InfoScale Volume Replication Using Azure Load Balancer

Recommended Architecture and Configuration Guidelines

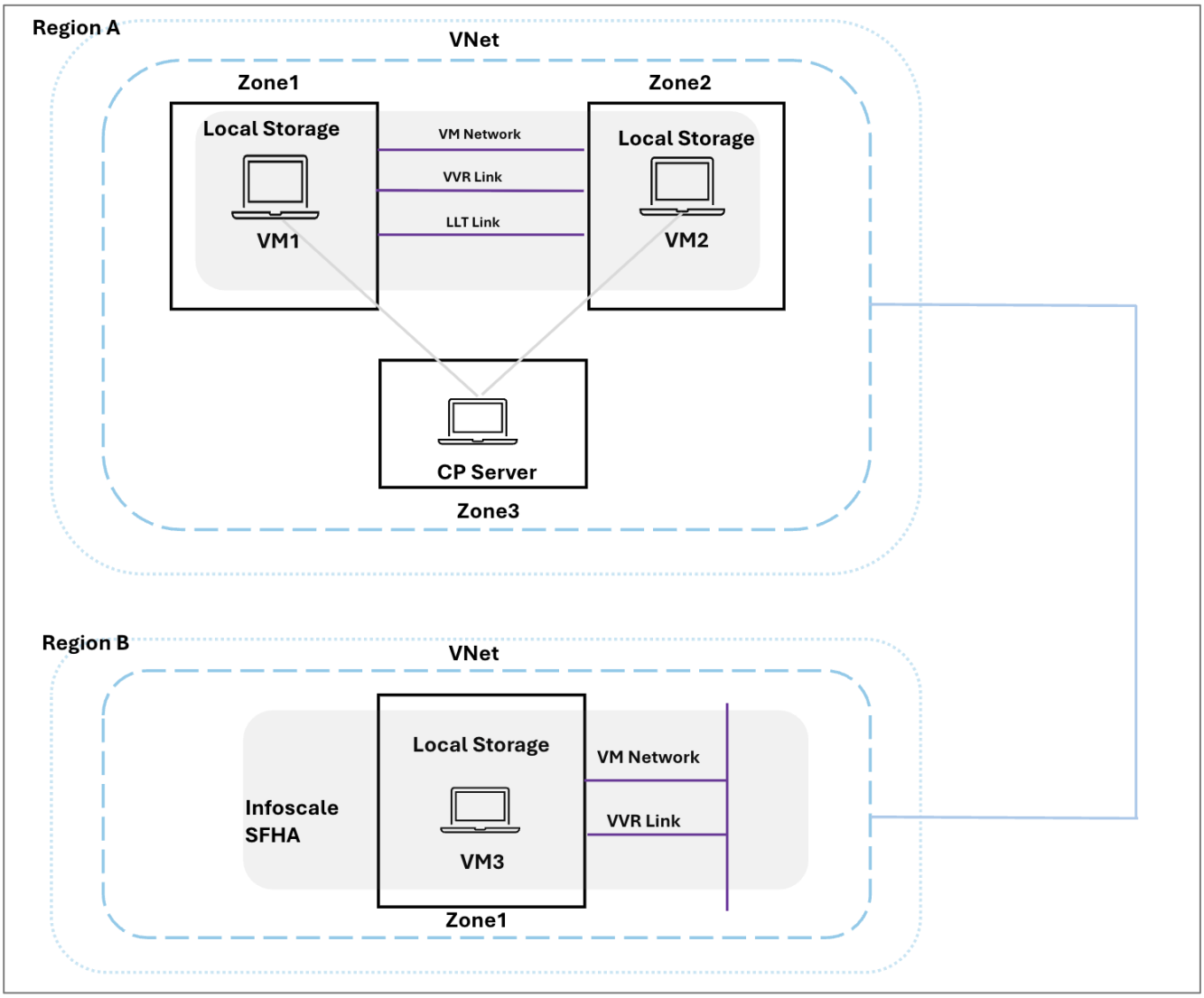
Technote | 2025-10-10

Contents

Use case summary and requirements

InfoScale Enterprise supports replication using its Veritas Volume Replicator (VVR) component in the Azure cloud. This document employs a sample use case to describe how to set up such a replication configuration. The sample use case includes a production site with two hosts (InfoScale cluster nodes) and a disaster recovery (DR) site with one host. Both hosts at the production site are in different Availability Zones (AZs) within the same subnet.

InfoScale cluster nodes at the production site in an Azure cloud instance can be configured with a load balancer service. Azure Load Balancer is a fully managed service that lets you distribute traffic to your backend virtual machines (VMs) and provide high availability (HA) for your applications. With Standard Load Balancer, you can scale your applications and create highly available services. It supports inbound as well as outbound scenarios, provides low latency and high throughput, and scales up to millions of flows for TCP and UDP applications.



Sample InfoScale Replication Configuration in Azure Cloud

In this sample InfoScale configuration, the primary and secondary sites for replication in the Azure cloud are set up as follows:

- InfoScale HA is configured between two VMs (cluster nodes—VM1 and VM2) in different AZs (Zone1 and Zone2) within the same region (Region A—for example, East US).

  A Flexible Storage Sharing (FSS) disk group and a mirror volume are configured across the two cluster nodes to provide data synchronization between the nodes.

- A single-node cluster (VM3) serves as the InfoScale DR site, which is configured in a different region (Region B—for example, West US).

