

David Ezeji
11/04/2019

Jenkins Project

Overview: Spun up two Linux instances on AWS and installed Jenkins on the first instance so that I could automate the processes of building my application and deploying it to the second instance. On the second instance I created a Docker container to host my application.

1. Created the first Linux instance which will host Jenkins

Services ▾

Search for services, features, marketplace products, and docs [Option+5]

DavidEzaji ▾ N. Virginia ▾ Support ▾

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.

[Learn more](#) about Amazon EC2 security groups.

Assign a security group:

☒ Create a new security group
☐ Select an existing security group

Security group name:
 Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP	TCP	8080	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning
 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Services ▾

Search for services, features, marketplace products, and docs [Option+5]

DavidEzaji ▾ N. Virginia ▾ Support ▾

New EC2 Experience

Feedback English (US) ▾

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

Instances (1)

Filter instances

search: i-Ofed7363f6712b6d2 Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Jenkins, CI Se...	i-Ofed7363f6712b6d2	Running	t2.micro	-	No alarms	us-east-1

Select an instance above

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

2. Installed Java onto the instance

```

last login: Wed Sep 29 15:29:15:02 2021 from ec2-18-206-107-24.compute-1.amazonaws.com
[ec2-user@ip-172-31-23-236 ~]$ sudo su -
last login: Wed Sep 29 15:30:02 UTC 2021 on pts/0
[root@ip-172-31-23-236 ~]# java -version
bash: java: command not found
[root@ip-172-31-23-236 ~]# yum install java-1.8.0-openjdk
Loading plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                | 3.7 kB  00:00:00
amzn2xtra-docker                          | 3.0 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package java-1.8.0-openjdk.x86_64 !1:1.8.0.302.b08.0.amzn2.0.1 will be installed
--> Processing Dependency: java-1.8.0-openjdk-headless(x86_64) = 1:1.8.0.302.b08.0.amzn2.0.1 for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: xorg-x11-fonts-Type1 for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libjvm.so(SUNWprivate_1.1)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libjls.so(SUNWprivate_1.1)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libsound.so.2(ALSA_0.9)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libsound.so.2(ALSA_0.9)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libtcmalloc(x86_64) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libtcmalloc(x86_64) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: fontconfig(x86-64) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: fontconfig(x86-64) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64
--> Processing Dependency: libvm.so()(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.302.b08.0.amzn2.0.1.x86_64

```

- Created a Java path within the .bash_profile directory, so that Jenkins would be able to find it once downloaded.

```
Last login: Wed Sep 29 16:46:39 2021 from ec2-16-206-107-25.compute-1.amazonaws.com
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
11 package(s) needed for security, out of 35 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-23-236 ~]$ cd /
[ec2-user@ip-172-31-23-236 ~]$ ls -la
.  ..  autorelabel  bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  usr  var
[ec2-user@ip-172-31-23-236 lib]$ ls -la
.  ..  dracut  grub  java-1.7.0  jvm-common  kernel  modules-load.d  sendmail  systemd
firmware  java-1.8.0  jvm-exports  locale  python2.7  sendmail.postfix  tapfiles.d
binfmt.d  fontconfig  java-1.5.0  java-ext  jvm-private  modprobe.d  python3.7  sse2  udev
debug  games  java-1.6.0  jvm  kbd  modules  rpm  sysctl.d  yum-plugins
[ec2-user@ip-172-31-23-236 lib]$ cd jvm
[ec2-user@ip-172-31-23-236 jvm]$ ls -la
.  ..  java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64  jre-1.8.0  jre-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
jre
[ec2-user@ip-172-31-23-236 jvm]$ cd java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
[ec2-user@ip-172-31-23-236 java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64]$ pwd
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
[ec2-user@ip-172-31-23-236 java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64]$
```

i-Ofed7363f6712b6d2 (Jenkins_CI_Server)

Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

```
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME
export PATH

. bash_profile" 12L, 261C written
[root@ip-172-31-23-236 ~]#
```

i-Ofed7363f6712b6d2 (Jenkins_CI_Server)

Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

- Installed Apache Maven onto my EC2 instance and set a path for it (Jenkins won't install without Maven being present)

```
[root@ip-172-31-23-236 ~]# mvn -v
Apache Maven 3.5.2 (138ed61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T07:58:13Z)
Maven home: /usr/share/apache-maven
Java version: 1.8.0_302, vendor: Red Hat, Inc.
Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.14.243-185.433.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-23-236 ~]#
```

i-Ofed7363f6712b6d2 (Jenkins_CI_Server)

Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

```
. bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
MAVEN_HOME=/usr/share/apache-maven
export PATH=$PATH:$HOME/bin:$JAVA_HOME:$MAVEN_HOME:$PATH
export PATH

. bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
MAVEN_HOME=/usr/share/apache-maven
export PATH=$PATH:$HOME/bin:$JAVA_HOME:$MAVEN_HOME:$PATH
export PATH

. bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86_64
MAVEN_HOME=/usr/share/apache-maven
export PATH=$PATH:$HOME/bin:$JAVA_HOME:$MAVEN_HOME:$PATH
export PATH
```

i-Ofed7363f6712b6d2 (Jenkins_CI_Server)

Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

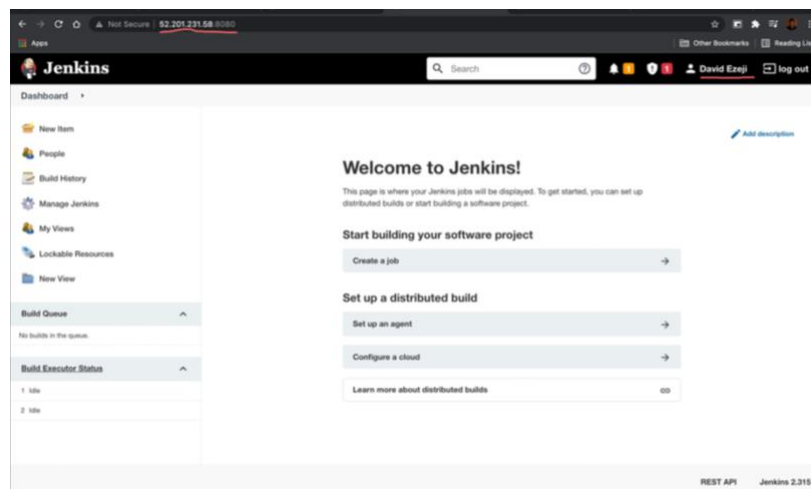
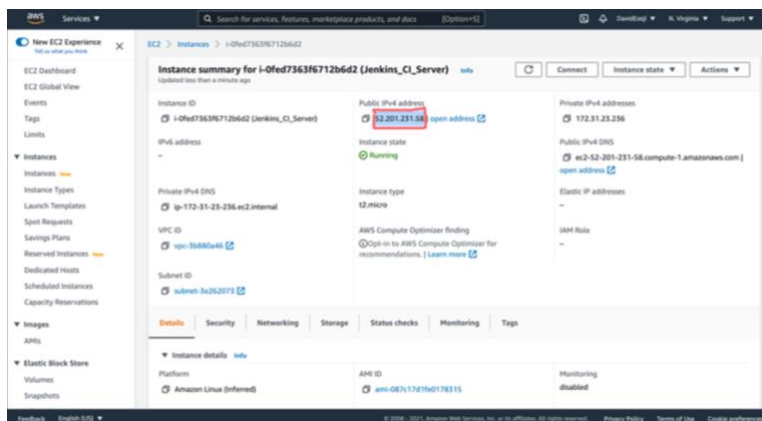
5. Installed Jenkins and started the Jenkins server

```
last login: Fri Oct 8 00:11:39 2021 from ec2-18-208-107-24.compute-1.amazonaws.com
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-23-236 ~]$ sudo su -
last login: Fri Oct 8 00:11:40 UTC 2021 on pts/0
[root@ip-172-31-23-236 ~]# sudo systemctl start jenkins
[root@ip-172-31-23-236 ~]# systemctl status jenkins
● jenkins.service - LSB: Jenkins Automation Server
   Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)
   Active: active (running) since Fri 2021-10-08 00:24:55 UTC; 49s ago
     Docs: man:systemd-sys-generator(8)
    Process: 29223 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/SUCCESS)
    Group: /system.slice/jenkins.service
    CGroup: /system.slice/jenkins.service
            └─29227 /usr/bin/java -Djava.awt.headless=true -DJENKINS_HOME=/var/lib/jenkins -jar /usr/lib/jenkins/jenkins.war --logfile=/var...

Oct 08 00:24:55 ip-172-31-23-236.ec2.internal systemd[1]: Starting LSB: Jenkins Automation Server...
Oct 08 00:24:55 ip-172-31-23-236.ec2.internal jenkins[29223]: Starting Jenkins [ OK ]
Oct 08 00:24:55 ip-172-31-23-236.ec2.internal systemd[1]: Started LSB: Jenkins Automation Server.
[root@ip-172-31-23-236 ~]#
```

i-Ofed7363f6712b6d2 (Jenkins_CI_Server)
Public IP: 52.201.231.58 Private IP: 172.31.23.236

