Date Published: 06/07/2022

Student Cluster Competition - Tutorial 2

Table of Contents

Overview

This tutorial will demonstrate how to access web services that are on your virtual cluster via the web browser on your local computer. It will also cover basic authentication and central authentication.

In this tutorial you will:

- [ ] Install a web server.
- [ ] Create an SSH tunnel to access your web service.
- [ ] Create new local user accounts.
- [ ] Add local system users to sudoers file for root access.
- [ ] Share directories between computers.
- [ ] Connect to machines without a password using public key based authentication.
- [ ] Install and use central authentication.

Date Published: 06/07/2022

Part 1 - Remote Web Service Access

During the course of the competition, you will set up services on your head node that you need to access via a web browser. Considering that your cluster is not directly accessible from the internet, you will need to create an **SSH tunnel** (utilising port forwarding) between your local machine and your head node to access these webpages.

> **! >>> Please read and familiarise yourself with the concept of a *port* before continuing:**
>
> - **Dummies.com - Network Basics: TCP/UDP Socker and Port Overview**
> - **Wikipedia - Port (computer networking)**

We'll demonstrate how this is done. First, let's create a simple web page that you can serve from your head node. To do this, you need to install a web server. Apache is a standard and widely used open-source web server. To install it:

```
~$ dnf install httpd
```

You'll then need to enable and start the web server service.

```
~$ systemctl start httpd   # Starts the service
~$ systemctl enable httpd  # Sets service to start on reboot automatically
```

Confirm that it's up and running with the following:

```
~$ systemctl status httpd
```

You now have a running web server. Apache has a default available test-page, but since **the networks for your cluster aren't available outside the ACE Lab private network**, you **can't** simply type the IP address of the head node into your local web browser to access it. To access the test-page (served by your head node) you must establish a tunnel between your local machine and the head node, using the ACE Lab login node (ssh.ace.chpc.ac.za) as a middle-man. This is done using `ssh`.

You need to establish a specific SSH tunnel to achieve this. The specific tunnel demonstrated below is known as an SSH forward tunnel, or SSH local port forwarding. To achieve this, you must tell the `ssh` client on your **local machine (computer at home)** that you will be sending and receiving data to and from a specific port on the **target machine (your head node in this case)** via a specific port on your **local machine**.

Once this tunnel is established, you will be able to open your **local web browser** and access the **local port (the one that you configured to have data forwarded to from the head node)** to see the data forwarded from the target port. Connecting to the local port will request that the data be sent from the target machine - showing you the web page as if you were on the same network as the head node.

Web traffic is by default served on **port 80**. This is thus chosen as the **target port on the head node**, as you want to be able to view this web traffic on your local computer. For the **local port**, we can choose **any port number greater than 1000**, as anything over 1000 is non-system and non-privileged (doesn't require root access).

> **! >>> Please note that the IP address used below is an example. ! >>> The IP address to use instead of 192.168.0.xx is your HEADNODE public IP.**

Date Published: 06/07/2022

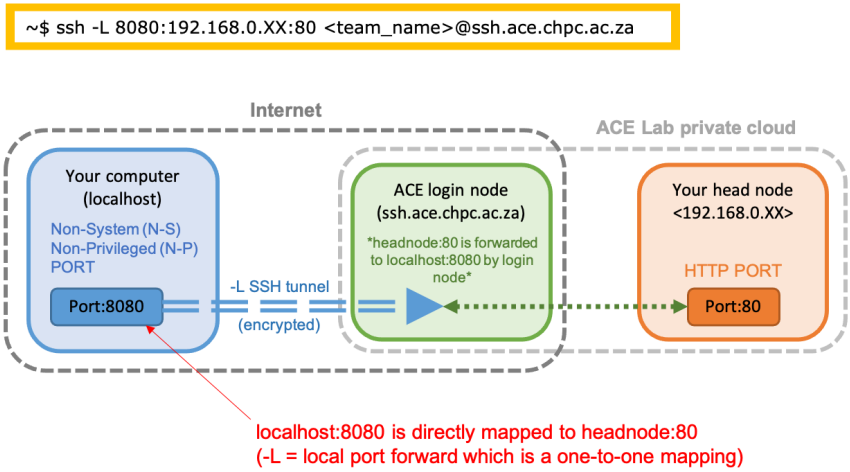

Figure 1.1: SSH -L tunnel as described below.

