oVirt

RAM Snapshots

21.10.2013

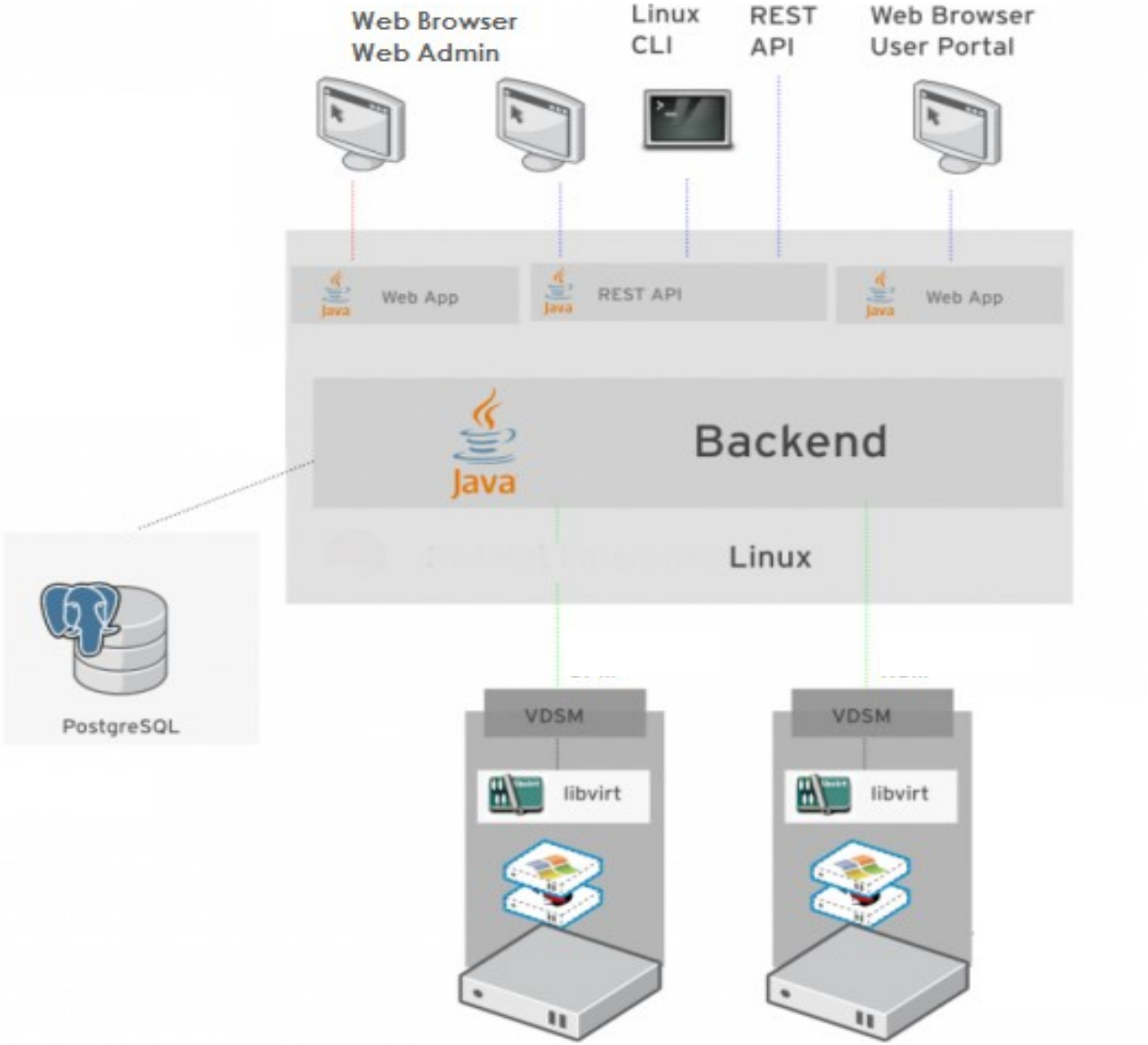
Arik Hadas
Red Hat

Agenda

oVirt

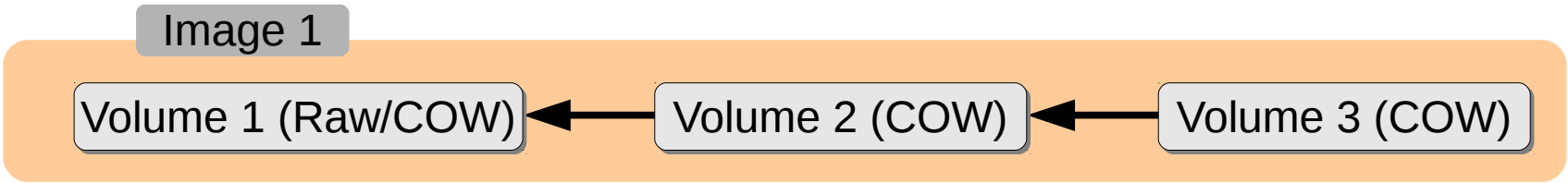
- Quick overview of oVirt architecture & volume types
- Overview of snapshots in oVirt
- Deep dive into RAM snapshot feature
  - The concept
  - Implementation details
  - How to use

oVirt architecture



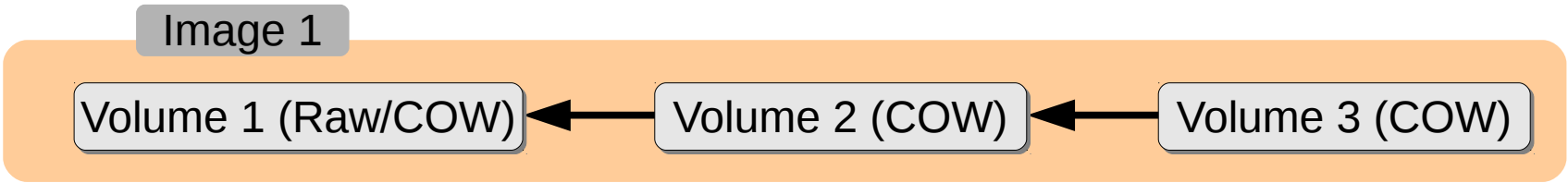
Virtual disk structure

- Virtual disk image is composed of volumes
- Volume Types
  - Raw - plain binary data
  - COW – only data that was changed
    - QCOW2 (QEMU Copy On Write 2)



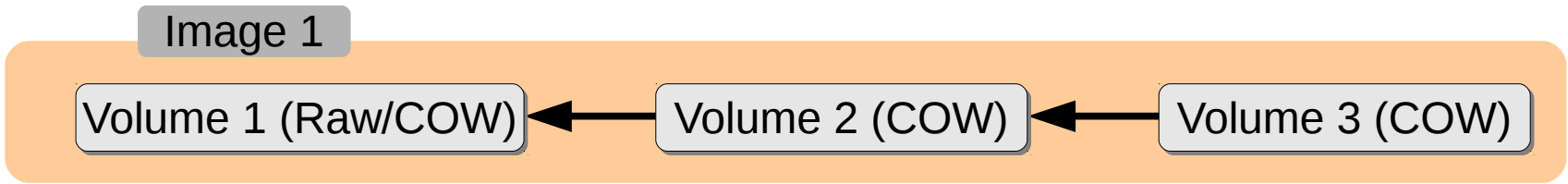
Write data to disk with COW volume

- The data will be stored in the last COW volume
- COW volumes contain the data that was changed after they were created
- All the volumes except the last one become read-only



Read data from disk with COW volume

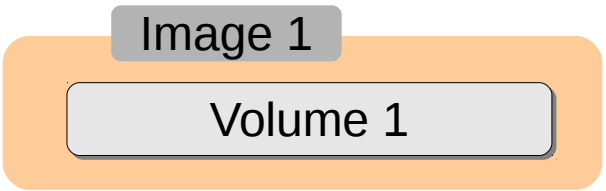
- We will try to read the data from the last volume
  - The volume contains meta-data that indicates which data exist in the volume
- If the data exist in the volume
  - We'll read it from the volume
- Otherwise
  - We'll try to read it from the 'parent' volume



Snapshot usages in oVirt (1)

- **Backup & Restore**
  - Create snapshot
  - Preview snapshot
  - Commit to snapshot

