

Storage Configuration Guide

Document Overview

Goal

The purpose of this document is to outline the configuration of SolidFire storage in a CloudStack Infrastructure-as-a-Service (IaaS) environment, outline how to leverage SolidFire Quality of Service (QoS) capabilities and demonstrate how to automate the configuration.

Audience

This document is intended to assist a solution architect, sales engineer, consultant or IT administrator with basic configuration and proof of concept efforts. The document assumes the reader has an architectural understanding of CloudStack and has reviewed related administration guides.

Prerequisite(s)

Configuring storage in a CloudStack environment requires that the CloudStack platform has already been installed, configured and networked to the SolidFire cluster. Additional documentation on CloudStack installation can be found at:

<http://cloudstack.apache.org/docs/en-US/index.html>

Overview

Cloud infrastructures today demand levels of storage performance, scale and multi-tenancy that are rarely encountered in traditional enterprise IT settings. The mixed workload nature of these environments further exacerbates the problem by diminishing the benefit of any application specific optimizations. Unfortunately traditional storage systems were not designed to address these unique challenges. Lacking the raw performance at scale, these legacy systems fail to keep pace with the dynamic growth and efficiency mandates of cloud infrastructure. Without the necessary performance controls (i.e. quality-of-service), most storage systems are ill equipped to accommodate the mixed workloads across a multi-tenant environment. The SolidFire storage system has been architected specifically to address these issues.

To ensure that a SolidFire system can be fully leveraged within a cloud infrastructure requires integration with cloud management and orchestration technologies like Apache CloudStack. In these environments, storage is often interfaced, provisioned and managed through a platform like CloudStack. In its current form CloudStack allows an administrator to provision two types of storage:

- **{*}Primary Storage{}**: Primary storage is configured at the cluster level and is intended to store the disk volumes of all the virtual machines (VMs) in the associated cluster. A cluster consists of multiple hosts of similar type running a common hypervisor with dedicated storage to the cluster. The biggest issue with this configuration is the performance variability from all the virtual machines on the cluster contending for the same Primary Storage resource.
- **{*}Secondary Storage{}**: Secondary storage is typically NFS or object storage configured at the Zone level (multiple clusters). Secondary Storage is also shared storage, but shared across the entire zone. The normal use case for Secondary Storage is to store ISOs, templates and snapshots. Housing less performance sensitive data types, it is typically not an issue that secondary storage is shared.

This configuration guide describes the step-by-step process for provisioning a shared Primary Storage pool with SolidFire in a CloudStack environment. This includes a guided walk-through of the provisioning process in CloudStack, the underlying hypervisor and SolidFire storage. The document covers manual configuration from the user interface as well as example code for automating the provisioning process (See Appendix A for code examples).

Following the walkthrough for creating a shared Primary Storage pool is an advanced Quality-of-Service section. Here we describe several approaches for configuring SolidFire with CloudStack to achieve quality-of-service (QoS) per disk in your cloud infrastructure.

Provisioning a Primary Storage Volume

The volume provisioning process requires configuration on the SolidFire storage system, in the hypervisor and in CloudStack. This guide will walk through the following key steps:

- Create a Volume in SolidFire
- Add a Volume to the Hypervisor
- Defining a Volume as Primary Storage in CloudStack

Creating a Volume in SolidFire

The first step in the volume provisioning process is to create a volume on the SolidFire storage system. SolidFire volumes can be created via the SolidFire UI or API. For this step in the walkthrough we will configure the volume using the SolidFire UI.

