VERITAS

Veritas InfoScale Enterprise on AWS Cloud

An example of a MySQL deployment.

White Paper | March 2022

Contents

Revision History

Revision	Date	Author
Rev 1.0 1	June 2021	Initial version

Introduction

Executive Summary

Veritas InfoScale™ Enterprise is a software-defined high availability and storage management solution for enterprise IT applications. InfoScale is platform and infrastructure agnostic and can manage applications on nearly any platform that offers storage provisioning and management. As more companies consider cloud deployment architecture, the deployment of InfoScale Enterprise on AWS cloud becomes integral for organizations that desire business continuity for their applications running in the cloud.

InfoScale provides several advanced features for AWS, including the ability to replicate application data across AWS availability zones or regions for ease of migration and disaster recovery (DR). InfoScale offers application and database agents for several enterprise IT applications, including SAP, Oracle, Microsoft SQL, and custom applications. InfoScale can fully automate application high availability (HA) and DR for applications running on AWS by managing application failover and failback operations as well as replication between sites.

This document provides a walkthrough of how to deploy MySQL in an active/passive scenario using InfoScale Enterprise on AWS cloud.

Scope

The document provides step-by-step instructions on how to deploy InfoScale Enterprise on AWS and how to deploy a MySQL database within this environment in a highly available configuration managed by InfoScale. This is a sample deployment and you should refer to the Veritas product documentation and AWS for definitive and full installation, administration, and configuration details. There will be references to these documents in this guide.

Target Audience

This document is for customers, partners, and Veritas field personnel interested in understanding how to quickly deploy InfoScale Enterprise on AWS public cloud.

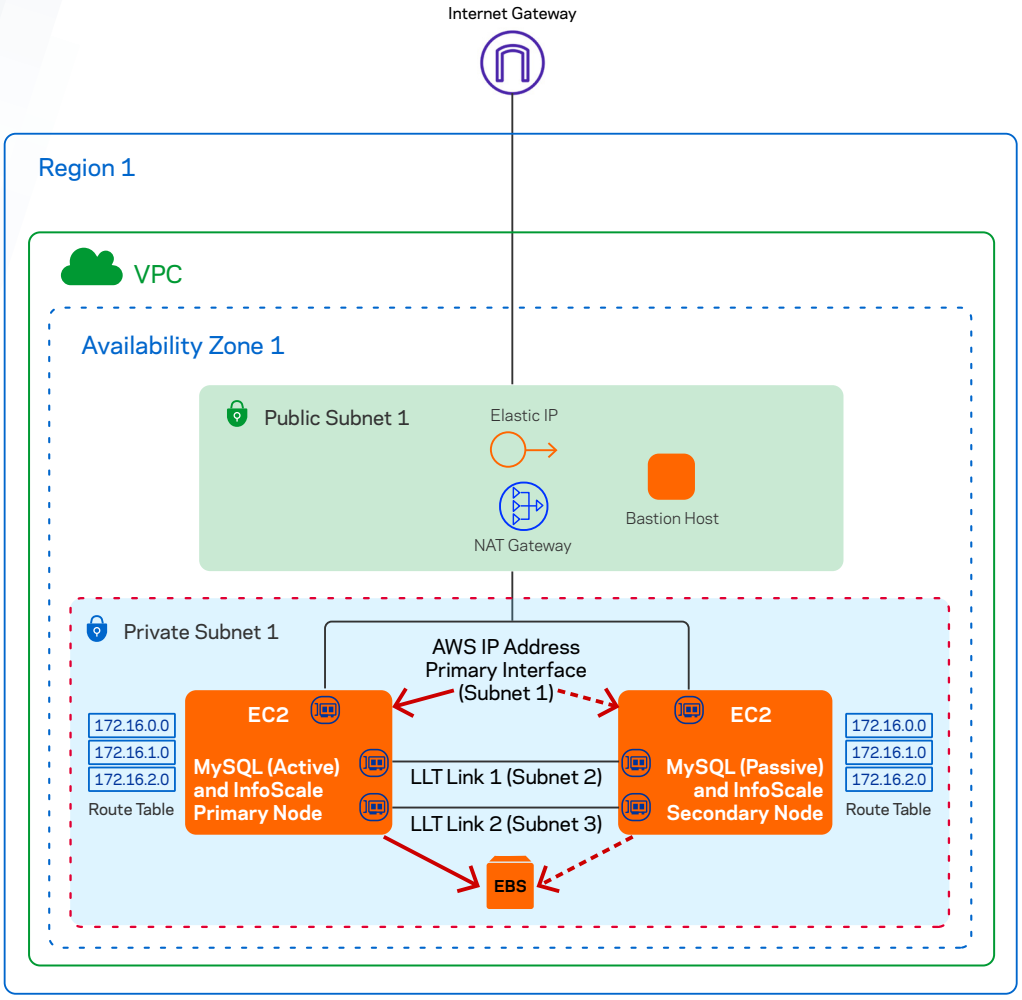
Topology Overview

Figure 1 illustrates a topology deployment example of InfoScale Enterprise on AWS. A single instance of MySQL is made highly available in a two-node cluster with InfoScale Storage Foundation High Availability. MySQL is deployed in active/passive mode as follows:

1. There is a single VPC with two private and one public subnet on AWS. For host name resolution the /etc/hosts file is used, so Amazon Route 53 is not needed. A Network Address Translation (NAT) gateway in a public subnet is created to allow access to the Internet and a bastion host for remote management of the EC2 instances within the private subnets. Appropriate routes are created between the public and private subnets for Internet access, cluster inter-communication over the Low Latency Transport (LLT) protocol, and application-specific traffic. There are three instances deployed:
    1) Bastion Host—for remote management of the EC2 instances used to form an InfoScale cluster
    2) Primary Node—hosting InfoScale and MySQL
    3) Secondary Node—hosting InfoScale and MySQL

For the primary and secondary node, an Elastic IP that has routed to the Internet gateway is associated with the primary interface on each of the EC2 instances for Internet access. There are two additional network interfaces added to the EC2 instances for cluster-communication over the Low Latency Transport (LLT) protocol. Appropriate routes are created between the public and private subnets for Internet access, cluster inter-communication, and application-specific traffic.

2. A highly available single instance of MySQL is set up in EC2 instance 1 and EC2 instance 2 with InfoScale.



3. The database and data files on an EBS volume are online in EC2 instance 1, which will be considered the primary active instance, and EC2 instance 2, which will be the secondary passive instance. In case of failure on the primary instance, MySQL will failover to the secondary instance and the EBS volume containing the data files are detached from the primary instance and attached to the secondary instance. The InfoScale virtual IP will also be re-assigned to the network interface on the secondary EC2 instance.

4. InfoScale EBSVol, AWSIP, and MySQL Agents are installed on each node and used to manage the active/passive MySQL cluster within the AWS cloud.

Figure 1. A high-level view of deployment on AWS.

Software and Compute Resources

Table 1 shows the software and compute resources used in this guide.

Table 1. Resource Summary

Component	Resource
AWS Cloud	VPC with public and private subnet, NAT, and Elastic IP
	1 x EC2 Instance of type t2.medium with Windows 2019 Base (AWS Marketplace)
	2 x EC2 Instance of type c4.2xlarge with RHEL 7.7 AMI (AWS Marketplace)
Veritas	InfoScale Enterprise 7.4.3
	InfoScale Operations Manager 7.4.2.300
	InfoScale MySQL Agents
Utilities	Chrome
	MobaXterm
	WinSCP

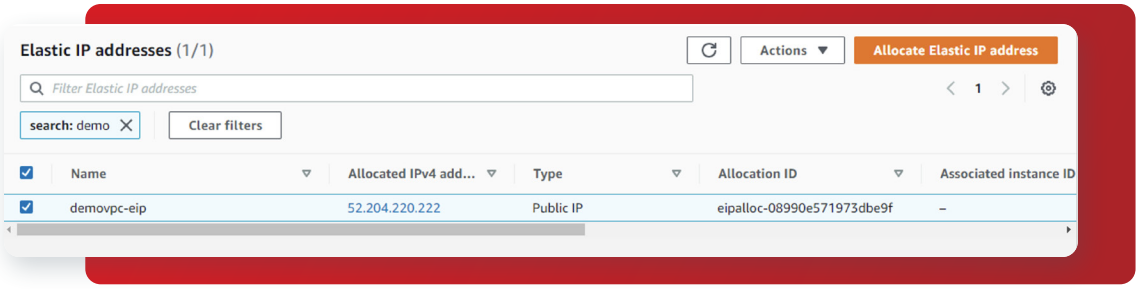
Step 1: Configuration of Amazon VPC

The Amazon VPC defines the virtual network where the AWS resources such as EC2 instances or bastion hosts are launched. In this example, a VPC with public and private subnets is defined. The private subnet can access the Internet via the public subnet using Network Address Translation (NAT).

1. Log on to the AWS management console using your credentials.

2. From the AWS management console, select the **VPC service**.

3. On left pane, select **Elastic IP** and click the **Allocate Elastic IP address** button to assign an Elastic IP address needed for NAT. An Elastic IP is allocated as shown below.



4. On the left pane, select **VPC Dashboard** and click on **Launch VPC Wizard** to create the VPC.

5. Select **VPC with Public and Private Subnets**. Fill in the following fields for the public subnet where the bastion host will be connected and a private subnet for the application-specific traffic.

   a. **IPV4 CIDR block**: 10.34.0.0/16

   b. **VPC name**: demo-vpc

   c. **Public subnet IPv4 CIDR**: 10.34.0.0/24

   d. **Availability Zone**: us-east-1b

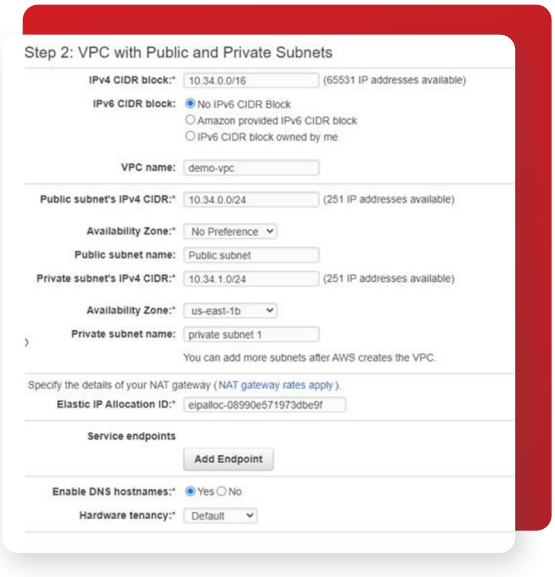
   e. **Subnet name**: Public subnet

   f. **Private subnet IPv4 CIDR**: 10.34.1.0/24

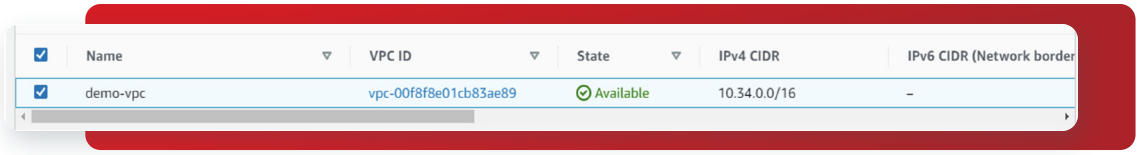
   g. **Availability Zone**:us-east-1b

   h. **Subnet name**: private subnet 1

   i. Select the **Elastic IP** created in previous step 3.



Leave the rest of the fields as defaults and **Click Create**. This action creates a VPC with an IPv4 CIDR block of 10.34.0.0/16. Public subnets and private subnets are created within this CIDR block.



