**Certified
Jenkins Engineer (CJE)**

Certification Exam Study Guide

CERTIFIED
2016
Jenkins Engineer

CloudBees 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 Study Guide 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.

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 **1.625.2** of the Jenkins core. Unless otherwise specified, all questions are based on an out-of-the-box standard installation of Jenkins, without any additional plugins ("base" Jenkins).

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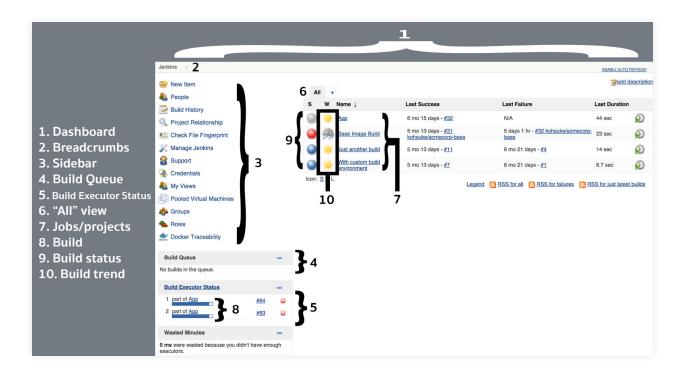
Plugins:

This exam will primarily cover questions about a "base" Jenkins installation, but knowledge of the following plugins will also be covered. Candidates are expected to know the functionality/uses of these plugins but will not be tested on detailed usage:
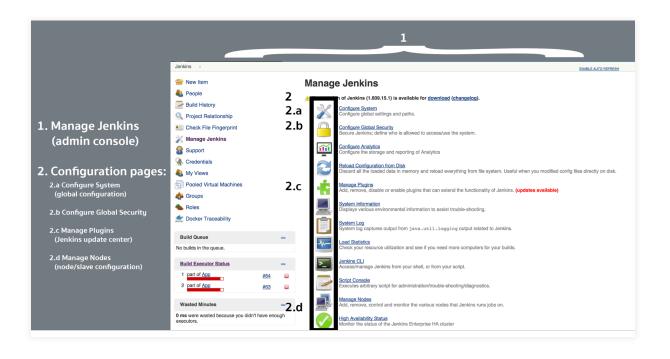
- Amazon EC2 Plugin
- Build Pipeline Plugin
- CloudBees Docker Build and Publish Plugin
- CloudBees Folders Plugin
- Copy Artifact Plugin
- Credentials Plugin
- Delivery Pipeline Plugin
- Disk Usage Plugin
- Docker Plugin
- Email-ext Plugin
- Fingerprint Plugin
- Git Plugin
- Mailer Plugin
- IRC Plugin
- JUnit Plugin
- Jabber Plugin
- Matrix Project Plugin
- NodeLabel Parameter Plugin
- Parameterized Trigger Plugin
- Pipeline Plugin (formerly known as Workflow)
- Promoted Builds Plugin
- Radiator View Plugin
- SMS Notification Plugin
- Script Security Plugin
- Skype Plugin

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:

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

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)

Manage Jenkins

New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
Support
Credentials
My Views
Pooled Virtual Machines
Groups
Roles
Docker Traceability

of Jenkins (1.609.15.1) is available for download (changelog).

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Analytics
Configure the storage and reporting of Analytics

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (updates available)

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from java.util.logging output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

Manage Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

High Availability Status
Monitor the status of the Jenkins Enterprise HA cluster

Build Queue
No builds in the queue.

Build Executor Status
1 part of App #64
2 part of App #63

Wasted Minutes
0 ms were wasted because you didn't have enough executors.

---

1. Job/project configuration page

2. Build History:
2.a Stable build
2.b Failed build

Back to Dashboard
Status
Changes
Workspace
Build Now
Delete Project
Configure
GitHub
Exports
Skip Builds
GitHub Hook Log
Git Polling Log
Move
Promote Configuration
Groups
Roles

Project name: Base Image Build
Description

[Escaped HTML] Preview

☐ Discard Old Builds
☐ Show Consolidated Build View
☐ Docker Container
☐ Enable Git validated merge support

GitHub project: https://github.com/kohsuke/docker-jenkins-demo-base/

☐ This build is parameterized
☐ Permission to Copy Artifact
☐ Promote builds when...
☐ Override the default scheduler preference
☐ Disable Build (No new builds will be executed until the project is re-enabled.)
☐ Execute concurrent builds if necessary
☐ Restrict where this project can be run

Advanced Project Options

Advanced...

Source Code Management

○ None
● Git

Build History trend
#32 kohsuke/acmecorp-base
Dec 30, 2015 1:12 PM
#31 kohsuke/acmecorp-base
Jul 24, 2015 8:50 PM
#30 kohsuke/acmecorp-base
Jul 16, 2015 7:23 AM

Save   Apply

1. Key CI/CD/Jenkins Concepts

This topic comprises approximately 27% 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

- Configuration Management (Tools such as Chef, Puppet, etc.)
  - Elements of software configuration management
  - Importance of software configuration management

- Using 3rd party tools
  - How to use 3rd party tools with Jenkins

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/articles/
  - 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://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 34% of the exam. Questions cover the following topics:

- Jobs
  - Organizing jobs in Jenkins
  - Parameterized jobs
  - Usage of Freestyle/Pipeline/Matrix/Maven/Literate

- 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, build radiators
  - 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 23% 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.)

- Upstreams and downstreams

- 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 concurrency

- Visualization
  - Options to visualize jobs' relationships
  - When to use various options for visualizing jobs' relationships
  - Information offered by a build pipeline view
  - How to set up build pipeline visualization

- 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

- CD Metrics
  - KPIs/metrics for CI/CD
  - Determining how many builds failed, succeeded
  - Determining how long a build takes
  - Determining how often code is checked-in
  - How to use metrics/KPIs
- Notifications
  - How to radiate information on CD pipelines to teams

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

- https://cloudbees.zendesk.com
  - ° Injecting Secrets into Jenkins Build Jobs

- https://www.cloudbees.com
  - ° Credentials API Jenkins

- http://documentation.cloudbees.com
  - ° List views

- https://github.com
  - ° crontab
  - ° 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 16% 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
- Containerization
- Traceability
- High availability
- Automatic repository builds

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

- http://documentation.cloudbees.com
  - ° Cookbook
  - ° Distributed Builds Architecture
  - ° Choosing the Right Hardware
  - ° Architecting for Scale
  - ° Containers
  - ° Pipeline as Code (formerly Workflow as Code)
  - ° High Availability

- 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

Answers to Sample Questions:
   1. C
   2. B
   3. A
   4. A