

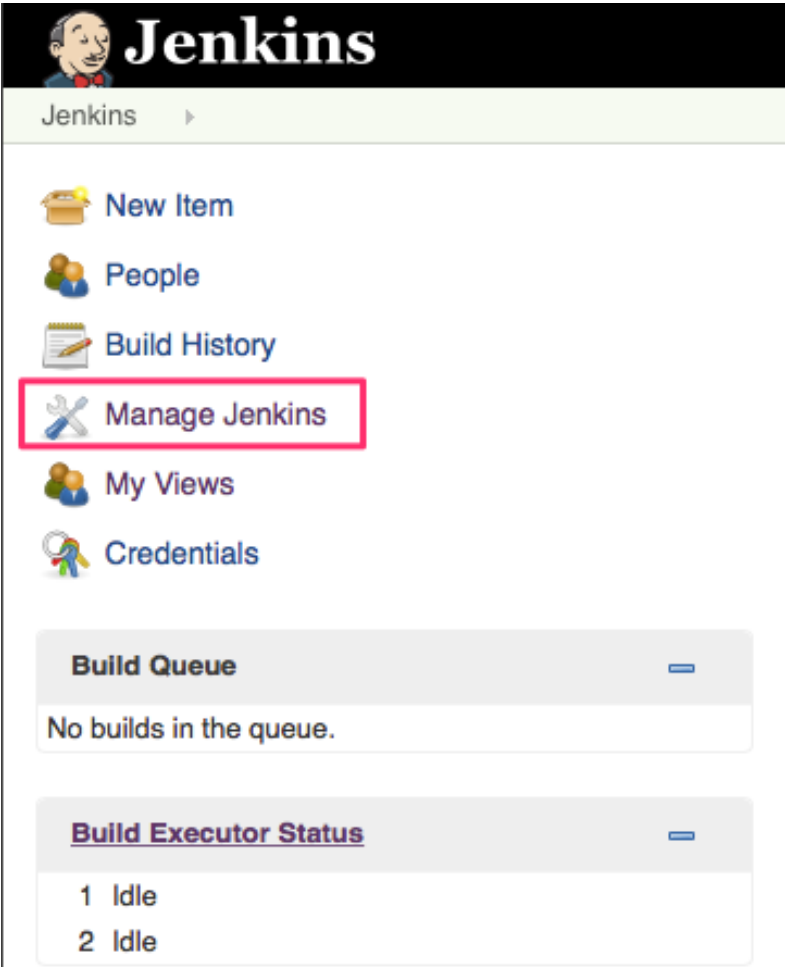
Setup Details:

- Server: 10.9.8.45
- SSH Credentials: root/Test!2345
- Jenkins Credentials: ansible/test123

Step 1: Install necessary plugins

- Git Client Plugin
- Slack Notification Plugin
- Environment Injector Plugin

This can be done by selecting “Manage Jenkins” and “Manage Plugins”:



Manage Jenkins

New version of Jenkins (2.19.4) is available for download (changelog).

Slave to master security subsystem is currently off. Please read the documentation and consider turning it on.

Configure System

Configure global settings and paths.

Configure Global Security

Secure Jenkins, define who is allowed to access/use the system.

Configure Credentials

Configure the credential providers and types

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)

Filter: \slack\

Updates Available Installed Advanced

Install	Name	Version
<input type="checkbox"/>	Slack Upload Plugin Allows users to upload files to Slack	1.7
<input type="checkbox"/>	Cucumber Slack Notifier Plugin A plugin to send a summarised cucumber report to a slack channel.	0.8.3
<input type="checkbox"/>	Slack Notification Plugin This plugin allows to post build notifications to a Slack channel.	2.1