1. In the SolidFire UI, click on the Volumes tab and then click on the Create New Volume button in the UI, to open the "Create New Volume" window.
2. Enter the Volume Name (May be 1 to 64 characters in length).
3. Click on the tenant Account that will have access to the volume (if there are more than 50 names in the list, the dropdown list will not appear. Begin typing and the auto-complete function will display possible values for you to choose from).
4. Enter the volume Total Size (See the "Sizing a SolidFire Volume" section for additional guidance)

Note: When entering volume size, 1GB = 1 000 000 000 bytes, or 1GiB = 1 073 741 824

1. Select whether or not to enable 512k-byte emulation. This option is necessary to support operating systems that do not recognize native 4K drives such as VMware ESX. By default this is checked to provide 512-byte emulation.
2. Set Quality of Service Settings values or accept default values
 - a. Min IOPS – Minimum number of IOPS guaranteed to the volume; shared by all VMs and data disks provisioned on the volume
 - b. Max IOPS – Maximum number of IOPS allowed to the volume; shared by all VMs and data disks provisioned on the volume
 - c. Burst IOPS – Allow for short bursts over the Max IOPS allowed on the volume; shared by all VMs and data disks provisioned on the volume
3. Click on Create Volume

Sizing a SolidFire Volume

The current storage provisioning process within CloudStack assumes that VMs will share storage volumes. This means that VMs share the volume capacity and contend with each other for the IOPS available to that specific volume. In order to deliver more predictable performance to the VMs and better exploit SolidFire's QoS features, the administrator needs to create multiple primary storage pools. Each pool created will have its own QoS settings in support of the different service offerings.

For example an administrator could create Gold, Silver and Bronze service offerings in CloudStack where each service offering leveraged one or more SolidFire volumes with guaranteed capacity and IOPS. CloudStack would then provision the VM by service offering into the appropriate volume providing more predictable performance.

The sizing process outlined below provides guidance on how to size the capacity and IOPS of the SolidFire volume associated to each service offering (e.g. Gold, Silver and Bronze)

Sizing Process	Example
1. Define the max number of VMs in the cluster	500 VMs
1. Define the expected number of VMs of the cluster per tier. In this example we have 500 VMs separated into 3 tiers: Gold, Silver and Bronze. Gold – Number of VMs from the cluster allocated to Gold Silver – Number of VMs from the cluster allocated to Silver Bronze – Number of VMs from the cluster allocated to Bronze	50 VMs 100 VMs 350 VMs
1. Define the average VM size in GBs per tier. The VMs can be the same size or different sizes depending on the customer profile. {*}Gold -- Expected size of a VM in GBs for a Gold tier Silver – Expected size of a VM in GBs for a Silver tier Bronze – Expected size of a VM in GBs for a Bronze tier	300 GBs 100 GBs 50 GBs
1. Define the average number of IOPS each VM will require per tier. This is the sum of expected performance for all VMs in the tier. {*}Gold -- Expected performance of a VM in IOPS for a Gold tier Silver – Expected performance of a VM in IOPS for a Silver tier Bronze – Expected performance of a VM in IOPS for a Bronze tier	2,500 IOPS 1,000 IOPS 100 IOPS
1. Define the SolidFire volume size for each tier by taking the number of VMs in the tier * the average size of the VMs in the same tier. {*}Gold -- (50 VMs * 300 GBs) {*}Silver -- (100 VMs * 100 GBs) {*}Bronze -- (350 VMs * 50 GBs)	15,000 GBs 10,000 GBs 17,500 GBs
1. Determine the total number of IOPS needed to satisfy each tier by taking the number of VMs in the tier * the average IOPS per VM in the tier. {*}Gold -- (50 VMs * 2,500 IOPS) {*}Silver -- (100 VMs * 1,000 IOPS) {*}Bronze -- (350 VMs * 100 IOPS)	125,000 IOPS 100,000 IOPS 35,000 IOPS
1. Determine the number of volumes needed to satisfy each tier by taking the total IOPS per tier / the max IOPS of a SolidFire volume (15,000). {*}Gold -- (125,000 / 15,000) {*}Silver -- (100,000 / 15,000) {*}Bronze -- (35,000 / 15,000)	9 Vols/15,000 IOPS each* 7 Vols/15,000 IOPS each* 3 Vols/15,000 IOPS each*
1. Determine the size of each SolidFire volume per tier by taking the total number of GBs per tier and dividing by the number of volumes needed.**	

(*)Gold --* (15,00 GBs / 9 volumes) (*)Silver --* (10,000 GBs / 7 volumes) (*)Bronze --* (17,500 GBs / 3 volumes)	2,000 GBs per Volume* 2,000 GBs per Volume* 6,000 GBs per Volume*
Notes: * Quantity is Rounded Up ** Volumes will be thin provisioned on SolidFire	

Adjusting Volumes After Initial Configuration

Adding a Volume

- An additional volume is first created on SolidFire.
- The SolidFire volume is then added to the hypervisor as a new storage repository or datastore.
- The volume is added as primary storage in CloudStack and tagged by tier.