1. Firstly, let us allow web traffic through the head node firewall (this is done on the **head node**):

```
~$ firewall-cmd --zone=external --add-service=http --permanent
~$ firewall-cmd --reload
```

2. On your **local machine**, open up the tunnel from the head node's port 80 to your machine's port 80 (Figure 1.1):

```
~$ ssh -L 8080:10.128.24.XX:80 <team_name>@ssh.ace.chpc.ac.za
```

   This command uses the following syntax:

```
ssh -L <localhost_port:target_host:target_host_port> <username>@<remote_host>
```

   The `-L` specifies that you want to create a port forward to/from your `<localhost_port>` to the `<target_host_port>` of the `<target_host>`.

3. Open up your browser and visit:

```
http://127.0.0.1:8080 # 127.0.0.1 is a reference to your own machine, you could also say http://localhost:8080
```

If it is up and running correctly, you should see the default test-page for your Apache server (Figure 1.2).

Date Published: 06/07/2022

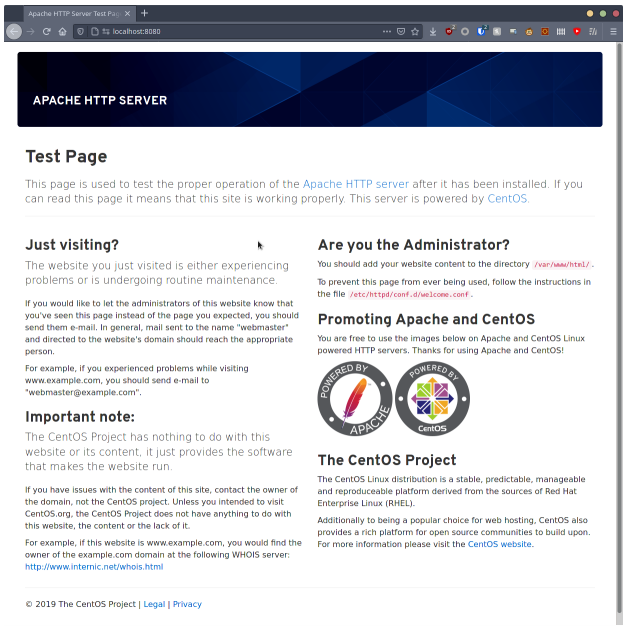

Figure 1.2: The default page for the Apache web server seen through the local browser.

Windows users with PuTTY, please follow this guide: https://stackoverflow.com/questions/4974131/how-to-create-ssh-tunnel-using-putty-in-windows/29168936#29168936.

To clarify, what the `ssh` command above does is the following:

The `-L 8080:10.128.24.XX:80` tells the `ssh` client that you want to map your local machine port `8080` to the head node's port `80`. However, your local machine can't reach the head node directly since you're not on the same network. In order to know how to get to the head node, you still connect to the login node for the ACE Lab, which is why you specify the `<team_name>@ssh.ace.chpc.ac.za`. The `ssh` client will know to map your port `8080` to the head node's port `80` **via** `ssh.ace.chpc.ac.za`.

Date Published: 06/07/2022

Part 2 - Local User Account Management

In enterprise systems and HPC, it's common to manage user accounts from one central location. These network accounts are then synchronised to the machines in your fleet via the network. This is done for safely, security and management purposes.

When creating a user account locally on a Linux operating system, it's provided with a user ID (uid) and a group ID (gid). These are used to tell the operating system which user this is and which groups of permissions they belong to. When you create a user with the default settings of the built-in user creation tools, it will generally increment on from the last UID used. This can be different for different systems. If UID/GID numbers do not match up across the nodes in your cluster, there can be all sorts of headaches for some of the tools and services that we will set up later in this competition.

We're going to demonstrate some of this.

Right now you have one user: `root`. `root` is the default super-user of Linux operating systems. It is all powerful. It is generally **NOT recommended** to operate as `root` for the majority of things you would do on a system. This is to prevent things from going wrong.

