# Linux: Demo

Prerequisites: 1. Ubuntu Server 17.04. 2. 'root' User. 3. 'demo' User (No root Permissions). 4. Configure SSH keys. 5. Login with 'demo' user.

In this document we'll cover:

1. Reading Linux Documentation
2. Connecting to a Linux Machine
   1. SSH Cients
   2. Using passwords
   3. Using SSH keys
3. Installing Redis on a Ubuntu Server
4. Configure Redis as a Service
5. Installing NGINX on a Ubuntu Server
6. Create a New Linux User
7. Create SSH Key Pair
8. Switching User in Linux

## Reading Documentation

A good place to find documentation on Linux Commands is at Linux Command Reference Index.

The most common Linux Command format is:

```
[command] [options] [arguments]
```

For example:

```
ls --all /opt/
```

Where:

Command: `ls`

Options: `--all`

Arguments: `/opt/`

Sometimes you may encounter documentation where the command is in the following format:

```
curl [options] -u <username>:<password> <url>
```

Note that there are two different brackets, <> and [].

`<username>` means replace this with a username.

`<password>` means replace this with a password.

`<url>` means replace this with a url.

`[options]` means replace this with a valid option of this command.

# Connecting to a Machine

SSH stands for Secure Shell.

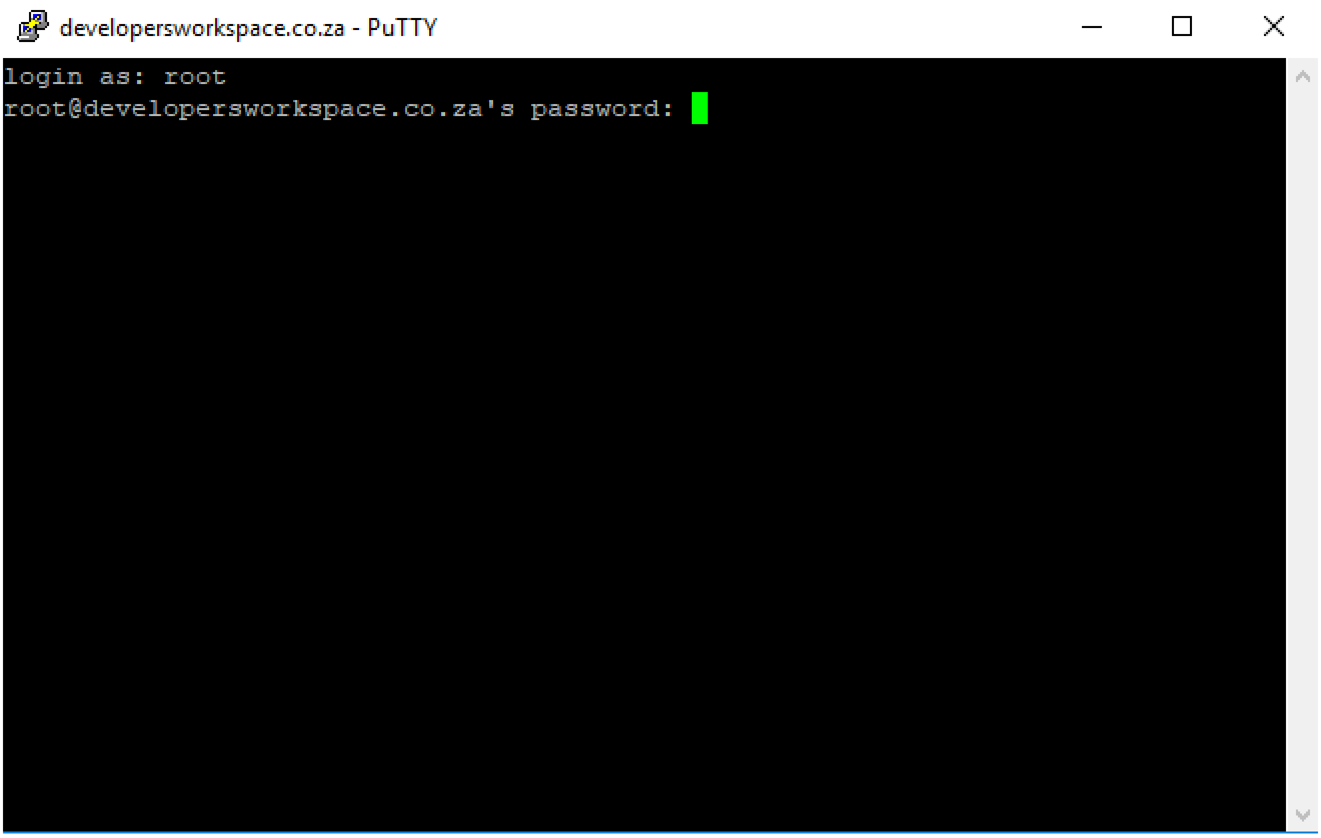
Uses encryption to secure data with one of the follow ciphers:

- aes128-ctr
- aes192-ctr
- aes256-ctr MACs
    - hmac-sha1
    - umac-64@openssh.com
    - hmac-ripemd160
    - hmac-sha1-96
    - hmac-sha2-256
    - hmac-sha2-512
    - hmac-sha2-512-96