Configuration requirements for this HA-DR use case

VM requirements:

- In the East US region:
  - VM1 configured with an InfoScale-supported Linux version in Zone1
  - VM2 configured with an InfoScale-supported Linux version in Zone2
- In the West US region, VM3 configured with an InfoScale-supported Linux version in Zone1

Network interface card (NIC) requirements:

- HA site (East US region):
  - One NIC (NIC1) on both VM1 and VM2 for the VM public network
  - One NIC (NIC2) for VVR replication
  - Two NICs (NIC3 and NIC4) on both VM1 and VM2 for the LLT network
- DR site (West US region):
  - One NIC (NIC1) on VM3 for the VM public network
  - One NIC (NIC2) on VM3 for VVR replication

IP address requirements:

- HA site (East US region):
  - One public IP address for NIC1 on both VM1 and VM2 for the VM public network
  - One public IP address for NIC1 on both VM1 and VM2 for GCO
  - One public IP address for NIC2 on both VM1 and VM2 for VVR replication
  - One IP address each for NIC3 and NIC4 on both VM1 and VM2 for LLT over UDP
  - One load balancer frontend IP in network of NIC2

- DR site (West US region):
  - One public IP address for NIC1 on VM3 for the VM public network
  - One public IP address for NIC1 on VM3 for the GCO IP
  - One public IP address for NIC2 on VM3 for VVR replication

Load balancer requirements:

- A load balancer configured at the HA site (East US region)
- A floating IP configured as the frontend IP for the load balancer
- A virtual IP assigned to NIC1 of VM1 and NIC1 of VM2, and configured as backend IPs for the load balancer

Port requirements:

Port numbers	Used for
UDP 50000-50008	LLT connectivity between the cluster nodes
UDP 4145	IANA approved heartbeat communication between the primary (HA) site and the secondary (DR) site (All the primary site nodes should be able to connect to all the secondary site nodes and vice versa on this port.)
TCP 4145	IANA-approved port for the TCP listener service
TCP 8199	IANA-approved port for communication between the `vradmind` daemons on the primary and secondary sites
TCP 8989	Communication between the `vxrsyncd` daemons at both the sites for difference-based synchronization
TCP 14155	Global Cluster Option (GCO) Wide Area Network (WAN)

Storage configuration

Disks are attached to each VM as per the needs of the application.

Recommendation: Use separate disks for the application data and for the Storage Replicator Log (SRL).

Host	Data volume (500 GB disks)	SRL volume (128 GB disks)
VM1	VM1_disk1	VM1_disk2
VM2	VM2_disk1	VM2_disk2

Host	Data volume (500 GB disks)	SRL volume (128 GB disks)
VM3	VM3_disk1	VM3_disk2

HA site storage configuration

Create an FSS disk group using the disks from both the cluster nodes. Then, create mirrored volumes for the application data and for SRL.

To configure storage at the HA site

1. Initialize disks on each node using the `vxdisksetup` command.

```
# /etc/vx/bin/vxdisksetup -i <disk_name>
```

2. Create an FSS disk group using the `vxdg` command with the `-o fss` option.

```
# vxdg -s -o fss init dg01 VM1_disk1 VM1_disk2 VM2_disk1 VM2_disk2
```

3. Create data volumes for the application (Sybase—in this use case) data.

```
# vxassist -g dg01 make dvol11 200g alloc=VM1_disk1, VM2_disk1
# vxassist -g dg01 make dvol12 100g alloc=VM1_disk1, VM2_disk1
# vxassist -g dg01 make dvol13 100g alloc=VM1_disk1, VM2_disk1
# vxassist -g dg01 make dvol14 80g alloc=VM1_disk1, VM2_disk1
```

4. Create the SRL volume.

```
# vxassist -g dg01 make srl 150g alloc=VM1_disk2,VM2_disk2
```

5. Create the Veritas File System (VxFS) file systems.

```
# mkfs -t vxfs /dev/vx/rdsk/dg01/dvol11
# mkfs -t vxfs /dev/vx/rdsk/dg01/dvol12
# mkfs -t vxfs /dev/vx/rdsk/dg01/dvol13
# mkfs -t vxfs /dev/vx/rdsk/dg01/dvol14
```

6. Mount the VxFS file systems.

```
# mount -t vxfs -o cluster /dev/vx/dsk/dg01/dvol11 /appdr/dvol11
# mount -t vxfs -o cluster /dev/vx/dsk/dg01/dvol12 /appdr/dvol12
# mount -t vxfs -o cluster /dev/vx/dsk/dg01/dvol13 /appdr/dvol13
# mount -t vxfs -o cluster /dev/vx/dsk/dg01/dvol14 /appdr/dvol14
```

DR site storage configuration

Create a private disk group using the disks from the node. Then, create volumes for the application data and for SRL.

To configure storage at a DR site

1. Initialize disks on the node using the `vxdisksetup` command.

```
# /etc/vx/bin/vxdisksetup -i <disk_name>
```

2. Create a private disk group using the `vxdg` command.

```
# vxdg init dg01 VM3_disk1 VM3_disk2
```

3. Create data volumes for the Sybase binary, databases, and logs.

```
# vxassist -g dg01 make dvol11 200g alloc=VM3_disk1
# vxassist -g dg01 make dvol12 100g alloc=VM3_disk1
# vxassist -g dg01 make dvol13 100g alloc=VM3_disk1
# vxassist -g dg01 make dvol14 80g alloc=VM3_disk1
```

4. Create the SRL volume.

```
# vxassist -g dg01 make srl 150g alloc=VM3_disk2
```

Load balancer configuration

This section describes how to configure the Azure Load Balancer for the GCO and VVR replication IPs.

A load balancer uses a frontend IP and one or more backend IPs. A frontend IP moves between the nodes in the cluster, and backend IPs are assigned to physical interfaces. The load balancer routes network traffic from the frontend IP to the appropriate backend IPs as necessary.

