Deploying SwiftStack Object Storage for VMware Integrated OpenStack

April 2015
Table of Contents

Introduction

Important Notes

Step-by-Step Details

Step 1: Register for the SwiftStack Management Service
Step 2: Create Your SwiftStack Nodes
Step 3: Configure Your SwiftStack Cluster
Step 4: Put Your First Data into SwiftStack!
Step 5: Connect SwiftStack to VMware Integrated OpenStack (VIO)
Step 6: Configure OpenStack Glance to Use SwiftStack Object Storage

Appendix
Introduction

For VMware customers, VMware Integrated OpenStack (VIO) provides a simple and fast way to allow software development teams to innovate using the industry-standard OpenStack API on top of their existing vSphere infrastructure. One critical component of OpenStack is Swift object storage, and SwiftStack provides simple-to-deploy, simple-to-manage, and simple-to-monitor management software for OpenStack Swift (you might say, SwiftStack makes Swift “enterprise-ready”). Together, VMware and SwiftStack provide an easy way to enable OpenStack Swift object storage VIO deployments on top of vSphere infrastructure.

As with most OpenStack deployments, SwiftStack object storage is typically used in three ways in VMware Integrated OpenStack deployments:

1. **Glance virtual machine images**: The OpenStack Glance service can store its virtual machine image files in Swift object storage.

2. **Cinder volume backups**: The OpenStack Cinder service can be configured to backup its block storage volumes to Swift object storage.

3. **Native application storage**: With the exponentially increasing growth of unstructured data, modern application developers are rapidly developing against the Swift API to give their applications access to massively scalable and geographically distributed Swift object storage systems.

This document describes the steps necessary to deploy SwiftStack and integrate it with the Keystone authentication service in VMware Integrated OpenStack. It is assumed that VIO is already deployed. (For help with VIO, please see VMware’s documentation.) For additional details about integrating SwiftStack with OpenStack Glance and Cinder backup services, refer to the OpenStack integration information on http://swiftstack.com/partners.

Important Notes

**A SwiftStack deployment as described in this document is specifically intended for development and testing environments—not for production scenarios.** Production SwiftStack deployments are capable of scaling to span multiple continents and include thousands of nodes and exabytes of usable storage capacity. At this scale—and even at smaller scale—**production SwiftStack systems are deployed on bare metal hardware with direct access to individual spinning drives and/or solid state drives**, and Swift’s availability and data protection capabilities have been proven at scale many times over—in both private and public clouds alike. **SwiftStack will support VIO users deploying three-node Swift clusters in ESX according to this document and containing up to 10TB of unique data in test and development workloads.** If you are interested in SwiftStack...
for a production environment or an environment containing more than 10TB of unique data, please email contact@swiftstack.com.

Step-by-Step Details

Step 1: Register for the SwiftStack Management Service
The following steps will guide you to register for an account on the SwiftStack Management Service, add an administrative user to manage the account, and create your SwiftStack cluster. Subsequent steps will guide you through adding nodes to that cluster and using it.

1. Register at swiftstack.com for an account on the SwiftStack Management Service.

   Make sure you read and accept the terms of service and continue to the “Invite Administrator” menu:
Click on “Agree and Start Trial” to continue to the menu where you can choose the administrator credentials for the person you want to invite (for test purposes this could be the same person that has created the initial SwiftStack account). An email will be sent to the mail address you filled in. Open the email, and follow the link provided. You will be directed to the login page for the SwiftStack Controller:

After successful registration, login to the SwiftStack controller using the administrative credentials you created earlier, and create a cluster in your SwiftStack Management Service account.

