GLADINET, INC

Gladinet Cloud Enterprise

Multi-Site Deployment Guide

Gladinet, Inc.

9/27/2013

This document discusses the technologies behind Gladinet Cloud Enterprise

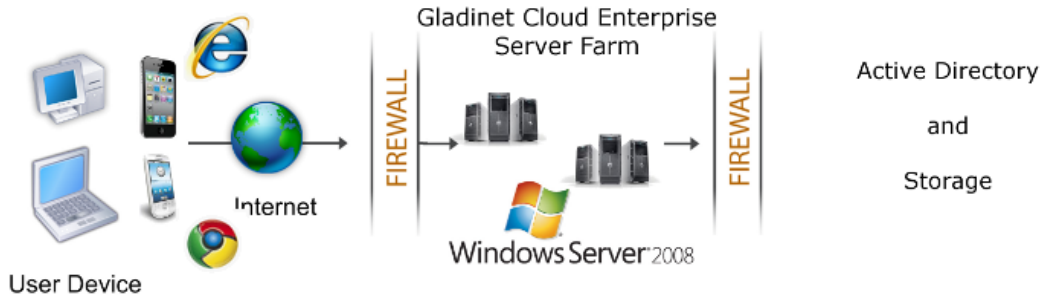
Table of Contents

Overview of Gladinet Cloud Enterprise

Gladinet Cloud provides value-added services on top of cloud storage services or local storage services. Cloud Storage services include those from OpenStack, Amazon S3 and its compatibles, Google Cloud Storage, HP Cloud Storage and many others. Local Storage Services include file server Storage, SAN or NAS storage. Gladinet Cloud value-added services can be summarized as Backup, Access, Sync and Share, Identity, Control and Collaboration (BASIC).

Gladinet Cloud Enterprise is a cluster of web services built on the Microsoft Web Platform. It provides the BASIC value-added services that facilitate online storage access for PCs, Macs, File Servers, Web Browsers, and Mobile Devices.

The services can be deployed in flexible combinations to meet different needs. For example, you can deploy it on-premise as a private cloud; or you can deploy it off-premise in a data center, managed by your managed service provider (MSP); or you can deploy it in Amazon EC2-like environment as virtual-private deployment.



There are three different types of machines (or Virtual Machine). In the smallest deployment unit, the three different logical nodes can co-exists in one single machine.

- Web Front Node
  The Account Management, Sign-in and Load-balancing services will be installed on this physical machine. Depending on the load, you may need 1 to N such nodes.

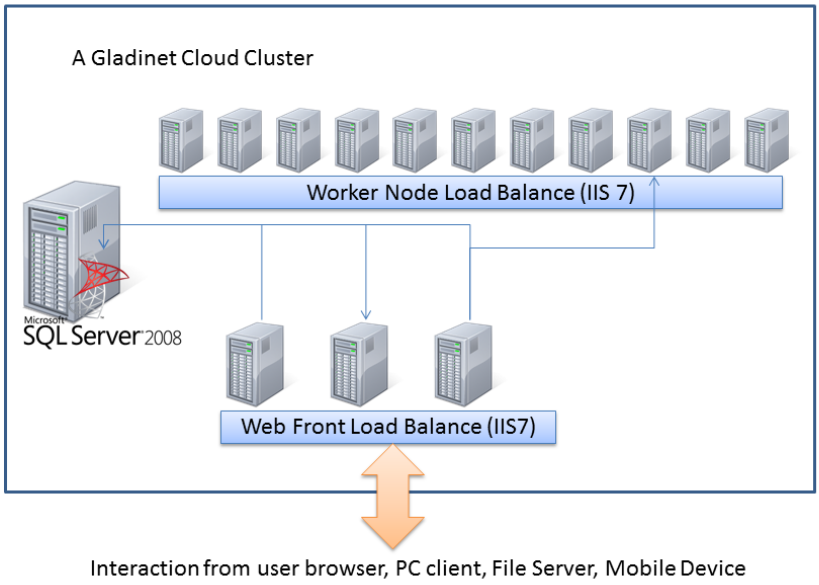
  Example: ACME Corporation deployed 2 web front nodes node1.acme.com and node2.acme.com. Each node is running a copy of Gladinet Cloud Enterprise, connected to the same SQL database.

