



Basic CI/CD Pipeline with Jenkins

<https://www.jenkins.io/>

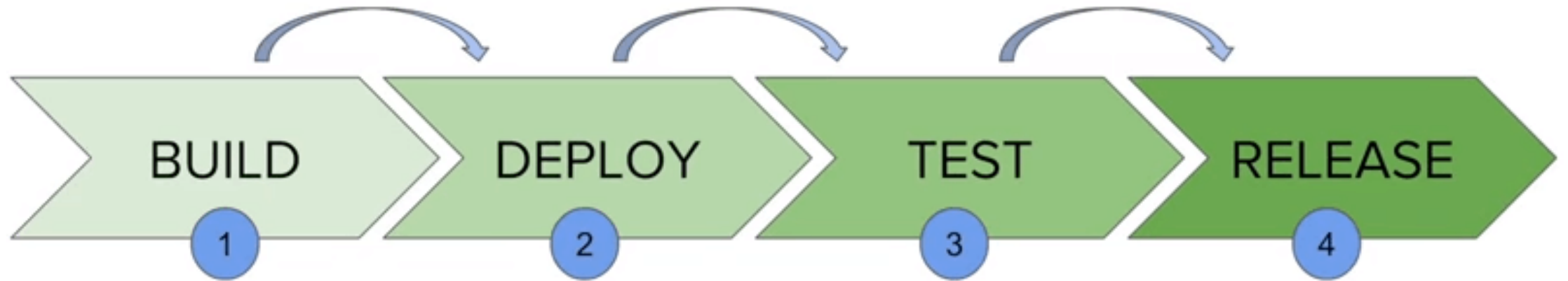
Case Study

- 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 jenkins/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



