

Ubuntu® Linux

Mostafa Abd-ElHamid Atwa



MOSTAFA A. HAMID

UBUNTU[®] LINUX

Ubuntu® Linux

1st edition

© 2017 Mostafa A. Hamid & bookboon.com

ISBN 978-87-403-1822-7

CONTENTS

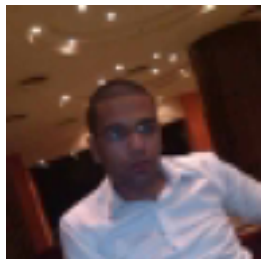
| | | |
|----------|---|-----------|
| | About the Author | 6 |
| | About the Technical Reviewer | 7 |
| 1 | Canonical® LTD. | 9 |
| 1.1 | The Company Behind UBUNTU | 9 |
| 1.2 | UBUNTU® LINUX | 10 |
| 2 | Domain Name System on UBUNTU® | 11 |
| 2.1 | Installing DNS on UBUNTU using Bind9 | 11 |
| 2.2 | DNS Configuration, Maintenance, Management and Automation | 11 |
| 3 | UFW on UBUNTU® | 22 |
| 3.1 | UFW Roles and Rules | 23 |
| 3.2 | Working with Ports | 23 |
| 3.3 | Working with Protocols | 24 |
| 3.4 | IP Masquerading | 26 |
| 3.5 | UFW logging | 26 |

CMO INSPIRED CONFERENCE
25 OCTOBER | DE VERE BEAUMONT ESTATE | OLD WINDSOR UK

Join Over 100 Chief Marketing Officers & Digital Innovators

| | | |
|----------|---|-----------|
| 4 | Mail Server on UBUNTU® | 27 |
| 4.1 | Postfix on UBUNTU® | 27 |
| 4.2 | Installing Open Webmail on UBUNTU® | 38 |
| 5 | Kerberos and Federation Services on UBUNTU® | 41 |
| 6 | Working with Web Servers on UBUNTU® | 50 |
| 6.1 | Installing Glassfish Server on UBUNTU® | 50 |
| 6.2 | Installing and Configuring Apache Web Server® | 50 |
| 6.3 | Installing JBOSS Application Server | 51 |
| 6.4 | Installing and configuring NGINX on UBUNTU® | 52 |
| 7 | Working with Database Servers on UBUNTU® | 53 |
| 7.1 | Installing and Configuring MySQL Server | 53 |
| 7.2 | Installing and Configuring Oracle Database Server 12c Express Edition | 54 |
| | End notes | 55 |

ABOUT THE AUTHOR



Mostafa A. Hamid

Certified CISSP 2013, LPIC 101, IBM RUP, and Certified in PHP, JS, and Java from SUNY Potsdam and The American Chamber of Commerce

Certified in IOTx from MIT

ABOUT THE TECHNICAL REVIEWER

Mohamed Abbarra

Technical Engineer at TechWorld

Bachelor or IS from The University of Hyderabad

E-Mail: mohamed.abarra.in@gmail.com

The company behind UBUNTU® Canonical LTD.

CANONICAL

Sales enquiries

- Americas +1 888 9861322
- Germany +49 800 1838219
- France +33 800914061
- Spain +34 900 833872
- UK and rest of world +44 800 0588704

pr@canonical.com

webmaster@ubuntu.com

Canonical address

If you want to speak to someone right away, you can call or write to Canonical. The London-based office is a good place to start.

- **Canonical Group Limited**
- 5th Floor, Blue Fin Building
- 110 Southwark Street
- SE1 0SU
- London, United Kingdom
- **Main switchboard number:** +44 20 7630 2400

Main fax number: +44 20 7630 2401

1 CANONICAL® LTD.

1.1 THE COMPANY BEHIND UBUNTU

Canonical LTD is an English company for the founder named Mark Shuttleworth from South Africa and he is the main brain of the company.

The company's headquarter is in England.

Some of the products offered by this company are:



UBUNTU and this is our main focus of this book and it is an operating system that has made a revolution in today's technology for desktop use by regular users, developer and cloud devops and by businesses and governments. The reason behind UBUNTU of canonical Ltd's expansion to be #1 LINUX distribution in the world is because it is free of charge and Canonical® Ltd is providing support to it for money.

These were some of the characteristics of the company behind UBUNTU and the other products that they offer are:

JUJU

MAAS

GNU Bazaar

And more...

Canonical LTD website is:

<http://canonical.com>

Up next, we will get to know more details about UBUNTU and the origin of LINUX.

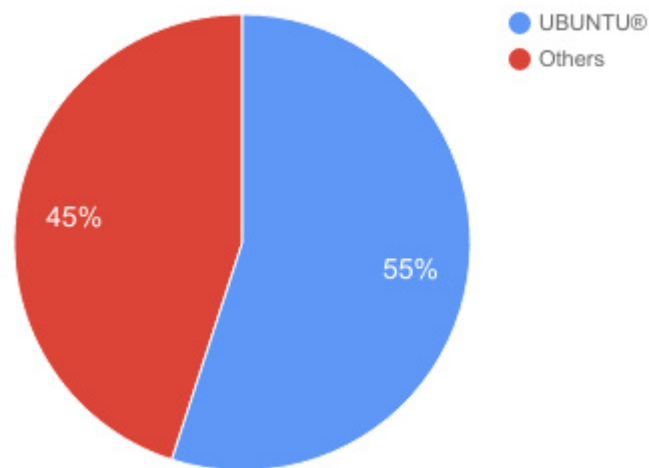
1.2 UBUNTU® LINUX

UBUNTU is an operating system provided by Canonical® Ltd. in 3 main editions:

- Desktops: UBUNTU® Desktop
- Servers: UBUNTU® Server
- Mobiles, Tablets and Phablets: UBUNTU® Touch

Throughout our book, we will be working with commands for UBUNTU Desktop and UBUNTU Server editions and the reason is because this book focuses mainly on system administration and devops operations to be done on UBUNTU to manage organizational infrastructure and work with information technology and devops tasks on these 2 editions.

UBUNTU® VS Other Operating Systems



As we can see in the previous graph that shows the share of UBUNTU vs Other operating systems in the cloud deployments used by developers, system administrators and devops engineers.

2 DOMAIN NAME SYSTEM ON UBUNTU®

2.1 INSTALLING DNS ON UBUNTU USING BIND9

What is DNS and what are the packages required to install it on UBUNTU®?

DNS is a short name standing for Domain Name System, some companies call it domain name server and some other companies call it domain name service.

The domain name system as we will call it throughout this book, is used to control machines grouped together under this domain name.

Policies to manage these computers are grouped together and assigned to all the members under this domain name system.

Also the DNS is a kind of a protection mechanism and a governor that governs all the computers and other devices joining it.