When logged in to the head node or compute node, check the UID and GID of `root` by using the `id` command.

```
~$ id
```

You should see something like the following:

```
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

This shows that `root` is the user `0` and it's primary group (GID) is group `0`. It also states that it only belongs to one group, which is the `root` group (`0`).

Create CentOS User Account

Head Node

Let us now create a user account on the head node:

1. Log into the head node
2. Use the `adduser` command to create a new user called `centos` and then give it a password.

   ```
   [root@headnode ~]$ adduser -U -m centos
   [root@headnode ~]$ passwd centos
   ```

   `-U` tells `adduser` to create a group for the user and `-m` means to create the user home directory.
3. Check the ID of the new user

   ```
   [root@headnode ~]$ id centos
   ```

   You'll see something like the following:

```
uid=1000(centos) gid=1000(centos) groups=1000(centos)
```

As you can tell, it has a different ID for the user and group than `root`.

Compute Node

Log into thetest compute node and try to verify that the `centos` user **does NOT exist** there:

```
[root@computenode ~]$ id centos
```

You'll be prompted with an error:

```
id: ‘centos’: no such user
```

We will now create the same user here. Follow the steps above for creating the `centos` user on the compute node.

Super User Access

The `centos` user will not have the privileges to do anything that modify system files or configurations. Many Linux operating systems come with a program called `sudo` which manages and allows normal user accounts to access `root` privileges.

A user is only able to evoke root privileges if their account has been explicitly added to at least one of the following:

- the default sudo users group (the actual term of this group varies across Linux variants, such as **wheel sudoers** etc.)
- a newly created sudo users group,
- or, if the user has been explicitly added as a privileged user directly in the Sudo configuration file.

The `sudo` program is controlled by a file located at `/etc/sudoers`. This file specifies which users and/or groups can access superuser privileges. In this file for a default **CentOS 8** installation, it specifies that the user `root` is allowed to run all actions and any user in the `wheel` group is also allowed to:

```
## Allow root to run any commands anywhere
root    ALL=(ALL)   ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel    ALL=(ALL)   ALL
```

To avoid modifying `/etc/sudoers` directly, we can just add `centos` to the `wheel` group.

On each of your nodes, add the `centos` user to the `wheel` group:

```
[root@node ~]$ gpasswd -a centos wheel
```

Now log out and then log back into your node as `centos`. You can use `sudo` one of two ways:

1. To become the `root` user:

Date Published: 06/07/2022

```
[centos@headnode ~]$ sudo su
```

2. To run a command with superuser privileges:

```
[centos@headnode ~]$ sudo <command>
```

`sudo` will prompt you for the `centos` user password when you run it.

> **! >>> From now on, you should use the `centos` user for all the configuration you can and should avoid logging in as the `root` user.**

Date Published: 06/07/2022

Part 3 - Network File System

Network File System (NFS) enables you to easily share files and directories over the network. NFS is a distributed file system protocol that we will use to share files between our nodes across our private network. It has a server-client architecture that treats one machine as a server of directories, and multiple machines (clients) can connect to it.

This tutorial will show you how to export a directory on the head node and mount it through the network on the compute nodes. With the shared file system in place it becomes easy to enable **public key based ssh authentication**, which allows you to ssh into all the computers in your cluster without requiring a password.

NFS Server (head node)

The head node will act as the NFS server and will export the `/home/` directory to the compute node. The `/home/` directory contains the home directories of all the the non-`root` user accounts on most default Linux operating system configurations.

1. Firstly, install the NFS service on the head node:

```
[centos@headnode ~]$ sudo dnf install nfs-utils
```

2. We're going to simplify the configuration of NFS by using the older version, 3. To do so, edit the file (you might need to create it) `/etc/sysconfig/nfs` and add the following:

```
MOUNTD_NFS_V3="yes"
RPCNFSDARGS="-N 4"
```

3. NFS shares (directories on the NFS server) are configured in the `/etc/exports` file. Here you specify the directory you want to share, followed by the IP address or range you want to share to and then the options for sharing. We want to export the `/home` directory, so edit `/etc/exports` and add the following:

```
/home    10.0.0.0/24(rw,async,insecure,no_root_squash)
```

4. Start and enable the `nfs-server` service using `systemctl`.

5. Since we trust our internal network, for the future of this competition we're going to allow all ports of the internal cluster network to be unblocked. **This is not recommended in production environments!**:

```
[centos@headnode ~]$ sudo firewall-cmd --permanent --zone=internal --set-target=ACCEPT
[centos@headnode ~]$ sudo firewall-cmd --reload
```

NFS Client (compute node)

The compute node acts as the client for the NFS, which will mount the directory that was exported from the server (`/home`). Once mounted, the compute node will be able to interact with and modify files that exist on the head node and it will be synchronised between the two.

Mounting An NFS Mount

The `nfs-utils` package needs to be installed before you can do anything NFS related on the compute node. Since the directory we want to mount is the `/home` directory, the user can not be in that directory.

Date Published: 06/07/2022

1. Once installed, mount the /home directory from the head node using the `mount` command:

```
[centos@computenode ~]$ sudo mount -t nfs <headnode_ip_or_hostname>:/home /home
```

2. Once done, you can verify that the `/home` directory of the head node is mounted by using `df -h`:

```
[centos@computenode ~]$ df -h
```

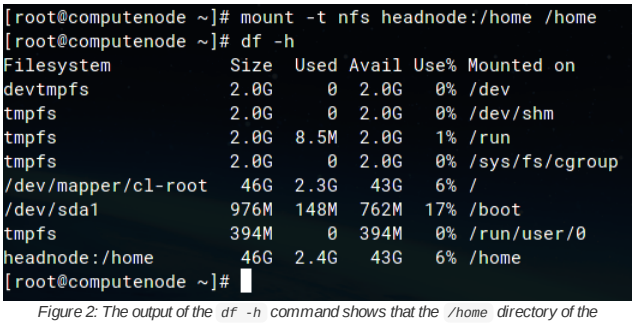

Figure 2: The output of the `df -h` command shows that the `/home` directory of the head node is mounted on the `/home` directory of the compute node.

With this mounted, it effectively replaces the `/home` directory of the compute node with the head node's one until it is unmounted. To verify this, create a file on the compute node's `centos` user home directory (`/home/centos`) and see if it is also automatically on the head node. If not, you may have done something wrong and may need to redo the above steps!

Making The NFS Mount Permanent

Using `mount` from the command line will not make the mount permanent. It will not survive a reboot. To make it permanent, we need to edit the `/etc/fstab` file on the compute node. This file contains the mappings for each mount of any drives or network locations on boot of the operating system.

1. First we need to unmount the mount we made:

```
[centos@computenode ~]$ sudo umount /home
```

2. Now we need to edit the `/etc/fstab` file and add this new line to it (be careful not to modify the existing lines!):

```
headnode.cluster.scc:/home    /home     nfs    vers=3,_netdev,intr    0 0
```

   The structure is: `<host>:<filesystem_dir> <local_location> <filesystem_type> <filesystem_options> 0 0`. The last two digits are not important for this competition and can be left at 0 0.

   For the nfs options listed above:

   - `vers=3` means that we want to force NFSv3.
   - `_netdev` tells the operating system to only mount the device once network has been established (it's a network device).
   - `intr` allows NFS operations to be interrupted in the case that the server is unreachable.

3. With this done, we can mount the new `/etc/fstab` entry:

```
[centos@computenode ~]$ sudo mount -a
```

Date Published: 06/07/2022

4. Once again, you can verify that the `/home` directory of the head node is mounted by using `df -h`:

```
[centos@computenode ~]$ df -h
```

Date Published: 06/07/2022

Part 4 - Password-less SSH

When managing a large fleet of machines or even when just logging into a single machine repeatedly, it can become very time consuming to have to enter your password repeatedly. Another issue with passwords is that some services may rely on directly connecting to another computer and can't pass a password during login. To get around this, we can use public key based authentication to replace passwords.

1. Generate an SSH key-pair for your user. This will create a public and private key for your user in `/home/<username>/.ssh`. The private key is your identity and the public key is what you share with other computers.

   ```
   [centos@headnode ~]$ ssh-keygen
   ```

   You can hit enter with the defaults/empty for all the prompts.

2. Copy the public key generated by `ssh-keygen` into the `authorized_keys` file in the same directory.

   ```
   [centos@headnode ~]$ cd ~/.ssh
   [centos@headnode .ssh]$ cat id_rsa.pub > authorized_keys
   ```

   Since your `/home` directory is shared with your compute node, this will look the same on the compute node.

3. SELinux, the security engine that **CentOS 8** uses, may complain about permissions for this directory if you try to use public key authentication now. To fix this, run the following commands:

   ```
   [centos@headnode ~]$ chmod 700 ~/.ssh/
   [centos@headnode ~]$ chmod 600 ~/.ssh/authorized_keys
   [centos@headnode ~]$ sudo restorecon -R -v ~/.ssh
   ```

4. SSH to the **compute node** and run the following command:

   ```
   [centos@computenode ~]$ sudo setsebool -P use_nfs_home_dirs 1
   ```

5. Exit **back to the head node**

6. SSH to your compute node without a password and land on the shared filesystem. If you are prompted with a password it means that something is not set up correctly.

> **! >>> `chmod` and `chown` are Linux permission and ownership modification commands. To learn more about these commands and how they work, please go to the following link: https://www.unixtutorial.org/difference-between-chmod-and-chown/.**

How this works is that you copy the public key to the computer that you want to connect to without a password's `authorized_keys` file. When you SSH to the machine that you copied your public key to, the `ssh` tool will send a challenge that can only be decrypted if the target machine has the public key and the local machine has the private key. If this succeeds, then you are who you say you are to the target computer and you do not require a password. Please read this for more detailed information.

Date Published: 06/07/2022

Part 5 - Central User Management

Out-Of-Sync Users and Groups

When managing a large cluster of machines, it gets really complicated to manage user ID and group ID mappings. With things like shared file systems (e.g. NFS), if user account names are the same, but IDs don't match across machines then we get permission problems.

If users are created out-of-sync across the cluster then this becomes a problem very quickly. Let us take Alice and Bob for example:

1. Alice and Bob are both system administrators working on a cluster.
2. There is no central authentication and user/group accounts are made manually.
3. Alice creates a user `alice` on the head node using the `adduser` command listed in this tutorial.
4. While Alice does this, Bob creates user `bob` on the compute node in the same way.
5. Alice then creates user `alice` on the compute node.
6. Bob creates `bob` on the head node.

Even though the names are the same:

- `alice` on the **head node** has a UID/GID of `1000` / `1000`
- `bob` on the **head node** has a UID/GID of `1001` / `1001`
- `alice` on the **compute node** has a UID/GID of `1001` / `1001`.
- `bob` on the **compute node** has a UID/GID of `1000` / `1000`.

These do not match, so if Alice wants to create a file on the head node and access that file on the compute node she will get permission errors as `1000` is not the same as `1001`.

User- and group- names do not matter to Linux, only the numerical IDs. Let us demonstrate this now.

Head Node

1. Create a new user on the head node, let's call it `outofsync`. If you check it's IDs with `id outofsync`, you should see it belongs to UID/GID `1001`.

2. Set the password for this user and log in as this user.

3. Create a file in the home directory of `outofsync` (`/home/outofsync`) called `testfile.txt` and put some words in it.

Compute Node

1. Create a new user on the compute node called `unwittinguser`. If you check the ID of this user, you will see that `unwittinguser` has UID/GID of `1001`.

2. Create a new user on the compute node called `outofsync`. If you check the ID of this user, you will see that `outofsync` has UID/GID of `1002`.

3. Set the password for the `outofsync` user.

4. Log into the compute node as `outofsync`.

5. You will see that the terminal complains about permission errors and that you aren't logged into the user's home directory.

6. You will not be able to read the `testfile.txt` file in `/home/outofsync/testfile.txt` if you tried.

Date Published: 06/07/2022

This happens because you have an NFS mount for `/home`, replacing (while mounted) the compute node's `/home` with the head node's `/home` and the UID/GID for `outofsync` on the compute node does not match the one on the head node.

Check `ls -ln /home/outofsync` on the **head node** and you'll see that the `testfile.txt` belongs to `1001`, not `1002`.

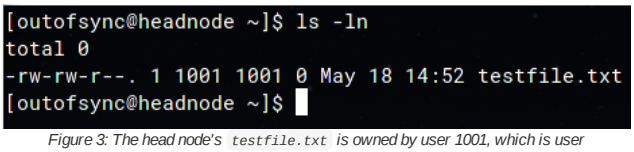

Figure 3: The head node's `testfile.txt` is owned by user 1001, which is user `outofsync` on the head node.

Clean Up

Before proceeding, you must delete the users that you have created on the machines.

To delete a user you can use the command below:

```
~$ sudo userdel -r <username>
```

Do this command for:

- `outofsync` on the head node.
- `unwittinguser` on the compute node.
- `outofsync` on the compute node.

FreeIPA

FreeIPA is a collection of tools that work together to provide central user account management. It provides identity and authentication services for Linux environments. It can also manage DNS and NTP, but we won't use it for that purpose in this tutorial series. Using FreeIPA, you will be able to keep your cluster user account IDs synchronised and manage everything from one central place.

> **! >>> Learn about LDAP, Kerberos and other authentication tools which make up FreeIPA here: https://www.freeipa.org/page/About.**

FreeIPA Server (Head Node)

> **! >>> It is recommended to run the commands below in a `screen` or `tmux` session as some of them run for extended periods of time.**

The FreeIPA server is provided via the "Identity Management DL1" module of **CentOS 8** AppStream. To enable that, use the command below:

```
[centos@headnode ~]$ sudo dnf module enable idm:DL1
```

You can now install the FreeIPA server:

```
[centos@headnode ~]$ sudo dnf install ipa-server
```

Once the software is installed, we can initialise the FreeIPA server.

Date Published: 06/07/2022

```
[centos@headnode ~]$ sudo ipa-server-install --mkhomedir --no-ntp --no-dns-sshfp --no-host-dns
```

You'll be asked a couple of questions, for them answer as follows:

Prompt	Your Answer
Do you want to configure integrated DNS (BIND)?	no
Enter the fully qualified domain name of the computer…	`headnode.cluster.scc`
Please confirm the domain name	`cluster.scc`
Please provide a realm name	`CLUSTER.SCC`
Directory Manager password	`<choose_your_own>`
IPA admin password	`<choose_your_own>`

Confirm the details that pop up in the summary screen. If you are happy, type `yes` for `Continue to configure the system with these values?`.

The setup will now take some time. Please be patient and do not cancel it.

Once done, you will be prompted to provide the `centos` user with an admin kerberos ticket. A user needs an administrative kerberos ticket to have the ability to manage FreeIPA. This allows the given user to address FreeIPA via the command line. However, the more interesting way of managing FreeIPA is with the web interface which is discussed later.

If you are not prompted by the installer automatically, you can manually initialise this step afterwards by running the following command:

```
[centos@headnode ~]$ kinit admin
```

Lastly, run the following command:

```
[centos@headnode ~]$ ipa config-mod --defaultshell=/bin/bash
```

FreeIPA Client (Compute Node)

> **! >>> It is recommended to run the commands below in a `screen` or `tmux` session as some of them run for extended periods of time.**

The FreeIPA client is provided via the "Identity Management client" module of **CentOS 8** AppStream. To enable that, use the command below:

```
[centos@computenode ~]$ sudo dnf module enable idm:client
```

Install the FreeIPA client packages.

```
[centos@computenode ~]$ sudo dnf install ipa-client
```

Date Published: 06/07/2022

Now, to add this compute node to the FreeIPA server you need to run the `ipa-client-install` tool with some parameters related to your server.

