

# Dell EMC ECS: Configuring VMware NSX-T Load Balancer

## Abstract

This document describes how to configure a VMware® NSX-T load balancer with Dell EMC™ ECS.

July 2019

## Revisions

Date	Description
July 2019	Initial release

## Acknowledgements

This paper was produced by the Unstructured Technical Marketing Engineering and Solution Architects team.

Author: [Rich Paulson](#)

This document may contain certain words that are not consistent with Dell's current language guidelines. Dell plans to update the document over subsequent future releases to revise these words accordingly.

This document may contain language from third party content that is not under Dell's control and is not consistent with Dell's current guidelines for Dell's own content. When such third party content is updated by the relevant third parties, this document will be revised accordingly.

The information in this publication is provided "as is." Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2021 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. [3/8/2021] [Configuration and Deployment] [H17796]

# Table of contents

Revisions.....	2
Acknowledgements.....	2
Table of contents .....	3
Executive summary.....	5
Objectives .....	5
Audience .....	5
<b>1 Solution overview .....</b>	<b>6</b>
1.1 ECS overview .....	6
1.2 ECS constructs.....	7
1.3 Solution architecture .....	8
1.4 VMware NSX-T load balancer constructs .....	9
1.5 VMware NSX-T load balancer deployment options.....	11
1.6 Key components .....	12
<b>2 Solution implementation .....</b>	<b>14</b>
2.1 Implementation workflow .....	14
2.2 Installation and configuration steps .....	14
2.2.1 NSX Unified Appliance .....	14
2.2.2 Compute manager .....	15
2.2.3 Transport node .....	15
2.2.4 Transport zone.....	16
2.2.5 Tunnel endpoints .....	16
2.2.6 NSX edge .....	16
2.2.7 Segments.....	16
2.2.8 Tier-1 gateway .....	17
2.2.9 Load balancer .....	18
2.2.10 Health monitor .....	19
2.2.11 Server pool .....	22
2.2.12 Virtual servers .....	23
2.3 Statistics monitoring .....	24
2.4 Deployment examples .....	25
2.4.1 Example: SSL offloading .....	25
2.4.2 Example: NFS through NSX-T .....	31
2.4.3 Example: Site failover in an ECS multi-site configuration .....	33
<b>3 Best practices .....</b>	<b>35</b>

- A Troubleshooting.....36
  - A.1 View access logs .....36
  - A.2 Packet captures .....36
- B Technical support and resources .....38
  - B.1 Related resources.....38
    - B.1.1 ECS product documentation.....38
    - B.1.2 VMware NSX-T load balancer documentation .....38

## Executive summary

The explosive growth of unstructured data and cloud-native applications has created demand for scalable cloud storage infrastructure in the modern data center. Dell EMC™ ECS is the third-generation object store platform from Dell EMC. ECS is designed from the ground up to deliver modern cloud storage API, distributed data protection, and active/active availability spanning multiple data centers.

Managing application traffic both locally and globally can provide high availability (HA), as well as efficient use of ECS clustered network, RAM, and CPU resources. HA is obtained by directing application traffic to known-to-be-available local or global storage resources. An IP load balancer is required when deploying ECS to ensure application connections are evenly distributed across local or remote data center ECS nodes.

## Objectives

This document is a reference guide for configuring the VMware® NSX-T load balancer with ECS. An external load balancer (traffic manager) is required with ECS for applications that do not proactively monitor ECS node availability or natively manage traffic load to ECS nodes. Directing application traffic to ECS nodes using local DNS queries, as opposed to a traffic manager, can lead to failed connection attempts to unavailable nodes and unevenly distributed application load on ECS.

The ECS HDFS client, CAS SDK and ECS S3 API extensions are outside of the scope of this paper. The ECS HDFS client, which is required for Hadoop connectivity to ECS, handles load balancing natively. Similarly, the Centera™ Software Development Kit for CAS access to ECS has a built-in load balancer. The ECS S3 API also has extensions leveraged by certain ECS S3 client SDKs which allow for balancing load to ECS at the application level. Furthermore, Dell EMC applications developed using the ECS S3 client SDKs like the Dell EMC CIFS-ECS Gateway and the Dell EMC ECS Streamer driver for Veritas Enterprise Vault® have optional native load balancing of connections to ECS.

Dell EMC takes no responsibility for customer load balancing configurations. All customer networks are unique, with their own requirements. It is extremely important for customers to configure their load balancers according to their own circumstance. We only provide this paper as a guide. VMware, Dell EMC Processional Services, or a qualified network administrator should be consulted before making any changes to your current load balancer configuration

## Audience

This document is intended for administrators who deploy and configure Dell EMC ECS with a load balancer. This guide assumes a high level of technical knowledge for the devices and technologies. It is highly recommended to review the [NSX-T Load Balancing documentation](#) as a prerequisite to this guide.

# 1 Solution overview

This section provides an overview of the integration components of Dell EMC ECS and the VMware NSX-T load balancer and the key technologies used.

## 1.1 ECS overview

ECS provides a complete software-defined, strongly-consistent, indexed, cloud-storage platform that supports the storage, manipulation, and analysis of unstructured data on a massive scale. Client access protocols include an S3-compatible API (with additional Dell EMC extensions for retention, byte range append/update/overwrite, and indexed metadata search), Dell EMC Atmos™, OpenStack Swift, and Dell EMC Centera Content Addressable Storage (CAS API, NFS, and HDFS). Object access for S3, Atmos, and Swift is achieved through REST APIs. Objects are written, retrieved, updated and deleted through HTTP or HTTPS calls using REST verbs such as GET, POST, PUT, DELETE, and HEAD. Atmos and S3 buckets can be configured for native file access using NFSv3 and Apache® Hadoop® Compatible File System (HCFS)

ECS was built as a completely distributed system following the principle of cloud applications. In this model, all hardware nodes provide the core storage services. Without dedicated index or metadata nodes, the system has limitless capacity and scalability.

Service communication ports are a key consideration when configuring VMware NSX-T to balance the load to the ECS nodes. See Table 1 below for a complete list of protocols used with ECS and their associated ports. In addition to managing traffic flow, port access and port re-mapping is a critical piece to consider when firewalls are in the communication path. For more information on ECS ports, refer to the ECS [Security Configuration Guide](#).

For a more thorough ECS overview, review the ECS [Overview and Architecture](#) white paper.

Table 1 ECS protocols and associated ports

Protocol	Transfer protocol or daemon service	Port
S3	HTTP	9020
	HTTPS	9021
Atmos	HTTP	9022
	HTTPS	9023
Swift	HTTP	9024
	HTTPS	9025
NFS	portmap	111
	mountd, nfsd	2049
	lockd	10000

## 1.2 ECS constructs

Understanding the main ECS constructs is necessary in managing application workflow and load balancing. This section details each of the upper-level ECS constructs.

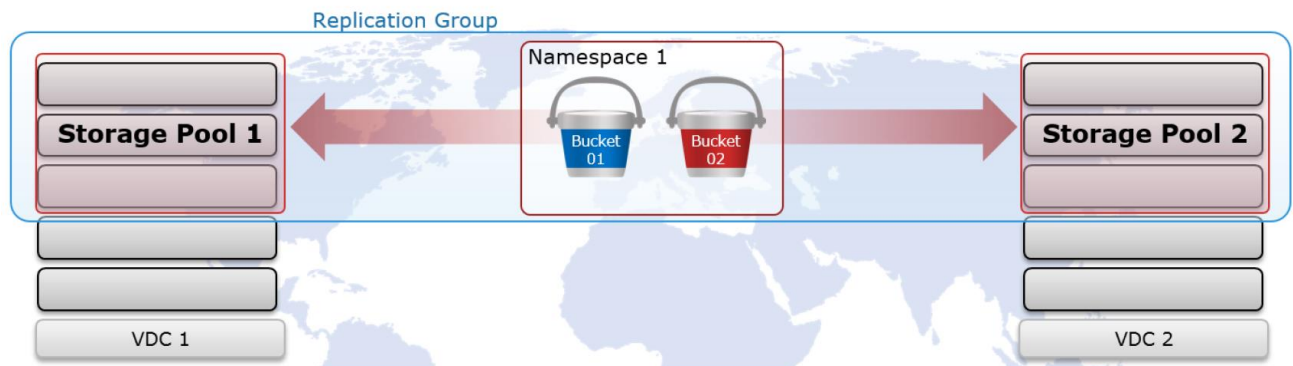


Figure 1 ECS upper-level constructs

**Storage pool:** The first step in provisioning a site is creating a storage pool. Storage pools form the basic building blocks of an ECS cluster. They are logical containers for some or all nodes at a site.

ECS storage pools identify which nodes will be used when storing object fragments for data protection at a site. Data protection at the storage pool level is rack, node, and drive aware. System metadata, user data and user metadata all coexist on the same disk infrastructure.

Storage pools provide a means to separate data on a cluster, if required. By using storage pools, organizations can organize storage resources based on business requirements. For example, if separation of data is required, storage can be partitioned into multiple different storage pools. Erasure coding (EC) is configured at the storage pool level. The two EC options on ECS are 12+4 or 10+2 (aka cold storage). EC configuration cannot be changed after storage pool creation.

Only one storage pool is required in a VDC. Generally, at most two storage pools should be created, one for each EC configuration, and only when necessary. Additional storage pools should only be implemented when there is a use case to do so, for example, to accommodate physical data separation requirements. This is because each storage pool has unique indexing requirements. As such, each storage pool adds overhead to the core ECS index structure.

A storage pool should have a minimum of five nodes and must have at least three or more nodes with more than 10% free space to allow writes.

**Virtual Data Center (VDC):** VDCs are the top-level ECS resources and are also generally referred to as a site or zone. They are logical constructs that represent the collection of ECS infrastructure you want to manage as a cohesive unit. A VDC is made up of one or more storage pools.

Between two and eight VDCs can be federated. Federation of VDCs centralizes and thereby simplifies many management tasks associated with administering ECS storage. In addition, federation of sites allows for expanded data protection domains that include separate locations.

