Bring Your Own Device Guide - Part I: Preparation

This document describes the software requirements for completing the Distributed Systems assignments on your own Windows, Mac or Linux computer.

Common Requirements

Java SE Development Kit

Please make sure that you have Java SE Development Kit version **8** installed on your machine.

On Ubuntu, execute the following command in your terminal to make sure you have OpenJDK version 8 installed on your local system:

```
sudo apt-get install openjdk-8-jdk
```

For some assignments it is necessary to have the JAVA_HOME environment variable correctly configured, and have the bin directory in your JAVA_HOME in your path.
For more information on setting JAVA_HOME and adding Java to your path in Windows, Linux and OS X, follow the respective links.

Visual Paradigm

If you would like to use Visual Paradigm for drawing your diagrams, you can find the product installer and license code at https://ap.visual-paradigm.com/kuleuven.

Preparing for RMI

If you don't have an Eclipse installed on your local machine, we recommend you to install the Eclipse IDE for Enterprise Java Developers. The edition for Enterprise Java developers might not be necessarily required to accomplish the RMI assignments, but you will need this for later assignments (i.e. GAE). However, if you already have another edition installed (e.g. Eclipse IDE for Java), this should be sufficient for the RMI assignments.

Remote Setup for RMI

Our remote endpoint for RMI is only accessible on the computers in the labs at the ground floor of 200A. If you want to use it on your own laptop, you will need an ssh-tunnel to one of those machines.

Detailed instructions to be able to ssh into one of the machines can be found here. You will need to generate your ssh-keys if you haven't done this already. This process might take a day to complete!

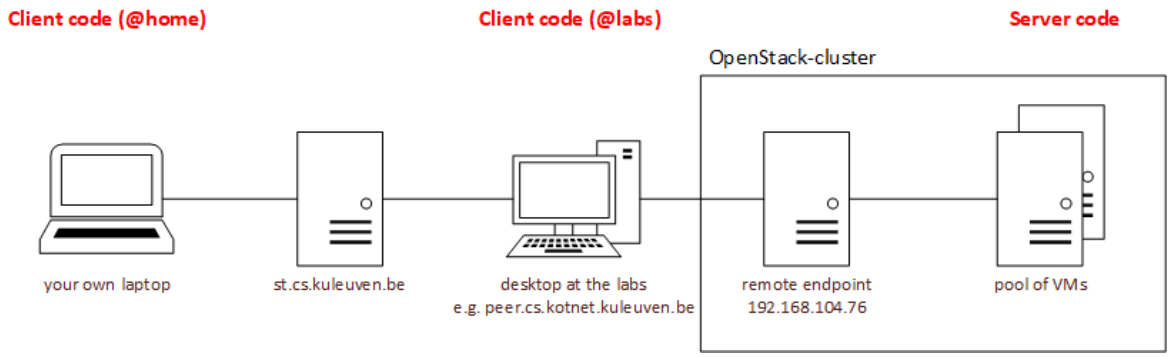
Once you are able to login using ssh, you can use the script below to setup the tunnels. Make sure you change the variables to use the correct path to your ssh-key, the actual ports that you are allocated and your own r-number . When the tunnel is active, our remote endpoint can be reached via `127.0.0.1:8080` on your computer. Be sure to use this IP and port in your ant-script!

