Group 2

App URL http://164.90.161.120:5173/

Github url https://github.com/WarrenG123/calculator

Version Control Systems (VCS)

VCS is a software tool that allows managing changes to source code and tracking different versions of files over time.

VCS keeps a record of every change made to the source code, allowing developers to view the history, revert to previous versions, and collaborate effectively.

VCS provides features like branching and merging, enabling parallel development, experimentation, and isolation of changes.

- Some popular VCSs include:
    - Git
    - Mercurial
    - Subversion

-VCS is essential for teams working on software projects to maintain code integrity, track changes, and collaborate efficiently.

Source Code Managers

Source code managers are platforms or services built on top of version Control Systems providing additional features and functionalities for managing source code.

Source code managers offer a web-based interface and tools that enhance collaboration, project management, and code quality.

-They often include features such as issue tracking, code review, continuous integration, and deployment automation.

Source code managers make it easier for teams to collaborate, discuss code changes, manage project milestones, and track issues or feature requests.

-Popular source code managers include GitHub, GitLab, and Bitbucket, which provide a range of features to facilitate collaboration and project management.

- Some popular SCMs include:
    - GitHub
    - Gitlab
    - Bitbucket

Benefits of using a VCS or SCM

- Source Code Organization: SCM tools provide a structured and organized approach to managing source code files within a project, making it easier to locate, modify, and maintain code files.
- Collaboration and Code Review: SCM platforms often include collaboration features, such as pull requests, code reviews, and comments.
- Continuous Integration and Delivery (CI/CD): SCM platforms integrate with CI/CD pipelines, automating the build, test, and deployment processes.
- Collaboration: VCS facilitates collaboration among team members by allowing concurrent work on the same files, managing conflicts, and providing mechanisms for merging changes made by multiple contributors.
- History Tracking: VCS allows you to track the complete history of changes made to files, enabling you to review previous versions, understand modifications, and revert to earlier states if needed.
- Experimentation and Versioning: VCS allows developers to experiment with different ideas, create branches for new features, and maintain a version history that can be used for documentation, releases, and maintaining different product versions.
- Traceability: VCS allows you to track who made specific changes, when they were made, and why. This is valuable for accountability, auditing, and troubleshooting.
- Rollback and Recovery: With VCS, you can easily roll back to a stable or previous version in case of issues or bugs.

AWS (Amazon Web Services)

A comprehensive cloud computing platform offering a wide range of services for computing power, storage, databases, networking, and more.

Provides services like Amazon EC2 for virtual server instances, Amazon S3 for object storage, Amazon RDS for managed databases, and AWS Lambda for serverless computing.

Offers scalability, flexibility, and a vast ecosystem of services, making it suitable for a variety of applications and workloads.

Azure (Microsoft Azure)

A cloud computing platform by Microsoft that provides a wide range of services similar to AWS.

Offers services such as Azure Virtual Machines, Azure Blob Storage, Azure SQL Database, and Azure Functions for serverless computing.

Provides integration with other Microsoft tools and technologies, making it appealing for organizations already using Microsoft products.

Digital Ocean

A cloud infrastructure provider focused on simplicity and developer-friendly features.

Offers scalable virtual machines called Droplets, managed databases, block storage, and load balancers.

Provides an intuitive user interface, straightforward pricing, and easy-to-use APIs, making it popular among developers and small to medium-sized businesses.

GCP (Google Cloud Platform)

Google's cloud computing platform offers a wide range of services and tools.

Provides services like Google Compute Engine for virtual machines, Google Cloud Storage for object storage, Google Cloud SQL for managed databases, and Google Cloud Functions for serverless computing.

Emphasizes data analytics, machine learning, and AI capabilities, and integrates well with other Google services.

Heroku

A cloud platform that simplifies application deployment and management.

Supports multiple programming languages and frameworks, making it versatile for different application types.

Offers a straightforward deployment process, automatic scaling, and managed services, allowing developers to focus on code rather than infrastructure.

Render

A modern cloud platform designed for developers to deploy web applications and static sites.

