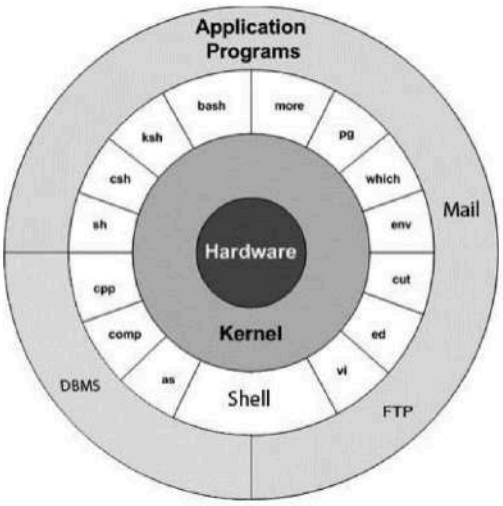
- **Kernel:**
  - The kernel is the heart of the operating system.
  - It interacts with hardware and most of the tasks like memory management, task scheduling and file management.
- **Shell:**
  - The shell is the utility that processes your requests.
  - When you type in a command at your terminal, the shell interprets the command and calls the program that you want.
  - The shell uses standard syntax for all commands.
  - C Shell, Bourne Shell and Korn Shell are most famous shells which are available with most of the Unix variants.
- **Commands and Utilities:**
  - There are various commands and utilities which you would use in your day to day activities.
  - **cp**, **mv**, **cat** and **grep** are few examples of commands and utilities.
  - There are over 250 standard commands plus numerous others provided through 3rd party software.
  - All the commands come along with various optional options.

Files and Directories:

- All data in Linux is organized into files. All files are organized into directories.
- These directories are organized into a tree-like structure called the filesystem.

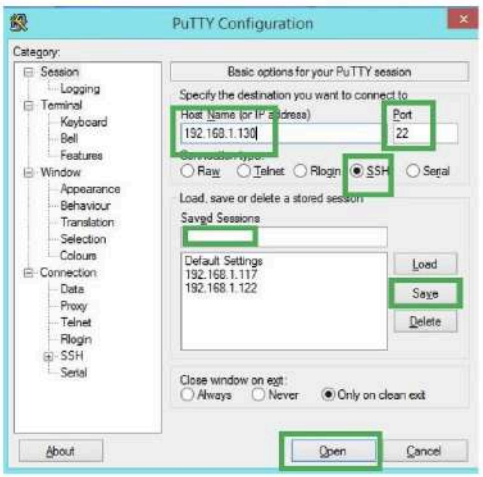
Simplified architecture of Linux (III)



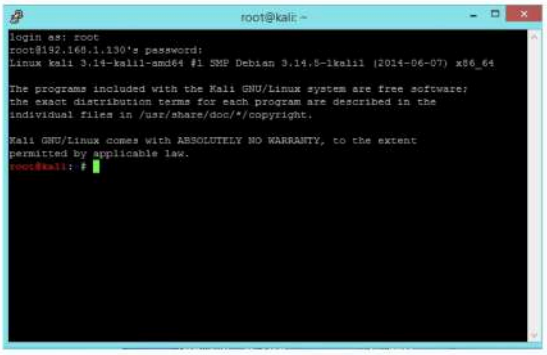
Remote access to a linux server (I)

- Usually is done via SSH
- The SSH server can be installed like this:
  - sudo apt-get install openssh-server // in Ubuntu
  - yum install openssh-server // In RedHat, CentOS
- Start the SSH server:
  - sudo service ssh restart // in Ubuntu
  - service sshd start //in Redhat, CentOS
- Download a terminal emulator client:
  - putty or Ericom Interconnect

Run putty, enter the hostname/IP, the port (default is 22) and hit "Open".



Enter the user/password and you are connected to the Linux BASH environment



BASH – the Linux shell

- BASH is a programming/scripting language
- BASH shell is the Linux equivalent of the Windows **cmd**
- BASH is a command processor that typically runs in a text window, where the user types commands that cause actions
- BASH runs scripts (python, perl, etc)
- It has been ported to Windows (via Cygwin)

- When you do not know what a command does:
  - **man** – stands for manual
    - man ls
    - man cd
    - man grep
    - etc,

- **whoami** – shows the user you are currently logged in with
- **users** – displays (all) the users currently logged in

- So you are logged into this black Linux shell, but you have no info about the type of Linux distro or the architecture…
- **uname** – prints the name, version and other details about the current machine and the operating system running on

```
root@Ubuntu14:/home/cristian# uname -a
Linux Ubuntu14 3.16.0-30-generic #40~14.04.1-Ubuntu SMP Thu Jan 15 17:43:14 UTC
2015 x86_64 x86_64 x86_64 GNU/Linux
root@Ubuntu14:/home/cristian#
```

- **lsb_release -a** - prints Distribution information.

```
root@Ubuntu14:/home/cristian# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 14.04.2 LTS
Release:        14.04
Codename:       trusty
root@Ubuntu14:/home/cristian#
```

- Make a **copy** of a file using the **cp** command.
  - cp source_file destination_file

- Renaming a file with the **mv** command:
  - mv old_file new_file

- Delete one or multiple files with **rm**:
  - rm filename1
  - rm filename1 filename2 filename3 //multiple files
  - rm –r –f /home/cristian/* // deletes all files in /home/Cristian without confirmation
  - rm *.txt ./ //deletes all .txt files in the current directory

- **cd** - The cd command is used to change the current directory (i.e., the directory in which the user is currently working) in Linux.
  - cd /home/cristian
  - cd ~ // "~" stands for the user's home directory

- **ls** - list directory contents
  - ls –lh /home/
  - ls ./

- **cat** - (short for concatenate) command is one of the most frequently used commands on Linux
- It can be used for:
    - Display text file on screen
    - Read text file
    - Create a new text file
    - Modifying file

Read text file

cat file_name

cat /path/to/file

```
[root@fcsteaua ~]# cat /root/blockip.sh
#!/bin/bash
BLOCKDB="/root/ip.blocked"
IPS=$(grep -Ev "^#" $BLOCKDB)
for i in $IPS
do
iptables -A INPUT -s $i -j DROP
iptables -A OUTPUT -d $i -j DROP
done
[root@fcsteaua ~]#
```

- **Create a new text file**
    - cat > newfile.txt // can be done with the **touch** command

```
[root@fcsteaua ~]# cat > foo.txt
I'm inserting some text here
pressing enter to go to the next line
and pressing Ctrl+D to save and exit
[root@fcsteaua ~]#
```

- **Modifying file:**
    - To append (add data to existing) data to a file called foo.txt, enter:

```
[root@fcsteaua ~]# cat >> foo.txt
I need to add some new text to this file
the existing text in the file will
not be overwritten
[root@fcsteaua ~]#
```

- **Extra:**
    - List the foo.txt file and display line numbers
    - Very useful when you encounter script errors

```
[root@fcsteaua ~]# cat -n foo.txt
     1  I'm inserting some text here
     2  pressing enter to go to the next line
     3  and pressing Ctrl+D to save and exit
     4  I need to add some new text to this file
     5  the existing text in the file will
     6  not be overwritten
[root@fcsteaua ~]#
```

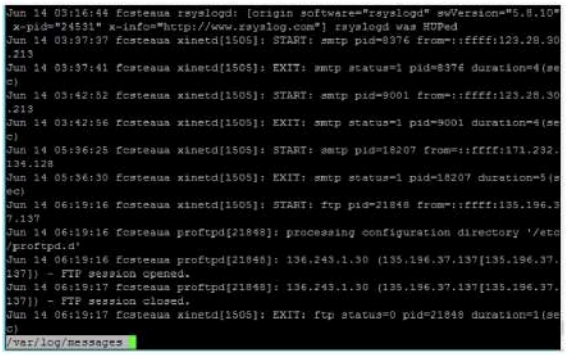
- **grep** - searches the named input FILEs for the lines that match the specified pattern
    - grep is the equivalent of **findstr.exe** in Windows
    - Example:

```
[root@fcsteaua ~]# grep error /var/log/messages
Jun 19 09:43:05 fcsteaua proftpd[30145]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01
:4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:05 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01
:4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:08 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01
```

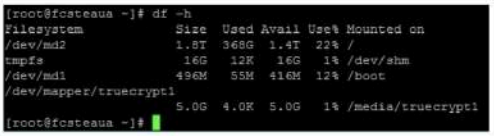
        - I want to list the /var/log/messages file for the "error" patern
            - **grep error /var/log/messages**
            - Or with pretty colors
            - **grep -i error /var/log/messages**

```
[root@fcsteaua ~]# grep --color error /var/log/messages
Jun 19 09:43:05 fcsteaua proftpd[30145]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:05 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:08 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
4f8:211:1a9d::2]) - error setting listen fd IPV6_TCLASS: Protocol not available
Jun 19 09:43:08 fcsteaua proftpd[30146]: 136.243.1.30 (2a01:4f8:211:1a9d::2[2a01:
```

- **more** - is a filter for paging through text one screenful at a time
- **less** - is a program similar to more (1), but which allows backward movement in the file as well as forward movement.
    - The syntax:
        - **more** /my/log/file
        - **less** /my/log/file

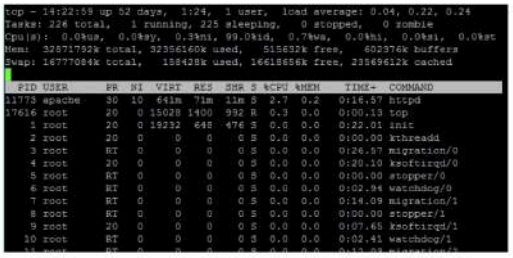


- Hard drive usage:
    - **df** - displays the amount of disk space available on the file system

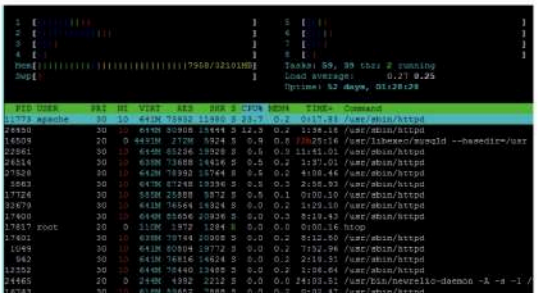


    - **du -** estimates and displays the disk space used by files and directories

    - Processor, memory, general server load
        - **top -** provides a dynamic real-time view of a running system. It can display system summary information, as well as a list of processes or threads currently being managed by the kernel



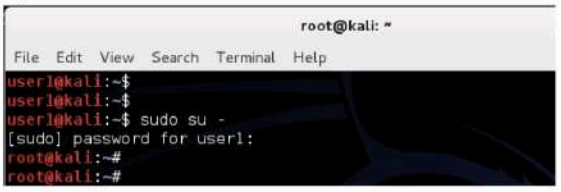
        - Processor, memory, general server load
            - **htop –** similar to top, but with more details and fancier colors



- The equivalent of the "C:\" partition in Windows is referred in Linux as "**/**" – also called "root directory".
- The Linux filesystem has the root directory at the top of the directory tree. The following list of directories are subdirectories of the root directory:
  - **/bin:**
    Contains executable programs such as ls ("dir" in Windows) and cp ("copy" in Windows). These programs are designed to make the system usable.
  - **/etc**
    Contains configuration files which are local to the machine. Programs store configuration files in this directory and these files are referenced when programs are run.
  - **/home**
    Contains user account directories. Each user created by the system administrator will have a subdirectory under /home with the name of the account. This is the default behaviour of

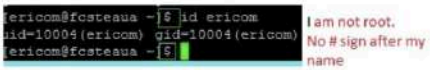
- **/mnt**
  Used for mounting temporary filesystems. When mounting a CD-ROM for instance, the standard mount point location is /mnt/cdrom.
- **/opt**
  Used for storing random data that has no other logical destination.
- **/proc**
  Provides information about running processes and the kernel. A directory is provided for each running process. Useful system information such as the amount of Random Access Memory (RAM) available on the system as well as Central Processing Unit (CPU) speed in Megahertz (MHz) can be found within the /proc directory.
- **/root**
  This is the home directory for the super user (root). This directory is not viewable from user accounts. The /root directory usually contains system administration files.
- **/sbin**
  Similar to /bin, this directory contains executable programs needed to boot the system, however the programs within /sbin are executed by the root user.
- **/tmp**
  This directory is used for temporary storage space. Files within this directory are often cleaned out either at boot time or by a regular job process.
- **/usr**
  Used to store applications. When installing an application on a Debian GNU/Linux machine, the typical path to install would be /usr/local. You will notice the directory structure within /usr appears similar to the root directory structure.
- **/var**
  This directory contains files of variable file storage. Files in /var are dynamic and are constantly being written to or changed. This the directory where websites are usually stored in

- Linux has limited access users and, by default, one administrator (called "**root**")
- **root** is the user name or account that by default has access to all commands and files on Linux.
- It is also referred to as the root account, root user and the superuser.
- You can grant root like access to limited users using **sudo** (see "Run as Administrator in Windows")

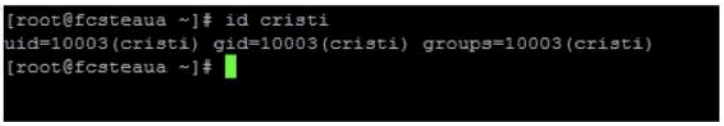


With **sudo**, as a limited permissions user, you can be granted, temporarily, administrator/root access to execute commands usually restricted to only the root user.
sudo is used in Linux Debian derivatives distros (Ubuntu, SteamOS from Valve, Kali Linux, etc) – but not limited to only Debian
sudo can be installed on any Linux system
Not every user can use sudo. That user must be present in the /etc/sudoers file
In the BASH environment/the linux shell, the root user can be recognized by

- the pound sign (**#**). Limited users can be recognized by the "$" sign after their name.
- When not sure about the user you are currently logged in, issue the **whoami** command



- All users have:
  - user IDs (**uid**), group IDs (**gid**).
  - The **uid** and **gid** are always decimal numbers and start from 1000 or 10000
  - The root superuser usually has **uid** and **gid** 0 (zero)
  - A specific user can be member of multiple groups.
- The **id** command show all the information you need to know about a user
- Try issuing the **id root** command and see what happens



- How do I **add a new user** via the linux shell?
  - useradd testuser –p test123
- The command above created a new user called testuser with the password test123
- How do I **assign a user to another group**?
- usermod –G root testuser
- I added the **user testuser** to the **root group**.

Create a new group:

```
[root@fcsteaua ~]# groupadd connect-group
[root@fcsteaua ~]# cat /etc/group | grep connect
connect-group:x:10005:
[root@fcsteaua ~]#
```

Delete a group:

```
[root@fcsteaua ~]# groupdel connect-group
[root@fcsteaua ~]# cat /etc/group | grep connect
[root@fcsteaua ~]#
```

Change the password of a user with the **passwd** command:

```
[root@fcsteaua ~]# passwd ericom
Changing password for user ericom.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@fcsteaua ~]#
```

- Login as root if you are changing a password for an account different than yours
  - If you are logged in with a limited user account, use the **su** command or **sudo su** to login as root

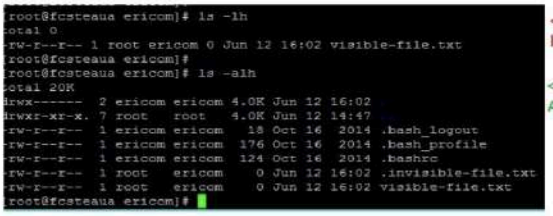
Each file and directory has three user based permission groups:

- **owner** - The Owner permissions apply only the owner of the file or directory, they will not impact the actions of other users.
- **group** - The Group permissions apply only to the group that has been assigned to the file or directory, they will not effect the actions of other users.
- **all users** - The All Users permissions apply to all other users on the system, this is the permission group that you want to watch the most.

- Permission Types
- Each file or directory has three basic permission types:
  - The **read permission** grants the ability to read a file. When set for a directory, this permission grants the ability to read the names of files in the directory, but not to find out any further information about them such as contents, file type, size, ownership, permissions.
  - The **write permission** grants the ability to modify a file. When set for a directory, this permission grants the ability to modify entries in the directory. This includes creating files, deleting files, and renaming files.
  - The **execute permission** grants the ability to execute a file. This permission must be set for executable programs, including shell scripts, in order to allow the operating system to run them. When set for a directory, this permission grants the ability to access file contents and meta-information if its name is known, but not list files inside the directory, unless read is set also

View the permissions:

- **ls** is the utility you need
- Is the equivalent of **dir** in Windows
- Standard usage is ls –lh (list, show permissions and display them in human readable format)
- Any file or folder that starts with a dot character (for example, /home/user**/.config**), commonly called a dot file or dotfile, is hidden.



<< List only non-hidden files

<< List non-hidden AND hidden files

- Reading the file and directory permissions

-rw-r--r-- 1 root ericom 0 Jun 12 16:02 file.txt

- The first character (-) indicates the file type and is not related to permissions. The remaining nine characters are in three sets, each representing a class of permissions as three characters:

Each of the three characters represent the read, write, and execute permissions:
- r if reading is permitted, - if it is not.
- w if writing is permitted, - if it is not.
- x if execution is permitted, - if it is not.

- The **second** set represents the **group class.**

rw-	r--	r--
The owner (root) can read and write the file	The users in the ericom group can read the file	Everyone else can read the file

- Another example:

-rwxr-x--- 1 root ericom 144K Jun 12 11:02 script.sh

rwx	**r-x**	**---**
Owner (root in this case) can read, write and execute the file	The users in the ericom group can read and execute the file	Everyone else cannot read, write or execute the files.

The alternative to the symbolic (rwx) permission system:
Meet the octal notation:

Symbolic Notation	Octal Notation	English
----------	0000	no permissions
---x--x--x	0111	execute
--w--w--w-	0222	write
--wx-wx-wx	0333	write & execute
-r--r--r--	0444	read
-r-xr-xr-x	0555	read & execute
-rw-rw-rw-	0666	read & write
-rwxrwxrwx	0777	read, write, & execute

- Modify the permissions with **chmod**
- When you:
  - grant permission you use the plus sign "+"
  - take permission away you will use the minus sign "-"

Example 1:

Grant permission for read, write and execute to the file owner

chmod u+rwx file.txt //in octal: chmod 700 file.txt

Example 2:

Take away all privileges from user eircom for file.txt

chmod u-rwx file.txt

Example 3:

Grant permission for read, write and execute for user, group and everyone else

chmod ugo+rwx file.txt // in octal: chmod 777 file.txt

Example 2:

Take away all privileges from user, group and everyone else

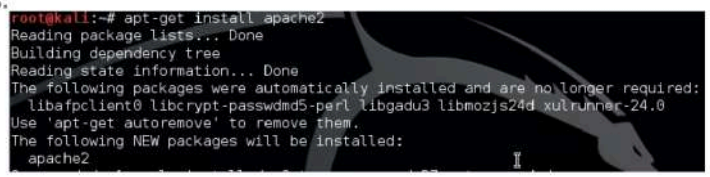
chmod ugo-rwx file.txt // in octal: chmod 000 file.txt

Example 3:

Grant recursive permission in a specific directory

chmod -R ugo+rwx /path/to/my/directory // in octal: chmod –R 777 /path/to/my/directory

- In Debian & Ubuntu like systems:
- apt-get install apache2
- // installs the Apache httpd server



- In Redhat and CentOS like systems:
- yum install httpd
- //installs Apache httpd server. See the difference in names!

RPMs and DEB files

- **RPM Package** Manager (**RPM**) (originally Red Hat **Package** Manager) is a **package** management system. The name **RPM** variously refers to the **.rpm file** format, **files** in this format, software packaged in such **files**, and the **package** manager itself.

- **deb** is the extension of the Debian software **package** format and the most often used name for such binary **packages.**

  - In Debian & Ubuntu like systems:

  - **wget** http://www.eu.apache.org/dist//directory/apacheds/dist/2.0.0-M20/apacheds-2.0.0-M20-amd64.deb
  - //download the file DEB file
  - **chmod** +x apacheds-2.0.0-M20-amd64.deb // make the file executable
  - **dpkg** -i apacheds-2.0.0-M20-amd64.deb // install the Apache DEB package
  - /etc/init.d/apache2 start ///start Apache

  - In Redhat and CentOS like systems:
  - **wget** ftp://rpmfind.net/linux/centos/5.11/os/i386/CentOS/httpd-2.2.3-91.el5.centos.i386.rpm
  - //download the RPM file
  - **chmod+x** httpd-2.2.3-91.el5.centos.i386.rpm // make the file executable
  - **rpm** -i httpd-2.2.3-91.el5.centos.i386.rpm // install the httpd RPM file
  - **service** httpd start // start the Apache server

- Software can be installed from the code source without being a developer
- You need root access or you can use **sudo**
- You will need a C compiler (called GCC in Linux)
- Access to a BASH console is mandatory

Example. Install **pidgin** from source code in Ubuntu.