  ACME Corporation acquired a domain name (DNS) of cloud.acme.com which load balances between node1.acme.com and node2.acme.com.

When a user points their browser to https://cloud.acme.com, it is directed to one of the nodes which host the login page.

- Worker Node
  This node will contain services like Web Browser Based File Manager, Storage Service Connectors, and etc. Again, additional nodes can be added as the load increases. Because there is cache information located on each node, users will have an affinity to a single node once it is assigned.

- Database Node
  The database contains persistent information for the system. In general, once a user is logged in, database access is no-longer needed for normal operation. If the database is down, most end user operations can continue with cached information (However, a user needs to access the database at least once to login and cache information).



Dependency Components

Gladinet Cloud Enterprise is built on top of the Microsoft Web Platform, including Internet Information Server 7 (IIS), .Net Framework 4, ASP.NET 4 and SQL Server or SQL Server Express. The base operating system is Windows 2008 with Service Pack 2 or R2. You can also use Windows 2012 too, which comes with .Net Framework 4.5 and ASP.NET 4.5.

Windows 2008 (SP2/R2)

The base operating system provides the base of the Microsoft Web Platform. It will be loaded with the mentioned Microsoft components before the Gladinet Cloud Enterprise is installed.

Note: (Updated 9/27/2013) Windows 2012 or Windows 2012 R2 are supported too.

SQL Server

SQL Server is used to store static configuration information, such as user name, email, storage configuration, file and folder sharing information, etc. It is recommended that the SQL Server has daily backups since it holds configuration information for the service to run properly.

[Multi-Site Note]: Since we will have SQL Server on multiple sites, SQL Server Standard and above Edition are recommended for the Replication related features. SQL Server 2008 R2 is recommended to setup transactional replication with Updatable Subscription and Queued Updating.

.Net Framework 4

Gladinet Cloud Enterprise Server is built with .Net Framework 4. It is also compatible with .Net Framework 4.5, which comes as default on Server 2012.

(Note: Most of the Access Clients are built with native code on each platform. Example, Windows Client built with Visual C++, Mac Client built with Object-C and etc.)

ASP.NET

Gladinet Cloud Enterprise web browser user interface is written in ASP.NET, HTML and Javascript.

Internet Information Server

Gladinet Cloud Enterprise services are hosted inside Internet Information Server(IIS). It provides brokerage functionalities between the Access Clients and the backend storage. It is also a value-add layer on top of the backend storage.

Recommended Hardware Specification

Memory: 4GB

Hard Drive: 500G

32-bit or 64-bit platform

Operating System: Windows 2008 Service Pack 2 or R2, Windows 2012 or Windows 2012 R2

CPU: Intel

(Virtual Machines are recommended.)

The following will be specific to multi-site deployment, Please reference the Gladinet Cloud Enterprise Deployment Guide first before looking into multi-site deployment.

Two-Site Deployment

Overall Architecture

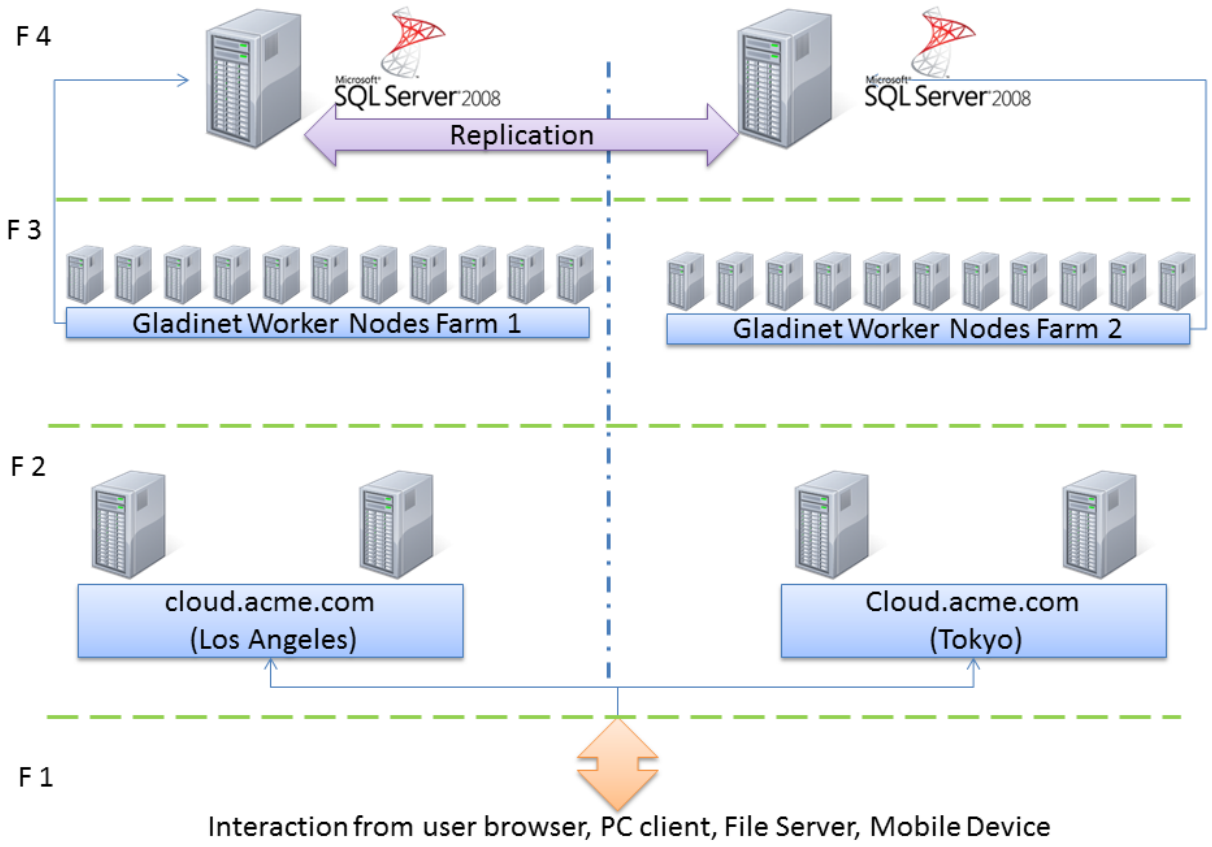
As shown in the following diagram, the architecture is broken into 4 different functionality layers.

F1 – A global load balancer that directs users to the nearest site. In the case of a disaster, the load balancer rule can be changed to direct users to one of the two sites.

F2 – The web front nodes can run the same Gladinet Cloud Enterprise software on the nodes. The nodes will have exactly same external DNS name. As shown in the picture, all the 4 different nodes will have cloud.acme.com external DNS name. The web front node's functionality is to do authentication and load balance a specific user to a specific worker node.

F3 – The worker nodes will have different external DNS name each. For example, for the worker nodes in Los Angeles, la1.acme.com, la2.acme.com, …., laN.acme.com, and for the worker nodes in Tokyo, tk1.acme.com, tk2.acme.com, tk3.acme.com, …, tkN.acme.com

F4 – The database nodes. For multi-site deployment, we recommend having one SQL Server per site to guaranteed local database access. The SQL Database between sites can be transactional replication with updatable subscription, and queued updating.

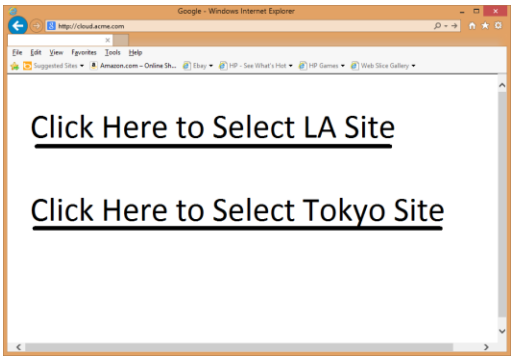

F 4
SQL Server 2008
Replication
SQL Server 2008
F 3
Gladinet Worker Nodes Farm 1
Gladinet Worker Nodes Farm 2
F 2
cloud.acme.com
(Los Angeles)
Cloud.acme.com
(Tokyo)
F 1
Interaction from user browser, PC client, File Server, Mobile Device

