Basic CI/CD Pipeline with Jenkins

https://www.jenkins.io/

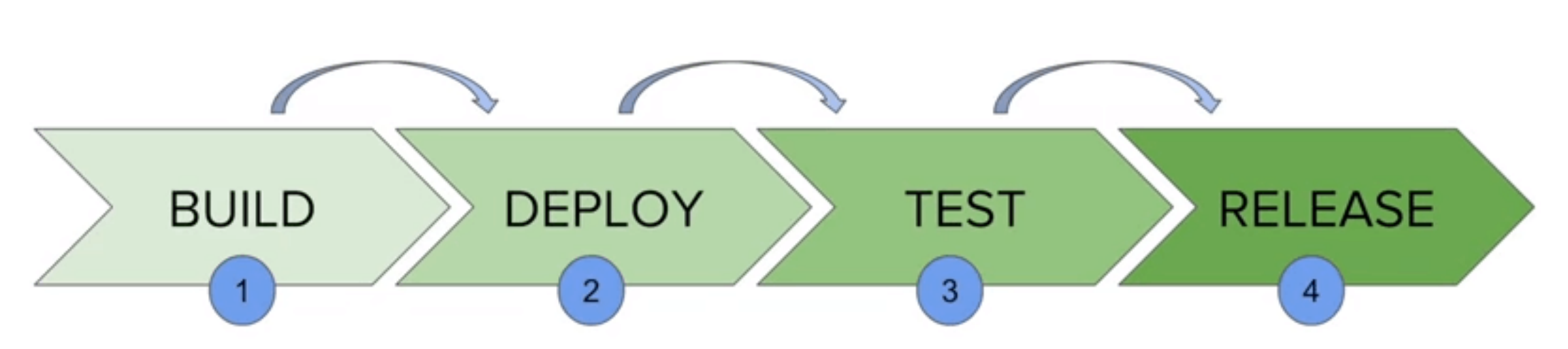
Case Study

- Client have a portfolio of web applications that are somewhat unstable due to frequency of changes to the source code
- Client would like to implement automation to lower application deployment overhead

Solution

- Build out flexible CI/CD pipelines for each web application in portfolio

CI/CD pipeline overview



Jenkins Setup on Linux

```
$ apt install docker -y # install docker

$ docker pull jenkins/jenkins:lts-jdk11 # pull docker jenkins image

$ docker run -p 8080:8080  jenkin/jenkins:lts-jdk11 # run latest jenkins image via docker

$ docker run -p 8080:8080 -p 5000:5000 jenkins/jenkins:lts-jdk11 # launch jenkins via docker
```

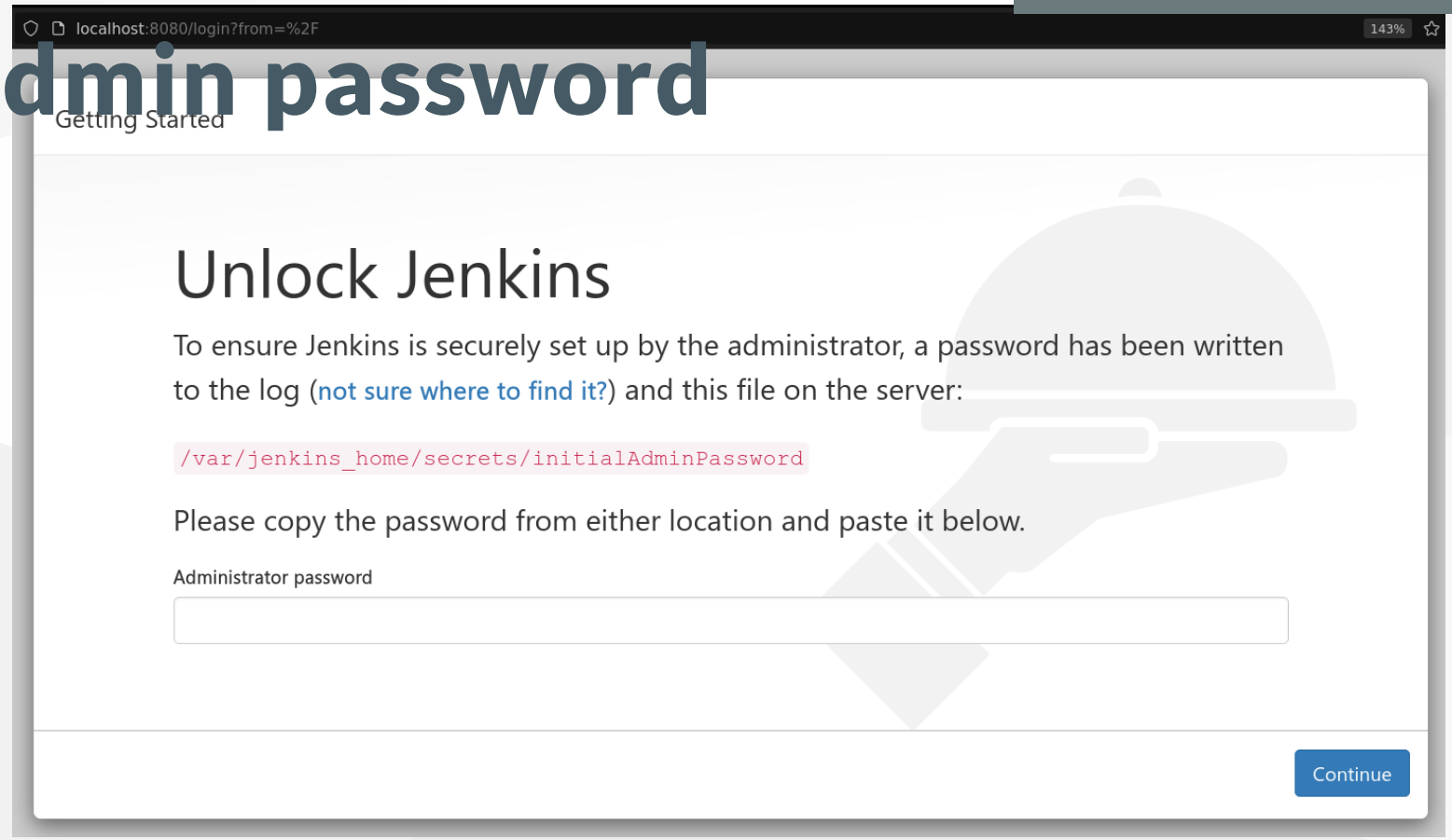
jenkins configurations in a container (via docker)

```
$ docker run -p 8080:8080 -p 5000:5000 jenkins/jenkins:lts-jdk11 # launch jenkins via docker
```