**Replication group:** Replication groups are logical constructs that define where data is protected and accessed. Replication groups can be local or global. Local replication groups protect objects within the same VDC against disk or node failures. Global replication groups span two or more federated VDCs and protect objects against disk, node, and site failures.

The strategy for defining replication groups depends on multiple factors including requirements for data resiliency, the cost of storage, and physical versus logical separation of data. As with storage pools, the minimum number of replication groups required should be implemented. At the core ECS indexing level, each storage pool and replication group pairing is tracked and adds significant overhead. It is best practice to create the absolute minimum number of replication groups required. Generally, there is one replication group for each local VDC, if necessary, and one replication group that contains all sites. Deployments with more than two sites may consider additional replication groups, for example, in scenarios where only a subset of VDCs should participate in data replication, but, this decision should not be made lightly.

**Namespace:** Namespaces enable ECS to handle multi-tenant operations. Each tenant is defined by a namespace and a set of users who can store and access objects within that namespace. Namespaces can represent a department within an enterprise, can be created for each unique enterprise or business unit, or can be created for each user. There is no limit to the number of namespaces that can be created from a performance perspective. Time to manage an ECS deployment, on the other hand, or, management overhead, may be a concern in creating and managing many namespaces.

**Bucket:** Buckets are containers for object data. Each bucket is assigned to one replication group. Namespace users with the appropriate privileges can create buckets and objects within buckets for each object protocol using its API. Buckets can be configured to support NFS and HDFS. Within a namespace, it is possible to use buckets as a way of creating subtenants. For performance reasons, it is not recommended to have more than 1000 buckets per namespace. Generally, a bucket is created per application, workflow, or user.

## 1.3 Solution architecture

The NSX-T logical load balancer offers high-availability service for applications and distributes the network traffic load amongst multiple nodes in the VDC. Figure 2 below shows the NSX-T load balancer and ECS logical architecture.

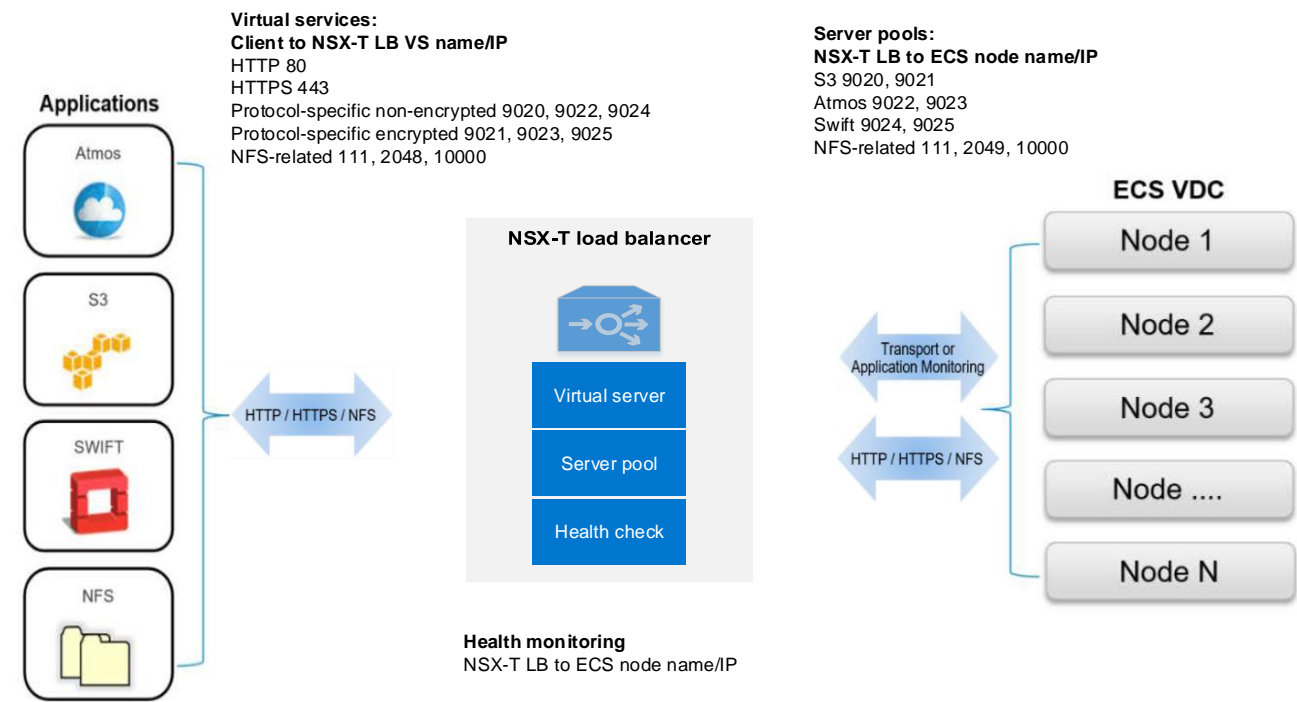


Figure 2 Logical architecture



The load balancer distributes incoming service requests evenly among multiple servers in such a way that the load distribution is transparent to users. Load balancing helps in achieving optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload.

You can map a virtual IP address to a set of pool servers for load balancing. The load balancer accepts TCP, UDP, HTTP, or HTTPS requests on the virtual service IP address and decides which pool server to use.

## 1.4 VMware NSX-T load balancer constructs

The Load balancer includes virtual servers, server pools, and health checks monitors.

A load balancer is connected to a Tier-1 logical router. The load balancer hosts single or multiple virtual servers. A virtual server is an abstract of an application service, represented by a unique combination of IP, port, and protocol. The virtual server is associated to single to multiple server pools. A server pool consists of a group of servers. The server pools include individual server pool members.

Figure 3 below shows the various constructs of an Edge Node with a Tier-1 logical router and NSX-T load balancer servicing a Dell EMC ECS VDC.

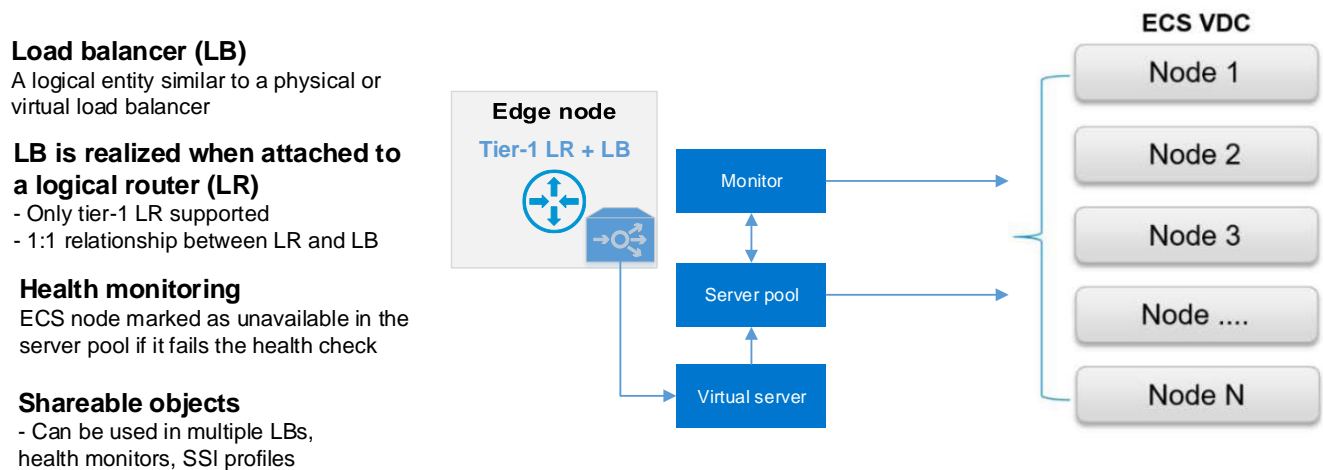


Figure 3 NSX-T load-balancer constructs

The NSX-T Data Center load balancer supports the following features:

- Layer 4: TCP and UDP
- Layer 7: HTTP and HTTPS with load balancer rules support
- Server pools: Static and dynamic with NSGroup
- Persistence: Source-IP and Cookie persistence mode
- Health check monitors: Active monitor which includes HTTP, HTTPS, TCP, UDP, and ICMP, and passive monitor
- Source network address translation (SNAT): Transparent, Automap, and IP List
- HTTP upgrade: For applications using HTTP upgrade such as WebSocket, the client or server requests for HTTP Upgrade, which is supported. By default, NSX-T Data Center supports and accepts HTTPS upgrade client request using the HTTP application profile.

To detect an inactive client or server communication, the load balancer uses the HTTP application profile response timeout feature set to 60 seconds. If the server does not send traffic during the 60 seconds interval, NSX-T Data Center ends the connection on the client and server side.

- **Load balancer:** Available in small, medium, and large sizes. Based on the load balancer size, the load balancer can host different virtual servers and pool members.
- **Virtual servers:** Virtual servers receive all the client connections and distribute them among the servers. A virtual server has an IP address, a port, and a protocol.
  - For Layer 4 virtual servers, lists of ports ranges can be specified instead of a single TCP or UDP port to support complex protocols with dynamic ports
  - Load balancer rules are supported for only Layer 7 virtual servers with an HTTP application profile. Different load balancer services can use load balancer rules.
- **Server pools:** A server pool consists of one or more servers that are configured and running the same application. A single pool can be associated to both Layer 4 and Layer 7 virtual servers
- **Application profile:** Application profiles define the behavior of a particular type of network traffic. The associated virtual server processes network traffic according to the values specified in the application profile. Fast TCP, Fast UDP, and HTTP application profiles are the supported types of profiles.
- **Persistent profile:** To ensure stability of stateful applications, load balancers implement persistence which directs all related connections to the same server. Different types of persistence are supported to address different types of application needs. For example, NFS requires a persistent connection.
- **Active health monitor:** The active health monitor is used to test whether a server is available. The active health monitor uses several types of tests such as sending a basic ping to servers or advanced HTTP requests to monitor application health
- **Passive health monitor:** Load balancers perform passive health checks to monitor failures during client connections and mark servers causing consistent failures as DOWN.
- **Services router (SR):** Services Router SR is used for centralized services such as Connectivity to the physical network, Load Balancing, NAT and Edge Firewall
- **Distributed router (DR):** Provides in kernel distributed routing on the hypervisor which prevents hair pinning and takes care of East-West traffic in the Data Center. Distributed Routing also provides routing closest to the source.
- **Logical switch:** reproduces switching functionality, broadcast, unknown unicast, multicast (BUM) traffic, in a virtual environment completely decoupled from the underlying hardware

## 1.5 VMware NSX-T load balancer deployment options

When the load balancer is deployed, a decision needs to be made whether to place it in one-armed mode or in-line mode.

