1. **[**ec2-user ~]$ sudo yum update –y
2. Add the Jenkins repo using the following command:
3. **[**ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \

https://pkg.jenkins.io/redhat-stable/jenkins.repo

1. Import a key file from Jenkins-CI to enable installation from the package:

**[**ec2-user ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

**[**ec2-user ~]$ sudo yum upgrade

1. Install Java (Amazon Linux 2023):

**[**ec2-user ~]$ sudo dnf install java-17-amazon-corretto -y

1. Install Jenkins:

**[**ec2-user ~]$ sudo yum install jenkins -y

[root@ip-10-0-29-163 ~]# systemctl enable jenkins

Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.

[root@ip-10-0-29-163 ~]# systemctl start jenkins

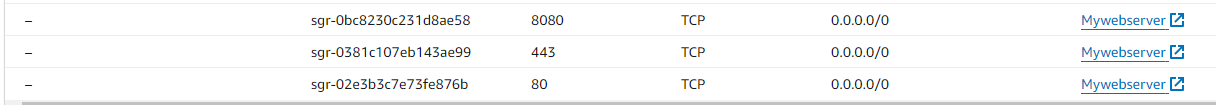
[root@ip-10-0-29-163 ~]# jenkins --version

2.426.2

[root@ip-10-0-29-163 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword

cd09c51f67c1487da2997c8016fee388

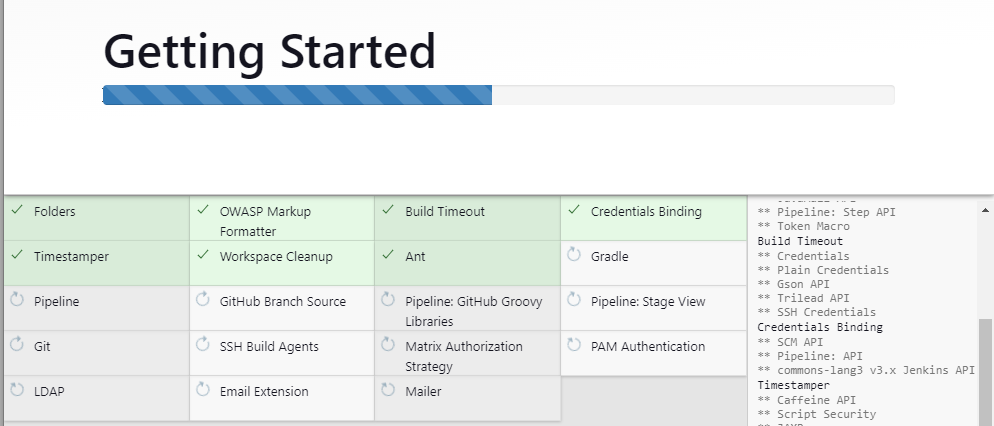
[root@ip-10-0-29-163 ~]#



<http://3.143.147.122:8080/>

Jenkins

jenkins



Integrate Maven to Jenkins and add Github credentials to Jenkins

Git credential personal access token for jenkins

ghp\_mgwUP7SZFSWIqFMy6qn8dCAe2WlJx427nBdi

ghp\_FAD1ZckvYaXaDxK4DCzmWzfWshAgQV33V8ti

create pipeline script for build and test artifacts and create CI job on Jenkins

PS C:\Users\shrijandra\EKS> cd '.\CICD Argo\'

PS C:\Users\shrijandra\EKS\CICD Argo> git add .

fatal: not a git repository (or any of the parent directories): .git

PS C:\Users\shrijandra\EKS\CICD Argo>git init

PS C:\Users\shrijandra\EKS\CICD Argo> git add .

PS C:\Users\shrijandra\EKS\CICD Argo> git commit -m "added jenkinsfile"

PS C:\Users\shrijandra\EKS\CICD Argo> git checkout -b shrijandra/web-app

Switched to a new branch 'shrijandra/web-app'

PS C:\Users\shrijandra\EKS\CICD Argo> git branch -a

master

\* shrijandra/web-app

PS C:\Users\shrijandra\EKS\CICD Argo> git push https://github.com/shrijandra/web-app.git

fatal: unable to access 'https://github.com/shrijandra/web-app.git/': OpenSSL SSL\_connect: Connection was reset in connection to github.com:443

PS C:\Users\shrijandra\EKS\CICD Argo> git config --global --unset-all remote.origin.proxy

PS C:\Users\shrijandra\EKS\CICD Argo> git push https://github.com/shrijandra/web-app.git

Enumerating objects: 25, done.

