Technical Report

NetApp Private Storage for Amazon Web Services (AWS)

Solution Architecture and Deployment Guide

Mark Beaupre, NetApp
April 2016 | TR-4133

Abstract

This document describes the architecture for the NetApp® Private Storage for Amazon Web Services (AWS) solution and provides procedures for deploying and testing the solution.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 NetApp Private Storage for AWS Solution

This document describes the storage architecture of the NetApp Private Storage for Amazon Web Services solution and provides procedures for deploying and testing the solution.

1.1 Assumptions

This document assumes that the reader has working knowledge of the following:

- Amazon Web Services (AWS)
- NetApp storage administration
- Network administration
- Windows and/or Linux administration

1.2 Use Case Overview

The NetApp Private Storage for AWS solution is a cloud-connected storage architecture that allows enterprises to build an agile cloud infrastructure that combines the scalability and flexibility of the AWS cloud with the control and performance of NetApp storage.

NetApp storage is deployed at an Equinix colocation data center where the AWS Direct Connect service is available, and the NetApp storage is connected to AWS compute resources through the AWS Direct Connect network service.

Typical use cases for the NetApp Private Storage for AWS solution include the following:

- Oracle, Microsoft SQL Server, and SAP primary workloads
- Disaster recovery
- Development and testing
- Big data analytics
- Data with compliance requirements
- Data center migration and consolidation

For more information about NetApp Private Storage use cases, see NVA-0009 NetApp Validated Architecture.

1.3 Technical Overview

The NetApp Private Storage for AWS solution combines computing resources from AWS with NetApp storage deployed at AWS Direct Connect data centers. Connectivity from the NetApp storage to the AWS cloud is made possible by the AWS Direct Connect network service.

Customers who deploy the NetApp Private Storage for AWS solution at Equinix colocation data centers can also provision AWS Direct Connect network connections through the Equinix Cloud Exchange portal in 200Mb/sec or 500Mb/sec bandwidth sizes. 1Gb/sec and 10Gb/sec Direct Connect connections are provisioned manually by cross connect.

In the AWS Direct Connect data center, the customer provides network equipment (switch or router) and NetApp storage systems. Virtual machines (VMs) in the AWS cloud connect to the NetApp storage through IP-based storage protocols (iSCSI, CIFS, or NFS). Additional MPLS or point-to-point VPN network resources can be used to provide connectivity between AWS regions as well as connectivity to on-premises data centers.

2 NetApp Private Storage for AWS Solution Architecture

This section describes the components of the solution architecture and explains the data security elements that are provided.

2.1 Solution Architecture Components

The solution architecture consists of the following components:

- AWS EC2
- AWS VPC
- AWS Direct Connect
- Equinix colocation data center (AWS Direct Connect data center)
- Equinix Cloud Exchange
- Border Gateway Protocol
- Customer-provided layer 3 network equipment
- NetApp storage (FAS and FlexArray)

AWS EC2

Amazon EC2 is a web service that provides resizable computing capacity in the cloud. This environment provides preconfigured VM templates called AMIs.

EC2 Locations

The AWS EC2 service is available on a per-AWS region basis. Each AWS region is tied to a specific geographic location.

The AWS EC2 management web interface is used to deploy EC2 VM resources for the NetApp Private Storage for AWS solution. Advanced Amazon users can programmatically deploy EC2 VMs through APIs and scripts that use AWS command-line tools or AWS modules for Windows PowerShell®.

For more information about locations where the AWS EC2 service is available, refer to AWS EC2 Product Details.

EC2 Instance Types

EC2 VMs have various instance types that support the computing needs of a customer. Each instance type is a combination of CPU, memory, storage, and network bandwidth.

For more information about the available EC2 instance types, refer to AWS EC2 Instances.

Note: Not all instance types are available for all AWS regions.

Available Operating Systems

In addition to different instance types, EC2 VMs can run different OSs. Amazon EC2 VMs can run Windows or Linux OSs. For a list of available operating systems in AWS, refer to the Amazon Marketplace.

For each OS and application type, you can validate version compatibility with the NetApp client software and Data ONTAP® version through the NetApp Interoperability Matrix Tool. (This site requires a NetApp Support account login.)

AWS Virtual Private Cloud

The AWS VPC service provides isolated RFC 1918 IPv4 address ranges (`10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`) into which EC2 VMs can be deployed.

A VPC can be customized—including its Classless Inter-Domain Routing (CIDR) IP address ranges, subnets, routing, gateways, Domain Name System (DNS) settings, and network security—through access control lists (ACLs) and security groups.

A VPC can be connected to the customer network located in the Direct Connect data center or to on-premises customer networks through a point-to-point VPN.

The VPC can span multiple Availability Zones within an AWS region. VPC subnets cannot span multiple Availability Zones.

For more information, refer to the AWS Virtual Private Cloud documentation.

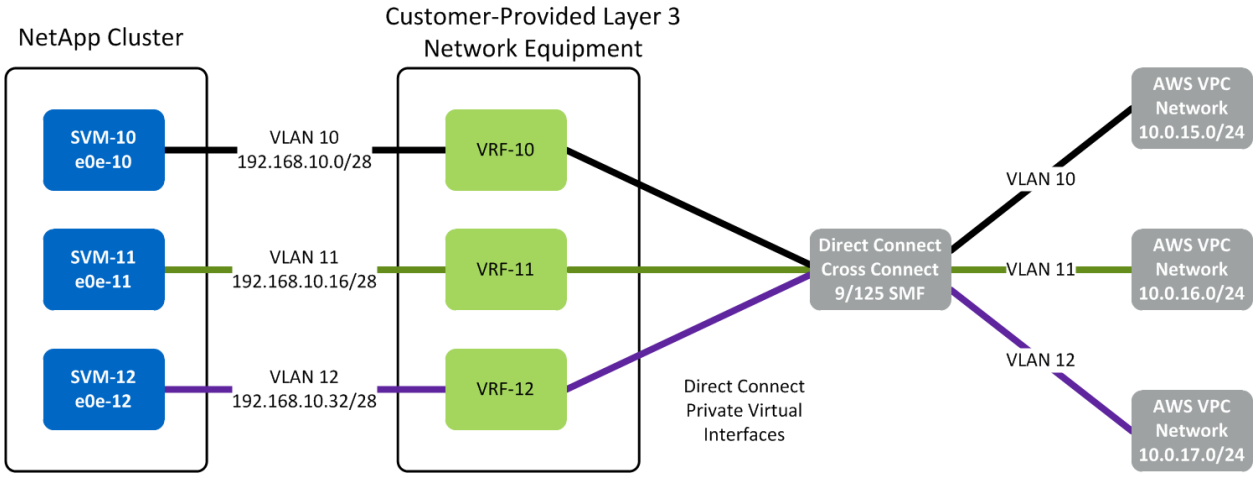
AWS Direct Connect

AWS Direct Connect is used to establish a dedicated network connection between the customer-provided network switch or router in the AWS Direct Connect data center and the Amazon VPC. Direct Connect supports the use of industry standard 802.1Q virtual local area networks (VLANs).

By using multiple VLANs, customers can partition the Direct Connect dedicated connection into multiple Direct Connect private virtual interfaces. As Figure 1 shows, each Direct Connect private virtual interface is associated with a unique VLAN tag.

This network segregation goes from the VPC, across the Direct Connect network connection (cross connect), through the Direct Connect private virtual interface, through dedicated virtual routing and forwarding (VRF) instances, and then down to VLAN interfaces used by logical interfaces (LIFs) on the storage virtual machines (SVMs, formerly known as Vservers) on the NetApp storage cluster.

Figure 1) AWS Direct Connect network architecture.



Direct Connect connections come in two types: 1Gb Ethernet and 10Gb Ethernet. The connection from the VPC to the network switch or router in the Equinix colocation data center is a layer 2 connection from each AWS VGW used by the VPC. A cross-connect cable is patched from the AWS point of presence (PoP) in the Direct Connect data center to the customer network demarcation panel in the Direct Connect data center.

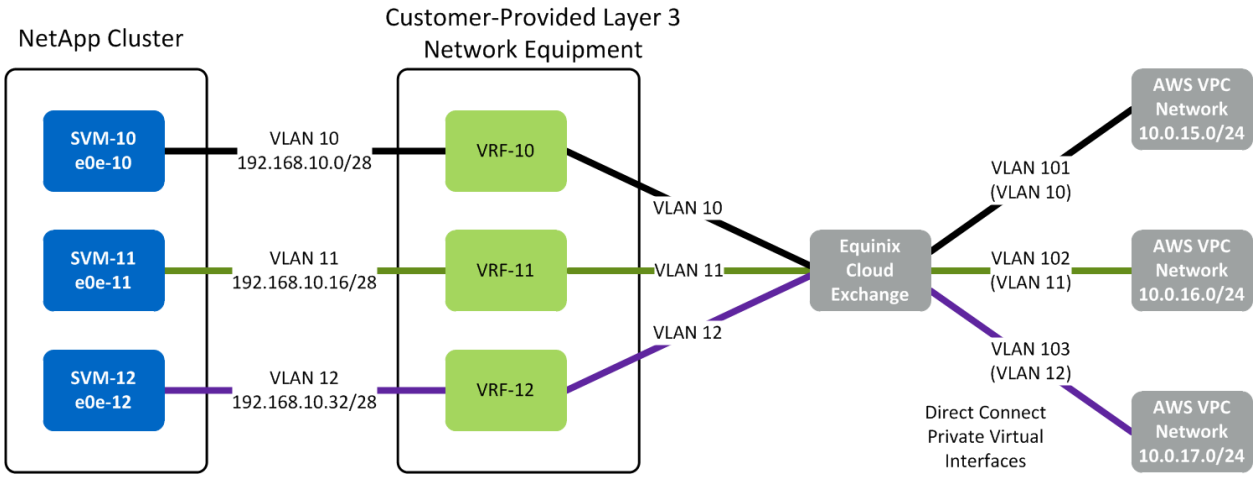
Customers who deploy the NetApp Private Storage for AWS solution at Equinix colocation data centers can also provision AWS Direct Connect network connections through the Equinix Cloud Exchange in

200Mb/sec or 500Mb/sec bandwidth sizes. 1Gb/sec and 10Gb/sec Direct Connect connections are provisioned manually by cross connect.

The Equinix Cloud Exchange performs 802.1Q VLAN ID translation, so the VLAN of the Direct Connect virtual network interface can be different from the VLAN number of the network in the Equinix colocation data center.

Figure 2 shows the AWS Direct Connect network architecture with the Equinix Cloud Exchange.

Figure 2) AWS Direct Connect network architecture with Equinix Cloud Exchange.



For more information about the Equinix Cloud Exchange, refer to the Equinix Cloud Exchange webpage.

NetApp recommends that redundant Direct Connect network connections be connected to the customer-provided redundant network equipment in the AWS Direct Connect data center (Equinix).

For more information about AWS Direct Connect, refer to the AWS Direct Connect User Guide.

Equinix Colocation Data Center (AWS Direct Connect Location)

AWS Direct Connect Point of Presence

AWS Direct Connect locations provide connectivity to the AWS cloud through AWS Direct Connect network connections. Equinix and other colocation providers have AWS Points of Presence (PoP) in their data centers, which offer private connectivity to AWS that does not go over the Internet.

Note: Each PoP connects to only a single AWS region. It is very important to deploy NetApp storage into the correct AWS Direct Connect location for the AWS region that you want to use.

A list of AWS Direct Connect locations can be found in the AWS Direct Connect FAQs documentation.

Most Equinix data centers are close to the AWS cloud; therefore, the latencies between Equinix and AWS can range from low to very low.

Because customers might experience varying latencies, NetApp recommends validating the latency of the network connectivity to AWS before deploying workloads into the NetApp Private Storage for AWS solution.

Physical Security

Equinix data centers offer a secure, highly available environment for the customer-owned NetApp storage and network equipment for the NetApp Private Storage for AWS solution. Equinix provides a high degree of physical security.

Customers have the option of deploying their storage into dedicated secure cages or into secure cabinets in shared cages.

For more information about Equinix physical security, see the Equinix Physical Security webpage, or contact your Equinix account team.

Operational Security

Equinix data centers have a minimum N+1 power and cooling system redundancy. Many Equinix data centers have N+2 power and cooling system redundancy.

For more information about Equinix operational reliability, refer to the Equinix Operational Reliability webpage, or contact your Equinix account team.

Equinix Cloud Exchange

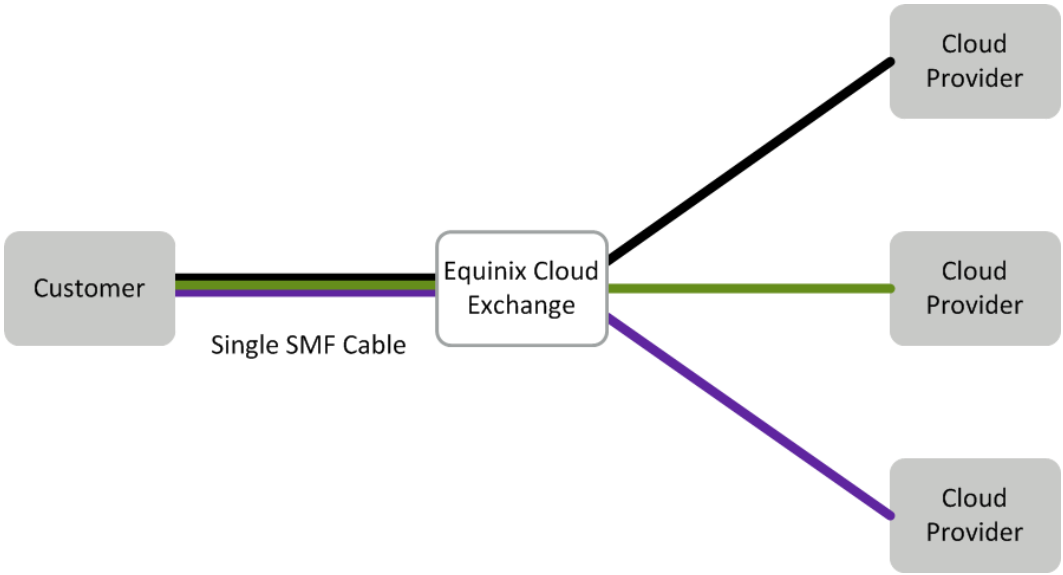
As Figure 3 shows, the Equinix Cloud Exchange allows customers to connect rapidly to multiple network and cloud service providers over just one single-mode fiber (SMF) optical cable. The dynamic connectivity of the Cloud Exchange provides the ability to quickly connect and disconnect cloud services as customers' technical and business requirements change.

