White Paper

Deployment Practices and Guidelines for NetScaler 10.1 on Amazon Web Services

Citrix NetScaler on Amazon Web Services (AWS) enables enterprises to rapidly and cost-effectively leverage world-class NetScaler application delivery capabilities within their Amazon Web Services deployments. NetScaler on AWS combines the elasticity and flexibility of the AWS Cloud with the same optimization, security and control NetScaler provides for the most demanding websites and applications in the world.

Because the corresponding Amazon Machine Image (AMI) is a packaging of the same binary used on NetScaler MPX™/NetScaler SDX™ hardware and NetScale VPX™ virtual appliances, enterprises obtain all of the same L4-7 functionality familiar from their on-premise deployments, including load balancing, content switching, global server load balancing, application firewall and SSL VPN. This enables numerous compelling use cases, from hybrid cloud (i.e., spillover) and production delivery scenarios, to implementations for business continuity and application development and testing.

Due to some of the design characteristics of the AWS Cloud, however, there are a handful of differences that network architects need to be aware of with regard to how NetScaler® on AWS works and, therefore, how it needs to be configured. For example, because AWS does not expose Layer 2 networking capabilities to customers, administrators will need to enable high availability pairs differently than they do with their on-premise deployments.

Broken into three distinct sections, this paper:

- Further examines the potential use cases for NetScaler on AWS, including the role of Citrix® CloudBridge™.
- Provides a primer on the nomenclature, high-level architecture and networking details applicable to Amazon Web Services.
- Delivers prescriptive guidance on how to account for deploying NetScaler on AWS given the AWS network design.

Use Cases

Compared to alternative solutions that require each service to be deployed as a separate virtual appliance, NetScaler on AWS combines L4 load balancing, L7 traffic management, global server load balancing, server offload, application acceleration, application security and other essential application delivery capabilities in a single AMI, conveniently available via the AWS Marketplace. Furthermore, everything is governed by a single policy framework and managed with the same, powerful set of tools used to administer on-premise NetScaler deployments. The net result is that NetScaler on AWS enables several compelling use cases that not only support the immediate needs of today's enterprises, but also the ongoing evolution from legacy computing infrastructures to enterprise cloud datacenters.

Production Delivery – Enterprises actively embracing AWS as an infrastructure- as-a-service (IaaS) offering for production delivery of applications can now front-end those applications with the same cloud networking platform used by the largest websites and cloud service providers in the world. Extensive offload, acceleration and security capabilities can be leveraged to enhance performance and reduce costs, at the same time that global server load balancing is used to ensure availability across AWS availability zones, AWS regions, and between AWS and on-premise datacenters.

Hybrid Cloud Designs – With NetScaler on AWS, hybrid clouds that span enterprise datacenters and extend into AWS can benefit from the same NetScaler cloud networking platform, significantly easing the transition of applications and workloads back and forth between a private datacenter and AWS. The full suite of NetScaler capabilities, ranging from intelligent database load balancing with DataStream to unprecedented application visibility with AppFlow and real-time monitoring and response with Action Analytics, can be leveraged with NetScaler on AWS.

Business Continuity – Enterprises looking to use AWS as part of their disaster recovery and business continuity plans can rely upon NetScaler global server load balancing running both on-premise and within AWS to continuously monitor availability and performance of both enterprise datacenters and AWS environments, ensuring users are always sent to the optimal location.

Development and Testing – Enterprises running production delivery on-premise but using AWS for development and testing can now include NetScaler within their AWS test environments, speeding time-to-production due to better mimicry of the production implementation within their test environments.

In each use case, network architects can also leverage Citrix CloudBridge— configured either as a standalone instance or as feature of a NetScaler platinum edition instance—to secure and optimize the connection between the enterprise datacenter(s) and the AWS Cloud, thereby speeding data transfer/synchronization and minimizing network costs.

AWS Nomenclature and Networking Details

Taking maximum advantage of the opportunities for NetScaler on AWS requires an understanding of the AWS Cloud and how it works.