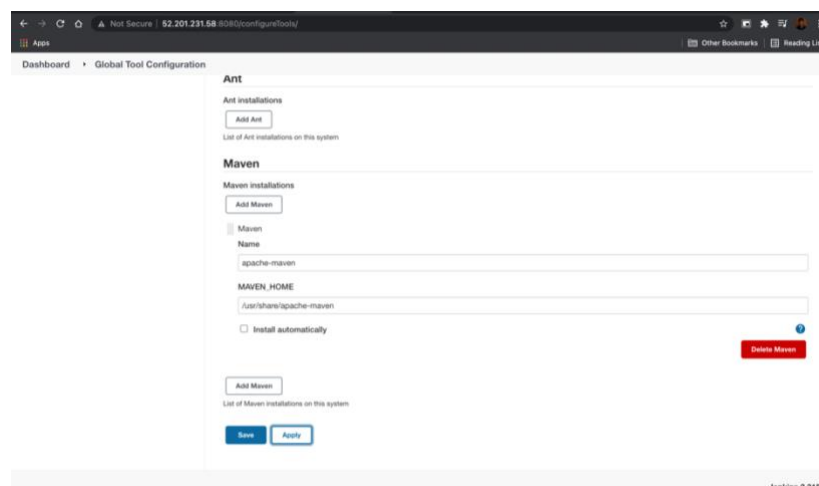
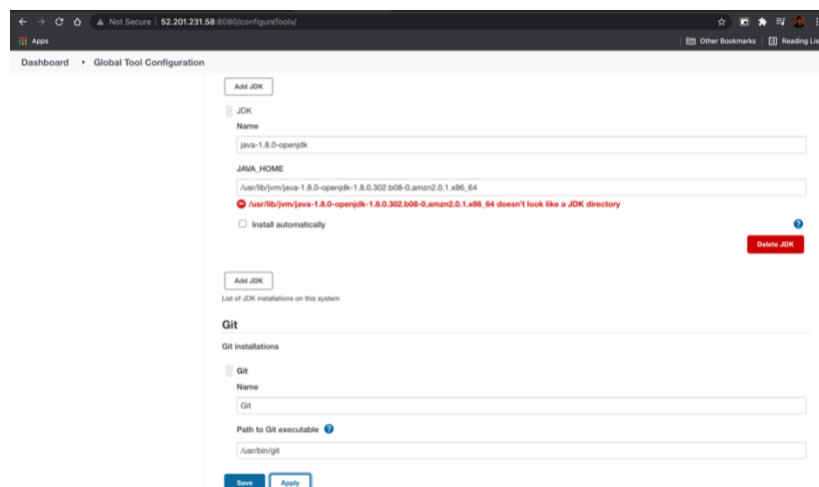


6. Installed Git on the EC2 instance

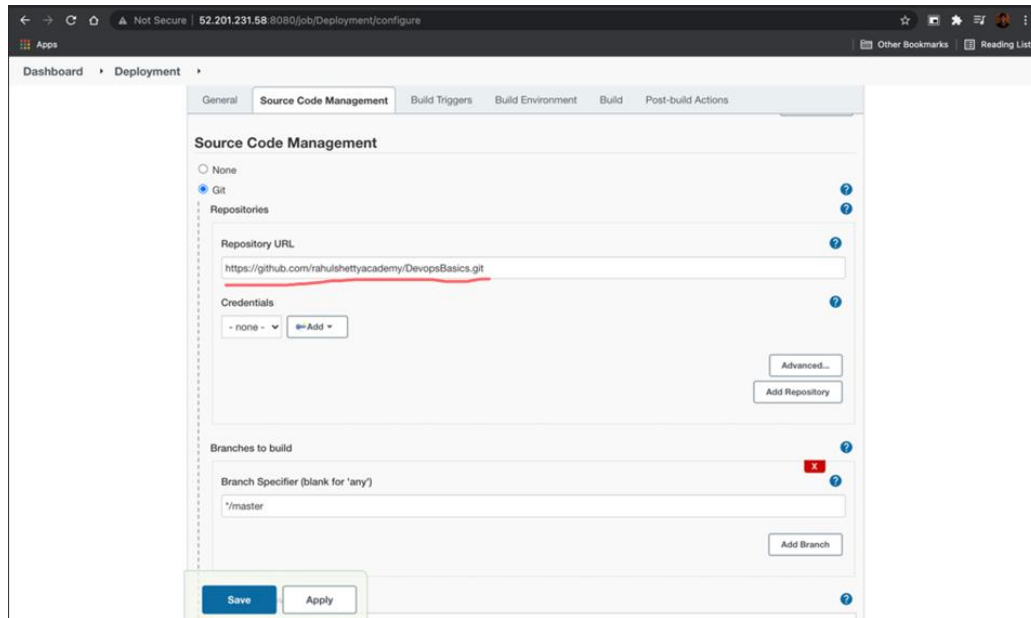
```
12/71: git-2.32.0-1.amzn2.0.1.x86_64.rpm | 126 kB 00:00:00
13/71: git-core-doc-2.32.0-1.amzn2.0.1.noarch.rpm | 2.7 MB 00:00:00
14/71: git-core-2.32.0-1.amzn2.0.1.x86_64.rpm | 4.8 MB 00:00:00
15/71: perl-Error-0.17020-2.amzn2.noarch.rpm | 32 kB 00:00:00
16/71: perl-git-2.32.0-1.amzn2.0.1.noarch.rpm | 43 kB 00:00:00
17/71: perl-TermReadKey-2.30-20.amzn2.0.2.x86_64.rpm | 31 kB 00:00:00
-----
Total | 23 MB/s | 7.8 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : git-core-2.32.0-1.amzn2.0.1.x86_64 1/7
Installing : git-core-doc-2.32.0-1.amzn2.0.1.noarch 2/7
Installing : perl-Error-0.17020-2.amzn2.noarch 3/7
Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 4/7
Installing : perl-git-2.32.0-1.amzn2.0.1.noarch 5/7
Installing : git-2.32.0-1.amzn2.0.1.x86_64 6/7
Verifying : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 7/7
Verifying : git-core-doc-2.32.0-1.amzn2.0.1.noarch 1/7
Verifying : perl-git-2.32.0-1.amzn2.0.1.noarch 2/7
Verifying : perl-Error-0.17020-2.amzn2.noarch 3/7
Verifying : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 4/7
Verifying : git-core-2.32.0-1.amzn2.0.1.x86_64 5/7
Verifying : perl-Error-0.17020-2.amzn2.noarch 6/7
Verifying : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 7/7
Installed:
git.x86_64 0:2.32.0-1.amzn2.0.1
Dependency Installed:
emacsfilesystem.noarch 1:27.2-4.amzn2.0.1 git-core.x86_64 0:2.32.0-1.amzn2.0.1 git-core-doc.noarch 0:2.32.0-1.amzn2.0.1
perl-Error.noarch 1:0.17020-2.amzn2 perl-git.noarch 0:2.32.0-1.amzn2.0.1 perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2
Complete!
[root@ip-172-31-23-236 ~]#
```

