✔ How To Install Jenkins on Ubuntu 18.04

Posted July 9, 2018   👁 27.1k   SYSTEM TOOLS   UBUNTU   UBUNTU 18.04

By: Melissa Anderson   By: Kathleen Juell

Not using **Ubuntu 18.04**? Choose a different version:

Ubuntu 16.04	❯
Automated: Docker	request
Automated: Bash	request

Introduction

Jenkins is an open-source automation server that automates the repetitive technical tasks involved in the continuous integration and delivery of software. Jenkins is Java-based and can be installed from Ubuntu packages or by downloading and running its web application archive (WAR) file — a collection of files that make up a complete web application to run on a server.

In this tutorial, you will install Jenkins by adding its Debian package repository, and using that repository to install the package with `apt` .

Prerequisites

To follow this tutorial, you will need:

- One Ubuntu 18.04 server configured with a non-root sudo user and firewall by following the Ubuntu 18.04 initial server setup guide. We recommend starting with at least 1 GB of RAM. See Choosing the Right Hardware for Masters for guidance in planning the capacity of a production Jenkins installation.

- Java 8 installed, following our guidelines on installing specific versions of OpenJDK on Ubuntu 18.04.

Step 1 — Installing Jenkins

The version of Jenkins included with the default Ubuntu packages is often behind the latest available version from the project itself. To take advantage of the latest fixes and features, you can use the project-maintained packages to install Jenkins.

First, add the repository key to the system:

```
$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

When the key is added, the system will return `OK`. Next, append the Debian package repository address to the server's `sources.list`:

```
$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins
```

When both of these are in place, run `update` so that `apt` will use the new repository:

```
$ sudo apt update
```

Finally, install Jenkins and its dependencies:

```
$ sudo apt install jenkins
```

Now that Jenkins and its dependencies are in place, we'll start the Jenkins server.

Step 2 — Starting Jenkins

Let's start Jenkins using `systemctl`:

```
sudo systemctl start jenkins
```

Since `systemctl` doesn't display output, you can use its `status` command to verify that Jenkins started successfully:

```
$ sudo systemctl status jenkins
```

If everything went well, the beginning of the output should show that the service is active and configured to start at boot:

Output

- jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Mon 2018-07-09 17:22:08 UTC; 6min ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1153)
   CGroup: /system.slice/jenkins.service

Now that Jenkins is running, let's adjust our firewall rules so that we can reach it from a web browser to complete the initial setup.

Step 3 — Opening the Firewall

By default, Jenkins runs on port `8080`, so let's open that port using `ufw`:

```
$ sudo ufw allow 8080
```

Check `ufw`'s status to confirm the new rules:

```
$ sudo ufw status
```

You will see that traffic is allowed to port `8080` from anywhere:

```
Output
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
8080                       ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
8080 (v6)                  ALLOW       Anywhere (v6)
```

Note: If the firewall is inactive, the following commands will allow OpenSSH and enable the firewall:

```
$ sudo ufw allow OpenSSH
$ sudo ufw enable
```

With Jenkins installed and our firewall configured, we can complete the initial setup.

Step 4 — Setting Up Jenkins

To set up your installation, visit Jenkins on its default port, `8080`, using your server domain name or IP address: `http://`your_server_ip_or_domain`:8080`

You should see the **Unlock Jenkins** screen, which displays the location of the initial password:
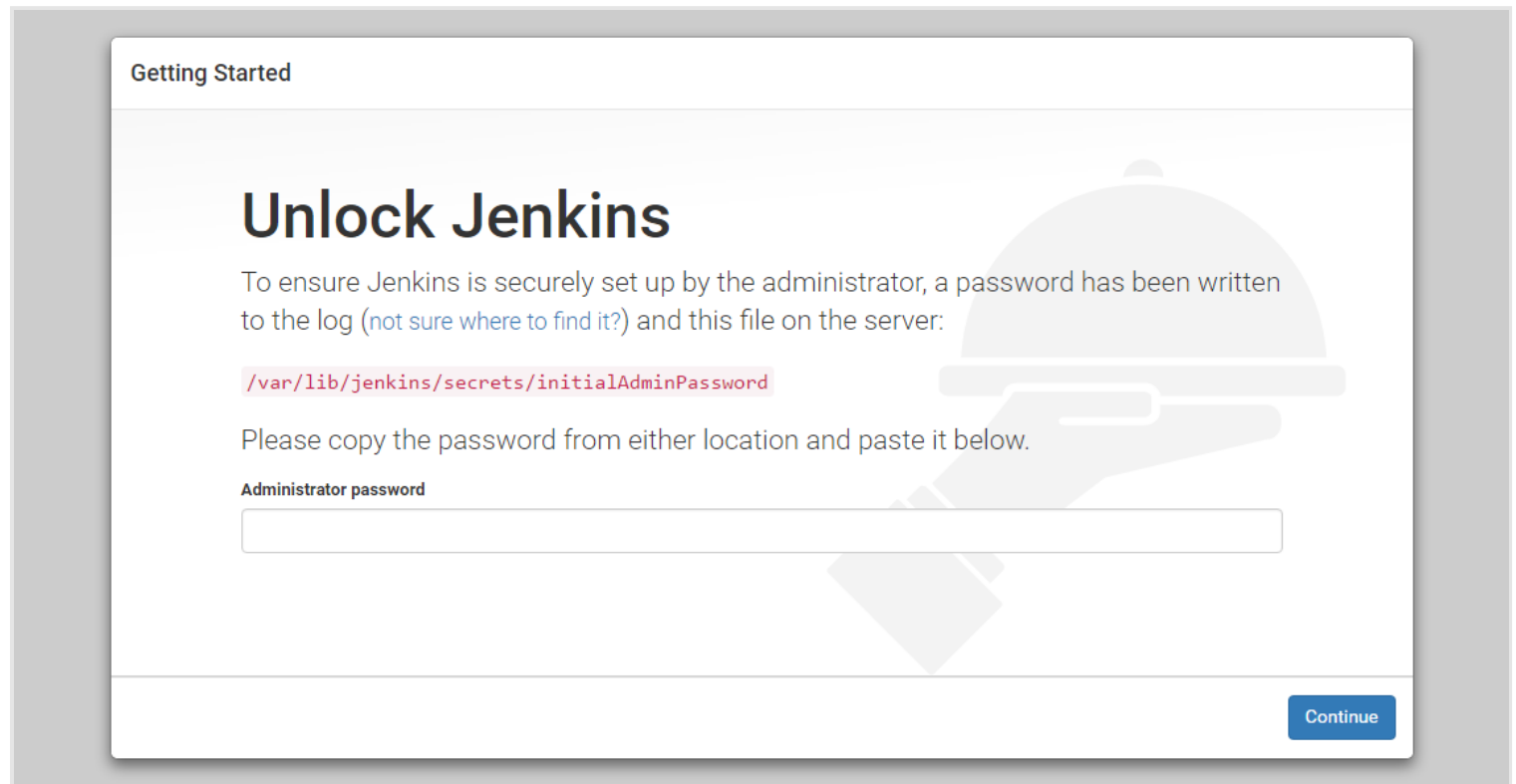
In the terminal window, use the `cat` command to display the password:

```
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Copy the 32-character alphanumeric password from the terminal and paste it into the **Administrator password** field, then click **Continue**.

The next screen presents the option of installing suggested plugins or selecting specific plugins:

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins	Select plugins to install
Install plugins the Jenkins community finds most useful.	Select and install plugins most suitable for your needs.

We'll click the **Install suggested plugins** option, which will immediately begin the installation process:

Getting Started

Getting Started

✔ Folders	✔ OWASP Markup Formatter	✔ Build Timeout	✔ Credentials Binding	** Pipeline: Milestone Step ** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI)
✔ Timestamper	✔ Workspace Cleanup	✔ Ant	✔ Gradle	** Jackson 2 API ** JavaScript GUI Lib: ACE Editor bundle
↻ Pipeline	↻ GitHub Branch Source	↻ Pipeline: GitHub Groovy Libraries	✔ Pipeline: Stage View	** Pipeline: SCM Step ** Pipeline: Groovy ** Pipeline: Input Step
↻ Git	↻ Subversion	↻ SSH Slaves	↻ Matrix Authorization Strategy	** Pipeline: Stage Step ** Pipeline: Job ** Pipeline Graph Analysis ** Pipeline: REST API
↻ PAM Authentication	↻ LDAP	↻ Email Extension	↻ Mailer	** JavaScript GUI Lib: Handlebars bundle ** JavaScript GUI Lib: Moment.js bundle **Pipeline: Stage View** ** Pipeline: Build Step ** Pipeline: Model API ** Pipeline: Declarative Extension Points API ** Apache HttpComponents Client 4.x API ** JSch dependency

When the installation is complete, you will be prompted to set up the first administrative user. It's possible to
skip this step and continue as `admin` using the initial password we used above, but we'll

SCROLL TO TOP

create the user.

Note: The default Jenkins server is NOT encrypted, so the data submitted with this form is not protected. When you're ready to use this installation, follow the guide How to Configure Jenkins with SSL Using an Nginx Reverse Proxy on Ubuntu 18.04. This will protect user credentials and information about builds that are transmitted via the web interface.

Enter the name and password for your user:

Create First Admin User

Username: sammy

Password: •••••••••

Confirm password: •••••••••

Full name: Sammy the Shark

E-mail address: sammy@example.com

Jenkins 2.121.1   Continue as admin   Save and Continue

You will see an **Instance Configuration** page that will ask you to confirm the preferred URL for your Jenkins instance. Confirm either the domain name for your server or your server's IP address:

Instance Configuration

Jenkins URL:

http://203.0.113.0:8080

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.
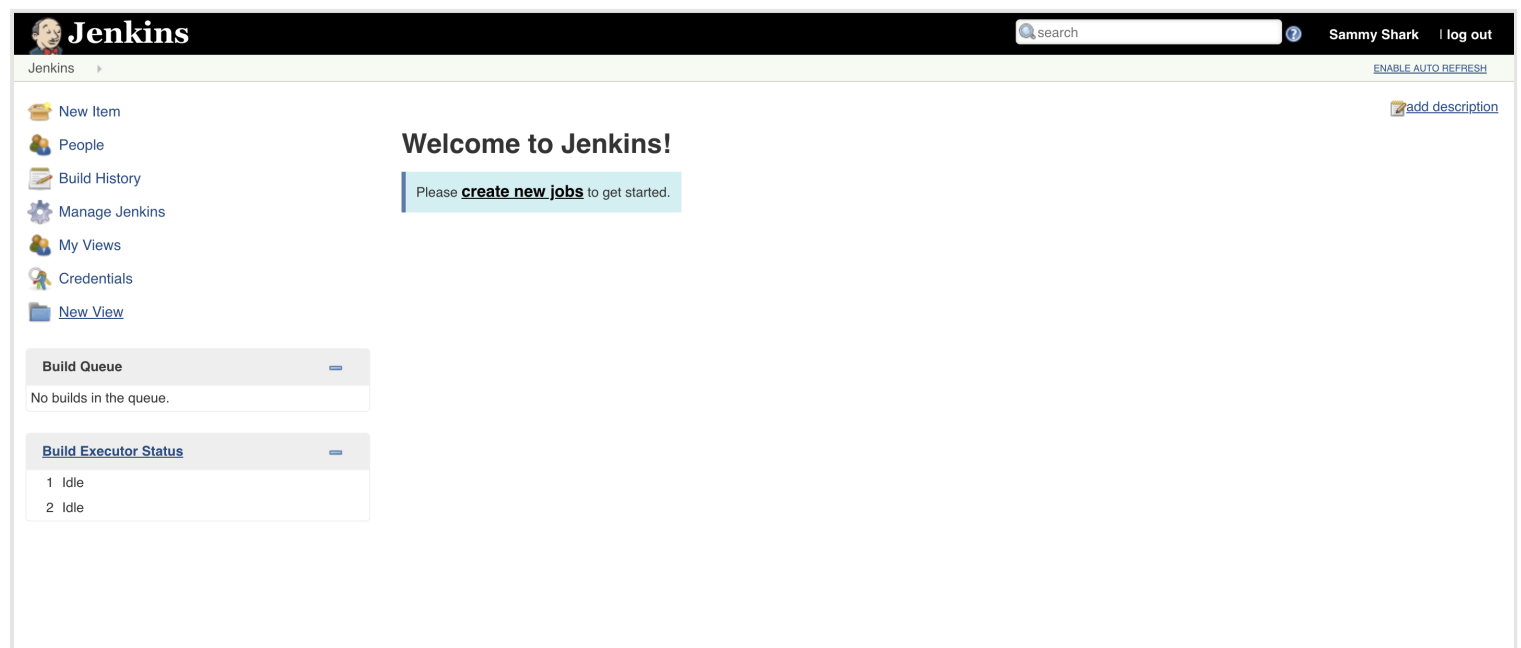
Jenkins 2.121.1

Not now   **Save and Finish**

After confirming the appropriate information, click **Save and Finish**. You will see a confirmation page confirming that **"Jenkins is Ready!"**:

SCROLL TO TOP

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Click **Start using Jenkins** to visit the main Jenkins dashboard:

At this point, you have completed a successful installation of Jenkins.

Conclusion

In this tutorial, you have installed Jenkins using the project-provided packages, started the server, opened the firewall, and created an administrative user. At this point, you can start exploring Jenkins.

When you've completed your exploration, if you decide to continue using Jenkins, follow Configure Jenkins with SSL Using an Nginx Reverse Proxy on Ubuntu 18.04 to protect your passwords, as

well as any sensitive system or product information that will be sent between your machine and the server in plain text.

By: Melissa Anderson   By: Kathleen Juell

♡ Upvote (2)   ⊡ Subscribe   ⬆ Share

We just made it easier for you to deploy faster.

TRY FREE

Related Tutorials

How To Configure OpenLDAP and Perform Administrative LDAP Tasks

How To Change Account Passwords on an OpenLDAP Server

How To Use LDIF Files to Make Changes to an OpenLDAP System

How To Manage and Use LDAP Servers with OpenLDAP Utilities

How To Use Systemctl to Manage Systemd Services and Units

2 Comments

Leave a comment...

SCROLL TO TOP

Log In to Comment

x41102 *September 6, 2018*

0

Hi,

Thanks for this. Ran into an error after installation though, jenkins failed to start:

"""

ERROR: No Java executable found in current PATH: /bin:/usr/bin:/sbin:/usr/sbin

If you actually have java installed on the system make sure the executable is in the aforementioned

"""

Had to install Java(8) and it was all good after that. Read somewhere else that 18.04 defaults to Java 9 which Jenkins "doesn't want".

---

digital40de01d99ecb8cba93c *November 10, 2018*

0

Thank you, worked like a charm. Concise and easy to read.

Community   Tutorials   Questions   Projects   Tags   Newsletter   RSS 🔊

Distros & One-Click Apps   Terms, Privacy, & Copyright   Security   Report a Bug   Write for DOnations   Shop

SCROLL TO TOP