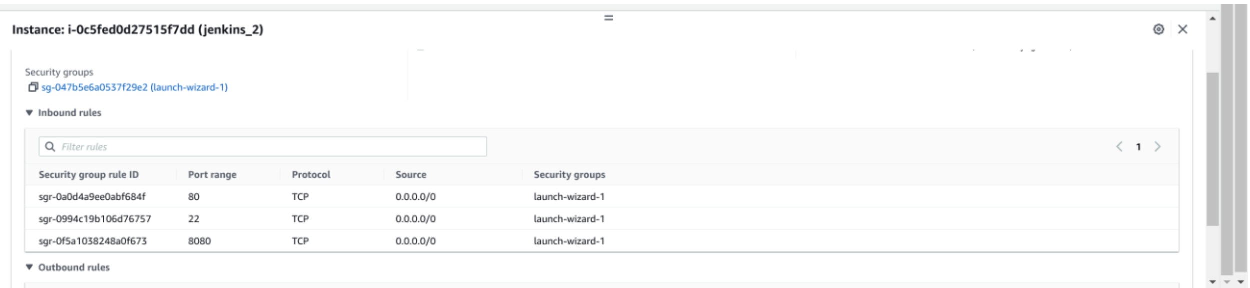
kuralabs_deployment_2

This is a simple CI/CD pipeline set up from start to finish and deploy an app to Elastic Beanstalk.

Create a Jenkins server

- ☐ Launch an AWS Ubuntu Instance that listens on ports 22, 80 and 8080
- ☐ Install Jenkins
- ☐ Activate the Jenkins user and grant sudoers permissions



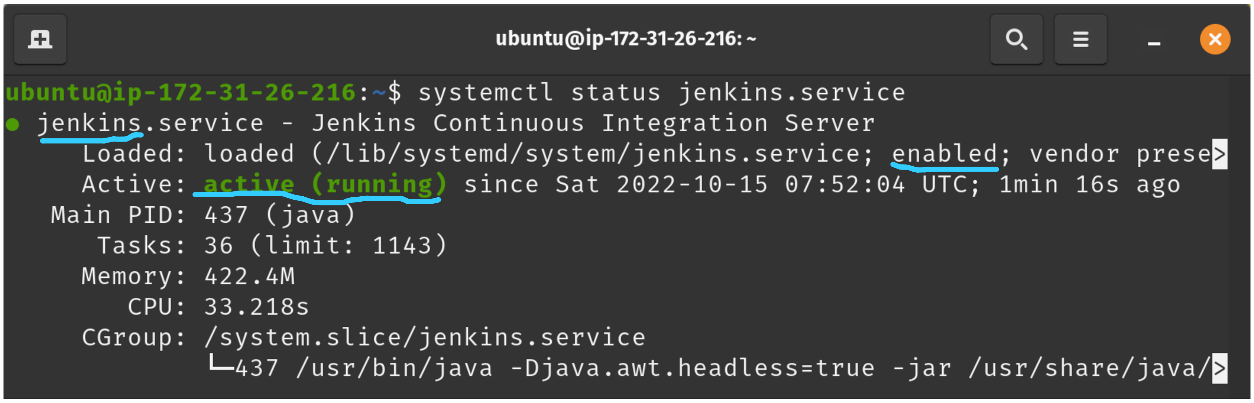
Instance: i-0c5fed0d27515f7dd (jenkins_2)

Security groups
sg-047b5e6a0537f29e2 (launch-wizard-1)

▼ Inbound rules

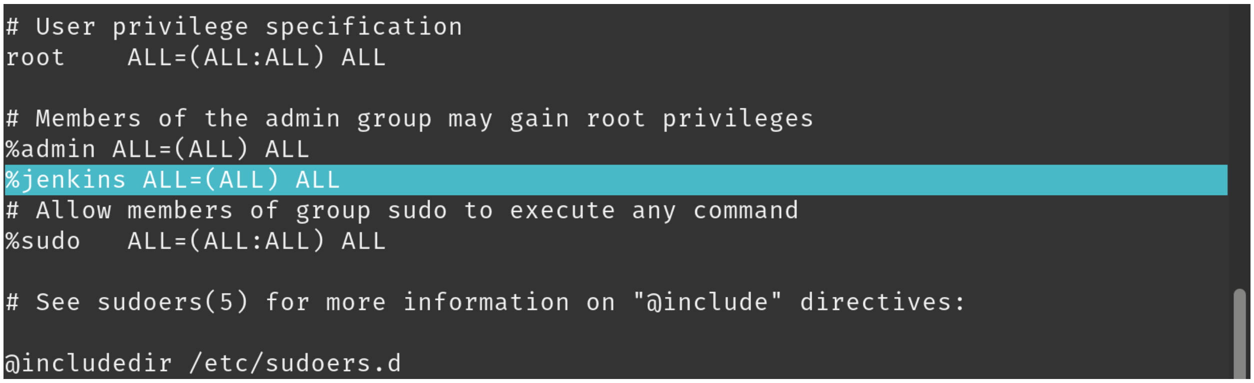
Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0a0d4a9ee0abf684f	80	TCP	0.0.0.0/0	launch-wizard-1
sgr-0994c19b106d76757	22	TCP	0.0.0.0/0	launch-wizard-1
sgr-0f5a1038248a0f673	8080	TCP	0.0.0.0/0	launch-wizard-1