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

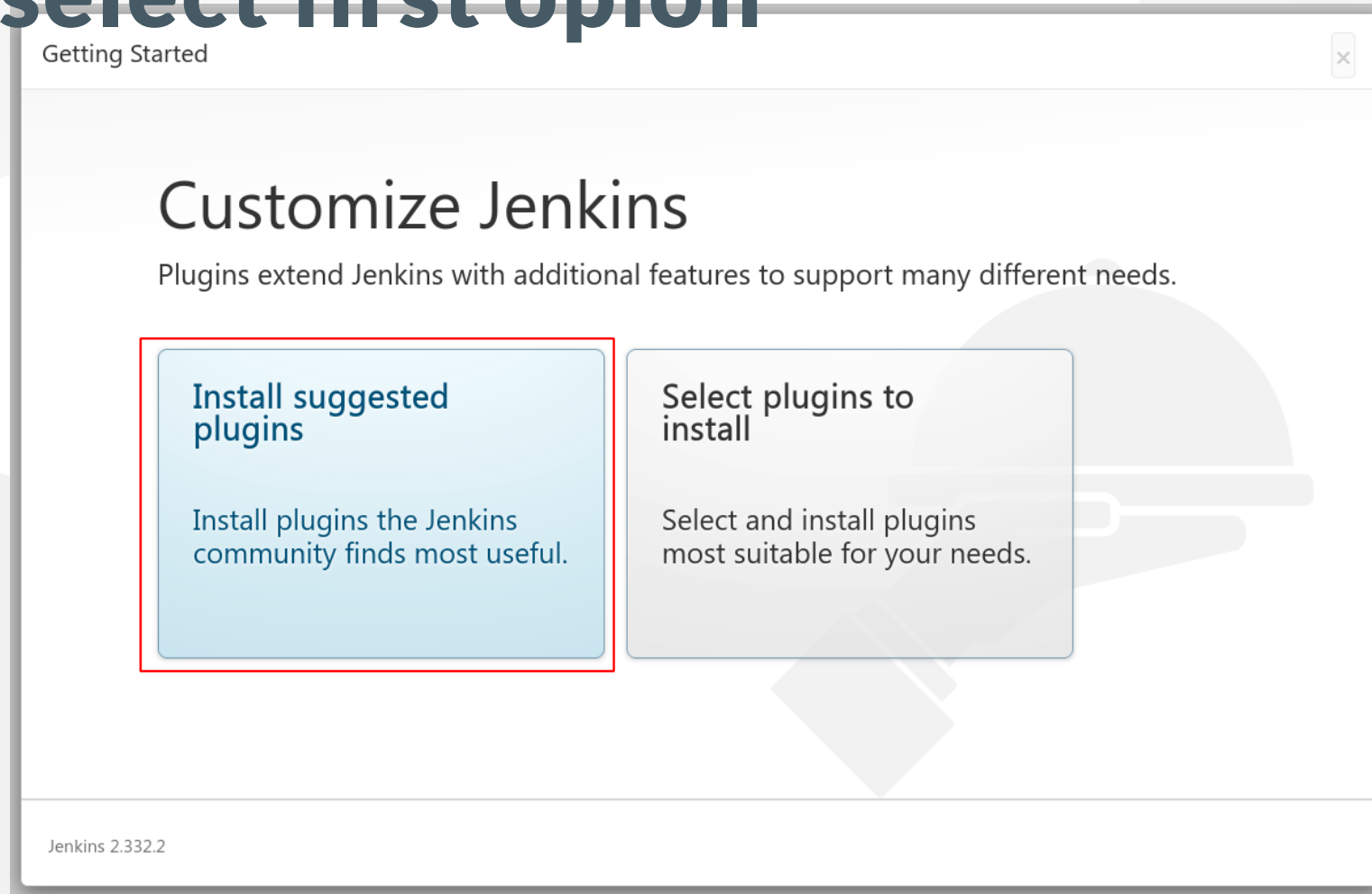
```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

step 3: select first option



step 4: wait for default configurations to apply

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle
⚙ Pipeline	⚙ GitHub Branch Source	⚙ Pipeline: GitHub Groovy Libraries	⚙ Pipeline: Stage View
⚙ Git	⚙ SSH Build Agents	⚙ Matrix Authorization Strategy	⚙ PAM Authentication
⚙ LDAP	⚙ Email Extension	✓ Mailer	

Kit installer

** Command Agent Launcher

** bouncycastle API

** JavaScript GUI Lib: ACE Editor bundle

** Pipeline: SCM Step

** Pipeline: Groovy

** Pipeline: Job

** Apache HttpComponents Client 4.x API

Mailer

** Pipeline: Basic Steps

Gradle

** Pipeline: Milestone Step

** Pipeline: Build Step

** Pipeline: Stage Step

** Pipeline: Model API

** Pipeline: Declarative

Extension Points API

** JSch dependency

** Git client

** - required dependency

Jenkins 2.332.2

step 5: accept default user **admin** settings

Getting Started

Create First Admin User

Username:

Password:


Confirm password:

Full name:

E-mail address:

Jenkins 2.332.2

[Skip and continue as admin](#) [Save and Continue](#)



step 6: save and finish

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

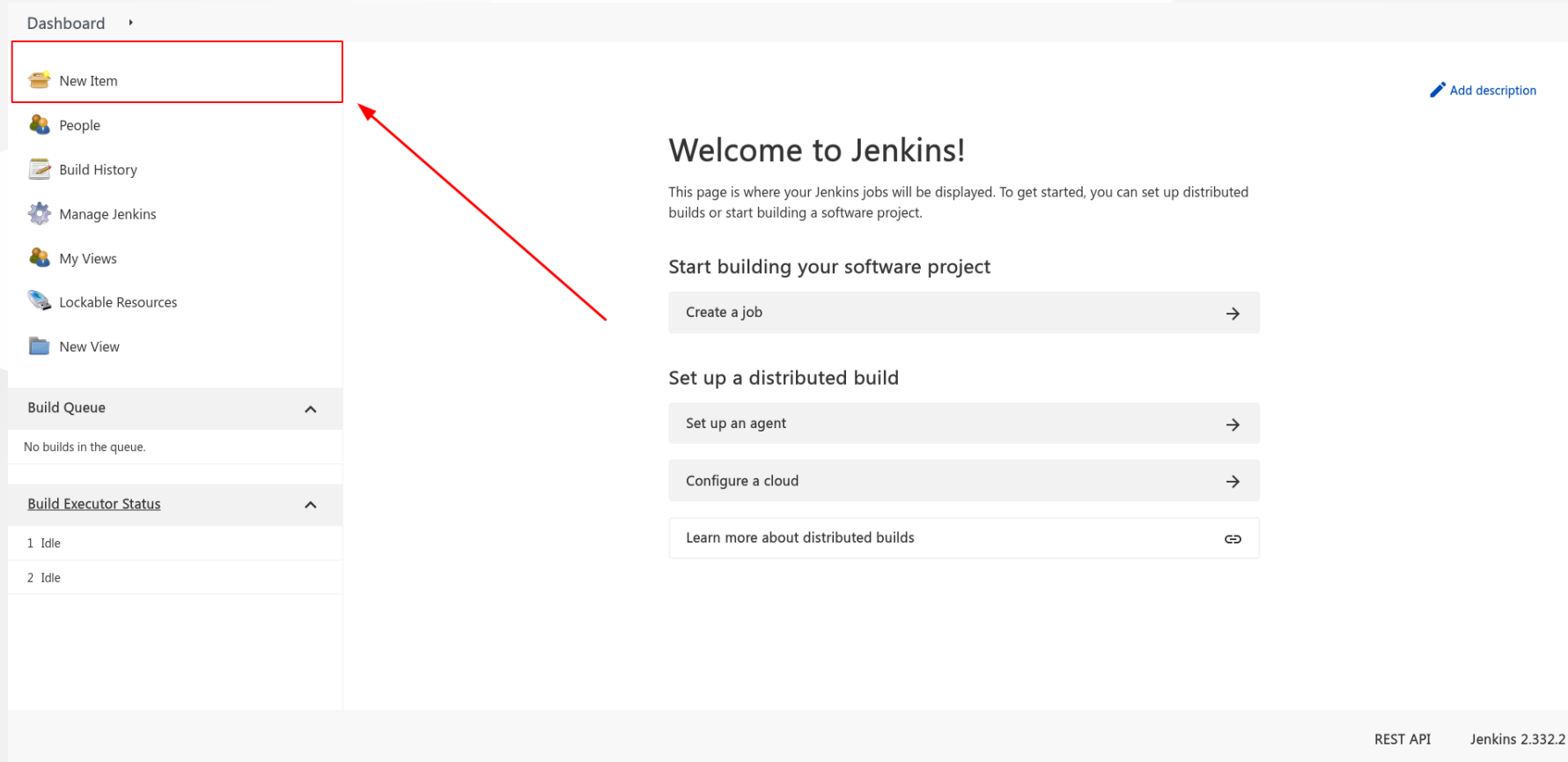
Jenkins 2.332.2

[Not now](#) [Save and Finish](#)

First Pipeline Workflow








- 1. open web browser: `localhost: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



The screenshot shows the Jenkins Dashboard interface. On the left sidebar, the 'New Item' link is highlighted with a red rectangular box. A red arrow points from this box towards the main content area. The main content area displays a 'Welcome to Jenkins!' message, followed by instructions on how to get started. Below this, there are three sections: 'Start building your software project' with a 'Create a job' button, 'Set up a distributed build' with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. The bottom right corner of the dashboard shows links for 'REST API' and 'Jenkins 2.332.2'.

