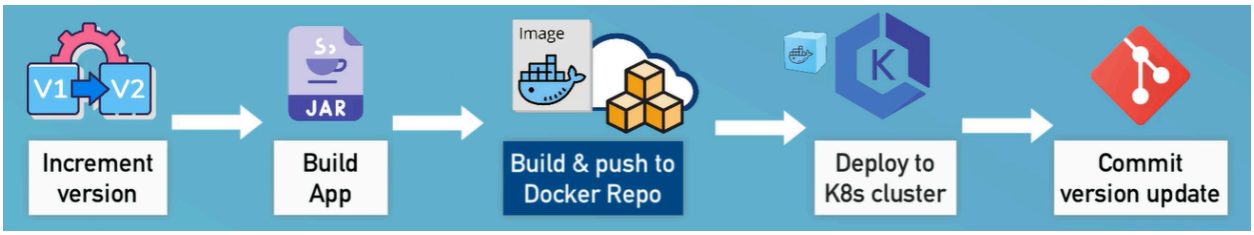
# Demo Project: Complete CI/CD Pipeline with EKS and AWS ECR

In this guide, we focus on setting up the AWS ECR repository, configuring Jenkins credentials and secrets, updating the Jenkinsfile to push images to ECR, and finally executing the Jenkins pipeline.






## Step 1: Create ECR Repository

1. **Navigate to AWS ECR:**
   - Open the AWS Management Console.
   - Go to **Elastic Container Registry (ECR)**.
2. **Create a New Private Repository:**
   - Click **Create repository**.
   - Set the repository name to `java-maven-app`.
   - Leave the default settings (or adjust according to your requirements).

- Click **Create repository**.

---

# Step 2: Create Credentials in Jenkins

1. In the terminal retrieves the authentication password for logging into an AWS ECR:

   `aws ecr get-login-password --region us-east-1`

2. In the Jenkins UI, navigate to **Manage Jenkins → Credentials → Global credentials**.
3. Click **Add Credentials**.
   - Username: AWS
   - Password: Paste from `aws ecr get-login-password --region us-east-1`
   - ID: `ecr-credentials`

---

# Step 3: Create Secret for AWS ECR

- **Create the Secret:**
  - Use the command below to create a Docker registry secret that Kubernetes will use to pull images from AWS ECR:

    ```
    kubectl create secret docker-registry aws-registry-key \
    --docker-server=038462748802.dkr.ecr.us-east-1.amazonaws.com \
    --docker-username=AWS \
    --docker-password=<paste-authentication-token>
    ```

  - Replace `<paste-authentication-token>` with the token you obtained from `aws ecr get-login-password --region us-east-1`.

---

# Step 4: Update Jenkinsfile

1. **Create a New Branch for Deployment:**
    - From your `jenkins-jobs` branch, create a new branch (e.g., `jenkins-jobs-AWS`): `git checkout -b jenkins-jobs-AWS`

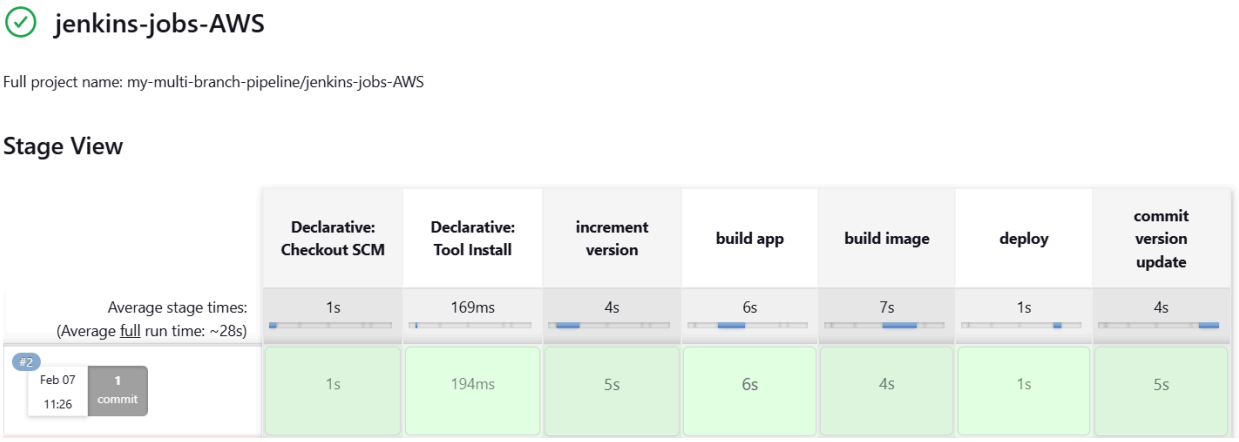
2 . **Update the Jenkinsfile:**
    - Modify the deploy stage in your Jenkinsfile to push to AWS ECR and deploy to your EKS cluster.

3. **Commit and Push Changes:**
    - Commit your updated Jenkinsfile to the `jenkins-jobs-AWS` branch and push it to your repository.

---

# Step 5: Execute Jenkins Pipeline

1. **Update Jenkins Multi-Branch Pipeline:**
    - In the Jenkins UI, update your multibranch pipeline configuration to include the new branch (`jenkins-jobs-AWS`).
    - Save the configuration.
2. **Trigger the Pipeline:**
    - Manually trigger the pipeline from the Jenkins UI.
    - Monitor the console output to ensure all stages complete successfully.




3. **Verify the Deployment in EKS:**

- Check the AWS ECR console the image with it’s version.
- Check that the pods are running: `kubectl get pod`
- To see details and confirm that the image was pulled from AWS ECR, run:

  `kubectl describe pod <pod id>`

  - Example:

```
kubectl describe pod java-maven-app-77cd7f669f-57db
...
Containers:
  java-maven-app:
    Image:        038462748802.dkr.ecr.us-east-1.amazonaws.com/java-maven-app:1.1.4-2
```

---

# Step 6: Clean up

1. **Delete the Deployment (if needed):** `kubectl delete deployment <deployment-name>`
2. **Optionally, Delete the EKS Cluster:** `eksctl delete cluster --name <your-cluster-name> --region <your-region>`

---