Build Docker Image Using Jenkins Pipeline & Push to AWS ECR

> In this tutorial we will learn how to install Jenkins, Docker, install required Jenkins plugins for docker, run automatic Docker Build with Dockerfile using Jenkins Pipeline, pushing the docker Image to AWS ECR

Configuring the System & Environment

Step 1: First Let's install Jenkins, in this tutorial we are using AWS Linux AMI 2, and for that is are the installations steps documentation:

`Download and install Jenkins`

To download and install Jenkins:

- To ensure that your software packages are up to date on your instance, use the following command to perform a quick software update:

  ```
  [ec2-user ~]$ sudo yum update –y
  ```

- Add the Jenkins repo using the following command:

  ```
  [ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo
  https://pkg.jenkins.io/redhat-stable/jenkins.repo
  ```

- 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.key
  [ec2-user ~]$ sudo yum upgrade
  ```

- Install Java:

  ```
  [ec2-user ~]$ sudo amazon-linux-extras install java-openjdk11 -y
  ```

- Install Jenkins:

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

- Enable the Jenkins service to start at boot:

  ```
  [ec2-user ~]$ sudo systemctl enable jenkins
  ```

- Start Jenkins as a service:

```
[ec2-user ~]$ sudo systemctl start jenkins
```

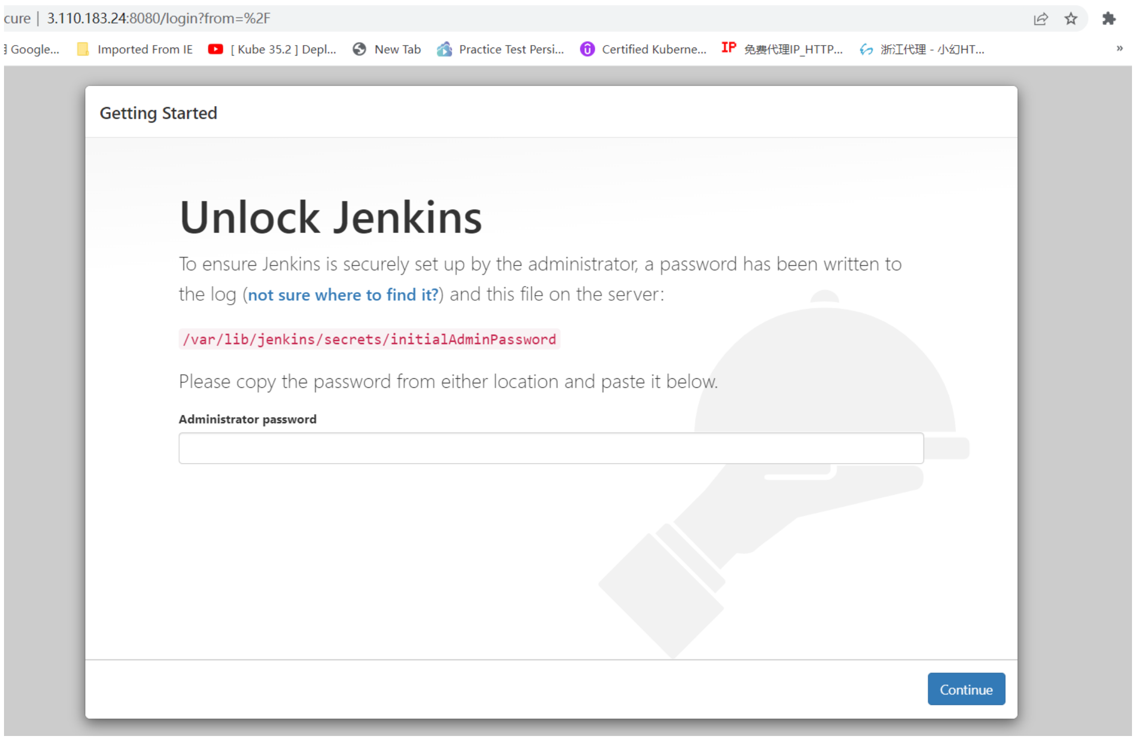
- You can check the status of the Jenkins service using the command:

```
[ec2-user ~]$ sudo systemctl status jenkins
```

Configure Jenkins:

Jenkins is now installed and running on your EC2 instance. To configure Jenkins:

- Connect to http://<your_server_public_DNS>:8080 from your favorite browser. You will be able to access Jenkins through its management interface:


cure | 3.110.183.24:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

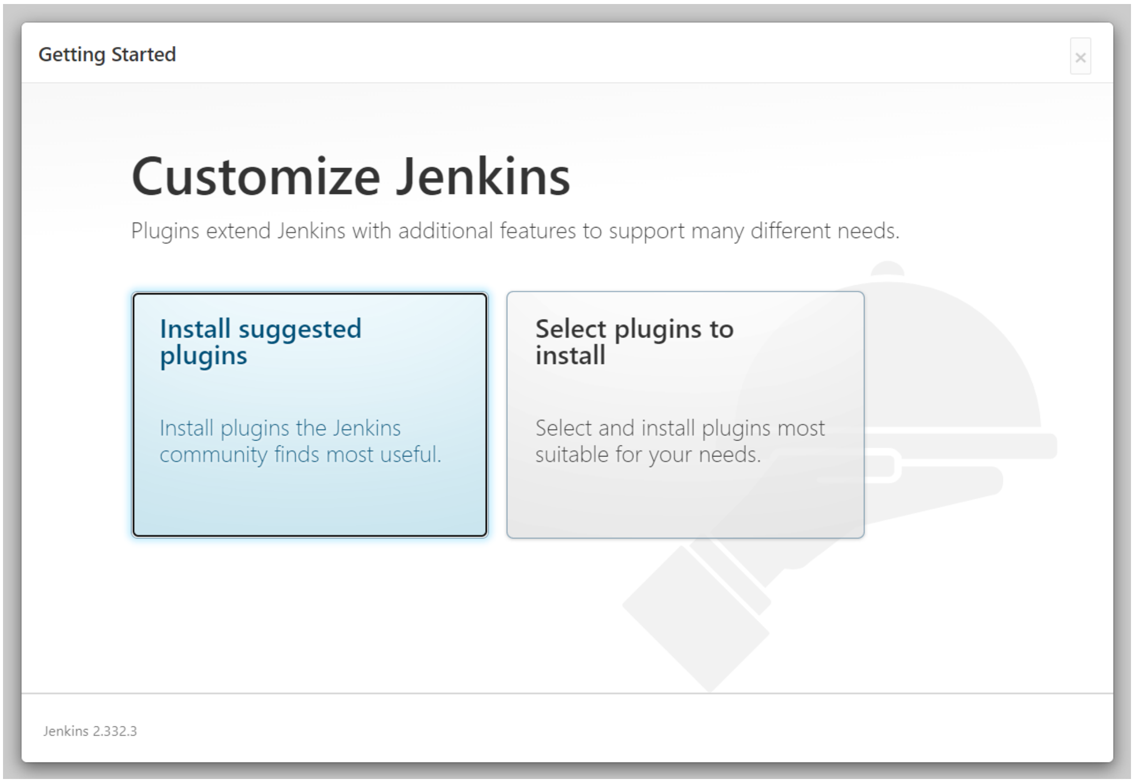
Please copy the password from either location and paste it below.

Administrator password

- As prompted, enter the password found in **/var/lib/jenkins/secrets/initialAdminPassword**. Use the following command to display this password:

```
[ec2-user ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

- The Jenkins installation script directs you to the **Customize Jenkins page**. Click **Install suggested plugins**.

