
IK1203 VT19 - Networks and Communication

Lab 2 Network Services

Student Name: _____

Date: _____

Lab Instructor: _____

Network Systems Lab (NSLab)
Communication Systems Division (COS)
EECS School
KTH - Royal Institute of Technology
Isafjordsgatan 39
164 40 Kista

Task 1

Lab Preparation

This lab will give you some hands-on experience working with Virtual Machines and setting up internet services and servers. You will learn how to set up a Linux virtual machine, configure a small virtual network and test some basic network services.

Note 1: Please pass the Lab-2 quiz from the Canvas course website before coming to the lab session, it is important to show that you read the lab manual already and are familiar with the required concepts. You cannot start the lab session without passing the quiz in advance.

Note 2: Please come to the lab on time, for the session that you are registered in daisy.

Note 3: Before coming to the lab session, please download Virtual Box software and the given virtual machine image file on your laptop. Installing Virtual Box on your laptop and setting up virtual machines is also highly recommended (Task 2).

1.1 Learning Objectives

Upon completion of this lab, you will be able to demonstrate the following:

- Installation and configuration of Virtual Machines using VirtualBox.
- Operation of a DHCP server in a network.
- Operation of a DNS server in a network.

1.2 Introduction

This section provides some introductory information about the DHCP and DNS servers which might be helpful to pass the quiz on Canvas.

1.2.1 DHCP Servers

Dynamic address assignment methods are often classified as being either stateful or stateless. In stateful address assignment, there is a server that keeps track of the IP addresses that are currently in use and the IP addresses that could be assigned to new hosts. The example of stateful address assignment that we will use in this lab, is the Dynamic Host Configuration Protocol (DHCP). In stateless address assignment, it is up to each individual host to find an appropriate address to use, with the help of a router that announces the network prefix the hosts are attached to. There are many examples of stateless address assignment protocols such as Appletalk, IPX, CLNP and IPv6. We will, however, not look into stateless address assignment methods in this lab.

1.2.2 DNS Servers

It would be really weird to surf the internet like `http://173.194.71.147`. Its much more convenient for us to use some thing like `http://www.google.com` instead. This, however, relies on the Domain Name System (DNS) service. DNS is a directory service, that maps human readable names to IP addresses and vice versa. It is a distributed and hierarchical service. A network node has to be set up with the DNS servers address, so that whenever a name translation is required, it can contact the DNS server to get the IP address of the node. In a DNS server, the mappings of IPs to names are stored in its scope, for the out of scope entries the server just propagates the requests to higher level servers.

1.2.3 Querying DNS Servers

The Domain Information Groper (DIG) is a powerful tool that can be used to query DNS servers. Dig is particularly useful for troubleshooting. You are required to show the output of several dig commands to the Lab Instructors in order for them to verify your configuration. First, let us try to decipher the output of the dig command. When you pass a domain name to the dig command, by default it displays the A record (the ip-address of the site that is queried) as shown below. Figure 1.1 displays the output of the dig command for google.com.

```
student@laptop:~$ dig www.google.com

;<<>> DiG 9.11.3-1ubuntu1.3-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51317
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6a5b31197458c00204abb6c45c458e671e6ffe9bd39a151 (good)
;; QUESTION SECTION:
;www.google.com.                IN                A

;; ANSWER SECTION:
www.google.com.                271               IN                A                216.58.207.228

;; AUTHORITY SECTION:
google.com.                    96266            IN                NS                ns4.google.com.
google.com.                    96266            IN                NS                ns1.google.com.
google.com.                    96266            IN                NS                ns3.google.com.
google.com.                    96266            IN                NS                ns2.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.                262282           IN                A                216.239.32.10
ns2.google.com.                262162           IN                A                216.239.34.10
ns3.google.com.                87251            IN                A                216.239.36.10
ns4.google.com.                98509            IN                A                216.239.38.10
ns1.google.com.                262282           IN                AAAA             2001:4860:4802:32::a
ns2.google.com.                262162           IN                AAAA             2001:4860:4802:34::a
ns3.google.com.                98509            IN                AAAA             2001:4860:4802:36::a
ns4.google.com.                98509            IN                AAAA             2001:4860:4802:38::a

;; Query time: 0 msec
;; SERVER: 130.237.72.201#53(130.237.72.201)
;; WHEN: Mon Jan 21 10:18:31 CET 2019
;; MSG SIZE rcvd: 335
```

Figure 1.1: Output of the dig command

The dig command output has the following sections:

Header: This displays the dig command version number, the global options used by the dig command, and few additional header information.