6. On the left pane, select **Subnets** and click on the **Create Subnet** button to create the two additional subnets needed for the InfoScale LLT inter-cluster communication (LLT1 and LLT2). For VPC-ID, select the demo-vpc created in step 5. Under **Subnet** settings, specify the following fields:

a. **Subnet name**: private subnet 2

b. **Availability Zone (same zone as in VPC)**: us-east-1b

c. **IPv4 CIDR block (should be in different subnet but same CIDR block as in VPC)**: 10.34.2.0/24

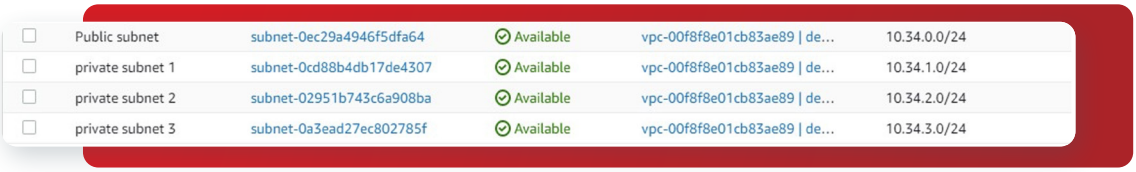
Click on **Add new subnet** to add the second subnet information:

a. **Subnet name**: private subnet 3

b. **Availability Zone (same zone as in VPC)**: us-east-1b

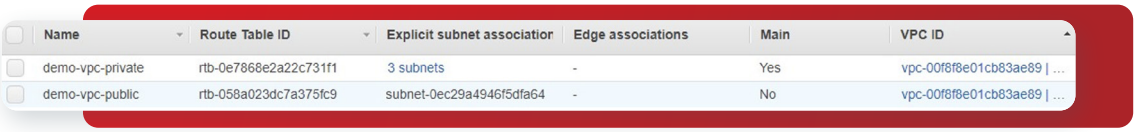
c. **IPv4 CIDR block (should be in different subnet but same CIDR block as in VPC)**: 10.34.3.0/24

Click the **Create Subnet** button. The result of this step and the previous step 5 is a public subnet for the bastion host, private subnet 1 for application-specific traffic, and private subnet 2 and subnet 3 are for LLT1 and LLT2 traffic.



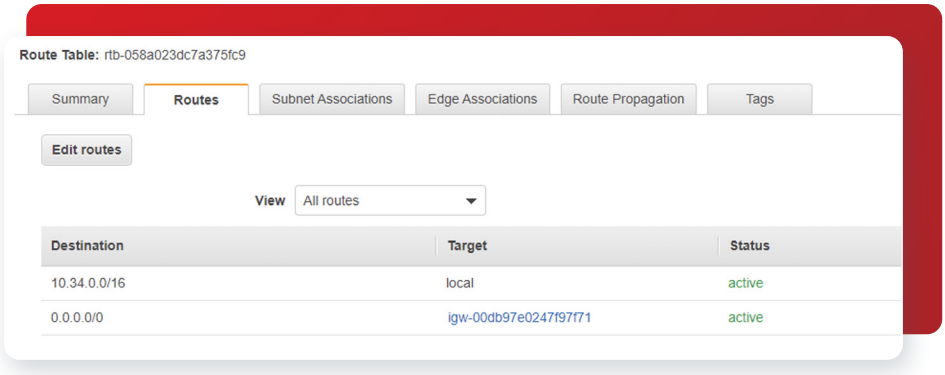
Public subnet	subnet-0ec29a4946f5dfa64	Available	vpc-00f8f8e01cb83ae89 | de...	10.34.0.0/24
private subnet 1	subnet-0cd88b4db17de4307	Available	vpc-00f8f8e01cb83ae89 | de...	10.34.1.0/24
private subnet 2	subnet-02951b743c6a908ba	Available	vpc-00f8f8e01cb83ae89 | de...	10.34.2.0/24
private subnet 3	subnet-0a3ead27ec802785f	Available	vpc-00f8f8e01cb83ae89 | de...	10.34.3.0/24

7. Validate the routes by selecting the **Route Tables** on the left pane. In this example, the route table with one **Explicit subnet association**, modify the **name** to demo-vpc-public, and the one with 3 subnets association modify the **name** to demo-vpc-private.

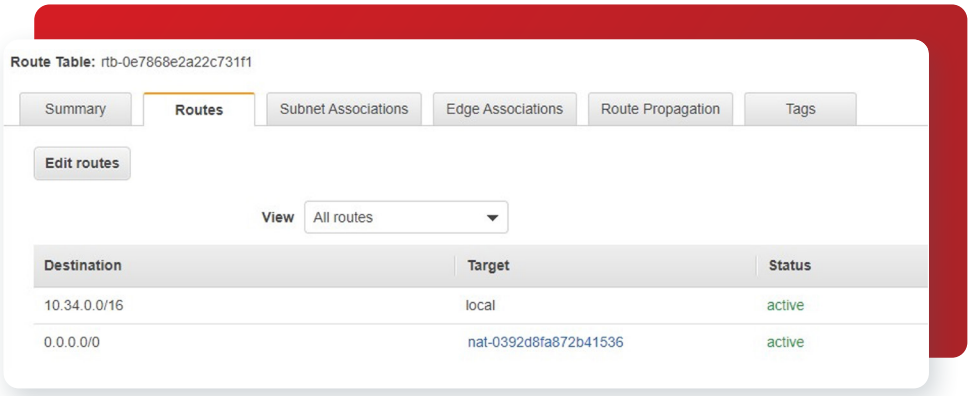


Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
demo-vpc-private	rtb-0e7868e2a22c731f1	3 subnets	-	Yes	vpc-00f8f8e01cb83ae89 | ...
demo-vpc-public	rtb-058a023dc7a375fc9	subnet-0ec29a4946f5dfa64	-	No	vpc-00f8f8e01cb83ae89 | ...

From the routes created, within the VPC, the public subnet is connected to the Internet Gateway and associated with the public subnet 10.34.0.0/24.



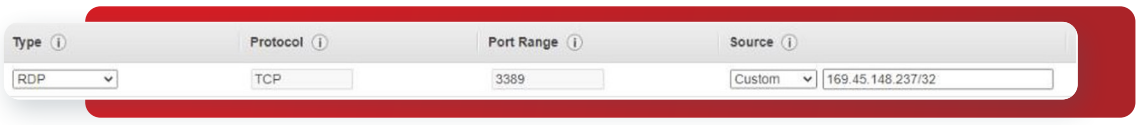
The private subnets have a NAT gateway that allows EC2 instances inside these subnets to access the Internet and is associated with the other 3 private subnets 10.34.1.0/24, 10.34.2.0/24, and 10.34.3.0/24.



Step 2: Creation and Configuration of the Bastion Host

To access the EC2 instances created in the private subnet externally from a local computer, a bastion host is created and deployed. An EC2 instance of type and Windows 2019 Base server is created in this step and placed on a public subnet. An external IP will be assigned to be able to access the Internet.

1. Log on to the AWS management console.
2. Select the AWS service **EC2**.
3. From the EC2 dashboard, click on **Launch instance**.
4. Enter in the search field **windows**. Click on the left pane, "**AWS Marketplace**," and select Microsoft Windows Server 2019 Base.
5. Choose an Instance Type size. In this example, selected **t2.medium** (2 vCPUs and 4 GiB of memory).
6. Click **Next Configure Instance Details**. Fill in the instance details as follows:
   a. **VPC Network** created in the previous section: demo-vpc
   b. **Subnet**: Public Subnet
   c. **Auto-Assign Public IP**: Enable
7. Click **Next Add Storage**. Verify the storage size should be 30 GB or higher.
8. Click **Next Configure Security Groups** and fill in as follows:
   a. **Assign a security group**: Create a new security group
   b. **Security Group name**: demo-sg-bastion
   c. Remove SSH type and keep RDP, specify the **IP** address of the computer that will be used for RDP into the bastion host under the Source field. To determine the IP address of the computer, from a browser on a local computer go to http://checkip.amazonaws.com/. This site will return the IP address of the local computer.



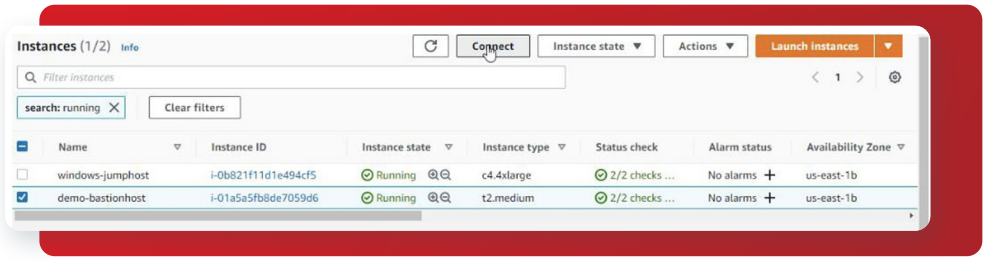
9. Click **Review and Launch**.
10. Click **Launch**.

11. Create a new key pair, provide a **Key pair name** (for example, dpair), and **Download Key Pair**. Save the key, dpair.pem, on the local computer.

12. Click **Launch Instances**.

13. Click **View Instances**.

14. Edit the name of the EC2 instance: **demo-bastionhost**. Select the EC2 instance and click on **Connect**.



15. Select the **RDP client tab**. Click on **Get password**. Click **Browse** and specify the location of the key pair, dpair.pem , downloaded in step 10. Click **Decrypt password**.

16. Save and copy the **Password** and click on **Download remote desktop file**. The password will be used to logon to the bastion host from the local computer.

17. Double-click on **remote desktop file** (i.e. demo-bastionhost.rdp). Click on **connect** and enter the **password** copied in step 15.

18. Once on the bastion host, to download files using the Windows default browser (IE) without the security checks, turn off **IE security enhanced** from the Server Manager. Also turn off **Windows Defender Firewall**.

19. Start the browser (IE), download the utilities listed in Table 2, and install accordingly.

Table 2. Utilities

Utility	Purpose	Download Site
Google Chrome	Browser	https://www.google.com/chrome/
Moba Xterm	SSH to EC2 Linux Instances	https://mobaxterm.mobatek.net/download-home- edition.html
WinSCP	Copy files from Windows to Linux	https://winscp.net/eng/download.php

Step 3: Creation of EC2 Instances for a Two-Node InfoScale Cluster

There are two EC2 instances configured in this step for setup of a two-node InfoScale cluster. It is typically a best practice to select an interface that is sized specifically for the application. In this example, we have selected an instance type of c4.2xlarge. We selected this instance type because three network interfaces can be attached to the instance and it has enough CPU and memory to run MySQL. In this example, we will create three network interfaces for application-specific and LLT traffic.

Create EC2 Instances

1. Log on to the AWS management console.

2. Select the AWS service **EC2**.

3. From the EC2 dashboard, click on **Launch instance**.

4. In the search field, enter **rhel**. Click on the left pane, "**AWS Marketplace**" and **select** RHEL 7. In this example, we use RHEL 7.7. For successful deployment, it is important to select both an operating system and kernel that are supported by InfoScale. Refer to the list of supported RHEL kernel versions on the Veritas Sort page.

5. Choose an Instance Type size. In this example, we have selected **c4.xlarge** (8 vCPUs and 15 GiB of memory). This instance type also allows allocation of a maximum of four network interfaces. In this example, we have created three network interfaces: one for application-specific traffic and two for inter-cluster communication (LLT1 and LLT2).

6. Click **Next Configure Instance Details**. Fill in the instance details as follows:

   a. **Number of instances**: 2

   b. **VPC Network created in previous section**: demo-vpc

   c. **Subnet**: private subnet 1

   d. Under **Network interfaces section**, click on **Add Device** to add another network interface. Select **subnet**: private subnet 2. You can add only two network interfaces at this point. The third one will be added later once the instance has been launched.