For an InfoScale VVR configuration, the GCO backend pool is configured with the GCO IP and the VVR backend pool is configured with the IPs used for VVR replication.

The following IP addresses are used in the production cluster:

Production cluster resource	Purpose	IP address	Assigned to host
GCO	Frontend IP	10.100.13.80	
	Backend IPs	10.100.13.90	is-ha-poc1
		10.100.13.91	is-ha-poc2
RVG1	Frontend IP	10.100.13.77	
	Backend IPs	10.100.13.78	is-ha-poc1
		10.100.13.79	is-ha-poc2
RVG2	Frontend IP	10.100.15.137	

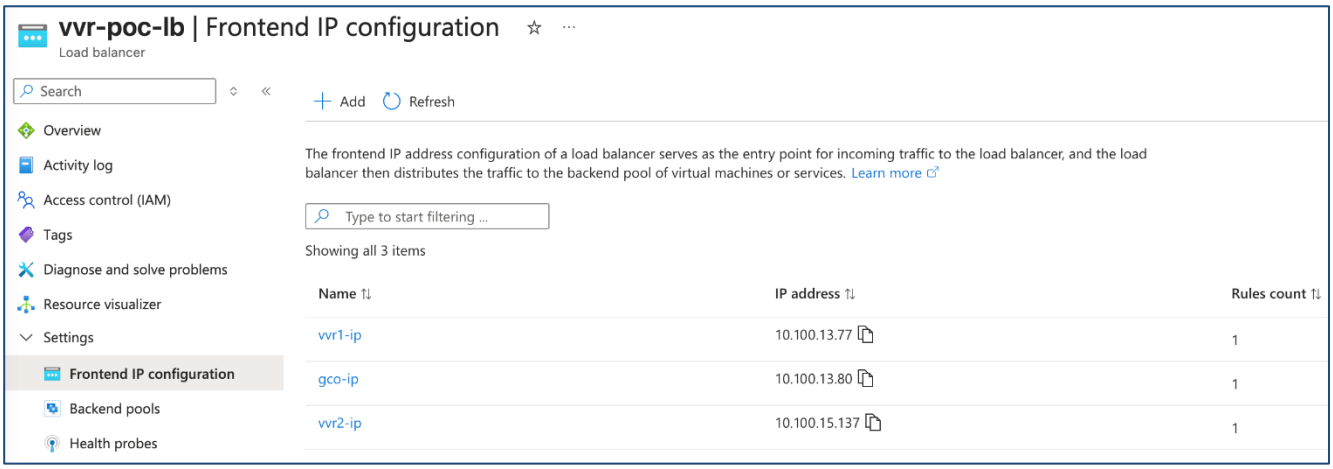
Production cluster resource	Purpose	IP address	Assigned to host
	Backend IPs	10.100.15.134	is-ha-poc1
		10.100.15.135	is-ha-poc2

The following IP addresses are used in the DR cluster:

DR cluster resource	Purpose	IP address	Assigned to host
GCO	No load balancer	10.100.16.131	is-ha-poc3
RVG1		10.100.16.132	is-ha-poc3
RVG2		10.100.16.136	is-ha-poc3

Frontend IP configuration

Create the load balancer instance and configure the frontend IP and backend IP to be used for GCO and for VVR replication.



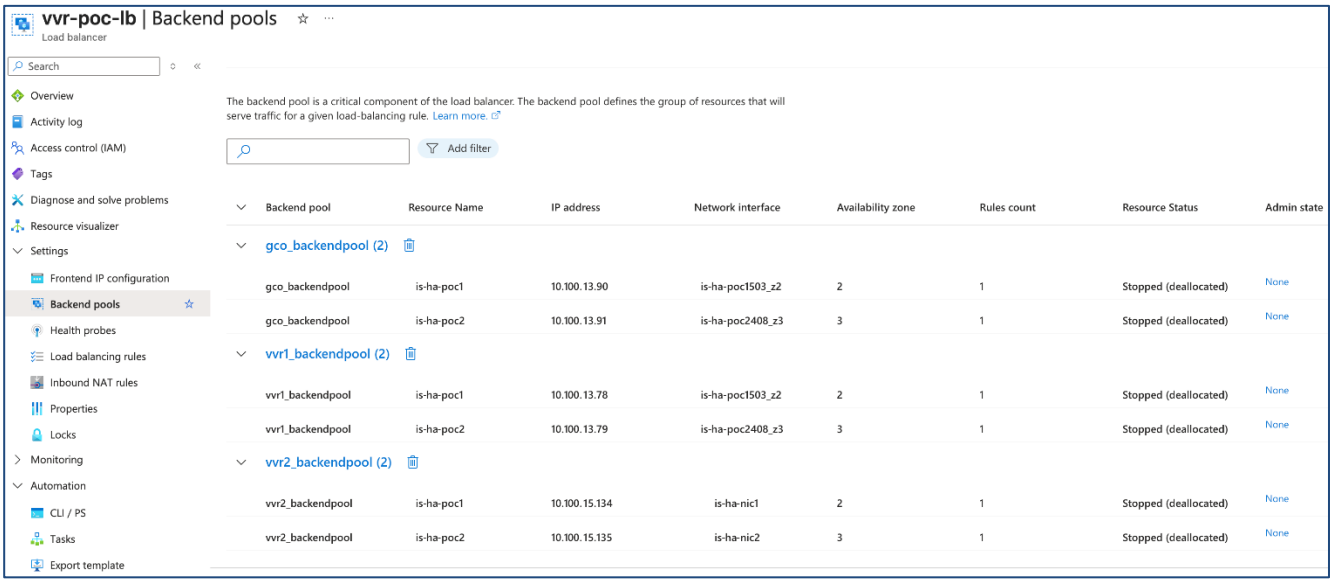
Frontend IP configuration for the load balancer

In this use case, `vvr-poc-lb` is the load balancer with frontend IPs configured for GCO (`gco_ip`) and for the two replicated volume groups (RVGs) (`vvr1-ip` and `vvr2-ip`).

