



A Getting Started Guide to Setting up Jenkins

by Greg Sybolt MVB · Sep. 21, 16 · DevOps Zone

“Automated Testing: The Glue That Holds DevOps Together” to learn about the key role automated testing plays in a DevOps workflow, brought to you in partnership with Sauce Labs.

The goal of this getting started guide is to help teams get Jenkins continuous integration (CI) servers configured and discover how to make a newly deployed CI infrastructure fully operational. Jenkins is a leading open source CI server. It is flexible, providing hundreds of plugins to support building, testing, and deployment, and is capable of automating any project. Jenkins CI infrastructure can be deployed to on-prem, in the cloud using configuration management tools, and third-party vendor. For the purpose of this article, let's assume that our Jenkins CI servers are deployed in the cloud and focus on configuration of Jenkins' web interface. I will walk through various processes and steps to set up a new Jenkins CI server for production.

Recommended Best Practices for CI Architecture

Don't jump head first into configuration and creating a pipeline without planning, designing, and establishing standards for your CI architecture. Taking the time to think about infrastructure first will enable a stable and restorable infrastructure. Let's review a few recommended best practices to consider in your future CI pipeline.

Backup CI Server (Fail Safe)

It may seem obvious, but I recommend setting up a backup process for Jenkins configuration. Script a Jenkins job to use the thinBackup plugin or S3 plugin to send the Jenkins configuration to an Amazon S3 (cloud storage).

Pipeline Configuration

Here are recommendations to consider:

- **Set environment variables** (i.e., hidden passwords, SSH keys, API keys, etc.)
- **Security**—create generic reusable jobs and naming conventions (i.e., jobs and environment variables)
- **Keep jobs small**—modulization and scalable infrastructure that allow for auto-scaling of Slave nodes

Discovery of Plugins

Be conservative with plugin usage. In my experience, plugins change all the time and become fragile. For the discovery mode of finding the right plugins for your CI infrastructure, I recommend evaluating if you can script the same functionality yourself rather than being dependent on third-party plugins. Just minimize your plugin usage. Having an upgrade process is important. Don't test upgrades on the live production CI server. Set up a test environment to check all new plugins or new Jenkins versions using existing jobs before applying a change to your production server.

Configuration Management Tooling

I highly suggest using a configuration management tool to automate the process to achieve speed, scale, repeatability, and consistency. I recommend using Scalr and Chef tool solutions.

Repositories

Consider creating GitHub webhooks for version control repositories. I recommend using Artifactory (cloud storage) for CI artifacts.

The Basics of Setting Up a Jenkins Master Server

In this section of the guide, I will walk you through some steps to set up and manage Jenkins.

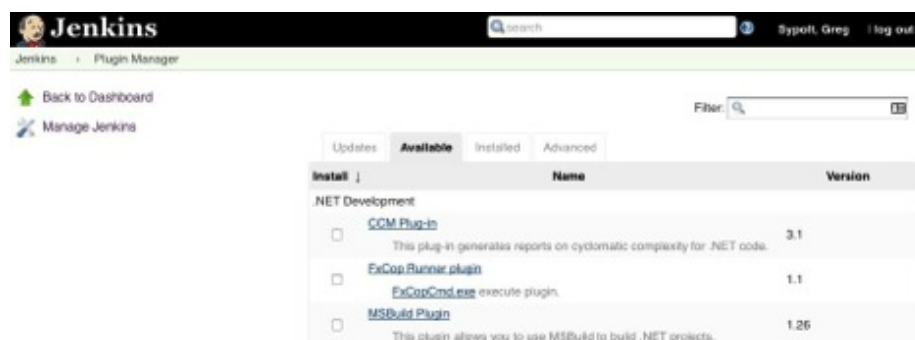
Discover, Install and Manage Plugins

Jenkins has hundreds of useful plugins. Plugins will eliminate the need to create custom scripting to solve common problems with minimal pain. Just remember plugins change often and become fragile. Minimize your plugin usage if possible.

Steps

1. Navigate to Jenkins Web Interface > **Login as Admin** > **Manage Jenkins** > **Manage Plugins** > **Available (Tab)**.

2.



3. Select the checkbox for all the plugins you want to install.
4. Select “download now and install after restart” at the bottom of the page.



5. After Jenkins finishes restarting, the installed plugins will appear under **Manage Plugins**.

5. After Jenkins finishes restarting, the installed plugins will appear under **Manage Plugins > Installed (Tab)**.

Here are a few Jenkins plugin recommendations:

- **Git:** allows you to integrate GitHub to clone repository.
- **GitHub Pull Request Builder:** builds pull requests in GitHub and reports results.
- **Swarm:** enables slaves to auto-discover the nearby Jenkins master and join it automatically.
- **Sauce OnDemand:** allows you to integrate Sauce Labs Selenium testing with Jenkins.
- **Pipeline:** suite of plugins that lets you orchestrate automation, simple or complex.
- **Slack:** allows posting of build notifications to a Slack channel.
- **Thinbackup:** simply backs up the global and job-specific configurations.

Configure System Settings

As an admin of Jenkins, the Configure System page is a critical configuration section. This page represents a variety of sections, each correlating to a different configuration area from generic Jenkins settings and defining global environment variables. Most installed plugins are configured on this page.

Steps

1. Navigate to Jenkins Web Interface > **Login as Admin > Manage Jenkins > Configure System**.
2. Configure the root directory for workspace and build record.

