Key Commands

General

Command	Description
`man -k iscsi`	Searches short descriptions
`man -K iscsi`	Full search
`find /usr/share/doc -type f -name *.html`	Find all HTML files in the dir
`grep volume README`	Searches for the word “volume” in the README file
`grep -i volume README`	Case-insensitive search for the word “volume” in the README file
`diff README README.2`	Diff two files
`wdiff README README.2`	Diff two files

Process management

Command	Description
`kill -l`	Lists all kill signals
`ps -F -p $(pgrep sshd)`	Process details for sshd
`pmap 1404`	Memory map of the requested process
`pgrep fail2ban`	Get process ID for specified term
`pkill sleep`	Kill all `sleep` processes
`nice`	Current process’s priority
`ps -l`	Lists processes and their priority
`nice -n 19 sleep 1000`	Starts a low-priority process
`renice -n 0 1393`	Changes the priority of an existing process
`renice -n 10 $(pgrep sleep)`	Changes the priority of an existing process

Notes:

- Nice value range: -20 (high) to +19 (low)
- Regular users can only use nice values >=0
- Default priority set in `/etc/security/limits.conf`
    - Set in `/etc/security/limits.d/`

Resource management

Command	Description
`top`	Activity monitor
`free`	Memory resources
`pwdx $(pgrep squid)`	Working directory for the process (from `/proc/<id>/cwd`)
`uptime`	System uptime
`cat /proc/loadavg`	Load average
`lscpu`	Lists the CPUs in the system
`vmstat`	Virtual memory stats
`iostat -m 5 3`	CPU and IO stats
`pidstat -p 1614 5 3`	Stats for a specific process
`mpstat 5 3`	Per-CPU stats

`sysstat`

Package: `sysstat` Configuration: `/etc/sysconfig/sysstat`

Command	Description
`sar`	`sysstat` reporting tool
`sar -A`	All stats
`sar -n ALL`	All networking stats
`sar -s 14:50:00 -e 15:10:00`	Limit to a time period

Notes:

- `/etc/cron.d/sysstat` is created to compile reports

Scheduling tasks

- `cron`: Schedules jobs
    - Configuration: `/etc/crontab`
- `anacron`: Handles systems not running 24x7
    - Configuration: `/etc/anacrontab`
- `at`: Used for one-off jobs and batches
    - `atq` - lists queued jobs
    - `atrm` - removes a job

Shared libraries

Command	Description
`ldd /usr/bin/ls`	Lists the shared libraries used by the `ls` command
`ldconfig -p`	Display the linker cache

Add new library locations to a file in `/etc/ld.so.conf.d` or add to `$LD_LIBRARY_PATH`

Logging

Configuration:

- `/etc/rsyslog.conf`
- `/etc/rsyslog.d`

Configure a logging server

Enable SELinux and the firewall for syslog to listen on port 10514:

```
semanage port -a -t syslogd_port_t -p tcp 10514
semanage port -l|grep syslogd
firewall-cmd --zone=lab-internal --add-port=10514/tcp --permanent
firewall-cmd --reload
```

Configure the port in `/etc/rsyslog.conf` by including the following line:

```
$template TmplAuthpriv, "/var/log/remote/auth/%HOSTNAME%
/%PROGRAMNAME:::secpath-replace%.log"
$template TmplMsg, "/var/log/remote/msg/%HOSTNAME%/%PROGRAMNAME:::secpath-
replace%.log"

$ModLoad imtcp

# Adding this ruleset to process remote messages
$RuleSet remote1
authpriv.*   ?TmplAuthpriv
*.info;mail.none;authpriv.none;cron.none   ?TmplMsg
$RuleSet RSYSLOG_DefaultRuleset   #End the rule set by switching back to
the default rule set
$InputTCPServerBindRuleset remote1  #Define a new input and bind it to the
"remote1" rule set

$InputTCPServerRun 10514
```

Restart `rsyslog`

Configure the client

Configure the logserver in `/etc/rsyslog.conf` by including the following line:

```
*.* @@172.16.1.1:10514
```

Users and Groups

Primary files

- `/etc/passwd` - local users
- `/etc/shadow` - local user passwords
- `/etc/group` - local groups
- `/etc/gshadow` - local group passwords
- `/etc/nsswitch.conf` - Name Service Switch config file
- `/etc/profile.d/` - profile scripts
- `/etc/skel` - Home directory template
- `/etc/login.defs` - Login defaults such as password config
- `/etc/default/useradd` - Defaults for new users

Commands

Command	Description
`getent`	Gets entries from the administrative database
`getent passwd penguin`	`passwd` entry for the requested user
`id`	Info about current user
`id penguin`	Info about specified user
`useradd`	Create a user
`usermod`	Modify a user
`userdel`	Delete a user
`groupadd`	Create a group
`chsh`	Change user shell
`chmod g+s <file/dir>`	Set the Group ID (SGID)

Shell and login scripts

- `profile` scripts are used at login and should include environment settings.
    - `/etc/profile`
    - `/etc/profile.d`
    - `~/.bash_profile`

- `bashrc` scripts are executed for interactive non-login shells. The `profile` scripts will generally call `bashrc` scripts.
    - `/etc/bashrc`
    - `~/.bashrc`
- `logout` scripts are executed at logout
    - `~/.bash_logout`

PAM

Configuration:

- `/etc/pam.d/`
- `/etc/security/`
    - Password quality: `/etc/security/pwquality.conf`
    - Limit resource access (`ulimit`): `/etc/security/limits.conf`

Various PAM modules are available in `/lib64/security`

Can use `authconfig`:

```
authconfig --savebackup=/backups/authconfigbackup20170701


authconfig --passminlen=9 --passminclass=3 --passmaxrepeat=2
--passmaxclassrepeat=2 --enablerequpper --enablereqother --update
```

Package management

- Yum repositories: `/etc/yum.repos.d`
- Cache: `/var/cache/yum/`

Command	Description
`rpm -V nmap`	Verify a package
`rpm -qf /etc/hosts`	Query the package that installed a resource
`yum repolist`	List Yum repos
`yum list installed`	List all installed packages
`yum whatprovides lsof`	Lists the package providing a file/app (e.g. `lsof`)
`yum update kernel`	Update the kernel

Source packages

Download and compile from source:

```
yum install yum-utils ncurses-devel bzip2 gcc
yumdownloader --source zsh
cd rpmbuild/SOURCES/
tar -xjf zsh-5.0.2.tar.bz2
cd zsh-5.0.2/
./configure
make
make install
```

Date and time

Command	Description
`timedatectl`	Get current date/time info
`timedatectl list-timezones`	List all timezones
`timedatectl set-timezone time_zone`	Set the timezone

Chrony

- Package: `chrony`

- Service: `chronyd`
- Configuration: `/etc/chrony.conf`
    - Sample time server entry: `server 0.centos.pool.ntp.org iburst`

Command	Description
`chronyc`	CLI for Chrony config
`chronyc tracking`	Current time tracking stats
`chronyc sources`	Details of time sources

Systemd management

Services

Command	Description
`systemctl start httpd`	Start the service
`systemctl stop httpd`	Stop the service
`systemctl enable httpd`	Enable the service to start at boot

Run levels/targets

Now called `targets` in Systemd

Command	Description
`runlevel`	Get the current run level
`systemctl get-default`	Gets the default target
`systemctl set-default multi-user.target`	Sets the default target
`systemctl isolate rescue.target`	Changes the current run level

To change at boot time:

1. In the Grub menu, press `e` to edit
2. At the end of the `Linux16` line, append `systemd.unit=rescue.target`
3. Login as `root`
4. If you need to edit anything: `mount -o remount,rw /`

Networking

Configuration:

- `/etc/sysconfig/network-scripts/`
- `/etc/hosts` - local static lookups
- `/etc/hostname` - the hostname
- `/etc/nsswitch.conf` - name service switching
- `/etc/resolv.conf` - resolver config

Command	Description
`hostnamectl`	Current hostname
`hostnamectl set-hostname router.lab.example.com`	Set hostname
`ip a s`	IP details for all network devices
`ip a s enp0s3`	IP details for the specified network device
`ip link show enp0s3`	Device info
`ethtool enp0s3`	Network driver info
`ls /sys/class/net/`	Lists all networking devices
`netstat -t`	All active connections
`netstat -tulpn`	Lists all ports and backing processes
`watch -n 5 -x netstat --interfaces`	Handy network activity monitor
`nmap router.lab.example.com`	Port scanner
`iptables --list`	Lists all rules for all chains

Networking config

Command	Description
`ip addr add 172.17.67.3/16 dev enp0s8`	Temporarily add the IP address

IP Forwarding

To enable IP Forwarding edit `/etc/sysctl.conf` to feature:

```
net.ipv4.ip_forward = 1
```

Load the changes with:

```
sysctl -p
```

Firewall

Command	Description
`firewall-cmd --state`	Current firewall state
`firewall-cmd --reload`	Sets config and reloads
`firewall-cmd --get-services`	Lists all pre-canned services known to the firewall
`firewall-cmd --get-zones`	Lists all defined zones
`firewall-cmd --list-all --zone=lab-internal`	Get details on a zone
`firewall-cmd --add-service=ssh --permanent --zone=lab-internal`	Adds the `SSH` service to the firewall

Advanced language help: `man 5 firewalld.richlanguage`

IP Masquerading

```
firewall-cmd --zone=external --add-masquerade --permanent --zone=lab-dmz
firewall-cmd --reload

#Check:
firewall-cmd --permanent --query-masquerade --zone=lab-dmz
```

Note: This will also configure IP Forwarding

Port forwarding

```
firewall-cmd --permanent --zone=lab-dmz --add-forward-
port=port=80:proto=tcp:toaddr=172.16.100.50:toport=8080
```

Tunnels

Listen locally on 2222 and tunnels to `172.16.1.50`, then calling into port 22:

```
ssh -f -L 2222:localhost:22 ansible@172.16.1.50 -N
```

DHCP

Client

- Package: `dhclient`

Configure the appropriate `/etc/sysconfig/network-scripts/ifcfg-` file with:

```
BOOTPROTO="dhcp"
```

Server

- Package: `dhcp`
- Configuration: `/etc/dhcp/dhcpd.conf`
- Leases: `/var/lib/dhcpd/dhcpd.leases`

Example config:

```
option domain-name "lab.example.com";
option domain-name-servers ns.lab.example.org;
shared-network lab {
  option subnet-mask 255.255.255.0;
  option domain-search "lab.example.com";
  option domain-name-servers 172.16.1.1;
  option time-servers 172.16.1.1;
  next-server 172.16.1.1;
  filename "pxelinux.0";

# The Internal subnet
  subnet 172.16.1.0 netmask 255.255.255.0 {
    option routers 172.16.1.1;
    range 172.16.1.100 172.16.1.199;
    #option auto-proxy-config " http://proxy.lab.example.com/proxy
/proxy.pac";

    host canary {
     option host-name "canaryinternal.lab.example.com";
     hardware ethernet 08:00:27:c7:13:9e;
     fixed-address 172.16.1.50;
    }
  }
}
```

Check the configuration: `dhcpd -t -cf /etc/dhcp/dhcpd.conf`

DNS

Client

- Configuration:
    - `/etc/resolv.conf`

Configure the appropriate `/etc/sysconfig/network-scripts/ifcfg-` file with:

```
PEERDNS=no
DNS1=
DNS2=
```

Or via DHCP.

Server

- Packages: `bind bind-utils`
- Configuration:
    - `/etc/named.conf` - primary configuration
    - `/var/named` - configuration items
- Log: `/var/named/data/named.run`
- Samples: `/usr/share/doc/bind-9.9.4/sample/`

Options

Command	Description
`listen-on port 53 { 172.16.1.1; 127.0.0.1; };`	Sets port and host address
`allow-query { 172.16.0.0/16; 127.0.0.1; };`	The hosts the server will respond to
`recursion yes;`	Provides a caching server
`forwarders {8.8.8.8; 8.8.4.4};` `forward only;`	Configure forwarding

Create a forward lookup zone

Add to `/etc/named.conf`:

```
zone "lab.example.com." {
    type master;
    file "named.lab";
    allow-update { none; };
};
```

Then in `/var/named/named.lab`:

```
$TTL 3H
$ORIGIN lab.example.com.

lab.example.com. IN SOA router.lab.example.com. root.lab.example.com. (
    1 ; serial - increment this on changes
    1D ; refresh
    1H ; retry
    1W ; expire
    3H) ; minimum

lab.example.com. NS router.lab.example.com.
router A  172.16.1.1
time  CNAME  router
centos-mirror CNAME router
mirror CNAME router
proxy CNAME router
mail CNAME router
lab.example.com. MX 10 mail.lab.example.com
canaryinternal A 172.16.1.50
canarydmz A 172.16.100.50
```

Validate:

- `named-checkzone lab.example.com named.lab`
- `named-checkconf`

LDAP

Client

- Packages: `openldap-clients nss-pam-ldapd`
- Configuration: `etc/nsswitch.conf`

Configuration:

```
# Enable a user's home dir to be created ad-hoc:
authconfig --enablemkhomedir --update

# Configure LDAP for use User Information and/or Authentication
authconfig-tui
```

Command	Description
`ldapsearch -x -b 'dc=lab,dc=example,dc=com' '(uid=penguin)'`	Search for a user

Kerberos

Make sure time services are correctly configured.

Server

Configure a Kerberos server via `kadmin` and `kadmin.local`

Command	Description
`listprincs`	List principals
`addprinc root/admin`	Add the `root` principal with `admin` rights
`addprinc penguin`	Add a normal user
`addprinc -randkey host/server2.lab.example.com` `ktadd host/server2.lab.example.com`	Add a server prinipal

Client

- Packages: `krb5-workstation pam_krb5`
- Configuration:
    - `/etc/krb5.conf`
    - Use `authconfig-tui` to setup Authentication

In order to get keys and access systems:

Command	Description
`kinit`	Get a ticket
`klist`	List current tickets
`kdestroy`	Removes the ticket

Enable in SSH

Set the following to `yes` and `systemctl reload sshd`:

- `GSSAPIAuthentication`
- `GSSAPIDelegateCredentials`

Then:

```
authconfig --enablekrb5 --update
```

Filesystem management

ACLs

Command	Description
`getfacl test.txt`	List ACLs
`setfacl -m u:puffin:r test.txt`	Grant read to the `puffin` user
`setfacl -m g:birds:rw test.txt`	Grant read/write to the `birds` group
`setfacl -x g:birds test.txt`	Removes access from the `birds` group
`setfacl -b test.txt`	Removes all ACLs

SELinux

Configuration:

- `/etc/selinux/config`
- `/etc/selinux/targeted/contexts/`

Packages:

- `setools`
- `setools-console`
- `policycoreutils-python`
- `setroubleshoot`
- `selinux-policy-doc`

Command	Description
`ls -Z`	List files with their SELinux context
`ls -dZ /var/spool/squid`	Directory context
`ps axZ`	List processes with their SELinux context
`id -Z`	User context
`sestatus`	SELinux status
`getenforce`	SELinux mode
`setenforce 0`	Set to permissive
`seinfo`	Policy query
`ausearch -m avc`	SELinux alerts in the audit log

Manage policies and contexts

Command	Description
`chcon`	Changes a file's security context - only lasts until next restore
`restorecon`	Restores a file's context to the default
`getsebool -a`	Lists all boolean config
`setsebool -P samba_export_all_rw 1`	Enables Samba to read/write all files
`semanage port -l`	Port mappings
`semanage permissive -a smbd_t`	Sets the `smbd_t` process type to be permissive

Example of configuring SELinux for Squid

Audit log revealed an issue in permissive mode:

```
time->Sat Mar 24 14:18:27 2018
type=PROCTITLE msg=audit(1521865107.759:118):
proctitle=2873717569642D3129002D66002F6574632F73717569642F73717569642E636F6E6
type=SYSCALL msg=audit(1521865107.759:118): arch=c000003e syscall=2
success=yes exit=18 a0=55c2f9f7ce40 a1=641 a2=1a4 a3=64697571732f6c6f
items=0 ppid=1603 pid=1605 auid=4294967295 uid=23 gid=23 euid=23 suid=0
fsuid=23 egid=23 sgid=23 fsgid=23 tty=(none) ses=4294967295 comm="squid"
exe="/usr/sbin/squid" subj=system_u:system_r:squid_t:s0 key=(null)
type=AVC msg=audit(1521865107.759:118): avc:  denied  { append open } for
pid=1605 comm="squid" path="/var/spool/squid/00/00/000000E5" dev="dm-1"
ino=8766949 scontext=system_u:system_r:squid_t:s0
tcontext=system_u:object_r:unlabeled_t:s0 tclass=file
type=AVC msg=audit(1521865107.759:118): avc:  denied  { create } for
pid=1605 comm="squid" name="000000E5" scontext=system_u:system_r:squid_t:s0
tcontext=system_u:object_r:unlabeled_t:s0 tclass=file
```

Note that the service context (`scontext=system_u:system_r:squid_t:s0`) was trying to access the type context (`tcontext=system_u:object_r:unlabeled_t:s0 tclass=file`).

Some checks:

- Check the squid process with `axZ|grep squid`
- The `getsebool squid_connect_any` command indicates squid can connect to any port.
- Check the cache dir with `ls -dZ /var/spool/squid`
- Look at the file contexts for squid: `semanage fcontext --list|grep squid`
- `man squid_selinux` indicated that `squid_cache_t` is used for cache files

Configure the context:

```
semanage fcontext -a -t squid_cache_t "/var/spool/squid(/.*)?"
restorecon -R -v /var/spool/squid
systemctl restart squid
```

The new entry is in `cat /etc/selinux/targeted/contexts/files/file_contexts.local`

I could have used `semanage permissive -a squid_t` but I wanted to be specific.

Storage management

Handy commands:

- `lsof`: lists open files

Mounting a filesystem

- `/mnt` is used for temporarily mounting
- `/mnt/cdrom` is usually a symbolic link

Command	Description
`mount`	List all currently mounted filesystems
`mount /dev/cdrom /mnt/`	Mount the CD/DVD to `/mnt`
`umount /mnt`	Unmounts the filesystem
`eject /dev/cdrom`	Ejects (and unmounts) the CD/DVD

fstab

XFS example:

```
UUID=be9df528-6544-4592-99a8-b3a6d223ed3e /var/ftp/pub/distro     xfs
defaults        0 0
```

Storage devices

Command	Description
`lsblk`	Lists block devices
`parted /dev/sdb print`	Get the partition details for a device
`blkid -o list`	Prints block device info, inc fs type and UUID
`shred -v /dev/sdb`	Deletes the device

Partitioning

Command	Description
`parted /dev/sdb`	Starts the `parted` cli for the nominated device
cli: `print`	Info about the device
cli: `mklabel gtp`	Sets the disk to use the GUID partition table
cli: `mkpart primary xfs 0% 25%`	Creates a primary partition with the `xfs` filesystem
cli: `rm 1`	Removes partition 1
cli: `quit`	Exits the `parted` cli

XFS

Command	Description
`xfs_info /dev/sdb1`	

System Storage manager

Install:

```
yum install system-storage-manager lvm2
```

Command	Description
`ssm list`	Lists info about detected devices, pools, volumes
`ssm create --fs xfs -s 5G -n centos-distro /dev/sdc1 /dev/sdc2`	Creates a new 5Gig logical volume in the default pool (`lvm_pool`). The pool has 2 physical volumes (`/dev/sdc1`, `/dev/sdc2`)
`ssm create --fs xfs -s 10G -n squid-cache`	Creates a new logical volume in the default pool
`ssm mount /dev/lvm-pool/centos-distro /mnt`	Temporary mount of a logical volume