i-Ofed7363f6712b6d2 (Jenkins_CI_Server)
Public IP: 52.201.231.58 Private IP: 172.31.23.236

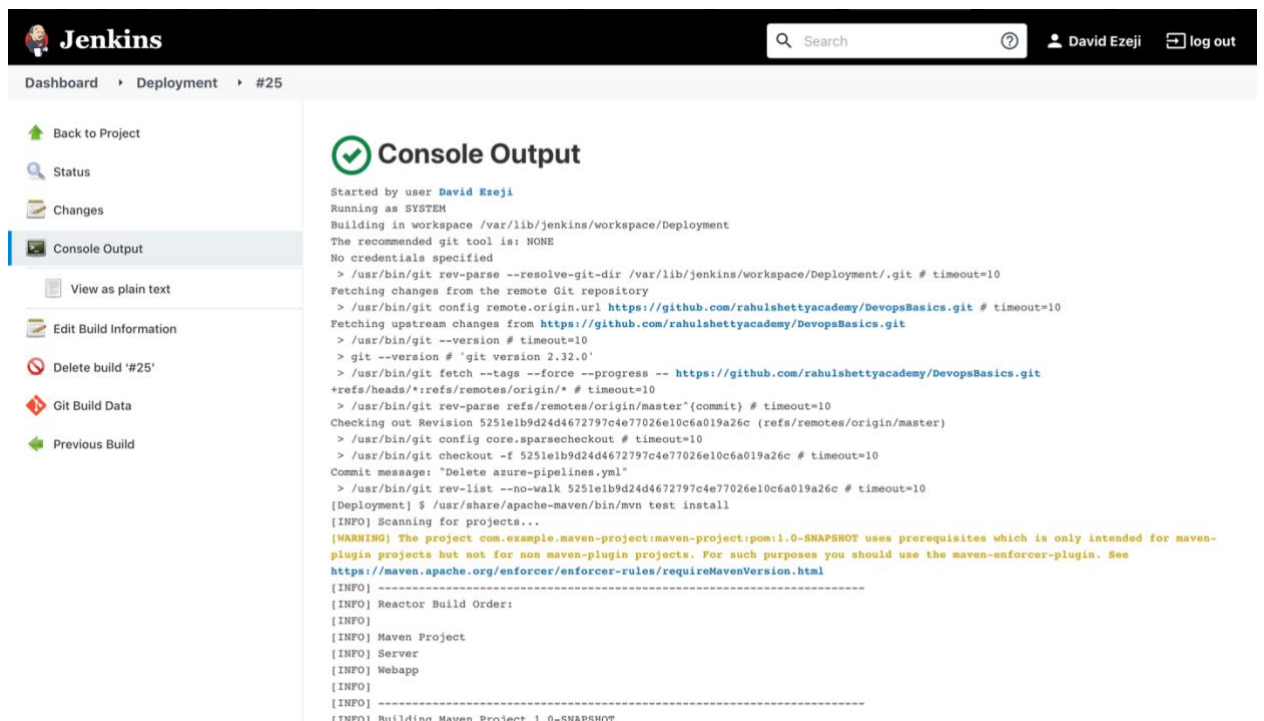
7. Created a new job for Jenkins that will automatically deploy apps. I then configured Jenkins using path variables.



