How To Install Apache2 (LAMP) Ubuntu 18.04

How to install apache2 (LAMP) server

LAMP is a acronym of the names in the applications stack, Linux OS, Apache server, MySQL database and PHP programming language.

Together they build a framework to run web applications and host sites like WordPress , all applications in the stack are open source and released on most Linux distributions.

Requirements

1. Ubuntu 18.04 LTS
2. SSH access to the server (Setup SSH)
3. A non root user with sudo privileges (Add sudo user)
4. Enabled firewall (Setup ufw)

5. Configured hostname (Setup hostname)
6. DNS entries

Step 1: Install Apache2

1.1 Lets start by updating the repository's and software packages.

```
sudo apt update
sudo apt upgrade -y
```

1.2 Install the apache2 package.

```
sudo apt install apache2 -y
```

1.3 Confirm installation.

```
sudo systemctl status apache2
```

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2019-06-19 19:08:58 UTC; 3min 21s ago
 Main PID: 7685 (apache2)
    Tasks: 55 (limit: 2213)
   CGroup: /system.slice/apache2.service
           ├─7685 /usr/sbin/apache2 -k start
           ├─7878 /usr/sbin/apache2 -k start
           └─7879 /usr/sbin/apache2 -k start
```

```
Jun 19 19:08:48 iphone systemd[1]: Starting The Apache HTTP Server...
Jun 19 19:08:58 iphone apachectl[7660]: AH00558: apache2: Could not reliably det
Jun 19 19:08:58 iphone systemd[1]: Started The Apache HTTP Server.

toor@iphone:~$
```

Step 2: Configure Firewall

2.1 Add firewall rules for Apache.

```
sudo ufw allow in "Apache Full"
```

```
Rule added
Rule added (v6)
toor@iphone:~$
```

2.2 Display firewall rules and confirm that the firewall is configured

```
sudo ufw status
```

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
```

```
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)

toor@iphone:~$
```

2.3 Confirm that you can browse to the site

```
http://your_server_ip
```

Step 3: Create the Directory Structure

3.1 Virtual host enables us to have multiple websites on one server, each website can have its own home folder “document root” and a unique SSL certificate ,we can have different security policies for each site, and much more.

Create the directory structure.

```
/var/www/
├── Domain-1.local
│ └── html
├── Domain-2.local
│ └── html
```

Before we create the directory for the site, make sure to configure hostname and hosts file.

I will create a website for my local lab domain ceh.local, the /etc/hots file should the the fallowing entry in it.

```
YOUR-IP-ADDRESS ceh.local
```

In this example i am creating a virtual hosts directory called “ceh.local” and i am using the -p flag to create parent directories.

```
sudo mkdir -p /var/www/ceh.local/html
```

3.2 Assign ownership of the directory to current user

```
sudo chown -R $USER:$USER /var/www/ceh.local/html
```

3.3 Set directory permissions

```
sudo chmod -R 755 /var/www/ceh.local/html
```

3.4 Create a sample index.html file using your favorite editor and add it to the root directory.

```
sudo nano /var/www/ceh.local/html/index.html
```

Add the html code bellow

```
<!DOCTYPE html>
```

```
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Welcome to ceh.local</title>
  </head>
  <body>
    <h1>Success! ceh.local home page</h1>
  </body>
</html>
```

Exit & Save

Step 4: Configure Virtual Hosts File

Apache virtual hosts configuration files are stored in.

- /etc/apache2/sites-enabled
- /etc/apache2/sites-available

Lets add a Virtual Hosts configuration file for domain1.local

```
sudo nano /etc/apache2/sites-available/ceh.local.conf
```

Add the fallowing lines and modify them to your site.

```
<VirtualHost *:80>
    ServerName ceh.local
    ServerAlias www.ceh.local
    ServerAdmin admin@ceh.local
    DocumentRoot /var/www/ceh.local/html

    <Directory /var/www/ceh.local/html>
        Options -Indexes +FollowSymLinks
```

```
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/ceh.local-error.log
    CustomLog ${APACHE_LOG_DIR}/ceh.local-access.log combined
</VirtualHost>
```

Exit & Save

4.2 Enable the Virtual Hosts configuration file file with a2ensite

```
sudo a2ensite ceh.local.conf
```

```
Enabling site domain1.local.
To activate the new configuration, you need to run:
  systemctl reload apache2
toor@iphone:~$
```

4.3 Disable the default site defined in 000-default.conf

```
sudo a2dissite 000-default.conf
```

```
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
toor@iphone:~$
```

4.4 Test the configuration file for any syntax errors

```
sudo apache2ctl configtest
```

```
Syntax OK
toor@iphone:/var/www/domain1.local/html$
```

4.5 Restart the Apache service for the changes to take effect

```
sudo systemctl restart apache2
```

4.6 Confirm that the service have started

```
sudo systemctl status apache2
```

4.7 launch a web browser and start browsing ceh.local

Step 5: Install MySQL

5.1 To install MySQL run

```
sudo apt install mysql-server -y
```

5.2 Verify that MySQL service is running

```
sudo systemctl status mysql
```

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Fri 2019-06-21 11:59:29 UTC; 8s ago
 Main PID: 17807 (mysqld)
    Tasks: 27 (limit: 2322)
   CGroup: /system.slice/mysql.service
           └─17807 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi

Jun 21 11:59:28 srv6 systemd[1]: Starting MySQL Community Server...
Jun 21 11:59:29 srv6 systemd[1]: Started MySQL Community Server.
lines 1-10/10 (END)
toor@srv6:~$
```

5.3 The default MySQL user “root” have a blank password., we need to secure the MySQL server and remove the default database.

```
sudo mysql_secure_installation
```

Then enter the following security questions

- VALIDATE PASSWORD plugin = NO
- Set root password and confirm
- Remove anonymous users? = YES
- Disallow root login remotely? = NO
- Remove test database and access to it? = YES
- Reload privilege tables now? = YES

5.4 Start from MySQL Server 5.7, if you do not provide a password to root user during the installation, it will use auth_socket plugin for authentication.

If we want to configure a password authentication, we need to run the following commands.

```
sudo mysql
```

5.5 Display current configuration

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

5.6 Alter authentication_string for the root user

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'changeme';
```

5.7 Flush the privileges and update the changes

```
FLUSH PRIVILEGES;
```

5.8 Display current configuration

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

5.9 Exit from the mysql prompt:

```
exit
```

Step 6: Install PHP

6.1 PHP is a server side scripting language used to generate dynamic content on websites and applications.

Install PHP (default version is PHP 7.2) and some of the basic modules for web deployments.

```
sudo apt install php php-common php-mysql php-gd php-cli -y
```

6.2 Create info.php file in the Apache root document folder.

Usually, the apache2 root document folder will be /var/www/html/ or /var/www/ in most Debian based Linux distributions.