2. Create your first cluster
a. Login to your SwiftStack controller account.

b. At the following menu you can set the cluster, credentials. Most important is the “Network Configuration” part. You will need to fill in the IP address, and—if your setup requires it—the fully qualified domain name for the SwiftStack API interface (see next chapter for DNS information used in this tutorial). When the cluster is operational and all nodes are deployed, you can change the IP address and/or the FQDN here. The default setup will use a SwiftStack “Virtual Load Balancer” to assign the IP address to the cluster. Be aware that you should not have another load balancer or similar system in the same subnet as your cluster (and the nodes).
After you have completed the form, click “Submit Changes”.
c. Now you are ready to start deploying your nodes:
Step 2: Create Your SwiftStack Nodes

The following steps will guide you through downloading a pre-built VMware appliance OVA, which is essentially an Ubuntu Linux virtual machine, importing it into ESX, and installing the SwiftStack software. (In a production environment, instead of importing an OVA into your ESX infrastructure, you would install real Linux servers and then follow the same subsequent steps to install SwiftStack and include those nodes in your SwiftStack cluster.)

1. Download the VMWare Swiftstack OVA
   a. Preparation: Make sure you have IP addresses reserved and have operational forward and reverse DNS lookup zone records in the DNS server you are using. In the examples below, the following forward lookup records have been setup and used for zone example.com:
      swiftstackapi A 10.0.0.60
      node01 A 10.0.0.61
      node02 A 10.0.0.62
      node03 A 10.0.0.63

      The following reverse lookup records for zone 0.0.10.in-addr.arpa:
      60 PTR swiftstackapi.example.com.
      61 PTR node01.example.com.
      62 PTR node02.example.com.
      63 PTR node03.example.com.

   b. Download the SwiftStack VMWare image (in OVA format) from https://files.swiftstack.com/SwiftStack_node_image.ova. (To confirm the accuracy of your download, compare your MD5 checksum to that in https://files.swiftstack.com/SwiftStack_node_image.md5.) To create a small SwiftStack cluster, the image needs to be deployed three times. We recommend to download the image once, store it on local storage, and deploy from there to your VMWare data center cluster.
c. Login to the VMWare vSphere™ Web Client using an account with enough rights to deploy and start virtual machines:
d. **Navigate to the “vCenter -> vSphere -> VMs and Templates” Inventory view:**
e. Right click on the vSphere or data center name and choose “Deploy OVF Template”:

f. Choose “Local file” and navigate to the location where you have downloaded the SwiftStack node image using the “Browse” button:
g. Type a unique name to identify the new virtual machine:

h. Choose a datacenter cluster or a specific ESXi host to deploy the image:

i. Choose a data store; consider that the performance of SwiftStack depends on network connectivity, compute resources in its nodes, and the disks where data is stored, so choosing a
higher-performing data store here will result in higher-performing SwiftStack as well:

j. Choose the network that will be attached to the new image. Eventually, the SwiftStack cluster nodes need to be configured inside the same network as the VMWare Integrated OpenStack management servers. This can be changed later, but it is important at this stage that you choose a source network that can reach the Internet and your local DNS servers.

k. Review the settings for the new virtual machine, and make any corrections if necessary:
l. Keep an eye on the recent tasks status, and make sure deployment of template succeeds:

![Recent Tasks Screenshot]

m. Start the OVA image, and open a vSphere console. The very first time the image boots it will ask for network configuration information. Please answer the questions carefully with the values available for your organization. In the following screenshot, the IP addresses from the example above are used:

![Network Configuration Screenshot]

n. Answer `y` (and press enter) at the “Reboot now?” prompt to allow the system to reboot.

3. **Install the SwiftStack software in your first virtual node in ESX.**
   a. You may want to connect to the new node using a tool such as Putty or any other SSH client. This will make it easier to copy the claim URL and paste it in a browser window in the following steps.
   
   Connect and login as user `swiftstack` with password: `password`
b. Run the following command, and review its output:
   `curl https://platform.swiftstack.com/install`

c. Be aware that if you have previously installed OpenStack Swift on the node, it will be replaced with the version included in SwiftStack. When your review is complete and you are ready to install, run the command:
   `!! | bash`

d. Wait for the software installation to finish; after a few minutes, a claim URL will appear. The identifier at the end of the URL is unique and will be different for each new node added (for example):
   +-----------------------------------------------------------
   | Your claim URL is:                                       |
   | https://platform.swiftstack.com/claim/5b5d67e6-9b34-11e4-b2ea-000c2948e5e9 |
   +-----------------------------------------------------------

   You will need this URL in the next steps.
Step 3: Configure Your SwiftStack Cluster

The following steps will guide you through configuration of your SwiftStack nodes in the SwiftStack Management Service and the final steps necessary to make your SwiftStack cluster ready for use!

