Overview and preparation

This section includes the following topics:

- Overview of InfoScale solutions in cloud environments
- InfoScale agents for monitoring resources in cloud environments
- Preparing for InfoScale installations in cloud environments
- Installing the AWS CLI package
- VPC security groups example

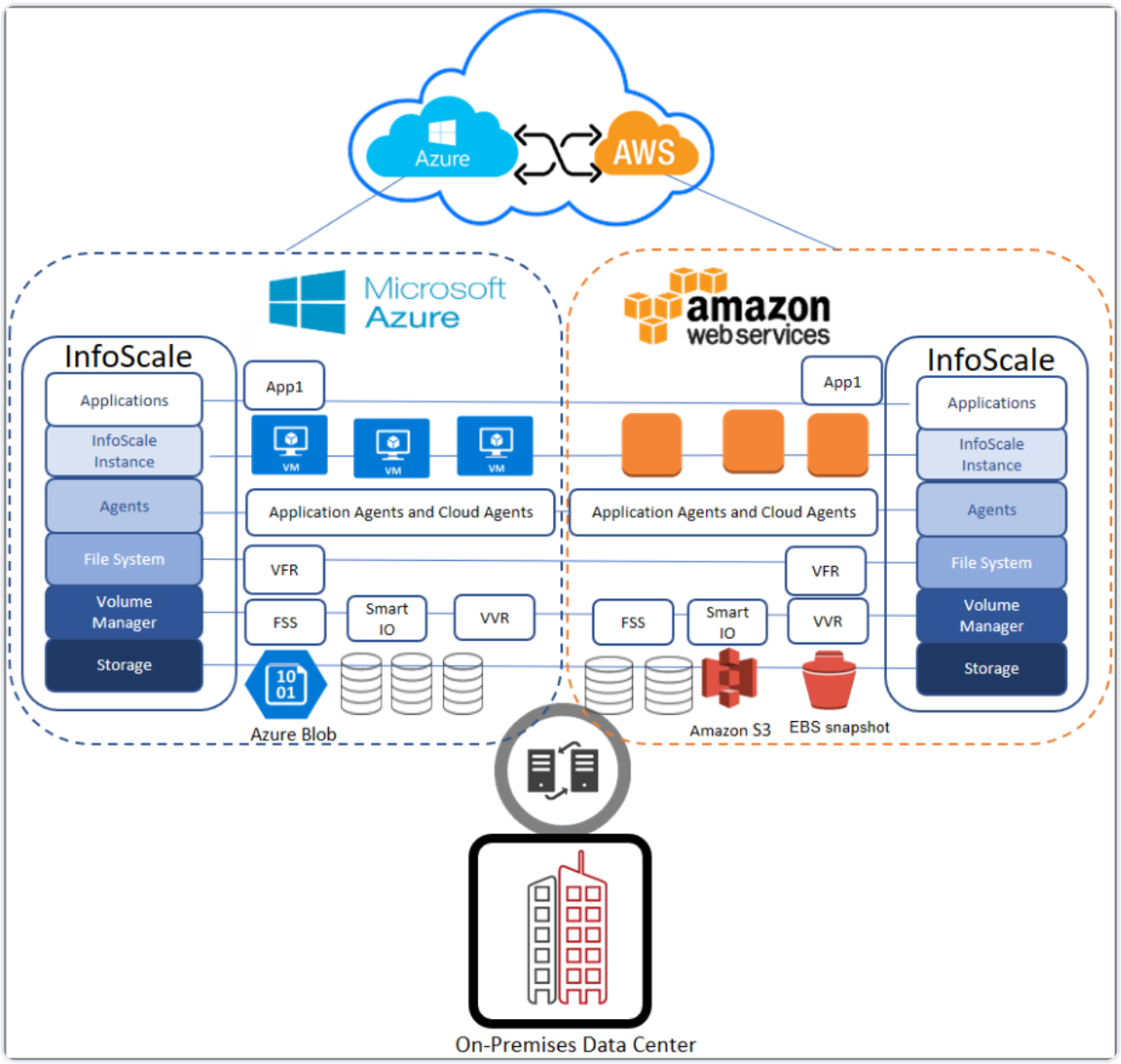
Overview of InfoScale solutions in cloud environments

The Veritas InfoScale® product suite helps you make your applications highly available and resilient to disruptions. The applications may reside in your on-premises data centers or in public, private, or hybrid cloud environments. With InfoScale, you can combine cost-effective platform independence with consistent high application performance and availability. Veritas supports InfoScale configurations in the Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) cloud environments.

Veritas InfoScale Storage components let you manage various kinds of storage, which includes SAN, local flash, SSD, DAS, and cloud S3 targets. Veritas InfoScale Availability agents provide monitoring and failover capabilities for the networking, storage, and replication resources that are associated with an application in the cloud.

Using these components you can configure an application for:

- Migration from on-premises to cloud or from cloud to cloud
- High availability or disaster recovery within a cloud
- Disaster recovery from on-premises to cloud or across clouds



Additionally, the InfoScale Operations Manager application migration wizard lets you automate the migration of your workload across OS platforms into the public cloud.

InfoScale agents for monitoring resources in cloud environments

InfoScale provides agents to identify the various resources that are used by applications in cloud environments. The agents monitor and manage the resource states to provide HA and DR for the configured applications.

Table: InfoScale agents for AWS

NAME	PURPOSE
AWSIP	Manages the private IP, elastic IP, and overlay IP resources in AWS.
AWSRoute53	Updates and monitors the host name
	to IP address mapping with the Amazon Route 53 DNS web service.
EBSVol	Attaches and detaches EBS volumes to and from Amazon EC2 instances, and monitors their state when attached to an instance.

Table: InfoScale agents for Azure

NAME	PURPOSE
AzureIP	Manages the private IP, public IP, and overlay IP resources in Azure.
AzureDNSZone	Provides DNS-based traffic routing and failover\:
	- Monitors and updates the host name to resource record mapping.
	- Performs the mapping for the Azure DNS domain during failover of nodes across subnets or regions.
AzureDisk	Attaches managed and unmanaged data disks
	to an Azure VM of the same or of a different resource group, monitors their state, and detaches them.

Table: InfoScale agents for GCP

NAME	PURPOSE
GoogleIP	Manages the private IP and the overlay IP resources in GCP.
GoogleDisk	Attaches persistent disks
	to an GCP VM instance of the same or of a different resource group, monitors their state, and detaches them.

For details on each of these agents, refer to the following documents:

- *Cluster Server Bundled Agents Reference Guide - Linux*
- *Cluster Server Bundled Agents Reference Guide - Windows*

Preparing for InfoScale installations in cloud environments

> **Note:** InfoScale installations in cloud environments are supported on the Microsoft Windows, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS, and Oracle Enterprise Linux platforms.

> Preparing for InfoScale product installations in cloud involve the following tasks:

1. Launch the instances on which you plan to install the InfoScale products.
2. Create the required network interfaces on all the instances and restart the network services.
3. Where required, add swap files on the instances.
4. Make sure that instances have public connectivity through public IP, elastic IP, the NAT gateway, or over VPN.

In case of Linux systems, this connectivity is also required so that any missing platform RPMs can be installed.

1. Copy the appropriate InfoScale installable files on one or all of the instances.
2. To allow remote installation of an InfoScale product:

- On Linux systems, enable root login over SSH on all the instances and set a password for the root user.
- On Windows systems, set the necessary administrative user and group permissions.

3. (AWS only) Ensure that the AWS CLI is installed on each InfoScale node.

See Installing the AWS CLI package.

1. Ensure that the security groups in your VPC are properly configured and all the UDP ports for the LLT high priority links are enabled in the VPC and the VM firewall.

For example, you must enable network ports 50000 to 50006 for 2 high priority links, 50000 to 50007 for 3 high priority links, and so on up to 8 high priority links. These examples are based on the default port number 50000. If the default port number in your environment are different, use the corresponding port range. You can find the default port number mentioned in `/etc/lltab`.

1. (GCP only) When you add multiple interfaces to a VM instance in the Google Cloud you need to set up the routing mechanism that is necessary for your platform.
   - On Linux systems, enable policy-based routing. For details, refer to the Google documentation at:

create-use-multiple-interfaces

- On Windows systems, add persistent routes. For example:

-	VM1	VM2
Subnet	10.208.8.0/22	10.209.4.0/22
Add persistent route	`route add 10.208.4.0 mask 255.255.252.0 10.208.8.1 -p`	`route add 10.208.8.0 mask 255.255.252.0 10.208.4.1 -p`

Ensure that you add persistent routes for every interface on every VM instance.

After the preparatory tasks are complete, run the appropriate installer for your platform on each compute instance. Follow the installer prompts to complete the installation. For details, refer to the *Veritas InfoScale Installation Guide - Linux* or *Veritas InfoScale Installation and Upgrade Guide - Windows*.

On AWS, you can also deploy InfoScale and create the required configurations using the CloudFormation Template (CFT) and the corresponding Amazon Machine Images (AMIs).

B07CYGD14V?qid=1542877554194&sr=0-3

More Information

VPC security groups example

Veritas InfoScale® 8.0.1 Solutions in Cloud Environments - v8.0.1

Installing the AWS CLI package

To install the AWS CLI package

1. Enter the following commands sequentially:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```

```
$ unzip awscli-bundle.zip
```

```
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

1. Create the Identity and Access Management (IAM) role.

For details, refer to the AWS documentation about Identity and Access Management.

1. Configure the AWS CLI.

```
$ aws configure
```

1. Verify that the CLIs are working properly.

```
$ aws ec2 describe-instances
```

VPC security groups example

A security group acts as a virtual firewall for the instance to control inbound and outbound traffic. Therefore, you must make sure that in cluster configuration, ports are enabled in the security groups so that the VMs can communicate with each others.

Consider an example with two VMs that uses two subnets:

Subnet 1: 172.17.1.0/24

Subnet 2: 172.17.2.0/24

With this setup, the following table describes the inbound rules for a security group.

