

Creating Private Cloud & Implementations (DevOps)

Khem Raj Neupane

Amber Karki

Project Report

ICT Cloud & Infrastructure Project

13.06.2018



Table of contents

1	Introduction	4
2	Network Architecture.....	5
3	Required Tools	6
4	Installation Process	6
4.1	ESXi Host Installation.....	7
4.1.1	Management Network Configuration.....	10
4.2	vCenter Server Appliance Installation and Configuration (VCSA)	13
4.2.1	Installation Requirements.....	14
4.2.2	Hardware Software Requirements	15
4.3	Installing VMware Workstation Pro 15	24
4.4	Creating VMs and Installing Windows Server (Winserver)	27
4.5	Installing Windows Server 2016 As Routing Server (WinRouter).....	35
5	Editing and Deleting VMs	35
6	Configuration of the Required Tools	36
6.1	Host Esxi 6.7 Configuration	36
6.1.1	Manage, Monitor Host & Create VMs	36
6.1.2	Storage.....	38
6.1.3	Networking.....	39
6.1.3.1	Creating vSwitches	39
6.1.3.2	Creating Port Group	39
6.1.3.3	Firewall Rules from ESXi Web Interface.....	40
6.2	vCenter Server Appliance.....	40
6.2.1	VMware Appliance Management	41
6.2.2	vSphere Client.....	41
6.3	Winserver Configuration.....	43
6.3.1	Network Connection Setting	44
6.3.2	AD Domain Services Installation.....	44
6.3.3	DNS Management.....	46
6.3.4	Forward Lookup Zones	47

6.3.5	Reverse Lookup Zones.....	48
6.3.6	Lookup Zones Test.....	50
6.4	WinRouter Network Topology	51
6.4.1	WinRouter Installation and Configuration	53
6.4.1.1	Routing Services Installation.....	54
6.4.1.2	NAT and LAN Routing Configuration.....	56
6.5	Network Connections	60
6.5.1	Remote Desktop Connection.....	60
6.5.2	SSH connection (PuTTY)	61
6.5.3	Direct Console User Interface.....	61
7	Continuous Integration and Deployment (CID)	62
7.1	DevOps	62
7.1.1	Webserver1 Configuration	63
7.1.2	JDK Installation and Setting Environment Variable.....	66
7.1.3	Tomcat Installation and Configuration	68
7.1.4	Jenkins.....	70
7.1.4.1	Jenkins Installation	70
7.1.4.2	Jenkins Configuration	72
8	Conclusion	84
9	References	85

1 Introduction

Cloud computing has been a buzzword in the information technology that has increased a lot of efficiencies, control, and resource management in cloud adapting companies. Massive computing resources distributed to the users have tremendously benefited them however, these benefits are extracted at the cost of issues like security, uptime performances, load balance etc. This is where our target of private could come into play.

The primary objective of the project is creating **private cloud environment** by using in-house resources and implementing IaaS, PaaS and SaaS service models for Business Optimization. The implementation scope is that the SMEs can not only create infrastructure with locally available means but also that they use this cloud model, customize it according to their own requirements and benefit significantly in terms of security, business efficiency, cost-effectiveness.

The main motivating factor behind this project is the private cloud infrastructure adoption trend of companies to optimize their own resources for several reasons as aforementioned. Throughout this project, the foundation for the cloud technology is first established, then this project moves further into deploying and implementing services, in which a test, e.g. using Jenkin, has been carried out at the end. This project also aims at making cloud aspirants understand what cloud infrastructure is like, what are the hardware and software requirements, management tools and how to configure them and so on. How and how much has the project achieved this objective, what can be further done in the future etc. have been explained chronologically with elaborated description and depictions.

2 Network Architecture

For this project, the provided static IP addresses, belonging to the tielab domain of Haaga-Helia, are in this range 172.28.230.35-40. Hence, for every public IP address, used in this project will not be anything other than that range, however in some configurations, IP addresses will be assigned automatically by the DHCP server of tielab of Haaga-Helia as well. Following **figure 1** describes most exactly about what is the cloud infrastructure and network architecture that this project has successfully built.

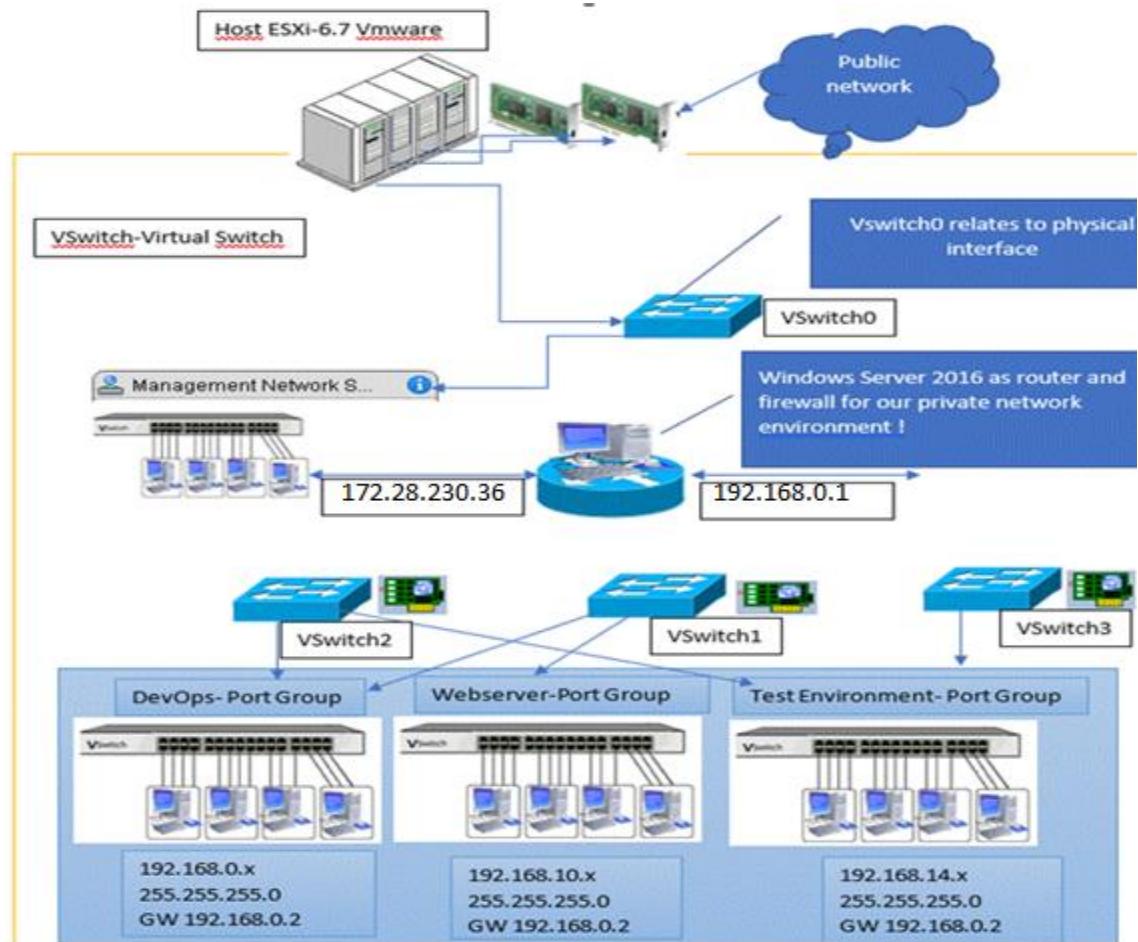


Figure 1, Network topology

The installed VMware Esxi- 6.7 accesses the public network of tielab domain through its two physical adaptors vmnic0 and vmnic1. vSwitch0 connects through physical interfaces and the Management Network also uses this switch. Three vSwitches are created- vSwitch1,

vSwitch2 and vSwitch3 along with three port groups- Webserver-Port Group, DevOps-Port Group and TestEnvironment-Port Group which are assigned to each switch respectively.

Under these port groups, many virtual machines are installed and configured in such a way that they use private network **gateway: 192.168.0.2. This gateway is the IPv4 address of the Routing Server.** Hence, Routing Server separates the private and public networks.

Here, the network that connects all the port groups with vSwitches 1,2,3 are inside the private domain. Now, each VMs within and outside the private domain can communicate with each other.

3 Required Tools

To meet the aforementioned network architecture, the following VMware tools and Operating Systems have been installed-

- VMware Esxi 6.7 hypervisor
- VMware vCenter Server
- VMware Workstation Pro 15
- Routing Server (Windows Serve 2016)
- Domain Controller Server (Windows Server 2016)

In the following sections, we will be explaining all about these tools and their needs for this project and their installation, configuration etc. processes step by step.

4 Installation Process

The virtualization tool, VMware Esxi 6.7 has been installed in the bare-metal hardware. This tool, along with many other related and required tools, can be downloaded for free from VMware's official website (<https://www.vmware.com/>). For the download, it is required to have a login credentials to the VMware official website. For new users, it is very easy to register in <https://my.vmware.com/web/vmware/registration>.

Since, VMware ESXi itself is like an operating system, so it is possible to run it in the bare metal, however it can also be installed on top of already running operating system or virtual machines. The main purpose of choosing VMware over other virtualization tools is that it has more configuration options and features from creating virtual machines to configuring and creating networks (private, public), routing, setting switches and ports, easy to manage either from web-client or from VMware workstation, a great tool that works as a common platform for all the installed devices, machines etc. to appear in one place from where their central management can be done very easily and effectively. Other systems installed are Windows Server (172.28.230.38) for FQDN, forward and reverse lookup zones, acting as a network router, Active Directory Domain Services etc. Their installations and roles in this project are discussed later herein.

4.1 ESXi Host Installation



Figure 2, ESXi 6.7 installation 1

- Downloading ISO file from VMware official link site
- Choosing bootable option-cd/external hard-drive/USB drive etc.
- Setting BIOS configuration for the booting
- Booting with iso image

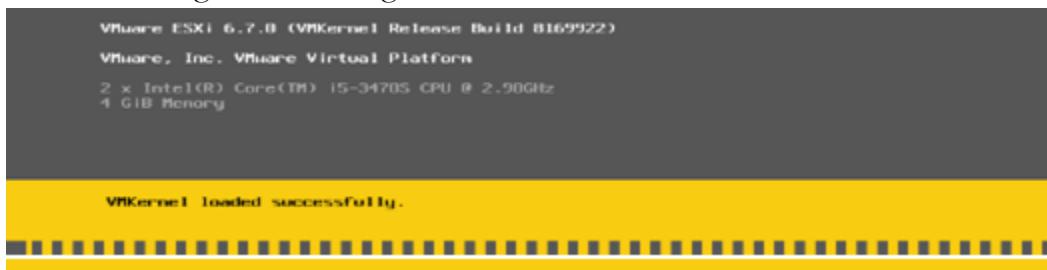


Figure 3, ESXi 6.7 installation 2

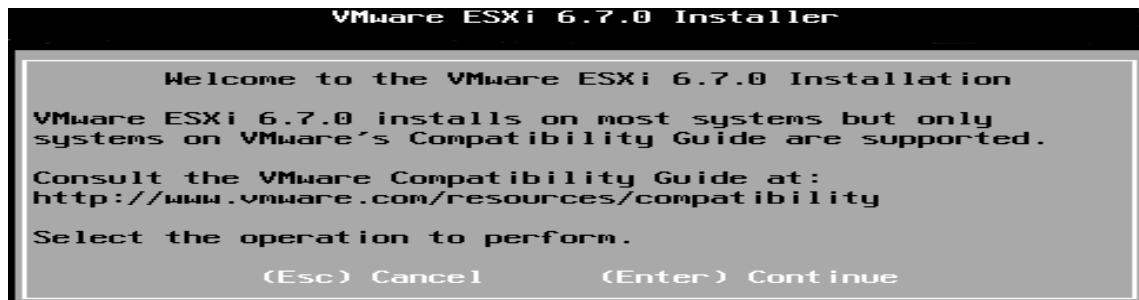


Figure 4, ESXi 6.7 installation 3

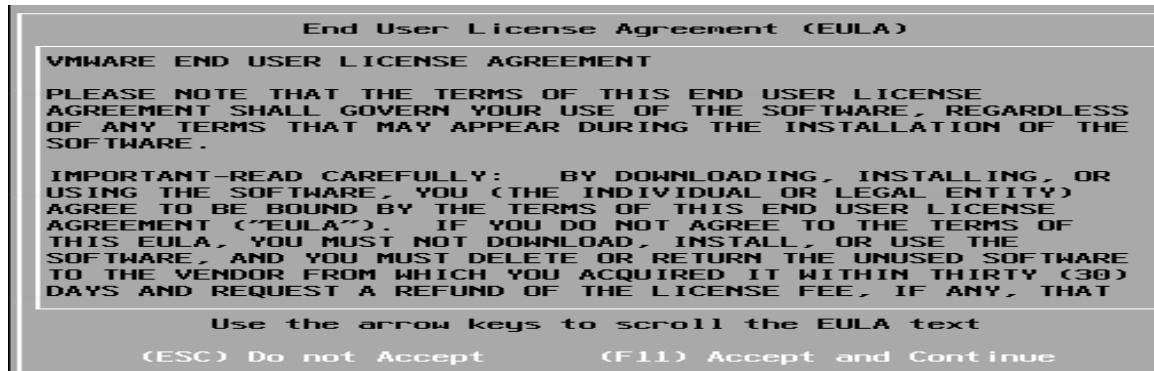


Figure 5, ESXi 6.7 installation 4

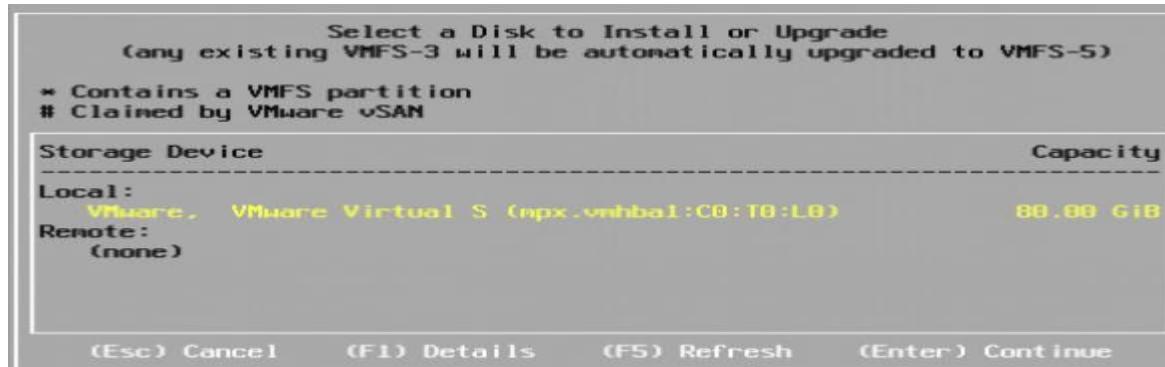


Figure 6, ESXi 6.7 installation 5

Root password can be set and confirmed for the first time here, nevertheless this can be changed later from the ESXi's GUI, after its complete installation.

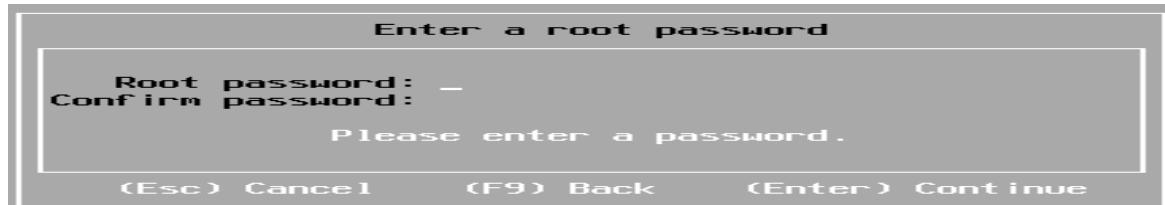


Figure 7, ESXi 6.7 installation 6

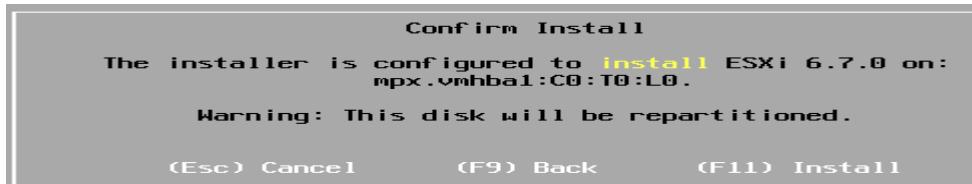


Figure 8, ESXi 6.7 installation 7

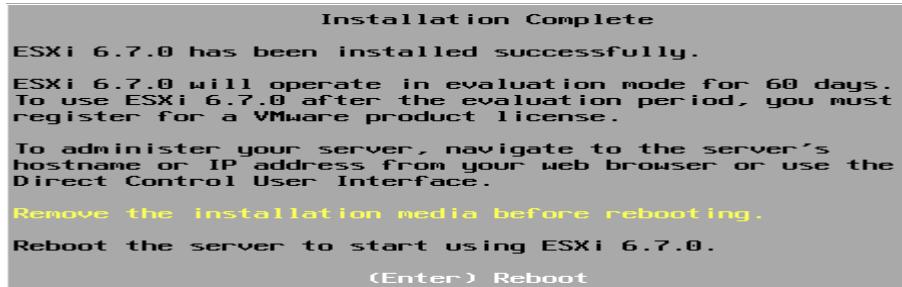


Figure 9, ESXi 6.7 installation 8

- As the installation is complete, the iso image or cd/external hard drive/USB drive, whatever is used, must be **unmounted** so that it does not load from the scratch again when the host is restarted next time.
- There are numbers of system configuration options to be taken into attention carefully because this is where the entire cloud network depends. However, this configuration can be done any time later when required (not the first or the last time configuration).

After the installation, the first default GUI of the Host Esxi looks like the following **figure 10**. However, many levels of configurations will be performed later on as per the project's requirements and that will be explained in the later parts of this project. Each time the host needs to be configured and customized, simply **F2** is pressed and username and root password are supplied to enter the System customization.

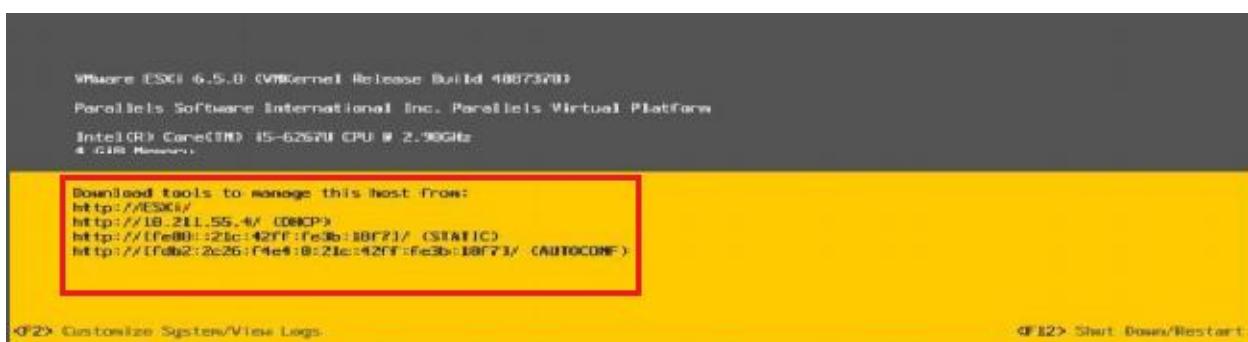


Figure 10, ESXi 6.7 installation completed.



Figure 11, ESXi 6.7 System Customization.

There are many system customization options along with their information, however more focus is laid on Management Network Configuration as explained in point 4.1.1.

4.1.1 Management Network Configuration

Management network configuration allows to configure the following options:

- Network Adapters:
This option displays all the connected physical adapters with the host and allows to connect/disconnect the host with them.

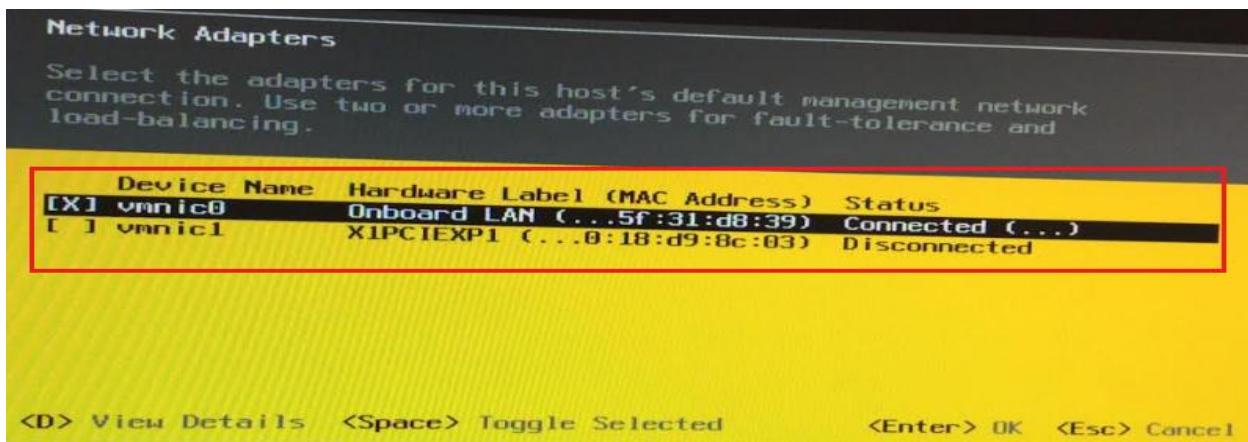


Figure 12, ESXi 6.7 Network Adapters vnic0 and vnic1

- VLAN Configuration:
This option allows to configure VLAN, if needed otherwise it will be safe without any configurations.
- IPv4 Configuration:

This option allows to set and configuration for the static IPv4, Subnet Mask and default Gateway to the host.

- IPv6 Configuration:

This is also the same configuration as in IPv4.

- DNS Configuration:

This option allows to configure primary DNS server, Alternative DNS server and host name.

- Custom DNS Suffixes:

- This option allows to set suffixes for the host if needed. The host name in this project is **Esxi1**. In the case of many hosts, is recommended to provide unique host names for their distinctive identification.

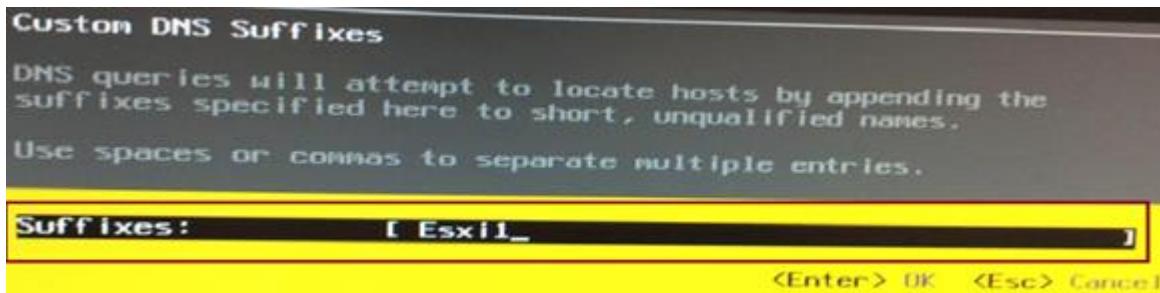


Figure 13, ESXi 6.7, giving the host a name as 'Esxi1'

- Ipv4 Configuration:

The Esxi host already contains default network setting, however from the three choices **Disable/Use dynamic/Set static IPv4 address and network configuration**, static one has been selected and Ipv4 address, Subnet Mask and Default Gateway is specified. The selection can be made by pressing space bar from the keyboard to move between these options and hitting the enter button. Normally, IPv6 does not need to be configured in this or any other configuration, therefore 'Disable IPv6' option needs to be chosen.

The configuration looks like this:

IPv4 Address: **172.28.230.39** (later, in DNS setting, this IPv4 address will be resolved in the Windows Server.)

Subnet Mask: **255.255.0.0**

Default Gateway: **172.28.1.254**

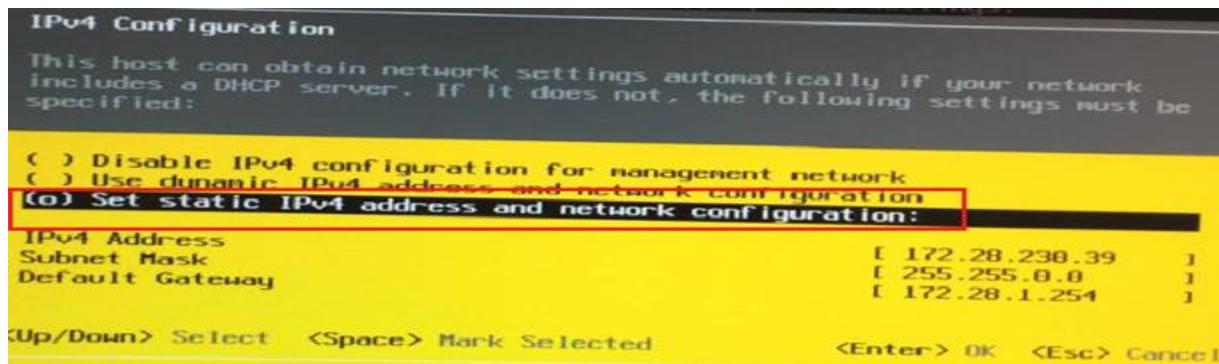


Figure 14, ESXi 6.7, IPv4 config.

- DNS Configuration:

Setting the Primary DNS Server to **172.28.11.67** for creating network environment in the host level later.

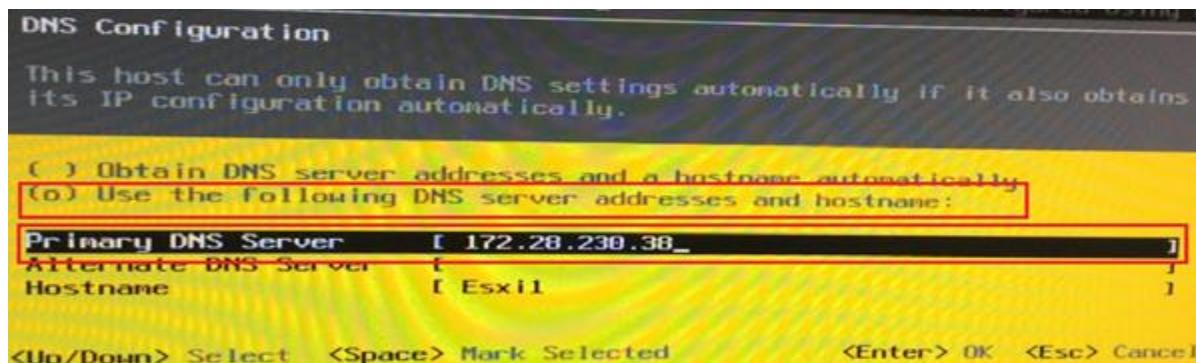


Figure 15, ESXi 6.7

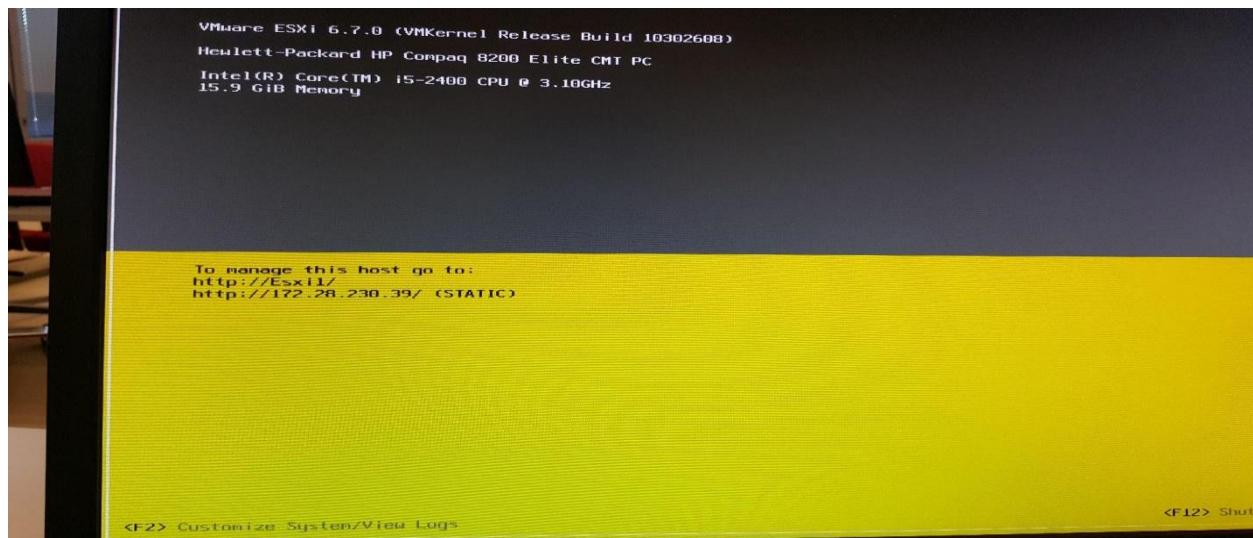


Figure 16, ESXi installed and running

By this stage, the ESXi host installation and configuration has been completed. So, the next step is to open and manage this host in the vCenter Server. We can either start its HTML 5 web-based vSphere client interface simply by entering the host's IPv4 address as URL address- <https://172.28.230.39/ui/#/host> or even in VMware Workstation with root login credentials, however workstation needs to be installed for this purpose.

Once the above link is clicked, it will lead to unsecured connection page as in the **figure 17** where yet, 'Proceed to 172.28.230.39 (unsafe)' option needs to be clicked to reach the actual login page of VMware Esxi host as in the **figure 18**.

This server could not prove that it is **172.28.230.39**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

[Proceed to 172.28.230.39 \(unsafe\)](#)

Figure 17, Unsecure Host ESXi 6.7 web log in.

What can be done after login inside this GUI, all about it has been clarified in the configuration section.

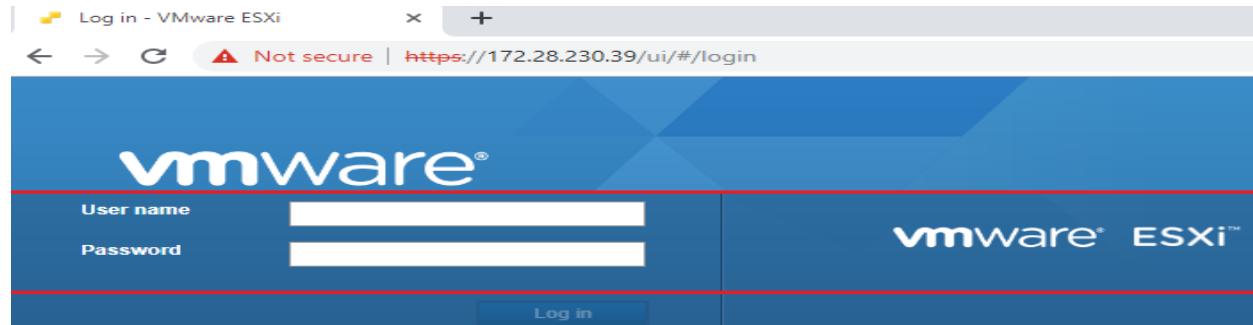


Figure 18, Host ESXi 6.7 web log in

4.2 vCenter Server Appliance Installation and Configuration (VCSA)

VMware vCenter Server Appliance (VCSA) (IPv4: **172.28.230.40**) is a centralized management tool which controls and configures the hosts from a single console and even automates some functionalities.

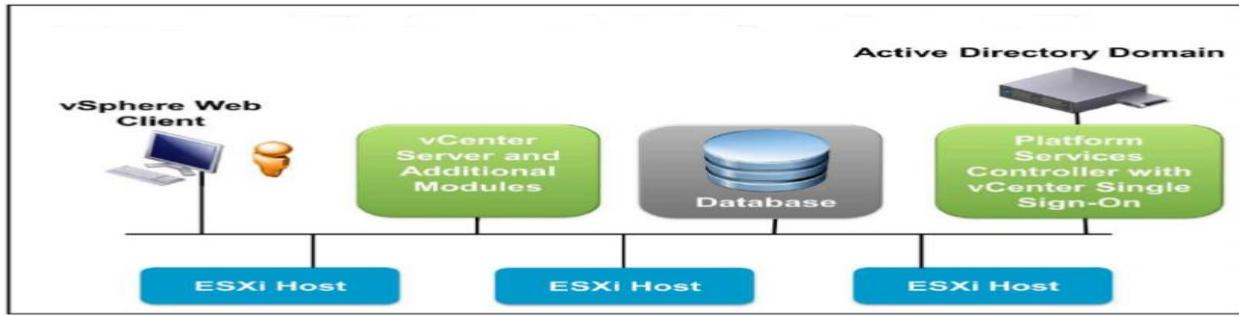


Figure 19, vCenter Server Architecture from VMware

It is the centralized management tool for the Esxi host. It requires database server that hosts the database repository, however the database ‘Microsoft SQL Server 2008 R2 Express’ comes by default but with limited scalability. However, for the purpose of this project the default database is enough. It supports bunches of other databases - Microsoft SQL 2005 SP3, IBM DB2, Oracle 10g R2. But if you are planning to scale your environment with more than 5 hosts and 50 virtual machines, the default database is not enough. VCSA is connected to our host and all the necessary configurations are made through this.

In our project, we required the VCSA for:

- Creating Virtual Switches
- Creating Standard Virtual Port Groups
- Creating Virtual Machines and assigning them to the destination port groups
- Creating private network
- Establishing connections between hosts and VMs etc.
- Creating Remote Desktop Connection for the virtual machines.

