

KNIME Server Installation Guide

KNIME AG, Zurich, Switzerland
Version 4.11 (last updated on 2020-07-27)



Table of Contents

Introduction.....	1
Prerequisites.....	2
Server	2
Client.....	3
The installation owner.....	4
Installation.....	5
Summary	5
Launch the installer.....	5
Path to the system's JDK location	6
Path to the installation folder.....	6
KNIME Server Executor configuration.....	6
KNIME Server Workflow Repository	7
KNIME Server license file	8
KNIME Server port configuration.....	8
KNIME Server mountpoint and Context Root.....	8
KNIME Server admin user configuration	8
KNIME Server email resource	9
Installing and launching KNIME Server, and final configuration steps	9
Autoinstall file	9
Additional installer command-line flags.....	10
Starting/stopping KNIME Server	10
Automatically starting KNIME Server.....	12
Defining a Linux runlevel script	12
Creating a Windows service.....	14
Running Executors as services.....	14
Linux with systemd	15
Windows.....	15

Introduction

KNIME Server is an enterprise-grade solution for Advanced Analytics workloads such as sharing workflows, and executing workflows. The Server is based on the Tomcat application server, and uses a core of KNIME Analytics Platform in order to execute workflows. The KNIME Server installer can install both of these components that make up KNIME Server. This document aims to give a quick overview of the steps needed to perform the installation, and a short description of the options that can be changed at install time. After installation, you can refer to the [KNIME Server Administration Guide](#) for detailed information about the Server architecture, configuration options and general administration tasks. More details about the functionality of KNIME Server are listed in the following paragraphs.

KNIME Server provides KNIME software users with a shared remote workflow repository. Workflows can be uploaded to and stored on the server or retrieved from it to a local KNIME workspace. Workflows on the server can be executed as jobs either as needed or as part of a one time or periodically scheduled event. Additionally, workflows may be executed through a web browser using the KNIME WebPortal or via RESTful web service interfaces which allow control from 3rd party applications. The KNIME Explorer view, which is built into KNIME Analytics Platform can be used to log into remote repositories, control user access permissions and schedule workflow execution tasks.

KNIME Server consists of two components: a server side component and a client interface. The server side component is installed on an application server (Tomcat) together with a KNIME Analytics Platform installation which will be responsible for executing workflows. The latter is referred to as "Executor". The client interface allows interaction with the server repository and is integrated in KNIME Analytics Platform (via the KNIME ServerSpace extension), and KNIME WebPortal, or accessible via the REST API.

Application server and Executor can either be installed on the same host (for KNIME Server Small and KNIME Server Medium), or on separate hosts (KNIME Server Large). Beginning with KNIME Server 4.11, we use an integrated message broker for communication between application server and Executor, based on Apache Qpid (<https://qpid.apache.org/>). This can be used for running either KNIME Server Small or KNIME Server Medium. It can also be used for KNIME Server Large in case you plan to only use a single Executor running on the same host as the application server. Apache Qpid is bundled as part of all KNIME Server installations, so no additional setup is necessary.

If you want to set up distributed KNIME Executors for KNIME Server Large, please follow all steps in this document related to the main installation, and then follow the additional steps outlined [here](#). This explains how to setup RabbitMQ (which is used as message broker instead of Qpid), as well as how to add additional Executors.