Provides a streamlined and simple deployment experience with features like automatic SSL, built-in static site hosting, and automatic scaling.

Offers support for popular programming languages, databases, and frameworks, making it suitable for various types of applications.

Each platform has its own strengths, pricing models, and target audiences, so it's important to evaluate them based on your specific project requirements and preferences.

Other uses of the CI

Automated Testing: CI enables automated execution of tests to identify bugs and compatibility issues.

Code Quality Analysis: CI performs code analysis to enforce coding standards and identify potential issues.

Continuous Code Reviews: CI facilitates code reviews for collaboration, feedback, and best practices.

Continuous Documentation Generation: CI automates the generation of up-to-date documentation for the project.

Performance and Load Testing: CI includes tests to assess application performance under stress.

Security Scanning: CI integrates security scanning tools to identify vulnerabilities and insecure practices.

Continuous Monitoring and Feedback: CI integrates with monitoring tools to provide feedback on application health and performance.

Steps to create a pipeline with Jenkins Blue Ocean

Install Jenkins.

Install the Blue Ocean plugin.

Create a new pipeline.

Connect to your source control system.

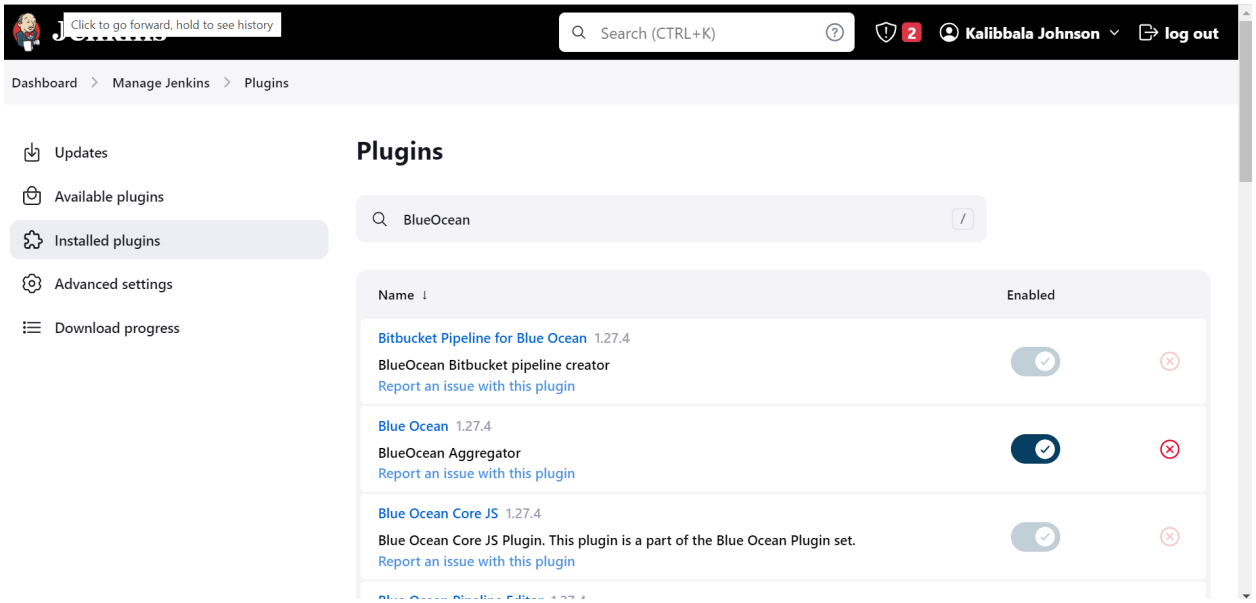
Select the repository and branch for the pipeline.

