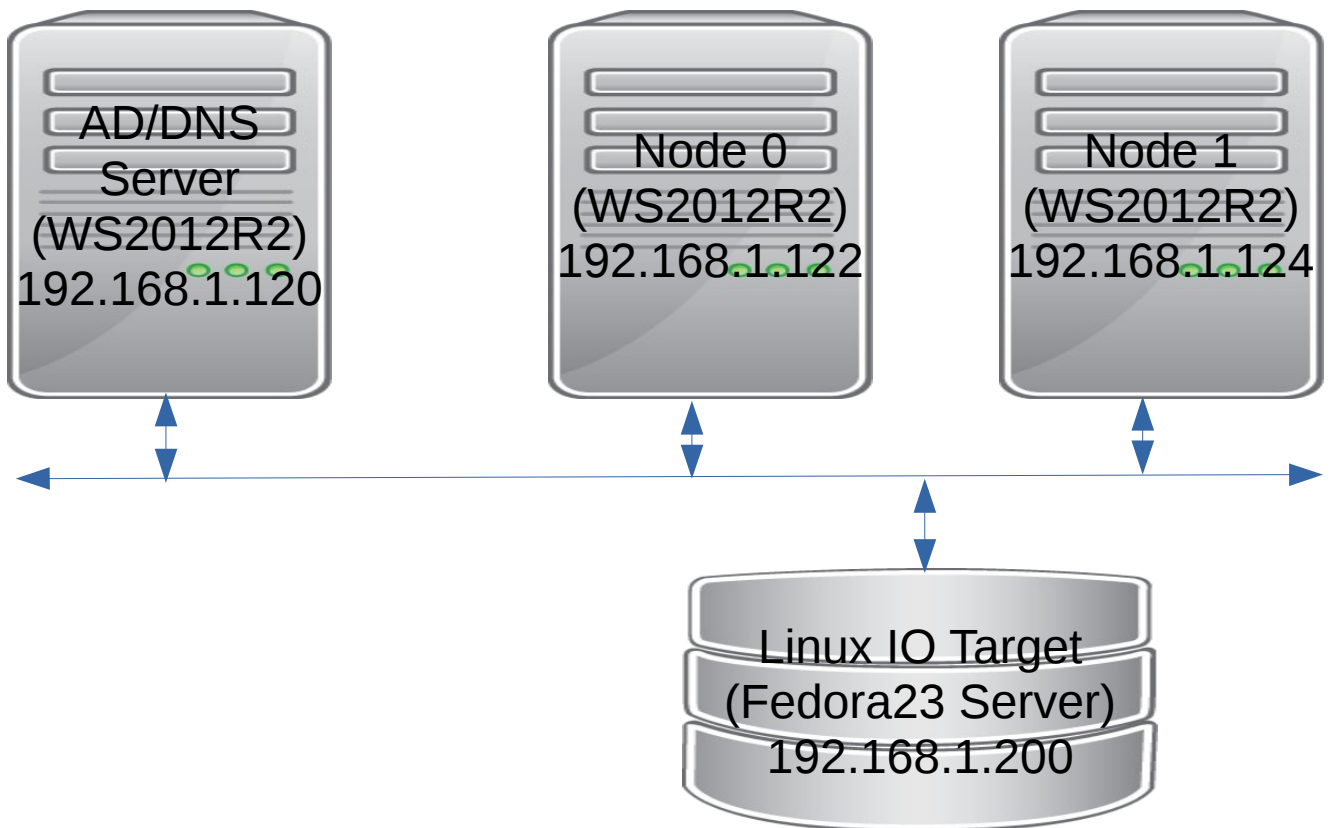My current MS Failover Clustering testing configuration looks as follow:

First of all we need to be sure that Linux -IO Target (LIO) has been installed. Try executing Linux-IO Target management shell "targetcli" and if it succeeded then we are fine. If not – go and install it.

Next we have to create and configure iSCSI backend. Here is how it looks on my system:

```
[vrozenfe@jack msfc]$ targetcli
targetcli shell version 2.1.fb42
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

You are not root, disabling privileged commands.

/> ls
o- / ........................................................................................................................ [...]
  o- backstores ........................................................................................................... [...]
  | o- block ................................................................................................ [Storage Objects: 0]
  | o- fileio ............................................................................................... [Storage Objects: 1]
  | | o- disk01 ............................................ [/home/vrozenfe/work/images/disk01.img (10.0GiB) write-back activated]
  | o- pscsi ................................................................................................ [Storage Objects: 0]
  | o- ramdisk .............................................................................................. [Storage Objects: 0]
  o- iscsi .................................................................................................... [Targets: 1]
  | o- iqn.2016-03.local.server:sas ........................................................................... [TPGs: 1]
  |   o- tpg1 .................................................................................... [no-gen-acls, no-auth]
  |     o- acls .................................................................................................. [ACLs: 2]
  |     | o- iqn.2008-11.org.linux-kvm:5b959a7f-e33f-4229-97b4-da6fe8fb7062 ............................. [Mapped LUNs: 1]
  |     | | o- mapped_lun0 ........................................................................ [lun0 fileio/disk01 (rw)]
  |     | o- iqn.2008-11.org.linux-kvm:4c377bf7-23b1-3413-a3c1-bc9f5b32a344 ............................. [Mapped LUNs: 1]
  |     |   o- mapped_lun0 ........................................................................ [lun0 fileio/disk01 (rw)]
```

```
|    o- luns ................................................................................................................... [LUNs: 1]
|    | o- lun0 .................................................... [fileio/disk01 (/home/vrozenfe/work/images/disk01.img)]
|    o- portals ............................................................................................................. [Portals: 1]
|      o- 0.0.0.0:3260 ................................................................................................... [OK]
o- loopback ................................................................................................................ [Targets: 0]
o- vhost ...................................................................................................................... [Targets: 0]
```

There are a lot of usefull step-by-step information regarding to building iSCSI target with
targetcli utility, one of them can be found here:
http://www.server-world.info/en/note?os=CentOS_7&p=iscsi

However there is one thing that probably requires some explanation. When we create ACLs
we need to provide correct IQNs like:

iqn.2008-11.org.linux-kvm:5b959a7f-e33f-4229-97b4-da6fe8fb7062 and
iqn.2008-11.org.linux-kvm:4c377bf7-23b1-3413-a3c1-bc9f5b32a344

where 'iqn.2008-11.org.linux-kvm:' is a default prefix that QEMU uses if the initiator name has not been
specified, and "5b959a7f-e33f-4229-97b4-da6fe8fb7062 " or "5b959a7f-e33f-4229-97b4-da6fe8fb7062 " are the UUIDs
associated with the relevant VMs. It is simply how the QEMU iSCSI layer builds INQ names by
concatenating two strings.

Next we need to start working on Windows setup.
First of all we need to setup a AD/DNS Server. In my case I used a physical WS2012R2 system for
this purpose, but it can be a VM as well. Then I created two WS2012R2 virtual machines. For the
reference, please see below the contents of both command files I use for running my two-node
failover clustering setup:

```
[vrozenfe@jack fc]$ cat fc0.sh
#!/bin/sh

IMG='/ws2012r2_fc0.qcow2'
QEMU=/home/vrozenfe/work/upstream/qemu/x86_64-softmmu/qemu-system-x86_64
DISC='iscsi://192.168.1.200:3260/iqn.2016-03.local.server:sas/0'
VNC='-vnc 0.0.0.0:5'

sudo $QEMU -cpu qemu64,+x2apic,family=0xf,hv_vapic,hv_spinlocks=0xfff,hv_time -boot c -m
1G -smp 4,maxcpus=4,cores=2,threads=1,sockets=2 -usbdevice tablet -drive
file=/home/vrozenfe/work/images$IMG,if=none,id=drive-ide0-0-
0,cache=off,werror=stop,rerror=stop -device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-
0,id=ide0-0-0,bootindex=1 -netdev tap,id=hostnet0 -device
e1000,netdev=hostnet0,mac=26:3A:42:3F:5B:70,id=net0 -boot dc -uuid 5b959a7f-e33f-4229-97b4-
da6fe8fb7062 -rtc-td-hack -global kvm-pit.lost_tick_policy=discard -monitor stdio -name FC0
-enable-kvm -vga cirrus $VNC -drive
file=$DISC,if=none,media=disk,format=raw,rerror=stop,werror=stop,readonly=off,aio=threads,cac
he=none,cache.direct=on,id=drive-hotadd,serial=sas-test -device virtio-scsi-pci,id=scsi-hotadd
-device scsi-block,drive=drive-hotadd,id=hotadd,bus=scsi-hotadd.0,bootindex=2
```

