



How to Install WildFly (JBoss) on CentOS 7

Posted Mar 1, 2019 • 7 min read



WildFly, formerly known as JBoss is a cross-platform open source application runtime written in Java that helps you build amazing applications. WildFly is flexible, lightweight, and it is based on pluggable subsystems that can be added or removed as needed.

This tutorial explains how to install WildFly application server on CentOS 7.

Prerequisites

You'll need to be logged in as a user with sudo access to be able to install packages on your CentOS system.



WildFly 9 requires Java SE 8 or later. In this tutorial we will install OpenJDK, the open source implementation of the Java Platform which is the default Java development and runtime in CentOS 7.

Install the OpenJDK package by running:

```
$ sudo yum install java-1.8.0-openjdk-devel
```

Step 2: Create a User

Running WildFly as the root user is a security risk and not considered best practice.

To create a new system user and group named `wildfly` with home directory `/opt/wildfly` run:

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

Step 3: Install WildFly

At the time of writing, the latest version of WildFly is `16.0.0`. Before continuing with the next step you should check the [download page](#) for a new version. If there is a new version replace the `WILDFLY_VERSION` variable in the command below.

Download the WildFly archive in the `/tmp` directory using the following `wget` command:



When the download is completed, unpack the tar.gz file and move it to the `/opt` directory:

```
$ sudo tar xf /tmp/wildfly-$WILDFLY_VERSION.tar.gz -C /opt/
```

Next, create a symbolic link `wildfly` that will point to the WildFly installation directory:

```
$ sudo ln -s /opt/wildfly-$WILDFLY_VERSION /opt/wildfly
```

WildFly will run under the `wildfly` user which needs to have access to the WildFly installation directory.

Change the directory ownership to user and group `wildfly` with the following chown command:

```
$ sudo chown -RH wildfly: /opt/wildfly
```

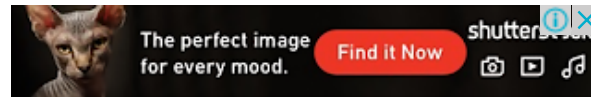
Step 4: Configure Systemd

The WildFly package includes files necessary to run WildFly as a service.

Start by creating a directory which will hold the WildFly configuration file:

```
$ sudo mkdir -p /etc/wildfly
```

Copy the configuration file to the `/etc/wildfly` directory:



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

This file allows you to specify the WildFly mode and bind address. By default, WildFly will run in a standalone mode and will listen on all interfaces. You can edit the file according to your needs.

`/etc/wildfly/wildfly.conf`

```
# The configuration you want to run
WILDFLY_CONFIG=standalone.xml

# The mode you want to run
WILDFLY_MODE=standalone

# The address to bind to
WILDFLY_BIND=0.0.0.0
```

Next copy the WildFly `launch.sh` script to the `/opt/wildfly/bin/` directory:

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

The scripts inside `bin` directory must have executable flag:

```
$ sudo sh -c 'chmod +x /opt/wildfly/bin/*.sh'
```

Copy the systemd unit file named to the `/etc/systemd/system/` directory:



Notify systemd that we created a new unit file:

```
$ sudo systemctl daemon-reload
```

Start the WildFly service and enable it to be automatically started at boot time by running:

```
$ sudo systemctl start wildfly  
$ sudo systemctl enable wildfly
```

Verify that the service is running:

```
$ sudo systemctl status wildfly
```



```
Loaded: loaded (/etc/systemd/system/wildfly.service; disabled; vendor preset: disabled)
Active: active (running) since Fri 2019-03-01 20:13:39 UTC; 3s ago
Main PID: 3680 (launch.sh)
CGroup: /system.slice/wildfly.service
```

Step 5: Adjust the Firewall

If your server is protected by a firewall and you want to access the WildFly instance from the outside of the local network you also need to open port 8080.

Use the following commands to open the necessary port:

```
$ sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp
$ sudo firewall-cmd --reload
```

When running a WildFly application in a production environment most likely you will have a load balancer or reverse proxy and it's a best practice to restrict access to port 8080 only to your internal network.

Step 6: Configure WildFly Authentication

Now that WildFly is installed and running the next step is to create a user who will be able to connect using the administration console or remotely using the CLI.

To add a new user use the `add-user.sh` script that is located in the WildFly's bin directory:

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



Output

```
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
```

Select a and hit Enter :

Next, the script will prompt you to enter the details of the new user:

Output

```
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : linuxize
Password recommendations are listed below. To modify these restrictions edit the add-user
- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, admini
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or
About to add user 'linuxize' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'linuxize' to file '/opt/wildfly-16.0.0.Final/standalone/configuration/mgmt-us
Added user 'linuxize' to file '/opt/wildfly-16.0.0.Final/domain/configuration/mgmt-users
Added user 'linuxize' with groups to file '/opt/wildfly-16.0.0.Final/standalone/configur
Added user 'linuxize' with groups to file '/opt/wildfly-16.0.0.Final/domain/configuratic
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection fo
yes/no? yes
To represent the user add the following to the server-identities definition <secret value
```

The new user will be added to the properties files used for authentication.

Step 6: Test the WildFly Installation

To access the default WildFly page, open your browser and type:

```
http://<your_domain_or_IP_address>:8080
```

Assuming the installation is successful, a screen similar to the following will appear:



Welcome to WildFly

Your WildFly instance is running.

