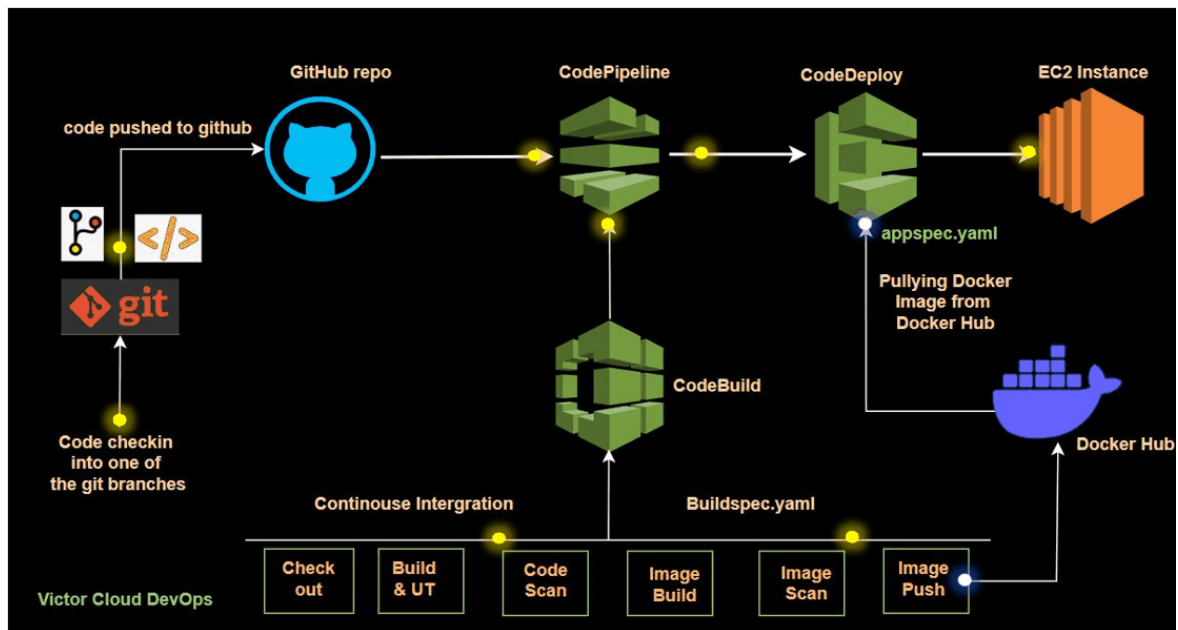


## Deploy Java Project using Codepipeline



## What is CodePipeline..??

AWS CodePipeline is a continuous delivery service provided by Amazon Web Services (AWS) that helps you automate the software release process for your applications.

with AWS codepipeline , you can model, Visualize, and automate the different stages of your software delivery process, from building and testing to deploiting and monitoring

It enables you to rapidly and reliably deliver features and updates to your customers, while reducing the risk of errors and downtime.

First of all, what is the CI/CD? Following definition is this

A CI/CD pipeline is a series of automated steps that software development teams use to create, test, and deploy applications. CI/CD stands for continuous integration and continuous delivery or deployment.

- **continuous integration (CI):**

Continuous integration is the practice of frequently merging code changes into a central repository. This allows developers to identify issues early.

- **Continuous delivery (CD):**

Automatically releases the application to its intended environment.

**CI/CD pipelines help software development teams:**

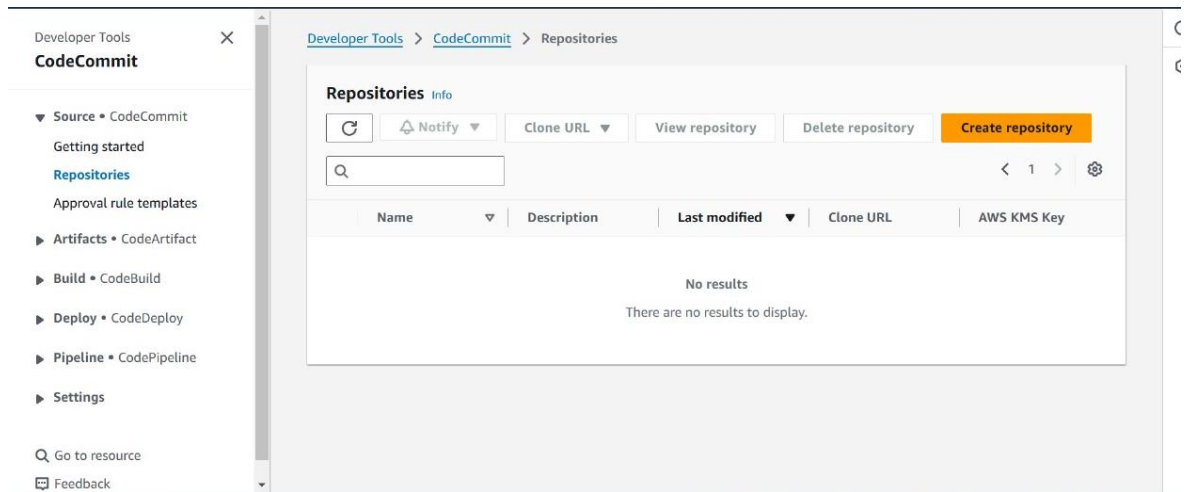
- Improve software quality
- Speed up delivery
- Reduce the risk of errors
- Detect bugs early
- Fix bugs promptly

**Maintain a continuous cycle of software development and updates**

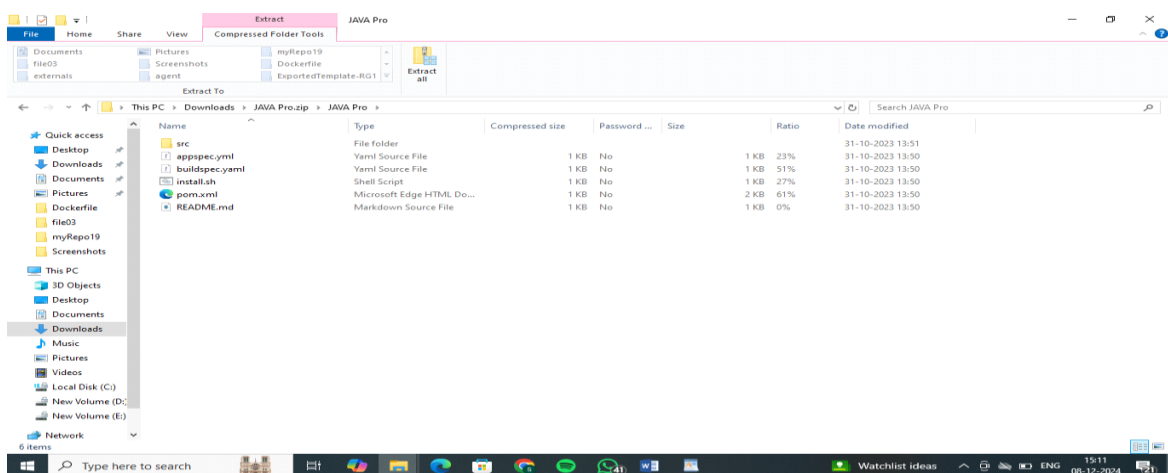
**CI/CD is a foundational part of DevOps, which helps shorten the software development lifecycle.**

