



Introduction to Declarative Pipeline

on CloudBees Core for Kubernetes

www.cloudbees.com

What's Wrong With Freestyle Jobs?

While the Freestyle job type has served the Hudson/Jenkins community well for years it has some major issues including:

- **UI Bound** - The configuration of a job is limited to what can be expressed via the limits of the Jenkins' UI and doesn't allow for building complicated workflows with features like:
 - Control over where builds are executed
 - Flow control (if-then-else, when, try-catch-finally)
 - Ability to run steps on multiple agents
 - Ability to run steps in parallel
- **Not Auditable** - The creation and editing of jobs isn't auditable without using additional plugins

What is a Jenkins Pipeline?

Jenkins Pipeline (formerly known as Workflow) was introduced in 2014 and built into Jenkins 2.0 when it was released.

Pipelines are:

- A Job type - The configuration of the job and steps to execute are defined in a script (Groovy or Declarative based with a Domain Specific Language) that can be stored in an external SCM
- Auditable - changes can be easily audited via your SCM
- Durable - can keep running even if the master fails
- Distributable - pipelines can be run across multiple agents including execution of steps in parallel
- Pausable - can wait for user input before proceeding
- Visualizable - enables status-at-a-glance dashboards like the built in Pipeline Stage View and Blue Ocean

Why You Should Use Declarative Instead of Scripted

While Declarative Pipelines use the same execution engine as Scripted pipelines

Declarative adds the following benefits:

- Easier to Learn - the Pipeline DSL (Domain Specific Language) is more approachable than Groovy making it quicker to get started
- Richer Syntax - Declarative provides richer syntactical features over Scripted Pipeline syntax
- Syntax Checking - Declarative syntax adds the following types of syntax checking that don't exist for Scripted pipelines:
 - Immediate runtime syntax checking with explicit error messages.
 - API and CLI based file linting

What is a Multibranch Pipeline?

The **Multibranch Pipeline** project type enables you to implement different Jenkinsfiles for different branches of the same project. In a Multibranch Pipeline project, Jenkins **automatically discovers, manages and executes** Pipelines for branches which contain a Jenkinsfile in source control.

A **Github Organization** or **Bitbucket Project Pipeline Project** scans for repositories that have a Jenkinsfile and creates a **Multibranch Pipeline** project for each one it finds.

Core v2 on K8s Features



Quick Install

- Industry standard tool (kubectl) for installation and updates
- Initial install takes a few minutes on an existing cluster



Built-in Fault Tolerance

- JENKINS_HOME is managed externally of cluster nodes
- Unhealthy CJOC and Masters are automatically restarted on a healthy node



Elastic and Ephemeral Jenkins Agents

- Dynamically spins up K8s pods as agents, stops and removes on job completion



Straightforward, Fast, Dynamic Provisioning of Jenkins Masters

- Easy and quick to create Masters of any size
- Entire Jenkins Master lifecycle is managed in CJOC
- Team Masters with recipes, plugin catalogs and simplified RBAC model

Core v2 Traditional and Modern Features



Role Based Access Control

- SSO between CJOC and Masters
- Hierarchical authorization strategies for Masters and Folders



CloudBees Assurance Program with Beekeeper

- CloudBees Verified Plugins, including top OSS Plugins, and LTS Core
- Integrated with CloudBees Custom Update Center



