VERITAS

Workload Migration and Auto Recovery to AWS Outposts Using NetBackup Resiliency Platform

Deployment Guide | July 2022

Contents

Revision History	
Rev 1.0 December 2021	Initial version

Introduction

NetBackup Resiliency Platform allows for automation and orchestration of system migration and disaster recovery to an AWS Outposts edge device and/or direct to the AWS public cloud. Organizations that are not yet prepared to migrate all their assets and environments to a public cloud or would like to keep their digital assets on-premises but desire the features and services a public cloud offers can deploy an edge device such as AWS Outposts. Using NetBackup Resiliency Platform within AWS Outposts assists in migrating legacy environments into a modern infrastructure.

Scope

The purpose of this document is to provide step-by-step instructions on how to deploy NetBackup Resiliency Platform on AWS Outposts for workload migration and auto recovery. This is a sample deployment and readers should refer to the Veritas and AWS Outposts product documentation for definitive and full installation, administration, and configuration details. There are references to these documents in this guide. This guide assumes AWS Outposts is already configured and installed in the organization's data center and is accessible via the AWS management console. This document is targeted for customers, partners, and Veritas field personnel interested in understanding how to deploy NetBackup Resiliency Platform on AWS Outposts.

Solution Overview

AWS Outposts provides an on-premises platform as a service (PaaS). You get compute (EC2), networking (10Ge interconnects), storage (SSD solid state drives), and an SSD-based instance store. To automate and orchestrate the migration and recovery of a workload like VMware, you need to deploy NetBackup Resiliency components on both the data center and on AWS Outposts. As shown in Figure 1, NetBackup Resiliency Platform is responsible for the discovery of servers on-premises, a backup of the VMware workload, and storage of the images in an AWS S3 bucket residing in the same region as AWS Outposts. The organization can then conduct a test rehearsal and/or recovery of the workload using NetBackup Resiliency Platform on AWS Outposts with the images that were stored in the AWS S3 bucket. The components implemented in this deployment include:

- **Resiliency Platform Resiliency Manager (RM)**—Provides the services necessary to protect the assets, the management console, and the data repository.
- **Resiliency Platform Infrastructure Management Server (IMS)**—Discovers, monitors, and manages the assets.
- **Primary Server**—Manages and controls the backup and recovery activities and hosts the catalog that contains policies and schedules, metadata about the backup jobs, and media, device, and image metadata information.
- **Media Server**—Writes client data as backup images to varying types of storage such as local disks, tape, network-attached storage (NAS), storage array network (SAN), or the cloud and later restores the data to the client.

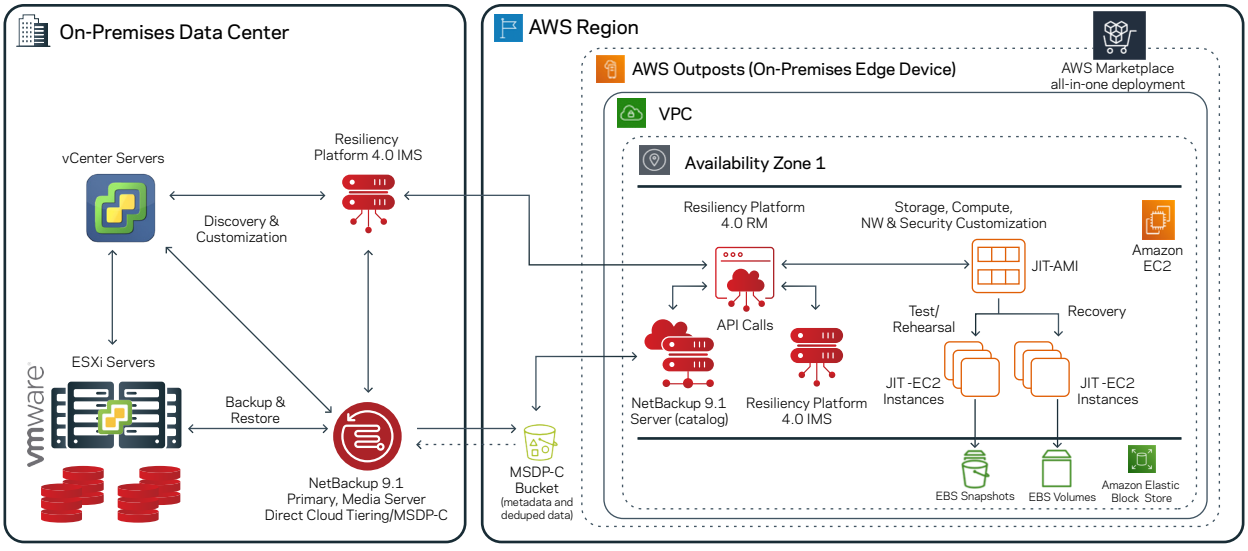


Figure 1. NetBackup Resiliency Platform on-premises and on AWS Outposts.

Resources

Software and Compute

Table 1 shows the software and compute resources used in this guide. The guide assumes AWS Outposts is installed, networked, and accessible in the data center with user accounts already configured. Depending on compute and storage resources ordered for AWS Outposts, the instance type available can vary. When selecting an EC2 instance for deploying NetBackup Resiliency Platform components, it should have enough CPU and memory to meet the application requirements.

Component	Resource
AWS Outposts	VPC with two subnets, route tables that route to the Internet and local gateway, and Elastic IPs
	1 x EC2 Instance type for RM v4.0 (m5.2xlarge)
	1 x EC2 Instance type for IMS v4.0 (m5.2xlarge)
	1 x EC2 Instance type for NetBackup 9.1 (acting both as primary and media server) for NetBackup Image Sharing (m5.2xlarge)
	Several EC2 instances for recovery of VM workloads (m5.xlarge)
On Premises Resources	2 x VM instances that have NetBackup 9.1 (primary and media servers). Primary: 8 CPUs and 64 GB, with 100 GB for system disk and 500 GB for MSDP. Media server: 8 CPUs and 64 GB, with 100 GB for system disk and 1 TB for MSDP.
	1 x VM instance that has IMS v4.0 (8 CPUs, 64 GB of memory, 30 GB and 60 GB hard disk)
	VM (VCenter) Server, ESXi servers
Utilities	Google Chrome
	MobaXterm
	WinSCP

Table 1. Resource Summary

Network Topology

AWS Outposts was connected to the on-premises data center corporate network with access to the public Internet and machines within the data center (see Figure 2). A CIDR range of customer-owned IP addresses is provided by the on-premises IT team so machines within the data center can access the EC2 instances created on AWS Outposts. AWS Outposts sends hardware machine state, status, and configuration information for monitoring by AWS support personnel. Only specific ports on specific machines were opened on the firewall. To protect the machines on the corporate network, firewall rules were configured to NOT allow any requests or traffic to be initiated by instances within AWS Outposts.

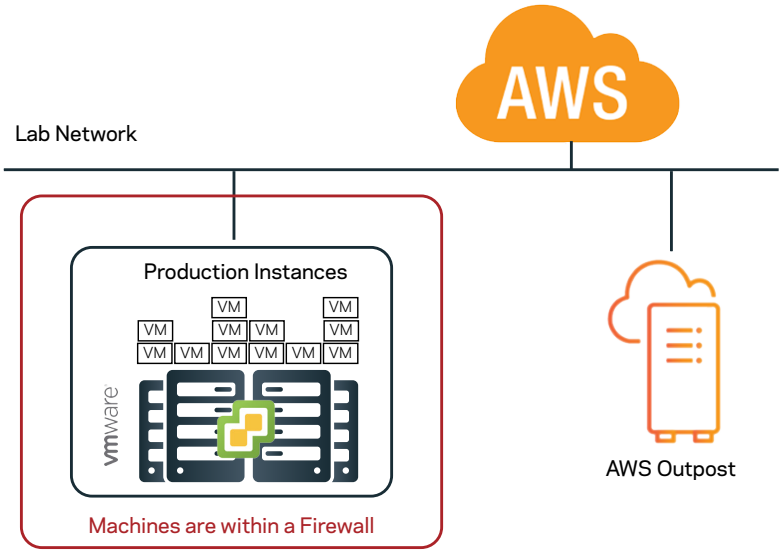


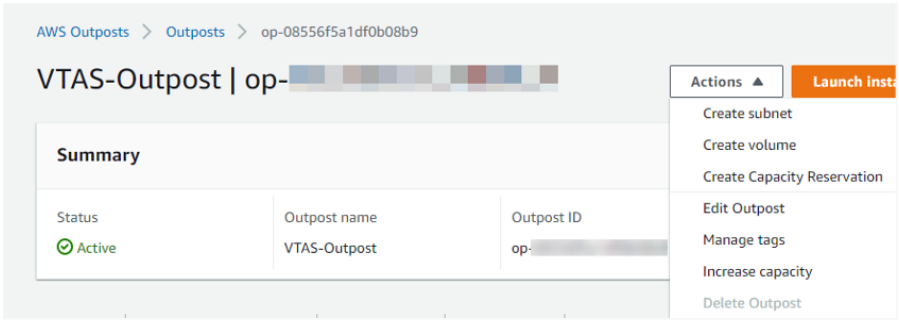
Figure 2. The network topology of AWS Outposts connected to a corporate on-premises data center and the Internet.

Step 1: Amazon VPC Configuration

The Amazon Virtual Private Cloud (VPC) defines the virtual network where the AWS resources (EC2 instances, bastion hosts) are launched. AWS Outposts has a defined VPC with a default route already created once deployed in your data center. In this example, an additional two subnets are created within AWS Outposts and Elastic IPs are also created to attach to the primary interfaces of the EC2 instances for Internet access. (NOTE: Only one subnet is needed; however, the NetBackup Resiliency Platform CloudFormation templates (CFTs) ask for an additional subnet.)

Create Subnets on AWS Outposts

1. Log on to the AWS management console using your credentials.
2. From the AWS management console, select **AWS Outposts**.
3. Select **AWS Outposts already present** (for example, op-xxxxx).
4. Select **Actions** on top and select **Create subnet**.

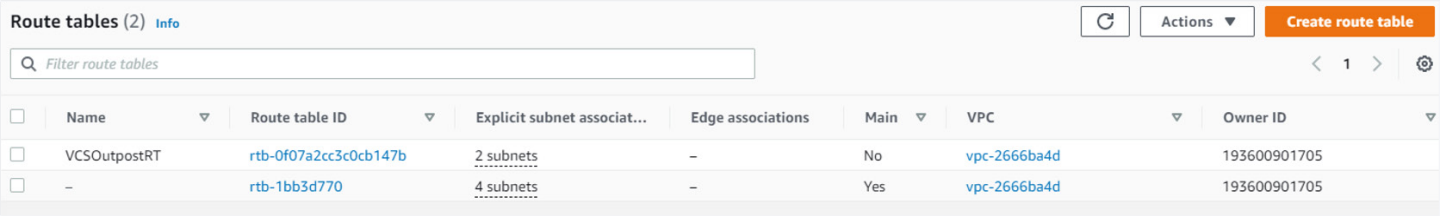


5. In the next screen, select the **VPC ID** (vpc-xxxxx) that was deployed on AWS Outposts. Provide subnet names and an IPv4 CIDR block within the associated VPC IPv4 CIDRs. In this example, we created two subnets as follows:
   a. **Subnet name**: psn1
   b. **IPv4 CIDR block**: 172.31.51.0/24
   c. Click **Add new subnet**.
   d. **Subnet name**: psn2
   e. **IPv4 CIDR block**: 172.31.52.0/24
   f. Click **Add new subnet**.

