



# **Certified Jenkins Engineer (CJE) – 2017**

Certification Exam Study Guide

## CloudBees<sup>®</sup>, Inc., offers two certifications for Jenkins engineers

- The Certified Jenkins Engineer (CJE) exam consists of 60 multiple-choice questions testing knowledge of open-source Jenkins.
- The Certified CloudBees Jenkins Platform Engineer (CCJPE) exam consists of 90 questions: 60 questions testing knowledge of open-source Jenkins and 30 questions testing knowledge of the CloudBees Jenkins Platform.

This Study Guide is for the Certified Jenkins Engineer (CJE) exam. To view the Study Guide for the Certified CloudBees Jenkins Platform Engineer (CCJPE) exam, [click here](#).

## Introduction

This document is intended to help you prepare for the Certified Jenkins Engineer (CJE) exam.

The exam consists of 60 multiple-choice questions, divided into 4 sections, which will test your skills as a [Jenkins Engineer](#).

In this guide, you will find a list of the topics tested on the exam, links to external references, and sample questions.

## What's new?

**Note: All changes from 2016 to 2017 are indicated with a [new] sign in this document**

Main differences between 2016 and 2017 certification exams:

- Questions about open-source Jenkins are now based on Jenkins 2.19.4
- Pipeline related questions upgraded to [the latest syntax](#) coming with [the version 2.4 of Pipeline plugin](#)
- Open-source section includes questions on Multibranch and Pipeline Global Libraries
- Plugins covered in the exam now include only those in the "suggested" set (see below for details)
- Questions about CJP are now based on CJP 2.7.20.2
- CJP questions now include a section about CloudBees Assurance Program

- Questions about Docker and Configuration Management have been removed

## Structure

This exam is comprised of 4 sections:

1. Key CI/CD/Jenkins concepts
2. Jenkins usage
3. Building Continuous Delivery (CD) Pipelines
4. CD-as-code best practices

All questions are based on version [2.19.4](#) [new] of the Jenkins core.

All questions are based on an out-of-the-box standard installation of Jenkins ("base" Jenkins), with the default recommended plugin set installed ("Suggested plugins"). See section "Plugins" for more information.

**NOTE:** On the exam, questions are presented in random order, not in sections.

## Plugins

Questions in sections 1–4 primarily cover questions about a "base" Jenkins installation, but knowledge of the "suggested" plugins will also be covered. Candidates are expected to know the functionality/uses of these plugins but will not be tested on detailed usage.

[new] The "suggested" plugins are the default plugins installed by the "Setup Wizard" on a fresh new Jenkins installation. You can find the exhaustive list, bound to a fixed Jenkins version, by following this link: [Jenkins 2.19.4 suggested plugin list](#).

[new] Please note that the "[Pipeline Plugin](#)" is itself an aggregation of plugins implementing the Pipeline and related features. It includes the following capabilities:

- [Pipeline Multibranch](#)
- [Pipeline Shared Groovy Libraries](#)
- [Pipeline Stage View](#)

## Terminology

Please also note the following:

- **SCM** stands for “source code management” unless otherwise specified.
- **Pipeline** refers to the job type created by the Pipeline plugin (formerly known as the “Workflow plugin”), except where used generically (e.g., “CD pipelines”) or in the names of specific plugins (e.g., “Build Pipeline plugin”).
- Various UI elements in Jenkins will be referred to using the following terms:

The image shows a screenshot of the Jenkins web interface, specifically the 'Manage Jenkins' page. A large bracket labeled '1' spans the top of the page. On the left side, a list of navigation links is shown, with a bracket labeled '2' encompassing the entire list. Below this, a list of configuration pages is shown, with a bracket labeled '2.a' encompassing 'Configure System' and 'Configure Global Security', a bracket labeled '2.b' encompassing 'Configure Analytics' and 'Reload Configuration from Disk', a bracket labeled '2.c' encompassing 'Manage Plugins', 'System Information', 'System Log', 'Load Statistics', 'Jenkins CLI', 'Script Console', 'Manage Nodes', and 'High Availability Status', and a bracket labeled '2.d' encompassing 'Build Queue' and 'Build Executor Status'. The 'Build Queue' section shows 'No builds in the queue.' and the 'Build Executor Status' section shows two executors, '1 part of App' and '2 part of App', both with red progress bars and status icons. The 'Wasted Minutes' section shows '0 ms were wasted because you didn't have enough executors.'