If you have followed the guide then the the file should be in /var/www/ceh.local/html/

```
sudo nano /var/www/ceh.local/html/info.php
```

Add the following lines

```
<?php
phpinfo();
?>
```

Exit and save

6.3 Restart Apache

```
sudo systemctl restart apache2
```

6.4 Test PHP page, open a web browser and enter “http://ceh.local/info.php”

Step 7: Install PhpMyAdmin

7.1 With phpMyAdmin we can administrating MySQL from a web browser, start by adding the needed repository.

```
sudo add-apt-repository universe
```

7.2 Install phpmyadmin

```
sudo apt install phpmyadmin -y
```

Go through the package installation process, select Apache2 and configure a password for the phpmyadmin database.

7.3 Restart Apache

```
sudo systemctl restart apache2
```

Conclusion

We have installed installing Apache2, MySQL, PHP and Virtual Hosts on a Ubuntu server and secured it.

For administration of the website we have installed phpmyadmin.

How To Install a FTP Server On Ubuntu Server 18.04

VsFTPD ” Very Secure FTP Daemon”

VsFTPD “very secure FTP daemon” is an open source FTP server for Linux systems, in this quick guide we will install VsFTPD on a Ubuntu server and secure the FTP server with SSL/TLS. Please visit the official website of VsFTPD if you need more information about the application.

Requirements

- Ubuntu Server 18.04
- User with sudo privileges.
- Static IP address
- Configured firewall
- Server connected to internet

For more information on how to create a sudo and configure a static IP please see the quick guides Create Sudo User , Set Static IP address and Configure Ubuntu Firewall.

Install VsFTPD

Vsftpd is available in Ubuntu 18.04 default repository and do not require any extra pre configuration.

Run the following command to install Vsftpd

```
sudo apt-get install vsftpd -y
```

Wait for the application to finish installing, start the Vsftpd service and enable it to start on boot.

```
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
```

Verify that the VsFTPD is up and running.

```
sudo systemctl status vsftpd
```

```
root@iphone:~# sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset:
enabled)
   Active: active (running) since Sat 2019-06-08 18:17:39 UTC; 2min 54s ago
 Main PID: 2311 (vsftpd)
    Tasks: 1 (limit: 2214)
   CGroup: /system.slice/vsftpd.service

Jun 08 18:17:39 iphone systemd[1]: Starting vsftpd FTP server...
Jun 08 18:17:39 iphone systemd[1]: Started vsftpd FTP server.
```

Configure The Firewall

We need to open port 20 and 21 for active FTP and ports 40000-50000 for passive FTP.

```
sudo ufw allow 20/tcp
```

```
sudo ufw allow 21/tcp
```

```
sudo ufw allow 40000:50000/tcp
```

Display the firewall rules.

```
sudo ufw status
```

```
root@iphone:~# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)

root@iphone:~#
```

Create FTP User

Create a low privileges user that can be used to access the FTP server.

When prompted enter password and user information for the user.

```
sudo adduser ftpuser
```

```
root@iphone:~# sudo adduser ftpuser
Adding user `ftpuser' ...
Adding new group `ftpuser' (1001) ...
Adding new user `ftpuser' (1001) with group `ftpuser' ...
Creating home directory `/home/ftpuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ftpuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@iphone:~#
```

Create a FTP Directory For The FTP User

First we want to create a FTP folder for the ftpuser.

```
sudo mkdir /home/ftpuser/ftp
```

Next we want to set the folder ownership.

```
sudo chown nobody:nogroup /home/ftpuser/ftp
```

Remove write permissions to the ftp folder.

```
sudo chmod a-w /home/ftpuser/ftp
```

Verify FTP folder permissions.

```
sudo ls -la /home/ftpuser/ftp
```

```
root@iphone:/home/ftpuser# sudo ls -la /home/ftpuser/ftp
total 8
dr-xr-xr-x 2 nobody  nogroup 4096 Jun  8 19:01 .
drwxr-xr-x 3 ftpuser ftpuser 4096 Jun  8 19:02 ..
root@iphone:/home/ftpuser#
```

Create a directory for file uploads and assign ownership to ftpuser.

```
sudo mkdir /home/ftpuser/ftp/files
sudo chown ftpuser:ftpuser /home/ftpuser/ftp/files
```

Verify the new folder permission.

```
sudo ls -la /home/ftpuser/ftp
```

```
root@iphone:/home/ftpuser/ftp/files# sudo ls -la /home/ftpuser/ftp
total 12
dr-xr-xr-x 3 nobody  nogroup 4096 Jun  8 19:08 .
drwxr-xr-x 3 ftpuser ftpuser 4096 Jun  8 19:02 ..
drwxr-xr-x 2 ftpuser ftpuser 4096 Jun  8 19:08 files
```

```
root@iphone:/home/ftpuser/ftp/files#
```

Create and add txt file to the files folder we created in the step above.

```
echo "Test create txt file" | sudo tee /home/ftpuser/ftp/files/txt01.txt
```

Configuring VsFTPD

Edit the VsFTPD configuration file vsftpd.conf

```
cd etc/
sudo nano vsftpd.conf
```

```
##
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
##
```

Enable uploading to the FTP server by uncomment the write_enable parameter.

```
##
write_enable=YES
##
```

Prevent the FTP users from accessing files or to run commands outside there directory by uncomment the chroot_local_user=YES parameter.

```
##
chroot_local_user=YES
##
```

Scroll down to the bottom and add the the port range for passive FTP.

```
pasv_min_port=40000
pasv_max_port=50000
```

Previously we created a ftp/file directory and folder for the ftpsuser, now we need to configure VsFTPD to log the ftpuser to home ftp directory we created.

Add the line bellow.

```
user_sub_token=$USER
local_root=/home/$USER/ftp
```

Restart the daemon.

```
sudo systemctl restart vsftpd
```

Testing The FTP Access

You can use a ftp client like FileZilla or the command line to confirm that you can access the ftp server and that you can see the txt file you created in the ftpuser ftp directory.

I am using the command line on the FTP server in this example to confirm that i can access the FTP and that i can download the txt01.txt.

```
root@iphone:/# ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 3.0.3)
Name (127.0.0.1:toor): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Lets confirm that we can change to the “files” directory.

```
ls
cd files
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1001     1001         4096 Jun 08 19:17 files
226 Directory send OK.
ftp> cd files
250 Directory successfully changed.
ftp>
```

List the directory and use the get command to transfer the test file.

```
ls
get txt01.txt
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0              21 Jun 08 19:16 txt01.txt
226 Directory send OK.
ftp> get txt01.txt
local: txt01.txt remote: txt01.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for txt01.txt (21 bytes).
226 Transfer complete.
21 bytes received in 0.00 secs (259.5926 kB/s)
ftp>
```

Upload the file with a new name to test users write permissions. To upload a file we use the put command.