[vrozenfe@jack fc]$ cat fc1.sh

```sh
#!/bin/sh
IMG='/ws2012r2_fc1.qcow2'
QEMU=/home/vrozenfe/work/upstream/qemu/x86_64-softmmu/qemu-system-x86_64
DISC='iscsi://192.168.1.200:3260/iqn.2016-03.local.server:sas/0'
VNC='-vnc 0.0.0.0:7'
```

sudo $QEMU -cpu qemu64,+x2apic,family=0xf,hv_vapic,hv_spinlocks=0xfff,hv_time -boot c -m 1G -smp 4,maxcpus=4,cores=2,threads=1,sockets=2 -usbdevice tablet -drive file=/home/vrozenfe/work/images$IMG,if=none,id=drive-ide0-0-0,cache=off,werror=stop,rerror=stop -device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0,bootindex=1 -netdev tap,id=hostnet0 -device e1000,netdev=hostnet0,mac=26:3A:42:3F:5B:72,id=net0 -boot dc -uuid 4c377bf7-23b1-3413-a3c1-bc9f5b32a344 -rtc-td-hack -global kvm-pit.lost_tick_policy=discard -monitor stdio -name FC1 -enable-kvm -vga cirrus $VNC -drive file=$DISC,if=none,media=disk,format=raw,rerror=stop,werror=stop,readonly=off,aio=threads,cache=none,cache.direct=on,id=drive-hotadd,serial=sas-test -device virtio-scsi-pci,id=scsi-hotadd -device scsi-block,drive=drive-hotadd,id=hotadd,bus=scsi-hotadd.0,bootindex=2