1

2

2.a

2.b

2.c

2.d

1. Manage Jenkins (admin console)

2. Configuration pages:

2.a Configure System (global configuration)

2.b Configure Global Security

2.c Manage Plugins (Jenkins update center)

2.d Manage Nodes (node/slave configuration)

The screenshot shows the Jenkins Dashboard with the following numbered callouts:

1. Dashboard
2. Breadcrumbs
3. Sidebar
4. Build Queue
5. Build Executor Status
6. "All" view
7. Jobs/projects
8. Build
9. Build status
10. Build trend

The dashboard includes a sidebar with navigation links, a main table of builds, and sections for Build Queue, Build Executor Status, and Wasted Minutes.

S	W	Name	Last Success	Last Failure	Last Duration
		App	6 mo 15 days - #32	N/A	44 sec
		Base Image Build	5 mo 13 days - #31 kohsuke/acmecorp-base	5 days 1 hr - #32 kohsuke/acmecorp-base	23 sec
		Just another build	5 mo 13 days - #11	6 mo 21 days - #4	14 sec
		With custom build environment	5 mo 13 days - #7	6 mo 21 days - #1	6.7 sec

The screenshot shows the Jenkins Job Configuration page for 'Base Image Build' with the following numbered callouts:

1. Job/project configuration page
2. Build History:
  - 2.a Stable build
  - 2.b Failed build

The configuration page includes fields for Project name, Description, and various options like Discard Old Builds, Show Consolidated Build View, Docker Container, and GitHub project. The Build History section shows a list of builds with their status and timestamps.

Build Number	Status	Timestamp
#32 kohsuke/acmecorp-base	Failed	Dec 30, 2015 1:12 PM
#31 kohsuke/acmecorp-base	Stable	Jul 24, 2015 8:50 PM
#30 kohsuke/acmecorp-base	Failed	Jul 16, 2015 7:23 AM

## 1. Key CI/CD/Jenkins Concepts

This topic comprises approximately 18% of the exam. Questions cover the following topics:

- Continuous Delivery/Continuous Integration Concepts
  - Define continuous integration, continuous delivery, continuous deployment
  - Difference between CI and CD
  - Stages of CI and CD
  - Continuous delivery versus continuous deployment
- Jobs
  - What are jobs in Jenkins?
  - Types of jobs
  - Scope of jobs
- Builds
  - What are builds in Jenkins?
  - What are build steps, triggers, artifacts, and repositories?
  - Build tools configuration
- Source Code Management
  - What are source code management systems and how are they used?
  - Cloud-based SCMs
  - Jenkins changelogs
  - Incremental updates v clean check out
  - Checking in code
  - Infrastructure-as-Code
  - Branch and Merge Strategies
- Testing
  - Benefits of testing with Jenkins
  - Define unit test, smoke test, acceptance test, automated verification/functional tests
- Notifications
  - Types of notifications in Jenkins
  - Importance of notifications
- Distributed Builds
  - What are distributed builds?
  - Functions of masters and agents
- Plugins
  - What are plugins?
  - What is the plugin manager?
- Jenkins Rest API

- How to interact with it
  - Why use it?
- Security
  - Authentication versus authorization
  - Matrix security
  - Definition of auditing, credentials, and other key security concepts
- Fingerprints
  - What are fingerprints?
  - How do fingerprints work?
- Artifacts
  - How to use artifacts in Jenkins
  - Storing artifacts
- Using 3rd party tools
  - How to use 3rd party tools
- Installation Wizard **[new]**
  - What is the Jenkins Installation Wizard?
  - How to use the Wizard?
  - Which configurations are covered by the Installation Wizard?

