[admin@ec2-15-207-115-34.ap-south-1.compute.amazonaws.com](mailto:admin@ec2-15-207-115-34.ap-south-1.compute.amazonaws.com" \o "mailto:admin@ec2-15-207-115-34.ap-south-1.compute.amazonaws.com" \t "_blank)

15.207.115.34

**githubCodeRepoCredentials**

**# docker installation in aws linux/Debian vm**

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

you’ll encounter with error sudo: gpg: command not found (23) Failed writing body, to fix it 1st do below step :

sudo apt install gnupg

after that curl step and then

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io

docker –version

then run docker run or docker help , you’ll hit with permission denied error i.e

docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/create": dial unix /var/run/docker.sock: connect: permission denied. See 'docker run --help'.

To fix above error , follow below steps

sudo usermod -aG docker <username> to find username run whoami

cd /var/run and check permission and follow below steps :



Now , you can smoothly run docker .

**# MySQL installation**

docker pull mysql

docker run --name mysql-container -e MYSQL\_ROOT\_PASSWORD=root -p 3306:3306 -d mysql:8.0

mysql -h 15.207.115.34 -P 3306 -u root -p

you’ll encounter -bash: mysql: command not found, to fix it

docker exec -it mysql-container mysql -u root -p

or

sudo apt install mysql-client

E: Package 'mysql-client' has no installation candidate

sudo apt update

sudo apt install default-mysql-client

mysql -h 15.207.115.34 -P 3306 -u root -p

**# Jenkins installation as docker container**

docker run –name jenkins-container -p 8000:8080 -p 50000:50000 -d -v /var/run/docker.sock:/var/run/docker.sock -v jenkins\_home:/var/jenkins\_home jenkins/jenkins:lts

<http://15.207.115.34:8000/>

jenkins

Jenkins@123

**# sonarqube installation**

**docker run** -d --name sonarqube -p 9090:9000 -p 9092:9092 **sonarqube**

admin

admin123

sqa\_089d53b3f8c642bd9112403d023e114a48864713 -> sonar global token

sonar\_token1 -> sqa\_dcf0018415d271a761c9e342e42edf73c9a9621b

<https://docs.sonarqube.org/latest/devops-platform-integration/github-integration/>

**# Nexus installation and integration with Jenkins**

<https://help.sonatype.com/repomanager3>

First install ,

* JDK version 1.8 or newer
* Nexus Repository Manager OSS version 2
* [Jenkins CI version 2 or newer](https://www.theserverside.com/tutorial/Need-a-CI-tool-Heres-a-Jenkins-tutorial-for-beginners)

<https://www.devopsschool.com/blog/how-to-install-and-configure-sonatype-nexus/>

<https://www.devopsschool.com/blog/nexus-integration-with-jenkins-to-upload-download-package/>

Now in Jenkins ,

**Install the Nexus Jenkins integration plugin**

[Nexus Artifact Uploader](https://plugins.jenkins.io/nexus-artifact-uploader)

[Nexus PlatformVersion](https://plugins.jenkins.io/nexus-jenkins-plugin)

### Configure the Jenkins Nexus plugin

In the Jenkins dashboard, navigate to "Manage Jenkins" > "Configure System".

Scroll down to the "Nexus Repository Manager" section and provide the Nexus URL, username, and password.

### Create a Jenkins build job

### sudo apt-get remove --purge Jenkins

### sudo dpkg -r Jenkins

### sudo rm -rf /var/lib/Jenkins

### sudo rm /etc/apt/sources.list.d/jenkins.list

### sudo apt-get update

### <https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>

### sudo cat /var/lib/jenkins/secrets/initialAdminPassword

### sudo dpkg -i jenkins\_ 2.303.2\_all.deb

### wget<http://pkg.jenkins.io/debianstable/binary/jenkins_2.387.3_all.db>

### sudo rm /etc/apt/sources.list.d/jenkins.list

ghp\_CZx5XNxXTyzg2kOSqLoYpiA7v7JnQw417tEm.

**Azure devops for SAP commerce Hybris**

Connected with git repo and it’s branch where yaml file for pipeline is present

Encounter error as No hosted parallelism has been purchased or granted. To request a free parallelism grant, please fill out the following form <https://aka.ms/azpipelines-parallelism-request>

It is resoled either by requesting azure to grant by filling this form <https://forms.office.com/pages/responsepage.aspx?id=v4j5cvGGr0GRqy180BHbR63mUWPlq7NEsFZhkyH8jChUMlM3QzdDMFZOMkVBWU5BWFM3SDI2QlRBSC4u>

Or create your own agent , and we choose 2nd one by creating gcp Jenkins-vm as agent by applying steps given below:

<https://www.youtube.com/watch?v=aonA7Kb7WGE>

Go to dcxcommerce/sapcommerce/Settings/Agent pools and created agent pool named myownagent

Then click agent tab and add your 1st agent to it , you’ll get link for creating agent based on different os .

I choose linux one as gcp vm is linux based.

<https://vstsagentpackage.azureedge.net/agent/3.220.5/vsts-agent-linux-x64-3.220.5.tar.gz>

~/$ mkdir myagent && cd myagent

~/myagent$ tar zxvf ~/Downloads/vsts-agent-linux-x64-3.220.5.tar.gz

~/myagent$ ./config.sh

~/myagent$ ./run.sh

and run further steps of pipeline

Then we encounter permission related error and it’s resolced by following steps

ls -l /tmp/rollup-plugin-progress

-rw-r--r-- 1 jenkins jenkins 3 Jun 19 18:25 /tmp/rollup-plugin-progress

sudo chown shalu\_a\_gupta /tmp/rollup-plugin-progress

ls -l /tmp/rollup-plugin-progress

-rw-r--r-- 1 shalu\_a\_gupta jenkins 3 Jun 19 18:25 /tmp/rollup-plugin-progress

After successful build we replace exact path with below var :

Agent.BuildDirectory ⬄ /home/shalu\_a\_gupta/myagent/\_work/1

Agent.WorkFolder ⬄ /home/shalu\_a\_gupta/myagent/\_work

The Agent.BuildDirectory variable refers to the root directory of the current build or pipeline run on the agent. It provides the path where your source code and other artifacts are downloaded and where subsequent build tasks are executed

The Agent.WorkFolder variable refers to the root directory of the current agent workspace. It provides the path where the pipeline job is executed and where the source code, build artifacts, and other files related to the pipeline are stored.

Below is the yaml code :

trigger:

branches:

include:

- main-cc-ext-azure

pool: myownagent

stages:

- stage: ExtractZipFiles

displayName: 'Extract Zip Files'

jobs:

- job: ExtractZipFilesJob

displayName: 'Extract Zip Files'

steps:

- task: ExtractFiles@1

displayName: 'Extract Files'

inputs:

archiveFilePatterns: '$(Agent.WorkFolder)/CXCOMM220500P.ZIP'

destinationFolder: '$(Agent.WorkFolder)/hybris\_suite'

- script: |

cp -R $(Agent.BuildDirectory)/s/core-customize/bin/custom $(Agent.WorkFolder)/hybris\_suite/hybris/bin

cp -R $(Agent.BuildDirectory)/s/core-customize/config $(Agent.WorkFolder)/hybris\_suite/hybris

- task: ExtractFiles@1

displayName: 'ExtractIntegration'

inputs:

archiveFilePatterns: '$(Agent.WorkFolder)/CXCOMINTPK220500P\_1-80006941.ZIP'

destinationFolder: '$(Agent.WorkFolder)/hybris\_integration\_suite'

- script:

cp -R $(Agent.WorkFolder)/hybris\_integration\_suite/\* $(Agent.WorkFolder)/hybris\_suite

- script: |

cd $(Agent.WorkFolder)/hybris\_suite/hybris/bin/platform

. ./setantenv.sh

ant clean all