- **sudo apt-get install build-essential** // this will install the compiler and other required libraries

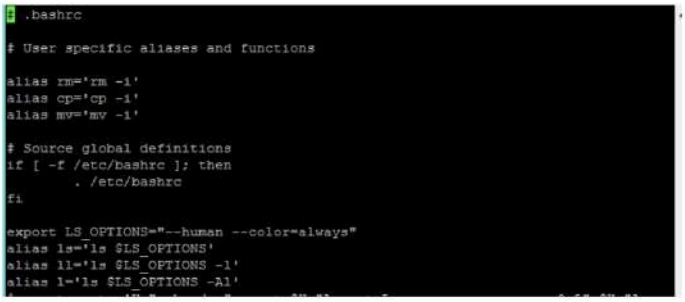
Now you'll need your desired application's source code. These packages are usually in compressed files with the .tar.gz or .tar.bz2 file extensions.

- **wget** http://downloads.sourceforge.net/project/pidgin/Pidgin/2.10.11/pidgin-2.10.11.tar.bz2
- **tar** -xjvf pidgin-2.10.11.tar.bz2 // extract the content of the archive
- **cd** pidgin-2.10.11 // navigate to the new created directory
- **./configure // configure the new install**
- **make** // compile the program
- **make install** // install the software on your system

- HTTP server:
  - Apache (httpd), nginx
- SQL:
  - Mysql (mysqld), SQLite, postgresql
- FTP servers:
  - Proftpd, Pure-FTPd, vsFTPd, Filezilla
- DNS servers (Bind),
- Firewall (iptables, ipchains),
- SMTP servers (postfix, qmail, sendmail),
- POP3 / IMAP servers (Dovecot, Courier)
- Remote access server (OpenSSH)

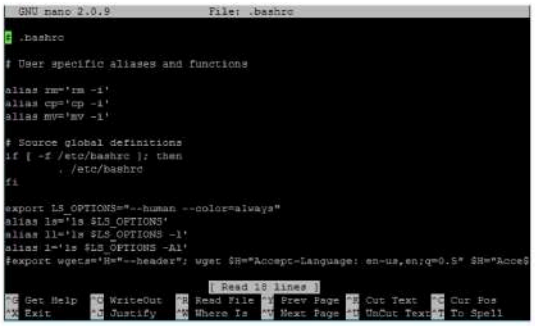
- **Text editors**
  - vi

Vi is a powerful text editor included with most Linux systems, even embedded ones. Sometimes you'll have to edit a text file on a system that doesn't include a friendlier text editor, so knowing Vi is essential.



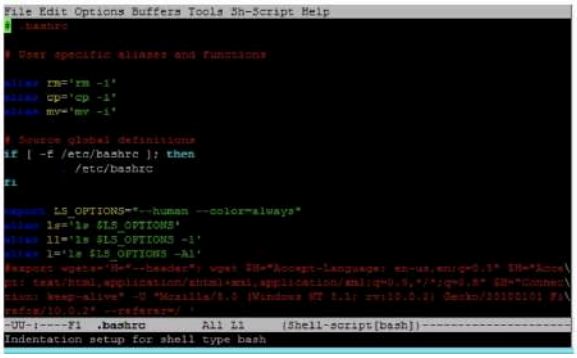
- **Text editors**
  - nano

nano is a small and friendly text editor. Besides basic text editing, nano offers many extra features like an interactive search and replace, go to line and column number.



- **Text editors**
  - emacs

Emacs is one of the oldest and most versatile text editors available for Linux and UNIX-based systems. It's been around for a long time and is well known for its powerful and rich editing features.



How to run Linux scripts

- You have the blockip.sh script that is located /home/eircom
- First check if the script can be executed by the user you are currently logged in with:
  - ls -lh /home/ericom/blockip.sh
- If you cannot execute it, do a:
  - **chmod u+rx /home/cristi/blockip.sh** // or **chmod 500 /home/cristi/blockip.sh**
- Run the script:
  - **/home/ericom/blockip.sh** // or if you are already in the /home/eircom, run it with **./blockip.sh**
  - If your connection drops your script might crash
- Make the script run after you exit the shell or the connection is interrupted:

- Most seen file extensions are **.tar.gz** and **.tar.bz2** which is a tar archive further compressed using **gzip** or **bzip** algorithms respectively.
- Create archives
  - tar -cvf mynewarchive.tar /var/www
  - (will create mynewarchive.tar with the content of /var/www)
- Extract a tar.gz archive:
  - tar -xvzf tarfile.tar.gz
- Extract tar.bz2/bzip archives
  - tar -xvjf archivefile.tar.bz2
- Extract files to a specific directory or path
  - tar -xvzf abc.tar.gz -C /opt/folder/

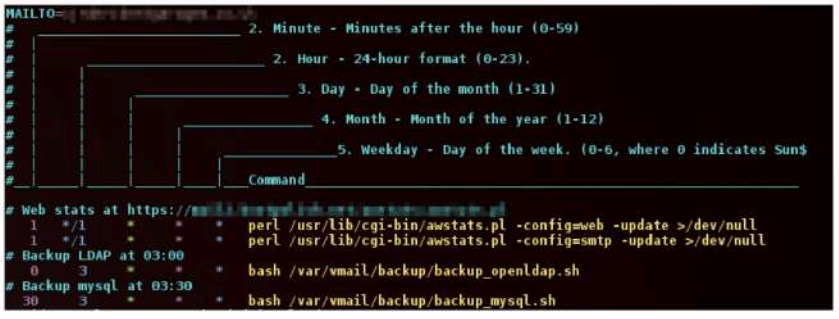
- Extract a single file
  - tar -xz -f archive.tar.gz "./new/file.txt"

- Extract multiple files
  - tar -xv -f abc.tar.gz "./new/cde.txt" "./new/abc.txt"

- Extract multiple files using wildcards
  - tar -xv -f abc.tar.gz --wildcards "*.txt"

Automatically perform tasks – cron (I)

- **cron** is the system process which will automatically perform tasks for you according to a set schedule. The schedule is called the crontab, which is also the name of the program used to edit that schedule.
- The **crontab** is a list of commands that you want to run on a regular schedule, and also the name of the command used to manage that list.



- **How to use crontab**
  - In BASH issue the following commands:
    - crontab –e // edit the cron for the user you are currently logged in with
    - crontab –l // list the current crontab file
    - The crontab file is usually edited with the **vi** text editor (see http://www.shortcutworld.com/en/linux/vi.html for the shortcuts)

Automatically perform tasks – backup with tar & cron (I)

- Backup your files with **tar:**
  - tar -cf backup.tar /var/www/vhosts/
  - tar -cvz -f archive-$(date +%Y%m%d).tar.gz /var/www/vhosts/
  - nohup tar -cf backup.tar /var/www/vhosts/ & // this will keep the backup running if you disconnect from the BASH session
  -
  - **Use crontab to schedule automatic backup:**
  - Add this line to crontab to backup your files every day at 4:00 AM
    - 0 4 * * * tar -cvz -f archive-$(date +%Y%m%d).tar.gz /var/www/vhosts/

- **Use crontab to schedule automatic backup:**
- Add this line to crontab to backup your files every day at 4:00 AM
  - 0 4 * * * /bin/tar -cvz -f archive-$(date +%Y%m%d).tar.gz /var/www/vhosts/

The steps:

1. crontab -e
2. Go to the end of the file
3. Press the "i" key (for insert)
4. Paste the backup command here (push the scroll button on the mouse or shift+insert)
5. Press the ESC key
6. Type :wq //to save and close crontab:

The iptables firewall (I)

- **What is iptables ?**
- Iptables is a rule based firewall system and is normally pre-installed on a Linux operating system which is controlling the incoming and outgoing packets. By-default the iptables is running without any rules, we can create, add, edit rules to it.
  - service iptables start|stop|restart|status // check the status of the iptables service in Redhat/CentOS
  - sudo iptables -L -n -v // check the status of the iptables service in Debian, Ubuntu

- Iptables -L // list the current rules of the iptables firewall
- iptables –flush // delete all the rules temporarily.

```
[root@fcsteaua home]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  92.83.92.229         anywhere
DROP       all  --  92.83.92.229         anywhere
DROP       all  --  92.83.92.229         anywhere
DROP       all  --  1.80.0.0/13          anywhere
DROP       all  --  1.92.0.0/14          anywhere
DROP       all  --  192.1.broad.ha.dynamic.163data.com.cn/13  anywhere
DROP       all  --  0.0.202.1.static.bjtelecom.net/15  anywhere
DROP       all  --  1.204.0.0/14         anywhere
DROP       all  --  14.144.0.0/12        anywhere
DROP       all  --  14.208.0.0/12        anywhere
DROP       all  --  23.80.64.0.rdns.as15003.net/24  anywhere
DROP       all  --  23.104.141.0.rdns.as15003.net/24  anywhere
DROP       all  --  23.105.14.0.rdns.racklot.com/24  anywhere
DROP       all  --  27.8.0.0/13          anywhere
DROP       all  --  27.16.0.0/12         anywhere
DROP       all  --  27.36.0.0/14         anywhere
DROP       all  --  27.40.0.0/13         anywhere
DROP       all  --  27.50.128.0/17       anywhere
DROP       all  --  27.54.192.0/18       anywhere
DROP       all  --  27.106.128.0/18      anywhere
DROP       all  --  27.115.0.0/17        anywhere
DROP       all  --  27.148.0.0/14        anywhere
^C
[root@fcsteaua home]#
```

- (Me blocking (not) most of China's IPs)

The logs (I)

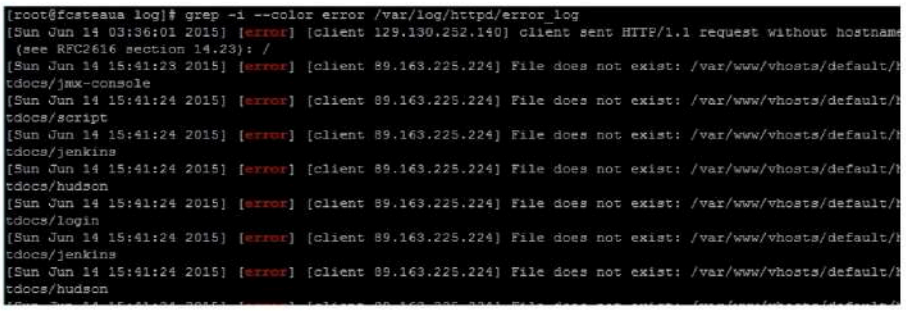
- The default log folder in Linux is /var/log
- How do I view log files on Linux?
- Go to /var/log directory using the following cd command:
  - # cd /var/log
- To list files use the following ls command:
  - # ls or ls -lh

```
[root@fcsteaua log]# ls
anaconda          dracut.log          maillog.processed.1.gz  pluto                   spooler
boot.log          dracut.log-20150501 maillog.processed.2.gz  ppp-horde               spooler-20150524
btmp              fcs_cron.log        maillog.processed.3.gz  sa-update.log           spooler-20150531
btmp-20150601     httpd               mailman                 sa-update.log-20150301  spooler-20150607
CentOSMirror      iptraf              messages                sa-update.log-20150401  spooler-20150614
cron              lastlog             messages-20150524       sa-update.log-20150501  sw-cp-server
cron-20150524     maillog             messages-20150531       sa-update.log-20150601  tallylog
cron-20150531     maillog-20150524    messages-20150607       secure                  wtmp
cron-20150607     maillog-20150531    messages-20150614       secure-20150524         yum.log
cron-20150614     maillog-20150607    mysqld.log              secure-20150531         yum.log-20150101
dmesg             maillog-20150614    newrelic                secure-20150607
dmesg.old         maillog.processed   ntpstats                secure-20150614
[root@fcsteaua log]#
```

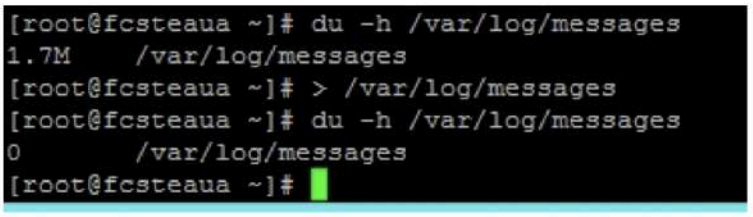
Common logs and their location in Linux:

- /var/log/messages : General message and system related stuff
- /var/log/auth.log : Authenication logs
- /var/log/kern.log : Kernel logs
- /var/log/cron.log : Crond logs (cron job)
- /var/log/maillog : Mail server logs
- /var/log/qmail/ : Qmail log directory (more files inside this directory)
- /var/log/httpd/ : Apache access and error logs directory
- /var/log/lighttpd/ : Lighttpd access and error logs directory
- /var/log/boot.log : System boot log
- /var/log/mysqld.log : MySQL database server log file
- /var/log/secure or /var/log/auth.log : Authentication log
- /var/log/utmp or /var/log/wtmp : Login records file
- /var/log/yum.log : Yum command log file

- Display a specific log file:
  - # **less** /var/log/messages
    # **more** -f /var/log/messages
    # **cat** /var/log/messages
    # **tail** -f /var/log/messages
    # **grep** -i error /var/log/messages
  - **grep** with pretty colors:



- To empty large files you need to issue one of the following commands:
  - **>** /path/to/large/logfile
  - **echo** " " > /path/to/large/logfile



- In the screen shot above I am emptying my 1.7 MB /var/log/messages log file

Networking in Linux (I)

- The is no "Local area connection"
- **Naming convention is:**
  - eth0
  - eth1, etc
- **Subinterfaces/virtual network cards** are noted with "."
  - eth0.1, eth0.2,
  - eth 1.1, eth1.2, etc
- Networking config files are in /etc/sysconfig/network-scripts/

```
[root@fcsteaua log]# cd /etc/sysconfig/network-scripts/
[root@fcsteaua network-scripts]# ls
ifcfg-eth0    ifdown-ippp  ifdown-routes  ifup-bnep  ifup-plip    ifup-sit          network-functions
ifcfg-lo      ifdown-ipv6  ifdown-sit     ifup-eth   ifup-pluab   ifup-tunnel       network-functions-ipv6
ifdown        ifdown-isdn  ifdown-tunnel  ifup-ippp  ifup-post    ifup-wireless     route-eth0
ifdown-bnep   ifdown-post  ifup           ifup-ipv6  ifup-ppp     init.ipv6-global
ifdown-eth    ifdown-ppp   ifup-aliases   ifup-isdn  ifup-routes  net.hotplug
[root@fcsteaua network-scripts]#
```

- **Modify DNS servers:**
  - /etc/resolv.conf - is the file you need
- List it's contents with
  - cat /etc/resolv.conf

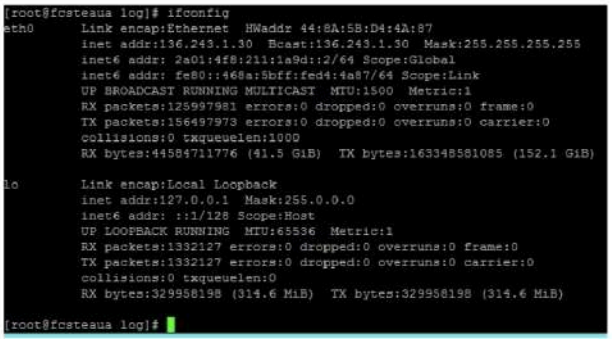
```
[root@fcsteaua network-scripts]# cat /etc/resolv.conf
### Hetzner Online AG installimage
# nameserver config
nameserver 213.133.100.100
nameserver 213.133.98.98
nameserver 213.133.99.99
```

  - Add or delete existent DNS servers, just edit /etc/resolv.conf with a text editor (vi, nano, etc)

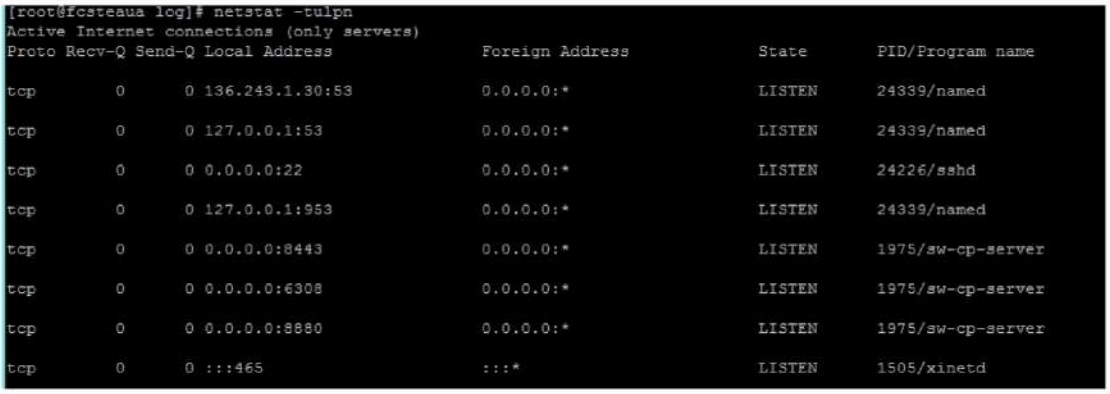
```
  GNU nano 2.0.9                    File: /etc/resolv.conf

### Hetzner Online AG installimage
# nameserver config
nameserver 213.133.100.100
nameserver 213.133.98.98
nameserver 213.133.99.99
nameserver 2a01:4f8:0:a0a1::add:1010
nameserver 2a01:4f8:0:a102::add:9999
nameserver 2a01:4f8:0:a111::add:9898
```

Networking in Linux (III)

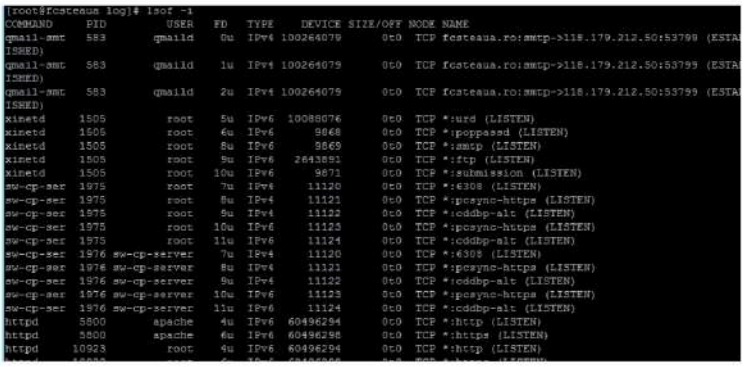
- **ifconfig** - ifconfig stands for "interface configuration". It is used to view and change the configuration of the network interfaces on your system. See **ipconfig** in Windows.



netstat – a useful tool for checking your network configuration and activity



lsof - a command meaning "list open files", which is used in many Unix-like systems to report a list of all open files and the processes that opened them.



route - view and manipulate the TCP/IP routing table in both Unix-like and Microsoft Windows operating systems.
Or **ip route list**

```
[root@fcsteaua log]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
136.243.1.1     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0
0.0.0.0         136.243.1.1     0.0.0.0         UG    0      0        0 eth0
[root@fcsteaua log]#
```

```
[root@fcsteaua log]# ip route list
136.243.1.1 dev eth0  proto kernel  scope link  src 136.243.1.30
169.254.0.0/16 dev eth0  scope link  metric 1002
default via 136.243.1.1 dev eth0
[root@fcsteaua log]#
```

- **Add a default route:**
  - ip route add default via 192.168.1.254

Delete route from table:

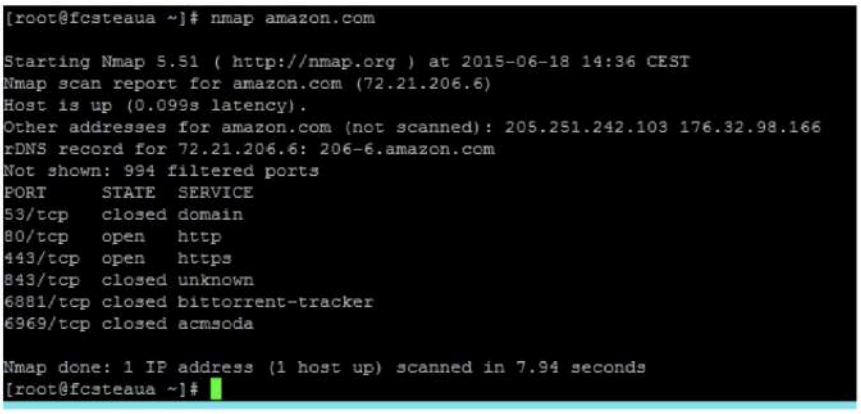
- ip route delete 192.168.1.0/24 dev eth0

- **ping** – utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer and back.
- In Windows you need to ping –t to ping forever.
- In Linux this is the default behaviour. Ctrl+C or Ctrl+Z to stop any Linux command from running continuous.