Customers can use the Cloud Exchange portal to request connectivity to AWS through Direct Connect.

Note: Use of the Equinix Cloud Exchange with AWS Direct connect is only available when provisioning sub-1Gb/sec AWS Direct Connect network connections.

To purchase ports on the Equinix Cloud Exchange, contact your Equinix account team.

Figure 3) Equinix Cloud Exchange high-level architecture.



Border Gateway Protocol (BGP)

BGP is used to support network routing between the AWS VPC and the network in the Direct Connect data center over the AWS Direct Connect network connection.

The network in the Equinix colocation data center is directly connected to the customer-provided layer 3 network equipment. The BGP configuration advertises local network routes to the VPC network over the Direct Connect network connection. It also receives the BGP advertisements from the VPC network over the Direct Connect network connection.

Customer-Provided Layer 3 Network Equipment

The customer-provided network equipment is in the same AWS Direct Connect data center as the NetApp storage. NetApp does not certify specific network equipment to be used in the solution; however, the network equipment must support the following features:

- BGP
- At least one 9/125 SMF (1Gb/sec or 10Gb/sec) port
- 1000BASE-T Ethernet ports
- 802.1Q VLAN tags

The following features are optional:

- Equinix Cloud Exchange
- QinQ (stacked) VLAN tags
- VRF
- Redundant network switches
- Redundant 9/125 SMF (1Gb/sec or 10Gb/sec) ports
- 10GbE ports

Required Features

As noted in the previous section, “Border Gateway Protocol,” BGP is used to route network traffic between the local network in the Direct Connect data center and the AWS VPC network.

Direct Connect requires a minimum of one physical connection (9/125 SMF) from the customer-owned network equipment to AWS (or to the Equinix Cloud Exchange).

1000BASE-T network ports on the switch provide network connectivity from the NetApp storage cluster. Although these ports can be used for data, NetApp recommends using 1GbE ports for node management and out-of-band management.

802.1Q VLAN tags are used by Direct Connect private virtual interfaces (and the Equinix Cloud Exchange) to segregate network traffic on the same physical network connection.

Optional Features

The Equinix Cloud Exchange allows customers to connect quickly to the AWS cloud and to other clouds without the need to provision additional cross connects or use additional ports on the customer-provided network equipment.

The Equinix Cloud Exchange can use QinQ VLAN tags to support the routing of the network traffic from the network to AWS. The outer service tag (S-tag) is used to route traffic to AWS from the Cloud Exchange. The inner customer tag (C-tag) is passed on to AWS for routing to the AWS VPC through the Direct Connect network connection.

Redundant network switches protect against a loss of Direct Connect service caused by switch failure.

Note: For information about configuring redundant network switches, consult your network equipment vendor’s documentation.

Redundant 9/125 SMF ports protect against a loss of Direct Connect service caused by a port or cable failure.

Connecting 10GbE ports on the storage to the switch provides the highest amount of bandwidth capability between the switch and the storage to support data access.

NetApp FAS Storage and FlexArray

Both NetApp clustered Data ONTAP and Data ONTAP operating in 7-Mode can function with the NetApp Private Storage for AWS solution; however, NetApp highly recommends using clustered Data ONTAP with the solution.

2.2 Solution Architecture Diagrams

Figure 4 shows the architecture of the NetApp Private Storage for AWS solution.

Figure 4) NetApp Private Storage for AWS solution architecture.

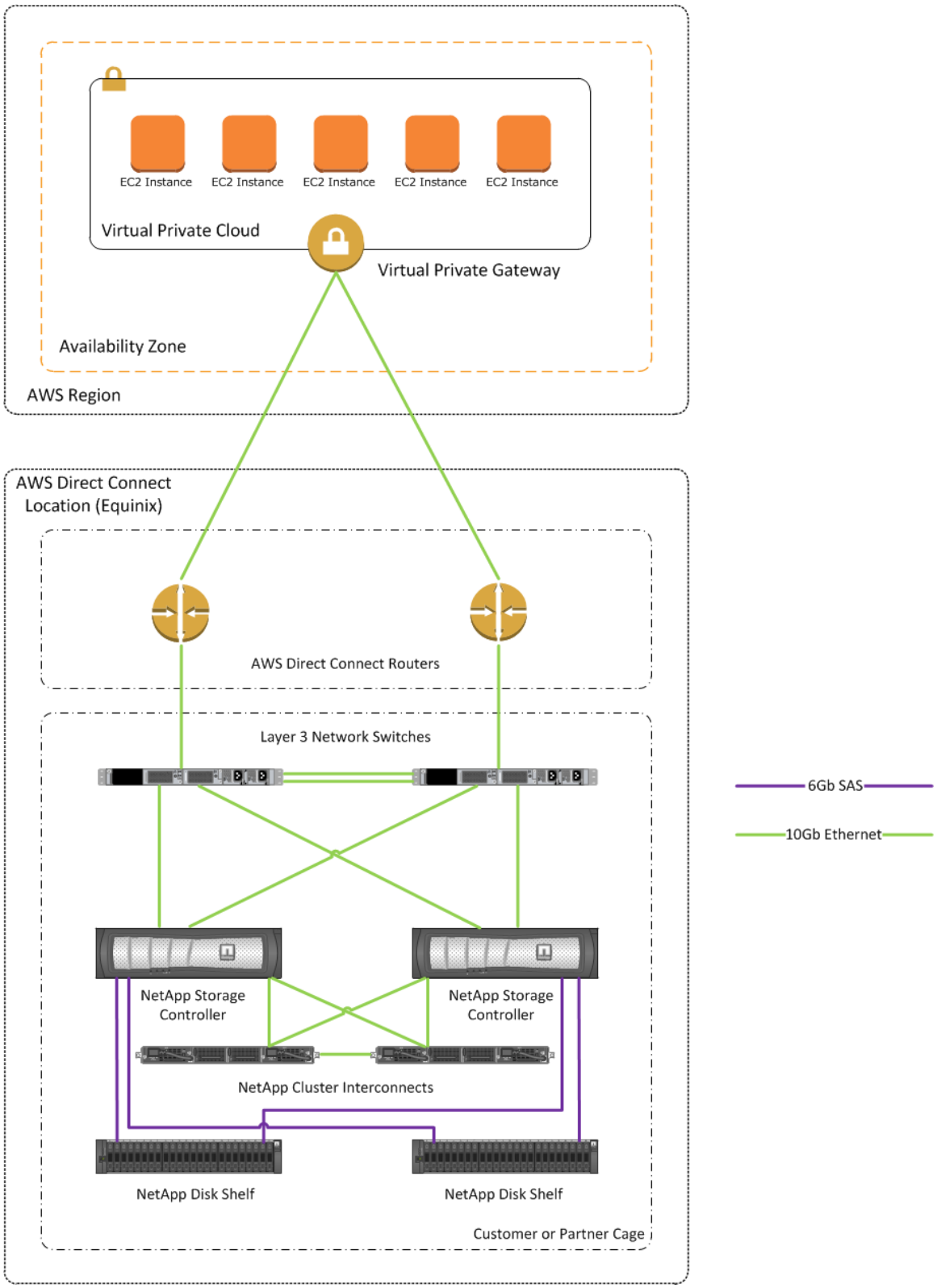
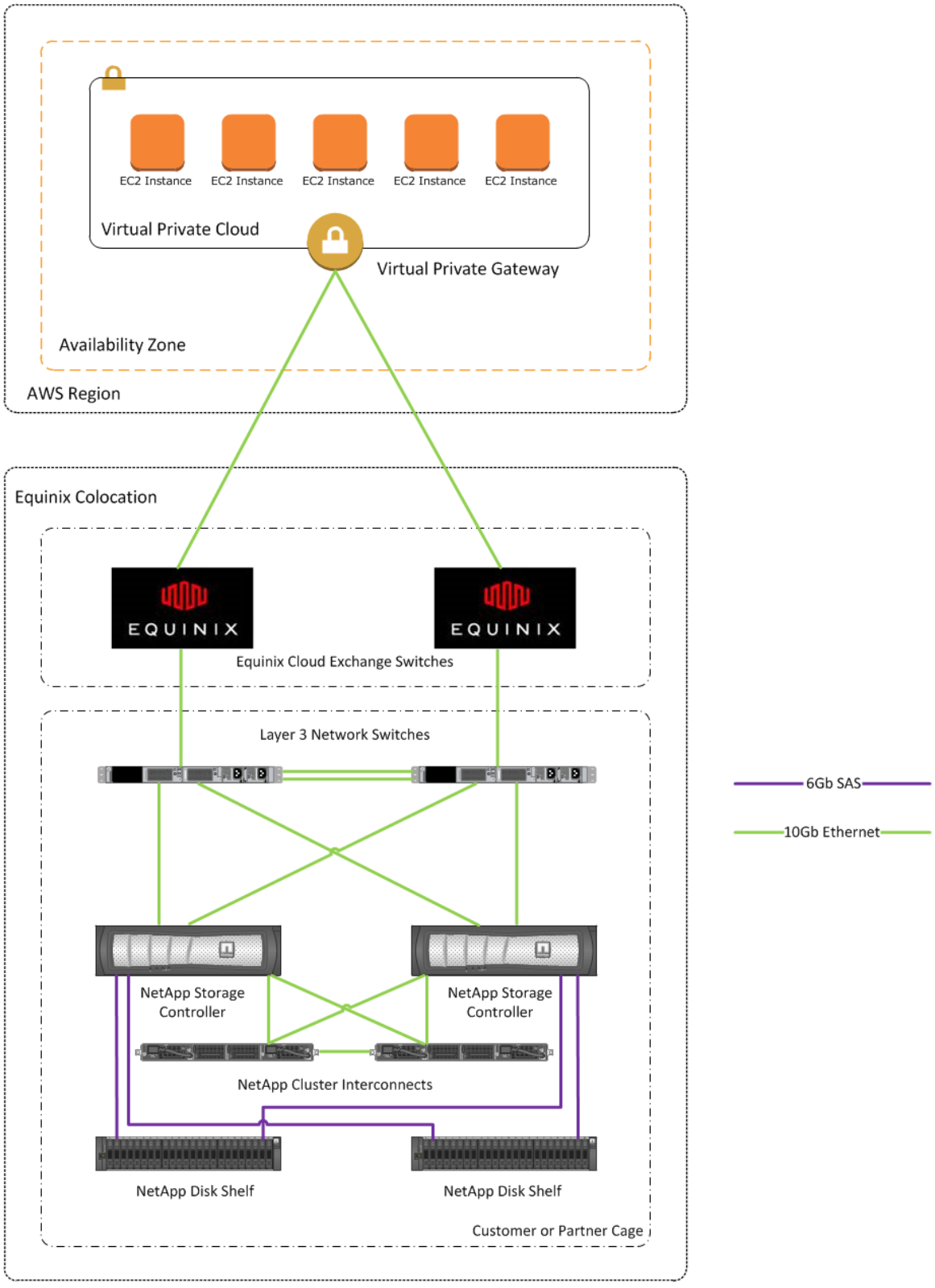


Figure 5 shows the architecture of the NetApp Private Storage for AWS solution with the Equinix Cloud Exchange.

Figure 5) NetApp Private Storage for AWS solution architecture with Equinix Cloud Exchange.



2.3 Solution Architecture Data Security Elements

NetApp Private Storage for AWS allows customers to store their data on NetApp storage that they own or control so that they can maintain the compliance and security of their data.

The solution contains the following security-related elements:

- AWS VPC
- AWS Direct Connect
- Physical security for the Equinix colocation data center
- NetApp storage encryption
- Third-party network security hardware and software

AWS Virtual Private Cloud

AWS VPC provides network isolation for the resources (VMs, services, and so forth) that are provisioned in it. EC2 VMs provisioned in a VPC can communicate with each other within the network. Resources external to the VPC do not have access to the resources in the VPC.

AWS VPCs can be accessed securely through a site-to-site VPN or through an AWS Direct Connect network connection at a Direct Connect data center such as Equinix.

AWS Direct Connect

AWS Direct Connect is a private, secure network connection that does not traverse the Internet. AWS connects to NetApp storage in the Equinix colocation data center through physical cross connects that are not shared with other customers.

The Equinix Cloud Exchange provides an additional layer of segregation by leveraging VLAN tags for Cloud Exchange virtual circuits to further isolate network traffic between Amazon customers and Equinix.

Physical Security for Equinix Colocation Data Center

Equinix provides state-of-the-art physical security at all of its data centers where AWS Direct Connect is available. The data centers have security guards and security systems to provide video surveillance. The security systems have biometric hand scanners combined with mantrap interlocks to restrict access to authorized personnel only.

For more information about Equinix Physical security, refer to the Equinix Physical Security webpage.

NetApp Storage Encryption

NetApp Storage Encryption software is the NetApp implementation of full-disk encryption that uses self-encrypting drives from leading vendors, allowing data on NetApp storage to be fully encrypted while maintaining storage efficiency and performance.

For more information, refer to NetApp Storage Encryption.

Third-Party Network Security Hardware and Software

Third-party security hardware and software devices can be used with the NetApp Private Storage for AWS solution, providing that the security solution can work in a TCP/IP environment. This qualifier exists because the connectivity to the cloud does not support Fibre Channel or Fibre Channel over Ethernet. NetApp does not certify third-party security solutions that can be used with the NetApp Private Storage for AWS solution. To implement a security solution with the NetApp Private Storage for AWS solution, contact your NetApp account team for further guidance.