In one-armed mode (most popular deployment for ECS and used in the detailed configuration below), the load balancer is deployed in parallel to the application servers and the client traffic is translated to ensure that the load balancer has access to all packets in the connection flow. In this mode, the load balancer must be configured with Source NAT (SNAT) to force server's response through the load balancer.

If client IP address visibility is need on the server (ECS), the **X-Forwarded-For** header injection option in the load balancer HTTP application profile is available. With this option the client IP address is added in the X-Forwarded-For HTTP header in all requests sent to the server (ECS).

---

**Note:** X-forwarded-For would also need to be enabled on the ECS nodes, this change requires contacting Dell EMC Support.

---

Figure 4 below shows the one-armed mode with a standalone Tier-1 logical router that has a load balancer attached to it using a single VLAN segment attached to ECS.

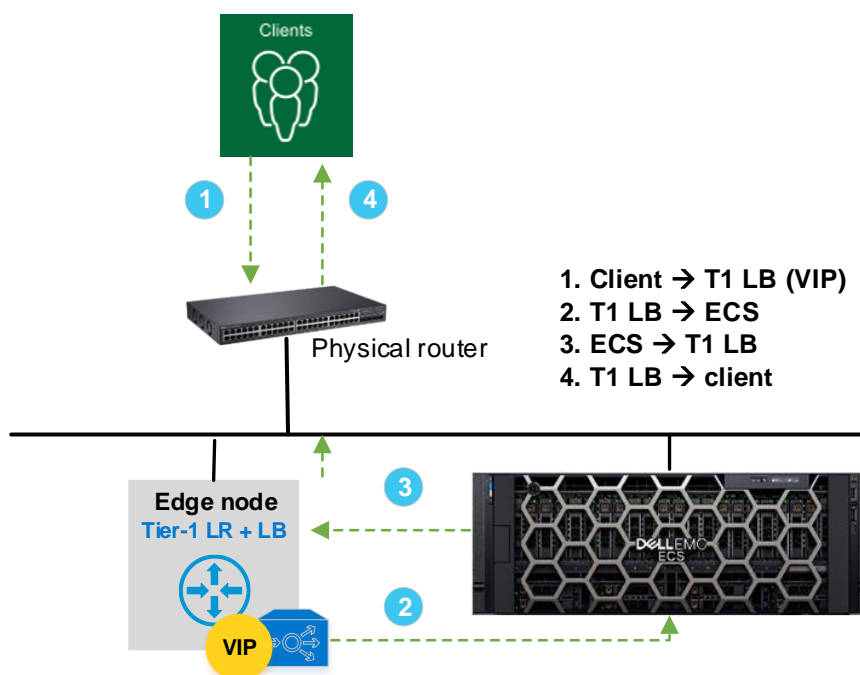


Figure 4 One-arm deployment

In an in-line deployment, the load balancer is also the default gateway for backend servers or is “naturally” inline between the backend servers and the clients via the network topology.

In this deployment, there is no need for the load balancer to do SNAT nor X-Forwarded-for header injection since the server's response will always be sent through the load balancer.

The below figure shows an in-line deployment with a physical router connected to tier-0 logical router and a tier-1 logical router with load balancing service that is connected to a VLAN (through a tier-1 service interface) to ECS.

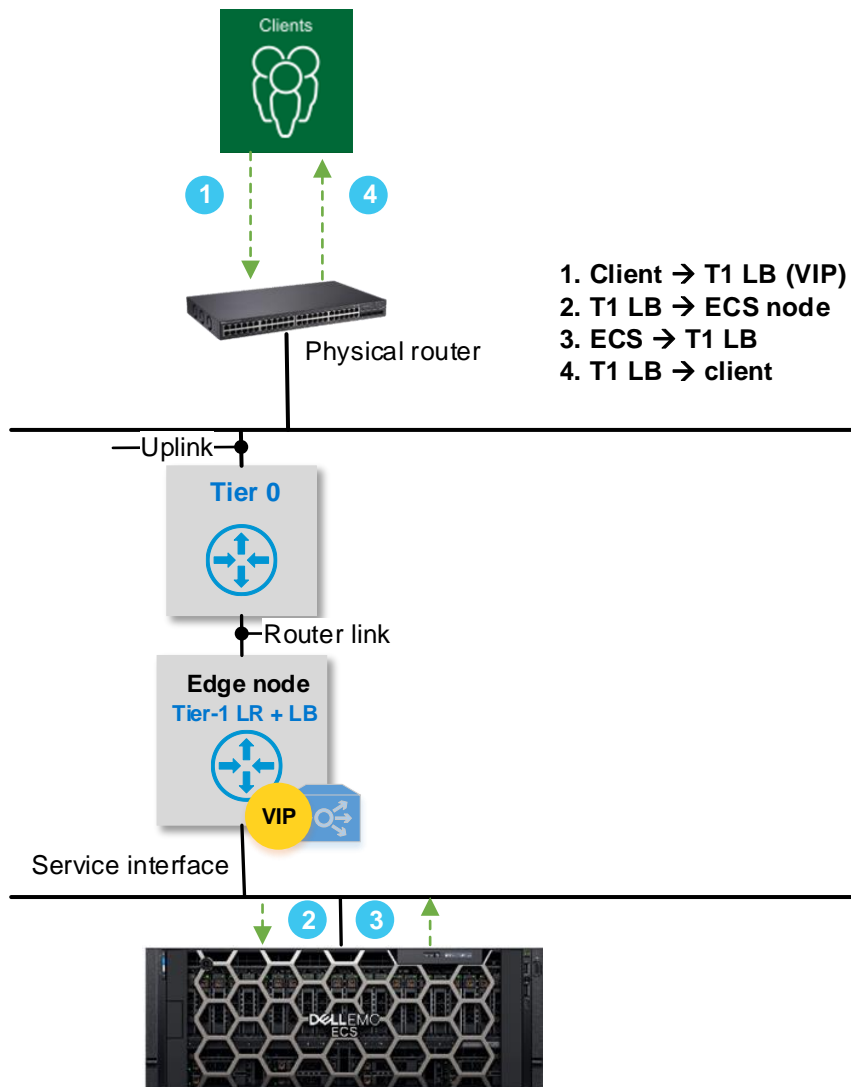


Figure 5 Inline deployment

## 1.6 Key components

The following components and versions were used to validate this solution.

Table 2 Dell EMC components

Component	Description
ECS appliance	EX300 (5 Nodes)
ECS version	3.3.0

Table 3 VMware components

Component	Description
ESXi™	6.7.0
vSphere®	6.7.0.20000
NSX Unified Appliance	2.4.0.0.0.12456291
NSX Edge	2.4.0.0.0.12454265

## 2 Solution implementation

This section describes the high-level steps required to deploy and configure a VMware NSX-T load balancer with ECS.

### 2.1 Implementation workflow

The below are the minimal steps required to implement the solution. Note that the NSX-T components were configured solely for load balancing traffic to and from the ECS VDCs.

<b>Step 1: Deploy and Configure the NSX Unified Appliance</b> NSX-T Manager and Controller
<b>Step 2: Configure a Compute Manager</b> Manages resources such as hosts and VMs
<b>Step 3: Add the Transport Node</b> Controls which hosts a logical switch can reach
<b>Step 4: Add Transport Zones</b> Endpoint which participates the NSX-T data plane
<b>Step 5: Add an IP Pool</b> Tunnel end points (TEPs). Used to identify the transport node.
<b>Step 6: Deploy and Configure an Edge VM</b> Provides routing services and connectivity to external networks
<b>Step 7: Add a Segment</b> Connects north/south bound direction to the gateway
<b>Step 7: Configure the Tier-1 gateway</b> Configure service interfaces and routes
<b>Step 7: Create the Load Balancer</b> A Load balancer is attached to a tier-1 gateway
<b>Step 10: Create the Health Monitor</b> Monitor ECS node status
<b>Step 11: Create the Server Pool</b> Add each node in the VDC
<b>Step 12: Create the Virtual Server</b> Setup layer 4 or 7 virtual services

Figure 6 Implementation steps

### 2.2 Installation and configuration steps

The following installation and configuration steps are meant to be a guide for configuring an NSX-T Logical Load Balancer with ECS. Note that the installation and configuration of NSX-T Data Center appliance and associated network configuration is beyond the scope of this document. The VMware NSX-T [documentation](#) should be referenced for more detail.

#### 2.2.1 NSX Unified Appliance

NSX-T Data Center network virtualization programmatically creates, deletes, and restores software-based virtual networks. With network virtualization, the functional equivalent of a network hypervisor reproduces the complete set of Layer 2 through Layer 7 networking services (for example, switching, routing, access control, firewalling, QoS) in software.

NSX-T Data Center 2.4 combines the role of the nsx-manager appliance and the nsx-controller into a single appliance. The NSX manager can be installed in various sizes depending on the deployment requirements.

Table 4 NSX manager VM resource requirements

Appliance Size	Memory	vCPU	Disk Space	VM Hardware Version
NSX Manager Extra Small	8 GB	2	200 GB	10 or later
NSX Manager Small VM	16 GB	4	200 GB	10 or later
NSX Manager Medium VM	24 GB	6	200 GB	10 or later
NSX Manager Large VM	48 GB	12	200 GB	10 or later

Reference the [NSX-T Data Center Installation guide](#) for detailed installation steps.

## 2.2.2 Compute manager

A compute manager, for example, vCenter Server, is an application that manages resources such as hosts and VMs. NSX-T Data Center polls compute managers to find out about changes.