Backend IP configuration

Backend IPs used in backend pools are IP addresses assigned to physical interfaces or alias interfaces, which are used for VVR replication and GCO.

Create backend pools for GCO and for the two RVGs.



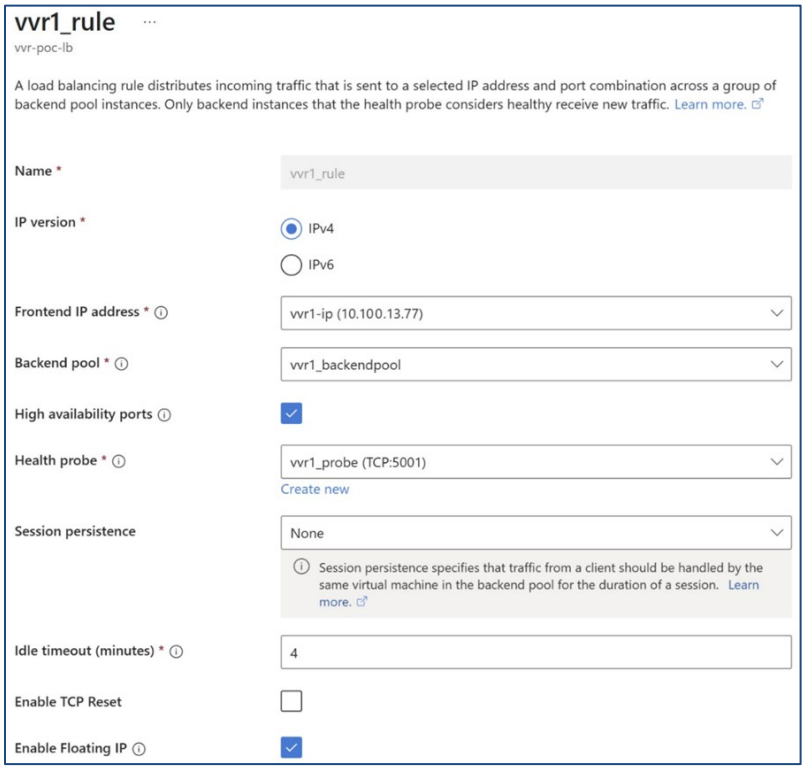
Backend IP configuration for the load balancer

In this use case, the backend pools are `gco_backendpool`, `vvr1_backendpool`, and `vvr2_backendpool`.

Load balancing rule configuration

Configure load balancing rules for RVG replication.

Note: While configuring the rules, make sure to enable floating IP support.



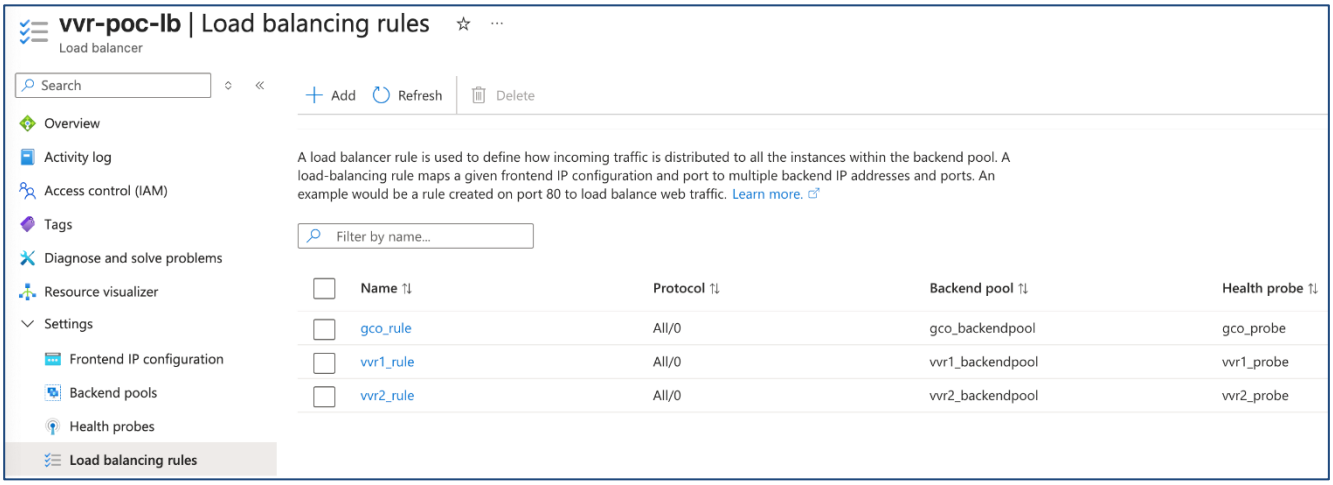
Load balancing rule configuration

The port used for **Health probe** in this configuration must also be updated in VCS configuration file mentioned in the following sections:

- VCS configuration for GCO at HA site on page 11
- RVGLogowner resource on page 16

In this use case, `vvr1_rule` is the load balancing rule for replication from one of the two RVGs.