QUESTION SECTION: This displays the question it asked the DNS (i.e This is your input). Since we said “dig www.google.com”, and the default type that the dig command uses, is an A record, it is indicated that we asked for the A record of the google.com website (In the question section, dig reminds us of our query. The default query is for an Internet address (A)).

ANSWER SECTION: This displays the answer it receives from the DNS (i.e This is your output). This is the A records of www.google.com. Finally, we get our answers: the addresses of www.google.com. Notice that in this case www.google.com is probably not a single machine or more correctly a single IP address.

AUTHORITY SECTION: This displays the DNS name server that has the authority to respond to this query. Basically this displays available name servers of google.com. The authority section tells us what DNS servers can provide an authoritative answer to our query. In this example, google.com has four name servers.

ADDITIONAL SECTION: This displays the IP address of the name servers listed in the AUTHORITY SECTION. This field typically includes the IP addresses of the DNS servers listed in the AUTHORITY SECTION.

Stats Section: This section is found at the bottom of dig command reply, and shows the statistics including how much time it took to execute this query.

1.2.4 DNS Server Configuration

In this lab, we will configure the DNS server for our network. We will use the Berkeley Internet Name Daemon (BIND), which is the reference implementation of DNS protocol in Linux. We will set up a Master Name Server in BIND. A zone is a part of a domain name for which we will be configuring. For example, if our domain is example.com our zone is example.com. If we have a sub-domain under this domain, for instance, eng.example.com then another zone would be eng.example.com. It should be noted that our domain-name “example” is also a sub-domain of the global TLD (.com). Zones are described in zone files (sometimes called master files and normally located in /var/named). Please note that a zone ‘master’ is a server which gets its zone data from a local source as opposed to a ‘slave’ which gets its zone data from an external (networked) source (typically the ‘master’ but not always). Master status is defined in BIND by including ‘type master’ in the zone declaration section of the named.conf file, as shown by the following fragment:

```
zone "example.com" {
    type master;
    file "pri.example.com";
};
```

Figure 1.2: example.com fragment from named.conf

A minimal public zone file is shown below:

```
$TTL 86400
$ORIGIN example.com.
@ IN SOA ns1.example.com. root.example.com. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    86400 ) ; Negative Cache TTL
    IN NS ns1.example.com.
    IN MX 10mail.example.com.
ns1 IN A 192.168.100.1
www IN A 192.168.100.2
mail IN CNAME www.example.com.
```

Figure 1.3: zone file fragment for example.com

1.2.5 Reverse Mapping Overview

A normal DNS query would be in the form of ‘what is the IP of host=www, in domain=example.com’. There are times however when we want to be able to find out the name of the host whose IP address = x.x.x.x. Sometimes this is required for diagnostic purposes, more frequently these days it is used for security purposes to trace a hacker or spammer. Indeed, most modern mailing systems use reverse mapping to provide simple, first-cut, authentication

using a dual look-up process - IP to name and name to IP. IPv4 reverse mapping is not mandatory, though, as indicated by the mail example, it is essential for hosts that send mail, using either a Mail Transfer Agent (MTA) or a Mail User Agent (MUA).