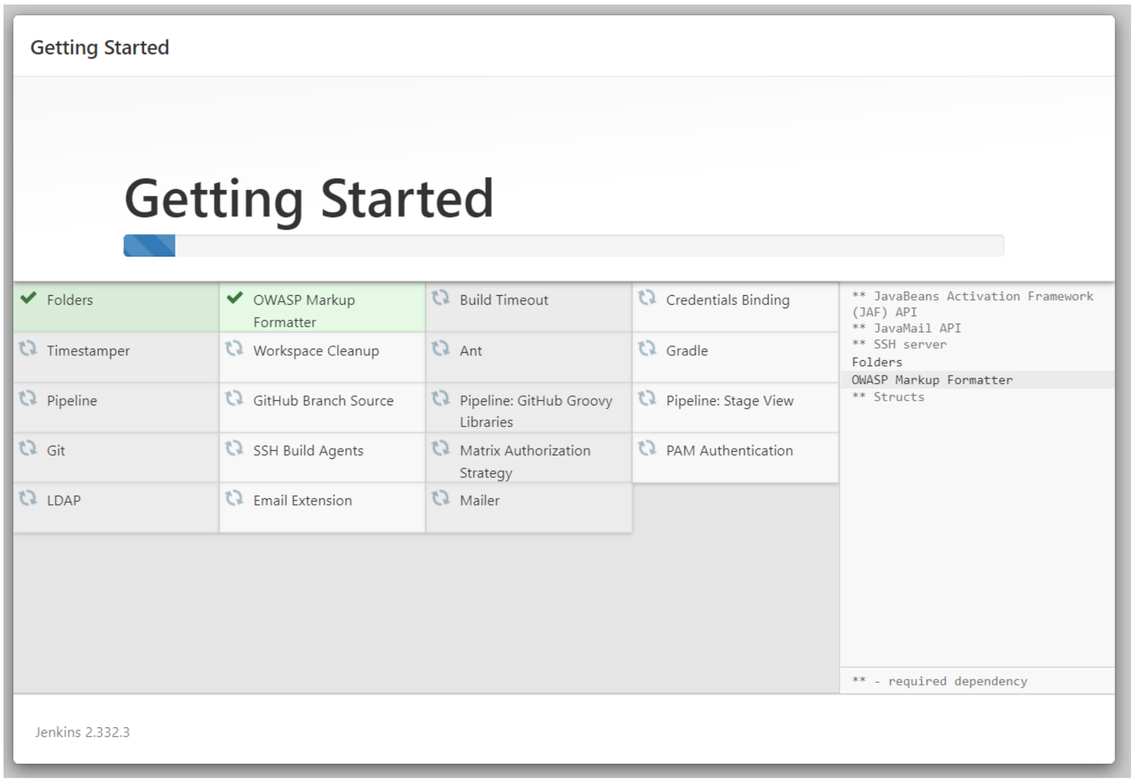

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.

Jenkins 2.332.3


Getting Started

Getting Started

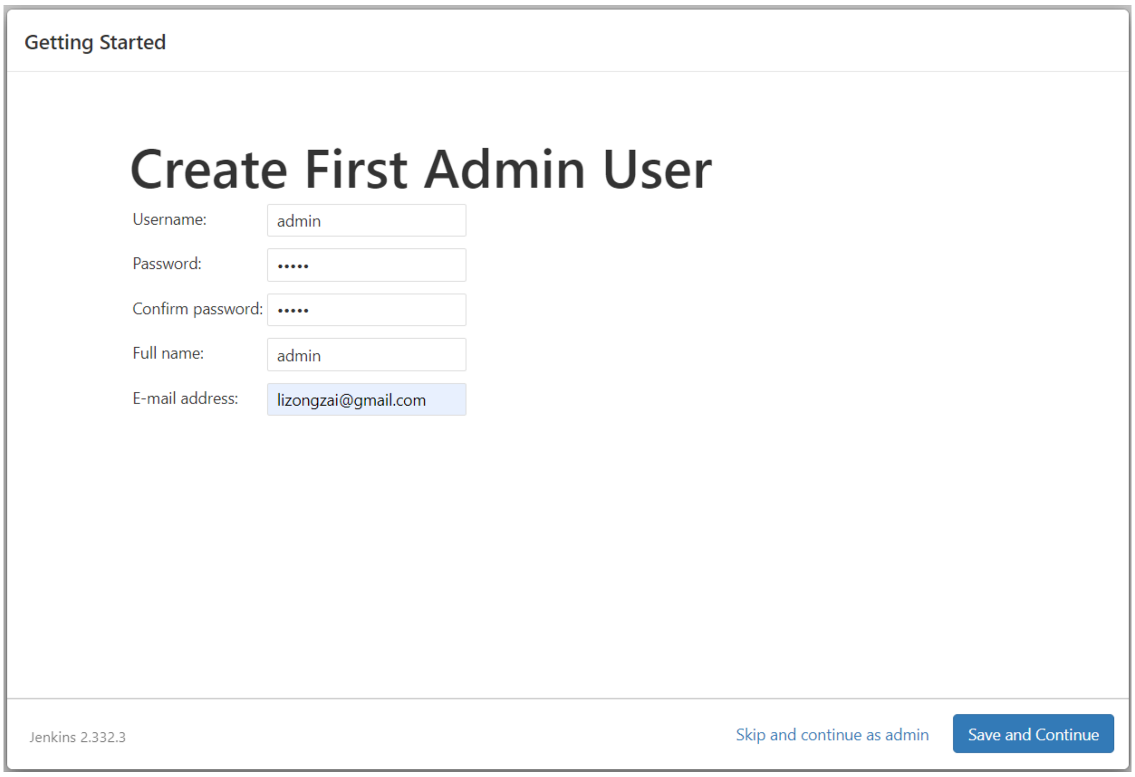
✓ Folders	✓ OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
** SSH server
Folders
OWASP Markup Formatter
** Structs
```