These online resources provide entry points to understanding the above topics:

- <http://www.martinfowler.com>
  - [Continuous Integration](#)
  - [Continuous Delivery](#)
  - [Deployment Pipeline](#)
- <http://www.informit.com>
  - [CD Pipeline Anatomy](#)
- <http://devops.com>
  - [What is a CD pipeline](#)
- <https://jaxenter.com>
  - [Implementing Continuous Delivery](#)
- <http://www.infoq.com>
  - [Orchestrating Pipelines Jenkins](#)
- <http://technologyconversations.com>
  - [Continuous Delivery Introduction to Concepts and Tools](#)
- <https://en.wikipedia.org>

- [Continuous delivery](#)
- [Artifact software development](#)
- [Build automation](#)
- [Distributed version control](#)
- [List of version control software](#)
- [Smoke testing \(software\)](#)
- <https://jenkins.io> [new]
  - [Jenkins Installation and Setup](#) [new]
  - [Jenkins Documentation](#) [new]
  - [Jenkins Pipeline](#) [new]
  - [Jenkins HandBook](#) [new]
  - <https://plugins.jenkins.io> [new]
- <https://www.safaribooksonline.com>
  - [Jenkins the Definitive Guide](#)
- <https://wiki.jenkins-ci.org>
  - [Administering Jenkins](#)
  - [Terminology](#)
  - [Extreme feedback lamp switch gear style](#)
  - [Distributed builds: Offline status and retention strategy](#)
  - [Remoting issue](#)
  - [Remote access API](#)
  - [Matrix based security](#)
  - [Securing Jenkins](#)
  - [Quick and Simple Security](#)
- <http://docs.openstack.org>
  - [Jenkins job builder](#)
- <https://www.simple-talk.com>
  - [Branching and merging](#)
- <http://stackoverflow.com>
  - [What is unit test, integration test, smoke test, regression test?](#)
- <https://www.cloudbees.com/>
  - [Notifications](#)
- <http://searchsecurity.techtarget.com/>



- [Authentication authorization and accounting](#)

## 2. Jenkins usage (features and functionality)

This topic comprises approximately 23% of the exam. Questions cover the following topics:

- Jobs
  - Organizing jobs in Jenkins
  - Parameterized jobs
  - Usage of Freestyle/Pipeline/Matrix jobs
- Builds
  - Setting up build steps and triggers
  - Configuring build tools
  - Running scripts as part of build steps
- Source Code Management
  - Polling source code management
  - Creating hooks
  - Including version control tags and version information
- Testing
  - Testing for code coverage
  - Test reports in Jenkins
  - Displaying test results
  - Integrating with test automation tools
  - Breaking builds
- Notifications
  - Setup and usage
  - Email notifications, instant messaging
  - Alarming on notifications
- Distributed Builds
  - Setting up and running builds in parallel
  - Setting up and using SSH agents, JNLP agents, cloud agents
  - Monitoring nodes
- Plugins
  - Setting up and using Plugin Manager
  - Finding and configuring required plugins
- CI/CD
  - Using Pipeline (formerly known as “Workflow”)
  - Integrating automated deployment

- Release management process
  - Pipeline stage behavior
- Jenkins Rest API
  - Using REST API to trigger jobs remotely, access job status, create/delete jobs
- Security
  - Setting up and using security realms
  - User database, project security, Matrix security
  - Setting up and using auditing
  - Setting up and using credentials
- Fingerprints
  - Fingerprinting jobs shared or copied between jobs
- Artifacts
  - Copying artifacts
  - Using artifacts in Jenkins
  - Artifact retention policy
- Alerts
  - Making basic updates to jobs and build scripts
  - Troubleshooting specific problems from build and test failure alerts

These online resources provide entry points to understanding the above topics:

- <https://wiki.jenkins-ci.org>
  - [Distributed builds](#)
  - [Post-initialization script](#)
  - [Features controlled by system properties](#)
- <http://blog.cloudbees.com>
  - [Parallelism and Distributed Builds with Jenkins](#)

### 3. Building Continuous Delivery (CD) Pipelines

This topic comprises approximately 16% of the exam. Questions cover the following topics:

- Pipeline Concepts
  - Value stream mapping for CD pipelines
  - Why create a pipeline?
  - Gates within a CD pipeline
  - How to protect centralized pipelines when multiple groups use same tools
  - Definition of binary reuse, automated deployment, multiple environments