# Step1:

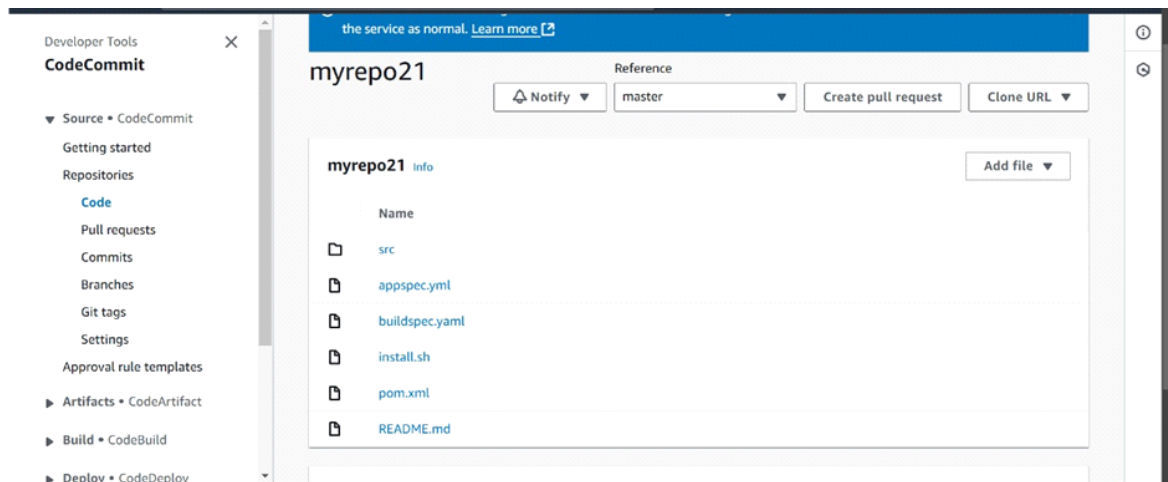
Create repository in code commit



- Create folder on desktop and clone the repo here
- Open the repo and paste the java project inside it

