostname type	IP name: ip-172-31-18-23.ec2.internal
	Answer private resource DNS name	IPV4 (A)
Networking	Public IPv4 address	54.242.107.95 open address
	Instance state	Running
	Private IP DNS name (IPv4 only)	ip-172-31-18-23.ec2.internal
	Instance type	t2.micro
Security	VPC ID	vpc-0f7768310869e62ba (DefaultVPC)
	Private IPv4 addresses	172.31.18.23
	Public IPv4 DNS	ec2-54-242-107-95.compute-1.amazonaws.com open address
	Elastic IP addresses	-

```
#!/bin/bash
sudo apt update
sudo apt install openjdk-11-jdk -y
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
```

3. We can see jenkins running on the port 8080,



4. Connect to Jenkins server from local bash using the key.

```
satzw@LAPTOP-C4RG1671 MINGW64 /e/devops/Jenkins
$ ssh -i jenkins-key.pem ubuntu@54.242.107.95
The authenticity of host '54.242.107.95 (54.242.107.95)' can't be established.
ED25519 key fingerprint is SHA256:oCHbTBMGJzJ11JSfwZquCoSSjuIP1T4H/puIrxhx7Go.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.242.107.95' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Sep 30 10:19:18 UTC 2022

System load:  0.0                Processes:    99
Usage of /:   33.3% of 7.57GB    Users logged in: 0
Memory usage: 50%              IPv4 address for eth0: 172.31.18.23
Swap usage:  0%

26 updates can be applied immediately.
```

5. Check jenkins is running,

```
ubuntu@ip-172-31-18-23:~$ systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
   Active: active (running) since Fri 2022-09-30 08:50:32 UTC; 1h 42min ago
     Main PID: 5216 (java)
        Tasks: 38 (limit: 1143)
      Memory: 315.3M
     CGroup: /system.slice/jenkins.service
            └─5216 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java>

Sep 30 08:50:32 ip-172-31-18-23 systemd[1]: Started Jenkins Continuous Integrat>
Sep 30 08:50:32 ip-172-31-18-23 jenkins[5216]: 2022-09-30 08:50:32.312+0000 [id>
Sep 30 08:50:32 ip-172-31-18-23 jenkins[5216]: 2022-09-30 08:50:32.313+0000 [id>
Sep 30 08:50:32 ip-172-31-18-23 jenkins[5216]: 2022-09-30 08:50:32.324+0000 [id>
Sep 30 08:59:21 ip-172-31-18-23 jenkins[5216]: 2022-09-30 08:59:21.955+0000 [id>
Sep 30 08:59:21 ip-172-31-18-23 jenkins[5216]: 2022-09-30 08:59:21.957+0000 [id>
Sep 30 09:59:21 ip-172-31-18-23 jenkins[5216]: 2022-09-30 09:59:21.955+0000 [id>
Sep 30 09:59:21 ip-172-31-18-23 jenkins[5216]: 2022-09-30 09:59:21.958+0000 [id>
Sep 30 10:12:49 ip-172-31-18-23 jenkins[5216]: 2022-09-30 10:12:49.601+0000 [id>
Sep 30 10:12:49 ip-172-31-18-23 jenkins[5216]: 2022-09-30 10:12:49.605+0000 [id>
lines 1-19/19 (END)
```

6. Get the jenkins password from cat/var/lib/jenkins/secrets/initialAdminPassword

```
ubuntu@ip-172-31-18-23:~$ sudo -i
root@ip-172-31-18-23:~# cat /var/lib/jenkins/secrets/initialAdminPassword
b94c18a59fc44907a8e7df8309813b38
root@ip-172-31-18-23:~#
```

With pwd login to the jenkins portal.

7. Update global tool configuration,
 - a. Create a oracle account
 - b. Select jdk8u221 and choose oracle account
 - c. Select Maven 3.8.6

Dashboard > Manage Jenkins > Global Tool Configuration

↑ Back to Dashboard
⚙️ Manage Jenkins

Global Tool Configuration

Maven Configuration

Default settings provider
Use default maven settings

Default global settings provider
Use default maven global settings

JDK
JDK installations...