AWS High-level Architecture

EC2 versus VPC. AWS encompasses multiple different services, such as Amazon Simple Storage Services (S3), Amazon Elastic Compute Cloud (EC2), and Amazon Virtual Private Cloud (VPC). The distinction between the latter two is important in this case. In particular, with EC2, virtual machine instances are limited to a single networking interface and single IP address. Furthermore, there are minimal networking features and controls. This precludes the use of EC2 for NetScaler— which requires a minimum of three IP addresses—and is why NetScaler instances can only be launched within an AWS VPC.

VPCs not only support virtual machines with multiple interfaces and multiple private and public IP addresses, but also allow you to create and control an isolated virtual networking environment, with its own IP address range, subnets, routing tables and network gateways. (Note on terminology: Accepted AWS nomenclature is to refer to all virtual machine instances as “EC2 instances,” but to then qualify this accordingly if they are launched within a VPC. Thus, it is not uncommon to see NetScaler on AWS referenced as an EC2 instance, even though it’s technically “an EC2 instance launched within a VPC.”)

Regions and Availability Zones. Within the AWS Cloud, Regions refer to a specific geographic location, such as US East. Within each Region there are at least two Availability Zones, each of which can be thought of as an independent cloud datacenter that has been engineered to be insulated from failures in other Availability Zones and to provide inexpensive, low-latency network connectivity to other Availability Zones within the same Region. By implementing instances in separate Availability Zones, you can protect your applications from failures that impact a single location.

Limitations and dependencies for network architects to be aware of at this level include the following:

- Although a Virtual Private Cloud can span multiple Availability Zones, it cannot span multiple Regions.
- Individual subnets within a VPC cannot span multiple Availability Zones.
- All traffic entering or leaving a VPC must be routed via a corresponding default Internet gateway.

AWS Networking Details

ENIs and EIPs. NetScaler instances launched into a VPC can have up to eight elastic network interfaces (ENIs). In turn, each ENI can be assigned one or more private IP addresses, with each of these optionally being mapped to an elastic IP address that is publicly routable. What makes the network interfaces and IP addresses “elastic” in this case is the ability to programmatically re-map them to other instances—a feature that enables recovery from instance or Availability Zone failures without having to wait for hardware replacements, or for DNS changes to fully propagate to all of your customers. Other details to account for include the following:

- An instance can have different ENIs in different subnets (but not in different Availability Zones).
- Each ENI must have at least one IP address assigned to it, and must be assigned to a Security Group (see below).
- Addresses 1 to 4 for each subnet (i.e., 10.x.x.1-4) are reserved for use by Amazon.
- NetScaler is only aware of private IP addresses. Any EIPs that are assigned will not show up within the NetScaler CLI or any related management tools.