Then we have specify network settings for both VM and add them to domain.
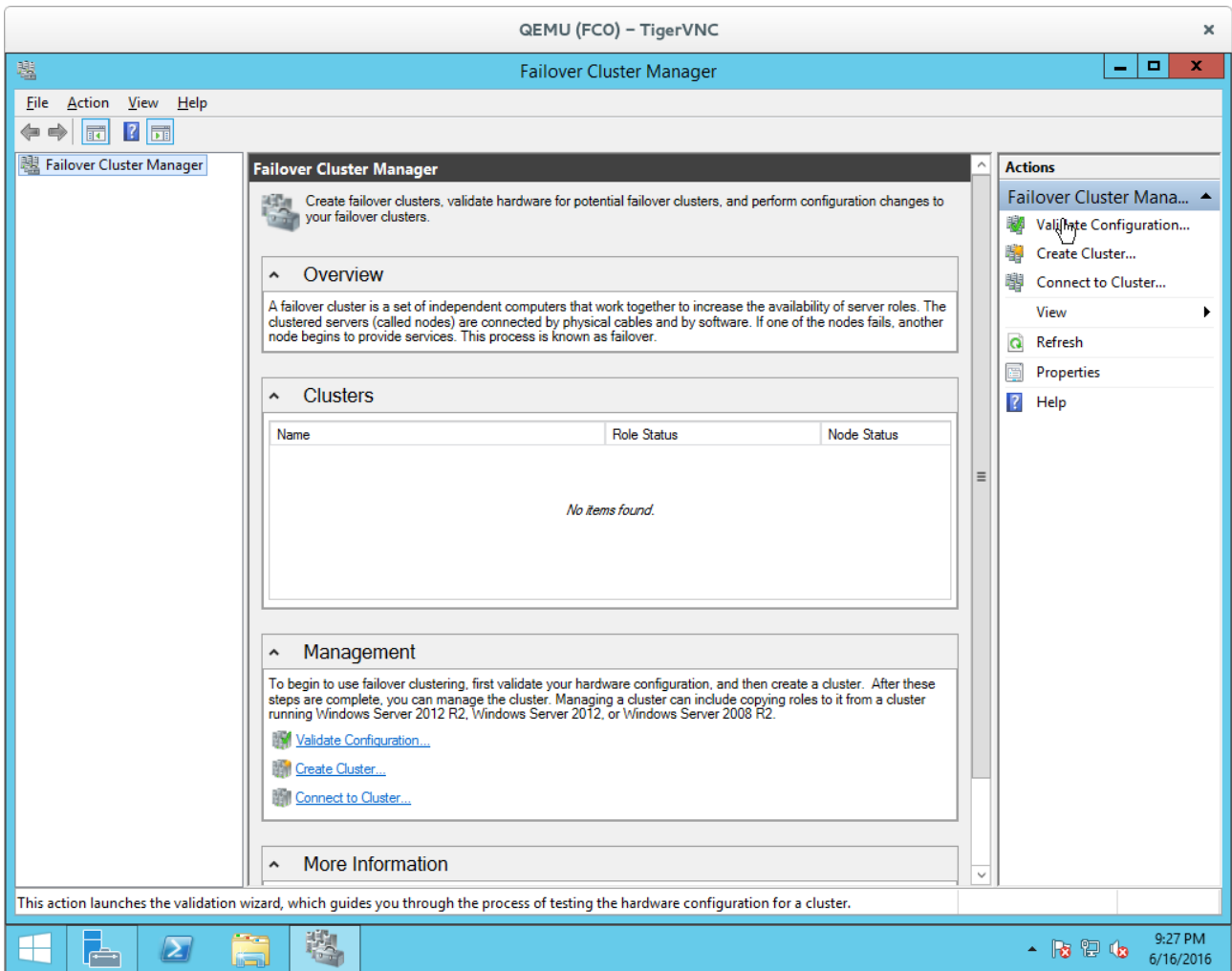
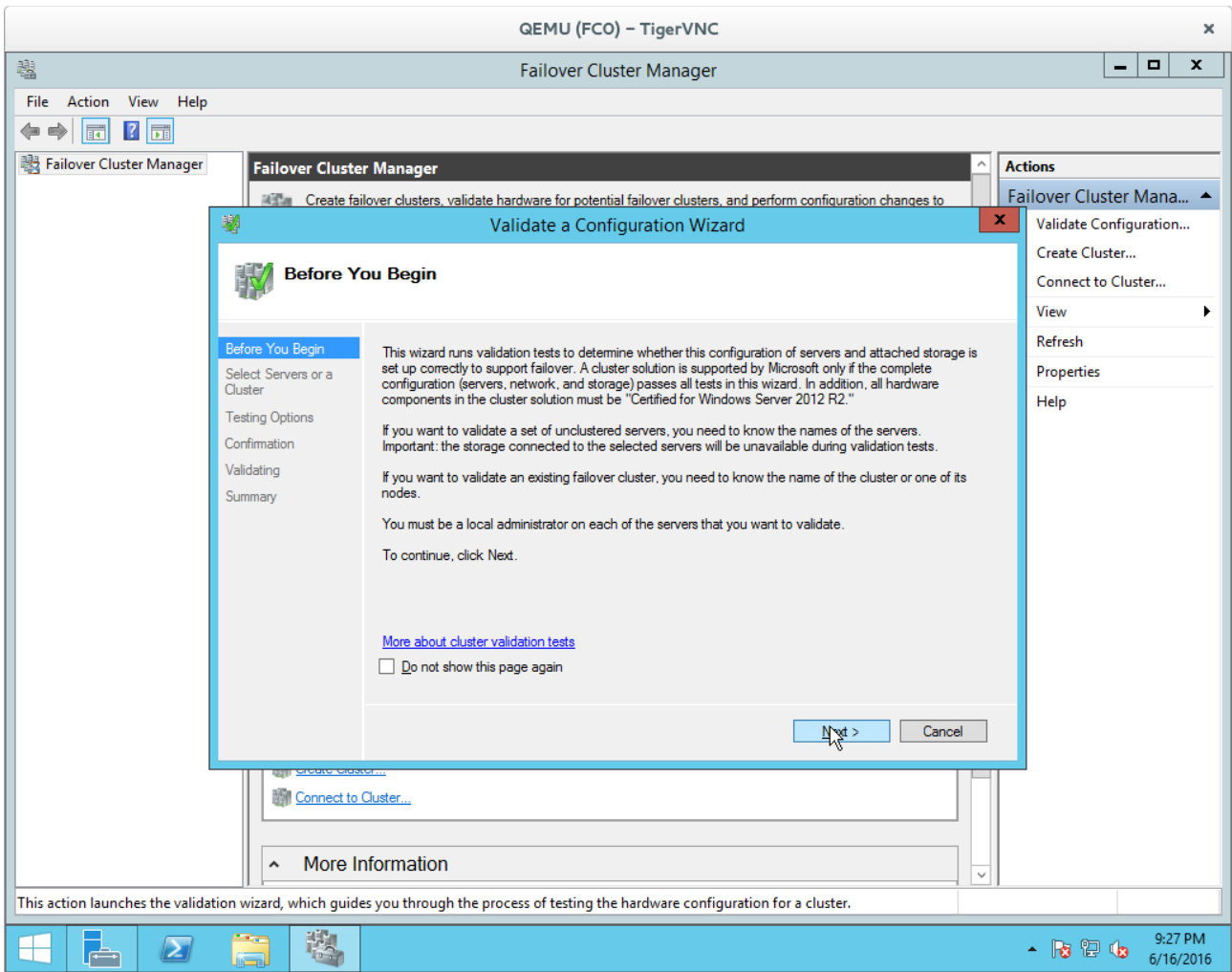This is how Node0 network configuration looks like:
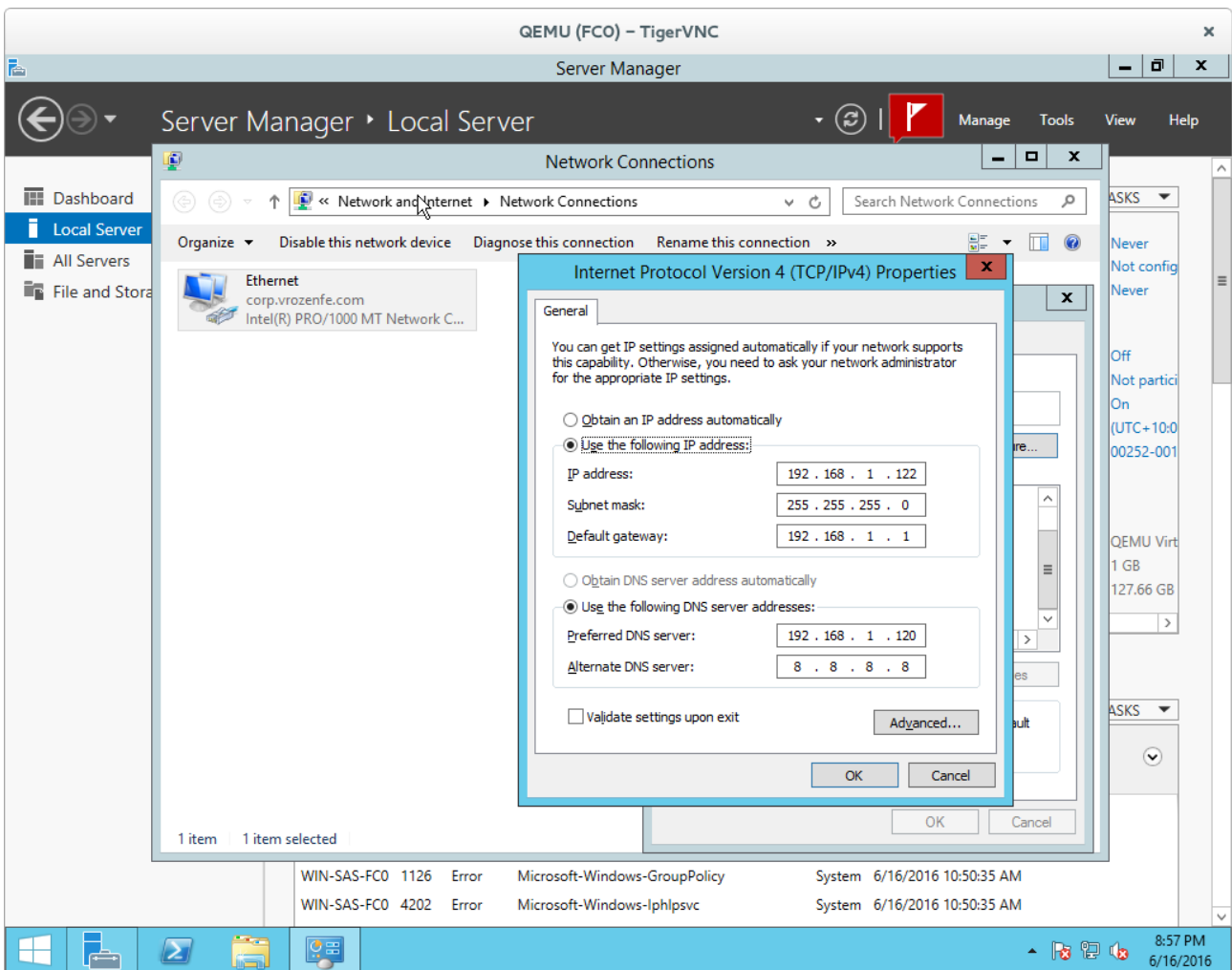