We can make a zone file to describe all the hosts (nodes) in the Reverse Mapped zone using PTR Records. The resulting file will look something like this:

```
$TTL      86400 ; 24 hours, could have been written as 24h or 1d
$ORIGIN 100.168.192.IN-ADDR.ARPA.
@ 1D IN    SOA ns1.example.com.      root.example.com. (
                                2002022401 ; serial
                                3H ; refresh
                                15 ; retry
                                1w ; expire
                                3h ; minimum
                                )
IN       NS      ns1.example.com.
IN       NS      ns2.example.com.
1 IN     PTR      ns1.example.com.
2 IN     PTR      www.example.com.
```

Figure 1.4: reverse zone file fragment for example.com

1.2.6 The Resolver

resolv.conf is the name of a file used in various operating systems to configure the Domain Name System (DNS) resolver library. The file is a plain-text file usually created by the network administrator or by applications that manage the configuration tasks of the system. The **resolv.conf** program is one such program on FreeBSD or other Unix machines which manages the **resolv.conf** file.

In most Unix-like operating systems and others that implement the BIND Domain Name System (DNS) resolver library, the **resolv.conf** configuration file contains information that determines the operational parameters of the DNS resolver. The DNS resolver allows applications running in the operating system to translate human-friendly domain names into the numeric IP addresses that are required for access to resources on the local area network or the Internet. The task of determining IP addresses from domain names is called resolving.

The **resolv.conf** file contains the IP addresses of name servers that are available to the host in question. This is typically done in the following manner:

```
nameserver 192.16.1.254
```

Since we will set up our own DNS server, we need this entry to point back to the same computer.

Task 2

Virtual Machine Installation

In this lab, you will install two virtual machines using VirtualBox and configure some internet services on these machines. At the end of these steps, you should be able to start two virtual machines and connect them in the way shown in Figure 2.1.

Note: It is mandatory to download the VirtualBox and virtual machine image from <https://home.ug.kth.se>, before the lab session. It is highly recommended to install the VirtualBox and also configure the virtual machines before the lab session in order to speed up the process.

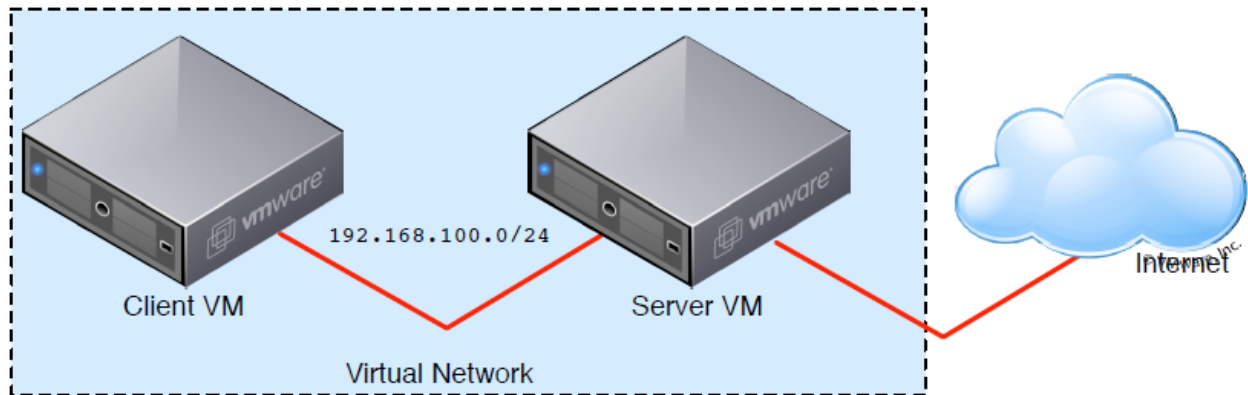


Figure 2.1: Virtual Machine Topology

2.1 Preparing your System

You will need at least 1GB memory and 16GB of free space on your computer's hard disk drive. You will also need to download the following software.

- A VirtualBox (latest edition) distribution for your platform of choice.
- An Ubuntu 14.04 image

2.1.1 VirtualBox Installation

To Install VirtualBox, perform the following steps:

1. Download the latest release of VirtualBox for your operating system from <https://www.virtualbox.org/wiki/Downloads>
2. Run the installation wizard and accept all of the default options.

When the installation of VirtualBox is complete, run the application. You should get the following main window shown in Figure 2.2.