- **ping** can be blocked by any firewall software. Is there an alternative to ping ?
- Yes.
- Introducing **hping. -** hping is a free packet generator and analyser for the TCP/IP protocol.

```
[root@fcsteaua network-scripts]# ping amazon.com
PING amazon.com (176.32.98.166) 56(84) bytes of data.
^C
--- amazon.com ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4525ms

[root@fcsteaua network-scripts]# hping -S -p 80 amazon.com
HPING amazon.com (eth0 176.32.98.166): S set, 40 headers + 0 data bytes
len=46 ip=176.32.98.166 ttl=238 DF id=11560 sport=80 flags=SA seq=0 win=8190 rtt=91.5 ms
len=46 ip=176.32.98.166 ttl=238 DF id=14554 sport=80 flags=SA seq=1 win=8190 rtt=93.9 ms
len=46 ip=176.32.98.166 ttl=239 DF id=44755 sport=80 flags=SA seq=2 win=8190 rtt=97.7 ms
len=46 ip=176.32.98.166 ttl=238 DF id=29169 sport=80 flags=SA seq=3 win=8190 rtt=93.3 ms
^C
--- amazon.com hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 91.5/94.1/97.7 ms
[root@fcsteaua network-scripts]#
```

- DNS tools:
- **host -** host is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa.
- **dig** - is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried.
- Examples:
  - host ericom.com 8.8.8.8
  - dig @8.8.8.8 eircom.com in A
- **nmap** - (Network Mapper) is a security scanner used to discover hosts and services on a computer network, thus creating a "map" of the network.
- What can be done with nmap ?
  - Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.
  - Port scanning – Enumerating the open ports on target hosts.
  - Version detection – Interrogating network services on remote devices to determine application name and version number.
  - OS detection – Determining the operating system and hardware characteristics of network devices.
- **nmap** example. Probing for open ports

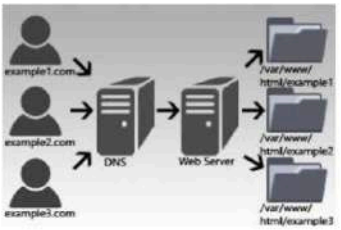


- **nmap** example. OS detection and open ports

```
[root@fcsteaua ~]# nmap -O wikipedia.com

Starting Nmap 5.51 ( http://nmap.org ) at 2015-06-18 14:42 CEST
Nmap scan report for wikipedia.com (91.198.174.192)
Host is up (0.011s latency).
rDNS record for 91.198.174.192: text-lb.esams.wikimedia.org
Not shown: 995 closed ports
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
179/tcp  open     bgp
443/tcp  open     https
5666/tcp filtered nrpe
Device type: WAP|specialized|general purpose|PBX|webcam
Running (JUST GUESSING): Netgear embedded (94%), Crestron 2-Series (93%), Linux 2.6.X|2.4.X (89%), V
odavi embedded (88%), AXIS Linux 2.6.X (85%)
Aggressive OS guesses: Netgear DG834G WAP (94%), Crestron XPanel control system (93%), Linux 2.6.22
(89%), Linux 2.6.17 - 2.6.35 (89%), Linux 2.6.23 - 2.6.33 (88%), Vodavi XTS-IP PBX (88%), Linux 2.6.
31 (87%), Linux 2.4.26 (Slackware 10.0.0) (87%), Linux 2.6.24 (87%), Linux 2.6.13 - 2.6.31 (87%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.12 seconds
```

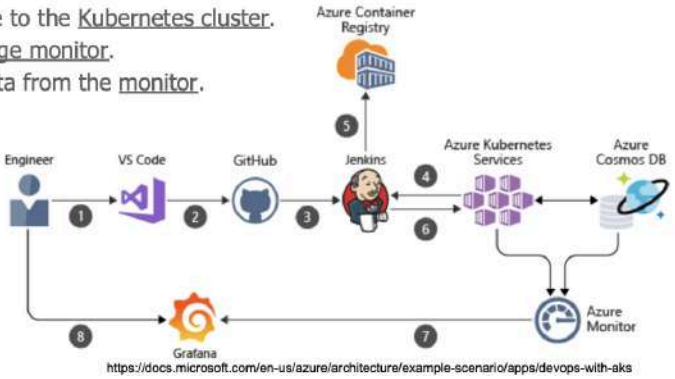
Adding new domain to shared hosting

1. Create an OS user
2. Allocate physical storage
3. Add necessary DNS records
4. Add necessary virtual host records to the web server config
5. Generate and install certificates for HTTPS
6. Configure FTP access to the virtual host
7. ...
8. <Configure another service offered by the server>



CI/CD pipeline

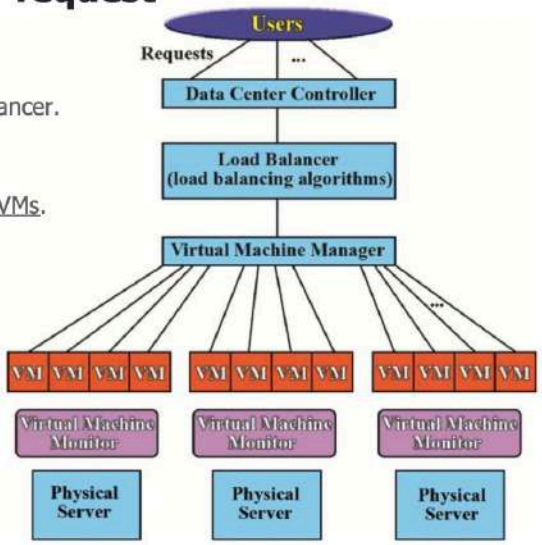
1. A developer makes to a web application source code.
2. The code change is committed to a version control system (VCS).
3. A GitHub webhook triggers a Jenkins project build job.
4. The Jenkins build launches a container build process in Kubernetes.
5. An image is created from the code and pushed to a container registry.
6. Jenkins deploys the updated container image to the Kubernetes cluster.
7. The web application reports metrics to a usage monitor.
8. A visual dashboard solution visualizes the data from the monitor.



https://docs.microsoft.com/en-us/azure/architecture/example-scenario/apps/devops-with-aks

Load balancer processing an HTTP request

1. The request arrives.
2. A Data Center Controller submits a task to the Load Balancer.
3. The Load Balancer contacts a Virtual Machine Manager.
4. The Load Balancer assigns the task to a suitable VM.
5. A Virtual Machine Monitor (VMM) creates and manager VMs.



Google processes your search string

1. The browser performs a DNS lookup.
2. A DNS-based load balancer selects a geographically optimal cluster.
3. The browser connects to the selected cluster for searching.
4. A cluster-local hardware-based load balancer selects a Google Web server (GWS).
5. The GWS queries the Index servers for a list of relevant documents.
6. The GWS fetches the documents from the Document servers.

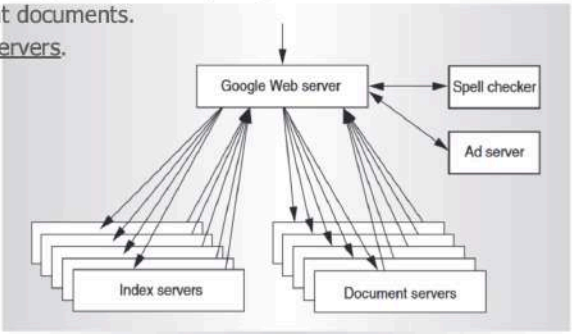


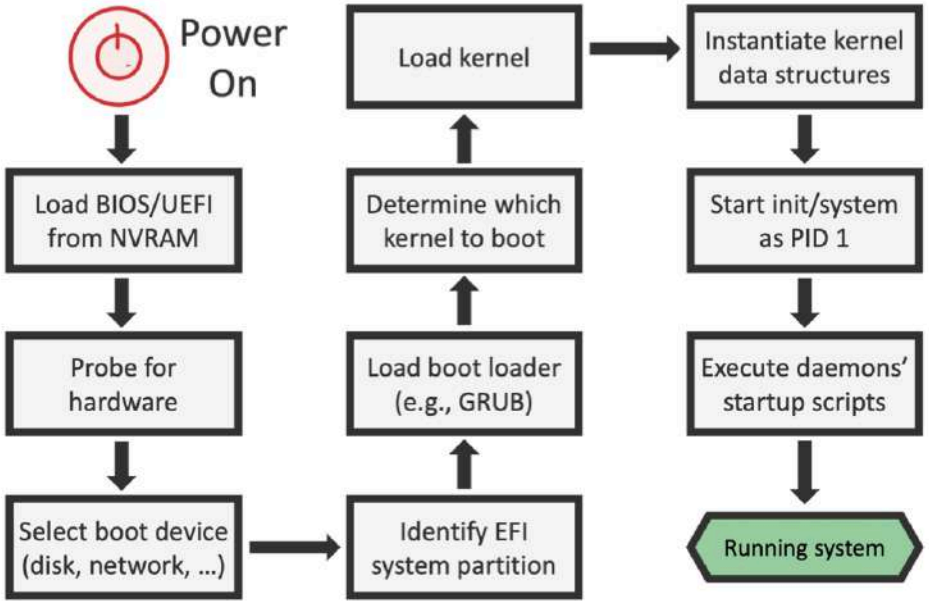
Figure 1. Google query-serving architecture.

Key challenges

1. Understand objectives of the business.
2. Keep the whole picture in mind.
3. Manage your risks.
4. Estimate where changes may come from.
5. MASTER YOUR TOOLS.
6. Know the people in charge of subcomponents.

https://docs.microsoft.com/en-us/azure/architecture/example-scenario/apps/devops-with-aks
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/908d5966b1fa946034e382e608999d51e70d5b22.pdf

Linux boot process

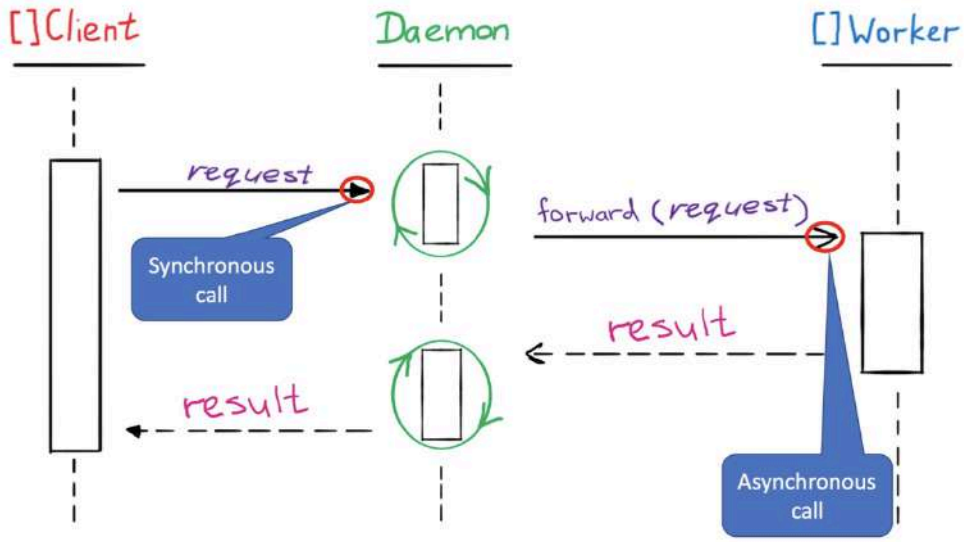


What is a daemon?

daemon

- **variant spelling of *DEMON***
  - : an evil spirit
  - **:** a source or agent of evil, harm, distress, or ruin
- *usually* **daemon:** an attendant power or spirit
- *usually* **daemon**, mythology**:** a supernatural being whose nature is intermediate between that of a god and that of a human being
- one that has exceptional enthusiasm, drive, or effectiveness
- **daemon :** a software program or process that runs in the background

A daemon



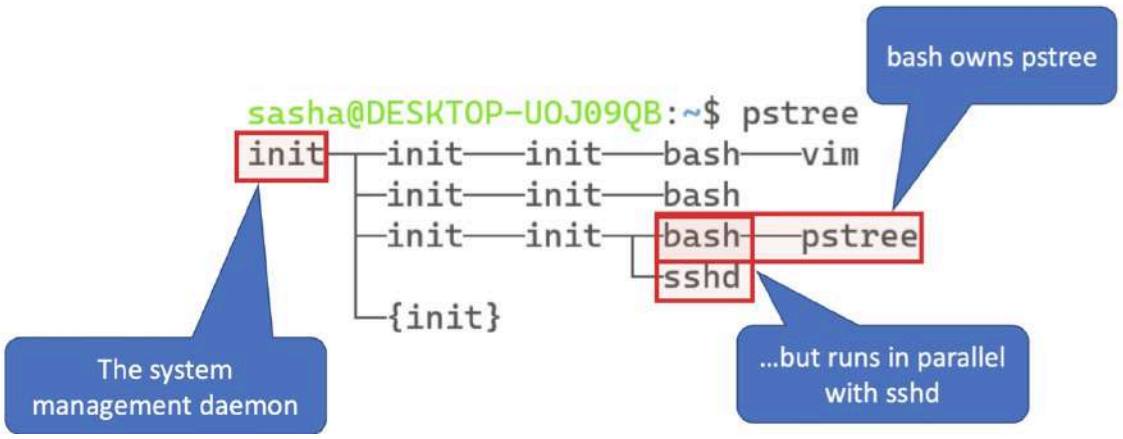
daemon vs. service

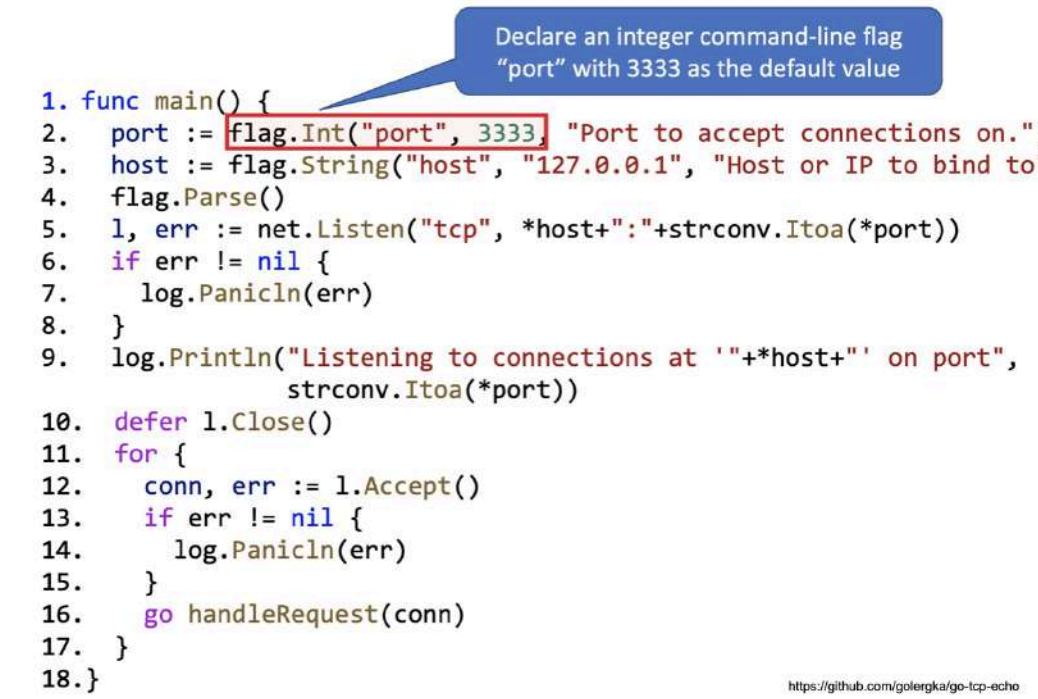
- **[daemon:]** A software program or process that runs in the background.
- **[service:]** A contract with the clients of the associated daemon; defined by an application-level communication protocol.
- **[example:]** The sshd daemon, which works according to the SSH protocol, offers a secure shell service.

Example: SSH

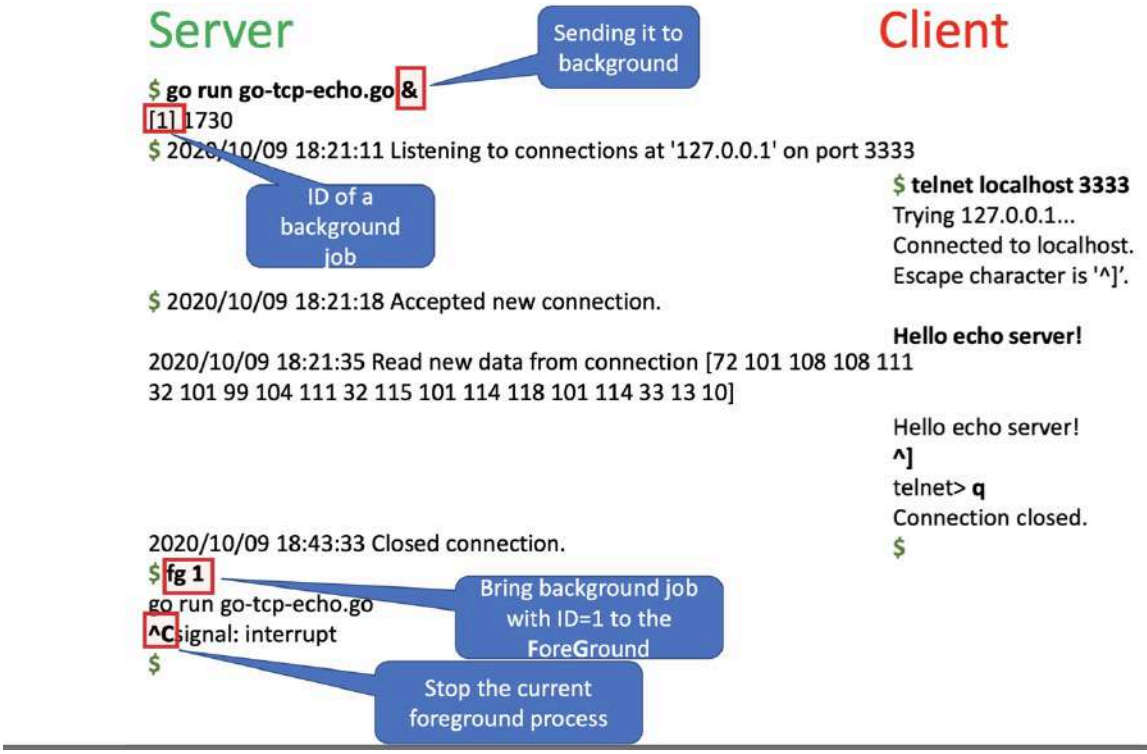
```
$ pidof sshd
$

$ sudo service ssh start          * Starting OpenBSD Secure Shell server sshd
$ !-2
pidof sshd 1679

$ sudo netstat -anlp | grep sshd
tcp     0      0      0.0.0.0:22      0.0.0.0:*       LISTEN      1679/sshd
tcp6    0      0      :::22           :::*            LISTEN      1679/sshd
```