## 2.2.3 Transport node

A Transport Node is defined as an endpoint which participates in the NSX-T data plane. A transport node has two types. Edge Transport Node (NSX Edges) and Host Transport Node (ESX Hosts).

NSX-Edge

Overview

Monitor

Related ▾

▼ Summary

EDIT

Name

NSX-Edge

ID

dc3dcef4-e441-411e-9a43-9f54d83c5bf0

Location

Description

External ID

dc3dcef4-e441-411e-9a43-9f54d83c5bf0

Configuration State

● Success

Deployment Type

Virtual Machine

Management IP

10.246.25.210

Host

NSX Version

2.4.0.0.0.12454265

Controller Connectivity

● Up

Manager Connectivity

● Up

Transport Zones

[TZ-VLAN](#)  
[TZ-Site-A](#)

Edge Cluster

Edge-Cluster

Logical Routers

1

> Tags

MANAGE

Figure 7 Edge transport node

## 2.2.4 Transport zone

Transport zones dictate which hosts can participate in the use of a network. A transport zone does this by limiting the hosts that can "see" a logical switch.

A logical switch can be of two types. VLAN Logical Switch or Overlay Logical Switch. In this guide only a VLAN Logical switch is used. When a transport zone is configured, it must be defined with an N-VDS name. N-VDS is the new type of VMware Virtual Switch that is introduced with NSX-T. Hence when a transport node is configured as part of a transport zone an N-VDS will be automatically provisioned on that transport node. An NSX Edge will be configured in the next step as an Edge Transport node and it will be attached to the respective Transport Zone.

Name	TZ-VLAN
ID	4795edb9-37f8-473d-8a09-8fc6723b0986
Location	
Description	
Traffic Type	VLAN
N-VDS Name	N-VDS
Host Membership Criteria	Standard
Uplink Teaming Policy Names	
Logical Ports	1
Logical Switches	2

Figure 8 VLAN logical switch

## 2.2.5 Tunnel endpoints

Tunnel endpoints are the source and destination IP addresses used in the external IP header to identify the clients originating and end the NSX-T Data Center encapsulation of overlay frames. Note that you can create an IP pool to use for the tunnel endpoints.

---

**Note:** At least 1 TEP is currently required for now even if you're not using an Overlay.

---

## 2.2.6 NSX edge

The NSX Edge Transport Node provides routing services and connectivity to network NSX Edges that are external to the NSX-T Data Center deployment.

An NSX Edge is required if you want to deploy a tier-0 or tier-1 logical router with stateful services such as network address translation (NAT), VPN, and so on.

---

**Note:** An NSX Edge must belong to an NSX Edge Cluster even if you have only one NSX Edge deployed.

---

## 2.2.7 Segments

A segment, also known as a logical switch, provides virtual Layer 2 switching for VM and Gateway interfaces. A segment gives tenant network administrators the logical equivalent of a physical Layer 2 switch.



Segment Name	Uplink & Type	Subnets	Status
LS-VLAN25	None - Flexible		Up
L2 VPN	Transport Zone	TZ-VLAN   VLAN	
VPN Tunnel ID	VLAN	25	
Domain Name	IP Address Pool		
Tags	0		
<b>PORTS</b>			
Segment Ports	0		
<b>SEGMENT PROFILES</b>			
IP Discovery	default-ip-discovery-profile	Spoof Guard	default-spoofguard-profile
MAC Discovery	default-mac-discovery-profile	Segment Security	default-segment-security-profile
QoS	-		
<a href="#">ADVANCED CONFIGURATION</a>			

Figure 9 Logical switch segment

## 2.2.8 Tier-1 gateway

A tier-1 gateway performs the functions of a tier-1 logical router. It has downlink connections to segments.

Since we are only load balancing ECS, we created a standalone tier-1 logical router which has no downlink and no connection to a tier-0 router. It has a service router but no distributed router. A Service Router is responsible for delivering services that are not currently implemented in a distributed fashion, such as stateful NAT. The service router is deployed on an NSX Edge node.

The below is an example of the Tier-1 Standalone Gateway associated with a service interface (LS-VLAN25 logical switch) and a static route to our VLAN gateway.

Tier-1 Gateway Name	Linked Tier-0 Gateway	#Linked Segments	Status
T1-Standalone	Select Tier-0 Gateway		
Fail Over	Non Preemptive	IP Address Management	No IP Allocation Set
Edge Cluster	Edge-Cluster	Edges	Auto Allocated Set
Tags	Tag (Required) Scope (Optional)		
Route Advertisement			
<b>SERVICE INTERFACES</b>			
Service Interfaces	1		
<b>STATIC ROUTES</b>			
Static Routes	1		
<a href="#">ADVANCED CONFIGURATION</a>			

Figure 10 Tier-1 gateway

Interfaces

Tier-1 GatewayT1-Standalone#Interfaces 1

ADVANCED CONFIGURATION

EXPAND ALL

Q Search

	Name	IP Address / Mask	Connected To(Segment)	Tags	Status
:	LS-VLAN25	10.246.25.213/24	LS-VLAN25	0	Up

Figure 11 Tier-1 gateway service interface

Static Routes

Tier-1 GatewayT1-Standalone#Static Routes 1

ADVANCED CONFIGURATION

Q Search

Name	Network	Next Hops
default	0.0.0.0/0	1

Next Hops

Tier-1 GatewayT1-StandaloneStatic Route default#Next Hops 1

Q Search

IP Address	Admin Distance	Interfaces
10.246.25.1	1	None

Figure 12 Tier-1 gateway static route

2.2.9 Load balancer

The load balancer distributes incoming service requests evenly among multiple servers in such a way that the load distribution is transparent to users. Load balancing helps in achieving optimal resource utilization, maximizing throughput, minimizing response time, and avoiding overload.

The below is an example of a Load Balancer configuration that is associated with a tier-1 standalone gateway.

Name	Size	Tier-1 Gateway	Virtual Servers	Status
ECS-LB	Small	T1-Standalone		
Description	Dell EMC ECS Load Balancer		Error Log Level	Info
Tags	Tag (Required) Scope (Optional)		Admin State	Enabled

Figure 13 Logical load balancer

## 2.2.10 Health monitor

With ECS, use of the S3 Ping operation is recommended in monitoring the ECS S3 service port. This operation is documented inside the Dell EMC ECS [REST API Reference Guide](#).

The S3 Ping operation is dependent upon the fabric layer inside the ECS software. The fabric layer of the ECS software stack provides clustering and system health among other things. It is responsible for keeping required services up and running and managing resources such as disks, containers, and the network. It tracks and reacts to environmental changes such as failure detection and provides alerts related to system health. The S3 Ping operation uses the fabric layer to determine the state of the node's maintenance mode.

Several different health check types are available however the two types used for the ECS S3 Ping check method are the HTTP and HTTPS protocols.

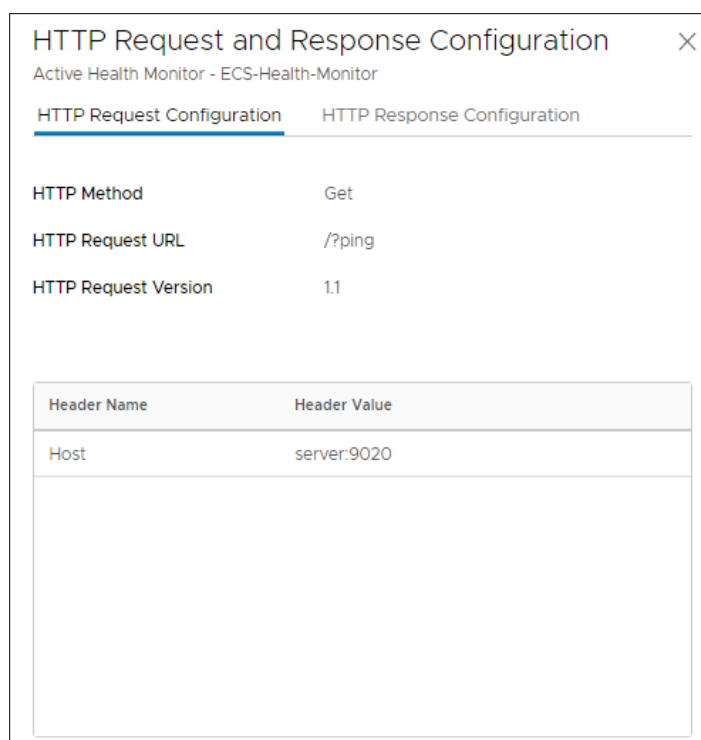
To create an ECS active health monitor, navigate to **Networking > Load Balancing > Monitors > Active > Add Active Monitor** and select either HTTP or HTTPS.

- **Name and Description:** Enter a name and description for the active health monitor.
- **Monitoring Port:** Set port to monitor. For example. To check S3 status using HTTP use port 9020.
- **Monitoring interval:** Time in seconds that the monitor sends another connection request to the server.
- **Timeout period:** Set the number of times the server is tested before it is considered as DOWN.
- **Fail count:** When consecutive failures reach this value, the server is considered temporarily unavailable.
- **Rise count:** Time the server is tried again to check if it is available.
- **Tags (optional):** Enter tags to make searching easier.

Click the **Configure** link under the Additional Properties section to configure the request and response attributes. Note that the following example is using HTTP.

### HTTP Request Configuration tab

- **HTTP Method:** Get
- **HTTP Request URL:** /?ping
- **HTTP Request Version:** 1.1



The screenshot shows a dialog box titled "HTTP Request and Response Configuration" with a close button (X) in the top right corner. Below the title bar, it says "Active Health Monitor - ECS-Health-Monitor". There are two tabs: "HTTP Request Configuration" (which is selected and underlined) and "HTTP Response Configuration". Under the "HTTP Request Configuration" tab, there are three fields: "HTTP Method" with the value "Get", "HTTP Request URL" with the value "/?ping", and "HTTP Request Version" with the value "1.1". Below these fields is a table with two columns: "Header Name" and "Header Value". The table contains one row with "Host" as the header name and "server:9020" as the header value.