```
Inbound rules


 Type                  Protocol     Port range           Source


ALL UDP                 UDP         0-65535             172.17.1.0/24

All UDP                 UDP         0-65535             172.17.2.0/24
```

This table indicates that the inbound traffic is allowed between the instances that are assigned to the security group.

Configurations for Amazon Web Services - Windows

This section includes the following topics:

- Replication configurations in AWS - Windows
- HA and DR configurations in AWS - Windows

Replication configurations in AWS - Windows

The procedure for setting up replication in AWS depends on where your primary data center and your secondary data center are located.

Follow the procedure that is appropriate for your setup:

- See Replication from on-premises to AWS - Windows.
- See Replication across AZs in an AWS region - Windows.
- See Replication across AWS regions - Windows.

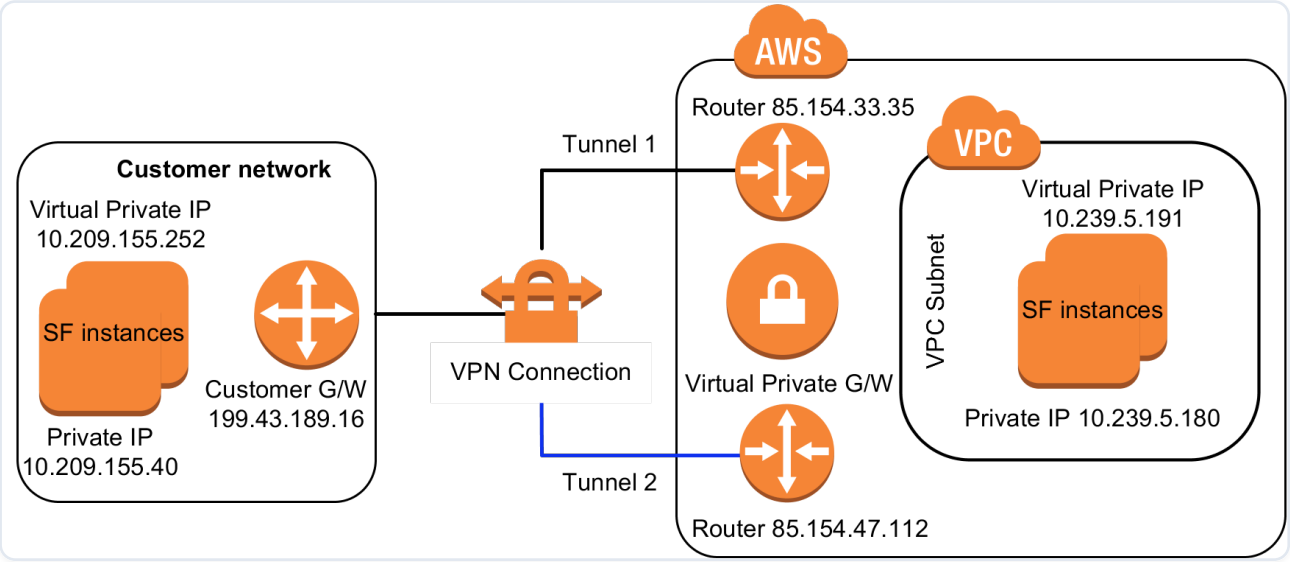
Replication from on-premises to AWS - Windows

In this scenario, data is replicated from an on-premises data center to an on-cloud data center.

Sample configuration

The following graphic illustrates the configuration.

Figure: Sample replication configuration from on-premises to AWS



Prerequisites

Ensure that the following requirements are met before you proceed with the configuration:

- Required ports are open for communication between an on-premise data center and an on-cloud data center.

For details, refer to the *Veritas InfoScale Installation and Upgrade Guide - Windows*.

- The virtual private IP addresses are plumbed on both the nodes.
- The virtual private IP addresses are configured within the subnet.

Setting up replication

The following procedure lists high-level tasks to configure replication from an on-premises data center to a cloud data center.

To set up replication from on-premises to AWS

1. Using AWS portal, create a Virtual Private Cloud (VPC) with a valid CIDR block, for example 10.239.0.0/16.
2. Create a subnet in the VPC created and assign it to an Availability Zone.
3. Create an EC2 instance and associate it with the VPC created.
4. Configure a virtual private gateway and associate it to the VPC.
5. Configure a gateway in the on-premise data center.
6. Create route table entries.
7. Associate the subnet with the route table.
8. Enable route propagation to automatically propagate the routes to the table.

On the Route Propagation tab in the details pane, choose Edit, and select the virtual private gateway that you created.

1. Create a VPN connection.
2. Download the VPN configuration file.
3. Create a VPN tunnel between the on-premise network and the on-cloud network.
4. Install the appropriate InfoScale product on EC2 instances in both the data centers.
5. Create VxVM disk groups, VxVM volumes, Replicated Volume Group (RVG), and RLinks.
6. Set up replication between the on-premise and on-cloud instances.
7. Verify the status of replication.

   ```
   \# vradmin -g dg_name repstatus rvg_name
   ```

   Ensure that the replication status shows:

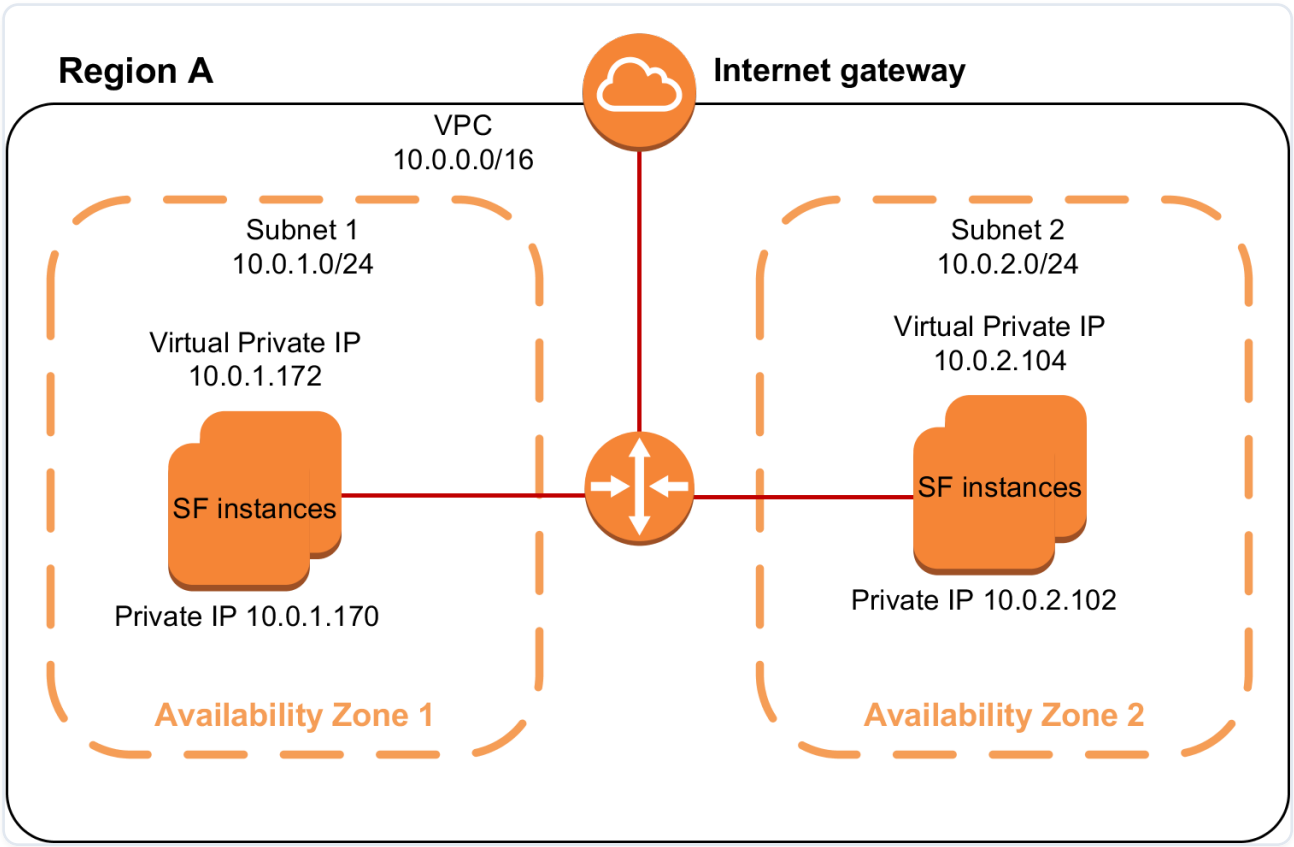
   ```
   Replication status: replicating (connected)
   ```

Replication across AZs in an AWS region - Windows

In this scenario, data is replicated between two on-cloud data centers that are located in different Availability Zones within a same region.

The following diagram illustrates a sample configuration for setting up replication between data centers that are located in different Availability Zones within a same region.

Figure: Replication across Availability Zones



Setting up replication across AZs within the same region

Perform the steps in the following procedure to set up replication across AZs within the same region.

To set up replication across AZs within the same region

1. Create a VPC with a valid CIDR block, for example, 10.0.0.0/16.
2. Create the internet gateway and attach it to the VPC.
3. Modify the VPC route table such that the two instances across availability zones can communicate with each other using private IP addresses.
4. Create two subnets - one subnet for the primary site in AZ1 and the second subnet for the secondary site in AZ2 with valid CIDR range, for example 10.0.1.0/24 and 10.0.2.0/24 respectively.
5. Launch the EC2 instances in the primary and secondary subnets. Install InfoScale on the instances.
6. Verify connectivity between the virtual private IP addresses of the instances.

   ```
   \# ping PIP
   ```

   a. Install the appropriate InfoScale product on EC2 instances in both the data centers.

b. Create VxVM disk groups, VxVM volumes, Replicated Volume Group (RVG), and RLinks.
c. Set up replication between the instances using private IP address or virtual private IP address.
d. Verify the status of replication.