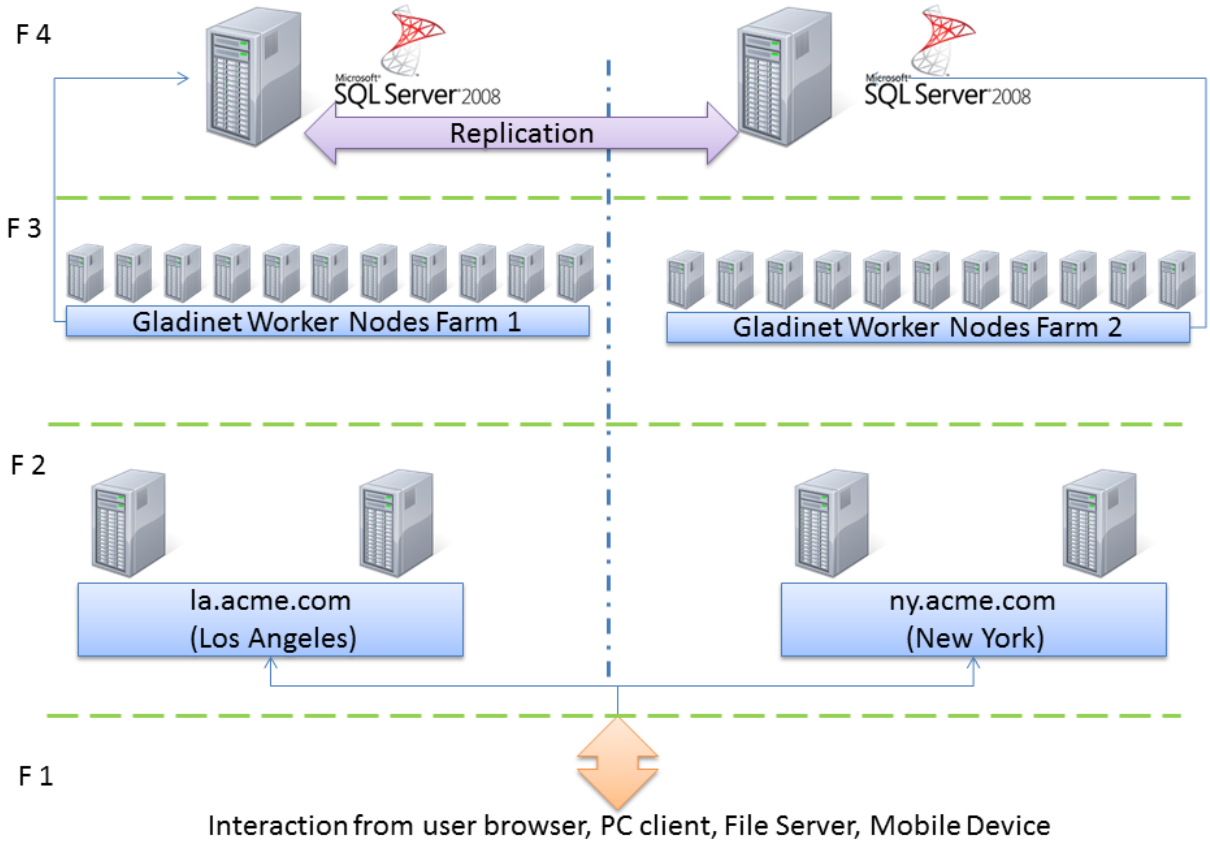
Global Load Balancer

First there needs to be a global load balancer that can direct users based on geographic location to the two different sites. For example, based on the user's connection and related information, the cloud.acme.com can direct users to either the ACME LA site or ACME Tokyo site.

Without Global Load Balancer

It is also possible to deploy the solution without the global load balancer. Instead of having cloud.acme.com as the DNS name on all the web front nodes, you can have la.acme.com as the DNS name for the web front nodes in the Los Angeles region, and have tokyo.acme.com as the DNS name for the web front nodes for the Tokyo region. You can provide a web page as simple as the following picture and ask the user to pick one data center to continue.