4.2.1 Installation Requirements

The most recent version of VCSA needs the following hardware and software requirements can be found from this link- (<https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.esxi.upgrade.doc/GUID-DEB8086A-306B-4239-BF76-E354679202FC.html>)

4.2.2 Hardware Software Requirements

- Number of CPUs - Two 64-bit or one 64-bit dual-core processor
- Processor - 2.0GHz or higher Intel or AMD processor
- Memory - 4GB RAM minimum
- Networking - Gigabit connection recommended
- 64-bit operating system

During the installation process, VCSA needs to use a FQDN, therefore it is highly recommended to configure a domain name in the DNS management of Windows Server, which is mentioned elaborately in the upcoming sections. However, all the VMs that we create for all purposes must be registered in the DNS and look up zones must be resolved.

After understanding the hardware and software requirements, the latest version of VCS 6.7, is downloaded from the VMware official site.

Access the download path then right click on ISO file for mounting then follow this path
(F:\vcsa-ui-installer\win32\installer.exe)

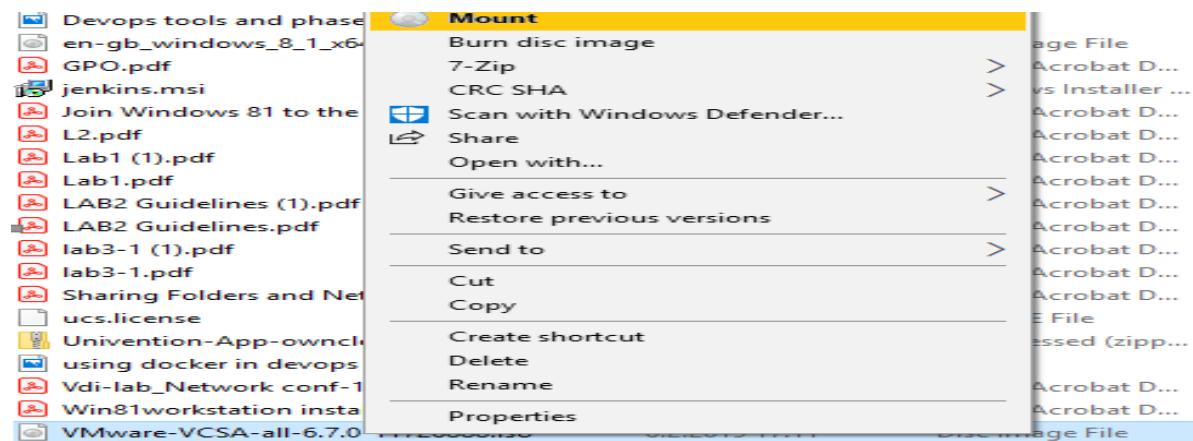


Figure 20, Mounting ISO image for VCSA installation

Once the mounting is done, following the 'vcsa-ui-installer' path then inside 'win32' there is the 'installer'. Double clicking this will lead to the following options as in the **figure 21**, from where clicking 'Install' takes to install it afresh (here, we are not concerned with Upgrade, Migrate or Restore options). Rest can be followed exactly as in the following figures:



Figure 21, Fresh install of VCSA 1

This screenshot shows the first step of the vCenter Server Installation process:

Introduction

This installer allows you to install a vCenter Server Appliance 6.7 or a Platform Services Controller 6.7.

The left sidebar lists the steps:

1. Introduction
2. End user license agreement
3. Select deployment type
4. Appliance deployment target
5. Set up appliance VM
6. Select deployment size
7. Select datastore
8. Configure network settings
9. Ready to complete stage 1

The main area shows "Stage 1" with the "Deploy appliance" step, which involves a box with a green checkmark icon. "Stage 2" is shown below it with the "Set up appliance" step, which involves a box with gears.

Text at the bottom: "Installing the appliance is a two stage process. The first stage involves deploying a new appliance to the target vCenter Server or ESXi host. The second stage completes the setup of the deployed appliance. Click Next, to proceed with stage 1."

Figure 22, Phase -1, vCenter Server Installation 2

This screenshot shows the "End user license agreement" screen of the installation wizard:

End user license agreement

Read and accept the following license agreement.

The text of the license agreement is partially visible, mentioning "Infringement Claim" and "Remedies". A checkbox at the bottom is checked, indicating acceptance.

The left sidebar shows the step 2: "End user license agreement".

Figure 23, Phase -1, vCenter Server Installation 3

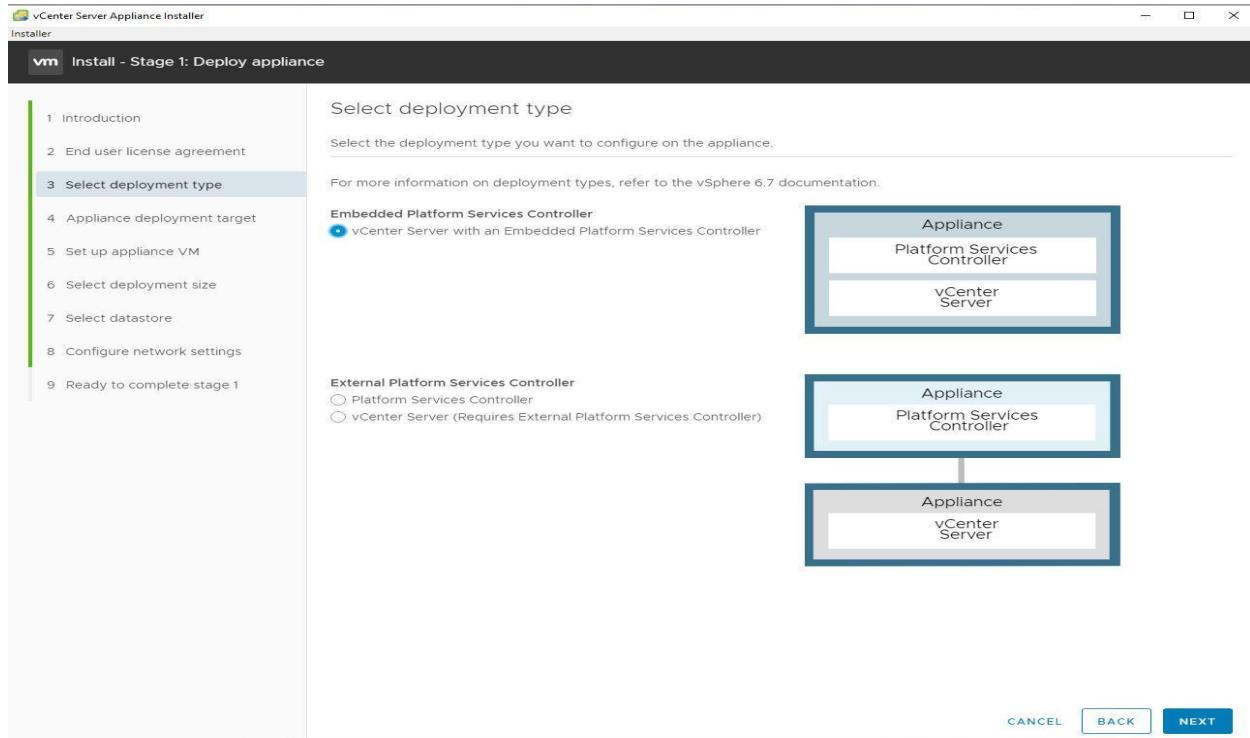


Figure 24, Phase -1, vCenter Server Installation 4

Deployment target should be the Esxi host – 172.28.230.39, HTTPS port is 443 by default and login credentials can be decided first time here, so it is important to note them down, however they can be changed later on.

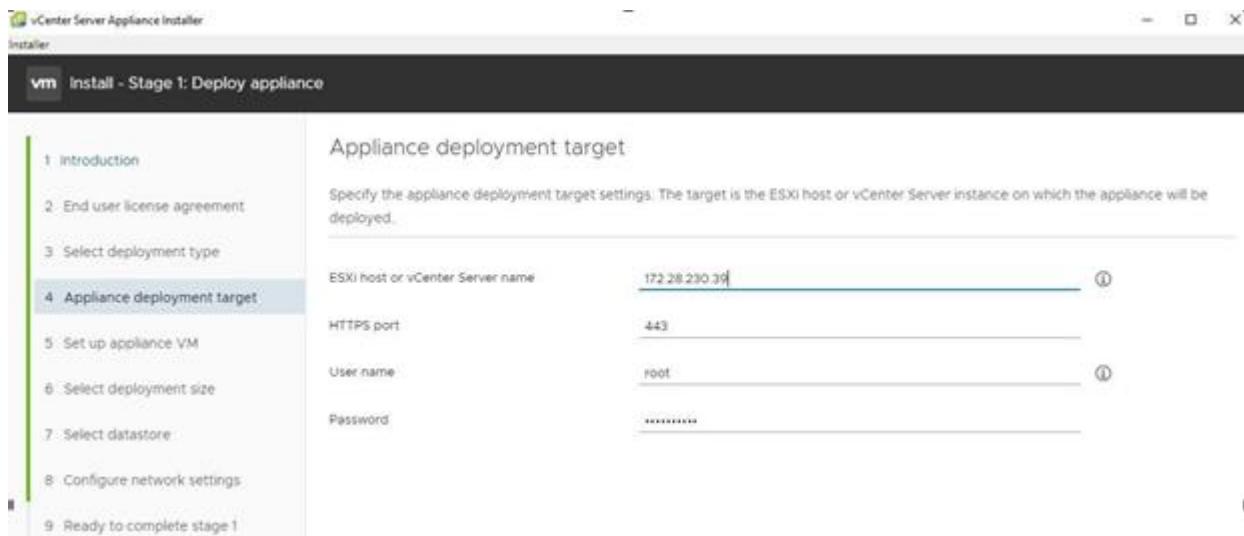


Figure 25, Phase -1, vCenter Server Installation 5

Provide any name for this virtual machine and set its root password.

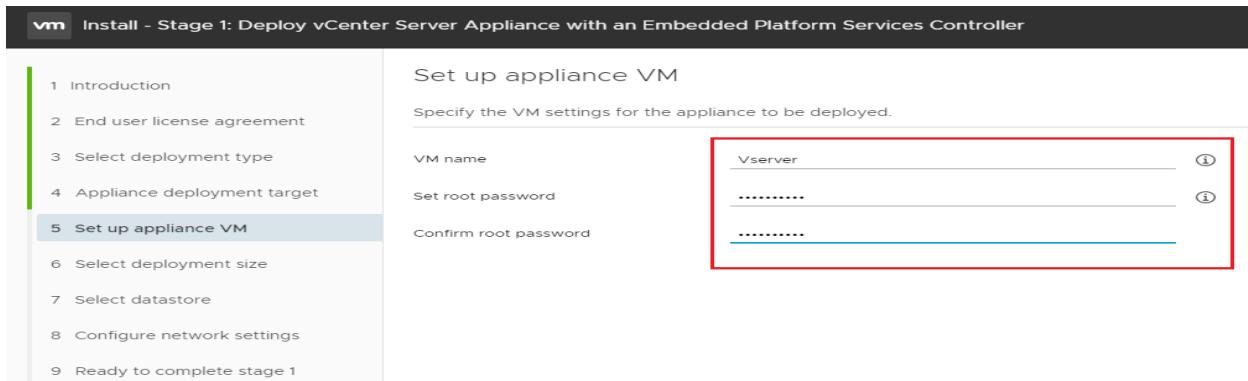


Figure 26, Phase -1, vCenter Server Installation 6

Tiny deployment size and default storage size has been chosen according to the provided resource and this option is enough for this project purpose.

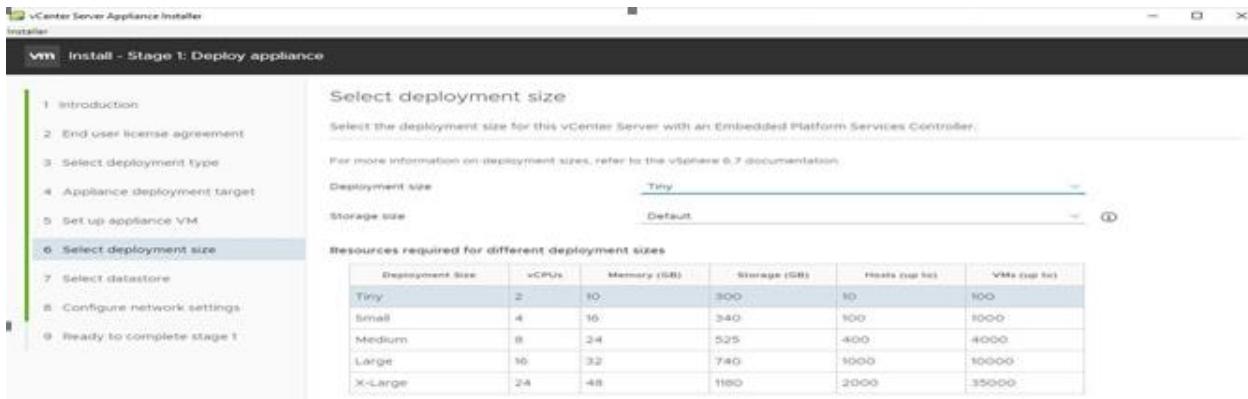


Figure 27, Phase -1, vCenter Server Installation 7

Any available data centre that has enough space can be chosen as VCSA's storage location.

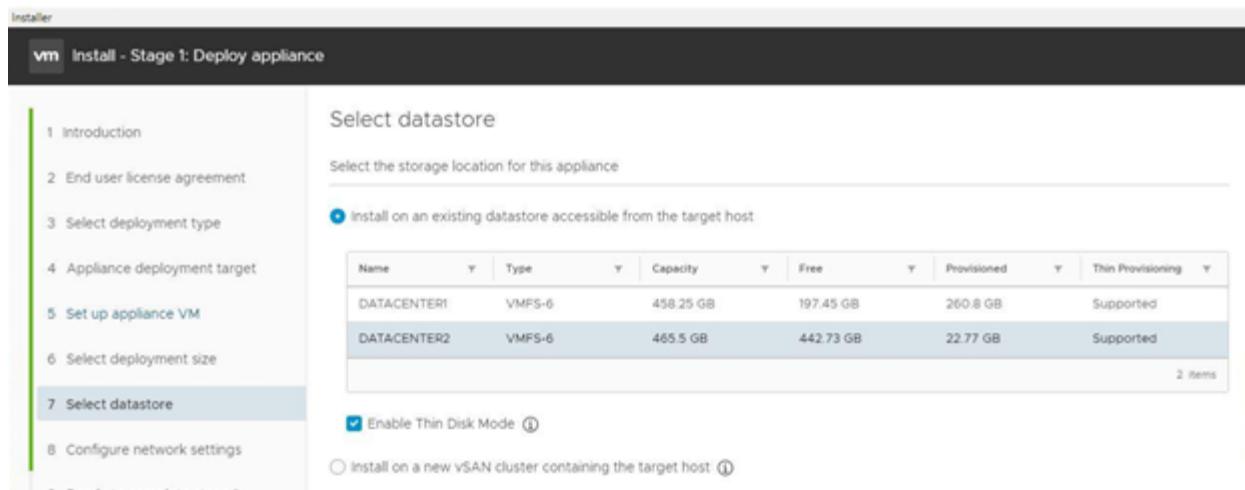


Figure 28, Phase -1, vCenter Server Installation 8

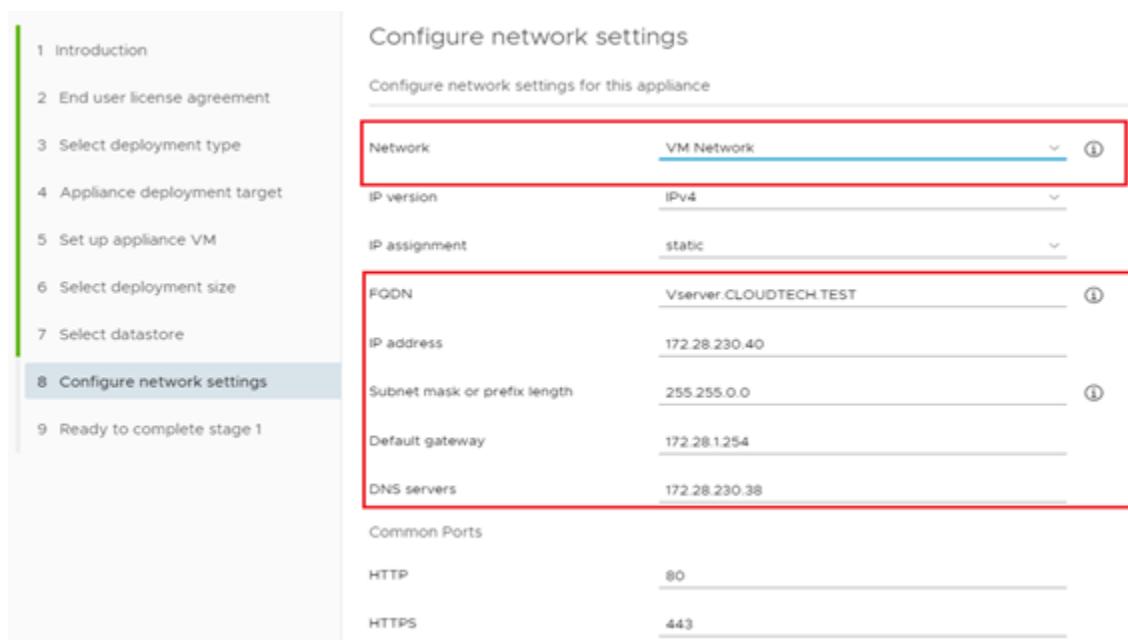


Figure 29, Phase -1, vCenter Server Installation 9

In network setting configuration, the network option can detect whatever the available networks are inside the Esxi host. A purposeful network, where the VCSA needs to be placed or installed, can be selected. In this project, the default 'VM Network' has been chosen. Another important thing to remember is to provide a fully qualified domain name (FQDN) for VCSA. Unless a FQDN is provided or not a FQDN or non-existing domain name is provided, its second phase of installation will fail, despite that the first phase will go successfully. Make sure that the DNS server, where the DNS record with reverse/forward lookup zone

are registered, is running at the moment of VCSA installation. See point **6.3 'Winserver Configuration'**. Other required and important network configuration here are to provide IP address and DNS Server to use (public IP address used here come from within the IP address range provided to us) and the default gateway is the school's teilab domain **172.28.1.254**.

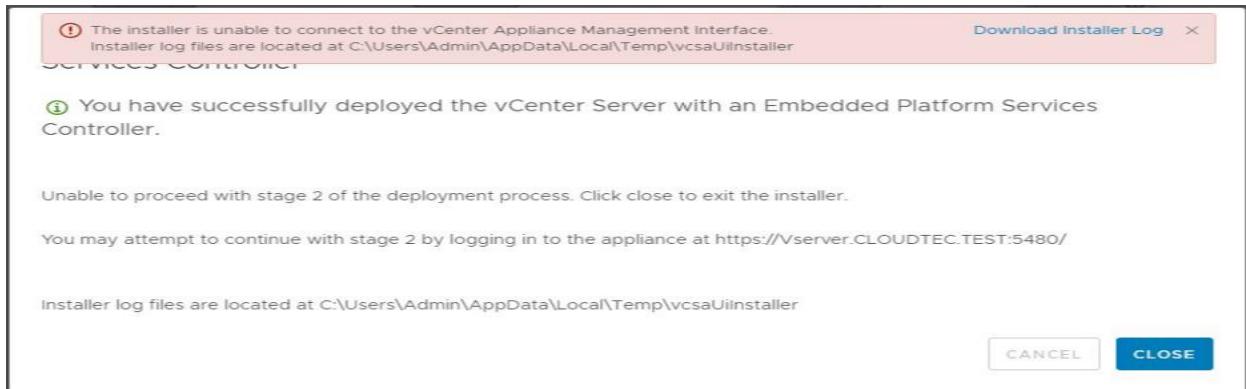


Figure 30, End of the first phase installation gives a link- <https://Vserver.CLOUDTECH.TEST:5480/>

By clicking the link <https://Vserver.CLOUDTECH.TEST:5480/>, obtained from **figure 30**, we enter the second stage of installation.

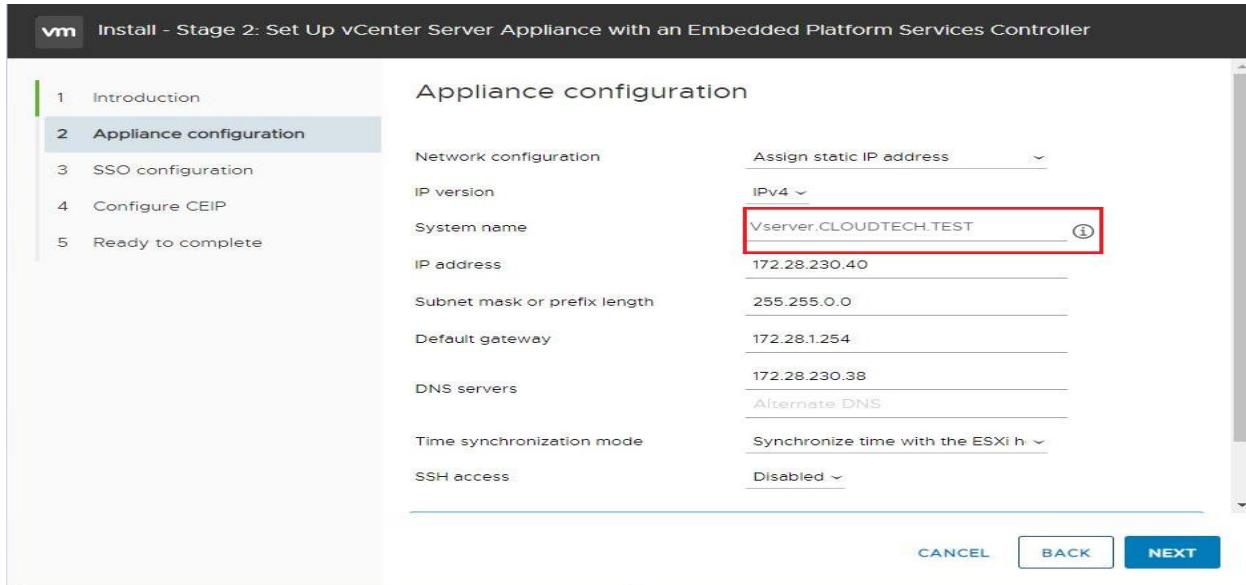


Figure 31, Stage -2, VCSA Installation 2

Here, remember to provide FQDN for System name.

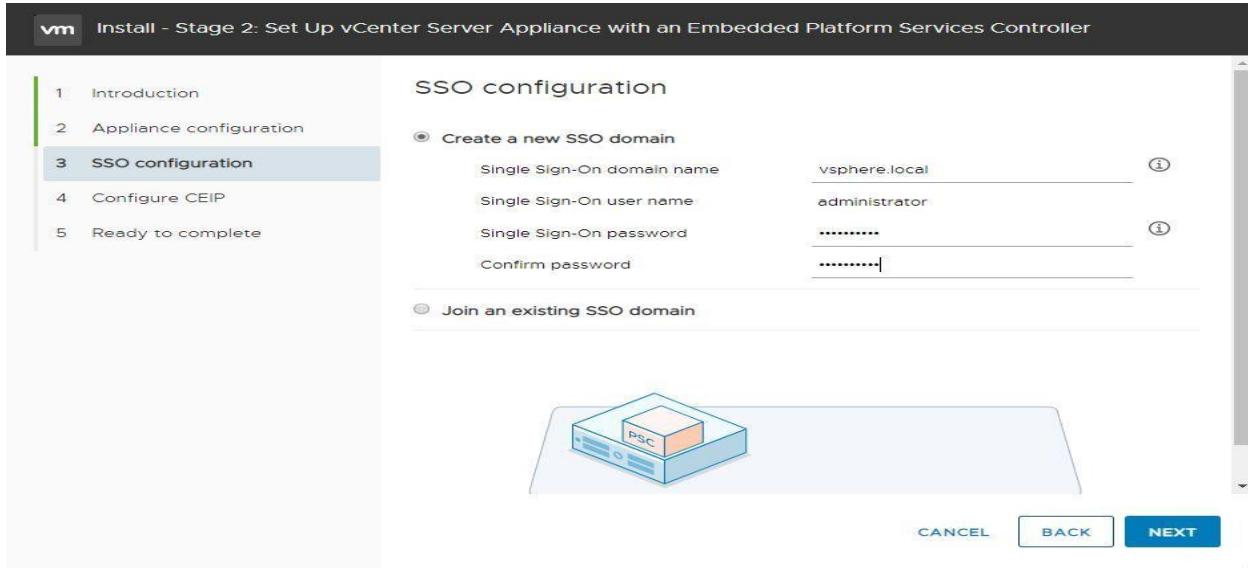


Figure 32 Stage -2, VCSA Installation 3

Provide SSO domain name as default **vsphere.local**.



Figure 33 Stage -2, VCSA Installation 4



Figure 34 Stage -2, VCSA Installation 5

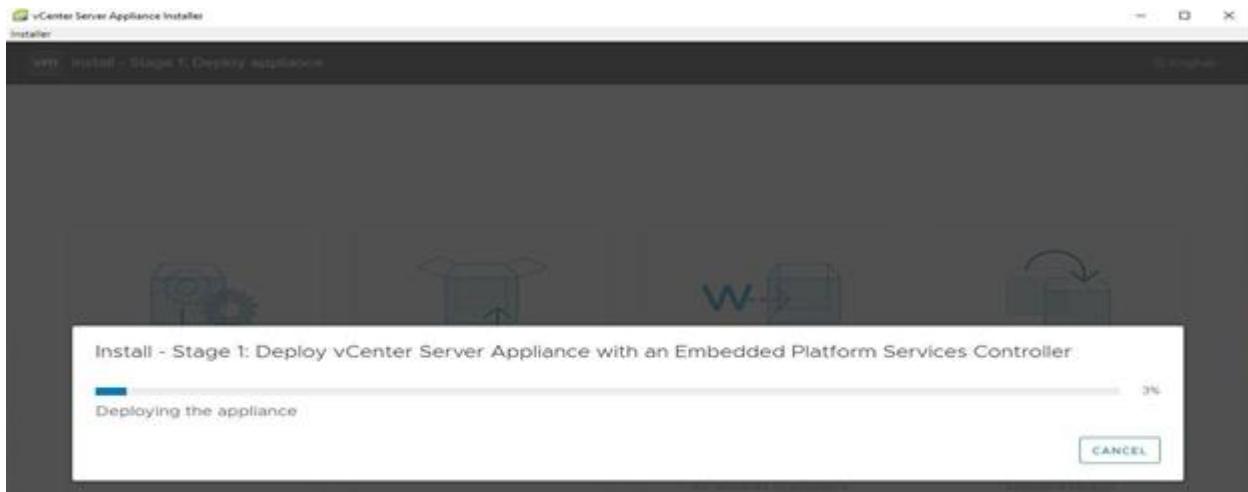


Figure 35, Stage -2, VCSA Installation 6

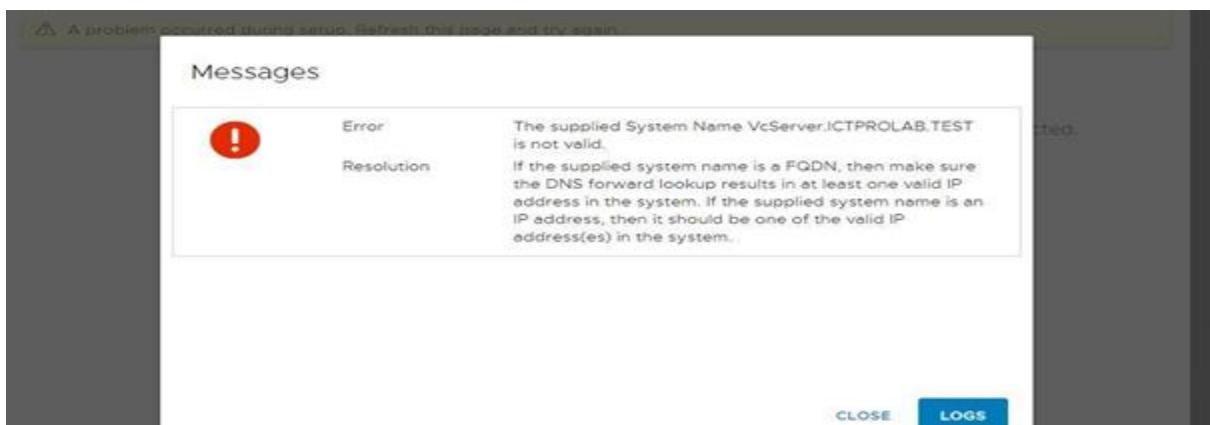


Figure 36, Stage -2, VCSA Installation 7

Figure 36 is just an example of an error arising from the problem in FQDN that was provided in the first stage installation in **figure 28** and the stage 2 installation in **figure 30**. For the demonstration purpose, this domain name registry from the DNS Server was removed while installing VCSA so as to get this error message for better learning.



Figure 37, Stage -2, VCSA Installation 8

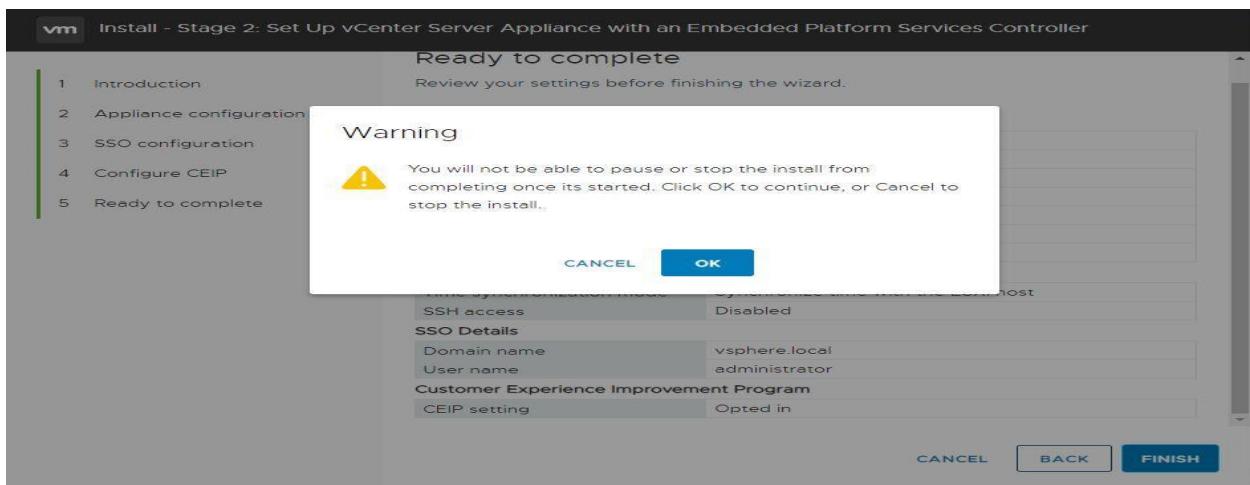


Figure 38, Stage -2, VCSA Installation 9

Figure 38 is just a warning so click ok and the installation will complete as in figure 39.

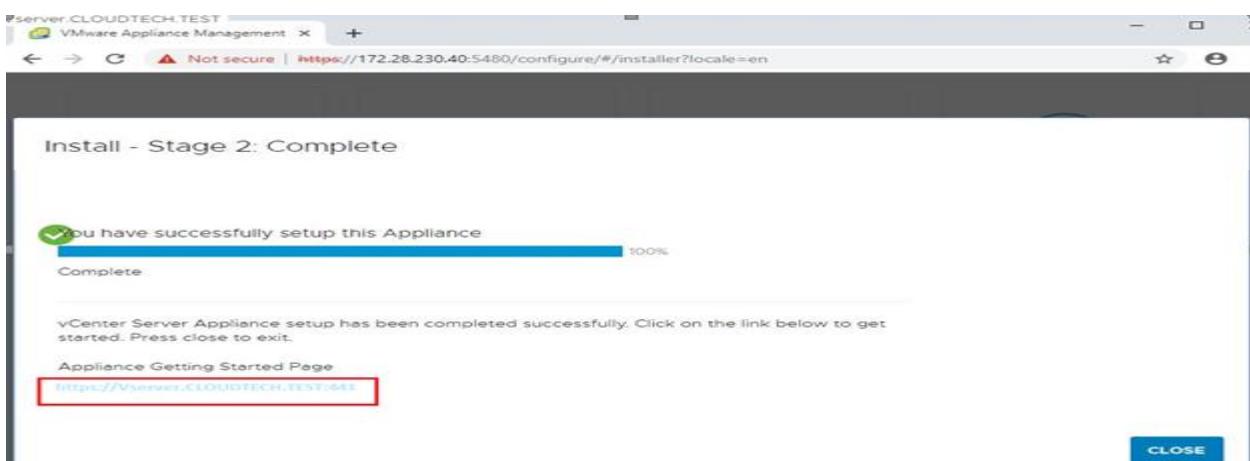


Figure 39, Stage -2, VCSA Installation Complete 10

Now that the installation has finished, it can be accessed in the browser by using the link as in **figure 39**. There are ‘HTML5’ and ‘FLEX’ versions available, as the ‘FLEX’ version requires flash player and is heavy too, hence HTML5 is lighter and more effective version recently (**figure 40**).