Extending a Volume

- Extending volumes will be supported in the SolidFire Element 5 OS release.
- It is recommended that additional volumes be created and added as primary storage prior to the Element 5 release.

Adjusting Performance IOPS

- IOPS on the SolidFire SAN can be changed dynamically (without notifying the hypervisor or CloudStack).
- Minimum IOPS values can be increased up to 15,000 on a single volume, but they cannot be over committed.
- If you have maxed out the Min IOPS value (15,000), but notice that the effective IOPS are running into your Max IOPS value, you can make use of the SolidFire UI or API to increase the number of Max IOPS to the volume. IOPS trends can be tracked via the UI or API, as well.
- If you notice your volume bursts often and runs into the Burst IOPS ceiling, you can also change this value dynamically on the SolidFire SAN with notifying the hypervisor or CloudStack.
- All IOPS values for a given volume can be changed dynamically.

Adding a Volume to the Hypervisor

After creating the volumes on SolidFire, you will need to pre-mount the storage on the hypervisor via the hypervisor tools (e.g. XenCenter or vSphere Client) to make it aware of the iSCSI target(s).

Setting up Primary Storage for XenServer

Adding a Storage Repository for XenServer

1. Type - Create an iSCSI-based Storage Repository.
2. Name - Name and describe the characteristics of the Storage Repository.
3. IP Address - Provide the target IP address of the SolidFire cluster in the Target Host field, fill in the CHAP credentials, discover the Target IQN, and select LUN 0.

Creating Multiple Volumes

To create additional storage repositories in the resource pool, select the Resource Pool and click on the “New Storage” button. This process is necessary for each SolidFire volume created.

Setting Up Primary Storage for VMware

The process for attaching a SolidFire volume to the ESX/ESXi hypervisor assumes that the VMware virtual infrastructure has already been defined into datacenter containers and host clusters. In VMware a cluster is a collection of ESX/ESXi hosts and associated virtual machines intended to work together as a unit. The Datacenter is logical aggregation of Clusters / Hosts, VMs, Networks and Datastores.

To configure SolidFire storage with VMware we will walkthrough the following process:

- Add an iSCSI HBA to each host
- Configure the iSCSI HBA for each host
- Add a DataStore to the Hypervisor

Add an iSCSI HBA for each host

1. Log into VMware vSphere Client using a vSphere client
2. Choose a host from the cluster
3. Select the Configuration tab for the host
4. Select Storage Adapters from Hardware panel
5. Click on the Add... button
6. Click on the Add Software iSCSI Adapter radio button
7. Click the OK button

Configure the iSCSI HBA for each host

1. Right click on the newly added iSCSI storage adapter and select “Properties.”
2. Select the Static discovery tab in the “Properties” dialog box

3. Click on the Add... button to add the "Add Static Target Server"
 - a. iSCSI Server: Fill in the Storage Virtual IP Address (SVIP) of the SolidFire Cluster
 - b. Port: 3260
 - c. iSCSI Target Name: This is the IQN (iSCSI Qualified Name) of the SolidFire volume to use for the VMware Datastore
4. If you are using a version of the SolidFire Element storage OS prior to version 5, CHAP authentication is required. In this situation, click on the "CHAP..." button.

