Maksim Paderin

ANALYSIS OF SERVER CLUSTERING ITS USES AND IMPLEMENTATION

Bachelor’s thesis
Information Technology

2017

XAMK

South-Eastern Finland
University of Applied Sciences

XAMK
South-Eastern Finland
University of Applied Sciences

Author (authors)	**Degree**	**Time**
Maksim Paderin	Information Technology	December 2017

Title	
Analysis of server clustering, its uses and implementation	73 pages 0 pages of appendices

Commissioned by

South-Eastern University of Applied Sciences

Supervisor

Matti Juutilainen

Abstract

Clustering became an extremely important networking technology in the latter days. It perfectly coexists with other trending IT technologies and concepts such as virtualization, cloud computing and IoT.

The goal of this study is to learn more about clustering as a concept, about co-existing technologies, and about operating systems which could help to form networks, and, specifically, clusters. Another goal of the study is to apply gained knowledge to practice and implement a working cluster.

The practical part is targeted on the use of DigitalOcean and Amazon Web Services technologies in synergy with two very different operating systems, Linux and Windows.

Keywords

clustering, networking, servers, vmware, windows, linux, clouds, virtualization, docker, digitalocean, aws

CONTENTS

1 INTRODUCTION

The general idea and aim of the thesis is research what the clustering technology consists of, study its terms and aspects, and to create and run a server cluster after the choosing the best methods and technologies to do that. The role of server is web server.

As mentioned above, desired goal of the thesis is getting a deeper knowledge, both theoretical and practical, in server clustering. The second important aim of writing this thesis is to learn manage time properly in and put tasks in order. The additional aim of the thesis is to code an application to run on a cluster. It also benefits my skills as a developer.

The thesis consists of four main chapters. The first one is a theoretical study on what a cluster is, its advantages and disadvantages, and why it can be useful in real life. The second one is also theoretical, it covers the principles of clusters' workflow in general. The third one is about researching differences between different software: which virtualization/cloud technologies to use, which OS to select and, of course, deciding which operating system and software suit the best for the needs of the project. It also includes description of methods and technologies specific to certain OSs. The fourth part is a description of the practical implementation, in other words, it lists all the steps of configurations, settings and code.

Main reference in everything related to Windows, including Microsoft Azure, is MSDN (Microsoft, 2017), for VMWare it is VMWare Documentation Center (VMWare, 2017). Main reference for practical part is AWS website (Amazon Web Services, Inc, 2017). Technologies change over time, and there aren't a lot of books written during the last 5-10 years, that's the main reason of using web portals over books.

2 SERVER CLUSTERING CONCEPTS

In this part I will cover the theory related to clusters. This includes network concepts, server concepts and other complex principles specific to clusters.

Terms and brief overview of technologies related to clustering will be also covered in this part.

It's possible to find needed terms on the Internet, a proper guide is present at DigitalOcean website: https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols.

2.1 Clusters as technology

The core term of server clustering is server itself. It is a powerful computer able to provide demanding services over a network during a long period of time. The proper server should be durable and oriented on providing needed service during the maximum time possible without interruptions and lose of data.

The server cluster is a set of servers which are connected to each other and communicate to provide highly redundant and available services. The main difference between a cluster and just a group of servers is the fact that each server within a cluster does the same tasks, therefore, clusters are generally used for demanding applications and services, because such tasks will have more processing power to use.

The main goal of using clustering is to establish failure-tolerant network, implement quality of service improvements and also structurize the servers within a network. Example technologies used in clustering are, for example, server consolidation, load balancing etc.

2.2 Cluster roles and architecture

A typical cluster consists of server manager (supervisor), servers themselves (server nodes) and networking equipment. In complex solutions, there could be any number of clusters within the same LAN.

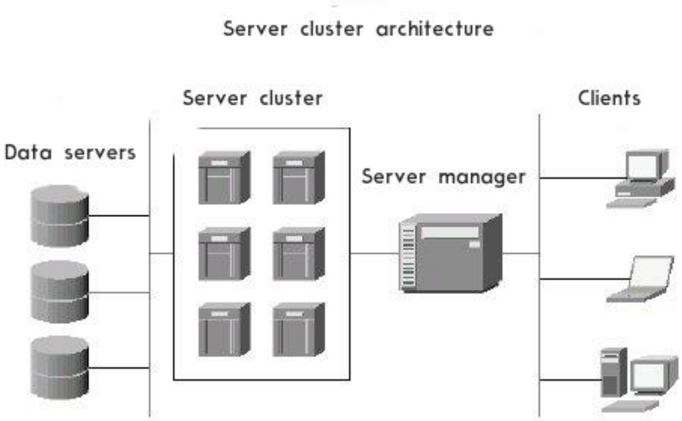


Figure 1. Part of LAN representing typical cluster architecture (Paul Chin, 2004)

Very important term is LAN. It is a network with a relatively small area of coverage which is considered as the smallest possible network. It connects to other LANs and thus forms WANs, and, in the highest extent, the Internet. The term Internet will mean the theoretical group of all other networks except the one given.

LAN usually consists of networking equipment, couple of servers and client (user) devices. It can be wireless (WLAN) or wired, although I will not cover wireless solution in this project. In other words, the term LAN will mean only the wired option.

User is any device which doesn’t provide any services within the given LAN, but uses them. In this project client devices will consist of desktop computers as long as I will concentrate on enterprise appliances and not cover mobile devices.

The role of server supervisor is to manage nodes and be able to transfer data from cluster and to it. During the process of configuration, a special software is installed to nodes in order to be able to be managed by supervising server, but with the fact that this server will see all the nodes as one big cohesive machine.

This special software is called cluster management agent. It provides abstraction. This is a special technique which principle is in hiding or removing all unneeded elements and presenting only the essential data. In clustering, all the cluster members are represented as one big set with clear information about status of them, and cluster settings which are easy to configure (this is the way cluster is shown in cluster management software) while other information such as hardware details is hidden or even not present. Examples of cluster management agents are MySQL cluster management agent and Oracle Managent Agents. There are plenty of others as well.

The role of the server nodes is very straightforward. A cluster uses their processing power (hardware power) to run applications, store data, provide continuing services etc. Nodes will exchange a lot of data among each other, therefore hardware should be fault-tolerant.

Network equipment in the easiest case can consists only of cables and one switch, but when there are several clusters and more complex network structure, use of other devices is encouraged. Most used networking devices in general are switch and router. All of them will almost always be used in enterprise networks with clustering, though router is not obligatory to have.

Switch is a network device whose main task is to receive, process and forward data packets within the network. It is capable of multicasting and this differentiates it from a hub, an obsolete “version of switch” which will not be covered in this document.

Router is a network device capable of inter-networking communication. It is also capable of receiving, processing and forwarding data.

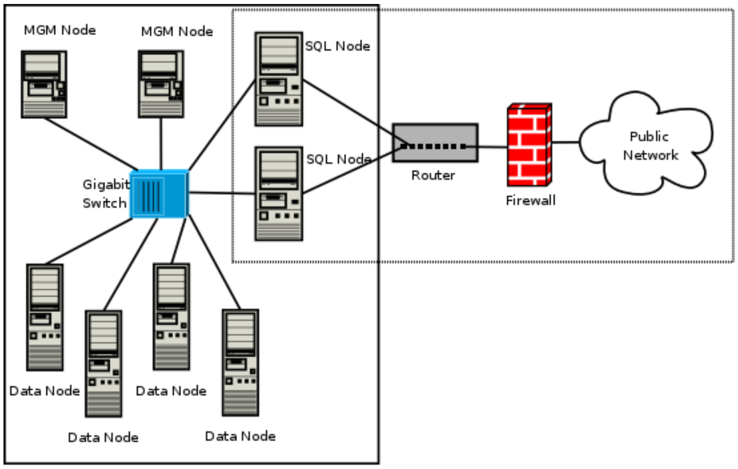


Figure 2. Example of a more complex LAN with clusters (no author)

In LAN pictured in Figure 2 there are three clusters (data, management and database ones) which are connected with a powerful switch. LAN connects to the internet via a router with a firewall enabled. Cluster manager is not present in the figure as long as its main point is to show the use of switches and routers in clustering. Despite that this network is relatively complex, it’s still not an enterprise-level example.

2.3 Hierarchical internetworking model and clustering

HIM is a network design model which was proposed by Cisco corporation. According to this model, networks should be designed and build with logical division to core, distribution and access layers.

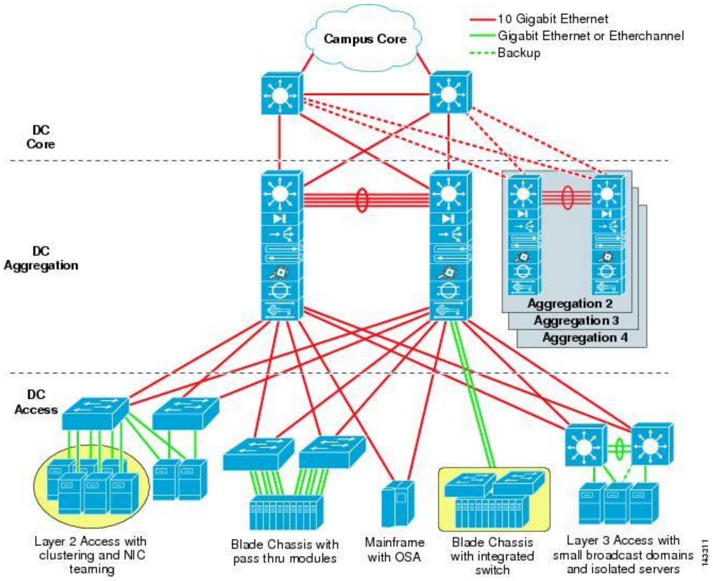

Campus Core
10 Gigabit Ethernet
Gigabit Ethernet or Etherchannel
Backup
DC Core
DC Aggregation
Aggregation 2
Aggregation 3
Aggregation 4
DC Access
Layer 2 Access with clustering and NIC teaming
Blade Chassis with pass thru modules
Mainframe with OSA
Blade Chassis with integrated switch
Layer 3 Access with small broadcast domains and isolated servers
143311

Figure 3. An example of network built following a hierarchical internetworking model (Cisco)

The goal of the core layer is to provide extremely fast forwarding of data. It consists of very powerful switches, routers and the best possible cables (10+Gbit Ethernet). The best possible technologies are preferred, all the devices should be new and perfectly operating not to slow down the data transfer.

The task of the distribution layer is the brain of the network. All the routing, security, domain settings and policies are applied and managed from this layer. Access layer is the layer client nodes connect to. This layer focuces on having the best possible connectivity such as Ethernet ports, servers to connect to etc. As can be seen in Figure 3, clusters belong to the access layer. They are not involved into high-speed transferring of data and they do not process any networking logics, but their task is to provide access to the services.

2.4 Cluster benefits

The benefits of clusters (in case of the functions needed being enabled) in comparison to non-cluster options:

a) Higher performance
b) Load balancing
c) Much better fault tolerance
d) Scalability
e) Less need for maintenance
f) Ease of server management
g) Server consolidation

Higher performance is the core aspect of clustering technology. Several servers forming a cluster are considered as one machine by other network devices. Therefore processing power is almost the same as the sum of processing power of separate servers, i.e. there is less overhead.

Load balancing is a feature which is very crucial as well. It helps cluster nodes to share their hardware usage data to allow a manager to evenly distribute workload among all the server nodes. This greatly reduces the hardware wear because the feature will prevent situations when one node is loaded at 95% and other ones at

5%. This also helps to reduce downtime, so that there will be no extremely high and low workloads.

Fault tolerance is also one of the main features of clustering. Connected servers immediately take over the tasks from a failed one. Therefore, all servers in a cluster need to shut down at the same time in order to get access error. Complete shutdown is possible in case of power (electricity) failure. Thus, uninterruptible power supplies (UPSs) should be properly installed to the cluster servers.

Scalability is present in clusters due to the fact that the system is easily expandable. If somebody has a cluster and wants to add a new server to it, there is a need just to connect all the cables needed, enable clustering on a server and it will be ready for use. All the other nodes in a cluster will start to communicate with it, and there will be no need to enable anything on them. Scalability also means that the system adjusts to almost everything automatically, if it can afford it.

Network administrators will be happier because of the possibility to spend less time on clusters in comparison to a group of separated servers. As mentioned before, in most cases a cluster tries to repair itself, and usually there is no need to configure something in the whole cluster system in case only one node experiences issues, it can be stopped and configured independently from other nodes. Unfortunately, while being a time-saving technology, clustering is money-spending one, the cost of setting up a cluster is quite high.

There is relative advantage in terms of management. All commands and configurations can easily be shared to all cluster nodes and it saves a lot of time, and also reduces the chance of human error. Unfortunately, server clustering is a complex technology and it leads to specific risks, which are listed below.

Server consolidation is a method reducing configuration and operating costs via decreasing the number of servers. This can be easily achieved via using virtual servers. My aim is to use them in my projecs.

As a whole, these benefits make clustering an extremely useful technology in both simple and complex enterprise solutions, although it has some flaws. I will cover them and explain how to avoid them.

2.5 Cluster limitations and avoiding them

Clusters have the following (and some others) disadvantages:

a) High costs
b) Compatibility issues
c) Physical limitations
d) Complexity

High costs can be avoided by using virtualization. I will use cloud servers, thus I will spend the least amount of money possible, aroud EUR 5-10 for running servers. The whole project will be virtualized. Unfortunately, although virtualization will help in enterprise environment, clustering will still be expensive.

Compatibility issues could also be avoided via pre-choosing the software needed. I will analyze all the options in the chapter four, section six of this project. In an enterprise environment the same method will also help.

Physical limitations will not apply to this project, because it will run in a cloud environment. Obviously, companies should consider this in a real environment.

Complexity leads to the need for a high-qualified specialist to work with clusters. As long as this is my study and practical project and not a real situation, this disadvantage doesn’t affect me.

To sum up, I can say that the disadvantages of clustering almost do not relate to my project, although they will affect companies which want to set up clusters in their networks, and this should be taken into account.

2.6 Failover types

As long as failure protection is one of the most important features of clustering, I will cover this benefit in detail. There are three types of failover methods which can be used in clusters, though not all are preferred.

2.6.1 Hot failover

Hot failover, as all “hot” technologies, means that a service application was designed specifically for cluster needs, and the application can easily switch nodes without interruptions and errors. It’s the most preferred method for all main services of clusters, e.g. financial (banking or e-commerce) apps, continuous web apps etc, although there could be less need for hot failover for less important services.

Hot failover is achieved via very frequent backing up of data. It is extremely expensive and requires a lot of disk space, but hot failover is necessary when the cost of recovery from a potential error is very high. This technology also requires a lot of pre-configurations, the use of the power backup hardware like UPSs (uninterruptible power supplies) and the best networking hardware and equipment possible.

2.6.2 Warm failover

This type of failover is provided by a lot of software creators. It basically means that an app can restart itself automatically with the minimal possible or even no downtime. If software developers offer this type of failover, it is a good idea to use it for non-crucial service, because it will save time and human resources.

Warm failover doesn’t require very frequent backing up of data, but that also leads to possible delays in case of cluster errors. This type of failover is much less expensive to establish than hot failover, but still requires qualified network engineers and high-quality hardware.

2.6.3 Cold failover

This type of failover is not automatical, i.e. it means that a network engineer should power the new node up and start the app there. This type of failover is the cheapest one and the easiest to establish (because it doesn’t require any specific changes), but causes long delays and requires human attention even in case of the simplest possible errors. The main goal of clustering is to provide highly available services, and therefore, it leads to the need of avoiding cold failovers as much as possible.

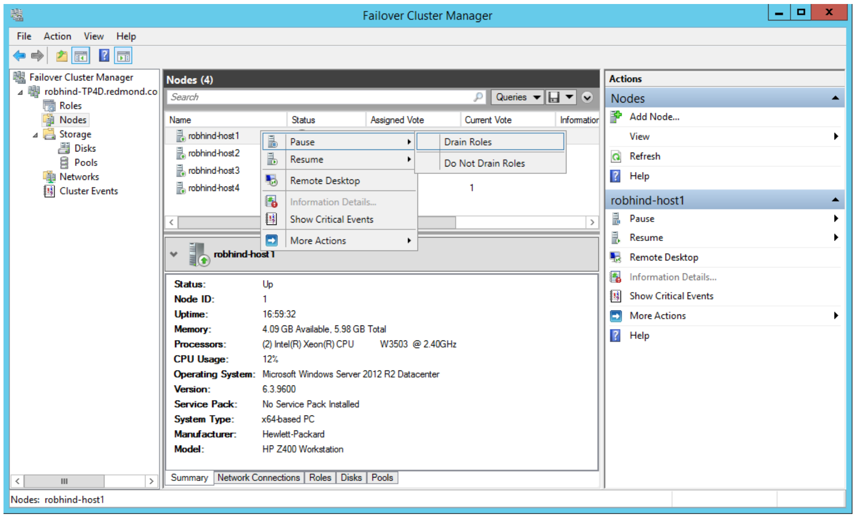
2.6.4 Conclusion for failovers

Downtime losses, building and configuration costs and potential recovery costs must always be considered while setting up a new cluster with a new service. Warm failover is balanced solution, it is not as expensive as hot failover but provides good protection (unlike the cold one), i.e. on average it is used the most in enterprise environment.

2.7 Cluster management

As mentioned before, in order to use a cluster manager to supervise and configure nodes it’s needed to, first of all, enable clustering on them. This will install a special tool called cluster management agent. The agent will be installed on the top of node OS and server as an intermediator between the manager and the node itself, in other words, cluster agents will receive the command from the manager and then send it to nodes.

In normal conditions all the needed configurations and changes can be applied through a server manager. Windows Server provides a special tool called Microsoft Failover Cluster Manager (former Microsoft Cluster Server), although other options can be used as well.



Status: Up
Node ID: 1
Uptime: 16:59:32
Memory: 4.09 GB Available, 5.98 GB Total
Processors: (2) Intel(R) Xeon(R) CPU W3503 @ 2.40GHz
CPU Usage: 12%
Operating System: Microsoft Windows Server 2012 R2 Datacenter
Version: 6.3.9600
Service Pack: No Service Pack Installed
System Type: x64-based PC
Manufacturer: Hewlett-Packard
Model: HP Z400 Workstation

Figure 4. Modern version of Windows Failover Cluster Manager (Microsoft)

Linux, as always, offers a lot of freedom to a network administrator. Usually different programs should be used for each aspect of the configuration, for example separate tools are for parallel configuration of servers, deployment of packages into several nodes and monitoring. I will name and cover them in chapter four, section two.

In case something goes wrong with one or several (not all) nodes, they can be stopped, thus isolated, from properly operating servers, and then configured directly. It will not affect the work of the cluster due to the load balancing feature. Certainly, there could be a need to shut all the server nodes down and configure them, but clustering technology allows to minimize the necessity of such procedures.

3 PRINCIPLES OF CLUSTERS' WORKFOLOW

A very important part of studying and working with technology is to understand how exactly it operates. I will cover the main ideas and terms in this part and the features specific to Windows and Linux in chapter four.

3.1 Quorum as a technology

Quorum is one of the most important terms in clustering. Roughly speaking, it is a joint storage for some or all cluster nodes (depending on a cluster type) which each node with access to it will use to maintain its configuration appropriate for cluster.

Quorum resource helps a cluster to provide two of its main features. Firstly, it stores constantly an updated version of the cluster database and each new / recovered node will compare its state to the state stored in the quorum resource. Secondly, it prevents cluster nodes from splitting caused by node interaction issues which cause the existence of two separate groups with different states within one cluster.

A cluster distinguishes two type of cluster members – nodes and the disk witness (also known is file share witness). A disk witness is a disk (part of quorum) which keeps the cluster configuration.

Quorum can belong to one of the several types. Each type represents the configuration of the quorum and the quantity of failures that a cluster can survive. The important point is the fact that the cluster must stop running, if the number of failures is higher than prescribed for the chosen quorum type in order to prevent further and more crucial errors. This is calculated by the technology called votes. Vote distribution methodology depends on which type of a cluster and quorum are used.

3.2 Quorum types

The quorum types can easily resemble RAID types because basically quorum is an array of a disk, and the storage arrays principles generally work the same way in any storage solution.

The important term for all the types is vote. Vote is a special technology which is used by quorum to determine, if a cluster must stop operating or not. There are a lot of different quorum types depending on the operating system, hardware provider and the software provider used. For example, IBM provides its own quorum types, if cluster uses specific IBM technologies (IBM, Quorum types).

3.3 Prevention of data corruption

There is a special method called fencing which is used to avoid quorum corruption. Quorum corruption can be caused by node recovering from a shut down, when it will propagate its state the other machines.

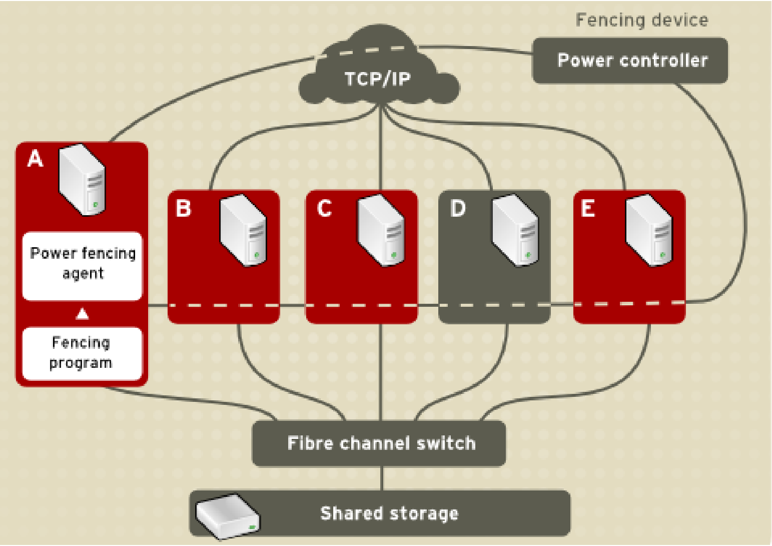


Figure 5. RedHat example of fencing enabled on a cluster (RedHat)

The principle of fencing is in isolating a crashed device from a quorum device to not let it corrupt the quorum storage. Fencing technique depends on the software used, but usually it sends a special command or script to nodes and those unable to respond will be shut down. Sometimes smart power switches are used to implement fencing.

3.4 Prevention of brain splitting

Brain splitting is an issue already mentioned adbove, it happens when two or more groups of nodes lose connection to each other, and after the connection is restored, there are two separate cluster states which will cause a conflict.