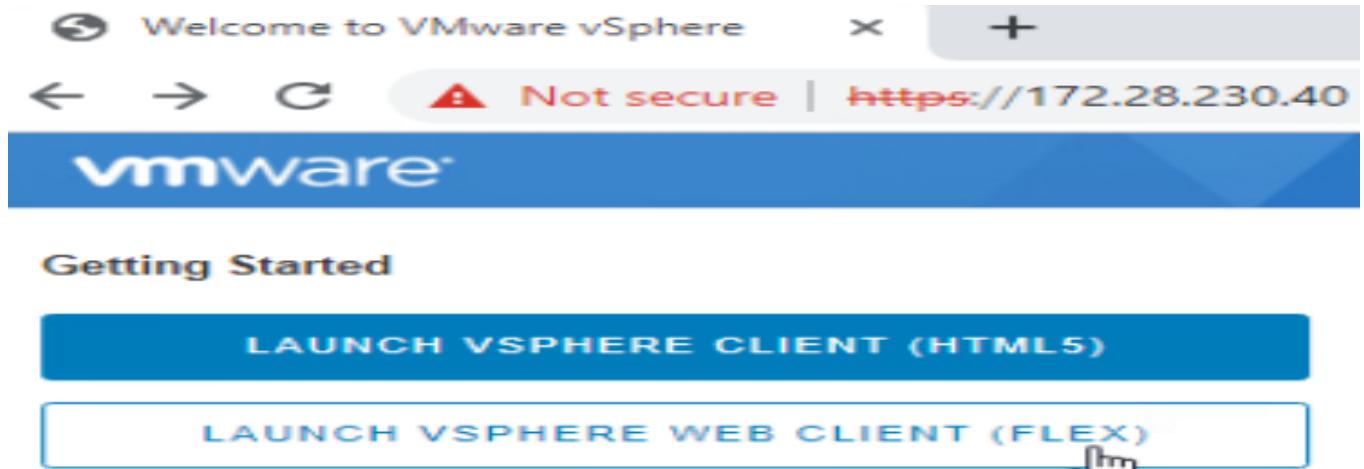


Figure 40, VCSA logging in process

4.3 Installing VMware Workstation Pro 15

VMware Workstation is a platform that allows complete setup even for own private network environment with the running operation system. In another word, it also works as a remote desktop connection for VMware products. It has that kinds of capability that supports all kinds of security features. Another interesting feature is that it has its own DHCP server configuration if needed. It also allows to connect to the host along with virtual machines creations to their configuration that can be changed or edited or removed as needed. The latest version of VMware Workstation PRO 15 is an improved version that contains security and bug fixes features along with its compatibility with these operating systems- Windows, Ubuntu 19.04, RHEL 8.0, Fedora 30, ESXI 6.7.

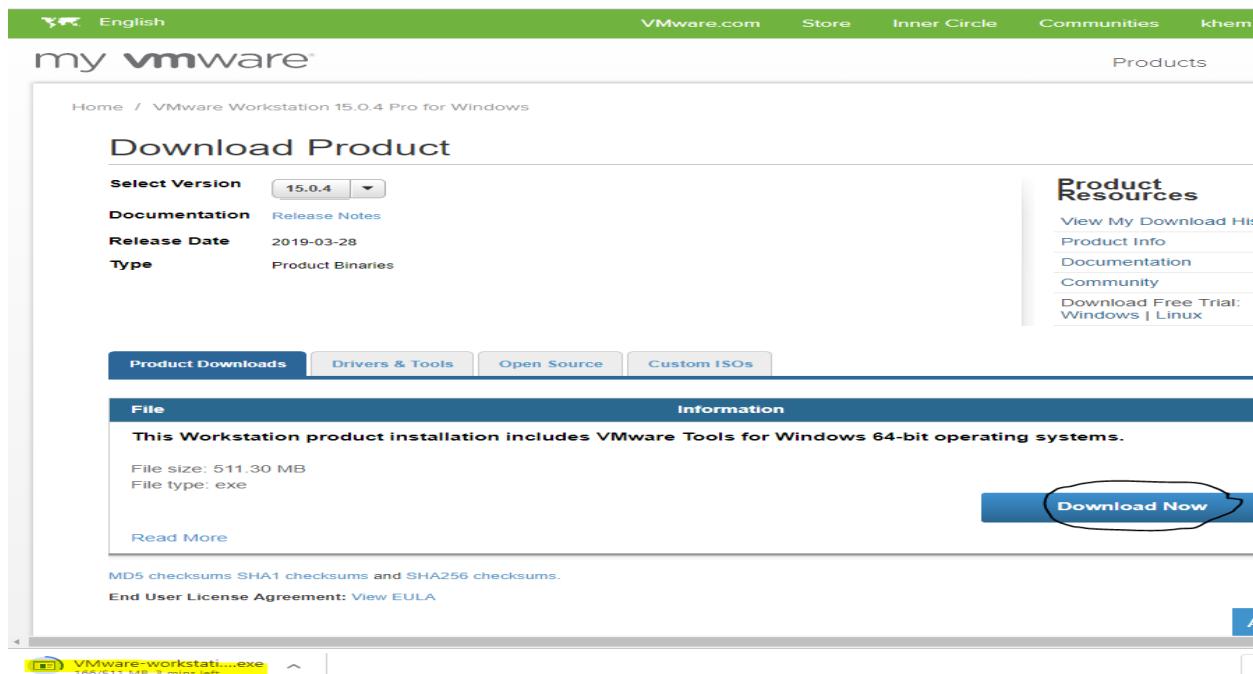


Figure 41, Installing VMware WorkStation Pro 15

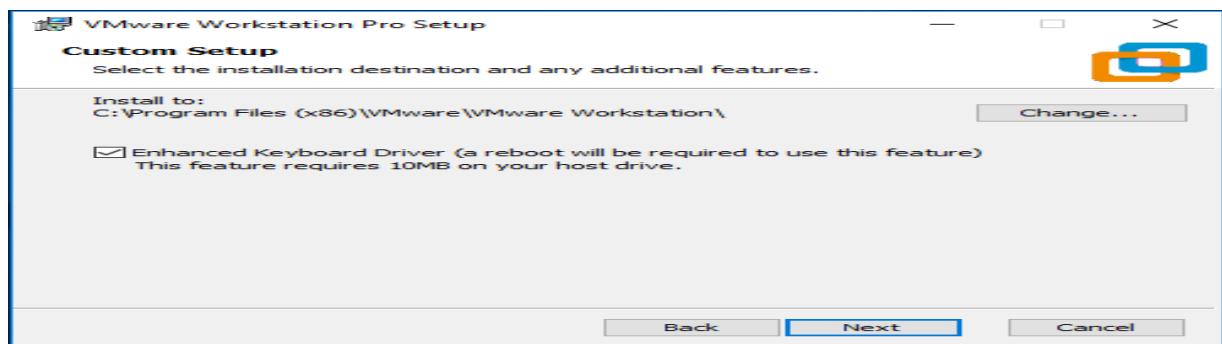


Figure 42, Installing VMware WorkStation Pro 15

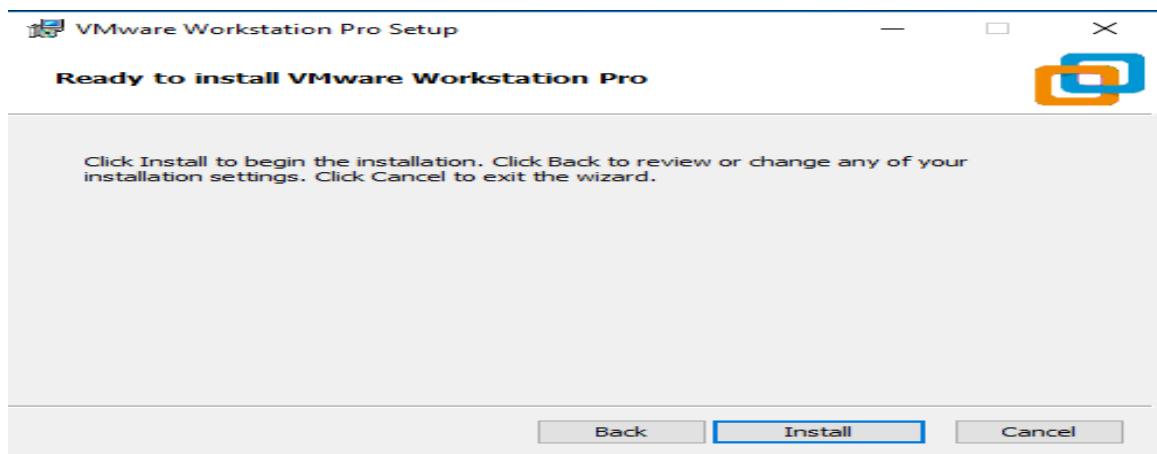


Figure 43, Installing VMware WorkStation Pro 15

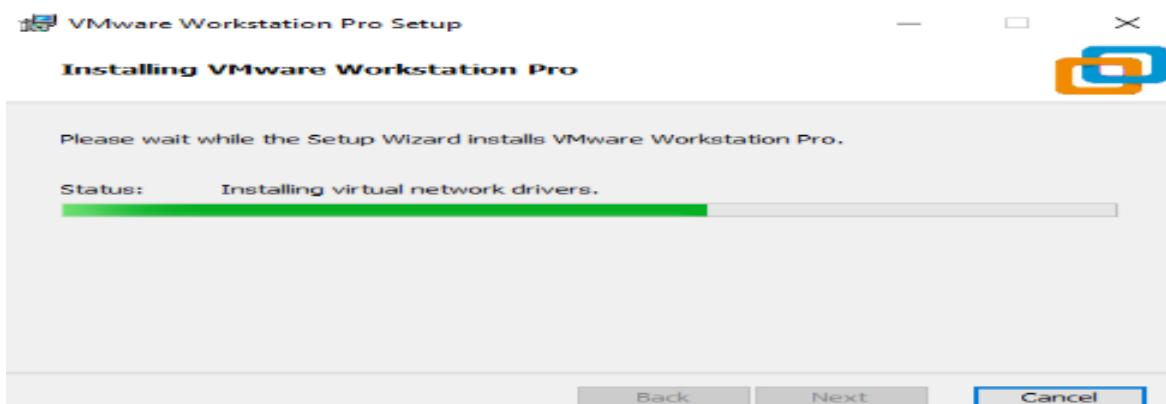


Figure 44, Installing VMware WorkStation Pro 15

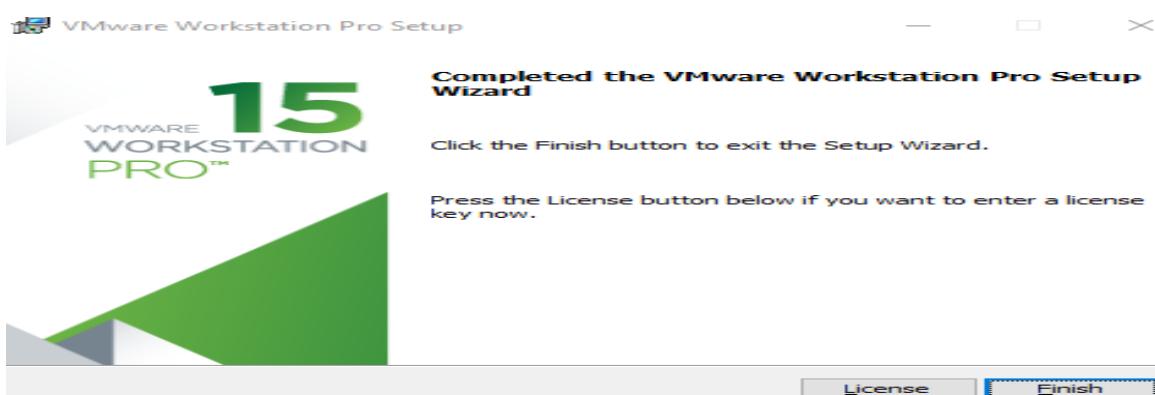


Figure 45, Installing VMware WorkStation Pro 15

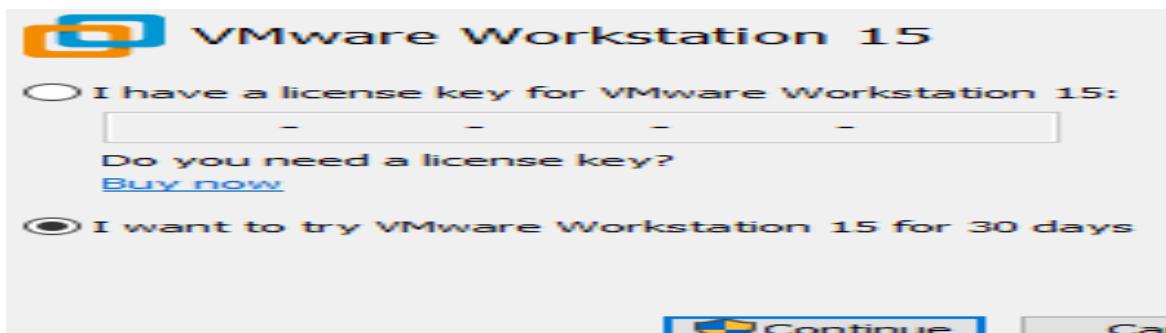


Figure 46, Installing VMware WorkStation Pro 15

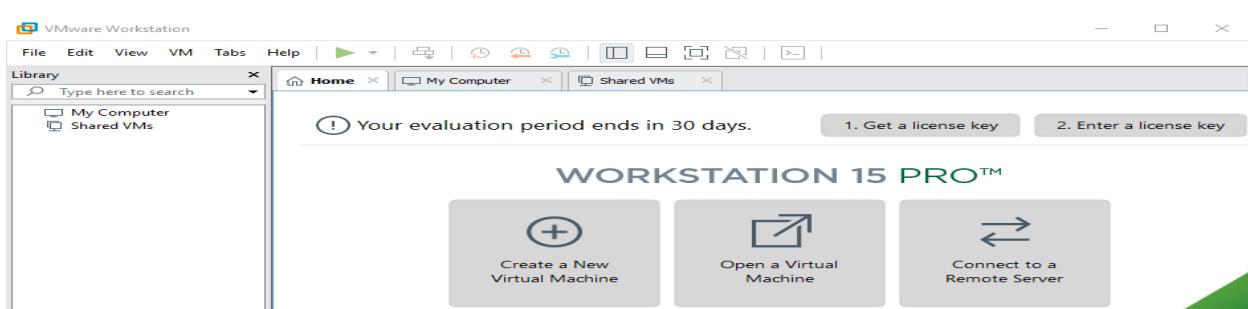


Figure 47, VMware WorkStation 15 PRO, first look after installation.

4.4 Creating VMs and Installing Windows Server (Winserver)

Now the 'VMware Workstation 15 PRO' has been set from where installing VMs is very easy, however it can be easily done from web login to HTML5 client on the host, too. For the purpose of this project, VMs with different operating systems e.g. Windows, Ubuntu Linux, CentOS have been installed. Here are the processes for a complete VM installation for Windows Server 2016 using HTML5 based interface as below:

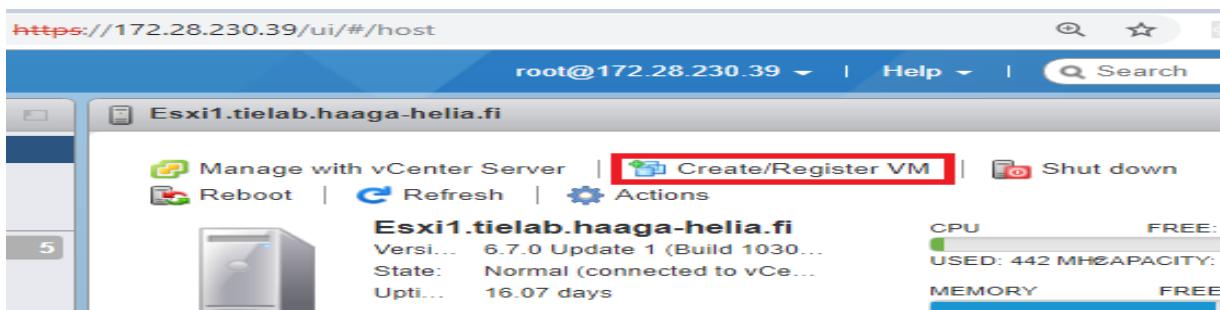


Figure 48, Creating VMs and Winserver in HTML5 GUI 1

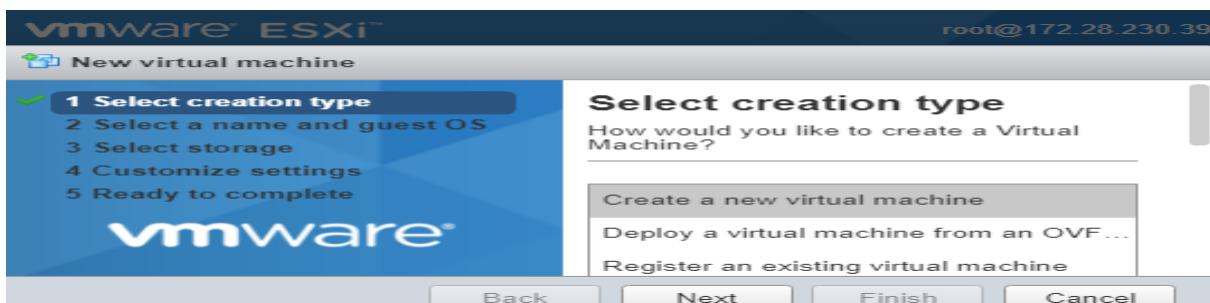


Figure 49, Creating VM & Winserver in HTML5 GUI 2

For our case, we don't have any Virtual machine template or OVF/OVA downloaded or saved file in our datacentre, so we choose to create a new virtual machine. But it is also possible to create VMs from template or OVF/OVA file from the data centre if we already have saved VMs with required configuration.

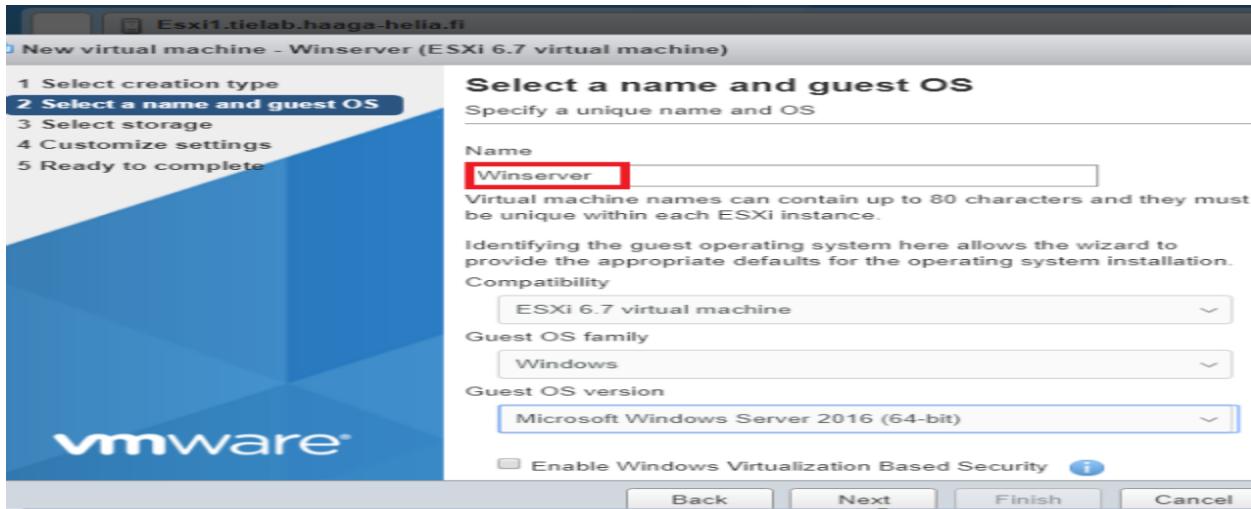


Figure 50, Creating VM & Winserver in HTML5 GUI 3

In figure 50, purposeful name for the VM along with guest OS compatibility and versions are selected.

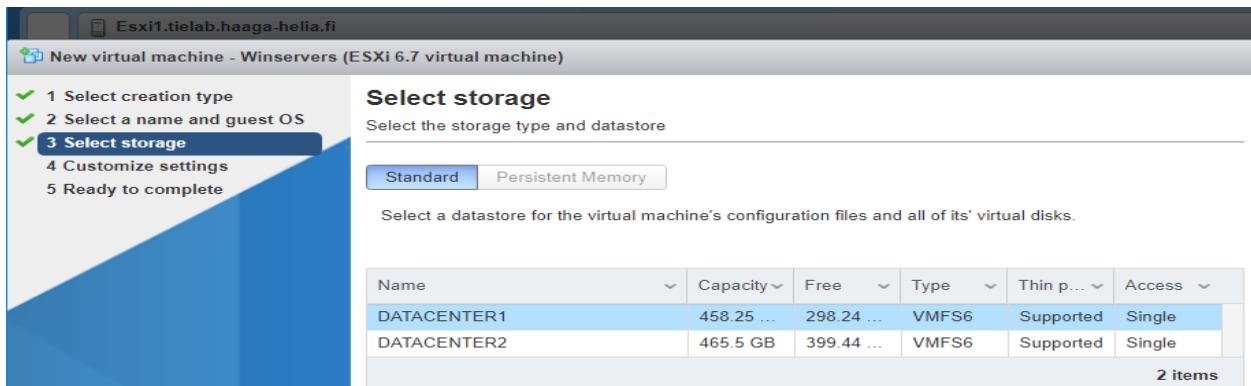


Figure 51, Creating VM & Winserver in HTML5 GUI 4

In figure 51, the datastore for the VMs is selected. In this case, there are only two data centres- DATACENTER1 & DATACENTER2, where the first one is selected.

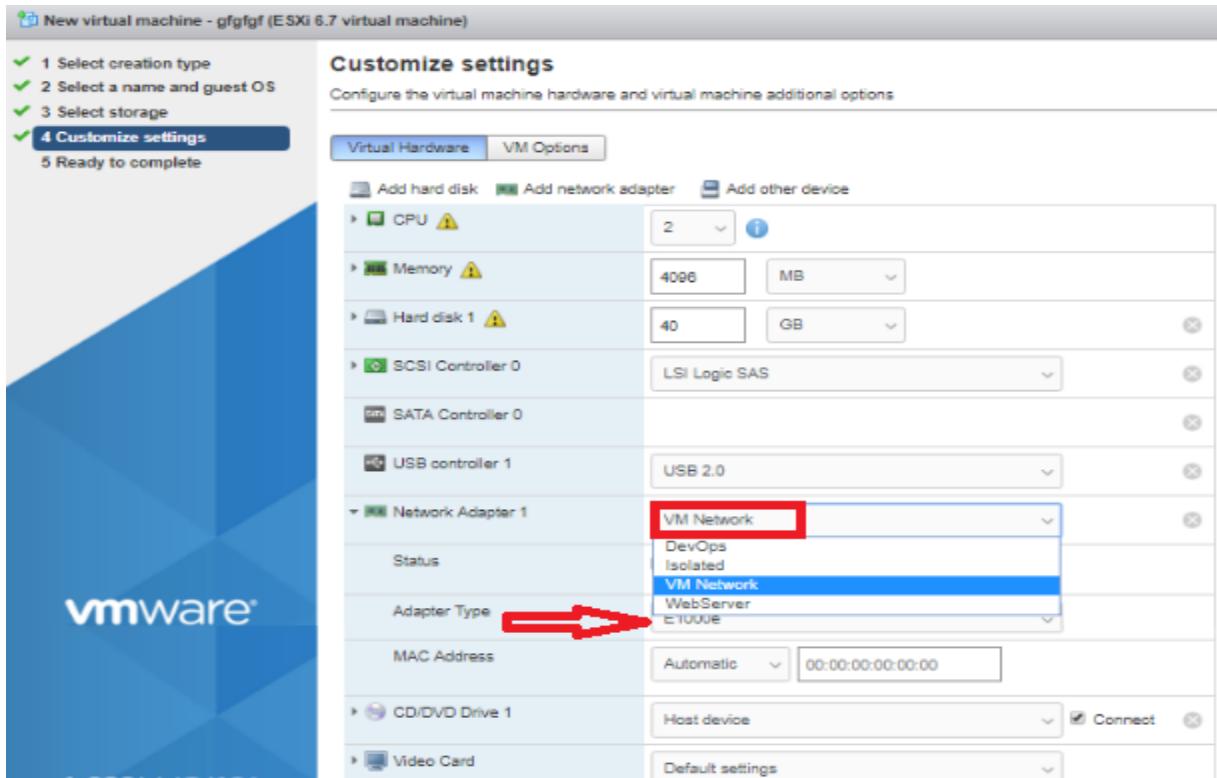


Figure 52, Creating VM & Winserver in HTML5 GUI 5

In **figure 52**, it is important to make suitable network adapter selection because it determines this VMs's placement and its network configuration. By default, it takes the VM network adaptor, which is opted in this project, too. However, we have created a few different proposed networks, e.g.- DevOps, Isolated, WebServer about which will be discussed more in the port and switch configuration section.

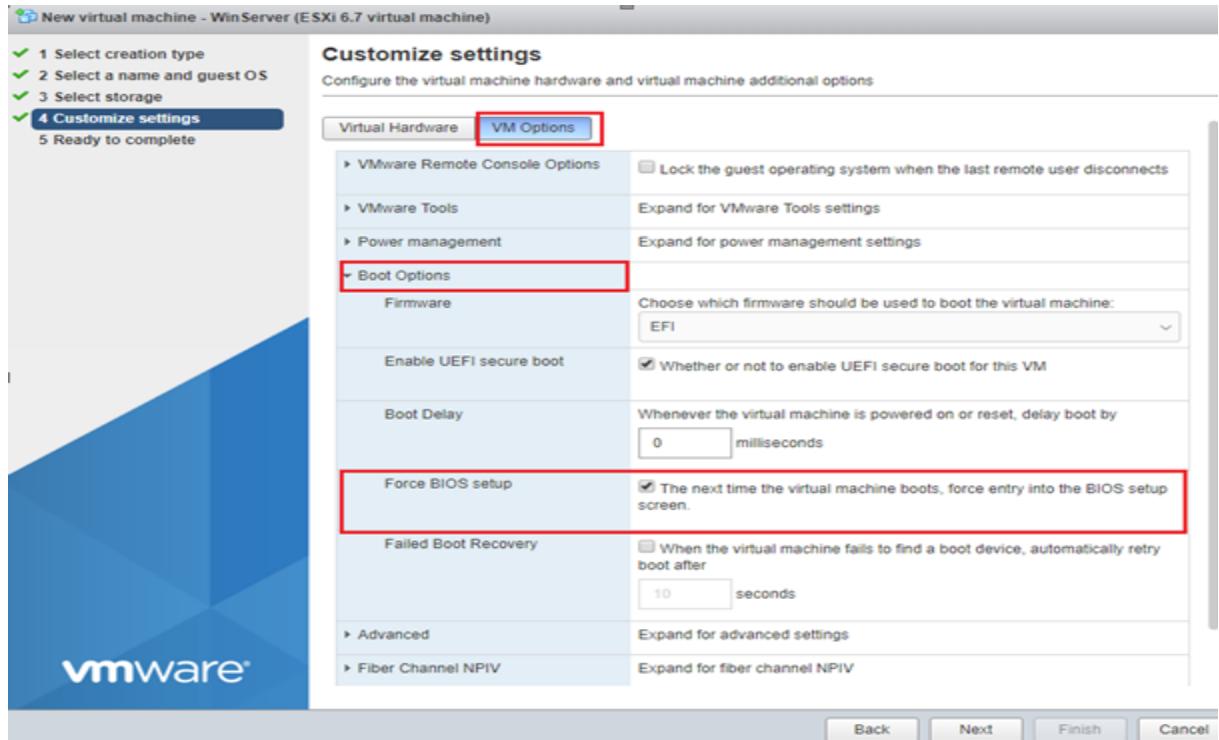


Figure 53, Creating VM & Winserver in HTML5 GUI 6

In figure 53, it creates BIOS setup for booting when the VM is powered on next time for booting the iso image in the boot manager (figure 55 and 56).

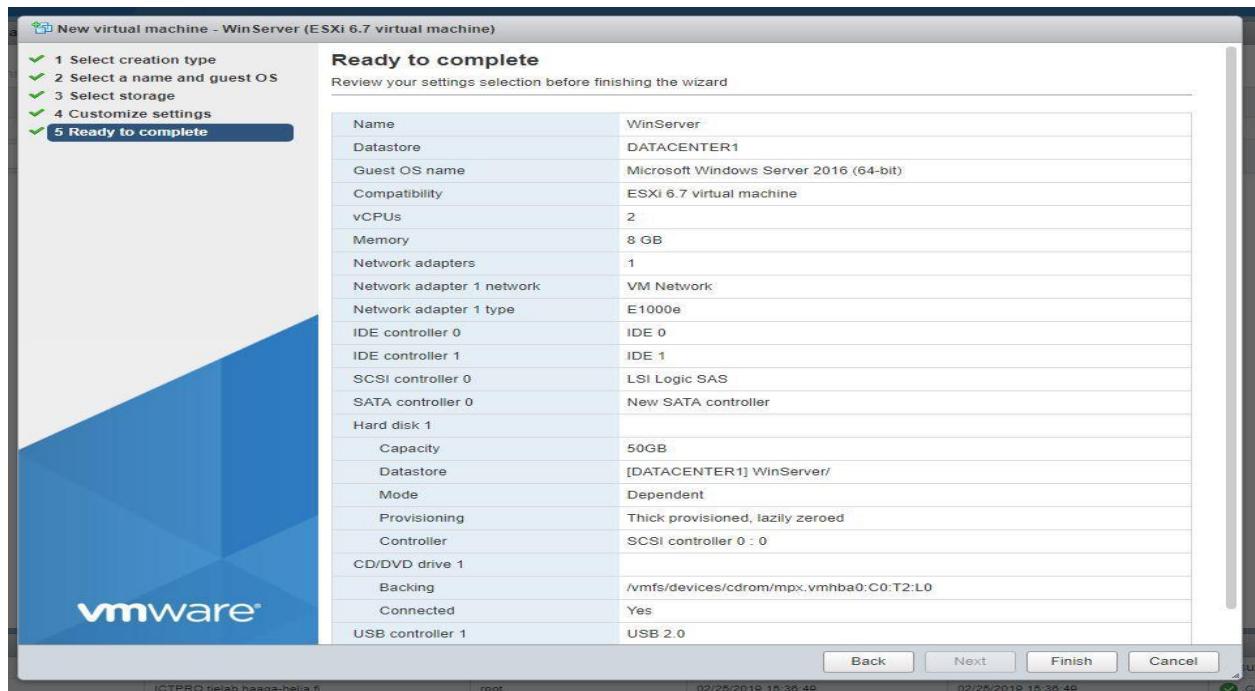


Figure 54, Creating VM & Winserver in HTML5 GUI 7

In **figure 55**, power on the machine and open the instance in Web-console or VMware Remote Console. Web-console does not need to be downloaded but VMware Remote Console.

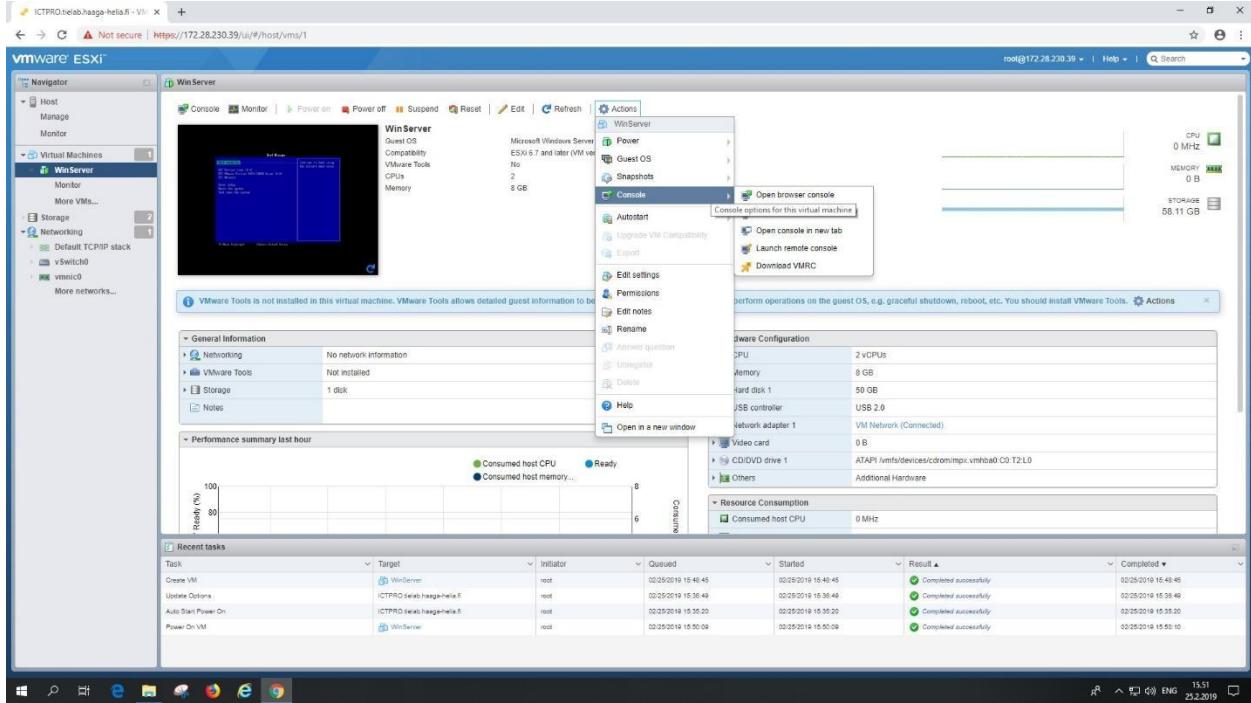


Figure 55, Creating VM & Winserver in HTML5 GUI, from Web-console.