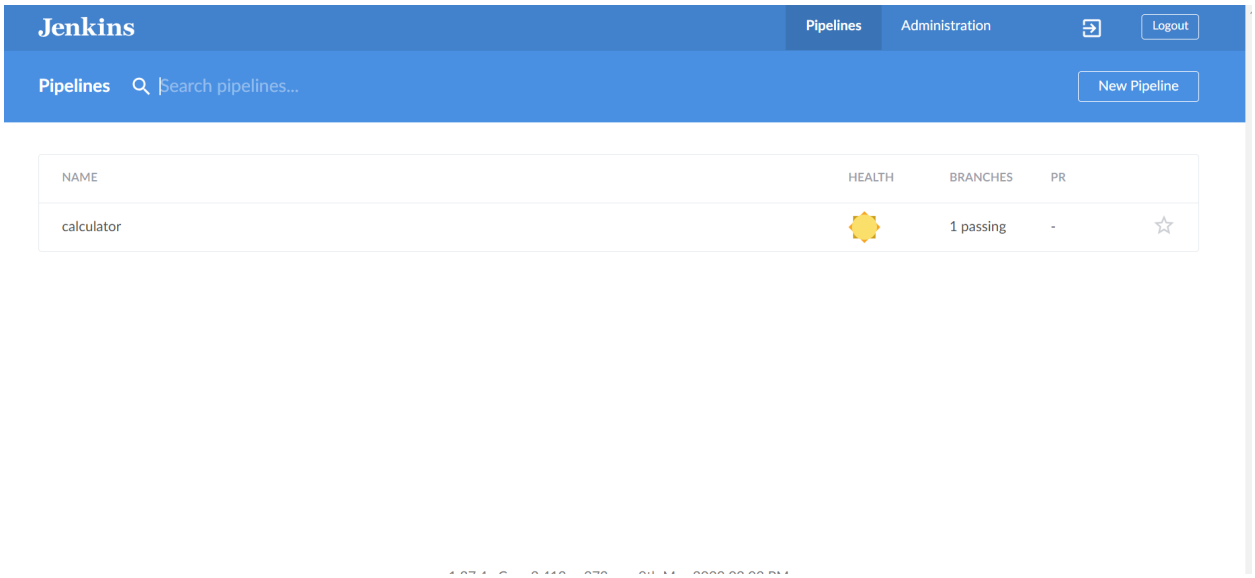
Configure the pipeline definition (e.g. define the stages and steps for each stage).

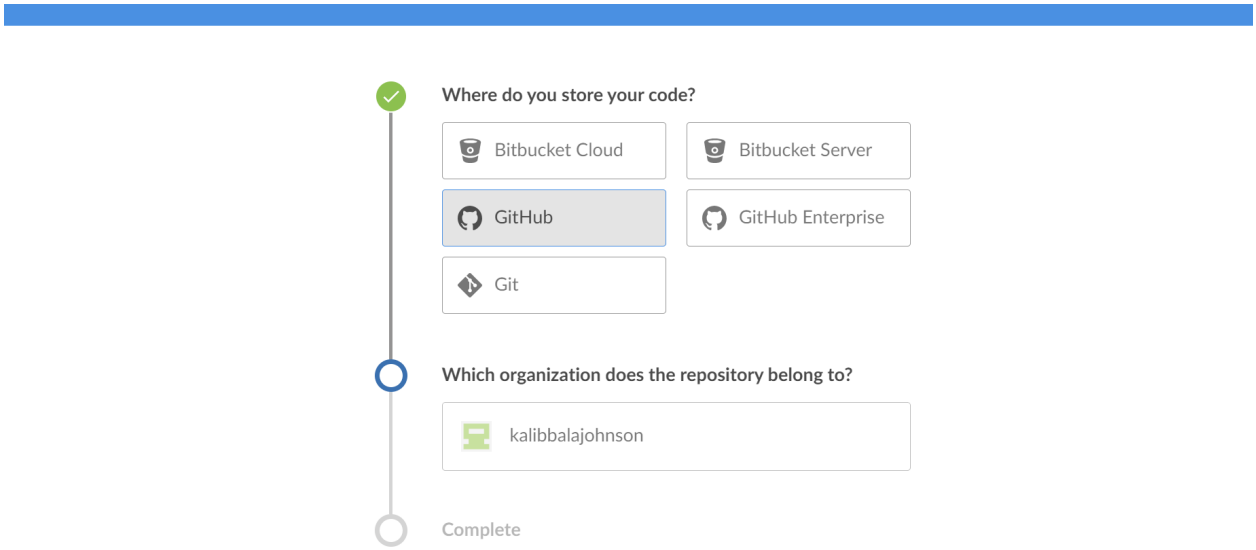
Customize pipeline settings, such as agent configuration, triggers, environment variables, and notifications.