Dashboard ▾

-  New Item
-  People
-  Build History
-  Manage Jenkins
-  My Views
-  Lockable Resources
-  New View

Build Queue ^

No builds in the queue.

Build Executor Status ^

1 Idle

2 Idle

[Add description](#)

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

REST API Jenkins 2.332.2

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

Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Organization Folder

step 3: click on Pipeline tab

General Build Triggers Advanced Project Options **Pipeline**

[Plain text] [Preview](#)

- ☐ Discard old builds ?
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the controller restarts
- ☐ GitHub project
- ☐ Pipeline speed/durability override ?
- ☐ Preserve stashes from completed builds ?
- ☐ This project is parameterized ?
- ☐ Throttle builds ?

Build Triggers

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Disable this project ?
- ☐ Quiet period ?

?

step 4: click on Pipeline tab

General Build Triggers Advanced Project Options **Pipeline**

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3   stages {
4     stage('Checkout') {
5       steps {
6         echo 'checking out source code from git repo'
7       }
8     }
9     stage('Build') {
10      steps {
11        echo 'building application'
12      }
13    }
14    stage('Deploy') {
15      steps {
16        echo 'deploy application'
17      }
18    }
19  }
20 }
```

try sample Pipeline...

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save Apply

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

Dashboardbasic-pipeline-example

Back to Dashboard

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build Historytrend

Filter builds...

Pipeline basic-pipeline-example

Recent Changes

Stage View

Average stage times:
(Average full run time: ~3s)

#1
Apr 10
09:38
No
Changes

Checkout	Build	Deploy	Test	Release	Production
227ms	94ms	82ms	94ms	99ms	89ms

Permalinks

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>