First let us determine the packages needed to install DNS:

1. BIND9
2. DNSUTILS

Now, we will install DNS on ubuntu using 2 commands¹ as follows:²

```
sudo apt-get install bind91
```

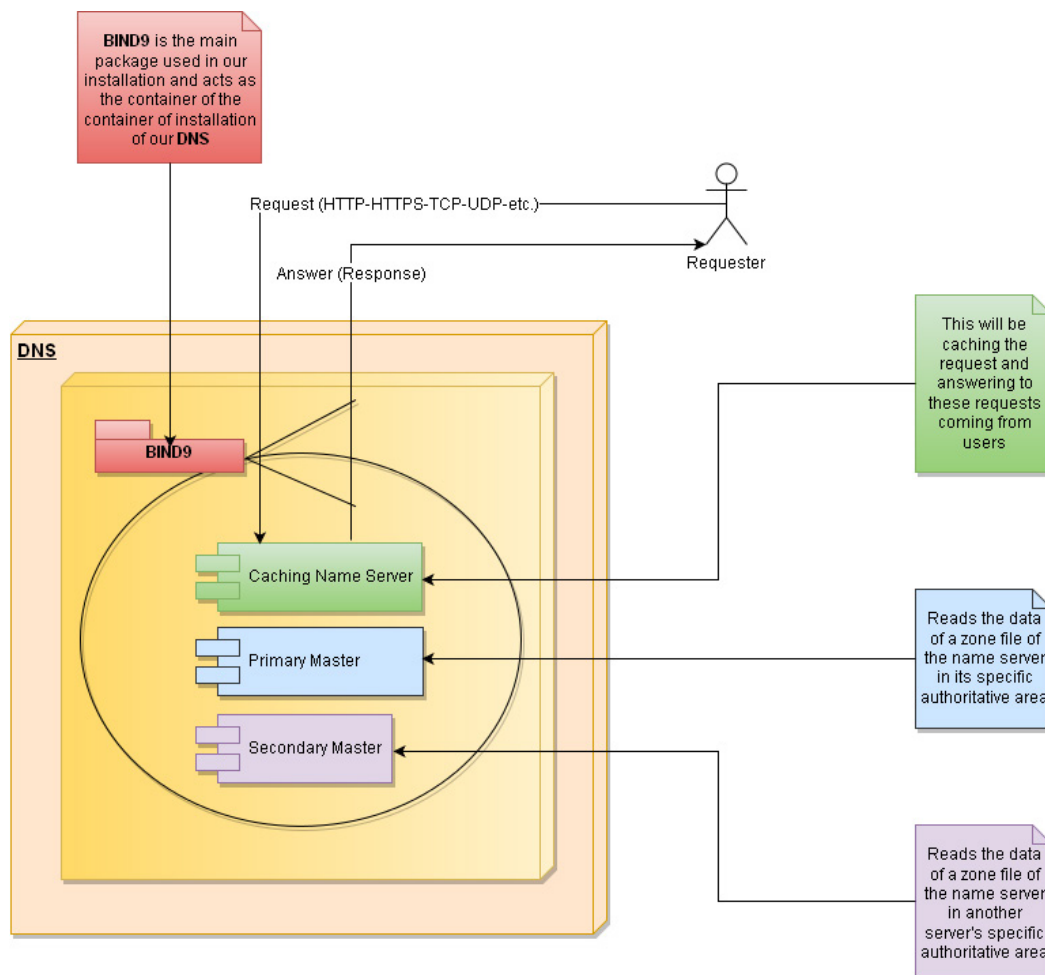
```
sudo apt-get install dnsutils2
```

Lets us go ahead and move on to our configuration stage.

2.2 DNS CONFIGURATION, MAINTENANCE, MANAGEMENT AND AUTOMATION

DNS Configuration can be done using the following settings and some of these settings can be changed according to your environment.

First of all, let us see the following diagram ³which illustrates the structure of the DNS environment and notice that this structure may vary according to the needs of your environment.



Now, we saw a UML 2.5 diagram drawn in <http://draw.io> and let see this in the configuration action, then, we will have to see a more detailed diagram shows the connection between components of each of these configuration option.

Let us get to know the **caching name server**⁴ machine configuration.

Edit a file named `named.conf.options` in the following directory using the following command:

```
sudo nano /etc/bind/named.conf.options
```

Note that the file will open in the terminal in text editing mode and find a section inside the file called forwarders as follows:

By pressing **CTRL + W** on your keyboard and nano will allow you to search for the section called forwarders by typing forwarders and pressing the **return key “enter key”** on your keyboard.

```
forwarders{
    208.67.222.222;
    208.67.220.220;
};
```

Of course you are welcome to put your own IP addresses if you have 2 static IP addresses from your ISP, but anyway these IP addresses are for public use by any person for free available from <http://opendns.org>

After all, we need to fire up the following command to see the installation of our DNS and check if it works or not:

```
dig -x your-ip-address
```

You need to change your-ip-address to your static ip address provided by your ISP or your DHCP IP address provided by your router ex: 192.168.1.2

Let us go ahead and move next to our next configuration option which is the **primary master**⁵ machine and we will apply the following configuration option:

If you will use 1 single machine for the installation of your DNS infrastructure, we will follow this configuration option only.

Now, we will edit a file called named.conf.local and located in the following directory `/etc/bind/named.conf.local` using this command:

```
sudo nano /etc/bind/named.conf.local
```

And the zone entries will be following the template:

```
zone "example.com" {  
    type master;  
    file "/etc/bind/db.example.com";  
};
```

Of course change **example.com** to your own domain name.

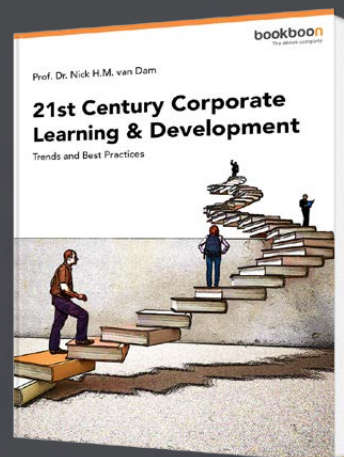
If you are planning to use DDNS (Dynamic Domain Name System) the directory will change to the following:

```
zone "example.com" {  
    type master;  
    file "/var/lib/bind/db.example.com";  
};
```

Free eBook on Learning & Development

By the Chief Learning Officer of McKinsey

[Download Now](#)



The DDNS is useful if you will make the domain dynamically listening to changes from outside of the machine for automation purposes for example if the customers are going to change properties of the DNS using a GUI or a graphical user interface that will control the DDNS remotely.

Now, let us execute the following command to copy a file using cp command from its original place to another as a copy and prepare it for editing as follows:

```
sudo cp /etc/bind/db.local /etc/bind/db.example.com
```

Now, type in the following command to edit the file that we copied in the previous step:

```
$TTL 604800
@ IN SOA example.com. root.example.com. (
        2          ; Serial
        604800     ; Refresh
        86400      ; Retry
        2419200    ; Expire
        604800 )   ; Negative Cache TTL
;
@ IN A 192.168.0.2
;
@ IN NS ns.example.com.
@ IN A 192.168.0.2
@ IN AAAA ::1
ns IN A 192.168.0.2
```

Please be noted to change **example.com** and **root.example.com** to the domain of your choice and the IP address **192.168.0.2** to the IP address of your ISP or the IP address of your router.

Next, we go ahead and move on the next section of our configuration and run one of the following 2 commands to restart the BIND9 service after we made our modifications to our configuration files:

```
sudo systemctl restart bind9.service
or
sudo service bind9 restart
```

Next, we move on our next configuration setting by adding reverse name zone to our named.conf.local in the directory /etc/bind/named.conf.local

By running the following command, we will edit the requested file to modify it as follows:

```
sudo nano /etc/bind/named.conf.local
```

And add a zone record just the the following:

```
zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
};
```

And of course you will need to replace the 0.168.192 to the first 3 octets of your IP address in reverse format and the last 192 in the entry is also according to your IP address.