Counting objects: 100% (25/25), done.

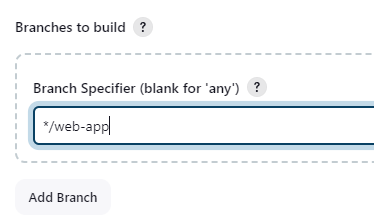
PS C:\Users\shrijandra\EKS\CICD Argo> git config --global user.name

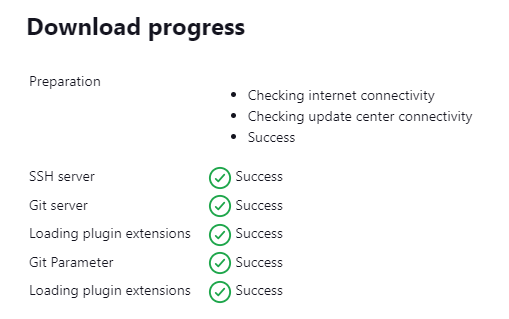
shrijandra

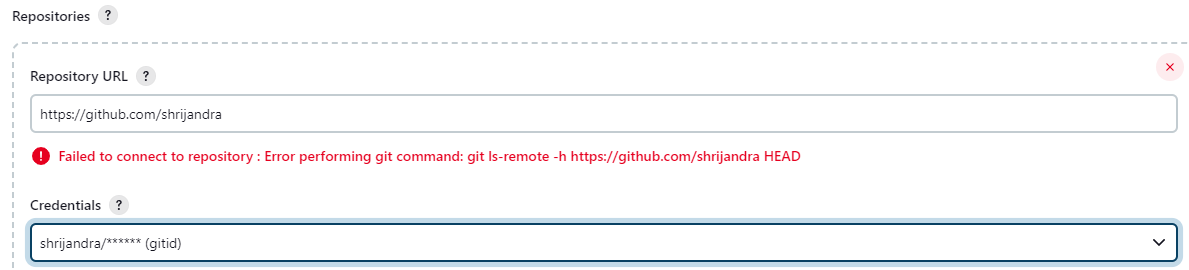
PS C:\Users\shrijandra\EKS\CICD Argo> git config --global user.email

shrijandra@yahoo.com

PS C:\Users\shrijandra\EKS\CICD Argo>







In console output

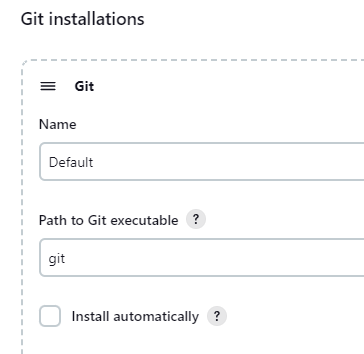
java.io.IOException: Cannot run program "git" (in directory "/var/lib/jenkins/caches/git-f5476d591911f224c882665611560dc9"): error=2, No such file or directory



***[root@ip-10-0-29-163 ~]# yum install git -y***

Last metadata expiration check: 1:26:19 ago on Tue Dec 26 03:31:47 2023.

Dependencies resolved.





ERROR: /var/lib/jenkins/workspace/web-app@script/9b9e2a24fb8b77f9ebdd58dbef60ff37c76b09addb8291db67a570022597f18c/Jenkinsfile not found

***File name change from jenkins to Jenkins resolved the above issue.***

The file name in Git was updated manually causing below error while pushing from local PC.

PS C:\Users\shrijandra\EKS\CICD Argo> git push https://github.com/shrijandra/web-app.git

To https://github.com/shrijandra/web-app.git

! [rejected] shrijandra/web-app -> shrijandra/web-app (fetch first)

error: failed to push some refs to 'https://github.com/shrijandra/web-app.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

PS C:\Users\shrijandra\EKS\CICD Argo> git add .