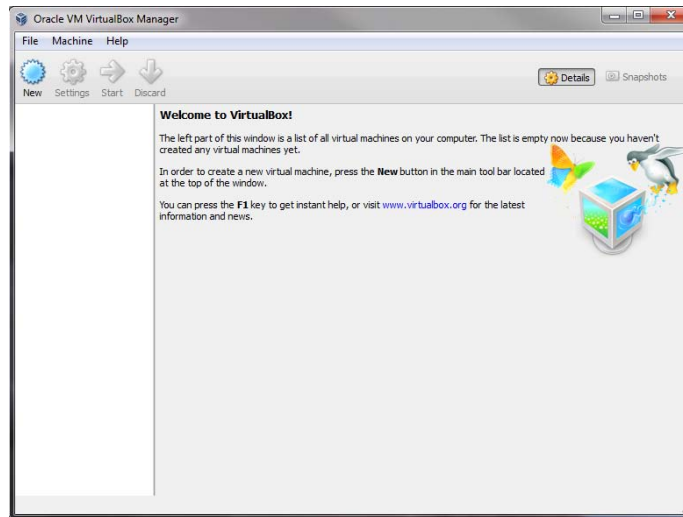


Figure 2.2: VirtualBox Main Window

2.1.2 Setting up the VM

Download the Ubuntu 14.04 image from <https://home.ug.kth.se> following the path as shown in the Figure 2.3.