3 NetApp Private Storage for AWS Deployment Overview

This section describes the standard deployment methodology for NetApp Private Storage for AWS. However, because no two customer environments are the same, NetApp has delivery partners who specialize in deploying NetApp Private Storage solutions. These partners are experienced and can help make your NetApp Private Storage for AWS deployment a success. For more information about NetApp Private Storage delivery partners, contact your NetApp account representative.

The high-level deployment workflow consists of the following phases and tasks:

1. Planning:
   a. Preinstallation and Site Preparation
2. Deployment:
   a. Installing the Equipment in the Equinix Data Center
   b. Setting Up AWS Virtual Private Cloud Network
   c. Setting Up the AWS Direct Connect
   d. Setting Up the Customer Network Switch
   e. Configuring NetApp Storage
3. Validation:
   a. Testing Connections and Protocols

3.1 Preinstallation and Site Preparation

The preinstallation and site preparation take place during the planning phase of the NetApp Private Storage for AWS workflow. This includes:

1. Establishing the colocation power and space requirements.
2. Ordering space and power.
3. Ordering the network, storage, and rack hardware.
4. Creating an IP address plan.
5. Obtaining AWS and Equinix customer portal accounts.
6. Ordering the Equinix Cloud Exchange port from Equinix. (sub-1Gbps Direct Connect connections only).
7. Creating an inbound shipment request through Equinix.
8. Installing the AWS Command Line Interface (CLI) tools

The final step of this phase involves validating that all preinstallation and site preparation tasks have been completed and the workflow is ready to move to the next phase.

Establishing Colocation Power and Space Requirements

Use the NetApp Hardware Universe or contact your NetApp account team to determine the power and space requirements for the NetApp storage you want to deploy with the NetApp Private Storage for AWS solution.

See the technical specifications or contact your network switch vendor about the power and space requirements of the network equipment that you want to deploy with NetApp Private Storage for AWS.

Ordering Space and Power

There are two types of colocation space at Equinix: shared and dedicated. A shared space is a secure cage containing secure cabinets used by multiple customers. Customers are required to use Equinix racks in a shared-space configuration.

A dedicated space is a secure cage that is assigned to a single customer. The smallest dedicated cage consists of five cabinets. Customers can use Equinix standard racks or use their own.

It is recommended that customers use redundant power connections connected to separate power distribution units (PDUs) so that the NetApp Private Storage solution can survive the loss of a single power connection.

The typical power connection configuration used with NetApp Private Storage is 208V/30A single-phase AC power. The voltage specifications may vary from region to region.

Contact your Equinix account team for more information about the available space and power options in the Equinix data center where you want to deploy NetApp Private Storage.

Ordering Network, Storage, and Rack Hardware

If you require more than six ports of power on a PDU, you need to purchase a third-party PDU, or order additional power connections from Equinix. Equinix sells PDUs that fit well with its cabinets. The Equinix cabinets are standard 42U, 4-post racks.

Contact your NetApp account team to make sure that you are ordering the appropriate rail kits for your cabinets.

If you are using a secure cabinet in a shared cage, you need to order a top-of-rack demarcation panel to connect the network equipment to AWS. The type of demarcation panel should be 24-port SC optical.

Creating an IP Address Plan

The creation of the IP address plan for NetApp Private Storage is critical. The data in Table 1 is used while configuring the NetApp Private Storage network. As a reminder, the unit of tenancy is an SVM connected to an AWS Virtual Private Cloud (VPC) network through an AWS Direct Connect (DX) private virtual interface.

Table 1) NetApp Private Storage IP address plan.

Tenant	Tenant VLAN	NetApp Private Storage SVM Network	BGP Peering Network	BGP Authentication Key	BGP ASN	AWS Network	AWS Region

The column headings are defined as follows:

- **Tenant**. The name or description of the NetApp Private Storage tenant.
- **Tenant VLAN.** The VLAN number that the NetApp Private Storage tenant uses to connect the NetApp storage assigned to them to the AWS VPC over a Direct Connect (DX) private virtual interface (for example, 100).

- **NetApp Private Storage SVM Network**. The network CIDR that is used by the NetApp SVM logical network interfaces. The network is typically a private network CIDR (for example, 192.168.100.0/28), but can be a public network CIDR if you are using a DX public virtual interface.
- **BGP Peering Network**. A network that is a /30 network. (for example, 169.254.253.0/30). The lower IP address number (for example, 169.254.253.1/30) is assigned to the layer 3 interface on the network equipment in Equinix and the higher number (for example, 169.254.253.2/30) is assigned to the AWS Virtual Private Gateway.
- **BGP Authentication Key**. A text string that represents a shared key between the network equipment in Equinix and AWS. This key securely establishes the BGP session. The BGP key used in our example is eea0a828f3e5fe02687cce9c.
- **BGP Autonomous System Number** (**BGP ASN**). A unique number assigned to the network equipment in Equinix. The ASN can be a private or public number. Private ASN numbers range from 64512 to 65535.

  Note: If you are using multiple clouds with NetApp Private Storage, avoid the use of 64514 and 64515 because these ASNs are used by SoftLayer and Azure respectively. AWS does not have any restrictions on the private ASN that you can use.

- **AWS Network.** The network CIDR of the AWS VPC (for example, 10.10.100.0/24).
- **AWS Region.** The AWS region in which the VPC is created and connected through AWS DX.

Obtaining AWS and Equinix Customer Portal Accounts

If you do not have an AWS account, go to https://aws.amazon.com to create one.

Contact your Equinix account team to get your account set up in the Equinix Customer Portal.

Ordering Equinix Cloud Exchange Port from Equinix

This action is only needed if you are connecting to AWS with sub-1Gbps Direct Connect network connections.

Contact your Equinix account team to order a Cloud Exchange port.

Creating an Inbound Shipment Request Through Equinix

Equinix physical security procedures require that there be an inbound shipping request for any shipments sent to an Equinix data center. The shipping addresses for the data center (also known as IBX), can be found in the Equinix Customer Portal.

In the inbound shipment request, make sure to provide the shipper, shipment tracking number, number and weight of items in the shipment, and date on which the shipment is expected to arrive at the IBX.

When shipping equipment to the Equinix data center, the format of the address should be as follows:

Name of cage/suite
c/o Equinix
Address of the data center

For more information about Equinix shipping and receiving procedures for your IBX, see the Equinix Customer Portal or contact your Equinix Client Services manager.

Installing AWS Command Line Interface (CLI) Tools

See the AWS CLI Getting Set Up Documentation for instructions on setting up and installing AWS CLI Tools on Windows, Linux, or Mac.

3.2 Installing the Equipment in the Equinix Data Center

You can begin to install the equipment in the data center after the preinstallation and site preparation phase is complete.

Perform the following steps to set up the data center:

1. Set up security access to the Equinix data center and cage.
2. Make sure that all required materials (hardware, software, accessories, and so on) are available onsite.
3. Install the NetApp storage in the rack.
4. Install the customer-provided network equipment in the rack.

Setting Security Access to Equinix Data Center and Cage

Use the Equinix Customer Portal to create a security access request for the Equinix IBX where the NetApp Private Storage solution is being deployed. The security access registration process includes a biometric scan, PIN assignment, and security card assignment (depending on the IBX). You need to bring a government-issued identification to the IBX.

Note: It is vital that the name on the security access request is identical to the government-issued identification, or Equinix security will not process the request.

After the security access process is complete, you are able to visit the Equinix IBX without the need for an Equinix work visit request.

Verifying the Availability of Required Materials Onsite

The shipment can be inventoried in person, or the Equinix SmartHands technicians can inventory the shipment. If you want to have the Equinix SmartHands technicians inventory the shipment, use the Equinix Customer Portal to create a SmartHands request.

Installing NetApp Storage in the Rack

If you are using an Equinix cabinet in a shared cage, the NetApp storage can be installed in person, or you can have a NetApp Partner install the storage.

If you are using a dedicated Equinix cage, the racks in the cage must be installed. Use the Equinix Customer Portal to create an Equinix SmartHands request to have the racks installed.

If you are having a NetApp partner install the storage, use the Equinix Customer Portal to create a work visit request for the partner engineers. The engineers need to bring a government-issued identification and the names on the work visit request must match the government-issued identification.

Due to Equinix safety rules, the power distribution units (PDUs) in the rack need to be connected to Equinix power by an Equinix SmartHands technician. Use the Equinix Customer Portal to create a SmartHands request to connect the PDUs.

Installing the Customer-Provided Network Equipment in the Rack

The network equipment can be installed at the same time as the NetApp storage.

If the network equipment is to be installed at a different time, use the Equinix Customer Portal to create a work visit request for the partner engineers. The engineers need to bring a government-issued identification and the names on the work visit request must match the government-issued identification.

3.3 Setting Up AWS Virtual Private Cloud Network

To set up the AWS virtual private cloud (VPC) network, complete the following steps:

Note: It is required to have an AWS account and the AWS CLI Tools installed on an Internet-connected computer.

Note: Obtain the information from the NetApp Private Storage IP Address plan in Table 1

1. On the computer where the AWS CLI Tools is installed, open a shell session and type the following command to configure the AWS CLI profile:

```
aws configure
AWS Access Key ID [None]: <<aws_access_key>>
AWS Secret Access Key [None]: <<aws_secret_access_key>>
Default region name [None]: <<aws_region>>
Default output format [None]: <<format>>
```

   Where:

   - `<<aws_access_key>>` is the access key of the AWS IAM account used to deploy the AWS resources.
   - `<<aws_secret_access_key>>` is the secret access key of the AWS IAM account used to deploy the AWS resources.
   - `<<aws_region>>` is the default region where the VPC will be deployed. You can also set this value to `None` and specify the region on every CLI command.
   - `<<format>>` is the output format from each command. The value text generates output in a formatted text table and the value `json` with generate output in a JSON format. The default value is `json`.

2. Run the following command to create the VPC:

```
aws ec2 vpc-create –cidr-block <<10.10.100.0/24)
```

   Where:

   - `<<cidr_block>>` is the CIDR of the network (i.e., `10.10.100.0/24`).

   The output of the command is as follows:

```
{
    "Vpc": {
        "InstanceTenancy": "default",
        "State": "pending",
        "VpcId": "vpc-1843057d",
        "CidrBlock": "10.10.100.0/24",
        "DhcpOptionsId": "dopt-1aa6e972"
    }
}
```

   Note: The value of `State` will change from `pending` to `available` in a few seconds.

3. Run the following command to create the subnet:

```
aws ec2 subnet-create –vpd-id <<vpc-id>> --cidr-block <<cidr-block>>
```

Where:

- `<<vpc-id>>` is the value of `VpcId` parameter from step 2.
- `<<cidr-block>>` is the CIDR block of the subnet (i.e., `10.10.100.0/24)` that creates a subnet the same size as the VPC).

Note: You can have multiple subnets in the VPC. This sample deployment uses a single subnet the size of the VPC network.

The output of the command is as follows:

```
{
    "Subnet": {
        "VpcId": "vpc-1843057d",
        "CidrBlock": "10.10.100.0/24",
```

```
        "State": "pending",
        "AvailabilityZone": "us-west-1a",
        "SubnetId": "subnet-78bb881d",
        "AvailableIpAddressCount": 251
    }
}
```

4. Run the following command to create an Internet gateway (IGW) that will be attached to the VPC:

```
aws ec2 create-internet-gateway
```

The output of the command is as follows:

```
{
    "InternetGateway": {
        "Tags": [],
        "InternetGatewayId": "igw-275ee342",
        "Attachments": []
    }
}
```

5. Run the following command to attach the IGW to the VPC:

```
aws ec2 attach-internetgateway --internet-gateway-id <<igw-id>> --vpc-id <<vpc-id>>
```

Where:

- `<<igw-id>>` is the value of the `InternetGatewayId` parameter from step 4.
- `<<vpc-id>>` is the value of the `VpcId` parameter from step 2.

6. Run the following command to create the virtual private gateway (VGW) for the VPC:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

The output of the command is as follows:

```
{
    "VpnGateway": {
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-96c498d3",
        "VpcAttachments": []
    }
}
```

7. Run the following command to attach the VGW:

```
aws ec2 attach-vpn-gateway --vpn-gateway-id <<vgw-id>> --vpc-id <<vpc-id>>
```

Where:

- `<<vgw-id>>` is the value of the `VpnGatewayId` parameter from step 6.
- `<<vpc-id>>` is the value of `VpcId` parameter from step 2.

8. Run the following command to get information about the security group for the VPC:

```
aws ec2 describe-security-groups --filters Name=vpc-id,Values=<<vpc-id>>
```

Where:

- `<<vpc-id>>` is the value of `VpcId` parameter from step 2.

The output of the command is as follows:

```
{
    "SecurityGroups": [
        {
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
```

```
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "UserIdGroupPairs": [],
                    "PrefixListIds": []
                }
            ],
            "Description": "default VPC security group",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [],
                    "UserIdGroupPairs": [
                        {
                            "UserId": "666029239484",
                            "GroupId": "sg-74ea2210"
                        }
                    ],
                    "PrefixListIds": []
                }
            ],
            "GroupName": "default",
            "VpcId": "vpc-1843057d",
            "OwnerId": "666029239484",
            "GroupId": "sg-74ea2210"
        }
    ]
}
```

9. Run the following commands to configure the VPC security group for the VPC:

```
aws ec2 authorize-security-group-ingress --group-id <<sg-id>> --protocol tcp --port 22 --cidr
<<source-network>>

aws ec2 authorize-security-group-ingress --group-id <<sg-id>> --protocol tcp --port 3389 --cidr
<<source-network>>
aws ec2 authorize-security-group-ingress --group-id <<sg-id>> --ip-permissions
'[{\"IpProtocol\":\"-1\",\"IpRanges\":[{\"CidrIp\":\"<<svm-cidr>>\"}]}]'
```

Where:

- `<<sg-id>>` is the value of the `GroupId` parameter from step 8.
- `<<source network>>` is the network CIDR where you are accessing the VPC (`0.0.0.0/0` if you don't have a specific IP address).
- `<<svm-cidr>>` is the NetApp SVM network CIDR (i.e., `192.168.100.0/28`).

Note: The rules to open up SSH or RDP from the Internet is optional. These rules are for ease of administration. It is an AWS best practice not to open up ports to the entire Internet (0.0.0.0/0).