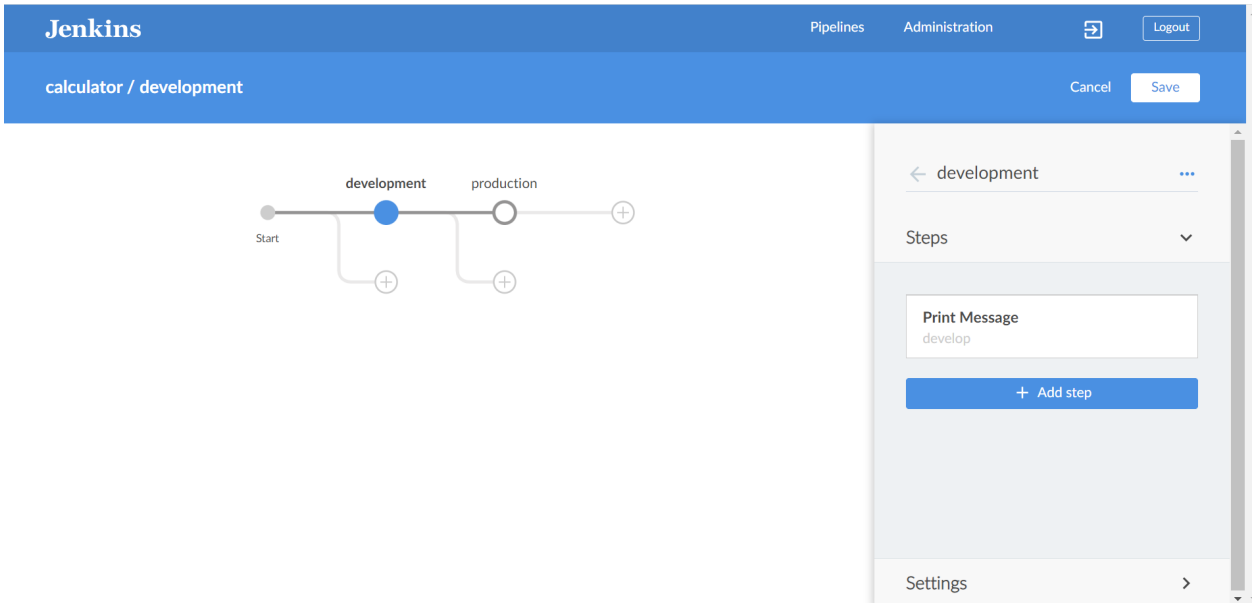
View and run the pipeline in the Blue Ocean interface.

Monitor the pipeline execution, including stages, steps, and any errors or warnings.