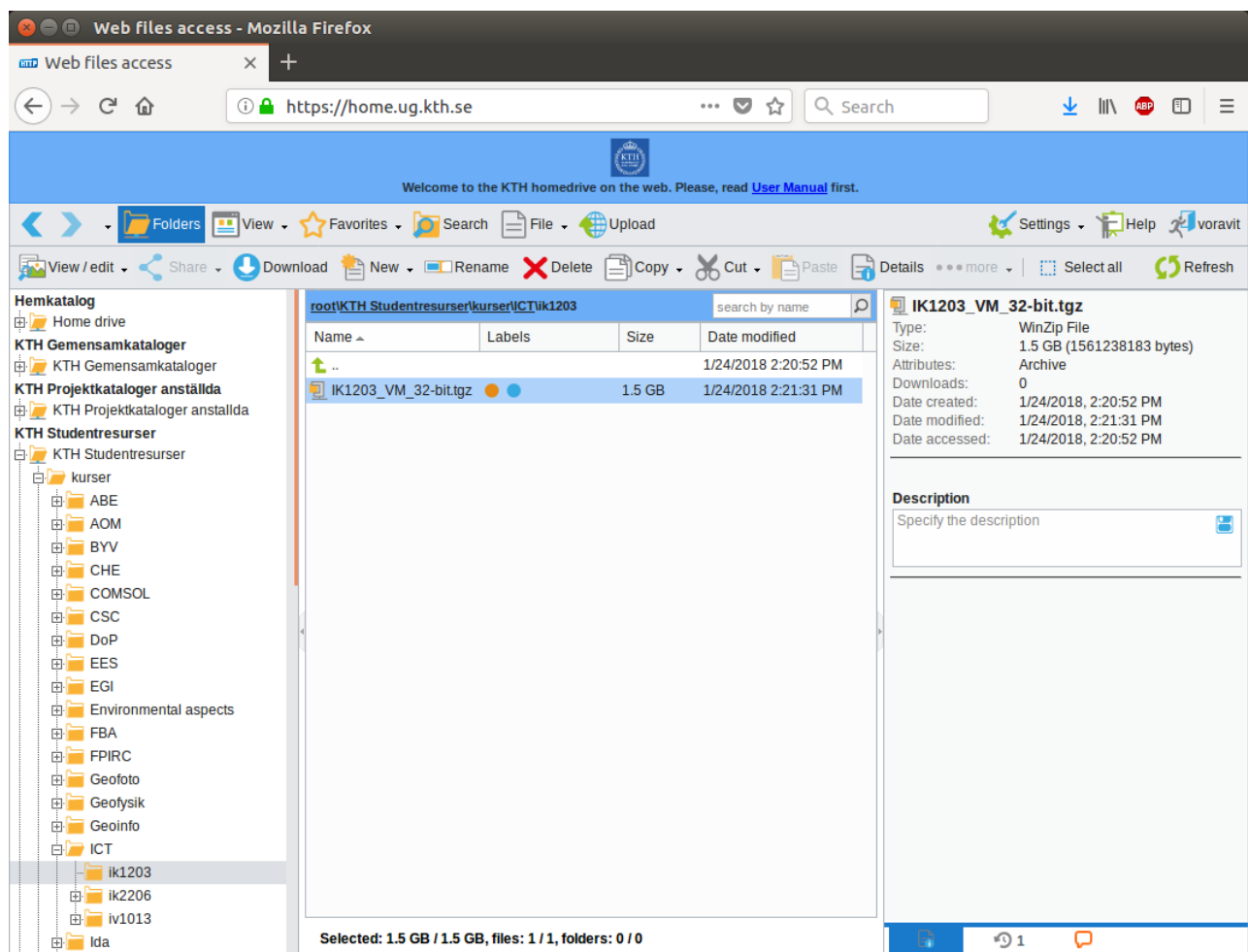


Figure 2.3: Location of Ubuntu 14.04 image (KTH Studentresurser→kurser→ICT→ik1203)

Unzip it. You should have an uncompressed file “IK1203_VM_32-bit.vdi”.

For this Ubuntu system:

Username: student

Password: lab_IK1203

Follow these instructions to set up the virtual machine:

1. Click the **New** icon in the VirtualBox main window. This brings up the dialog depicted in Figure 2.4(a).
2. In this dialog, enter the following parameters:
 - **Name:** lab2_server
 - **Type:** Linux
 - **Version:** Ubuntu (32-bit)
3. Click **next** to proceed with the virtual machine creation. At this point, a folder “lab2_server” should be created inside the “VirtualBox VMs” folder on your home directory. Now, move the file “IK1203_VM_32-bit.vdi” into “lab2_server” folder.
4. In the next dialog (shown in Figure 2.4(b)) you will have the option of selecting the memory size to allocate to your virtual machine. Leave the default option and click next to continue.
5. In the following dialog (shown in Figure 2.4(c)) you will be asked to create a hard drive for your virtual machine. Select the “**Use an existing virtual hard disk file**” option and select the file “IK1203_VM_32-bit.vdi”. Then, click **create** to continue.
6. Right click on the lab2_server image that you have just created in VirtualBox. Choose **Clone**.
7. Change the name to lab2_client, then continue.
8. Choose **Linked clone** as clone type, then continue.

If you have followed all of the instructions correctly, you should be returned to the main window. On the left side, you should see the two new virtual machines that you have created.

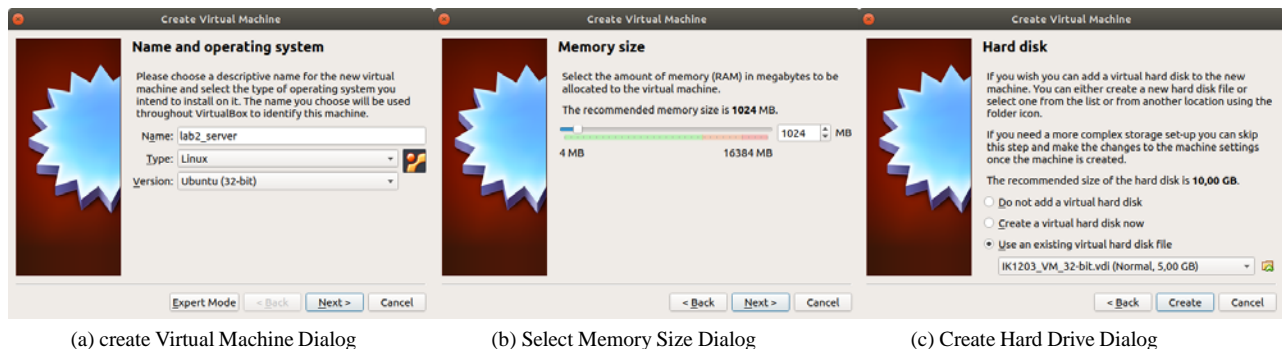


Figure 2.4: Create Virtual Machine Dialogs

2.2 Virtual Machine Configuration

Before starting the newly created virtual machines, we will configure them so that they are arranged as described in our topology in Figure 2.1. Let us first configure the server VM.