```
\# vradmin -g dg_name repstatus rvg_name
```

Ensure that the status shows:

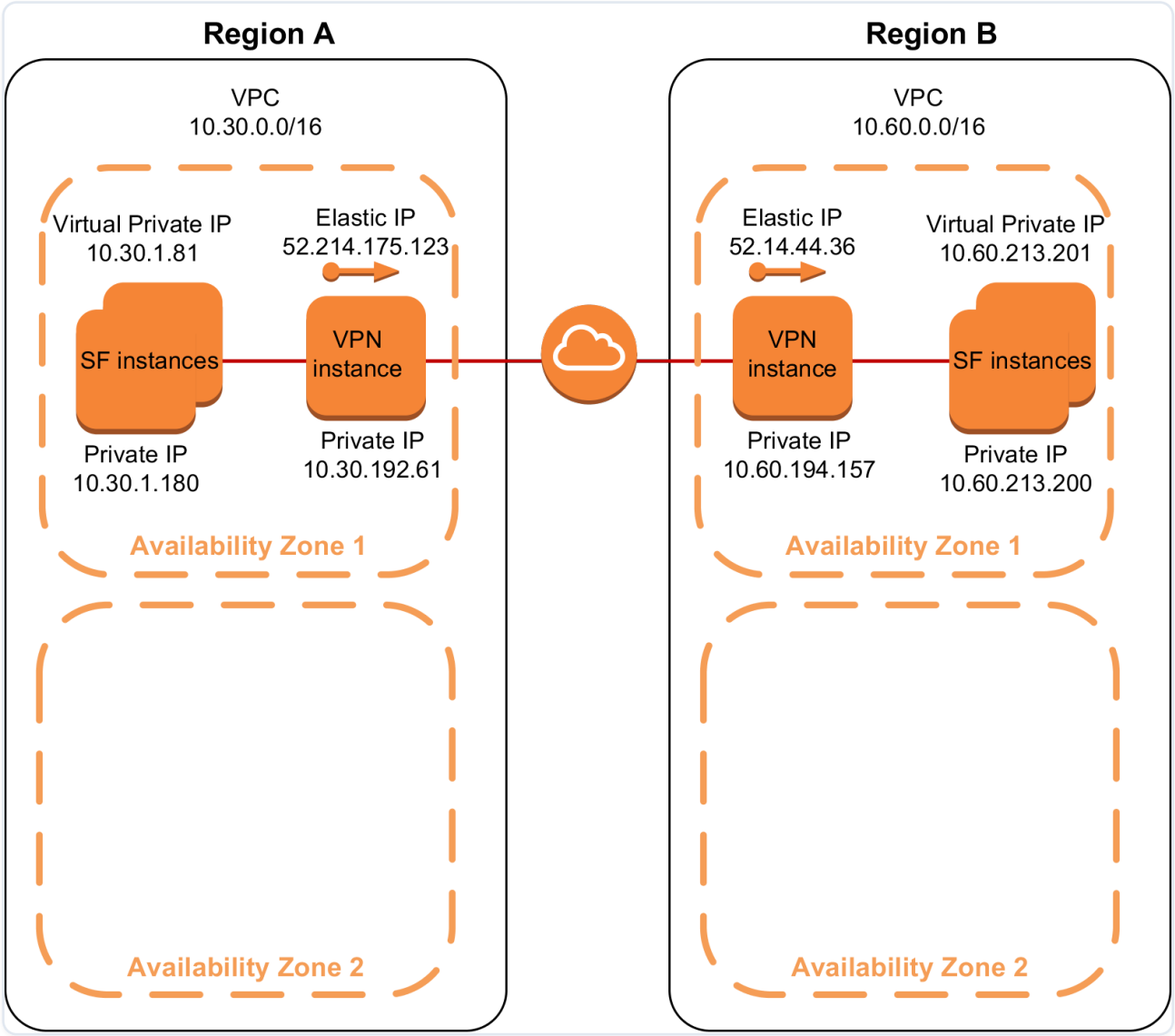
```
Replication status: replicating (connected)
```

Replication across AWS regions - Windows

In this scenario, replication is set up across Availability Zones configured in different regions. The configuration uses software VPN `Openswan/OpenVPN` to connect the VPCs across different regions.

The following diagram illustrates a sample configuration for setting up replication across regions.

Figure: Replication across regions



Setting up replication across regions

Perform the steps in the following procedure to set up replication across regions.

To set up replication across regions

1. Create two VPCs with valid CIDR blocks in different regions, for example, 10.30.0.0/16 and 10.60.0.0/16 respectively.
2. Create a primary site EC2 instance.
3. Create a primary site VPN instance, which belongs to the same VPC as that of the primary EC2 instance.
4. Modify the route table on the primary site. Ensure that the route table entry directs the secondary site traffic through the primary site VPN instance.
5. Create a secondary site EC2 instance.

6. Create a secondary site VPN instance, which belongs to the same VPC as that of the secondary EC2 instance.
7. Modify the route table on the secondary site. Ensure that the route table entry directs the primary site traffic through the secondary site VPN instance.
8. Set up connectivity across regions using software VPN.
9. Install the appropriate InfoScale product on the EC2 instances in both the data centers.
10. Create VxVM disk groups, VxVM volumes, Replicated Volume Group (RVG), and RLinks.
11. Set up replication between the instances using the private IP address or virtual private IP address.
12. Verify the status of replication.

    ```
    \# vradmin -g dg_name repstatus rvg_name
    ```

    Ensure that the replication status shows:

    ```
    Replication status: replicating (connected)
    ```

HA and DR configurations in AWS - Windows

InfoScale Enterprise lets you configure applications for high availability (HA) and disaster recovery (DR) in Amazon Web Services (AWS) cloud.

For more information on the supported use cases, read the following topics:

- See Failover within a subnet of an AWS AZ using virtual private IP - Windows.
- See Failover across AWS subnets using overlay IP - Windows.
- See Public access to InfoScale cluster nodes in AWS using Elastic IP - Windows.
- See DR from on-premises to AWS and across AWS regions or VPCs - Windows.
- See DR from on-premises to AWS - Windows.

InfoScale provides the AWSIP agent and the Route53 agent to manage the network resources for cluster communication, failover, and failback. The agents monitor and manage the IP resources within cluster nodes in the AWS cloud or cluster nodes that are spread across on-premises and AWS. For more information on the agents, see the *Cluster Server Bundled Agents Reference Guide* for your operating system.

Application data is data is stored on the Elastic Block Store (EBS) volumes that are attached to Elastic Compute Cloud (EC2) instances. InfoScale lets you configure replication between EC2 instances, which is further used to support various HA and DR scenarios for applications in AWS.

For more information on the supported replication configurations in AWS, see the following documents:

- *Cluster Server Administrator's Guide - Windows*
- *Volume Replicator Administrator's Guide - Windows*

InfoScale provides configuration wizards that you can use to create, modify, and delete file share and SQL Server HA configurations.

For details about the usage of the File Share Configuration Wizard, refer to the following documents as appropriate:

- *Cluster Server Administrator's Guide - Windows*
- *Storage Foundation and High Availability Solutions, Solutions Guide - Windows*

For details about the usage of the SQL Server Configuration Wizard, refer to the following documents as appropriate:

- *Cluster Server Implementation Guide for Microsoft SQL Server - Windows*
- *Storage Foundation and High Availability Solutions HA and DR Solutions Guide for Microsoft SQL Server - Windows*

Failover within a subnet of an AWS AZ using virtual private IP - Windows

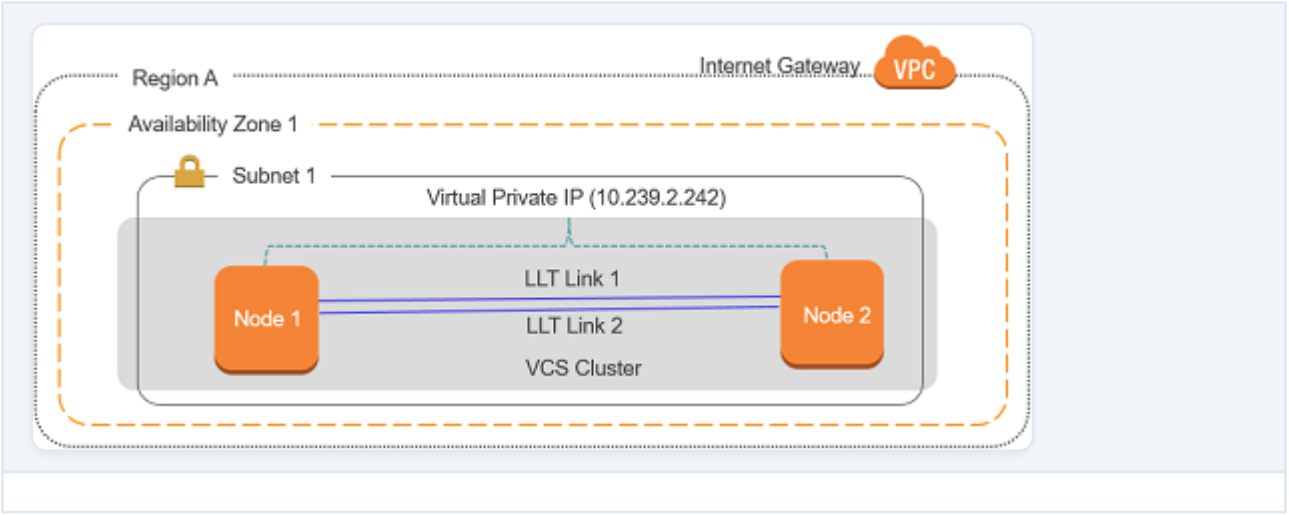
InfoScale clusters let you fail over IPs - and thereby, the application configured for HA - within the same subnet of an AZ.

The following information is required:

- The virtual private IP (secondary private IP) address to be failed over
- The device to which the IP should be plumbed
- The directory in which the AWS CLI is installed; this input is not required if it is provided in the `PATH` environment variable

Sample configuration with private IP

The following graphic depicts a sample failover configuration within the same subnet using a virtual private IP:



The sample configuration includes the following elements:

- A virtual private cloud (VPC) is configured in Region A of the AWS cloud.
- An application is configured for HA using an InfoScale cluster that comprises two nodes, Node1 and Node2, which are EC2 instances.
- Both the cluster nodes exist in the same subnet.
- A virtual private IP is configured, which is failed over from one node to the other as part of the failover or the failback operations.

Sample service group configuration with private IP

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (

        SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1  }

        AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }

        )



        AWSIP awsip (

                PrivateIP = "10.239.3.96"

                Device@WIN-38PNEVJSR2K  = 12-7F-CE-5B-E2-6E
```

```
                Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F

                )

        IP ipres (
                Address = "10.239.3.96"
                SubNetMask = "255.255.254.0"
                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
                )

        NIC nicres (
                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
                )

ipres requires nicres
awsip requires ipres
```

Failover across AWS subnets using overlay IP - Windows

InfoScale clusters let you fail over IPs - and thereby, the application configured for HA - between different subnets in the same AZ or in different AZs.

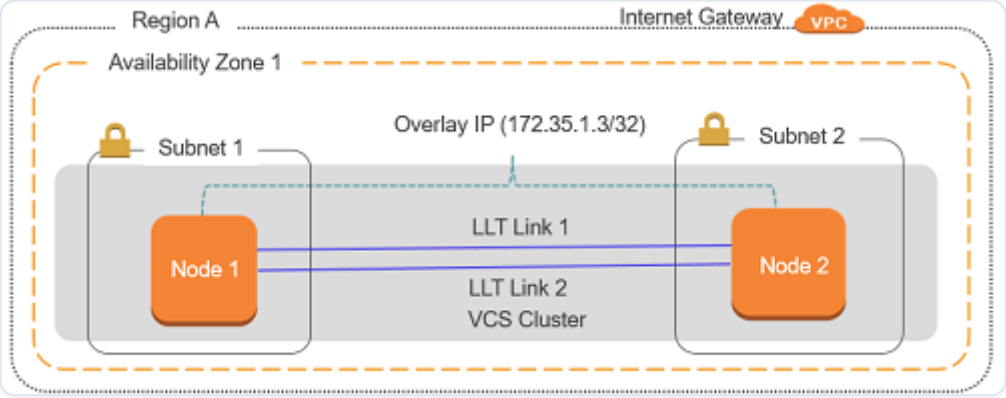
The following information is required:

- The overlay IP address to be used for failover
- The device to which the IP should be plumbed
- The directory in which the AWS CLI is installed; this input is not required if it is provided in the PATH environment variable

AWS does not allow the private IP of one subnet to be failed over to a different subnet. To overcome this limitation, InfoScale Enterprise provides an overlay IP, which is defined at the VPC level, so that it can be used across subnets.

Sample configuration with overlay IP for failover across subnets in the same AZ

The following graphic depicts a sample failover configuration across subnets within the same AZ using an overlay IP:



The sample configuration includes the following elements:

- A virtual private cloud (VPC) is configured in Region A of the AWS cloud.
- An application is configured for HA using an InfoScale cluster that comprises two nodes, Node 1 and Node 2, which are EC2 instances.

Node 1 exists in Subnet 1 and Node 2 exists in Subnet 2.

- An overlay IP is configured, which allows the private IP of a node to be failed over from one subnet to another in an AZ as part of the failover or the failback operations.

Sample service group configuration with overlay IP for failover across subnets in the same AZ

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (

        SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1  }
```

```
        AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }
        )


        AWSIP overlay (
                OverlayIP = "172.16.8.55/32"
                Device @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
                Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F
                RouteTableIds = { rtb-c5272ca3, rtb-fb97ac9d }
                )
        IP ipres (
                Address= "172.16.8.55"
                SubNetMask = "255.255.254.0"
                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
                )

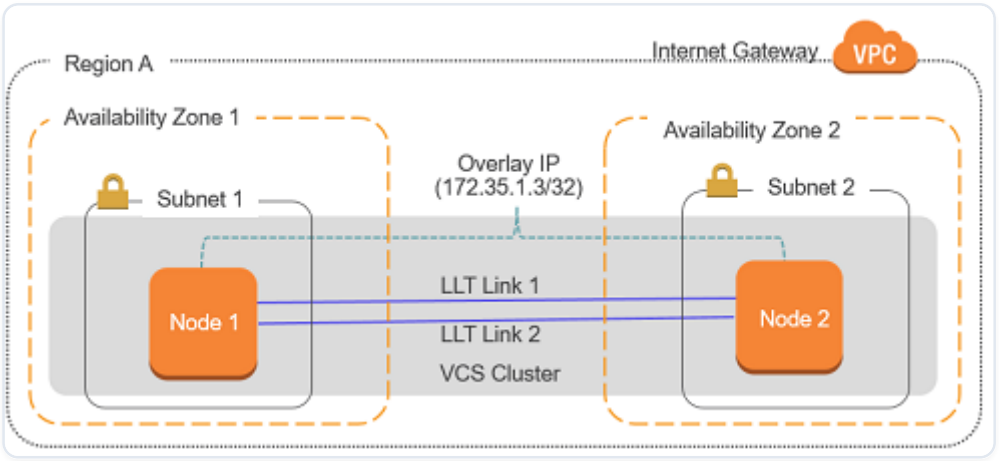

        NIC nicres (
                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
                )


ipres requires nicres
```

```
overlayip requires ipres
```

Sample configuration with overlay IP for failover across subnets in different AZs

The following graphic depicts a sample failover configuration across subnets in different AZs using an overlay IP:



Sample service group configuration with overlay IP for failover across subnets in different AZs

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (

        SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1  }

        AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }

        )


        AWSIP overlay (

                OverlayIP = "172.16.8.55/32"

                Device @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E

                Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F

                RouteTableIds = { rtb-c5272ca3, rtb-fb97ac9d }
```

```
                )

        IP ipres (

                Address = "172.16.8.55"

                SubNetMask = "255.255.254.0"

                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"

                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"

                )


        NIC nicres (

                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"

                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"

                )


ipres requires nicres

overlayip requires ipres
```

Public access to InfoScale cluster nodes in AWS using Elastic IP - Windows

To allow public access to an InfoScale cluster node or to an application configured for HA or DR in AWS, specify the IP to be used in the `ElasticIP` attribute for the AWSIP resource. For example, if you have an application that needs to be highly available and to be accessible globally, you can use the `ElasticIP` attribute of the AWSIP agent to ensure both.