Similarly, create rules for GCO and for replication from the other RVG:

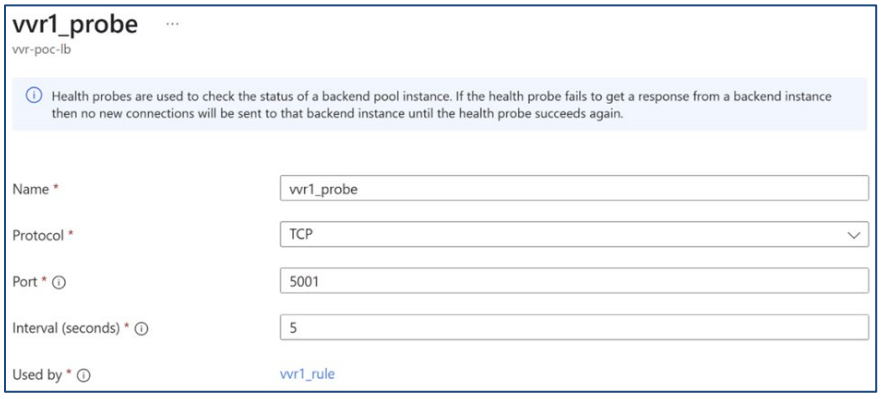


Load balancing rules

Health probe configuration

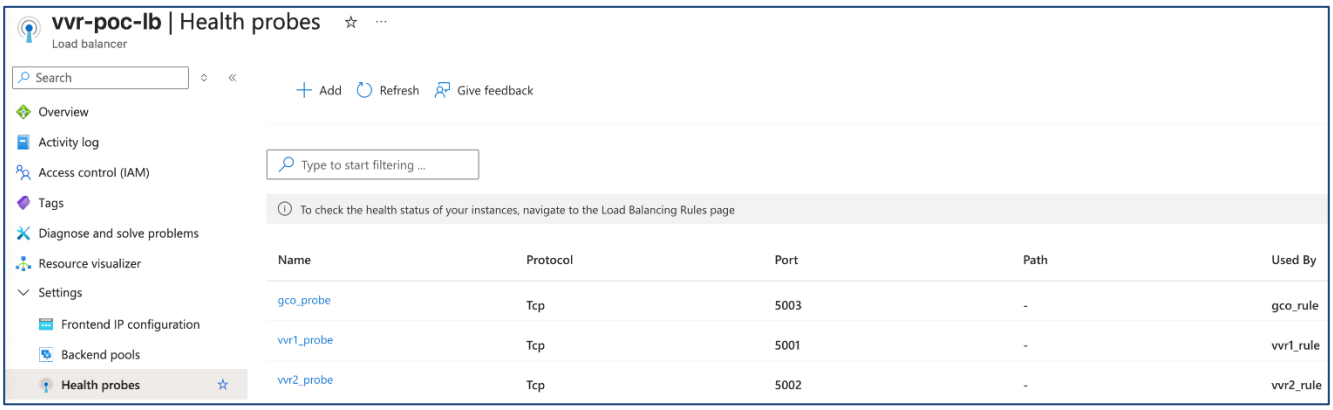
A health probe is a test that the load balancer runs continuously to check whether each backend resource—in this use case, each of the IP addresses configured for GCO and replication—is healthy and able to serve traffic.

Configure a health probe for one of the two RVGs with the appropriate protocol and the desired network port.



Health probe configuration

Similarly, configure health probes for GCO and for replication from the other RVG.



Health probes

VCS configuration for GCO

Considerations for the GCO configuration:

- HA is configured across regions. Therefore, when a disaster occurs, a global cluster configuration is required to provide the ability to fail over an application between the geographically distributed clusters.
- In the East US region, the cluster is configured using two VMs located in different AZs.
- In the West US region, single-node cluster is configured to act as the DR site.
- A GCO IP is already configured for the Azure Load balancer as described in the "Load balancer configuration" section on page 7.

VCS configuration for GCO at HA site

In the East US region, there are two nodes in the production cluster.

For the GCO configuration, add the backend IPs for the load balancer in the VCS configuration file (`/etc/VRTSvcs/conf/config/main.cf`).

```
cluster clus_is-ha-poc1 (
    SecInfo256 = "Some string value"
    UserNames = { admin = "Encrypted password" }
    ClusterAddress = "10.100.13.80"
    ProtocolNumber = 11000
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    IV256List = { "key" = "value" }
)
```

```
remotecluster clus_is-ha-poc3 (
    ClusterAddress = "10.100.16.131"
)
```

Configure the VCS service group for the WAC resource.

```
group ClusterService (
    SystemList = { is-ha-poc1 = 0, is-ha-poc2 = 1 }
    AutoStartList = { is-ha-poc1, is-ha-poc2 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
)
Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart -any"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac -any" }
    RestartLimit = 3
)
IP gco_backend_ip (
    Device = eth0
    Address @is-ha-poc1 = "10.100.13.90"
    Address @is-ha-poc2 = "10.100.13.91"
    NetMask @is-ha-poc1 = "255.255.255.0"
    NetMask @is-ha-poc2 = "255.255.255.0"
)
IP gco_ip (
    Device = eth0
    Address = "10.100.13.80"
    NetMask = "255.255.255.0"
)
NIC gco_nic (
    Device = eth0
)
Process gco_5003 (
    PathName = "/opt/script/port_probe.sh"
    Arguments = "gco 5003"
    PidFile = "/var/VRTSvcs/lock/volatile/gco_pid"
)
```

```
gco_5003 requires gco_backend_ip
gco_5003 requires gco_ip
gco_backend_ip requires gco_nic
gco_ip requires gco_nic
wac requires gco_ip
```

VCS configuration for GCO at DR site

In the West US region, a single-node DR cluster is configured.