Let us go ahead and continue with our installation configuration option and we will create a file called db.192 or db.(first-octet-of-your-IP) which will be identical to the file in our last directory that we put in the previous zone entry as follows:

```
sudo nano /etc/bind/db.192
or
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Next, we go ahead and move on to our file contents and we will put inside the file the following domain record entry:

```
$TTL 604800
@      IN      SOA      ns.example.com. root.example.com. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@      IN      NS       ns.
10     IN      PTR      ns.example.com.
```

Next, we move on to the next section we press CTRL + X and save the file with the same name it is prompting with.

Then, we move on to our next configuration option which is secondary master after restarting BIND9.

To restart BIND9, we go ahead and perform the following command:

```
sudo service bind9 restart
or
sudo systemctl restart bind9.service
```

Our third configuration option will be the secondary master.

First, we will add two entries to our named.conf.local file into the directory /etc/bind/named.conf.local as follows:

```
sudo nano /etc/bind/named.conf.local
```

And then we will add the following two zone entries into our file as follows:

```
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
    allow-transfer { 192.168.0.3; };
};

zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
    allow-transfer { 192.168.0.3; };
};
```

Now, we configured the 2 address zones, one in reverse mode and the other one is the regular zone entry and notice that we have added “allow-transfer” directive to allow remote configuration changes to transfer the zone.

Then, we will go ahead and configure our primary master on the IP: 192.168.0.2 with the following changes:

Type in the following command to edit our file:

```
sudo nano /etc/bind/named.conf.local
```

Then change the zones entries to the following:

```
zone "example.com" {
    type slave;
    file "db.example.com";
    masters { 192.168.0.2; };
};

zone "0.168.192.in-addr.arpa" {
    type slave;
    file "db.192";
    masters { 192.168.0.2; };
};
```

Now, let us go ahead and restart BIND9 service using one of the following 2 commands as follows:

```
sudo service bind9 restart
or
sudo systemctl restart bind9.service
```

And now one last [Optional] thing is that if you want the primary master to notify the secondary master with any change, we will need to add the following lines into our named.conf.local of our primary master machine as follows:

Edit the file using the following command:

```
sudo nano
```

Then change the contents of the file to the following:

```
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
    allow-transfer { 192.168.0.2; };
    also-notify { 192.168.0.2; };
};

zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
    allow-transfer { 192.168.0.2; };
    also-notify { 192.168.0.2; };
};
```

Note the changes we made to this file by adding two additional directives:

`allow-transfer`: (allowing the transfer to happen to a specific IP address).

`also-notify`: (allowing the notifications to be sent to a specific IP address).

And finally, we need to edit the file named `resolv.conf` to the following data and let us edit the file using the command:

```
sudo nano /etc/resolv.conf
```

Then, we will add the following content and replace the previously assigned content as follows:

```
nameserver 192.168.0.2
nameserver 192.168.0.3
```

This configuration should be done on each machine with its own IP address.

If you did not make automation with a cronjob to handle the task of changing the content each and every time the machine restart to edit the content of the file `/etc/resolv.conf`. You will need to modify the content manually each and every time the machine restarts or the operating system restarts.

Next, we will go ahead and create an automation script that will run at startup each time the user is logged in or the computer starts.

First, we will create an initialization script called `modify-resolv.sh` into the directory `/etc/init.d/modify-resolv.sh` using the following command:

```
sudo nano /etc/init.d/modify-resolv.sh
```

After that we will add the following contents to our file

```
sed -i -e "nameserver 192.168.0.2\nnameserver  
192.168.0.3" /etc/resolv.conf
```

This way, we will automate the process of changing the `/etc/resolv.conf` file using the crontab solution as follows:

```
chmod ugo+x /etc/init.d/modify-resolv.sh  
update-rc.d modify-resolv.sh defaults
```

After that, if you restart your machine or logged out of the device, you will find that these scripts will automate the process of the modification of `resolv.conf` file which points to your DNS.

Now, some of the troubleshooting operations and check to our installation and see if everything is okay.

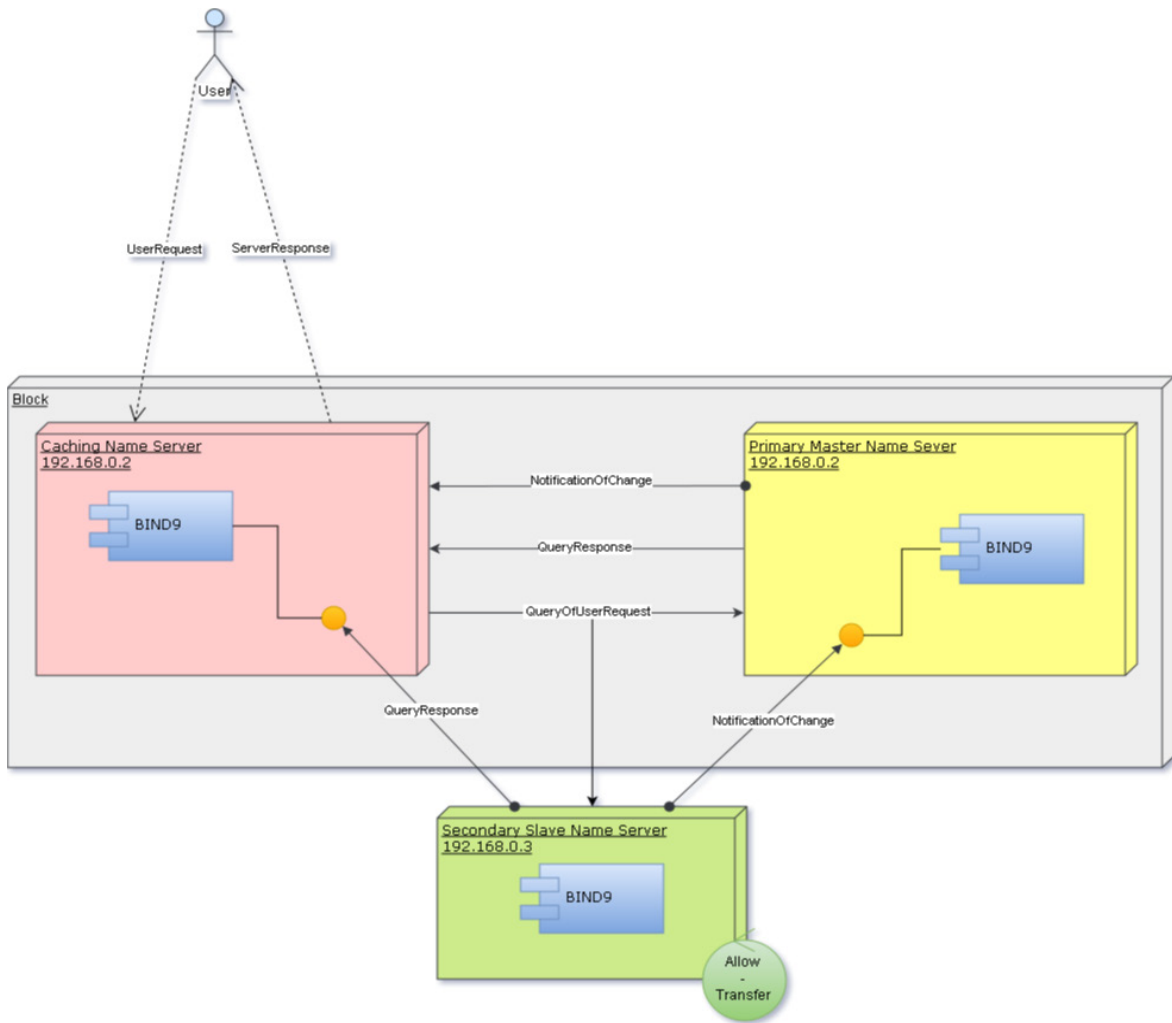
The command is:

```
named-checkzone example.com /etc/bind/db.example.com
```

If you see okay, in the output, and propagated to the name system, then you have done a great DNS installation and everything works just fine.

Else, you will need to re-configure the files as we did in the previous steps to achieve a working DNS installation.

Finally, let us see what we did into a diagram with details of all the operations:



3 UFW ON UBUNTU®

The **UFW** is the short acronym which stands for Uncomplicated Firewall. The firewall is a kind of a program that enables users of having access to a specific IP or deny access to it and also enable users have access to specific IP using specific port or deny access to IP using specific port.

Let us get to recognize the port numbers and what are its functionalities. The port numbers are numbers associated with IP addresses allowing traffic to pass through it and some of these famous numbers and its uses are:

| Port | Common Usage |
|------------|---|
| 80 8080 | are the most commonly used port number for HTTP requests and responses |
| 443 | is the most commonly used port number for HTTPs requests and responses |
| 22 | is the most commonly used port number for SSH requests and responses |
| 23 | is the most commonly used port number for Telnet requests and responses |
| 25 | is the most commonly used port number for SMTP requests and responses |
| 20 | is the most commonly used port number for FTP requests and responses |

Now, let us go ahead and move next to our installation of UFW on UBUNTU®. The installation requires the following packages to be installed:

```
sudo apt-get install ufw
```

Note that UBUNTU® is equipped already with UFW installed and configured with the most secured options.

Let us enable the firewall by using the command:

```
sudo ufw enable
```

The previous command will be enabling the firewall to start with its default configuration.

Let us go ahead and start doing some initial configuration.

The following command will show you the status of your current UFW service:

```
sudo ufw status verbose
```

It will show you if the firewall is enabled or not, logging or not, default incoming and outgoing rules, and if it is skipping user profile or not.

The following command will show you the list of exceptions to the rules applied to your current UFW service:

```
sudo ufw show raw
```

The next section will be about applying UFW roles and rules.

3.1 UFW ROLES AND RULES

UFW accepting commands to be acting as Roles and Rules.

Rules in firewall are allow or deny and the default in nearly most of the firewalls allow the world is deny that will be denying access to specific resource or destination.

As we saw in our previous table of port numbers, we will be using it in making some rules to the firewall.

3.2 WORKING WITH PORTS

Let us go ahead and enable and disable incoming traffic on specific port numbers as follows:

```
sudo ufw allow 53
sudo ufw deny 22
sudo ufw allow 80
sudo ufw deny 8080
sudo ufw allow 53/tcp
```

Now, we have made a some Roles containing Rules governing incoming packets through some ports like 53, 22, 80, 8080, 52/tcp.

Let us see each of these commands in details:

The first command allows packets to be incoming on port 53

The second command denies packets to be incoming on port 22

The third command allows packets to be incoming on port 80

The fourth command denies packet to be incoming on port 8080

3.3 WORKING WITH PROTOCOLS

The fifth command allows all packets to be incoming on port 53 and using TCP protocol because there are some other protocols that can be used like UDP and can be also used in this command this way:

```
sudo ufw allow 53/udp
```

The next command will delete existing rule as follows:

```
sudo ufw delete deny 80/tcp
```



Discover the truth at www.deloitte.ca/careers

Deloitte.

© Deloitte & Touche LLP and affiliated entities.

Now, UFW also can allow and deny traffic using service name. If you remember the commands used to restart a service by service name like BIND9 for example, and this is how we will use the same principal in UFW commands as follows:

```
sudo ufw allow bind9
sudo ufw deny bind9
sudo ufw delete allow bind9
sudo ufw delete deny bind9
```

Now, we did 4 commands to allow and deny packets to be incoming by service name and deleted the rule by service name.

Some advanced UFW features are:

```
sudo ufw allow from 192.168.0.4 to any port 22
sudo ufw allow from 192.168.1.0/24
sudo ufw allow from 192.168.0.4 to any port 22
sudo ufw allow from 192.168.0.4 to any port 22 proto tcp
sudo ufw status numbered
sudo ufw delete 1
sudo ufw insert 1 deny from 192.168.0.5
```

Let us see each and every command of these:

- The first command will allow incoming packets from host with IP address 192.168.0.4 using port 22
- The second command will allow from 192.168.1.0 with CIDR (Classless Interdomain Routing) subnet mask number /24
- The third command will allow from 192.168.0.4 on port 22
- The fourth command will allow from 192.168.0.4 on port 22 and using protocol TCP
- The fifth command will get the status of the current rules and each rule will have a number and these numbers are used the following 2 command
- The sixth command will delete the rule number 1 appeared in the previous command
- The seventh command will insert a rule and assign number 1 to it if it not exists or will overwrite the existing number 1 rule with deny from 192.168.0.5

One last thing in our advanced topics will be IP masquerading.

3.4 IP MASQUERADING

If you will enable IP masquerading on your machine, let us examine the following procedures:

Edit a file named `sysctl.conf` in the following directory: `/etc/sysctl.conf` using the following command:

```
nano /etc/sysctl.conf
```

Then, we will uncomment the following lines of code and this means to remove the `#` or `;` preceding the specified line:

```
net.ipv4.ip_forward=1
net.ipv6.conf.default.forwarding=1
```

The previous 2 lines of configuration options will enable the IPv4 and IPv6 forwarding.

Then execute the following 2 commands:

```
sudo sysctl -p
sudo iptables -t nat -A POSTROUTING -s
192.168.0.4/16 -o eth0 -j MASQUERADE
```

Let us assume that your machine will have the IP address of `192.168.0.4` and the ethernet adapter you are using is named: `eth0` on the UBUNTU® system and if you are using wireless interface, then the default wireless interface name provided by the system on UBUNTU® is: `wlan0`

3.5 UFW LOGGING

To enable and disable logging into your UFW instance, we use the following 2 commands:

```
sudo ufw logging on
sudo ufw logging off
```

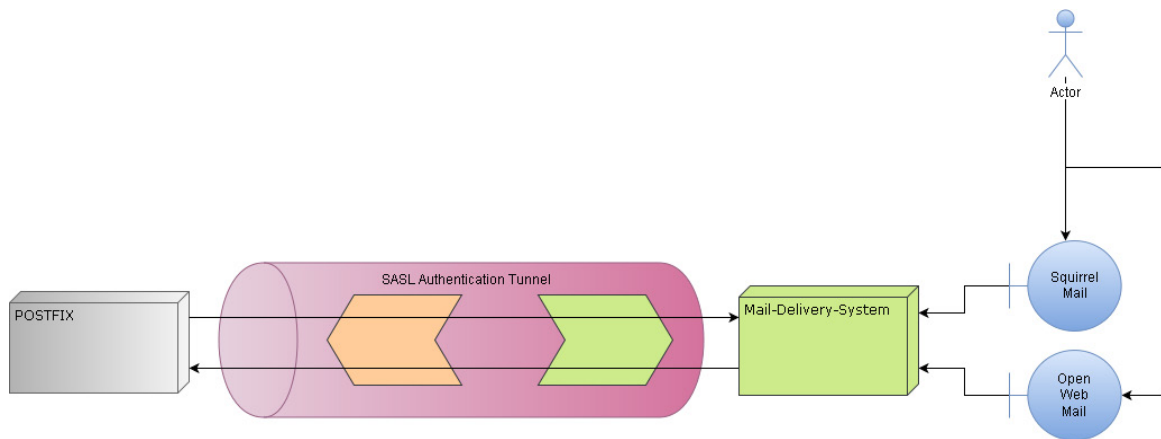
4 MAIL SERVER ON UBUNTU®

4.1 POSTFIX ON UBUNTU®

The mail server on UBUNTU® requires packages to install and these packages are:

```
POSTFIX
DOVECOT-CORE
MAIL-STACK-DELIVERY
```

Now, let us go ahead and start our installation process after taking a look at the following diagram which will illustrate how these components operate all together:



Now, we will go ahead and move on to POSFIX installation:

```
sudo apt install postfix
sudo dpkg-reconfigure postfix
```

Then, we will use the following configuration options:

```
Internet Site
mail.example.com
manon
mail.example.com, localhost.localdomain, localhost
No
127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.2/24
0
+
all
```

Of course you will need to replace 192.168.0.2 with your device IP address and /24 with your CIDR network mask and let us go ahead and move on to the next configuration setting

We will need to run the following command to tell POSTFIX that our mail directory will be Maildir as follows:

```
sudo postconf -e 'home_mailbox = Maildir/'
```

Then, we will configure POSTFIX to use SASL authentication as follows:

```
sudo postconf -e 'smtpd_sasl_type = dovecot'
sudo postconf -e 'smtpd_sasl_path = private/auth'
sudo postconf -e 'smtpd_sasl_local_domain ='
sudo postconf -e 'smtpd_sasl_security_options = noanonymous'
sudo postconf -e 'broken_sasl_auth_clients = yes'
sudo postconf -e 'smtpd_sasl_auth_enable = yes'
sudo postconf -e 'smtpd_recipient_restrictions = \
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
openssl genrsa -des3 -out server.key 2048
```

Then enter your appropriate information required in the prompts.

After that, let us continue with our next step:

```
openssl x509 -req -days 365 -in server.  
csr -signkey server.key -out server.crt  
sudo cp server.crt /etc/ssl/certs  
sudo cp server.key /etc/ssl/private  
sudo mkdir /etc/ssl/CA  
sudo mkdir /etc/ssl/newcerts  
sudo sh -c "echo '01' > /etc/ssl/CA/serial"  
sudo touch /etc/ssl/CA/index.txt
```

Then, we will move on to the next section of our configuration which will be configuring OPENSSL package already installed with our new certificates as follows:

```
sudo nano /etc/ssl/openssl.cnf
```

After that, we will change the configuration settings into this file as follows:

```
dir           = /etc/ssl  
database      = $dir/CA/index.txt  
certificate   = $dir/certs/cacert.pem  
serial        = $dir/CA/serial  
private_key   = $dir/private/cakey.pem
```

Now, we will create a self signed root certificate as follows:

```
openssl req -new -x509 -extensions v3_ca -keyout  
cakey.pem -out cacert.pem -days 3650
```

Then, we will move on to the next section which will be the root certificate and the key file as follows:

```
sudo mv cakey.pem /etc/ssl/private/  
sudo mv cacert.pem /etc/ssl/certs/
```

Finally, we will create a self signed certificate signing request as follows:

```
sudo openssl ca -in server.csr -config /etc/ssl/openssl.cnf
```

After that, we will continue working on postfix configuration to use the certificates that we have used as follows:

```
sudo postconf -e 'smtp_tls_security_level = may'
sudo postconf -e 'smtpd_tls_security_level = may'
sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
sudo postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
sudo postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
sudo postconf -e 'smtpd_tls_loglevel = 1'
sudo postconf -e 'smtpd_tls_received_header = yes'
sudo postconf -e 'myhostname = mail.example.com'
sudo postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'
sudo systemctl restart postfix.service
OR
sudo service postfix restart
```

Now, we will install our second package called dovecot for SASL authentication as follows:

```
sudo apt install dovecot-core
```

Then, we will configure the dovecot-core package as follows:

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

After that, we will need to change the contents of the configuration file to fit the contents as follows:

```
service auth {
  unix_listener auth-userdb {
    #mode = 0600
    #user =
    #group =
  }
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
}
```

For outlook clients, we will need to move on to edit the following file:

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

We will add the following line into our file as follows:

```
auth_mechanisms = plain login
```

Now, we will install our last package which is the mail-stack-delivery as follows:

```
sudo apt install mail-stack-delivery
```

After that, configure the POSTFIX configuration file as follows:

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

Then, we will change the configuration settings as follows:

```
smtpd_tls_cert_file = /etc/ssl/certs/ssl-mail.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-mail.key
```

Of course you will need to change the certificates directory and file names to your appropriate files and directory inside your system as configured exactly in creating self-signed certificate section.

Now, we will test our environment as follows using telnet command to connect to our host on the specified mail port:

```
telnet mail.example.com 25
```

At the end, you will need to get to type the following command and tell ehlo to our server using the domain name as follows:

```
ehlo mail.example.com
```

We have now finished our installation and configuration procedures of our mail server and we have 2 remaining things, is to set up a comprehensive user interface and get the logging data, and troubleshooting the mail server to be working just fine.

We will complete our configuration security protocols enablement for security reasons because most of the problems that comes into our installation is because of security threats that arise during working on something essential.

To install SMTPs, we will edit the file `/etc/postfix/master.cf` as follows:

```
sudo nano /etc/postfix/master.cf
```

Then we will make the following configuration changes:

```
smtps inet n - - - smtpd
-o smtpd_tls_wrappermode=yes
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
-o milter_macro_daemon_name=ORIGINATING
```

Mail server logging configuration should be done the following way:

```
sudo postconf -e 'smtpd_tls_loglevel = 4'
```

What if you do not receive a mail from a specific domain?

```
sudo postconf -e 'debug_peer_list = domain-with-problems.com'
```

Of course after each configuration change, you must restart the services or service related to the changes of the configuration you made.

Now, let us install the following contents for the dovecot package as follows:

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

We will modify to the configuration of the dovecot package as follows:

```
sudo nano /etc/dovecot/dovecot.conf
```

Let us add the following configuration option to it:

```
protocols = pop3 pop3s imap imaps
pop3_uidl_format = %08Xu%08Xv
mail_location = maildir:/home/%u/Maildir
# not ideal if your client is outlook
disable_plaintext_auth = no
ssl = yes
ssl_cert_file = /etc/ssl/certs/cert-file-name.pem
ssl_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
listen = *
protocol imap {
    listen = *:143
    ssl_listen = *:993
    login_greeting_capability = yes
    imap_client_workarounds = tb-extra-mailbox-sep
}
protocol pop3 {
    listen = *:110
    ssl_listen = *:995
}
```

be > your degree

Bring your talent and passion to a global organization at the forefront of business, technology and innovation. Discover how great you can be.

Visit accenture.com/bookboon

Be greater than.
consulting | technology | outsourcing

accenture
High performance. Delivered.

© 2013 Accenture. All rights reserved.

The previous configuration is about configuring protocols to be used by dovecot which will be pop3, pop3s, imap and imaps, then we added the format of the pop3 and mail location which will be ~/Maildir

Let us now set up the Maildir as follows:

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

Then we will configure it with the following:

```
home_mailbox = Maildir/
```

Now, we will update all the users with Maildir to be added to their UBUNTU Linux Skeleton as follows:

```
sudo maildirmake.dovecot /etc/skel/Maildir
sudo maildirmake.dovecot /etc/skel/Maildir/.Drafts
sudo maildirmake.dovecot /etc/skel/Maildir/.Sent
sudo maildirmake.dovecot /etc/skel/Maildir/.Trash
sudo maildirmake.dovecot /etc/skel/Maildir/.Templates
sudo cp -r /etc/skel/Maildir /home/myuser/
sudo chown -R usernameismanon:rootisgroup /home/myuser/Maildir
sudo chmod -R 700 /home/myuser/Maildir
sudo service dovecot start
```

To test the protocols, we will run the following commands:

```
telnet localhost pop3
telnet localhost imap2
openssl s_client -connect mail.domain.ext:993
A1 LOGIN username password
A2 LIST "" "*"
A3 EXAMINE INBOX
```

Do you want to install an antivirus scanner for your mail server and scan every mail?

Today mailing services are full of spams and virus attacks and it is a package called Amavis-New and this is an antivirus that will be scanning infected e-mails and exclude them automatically.

Let us go through the installation and package configuration of the e-mail antivirus as follows:

To install amavis-new, we will need to run the following commands:

```
sudo apt-get install amavisd-new spamassassin clamav-daemon
sudo apt-get install libnet-dns-perl libmail-spf-perl pyzor razor
sudo apt-get install arj bzip2 cabextract cpio file gzip
lha nomarch pax rar unrar unzip unzoo zip zoo
```

Next, we will configure the following contents into our postfix as follows:

```
sudo postconf -e "content_filter = smtp-amavis:[127.0.0.1]:10024"
```

Now, we will edit the following file `/etc/postfix/master.cf` as follows:

```
sudo nano /etc/postfix/master.cf
```

Then add the following at the end of the file:

```
smtp-amavis unix - - - - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes
-o max_use=20

127.0.0.1:10025 inet n - - - - smtpd
-o content_filter=
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_delay_reject=no
-o smtpd_client_restrictions=permit_mynetworks,reject
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o smtpd_data_restrictions=reject_unauth_pipelining
-o smtpd_end_of_data_restrictions=
-o mynetworks=127.0.0.0/8
-o smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
-o smtpd_client_connection_count_limit=0
-o smtpd_client_connection_rate_limit=0
-o receive_override_options=no_header_body_
checks,no_unknown_recipient_checks
```

Now, restart our postfix service and execute the following commands:

```
sudo /etc/init.d/postfix reload
chmod -R 775 /var/lib/amavis/tmp
```

Now, we will continue to our GUI installation using squirrelmail and its components.

4.2 INSTALLING OPEN WEBMAIL ON UBUNTU®

Installing open webmail requires the following packages and programming languages to be installed on your system:

- perl
- libauthen-pam-perl
- libconvert-asn1-perl
- libmd5-perl
- libnet-ldap-perl
- perl-suid
- wwwconfig-common
- libpg-perl

Now, let us install these packages as follows:

```
sudo apt-get install perl libauthen-pam-perl libconvert-asn1-perl  
libmd5-perl libnet-ldap-perl perl-suid wwwconfig-common libpg-perl
```

Now, let us go ahead and move on to our configuration of these packages to integrate them with our POSTFIX package:

```
sudo rm /var/www/openwebmail/index.html  
sudo ln -s /var/www/openwebmail/redirect.  
html /var/www/openwebmail/index.html
```

Now, browse <http://localhost/openwebmail>

And you will find the mail up and running.

After that if you want to login use your username and password or any of the system's username and password and to add a user add a user to the system or if you want to delete a user you will need to delete the user from the system.

One more thing left is the alternative to openwebmail which is squirrelmail and let us see how it is installed and configured.

To install squirrelmail, first let us install the lamp-server package that contains the PHP, MySQL, and Apache web server as follows:

```
sudo apt-get install tasksel  
sudo tasksel install lamp-server
```

The provide it with mysql root password when prompted and record the password into a piece of paper or another reminder just not to forget it because we will need it into other configurations and installations of other packages.

Let us install the squirrelmail as follows:

```
sudo apt-get install squirrelmail
```

After that let us configure our squirrelmail as follows:

```
sudo squirrelmail-configure
```

Into option 2, let us make the following changes:

```
Update IMAP Settings : localhost:143  
Update SMTP Settings : localhost:25
```

And then, into option 4 allow the following setting number 11:

```
Allow server-side sorting
```

Execute the following commands to integrate squirrelmail with Apache web server as follows:

```
sudo cp /etc/squirrelmail/apache.conf /etc/  
apache2/sites-available/squirrelmail  
sudo ln -s /etc/apache2/sites-available/squirrelmail /  
etc/apache2/sites-enabled/squirrelmail  
OR  
sudo a2ensite squirrelmail  
sudo service apache2 restart
```

Now, browse the following URL and you will find it working:

<http://localhost/squirrelmail>

And this way, we finished our installations, configurations and troubleshooting for all the mail server components.

5 KERBEROS AND FEDERATION SERVICES ON UBUNTU®

Let us go...!!!

This part is a kind of heavy metal song that you are listening to and makes noise in the head a little bit but definitely, it will be interesting.

First, let us get to recognize what are Kerberos, and Federation Services.

Kerberos and Federation Services are kinds of protection to our domain that we installed using bind9.

The packages required to install Kerberos and Federation Services on UBUNTU are:

- slapd
- ldap-utils
- krb5-kdc-ldap
- krb5-kdc
- krb5-admin-server
- ldapscripts

We will go ahead and install OpenLDAP first as follows:

```
sudo apt install slapd ldap-utils
```

We will now continue with creating a new LDIF file called `/etc/ldap/slapd.d/add_content.ldif` using the following command:

```
sudo nano /etc/ldap/slapd.d/add_content.ldif
```

And add the following content into it:

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

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

dn: cn=miners,ou=Groups,dc=example,dc=com
objectClass: posixGroup
cn: miners
gidNumber: 5000

dn: uid=manon,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: manon
sn: Niazi
givenName: Manon
cn: Manon Niazi
displayName: Manon Niazi
uidNumber: 10000
gidNumber: 5000
userPassword: manonldap
gecos: Manon Niazi
loginShell: /bin/bash
homeDirectory: /home/manon
```

The content must be modified to the conditions of your environment such as the usernames and passwords, the directories, etc.