☐	psn3	subnet-0d4ab0d2c87e386b2	Available	vpc-2666ba4d	172.31.53.0/24	–
☐	psn2	subnet-034701668e5bcd7c1	Available	vpc-2666ba4d	172.31.52.0/24	–

Edit Subnet Associations

6. After you have created the subnets as described above, they are placed by default on the main route table. However, the subnet associated with the primary interface will be associated with the customized route that connects to the Internet gateway and local gateway. As previously mentioned, the VPC should have two routes, one default main and one customized to have a route to the Internet gateway and local gateway. In this example, we do so by adding psn1 to the custom route table named VCSOutpostRT.



   a. Select from Services: VPC.
   b. On the left pane, select Route Table.
   c. Select the customized route table (that is, VCSOutpostRT).
   d. Click on the Subnet associations tab and click on Edit subnet associations.
   e. Select psn1 to be associated with the customized route table. The network interfaces associated with this subnet will have access to the Internet and local gateway.

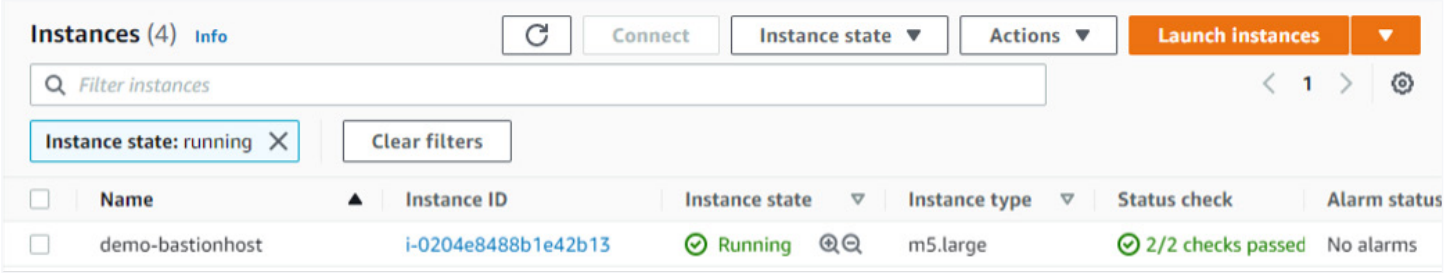
Step 2: Bastion Host Creation and Configuration

To access the EC2 instances created in the private subnet externally from a local computer within your network, you need to create and deploy a bastion host. In this step, you will create an EC2 instance of type and Windows 2019 server to be placed on a public subnet. An external IP will be assigned to be able to access the Internet.

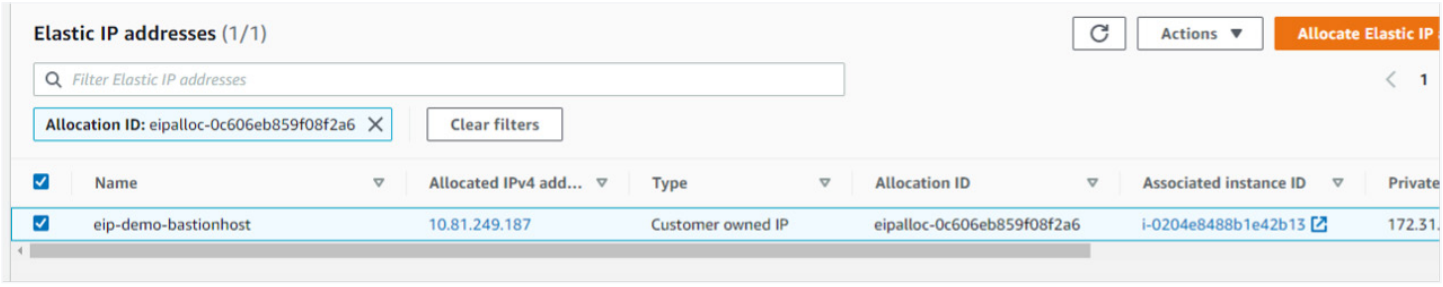
1. Log on to the AWS management console.
2. Select AWS service **EC2**.
3. From the EC2 dashboard, click on **Launch instance**.
4. Enter in the search field **windows**. Click on the left pane, "**AWS Marketplace**," and **select** Microsoft Windows Server 2019 Base.
5. Choose **an Instance Type** size. In this example, we have selected **m5.large** (2 vCPUs and 8 GB of memory).
6. Click **Next Configure Instance Details**. Fill in the instance details as follows:
   a. **AWS Outposts VPC Network**: vpc-xxx
   b. **Subnet**: psn1
   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 the details as follows:
   a. **Assign a security group**: Create a new security group.
   b. **Security Group name**: test-sg
   c. Remove SSH and RDP type and add a rule with All traffic type and specify the IPv4 block that will be connecting to the bastion host and the IPv4 block of the VPC.

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 your 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. To connect to the EC2 instance from your local computer, you need to create an Elastic IP address from a customer pool of IPv4 addresses and associate it with the instance.
   a. On the left pane, select **Elastic IPs**.
   b. Click on **Allocate new address**.
   c. Select **Customer owned pool of IPv4 addresses**. (This pool is pre-assigned by on-premises data center network personnel during configuration of AWS Outposts.)
   d. Click **Allocate**.
   e. Select the newly created Elastic IP address, then click on the **Actions** button and **Associate Elastic IP** address to the bastion host instance and the private IP address of the bastion host.



16. Connect to the bastion host instance.
    a. Click on **Instances** on the left side and select the bastion host (for example, demo-bastionhost).
    b. Click the **Connect** button.
    c. Select the **RDP client tab**. Click on **Get password**. Click Browse and specify the location of the key pair, dpair.pem, you downloaded in Step 11. Click **Decrypt password**. (Note: It make take 4 minutes after launch before you can do a **Get password**.)
17. Save and copy the **Password**. You will use the Password to log on to the bastion host from your local computer.
18. On your local computer, start Remote Desktop and enter the Elastic IP created in step 15e. Click on **connect** and enter the **password** copied in Step 17.
19. 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**.
20. Start the browser (IE), download the utilities listed in Table 2, and install them accordingly.

Utility	Purpose	Download Site
Google Chrome	Browser	https://www.google.com/chrome/
MobaXterm	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

Table 2. Utilities

Step 3: RM and IMS Installation and Deployment

The AWS Marketplace has cloud formation templates available to deploy the RM and IMS components of NetBackup Resiliency Platform. This section describes the deployment and configuration of these components on AWS Outposts using these templates.

On AWS Outposts RM and IMS

Elastic IP Assignments

Prior to launching the templates, you need to allocate customer-owned elastic IPs as well as IPs from the Amazon pool. These IPs are associated with the primary interfaces of the RM and IMS EC2 instances. The customer-owned IPs are used for connection and access to EC2 instance from machines in the on-premises data center and the public IPs from the Amazon pool are for external access to the Internet. Allocate the Elastic IPs by following these steps:

1. Log on to the AWS management console.
2. Select the AWS service **EC2**.
3. To connect to the EC2 instance from your local computer, you need to create an Elastic IP address from a customer pool of IPv4 addresses and associate it with the instance. Customer-owned Elastic IPs need to be created for RM, IMS, and NetBackup EC2 instances.
    a. On the left pane, select **Elastic IPs**.
    b. Click on **Allocate new address**.
    c. Select **Customer owned pool of IPv4 addresses**. (This pool is pre-assigned by on-premises data center network personnel during configuration of AWS Outposts.)
    d. Click **Allocate**.
    e. Repeat steps 4b–4d two more times to allocate Elastic IPs for the other EC2 instances.

4. For the EC2 instance to access the Internet, you need to allocate a public IP from the Amazon pool of resources. Thus similarly, create three public Elastic IPs for RM, IMS, and NetBackup.
   a. On the left pane, select Elastic IPs.
   b. Click on Allocate new address.
   c. Select Amazon pool of IPv4 addresses.
   d. Click Allocate.
   e. Repeat steps 5b–5d two more times to create public Elastic IPs.

RM Deployment

1. From the AWS management console, go to the AWS Marketplace.
2. Select **Discover Products** on the left pane and enter in the Search Field: Veritas Resiliency Platform.
3. Click on Veritas Resiliency Platform and subscribe.
4. Once the subscription has been approved, click on **Continue to Configuration** or go to "Manage subscriptions" from the AWS Marketplace, select Veritas Resiliency Platform, and under **Actions** launch the template.
5. Select the CloudFormation template for **Veritas Resiliency Platform Resiliency Manager Install**.
   a. Select software version 4.0.0.0
   b. Select the **Region** where AWS Outposts is deployed: US East Ohio
   c. Click on **Continue to launch**.
   d. Click on **Launch**.
6. On AWS Outposts, the deployment of the template requires a lot more time than the default of 20 minutes (1200 seconds). Therefore, we recommend increasing the BootStrapWaitCondition timeout variable to 35 minutes (2100 seconds).
   a. Click **View in Designer**.
   b. In the Design, click on the **BootStrapWaitConditionHandle**  icon.
   c. In the bottom pane is a YAML view of the template. Scroll down to Timeout in the BootStrapWaitCondition and modify the Timeout from 1200 to 2100 seconds.

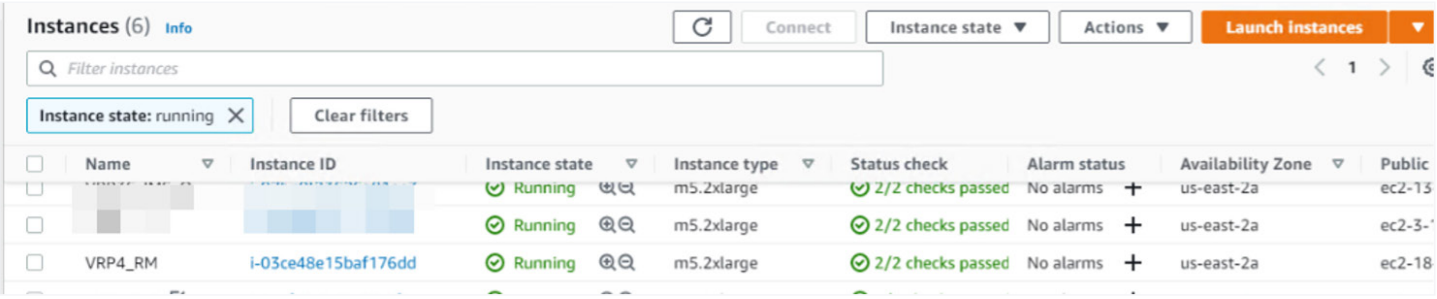
```
1320   BootstrapWaitConditionHandle:
1321     Type: 'AWS::CloudFormation::WaitConditionHandle'
1322     Metadata:
1323       'AWS::CloudFormation::Designer':
1324         id: fc4fe4ca-dc9d-4d47-a001-d9ee879b6d16
1325   BootstrapWaitCondition:
1326     Type: 'AWS::CloudFormation::WaitCondition'
1327     DependsOn: EC2Instance
1328     Properties:
1329       Count: 1
1330       Handle: !Ref BootstrapWaitConditionHandle
1331       Timeout: 2100
```

   d. On the top pane, click on the **check mark** to validate the template  .
   e. On the top pane, click on the **cloud icon** to create the stack template  .
   f. Click **Next**.
7. Provide the required stack parameters and use the default values for the rest of the parameters not specified below:
   a. **Stack Name**: VRP4_RM
   b. **Instance Name**: VRP4_RM
   c. **EC2 Instance Type** (may vary depending on instances available on AWS Outposts): m5.2xlarge
   d. **Key Pair for SSH access**: demokp1 (NOTE: This should be an existing key pair.)
   e. **Network Configuration** VPC ID of AWS Outposts: vpc-2666ba4d (172.31.0.0/16)
   f. **Eth0 subnet** (specify the subnet created in section AWS VPC Configuration): psn1
   g. **Eth1 subnet** (specify the subnet created in section AWS VPC Configuration): psn2 (**NOTE**: This eth1 will not be used in this deployment; however, the template expects a valid subnet here because otherwise stack creation fails.)
   h. Is the **Eth0 Network Interface behind NAT**: Yes (**NOTE**: Access to this instance from the data center is through the customer owned IP.)

i. **Eth0 NAT Hostname**: ip-10-xx-xx-xx.us-east-2.compute.internal (**NOTE**: Usually in the form ip-x-xx-xx.<region>.compute.internal.)

j. **Eth0 NAT IP**: 10.xx.xx.xx

8. Click **Next**. In the section Stack failure options, select, "**Preserve successfully provision resources**" to preserve the resources that have been successfully created instead of deleting them in case of stack failure creation.
9. Click **Next**. Review the parameters entered and place a **check mark** on "I acknowledge that AWS CloudFormation might create IAM resources."
10. Create **Stack**.
11. Watch the **Events** tab in the CloudFormation Stacks page. After the **EC2 instance creation has completed** and the **BootstrapWaitCondition** has initiated, associate the public and customer-owned Elastic IPs to the primary interface (eth0) of the RM EC2 instance.
    a. Go to **Services** and select EC2.
    b. On the left pane under **Network and Security**, click on **Elastic IPs**.
    c. Select one of the **public Elastic IPs** created in the Elastic IPs creation section and click on Actions.
    d. Select Associate Elastic IP address.
    e. Select the instance to associate the Elastic IP: VRP4_RM
    f. Choose the **public IP** to associate with the Elastic IP.
    g. Click **Associate**.
    h. Repeat steps d to g; however, select one of the **customer-owned IPs**.
12. Wait for the RM stack creation to complete.
13. Go to the EC2 dashboard page and confirm that VRP4_RM was created.



14. Modify the **security groups** of VRP_RM so SSH is allowed in the inbound rules for the machines in the on-premises data center.
    a. Go to the EC2 page and select the EC2 instance for the RM you created: VRP4_RM
    b. In the bottom pane, select the **Security** tab and then select the security group "*RMRMSecurityGroup*."
    c. Click on **Edit Inbound Rules**.
    d. **Add rule SSH** as type and enter the IP range of systems from the data center accessing this EC2 instance: 10.xx.xx.0/24
    e. **Add rule SSH** as type and enter the IP range of the private subnet on AWS Outposts: 172.31.0.0/16
    f. Click **Save rules**.
15. SSH to VRP4_RM from one of the systems in the data center and enter the new password as prompted and re-login with the newly entered password.
16. Validate that all services are running as normal by entering on prompt: **manage -> services status ALL**. All services should be running except for the database service, which is not yet configured.

17. Download and install the necessary NetBackup Resiliency Platform hotfixes from https://www.veritas.com/content/support/en_US/downloads. Follow the instructions in the README file of the download. In this deployment, we had to install hotfix 4.0.0.2, for example, so we executed the following steps from the RM CLI:
    a. utilities> **sftp-session start put patch**
    b. Use WinSCP to copy from Windows to the RM the hotfixes
    c. hotfix> **apply-hotfix 4.0.0.2**
    d. utilities> troubleshoot run-tool fix-root-expired -a
18. In this example, we were not using a DNS and therefore had to modify the /etc/hosts file on the RM. The RM hostname need to be resolvable from systems on-premises. From the RM CLI, get into the shell as follows:
    a. support -> shell
    b. Enter the initial password for the RM (**NOTE**: You need to get this password from Veritas Support).
    c. Modify the /etc/hosts files to include the hostnames and IPs of all on-premises systems such as the VCenter, NetBackup (primary, media), and the IMS that is required.

IMS Deployment

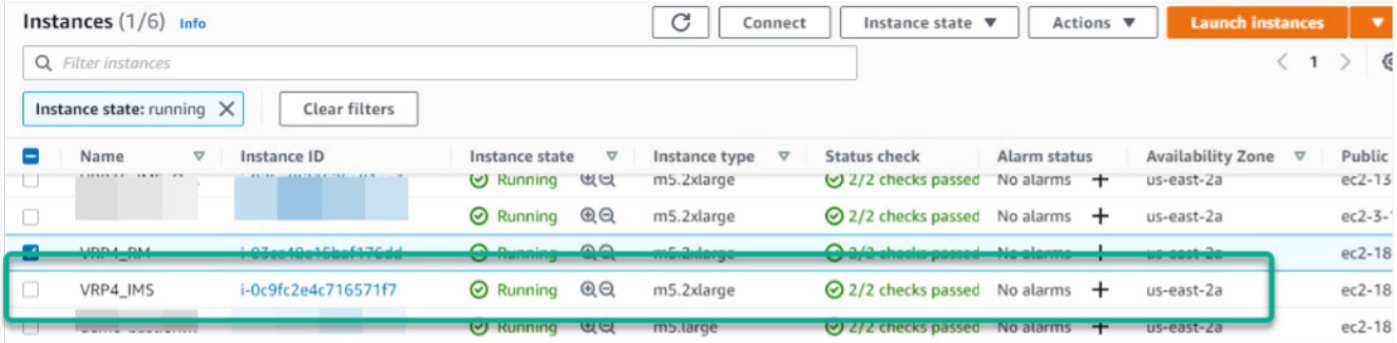
Subscription to the NetBackup Resiliency Platform CFTs is needed prior to deployment of the IMS. The steps to deploy the RM are similar to the IMS deployment steps.

1. Select the CloudFormation template for **Veritas Resiliency Platform Infrastructure Management Server Install**:
    a. Select software version **4.0.0.0**.
    b. Select the **Region** where AWS Outposts is deployed: US East Ohio
    c. Click on **Continue to Launch**.
    d. Click **Launch**.
2. On AWS Outposts, the deployment of the template requires a lot more time than the default of 20 minutes (1200 seconds). Therefore, we recommend increasing the **BootStrapWaitCondition** timeout variable to 35 minutes (2100 seconds).
    a. Click View in Designer.
    b. In the Design, Click on the BootStrapWaitConditionHandle icon. 
    c. In the bottom pane, there is a YAML view of the template. Scroll down to Timeout in the BootStrapWaitCondition section and modify the Timeout from 1200 to 2100 seconds.

```
1320   BootstrapWaitConditionHandle:
1321     Type: 'AWS::CloudFormation::WaitConditionHandle'
1322     Metadata:
1323       'AWS::CloudFormation::Designer':
1324         id: fc4fe4ca-dc9d-4d47-a001-d9ee879b6d16
1325   BootstrapWaitCondition:
1326     Type: 'AWS::CloudFormation::WaitCondition'
1327     DependsOn: EC2Instance
1328     Properties:
1329       Count: 1
1330       Handle: !Ref BootstrapWaitConditionHandle
1331       Timeout: 2100
```

    d. On the top pane, click on the **check mark** to validate the template  .
    e. On the top pane, click on the **cloud** icon to create the stack template  .
    f. Click **Next**.

3. Provide the required stack parameters and use the default values for the rest of the parameters not specified below.
    a. **Stack Name**: VRP4_IMS
    b. **Instance Name**: VRP4_IMS
    c. **EC2 Instance Type** (may vary depending on instances available on AWS Outposts): m5.2xlarge
    d. **Key Pair for SSH** access: demokp1 (**NOTE**: This should be an existing key pair.)
    e. **Network Configuration** (specify VPC ID of AWS Outposts): vpc-2666ba4d (172.31.0.0/16)
    f. **Eth0 subnet** (specify the subnet created in the section AWS VPC Configuration): psn1
    g. **Eth1 subnet** (specify the subnet created in the section AWS VPC Configuration): psn2 (**NOTE**: This eth1 subnet will not be used in this deployment; however, the template expects a valid subnet here because otherwise stack creation fails.)
    h. Is the **Eth0 Network Interface behind NAT**: Yes (**NOTE**: Access to this instance from the data center is through the customer-owned IP.)
    i. **Eth0 NAT Hostname**: ip-10-xx-xx-xxx.us-east-2.compute.internal (**NOTE**: Usually in the form ip-x-xx-xx.<region>.compute.internal.)
    j. **Eth0 NAT IP**: 10.xx.xx.xxx
4. Click **Next**. In the section Stack failure options, select, "**Preserve successfully provision resources**" to preserve the resources that have been successfully created instead of deleting them in case of stack failure creation.
5. Click Next. Review the parameters entered and place a **check mark** on "I acknowledge that AWS CloudFormation might create IAM resources."
6. Click on **Create Stack**.
7. Watch the Events tab in the CloudFormation Stacks page. After the EC2 instance creation has completed and the BootstrapWaitCondition has initiated, associate the public and customer-owned Elastic IPs to the primary interface (eth0) of the IMS EC2 instance.
    a. Go to **Services** and select **EC2**.
    b. On the left pane under **Network and Security**, click on **Elastic IPs.**
    c. Select one of the **public Elastic IPs** created in the Elastic IPs creation section and click on **Actions**.
    d. Select **Associate Elastic IP address**.
    e. Select **the instance to associate** the Elastic IP: VRP4_IMS
    f. Choose the **public IP** to associate with the Elastic IP.
    g. Click **Associate**.
    h. Repeat steps d to g; however, select one of the **customer-owned IPs**.
8. Wait for the IMS stack creation to complete.
9. Go to the EC2 dashboard page and confirm that VRP4_IMS was created.



10. Modify the security groups so that SSH is allowed in the inbound rules.
    a. Go to the **EC2 page** and select the EC2 instance for the IMS you created: VRP4_IMS
    b. In the bottom pane, select the **Security tab** and select the security group "*RMIMSSecurityGroup*."
    c. Click on **Edit Inbound Rules**.
    d. **Add rule** SSH as type and enter the IP range of systems from the data center accessing this EC2 instance: 10.xx.xx.0/24
    e. **Add rule** SSH as type and enter the IP range of the private subnet on AWS Outposts: 172.31.0.0/16.
    f. Click **Save rules**.
11. SSH to VRP4_IMS from one of the systems in the data center via MobaXterm and enter the new password as prompted. Re-login with the newly entered password.
12. Validate that all services are running as normal by entering on the prompt: **manage -> services status ALL**.
13. Download and install the necessary NetBackup Resiliency Platform hotfixes from https://www.veritas.com/content/support/en_US/downloads. Follow the instructions in the README file of the download. In this deployment, we installed hotfixes 4.0.0.1 and 4.0.0.2, for example, so we executed the following steps from the RM CLI:
    a. **utilities>** sftp-session start put patch
    b. Use WinSCP to copy from Windows to the RM.
    c. **hotfix>** apply-hotfix 4.0.0.1
    d. **hotfix>** apply-hotfix 4.0.0.2
    e. **utilities>** troubleshoot run-tool fix-root-expired -a
14. In this example, we were not using a DNS. Therefore, we modified the /etc/hosts file on the RM. The NetBackup Resiliency Platform components would need to be able to resolve the hostnames of systems on-premises. From the RM CLI, get into the shell as follows:
    a. support -> shell
    b. Enter the initial password for the IMS (**NOTE**: You need to get this password from Veritas Support.)
    c. Modify the **/etc/hosts** files to include the hostnames and IP of all on-premises systems such as the vCenter, NetBackup (primary and media), and RM used in the deployment.

On-Premises Data Center IMS

In this sample deployment, an IMS is deployed in the data center to discover and monitor the VMware assets on-premises. After it is deployed, it is registered on the RM residing on AWS Outposts.

1. Download the IMS for VMware Virtual Appliance (*.ova) from https://www.veritas.com/content/support/en_US/downloads.
2. Please follow the product documentation (https://sort.veritas.com/documents?prod=itrp) to install and deploy the IMS in your VMware environment.
3. Download and install the hotfixes following the same procedure used for the IMS on AWS Outposts.
4. After the IMS is deployed, modify the /etc/hosts file similarly to what was done for the RM and IMS on AWS Outposts. Provide the hostname and IP addresses (use the customer-owned IPs) for the RM and IMS on AWS Outposts. In this example, they were:

```
10.xx.xx.xx   ip-10-xx-xx ip-xx-xx-xx-xx.us-east-2.compute.internal
10.xx.xx.xxx ip-10-xx-xx-xxx ip-xx-xx-xx-xxx.us-east-2.compute.internal
```

Step 4: NetBackup 9.1 Deployment

You need to deploy NetBackup on-premises prior to deploying NetBackup on AWS Outposts because it assumes you have already created the bucket where the backup images are stored.

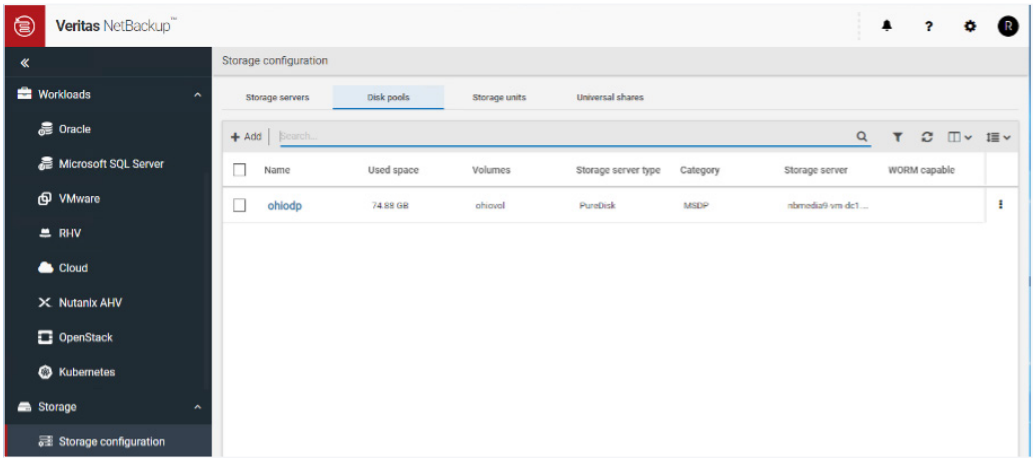
On-Premises NetBackup

For NetBackup primary and media server installation and configuration, please refer to the BYOS NetBackup deployment product documentation.

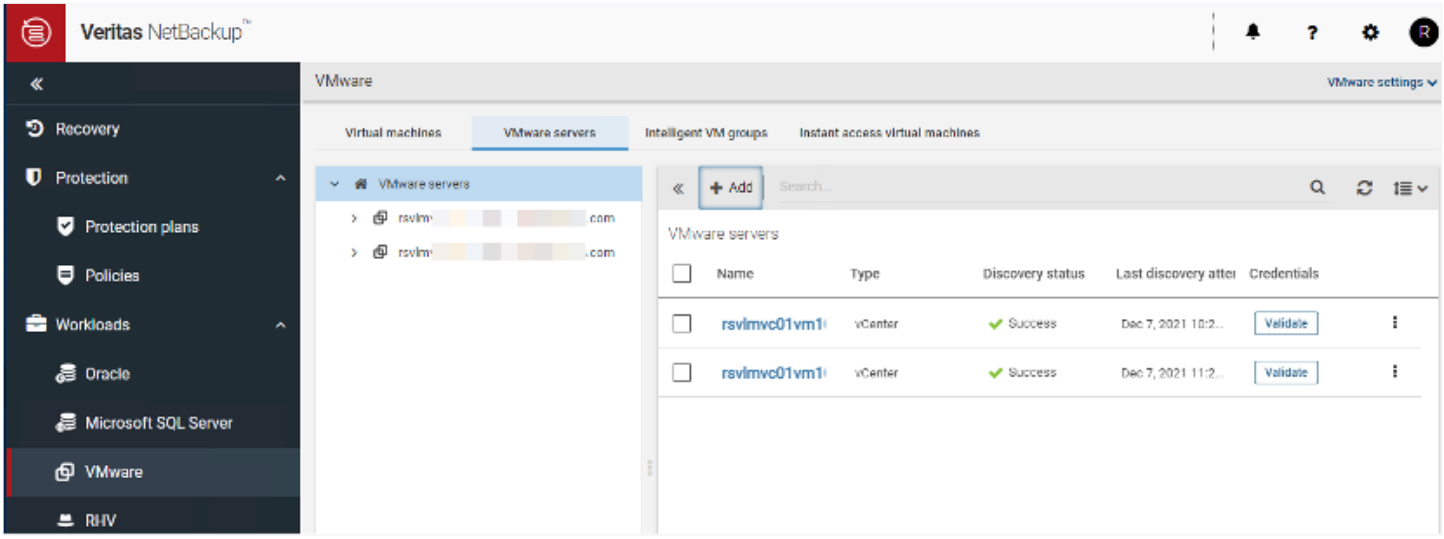
1. Download NetBackup from https://www.veritas.com/content/support/en_US/downloads.
2. Please follow the NetBackup Installation Guide in product documentation (https://www.veritas.com/support/en_US/article.100040135) to install and deploy NetBackup primary and media servers on desired systems or a virtual machine (VM) environment. In this example, we deployed NetBackup primary and media servers on a VM with RHEL 7.8 as the guest operating system.
3. Modify the /etc/hosts files to include the IP addresses and hostnames for the NetBackup primary and media servers, customer-owned IP addresses and hostnames of the RM and IMS on AWS Outposts, and the IP address and hostname of the IMS on-premises such that they can be resolvable.
4. Create an S3 bucket in the AWS cloud in the same region as AWS Outposts.
5. Create an S3 User in IAM of the AWS management console with both S3 permissions. For example, JSON view of policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

6. Create a media server deduplication pool (MSDP) storage server and logical storage unit (LSU) pointing to the S3 bucket on the AWS cloud created in Step 5. Use the secret key and access key created in the previous step. For more details, refer to WebUI Administrator's Guide v 9.1, https://www.veritas.com/support/en_US/doc/146133534-146134575-0/v136631110-146134575, and the NetBackup Deduplication Guide, https://www.veritas.com/content/support/en_US/doc/25074086-149019166-0/v141214178-149019166.



7. Add the VMware servers or vCenter to NetBackup.



On AWS Outposts NetBackup

The NetBackup instance deployed on AWS Outposts is the NetBackup Cloud Recovery Server (CRS). In this example, we did not use the NetBackup CRS CFT available on the AWS Marketplace. Instead, we installed NetBackup manually on an EC2 instance with RHEL 7.7.

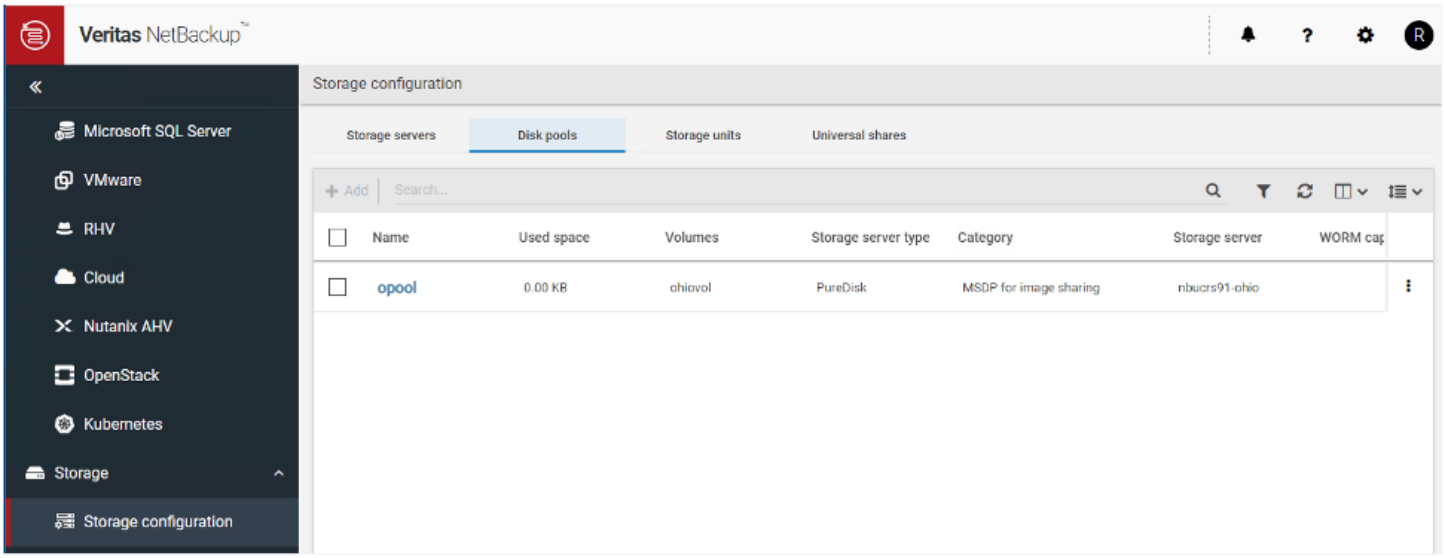
1. Download NetBackup from https://www.veritas.com/content/support/en_US/downloads.
2. Please follow the NetBackup Installation Guide product documentation (https://www.veritas.com/support/en_US/article.100040135) to install and deploy NetBackup primary and media servers on selected platforms.
3. NetBackup should be accessible by the RM and IMS on AWS Outposts. In this example, modify the /etc/hosts on the RM and IMS to include the private IP address and hostname of NetBackup. On the instance running NetBackup, modify the /etc/hosts file to include the hostname and private IP of the RM and IMS.

4. From the AWS management console, select IAM. Modify the user settings for the bucket created to store the NetBackup images created in the previous step to include certain EC2 permissions. These permissions allow the user to be able to read and convert the images stored within the bucket. Edit the policy for the user and enter the bolded text to the existing policy for the user. The JSON for the policy is shown below:

```
{
  “Version”: “2012-10-17”,
  “Statement”: [
    {
      “Action”: [
        “s3:*”
      ],
      “Resource”: [
        “*”
      ],
      “Effect”: “Allow”
    },
    {
      “Action”: [
```

```
            "ec2:CreateTags",
            "ec2:DescribeImages",
            "ec2:DescribeImportImageTasks",
            "ec2:ImportImage",
            "iam:ListRolePolicies",
            "iam:ListRoles",
            "iam:GetRole",
            "iam:GetRolePolicy",
            "iam:CreateRole",
            "iam:PutRolePolicy",
            "s3:ListAllMyBuckets",
            "ec2:DescribeCoipPools",
            "ec2:DescribeLocalGatewayRouteTableVpcAssociations",
            "ec2:AllocateAddress",
            "ec2:*"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow"
    }
  ]
}
```

5. Create a storage server that is a **Media Server Deduplication Pool (MSDP) for image sharing**. Create a disk pool and corresponding storage unit that specifies the volume and bucket that was created in the NetBackup on-premises. For more details, refer to the WebUI Administrator's Guide v 9.1, https://www.veritas.com/support/en_US/doc/146133534-146134575-0/v136631110-146134575, and the NetBackup Deduplication Guide, https://www.veritas.com/content/support/en_US/doc/25074086-149019166-0/v141214178-149019166. Use the same IAM user and specify the secret keys that were used to create the bucket on-premises.



Step 5: Configuration of NetBackup Resiliency Platform Components

The steps in this section link all the components installed and deployed in this solution. The process involves using the RM web user interface, the NetBackup Java Console, and the IMS command-line interface on-premises. All the hostnames of the components used in this solution should be resolvable for these steps to work properly. You can do so either by using /etc/hosts files or DNS. In this example deployment, we used /etc/hosts files. The hostnames and IP address would also need to be added to the hosts file of systems connecting to the RM and NetBackup management consoles.

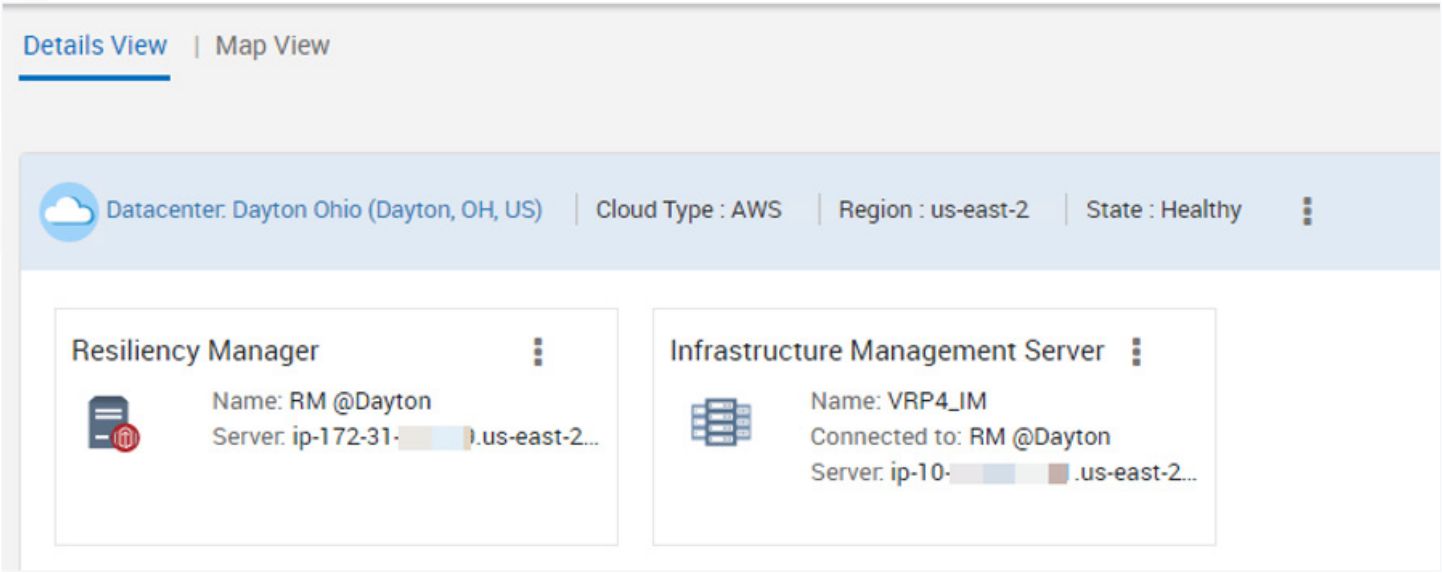
Bucket S3 Creation for RM Use

1. Create an AWS S3 bucket in the cloud for use by NetBackup Resiliency Platform in the same region as AWS Outposts:
   a. Log on to the AWS management console.
   b. Select **S3** service.
   c. Click on **Create Bucket**.
   d. Specify the name, same region as AWS Outposts, and public access setting. In this example, we provided US-East-2 (Ohio) and set the access to public.

Configure the RM and IMS on AWS Outposts

1. Log on to the RM graphical user interface (GUI) using the RM IP (customer-owned IP), (that is, https://<customer-owned-ip>) with user (admin@vrp.local) and credentials.
2. Configure Resiliency Domain by specifying the following parameters:
   a. **Domain Name**: AWS Outposts
   b. Click **Continue**.
3. Configure an AWS Outposts Datacenter by specifying the following parameters:
   a. Click on **Cloud Datacenter** (to indicate it is a cloud on-premises data center)
   b. **Datacenter Name**: Dayton Ohio
   c. **Datacenter Location**: Data, OH, US
   d. Click **Create**.
4. Set up a Cloud Configuration with the following parameters:
   a. **Cloud Type**: AWS
   b. **AWS Configuration Name** (any name): AWS Outposts
   c. **S3 Bucket Name**: <the bucket name created in the previous section>
   d. **Region**: us-east-2
   e. Click **Submit**.
5. Configure the Infrastructure Management Server (IMS) on AWS Outposts as follows:
   a. From the dashboard, click on Settings  on the top right.
   b. Select **Infrastructure** settings.
   c. Click on the "**Infrastructure Management Server +**" at the top right to add the IMS.
   d. Select **Data Center**: Dayton Ohio
   e. **Provide Friendly Name/Tag for IMS**: VRP4_IM
   f. **Server** (FQDN of IMS): <customer-owned IP>.us-east-2.compute.internal (that is, ip-10-81-xx-xx.us-east-2.compute.internal)
   g. Because the IMS is directly accessible by the RM, **provide the username: admin and password** provided during install.
   h. Click **Submit**.

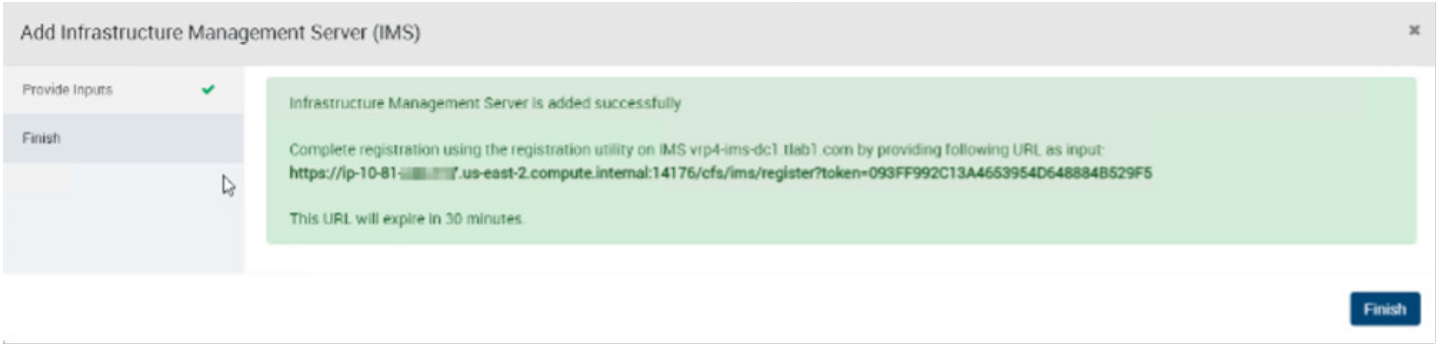
6. Validate everything is healthy and connected.



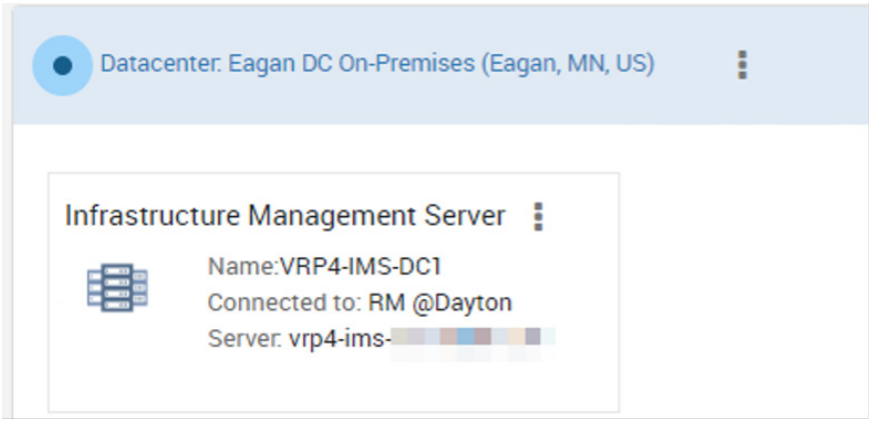
Configure On-Premises IMS

For security, the firewall rules between the on-premises data center and AWS Outpost prevent the request from being initiated from AWS Outposts and entering the data center. The IMS needs to be registered with the RM via the IMS command-line interface (CLI).

1. Configure the on-premises data center as follows:
   a. Click on the "**Datacenter +**" to add the on-premises data center.
   b. Specify the **Datacenter Location**: Eagan, MN, US
   c. **Datacenter Name**: Eagan DC On-Premises
   d. Click **Submit** and **Finish**.
2.
   a. Click on the "**Infrastructure Management Server +**" at the top right to add the IMS.
   b. **Data Center**: Eagan DC On-Premises
   c. **Friendly Name/Tag**: VRP4-IMS-DC1
   d. **Server** (FQDN of IMS): vrp4-ims-xx.xxx.com
   e. Because the IMS is not directly accessible by the RM on AWS Outposts as mentioned above, do not provide any username and password in this section. Remove any default values if present.
   f. Click **Submit**.
   g. Copy the URL generated.

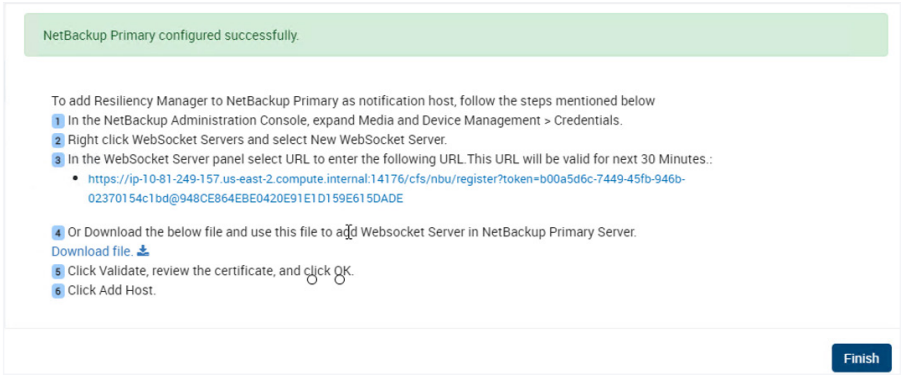


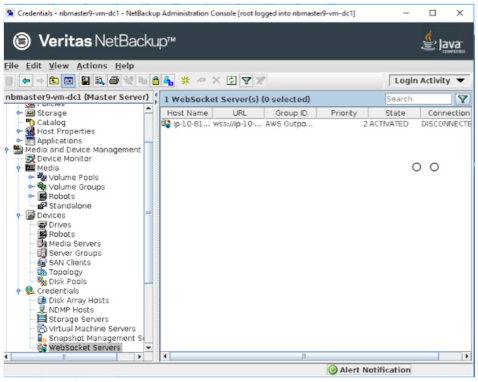
3. Register the IMS URL on the IMS CLI.
   a. **SSH** into the IMS CLI.
   b. Enter **manage**.
   c. Enter **configure ims_register**.
   d. Provide the **IMS Registration URL when prompted**: <the URL copied in the previous step>
   e. Confirm registration of the Eagan on-premises data center and IMS on the RM dashboard.



Register the NetBackup Primary Server (on-premises) on RM

1. In Settings, in Infrastructure, select the **Copy Manager** under the Eagan DC On-Premises data center.
2. Click on "**+ NetBackup Primary**" on the top right.
3. Provide the following information:
   a. **Infrastructure Management Server**: <select the on-premises IMS from the drop-down menu>
   b. **NetBackup Primary Server**: FQDN of NetBackup
   c. Select **Credential** and enter username and password for NetBackup.
   d. **Uncheck** the NetBackup Primary Server is accessible from this Resiliency Manager. Due to firewall rules requests and access to the systems in the data center cannot be initiated from AWS Outposts.
   e. Click **Next**.
4. Follow the instructions in the next page. This step requires downloading the file if using NetBackup 9.1 or later and copying the JSON file to the NetBackup on-premises and using the NetBackup Java administration console to upload the file and add the host.





5. Click **Finish** when done.

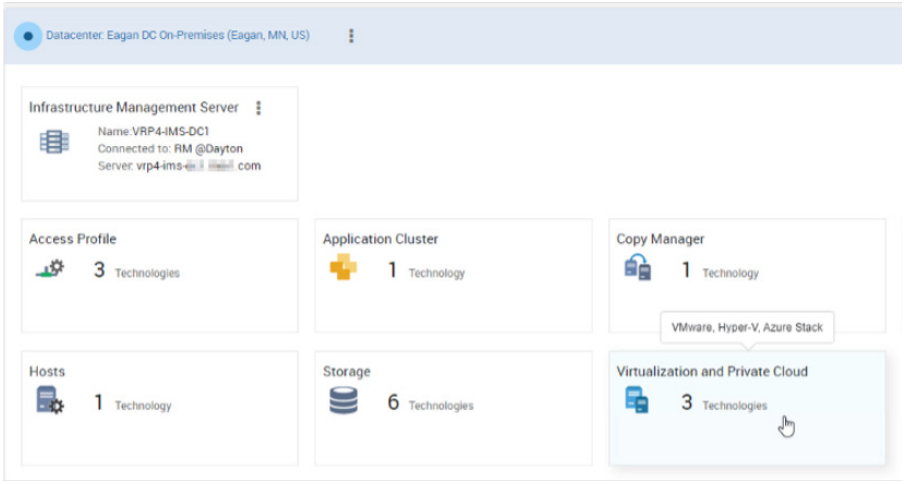
Configure the NetBackup Cloud Recovery Server in the RM

1. In Settings, in Infrastructure, select **Copy Manager** under the Dayton Ohio data center.
2. Click on "**+ NetBackup Cloud Recovery Server.**"
3. Specify the NetBackup information (**NOTE**: The NetBackup hostname should be resolvable from the IMS and RMS.)
   a. **NetBackup Cloud Recovery Server**: <FQDN or the hostname of NetBackup deployed on AWS Outposts>
   b. Connect using **Credential** and enter the username and password.
   c. Click **Submit** and **Finish**.

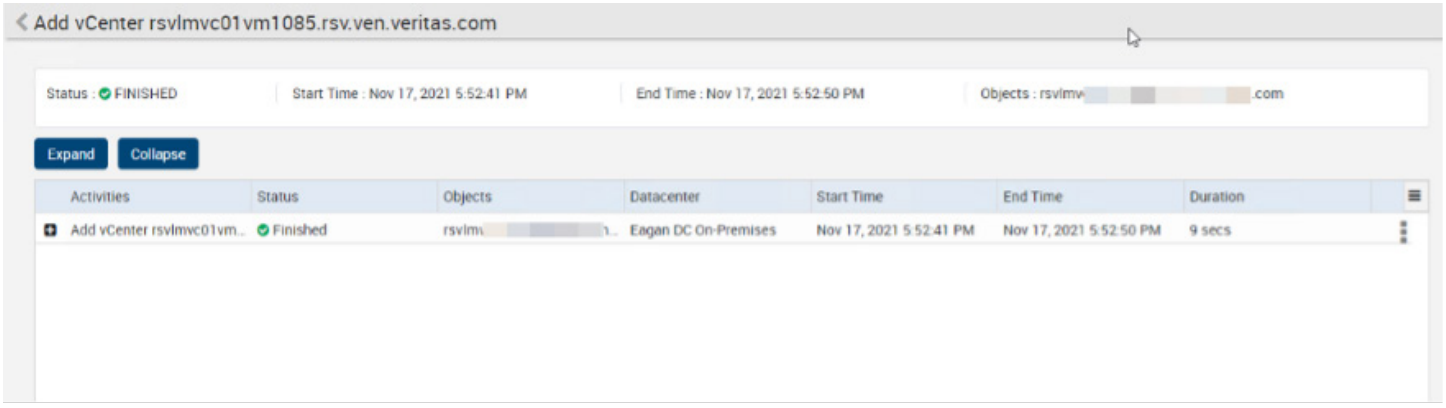
Configure VMware Assets On-Premises

The vCenter/ESXi needs to be reachable from the on-premises IMS, so add the hostname and IP for the vCenter/ESXi server in the /etc/hosts file of the IMS.

1. In the RM GUI, in Settings and under Infrastructure, click on Virtualization and Private Cloud under the data center for Eagan DC On-Premises.
2. Click on "**+ vCenter.**"
3. Enter the vCenter login information:
   a. **Server**: FQDN of vCenter
   b. **Username**: <the username for vCenter>
   c. **Password**: <the password for vCenter>
   d. **Discovery Host name**: <FQDN of the IMS on-premises>
   e. Click **Next**.



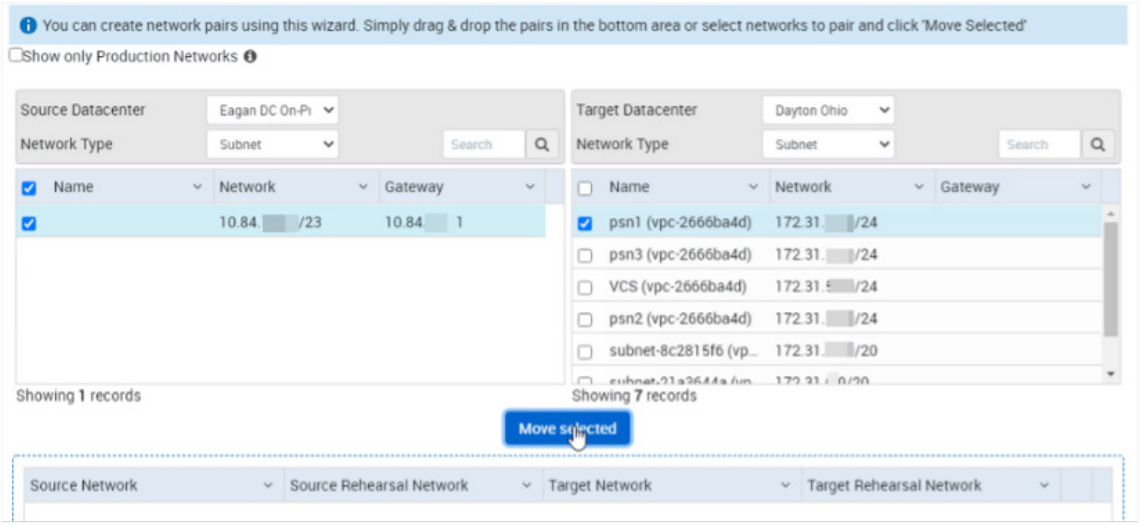
4. Select the ESXi to monitor and manage or auto-discover the ESXi.
5. Click **Next** and **Submit**.
6. Click **Finish**.
7. Validate the vCenter has been added.



Configure Network Pairing

The network pairing links the subnets' details on AWS Outposts with the on-premises subnet. During the automatic recovery or migration of a workload, IP addresses are allocated onto the subnet specified in this section.

1. Click on the left pane **Infrastructure Pairing** icon .
2. Click **New Network Pair**.
3. Select the parameters.
   a. **Source Datacenter**: Eagan DC On-Premises
   b. **Network Type**: Subnet
   c. **Select the network** and gateway for the Eagan On-Premises Datacenter.
   d. **Target Datacenter**: Dayton, OH
   e. **Network Type**: Subnet
   f. Select the **network/subnet** and gateway for the Dayton DC.
4. Click on **Move selected**.



5. Click **Submit** and **Finish**.

Step 6:Validation of Solution

Validation of this solution involves backing up a VM workload in the on-premises environment using NetBackup and an AWS S3 bucket as the target storage. You then create resiliency groups on the RM to conduct a recovery or rehearsal of the workload onto AWS Outposts.