- Elements of your ideal CI/CD pipeline - tools
  - Key concepts in building scripts (including security/password, environment information, etc.)
- Upstream and downstream
  - Triggering jobs from other jobs
  - Setting up the Parameterized Trigger plugin
  - Upstream/downstream jobs
- Triggering
  - Triggering Jenkins on code changes
  - Difference between push and pull
  - When to use push vs pull
- Pipeline (formerly known as “Workflow”)
  - Benefits of Pipeline vs linked jobs
  - Functionalities offered by Pipeline
  - How to use Pipeline
  - Pipeline stage view **[new]**
- Folders
  - How to control access to items in Jenkins with folders
  - Referencing jobs in folders
- Parameters
  - Setting up test automation in Jenkins against an uploaded executable
  - Passing parameters between jobs
  - Identifying parameters and how to use them: file parameter, string parameter
  - Jenkins CLI parameters
- Promotions
  - Promotion of a job
  - Why promote jobs?
  - How to use the Promoted Builds plugin
- Notifications
  - How to radiate information on CD pipelines to teams
- Pipeline Multibranch and Repository Scanning **[new]**
  - Usage of Multibranch jobs
  - Scanning GitHub and BitBucket Organization
  - Scanning basic SCM repositories
- Pipeline Global Libraries **[new]**
  - How to share code across Pipelines
  - Usages of the Shared Libraries
  - Interaction with Folders and Repository scanning
  - Security and Groovy sandbox

These online resources provide entry points to understanding the above topics:

- <https://jenkins.io/> [new]
  - [Handbook](#) [new]
  - [Pipeline](#) [new]
  - [Pipeline Global Shared Libraries](#) [new]
  - [Pipeline Multibranch](#) [new]
  - [Controlling the Flow with Stage, Lock, and Milestone](#) [new]
- <https://plugin.jenkins.io/> [new]
  - [Pipeline Plugin 2.4](#) [new]
- [CloudBees Knowledgebase](#)
  - [Injecting Secrets into Jenkins Build Jobs](#)
- <https://www.cloudbees.com>
  - [Credentials API Jenkins](#)
- [CloudBees Documentation](#)
  - [List views](#)
- <https://github.com>
  - [confab](#)
  - [help-spec](#)
  - [pause and resume execution](#)
  - [Executor Step Test](#)
  - [Write File Step](#)
- <http://wiki.jenkins-ci.org>
  - [Jenkins CLI](#)

#### 4. CD-as-Code Best Practices

This topic comprises approximately 10% of the exam. Questions cover the following topics:

- Distributed builds architecture
- Fungible (replaceable) agents
- Master-agent connectors and protocol
- Tool installations on agents
- Cloud agents
- Traceability
- High availability

These online resources provide entry points to understanding the above topics:

- <http://go.cloudbees.com>
  - [Cookbook](#)
  - [Distributed Builds Architecture](#)
  - [Choosing the Right Hardware](#)
  - [Architecting for Scale](#)
  - [Pipeline as Code](#) (formerly "Workflow as Code")
- <http://wiki.jenkins-ci.org>
  - [Remoting](#)

### Sample Questions

1. By definition, what does a Continuous Delivery pipeline consist of?
  - A. Backlog items
  - B. Artifacts
  - C. Stages
  - D. Tickets
  - E. Commitments
  
2. You need to execute a shell script (`/usr/bin/prepare-env`) just before a Linux agent is started. How do you achieve this?
  - A. Use the "Suffix Start Agent Command" configuration option on the agent configuration.
  - B. Use the "Prefix Start Agent Command" configuration option on the agent configuration.
  - C. Configure a `.profile` file containing a call to `/usr/bin/prepare-env` in the home directory of the OS user which runs the agent process.
  - D. Add a shell step to each Job tied to this agent to execute the shell script.
  
3. Suppose you are asked to obtain the `config.xml` of a folder (`myFolder`) from a script or HTTP client using the Jenkins Remote API. The folder exists at the root of a Jenkins master. Which URL pattern is correct for obtaining this configuration file?
  - A. `root/job/myFolder/config.xml`
  - B. `root/folder/myFolder/config.xml`
  - C. `root/myFolder/config.xml`
  - D. `root/myFolder?param=config.xml`
  - E. `root/api/getConfig?source=myFolder`

4. What architecture is recommended by the Jenkins Cookbook for a scalable Jenkins environment?

- A. Distributed Builds Architecture
- B. Central Master Architecture
- C. Automatic Builds Architecture
- D. Manual Polling Architecture
- E. One-Shot Build Architecture

5. In a Cluster Operations Job, which THREE of the following steps can be applied to a Client Master only?

- A. Install Jenkins
- B. Upgrade Jenkins
- C. Upgrade all plugins
- D. Install plugin
- E. Uninstall Jenkins

Answers to Sample Questions:

- 1. C
- 2. B
- 3. A
- 4. A
- 5. B, C, D