# Preparations

# create 3 machines S1..S3 using script CreateVMs.ps1A computer screen shot of a computer program

Description automatically generated

# start the machines

# add three new computers to DC S1..S3 (DC quick setup.ps1 run from the DC machine)

# Go through the initial setup and set passwords for the machines

# add the machines to dc (RenameServers.ps1 + ServersToDomain.ps1)

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

**# dhcp ip range for the machines 192.168.99.119-121**

# # TASK 1

**# add NLB role for the s1, s2, s3**

Invoke-Command -VMName $SERVERNAME -ScriptBlock {Install-WindowsFeature NLB -IncludeManagementTools}

**# add new virtual switch (NLB) for the s1, s2, s3 and set the same ip addresses manually (but in network 98 just in order to keep the main working network clean)**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# # TASK 2

# if the Storage will be on the DC. We will Make the additional DISKS to it

# 1GB for witness, 5GB for file sharing

New-VHD -Path "C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\1gb.vhdx" -SizeBytes 1gb -Dynamic

New-VHD -Path "C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\5gb.vhdx" -SizeBytes 5gb -Dynamic

**# I'll isolate the storage in another network (97) and use the same ip addresses again**

**# everywhere the new network name is Ethernet 2, so I'll take advantage on it and i'll automate the process of renaming and setting IP addresses**

$credentials = Get-Credential

Invoke-Command -VMName $SERVERNAME -Credential $credentials -ScriptBlock {

    Rename-NetAdapter -Name "Ethernet 2" -NewName "Storage"

    New-NetIPAddress -InterfaceAlias "Storage" -IPAddress "192.168.97.$IP" -PrefixLength 24

}

A black screen with text

Description automatically generated

**# Using that with DC, S1, S2, S3 -> 109, 119, 120, 121**

**# only the servers will be isolated also in private 98 network (another virtual switch was created for the private network)**

**Invoke-Command -VMName $SERVERNAME -Credential $credentials -ScriptBlock {**

**Rename-NetAdapter -Name "Ethernet 2" -NewName "Private"**

**New-NetIPAddress -InterfaceAlias "Private" -IPAddress "192.168.98.$IP" -PrefixLength 24**

**}**

**# Using that with S1, S2, S3 -> 119, 120, 121**

**# Going through the process of iSCSI target and initiator configuring**

**# It's the same as in practice until the Scale-Out File Server Configuration is needed:**

- install target server on DC

- create quorum disk and target (1gb)

- create shared disk and target (5gb)

- add 97.199-121 ip addresses to the targets

- configure initiator on S1, S2, S3 with the corresponding addresses in network 97

- Only on S1 i'm bringing the newly added disks online/gpt/with the corresponding names and letters