```
** - required dependency
```

Jenkins 2.332.3

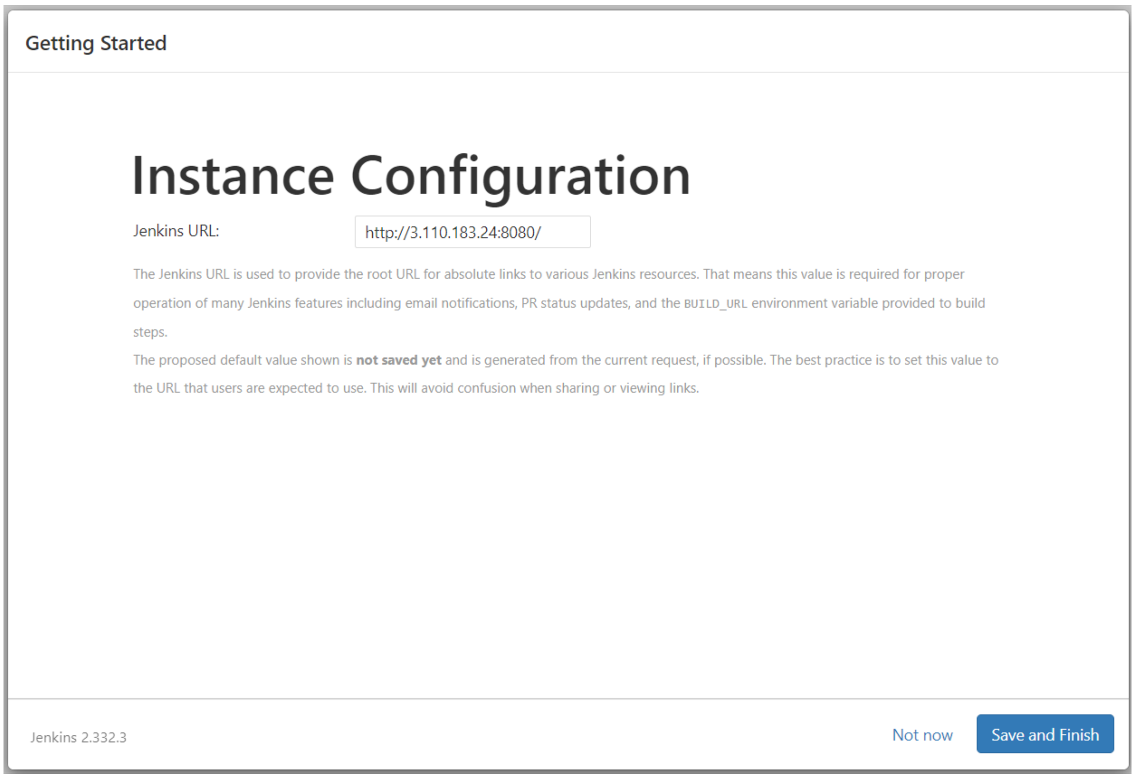
- Once the installation is complete, **Create First Admin User**, click **Save and Continue**.


Getting Started

Create First Admin User

Username:	admin
Password:	•••••
Confirm password:	•••••
Full name:	admin
E-mail address:	lizongzai@gmail.com

Jenkins 2.332.3


Getting Started

Instance Configuration

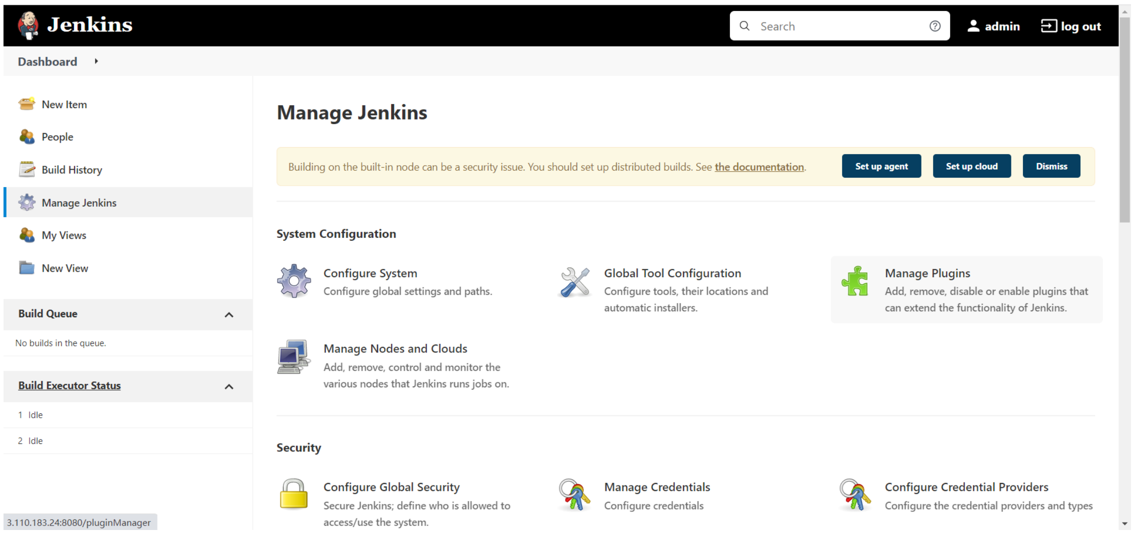
Jenkins URL: http://3.110.183.24:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.332.3

- On the left-hand side, click **Manage Jenkins**, and then click **Manage Plugins**.



Manage Jenkins

No builds in the queue.

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

System Configuration

Configure System — Configure global settings and paths.

Global Tool Configuration — Configure tools, their locations and automatic installers.

Manage Plugins — Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds — Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

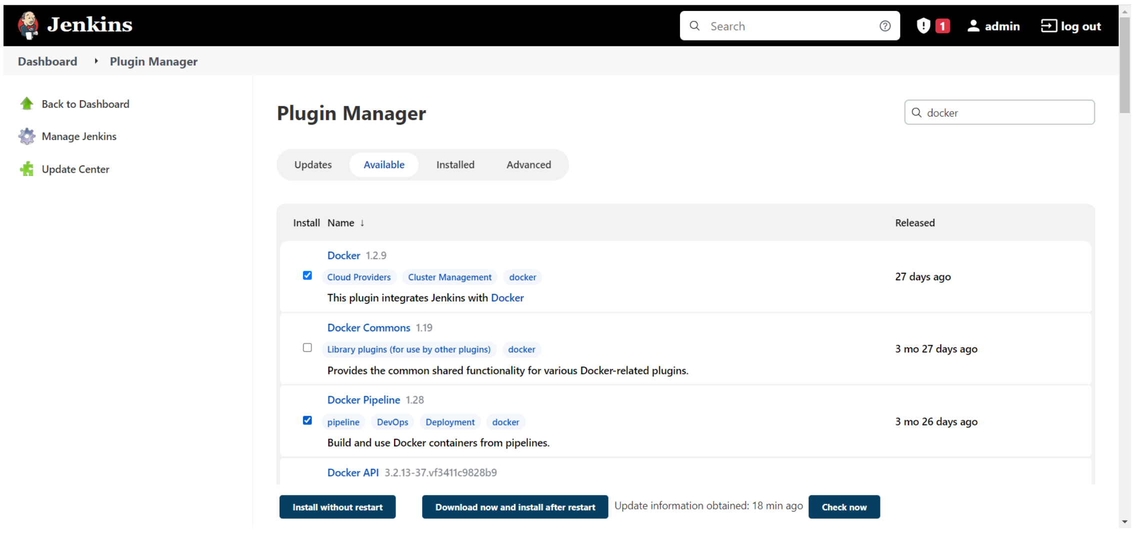
Security

Configure Global Security — Secure Jenkins; define who is allowed to access/use the system.

Manage Credentials — Configure credentials

Configure Credential Providers — Configure the credential providers and types

- Click on the **Available** tab, and then enter **Amazon EC2 plugin** at the top right.
- Select the check box next to **Amazon EC2 plugin**, and then click **Install without restart**.



Plugin Manager

Install	Name ↓	Released
☑	Docker 1.2.9 — Cloud Providers, Cluster Management, docker — This plugin integrates Jenkins with Docker	27 days ago
☐	Docker Commons 1.19 — Library plugins (for use by other plugins), docker — Provides the common shared functionality for various Docker-related plugins.	3 mo 27 days ago
☑	Docker Pipeline 1.28 — pipeline, DevOps, Deployment, docker — Build and use Docker containers from pipelines.	3 mo 26 days ago
	Docker API 3.2.13-37.vf3411c9828b9	

Update information obtained: 18 min ago

- Click **Add a new cloud**, and select **Amazon EC2**. A collection of new fields appears.
- Fill out all the fields. (Note: You will have to Add Credentials of the kind AWS Credentials.)

Configure Jenkins:

After completing this tutorial, be sure to delete the AWS resources that you created so that you do not continue to accrue charges.

Delete your EC2 instance

1. In the left-hand navigation bar of the Amazon EC2 console, choose **Instances**.
2. Right-click on the instance you created earlier and select **Terminate**.

To install Docker on an Amazon EC2 instance

1. Launch an instance with the Amazon Linux 2 AMI. For more information, see Launching an instance in the *Amazon EC2 User Guide for Linux Instances* .
2. Connect to your instance. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances* .

3. Update the installed packages and package cache on your instance.

   ```
   sudo yum update -y
   ```

4. Install the most recent Docker Engine package.

   ```
   sudo amazon-linux-extras install docker
   ```

5. Start the Docker service.

   ```
   sudo service docker start
   sudo systemctl enable docker
   ```

6. Add the `ec2-user` to the `docker` group so you can execute Docker commands without using `sudo`.

   ```
   sudo usermod -a -G docker ec2-user
   ```

7. Log out and log back in again to pick up the new `docker` group permissions. You can accomplish this by closing your current SSH terminal window and reconnecting to your instance in a new one. Your new SSH session will have the appropriate `docker` group permissions.

8. Verify that you can run Docker commands without `sudo`.

   ```
   sudo docker info
   ```

Step 2: Add Jenkins user to Docker group