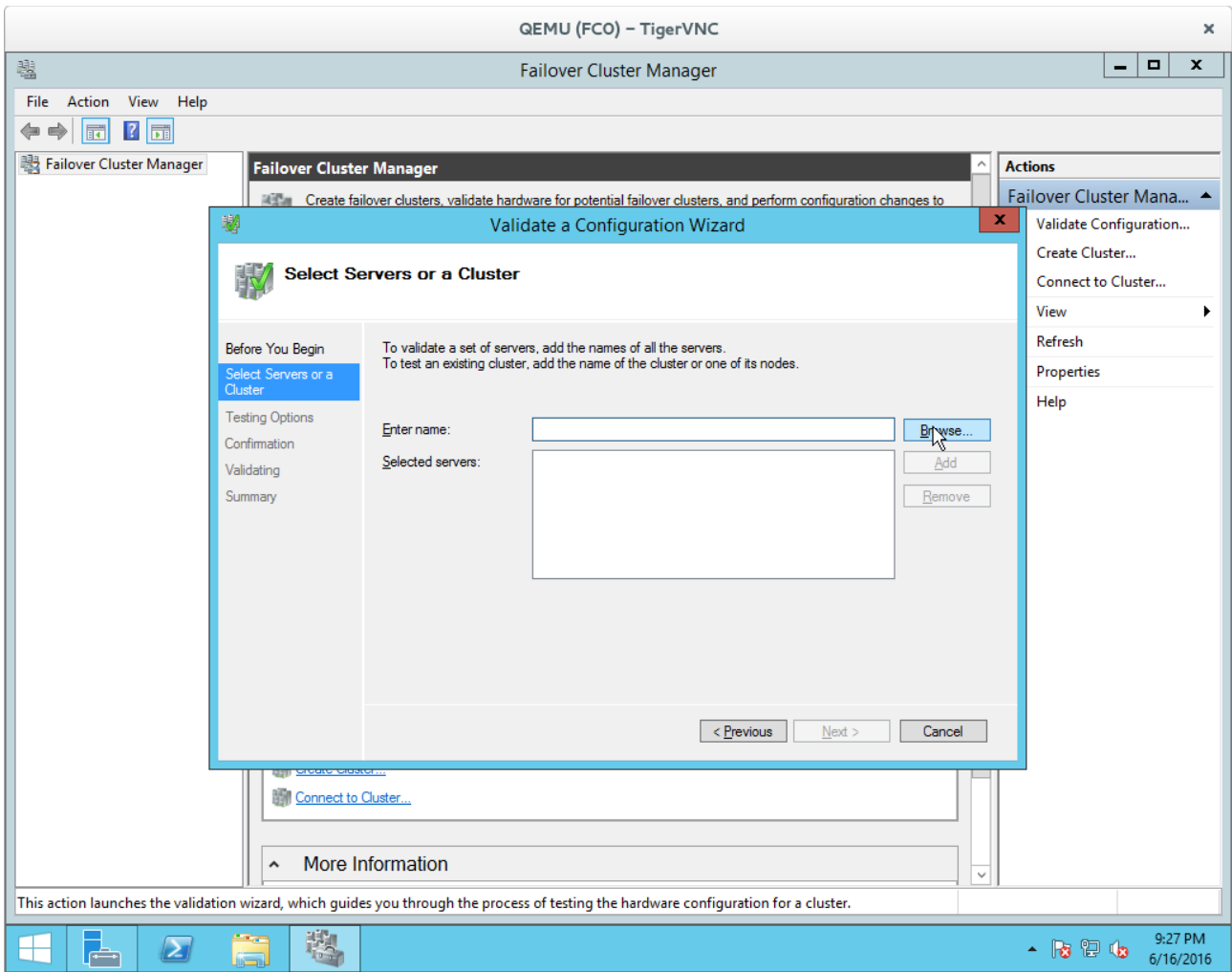And finally we have to add Failover Clustering feature on both nodes:
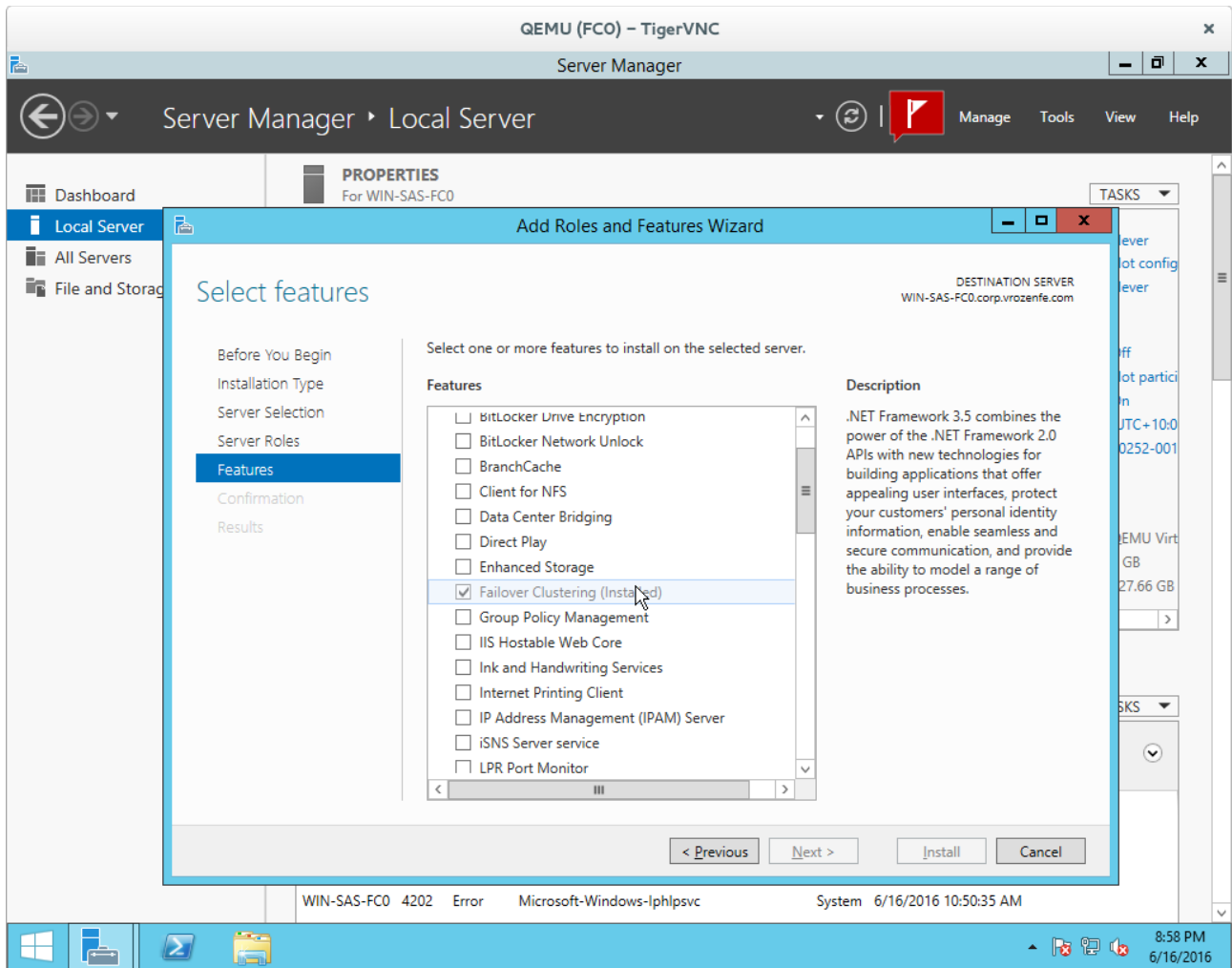
After that stage we should be able to start Faiover Clustering Validation Wizard.
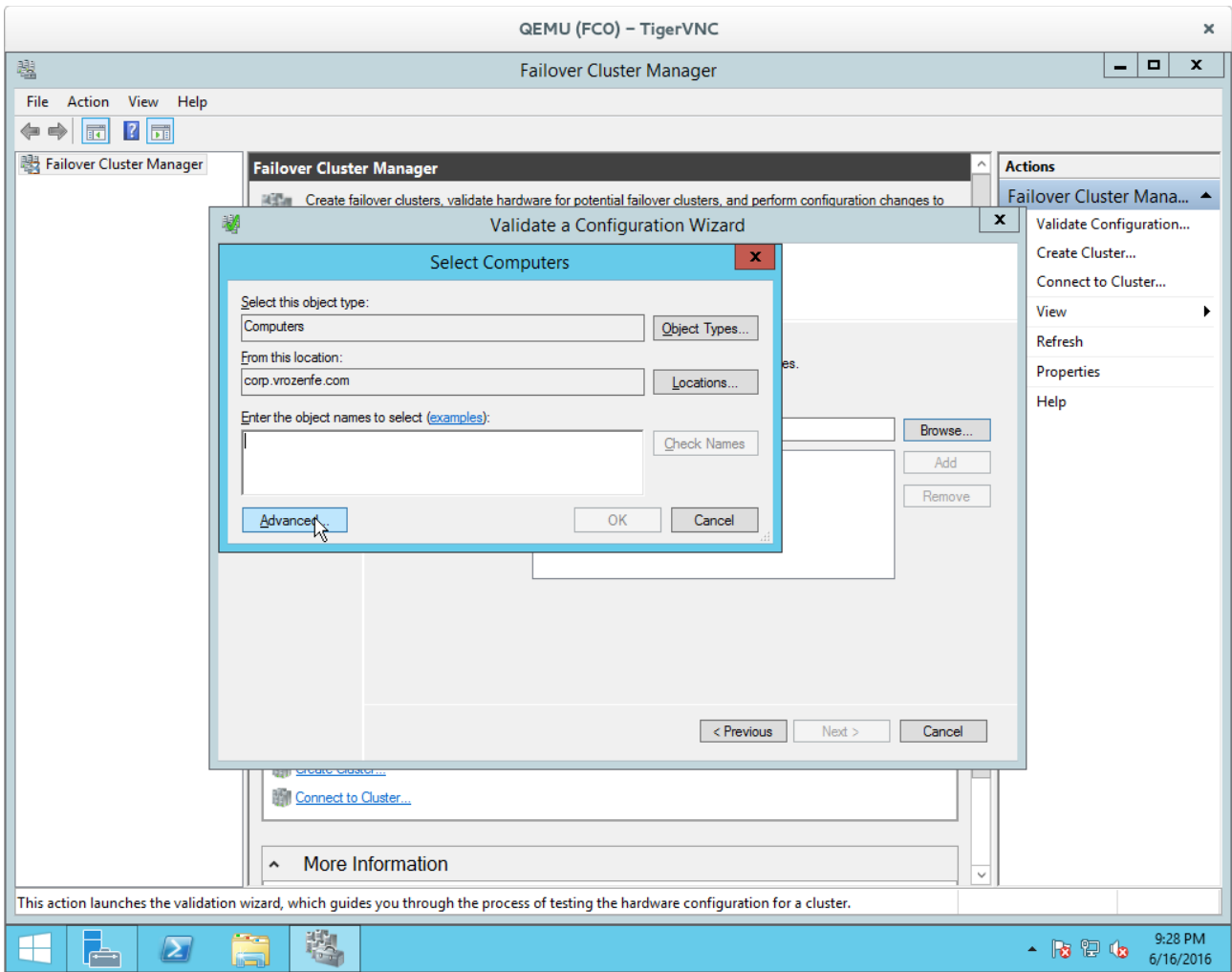