1. **Complete the simple cluster and node configuration on the SwiftStack Management Service.**
   a. Make sure you have an active login session to your cluster. Open a second browser window or tab and copy the claim URL into a browser, and press enter:

   ![Claiming Node](image1.png)

   b. Click “Claim Node”. The SwiftStack Controller will make a few attempts to contact the node until it succeeds:

   ![Claiming Node](image2.png)
c. Once again, click on “Claim Node” to make the node available to your organization:

![SwiftStack interface showing a newly claimed node]

You have 1 unprovisioned node. You can ingest it now by selecting “Add Nodes” twice.

TestDevelopment
0 nodes, 0 gateways

Manage | Monitor | Users & Accounts | Add Nodes | Remove Cluster

Create New Cluster:
Name*:
Deployment Status*:
Testing

Click Cluster

---

d. Now add the new node to your cluster by clicking on “Add Nodes”. You are now ready to ingest the node:

![SwiftStack interface showing unprovisioned nodes]

No nodes have been added to this cluster yet.

Unprovisioned Nodes

<table>
<thead>
<tr>
<th>Hostname</th>
<th>MAC Address</th>
<th>Select a Zone</th>
<th>Select a Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>sanode01.internal.swiftstack.com</td>
<td>00:0c:29:48:e5:a9</td>
<td>Region 1, Zone 1 (r1z1)</td>
<td>Swift Node</td>
</tr>
</tbody>
</table>

Ingest Now
e. Click “Ingest now”. The node will be added to your cluster; make sure the operation reports back with a green success message:

f. At this stage, you can re-assign the network interfaces. As the standard image only uses one network interface, there is no need to adjust the defaults. The new node is still disabled and will not function until drives have been assigned. Click “Reassign Interfaces” to be directed to the “Unmanaged Drives” menu.

g. The downloaded image has been preconfigured with one system drive and five 10 gigabyte thin-provisioned, unformatted drives. Check they are detected by the SwiftStack controller:
h. Select all drives at once by clicking on the marked checkbox next to “All Unmanaged Drives;” only drives marked as available will be used and formatted:

Unmanaged Drives

Tick to mark all or none of the available drives

<table>
<thead>
<tr>
<th>Drive Path</th>
<th>Size</th>
<th>Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>dm-0</td>
<td>6.1 GiB</td>
<td>Ignored</td>
</tr>
<tr>
<td>sdb</td>
<td>10.7 GiB</td>
<td>Available</td>
</tr>
<tr>
<td>sdc</td>
<td>10.7 GiB</td>
<td>Available</td>
</tr>
<tr>
<td>sdd</td>
<td>10.7 GiB</td>
<td>Available</td>
</tr>
<tr>
<td>sde</td>
<td>10.7 GiB</td>
<td>Available</td>
</tr>
<tr>
<td>sdf</td>
<td>10.7 GiB</td>
<td>Available</td>
</tr>
</tbody>
</table>

i. Click “Format;” after a few minutes, the SwiftStack Controller will display the “Manage Swift Drives” page:

Manage Swift Drives

This node currently has the role Swift Node. Node roles dictate which Swift services a node should run and in which policies its drives may participate.