```
[centos@computenode ~]$ sudo ipa-client-install --principal=admin --domain=cluster.scc \
                        --server=headnode.cluster.scc --realm=CLUSTER.SCC -W --mkhomedir \
                        --ntp-server=headnode.cluster.scc --no-dns-sshfp
```

You will be prompted with a question "**Proceed with fixed values and no DNS discovery?**. Type `yes` and hit enter.

Verify that the information is correct in the next prompt. If so, type `yes` and hit enter.

When prompted with the password for `admin@CLUSTER.SCC`, enter the **IPA admin password** and hit enter. After the install is successful, verify that it is working bu running `id admin`. You should see the user admin that belongs to the admins group.

Congratulations! You've just configured central authentication. The user `admin` exists on the head node and the compute node as managed by FreeIPA.

FreeIPA Web Interface

Dynamic SSH Tunnel

The FreeIPA web interface is hosted on the head node and available on port `443 (https)`. You won't be able to access this interface using the SSH Tunnel technique described in Part 1), as when you enter the address to access the forwarded port (something like http://127.0.0.1:1234) FreeIPA's web interface will automatically redirect you to `https://headnode.cluster.scc`. Since `headnode.cluster.scc` (or whatever your head node name is), does not exist on your local network at home, this will result in your browser telling you that it can't find that server.

We'll need to use a **dynamic SSH tunnel** (SOCKS proxy) to get around this. This allows SSH to forward **ALL remote (target) port traffic** to and from a port on your local machine, and can include remote DNS.

0. First, open two terminals on your personal computer.

1. In terminal one, type:

   ```
   ssh -L 1234:<headnode_ip>:22 <team_name>@ssh.ace.chpc.ac.za
   ```

   And sign in with your team password.

2. Leave that terminal open, go to the second terminal and type:

   ```
   ssh -p 1234 -D 1235 centos@127.0.0.1
   ```

   The `-p 1234` tells `ssh` to use `1234` as the SSH port instead of the default `22`. The `-D 1235` tells `ssh` to open `1235` on your local computer for sending and receiving all port traffic to and from the target machine (`127.0.0.1:1234`, which in this case is `<headnode_ip>:22` because of the `-L`.)

You are essentially **hopping through** `ssh.ace.chpc.ac.za` into your head node directly.

Windows user with PuTTY, please read the following PDF: https://webdevolutions.blob.core.windows.net/blog/pdf/how-to-configure-an-ssh-tunnel-on-putty.pdf. Please refer to Step 4.

Firefox and Proxy Configuration

`Date Published: 06/07/2022`

The Firefox browser will allow the easiest proxy configuration. Please download and install Firefox from here: https://www.mozilla.org/en-US/firefox/download/.

Once downloaded and opened, go to the `three line menu` at the top right and click on `Preferences`. In the `Find in Preferences` search bar type "proxy" and click `Settings` next to the **Network Settings** option.

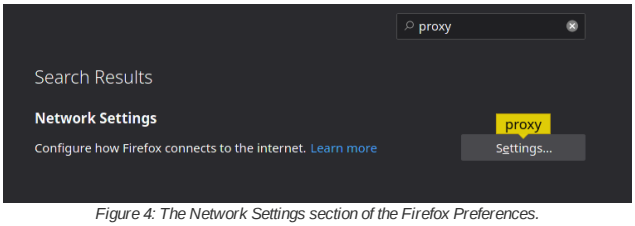

Figure 4: The Network Settings section of the Firefox Preferences.

In this pop-up, change The proxy setting to `Manual Proxy Configuration` and delete `HTTP Proxy`, `HTTPS Proxy` and `FTP Proxy` and set their port numbers to `0`.

In `SOCKS Host` enter `127.0.0.1` and for the port enter `1235`.

Select `SOCKS v5` and **tick on** `Proxy DNS when using SOCKS v5`.

Now click `OK` and open a new tab in Firefox.

Now you can enter https://headnode.cluster.scc/ipa/ui in your browser and you'll get access to the FreeIPA Web interface. With this, you can log in using the FreeIPA admin user and password.

> **! >>> Keep both of the SSH sessions open while you use the proxy on Firefox**

> **! >>> Remember to set your proxy settings back to `No Proxy` if you want to use your own local internet on Firefox.**

Creating a User

With the proxy up and set in Firefox, go to https://headnode.cluster.scc/ipa/ui and log in with the `admin` user and the password you set up for **IPA admin password**.

Once logged in, you'll see that by default there is an `admin` user with some UID assigned to it. We can ignore this user for now.

We need to do two things here:

- Create a group and give it superuser permissions.
- Create a user to replace the `centos` user.
- Add the new user to the above mentioned group.

Creating the Group

1. In the main menu, click "Groups".

2. Under "User Groups", click "+ Add" and name the group "sysadmin" (short for systems administrator, that's what you are!).

3. Click "Add" at the bottom.

4. Now at the top, click the "Policy" button and go to the "Sudo" -> "Sudo Rules" section.

5. Click "+ Add", name it "sysadmin sudo" and click "Add and Edit".

6. Click "+ Add" next to "User Groups" under **"Who"**, tick the "sysadmin" group, click the `>` arrow and click "Add".

7. 8. Click the "Any Host" button under **"Run Commands"**.

Date Published: 06/07/2022

8. Click the "Any Command" button under **"Run Commands"**.

9. Click the "Anyone" and "Any Group" buttons under **"As Whom"**.

10. Click the "Save" button at the top of the page (under "Settings").

Creating the Users

Make sure that you go back to the main menu by clicking the "Identity" button.

1. In the main menu, click the "+ Add" button on the top right.

2. Create a user account for each of your team members (one at a time).

    - Give the user a user name (User login). Make it the first letter of their first name followed by the full surname. As an example: if your name is Bob John, you could make it bjohn.
    - Enter the first and last name in the boxes.
    - Specify a new password and repeat that in the "verify password" box (this is a temporary password and you will be requested to change it on login.)
    - Leave everything else empty.
    - Click "Add and Edit" at the bottom.

3. Click User Groups, "+ Add" and add the "sysadmin" group to the user. Click "Add".

4. Repeat the above 1-3 for each user in your team.

You should now be able able to log into your user accounts and access root. Please use your own account when interacting with the cluster going forward. **You may need to repeat the ssh keypair-based authentication for your users in order to log in to the compute node without a password, test it first.**