2.2.1 Server Virtual Machine Configuration

To configure the Server VM, do the following:

1. Select the server VM in the VirtualBox Manager window and click the **Settings** icon in the toolbar. This displays the Virtual Machine settings dialog.
2. Select the **Network** settings, and under **Adapter 1** change the attachment point to **NAT** (if it is not already).
3. Expand the **Advanced** tab and make sure the **Cable connected** option is checked.
4. Next, under the **Adapter 2** tab, check the **Enable Network Adapter** checkbox.
5. Change the attachment point for this adapter from **Not Attached** to **Internal Network**. Make sure the **Name** is **intnet**.
6. Expand the **Advanced** tab and also make sure the **Cable connected** option is checked for this network adapter.

2.2.2 Client Virtual Machine Configuration

Let us proceed to make similar changes to the Client VM.

1. Select the client VM in the VirtualBox Manager window and click the **Settings** icon in the toolbar. This displays the Virtual Machine settings dialog.
2. Select the **Network** settings, and under **Adapter 1** change the attachment point from **NAT** to **Internal Network**. Also make sure that the **Name** is **intnet**. Note that **Adapter 2** in the next tab should not be used (The checkbox “Enable Network Adapter” should not be checked!)
3. Expand the **Advanced** tab and make sure the **Cable connected** option is checked.

Task 3

DHCP Server Configuration

Congratulations, now you can start the virtual machines!

Start both server and client VMs by selecting them and click **Start**. Login with the user name and password given in the previous section.

Note: if you suffer from a 'log in loop', that the virtual machines return to the input password interface even after you type the correct password, then it is probably due to the reason that the virtualization option is not enabled in your BIOS.

Solution: Enter the BIOS interface before entering your system (Search in google 'YOUR-LAPTOP-Series' + BIOS to find the approach, usually by pressing F2/F8/F10 after your start you laptop).

Then find the hardware accelerated virtualization option in BIOS and enable it. Save the changes and restart your laptop.

By default, there is no IP address assigned for **eth1** on server VM and **eth0** on client VM. Start the Terminal in both server and client VM and proceed with the following steps:

3.1 Networking the Server VM

To network your **Server** Virtual Machine, do the following:

1. List all of the available interfaces on the VM using `ifconfig` command.
2. If **eth0** is not listed in your listing above, initiate a dhcp client request for it. This is the NAT enabled interface. Confirm that an IP address has been assigned. Please check that assigned address is **not** in the range of 192.168.100.0/24 network which will be used later on. (Hint: the command for doing this follows the same syntax as you have answered in one of the questions on moodle quiz. You need to configure your interface with **sudo** in the beginning of command in order to get the root user privileges)
3. Assign the static IP address **192.168.100.1/24** to the network adapter **eth1**. In this case, the IP needs to be assigned manually. Click the Network Manager icon in the toolbar at the top (a symbol with two arrows or the wifi symbol), go to Edit Connection. Click over the Wired connection 1 and Edit. In the Ethernet tab, select interface **eth1** in the Device MAC address, then change to the IPv4 Settings tab and change the Method to Manual. Click on Add and add the Address 192.168.100.1 and Netmask 24. Then save the changes and restart the server virtual machine. After rebooting, use `ifconfig` command in the terminal to verify that **eth1** interface has been assigned the correct IP address 192.168.100.1.
4. Check to be sure that you have internet reachability by sending a ping to: `www.google.com`. If it fails, there is something wrong with your configuration and you may need to ask the lab assistants for some help.
5. List all of the available interfaces on the VM again using `ifconfig` to make sure that **eth0** and **eth1** are properly configured.