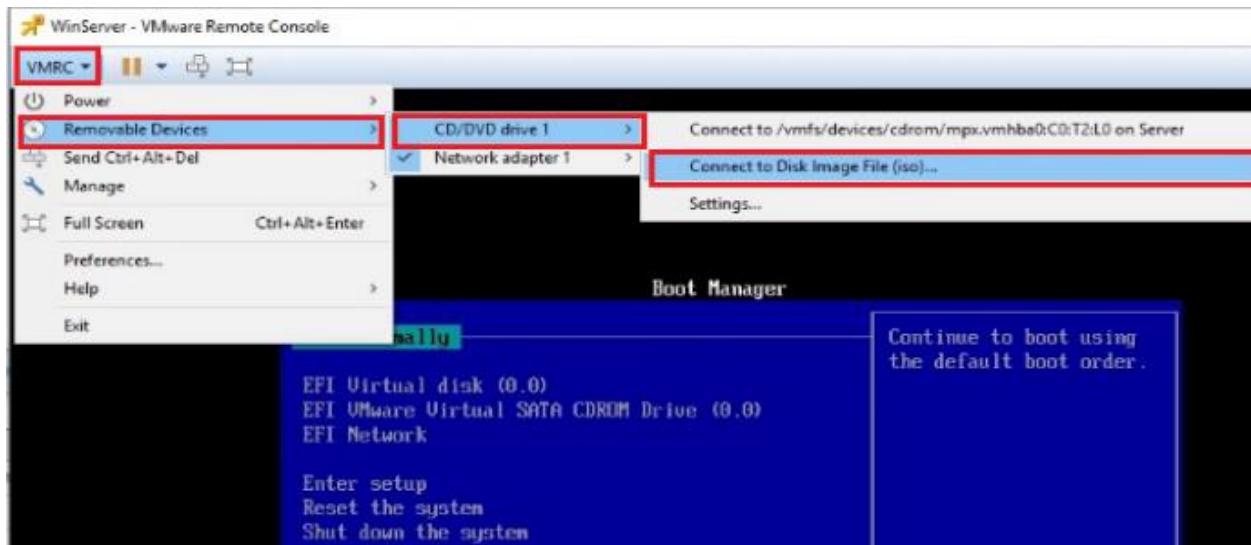


Figure 56, Creating VM and Winserver in HTML5 GUI, from Web-console.

Connect to Disk Image File(iso) for Windows Server Installation. This will take the installation to the Boot Manager from where the 'EFI VMware Virtual SATA CDROM Drive (0.0)' can be browsed with upward or downward arrow and pressing 'Enter' button as in **figure 57**.

The above booting can be carried out in VMware WorkStation, too by pressing ‘CD/DVD (SATA)’, this example is displayed in **figure 58** and **59** only.

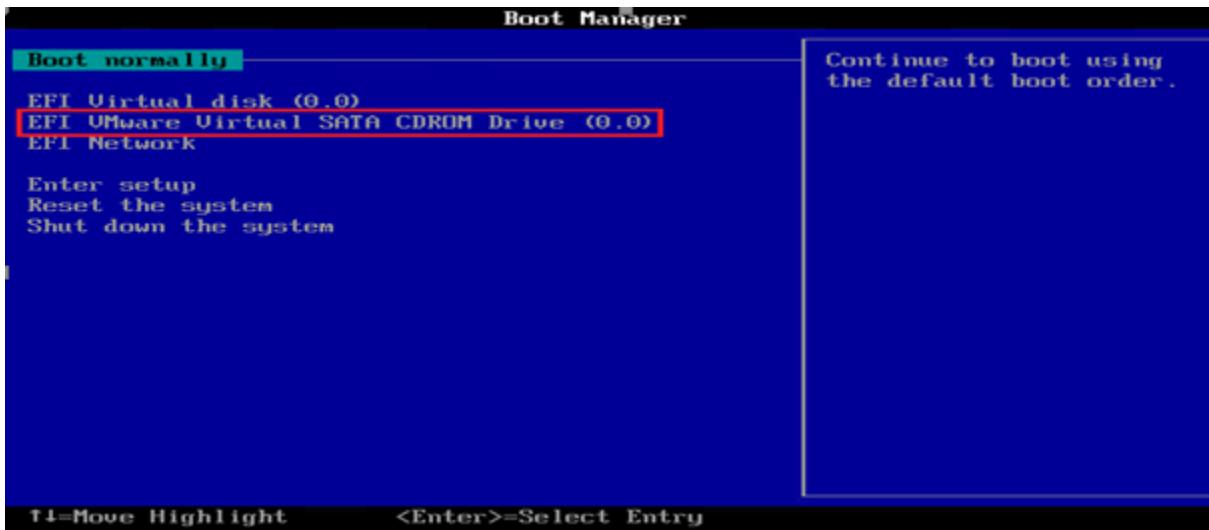


Figure 57, Creating VM and Installing Winserver in HTML5 GUI, from Web-console.



Figure 58, Creating VM and Installing Winserver in VMware WorkStation

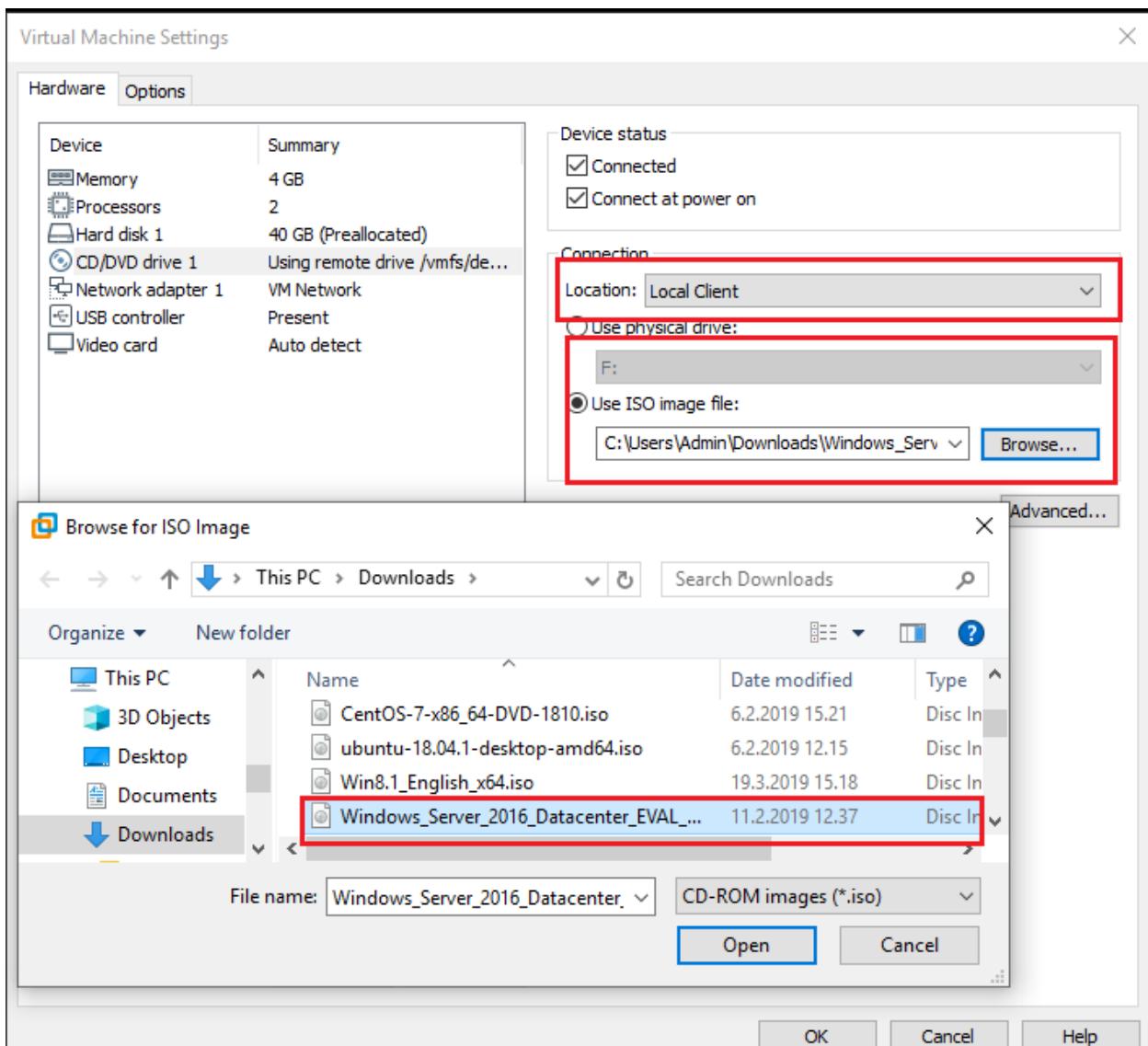


Figure 59, Creating VM and Installing Winserver in VMware WorkStation

We recommend using VMware Workstation for the installation because it provides connection options from Remote client or Local client location. In this case, the ISO image is downloaded and placed in the local client storage, so this option has been chosen. Once ISO image file is browsed, it leads to the ISO image file location as in the above **figure 59**. Then just click ok. Choosing GUI version of Windows Server 2016 is important (**figure 60**) because it is easier to use than the console or command prompt version.

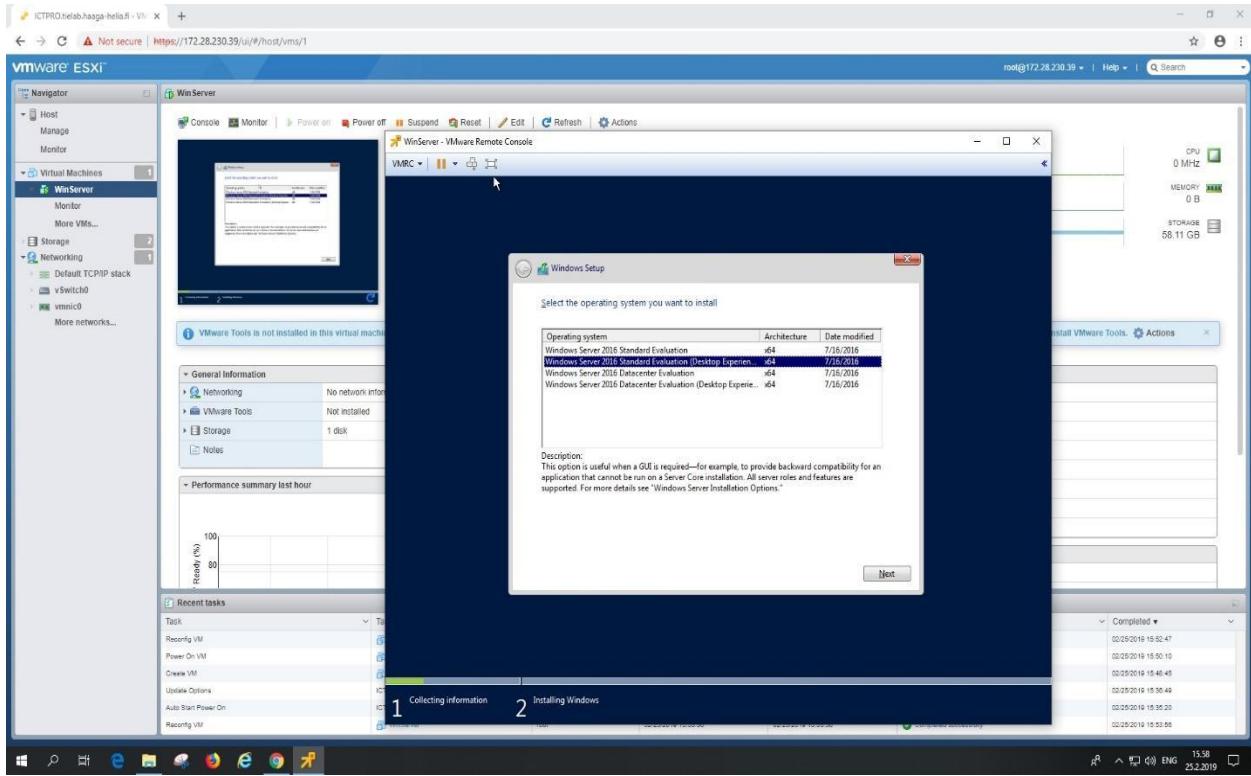


Figure 60, Creating VM & Winserver in HTML5 GUI 10

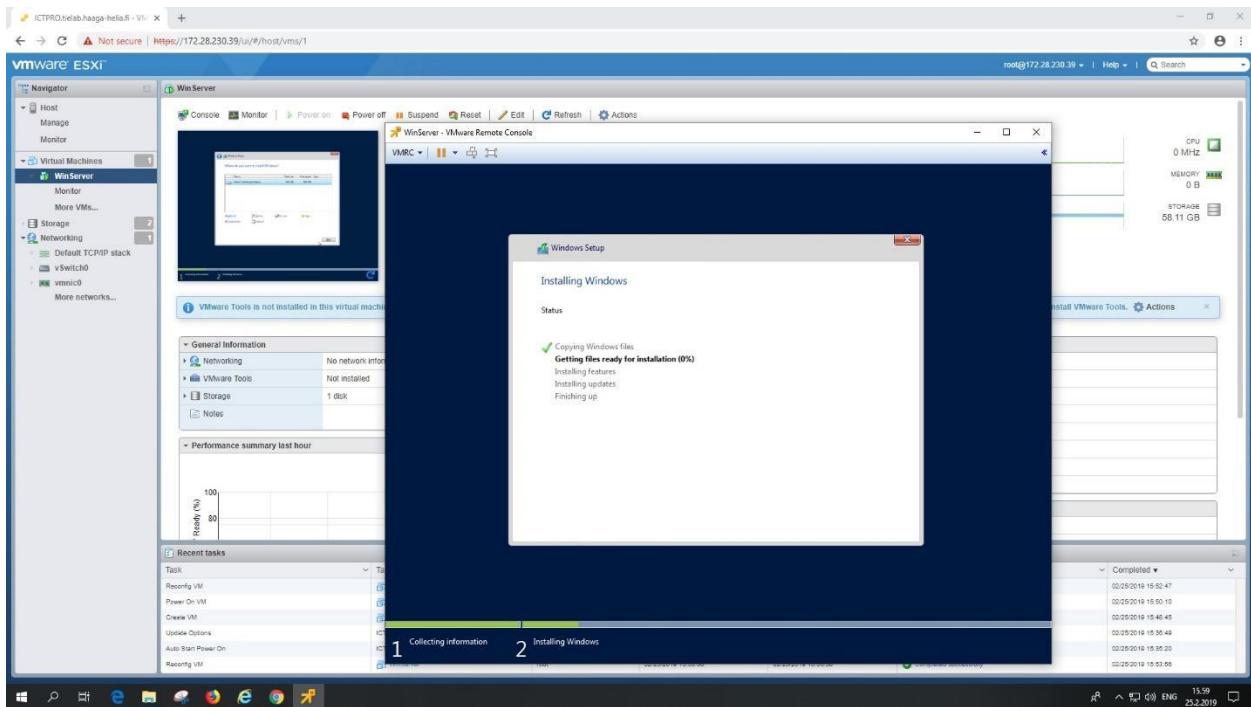


Figure 61, Creating VMs and Installing Winserver in HTML5 GUI 11

4.5 Installing Windows Server 2016 As Routing Server (WinRouter)

The same installation process, as explained in the previous sections ‘Creating VMs and Installing Windows Server’, can be followed here for Routing Server (WinRouter) installation, too. For this, Microsoft Windows Server 2016 64 bits is used. In this project, WinRouter is required for creating/configuring Private (IPv4: **192.168.0.1**) and Public (IPv4: **172.28.230.36**) networks, which also means WinRouter has two IPv4 addresses as mentioned above.

5 Editing and Deleting VMs

It is also important to know how to delete and configure VMs after creating them because we need many changes in times. To delete a VM, it is advisable to power off the machine first. It can be deleted permanently only from the host or vSphere Client level otherwise it will remain in the inventory of the datacentre. Therefore, deletion can be made directly from the inventory level, however minor changes can be performed from the VMware Workstation Pro.

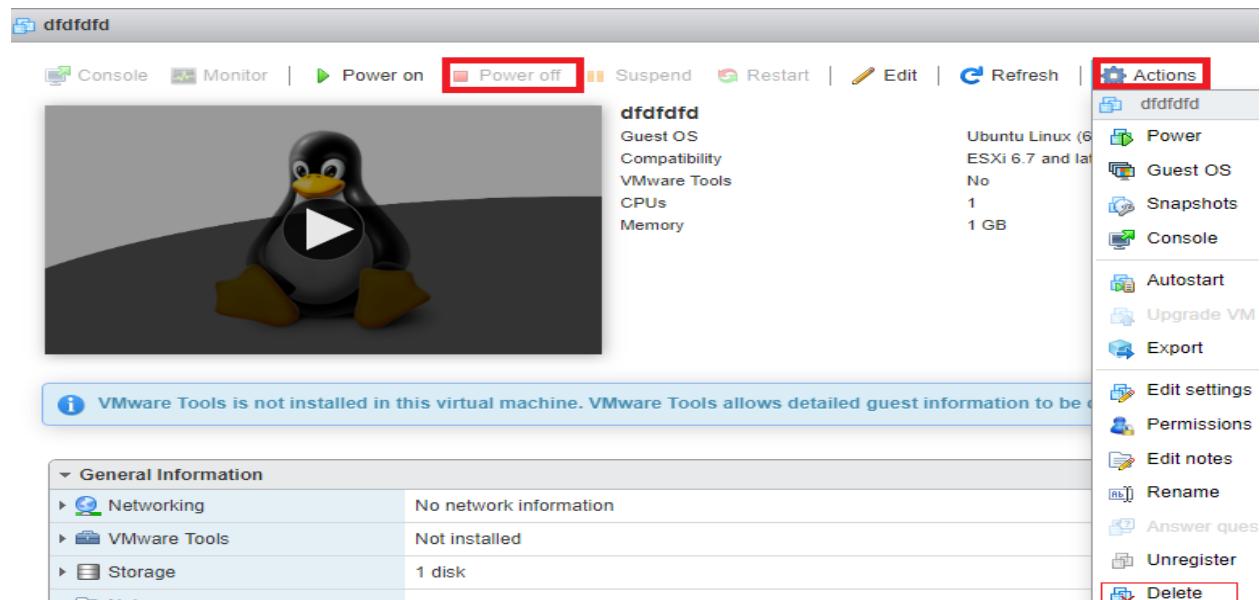


Figure 62, Power off Actions and Delete.

6 Configuration of the Required Tools

This section will cover all about the configurations of the required tools as mentioned in the heading **3** above.

6.1 Host Esxi 6.7 Configuration

Host configuration can be done from the vSphere Web client, vCenter Server, Web interface for ESXi or through SSH connection where, in this project, most of the configuration are made in web interface which is written in HTML5 and JavaScript, for which nothing needs to be installed, once the host is installed then just opening the IP address of the host as web URL (**figure 61**) and supplying log in credentials will open the interface.

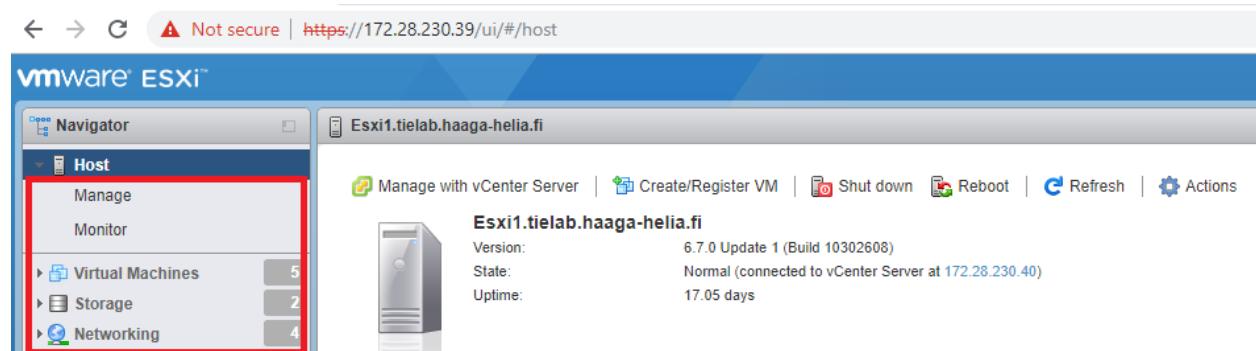
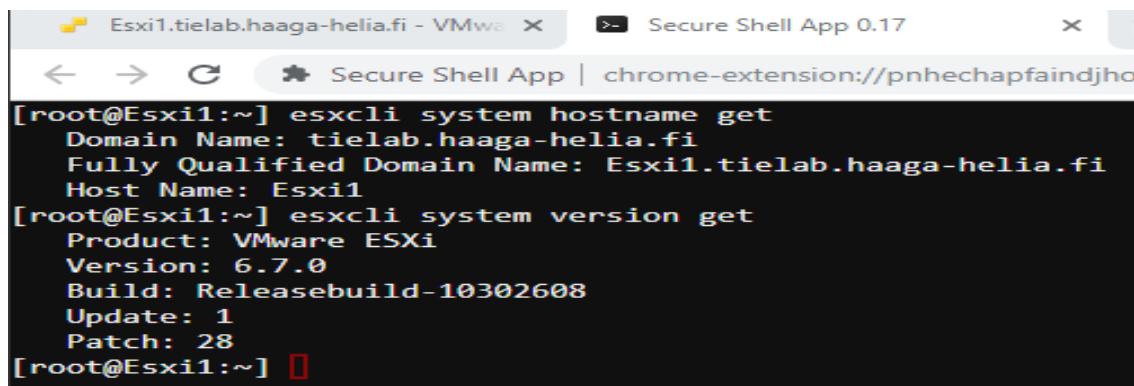


Figure 63, Host ESXi 6.7 host

6.1.1 Manage, Monitor Host & Create VMs

- Creating a new VMs from the scratch, with required customizations on processors, memory, network connections and storage, see **point 4.4** above (Creating VMs and Installing Windows Server) or deploying an already defined VMs from OVF/VMDK files or registering a VM that already exists in the datastore (**figure 49**).
- Reboot, refresh, shut down or perform other actions on the host e.g.- connect the host with or disconnect from the vCenter Server, disable or enable SSH connection (e.g.- for managing the host from PuTTY, however unless it is necessary for administrative purpose, it is recommended to disable the SSH connection also the console

shell), place Esxi in maintenance mode when needed for its servicing e.g.- adding more memory. For this purpose, all the VMs must be migrated to another host to run them there, if another host exists, otherwise the machines must be powered off for entering the host to maintenance mode. It has also the option for accessing the host with Direct Console User Interface (DCUI) through SSH Console which is easy to start in the web browser and good tool for troubleshooting or accessing information e.g.- **figure- 64** below:



```
Esxi1.tielab.haaga-helia.fi - VMwa Secure Shell App 0.17
Secure Shell App | chrome-extension://pnhechafaindjho
[root@Esxi1:~] esxcli system hostname get
  Domain Name: tielab.haaga-helia.fi
  Fully Qualified Domain Name: Esxi1.tielab.haaga-helia.fi
  Host Name: Esxi1
[root@Esxi1:~] esxcli system version get
  Product: VMware ESXi
  Version: 6.7.0
  Build: Releasebuild-10302608
  Update: 1
  Patch: 28
[root@Esxi1:~] 
```

Figure 64, SSH Console to DCUI

However, SSH console for network configuration, changing DNS Servers, Gateway, IPv4, etc. can be costly, if it goes wrong.

- Options to open any VM in browser console, new window console, new window in the same tab, remote console or download VMRC.

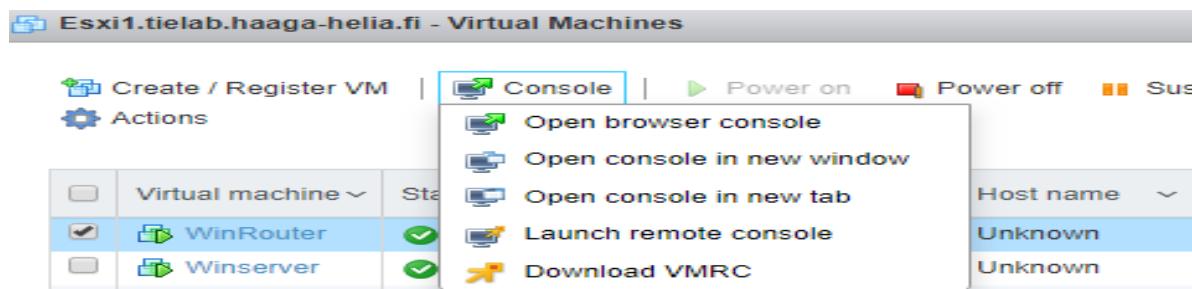


Figure 65, console options for VMs

- Contains system, hardware setting, license management, packages, install updates, security and users with roles, permissions (**figure 67**) set firewall rules etc. and all these recent activities can be viewed on the same page.

6.1.2 Storage

ESXi datastores are meant for storing virtual machines files e.g.- virtual machine templates, floppy images and ISO images. Virtual Machine File System (VMFS) and Network File Systems (NFS) are two main types of file systems used to package the VMs into the store. Storage can be residing within the same host where the host's physical store is mapped by the logical store and partition is possible according to different needs or usages. In NFS type, the storage is located in the NFS server, however this project deals with VMFS datastores only.

In this section, new datastores can be created, deleted, different users can be given different access permissions to stores. In the **figure 67**, the user is given access to DATACENTER2 but not DATACENTER1. By browsing through the datastore, we can see what are different datastores, which are the available VM, creating more directories to store VMs or ISO images, etc. as in **figure 66**.

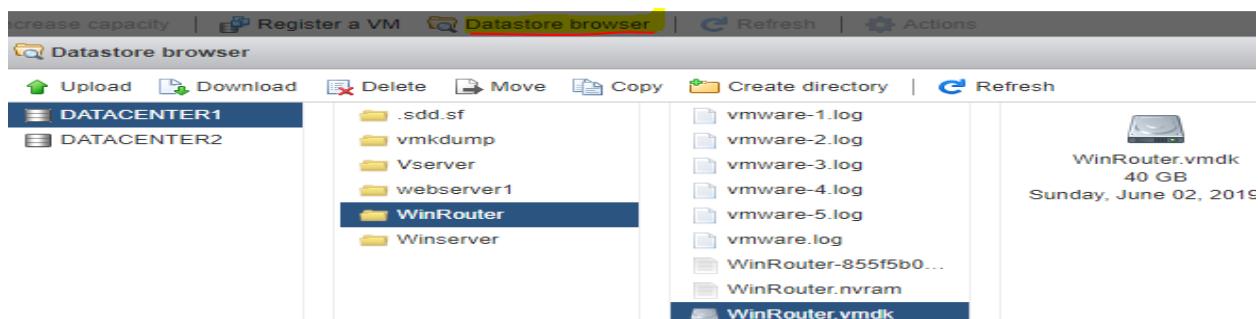


Figure 66, Datastore browse

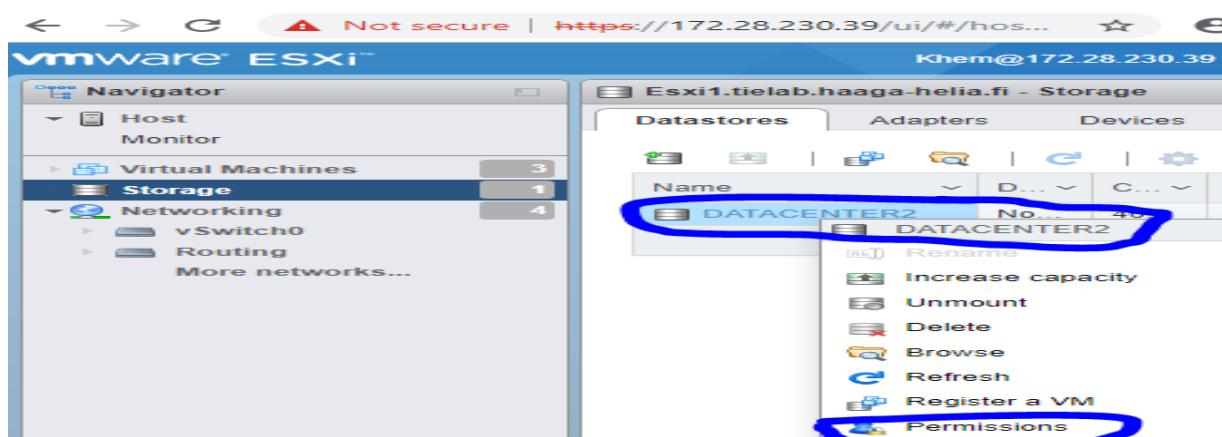


Figure 67, Not the root user, no access to DATACENTER1

6.1.3 Networking

This part covers how to add Standard Virtual Switches, how uplinks/physical adapters are added to them, how port groups are added, and virtual switches are added to them.

6.1.3.1 Creating vSwitches

Go to Networking => Virtual switches => Add standard virtual switch => vSwitch Name

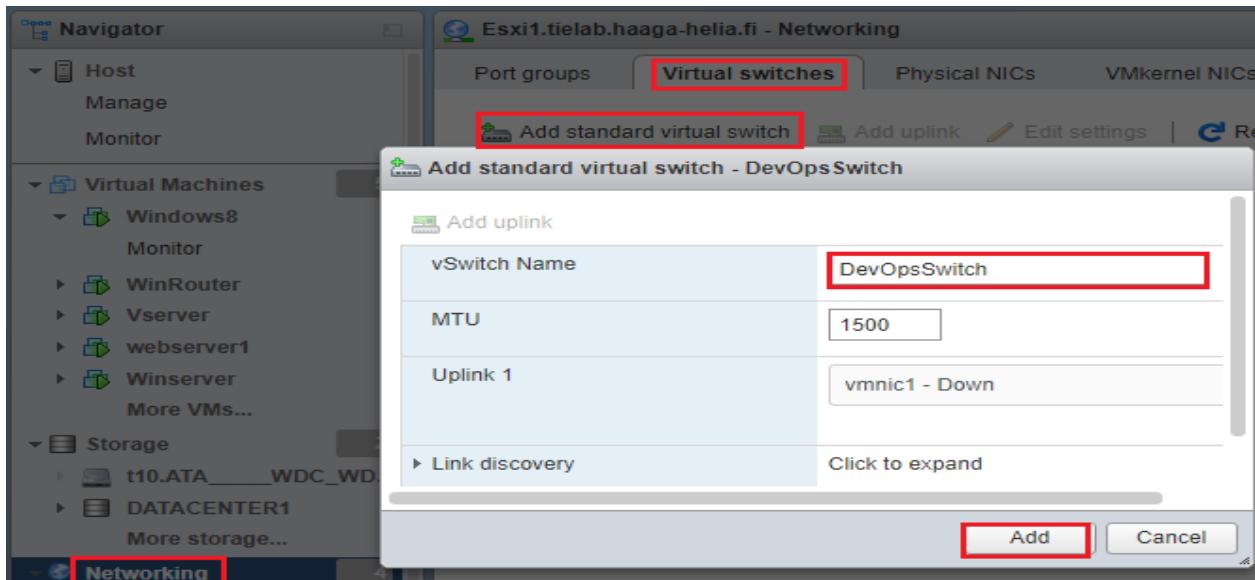


Figure 68, Creating virtual switch

6.1.3.2 Creating Port Group

For creating port groups from the host's web interface, go to Networking option and then Add port group, then fill the all information about Name of the port group, VLAN ID if needed. Choose appropriate Virtual Switch, assign to a port group then Add. This will create a new port group in the network environment.

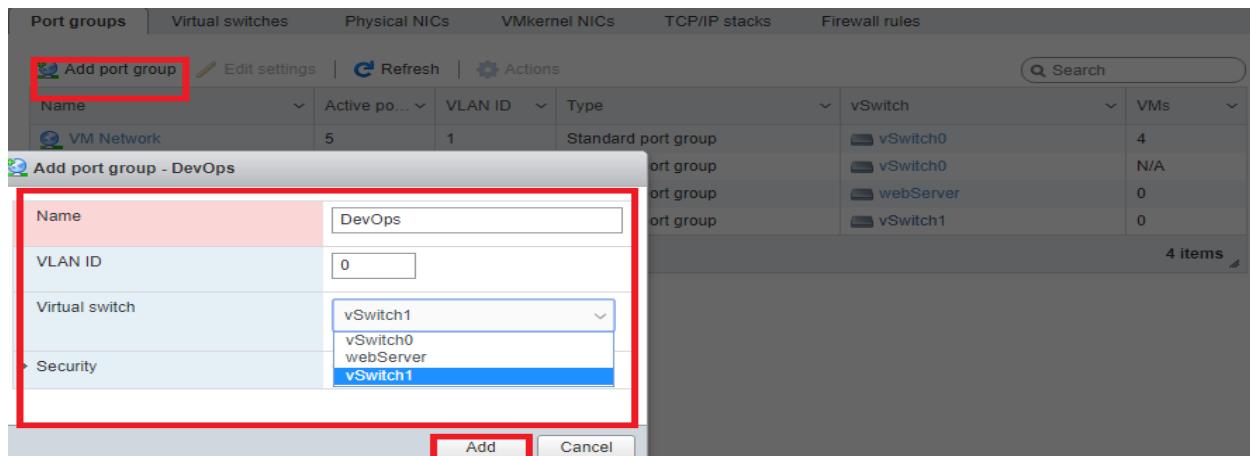


Figure 69, Creating virtual switch

6.1.3.3 Firewall Rules from ESXi Web Interface

VMware host Esxi 6.7 has a great feature about the security concern which is firewall. In the host level, it has own Firewall rules setting, from where the host's network can be secured. Under this option, we can configure about the connectivity from the outside and inside to the network.

