**Networking**

1. [Configure networking and hostname resolution statically or dynamically](https://github.com/StenlyTU/LFCS-official/blob/main/stuff/Networking.md#configure-networking-and-hostname-resolution-statically-or-dynamically)
2. [Configure network services to start automatically at boot](https://github.com/StenlyTU/LFCS-official/blob/main/stuff/Networking.md#configure-network-services-to-start-automatically-at-boot)
3. [Implement packet filtering](https://github.com/StenlyTU/LFCS-official/blob/main/stuff/Networking.md#implement-packet-filtering)
4. [Start, stop, and check the status of network services](https://github.com/StenlyTU/LFCS-official/blob/main/stuff/Networking.md#start-stop-and-check-the-status-of-network-services)
5. [Statically route IP traffic](https://github.com/StenlyTU/LFCS-official/blob/main/stuff/Networking.md#statically-route-ip-traffic)
6. [Synchronize time using other network peers](https://github.com/StenlyTU/LFCS-official/blob/main/stuff/Networking.md#synchronize-time-using-other-network-peers)

**Configure networking and hostname resolution statically or dynamically**

* ip addr show -> Show IP addresses configuration. Short syntax - ip a s
* ip netns -> Show the available network namespaces.
* nmtui

*Network Manager Text User Interface* - Graphical interface to manage network connections configuration

* + Manual means that IP will be configured manually
  + Automatic means that will be used DHCP protocol
  + **NOTE**: IP must be inserted with syntax IP/NETMASK (e.t. 192.168.0.2/24)
* All network configuration will be stored in /etc/sysconfig/network-scripts
* If there is need to change IP configuration of an interface without using nmtui remember to shutdown interface, change IP, restart interface
  + ip link set eth0 down Shutdown interface eth0
  + ip addr add 192.168.0.2/24 dev eth0 Assign IP 192.168.0.2/24 to interface eth0
  + ip link set eth0 up Restart interface eth0

***Hostname & Resolving:***

* There are 3 different hostnames: static, transient and pretty hostname.
* The hostname can be changed editing /etc/hostname.
  + hostname -> show current hostname.
  + Alternative: hostnamectl set-hostname your-new-hostname -> set hostname equal to your-new-hostname.
  + Reboot is required to see new hostname applied.
* In /etc/hosts is configured a name resolution that take precedence of DNS - configurable in /etc/nsswitch.conf
  + It contains static DNS entry.
  + It is possible add hostname to row for 127.0.0.1 resolution, or insert a static IP configured on principal interface equal to hostname.
  + Alias can be added following this format: **IP FQDN ALIAS**.
* In /etc/resolv.conf there are configured DNS servers entry.
* It is possible to insert more than one *nameserver* as backup (primary and secondary).
* Use: dig www.pluralsight.com -> to resolve the site and see the IP address.

**Configure network services to start automatically at boot**

Network Manager

* Its purpose is to automatically detect, configure, and connect to a network whether wired or wireless such as VPN, DNS, static routes, addresses, etc which is why you'll see #Configured by NetworkManager in /etc/resolv.conf, for example. Although it will prefer wired connections, it will pick the best known wireless connection and whichever it deems to be the most reliable. It will also switch over to wired automatically if it's there. It's not necessary and many (including me) disable it as most would rather manage their own network settings and don't need it done for them.
* systemctl stop NetworkManager.service
* systemctl disable NetworkManager.service