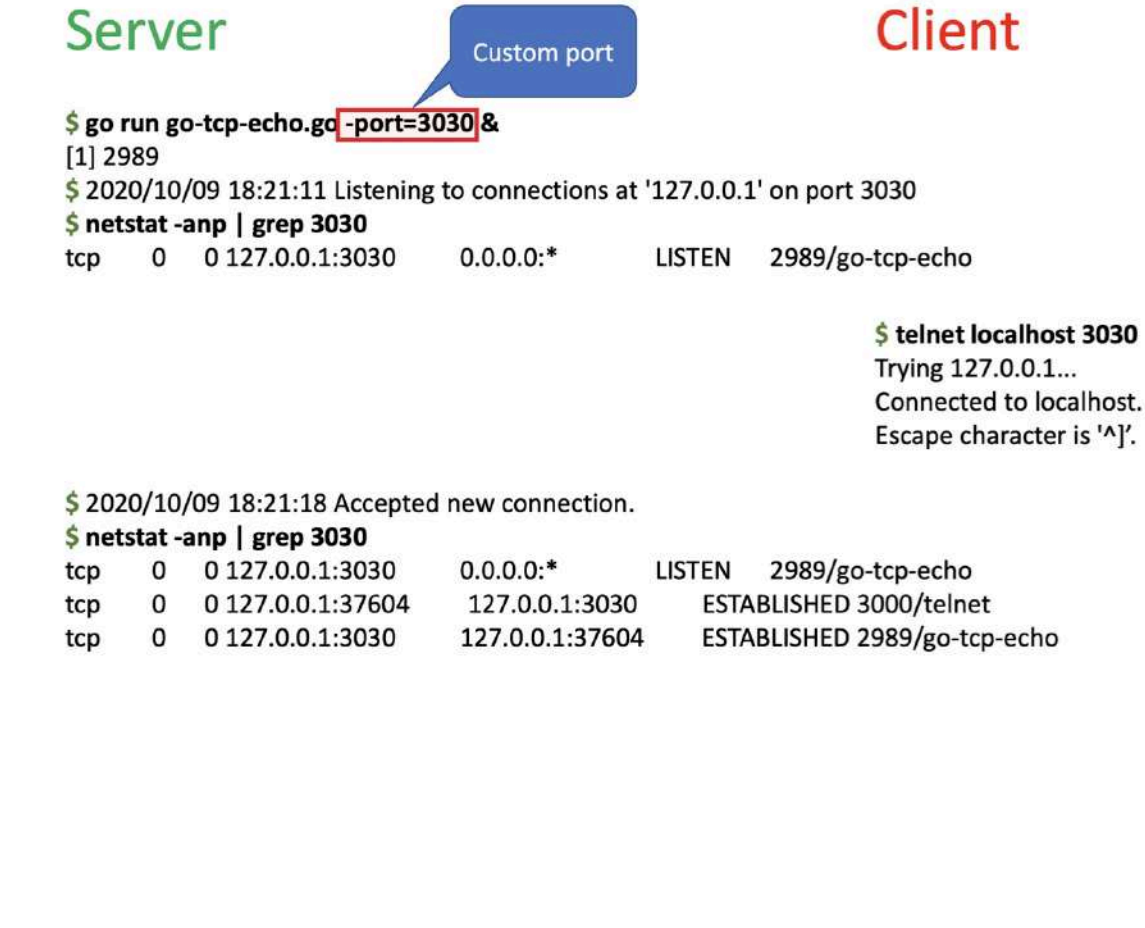
Process tree

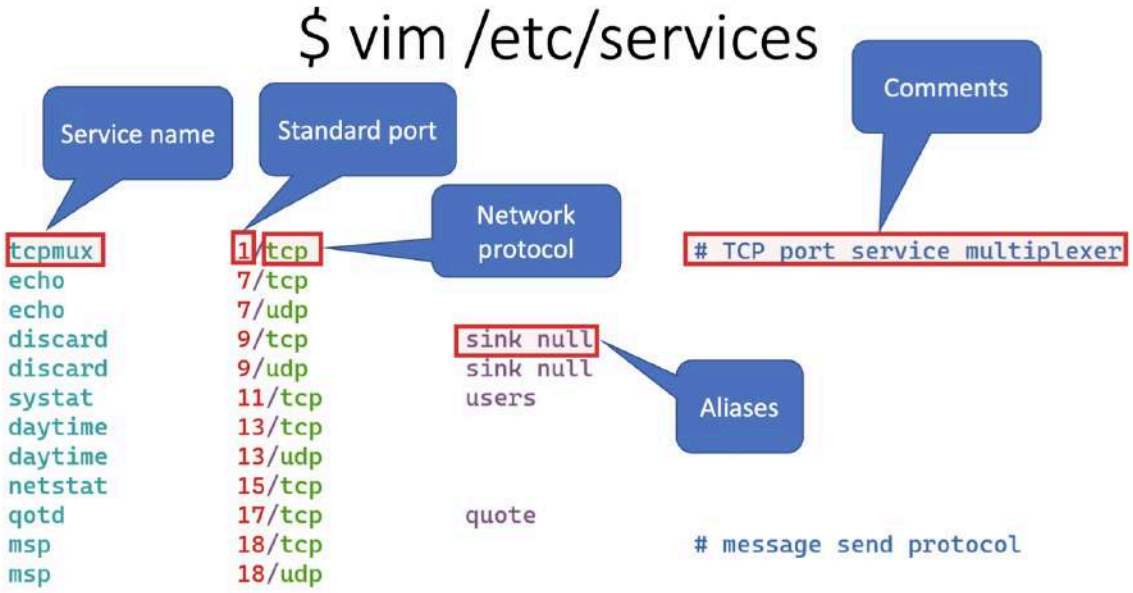





```
1. func handleRequest(conn net.Conn) {
2.   log.Println("Accepted new connection.")
3.   defer conn.Close()
4.   defer log.Println("Closed connection.")
5.   for {
6.     buf := make([]byte, 1024)
7.     size, err := conn.Read(buf)
8.     if err != nil {
9.       return
10.    }
11.    data := buf[:size]
12.    log.Println("Read new data from connection", data)
13.    conn.Write(data)
14.  }
15.}
```


Server
Client
Sending it to background
$ go run go-tcp-echo.go &
[1] 1730
$ 2020/10/09 18:21:11 Listening to connections at '127.0.0.1' on port 3333
ID of a background job
$ telnet localhost 3333
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
$ 2020/10/09 18:21:18 Accepted new connection.
Hello echo server!
2020/10/09 18:21:35 Read new data from connection [72 101 108 108 111
32 101 99 104 111 32 115 101 114 118 101 114 33 13 10]
Hello echo server!
^]
telnet> q
Connection closed.
$
2020/10/09 18:43:33 Closed connection.
$ fg 1
Bring background job with ID=1 to the ForeGround
go run go-tcp-echo.go
^Csignal: interrupt
$
Stop the current foreground process


Server
Client
Custom port
$ go run go-tcp-echo.go -port=3030 &
[1] 2989
$ 2020/10/09 18:21:11 Listening to connections at '127.0.0.1' on port 3030
$ netstat -anp | grep 3030
tcp 0 0 127.0.0.1:3030 0.0.0.0:* LISTEN 2989/go-tcp-echo
$ telnet localhost 3030
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
$ 2020/10/09 18:21:18 Accepted new connection.
$ netstat -anp | grep 3030
tcp 0 0 127.0.0.1:3030 0.0.0.0:* LISTEN 2989/go-tcp-echo
tcp 0 0 127.0.0.1:37604 127.0.0.1:3030 ESTABLISHED 3000/telnet
tcp 0 0 127.0.0.1:3030 127.0.0.1:37604 ESTABLISHED 2989/go-tcp-echo



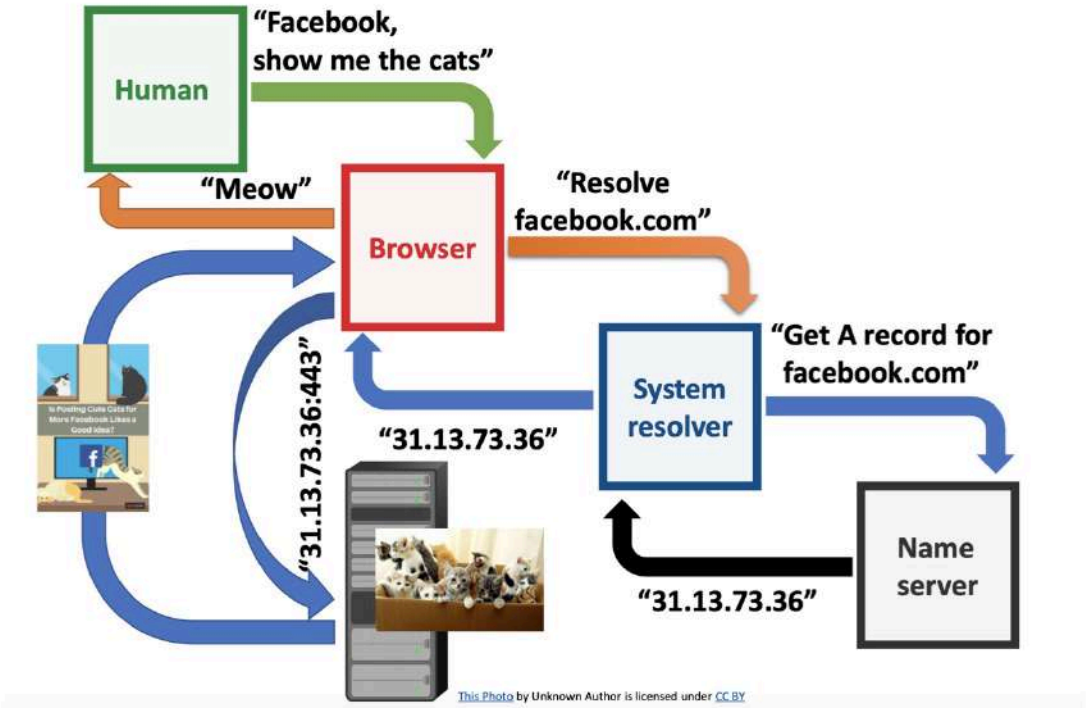
Tools related to daemons

```
sasha@DESKTOP-UOJ09QB:~$ apropos daemon
acpid (8)              - Advanced Configuration and Power Inte
arpd (8)               - userspace arp daemon.
authorized_keys (5)    - OpenSSH SSH daemon
bdflush (2)            - start, flush, or tune buffer-dirty-fl
cron (8)               - daemon to execute scheduled commands
daemon (3)             - run in the background
daemon (7)             - Writing and packaging system daemons
dbus-daemon (1)        - Message bus daemon
dirmngr (8)            - CRL and OCSP daemon
dmeventd (8)           - Device-mapper event daemon
dockerd (8)            - Enable daemon mode
faked (1)              - daemon that remembers fake ownership/
faked-sysv (1)         - daemon that remembers fake ownership/
faked-tcp (1)          - daemon that remembers fake ownership/
git-credential-cache--daemon (1) - Temporarily store user cr
git-daemon (1)         - A really simple server for Git reposi
HTTP::Daemon (3pm)     - a simple http server class
iscsid (8)             - Open-iSCSI daemon
```

Widespread daemons

- Managing other daemons (xinetd, inetd)
- Remote access (sshd)
- Domain name resolution (named)
- Authentication (ldap, ldaps)
- Mail daemons (smtpd, imapd)
- Web servers (httpd, nginx)
- File services (ftpd, nfsd, mountd, lockd, statd)
- Cloud computing (xen server)
- Microservices (dockerd, containerd)
- Logging daemons (klogd, syslogd)

https://linux.die.net/man/8/netstat
https://man7.org/linux/man-pages/man1/apropos.1.html
https://man7.org/linux/man-pages/man1/pstree.1.html
https://linux.die.net/man/1/grep
https://man7.org/linux/man-pages/man1/pidof.1.html
https://man7.org/linux/man-pages/man1/kill.1.html
https://man7.org/linux/man-pages/man1/top.1.html



DNS

- Is a distributed database
- Started from a few hundred hosts inside the US
- Scaled to >1B hosts serving >3B users
- Without DNS, the Internet would never work
- Relies on UDP based daemons listening to port 53

GUI based DNS providers

- Amazon Route 53
- Cloudflare DNS
- GoDaddy Premium DNS • ClouDNS
- Google Cloud DNS
- Azure DNS
- UltraDNS
- DNSMadeEasy

Configuring DNS infrastructure

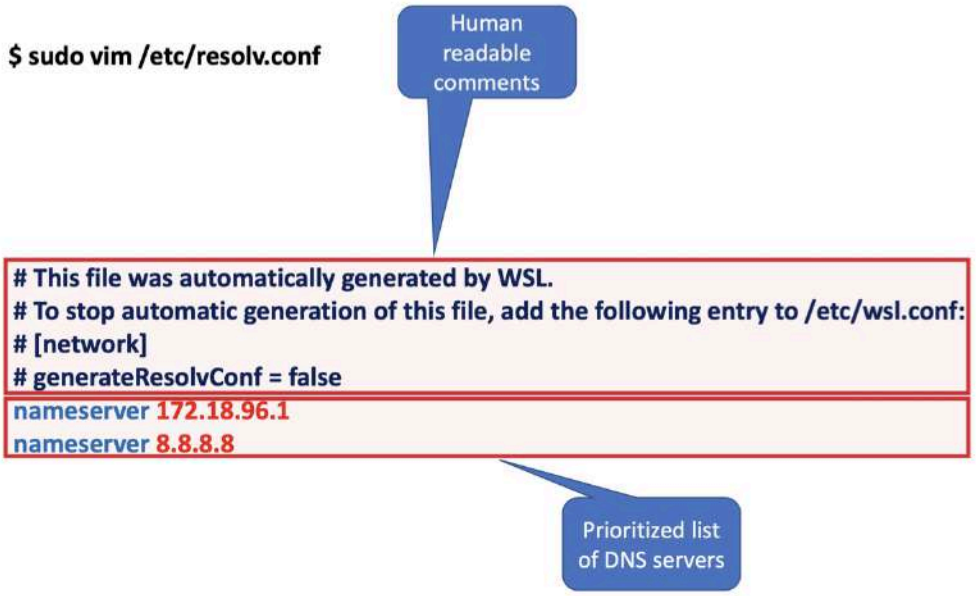
1. Configure all hosts as DNS clients
2. Configure some hosts as DNS servers

Configuring DNS clients

1. What DNS servers to use
2. How to use the DNS servers
3. When to use DNS servers
4. When to use /etc/hosts

What DNS servers to use

$ sudo vim /etc/resolv.conf



How to use DNS servers

$ nslookup linux.org.ru
Server: 172.18.96.1
Address: 172.18.96.1#53

Non-authoritative answer:
Name: linux.org.ru
Address: 178.248.233.6

How to use DNS servers

```
$ nslookup university
Server: 172.18.96.1
Address: 172.18.96.1#53

** server can't find university: SERVFAIL
```

How to use DNS servers

```
$ sudo vim /etc/resolv.conf
search innopolis.ru
nameserver 172.18.96.1
```

Suffix for not-fully-qualified hostnames

```
$ sudo service bind9 restart
 * Stopping domain name service... bind9          [ OK ]
 * Starting domain name service... bind9          [ OK ]
$ nslookup university
Server:         172.18.96.1
Address:        172.18.96.1#53

Non-authoritative answer:
Name:   university.innopolis.ru
Address: 185.27.192.220
```

Appended by BIND9

How to use DNS servers

```
$ nslookup portal
Server: 172.18.96.1
Address: 172.18.96.1#53

** server can't find portal: SERVFAIL
```

How to use DNS servers

$ sudo vim /etc/resolv.conf

search innopolis.ru university.innopolis.ru
nameserver 172.18.96.1

Another suffix for not-fully-qualified hostnames (<= 3)

$ sudo service bind9 restart

* Stopping domain name service... bind9 [OK]
* Starting domain name service... bind9 [OK]

$ nslookup portal

Server: 172.18.96.1
Address: 172.18.96.1#53

Appended by BIND9

Non-authoritative answer:
Name: portal.university.innopolis.ru
Address: 188.130.155.250

When to use DNS servers

$ sudo vim /etc/hosts

127.0.0.1 localhost
127.0.1.1 DESKTOP-UOJ09QB.localdomain DESKTOP-UOJ09QB
188.130.155.250 portal

Explicitly specified IP address

$ ping portal

PING portal (188.130.155.250) 56(84) bytes of data.
64 bytes from portal (188.130.155.250): icmp_seq=1 ttl=126 time=2.26 ms

When to use DNS servers

$ sudo vim /etc/nsswitch.conf

```
# /etc/nsswitch.conf
#
passwd:       compat systemd
group:        compat systemd
shadow:       compat
gshadow:      files
hosts:        files dns
networks:     files
protocols:    db files
services:     db files
ethers:       db files
rpc:          db files
netgroup:     nis
```

"First try /etc/hosts, in case of failure proceed with a DNS lookup"

When to use DNS servers

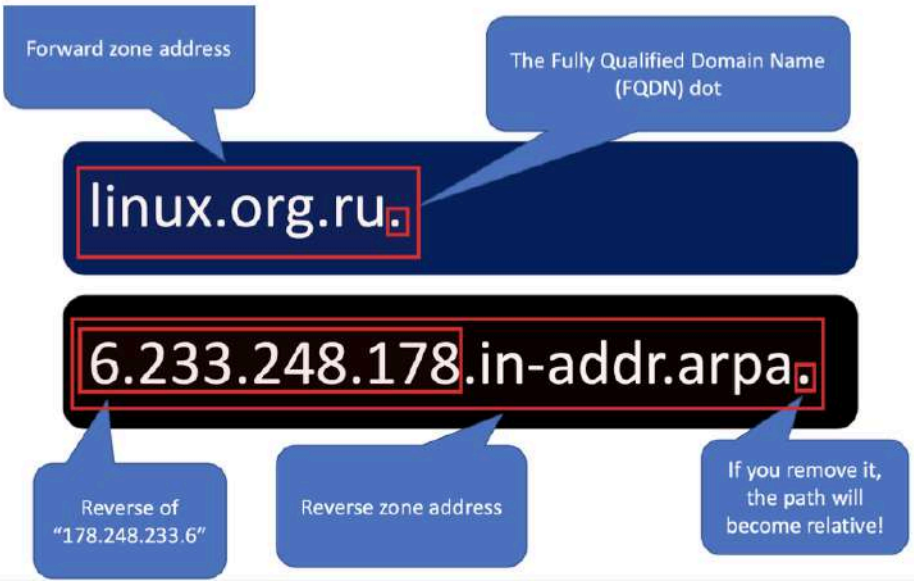
$ sudo vim /etc/nsswitch.conf

```
# /etc/nsswitch.conf
#
passwd:       compat systemd
group:        compat systemd
shadow:       compat
gshadow:      files
hosts:        dns [!NOTFOUND=return] files
networks:     files
protocols:    db files
services:     db files
ethers:       db files
rpc:          db files
netgroup:     nis
```

"Try DNS lookup first; only proceed with /etc/hosts if the DNS lookup does not return NOTFOUND"

The DNS namespace

The DNS namespace

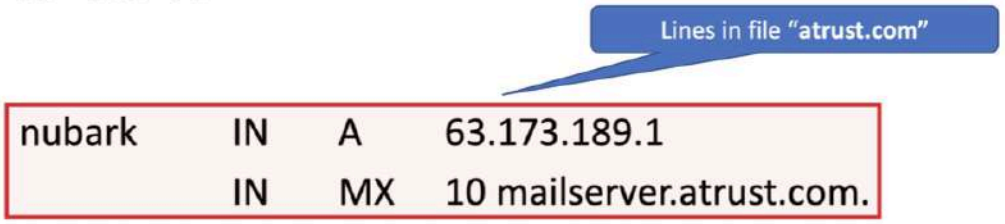


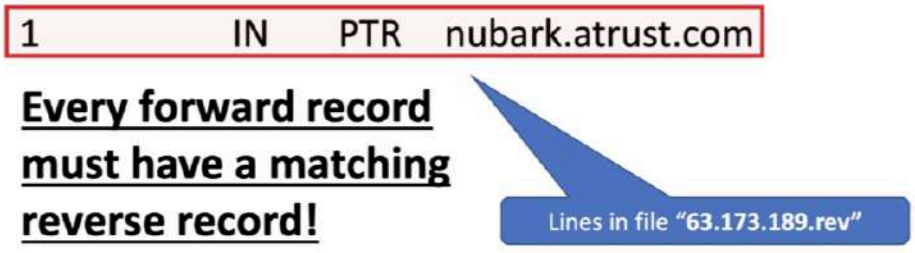
Name servers

- Answer queries about your site's hostnames and IP addresses.
- Ask about both local and remote hosts on behalf of your users.
- Cache the answers to queries to answer faster next time.
- Communicate with other local name servers to keep DNS data in-sync.

