Runing JIPipe on server

Prepackaged

Running JIPipe on a Linux Server using MobaXterm on Windows

This guide explains how to upload and run the prepackaged Linux version of JIPipe on a remote server using **MobaXterm** on Windows.

---

1. Download JIPipe

Open the official JIPipe download page:

https://jipipe.hki-jena.de/download

Download the **Prepackaged Linux64 ZIP** package.

The prepackaged version already contains Fiji/ImageJ and JIPipe.

Linux instructions are also available here:

https://applied-systems-biology.github.io/JIPipe-Documentation/prepackaged-jipipe.html#-uj2awg_7

---

2. Connect to the Server using MobaXterm

1. Open **MobaXterm**
2. Click **Session**
3. Select **SSH**
4. Enter:

- **Remote host**: your server hostname or IP
- **Specify username**: your Linux username

5. Click **OK**

You will now get a Linux terminal inside MobaXterm.

---

3. Upload the ZIP File

MobaXterm includes a built-in SFTP file browser on the left side.

Method 1: Drag and Drop (Recommended)

- Drag the downloaded `JIPipe-*-Prepackaged-Linux64.zip` file into the desired server directory.

Example destination:

```
/opt/jipipe/
```

Method 2: Use SCP Command

You can also upload from the terminal:

```
scp JIPipe-*-Prepackaged-Linux64.zip username@server:/opt/jipipe/
```

---

4. Unpack the ZIP File

Inside the terminal:

```
cd /opt/jipipe
unzip JIPipe-5.3.0-Prepackaged-Linux64.zip
```

This creates the extracted JIPipe directory.

---

5. Fix Permissions (Important)

If you get a **Permission denied** error when starting JIPipe, make the launcher executable.

The launcher scripts are located inside the `bin` folder:

```
cd JIPipe/bin
chmod +x ImageJ-linux64
chmod +x jipipe.sh
```

You can verify the permissions:

```
ls -l ImageJ-linux64
ls -l jipipe.sh
```

---

6. Start JIPipe

Start JIPipe using:

```
bash jipipe.sh
```

or:

```
./jipipe.sh
```

---

7. Running JIPipe on Remote Servers

For remote Linux servers, we currently recommend using **MobaXterm X11 forwarding** to display the JIPipe GUI remotely.

MobaXterm automatically provides an X11 server on Windows.

Before connecting:

1. Open **MobaXterm**
2. Go to:
   - **Settings → X11**
3. Ensure:
   - **X11 server is enabled**

When creating the SSH session:

1. Open **Session**
2. Select **SSH**
3. Enable:
   - **X11-Forwarding**

Then connect normally and start JIPipe:

```
cd /opt/jipipe/JIPipe/bin
bash jipipe.sh
```

The JIPipe GUI should appear on your Windows desktop.

---

8. Advanced / Untested: Headless Execution using Xvfb

> ⚠ Advanced / currently untested setup.

If the server has no graphical desktop environment, it may be possible to run JIPipe using a virtual display with `Xvfb`.

Install:

```
sudo apt install xvfb
```

Run:

```
xvfb-run bash jipipe.sh
```

This setup has not been fully tested with JIPipe workflows.

---

9. Keep JIPipe Running after Disconnecting

Use `screen`:

```
screen -S jipipe
cd /opt/jipipe/JIPipe/bin
```

```
bash jipipe.sh
```

Detach from the session:

```
Ctrl+A then D
```

Reconnect later:

```
screen -r jipipe
```

Building on the server

Building JIPipe from Source on a Linux Server using MobaXterm on Windows

This guide explains how to build and run JIPipe from source code on a Linux server using **MobaXterm** and **IntelliJ Remote Development**.

---

1. Requirements

JIPipe requires:

- Java
- Maven
- Git

Maven is already installed on the **Chrysos** server at the moment.

If Maven or Java is missing on another server, contact **Ruman** for installation support.

All steps below are performed through **MobaXterm**.

---

2. Connect to the Server using MobaXterm

1. Open **MobaXterm**
2. Click **Session**
3. Select **SSH**
4. Enter:
   - **Remote host**: your server hostname or IP

- **Specify username**: your Linux username

5. Click **OK**

You should now have a Linux terminal inside MobaXterm.

---

3. Clone the JIPipe Repository

Clone the repository from the internal ASB GitLab server:

```
git clone https://asb-git.hki-jena.de/RGerst/jipipe
```

Go into the repository:

```
cd jipipe
```

You can either:

- build directly on `master`
- or switch to your own branch

Example:

```
git checkout my-branch
```

---

4. Recommended: Use IntelliJ Remote Development

We strongly recommend using **IntelliJ IDEA Remote Development** for building JIPipe on remote servers.

Benefits:

- easier Maven handling
- easier indexing
- integrated terminal
- easier debugging
- easier remote GUI support

Open the remote project in IntelliJ and allow Maven indexing to finish before building.

---

5. Install Missing Maven Dependencies (Very Important)

Some Maven dependencies are no longer available online and must be copied manually.

Download or copy the dependency files from:

https://asb-git.hki-jena.de/RGerst/jipipe-distribution-files/-/tree/main/scripts/maven-dependencies?ref_type=heads

These dependencies must be copied into the local Maven repository:

```
~/.m2/repository
```

The `.m2` folder is hidden.

To locate it:

```
cd ~
ls -A
```

You should see:

```
.m2
```

Copy the dependency folders into:

```
~/.m2/repository
```

> ⚠ This step is extremely important.
> Without these dependencies the Maven build may fail.

You can use the MobaXterm file browser (left side panel) to drag-and-drop the dependency folders into the correct location.

---

6. Build JIPipe

Inside the repository root:

```
cd ~/jipipe
```

Run a full Maven install:

```
mvn clean install
```

This may take a long time during the first build.

Wait until all JAR files are built successfully.

7. Run JIPipe from Source

DISABLE CONDA env and make sure $JAVA_HOME is correct and points to the correct sdk, ex : /usr/lib/jvm/java-21-openjdk-amd64.

It can be done like this :

```
export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
```

After the build finishes, start JIPipe using:

```
java --add-opens=java.base/java.lang=ALL-UNNAMED -cp "$(cat jipipe-launcher/cp.txt):jipipe-launcher/target/classes" org.hkijena.jipipe.launcher.JIPipeLauncher
```

You can also run it through maven

```
MAVEN_OPTS="--add-opens=java.base/java.lang=ALL-UNNAMED" mvn -pl jipipe-launcher exec:java -Dexec.mainClass=org.hkijena.jipipe.launcher.JIPipeLauncher
```

Example terminal location:

```
(base) dkhatri@chrysos:~/jipipe$
```

If everything was built correctly, JIPipe should start normally.

For remote GUI usage, we recommend using **MobaXterm X11 forwarding**.

Before connecting:

1. Open **MobaXterm**
2. Go to:

    - **Settings → X11**
3. Ensure:
    - **X11 server is enabled**

When creating the SSH session:

1. Open **Session**
2. Select **SSH**
3. Enable:
    - **X11-Forwarding**

Then launch JIPipe normally.

The JIPipe GUI should appear on your Windows desktop.

---

8. Troubleshooting

Missing Dependencies

If Maven reports missing artifacts:

- verify that all manually provided dependencies were copied into:

```
~/.m2/repository
```

- restart IntelliJ Maven indexing if necessary

Permission Issues

If scripts cannot be executed:

```
chmod +x *.sh
```

Maven Not Found

Check Maven installation:

```
mvn --version
```

If the command is not found, contact **Ruman**.