In order to simplify the configuration of our server, we will use Webmin. This will make the configuration of our network services a lot simpler. The Webmin can be accessed using Firefox browser in the server VM by visiting <https://localhost:10000>. Once you get to the Webmin, log in using the student user credentials as given in the last section.

3.2 DHCP Server Configuration

Before we configure the DHCP server, let us confirm that the IP address has been properly assigned to **eth1**. An incorrect IP address assignment on this interface will cause the DHCP server to fail. To confirm the configuration of **eth1** using Webmin:

1. In Webmin, go to **Networking** → **Network Configuration** → **Network Interfaces**.
2. Confirm that **eth1** has the correct IP address settings (192.168.100.1). If not, click the device and configure it accordingly.

To configure a DHCP Server using Webmin:

1. Select **Servers** → **DHCP Server** → **add a new subnet**.
2. In the form that appears, enter the following information:
 - (a) **Subnet Description:** Client subnet
 - (b) **Network Address:** 192.168.100.0
 - (c) **Netmask:** 255.255.255.0
 - (d) **Address ranges:** 192.168.100.100 - 192.168.100.200
3. Leave everything else the same and click **create**
4. Check if the DHCP server listens for incoming queries on **eth1** interface. You can check it (and change if necessary) using **Edit Network Interface** button.
5. If everything was entered correctly, you can start the DHCP server by clicking the **Start Server** button at the bottom of the page.

3.3 Networking the Client VM

Now that we have configured the DHCP server correctly, we need to test it to see if our configuration works as expected. To do this, start the client VM, if you have not already done so.

1. List all of the available interfaces on the VM using `ifconfig`.
2. Initiate a DHCP client request on **eth0**. It should get an IP address from the DHCP server running in the server VM, within the address range 192.168.100.100 - 192.168.100.200
3. Confirm that the IP address has been correctly assigned. If you have problems with this step consult the lab assistants.
4. Look into the information contained in the files `/etc/dhcp/dhcpd.conf` located in the server VM and `/var/lib/dhcp/dhclient.leases` located in the client VM, be able to explain them to the lab assistants. (Hint: you can use 'gedit' text editor to open these files)

Task 4

DNS Server Configuration

4.1 Set up the resolver Using Webmin

1. In Webmin on the server VM, go to **Networking** → **Network Configuration** → **Hostname and DNS Client**
2. Under DNS servers, you should already have IP addresses specified. Webmin allows you to specify a Primary, Secondary and Tertiary DNS Server. Shift the existing entries down making them Secondary and Tertiary. Add the IP address **192.168.100.1** at the top of the list making it Primary.
3. In the search domains, select **listed** and enter **mylabdomain.com** in the text field.

If you cannot save the changes in Webmin, you can do as follows:

Inside the server VM, enter the command `sudo gedit /etc/resolv.conf` and modify the file to look like this:

```
nameserver 192.168.100.1
nameserver 127.0.1.1
search mylabdomain.com it.kth.se
```

Then save the file and you should be able to see the changes after refreshing the Webmin page by going to **Networking** → **Network Configuration** → **Hostname and DNS Client**

4.2 DNS Server Configuration

In Webmin, go to Servers. Then click on “**BIND DNS Server**”.

4.2.1 Creating a Master Zone

1. Click on “**Create master zone**”
2. Enter **mylabdomain.com** into the **Domain name / network** box.
3. For the master server, type in **ns1.mylabdomain.com** name as well.
4. Check the **Add NS record for master server?** box, if it is not already checked.
5. Enter **root@mylabdomain.com** in the **Email address** field.
6. Click **Create**

You will then be redirected to the **Edit Master Zone** page for the domain you have just created. This page can be used to add different types of DNS Records. We will now proceed to create records for our domain. First type of records we want to add is **A Records** (Address Records). To add **A Records** do the following:

4.2.2 Creating A Records (Address Records)

1. From the **Edit Master Zone** Page.
2. Click on **Address**
3. Leave the **Name** field blank, and type in **192.168.100.1** in the **Address** field.