Web Front Nodes

For two-site deployments, there are two sets of web front nodes. The web front nodes will have wild card SSL certificates such as *.acme.com and they will be named the same such as cloud.acme.com. Without the global load balancer, you can also name the web front nodes as la.acme.com and tokyo.acme.com and provide a way for users to reach these two sites. The web front nodes will have their web.config files modified to point to the local group of worker nodes.

For example, on the LA web front nodes, please modify the web.config for the Default Web Site and add the following lines, (Assuming there are 3 worker nodes in the LA region)

```
<!--

      Load balancer

    -->

    <add key="LBNodeCount" value="3" />

    <add key="LBNode1"
value="https://la1.acme.com/namespace/n.svc/;https://la1.acme.com/portal/;la1.acme.com
:80;http://la1.acme.com:8080/localstor/n.svc/" />

    <add key="LBNode2"
value="https://la2.acme.com/namespace/n.svc/;https://la2.acme.com/portal/;la2.acme.com
:80;http://la2.acme.com:8080/localstor/n.svc/" />

  <add key="LBNode3"
value="https://la3.acme.com/namespace/n.svc/;https://la3.acme.com/portal/;la3.acme.com
:80;http://la3.acme.com:8080/localstor/n.svc/" />
```

You can also make the same modification to the Tokyo web front nodes. And replace laN.acme.com with tkN.acme.com, if you have more than 3 nodes, you just change the LBNodeCount to the real count and keep on adding LBNodeN entries.

Worker Nodes

Worker nodes are running the same Gladinet Cloud Enterprise software. The external DNS name will need to be on the same level as the DNS name of the web front node. For example, if the web front node DNS name is cloud.acme.com, the worker nodes need to be on the same DNS name level such as <name>.acme.com. If it is on a different level, such as <name>.la.acme.com, it will break the same domain cookie used in between nodes. Worker nodes are stateless and rely on the authentication done on the web front node. If a worker node's node name is not in the LBNode entry list, the worker node will not be part of the load. This is one of the way you can retire a worker node and replace it with a new one. Worker nodes need to be directly accessible by various Access clients. Their DNS names need to be public DNS name and the HTTPS ports need to be open.

Database Nodes

We recommend Microsoft SQL Server Standard edition and above to leverage replication features. The database content is changed in-frequently, such as when a user is added, a new team folder is published,

or a password is changed. It is static configuration information. There are two database replication modes we recommend. In practice, you can try each out and see which one fits you best.

Replication – transactional replication mode

In the transactional replication mode, there is a publisher and there are subscribers. We recommend SQL Server 2008 R2 standard edition and setup transactional replication with updatable subscription, and queued updating.

In this mode, the publisher will create an initial snapshot for the subscribers to use to 'seed' the subscriber database. After that, changes at the publisher will be distributed to the subscriber. Subscriber will use MSDTC to link to remote publisher so subscriber side of the updates will be pushed to the publisher too.

Database Notes

Since successful database operations are key to the operating of Gladinet Cloud Enterprise, we recommend the SQL database local to the Worker nodes and web front nodes. The locality provides the best reliability of the service and the replication will take care of the multi-site deployment.

Extra Subscribers for Backup

Since you are already replicating SQL database for multi-site deployment, you can also use extra subscribers in the replication for backup purpose.