Snapshot Name	Volumes
Active VM	Volume 1

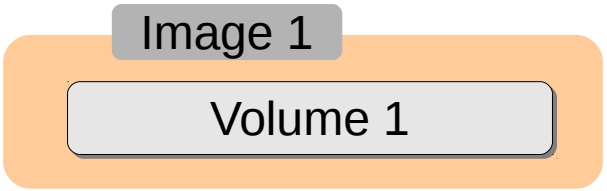


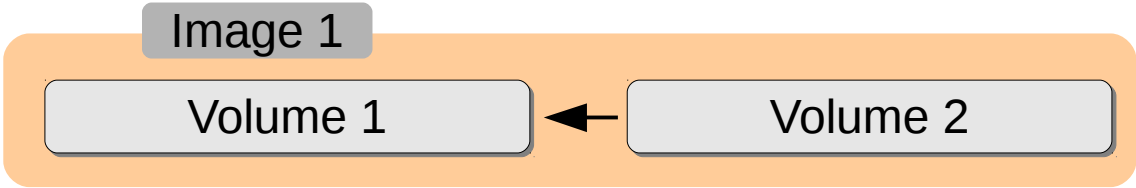
Snapshot usages in oVirt (1)

- **Backup & Restore**
  - **Create snapshot**
  - Preview snapshot
  - Commit to snapshot

Snapshot Name	Volumes
Active VM	Volume 1

Snapshot Name	Volumes
Active VM	Volume 2
Snapshot 1	Volume 1



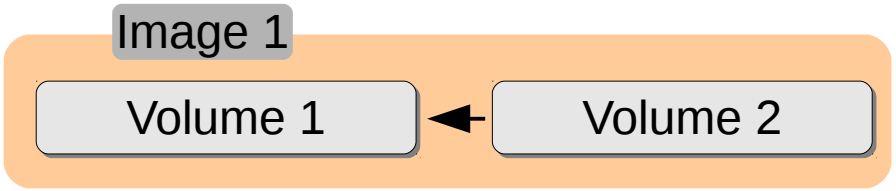


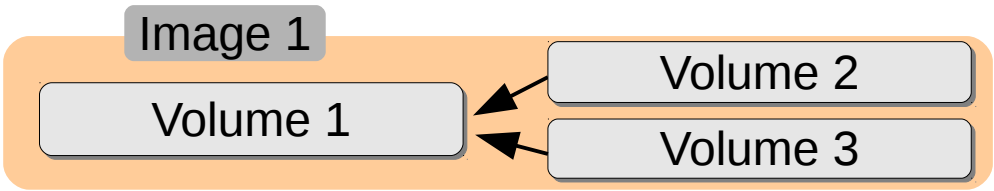
Snapshot usages in oVirt (1)

- **Backup & Restore**
  - Create snapshot
  - **Preview snapshot**
  - Commit to snapshot

Snapshot Name	Volumes
Active VM	Volume 2
Snapshot 1	Volume 1

Snapshot Name	Volumes
Active VM	Volume 3
Previous Active VM	Volume 2
Snapshot 1	Volume 1



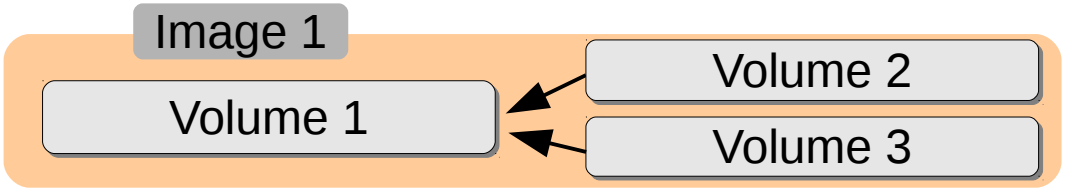


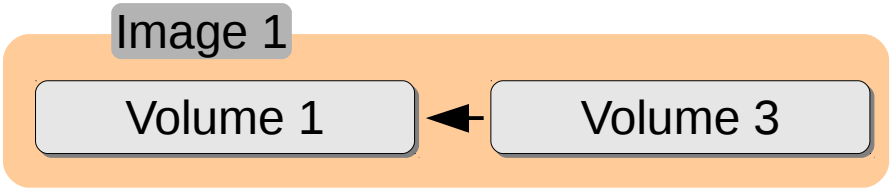
Snapshot usages in oVirt (1)