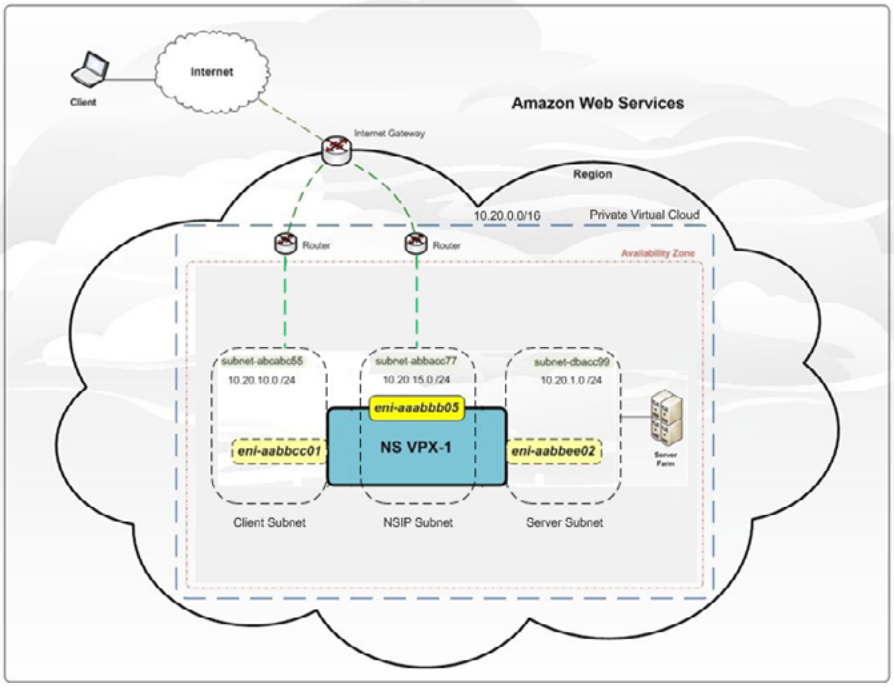


Figure 1: EC2 instance of NetScaler launched within an AWS VPC and configured with three ENIs on three subnets (one each for management, client, and server connections) and external network connectivity for the first two.

Source: http://support.citrix.com/proddocs/topic/netscaler-vpx-10-5/nsvpx-aws-hownsvpx-aws-works-con.html

Layer 2 Networking. With AWS, no layer 2 networking capabilities are exposed. This has three relevant implications. First, security groups are used for L2 isolation in place of VLANs. These act as a virtual firewall, controlling inbound and outbound traffic flow at the instance level (whereas network ACLs perform a similar function at the subnet level). The second implication is that the L2 mechanism typically used for failover between NetScaler pairs configured for high availability is not supported, requiring implementation of an alternate approach. The final implication is that the following capabilities are not supported for NetScaler on AWS:

- Gratuitous ARP (or GARP)
- L2 mode
- Tagged VLAN
- Virtual MAC (or VMAC)

Instance Types. EC2 instance types define the resources available to a virtual machine in terms of processing power, memory, disk, ENIs, private IP addresses per ENI, and network I/O. Because NetScaler requires at least two vCPUs and 2GB of memory, this dictates use of an EC2 instance type of M3.

Other aspects of the AWS Cloud that network architects should keep in mind are that both servers and the underlying network are shared resources.

NetScaler on AWS Deployment Practices and Guidelines

The following sections identify required practices and recommended guidelines for deploying NetScaler on AWS.

Basic Configuration

Citrix supplies the NetScaler AMIs for each AWS region. As a result, there is no need for you to download code or create your own AMIs. In fact, it is not possible to create your own NetScaler AMI. All you need to do to get started is select the NetScaler AMI from AWS Marketplace and then launch it by:

- Selecting the Region, VPC, and Availability Zone to launch the instance into;
- Setting up each ENI, which includes configuring the subnet it belongs to, the security group(s) it belongs to, and the private IP addresses that are assigned to it;
- Assigning EIPs to private IP addresses, as needed; and,
- Logging into the NetScaler instance to apply its license and configure how you want it to process application traffic

Citrix recommends using a configuration with at least three ENIs and three subnets, one pair of each for management traffic (i.e., NSIP), client-side traffic (i.e., VIPs), and server-side traffic (i.e., SNIPs and MIPs—as shown in Figure 1. Although the use of fewer ENIs and subnets is technically supported, such implementations are not recommended due to resulting limitations and the increased potential for configuration errors. For example, if only a single ENI and subnet are used, then there is no support for high availability (HA) pairs.

Establishing access to/from resources outside the VPC requires one of three configurations. Using remote management of a NetScaler instance as an example, the options are to (1) assign an EIP to the NSIP, (2) setup NAT for the NSIP, or (3) configure a VPN tunnel into the VPC.

Routing for a VPC

To configure routing for NetScaler on AWS, administrators must first understand the following conditions and requirements:

- Every VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If you don't explicitly associate a subnet with a particular route table, the subnet uses the main route table.
- Each route in a table specifies a destination CIDR and a next hop, and routes are resolved using longest prefix match.
- You can enable Internet access to any subnet in the VPC by adding to its route table a default route with next hop as the Internet Gateway of the VPC In many cases, the VPC's Internet Gateway is added to the main route table itself.

- The AWS VPC architecture has a restriction that traffic sourced from a given ENI can only use the ENI's corresponding default gateway as the next hop to reach remote destinations. For example, server response traffic from the NetScaler with the VIP as the source IP address can only use the VIP subnet's default gateway to reach remote clients. In other words, the NSIP subnet's default gateway cannot be used as the next hop for server response traffic.