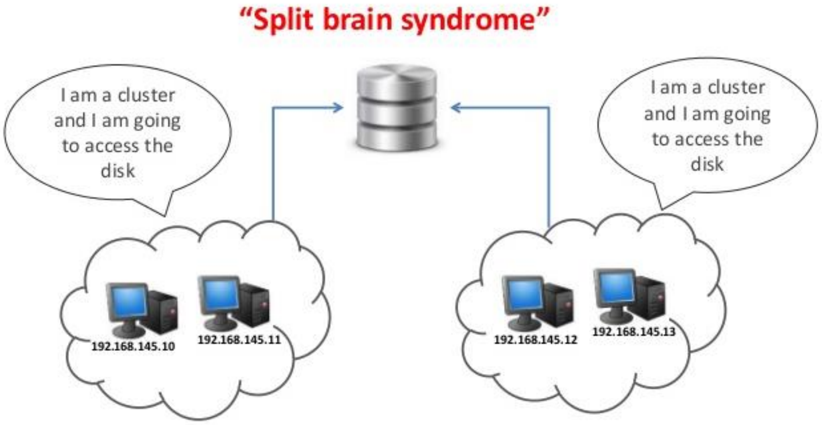

"Split brain syndrome"
I am a cluster and I am going to access the disk
I am a cluster and I am going to access the disk
192.168.145.10
192.168.145.11
192.168.145.12
192.168.145.13

Figure 6. Very basic example of brain splitting (no author)

For determining the "dominating" part the vote system will be used. The part with the most votes will write its state to the quorum device and other parts will need to use this state. In case of a vote tie, there should be a special tie-braker algorithm which will determine the master cluster part.

4 SOFTWARE FOR IMPLEMENTATION

In this part I will cover the differences between Windows and Linux clustering, and clustering software. The most common abbreviations in this part will be Command Line Interface (CLI) and Graphical User Interface (GUI). There could also be abbreviations of some tools, but during the first time they are mentioned there will be a full name.

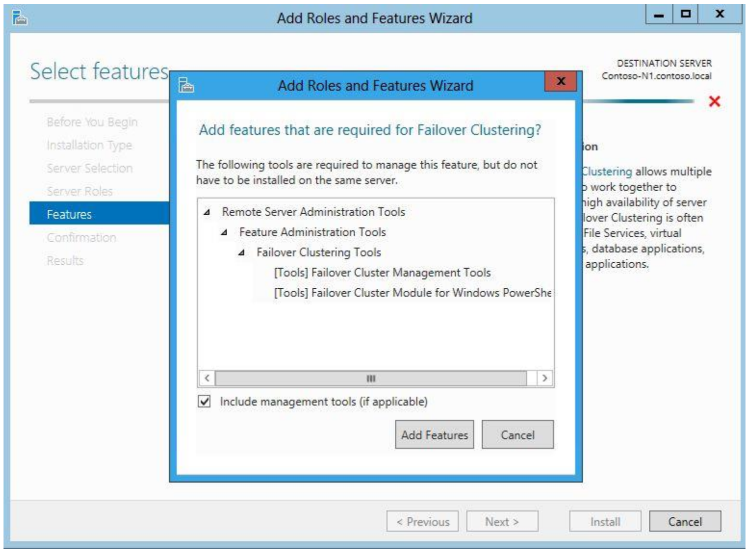
4.1 Windows in server clustering

Windows is a proprietary operating system and its main feature is very high dependability on Microsoft services, and this causes limitations of options to choose from, although the given ones are quite good.

The default stack (set of software) for typical Windows cluster is Windows Server itself which runs on top of Hyper-V. The typical management software is Microsoft Failover Cluster Manager and the typical cloud service is Microsoft Azure. In the perfect (for Microsoft) case there will also be Internet Information Service (IIS) as a web server and Microsoft SQL server for hosting a database. The two most popular Windows stacks are called WISA and WINS. Another important aspect is that Microsoft solutions rely more on GUI tools whereas Linux is more CLI-oriented.

4.1.1 Management tools

Clustering is available only in the server edition of Windows. There are several GUI tools available. The main one is a special tool mentioned above which is called Microsoft Failover Cluster Manager. It’s enabled as a role in the Tools section in Server Manager. It is a pre-built management tool in Windows Server.



Add features that are required for Failover Clustering?

The following tools are required to manage this feature, but do not have to be installed on the same server.

- Remote Server Administration Tools
  - Feature Administration Tools
    - Failover Clustering Tools
      - [Tools] Failover Cluster Management Tools
      - [Tools] Failover Cluster Module for Windows PowerShe

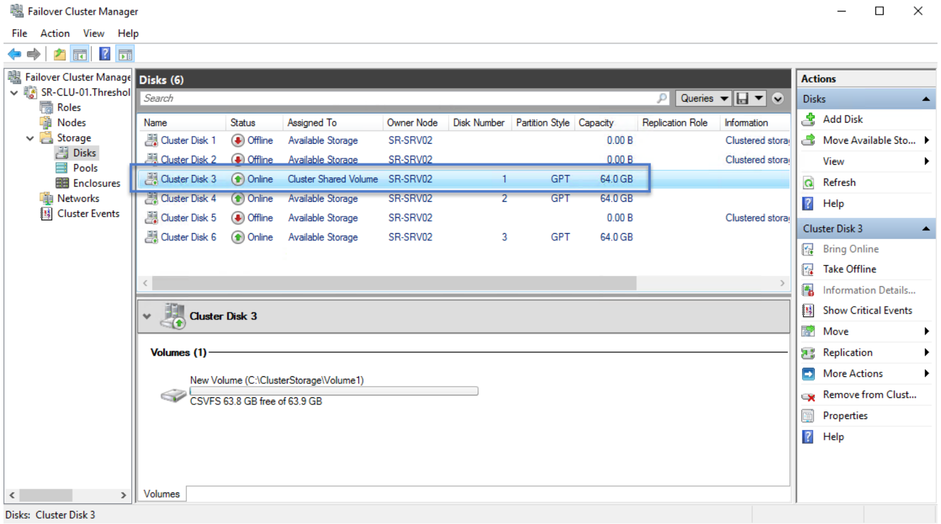
☑ Include management tools (if applicable)

ion

Clustering allows multiple
work together to
igh availability of server
lover Clustering is often
File Services, virtual
, database applications,
applications.

Figure 7. The process of installing Failover Cluster Manager tool (Microsoft, 2012)

The tool itself is straight-forward and offers a clean interface which helps to manage the cluster without difficulties. There is the possibility to configure a cluster via a command line (PowerShell) as well. As always, GUI is generally more intuitive, but the command line causes less overhead.



Name	Status	Assigned To	Owner Node	Disk Number	Partition Style	Capacity	Replication Role	Information
Cluster Disk 1	Offline	Available Storage	SR-SRV02			0.00 B		Clustered stora
Cluster Disk 2	Offline	Available Storage	SR-SRV02			0.00 B		Clustered stora
Cluster Disk 3	Online	Cluster Shared Volume	SR-SRV02	1	GPT	64.0 GB		
Cluster Disk 4	Online	Available Storage	SR-SRV02	2	GPT	64.0 GB		
Cluster Disk 5	Offline	Available Storage	SR-SRV02			0.00 B		Clustered stora
Cluster Disk 6	Online	Available Storage	SR-SRV02	3	GPT	64.0 GB		

New Volume (C:\ClusterStorage\Volume1)
CSVFS 63.8 GB free of 63.9 GB

Disks: Cluster Disk 3

Figure 8. Dashboard of MFCM (Microsoft)

A good benefit of MFCM is in the supporting tests. For example, there is a feature called planned failover. It simulates errors on nodes, and thus network engineers can learn what will happen in case of a real threat.

As for command line configuration, there is also a possibility to manage a cluster from PowerShell. First of all, it should be launched with administrator privileges. Secondly, cluster management tools should be installed using the command *import-module failoverclusters*.

There are a lot of commands (cmdlets) available which are useful and can easily be a proper replacement for MFCM if network administrators prefers CLI solutions to GUI ones. For the list of cmdlets, visit Microsoft PowerShell guide, 2008.

4.1.2 Windows quorum types

The first aspect I want to mention is the types of quorum in Windows. There are several of them and their main difference is the organization of the vote system. The figures below will show the most extreme (closest to the shutdown) example of a cluster which will still operate with given configurations.

a) Node Majority

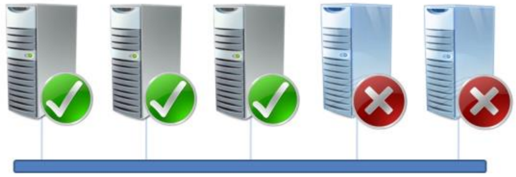


Figure 9. Node majority cluster structure (Penton, ITPro)

In this type of cluster there is no disk witness. Therefore, all the data is duplicated only to nodes. That also means that nodes get the votes, one per each node. A cluster considers itself working and the changes commited if it’s possible to apply the changes to half of the all nodes plus one (n/2+1) nodes. This type of quorum is good for organizing clusters with an odd number of nodes (including a single-node cluster) and when there is no possibility to have a special quorum storage. This type of quorum is easy to configure but not really reliable.

b) Node and Disk Majority

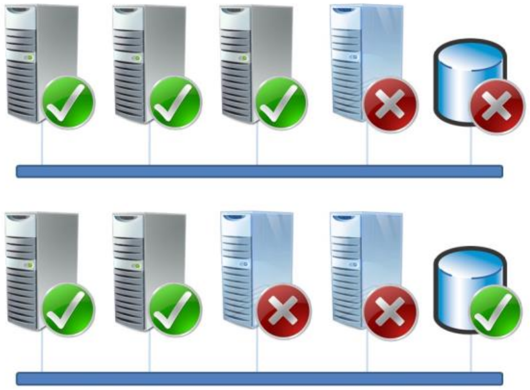


Figure 10. Node and disk majority cluster structure (Penton, ITPro)

In this type of quorum each node and a disk witness get the vote. There must be a majority of votes available, i.e. at least half of the nodes should be operating, if the disk witness is working and one more than a half (n/2+1), if the disk witness is shut down. This type of quorum is good for clusters with an even number of nodes.

c) Node and File Share Majority

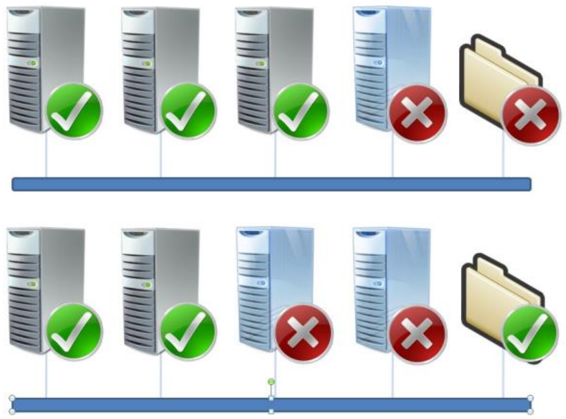


Figure 11. Node and file share majority cluster structure (Penton, ITPro)

It’s completely the same type of quorum as the previous one, but the disk witness is replaced with a file share (file storage) witness. Everything stated above for node and disk majority is also valid for this type.

d) No Majority / Disk Only

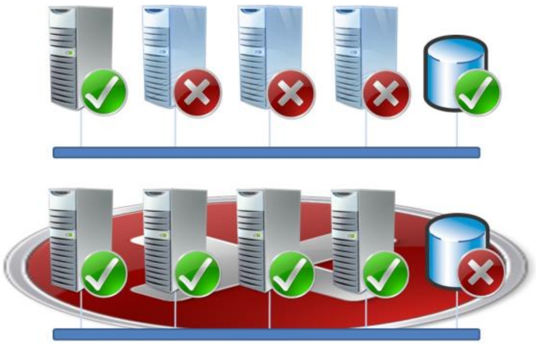


Figure 12. No majority / Disk only cluster structure (Penton, ITPro)

This type of quorum always operates, if a data witness is available and never operates, if the data witness is shut down. The status of nodes doesn’t matter. This is the worst possible solution for quorum in Windows, because a single error can stop the whole cluster from operating. It’s unreliable and unrecommended.

e) Dynamic quorum

This type of quorum was introduced in Windows Server 2012. It can dynamically assign votes to nodes and adjust quorum value to prevent a cluster from shutting down. This type of a cluster can work with one node up, this situation is called last-man standing. This is a default quorum type for Windows Server 2012 and later versions.

f) Cloud witness

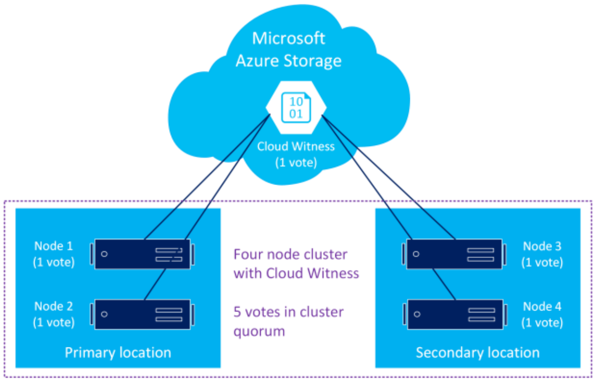


Figure 13. Cloud witness cluster(s) structure (no author, 2016)

The newest method of quorum organization was introduced only in the latest edition of Windows Server 2016. In this type of quorum nodes are connected to an Azure cloud storage via HTTPS. This storage also gets the vote and works the same way as a node and disk / node and file share majority, although the same Azure storage can be attached to several clusters. Azure will store different IDs for each cluster and distinguish them via this. This is an expensive but secure and reliable solution.

g) **Conclusion**

To summarize all the quorum types, table lists the best possible uses for each of them.

Table 1. Brief comparison of quorum types available in Windows Server.

Quorum Type	Odd-numbered clusters	Even-numbered clusters	Multi-site clusters	Good without shared storage
NM	+		+	+
NaDM		+		
NaFSM		+	+	+
NM/DO				
DQ	+	+		+
CW		+	+	+

Node majority is the best option for an odd number of nodes, including single-node solutions, and quite good for multi-site clusters

Node and disk majority is the best option for an even number of nodes (including two-node solutions).

Node and file share majority is the same as for NaDM, but is also good for multi-site clusters.

No majority / Disk only must be avoided at all costs as long as a single error ruins the whole cluster.

Dynamic quorum is the best solution in newer editions of Windows Server, because it helps to sustain a lot of simultaneous errors.

Cloud witness is without a doubt the best option for multi-site clusters, the same storage can support several clusters at a time, very good option when there is no physical storage available.

4.1.3 Virtualization and clusters in Windows

The very good benefit of Windows Server is built-in support for virtualization. It is available via the use of Hyper-V. Virtualization is one of the most important techniques in IT. It includes all the methods of creating a virtual, abstract version of devices, plaftorms and resources. The main benefits of virtualization are higher utilization of servers through their consolidation, reduced costs, and better control over the system which is good for network administrators.

Clusters benefit from all of these main advantages of virtualization, and this means that it should be used in all cases, when it is needed and the company can afford it. Other solutions can be used as well, but I will cover them in chapter four, section three, because they are not developed by Microsoft.

4.1.4 Microsoft Azure in clustering

Cloud services, apps and solutions are extremely important, even core, aspects of nowadays' technology. Cloud computing refers to using a network or the Internet for storing and locating services, files and applications instead of running them locally. It covers all aspects of IT, from simple mobile apps to complex enterprise networks.

Microsoft Azure is a proprietary cloud service which has a lot of very useful appliances and roles. I will cover only the ones somehow related to clustering (though almost all of them are very important in clusters). The first two roles are very crucial and the next ones are supplementary but still in high demand.

Azure can serve as a virtual machine. This is an example of Infrastructure as a Service (IaaS). Azure allows VMs (both Windows and Linux, which is extremely good and allows flexibility) to be installed in a cloud within a very short time. It also supports templating (creating several VMs according to pre-given template), and that makes Azure a very convenient solution for network engineers. As long as clusters could and should be virtualized in order to reduce the installation costs, this use of Azure is in high demand.

It can also serve as a storage. This is the second most important use of Microsoft Azure. As covered before, cloud storage is the core aspect of the Cloud Witness quorum type. Using Azure in this role will significantly ease the management of multi-site clustering systems and will provide extra flexibility.

Azure can serve as a SQL database for one or several clusters. Cluster nodes almost always need databases to fetch data from. This use of Azure is not directly related to the installation of servers, but can still help to reduce costs because there will be no need to configure a local data server within the network.

Azure can also serve as a location for different types of apps and application programming interfaces (APIs). It allows the cluster to have permanent and quick access to the functions needed. There is a plenty of useful tools which can serve from Azure. For example, Microsoft itself provides Office 365 API for better service management.

Azure can serve as cloud manager for Active Directory. In theory it can be used to manage policies, security and other additional settings. This role is really useful because, as was already mentioned, the same Azure cloud can be connected to several clusters.

Summing up, Microsoft Azure is a very useful service whose main disadvantage is its exlusiveness to Windows Server (which is a disadvantage in general, but not for Windows-oriented networks), though Azure provides a lot of Linux related apps and OSs which almost neglect the disadvantage.

The main reasons to use Azure are the following:

- Cost: Azure is neither free nor very cheap but it’s still less expensive than creating the new infrastructure from scratch. This is especially important for small companies which cannot afford big data centers with a lot of physical devices.
- Flexibility: As mentioned above, Azure provides both Windows and Linux oriented services, APIs, apps and OSs, which gives possibility to choose

from several options (which is always better because different companies require different methods and technologies).

- Solid background: Azure is not only a very solid service, but, what is more important, it is developing fast. Microsoft is interested in having all their main services represented as Azure API or app and it means that network engineers and developers have a lot of technologies to choose from.
- Ease of access and recovery: Microsoft has a lot of servers and Azure nodes throughout the globe, and that gives high-speed access to the Azure cloud, and such strong IT company as Microsoft can ensure the stability of this cloud service.

4.1.5 Conclusion for Windows

The two main disadvantages of Windows as a clustering operating system are its cost (it’s impossible to create a Windows cluster for free if company doesn’t own Windows Server license yet) and limited flexibility. The first one is crucial, because not all companies can afford to buy a stack of Microsoft apps, although, as it was already mentioned in the disadvantages of clustering in general, this drawback doesn’t relate to my thesis. Limited flexibility can be expanded via the use of external cross-platform solutions.

The advantages of Windows in clustering are good and long-lasting customer support from Microsoft, and a human-friendlier and more intuitive interface in comparison to Linux and Microsoft Azure as a platform. Clearly distributed quorum types are also good.

4.2 Linux in server clustering

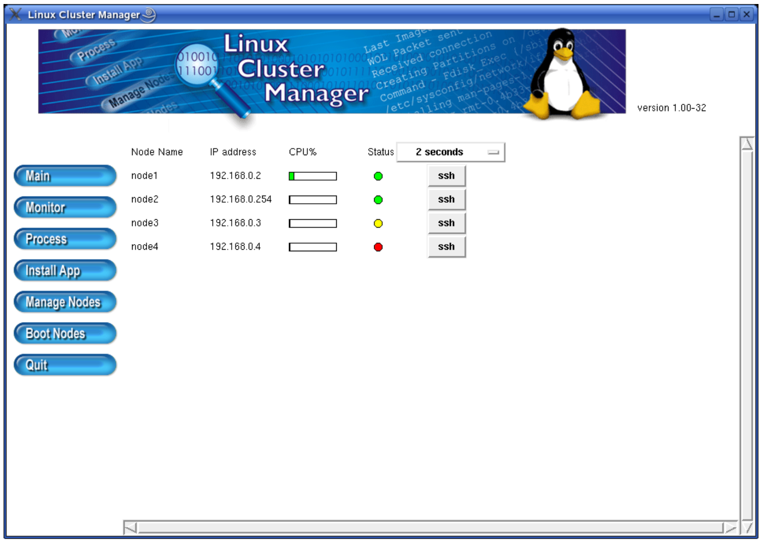
Linux is well-known for its open-sourceness. This leads to the possibility to choose between very big number of different software, because there is no operating system owner which will force its solutions to be used.

This can also result in several drawbacks. First of all, Linux options are not as obvious as Windows ones, and thus network administrators will need to research possible solutions and then create or find the most suitable stack. The second disadvantage is the absence of customer support. It’s not very crucial, because Linux has great documentation, but having a possibility for support is never a bad option. The third drawback is that Linux is harder to configure properly. It requires more qualified people to work with, because all the aspects of configuring and managing are less obvious in Linux.

4.2.1 Management tools

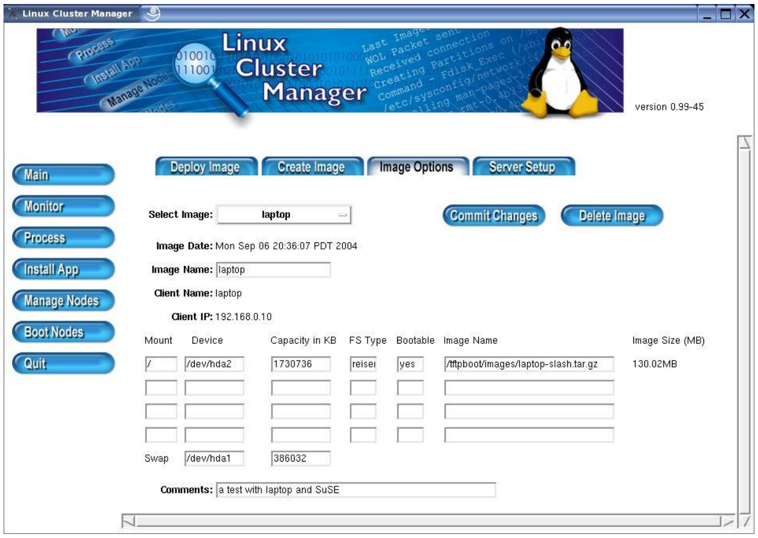
There are plenty of different tools for networks with different needs. I will mention the most popular ones which will suit the most popular needs. All the tools in this part of the study are considered open-source, and thus free to download and use, until stated otherwise.

There are several GUI management tools for Linux but they are less common than CLI solutions. One of the most popular one is Linux Cluster Manager (LCM).



Node Name	IP address	CPU%	Status	2 seconds
node1	192.168.0.2		🟢	ssh
node2	192.168.0.254		🟢	ssh
node3	192.168.0.3		🟡	ssh
node4	192.168.0.4		🔴	ssh

Figure 14. LCM Dashboard (LCM)



Select Image: laptop

Image Date: Mon Sep 06 20:36:07 PDT 2004

Image Name: laptop

Client Name: laptop

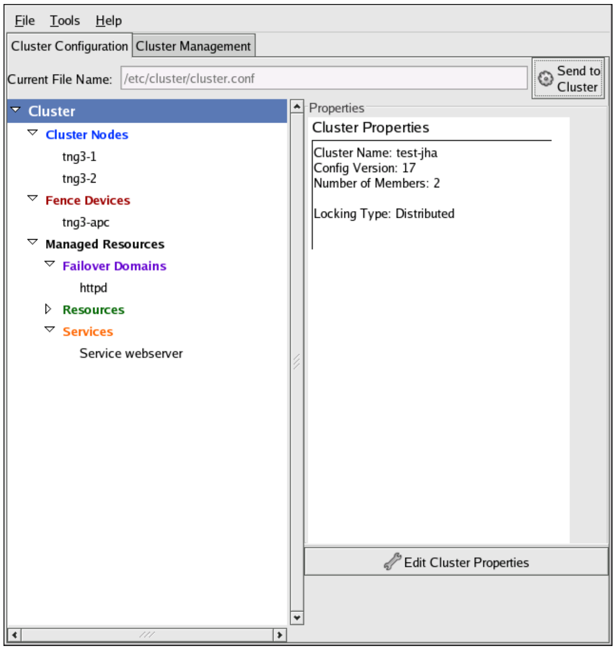
Client IP: 192.168.0.10

Mount	Device	Capacity in KB	FS Type	Bootable	Image Name	Image Size (MB)
/	/dev/hda2	1730736	reiser	yes	/tftpboot/images/laptop-slash.tar.gz	130.02MB
Swap	/dev/hda1	386032				

Comments: a test with laptop and SuSE

Figure 15. LCM Image manager (LCM)

The principle of this tool is the same as that Microsoft Failover Cluster Manager. It also has a dashboard and different views to manage and configure different aspects of the cluster. Examples can be seen in figures 14 and 15 above. The interface may look more outdated in comparison to MFCM, but it is valid for the majority of open-source tools.