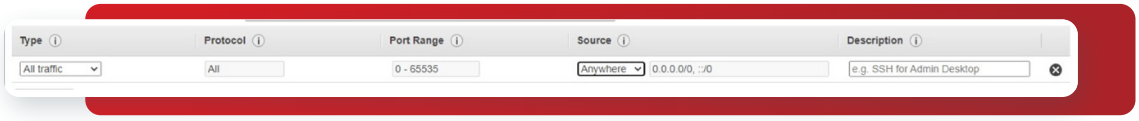
7. Click Next **Add Storage**. Modify the storage size of the Root volume to **60GiB** or higher. Click **Add New Volume** to add another volume for the swap space needed for InfoScale. The recommended volume size for swap space should be double the size of the amount of memory provisioned. In this example, specify the size of the EBS Volume to be 40 GiB. Place a **check mark** on **Delete on Termination** for the EBS volume.

8. Click on **Next Tags**, click **Next Configure Security Groups**, and fill in as follows:

   a. **Assign a security group**: Create a new security group

   b. **Create new security group. Security Group name**: demo-sg-private

   c. Remove SSH type and RDP rules, click **Add Rule**, and specify Type to be **All traffic** and source to be **Anywhere**. Because these EC2 instances are deployed within a private subnet, all traffic is allowed. However, if you want to limit the traffic to specific ports needed by InfoScale, refer to the InfoScale Ports and Services documentation.



9. Click **Review and Launch**. Then, click **Launch**.

10. Use the same **key pair** from the previous section, **dpair.pem**.

11. Click **Launch Instances**.

12. Click **View Instances**.

13. Edit the names of the instances you just created: **demosys1** and **demosys2**

14. To configure the third network interface in the third private network to be used for LLT, on the left pane under Network & Security, select **Network Interfaces**. To attach a network interface to the first EC2 instance, demosys1, click on **Create network** interface and fill out the following fields:

   a. **Description**: demosys1-priv

b. **Subnet**: private subnet 3 (10.34.3.x)

c. **Use existing Security Group**: demo-sg-private

Repeat the above steps for the network interface to be attached to the second EC2 instance, demosys2. Click on Create network interface and fill out the following fields:

a. **Description**: demosys2-priv

b. **Subnet**: private subnet 3 (10.34.3.x)

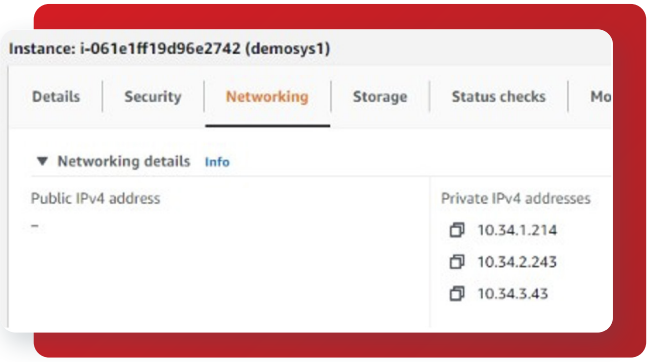
c. **Use existing Security Group**: demo-sg-private

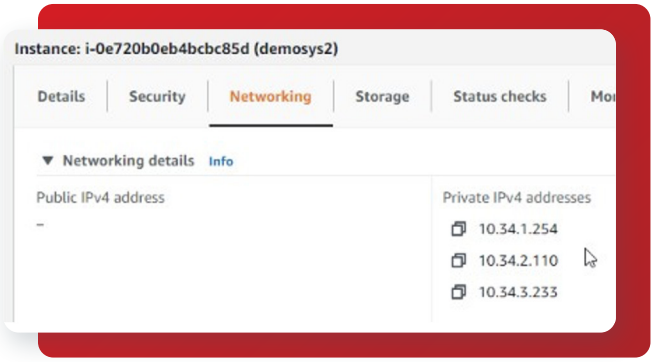
15. Attach the network interfaces you just created to each of the EC2 instances:

a. Select **demosys1-priv** interface, click on the **Actions** button at the top, and select **Attach** to EC2 instance demosys1.

b. Select **demosys2-priv** interface, click on the **Actions** button at top, and select **Attach** to EC2 instance demosys2.

16. Select each of the instances and review the network IPs of each instance. Note down the network IPs addresses for each of the EC2 instances. The IPs on private subnet 1 (10.34.1.x) are IP linked to network interface "eth0" on each interface and IP used for SSH. IPs on private subnet 2 (10.34.2.x) and private subnet 3 (10.34.3.x) are linked to eth1 and eth2 network interfaces for LLT1 and LLT2 traffic.



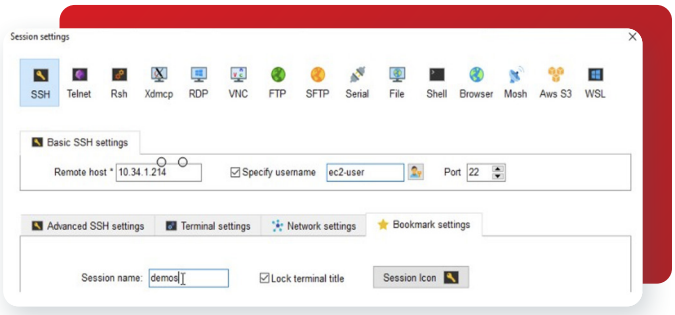


Connect to an EC2 Instance

1. Open Notepad from the bastion host and open the dpair.pem from the local computer. Cut and paste the dpair.pem file from your local computer to the bastion host Notepad and save as dpair.pem. This key pair is used to initially SSH onto the EC2 instances as ec2-user.

2. Start an SSH session from MobaXterm and enter the IP address of the EC2 instance (demosys1) on subnet 10.34.1.x (i.e. 10.34.1.214) and specify the user as ec2-user. Select the **Advanced SSH setting** tab, place a check mark on **Use private key**, and enter **path** of dpair.pem file on the bastion host. Select **Bookmark** settings, enter the Session name demosys1, and click **Ok**.

3. Repeat Step 2 for the other EC2 instance (demosys2).



Step 4: Configuration of Nodes

Configuration of instances in preparation for InfoScale Enterprise installation involves modifying hostnames, creating swap space, modifying the network interfaces, routes, and rules for application- and InfoScale-specific traffic (LLT1 and LLT2), passwordless SSH, and hosts file so you can access instances using names. All these steps require root access to the instance.

Modification of Hostname

1. After you SSH onto the host, sudo as root: **sudo su -**

2. Modify the hostname on the EC2 instance: **hostnamectl set-hostname <name i.e. demosys1>**

3. Exit out of the shell and re-logon so the name of hosts changes.

Make Swap Space

1. Make a swap space. Use the disk device not being used as the root volume. In this example, we added a 40 GiB disk to the EC2 instance during creation. Described below are the steps to make the swap space. Execute the commands in bold font.

```
# gdisk /dev/xvdb
GPT fdisk (gdisk) version 1.0.3
Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present
Creating new GPT entries.
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-83886046, default = 2048) or {+-}size{KMGTP}: L
First sector (34-83886046, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-83886046, default = 83886046) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'
Command (? for help): w
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdf.
The operation has completed successfully.
# partprobe /dev/xvdb

#  lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  40G 0 disk
├─xvda1 202:1    0  1M  0 part
└─xvda2 202:2    0  40G  0 part /
xvdb  202:80   0  40G  0 disk
└─xvdb1 202:81  0  40G  0 part

# mkswap /dev/xvdb1
Setting up swapspace version 1, size = 40 GiB (42948599808 bytes)
no label, UUID=4b48ac93-1c8d-455b-9365-2cb18bde3e23
```

2. Modify the **/etc/fstab** and include the UUID from the output of the mkswap command above and specify it as swap with the options as follows:

UUID=4b48ac93-1c8d-455b-9365-2cb18bde3e23 swap defaults,_netdev,x-initrd.mount

Execute the following commands to enable the swap device:

a. swapon -a

b. swapon -s

c. dracut -f -v

d. Validate the swap space by executing free -h:

```
# free -h
```

total	used	free	shared	buff/cache	available
Mem:	14 GB	256 MB	14 GB	16 MB/252 MB	14 G B
Swap:	39 GB	0 B	39 GB		

Modify Networking, Routes, and Rules

1. In this example, network scripts are used instead of Network Manager for enabling the interfaces, rules, and routes. Execute the following steps to enable network scripts and disable NetworkManager:

a. Disabling Network Manager can remove the contents of this /etc/resolv.conf. Prior to disabling Network Manager and enabling network scripts, save a copy of the /etc/resolv.conf.

cp /etc/resolv.conf /etc/resolv.conf.orig

b. Enable networking services: **systemctl enable network**

Stop NetworkManager: **service NetworkManager stop**

c. Disable NetworkManager from starting at reboot: **systemctl disable NetworkManager**

d. Check NetworkManager has been disabled:

systemctl list-unit-files | grep NetworkManager

2. In RHEL, only network interface eth0 has been configured and the other two network interfaces, eth1 and eth2, are not configured. In this step, you will create the configuration files for eth1 and eth2.

a. Change directory to /etc/sysconfig/network-scripts: **cd /etc/sysconfig/network-scripts**

b. Copy ifcfg-eth0 to ifcfg-eth1: **cp ifcfg-eth0 ifcfg-eth1**

c. Edit ifcfg-eth1 : **vi ifcfg-eth1**

Modify the following fields:

```
BOOTPROTO=static
DEVICE=eth1
HWADDR=<ether address of network interface eth1 from output of ifconfig>
```

Add the following lines:

```
NM_CONTROLLED=no
IPADDR=<IPADDR in private subnet 2 for instance,
10.34.2.243> NETMASK=<netmask for instance 255.255.255.0>
STARTMODE=auto
```

d. Copy the ifcfg-eth1 to ifcfg-eth2 and edit ifcfg-eth2 by modifying the following fields:

```
DEVICE=eth2
HWADDR=<ether address of network interface eth2 from output of ifconfig>
IPADDR=<IPADDR in private subnet 3 for instance, 10.34.3.43>
```

e. Bring up the network interfaces:

```
ifup eth1
ifup eth2
```

f. Validate the interfaces are up and have IP addresses assigned to them by executing: **ifconfig -a**

Sample output:

```
[root@demosys1 ~]# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.34.1.214  netmask 255.255.255.0  broadcast 10.34.1.255
        inet6 fe80::c3d:2ff:feff:ea0b  prefixlen 64  scopeid 0x20<link>
        ether 0e:3d:02:ff:ea:0b  txqueuelen 1000  (Ethernet)
        RX packets 1464  bytes 405752 (396.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1727  bytes 285509 (278.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.34.1.111  netmask 255.255.255.0  broadcast 10.34.1.255
        ether 0e:3d:02:ff:ea:0b  txqueuelen 1000  (Ethernet)

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.34.2.243  netmask 255.255.255.0  broadcast 10.34.2.255
        inet6 fe80::c48:6eff:fec5:b6df  prefixlen 64  scopeid 0x20<link>
        ether 0e:48:6e:c5:b6:df  txqueuelen 1000  (Ethernet)
        RX packets 4239  bytes 454752 (444.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4768  bytes 884932 (864.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.34.3.43  netmask 255.255.255.0  broadcast 10.34.3.255
        inet6 fe80::c1f:f6ff:fefb:8921  prefixlen 64  scopeid 0x20<link>
        ether 0e:1f:f6:fb:89:21  txqueuelen 1000  (Ethernet)
        RX packets 4289  bytes 449220 (438.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4773  bytes 735597 (718.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3. Routes and rules need to be specified for eth1 and eth2 so traffic can be routed appropriately for these interfaces. Without specifying the routes and rules, these network interfaces are not pingable or accessible.

a. Modify the routing tables file rt_tables and add number and string. The string name associated with number is used when adding the routes and defining the rules for the interface.