Header Name	Header Value
Host	server:9020

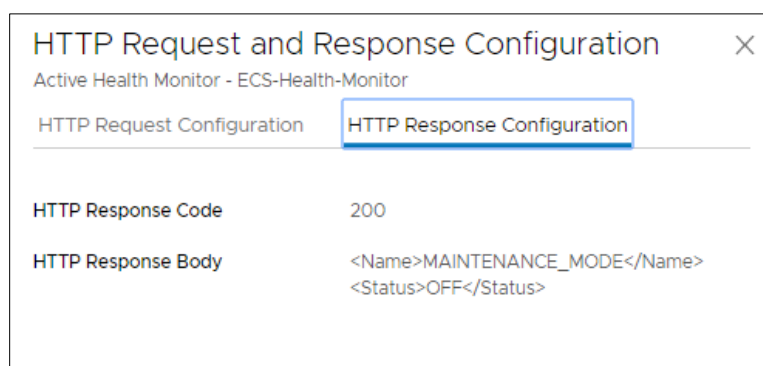
Figure 14 ECS health monitor request configuration

Click the **Add** button to add a header name and value.

- **Header Name:** Host
- **Header Value:** The ECS service port being monitored using the format 'server:port'. i.e. server:9020

#### HTTP Response Configuration tab

- **HTTP Response Code:** 200
- **HTTP Response Body:** <Name>MAINTENANCE\_MODE</Name><Status>OFF</Status>



The screenshot shows the same dialog box as Figure 14, but with the "HTTP Response Configuration" tab selected and highlighted with a blue border. The "HTTP Request Configuration" tab is now greyed out. Under the "HTTP Response Configuration" tab, there are two fields: "HTTP Response Code" with the value "200" and "HTTP Response Body" with the value "<Name>MAINTENANCE\_MODE</Name><Status>OFF</Status>".

Figure 15 ECS health monitor response configuration

If the server pool receives a reply pattern from any of the ECS nodes that do not match the response code, then the node will be marked as disabled and automatically re-enabled once the pattern matches.

The screenshot shows the configuration page for an ECS Health Monitor. At the top, there is a table with columns: Name, Protocol, Monitoring Port, Monitoring Interval, Timeout Period (sec), and Server Pools. The first row shows 'ECS-Health-Monitor' with a red asterisk, Protocol 'HTTP', Monitoring Port '9020', Monitoring Interval '5', and Timeout Period '15'. Below the table, there is a 'Description' field with a placeholder 'Enter Description'. To the right, there are 'Fail Count' and 'Rise Count' fields, both set to '3'. Below these, there are 'Tag (Required)' and 'Scope (Optional)' fields, with a note 'Maximum 30 tags are allowed.' and a checkmark icon. Under 'Additional Properties', there are 'HTTP Request' and 'HTTP Response' sections, each with a 'Configure' link. At the bottom, there are 'SAVE' and 'CANCEL' buttons.

Figure 16 ECS health monitor

The screenshot shows a dialog box titled 'HTTP Request and Response Configuration' with a close button (X). Below the title, it says 'Active Health Monitor - ECS-Health-Monitor'. There are two tabs: 'HTTP Request Configuration' and 'HTTP Response Configuration', with the latter being selected. The 'HTTP Response Configuration' section contains two fields: 'HTTP Response Code' set to '200' and 'HTTP Response Body' set to '<Name>MAINTENANCE\_MODE</Name>' and '<Status>OFF</Status>'. The dialog box has a light gray background and a white border.

Figure 17 ECS health monitor response configuration

## 2.2.11 Server pool

The server pool consists of the nodes in the ECS VDC and supports various algorithm balancing methods and source NAT (SNAT) translation modes.

A server pool can be created using the following balancing methods and SNAT translation modes.

Option	Description
ROUND_ROBIN	Incoming client requests are cycled through a list of available servers capable of handling the request. Ignores the server pool member weights even if they are configured.
WEIGHTED_ROUND_ROBIN	Each server is assigned a weight value that signifies how that server performs relative to other servers in the pool. The value determines how many client requests are sent to a server compared to other servers in the pool. This load balancing algorithm focuses on fairly distributing the load among the available server resources.
LEAST_CONNECTION	Distributes client requests to multiple servers based on the number of connections already on the server. New connections are sent to the server with the fewest connections. Ignores the server pool member weights even if they are configured.
WEIGHTED_LEAST_CONNECTION	Each server is assigned a weight value that signifies how that server performs relative to other servers in the pool. The value determines how many client requests are sent to a server compared to other servers in the pool. This load balancing algorithm focuses on using the weight value to distribute the load among the available server resources. By default, the weight value is 1 if the value is not configured and slow start is enabled.
IP-HASH	Selects a server based on a hash of the source IP address and the total weight of all the running servers.

Mode	Description
Auto Map Mode	Load Balancer uses the interface IP address and ephemeral port to continue the communication with a client initially connected to one of the server's established listening ports. SNAT is required. Enable port overloading to allow the same SNAT IP and port to be used for multiple connections if the tuple (source IP, source port, destination IP, destination port, and IP protocol) is unique after the SNAT process is performed. You can also set the port overload factor to allow the maximum number of times a port can be used simultaneously for multiple connections.
Disable	Disable the SNAT translation mode.
IP Pool	Specify a single IP address range, for example, 1.1.1-1.1.1.10 to be used for SNAT while connecting to any of the servers in the pool. By default, from 4000 through 64000-port range is used for all configured SNAT IP addresses. Port ranges from 1000 through 4000 are reserved for purposes such as, health checks and connections initiated from Linux applications. If multiple IP addresses are present, then they are selected in a Round Robin manner. Enable port overloading to allow the same SNAT IP and port to be used for multiple connections if the tuple (source IP, source port, destination IP, destination port, and IP protocol) is unique after the SNAT process is performed. You can also set the port overload factor to allow the maximum number of times a port can be used simultaneously for multiple connections.

Figure 18 Algorithm balancing methods and SNAT modes

To create a server pool, navigate to **Networking > Load Balancing > Server Pools > Add Server Pool**

In the below example, we've added a 5 node EX300 to our server pool using the 'least connections' balancing method and the Automap source NAT. We've also selected to use a custom active health monitor to monitor the ECS S3 service port.

Name	Algorithm	Members/Group	Virtual Servers	Status
ECS-EX300-Pool *	Least Conn ▾	5		
Description		Active Monitor		
<input type="text" value="Enter Description"/>		ECS-Health-Monitor ⓧ ▾		
SNAT Translation Mode				
<input type="text" value="Automap"/>				
> Additional Properties <div> <input type="button" value="SAVE"/> <input type="button" value="CANCEL"/> </div>				

Figure 19 Server pool

Click the **Select Members**'link under the members/Group column to add each node in the ECS VDC to the pool.

**Note:** You can either enter each node in the ECS VDC manually or select a pre-configured group of server pool members.

Configure Server Pool Members

Server Pool - ECS-EX300-Pool

☒ Enter individual members

☐ Select a group

ADD MEMBER

Search

Name	IP	Port	Weight	State	Backup Member	Max Concurrent Connections
⋮ node5	10.246.25.185	9020	1	Enabled	● Disabled	
⋮ node4	10.246.22.184	9020	1	Enabled	● Disabled	
⋮ node3	10.246.25.183	9020	1	Enabled	● Disabled	
⋮ node2	10.246.22.182	9020	1	Enabled	● Disabled	
⋮ node1	10.246.22.181	9020	1	Enabled	● Disabled	

Figure 20 Server pool members

Once changes are applied, server pool status should show that the pool is up

	Name	Algorithm	Members/Group	Virtual Servers	Status
⋮ > ⚙	ECS-EX300-Pool	Least Connection	5	1	● Up ↻

Figure 21 Server pool status

2.2.12 Virtual servers

NSX-T supports both Layer 4 and Layer 7 virtual servers and includes several virtual server components such as application profiles, persistent profiles, and load balancer rules.

For this example, we chose to use Layer 4 and the 'default-tcp-lb-app-profile (FAST TCP) application profile. We recommend using source IP as the persistence profile because it keeps a client's application traffic pinned to the same pool member and allows for efficient use of the ECS cache. We generally recommend source IP as the persistence profile for all virtual servers in use with ECS. Source address affinity persistence directs session requests to the same server based solely on the source IP address of a packet.

As with all configuration choices, be sure to understand the options available to make appropriate traffic management decisions for each workflow and architecture deployed.

This example also utilizes the friendly port 80 on the front-end and port forwards requests to S3 HTTP service port 9020.

Name	IP Address	Ports	Type	Load Balancer	Server Pool	Status
EX300	10.246.25.216	80	L4 TCP	ECS-LB	ECS-EX300-P	

Description: Dell EMC ECS EX300 V5

Application Profile: default-tcp-lb-app-profile

Persistence: Source IP

Source IP: default-source-ip-lb-persistence-profile

Additional Properties:

- Max Concurrent Connections: Unlimited
- Max New Connection Rate: Unlimited
- Default Pool Member Ports: Enter Ports or Port Ranges (e.g. 8080, 80-90, 443)

Admin State: ☒ Enabled

Tags: Tag (Required) Scope (Optional)

Buttons: SAVE, CANCEL

Figure 22 Virtual server

## 2.3 Statistics monitoring

Statistics for the pool members can be displayed by navigating to **Advanced Networking & Security > Load Balancing > Server Pools**. Click the pool name and click the **Pool Member Statistics** tab.

Overview Virtual Servers Pool Members <u>Pool Member Statistics</u>						
Display Statistics from Load Balancer		ECS-LB				
IP:Port	Status	Current Sessions	Max Sessions	Bytes in	Bytes out	Http Request Rate
10.246.22.182:9020	↑ UP	0	1	1264	1274	0
10.246.22.181:9020	↑ UP	0	1	1678	11484	0
10.246.22.185:9020	↑ UP	32	57	4676690	1615501719	0
10.246.22.184:9020	↑ UP	0	1	14575	5395932	0
10.246.22.183:9020	↑ UP	0	0	0	0	0

Figure 23 Pool member statistics

Virtual Server connection and throughput details can be displayed by navigating to **Advanced Networking & Security > Load Balancing > Virtual Servers**. Click the virtual server name and go to the Statistics tab.