Sample service group configuration with Elastic IP

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AWSIPGrp (
        SystemList = { WIN-38PNEVJSR2K = 0 , WIN-39PNEVJSR2K = 1  }
        AutoStartList = { WIN-38PNEVJSR2K, WIN-39PNEVJSR2K }
        )


        AWSIP elasticIP (
                PrivateIP = "10.239.3.95"
                ElasticIP = "34.193.196.156"
                Device @WIN-38PNEVJSR2K = " 12-7F-CE-5B-E2-6E"

  Device@WIN-39PNEVJSR2K = 12-7F-CE-5B-E2-6F


                )                 )
        IP ipres (
                Address = "10.239.3.95"
                SubNetMask = "255.255.254.0"
                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
                MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
                )


        NIC nicres (
                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
```

```
                    MACAddress @WIN-39PNEVJSR2K = "12:7F:CE:5B:E2:6F"
                    )


ipres requires nicres

elasticIP requires ipres
```

DR from on-premises to AWS and across AWS regions or VPCs - Windows

InfoScale Enterprise lets you use the global cluster option (GCO) for DR configurations. You can use a DR configuration to fail over applications across different regions or VPCs in AWS. The cluster nodes can be in the same AZ or in different AZs.

The following information is required:

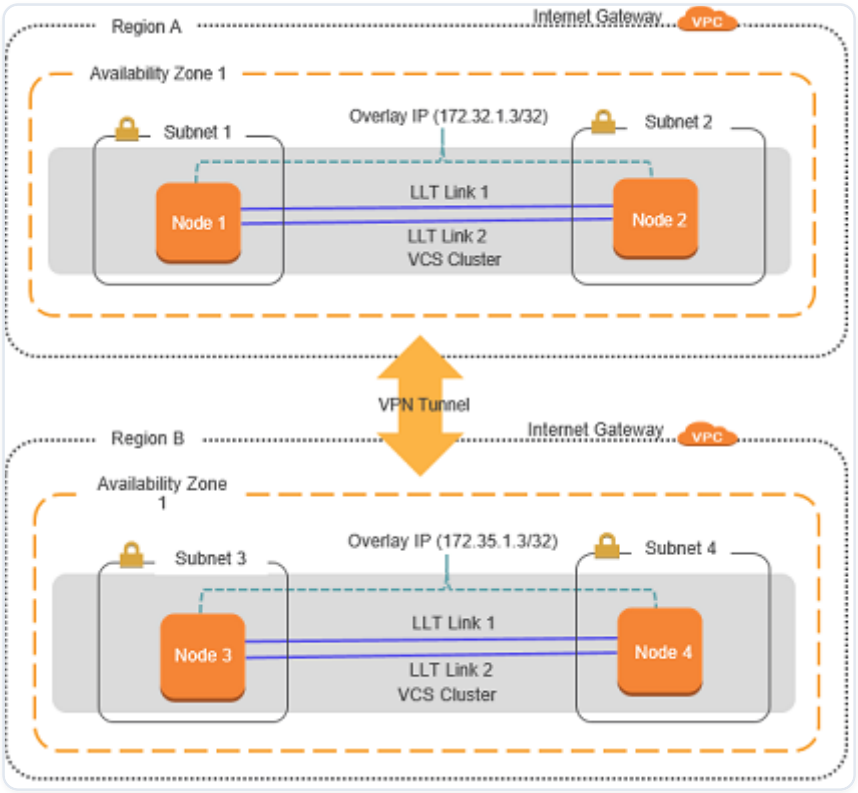
- VPN tunnel information between regions or VPCs
- The IP address to be used for cross-cluster communication:
  - Virtual private IP for cluster the nodes that exist in the same subnet
  - Overlay IP for cluster the nodes that exist in different subnets

You can also use GCO to configure applications for DR from an on-premises site to AWS.

> **Note:** If you use an Amazon VPN tunnel in a global cluster configuration between an on-premises site and AWS, the cluster nodes in the cloud must be in the same subnet.

Sample configuration for DR across regions or VPCs

The following graphic depicts a sample DR configuration across AWS regions:



The sample configuration includes the following elements:

- VPN tunnel between Region A and Region B
- The primary site has the following elements:
  - A virtual private cloud, VPC 1, is configured in Region A of the AWS cloud.
  - An application is configured for HA using an InfoScale cluster that comprises two nodes, Node 1 and Node 2, which are EC2 instances.

Node 1 exists in Subnet 1 and Node 2 exists in Subnet 2.

- The overlay IP allows the private IP of a node to be fail over from one subnet to another in an AZ during failover or failback.
- The secondary site has the following elements:
- A virtual private cloud, VPC 2, is configured in Region B of the AWS cloud.
- The same application is configured for HA on Node 3 and Node 4, which exist in Subnet 3 and Subnet 4 respectively.
- The overlay IP allows the private IP of a node to fail over from one subnet to another in an AZ.

Sample service group configuration for GCO across regions

The following snippet is a service group configuration from a sample VCS configuration file (main.cf) at the primary site (Region A):

```
include "types.cf"



cluster sitever (

     ClusterAddress = "172.32.1.2"

  SecureClus = 1

     )



remotecluster sitecal (

     ClusterAddress = "172.35.1.2"

     ConnectTimeout = 3000

     SocketTimeout = 3000

     )



heartbeat Icmp (

     ClusterList = { sitecal }

     Arguments @sitecal = { "172.35.1.2" }

     )



system ip-172-31-21-156 (

     )



system ip-172-31-61-106 (
```

```
    )


group ClusterService (
    SystemList = { ip-172-31-21-156 = 0, ip-172-31-61-106 = 1 }
  AutoStartList = { ip-172-31-21-156, ip-172-31-61-106 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )


   AWSIP Aws_Ipres (
        OverlayIP = "172.32.1.2/32"
        Device = eth0
        AWSBinDir = "/usr/local/bin"
        )


   Application wac (
        StartProgram = "/opt/VRTSvcs/bin/wacstart -secure"
        StopProgram = "/opt/VRTSvcs/bin/wacstop"
        MonitorProcesses = { "/opt/VRTSvcs/bin/wac -secure" }
        RestartLimit = 3
        )
```

```
    IP Ipres (
         Device = eth0
         Address = "172.32.1.2"
         NetMask = "255.255.255.0"
         )

    NIC gconic (
         Device = eth0
         )

    Aws_Ipres requires Ipres
    Ipres requires gconic
    wac requires Ipres
```

The following snippet is a service group configuration from a sample VCS configuration file (main.cf) at the secondary site (Region B):

```
include "types.cf"

cluster sitecal (
        ClusterAddress = "172.35.1.2"
        SecureClus = 1
        )
```

```
remotecluster sitever (
        ClusterAddress = "172.32.1.2"
        ConnectTimeout = 3000
        SocketTimeout = 3000
        )

heartbeat Icmp (
        ClusterList = { sitever }
        Arguments @sitever = { "172.32.1.2" }
        )

system ip-172-34-20-109 (
        )

system ip-172-34-30-231 (
        )

group ClusterService (
        SystemList = { ip-172-34-20-109 = 0, ip-172-34-30-231 = 1 }
        AutoStartList = { ip-172-34-20-109, ip-172-34-30-231 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
```

```
        )


AWSIP Aws_Ipres (
        OverlayIP = "172.35.1.2/32"
        Device = eth0
        AWSBinDir = "/usr/local/bin"
        )


Application wac (
        StartProgram = "/opt/VRTSvcs/bin/wacstart -secure"
        StopProgram = "/opt/VRTSvcs/bin/wacstop"
        MonitorProcesses = { "/opt/VRTSvcs/bin/wac -secure" }
        RestartLimit = 3
        )


IP Ipres (
        Device = eth0
        Address = "172.35.1.2"
        NetMask = "255.255.255.0"
        )


NIC gconic (
```

```
        Device = eth0

        )



Aws_Ipres requires Ipres

Ipres requires gconic

wac requires Ipres
```

DR from on-premises to AWS - Windows

InfoScale Enterprise lets you use the global cluster option (GCO) for DR configurations. You can use a DR configuration to fail over applications from on-premise cluster to AWS.

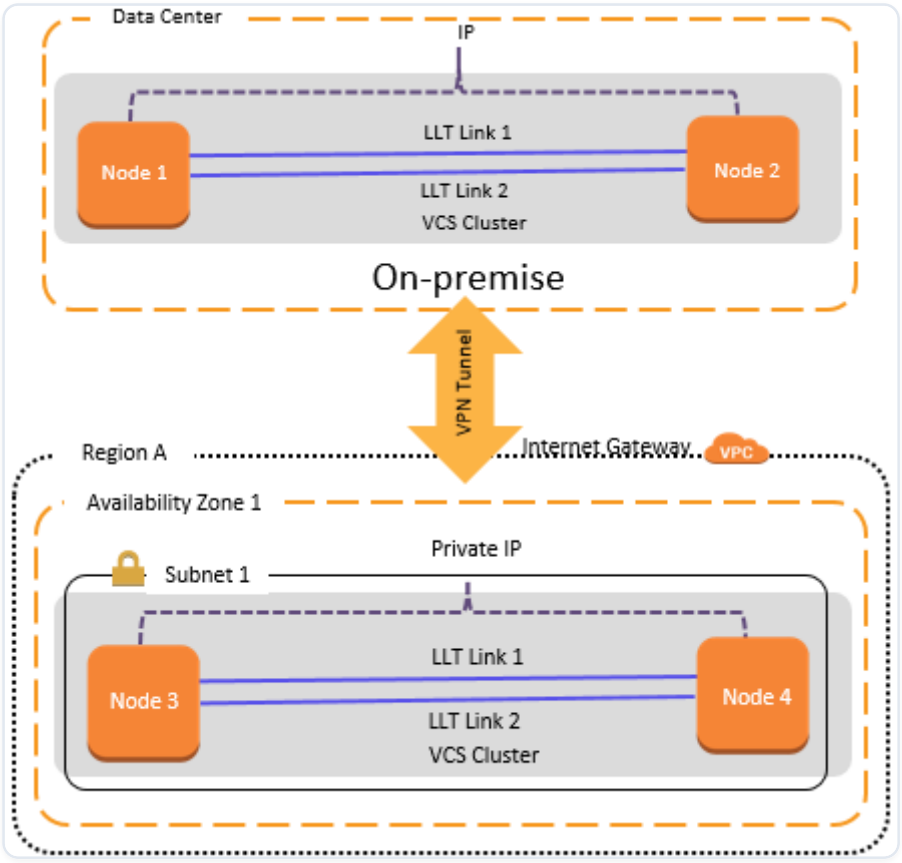
The following information is required:

- VPN tunnel information between regions or VPCs
- The Virtual private IP address to be used for cross-cluster communication.

> **Note:** If you use an Amazon VPN tunnel in a global cluster configuration between an on-premises site and AWS, the cluster nodes in the cloud must be in the same subnet.

Sample configuration for DR across regions or VPCs

The following graphic depicts a sample DR configuration from on-premise cluster to AWS cloud:



The sample configuration includes the following elements:

- VPN tunnel between on-premise data center and Region A
- The primary site has the following elements:
  - Cluster nodes in the same subnet
  - Virtual private IP for cross-cluster communication
- The secondary site has the following elements:
  - A virtual private cloud, VPC, is configured in Region Aof the AWS cloud
  - The same application is configured for HA on Node 3 and Node 4, which exist in the same subnet

Sample service group configuration for GCO from on-premise to cloud

The following snippet is a service group configuration from a sample VCS configuration file (main.cf) for a single node:

```
include "types.cf"
```

```
cluster vvr_aws (
          ClusterAddress = "10.239.3.96"
          SecureClus = 1
          )


remotecluster vvr_cloud (
          ClusterAddress = "10.209.57.98"
          ConnectTimeout = 30000
          SocketTimeout = 30000
          )


heartbeat Icmp (
          ClusterList = { vvr_cloud }
          AYARetryLimit = 10
          Arguments @vvr_cloud = { "10.209.57.98" }
          )


system WIN-38PNEVJSR2K (
          )


group ClusterService (
          SystemList = { WIN-38PNEVJSR2K = 0 }
```

```
        AutoStartList = { WIN-38PNEVJSR2K }

        )



        AWSIP gcoawsip (

                PrivateIP = "10.239.3.96"

                Device = 12-7F-CE-5B-E2-6E

                )



        IP csg_ip (

                Address = "10.239.3.96"

                SubNetMask = "255.255.254.0"

                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"

                )



        NIC csg_nic (

                MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"

                )



        Process wac (

                StartProgram @WIN-38PNEVJSR2K = "\"C:\\Program Files\\


               Veritas\\Cluster Server\\bin\\wac.exe\""
```

```
                    StopProgram @WIN-38PNEVJSR2K = "\"C:\\Program Files\\

                  Veritas\\Cluster Server\\bin\\wacstop.exe\""