Shared Agents

- Share static agents across multiple Masters

Job Templating

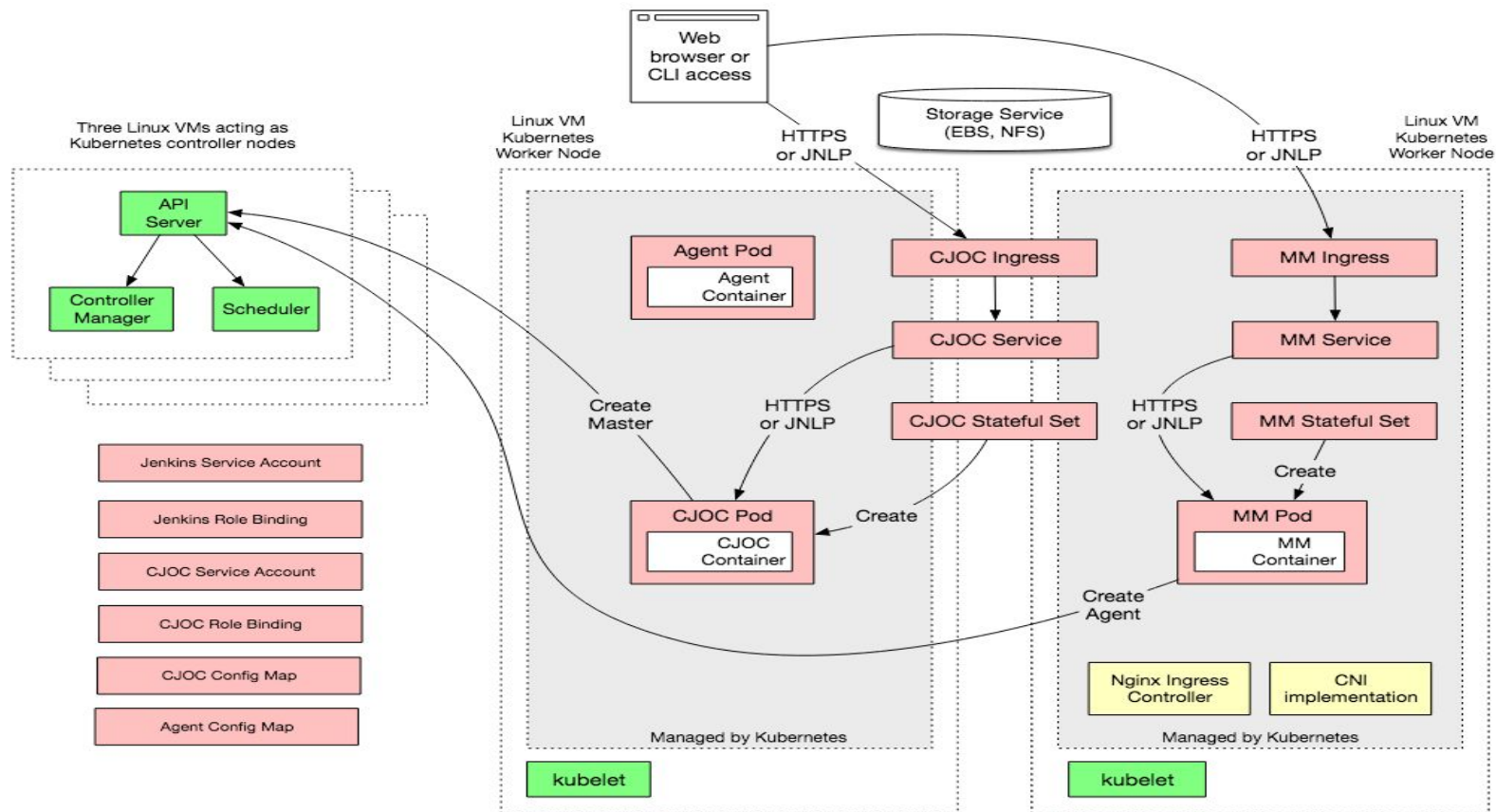


Expert Support

- Support from the experts 24/7
- Extensive Knowledge Base and Online Training



CloudBees Core on K8s Architecture



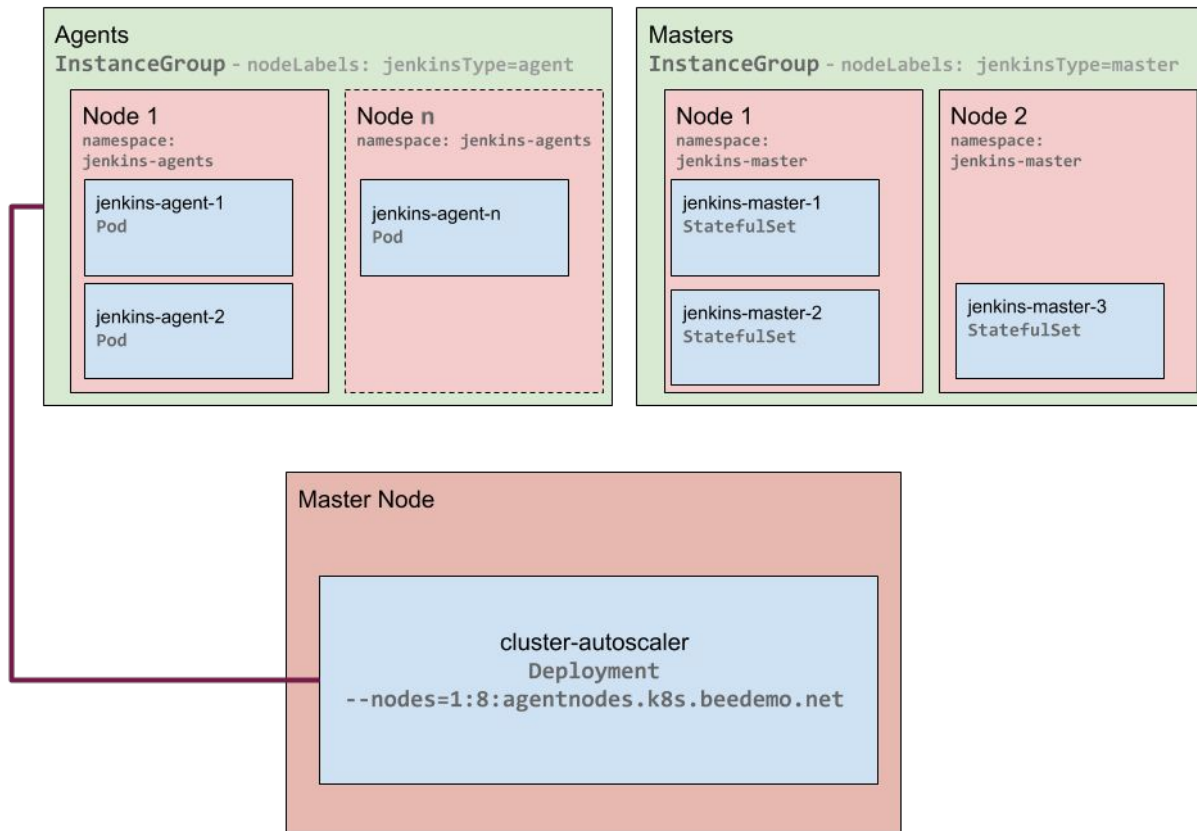
Workshop Cluster Configuration

- [PodSecurityPolicy](#) - privileged and restricted
- Auto-scaling agent pool in dedicated namespace - [PodNodeSelector](#)
 - kops edit ig agentnodes
 - overprovisioning configuration - [Priority Admission Controller](#)
- Masters in dedicated agent pool and namespace, but don't autoscale
 - kops edit ig nodes
 - [Custom container image](#)
- Kaniko is used for all builds with all pushes to ECR
 - automatically apply ECR rules to clean-up container images
- Custom JNLP agent utilizes config map to get agent.jar



