

DevOps Foundations: Version Control and CI/CD with Jenkins

Course-End Project 1: Insured Assurance

Objective

To create a GitHub Actions CI/CD pipeline workflow for invoking the deployment of a Java application as a Jenkins job using Tomcat Apache

Real-time scenario:

Insured Assurance, a leading global insurance provider based in the US, offers a range of products including home, health, car, and life insurance. The company is transitioning to a DevOps architecture and aims to automate code builds and deployments across various environments. To meet this need, it has adopted GitHub Actions for code checkout, building, and testing automation and Jenkins for continuous deployment. As a DevOps engineer at Insured Assurance, you are tasked with implementing a CI/CD pipeline using GitHub Actions and Jenkins.

Tasks

The following tasks outline the process of implementing CI/CD using GitHub Actions and Jenkins:

- 1. Create a code repository on GitHub*
- 2. Create a GitHub Actions pipeline to perform continuous integration*
- 3. Configure Tomcat Apache for automated code deployment*
- 4. Integrate the GitHub Actions pipeline to invoke the Jenkins pipeline*
- 5. Invoke pipeline to validate automated deployment*

Solution:

Task 1: Create a code repository on GitHub

Forking repository from <https://github.com/demo1orgtoday/MavenBuild-SL.git>

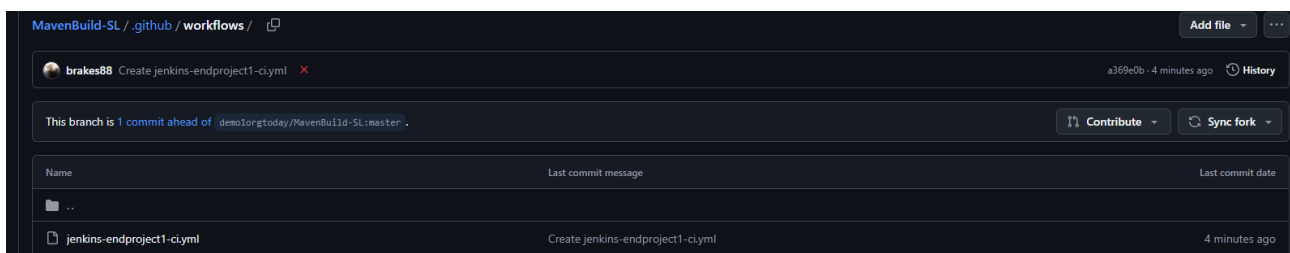
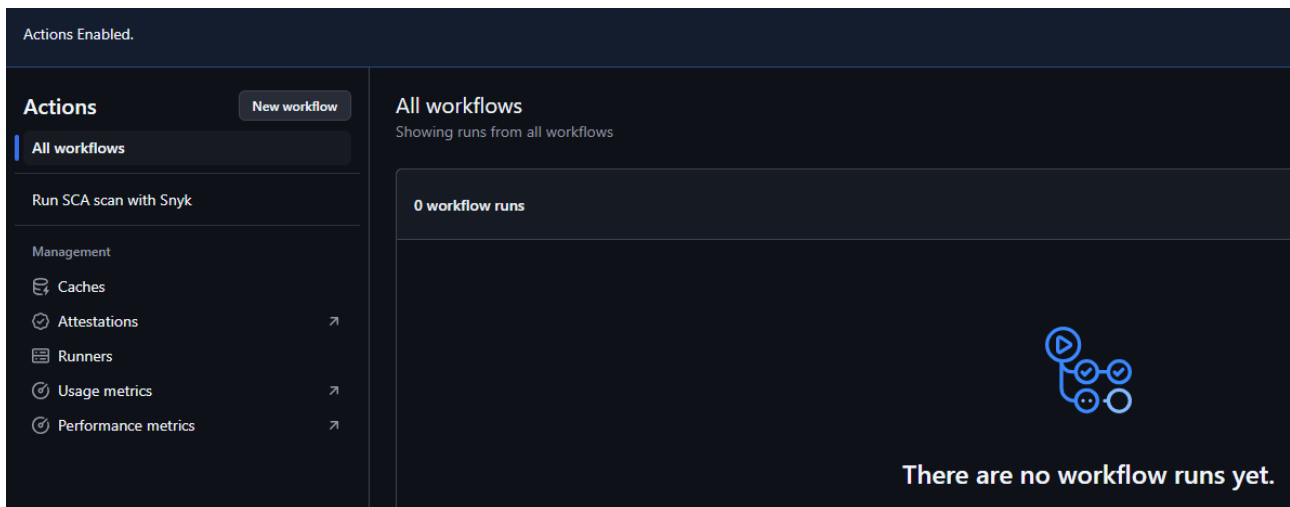
The screenshot shows a GitHub repository page for 'MavenBuild-SL' (Public). It is a fork of 'demo1orgtoday/MavenBuild-SL'. The repository has 1 branch (master) and 0 tags. The main branch is up to date with the upstream repository. The repository contains 456 commits by Sonal0409, last updated 2 days ago. The file list includes: .github/workflows, src, webapp, Dockerfile, Jenkinsfile, Jenkinsfile-SL, README.md, deploy.yml, pom.xml, and sonar-project.properties. The right sidebar shows the 'About' section with no description, website, or topics provided. It also lists 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (Java 83.2%, Smarty 15.7%, Dockerfile 1.1%).