                    MonitorProgram @WIN-38PNEVJSR2K = "\"C:\\Program Files\
\

                Veritas\\Cluster Server\\bin\\wacmonitor.exe\""

                     )


            csg_ip requires csg_nic

            wac requires gcoawsip

            wac requires csg_ip


group fileshare (

            SystemList = { WIN-38PNEVJSR2K = 0 }

            ClusterList = { vvr_cloud = 1, vvr_aws = 0 }

            Authority = 1

            ClusterFailOverPolicy = Auto

            )


            AWSIP vvrawsip (

                     PrivateIP = "10.239.3.97"

                     Device @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
```

```
        )


FileShare fileshare-FileShare (
        PathName = "\\"
        ShareName = testshare
        LanmanResName = fileshare-Lanman
        MountResName = mounvres
        UserPermissions = { Everyone = FULL_CONTROL }
        )


IP fileshare-IP (
        Address = "10.239.3.97"
        SubNetMask = "255.255.254.0"
        MACAddress @WIN-38PNEVJSR2K = 12-7F-CE-5B-E2-6E
        )


Lanman fileshare-Lanman (
        VirtualName = FILESHARE-DHAWA
        IPResName = fileshare-IP
        )


MountV mounvres (
```

```
        MountPath = "e:"
        VolumeName = datavol
        VMDGResName = vmnsdg
        )


NIC fileshare_NIC (
        MACAddress @WIN-38PNEVJSR2K = "12:7F:CE:5B:E2:6E"
        )


RVGPrimary rvgprimary_1 (
        RvgResourceName = vvrsg-VvrRvg
        AutoResync = 1
        )


requires group vvrsg online local hard
vvrawsip requires fileshare_NIC
fileshare-FileShare requires mounvres
fileshare-FileShare requires fileshare-Lanman
fileshare-IP requires vvrawsip
fileshare-Lanman requires fileshare-IP
mounvres requires rvgprimary_1
```

```
group vvrsg (
        SystemList = { WIN-38PNEVJSR2K = 0 }
        AutoStartList = { WIN-38PNEVJSR2K }
        )


        Proxy proxy_ip (
                TargetResName = csg_ip
                )


        Proxy proxy_nic (
                TargetResName = csg_nic
                )


        VMNSDg vmnsdg (
                DiskGroupName = vvrdg
                DGGuid = 03d725b1-fca3-49b2-b722-38984835e6b0
                )


        VvrRvg vvrsg-VvrRvg (
                RVG = rvg
                VMDgResName = vmnsdg
                IPResName = proxy_ip
```

```
                )


        vvrsg-VvrRvg requires proxy_nic

        vvrsg-VvrRvg requires proxy_ip

        vvrsg-VvrRvg requires vmnsdg
```

Configurations for Microsoft Azure - Windows

This section includes the following topics:

- Replication configurations in Azure - Windows
- HA and DR configurations in Azure - Windows

Replication configurations in Azure - Windows

The procedure for setting up replication in Azure depends on where your primary data center and your secondary data center are located.

Follow the procedure that is appropriate for your setup:

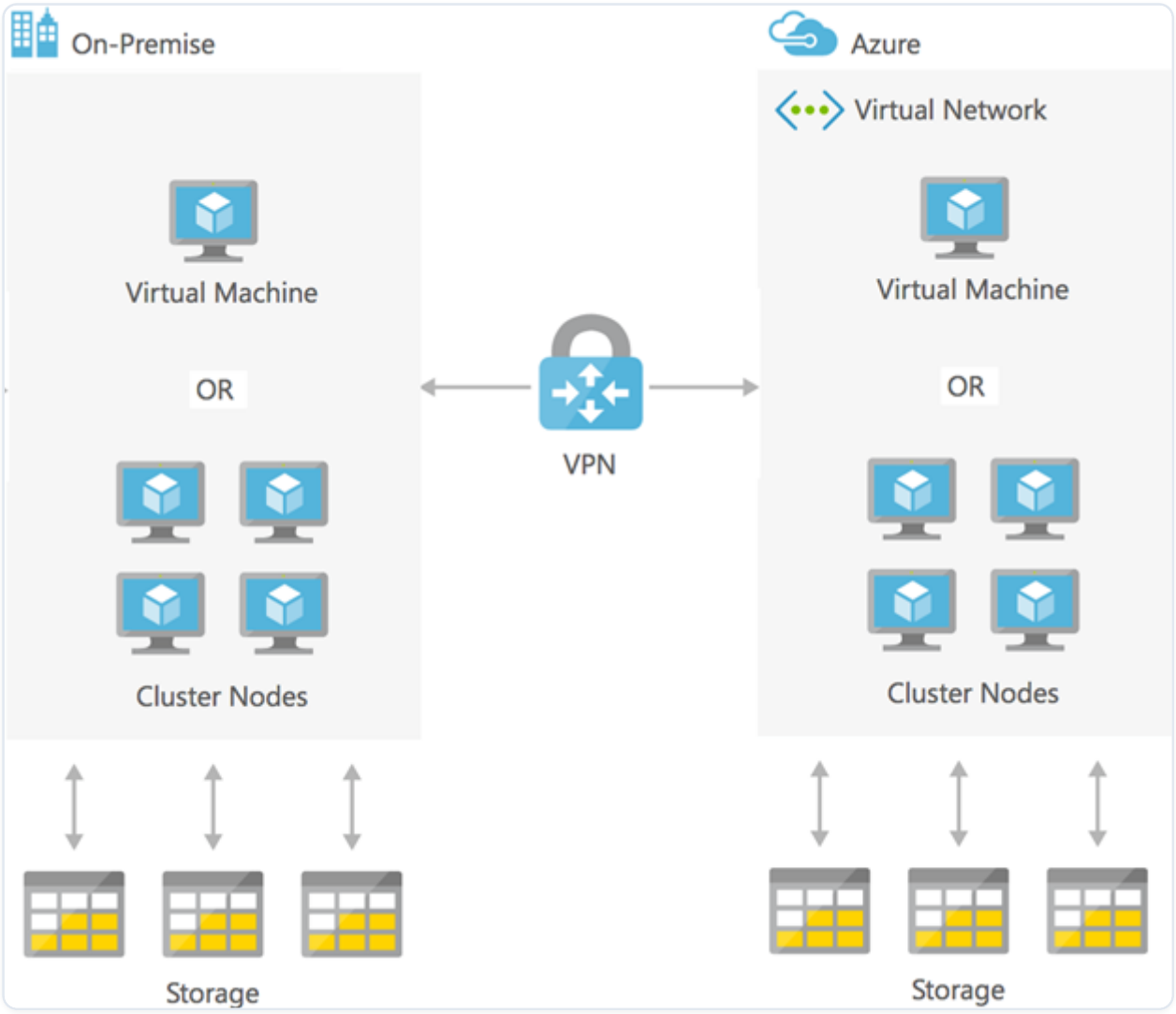
- See Replication from on-premises to Azure - Windows.
- See Replication within an Azure region - Windows.
- See Replication across Azure regions - Windows.

Replication from on-premises to Azure - Windows

The following diagram illustrates the sample configuration for setting up replication between an on-premises data center to an Azure data center:

> **Note:** For the ease of use, the machines, whether virtual or physical, are commonly mentioned as virtual machines. Your on-premises data center may include physical machines instead of the virtual machines. In any case, the procedure to set up replication from an on-premises data center to a cloud data center remain the same for physical and well as virtual machines.

> Figure: Sample configuration for setting up replication between an on-premises data center to cloud data center



About setting up replication between an on-premise data center to on-cloud data center

Replication between an on-premise data center to on-cloud data center involves the following high-level steps:

1. Prepare the setup at on-premise data center
2. Prepare the setup at on-cloud data center
3. Establish a tunnel from on-premise data center to on-cloud data center
4. Deploy setup

The following sections provide details about performing each of these steps.

Preparing the setup at on-premise data center

To prepare the setup at on-premise data center

1. Enable the ports that are used for inbound and outbound communication.

For details, refer to the *Veritas InfoScale Installation and Upgrade Guide - Windows*.

1. Create a subnet and a local VPN gateway.
2. Note the address space that is allotted for the subnet and the public IP address that is allotted for the local VPN gateway.

Preparing the setup at on-cloud data center

To prepare the setup at on-cloud data center

1. Using Microsoft Azure portal, create a resource group.
2. Create a VNet in the resource group created and specify an IP address space for the VNet.

The IP address range must be diff on the on-premise subnet and on the on-cloud subnet.

1. Create a gateway subnet.
2. Create a VPN gateway and associate it with the created VNet.

Note the public IP address that is allotted for the on-cloud VPN gateway.

1. Create a local network gateway.

When you create the local network gateway, you must provide the on-premise subnet IP address range and the public IP address of the on-premise local VPN gateway.

1. Establish a tunnel from on-cloud to on-premise network.

To establish the tunnel, create a connection of type **Site-to-Site (IPSec)** and choose the on-cloud VPN gateway and the local network gateway.

1. Provide a shared key (alpha-numeric key).

A shared key is a pass-phrase. This pass-phrase is required when you establish a tunnel from on-premise data center to cloud data center.

Establishing a tunnel from on-premise data center to on-cloud data center

To establish a tunnel from on-premise data center to on-cloud data center, use the following parameters:

- Public IP address that is allotted for the on-cloud VPN gateway
- Shared key (alpha-numeric key) that was provided while establishing a tunnel from on-cloud data center to on-premise data center
- On-cloud VPN gateway configuration type (Policy based or Route based)

Deploying the setup

To deploy the setup (in both the data centers)

1. Create virtual machines in the subnets created.
2. Provision storage.
3. Install the appropriate InfoScale product.
4. Create VxVM disk groups, VxVM volumes, Replicated Volume Group (RVG), and RLinks.
5. Flush the iptables on both the virtual machines.

```
# iptable -F
```

1. Set up replication between the virtual machines using the private IP address or the virtual IP address.
2. Verify the replication status.