```
put txt01.txt txt01-upload.txt
```

```
ftp> put txt01.txt txt01-upload.txt
local: txt01.txt remote: txt01-upload.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
21 bytes sent in 0.00 secs (1.0541 MB/s)
ftp>
```

Listing the files directory should show two files now.

```
ls
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-------    1 1001     1001           21 Jun 08 21:00 txt01-upload.txt
-rw-r--r--    1 0        0              21 Jun 08 19:16 txt01.txt
226 Directory send OK.
ftp>
```

(Optional) Secure The FTP Server With TLS

Lets start adding the firewall rule for TLS traffic, add port 990 to the firewall access list.

```
sudo ufw allow 990/tcp
```

```
root@iphone:/# sudo ufw allow 990/tcp
Rule added
Rule added (v6)
root@iphone:/#
```

Confirm firewall status

```
sudo ufw status
```

```
root@iphone:/# sudo ufw status
```

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
21/tcp                     ALLOW       Anywhere
40000:50000/tcp            ALLOW       Anywhere
990/tcp                    ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
21/tcp (v6)                ALLOW       Anywhere (v6)
40000:50000/tcp (v6)       ALLOW       Anywhere (v6)
990/tcp (v6)               ALLOW       Anywhere (v6)

root@iphone:/#
```

Create a OpenSSL certificate

Create a OpenSSL certificate for TLS/SSL encryption, first make a directory where you can save the certificate.

```
sudo mkdir /etc/ftpcert
```

Now we will create a new certificate, use the -days flag to make it valid for two years, 730 days. Next set the bit value of the RSA key, i am running with a 2048-bit RSA key.

Type in the -keyout and the -out flag, the flags will set the key values for the private key and the certificate.

NOTE: Setting both flags with the same value will create both the private key and the certificate in the same file.

You will be asked to enter details like country, state, etc. You don’t have to fill in the information. Just keep pressing ENTER for defaults.

```
sudo openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout
/etc/ftpcert/vsftpd.pem -out /etc/ftpcert/vsftpd.pem
```

Confirm that the private key and the certificate is the ftpscert directory.

```
root@iphone:/# cd /etc/ftpcert/
root@iphone:/etc/ftpcert# ls
vsftpd.pem
root@iphone:/etc/ftpcert#
```

Next we need to configure vsftpd to allow TLS/SSL traffic and point out the directory of the private key and the certificate , open the vsftpd configuration file with a editor.

```
cd etc/
sudo nano vsftpd.conf
```

Scroll down until you find the rsa parameters, Comment them out and replace them with new lines that points out the privet key and the certificate we created.

```
##
# rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
# rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
##

rsa_cert_file=/etc/ftpcert/vsftpd.pem
rsa_private_key_file=/etc/ftpcert/vsftpd.pem
```

Configure FTP connections to use use SSL/TLS, change the ssl_enable=NO parameter to YES.

```
##
ssl_enable=YES
##
```

Now add the following lines to deny anonymous connections over SSL and to require SSL for logging and transferring data.

```
##
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
##
```

Configure the server to use the TLS protocol

```
##
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
##
```

Last configure SSL reuse parameter to NO due that it can have conflicts with FTP clients, next we need to use high encryption cipher suite, which means that the key lengths is equal to or greater than 128 bits.

Paste thee lines below.

```
##
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

```
##
```

The configuration should have the below entry’s configured.

```
#
rsa_cert_file=/etc/ftpcert/vsftpd.pem
rsa_private_key_file=/etc/ftpcert/vsftpd.pem
#
#
ssl_enable=YES
#
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
##
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
#
require_ssl_reuse=NO
ssl_ciphers=HIGH
#
```

Restart the VsFTPD to load the new configuration.

```
sudo systemctl restart vsftpd
```

Confirm FTP TLS Configuration

Download a FTP client like FileZilla, you grab the FileZilla client from the official site https://filezilla-project.org/

Run and install the FTP client, when connecting to the FTP server use “Require explicit

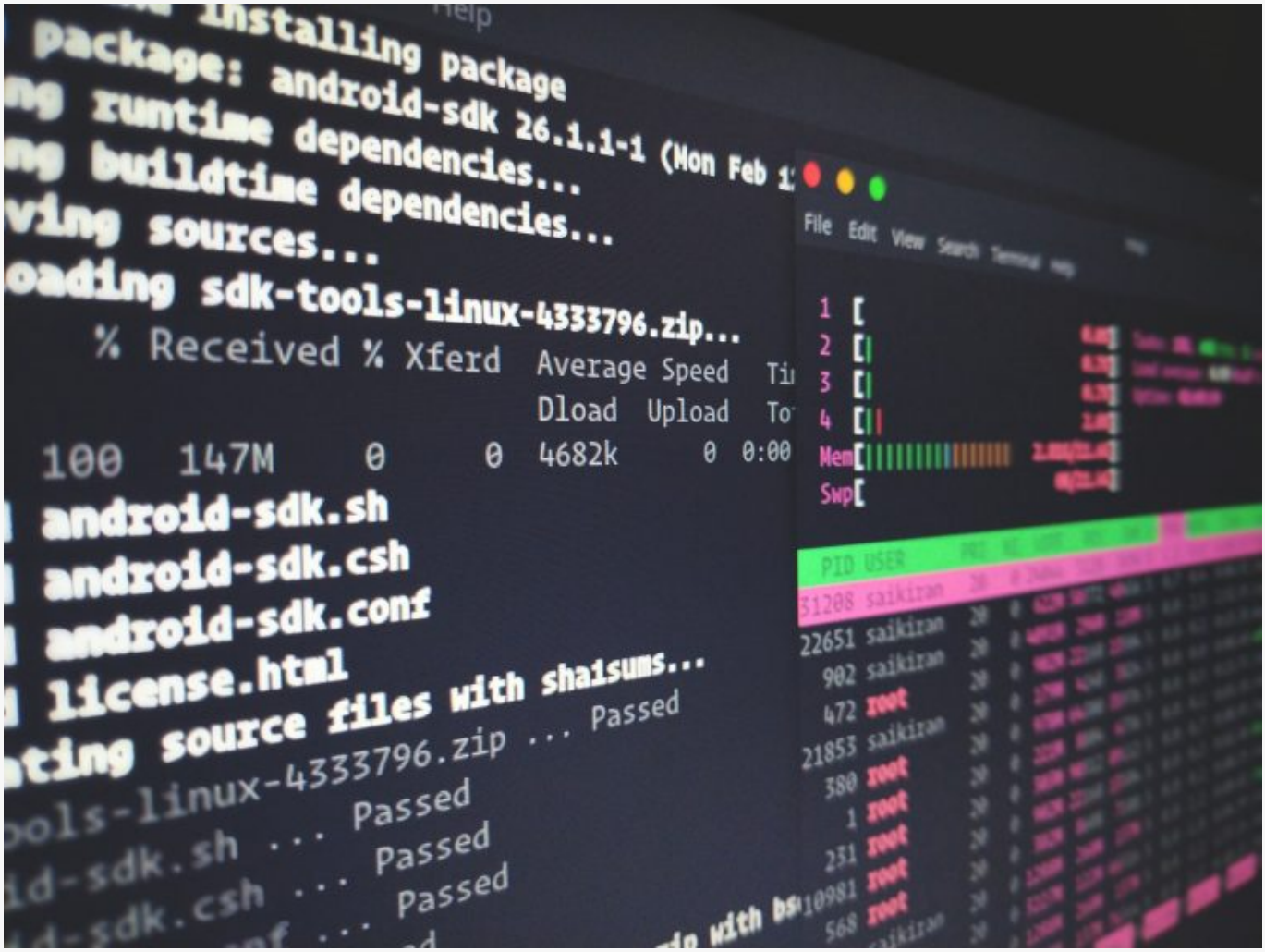
FTP over TLS”. If everything is configured correct you should be grated with a pop up windows that displays the server certificate we created.

If you try to connect to the FTP server with just plain FTP protocol, you will get an error and you wont be able to connect to the server.