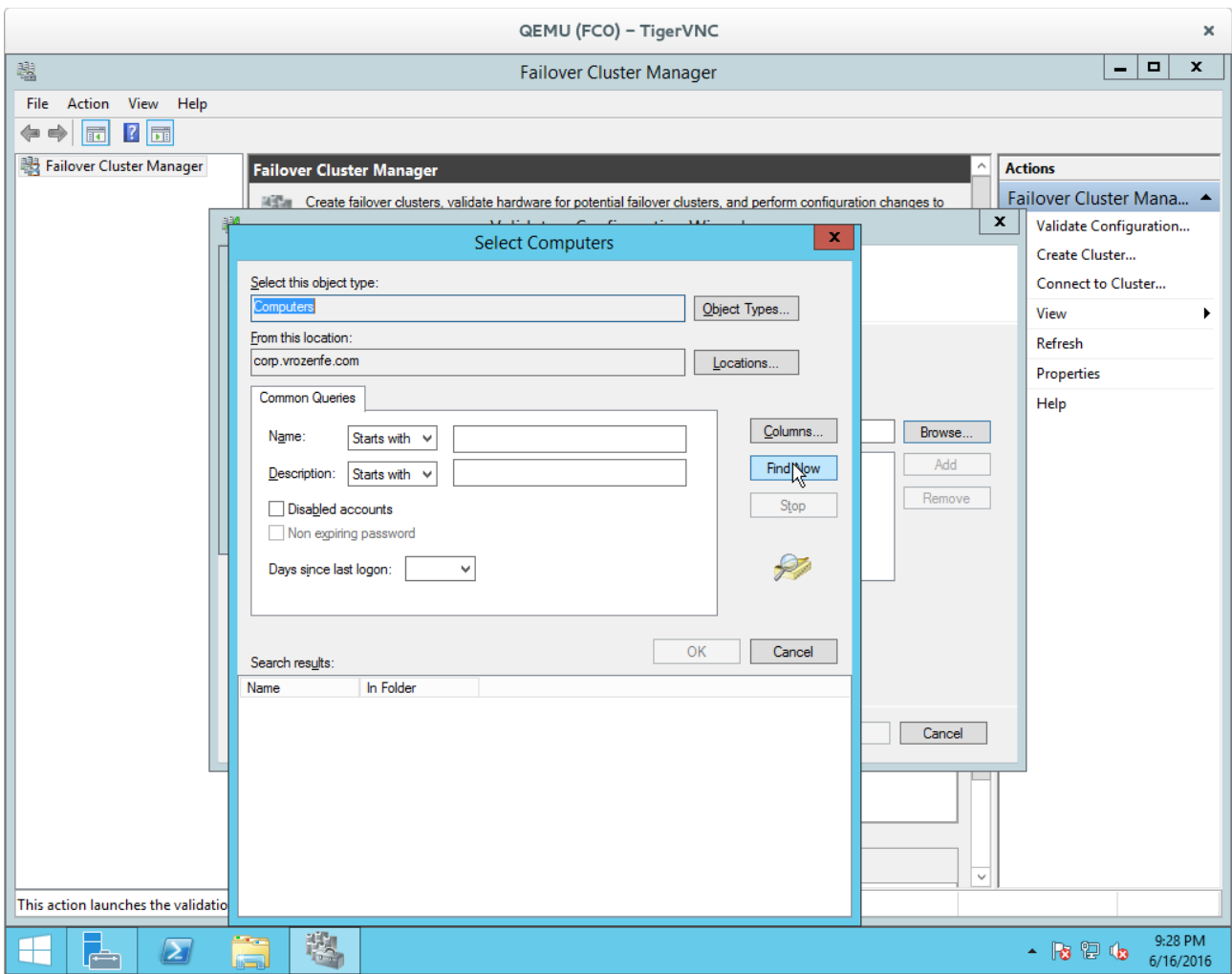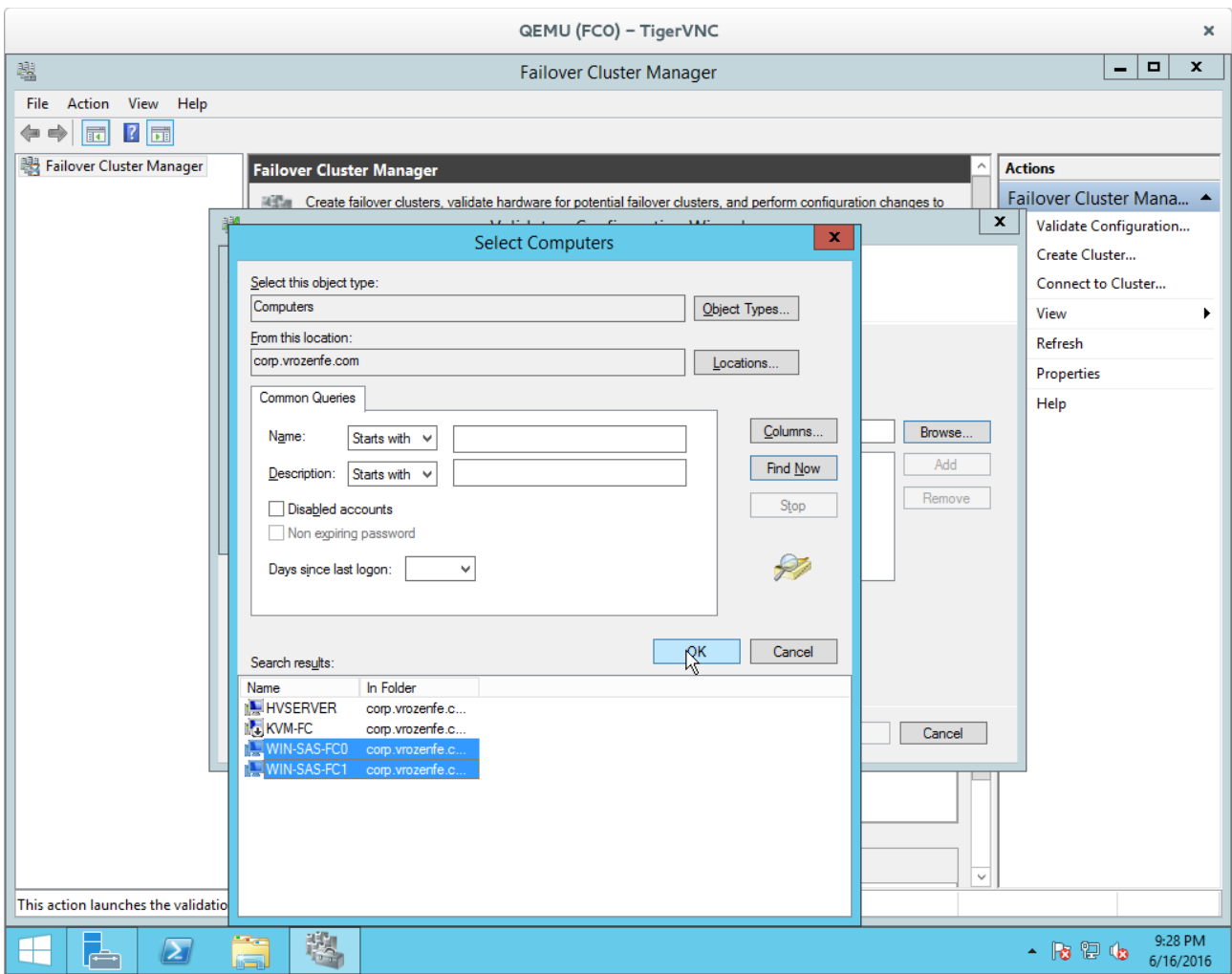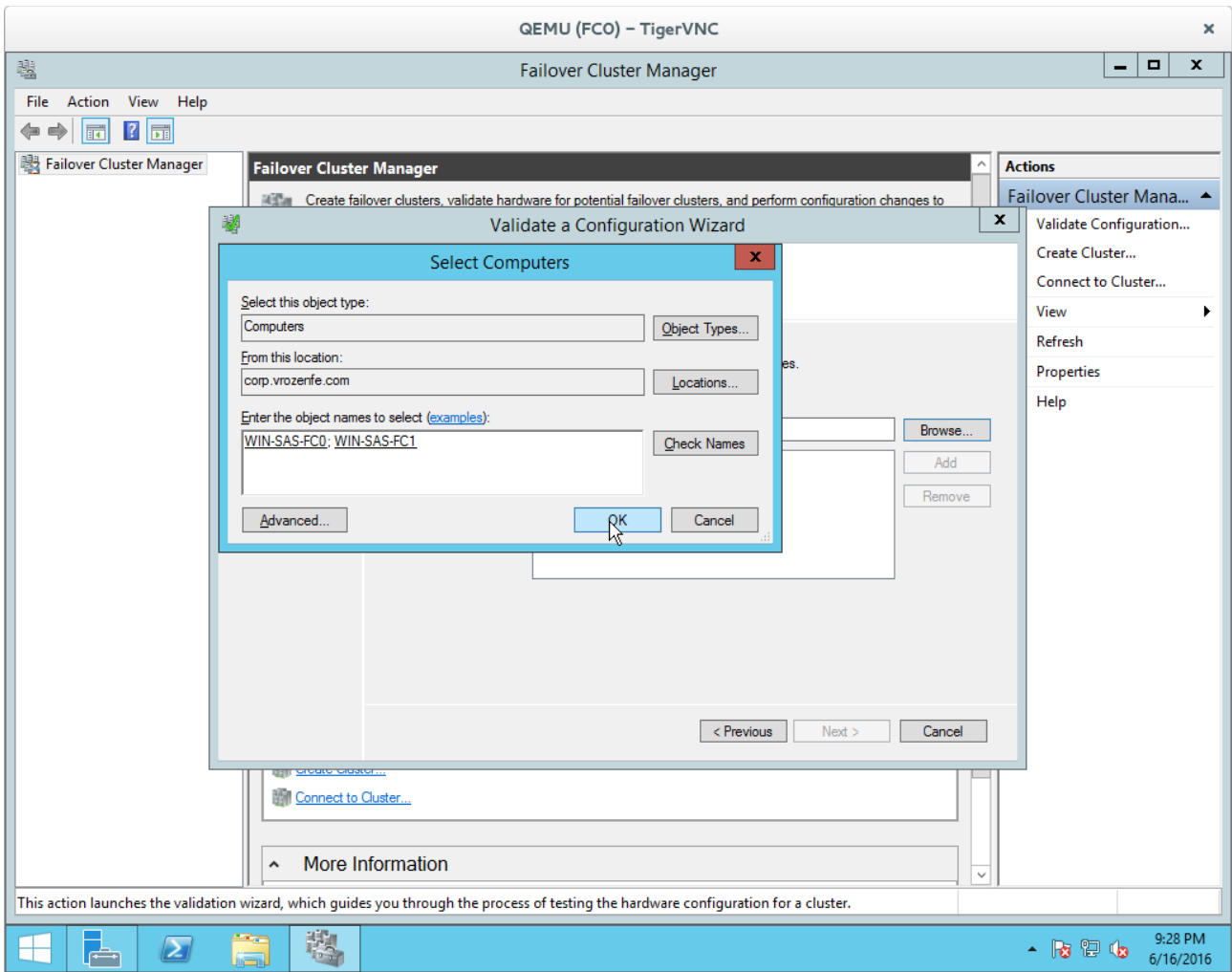We have to start