Example `/etc/fstab` entry:

```
/dev/lvm_pool/centos-distro /var/ftp/pub/distro   xfs     defaults    0 0
```

RAID

Levels:

- linear - spans storage over different sized disks
- Raid 0 same as linear but same size disks
- Raid 1 mirror over 2 disks
- Raid 4-6 data is striped with parity over 3 or more disks

Packages: `mdadm`

Command
`mdadm --create /dev/md0 --verbose --level=mirror --raid-devices=2 /dev/sdb13 /` `mkfs.xfs /dev/md0`
`mdadm --create /dev/md1 --level 5 --raid-devices=3 /dev/sdb10 /dev/sdb9 /dev/s` `mkfs.xfs /dev/md1`
`mdadm --detail --scan >> /etc/mdadm.conf`
`mdadm --stop /dev/md0`
`mdadm --assemble --scan`

To add/replace devices:

Command	Description
`mdadm --manage /dev/md1 --add-spare /dev/sdb7` `mdadm --detail /dev/md1`	Adds a spare device
`mdadm --manage /dev/md1 --replace /dev/sdb9`	Drops a disk out
`mdadm --detail /dev/md1`	Checks device status in array
`mdadm --manage /dev/md1 --re-add /dev/sdb9`	Adds device back as spare

Sample `fstab` entry:

```
/dev/md1 /data xfs defaults 0 0
```

iSCSI block storage server

- iSCSI *targets* are servers that share out block devices
- iSCSI *initiators* are clients

Server

Packages: `targetd targetcli`

Run `targetcli` and configure a target:

```
backstores/block create mediashare /dev/lvm_pool/mediashare
iscsi/ create iqn.2018-03.com.example.lab.router:media
cd iscsi/iqn.2018-03.com.example.lab.router:media/tpg1/
luns/ create /backstores/block/mediashare
acls/ create iqn.2018-03.com.example.lab.canaryinternal:media
cd /
ls
saveconfig
exit
```

Client

Packages: `iscsi-initiator-utils`

Configure the initiator name: `/etc/iscsi/initiatorname.iscsi`

```
InitiatorName=iqn.2018-03.com.example.lab.canaryinternal:media
```

Then access the portal:

```
iscsiadm -m discovery -t st -p 172.16.1.1
```

and connect:

```
iscsiadm -m node -T iqn.2018-03.com.example.lab.router:media -l

iscsiadm --mode node --targetname iqn.2018-03.com.example.lab.router:media
-l

# Create a filesystem
mkfs.xfs /dev/sdc
```

To disconnect:

```
iscsiadm -m node -T iqn.2018-03.com.example.lab.router:media -u
```

NFS

Packages: `nfs-utils` Configuration: `/etc/exports/` Services: - `nfs-server` - `rpcbind`

Sample `/etc/exports/` entry:

```
/share/nfs *(ro)
```

Export shares using:

```
exportfs -r
```

Mount the share:

```
mount -t nfs server2.lab.example.com:/share/nfs /mnt
```

AutoFS

Package: `autofs`

Configuration: `/etc/auto.misc`

Add an automount to `/etc/auto.misc`:

```
nfsshare        -ro,soft,intr           server1.lab.example.com:/share/nfs
```

This will appear in `/misc/nfsshare`

CIFS

Package: `samba`

Prepare a share directory:

```
mkdir -m 1777 /share
```

Setup the share in `/etc/samba/smb.conf`:

```
[share]
path=/share
writable=yes
```

Prep the password with `smbpasswd -a root`

SELinux config:

```
semanage fcontext -a -t samba_share_t '/share(/.*)?'
restorecon -R /share
```

Firewall:

```
firewall-cmd --add-service=samba --permanent
firewall-cmd --reload
```

Connect: `smbclient //localhost/share`

Mount the share:

```
yum install cifs-utils
mount.cifs //172.16.1.50/share /mnt
```

Or use autofs with an `/etc/auto.misc` entry:

```
samba   -fstype=cifs,rw,noperm,user=root,password=PASSWORD ://172.16.1.50
/share
```

GRUB

- Configuration:
  - `/etc/default/grub`
  - `/boot/grub2/grub.cfg`
  - `/etc/grub.d/`

Command	Description
`grubby --default-kernel`	Displays the default kernel
`grubby --set-default /boot/<kernel>`	Change the default
`grubby --info=ALL`	Info for all known kernels
`grubby --info /boot/<kernel>`	get info about a specific kernel
`grubby --remove-args="rhgb quiet" –update-kernel /boot/<kernel>`	Change a kernel arg
`grub2-mkconfig -o /boot/grub2/grub.cfg`	Used when changing config files outside `grubby`

PXE Boot

Packages: `syslinux tftp tftp-server`

Copy across files:

```
cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot/
```

From, the ISO, also copy `vmlinuz` and `initrd.img`

Start the tftp server

```
systemctl start tftp.socket
systemctl enable tftp.socket
```

DHCP

Edit `/etc/dhcp/dhcp.conf` and in the subnet config, add:

```
next-server <ip of pxe server>;
filename “pxelinux.0”;
```

Test the config and restart:

```
dhcpd -t -cf /etc/dhcp/dhcp.conf
systemctl restart dhcpd
```

PXE Boot menu

Configuration file: `/var/lib/tftpboot/pxelinux.cfg/default`

Example:

```
default menu.32
prompt 0
timeout  1000
ontimeout local

menu title Boot menu

label local
    menu Boot from local disk
    LOCALBOOT 0
```

Services

FTP

Package: `vsftpd`

Configuration: `/etc/vsftpd/vsftpd.conf`

MariaDB

Package: `mariadb-server`

HTTP

Packages: `httpd`

Firewall:

```
firewall-cmd --add-service={http,https} --permanent
```

SSL

Package: `httpd`

Configuration: `/etc/httpd/conf.d/ssl.conf`

Create a key and add it:

```
openssl req -new -nodes -x509 -keyout canary.key -out canary.crt
chmod 400 canary.key canary.crt
cp canary.crt /etc/pki/tls/certs/
cp canary.key /etc/pki/tls/private/
```

PHP

Package: `httpd mod_php`

Configuration: `/etc/httpd/conf.d/php.conf`

Mail

Package: `postfix`

Configuration: `/etc/postfix/main.cf`

Sample MX entry:

```
lab.example.com. MX 10 mail.lab.example.com
```

Aliases in `/etc/aliases` - run `newaliases` after edits.

SMTP Relay

On the host mail server:

```
postconf -e inet_protocols=ipv4
postconf -e inet_interfaces=all
postconf -e mydestination=localhost,lab.example.com,router.lab.example.com
systemctl restart postfix

firewall-cmd --add-service=smtp --permanent
firewall-cmd --reload

postconf mynetworks
```

On the client:

```
postconf -e inet_protocols=ipv4
postconf -e inet_interfaces=all
postconf -e relayhost=mail.lab.example.com
postfix check
systemctl restart postfix
```

IMAP

Package: `dovecot`

Configuration: - `/etc/dovecot/dovecot.conf` - `/etc/dovecot/conf.d`

Firewall:

```
firewall-cmd --add-service={imap,imaps} --permanent --zone=lab-internal
firewall-cmd --reload
```

In `/etc/dovecot/dovecot.conf`:

```
protocols = imap pop3 lmtp
listen = *
```

In `/etc/dovecot/conf.d/10-mail.conf`:

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

Add appropriate users to the `mail` group:

```
usermod -a -G mail puffin
```

Check with `mutt`:

```
mutt -f imaps://user@hostname/
```