Linux Root Diectory Strucuture

/ — The Root Directory

Everything on your Linux system is located under the / directory, known as the root directory. You can think of the / directory as being similar to the C:\ directory on Windows — but this isn't strictly true, as Linux doesn't have drive letters. While another partition would be located at D:\ on Windows, this other partition would appear in another folder under / on Linux.

/bin — Essential User Binaries

The /bin directory contains the essential user binaries (programs) that must be present when the system is mounted in single-user mode. Applications such as Firefox are stored in /usr/bin, while important system programs and utilities such as the bash shell are located in /bin. The /usr directory may be stored on another partition — placing these files in the /bin directory ensures the system will have these important utilities even if no other file systems are mounted. The /sbin directory is similar — it contains essential system administration binaries.

/boot — Static Boot Files

The /boot directory contains the files needed to boot the system — for example, the GRUB boot loader's files and your Linux kernels are stored here. The boot loader's configuration files aren't located here, though — they're in /etc with the other configuration files.

/cdrom — Historical Mount Point for CD-ROMs

The /cdrom directory isn't part of the FHS standard, but you'll still find it on Ubuntu and other operating systems. It's a temporary location for CD-ROMs inserted in the system. However, the standard location for temporary media is inside the /media directory.

/dev — Device Files

Linux exposes devices as files, and the /dev directory contains a number of special files that represent devices. These are not actual files as we know them, but they appear as files — for example, /dev/sda represents the first SATA drive in the system. If you wanted to partition it, you could start a partition editor and tell it to edit /dev/sda.

This directory also contains pseudo-devices, which are virtual devices that don't actually correspond to hardware. For example, /dev/random produces random numbers. /dev/null is a special device that produces no output and automatically discards all input — when you pipe the output of a command to /dev/null, you discard it.

/etc — Configuration Files

The /etc directory contains configuration files, which can generally be edited by hand in a text editor. Note that the /etc/ directory contains system-wide configuration files — user-specific configuration files are located in each user's home directory.

/home — Home Folders

The /home directory contains a home folder for each user. For example, if your user name is bob, you have a home folder located at /home/bob. This home folder contains the user's data files and user-specific configuration files. Each user only has write access to their own home folder and must obtain elevated permissions (become the root user) to modify other files on the system.

/lib — Essential Shared Libraries

The /lib directory contains libraries needed by the essential binaries in the /bin and /sbin folder. Libraries needed by the binaries in the /usr/bin folder are located in /usr/lib.

/lost+found — Recovered Files

Each Linux file system has a lost+found directory. If the file system crashes, a file system check will be performed at next boot. Any corrupted files found will be placed in the lost+found directory, so you can attempt to recover as much data as possible.

/media — Removable Media

The /media directory contains subdirectories where removable media devices inserted into the computer are mounted. For example, when you insert a CD into your Linux system, a directory will automatically be created inside the /media directory. You can access the contents of the CD inside this directory.

/mnt — Temporary Mount Points

Historically speaking, the /mnt directory is where system administrators mounted temporary file systems while using them. For example, if you're mounting a Windows partition to perform some file recovery operations, you might mount it at /mnt/windows. However, you can mount other file systems anywhere on the system.

/opt — Optional Packages

The /opt directory contains subdirectories for optional software packages. It's commonly used by proprietary software that doesn't obey the standard file system hierarchy — for example, a proprietary program might dump its files in /opt/application when you install it.

/proc — Kernel & Process Files

The /proc directory similar to the /dev directory because it doesn't contain standard files. It contains special files that represent system and process information.

/root — Root Home Directory

The /root directory is the home directory of the root user. Instead of being located at /home/root, it's located at /root. This is distinct from /, which is the system root directory.

/run — Application State Files

The /run directory is fairly new, and gives applications a standard place to store transient files they require like sockets and process IDs. These files can't be stored in /tmp because files in /tmp may be deleted.

/sbin — System Administration Binaries

The /sbin directory is similar to the /bin directory. It contains essential binaries that are generally intended to be run by the root user for system administration.

/selinux — SELinux Virtual File System

If your Linux distribution uses SELinux for security (Fedora and Red Hat, for example), the /selinux directory contains special files used by SELinux. It's similar to /proc. Ubuntu doesn't use SELinux, so the presence of this folder on Ubuntu appears to be a bug.

/srv — Service Data

The /srv directory contains "data for services provided by the system." If you were using the Apache HTTP server to serve a website, you'd likely store your website's files in a directory inside the /srv directory.

/tmp — Temporary Files

Applications store temporary files in the /tmp directory. These files are generally deleted whenever your system is restarted and may be deleted at any time by utilities such as tmpwatch.

/usr — User Binaries & Read-Only Data

The /usr directory contains applications and files used by users, as opposed to applications and files used by the system. For example, non-essential applications are located inside the /usr/bin

directory instead of the /bin directory and non-essential system administration binaries are located in the /usr/sbin directory instead of the /sbin directory. Libraries for each are located inside the /usr/lib directory. The /usr directory also contains other directories — for example, architecture-independent files like graphics are located in /usr/share.

The /usr/local directory is where locally compiled applications install to by default — this prevents them from mucking up the rest of the system.

/var — Variable Data Files

The /var directory is the writable counterpart to the /usr directory, which must be read-only in normal operation. Log files and everything else that would normally be written to /usr during normal operation are written to the /var directory. For example, you'll find log files in /var/log.

IT601P – System and Network Administration
(Practical)

P1 - Walk Through Linux Server

Arif Husen

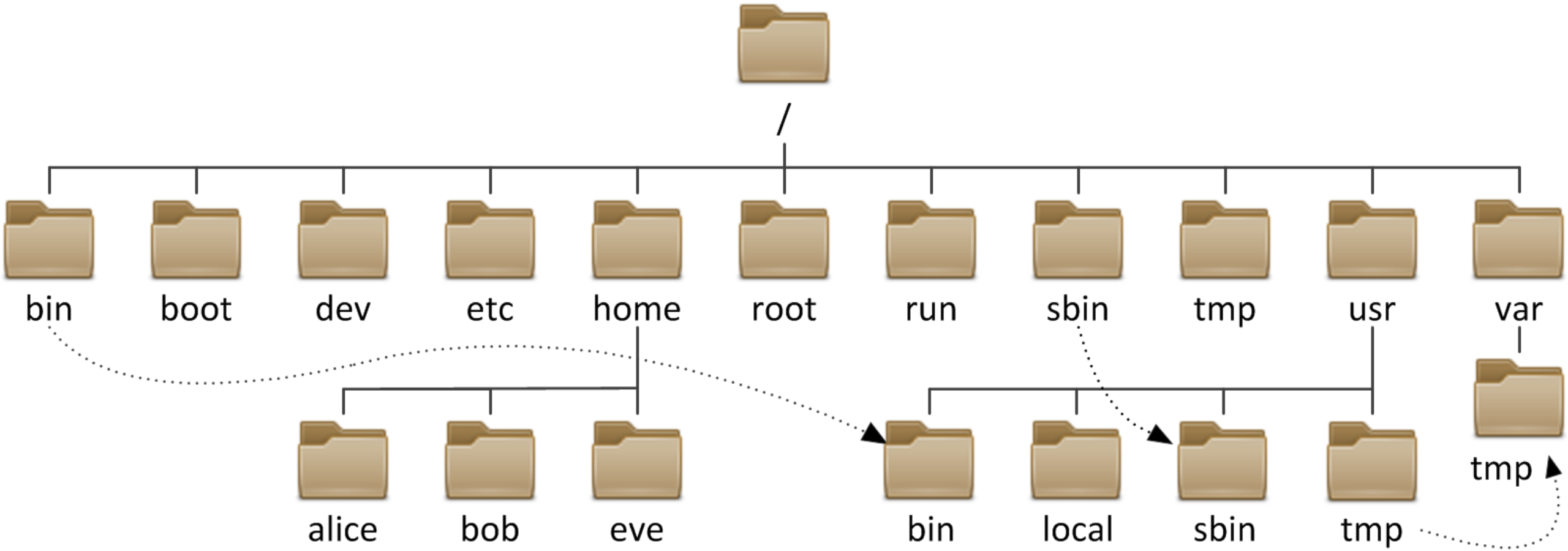
Department of Computer Science and Information Technology,
Virtual University of Pakistan

Contents

- Installing Linux Server
  - Preparing Oracle Virtual Box
  - Downloading Linux Server OS
  - Setting Up OS

- Walk Through Linux Server Basic Commands
  - Creating folders and files
  - Creating files using redirection
  - Moving and manipulating files
  - About Files
  - The command line and the superuser
  - Hidden files
  - Mount points

Directory Structure



Directory Structure

/ — The Root Directory

/bin — Essential User Binaries

/boot — Static Boot Files

/cdrom — Historical Mount Point for CD-ROMs

/dev — Device Files

/etc — Configuration Files

/home — Home Folders

/lib — Essential Shared Libraries

/lost+found — Recovered Files

/media — Removable Media

/mnt — Temporary Mount Points

Directory Structure

/opt — Optional Packages

/proc — Kernel & Process Files

/root — Root Home Directory

/run — Application State Files

/sbin — System Administration Binaries

/selinux — SELinux Virtual File System

/srv — Service Data

/tmp — Temporary Files

/usr — User Binaries & Read-Only Data

/var — Variable Data Files

IT601P – System and Network Administration
(Practical)

P1.2 - Walk Through Linux Server

Arif Husen

Department of Computer Science and Information Technology,
Virtual University of Pakistan

Walk Through Linux Server Basic Commands

- Location
- Creating folders and files
- Creating files using redirection
- Moving and manipulating files
- About Files
- The command line and the superuser
- Hidden files
- Mount points

Location

- pwd
- cd
- Relative path
- Absolute path

IT601P – System and Network Administration <Practical>

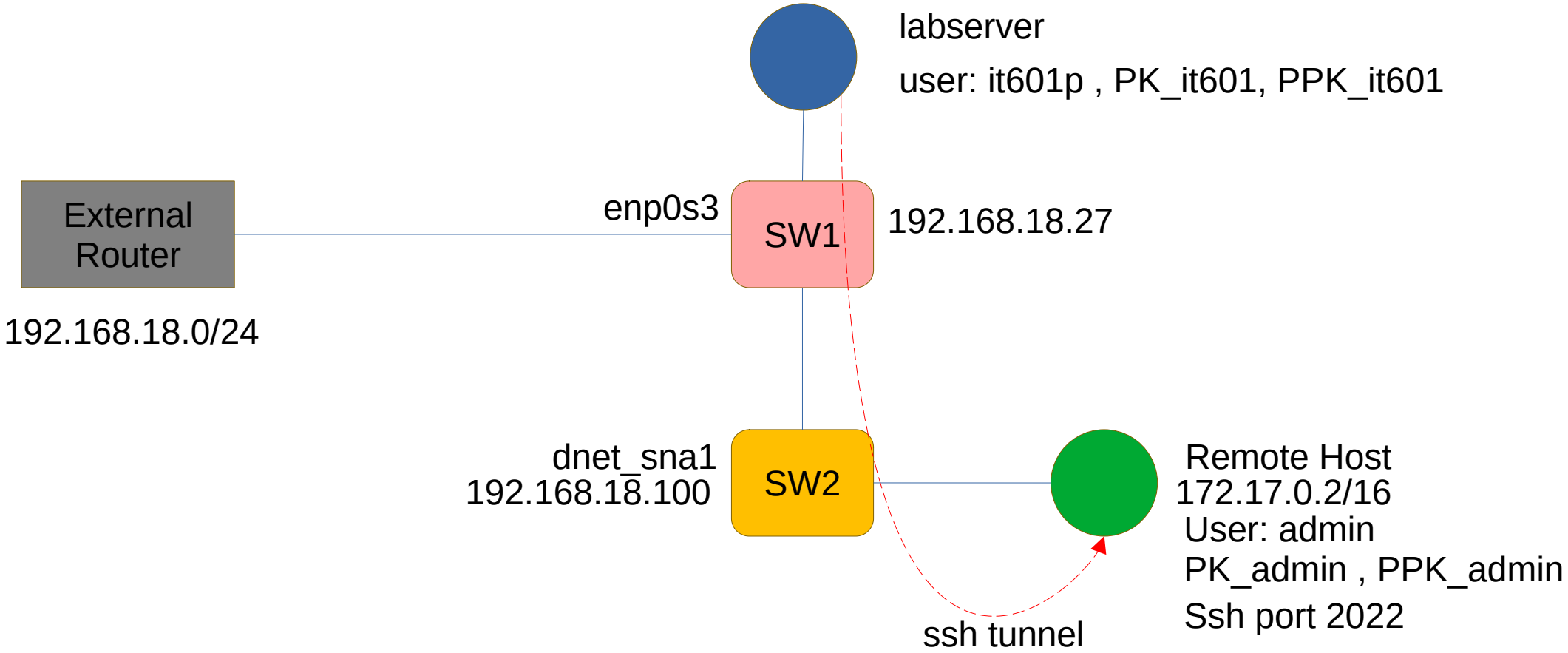
Enabling and Configuring SSH Access on Linux Systems

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

Lab Setup


labserver
user: it601p , PK_it601, PPK_it601
External Router
enp0s3
SW1
192.168.18.27
192.168.18.0/24
dnet_sna1
192.168.18.100
SW2
Remote Host
172.17.0.2/16
User: admin
PK_admin , PPK_admin
Ssh port 2022
ssh tunnel

SSH Access with Remote Username and Password

1 – view Ip address information

$ Ifconfig -a |more
$ Ip a

2 – Update APT repositories and install openssh-server

$ sudo apt update
$ sudo apt install openssh-server

3 – Check status of the ssh server

$ sudo systemctl status ssh

4 – Check status current sshd configuration and default port to 2022

$ sudo Vi */etc/ssh/sshd_conf*

5 – connect remote server with ssh on custom port and using remote user password

$ ssh -p 2022 <remoteUsername>@<remoteIpAddress>

SSH Access with Public Key (password less)

1 – Generate RSA keys at labserver

```
$ ssh-keygen -t rsa -b 4096
#define passphrase
```

2 – verify key pair

```
$ ls !/.ssh/
```

3 – copy public key of lab server user to remove server

```
$ ssh-copy-id  “-p <remotePort> <remoteUsername>@<remoteIpAddress>”
```

4 – connect remote server with ssh on custom port without remote user password

```
$ ssh -p 2022 <remoteUsername>@<remoteIpAddress>
```

Docker file

- **Docker Image:**

  A Docker Image is a read-only file with a bunch of instructions. When these instructions are executed, it creates a Docker container.

- **Dockerfile:**

  Dockerfile is a simple text file that consists of instructions to build Docker images.

- **Syntax**

```
# comments
command argument argument1...
```

- **Docker Commands for Creating a Dockerfile**

FROM - Creates a layer from the ubuntu:18.04

FROM ubuntu:18.04

PULL - Adds files from your Docker repository

PULL. /file

RUN - Builds your container

RUN make /file

CMD - Specifies what command to run within the container

CMD python /file/file.py

ENTRYPOINT

- allows specifying a command along with the parameters
- Syntax, ENTRYPOINT application "arg, arg1".
- Example , ENTRYPOINT echo "Hello, $name".

ADD

- ADD command helps in copying data into a Docker image
- Syntax , ADD /[source]/[destination]
- Example , ADD /root_folder/test_folder

ENV

- ENV provides default values for variables that can be accessed within the container
- Syntax , ENV key value
- Example , ENV value_1

MAINTAINER

- MAINTAINER declares the author field of the images
- Syntax , MAINTAINER [name]
- Example , MAINTAINER author_name

Docker Commands

Name	Description
attach	Attach local standard input, output, and error streams to a running container
build	Build an image from a Dockerfile
commit	Create a new image from a container's changes
config	Manage Swarm configs
container	Manage containers
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
exec	Execute a command in a running container
export	Export a container's filesystem as a tar archive
image	Manage images
images	List images
inspect	Return low-level information on Docker objects
kill	Kill one or more running containers
load	Load an image from a tar archive or STDIN
network	Manage networks
port	List port mappings or a specific mapping for the container
ps	List containers
rename	Rename a container
restart	Restart one or more containers
rm	Remove one or more containers
rmi	Remove one or more images
run	Create and run a new container from an image
save	Save one or more images to a tar archive (streamed to STDOUT by default)
stats	Display a live stream of container(s) resource usage statistics
stop	Stop one or more running containers
top	Display the running processes of a container
volume	Manage volumes

Build a Docker Image with Dockerfile

- Create a directory where the dockerfile will be stored.
  - mkdir c1 && cd c1
- Create dockerfile
  - sudo vi dockerfile

```
####Contents of Dockerfile####

FROM ubuntu:22.04

MAINTAINER Virtual_University_of_Pakistan

ENV DEBIAN_FRONTEND noninteractive

RUN apt-get update && apt-get install -y ubuntu-server

EXPOSE 2022
#expose the port 2022

CMD ["echo", "Welcome to it601p"]
#above command will print the message and container will exit.

#To keep it running  run bash and keep it running

CMD [“PROGRAM_TO_RUN”]
```

Build a Docker Image with Dockerfile

- **Build the image using above docker file**
  - docker build <options>
  - docker build --help

- **Some Options**

  --file, -f : docker file name

  --label : set meta data for image

  --quite -q : suppress output

  --rm : Remove intermediate containers

  --tag, -t : set name or tag

  --add-host : Add custom host to IP mapping

  --network : set the networking mode for run commands

- **Upon completion , it returns the image id**

Option	Description
--build-arg	Set build-time variables
--cache-from	Images to consider as cache sources
--cgroup-parent	Optional parent cgroup for the container
--compress	Compress the build context using gzip
--cpu-period	Limit CPU CFS (Completely Fair Scheduler) period
--cpu-quota	Limit the CPU CFS (Completely Fair Scheduler) quota
--cpu-shares , -c	CPU shares (relative weight)
--cpuset-cpus	CPUs in which to allow execution (0-3, 0,1)
--cpuset-mems	MEMs in which to allow execution (0-3, 0,1)
--disable-content-trust	TRUE, Skip image verification
--iidfile	Write the image ID to the file
--isolation	Container isolation technology
--memory , -m	Memory limit
--memory-swap	Swap limit equal to memory plus swap: -1 to enable unlimited swap
--no-cache	Do not use cache when building the image
--platform	Set platform if server is multi-platform capable
--pull	Always attempt to pull a newer version of the image
--security-opt	Security options
--shm-size	Size of /dev/shm
--squash	Squash newly built layers into a single new layer
--target	Set the target build stage to build.
--ulimit	Ulimit options

Build a Docker Image with Dockerfile

- **Verify the image is crated**
  - docker images
- **Run an image**
  - docker run name:tag <options>

--hostname , -h	Container host name
--interactive , -i	Keep STDIN open even if not attached
--ip	IPv4 address (e.g., 172.30.100.104)
--label , -l	Set meta data on a container
--link	Add link to another container
--mount	Attach a filesystem mount to the container
--name	Assign a name to the container
--network	Connect a container to a network
--rm	Automatically remove the container when it exits
--mac-address	Container MAC address (e.g., 92:d0:c6:0a:29:33)
--memory , -m	Memory limit
--volume , -v	Bind mount a volume
--workdir , -w	Working directory inside the container

Create a New Container

Now, create a Docker container from the Docker image we created in the previous step.

```
docker run --name it601p_m1
```

IT601P – System and Network Administration
(Practical)

Lab 8 – DNS Service

Arif Husen

Department of Computer Science and Information Technology,
Virtual University of Pakistan

DNS Service

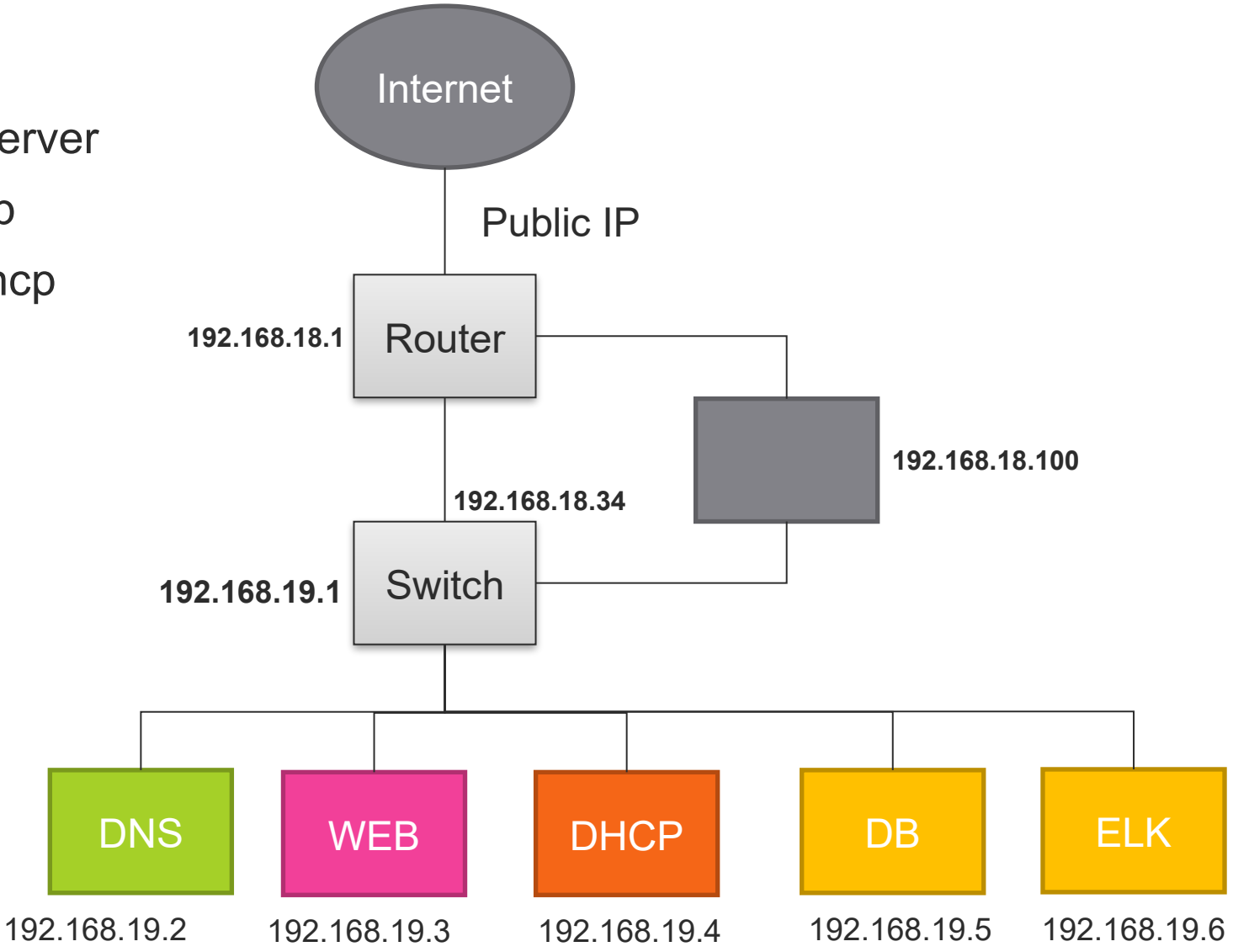
Packages to install

1. bind9
2. bind9utils
3. bind9-doc

Sudo apt aupdate && Sudo apt install bind9 bind9utils bind9-doc

Information :

1. Local Networks : 192.168.19.0/24
2. DNS SERVER: 192.168.19.2 , bindserver
3. WEB_SERVER : 192.168.19.3 , web
4. DHCP_SERVER : 192.168.19.4 , dhcp
5. DB_SERVER : 192.168.19.5 , db
6. ELK_SERVER : 192.168.19.6 , elk
7. DOMAIN : vusna.com



Configure the /etc/bind/named.conf.options

```
acl LAN {
192.168.0.0/16;
};
options {
        directory "/var/cache/bind";
        allow-query { localhost; LAN; };
        forwarders { 192.168.18.1; };
        recursion yes;
};
```