**# My DC already has File server installed so this is needed only on S1..S3:**

**Install-WindowsFeature *-Name* FS-FileServer**

**# ofcourse with Invoke-Command i can use the same command on S1..S3**

**Invoke-Command -VMName $SERVERNAME -Credential $credentials -ScriptBlock {**

**Install-WindowsFeature -Name FS-FileServer**

**}**

**# I am installing only Failover Clustering**

**Install-WindowsFeature *-Name* Failover-Clustering *-IncludeManagementTools***

**# when i see that the same procedure is needed on S1..S3 I'm doing it via powershell command:**

**# the credentials are still in a variable:**

**Invoke-Command *-VMName* $SERVERNAME *-Credential* $credentials *-ScriptBlock* {**

**Install-WindowsFeature *-Name* Failover-Clustering *-IncludeManagementTools***

**}**

A black screen with white text

Description automatically generated

# validate configuration

# delete the NLB cluster via RSAT-NLB from DC, because it makes problems with the cluster creation

# remove the network cards connected to NLB switch and remove the switch as well

# reboot all machines

# of course there was a problem with the validation, so I’ve recreated the environment

# the new dns is 99.136, S1..S3 has 137..139 IP addresses,

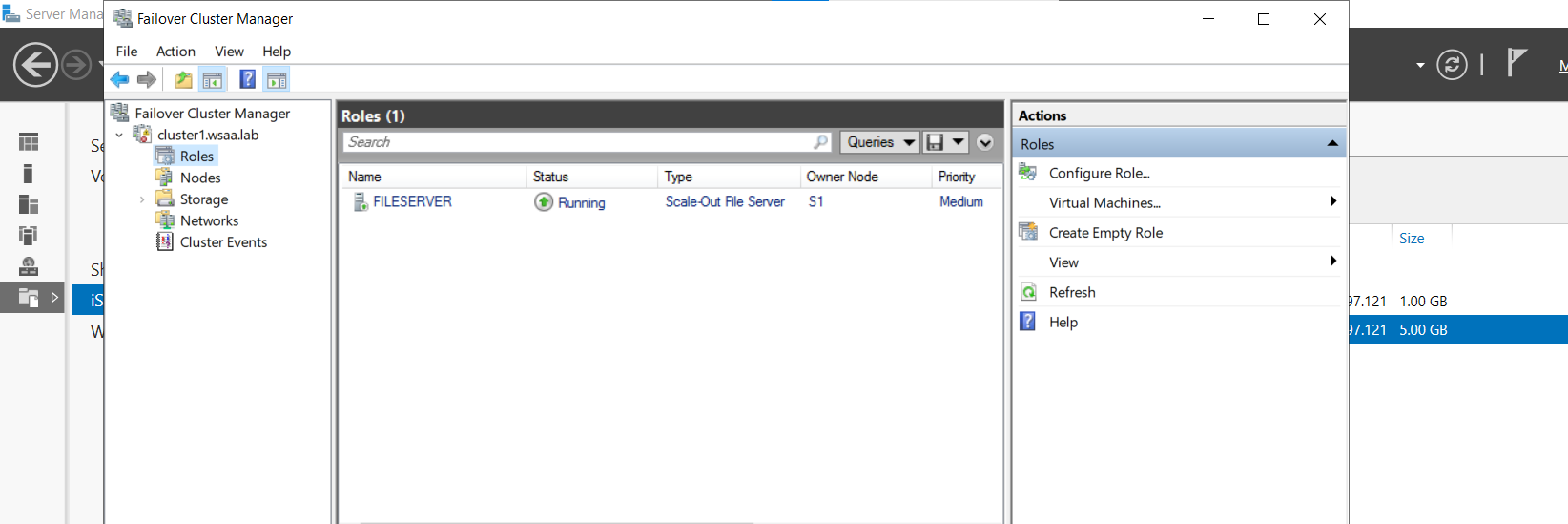
# but I’ve kept the 97 network for Storage and 98 network for Private with 119, 120, 121 as IP addresses to the Servers.

# DC was only connected to the STORAGE, so it has 97.109 IP there

Set-VMNetworkAdapter -VMName $SERVERNAME -MACAddressSpoofing on

# validate configuration and create cluster named "cluster1"

# Configure role - File Server - Scale-Out File Server for application data - Add NETBIOS name - Next - Configure HA - Finish



# # TASK 3

# Preparations:

# Open ports 2377 TCP, 7946 TCP/UDP, 4789 UDP (these are open by default, maybe from docker installation)

# Install docker engine 1.13.0 or later

# initialize the cluster (my address is 192.168.23.97)

docker swarm init --advertise-addr 192.168.23.97 --listen-addr 192.168.23.97:2377

A screenshot of a computer program

Description automatically generated

# After that I am joined to the swarm and I am the manager

# creating an overlay network and test configuration

A screen shot of a computer

Description automatically generated

# deploy service to a swarm

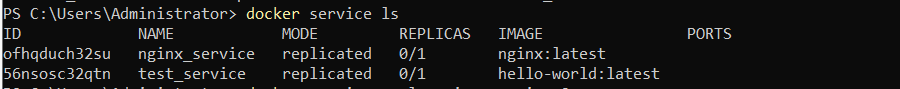
docker service create --name=test\_service --endpoint-mode dnsrr --network=softuni\_homework hello-world

# this image is not a service and will not run properly, but it checks out on

docker service ls  
# I am trying the same with nginx

docker service create --name=nginx\_service --network=softuni\_homework nginx

# there is another error, but the service is registered



# scaling the services

docker service scale nginx\_service=3

docker service scale test\_service=2

A black screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated