

Outgoing connections

Connection restrictions

Outgoing connections, from Anselm Cluster login nodes to the outside world, are restricted to following ports:

Port	Protocol
---	---
22	ssh
80	http
443	https
9418	git

Please use **ssh port forwarding** and proxy servers to connect from Anselm to all other remote ports.

Outgoing connections, from Anselm Cluster compute nodes are restricted to the internal network. Direct connections from compute nodes to outside world are cut.

Port forwarding

Port forwarding from login nodes

Port forwarding allows an application running on Anselm to connect to arbitrary remote host and port.

It works by tunneling the connection from Anselm back to users workstation and forwarding from the workstation to the remote host.

Pick some unused port on Anselm login node (for example 6000) and establish the port forwarding:

```
local $ ssh -R 6000:remote.host.com:1234 anselm.it4i.cz
```

In this example, we establish port forwarding between port 6000 on Anselm and port 1234 on the remote.host.com. By accessing localhost:6000 on Anselm, an application will see response of remote.host.com:1234. The traffic will run via users local workstation.

Port forwarding may be done **using PuTTY** as well. On the PuTTY Configuration screen, load your Anselm configuration first. Then go to Connection->SSH->Tunnels to set up the port forwarding. Click Remote radio button. Insert 6000 to Source port textbox. Insert remote.host.com:1234. Click Add button, then Open.

Port forwarding may be established directly to the remote host. However, this requires that user has ssh access to remote.host.com

```
$ ssh -L 6000:localhost:1234 remote.host.com
```

Note: Port number 6000 is chosen as an example only. Pick any free port.

Port forwarding from compute nodes

Remote port forwarding from compute nodes allows applications running on the compute nodes to access hosts outside Anselm Cluster.

First, establish the remote port forwarding from the login node, as described above.

Second, invoke port forwarding from the compute node to the login node. Insert following line into your jobscript or interactive shell

```
$ ssh -TN -f -L 6000:localhost:6000 login1
```

In this example, we assume that port forwarding from login1:6000 to remote.host.com:1234 has been established beforehand. By accessing localhost:6000, an application running on a compute node will see response of remote.host.com:1234

Using proxy servers

Port forwarding is static, each single port is mapped to a particular port on remote host. Connection to other remote host, requires new forward.

Applications with inbuilt proxy support, experience unlimited access to remote hosts, via single proxy server.

To establish local proxy server on your workstation, install and run SOCKS proxy server software. On Linux, sshd demon provides the functionality. To establish SOCKS proxy server listening on port 1080 run:

```
local $ ssh -D 1080 localhost
```

On Windows, install and run the free, open source Sock Puppet server.

Once the proxy server is running, establish ssh port forwarding from Anselm to the proxy server, port 1080, exactly as described above.

```
local $ ssh -R 6000:localhost:1080 anselm.it4i.cz
```

Now, configure the applications proxy settings to **localhost:6000**. Use port forwarding to access the proxy server from compute nodes as well .