Taxonomy of name servers

Resource records





Resource records

Resource record ::= [name] [ttl] [class] type data

- **name** identifies the described entity (repeating name can be omitted)
- **ttl** is the caching period
- **class** specifies the network type (IN for Internet)
- **type** defines the purpose of the record
- **data** defines the useful contents of the record

Resource records

Zone	Basics	Security	Optional
SOA (Start of Authority)	A (IPv4 Address)	DS (Delegation Signer)	CNAME (Canonical Name)
NS (Name Server)	AAAA (IPv6 Address)	DNSKEY (Public Key)	SRV (Service)
	PTR (Pointer)	NSEC (Next Secure)	TXT (Text)
	MX (Mail Exchanger)	NSEC3 (Next Secure v3)	
		RRSIG (Signature)	

The essential ones

SOA record

name	example.com
record type	SOA
MNAME	ns.primaryserver.com
RNAME	admin.example.com
SERIAL	1111111111
REFRESH	86400
RETRY	7200
EXPIRE	4000000
TTL	11200

NS record

example.com	record type:	value:	TTL
@	NS	ns1.exampleserver.com	21600

A record

example.com	record type:	value:	TTL
@	A	192.0.2.1	14400

AAAA record

example.com	record type:	value:	TTL
@	AAAA	2001:0db8:85a3:0000: 0000:8a2e:0370:7334	14400

MX record

example.com	record type:	priority:	value:	TTL
@	MX	10	mailhost1.example.com	45000
@	MX	20	mailhost2.example.com	45000

Users are numbers

```c
#include <sys/types.h>
#include <pwd.h>
#include <stdio.h>

int main() {
    struct passwd *pwd_root = getpwuid(0);
    struct passwd *pwd_bind = getpwnam("bind");
    printf("Name of user 0 = %s\n", pwd_root->pw_name);
    printf("BIND home dir  = %s\n", pwd_bind->pw_dir);
}
```

Gets user info by user ID (UID)

Gets user info by user name

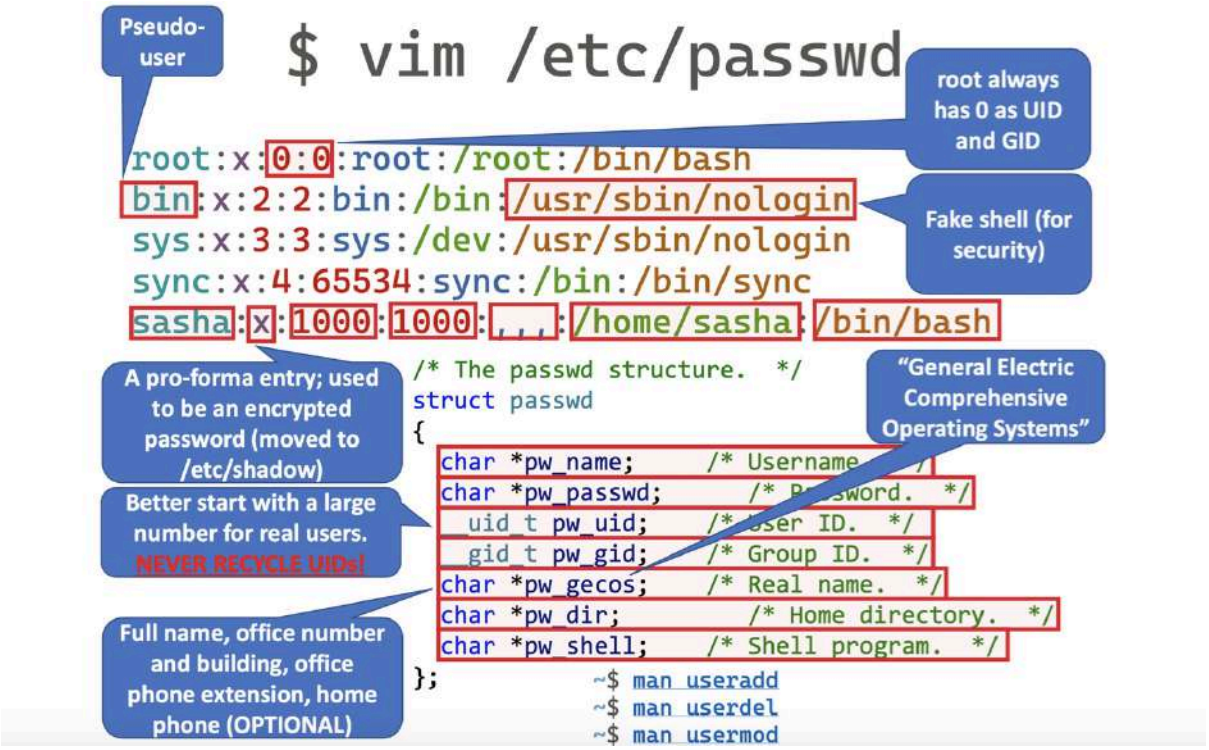
TERMINAL:

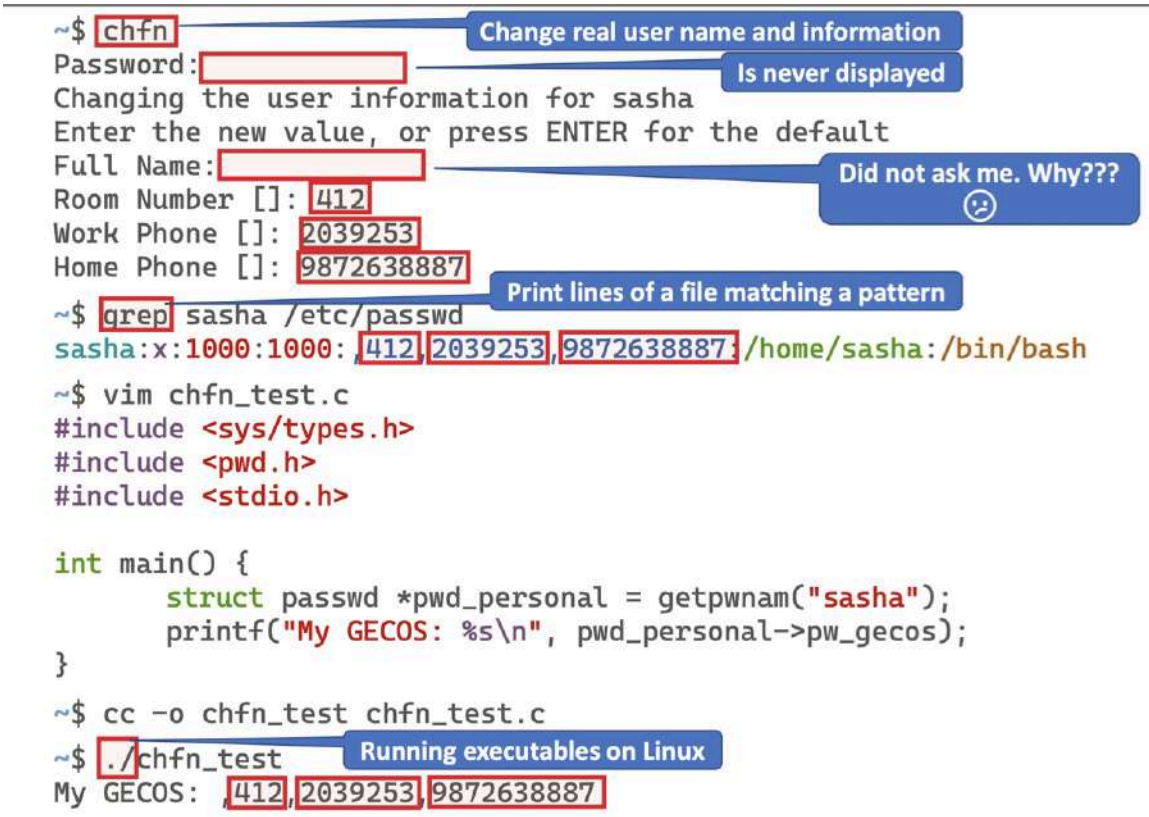
Name of user 0 = root

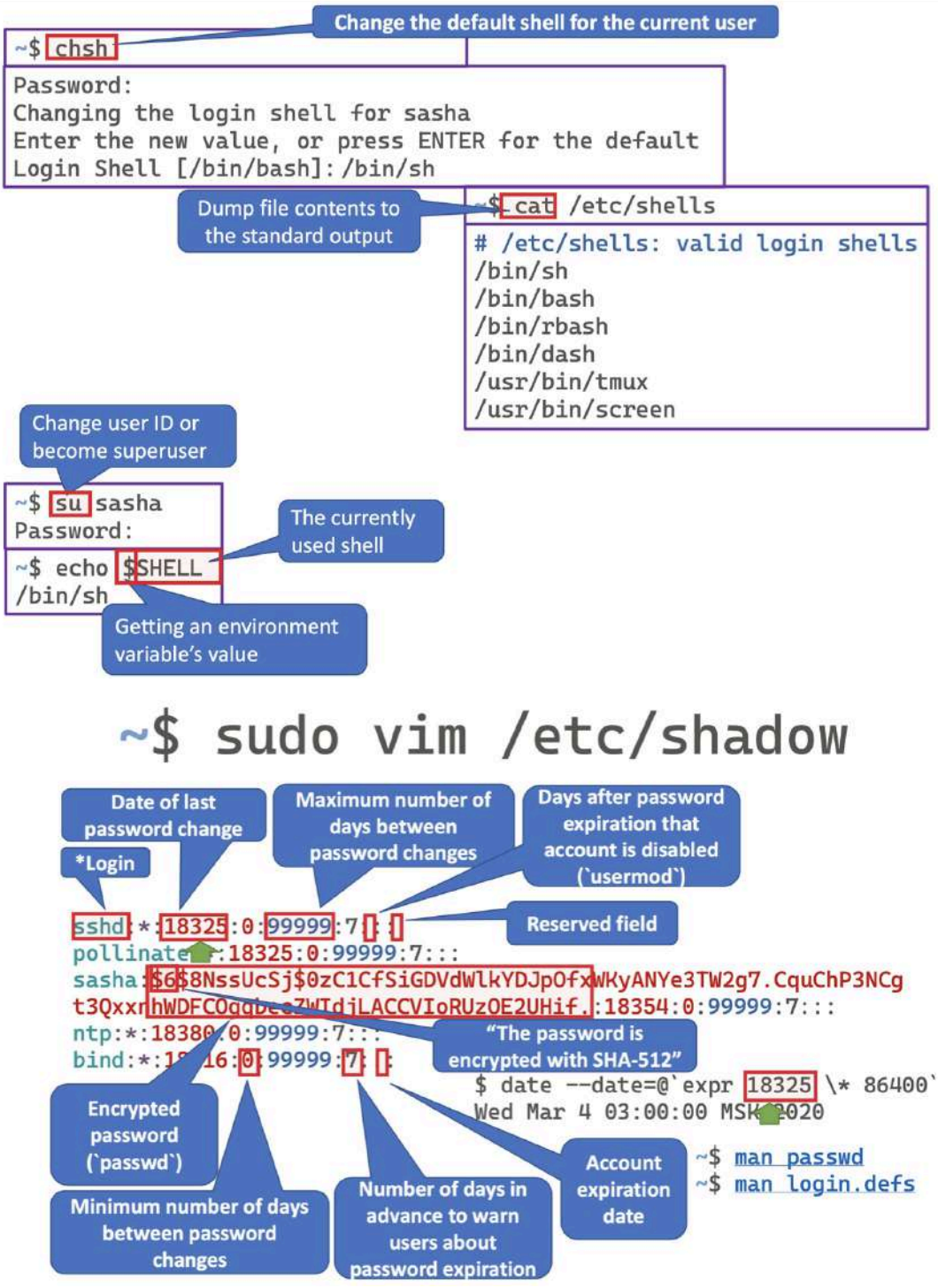
BIND home dir = /var/cache/bind

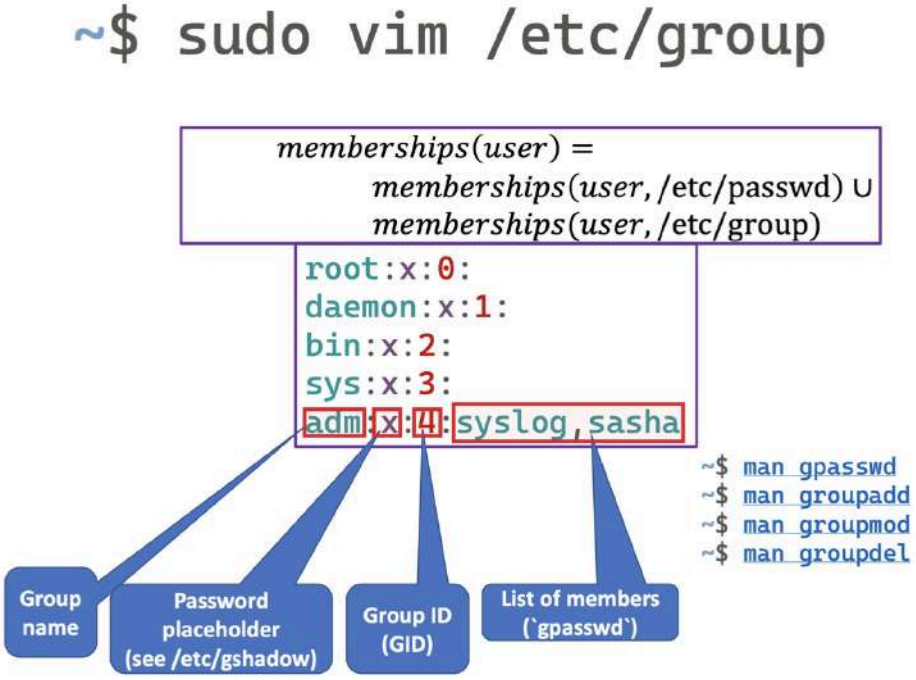
For detailed information

```
~$ man getpwuid
```


$ vim /etc/passwd
Pseudo-user
root always has 0 as UID and GID
Fake shell (for security)
root:x:0:0:root:/root:/bin/bash
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
sasha:x:1000:1000:,,,:/home/sasha:/bin/bash
A pro-forma entry; used to be an encrypted password (moved to /etc/shadow)
Better start with a large number for real users.
NEVER RECYCLE UIDs!
Full name, office number and building, office phone extension, home phone (OPTIONAL)
"General Electric Comprehensive Operating Systems"
/* The passwd structure. */
struct passwd
{
char *pw_name; /* Username */
char *pw_passwd; /* Password. */
__uid_t pw_uid; /* User ID. */
__gid_t pw_gid; /* Group ID. */
char *pw_gecos; /* Real name. */
char *pw_dir; /* Home directory. */
char *pw_shell; /* Shell program. */
};
~$ man useradd
~$ man userdel
~$ man usermod


~$ chfn
Change real user name and information
Password:
Is never displayed
Changing the user information for sasha
Enter the new value, or press ENTER for the default
Full Name:
Did not ask me. Why??? ☹
Room Number []: 412
Work Phone []: 2039253
Home Phone []: 9872638887
Print lines of a file matching a pattern
~$ grep sasha /etc/passwd
sasha:x:1000:1000:,412,2039253,9872638887:/home/sasha:/bin/bash
~$ vim chfn_test.c
#include <sys/types.h>
#include <pwd.h>
#include <stdio.h>
int main() {
struct passwd *pwd_personal = getpwnam("sasha");
printf("My GECOS: %s\n", pwd_personal->pw_gecos);
}
~$ cc -o chfn_test chfn_test.c
~$ ./chfn_test
Running executables on Linux
My GECOS: ,412,2039253,9872638887


Change the default shell for the current user
~$ chsh
Password:
Changing the login shell for sashа
Enter the new value, or press ENTER for the default
Login Shell [/bin/bash]: /bin/sh
Dump file contents to the standard output
~$ cat /etc/shells
/etc/shells: valid login shells
/bin/sh
/bin/bash
/bin/rbash
/bin/dash
/usr/bin/tmux
/usr/bin/screen
Change user ID or become superuser
~$ su sasha
Password:
~$ echo $SHELL
/bin/sh
The currently used shell
Getting an environment variable's value
~$ sudo vim /etc/shadow
Date of last password change
Maximum number of days between password changes
Days after password expiration that account is disabled (`usermod`)
*Login
Reserved field
sshd:*:18325:0:99999:7:::
pollinate:*:18325:0:99999:7:::
sasha:$6$8NssUcSj$0zC1CfSiGDVdWlkYDJpOfxWKyANYe3TW2g7.CquChP3NCg
t3QxxrhWDFCOqqbee7WIdjLACCVIoRUzOE2UHif.:18354:0:99999:7:::
ntp:*:18380:0:99999:7:::
bind:*:18416:0:99999:7:::
"The password is encrypted with SHA-512"
$ date --date=@`expr 18325 \* 86400`
Wed Mar 4 03:00:00 MSK 2020
Encrypted password (`passwd`)
Minimum number of days between password changes
Number of days in advance to warn users about password expiration
Account expiration date
~$ man passwd
~$ man login.defs



Identity & access management: key concepts

- *User identity*
  - abstract representation of an individual who needs access to a system or application.
- *Authentication*
  - the act of checking if an individual is the legitimate owner of an identity.
- *Authorization*
  - the act of checking if an authenticated individual has the right to perform the requested operation.
- *Single Sign-On (SSO)*
  - technology letting users log in to several systems with a single authentication token.

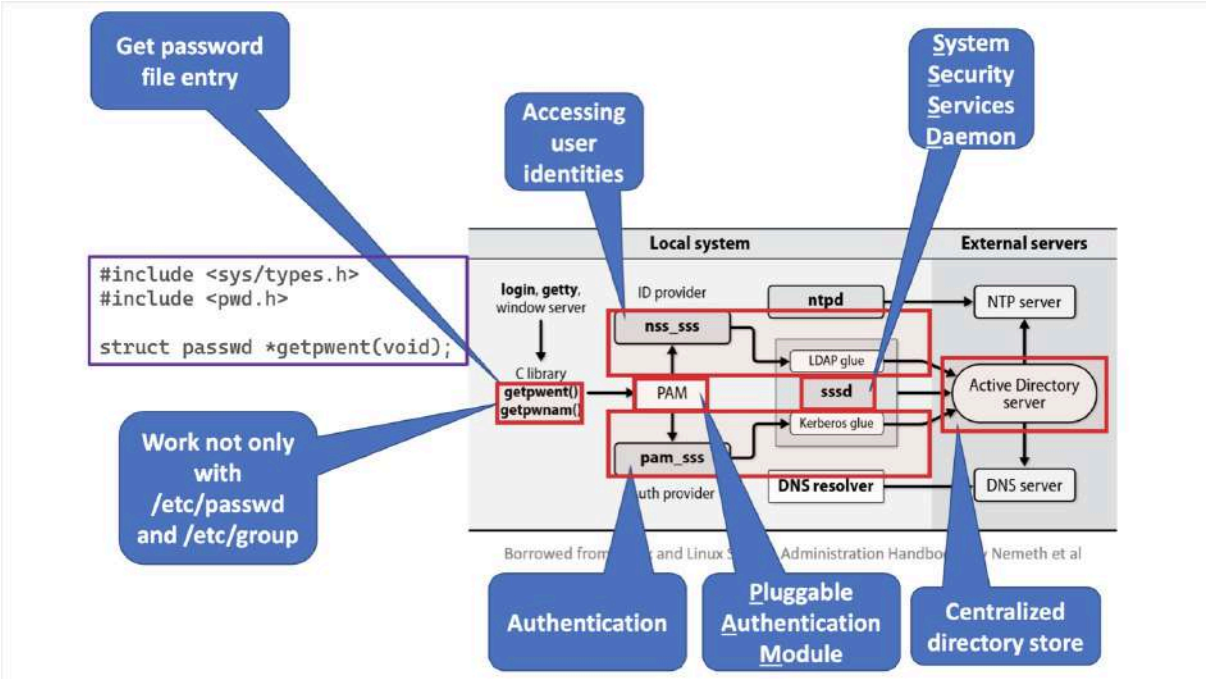
Core elements

- A centralized directory store that contains user identity and authorization information; typically relies on **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP)
  - Microsoft Active Directory (AD)

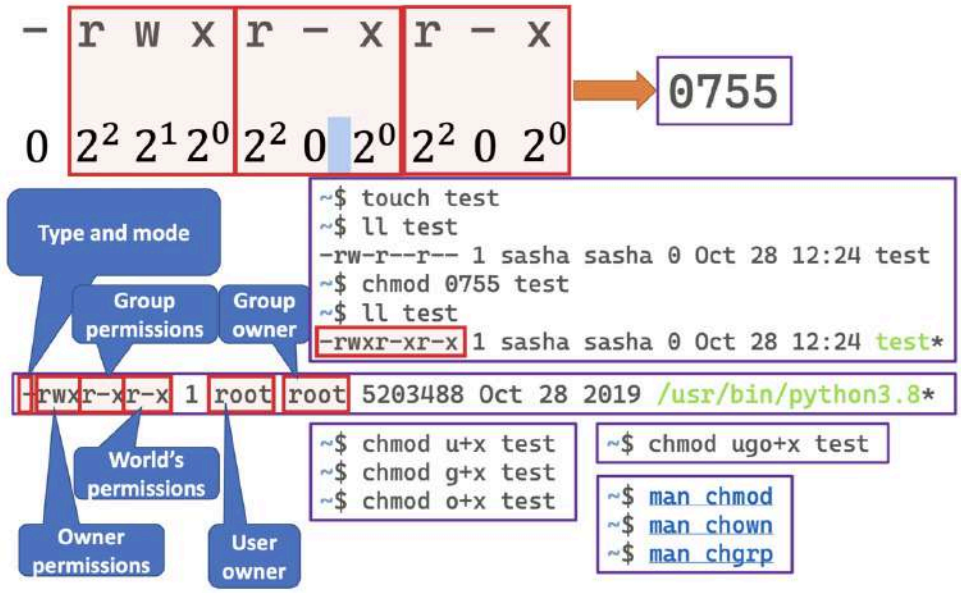
- A tool for managing user information in the directory.
  - phpLDAPadmin
  - ApacheDirectoryStudio
  - AD Users and Computers

- A mechanism for authenticating user identities.
  - Directly against LDAP
  - Kerberos ticket-based system