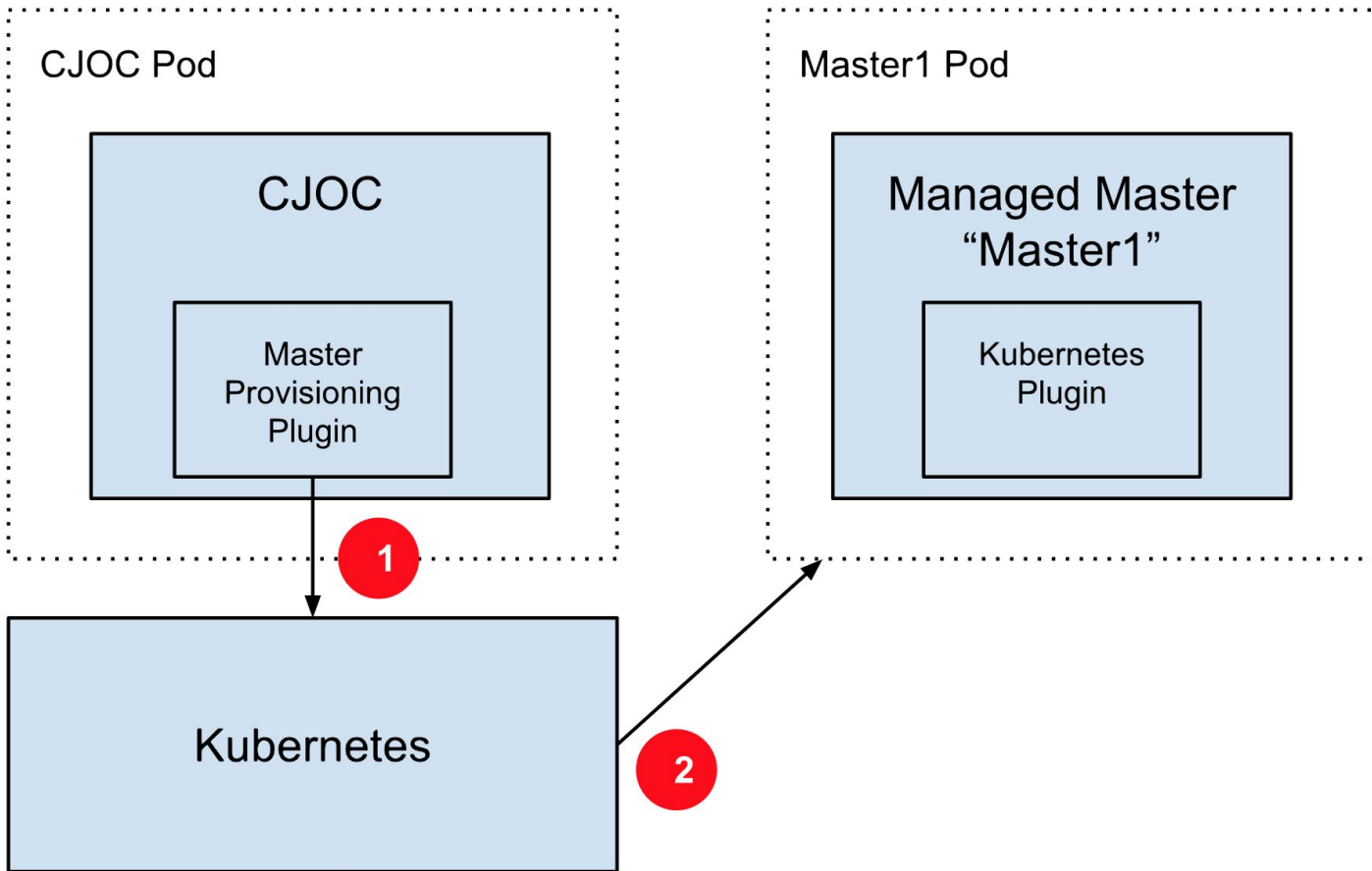
Autoscaling Jenkins Agents with Kubernetes