```
Status: Connection established, waiting for welcome message...
Response:       220 (vsFTPd 3.0.3)
Command:        USER ftpuser
Response:       530 Non-anonymous sessions must use encryption.
Error:  Could not connect to server
```

Conclusion

In this quick guide we have installed a FTP server on Ubuntu18.04.02, generated a certificate with OpenSSL and secured the server connectivity with TLS.



How To Configure NTP Server Ubuntu 18.04 Bionic Beaver

Step 1: Configure NTPd Server

1.1 Run commands in root

1.2 Disable timesyncd

1.2.a Verify that timesyncd is disabled

1.3 Edit timesyncd.conf

1.3.a Remove hashtag from NTP statement

1.3.b Add your NTP server pool address

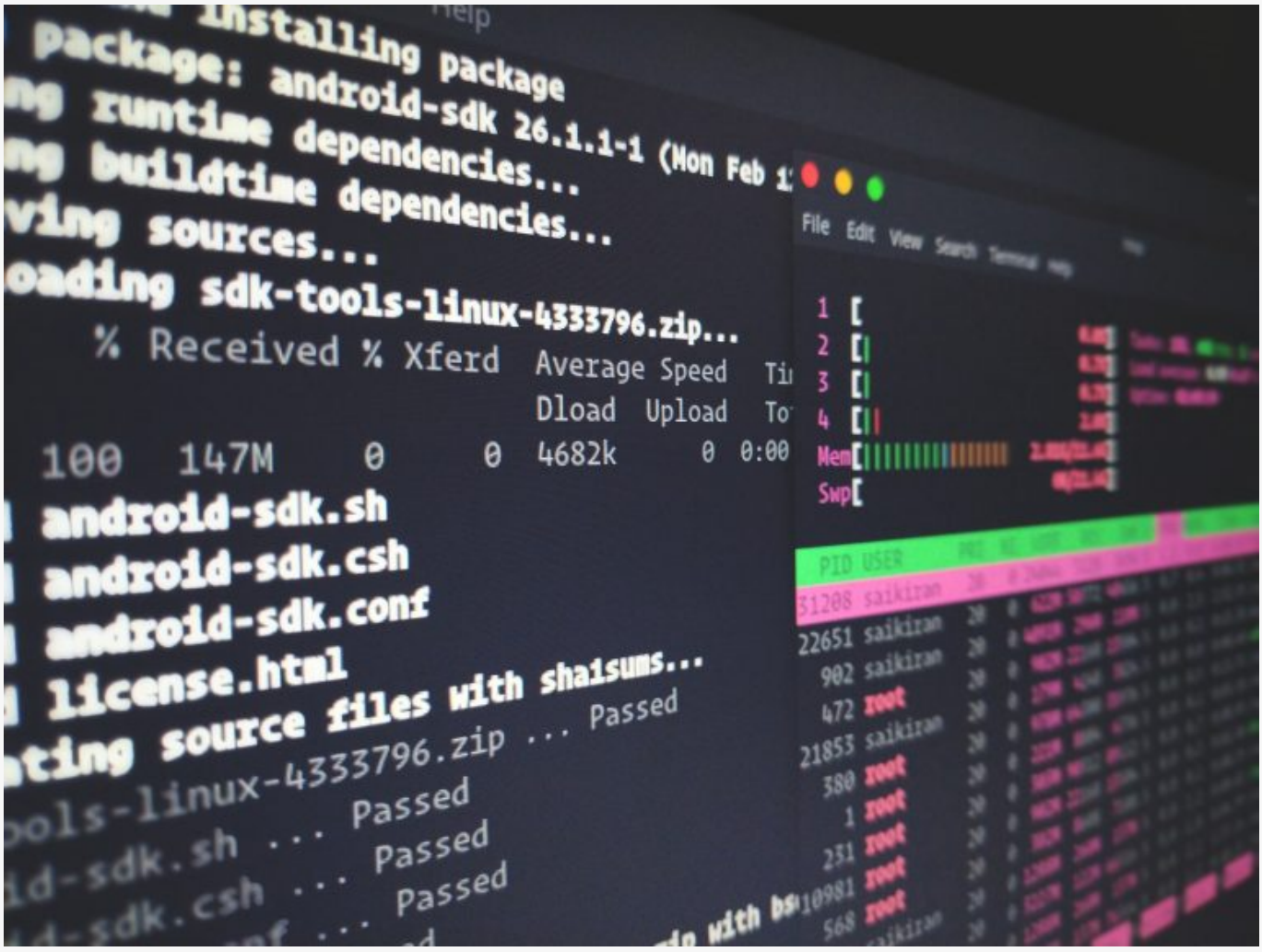
Exit and Save

1.4 Restart systemd-timesyncd

1.4.a Verify ntp pool change

Step 2: Allow NTPd traffic in firewall

2.1 Allow NTPd traffic in firewall



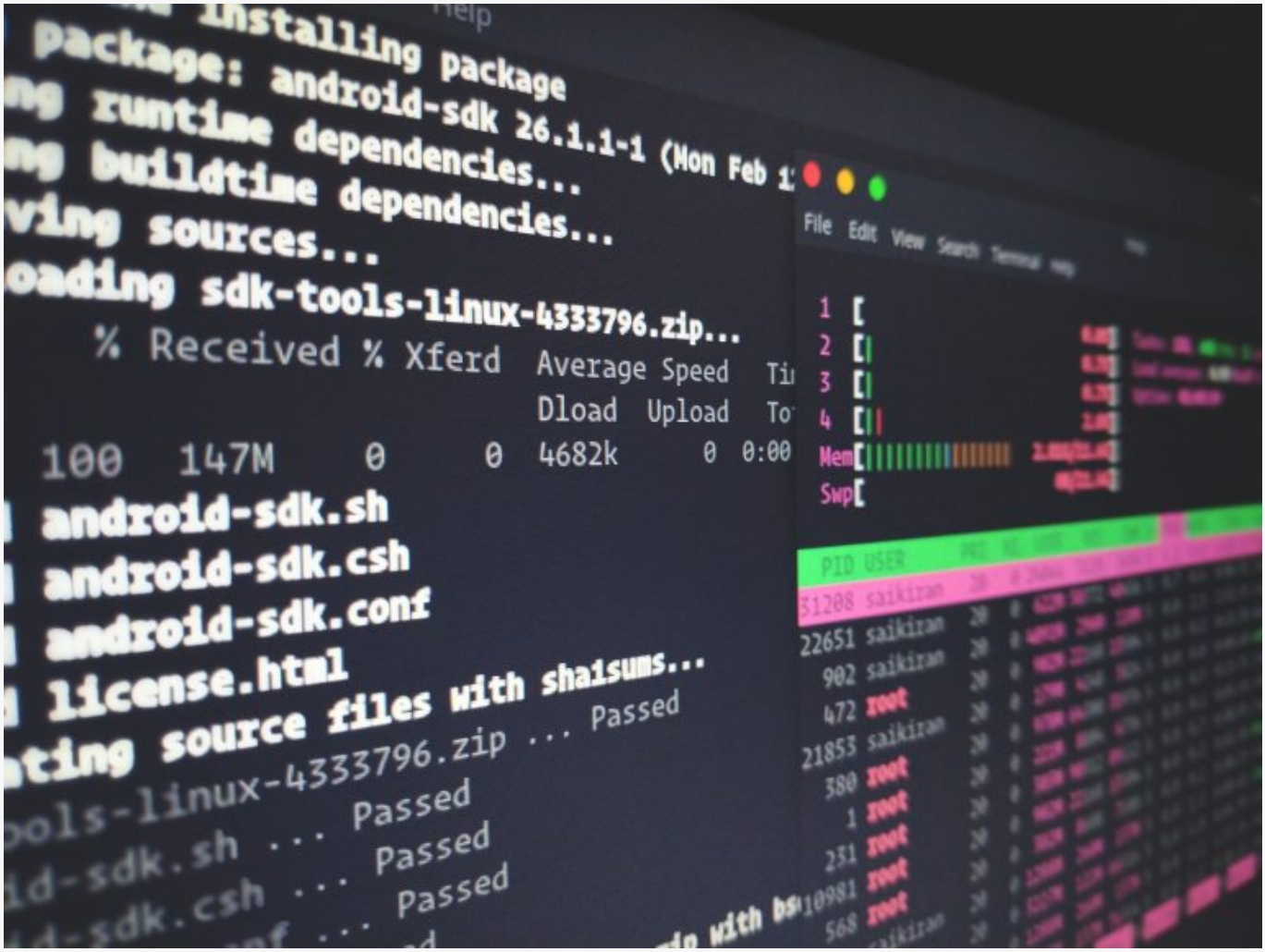
How To Configure Time Zone Ubuntu 18.04 Bionic Beaver

Step 1: Set and configure time zone

