vCloud Availability for vCloud Director 1.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-002212-01

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About vCloud Availability for vCloud Director

vCloud Availability for vCloud Director Guide provides information on how to install and configure the vCloud Availability for vCloud Director 1.0 DRaaS service.

Intended Audience

This information is intended for anyone who wants to install, upgrade, or use vCloud Availability for vCloud Director. The information is written for experienced Windows or Linux system administrators who are familiar with virtual machine technology and datacenter operations.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

Updated Information

This *vCloud Availability for vCloud Director Guide* is updated with each release of the product or when necessary.

This table provides the update history of the *vCloud Availability for vCloud Director Guide*.

Revision	Description
EN-002212-01	■ Updated file name in topic "vRCS," on page 31. ■ Updated file name in topic "vRS," on page 34.
EN-002212-00	Initial release.

Introduction 1

This section describes the core architecture of the vCloud Availability for vCloud Director service.

vCloud Availability for vCloud Director is a Disaster Recovery-as-a-Service (DRaaS) solution that provides simple and secure asynchronous replication and failover for vSphere managed workloads. The service operates through a vCloud Air Network Service Provider, and each installation provides recovery for multiple tenants. The service provides the following features:

- Self-service protection, failover and failback workflows per VM
- Recovery point objective (RPO) from 15 minutes to 24 hours
- Initial data seeding by shipping a disk

For the service provider, vCloud Availability for vCloud Director:

- Integrates with existing vSphere environments
- Multi-tenant support
- Built-in encryption of replication traffic
- Supports multiple vSphere versions
- Supports multiple ESXi versions
- Individual systems are isolated as virtual machine files
- Full integration with vCenter web client
- Automation provided through standard web service APIs

Failover from On-Premises to Cloud
Replicates data from on-premises vSphere workloads to service provider cloud environments. After the virtual machines are replicated, failover support for executing the workloads in the cloud. Recovery Point Objective (RPO) can be configured from 15 minutes to 24 hours.

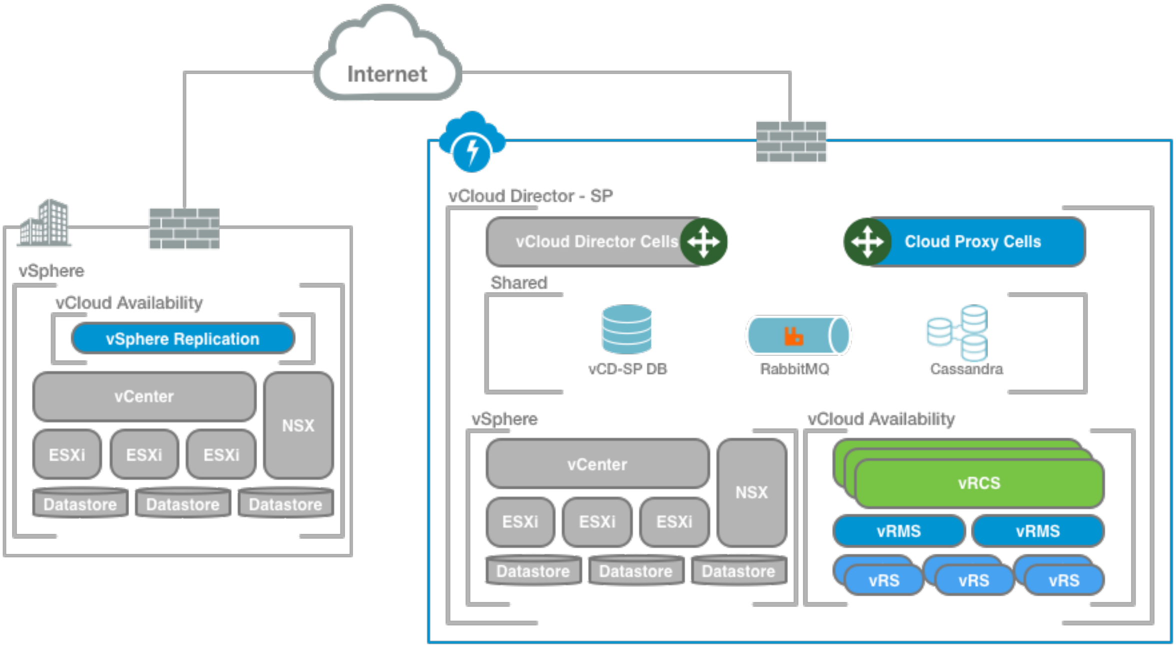
Fail back to on-premises
For failover loads that have been migrated to the cloud, changes can be replicated back to the on-premise environment. Workloads can then failback for execution in the on-premise environment.

Multiple Point In Time (MPIT) Recovery
Up to 24 restore points can be created. Depending on the RPO configuration, restoration is available from any recovery point.

Architecture

The architecture of the solution relies on the service provider environment that provides the replication target and the customer, or tenant, environment that employs vSphere replication to move the data to the service provider. In the service provider environment, multiple components operate together to support replication, secure communication, and storage of the replicated data. Each service provider can support recovery for multiple customer environments that can scale to handle increasing loads for each tenant, and for multiple tenants.

On the tenant side, a single VM instance is deployed in the tenant vSphere environment. This provides management service that is used to oversee the replication operation for each replicated VM. Standard vSphere Replication is used to exchange this information with the service provider infrastructure.



Components

Table 1-1. The different components in the architecture handle the management of the overall service, and the storage of the replicated information.

Name	Abbreviation/Internal Name	Description
vSphere Replication Cloud Service	vRCS/HCS	A tenant-aware replication manager that provides the required API for managing the service and all the components. vRCS registers as a vCloud Director extension enabling the functionality through the existing vCloud Director interface.
vSphere Replication Manager	vRMS/HMS	The management server manages and monitors the replication process from tenant VMs to the service provider environment. A vRMS service runs for each vCenter and tracks changes to VMs and infrastructure related to replication.
vSphere Replication Server	vRS/HBR	The replication server that sends or receives the replication information and records the changes for each replicated VM.
vCloud Tunneling Agent	vCTA	vCloud Tunneling Agent is a software component which supports tunneling functionality at the on-premise datacenter. vCloud Tunneling Agent is responsible for orchestrating secure tunnel creation for both to-the-cloud and from-the-cloud tunnels.
vCloud Director	vCD	vCloud Director is a software solution that enables service providers to build secure, multi-tenant private clouds by pooling infrastructure resources into virtual datacenters and exposing them to users through Web-based portals and programmatic interfaces as fully automated, catalog-based services.
Cloud Proxy	n/a	Provides the vCD endpoint for tunnels use to replicated data from on-prem vCTA to/from vCD.
Cassandra	C	Cassandra is used to store metadata about the replication, replicated VM instances, and infrastructure elements required to support the service. Cassandra is used as a fault-tolerant data store.
RabbitMQ	n/a	An open source message broker that implements the Advanced Message Queuing Protocol (AMQP). When vRCS registers as a vCD extension, RabbitMQ is used to exchange information with vCloud Director.

Deployment of Components 2

Deployment involves installing components within the service provider environment, and just one component within each tenant environment.

Deployment of vCloud Availability for vCloud Director is based on configuring several different components. Some components, such as Cassandra, or NSX may already be deployed within your environment. All the components work together to provide the overall service. Some components are required only once, others are required two or three times to provide redundancy, some are required multiple times to support an increasing number of protected virtual machines.

A typical deployment includes the following components:

- Tenant Service
  - The tenant service, which consists of the vSphere Replication Appliance, is installed on-site in an existing vSphere environment and provides the necessary tools to replicate information to the service provider site.
- Service Provider Service
  - The service provider side of the system supports one or more tenants. A Cloud Proxy provides network connectivity, a replication manager handles the replication of data, managers control the replication, and a cluster of appliances receive the disk updates and store the information ready for the failover of operation from the tenant site to the service provider environment. The number of instances of each component required varies depending on the number of VMs on the tenant side that need to be protected.

For scaling purposes, to increase the number of supported VMs that can be added to the disaster recovery solution, the following components should be installed multiple time:

Depending on the deployment type, different numbers and allocations of the components should be used.

This chapter includes the following topics:

Component Versions and Compatibility

Different versions of supported product environments are available for tenant and service provider environments that can support vCloud Availability for vCloud Director.

Service Provider Components

The system requirements for the provider side lists all the components required to support the vCloud Availability for vCloud Director service. VMware components are available for download from the VMware Web site.

- VMware ESXi 6.0.0u2
- NSX for vSphere 6.1.5
- vCloud Director 8.10
- vCloud Availability for vCloud Director 1.0, separate versions are available for different environments:
  - Hyper-Converged Software (HCS) Only, vCloud_Availability_4vCD_Cloud_Service_OVF10
  - Host Based Replication (HBR) Only, vCloud_Availability_4vCD_AddOn_OVF10
  - HCS and HBR combined, vCloud_Availability_4vCD_OVF10
- RabbitMQ 3.4.3 (from https://www.rabbitmq.com/download.html)
- Cassandra 2.2 (requires Java SE7, from http://cassandra.apache.org/download/)

Tenant Components

The tenant side of the deployment supports multiple versions of vCenter Server and associated components when replicating to the required service provider environment.

The following table details the compatibility of different installed tenant environments and the supported functionality.

- Failover - replication and automated failover.
- Failover-Test - test the viability of the failover without initiating true failover.
- Failback - replication back to the on-premise system after a failover.
- Multiple Point-In-Time (MPIT) recovery points

Table 2-1. Tenant Feature and Component Compatibility

Feature Compatibility	vCenter Server 6.0u1	vCenter Server 5.5u2	vCenter Server 5.1
Supported ESXi Versions	6.0, 5.5, 5.1, 5.0	5.5, 5.1, 5.0	5.1, 5.0
Supported vSphere Replication Versions	6.1.x, 6.0.x	5.8.x, 5.6.x	5.8.x, 5.6.x
Failover Supported	Yes	Yes	Yes
Failover-Test Supported	Yes	Yes	Yes
Failback Supported	Yes	No	No
Multiple Point-In-Time (MPIT) Supported	Yes	No	No

Development and Testing Deployments

Development and test environments can use a minimum configuration to confirm and test the service.

For development and testing deployments, the architecture and system count for each component should use the following configuration for a base installation. In these types of deployment, the configuration provides the bare minimum required to support the service.

In a development or test environment, to verify functionality and develop the final solution, deployment can consist of one of each of the following components.

- Cassandra
- Cloud Proxy (can be executed on the vCloud Director cell)
- RabbitMQ
- vCloud Director
- vRCS
- vSphere Replication Manager
- vRS

Production Deployment

Production deployments of vCloud Availability for vCloud Director have specific sizing and component configurations.

A production deployments uses multiple components to provide support for a larger number of protected virtual machines, and to provide fault-tolerance within the DRaaS environments.

Production Architecture

Production deployments must meet certain requirements.

- At each tenant site, there is one or more single-tenant environment that needs to be protected.
- In the service provider disaster recovery site, one or more vCloud Director is configured with a specific number of components designed to handle the required number of VMs from each tenant.
- A single vCloud Director environment in a datacenter is designed to host up to 500 individual tenants.

Using this information as a base, a new vCloud Availability for vCloud Director service pod must be configured and installed for each 100 tenants that use the service.

In a single vCloud Director deployment, there is a limit to the number of VMs that can be replicated as part of the DR solution. The exact combination depends on the number of VMs that must be supported combined with the system limits for each component.

Component Sizing

Individual components have a minimum installation count required for a base installation.

Table 2-2. Relative Component Sizing

Component	Related Component
vCloud Director	2 vCenter Server Appliances
vSphere Replication Manager	4 vSphere Replication Servers

Component Limits

Individual components have limits for the maximum number of supported services, instances, or connections required.

Table 2-3. Component Counts and Limits per pod

Component	Limit
Cloud Proxy	2000 Connections
vSphere Replication Server	500 active replications
Tenants	500 per vCloud Director
vCloud Director	10 vCenter Server Instances

Sample Deployment Scaling

Using the information on sizing and configurations, for a single pod, supporting up to the maximum of 100 tenants.

Table 2-4. Component Counts for Production Deployments

Protected VMs	500	1000	2000	3000	5000	10000
vRS	12	24	48	60	120	240
vRCS	2	2	3	3	3	3
Cloud Proxy	2	2	2	2	3	5

Sample Deployment Configuration

Production deployments depends on the number of VMs and Tenants that need to be supported. An example configuration of a production deployment is provided in the table.

Table 2-5. Component Deployment for Production Deployments

Component	Host 1	Host 2	Host 3	Host 4	Quantity
Cassandra	Yes	Yes	Yes	Yes	3
Cloud Proxy			Yes	Yes	2
Microsoft SQL Server	Yes	Yes		Yes	3
NFS				Yes	1
NSX Manager				Yes	1
RabbitMQ		Yes	Yes	Yes	3
vCenter Server Appliance	Yes				1
vCloud Director	Yes	Yes			2
vSphere Replication Cloud Service		Yes			1
vSphere Replication Manager		Yes			1
vSphere Replication Server	Yes	Yes	Yes	Yes	4

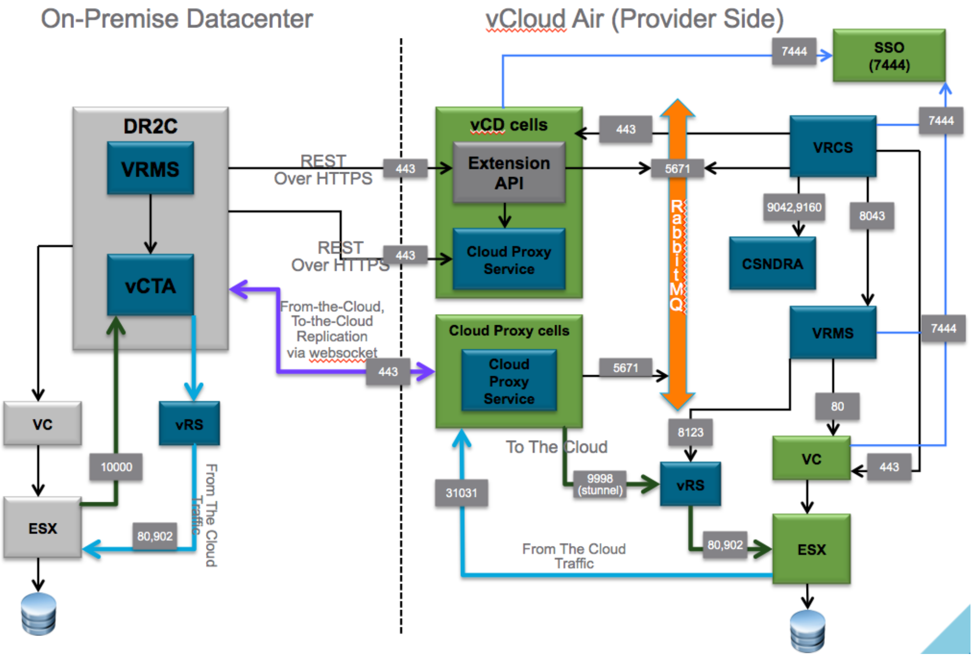
Within a production deployment the underlying network architecture is important. The recommended network configuration:

- Each underlying physical ESXi installation is configured with VMXNET3 high speed network adapter, connected to two separate 10Gbe switches using NIC teaming.
- The two switches are connected to each other via two 40gbe QSFP cables.
- The switches are configured to present a VLAN that is configured as a port group on the ESXi hosts.
- The virtual machines that make up the environment are all configured in this flat broadcast domain.

Firewall and Port Configuration

Network Firewall ports that are required to be open between different components and systems.

The diagram below shows the flow of network ports and data through a typical deployment on both the service provider and tenant side.



The following table provides a list of ports that should be open between the different systems and components.

Table 2-6. Firewall Port Component Configurations within a Service Provider Deployment

Source	Destination	Port Number
Cloud Proxy	vCloud Director DB	1433
Cloud Proxy	RabbitMQ	5671
Cloud Proxy	vSphere Replication	9998
Cloud Proxy	vSphere Replication	31031
ESXi	vCloud Director	31031
ESXi	Cloud Proxy	31031
External	Cloud Proxy	443
Operator	vSphere Replication Manager	5480
Operator	vSphere Replication	5480

Table 2-6. Firewall Port Component Configurations within a Service Provider Deployment (Continued)

Source	Destination	Port Number
SSO	Active Directory	389
vCloud Director	RabbitMQ	5671
Cloud Proxy	vCloud Director	61616
vCloud Director	Cloud Proxy	61616
vCloud Director	SSO	7444
vSphere Replication Manager	SSO	7444
vSphere Replication Cloud Service	SSO	7444
vSphere Replication Cloud Service	vCenter Server	80
vSphere Replication Cloud Service	vCloud Director	443
vSphere Replication Cloud Service	vCenter Server	443
vSphere Replication Cloud Service	External	443
vSphere Replication Cloud Service	RabbitMQ	5671
vSphere Replication Cloud Service	vSphere Replication Manager	8043
vSphere Replication Cloud Service	Cassandra	9042
vSphere Replication Cloud Service	Cassandra	9160
vSphere Replication Manager	vCenter Server	80
vSphere Replication Manager	vCenter Server	80
vSphere Replication Manager	vCenter Server	443
vSphere Replication Manager	vSphere Replication	8123
vSphere Replication	ESXi	80
vSphere Replication	ESXi	902
vShield Manager/NSX	SSO	7444

For the deployment within a tenant, the following ports must be open:

Table 2-7. Firewall Port Configurations within a Tenant Environment

Source	Destination	Port Number
vSphere Replication Appliance	vCenter Server	80
vSphere Replication	ESXi	80
vSphere Replication	ESXi	902 (TCP and UDP)
Operator	vSphere Replication Appliance	5480
vCenter Server	vSphere Replication Appliance	8043
vSphere Replication	vSphere Replication Manager	8123
Operator	vCenter Server	10443

Table 2-7. Firewall Port Configurations within a Tenant Environment (Continued)

Source	Destination	Port Number
ESXi	vSphere Replication at Service Provider	31031
vSphere Replication	Service Provider	902
Operator	vSphere Replication	5480
vSphere Replication Manager	vSphere Replication	8123
ESXi	vSphere Replication	31031
ESXi	vCenter Server at Service Provider	80
vSphere Replication Appliance	vCloud Director at Service Provider	443
ESXi	vSphere Replication Appliance at Service Provider	10000-10010

Prerequisites Before Installation 3

Before configuring your service provider environment to support the DRaaS service, certain systems should be in place.

This chapter includes the following topics:

vCloud Director

vCloud Director must be configured to support an environment suitable for securely supporting multiple tenants.

The vCloud Director should be configured to use these settings:

- AMQP messaging over SSL.
- Public URL and certificates should be configured and enabled.
- Registered with shared Single Sign On.

For the certificates used in a development or testing deployment, use a wildcard certificate. Using a wildcard certificate allows multiple hosts and subdomains within your certificate domain to share the same certificates. For example:

```
*.provider.com
```

During installation, after completing the configuration, copies of the following files must be retained for later use:

- `/opt/vmware/vcloud-director/etc/responses.properties`
- `certificates.ks`

Configure NTP Synchronization in Your Environment

The operating system time should be synchronized between every vSphere Replication appliance in the environment by using an NTP server.

By default, the vSphere Replication appliance is synchronized with the ESXi host on which it resides. You must disable the NTP synchronization with the host and configure the vSphere Replication appliance and the vCenter Server to synchronize with an external NTP server.

Procedure

1 Configure NTP synchronization on the vSphere Replication appliance.

   a In the vSphere inventory tree, locate the vSphere Replication appliance, right-click and select **Edit Settings**.

   b On the **VM Options** tab, click **VMware Tools**.

   c Deselect the **Synchronize guest time with host** check box.

2 Configure NTP in the vSphere Replication appliance console:

   If you configured an NTP server during the deployment, you can skip this step.

   a Log in to the vSphere Replication appliance console as the root user.

   b Run the command:

```
yast2 ntp-client add server=your_chosen_time_server
```

   Note This console error can be ignored:

```
Error: Cannot update the dynamic configuration policy.
```

   c Install the NTP client:

```
yast2 ntp-client enable
```

   d Run this command:

```
sntp -P no -r  your_chosen_time_server
```

   e Restart NTP:

```
service ntp restart
```

   f The `ntp.conf` file is updated and the NTP synchronization is configured successfully.

Cassandra Installation and Configuration

Cassandra is used to store metadata and should be configured to support storage of the metadata for replication services.

Prerequisites

For this example CentOS 6.5 was used as the server OS.

Java 1.7.X must be installed before installing Cassandra.

```
# cd /opt
# wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com
%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/7u79-
b15/jdk-7u79-linux-x64.tar.gz"
# tar xzf jdk-7u79-linux-x64.tar.gz
```

Update Security

```
# wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com
%2F; oraclelicense=accept-securebackup-cookie" http://download.oracle.com/otn-
pub/java/jce/7/UnlimitedJCEPolicyJDK7.zip
# unzip UnlimitedJCEPolicyJDK7.zip
# cp UnlimitedJCEPolicy/*.jar /opt/jdk1.7.0_79/jre/lib/security/
```

Install and Configure Java

```
# cd /opt/jdk1.7.0_79/
# alternatives --install /usr/bin/java java /opt/jdk1.7.0_79/bin/java 2
# alternatives --config java
```

Verify the version of Java is installed and active:

```
# java -version
```

Procedure

1 Run the following commands to add the DataStax Community repository and install Cassandra.

   a Create the file /etc/yum.repos.d/datastax.repo. The contents are:

   ```
   [datastax]
   name = DataStax Repo for Apache Cassandra
   baseurl = https://rpm.datastax.com/community
   enabled = 1
   gpgcheck = 0
   ```

   b Install Cassandra

   ```
   # yum install dsc22 cassandra22 -y
   ```

   c Start and verify the newly installed Cassandra.

   ```
   # service cassandra start
   ```

   d Check Cassandra service status:

   ```
   # service cassandra status
   ```

   e Enter Cassandra command line to verify setup:

   ```
   # cqlsh
   ```

   If an error regarding python occurs when executing cqlsh, update Python to Python 2.7:

   ```
   # yum install -y centos-release-SCL
   # yum install -y python27
   # scl enable python27 bash
   # echo "/usr/lib/python2.7/site-packages/"
   > /opt/rh/python27/root/usr/lib/python2.7/site-packages/usrlocal.pth
   ```

2 Modify Cassandra to enable SSL

   Cassandra requires SSL communication between client and node to enable vRCS to communicate with Cassandra.

   a On each node, create a certificate:

   Generate SSL certificate

   ```
   # /opt/jdk1.7.0_79/bin/keytool -keystore /etc/cassandra/conf/.keystore \
   -storepass vmware -validity 365 -storetype JKS -genkey -keyalg RSA \
   -alias ${CASS_NODE} -dname 'cn=${CASS_NODE}, ou=DR2C, o=VMware, c=US' \
   -keypass vmware
   ```

   b Export Cassandra certificate. In `cloud-cassandra-X.pem` the X represents the node number.

   ```
   # /opt/jdk1.7.0_79/bin/keytool -export -rfc \
   -keystore /etc/cassandra/conf/.keystore -storepass vmware \
   -file /root/cloud-${CASS_NODE}.pem -alias ${CASS_NODE}
   ```

c Copy `.pem` files to all other servers

d Import each certificate into `truststore`:

```
# /opt/jdk1.7.0_79/bin/keytool -noprompt -import -trustcacerts \
   -alias ${CASS_NODE} -file /root/cloud-${CASS_NODE}.pem \
   -keystore /etc/cassandra/conf/.truststore -storepass vmware
```

The truststore contains a copy of all the nodes pem certificate.

3 Modify Cassandra to enable SSL

a Enable client communication with Cassandra over SSL by editing: /etc/cassandra/conf/cassandra.yaml

```
# Comment out listen_address and bind to listen_interface instead
#listen_address: localhost
listen_interface: eth1

# Comment out rpc_address and bind to rpc_interface instead
#rpc_address: localhost
rpc_interface: eth1

# ---------------- Further down in file
server_encryption_options:
   internode_encryption: all
   keystore: /etc/cassandra/conf/.keystore
   keystore_password: vmware
   truststore: /etc/cassandra/conf/.truststore
   truststore_password: vmware
   require_client_auth: true
   store_type: JKS
#-------------------------

# ---------------- Further down in file
Client_encryption_options:
enabled: true
keystore: /etc/cassandra/conf/.keystore
keystore_password: vmware
require_client_auth: true
# Set trustore and truststore_password if require_client_auth is true
truststore: /etc/cassandra/conf/.truststore
truststore_password: vmware
# More advanced defaults below:
# protocol: TLS
# algorithm: SunX509
store_type: JKS
```

b Restart Cassandra

```
# service cassandra restart
```

Service Provider Installation 4

Installation and configuration of the service provider creates the environment where virtual machines are stored for DRaaS

Before starting the process for each component, there should be a suitable vCloud Director environment:

- vCloud Director is installed and configured
- Ensure that the directory `/opt/vmware/vcloud-director/data/transfer` has been shared (through NFS or other means) between the vCloud Director cells. This is required to ensure that OVFs can be uploaded and downloaded correctly.

  If the directory is not shared replication configuration operations may fail intermittently.
- vCloud Director Extensibility (AMQP) is properly configured
- Single Sign-On is configured and set up

This chapter includes the following topics:

Shell Shortcuts During Installation

Installation and configuration of the service provider is made easier by using shell shortcuts and variables to replace commonly used values

Prerequisites

Some of the commands in this manual use BASH completion to fill in variables. Using variables makes the commands easier to copy-paste if the variables are correctly declared before starting the installation.

Ensure you are familiar process by trying this example:

```
# SSO_URL=10.0.0.5
# SSO_PORT=443
# echo ${SSO_URL}:${SSO_PORT}
10.0.0.5:443
```

Procedure

- ◆ In order for commands to work, you must define all of the environment variables. You can use the following list to fill in the correct values as needed. Copy and paste this section to a text editor. Update the addresses, user names and passwords where appropriate. The installation steps tell you when to copy and paste the contents to a terminal.

```
SSO_URL=<IP address or FQDN for the SSO server>
SSO_USER=administrator@vsphere.local
SSO_PASSWORD=<your password>
RESOURCE_VC_IP=<IP address or FQDN for the Tenant Resource vCenter server>

MGMT_VC_IP=<IP address or FQDN for the Management vCenter server>
MGMT_VC_USER=administrator@vsphere.local
MGMT_VC_PASSWORD=<your password>

VCD_IP=<IP address or FQDN for the vCloud Director>
SITE_NAME=<Name for this vCD cell>
VRMS_IP=<IP address or FQDN for the vRMS VM>
AMQP_IP=<IP address or FQDN for the RabbitMQ cluster>

# DO NOT EDIT BELOW THIS LINE
SSO_THUMBPRINT=`openssl s_client -connect $SSO_URL:443 \
    -tls1 -verify 0 < /dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout | grep Fingerprint | head -n1 | awk -F= '{print $2}'`

RESOURCE_VC_THUMBPRINT=`openssl s_client -connect $RESOURCE_VC_IP:443 \
    -tls1 -verify 0 < /dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout | grep Fingerprint | head -n1 | awk -F= '{print $2}'`

MGMT_VC_THUMBPRINT=`openssl s_client -connect $MGMT_VC_IP:443 \
    -tls1 -verify 0 < /dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout | grep Fingerprint | head -n1 | awk -F= '{print $2}'`

VRMS_THUMBPRINT=`openssl s_client -connect $VRMS_IP:5480 \
    -tls1 -verify 0 </dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout | grep Fingerprint | head -n1 | awk -F= '{print $2}'`

AMQP_THUMBPRINT=`openssl s_client -connect $AMQP_IP:5671 </dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout | grep Fingerprint | head -n1 | awk -F= '{print $2}'`
```

With the shortcuts in place, configuration and installation of the remainder of components should be more straightforward as the information available with fewer steps.

What to do next

Install and configure the individual components of the DRaaS service.

Cloud Proxy

Cloud Proxy provides the endpoints used for replicating data into the service provider deployment. Installation requires configuration of vCloud Director and network interfaces.

Prerequisites

The vCloud Director Cloud Proxy cells are configured the same way as your existing vCloud Director cells with a few modifications. For a proof of concept, the Cloud Proxy can be run on a single vCloud Director cell instance.

- Be sure all vCloud Director cells have proper FQDN.
- NTP is configured
- OpenSSL version used in the guest OS of vCloud Director cell must be 1.0.1e-30 or later.

Procedure

1 Pre-installation

  a Copy installation file, configuration file and certificates file to new vCD Cloud Proxy Cell.

  - Install File: `vmware-vcloud-director-X.X.X-YYYY.bin`
  - Configuration File: `/opt/vmware/vcloud-director/etc/responses.properties`
  - Certificate File: `certificates.ks` This is located where you generate it.

  b Mount shared NFS storage.

  Make sure you have mounted the shared NFS storage to your vCD/Cloud Proxy `/opt/vmware/vcloud-director/data/transfer`

  c vCD Cloud Proxy Second Network Interface

  The vCD installation requires a second NIC to be present, but the cloud proxy does not utilize this. If you have already provisioned your VM with a second NIC you can set the IP address to a single CIDR address, for example 192.168.254.254/32, and you do not need to configure the alias NIC.

  d If required, set up an alias NIC:

```
# ifconfig interface:cons 192.168.254.254 up
```

2 Install

Run the vCloud Director install script: `vmware-vcloud-director-X.X.X-YYYY.bin`

- Do not run the configuration
- Do not start the `vmware-vcd` service

3 Configure

Use the `responses.properties` file to configure the vCloud Director cell, make sure you do not start the `vmware-vcd` service:

```
# /opt/vmware/vcloud-director/bin/configure -r \
  /opt/vmware/vcloud-director/data/transfer/responses.properties
```

4 Specialize a vCloud Director cell to become a dedicated Cloud Proxy cell

Edit `/opt/vmware/vcloud-director/etc/global.properties`:

Add the following property:

```
com.vmware.cell.runtime.application=com.vmware.vcloud.cloud-proxy-
server.cloudProxyApplication
```

5 Second NIC

The second NIC or alias that you used for the install is no longer required. You can safely turn off the interface.

```
ifconfig interface:cons down
```

6 Start the vCD service

```
service vmware-vcd start
```

7 Modify Cloud Proxy address

If you are running separate Cloud Proxy instances you must change the address for the Cloud Proxy server.

To see the currently configured Cloud Proxy address:

```
GET
Header: Accept: application/*+xml;version=6.0
Content-Type: application/vnd.vmware.vcloud.hybridSettings+xml
URL: https://{VCLOUD_URL}/api/admin/hybrid/settings
The return will contain:
<CloudProxyBaseUri>wss://example.vmware.com/socket/cloudProxy</CloudProxyBaseUri>
<CloudProxyFromCloudTunnelHost>
example.vmware.com
</CloudProxyFromCloudTunnelHost>
```

CloudProxyBaseUri The address that your tenants connect to send data.

CloudProxyFromCloudTunnelHost The address that your ESXi hosts use to communicate back to the tenants.

To modify the addresses in use a submission using the web services API must be used. For example:

```
PUT
Header: Accept: application/*+xml;version=6.0
Content-Type: application/vnd.vmware.vcloud.hybridSettings+xml
URL: https://{VCLOUD_URL}/api/admin/hybrid/settings
Body:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:HybridSettings xmlns="http://www.vmware.com/vcloud/versions"
xmlns:ns2="http://www.vmware.com/vcloud/v1.5">
    <ns2:Link rel="edit" href="https://{VCLOUD_URL}/api/admin/hybrid/settings"
type="application/vnd.vmware.vcloud.hybridSettings+xml"/>
    <ns2:CloudProxyBaseUriOverride>wss://{CLOUD_PROXY_URL}:
443/socket/cloudProxy</ns2:CloudProxyBaseUriOverride>

<ns2:CloudProxyFromCloudTunnelHostOverride>{REVERSE_CLOUD_PROXY_URL}</ns2:CloudProxyFromCloud
TunnelHostOverride>
</ns2:HybridSettings>
```

vRMS

vRMS manages vSphere Replication for each tenant and must be installed to oversee replication.

Prerequisites

Currently the VAMI interface only supports configuration in the same vCenter Server. Register the command line utilities to separate management and tenant resources in vCenter Server.

Procedure

1 Deploy the `vCloud_Availability_4vCD_OVF10.ovf` appliance to your management vCenter Server Web UI. During the deployment you can configure vCPU count, password and NTP servers.

2 Check that SSH is enabled and NTP is correctly configured.

3 Register the vRMS with the resource backing vCenter. Generate the UUID which is required in subsequent commands:

```
# uuidgen > vRMS-UUID.txt
# VRMS_UUID=`cat vRMS-UUID.txt`
```

4 Obtain the `HMS_KEYSTORE_PASSWORD`:

```
# grep -i hms-keystore-password /opt/vmware/hms/conf/hms-configuration.xml
<hms-keystore-password>2YLAtvOupK67LQ87</hms-keystore-password>
# HMS_KEYSTORE_PASSWORD="2YLAtvOupK67LQ87"
```

5 Create a vSphere Replication solution user in shared SSO instance:

```
# /usr/bin/sudo -u vrms /opt/vmware/hms/bin/va-util.sh \
  --cmd createsolutionuser \
  --overwrite true \
  --ls https://${SSO_URL}/lookupservice/sdk \
  --lsthumbprint ${SSO_THUMBPRINT} \
  --user ${SSO_USER} \
  --pass ${SSO_PASSWORD} \
  --vc ${RESOURCE_VC_IP} \
  --vcport 80 \
  --vcthumbprint ${RESOURCE_VC_THUMBPRINT} \
  --username com.vmware.vr-${VRMS_UUID} \
  --description "vSphere Replication vRMS Solution User - ${VRMS_UUID} " \
  --keystore /opt/vmware/hms/security/hms-keystore.jks \
  --keystorepass ${HMS_KEYSTORE_PASSWORD} \
  --keystorealias jetty
```

6 Register vRMS with Lookup Service.

```
# /usr/bin/sudo -u vrms /opt/vmware/hms/bin/va-util.sh \
  --cmd existingserviceregistration \
  --ls https://${SSO_URL}/lookupservice/sdk \
  --lsthumbprint ${SSO_THUMBPRINT} \
  --user ${SSO_USER} --pass ${SSO_PASSWORD} \
  --vc ${RESOURCE_VC_IP} \
  --vcport 80 \
  --vcthumbprint ${RESOURCE_VC_THUMBPRINT} \
  --serviceattr {} \
  --keystore /opt/vmware/hms/security/hms-keystore.jks \
  --keystorepass ${HMS_KEYSTORE_PASSWORD} \
  --keystorealias jetty \
  --hmsserveruuid ${VRMS_UUID} \
  --hmsaddress ${VRMS_IP} \
  --hmsvmomiport 8043
```

7 Re-Configure vRMS so that it points to SSO.

```
# /opt/vmware/hms/bin/hms-configtool \
  -cmd reconfig \
  -configfile /opt/vmware/hms/conf/hms-configuration.xml \
  -property hms-ls-url=https://${SSO_URL}/lookupservice/sdk \
  -property hms-ls-thumbprint=${SSO_THUMBPRINT} \
  -property hms-solution-username=com.vmware.vr-${VRMS_UUID}
```

Output should be:

```
New configuration successfully written to file /opt/vmware/hms/conf/hms-configuration.xml
Unable to configure HMS: null
```

8 Register vRMS with resource VC.

```
# /usr/bin/sudo -u vrms /opt/vmware/hms/bin/va-util.sh \
  -cmd certauth \
  --user ${SSO_USER} \
  --pass ${SSO_PASSWORD} \
  --host ${RESOURCE_VC_IP} \
  --port 80 \
  --thumbprint ${RESOURCE_VC_THUMBPRINT} \
  --keystore /opt/vmware/hms/security/hms-keystore.jks \
  --keystorepass ${HMS_KEYSTORE_PASSWORD} \
  --keystorealias jetty \
  --extkey com.vmware.vcHms
```

9 Reconfigure vRMS to use the resource vCenter Server.

```
# /opt/vmware/hms/bin/hms-configtool \
  -cmd reconfig \
  -configfile /opt/vmware/hms/conf/hms-configuration.xml \
  -property hms-sitename="${SITE_NAME}" \
  -property hms-localvc-address=${RESOURCE_VC_IP} \
  -property hms-localvc-thumbprint=${RESOURCE_VC_THUMBPRINT} \
  -property hms-address=${VRMS_IP} \
  -property javax-persistence-database=pgsql \
  -property javax-persistence-jdbc-url=jdbc:postgresql:vrmsdb \
  -property hms-db-user=vrmsdb \
  -property hms-db-password=dW51c2Vk \
  -property hms-localvc-adminmail=CHANGEME@example.com
```

10 Run `config` command

```
# /opt/vmware/hms/bin/hms-configtool --cmd config
```

11 Edit `/opt/vmware/hms/hms-configuration.xml` and update the UUID with the old one as shown. The configuration command generated a new UUID, but the old UUID is the one registered to SSO.

```
# vi /opt/vmware/hms/conf/hms-configuration.xml
```

12 Update the following line:

```
<hms-server-uuid>$VRMS_UUID</hms-server-uuid>
```

Also change:

```
<hms-embedded-hbr>true</hms-embedded-hbr>
```

To

```
<hms-embedded-hbr>false</hms-embedded-hbr>
```

13 Run Python script to generate /opt/vmware/hms/conf/extension.xml

Go to Python Prompt and run two commands:

```
# cd /opt/vmware/share/htdocs/service/hms/cgi
# python
>>> import commands
>>> commands.RecreateExtensionXmlCommand().run()
```

14 Update Postgres Embedded DB

```
# /opt/vmware/vpostgres/current/bin/psql -U vrmsdb <<EOF
update localserverentity set sitename = '$SITE_NAME';
EOF
```

15 Restart vRMS service

```
# service hms restart
```

16 Verify vRMS started properly by watching the log

```
# tail -f /opt/vmware/hms/logs/hms.log
```

17 Disable the embedded vRS service

```
# service hbrsrv stop
# chkconfig hbrsrv off
```

vRCS

vRCS components provide the API management interface for configuring the service

Prerequisites

For developer and test environments, you can use one vRCS appliance with 2 vCPU. For production deployments use three configured with 4 vCPUs each.

Procedure

1 Deploy the vCloud_Availability_4vCD_Cloud_Service_OVF10.ovf appliance to your tenant management vCenter Server Web UI. During the deployment, you can configure vCPU count, password, and NTP servers.

After deployment, a message on a blue screen to manage the vSphere Replication Cloud Service VM at: https://XX.XX.XX.XX:5480/ appears. You can use the VAMI interface to change some settings like password and network, but the configuration of the appliance requires the command line.

2 Access the vRCS appliance remote console using root and the password that you configured during the deployment. Note there is no default password; this must be configured through the vCenter deployment.

Make sure you have enabled SSH and properly configured NTP. See Appendix D for more information.

Update the vRCS configuration file

3 Generate UUID which is required in subsequent commands.

```
# uuidgen > vRCS-UUID.txt
# VRCS_UUID=`cat vRCS-UUID.txt`
```

4 Obtain the HMS_KEYSTORE_PASSWORD:

```
# grep -i hms-keystore-password /opt/vmware/hms/conf/hms-configuration.xml
   <hms-keystore-password>2YLAtvOupK67LQ87</hms-keystore-password>


#  HMS_KEYSTORE_PASSWORD="2YLAtvOupK67LQ87"
```

5 Check the configuration variables are correct:

```
# echo $VRCS_UUID
# echo $SSO_URL
# echo $SSO_THUMBPRINT
# echo $AMQP_IP
# echo $AMQP_THUMBPRINT
```

6 Open the configuration file `/opt/vmware/hms/conf/hcs-config.properties` and set the UUID and other settings as following:

```
hcs.uuid=<UUID that was generated>
ls.url=https://<SSO_URL>:7444/lookupservice/sdk
ls.thumbprint=<SSO_THUMBPRINT>
amqp.host=<AMQP_IP>
amqp.port=5671
amqp.user=<USERNAME>
amqp.pass=<PASSWORD>
amqp.vHost=/
amqp.extension.exchange=<Exchange name used>
amqp.notification.exchange=<Exchange name used>
amqp.tunnelingApp.exchange=<Exchange name used>
amqp.thumbprint=<AMQP_THUMBPRINT>
cassandra.replication.factor = <Replication factor - 3 for production>
```

7 Configure vRCS solution users by creating a list file of resource for vCenters in `/root/vc-list.txt`

The vc-list file contains one line for each resource vCenter Server that you want to use. vCenter supports up to 10.

From the command line run:

```
# rm /root/vc-list.txt

# echo "${RESOURCE_VC_IP}:80 | ${RESOURCE_VC_THUMBPRINT}" >> /root/vc-list.txt

# /opt/vmware/hms/bin/va-util.sh --cmd createsolutionuser --overwrite true \
--ls https://${SSO_URL}/lookupservice/sdk --lsthumbprint ${SSO_THUMBPRINT} \
--user ${SSO_USER} --pass ${SSO_PASSWORD} --username com.vmware.vr-${VRCS_UUID} \
--description "vSphere Replication vRCS Solution User - ${VRCS_UUID}" \
--keystore /opt/vmware/hms/security/hms-keystore.jks \
--keystorepass ${HMS_KEYSTORE_PASSWORD} --keystorealias jetty --vclist /root/vc-list.txt
```

Note The option switch of `--overwrite true` is required if previous attempt fails.

8 Import vRCS solution users into vCD

The import of the vRCS solution users can be done through the UI or API.

You should import the user as `com.vmware.vr-<hcs-uuid>@vsphere.local`.

a Log in to vCD Select **Administration** tab

b Select **Users** from column on left

c Press the **Import Users** button

d Copy / Paste the username `com.vmware.vr-uuid@vsphere.local`. Make sure to include the `@vsphere.local` or an alternative local domain that matches your own domain or deployment location, for example `@newyork.local`.

9 Export vRCS certificate from all vRCS nodes and import certificates into all Cassandra Nodes.

Log in to the Cassandra server and import the certificate:

```
# VRCS_NODE=10.158.15.194
# openssl s_client -connect ${VRCS_NODE}:5480 -tls1 </dev/null 2>/dev/null \
    | openssl x509 > /root/hcs1.crt
# /opt/vmware/vcloud-director/jre/bin/keytool -noprompt -import \
          -trustcacerts -alias cloud-${VRCS_NODE} \
          -file /root/hcs1.crt \
          -keystore /opt/apache-cassandra/conf/.truststore \
          -storepass vmware
```

10 Restart the Cassandra service.

11 Register Cassandra in SSO lookup service.

Register each node:

```
# CASS_NODE=<name used to identify Cassandra node>
# CASS_PEM=`openssl s_client -connect ${CASS_NODE}:9042 \    -tls1 </dev/null 2>/dev/null|
openssl x509`
# /usr/bin/sudo -u vrms /opt/vmware/hms/bin/va-util.sh \    --cmd regcassandra --ls https://$
{SSO_URL}/lookupservice/sdk \    --lsthumbprint ${SSO_THUMBPRINT} --user ${SSO_USER} --pass $
{SSO_PASSWORD} \    --address ${CASS_NODE} --port 9042 --pem "${CASS_PEM}"
```

From the vCenter Server instance that provides the SSO functionality, you can list and unregister SSO solutions.

Log in to server:

```
# ssh root@${SSO_VC_SERVER}
# shell.set -enabled True
# shell
# cd /usr/lib/vmidentity/tools/scripts/
```

List SSO:

```
# python lstool.py list --url https://localhost/lookupservice/sdk
--no-check-cert
```

Unregister

```
# python lstool.py unregister --url
https://localhost/lookupservice/sdk \
  --user ${SSO_USER} --password ${SSO_PASSWORD} --no-check-cert --
id ${SOLUTION_ID}
```

Note The SOLUTION_ID can be obtained from the list command.

12 Set vRCS to start

```
# chkconfig hcs on
```

Reboot the server

```
# reboot
```

13 You can also watch the log to make sure it starts properly:

```
# tail -f /opt/vmware/hms/logs/hcs.log
```

14 Enable Replication Rights in vCD

In order to replicate VMs the roles that you set up for your tenant, organizations must have the correct replication rights added.

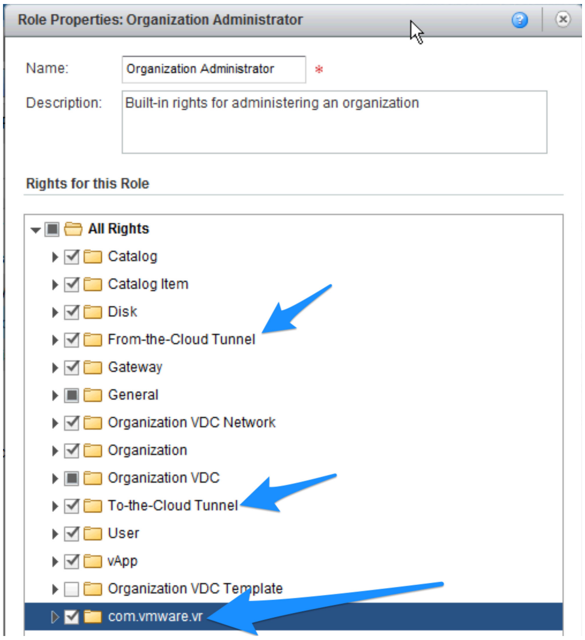
From vCD Administration UI

a Select Roles

b Select Role that you would like to add Replication to (Example: Organization Administrator)

c Add the following rights:

- com.vmware.vr
- From-the-Cloud Tunnel
- To-the-Cloud Tunnel



Role Properties: Organization Administrator

Name: Organization Administrator *

Description: Built-in rights for administering an organization

Rights for this Role

- ▣ All Rights
  - ☑ Catalog
  - ☑ Catalog Item
  - ☑ Disk
  - ☑ From-the-Cloud Tunnel
  - ☑ Gateway
  - ▣ General
  - ☑ Organization VDC Network
  - ☑ Organization
  - ▣ Organization VDC
  - ☑ To-the-Cloud Tunnel
  - ☑ User
  - ☑ vApp
  - □ Organization VDC Template
  - ☑ com.vmware.vr

vRS

vRS handles the replication process for each protected virtual machine, handling the ongoing replication from vSphere to the service provider site

For POC environments, you can use one vRCS appliance with 2 vCPU to test the deployment. For production deployments use at least three vRS units configured with 4 vCPU each to handle the replication load from multiple tenant VMs.

Deploy the `vCloud_Availability_4vCD_AddOn_OVF10.ovf` appliance to your management vCenter Server Web UI. Make sure that you have enabled SSH and properly configured NTP.

Procedure

1 Configure

  a Add the vRMS thumbprint to vRS

  SSH to the vRS instance to add the thumbprint:

```
# hbrsrv-guestinfo.sh set guestinfo.hbr.hms-thumbprint ${VRMS_THUMBPRINT}
```

  Note Make sure that the thumbprint is captured and inserted by using uppercase only.

  b Restart the vRS service:

```
# service hbrsrv restart
```

  c Verify the vRMS thumbprint:

```
# hbrsrv-guestinfo.sh get guestinfo.hbr.hms-thumbprint
```

  d Get the vRS thumbprint and MoRef ID. The thumbprint and MoRef ID are needed in future steps to complete the configuration.

```
# hbrsrv-guestinfo.sh get guestinfo.hbr.hbrsrv-thumbprint
```

  The MoRef ID can be obtained from the vRS VM configuration.

```
# grep vCenterId /opt/vmware/etc/vami/ovfEnv.xml
```

  Example MoRef ID fingerprint:

```
vm-107
```

  Note The MoRef ID can also be found by browsing the Management vCenter MOB: `https://vCenter-ip/mob`

  Log in and browse to:

  Go to **Content > rootFolder (group-d1) > childEntity (datacenter-21) > vmFolder > locate vRS VM display name** and obtain the VM ID (vm-107).

  e Remove the OVF password from vRS server. Execute `va-util.sh` tool to remove password

```
# MOREF_ID=vm-107
# sudo -u vrms /opt/vmware/hms/bin/va-util.sh -cmd droppasswd \
    -user ${MGMT_VC_USER} -pass ${MGMT_VC_PASSWORD} \
    -host ${MGMT_VC_IP} -port 80 -thumbprint ${MGMT_VC_THUMBPRINT} -vmid ${MOREF_ID}
```

  Example output:

```
# VM reconfiguration result: success
```

  f Update vRS server ulimit by editing `/etc/security/limits.conf` and appending the following two lines:

```
* soft nofile 65535
* hard nofile 65535
```

  Edit `/etc/pam.d/login` to include the following line:

```
session required pam_limits.so
```

  Reboot vRS server

```
# reboot
```

2 Get the vRS thumbprint:

```
# VRS_THUMBPRINT=`hbrsrv-guestinfo.sh get guestinfo.hbr.hbrsrv-thumbprint`

# echo $VRS_THUMBPRINT
```

3 Register vRS to vRCS via the web services API

Get the vimServerID:

```
GET  https://{VCD_URL}:443/api/admin/extension/vimServerReferences

POST
URL: https://{VCD_URL}:/api/admin/extension/vimServer/{vimServerID}/action/registerVrServer
Accept: application/*+xml;version=6.0;vr-version=3.0
Content-Type :  application/vnd.vmware.hcs.registerVrServerParams+xml

BODY:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ManageVrServerParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0">
    <ns2:VrThumbprint>{VRS_THUMBPRINT}</ns2:VrThumbprint>
    <ns2:VrManagementURI>https://{VRS_IP}:8123</ns2:VrManagementURI>
<ns2:VrTrafficPort>31031</ns2:VrTrafficPort>
</ns2:ManageVrServerParams>
```

Tenant Installation 5

Client configuration relies on the configuration of vSphere Replication within the tenant environment

You can find information about how to deploy and configure the vSphere Replication Appliance in the vSphere Replication documentation. Once deployed you will need to set up a Cloud Provider to replicate VMs.

This chapter includes the following topics:

Enable Replication on vCloud Director

Tenant setup configures the tenant side of the replication configuration.

In order to enable replication, you must enable replication on the tenant organization in vCloud Director. In this example we will be enabling replication for an organization named org1.

1. Get list of organizations; find org1 to get the link to org1's details.

   ```
   GET with https://{VCD_URL}/api/org
   ```

2. Use the link to find the org1 VDC:

   ```
   GET with https://{VCD_URL}/api/org/db391d18-46b8-4ba1-8e3d-df5f3ea1cc8a
   ```

3. Use the VDC link to get the details about the VDC:

   ```
   GET with https:// {VCD_URL}/api/vdc/88540577-739f-4f06-9212-88d20be9d198
   ```

   Find `rel="operation:enableReplication" href="`
   `https://{VCD_URL}/api/vdc/88540577-739f-4f06-9212-88d20be9d198/action/enableReplication"`

4. Enable replication on the VDC:

   ```
   POST with
   https://{VCD_URL}/api/vdc/88540577-739f-4f06-9212-88d20be9d198/action/enableReplication
   ```

Make sure you do a GET call on the task that you get back to make sure that the action is successful.

```
https://{VCD_URL}/api/task/{TASK_ID}
```

You can also check to see what VDCs are enabled for replication by doing a GET call to:

```
https://{VCD_URL}/api/org/{ORG_UUID}/enabledForReplicationVdcs
```

Configure Cloud Provider

The Cloud Provider should be configured to assign the correct service provider destination for replication.

Prerequisites

Open the vCenter Server administration interface.

Procedure

1. Open the vCenter Server by using the Web Client.
2. Navigate to the Connect to a Cloud Provider.
3. In the Manage tab, click **`vSphere Replication`**.
4. Click **Target Sites** and click the **Connect to Cloud Provider** icon.
5. **Connect to Cloud Provider**
   a. Enter the Cloud Provider Address: `vcd.provider.com` without the `/cloud` suffix
   b. Organization Name
   c. Username/Password that has the Replication Rights
6. Once configured right click and select **Configure Target Networks**
7. Enter **`vcd.provider.com`** in the **Cloud Provider Address** text box and an organization name in the **Organization name** text box.
8. Enter the username and password of a user that has replication rights.
9. Right click the target site and select **Configure Target Networks**.

Configure Replication

Configure the Replication targets for the tenant service that you want to include in the DRaaS.

Replication must be configured for each virtual machine that needs to be replicated.

Procedure

1. To configure replication, right-click on a VM and select **All vSphere Replication Actions > Configure Replication**
2. Select **Replicate to a cloud provider** and configure the options.

What to do next

Replications can be monitored from **vCenter > Monitor > vSphere Replication > Outgoing Replications**

Using a Self-Signed Certificate in a Development Environment

Using a self-signed certificate in the tenant configuration ensures security and encryption for tenant deployments

To use a self-signed certificate for your vCloud Director and Cloud Proxy, you must enable the certificate you create within the vSphere Replication Appliance.

Procedure

1 Copy the self-signed certificate to the client vSphere Replication Appliance and load it into the `keystore`.

   a Log in to vSphere Replication Appliance via remote console

   b Export the vCloud Director certificate and import it into the Java `keystore`:

   ```
   # openssl s_client -connect $VCD_IP:443 -tls1 </dev/null 2>/dev/null \    | openssl x509
   > /tmp/vcloud.pem
   # /usr/java/default/bin/keytool -noprompt \
                   -import -trustcacerts \
                   -alias vcloud \
                   -file /tmp/vcloud.pem \
                   -keystore /usr/java/default/lib/security/cacerts \
                   -storepass changeit
   ```

2 Restart `vcta` and `hms`:

```
# service hms restart
# service vmware-vcd restart
```

Diagnostic Information

6

Getting diagnostic information for vCloud Availability for vCloud Director deployments requires careful collection of the logs from each component

Because there are a number of different components on both the service provider and tenant installations, troubleshooting relies on careful investigation of the configuration and logs generated by each component. These then need to be examine in tandem to identify the errors.

This chapter includes the following topics:

Service Provider Diagnostics

The service provider deployment has multiple components. Some can generate diagnostic information which can be used when troubleshooting, other components must have this information collected manually

Due to the large number of different components used to support the vCloud Availability for vCloud Director deployment, logs must be collected from all the different systems. If the problem is affecting only one instance or component, the number of logs collected can be reduced.

Log File Locations

To collect diagnostic information, logs from all the following locations on all instances and appliances should be collected to help diagnose the problem. The table below lists the log locations for all the components.

Table 6-1. Log File Location for Service Provider Components

Component	Log Location
vCenter Server 5.x and earlier versions on Windows XP, 2000, 2003	`%ALLUSERSPROFILE%\Application Data\VMware\VMware VirtualCenter\Logs\\ >`
vCenter Server 5.x and earlier versions on Windows Vista, 7, 2008	`C:\ProgramData\VMware\VMware VirtualCenter\Logs\`
vSphere Replication Appliance	`/var/log/vmware/vpxd/`
vCD/Cloud Proxy	`/opt/vmware/vcloud-director/logs/vcloud-container-debug.log`
vRCS/HCS	`/opt/vmware/hms/logs`
vRMS/HMS	`/opt/vmware/hms/logs`, `/opt/vmware/var/log/lighttpd/error.log`, `/opt/vmware/etc/vami/ovfEnv.xml`, `/var/log/boot.omsg`, `/var/log/boot.msg`

Table 6-1. Log File Location for Service Provider Components (Continued)

Component	Log Location
vRS/HBR	`/var/log/vmware/`
hostd	`/var/run/log/hostd.log`
vmkernel	`/var/run/log/vmkernel.log`

vSphere Replication Diagnostics

The vSphere Replication server can generate a diagnostic bundle that can be used by GSS to diagnose replication issues. You can generate the diagnostic bundle by opening the vSphere Replication VAMI, selecting the **VRM** tab and clicking **Support**.

Click the **Generate** button and a `.zip` package will be created containing the logs.

vRCS Diagnostics

You can generate the vRCS support bundle by running the command:

```
# /opt/vmware/hms/bin/generatesupportbundle.sh
```

The output is located at: `/opt/vmware/hms/supportfolder`.

Find Job ID in vCD Org log for Configure Replication task. This can then be traced into the vRCS logs.

Tenant Diagnostics

The tenant side diagnostics require capturing the log content from the main appliance and vSphere components

Within a tenant environment it is important to collect the information from the vSphere Replication and the vSphere Replication Appliance components. If the problem exists within other systems, collect the logs from all the relevant components at the same time to ensure that the errors can be correlated correctly.

Log File Locations

To collect diagnostic information, logs from all the following locations on all instances and appliances should be collected to help diagnose the problem. The table below lists the log locations for all the components.

Table 6-2. Log File Location for Tenant Components

Component	Log Location
vCenter Server 5.x and earlier versions on Windows XP, 2000, 2003	`%ALLUSERSPROFILE%\Application Data\VMware\VMware VirtualCenter\Logs\\ >`
vCenter Server 5.x and earlier versions on Windows Vista, 7, 2008	`C:\ProgramData\VMware\VMware VirtualCenter\Logs\`
vSphere Replication Appliance	`/opt/vmware/hms/logs`
vCTA	`/opt/vmware/vcta/logs`
vRMS/HMS	`/opt/vmware/hms/logs`, `/opt/vmware/var/log/lighttpd/error.log`, `/opt/vmware/etc/vami/ovfEnv.xml`, `/var/log/boot.omsg`, `/var/log/boot.msg`
vRS/HBR	`/var/log/vmware/`
hostd	`/var/run/log/hostd.log`
vmkernel	`/var/run/log/vmkernel.log`

API Reference 7

The API provides access to the failover, test, and recovery processes.

The API provides access to the vCloud Availability for vCloud Director service including failover and failback, enabling the operations to be scripted or controlled through an external process. Using the API interface operations can be integrated with other services and systems, including integration into web and command-line control systems.

This chapter includes the following topics:

API Workflow

Configuration and recovery of protected virtual machines uses a fixed process through the API service

To configure and recover virtual machines protected by the service, perform the following tasks in vSphere Replication and service provider environment:

1. Using the vSphere Replication Appliance, specifically the vCenter Server Web Client UI extension, replicate the virtual machines that you plan to protect from your source site to the Service Provider.

   You must initiate replication to the cloud by using vSphere Replication at your source site because replication between your source site and the cloud is not symmetrical. You can initiate replication to the cloud from your source site but, for security reasons, you cannot communicate with the virtual machines at your source site from the cloud.

2. After replicating your virtual machines to the cloud, call the APIs to list the replications.
3. Using API calls test recovery for a virtual machine and clean up the test after you run it.
4. In the event that your source site becomes unavailable, recover your virtual machines by using failback API calls.

vCloud Director for Connection & Authentication

Before starting management through the API, a suitable connection and authentication process must be followed, and then the generated token must be used for further calls.

Table 7-1. vCloud Director Operations for connection and Authentication

Operation	Description	Headers
GET /api/versions	Every cloud has a login URL that a client can obtain by making an unauthenticated `GET` request to the vCloud Director API/versions URL. The response to this request also lists vCloud API versions that the server supports	
POST /api/sessions	Authenticates a user and creates a Session object that contains the URLs from which that user can begin browsing. Users who authenticate to the integrated identity provider use basic HTTP authentication. If the request is successful, the server returns HTTP response code 200 (OK) and headers that include an authorization header of the form: `x-vcloud-authorization: token` This header must be included in each subsequent vCloud API request.	Authorization: Basic encoded-credentials Accept: `application/*+xml;version=5.5` Supply credentials like: `user@organization:password`

vRS Registration Example

Registering a vRS instance into a vCloud Availability for vCloud Director deployment uses multiple steps and calls to the web service API.

The steps below show the sample process using a combination of the `curl` tool and shell variables to compose the required requests to register a new vRS instance. The same process can be used for other processes and steps during the installation and deployment.

Prerequisites

When using the web services API, any tool that can set and request the required information can used. In the example below, the command-line tool `curl`. The process requires multiple steps, first to obtain the authorization, then the registration URL, and finally the call that performs the registration.

Procedure

1 Create some variables that will be used in the rest of the registration process:

```
VCD_IP=<IP address or FQDN for the vCD host>
VRS_IP=<IP address or FQDN for the VRS host>
VRS_THUMBPRINT=`openssl s_client -connect $VRS_IP:5480 \
    -tls1 -verify 0 </dev/null 2>/dev/null | \
openssl x509 -fingerprint -noout | grep Fingerprint | \
head -n1 | \awk -F= '{print $2}'`
```

This step configures the IP address of the vRS instance, and the fingerprint required to access the information.

2. Generate variables to be used to hold information for the first access to the API:

   ```
   CONTENT="regVRS-content.txt"
   HEADERS="regVRS-headers.txt"
   ACCEPT='Accept: application/*+xml;version=5.6'
   USER='root@system'
   PASS='ca$hc0w'
   ```

   The `$CONTENT` variable is a file that will the information returned. The `$HEADERS` is the header material used to supply authentication and supported returned types.

3. Authenticate with the vCD API by using `curl` with the previously set variables.

   ```
   $ curl -k -o "$CONTENT" -D "$HEADERS" -X POST --user "$USER:$PASS" -H "$ACCEPT" "https://$
   {VCD_IP}/api/sessions"
   ```

   The information returned from this call will be placed into the two files reference by the variables. One containing the headers, and the other the body.

4. You can confirm that the process completed successfully by checking the content of the header file:

   ```
   $ head -n1 $HEADERS
   ```

   The returned header should contain a successful HTTP result code `200`, for example `HTTP/1.1 200 OK`.

5. Extract the authorization code from the returned header information. The code must be provided in future requests to authenticate the operations.

   ```
   $ grep x-vcloud-authorization "$HEADERS" | awk -F : '{print $2}' | tr -d ' '
   ```

6. Create a variable containing the return authorization code:

   ```
   $ VCD_COOKIE=93f2f3f0c07a4355b3466812ddf9987e
   ```

7. Obtain the URL for the VIM server by submitting another curl request, using the authorization code:

   ```
   $ curl -k -o "$CONTENT" -D "$HEADERS" \
       -H "x-vcloud-authorization: $VCD_COOKIE" \
       -H "$ACCEPT" \
       https://${VCD_IP}/api/admin/extension/vimServerReferences
   ```

   The returned header should contain a successful HTTP result code `200`.

8. Extract the `VimServerReference` from the returned data:

   ```
   $ cat $CONTENT | grep "vmext:VimServerReference" | awk -F\" '{print $2}'
   ```

9. Set the returned value into a variable:

   ```
   $ VIM_URL=https://10.158.12.163/api/admin/extension/vimServer/f88ce1f6-f8f3-489b-9f32-
   fac50b035f2b
   ```

10. Build the request body:

    ```
    ACCEPT='Accept: application/*+xml;version=6.0;vr-version=3.0'
    REGVRS_BODY="<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"yes\"?>
    <ns2:ManageVrServerParams xmlns=\"http://www.vmware.com/vcloud/v1.5\"
    xmlns:ns2=\"http://www.vmware.com/vr/v6.0\">
        <ns2:VrThumbprint>${VRS_THUMBPRINT}</ns2:VrThumbprint>
        <ns2:VrManagementURI>https://${VRS_IP}:8123</ns2:VrManagementURI>
    <ns2:VrTrafficPort>31031</ns2:VrTrafficPort>
    </ns2:ManageVrServerParams>"
    ```

11 Submit the request using the compiled request body:

```
$ curl -k -o "$CONTENT" -D "$HEADERS" \
    -X POST --data-binary "$REGVRS_BODY" \
    -H "x-vcloud-authorization: $VCD_COOKIE" \
    -H "Content-Type: application/vnd.vmware.hcs.registerVrServerParams+xml" \
    -H "$ACCEPT" ${VIM_URL}/action/registerVrServer
```

The returned header should contain a successful HTTP result code `200`.

12 Extract the URL required to perform the registration:

```
$ cat $CONTENT | grep Link | grep "application/vnd.vmware.vcloud.task+xml" | awk -F\"
'{print $4}'
# TASK_URL=https://10.158.12.163/api/task/49e6347e-0382-4843-b494-2eea01e77229
```

13 Submit the final request to perform the registration:

```
$ curl -k -o "$CONTENT" -D "$HEADERS" \
    -H "x-vcloud-authorization: $VCD_COOKIE" -H "$ACCEPT" $TASK_URL
```

The returned header should contain a successful HTTP result code `200`.

What to do next

Examine the content of the returned information and verify that the progress is `100` and that there are no errors listed. The vRS instance should now be registered.

Enabling Replication Example

Enabling replication for a VM a vCloud Availability for vCloud Director deployment uses multiple steps and calls to the web service API.

During the replication enablement process, the steps must be completed by a vCloud Director administrator.

Prerequisites

When using the web services API, any tool that can set and request the required information can used. In the example below, the command-line tool curl. The process requires multiple steps, first to obtain the authorization, then the registration URL, and finally the call that performs the registration.

Procedure

1 Create some variables that will be used in the rest of the process:

```
CONTENT="regOrg-content.txt"
HEADERS="regOrg-headers.txt"
ACCEPT='Accept: application/*+xml;version=5.6'
USER='administrator@System'
PASS='.s3cr3tP@ssw0rd!'
VCD_IP=10.158.12.128
```

2 Authenticate with the vCD API by using `curl` with the previously set variables. The authorization token returned will need to be used in all further commands.

```
# curl -k -o "$CONTENT" -D "$HEADERS" -X POST --user "$USER:$PASS" \
 -H "$ACCEPT" "https://${VCD_IP}/api/sessions"
```

3 Extract the authorization code from the returned header information:

```
# head -n1 $HEADERS
# grep x-vcloud-authorization "$HEADERS" | awk -F : '{print $2}' | tr -d ' '
```

4 Set a variable to contain the authorization code:

```
# VCD_COOKIE=93f2f3f0c07a4355b3466812ddf9987e
```

5 Obtain a list of configured organisations to identify the correct organization where the replication will be enabled:

```
# curl -k -o "$CONTENT" -D "$HEADERS" \
 -H "x-vcloud-authorization: $VCD_COOKIE" \
 -H "$ACCEPT" https://${VCD_IP}/api/org
```

6 Extract the organization:

```
# head -n1 $HEADERS
# cat $CONTENT | grep "application/vnd.vmware.vcloud.org+xml" | awk -F\" '{print $4 ":" $2}'
```

7 Obtain a list of the VDCs within the organization:

```
# curl -k -o "$CONTENT" -D "$HEADERS" -H "x-vcloud-authorization: $VCD_COOKIE" \
  -H "$ACCEPT" https://10.158.12.163/api/org/0be6ca24-5769-413e-b121-2d7290310dcb
# head -n1 $HEADERS
# cat $CONTENT | grep "application/vnd.vmware.vcloud.vdc+xml" | awk -F\" '{print $6 ":" $4}'
```

8 Obtain the registration URL required to enable replication to the selected organization:

```
# curl -k -o "$CONTENT" -D "$HEADERS" -H "x-vcloud-authorization: $VCD_COOKIE" \
  -H "$ACCEPT" https://10.158.12.163/api/vdc/0de76523-a3ec-4b90-878d-007527127ce0
# head -n1 $HEADERS
# cat $CONTENT | grep "enableReplication" | awk -F\" '{print $4}'
```

9 Enable replication by submitting a POST request using the URL retrieved in the previous step.

```
# curl -k -o "$CONTENT" -D "$HEADERS" -X POST -H "x-vcloud-authorization: $VCD_COOKIE" \
  -H "$ACCEPT" \
  https://10.158.12.163/api/vdc/0de76523-a3ec-4b90-878d-007527127ce0/action/enableReplication
# head -n1 $HEADERS
# cat $CONTENT | grep Link | grep "application/vnd.vmware.vcloud.task+xml" | awk -F\"
'{print $4}'
```

10 Confirm that the task completed successfully to enable replication.

```
# curl -k -o "$CONTENT" -D "$HEADERS" -H "x-vcloud-authorization: $VCD_COOKIE" \
  -H "$ACCEPT" https://10.158.12.163/api/task/49e6347e-0382-4843-b494-2eea01e77229
# head -n1 $HEADERS
# cat $CONTENT
```

Replication should now be enabled. If the task fails, look for the error, correct, and try to enable replication again.

REST Reference

With the REST API you can control the vCloud Availability for vCloud Director service through an accessible REST API.

Table 7-2. List of Operations for vCloud Availability for vCloud Director

Operation	Description
`GET /api/vr/versions`	Returns a list of all the supported version types
`GET /api/vdc/{vdc-id}/replications`	Returns a list of URIs for incoming replications for the VDC
`GET /api/vdc/{vdc-id}/peers`	Retrieves all VR Peers that are paired with the specified VDC

Table 7-2. List of Operations for vCloud Availability for vCloud Director (Continued)

Operation	Description
GET /api/vr/peers/{peer-id}/replications	Retrieves a page of incoming ReplicationGroups for the specified VR Peer
GET /api/org/{org-id}/replications	Returns a page with list of URIs for incoming replications for that 'Organization ID'
GET /api/vr/replications/{replication-id}	Returns basic status information for the incoming replication
GET /api/org/{org-id}/enabledForReplicationVdcs	Returns list of references to all VDCs for that 'Organization id' that have been enabled for replication
DELETE /api/vr/replications/{replication-id}	When a DELETErequest to this URL is handled, the incoming replication is destroyed. If the primary site is still available, the outgoing replication there must be manually destroyed
GET /api/vdc/{vdc-id}/recoveryDetails	Retrieves information about the recovery capabilities of this VDC
POST /api/vr/replications/{replication-id}/action/failover	A recover operation is started for the VMs in the replication. A vCloud task reflecting the operation progress is returned. After successful completion of the task, the VMs are brought to bootable state at the placeholder vApp
POST /api/vr/replications/{replication-id}/action/testFailover	A test bubble image creation is started for the VMs in the replication. Ongoing replication is not affected.
POST /api/vr/replications/{replication-id}/action/testCleanup	Test bubble image clean-up is started. A URI to a VCD task to monitor the progress of the clean-up is returned.
POST /api/vr/replications/{replication-id}/action/sync	Perform sync on an on-prem to cloud replication group
POST /api/vr/replications/{replicationGroupId }/action/powerOffVm	Power Offs the protected on-prem VM through REST API
POST /api/vr/{replication-id}/action/reverse	Reverses the replication
GET /api/vr/failbackreplications/{replication-id}	Returns the cloud to on-prem replication group with the given {replication-id}
GET /api/org/{org-id}/failbackreplications	List all the replications available from vCD to on-prem
POST /api/vr/failbackreplications/{failback-id} /action/failover	Recover a VM replicated from the Cloud to the on-prem VC
POST /api/vr/failbackreplications/{replicationGroupId}/action/testFailover	Test failover replication If the replication is already in the process of getting test-recovered the returned task will fail with OngoingFailover vendor-specific error code. If the replication is already test-recovered the returned task will fail with AlreadyTestRecovered vendor-specific error code.
POST /api/vr/failbackreplications/{replicationGroupId}/action/sync	Synchronize latest changes. If the source VM is powered-on performs an online sync. If it's powered-off performs an offline sync.
POST /vr/failbackreplications/{replicationGroupId}/action/powerOffVm	Power Off recovered VM. Failback replication group have to be recovered on the on-premises site in order to use this API .
POST /vr/failbackreplications/{replicationGroupId}/action/powerOnVm	Power On VM. Failback replication group have to be recovered on-prem site in order to use this API .

Table 7-2. List of Operations for vCloud Availability for vCloud Director (Continued)

Operation	Description
`POST /vr/failbackreplications/{replicationGroupId}/action/testCleanup`	Perform test clean up on a given cloud to on-prem replication group
`POST /vr/failbackreplications/{replicationGroupId}/action/reverse`	Reverse a failback replication. The operation uses the existing replication parameters to configure replication to the cloud. The replication destination is the originally protected VM on the tenant VDC.

GET /api/vr/versions

Table 7-3. HTTP Request

Operation	Description
`GET /api/vdc/{vdc-id}/replications`	Returns a list of all the supported version types

Table 7-4. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`8c4732fa15e64d9c89b0240ac3fe3b34`

GET /api/vdc/{vdc-id}/replications

Table 7-5. HTTP Request

Operation	Description
`GET /api/vdc/{vdc-id}/replications`	Returns a list of URIs for incoming replications for the VDC

Table 7-6. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`8c4732fa15e64d9c89b0240ac3fe3b34`

GET /api/vdc/{vdc-id}/peers

Table 7-7. HTTP Request

Operation	Description
`GET /api/vdc/{vdc-id}/peers`	Retrieves all the VR peers that are paired with the specified VDC

Table 7-8. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`8c4732fa15e64d9c89b0240ac3fe3b34`

GET /api/vr/peers/{peer-id}/replications

Table 7‑9. HTTP Request

Operation	Description
`GET /api/vr/peers/{peer-id}/replications`	Retrieves a page of incoming ReplicationGroups for the specified VR Peer

Table 7‑10. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`8c4732fa15e64d9c89b0240ac3fe3b34`

GET /api/org/{org-id}/replications

Table 7‑11. HTTP Request

Operation	Description
`GET /api/org/{org-id}/replications`	Returns a page with list of URIs for incoming replications for that 'Organization ID'

Table 7‑12. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`8c4732fa15e64d9c89b0240ac3fe3b34`

GET /api/vr/replications/{replication-id}/

Table 7‑13. HTTP Request

Operation	Description
`GET /api/vr/replications/{replication-id}`	Returns basic status information for the incoming replication

Table 7‑14. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`8c4732fa15e64d9c89b0240ac3fe3b34`

GET /api/org/{org-id}/enabledForReplicationVdcs

Table 7‑15. HTTP Request

Operation	Description
`GET /api/org/{org-id}/enabledForReplicationVdcs`	Returns list of references to all VDCs for that 'Organization id' that have been enabled for replication

Table 7-16. HTTP Headers

Header	Value
Accept	application/*+xml;version=6.0;vr-version=4.0
x-vcloud-authorization	8c4732fa15e64d9c89b0240ac3fe3b34

DELETE /api/vr/replications/{replication-id}

Table 7-17. HTTP Request

Operation	Description
DELETE /api/vr/replications/{replication-id}	When a DELETE request to this URL is handled, the incoming replication is destroyed. If the primary site is still available, the outgoing replication there must be manually destroyed

Table 7-18. HTTP Headers

Header	Value
Accept	application/*+xml;version=6.0;vr-version=4.0
x-vcloud-authorization	8c4732fa15e64d9c89b0240ac3fe3b34

GET /api/vdc/{vdc-id}/recoveryDetails

Table 7-19. HTTP Request

Operation	Description
GET /api/vdc/{vdc-id}/recoveryDetails	Retrieves information about the recovery capabilities of this VDC

Table 7-20. HTTP Headers

Header	Value
Accept	application/*+xml;version=6.0;vr-version=4.0
x-vcloud-authorization	8c4732fa15e64d9c89b0240ac3fe3b34
Content-Type	application/vnd.vmware.hcs.vrRecoveryDetails+xml

POST /api/vr/replications/{replication-id}/action/failover

Table 7-21. HTTP Request

Operation	Description
POST /api/vr/replications/{replication-id}/action/failover	A recover operation is started for the VMs in the replication. A vCloud task reflecting the operation progress is returned. After successful completion of the task, the VMs are brought to bootable state at the placeholder vApp

Table 7-22. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

Table 7-22. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailoverParams>
```

POST /api/vr/replications/{replication-id}/action/testFailover

Table 7-23. HTTP Request

Operation	Description
`POST /api/vr/replications/{replication-id}/action/testFailover`	A test bubble image creation is started for the VMs in the replication. Ongoing replication is not affected. A vCloud task reflecting the operation progress is returned. After successful completion of the task, the VMs are brought to bootable state at the placeholder vApp.

Table 7-24. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

Table 7-24. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovfenvelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailoverParams>
```

Table 7-24. HTTP Headers

Alternative Body Payload without automatically powering on vApp

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:TestFailoverParams
  xmlns="http://www.vmware.com/vcloud/v1.5"
  xmlns:ns2="http://www.vmware.com/vr/v6.0"
  xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
  xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
  xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
  xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
  xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
    <ns2:Synchronize>false</ns2:Synchronize>
</ns2:TestFailoverParams>
```

POST /api/vr/replications/{replication-id}/action/testCleanup

Table 7-25. HTTP Request

Operation	Description
`POST /api/vr/replications/{replication-id}/action/testCleanup`	Test bubble image clean-up is started. A URI to a VCD task to monitor the progress of the clean-up is returned. After successful completion of the task, VMs are removed from the placeholder vApp and the test bubble image is reverted. Test VMs are automatically powered off. Performs cleanup of the state of a previously executed test failover. Powers off and unregisters the test vApps from VCD and tells HBR server to release the test image files. No-op if there is no prior test failover executed on the Recovery Plan or an individual replication within it.

Table 7-26. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

Table 7-26. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns2:SyncParams
xmlns="http://www.vmware.com/vcloud/v1.5"
        xmlns:ns2="http://www.vmware.com/vr/v6.0"
        xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
        xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData"
        xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
        xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
        xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
        <ns2:RepeatOngoingOnlineSync>false</ns2:RepeatOngoingOnlineSync>
</ns2:SyncParams>
```

POST /api/vr/replications/{replication-id}/action/sync

Table 7-27. HTTP Request

Operation	Description
`POST /api/vr/replications/{replication-id}/action/sync`	Perform sync on an on-prem to cloud replication group

Table 7-28. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

POST /api/vr/replications/{replicationGroupId}/action/powerOffVm

Table 7-29. HTTP Request

Operation	Description
`POST /api/vr/replications/{replicationGroupId}/action/powerOffVm`	Power Offs the protected on-prem VM through REST API

Table 7-30. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

POST /api/vr/{replication-id}/action/reverse

Table 7-31. HTTP Request

Operation	Description
`POST /api/vr/{replication-id}/action/reverse`	Reverses the replication

Table 7-32. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

GET /api/vr/failbackreplications/{replication-id}

Table 7-33. HTTP Request

Operation	Description
`GET /api/vr/failbackreplications/{replication-id}`	Returns the cloud to on-prem replication group with the given {replication-id}

Table 7-34. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

GET /api/org/{org-id}/failbackreplications

Table 7-35. HTTP Request

Operation	Description
`GET /api/org/{org-id}/failbackreplications`	List all the replications available from vCD to on-prem

Table 7-36. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=6.0;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failoverParams+xml`

POST /api/vr/failbackreplications/{failback-id}/action/failover

Table 7-37. HTTP Request

Operation	Description
`POST /api/vr/failbackreplications/{failback-id}/action/failover`	Starts failover from cloud provider group

Table 7-38. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-38. HTTP Headers

Body Payload

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailbackFailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailbackFailoverParams>
```

POST /api/vr/failbackreplications/{replicationGroupId}/action/testFailover

Table 7-39. HTTP Request

Operation	Description
`POST /api/vr/failbackreplications/{replicationGroupId}/action/testFailover`	Test failover replication If the replication is already in the process of getting test-recovered the returned task will fail with OngoingFailover vendor-specific error code. If the replication is already test-recovered the returned task will fail with AlreadyTestRecovered vendor-specific error code.

Table 7-40. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-40. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailbackFailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailbackFailoverParams>
```

POST /api/vr/failbackreplications/{replicationGroupId}/action/sync

Table 7-41. HTTP Request

Operation	Description
`POST /api/vr/failbackreplications/{replicationGroupId}/action/sync`	Synchronize latest changes. If the source VM is powered-on performs an online sync. If it's powered-off performs an offline sync.

Table 7-42. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-42. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailbackFailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailbackFailoverParams>
```

POST /vr/failbackreplications/{replicationGroupId}/action/powerOffVm

Table 7-43. HTTP Request

Operation	Description
`POST /vr/failbackreplications/{replicationGroupId}/action/powerOffVm`	Power Off recovered VM. Failback replication group have to be recovered on the on-premises site in order to use this API.

Table 7-44. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-44. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:PowerOffReplicatedVmParams xmlns="http://www.vmware.com/vcloud/v1.5"
                xmlns:ns2="http://www.vmware.com/vr/v6.0"
                xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
                xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_VirtualSystemSettingData"
                xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
                xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_ResourceAllocationSettingData"
                xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
    <ns2:UseSoftPowerOff>false</ns2:UseSoftPowerOff>
    <ns2:SoftPowerOffTimeoutSeconds>30</ns2:SoftPowerOffTimeoutSeconds>
    <ns2:DisableVmMethods>false</ns2:DisableVmMethods>
</ns2:PowerOffReplicatedVmParams>
```

POST /vr/failbackreplications/{replicationGroupId}/action/powerOnVm

Table 7-45. HTTP Request

Operation	Description
`POST /vr/failbackreplications/{replicationGroupId}/action/powerOnVm`	Power On VM. Failback replication group have to be recovered on-prem site in order to use this API.