Verifying Configuration Changes

named-checkconf /etc/bind/named.conf.options

configure /etc/bind/named.conf.local

```
zone "vusna.com" IN {
        type master;
       file "/etc/bind/zones/vusna.com";
};

zone "0.116.10.in-addr.arpa" IN {
       type master;
       file "/etc/bind/zones/vusna.com.rev";
};
```

named-checkconf /etc/bind/named.conf.options

1) mkdir /etc/bind/zones

2) cp /etc/bind/db.local /etc/bind/zones/vusna.com

3) vi /etc/bind/zones/vusna.com

VUSNA.COM Zone Configuration

```
$TTL    604800
@       IN          SOA     vusna.com. root.vusna.com. (3; 604800; 86400; 2419200; 604800) ;
@       IN          NS      bindserver.vusna.com.
bindserver          IN      A           192.168.2.2
 web        IN      A           192.168.2.3
 dhcp       IN      A           192.168.2.4
 db         IN      A           192.168.2.5
 elk        IN      A           192.168.2.6
```

Create Reverse Zone file

1. cd /etc/bind/zones/
2. cp /etc/bind/db.127 /etc/bind/zones/vusna.com.rev
3. vi /etc/bind/zones/vusna.com.rev

Reverse Zone Configurations

```
$TTL    604800
@       IN      SOA     vusna.com. root.vusna.com. (
                              2   ;
                         604800 ;
                          86400 ;
                        2419200 ;
                         604800 )  ;

@       IN      NS      bindserver.vusna.com.
bindserver      IN      A       192.168.18.1
2   IN      PTR bindserver.vusna.com
3   IN      PTR     web.vusna.com
4   IN      PTR     dhcp.vusna.com
5   IN      PTR     db.vusna.com
6   IN      PTR     elk.vusna.com
```

named-checkzone vusna.com /etc/bind/zones/vusna.com.rev

Test Service

1) systemctl restart bind9

2) nslookup client1

IT601P – System and Network Administration
(Practical)

Lab 9 – Multi Webservices on Single Web Server with FQDNs

Arif Husen

Department of Computer Science and Information Technology,
Virtual University of Pakistan

Tasks

Configure Web Servers for two sites with FQDN

1 – it601.vusna.com
2 – it601p.vusna.com

Steps

1 - Create two conf files in /etc/apache2/

1- it601.conf
2- it601p.conf

2 – Create two content folders for it601 and it601p.

2 - Update ServerAdmin, ServerName and ServerAlias and Document root

3 – Reload Apache Server

4 – add A and PTR records for both services to web server
1 – comment out the existing web records
2 – Add A record for it601.vusna.com
3 – Add A record for it601.vusna.com
4 – Add PTR record for it601.vusna.com
5 – Add PTR record for IT601p.vusna.com

5 – Reload DNS Server

6 - Test the new services with cURL

IT601 – System and Network Administration

Lab – 10 : Configurations with htaccess

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

.htaccess

- **The .htaccess file in Apache is a tool that allows configurations at the directory and subdirectory level.**

- **Using .htaccess enables you to configure website permissions without altering server configuration files.**

- **Enabling .htaccess**
  - sudo vi /var/www/my_website.com/.htaccess
  - sudo vi /user/safe_location/.htpasswd

- **Common Uses**
  - Manage IP Addresses
  - Block Visitors by Referrer
  - Redirect Traffic
  - Set a 404 Page

Enable authentication

1 – Create password with htpasswd on secure location

```
htpasswd –c  <secure_location> <password_filename> <user_id>
```

2 – Create .htaccess file in the folder to be protected

```
vi  .htaccess
```

3 – Write the following lines in above file

```
 AuthUserFile /user/safe_location/.htpasswd
AuthGroupFile /dev/null
AuthName "Please Enter Password"
AuthType Basic
Require valid-user
```

4 - Reload the apache2

Allow. Deny IP Addresses

1 – Add the following lines to the htaccess file in the folder where restrictions are to be applied.

```
order deny, allow
deny from 192.168.0.54
allow from 192.168.0
```

Set a 404 Page

1 – Add the following lines to the htaccess file.

```
ErrorDocument 404 /404.html
```

➢Redirect Traffic

1 – Add the following lines to the htaccess file in the folder where redirection is to be applied.

```
Redirect301/Other_Website.com/index.html/My_Website.com/index.html
```

Block Visitors by Referrer

1 – Add the following lines to the htaccess file in the folder where restrictions are to be applied.

```
RewriteEngine on
# Options +FollowSymlinks
RewriteCond %{HTTP_REFERER} blockeddomain\.com [NC]
RewriteRule .* - [F]
```

IT601P – System and Network Administration
(Practical)

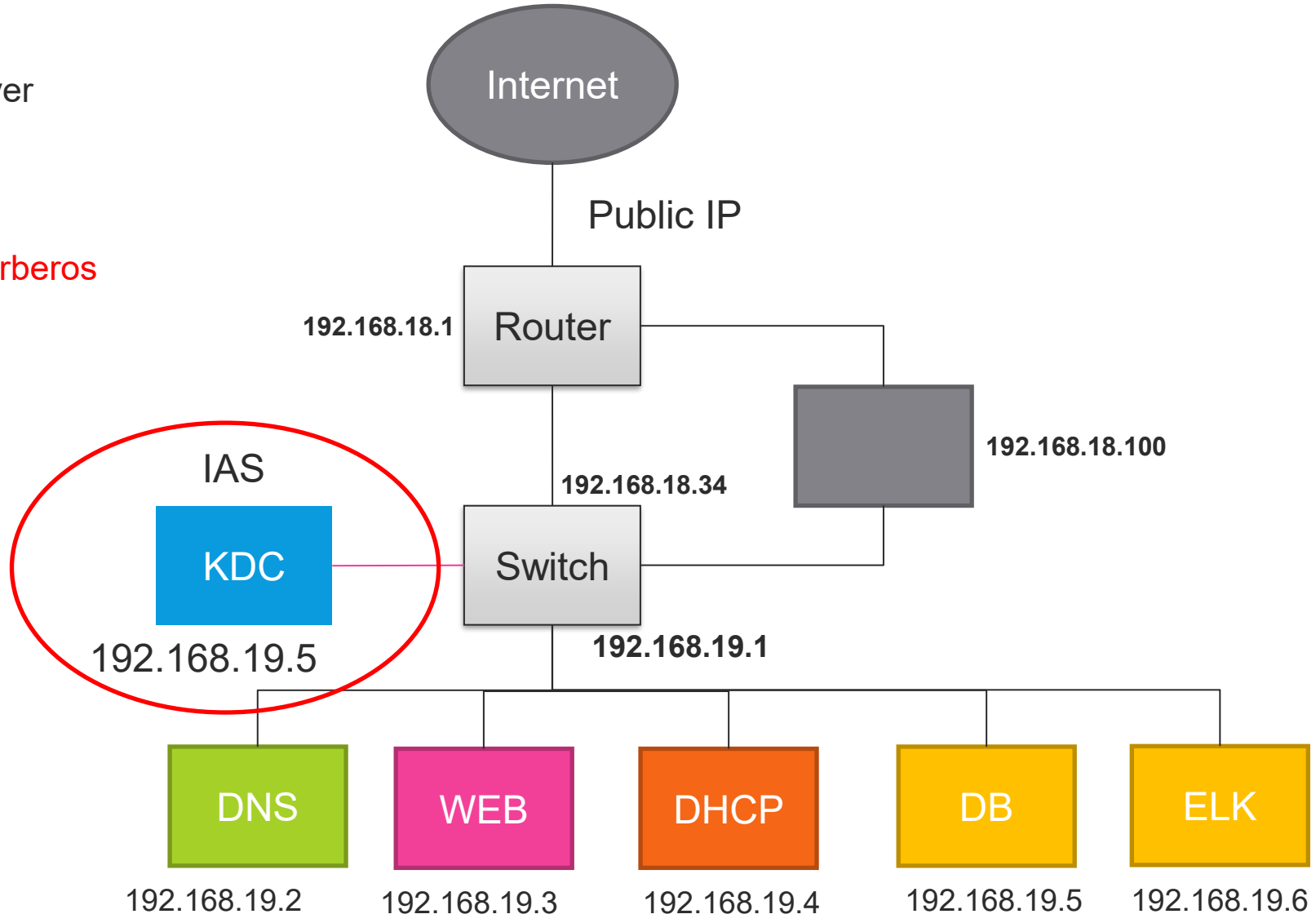
Lab 11 – Kerberos IAS

Arif Husen

Department of Computer Science and Information Technology,
Virtual University of Pakistan

Lab Environment

1. Local Networks : 192.168.19.0/24
2. DNS SERVER: 192.168.19.2 , bindserver
3. WEB_SERVER : 192.168.19.3 , web
4. DHCP_SERVER : 192.168.19.4 , dhcp
5. kerberos_SERVER : 192.168.19.5 , kerberos
6. ELK_SERVER : 192.168.19.6 , elk
7. DOMAIN : vusna.com



Tasks

- **Basic Information**
  - Realm: vusna.com
  - Primary KDC: kerberos.vusna.com
  - User principal: sna
  - Admin principal: sna/admin

Prerequisites

1- Setup the new server machine

2- assign IP addressing, SSH and Console Access

3- Set hostname

KDC Installation

1 - Install the Kerberos Packages

- krb5-kdc
- krb5-admin-server
- sudo apt install krb5-kdc krb5-admin-server

2 - Create the new realm with the kdb5_newrealm utility

- sudo krb5_newrealm

3 - you can reconfigure Kerberos from scratch, if need to change the realm name. Following commands can be used for this purpose.

- sudo dpkg-reconfigure krb5-kdc

KDC Configuration

4 - Create the first principal.

```
sudo kadmin.local

Authenticating as principal root/admin@vusna.com with password.

kadmin.local: addprinc sna

WARNING: no policy specified for sna@vusna.com; defaulting to no policy

Enter password for principal “sna@vusna.com":

Re-enter password for principal " sna@vusna.com “:

Principal “sna@ vusna.com" created.

kadmin.local: quit
```

Create Admin Principal

5 - Create an admin principal and assign permissions.

```
$ sudo kadmin.local

Authenticating as principal root/admin@vusna.com with password.

kadmin.local: addprinc sna/admin

WARNING: no policy specified for sna/admin@vusna.com; defaulting to no policy

Enter password for principal " sna/admin@vusna.com ":

Re-enter password for principal " sna/admin@vusna.com ":

Principal "sna/admin@vusna.com" created.

kadmin.local: quit
```

Create Admin Principal

6 - Set admin principal with appropriate Access Control List (ACL) permissions. The permissions are configured in the /etc/krb5kdc/kadm5.acl file:

```
sna/admin@vusna.com      *
```

The above will grant all privileges to any admin instance of a principal. See the kadm5.acl manpage for details.

7 - Restart the krb5-admin-server for the new ACL to take affect:

```
sudo systemctl restart krb5-admin-server.service
```

8 - The new user principal can be tested using the kinit utility:

```
$ kinit sna/admin
Password for sna/admin@vusna.com:
```

Verifying and Completing kerberos

9 - Verify information about the Ticket Granting Ticket (TGT):

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: sna/admin@vusna.com

Valid starting     Expires            Service principal
04/03/20 19:16:57  04/04/20 05:16:57  krbtgt/vusna.com@vusna.com
        renew until 04/04/20 19:16:55
```

- Your new Kerberos Realm is now ready to authenticate clients.

IT601P – System and Network Administration
(Practical)

Lab 12 – Installing LDAP

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

Installing LDAP

- sudo apt install slapd ldap-utils
- sudo dpkg-reconfigure slapd
- configure the OpenLDAP libraries with certain defaults in /etc/ldap/ldap.conf

```
BASE dc=vusna,dc=com
URI ldap://ldap.vusna.com
```

After installation, you will get two databases, or suffixes:

1. For data, based on your host's domain (dc=vusna,dc=com),
2. For configuration, with its root at cn=config.

To change the data on each we need different credentials and access methods:

dc=example,dc=com
The administrative user for this suffix is cn=admin,dc=example,dc=com and its password is the one selected during the installation of the slapd package.

cn=config
The configuration of slapd itself is stored under this suffix.

View Information

- **View the slapd-config DIT looks like via the LDAP protocol (listing only the DNs):**

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn
```

Q : SASL Quite mode
LLL : Print responses in LDIF format without comments
-Y : SASL mechanism
-H : LDAP URL
-b : basedn

```
dn: cn=config
dn: cn=module{0},cn=config
dn: cn=schema,cn=config
dn: cn={0}core,cn=schema,cn=config
dn: cn={1}cosine,cn=schema,cn=config
dn: cn={2}nis,cn=schema,cn=config
dn: cn={3}inetorgperson,cn=schema,cn=config
dn: olcDatabase={-1}frontend,cn=config
dn: olcDatabase={0}config,cn=config
dn: olcDatabase={1}mdb,cn=config
```

View Information

This is what the dc=vusna,dc=com DIT looks like:

```
$ ldapsearch -x -LLL -H ldap:/// -b
dc=vusna,dc=com dn

dn: dc=vusna,dc=com

dn: cn=admin,dc=vusna,dc=com
```

verify the authentication with ldapwhoami

```
$ ldapwhoami -x
anonymous

$ ldapwhoami -x -D cn=admin,dc=vusna,dc=com -W
Enter LDAP Password:
dn:cn=admin,dc=vusna,dc=com
```

Populating the directory

- **We will add the following:**

  A node called People (to store users)

  A node called Groups (to store groups)

  A group called academics

  A user called arif

Populating the directory

- **Create the following LDIF file and call it ldpa_data.ldif:**

```
dn: ou=People,dc=vusna,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc= vusna,dc=com
objectClass: organizationalUnit
ou: Groups

dn: cn=csd,ou=Groups,dc=
vusna,dc=com
objectClass: posixGroup
cn: csd
gidNumber: 5000
```

```
dn: uid=arif,ou=People,dc=vusna,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: arif
sn: husen
givenName: Rashid
cn: Arif Rashid
displayName: Arif H Rashid
uidNumber: 10000
gidNumber: 5000
userPassword: {CRYPT}x
gecos: Arif Rashid
loginShell: /bin/bash
homeDirectory: /home/arif
```

Populating the directory

- **Add the ldif data**

```
$ ldapadd -x -D cn=admin,dc=vusna.com,dc=com -W -f ldap_data.ldif
Enter LDAP Password: ********

adding new entry "ou=People,dc=vusna,dc=com"
adding new entry "ou=Groups,dc= vusna,dc=com"
adding new entry "cn=miners,ou=Groups,dc= vusna,dc=com"
adding new entry "uid=arif,ou=People,dc= vusna,dc=com"
```

- **check that the information has been correctly added with the ldapsearch utility.**

```
$ ldapsearch -x -LLL -b dc= vusna,dc=com '(uid=arif)' cn gidNumber
dn: uid=arif,ou=People,dc= vusna,dc=com
cn: Arif Rashid
gidNumber: 5000
```

Change Password

- **To change the password to something valid, you can now use ldappasswd:**

```
$ ldappasswd -x -D cn=admin,dc=vusna,dc=com -W -S
uid=john,ou=people,dc= vusna,dc=com
New password:
Re-enter new password:
Enter LDAP Password:
```

Seting up the Accoun Manager

- **Install the following packages required for account manager**
  - apache2
  - php
  - php-cgi
  - libapache2-mod-php
  - php-mbstring
  - php-common php-pear

```
sudo apt -y install apache2 php php-cgi libapache2-mod-php php-mbstring
php-common php-pear
```

- **enable php-cgi PHP extension.**

```
sudo a2enconf php*-cgi
sudo systemctl reload apache2
```

- **Install LDAP Account Manager**

```
sudo apt -y install ldap-account-manager
```

- **Acessing the Account Manager**

  http://(server's hostname or IP address)/lam

Configure the account manager

- The LDAP Account Manager Login form will be shown. We need to set our LDAP server profile by clicking on

  LAM configuration -> Edit server profiles

  This will ask you for LAM Profile name Password:

  Default password is lam

- change is Profile Password, this is at the end of General Settings page.

- Set LDAP Server address and Tree suffix.

- Set Dashboard login by specifying the admin user account and domain components under “Security settings” section.

- Switch to “Account types” page and set Active account types LDAP suffix and List attributes.

Configure the account manager

- Add user accounts and groups with LDAP Account Manager
- Login with the accountadmin to LAM dashboard to start managing user accounts and groups.
- Then Follow the intuitive interface

Configure the LDAP Client

- sudo apt-get update
- sudo apt -y install libnss-ldap libpam-ldap ldap-utils
- Configure authentication:

After the installation, edit /etc/nsswitch.conf and add LDAP authentication to passwd and group lines.

```
passwd: compat systemd ldap
group:    compat systemd ldap
shadow: compat ldap
```

Integeration of LDAP with kerberos

- Modify the file /etc/pam.d/common-password. Remove use_authtok on line 26 to look like below.

```
password [success=1 user_unknown=ignore
default=die] pam_ldap.so try_first_pass
```

- Enable creation of home directory on the first login by adding the following line to the end of file /etc/pam.d/common-session

```
session optional pam_mkhomedir.so skel=/etc/skel umask=077
```

- Restart the nscd service.

```
root@server1:~# su - arif
```

IT601P – System and Network Administration
(Practical)

Lab 12 – Configuring LDAP Clients

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

Configure the LDAP Client

- sudo apt-get update
- sudo apt -y install libnss-ldap libpam-ldap ldap-utils
- Configure authentication:

After the installation, edit /etc/nsswitch.conf and add LDAP authentication to passwd and group lines.

```
passwd: compat systemd ldap
group:    compat systemd ldap
shadow: compat ldap
```

Integeration of LDAP with kerberos

- Modify the file /etc/pam.d/common-password. Remove use_authtok on line 26 to look like below.

```
password [success=1 user_unknown=ignore
default=die] pam_ldap.so try_first_pass
```

- Enable creation of home directory on the first login by adding the following line to the end of file /etc/pam.d/common-session

```
session optional pam_mkhomedir.so skel=/etc/skel umask=077
```

- Restart the nscd service.

```
root@web_server:~# su - arif
```

Practice Tasks

- Configure LDAP client with password authentication

- Configure Windows client with password authentication

- Configure LDAP as backend to Kerberos Authentication

IT601P – System and Network Administration
(Practical)

Lab 15 – Email Service

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

Installing Email Service

- **Add MX entry to DNS server**

```
email        IN          A     192.168.19.9
9            IN          MX    email.vusna.com
```

- **Install Postfix**
  - **sudo apt-get update && apt install postfix**

- **Select Internet Site when prompted**

  - **No configuration :** installation process will not config any parameters.
  - **Internet Site** : using Postfix for sending emails to other MTAs and receiving email from other MTAs.
  - **Internet with smarthost :** means using postfix to receive email from other MTAs, but using another smart host to relay emails to the recipient.
  - **Satellite system :** means using smart host for sending and receiving email.
  - **Local** only means emails are transmitted only between local user accounts.

Installing Email Service

- Specify System Mail Name : email.vusna.com
- Check postfix version with command : postconf mail_version
- Use ss (Socket Statistics) utility to check status of SMTP port 25
- Install telnet and confirm the opened ports : apt install telnet
  - telnet email.vusna.com 25
- Cread user admin and set password
  - Adduser admin mail
- Use Mail Program to send and receive email
  - sudo apt-get install mailutils
  - mail -a FROM:admin@email.vusna.com admin@email.vusna.com
  - Check email with mail

Installing Email Service

- **Create Email Alias**

```
Vi /etc/aliases
Postmaster : admin
Contactus: admin
```

- **Load new aliases**

```
sudo newaliases
```

- **Set other parameters like attachment size, protocols in /etc/postfixmain.cf**

```
inet_protocols = all
```

```
inet_protocols = ipv4
```

```
mailbox_size_limit = 0
```

Install and configure IMAP and POP3 Services

Enable Submission Service in Postfix

```
vi /etc/postfix/master.cf
```

```
submission    inet    n    -    y    -    -    smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_tls_wrappermode=no
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
  -o smtpd_sasl_type=dovecot
  -o smtpd_sasl_path=private/auth
```

Enable SMTP over Port 465

vi /etc/postfix/master.cf

```
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
  -o smtpd_sasl_type=dovecot
  -o smtpd_sasl_path=private/auth
```

Configure TLS certificate and private key

Copy server certificates in /etc/postfix/certs

```
sudo vi /etc/postfix/main.cf
```

```
#Enable TLS Encryption when Postfix receives incoming emails
smtpd_tls_cert_file=/etc/postfix/certs/vusna.com.pem
smtpd_tls_key_file=/etc/postfix/certs/vusna.com.pem
smtpd_tls_security_level=may
smtpd_tls_loglevel = 1
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache

#Enable TLS Encryption when Postfix sends outgoing emails
smtp_tls_security_level = may
smtp_tls_loglevel = 1
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

#Enforce TLSv1.3 or TLSv1.2
smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtpd_tls_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtp_tls_mandatory_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
smtp_tls_protocols = !SSLv2, !SSLv3, !TLSv1, !TLSv1.1
```

Configure TLS certificate and private key

Restart postfix service

```
sudo systemctl restart postfix
```

Confirm active ports

```
sudo ss -lnpt | grep master
```

Install IMAP and POP3 Server

- Installation

```
sudo apt install dovecot-core dovecot-imapd dovecot-pop3d
```

- Check Dovecot version:

```
dovecot --version
```

- Enabling IMAP/POP3 Protocol

```
sudo vi  /etc/dovecot/dovecot.conf
```

```
protocols = imap pop3
```

Configuring Mailboxes

- Find configuration of mail spool directory

```
postconf mail_spool_directory
```

```
mail_spool_directory = /var/mail
```

```
sudo vi  /etc/dovecot/conf.d/10-mail.conf
```

```
mail_location = maildir:~/Maildir
```

```
mail_privileged_group = mail
```

- Assign the dovecot the mail group

```
sudo adduser dovecot mail
```

Deliver Email to Message Store

- Install the Dovecot LMTP Server.

```
sudo apt install dovecot-lmtpd
```

```
sudo vi  /etc/dovecot/dovecot.conf
```

```
protocols = imap lmtp
```

```
sudo vi /etc/dovecot/conf.d/10-master.conf
```

```
service lmtp {
 unix_listener /var/spool/postfix/private/dovecot-lmtp {
   mode = 0600
   user = postfix
   group = postfix
  }
}
```

```
sudo vi  /etc/postfix/main.cf
```

```
mailbox_transport = lmtp:unix:private/dovecot-lmtp
smtputf8_enable = no
```

Configuring Authentication Mechanism

```
sudo vi  /etc/dovecot/conf.d/10-auth.conf
```

```
disable_plaintext_auth = yes
```

```
auth_username_format = %n
```

```
auth_mechanisms = plain login
```

Configuring SSL/TLS Encryption

```
sudo vi /etc/dovecot/conf.d/10-ssl.conf
```

```
ssl = required
```

```
ssl_cert = </etc/dovecot/private/dovecot.pem
ssl_key = </etc/dovecot/private/dovecot.key
```

```
ssl_prefer_server_ciphers = yes
```

```
ssl_min_protocol = TLSv1.2
```

Configuring SASL Authentication

```
sudo vi  /etc/dovecot/conf.d/10-master.conf
```

```
service auth {
    unix_listener /var/spool/postfix/private/auth {
      mode = 0660
      user = postfix
      group = postfix
    }
}
```

Auto-create Sent and Trash Folder

```
sudo vi /etc/dovecot/conf.d/15-mailboxes.conf
```

```
mailbox Trash {
    auto = create
    special_use = \Trash
 }
```

```
sudo systemctl restart postfix dovecot
```

- Install SASL

```
apt-get install libsasl2-modules
```

Configure Microsoft Outlook

Email : admin@email.vusna.com
Pasword : abcd1234

Imap server : email.vusna.com
Smtp server : email.vusna.com

Imap port : 143 , Auth : STARTTLS, Normal Password
smtp port : 575 , Auth : STARTTLS, Normal Password

Send and receive emails