cloudbees/cloudbees-core-oc: Operations center

cloudbees/cloudbees-core-cm: Client controller

docker run -d -p 8080:8080 -p 50000:50000 -v jenkins-data:/var/jenkins\_home cloudbees/cloudbees-core-oc

docker run -d -p 8080:8080 -p 50000:50000 -v jenkins-data:/var/jenkins\_home cloudbees/cloudbees/cloudbees-core-cm

AGENDA

1. Overview of Jenkins and Cloud Bees CI Capabilities - Theory

2. Plugins Installation and Integration - Demo + theory

3. PipeLine fundamental ,Parameters, and Environment Variables Creation:(Decelrative Pipeline, Groovy Shared Library) Demo + theory

4. CloudBees CI enterprise Capapbilities (Pipeline template, Pipeline Policy and Configuration as Code) Theory

5. CloudBees CD/RO Feature walkthrough - theory

Session-1 - Slide

Agenda

A build agent is a computer that handles the tasks of running builds, at the direction of a controller. It can be a virtual machine (VM), a Docker container, or it can run on bare metal. Within the context of a CloudBees CI cluster, a build agent can be a shared resource for client controllers. A build agent can also be dedicated to a specific controller.

For example, you might want to test a cross-platform code base with a different operating system. In this case, you can set up different Jenkins agents or slaves and run the job against them.

Also, a Jenkins distributed architecture will reduce the load on the master server.

PLUGIN MANAGEMENT

--------------------------------------

Verified plugins in the **CloudBees Assurance Program** are both independently stable and tested as a whole to ensure compatibility with each other and the CloudBees CI instance. This provides greater stability and security for CloudBees CI environments.

Beekeeper continuously monitors your instance to check the installed plugins against the recommended configuration. The plugins page provides information a

CloudBees Update Center - Update Center will provide the Jenkins Administrator with the ability to both restrict the plugins available and host their own custom plugins.

<https://docs.cloudbees.com/docs/cloudbees-ci/latest/plugin-management/configuring-plugin-catalogs>

A plugin catalog is a configuration file that specifies plugins and their associated versions.

**When to plugin catalog**

A plugin catalog is useful when you want to:

* Ensure that certain plugins are kept at a consistent version.
* Provide an alternative location for plugins if Jenkins is not connected to the internet.
* Install and maintain plugins that are not in CloudBees Assurance Program, for example custom plugins or older versions of plugins.

Using Jenkins CLI add your plugin-catalog.json to your operations center

jenkins-cli plugin-catalog --put < additional-pipeline-plugins-catalog.json

The following Pipeline project types are available:

* [Declarative Pipeline](https://docs.cloudbees.com/docs/cloudbees-ci/latest/pipelines/types-of-pipelines#_declarative_pipeline)
* [Scripted Pipeline](https://docs.cloudbees.com/docs/cloudbees-ci/latest/pipelines/types-of-pipelines#_scripted_pipeline)
* [Freestyle projects](https://docs.cloudbees.com/docs/cloudbees-ci/latest/pipelines/types-of-pipelines#_freestyle_projects)
* [Multibranch Pipelines](https://docs.cloudbees.com/docs/cloudbees-ci/latest/pipelines/types-of-pipelines#_multibranch_pipelines)

## Features for Pipeline standardization

**Pipeline Templates**

Pipeline Templates can be used to define reusable Pipelines. This helps in bringing about consistency across your Pipelines.

**Pipeline Policies**

Pipeline Policies are runtime validations that work for both Scripted and Declarative Pipelines and provide administrators a way to include warnings for or block the execution of Pipelines that do not comply with certain regulatory requirements, rules, or best practice guidelines. For more information on concepts related to Pipeline Policies, see [Using Pipeline Policies](https://docs.cloudbees.com/docs/cloudbees-ci/latest/pipelines/pipeline-policies).

**Organization folders**

Organization folders are a way to automatically create Multibranch projects for every repository in a GitHub organization or Bitbucket team. For more information, please see [Organization folders](https://docs.cloudbees.com/docs/cloudbees-ci/latest/pipelines/pipeline-as-code#_organization_folders).

**Configuration as Code for controllers**

Configuration as Code for controllers simplifies the management of a CloudBees CI cluster by capturing the configuration of CloudBees CI controllers in human-readable declarative configuration files which can then be applied to a controller in a reproducible way.