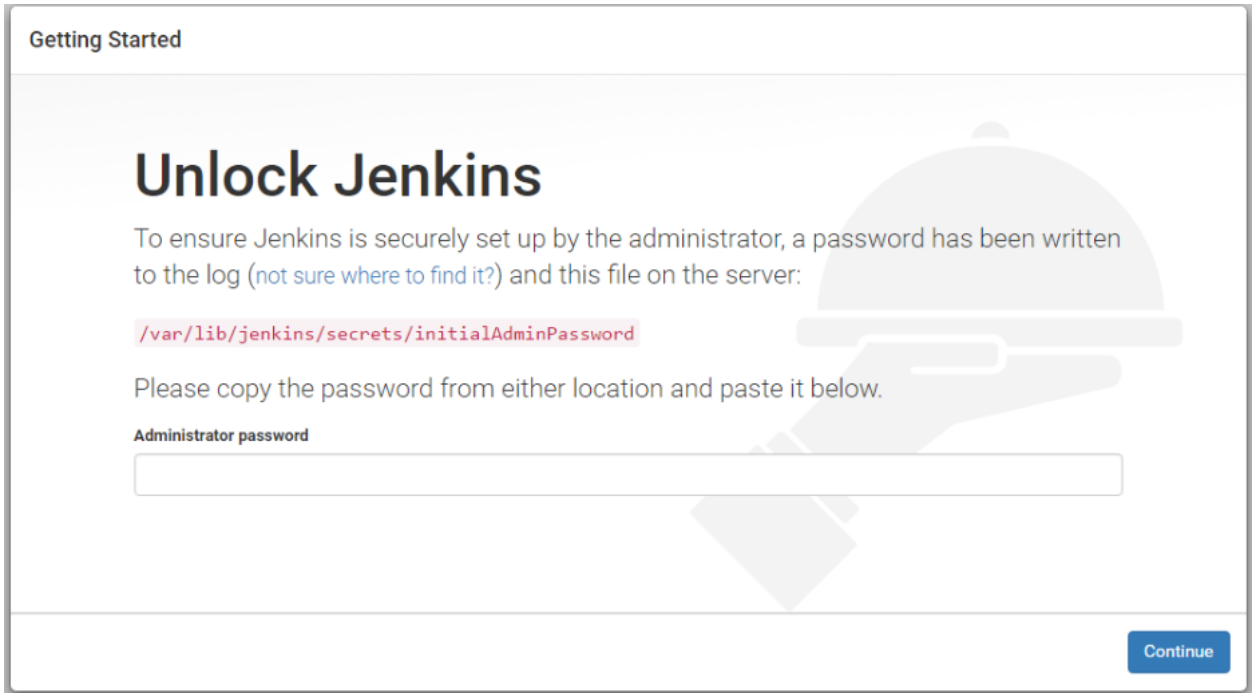


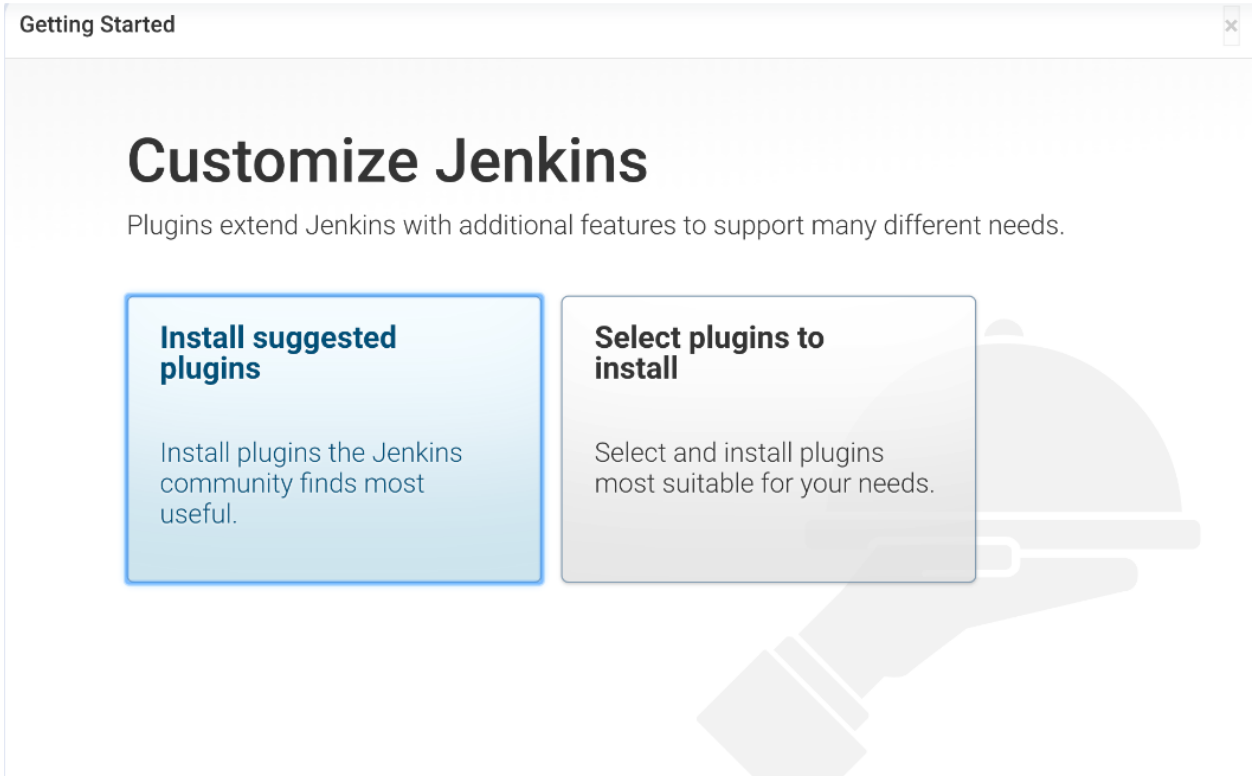

Where do you store your code?
Bitbucket Cloud
Bitbucket Server
GitHub
GitHub Enterprise
Git
Which organization does the repository belong to?
kalibbalajohnson
Complete