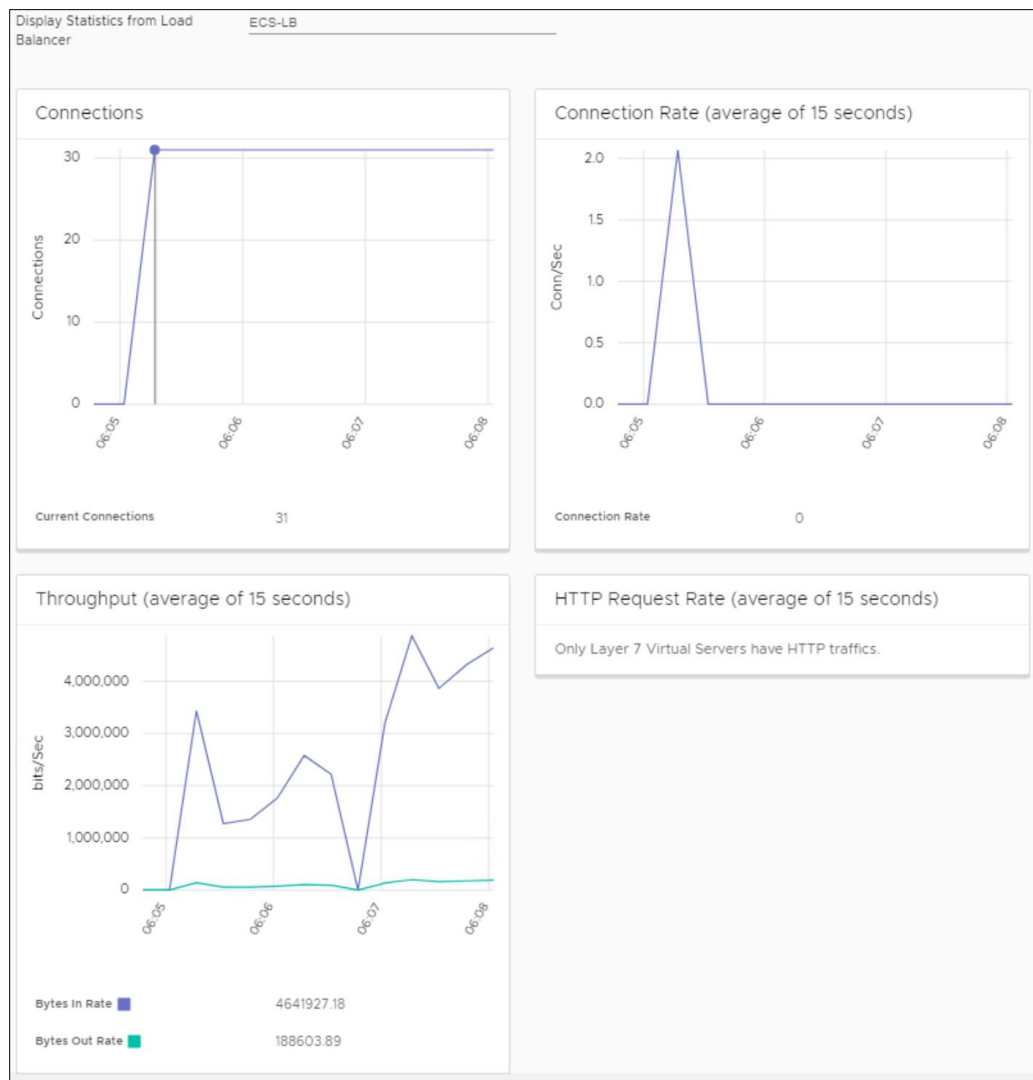


Figure 24 Virtual server statistics

## 2.4 Deployment examples

### 2.4.1 Example: SSL offloading

The primary configuration options available for encrypting client traffic to ECS are:

- ECS terminated SSL connectivity. End-to-end traffic encryption between client and ECS.
- NSX-T terminated SSL connectivity. Encrypted traffic between the client and the NSX-T Load Balancer. No encryption between NSX-T and ECS.
- ECS and NSX-T terminated SSL connectivity. Traffic is encrypted twice, first between client and NSX-T, and second between NSX-T and ECS.

SSL termination is a CPU intensive task, so it is recommended, when appropriate, to terminate SSL and offload encryption processing overhead off ECS. Each workflow should be assessed to determine if traffic requires encryption at any point in the communication path. Generally, administrators will use an SSL certificated signed by a trusted Certificate Authority, especially in a production environment. However, for this example we'll be using a self-signed certificate.

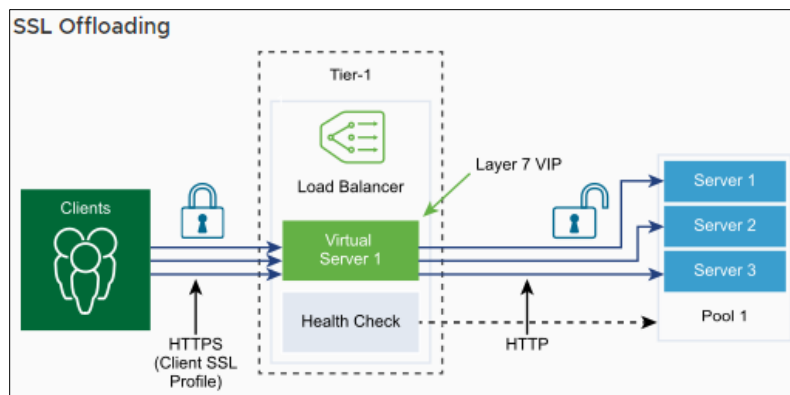


Figure 25 SSL offloading

The steps for this example include the following:

1. Create SSL key and self-signed certificate
2. Upload the certificate to NSX-T
3. Create the client SSL profile
4. Create a virtual server for NSX-T terminated SSL connectivity to ECS
5. Validate client connectivity

#### Step 1: Create the SSL key and self-signed certificate

OpenSSL is an open source implementation of the SSL and TLS protocols which we will be using to create the private SSL key and certificate. There's no lack of information on how to use this tool to generate a self-signed certificate so this document will only outline the general steps.

A private key is required for self-signed and CA requests certificates. An example of how to generate the private key is shown in the below figure. Permissions are modified to safeguard from accidental modification or deletion.

#### # openssl genrsa -des3 -out certificate.pem 2048

Generating RSA private key, 2048 bit long modulus

.....+++

.....+++

e is 65537 (0x10001)

Enter pass phrase for certificate.pem:

Verifying - Enter pass phrase for certificate.pem:

#### # chmod 0400 certificate.pem

OpenSSL does not allow passing of Subject Alternate Names (SANs) through the command line, so a configuration file can be created to define them. OpenSSL provides a sample configuration file (openssl.cnf) that can be used as a template. For our self-signed certificate, we've adding the IP and DNS of the SAN:

DNS.1 = nsx-ecs.richp.local

IP.1 = 10.246.25.213

The command to create the self-signed certificate is shown below. Note that we set the Common Name to "\*.nsx-ecs.richp.local" to support a wildcard DNS entry.

```
# openssl req -x509 -new -key certificate.pem -config openssl.cnf -days 365 -out ecs-ssl.crt
```

The x509 option tells req to create a self-signed certificate, the -days option specifies that the certificate will be valid for 365 days.

Verify that the SANs and CN are correct by using the below command:

```
# openssl x509 -in ecs-ssl.crt -noout -text
```

## Step 2: Upload the certificate to NSX-T

From the NSX-T management UI, navigate to **System > Certificates**, click the **Import** drop-down and select **Import Certificate**.

Figure 26 Import an SSL certificate

- **Name:** Give the SSL certificate a unique name that identifies the certificate
- **Certificate Contents:** Paste the contents or browse to the self-signed certificate file.
- **Private Key:** Paste the contents or browse to the private key file.
- **Passphrase:** Enter the passphrase used to create the certificate
- **Service Certificate:** This defaults to Yes and signifies that the certificate will be used with a load balancer

Click the **Import** button.

<input type="checkbox"/>	Certificate <span>↑</span>	ID	Issued To	Issued By	Validity	Type
<input type="checkbox"/>	ECS-SSL-Certificate	ECS-...cate	*.nsx-ecs.richp.local	*.nsx-ecs.richp.local	5/22/2019 - 5/21/2020	Certificate

Figure 27 SSL certificate details

**Step 3: Create the client SSL profile**

From the NSX-T management UI, navigate to **Advanced Networking & Security > Load Balancers > Profiles** and select the **'SSL profiles'** tab.

Click the **'ADD'** dropdown and select **Client Side SSL**.

**General tab**

- **Name:** Enter a descriptive name for the profile.
- **SSL Ciphers:** Assign the SSL ciphers to be included in the Client SSL profile .

Figure 28 Client SSL profile general information

**Protocols and Sessions tab**

- **Supported SSL Protocols:** Select the SSL protocols to be included in the Client SSL profile. Note that protocol versions TLS1.1 and TLS1.2 are enabled by default.

---

**Note:** SSL protocol versions TLS1.1 and TLS1.2 are enabled by default. As of ECS version 3.1 and later, only TLS1.2 is supported. If the application requires an earlier version of TLS, contact Dell EMC Support.

---

- **Session Caching:** SSL session caching allows the SSL client and server to reuse previously negotiated security parameters avoiding the expensive public key operation during an SSL handshake
- **Session Cache Entry timeout:** Enter the cache timeout in seconds to specify how long the SSL session parameters must be kept and can be reused.
- **Prefer Server Cipher:** Toggle the button so that the server can select the first supported cipher from the list it can support. During an SSL handshake, the client sends an ordered list of supported ciphers to the server.