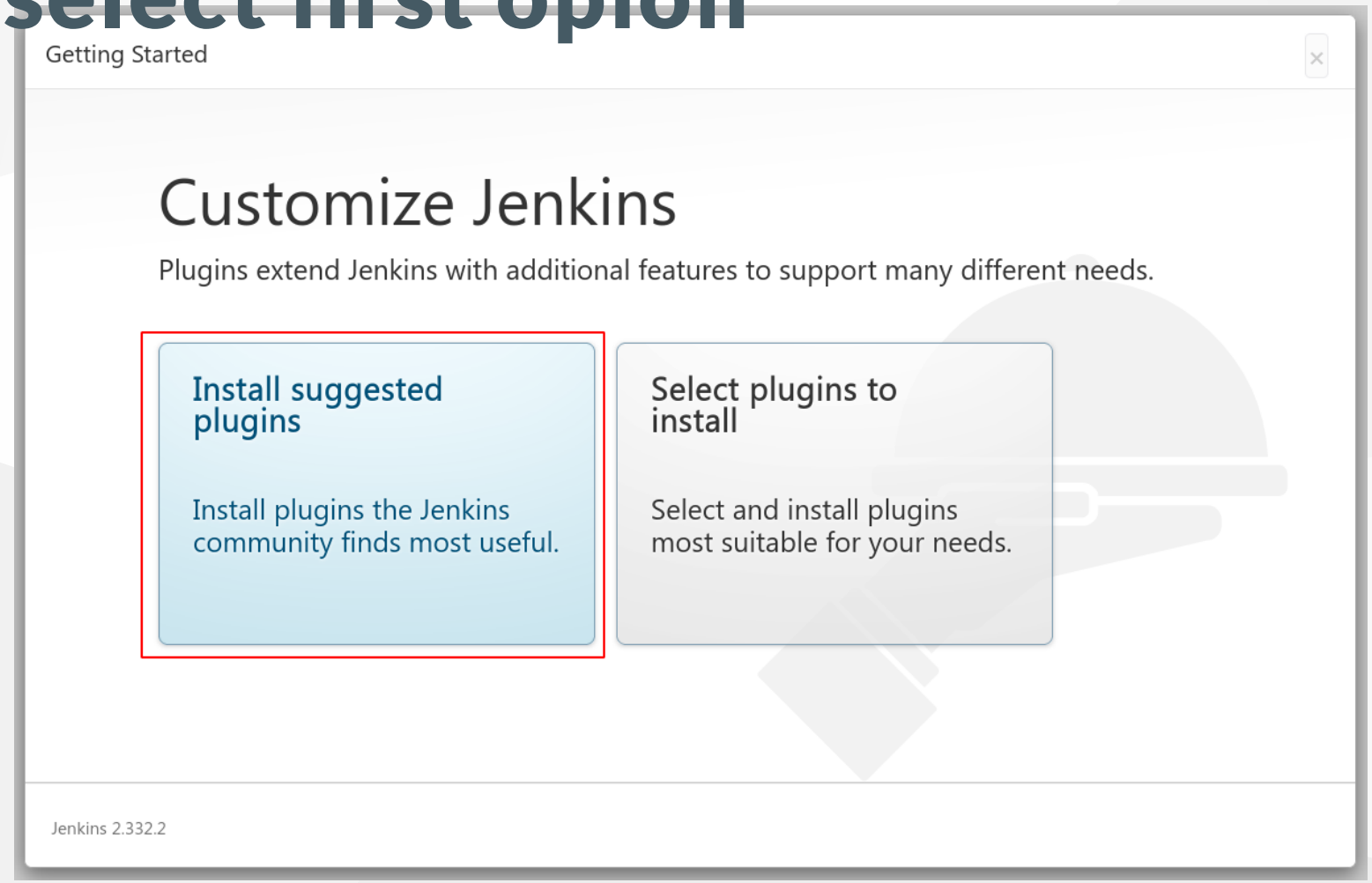
step 1: copy admin from command line

```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

498eacb15add4e8b9469df771a23586a

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

************************************************************
************************************************************
************************************************************
```

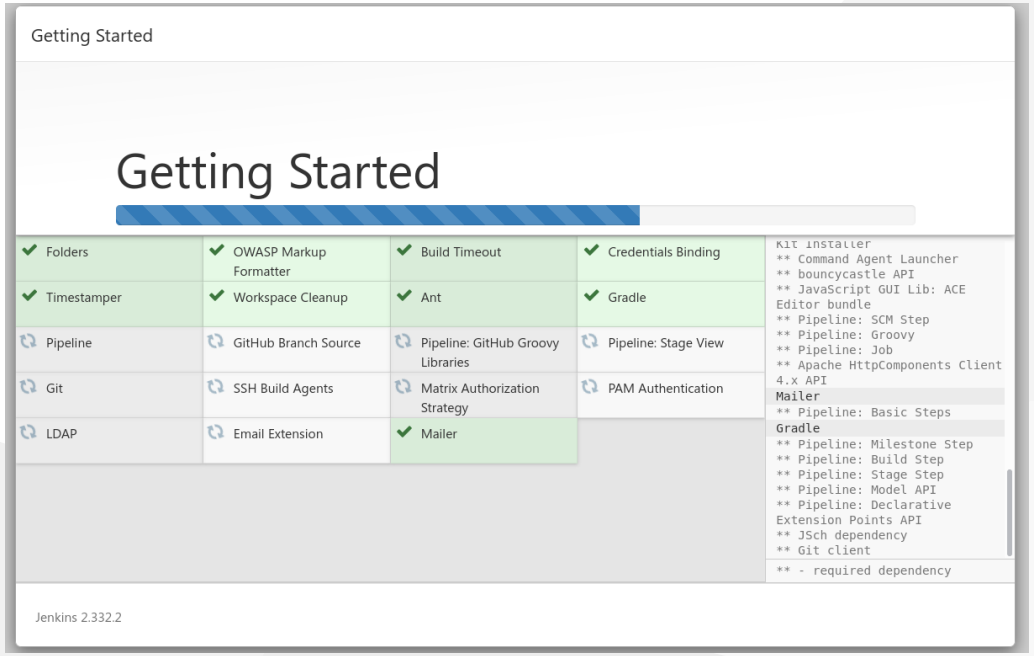
step 2: launch web browser: `localhost:8080` > paste admin password