▼ Outbound rules



ubuntu@ip-172-31-26-216: ~

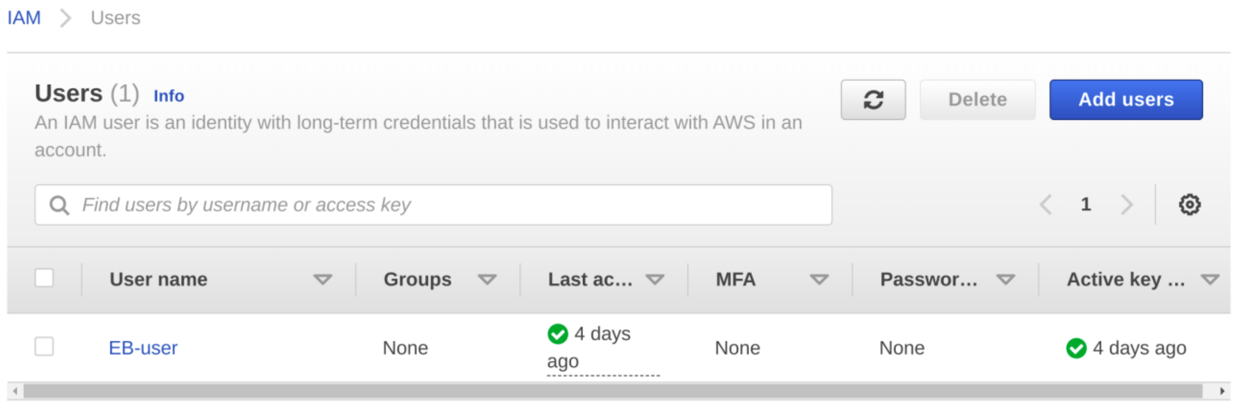
```
ubuntu@ip-172-31-26-216:~$ systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
     Active: active (running) since Sat 2022-10-15 07:52:04 UTC; 1min 16s ago
   Main PID: 437 (java)
      Tasks: 36 (limit: 1143)
     Memory: 422.4M
        CPU: 33.218s
     CGroup: /system.slice/jenkins.service
             └─437 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/>
```



```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
%jenkins ALL=(ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

Create a Jenkins user in your AWS account

- □ Type IAM in the AWS console and select Users and select Add users
- □ Enter the user name “EB-user” and then select Programmatic access
- □ Hit Next, and select “Attach existing policies directly” and check the box labeled “AdministratorAccess”
- □ Create the user and then copy and save the “access key ID” and the secret access key”



IAM > Users

Users (1) Info

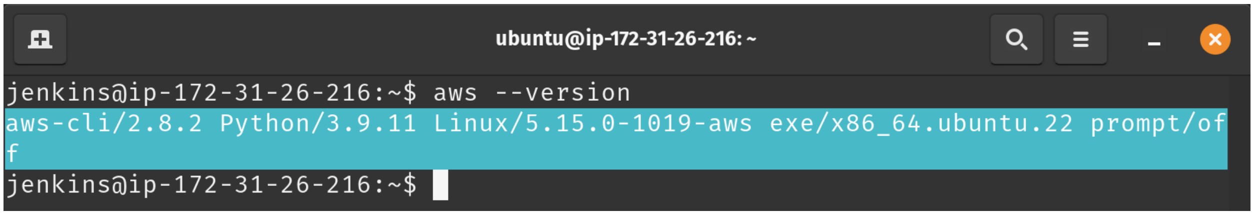
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Groups	Last ac...	MFA	Passwor...	Active key ...
□	EB-user	None	✓ 4 days ago	None	None	✓ 4 days ago

Install AWS CLI on the Jenkins Server EC2 and Configure

- □ Install the unzip package
  ``` sudo apt install unzip -y ```

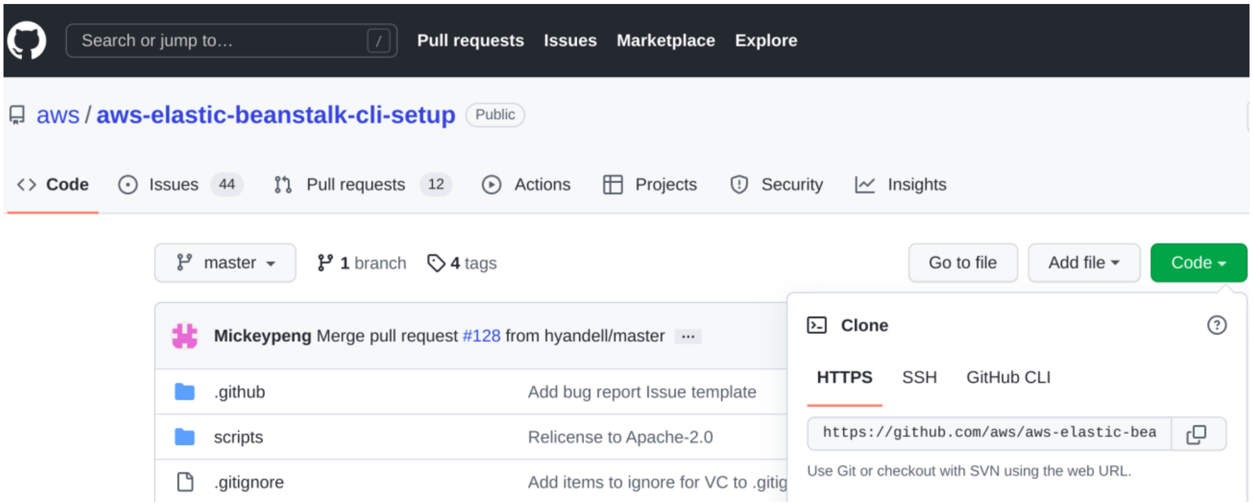
- □ View/Follow the steps here
  https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html