For GCO configuration, add the following details in the VCS configuration file (`/etc/VRTSvcs/conf/config/main.cf`).

```
cluster clus_is-ha-poc3 (
    SecInfo256 = "Some string value"
    UserNames = { admin = "Encrypted password" }
    ClusterAddress = "10.100.14.132"
    ProtocolNumber = 11000
    Administrators = { admin }
    IV256List = { "key" = "value" }
)
remotecluster clus_is-ha-poc1 (
    ClusterAddress = "10.100.13.80"
)
heartbeat Icmp (
    ClusterList = { clus_is-ha-poc1 }
    Arguments @clus_is-ha-poc1 = { "10.100.13.80" }
)
system is-ha-poc3 (
)
```

Configure the VCS service group for the WAC resource.

```
group ClusterService (
    SystemList = { is-ha-poc3 = 0 }
    AutoStartList = { is-ha-poc3 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
)
Application wac (
```

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

Replication configuration

To provide HA for applications across different regions, data from the production site (East US region) must be replicated to the DR site (West US region). If a disaster occurs at the production site, the data is available at the DR site, and the application can be started at the DR site. InfoScale VVR can be configured for this use case.

Considerations for this configuration:

- The data volumes and the SRL volume must have the same name and the same size at both the sites.

  For details, refer to "Storage configuration" on page 5.

- Configure the Azure Load Balancer according to the network subnets configuration of the hosts.

  For details, refer to "Load balancer configuration" on page 7.

- **Recommendation**: Use a different RVG for each application.

To configure VVR

1. Authenticate the primary site disk group at the secondary site.

   Get disk group ID from the primary site.

   ```
   # vxdg list dg01 | grep dgid:
   dgid:    1713449882.24.vvr-prod1
   ```

   At the secondary site, append the disk group ID of the primary site to the `/etc/vx/vras/.rdg` file.

   At the primary site, append the disk group ID of the secondary site to the `/etc/vx/vras/.rdg` file.

   ```
   # cat /etc/vx/vras/.rdg | grep vvr-prod1
   # 1713449882.24.vvr-prod1
   ```

2. Configure the RVG primary.

```
# vradmin -g dg01 createpri rvg01 dvol11, dvol12, dvol13, dvol14 srl
```

3. Configure the RVG secondary using the frontend IP of the load balancer.

   Note: If you have assigned virtual IPs—not the static IP allocated by Azure DHCP—to the network interfaces of a host for configuring VVR replication, make sure to enable the following tunable on each node in the production cluster and the DR cluster. Otherwise, the replication will not work.

```
# vxtune vol_vvr_port_bind any_ip
# vradmin -g dg01 addsec rvg01 10.100.13.77 10.100.16.132
```

   Here, 10.100.13.77 is the frontend IP address of the load balancer.

   Note: For the load balancer frontend IP to work, Azure adds a special IP routing rule on the eth0 interface by default. In case you want to configure the replication or the LLT network on an interface other than eth0, Red Hat recommends that you add the IP routing rule manually on the interface being used.

   For details, refer to "Routing rules configuration" on page 20.

4. Start replication with full data volume sync.

```
# vradmin -g dg01 -a startrep rvg01 10.100.16.132
```

VCS resource configuration for VVR and application availability

Configure the following VCS resources to provide HA for the VVR replication and the application:

- RVGShared and RVG
- RVGLogowner
- RVGSharedPri, RVGPrimary, and application resources (Sybase—in this use case)

This section describes the changes required in the cluster configuration file (`/etc/VRTSvcs/conf/config/main.cf`) to configure resources.

RVGShared and RVG resources

At the HA site (East US region), create a parallel group called RVGgroup and add the RVGShared and CVMVolDg resources to it.

```
group RVGgroup_1 (
    SystemList = { is-ha-poc1 = 0, is-ha-poc2 = 1 }
    Parallel = 1
```

```
    AutoStartList = { is-ha-poc1, is-ha-poc2 }
)
CVMVolDg voldg_res_1 (
    CVMDiskGroup = dg01
    CVMActivation = sw
)
RVGShared rvg_res_1 (
    RVG = rvg01
    DiskGroup = dg01
)
requires group cvm online local firm
rvg_res_1 requires voldg_res_1
```

At the DR site (West US region), create a group called RVGgroup and add the RVG and DiskGroup resources to it.

```
group RVGgroup_1 (
    SystemList = { is-ha-poc3 = 0 }
    AutoStartList = { is-ha-poc3 }
)
DiskGroup dg_res_1 (
    DiskGroup = dg01
)
RVG rvg_res_1 (
    RVG = rvg01
    DiskGroup = dg01
)
rvg_res_1 requires dg_res_1
```

RVGLogowner resource

At the HA site (East US region), create a failover group called RLOGowner and add the RVGLogowner resource to it.

For the load balancer configuration, mention the backend IP of the load balancer and the health probe port in the VCS `main.cf` file as follows:

```
group RLOGowner_1 (
    SystemList = { is-ha-poc1 = 0, is-ha-poc2 = 1 }
    AutoStartList = { is-ha-poc1, is-ha-poc2 }
```

```
    OnlineRetryLimit = 2
)
IP backend-ip_1 (
    Device = eth1
    Address @is-ha-poc1 = "10.100.13.78"
    Address @is-ha-poc2 = "10.100.13.79"
    NetMask @is-ha-poc1 = "255.255.255.0"
    NetMask @is-ha-poc2 = "255.255.255.0"
)
IP lb_ip_1 (
    Device = eth1
    Address = "10.100.13.77"
    NetMask = "255.255.255.0"
)
Process vvr_5001 (
    PathName = "/opt/script/port_probe.sh"
    Arguments = "port1 5001"
    PidFile = "/var/VRTSvcs/lock/volatile/port1_pid"
)
Proxy lb_nic_proxy_1 (
    TargetResName = gco_nic
)
RVGLogowner logowner_1 (
    RVG = rvg01
    DiskGroup = dg01
)
requires group RVGgroup_1 online local firm
backend-ip_1 requires lb_nic_proxy_1
lb_ip_1 requires lb_nic_proxy_1
logowner_1 requires lb_ip_1
vvr_5001 requires lb_ip_1
```