This example uses the machine `peer`, it's best to use some random machine, so that not everyone uses the same machine. The list of available machines can be found here. This link is only available inside the department, but can also be accessed via an ssh-tunnel: `ssh -L 10080:mysql.cs.kotnet.kuleuven.be:80 <r-number>@st.cs.kuleuven.be`, the site can be found on `localhost:10080`.

```
# Results in:
# ssh -A -t -i <path to ssh-key, e.g. ~/.ssh/id_rsa> -L 8080:localhost:8080 -L
11001:localhost:11001 -L 11002:localhost:11002 -L 11003:localhost:11003 -L
11004:localhost:11004 -L 11005:localhost:11005 <r-number>@st.cs.kuleuven.be
ssh -L 8080:192.168.104.76:8080 -L 11001:192.168.104.76:11001 -L
11002:192.168.104.76:11002 -L 11003:192.168.104.76:11003 -L
11004:192.168.104.76:11004 -L 11005:192.168.104.76:11005 -N -o
StrictHostKeyChecking=no <r-number>@<name of machine, e.g.
peer>.cs.kotnet.kuleuven.be

SSHKEY="<path to ssh-key, e.g. ~/.ssh/id_rsa>"
PORTSTART=11001
PORTEND=11005
ENDPOINT="192.168.104.76"
STUDENTNR="<r-number>"
HOST="<name of machine, e.g. peer>"

SSHCOMMAND="ssh -A -t -i ${SSHKEY} -L 8080:localhost:8080 "
for PORT in $(seq ${PORTSTART} ${PORTEND})
do
  SSHCOMMAND+="-L ${PORT}:localhost:${PORT} "
done
SSHCOMMAND+="${STUDENTNR}@st.cs.kuleuven.be ssh -L 8080:${ENDPOINT}:8080 "
for PORT in $(seq ${PORTSTART} ${PORTEND})
do
  SSHCOMMAND+="-L ${PORT}:${ENDPOINT}:${PORT} "
done
SSHCOMMAND+="-N -o StrictHostKeyChecking=no
${STUDENTNR}@${HOST}.cs.kotnet.kuleuven.be"

eval ${SSHCOMMAND}
```

> Make sure to replace each <...> with the correct info and to change the port range to the ports you are assigned!

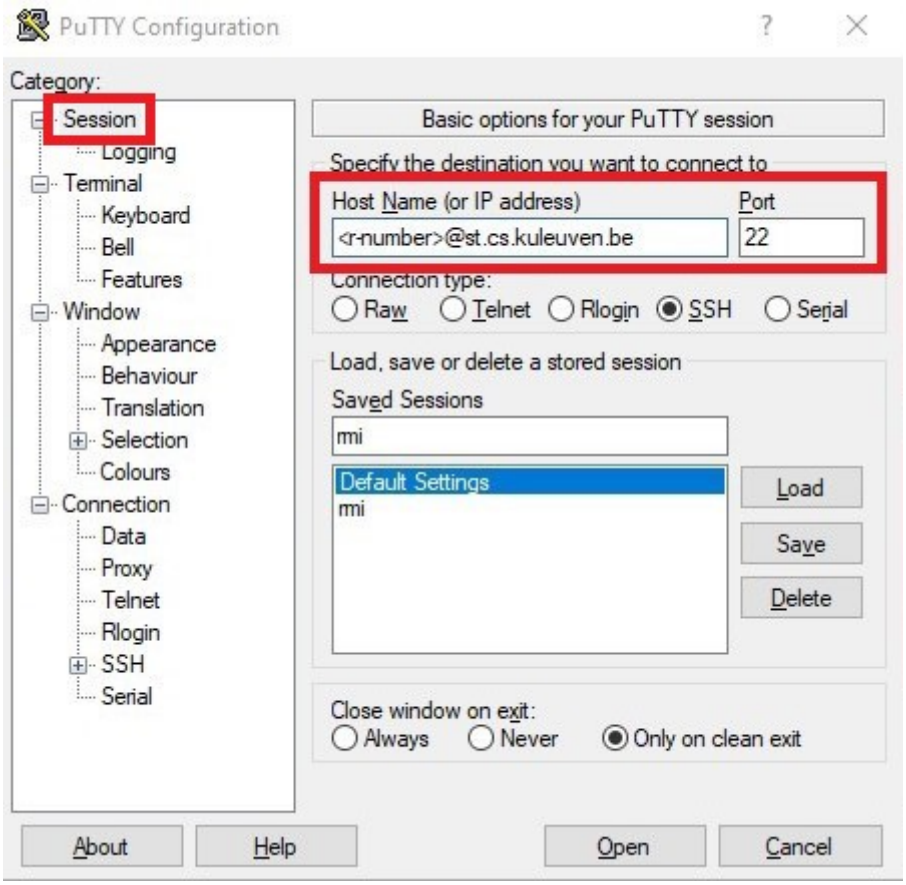
This is how the full distributed setup looks like (hiding firewalls and routers):