## SSH Clients:

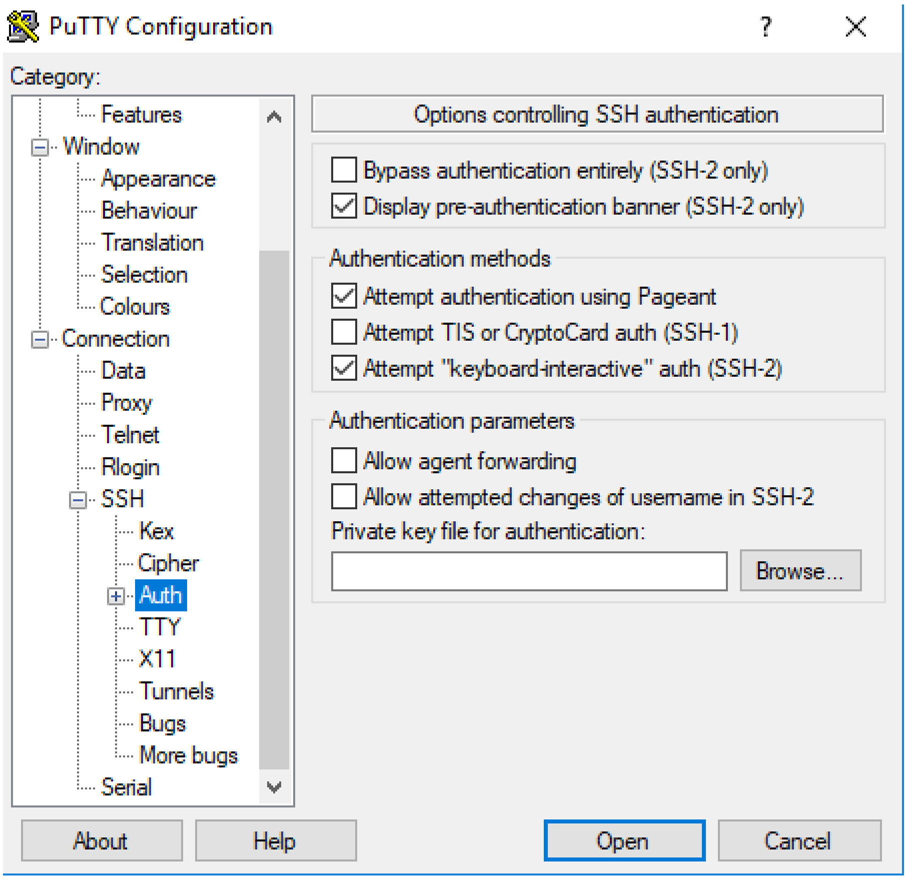
- Putty
- SuperPutty
- Kitty
- SmarTTY
- mRemoteNG
- Terminals

## Using passwords



```
login as: root
root@developersworkspace.co.za's password:
```

## Using SSH keys



Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)

☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

# Installing Redis

`sudo` stands for 'superuser do'

`sudo apt-get update` downloads the package lists from the repositories and "updates" them to get information on the newest versions of packages and their dependencies.

`sudo apt-get install build-essential` installs all the packages needed to compile a debian package. Includes the gcc/g++ compilers.

`sudo apt-get install tcl8.5` installs TCL which is a very powerful dynamic programming language.

`wget http://download.redis.io/releases/redis-stable.tar.gz` downloads the Redis source to our **current/working** directory.

`tar xzf redis-stable.tar.gz` unzip/untar the downloaded file.

We can use `ls` to see the downloaded files.

`cd redis-stable` changes to 'redis-stable' directory.

`make` compiles the source code using the gcc/g++ compilers.

`make install` installs the compiled source by running a TCL script.

Redis is now installed. We can test this by running `redis-cli` which should give us an error as their is no instance of Redis running.

To run a Redis instance we need a configuration file which we can create by using 'nano' editor.

`sudo mkdir /etc/redis` create the directory.

`sudo nano /etc/redis/6379.conf` opens/creates the file at '/etc/redis/6379.conf' in 'nano'.

If we just wanted to create a file without opening it we could have used `touch /etc/redis/6379.conf`.

`redis-server /etc/redis/6379.conf` run Redis Server with the config as a parameter.

## Configure Redis as a Service

In this example we'll be create systemd services.

'systemd' works with unit file which is what the service configuration file is called.

These files contains configurations, such as:

- Description.
- When to start, before or after the network service.
- Type of service.
- What executable to execute.

```
[Unit]
Description=Redis Server

[Service]
User=root
Group=root
ExecStart=/usr/local/bin/redis-server /etc/redis/6379.conf
Restart=always

[Install]
WantedBy=multi-user.target
```

When this file is created in `/etc/systemd/system` as 'my-redis.service', it's ready to be executed.

To start a service we can run:

```
sudo systemctl start my-redis
```

and to stop a service:

```
sudo systemctl stop my-redis
```

## Installing NGINX

1. `sudo apt-get update`
2. `sudo apt-get install nginx`
3. `sudo ufw allow 'Nginx HTTP'`

## Create a New User

```
sudo useradd -m <username>
```

```
sudo passwd <username>
```

## Switching User

```
su - <username>
```