Reduce slow-downs related to lack of infrastructure



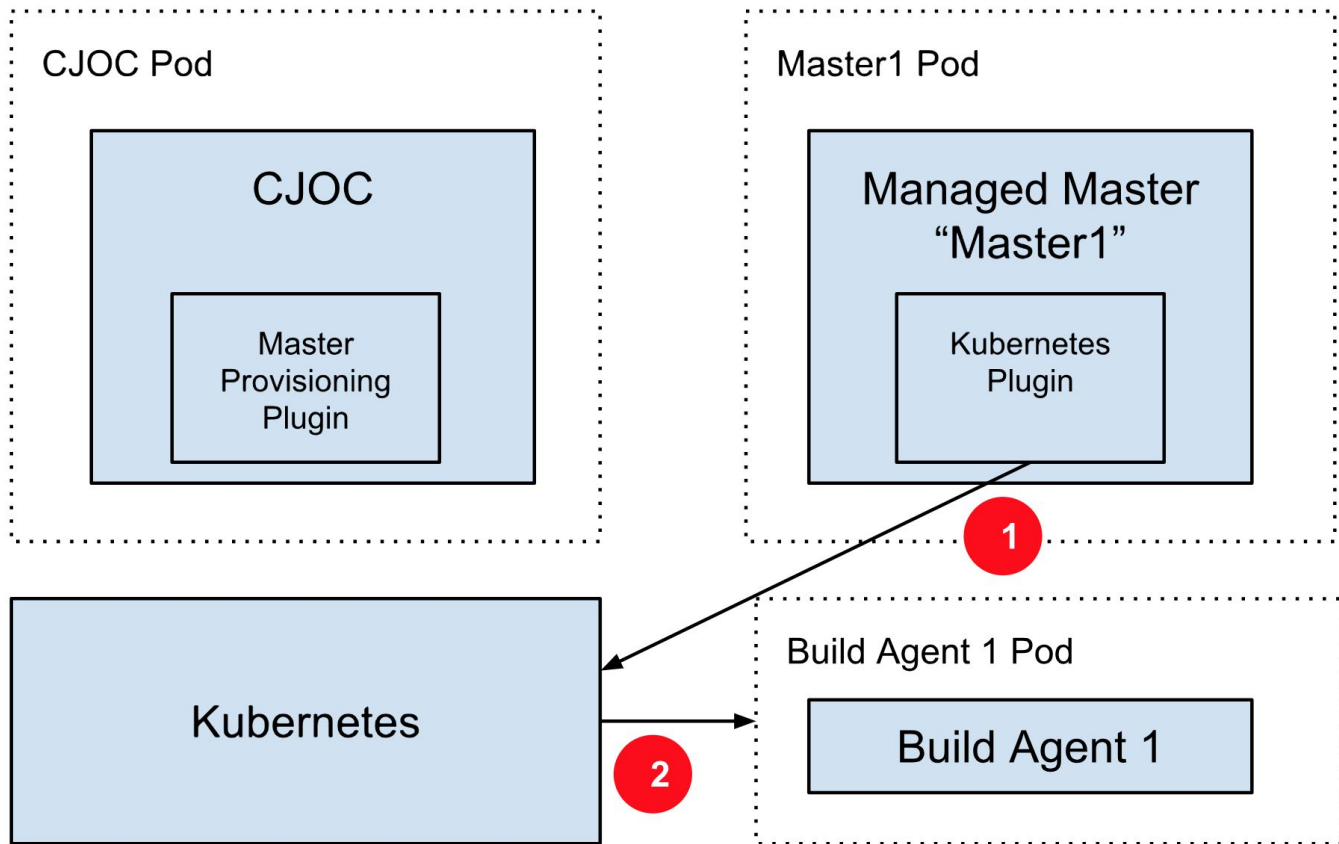


Master Provisioning





Agent Provisioning





Pipeline Template Catalogs

Advanced Pipeline Templates for Core v2



Why?

- **Organization Wide Governance:** ensure teams follow centralized dev process
- **Multi-branch/Org Folder Jobs:** CB Pipeline Job Templates don't support Multi-branch/Org Folder Jobs.
- **Replay:** Anyone with build permissions on a Pipeline Job Template or Custom Pipeline as Code Script based job can *Replay* it and make changes to the Pipeline script.
- **Not CasC:** Although the Pipeline Job Template can be version controlled, the Template itself must be created and managed via Jenkins.
- **No Parameters:** Custom Pipeline as Code Scripts don't support parameters.
- **No marker file support:** CB Pipeline Job Templates don't support custom marker files.