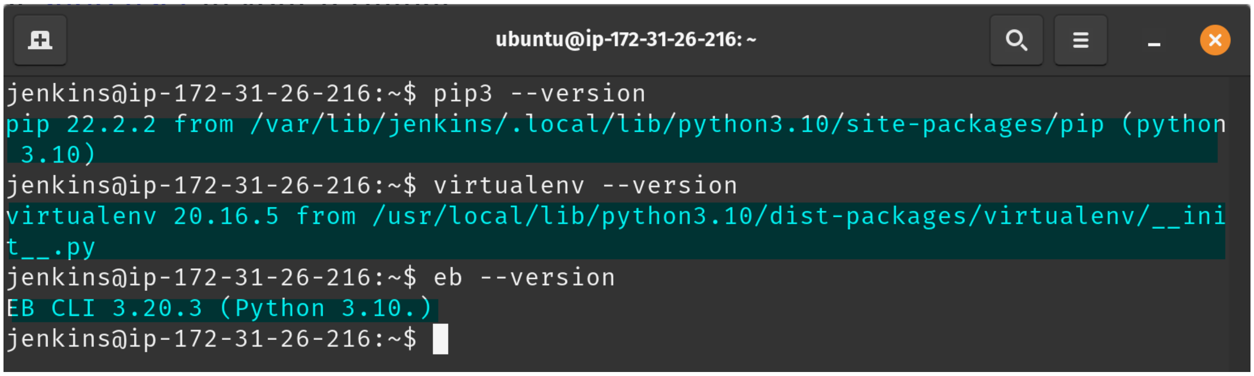
```
jenkins@ip-172-31-26-216:~$ aws --version
aws-cli/2.8.2 Python/3.9.11 Linux/5.15.0-1019-aws exe/x86_64.ubuntu.22 prompt/of
f
jenkins@ip-172-31-26-216:~$
```

Install EB CLI in the Jenkins Server EC2 user

- □ Install Python Pip https://linuxize.com/post/how-to-install-pip-on-ubuntu-20.04/
- □ Install virtualenv
- □ Install the EB CLI using the setup script here https://github.com/aws/aws-elastic-beanstalk-cli-setup



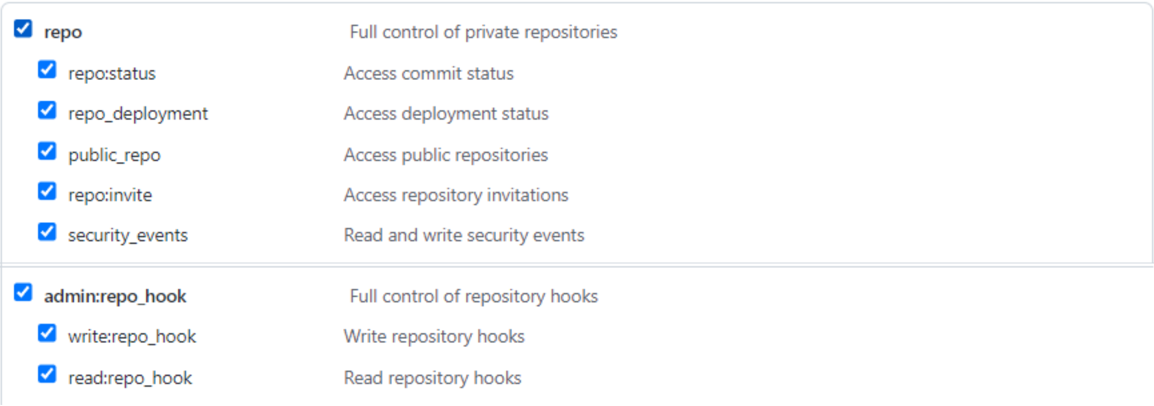
Search or jump to...
Pull requests
Issues
Marketplace
Explore
aws / aws-elastic-beanstalk-cli-setup
Public
Code
Issues 44
Pull requests 12
Actions
Projects
Security
Insights
master
1 branch
4 tags
Go to file
Add file
Code
Mickeypeng Merge pull request #128 from hyandell/master
.github
Add bug report Issue template
scripts
Relicense to Apache-2.0
.gitignore
Add items to ignore for VC to .gitig
Clone
HTTPS
SSH
GitHub CLI
https://github.com/aws/aws-elastic-bea
Use Git or checkout with SVN using the web URL.



```
jenkins@ip-172-31-26-216:~$ pip3 --version
pip 22.2.2 from /var/lib/jenkins/.local/lib/python3.10/site-packages/pip (python
 3.10)
jenkins@ip-172-31-26-216:~$ virtualenv --version
virtualenv 20.16.5 from /usr/local/lib/python3.10/dist-packages/virtualenv/__ini
t__.py
jenkins@ip-172-31-26-216:~$ eb --version
EB CLI 3.20.3 (Python 3.10.)
jenkins@ip-172-31-26-216:~$
```

Connect GitHub to Jenkins Server

- □ Create an access token in GitHub with repo and admin:repohook scopes, copy and save your credentials, your going to need this when setting up your CI/CD pipeline in the next step
- □ Fork the Deployment repo to your local server



☑ repo	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
☑ admin:repo_hook	Full control of repository hooks
☑ write:repo_hook	Write repository hooks
☑ read:repo_hook	Read repository hooks