```
sudo usermod -a -G docker jenkins
```

Step 3: Restart Jenkins service

```
sudo service jenkins restart
```

Step 4: Reload system daemon

```
sudo systemctl daemon-reload
```

Step 5: Restart Docker service

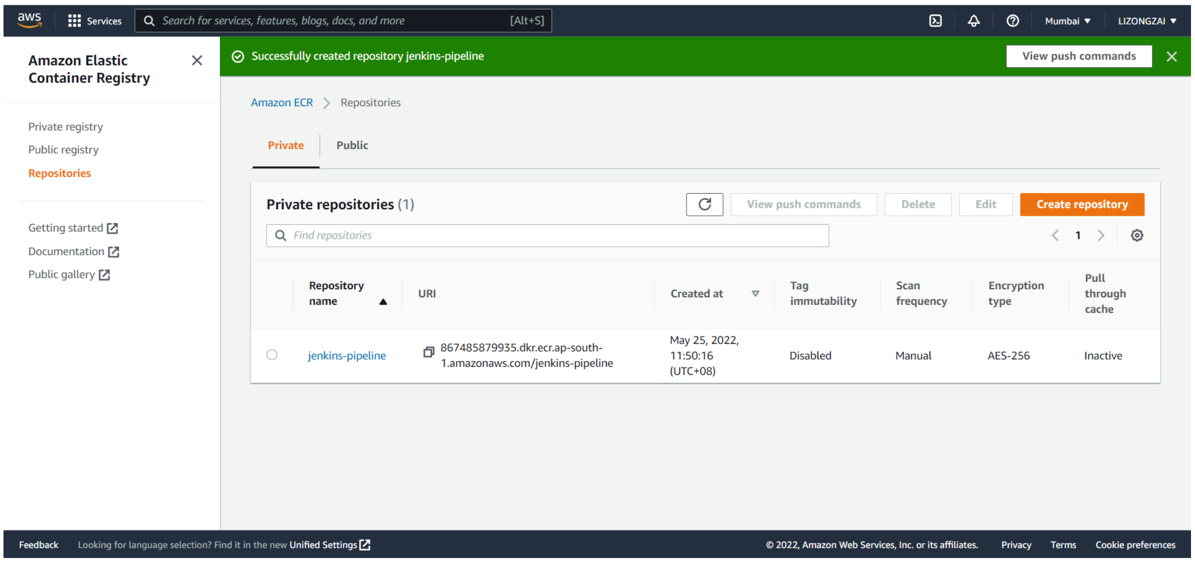
```
sudo service docker restart
```

Step 6: Install Jenkins plugins for Docker

1. Docker plug-in
2. Docker pipeline plug-in