```
echo 2 eth1_rt >> /etc/iproute2/rt_tables echo 3 eth2_rt >> /etc/iproute2/rt_tables
```

b. Manually add the routes and rules by using the command:

```
ip route add default via <subnet gateway> dev <network interface> table <name of table>
ip rule add from <ip address> dev <network interface> lookup <name of table> prio <number>
ip rule add to <IPv4 CIDR block of private subnet> lookup <network interface> prio <number>
```

For example, for eth1 routes and rules:

```
ip route add default via 10.32.1.1 dev eth1 table eth1_rt
ip rule add from 10.32.1.141 lookup eth1_rt prio 1000
ip rule add to 10.32.1.0/24 lookup eth1_rt prio 1000
```

And, for eth2 routes and rules:

```
ip route add default via 10.33.1.1 dev eth2 table eth2_rt ip rule add from 10.33.1.75 lookup eth2_
rt prio 1000
ip rule add to 10.33.1.0/24 lookup eth2_rt prio 1000
```

c. Check the routes and rules using these commands:

```
ip route show table eth1_rt ip route show table eth2_rt ip rule show
```

Example output:

```
[root@demosys1 ~]# ip route show table eth1_rt
default via 10.34.2.1 dev eth1
[root@demosys1 ~]# ip route show table eth2_rt
default via 10.34.3.1 dev eth2
[root@demosys1 ~]# ip rule show
0:      from all lookup local
1000:   from 10.34.2.243 lookup eth1_rt
1000:   from all to 10.34.2.0/24 lookup eth1_rt
1000:   from 10.34.3.43 lookup eth2_rt
1000:   from all to 10.34.3.0/24 lookup eth2_rt
32766:  from all lookup main
32767:  from all lookup default
```

d. Validate that eth1 and eth2 are accessible by pinging the IP addresses from the bastion host. If unable to ping, validate the IP addresses and IPv4 CIDR of the routes and rules.

e. Re-validate that there is a route to the Internet by running a ping of www.google.com. If unable to ping out, check the /etc/resolv.conf to validate that the nameserver and search are set. If not, then copy the saved copy of /etc/resolv.conf done in Step 5a.

4. Make the rules and routes persistent by creating route and rule files.

a. Create a route file for eth1 and eth2 (route-eth1 and route-eth2). For example, contents of route-eth1 and route-eth2.

```
# cat route-eth1
default via 10.34.2.1 dev eth1 table eth1_rt
# cat route-eth2
default via 10.34.3.1 dev eth2 table eth2_rt
```

b. Create rule files for eth1 and eth2 (rule-eth1 and rule-eth2). For example, contents of rule-eth1 and rule- eth2.

```
# cat rule-eth1
from 10.34.2.243 lookup eth1_rt prio 1000
to 10.34.2.0/24 lookup eth1_rt prio 1000

#cat rule-eth2
from 10.34.3.43 lookup eth2_rt prio 1000 to 10.34.3.0/24 lookup eth2_rt prio 1000
```

5. Repeat steps 1-8 on the second EC2 instance, using the hostname of second EC2 instance (for example, demosys2) and the private IP address associated with the second EC2 instance.

Modify Hosts File and Passwordless Access

1. Once you have configured both EC2 instances, modify the **/etc/hosts** file on each instance including the bastion host file (**C:/Windows/System32/drivers/etc/hosts**) with the IP address of eth0 and the hostnames of Linux instances and the IP address and name of the bastion host instance. For example:

```
10.34.1.214 demosys1
10.34.1.254 demosys2
10.34.0.4 EC2AMAZ-ADJDQF9.ec2.internal
```

2. InfoScale requires passwordless access to each instance. Execute the following on each instance:

a. Set a password for root: **passwd root**

b. Edit the **/etc/ssh/sshd_config** file and modify as follows:

```
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

c. Restart the sshd service: **service sshd restart**

d. Create a key-gen and copy to the other instance. For example, ssh-keygen (NOTE: no need for a passphrase).

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@demosys2 ssh root@demosys1
```

e. Repeat steps a-d on the other EC2 instance on demosys2.

3. InfoScale operates in permissive mode in Linux. Therefore, modify the SELinux policy for RHEL to permissive mode.

a. Edit the config file **vi /etc/selinux/config** and modify the SELINUX variable to permissive: SELINUX=permissive

b. Reboot.

c. Once booted, check the mode setting using command: **getenforce**

4. Re-validate each network interface IP address on each subnet is pingable from each EC2 instance and the bastion host and validate that the Internet is pingable from each EC2 instance.

Step 5: Installation of AWS CLI Bundle and Configuration of IAM Role

Prior to installation of InfoScale, you need to configure AWS CLI and IAM roles and policy on each Linux EC2 instance.

1. Log on as root on one EC2 instance.

2. Install AWS CLI version 2 (https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html#cliv2-linux- install)

   a. Install unzip: **yum install unzip**

   b. Execute the following to pull the AWS CLI bundle and install.

   curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" unzip awscliv2.zip
   sudo ./aws/install

   c. Validate the install of AWS CLI by executing

   # /usr/local/bin/aws --version

3. Log on as root to the other EC2 instance and repeat step 2.

4. The InfoScale Agents AWSIP and EBSVol are installed by default during the installation and configuration of InfoScale. An IAM policy and role is required and needs to be created and attached to each instance. Create the IAM policy as follows:

   a. Log on to the AWS management console.

   b. Select the AWS Service **IAM**.

   c. On the left pane, select Policies and click on **Create policy**. Click on the **JSON** tab and enter the following within the Action brackets. These actions are required for the EBSVol and AWSIP InfoScale Agents.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
       "Action": [
            "ec2:DescribeVolumes",
            "ec2:AttachVolume",
            "ec2:DetachVolume",
            "ec2:AssignPrivateIpAddresses",
            "ec2:DescribeAddresses",
            "ec2:DisassociateAddress",
            "ec2:AssociateAddress",
            "ec2:AssignPrivateIpAddresses",
            "ec2:AssignIpv6Addresses",
            "ec2:UnassignIpv6Addresses",
            "ec2:DescribeInstances",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:CreateRoute",
```

```
            "ec2:DeleteRoute",
            "ec2:ReplaceRoute"
        ],
        "Effect": "Allow",
        "Resource": "*"
      }
    ]
}
```

d. Click on **Next Tags** and click **Review**.

e. **Enter Name**: InfoScalePolicy

f. **Enter Description**: Policy for EBSVol and AWSIP

g. Click on **Create policy**.

5. Create an IAM role.

a. On the left pane, click on **Roles**.

b. Click on **Create role**.

c. Select **AWS Service** and **Use case**: EC2.

d. Click **Next Permissions** and search for the policy created in previous step, **InfoScale Policy**. Place a **check mark** in the box to select InfoScalePolicy

e. Click **Next Tags** and click **Next Review**.

f. **Add Role name**: InfoScaleRole

g. Click **Create role**.

6. Attach a role to each Linux EC2 instance (demosys1 and demosys2).

a. Go to **Services** and select **EC2**.

b. Click on **Instances Running** from the EC2 dashboard.

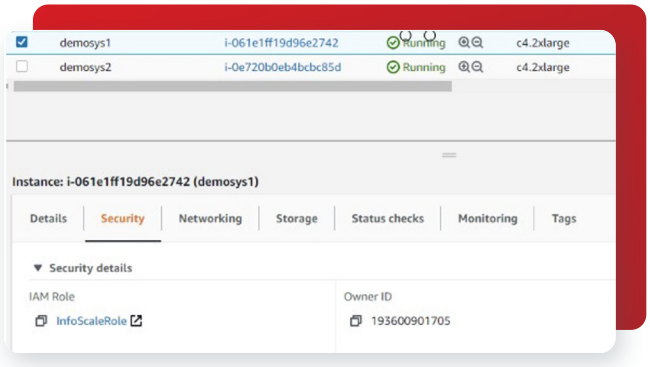
c. Select the first instance, **demosys1**. Click on **Actions -> Security -> Modify IAM Role**.

d. Choose the IAM role you created in step 5: **InfoScaleRole**.

e. Click **Save**.

f. **Repeat** steps 6c to 6e for the other EC2 instance, demosys2.

g. Validate role settings by selecting each instance and expanding the **Security** tab details.



For more information on AWS IAM roles and policies for the InfoScale Agents, refer to the Veritas Cluster Server

7.4.3 Bundled Agents Reference Guide:

- EBSVol Agent: https://www.veritas.com/content/support/en_US/doc/79620650-145506806-0/v128221661-145506806
- AWSIP Agent: https://www.veritas.com/content/support/en_US/doc/79620650-145506806-0/v125209990-145506806
- Another good resource is the Veritas InfoScale Cloud Solutions Guide: https://www.veritas.com/content/support/en_US/doc/130803809-130803829-0/v132424972-130803829

Step 6: Installation and Configuration of InfoScale Enterprise

This section steps through the InfoScale Enterprise installation and configuration process. In this example, we use InfoScale Enterprise 7.4.3. For simplicity, we do not use certain features of InfoScale Enterprise in this deployment such as I/O fencing and enforced security. This walk-through example is meant to be an introduction. Refer to the InfoScale product documentation for more detailed information at https://sort.veritas.com/documents.

1. Download the latest InfoScale Enterprise binaries and patches from https://www.veritas.com/content/support/en_US/downloads to the bastion host. You will need an account to download. In this example, we have downloaded the *.tar.gz version.

2. Use WinSCP to copy the binaries to one of the EC2 RHEL instances, demosys1.

3. Log on to the EC2 instance, demosys1, where you copied the binaries. Extract the InfoScale Enterprise binaries: tar xvfz *.tar.gz

4. Change into the directory from which the binaries were extracted and **Run** the installer.

5. Run a pre-check to double-check all the prerequisites for InfoScale Enterprise installation.

   a. Enter P to Perform a Pre-Installation Check

   b. Enter **4** to select Veritas InfoScale Enterprise.

   c. Enter **4** to select Storage Foundation Cluster File System HA (SFCFSHA).

   d. Enter **demosys1 demosys2** for system names to install.

   e. Resolve any prerequisites. In this example, the only prerequisites not fulfilled are the patches and rpms. Use option **1** to automatically install the required patches and rpm on both EC2 instances.

6. After pre-checks, enter **y** to continue with the installation, **Accept the license**, and enter **y** to install on both instances.

7. After successful installation, select option **2** to enable keyless licensing and select **4** to license InfoScale Enterprise.

8. Enter **y** to configure InfoScale Enterprise on demosys1 and demosys2.

9. Enter **telemetry.veritas.com** for the IP address of the edge server that is already preconfigured.

10. As mentioned, we do not use I/O fencing in this example deployment. Therefore, in the next set of questions, respond as follows:

    Do you want to configure container storage in Enabled mode? [y,n,q,?] (n)
    Do you want to configure I/O Fencing in enabled mode? [y,n,q,?] (y) **n**

11. Enter the unique cluster name: [q,?] **dcluster**.

12. The next question relates to the configuration of the heartbeat links. For cloud scenarios, configure the heartbeat links **using LLT over UDP** for use within AWS by entering as follows:

How would you like to configure heartbeat links? [1-5,b,q,?] (5) **2**

13. The network interfaces eth1 and eth2 and their associated IP addresses will be auto-discovered, so for the next set of questions, just hit enter and use the **defaults**. For example:

Enter the NIC for the first private heartbeat link on demosys1: [b,q,?] (eth1)

Some configured IP addresses have been found on the NIC eth1 in osys1. Do you want to choose one for the first private heartbeat link? [y,n,q,?] (y)

Please select one IP address: [1-2,b,q,?] (1) 1

Enter the UDP port for the first private heartbeat link on demosys1: [b,q,?] (50000)

Enter the NIC for the second private heartbeat link on demosys1: [b,q,?] (eth2) Please select one IP address: [1-2,b,q,?] (1) 1

Enter the UDP port for the second private heartbeat link on osys1: [b,q,?] (50001)

Are you using the same NICs for private heartbeat links on all systems? [y,n,q,b,?] (y)

Some configured IP addresses have been found on the NIC eth1 in demosys2, Do you want to choose one for the first private heartbeat link? [y,n,q,?] (y)

Please select one IP address: [1-2,b,q,?] (1) **1**

The UDP Port for this link: 50000

Some configured IP addresses have been found on the NIC eth2 in demosys2. Do you want to choose one for the second private heartbeat link? [y,n,q,?] (y)

Please select one IP address: [1-2,b,q,?] (1) **1**

The UDP Port for this link: 50001

Enter a unique cluster ID number between 0-65535: [b,q,?] (57862)

14. No need to check the cluster ID because there will only be one cluster deployed in this example. Therefore, answer **no** to this question.

