RBD improvements based on new Storage subsystem

Current RBD implementation

The current RBD implementation only works under KVM since KVM is the only hypervisor which supports RBD as a block device.

Currently you can use RBD for both root and datadisk volumes, but you can't snapshot or backup these.

This is due to the way CloudStack currently works, it tries to backup a snapshot directly to Secondary Storage which doesn't work with RBD. (qemu-img doesn't play nice).

Information about RBD: http://ceph.com/docs/master/rbd/rbd/

Goals for 4.2

The 4.2 release of CloudStack will introduce a new storage subsystem (Storage+subsystem+2.0), this will allow us to implement the remaining features for RBD.

New RBD features

Snapshotting

In the new subsystem snapshots have been de-coupled from backups, so it allows us to snapshot volumes without backing them up to Secondary Storage. The snapshot will stay inside the Ceph cluster.

Backups

When a backup is created from a snapshot it will extract the snapshot from the RBD primary storage and copy it in RAW or QCOW2 (to be determined) format to Secondary Storage.

Cloning / Layering

One of the features recently introduced in RBD (Since Ceph version 0.55) is cloning. It allows for one "golden" image where VMs boot from, but their write actions (deltas) are stored in their own image.

This way it enables a admin to spin up hundreds of virtual machines without copying data, it can happen instantly.

Communication with the Ceph cluster

The new Storage subsystem allows for plugins to directly communicate with a storage API without going through the hypervisor.

In the current RBD implementation the creation of volumes is done by libvirt on the hypervisor. Although it might be easier to do this directly from the management server it should not be done. This would require a L3 link from the management server directly to the Ceph monitors which could compromise the whole Ceph cluster when the management server becomes compromised.

By keeping all this communication go through the hypervisors we can abstract this in the APIs and the Ceph cluster can stay "buried" deep in the network.

libvirt (with the RBD storage pool backend) can still do the heavy lifting for creating the RBD volumes, snapshotting and maybe even the cloning. The last will probably involve a update of the libvirt storage driver.

libvirt vs 'rbd' cli tool

The best approach would to avoid the usage of the 'rbd' cli tool by the CloudStack agent.

Although this tool can do everything we want, it will involve executing a CLI tool which can cause issues, not to mention the Cephx credentials which will show up in the process list.

If libvirt can't offer everything we need it might be useful to wrap librbd into Java bindings.