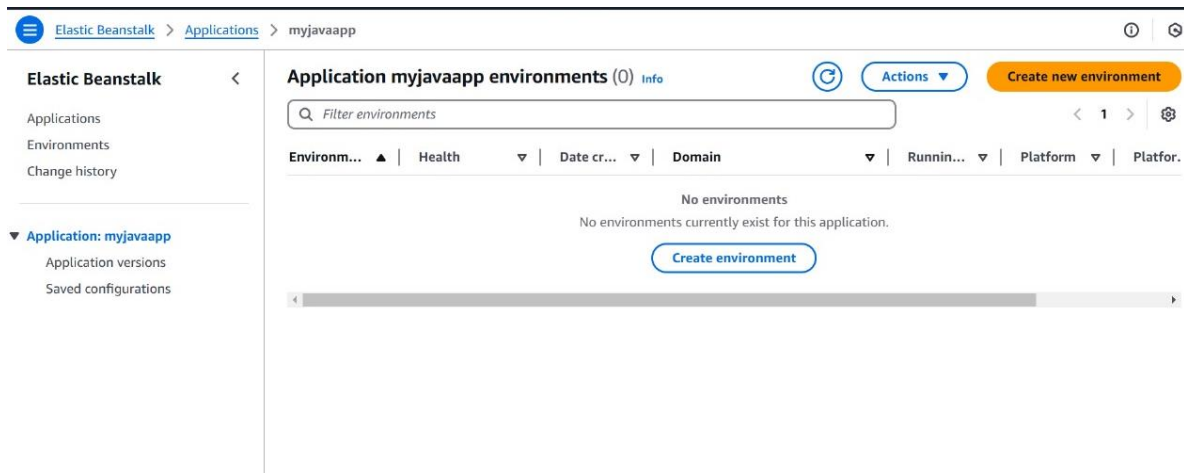


- `cd repo`
- `git add .`
- `git commit -m "msg"`
- `git push`

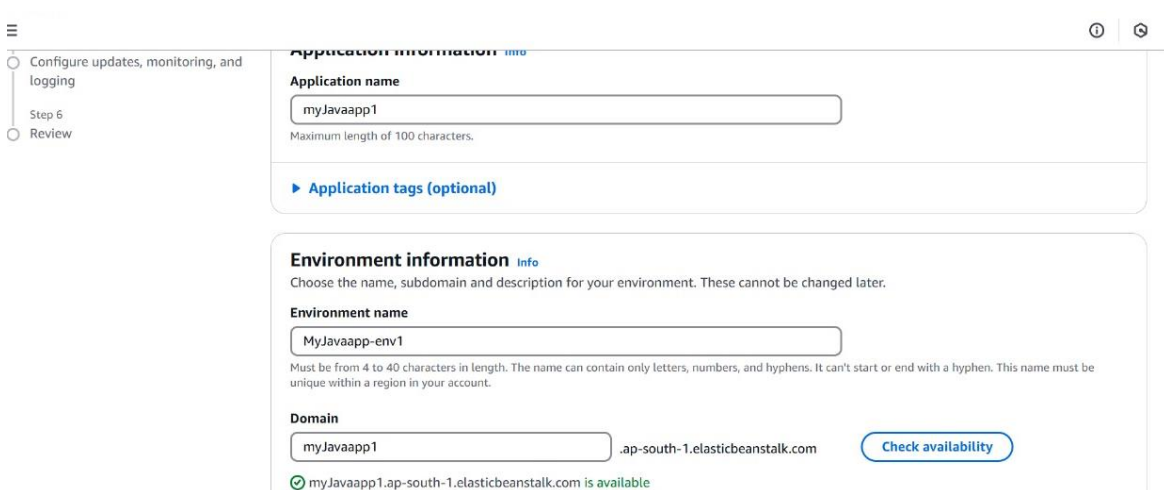


## Step:2

now search Elastic Beanstalk in services



Create application – provide app name – myjavaapp—give environment name – domain as myjavaapp (check availability )-- choose platform as tomcat – choose platform number latest (tomcat) –next—



in services access choose create and new service role – select Default VPC -- tick the activated checkbox – tick all availability zone –next – under EC2 instance profile – next—select Default vpc— tick the activated checkbox – tick all availability zone – next—Under EC2 security group select Linux security group – remove the default instance type and t2.micro –(Note if t2.micro is not available then select t3.micro)—Next—select enhanced in monitoring – under cloudwatch custom matrices— select “instance health” —under cloudwatch custom matrices select ApplicationRequestTotal— Next—submit

## Step:3

Search code pipeline—create pipeline –provide name – next—select repository name and branch name as master – next—

Step 1

Choose creation option

Step 2

Choose pipeline settings

Step 3

**Add source stage**

Step 4

Add build stage

Step 5

Add deploy stage

Step 6

Review

Step 3 of 6

Info

Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

Repository name

Choose a repository that you have already created where you have pushed your source code.

myrepo21

Branch name

Choose a branch of the repository

master

Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

Amazon CloudWatch Events (recommended)

AWS CodePipeline

under build provider AWS code build—click on create project—under new open window provide the project name – os as Linux –under build spec select us build spec file—untick the cloud watch logs—continue to code pipeline – next—under deploy provide – select AWS elastic beanstalk – select application name and enviornment name—next.

This is not actually a step. Just shows us all the settings we selected previously in each stage. Make sure everything is okay and click on create pipeline  
Now you can see your pipeline and it starts automatically when we create.

Developer Tools

CodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Developer Tools > CodePipeline > Pipelines > mypipeline21

mypipeline21

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2

Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: e60692e6-76da-4e72-aaac-29ff8d4b48d0