Create a Multibranch Build

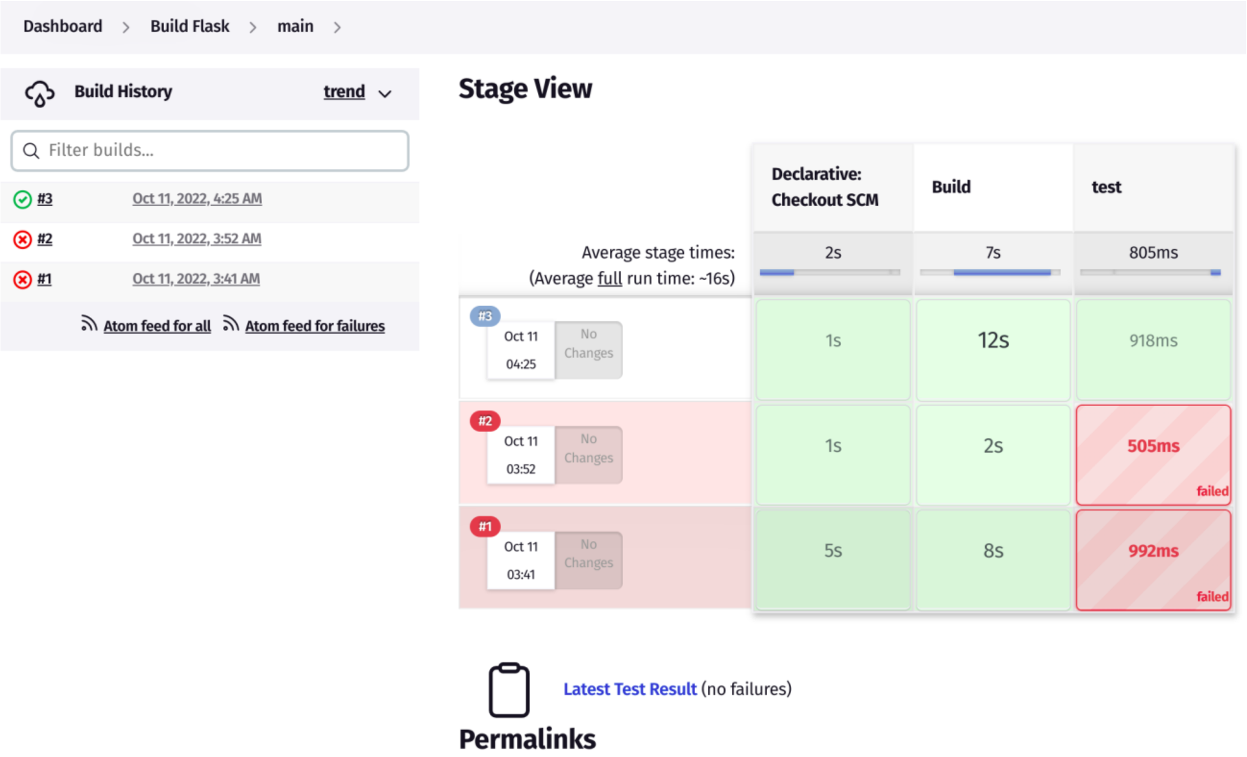
- □ Log into Jenkins GUI and select "New Item" and enter the name of your item
- □ Select Multibranch Pipeline and complete the basic information (i.e. Display Name and Description
- □ Add a Branch Source (GitHub from the drop-down) and use the use your GitHub user/access token as your username/password
- □ Complete the remaining items in the form, select apply then save
- □ If you don't see a build initiating, select Scan Repository in the left-hand side of the screen

Debugging/Build Troubleshooting:

In the image below, there were three builds initiated of which two were unsuccessful. The following image is of the Stage Logs(Build). The initial builds failed because the python3.10-venv package was not installed.

This note was not in the Stage Logs(test) as one would expect because it's the stage that failed but instead it was noted in the Stage Logs(Build) which was successful.

Once I installed the python3.10-venv package and initiated the build once more, it was successful.


Dashboard > Build Flask > main >

Build History trend

Filter builds...

✓ #3	Oct 11, 2022, 4:25 AM
⊗ #2	Oct 11, 2022, 3:52 AM
⊗ #1	Oct 11, 2022, 3:41 AM

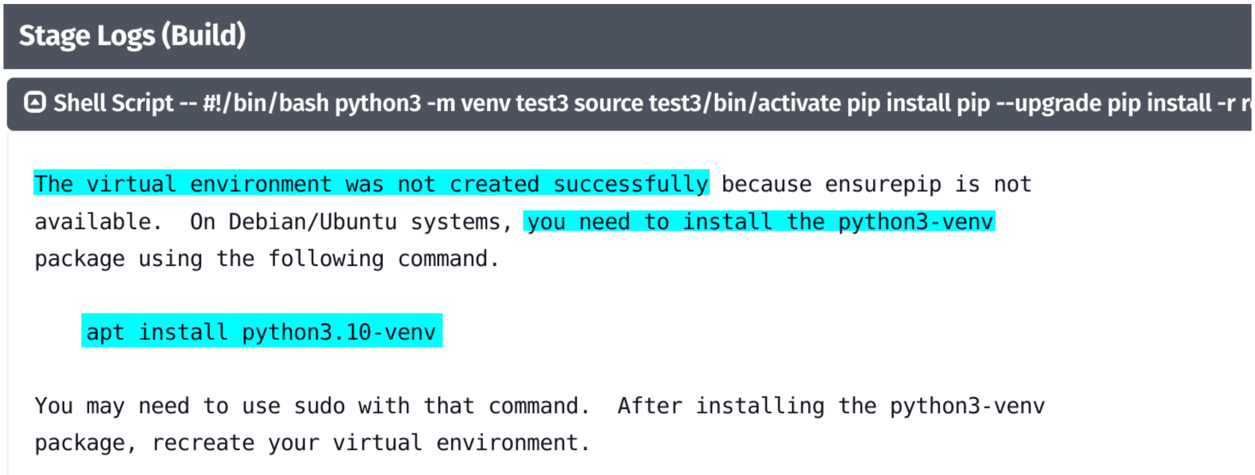
Atom feed for all Atom feed for failures