Prerequisites

Server

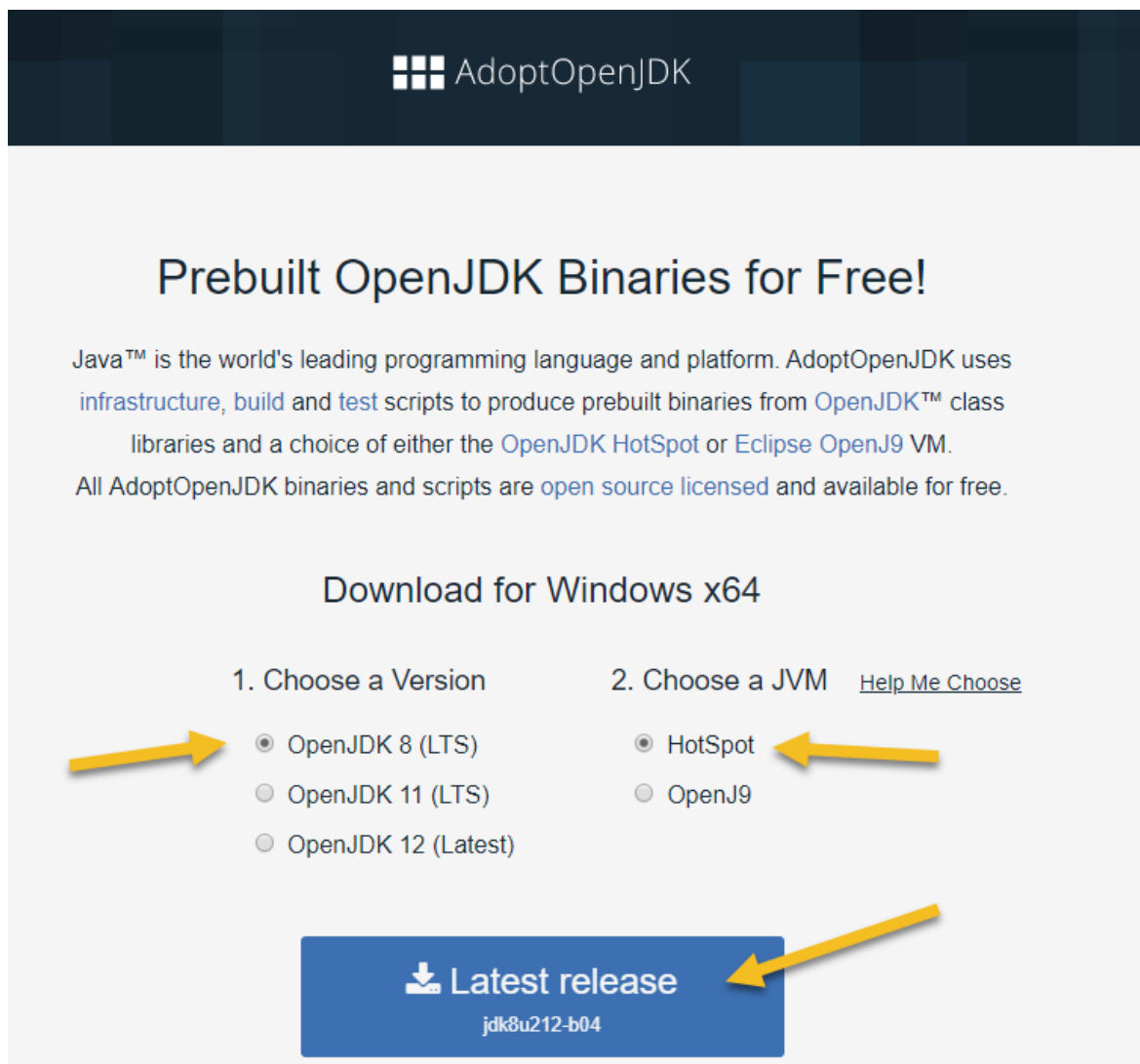
- The operating system on the server can be either Windows or Linux. The following operating systems are officially supported:

- Windows Server 2016, 2019
- Ubuntu 16.04 LTS, and 18.04 LTS and derivatives
- RHEL/CentOS 7.x and 8



At the time of this release, the Tableau Hyper format will not work on RHEL/CentOS 8 systems, and requires a fix on the Tableau-side. If you plan to use Tableau Hyper from KNIME Server, please use an older version of RHEL/CentOS until this is resolved.