Note: The quotes and escape characters for the `--ip-permissisons` parameter depends on the shell you use. See Quoting Strings in AWS CLI.

10. Run the following command to determine the route table ID for the VPC:

```
aws ec2 describe-route-tables --filters Name=vpc-id,Values=<<vpc-id>>
```

Where:

- `<<vpc-id>>` is the value of the `VpcId` parameter from step 2.

The output of the command is as follows:

```
{
    "RouteTables": [
        {
            "Associations": [
                {
                    "RouteTableAssociationId": "rtbassoc-7c878919"
```

```
                    "Main": true,
                    "RouteTableId": "rtb-d6c29ab3"
                }
            ],
            "RouteTableId": "rtb-d6c29ab3",
            "VpcId": "vpc-1843057d",
            "PropagatingVgws": [],
            "Tags": [],
            "Routes": [
                {
                    "GatewayId": "local",
                    "DestinationCidrBlock": "10.10.100.0/24",
                    "State": "active",
                    "Origin": "CreateRouteTable"
                }
            ]
        }
    ]
}
```

11. Run the following commands to configure the route table:

```
aws ec2 create-route --route-table-id <<rt-id>> --destination-cidr-block 0.0.0.0/0 --gateway-id
<<igw-id>>

aws ec2 enable-vgw-route-propagation --route-table-id <<rt-id>> --gateway-id <<vgw-id>>
```

Where:

- `<<rt-id>>` is the value of the `RouteTableId` parameter from step 10.
- `<<igw-id>>` is the value of the `InternetGatewayId` parameter from step 4.
- `<<vgw-id>>` is the value of the `VpnGatewayId` parameter from step 6.

Note: The route to the Internet (0.0.0.0/0) allows for the EC2 virtual machine instances to access the Internet.

3.4 Setting Up the AWS Direct Connect

To set up the AWS Direct Connect (DX) network connection complete the following steps.

Note: Obtain the information from the NetApp Private Storage IP Address plan in Table 1.

1. Run the following command to create the Direct Connect network connection:

```
aws directconnect create-connection --location <<location>> --bandwidth <<bw>> --connection-name
<<connection-name>>
```

Where:

- `<<location>>` is the location identifier for the Equinix data center (i.e., `EqSV5`, which is the Equinix SV5 data center that has an AWS PoP that connects to the us-west-1 region).
- `<<bw>>` is the bandwidth of the connection. Values can be `1Gbps` or `10Gbps`.
- `<<connection-name>>` is the connection name.

The output of the command is as follows:

```
{
    "ownerAccount": "666029239484",
    "connectionId": "dxcon-ffsdnve4",
    "connectionState": "requested",
    "bandwidth": "10Gbps",
    "location": "EqSV5",
    "connectionName": "NPS",
    "region": "us-west-1"
}
```

Note: After the Direct Connect creation request is made, AWS sends a Letter of Authorization (in pdf format) to the e-mail address associated with the AWS account used to make the request.

Use the Equinix Customer Portal to create a cross-connect request to AWS for 1Gbps or 10Gbps connections. The Letter of Authorization sent to you from AWS is used by Equinix to patch a cross connect from the AWS PoP to the demarcation panel in the cage.
Contact your Equinix Client Services manager if you have any questions on how to submit a cross-connect request.

2. Patch a single-mode fiber (SMF) duplex cable from the demarcation panel where the cross connect is patched to the network equipment in the cage/cabinet.
3. After the cross connect is patched, schedule a network turn up using the Equinix Customer Portal or through your Equinix Client Services Manager.
4. Run the following command to verify that the Direct Connect connection is turned up correctly:

```
aws directconnect describe-connections --connection-id <<conn-id>>
```

Where:

- `<<conn-id>>` is the value of the `connectionId` parameter from step 1.

The output of the command is as follows:

```
{
    "connections": [
        {
            "ownerAccount": "666029239484",
            "connectionId": "dxcon-ffsdnve4",
            "connectionState": "available",
            "bandwidth": "10Gbps",
            "location": "EqSV5",
            "connectionName": "NPS",
            "region": "us-west-1"
        }
    ]
}
```

Note: The `connectionName` parameter value appears as `available` if the Direct Connect network connection has been provisioned correctly. If the value does not appear as `available`, check the cross-connect patch cable and the network equipment port configuration. Troubleshooting this connection may involve contacting either Equinix support, AWS support, or both.

5. Run the following command to create a Direct Connect private virtual interface:

```
aws directconnect --connection-id <<conn-id>> --new-private-virtual-interface
virtualInterfaceName=<<pvi-name>>,vlan=<<vlan>>,asn=<<asn>>,authkey=<<bgp-
key>>,amazonAddress=<<aws-peer-ip>>,customerAddress=<<cust-peer-ip>>,virtualGatewayId=<<vgw-id>>
```

Where:

- `<<conn-id>>` is the value of the `connectionId` parameter from step 4.
- `<<pvi-name>>` is the name of the private virtual interface (for example, `NPS-PVI`).
- `<<vlan>>` is the VLAN number of the private virtual interface (i.e., 100).
- `<<asn>>` is the autonomous system number of your network equipment in Equinix (i.e., 64514).
- `<<bgp-key>>` is the BGP authentication key (i.e., eea0a828f3e5fe02687cce9c).
- `<<aws-peer-ip>>` is the AWS BGP peer IP address (i.e., 169.254.253.2/30).
- `<<cust-peer-ip>>` is the your BGP peer IP address (i.e., 169.254.253.1/30).
- `<<vgw-id>>` is the value of the `VpnGatewayId` parameter from step 6 of section 3.3.

The output of the command is as follows:

```
{
    "virtualInterfaceState": "pending",
    "asn": 64514,
    "vlan": 100,
    "customerAddress": "169.254.253.1/30",
    "ownerAccount": "666029239484",
    "connectionId": "dxcon-ffsdnve4",
    "virtualGatewayId": "vgw-94c79bd1",
    "virtualInterfaceId": "dxvif-fgs4p5ka",
    "authKey": "eea0a828f3e5fe02687cce9c",
    "routeFilterPrefixes": [],
    "location": "EqSV5",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<logical_connection
id=\"dxvif-fgs4p5ka\">\n  <
vlan>100</vlan>\n  <customer_address>169.254.253.1/30</customer_address>\n
<amazon_address>169.254.253.2/30</amazon_add
ress>\n  <bgp_asn>64514</bgp_asn>\n  <bgp_auth_key>eea0a828f3e5fe02687cce9c</bgp_auth_key>\n
<amazon_bgp_asn>7224</amaz
on_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "169.254.253.2/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "delete-me"
}
```


Note: The value of the `virtualInterfaceState` parameter will initially appear as `pending` and after a few seconds will appear as `down` until your network equipment in Equinix is configured and the BGP session is established.

3.5 Setting Up the Customer Network Switch

Note: Obtain the information from the NetApp Private Storage IP Address plan in Table 1.

The customers can use any brand or model layer-3 network switch that meets the following requirements:

- Has Border Gateway Protocol BGP licensed and enabled
- Has at least one 9/125 single-mode fiber (SMF) 1Gbps or 10Gbps port available
- Has 1000BASE-T Ethernet ports
- Supports 802.1Q VLAN tags

The steps to set up the customer-provided network switch are as follows:

1. Perform the initial switch configuration (host name, SSH, user names, and so on).
2. Create and configure the virtual local area network (VLAN) interface.
3. Create and configure the virtual routing and forwarding (VRF) instances.

Note: See your switch manufacturer's documentation for specific configuration commands.

Sample Switch Configuration Commands

The following are commands for a Cisco Nexus switch running Cisco NX-OS:

```
config t

vrf-context <<vrf_name>>

vlan <<vlan>>

interface vlan <<vlan>>
no shutdown
vrf member <<vrf-name>>
ip address <<cust-peer-address>>/30
ip address <<local-subnet-gateway-address>>/<<cidr>> secondary
exit

router bgp <<asn>>
```

```
vrf <<vrf-name>>
address-family ipv4 unicast
network <<local-subnet>>/<<cidr>>
exit
neighbor <<aws-peer-address>> remote-as 7224
password 0 <<bgp-key>>
address-family ipv4 unicast
exit
end

copy running-config startup-config
```

Where:

- `<<vrf-name>>` is the VRF name.

Note: As a good-practice naming convention, NetApp recommends embedding the VLAN ID into the VRF name (for example, `vrf-100`).

- `<<vlan>>` is the VLAN number of the VLAN used by the AWS Direct Connect virtual interface.
- `<<cust_peer_address>>` is the local peer address for the AWS Direct Connect virtual interface.
- `<<local-subnet-gateway-address>>` is the gateway address of the local subnet used by the NetApp SVM.
- `<<cidr>>` is the CIDR number for the local SVM subnet.
- `<<asn>>` is the ASN of the local network, which can be private or public.

Note: The ASN of AWS is `7224`.

- `<<local-subnet>>` is the local subnet network.
- `<<aws-peer-address>>` is the AWS peer address for the AWS Direct Connect virtual interface.
- `<<bgp-key>>` is a string used to secure the BGP-peering session.

Sample Switch Configuration

The following is a sample switch configuration for a Cisco Nexus switch running Cisco NX-OS:

```
vrf context vrf-100

vlan 100

interface vlan 100
   no shutdown
  vrf member vrf-100
  no ip redirects
  ip address 169.254.253.1/30
  ip address 192.168.100.1/28 secondary

router bgp 64514
vrf vrf-100
  address-family ipv4 unicast
  network 192.168.100.0/28
neighbor 169.254.243.2 remote-as 7224
  address-family ipv4 unicast
```

3.6 Configuring NetApp Storage

Note: Obtain the information from the NetApp Private Storage IP Address plan in Table 1.

To steps to configure the NetApp storage are as follows:

1. Create VLAN interface ports on cluster nodes.

2. Create a storage virtual machine (SVM) on cluster.
3. Create logical interfaces (LIFs) on the SVM that uses the VLAN interface ports:
   a. Management LIF
   b. CIFS/NFS LIF
   c. iSCSI LIFs

3.7 Testing Connections and Protocol Access

Perform the procedures listed in this section to verify and test the Direct Connect network connection and in the NetApp Private Storage AWS environment.

Preparing AWS Virtual Machine Instance

An AWS virtual machine needs to be created in the VPC subnet connected to the Direct Connect network connection.

If you know how to deploy an AWS virtual machine, you can skip to step 6.

1. Run the following command to create an AWS key pair:

```
aws ec2 create-key-pair --key-name <<nps-key-pair>>
```

   Where:

   – `<<nps-key-pair>>` is the name of the key pair that you are creating (i.e., `nps-key-pair`)

   The output of the command appears as follows:

```
{
    "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAk/YCimc9QUUqHAMXKnls/+1f+S7TS/9ydKVm5KFVGt8noUqzlJy
RPYiDzjbe\nxkZJouB3FcjyIWdYNvjFJtMYEXLD38oScMM8JRLfoiaiqS4oYniR0pd7SC/pwhYlP2mW2PbostGv\n3zPAmDYR
2BYpUA+isar0DVpt7/kdymI
onwR1EA1umnHINQ+56VKR2hE76ZEbSPsN0kFczgalVjdU\nR5fn3mnGALq604S6XcqgINT+vpR57CJ2KoG3u4NVrwC+Nxid4g
6yx7X8XziPKTPFaJ9etxAe+
ln2\nJVViD6/Z1LkbJRey9JniTW2LW17954izPwG0AkE2V4FxEtGcYpL63QIDAQABAoIBAQCRJnHvnL/2\n3AcnW9DKFjzcBN
518X3PvFVoISsDd1Gp/uk9z
a7GR6IkH1xOXH3HPQTYCCRK9sqyAlQD+aBPKh8e\nXtvRGsdkX+MM66r/w68nEOUFVQqY8Yo7kuVbRaZcwxT825wEcLh72qVB
rSlsNrFonS4Z5X9t6hyX\no
mebNxhqhdLKMN6WFNCj3P2s9X2eB7q+R2+Qdx81yG33m3MBljo4DKd0Y+qZDBGO+JSnGr5EkeJ8\nSzoCKVV6BdfkqPj5U3Ng
l0TDLh47B94wikwepui46sN
nkw8Crc/F/5JqzRwRDJQC1R+r3wn7JKwQ\n2XTLETVVHZ49vkuXEJx5RMESxqwBAoGBAMmNakitKwwDI4ix0mokeAhb1SJFsn
e02UDqNYDHBUjj\ne40FcuX
Z6uEAQsdFH1q6y9CrTug5boyAVI/lYtRk21tgXqk42WGN1H/OizKBHECa7Gf2dkwAamgT\nsm8GXZ8kM4hAaip/PYqFz3l7vU
w6wuNYSdZY/+sgNkaJrXQEg
O1BAoGBALvuaz14vQhTMDY10M0F\nY5Y/ULoFR5mlqptGzLh02MXHkMUL+2s9vJWYqxp0PNdoO7SKg0tXbQWKRtKlqBqUv3rv
yRTO+fdo\nRM331ipGUV6L+
BI/4d58H1vqG4SB2F+uHTrJHnrOBHLQ0BnIi/6V/Rgja+UiNH+1Adq/i0sz0Pqd\nAoGASUZCR529MqbJel67kp8jjFZmjAku
stpqyXDcdq7kOvWI62H5vZq
WC6renDeDCKc0he5hx4dp\nq2bUV2Qbo//ux6+nenoSfabkhHqAnKHXtZ0dKUavBz53cNMIYdi8nKi6m2RftIeBxiSToyTbUy
Vh\nyOlGHslIcSDK2qYfCzi
/7oECgYB1lNpr/5rmFoZXpJwban/AhaTQfnUp4Mk5lbaIwjLOq+ocHz8S\n/fF796U2S/u1GnYgo3k6Llz71mQxb4wZ9W9IT7
aT2lvzO72w0Juin5RRf2mao
XWEakyEiZjl8weZ\n9ao8GZGEbFnELHnHZ/6PLvDHpYHgPuJx9KJQZv+T8buS0QKBgBDpCp3iW6gVd0VpW2MQyiNpdQ/B\nA5
YbLtSCryvnBhZ+//WDNBh/T
Nmxcm3VLEEC+TSwE3BSONN9k/fMaAwp3bKM/DSocH3DMr34t0EY\nZER3BOrZwIzNESKw9Yw1/S42qoiHoGUDCT7btxVaO2fg
aJDuE+ClGVUzinzL1TKKC+2
e\n-----END RSA PRIVATE KEY-----",
    "KeyName": "nps-key-pair",
    "KeyFingerprint": "2e:13:6d:08:6b:c0:aa:dd:03:95:20:d8:7f:e7:04:bf:12:32:f2:97"
}
```

2. Copy the value of the parameter `KeyMaterial` to a text file with a `.pem` extension. Do not include the quotes.
3. Run the following command to create an elastic IP address:

```
aws ec2 allocate-address --domain vpc
```

The output of the command appears as follows:

```
{
    "PublicIp": "52.9.185.69",
    "Domain": "vpc",
    "AllocationId": "eipalloc-ef1ec68a"
}
```

4. Provision an AWS virtual machine in the VPC subnet by using either the AWS EC2 dashboard, or running the following AWS CLI command:

```
aws ec2 run-instances --image-id <<image-id>> --count 1 --instance-type t2.micro --key-name
<<nps-key-pair>> --security-group-ids <<sg-id>> --subnet-id <<subnet-id>>
```

Where:

- `<<image-id>>` is the image identifier from the AWS marketplace. (i.e., `ami-06116566` for an Ubuntu Server 14.04 LTS 64-bit virtual machine, or `ami-df4437bf` for a Windows Server 2012 R2 Base virtual machine).
- `<<instance-type>>` is the instance type that you want to use for the virtual machine (for example, `t2.micro`).
- `<<nps-key-name>>` is the AWS key pair created in step 1.
- `<<sg-id>>` is the value of the `GroupId` created in step 8 of section 3.3.
- `<<subnet-id>>` is the value of the `SubnetId` parameter in step 3 of section 3.3.