CHAP Target Authenticates Host

1. Select option: Choose "Use CHAP"
2. Name: Fill in with the account "Username" from SolidFire
3. Secret: Fill in with Initiator Secret from SolidFire

Mutual CHAP (host authenticates target)

1. Select option: Choose "Use CHAP"
2. Name: Fill in with the account "Username" from SolidFire
3. Secret: Fill in with Target from SolidFire

Add a Datastore to the Hypervisor

1. Right click in vSphere client. The "File" menu will display.
2. Select Add Datastore.... the "Add Storage" wizard will display.
3. Click on a host IP address to select a host.
4. Click Next
5. Select Storage Type: Choose the "Disk/LUN" radio option and click "Next"
 - a. Select Disk/LUN: Select the target SolidFire iSCSI Volume and click "Next"
 - b. File System Version: Select the preferred version of VMFS and click "Next"
 - c. Current Disk Layout: New volumes will be blank, click "Next"
 - d. Properties: Provide a name for the Datastore, click "Next"
 - e. Formatting: Select "Use the Maximum available space" radio option and click "Next"
6. Verify settings and click "Finish"

Adding and Using Primary Storage in CloudStack

Once the hypervisor is aware of the SolidFire volumes, it is time to create the storage pool(s) in CloudStack and create service offerings to use the storage pool(s). In CloudStack, Primary Storage is associated with a single compute cluster.

Adding Primary Storage

To create a primary storage pool in CloudStack, log into CloudStack as an Admin and go to the Infrastructure tab. From the Infrastructure tab choose Primary Storage and click "Add Primary Storage" and configure it from the dialog box.

The configuration of the Primary Storage pool is slightly different for each hypervisor. The "Add Primary Storage" dialog box will have unique fields specific to the type of hypervisor.

Primary Storage Configuration for a XenServer Cluster

- Zone – The CloudStack geographical container being configured with SolidFire
- Pod – The CloudStack rack containing one or more clusters being configured with SolidFire
- Cluster – The CloudStack group of hosts being configured with SolidFire
- Name – This field should be descriptive, but does not have to be an IP Address.
- Protocol – Select "vmfs"
- SR Name-Label – The name of the XenServer Storage Repository that was created.
- Storage Tags – The field should be set to a name that uniquely identifies this Primary Storage.

Once this process is complete the volume will be listed in the Infrastructure tab. The process of creating a primary storage pool is required for each SolidFire Volume.

Primary Storage Configuration for a VMware Cluster

- Zone – Zone for the SolidFire cluster
- Pod – Pod for the SolidFire cluster
- Cluster – Cluster for the SolidFire
- Name – This field should be descriptive, but does not have to be an IP Address
- Protocol – Select PreSetup for the Protocol field as we have already configured XenServer to access the iSCSI volume of interest
- Server – IP address or domain name of the VMware vSphere Client Server that is managing the cluster
- vSphere Client Datacenter – Name of the datacenter container in vSphere Client
- vSphere Client Datastore – Name of the datastore in vSphere Client

Once this process is complete the volume will be listed in the Infrastructure tab. The process of creating a primary storage pool is required for each SolidFire Volume.

Using Primary Storage in CloudStack

Once Primary Storage has been defined in CloudStack it can be used for root disks or as additional data disk on a VM. In CloudStack the root disk primary storage pool (e.g. Gold, Silver, Bronze) is chosen during the definition of a "Compute Offering". A compute offering is simply a profile of an instance. Data disks on the other hand, can be added to an instance after it has been defined.

Primary Storage for a Compute Offering

As an admin in CloudStack, to attach Primary Storage to a Compute Offering you need to go to the Services Offerings tab and select "Add Compute Offering".

You can define the compute offering and associate it with the desired primary storage using the "Storage Tags" field in the dialog box. The storage tag, created during the "Defining a Volume as Primary Storage in CloudStack" step earlier in the sequence, is how primary storage is attached to the Compute Offering.

Launch an Instance based on the Defined SolidFire Volume

Once a compute offering has been defined, an instance can be launched that uses the associated primary storage pool (e.g. Gold, Silver, Bronze).

1. Go to the Instances Tab
2. Click on Add Instance
3. Follow the Add Instance wizard and select the Compute Offering that leverages the desired SolidFire Volume

Primary Storage for a Data Disk

As an admin in CloudStack, to attach Primary Storage to a Data Disk offering go to the Services Offerings tab and choose "Add Disk Offering". Define the disk offering from the dialog box. As with the Compute Offering process described above, the "Storage Tags" field is how primary storage is attached to the profile.