- The server may also run on other operating systems, however, we neither test nor support them through the KNIME Server standard support package.
 - We strongly recommend a 64bit operating system with at least 32GB RAM and 8 CPU cores.
- Java Development Kit (JDK) 8, Hotspot, 64bit
 - Ubuntu 16.04 / 18.04: `sudo apt-get install -y openjdk-8-jdk`
 - RHEL/CentOS 6.7+ & 7: install `java-1.8.0-openjdk-devel` via the package manager
 - For all other OS please download and install the latest Hotspot OpenJDK 8 (LTS) from [the AdoptOpenJDK Website](#)



JDK ≥ 9 and OpenJ9 JVM is not supported at this time.

- Local hard disk for workflow repository partition (SSD class – or equivalent)
- 250GB disk space on the workflow repository partition

For generation of images using the Chromium extension the following additional extensions are required under Linux:

- Ubuntu: `sudo apt install -y libgtk-3-0`
- RHEL 7/CentOS: `sudo yum install libXScrnSaver`

Client

- Operating system Windows, Linux, or macOS (independent from the server OS)

The installation owner

We recommend that both the Tomcat server as well as the KNIME Server Executor are installed by the same user. This user does not need administration privileges. The recommended user name is `knime`. No other user requires write access to any files created during the installation.



This user must perform the entire KNIME Server Executor installation including unpacking of KNIME Analytics Platform and installation of all desired additional features. If additional features are added later with a user that cannot write to the installation folder, the installation may appear to work normally, but causes problems some time later which can be difficult to track down.

Installation

It is recommended that the entire installation process is performed by the same user, e.g. `knime` (see [The installation owner](#) above). Don't install the server as Administrator or root user! Follow the installation instructions step by step. Don't skip anything unless it is marked as optional.

The KNIME Server installer is a single JAR file that works on all supported platforms (Windows, Linux, macOS). Use of the JAR file is described in the following sections.



Do **not** install any KNIME Server components as Administrator or root user! This is also true for any **future changes** to the installation, such as upgrades or installing extensions.

Summary

We briefly describe the installation options available in this guide. For full details about any of the options see the [KNIME Server Administration Guide](#).

Before you start, make sure that the Java 8 JDK is installed (see [Prerequisites](#)).