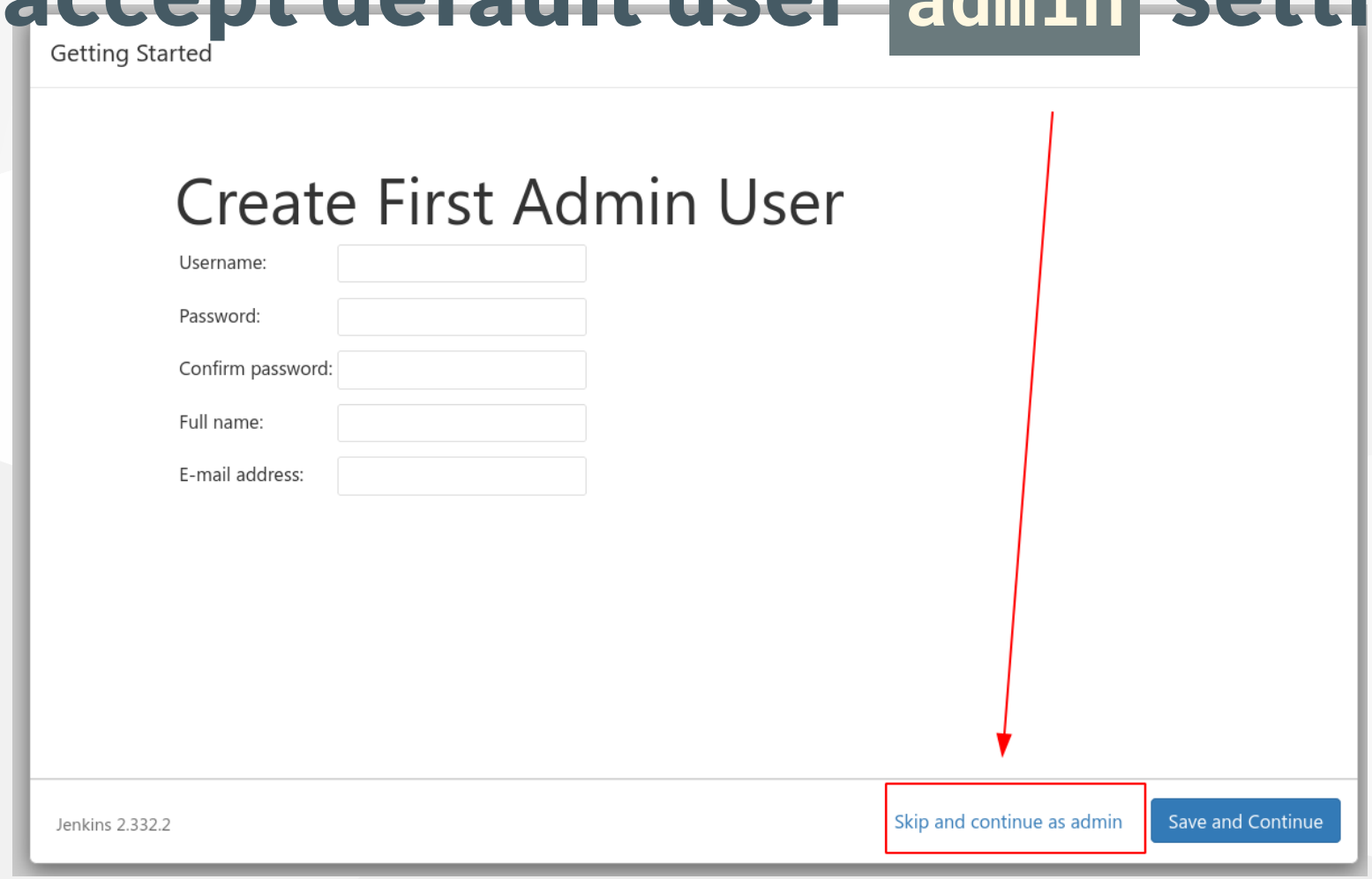
step 3: select first opion



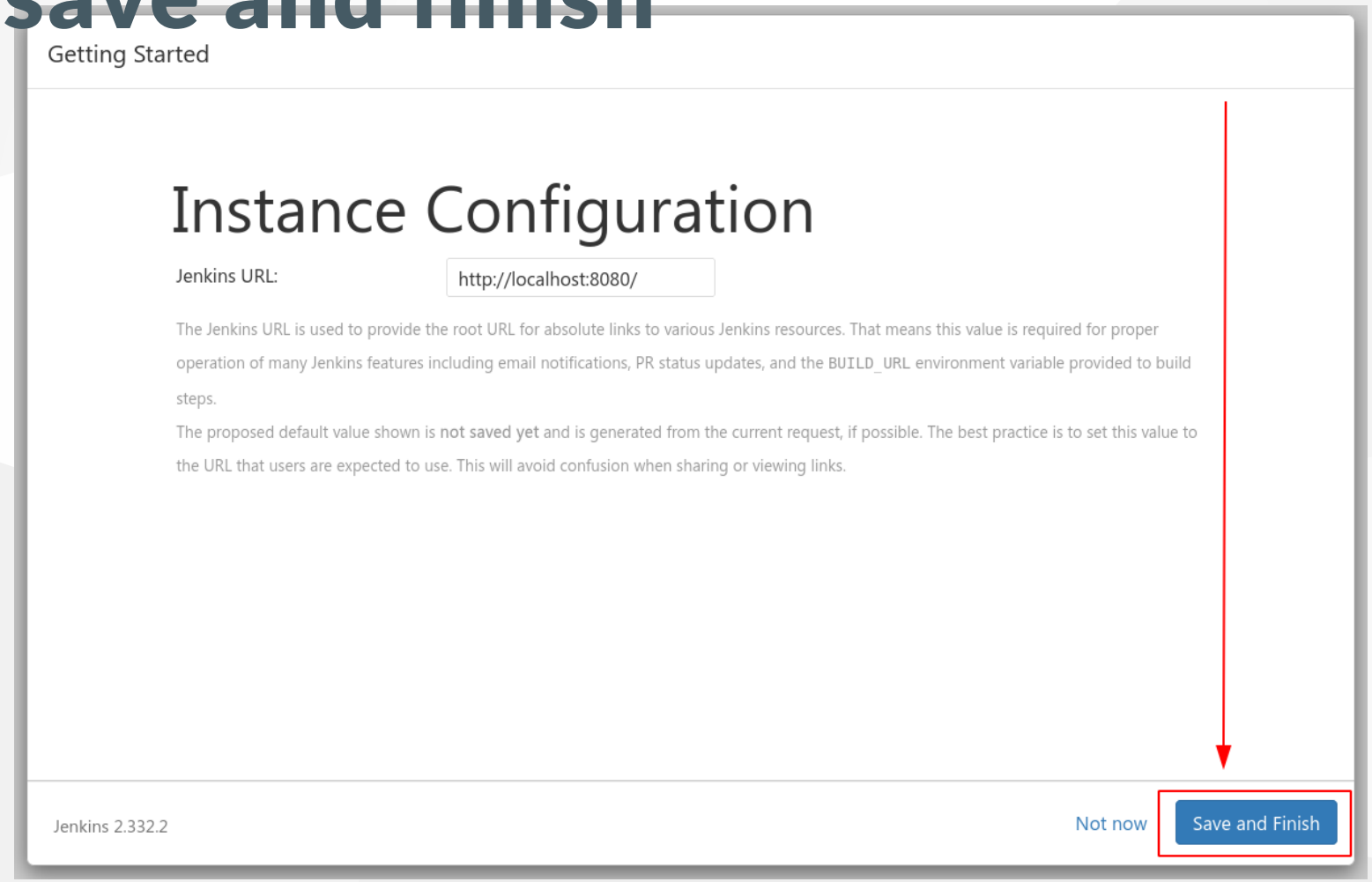
step 4: wait for default configurations to apply