Adding a Volume to an Instance

Once a data disk profile has been created from the "Create Disk Offering" process a volume from that primary storage pool can then be attached to an instance.

1. As an Admin in CloudStack, go to the Storage tab
2. Complete the Add Volume dialog box
 - a. Name – Provide a name for the volume
 - b. Availability Zone – Zone for the SolidFire cluster
 - c. Disk Offering – Select the desired volume
3. Select the Volume that was created
4. Select the Instance and attach the disk

Advanced QoS Configuration

The tiering scheme outlined above, while yielding more predictable performance per tier, still has multiple virtual machines contending for resources within each tier. To achieve a more granular level of QoS underneath CloudStack requires the creation of a SolidFire volume per VM / Disk. There are two clear benefits from this level of provisioning granularity:

- Each VM / Disk would have a dedicated SolidFire volume
- Each SolidFire volume would deliver guaranteed IOPS

In its current form, the process for providing QoS per VM in CloudStack requires a workflow outside of CloudStack. To access this level of QoS granularity there are two options to consider, one for each of the possible primary storage offerings within CloudStack; Compute and Data Disk. For creating a 'Compute Offering' that uses a dedicated SolidFire volume with defined QoS we have created a workflow called On-Demand Create. For creating a Data Disk Offering that can then be attached to a VM as a data disk, we have created a workflow called The On-Demand Attach. Both of these processes are outlined in detail below.

Option 1: On-Demand Create Process

1. Create a SolidFire volume using the "CreateVolume" API command. You receive a volume ID in response to the successful execution of this command.
2. Retrieve the IQN of the newly created volume by querying the SolidFire cluster using the "ListActiveVolumes" API command, passing in the volume ID you received a step earlier (to be used as the start volume ID) and limiting the response to a single volume.
3. To retrieve CHAP credentials for the volume, execute the SolidFire API command GetAccountByID (all volumes for a given Account ID share the same CHAP credentials).
4. CloudStack does not directly support CHAP credentials, so you must communicate via API with your hypervisor software to set up access from the hypervisor to the iSCSI target. For XenServer, this means creating a storage repository based on the iSCSI volume. For VMware, this means creating a datastore based on the iSCSI volume. For KVM, this means setting up a shared mount point based on the iSCSI volume.
5. Create a Primary Storage pool in CloudStack by issuing the createStoragePool API command. Give this Primary Storage a sufficiently unique name (ex. SolidFire736256477, where the number portion is pseudo random). Assign the Primary Storage's Storage Tags field the same value as its name. The type of this Primary Storage is PreSetup for XenServer, vmfs for VMware, and SharedMountPoint for KVM.
6. Create a Compute Offering in CloudStack by issuing the createServiceOffering API command. Give this Compute Offering a sufficiently unique name as was done for the Primary Storage created above. Assign the Compute Offering's Storage Tags field the same value as you assigned to the Storage Tags field of the Primary Storage created above.
7. Spin up a VM via CloudStack by issuing the deployVirtualMachine API command, passing in the ID of the Compute Offering created above.
8. Once the VM has been deployed, you may delete the Compute Offering created above.

Option 2: On-Demand Attach Process

Same as the On-Demand Create process with the following exceptions:

- In step 6, instead of creating a Compute Offering, you will create a Data Disk Offering. The CloudStack API call is `createDiskOffering`.
- In step 7, instead of spinning up a VM, you will create and attach a data disk. The CloudStack API calls are the following: `createVolume` and `attachVolume`.
- In step 8, instead of deleting the Compute Offering, you can delete the Data Disk Offering.

Achieved outside of CloudStack today, SolidFire is working to ensure these workflows and the resulting QoS provisioning granularity are accessible natively through CloudStack in future releases.

Summary

By now you should have a stronger familiarity with current recommended best practices for provisioning a SolidFire storage system, and its advanced QoS functionality, within a CloudStack-based infrastructure. Both Apache CloudStack and the SolidFire are dynamic and rapidly evolving platforms that will continue to enhance the ability to provision, manage and automate storage resources in a cloud infrastructure.