```
sudo yum install -y git
```

Step 7: Create AWS ECR Repo in AWS



Amazon Elastic Container Registry

Successfully created repository jenkins-pipeline

View push commands

Private registry
Public registry
Repositories

Getting started
Documentation
Public gallery

Amazon ECR > Repositories

Private | Public

Private repositories (1)

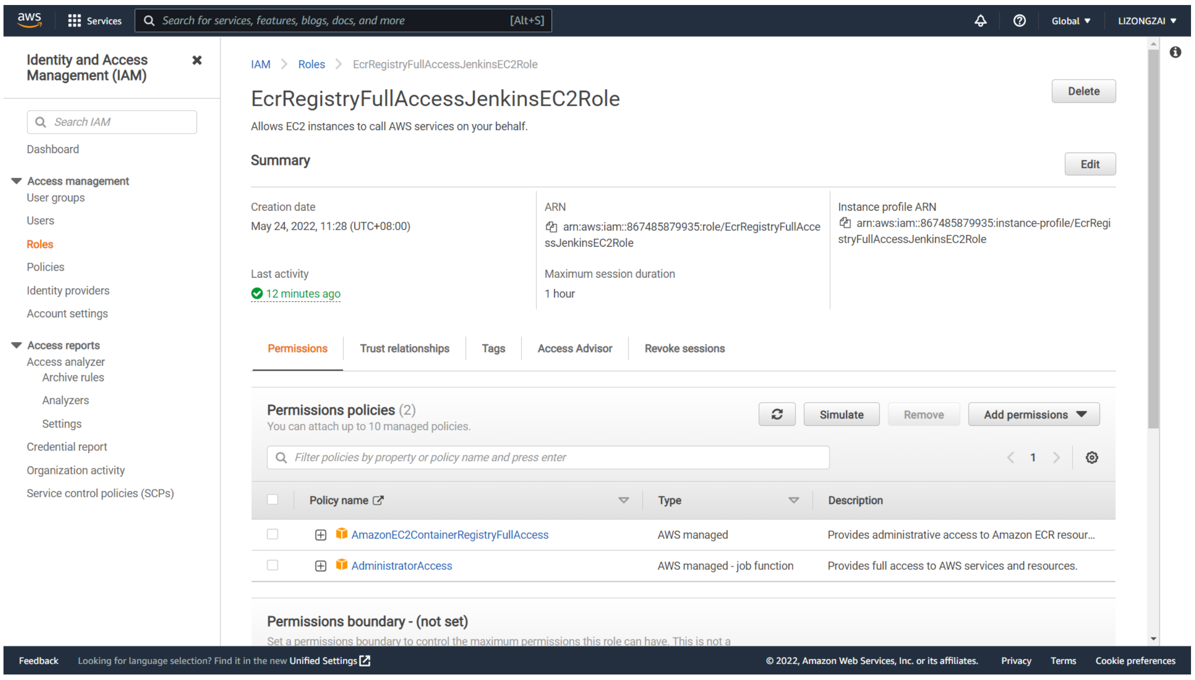
View push commands | Delete | Edit | Create repository

Find repositories

	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
○	jenkins-pipeline	867485879935.dkr.ecr.ap-south-1.amazonaws.com/jenkins-pipeline	May 25, 2022, 11:50:16 (UTC+08)	Disabled	Manual	AES-256	Inactive

Feedback   Looking for language selection? Find it in the new Unified Settings   © 2022, Amazon Web Services, Inc. or its affiliates.   Privacy   Terms   Cookie preferences

Step 8: Create IAM role and With `AmazonEC2ContainerRegistryFullAccess` policy and attach with the Jenkins EC2 Instance



Identity and Access Management (IAM)

Search IAM

Dashboard

Access management
User groups
Users
Roles
Policies
Identity providers
Account settings

Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)

IAM > Roles > EcrRegistryFullAccessJenkinsEC2Role

EcrRegistryFullAccessJenkinsEC2Role

Delete

Allows EC2 instances to call AWS services on your behalf.

Summary

Edit

Creation date
May 24, 2022, 11:28 (UTC+08:00)

ARN
arn:aws:iam::867485879935:role/EcrRegistryFullAccessJenkinsEC2Role

Instance profile ARN
arn:aws:iam::867485879935:instance-profile/EcrRegistryFullAccessJenkinsEC2Role

Last activity
12 minutes ago

Maximum session duration
1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (2)
You can attach up to 10 managed policies.

Simulate | Remove | Add permissions ▼

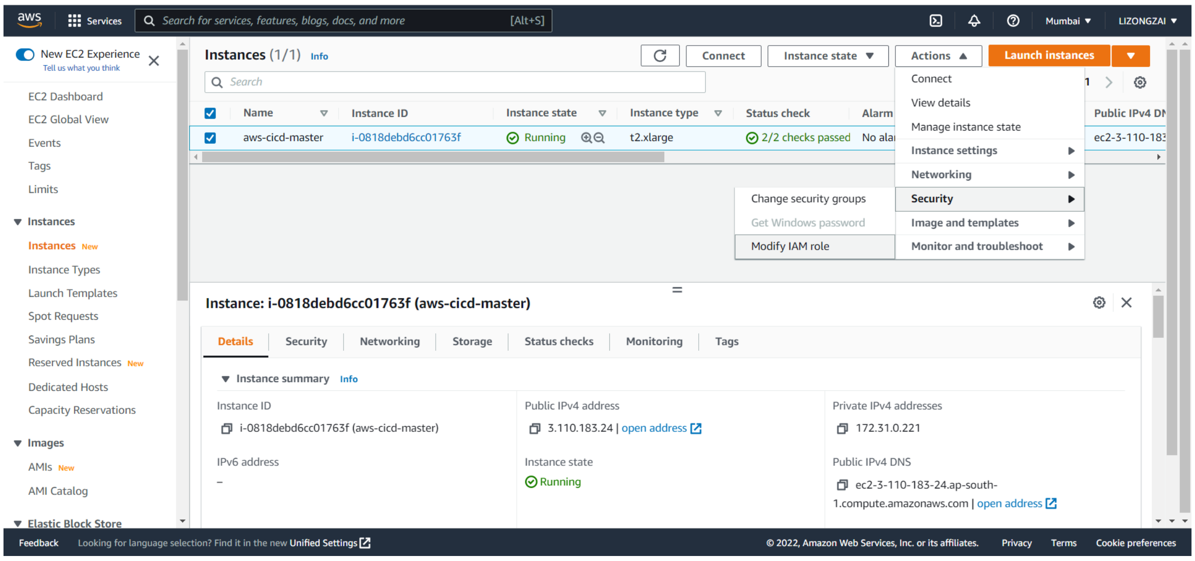
Filter policies by property or policy name and press enter

	Policy name	Type	Description
☐	AmazonEC2ContainerRegistryFullAccess	AWS managed	Provides administrative access to Amazon ECR resour...
☐	AdministratorAccess	AWS managed - job function	Provides full access to AWS services and resources.

Permissions boundary - (not set)

Set a permissions boundary to control the maximum permissions this role can have. This is not a

Feedback   Looking for language selection? Find it in the new Unified Settings   © 2022, Amazon Web Services, Inc. or its affiliates.   Privacy   Terms   Cookie preferences



Name	Instance ID	Instance state	Instance type	Status check	Alarm	Public IPv4 DN
aws-cicd-master	i-0818debd6cc01763f	Running	t2.xlarge	2/2 checks passed	No ala	ec2-3-110-183

Instance: i-0818debd6cc01763f (aws-cicd-master)

Instance summary Info

Instance ID
i-0818debd6cc01763f (aws-cicd-master)

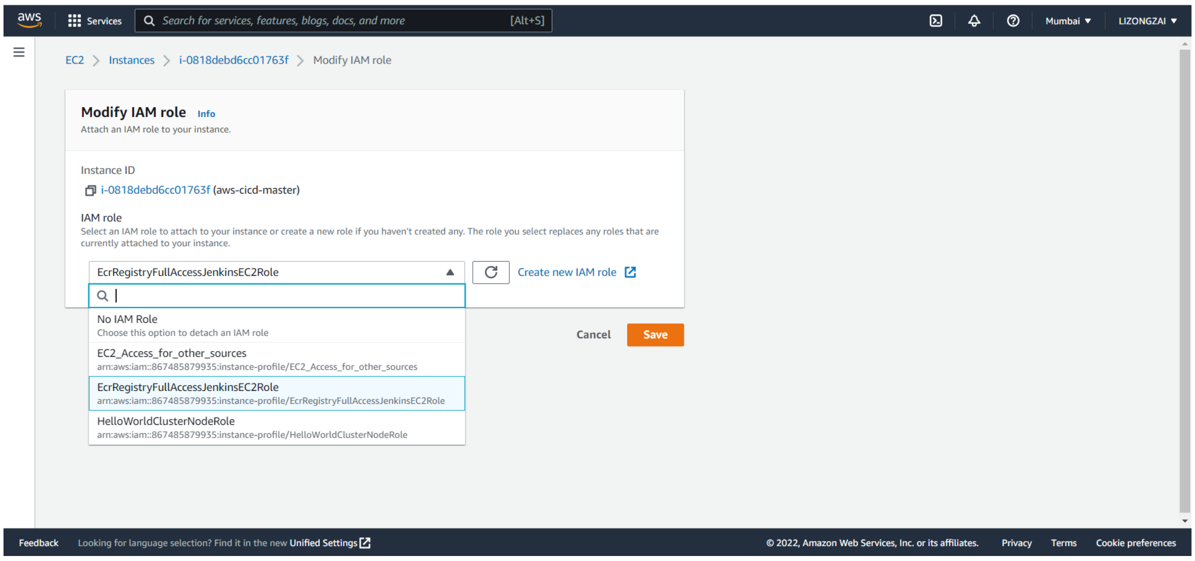
Public IPv4 address
3.110.183.24 | open address

Private IPv4 addresses
172.31.0.221

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-3-110-183-24.ap-south-1.compute.amazonaws.com | open address


Services
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
LIZONGZAI
EC2 > Instances > i-0818debd6cc01763f > Modify IAM role
Create new IAM role
Cancel
Save
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Modify IAM role Info
Attach an IAM role to your instance.

Instance ID
i-0818debd6cc01763f (aws-cicd-master)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

EcrRegistryFullAccessJenkinsEC2Role

No IAM Role
Choose this option to detach an IAM role

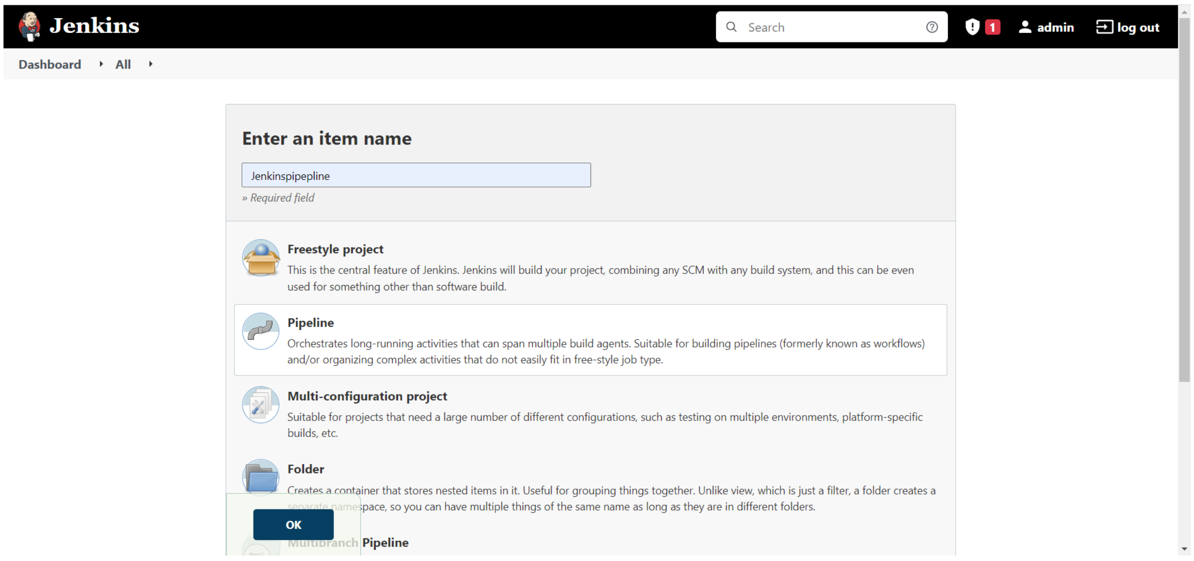
EC2_Access_for_other_sources
arn:aws:iam::867485879935:instance-profile/EC2_Access_for_other_sources

EcrRegistryFullAccessJenkinsEC2Role
arn:aws:iam::867485879935:instance-profile/EcrRegistryFullAccessJenkinsEC2Role

HelloWorldClusterNodeRole
arn:aws:iam::867485879935:instance-profile/HelloWorldClusterNodeRole

Creating Jenkins Pipeline

Step 1 – Create a pipeline in Jenkins, with your project name



Enter an item name

Jenkinspipeline

» Required field

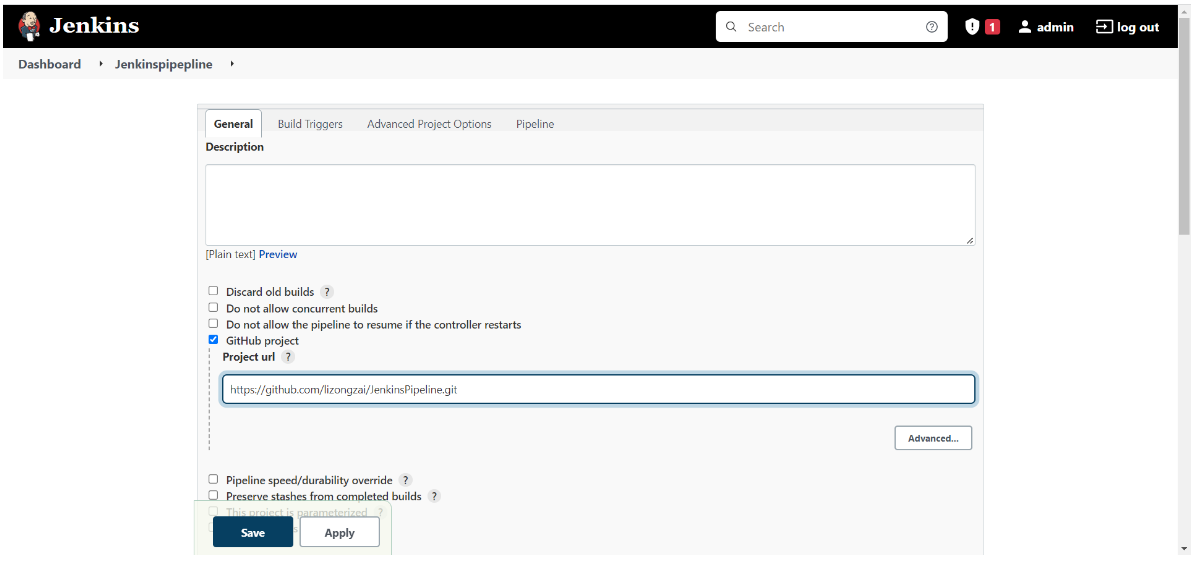
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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



Description

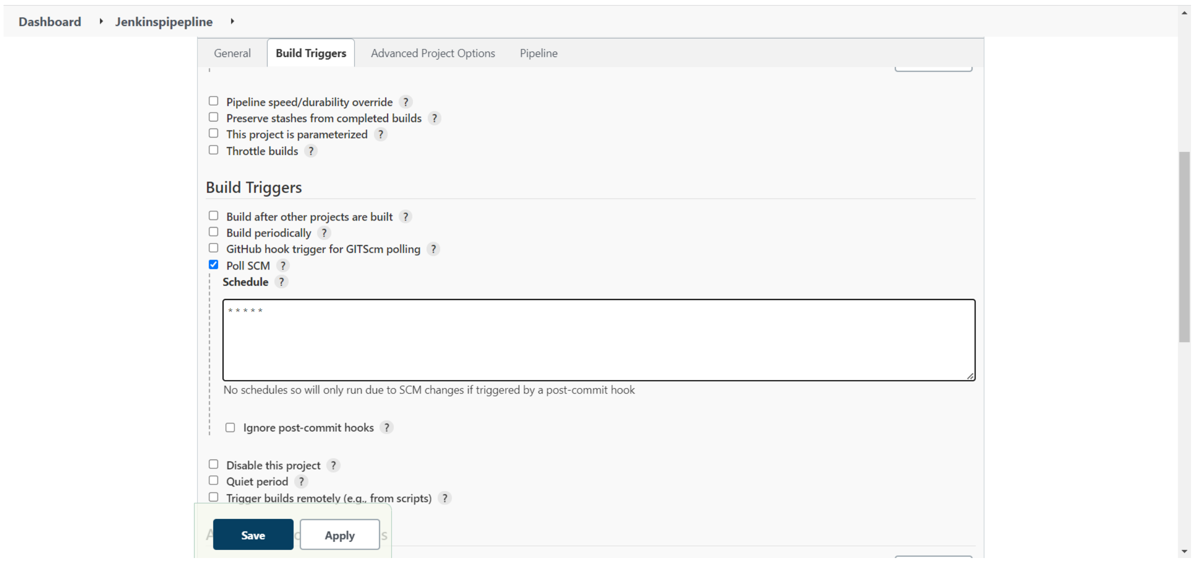
[Plain text] Preview

- [ ] Discard old builds
- [ ] Do not allow concurrent builds
- [ ] Do not allow the pipeline to resume if the controller restarts
- [x] GitHub project

Project url

https://github.com/lizongzai/JenkinsPipeline.git

- [ ] Pipeline speed/durability override
- [ ] Preserve stashes from completed builds
- [ ] This project is parameterized



- [ ] Pipeline speed/durability override
- [ ] Preserve stashes from completed builds
- [ ] This project is parameterized
- [ ] Throttle builds

Build Triggers

- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [x] Poll SCM

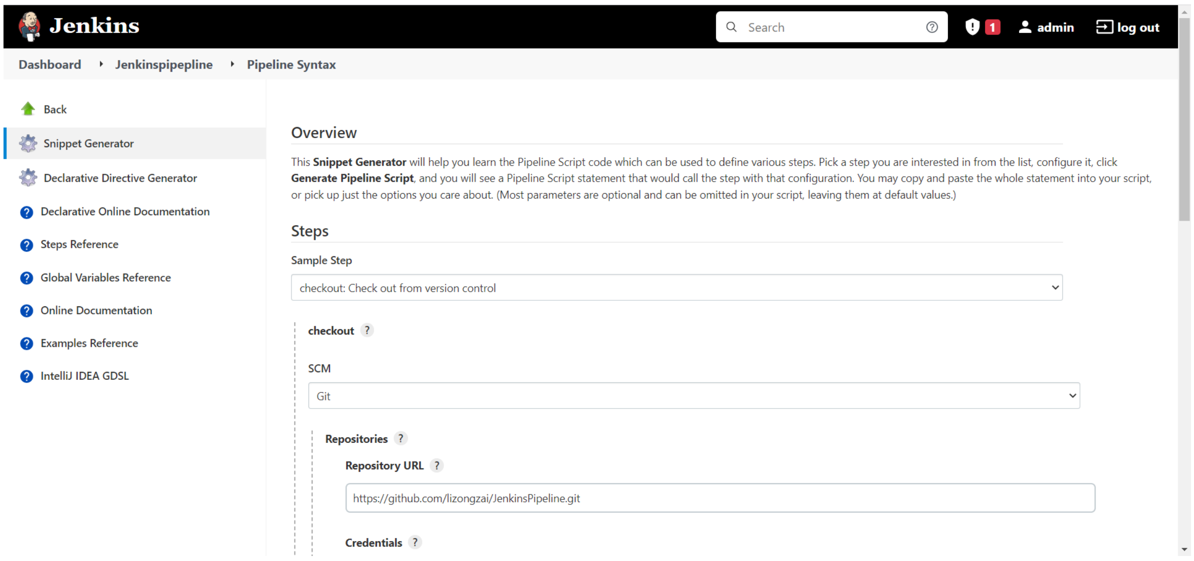
Schedule