1.1 List available time zones

1.2 Set the time zone

1.3 Verify time and time zone



How To Update/Upgrade Ubuntu 18.04 Bionic Beaver

Step 1: Edit the repository config file

1.1 Edit Sources list

1.1.a Add the fallowing repos

Exit and Save

Step 2: Update/Upgrade

2.1 Update installed packages

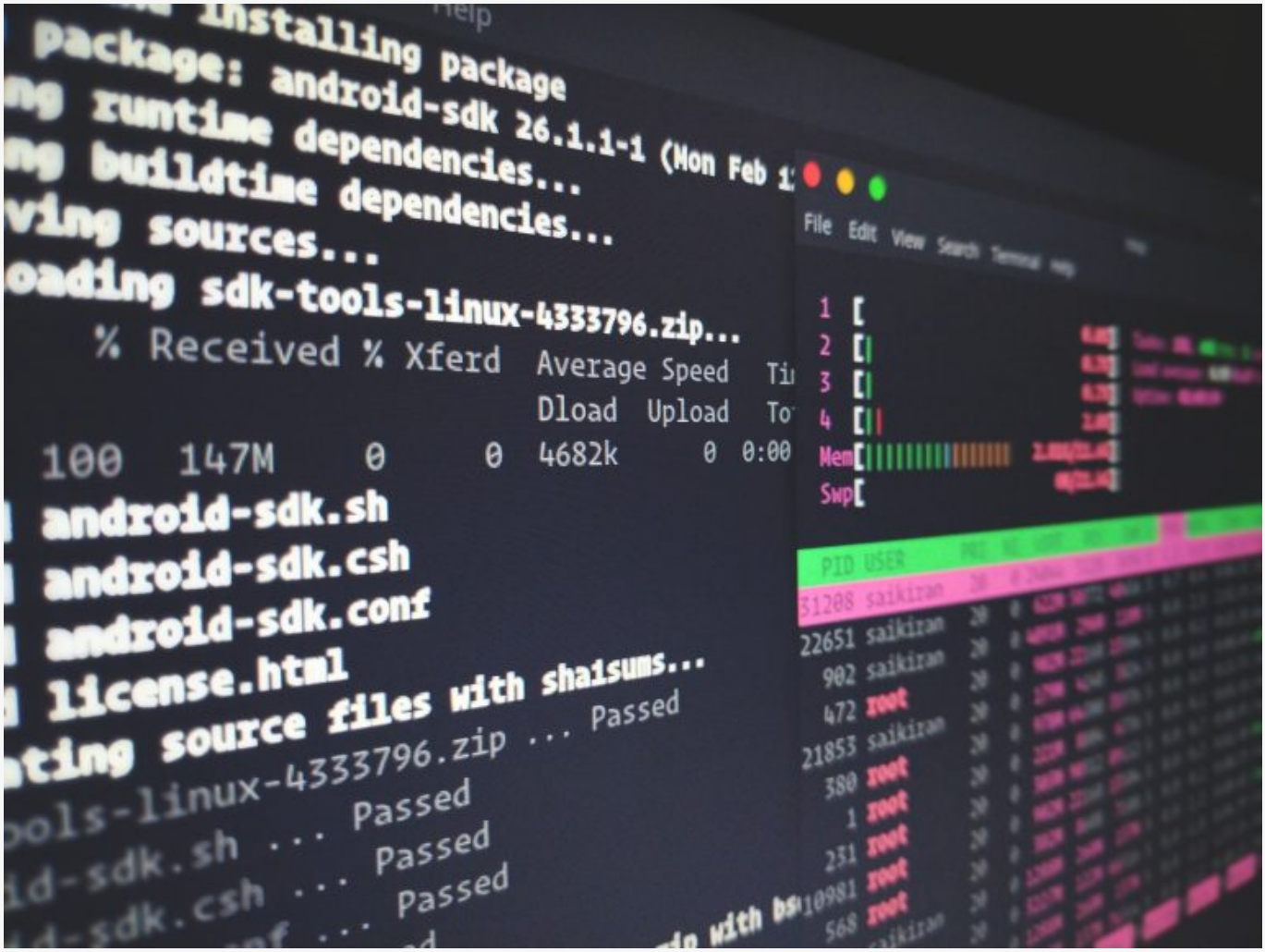
2.2 Upgrade installed packages to the latest versions

2.3 Update dependencies

2.4 Auto-remove old files

Step 3: Reboot

3.1 Reboot the server



How To Set Hostname Ubuntu 18.04 Bionic Beaver

Step 1: Change hostname

1.1 Display hostname

1.1.a Set hostname

Step 2: Verify parameter change

2.1 Verify hostname change

2.2 Verify /etc/hostname

Exit

Step 3: Set static table lookup for hostname

3.1 Edit /etc/hosts

3.1.a Change the old hostname

Exit and Save

3.2 Check if the “cloud.cfg” is installed (This part can be skipped if the file is missing)

3.2.a Edit cloud.cfg

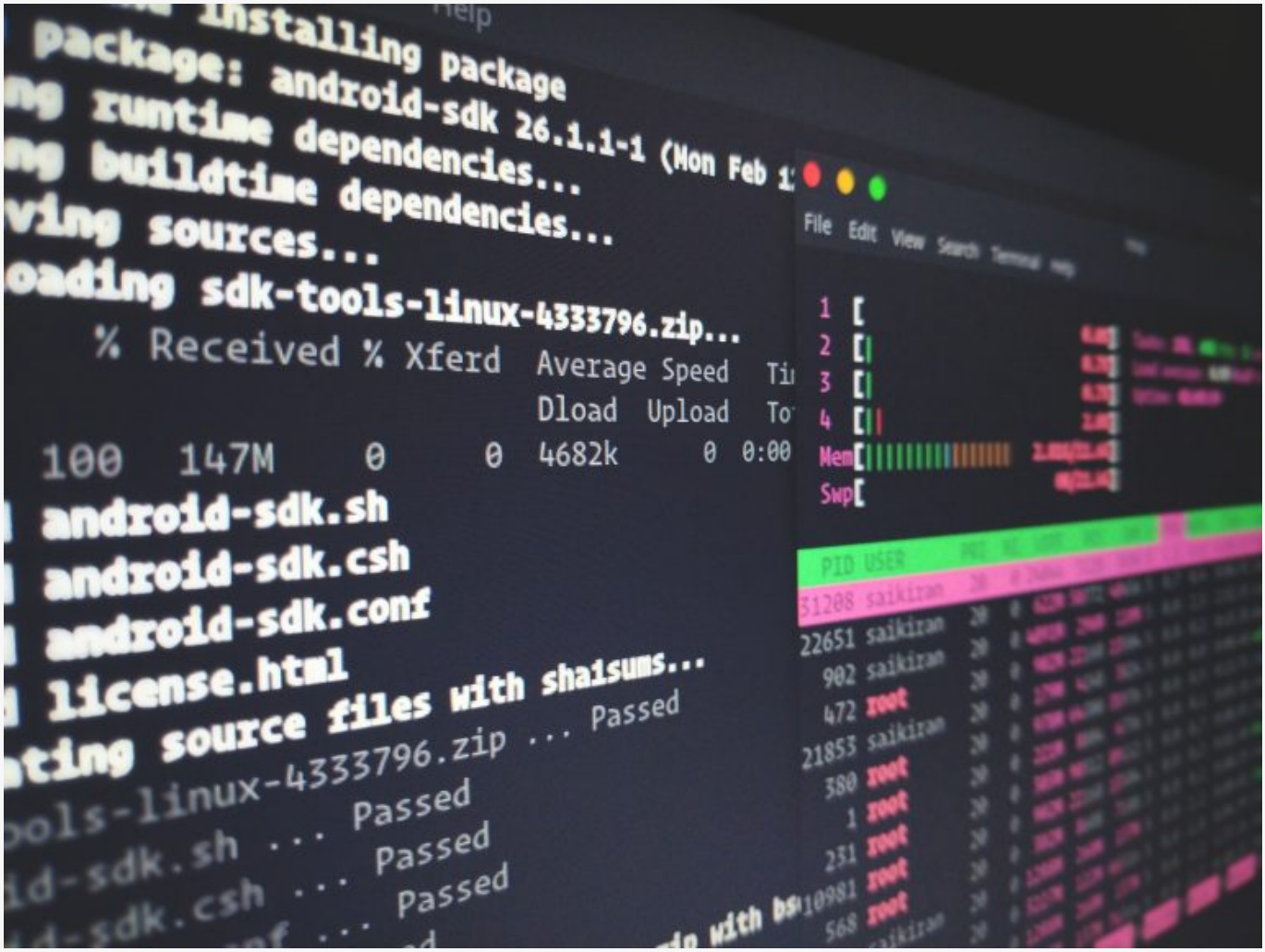