Sample contents of the health probe file:

```
# cat /opt/script/port_probe.sh
#!/bin/bash
res=$1
port=$2
```

```
pid_file="/var/VRTSvcs/lock/volatile/${res}_pid"
NC="/usr/bin/nc"

old_pid=`ps -eo pid,args | grep -w "nc" | grep -w $port$ | awk '{ print $1
}`
if [ -n $old_pid ]; then
    kill -9 $old_pid
fi

$NC -l -k $port &
pid=$!
echo $pid > $pid_file

exit 0
```

RVGSharedPri, RVGPrimary, and application resources

At the HA site (East US region), configure the application service group as a global group. Add the application, RVGSharedPri, and CFSMount resources to the service group.

```
group app_grp_1 (
    SystemList = { is-ha-poc1 = 0, is-ha-poc2 = 1 }
    ClusterList = { clus_is-ha-poc1 = 0, clus_is-ha-poc3 = 1 }
    Authority = 1
    AutoStartList = { is-ha-poc1, is-ha-poc2 }
    )
CFSMount mnt_dvol11 (
    MountPoint = "/sybasedir1/dvol11"
    BlockDevice = "/dev/vx/dsk/dg01/dvol11"
    )
CFSMount mnt_dvol12 (
    MountPoint = "/sybasedir1/dvol12"
    BlockDevice = "/dev/vx/dsk/dg01/dvol12"
    )
CFSMount mnt_dvol13 (
    MountPoint = "/sybasedir1/dvol13"
    BlockDevice = "/dev/vx/dsk/dg01/dvol13"
    )
```

```
CFSMount mnt_dvol14 (
    MountPoint = "/sybasedir1/dvol14"
    BlockDevice = "/dev/vx/dsk/dg01/dvol14"
    )
RVGSharedPri sharedpri_res_1 (
    RvgResourceName = rvg_res_1
    OnlineRetryLimit = 0
    )
requires group RVGgroup_1 online local firm
mnt_dvol11 requires sharedpri_res_1
mnt_dvol12 requires sharedpri_res_1
mnt_dvol13 requires sharedpri_res_1
mnt_dvol14 requires sharedpri_res_1
```

At the DR site (West US region), configure the application service group as a global group. Add the application, RVGPrimary, and Mount resources to the service group.

```
group app_grp_1 (
    SystemList = { is-ha-poc3 = 0 }
    ClusterList = { clus_is-ha-poc1 = 0, clus_is-ha-poc3 = 1 }
    )
Mount mnt_dvol11 (
    MountPoint = "/sybasedir1/dvol11"
    BlockDevice = "/dev/vx/dsk/dg01/dvol11"
    FSType = vxfs
    FsckOpt = "-y"
    )
Mount mnt_dvol12 (
    MountPoint = "/sybasedir1/dvol12"
    BlockDevice = "/dev/vx/dsk/dg01/dvol12"
    FSType = vxfs
    FsckOpt = "-y"
    )
Mount mnt_dvol13 (
    MountPoint = "/sybasedir1/dvol13"
    BlockDevice = "/dev/vx/dsk/dg01/dvol13"
    FSType = vxfs
    FsckOpt = "-y"
```

```
    )
Mount mnt_dvol14 (
    MountPoint = "/sybasedir1/dvol14"
    BlockDevice = "/dev/vx/dsk/dg01/dvol14"
    FSType = vxfs
    FsckOpt = "-y"
    )
RVGPrimary pri_res_1 (
    RvgResourceName = rvg_res_1
    OnlineRetryLimit = 0
    )
requires group RVGgroup_1 online local hard
mnt_dvol11 requires pri_res_1
mnt_dvol12 requires pri_res_1
mnt_dvol13 requires pri_res_1
mnt_dvol14 requires pri_res_1
```

Routing rules configuration

If VVR replication is configured with the load balancer using IP addresses assigned to a network interface other than eth0, you must configure policy-based routing for the VVR replication as well as LLT traffic. Policy-based routing (PBR) controls network traffic flow in systems with multiple network interfaces. Traditional routing methods may lead to inconsistent or asymmetric paths. PBR allows administrators to define custom routing policies that ensure traffic from each source follows the correct network path.

Note: The following configuration is provided only for reference. For further details, contact your OS vendor.

LLT configuration

At the production site, eth3 (wired connection 3) and eth4 (wired connection 4) are used for both the VMs for LLT network with following IP address:

- VM1 (production site node 1):
  - LLT1 IP: 10.100.14.23/25
  - LLT2 IP: 10.100.14.24/25
- VM2 (production site node 2):
  - LLT1 IP: 10.100.14.25/25
  - LLT2 IP: 10.100.14.26/25

Add routing rules such that the LLT traffic uses eth3 and eth4 for LLT traffic. In this use case, a routing table with priority 8 has been used for eth3 and a routing table with priority 9 has been used for eth4.