Current File Name: /etc/cluster/cluster.conf

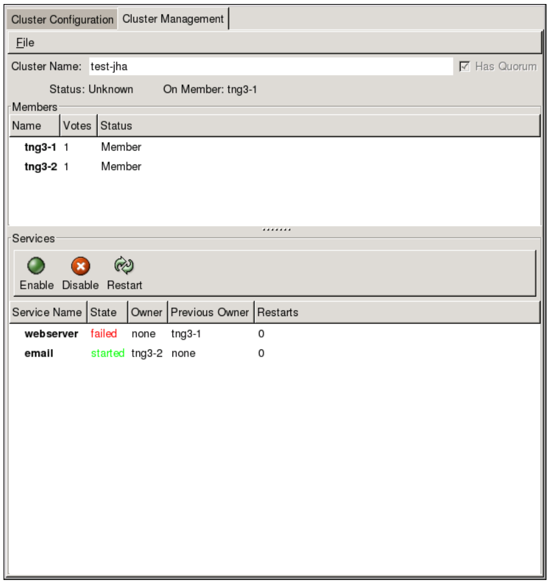
Cluster Properties

Cluster Name: test-jha
Config Version: 17
Number of Members: 2

Locking Type: Distributed

Figure 16. Cluster Administration GUI (Cluster Configuration Tool window) from Red Hat (Red Hat, Inc)

The second option is a special GUI tool which is a part of Red Hat Cluster Suite. It doesn't have an official name and is called Cluster Administration GUI. Unlike the previous option, this one is not free because it comes with suite which costs a lot of money. Red Hat has a policy close to Microsoft – it sells its distributions of Linux, but at the same time offers high-quality customer support. The majority of other distributions doesn't have a company behind them. Thus, customer support is mostly limited to official documentation.



Cluster Name: test-jha ☑ Has Quorum

Status: Unknown On Member: tng3-1

Members

Name	Votes	Status
tng3-1	1	Member
tng3-2	1	Member

Services

Service Name	State	Owner	Previous Owner	Restarts
webserver	failed	none	tng3-1	0
email	started	tng3-2	none	0

Figure 17. Cluster Status Tool. (Red Hat, Inc)

This software is divided into two main tools – Cluster Configuration Tool and Cluster Status Tool. The first one is used for creating, editing and propagating cluster files, whereas the second one is used for management. The use of two given tools is quite apparent. Cluster Administration GUI is a part of paid suite and should be used, if the clusters run software from Red Hat. If not, there is much more sense in using Linux Cluster Manager or other free tools.

There are more GUI tools but I wanted to point these out as examples of one free and open commercial software. The example of other commercial solutions (but not embedded into the distributive) can be CFEngine (Northern.tech, Inc).

There is an extremely high number of CLI management tools for Linux. I will not cover all of them, but will mention the most important ones for each aspect of clustering management. As in the GUI section, all the software is free until stated otherwise.

Two necessary requirements for a clustering management tool (or stack of tools) is the possibility to configure (implement commands) and manage (gather the status) clusters.

Popular tools for configuring clusters via a command line are:

- GNU Parallel (Free Software Foundation, Inc.). It provides the possibility to execute commands parallelly (to several machines at the same time). This is extremely useful in clustering because all the clusters must always have the same state.
- Fabric (Jeff Forsier, 2017). It's an extremely useful tool for executing commands both remotely and locally, uploading and downloading files and for some additional functionality. It requires Python to run because the commands in Fabric are written in Python as well.
- Munge (Chris Dunlap, 2017). This is a tool for implementing authentication on cluster nodes if it's needed.

Popular tools for managing clusters via a command line are:

- RDDtool (Tobias Oetiker, 2017). It logs the servers' actions and state and draws the graph if needed. Its good feature is the possibility to embed the tool into scripts and apps written in different languages like Perl, Python, Ruby etc.
- FreeIPMI (FreeIPMI Core Team, 2017). It is a tool used for collecting data about temperature of hardware, electrical data (e.g. voltage, power supply data), basic errors.

To sum up, there are plenty of tools for different tasks. Almost all tools have analogs based on another programming language or designed for another distribution. That leads to the main advantage of Linux over Windows – it is much more flexible and offers much more freedom.

4.2.2 Quorum in Linux

Unlike in Windows, there are no pre-configured quorum types in Linux. Linux clusters should also have fencing a device which will isolate a stopped node from other ones, and a quorum device which will store the state of the nodes, although it is not obligatory to have it.

The new term which is connected to quorum is introduced in Linux and called fencing wars. It is quite similar to brain-splitting, but the action causing the problem is different. If there are two nodes in a cluster and they lose connection to each other, they will both try to fence (isolate) another node and restart all the services in a cluster, because both nodes will suppose that the other one is broken. The quorum device will solve this issue by propagating the last saved state to both nodes after they are recovered. Two-node clusters are the most prone to fencing wars, so in Linux there is more need to use a quorum device for them than for others, although the quorum device is always a good solution for clusters.

4.2.3 Cluster virtualization in Linux

Some distributions, e.g. Red Hat and CentOS, can create virtual clusters through using built-in tools. A good guide about setting up a virtual cluster in Red Hat Linux is present on TechTarget (Stuart Burns, 2016). Unfortunately, as long as Linux doesn’t have one company behind it, there are no universal solutions. There are a lot of good external options for virtualization, and I will cover them in chapter four, sections three and four.

4.2.4 Conclusion for Linux

The main disadvantages of Linux are generally reversed advantages of Windows. It’s less user-friendly and intuitive, and has problems with customer support which is mostly limited to six months of support and good official documentation. The advantages of Linux are, again, reversed disadvantages of Windows: much lower costs and more flexibility in software to choose from.

All the advantages and disadvantages of Linux are mostly not valid for Red Hat distributions (Red Hat Enterprise Linux and partly Fedora with CentOS) because this company utilizes the same business model as Microsoft, for example they sell the product but offer proprietary solutions limited to their system and also offer long support. This means that it’s possible to choose Linux, but to organize cluster technologies in the Microsoft way.

4.3 VMware in server clustering

VMware is a company which is currently a market leader in virtualization software. It also provides possibilities for virtualized clustering which can run on top of any existing OS (using a web application) or using native Windows-based software. I will introduce VMware more briefly than Linux and Windows, despite it being the current market leader in clustering, because I will not be able to use VMware. It is a paid option and unftortunately I have no credits for virtualizing it and thus for implementing virtualization in virtual environment.

4.3.1 VMware stack

VMware solution requires several applications to be installed. I will call them “VMware stack”.

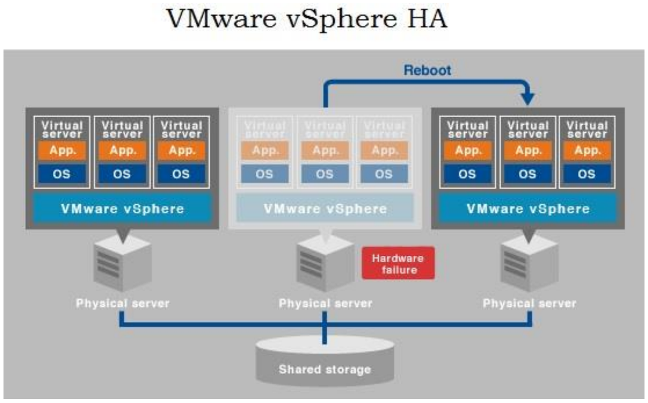


Figure 18. Structure of the VMware cluster (VMware HA) with an example of a hardware issue and reboot process (vStackL, 2016)

The core one is VMware ESXi. ESXi is a bare-metal hypervisor, i.e. it runs on top of the hardware and provides OS elements needed such as kernel, services etc. It also runs hosts. Platform called vSphere works on top of ESXi. First of all, vSphere has its own server called vCenter Server. vSphere clients connect to vCenter Server (or to hosts directly) and it’s possible to manage virtual servers using them.

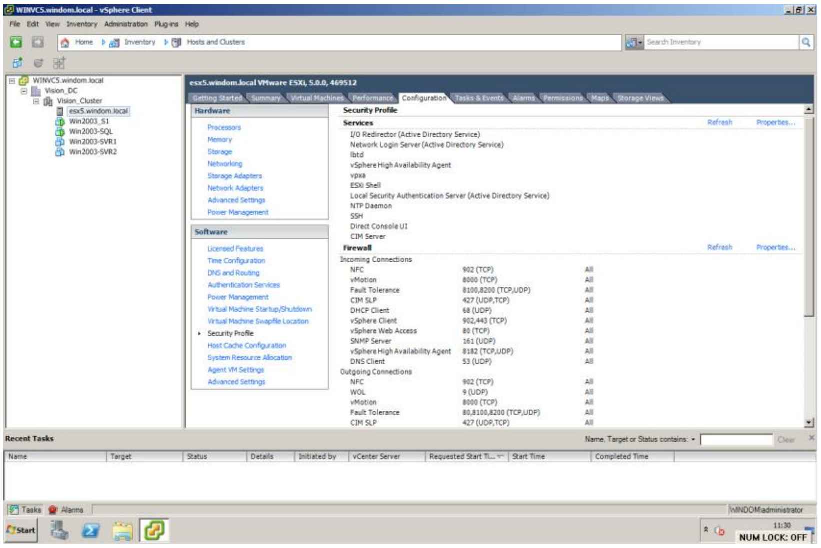
4.3.2 Cluster types

There are two types of VMware clusters – DRS (distributed resource scheduler) and HA (high availability).

- DRS cluster has a pool which consists of resources of hosts and is expanded when a new host is added. In this type of cluster load balancing (even load distribution), power management and virtual machines placement can be enabled and used.
- HA cluster gathers all hosts into itself and manages them. Once some host has experienced failures, the cluster transfers all its virtual machines to other hosts in case there is enough storage space available.

4.3.3 Cluster management

Management process is done via one of the two methods. The first one is the tool mentioned before (vSphere Client). It runs on top of the host and gives at opportunity to create virtual servers (nodes) and also manage them. The client is available only on Windows.



```
Security Profile
Services
  I/O Redirector (Active Directory Service)
  Network Login Server (Active Directory Service)
  lbtd
  vSphere High Availability Agent
  vpxa
  ESXi Shell
  Local Security Authentication Server (Active Directory Service)
  NTP Daemon
  SSH
  Direct Console UI
  CIM Server
Firewall
Incoming Connections
  NFC                              902 (TCP)                  All
  vMotion                          8000 (TCP)                 All
  Fault Tolerance                  8100,8200 (TCP,UDP)        All
  CIM SLP                          427 (UDP,TCP)              All
  DHCP Client                      68 (UDP)                   All
  vSphere Client                   902,443 (TCP)              All
  vSphere Web Access               80 (TCP)                   All
  SNMP Server                      161 (UDP)                  All
  vSphere High Availability Agent  8182 (TCP,UDP)             All
  DNS Client                       53 (UDP)                   All
Outgoing Connections
  NFC                              902 (TCP)                  All
  WOL                              9 (UDP)                    All
  vMotion                          8000 (TCP)                 All
  Fault Tolerance                  80,8100,8200 (TCP,UDP)     All
  CIM SLP                          427 (UDP,TCP)              All
```

Figure 19. vSphere Client window running in the old version of Windows Server (VirtuallyLG, 2012)

The interface of the applications resembles Microsoft Failover Cluster Manager a lot, and this is valid for user experience. Options and views are intuitive and well-structurized, the client is human-friendly. Another option is to use vSphere Web Client. As it follows from the title, this version of the client runs from a browser. It is cross platform, but has limited functionality in comparison to the native client, and cannot connect directly to hosts. Only connection to vCenter Server is valid.

Table 2 summarizes both options.

Table 2. Comparison of vSphere Client and vSphere Web Client

	vSphere Client	vSphere Web Client
Location	needs to be insalled	runs from browser
Platform	only Windows	cross-platform
Valid connections	to hosts and vCenter Server	only to vCenter Server
Functionality	all functionality present	possible to desploy VMs and manage clusters' state. Impossible to configure anything
Extendability	absent	supports plug-ins

The desktop application is suitable for network administrator and engineers, in other words, for people who will configure the network. For people working with status management, like help desk, operators and managers Web Client will be more appropriate.

4.3.4 vCloud Air

vCloud Air is a cloud computing service very similar to Microsoft Azure, it’s also Infrastructure as a Service (IaaS) product. About half a year ago, in the second quarter of 2017, vCloud was sold to a French cloud provider called OVH, but its services were well integrated into the new owner’s existing ones and are basically intact. OVH provides server solutions (virtual private server, dedicated server etc.), cloud solutions (private cloud, public cloud) and other types of services (CDN, email, Software as a Service solutions).

4.3.5 Conclusion for VMware

VMware solutions are a very solid choice for most types of companies. Its emphasis on virtualization makes it cheaper than Windows, although it will still cost a lot, but at the same time it’s still more user-friendly than Linux. VMware is a very good compromise solution.

The main disadvantage is the necessity to use VMware, and OVH in case of cloud solutions, software, in other words, VMware is not a flexible solution when it comes to the management and hypervisor software.

4.4 Other solutions

There are a lot of different clustering options available, but I only cover Docker Swarm briefly. Docker is a containers technology. Container is an abstract virtualization of an operating system. Therefore it can be considered as a virtual server.

Docker Swarm is a special mode which provides a possibility to form a group (basically a cluster) of Docker Engines (containers, i.e. virtual servers). Swarm has a lot of benefits, e.g. it supports basic clustering features (load balancing, scalability, easy management etc.). It’s secure due to Transport Layer Security mutual authentication, it’s flexible, for example it’s possible to declare different functionality to different parts of the clusters, and a lot of companies (Red Hat, Microsoft, IBM) openly support Docker. Therefore, it is updated periodically and is a very promising technology. The last big advantage of Docker is its cross-platform availability. It’s available for Windows, Linux, MacOS and even FreeBSD.

A lot of other solutions by different companies utilize Docker containers technology (Mesos by Apache, Kubernetes by Google and others). All of these technologies are listed by Anand Akela, 2016.

4.5 Choosing the proper OSs

Windows released a new Server edition last year (Windows Server 2016). This is the main reason why I decided to choose it. I’m very interested in Docker containers technology as well, and due to its virtualizing nature and the possibility to run on any operating system I would have a chance to implement a Docker cluster, if I had time. Windows Server will allow me to do this. Windows will be ran in a cloud environment.

4.6 Software for the project

As long as I plan to use Windows, the obvious choice for virtualization is Hyper-V. I have no plans for using VMware solutions but, as mentioned above, I will try to work with Docker, if I had time.

As for non-virtualization software, I plan to use software mentioned in chapter four, section two.

4.7 Cloud provider

There are a lot of cloud service providers, but the three main ones are Microsoft Azure, Amazon Web Service and DigitalOcean. All of them are commercial options, but with a possibility for free trial and credits for students. I have used DigitalOcean before and had never any problems with it. I also have credits there, but for this project this option is not considered by me because it doesn’t offer Windows. Microsoft Azure is Windows-centered, but Xamk is not able to provide credits for me, thus I will use Amazon Web Service (AWS).

5 PRACTICAL PART

In this part of the thesis covers the installation, configuration and testing processes. These include CMD commands, all the necessary configurations done with GUI and other important steps. Very basic theory can also be present in this part, but mostly in form of explanations.

HPC cluster is high performance cluster and nowadays this term means the same as just term cluster because all the clusters are designed for continuous service work.

All the figures appearing in this chapter are captured by me unless stated otherwise, and thus have no official source.

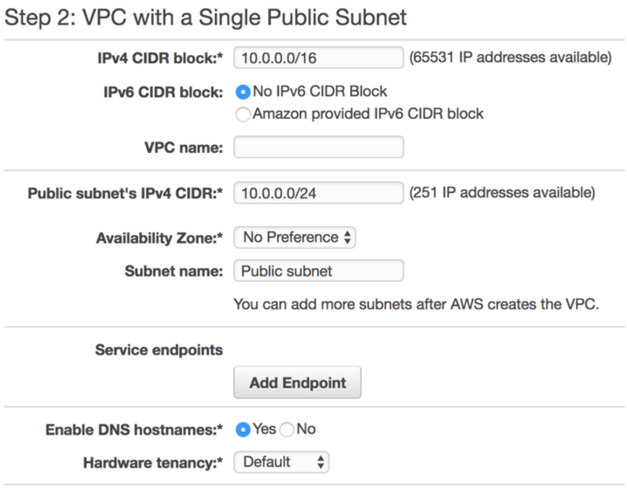
5.1 Setting up Virtual Private Cloud and testing

The first important step of installing and configuring a cluster is to set up virtual private cloud (VPC). In order to this, I need to run VPC wizard. Creation of instance and security group is used to check if VPC is able of hosting a server or accepting security settings.

Amazon has extremely useful tutorials. For the installation and configuration purposes, I will mostly use them as guidelines (Amazon Web Services, Inc, 2017).

5.1.1 Creating a network

The first window is about creating a private network, subnet inside this network and using gateway. It also automatically creates routing table for possible future needs. This step is very straightforward.



Step 2: VPC with a Single Public Subnet

IPv4 CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

IPv6 CIDR block: ● No IPv6 CIDR Block
○ Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR:* 10.0.0.0/24 (251 IP addresses available)

Availability Zone:* No Preference

Subnet name: Public subnet

You can add more subnets after AWS creates the VPC.

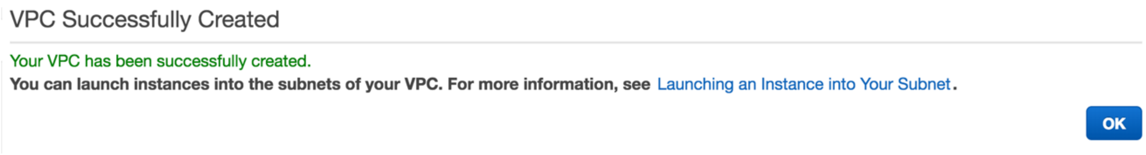
Service endpoints

Enable DNS hostnames:* ● Yes ○ No

Hardware tenancy:* Default

Figure 20. Window of network configuration in VPC wizard

The only knowledge needed for setting VPC up is understanding of subnet masks and what the default gateway is. Wizard provides values by default, thus almost nobody can face issues while completing this step. I haven’t faced them as well and successfully created a VPC.



VPC Successfully Created

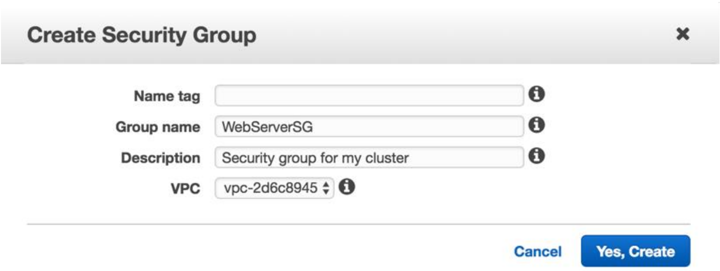
Your VPC has been successfully created.
You can launch instances into the subnets of your VPC. For more information, see Launching an Instance into Your Subnet**.**

Figure 21. Confirmation of VPC being created

My VPC is created in Ohio (Eastern USA) zone. Due to Ohio being located half a globe away from Finland, this can cause some issues related to connection speed, but they should not be crucial enough to affect the project.

5.1.2 Configure security groups

The next important step is configuring security groups. This is very similar to creating access lists on Cisco devices. The purpose of this step is to establish security on VPC and therefore protect possible instances from attacks and thefts.



Create Security Group

Name tag:

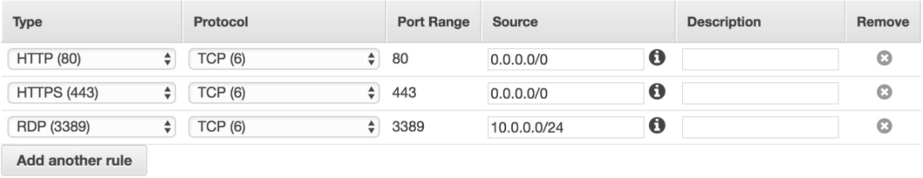
Group name: WebServerSG

Description: Security group for my cluster

VPC: vpc-2d6c8945

Figure 22. Creation of the security group in AWS console

As well as in Cisco devices' access control lists, there are inbound and outbund rules which regulate and filter the traffic going to, within and from the network. Summing up, this step can be described as configuring ACLs using web GUI.

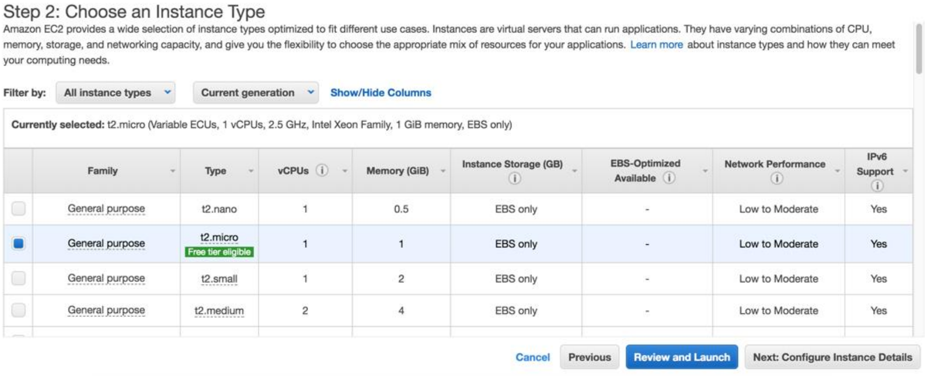


Type	Protocol	Port Range	Source	Description	Remove
HTTP (80)	TCP (6)	80	0.0.0.0/0		⊗
HTTPS (443)	TCP (6)	443	0.0.0.0/0		⊗
RDP (3389)	TCP (6)	3389	10.0.0.0/24		⊗

Figure 23. Configuration of inbound rules for my VPC

5.1.3 Creating an instance

The next step is to launch an instance in the VPC. Purpose of it is mainly in the testing of possibility of VPC to host instances, but it helps to orient in AWS configuration windows as well. There are a lot of sub-steps but they are all straightforward. A lot of options are provided by default using customer's VPC data.