8. Sourced code that will be built/tested from a Github public repository



9. Built the project



10. Jenkins project is now installed onto the ec2 instance and I have created a .war file

Dashboard » Deployment » #25

```
[INFO] --- maven-war-plugin:2.2:war (default-war) @ webapp ---
[WARNING] An illegal reflective access operation has occurred
[WARNING] Illegal reflective access by com.thoughtworks.xstream.core.util.Fields
(file:/var/lib/jenkins/.m2/repository/com/thoughtworks/xstream/xstream-1.3.1.jar) to field
java.util.Properties.defaults
[WARNING] Please consider reporting this to the maintainers of com.thoughtworks.xstream.core.util.Fields
[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
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in /var/lib/jenkins/workspace/Deployment/webapp/target/webapp
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/Deployment/webapp/src/main/webapp]
[INFO] Webapp assembled in [33 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/Deployment/webapp/target/webapp.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ webapp ---
[INFO] Installing /var/lib/jenkins/workspace/Deployment/webapp/target/webapp.war to
/var/lib/jenkins/.m2/repository/com/example/maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[INFO] Installing /var/lib/jenkins/workspace/Deployment/webapp/pom.xml to /var/lib/jenkins/.m2/repository/com/example/maven-
project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Maven Project ..... SUCCESS [ 0.556 s]
[INFO] Server ..... SUCCESS [ 2.045 s]
[INFO] Webapp ..... SUCCESS [ 1.090 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.953 s
[INFO] Finished at: 2021-10-11T14:53:52Z
[INFO] Final Memory: 11M/28M
[INFO] -----
Finished: SUCCESS
```

REST API Jenkins 2.315

```
[root@ip-172-31-23-236 Deployment]# ls -a
. . . AzureDockerfile Dockerfile .git pom.xml .project README.md server .settings webapp
[root@ip-172-31-23-236 Deployment]# cd webapp
[root@ip-172-31-23-236 webapp]# ls -a
. . . classpath .gitignore pom.xml .project .settings src target
[root@ip-172-31-23-236 webapp]# cd target
[root@ip-172-31-23-236 target]# ls -a
. . . maven-archiver surefire webapp webapp.war
[root@ip-172-31-23-236 target]#
```

i-0fed7363f6712b6d2 (Jenkins_CI_Server)

Public IPs: 52.201.231.58 Private IPs: 172.31.23.236

11. Spun up a second EC2 instance which will receive apps deployed from the first instance

Amazon Web Services (AWS) console showing EC2 instances.

Instances (1/2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Application_S...	i-04c6f71ef7ed09a6a	Running	t2.micro	2/2 checks passed		us-east-1a	ec2-
Jenkins_CI_Se...	i-0fed7363f6712b6e2	Running	t2.micro	2/2 checks passed		us-east-1b	ec2-

Instance: i-04c6f71ef7ed09a6a (Application_Server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-04c6f71ef7ed09a6a (Application_Server)	52.91.69.226 open address	172.31.55.92

IPv6 address

Instance state: Running

Public IPv4 DNS

ec2-52-91-69-226.compute-1.amazonaws.com | open address

Private IPv4 DNS

Elastic IP addresses

12. Connected the first instance (Jenkins_CI_Server) to the second instance (Application_Server) via Jenkins

Dashboard

configuration

Disable exec

SSH Servers

SSH Server

Name

Application_Server

Hostname

52.91.69.226

Username

ec2-user

Remote Directory

Advanced...

Test Configuration

Default

Add

Save

Apply

Advanced...

Dashboard

configuration

Timeout (ms)

300000

☐ Disable exec

Proxy type

1

Proxy host

Proxy port

Proxy user

Proxy password

Success

Test Configuration

Delete

Add

Save

Apply

13. Successfully deployed the application to the “Application_Server” instance by sending over the webapp.war file

The screenshot displays the Jenkins web interface for a project named 'DeploytoAppServer'. The left sidebar provides navigation options. The main area features a title 'Project DeploytoAppServer' and two primary sections: 'Workspace' and 'Recent Changes'. The 'Recent Changes' section lists five build events, each with a status icon (green for success, red for failure), a timestamp, and a link to the build details. At the bottom, a 'Permalinks' section contains a table with columns for build number, status, and timestamp, providing direct links to specific build pages.