The output of the command appears as follows:

```
{
    "OwnerId": "666029239484",
    "ReservationId": "r-8a196b4a",
    "Groups": [],
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "",
            "RootDeviceType": "ebs",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2016-04-15T19:00:45.000Z",
            "PrivateIpAddress": "10.10.100.240",
            "ProductCodes": [],
            "VpcId": "vpc-1843057d",
            "StateTransitionReason": "",
            "InstanceId": "i-7a79eab9",
            "ImageId": "ami-06116566",
            "PrivateDnsName": "ip-10-10-100-240.us-west-1.compute.internal"
            "KeyName": "nps-key-pair",
            "SecurityGroups": [
                {
                    "GroupName": "default",
                    "GroupId": "sg-74ea2210"
                }
            ],
            "ClientToken": "",
            "SubnetId": "subnet-78bb881d",
            "InstanceType": "t2.micro",
            "NetworkInterfaces": [
                {
                    "Status": "in-use",
                    "MacAddress": "02:c5:05:94:fa:65",
```

```
                    "SourceDestCheck": true,
                    "VpcId": "vpc-1843057d",
                    "Description": "",
                    "NetworkInterfaceId": "eni-b720c5d1",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateIpAddress": "10.10.100.240"
                        }
                    ],
                    "Attachment": {
                        "Status": "attaching",
                        "DeviceIndex": 0,
                        "DeleteOnTermination": true,
                        "AttachmentId": "eni-attach-33629df8",
                        "AttachTime": "2016-04-15T19:00:45.000Z"
                    },
                    "Groups": [
                        {
                            "GroupName": "default",
                            "GroupId": "sg-74ea2210"
                        }
                    ],
                    "SubnetId": "subnet-78bb881d",
                    "OwnerId": "666029239484",
                    "PrivateIpAddress": "10.10.100.240"
                }
            ],
            "SourceDestCheck": true,
            "Placement": {
                "Tenancy": "default",
                "GroupName": "",
                "AvailabilityZone": "us-west-1a"
            },
            "Hypervisor": "xen",
            "BlockDeviceMappings": [],
            "Architecture": "x86_64",
            "StateReason": {
                "Message": "pending",
                "Code": "pending"
            },
            "RootDeviceName": "/dev/sda1",
            "VirtualizationType": "hvm",
            "AmiLaunchIndex": 0
        }
    ]
}
```


5. Run the following command to associate the AWS elastic IP address with the virtual machine created in step 4.

```
aws ec2 associate-address --instance-id <<instance-id>> --allocation-id <<eip-id>>
```

   Where:

   - <<`instance-id`>> is the value of the `InstanceId` parameter from step 4.
   - <<`eid-id`>> is the value of the `AllocationId` parameter from step 3.
   - The output of the command appears as follows:


```
{
    "AssociationId": "eipassoc-1b43937f"
}
```


6. Log in to the AWS virtual machine provisioned in step 4. Use an SSH client to connect to a Linux virtual machine or an RDP client to connect to a Windows virtual machine.

   - See the AWS EC2 Documentation on how to connect to your Linux instance.
   - See the AWS EC2 Documentation on how to connect to your Windows instance.

Testing Network Connectivity

1. Use the ping utility on the AWS virtual machine instance to verify network connectivity. On the VM, run the following command to ping the SVM network gateway on your switch in Equinix:

```
ping <<svm-gateway>>
```

   Where:

   – <<svm-gateway>> is the IP address of the layer 3 interface on your switch in Equinix (i.e., 192.168.100.1).

   The output of the command appears as follows:

```
Pinging 192.168.100.1 with 32 bytes of data:
Reply from 192.168.100.1: bytes=32 time=2ms TTL=251
Reply from 192.168.100.1: bytes=32 time=1ms TTL=251
Reply from 192.168.100.1: bytes=32 time=1ms TTL=251
Reply from 192.168.100.1: bytes=32 time=1ms TTL=251

Ping statistics for 192.168.100.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

   Note: On the first ping attempt, there may be one or two dropped packets, after which there should be no dropped packets.

   Note: The output of the ping command varies on the operating system used.

2. On the VM, run the following command to ping the NetApp SVM LIF:

```
ping <<svm-lif>>
```

   where

   – <<svm-lif>> is the IP address of the network interface on the NetApp SVM (i.e., 192.168.100.2).

   The output of the command appears as follows:

```
Pinging 192.168.100.2 with 32 bytes of data:
Reply from 192.168.100.2: bytes=32 time=2ms TTL=251
Reply from 192.168.101.2: bytes=32 time=1ms TTL=251
Reply from 192.168.100.2: bytes=32 time=1ms TTL=251
Reply from 192.168.100.2: bytes=32 time=1ms TTL=251

Ping statistics for 192.168.100.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

   Note: The output of the ping command varies depending on the operating system used.

Testing iSCSI Protocol Connectivity

1. Use the iSCSI software initiator on your AWS virtual machine instance to establish iSCSI sessions to the iSCSI LIFS created in section 3.6.

   Note: Refer to the documentation of the operating system of the AWS virtual machine instance on how to establish an iSCSI session

   Note: See the SAN Administration Guide from the NetApp Support site for the version of Data ONTAP that you are using on the NetApp Private Storage system.

2. The successful outcome of the test is that an iSCSI session is be successfully established from the iSCSI software initiator on the AWS virtual machine instance to the iSCSI LIF on the NetApp Private Storage.

Verifying iSCSI LUN Access

1. From a local administration host, or from the AWS virtual machine instance, create an aggregate, flexible volume, LUN, and igroup using the Data ONTAP CLI or NetApp OnCommand® System Manager software.

   Note: The commands and/or workflows to create these storage primitives depends on the version of Data ONTAP used on the NetApp Private Storage system.

   Note: See the SAN Administration Guide from the NetApp Support site for the version of Data ONTAP that you are using on the NetApp Private Storage system.

2. After configuring the NetApp storage, use iSCSI tools on the AWS virtual machine instance to discover the iSCSI LUN (i.e., iscsiadm, or Windows ISCSI control panel application, and so on)

   Note: Refer to the documentation of the operating system of the AWS virtual machine instance on how to discover the iSCSI LUN.

3. After the iSCSI LUN has been discovered by the AWS virtual machine instance, create a file system on the LUN and mount the file system.

   Note: Refer to the documentation of the operating system of the AWS virtual machine instance on how to discover the iSCSI LUN.

4. Use the CD utility on your AWS virtual machine instance connected to the iSCSI LUN. Write a text file and save it to the iSCSI LUN.

   Note: Refer to the documentation of the operating system of the AWS virtual machine instance on how to write and save a file.

5. The successful outcome of this test is that you will be able to access the LUN file system and write a file to it.

Verifying SMB Protocol Connectivity

1. To perform this test, you need an AWS VM instance running the Windows operating system deployed to the VPC that is connected to the Direct Connect network. If you do not have a Windows VM instance deployed, deploy one before proceeding to step 2.
2. From a local administration host, or from the AWS VM instance, create a flexible volume, and junction point on the NetApp Private Storage system.

   Note: Refer to the File Access Management Guide for CIFS from the NetApp Support site for the version of Data ONTAP that you are using on the NetApp Private Storage system.

3. After creating the SMB share, use the AWS VM instance to access the share. Write a text file and save it to the SMB share.
4. The successful outcome of this test is that you will be able to access the SMB share and write a file to it.

Verifying NFS Protocol Connectivity

1. To perform this test, you need an AWS VM instance running the Linux operating system deployed to the VPC that is connected to the Direct Connect network. If you do not have a Linux VM instance deployed, deploy one before proceeding to step 2.
2. From a local administration host, or from the Linux VM instance, create a flexible volume, and junction point on the NetApp Private Storage system.

   Note: Refer to the File Access Management Guide for NFS on the NetApp Support site for the version of Data ONTAP that you are using on the NetApp Private Storage system.

3. After creating the NFS export, use the AWS VM instance to mount the export. Write a text file and save it to the NFS export.

4. The successful outcome for this test is that you will be able to access the NFS export and write a file to it.

Testing AutoSupport

For NetApp AutoSupport™ to work, the NetApp storage must have access to the Internet or to a mail host that has access to the Internet. You can accomplish this in one of the following ways:

- Set up a mail host in the VPC that is connected to the storage.
- Set up a network connection to the Internet in the colocation where the storage is located.
- Set up a network connection on premises over a VPN or MPLS connection.

Note: Refer to the System Administration Guide from the NetApp Support site for the version of Data ONTAP that you are using on the NetApp Private Storage system.

3.8 Performance Test Guidelines

The concepts underlying performance testing with NetApp Private Storage for AWS are similar to those for performance testing in other environments. The following sections describe considerations to take into account when conducting performance testing in the NetApp Private Storage for AWS solution environment.

Goals of Performance Testing

Performance tests are used to validate the performance of the storage, network, and computing resources, given a specific workload that is an estimate of a real-world workload.

All architectures have limits to their performance. The goal of performance testing is not to see how much load you can put in the environment before things break. Instead, the goal is to follow an iterative, deliberate process that results in data that can be plotted and analyzed so that architects can anticipate performance based on a given workload (that is, performance curves).

NetApp Storage Considerations for Performance Testing

The considerations for sizing NetApp storage are the same in the NetApp Private Storage for AWS solution architecture as in typical deployments of NetApp storage. NetApp storage requires the following considerations:

- **Number and type of NetApp controllers.** Are the number and type of controllers used in the testing appropriate for the performance testing?
- **Number and type of disks in the aggregates.** Do the number and type of disks in the aggregate used in the testing have enough IOPS and storage capacity for the testing?
- **NetApp Flash Cache® caching.** Are Flash Cache adapters installed in the storage controller nodes?
- **Cluster node network connectivity.** What is the bandwidth of network connections (1GbE or 10GbE), and how many connections are used to connect the storage to the network equipment in the Equinix colocation data center that is connected to the AWS cloud?

Network Equipment Considerations for Performance Testing

The considerations for the network equipment in the NetApp Private Storage for AWS solution architecture are the same as those in typical network environments. The network equipment requires the following considerations:

- **Available CPU and memory.** Does the switch that is being used have enough resources to support the performance testing? Adding more workload to an oversubscribed network switch might contribute to invalid performance testing results.

- **Network ports used.** What is the bandwidth of network connections (200Mbps, 500Mbps, 1Gbps, or 10Gbps), and what is the number of connections used to connect to the storage and to AWS? Is there enough bandwidth available to accommodate a performance test?

AWS Considerations for Performance Testing

It is very important to understand how the components of the AWS cloud can affect performance testing. The following considerations apply to the AWS cloud:

- **AWS Direct Connect network connection.** Is there enough bandwidth available to accommodate performance testing? Contention for network bandwidth can affect performance testing results. Be sure that there is enough network bandwidth to support the testing.
- **EC2 VM instance type.** Verify that you are using the proper instance type for performance testing. AWS throttles network throughput for smaller instance types and allocates more network bandwidth for larger instance types. Having the correct instance type is critical for a successful performance test.

Load-Generation and Monitoring Tools for Performance Testing

The load-generation and monitoring tools used for performance testing in the NetApp Private Storage for AWS solution architecture are the same as those used in typical NetApp storage environments. Consider the following guidelines:

- **Know which tool you want to use.** Each tool has advantages and disadvantages. Understanding the correct tool for your performance testing can provide more accurate test results.
- **Know your workload.** What kind of workload will you be testing? Understanding the I/O patterns of the workloads you are testing helps make it possible to configure the load generation tool correctly so that the testing can accurately model the performance.
- **Monitor the stack.** Implement monitoring for the computing, network, and storage resources so that bottlenecks can be identified. Collect performance data from each stack so that analysis can provide a more complete picture of how the NetApp Private Storage for AWS solution architecture is performing.