Step 2: Choose an Instance Type

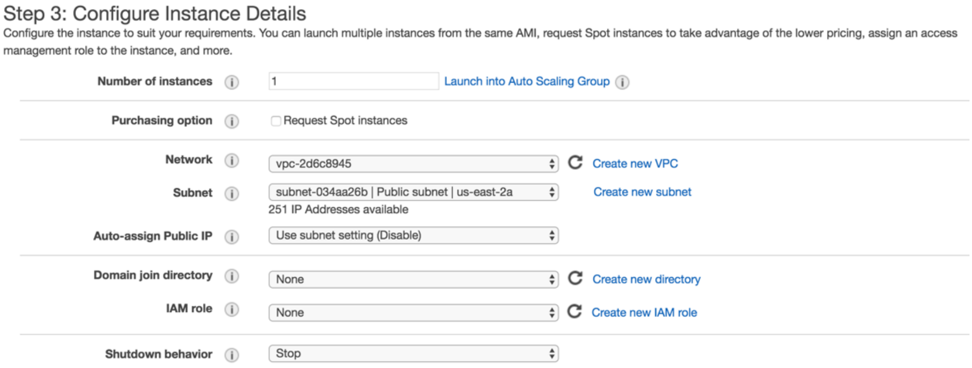
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

Figure 24. Selection of instance's hardware

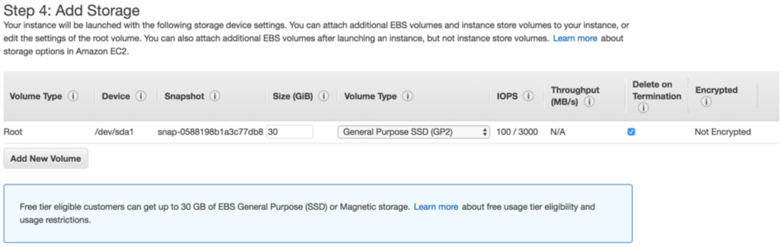


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	□ Request Spot instances
Network	vpc-2d6c8945
Subnet	subnet-034aa26b | Public subnet | us-east-2a 251 IP Addresses available
Auto-assign Public IP	Use subnet setting (Disable)
Domain join directory	None
IAM role	None
Shutdown behavior	Stop

Figure 25. Selection of assosiated network and subnet



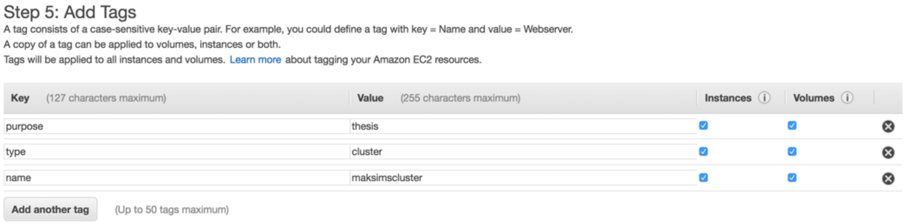
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0588198b1a3c77db8	30	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Figure 26. Selection and configuration of storage option

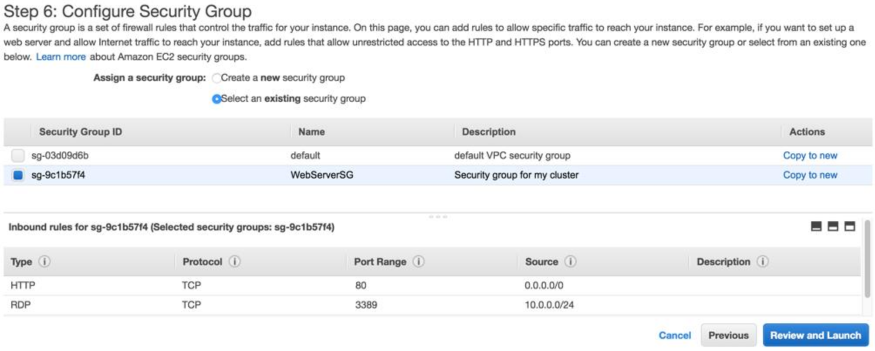


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes	
purpose	thesis	☑	☑	⊗
type	cluster	☑	☑	⊗
name	maksimscluster	☑	☑	⊗

Figure 27. Selection of tag pairs



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

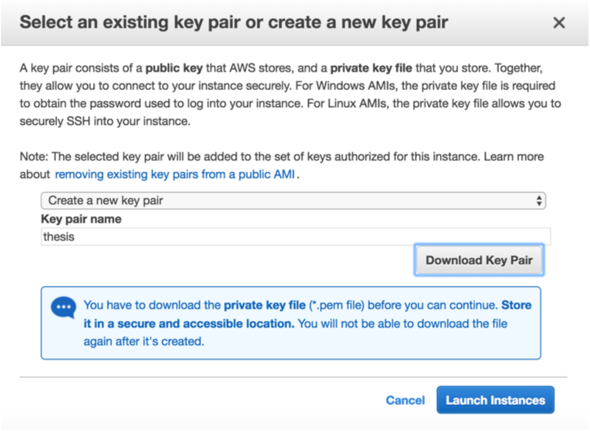
Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security Group ID	Name	Description	Actions
□	sg-03d09d6b	default	default VPC security group	Copy to new
■	sg-9c1b57f4	WebServerSG	Security group for my cluster	Copy to new

Inbound rules for sg-9c1b57f4 (Selected security groups: sg-9c1b57f4)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
RDP	TCP	3389	10.0.0.0/24	

Figure 28. Selection of security groups



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

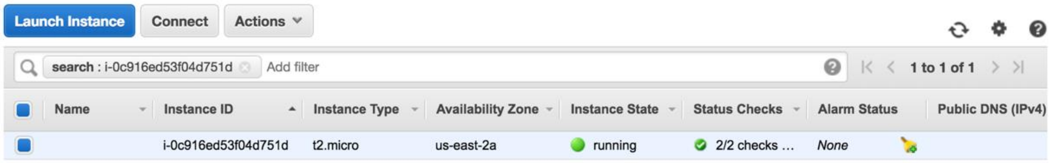
Key pair name

thesis

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Figure 29. Configuration of key pair

As a result of these steps, my VPC now hosts a Windows Server 2016 instance. It’s possible to connec to it using Remote desktop file which can be downloaded from AWS console, though I will not do it right now.

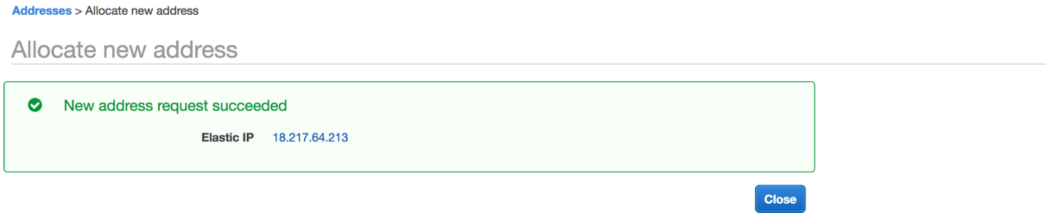


■	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
■		i-0c916ed53f04d751d	t2.micro	us-east-2a	● running	✓ 2/2 checks ...	*None*	

Figure 30. Status screen of my instance

5.1.4 Configuring elastic IP address

The next step is to configure elastic IP address. In AWS non-default VPC (i.e. the VPC created by customer) doesn’t have any public IP addresses, therefore it needs one to communicate with the Internet.

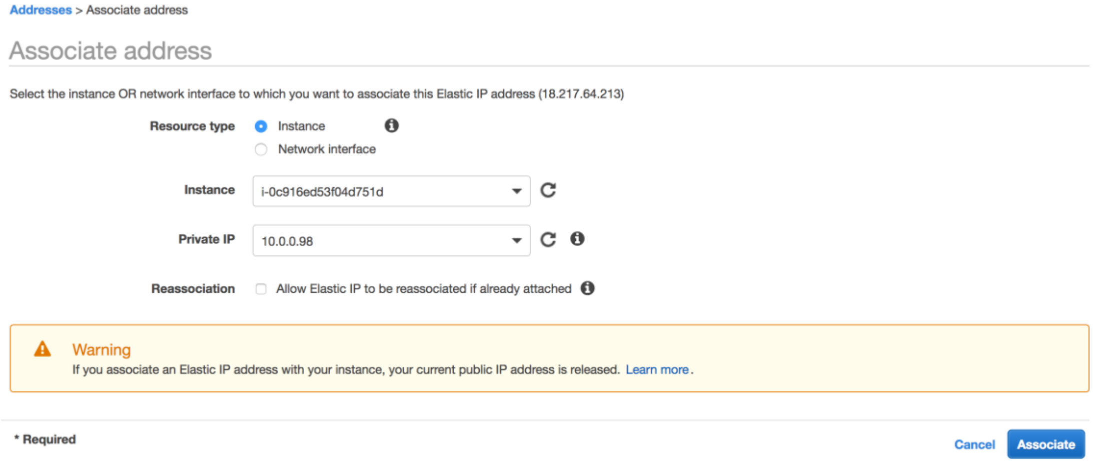


Allocate new address

New address request succeeded

Elastic IP 18.217.64.213

Figure 31. Confirmation of new public IP address being allocated



Associate address

Select the instance OR network interface to which you want to associate this Elastic IP address (18.217.64.213)

Resource type ● Instance
○ Network interface

Instance i-0c916ed53f04d751d

Private IP 10.0.0.98

Reassociation □ Allow Elastic IP to be reassociated if already attached

Warning
If you associate an Elastic IP address with your instance, your current public IP address is released. Learn more.

* Required

Figure 32. Process of association the instance with IP

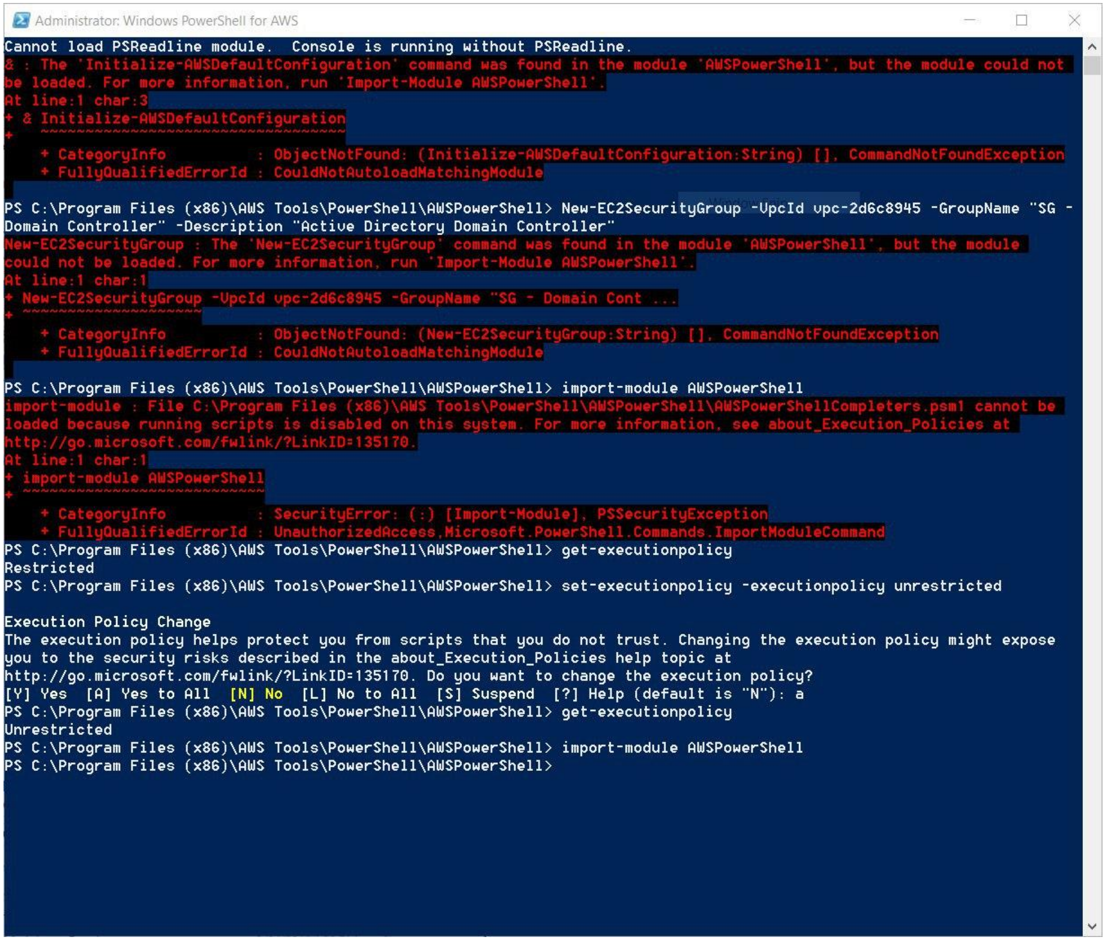
This step was straightfoward as well. To receive an elastic IP, there is only a need to press button ”Get IP”, and as for association, there is only a need to select customer’s instance’s data from dropdown menus.

5.2 Setting up cluster

This section is about creating and configuring a domain controller and nodes. This process is also straightforward and consists of web GUI forms, but I will cover them anyway.

5.2.1 Creating security group for domain members

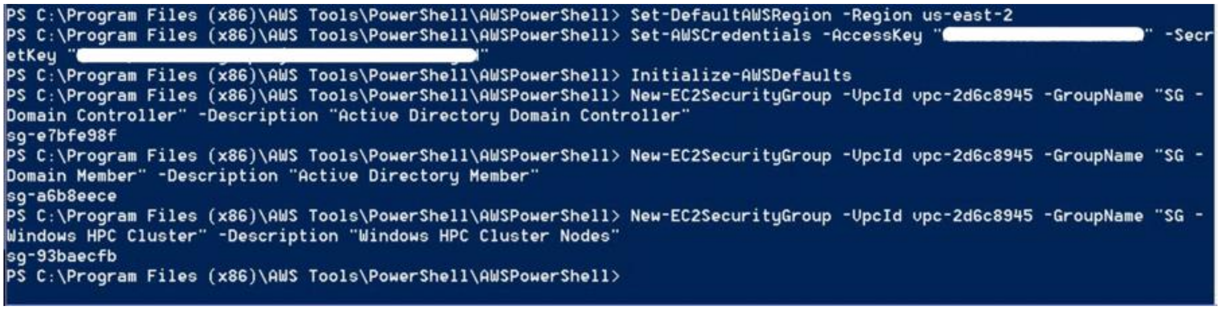
The first step is a prerequisite to actual creation of the domain controller. Its goal is to create several security groups and assign rules to them. This process is done via Windows PowerShell, and this caused problems for me because I am a Mac user. In my opinion, this is one of the worst aspects of AWS because people using different OSs should be able to configure Windows Server instances, and not only Windows users. Thus, I was forced to install Windows as secondary operating system to my MacBook. In order to connect to Amazon servers, I used a special tool called AWS tools for Windows PowerShell. It is available at AWS website (Amazon Web Services, Inc.).



```
Cannot load PSReadline module.  Console is running without PSReadline.
& : The 'Initialize-AWSDefaultConfiguration' command was found in the module 'AWSPowerShell', but the module could not
be loaded. For more information, run 'Import-Module AWSPowerShell'.
At line:1 char:3
+ & Initialize-AWSDefaultConfiguration
+   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (Initialize-AWSDefaultConfiguration:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CouldNotAutoloadMatchingModule

PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> New-EC2SecurityGroup -VpcId vpc-2d6c8945 -GroupName "SG -
Domain Controller" -Description "Active Directory Domain Controller"
New-EC2SecurityGroup : The 'New-EC2SecurityGroup' command was found in the module 'AWSPowerShell', but the module
could not be loaded. For more information, run 'Import-Module AWSPowerShell'.
At line:1 char:1
+ New-EC2SecurityGroup -VpcId vpc-2d6c8945 -GroupName "SG - Domain Cont ...
+ ~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (New-EC2SecurityGroup:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CouldNotAutoloadMatchingModule

PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> import-module AWSPowerShell
import-module : File C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell\AWSPowerShellCompleters.psm1 cannot be
loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at
http://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ import-module AWSPowerShell
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [Import-Module], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess,Microsoft.PowerShell.Commands.ImportModuleCommand
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> get-executionpolicy
Restricted
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> set-executionpolicy -executionpolicy unrestricted

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): a
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> get-executionpolicy
Unrestricted
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> import-module AWSPowerShell
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
```

Figure 33. PowerShell with errors and process of policy changing

By default, PowerShell resuses to run the scripts due to its security policy being set to restricted mode. As mentioned in guide present on Microsoft website (Juan Pablo Jofre et al.), this mode means that PowerShell is allowed to run single commands, but prohibited to run scripts from any sources. In order to change the mode to unrestricted, I ran *set-ExecutionPolicy –ExecutionPolicy unrestricted*. This is insecure, and thus not recommended in real working environment, but as long as I change the security policy on my local machine, it is acceptable. The next thing was to import Amazon Web Services Power Shell module and it was done via *import-module AWSPowerShell*.



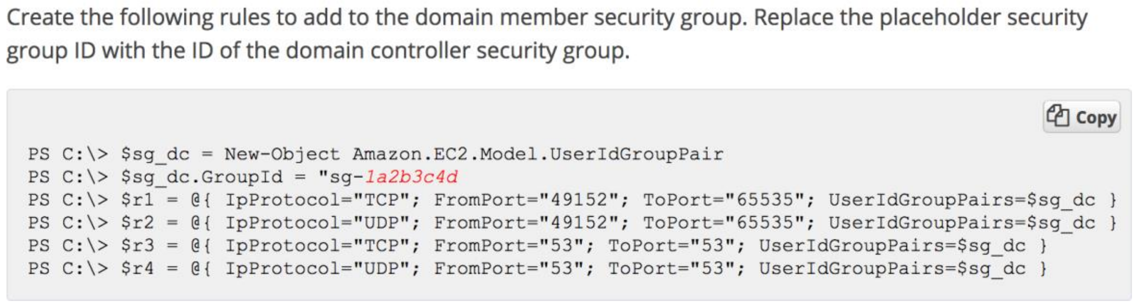
```
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Set-DefaultAWSRegion -Region us-east-2
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Set-AWSCredentials -AccessKey "[redacted]" -Secr
etKey "[redacted]"
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Initialize-AWSDefaults
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> New-EC2SecurityGroup -VpcId vpc-2d6c8945 -GroupName "SG -
Domain Controller" -Description "Active Directory Domain Controller"
sg-e7bfe98f
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> New-EC2SecurityGroup -VpcId vpc-2d6c8945 -GroupName "SG -
Domain Member" -Description "Active Directory Member"
sg-a6b8eece
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> New-EC2SecurityGroup -VpcId vpc-2d6c8945 -GroupName "SG -
Windows HPC Cluster" -Description "Windows HPC Cluster Nodes"
sg-93baecfb
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
```

Figure 34. Process of configuring Amazon credentials and creating security groups

In order to apply settings to my VPC, I needed to provide my data to AWS tools. Since my VPC is located in Ohio, USA, the command *Set-DefaultAWSRegion –Region us-east-2* was used to make Ohio server zone a default one for me.

Then I used command *Set-AWSCredentials –AccessKey "xxx" –SecretKey "xxx"*. Where both *xxx* stand for Amazon Access Key and Amazon Secret Access Key. They are both provided by Amazon secretly and are used in same way as login and password, the second one is even not accessible once it was assigned to a user and shown once. Due to thesis being publicly available and Theseus, I hid these keys. The next command was *Initialize-AWSDefauls* and it initializes (activates) two previous commands.

The next step was to create three security groups. This was implemented via issuing the same command, *New-EC2SecurityGroup –VpcId vpcid_of_users_vpc –GroupName "desired security group name" –Description "desired description for security group"* three times with slightly different attributes. All the outputs shown in Figure 34 are security group IDs used in next steps.

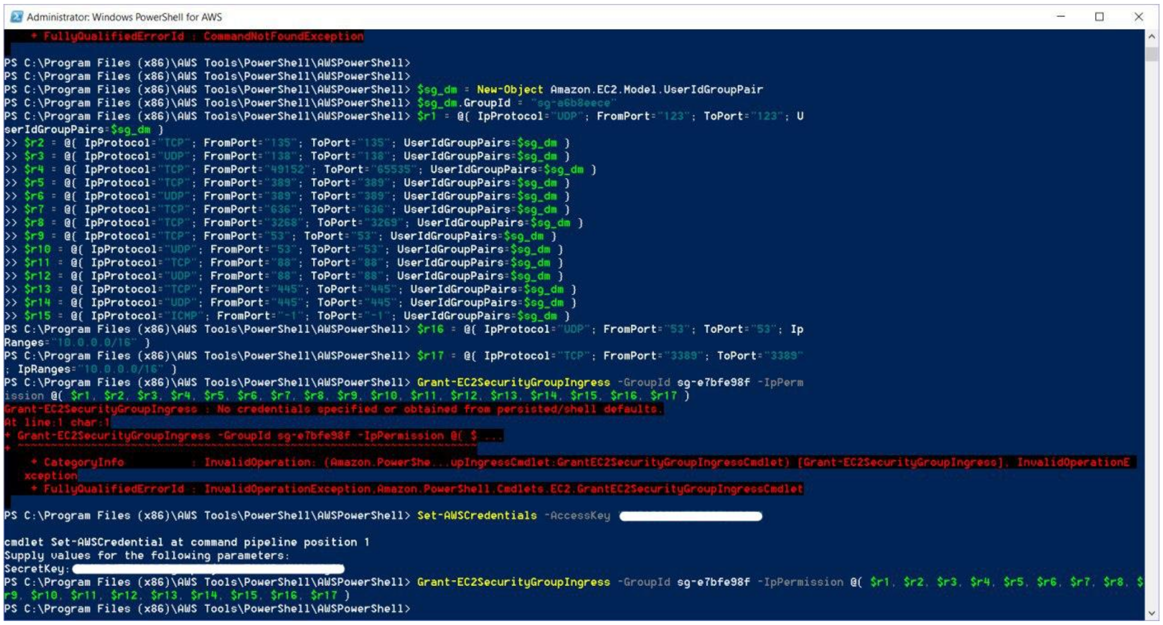


Create the following rules to add to the domain member security group. Replace the placeholder security group ID with the ID of the domain controller security group.

```
PS C:\> $sg_dc = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\> $sg_dc.GroupId = "sg-1a2b3c4d
PS C:\> $r1 = @{ IpProtocol="TCP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dc }
PS C:\> $r2 = @{ IpProtocol="UDP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dc }
PS C:\> $r3 = @{ IpProtocol="TCP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dc }
PS C:\> $r4 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dc }
```

Figure 35. Sreenshot from AWS tutorial containing an error

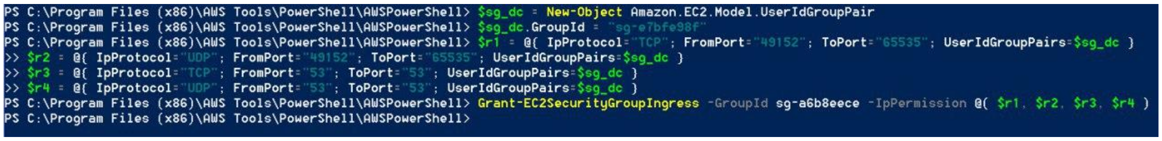
The next step was a little bit confusing for me because AWS tutorial contained a crucial error. As can be seen in Figure 35, second line in commands snippet has an opening quotes but doesn’t have a closing one. This error repeats in all three code snippets for different security groups. I tried provided commands and found out the error empirically. Luckily, incorrect settings didn’t ruin the existing settings because these commands executed incorrectly just don’t work.