- **Backup & Restore**
  - Create snapshot
  - Preview snapshot
  - **Commit to snapshot**

Snapshot Name	Volumes
Active VM	Volume 3
Previous Active VM	Volume 2
Snapshot 1	Volume 1

Snapshot Name	Volumes
Active VM	Volume 3
Snapshot 1	Volume 1

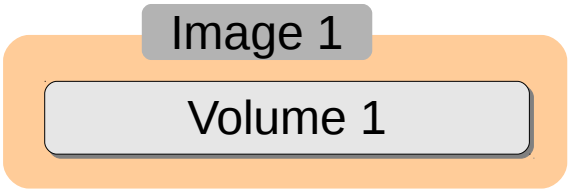




Snapshot usages in oVirt (2)

Stateless VM implementation

Snapshot Name	Volumes
Active VM	Volume1

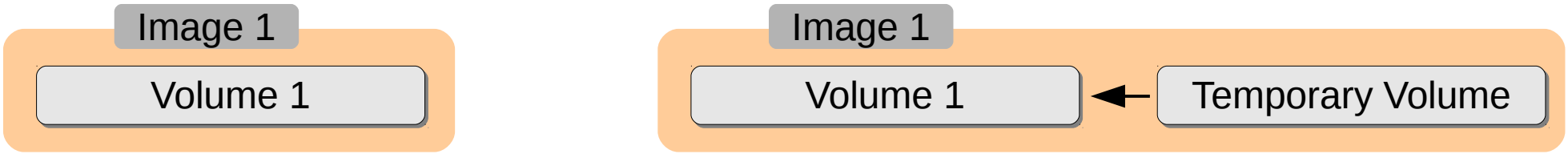


Snapshot usages in oVirt (2)