3.2.b Change the preserve_hostname value from false to true

Exit and Save

Step 4: Verify with a reboot

4.1 Reboot to verify change

4.2 Verify hostname



How To Configure Firewall Ubuntu 18.04 Bionic Beaver

Step 1: Configure firewall (UFW)

1.1 Run commands in root

1.1.a Enable the firewall

Type “Y” to proceed

1.1.b Deny incoming traffic

1.1.c Allow outgoing traffic

1.2 Allow default SSH connection on port 22

NOTE: If you are using a different port then use the statement below

1.3 Check firewall status

Step 2: Deleting rules

2.1 Determine firewall rule

2.2 Delete rule

2.2.a Repeat for ipv6 rule

Useful firewall rules

Enable HTTP

Enable HTTPS

Deny HTTP

Allow specific range of TCP ports

Allow specific range of UDP ports

Allow specific IP Addresses

Allow specific IP Addresses from a subnet

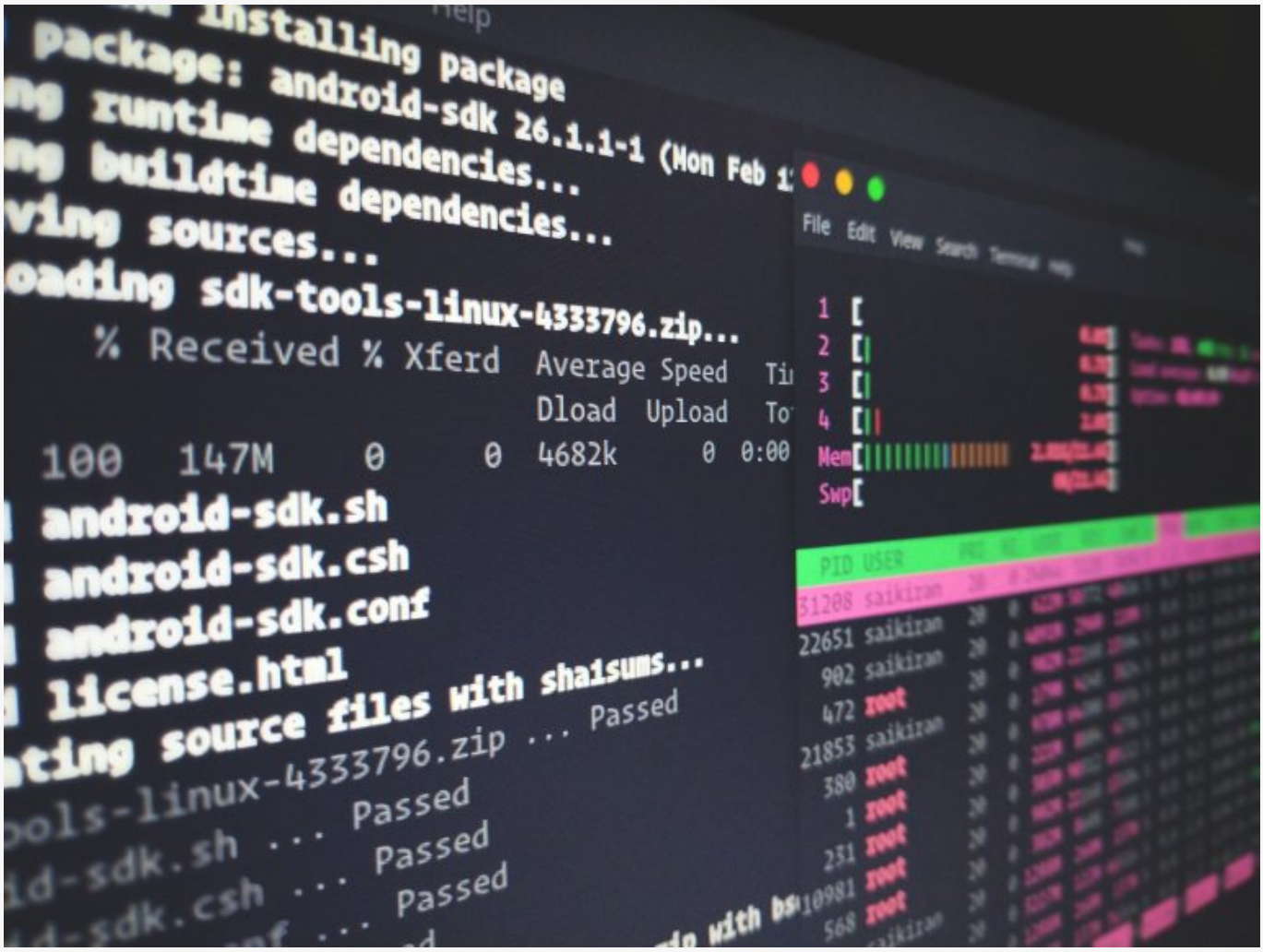
Allow specific IP Addresses and port

Allow specific IP Addresses from a subnet and port

Allow trafic to a specific network interface

Reload firewall rules

Restart UFW