File	Update	Time
.github/workflows	Update main.yml	2 days ago
src	Update index.jsp	2 years ago
webapp	done helm charts	last year
Dockerfile	Update Dockerfile	last year
Jenkinsfile	Update Jenkinsfile	last year
Jenkinsfile-SL	Rename Jenkinsfile to Jenkinsfile-SL	last year
README.md	Update README.md	2 years ago
deploy.yml	Update deploy.yml	2 years ago
pom.xml	Update pom.xml	8 months ago
sonar-project.properties	Create sonar-project.properties	3 years ago

Riccardo Braga
Braric88@gmail.com

Task 2: Create a GitHub Actions pipeline to perform continuous integration

Created new empty workflow



Initial GitHub Action Code:

```
name: Jenkins-EndProject1-CI
on:
  push: # run when there is commit to repo
  workflow_dispatch: # run manually
jobs:
  CICDjob:
    runs-on: ubuntu-latest
    steps:
      - name: Clone the repo on ubuntu server
        uses: actions/checkout@v4
      - name: Install Java and maven on ubuntu server
        uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
          java-version: '11'
          cache: 'maven'
      - name: Build the code
        run: mvn package
```

← Jenkins-EndProject1-CI

✓ Create jenkins-endproject1-ci.yml #1

Summary

Jobs

✓ CICDjob

Run details

Usage

Workflow file

CICDjob

succeeded now in 13s

> ✓ Set up job

> ✓ Clone the repo on ubuntu server

> ✓ Install Java and maven on ubuntu server

> ✓ Build the code

> ✓ Post Install Java and maven on ubuntu server

> ✓ Post Clone the repo on ubuntu server

> ✓ Complete job

Task 3: Configure Tomcat Apache for automated code deployment

Installed tomcat9 in lab

```
root@ip-172-31-42-1: ~
File Edit View Search Terminal Help
root@ip-172-31-42-1:~# apt update && apt install -y tomcat9 tomcat9-admin
Get:1 https://packages.microsoft.com/repos/azure-cli jammy InRelease [3596 B]
Get:2 https://packages.microsoft.com/repos/code stable InRelease [3590 B]
Get:3 https://packages.microsoft.com/repos/azure-cli jammy/main arm64 Packages [2347 B]
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [129 kB]
Hit:5 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:6 https://packages.microsoft.com/repos/code stable/main amd64 Packages [19.1 kB]
Get:7 https://packages.microsoft.com/repos/code stable/main armhf Packages [19.4 kB]
Get:8 https://packages.microsoft.com/repos/code stable/main arm64 Packages [19.3 kB]
Get:9 http://ap-south-1a.clouds.ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [128 kB]
Hit:10 https://ppa.launchpadcontent.net/pipewire-debian/pipewire-upstream/ubuntu jammy InRelease
Get:11 https://baltocdn.com/helm/stable/debian all InRelease [7652 B]
Get:12 http://ports.ubuntu.com/ubuntu-ports jammy-security/main arm64 Packages [2075 kB]
```

Updated user information in /etc/tomcat9/tomcat-users.xml

```
root@ip-172-31-42-1: ~
File Edit View Search Terminal Help
- manager-jmx - allows access to the JMX proxy and the status pages
- manager-status - allows access to the status pages only

The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!-- ..> that surrounds them. You
will also need to set the passwords to something appropriate.
-->
<!--
<user username="admin" password="<must-be-changed>" roles="manager-gui"/>
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- ..> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<user username="tomcat" password="password" roles="admin-gui,manager-gui,manager-script"/>
</tomcat-users>
~
~
~
:wq
```

Riccardo Braga
Braric88@gmail.com

Changed tomcat port in /etc/tomcat9/server.xml

```
root@ip-172-31-42-1: ~
File Edit View Search Terminal Help

<!--The connectors can use a shared executor, you can define one or more named thread pools-->
<!--
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
      maxThreads="150" minSpareThreads="4"/>
-->

<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html
Java AJP  Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="9090" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
      port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
-->
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation. The default
SSLImplementation will depend on the presence of the APR/native
library and the useOpenSSL attribute of the AprLifecycleListener.
Either JSSE or OpenSSL style configuration may be used regardless of
: wq
```

Added java versions in /usr/libexec/tomcat9/tomcat-locate-java.sh

```
root@ip-172-31-42-1: ~
File Edit View Search Terminal Help

#
# Script looking for a Java runtime suitable for running Tomcat
#
# The script looks for the default JRE/JDK, OpenJDK and Oracle JDK
# as packaged by java-package. The Java runtime found is exported
# in the JAVA_HOME environment variable.
#

set -e

# Find the Java runtime if JAVA_HOME isn't already defined
if [ -z "$JAVA_HOME" ]; then
    # This function sets the variable JDK_DIRS
    find_jdks()
    {
        for java_version in 21 17 11 10 9 8
        do
            for jvmdir in /usr/lib/jvm/java-${java_version}-openjdk-* \
                /usr/lib/jvm/jdk-${java_version}-oracle-* \
                /usr/lib/jvm/jre-${java_version}-oracle-* \
                /usr/lib/jvm/java-${java_version}-oracle \
                /usr/lib/jvm/oracle-java${java_version}-jdk-* \
                /usr/lib/jvm/oracle-java${java_version}-jre-*
            do
                if [ -d "${jvmdir}" ]
                then
                    JDK_DIRS="${JDK_DIRS} ${jvmdir}"
                fi
            done
        done
    }
: wq
```

Riccardo Braga
Braric88@gmail.com

Restarted tomcat service

```
root@ip-172-31-42-1:~# systemctl restart tomcat9.service
root@ip-172-31-42-1:~# systemctl status tomcat9.service
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-05-13 14:21:42 UTC; 7s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Process: 21839 ExecStartPre=/usr/libexec/tomcat9/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
   Main PID: 21843 (java)
     Tasks: 38 (limit: 18910)
    Memory: 239.7M
       CPU: 7.851s
   CGroup: /system.slice/tomcat9.service
           └─21843 /usr/lib/jvm/java-21-openjdk-arm64/bin/java -Djava.util.logging.config.file=

May 13 14:21:45 ip-172-31-42-1 tomcat9[21843]: Deployment of deployment descriptor [/etc/tomcat9/conf/Catalina_localhost/localhost.xml]
May 13 14:21:45 ip-172-31-42-1 tomcat9[21843]: Deploying deployment descriptor [/etc/tomcat9/conf/Catalina_localhost/localhost.xml]
May 13 14:21:45 ip-172-31-42-1 tomcat9[21843]: The path attribute with value [/manager] in deployment descriptor is deprecated
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: At least one JAR was scanned for TLDs yet contained no TLDs
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: Deployment of deployment descriptor [/etc/tomcat9/conf/Catalina_localhost/localhost.xml]
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: Deploying web application directory [/var/lib/tomcat9/webapps/localhost/]
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: At least one JAR was scanned for TLDs yet contained no TLDs
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: Deployment of web application directory [/var/lib/tomcat9/webapps/localhost/]
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: Starting ProtocolHandler ["http-nio-9090"]
May 13 14:21:46 ip-172-31-42-1 tomcat9[21843]: Server startup in [3152] milliseconds
lines 1-22/22 (END)...skipping...
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-05-13 14:21:42 UTC; 7s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
   Process: 21839 ExecStartPre=/usr/libexec/tomcat9/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
   Main PID: 21843 (java)
     Tasks: 38 (limit: 18910)
    Memory: 239.7M
       CPU: 7.851s
```



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`.

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking [here](#).

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.

Added Deploy to container Jenkins plugin

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SSH server

✔ Success

Deploy to container

✔ Success

Loading plugin extensions

✔ Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

Riccardo Braga
Braric88@gmail.com

Added tomcat user

New credentials

Kind

Username with password

Scope ?

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

Username ?

tomcat

☐ Treat username as secret ?

Password ?

●●●●●●●●

ID ?

tomcat-user

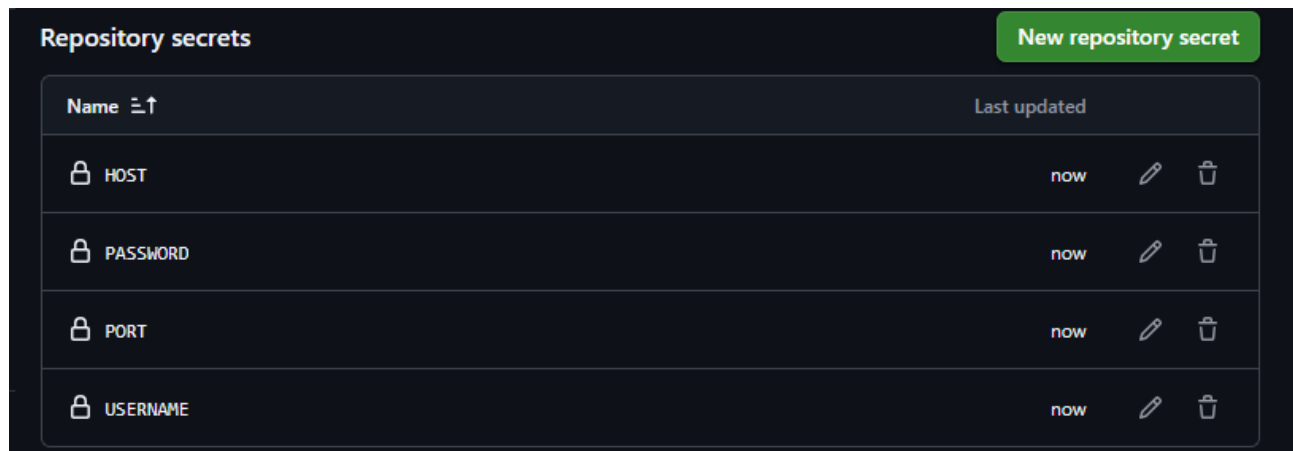
Description ?

tomcat-user

Create

Task 4: Integrate the GitHub Actions pipeline to invoke the Jenkins pipeline

Added secrets for lab connection



Updated action with .war copy and trigger for Jenkins job

```
name: Jenkins-EndProject1-CICD
on:
  push: # run when there is commit to repo
  workflow_dispatch: # run manually
jobs:
  CICDjob:
    runs-on: ubuntu-latest
    steps:
      - name: Clone the repo on ubuntu server
        uses: actions/checkout@v4
      - name: Install Java and maven on ubuntu server
        uses: actions/setup-java@v4
        with:
          distribution: 'temurin'
          java-version: '11'
          cache: 'maven'
      - name: Build the code
        run: mvn package
      - name: connect to Lab and deploy code on tomcat9
        uses: cross-the-world/ssh-scp-ssh-pipelines@latest
        with:
          host: ${secrets.HOST}
          user: ${secrets.USERNAME}
          pass: ${secrets.PASSWORD}
          port: ${secrets.PORT}
          connect_timeout: 10s
          first_ssh: |
            sudo chmod 777 /var/lib/tomcat9/webapps
          scp: |
            './target/*.war' => /var/lib/tomcat9/webapps
          last_ssh: |
            sudo systemctl restart tomcat9
      - name: Connect to Lab machine and trigger a Jenkins job
        uses: cross-the-world/ssh-scp-ssh-pipelines@latest
        with:
          host: ${secrets.HOST}
          user: ${secrets.USERNAME}
          pass: ${secrets.PASSWORD}
          port: ${secrets.PORT}
          connect_timeout: 10s
          first_ssh: |
```

Riccardo Braga
Braric88@gmail.com

```
curl -l -u admin:Root123$ http://localhost:8080/job/Jenkins-EndProject2/build?token=tomcatDeploy
```

Added Maven Tool

Maven installations

Add Maven

Maven

Name

mymaven

Required

☒ Install automatically ?

Install from Apache

Version

3.9.9

Add Installer

Add Maven

Create Jenkins pipeline

jenkins-EndProject2

» Required field



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

OK

Build Triggers

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Quiet period ?
- ☒ Trigger builds remotely (e.g., from scripts) ?

Authentication Token

tomcatDeploy

Use the following URL to trigger build remotely: JENKINS_URL/job/Jenkins-EndProject2/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME
Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

Pipeline

Definition

Pipeline script

Script ?

```
13      }
14      }
15      }
16      }
17      stage('Build and Deploy the Code')
18      {
19      +
20      +
21      +
22      +
23      +
24      +
25      +
26      +
27      +
28      +
29      +
30      +
31      }
```

try sample Pipeline... v

- ☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save

Apply

Jenkins Pipeline

```
pipeline{
    agent any

    tools{
        maven 'mymaven'
    }

    stages{
        stage('checkout code')
        {
            steps{
                git 'https://github.com/Sonal0409/DevOpsCodeDemo.git'
            }
        }

        stage('Build and Deploy the Code')
        {
            steps{
                sh 'mvn package'
            }

            post{
                success{
                    deploy adapters: [tomcat9(credentialsId: 'tomcat-user', path: '', url:
'http://localhost:9090')], contextPath: null, war: '**/*.war'
                }
            }
        }
    }
}
```

}
}
}

Task 5: Invoke pipeline to validate automated deployment

```
1 ▶ Run cross-the-world/ssh-scp-ssh-pipeline@latest
13 /usr/bin/docker run --name fe9c1893b8416bac366734666c8592_d89267 --label 859912 --workdir /github/workspace --rm -e "JAVA_HOME" -e "JAVA_HOME_11_X64" -e "INPUT_HOST" -e "INPUT_USER" -e "INPUT_PASS" -e "INPUT_PORT"
-e "INPUT_CONNECT_TIMEOUT" -e "INPUT_FIRST_SSH" -e "INPUT_KEY" -e "INPUT_SCP" -e "INPUT_LAST_SSH" -e "HOME" -e "GITHUB_JOB" -e "GITHUB_REF" -e "GITHUB_SHA" -e "GITHUB_REPOSITORY" -e "GITHUB_REPOSITORY_OWNER" -e
"GITHUB_REPOSITORY_OWNER_ID" -e "GITHUB_RUN_ID" -e "GITHUB_RUN_NUMBER" -e "GITHUB_RETENTION_DAYS" -e "GITHUB_RUN_ATTEMPT" -e "GITHUB_ACTOR_ID" -e "GITHUB_ACTOR" -e "GITHUB_WORKFLOW" -e "GITHUB_HEAD_REF" -e
"GITHUB_BASE_REF" -e "GITHUB_EVENT_NAME" -e "GITHUB_SERVER_URL" -e "GITHUB_API_URL" -e "GITHUB_GRAPHQL_URL" -e "GITHUB_REF_NAME" -e "GITHUB_REF_PROTECTED" -e "GITHUB_REF_TYPE" -e "GITHUB_WORKFLOW_REF" -e
"GITHUB_WORKFLOW_SHA" -e "GITHUB_REPOSITORY_ID" -e "GITHUB_TRIGGERING_ACTOR" -e "GITHUB_WORKSPACE" -e "GITHUB_ACTION" -e "GITHUB_EVENT_PATH" -e "GITHUB_ACTION_REPOSITORY" -e "GI
14 *****STARTING PIPELINES*****
15 *****Pipeline: RUNNING FIRST SSH*****
16 curl -l -u admin:Root123$ http://localhost:8080/job/Jenkins-EndProject2/build?token=token-tomcatDeploy
17 Error:
18 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
19                               Dload  Upload  Total   Spent    Left   Speed
20   0   0   0    0    0    0     0     0  --:--:-- --:--:-- --:--:--    0
21   0   0   0    0    0    0     0     0  --:--:-- --:--:-- --:--:--    0
22 SSH-SCP-SSH no scp input found
23 SSH-SCP-SSH no last_ssh input found
24 *****END PIPELINES*****
```



Permalinks

- [Last build \(#4\), 20 sec ago](#)
- [Last stable build \(#4\), 20 sec ago](#)
- [Last successful build \(#4\), 20 sec ago](#)
- [Last failed build \(#3\), 3 min 15 sec ago](#)
- [Last unsuccessful build \(#3\), 3 min 15 sec ago](#)
- [Last completed build \(#4\), 20 sec ago](#)

Riccardo Braga
Braric88@gmail.com

Filter contacts...			New contact
First Name	Last Name	Email	
George	White	george@white.com	
Daniel	Thompson	daniel@thompson.com	
Timothy	Jones	timothy@jones.com	
Peter	Wilson	peter@wilson.com	
Dan	Robinson	dan@robinson.com	
Dan	Davis	dan@davis.com	
Olivia	Davis	olivia@davis.com	
Dan	Smith	dan@smith.com	
Daniel	Anderson	daniel@anderson.com	
Alice	Thomas	alice@thomas.com	
Linda	Harris	linda@harris.com	
Daniel	Robinson	daniel@robinson.com	
Mike	Young	mike@young.com	
Umberto	Anderson	umberto@anderson.com	
Scott	Thompson	scott@thompson.com	
Rene	Martin	rene@martin.com	
Lisa	Martin	lisa@martin.com	
Peter	Martin	peter@martin.com	
Brian	Wilson	brian@wilson.com	
Scott	Miller	scott@miller.com	
John	White	john@white.com	