with "Validate Configuration" option.
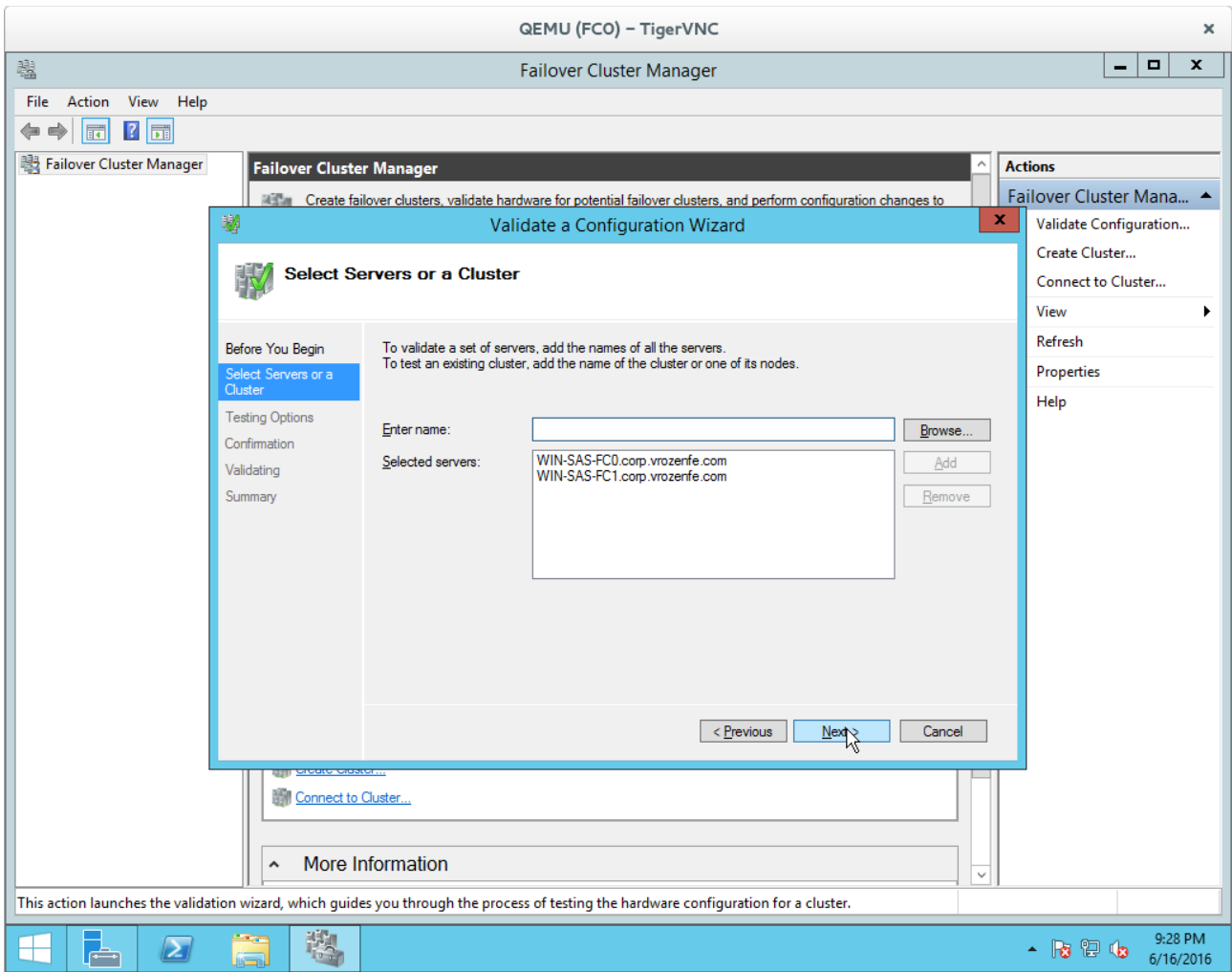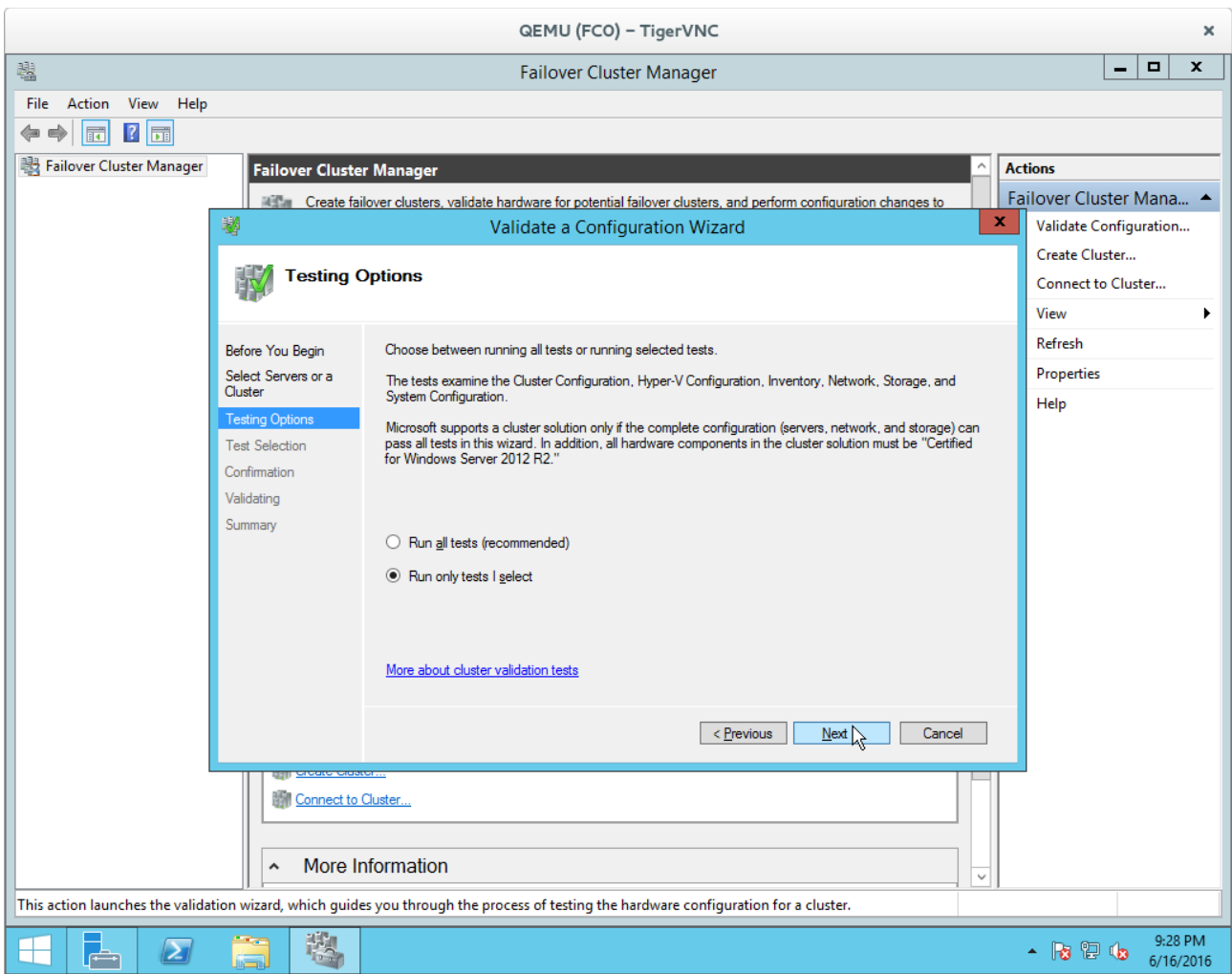
Press "Next"

Go to "Browser".
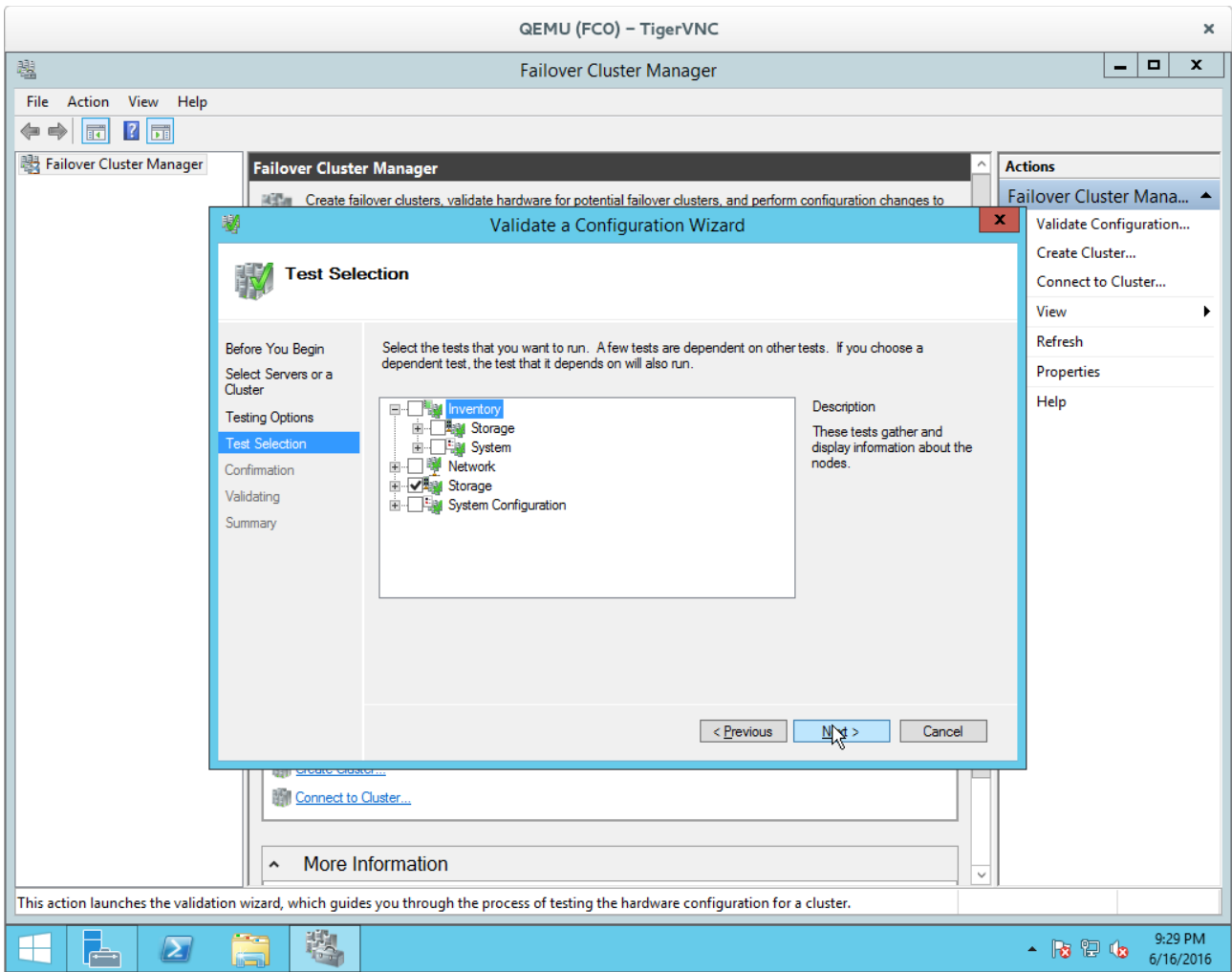
"Advanced".

Let the Wizard to find the relevant candidates.

And choose our two nodes.

Failover Cluster Manager

File   Action   View   Help

Failover Cluster Manager

Failover Cluster Manager

Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to

Validate a Configuration Wizard

Select Servers or a Cluster

Before You Begin

Select Servers or a Cluster

Testing Options

Confirmation

Validating

Summary

To validate a set of servers, add the names of all the servers.
To test an existing cluster, add the name of the cluster or one of its nodes.

Enter name: [   ]   Browse...

Selected servers:
WIN-SAS-FC0.corp.vrozenfe.com   Add
WIN-SAS-FC1.corp.vrozenfe.com
                                       Remove

< Previous   Next >   Cancel

Create Cluster...

Connect to Cluster...

∧   **More Information**

This action launches the validation wizard, which guides you through the process of testing the hardware configuration for a cluster.

Actions

Failover Cluster Mana...

Validate Configuration...

Create Cluster...

Connect to Cluster...

View   ▶
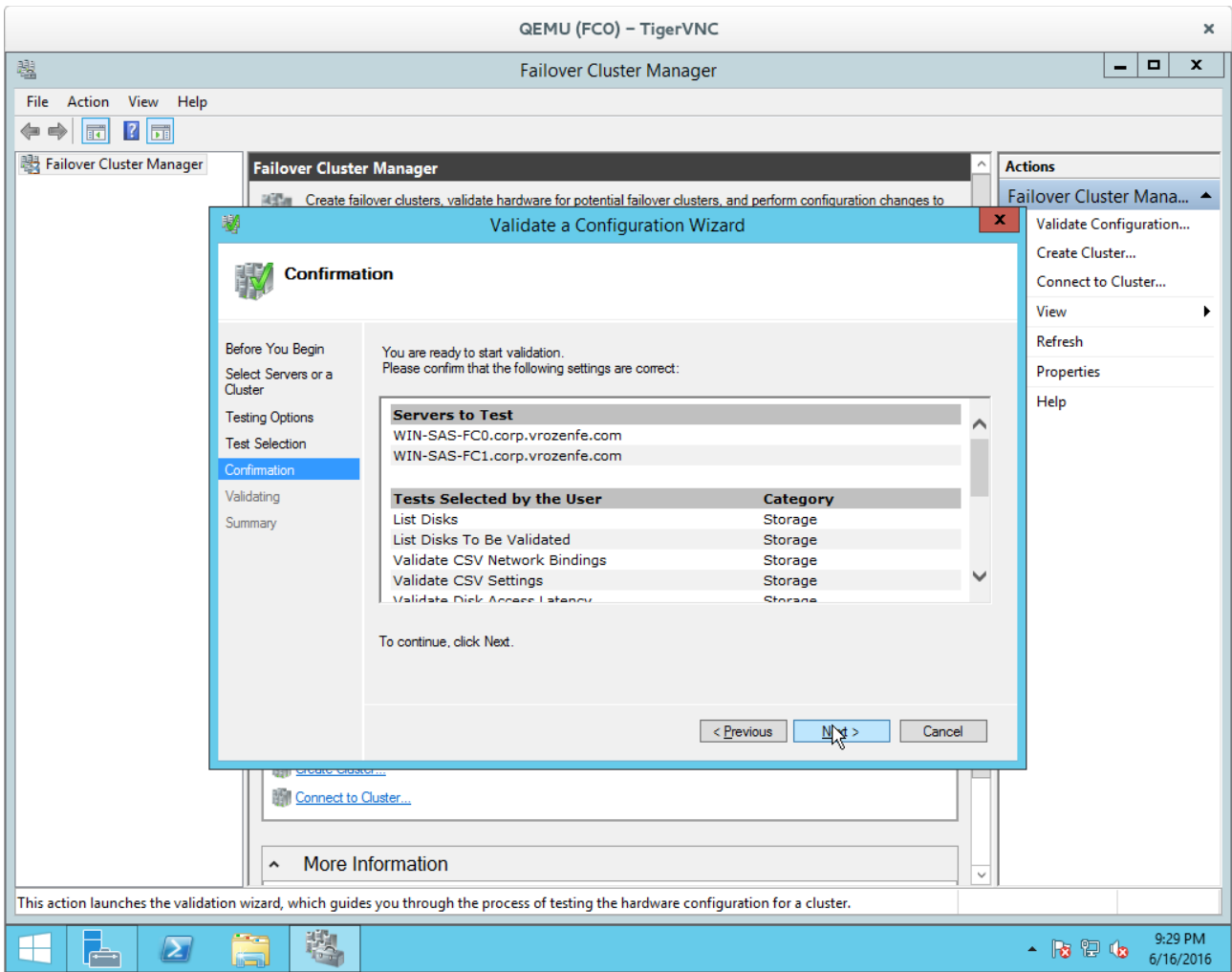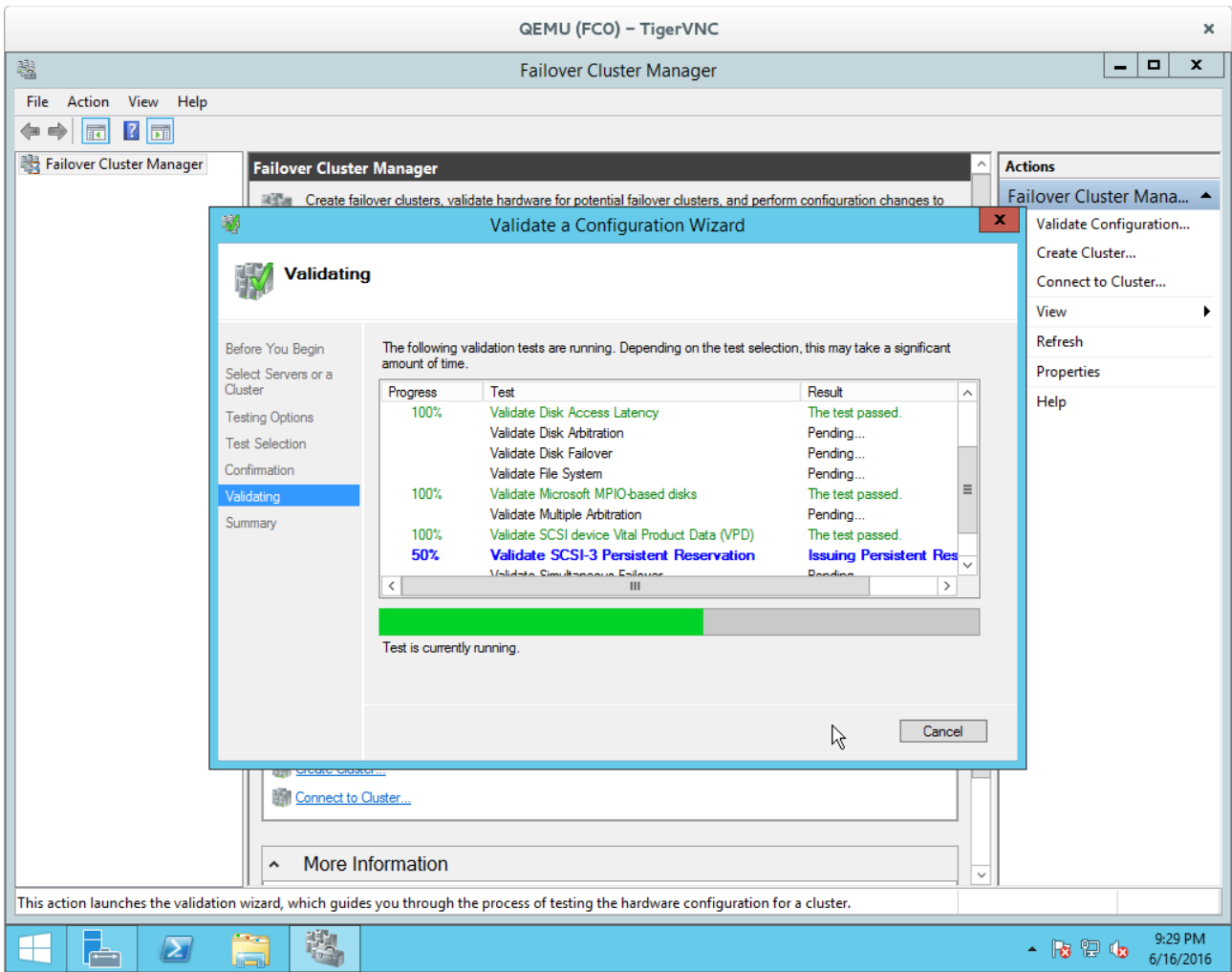
Refresh

Properties

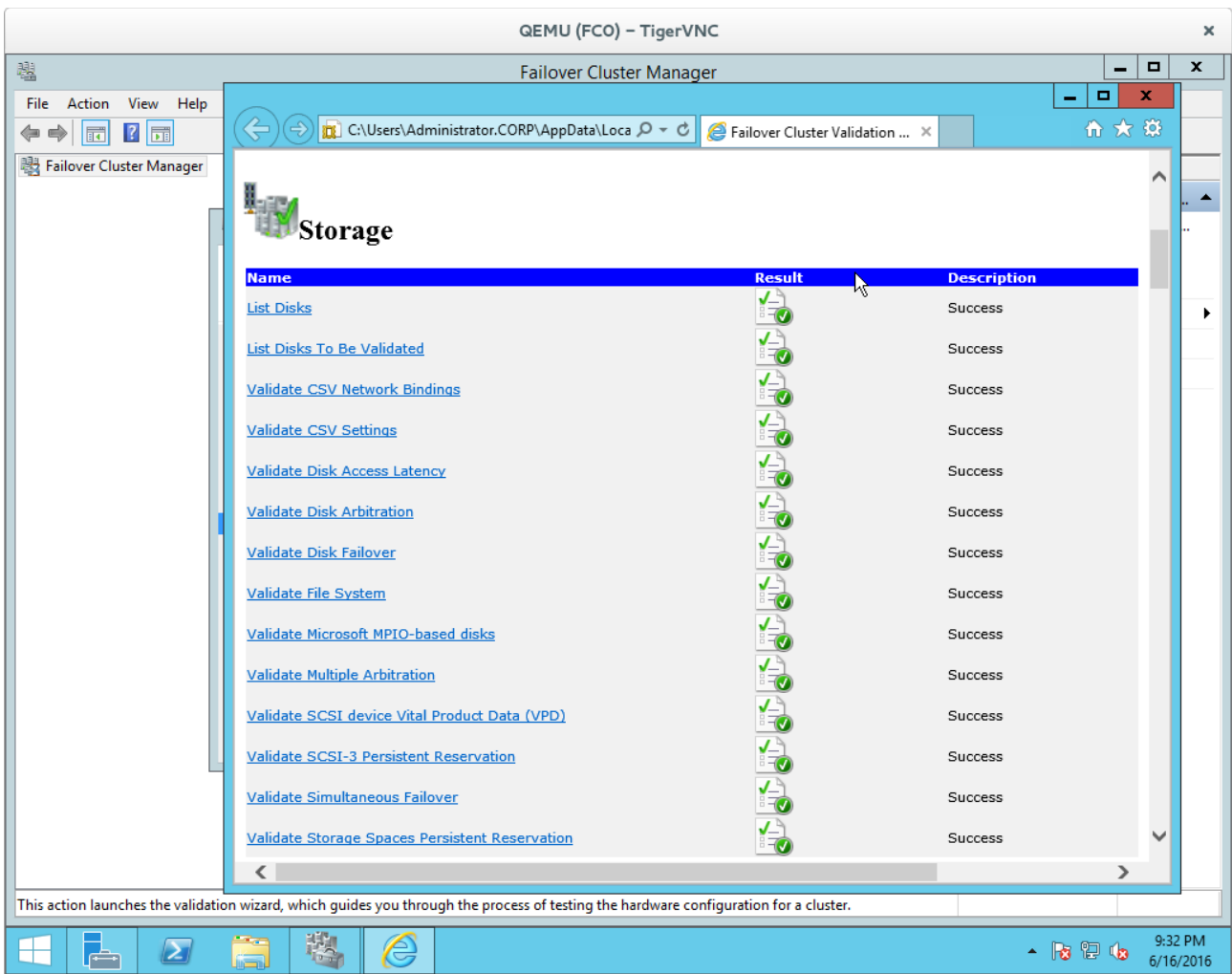Help

9:28 PM
6/16/2016

Run only selected tests this time.

Disable all options except for "Storage".

Run the test.

And get results.