4. Click **Create**.
5. Now add another A Record, type in **www** for the **name**, and type in **192.168.100.1** in the **Address** field.
6. Click **Create**.
7. Let's recall that we used **ns1.mylabdomain.com** when we created the zone. We also need to add an A Record for this. Type in **ns1** for the **name**, and type **192.168.100.1** in the **Address** field.
8. Click **Create**.
9. At the bottom, click **Return to Record Types**.

This is the bulk of setting up DNS through Webmin for our domain. We will need to start BIND in order to see the effects of our configuration. It is however, good practice to check that we have configured BIND correctly. From the **Edit Master Zone** Page, click the **Check Records** button at the bottom of the page. If no errors are reported, you can proceed with the lab instructions. If any errors are reported, ask the lab assistants to help you. In order to start BIND, click the **Start BIND** at the top right corner of the page.

Run the following dig command inside the terminal window of server VM and see the response:

```
dig @localhost mylabdomain.com
```

Make sure that you got an answer in the ANSWER section of reply to your query. Now, enter a dig command to find the Name Server (NS) record for your domain. This can be done by adding NS at the end of command given above. Observe the difference between two replies from the DNS.

4.2.3 Creating MX Records (Mail Exchange Records)

Create a Mail Exchange (MX) record with the following parameters:

- **Name:** mylabdomain.com
- **Mail Server:** mail.mylabdomain.com
- **Priority:** 1

Also add an address (A) record for **mail**, as you have done previously for **www** and **ns1**. Now, use dig command to find the MX record.

Note: Always click **Start BIND** in the top right corner after making any changes in the DNS server.

4.2.4 Modifying DHCP server configuration

Modify the DHCP server configuration in order to tell the client VM that it can now use the DNS server. Go to DHCP server previously created, press **Edit Client Options** button and enter the following parameters:

- **Domain name:** mylabdomain.com
- **DNS servers:** 192.168.100.1

You must click **Apply Changes** button in order for the changes in DHCP server to take effect.

4.2.5 Querying DNS server from client

From the client VM, force the **eth0** network interface to acquire an IP address (as done in Section 3.3). This time, the **/etc/resolv.conf** file should be updated with the IP address of name server. Check it by entering the following in terminal window of client VM:

```
gedit /etc/resolv.conf
```

Once the client has been properly configured, attempt to query the DNS server from client using the following commands:

```
dig mylabdomain.com
```

```
ping mylabdomain.com
```

Note: Before trying this, make sure that **eth1** interface on the server VM has the correct IP address 192.168.100.1

4.2.6 Adding a Reverse Mapping zone

Now, create a new zone in the DNS server in order to perform reverse lookups for 192.168.100.1. Create a new Master zone with the following parameters:

- **Zone Type:** Reverse
- **Domain name/Network:** 192.168.100.1
- **Master Server:** ns1.mylabdomain.com (Check the option “Add NS record for master server?”)
- **Email:** root@mylabdomain.com

Add a reverse address record with the following parameters:

- **Address:** 192.168.100.1
- **Hostname:** mylabdomain.com

In the client VM, enter the following dig command to look up for the reverse record:

```
dig -x 192.168.100.1
```

Congratulations!!
This Concludes the Lab.

Appendix

This lab instruction has been through many iterations of updates by the following people:

Johan Bilen (bilen@kth.se), Erik Eliasson (ekiasson@kth.se), Fredrik Lundevall (flu@kth.se), Karl-Johan Grinnemo (grinnemo@kth.se), Georgios Cheimonidis (gche@kth.se), Andreas Tsopelas (tsopelas@kth.se), Bruce Zamaere (bsiza@kth.se), Pawel Wiatr (wiatr@kth.se), Mozhgan Mahloo (mahloo@kth.se), Muhammad Rehan Raza (mrraza@kth.se), Yuxin Cheng (yuxinc@kth.se) and Voravit Tanyingyong (voravit@kth.se)