Administrator: Windows PowerShell for AWS

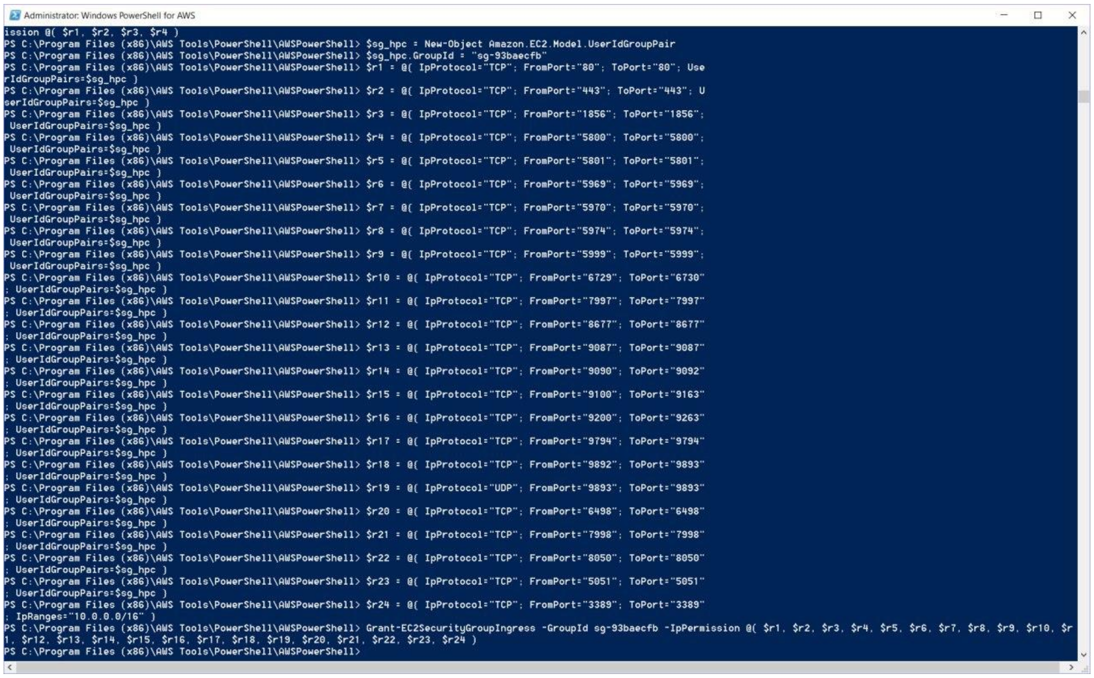
```
    + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $sg_dm = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $sg_dm.GroupId = "sg-a6b8eece"
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r1 = @{ IpProtocol="UDP"; FromPort="123"; ToPort="123"; U
serIdGroupPairs=$sg_dm }
>> $r2 = @{ IpProtocol="TCP"; FromPort="135"; ToPort="135"; UserIdGroupPairs=$sg_dm }
>> $r3 = @{ IpProtocol="UDP"; FromPort="138"; ToPort="138"; UserIdGroupPairs=$sg_dm }
>> $r4 = @{ IpProtocol="TCP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dm }
>> $r5 = @{ IpProtocol="TCP"; FromPort="389"; ToPort="389"; UserIdGroupPairs=$sg_dm }
>> $r6 = @{ IpProtocol="UDP"; FromPort="389"; ToPort="389"; UserIdGroupPairs=$sg_dm }
>> $r7 = @{ IpProtocol="TCP"; FromPort="636"; ToPort="636"; UserIdGroupPairs=$sg_dm }
>> $r8 = @{ IpProtocol="TCP"; FromPort="3268"; ToPort="3269"; UserIdGroupPairs=$sg_dm }
>> $r9 = @{ IpProtocol="TCP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dm }
>> $r10 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dm }
>> $r11 = @{ IpProtocol="TCP"; FromPort="88"; ToPort="88"; UserIdGroupPairs=$sg_dm }
>> $r12 = @{ IpProtocol="UDP"; FromPort="88"; ToPort="88"; UserIdGroupPairs=$sg_dm }
>> $r13 = @{ IpProtocol="TCP"; FromPort="445"; ToPort="445"; UserIdGroupPairs=$sg_dm }
>> $r14 = @{ IpProtocol="UDP"; FromPort="445"; ToPort="445"; UserIdGroupPairs=$sg_dm }
>> $r15 = @{ IpProtocol="ICMP"; FromPort="-1"; ToPort="-1"; UserIdGroupPairs=$sg_dm }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r16 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; Ip
Ranges="10.0.0.0/16" }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r17 = @{ IpProtocol="TCP"; FromPort="3389"; ToPort="3389"
; IpRanges="10.0.0.0/16" }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Grant-EC2SecurityGroupIngress -GroupId sg-e7bfe98f -IpPerm
ission @( $r1, $r2, $r3, $r4, $r5, $r6, $r7, $r8, $r9, $r10, $r11, $r12, $r13, $r14, $r15, $r16, $r17 )
Grant-EC2SecurityGroupIngress : No credentials specified or obtained from persisted/shell defaults.
At line:1 char:1
+ Grant-EC2SecurityGroupIngress -GroupId sg-e7bfe98f -IpPermission @( $ ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (Amazon.PowerShe...upIngressCmdlet:GrantEC2SecurityGroupIngressCmdlet) [Grant-EC2SecurityGroupIngress], InvalidOperationE
   xception
    + FullyQualifiedErrorId : InvalidOperationException,Amazon.PowerShell.Cmdlets.EC2.GrantEC2SecurityGroupIngressCmdlet

PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Set-AWSCredentials -AccessKey [redacted]

cmdlet Set-AWSCredential at command pipeline position 1
Supply values for the following parameters:
SecretKey: [redacted]
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Grant-EC2SecurityGroupIngress -GroupId sg-e7bfe98f -IpPermission @( $r1, $r2, $r3, $r4, $r5, $r6, $r7, $r8, $
r9, $r10, $r11, $r12, $r13, $r14, $r15, $r16, $r17 )
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
```

Figure 36. Configurations for domain members security group



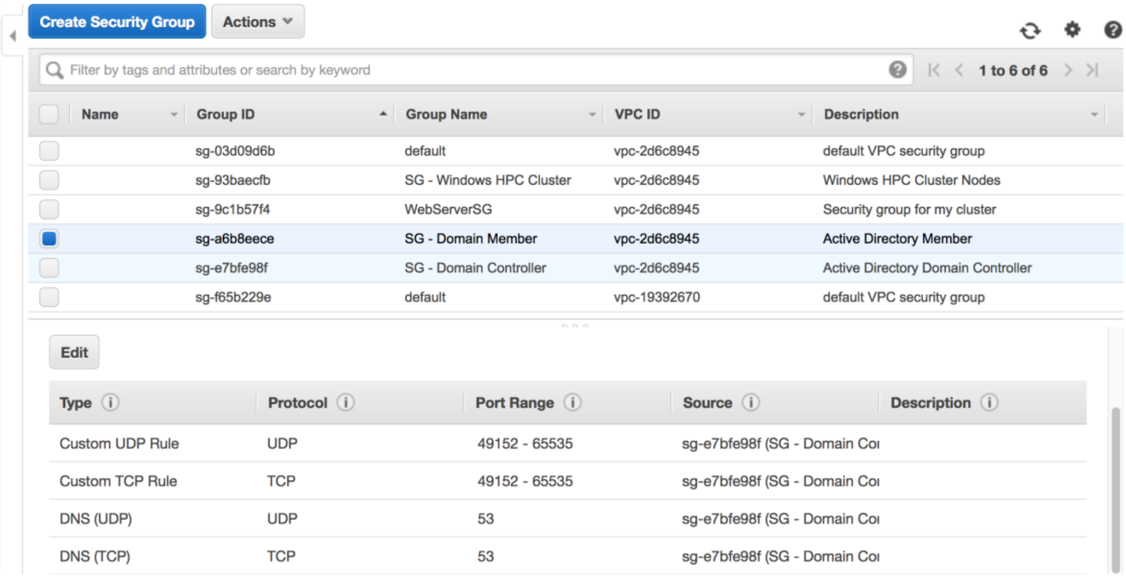
```
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $sg_dc = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $sg_dc.GroupId = "sg-e7bfe98f"
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r1 = @{ IpProtocol="TCP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dc }
>> $r2 = @{ IpProtocol="UDP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dc }
>> $r3 = @{ IpProtocol="TCP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dc }
>> $r4 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Grant-EC2SecurityGroupIngress -GroupId sg-a6b8eece -IpPermission @( $r1, $r2, $r3, $r4 )
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
```

Figure 37. Configurations for domain controller security group



```
ission @( $r1, $r2, $r3, $r4 )
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $sg_hpc = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $sg_hpc.GroupId = "sg-93baecfb"
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r1 = @{ IpProtocol="TCP"; FromPort="80"; ToPort="80"; Use
rIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r2 = @{ IpProtocol="TCP"; FromPort="443"; ToPort="443"; U
serIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r3 = @{ IpProtocol="TCP"; FromPort="1856"; ToPort="1856";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r4 = @{ IpProtocol="TCP"; FromPort="5800"; ToPort="5800";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r5 = @{ IpProtocol="TCP"; FromPort="5801"; ToPort="5801";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r6 = @{ IpProtocol="TCP"; FromPort="5969"; ToPort="5969";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r7 = @{ IpProtocol="TCP"; FromPort="5970"; ToPort="5970";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r8 = @{ IpProtocol="TCP"; FromPort="5974"; ToPort="5974";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r9 = @{ IpProtocol="TCP"; FromPort="5999"; ToPort="5999";
 UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r10 = @{ IpProtocol="TCP"; FromPort="6729"; ToPort="6730"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r11 = @{ IpProtocol="TCP"; FromPort="7997"; ToPort="7997"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r12 = @{ IpProtocol="TCP"; FromPort="8677"; ToPort="8677"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r13 = @{ IpProtocol="TCP"; FromPort="9087"; ToPort="9087"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r14 = @{ IpProtocol="TCP"; FromPort="9090"; ToPort="9092"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r15 = @{ IpProtocol="TCP"; FromPort="9100"; ToPort="9163"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r16 = @{ IpProtocol="TCP"; FromPort="9200"; ToPort="9263"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r17 = @{ IpProtocol="TCP"; FromPort="9794"; ToPort="9794"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r18 = @{ IpProtocol="TCP"; FromPort="9892"; ToPort="9893"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r19 = @{ IpProtocol="UDP"; FromPort="9893"; ToPort="9893"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r20 = @{ IpProtocol="TCP"; FromPort="6498"; ToPort="6498"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r21 = @{ IpProtocol="TCP"; FromPort="7998"; ToPort="7998"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r22 = @{ IpProtocol="TCP"; FromPort="8050"; ToPort="8050"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r23 = @{ IpProtocol="TCP"; FromPort="5051"; ToPort="5051"
; UserIdGroupPairs=$sg_hpc }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> $r24 = @{ IpProtocol="TCP"; FromPort="3389"; ToPort="3389"
; IpRanges="10.0.0.0/16" }
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Grant-EC2SecurityGroupIngress -GroupId sg-93baecfb -IpPermission @( $r1, $r2, $r3, $r4, $r5, $r6, $r7, $r8, $r9, $r10, $r
1, $r12, $r13, $r14, $r15, $r16, $r17, $r18, $r19, $r20, $r21, $r22, $r23, $r24 )
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell>
```

Figure 38. Configurations for HPC nodes security group



	Name	Group ID	Group Name	VPC ID	Description
□		sg-03d09d6b	default	vpc-2d6c8945	default VPC security group
□		sg-93baecfb	SG - Windows HPC Cluster	vpc-2d6c8945	Windows HPC Cluster Nodes
□		sg-9c1b57f4	WebServerSG	vpc-2d6c8945	Security group for my cluster
■		sg-a6b8eece	SG - Domain Member	vpc-2d6c8945	Active Directory Member
□		sg-e7bfe98f	SG - Domain Controller	vpc-2d6c8945	Active Directory Domain Controller
□		sg-f65b229e	default	vpc-19392670	default VPC security group

Type	Protocol	Port Range	Source	Description
Custom UDP Rule	UDP	49152 - 65535	sg-e7bfe98f (SG - Domain Co	
Custom TCP Rule	TCP	49152 - 65535	sg-e7bfe98f (SG - Domain Co	
DNS (UDP)	UDP	53	sg-e7bfe98f (SG - Domain Co	
DNS (TCP)	TCP	53	sg-e7bfe98f (SG - Domain Co	

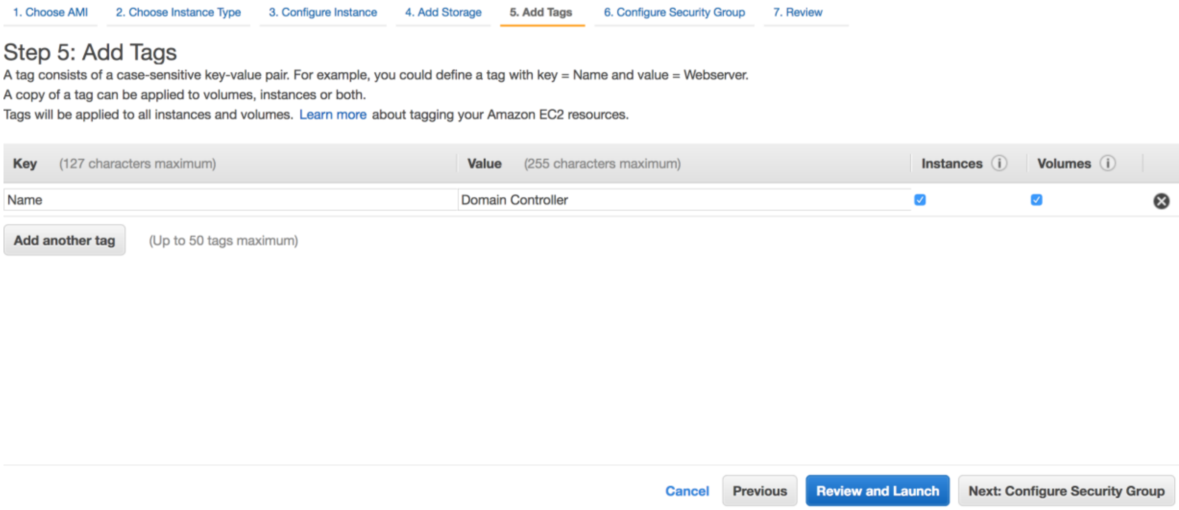
Figure 39. Verification of the result in web GUI

As can be seen in figures 36, 37, 38 and 39, needed commands are applied and the result is accepted by VPC. The figure 36 has one incorrect command but this command was overridden with the same command using correct attributes on the next line.

The process could have also been done using web GUI, but, first of all, tutorial provides only PowerShell commands, and I decided not to spent time converting commands to manual GUI settings one rule by one, and, secondly, I was interested in having experience in both web GUI and CLI configuring, although the need in PowerShell forced me to use another operating system and that I faced the errors in tutorial.

5.2.2 Installing prerequisites for a domain controller

The beginning is completely the same as the one mentioned in chapter five, section one, step three, I also create a “general purpose” Windows Server 2016 Amazon Machine Image (AMI) and then assign 30 GB to it as a storage. The following steps differ, and thus I will mention them more thoroughly.

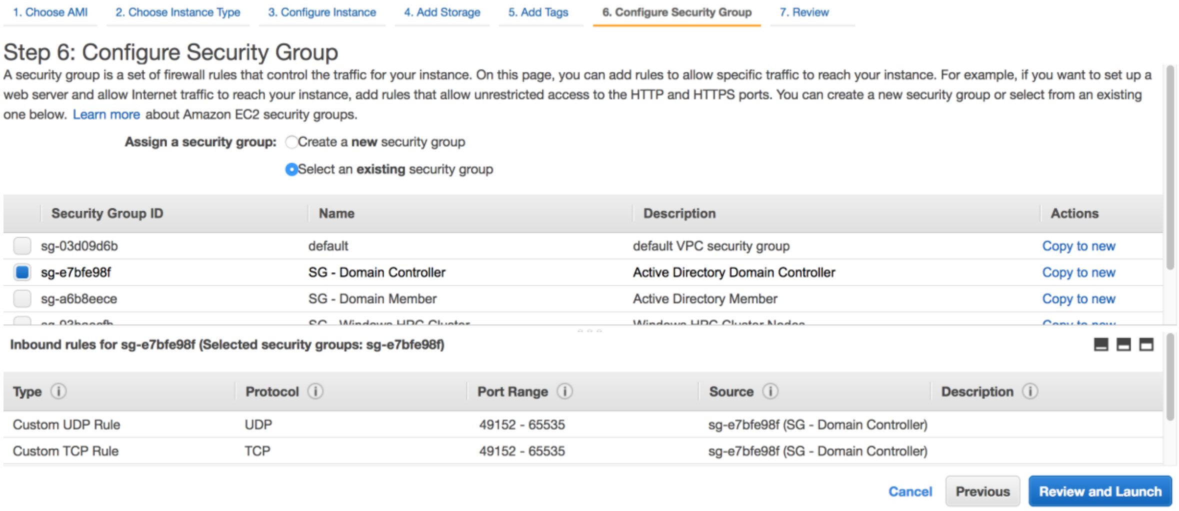


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	Domain Controller	☑	☑

Figure 40. Process of tag assigning



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

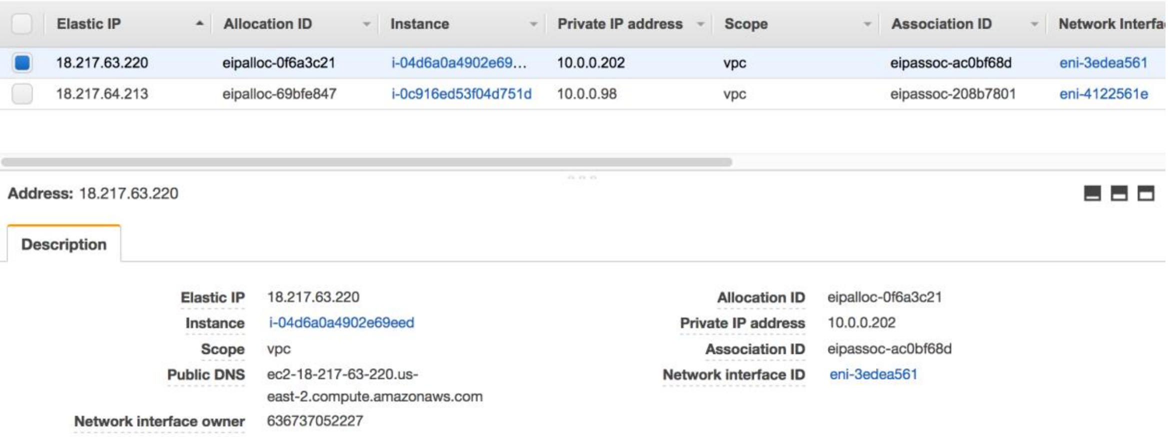
	Security Group ID	Name	Description	Actions
□	sg-03d09d6b	default	default VPC security group	Copy to new
☑	sg-e7bfe98f	SG - Domain Controller	Active Directory Domain Controller	Copy to new
□	sg-a6b8eece	SG - Domain Member	Active Directory Member	Copy to new
□	sg-03baacfb	SG - Windows HPC Cluster	Windows HPC Cluster Nodes	Copy to new

Inbound rules for sg-e7bfe98f (Selected security groups: sg-e7bfe98f)

Type	Protocol	Port Range	Source	Description
Custom UDP Rule	UDP	49152 - 65535	sg-e7bfe98f (SG - Domain Controller)	
Custom TCP Rule	TCP	49152 - 65535	sg-e7bfe98f (SG - Domain Controller)	

Figure 41. Process of assigning domain controller security group to a new instance

First of all, I needed to create a pair of tags. As can be seen in Figure 40, the key is “name” and the value is “domain controller”. The next step was to assign a domain controller security group, which was created by me in chapter five, section two, step one, to a new instance. This can be seen in Figure 41. Both steps were easy and straightforward.



□	Elastic IP	Allocation ID	Instance	Private IP address	Scope	Association ID	Network Interfa
■	18.217.63.220	eipalloc-0f6a3c21	i-04d6a0a4902e69...	10.0.0.202	vpc	eipassoc-ac0bf68d	eni-3edea561
□	18.217.64.213	eipalloc-69bfe847	i-0c916ed53f04d751d	10.0.0.98	vpc	eipassoc-208b7801	eni-4122561e

Address: 18.217.63.220

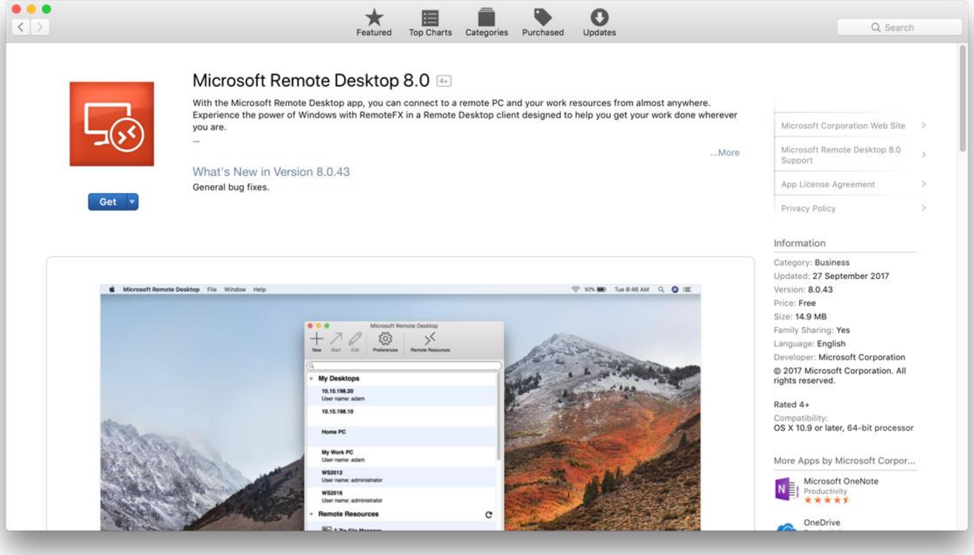
Elastic IP	18.217.63.220	Allocation ID	eipalloc-0f6a3c21
Instance	i-04d6a0a4902e69eed	Private IP address	10.0.0.202
Scope	vpc	Association ID	eipassoc-ac0bf68d
Public DNS	ec2-18-217-63-220.us-east-2.compute.amazonaws.com	Network interface ID	eni-3edea561
Network interface owner	636737052227		

Figure 42. The window with elastic IPs

The next thing to do was the association of elastic (public) IP address to an instance. This process as completely the same as mentioned in chapter five, section one, step four, and thus I will not cover it. The result can be seen in Figure 42. The selected IP is the new one.