Select Drives

<table>
<thead>
<tr>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Swift Drives</td>
</tr>
<tr>
<td>Unmounted</td>
</tr>
<tr>
<td>Missing</td>
</tr>
<tr>
<td>Failed</td>
</tr>
</tbody>
</table>

Select Drives by Policy Group

<table>
<thead>
<tr>
<th>Policy Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not in any Policies</td>
</tr>
</tbody>
</table>

j. Select the tickbox next to “All Swift Drives,” and assign a policy by clicking on “Add or Remove Policies.” Select both “Account & Container” and “Standard-
Replica.” Click “Add Policies,” and on the drop-down list, select “Immediately.”

k. The Controller will show an updated “Manage Swift Drives” page. Check the policy status of all five drives has been updated to reflect your choice.

l. Once the drives are configured, the last step is to enable the node:

Click “Enable” to move to the final stage in the deploy process:

Click “Enable Node” so you can deploy the nodes configuration; deploy the node
configuration, and click on “Click Here to Deploy”:

m. To complete the setup of your three-node cluster, you need to add another two nodes to it. Deploy two more SwiftStack OVA images into your ESX environment using vSphere; boot the new images; configure the network settings; reboot; install the SwiftStack software; claim; and deploy the node. Please refer to the procedure described above.

2. **Create a SwiftStack user.**
   a. After all three nodes are enabled within the cluster, one or more authorized users must be created for your cluster to allow access.

Because this is a new cluster, you must also create at least one Swift User for it. This user provides authentication of last resort in the event that external auth systems become unavailable.
b. In this tutorial, we will create one new user. Click “Create a User”:

c. Continue, click “+ Create New User”:

d. Fill in the form with the user credentials you desire, and click “Add SwiftStack Auth User”: 
e. Before the user is configured in the SwiftStack cluster, the changes must be deployed to the nodes; click “Deploy Changes”:

![Deploy TestDevelopment](image)

f. Click “Deploy Config to Swift Nodes” to complete the process.

g. The controller will show a status page after deployment has completed:

![Status Page](image)

Notice the addresses of the SwiftStack authorization and web console URLs.

**Step 4: Put Your First Data into SwiftStack!**

Now that your cluster is ready to use, the next few steps give you a simple example of how you can put your first data into it. This is useful to ensure everything is functioning properly before you connect it to your VMware Integrated OpenStack deployment, which will be the final part of this setup.

1. **Put your first data into SwiftStack!**
   
a. On the cluster “Monitor” tab (see last step in previous section), click the link to the web portal; you will be directed to the login Web console page:
b. Login as the user you created before.

c. For this (new) account, there are no containers and content yet. To create a container, click on the “+” sign under the “Container” header:

This will show the “Create a container” dialogue:

```
Create a container

Development

Standard-Replica

Cancel   OK
```

d. Fill in the name of the container you want to create, and click “OK.”
e. At the next page, you can drag-and-drop files into the container you just created. Please drag-and-drop some local files into the container. If you drag a large file, take notice of the progress bar:

![Progress bar](image)

f. When the upload has completed successfully, the progress bar will turn to green:

![Progress bar](image)
Step 5: Connect SwiftStack to VMware Integrated OpenStack (VIO)

You now have a three-node SwiftStack cluster running on your ESXi infrastructure and being managed by the SwiftStack Management Service. Now that everything is up and running, the final step is to connect SwiftStack to the Keystone authentication service in OpenStack and tell OpenStack to use SwiftStack as its storage for Glance (virtual machine) images. The following steps guide you through these last few pieces of configuration.

You must have a working VMWare Integrated OpenStack (VIO) system installed. There are a few very important preparations necessary before a SwiftStack cluster can participate successfully in a VIO deployment. Most important, the SwiftStack nodes must be inside the same subnet as the Management network used for the VIO cluster.

Check under vSphere that this is the case; during the deployment phase, select the VIO Management network:

SwiftStack nodes must be connected to VIO Management Network

This allows all VIO systems and SwiftStack nodes to communicate with each other:
In the VMWare vSphere Web client, determine the name and IP addresses of the VIO controller systems that are running to support the VMWare OpenStack implementation:
In the example shown above, the first VIO Controller is configured with IP address 192.168.152.139 in the management network 192.168.152/24. The SwiftStack nodes will need to be configured as part of this network. You will have to make sure they match the VIO implementation you are using.