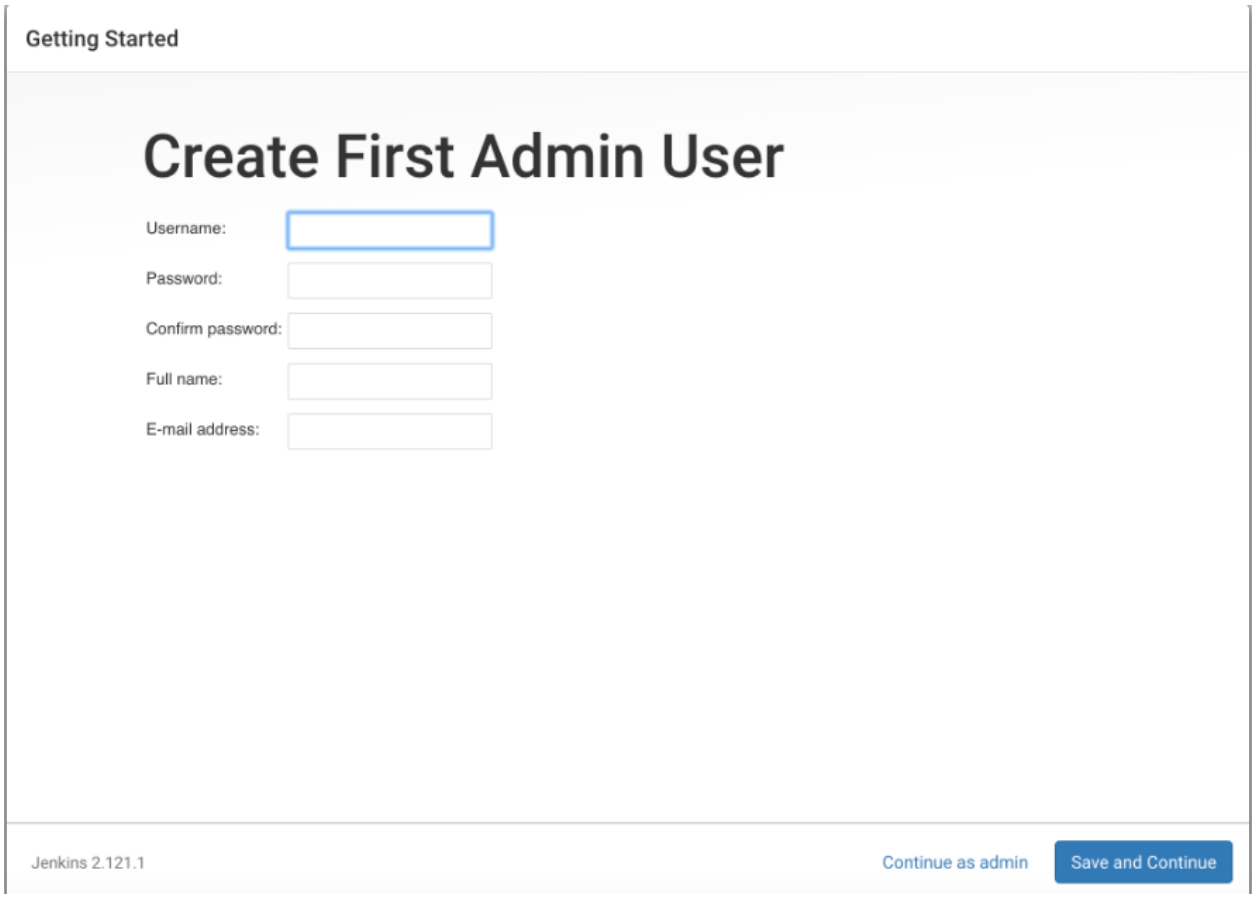

Jenkins
Pipelines
Administration
Logout
calculator / development
Cancel
Save
development
production
Start
development
Steps
Print Message
develop
Add step
Settings

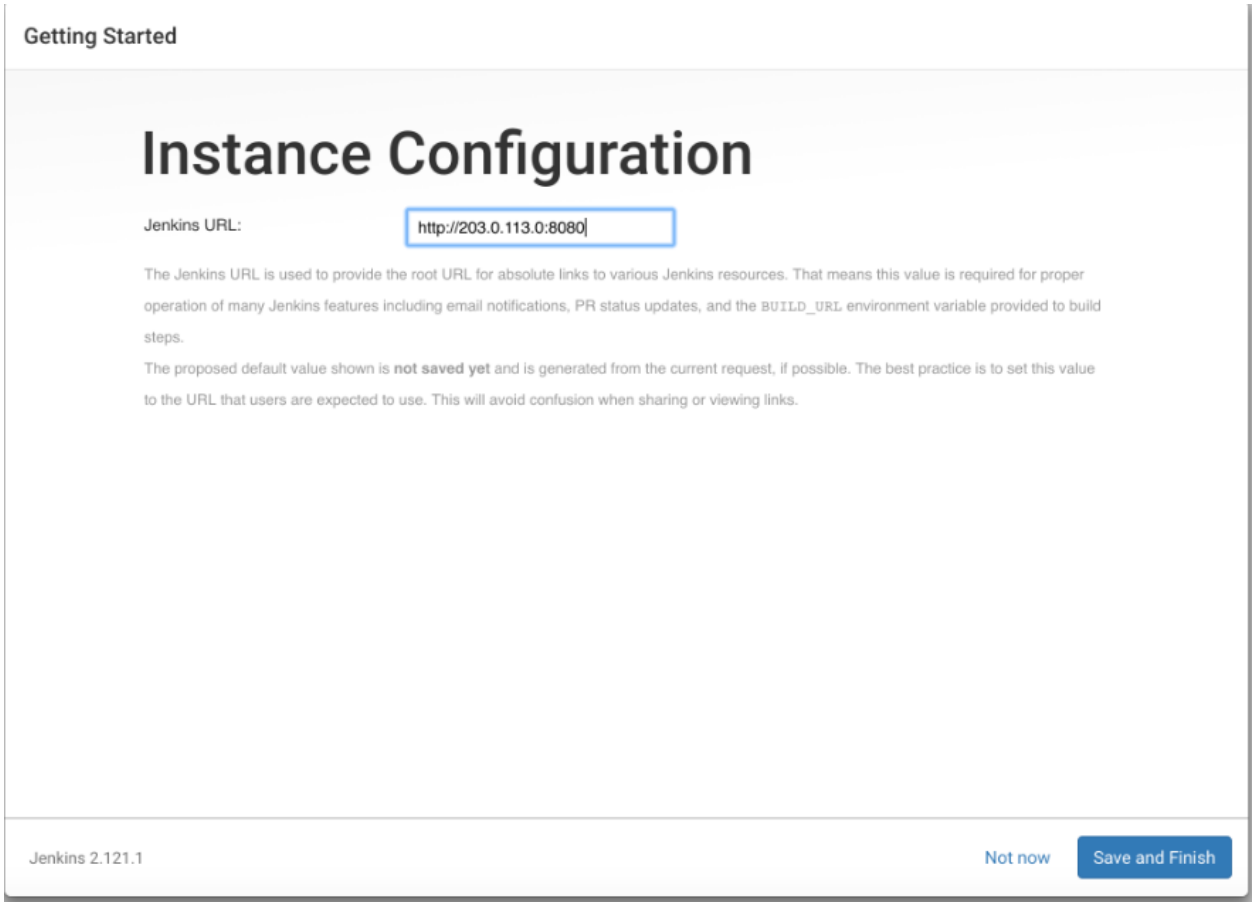
Steps for hosting

- Open command line and enter **ssh root@164.90.161.120** and then enter **password** to log into server.
- **sudo apt update**
- **sudo apt install docker-ce docker-ce-cli containerd.io**
- **sudo apt install git**
- **sudo apt install nodejs**
- Pull the jenkins image from docker hub with **docker pull jenkins/jenkins**
- Run it on port 8080 with **docker run --publish 8080:8080 jenkins/jenkins**
- Open any browser and enter **http://164.90.161.120:8080/**
- Type **docker ps** to see the running containers
- Type **docker logs <containerId>** and locate password




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.


Getting Started
Create First Admin User
Username:
Password:
Confirm password:
Full name:
E-mail address:
Jenkins 2.121.1
Continue as admin
Save and Continue



- Set up jenkins
- Install the necessary packages into the server
- Clone your repo and create and image and container to run a certain port
- Access both jenkins and the calculator through the IP address instead of localhost.

Logger

What is a Logger

This is a tool used to track and record events that occur during the execution of a program. It is a means of capturing important information about the flow and behavior of the program, which can be helpful for debugging, analyzing performance, and understanding user behavior.

When is a logger

- A Logger is used to generate log records, which contain information about specific events or messages that occur during program execution.
- Loggers are typically organized in a hierarchical structure, with parent and child loggers. This hierarchy allows for more granular control over logging behavior and enables the filtering and routing of log records to different destinations.
- Loggers can be configured to log messages at different severity levels, such as DEBUG, INFO, WARNING, ERROR, or CRITICAL. This allows developers to control the verbosity of the logs based on the importance of the information being logged.
- Loggers can be configured to output log records to different destinations, such as the console, files, databases, or external services. This flexibility allows developers to choose the most appropriate destination for their specific use case.