5.2.3 Configuration of domain controller

The next important part of creating of a cluster is configuration of domain controller. I fulfilled all the needed pre-requisites and the next step was to connect to the new instance. Each operating system has its own method of connection. I use MacOS, and the needed tool for it is Microsoft Remote Desktop. It can be found in Mac App Store, and it’s free.



Microsoft Remote Desktop 8.0 4+

With the Microsoft Remote Desktop app, you can connect to a remote PC and your work resources from almost anywhere. Experience the power of Windows with RemoteFX in a Remote Desktop client designed to help you get your work done wherever you are.

What's New in Version 8.0.43
General bug fixes.

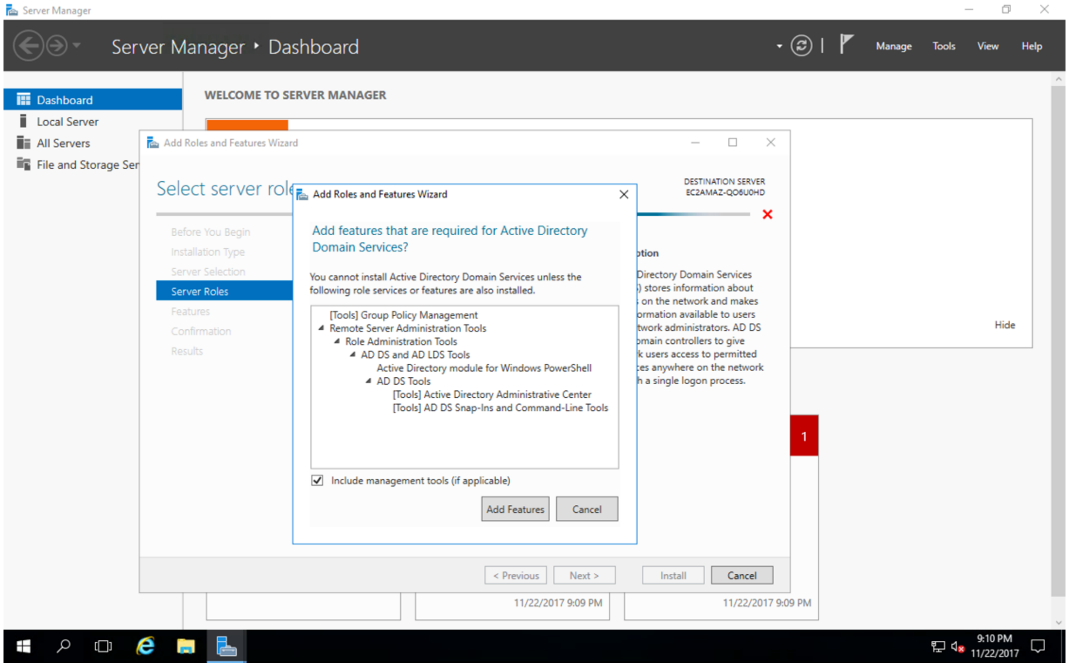
Information
Category: Business
Updated: 27 September 2017
Version: 8.0.43
Price: Free
Size: 14.9 MB
Family Sharing: Yes
Language: English
Developer: Microsoft Corporation
© 2017 Microsoft Corporation. All rights reserved.

Rated 4+
Compatibility:
OS X 10.9 or later, 64-bit processor

More Apps by Microsoft Corpor...
Microsoft OneNote
Productivity
OneDrive

Figure 43. Window of Microsoft Remote Desktop 8.0 in Mac App Store

After the actions done in chapter five, section two, step two, RDP access was not enabled on the instance. Therefore, I needed to manually create a new rule in web GUI. I allowed RDP traffic via 3389 port from everywhere (this is acceptable because instance still asks for password which is encrypted).



DESTINATION SERVER
EC2AMAZ-QO6U0HD

Add Roles and Features Wizard

Add features that are required for Active Directory Domain Services?

You cannot install Active Directory Domain Services unless the following role services or features are also installed.

[Tools] Group Policy Management
Remote Server Administration Tools
Role Administration Tools
AD DS and AD LDS Tools
Active Directory module for Windows PowerShell
AD DS Tools
[Tools] Active Directory Administrative Center
[Tools] AD DS Snap-Ins and Command-Line Tools

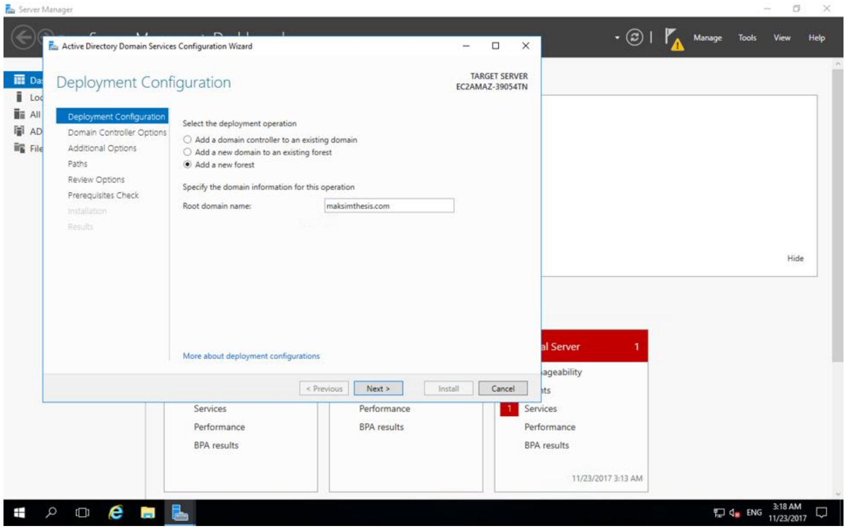
☑ Include management tools (if applicable)

Add Features | Cancel

ption
Directory Domain Services
) stores information about
on the network and makes
ormation available to users
twork administrators. AD DS
omain controllers to give
k users access to permitted
es anywhere on the network
h a single logon process.

Figure 44. Process of installation of Active Directory Domain Services

As a result, I succeeded to connect to a server using the credentials provided by Amazon. The system operated quite slow, but in my opinion this happened due to the fact that me and the server are in almost opposite parts of the globe.



Deployment Configuration

TARGET SERVER
EC2AMAZ-39054TN

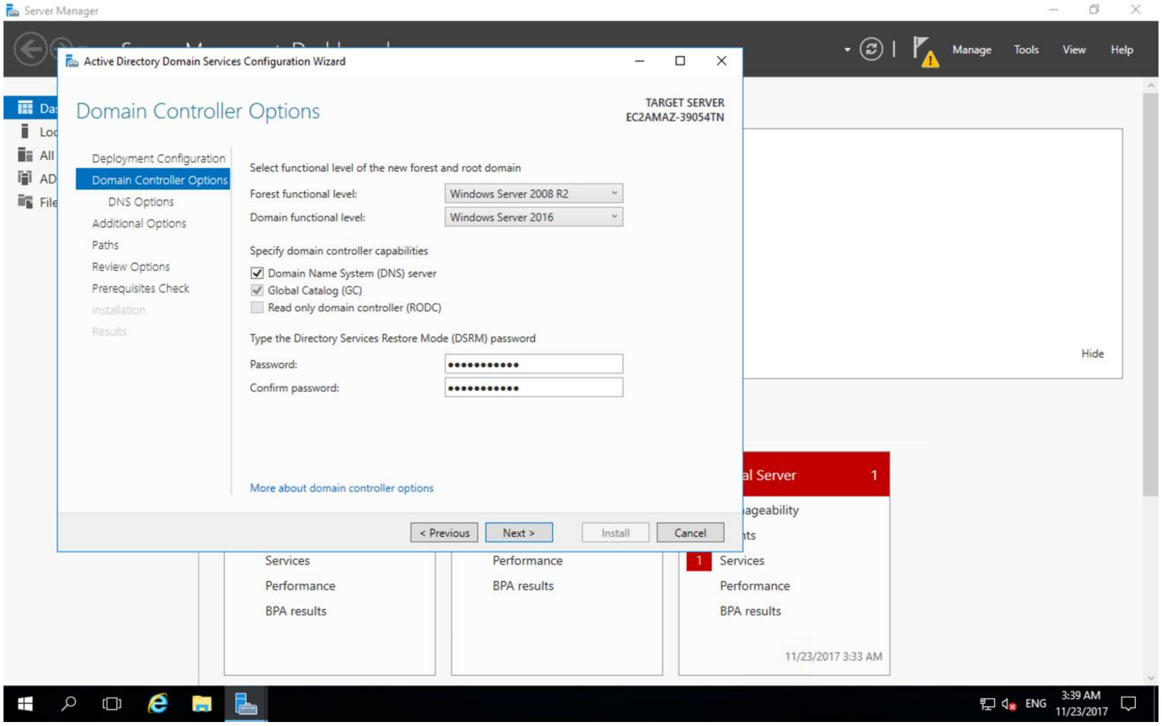
Select the deployment operation

○ Add a domain controller to an existing domain
○ Add a new domain to an existing forest
● Add a new forest

Specify the domain information for this operation

Root domain name: maksimthesis.com

Figure 45. Window of creation of a new forest



Domain Controller Options

TARGET SERVER
EC2AMAZ-39054TN

Select functional level of the new forest and root domain

Forest functional level: Windows Server 2008 R2

Domain functional level: Windows Server 2016

Specify domain controller capabilities

☑ Domain Name System (DNS) server
☑ Global Catalog (GC)
□ Read only domain controller (RODC)

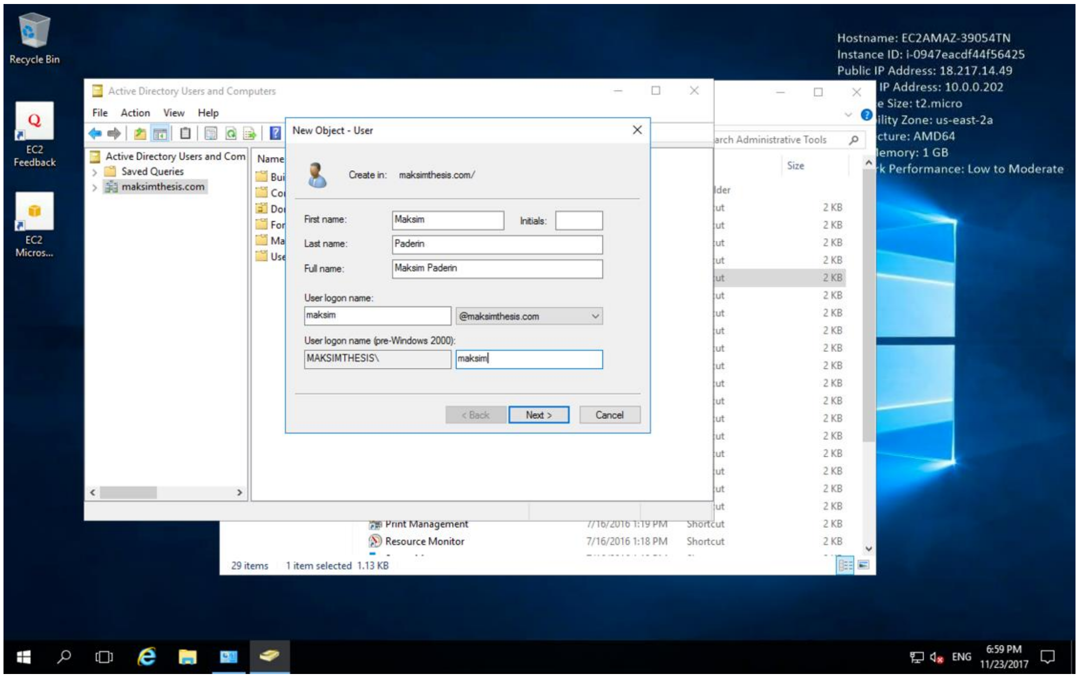
Type the Directory Services Restore Mode (DSRM) password

Password: ●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●

Figure 46. The window with domain controller options

While completeing this step I did not follow the tutorial completely. The tutorial suggests that I need to use *hpc.local* as my domain name, but I used *maksimthesis.com*. This should not cause any errors in the future, I will just need to use *maksimthesis.com* in each step mentioning *hpc.local*. After the creation of the new forest the server restarted. After the reboot I logged in as a member of the new forest.



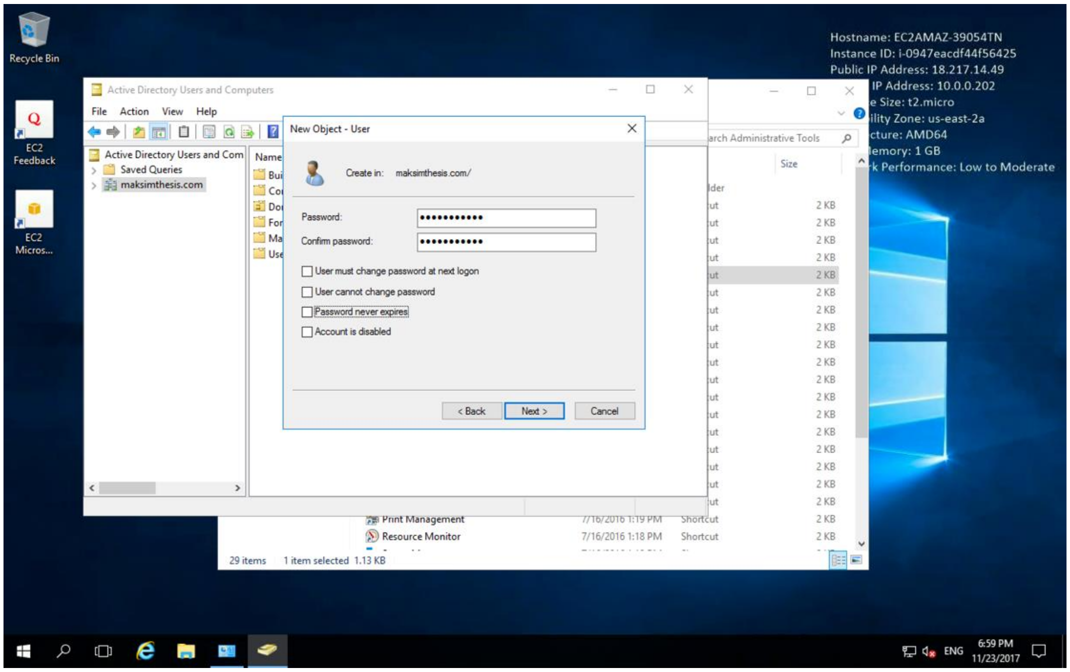
New Object - User

Create in: maksimthesis.com/

First name: Maksim
Initials:
Last name: Paderin
Full name: Maksim Paderin
User logon name: maksim @maksimthesis.com
User logon name (pre-Windows 2000): MAKSIMTHESIS\ maksim

< Back | Next > | Cancel

Figure 47. Creation of a domain user in Windows Server



New Object - User

Create in: maksimthesis.com/

Password: •••••••••••
Confirm password: •••••••••••

□ User must change password at next logon
□ User cannot change password
□ Password never expires
□ Account is disabled

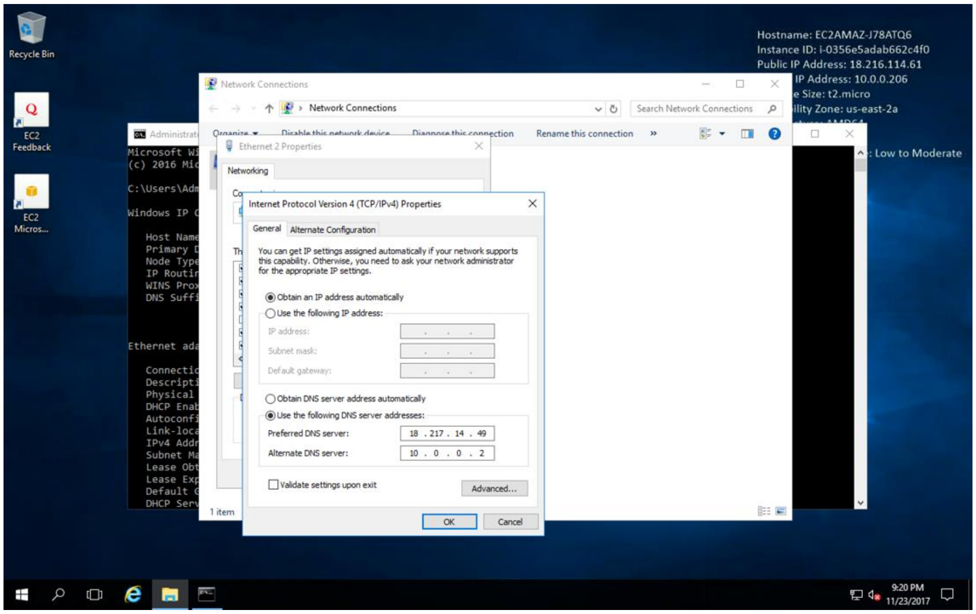
< Back | Next > | Cancel

Figure 48. Domain user password policies in Windows Server

While configuring a user, I improvised a little, and created a user with the name *maksim* instead of *hpcuser*. It should not affect the later configurations if I replace all *hpcuser*’s with *maksim*.

5.2.4 Creation of head node

First of all, I need to create an instance and associate it with two of previously created security groups, hpc cluster and domain member ones. I've covered the process of creation of instances, and thus I will not do it again. Then I needed to connect to this instance. Unlike domain controller instance, HPC head one should not have an elastic (public) IP address, but I was not able to connect to the instance without it. Therefore, I associated an IP address with it. I found out DNS settings using command *IPConfig /all*.



Hostname: EC2AMAZ-J78ATQ6
Instance ID: i-0356e5adab662c4f0
Public IP Address: 18.216.114.61
IP Address: 10.0.0.206
e Size: t2.micro
ility Zone: us-east-2a

Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:
IP address: . . .
Subnet mask: . . .
Default gateway: . . .

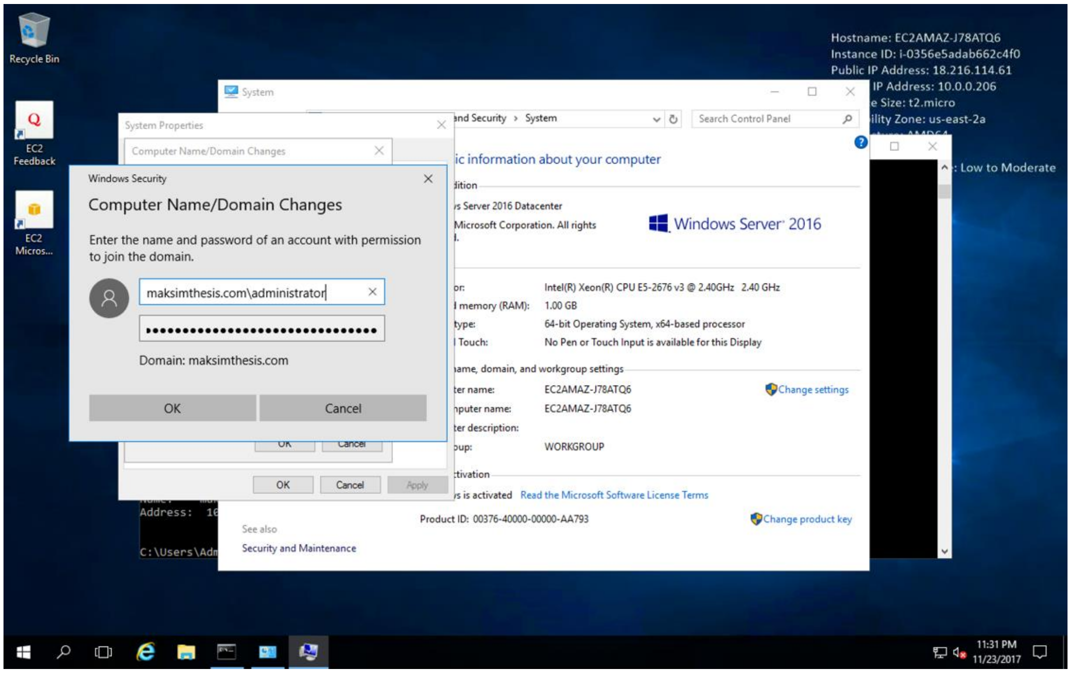
○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 18 . 217 . 14 . 49
Alternate DNS server: 10 . 0 . 0 . 2

□ Validate settings upon exit

```
Microsoft Wi
(c) 2016 Mic

C:\Users\Adm

Windows IP C

   Host Name
   Primary D
   Node Type
   IP Routin
   WINS Prox
   DNS Suffi


Ethernet ada

   Connectio
   Descripti
   Physical
   DHCP Enab
   Autoconfi
   Link-loca
   IPv4 Addr
   Subnet Ma
   Lease Obt
   Lease Exp
   Default G
   DHCP Serv
```

Figure 49. Network Connections window with IPv4 properties window open

Then I went to the Network Connections, where I edited the DNS addresses. These addresses can be seen in Figure 52. The next thing to was the assigning of new instance to eisting domain.