```
* * * * *
```

No schedules so will only run due to SCM changes if triggered by a post-commit hook

- [ ] Ignore post-commit hooks
- [ ] Disable this project
- [ ] Quiet period
- [ ] Trigger builds remotely (e.g., from scripts)

`click on Pipeline Syntx`



Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step: checkout: Check out from version control

checkout

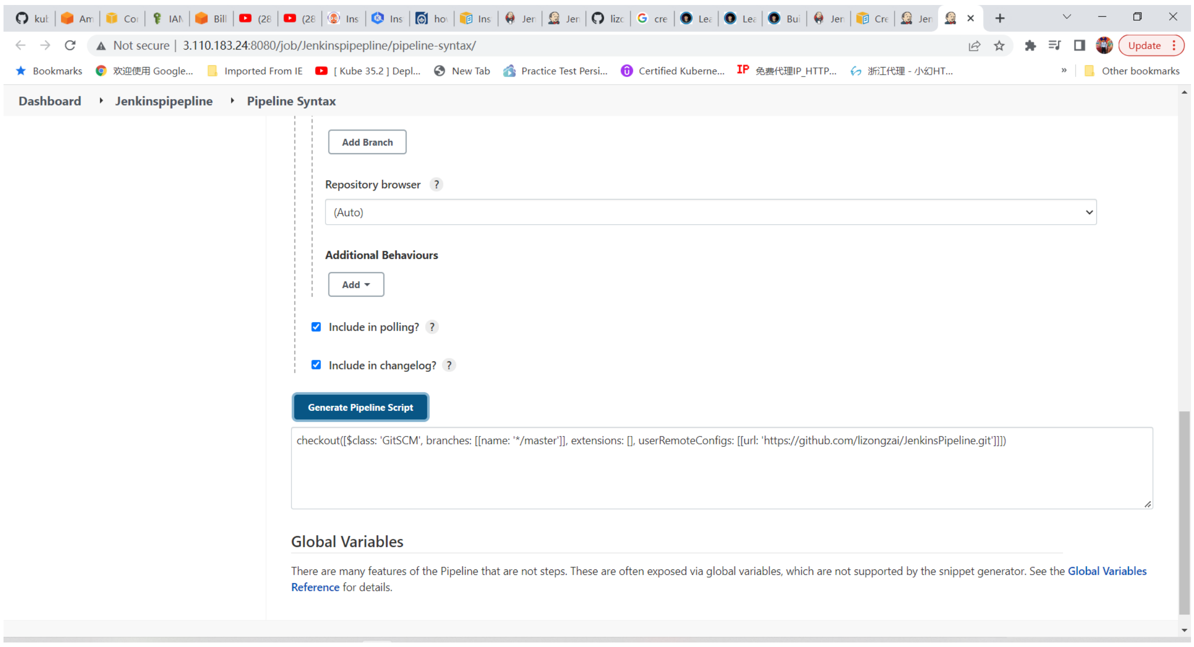
SCM: Git

Repositories

Repository URL: https://github.com/lizongzai/JenkinsPipeline.git

Credentials

`Generate Pipeline Script`


Not secure | 3.110.183.24:8080/job/Jenkinspipepline/pipeline-syntax/

Dashboard › Jenkinspipepline › Pipeline Syntax

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

☑ Include in polling? ?

☑ Include in changelog? ?

Generate Pipeline Script