Given these conditions and requirements, there are three deployment scenarios to consider when it comes to configuring routing for a NetScaler on AWS implementation.

Scenario 1 – Clients are on the Internet (i.e., using public IP addresses), all application servers are in the same Availability Zone and the same VPC, and the NSIP is on a separate interface from the VIP and the server subnets (illustrated in figure 1). This scenario requires 3 ENIs and three corresponding subnets, one pair each for management traffic (NSIP), client-side traffic (VIPs) and server side traffic (SNIPs/MIPs).

In this case, there are two types of external destinations that the NetScaler potentially needs to communicate with:

- Remote access clients that access the NSIP to manage and administer the NetScaler device.
- Internet clients that access the VIP.

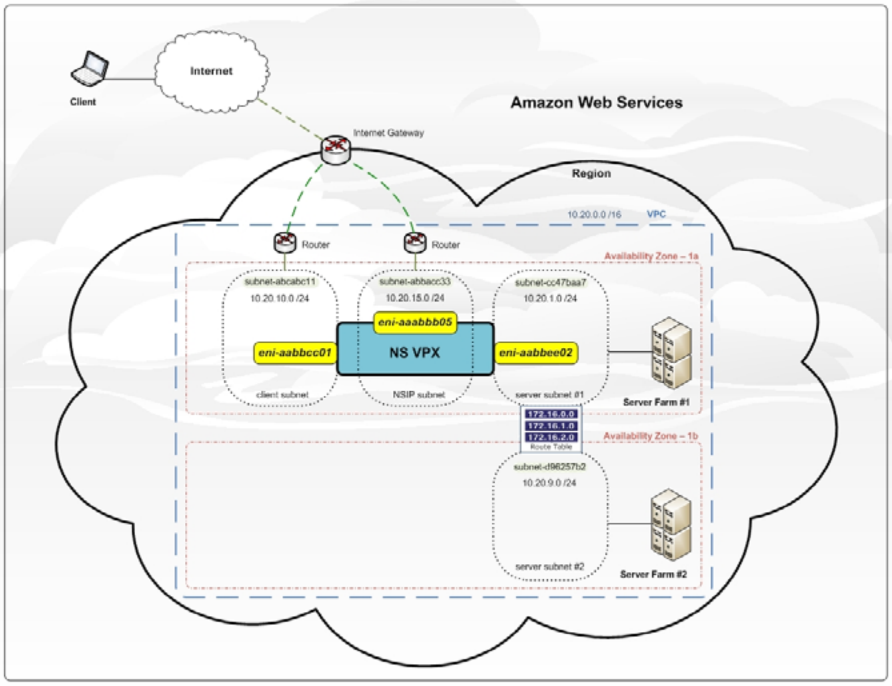
However, the NetScaler can only have one default route (0.0.0.0/0), and it can only be used by traffic sourced from one of the ENIs (in other words, the default route can only be used to reach one of the remote destinations described above). To get around this limitation and establish a path to both types of remote destinations, you need to implement Policy Based Routing (PBR) on the NetScaler.

Specifically, you can apply the default route to reach remote access clients for management purposes (i.e., point the default route to the default gateway corresponding to the NSIP ENI). Then for the VIP to client traffic, you can apply a PBR rule to match the source IP address(es) of the VIP(s) and with the next hop as the default gateway corresponding to the VIP subnet ENI. This is a required configuration for any deployment consisting of remote Internet clients accessing both the VIP and the NSIP.

To summarize, this deployment scenario requires configuration of the following routes and parameters:

- A default route pointing to the NSIP default gateway, for remote management.
- PBR policies with source IP addresses as the VIPs and next hop as the VIP subnet's default gateway, for responding to remote clients. Note: the default gateway for any subnet in a VPC is usually the first IP address in the subnet.
- A SNIP or a MIP in the same subnet as the servers, to communicate with the servers. Note that you can use either a SNIP or a MIP, although a SNIP is recommended for servers in local subnets. Also note that no explicit route is needed to reach the servers, as they belong to the same subnet as the SNIP/MIP ENI.
- A default route with next hop as the Internet gateway for the main routing table of the VPC, to enable external access. No other VPC routing table modifications are needed.

Scenario 2 – Clients are on the Internet, application servers are in the same VPC but different Availability Zone, and the NSIP is on a separate interface from the VIP and the server subnets. See Figure 2.



.
Figure 2: application servers are in the same VPC but different Availability zones or different VPC that has been Peered.

This scenario is identical to Scenario 1 with the exception that the servers to be load balanced are in a subnet that is located in a different Availability Zone. One common use case for such a deployment model would be for GSLB across Availability Zones.

As discussed previously, the AWS architecture allows a VPC to span multiple Availability Zones, but requires that each subnet be completely contained within one Availability Zone (i.e., a subnet cannot span multiple Availability Zones). Furthermore, an instance defined in one Availability Zone cannot be associated with an ENI that belongs to a different Availability Zone.

The latter restriction is significant to the NetScaler in this deployment scenario because this means that the server subnet would essentially have to be “remote” to the NetScaler, as AWS does not allow you to define a SNIP/MIP ENI that is in the same subnet as the remote servers. However, you can create a separate ENI for the SNIP/MIP and add a static route on the NetScaler to reach the remote server subnet, with the next hop as the default gateway of the SNIP/MIP ENI.

The net result is that this deployment scenario requires configuration of the following routes and parameters:

- A default route pointing to the NSIP default gateway, for remote management.
- PBR policies with source IP addresses as the VIPs and next hop as the VIP subnet's default gateway, for responding to remote clients.
- A SNIP or a MIP in a separate ENI, and a static route with destination as the remote server subnet and next hop as the SNIP's default gateway. Note that you can use either a SNIP or a MIP for this purpose. The key reason that a static route can be used here as opposed to a PBR rule is that the servers belong in a known subnet that can be reached through a specific route, unlike remote Internet clients that can only be reached through a default route.
- A default route with next hop as the Internet gateway for the main routing table of the VPC, to enable external access. No other VPC routing table modifications are needed. This includes requiring no changes for the SNIP subnet to reach the remote server subnet. Because both of these subnets are considered local, they can communicate using the local route entry that is present by default in the VPC routing tables.

Scenario 3 – VIPs and the NSIP are on different subnets within the same VPC, but application servers are external to the VPC. See Figure 3.

