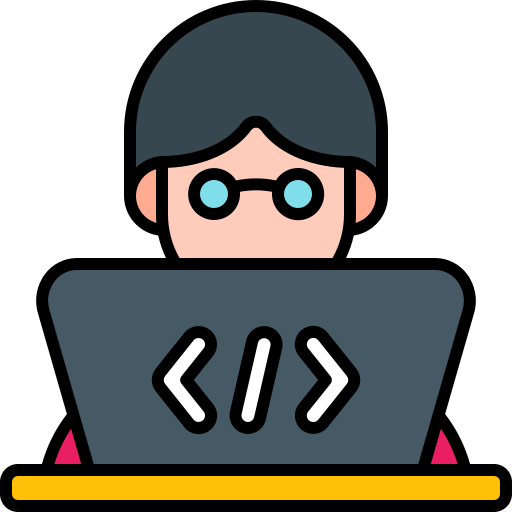
**JENKINS CI-CD WITH DOCKER, ECR & ECS**

**CONTINEOUS INTEGRATION (CI)**



**Developer(git)**



**GitHub**

Fetch Code

Build

Unit Test

Code Analysis

Upload Artifact



Docker Build



**Pre-requisites**

1. **AWS IAM user with ECR permission**
2. **Store AWS credentials in Jenkins**
3. **Create an ECR repository on AWS**
4. **Docker pipeline plug in has to be installed on Jenkins dashboard**
5. **Docker Engine has to be installed on Jenkins server**
6. **ECR plug in has to be installed on Jenkins dashboard**

**Steps:**

1. **Install Docker Engine in Jenkins server**
   1. **Add Jenkins user to docker group & reboot**
2. **Install AWS CLI**
3. **Create IAM user in AWS**
4. **Create ECR Repository in AWS**
5. **Plugins** 
   1. **Ecr, docker pipeline, aws sdk for credentials**
6. **Store AWS credentials in Jenkins**
7. **RUN the pipeline**

Install Docker Engine in Jenkins server

1. Login to Jenkins server
2. Run below commands to install Docker

$ curl -fsSL https://get.docker.com -o get-docker.sh

$ sh get-docker.sh

1. Add Jenkins user to Docker group

$ id jenkins

$ usermod -a -G docker jenkins

$ id jenkins

$ apt install awscli -y

$ reboot

AWS Setup

* Login to AWS console
* Search for IAM
* Click on “Users”
* Click “Add user”
* Give the username as “Jenkins”
* Select “Access Key – Programmatic access”
* Attach existing policy
* Search for “Registry”
* Select “AmazonEC2ContainerRegistryFullAccess”
* Search for “ecs”
* Select “AmazonECS\_FullAccess”
* Click next
* Create user
* Download .csv file
* Now create the ECR repository 🡪 search for ecr in the console search bar
* Select “Elastic Container Registry”
* Click “Create Repository”
* Keep it “Private”
* Give a name like “sscademyaapimg”
* Click “Create respository”

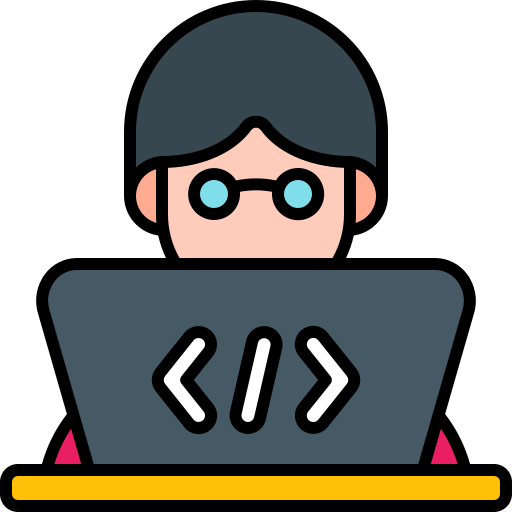
Jenkins Plugins setup

* Login to Jenkins dashboard
* Click on “Manage Jenkins”
* Click on “Manage plugins”
* Click on “Available”
* Search for “Docker pipeline”
* Search for “Amazon ecr”
* Search for “Amazon Web Services SDK::All”
* Search for “CloudBees Docker Build and Publish”
* Click install without restart

Store AWS credentials in Jenkins

* Go to Jenkins dashboard
* Manage Jenkins
* Manage Credentials
* Click on Jenkins
* Click on “Global credentials”
* Click Add credentials
* Select “AWS credentials”
* Give name as “awscreds”
* Give “Access key ID” & “Secret access key” which we have downloaded while creating the IAM user

**CONTINEOUS DEPLOYMENT (CD)**



**Developer(git)**



**GitHub**

Fetch Code

Build

Unit Test

Code Analysis

Upload Artifact



Docker Build



**Container Hosting Platforms**

* Docker Engine
* Kubernetes
  + Standalone, EKS, AKS, GKE, Openshift etc
* AWS ECS: Elastic Container Service
  + <https://aws.amazon.com/ecs/>

**Pre-requisites**

1. Create ECS cluster
2. Create ECS service
3. Install Jenkins plugin – pipeline: aws steps
4. Install AWS cli in Jenkins server
5. **Create ECS Cluster**

* Login to AWS cloud
* Search for “ecs” 🡪 Elastic Container Service
* Switch to “New ECS Experience”
* Click on “Get Started”
* Click on “Create Cluster” 🡪 give a name for the cluster
* Keep all the default options 🡪 select “AWS Fargate (serverless)
* In Monitoring section 🡪 Select “Use container Insights” option enabled
* Click on Create
* After the cluster is created 🡪 go to  option in the left
* Go to “Task definitions” – Click on “Create new task definition” 🡪 give name “sscademyapptask”
* Give container name as “sscademyapp” 🡪 copy & paste the URI of the docker image from our ECR
* Type the container port as “8080”, since we are running our Tomcat container on port “8080”
* Click Next 🡪 Keep all the options as default and click Next
* Click on Create
* Click on  🡪 go to “Clusters” 🡪 Click on our cluster which we have created
* Select “Service” 🡪 Click on “Create” 🡪 Application type as “Service”
* Select “Family” 🡪 “sscademyapptask” 🡪 Give service name as “sscademyappsvc”
* Click on “Load balancing option “
* Select “Application Load Balancer” 🡪 Create a new load balancer 🡪 sscademyappelbecs
* Give port number as “80” on which Load balancer listen
* Give Target group name as “sscademyecstg” 🡪 protocol “http”
* Health check path “ /login “
* Click on “Networking” 🡪 Create a new security group 🡪 type name as “sscademyappelb-sg”
* Click “Add new rule” 🡪 Type “http” 🡪 Source “Anywhere”
* Click on “Create”
* Go to “EC2”
* Go to “Target groups”
* Click on the Target group which we have created
* Click on “Health checks” 🡪 Click on “Edit”
* Click on “Advance health check settings”
* Change the port from “Traffic port” to “Override”
* Type port number as “8080”
* Change the “Healthy threshold” to 2
* Save the changes
* Go to “Security Groups” under “EC2” service
* Find our Security Group 🡪 Click on “Inbound rule” 🡪 Edit 🡪 Add rule 🡪 port 8080 🡪 Allowed from Anywhere
* Save rule
* Come back to ECS 🡪 click on our ECS 🡪 click on Networking 🡪 click on “Open address” under load balancer

**Install Jenkins plug in and Run the PAAC**

* Copy the cluster name which we have given in AWS ECS cluster & paste it in our PAAC
* Copy the service name as well and paste it in our PAAC
* Install the AWSCLI on Jenkins server if it is not installed
  + $ sudo apt install awscli -y
* Login to Jenkins dashboard 🡪 Manage Jenkins 🡪 Manage Plugins
* Click on Available 🡪 search for “Pipeline: AWS Steps” 🡪 Install without restart
* Now create a new job 🡪 Click on New Item
* Give name for the Pipeline 🡪 CICD-with-Docker
* Copy paste our PAAC
* Go to AWS ECS 🡪 go to Tasks 🡪 Go to the running task 🡪 note down the container ID
* Go to Jenkins and run the pipeline