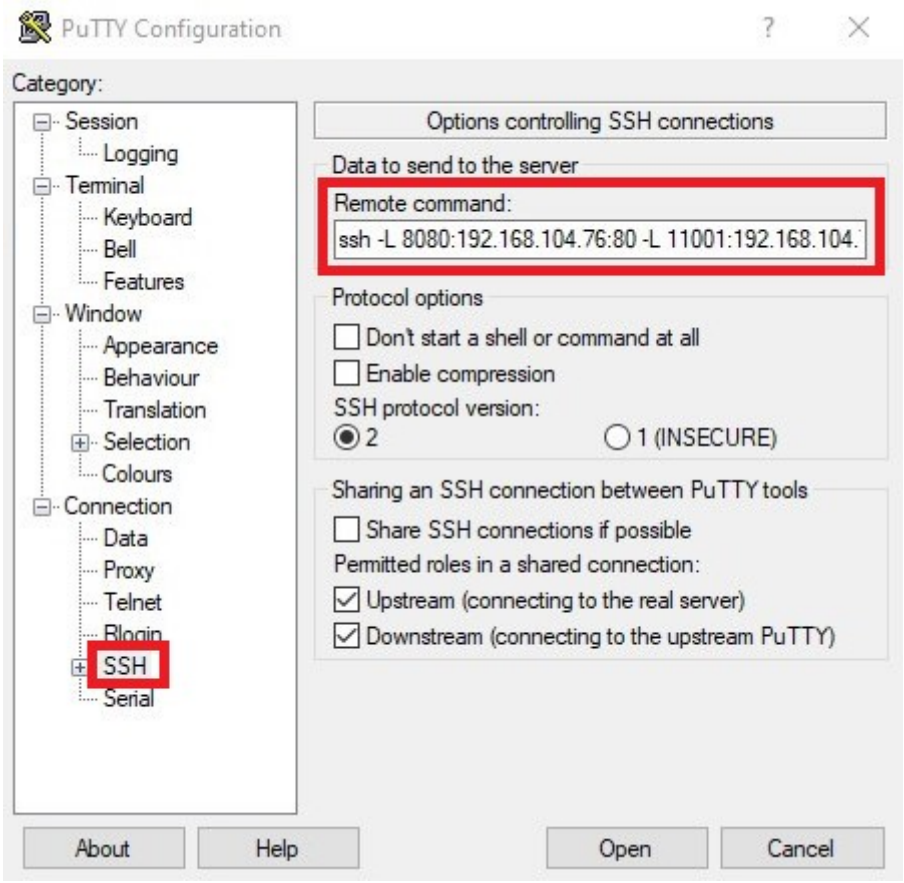
Windows

The above script only works on UNIX systems. On Windows, you can use the Windows Subsystem for Linux (WSL) or PuTTY. With WSL (recommended), you can use the same commands as above, otherwise you can use the below PuTTY configurations.

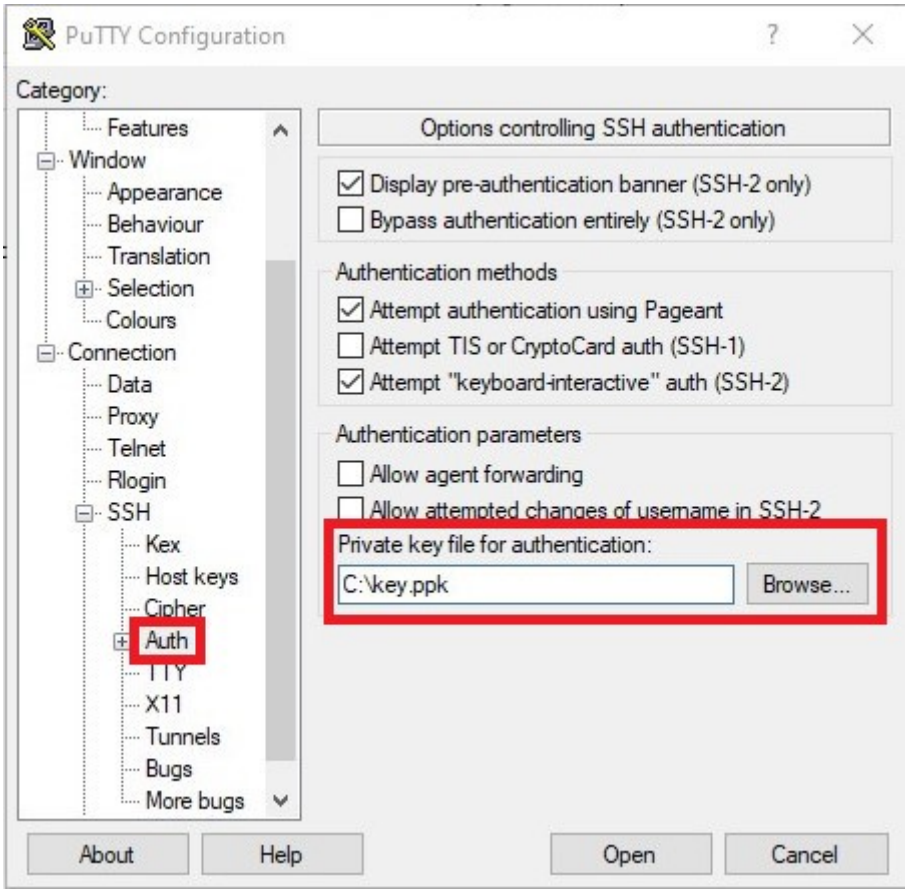
1. Fill in the host name and port, use your own r-number.