Git

Git installations

Git

Name
Default

8. Create a new job, in a freestyle project
 - a. Choose git <https://github.com/satzwebio/vprofile-repo.git>

Git

Repositories

Repository URL

https://github.com/satzwebio/vprofile-repo.git

Branches to build

Branch Specifier (blank for 'any')

*/vp-rem

Build Steps

Invoke top-level Maven targets

Maven Version
satzmaven

Goals
install

POM

Click on Buildnow to build the job and

↑ Back to Dashboard

Status

</> Changes

Workspace

▶ Build Now

⚙️ Configure

🗑️ Delete Project

✎️ Rename

Project satzbuid

My first simple project

Permalinks

- [Last build \(#1\), 43 sec ago](#)
- [Last stable build \(#1\), 43 sec ago](#)
- [Last successful build \(#1\), 43 sec ago](#)
- [Last completed build \(#1\), 43 sec ago](#)

Build History

trend ▾

✓ #1

Sep 30, 2022, 5:08 PM

📡 Atom feed for all

📡 Atom feed for failures

Console will show build is successfully completed

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (230 kB at 5.0 MB/s)
[INFO] Installing /var/lib/jenkins/workspace/satzbuild/target/vprofile-v2.war to /var/lib/jenkins/.m2/repository/com/visualpathit/vprofile/v2/vprofile-v2.war
[INFO] Installing /var/lib/jenkins/workspace/satzbuild/pom.xml to /var/lib/jenkins/.m2/repository/com/visualpathit/vprofile/v2/vprofile-v2.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 28.101 s
[INFO] Finished at: 2022-09-30T17:09:12Z
[INFO] -----
Finished: SUCCESS
```

9. Workspace will show the files the job generated.

Dashboard > satzbuilt >

↑ Back to Dashboard

Status

</> Changes

Workspace

▶ Build Now

⚙️ Configure

🗑️ Delete Project

✎ Rename

🔍 Build History trend ▾

#1 Sep 30, 2022, 5:08 PM

Workspace of satzbuilt on Built-In Node

satzbuilt / target /

→

classes

generated-sources/annotations

generated-test-sources/test-annotations

maven-archiver

maven-status/maven-compiler-plugin

site/jacoco

surefire-reports

test-classes/com/visualpathit/account

vprofile-v2

jacoco.exec

Sep 30, 2022, 5:09:07 PM 84.99 KB

vprofile-v2.war

Sep 30, 2022, 5:09:11 PM 43.27 MB

(all files in zip)

10. To archive the file that ends with .war inside any directory archive it. It in post build section.

Post-build Actions

≡ Archive the artifacts ?

Files to archive ?

Advanced...

Add post-build action ▾

After build last successful artifacts will be archived,

↑ Back to Dashboard

Project satzbuilt

My first simple project

Status

</> Changes

Workspace

▶ Build Now

⚙️ Configure

Last Successful Artifacts

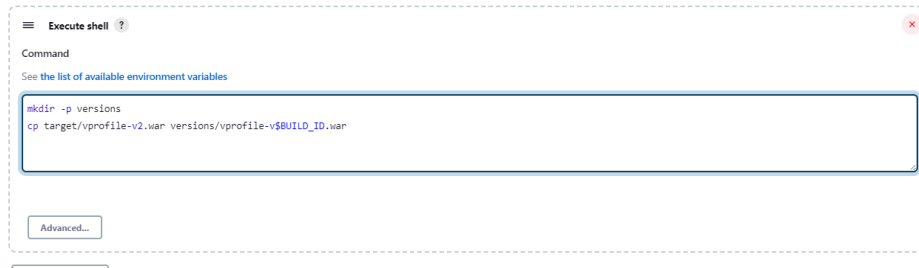
vprofile-v2.war

43.27 MB

view


Permalinks

To move the artifacts into a new directory, with a new filename every time. Add this in build step



```
mkdir -p versions
cp target/vprofile-v2.war versions/vprofile-v$BUILD_ID.war
```

11. The below option enable user to enter build the value with parameter. Create a variable like “VERSION”, then in build step use \$VERSION



[Plain text] [Preview](#)

☐ Change date pattern for the BUILD_TIMESTAMP (build timestamp) variable ?

☐ Discard old builds ?

☐ GitHub project

☒ This project is parameterized ?

☐ Filter

☐ Boolean Parameter necessary ?

☐ Choice Parameter

☐ Credentials Parameter

☐ File Parameter

☐ Multi-line String Parameter

☐ Password Parameter

☒ Run Parameter

☐ String Parameter

☐ None

Or you can select change date pattern and enter pattern below and use \$BUILD_TIMESTAMP in execute shell.



☒ Change date pattern for the BUILD_TIMESTAMP (build timestamp) variable ?

Date and Time Pattern ?

yy-MM-dd_HH:mm

You must specify a [java.text.SimpleDateFormat](#) pattern. For example give the following value: yyyyMMddHHmmss.

☐ Discard old builds ?