Source

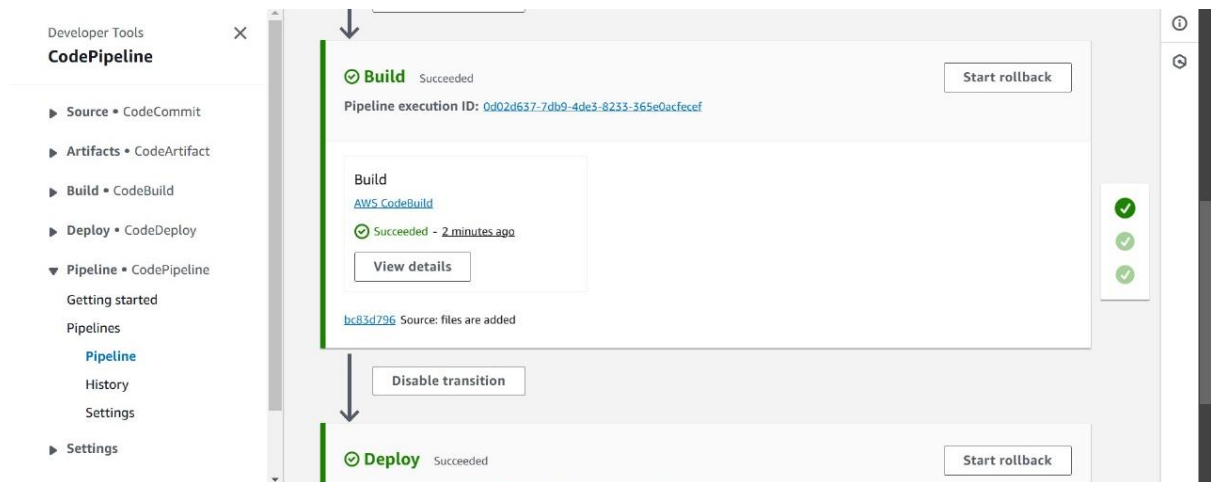
AWS CodeCommit

Succeeded - 1 minute ago

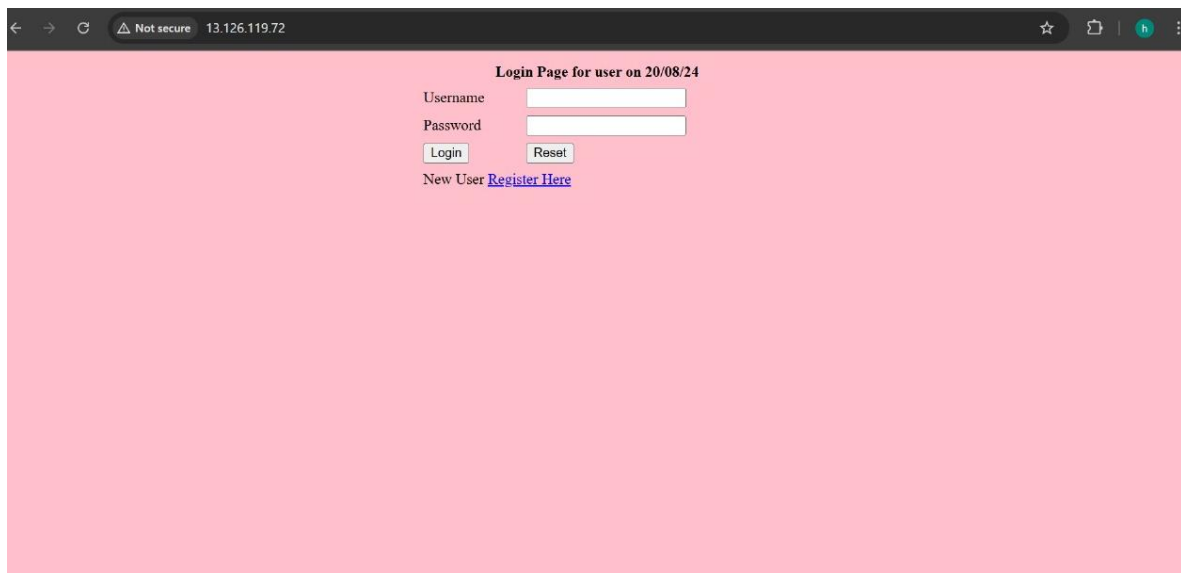
452160x8

View details

452160x8 Source: changes are done



now search ec2 under services and copy paste the public IP of instance in new tab



edit the index file in remote repo – go in repo name /src/main/webapp/index.jsp—select/edit and changes something – provide author name and email address and message –commit changes—after success refresh the index page – now date will be updated — Your web application should live now.

← → ↻ ⚠ Not secure 13.126.119.72 ☆ 📄 h ⋮

**Login Page for user on 4/12/24**

Username

Password

New User [Register Here](#)