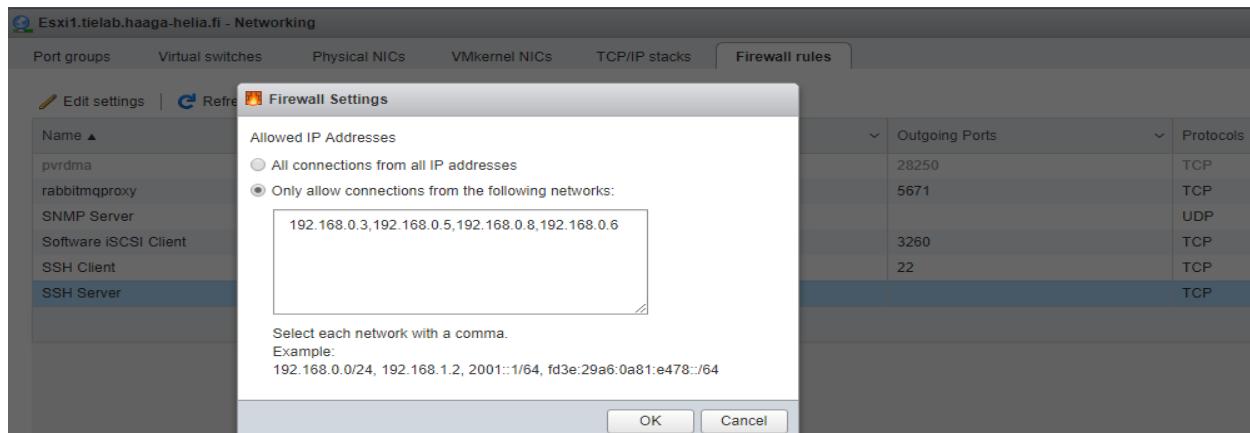


Figure 70, Firewall rules for different services

6.2 vCenter Server Appliance

After its installation, we will come across two interfaces: **VMware Appliance Management** from <https://172.28.230.40:5480> and **vSphere Client** from <https://172.28.230.40:443>.

6.2.1 VMware Appliance Management

From Appliance Management user interface, we can view the summary health status and single sign-on domain and status of the appliance, monitor CPU and Memory in different point of time, edit access methods (SSH login, DCLI, Console CLI Bash Shell), edit network settings (**figure 71**), firewall settings, different services and updates (automatic or time defined), create schedule backup and so on.

The screenshot shows the VMware Appliance Management web interface. The left sidebar has a navigation menu with options: Summary, Monitor, Access, Networking (which is selected and highlighted in blue), Firewall, Time, Services, Update, Administration, and Syslog. The main content area is titled "Network Settings". It displays the following information:

Hostname	Vserver.CLOUDTECH.TEST
DNS Servers	172.28.230.38
NIC 0	
Status	Up
MAC Address	00:0C:29:6E:98:23
IPv4 Address	172.28.230.40 / 16 (Static)
IPv4 Default Gateway	172.28.1.254

Below this, there is a "Proxy Settings" section with three entries:

HTTPS	Disabled
HTTP	Disabled
FTP	Disabled

Figure 71, VMware Appliance Management web interface

6.2.2 vSphere Client

As mentioned earlier, this vCenter Server centralizes configuration of the multiple ESXi hosts along with datacentre creation, creating host and cluster, networking etc. (**figure 72**). These configurations, however, can be achieved from the host's web interface also, see **heading 6.1- 'Host ESXi 6.7 Configuration'**.

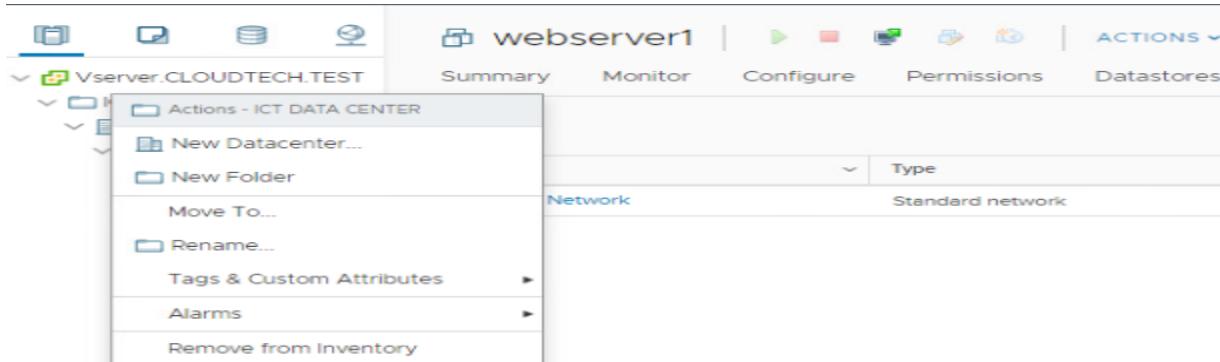


Figure 72, Adding new datacentre, renaming

Also creating templates and VMs with completely new configurations or by cloning another VM or by converting template to VM, removing any installed services from inventory level etc. are possible from vSphere Client interface (**Figure 73**).

New Virtual Machine

1 Select a creation type

2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete

Select a creation type
How would you like to create a virtual machine?

Create a new virtual machine

- Deploy from template
- Clone an existing virtual machine
- Clone virtual machine to template
- Clone template to template
- Convert template to virtual machine

This option guides you through creating a new virtual machine. You will be able to customize processors, memory, network connections, and storage. You will need to install a guest operating system after creation.

Figure 73, vSphere Client, creating, cloning, templating VMs

Creating new cluster, distributed switches or importing switches, distributed port groups, migrate virtual machines to another networks, etc. (**figure 74**).

Migrate VMs to Another Network

1 Select source and destination...
2 Select VMs to migrate
3 Ready to complete

Select source and destination networks
Select source and destination networks for the migration of virtual machine network adapters

Source network

Specific network

DevOps BROWSE ...

All virtual machine network adapters that are connected to this network will be migrated.

No network

All virtual machine network adapters that are not connected to any network will be migrated.

Destination network

WebServer BROWSE ...

Figure 74, Migrating VMs from DevOps network to WebServer network

This server can be joined to the Winserver's domain 'CLOUDTECH.TEST (figure 75). For this, go to Menu => Configuration => Active Directory Domain => Join AD. From here, also the password policies can be configured, identity sources (**vsphere.local** or **localos**), login message etc. can be customized and so on.

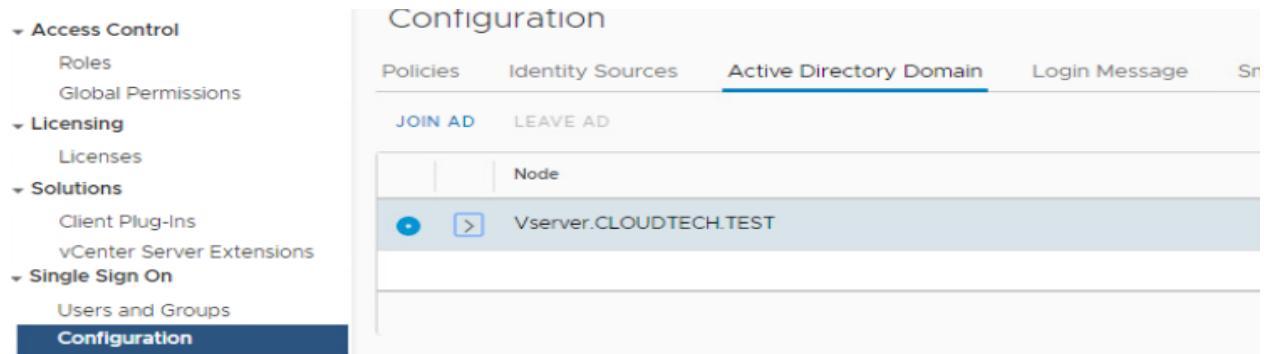
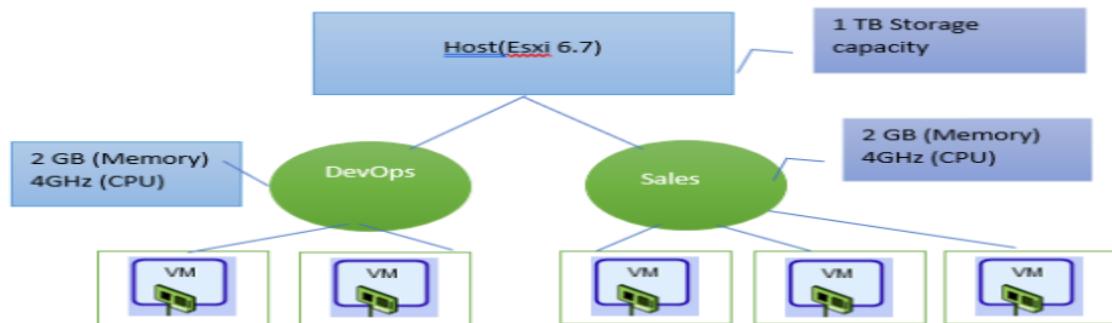


Figure 75, Joining the active directory domain.

Resource Pool also can be managed from vSphere Client. Resource pool is a mechanism that allows to delegate control over the host or cluster, that is the separation of resources from hardware or logical abstraction of the resources. Creating resources needs to connect to the host then right click on host => New Resource Pool then provide suitable name. For production environment higher priority share can be set whereas for normal test environment, normal share.



6.3 Winserver Configuration

The main target here is to establish this server also as a **DNS Server** and as **Active Directory Domain Controller**. All the steps required to meet these purposes are explained in points below.

6.3.1 Network Connection Setting

The first setting that can be started with is the DNS setting in the network interface where, instead of obtaining the IP automatically, the following IP address and DNS Server Address are used:



Figure 76, IPv4 properties

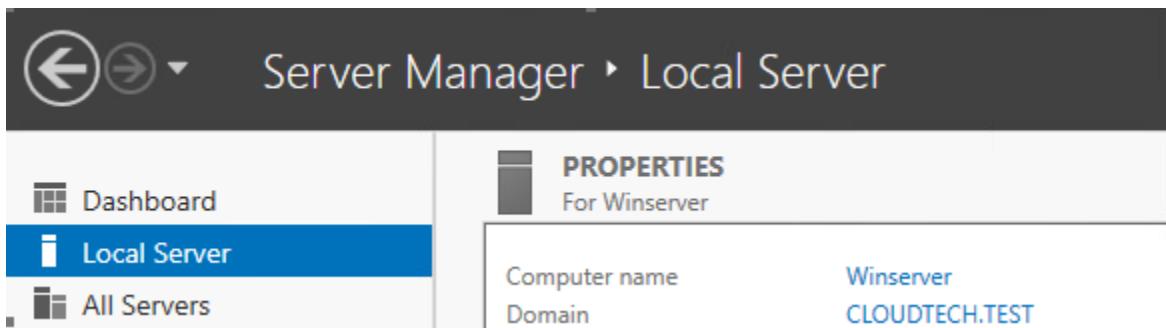


Figure 76, Computer name and domain name of Winserver

6.3.2 AD Domain Services Installation

Immediately after the Winserver installation, provide a meaningful computer name as 'Winserver'. As this server also works as Domain Controller, Active Directory Domain Services need to be installed as in the following instruction.

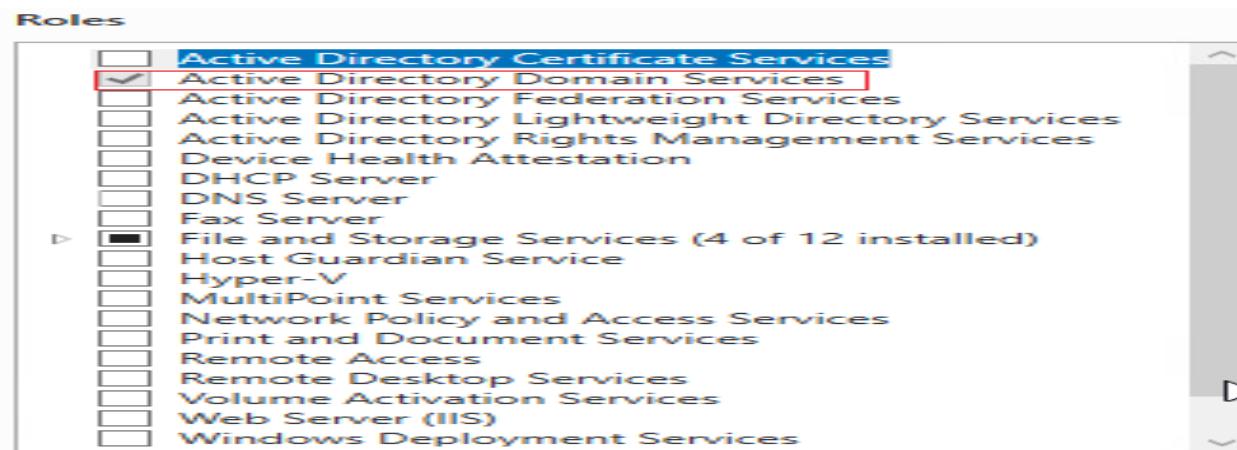


Figure 77, AD DC installation in Winserver

As in the **figure 77** above, go to Server Dashboard => manager =>Add roles and features =>Before You Begin =>Role-based or feature-based installation=> Server Selection from Server pool => Active Directory Domain Services => Add features and until the install appears, just keep clicking Next .

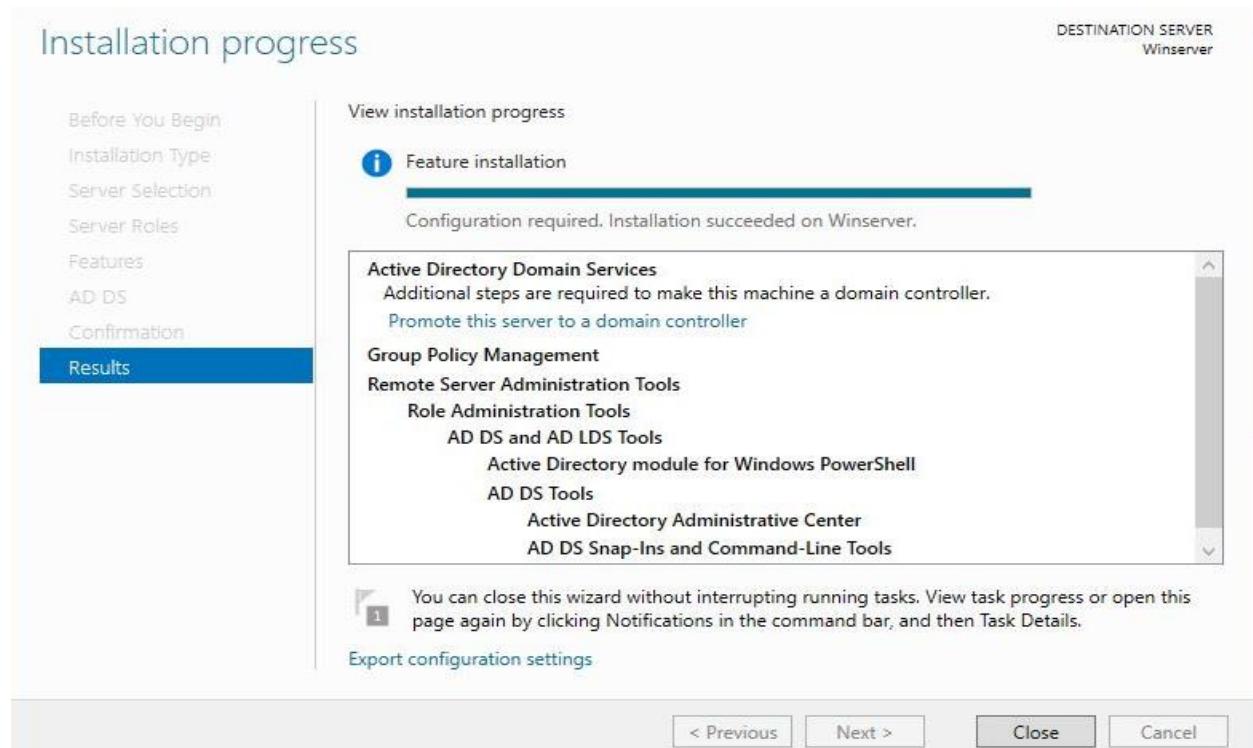


Figure 78, AD DC installation complete, promote to a domain controller option available

After the installation, promote this machine as domain controller which is a post-deployment configuration.

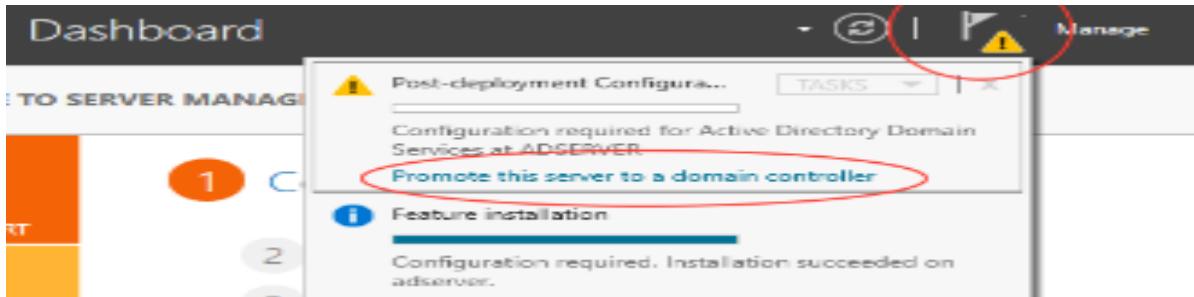


Figure 79, Promoting to domain controller

Now the installation has completed, it is time to configure for deployment. Steps can be followed as below:

Add a new forest, type a meaningful domain (CLOUDTECH.TEST) => Supply Password => DNS options then NetBIOS domain name (CLOUDTECH) appears automatically click Additional Options =>prerequisites check => then install.

Now that the forest and domains are created in AD DC, it can be checked in the PowerShell:

A screenshot of a Windows PowerShell window titled 'Winserver'. It shows two command outputs. The first command, 'Get-ADForest | fl Name,ForestMode', returns a single result: Name : CLOUDTECH.TEST ForestMode : Windows2016Forest. The second command, 'Get-ADDomain | fl Name,DomainMode', also returns a single result: Name : CLOUDTECH DomainMode : Windows2016Domain.

Figure 80, Checking forest and domains with PowerShell command

6.3.3 DNS Management

After the roles and features are installed, we do DNS configuration that includes forward and reverse lookup zones.

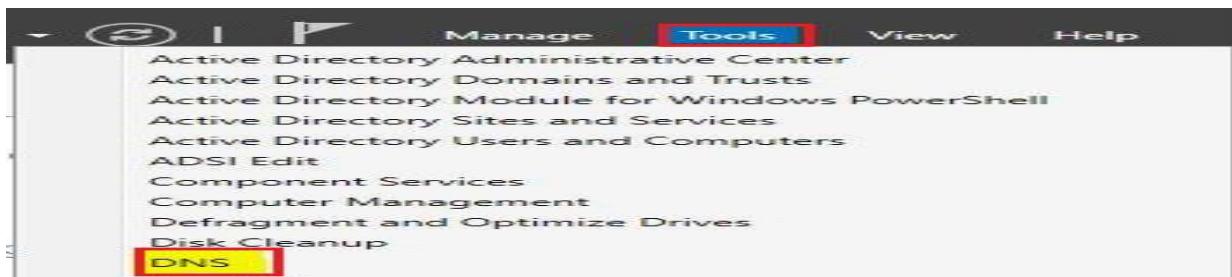


Figure 81, Configuring DNS server

6.3.4 Forward Lookup Zones

Before creating Forward and Reverse lookup Zones already registered forwarders should be deleted. In this way:

In the Server Manager, go to tools => DNS => WINSERVER => properties => forwarders => edit => select to delete

Select Forward Lookup Zones => CLOUDTECH.TEST right click => New Host (A or AAAA) ... then in the name space, provide a name for the machine to register along with its IPv4 address => check the PTR => Add Host as in the **figure** below:

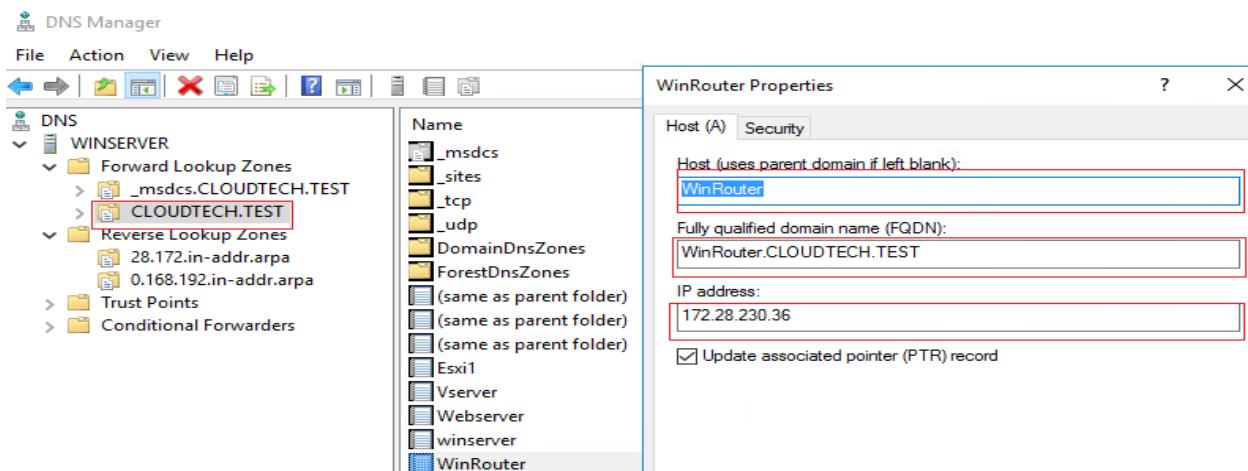
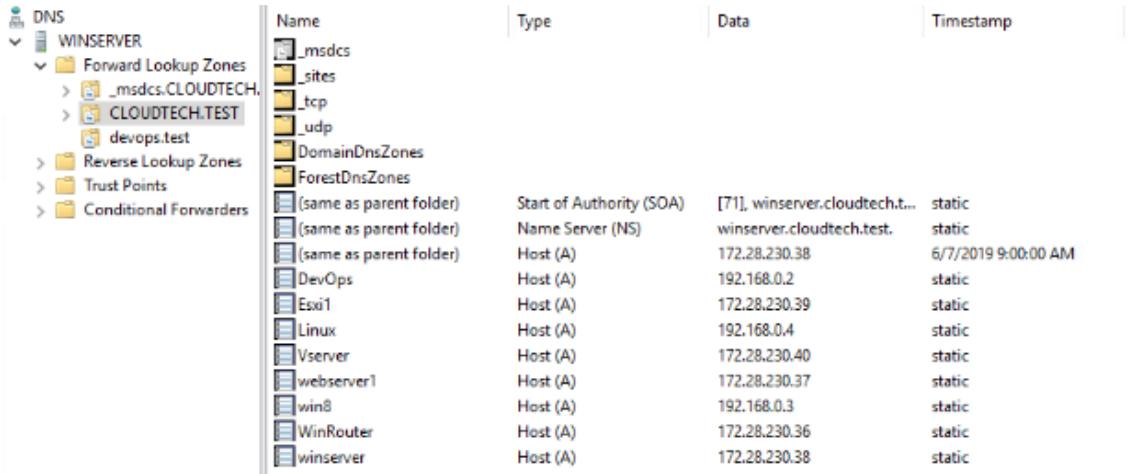


Figure 82, Forward reverse lookup zones 1

In the same way, all the machines should be registered here, then finally the DNS manager will look like the **figure 83** below:



The screenshot shows the Windows DNS Manager interface. On the left, a tree view displays the DNS structure under 'WINSERVER'. Under 'Forward Lookup Zones', there are entries for '_msdc', '_sites', '_tcp', '_udp', 'CLOUDTECH.TEST' (selected), 'devops.test', 'Reverse Lookup Zones', 'Trust Points', and 'Conditional Forwarders'. The 'CLOUDTECH.TEST' zone is expanded, showing its sub-records. On the right, a table lists these records with columns for Name, Type, Data, and Timestamp.

Name	Type	Data	Timestamp
_msdc	Start of Authority (SOA)	[71], winserver.cloudtech.t...	static
_sites	Name Server (NS)	winserver.cloudtech.test.	static
_tcp	(same as parent folder)	Host (A)	172.28.230.38
_udp	(same as parent folder)	Host (A)	192.168.0.2
DomainDnsZones	(same as parent folder)	Host (A)	172.28.230.39
ForestDnsZones	(same as parent folder)	Host (A)	192.168.0.4
(same as parent folder)	Host (A)	172.28.230.40	static
DevOps	Host (A)	172.28.230.37	static
Esxi1	Host (A)	192.168.0.3	static
Linux	Host (A)	172.28.230.36	static
Vserver	Host (A)	172.28.230.38	static
webserver1	Host (A)	192.168.0.3	static
win8	Host (A)	172.28.230.36	static
WinRouter	Host (A)	172.28.230.38	static
winserver	Host (A)	192.168.0.4	static

Figure 83, Final look up of the DNS manager

6.3.5 Reverse Lookup Zones

When creating addr.arpa in Reverse Lookup Zones, the network Id to be used depends on subnet mask. In this project there are two subnet masks **16 bits class B** (255.255.0.0) and **24 bits class C** (255.255.255.0) where for 16 bits subnet mask is used for public network and 24 bits subnet mask for private. For subnet mask 16 bits (255.255.0.0), **addr.arpa** in Reverse Lookup Zones for public network will be **172.28.** (figure 84) and for subnet mask 24 bits (255.255.255.0), **addr.arpa** in Reverse Lookup Zones for public network will be **192.168.0.** (figure 85). Follow the following steps to create reverse lookup zones:

Right click on Reverse Lookup Zones=>New Zone=>Primary Zone=>To all DNS server running on domain controllers in this domain =>IPv4=> then next and finished.

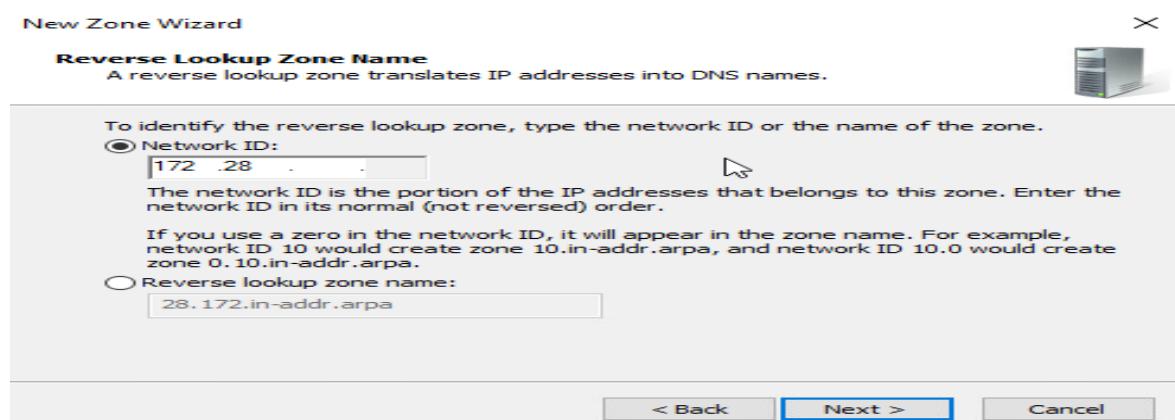


Figure 84, Reverse lookup zones for /16 Subnet mask

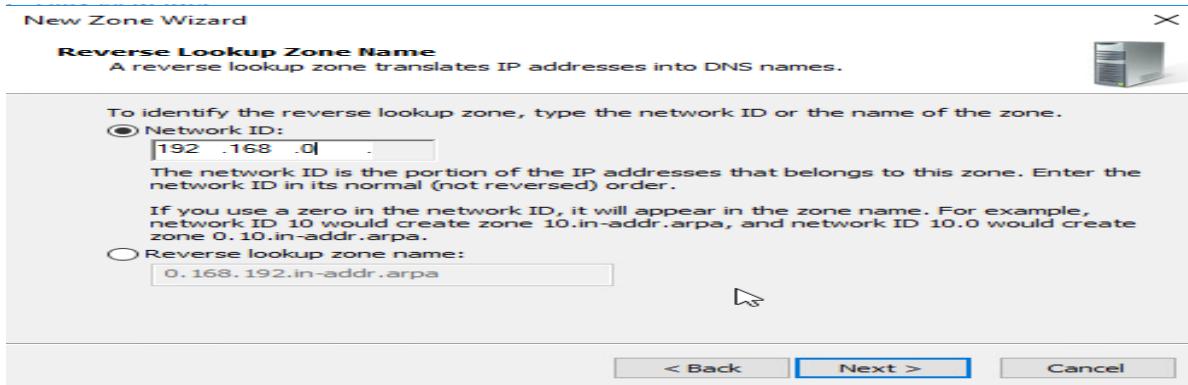


Figure 85, Reverse lookup zones for /24 Subnet mask

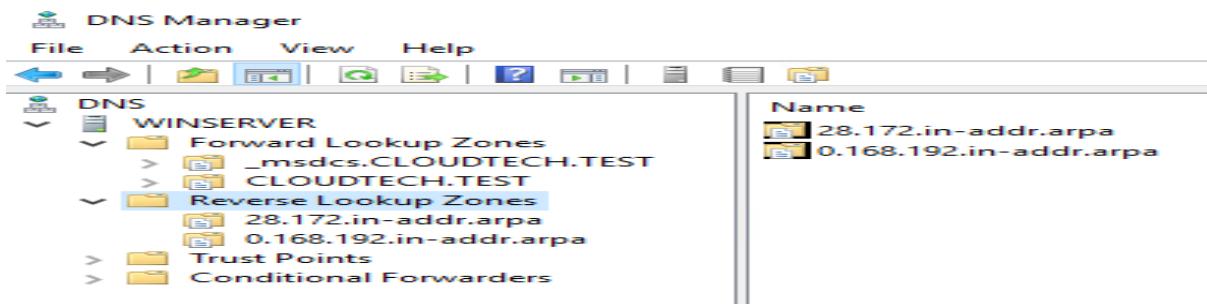


Figure 86, Reverse look up zones for private and public network

Adding new pointer PTR to create Reverse Lookup Zone in-addr.arpa for the machine which are already registered in the Forward Look up Zones. Go to the Reverse Lookup Zones => right mouse click to addr.arpa => New pointer (PTR) ... enter new resource record as in **figure 87**.

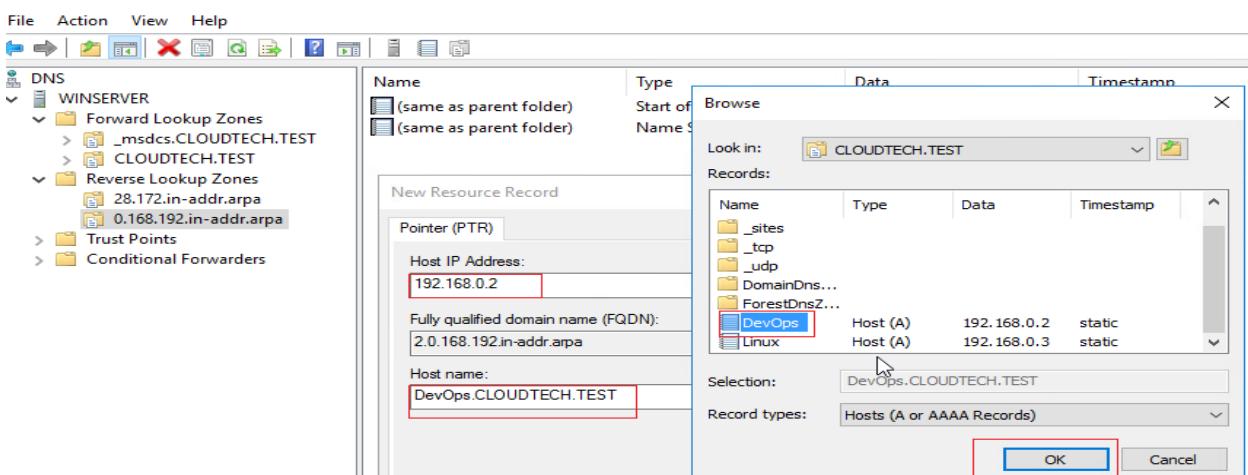
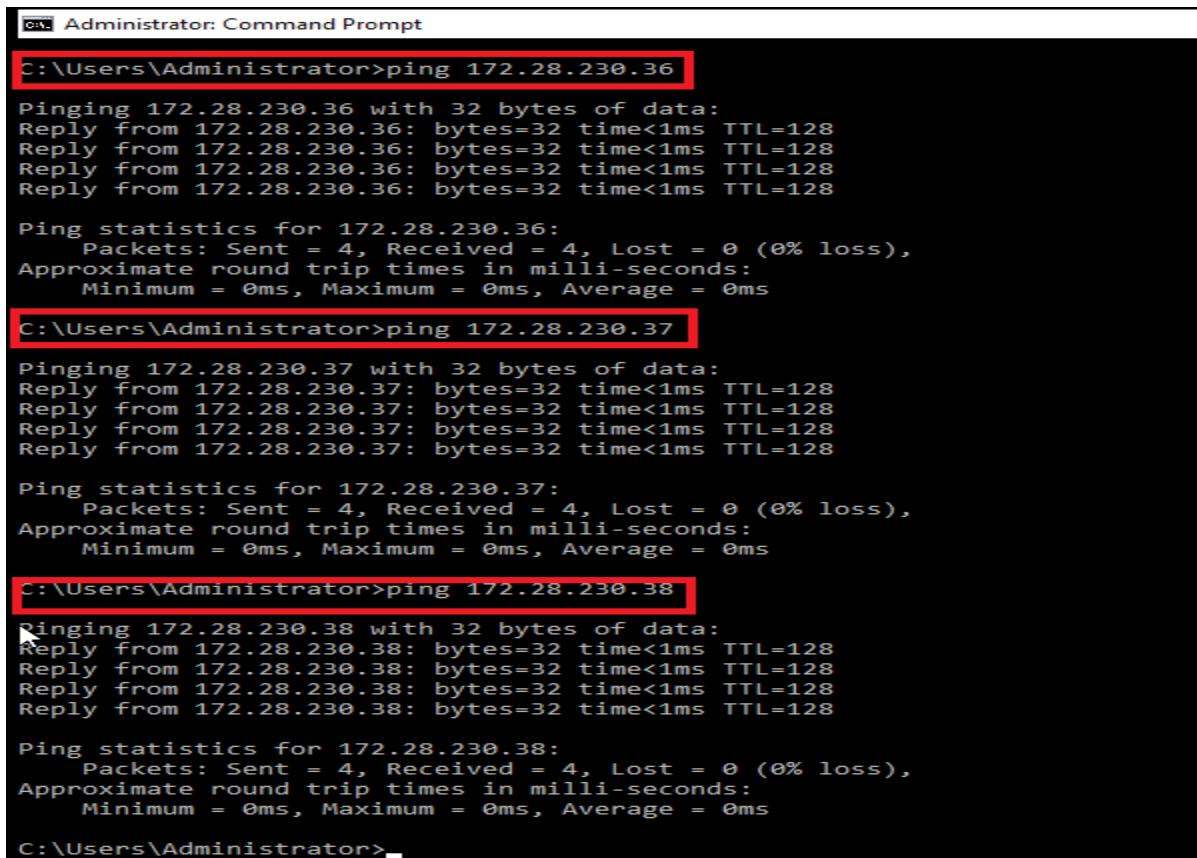


Figure 87, PTR for DevOps machine

This server is configured as a Domain Controller. It's IPv4 address, as preferred DNS server, comes from the given IPv4 range (**172.28.230.36 - 40**). The default gateway is the same as of tielab of Haaga-Helia. The primary purpose of this DNS server is to register all the ESXi level servers along with the created VMs with different operating systems, in this server.