How To Install SSH Server Ubuntu 18.04 Bionic Beaver

Step 1: Install & Configure SSH Server

1.1 Install SSH server

1.1.a Edit SSH server configuration file

1.1.b Remove the hashtag from the port statement

Exit and Save

1.2 Restart SSH daemon

1.3 Verify status

SSH server is read to be used on port 22

Step 2: Configure SSH Server

2.1 Change SSH Server port

2.1.a Change the port number

Exit and Save

2.1.b Restart the SSH service

2.1.c Verify status

SSH server is ready to be used on port 888

2.2 Disable the option to login in directly as root via SSH

2.2.a Change the PermitRootLogin parameter to no

Exit and Save

2.2.b Restart the SSH service

Useful Commands

SSH Service status

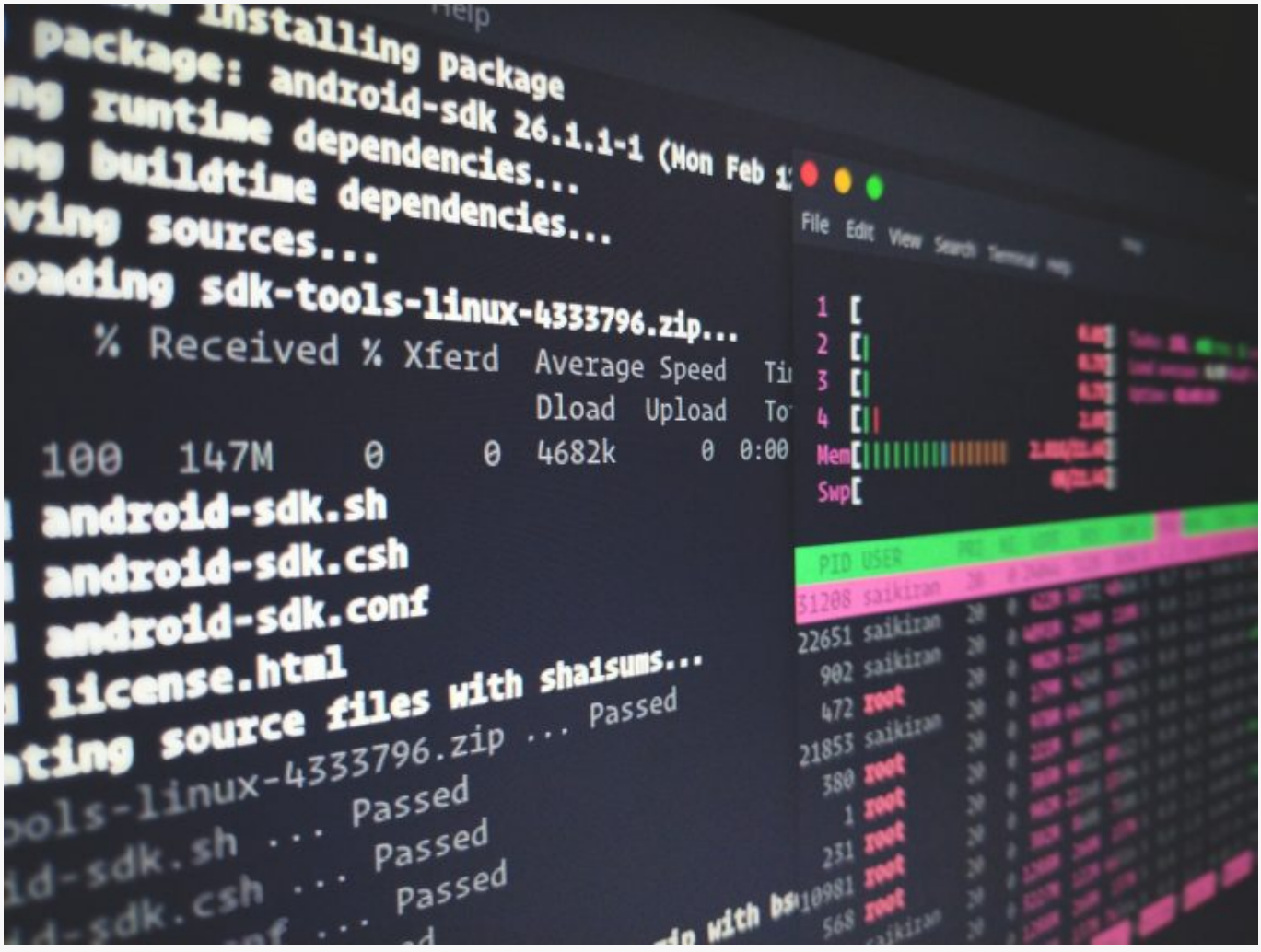
Restart SSH service

Stop SSH service

Start SSH service

Disable SSH service next boot

Enable SSH service next boot



How To Static IP Ubuntu 18.04 Bionic Beaver

Step 1: Set static IP address

1.1 List Netplan to see name of the configuration file