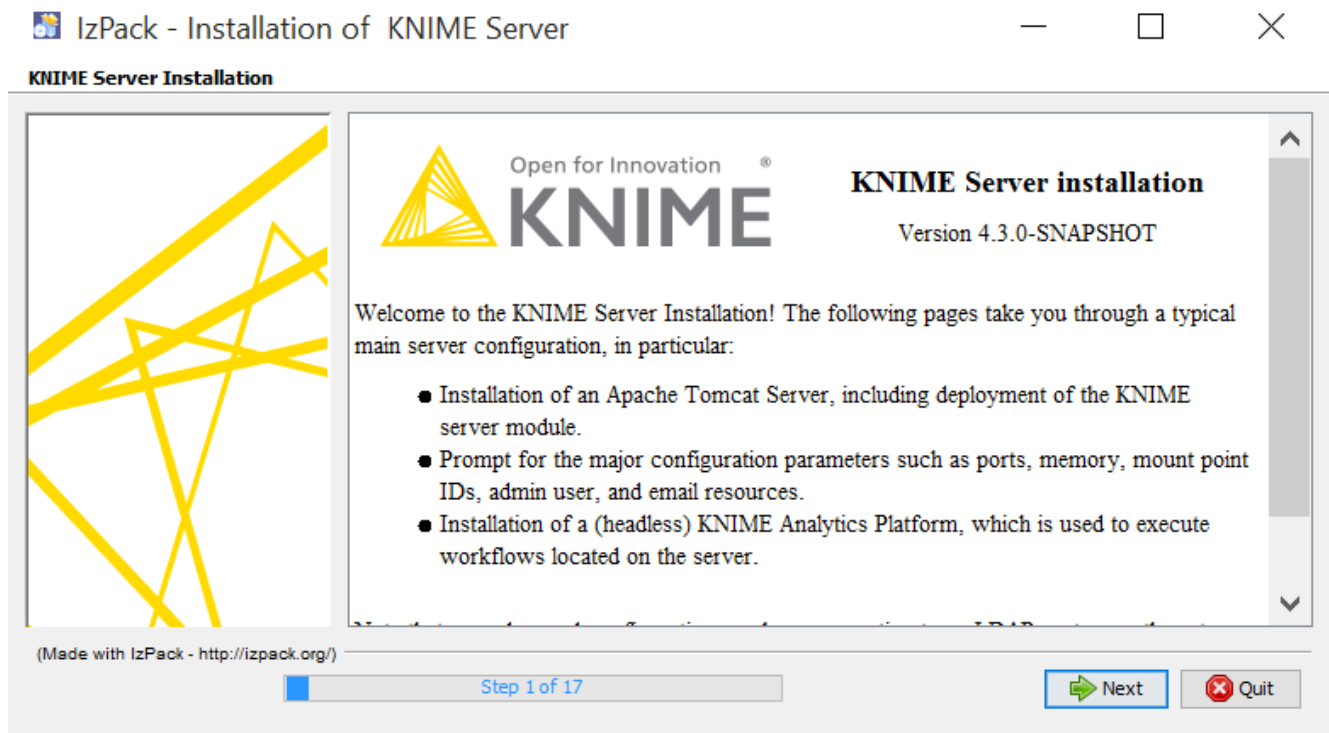
In case you are installing KNIME Server on a machine which already has RabbitMQ installed, please stop the RabbitMQ service first before starting the installation. Otherwise, the embedded message broker Apache Qpid will try to use the same ports as RabbitMQ, which prevents a successful launch of KNIME Server on the initial startup. If you plan to use KNIME Server with multiple KNIME Executors via RabbitMQ, please set the corresponding path to RabbitMQ in `knime-server.config` after finishing the installation, and disable the embedded broker (`com.knime.enterprise.executor.embedded-broker=false`). At this point, you can also restart the RabbitMQ service.

Launch the installer

The installer is called `knime-server-installer-4.x.y.jar` (where `x.y` is the current release number). It can be downloaded from [here](#). The installer can be run by issuing the command (at the Linux terminal, or `cmd.exe` prompt):

```
java -jar knime-server-installer-4.x.y.jar
```

Issuing the command will launch the installer GUI:



It is also possible to step through the installation via the command line. In this case issue the command:

```
java -jar knime-server-installer-4.x.y.jar -console
```

Path to the system's JDK location

The Java JDK version must be 1.8.x, and you should use the browse button to select the folder that contains the bin folder.

Path to the installation folder

The installation folder will (by default) contain all of the files that KNIME requires, which includes Tomcat, the KNIME Server Executor and the KNIME Server repository. The installation folder should be empty, and if it does not exist will be created. If prompted choose OK.

KNIME Server Executor configuration

There are three options to choose from:

1. Select existing KNIME Server Executor (default): This means that you need to have previously installed a compatible KNIME Server Executor using one of the downloads from the KNIME Website. Note that the downloadable **KNIME Server Executor build** contains the required extension, but a standard **KNIME Analytics Platform build** will not be sufficient. In the latter case the "KNIME Executor Connector" (in category "KNIME Server Executor (server-side extension)") must be installed. The extension is available from the pre-registered KNIME Analytics Platform update site. To install it run the KNIME Analytics Platform instance that is intended to be the Executor and navigate to *File → Install KNIME Extensions...* and select *KNIME Executor connector* under *KNIME Server Executor (server-side extension)*. Finally, click on "next" and follow the instructions.
2. Install KNIME Server Executor: This will install a new KNIME Server Executor, and will result in several additional steps that must be followed. This option requires internet access. In case you can't connect to the internet from the server machine, please download a KNIME Executor before starting the installation and then choose option 1 on the current installation step.
3. Disable server-side workflow execution: This option means that workflows cannot be executed on KNIME Server.