```
# vradmin -g dg_name repstatus rvg_name
```

Ensure that the replication status shows:

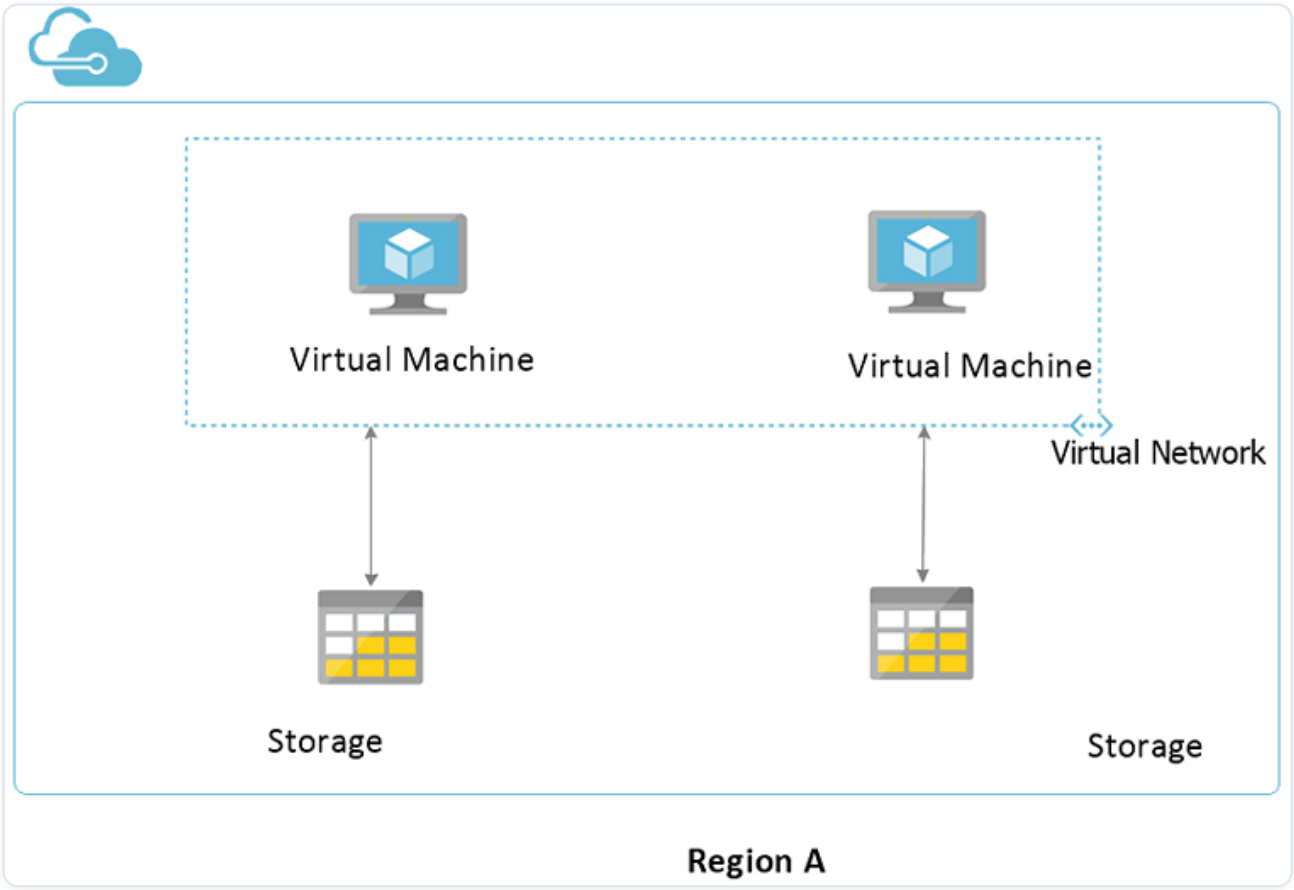
```
Replication status: replicating (connected)
```

Replication within an Azure region - Windows

In an Azure cloud environment, in a single region, you can provision your setup across virtual networks (VNets) or within a VNet.

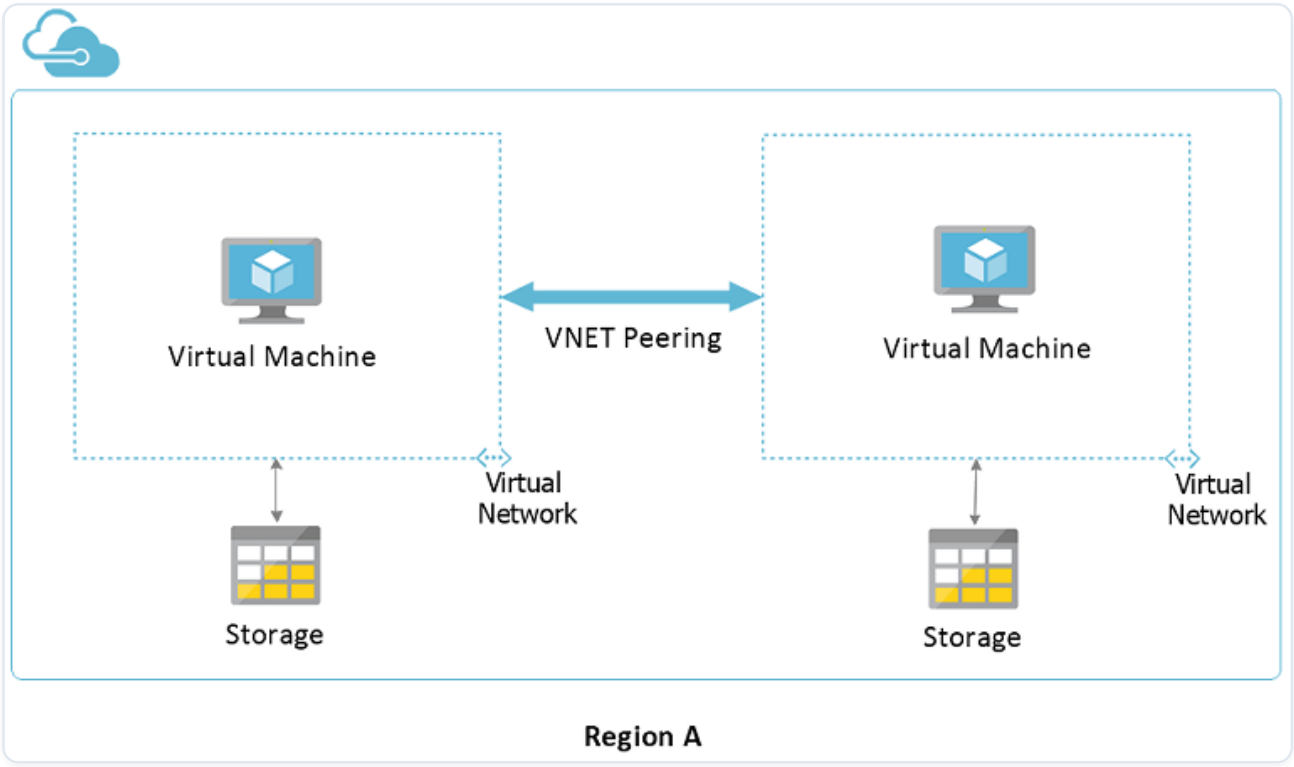
The following diagram illustrates the sample configuration for setting up replication between the same VNet:

Figure: Sample configuration for setting up replication between same VNet



The following diagram illustrates the sample configuration for setting up replication across VNets:

Figure: Sample configuration for setting up replication across VNets



Setting up replication in the same VNet

To set up replication in the same VNet within the same region

1. Enable the ports that are used for inbound and outbound communication.

For details, refer to the *Veritas InfoScale Installation and Upgrade Guide - Windows*.

1. Using Microsoft Azure portal, create a VNet and specify an IP address space for the VNet.
2. Create a subnet in the VNet created.

For details about creating a VNet, specifying an IP address space, and creating a subnet, refer to Microsoft documentation.

1. Create two virtual machines within the subnets and provision storage.
2. Install the appropriate InfoScale product on both the virtual machines.
3. Create VxVM disk groups, VxVM volumes, Storage Replicator Log (SRL), Replicated Volume Group (RVG), and RLinks.
4. Flush the iptables on both the virtual machines.

```
# iptable -F
```

1. Set up replication between the virtual machines using the private IP address or the virtual IP address.
2. Verify the replication status.

```
# vradmin -g dg_name repstatus rvg_name
```

Ensure that the replication status shows:

```
Replication status: replicating (connected)
```

Setting up replication across VNet

To set up replication across the VNets, within the same region

1. Using Microsoft Azure portal, create two VNets and specify an IP address space for each VNet.
2. Set up VNet Peering between the two VNets.
3. Create a subnet in each VNet.
4. Create a virtual machine in each subnet and provision storage.
5. Install the appropriate InfoScale product on both the virtual machines.
6. Create VxVM disk groups, VxVM volumes, Storage Replicator Log (SRL), Replicated Volume Group (RVG), and RLinks.
7. Flush the iptables on both the virtual machines.

```
# iptable -F
```

1. Set up replication between the virtual machines using the private IP address or the virtual IP address.
2. Verify the replication status.

```
# vradmin -g dg_name repstatus rvg_name
```

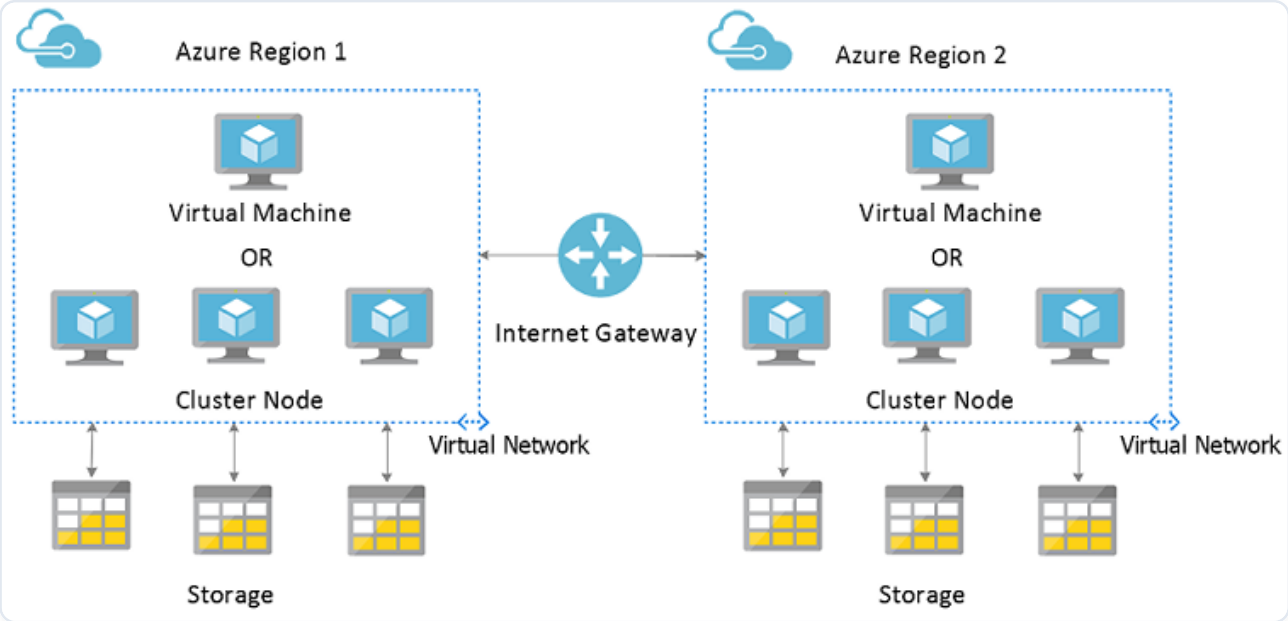
Ensure that the replication status shows:

```
Replication status: replicating (connected)
```

Replication across Azure regions - Windows

The following diagram illustrates the sample configuration for setting up replication across regions:

Figure: Sample configuration for setting up replication across regions



Setting up replication across regions

To set up replication across region

1. Enable the ports that are used for inbound and outbound communication.

For details, refer to the *Veritas InfoScale Installation and Upgrade Guide - Windows*.

1. Using Azure portal, create a resource group (RG) in both the regions.
2. Create a VNet in each region and specify a non-overlapping IP address space for each VNet.
3. Create a subnet and a gateway subnet under the VNets created in both the regions.

4. Create a virtual network gateway in both the regions.
5. Establish a connection in between the two virtual network gateways.
6. Create virtual machines in the subnets and provision storage.
7. Install the InfoScale product.
8. Create VxVM disk groups, VxVM volumes, Replicated Volume Group (RVG), and RLinks.
9. Flush the iptables on both the virtual machines.

```
# iptable -F
```

1. Set up replication using the private IP address or the virtual IP address.
2. Verify the replication status.