```
checkout([$class: 'GitSCM', branches: [[name: '*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/lizongzai/JenkinsPipeline.git']]])
```

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the Global Variables Reference for details.

Step 2: Use below pipeline code

In place of “**YOUR_ACCOUNT_ID_HERE**” paste your AWS Account ID

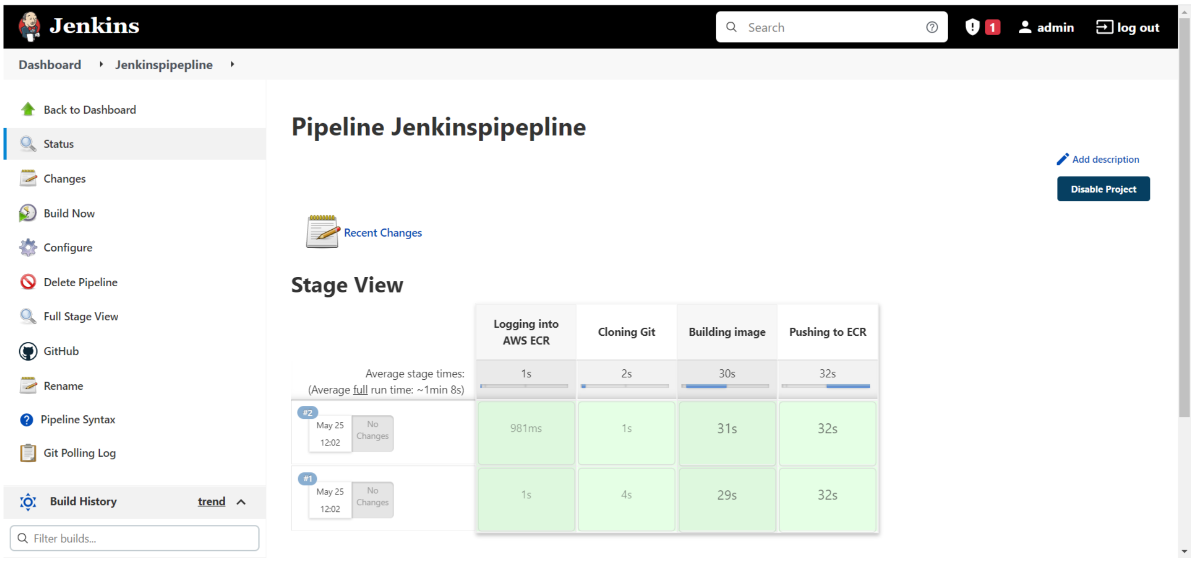
In place of “**CREATED_AWS_ECR_CONTAINER_REPO_REGION**” copy created ECR repo region id

“**IMAGE_REPO_NAME**” set your ECR repository name

“**IMAGE_TAG**” mention your desired tag