6.3.6 Lookup Zones Test

After the DNS configuration and lookup zones are created, it is necessary to test that the machines are actually registered in the DNS records. This test can be done with 'ping' command to confirm that the registered machines are accessible as in **figure 88**.



```
C:\ Administrator: Command Prompt
C:\Users\Administrator>ping 172.28.230.36
Pinging 172.28.230.36 with 32 bytes of data:
Reply from 172.28.230.36: bytes=32 time<1ms TTL=128

Ping statistics for 172.28.230.36:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>ping 172.28.230.37
Pinging 172.28.230.37 with 32 bytes of data:
Reply from 172.28.230.37: bytes=32 time<1ms TTL=128

Ping statistics for 172.28.230.37:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>ping 172.28.230.38
Pinging 172.28.230.38 with 32 bytes of data:
Reply from 172.28.230.38: bytes=32 time<1ms TTL=128

Ping statistics for 172.28.230.38:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>
```

```
C:\Users\Admin>ping 172.28.230.39
Pinging 172.28.230.39 with 32 bytes of data:
Reply from 172.28.230.39: bytes=32 time<1ms TTL=64

Ping statistics for 172.28.230.39:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Admin>
```

Figure 88, Checking the DNS registration with ping command on all machines 1

```
C:\Users\Administrator>ping 172.28.230.40
Pinging 172.28.230.40 with 32 bytes of data:
Reply from 172.28.230.40: bytes=32 time<1ms TTL=64

Ping statistics for 172.28.230.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Checking the DNS registration with ping command on all machines 2

6.4 WinRouter Network Topology

The main reason for this WinRouter configuration is to separate the private and public networks. The host gets access to the physical adaptors- **nic0**, **nic1**. Virtual switches are attached to these two uplinks for this whole network environment. Required port groups are created and joined with these switches (**figure 89**), where VM Network (default network port group) has many VMs -Windows Server, WebServer, vCenter Server, WinRouter and so on. Every VMs has vnic which are connected to the VM Network port group.

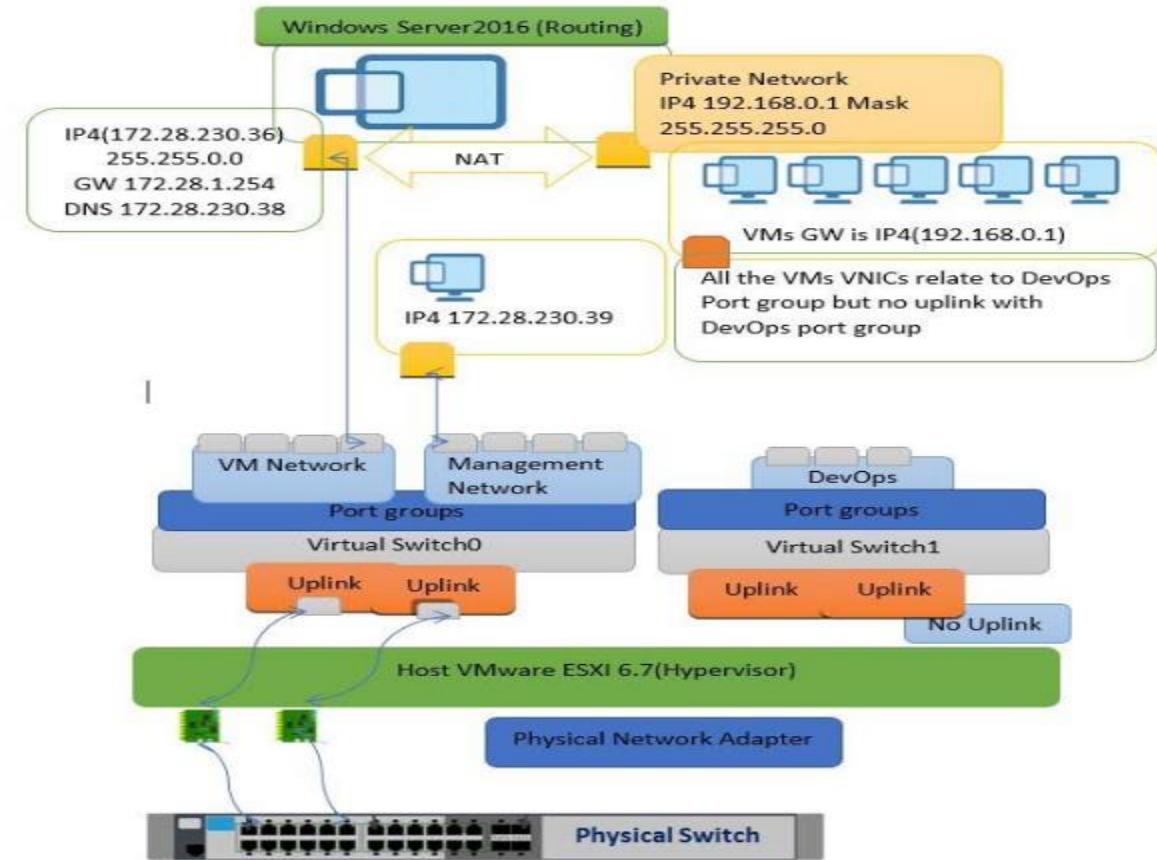


Figure 89, WinRouter network topology

The WinRouter is the major role player for network creation and separation. It takes two interfaces with separate subnets- (172.28.230.36 public through ESXi's default VM Network) and (192.168.0.1 private) where the public network IP comes from the given range of static IP address which are already mentioned above, and the private one is our own assignment. All the private IP addresses are connected to the DevOps port group only. DevOps port group is connected with DevOps vSwitch and vSwitch doesn't have any uplink connectivity. Which means DevOps port group is isolated from outside connectivity. The main purpose of this isolation is to create separate and secure DevOps platform with Jenkins integration and Apache Tomcat Server which will be discussed in **heading 7**. This server will serve as LAN and NAT Routing, where all the request to private IP and all the request to public network will be handled.

6.4.1 WinRouter Installation and Configuration

To install windows Server 2016 as a routing machine can be followed as in **heading 4.4** and after which roles and features need to be added make it work as a Routing Machine. Now we need to add two interfaces with two Network Adapters to the Routing Server and that can be done either from VMWare Workstation or Host web login line below.

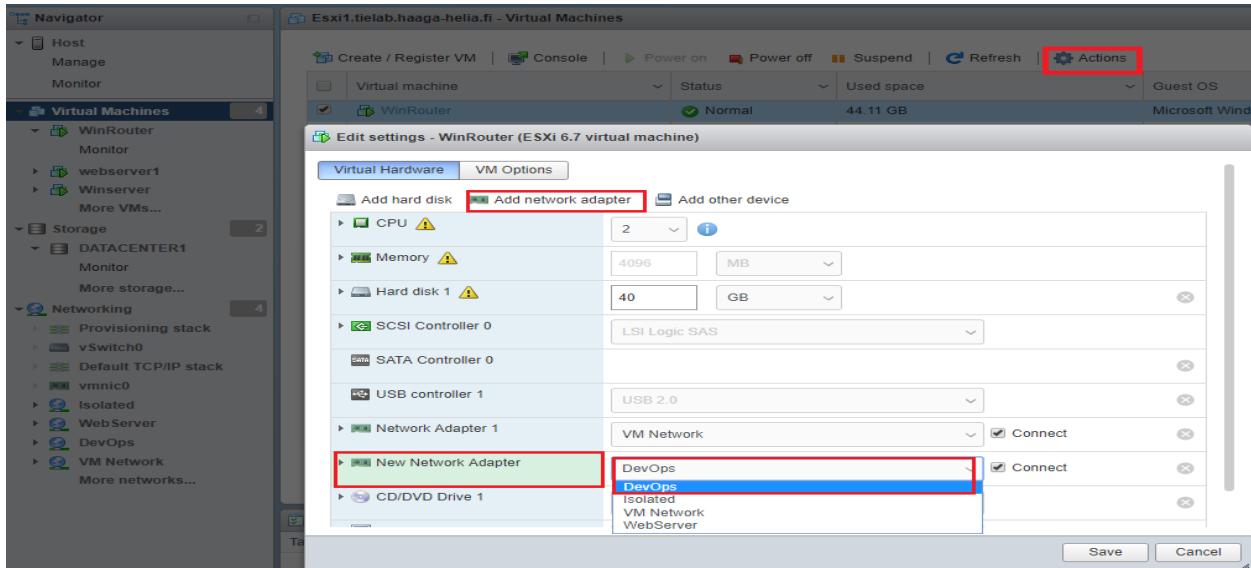


Figure 90, Adding network adaptors

Long into the Host with web log in GUI then select WinRouter and click Action, enter in Edit settings, click Add network adapter and select suitable port group then save. In this project, private and public networks are required, hence two network adapters are needed. After that the WinRouter gets two interfaces- public (172.28.230.36) and private (192.168.0.1) which can be confirmed by checking ‘ipconfig’ as in the **figure 91**.

```
Ethernet adapter Ethernet0: [red box]
  Connection-specific DNS Suffix . :
  IPv4 Address . . . . . : 172.28.230.36
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 172.28.1.254

Ethernet adapter Ethernet1: [red box]
  Connection-specific DNS Suffix . :
  IPv4 Address . . . . . : 192.168.0.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :
```

Figure 91, ‘ipconfig’ on WinRouter displays two network interfaces

The same can be confirmed from the Local Server properties as well, like in **figure 92**.

The screenshot shows the 'PROPERTIES' window for the 'Local Server' named 'WinRouter'. The left sidebar lists 'Dashboard', 'Local Server' (which is selected and highlighted in blue), 'All Servers', 'File and Storage Services', and 'IIS'. The main pane displays 'Computer name' as 'WinRouter' and 'Workgroup' as 'WORKGROUP'. Under 'Network & Sharing Center', it shows 'Windows Firewall' status as 'On', 'Remote management' as 'Enabled', 'Remote Desktop' as 'Disabled', and 'NIC Teaming' as 'Disabled'. Below this, two network interfaces are listed: 'Ethernet0' with IP '172.28.230.36' and 'Ethernet1' with IP '192.168.0.1'. To the right, sections for 'Last installed updates', 'Windows Defender', and 'Feedback & Diagnostics' are visible.

Figure 92, network interfaces

6.4.1.1 Routing Services Installation

Now, the WinRouter has confirmed two interfaces connected with **Ethernet0** and **Ethernet1** adapters as in the **figure 92**, now it is time to add roles and features to install router.

The screenshot shows the 'Server Manager' dashboard. The left sidebar includes 'Dashboard', 'Local Server', 'All Servers', 'File and Storage Services', and 'IIS'. The main area is titled 'WELCOME TO SERVER MANAGER' with sections for 'QUICK START', 'WHAT'S NEW', and 'LEARN MORE'. A 'Configure' button is prominent. The 'Manage' menu is open, showing options like 'Add Roles and Features' (which is highlighted with a red box), 'Remove Roles and Features', 'Add Servers', 'Create Server Group', and 'Server Manager Properties'. Step-by-step instructions for configuration are listed below the menu.

Figure 93, Router installation 1

The screenshot shows the 'Add Roles and Features Wizard' with the title 'Select installation type'. The left sidebar lists 'Before You Begin', 'Installation Type' (which is selected and highlighted in blue), 'Server Selection', 'Server Roles', 'Features', 'Confirmation', and 'Results'. The main pane describes the installation type and provides two options: 'Role-based or feature-based installation' (selected) and 'Remote Desktop Services installation'. The destination server is listed as 'WinRouter'. A note at the bottom states: 'Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).'

Figure 94, Router installation 2



Figure 95, Router installation 3

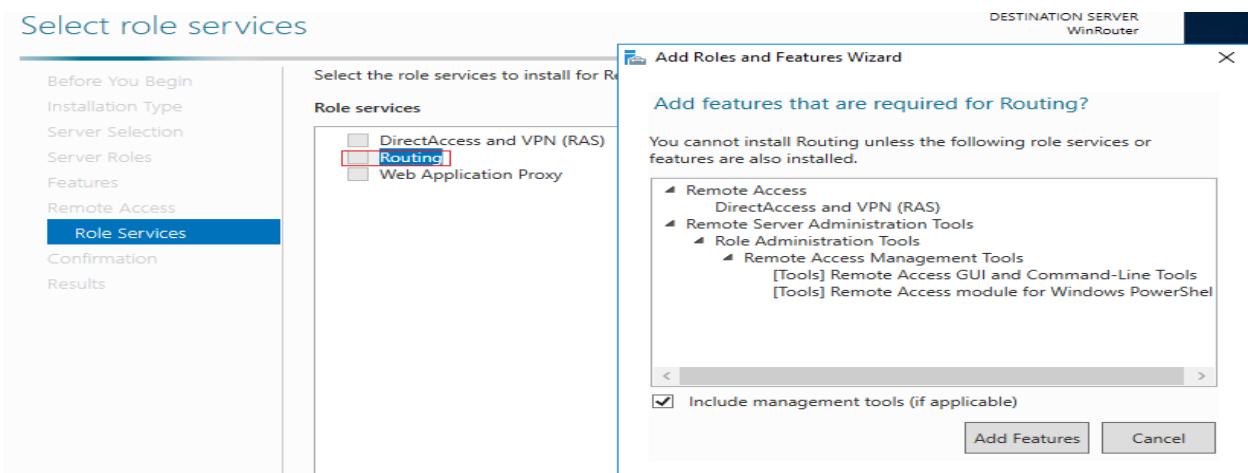


Figure 96, Router installation 4



Figure 97, Router installation 5

The options in the figure 97, appear selected by default and are required, too.

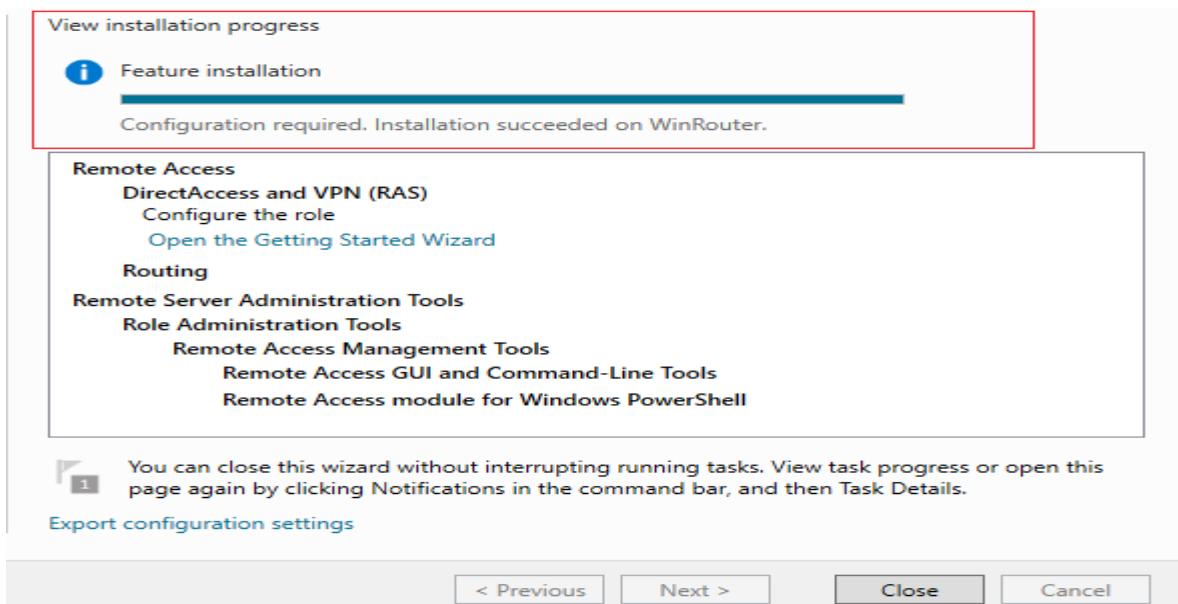


Figure 98, Router installation 6

Now that the required Remote Access and Remote Server Administration Tools are installed, the customary process of deployment can be left out and jump into tools and follow these steps:

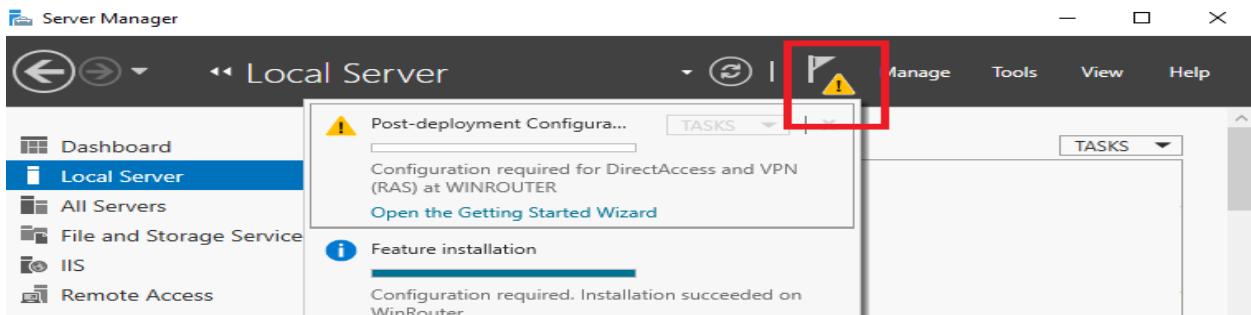


Figure 99, Post deployment config

6.4.1.2 NAT and LAN Routing Configuration

The following configuration is required to complete the router.

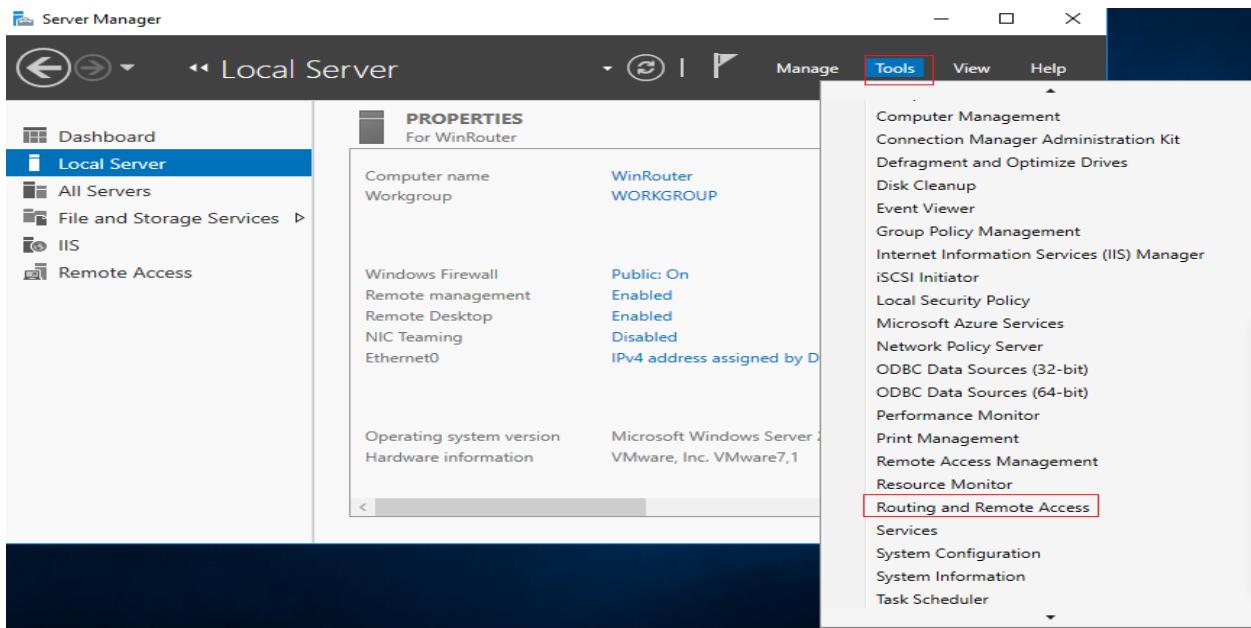


Figure 100, Routing config 1.

Go to Tools=>Routing and Remote Access=>

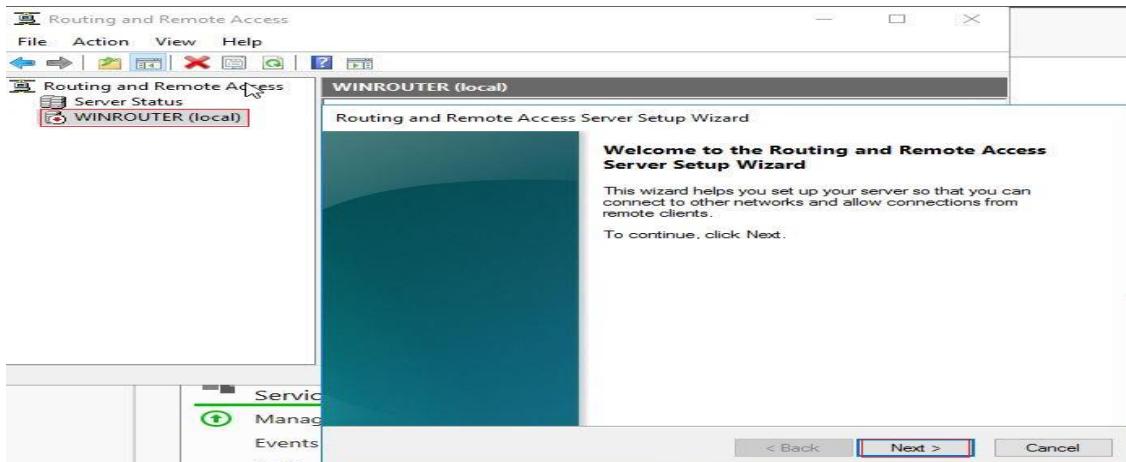


Figure 101, Routing config 2.

Right click on WINROUTER=>Customer configuration and Next.

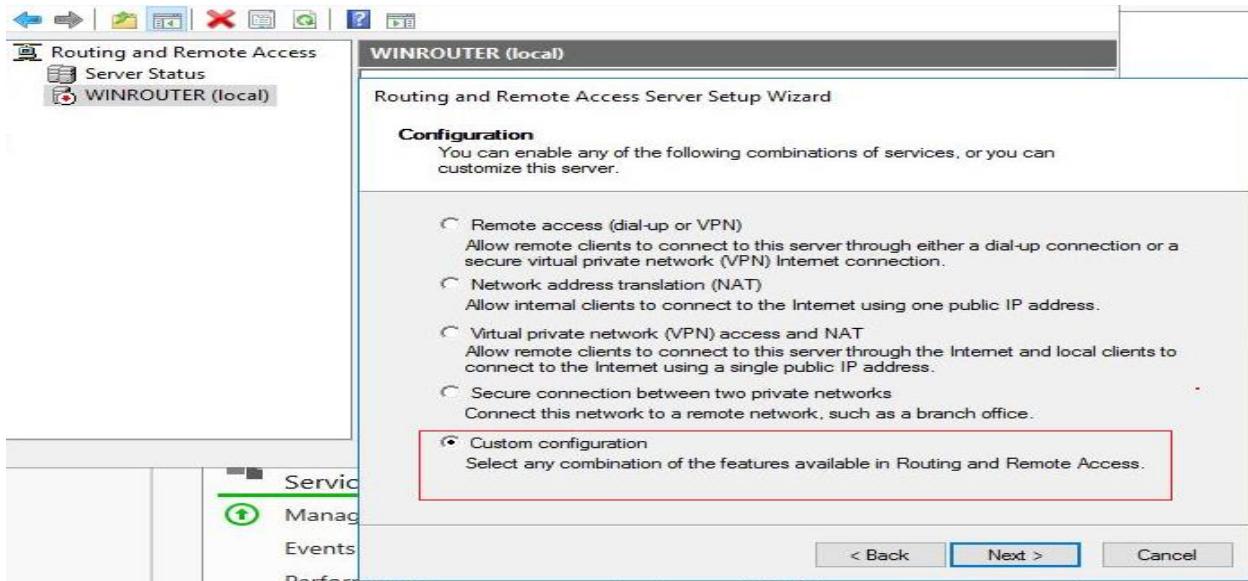


Figure 102, Routing config 3.

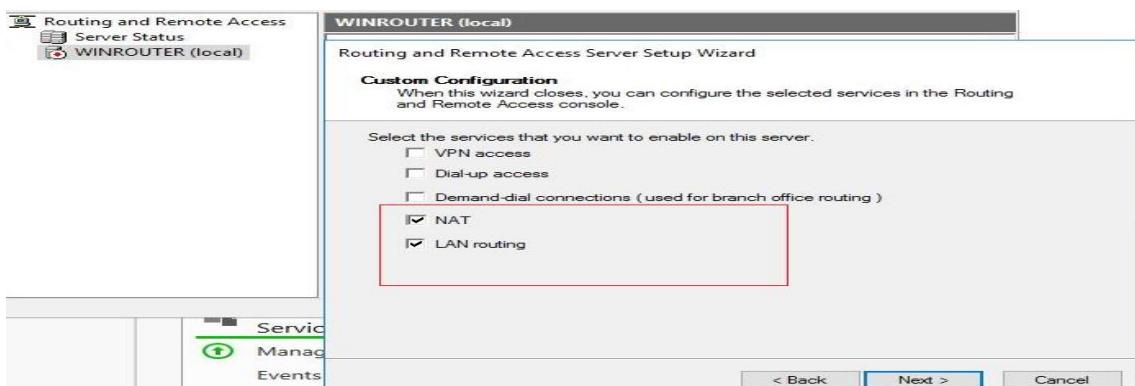


Figure 103, Routing config 4.

Choose both NAT and LAN routing=>Next.

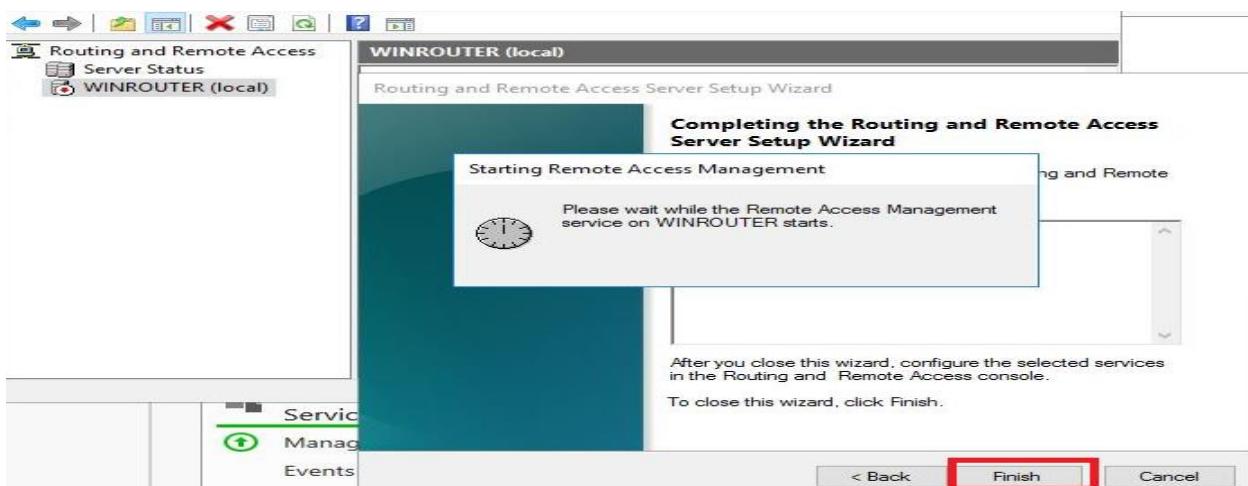


Figure 104, Routing config 5.

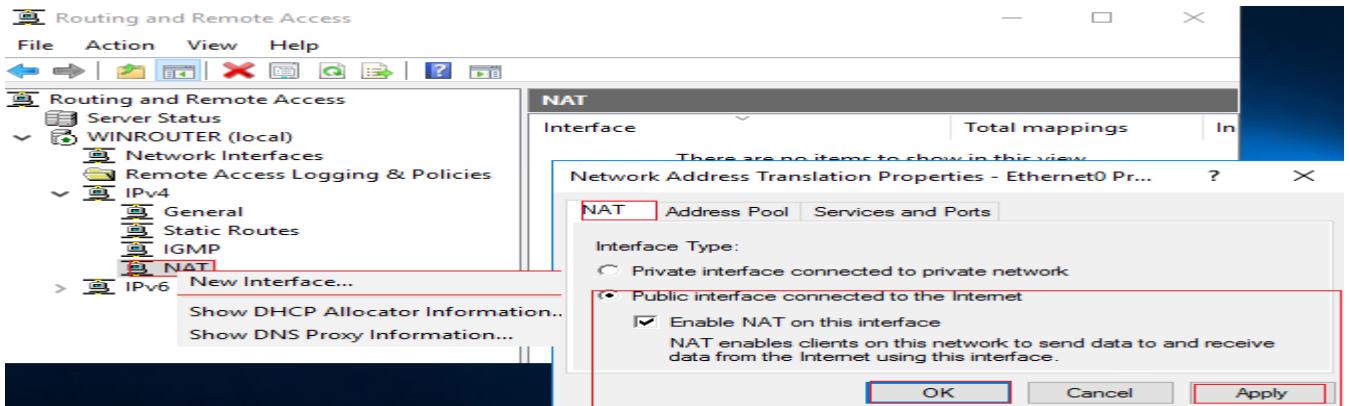


Figure 105, Routing config 6.

Right click NAT=>New Interface=>NAT interface type (**figure 105**) from the list like in **figure 106**, select the interface to run this routing protocol.

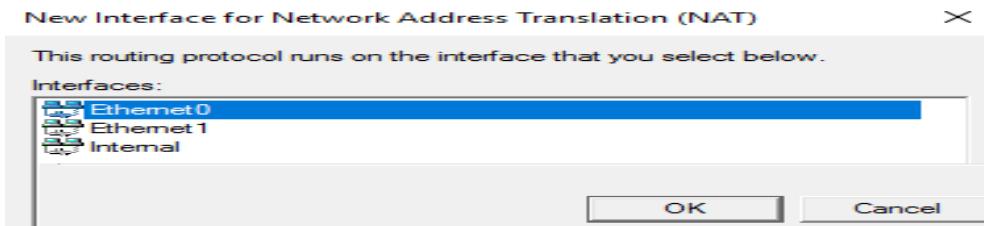


Figure 106, Routing config 7.

So far, the network address translation is done for **Ethernet1** as private interface and **Ethernet0** as public. For the private network, NAT port redirection rules can be made with the available services and ports so that each private individual machine, residing in the VMware ESXi, can be given separate incoming port numbers so that remote access can be allowed. The following **figure** describes how TCP protocol is configured for remote access to a private network through a defined incoming port (**figure 108**).

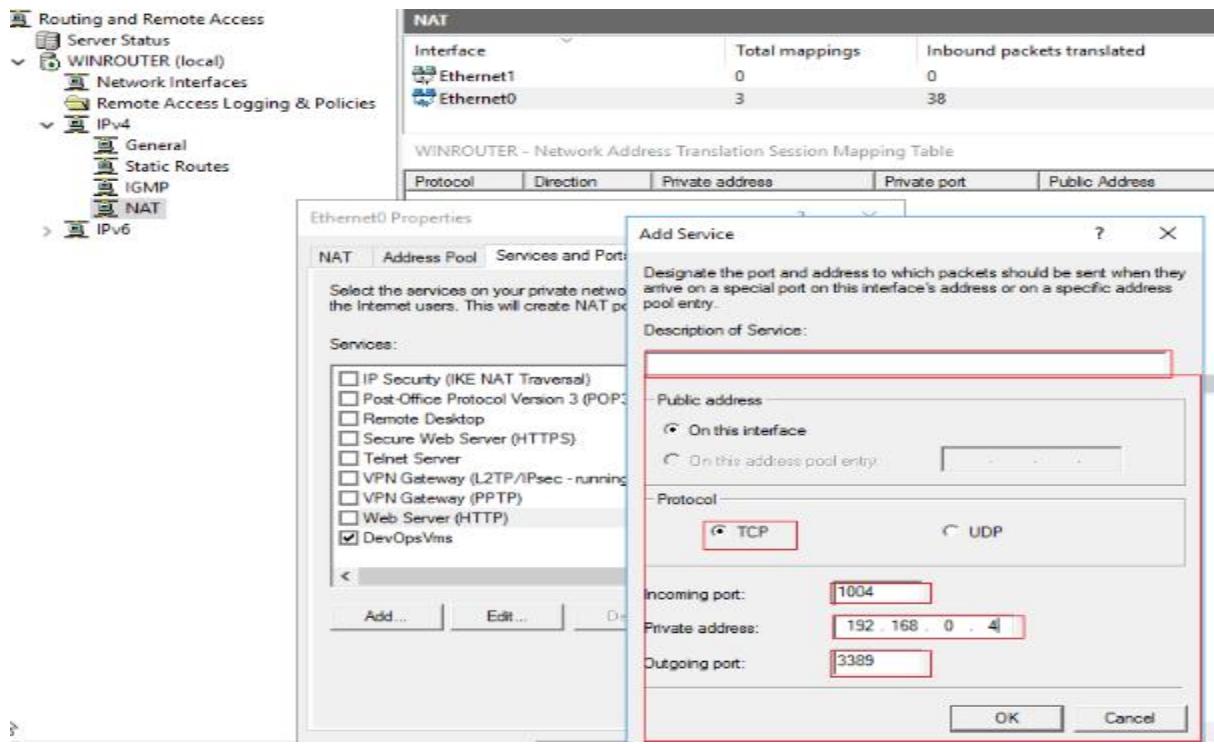


Figure 107, Creating rules for remote access.

6.5 Network Connections

This section talks about how to make get access to the ESXi host or any other machine running inside its network.

6.5.1 Remote Desktop Connection

As in **figure 107**, this rule can be set for other machines, too.



Figure 108, Remote desktop connection with port 1004.

6.5.2 SSH connection (PuTTY)

This connection can be started or stopped either from direct console of the ESXi host (by pressing F2 and providing credentials=> Troubleshooting Options =>Enter), or from vSphere Client- Host=>Configuration=> Security Profile=>properties=>check SSH, from host web interface it can be enabled or disabled with the firewall rules as in **figure 109**.

Name ▲ ▼	Key	Incoming Ports	Outgoing Ports	Protocols	Service
Software iSCSI ...	iSCSI		326U	TCP	N/A
SSH Client	sshClient		22	TCP	N/A
SSH Server	sshServer	22		TCP	N/A
syslog			1514, 514	UDP, TCP	N/A
vCenter Update			80, 9000	TCP	N/A

Figure 109, Enable-disable the SSH Server

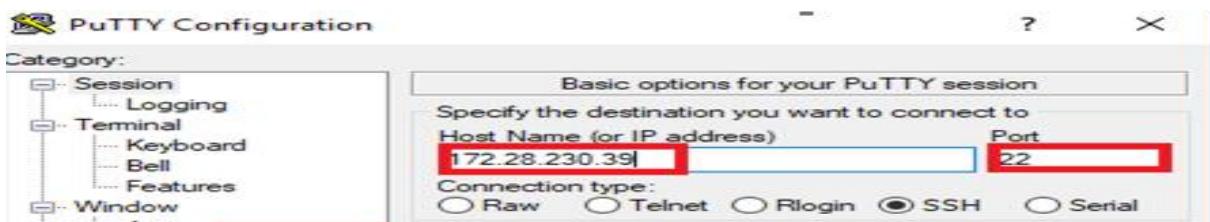


Figure 110, SSH connection in PuTTY

6.5.3 Direct Console User Interface

This service can be opened in the web user interface, so no need to make SSH connection through PuTTY. Just go to Host=>Actions => SSH Console. Then provide the root password and user. Few examples about how this console looks:

```

[root@Esxi1:~] esxcli network ip dns server list
  DNServers: 172.28.11.67
[root@Esxi1:~] esxcli network ip dns search list
  DNSSearch Domains: tielab.haaga-helia.fi, 172.28.230.38, 172.28.230.39
[root@Esxi1:~] esxcli network ip connection list
Proto Recv Q Send Q Local Address      Foreign Address      State       World ID CC Algo
-----  -----  -----  -----  -----  -----  -----  -----  -----
tcp      0        0    127.0.0.1:8307  127.0.0.1:28115  ESTABLISHED 2098548 newreno
tcp      0        943  127.0.0.1:28115  127.0.0.1:8307  ESTABLISHED 2128363 newreno
tcp      0        0    127.0.0.1:80   127.0.0.1:32995  ESTABLISHED 2098573 newreno
tcp      0        0    127.0.0.1:32995  127.0.0.1:80   ESTABLISHED 2131802 newreno
tcp      0        0    127.0.0.1:61715  127.0.0.1:8307  TIME_WAIT   0
tcp      0        0    127.0.0.1:23235  127.0.0.1:80   TIME_WAIT   0
tcp      0        0    127.0.0.1:12757  127.0.0.1:8307  TIME_WAIT   0
[root@Esxi1:~] esxcli network firewall get
  Default Action: DROP
  Enabled: true
[root@Esxi1:~] esxcli network nic queue loadbalancer state list
NIC      Enabled
-----
vmnic0   true
vmnic1   true

```

Figure 111, Direct Console User Interface (DCUI)

7 Continuous Integration and Deployment (CID)

CID is an automated journey of an application development from the scratch to seeing its compile, package, build, test and deployment not only once in a long time but as frequently as required and making it possible to deliver a ready product as soon as at a click.

7.1 DevOps

DevOps is a software development model that integrates the developers and operations teams together into a common platform in order to improve mutual cooperation, collaboration among the team members by also improving productivity through automation of the development activities, reducing error prone operations and continuous measurement of application performance. With the advent of DevOps, development has become incredibly faster, more effective and deployment has become automated that the developers do not have to spend much time on this part as the operation team automates the deployment, integrate every detail change continuously and deploy incessantly. In recent years, due to the growing use of agile software development frameworks, it has become an urgent need to continuously deliver product to the customer, adding or updating new features frequently

etc. This ultimately makes more use of the DevOps models. The trend has become such that no software company can stand out in the competition without embracing this fast track solution. Following traditional methods to develop applications will definitely leave companies far behind because, by the time they introduce or add one more feature, smarter companies (with DevOps in practice) will have already developed such features or applications. This is the main motivation to carry out this project on developing DevOps environment by using the most suitable tool, Jenkins.

To setup DevOps lab environment in **webserver1**, its required installation, promoting and configuration is done as a Webserver (**heading 7.1.1**). Along with Jenkins 2.180, it needs JDK8, JRE, Container (Apache-tomcat application server in this project), Maven (Build management tool based on Project Object Model), version control (GIT) to be installed about which are talked about in the following sections.

7.1.1 Webserver1 Configuration

Promoting Windows Server 2016 as a Webserver is done the same way as in **heading 6.4.1**, however from add roles and features, it requires Web Server (IIS) to be installed (**figure 112**).

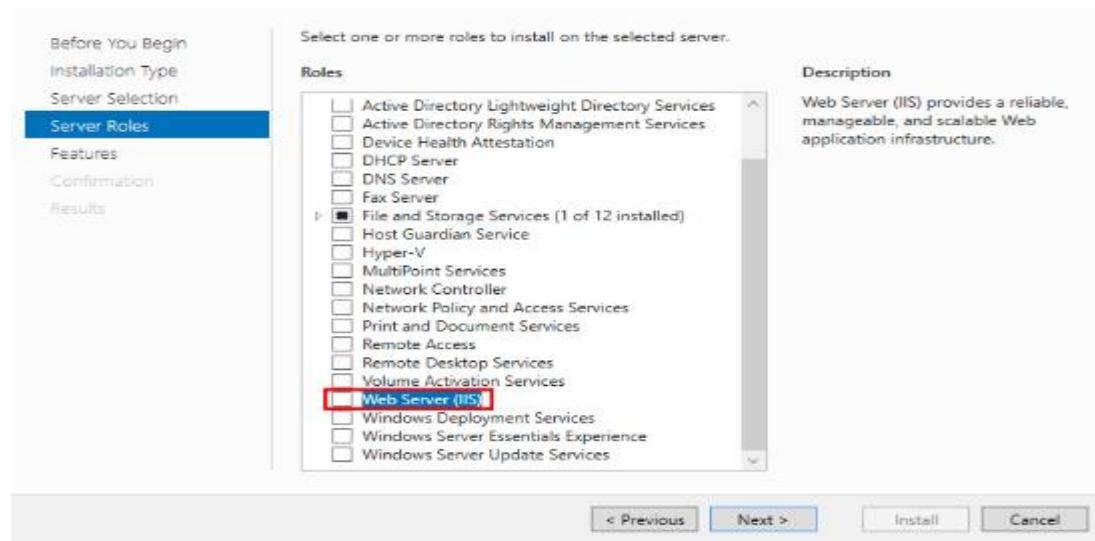


Figure 112, Web Server IIS installation

When the installation arrives to ‘Web Server Role (IIS)’ then select all the roles as in the **figure 113**.

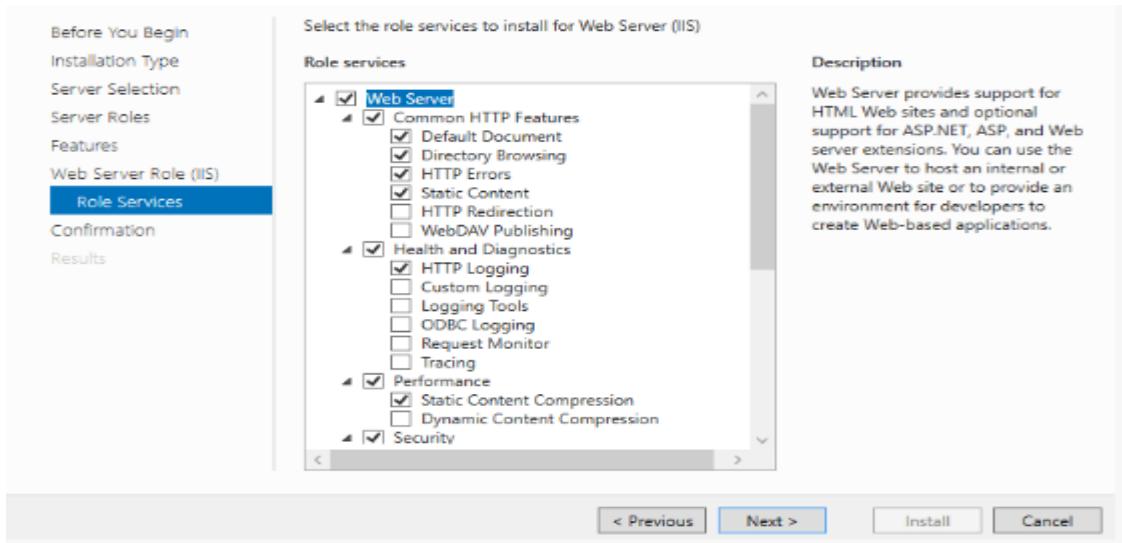


Figure 113, Web Server role services installation

When its installation completes it runs in localhost which is configured to bind as Webserv-
er1’s IPv4 address, that’s why instead of typing localhost the IP address can also be used as
in the **figure 114**.



Figure 114, Web Server running test

The main target of this IIS configuration is that a web domain can be created in which Jenkins deployment can be carried out. In this project, the web domain- www.devops.com is created for the deployment purpose. To create this web domain, open the server dashboard and go to the tools then click on IIS manager => WEBSERVER right mouse click => Add website and fill in the required information (**figure 115**) => OK

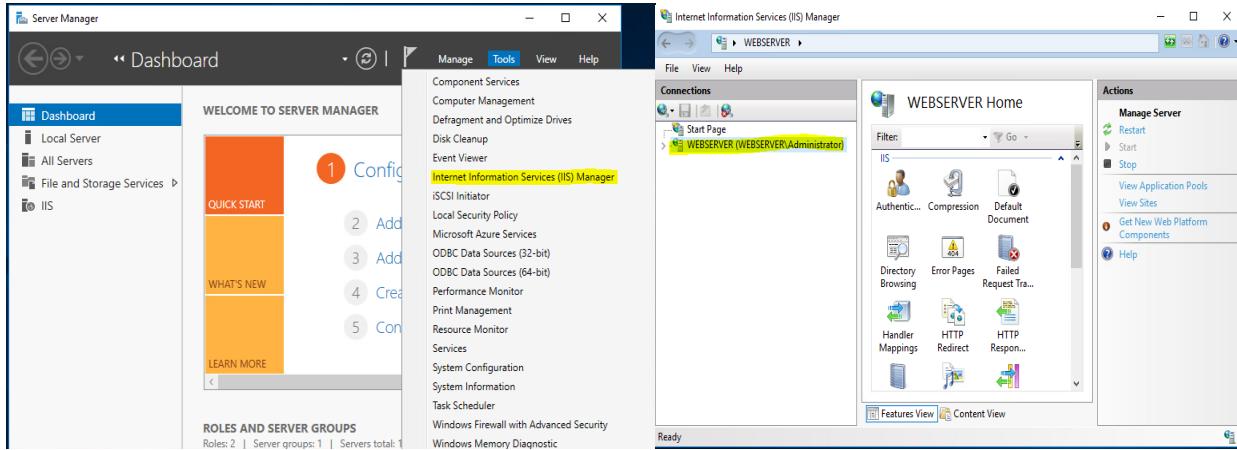


Figure 115, Web Server, creating domain-website 1

Then go to the Sites => devops => Default Document => default page (e.g.- index.html). This default page can be created inside the WWW physical local folder on the localhost (*C:/*) otherwise it cannot locate the default pages.

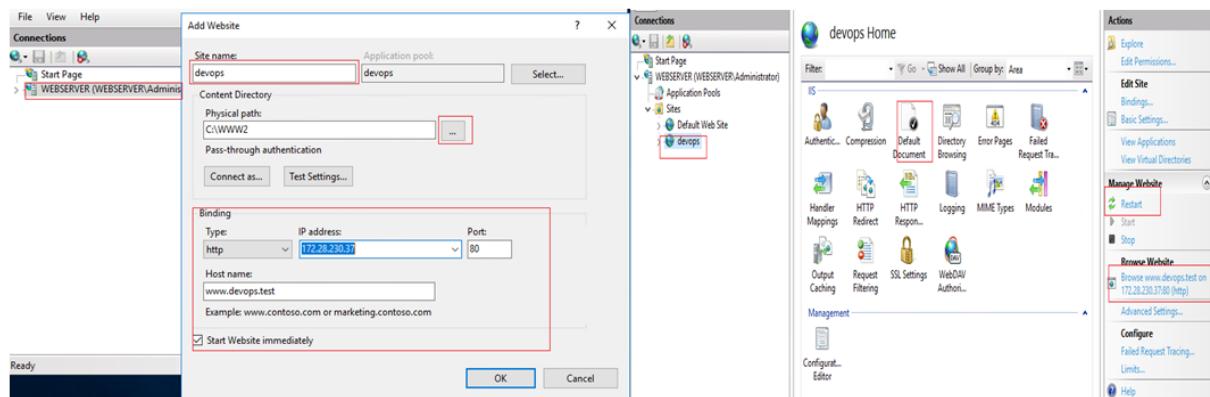


Figure 116, Web Server, creating domain-website 2

Just by creating website it won't be able to display in the (<https://localhost:80>) because it still needs own zone and FQDN, so its own zone should be created in the DNS server.

Forward Lookup Zone right click on it => New Zone => Next => Zone type (de-

fault) => Active Directory Replication Scope (Default) => provide a meaningful Zone name => Dynamic Update (Default) => Finish. After creating a new zone, create a new HOST-right click on created **New Zone (figure 117)**.

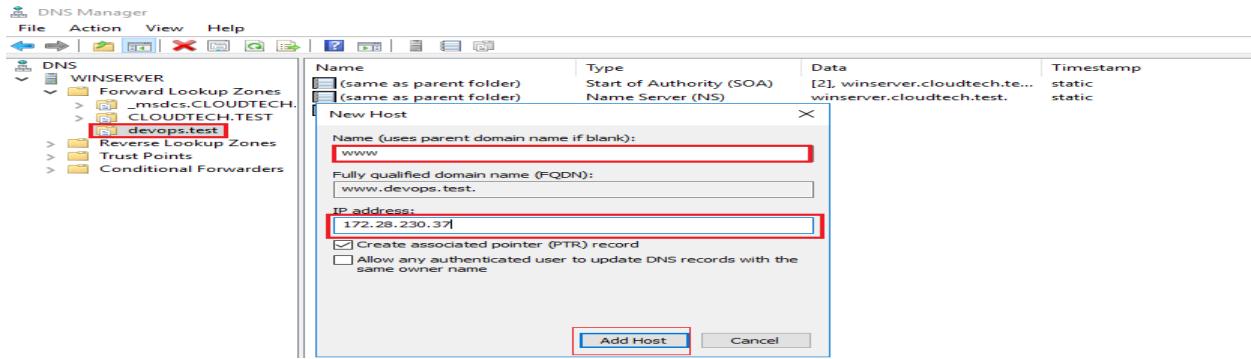


Figure 117, Web Server, creating domain-website 3

After this configuration, go to Webserver1, open the browser and browse the URL www.devops.test.



Figure 118, Web Server, domain-website up and running

7.1.2 JDK Installation and Setting Environment Variable

JDK creates java software development environment. It comes along with runtime environment (JRE) and compiler javac to build java application. JDK can simply be downloaded from the official site <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> where it requires user login to start downloading. Windows x64 JDK version 8 with .exe extension is downloaded here. Start by double clicking the downloader file and accepting. Finally, it creates a Java directory in 'C:\Program Files\'

To set the system environment variables path go to *control panel\System and Security\System*.

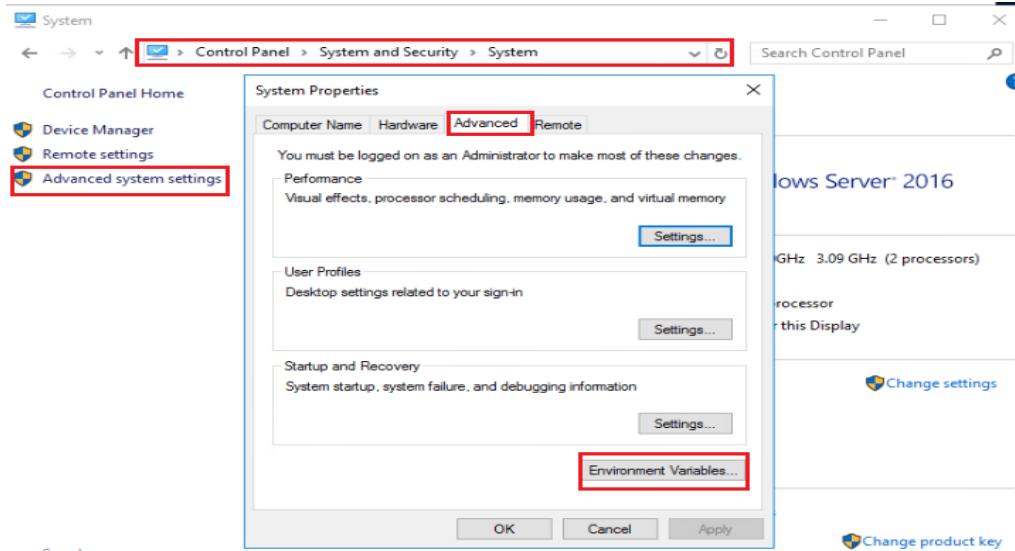


Figure 119, Setting system variable paths 1

To set this path write 'JAVA_HOME' as variable name and the absolute path (**figure 120**).

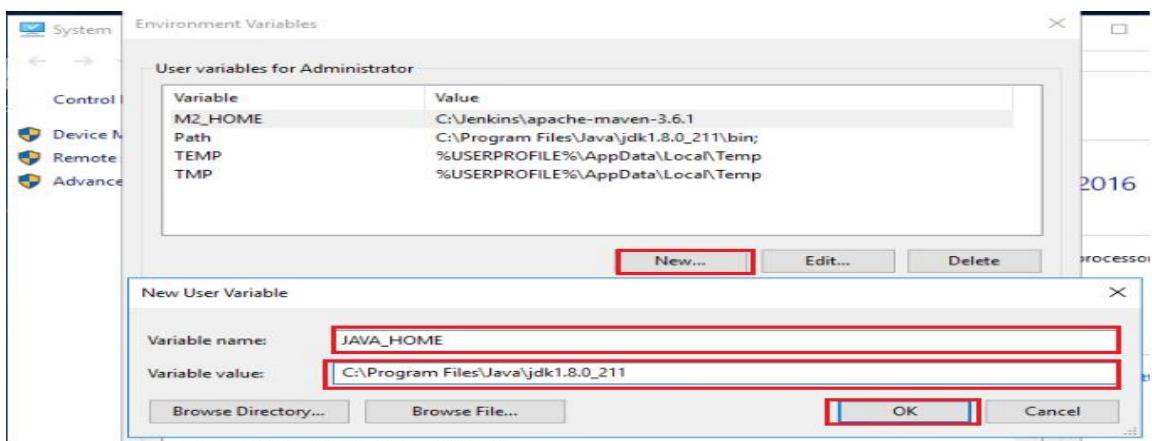


Figure 120, Setting system variable paths 2

If java is installed and runtime variable are set, the following command can display the java version:

```
C:\> Administrator: Command Prompt
C:\Users\Administrator>java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
C:\Users\Administrator>
```

Figure 121, Installed java version check

Now, one simple example of java compilation can be carried out to be sure that java is working and compiling java files properly.

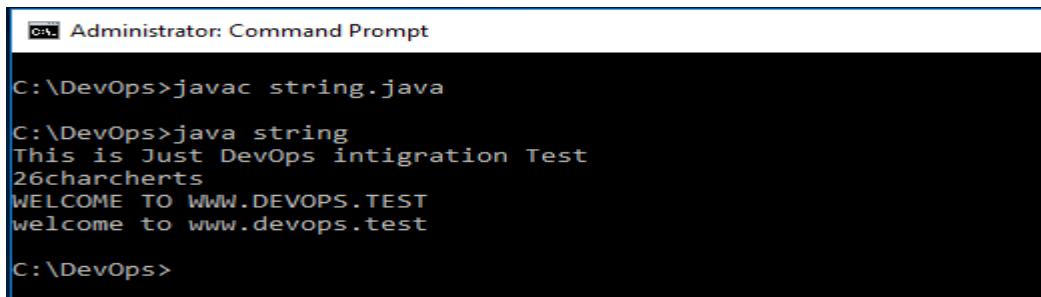
```

public class string {
    public static void main(String[] args) {
        String Stri = new String("Welcome to www.devops.test");
        System.out.println("This is Just DevOps intigration Test" );
        System.out.println(Stri.length()+"charcherts" );
        System.out.println(Stri.toUpperCase());
        System.out.println(Stri.toLowerCase());
    }
}

```

Figure 122, Simple .java code file

Save the above file in any folder with **.java** file extension and go to command line and follow the command as in **figure121**.



The screenshot shows an Administrator Command Prompt window. The user runs 'javac string.java' which compiles the Java code. Then, 'java string' is run, outputting the string "This is Just DevOps intigration Test" followed by its length (26) and the word "charcherts". The output also includes the uppercase version "WELCOME TO WWW.DEVOPS.TEST" and the lowercase version "welcome to www.devops.test". The prompt then returns to 'C:\DevOps>'.

Figure 123, Simple java program compiling

7.1.3 Tomcat Installation and Configuration

Go to its official website <https://tomcat.apache.org/download-70.cgi>, choose the appropriate version (for the moment Jenkins has good support for Tomcat 8 version), running platform (Windows installer 64-bit) and accept the installation.

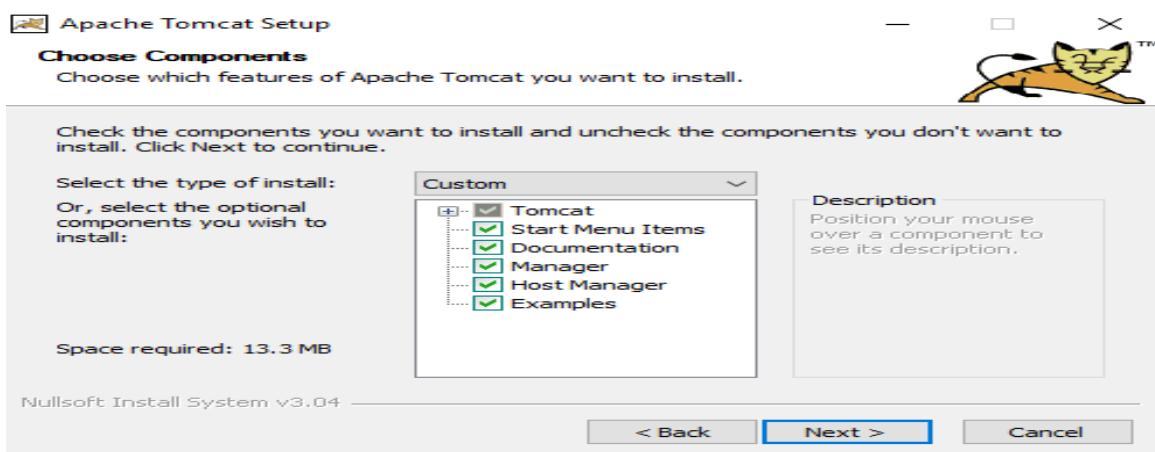


Figure 124, Installing Tomcat8

Select all the options then in the next window basic configuration for the port, shutdown port, administrator login credentials and roles can be defined, however they can be changed later.

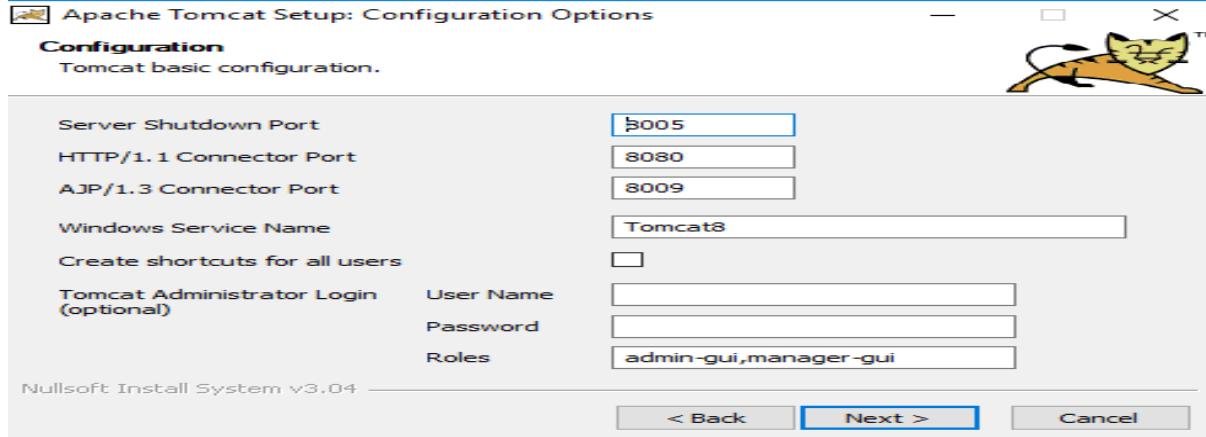


Figure 125, Setting ports, users, roles for Tomcat8

It is also better to make sure that there are no other applications running on port 8080 (**figure 126**). It can also be changed as per required from its **server.xml** file located in **C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf\server.xml**.

```
C:\>netstat -aon | findstr 8080
TCP  0.0.0.0:8080          0.0.0.0:0          LISTENING      4064
TCP  [::]:8080             [::]:0            LISTENING      4064
TCP  [::1]:8080             [::1]:50646        TIME_WAIT     0
TCP  [::1]:8080             [::1]:50647        TIME_WAIT     0
TCP  [::1]:8080             [::1]:50649        TIME_WAIT     0
TCP  [::1]:8080             [::1]:50650        TIME_WAIT     0
TCP  [::1]:8080             [::1]:50651        TIME_WAIT     0
TCP  [::1]:8080             [::1]:50652        FIN_WAIT_2    4064
TCP  [::1]:8080             [::1]:50653        FIN_WAIT_2    4064
TCP  [::1]:50652            [::1]:8080         CLOSE_WAIT   12240
TCP  [::1]:50653            [::1]:8080         CLOSE_WAIT   12240

C:\>netstat -aon | findstr 9090
C:\>
```

Figure 126, Port 8080 check

For auto deploy configuration to Tomcat, it needs to configure roles from **tomcat-users.xml** file as in the following **figure 127**. This file is located in **C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf\tomcat-users.xml**.