Selecting option 1 or 2 will automatically add additional settings to the Executor's `knime.ini` file that are needed to establish a connection to the application server, and that can be used to manage the Executor's preference setting.



Changes to the `knime.ini` file will **not** be made if you point the installer to an existing workflow repository in the next step. This is done to not contradict user-defined settings in `knime-server.config`.

KNIME Server Workflow Repository

KNIME Server stores all workflows and related configuration information in the KNIME Server Workflow Repository. The folder should exist on a local drive and have enough free space to store the workflows and jobs. We recommend a minimum of 100 GB, although this may need to be increased considerably depending on the intended use case.

By default, the KNIME Server Workflow Repository is created within the KNIME Server installation folder. It is possible to use a different location (e.g. on another hard drive) or an existing folder. In case the folder exists, the installer will not modify the folder, assuming that you pointed it to the workflow repository of a previous installation.

KNIME Server license file

The KNIME Server application works only as expected if a valid license file is provided. If you have not received a license file, please contact KNIME (contact@knime.org), or get in touch with your local KNIME representative.

If you performed the installation without the license file, you can simply copy the KNIME Server license file into the `<knime-server-repository>/licenses` folder. The file name must end with `.xml`, and there is no need to restart the server after adding the license file.

KNIME Server port configuration

KNIME Server and its clients communicate by either HTTP, or HTTPS. Default ports are 8080, and 8443. These ports can be changed in the range: 1024-49151.

KNIME Server mountpoint and Context Root

The KNIME Server mountpoint is used to describe the KNIME Server to the clients that connect to it. For simple installations, the default name, or a simple name e.g. *mycompany-knime* is usually appropriate. For an environment with multiple KNIME Servers, mountpoints such as *knime-server-prod* or *knime-server-dev* would allow users to distinguish between development and production environments. For a full description of the mountpoint see the [KNIME Server Administration Guide](#).

The Context Root (CONTEXT) is used to define the KNIME WebPortal URL. All WebPortal functionality will be available at:

```
http://HOSTNAME:PORT/CONTEXT
```

KNIME Server admin user configuration

Many administrative tasks on KNIME Server can be performed by the administrator user, e.g. setting workflow/file permissions or adding/removing users. The default username is `knimeadmin`, and the password is set to the hostname of the machine. It is strongly recommended that you change this password. If using LDAP authentication, changes are required to the `server.xml` file in `<installation-path>/apache-tomcat*/conf`. Please refer to the [KNIME Server Advanced Setup Guide](#) for full information.

KNIME Server email resource

The SMTP server, SMTP user, and 'from'-address can be altered. For advanced configuration options consult the [KNIME Server Administration Guide](#).

Installing and launching KNIME Server, and final configuration steps

The final four steps will install, and if successful launch KNIME Server. Under Windows a new command prompt window will appear, closing this window will shutdown KNIME Server. The penultimate page displays information about the installation, such as the URLs required to connect to the server. The final page is able to generate an automatic installation script that can be used to automate future installations.

The Executor needs to be started separately with

```
<executor directory>/start-executor.bat (Windows)
```

or

```
<executor directory>/start-executor.sh (Linux)
```

In case the executor startup script is not present, you can start the Executor from command line by running

```
./knime -nosplash -consolelog -application  
com.knime.enterprise.slave.KNIME_REMOTE_APPLICATION
```



Please make sure to start the KNIME Executor after starting KNIME Server. Otherwise, the Executor will not be able to acquire the core tokens it needs to operate.



In case the installer did not add the path to Qpid to the Executor's knime.ini, please add the following line to the bottom of your knime.ini file:
`-Dcom.knime.enterprise.executor.msgq=amqp://knime:20knime16@localhost/`

Autoinstall file

Once successfully executed, the KNIME Server installer generates a file that can be saved. The xml file is called `auto-install.xml` and describes all parameters that were set during the

installation process. In case you want to perform an installation using the same parameters you can simply run:

```
java -jar knime-server-installer-4.x.y.jar <path-to>/auto-install.xml
```

Additional installer command-line flags

The installer reads out system properties specified on command-line, which are usually not required in a standard installation but can be useful for automated (update) scripts.