Network

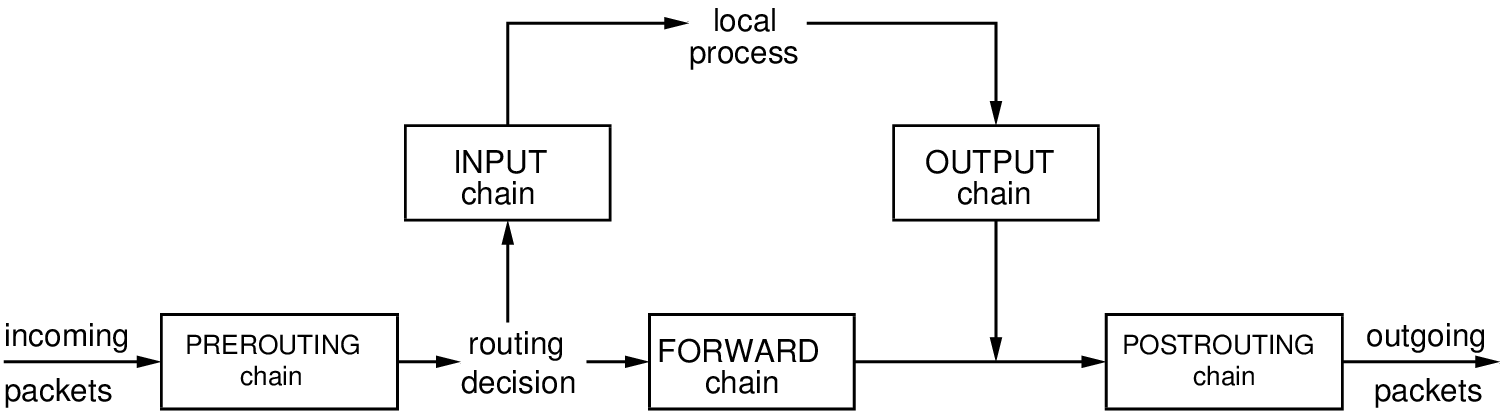
* systemctl status network -> to check network configuration status.
* systemctl restart network -> to reload network configuration.

References:

* <https://unix.stackexchange.com/questions/449186/what-is-the-usage-of-networkmanager-in-centos-rhel7>

**Implement packet filtering**

* The firewall is managed by Kernel
* The kernel firewall functionality is Netfilter
* Netfilter will process information that will enter and will exit from system
  + For this it has two tables of rules called chains:
    - *INPUT* that contains rules applied to packets that enter in the system
    - *OUTPUT* that contains rules applied to packets that leave the system
* Another chain can be used if system is configured as router: *FORWARD*
* Finally there are other two chains: PREROUTING, POSTROUTING

[](https://camo.githubusercontent.com/73f8337a976b3892f6883d52a2abbf5292bc8f4b11f88714b47e19b4925d2843/68747470733a2f2f73616e64696c616e64732e696e666f2f6e736c2f69707461626c65732d636861696e732d322d723636362e706e67)

* The above picture shows the order in which the various chains are valued. The arrows indicate the route of the packages:
  + Incoming packets are generated from the outside
  + Outgoing packets are either generated by an application or are packets in transit
* The rules inside chains are evaluated in an orderly way.
  + When a rule match the other rules are skipped
  + If no rules match, default policy will be applied
    - Default policy:
      * ACCEPT: the packet will be accepted and it will continue its path through the chains
      * DROP: the packet will be rejected
* The utility to manage firewall is iptables
* iptables will create rules for chains that will be processed in an orderly way
* firewalld is a service that use iptables to manage firewalls rules
  + Starting firewalld service will automatically stop iptables service.
* firewall-cmd is the command to manage firewalld
* This is table which chain is impemented in which tables:

| **Tables↓/Chains→** | **PREROUTING** | **INPUT** | **FORWARD** | **OUTPUT** | **POSTROUTING** |
| --- | --- | --- | --- | --- | --- |
| routing |  |  |  | ✓ |  |
| raw | ✓ |  |  | ✓ |  |
| mangle | ✓ | ✓ | ✓ | ✓ | ✓ |
| nat (DNAT) | ✓ |  |  | ✓ |  |
| filter |  | ✓ | ✓ | ✓ |  |
| security |  | ✓ | ✓ | ✓ |  |
| nat (SNAT) |  | ✓ |  |  | ✓ |