```
# vradmin -g dg_name repstatus rvg_name
```

Ensure that the replication status shows:

```
Replication status: replicating (connected)
```

HA and DR configurations in Azure - Windows

InfoScale Enterprise lets you configure applications for high availability (HA) and disaster recovery (DR) in Azure cloud environment.

For more information on the supported use cases, read the following topics:

- See Failover within an Azure subnet using private IP - Windows.
- See Failover across Azure subnets using overlay IP - Windows.
- See Public access to cluster nodes in Azure using public IP - Windows.
- See DR from on-premises to Azure and across Azure regions or VNets - Windows.

InfoScale Enterprise provides the AzureIP agent and the Azure DNS zone agent to manage the network resources for cluster communication, failover, and failback. These agents monitor and manage the IP resources within cluster nodes in the Azure cloud. The AzureAuth agent handles the Azure related authentication. For more information on the agents, see *Cluster Server Bundled Agents Reference Guide - Windows.*

> **Note:** Azure recommends that you configure Azure VMs in the same Availability Set to ensure that at least one node from the cluster is always operational.

> In an Azure environment, application data is stored on the Azure data disks that are attached to the Azure virtual machines. The AzureDisk agent provides high availability of the Azure disks during fail-over of an application. This agent supports high availability of managed and unmanaged disks.

VCS lets you configure replication between Azure virtual machines, which is further used to support various HA and DR scenarios for applications in Azure cloud.

For more information on the supported replication configurations in Azure, see the following documents:

- *Cluster Server Administrator's Guide - Windows*
- *Volume Replicator Administrator's Guide - Windows*

Failover within an Azure subnet using private IP - Windows

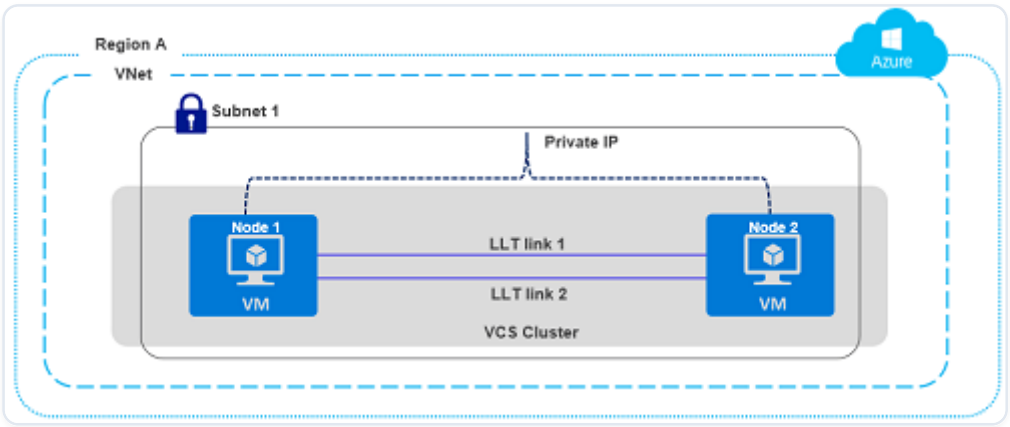
InfoScale clusters let you fail over IPs - and thereby, the application configured for HA - within the same subnet in the same VNet.

The following information is required:

- A private IP (secondary private IP) address to be failed over
- The device to which the IP should be plumbed

Sample configuration with private IP

The following graphic depicts a sample failover configuration within the same subnet using a private IP:



The sample configuration includes the following elements:

- A Azure virtual network (VNet) is configured in Region A of the Azure cloud

- An application is configured for HA using a cluster that comprises two nodes, Node1 and Node2, which are Azure virtual machines
- Both the cluster nodes exist in the same subnet
- A private IP is configured, which is failed over from one node to the other as part of the failover or the failback operations

Sample service group configuration with private IP

The following snippet is a service group configuration from a sample configuration file (main.cf):

```
group AzureAuthGrp (

    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }

    Parallel = 1

    )



    AzureAuth azureAuth (

        SubscriptionId = 640a326-fga6-90gh-b616-c1e9bb

        ClientId = e8d899-d32a-47d04-8986-be739104d

        SecretKey = fntPgnUnhTprQrqTRonSlpRhnGrrNklFngLs

        TenantId = 9fjkabae-2348-4308-b503-6667d61

        )



        Phantom phres (

        )



group AzureIPGrp (
```

```
    SystemList = { AzureVM1 = 0, AzureVM2 = 1 }

    )



        IP IP_res (

            Address = "10.1.5.67"

            SubNetMask = "255.255.255.0"

            MACAddress @AzureVM1 = 00-0D-3A-91-73-A0

            MACAddress @AzureVM2 = 00-0D-3A-92-03-DC

            )



        NIC NIC_res (

            MACAddress @AzureVM1 = 00-0D-3A-91-73-A0

            MACAddress @AzureVM2 = 00-0D-3A-92-03-DC

            )



        AzureIP AzureIP_res (

            PrivateIP = "10.1.5.42"

            MACAddress @AzureVM1 = 00-0D-3A-91-73-A0

            MACAddress @AzureVM2 = 00-0D-3A-92-03-DC

            AzureAuthResName = azureAuth

            )
```

```
IP_res requires AzureIP_res

AzureIP_res requires NIC_res
```

Failover across Azure subnets using overlay IP - Windows

InfoScale clusters let you fail over IPs - and thereby, the application configured for HA - between different subnets in the same VNet.

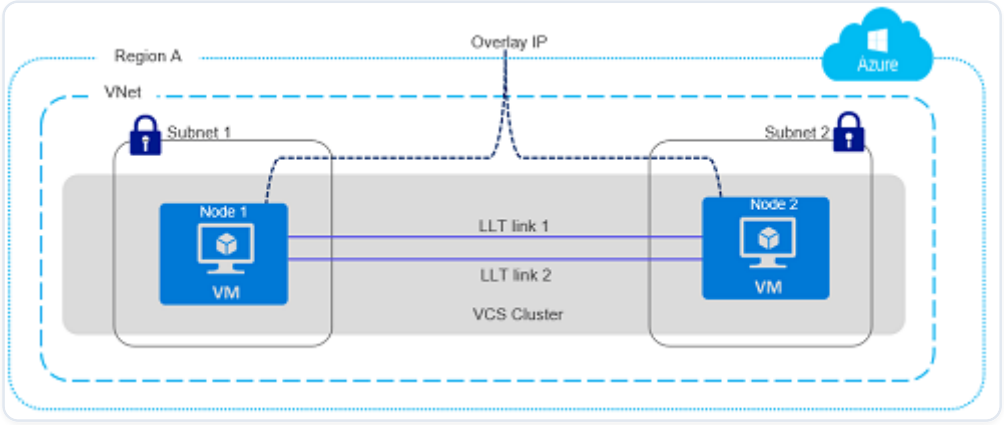
The following information is required:

- The IP address outside VNet to be used for failover
- The device to which the IP should be plumbed

Azure does not allow the private IP of one subnet to be failed over to a different subnet. To overcome this limitation, provide an overlay IP, which is outside the VNet level, so that it can be used across subnets.

Sample configuration with Overlay IP for failover across subnets in the same VNet

The following graphic depicts a sample failover configuration across subnets within the same VNet using an overlay IP:



The sample configuration includes the following elements:

- A Azure virtual network (VNet) is configured in Region A of the Azure cloud
- An application is configured for HA using a cluster that comprises two nodes, Node1 and Node2, which are Azure virtual machines
- Node 1 exists in Subnet 1 and Node 2 exists in Subnet 2.

- An overlay IP is configured to allow redirecting IP address traffic to another cluster node belonging to different subnet within the same VNet as a part of the failover or the failback operations

Sample service group configuration with overlay IP for failover across subnets in the same VNet

The following snippet is a service group configuration from a sample VCS configuration file (main.cf):

```
group AzureAuthGrp (

                SystemList = { AzureVM1 = 0, AzureVM2 = 1 }

                Parallel = 1

                )



                AzureAuth azureAuth (

                      SubscriptionId = 640a326-fga6-90gh-b616-c1e9bb

                      ClientId = e8d899-d32a-47d04-8986-be739104d

                      SecretKey = fntPgnUnhTprQrqpiNtrItpRhnGrrNklFngLs

                      TenantId = 9fjkabae-2348-4308-b503-6667d61

                                )



                Phantom phres (

                                )



group AzureOverlayIPGrp (

                SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
```

```
                )

                IP IP_res (
                                Address = "192.168.3.88"
                                SubNetMask = "255.255.255.0"
                                MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
                                MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
                                )

                NIC NIC_res (
                                MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
                                MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
                                )

                AzureIP AzureOverlayIPres (
                         MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
                         MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
                         OverlayIP = "192.168.3.88"
                         RouteTableResourceIds = {
                             "/subscriptions/640a326-fga6-90gh-b616-
c1e9bb/
                                                  resourceGroups/TestRG/
```

```
providers/Microsoft.Network/


                                                routeTables/
testRoute_eastUS2",

                              "/subscriptions/640a326-fga6-90gh-b616-
c1e9bb/


                                             resourceGroups/TestRG2/
providers/Microsoft.Network/


                                          routeTables/route1" }

                             AzureAuthResName = azureAuth

                               )

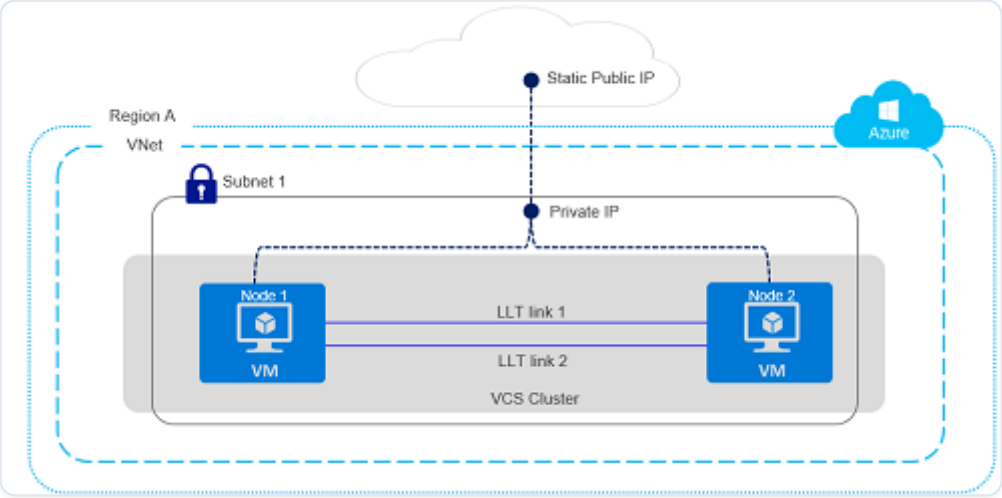


                IP_res requires AzureOverlayIPres

                AzureOverlayIPres requires NIC_res
```

Public access to cluster nodes in Azure using public IP - Windows

To allow public access to a cluster node or to an application configured for HA in Azure environment, specify the IP in the PublicIP attribute for the AzureIP resource. This IP is used to map the Public IP address to a secondary private IP address. For example, if you have an application that needs to be highly available and to be accessible globally, you can use the PublicIP attribute of the AzureIP agent to ensure both.



Sample service group configuration with Public IP

```
group AzureAuthGrp (

                SystemList = { AzureVM1 = 0, AzureVM2 = 1 }

                Parallel = 1

                )



                AzureAuth azureAuth (

                    SubscriptionId = 640a326-fga6-90gh-b616-c1e9bb

                    ClientId = e8d899-d32a-47d04-8986-be739104d

                    SecretKey = fnlhPhrQpiNtrItpRhnGrrNklFngLs

                    TenantId = 9fjkabae-2348-4308-b503-6667d61

                                )



                Phantom phres (

                                )
```

```
group AzurePublicIPGrp (
                SystemList = { AzureVM1 = 0, AzureVM2 = 1 }
                )


                IP IP_res (
                     Address = "10.1.5.67"
                     SubNetMask = "255.255.255.0"
                     MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
                     MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
                          )


                NIC NIC_res (
                     MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
                     MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
                            )


                AzureIP AzurePublicIPres (
                     PrivateIP = "10.1.5.42"
                     MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
                     MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
                     PublicIP = "52.165.223.2"
                     AzureAuthResName = azureAuth
```

```
                    )