Would you like to check if the cluster ID is in use by another cluster? [y,n,q] (y) **n**

15. Validate the correctness of the cluster information selected and hit **enter** to continue with installation.

16. The next series of questions deals with configuration of the Cluster Virtual IP. The Cluster Virtual IP is associated with the eth0 network interface. The virtual IP address should be **an IP address not in use and in the same subnet as the private subnet 1** (10.34.1.x). Example responses to questions:

Do you want to configure the Virtual IP? [y,n,q,?] (n) **y**
Enter the NIC for Virtual IP of the Cluster to use on osys1: [b,q,?] (eth0)
Is eth0 to be the NIC used by all systems? [y,n,q,b,?] (y)
Enter the Virtual IP address for the Cluster: [b,q,?] **10.34.1.111**
Enter the NetMask for IP 10.31.1.113: [b,q,?] (255.255.255.0)
Is this information correct? [y,n,q] (y) **y**

17. Other questions were asked, and we entered the following:

Would you like to configure VCS cluster in secure mode [y,n,q,?] (y) n

Are you sure that you want to proceed with non-secure installation? [y,n,q] (n) y

Do you wish to accept the default cluster credentials of 'admin/password'? [y,n,q] (y) Do you want to add another user to the cluster? [y,n,q] (n)

Do you want to configure SMTP notification? [y,n,q,?] (n) Do you want to configure SNMP notification? [y,n,q,?] (n)

Do you want to configure the Global Cluster Option? [y,n,q,?] (n)

Do you want to stop InfoScale Enterprise processes now? [y,n,q,?] (y)

18. After successful completion of configuration, validate the running status of cluster. Run:

/opt/VRTS/bin/hastatus -summary

19. Add the following line into the .bashrc file to add the path of InfoScale binaries: /opt/VRTS/bin and source the **.bashrc** file to reload the variables.

export PATH=$PATH:/opt/VRTS/bin

20. You need to register the cluster virtual IP assigned in step 16 on the primary EC2 instance (demosys1) as a secondary IP. This IP is assigned to the secondary EC2 instance, demosys2, after a failover. The InfoScale AWSIP Agent is responsible for attaching and detaching this IP to the EC2 instances within the cluster.

a. Log on to AWS management console.

b. Select **AWS Service**: EC2

c. Click on **running instances** and select the **primary instance, demosys1**.

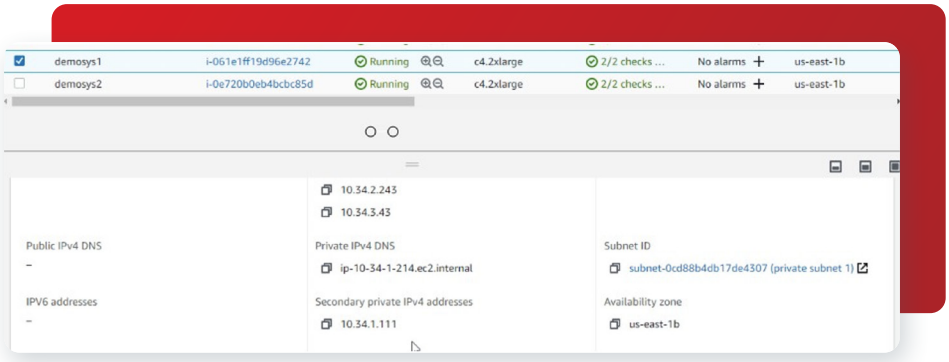
d. Click on **Action->Network->Manage IP**.

e. Expand **eth0** and click on **Assign New IP address**.

f. **Enter the cluster virtual IP**: 10.34.1.111

g. Click **Save** and **Confirm**.

h. Validate the secondary private IP has been set in the Network settings of the instance.

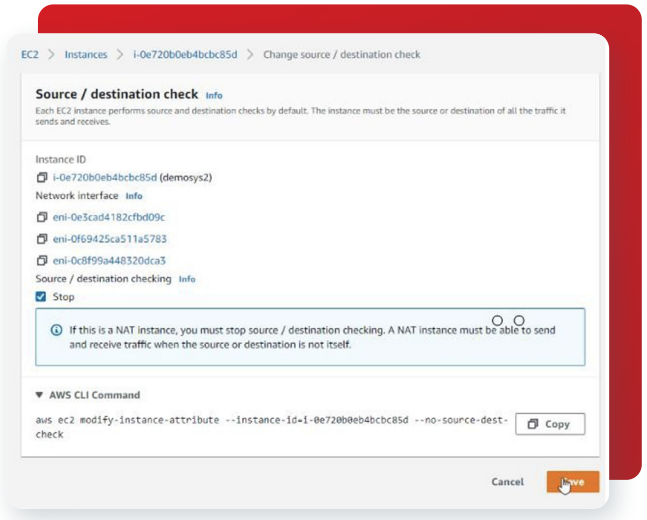


21. Stop the Source/destination check in the network of each EC2 instance.

a. Click on the EC2 instance, **demosys1**, and click on **Actions->Network->Change source/destination check**.

b. Place a **check mark on Stop** and click **Save**.

c. Repeat steps **a-b** for the second EC2 instance, **demosys2**.



Step 7: Installation of Veritas InfoScale Operations Manager

Veritas InfoScale Operations Manager (VIOM) is a useful tool for the visualization and management of InfoScale cluster(s). In this example, we have installed and configured the VIOM on the bastion host.
For more information on the VIOM, refer to https://www.veritas.com/content/support/en_US/doc/Viom_quickstart_guide_74_pdf.

1. Modify the security group for the bastion host to accept traffic from the EC2 instances on the private subnet from the AWS management console.

   a. Log on to AWS management console.

   b. Select **AWS Service**: EC2.

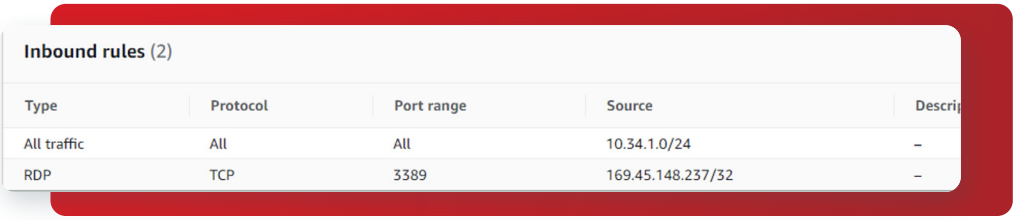
   c. Select **Running Instances**.

   d. Select the EC2 instance for the bastion host, **demo-bastionhost**.

   e. At the bottom pane, click on the **Security** tab and select the security group: **demo-sg-bastion**.

   f. Click on **Edit inbound rules**.

   g. Click **Add rule** and select Type: All traffic from Custom source 10.34.1.0/24.



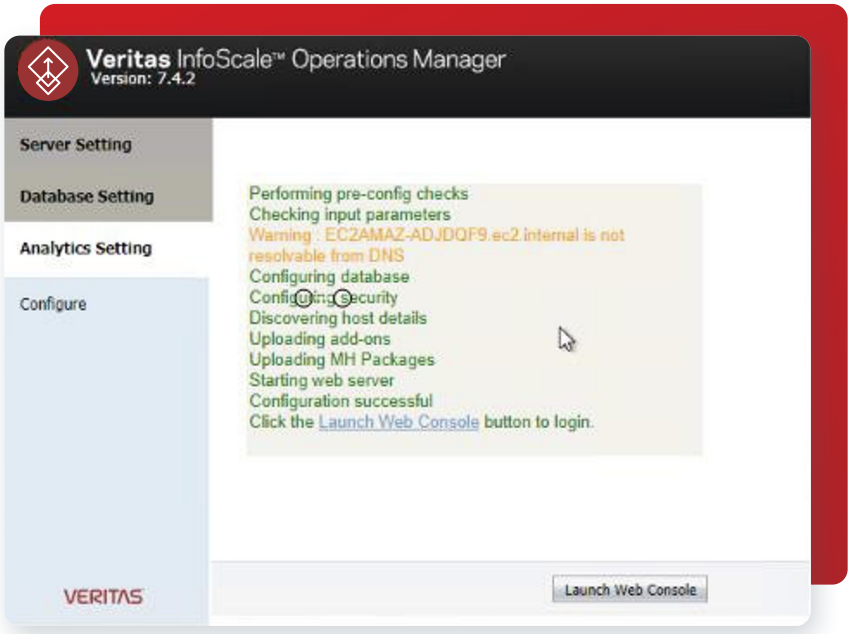
2. RDP onto the bastion host.

3. Download the latest Veritas InfoScale Operations Manager (VIOM) and patches from https://www.veritas.com/content/support/en_US/downloads to the bastion host. You will need an account to download. In this example, we have downloaded VIOM version 7.4.2 (*Win *Full*.zip file).

4. Extract the file and double-click on the application installer to install the application.

5. Once it is installed, it refers you to a browser and opens the site http://localhost:5634 to do the configuration. Use the bastion host Administrator credentials to log on to site. It will automatically pick up the Server Name and Address of the bastion host. Note: The EC2 instance names for the InfoScale Cluster and bastion host should be in the hosts file on each instance so the names are resolvable. Ensure you do so in the Host Configuration section, step 10. Otherwise, the configuration can hang or fail during the Analytics Setting phase.

6. After configuration, click on **Launch Web Console** (for example, https://ec2amaz-adjdqf9.ec2.internal:14161/).

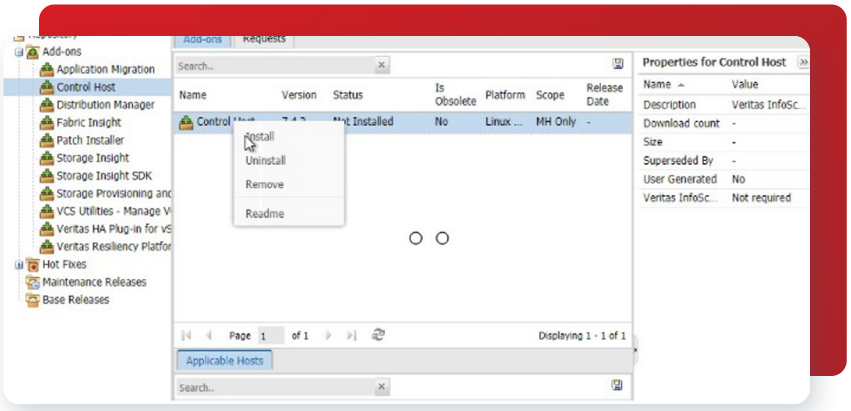


7. Enter the bastion host Administrator credentials to log on to VIOM management.

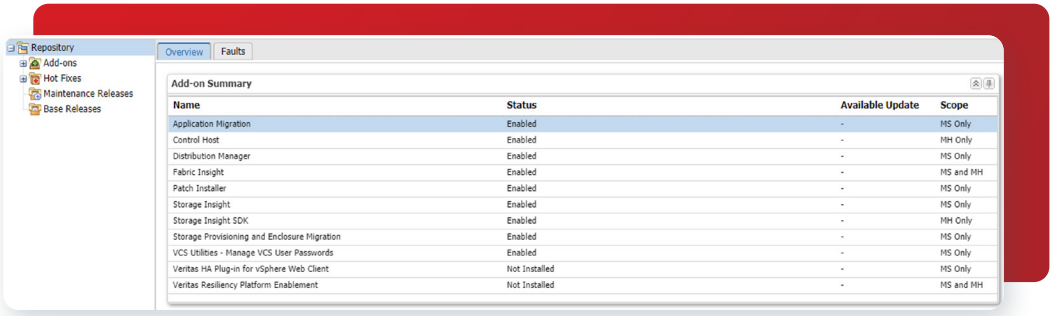
8. Install the **Add-ons** so the hosts can be discoverable and controlled and other management services are enabled.

   a. Click on the cog **settings** button on the top right of the window and click on the Deployment icon.

   b. Expand **Add-ons**, select **Control Host**, and right-click and select **Install**.