Home directory	/opt/jenkins
Workspace Root Directory	<input data-bbox="820 1218 1439 1267" type="text" value="\${JENKINS_HOME}/workspace/\${ITEM_FULLNAME}"/>
Build Record Root Directory	<input data-bbox="820 1279 1439 1328" type="text" value="\${ITEM_ROOTDIR}/builds"/>

3. Set the Jenkins Master executors to 0, setting up separate cloud instances to be the workers; Jenkins Master will only be the orchestrator.

# of executors	<input data-bbox="979 1451 1308 1500" type="text" value="0"/>
----------------	---

4. Set environment variables (i.e. SLACK_TOKEN, SAUCE_API_KEY).

GitHub Pull Request Builder	
GitHub Auth	<input data-bbox="820 1608 1308 1644" type="text" value="GitHub Server API URL"/> <input data-bbox="979 1615 1308 1644" type="text" value="https://api.github.com"/> <input data-bbox="820 1655 1308 1691" type="text" value="Jenkins URL override"/> <input data-bbox="820 1702 1308 1738" type="text" value="Shared secret"/> <input data-bbox="979 1749 1123 1778" type="text" value="Automation"/> <input data-bbox="1139 1749 1225 1778" type="button" value="Add"/>

5. Configure the installed plugins (i.e. GitHub, Sauce Labs, Slack, etc.). This typically involves adding API keys or shared secrets.

Configure Global Settings

Straight out of the box, Jenkins will allow anyone to run anything as a user of Jenkins along with admin permissions, which is bad. I suggest enabling Lightweight Directory Access Protocol (LDAP), which allows you to use corporate service. Users can log into Jenkins with their usual company login

credentials.

Steps

1. Navigate to Jenkins Web Interface > **Login as Admin** > **Manage Jenkins** > **Configure Global Security**.
2. Select the checkbox to enable security.



3. Set TCP port for JNLP slave agents to 9000.
4. Select LDAP from the Access Control (Security Realm) section and enter your LDAP server address:



5. Select matrix-based security from the Access Control (Authorization) section.
6. Select the checkbox for Prevent Cross Site Request Forgery Exploits, and Enable Slave → Master Access Control.

Add Node (Slave)

There are multiple ways to add and configure a slave node farm. In all cases, a build, test, or deploy job are being run on a slave node. When it comes to creating multiple slave nodes, it is typically a straightforward process.

Advanced – we create a farm on Scalr with a Jenkins Master server and at least a handful of slave nodes for building and testing the application. We use the Swarm plugin to enable slave nodes to auto-discover a nearby Jenkins Master and join it automatically. You can also manually run a curl command and add slave nodes to the Jenkins Master. It's clear as mud, right? In layman terms, a slave node is a machine configured to clone the source code, compile, execute tests, or any tools needed. Lastly, the build results are stored on the Jenkins Master, and artifacts should always end up in artifact storage (i.e. Artifactory) not on the Jenkins Master.

Steps

1. Check that the new Node is **ONLINE**.
2. Navigate to Jenkins Web Interface > **Login as Admin** > **Manage Jenkins** > **Manage Nodes**

Jenkins is a very flexible application that involves a great deal of control for admins. It is easy to see when accessing the Jenkins admin page that we have only scratched the surface but have laid a solid foundation to continue learning how to set up and manage Jenkins.

Go Time! Ready to Create a CI Pipeline

Now that Jenkins is configured, you are ready to orchestrate or maintain a CI pipeline. I recommend checking out the Jenkins documentation Getting Started with Pipeline to create and deploy your first CI pipeline.



Source: <https://jenkins.io/images/pipeline/realworld-pipeline-flow.png>

Conclusion

The main motivation behind this article is to identify the basic settings and recommended best practices for setting up Jenkins Master and Slave architecture, then have the incentive and convenience to participate in CI architecture, along with enforcing discipline by assigning blame if the build, deployment, or test fails.



Source: CloudBees

Greg Sypolt (@gregsypolt) is a Senior Engineer at Gannett – USA Today Network and co-founder of Quality Element. He is a passionate automation engineer seeking to optimize software development quality while coaching team members on how to write great automation scripts and helping the testing community become better testers. Greg has spent most of his career working on software quality—concentrating on web browsers, APIs, and mobile. For the past 5+ years, he has focused on the creation and deployment of automated test strategies, frameworks, tools and platforms and continuous integration.

Learn about the importance of automated testing as part of a healthy DevOps practice, brought to you

in partnership with Sauce Labs.

Like This Article? Read More From DZone



NGINX Amplify Reports Improve Server Configuration



Logging AWS Elastic Beanstalk With ELK



SpringOne Replay: Implementing and Extending Config Server



Free DZone Refcard Getting Started With Docker

Topics: JENKINS , SERVER , CONFIGURATION

Published at DZone with permission of Greg Sypolt, DZone MVB. [See the original article here.](#)

Opinions expressed by DZone contributors are their own.

Get the best of DevOps in your inbox.

Stay updated with DZone's bi-weekly DevOps Newsletter. [SEE AN EXAMPLE](#)

SUBSCRIBE

DevOps Partner Resources

The Ultimate Monitoring Tools Landscape

Papertrail



Setting Up the Critical DevOps Role of Enterprise Release Management

Plutora



The DevOps Journey - From Waterfall to Continuous Delivery

Sauce Labs



Redefine application release processes and unify your DevOps strategy.

Atomic