```
pipeline {
    agent any
    environment {
        AWS_ACCOUNT_ID="YOUR_ACCOUNT_ID_HERE"
        AWS_DEFAULT_REGION="CREATED_AWS_ECR_CONTAINER_REPO_REGION"
        IMAGE_REPO_NAME="ECR_REPO_NAME"
        IMAGE_TAG="IMAGE_TAG"
        REPOSITORY_URI =
"${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_DEFAULT_REGION}.amazonaws.com/${IMAGE_REPO_NAME}"
    }

    stages {

         stage('Logging into AWS ECR') {
            steps {
                script {
                sh "aws ecr get-login-password --region ${AWS_DEFAULT_REGION} | docker
login --username AWS --password-stdin
${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_DEFAULT_REGION}.amazonaws.com"
                }

            }
        }
        stage('Cloning Git') {
            steps {
                checkout([$class: 'GitSCM', branches: [[name: '*/master']],
doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [],
userRemoteConfigs: [[credentialsId: '', url:
'https://github.com/sd031/aws_codebuild_codedeploy_nodeJs_demo.git']]])
            }
        }

    // Building Docker images
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build "${IMAGE_REPO_NAME}:${IMAGE_TAG}"
        }
      }
    }

    // Uploading Docker images into AWS ECR
    stage('Pushing to ECR') {
     steps{
         script {
                sh "docker tag ${IMAGE_REPO_NAME}:${IMAGE_TAG}
${REPOSITORY_URI}:$IMAGE_TAG"
                sh "docker push
${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_DEFAULT_REGION}.amazonaws.com/${IMAGE_REPO_NAME}:${IMA
GE_TAG}"
         }
        }
      }
    }
}
```

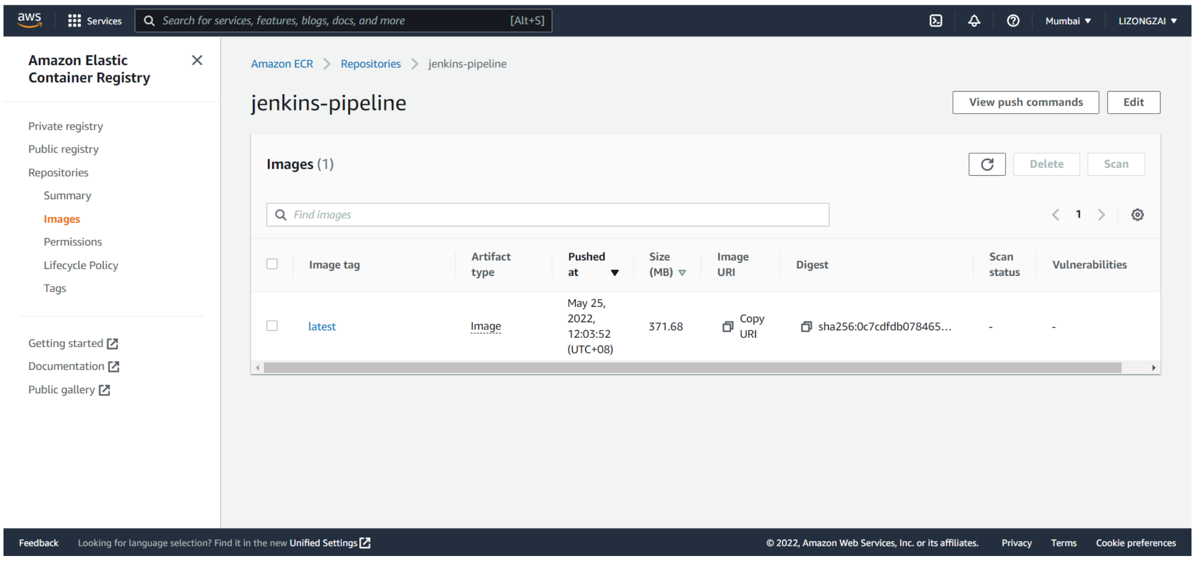
Step 3: Click on Build to see build happening properly and Docker image getting published to AWS ECR



Pipeline Jenkinspipepline

Stage View

	Logging into AWS ECR	Cloning Git	Building image	Pushing to ECR
Average stage times: (Average full run time: ~1min 8s)	1s	2s	30s	32s
#2 May 25 12:02 No Changes	981ms	1s	31s	32s
#1 May 25 12:02 No Changes	1s	4s	29s	32s


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
LIZONGZAI
Amazon Elastic Container Registry
Private registry
Public registry
Repositories
Summary
Images
Permissions
Lifecycle Policy
Tags
Getting started
Documentation
Public gallery
Amazon ECR > Repositories > jenkins-pipeline
View push commands
Edit
Delete
Scan
Find images

jenkins-pipeline

Images (1)

	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
	latest	Image	May 25, 2022, 12:03:52 (UTC+08)	371.68	Copy URI	sha256:0c7cdfdb078465…	-	-

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences