Step-by-Step Documentation: Multi-Branch CI/CD Pipeline using Jenkins, GitHub, and AWS EKS

Prerequisites:

Jenkins Server – Set up with the necessary plugins:

- Pipeline
- Git
- GitHub plugin
- Kubernetes plugin
- Docker plugin (if Docker is part of the pipeline)
- Multibranch Pipeline plugin

1- GitHub Repository – A GitHub repo with multiple branches (11 branches for 11 microservices). Fork the repo for use

Source code : Github repo

2- Service Account for RBAC – Configured to manage access for Jenkins.

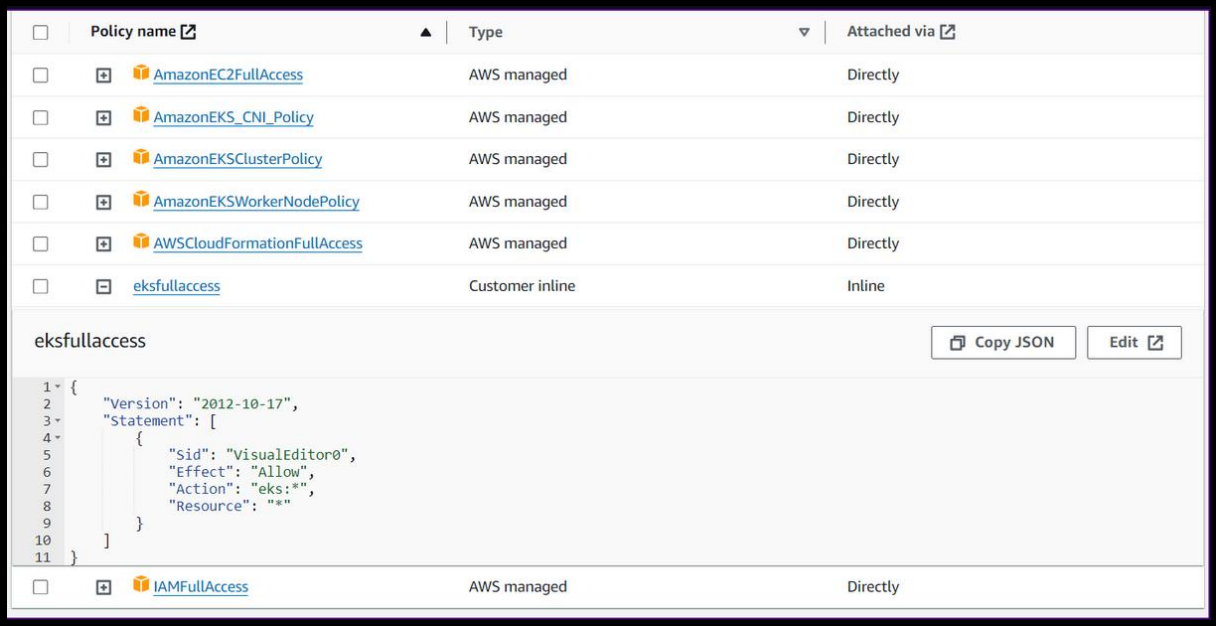
3- Docker Registry (e.g., Docker Hub) – For storing Docker images.

4- kubectl and eksctl installed – For EKS management.

EKS installation

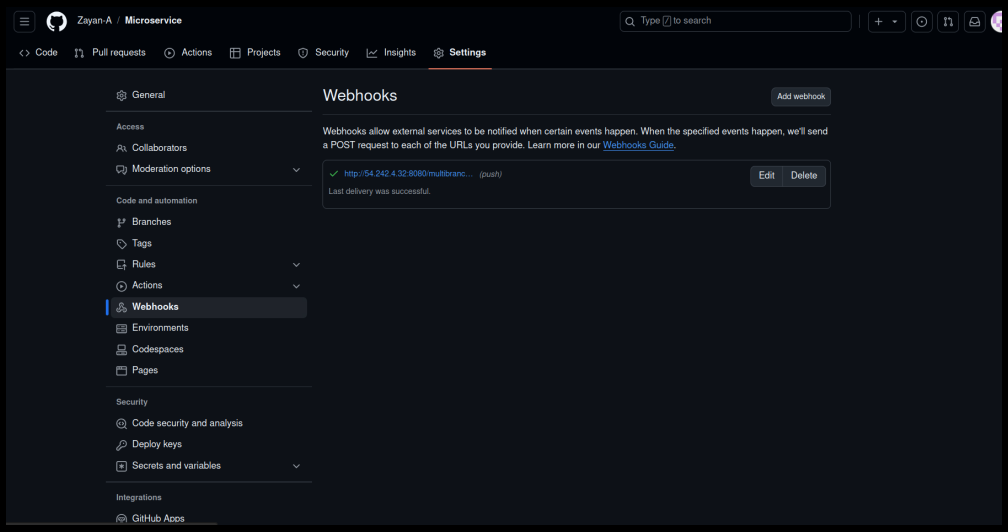
AWSCLI installed awscli installations

IAM Role with the following policies:



Step 1: Set up GitHub Webhook for Jenkins

1. **In GitHub:**
   - Go to **Settings** > **Webhooks**.
   - Add a new webhook with the following details:
     - **Payload URL**: `http://<your-jenkins-server>/github-webhook/`
     - **Content type**: `application/json`
     - Choose to send the webhook on **push events** or **pull requests**.



2. **In Jenkins:**
   - Enable the **GitHub plugin** to allow Jenkins to trigger builds based on the webhook.
   - Go to **Manage Jenkins** > **Configure System** > **GitHub** section and add your GitHub repository details.

Step 2: Create Multibranch Pipeline in Jenkins

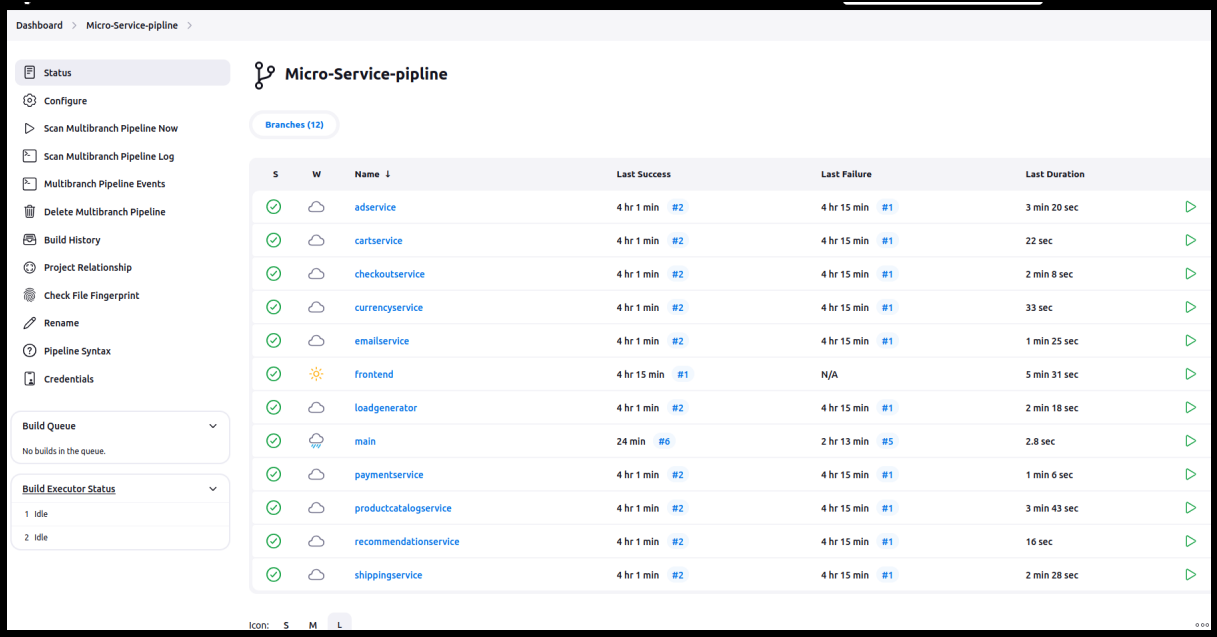
1. **In Jenkins:**
    - Go to **New Item** and select **Multibranch Pipeline**.
    - Provide a name for the pipeline (e.g., `Multi-Microservice-Pipeline`).
    - Under **Branch Sources**, add your **GitHub** repository:
        - Configure the GitHub credentials and URL to the repository.
        - Set it to discover all branches (11 branches for each microservice).
2. **Branch Strategy**:
    - Jenkins will automatically detect the branches and create a job for each microservice.

Step 3: Define Jenkinsfile for each Microservice

I have jenkins file for every microservice in my repo already, and your jenkins pipeline will automatically pick it up as soon as you configure the webhook properly
You can view these files in every branch named as **Jenkinsfile**

You will need to change the image build name and push name in every branch in the github with your own docker hub repo name

You will have to change the image name with your docker hub image name (username/reponame:latest) in deployment.yaml file in main branch



Step 4: Set up AWS EKS Cluster with RBAC Service Account

AWS EKS Cluster.

Change region, keyname with your own

```
eksctl create cluster --name=EKS-1 \

                     --region=ap-south-1 \

                     --zones=ap-south-1a,ap-south-1b \

                     --without-nodegroup

eksctl utils associate-iam-oidc-provider \

   --region ap-south-1 \

   --cluster EKS-1 \

   --approve



eksctl create nodegroup --cluster=EKS-1 \

                      --region=ap-south-1 \

                      --name=node2 \

                      --node-type=t3.medium \

                      --nodes=3 \

                      --nodes-min=2 \

                      --nodes-max=4 \

                      --node-volume-size=20 \

                      --ssh-access \

                      --ssh-public-key=DevOps \

                      --managed \

                      --asg-access \

                      --external-dns-access \

                      --full-ecr-access \
```

```
                        --appmesh-access \

                        --alb-ingress-access
```

Create a Service Account for Jenkins in AWS EKS:

- Create a service account with the necessary RBAC permissions for Jenkins.

```
apiVersion: v1
kind: ServiceAccount
metadata:
 name: jenkins-sa
 namespace: default  # Change this to the namespace where Jenkins is
deployed
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 name: jenkins-limited-cluster-role
rules:
 - apiGroups: ["apps"]
   resources: ["deployments"]
   verbs: ["get", "list", "create", "delete", "patch"]
 - apiGroups: [""]
   resources: ["services", "pods", "namespaces"]
   verbs: ["get", "list", "create", "delete", "patch"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
 name: jenkins-cluster-role-binding
subjects:
 - kind: ServiceAccount
   name: jenkins-sa
   namespace: default
roleRef:
 kind: ClusterRole
 name: jenkins-limited-cluster-role  # Ensure this matches the correct
ClusterRole name
 apiGroup: rbac.authorization.k8s.io
```

Configure Jenkins to Use the Service Account:

- In Jenkins, add a Kubernetes credential for the service account.
- Navigate to Manage Jenkins > Manage Credentials > Global and add the service account token as a secret text.

Step 5: Build and Deploy Microservices to EKS

I have provided the deployment.yaml file in the main branch in the repo, it is already configured for deploying on the eks cluster you can review it for further details

Step 6: Verify Deployment

1. **Check Pods and Services in EKS**:
    - After the pipeline completes, verify the deployment by checking the status of the pods and services in your EKS cluster:
2. **Bash command**

```
kubectl get pods

kubectl get svc
```

Access the Webapp using the Load Balancer arn

Conclusion

By following this guide, you have set up a multi-branch Jenkins CI/CD pipeline that automatically triggers builds and deployments for each microservice branch. The microservices are built, containerized, pushed to a Docker registry, and deployed to AWS EKS using Kubernetes manifests, with RBAC for secure access.

If you provide your own GitHub repository link, you can customise the `Jenkinsfile` and Kubernetes manifests according to the specific microservices.

Activities Firefox Web Browser 12 سبتمبر 9:23 ش

Microservice/Jenkinsfile · ChatGPT (27) WhatsApp Docker Hub Container Im Instances | EC2 | us-east-1 Online Boutique

adcdaee732c95466e8d4c1879b5fa347-2091182776.us-east-1.elb.amazonaws.com/cart

ONLINEBOUTIQUE

$ USD

local

Cart (1)

Empty Cart | Continue Shopping

Mug

SKU #6E92ZMYYFZ

Quantity: 1 $8.99

Shipping $8.99

Total $17.98

Shipping Address

E-mail Address
someone@example.com

Street Address
1600 Amphitheatre Parkway

Zip Code
94043

City
Mountain View

State
CA

Country
United States

Payment Method

Credit Card Number
4432-8015-6152-0454

Month
January

Year
2025

CVV
•••

Place Order