- `-Dknime.server.executor.memory=<value>` (<value> is for instance 4G or 2500M)
The value specified is added to the KNIME Server Executor's `knime.ini` file and specifies how much memory a KNIME Server Executor is allowed to use at most. If not given, a default is used, which is based on the system's available main memory.
- `-Dknime.server.nostart=true`
If specified, KNIME Server is not started after the installation.

Starting/stopping KNIME Server

KNIME Server can be started and stopped using scripts in the `<tomcat folder>/bin` folder. It is highly recommended that you follow the steps in the following section [Automatically starting KNIME Server](#), this ensures that KNIME Server and KNIME Executor services can be quickly and easily restarted.

For Linux/macOS restart with

```
<tomcat folder>/bin/shutdown.sh  
<tomcat folder>/bin/startup.sh
```

and for Windows use

```
<tomcat folder>/bin/shutdown.bat  
<tomcat directory>/bin/startup.bat
```

The Executor needs to be started with

```
<executor directory>/start-executor.bat (Windows)
```

or

```
<executor directory>/start-executor.sh (Linux)
```

Please make sure to adjust the permissions of the start-executor.sh file on Linux to also allow execution by the user running the Executor, e.g.

```
chmod 744 start-executor.sh
```

In case the executor startup script is not present, you can start the Executor from command line by running

```
./knime -nosplash -consolelog -application  
com.knime.enterprise.slave.KNIME_REMOTE_APPLICATION
```



Please make sure to start the KNIME Executor after starting KNIME Server. Otherwise, the Executor will not be able to acquire the core tokens it needs to operate.

After starting you should be able to access KNIME Server from both KNIME Analytics Platform and KNIME WebPortal (in your web browser). More details can be found in the [KNIME Explorer User Guide](#) and [KNIME WebPortal User Guide](#).

The URL entered in the KNIME Analytics Platform mount point dialog is the same as for the WebPortal and defaults to

```
http://<server>:8080/knime/
```

If you encounter problems please have a look at the log files (see section *Log Files* in the [KNIME Server Administration Guide](#) for details).

In some cases it may take quite some time (up to several minutes) until the server responds to requests on Linux systems. This is usually caused by insufficient entropy for the random number generator used by Tomcat. A description on how to remedy this issue is included in the [KNIME Server Administration Guide](#)

Automatically starting KNIME Server

In order to start KNIME Server after a reboot, you need to define a service (Windows) or a runlevel script (Linux) that starts Tomcat and brings it down at a machine shut down.

Defining a Linux runlevel script

Depending on your Linux distribution you can define a SysVinit start script or systemd service as follows (as `root` user):

Ubuntu >=16.04, Redhat 7.x, and derivatives (Systemd)

1. Copy the whole folder to the root of your file system.

```
sudo cp -r <install-dir>/install-data/linux-runlevel-templates/systemd/. /
```

The folder includes the systemd service description for `knime-server`, a start script, and a systemd override file that allows configuration of the service (such as file system location or the `userid` under which the server should run).

2. Run

```
sudo systemctl daemon-reload
```

3. Run

```
sudo systemctl edit knime-server.service
```

Adjust the settings in the editor that will open, and save the changes. Make sure that the `User` specified in this file exists on the system. Otherwise startup will fail.

4. Enable the service with

```
sudo systemctl enable knime-server.service
```

Ubuntu 14 and derivatives (SysVinit)

1. Copy the runlevel script

```
<install-folder>/install-data/linux-runlevel-  
templates/sysvinit/Ubuntu/etc/init.d/knime-server
```

or that is provided in the installation package to `/etc/init.d/knime-server` and make it executable.