VM Requirements Prep

For VMs to be recovered on AWS and/or AWS Outposts, you need to configure and/or install certain requirements on the VMs prior to backup, which include:

- Some EC2 instances require you to download and install AWS ENA (Elastic Network Adapter) and NVMe drivers onto the VM's guest operating systems prior to backing it up.
- The boot options for the VM would need to be Bios Legacy instead of EFI, especially for the Windows platform.
- Guest VMware tools need to be installed on the VMs being backed up.
- For the Linux platform, disable predictable network interface names by adding net.ifnames=0 option in the /etc/default/grub file.

Windows VM

1. Validate that the boot options for the VM is BIOS. (**NOTE**: When the VM is modified from EFI to BIOS it may require a re-install of the guest operating system.)
2. Mount and install the VMware Guest tools.
3. Download and install (install.ps1) the AWS ENA drivers. For more details, refer to https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/enhanced-networking-ena.html
4. Download and install (install.ps1) the NVME drivers. For more details, refer to https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/aws-nvme-drivers.html
5. Reboot the VM.

Linux VM

1. Validate that the boot options for the VM is BIOS. (**NOTE**: When the VM is modified from EFI to BIOS it may require a re-install of the guest operating system.)
2. Mount and install the VMware Guest tools.
3. Download and install the AWS ENA drivers if not already installed. For further details and where to download the drivers, refer to https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking-ena.html. ENA drivers may already be installed on some versions of the Linux operation system such as RHEL 7.7 and later. Validate by using the command "modinfo ena." An example output:

```
[root@aportal ~]# modinfo ena
filename:       /lib/modules/3.10.0-1127.el7.x86_64/kernel/drivers/net/ethernet/amazon/ena/ena.ko.xz
version:        2.0.3K
license:        GPL
description:    Elastic Network Adapter (ENA)
author:         Amazon.com, Inc. or its affiliates
retpoline:      Y
rhelversion:    7.8
srcversion:     3E0F01EBC629D59AD4CEA6A
alias:          pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
intree:         Y
vermagic:       3.10.0-1127.el7.x86_64 SMP mod_unload modversions
signer:         Red Hat Enterprise Linux kernel signing key
sig_key:        D8:63:12:62:84:DB:0E:83:32:60:9E:1E:3B:DF:C5:AE:66:33:0B:73
sig_hashalgo:   sha256
        parm:           debug:Debug level (0=none,...,16=all) (int)
```

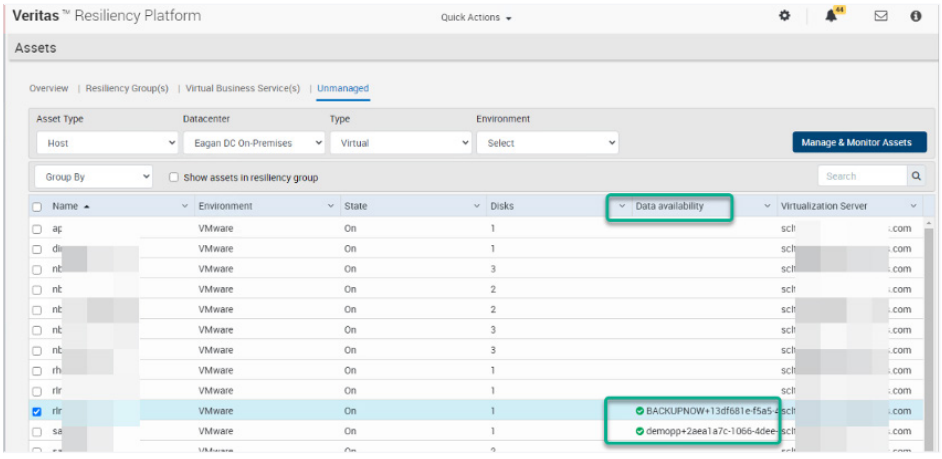
4. If your system uses predictable network interfaces by default, disable it.
   a. Validate by checking the system or udev versions on RPM-based systems:
      [root~]# **rpm -qa | grep -e '^systemd-[0-9]\+\|^udev-[0-9]\+'**
      systemd-219-73.el7.1.x86_64
   b. If the system or udev version is 197 or greater as shown in the example output above (that is, system-219*), disable predictable network interfaces; otherwise, it will rename ethernet network interfaces and cause issues in connecting to instance after recovery. Enter the following to add net.ifnames=0 option to /etc/default/grub file:
      [root~]# **sudo sed -i '/^GRUB_CMDLINE_LINUX/s/\"$/\ net\.ifnames\=0\"/' /etc/default/grub**
5. Download and install the NVME drivers if not installed. Run the following command to validate if the kernel module is installed:
   [root~]# **modinfo nvme**
   filename: /lib/modules/3.10.0-1127.el7.x86_64/kernel/drivers/nvme/host/nvme.ko.xz
   version: 1.0
   license: GPL
   Per the NetBackup Resiliency Platform product documentation, enable the NVME drivers on Linux by entering the following command:
   [root~]# **echo "add_drivers += \" nvme \" " > /etc/dracut.conf.d/ena.conf**
   [root~]# **dracut -f -v**
6. Do **NOT reboot** after install of the drivers. If you do a reboot, it modifies the network interfaces.

Backup of VM Workload

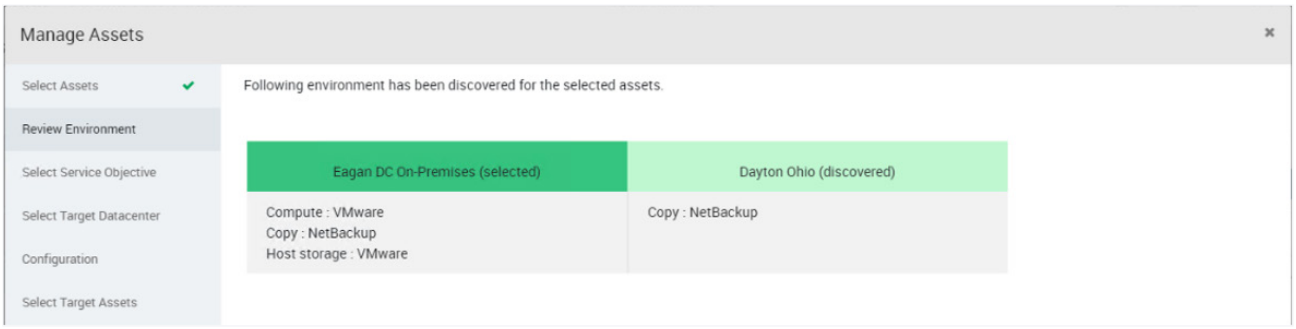
1. Log on to the NetBackup web graphical user interface (GUI).
2. On the left pane, select **Protection Plans** and define a **protection plan** indicating the schedule and the backup storage target. The storage target is the AWS S3 bucket created in and specified in the NetBackup on-premises deployment section.
3. Select the VMs to back up from the VMware workload.
4. Click on **Backup now**.
5. Select the protection plan defined in step 2 to use to conduct the backup.
6. Click on **Activity monitor** on the left pane and validate the backup completes successfully.

Creation of Resiliency Groups

1. Log onto the RM GUI.
2. Click on the **Assets** icon  on the left pane.
3. Click on the **Unmanaged** tab.
4. Specify the following parameters:
   a. **Asset Type**: Host
   b. **Datacenter**: Eagan DC On-Premises
5. Select the VM backed up by NetBackup on-premises. (**NOTE**: The VM should have Data Availability information to be able to do a recovery as shown below.)
6. Click **Next** in the next page.

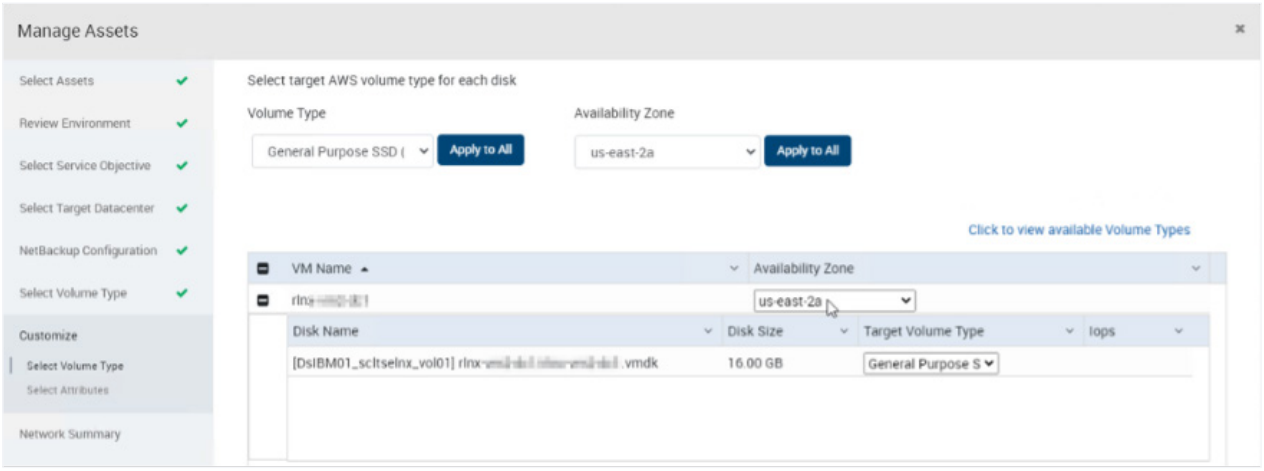


7. Validate there is a NetBackup copy manager on the selected source data center and discovered data center. For example:

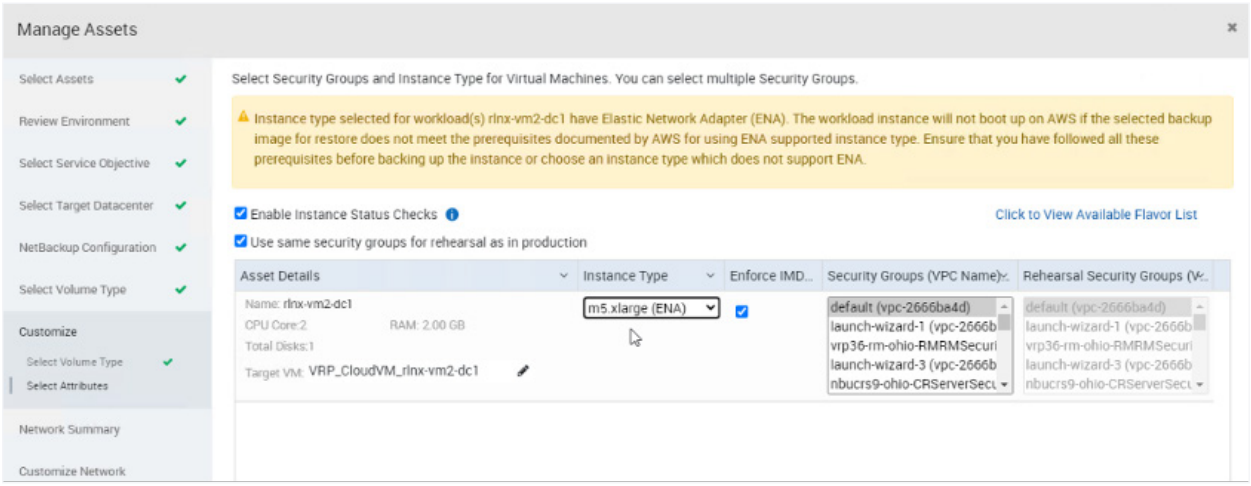


Click **Next**.