Stage View

	Declarative: Checkout SCM	Build	test
Average stage times: (Average full run time: ~16s)	2s	7s	805ms
#3 Oct 11 04:25 No Changes	1s	12s	918ms
#2 Oct 11 03:52 No Changes	1s	2s	505ms failed
#1 Oct 11 03:41 No Changes	5s	8s	992ms failed

Latest Test Result (no failures)

Permalinks

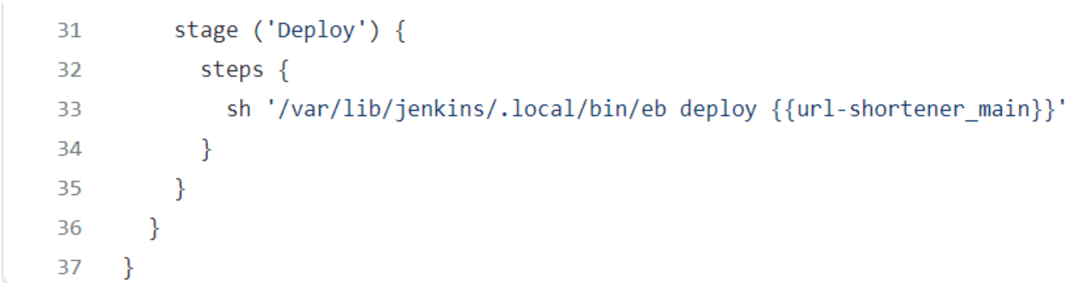

Stage Logs (Build)

Shell Script -- #!/bin/bash python3 -m venv test3 source test3/bin/activate pip install pip --upgrade pip install -r r

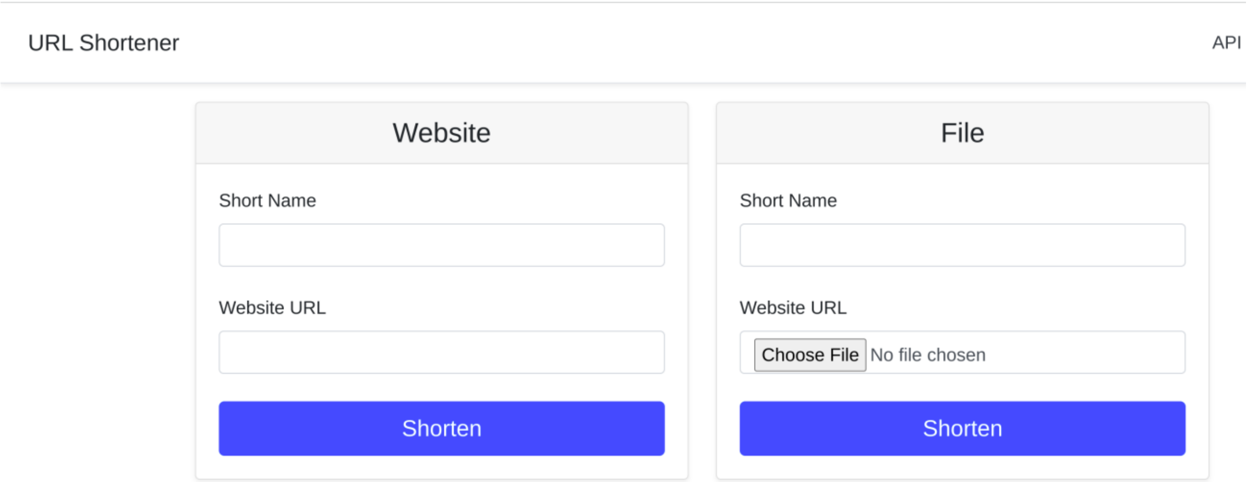
```
The virtual environment was not created successfully because ensurepip is not
available.  On Debian/Ubuntu systems, you need to install the python3-venv
package using the following command.

    apt install python3.10-venv

You may need to use sudo with that command.  After installing the python3-venv
package, recreate your virtual environment.
```

Deploy the Application from Elastic Beanstalk CLI

- □ On the Jenkins server as the Jenkins user change directories
  ‘‘‘ cd /var/lib/Jenkins/workspace/{{name of your project}}/
- □ Configure a local directory for your EB environment and source code
  ``` eb init ```
  ``` eb create ```
- □ Add a deployment stage to the pipeline in your Jenkinsfile



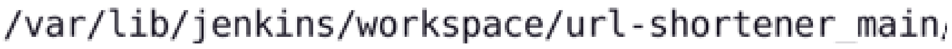
```
      stage ('Deploy') {
        steps {
          sh '/var/lib/jenkins/.local/bin/eb deploy {{url-shortener_main}}'
        }
      }
    }
  }
```



Debugging/Build Troubleshooting:

The original deployment document instructed the user the cd into “/var/workspace/{the name of your project}”. However, the correct path is /var/lib/jenkins/workspace/{the name of your project}