***Firewalld:***

* firewalld is enabled by default in CentOS and it's idea is to simplify the work with iptables.
* It works with zone, *public* is default zone
* The *zone* is applied to an interface
  + The idea is that we can have safe zone, e.g. bound to an internal interface, and unsafe zone, e.g. bound to external interfaces internet facing
  + firewall-cmd --get-default-zone || --get-active-zones || --get-zones || --set-default-zone
* firewall-cmd --list-all show current configuration
  + services -> service that are allowed to use interface
  + ports -> ports that are allowed to use interface

Services:

* firewall-cmd --get-services shows the list of default services
  + The services are configured in /urs/lib/firewalld/services
  + /urs/lib/firewalld/services contains xml file with service configuration
  + /etc/firewalld/services where new services are created automatically.
* firewall-cmd --add-service <service-name> add service to current configuration
  + **NOTE**: it isn't a permanent configuration. To make it use: --permanent.
* firewall-cmd --reload reload firewalld configuration
  + **NOTE**: If a service was added with previous command now it disappeared
* firewall-cmd --add-service <service-name> --permanent add service to configuration as permanent
  + **NOTE**: Now if firewalld configuration is reloaded service it is still present

Ports:

* firewall-cmd --add-port 4000-4005/tcp -> Open TCP ports from 4000 to 4005
* firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1 -p tcp -m tcp --dport 80 -j ACCEPT
  + Add a firewall rule using iptables syntax
  + This add permanently a rule as first in OUTPUT chain to allow connections to TCP destination port 80

***iptables:***

* The firewalld daemon can be substitute with iptables daemon (the configuration that was in place until recently)
  + systemctl stop firewalld
  + iptables -L
    - More verbose output iptables -L -v
    - Show configuration of iptables chains
    - Note that policies is set equal to ACCEPT for every chain. This means that no package will be rejected. This is equal to have a shut downed firewall
  + systemctl disable firewalld
  + yum -y install iptables-services
  + systemctl enable iptables
* With this configuration rules must be inserted
* iptables -P INPUT DROP -> Set default policy to DROP for INPUT chain

***iptables rules syntax:***

* iptables {-A|I} chain [-i/o interface][-s/d ipaddres] [-p tcp|upd|icmp [--dport|--sport nn…]] -j [LOG|ACCEPT|DROP|REJECTED]
* -L || -nvL - Will output all current rules sorted by chain
* -S - List all active iptables rules. Just like real commands
* {-A|I} chain
  + -A append as last rule
  + -I insert. This require a number after chain that indicate rule position
* [-i/o interface]
  + E.g. -i eth0 - the package is received (input) on the interface eth0
* [-s/d ipaddres]
  + -s Source address. ipaddres can be an address or a subnet
  + -d Destination address. ipaddres can be an address or a subnet
* [-p tcp|upd|icmp [--dport|--sport nn…]]
  + -p protocol
  + --dport Destination port
  + --sport Source port
* -j [LOG|ACCEPT|DROP|REJECTED]
  + ACCEPT accept packet
  + DROP silently rejected
  + REJECTED reject the packet with an ICMP error packet
  + LOG log packet. Evaluation of rules isn't blocked.

***Examples:***

* iptables -A INPUT -i lo -j ACCEPT - Accept all inbound loopback traffic
* iptables -A OUTPUT -o lo -j ACCEPT - Accept all outbound loopback traffic
* iptable -A INPUT -p tcp --dport 22 -j ACCEPT - Accept all inbound traffic for tcp port 22
* iptable -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  + This is a rule that is used to ACCEPT all traffic generated as a response of an inbound connection that was accepted. E.g. if incoming traffic for web server on port 80 was accepted, this rule permits to response traffic to exit from system without inserting specific rules in OUTPUT chain
* iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT -> Allow all incoming SSH traffic
* iptables -A INPUT -p tcp -s 15.15.15.0/24 --dport 873 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT -> Allow incoming Rsync traffic from Specific IP Address or Subnet
* iptables -I INPUT 2 -p tcp --dport 3306 -s 192.168.5.0/24 -j ACCEPT -> Accept mysql traffic from network 192.168.5.0/24 and insert the rule as second.
* iptables -D INPUT 2 -> Delete the second rule in INPUT chain.
* **NOTE:** Remember to use only one service to configure the Linux firewall. Do not mix ufw, firewalld, and iptables config files and frontends.
* **NOTE:** The file /etc/services contains a list of well know ports with services name