Stateless VM implementation

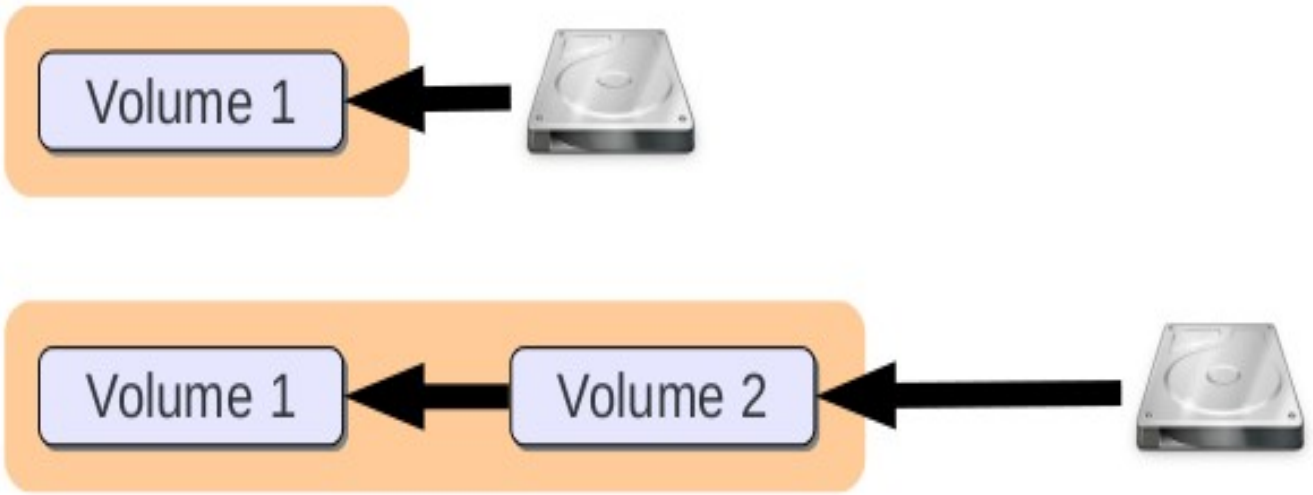
Snapshot Name	Volumes
Active VM	Volume1

Snapshot Name	Volumes
Active VM	Temporary volume
Stateless snapshot	Volume 1



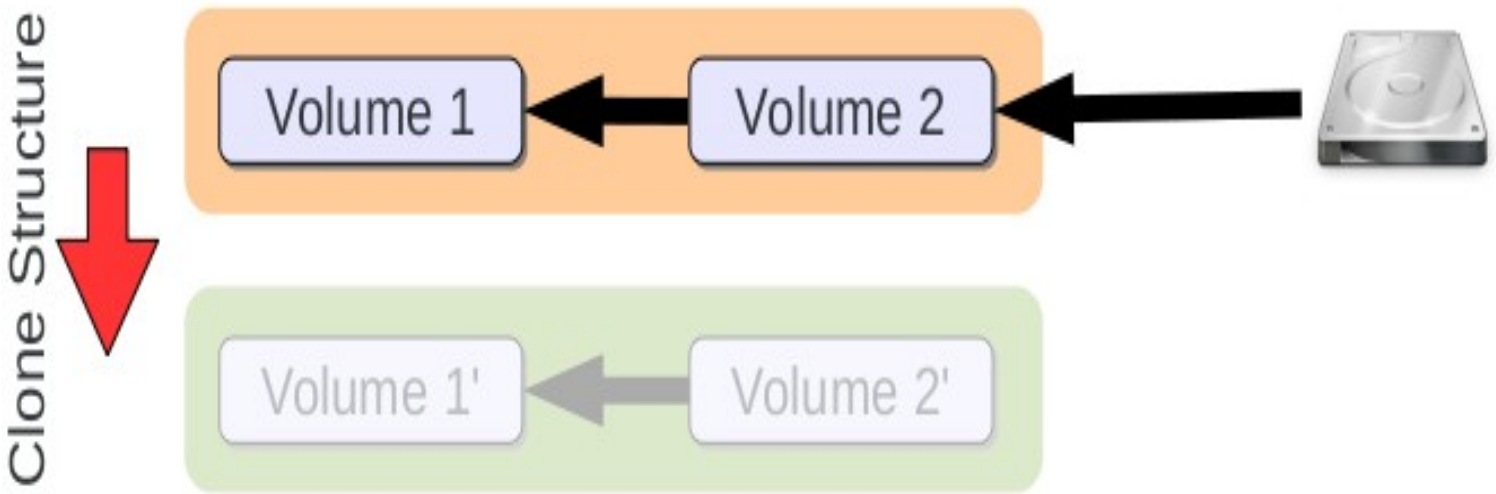
Snapshot usages in oVirt (3)

Preliminary step in Live Storage Migration process



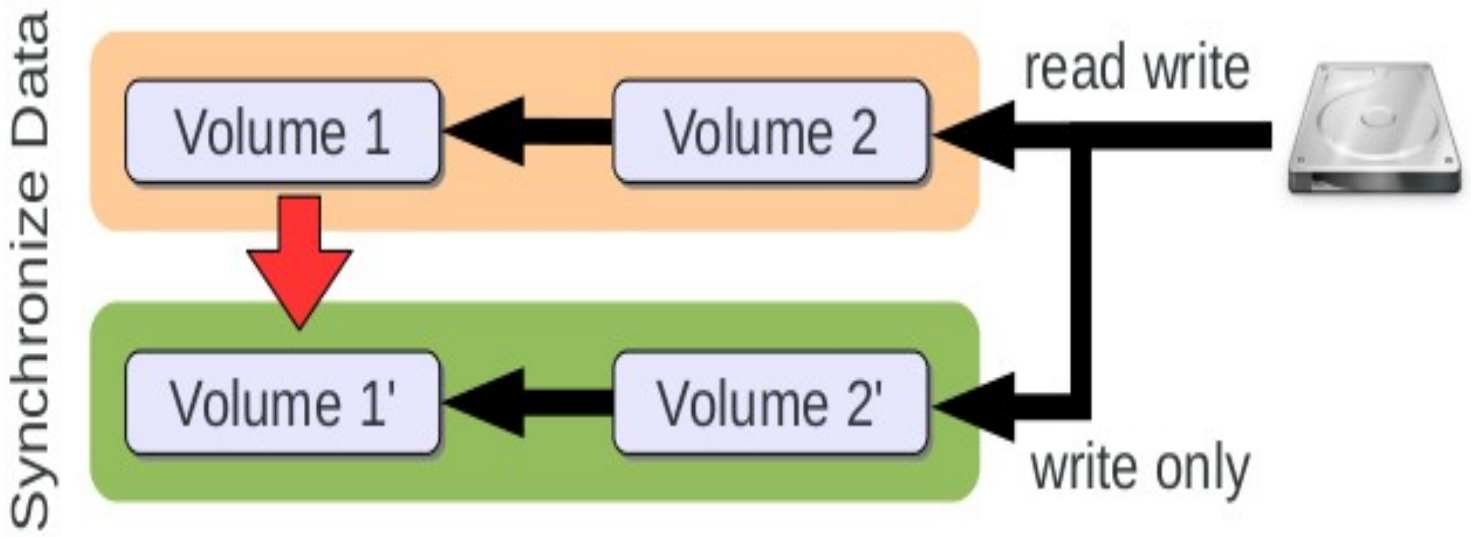
Snapshot usages in oVirt (3)