```
<role rolename="manager-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<role rolename="admin-gui"/>
<user username="admin" password="Password1!" roles="manager-script,admin-gui,manager-gui,manager-jmx,manager-status"/>
```

Figure 127, Installing Tomcat8

7.1.4 Jenkins

For the purpose of CID, there has been many tools out as free, open source projects e.g.- ElectricFlow, Jenkins, Octopus Deploy, AWS CodeDeploy, Bamboo, Cedar etc. However, this project uses Jenkins because it is found to be more reliable, rich built-in GUI, highly customizable, almost 400 plugins can be installed selectively as need, built in java language and runs in cross-platforms e.g.- Windows, Mac, UNIX, easy to download, highly scalable as build and deploy tasks can be distributed across its multiple slave machines, has the highest installation rate, and so on.

7.1.4.1 Jenkins Installation

Long-term Support and Weekly are two kinds of releases where former is more stable and better for long-term support, but the latter is released every week with bug fixes, better for short term-use (<https://jenkins.io/download/>). In this project, the long-term support one is used. It can be installed either in its **jenkins.war** file or executable file in zip. For jenkins.war file to run, it needs to be put in application server like Tomcat and can be run with command `java -jar jenkins.war`. However, in this project, **jenkins-2.180.zip** file has been downloaded because it is super easy to install with its **.exe** file, to get default plugins and to unlock the Jenkins user credentials. Installing JDK8 and java runtime environment (JRE) (**figure 119/120**) is required to run Jenkins. Once java version is there, it is good to install Jenkins.



Figure 125, Jenkins executable Windows installer package.

Jenkins installation wizard starts, then click next, provide destination folder and install. After the wizard completes, click finish and it gets started with unlock administrative password box as in **figure 126**:



Figure 126, Administrator path to unlock Jenkins

The admin password can be obtained by opening the file path `(x86)\Jenkins\secrets\initialAdminPassword`. Once the password is unlocked, it is recommended to allow the install suggested plugins initially, however there will be many other plugins to install as project starts to build in and deploy (**figure 127**).



Figure 127, Jenkins plugins install, default or custom

Then create first admin user and note down the credentials always.

Create First Admin User

Username:	<input type="text"/>
Password:	<input type="password"/>
Confirm password:	<input type="password"/>
Full name:	<input type="text"/>
E-mail address:	<input type="text"/>

Figure 128, Jenkins, first admin user credential set

After Jenkins is installed in the local computer and its home directory is **C:\Program Files (x86)\Jenkins**.

fingerprints	6/13/2019 12:58 AM	File folder
hudson.plugins.deploy.DeployPublisher....	5/24/2019 10:14 PM	File folder
jenkins-jobs	6/13/2019 12:48 AM	File folder
jobs	6/12/2019 2:10 PM	File folder
jre	5/15/2019 7:53 PM	File folder
labels	6/13/2019 1:08 AM	File folder
logs	5/17/2019 8:09 AM	File folder
nodes	6/13/2019 1:10 AM	File folder
plugins	6/12/2019 1:57 PM	File folder
repositoryPlugin	5/24/2019 9:33 PM	File folder
secrets	5/22/2019 8:14 PM	File folder
tools	5/22/2019 2:31 PM	File folder
updates	6/11/2019 10:52 PM	File folder
userContent	5/15/2019 7:53 PM	File folder
users	6/12/2019 12:39 PM	File folder
war	6/12/2019 12:18 PM	File folder
workflow-libs	5/15/2019 7:55 PM	File folder
workspace	6/12/2019 2:26 PM	File folder

Figure 129, Jenkins home directory

After that in instance configuration, jenkins URL <https://localhost:8080>, can be left default unless there are other applications running on port 8080. Once it is ready, open it in localhost at 8080 port and further plugins installation, configurations can be done before starting new jobs.

7.1.4.2 Jenkins Configuration

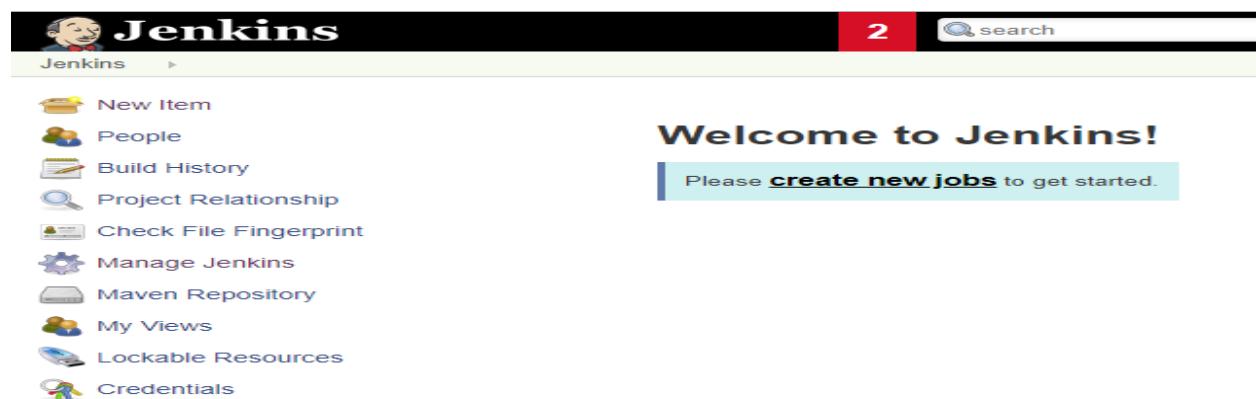


Figure 130, Jenkin's first UI view after a fresh installation

From Jenkin's interface, new item (task) can be created along with many other options. The tasks can be related to maven project or freestyle, or related to pipeline, external jobs (projects running outside Jenkins) etc, however before creating any new items, some basic or

default configuration can be set. Such configurations are done from ‘Manage Jenkins’ inside which there are system configuration, global security, credentials, global tools, manage plugins, manage nodes etc. configurations, however they all are not necessarily needed to configure but depends on type of project required to build, deploy, integrate etc.

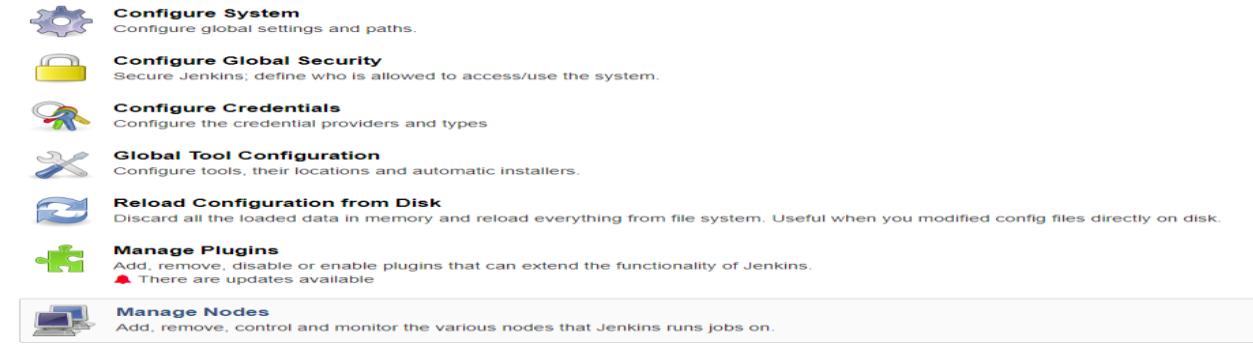


Figure 131, Jenkin's configuration options

Likewise, from ‘Configure System’ it is possible to set maven project properties, global properties, pipeline speed and so on. From ‘Configure Global Security’, security feature like access control, TCP ports for agents (ports can be fixed, random or disabled), API token etc can be configured as required. From ‘Global Tools Configuration’ JDK, GIT, MAVEN can be configured to auto-install or their Environment variable path (**figure 119-20**) can be provided here because every time Jenkins is building and deploying the project, it requires to reach those tools. For JDK auto-install option, oracle login account is required.

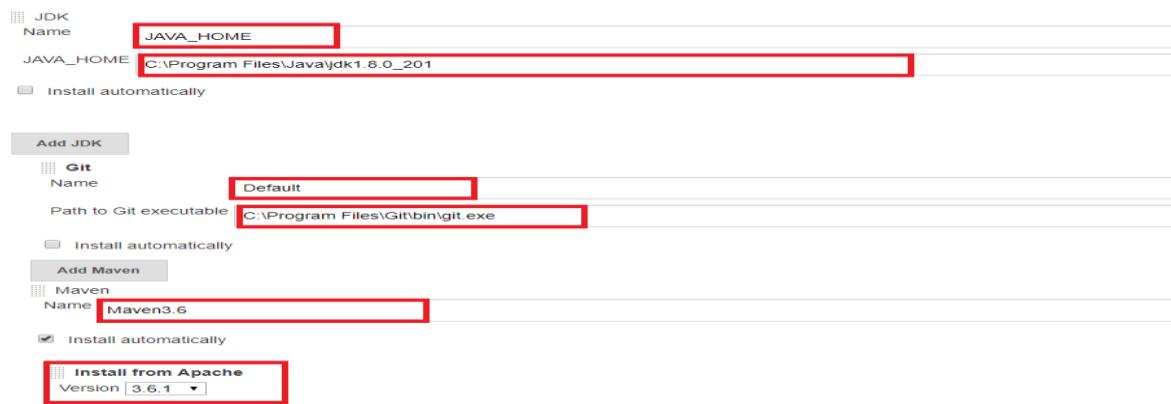
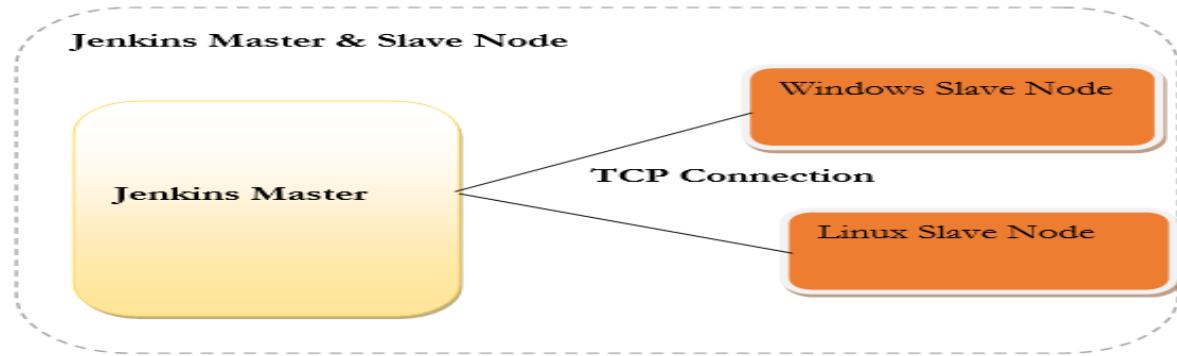


Figure 132, Jenkins Global Tools Configuration for JDK, Git and Maven

'Manage Plugins' contains plenty of plugins (Container, Pipelines, Maven, git, SCM, etc) needed to establish a complete CID DevOps Environment. Just need to download and install them as needed.

Jenkin's working environment concept is synonymous to Server-Client Architecture. In Server and Client Architecture, a question arises whether there is a need to setup workstation in the same physically location or in a remote server. In case of remotely working slave node, it communicates with servers through TCP/IP protocol and creates node virtually on the same localhost where the Master is managing, distributing, scheduling, monitoring tasks etc.



Jenkins Master and Slave nodes concept diagram

Creating node in Jenkins has the main purpose of building and testing in different environments (Windows, Linux, Unix). In this project, the Node has been created in locally not remotely. To create locally, it also needs its own Jenkins home directory where it can be placed. For example, in this project the path is `C:\ICT_Node\jenkins`

Go to Manage Jenkins => Manage Nodes => New Node (give a meaningful name) =>OK.

Figure 133, Creating nodes

Figure 134, Node configuration

In order to create a new node ICT_Node, it requires to configure it as in the figure above, where a node name, description, execution time (how many jobs can it execute at a time), remote directory (where the slave node's location is), usage. The launch method for the Windows node is better with 'Java Web Start', in case this option is not available, go to Man-

age Jenkins => Configure Global Security => TCP/JNLP => random. For the node properties, tool location for Git, JDK, Maven etc. can be specified same as before.

The screenshot shows the Jenkins interface with the 'Nodes' page selected. A table lists three nodes: 'ICT_Node' (selected and highlighted with a red border), 'master' (Windows 10, x86, In sync), and 'Window8' (Windows 8, x86, N/A). Below the table, a section titled 'Agent ICT_Node (This is just DevOps Testing)' provides instructions for connecting the agent. It includes a 'Launch' button (highlighted with a red border) and a command-line option: 'java -jar agent.jar -jnlpUrl http://localhost:8080/computer/ICT_Node/slave-agent.jnlp'. The 'Labels' section lists 'ICT_DevOps_Window_Node'. The 'Projects tied to ICT_Node' section shows 'None'.

Figure 135, Connecting node agent to Jenkins

Once the node is created, it appears as the figure above where just click the node name and launch and run as below.



Figure 136, Jenkins remote agent wizard

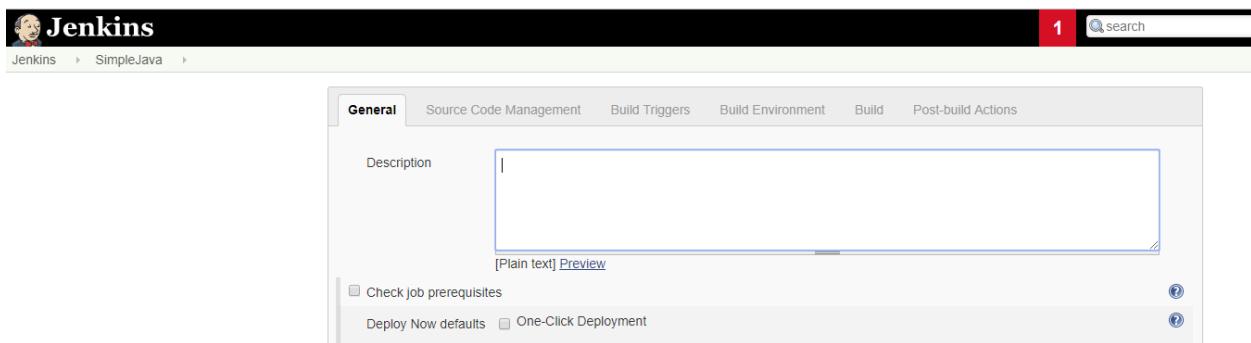
Now the node is ready to use. It can be given task by the Master node, for this, while the new task is being created then a task can be assigned to it. To create a new task, go to New Item (give a meaningful task name) =>freestyle => Ok

Enter an item name

» Required field

Freestyle project
 This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Figure 137, Creating new task

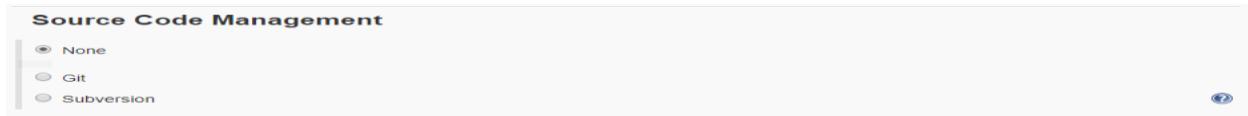


The screenshot shows the Jenkins interface for configuring a new task. At the top, there's a navigation bar with the Jenkins logo, a search bar, and a red notification badge with the number '1'. Below the navigation is a breadcrumb trail: Jenkins > SimpleJava >. The main content area has tabs for General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions. The General tab is selected. It contains fields for Description (with a placeholder 'A brief description of the job'), a checkbox for 'Check job prerequisites', and two buttons: 'Deploy Now defaults' and 'One-Click Deployment'.

Figure 138, New task configuration

In order to create a new task, Source Code Management, build triggers, build environment, Post-build action etc. fields need to be configured. In general section, just simple information can be configured and in the SCM section where and how this project pulls the code to build, deploy and in the build triggers section, how it can be triggered when the source code is changed in the version control. In the Build Environment section, how the project can be built and in the build section, other scripts can be added. In the post-build action, configuration can be done to decide where and how this project will be deployed and so on.

To integrate the locally available code or none-source code, simply check the 'Non' option (figure 139).



The screenshot shows the 'Source Code Management' configuration page. It features a radio button group with three options: 'None' (selected), 'Git', and 'Subversion'. There are also two small checkboxes at the bottom right of the page.

Figure 139, Integrating local project

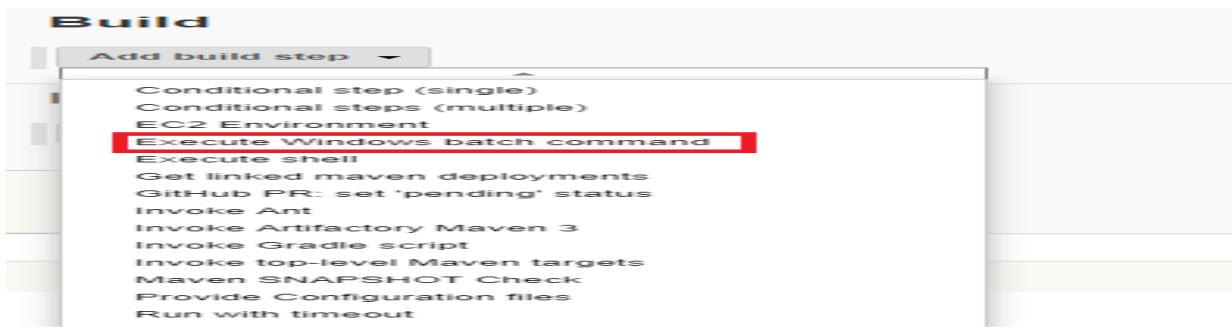


Figure 140, Setting Windows batch command 1

Because it is an example of locally available code to be integrated and built in Jenkins, it is required to supply the command for the build (**figure 141**).



Figure 141, Setting Windows batch command 2

This command allows the Jenkins to go C:/ directory, find ‘DevOps’ directory, compile the *string.java* file with ‘javac’ command, that in turn generates *.class* file, and run with ‘java string’ command.

After the required configuration as above, let’s go to Jenkins dashboard, click on project, click on build then the running build process can be seen in Jenkins console (**figure 143**).

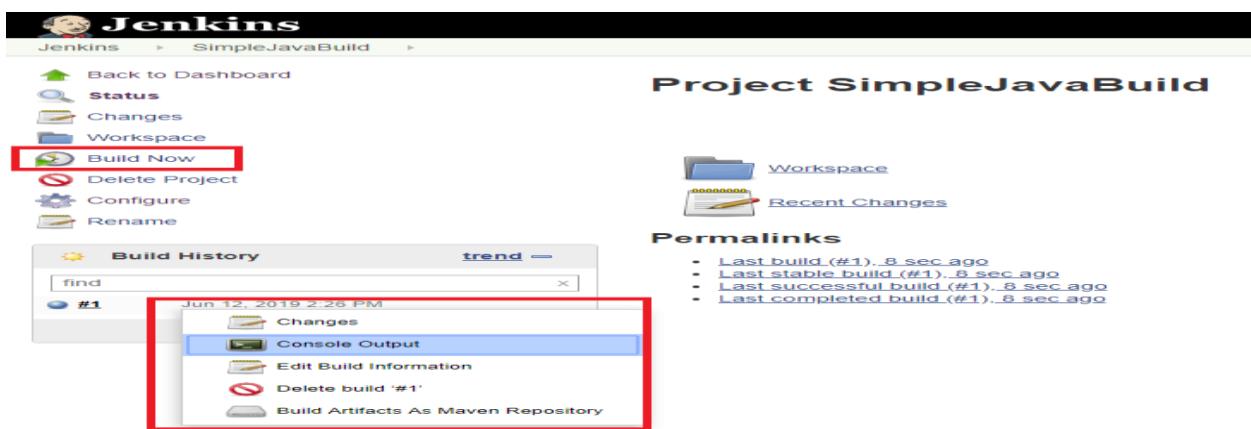


Figure 142, Non-source code build configuration

```

Started by user ICT Project
Running as SYSTEM
Building on master in workspace C:\Program Files (x86)\Jenkins\workspace\SimpleJava
[SimpleJava] $ cmd /c call C:\WINDOWS\TEMP\jenkins3343409479558139336.bat
C:\Program Files (x86)\Jenkins\workspace\SimpleJava>cd "C:\DevOps"
C:\DevOps>javac string.java
C:\DevOps>java string
This is Just DevOps intigration Test
26charts
WELCOME TO MwM.DEVOPS.TEST
welcome to www.devops.test

C:\DevOps>exit 0
Finished: SUCCESS

```

Figure 143, Jenkins compiles and builds the simple project successfully

How about the code which is somewhere in other version controls, for example if the code is in github repository? This time the source code management (SCM) pulls the code from the remote directory or repository.

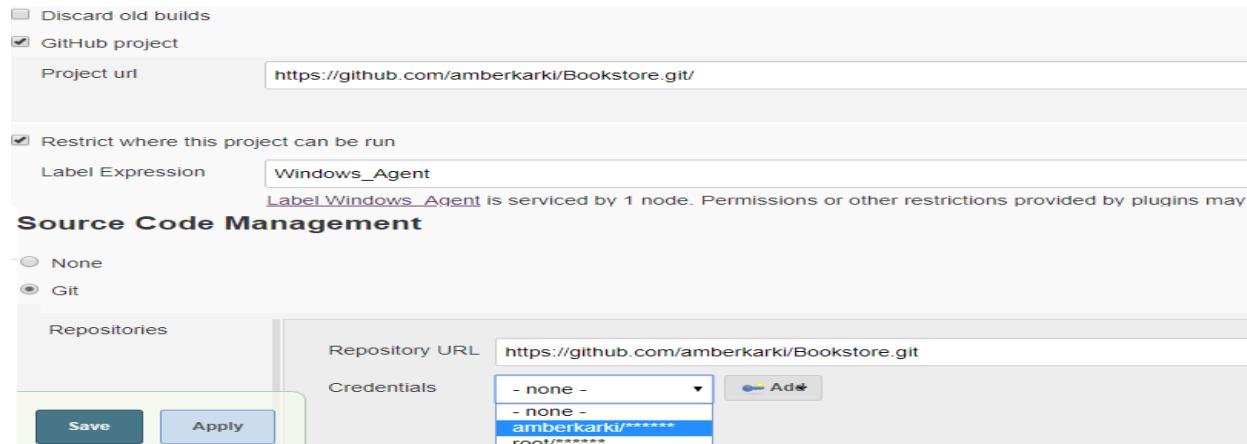


Figure 144, Jenkins compiles and builds the simple project successfully

For this github (remote) project, the basic new item creating is the same as in **figure 137-38**; the github url is provided, github credentials are better to be used here because sometimes Jenkins does not allow unauthorized codes to build. The build and deploy success process can be tracked from Jenkins console as in the following **figures 147-48**.

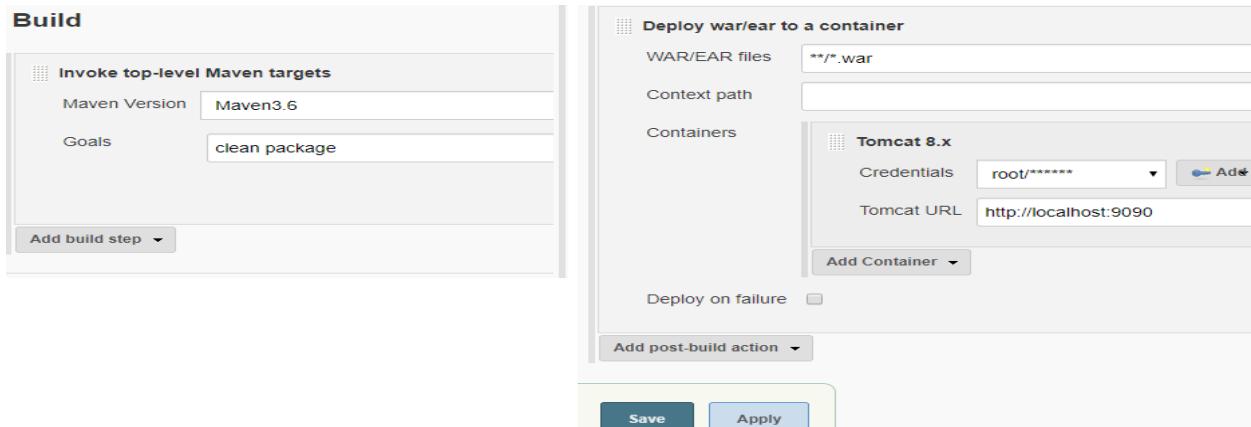


Figure 145, Build goals and deploy target

As mentioned earlier, every different project can have different configurations in Jenkins, there are different build and deployment configurations, too (figure 145). As this project is a Spring boot java project and the aim is to deploy in Tomcat8 container, it requires to download and install container plugin. To download the 'Deploy To Container Plugin' go to Manage Jenkins => Manage Plugins => Available => Search => Select => Download and Install.

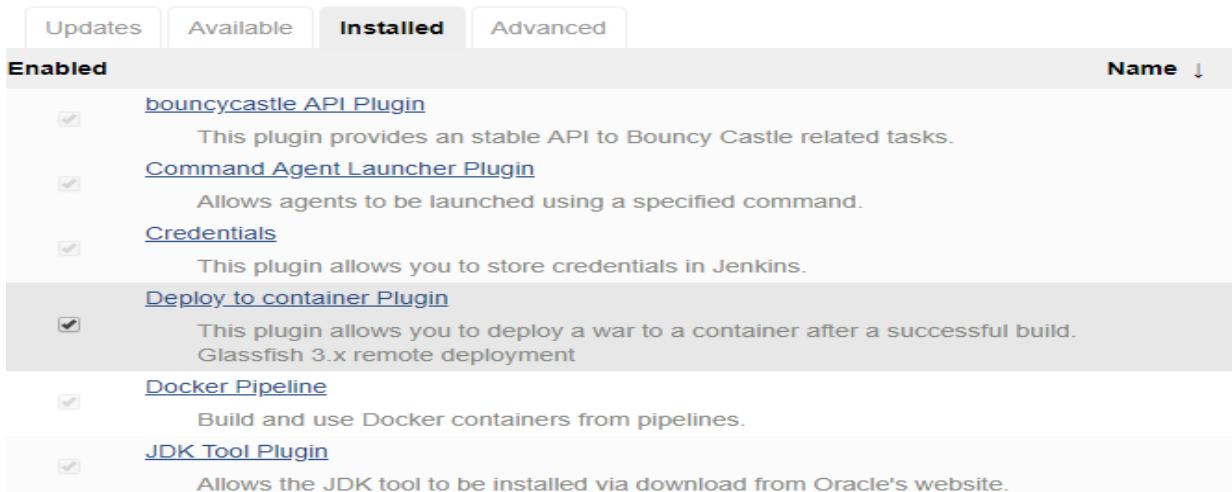


Figure 146, Installing container plugin

For this Bookstore project, in the Build, provide '*clean package*' input for goals along with Maven version 3.6. This Maven build lifecycle goal command packages the compiled code in distributable format. As the target is to deploy into the Tomcat, the distributable package format needs to be in .war extension. 'Context path' is the path name that the container finds it by this name and accepts it. Sometimes, if wrong context path is provided, it may not

be able to send it to the container, therefore it is wise not to write any context path unless it is correct. In the **figure 145**, Maven version is set with its clean package goals command and deployed .war file in to the Tomcat8 which is running into the <https://localhost:9090>. However, this localhost can be run in our domain server www.devops.test, too. Sometimes Tomcat8 does not take credentials while most of the times it takes, if it is not provided sometimes, the build process may not go success and throws some error message in the console, hence it is wise to provide it always.

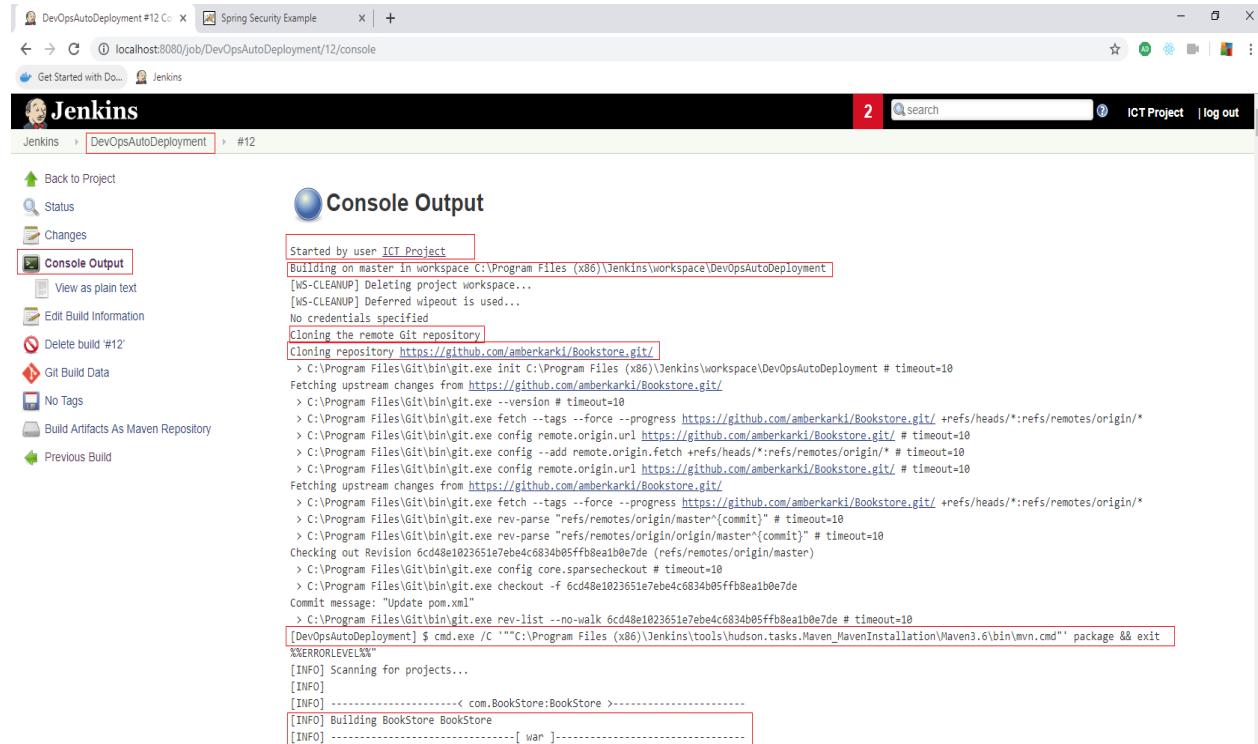


Figure 146, Jenkins building war file for Bookstore project to be deployed in Tomcat8 container

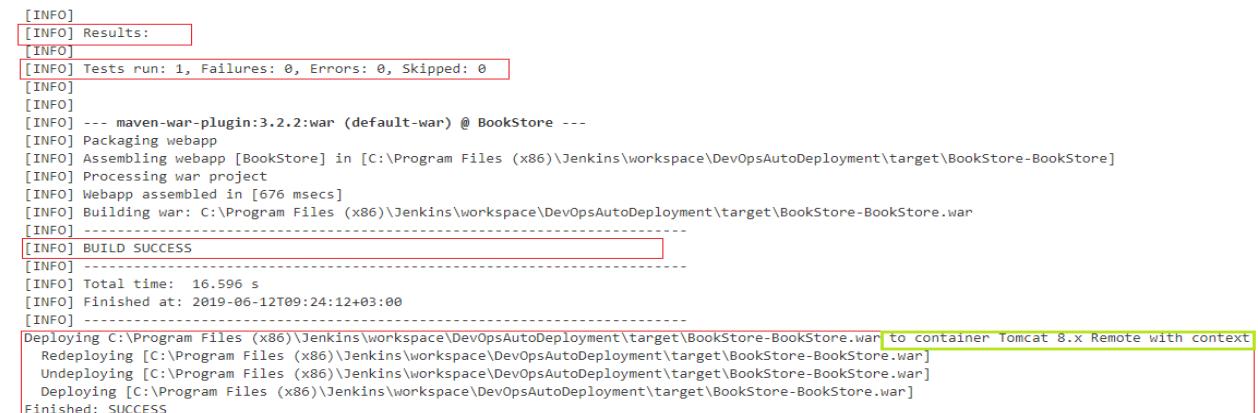


Figure 147, Build successful and sent to Tomcat8 container

```

[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ BookStore ---
[INFO]
[INFO] T E S T S
[INFO]
[INFO] Running com.BookStore.BookStoreApplicationTests
09:24:00.849 [main] DEBUG org.springframework.test.context.junit4.SpringJUnit4ClassRunner - SpringJUnit4ClassRunner constructor called with [class
com.BookStore.BookStoreApplicationTests]
09:24:00.856 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating CacheAwareContextLoaderDelegate from class
[org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate]
09:24:00.861 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating BootstrapContext using constructor [public
org.springframework.test.context.support.DefaultBootstrapContext(java.lang.Class,org.springframework.test.context.CacheAwareContextLoaderDelegate)]
09:24:00.890 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating TestContextBootstrapTrapper for test class
[com.BookStore.BookStoreApplicationTests] from class [org.springframework.boot.test.context.SpringBootTestBootstrapper]
09:24:00.914 [main] INFO org.springframework.boot.test.context.SpringBootTestBootstrapper - Neither @ContextConfiguration nor @ContextHierarchy found for test
class [com.BookStore.BookStoreApplicationTests], using SpringApplicationLoader

```

Figure 148, Testing project

Login to tomcat manager app with credentials and locate the ‘BookStore-BookStore’ context-path.



Tomcat Web Application M

Message:	OK												
Manager													
List Applications	HTML Manager Help												
Applications													
<table border="1"> <thead> <tr> <th>Path</th><th>Version</th><th>Display Name</th><th>Running</th></tr> </thead> <tbody> <tr> <td>/</td><td>None specified</td><td>Welcome to Tomcat</td><td>true</td></tr> <tr> <td>/BookStore-BookStore</td><td>None specified</td><td></td><td style="border: 2px solid red;">true</td></tr> </tbody> </table>		Path	Version	Display Name	Running	/	None specified	Welcome to Tomcat	true	/BookStore-BookStore	None specified		true
Path	Version	Display Name	Running										
/	None specified	Welcome to Tomcat	true										
/BookStore-BookStore	None specified		true										

Figure 149, Jenkins deployed the Bookstore project to Tomcat container

Now, the Bookstore project written in Springboot, Java, Thymeleaf etc. is deployed to the container (check ‘true’ marking in the **figure 149**), just click the application ‘BookStore-BookStore’ it opens as in following **figre 150**.

The figure consists of three vertically stacked screenshots of a web browser window. The top two screenshots show the same login page, while the bottom one shows a book list page.

Login Page (Top Two Screenshots):

The URL in the address bar is `localhost:9090/BookStore-BookStore/login`. The page contains the following form fields:

- User Name as user:
- Password as user:
-

Book List Page (Bottom Screenshot):

The URL in the address bar is `localhost:9090/BookStore-BookStore/booklist`. The page displays the following content:

Book List

Welcome user

[Sign Out](#)

Title	Author	Year	ISBN	Price	Category	Action
Mybook	Myauthor	Mydate	MyISBN	MyPrice	Science	Update
HisBook	HisAuthor	HisDate	HisISBN	HisPrice	Programming	Update
Zbook	Zauthor	Zdate	zISBN	Zprice	Romance	Update
ICT Infrastructure & Cloud Technology	ICT	14-06-2019	ISBN-16024271800727	€30.00	IT	Update

Figure 150, Bookstore application running

8 Conclusion

This report perfectly justifies the project topic and addresses the course topic: ICT Cloud and Infrastructure. Finally, a small level cloud platform has been created along with the DevOps services up and running inside it.

The founding tool for this project was VMware Esxi 6.7 hypervisor for virtualization, however the challenge was that its free trial version was finished and setting up the entire infrastructure from the scratch with a new license was slowing down the progress. We would have performed even better if we could have integrated VMware NSX also in VMware vCenter Server for entire network virtualization. In the future projects, this tool can be used for even better cloud infrastructure setup. We did not use DHCP server in this project but that could have been better option because it easily provides IP network addresses to all the VMs.

Our teacher Olavi Korhonen's guidance and cooperation were praiseworthy. He was available every now and then to help us solve problems, providing us with whatever physical resources (external hard drive, operating systems) required. Because of our own situations, we were not able to continue with this project for may weeks and we were also close to deadline with unfinished tasks, however he could understand our situation and extended the deadline, too, we remain grateful.

Despite the fact that it took almost 6 months of incessant study, research and hard labour, we are well satisfied with the knowledge we could amass and the confidence in this subject area.

9 References

- <https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.esxi.upgrade.doc/GUID-DEB8086A-306B-4239-BF76-E354679202FC.html>, last visited 20.01.2019.
- <https://my.vmware.com/web/vmware/registration>, last visited 02.02.2019.
- <https://docs.vmware.com/en/VMware-vSphere/6.7/vsphere-esxi-vcenter-server-67-networking-guide.pdf>, last visited 10.02.2019.
- <https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.install.doc/GUID-11468F6F-0D8C-41B1-82C9-29284630A4FF.html>, last visited 28.02.2019.
- https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vsphere.security.doc_50%2FGUID-92DF3F3CD17E-42BD-8620-9B0D780FE7F6.html, last visited 09.03.2019.
- https://www.youtube.com/watch?v=EPm8Fmhxasc&list=RDQMZGcKJ9ej2lg&start_radio=1, last visited 18.03.2019.
- <https://www.youtube.com/watch?v=J0pQ2dKFLbg>, last visited 01.04.2019.
- <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, last visited 12.04.2019.
- <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, last visited 20.4.2019.
- <https://tomcat.apache.org/download-70.cgi>, last visited 03.05.2019.
- <https://jenkins.io/download/>, last visited 08.05.2019.
- <https://jenkins.io/download/>, last visited 11.05.2019.
- <https://www.guru99.com/jenkin-continuous-integration.html>, last visited 12.05.2019.
- <https://scmquest.com/jenkins-master-slave-setup-and-configuration-with-screenshots/>, last visited 27.05.2019.
- <https://wiki.jenkins.io/display/JENKINS/Step+by+step+guide+to+set+up+master+and+agent+machines+on+Windows>, last visited 04.06.2019.

<https://www.youtube.com/watch?v=jfM1GiTOHSc>, last visited 04.06.2019.

<https://www.youtube.com/watch?v=jfM1GiTOHSc>, last visited 07.06.2019.

<https://www.youtube.com/watch?v=1e8I3A49ERc>, last visited 07.06.2019.

<https://www.youtube.com/watch?v=eIldJE4tSIA>, last visited 07.06.2019.