The commands used to add routes and routing rules on VM1 are as follows:

- Create a table with priority 8 with routing to subnet 10.100.14.0/25 using eth3.

```
# nmcli con mod "Wired connection 3" ipv4.route-table 8
```

- Use routing table 8 for packets originating from 10.100.14.23/25, which is the IP for LLT.

```
# nmcli con mod "Wired connection 3" +ipv4.routing-rules "priority 100 from
10.100.14.23/25 table 8"
```

- Use routing table 8 for packets originating from 10.100.14.23/25, which is the subnet for LLT.

```
# nmcli con mod "Wired connection 3" +ipv4.routing-rules "priority 101 to
10.100.14.0/25 table 8"
```

- Apply the configuration changes to the eth3 interface.

```
# nmcli device reapply eth3
```

- Run similar commands for eth4 on VM1.

```
# nmcli con mod "Wired connection 4" ipv4.route-table 9
# nmcli con mod "Wired connection 4" +ipv4.routing-rules "priority 100 from
10.100.14.24/25 table 9"
# nmcli con mod "Wired connection 4" +ipv4.routing-rules "priority 101 to
10.100.14.0/25 table 9"
# nmcli device reapply eth4
```

- Add an entry in `rt_table` file to specify the use of these routing tables.

```
# cat /etc/iproute2/rt_tables
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
```

```
8 table_llt_1
9 table_llt_2
```

Similarly, add the routing table and rules for IPs for LLT on VM2. Make sure to use the same table priorities as those on VM1 for the eth3 and eth4 interfaces respectively.

VVR replication configuration

In this sample use case, VVR replication is configured on the eth2 interface on both the production site nodes and on the DR site node. The following description is for RVG2. You need to configure rules is a similar manner for another RVGs that are not configured on eth0 interface.

- VM1 (production site node 1):
  - eth2 static IP: 10.100.15.140/25
  - eth2 IP for VVR: 10.100.15.134/25
  - eth2 gateway: 10.100.15.129
- VM2 (production site node 2):
  - eth2 static IP: 10.100.15.141/25
  - eth2 IP for VVR: 10.100.15.135/25
  - eth2 gateway: 10.100.15.129
- Load balancer:
  - Frontend IP: 10.100.15.137
  - Backend IPs: 10.100.15.134/25, 10.100.15.135/25 (IPs for VVR on the primary site nodes)
- VM3 (DR site node):
  - eth2 IP for VVR: 10.100.16.136/25
  - eth2 gateway: 10.100.16.129

On the nodes at the production site, add routing information on VM1 such that:

- Any traffic from 10.100.15.134 (VVR IP on VM1) to 10.100.16.128/25 (secondary site subnet) or to 10.100.15.135 (VVR IP on VM2), will be routed using the table 11 routing table.
- In that table, traffic destined for 10.100.16.128/25 (secondary site subnet), 10.100.15.135/25 (VVR IP on VM2) or 168.63.129.16 (health probe for Azure Load Balancer) will be forwarded via 10.100.15.129 (primary site gateway) over eth2.
- The priority number of the table is 11; you may select any other number based on your system requirements.

Add routes and routing rules on the VM1 using `nmcli` are as follows:

- Route traffic to prod2 VVR IP via 10.100.15.129

```
# nmcli con mod  "Wired connection 2" +ipv4.routes "10.100.15.135/25
10.100.15.129 table=11"
```

- Routes traffic to secondary side subnet 10.100.16.128/25 via gateway 10.100.15.129

```
# nmcli con mod  "Wired connection 2" +ipv4.routes "10.100.16.128/25
10.100.15.129 table=11"
```

- Routes traffic to host 168.63.129.16 via gateway 10.100.15.129

```
# nmcli con mod  "Wired connection 2" +ipv4.routes "168.63.129.16/32
10.100.15.129 table=11"
```

- Use routing table 11 for packets originating from prod1 VVR IP

```
# nmcli con mod  "Wired connection 2" +ipv4.routing-rules "priority 110
from 10.100.15.134/25 table 11"
```

- Use routing table 11 for all traffic destined to destination subnet from any source IP

```
# nmcli con mod  "Wired connection 2" +ipv4.routing-rules "priority 119 to
10.100.16.128/25 table 11"
```

- Use routing table 11 for packets going to prod VVR IP from any source IP

```
# nmcli con mod  "Wired connection 2" +ipv4.routing-rules "priority 120 to
10.100.15.135/25 table 11"
```

- Reapply configurations on eth2 without disconnecting

```
# nmcli device reapply eth2
```

Add an entry in the `rt_tables` file to specify the custom routing table.

```
# cat /etc/iproute2/rt_tables
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
```

```
#
#1      inr.ruhep
8 table_llt_1
9 table_llt_2
11 table_vvr
```

Similarly, run the commands on VM2 for routing traffic from 10.100.15.134 (VVR IP on VM2) to 10.100.15.128/25 (secondary site subnet) and 10.100.15.134 (VVR IP on VM11). Then, add routing rules so that the traffic uses table 11.

At the secondary (DR) site, use the following commands for bi-directional communication for VVR traffic:

```
# nmcli connection modify "Wired connection 2" +ipv4.routes
"10.100.16.128/25 10.100.16.128"
# nmcli connection modify "Wired connection 2" +ipv4.routes
"10.100.15.128/25 10.100.16.128"
# nmcli device reapply eth2
```

Thus, the secondary site gateway is added as the default gateway for the primary and secondary subnet traffic using eth2.

Note: If GCO is configured on an interface other than eth0, make sure to configure similar rules for the GCO traffic as well.

References

- InfoScale Installation Guide

  https://www.veritas.com/support/en_US/doc/109508799-159001289-1

- SFHA Configuration and Upgrade Guide

  https://www.veritas.com/support/en_US/doc/79757062-159001622-1

- SFCFSHA Configuration and Upgrade Guide

  https://www.veritas.com/support/en_US/doc/79735435-159001592-1

- InfoScale Solutions in Cloud Environments

  https://www.veritas.com/support/en_US/doc/130803809-158949452-1

This page is intentionally left blank.