Build Number	Status	Timestamp	Link
1	Success	Oct 11, 2021 7:03 PM	View build details
2	Success	Oct 11, 2021 7:03 PM	View build details
3	Success	Oct 11, 2021 7:03 PM	View build details
4	Success	Oct 11, 2021 7:03 PM	View build details
5	Success	Oct 11, 2021 7:03 PM	View build details

[illegible]

```
Last login: Mon Oct 11 19:02:15 2021 from ec2-18-206-107-24.compute-1.amazonaws.com

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-92 ~]$ sudo su -
Last login: Mon Oct 11 19:02:20 UTC 2021 on pts/1
Last failed login: Mon Oct 11 19:43:12 UTC 2021 from 200.70.56.204 on ssh:notty
There were 4 failed login attempts since the last successful login.
[root@ip-172-31-95-92 ~]# ls -la
.. .bash_logout .bash_profile .bashrc .chsrc .ssh .tcshrc
[ec2-user@ip-172-31-95-92 ~]# cd ..
[ec2-user@ip-172-31-95-92 ~]# ls -la
. .autorelabel bin boot dev etc home lib lib64 local media mnt opt proc root run sbin srv sys usr var
[ec2-user@ip-172-31-95-92 ~]# cd root
bash: cd: root: Permission denied
[ec2-user@ip-172-31-95-92 ~]# cd ..
[ec2-user@ip-172-31-95-92 ~]# ls -la
. .bash_history .bash_logout .bash_profile .bashrc .ssh webapp.war
[ec2-user@ip-172-31-95-92 ~]#
```

i-04c6f71ef7ed09a6a (Application_Server)
Public IP: 52.91.69.226 Private IP: 172.31.95.92

14. Installed Docker onto the "Application_Server" instance

```
(4/5): runc-1.0.0-2.amzn2.x86_64.rpm | 3.3 MB 00:00:00
(5/5): docker-20.10.7-3.amzn2.x86_64.rpm | 42 MB 00:00:01
-----
Total | 66 MB/s | 69 MB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : runc-1.0.0-2.amzn2.x86_64 1/5
Installing : containerd-1.4.6-3.amzn2.x86_64 2/5
Installing : libcgrouper-0.41-21.amzn2.x86_64 3/5
Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5
Installing : docker-20.10.7-3.amzn2.x86_64 5/5
Verifying : docker-20.10.7-3.amzn2.x86_64 1/5
Verifying : containerd-1.4.6-3.amzn2.x86_64 2/5
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64 3/5
Verifying : runc-1.0.0-2.amzn2.x86_64 4/5
Verifying : libcgrouper-0.41-21.amzn2.x86_64 5/5
Installed:
docker.x86_64 0:20.10.7-3.amzn2
Dependency Installed:
containerd.x86_64 0:1.4.6-3.amzn2 libcgrouper.x86_64 0:0.41-21.amzn2 pigz.x86_64 0:2.3.4-1.amzn2.0.1 runc.x86_64 0:1.0.0-2.amzn2
Complete!
[ec2-user@ip-172-31-95-92 ~]# usermod -aG docker ec2-user
[ec2-user@ip-172-31-95-92 ~]# sudo su ec2-user
[ec2-user@ip-172-31-95-92 ~]# cd ..
[ec2-user@ip-172-31-95-92 ~]# cd home
[ec2-user@ip-172-31-95-92 ~]# cd ec2-user
[ec2-user@ip-172-31-95-92 ~]# sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-95-92 ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[ec2-user@ip-172-31-95-92 ~]#
```

i-04c6f71ef7ed09a6a (Application_Server)
Public IP: 52.91.69.226 Private IP: 172.31.95.92

15. Created container from Tomcat server Image for the sake of deployment

```
Last login: Tue Oct 12 19:16:52 2021 from ec2-18-206-107-24.compute-1.amazonaws.com

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-92 ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
tomcat latest 6313f84af805 6 days ago 680MB
[ec2-user@ip-172-31-95-92 ~]# docker run -d -p 8080:8080 --name tomcat-server 6313f84af805
6da0361f0c7c6ff2a44c91677c052d7642301270152d7158e1e06e1a2786c502
[ec2-user@ip-172-31-95-92 ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6da0361f0c7c 6313f84af805 "catalina.sh run" 10 seconds ago Up 8 seconds 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp tomcat-server
[ec2-user@ip-172-31-95-92 ~]#
```

i-04c6f71ef7ed09a6a (Application_Server)
Public IP: 52.91.69.226 Private IP: 172.31.95.92