<https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/>

To create a new configuration bundle, create each of the files listed below. A CasC bundle consists of the following files:

* [**bundle.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_bundle_yaml)**:** This file is an index file that describes the bundle, and references the other files in the bundle.
* [**jenkins.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_jenkins_yaml)**:** This file contains the Jenkins configuration, as defined by the Configuration as Code plugin.
* [**plugins.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_plugins_yaml)**:** (Optional) This file contains a list of all plugins that should be installed.
* [**plugin-catalog.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_plugin_catalog_yaml)**:** (Optional) This file contains the plugin catalog definitions that should be created on the controller.
* [**items.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_items_yaml)**:** (Optional) This file contains the items to be created.
* [**rbac.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_rbac_yaml): (Optional) This file contains all role-based access control (RBAC) groups and roles defined at the root level.
* [**variables.yaml**](https://docs.cloudbees.com/docs/cloudbees-ci/latest/casc-controller/create-bundle#_variables_yaml): (Optional) This file defines the variables that can be used in the jenkins.yaml, items.yaml, and rbac.yaml files.

In a release pipeline, an approval needs to happen by a manager before a stage can start. What is used in the pipeline to represent the required approval?

stage

audit

gate

access control

In CloudBees CD/RO, what component allows the reuse of releases by providing a flexible abstraction of pipeline elements that can be assembled in other combinations that are predictable and repeatable?

Release Models

Pipeline Controls

Command Center

A Release Dashboard

You want to see release trends across multiple releases. Which dashboard or report do you use?

Audit and Compliance reports

Release Command Center dashboard

DORA metrics dashboard

Releases dashboard

The CloudBees Analytics server can collect data from external tools such as JIRA and ServiceNow. Where is this data displayed?

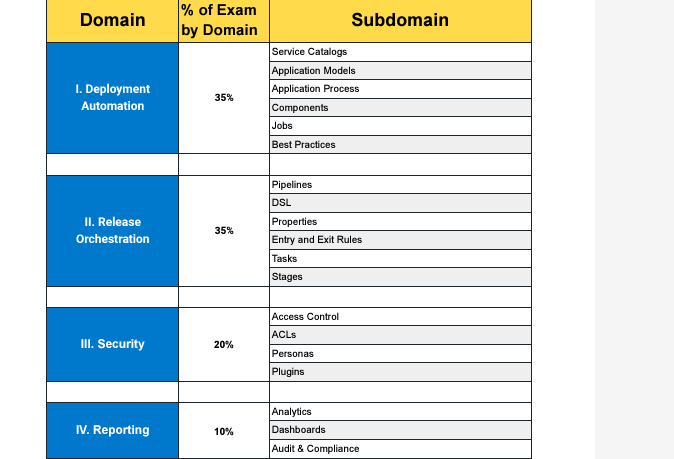
Operations Center

The Primary Control Center (PCC)

CD/RO Helm Plugin

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Topic | Sub topic |
| Cloudbees | 1. | General understanding walkthrough of cloudbees | Introduction to CloudBees Jenkins and its role  Overview of key features and benefits. |
| 2. | Plugins Installations. | Overview of Jenkins plugins and their role in extending functionality.  Highlighting some essential plugins for CI/CD. |
| 3. | Integrations with various tools configurations | Configuring Build Agent  Integration with Tools like Docker, Maven |
| 4. | CI creations | Creating and configuring CI pipelines in CloudBees Jenkins |
| 5 | template, parameters and environment vars creation | Parametrizing the pipeline for different environment |
| 6 | DSL editor config file walkthrough of pipeline | Decelerative/scripted Pipeline explanation using DSL.  Automating Jenkins file using Pipeline Template Cataloglog |
| 7. | CD/RO Introduction | Walkthrough of CD/RO platform |
|  | 8. | CD/RO dependent configurations and checkpoints creations |  |

The Release Command Center dashboard



|  |  |  |
| --- | --- | --- |
| Cloudbees CI | S.No | Topics |
| 1 | General understanding walkthrough of cloudbees-Jenkins |
| 2 |  |
| 3 | CI creations |
|  |  |
|  |  |
| 6 | Plugins installations |
| 7 | DSL editor config file walkthrough of pipeline |
| 8 | Integrations with various tools configurations |