PS C:\Users\shrijandra\EKS\CICD Argo> git commit -m 'syntax error fixed dfd'

On branch shrijandra/web-app

nothing to commit, working tree clean

so use git push -f origin master //force push

+ mvn clean package

/var/lib/jenkins/workspace/web-app@tmp/durable-42f38b61/script.sh: line 1: mvn: command not found

***sudo yum install maven -y***

[root@ip-10-0-29-163 ~]# mvn --version

Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.4)

Maven home: /usr/share/maven

Java version: 17.0.8.1, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86\_64

Default locale: en, platform encoding: UTF-8

OS name: "linux", version: "6.1.49-70.116.amzn2023.x86\_64", arch: "amd64", family: "unix"

[root@ip-10-0-29-163 ~]#

[ERROR] The goal you specified requires a project to execute but there is no POM in this directory (/var/lib/jenkins/workspace/web-app).

After having application with POM file still the same error.

[ERROR] The goal you specified requires a project to execute but there is no POM in this directory (/var/lib/jenkins/workspace/web-app). Please verify you invoked Maven from the correct directory. -> [Help 1]

Again the same error

[root@ip-10-0-29-163 ~]# mvn -e

[INFO] Error stacktraces are turned on.

[INFO] Scanning for projects...

[INFO] ------------------------------------------------------------------------

[INFO] BUILD FAILURE

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 0.163 s

[INFO] Finished at: 2023-12-26T06:53:11Z

[ERROR] No goals have been specified for this build. You must specify a valid lifecycle phase or a goal in the format <plugin-prefix>:<goal> or <plugin-group-id>:<plugin-artifact-id>[:<plugin-version>]:<goal>. Available lifecycle phases are: validate, initialize, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes, generate-test-sources, process-test-sources, generate-test-resources, process-test-resources, test-compile, process-test-classes, test, prepare-package, package, pre-integration-test, integration-test, post-integration-test, verify, install, deploy, pre-clean, clean, post-clean, pre-site, site, post-site, site-deploy. -> [Help 1]

org.apache.maven.lifecycle.NoGoalSpecifiedException: No goals have been specified for this build. You must specify a valid lifecycle phase or a goal in the format <plugin-prefix>:<goal> or <plugin-group-id>:<plugin-artifact-id>[:<plugin-version>]:<goal>. Available lifecycle phases are: validate, initialize, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes, generate-test-sources, process-test-sources, generate-test-resources, process-test-resources, test-compile, process-test-classes, test, prepare-package, package, pre-integration-test, integration-test, post-integration-test, verify, install, deploy, pre-clean, clean, post-clean, pre-site, site, post-site, site-deploy.

<https://plugins.jenkins.io/pipeline-maven/>

12/26/2023

 environment {

        PATH = "/usr/bin:$PATH"

    }

[ERROR] The goal you specified requires a project to execute but there is no POM in this directory (/var/lib/jenkins/workspace/web-app).

The below command clean up the workspace which is the reason no file inside (/var/lib/jenkins/workspace/web-app

[root@ip-10-0-29-163 ~]# cd /var/lib/jenkins/workspace/web-app

[root@ip-10-0-29-163 web-app]# ls

[root@ip-10-0-29-163 web-app]# ls -l

total 0

[root@ip-10-0-29-163 web-app]#

 stages {

        stage("clone code"){

            steps{

                git branch: 'shrijandra/web-app', credentialsId: 'gitid', url: 'https://github.com/shrijandra/web-app.git'

            }

        }

        stage("Cleanup Workspace"){

            steps {

                cleanWs()

              }

        }

Console output

[Pipeline] cleanWs

[WS-CLEANUP] Deleting project workspace...

[WS-CLEANUP] Deferred wipeout is used...

[WS-CLEANUP] done

[Pipeline] }

Defining tools as maven or use environment.

pipeline {

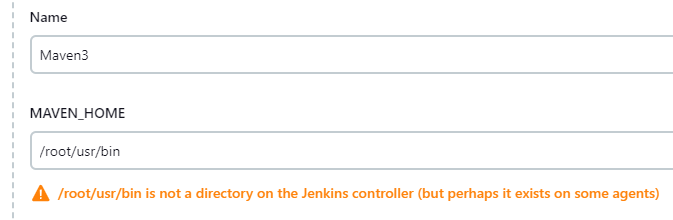
agent any

tools {

maven "Maven3"