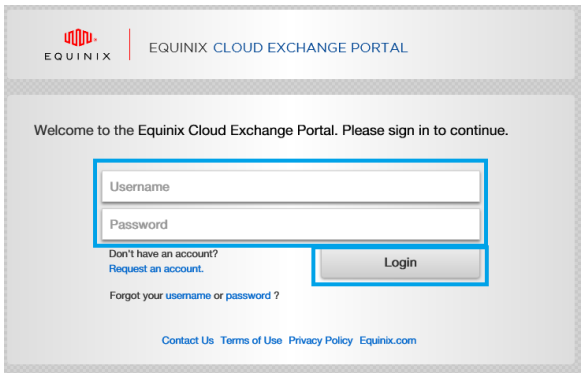
4 Using Equinix Cloud Exchange With Direct Connect

This procedure describes how to provision AWS Direct Connect through the Equinix Cloud Exchange. Only sub-1Gbps Direct Connect connections are available when using the Equinix Cloud Exchange to connect to AWS.

The Equinix Cloud Exchange is available in certain Equinix colocation data centers. For a current list of locations where the Equinix Cloud Exchange is available, refer to the Equinix Cloud Exchange datasheet.

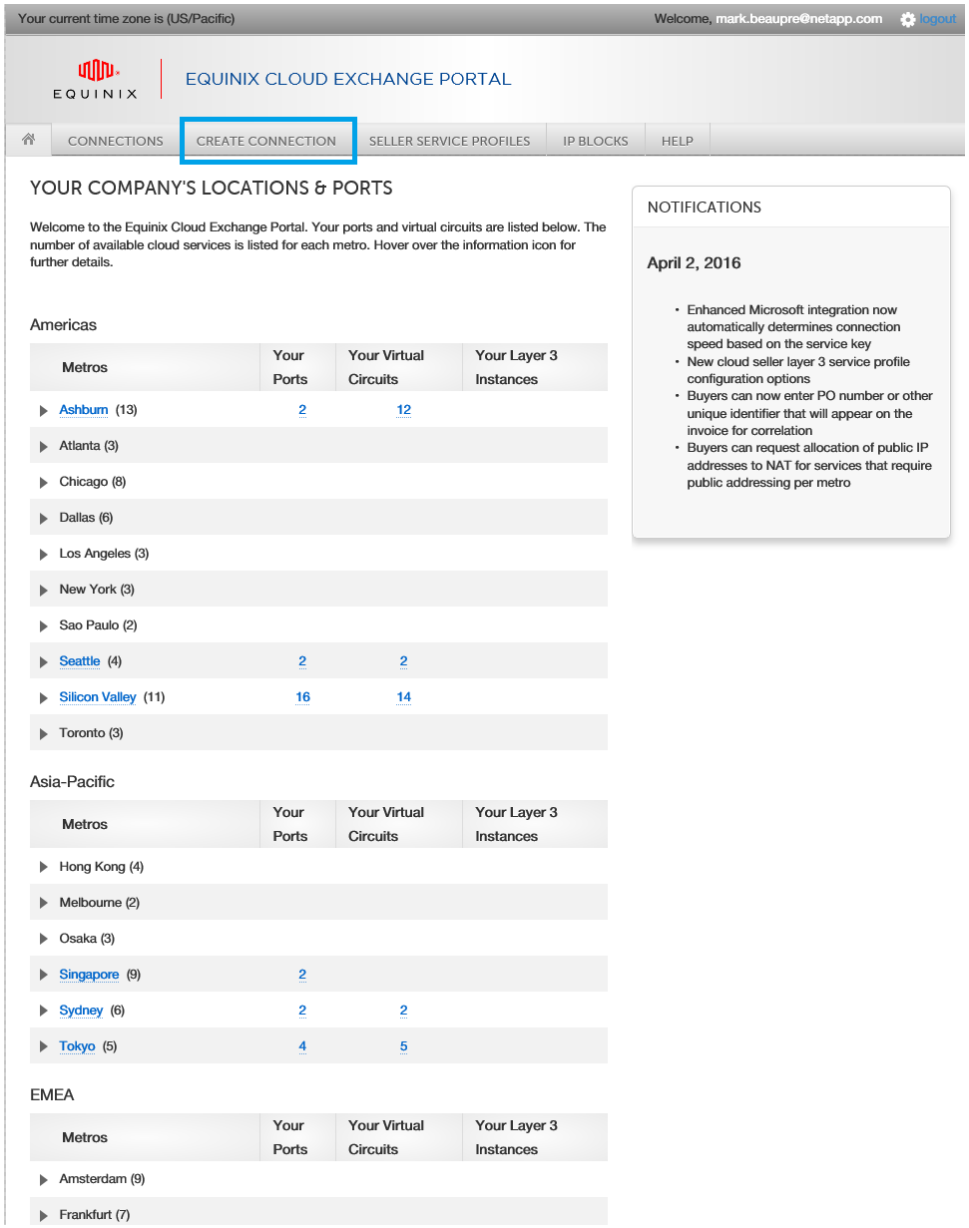
To use the Equinix Cloud Exchange to provision an AWS Direct Connect connection, complete the following steps:

1. From an Internet-connected computer, open a web browser and go to the Equinix Cloud Exchange Portal.
2. Sign in to the Equinix Cloud Exchange portal with the Cloud Exchange Portal credentials that Equinix assigned to you and click Login.

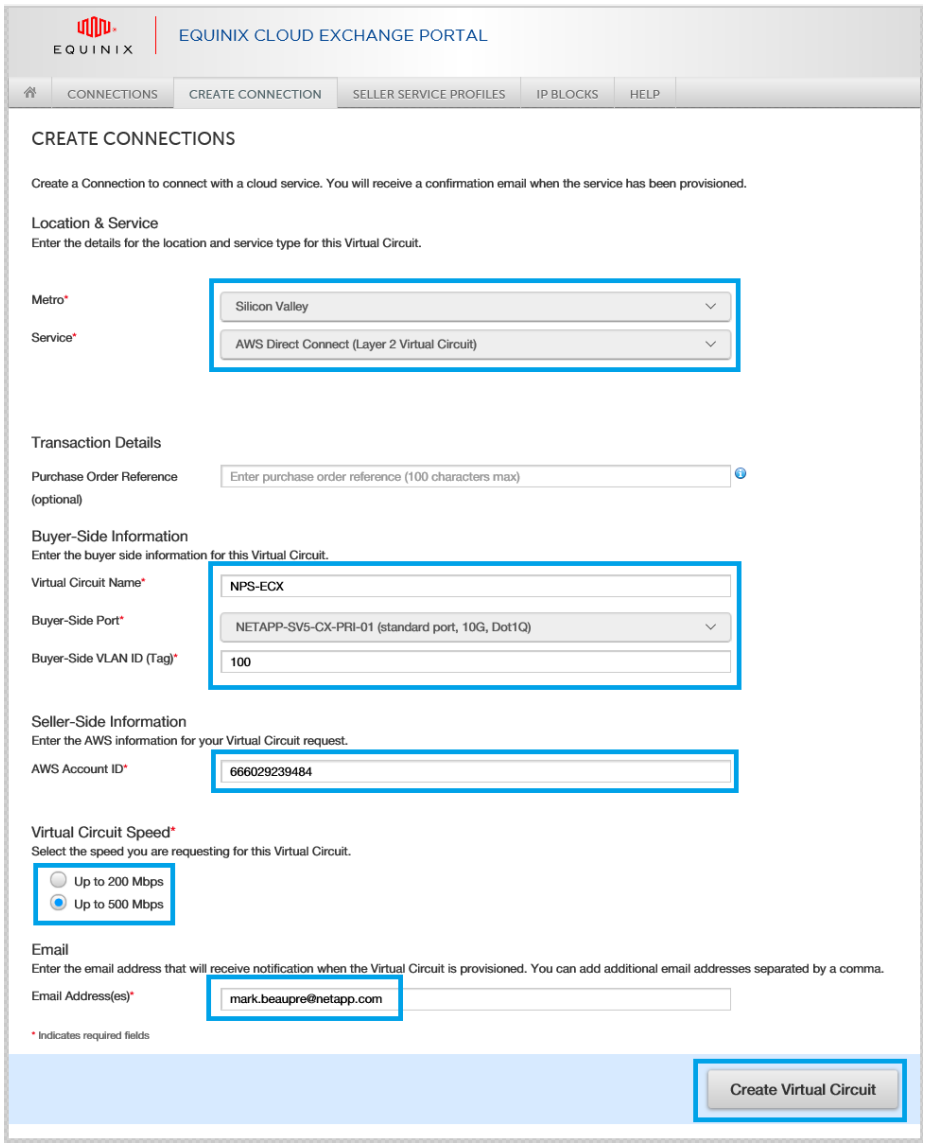


Note: If you do not have Cloud Exchange portal credentials, contact your Equinix account team.

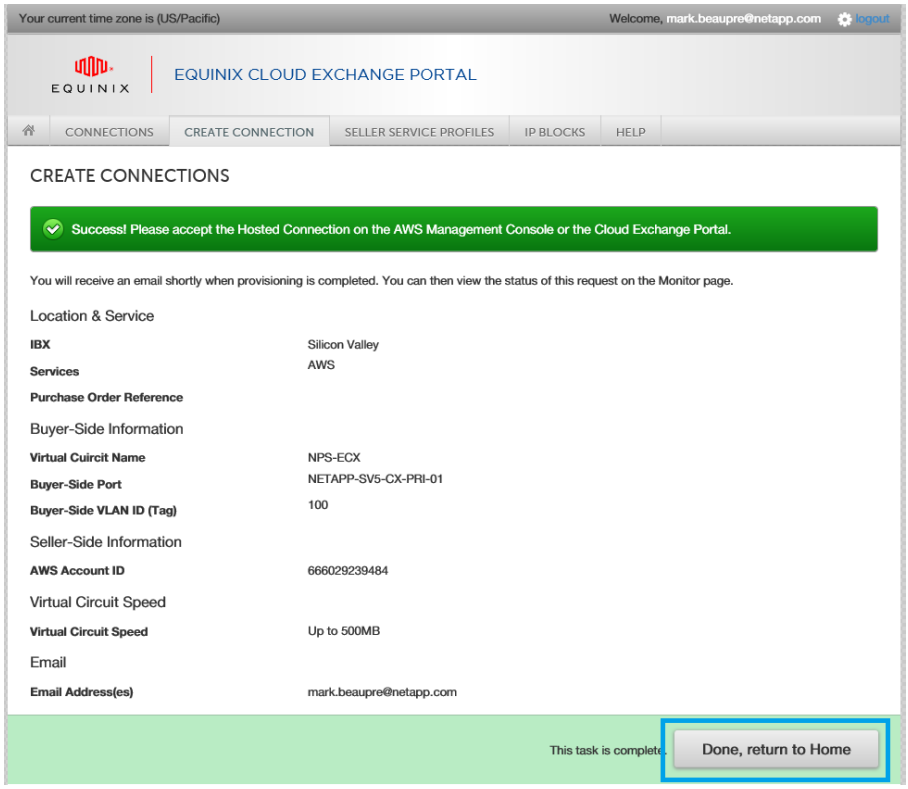
3. In the Cloud Exchange Portal, click the Create Connection tab.



4. In the Create Virtual Circuit dialog box, provide the following parameters and then click Create Virtual Circuit:
   - Metro: From the drop-down list, select the location of the Equinix colocation data center (in this example, `Silicon Valley`).
   - Service: From the drop-down list, select AWS Direct Connect.
   - Virtual Circuit Name: Provide a text string that identifies the Direct Connect virtual interface (in this example, `NPS-ECX`).
   - AWS Account ID: Provide your AWS account number.
   - Virtual Circuit Speed: Select 200Mbps or 500Mbps.
   - Email: Provide a valid e-mail address. This e-mail address is used for communication about the status of your virtual circuit request.



5. When the status of the virtual circuit request is displayed, click Done, Return to Home.



Note: Equinix notifies you when the virtual circuit has been created.

6. Using the AWS Command Line interface, run the following command to displays the connection created by the Equinix Cloud Exchange:

```
aws directconnect describe-connections --query 'connections[?connectionState==`ordering`]'
```

The output appears as follows:

```
[
    {
        "partnerName": "EQUINIX NNI",
        "vlan": 107,
        "ownerAccount": "666029239484",
        "connectionId": "dxcon-ffnod55u",
        "connectionState": "ordering",
        "bandwidth": "500Mbps",
        "location": "EqSV5",
        "connectionName": "NPS-ECX",
        "region": "us-west-1"
    }
]
```

Note: It may take a few seconds after receipt of the e-mail notification that the ECX virtual circuit has been created and when the Direct Connect connection is provisioned in AWS.

7. Run the following command to accept the AWS Direct Connect connection request:

```
aws directconnect confirm-connection --connection-id <<conn-id>>
```

Where:

– `<<conn-id>>` is the `connectionId` parameter from step 6.

The output appears as follows:

```
        {
    "connectionState": "pending"
}
```

8. After a few seconds, the connection state changes from a `pending` to `available`. Run the following command to verify that the connection is in the `available` state:

```
aws directconnect describe-connections --connection-id <<conn-id>>
```

Where:

- `<<conn-id>>` is the `connectionId` parameter from step 6.

The output appears as follows:

```
{
    "connections": [
        {
            "partnerName": "EQUINIX NNI",
            "vlan": 107,
            "ownerAccount": "666029239484",
            "connectionId": "dxcon-ffnod55u",
            "connectionState": "available",
            "bandwidth": "500Mbps",
            "location": "EqSV5",
            "connectionName": "NPS-ECX",
            "region": "us-west-1"
        }
    ]
}
```

5 AWS GovCloud

AWS GovCloud is a region completely separated from all other AWS regions. AWS GovCloud is used only by the United States government to run workloads and services in the AWS cloud subject to the strict compliance requirements of the US government.

See the AWS GovCloud documentation for more information about the GovCloud region.

Currently, there is only one AWS Direct Connect PoP for the AWS GovCloud region through cross connect. The location of this PoP is Equinix SV1/SV5 in San Jose, CA.

The functionality of the Direct Connect service in the GovCloud region is identical to the functionality of Direct Connect in the other commercial AWS regions. There are public and private virtual interfaces and the workflows to create and configure them are the same as the Direct Connect workflows in the other AWS commercial regions.

The compliance program that has the most effect on the NetApp Private Storage for AWS solution architecture is the US International Traffic in Arms Regulations (ITAR) program.

ITAR-regulated data are defense-related articles and services on the United States Munitions List (USML) and related technical data. Due to the nature of the data, only U.S. citizens are authorized to have access.