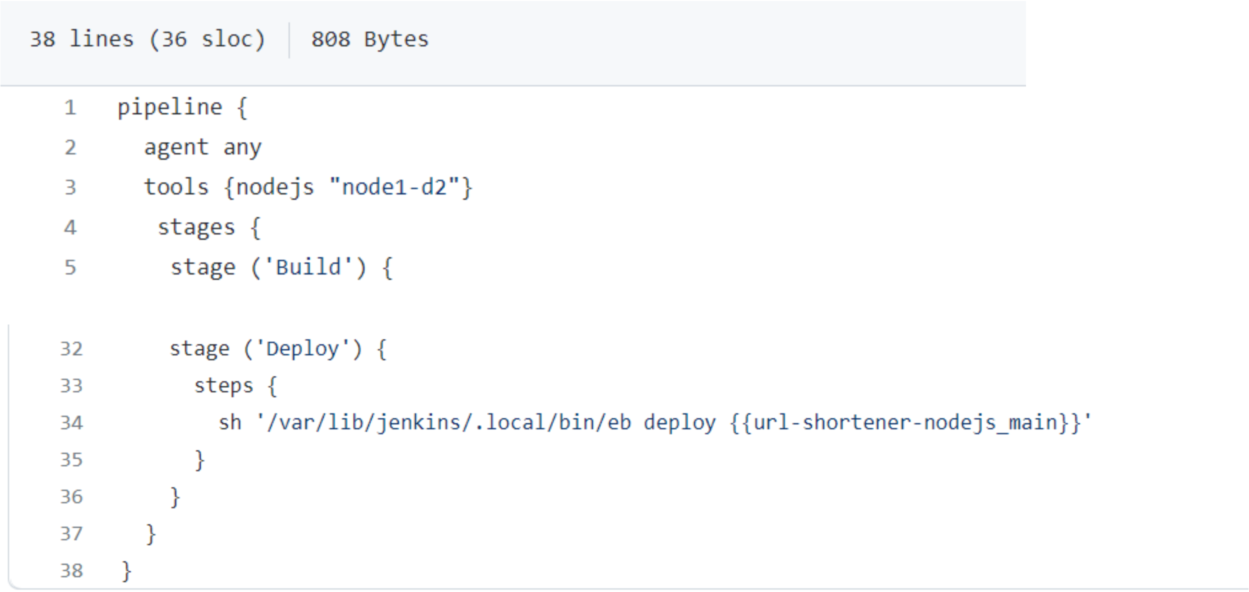
For Example:



```
/var/lib/jenkins/workspace/url-shortener_main
```

Add or Modify the Pipeline

- ☐ In Jenkins, find and install the NodeJS plugin
- ☐ Add a second test to the Jenkinsfile in GitHub
- ☐ Initiate a Build

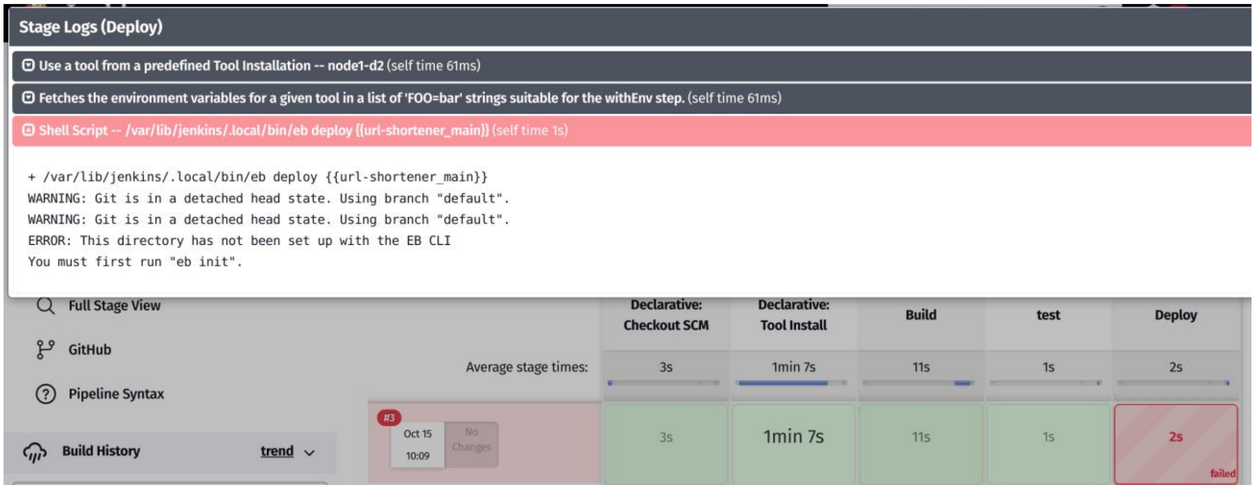


38 lines (36 sloc) | 808 Bytes

```
pipeline {
  agent any
  tools {nodejs "node1-d2"}
   stages {
    stage ('Build') {

      stage ('Deploy') {
        steps {
          sh '/var/lib/jenkins/.local/bin/eb deploy {{url-shortener-nodejs_main}}'
        }
      }
    }
}
```

Debugging/Build Troubleshooting:

According to the Logs the Build failed because the “eb init” command needed to be ran. The Build failed again because there was an error in the node name in the Jenkinsfile.