16. Moved the webapps.war file (this file exists on the Linux instance) into the webapps folder that is in the tomcat container.

```
Last login: Tue Oct 12 20:19:10 2021 from ec2-18-206-107-25.compute-1.amazonaws.com

 _ _ | _ | _ |
 _ _ | _ | _ |

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-92 ~]$ ls -la
. . . .bash_history .bash_logout .bash_profile .bashrc .ssh webapp.war
[ec2-user@ip-172-31-95-92 ~]$ sudo docker cp webapp.war tomcat-container:/usr/local/tomcat/webapps/webapps.war
[ec2-user@ip-172-31-95-92 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
4fa6abf89865   6313f84af805   "catalina.sh run"       37 minutes ago Up 37 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   tomcat-container
[ec2-user@ip-172-31-95-92 ~]$ docker exec -it tomcat-container/bin/bash
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
[ec2-user@ip-172-31-95-92 ~]$ docker exec -it tomcat-container /bin/bash
root@4fa6abf89865:/usr/local/tomcat# ls -la
. BUILDING.txt LICENSE README.md RUNNING.txt conf logs temp webapps.dist
.. CONTRIBUTING.md NOTICE RELEASE-NOTES lib native-jni-lib webapps work
root@4fa6abf89865:/usr/local/tomcat# cd webapps
root@4fa6abf89865:/usr/local/tomcat/webapps# ls -la
. . . ROOT docs examples host-manager manager webapps webapps.war
root@4fa6abf89865:/usr/local/tomcat/webapps#
```

i-04c6f71ef7ed09a6a (Application_Server)

Public IPs: 52.91.69.226 Private IPs: 172.31.95.92

17. The web application is successfully deployed and readily available on the “Application_Server” instance

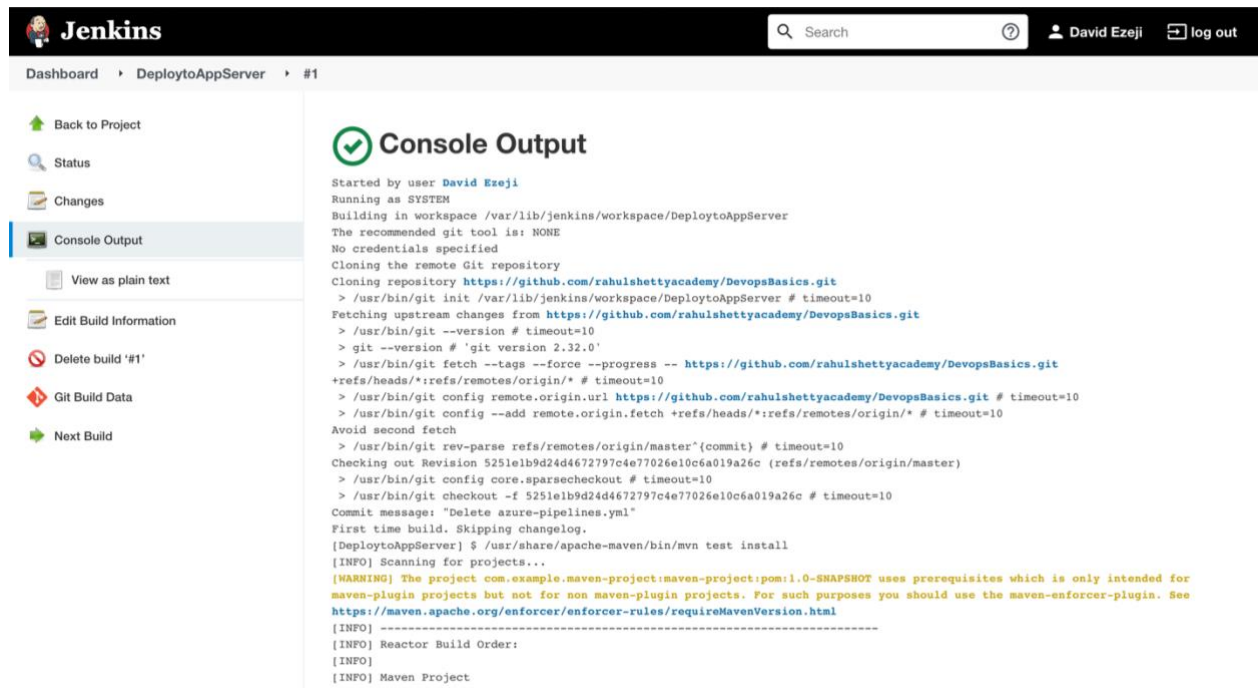
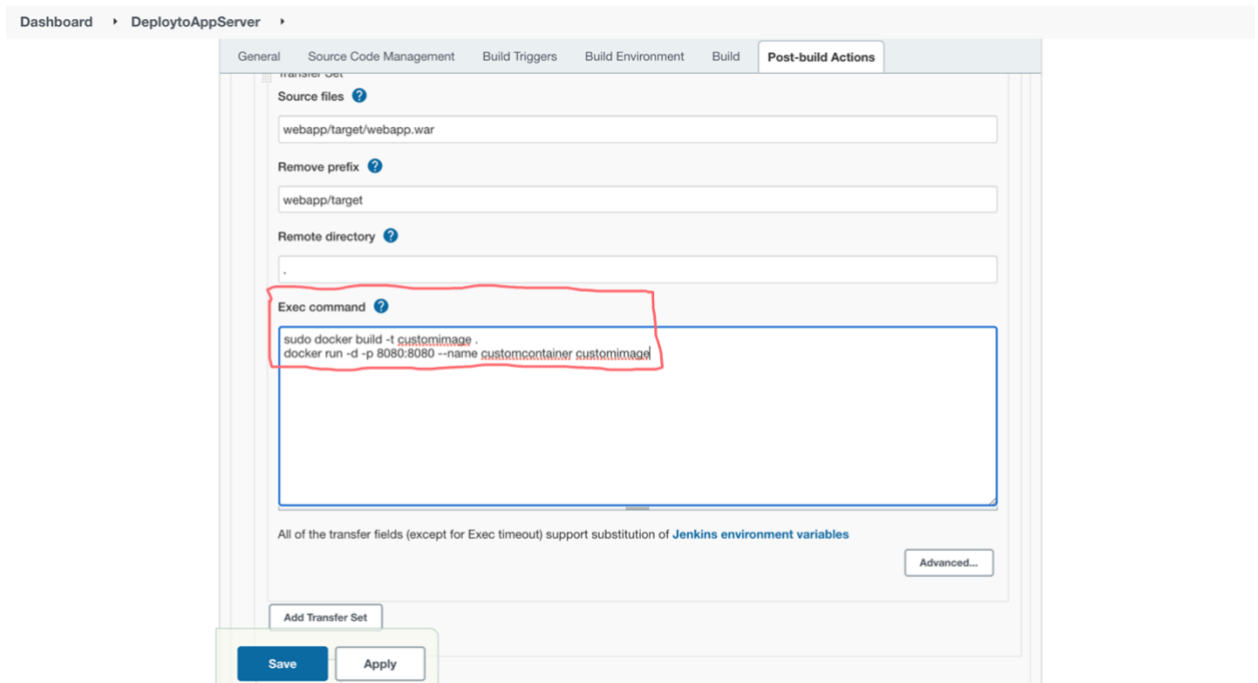
The screenshot displays the AWS Management Console interface. On the left, the navigation menu includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'Elastic Block Store'. The main content area shows the 'Instances (1/2)' page with a table of running instances. The first instance, 'Application_Server' (ID: i-04c6f71ef7ed09a6a), is highlighted. Below the table, the details for this instance are expanded, showing its public IP address (52.91.69.226) and private IP address (172.31.95.92). The instance is running on a t2.micro instance type in the us-east-1a availability zone.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Application_Server	i-04c6f71ef7ed09a6a	Running	t2.micro	2/2 checks passed		us-east-1a
Jenkins_CI_Server	i-0fed7363f6712b6d2	Running	t2.micro	2/2 checks passed		us-east-1b