1.2.a Open the 50-cloud-init.yaml configuration file

1.2.b Edit the configuration file with your IP network parameters

Exit and Save

Step 2: Apply Netplan configuration

2.1 Apply settings

2.2 Optional: Debug the netplan apply command

2.3 Reconnect to the server with new IP

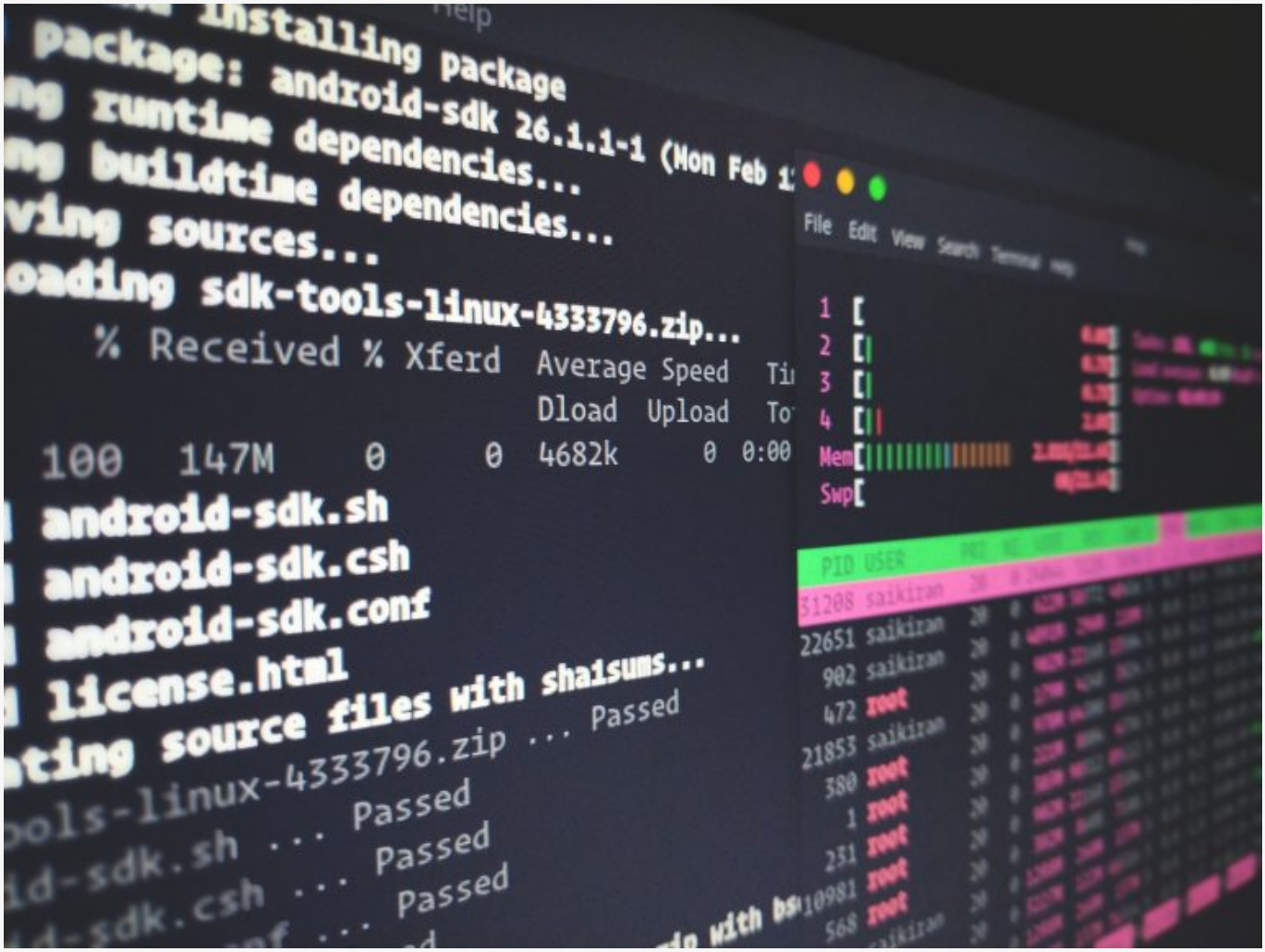
Step 3: Verify Connectivity

3.1 Check IP address

3.2 Check default route

3.3 Check internet connectivity

3.4 Check DNS



How To Create Root Account Ubuntu 18.04 Bionic Beaver

Step 1: Create a user

1.1 Run commands in root

1.2 Add user

Enter the user’s password twice

Press enter to fill the fields with default information

Enter Y then hit enter to continue

1.3 Add new user to sudo group

Step 2: Grant Root Privileges to the User

2.1 Edit /etc/sudoers

Find the following lines:

Add this line under “# User privilege specification”

Exit & Save

2.2 Reboot the server and log in with the new user

Step 3: Delete a user and home folder

3.1 Delete user and home folder