Install without restart

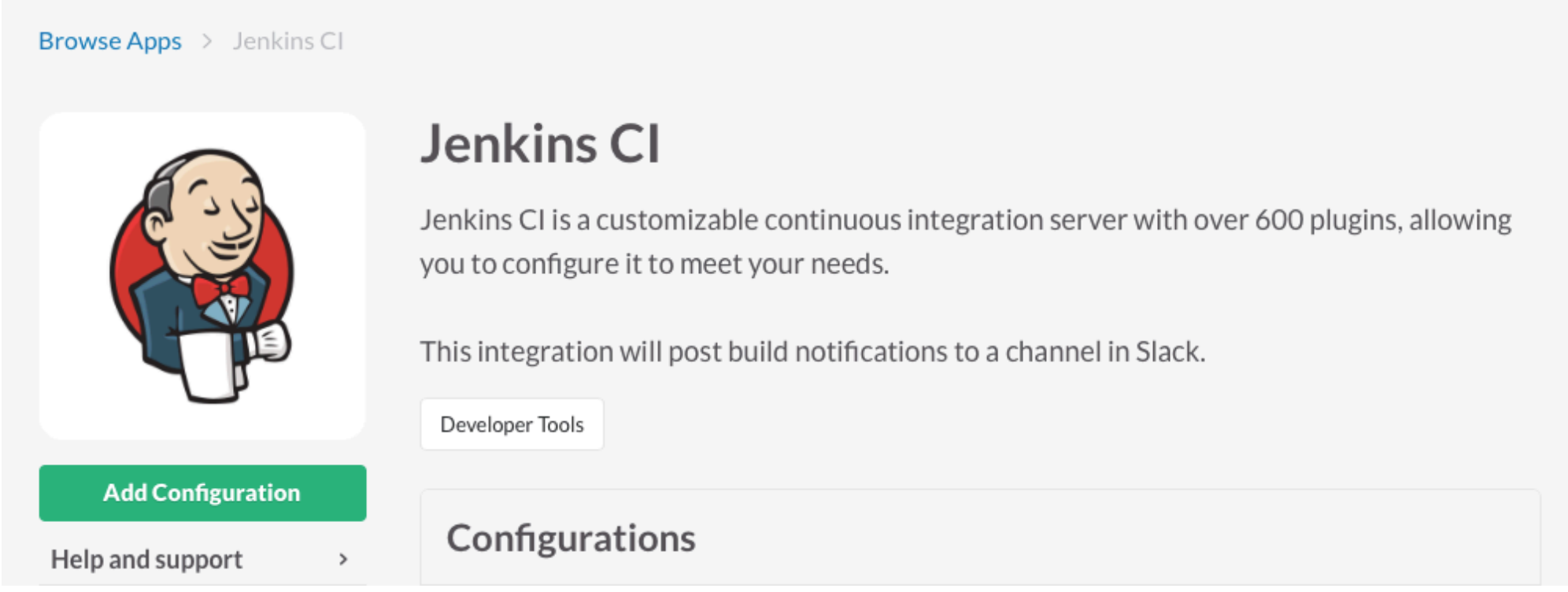
Download now and install after restart

Update information obtained: 15 hr ago

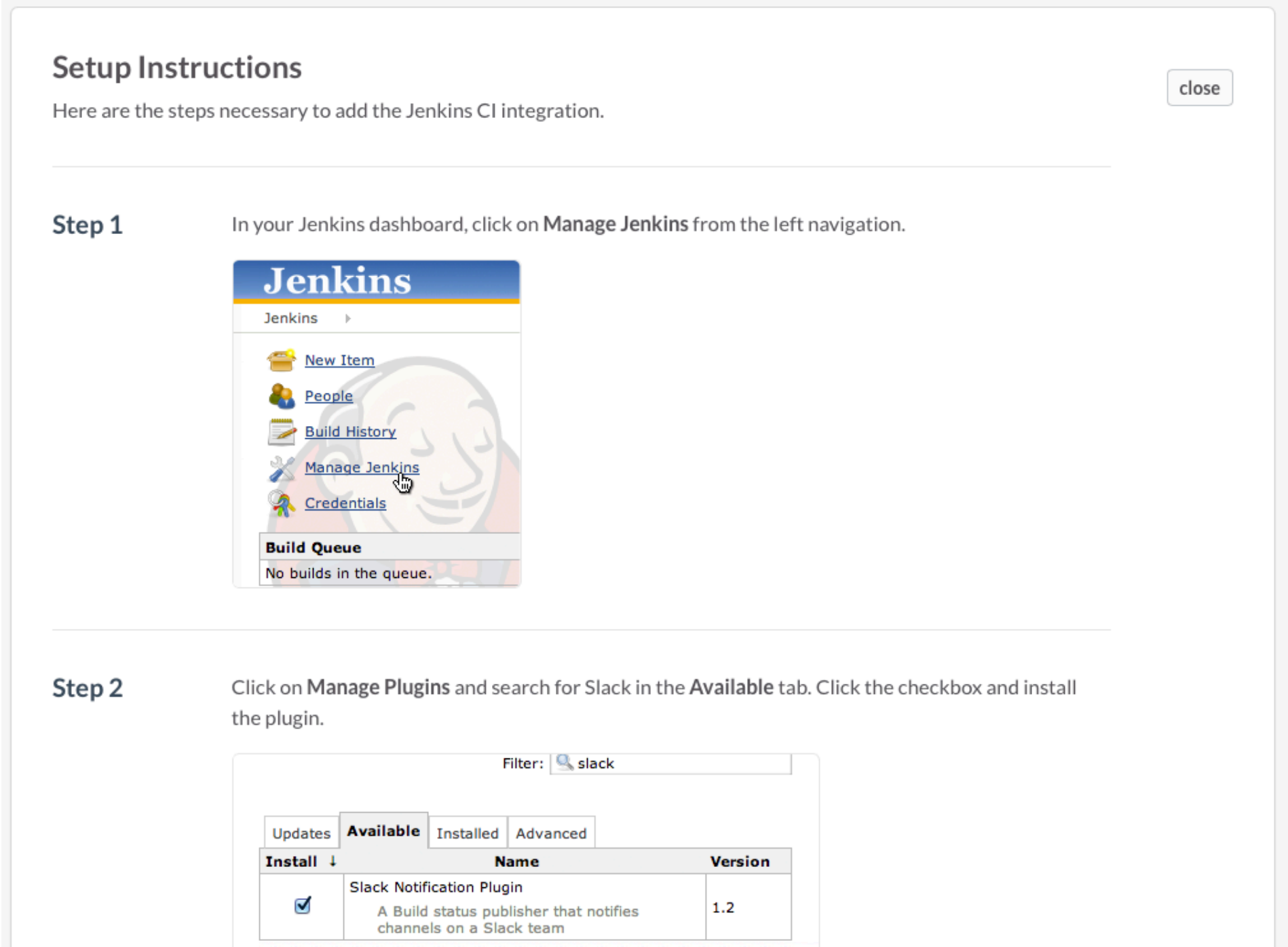
Check now

Step 2: Integration with Slack

- Install “jenkins” app in Slack



- Follow the setup instructions given by Slack [This page will have “Integration Token”]



- Add team-domain name, integration token and Channel name on Jenkins page

Global Slack Notifier Settings

Team Subdomain

pluribusautomation

Integration Token

xxxxxxxxxxxxxxxxxxxxxxxx

Exposing your Integration Token is a security risk. Please use the Integration Token Credential ID

Integration Token Credential ID

- none - Add

Channel

jenkins

Success

Test Connection

Save Apply

Step 3: Setup build jobs (Integration with Github)

- Setup password-less login for jenkins to the build server
 - Generate key pairs on jenkins server and copy the public key to build-server's .ssh/authorized_keys file
- Create a new build job
- Specify “Name, Description” for your project

Project name

Ansible-Syntax-Tests

Description

This job will do basic syntax testing for any new commit to the pluribus-ansible repo.

[Raw HTML] Preview

- Select SCM (Git) and setup credentials accordingly

Source Code Management

None

Bazaar

Git

Repositories

Repository URL

https://github.com/amits/pluribus-ansible/

Credentials

aashjain (Credentials)

Add

- Because Jenkins server is on a private network and not on a public network (hence, not accessible by Github), we are using poll model to know about any commits to the source repo. And hence, we must specify how often we want to poll. (time format is same as cron's time format)