2. Copy the configuration file

```
<install-folder>/install-data/linux-runlevel-  
templates/sysvinit/Ubuntu/etc/default/knime-server
```

from the installation package to `/etc/default/knime-server`. Adjust the variables in this config file.

3. Enable the start script with

```
update-rc.d knime-server defaults
```

Redhat 6.x and derivatives (SysVinit)

1. Copy the runlevel script

```
<install-folder>/install-data/linux-runlevel-  
templates/sysvinit/Redhat/etc/init.d/knime-server
```

or that is provided in the installation package to `/etc/init.d/knime-server` and make it executable.

2. Copy the configuration file

```
<install-folder>/install-data/linux-runlevel-  
templates/sysvinit/Redhat/etc/default/knime-server
```

from the installation package to `/etc/default/knime-server`. Adjust the variables in this config file.

3. Enable the start script with

```
chkconfig --add knime-server
```

Creating a Windows service

If you want to start and stop KNIME Server automatically under Windows, you need to create a Windows service.

To do so, first download the following three files and save them into <tomcat-folder>\bin:

[service.bat](#)

[tomcat9w.exe](#)

[tomcat.exe](#)

Open a command line window **as administrator**, navigate to <tomcat-folder>\bin and run the following command:

```
service.bat install
```

This will create a Windows service called "Apache Tomcat 9.0 Tomcat9" that can be started and stopped like any other Windows service (Control Panel ⇒ Administrative Tools ⇒ Services).

You may want to change the memory settings for when Tomcat is started as a service:

1. Start <tomcat-folder>\bin\tomcat9w.exe as Administrator.
2. Go to the "Java" tab and change the "Maximum Memory pool" value.

In order to uninstall the service, open a command line window **as administrator**, navigate to <tomcat-folder>\bin and run the following command:

```
service.bat uninstall
```

If you want to know more details, have a look at the [Tomcat documentation](#) which describes this process in detail.

Running Executors as services

It is also possible to run Executors as services that are automatically started during system startup (and stopped during shut down). This is the recommended method to use when not running on a docker deployment.



For increased security, we recommend to run both the Executor service by a different user than the Tomcat service.

Linux with systemd

Running Executors as services is only supported on Linux distributions that use *systemd* (e.g. Ubuntu >= 16.04, RHEL 7.x and derivatives). The following steps assume that you have a KNIME Executor installed that contains the *KNIME Executor connector* extension as described in the section [KNIME executor installation](#).

1. Copy the whole folder

```
<knime-installation>/systemd/
```

to the root of your file system. The folder includes the systemd service description for `knime-executor` and an override file that allows configuration of the service (such as file system location or the `userid` under which the Executor should run).

2. Run

```
systemctl daemon-reload
```

3. Run

```
systemctl edit knime-executor.service
```

Adjust the settings in the editor that opens, and save the changes. Make sure that the `User` specified in this file exists on the system. Otherwise startup will fail unless your version of *systemd* supports `DynamicUser`. In this case a temporary user account will be created.

4. Enable the service with

```
systemctl enable knime-executor.service
```

Windows

On Windows Executors can be run as Windows services by using *NSSM* (Non-Sucking Service Manager). The following steps assume that you have a KNIME Analytics Platform

installation that contains the *KNIME Executor connector* extension as described in [KNIME Server Installation Guide](#).

1. Edit

```
<knime-installation>/install-executor-as-service.bat
```

and adjust the variables at the top of the file to your needs.

2. Run this batch file **as administrator**. This will install the service.
3. Open the Windows Services application, look for the *KNIME Executor* service in the list and start it.
4. If you want to remove the executor service again, run the following **as administrator**:

```
<knime-installation>/remove-executor-as-service.bat
```

Note that if you move the KNIME Executor installation you first have to remove the service **before** moving the installation and then re-create it.

KNIME AG
Technoparkstrasse 1
8005 Zurich, Switzerland
www.knime.com
info@knime.com