Stage Logs (Deploy)

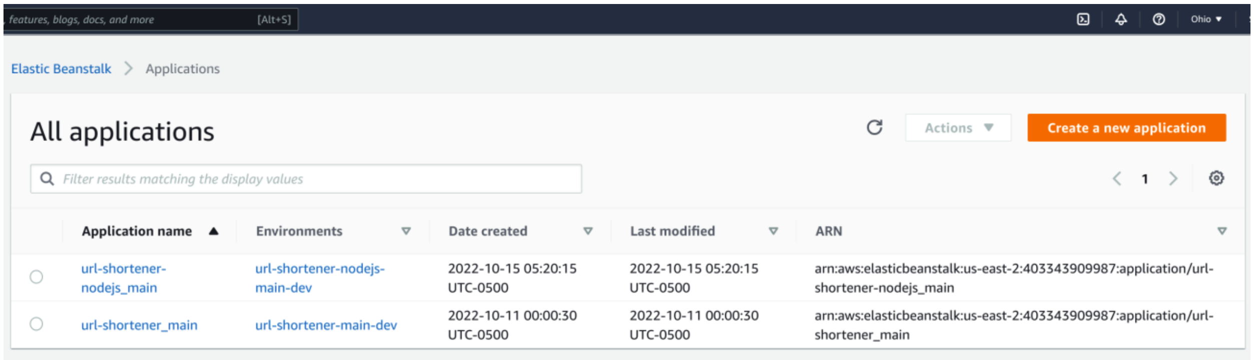
Use a tool from a predefined Tool Installation -- node1-d2 (self time 61ms)

Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 61ms)

Shell Script -- /var/lib/jenkins/.local/bin/eb deploy {{url-shortener_main}} (self time 1s)

```
+ /var/lib/jenkins/.local/bin/eb deploy {{url-shortener_main}}
WARNING: Git is in a detached head state. Using branch "default".
WARNING: Git is in a detached head state. Using branch "default".
ERROR: This directory has not been set up with the EB CLI
You must first run "eb init".
```

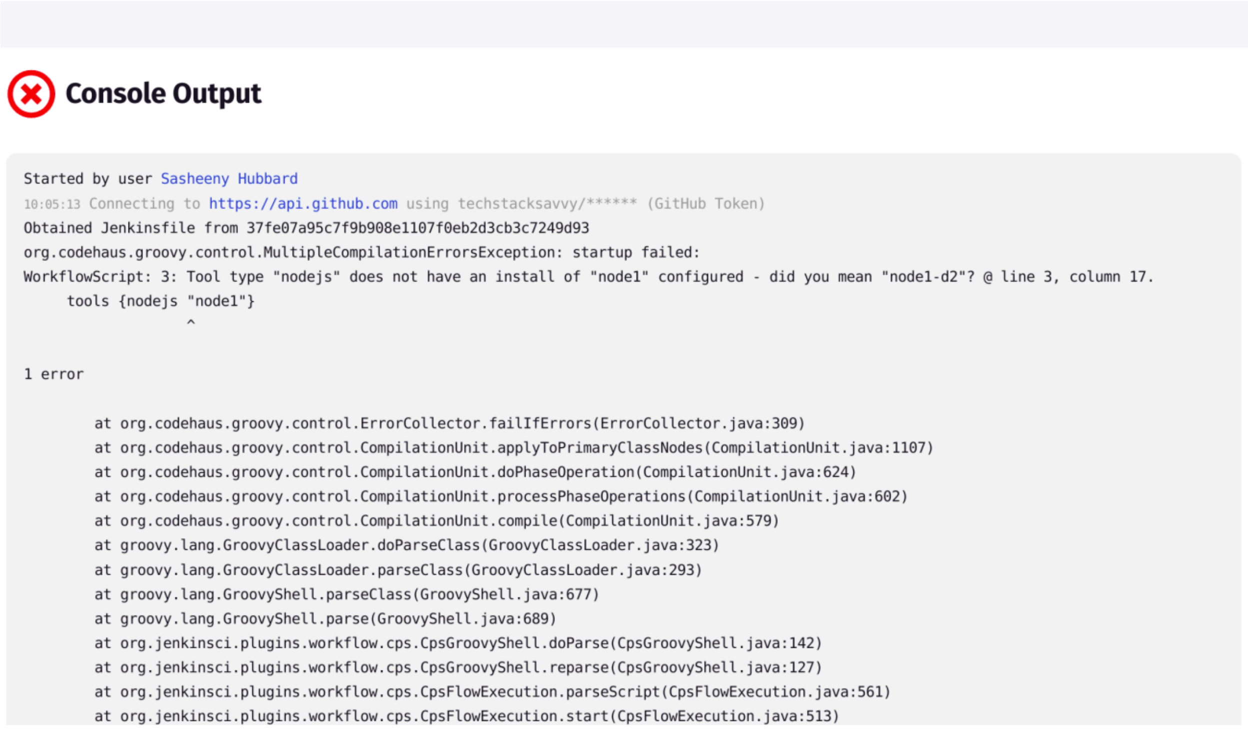
	Declarative: Checkout SCM	Declarative: Tool Install	Build	test	Deploy
Average stage times:	3s	1min 7s	11s	1s	2s
#3 Oct 15 10:09 No Changes	3s	1min 7s	11s	1s	2s failed


features, blogs, docs, and more
[Alt+S]
Ohio ▾
Actions ▼
Create a new application
Filter results matching the display values
1