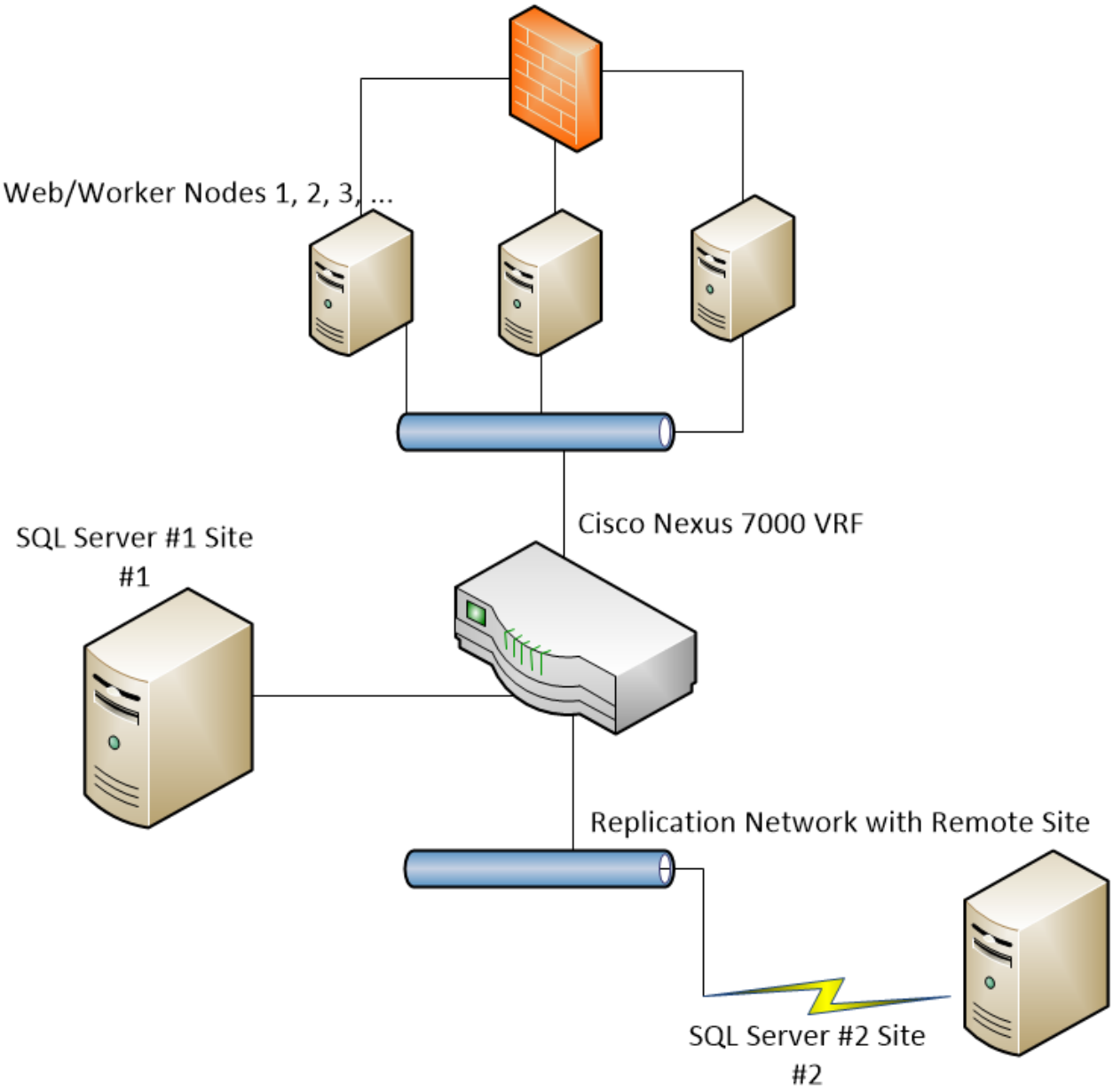
Multi-Site Deployment

Multi-site deployment is pretty much like the two-site deployment that you will replicate the SQL Server database among the sites.

Storage Notes

In this deployment guide, we didn't discuss the storage side of the deployment. Obviously, you need storage service available and replicated among the two sites as well. This deployment guide only focus on the infrastructure setup of the Gladinet Cloud Enterprise farm, which spans two sites and the replication of the persistent information in the SQL database.

Example Setup



In this example setup, all the servers are on Virtual Machine. The machine hosting the SQL Server is Windows 2008 R2 standard edition. The machine hosting the web node/worker node is Windows 2008 R2 web edition. All the VMs have one NIC. SQL Servers are all SQL 2008 R2 standard editions.

(a) All the SQL Servers are on the internal virtual private network across two sites. On this virtual private network, SQL server needs to do replication over SMB/CIFS (for snapshot), over SQL (for distribution) and over RPC (for MSDTC).
(b) All the servers have 1 NIC and the Cisco Nexus 7000 VRF route the traffic to the right place.
(c) Only port open on the firewall is port 443 for SSL traffic to the webnode/worker node and the SQL server can't be accessed from outside.