Hostname: EC2AMAZ-J78ATQ6
Instance ID: i-0356e5adab662c4f0
Public IP Address: 18.216.114.61

Windows Security

Computer Name/Domain Changes

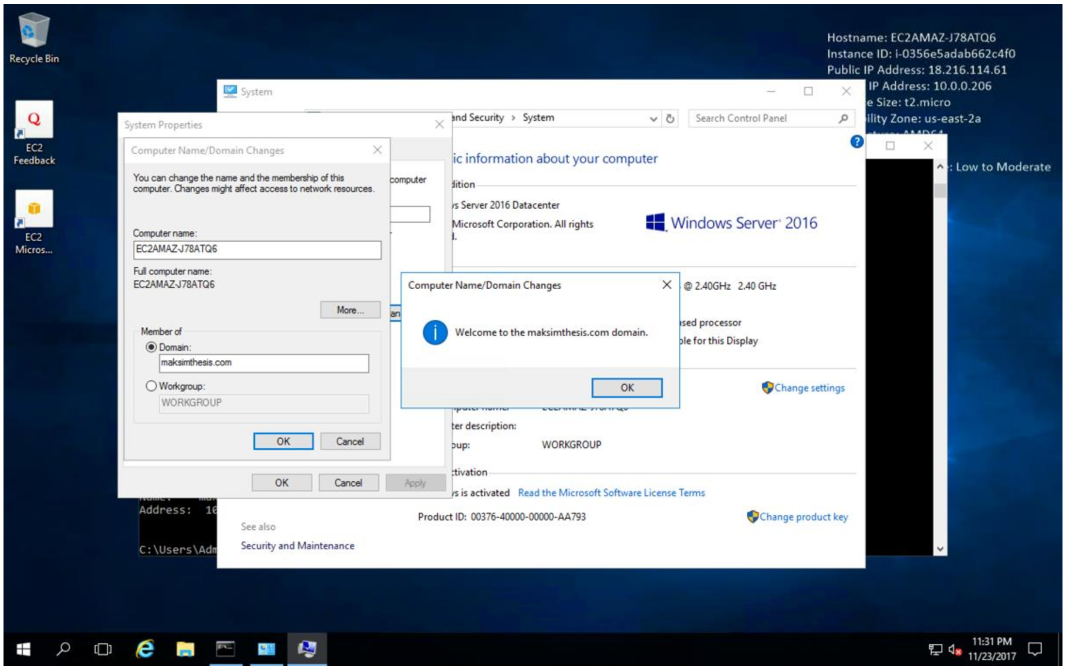
Enter the name and password of an account with permission to join the domain.

maksimthesis.com\administrator

••••••••••••••••••••••••••••••••

Domain: maksimthesis.com

Figure 50. Login window for maksimthesis.com domain



Computer Name/Domain Changes

You can change the name and the membership of this computer. Changes might affect access to network resources.

Computer name: EC2AMAZ-J78ATQ6

Full computer name: EC2AMAZ-J78ATQ6

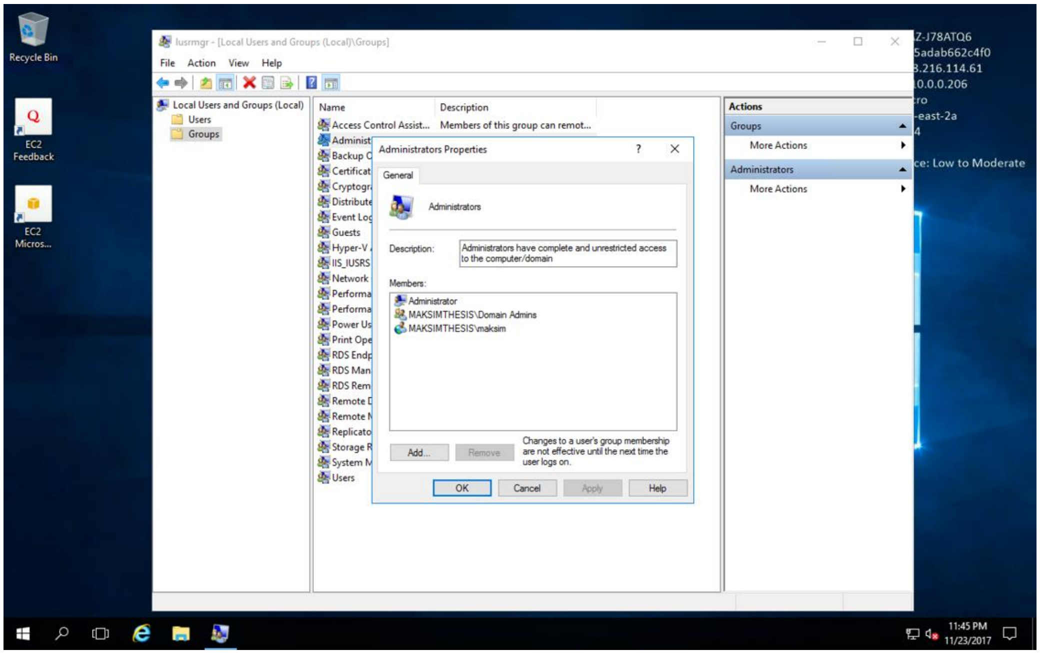
Member of
● Domain: maksimthesis.com
○ Workgroup: WORKGROUP

Computer Name/Domain Changes

Welcome to the maksimthesis.com domain.

Figure 51. Approval message from maksimthesis.com domain

I was successful to add the new instance to the maksimthesis.com domain using administrator credentials from domain controller instance. The next step was adding previously created *maksim* user to administrators group locally.



Administrators Properties

General

Administrators

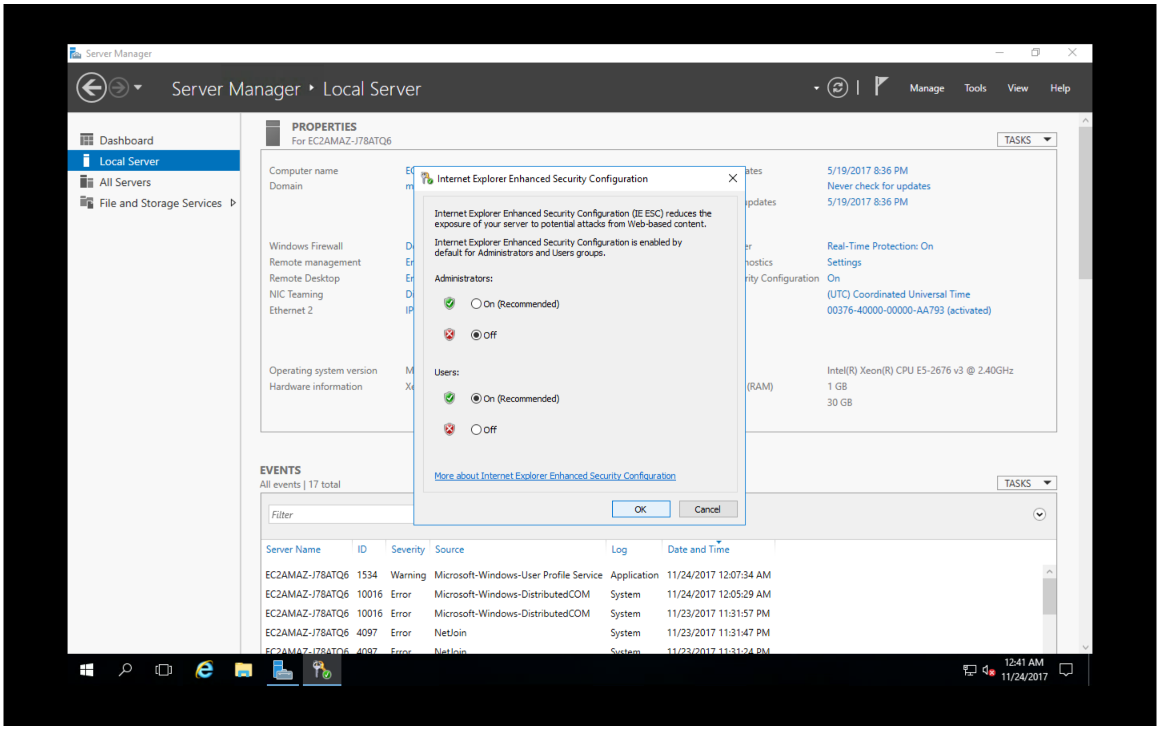
Description: Administrators have complete and unrestricted access to the computer/domain

Members:
- Administrator
- MAKSIMTHESIS\Domain Admins
- MAKSIMTHESIS\maksim

Changes to a user's group membership are not effective until the next time the user logs on.

Figure 52. Administrators list for HPC-head instance

The next important part of the configuration of HPC-head node is installing HPC pack. There is a prerequisite, I needed to disable Internet Explorer Enhanced Security Configuration in Server Manager. This is required because in the next step I downloaded the file from Microsoft website.



Internet Explorer Enhanced Security Configuration

Internet Explorer Enhanced Security Configuration (IE ESC) reduces the exposure of your server to potential attacks from Web-based content.

Internet Explorer Enhanced Security Configuration is enabled by default for Administrators and Users groups.

Administrators:

○ On (Recommended)

● Off

Users:

● On (Recommended)

○ Off

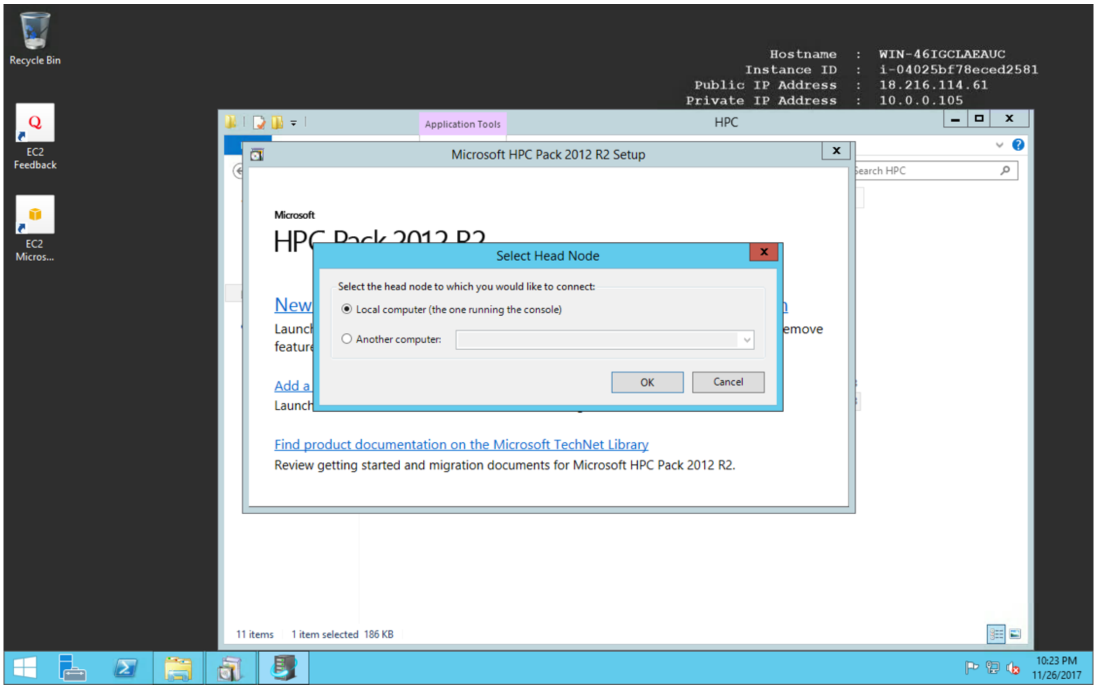
More about Internet Explorer Enhanced Security Configuration

Server Name	ID	Severity	Source	Log	Date and Time
EC2AMAZ-J78ATQ6	1534	Warning	Microsoft-Windows-User Profile Service	Application	11/24/2017 12:07:34 AM
EC2AMAZ-J78ATQ6	10016	Error	Microsoft-Windows-DistributedCOM	System	11/24/2017 12:05:29 AM
EC2AMAZ-J78ATQ6	10016	Error	Microsoft-Windows-DistributedCOM	System	11/23/2017 11:31:57 PM
EC2AMAZ-J78ATQ6	4097	Error	NetJoin	System	11/23/2017 11:31:47 PM

Figure 53. Window with IE ESC settings

I decided not to use HPC Pack 2016 and instead of it switch to HPC Pack 2012 R2. 2016 version requires a signed and encrypted cryptography archive with .pfx extension. I made a small research about self-generating these archives, but in the end just abandoned this idea and selected a 2012 version.

I faced another issue while doing this. The problem was in the fact that HPC Pack 2012 R2 requires Windows Server 2012 R2. Therefore, I was forced to abandon my old instance, create the new one as Windows Server 2012 R2 and repeat all the steps on it. I will not replace the previous figures mentioned in this big step, because everything except the operating system interface is completely the same.



```
         Hostname  :  WIN-46IGCLAEAUC
      Instance ID  :  i-04025bf78eced2581
Public IP Address  :  18.216.114.61
Private IP Address :  10.0.0.105
```

Microsoft HPC Pack 2012 R2 Setup

Select Head Node

Select the head node to which you would like to connect:

● Local computer (the one running the console)

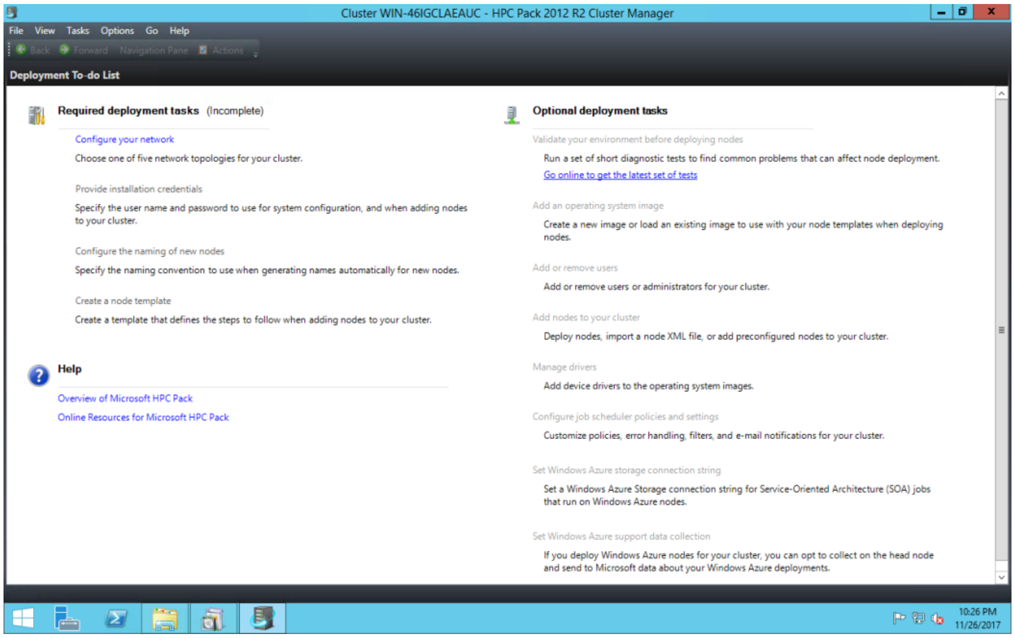
○ Another computer:

Find product documentation on the Microsoft TechNet Library

Review getting started and migration documents for Microsoft HPC Pack 2012 R2.

Figure 54. Initiual configuration Window right after HPC Pack 2012 R2 installation process

The next required action is to actually create a cluster. This is done via HPC cluster manager, which was installed with HPC Pack 2012 R2. The process is quite easy in comparison to all the issues I faced during the pack installation.



Required deployment tasks (Incomplete)

Configure your network

Choose one of five network topologies for your cluster.

Provide installation credentials

Specify the user name and password to use for system configuration, and when adding nodes to your cluster.

Configure the naming of new nodes

Specify the naming convention to use when generating names automatically for new nodes.

Create a node template

Create a template that defines the steps to follow when adding nodes to your cluster.

Help

Overview of Microsoft HPC Pack

Online Resources for Microsoft HPC Pack

Optional deployment tasks

Validate your environment before deploying nodes

Run a set of short diagnostic tests to find common problems that can affect node deployment.

Go online to get the latest set of tests

Add an operating system image

Create a new image or load an existing image to use with your node templates when deploying nodes.

Add or remove users

Add or remove users or administrators for your cluster.

Add nodes to your cluster

Deploy nodes, import a node XML file, or add preconfigured nodes to your cluster.

Manage drivers

Add device drivers to the operating system images.

Configure job scheduler policies and settings

Customize policies, error handling, filters, and e-mail notifications for your cluster.

Set Windows Azure storage connection string

Set a Windows Azure Storage connection string for Service-Oriented Architecture (SOA) jobs that run on Windows Azure nodes.

Set Windows Azure support data collection

If you deploy Windows Azure nodes for your cluster, you can opt to collect on the head node and send to Microsoft data about your Windows Azure deployments.

Figure 55. Window of HPC Cluster Manager

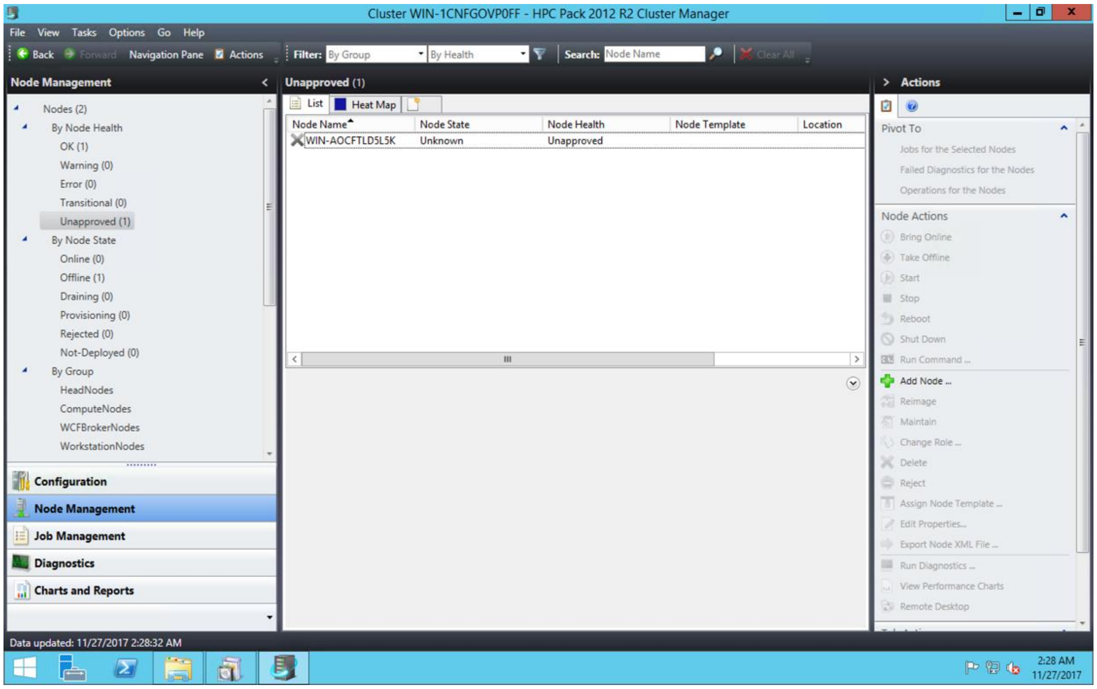
The whole process of configuring things in HPC Cluster Manager mostly consists of accepting the default values provided by the software. The whole process is described in AWS tutorial (Amazon Web Services, Inc.), and thus I will not attach any figures for this step. After this process, head node is created and configured, and there is a need to create a compute node.

5.2.5 Creation of compute node

The first big substep is completely the same as in chapter five, section two, step five, and this is the creation of an instance, configuration of DNS and domain user accounts. I didn’t want to repeat my own mistakes and created Windows Server 2012 R2 version from the very beginning. I will skip the description of this substep because of the reasons mentioned above.

The second big substep also almost completely repeats the process mentioned in the chapter five, section two, step five. The only crucial difference during HPC Pack 2012 R2 installation is that instead of choosing the option “Create a new HPC cluster by creating a head node” the option “Join an existing HPC cluster by creating a new compute node” needs to be selected. I will neither attach any figures nor describe this process as well.

The next important configuration to be done is to add the new compute node to existing cluster. This is done via head node, thus, I need to connect to HPC-Head instance once again and make the configurations from there.



Node Name	Node State	Node Health	Node Template	Location
WIN-AOCFTLD5L5K	Unknown	Unapproved		

Data updated: 11/27/2017 2:28:32 AM

Figure 56. Compute node listed as unapproved in head node's cluster manager

As can be seen in Figure 59, the compute node is listed as unapproved in HPC Pack 2012 R2 Cluster Manager on HPC-Head node. It’s the way it should be. The next thing to do is just to bring that unapproved node to life. After that, my cluster is ready.

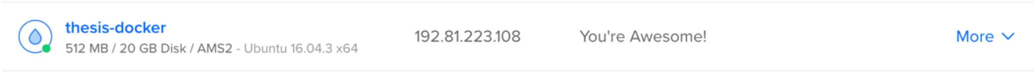
I decided not to implement any apps on my Windows cluster, because head and compute nodes lost connection to each other very often, and I initially had plans on implementing Docker in practical part of the project as well. I decided to stick with this idea, but implement it on Linux server instead of Windows one.

5.3 Experimenting with Docker

As mentioned before, I am really interested in trying Docker, a containers technology which supports clustering as well. I used tutorials from DigitalOcean website (finid et al., 2014-2017).

5.3.1 Installing prerequisites

First of all, I needed to run a server to host a docker. I chose a DigitalOcean for this task, and created the very basic Ubuntu 16.04 droplet possible. Droplet is a DigitalOcean name for a server instance.

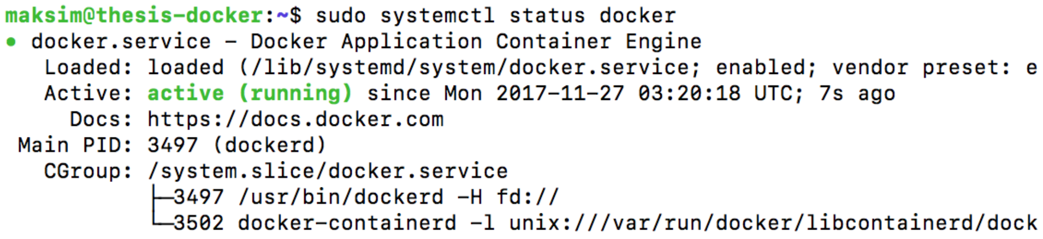


thesis-docker
512 MB / 20 GB Disk / AMS2 - Ubuntu 16.04.3 x64
192.81.223.108
You're Awesome!
More

Figure 57. Properties of my droplet

The command used to get connection to a server is *ssh root@192.81.223.108*. This command will probably not run on Windows machines, but will work on Linux and MacOS ones. I ran all the commands from my MacBook, and thus it worked.

The first prerequisite mentioned in a main guide (finid, 2017) is installation of Docker. For this, I needed a non-root user with sudo priveleges. It was implemented via issuing *adduser maksim* and *usermod –aG sudo maksim*. The next steps are straightforward because the DigitalOcean guide (finid, 2016) covers in a proper and clear way. The step mostly included the usage of apt-related commands. As can be seen in Figure 61 below, the Docker was installed successfully.

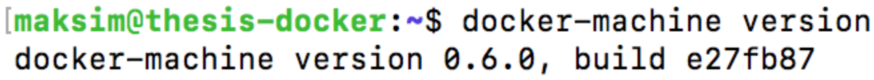


```
maksim@thesis-docker:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Mon 2017-11-27 03:20:18 UTC; 7s ago
     Docs: https://docs.docker.com
 Main PID: 3497 (dockerd)
   CGroup: /system.slice/docker.service
           ├─3497 /usr/bin/dockerd -H fd://
           └─3502 docker-containerd -l unix:///var/run/docker/libcontainerd/dock
```

Figure 58. Validation of Docker being installed

The next task was to receive a DigitalOcean token. I used the guide (Mitchell Anicas, 2014) which was very clear and easy to follow. The process included the receiving of a token via web GUI, and assigning this token to a variable in Linux. I didn’t face any issues. For security reasons, I will not provide my security token in this document.