- Loggers can also be configured to format log records in a specific way, including timestamps, log levels, module names, and custom message formats. This makes it easier to read and analyze log files.
- Loggers can be used in multiple programming languages, including Javascript, Python ,Java and also framework.
-
- There are different type of logger which is used in production
- Winston(popularly in use)
- Loggle(used most in the united state)
- Data Dog
- Splunk
-
-
-
- In this file we will be using Winston
-
-
- To install winston
- Npm install Winston

- For more documentation visit https://www.npmjs.com/package/winston
- Note that winston is use for server side rendering only which is not compatible with frameworks like react and vue
- For client side we used loglevel
- Npm install loglevel

Coverage Badge

What is Coverage Badge

- This is a visual indicator that displays the code coverage metrics of a software project. Code coverage measures the percentage of code lines or statements that are executed during automated tests. It provides insights into the effectiveness and thoroughness of the test suite.
- A coverage badge typically displays the code coverage percentage as a badge or icon that can be embedded in a project's documentation, README file, or website. The badge provides a quick and visual indication of the code coverage status, allowing developers and users to easily see how well the project is covered by tests.

Where are they used

- Coverage badges are often used in open-source projects, continuous integration (CI) pipelines, and version control repositories to provide a quick overview of the project's test coverage. The badge typically displays the coverage percentage and may use different colors or styles to indicate the level of coverage.
- The most common format for coverage badges is the Shields.io badge, which can be easily embedded into various platforms, including GitHub README files, project documentation, and websites.

Benefit

- They are way to communicate and promote code coverage in software projects.
- They serve as a visual representation of the project's testing efforts and help in assessing the overall quality and reliability of the codebase.

Website use : https://shields.io/