}

And check the Manage Jenkins – tools – Maven as below

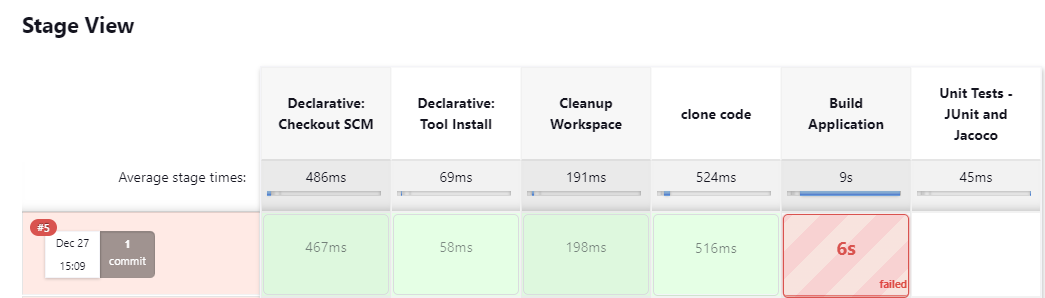


[root@ip-10-0-29-163 workspace]# cd devops

[root@ip-10-0-29-163 devops]# ls

Jenkinsfile cicd.docx pom.xml src target '~$cicd.docx'

[root@ip-10-0-29-163 devops]#



[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.1:compile (default-compile) on project simple-web-app: Compilation failure: Compilation failure:

[ERROR] /var/lib/jenkins/workspace/devops/src/main/java/org/mitre/web/HomeController.java:[24,24] package javax.annotation does not exist

[ERROR] /var/lib/jenkins/workspace/devops/src/main/java/org/mitre/web/HomeController.java:[49,10] cannot find symbol

[ERROR] symbol: class Resource

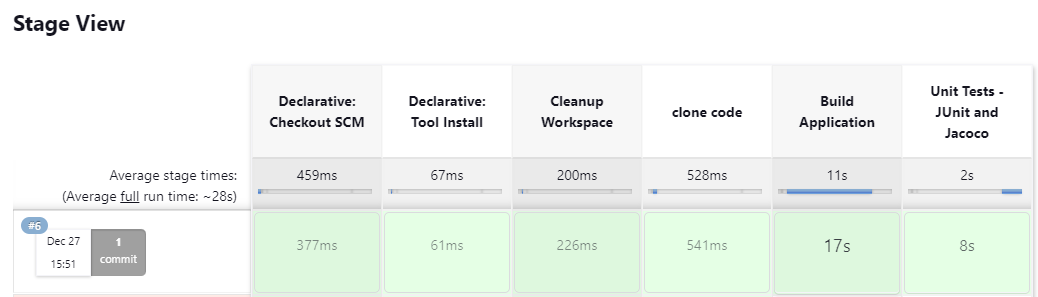
[ERROR] location: class org.mitre.web.HomeController

Changing another sample POM file Source code: https://github.com/ajautomation/PassParametersRunTimeViaMVN

[root@ip-10-0-29-163 devops]# ls

Jenkinsfile cicd.docx pom.xml src target '~$cicd.docx'

Build completed and Unit test completed.



pipeline {

    agent any

    tools {

        maven "Maven3"

    }

  //  environment {

  //      PATH = "/usr/bin:$PATH"

  //  }

    stages {

        stage("Cleanup Workspace"){

            steps {

                cleanWs()

              }

        }

        stage("clone code"){

            steps{

                git branch: 'shrijandra/web-app', credentialsId: 'gitid', url: 'https://github.com/shrijandra/web-app.git'

            }

        }

        stage("Build Application"){

            steps {

                sh "mvn clean package"

            }

        }

        stage('Unit Tests - JUnit and Jacoco'){

           steps {

                sh "mvn test"

          }

        }

    }

}

These are the default life cycle phases in maven

* **validate** - validate the project is correct and all necessary information is available
* **compile** - compile the source code of the project
* **test** - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
* **package** - take the compiled code and package it in its distributable format, such as a JAR.
* **verify** - run any checks on results of integration tests to ensure quality criteria are met
* **install** - install the package into the local repository, for use as a dependency in other projects locally
* **deploy** - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.