[Documentation](#) | [Quickstarts](#) | [Administration Console](#)

[WildFly Project](#) | [User Forum](#) | [Report an issue](#)

JBoss Community

To replace this page simply deploy your own war with / as its context path.
To disable it, remove the "welcome-content" handler for location / in the undertow subsystem.

Step 7: Access WildFly Administration Console

Command-Line Interface

To access the WildFly Administration Console from the command line you can use the `jboss-cli.sh` script.

Navigate to the WildFly's bin directory and run the script with the `--connect` option:

```
$ cd /opt/wildfly/bin/  
$ ./jboss-cli.sh --connect
```

You will be asked to enter your administrative username and password (created in step 6):

Output

```
Authenticating against security realm: ManagementRealm  
Username: linuxize  
Password:
```




From here you can deploy and undeploy your applications, manage users and groups and configure and monitor WildFly server.

Web Interface

If you prefer to manage your server from the GUI, WildFly also provides a web-based console.

By default, the WildFly administration console is available only from localhost at `http://localhost:9990/console`. Sign in using the user you have created in step 6.

WildFly

Homepage Deployments Configuration Runtime Patching Access Control

WildFly Application Server

Deployments

Add and manage deployments

Deploy an application to the server

1. Use the 'Add Deployment' wizard to deploy the application
2. Enable the deployment

Configuration

Configure subsystem settings

Define a datasource to be used by deployed applications. The proper JDBC driver must be deployed and registered.

1. Select the Datasources subsystem
2. Add a Non-XA or XA datasource
3. Use the 'Create Datasource' wizard to configure the datasource settings

Runtime

Monitor server status

View runtime information such as server status, JVM status, and server log files.

1. Select the server
2. View log files or JVM usage

Access Control

Manage user and group permissions for management operations

Assign roles to users or groups to determine access to system resources.

1. Add a new user or group
2. Assign one or more roles to that user or group

Patching

Manage WildFly Full patches

Apply a WildFly Full patch to the server.

1. Download the patch file to the local machine
2. Use the 'Apply Patch' wizard to select and apply the patch

Need Help?

General Resources

- [WildFly Home](#)
- [WildFly Documentation](#)
- [Admin Guide](#)
- [Model Reference Documentation](#)
- [Browse Issues](#)
- [Latest News](#)

Get Help

- [Access tutorials and quickstarts](#)
- [User Forums](#)
- [IRC](#)
- [Developers Mailing List](#)

3.0.6.Final Tools Settings

If you want to access the console from remote locations you'll need to make small modifications to the `wildfly.service`, `wildfly.conf` and `launch.sh` files.

Open the `wildfly.conf` and append `WILDFLY_CONSOLE_BIND=0.0.0.0` at the end of the file.



```
# The configuration you want to run
WILDFLY_CONFIG=standalone.xml

# The mode you want to run
WILDFLY_MODE=standalone

# The address to bind to
WILDFLY_BIND=0.0.0.0

# The address console to bind to
WILDFLY_CONSOLE_BIND=0.0.0.0
```

Open the `launch.sh` and edit the highlighted lines:

```
/opt/wildfly/bin/launch.sh
```

```
#!/bin/bash

if [ "x$WILDFLY_HOME" = "x" ]; then
    WILDFLY_HOME="/opt/wildfly"
fi

if [[ "$1" == "domain" ]]; then
    $WILDFLY_HOME/bin/domain.sh -c $2 -b $3 -bmanagement $4
else
    $WILDFLY_HOME/bin/standalone.sh -c $2 -b $3 -bmanagement $4
fi
```

Restart the service for changes to take effect:

```
$ sudo systemctl restart wildfly
```

Open the `wildfly.service` and edit the highlighted lines:

```
/etc/systemd/system/wildfly.service
```

```
[Unit]
Description=The WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
```



```
LimitNOFILE=102642
PIDFile=/var/run/wildfly/wildfly.pid
ExecStart=/opt/wildfly/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG $WILDFLY_BIND $WILDFLY
StandardOutput=null
```

[Install]

```
WantedBy=multi-user.target
```

Create the `/var/run/wildfly` directory and set correct permissions:

```
$ sudo mkdir /var/run/wildfly/
$ sudo chown wildfly: /var/run/wildfly/
```

Notify systemd that the unit file was changed:

```
$ sudo systemctl daemon-reload
```

Restart the WildFly service by executing:

```
$ sudo systemctl restart wildfly
```

Assuming that port `9990` is not blocked in your firewall, you should be able to access the WildFly administration console at `http://<your_domain_or_IP_address>:9990/console`.

Conclusion

You have successfully installed WildFly on your CentOS 7 system. You can now visit the official [WildFly Documentation](#) and learn more about the WildFly features.

If you hit a problem or have feedback, leave a comment below.

java wildfly centos



If you like our content, please consider buying us a coffee.
Thank you for your support!



BUY ME A COFFEE

Sign up to our newsletter and get our latest tutorials and news straight
to your mailbox.

Your email...

Subscribe

We'll never share your email address or spam you.

OCT 1, 2019

How to Install Java on CentOS 8





MAR 29, 2019

How to Install Elasticsearch on CentOS 7



MAR 25, 2019

How to Install Apache Cassandra on CentOS 7





Show comments (2)

© 2019 Linuxize.com

[Privacy Policy](#) [Contact](#)