References:

* <https://debian-handbook.info/browse/da-DK/stable/sect.firewall-packet-filtering.html>

**Start, stop, and check the status of network services**

* Network services are controlled as other daemon with systemctl command.
  + systemctl status servicename
* With netstat it is possible to list internet port opened by a process.
  + yum -y install net-tools
  + netstat -tln - Show TCP port opened by processes.

**Statically route IP traffic**

* ip route show
  + Print route
  + Alternative command route -n
* ip route add 192.0.2.1 via 10.0.0.1 [dev interface]
  + Add route to 192.0.2.1 through 10.0.0.1. Optionally interface can be specified
* To make route persistent, create a *route-ifname* file for the interface through which the subnet is accessed, e.g eth0:
  + vi /etc/sysconfig/network-scripts/route-eth0
  + Add line 192.0.2.1 via 10.0.0.101 dev eth0
  + service network restart to reload file
* ip route add 192.0.2.0/24 via 10.0.0.1 [dev ifname]
  + Add a route to subnet 192.0.2.0/24
* nmcli -> to work with NetworkManager service.
* To configure system as a router ip\_forward must be enabled:
  + echo 1 > /proc/sys/net/ipv4/ip\_forward
  + To make configuration persistent
    - echo net.ipv4.ip\_forward = 1 > /etc/sysctl.d/ipv4.conf
    - OR: echo net.ipv4.ip\_forward = 1 > /etc/sysctl.d/99-sysctl.conf

References:

* <https://my.esecuredata.com/index.php?/knowledgebase/article/2/add-a-static-route-on-centos>

**Synchronize time using other network peers**

* In time synchronization the concept of Stratum define the accuracy of server time.
* A server with Stratum 0 is the most reliable.
* A server synchronized with a Stratum 0 become Stratum 1.
* Stratum 10 is reserved for local clock. This means that it is not utilizable.
* The upper limit for Stratum is 15.
* Stratum 16 is used to indicate that a device is unsynchronized.
* Remember that time synchronization between servers is a slowly process.
* hwclock -> Show hardware clock - BIOS. Can be synched with system time.
  + hwclock --systohc -> sync system time to the hardware time.
  + hwclock --hctosys -> sync hardware time to the system time.
* timedatectl set-time "2016-10-30 13:30:22" - Part of Systemd ecosystem. Set the time.
* timedatectl set-timezone Europe/Sofia - Change the timezone.

***CHRONYD:***

* Default mechanism to synchronize time in CentOS 7.
* Chrony is systemd deamon. Start it with: systemctl start chronyd
* Configuration file /etc/chrony.conf
  + server parameters are servers that are used as source of synchronization.
* Use the command chronyc to interact with the chronyd.
  + chronyc sources -> contact server and show them status.
  + chronyc tracking -> show current status of system clock.
  + chronyc ntpdata -> show more info for every server.
* **NOTE**: if some of the commands above doesn't work please refer to this bug <https://bugzilla.redhat.com/show_bug.cgi?id=1574418>
  + Simple solution: setenforce 0
  + Package selinux-policy-3.13.1-229 should resolve the problem.

**NTP:**

* The old method of synchronization. To enable it Chronyd must be disabled. It's also service.
* Configuration file /etc/ntp.conf
  + server - Parameters are servers that are used as source of synchronization.
  + logfile /var/log/ntp.log - Specify log file.
* ntpq -p check current status of synchronization.
* ntpdate -q 2.bg.pool.ntp.org -> Set the local date and time from specific server.
* ntpstat -> Show NTP synchonization status.

Both NTP and CHRONYD are using port ***123***. CHRONYD is working also on port **323**.