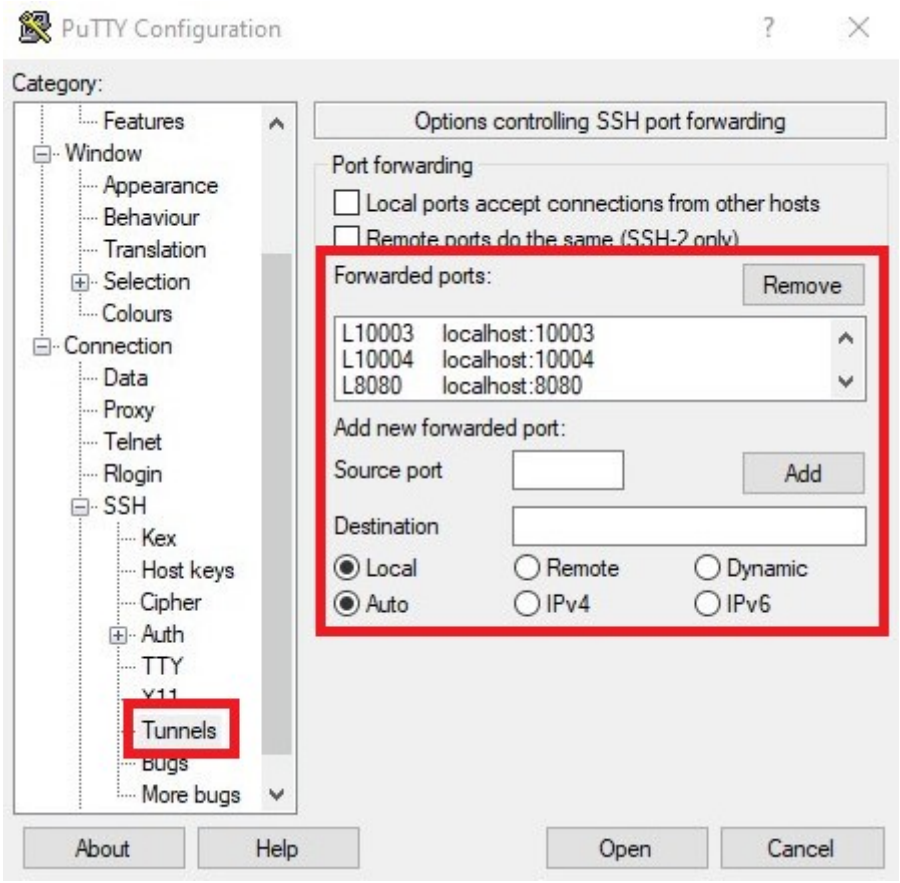
2. Add a remote command to setup the tunnels from `st.cs.kuleuven.be` to a machine in the lab. The full command is `ssh -L 8080:192.168.104.76:8080 -L 11001:192.168.104.76:11001 -L 11002:192.168.104.76:11002 -L 11003:192.168.104.76:11003 -L 11004:192.168.104.76:11004 -L 11005:192.168.104.76:11005 -N -o StrictHostKeyChecking=no <r-number>@<name of machine, e.g. peer>.cs.kotnet.kuleuven.be`. Note that you will need to change the ports, r-number and name of machine!