Add Client Profile

General

Protocols and Sessions

Supported SSL Protocols

Available(3)

Search

SSL\_V3

TLS\_V1

TLS\_V1\_2

Selected(1)

Search

TLS\_V1\_2

Session Caching

Enabled

Session Cache Entry Timeout (seconds)

300

Prefer Server Cipher

Enabled

CANCEL

OK

Figure 29 Client SSL profile supported protocols

Step 4. Create a virtual server for NSX-T terminated SSL connectivity to ECS

Navigate to **Networking > Load Balancing > Virtual Servers** and add a new L7 HTTP virtual server. Enter the virtual server information specifying 443 as the port, select the load balancer and pool to use.

Name	IP Address	Ports	Type	Load Balancer	Server Pool	Status
EX300-SSL-	10.246.25.213	443	L7 HTTP	ECS-LB	ECS-EX300-Pt	
Description		DellEMC ECS SSL offloading		Application Profile		default-http-lb-app-profile
Persistence		Disabled		SSL Configuration		Configure
<div>Load Balancer Rules</div> <div>Additional Properties</div>						
<div>SAVE</div> <div>CANCEL</div>						

Figure 30 NSX-T Terminated SSL connection to ECS

Click the **Configuration** link next to the **SSL Configuration** tag and select the certificate and Client SSL Profile which were previously created and click the **Save**.

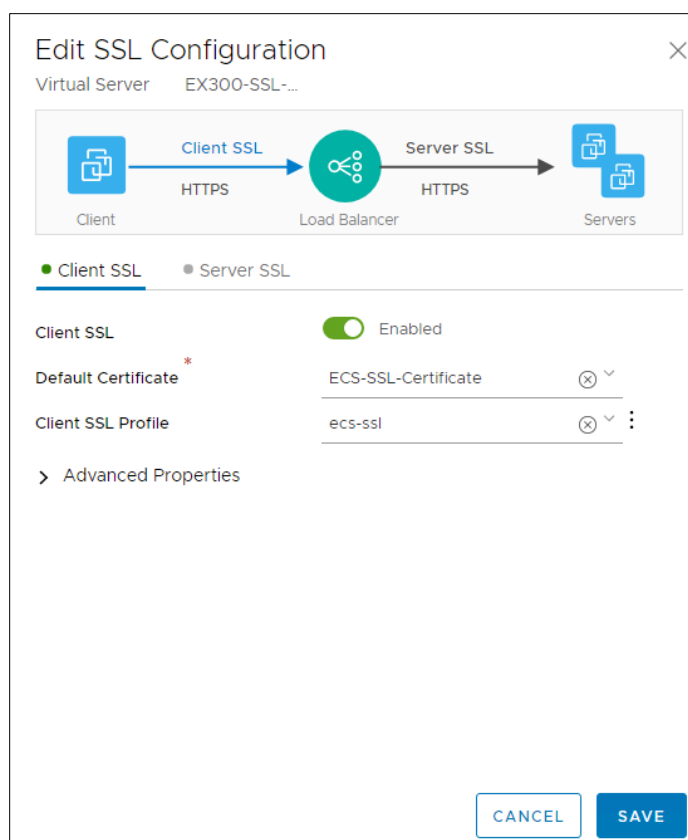


Figure 31 Virtual server SSL configuration

**Step 5:** Verify client connectivity

Use S3curl to list buckets of an ECS object user.

```
#s3curl --id=ecsid -- -ks https://nsx-ecs.richp.local | xmllint --format -
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>ecs-user</ID>
    <DisplayName>ecs-user</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>mybucket</Name>
      <CreationDate>2019-05-22T20:49:34.146Z</CreationDate>
      <ServerSideEncryptionEnabled>false</ServerSideEncryptionEnabled>
    </Bucket>
  </Buckets>
  <IsTruncated>false</IsTruncated>
</ListAllMyBucketsResult>
```

Upload a file to mybucket and list the contents of the bucket to show that file was uploaded.

```
#s3curl --id=ecsid --put=myfile.txt -- -ks https://nsx-ecs.richp.local/mybucket/myfile.txt
#s3curl --id=ecsid -- -ks https://nsx-ecs.richp.local/mybucket | xmllint --format -
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>mybucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>>false</IsTruncated>
  <ServerSideEncryptionEnabled>>false</ServerSideEncryptionEnabled>
  <Contents>
    <Key>myfile.txt</Key>
    <LastModified>2019-05-22T20:52:22.262Z</LastModified>
    <ETag>"0e3a1a927a256411b974d0aab932f473"</ETag>
    <Size>15</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>ecs-user</ID>
      <DisplayName>ecs-user</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

## 2.4.2 Example: NFS through NSX-T

ECS includes native file support with NFSv3 and is tightly integrated the other NFS server implementations, such as lockmgr, statd, nfsd, and mountd.

It is not recommended to balance NFS traffic load across ECS nodes without using persistence. This is because ECS nodes locally cache specific metadata attributes that NFS clients often query and read ahead (prefetch) NFS file data. These caching mechanisms allow fewer trips to disk which reduces system response time and generally improve sequential read throughput. Load balancing each client's NFS traffic severely reduces the benefits of these ECS cache mechanisms.

A client application should be tied to a single ECS node for the duration of the session. Only during a failure should the connection between client and ECS be moved to another ECS node.

The steps for this example are as follows:

1. Create the NFS server pool
2. Create the NFS related persistence profile
3. Create the NFS virtual server
4. Health monitoring

### Step 1: Create the NFS server pool

One pool is created for NFS traffic in our example. All ECS nodes are members of the pool and it is configured to listen on no port. This is because our Virtual Server will define all three NFS connectivity ports.

Server Pool Members						
Server Pool - ECS-EX300-NFS-Pool						
<input type="text" value="Search"/>						
Name	IP	Port	Weight	State	Backup Member	Max Concurrent Connections
EX300 Node 5	10.246.22.185		1	Enabled	● Disabled	
EX300 Node 4	10.246.22.184		1	Enabled	● Disabled	
EX300 Node 3	10.246.22.183		1	Enabled	● Disabled	
EX300 Node 2	10.246.22.182		1	Enabled	● Disabled	
EX300 Node 1	10.246.22.181		1	Enabled	● Disabled	

Figure 32 NFS server pool

**Step 2:** Create the NFS related persistence profile pool

A persistence profile is recommended for use by the NFS virtual server. Source IP as the persistence type is recommended along with a custom Timeout value of 86400 seconds to provide for a client to be persisted to the same NFS server for a 24-hour persistence period (adjust this value as appropriate for your environment).

Navigate to **Advanced Networking & Security > Load Balancing > Profiles > Persistence Profiles** and add a new profile of type **Source IP Persistence**

**Note:** A new persistence profile can also be created when creating the virtual server.

The screenshot shows the 'Profiles' tab in the VMware NSX-T Load Balancer interface. Under 'PERSISTENCE PROFILES', the 'ECS-Persistence-Profile' is selected. The left sidebar lists several profiles, with 'ECS-Persistence-Profile' checked. The right pane shows the 'Overview' for this profile, including its ID, description, and configuration details.

ECS-Persistence-Profile	
Name	ECS-Persistence-Profile
ID	ef425b3f-3965-41db-bbc6-d6dca9ac3805
Description	
Share Persistence	false
Persistence Entry	86400
Timeout (seconds)	
HA Persistence	false
Mirroring	
Purge Entries when	Full
Full	

Figure 33 NFS persistence profile

**Step 3:** Create the NFS virtual server

The NFS virtual server is created with each of the required NFS ports (see Table 1 for details). The ECS NFS server pool and persistence profile we created in the previous steps are associated with the virtual server.



Name	IP Address	Ports	Type	Load Balancer	Server Pool	Status
EX300-NFS *	10.246.25.253 * e.g. 10.10.10.10	2049 x 111 x 10000 x Enter Ports or Poi	L4 TCP	ECS-LB (x) v	ECS-EX300-N (x) v	
Description	DellEMC ECS NFS VS		Application Profile *		default-tcp-lb-app-profile (x) v	
Persistence	Source IP v					
Source IP *	ECS-Persistence-Profile (x) v					
> Additional Properties						
<div>SAVE</div> <div>CANCEL</div>						

Figure 34 NFS virtual server

**Step 4: NFS health monitoring**

Currently, it is not possible to associate multiple health monitors with a single server pool so only one of the three NFS ports are monitored for health.

One possible workaround for this would be to create 3 NFS server pools which contain the same ECS nodes but different ports (i.e. 1 with port 111, another with port 10000 and the 3<sup>rd</sup> with port 2049) all using the default TCP health check. Then create 3 separate virtual servers each with a unique NFS port then associate each with one of the NFS server pools.

### 2.4.3 Example: Site failover in an ECS multi-site configuration

In multiple-site ECS deployments the NSX-T load balancer can direct applications to an alternate ECS site in the case of a site outage, assuming the application can access storage at non-local sites within an acceptable level of performance.

Temporary site failure (TSO) refers to either a failure of the WAN connection between two sites or a temporary failure of an entire site (such as a power failure). ECS can detect and automatically handle any such temporary site failures. VDCs in a geo-replicated environment establish a heartbeat mechanism. Sustained loss of heartbeats for 15 minutes duration is indicative of a network outage and the system adjusts its behavior accordingly.

With **Access During Outage** enabled on a bucket and upon detecting a temporary outage, the system reverts to an eventual consistency model (reads/writes from a secondary (non-owner) site are accepted and honored). Further, a write to a secondary site during an outage causes the secondary site to take ownership of the object. This allows each VDC to continue to read and write objects from buckets in a shared namespace. An understanding of the concept of object owner on ECS, the access during outage (ADO) configuration and impact to object accessibility during a TSO are all critical to consider when planning for multisite multi-access object namespace. Refer to the [ECS Architectural and Overview](#) white paper for more details.

Currently, NSX-T does not support global load balancing however it can be configured to support an ECS multi-site DR configuration because the server pool can include backup members.

Server Pool Members

Server Pool - ECS-multi-site-pool

Q Search

Name	IP	Port	Weight	State	Backup Member	Max Concurrent Connections
ECS Site 2 Node 4	10.246.22.166	9020	1	Enabled	<div></div> Enabled	
ECS Site 2 Node 3	10.246.22.165	9020	1	Enabled	<div></div> Enabled	
ECS Site 2 Node 2	10.246.22.164	9020	1	Enabled	<div></div> Enabled	
ECS Site 2 Node 1	10.246.22.163	9020	1	Enabled	<div></div> Enabled	
ECS Site 1 Node 5	10.246.22.185	9020	1	Enabled	<div></div> Disabled	
ECS Site 1 Node 4	10.246.22.184	9020	1	Enabled	<div></div> Disabled	
ECS Site 1 Node 3	10.246.22.183	9020	1	Enabled	<div></div> Disabled	

CLOSE

Figure 35 Server pool with backup members

In the following figure, there are two ECS sites that are federated. If the primary members in the server pool which belong to VDC 1 become unavailable, then traffic will be automatically routed to the backup members in VDC 2. Traffic will be re-routed back to VDC 1 once connectivity has been restored.

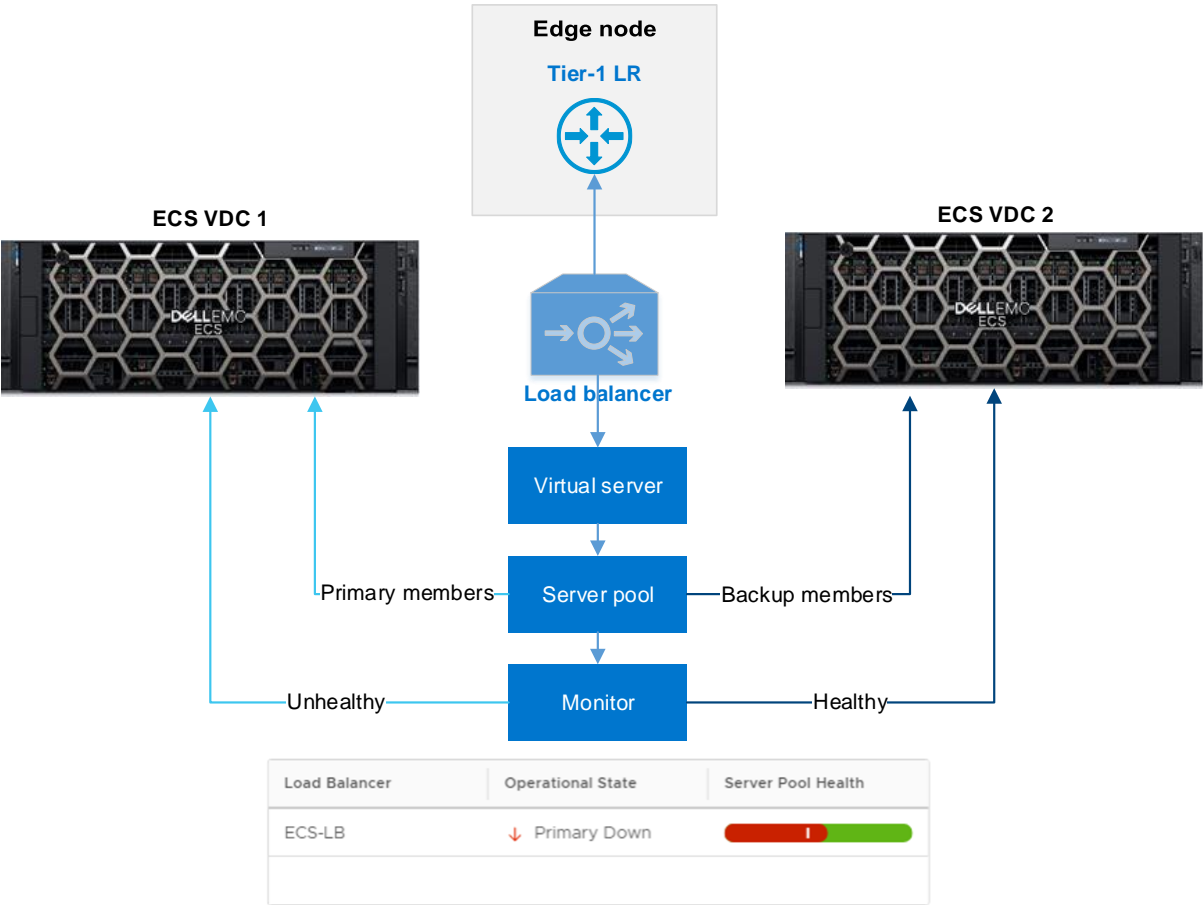


Figure 36 Route requests to ECS VDC 2 during temporary site outage

### 3 Best practices

The following best practices are recommended when using the VMware NSX-T load balancer with ECS.

Table 5 Best practices

Recommendation	Details
Do not use NSX-T load balancing for CAS traffic	The Centera SDK has a built-in load balancer and cannot function without direct access to all nodes.
Implement SSL Offloading	Terminate SSL connections at NSX-T when possible to reduce the load on the ECS nodes
If SSL termination is required on ECS nodes	Use Layer 4 (TCP) to pass through the SSL traffic to ECS nodes for handling. The certificates here are installed on the ECS nodes and not on the NSX-T virtual server.
Use session persistence for NFS workloads	For NFS workloads configure NSX-T to keep client sessions terminated on a single ECS node. Only during node failure should an NFS session be torn down and established on another ECS node. Balancing NFS traffic across ECS nodes is inefficient because it does not take advantage of ECS caching.

# A Troubleshooting

## A.1 View access logs

Enable Access Log under the Additional Properties section for the virtual server.

Figure 37 Enable the virtual server access log

SSH into the NSX Edge server to view the logs. Run the below command to tail the access log

**Note:** Run **get load-balancer** to obtain the load balancer and virtual server IDs.