step 5: accept default user `admin` settings



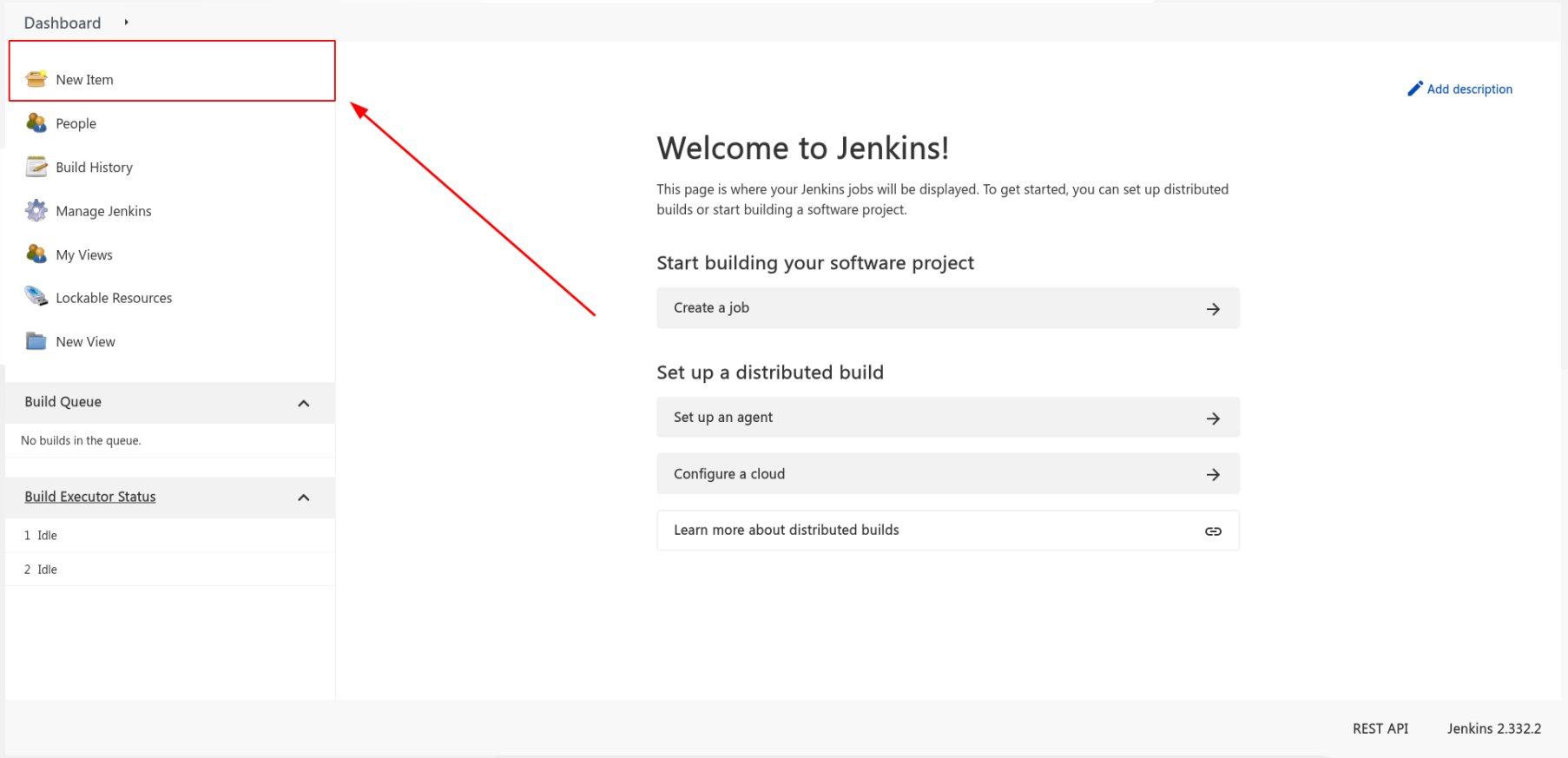
step 6: save and finish



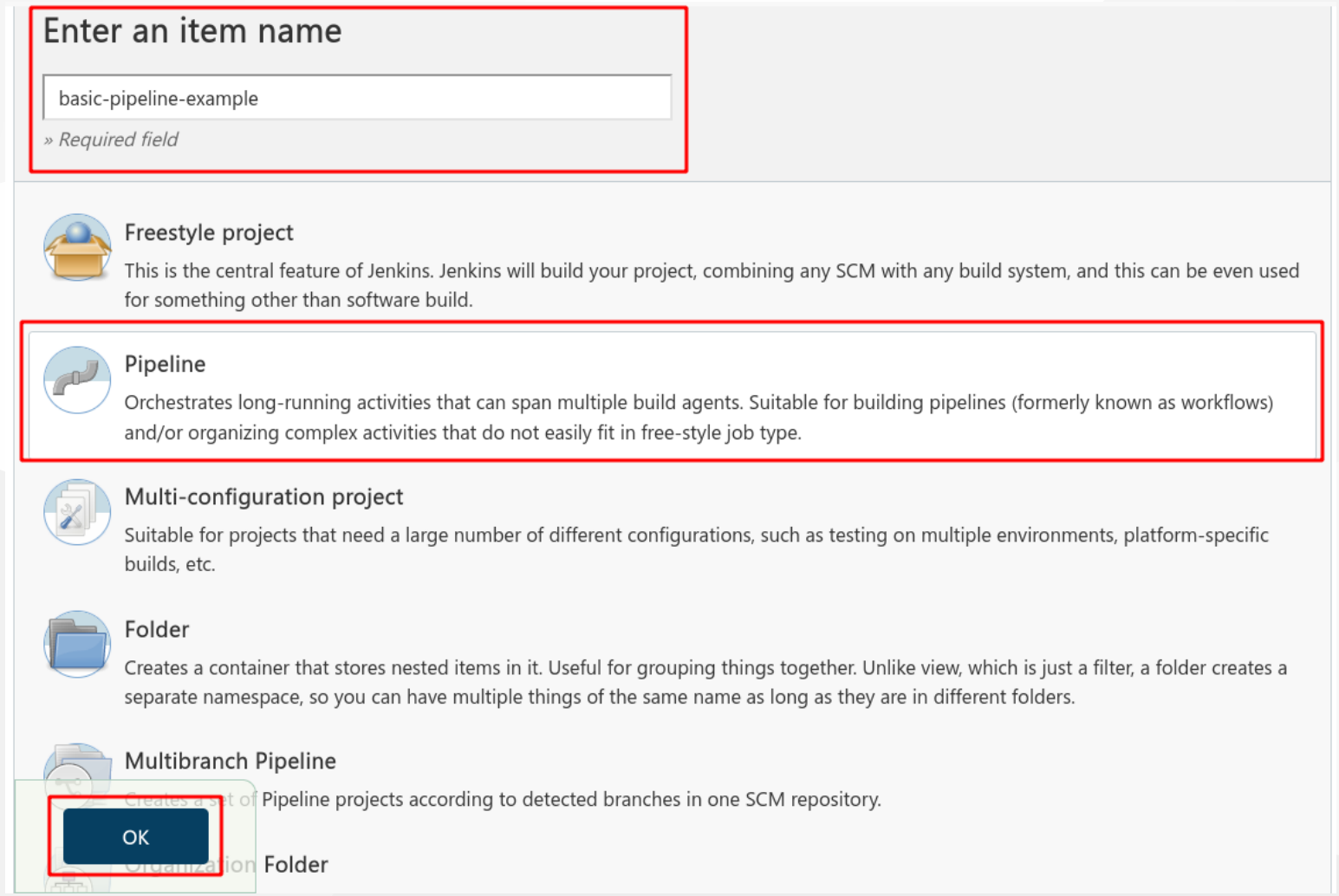
First Pipeline Workflow

- 1. open web browser: `locahost:8080`
- 2. new item > enter item name "my-basic-pipeline" > select pipeline > OK
- 3. `Pipeline` tab > Script text box
- 4. copy and paste contents of `basic pipeline example` from previous slide
- 5. `Build Now`