3. Provide the private key, you can use PuTTYgen to convert your ssh key to `.ppk` format (if not done already).



4. Add the ports to forward, you will need the following ports:
    - Source port: `8080`, Destination: `localhost:8080`
    - Source port: `<port in your provided range>`, Destination: `localhost:<same port in your provided range>`



5. On the sessions tab, you give this configuration a name and save it for later.

Preparing for JEE

We encourage you to use NetBeans to complete the JEE assignment. We use the Java EE edition of NetBeans IDE 8.2.

In addition, during the NetBeans installation, make sure you check *"Install GlassFish Application Server"*. During the assignment, you will be deploying your code to the GlassFish application server.

Preparing for GAE

Follow these steps to install the required software on your local machine:

1. Follow the Quickstart Guide of the Cloud Tools for Eclipse. Make sure to install the Eclipse IDE for Enterprise Java Developers.
2. Install the latest version of the Google Cloud SDK. Make sure to enable "Beta Commands" when using the interactive installer.
3. The first time you start the datastore, it will ask you to install the Cloud Datastore Emulator, type `Y` to install it.

Bring Your Own Device Guide - Part II: Typical Tasks

In this part, typical tasks for RMI, JEE and GAE related exercises are described for different operating systems.

Windows

Starting/Stopping RMI registry

To start RMI registry in `<DIR>` do:

1. Open command prompt (hit windows key and type `cmd`)
2. `cd <DIR>`
3. `start rmiregistry`

To stop RMI registry, close the RMI window.

Compiling Java sources from command line

When using file paths on Windows, for example, to add a jar file to the class path, as shown in section 5 of the first RMI assignment session (Bonus: distributed setup -> Run client application), do:

```
javac -cp .;<jar file> client/*.java
```

Launching a terminal

To get to a command prompt, open the Start Menu and type `cmd`. Launch the Command Prompt app.

Starting the Cloud Datastore emulator (for GAE)

In order to start the Cloud Datastore emulator, you will need to locate your installation of the Google Cloud SDK. If you automatically installed the SDK using the Cloud Tools for Eclipse plugin, the SDK is usually installed at `C:\Users\<user>\AppData\Local\Google\ct4j-cloud-sdk\LATEST\google-cloud-sdk`. The 'gcloud location' in the assignment is then the `bin/gcloud` program inside the SDK folder.

Mac

Launching a terminal

To get to a terminal, launch Terminal.app in the Application/Utils folder (or use spotlight to search for `Terminal` instead).

The configurations and commands should be similar to those that are performed on a Linux machine (see the assignment itself).

Starting the Cloud Datastore emulator (for GAE)

In order to start the Cloud Datastore emulator, you will need to locate your installation of the Google Cloud SDK. If you automatically installed the SDK using the Cloud Tools for Eclipse plugin, the SDK is usually installed at `$USER_HOME/Library/Application Support/google-cloud-tools-java/managed-cloud-sdk/<version>/google-cloud-sdk`. The 'gcloud location' in the assignment is then the `bin/gcloud` program inside the SDK folder.

Linux

Launching a terminal

On Ubuntu, you can open your terminal by typing `Ctrl` + `Alt` + `T` or launching the Terminal app.

Starting the Cloud Datastore emulator (for GAE)

In order to start the Cloud Datastore emulator, you will need to locate your installation of the Google Cloud SDK. If you automatically installed the SDK using the Cloud Tools for Eclipse plugin, the SDK is usually installed at `/home/<user>/.cache/google-cloud-tools-java/managed-cloud-sdk/LATEST/google-cloud-sdk`. The 'gcloud location' in the assignment is then the `bin/gcloud` program inside the SDK folder.

Bring Your Own Device Guide - Part III: Troubleshooting

Windows

Java EE

- You are able to build your project successfully, but getting the exception: “The module has not been deployed See the server log for details” while deploying your application.
    - **Solution:** Make sure that the path of the project directory does not contain any spaces. For example if your project is stored in `D:\Distributed Systems\ds_jee_1\`, a solution would be to remove the space between `Distributed` and `Systems`.
- When you run into trouble deploying or running your application in NetBeans, consider removing these directories to start fresh:
    - **Solution:** In the service view -> Right-Click on Databases->JavaDB and select properties. Go to database location and delete `.netbeans-derby`.

Mac

Java EE

- When you run into trouble deploying or running your application in NetBeans, consider removing these directories to start fresh (cfr. assignment, section 4, Important):
    - `/Users/$USER/Library/Application\ Support/NetBeans/{version}`
    - `/Users/$USER/.netbeans-derby`