- Centralized-identity-and-authentication-aware versions of the C library routines that look up user attributes.
  - Data sources are configured in /etc/nsswitch.conf
  - Default to /etc/passwd, /etc/group, /etc/shadow, /etc/gshadow



ll /usr/bin/python3.8



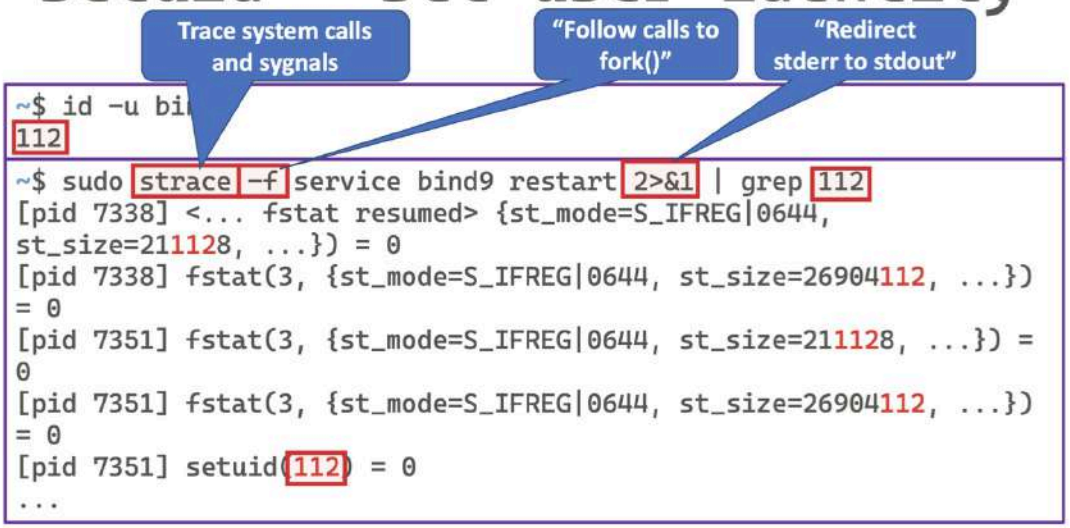
setuid - set user identity

```
~$ ps u -C named
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
bind 6749 0.0 0.2 739520 27340 ? Ssl 12:54 0:00 /usr/sbin/named -u bind
```

```
bind:x:112:118::/var/cache/bind:/usr/sbin/nologin
```

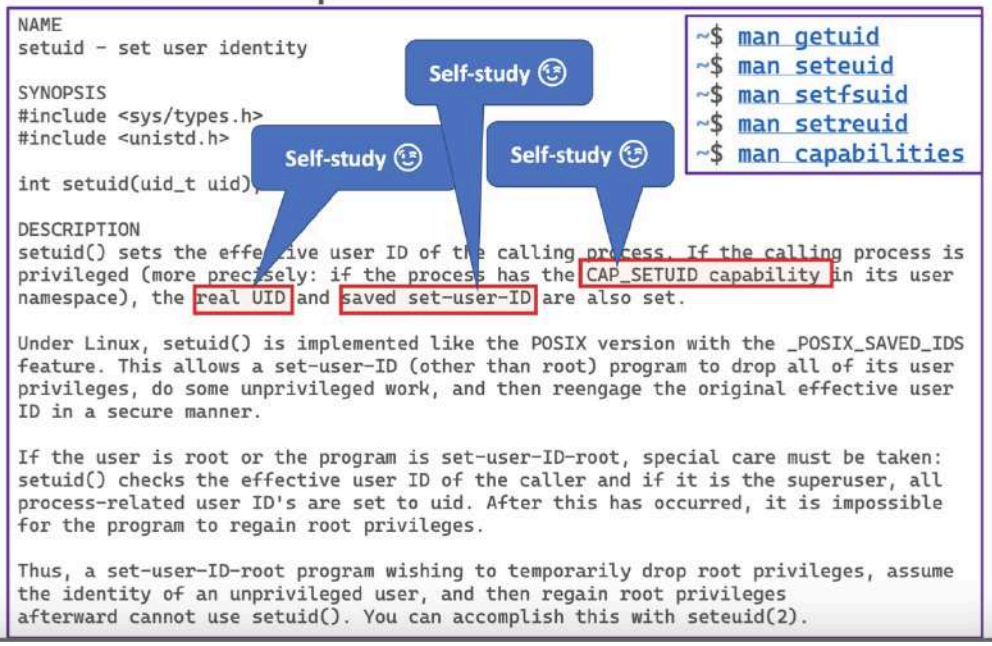
```
~$ service bind9 restart
* Stopping domain name service... bind9
rndc: error: open: /etc/bind/rndc.key: permission denied
rndc: could not load rndc configuration
[ OK ]
* Starting domain name service... bind9
chmod: changing permissions of '/run/named': Operation not
permitted
```

setuid – set user identity




```
~$ man strace
```

~$ man setuid



The old algorithm for having a grilled steak:

1. Go for hunting with a group of men.
2. Find and kill a buffalo.
3. Remove the skin.
4. Remove the guts.
5. Cut it into pieces.
6. Light a fire.
7. Grill it on the fire making sure it does not burn.

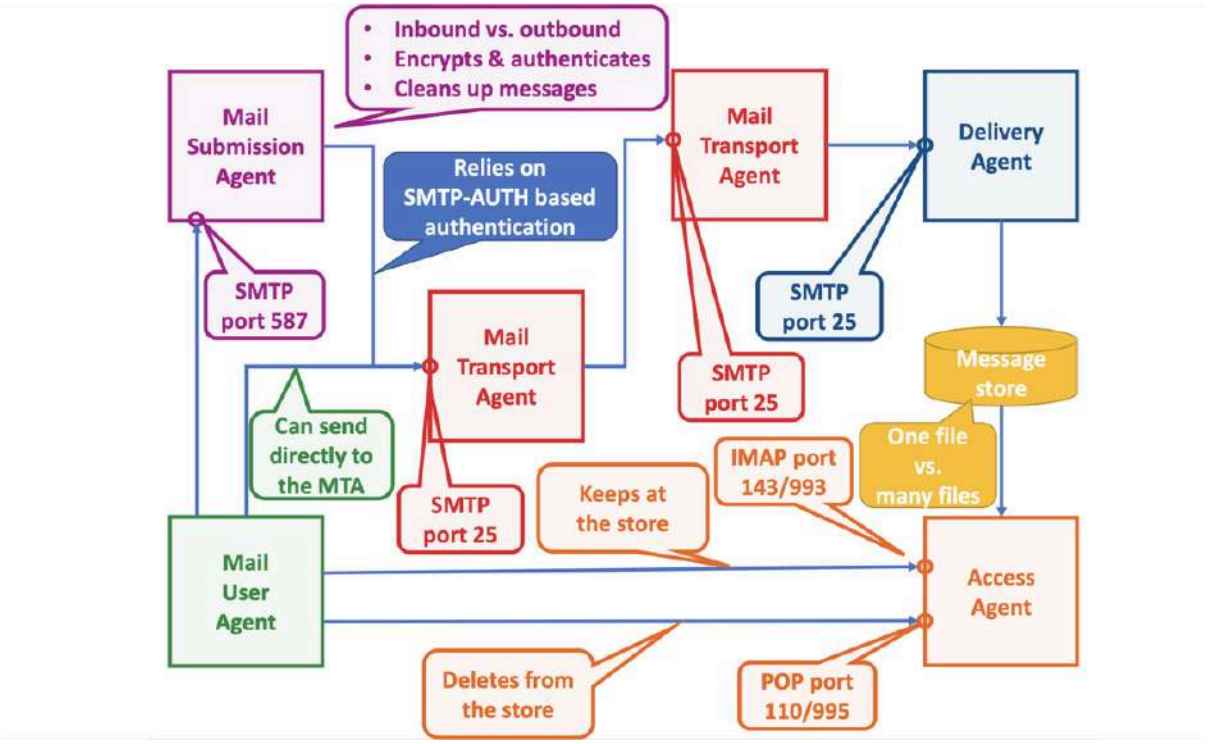
The modern algorithm:

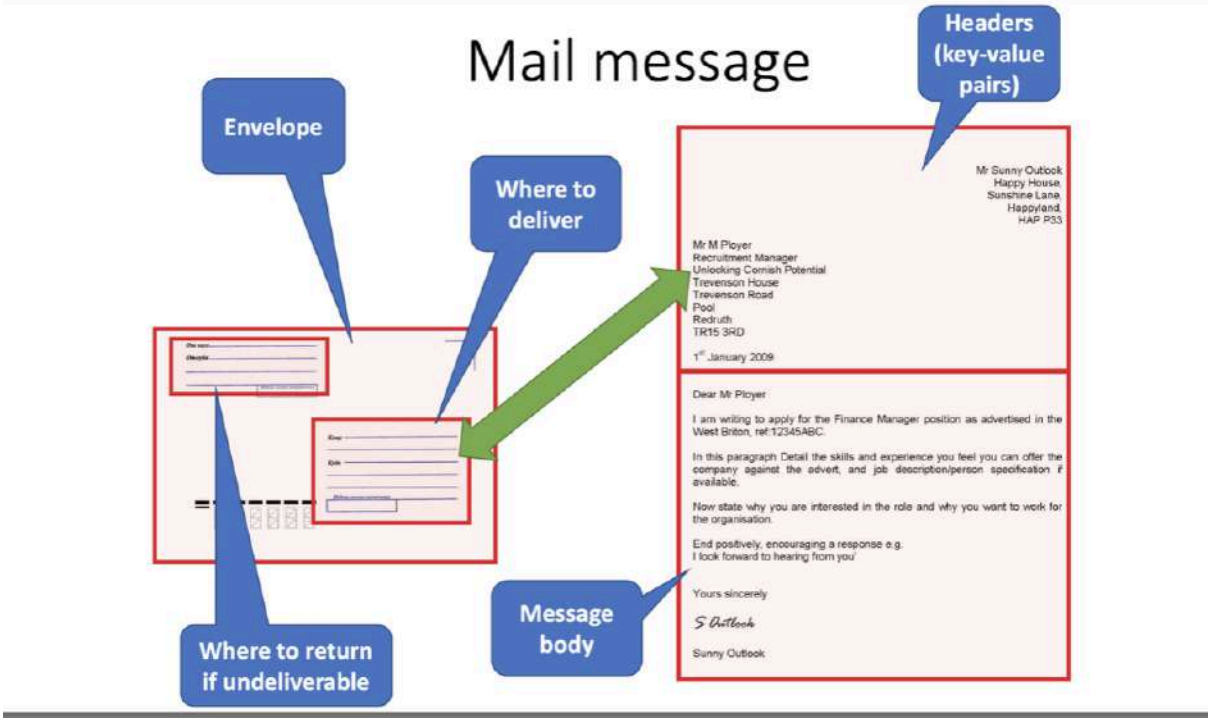
1. Go to the nearest grocery store.
2. Buy uniformly cut pieces and bring them home.
3. Put them into your electric grill.
4. Select the desired roast level and wait for a signal.

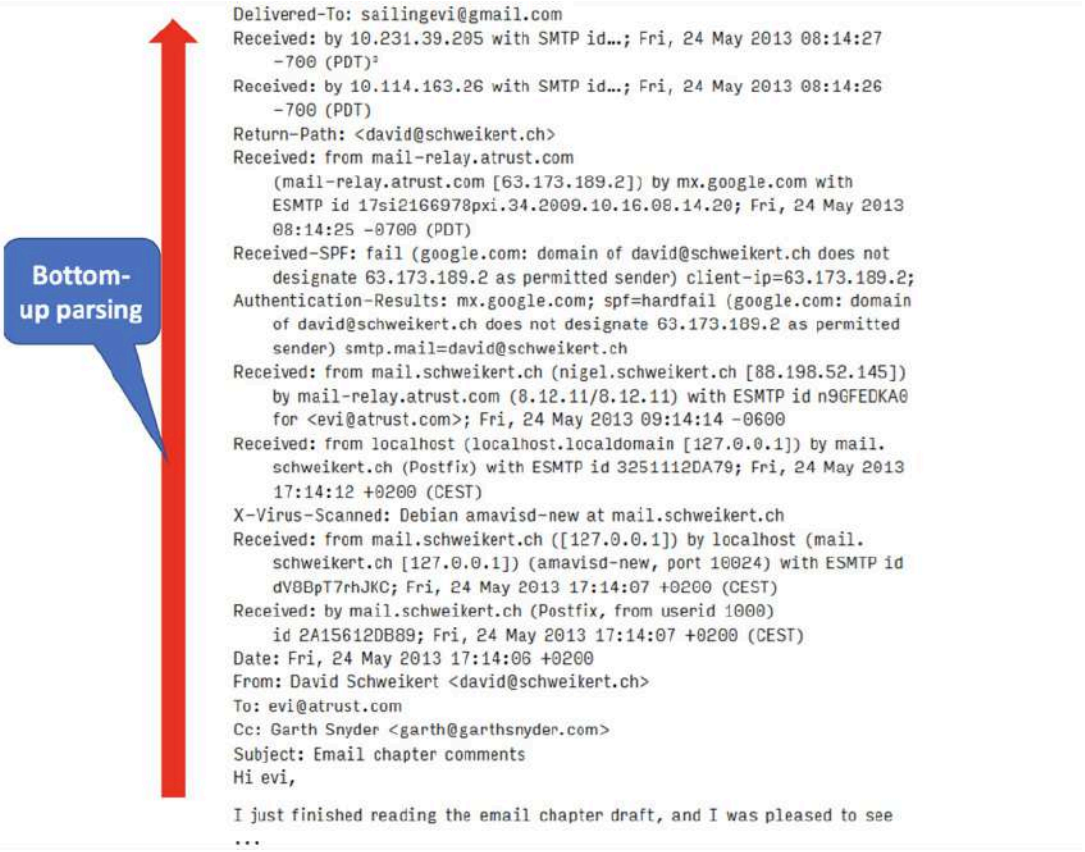
Key components of an email system

- Mail user agents (MUA) read and compose mail.
  - Thunderbird, Outlook, macOS Mail, Alpine, /bin/mail
- Mail submission agents (MSA) pack outgoing mail from the MUAs and submit it to the transport system (SMTP).
- Mail transport agents (MTA) route messages among machines (SMTP).
  - sendmail, Exim, Exchange, Postfix
- Delivery agents (DA) place messages in a local store (SMTP).
  - procmail, Maildrop
- Access agents (AA) connect MUAs to the store (IMAP/POP).
  - Cyrus, UW IMAP

The original user agent

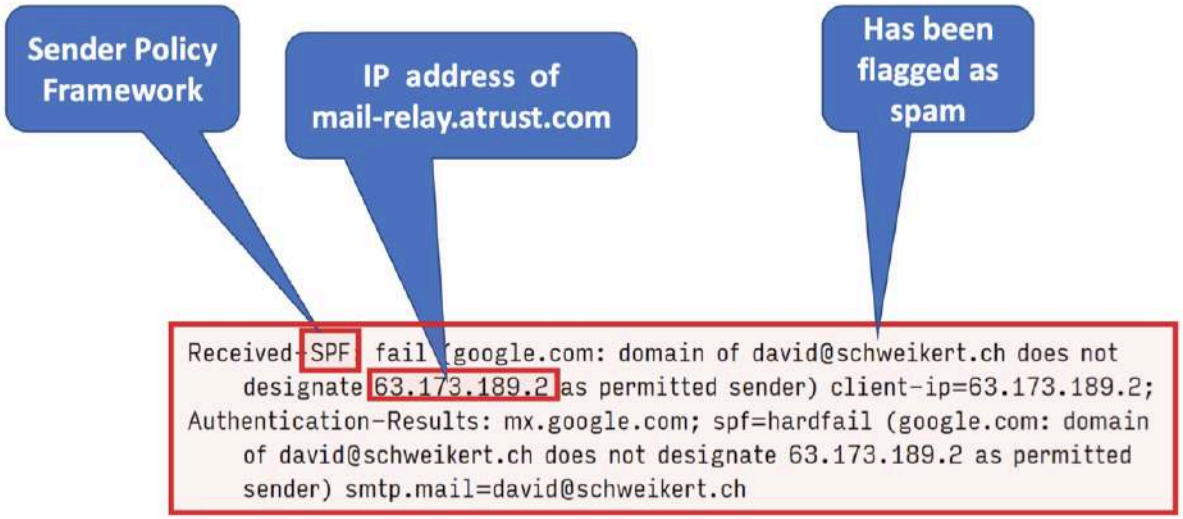

Inbound vs. outbound
Encrypts & authenticates
Cleans up messages
Mail Submission Agent
Relies on SMTP-AUTH based authentication
Mail Transport Agent
Delivery Agent
SMTP port 587
Mail Transport Agent
SMTP port 25
SMTP port 25
Message store
Can send directly to the MTA
One file vs. many files
IMAP port 143/993
SMTP port 25
Keeps at the store
Mail User Agent
Access Agent
Deletes from the store
POP port 110/995

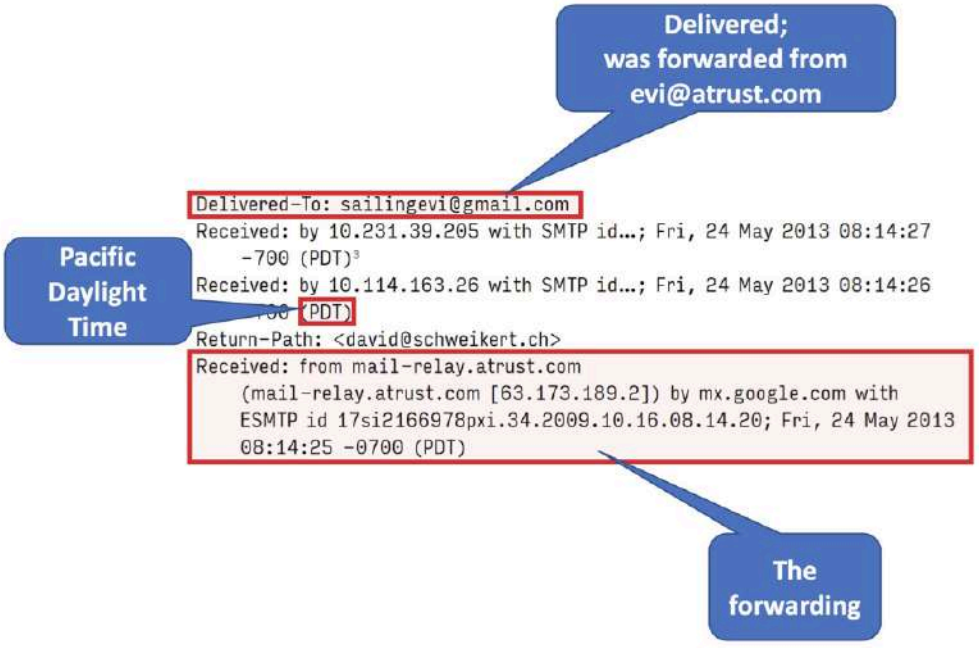

Mail message
Headers (key-value pairs)
Envelope
Where to deliver
Mr Sunny Outlook
Happy House
Sunshine Lane
Happyland
HAP P33
Mr M Ployer
Recruitment Manager
Unlooking Cornish Potential
Trevenson House
Trevenson Road
Pool
Redruth
TR15 3RD
1st January 2009
Dear Mr Ployer
I am writing to apply for the Finance Manager position as advertised in the West Briton, ref:12345ABC.
In this paragraph Detail the skills and experience you feel you can offer the company against the advert, and job description/person specification if available.
Now state why you are interested in the role and why you want to work for the organisation.
End positively, encouraging a response e.g.
I look forward to hearing from you'
Yours sincerely
S Outlook
Sunny Outlook
Message body
Where to return if undeliverable