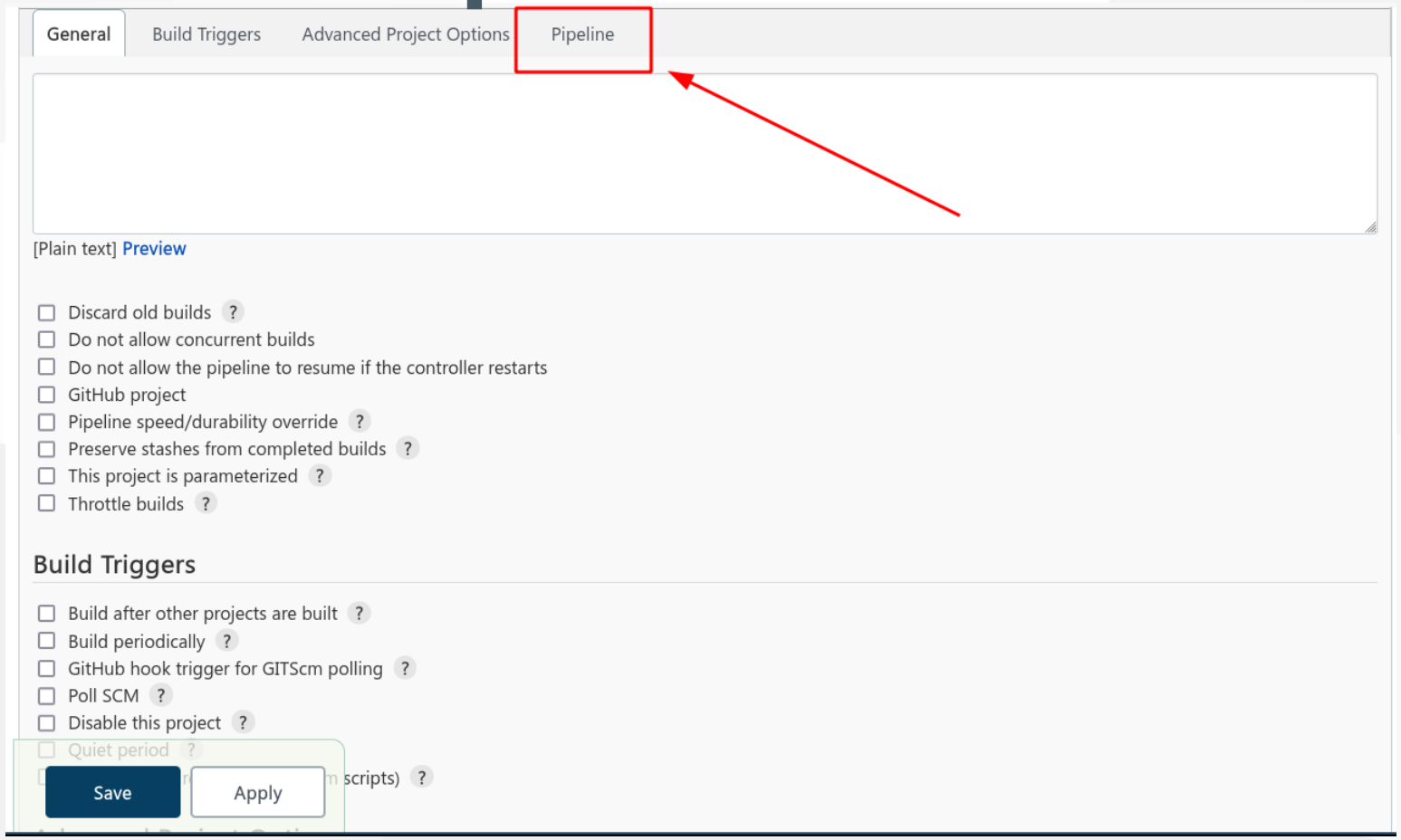
step 1: New Item



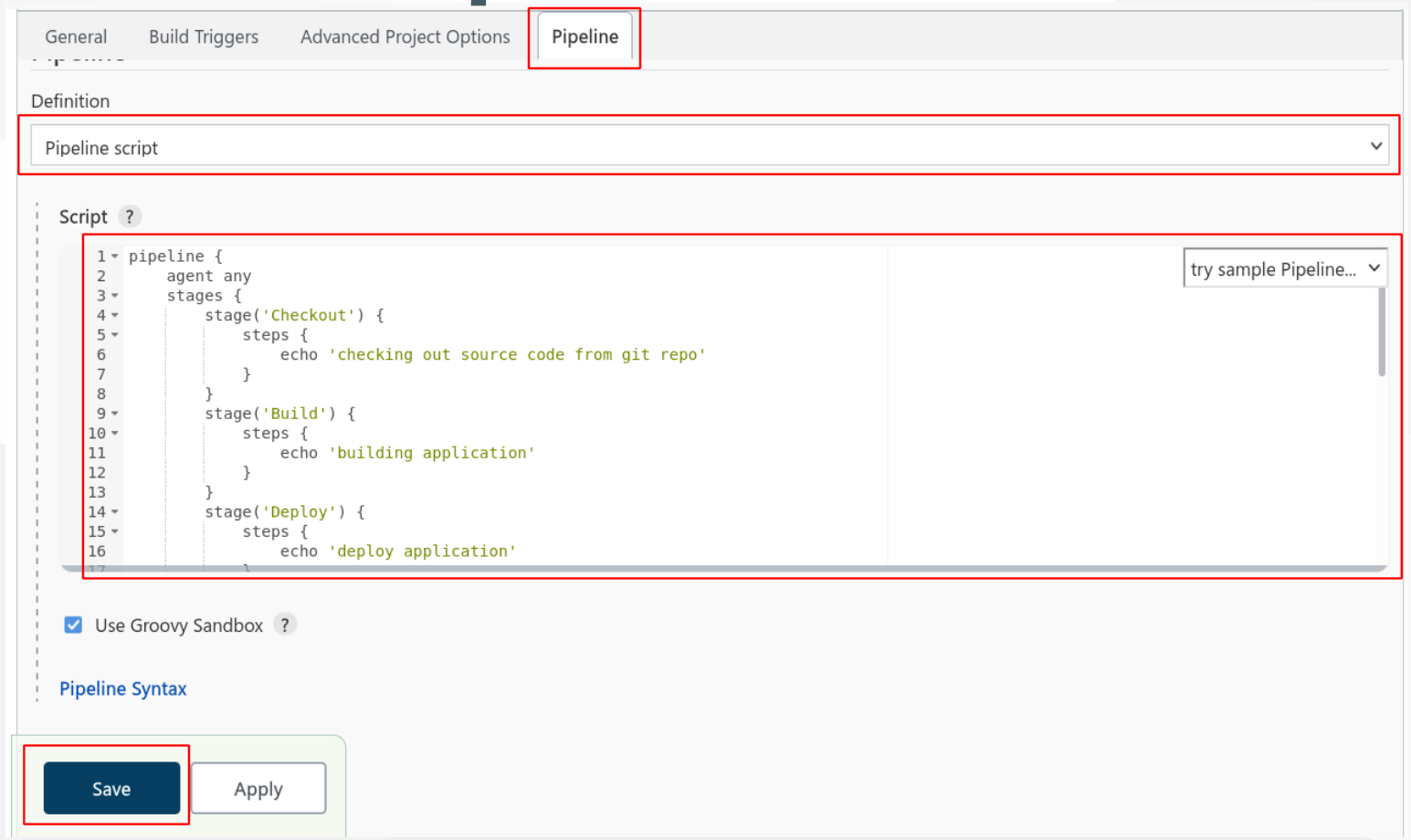
step 2:


Enter an item name
basic-pipeline-example
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
Pipeline projects according to detected branches in one SCM repository.
OK
Folder

step 3: click on Pipeline tab



step 4: click on Pipeline tab

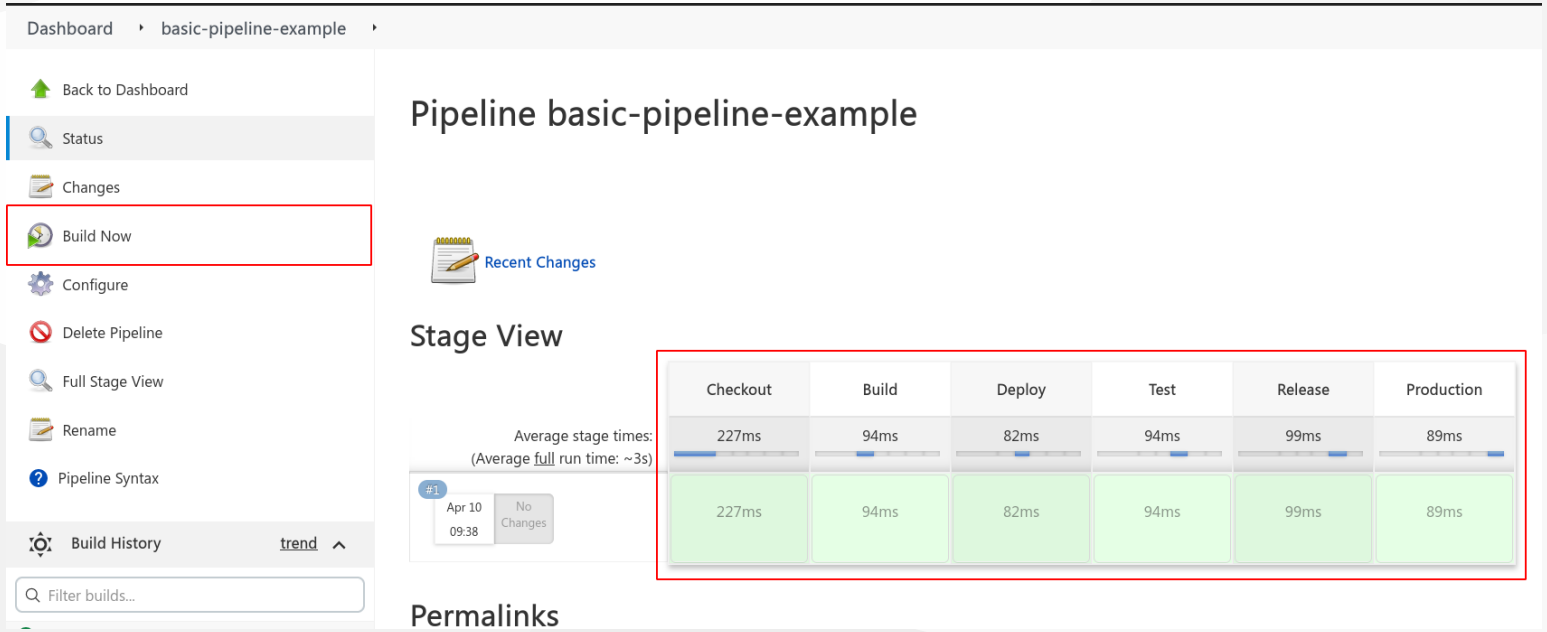


Basic Pipeline Example

“ Jenkinsfile (Declarative Pipeline) ”

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                echo 'building application'
            }
        }
        stage('Deploy') {
            steps {
                echo 'deploy application'
            }
        }
        stage('Test') {
            steps {
                echo 'testing application'
            }
        }
        stage('Release') {
            steps {
```

step 5: Build New



References

- https://www.jenkins.io/pipeline/getting-started-pipelines/
- https://docs.docker.com/engine/install/
- https://git-scm.com/book/en/v2/Getting-Started-Installing-Git