Table 2 defines the ITAR boundary as described in the AWS GovCloud Direct Connect User Guide.

Table 2) GovCloud ITAR boundary.

ITAR-Regulated Data Permitted	ITAR-Regulated Data Not Permitted
• If you are transferring any type of ITAR-regulated data through the AWS Direct Connect connection, you must encrypt the data that is being transferred by using a VPN tunnel.	• AWS Direct Connect metadata is not permitted to contain ITAR-regulated data. This metadata includes all of the configuration data that you enter when creating and maintaining AWS Direct Connect, such as connection names. • Do not enter ITAR-regulated data in the following console fields: – Connection Name – VIF Name

If you are managing ITAR-regulated data, a hardware VPN appliance is required in the cabinet/shared cage at Equinix to encrypt the network traffic between the NetApp storage and the AWS cloud compute. The VPN tunnel connects to the AWS VPC over a Direct Connect public virtual interface.

If you are managing non-ITAR-regulated data, with Direct Connect and GovCloud, use a private virtual interface with no hardware VPN appliance.

The Federal Risk and Authorization Management Program (FedRAMP) does not directly affect the technical aspects of the solution, but it does affect the ability of the solution to be deployed and managed.

Note: Currently, NetApp is working to secure FedRAMP certification for the NetApp Private Storage for AWS solution. Contact your NetApp account team for more information about the current status of FedRAMP certification and the availability of partners who have received the Agency Authorization to Operate (ATO).

The use cases for NetApp Private Storage for AWS are also valid for NetApp Private Storage in the AWS GovCloud region.

Deployment Considerations for NetApp Private Storage for AWS GovCloud

Although, AWS GovCloud is very similar in functionality to the AWS commercial regions, there are differences that must be taken into account when undertaking the NetApp Private Storage for AWS GovCloud deployment.

The high-level deployment workflow for NetApp Private Storage for AWS GovCloud consists of the following phases and tasks:

1. Planning:
   a. Preinstallation and Site Preparation
2. Deployment:
   a. Installing the Equipment in the Equinix Data Center
   b. Setting Up AWS Virtual Private Cloud Network
   c. Setting Up the AWS Direct Connect
   d. Setting Up the Customer Network Switch
   e. Setting Up the VPN Appliance
   f. Configuring NetApp Storage
3. Validation:
   a. Testing Connections and Protocols

The steps are similar to a NetApp Private Storage deployment using an AWS commercial region. Instead of repeating the deployment steps documented previously, the following sections provide the additional steps to use the AWS GovCloud region.

5.1 Planning

Preinstallation and Site Preparation

GovCloud AWS Account Setup

Only citizens of United States are authorized to use the AWS GovCloud resources. You must have a standard AWS account set up before you request access to the AWS GovCloud region. It is a best practice to set up a separate account linked to your standard account using consolidated billing.

See the AWS GovCloud User Guide for information about signing up for an AWS GovCloud account.

IP Address Plan for ITAR Data

The IP address plan is similar, except that there is an additional CIDR for the VPN tunnel endpoint and that the public IP address network CIDR is used.

Table 3) IP address plan for NetApp Private Storage for AWS GovCloud.

Tenant	Tenant VLAN	NetApp Private Storage SVM Network	VPN "Outside" Network	BGP Peering Network	BGP Authentication Key	BGP ASN	AWS Network	AWS Region

Note: For non ITAR data, the standard IP Address Plan template is used.

Site Preparation for ITAR Data

The power and space requirements of the VPN appliance(s) used for the ITAR data should be considered during the site preparation planning process.

In addition, the VPN appliance(s) need to be included in Equinix inbound shipment requests.

NPS for AWS GovCloud (non ITAR Data) Solution Architecture

The network architecture of Direct Connect with AWS GovCloud is the same as Direct Connect with commercial AWS regions. The main difference is the connection of the Direct Connect private virtual interface to a VPC in the AWS GovCloud Region.

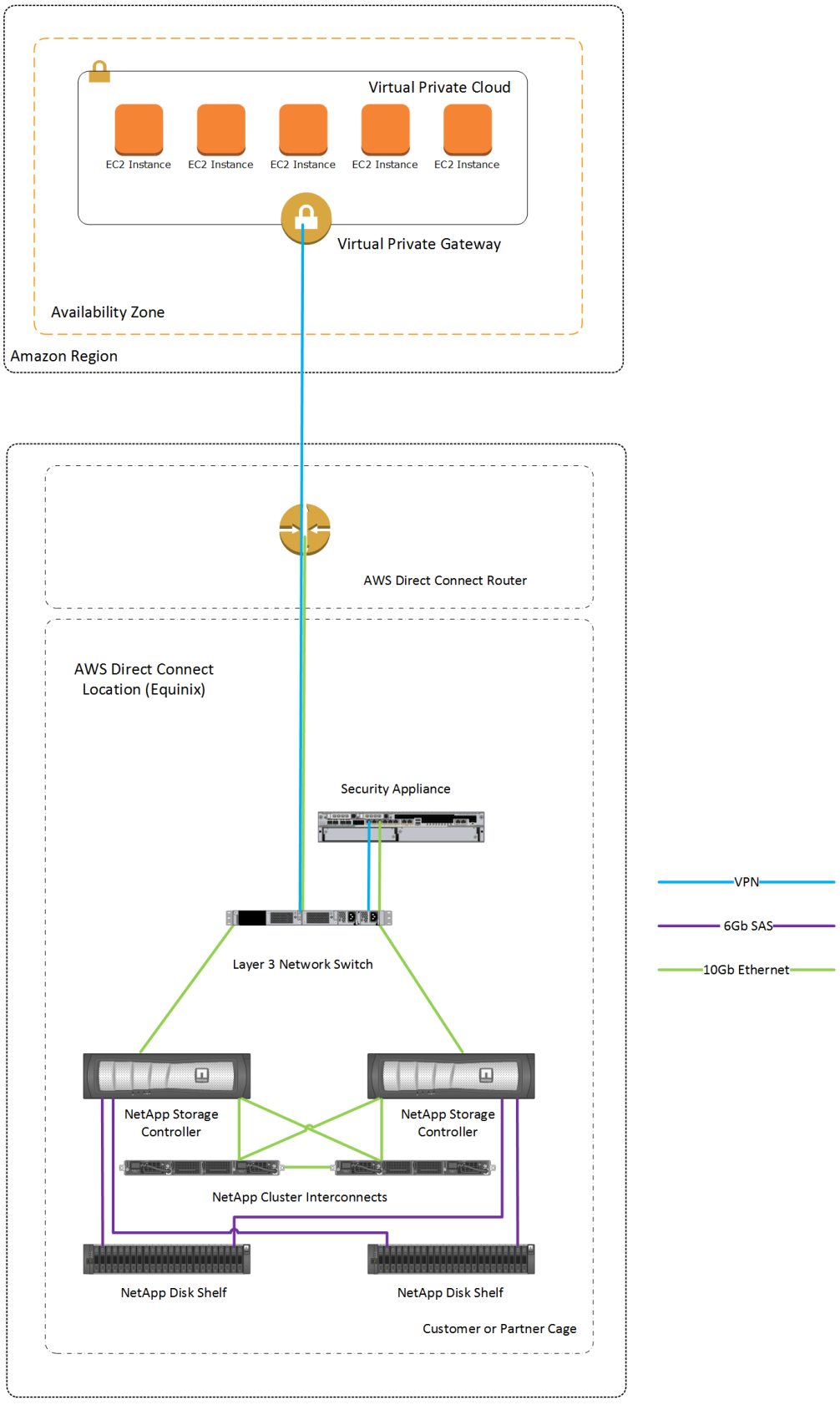
See Figure 4 for an illustration of the NetApp Private Storage solution.

NPS for AWS GovCloud (ITAR Data) Solution Architecture

In addition to the standard solution components for NetApp Private Storage, a security appliance is required for encrypting the network traffic between the customer network in Equinix and the AWS VPC network.

Figure 6 depicts a single VPC with a single hardware VPN appliance.

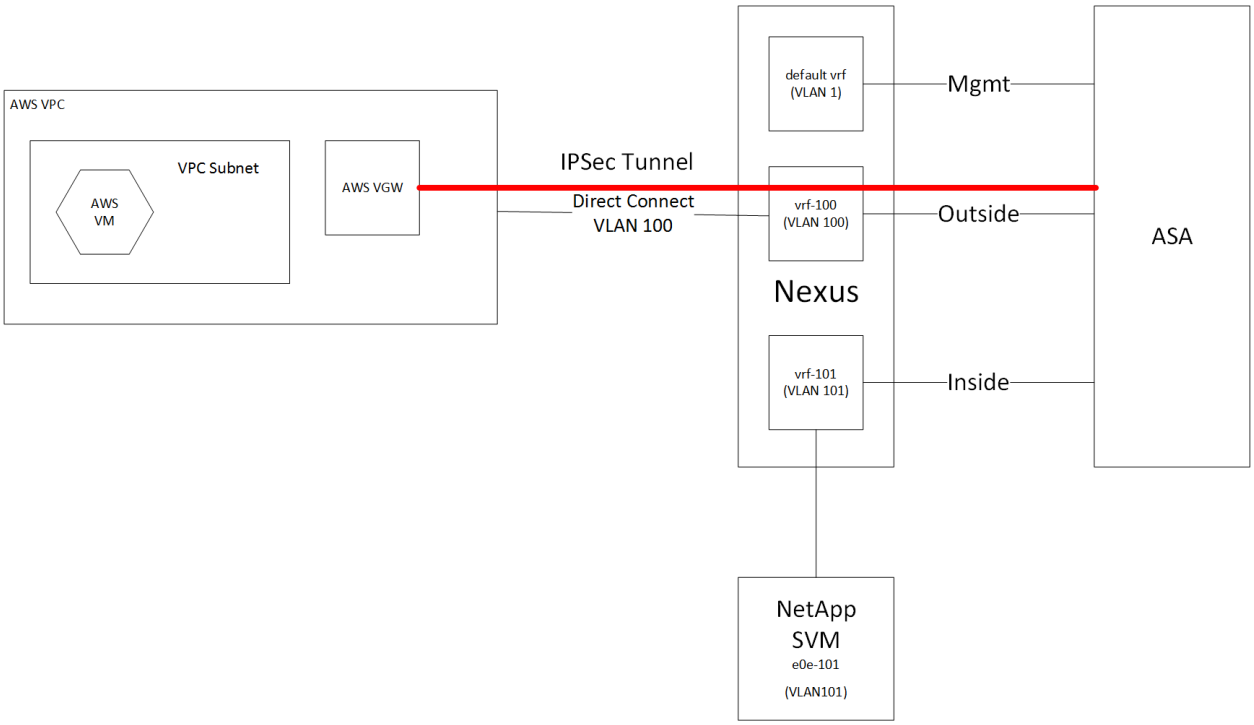
Figure 6) NPS for AWS GovCloud (ITAR) data network architecture.



NPS for AWS GovCloud (ITAR Data) Network Architecture

Figure 7 depicts the network architecture for a single VPC with single hardware VPN appliance.

Figure 7) NPS for AWS GovCloud (ITAR) data network architecture.



The Direct Connect public interface uses VLAN 100 and the “inside” network uses VLAN 101. Each tenancy consists of an “inner” and “outer” network.

Note: A Direct Connect private virtual interface requires you to use public IP addresses.

5.2 Deployment

Installing the Equipment in the Equinix Data Center

In addition to the standard equipment that comprises the NPS for AWS solution, a network security appliance is required in the cage at Equinix. See the security appliance vendor technical documentation for the power, cooling, and space (in rack units) required to operate the security appliance.

Make sure to include this VPN appliance in all Equinix inbound shipping requests.

Setting Up AWS Virtual Private Cloud Network

The AWS CLI commands to set up the VPC are the same as for non ITAR data.

However, a customer gateway and a VPN connection need to be created for the VPC.

Note: If you do not want to use the Amazon CLI to create the VPC, customer gateway, and VPN connection, you can use the Start VPC wizard from the VPC dashboard.