Table 7-46. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-46. HTTP Headers

Body Payload

```
<failbackPowerOnRecoveredVmParamsType xmlns="http://www.vmware.com/vr/v6.0"
xmlns:vcloud_v1.5="http://www.vmware.com/vcloud/v1.5"
name="xs:string">
    <vcloud_v1.5:VCloudExtension required="xs:boolean"/>
    <Description> xs:string </Description>
    <WaitForGuestTools> xs:boolean </WaitForGuestTools>
    <ToolsTimeoutSeconds> xs:int </ToolsTimeoutSeconds>
</failbackPowerOnRecoveredVmParamsType>
```

POST /vr/failbackreplications/{replicationGroupId}/action/testCleanup

Table 7-47. HTTP Request

Operation	Description
`POST /vr/failbackreplications/{replicationGroupId}/action/testCleanup`	Perform test clean up on a given cloud to on-prem replication group

Table 7-48. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-48. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailbackFailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailbackFailoverParams>
```

POST /vr/failbackreplications/{replicationGroupId}/action/reverse

Table 7-49. HTTP Request

Operation	Description
`POST /vr/failbackreplications/{replicationGroupId}/action/reverse`	Reverse a failback replication. The operation uses the existing replication parameters to configure replication to the cloud. The replication destination is the originally protected VM on the tenant VDC.

Table 7-50. HTTP Headers

Header	Value
`Accept`	`application/*+xml;version=5.6;vr-version=4.0`
`x-vcloud-authorization`	`3fec5df83ff34ef888287db0e8d7154d`
`Content-Type`	`application/vnd.vmware.hcs.failbackFailoverParams+xml`

Table 7-50. HTTP Headers

Body Payload

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FailbackFailoverParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ns2="http://www.vmware.com/vr/v6.0"
xmlns:ns3="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:ns4="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
xmlns:ns5="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ns6="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
xmlns:ns7="http://schemas.dmtf.org/ovf/environment/1">
<ns2:PowerOn>true</ns2:PowerOn>
</ns2:FailbackFailoverParams>
```

Note The on-prem VM needs to be powered off for this to work

Useful Operations 8

Useful operations for administering a vCloud Availability for vCloud Director installation.

Register vCloud Director with Shared SSO

Register vCloud Director with Shared SSO to which all backing resource vCenter Servers are registered.

Register VCD with lookup service of the tenant VC through REST API(do not user VCD UI)

```
PUT https://<vcd ip>:<port>/api/admin/extension/settings/lookupService
Accept: application/*+xml;version=6.0
Content-Type: application/*+xml;version=6.0
<LookupServiceParams xmlns="http://www.vmware.com/vcloud/extension/v1.5"
xmlns:vcloud_v1.5="http://www.vmware.com/vcloud/v1.5"
userName="SSO_ADMIN_USER" password="SSO_ADMIN_USER_PASS"><LookupServiceUrl>https://{SSO_URL_IP}:
7444/lookupservice/sdk</LookupServiceUrl>
</LookupServiceParams>
```

Note Here onwards vCloud Director can be accessed only with this URL `https://VCD IP or hostname/cloud/login.jsp`.

Enable SSO in vCloud Director UI: **Administration > Federation > Use vSphere Single Sign On**

This chapter includes the following topics:

Changing or Renewing SSL Certificates in vCloud Director

Changing or renewing a certificate in vCloud Director requires removing the existing registered certificate and adding the new one

Within the vCloud Director GUI

Prerequisites

If you must change the SSL certificate in vCloud Director, you must re-register it with the SSO instance.

Procedure

1 Go to **Administration > Federation**

2 Select Unregister

3 To register use the API call shown:

```
PUT https://<vcd ip>:<port>/api/admin/extension/settings/lookupService
Accept: application/*+xml;version=6.0
Content-Type: application/*+xml;version=6.0
<LookupServiceParams xmlns="http://www.vmware.com/vcloud/extension/v1.5"
xmlns:vcloud_v1.5="http://www.vmware.com/vcloud/v1.5"
userName="SSO_ADMIN_USER"
password="SSO_ADMIN_USER_PASS"><LookupServiceUrl>https://{SSO_URL_IP}:
7444/lookupservice/sdk</LookupServiceUrl>
</LookupServiceParams>
```

4 Restart cloud provider vRMS and vRCS instances.

RabbitMQ SSL Install/Configuration

RabbitMQ is used to exchange messages within a vCloud Director environment. Existing deployments will need to be updated to include SSL support.

Download an install RabbitMQ

```
# wget https://www.rabbitmq.com/releases/erlang/erlang-18.3-1.el6.x86_64.rpm
# rpm -i erlang-18.3-1.el6.x86_64.rpm
# wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.1/rabbitmq-server-3.6.1-1.noarch.rpm
# rpm --import https://www.rabbitmq.com/rabbitmq-signing-key-public.asc
# rpm -i rabbitmq-server-3.6.1-1.noarch.rpm
```

Create self signed Certificates

```
# wget https://github.com/michaelklishin/tls-gen/archive/master.zip
# unzip master.zip
# cd tls-get-master/basic
```

Replace `vcd-db.gcp.local` with your domain:

```
# CN=vcd-db.gcp.local PASSWORD=vmware make
# mv testca/ /etc/rabbitmq/
# mv server/ /etc/rabbitmq/
# mv server/ /etc/rabbitmq/
```

Set Owner:

```
# chown -R rabbitmq: /etc/rabbitmq/testca
# chown -R rabbitmq: /etc/rabbitmq/server
# chown -R rabbitmq: /etc/rabbitmq/client
```

Create the file `/etc/rabbitmq/rabbitmq.config`

```
[
  {ssl, [{versions, ['tlsv1.2', 'tlsv1.1', tlsv1]}]},
  {rabbit, [
     {ssl_listeners, [5671]},
     {ssl_options, [{cacertfile,"/etc/rabbitmq/testca/cacert.pem"},
                    {certfile,"/etc/rabbitmq/server/cert.pem"},
                    {keyfile,"/etc/rabbitmq/server/key.pem"},
            {versions, ['tlsv1.2', 'tlsv1.1', tlsv1]},
            {ciphers,  ["ECDHE-ECDSA-AES256-GCM-SHA384","ECDHE-RSA-AES256-GCM-SHA384",
             "ECDHE-ECDSA-AES256-SHA384","ECDHE-RSA-AES256-SHA384",
             "ECDH-ECDSA-AES256-GCM-SHA384","ECDH-RSA-AES256-GCM-SHA384",
             "ECDH-ECDSA-AES256-SHA384","ECDH-RSA-AES256-SHA384",
             "DHE-RSA-AES256-GCM-SHA384","DHE-DSS-AES256-GCM-SHA384",
             "DHE-RSA-AES256-SHA256","DHE-DSS-AES256-SHA256","AES256-GCM-SHA384",
             "AES256-SHA256","ECDHE-ECDSA-AES128-GCM-SHA256",
             "ECDHE-RSA-AES128-GCM-SHA256","ECDHE-ECDSA-AES128-SHA256",
             "ECDHE-RSA-AES128-SHA256","ECDH-ECDSA-AES128-GCM-SHA256",
             "ECDH-RSA-AES128-GCM-SHA256","ECDH-ECDSA-AES128-SHA256",
             "ECDH-RSA-AES128-SHA256","DHE-RSA-AES128-GCM-SHA256",
             "DHE-DSS-AES128-GCM-SHA256","DHE-RSA-AES128-SHA256","DHE-DSS-AES128-SHA256",
             "AES128-GCM-SHA256","AES128-SHA256","ECDHE-ECDSA-AES256-SHA",
             "ECDHE-RSA-AES256-SHA","DHE-RSA-AES256-SHA","DHE-DSS-AES256-SHA",
             "ECDH-ECDSA-AES256-SHA","ECDH-RSA-AES256-SHA","AES256-SHA",
             "ECDHE-ECDSA-DES-CBC3-SHA","ECDHE-RSA-DES-CBC3-SHA","EDH-RSA-DES-CBC3-SHA",
              "EDH-DSS-DES-CBC3-SHA","ECDH-ECDSA-DES-CBC3-SHA","ECDH-RSA-DES-CBC3-SHA",
             "DES-CBC3-SHA","ECDHE-ECDSA-AES128-SHA","ECDHE-RSA-AES128-SHA",
              "DHE-RSA-AES128-SHA","DHE-DSS-AES128-SHA","ECDH-ECDSA-AES128-SHA",
              "ECDH-RSA-AES128-SHA","AES128-SHA","EDH-RSA-DES-CBC-SHA","DES-CBC-SHA"]},
             {verify,verify_none},
                     {fail_if_no_peer_cert,false}]}]}
].
```

Start RabbitMQ

```
# service rabbitmq-server start
```

Enable RabbitMQ UI

To enable the UI on `http://server-name:15672/`

```
# rabbitmq-plugins enable rabbitmq_management
```

Create admin user to log in:

```
# rabbitmqctl add_user admin vmware
# rabbitmqctl set_permissions -p / admin ".*" ".*" ".*"
# rabbitmqctl set_user_tags admin administrator
```

Securing vRS Traffic

vRS prodives replication of data that should be secured using SSL and a certificate through `stunnel`

Securing VR Traffic with stunnel

Download the stunnel RPM:

```
# rpm -ivh
http://pkgs.clodo.ru/suse/test/213.141.145.240/SLES11SP2_UPD64/stunnel-4.36-0.10.1.x86_64.rpm
```

Generate `stunnel` certificate using the command shown. Use a CA signed certificate or self signed wild card certificate:

```
# cd /etc/stunnel
# openssl req -new -x509 -keyout stunnel.pem -out stunnel.pem -days 3650 -nodes -subj
"/C=US/ST=California/L=SanFrancisco/O=Palo Alto/CN=*.se.vpc.vmw"
```

Note The `stunnel` certificate can be used for all vSphere Replication servers as it is a wild card certificate and simplifies the importing of `stunnel` certificates into the Cloud Proxy truststore as mentioned in next section.

Create directories and change ownership and permissions

```
# mkdir /var/run/stunnel/
# mkdir /var/log/stunnel
# chown -R stunnel:nogroup /var/run/stunnel/ /var/log/stunnel
# chown stunnel:nogroup /etc/stunnel/stunnel.pem
# chmod 600 /etc/stunnel/stunnel.pem
```

Modify the `stunnel.conf` file to reflect the following configuration entries only

```
client = no
foreground=no   this needs to be added
pid = /var/run/stunnel/stunnel.pid
debug = 1
output = /var/log/stunnel/stunnel.log
cert = /etc/stunnel/stunnel.pem

[$VRS_HOSTNAME]
accept = 9998
connect = 31031
```

Start and Enable 'stunnel' service

```
service stunnel start
chkconfig stunnel on
```

Firewall Configuration

After starting `stunnel` on vRS appliance, you must drop packages from outside of the network to ports 31031, 44046, and 9998 must be allowed in firewall configuration.

Steps for SuSE firewall configuration:

```
# vi  /etc/sysconfig/SuSEfirewall2
```

Change from

```
FW_SERVICES_EXT_TCP="22 80 5480 8043 8123 10000:10020 31031 40404 41111 44046"
```

To

```
FW_SERVICES_EXT_TCP="22 80 5480 8043 8123 9998 10000:10020 40404 41111"
```

Restart SuSE firewall

```
# /etc/init.d/SuSEfirewall2_setup reload
```

Enable `stunnel` service in `TCP_WRAPPERS` in /etc/hosts.allow

```
# vi /etc/hosts.allow
```

Add following line

```
$VRS_HOSTNAME : ALL : ALLOW
```

Import Stunnel Certificates to Cloud Proxy TrustStore

Note This is required to use Self-signed certificates in `stunnel`

Copy `stunnel` certificate from one vRS server to one of the cloud Proxy cell to use wild card certification for `stunnel` for all vRS servers

```
# scp ${VRS_HOSTNAME}:/etc/stunnel/stunnel.pem ${CLOUDPROXY_HOSTNAME}:/tmp/
```

Convert `.pem` file to `.der`

```
# openssl x509 -outform der -in stunnel.pem -out stunnel.der
```

Import the certificate into `/opt/vmware/vcloud-director/jre/lib/security/cacerts` of Cloud proxy

```
# /opt/vmware/vcloud-director/jre/bin/keytool -import -alias stunnel_{VRS_HOSTNAME} -
keystore /opt/vmware/vcloud-director/jre/lib/security/cacerts -file stunnel.der
```

Restart cloud proxy service

```
# service vmware-vcd restart
```

Copy `/opt/vmware/vcloud-director/jre/lib/security/cacerts` from first cloud proxy cell to remaining cells and restart "vmware-vcd" service.

Index