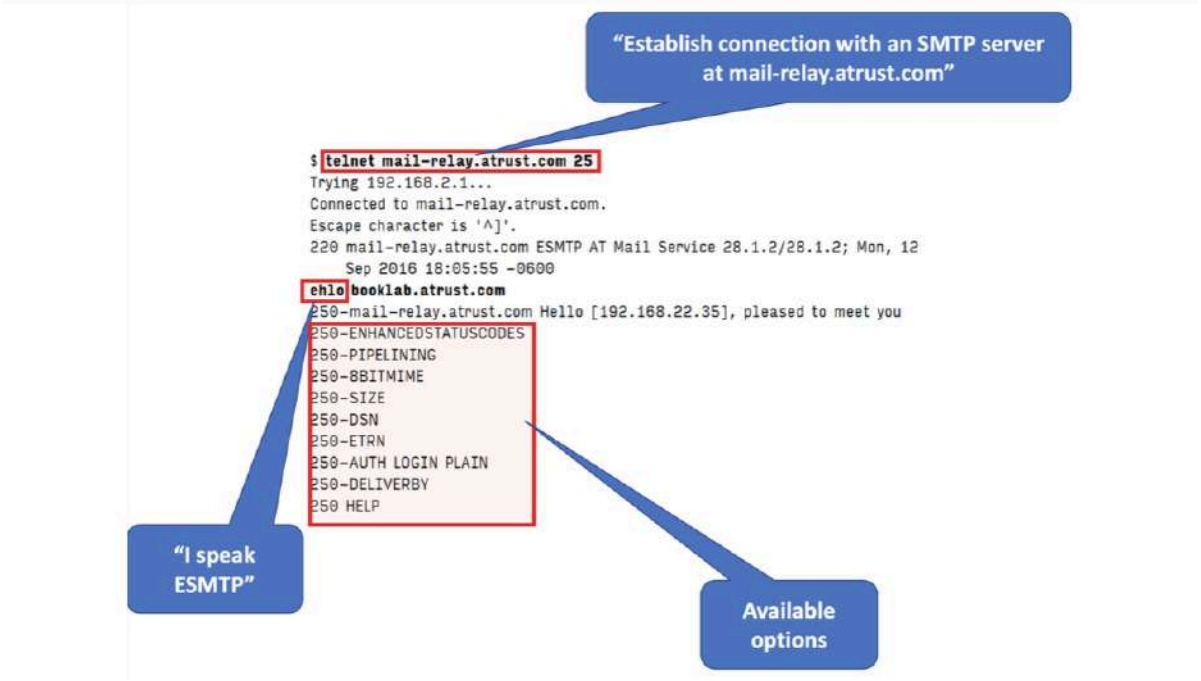

Bottom-up parsing
Delivered-To: sailingevi@gmail.com
Received: by 10.231.39.205 with SMTP id…; Fri, 24 May 2013 08:14:27 -700 (PDT)
Received: by 10.114.163.26 with SMTP id…; Fri, 24 May 2013 08:14:26 -700 (PDT)
Return-Path: <david@schweikert.ch>
Received: from mail-relay.atrust.com (mail-relay.atrust.com [63.173.189.2]) by mx.google.com with ESMTP id 17si2166978pxi.34.2009.10.16.08.14.20; Fri, 24 May 2013 08:14:25 -0700 (PDT)
Received-SPF: fail (google.com: domain of david@schweikert.ch does not designate 63.173.189.2 as permitted sender) client-ip=63.173.189.2;
Authentication-Results: mx.google.com; spf=hardfail (google.com: domain of david@schweikert.ch does not designate 63.173.189.2 as permitted sender) smtp.mail=david@schweikert.ch
Received: from mail.schweikert.ch (nigel.schweikert.ch [88.198.52.145]) by mail-relay.atrust.com (8.12.11/8.12.11) with ESMTP id n9GFEDKA0 for <evi@atrust.com>; Fri, 24 May 2013 09:14:14 -0600
Received: from localhost (localhost.localdomain [127.0.0.1]) by mail.schweikert.ch (Postfix) with ESMTP id 3251112DA79; Fri, 24 May 2013 17:14:12 +0200 (CEST)
X-Virus-Scanned: Debian amavisd-new at mail.schweikert.ch
Received: from mail.schweikert.ch ([127.0.0.1]) by localhost (mail.schweikert.ch [127.0.0.1]) (amavisd-new, port 10024) with ESMTP id dV8BpT7rhJKC; Fri, 24 May 2013 17:14:07 +0200 (CEST)
Received: by mail.schweikert.ch (Postfix, from userid 1000) id 2A15612DB89; Fri, 24 May 2013 17:14:07 +0200 (CEST)
Date: Fri, 24 May 2013 17:14:06 +0200
From: David Schweikert <david@schweikert.ch>
To: evi@atrust.com
Cc: Garth Snyder <garth@garthsnyder.com>
Subject: Email chapter comments
Hi evi,
I just finished reading the email chapter draft, and I was pleased to see
...


Was scanned for viruses
X-Virus-Scanned: Debian amavisd-new at mail.schweikert.ch
Received: from mail.schweikert.ch ([127.0.0.1]) by localhost (mail.schweikert.ch [127.0.0.1]) (amavisd-new, port 10024) with ESMTP id dV8BpT7rhJKC; Fri, 24 May 2013 17:14:07 +0200 (CEST)
Received: by mail.schweikert.ch (Postfix, from userid 1000) id 2A15612DB89; Fri, 24 May 2013 17:14:07 +0200 (CEST)
Received by the Postfix MTA mail server
Date: Fri, 24 May 2013 17:14:06 +0200
From: David Schweikert <david@schweikert.ch>
To: evi@atrust.com
From/To
Cc: Garth Snyder <garth@garthsnyder.com>
Subject: Email chapter comments
Central European Standard Time


Sending host
Received: from mail.schweikert.ch (nigel.schweikert.ch [88.198.52.145])
by mail-relay.atrust.com (8.12.11/8.12.11) with ESMTP id n9GFEDKA0
for <e1@atrust.com>; Fri, 24 May 2013 09:14:14 -0600
Receiving MTA host


Sender Policy Framework
IP address of mail-relay.atrust.com
Has been flagged as spam
Received-SPF: fail (google.com: domain of david@schweikert.ch does not designate 63.173.189.2 as permitted sender) client-ip=63.173.189.2;
Authentication-Results: mx.google.com; spf=hardfail (google.com: domain of david@schweikert.ch does not designate 63.173.189.2 as permitted sender) smtp.mail=david@schweikert.ch


Delivered; was forwarded from evi@atrust.com
Pacific Daylight Time
Delivered-To: sailingevi@gmail.com
Received: by 10.231.39.205 with SMTP id...; Fri, 24 May 2013 08:14:27 -700 (PDT)
Received: by 10.114.163.26 with SMTP id...; Fri, 24 May 2013 08:14:26 -700 (PDT)
Return-Path: <david@schweikert.ch>
Received: from mail-relay.atrust.com (mail-relay.atrust.com [63.173.189.2]) by mx.google.com with ESMTP id 17si2166978pxi.34.2009.10.16.08.14.20; Fri, 24 May 2013 08:14:25 -0700 (PDT)
The forwarding

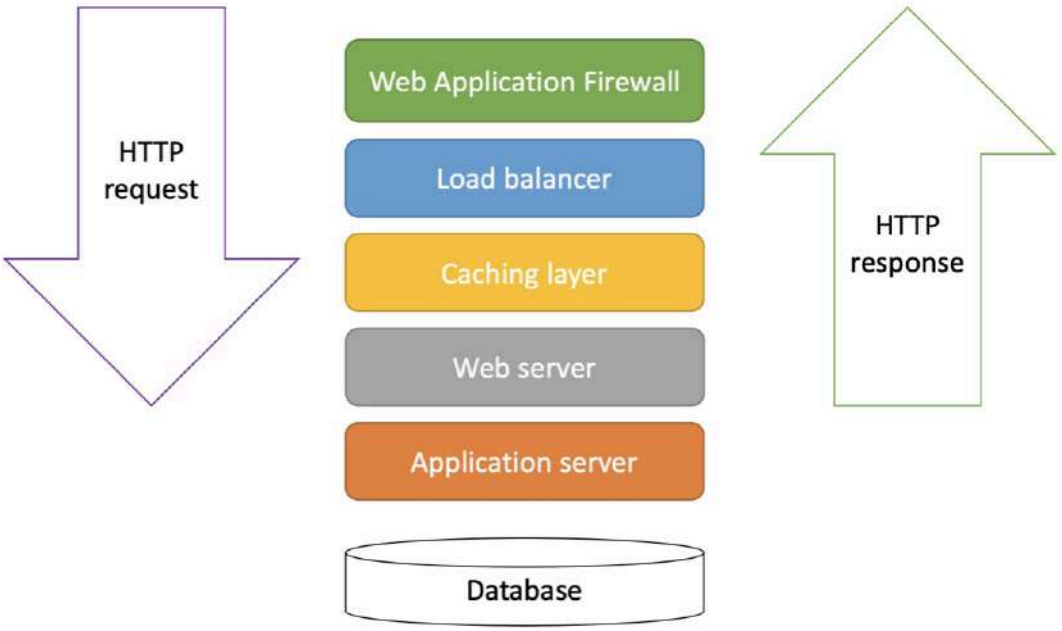


https://datatracker.ietf.org/doc/html/rfc4954
https://datatracker.ietf.org/doc/html/rfc3501
https://datatracker.ietf.org/doc/html/rfc5322

7. Web services

INNOPOLIS UNIVERS

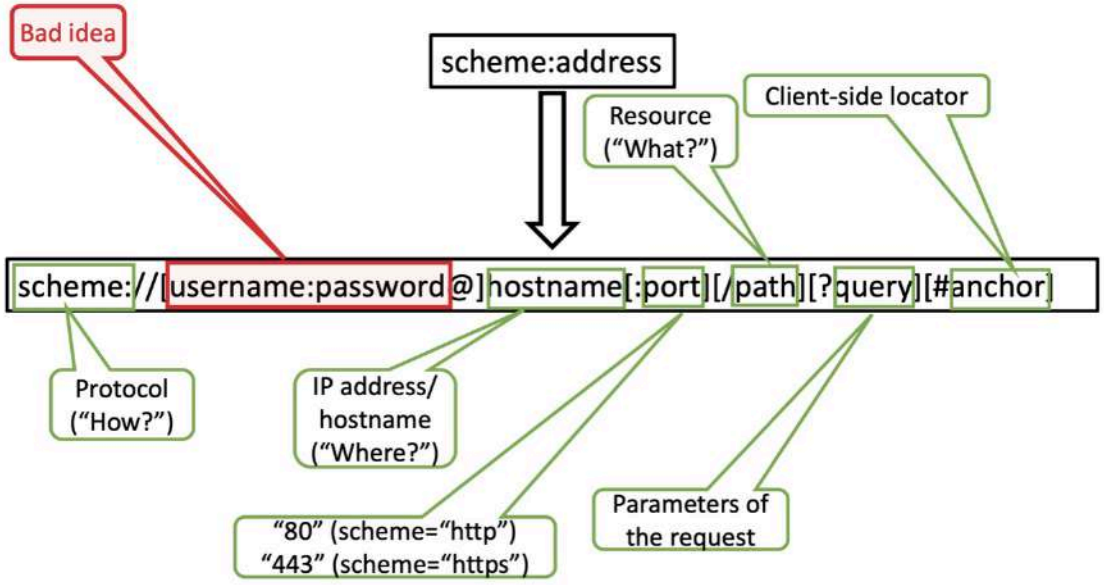
Web application stack



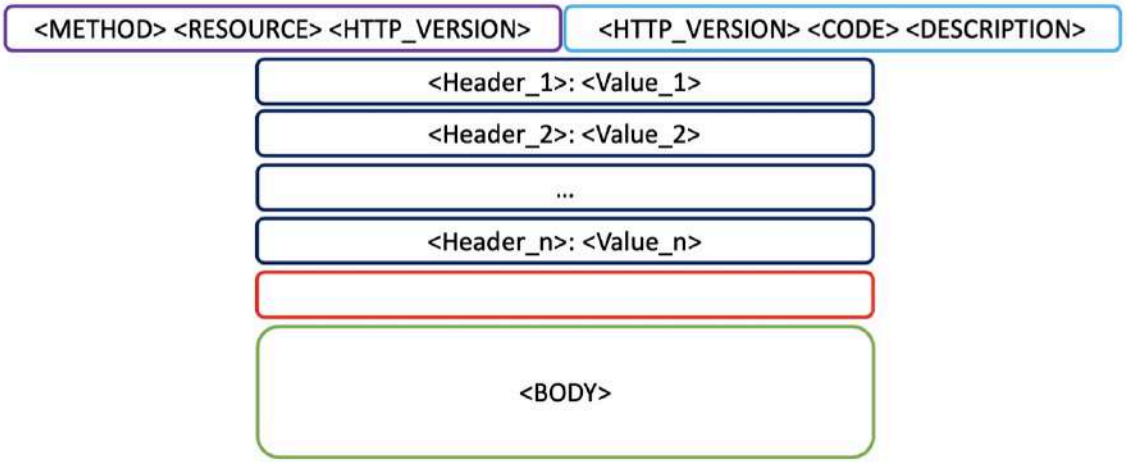
The HyperText Transfer Protocol

- The consumer facing protocol of the Web
- HTTP 1.0/1.1
  - Transfer plain text
  - Simplify reading & debugging
- HTTP/2
  - Maintain backward compatibility
  - Transfer binary data to improve performance
  - Complicate reading & debugging

Unified Resource Locators



HTTP message structure

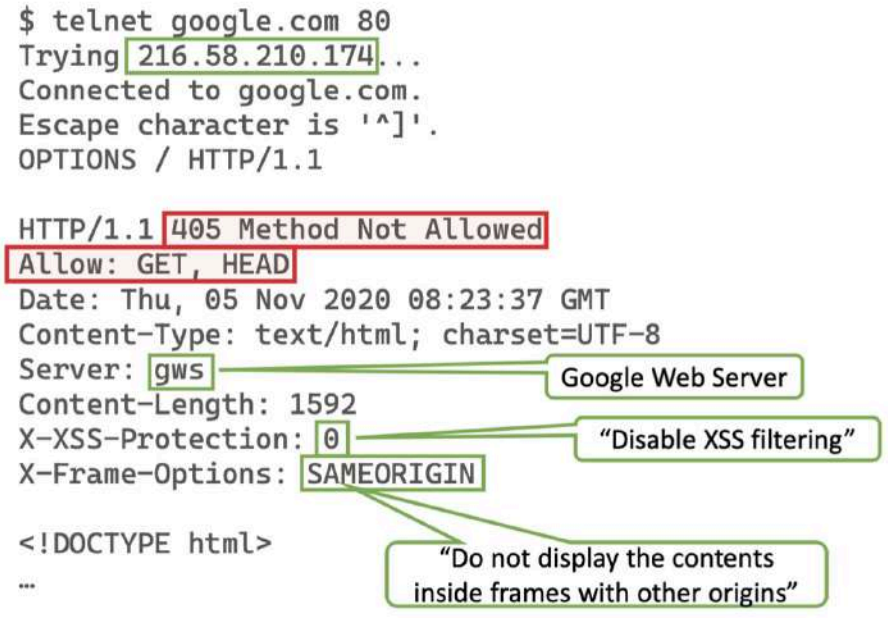


HTTP request methods

Method	Safe?	Description
GET	👍	Retrieve the resource
HEAD	👍	GET without the body
DELETE	👎	Delete the resource
POST	👎	Apply the request to the resource
PUT	👎	Replace existing contents
OPTIONS	👍	Show supported methods

HTTP response classes

Code	General indication
1xx	Request received; processing continues
2xx	Success
3xx	Further action needed
4xx	Unsatisfiable request
5xx	Server or environment failure



```
$ telnet 216.58.210.174 80
Trying 216.58.210.174...
Connected to 216.58.210.174.
Escape character is '^]'.
HEAD /search?q=linux HTTP/1.1

HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Date: Thu, 05 Nov 2020 08:57:41 GMT
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
Expires: Thu, 05 Nov 2020 08:57:41 GMT
Cache-Control: private
Set-Cookie: 1P_JAR=2020-11-05-08; expires=Sat, 05-Dec-2020 08:57:41
GMT; path=/; domain=.google.com; Secure
Set-Cookie: CGIC=; expires=Tue, 04-May-2021 08:57:41 GMT;
path=/complete/search; domain=.google.com; HttpOnly
...
```

"The response may be stored only by a browser's cache"

"Only send the cookie over HTTPS"

"Do not let JavaScript access this cookie"

```
$ telnet 216.58.210.174 80
Trying 216.58.210.174...
Connected to 216.58.210.174.
Escape character is '^]'.
POST / HTTP/1.1

HTTP/1.0 411 Length Required
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 1564
Date: Thu, 05 Nov 2020 10:08:57 GMT

<!DOCTYPE html>
...
```

The server need to know when to stop reading the body of the POST request

```
$ telnet 216.58.210.174 80
Trying 216.58.210.174...
Connected to 216.58.210.174.
Escape character is '^]'.
POST /search HTTP/1.1
Content-length: 7

q=linux
HTTP/1.1 405 Method Not Allowed
Allow: GET, HEAD
Date: Thu, 05 Nov 2020 10:13:07 GMT
Content-Type: text/html; charset=UTF-8
Server: gws
Content-Length: 1595
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
…
```

Commonly encountered headers

Name: example value	Direction
Host: www.admin.com	C
Content-Type: application/json	CS
Last-Modified: Wed, Sep 7 2016	S
Cookie: flavor=oatmeal	C
Content-Length: 423	CS
Set-Cookie: flavor=oatmeal	S
User-Agent: curl/7.37.1	C
Server: nginx/1.6.2	S
Upgrade: HTTP/2.0	C
Expires: Sat, 15 Oct 2016 14:02:...	S
Cache-Control: max-age=7200	CS
Referer: https://developer.mozilla.org/en-US/docs/Web/JavaScript	C

HTTP server types

Type	Purpose	Examples
Application server	Runs web app code, interfaces to web servers	Unicorn, Tomcat
Cache	Speeds access to frequently requested content	Varnish, Squid
Load balancer	Relays requests to downstream systems	Pound, HAProxy
Web app firewall	Inspects HTTP traffic for common attacks	ModSecurity
Web server	Serves static content, couples to other servers	Apache, NGINX

Virtual hosting requests

```
$ telnet 185.114.245.108 80
Trying 185.114.245.108...
Connected to 185.114.245.108.
Escape character is '^]'.
GET / HTTP/1.1
Host: innopolis.ru

HTTP/1.1 301 Moved Permanently
Server: nginx/1.16.1
Date: Thu, 05 Nov 2020 10:48:12 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://innopolis.ru/

<html>
...
```

Reuse TCP connection

Redirection

https://datatracker.ietf.org/doc/html/rfc2616
https://www.hjp.at/doc/rfc/rfc7034.html

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

Exercise 1 - Text processing:

Questions to answer:

1. What is /etc/apt/sources.list.d directory stands for? How can you use it? Provide an example.

By using directory */etc/apt/sources.list.d* we are going to be able to add new repositories to Ubuntu, without changing the */etc/apt/sources.list* file itself. We can add a new file of .list format to the directory and it would be usable by apt. The same thing is going to happen, if we want to delete a repository, we can delete it just from the directory, without dealing with the */etc/apt/sources.list* file. This directory is making our repository management more useful and easier. Below, you can see example of adding the repo into directory:

```
root@Nurbek001:~# vim /etc/apt/sources.list.d/newlist.list
```

```
root@Nurbek001:~# ls /etc/apt/sources.list.d
newlist.list
root@Nurbek001:~#
```

2. How can you add/delete 3rd party repositories to install required software? Provide an example.

Adding 3rd party repo:

```
root@Nurbek001:~# sudo sh -c "echo deb https://deb.nodesource.com/node_15.x focal main > /etc/apt/sources.l
root@Nurbek001:~# sudo apt-get update
Hit:1 http://ru.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Ign:5 http://archive.getdeb.net/ubuntu wily-getdeb InRelease
Get:6 https://deb.nodesource.com/node_15.x focal InRelease [4583 B]
Get:7 http://archive.canonical.com/ubuntu focal InRelease [12.1 kB]
Get:8 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1298 kB]
Err:9 http://archive.getdeb.net/ubuntu wily-getdeb Release
  404  Not Found [IP: 172.67.145.187 80]
Get:10 http://ru.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [269 kB]
Get:11 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.4 kB]
Get:12 http://ru.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [523 kB]
Get:13 http://ru.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [75.0 kB]
Get:14 http://ru.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [866 kB]
Get:15 http://ru.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [19.3 kB]
Get:16 http://ru.archive.ubuntu.com/ubuntu focal-security/main amd64 Packages [945 kB]
Get:17 http://ru.archive.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [8844 B]
Get:18 http://ru.archive.ubuntu.com/ubuntu focal-security/universe amd64 Packages [647 kB]
Get:19 http://ru.archive.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [12.7 kB]
Get:20 http://archive.canonical.com/ubuntu focal/partner amd64 Packages [856 B]
Get:21 http://archive.canonical.com/ubuntu focal/partner Translation-en [384 B]
Get:22 https://deb.nodesource.com/node_15.x focal/main amd64 Packages [770 B]
Reading package lists... Done
E: The repository 'http://archive.getdeb.net/ubuntu wily-getdeb Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@Nurbek001:~# sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 140 not upgraded.
Need to get 25.5 MB of archives.
After this operation, 116 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_15.x focal/main amd64 nodejs amd64 15.14.0-deb-1nodesource1 [25.5 MB]
Fetched 25.5 MB in 1s (29.2 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 122062 files and directories currently installed.)
Preparing to unpack .../nodejs_15.14.0-deb-1nodesource1_amd64.deb ...
```

Deleting 3rd party repo:

```
root@Nurbek001:~# ls /etc/apt/sources.list.d
canonical_partner.list  newlist.list  nodesource.list
root@Nurbek001:~# sudo rm -i /etc/apt/source.list.d/nodesource.list
rm: cannot remove '/etc/apt/source.list.d/nodesource.list': No such file or directory
root@Nurbek001:~# sudo rm -i /etc/apt/sources.list.d/nodesource.list
rm: remove regular file '/etc/apt/sources.list.d/nodesource.list'? y
root@Nurbek001:~# ls /etc/apt/sources.list.d
canonical_partner.list  newlist.list
root@Nurbek001:~#
```

3. When do you need to get a public key for the usage of the remote repo? Provide an example.

When we are going to add a 3rd party repository to our */etc/apt/sources.list* file or either */etc/apt/sources.list.d* directory, we are needing a public key in order to verify it with our system and help apt to trust it. For example, on the screenshot below we can see that there is an apt-key for *nodesource.list* repo, which was added in Exercise 1 - Question 2:

```
root@Nurbek001:~# apt-key list
/etc/apt/trusted.gpg
--------------------
pub   rsa4096 2014-06-13 [SC]
      9FD3 B784 BC1C 6FC3 1A8A  0A1C 1655 A0AB 6857 6280
uid           [ unknown] NodeSource <gpg@nodesource.com>
sub   rsa4096 2014-06-13 [E]

/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-archive.gpg
-----------------------------------------------------
pub   rsa4096 2012-05-11 [SC]
      790B C727 7767 219C 42C8  6F93 3B4F E6AC C0B2 1F32
uid           [ unknown] Ubuntu Archive Automatic Signing Key (2012) <ftpmaster@ubuntu.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-cdimage.gpg
-----------------------------------------------------
pub   rsa4096 2012-05-11 [SC]
      8439 38DF 228D 22F7 B374  2BC0 D94A A3F0 EFE2 1092
uid           [ unknown] Ubuntu CD Image Automatic Signing Key (2012) <cdimage@ubuntu.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2018-archive.gpg
-----------------------------------------------------
pub   rsa4096 2018-09-17 [SC]
      F6EC B376 2474 EDA9 D21B  7022 8719 20D1 991B C93C
uid           [ unknown] Ubuntu Archive Automatic Signing Key (2018) <ftpmaster@ubuntu.com>
```

Exercise 2 - Managing processes:

Questions to answer:

1. Describe how "top" works. Explain all important fields in its output based on your system resources.

The top command in Linux systems allows you to display a table with list of running processes and estimate how much resources they consume, i.e., how much load they put on the server. This information is going to be helpful in further optimization of the performance of the system.