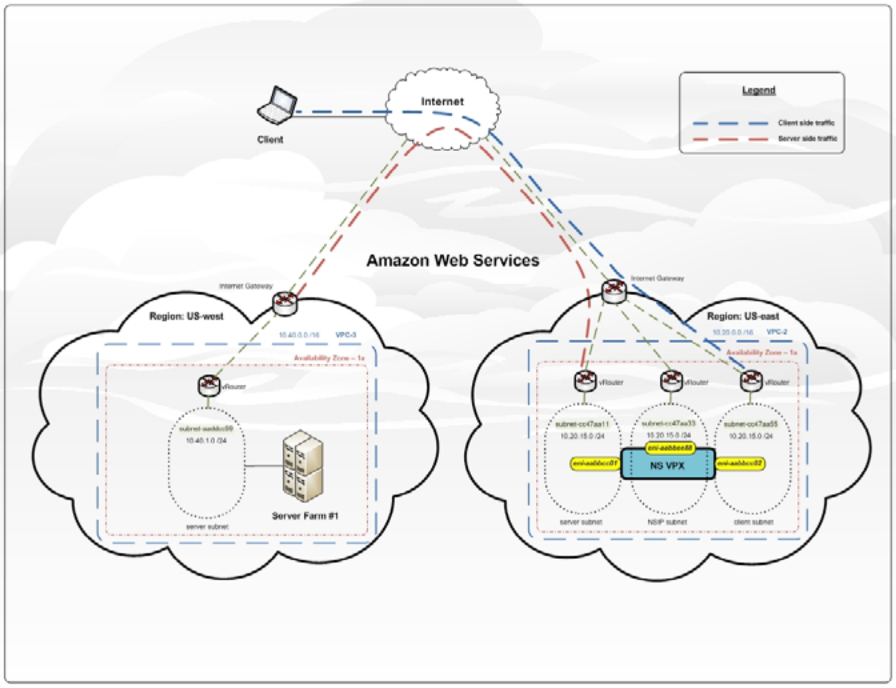


Figure 3: application servers are external to the VPC

This deployment scenario illustrates the case where the servers are external to the VPC, and is useful for customers who would want to re-purpose existing compute capacity in an existing VPC for a new deployment. Another variant of this deployment is where the servers are existing EC2 servers that a customer would want to utilize while defining a new VPC.

The main difference between this deployment scenario and the previous two is that in this case, the servers are external to the VPC and, therefore, need to be reached through a gateway. In this respect, the servers are no different from remote clients and can be reached by the NetScaler through the same routing mechanisms that are used to reach remote clients.

For this scenario to work correctly, the application servers themselves need to be mapped to EIPs so that they are publicly reachable. The SNIP IP addresses must also be mapped to elastic IP addresses so that the external application servers can respond to them.

Although the SNIP/MIP can technically be added in the same subnet as the VIPs to reach the remote servers, it is recommended that a separate ENI/subnet be created for this purpose.

To summarize, this deployment scenario requires configuration of the following routes and parameters:

- A default route pointing to the NSIP default gateway, for remote management.
- PBR policies with source IP addresses as the VIPs and next hop as the VIP subnet's default gateway, for responding to remote clients.
- A SNIP or a MIP in a separate ENI, and PBR policies with source IP addresses as the SNIP/MIP and next hop as the SNIP subnet's default gateway, for reaching external servers. Note that you can use either a SNIP or a MIP for this purpose. A PBR policy is needed here as the servers themselves may be scattered across different IP addresses and may not be contained in one known subnet.
- A default route with next hop as the Internet gateway for the main routing table of the VPC, to enable external access. No other VPC routing table modifications are needed.
- Although VPC allows you to do so, there is no need to associate the SNIP subnet with a separate routing table because by default, every VPC subnet is associated with the main routing table unless otherwise specified. Since in this case, the default route for external access has been added to the main routing table, the SNIP subnet can make use of it to reach the external servers.

High Availability Pairs

Because L2 networking is not exposed in the AWS Cloud, high availability pairs must use an alternate mechanism to accomplish failover from the primary node to the secondary node in the event of a primary-node outage. The alternate approach that's used in this case takes advantage of the "elastic" characteristic of ENIs – specifically, the ability to dynamically re-map them from one instance to another one. Effectively what happens during a failover event is that the data-plane ENIs from the primary node are migrated to the secondary node (see Figure 4).

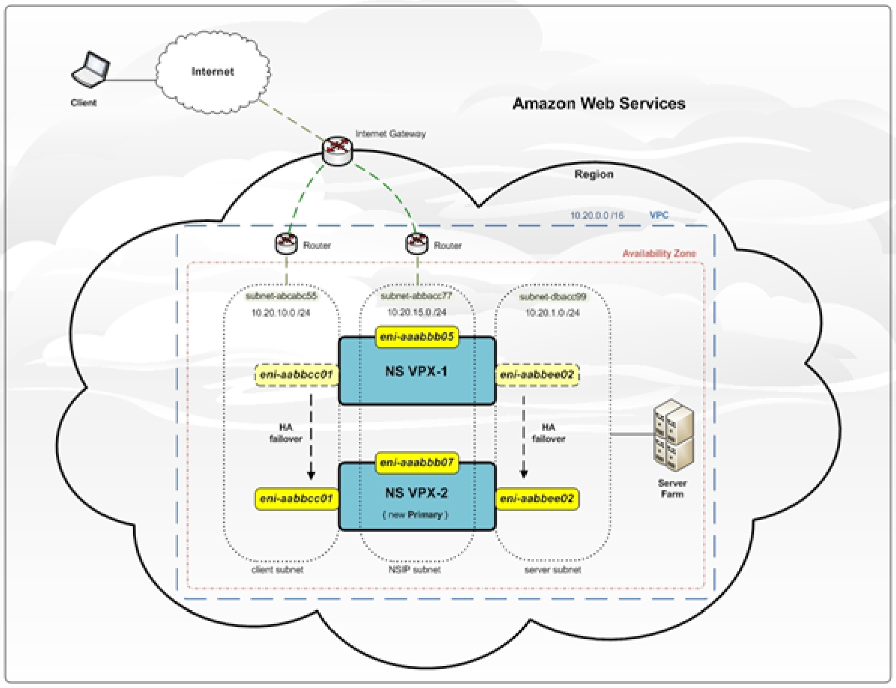


Figure 4: HA-pair failover for NetScaler on AWS.Source:

Source: http://support.citrix.com/proddocs/topic/netscaler-vpx-10-5/nsvpx-aws-ha-con.html

Prerequisites for this approach to work, along with associated implications, are as follows:

- The primary instance must be configured with at least two ENIs (i.e., the default one for management traffic and all others for data-plane traffic), while the secondary instance must be configured with only the default ENI (for management traffic).
- Because the default ENI cannot be migrated, it should not be configured to support data-plane traffic.
- The NSIP must configured with an EIP so that it can reach the AWS API server.
- All heartbeat packets utilize the management interfaces.
- The process of migrating data-plane ENIs on loss of heartbeat can take up to twenty seconds to accomplish.
- Because migrated ENIs must remain on the same subnets they started on, both instances must reside in the same Availability Zone—in other words, HA pair configurations cannot span Availability Zones.

For those scenarios where the potential for a twenty second delay is unacceptable, NetScaler global server load balancing (GSLB) functionality provides another option for network architects to consider. Although GSLB is typically used for load sharing across sites, it can also be configured to enable local failover.

Site-level Load Sharing and Failover

Applicability of site-level load sharing and failover capabilities will depend on how an enterprise decides to deploy its applications both across its own datacenters and the AWS Cloud. Put another way, unless the same applications are deployed in multiple locations, then there is no role for these capabilities.

That said, for organizations that are even mildly concerned about the availability of their applications, best practice will be to deploy them in multiple Availability Zones; while for those that availability is of paramount importance, best practice will be to pursue the added measure of deploying them in multiple AWS Regions. In most cases, therefore, site-level load sharing and failover will indeed have a role to play.

Whether that role is to direct users to the optimal site based on performance characteristics or to enable HA between Availability Zones, Regions, and/or an on- premise datacenter and the AWS Cloud, the solution is the same: NetScaler GSLB.

Setting up GSLB for NetScaler on AWS largely consists of configuring the NetScaler to load balance traffic to servers located outside the VPC that the NetScaler belongs to, such as within another VPC in a different Availability Region or an on-premise datacenter etc. The steps to configure NetScaler to load balance traffic to external servers are detailed in Scenario 3. To summarize:

- Configure a MIP/SNIP ENI on the NetScaler instance to reach the external servers.
- Add a Policy Based Route (PBR) with source IP address as the MIP and next hop as the default gateway of the MIP ENI, to route to the external servers outside the VPC.
- Add an Internet gateway route to the VPC's main routing table.
- Associate an EIP with the MIP created above, so that the external server can respond to the MIP.
- Configure GSLB parameters via the NetScaler GUI or CLI.

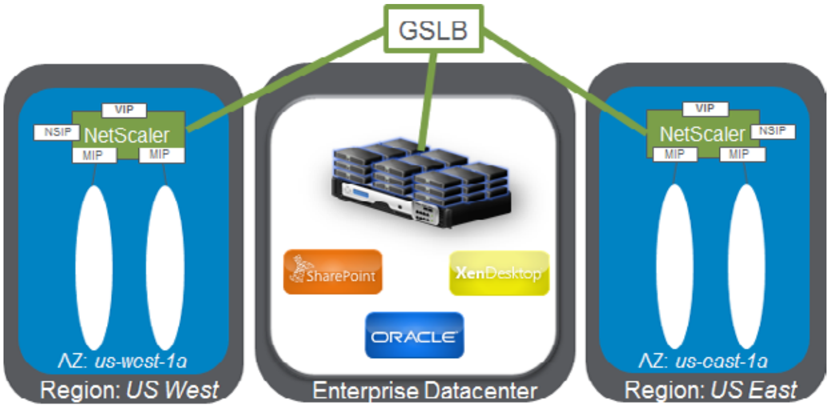


Figure 5: Load-sharing and HA between an on-premise datacenter and Availability Zones within two separate regions.