Instance: i-04c6f71ef7ed09a6a (Application_Server)

Instance summary	Info
Instance ID	i-04c6f71ef7ed09a6a (Application_Server)
Public IPv4 address	52.91.69.226 (open address)
Private IPv4 address	172.31.95.92
Instance state	Running
Instance type	t2.micro
Private IPv4 DNS	ip-172-31-95-92.ec2.internal

19. Used exec commands in Jenkins to automate the process of building a docker image and container from the dockerfile



20. Attached Github hook from Jenkins to payload URL in my Github repository to allow for continuous integration to happen (allows for changes in code on Github to automatically trigger a build in Jenkins)

← → ↻ 52.201.231.58:8080/configure ☆ ⓘ ⋮

Dashboard ▸ configuration

Lockable Resources

Add Lockable Resource

GitHub

GitHub Servers

Add GitHub Server

Re-register hooks for all jobs

Override Hook URL

☒ Specify another hook URL for GitHub configuration

Shared secrets

Add shared secret

Additional actions

Manage additional GitHub actions

GitHub API usage

GitHub API usage rate limiting strategy

Normalize API requests

GitHub Enterprise Servers

...

Save Apply

← → ↻ https://github.com/davidezij/RahulShettyDevops/settings/hooks/new ☆ ⓘ ⋮

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Pages

Moderation settings

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *

Content type

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

← → ↻ 52.201.231.58:8080/job/Deployment/configure ☆ ⓘ ⋮

Dashboard ▸ Deployment ▸

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Additional Behaviours

Add

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☐ Build after other projects are built

☐ Build periodically

☒ GitHub hook trigger for GITScm polling

☐ Poll SCM

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Send files or execute commands over SSH before the build starts

☐ Send files or execute commands over SSH after the build runs

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ Inspect build log for published Gradle build scans

☐ With Ant

Build

☐ Invoke top-level Maven targets

Maven Version

Save Apply