Then, we will execute the following command:

```
ldapadd -x -D cn=admin,dc=example,dc=com -W -f add_content.ldif
```

When prompted, enter the password you entered in our add_content.ldif

After that, execute the following command to search for the content you added:

```
ldapsearch -x -LLL -b dc=example,dc=com 'uid=manon' cn gidNumber
```

Next, we will create a file named `uid_index.ldif` into the following directory `/etc/ldap/slapd.d/uid_index.ldif` using the command:

```
sudo nano /etc/ldap/slapd.d/uid_index.ldif
```

And we will add the following content to our file as follows:

```
dn: olcDatabase={1}hdb,cn=config
add: olcDbIndex
olcDbIndex: uid eq,pres,sub
```

After that, execute the command:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f uid_index.ldif
```

Finally, you can also search for the entry we made in the previous step using the following command:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \
cn=config '(olcDatabase={1}hdb)' olcDbIndex
```

Now, into our `/etc/ldap/schema` directory, let us create a file named `schema_convert.conf` using the command:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/ldapns.schema
include /etc/ldap/schema/pmi.schema
```



What if you could build your future and create the future?

The innovation accelerator

One generation's transformation is the next's status quo. In the near future, people may soon think it's strange that devices ever had to be "plugged in." To obtain that status, there needs to be "The Shift".

.....Alcatel·Lucent 

www.alcatel-lucent.com/careers

Next, create a directory called `ldif_output` using the following command:

```
slapcat -f schema_convert.conf -F ldif_
output -n 0 | grep corba,cn=schema
```

And convert the output of the previous command using the following command:

```
slapcat -f schema_convert.conf -F ldif_output -n0 -H \
ldap:///cn={1}corba,cn=schema,cn=config -l cn=corba.ldif
```

Change it to the following:

```
dn: cn=corba,cn=schema,cn=config
```

Then remove the following entries:

```
structuralObjectClass: olcSchemaConfigentryUUID: 52109a02-66ab-
1030-8be2-bbf166230478creatorsName: cn=configcreateTimestamp:
20110829165435ZentryCSN: 20110829165435.935248Z#000000#000#00
0000modifiersName: cn=configmodifyTimestamp: 20110829165435Z
```

After that, we will continue with the following command to add a new entry:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f cn\=corba.ldif
```

We will go through user and group management of the LDAP as follows:

```
sudo apt-get install ldapscripts
```

Then edit the file `/etc/ldapscripts/ldapscripts.conf` as follows:

```
sudo nano /etc/ldapscripts/ldapscripts.conf
```

Add the following entries into your file or edit the entries if already existing:

```
SERVER=localhost
BINDDN='cn=admin,dc=example,dc=com'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
SUFFIX='dc=example,dc=com'
GSUFFIX='ou=Groups'
USUFFIX='ou=People'
MSUFFIX='ou=Computers'
GIDSTART=10000
UIDSTART=10000
MIDSTART=10000
```

Of course you will need to change the entries to fit your environment.

Then, we execute the following commands to modify the ldap passwd file:

```
sudo sh -c "echo -n 'manon' > /etc/ldapscripts/ldapscripts.passwd"
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

After that, we will create a user, delete the user, using the following commands:

```
sudo ldapadduser manon example
sudo ldapsetpasswd manon
sudo ldapdeleteuser manon
```

And, we will add, and remove groups as follows:

```
sudo ldapaddgroup niazi
sudo ldapdeletegroup niazi
```

After that, we will add and delete user to and from a group using the commands:

```
sudo ldapaddusertogroup manon niazi
sudo ldapdeleteuserfromgroup manon niazi
```

We will now load the new schema with the following command:

```
ldapadd -x -D cn=admin,cn=config -W -f /tmp/cn\=kerberos.ldif
```

Execute the following command to add an entry to your file as follows:

```
ldapmodify -x -D cn=admin,cn=config -W
```

Then add the contents as follows:

```
dn: olcDatabase={1}hdb,cn=config  
add: olcDbIndex  
olcDbIndex: krbPrincipalName eq,pres,sub
```

Then at the end, update the access control list using the command:

```
ldapmodify -x -D cn=admin,cn=config -W
```

Now, let us install the following packages to create our primary KDC:

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

Now, edit the following file `/etc/krb5.conf` and add the content after the command to it:

```
sudo nano /etc/krb5.conf
```

Then, edit the following contents of the file:

```
[libdefaults]
    default_realm = EXAMPLE.COM
[realms]
    EXAMPLE.COM = {
        kdc = kdc01.example.com
        kdc = kdc02.example.com
        admin_server = kdc01.example.com
        admin_server = kdc02.example.com
        default_domain = example.com
        database_module = openldap_ldapconf
    }
[domain_realm]
    .example.com = EXAMPLE.COM
[dbdefaults]
    ldap_kerberos_container_dn = dc=example,dc=com
[dbmodules]
    openldap_ldapconf = {
        db_library = kldap
        ldap_kdc_dn = "cn=admin,dc=example,dc=com"
        ldap_kadmind_dn = "cn=admin,dc=example,dc=com"
        ldap_service_password_file = /etc/krb5kdc/service.keyfile
        ldap_servers = ldaps://ldap01.example.
        com ldaps://ldap02.example.com
        ldap_conns_per_server = 5
    }
```

Now, create a realm using the following command and notice that this realm we will use while installing our web server another chapter, so please record the realm name and data into a piece of paper for use later in installing web server in the installing web server chapter.

```
sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com create -subtrees \
dc=example,dc=com -r EXAMPLE.COM -s -H ldap://ldap01.example.com
```

Then we will need to create something called a stash using the following command:

```
sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com stashsrvpw -f \
/etc/krb5kdc/service.keyfile cn=admin,dc=example,dc=com
```

Then, copy the CA file from the LDAP server using the following command:

```
scp ldap01:/etc/ssl/certs/cacert.pem . sudo
cp cacert.pem /etc/ssl/certs
```

Now, edit the ldap.conf file using the command:

```
sudo nano /etc/ldap/ldap.conf
```

Then, add these content to it:

```
TLS_CACERT /etc/ssl/certs/cacert.pem
```

Now, run the following command to add kerberos principals:

```
sudo kadmin.local
```

Now, restart all the related services:

```
sudo systemctl restart slapd.service
sudo systemctl start krb5-kdc.service
```

6 WORKING WITH WEB SERVERS ON UBUNTU®

Web server is a kind of a container that contains a web application programmed by software and web application programmer and the web application performs all of its functionality using the web server.

6.1 INSTALLING GLASSFISH SERVER ON UBUNTU®

First, let us get to install glassfish server as one of the most popular web application servers for java programming language as follows:

```
sudo apt-get update
sudo apt-get install default-jdk
wget http://download.java.net/glassfish/4.1.1/release/glassfish-4.1.1.zip
unzip glassfish-4.1.1.zip
glassfish4/bin/asadmin start-domain
```

Configuring Glassfish Server:

1. Open the URL <http://localhost:4848>
2. Then browse to the left side panel and select server-config
3. After that, go to Network Listeners, then
4. Set http-listener-1 to port 80 for default http listening

After that, we will move on to the next installation which is:

6.2 INSTALLING AND CONFIGURING APACHE WEB SERVER®

Apache web server is used as a stack bundled with PHP, and MySQL and PHP is the programming language and MySQL is the database engine.

Apache web server installation is as follows:

```
sudo apt-get update
sudo apt-get install tasksel
sudo tasksel install lamp-server
```

The configuration of apache on ubuntu using the following command:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Then, we will go to the section **AllowOverride** and set it to **all** as follows:

```
AllowOverride all  
OR  
AllowOverride All
```

This option will allow override if your application in PHP requires MVC pattern.

Then we will create a file called .htaccess where index.php is located of the application that you will deploy into your application server as follows:

```
sudo nano /var/www/html/.htaccess
```

Then we will put the contents if the application only redirects all routes to the index.php file as follows:

```
Options +FollowSymLinks  
RewriteEngine On  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteRule ^ index.php [L]
```

Now, we will work with our next installation which will be:

6.3 INSTALLING JBOSS APPLICATION SERVER

JBoss Application Server is another application server for hosting Java web applications. Installing JBoss requires the following commands:

```
wget http://download.jboss.org/wildfly/10.0.0.Final/
wildfly-10.0.0.Final.tar.gz
tar -xvzf wildfly-10.0.0.Final.tar.gz
mv wildfly-10.0.0.Final wildfly
chmod -R 755 wildfly
cd bin/
./add-user.sh
#this will add a user
./standalone.sh
#this will run the server
```

Now, let us go ahead and open the following URL:

<http://127.0.0.1:9990>

This way, we have finished installing our glassfish server and configured it by adding a new user.

6.4 INSTALLING AND CONFIGURING NGINX ON UBUNTU®

NGINX is a web server used for hosting PHP web applications just like Apache that we installed before.

Now, we will perform the command:

```
sudo sh -c "echo 'deb http://nginx.org/packages/mainline/ubuntu/
'$(lsb_release -cs)' nginx' > /etc/apt/sources.list.d/Nginx.list"
```

This command added a list into our sources.list.d into our **UBUNTU®** lists that it uses to get updates of the current packages.

```
sudo apt-get update
sudo apt-get install nginx
```

This way, we have installed NGINX as our web application server and please do not make an interfer on your machine by installing more than 1 web server per OS installation because most of the web servers use the same protocols and only one protocol can save one server at a time or you will need to re-configure each web server to use different protocol.

7 WORKING WITH DATABASE SERVERS ON UBUNTU®

Database servers are used mainly to store databases and keep data into it in collaboration with application or web application servers.

The most famous database servers are:
MySQL and Oracle DB 12c

Let us go ahead and install each one of these and we will start with:

7.1 INSTALLING AND CONFIGURING MYSQL SERVER

To install MySQL server, you will need to run the following command:

```
sudo apt-get install mysql-server
```

Some other people install maria db package and this will be done using the following command:

```
sudo apt-get install maria-db
```

Then for security reasons, you may run the following command to get your server up and running securely:

```
mysql_secure_install
```

The last thing, we need to install is oracle 12c database server express edition.

7.2 INSTALLING AND CONFIGURING ORACLE DATABASE SERVER 12C EXPRESS EDITION

Oracle Database Server 12c Express Edition is one of the mostly highly ranked database server in 2017 and it acts exactly as the data container that holds applications or web applications data.

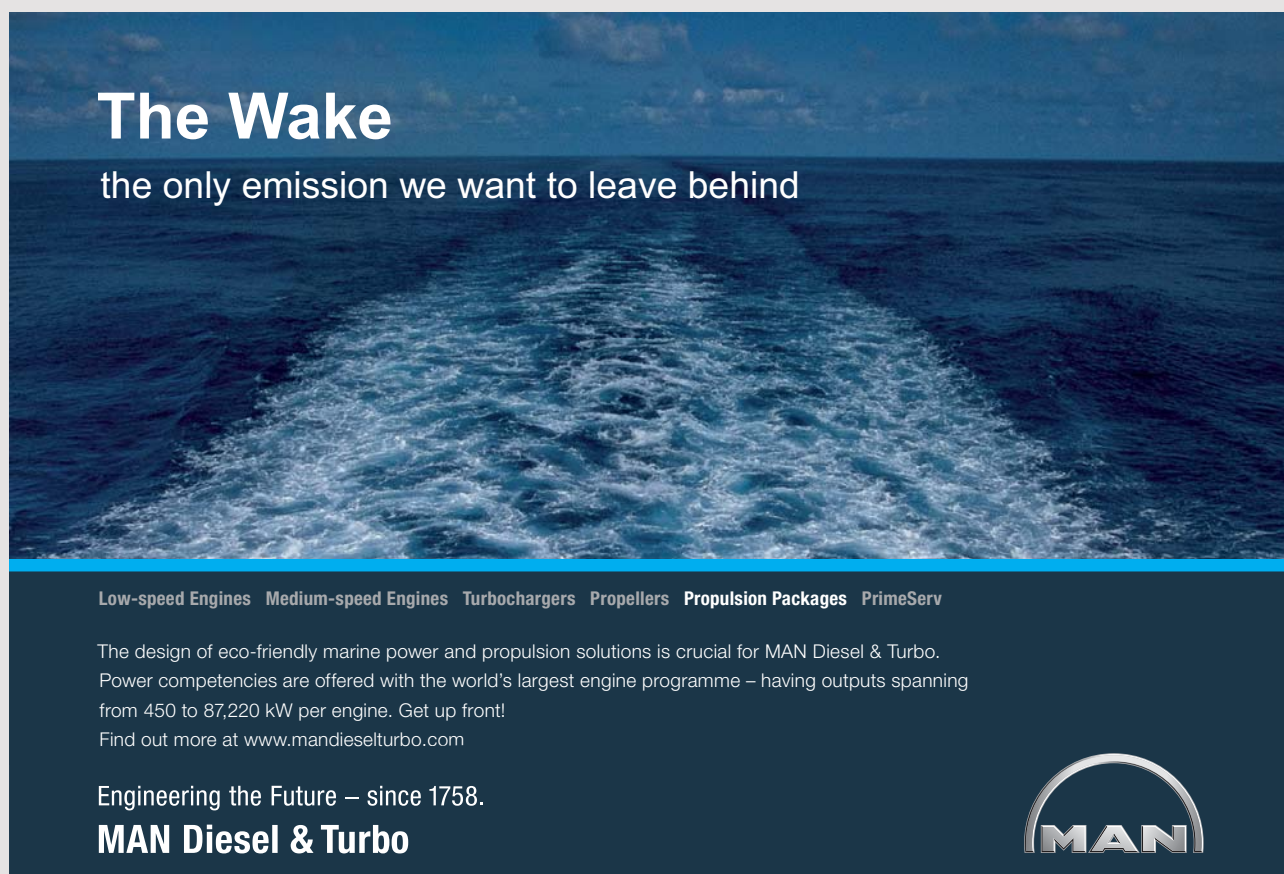
Download the latest application of oracle database server using the following link:

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/database12c-linux-download-2240591.html>

Run the following command to install oracle 12c and walk through the wizard:

```
path/to/db/runInstaller
```

This way, we finished most of this book's topics.



The Wake


the only emission we want to leave behind

Low-speed Engines Medium-speed Engines Turbochargers Propellers Propulsion Packages PrimeServ

The design of eco-friendly marine power and propulsion solutions is crucial for MAN Diesel & Turbo. Power competencies are offered with the world's largest engine programme – having outputs spanning from 450 to 87,220 kW per engine. Get up front! Find out more at www.mandieselturbo.com

Engineering the Future – since 1758.

MAN Diesel & Turbo



END NOTES

Please if you have any concerns regarding this book,

<http://bookboon.com>

Will be happy to listen to you.

Very much thanks to **Canonical LTD.** The company behind UBUNTU® for the technical review of the book.

Thank you so much for the time spent in reading this book and we all really hope it helps as much as it helps us.

For **Manon Niazi**

The Deutschlander

I left the Agricultural College and kept on learning and computer science in the computer academy until I meet Manon Niazi the Deutschlander because I want to make a device called “**Manon**”.