**How To Install WildFly on Ubuntu 20.04 lts**

WildFly, initially known as **JBoss Application Server** is an exceptionally fast and lightweight open-source implementation of the JEE (Java Enterprise Edition) container. WildFly is written in Java was developed by RedHat to help manage application runtime and build apps. This tool was developed in 2006 by JBoss Inc and later acquired by RedHat to become the WildFly. The name was changed due to the confusion brought in the portfolio by the product names. WildFly can be run on several Operating systems with support for both 32 and 64-bit.

Some of the key features of Wildfly are:

* **Incredible Web Performance and Scalability**
* **Efficient Memory Management**
* **Slim and Customizable Runtime**
* **Powerful Server Administration**
* **Unified Configuration & Management**
* **Domain & Standalone Operational Modes**
* **Concurrent and Fast Classloading**
* **Full Java EE 8 Support**

# Step 1 — Install JDK

**Wildfly** requires Java 8 or later versions to work. You can check and verify that Java is installed with the following command.

$ java -version

If java is not installed, you will see **“java: command not found”**. Run below commands to install Java.

$ sudo apt-get update$ sudo apt-get install default-jdk -y

After installation, check if java is installed correctly by executing below command

$ java -version

# Step 2 — Download and Extract Wildfly Server

Check Wildlfy [downloads](https://wildfly.org/downloads/) page for latest releases before downloading. For this tutorial, we will download **Wildfly 16.0.0.Final (Java EE Full & Web Distribution).**

We are going to install Wildfly to **/opt/** directory, so we will download the Wildfly package to that location.

Change directory to **/opt/** and download Wildfly to that directory.

$ cd /opt

$ sudo wget <https://download.jboss.org/wildfly/16.0.0.Final/wildfly-16.0.0.Final.tar.gz>

Extract the tar package and rename the extracted directory to **wildfly.** This will be Wildfly’s installation directory

$ sudo tar -xvzf wildfly-16.0.0.Final.tar.gz

$ sudo mv wildfly-16.0.0.Final /opt/wildfly

# Step 3 — Create User and Group for Wildfly

We should not run Wildfly under the root user for security reasons. Let’s create a group **wildfly** and add a user **wildfly** to it.

Additionally, the home directory of **wildfly** user will be the Wildfly’s installation directory i.e. **/opt/wildfly/**.

$ sudo groupadd wildfly

$ sudo useradd -r -g wildfly -d /opt/wildfly -s /sbin/nologin wildfly

# Step 4 — Change Permission and Ownership of the Wildfly Installation Directory

Next, we will modify ownership and permission of **/opt/wildfly/** directory. We will also give executable permissions to /**opt/wildfly/bin/** directory. While under **/opt/** directory, run the following commands:

$ sudo chown -R wildfly: wildfly

$ sudo chmod o+x /opt/wildfly/bin/

# Step 5 — Creating a SystemD Service File for Wildfly

Create a configuration directory for Wildfly under **/etc/** directory by the name **wildfly.**

$ cd /etc/

$ sudo mkdir wildfly

Copy Wildfly configuration file **/opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf** to **/etc/wildfly/** directory

$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/

Next, copy Wildfly launch script (**launch.sh**) under **/opt/wildfly/docs/contrib/scripts/systemd/** to **/opt/wildfly/bin/** directory

$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/

We need to make **wildfly** user as the owner of this script so that it can execute it:

$ sudo chown wildfly: /opt/wildfly/bin/launch.sh

Now, copy service definition file (**wildfly.service**) under **/opt/wildfly/docs/contrib/scripts/systemd/** to **/etc/systemd/system/** directory

$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/

Open **wildfly.service** in an editor

$ sudo nano /etc/systemd/system/wildfly.service

Make the changes marked as bold or you can simply copy/paste the below content as it is.

[Unit]Description=The WildFly Application Server  
After=syslog.target network.target  
Before=httpd.service[Service]Environment=LAUNCH\_JBOSS\_IN\_BACKGROUND=1  
EnvironmentFile=-/etc/wildfly/wildfly.conf  
User=wildfly  
**Group=wildfly**  
LimitNOFILE=102642  
PIDFile=/var/run/wildfly/wildfly.pid  
ExecStart=/opt/wildfly/bin/launch.sh $WILDFLY\_MODE $WILDFLY\_CONFIG $WILDFLY\_BIND  
StandardOutput=null[Install]WantedBy=multi-user.target

Save and exit the file.

Reload **systemd** manager configuration and enable **wildfly** service on system startup

$ sudo systemctl daemon-reload

$ sudo systemctl enable wildfly

To start **wildfly** system service:

$ sudo systemctl start wildfly

Once the service is started, we can check the status by running below command:

$ sudo systemctl status wildfly

Now access Wildfly server at:

[http://**<instance-public-ip>**:8080/](http://18.218.71.21:8080/auth/)

# Step 8 — Configure Wildfly Management Console

The Wildfly management console allows us to manage different aspects of the Wildfly server. e.g. configuring subsystems, server monitoring, managing deployments or access control.

By default, the management console is not made accessible remotely. To make it accessible remotely, we have to make small changes in 3 files. So let’s start

Open **wildfly.conf** file under **/etc/wildfly/** directory

$ sudo nano /etc/wildfly/wildfly.conf

Add a line at the end as shown below:

$WILDFLY\_MANAGEMENT\_CONSOLE\_BIND

Save and exit the file.

Now open **launch.sh** in **/opt/wildfly/bin/** directory and change its contents as shown below:

$ sudo nano /opt/wildfly/bin/launch.sh

Add these line

Save and exit the file

Finally, open Wildfly’s system service definition file (**wildfly.service**) under **/etc/systemd/system/** and make the changes as shown below:

$ sudo nano /etc/systemd/system/wildfly.service

$WILDFLY\_MANAGEMENT\_CONSOLE\_BIND

Save and exit the file

Since we have changed the service unit file, we have to inform the **systemd manager**

$ sudo systemctl daemon-reload

Now restart the **wildfly** service

$ sudo systemctl restart wildfly

Once restarted, Access the Wildfly management console at:

[**http://<instance-public-ip>:9990**](http://18.218.71.21:9990/)

We can successfully access the management console but as shown above, we need a management user to login.

We can use the **add-user.sh** script, packaged with Wildfly server distribution, to create a management user. Run the script with below command:

$ sudo /opt/wildfly/bin/add-user.sh

Once prompted, select to add a management user and provide your desired **username** and **password.**

In the last prompt, for enabling remote access for this user write **yes** or **y**.

After providing the required information, the script will verify the user creation as shown above.

We need to restart Wildfly server so our user is picked up during boot.

$ sudo systemctl restart wildfly

Now if we access the management console again, it will prompt for HTTP basic authorization.

Provide the management user credentials that we created above and click ‘ok’:

Once successfully logged in, we land inside the management console

Now we will deploy a sample Java application on the WildFly server. First, download the Helloworld Java Application from [GitHub](https://github.com/aeimer/java-example-helloworld-war) as below.

cd /opt/wildfly/standalone/deployments/

sudo curl -O <https://raw.githubusercontent.com/aeimer/java-example-helloworld-war/master/dist/helloworld.war>

View the files available.

ls

helloworld.war helloworld.war.deployed README.txt

From the output above, the file named **\*.deployed** shows that the application has been deployed successfully, otherwise the file will be named **\*.failed**. You can WildFly view logs as below.

The deployment can be viewed on the browser using the URL <http://IP_address:8080/helloworld/>