Repeat step b for any other services you want to enable such as storage provisioning or application migration.



9. Add the hosts to be managed, demosys1 and demosys2.

   a. Click on the **arrow** at the top left of the page to get back to **Settings**.

   b. Select the **Host** icon.

   c. Click the **Add Hosts** button and select **Agent**.

   d. Enter the **hostname**, demosys1, and the **username** and **password** for root.

   e. Click **Finish**.

   f. Repeat steps c-e for the second host, demosys2.

10. Download and install Hot Fixes 7.4.2.300 to the bastion host and each managed host, demosys1 and demosys2.

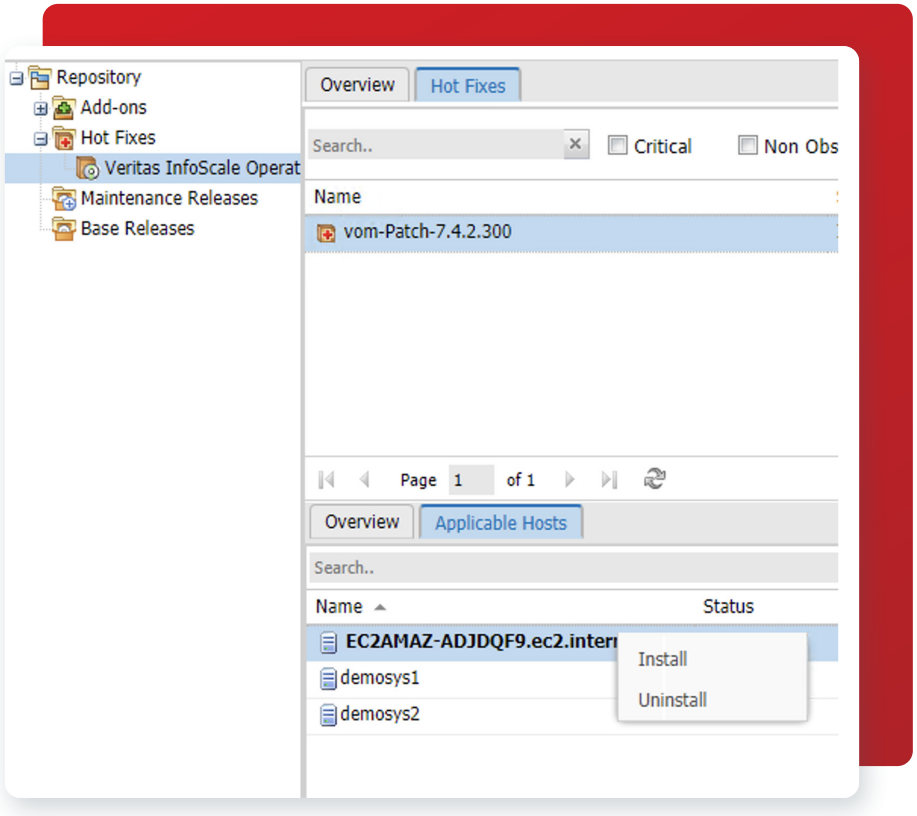
   a. Click on the **arrow** at the top left of the page to get back to **Settings**.

   b. Select the **Deployment** icon.

   c. Click on "**Upload solutions**." Browse to where you downloaded the hot fixes and click on "**Upload**."

   d. Expand **Hot Fixes** on the left pane and then click on the **Hot Fixes** tab.

   e. Click on the **Applicable Hosts** tab at the bottom, select bastion host, and right-click to Install hotfix on each host and the bastion host.



   f. Repeat step d for the other hosts, demosys1 and demosys2.

Step 8: Creation and Configuration of Disk for MySQL Data

For the MySQL database, an AWS EBS volume is allocated from the AWS management console. In addition, the disk is scanned and configured via the command line. Configure the disk and service group to mount the volume in preparation for the MySQL database to use it.

1. Log on to the AWS management console.
2. From the AWS Services, select **EC2**.
3. On the left pane, click on **Volumes**.
4. Click on **Create Volume**.
    a. Select **size** (for example, 100 GiB).
    b. Select the **same Availability Zone where the EC2 instances** are deployed: us-east-1b.
    c. Click **Create Volume**.
    d. Once it's been created, **name the volume** to better identify it in the Volume dashboard: mysqldata.
5. Attach the volume to the EC2 instance, demosys1.
    a. Select the newly created volume: mysqldata.
    b. Select the **EC2 instance, demosys1**.
    c. Click on the **Actions** button at the top.
    d. Click **Attach**.
    e. Note down the **Volume ID** of newly created volume. This volume ID is used when adding this volume to a service group for MySQL data.
6. Discover the volume on the EC2 instance, add it to a disk group, and create a filesystem on it. You can also create the disk group and filesystem on the device using VIOM; however, this step uses the command line.
    a. **Log onto** the EC2 instance as root, **demosys1**, where the volume was attached.
    b. Execute **lsblk** to see if the RHEL operating system recognizes the device.
    c. For Veritas Volume Manager to discover the device, run: **vxdctl -f enable**.
    d. To validate the disk has been discovered, run: vxdisk list.

    NOTE: If the disk is not discovered by Volume Manager, refer to the Troubleshooting section for tips on how to resolve this issue. The name of the disk from **lsblk** will not necessarily match the name of the device shown by Volume Manager.

    e. Initialize the device: vxdisksetup -I <device> vxdisksetup -i de1_xen-vd0_3
    f. Create a disk group: vxdg init <diskgroup name> <name>=<device> vxdg init dg1 dev1=de1_xen-vd0_3
    g. Create a volume of a certain size: vxassist -b -g <diskgroup name> make <volume name> <size> <name of device>

    vxassist -b -g dg1 make vol1 80g dev1

h. Create a vxfs filesystem on the volume: mkfs -t vxfs <raw block device of volume created, in the form /dev/rdsk/<diskgroup>/<volume>

mkfs -t vxfs /dev/vx/rdsk/dg1/vol1

i. Create a directory to mount the filesystem on each EC2 instance, demosys1 and demosys2: mkdir /mydata

Note: The mount will not be done at this point. It will be done by the service groups defined in VIOM described in the next set of steps.

7. Create a service group using VIOM that mounts the EBS volume and can failover/failback between the two EC2 instances, demosys1 and demosys2.

a. Log on to **VIOM** (https://ec2amaz-adjdqf9.ec2.internal:14161/) using a browser from the bastion host as Administrator.

b. On the left pane, click on **Availability**, expand **Uncategorized Clusters**, expand cluster (such as **dcluster**), and expand **Service Groups**.

c. Right-click on **Service Groups** and select **Create Service Group**.

d. Click **next** and enter the fields to create the service group:

i. **Name**: demo1

ii. **Type**: Failover

e. Click **next** and **configure system list**. Select both systems, demosys1 and demosys2, and click the **double arrows** to move the systems to the right-hand side of the window.

f. Click **Next** to configure resources. In this step, you'll create several resources—EBSVol, DiskGroup, and Mount— in this pane.

i. Create EBS Volume Resource. Give it the **Name** EBSVolR, select **Type** EBSVol, and click **Add**.
Click the **3 dots** on the right-hand side to fill in the required fields.

AWSBinDir: /usr/local/bin
VolumeID noted from Step 5e from AWS Management Console: vol-xxxxxxxxx

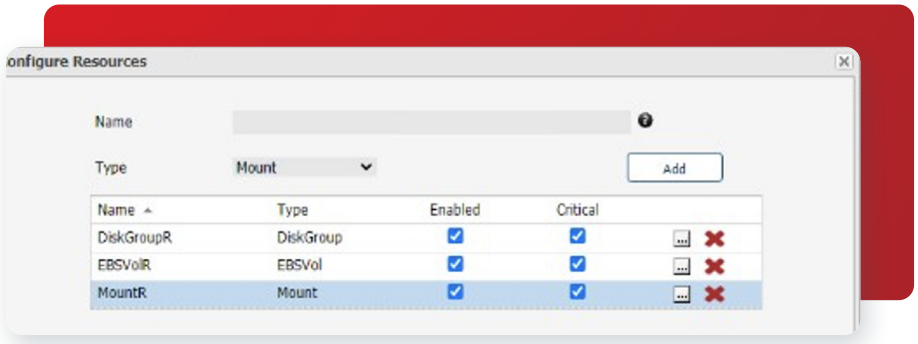
ii. Create the Disk Group Resource. Give it the **Name** DiskGroupR, select **Type** DiskGroup, and click **Add**.
Click the **3 dots** on the right-hand side to fill in the required fields.

DiskGroup: dg1

iii. Create the Mount resource. Give it the **Name** MountR, select **Type** Mount, and click **Add**.
Click the **3 dots** on the right-hand side to fill in the required fields.

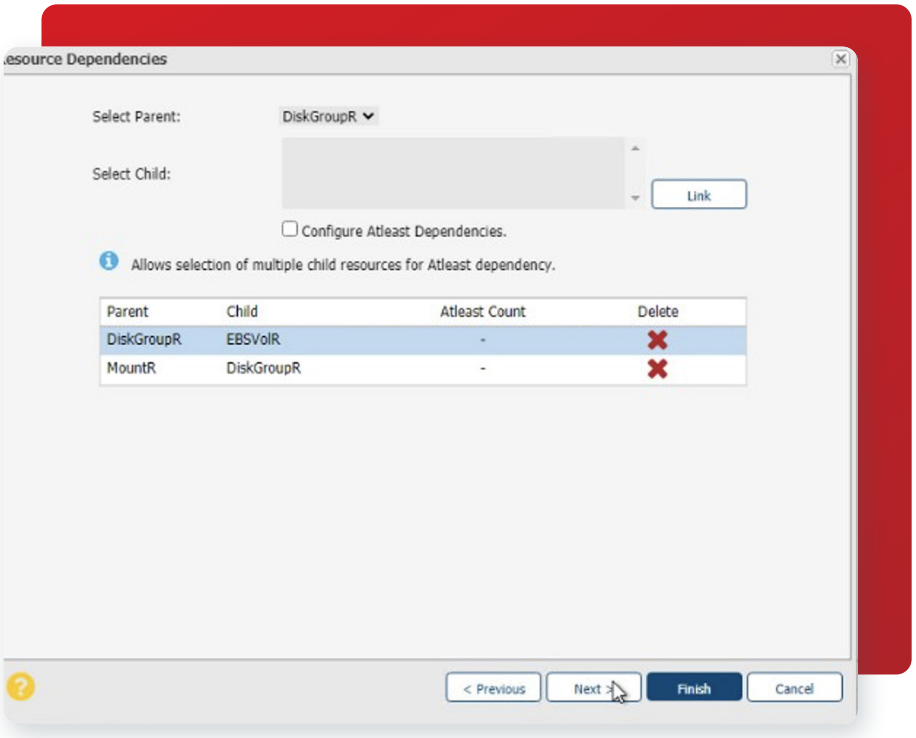
BlockDevice:/dev/vx/dsk/dg1/vol1
FsckOpt: -y
FSType: vxfs
Mountpoint: /mydata

iv. Click on the **Enabled** box for all resources.



v. Click **Next** to link the resources. The linking of resources indicates the order in which the resources will be brought online and offline during failover and failback. Link the resources together where the parent and child relationship are as follows:

MountR is parent to DiskGroupR
DiskGroupR is parent to EBSVol



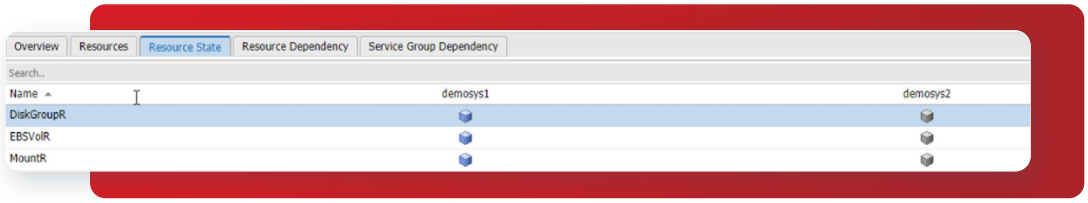
vi. Click **Finish**.

g. Some of the resources maybe enabled but not online—for example, the DiskGroup Resource and Mount Resource.