Preliminary step in Live Storage Migration process



Snapshot usages in oVirt (3)

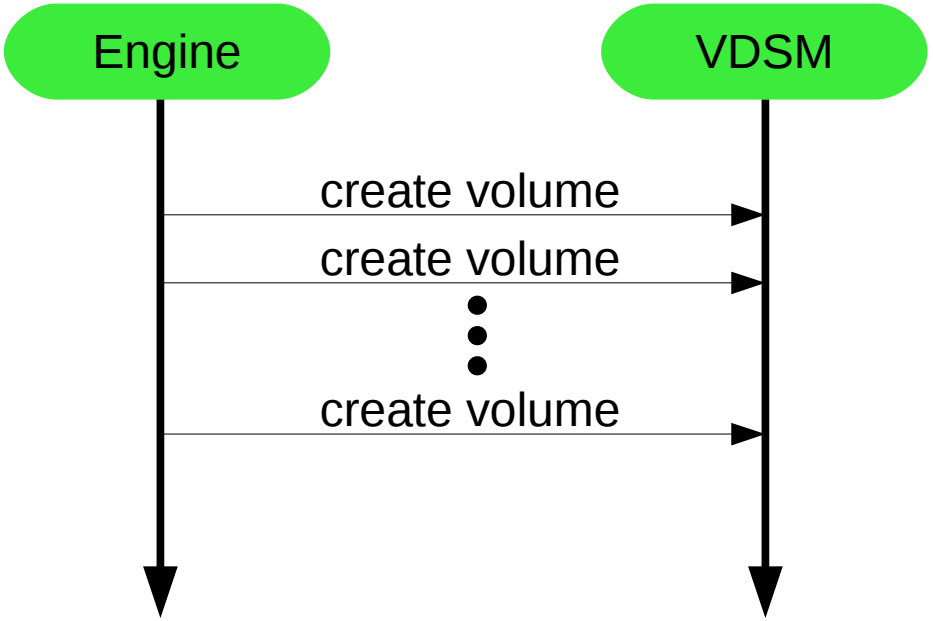
Preliminary step in Live Storage Migration process