The third, and the last, step to do was installation of Docker Machine. As in case with Docker installation, I also used a guide by a user finid (finid, 2016) which was also very straightforward. The installation mostly included configurations via built-in editors and permissions-changing commands.

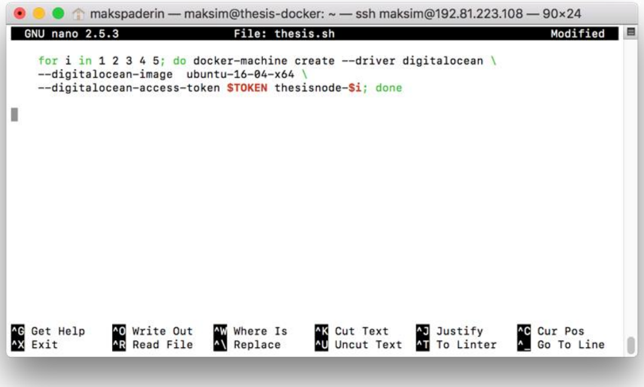


```
maksim@thesis-docker:~$ docker-machine version
docker-machine version 0.6.0, build e27fb87
```

Figure 59. Validation of Docker machine being installed

5.3.2 Creation of a cluster

First of all, I needed to write a Bash script which would automate the process of creating several nodes. It acts as a loop.

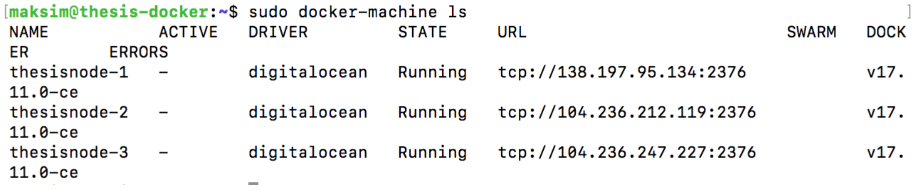


makspaderin — maksim@thesis-docker: ~ — ssh maksim@192.81.223.108 — 90×24

```
  GNU nano 2.5.3                  File: thesis.sh                           Modified

   for i in 1 2 3 4 5; do docker-machine create --driver digitalocean \
   --digitalocean-image  ubuntu-16-04-x64 \
   --digitalocean-access-token $TOKEN thesisnode-$i; done
```

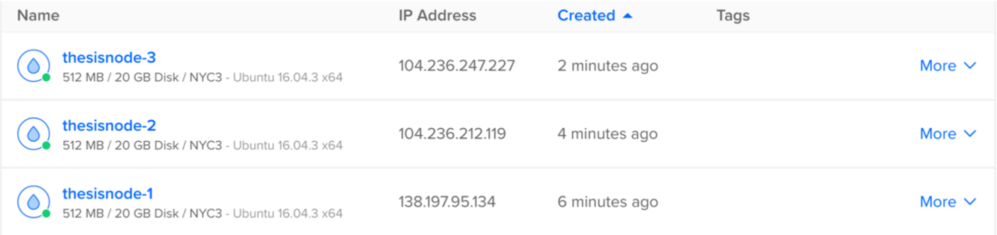
Figure 60. My first Bash script for Docker machine

The Figure 63 shows an example script for this purpose. In the end, I decided to run only three nodes, as in tutorial, instead of five. The nodes are created as separate DigitalOcean droplets.



```
maksim@thesis-docker:~$ sudo docker-machine ls
NAME           ACTIVE   DRIVER         STATE     URL                          SWARM   DOCK
ER         ERRORS
thesisnode-1   -        digitalocean   Running   tcp://138.197.95.134:2376            v17.
11.0-ce
thesisnode-2   -        digitalocean   Running   tcp://104.236.212.119:2376           v17.
11.0-ce
thesisnode-3   -        digitalocean   Running   tcp://104.236.247.227:2376           v17.
11.0-ce
```

Figure 61. Nodes of my Docker cluster in terminal

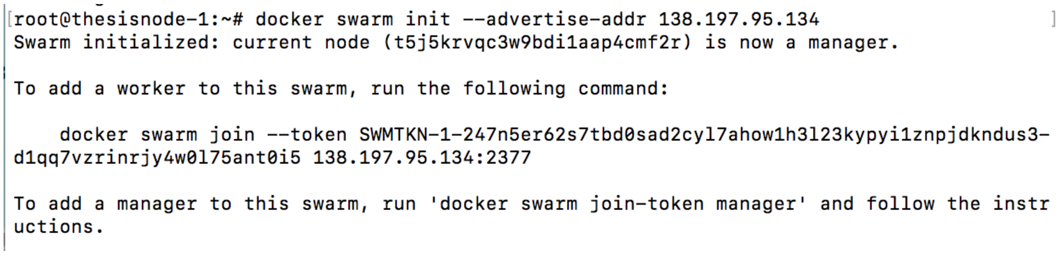


Name	IP Address	Created ▲	Tags	
thesisnode-3 512 MB / 20 GB Disk / NYC3 - Ubuntu 16.04.3 x64	104.236.247.227	2 minutes ago		More ⌄
thesisnode-2 512 MB / 20 GB Disk / NYC3 - Ubuntu 16.04.3 x64	104.236.212.119	4 minutes ago		More ⌄
thesisnode-1 512 MB / 20 GB Disk / NYC3 - Ubuntu 16.04.3 x64	138.197.95.134	6 minutes ago		More ⌄

Figure 62. Nodes of my Docker cluster in DigitalOcean web GUI

The next configuration to do was the firewall one. A very useful feature of docker is possibility to access nodes within terminal window of docker via issuing *sudo docker-machine ssh *the_name_of_the_node**. It works exactly the same way as *ssh user@ip* in Terminal, but runs all the commands as root by default. Using this command and method one mentioned in another excellent tutorial by Finid (finid, 2017), I configured the firewall on each of the three nodes.

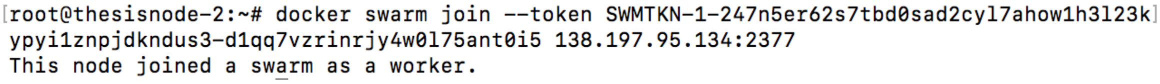
The next step in configuration was to initialize the cluster manager. As in the tutorial, my cluster's manager will be thesisnode-1.



```
root@thesisnode-1:~# docker swarm init --advertise-addr 138.197.95.134
Swarm initialized: current node (t5j5krvqc3w9bdi1aap4cmf2r) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-247n5er62s7tbd0sad2cyl7ahow1h3l23kypyi1znpjdkndus3-
d1qq7vzrinrjy4w0l75ant0i5 138.197.95.134:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instr
uctions.
```

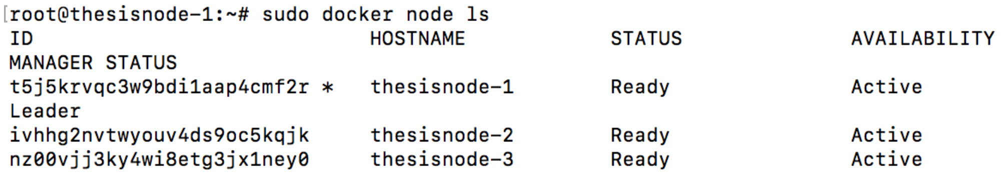
Figure 63. Promotion command issued on thesisnode-1 and its output

After that, the remaining nodes must be added to my cluster. It was very easy due to the needed command being provided by swarm initializer. This command required a Swarm token, or a Swarm ID, which shows which Swarm is the target to join, and IP of the manager, pointed at 2377 port. Exactly the same process was repeaten twice, for both thesisnode-2 and thesisnode-3.



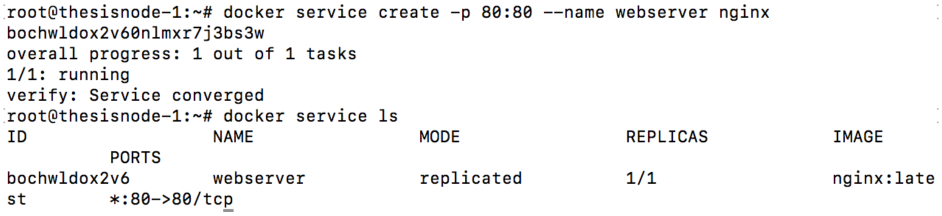
```
root@thesisnode-2:~# docker swarm join --token SWMTKN-1-247n5er62s7tbd0sad2cyl7ahow1h3l23k
ypyi1znpjdkndus3-d1qq7vzrinrjy4w0l75ant0i5 138.197.95.134:2377
This node joined a swarm as a worker.
```

Figure 64. Command for joining the swarm, and its output



```
root@thesisnode-1:~# sudo docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY
MANAGER STATUS
t5j5krvqc3w9bdi1aap4cmf2r *   thesisnode-1        Ready               Active
Leader
ivhhg2nvtwyouv4ds9oc5kqjk     thesisnode-2        Ready               Active
nz00vjj3ky4wi8etg3jx1ney0     thesisnode-3        Ready               Active
```

Figure 65. Cluster overview command issued on thesisnode-1, and its output



```
root@thesisnode-1:~# docker service create -p 80:80 --name webserver nginx
bochwldox2v60nlmxr7j3bs3w
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged
root@thesisnode-1:~# docker service ls
ID                  NAME                MODE                REPLICAS            IMAGE
          PORTS
bochwldox2v6        webserver           replicated          1/1                 nginx:late
st        *:80->80/tcp
```

Figure 66. Creation of nginx webserver on the cluster and its verification

As a result, Nginx startpage can be accessed via typing any of the cluster's IP addresses in a web browser. http://138.197.95.134/, http://104.236.212.119/ and http://104.236.247.227/ show exactly the same page, which means, that they operate as the whole one machine.

6 RESULTS & CONCLUSION

The theory part was very interesting and beneficial for me. The initial goal was achieved, and I studied the principles and the appliances of the clustering. I also learned a lot about how different operating systems and technologies have different methods of implementing the same models and solving the same issues. One of the new interesting aspects of this project for me was the study about virtualization and cloud technologies, which are closely related to the clustering.

As for practical part, the initial goal, which is the implementation of cluster, was also achieved. Moreover, in the end it became a comparison of AWS + Windows + GUI with DigitalOcean + Linux + Docker + CLI. Implementing two completely different solutions was an interesting thing to do, and I can subjectively judge what method I liked the most.

As for cloud providers, I prefer DigitalOcean over AWS for several reasons. Firstly, DigitalOcean has clearer design, and thus it's easier to find needed settings. Secondly, the tutorial base is supported by both DigitalOcean employees and users, and it leads to a much greater choice. The main disadvantage of DigitalOcean for my project was impossibility to run Windows in there, but in general DigitalOcean is more targeted on small-sized solutions, whereas AWS is much more enterprise-oriented.

As for Windows against Linux, I prefer Linux for server projects. During the configuration process of Windows, I faced a lot of problems. Sometimes I was forced to recreate the instance just because Windows failed an installation or lost connection to a domain. Linux always worked for me in exactly the same way as it was written in either documentation or tutorial. Docker was an extremely useful, and at the same time, very easy-to-follow technology, which allowed me to create a cluster within an evening. As for Windows, I spent about 5 days on different configurations, reinstallations of instances and troubleshooting of Windows issues, and decided not to implement any apps on Windows cluster due to it’s connectivity issues.

As for method of implementation, GUI-oriented solutions are generally much harder to configure incorrectly, although, I prefer CLI-oriented configurations for their fast and clear principles of work. Moreover, the more enterprise-oriented the network become, the more sence there is for the use of CLI methods due to much less overhead.

To sum up, as a result of this project, I managed to research the concept of clustering and implemented two clusters with completely different technologies, and also succeded to run a service on one of them.

REFERENCES

Microsoft. 2017. Microsoft Developer Network. WWW document. Available at: https://msdn.microsoft.com/ [Accessed 18 November 2017].

Justin Ellingwood, 14 January 2014. An Introduction to Networking Terminology, Interfaces, and Protocols. WWW document. Available at: https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols [Accessed 22 October 2017].

Microsoft. How a server cluster works. WWW document. Available at: https://technet.microsoft.com/en-us/library/cc738051(v=ws.10).aspx [Accessed 15 November 2017].

Microsoft, 28 March 2003. Clustering Technologies. WWW document. Available at: https://technet.microsoft.com/en-us/library/cc784941(v=ws.10).aspx [Accessed 23 October 2017].

Microsoft. Windows Clustering. WWW document. Available at: https://msdn.microsoft.com/en-us/library/aa373130(v=vs.85).aspx [Accessed 23 October 2017].

RohitShetty, 5 July 2011. High availability – cold, warm, hot. WWW document. https://www.ibm.com/developerworks/community/blogs/RohitShetty/entry/high_availability_cold_warm_hot?lang=en [Accessed 18 November 2017].

Microsoft. Understanding Quorum Configurations in a Failover Cluster. WWW document. Available at: https://technet.microsoft.com/en-us/library/cc731739.aspx [Accessed 26 October 2017].

IBM. Quorum types. WWW document. Available at: https://www.ibm.com/support/knowledgecenter/en/SGVKBA_3.2.2/com.ibm.rsct.admin/bl503_wiqt.htm [Accessed 26 October 2017].

Jethro Carr, 2008. Introduction to Linux clustering. WWW document. Available at: https://www.jethrocarr.com/wp-content/uploads/2010/09/introduction_linux_clustering_1.1.pdf [Accessed 26 October 2017].

Romain Serre, 17 November 2015. Understand failover cluster quorum. WWW document. Available at: http://www.tech-coffee.net/understand-failover-cluster-quorum/ [Accessed 28 October 2017].

Angelo Schalley, 08 May 2013. Hierarchical Design Model. Core – Distribution – Access layer. WWW document. Available at: https://www.schalley.eu/2013/05/08/hierarchical-design-model-core-distribution-access-layer/ [Accessed 30 October 2017].

Andreas M. Antonopoulos, 5 April 2005. Virtualization and clustering: combining two winning strategies. WWW document. Available at: https://www.networkworld.com/article/2319780/data-center/virtualization-and-clustering--combining-two-winning-strategies.html [Accessed 1 November 2017].

Sumologic, White Paper. What is Microsoft Azure, and Why Use it? PDF document. Available at: https://www.sumologic.com/wp-content/uploads/MicrosoftAzure-WP-1.pdf [Accessed 1 November 2017].

Microsoft. 2008. Microsoft Failover Cluster Power Shell. WWW document. Available at: https://technet.microsoft.com/en-us/library/ee461009.aspx [Accessed 2 November 2017].

Linux Cluster Manager. WWW document. Available at: http://linuxcm.sourceforge.net/ [Accessed 2 November].

CentOS. Cluster Administration GUI. WWW document. Available at: https://www.centos.org/docs/5/html/Cluster_Suite_Overview/s2-clumgmttools-overview-CSO.html [Accessed 2 November 2017].

Nothern.tech, Inc. CFEngine. WWW document. Available at: https://cfengine.com/product/ [Accessed 18 November 2017].

Free Software Foundation, Inc. 2010. GNU Parallel. WWW document. Available at: http://www.gnu.org/software/parallel/ [Accessed 18 November 2017].

Jeff Forsier, 2017. Fabric. WWW document. Available at: http://www.fabfile.org/ [Accessed 18 November 2017].

Chris Dunlap, 26 September 2017. MUNGE. WWW document. Available at: https://github.com/dun/munge [Accessed 18 November 2017].

Tobias Oetiker, 20 February 2017. RRDtool. WWW document. Available at: https://oss.oetiker.ch/rrdtool/ [Accessed 18 November 2017].

FreeIPMI Core Team, 10 July 2017. FreeIPMI. WWW document. Available at: http://www.gnu.org/software/freeipmi/index.html [Accessed 18 November 2017].

Thilina Uvindasiri. What is quorum disk and fencing wars? WWW document. Available at: https://www.tecmint.com/what-is-quorum-disk-and-a-fencing-wars/ [Accessed 2 November 2017].

Stuart Burns. A straightforward approach to Linux virtual clustering. WWW document. Available at: http://searchservervirtualization.techtarget.com/tip/A-straightforward-approach-to-creating-a-Linux-virtual-cluster [Accessed 2 November 2017].

VMWare, Inc. 2017. VMWare Documentation Center. WWW document. Available at: https://www.vmware.com/support/pubs/ [Accessed 18 November 2017].

VMware. Introduction to VMware DRS and VMware HA Clusters. WWW document. Available at: https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.wssdk.pg.doc_50%2FPG_Ch13_Resources.15.6.html [Accessed 3 November 2017].

VMware. vSphere Client and vSphere Web Client. WWW document. Available at: https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vsphere.introduction.doc_50%2FGUID-588861BB-3A62-4A01-82FD-F9FB42763242.html [Accessed 3 November 2017].

OVH. vCloud Air is now part of OVH. Official WWW portal of company. Available at: https://www.ovh.com/us/vcloud_air_powered_by_ovh/ [Accessed 3 November 2017].

Docker. Swarm mode overview. WWW document. Available at: https://docs.docker.com/engine/swarm/ [Accessed 3 November 2017].

Anand Akela. 5 May 2016. 4 Cluster management tools to compare. WWW document. Available at: https://blog.appdynamics.com/product/4-cluster-management-tools-to-compare/ [Accessed 3 November 2017].

Amazon Web Services, Inc. Launch a Windows virtual machine with Amazon EC2. WWW document. Available at: https://aws.amazon.com/getting-started/tutorials/launch-windows-vm/ [Accessed 8 November 2017].

Amazon Web Services, Inc. Tutorial: setting up a Windows HPC cluster on Amazon EC2. WWW document. Available at: http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ConfigWindowsHPC.html [Accessed 13 November 2017].

Amazon Web Services, Inc. Getting started with Amazon VPC. WWW document. Available at: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/getting-started-ipv4.html [Accessed 13 November 2017].

Juan Pablo Jofre et al. About Execution Policies. Publisher: Microsoft. WWW document. Available at: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-5.1 [Accessed 22 November 2017].

Amazon Web Services, Inc. AWS tools for Windows PowerShell. WWW document. Available at: https://aws.amazon.com/powershell/ [Accessed 22 November 2017].

Finid. 10 January 2017. How to Create a Cluster of Docker Containers with Docker Swarm in DigitalOcean on Ubuntu 16.04. WWW document. Available at: https://www.digitalocean.com/community/tutorials/how-to-create-a-cluster-of-docker-containers-with-docker-swarm-and-digitalocean-on-ubuntu-16-04 [Accessed 24 November 2017].

Finid. 3 November, 2016. How to install and use Docker on Ubuntu 16.04. WWW document. Available at: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04 [Accessed 24 November 2017].

Mitchell Anicas. 24 June, 2014. How to use the DigitalOcean API v2. WWW document. Available at: https://www.digitalocean.com/community/tutorials/how-to-use-the-digitalocean-api-v2 [Accessed 24 November 2017].

Finid. 20 May, 2016. How to provision and manage remote Docker hosts with Docker machine on Ubuntu 16.04. WWW document. Available at: https://www.digitalocean.com/community/tutorials/how-to-provision-and-manage-remote-docker-hosts-with-docker-machine-on-ubuntu-16-04 [Accessed 24 November 2017].

Finid. 9 January, 2017. How to configure the Linux firewall for Docker Swarm on Ubuntu 16.04. WWW document. Available at: https://www.digitalocean.com/community/tutorials/how-to-configure-the-linux-firewall-for-docker-swarm-on-ubuntu-16-04 [Accessed 24 November 2017].

Thomas Taege. 28 October, 2015. How to run Nginx in a Docker container in Ubuntu 14.04. WWW document. Available at: https://www.digitalocean.com/community/tutorials/how-to-run-nginx-in-a-docker-container-on-ubuntu-14-04#step-6-%E2%80%94-linking-the-container-to-the-local-filesystem [Accessed 24 November 2017].

LIST OF FIGURES

Figure 5. RedHat example of fencing enabled on cluster. RedHat. Available at: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Cluster_Suite_Overview/images/fence-example-pwr.png [Accessed 22 October 2017].

Figure 6. Very basic example of brain splitting. No author (figure is hosted on mirror website). Available at: https://image.slidesharecdn.com/hpserviceguard-arun-visolve-140423012535-phpapp01/95/high-availability-hp-serviceguard-9-638.jpg?cb=1398216584 [Accessed 22 October 2017].

Figure 7. The process of installing Failover Cluster Manager Tool. Microsoft, 6 April 2012. Available at: https://blogs.msdn.microsoft.com/clustering/2012/04/06/installing-the-failover-cluster-feature-and-tools-in-windows-server-2012/ [Accessed 22 October 2017].

Figure 8. Dashboard of MFCM. Microsoft. Available at: https://docs.microsoft.com/en-us/windows-server/storage/storage-replica/media/stretch-cluster-replication-using-shared-storage/storage_sr_onlinedisks2.png [Accessed 22 October 2017].

Figure 9. Node majority cluster structure. Penton, ITPro. File is not accessible anymore [Accessed 23 October 2017].

Figure 10. Node and disk majority cluster structure. Penton, ITPro. File is not accessible anymore [Accessed 23 October 2017].

Figure 11. Node and file share majority cluster structure. Penton, ITPro. File is not accessible anymore [Accessed 23 October 2017].

Figure 12. No majority / Disk only cluster structure. Penton, ITPro. File is not accessible anymore [Accessed 23 October 2017].

Figure 13. Cloud witness cluster(s) structure. February, 2015. Available at: http://blog.brodaric.com/wp-content/uploads/2015/02/CW-P2439-00.png [Accessed 25 October 2017].

Figure 14. LCM dashboard. LCM on Sourceforge.net. Available at: http://linuxcm.sourceforge.net/screen.html [Accessed 23 October 2017].

Figure 15. LCM image manager. LCM on Sourceforge.net. Available at: http://linuxcm.sourceforge.net/screen.html [Accessed 23 October 2017].

Figure 16. Cluster Administration GUI (Cluster Configuration Tool window) from Red Hat. Red Hat, Inc. Available at: https://www.centos.org/docs/5/html/Cluster_Suite_Overview/s2-clumgmttools-overview-CSO.html [Accessed 24 October 2017].

Figure 17. Cluster Status Tool. Red Hat, Inc. Available at: https://www.centos.org/docs/5/html/Cluster_Suite_Overview/s2-clumgmttools-overview-CSO.html [Accessed 24 October 2017].

Figure 18. Structure of VMware cluster (VMware HA) with an example of hardware issue and reboot process. vStackL, May 2016. Available at: http://www.vstackl.com/wp-content/uploads/2016/05/VMware-vSphere-HA-Example-vstackL.jpg [Accessed 25 October 2017].

Figure 19. vSphere Client window running in old version of Windows Server. VirtuallyLG, May 2012. Available at: https://virtuallylg.files.wordpress.com/2012/05/vcenter_ssh.jpg [Accessed 25 October 2017].