You will use this system to issue the corresponding Keystone (authentication) commands to prepare the VIO system to accept the SwiftStack cluster. You will need to know the VIO administrative user password (normally this is a user called admin).

You will need to add to the existing VIO configuration the following new objects and complete the associated steps:

- A new service definition for swift
- A new user called swift as part of the services tenant (project)
- Upgrade the new user to admin role
- Swift endpoints—these are the external and internal URLs used to access the SwiftStack services through OpenStack
Reconfigure Glance on both VIO controllers

From a Windows™ computer, use an SSH client such as Putty to login to the VIO Controller (on an Apple or Linux system, you can use the SSH client included with the system). Login to the VIO controller with username root and password vmware.

In the home directory of root, you will find a handy authentication file you can use to set the environment variables containing the username and Keystone authentication URLs for the running VIO system:

```
source cloudadmin.rc
```

Optionally, if you do not want to keep retyping the administrative password, set the following environment variable to the value of the password of the administrative user (not recommended in a production environment!):

```
export OS_PASSWORD=<password>
```

Create the new Swift service (if you haven’t set the OS_PASSWORD variable, you will be asked for a password):

```
keystone service-create --name=swift --type="object-store" --description="SwiftStack service"
```

Use the following command to confirm success:

```
keystone service-list
```

The system will respond with a list of available services; your new Swift services should be visible here:

```
+---------------------------------+-----+---------+-----------------
| id | name | type         | description      |
+---------------------------------+-----+---------+-----------------
| ca74a5be97b5427fba84364efb831f70 | cinder | volume    | OpenStack Volume Service |
| 5270db6a3253475b4dfb1e6ec557ae | cinderv2 | volumev2  | OpenStack Volume Service v2 |
| fe360d12e5da4ce5bd4e5672b19fa34d | glance | image     | OpenStack Image Service |
| 18892a4e3c974d02a6a279254c597581 | keystone | identity  | OpenStack Identity Service |
| 4a70b7f5d189465da2b25f1b8642f240 | neutron | network   | OpenStack Network Service |
| 6601b6f8ad04e048f6679e2dcaba010 | nova | compute   | OpenStack Compute Service |
| 6b97c46d85f74671b81647f5f83a7bbb | swift | object-store | SwiftStack service |
+---------------------------------+-----+---------+-----------------
```

Check if the generic OpenStack tenant service is created (this should already be done during the VIO installation; if it is not available, it needs to be created):

```
keystone tenant-list
```

```
+---------------------------------+-----+-----+
| id | name | enabled |
+---------------------------------+-----+-----+
| 4c2136a2310b4dce8542001824c4fb19 | admin | True |
| 507efa6091584228b6ffe7ab1b5b0de3 | service | True |
+---------------------------------+-----+-----+
```
Add the new Swift user:

```
keystone user-create --name=swift --pass=swiftpass --tenant-id=service
```

You can use the keystone user-list command to confirm the new user has been created (the user should also be visible in the VIO administrative user interface under Admin → Identity Panel → Users)

Add admin role to the new user:

```
keystone user-role-add --user-id swift --role-id admin --tenant-id service
```

To verify, use: `keystone user-role-list --user swift --tenant service`

Add end-points for the Swift service (this command should be typed/copied on one line). Make sure to replace the IP addresses with ones relevant for your VIO and SwiftStack configuration. All the IP addresses should be in the Management Network accessible by both SwiftStack nodes and VIO management:

```
keystone endpoint-create --region nova --service-id swift --publicurl "https://192.168.152.64/v1/KEY_\$(tenant_id)s" --adminurl "http://192.168.152.64/v1" --internal "http://192.168.152.64/v1/KEY_\$(tenant_id)s"
```

You can confirm the existence of the new end-point using the `keystone endpoint-list` command.
You can also see the new configuration in the VIO interface; login as your admin user, and navigate to Admin → System Panel → System Info:

![System Info](image)

The configuration of the VIO OpenStack system is now complete. To allow SwiftStack to participate, two Swift middleware components must be installed: Keystone Auth and Keystone Auth Token Support.
Login to your SwiftStack controller on https://platform.swiftstack.com/:

If you have multiple clusters, navigate to the one you want to integrate with VIO:
Click on “Manage”:

Deploy TestDevelopment

No pending configuration changes
Click on the “Middleware” icon to navigate to the available middleware options:

In the list, you will see both middleware components, “Keystone Auth” and “Keystone Auth Token Support,” marked as “inactive”.

<table>
<thead>
<tr>
<th>Middleware Component</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Large Object</td>
<td>active</td>
</tr>
<tr>
<td>FormPost</td>
<td>active</td>
</tr>
<tr>
<td>Rate Limit</td>
<td>active</td>
</tr>
<tr>
<td>Static Large Object</td>
<td>active</td>
</tr>
<tr>
<td>Swift Web Console</td>
<td>active</td>
</tr>
<tr>
<td>SwiftStack Authentication</td>
<td>active</td>
</tr>
<tr>
<td>SwiftStack Authorization</td>
<td>active</td>
</tr>
<tr>
<td>TempURL</td>
<td>active</td>
</tr>
<tr>
<td>Account Quotes</td>
<td>inactive</td>
</tr>
<tr>
<td>Bulk Operations</td>
<td>inactive</td>
</tr>
<tr>
<td>DNNAME Lookup</td>
<td>inactive</td>
</tr>
<tr>
<td>Container Quotas</td>
<td>inactive</td>
</tr>
<tr>
<td>Cross Domain</td>
<td>inactive</td>
</tr>
<tr>
<td>Domain Remap</td>
<td>inactive</td>
</tr>
<tr>
<td>Keystone Auth</td>
<td>inactive</td>
</tr>
<tr>
<td>Keystone Auth Token Support</td>
<td>inactive</td>
</tr>
<tr>
<td>LDAP-based Authentication for SwiftStackAuth</td>
<td>inactive</td>
</tr>
<tr>
<td>List Endpoints</td>
<td>inactive</td>
</tr>
<tr>
<td>Name Check</td>
<td>inactive</td>
</tr>
</tbody>
</table>
Click on “Keystone Auth” to navigate to its configuration menu:

Click the “Enabled” tick box, and add the role `_member_` to the default admin and swiftoperator roles (underscores must be included):

```
operate_roles               admin, swiftoperator, _member_
reseller_prefix            KEY_
reseller_admin_role        ResellerAdmin
```

Click “Submit.”

The system will return you to the Available Swift Middleware list; click on “Keystone Auth Token Support”: 
Keystone Auth Token Support

Configuring Keystone Auth Token Support

This middleware is required for Keystone Authentication/Authorization (along with the "Keystone Auth" middleware).

Settings

- **auth_admin_prefix**: [Field]
- **auth_host**: 192.168.152.128
- **auth_port**: 35357
- **auth_protocol**: http
- **auth_uri**: http://192.168.152.128:5000
- **admin_user**: swift
- **admin_password**: swiftpass
- **admin_tenant_name**: service
- **signing_dir**: /var/cache/swift
- **include_service_catalog**: False

[Submit] [Cancel]
Enable this component by clicking the “Enabled” tick box. Fill the configuration with the following settings:

- auth_admin_prefix: leave empty
- auth_host: IP address of Keystone authentication server, in the above example 192.168.152.128 (this information is available in the VIO admin console or from the admin URL obtained using the `keystone endpoint-list` command)
- auth_port: 35357 (admin URL endpoint)
- auth_protocol: http
- auth_uri: IP address and port of the Keystone authentication server, use here the internal URL obtained from the `keystone endpoint-list` command
- admin_user: User configured in VIO on behalf of Swift
- admin_password: Password configured for the admin_user
- admin_tenant_name: Tenant (project) for the admin_user
- signing_dir: /var/cache/swift
- include_service_catalog: False
- http_connect_timeout: 4
- http_request_max_tries: 1

Click “Submit”: 
Click “Click Here to Deploy”:

Click “Deploy Config to Swift Nodes;” the system will show the status screen:

Wait for the “Pending Jobs” → “Config Deploy in progress” message to change to the “Job Finished” status message (in green).
You have configured your SwiftStack cluster as part of your VIO environment. To test functionality from OpenStack VIO Keystone to SwiftStack, log into the VIO user interface as admin, and create a new user for testing purposes.

Login to the VIO console as the administrative user:

![Login page](image)

Navigate to Admin → Identity Panel → Projects:
Click on “+ Create Project,” and create a project called “John’s project”:

Navigate to Admin → Identity Panel → Users:

Click on “Create user,” and create a user with the following credentials:

Username: jdoe
Email: jdoe@example.com
Password: jdpassword
Primary Project: John’s project
Log back into one of the VIO controllers (account root, password vmware), and using the Swift command, check you can access John’s account:

```
swift -A http://192.168.152.128:5000/v2.0 -V 2 -U "John's project":jdoe -K jdpassword stat
```

The system will respond with a status message, indicating there are no containers or objects available yet:

```
Account: KEY_913b4464df694397bba0389c543fca3e
Containers: 0
Objects: 0
Bytes: 0
Content-Type: text/plain; charset=utf-8
X-Timestamp: 1427553977.87200
X-Trans-Id: txb65d62854e3c44acb28b2-005516beb9
X-Put-Timestamp: 1427553977.87200
```

If you see a message such as “Unauthorised. Check username, password and tenant name/id,” check that you are using the correct credentials (project and user name). If the user credentials are correct, carefully double-check that you have set all the configuration parameters and executed all the commands described in the previous steps correctly.

Upload a file, and create a new container (type command on single line):

```
swift -A http://192.168.152.128:5000/v2.0 -V 2 -U "John's project":jdoe -K jdpassword upload configuration /etc/resolv.conf
```
This response indicates the file is uploaded successfully:

```
/etc/resolv.conf
```

Check that the container exists using the list command:

```
swift -A http://192.168.152.128:5000/v2.0 -V 2 -U "John's project":jdoe -K jdpassword list
```

This will display a list of all container and file names:

```
configuration
```

Check if the uploaded file is available in the new container:

```
swift -A http://192.168.152.128:5000/v2.0 -V -U "John's project":jdoe -K jdpassword list configuration
```

The file name (including its original path) is shown:

```
/etc/resolv.conf
```

Notice the Keystone administrative authentication URL being used to access the SwiftStack object store.
Step 6: Configure OpenStack Glance to Use SwiftStack Object Storage

As we will be accessing the Swift object store, the configuration of the image management subsystem, Glance, needs to be adjusted. This can be done by editing both /etc/glance/glance-api.conf files on both VIO controllers. Remember, in the examples used above the authentication URL was:

http://192.168.152.128:5000/v2.0

Glance needs to be told about this and switch to Swift as the preferred object store. A few small changes must be made to Glance’s configuration file (on both VIO controllers) to enable Swift.

You will need to login to both VIO controllers to make the change to the configuration files. As a reminder, the IP addresses can be viewed from VMWare’s vSphere web client.

For VIO-Controller-0:
For VIO-Controller-1:

Login to the VIO-controller-0 system as user root (password `vmware`). Set your environment variables, and check that you can see the currently available images (the password you need to set is the one belonging to the VIO OpenStack administrative user):

```bash
source cloudadmin.rc
export OS_PASSWORD=<password>
glance image-list
```

Glance will show you the available images (slightly edited for clarity):

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Disk Format</th>
<th>Container Format</th>
<th>Size</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>b70a3392-0417-4889-b06c-bfca817ccdb9</td>
<td>cirros-so-0.3.2-i386</td>
<td>vmdk</td>
<td>bare</td>
<td>1234586</td>
<td>active</td>
</tr>
</tbody>
</table>
```

You can also see this in the VIO OpenStack web interface.

Download a copy of this image to make it available for testing later (copy the ID qualifier from the output of your `glance image-list` command):

```bash
glance image-download --file /root/test_cirros_image.vmdk b70a3392-0417-4889-b06c-bfca817ccdb9
```

Now you are ready to implement the new configuration. First, disable the Glance API on both controllers using the following command:

```bash
service glance-api stop
```

Use your favourite editor to make the following adjustments to the Glance configuration file on VIO-Controller-0 (you will do the other controller later towards the end of this chapter): `/etc/glance/glance-api.conf`
First, make a backup copy of the configuration file:

```bash
cd /etc/gla
cp glance-api.conf glance-api.backup.conf
```

Make the following changes to the `glance-api.conf` file, if you like using `vi` type (the simpler `nano` editor is also available):

```bash
vi glance-api.conf
```

Change:

around line 12: from `default_store = vsphere` to `default_store = swift`

around line 312: from `swift_store_auth_address = 127.0.0.1:5000/v2.0/` to `swift_store_auth_address = http://127.0.0.1:5000/v2.0/

around line 318: from `swift_store_user = jdoe:jdoe` to `swift_store_user = service:swift`

You might see another value than `jdoe:jdoe`, just replace it anyway.

around line 323: from `swift_store_key = a86850deb2742ec3cb41518e26aa2d89` to `swift_store_key = swiftpass`

You might see another key value, just replace it anyway. This is the password of the SwiftStack user swift created in a earlier step.

around line 331: from `swift_store_create_container_on_put = False` to `swift_store_create_container_on_put = True`

Save the changes to the file, and restart the glance api and registry services (only on VIO Controller-0 at this stage):

```bash
service glance-api start
service glance-registry reload
```

Test the new configuration by uploading the image we copied earlier (type this command on one line):

```bash
glance image-create --name "cirros-test" --disk-format vmdk --container bare --is-public True --progress < /root/test_cirros_image.vmdk
```

If all is well, the system will respond with a progress bar and a status display of the new image:

```
[==========================================] 100%
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>dde2fd6a1d57df5b2830f5fe502b9b17</td>
</tr>
<tr>
<td>container_format</td>
<td>bare</td>
</tr>
<tr>
<td>created_at</td>
<td>2015-03-29T13:31:30</td>
</tr>
<tr>
<td>deleted</td>
<td>False</td>
</tr>
<tr>
<td>deleted_at</td>
<td>None</td>
</tr>
<tr>
<td>disk_format</td>
<td>vmdk</td>
</tr>
<tr>
<td>id</td>
<td>84c8e256-a389-415b-9745-0b6adcb22a17</td>
</tr>
<tr>
<td>ia_public</td>
<td>True</td>
</tr>
<tr>
<td>min_disk</td>
<td>0</td>
</tr>
<tr>
<td>min_ram</td>
<td>0</td>
</tr>
<tr>
<td>name</td>
<td>cirros-test</td>
</tr>
<tr>
<td>owner</td>
<td>4c2136a2310b4dce8542829124c4fb19</td>
</tr>
<tr>
<td>protected</td>
<td>False</td>
</tr>
</tbody>
</table>
Now, check you are indeed using the Swift store for the newly uploaded image:

```
swift --os-auth-url http://127.0.0.1:5000/v2.0 --os-tenant-name service --os-username swift --os-password swiftpass list glance
```

The output will be the filename stored by Glance into the container `glance`:

```
84c8e256-a389-415b-9745-0b6adcb22a17
```

This ID number should match the number given by the `glance create-image` command.

When you have completed testing, make the same configuration change to VIO-Controller-2, and test again that images can still be downloaded and uploaded.

Congratulations!! You now have enabled your VMWare Integrated OpenStack cluster to use SwiftStack as the object storage for machine images.

**Appendix**

**To rerun the network configuration script in a SwiftStack node:**
The first-boot script `/usr/local/bin/firstBoot.sh` runs after every reboot, but it only asks to reconfigure the network on the first boot. You can enable the network configuration script to run again by removing `/var/log/SwiftStack_firstBoot.log` and executing `/usr/local/bin/firstBoot.sh`. 