Elastic Beanstalk > Applications

All applications

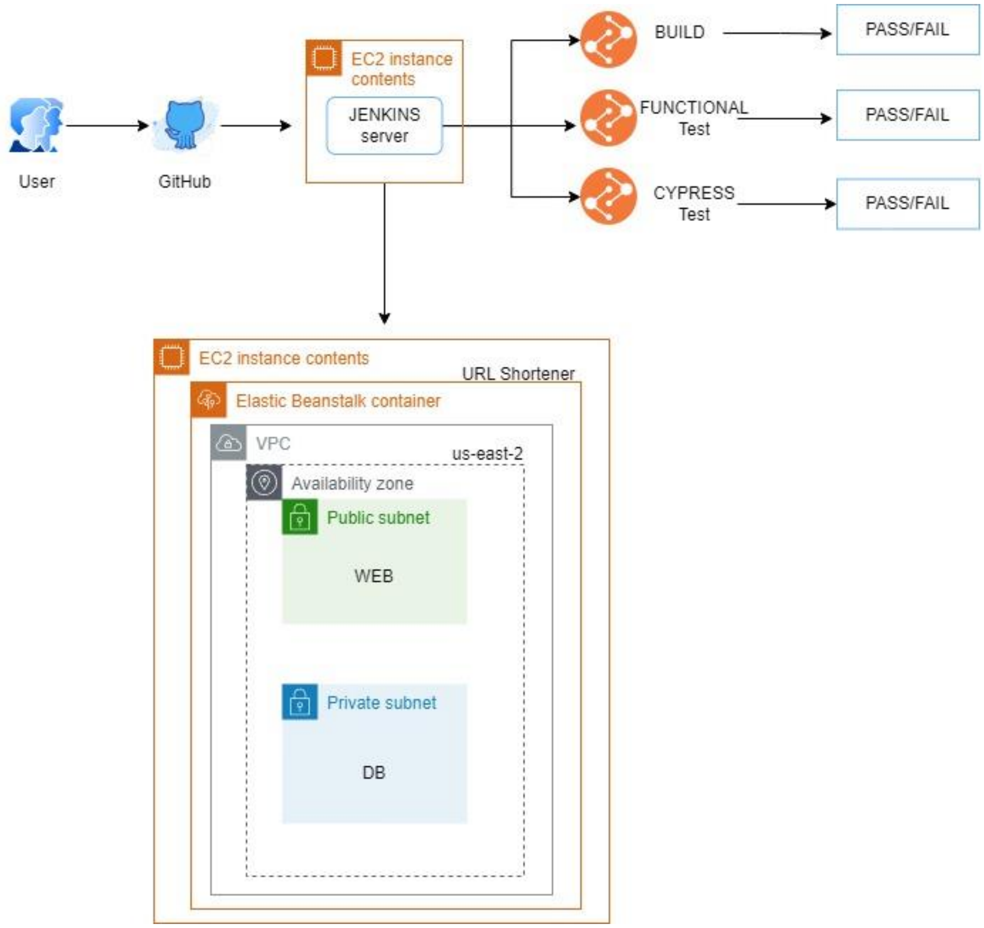
	Application name ▲	Environments	Date created	Last modified	ARN
○	url-shortener-nodejs_main	url-shortener-nodejs-main-dev	2022-10-15 05:20:15 UTC-0500	2022-10-15 05:20:15 UTC-0500	arn:aws:elasticbeanstalk:us-east-2:403343909987:application/url-shortener-nodejs_main
○	url-shortener_main	url-shortener-main-dev	2022-10-11 00:00:30 UTC-0500	2022-10-11 00:00:30 UTC-0500	arn:aws:elasticbeanstalk:us-east-2:403343909987:application/url-shortener_main



Console Output

```
Started by user Sasheeny Hubbard
10:05:13 Connecting to https://api.github.com using techstacksavvy/****** (GitHub Token)
Obtained Jenkinsfile from 37fe07a95c7f9b908e1107f0eb2d3cb3c7249d93
org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed:
WorkflowScript: 3: Tool type "nodejs" does not have an install of "node1" configured - did you mean "node1-d2"? @ line 3, column 17.
     tools {nodejs "node1"}
                   ^

1 error

        at org.codehaus.groovy.control.ErrorCollector.failIfErrors(ErrorCollector.java:309)
        at org.codehaus.groovy.control.CompilationUnit.applyToPrimaryClassNodes(CompilationUnit.java:1107)
        at org.codehaus.groovy.control.CompilationUnit.doPhaseOperation(CompilationUnit.java:624)
        at org.codehaus.groovy.control.CompilationUnit.processPhaseOperations(CompilationUnit.java:602)
        at org.codehaus.groovy.control.CompilationUnit.compile(CompilationUnit.java:579)
        at groovy.lang.GroovyClassLoader.doParseClass(GroovyClassLoader.java:323)
        at groovy.lang.GroovyClassLoader.parseClass(GroovyClassLoader.java:293)
        at groovy.lang.GroovyShell.parseClass(GroovyShell.java:677)
        at groovy.lang.GroovyShell.parse(GroovyShell.java:689)
        at org.jenkinsci.plugins.workflow.cps.CpsGroovyShell.doParse(CpsGroovyShell.java:142)
        at org.jenkinsci.plugins.workflow.cps.CpsGroovyShell.reparse(CpsGroovyShell.java:127)
        at org.jenkinsci.plugins.workflow.cps.CpsFlowExecution.parseScript(CpsFlowExecution.java:561)
        at org.jenkinsci.plugins.workflow.cps.CpsFlowExecution.start(CpsFlowExecution.java:513)
```

Deployment 2 Diagram


User
GitHub
EC2 instance contents
JENKINS server
BUILD
FUNCTIONAL Test
CYPRESS Test
PASS/FAIL
PASS/FAIL
PASS/FAIL
EC2 instance contents
URL Shortener
Elastic Beanstalk container
VPC
us-east-2
Availability zone
Public subnet
WEB
Private subnet
DB