2. Explain briefly each process state from the output of the "top" command.

- D = uninterrupted sleep
- R = running
- S = sleeping
- T = stopped by job control signal
- t = stopped by the debugger while tracing
- Z = zombie

3. What happens to a child process that dies and has no parent process to wait for it and what's bad about this?

In such a situation if there won't be a parent process, waiting for the child, when it dies, the child becomes a zombie process. It means that this process, becoming a zombie will consume the process ID in the process table. It is bad because process IDs are limited, and other processes can lack it.

4. How to know which process ID is used by an application?

ps aux | grep name_of_the_application

5. What is a zombie process and what could be the cause of it? How to find and kill the zombie process?

Zombie processes in Ubuntu aren't running and can't be killed, even with *sigkill*, they continue to hang in memory until their parent process is going to be terminated. The cause of such process creation is a situation, when the child process will die and there won't be a wait*()* command from its parent. You can find such processes using the *ps:*

ps aux | grep defunct

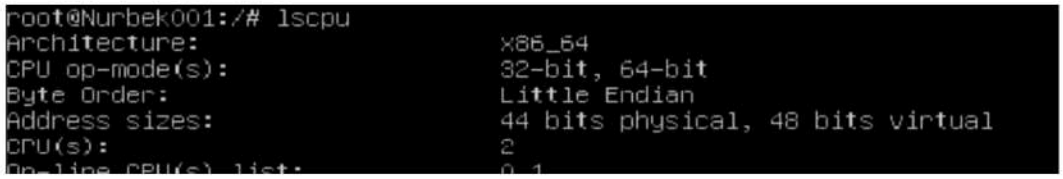
In order to kill such a zombie process we will need to find its "parent" first. After that we can send a termination signal to the parent according to its PID, and it will kill the zombie process as well.

Exercise 3 - Investigating hardware, access control, resources utilization:

Questions to answer:

1. Identify how many CPUs are in your environment? What is a Linux kernel module?

In order to see the amount of CPUs in my environment I used command *lscpu* and determined that I have 2 CPUs.



Linux Kernel module is a combination of individual pieces of code that can be loaded and unloaded from the kernel as needed. They extend the functionality of the kernel without needing to reboot the system.

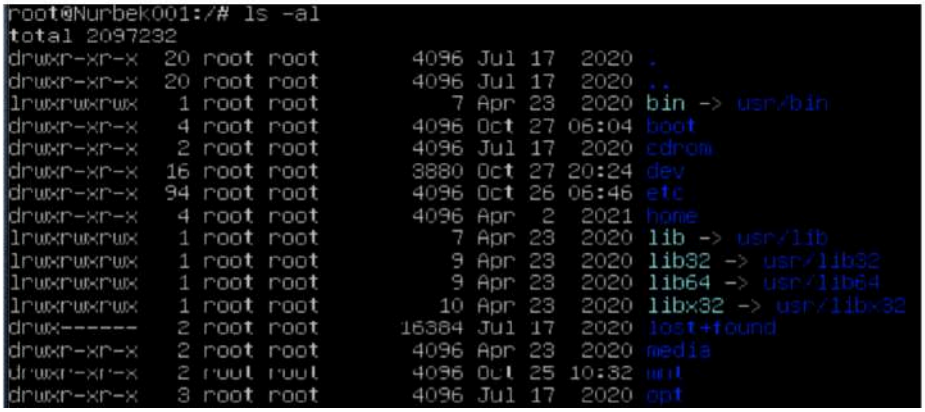
2. Which command will show statistics about your free/used memory? Describe all fields from the output of the command (for example point the difference between free and available)?

In order to see statistics about memory I used command *free*

- total - the total amount of memory available on a system.
- used - the amount of memory that is being used now.
- free - the amount of memory that isn't being used now.
- shared - the amount of memory shared between some processes in a system.
- buff/cache - the amount of memory that is used for caching.
- available - the amount of memory that is available for processes right now.

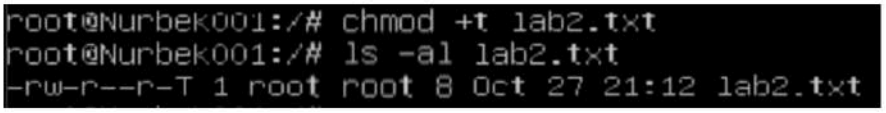
3. If you list the content of a directory using for example "$ls -al" what do numbers in the column following after permissions information tell you?

Second column contains numbers, which represents the amount of hard links for files and the amount of sub-directories for a directory.



4. What is the sticky bit? Show the file or directory with your configured sticky bit.

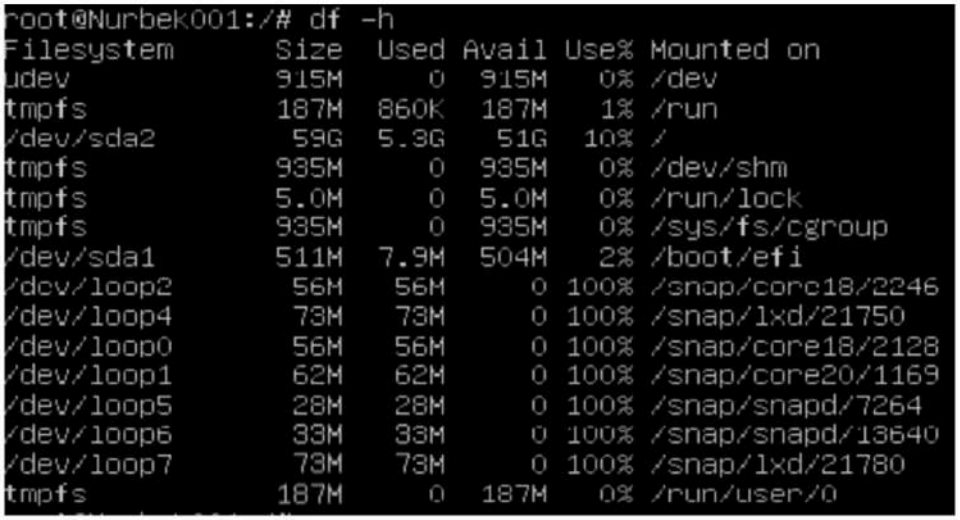
Sticky bit is a special bit for directories, which declares permissions for this directory. If the sticky bit is applied, then the user can delete the file only if he is the owner of the file or directory. Without the sticky bit, if a user can create files in a directory, they can also delete any files from that directory. As an example I created file *lab2.txt* and then configured it with a sticky bit:



On the screen above we can see "T", which is the sticky bit.

5. Which command will show the available disk space on the Unix/Linux system?

In order to see statistics about disk space I used command *df -h*



6. How to add a new system user without home directory and login?

In order to add a new system user without home dir and login I used command, which can be seen on a screenshot below:

```
root@Nurbek001:~# useradd -M homelessuser
```

We can check that user is created, by displaying all the users on a system with *getnet passwd* command, and out new user is displayed there as well:

```
homelessuser:x:1002:1002::/home/homelessuser:/bin/sh
```

7. Explain the differences among the following umask values: 000, 002, 022, 027, 077, and 277.

- 000 - file: rw- rw- rw- , directory: rwx rwx rwx.
- 002 - file: rw- rw- r--, directory: rwx rwx r-x.
- 022 - file: rw- r-- r--, directory: rwx r-x r-x.
- 027 - file: rw- r-- ---, directory: rwx r-x r--.
- 077 - file: rw- --- ---, directory: rwx --- ---.
- 277 - file: r-- --- ---, directory: r-x --- ---.

8. You have already configured swap in the exercise and the next step to increase or resize you swap space x2. Provide steps to do so.

Initial size of the swap was 2GB as we created it in the lab, so I needed to make the size 4GB. Here are the steps:

1. Turn of the swap

```
root@Nurbek001:~# swapoff -a
```

2. Create new swap file

```
root@Nurbek001:~# dd if=/dev/zero of=/swapfile1 bs=1G count=4
```

3. Make a new swap from that file

```
root@Nurbek001:~# mkswap /swapfile1
```

4. Turn on the swap

```
root@Nurbek001:~# sudo swapon /swapfile1
```

5. Check that size is changed to 4GB

```
root@Nurbek001:~# swapon --show
NAME       TYPE SIZE USED PRIO
/swapfile1 file   4G   0B   -2
```

Questions to answer:

1. Create a simple script to print odd numbers from 1 to 10.

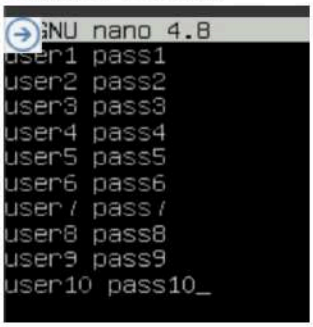
I have used two methods for application of this task, which can be seen on screenshot below(script *left* and output *right)*:

```
#!/bin/bash
echo "Method 1"
seq 1 2 10

echo "Method 2"
for i in {1..10..2}
do
        echo "$1"
done
exit 0
```

```
root@Nurbek001:~# ./task1.sh
Method 1
1
3
5
7
9
Method 2
1
3
5
7
9
```

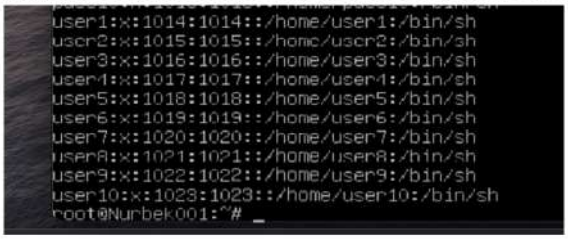
2. Create a shell script which makes new users and their corresponding passwords for 10 accounts in the system. Script should read the data from the file users.txt(you can write data to this file).

Content of *users.txt:*



Then I have created this script:

As we can see above, all 10 users have been added, and in order to check whether it happend or not, we can see the list of users, and as we see from it, my script works:



3. Provide an example of the shell script where you can pass result (not an exit code) of the executed function in a subshell to the parent's shell

shell 1:

shell 2:

output:

As we can see here one shell(*task3.sh*) calls another shell (*task3_1.sh*) during its own execution, and passes the *example* variable to the second shell, making its execution possible.

4. Create script that redirects system date and disk utilization in a file with systemd.

First of all, we are going to move to */usr/bin/* directory:

```
root@Nurbek001:/usr/bin# _
```

Next we create a shell itself:

```
#!/bin/bash

while true
do
        date >> /var/info.txt
        sudo du -sch / >> /var/info.txt
done
```

After that we create service for our shell:

```
[Unit]
Description=My Task4
[Service]
ExecStart=/usr/bin/task4.sh
[Install]
WantedBy=multi-user.target
```

Then we save it and run this service

```
root@Nurbek001:/usr/bin# systemctl start monitor-disk.service
```

So, in order to check whether it is working or not, we can check it by using:

```
root@Nurbek001:/usr/bin# cat /var/info.txt_
```

And we get:

```
15G     /
15G     total
Sun Oct 31 07:39:59 MSK 2021
15G     /
15G     total
Sun Oct 31 07:40:00 MSK 2021
15G     /
15G     total
Sun Oct 31 07:40:02 MSK 2021
15G     /
15G     total
Sun Oct 31 07:40:03 MSK 2021
15G     /
15G     total
Sun Oct 31 07:40:04 MSK 2021
```

Questions to answer:

1. Create backup for any directory with files inside. Create cron job which backups directory at the 5th day of every month.

First of all, I have created the shell script for a backup and used */home* directory, because it has a lot of files with users.

```
#!/bin/bash
backupt_date=$(date +'%Y-%m-%d')
backupt_name="/home/user1/backup-$backupt_date.tar.gz"
tar -cvpzf $backupt_name /home
echo $backupt_date
echo "backup is done"
```

Here we can see that I have created a timestamp variable, so it will be more clear which backup is done on a certain date. In order to test whether this script works,

```
/home/user10/.bash_logout
/home/pass5/
/home/pass5/.profile
/home/pass5/.bashrc
/home/pass5/.bash_logout
/home/pass2/
/home/pass2/.profile
/home/pass2/.bashrc
/home/pass2/.bash_logout
/home/user5/
/home/user5/.profile
/home/user5/.bashrc
/home/user5/.bash_logout
2021-11-03
backup is done
root@Nurbek001:/home/user1#
```

And we also can see that file of backup was created as well:

```
root@Nurbek001:/home/user1# ls
backup-2021-11-03.tar.gz  backup-
```

After that, I have created a cron job for automatization of this task, which can be seen below. It will call my script - *task1.sh* every 5th day of the every month.

```
root@Nurbek001:/home/user1# crontab -l
0 0 5 * * /home/user1/task1.sh
```

2. Install nginx and backup directory with location of index.html. Create cron job which backups directory at midnight every Sunday. Also script should delete old or previous backups.

First of all we found directory, containing index.html

```
root@Nurbek001:/var# ls
backups  cache  crash  info.txt  lib  local  lock  log  mail  opt  run  snap  spool  tmp  www
root@Nurbek001:/var# ls www
html
root@Nurbek001:/var# cd www
root@Nurbek001:/var/www# cd html
root@Nurbek001:/var/www/html# ls
index.nginx-debian.html
root@Nurbek001:/var/www/html# _
```

After that we create a script for backup:

```
#!/bin/bash
tar -cvpzf /home/user1/backup_for_task2.tar.gz /var/www/html/
echo "Backup for index.html is done"
```

I have runned it manually, without cron at first, and got positive result, here is the part of the output:

```
root@Nurbek001:/home/user1# ./task2.sh
tar: Removing leading `/' from member names
/var/www/html/
/var/www/html/index.nginx-debian.html
Backup for index.html is done
root@Nurbek001:/home/user1#
```

And we also can see that file of backup was created as well:

```
1# ls
 backup_for_task2.tar.gz
```

After that, I have created a cron job for automatization of this task, which can be seen below. It will call my script - *task2.sh* at midnight every Sunday(*second line)*.

```
root@Nurbek001:/home/user1# crontab -l
0 0 5 * * /home/user1/task1.sh
0 0 * * 0 /home/user1/task2.sh
```

3. Bonus: create a script which check availability of IP address or network interface(ethernet or wlan) and put this script to cron job which runs scripts every 5 minutes.

```
#!/bin/bash
ping -c2 192.168.1.1
if [ $? != 0 ]
then
        echo "no network" > ip_info.txt
else
        echo "network is available" >ip_info.txt
fi
```

Above you can see the code that was inspired by the code from the link in the lab(*https://forums.raspberrypi.com/viewtopic.php?t=249063*).

Here is the cron job for that:

```
*/5 * * * * /home/user1/task_bonus.sh
```

Questions to answer:

1. You have a range 172.16.200.0/22
   a. Identify subnet range

172.16.200.0 - 172.16.203.255

   b. How many usable IP addresses?

$2^{10} - 2 = 1022$

   c. Identify starting IP and ending IP

Starting IP = 172.16.200.1

Ending IP = 172.16.203.254

2. You have a range 10.16.200.12/17
   a. Identify subnet range

10.16.128.0-10.16.255.255

   b. How many usable IP addresses?

$2^{15} - 2 = 32766$

   c. Identify starting IP and ending IP

Starting IP = 10.16.128.1

Ending IP = 10.16.255.254

3. You have a range 192.168.0.0/24 and divide into small subnets
   a. Subnet with 29 hosts

In order to accommodate 29 hosts, we need a minimum subnet of /27. So, we split out initial range into 8 subnets:

- 192.168.0.0/27
- 192.168.0.32/27
- 192.168.0.64/27
- 192.168.0.96/27
- 192.168.0.128/27
- 192.168.0.160/27
- 192.168.0.192/27
- 192.168.0.224/27

   b. Subnet with 120 hosts

In order to accommodate 120 hosts, we need a minimum subnet of /25. So, we split out initial range into 2 subnets:

- 192.168.0.0/25
- 192.168.0.128/25

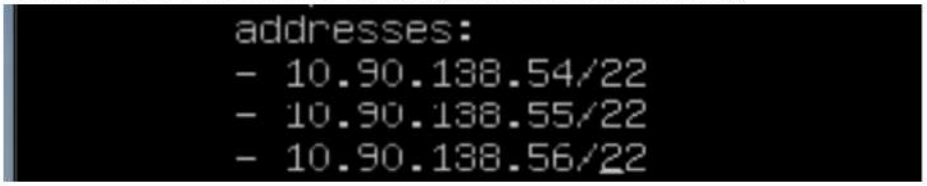
   c. Subnet with 60 hosts

In order to accommodate 29 hosts, we need a minimum subnet of /26. So, we split out initial range into 4 subnets:

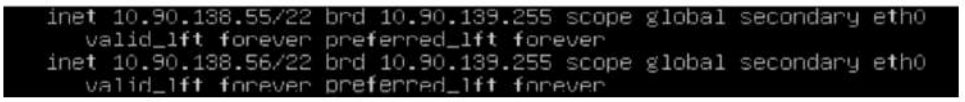
- 192.168.0.0/26
- 192.168.0.64/26
- 192.168.0.128/26
- 192.168.0.192/26

4. Add several IP addresses to interface using by netplan and ping them.

I have added two IP addresses to interface(*last two on screenshot below)*



As we can see, by using *ip addr show dev eth0*, changes have been made:



After that, we pinged them:
10.90.138.55:

```
root@Nurbek001:/etc/netplan# ping 10.90.138.55
PING 10.90.138.55 (10.90.138.55) 56(84) bytes of data.
64 bytes from 10.90.138.55: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 10.90.138.55: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 10.90.138.55: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 10.90.138.55: icmp_seq=4 ttl=64 time=0.026 ms
64 bytes from 10.90.138.55: icmp_seq=5 ttl=64 time=0.036 ms
```

10.90.138.56:

```
root@Nurbek001:/etc/netplan# ping 10.90.138.56
PING 10.90.138.56 (10.90.138.56) 56(84) bytes of data.
64 bytes from 10.90.138.56: icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from 10.90.138.56: icmp_seq=2 ttl=64 time=0.081 ms
64 bytes from 10.90.138.56: icmp_seq=3 ttl=64 time=0.040 ms
```

5. Add sub-interface and IP addresses using netplan

I have added a new subnet of *eth0:1* and added new *ip address* for that, which will not be handled my main interface:

```
  eth0:
      dhcp4: no
      addresses:
      - 10.90.138.54/22
      - 10.90.138.55/22
      - 10.90.138.56/22
      gateway4: 10.90.136.1
      dhcp6: true
  eth0:1:
      addresses: [10.16.200.12/17]
```

However, we see that there is a 100% packet loss on sub-interface IP address:

```
root@Nurbek001:/etc/netplan# ping 10.16.200.12
PING 10.16.200.12 (10.16.200.12) 56(84) bytes of data.
^C
--- 10.16.200.12 ping statistics ---
17 packets transmitted, 0 received, 100% packet loss, time 16384ms
```

It may be caused because only main interface is being used right now