```
nsx-edge> get load-balancer 0f1b169f-2714-4b54-9f2d-e77733415466 virtual-server 4b46766c-913b-47de-8ef9-6c4df9a6624a access-log follow
10.246.156.172 - - [12/Jun/2019:19:45:11 +0000] "GET / HTTP/1.1" 200 425 "-" "S3 Browser 7-6-9 https://s3browser.com"
10.246.156.172 - - [12/Jun/2019:19:45:11 +0000] "GET /mybucket/?delimiter=%2F&max-keys=1000 HTTP/1.1" 200 599 "-" "S3 Browser 7-6-9 https://s3browser.com"
```

## A.2 Packet captures

**Step 1:** Verify that you are on the active edge node of the load balancer

```
nsx-edge> get load-balancer 0f1b169f-2714-4b54-9f2d-e77733415466 status
```

Load Balancer

```
UUID           : 0f1b169f-2714-4b54-9f2d-e77733415466
Display-Name    : ECS-LB
Enabled         : True
LB-State        : ready
LR-HA-State     : active
Virtual Servers : 3
Up Virtual Servers : 3
Pools          : 3
Up Pools       : 3
```

**Step 2:** Locate the interface of the tier-1 hosting load balancer

---

**Note:** Certain information has been intentionally excluded from the below listing.

---

```
nsx-edge> get logical-router 69410ab1-58e3-4a25-9f19-620f5f98cfc1 interfaces
```

```
Logical Router
```

UUID	VRF	LR-ID	Name	Type
69410ab1-58e3-4a25-9f19-620f5f98cfc1	4	8	SR-T1-Standalone	SERVICE_ROUTER_TIER1

```
Interfaces
```

```

Interface : 15249021-6e8b-441c-bfd1-7a0f741ee95f ← This the T1 uplink interface
Ifuid     : 302
Name      : t1-T1-Standalone-76ff1020-71d9-
Mode      : lif
IP/Mask    : 10.246.25.213/24

Interface : f1f90b93-c157-4c60-a651-4f66f58acbcc
Ifuid     : 291
Mode      : loopback
IP/Mask    : 10.246.25.253/32;10.246.25.216/32;127.0.0.1/8;::1/128 ← This is the VIP on the loopback
interface

Interface : 11ef63f7-74d1-47db-869e-0ee3ac47d853 ← This is the T1 downlink interface
Ifuid     : 292
Name      : bp-sr0-port
Mode      : lif
IP/Mask    : 169.254.0.2/28;fe80::50:56ff:fe56:5300/64

```

**Step 3:** Start a packet capture on the Edge Node T1 “uplink” interface

The IP in red is the client IP and the one is blue is the VIP.

```

nsx-edge> start capture interface 15249021-6e8b-441c-bfd1-7a0f741ee95f expression port 443
17:42:53.314996 00:2a:6a:db:10:43 > 00:50:56:b9:70:60, ethertype 802.1Q (0x8100), length 64: vlan 0, p 0,
ethertype IPv4, 10.246.156.172.50058 > 10.246.25.213.443: Flags [R.], seq 1118678306, ack 155241183,
win 0, length 0
<base64>AFBWuXBgACpq2xBDgQAAAAgARQAAKAT+QAB/BiplCvacrAr2GdXDigG7Qq2tIglAyt9QFAAAWi
4AAAAAAAAAAAAA==</base64>

```

## B Technical support and resources

[Dell.com/support](https://dell.com/support) is focused on meeting customer needs with proven services and support.

[Storage technical documents and videos](#) provide expertise that helps to ensure customer success on Dell EMC storage platforms.

### B.1 Related resources

#### B.1.1 ECS product documentation

- Dell EMC ECS Product Documentation:
  - <https://community.emc.com/docs/DOC-73931>
- Dell EMC ECS Architecture and Overview:
  - <http://www.emc.com/collateral/white-papers/h14071-ecs-architectural-guide-wp.pdf>
- Dell EMC ECS Networking and Best Practices:
  - <http://www.emc.com/collateral/white-paper/h15718-ecs-networking-bp-wp.pdf>
- Dell EMC ECS Best Practices:
  - <https://www.emc.com/collateral/white-papers/h16016-ecs-best-practices-guide-wp.pdf>

#### B.1.2 VMware NSX-T load balancer documentation

- VMware NSX-T product documentation:
  - <https://docs.vmware.com/en/VMware-NSX-T-Data-Center/2.4/administration/GUID-46567C8D-A5C5-4793-8CDF-858E58FDE3C4.html>