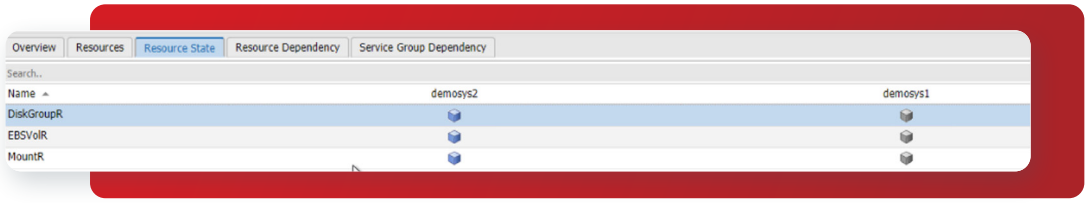
i. For the DiskGroup resource, right-click on **DiskGroupR** (gray box) and select **Online**. Select the system to turn the resource on. In this example, select **demosys1**. Click **Ok**.

ii. The MountR resource needs to be probed prior to turning it online. Right-click on **MountR** (gray box) and select **Probe** for **All systems**. Right-click on **MountR** and select **Online**. Select demosys1 as the system to online the resource.

iii. Validate that all resources are online for the system demosys1. The blue boxes indicates all are online.



h. To validate the demo service group can failover to the other system, demosys2, right-click on the **demo1** service group and select **Switch**. The resources running on demosys1 will go offline and then come online on demosys2.



Step 9: Installation and Configuration of MySQL

This section describes the installation of MySQL.

1. In the last step in the previous section, the EBS volume was mounted on the secondary node, demosys1. Therefore, use VIOM to switch the demo service group back to the primary node, demosys1. The EBS volume /mydata should be mounted on the primary node.

2. Download and install the InfoScale Agents required for MySQL on each EC2 instance, demosys1 and demosys2.

a. Log on via SSH to the EC2 instance primary node, **demosys1**.

b. Download the **Acc Library and MySQL Agents** for InfoScale for the version of the operating system (for example, RHEL 7) and architecture (such as x86-64) from https://sort.veritas.com/agents and place on each of the EC2 instances demosys1 and demosys2. In this example, we have downloaded agents for RHEL 7. Download the documentation as needed for reference.

c. Extract and install the Acc Library and MySQL Agents on the primary node, demosys1.

```
tar xvfz ../HA_ag_acc_library_7.0.3.0_InfoScale7.4.3_RedHatEnterpriseLinux_x86-64.tar.gz
rpm -ivh VRTSacclib-7.0.3.0-GENERIC.noarch.rpm

tar xvfz ../HA_ag_mysql_agent_7.0.5.0_InfoScale7.4.3_RedHatEnterpriseLinux_x86-64.tar.gz
rpm -ivh VRTSmysql-7.0.5.0-GENERIC.noarch.rpm
```

d. **Repeat steps a-c** for the secondary node, demosys2.

e. Import the MySQL types so the MySQL type definition is available for use by the cluster. Run the following command from the **primary node demosys1**:

/etc/VRTSagents/ha/conf/MySQL/MySQLTypes.cmd

3. Download and install the MySQL bundle on each EC2 instance (demosys1 and demosys2). In this example, download the RHEL 7 RPM bundle (tar file) for x86, 64-bit from https://dev.mysql.com/downloads/mysql. Refer to the appropriate installation instructions for more detail: https://dev.mysql.com/doc/refman/8.0/en/linux- installation-rpm.html.

a. Extract and install MySQL:

tar xvf ../mysql-8.0.23-1.el7.x86_64.rpm-bundle.tar
yum install mysql-community-{server,client,common,libs,test,devel,embedded}-*

b. Start mysqld and validate the status. This action will create the appropriate users, passwords, and initiate a simple database in /var/lib/mysql:

systemctl start mysqld systemctl status mysqld

c. Change the password for the root user within mysql:

sudo grep 'temporary password' /var/log/mysqld.log
mysql -u root -p <hit enter> and then enter <temporary password> (Note: The password does not show as you type it.)
mysql> alter user 'root'@'localhost' identified by '<new password>'

d. **Repeat steps a-c** on the other node, demosys2.

4. After install, change the directory of the default database location, Datadir of mysql, on the primary node (demosys1). At this point, the device being managed by InfoScale that was configured in the previous section, Step 5: Disk Configuration, should be in a mounted (/mydata) state on the primary node, demosys1.

a. Stop the mysql service: **service mysqld stop**

b. Copy the data from /var/lib/mysql to directory mydata (the mounted device managed by InfoScale):

cp -rap /var/lib/mysql/* /mydata

c. Change ownership to mysql user: **chown -R mysql:mysql /mydata**.

d. Modify the contents of **/etc/my.cnf** to recognize new location of Datadir. In this example, the new directory is /mydata.

In the [mysqld] section, modify datadir and the socket to the new location.
datadir=/mydata
socket=/mydata/mysql.sock

Add the following lines:
[client] port=3306
socket=/mydata/mysql.sock

e. **Repeat steps** c-d on secondary node, demosys2.

5. Restart the mysqld service on the primary node, demosys1, to validate that the new directory is set.

```
service mysqld start
mysql -u root -p -e "SELECT @@datadir"
```

Note: If you encounter any errors, refer to /var/log/mysqld.log for more information on the errors.

6. Stop the mysqld service on both nodes, demosys1 and demosys2. The next steps describe how to start and manage mysqld from InfoScale.

```
service mysqld stop
```

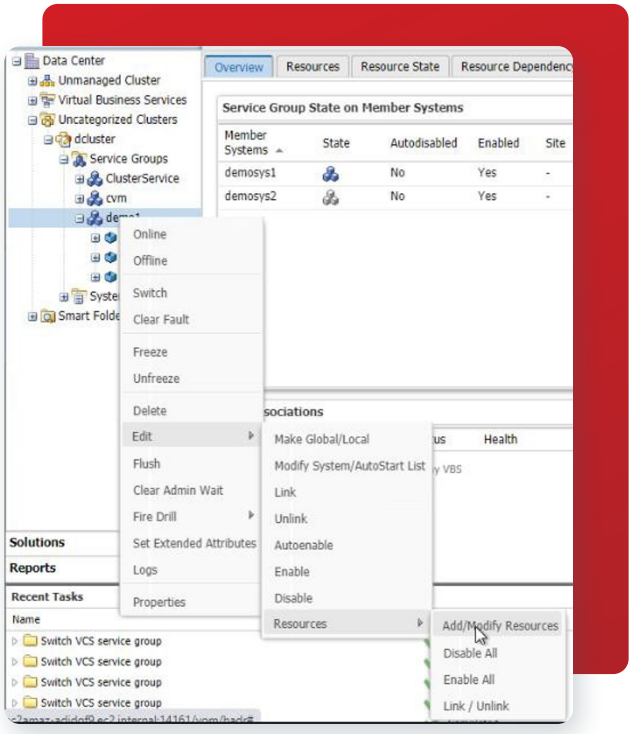
Step 10: Adding InfoScale Resources for MySQL in a Service Group

You use VIOM to configure service groups for MySQL. We created a demo1 service group in the previous steps and in this section, we will add additional resource groups to it. These new resources are linked to the previously created resources to provide the order in which the resources are brought online and offline during failover and failback operations between nodes in the cluster.

1. Log on to **VIOM** (https://ec2amaz-adjdqf9.ec2.internal:14161/) using a browser from the bastion host as Administrator.

2. On the left pane, click on **Availability**, expand **Uncategorized Clusters**, expand the cluster (that is, **dcluster**), expand **Service Groups**, and expand **demo1** to add resources.

3. Right-click on the demo1 service Group and select **Edit->Resources->Add/Modify Resources**.



4. Add the additional resources for MySQL as follows.

   a. Create the AWSIP Resource: give it the **Name** AWSIPR, select **Type** AWSIP, and click **Add**. Click the **3 dots** on the right-hand side to fill in the following fields:

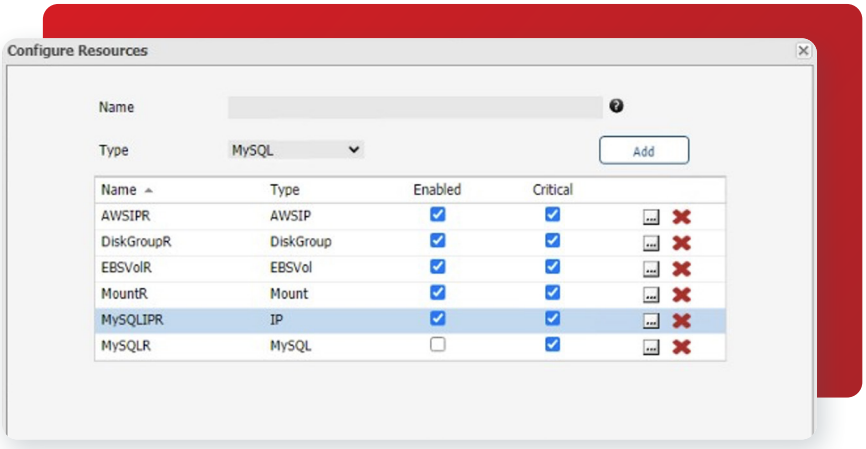
   AWSBinDir: /usr/local/bin
   Private IP: 10.31.1.112 (Note: Use an un-used IP. This IP is the IP for the application.)

b. Create the IP Resource: give it the **Name** MySQLIPR, select **Type** IP, and click **Add**. Click the **3 dots** on the right-hand side and fill in the following fields:

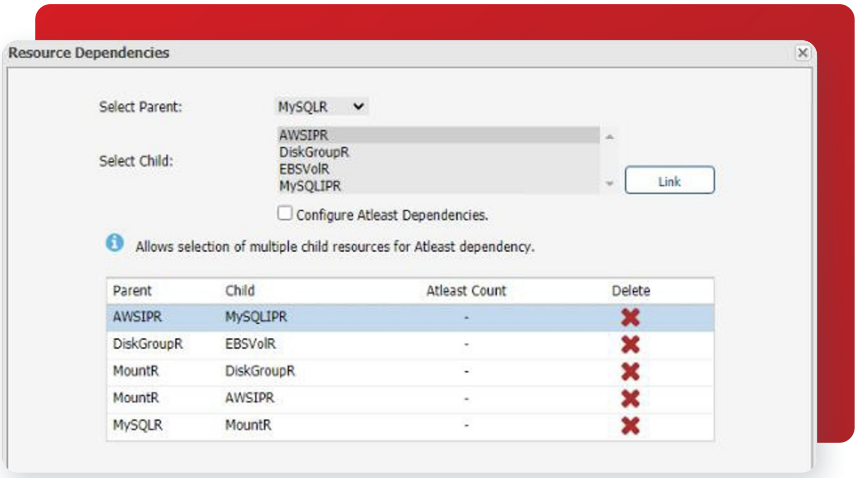
Address: 10.31.1.112 (the same IP address used in Step 4a)
Device: eth0 (the network interface used for application traffic)
NetMask: 255.255.255.0

c. Create the MySQL Resource. Give it the **Name** MySQLR, select **Type** MySQL, and click **Add**. Click the **3 dots** on the right-hand side and fill in the following fields:

BaseDir: /usr
DataDir: /mydata
Hostname: Select Apply To: **Selected Systems** and Enter **Attribute** values for each **hostname**: demosys1 for demosys1 system and demosys2 for demosys2 system.
MySQLAdmin: root
MySQLAdminPasswd: P@ssw0rd
MySQLUser: mysql
Port:3306
UseSystemID:1



d. Place a **check mark** on **Enabled** for the AWSIPR and MySQLIPR resources and click **Next**. The MySQLR resource will be enabled after creating a Cluster AWS IP under cluster services, as described in the next step.

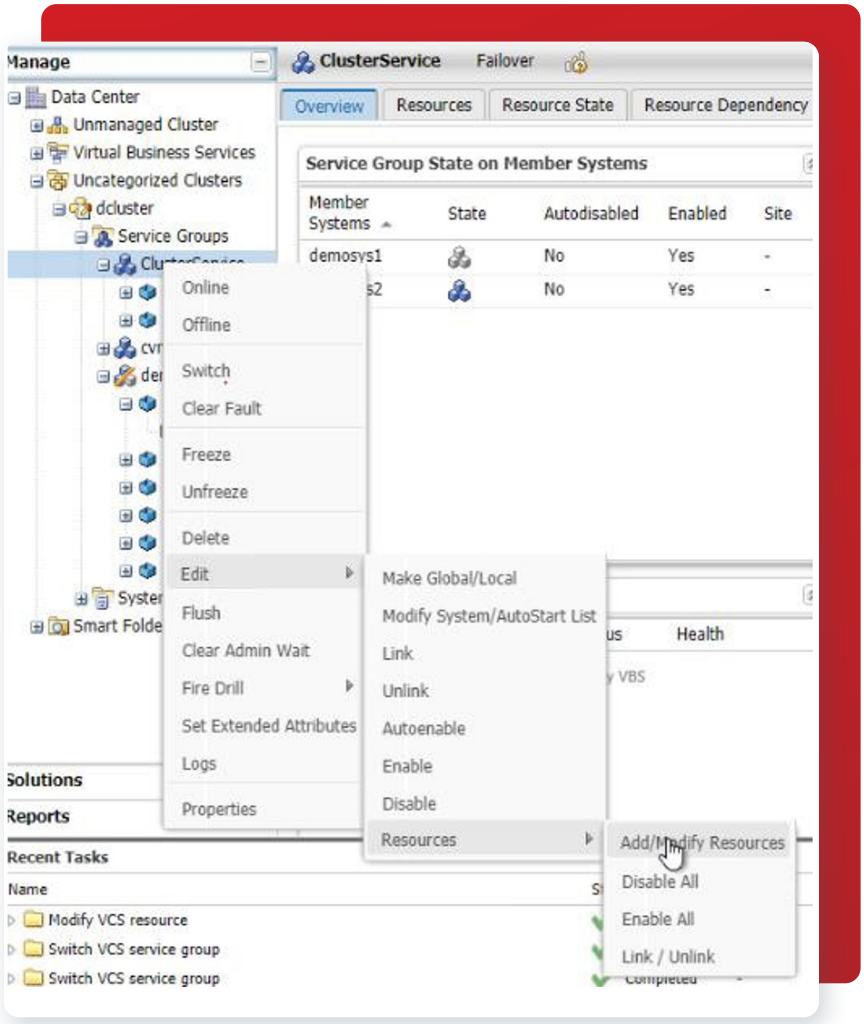


e. Link the resources together where the parent and child relationship are as follows and click **Finish**:

MountR is parent to AWSIPR
AWSIPR is parent to
MySQLIPR MySQLR is parent to MountR

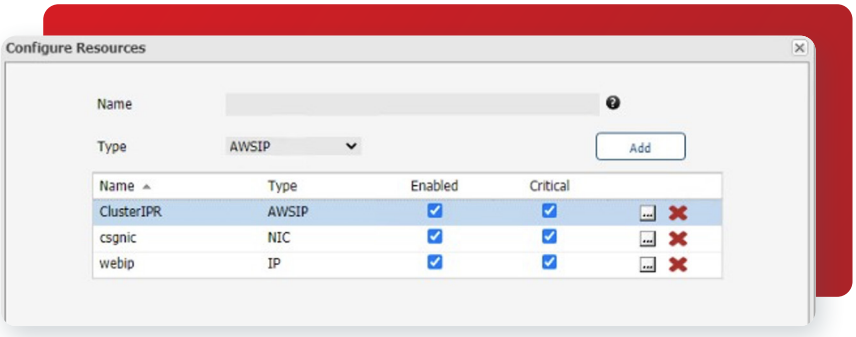
5. Before turning the MySQL resource online, add the **AWSIP Resource** in the ClusterService service group.

a. Right-click on the ClusterService service group and select **Edit->Resources->Add/Modify Resources**.



b. Create the AWSIP Resource: give it the **Name** ClusterIPR, select **Type** AWSIP, and click **Add**. Click the 3 dots on the right-hand side to fill in the following fields:

AWSBinDir: /usr/local/bin
Private IP: 10.31.1.111 (Note: Use the Cluster IP.)



c. Place a **check mark** in the **Enabled** box, click **Next**, and click **Finish**. This resource does not need to be linked to the other resources within the ClusterService group.

d. Expand **AWSIP**, right-click on the **ClusterIPR** resource (gray box), and select **Online** to online the resource.

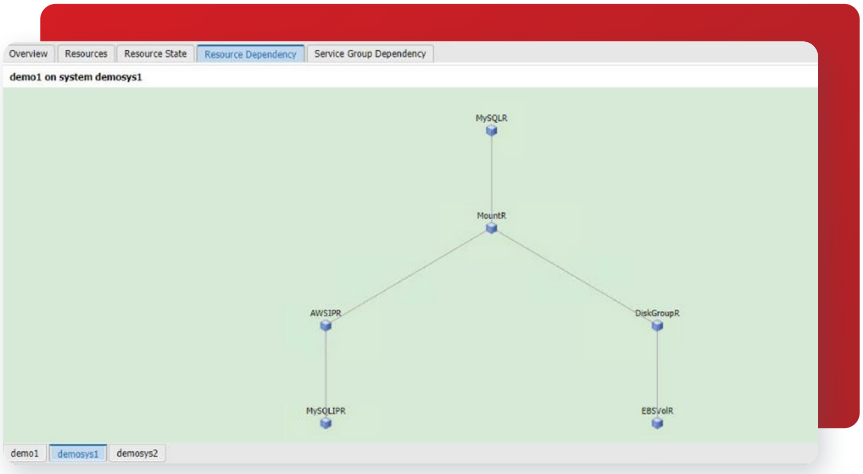
6. Enable and online the MySQLR resource:

a. Select the **demo1** service group and click **expand**.

b. Right-click on the **MySQLR** resource and select **Enable**. Click **Ok** to enable.

c. Right-click on the **MySQLR** resource and select **Online**. Click **Ok** to online the resource on demosys1.

7. Validate that all resources under demo1 are online. The blue boxes indicate all are online. Validate that the resource dependencies are correctly linked. The following view is displayed under the **Resource Dependency** tab of the demo1 service group. From demosys1, the services are online.



8. Validate that the demo1 service group can failover from demosys1 to demosys2.

a. Right-click on the **demo1** service group, select **Switch** to system demosys2, and click Ok.

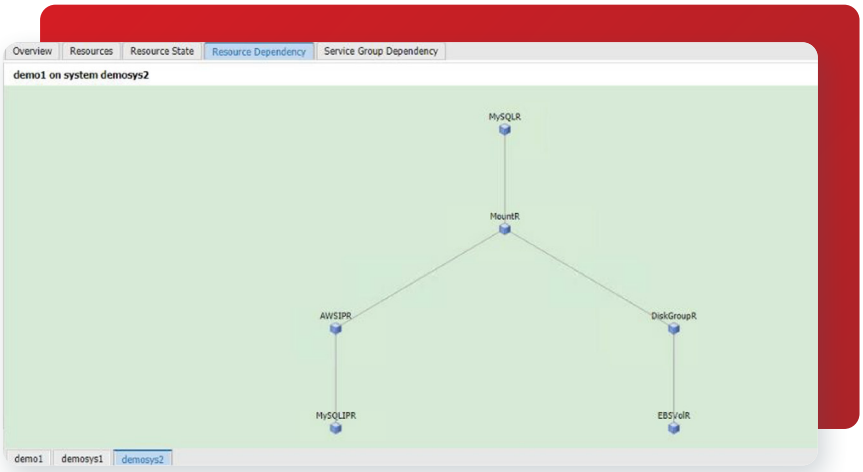
b. Validate the **demo1** service group has switched from demosys1 to demosys2. The **Resource Dependency** tab view shows that the resources are online on demosys2.

9. Another way to validate that the services are running on demosys2 is to do the following:

a. Log on to demosys2.

b. Validate the /mydata directory is mounted: **df -h**.

c. Check the status of MySql and validate it is active: **service mysqld status**.



Troubleshooting Tips

This section discusses some common issues you may encounter during configuration of InfoScale Enterprise on AWS.

Networking Issues

1. Unable to ping private IP addresses of hosts.

a. Check the Security Groups of the EC2 instances in the AWS management console. Make sure the required type of traffic is allowed.

b. Check the Routes on the VPC on the AWS management console.

c. Check the routes and rules at the operating system level in /etc/sysconfig/network-scripts.

2. Unable to ping the Internet from within EC2 instances.

   a. Double-check that the /etc/resolv.conf has the correct namesearch and DNS.

VIOM Issues

1. From VIOM, there is a question mark over the resources during the Online of resource group within the service group.

   a. Double-check the values within the Attributes.

2. Unable to add hosts.

   a. Double-check that you enabled the Control Hosts Add-ons under Settings.

   b. Double-check that the firewall is off on the bastion host.

   c. Check that the hostnames of the bastion host and the EC2 instances are resolvable (/etc/hosts file).

   d. Check that the security groups on the bastion hosts and the EC2 instances allows traffic to each other.

Volume Manager Issues

1. When a service group using an EBS volume is off-lined and that service group has been removed, exclusion of the device may not be cleared. Therefore, adding the EBS volume using the same device name (that is, /dev/sdf) from the AWS management console will not be discovered by the Volume Manager using the vxdctl -f enable or vxdisk scandisks command because the volume would be suppressed.

   a. There are several ways to resolve this issue:

      - Use vxdiskadm to unsuppress the device.
      - From the AWS management console, rename the device, for example /dev/sdg.
      - Use "vxdmpadm include path=<device name>" to include the device path again.

References

- AWS services
  - Amazon EC2
  - Amazon Virtual Private Cloud (Amazon VPC)
  - Amazon Route 53
  - Amazon Elastic Block Store (EBS)
- Veritas InfoScale
  - https://sort.veritas.com/documents

About Veritas

Veritas Technologies is a global leader in data protection and availability. Over 80,000 customers—including 87 percent of the Fortune Global 500—rely on us to abstract IT complexity and simplify data management. The Veritas Enterprise Data Services Platform automates the protection and orchestrates the recovery of data everywhere it lives, ensures 24/7 availability of business-critical applications, and provides enterprises with the insights they need to comply with evolving data regulations. With a reputation for reliability at scale and a deployment model to fit any need, Veritas Enterprise Data Services Platform supports more than 800 different data sources, over 100 different operating systems, more than 1,400 storage targets, and more than 60 different cloud platforms. Learn more at www.veritas.com. Follow us on Twitter at @veritastechllc.

2625 Augustine Drive
Santa Clara, CA 95054
+1 (866) 837 4827
veritas.com

For global contact information visit:
veritas.com/company/contact

V1532 03/22