Build Triggers

Trigger builds remotely (e.g., from scripts)

Build after other projects are built

Build periodically

Poll SCM

Schedule

* * * * *

Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Tuesday, December 20, 2016 6:59:37 AM UTC; would next run at Wednesday, December 21, 2016 6:00:37 AM UTC.

Ignore post-commit hooks

- Specify what this build job is supposed to do, options can be found in “Add build setup” button. For example, we do the following:
 - Inject environment variables
 - Execute shell script to do specify set of tasks

Build

Inject environment variables

Properties File Path

Properties Content

buildServer=10.9.8.200
userName=jenkins
jobWorkspace=/home/jenkins/pluribus-ansible
vaultPassword=/home/jenkins/jenkins-deps/vault_pass.txt
hosts1=/home/jenkins/jenkins-deps/hosts1

Execute shell

Command

#!/usr/bin/env bash
ssh \$userName@buildServer "cd \$jobWorkspace; git pull"

Execute shell

Command

#!/usr/bin/env bash
ssh \$userName@buildServer "cd \$jobWorkspace; find . -maxdepth

Add build step

- Since we have integrated with Slack, we can specify what type of notification Jenkins should send to Slack. This is achieved by adding Slack Notifications from “Add post-build action” and check boxes according to your usage

Post-build Actions

Slack Notifications

Notify Build Start

Notify Aborted

Notify Failure

Notify Not Built

Notify Success

Notify Unstable

Notify Back To Normal

Add post-build action

- Save it and you're done

Save Apply