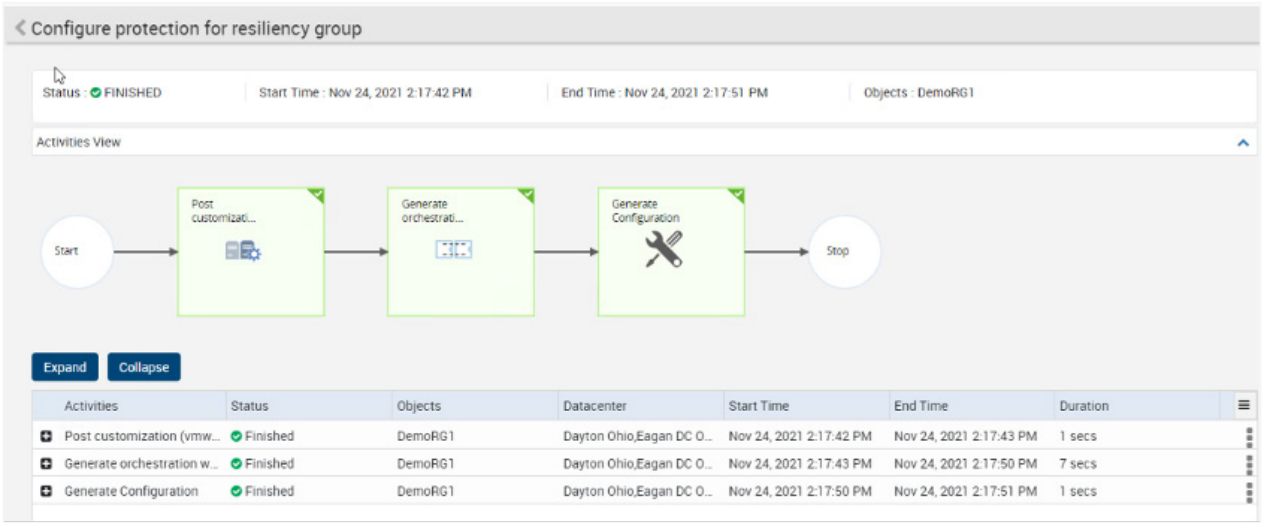
8. **Select the Service Objective** that would allow recovery of hosts. In this example, select "**Local and remove recovery of hosts (not monitored)**" and click **Next**.
9. Validate that the target data center is AWS Outposts and local is the on-premises data center. Click **Next**.
10. Validate the discovered NetBackup configuration for the selected assets and click **Next**.
11. Select **the target volume type** for each disk and the availability zone. The availability zone should be the same zone as AWS Outposts.



12. Select the **Security Groups** and Instance Type for the VMs. Select the Instance Type that is available on AWS Outposts. AWS Outposts is configured with specific instance types based on your specific requirements during purchase, installation, and configuration.



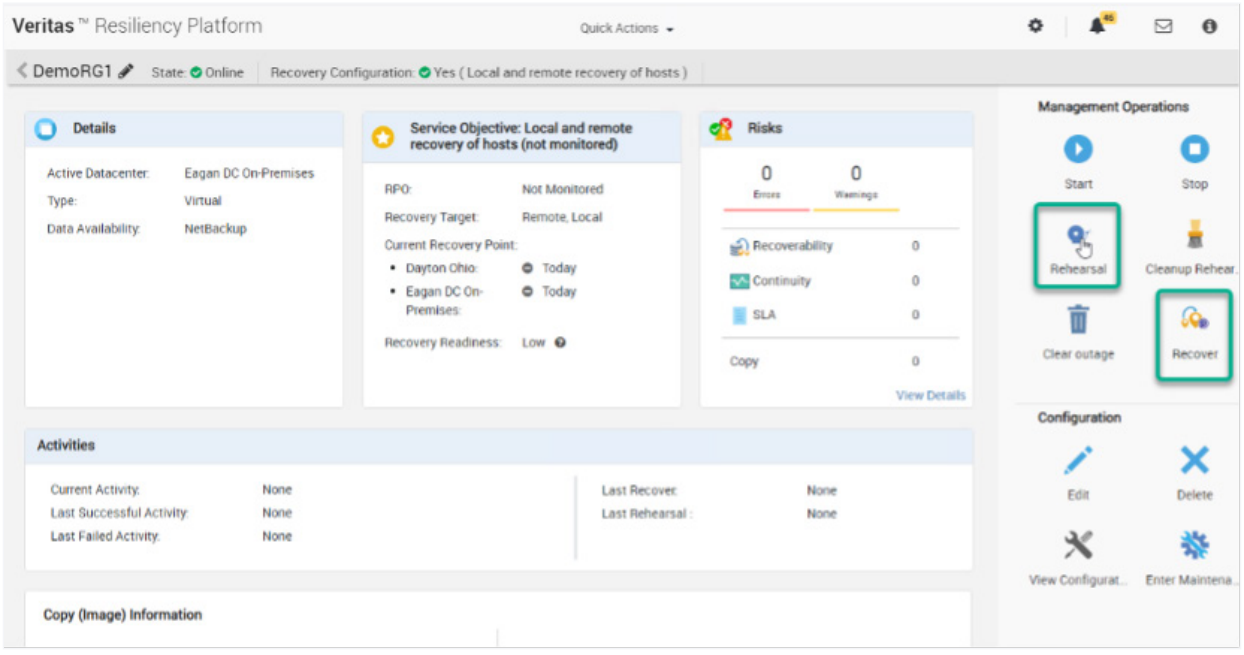
13. Validate Network mappings for the resiliency group and click **Next**.
14. Customize the network (optional). Click **Next**.
15. Provide the name for the resiliency group, for example DemoRG1. Click **Submit**.
16. Click **view progress**. An example view is shown below.



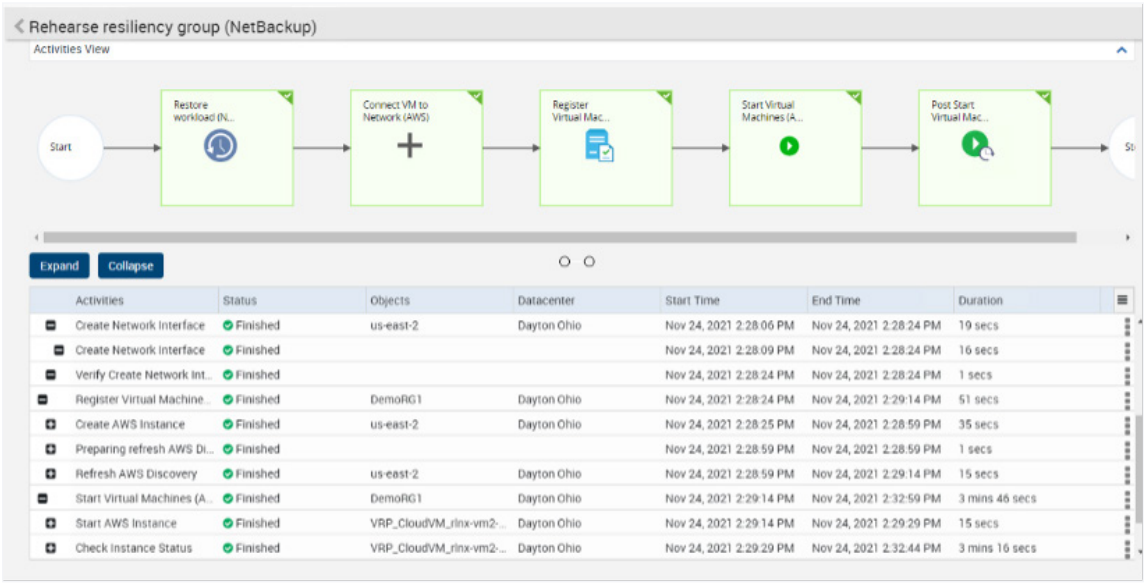
Recovery of VMs on AWS Outposts

Recovery of a VM on AWS Outposts assumes the VM is in an online state.

1. Click on the **Assets** icon and click on the **Resiliency Groups** tab.
2. Click on the **3 dots** on the far right-hand side of the resiliency group created in the previous section and select "**Details**."
3. Click on **Recover** or **Rehearsal** on the right pane. The difference between rehearsal and recover is that the VM on-premises will not be un-registered on VMware for a rehearsal operation. Prior to actual recovery, you would need to clean up any active rehearsals conducted.

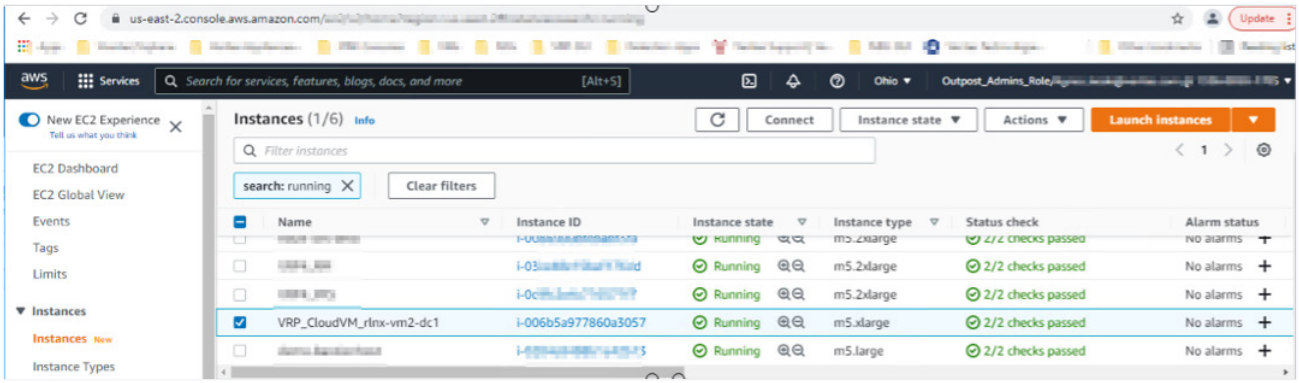


4. Select the **target datacenter**, which would be AWS Outposts (for example, Dayton Ohio) and click **Next**.
5. Select the **Recovery Points**. You can choose the **latest** or from a specific **time range**. Click **submit** after selection.
6. Click **view progress**. Below is an example of a completed rehearsal.



Validation of Creation

1. Log onto the AWS management console and validate that the instances recovered.



2. **SSH** or **RDP** onto the EC2 instance and validate the system is accessible and running from the bastion-host. (**NOTE**: The security group would need to have inbound rules to allow the IP of the bastion-host to SSH. Use the same credentials as on-premises. You do not need an AWS key pair to access the EC2 instances.)

References

AWS Services

- AWS Outposts
- Amazon EC2
- Amazon Virtual Private Cloud (Amazon VPC)
- Amazon Route 53
- Amazon Elastic Block Store (EBS)
- AWS Outposts networking

Veritas NetBackup Resiliency Platform

- AWS Outposts

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.

VERITAS

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

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

V1494 07/22