        IP_res requires AzurePublicIPres

        AzurePublicIPres requires NIC_res
```

DR from on-premises to Azure and across Azure regions or VNets - Windows

VCS lets you use the global cluster option (GCO) for DR configurations. You can use a DR configuration to fail over applications across different regions or VNets in Azure or between an on-premises site and Azure.

The following is required for on-premise to cloud DR using VPN tunneling:

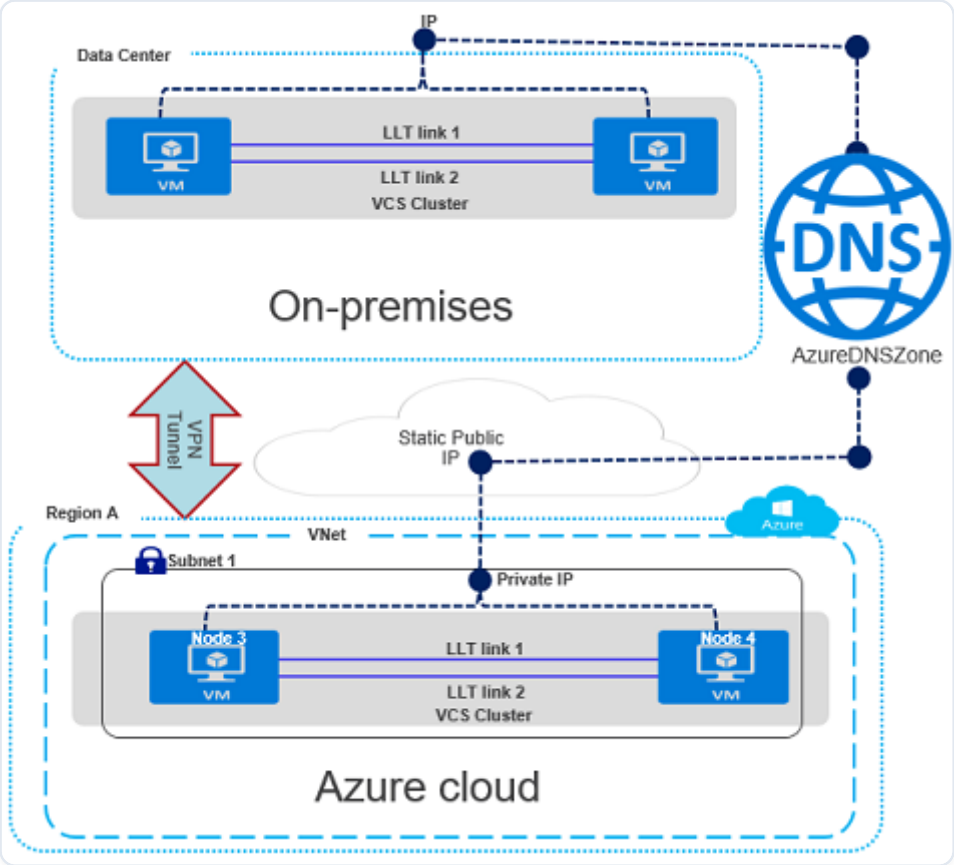
- Prepare the setup at on-premise data center
- Prepare the setup at on-cloud data center
- Establish a VPN tunnel from on-premise data center to cloud data center
- Virtual private IP for cluster nodes that exist in the same subnet. The IP address is used for cross-cluster communication

The following is required for region to region DR using VNet peering:

- Prepare the setup at both the data centers in the regions
- Establish a VNet peering from one region to the other region
- Virtual private IP for cluster nodes that exist in the same subnet. The IP address is used for cross-cluster communication

> **Note:** If you use an VPN tunnel between an on-premises site and Azure or you use VNet peering between Azure regions, the cluster nodes in the cloud must be in the same subnet.

Sample configuration with private IP



The sample configuration includes the following elements:

- VPN tunnel between on-premise data center and Region A
- The primary site has the following elements:
  - Cluster nodes in the same subnet
  - Virtual private IP for cross-cluster communication
- The secondary site has the following elements:
  - A VNet is configured in Region A of the Azure cloud
  - The same application is configured for HA on Node 3 and Node 4, which exist in Subnet
  - Virtual private IP for cross-cluster communication

Sample service group configuration with private IP

The following snippet is a service group configuration from a sample configuration file (main.cf):

```
cluster vcs_cluster1 (

    ClusterAddress = "10.3.3.100"

    SecureClus = 1
```

```
)


remotecluster vcs_cluster2 (
    ClusterAddress = "10.5.0.5"
)


heartbeat Icmp (
    ClusterList = { vcs_cluster2 }
    Arguments @vcs_cluster2 = { "10.5.0.5" }
)


system AzureVM1(
)
system AzureVM2(
)


group AzureAuthGrp (
    SystemList = { AzureVM1= 0, AzureVM2= 1 }
    Parallel = 1
)


    AzureAuth azurauth (
```

```
        SubscriptionId = 6940a326-abc6-40dd-b717-c1e9bcdf1d63
        ClientId = 8c891a8c-ghi2-473b-bbec-035ac50fb896
        SecretKey = gsiOssRooSpShuNoiQioNsjQlqHovUosQsrMt
        TenantId = 96dcased-0448-4308-b505-6789d61dd0e3
    )

    Phantom phres (
    )

group ClusterService (
    SystemList = { AzureVM1= 0, AzureVM2= 1 }
    AutoStartList = { AzureVM1, AzureVM2}
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
)

    Process wac (
        StartProgram @ AzureVM1 = "\"C:\\Program Files\\Veritas\\
                            Cluster Server\\bin\\wac.exe\""
        StartProgram @ AzureVM2 = "\"C:\\Program Files\\Veritas\\
                            Cluster Server\\bin\\wac.exe\""
        StopProgram @ AzureVM1 = "\"C:\\Program Files\\Veritas\\
```

```
                            Cluster Server\\bin\\wacstop.exe\""

        StopProgram @ AzureVM2 = "\"C:\\Program Files\\Veritas\\

                            Cluster Server\\bin\\wacstop.exe\""

        MonitorProgram @ AzureVM1 = "\"C:\\Program Files\\Veritas\\

                            Cluster Server\\bin\\wacmonitor.exe\""

        MonitorProgram @ AzureVM2 = "\"C:\\Program Files\\Veritas\\

                            Cluster Server\\bin\\wacmonitor.exe\""



    AzureIP azureipres (

        PrivateIP = "10.3.3.100"

MACAddress @AzureVM1 = 00-0D-3A-91-73-A0

MACAddress @AzureVM2 = 00-0D-3A-92-03-DC

AzureAuthResName = azurauth

    )



    IP gcoip (

        Address = "10.3.3.100"

        SubNetMask = "255.255.255.0"

MACAddress @AzureVM1 = 00-0D-3A-91-73-A0

MACAddress @AzureVM2 = 00-0D-3A-92-03-DC



    )
```

```
    NIC gconic (
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    )

    gcoip requires azureipres
    gcoip requires gconic
    wac requires gcoip

group VVR (
    SystemList = { AzureVM1= 0, AzureVM2= 1 }
    AutoStartList = { AzureVM1, AzureVM2}
)

    AzureDisk disk_res (
        DiskIds = {
            "/subscriptions/6940a326-abc6-40dd-b717-c1e9bcdf1d63/
         resourceGroups/TestRG/providers/Microsoft.Compute/
         disks/AzureDisk1"
        }
        VMResourceGroup = TestRG
```

```
        AzureAuthResName = azurauth
    )


    AzureDisk disk_res1 (
        DiskIds = {
            "/subscriptions/6940a326-abc6-40dd-b717-c1e9bcdf1d63/
           resourceGroups/TestRG/providers/Microsoft.Compute/
           disks/AzureDisk2"
            }
        VMResourceGroup = TestRG
        AzureAuthResName = azurauth
    )


    AzureDisk disk_res3 (
        DiskIds = {
            "/subscriptions/6940a326-abc6-40dd-b717-c1e9bcdf1d63/
         resourceGroups/TestRG/providers/Microsoft.Compute/
         disks/AzureDisk3"
            }
        VMResourceGroup = TestRG
        AzureAuthResName = azurauth
    )
```

```
    AzureIP azureipres_vvr (
        PrivateIP = "10.3.3.200"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
        AzureAuthResName = azurauth
    )

    AzureIP azureipres_vvr1 (
        PrivateIP = "10.3.3.201"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
        AzureAuthResName = azurauth
    )

    IP ip_vvr (
        Address = "10.3.3.200"
        SubNetMask = "255.255.255.0"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    )
```

```
    NIC nic_vvr (
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
    )


    VMNSDg dgres (
        DiskGroupName = vvrdg
        DGGuid = 6bee34d7-056d-4b01-ae63-397099792471
        )


    VvrRvg rvgres (
        RVG = rvg
        VMDgResName = dgres
        IPResName = ip_vvr
        )


    ip_vvr requires azureipres_vvr
    dgres requires disk_res
    dgres requires disk_res1
    ip_vvr requires nic_vvr
    rvgres requires ip_vvr
    rvgres requires dgres
```

```
group SG_FS (
    SystemList = { AzureVM1= 0, AzureVM2= 1 }
    ClusterList = { vcs_cluster1 = 0,    vcs_cluster2 = 1 }
    Authority = 1
    )

    FileShare SG_FS-FileShare (
        PathName = "\\PerfLogs"
        ShareName = PerfLogs
        LanmanResName = SG_FS-Lanman
        MountResName = SG_FS-MountV
        UserPermissions = { "VCSDOMAIN\\Administrator" = FULL_CONTROL }
        ShareSubdirectories = 1
        )

    AzureIP SG_FS-AzureIP (
        PrivateIP = "10.3.3.300"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
        AzureAuthResName = azurauth
    )
```

```
IP SG_FS-IP (
        Address = "10.3.3.300"
        SubNetMask = "255.255.255.0"
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
        )

    Lanman SG_FS-Lanman (
        VirtualName = AzureVirtualFS
        IPResName = SG_FS-IP
        DNSUpdateRequired = 1
        ADUpdateRequired = 1
        DNSCriticalForOnline = 1
        ADCriticalForOnline = 1
        )

    MountV SG_FS-MountV (
        MountPath = "G:"
        VolumeName = volume1
        VMDGResName = dgres
        )
```

```
    NIC SG_FS-NIC (
MACAddress @AzureVM1 = 00-0D-3A-91-73-A0
MACAddress @AzureVM2 = 00-0D-3A-92-03-DC
        )

    RVGPrimary SG_FS-RVGPrimary (
        RvgResourceName = rvgres
        )

    requires group VVR online local hard
    SG_FS-FileShare requires SG_FS-MountV
    SG_FS-FileShare requires SG_FS-Lanman
    SG_FS-IP requires SG_FS-NIC
    SG_FS-IP requires SG_FS-AzureIP
    SG_FS-Lanman requires SG_FS-IP
    SG_FS-MountV requires SG_FS-RVGPrimary
```

Replication to and across cloud environments

This section includes the following topics:

- Data replication in supported cloud environments
- Supported replication scenarios
- Setting up replication across AWS and Azure environments

Data replication in supported cloud environments

To offer scalable, cost-effective disaster recovery (DR) options for your business, Veritas provides the following technologies in tandem with cloud services:

- Flexible Storage Sharing (FSS) and Volume Replicator (VVR) for Linux environments

Using FSS, a technology that enables seamless cluster-wide sharing of local storage over the network, you can form clusters across zones within a region. Using VVR, you can replicate data across zones and regions for both standalone and clustered systems.

- VVR for Windows environments

You can replicate data across zones and across regions.

> **Note:** In these topics about replication,zoneis used as a common term to refer to an Availability Zone (AZ) in AWS, a VNet or a user-defined site in Azure, or a zone in Google cloud.

> With the help of these technologies, you can:

- Leverage the cloud as a DR site.
- Replicate application data from your on-premises data center to the cloud or within the cloud.
- Avoid the infrastructural costs that are needed to maintain a second physical site.

Data is always available, even if a failure occurs in a node, or a storage device, or a zone.

Supported replication scenarios

InfoScale provides the VVR component to replicate data in any of the following scenarios:

- Replication from on-premises to AWS, Azure, or Google Cloud

In this scenario, you configure replication between an on-premises data center and an AWS, an Azure, or a Google Cloud data center. The on-premises data center is the primary site and the cloud data center is the secondary site.

- Replication within a region

In this scenario, you configure replication between two cloud data centers that are located within a region. The data centers may exist either in the same zone or in different zones.

In case of Windows environments, if a network zone fails, InfoScale replicates data from one zone to the other by using VVR.

- Replication across regions

In this scenario, you configure replication between two cloud data centers that are located in different regions. The primary site is in one region and the secondary site is in another region.

- Replication across clouds

In this scenario, you configure replication between an Azure cloud and an AWS cloud. One of the clouds is the primary site and the other is the DR site.

Setting up replication across AWS and Azure environments

Setting up replication across clouds is similar to setting up replication from an on-premises (primary) data center to a cloud (secondary) data center. Consider one of the clouds as the primary data center and the other as the secondary data center.

The high-level process is as follows:

1. Prepare the deployment setup in each cloud.

See Replication from on-premises to AWS - Windows.

See Replication from on-premises to Azure - Windows.

1. Set up a communication channel between the clouds. For example, you can set up a virtual network gateway or you can configure a VPN-based connection.

2. Install the appropriate InfoScale product on the compute instances of the primary data and the secondary data center.

For details on installation in Windows environments, refer to the *Veritas InfoScale Installation and Upgrade Guide - Windows*.

1. Configure replication using the VVR option between the InfoScale nodes to which your source and target volumes are attached.

For details on configuring replication in Windows environments, refer to the *Veritas Volume Replicator Administrator's Guide* \- Windows.