Automatic Response and Scaling

One of the attractive features of AWS is the flexibility it provides for adding new instances of an existing application—for example, to support increased user demand. This scenario has a pair of potential implications for an associated NetScaler on AWS deployment.

First, there's the need for NetScaler to automatically account for the new application instances—that is, to ensure that their application traffic is processed according to the same optimization, security and control policies in place for the original set of instances. This can be accomplished in three ways:

- By applying the associated set of policies to a specific range of IP addresses and then ensuring that any new application instances that are introduced fall within that range;
- By leveraging the NetScaler "auto-scale" feature, which works by applying policies to a domain-based service group and then periodically polling DNS to identify the IP addresses of all application instances associated with the given group/domain name; or,
- By developing custom orchestration scripts that leverage AWS and NetScaler APIs to gather required data and respond accordingly (i.e., scale up or down).

The second implication involves the potential need to increase the capacity of the NetScaler implementation itself, in order to handle the increased application traffic. The recommended approach in this case is sometimes referred to as a "pilot light" configuration. This is where multiple NetScaler instances are configured in advance in a GSLB cluster arrangement, with all but an initial complement remaining inactive until needed.

Conclusion

For enterprises that are embracing pure and hybrid cloud delivery models, NetScaler on Amazon Web Services holds tremendous potential. With it, they can optimize, secure, and control their cloud-delivered applications—not only in the same manner as the world's largest websites, but also identically to how they're already doing so for their on-premise datacenters that leverage NetScaler. Fully realizing this potential depends, however, on having a thorough understanding of how the AWS Cloud differs from typical enterprise networks, and subsequently adhering to the deployment practices and guidelines specified herein that help account for these differences.

Corporate Headquarters
Fort Lauderdale, FL, USA

Silicon Valley Headquarters
Santa Clara, CA, USA

EMEA Headquarters
Schaffhausen, Switzerland

India Development Center
Bangalore, India

Online Division Headquarters
Santa Barbara, CA, USA

Pacific Headquarters
Hong Kong, China

Latin America Headquarters
Coral Gables, FL, USA

UK Development Center
Chalfont, United Kingdom

About Amazon Web Services
Launched in 2006, Amazon Web Services (AWS) began exposing key infrastructure services to businesses in the form of web services—now widely known as cloud computing. The ultimate benefit of cloud computing, and AWS, is the ability to leverage a new business model and turn capital infrastructure expenses into variable costs. Businesses no longer need to plan and procure servers and other IT resources weeks or months in advance. Using AWS, businesses can take advantage of Amazon's expertise and economies of scale to access resources when their business needs them, delivering results faster and at a lower cost. Today, Amazon Web Services provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers hundreds of thousands of enterprise, government and startup customers businesses in 190 countries around the world. AWS offers over 30 different services, including Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3) and Amazon Relational Database Service (Amazon RDS). AWS services are available to customers from data center locations in the U.S., Brazil, Europe, Japan, Singapore, and Australia.

About Citrix
Citrix (NASDAQ:CTXS) is a leader in mobile workspaces, providing virtualization, mobility management, networking and cloud services to enable new ways to work better. Citrix solutions power business mobility through secure, personal workspaces that provide people with instant access to apps, desktops, data and communications on any device, over any network and cloud. This year Citrix is celebrating 25 years of innovation, making IT simpler and people more productive. With annual revenue in 2013 of $2.9 billion, Citrix solutions are in use at more than 330,000 organizations and by over 100 million users globally. Learn more at www.citrix.com.