1. Run the following command to create a customer gateway:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip <<outside-ip>>
```

Where:

- <<outside-ip>> is the IP address of the outside interface on the VPN appliance.

2. Run the following command to create a VPN connection:

```
aws ec2 create-vpn-connection --type ipsec.1 --customer-gateway-id <<cgw-id>> --vpn-gateway-id
<<vgw-id>> --options "{\"StaticRoutesOnly\":<<boolean>>}"
```

Where:

- <<cgw-id>> is the customer gateway created in step 1.
- <<vgw-id>> is the AWS VGW that was created and attached to the VPC.
- <<boolean>> is either true or false. Set this value to false if you are using static routes instead of BGP over the IPsec connection.

3. If you've configure your VPN network connection to use static routes, run the following command to create a static route in the VPN network connections:

```
aws ec2 create-vpn-connection-route --vpn-connection-id <<vpn-id>> --destination-cidr-block
<<svm-cidr>>
```

Where:

- <<vpn-id>> is the VPN connection that was created in step 2.
- <<svm-cidr>> is the CIDR of the local network where the SVM is connected (i.e., 192.168.23.160/28)

Setting Up the AWS Direct Connect

The AWS CLI commands to set up the Direct Connect network connection is the same. However, the command to set up the Direct Connect public virtual interface to support ITAR data differs from creating the private virtual interface for non ITAR data.

1. Run the following command to create the Direct Connect public virtual interface:

```
virtualInterfaceName=<<pub-vi-name..,vlan=<<vlan>>,asn=<<asn>>,authKey=<<bgp-
key>>,amazonAddress=<<amzn-peer-ip>>,customerAddress=<<cust-peer-
ip>>,routeFilterPrefixes=[{<<peer-cidr>>},{<<outside-cidr>>}]
```

Where:

- <<conn-id>> is the value of the connectionId parameter from step 4 in section 3.4.
- <<pvi-name>> is the name of the private virtual interface (i.e., NPS-PVI).
- <<vlan>> is the VLAN number of the public virtual interface (i.e., 101).
- <<asn>> is the autonomous system number of your network equipment in Equinix (i.e., 64514).
- <<bgp-key>> is the BGP authentication key (i.e., eea0a828f3e5fe02687cce9c).
- <<aws-peer-ip>> is the AWS BGP peer IP address (i.e., 217.70.223.209/31).
- <<cust-peer-ip>> is the BGP peer IP address (i.e., 217.70.223.208/31).
- <<peer-cidr>> is the BGP peer CIDR network (i.e., 217.70.223.208/31).
- <<outside-cidr>> is the CIDR network of the outside network used by the VPN appliance. (i.e., 217.70.223.212/30). This CIDR can be bigger if you want to have more than one outside interface on the VPN security.

Note: The value of the virtualInterfaceState parameter will initially show as verifying until AWS has verified that the public CIDR networks can be used. After the public virtual interface is verified by AWS, it appears as down until your network equipment in Equinix is configured and the BGP session is established.

The turnaround on verifying public virtual interfaces is three business days. If after three days, the status hasn't changed, contact AWS support.

Setting Up the Customer Network Switch

The switch configuration is very similar except that you will be creating a VLAN, VLAN interface, and VRF for the outside interface for the VPN appliance.

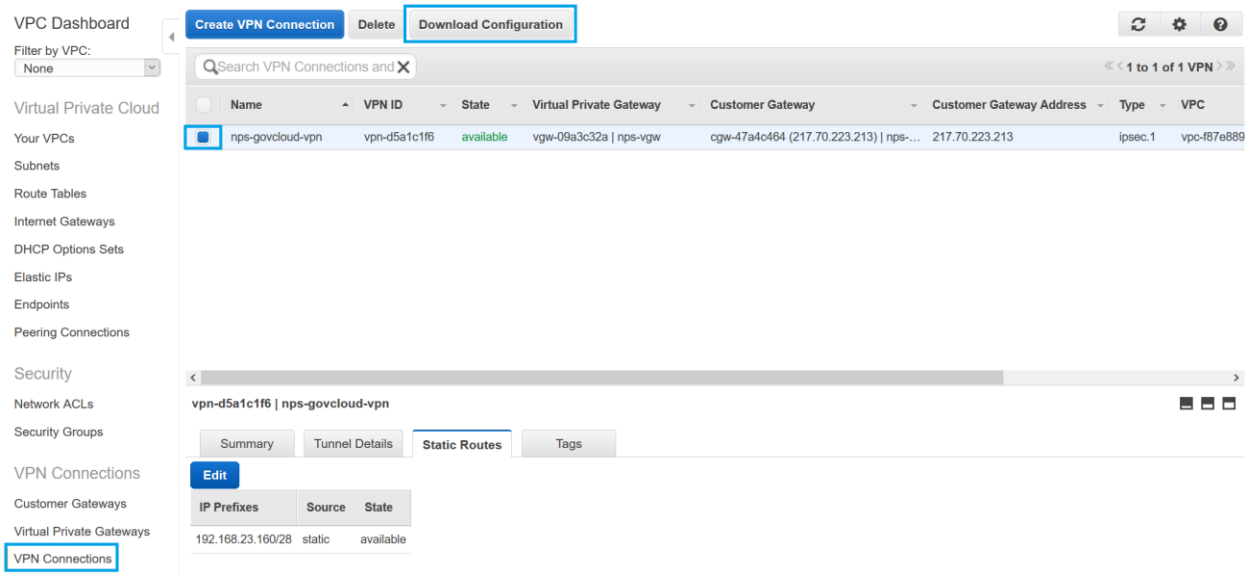
The SVM VLAN, VLAN interface, and VRF are configured the same. The inside interface of the VPN appliance will have an IP address from the SVM CIDR.

Setting Up VPN Appliance

The configuration details of the VPN appliance depend on the manufacturer and model of the appliance that you are using.

AWS offers templates to help set up the IPsec configuration options. These templates can be downloaded by clicking the VPN Connections within the VPC dashboard.

1. In the VPC Dashboard, click VPN Connections > your VPN connection > Download Configuration.



The configuration template is similar to the following. This template is based on ASA 8.2 or later.

```
! Amazon Web Services
! Virtual Private Cloud
!
! AWS utilizes unique identifiers to manipulate the configuration of
! a VPN Connection. Each VPN Connection is assigned an identifier and is
! associated with two other identifiers, namely the
! Customer Gateway Identifier and Virtual Private Gateway Identifier.
!
! Your VPN Connection ID                 : vpn-d5a1c1f6
! Your Virtual Private Gateway ID        : vgw-09a3c32a
! Your Customer Gateway ID               : cgw-47a4c464
!
!
! This configuration consists of two tunnels. Both tunnels must be
! configured on your Customer Gateway. Only a single tunnel will be up at a
! time to the VGW.
!
! You may need to populate these values throughout the config based on your setup:
! <outside_interface> - External interface of the ASA
! <outside_access_in> - Inbound ACL on the external interface
! <amzn_vpn_map> - Outside crypto map
! <vpc_subnet> and <vpc_subnet_mask> - VPC address range
! <local_subnet> and <local_subnet_mask> - Local subnet address range
```

```
! <sla_monitor_address> - Target address that is part of acl-amzn to run SLA monitoring

! ------------------------------------------------------------------------------
! IPSec Tunnels
! ------------------------------------------------------------------------------
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
!
! Note that there are a global list of ISAKMP policies, each identified by
! sequence number. This policy is defined as #201, which may conflict with
! an existing policy using the same number. If so, we recommend changing
! the sequence number to avoid conflicts.
!
crypto isakmp identity address
crypto isakmp enable <outside_interface>
crypto isakmp policy 201
  encryption aes
  authentication pre-share
  group 2
  lifetime 28800
  hash sha
exit
!
! The tunnel group sets the Pre Shared Key used to authenticate the
! tunnel endpoints.
!
tunnel-group 205.251.237.237 type ipsec-l2l
tunnel-group 205.251.237.237 ipsec-attributes
   pre-shared-key 0NWqoXiZpSPwOWydWC1gUQjFan2ed9NW
!
! This option enables IPSec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
   isakmp keepalive threshold 10 retry 3
exit
!
tunnel-group 205.251.237.206 type ipsec-l2l
tunnel-group 205.251.237.206 ipsec-attributes
   pre-shared-key 7QzoZu8f2ZZacRsKnyv5r61pts0tpQ99
!
! This option enables IPSec Dead Peer Detection, which causes periodic
! messages to be sent to ensure a Security Association remains operational.
!
   isakmp keepalive threshold 10 retry 3
exit

! ------------------------------------------------------------------------------
! #2: Access List Configuration
!
! Access lists are configured to permit creation of tunnels and to send applicable traffic over
them.
! This policy may need to be applied to an inbound ACL on the outside interface that is used to
manage control-plane traffic.
! This is to allow VPN traffic into the device from the Amazon endpoints.
!
access-list <outside_access_in> extended permit ip host 205.251.237.237 host 217.70.223.213
access-list <outside_access_in> extended permit ip host 205.251.237.206 host 217.70.223.213
! The following access list named acl-amzn specifies all traffic that needs to be routed to the
VPC. Traffic will
! be encrypted and transmitted through the tunnel to the VPC. Association with the IPSec security
association
! is done through the "crypto map" command.
!
! This access list should contain a static route corresponding to your VPC CIDR and allow traffic
from any subnet.
! If you do not wish to use the "any" source, you must use a single access-list entry for
accessing the VPC range.
! If you specify more than one entry for this ACL without using "any" as the source, the VPN will
function erratically.
```

```
! See section #4 regarding how to restrict the traffic going over the tunnel
!
!
access-list acl-amzn extended permit ip any <vpc_subnet> <vpc_subnet_mask>

!--------------------------------------------------------------------------------
! #3: IPSec Configuration
!
! The IPSec transform set defines the encryption, authentication, and IPSec
! mode parameters.
!
crypto ipsec transform-set transform-amzn esp-aes esp-sha-hmac
! The crypto map references the IPSec transform set and further defines
! the Diffie-Hellman group and security association lifetime. The mapping is created
! as #1, which may conflict with an existing crypto map using the same
! number. If so, we recommend changing the mapping number to avoid conflicts.
!
crypto map <amzn_vpn_map> 1 match address acl-amzn
crypto map <amzn_vpn_map> 1 set pfs group2
crypto map <amzn_vpn_map> 1 set peer  205.251.237.237 205.251.237.206
crypto map <amzn_vpn_map> 1 set transform-set transform-amzn
!
! Only set this if you do not already have an outside crypto map, and it is not applied:
!
crypto map <amzn_vpn_map> interface <outside_interface>
!
! Additional parameters of the IPSec configuration are set here. Note that
! these parameters are global and therefore impact other IPSec
! associations.
! Set security association lifetime until it is renegotiated.
crypto ipsec security-association lifetime seconds 3600
!
! This option instructs the firewall to clear the "Don't Fragment"
! bit from packets that carry this bit and yet must be fragmented, enabling
! them to be fragmented.
!
crypto ipsec df-bit clear-df <outside_interface>
!
! This configures the gateway's window for accepting out of order
! IPSec packets. A larger window can be helpful if too many packets
! are dropped due to reordering while in transit between gateways.
!
crypto ipsec security-association replay window-size 128
!
! This option instructs the firewall to fragment the unencrypted packets
! (prior to encryption).
!
crypto ipsec fragmentation before-encryption <outside_interface>
!
! This option causes the firewall to reduce the Maximum Segment Size of
! TCP packets to prevent packet fragmentation.
sysopt connection tcpmss 1387
!
! In order to keep the tunnel in an active state, the ASA needs to send traffic to the subnet
! defined in acl-amzn. SLA monitoring can be configured to send pings to a destination in the
subnet and
! keep the tunnel active. A possible destination for the ping is the VPC Gateway IP, which is the
! first IP address in one of your subnets.
! For example: a VPC with a CIDR range of 192.168.50.0/24 will have a gateway: 192.168.50.1.
!
! The monitor is created as #1, which may conflict with an existing monitor using the same
! number. If so, we recommend changing the sequence number to avoid conflicts.
!
sla monitor 1
   type echo protocol ipIcmpEcho <sla_monitor_address> interface <outside_interface>
   frequency 5
exit
sla monitor schedule 1 life forever start-time now
!
! The firewall must allow icmp packets to use "sla monitor"
icmp permit any <outside_interface>
```

```
!-----------------------------------------------------------------------------------
! #4: VPN Filter
! The VPN Filter will restrict traffic that is permitted through the tunnels. By default all
traffic is denied.
! The first entry provides an example to include traffic between your VPC Address space and your
office.
! You may need to run 'clear crypto isakmp sa', in order for the filter to take effect.
!
! access-list amzn-filter extended permit ip <vpc_subnet> <vpc_subnet_mask> <local_subnet>
<local_subnet_mask>
access-list amzn-filter extended deny ip any any
group-policy filter internal
group-policy filter attributes
vpn-filter value amzn-filter
tunnel-group 205.251.237.237 general-attributes
default-group-policy filter
exit
tunnel-group 205.251.237.206 general-attributes
default-group-policy filter
exit

!-----------------------------------------------------------------------------------
! #5: NAT Exemption
! If you are performing NAT on the ASA you will have to add a nat exemption rule.
! This varies depending on how NAT is set up.  It should be configured along the lines of:
! object network obj-SrcNet
!   subnet 0.0.0.0 0.0.0.0
! object network obj-amzn
!   subnet <vpc_subnet> <vpc_subnet_mask>
! nat (inside,outside) 1 source static obj-SrcNet obj-SrcNet destination static obj-amzn obj-amzn
! If using version 8.2 or older, the entry would need to look something like this:
! nat (inside) 0 access-list acl-amzn
! Or, the same rule in acl-amzn should be included in an existing no nat ACL.
!
!-----------------------------------------------------------------------------------
!  Additional Notes and Questions
!  - Amazon Virtual Private Cloud Getting Started Guide:
!       http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide
!  - Amazon Virtual Private Cloud Network Administrator Guide:
!       http://docs.amazonwebservices.com/AmazonVPC/latest/NetworkAdminGuide
!  - Troubleshooting Cisco ASA Customer Gateway Connectivity:
!
http://docs.amazonwebservices.com/AmazonVPC/latest/NetworkAdminGuide/Cisco_ASA_Troubleshooting.ht
ml
!  - XSL Version: 2009-07-15-1119716
```

Note: Edit the text to account for your environment.

Configuring NetApp Storage

Note: Obtain the configuration parameters of the NetApp storage from the address plan of NetApp Private Storage for AWS in Table 3.

The steps to configure the NetApp storage are as follows:

1. Create VLAN interface ports on cluster nodes.

   Note: This is a critical step. VLAN ports on the storage are used to logically tie the network interfaces to the Direct Connect network.

2. Create an SVM on the cluster.
3. Create the following LIFs on the SVM that uses the VLAN interface ports:
   a. Management LIF
   b. CIFS/NFS LIF
   c. iSCSI LIFs

Refer the Data ONTAP documentation from the NetApp Support site for the version of Data ONTAP that you are using on the NetApp Private Storage system.

5.3 Validation

Testing Connections and Protocols

The procedures and tools used to test the network and protocol connectivity are the same as section 3.7.

Note: The performance characteristics between the non ITAR and ITAR data is not the same. The encryption of the data over the Direct Connect network connection adds a performance penalty. The performance penalty depends on the type of the VPN appliance used.

References

The following references were used in this report:

- Amazon Web Services Direct Connect Getting Started Guide
  http://docs.amazonwebservices.com/DirectConnect/latest/GettingStartedGuide/Welcome.html
- Amazon Web Services Direct Connect User Guide
  http://docs.amazonwebservices.com/directconnect/latest/UserGuide/Colocation.html
- Amazon Web Services GovCloud User Guide
  http://docs.aws.amazon.com/govcloud-us/latest/UserGuide/welcome.html

Version History

Version	Date	Document Version History
Version 3.0	April 2016	Added deployment steps for AWS GovCloud; Changed deployment steps to CLI commands from screenshots
Version 2.0	October 2014	Updated layout and screenshots and clarified deployment steps
Version 1.0	February 2013	Initial release

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Trademark Information