Snapshot Types in oVirt

Offline Snapshot

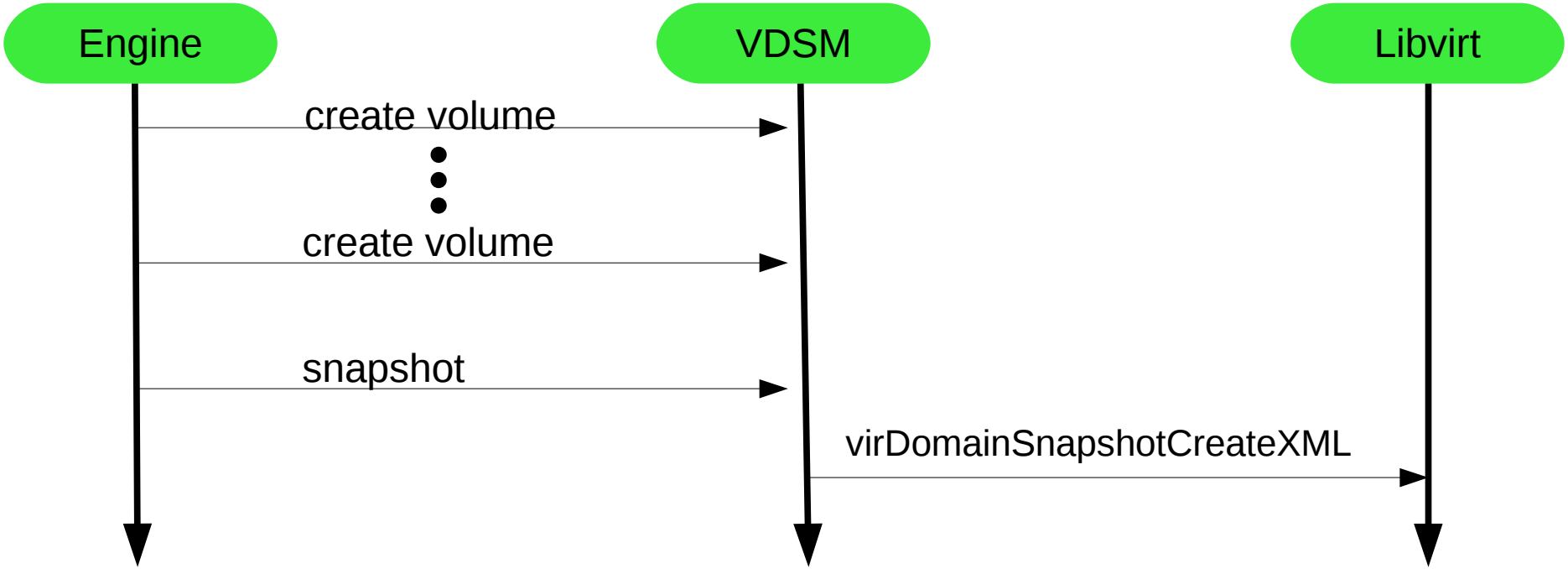
- Taking snapshot for VM which is not running
- Add volume for each of the VM disks
- The added volumes will be used on next run



Snapshot Types in oVirt

Live Snapshot

- Taking snapshot for running VM
- Add volume for each of the VM disks
- The VM switch to the added volumes



Libvirt API

virDomainSnapshotCreateXML (VM, xml, flags)

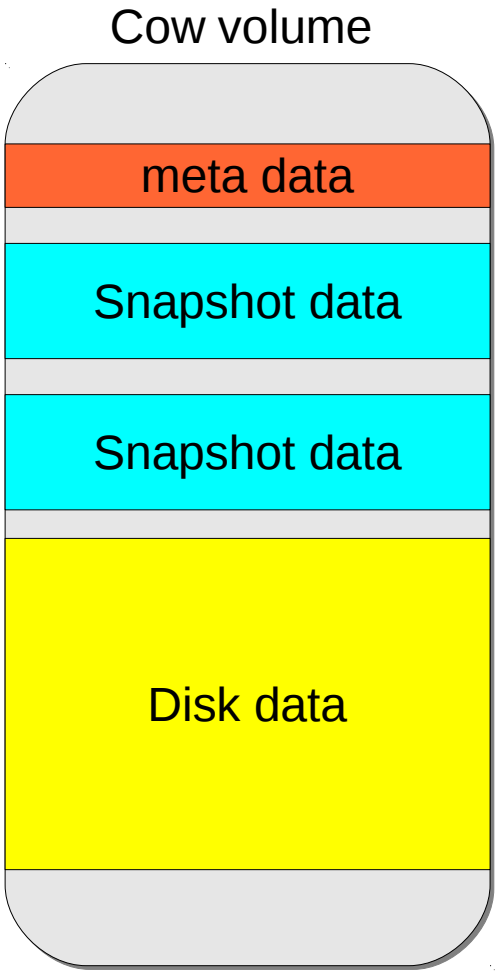
- VM: VM ID
- xml: snapshot properties in xml format
- flags: snapshot properties as flags

Snapshot types in libvirt

- Disks snapshot
  - Internal
  - External
- Memory state (VM state)
  - Piggy-backed
  - External
- System checkpoint
  - Disk snapshot + Memory state

Snapshot types in libvirt

- Disks snapshot
  - **Internal**
  - External
- Memory state (VM state)
  - Piggy-backed
  - External
- System checkpoint
  - Disk snapshot + Memory state



Snapshot types in libvirt

- Disks snapshot
  - Internal
  - External
- Memory state (VM state)
  - Piggy-backed
  - External
- System checkpoint
  - Disk snapshot + Memory state

Live snapshot in libvirt

oVirt

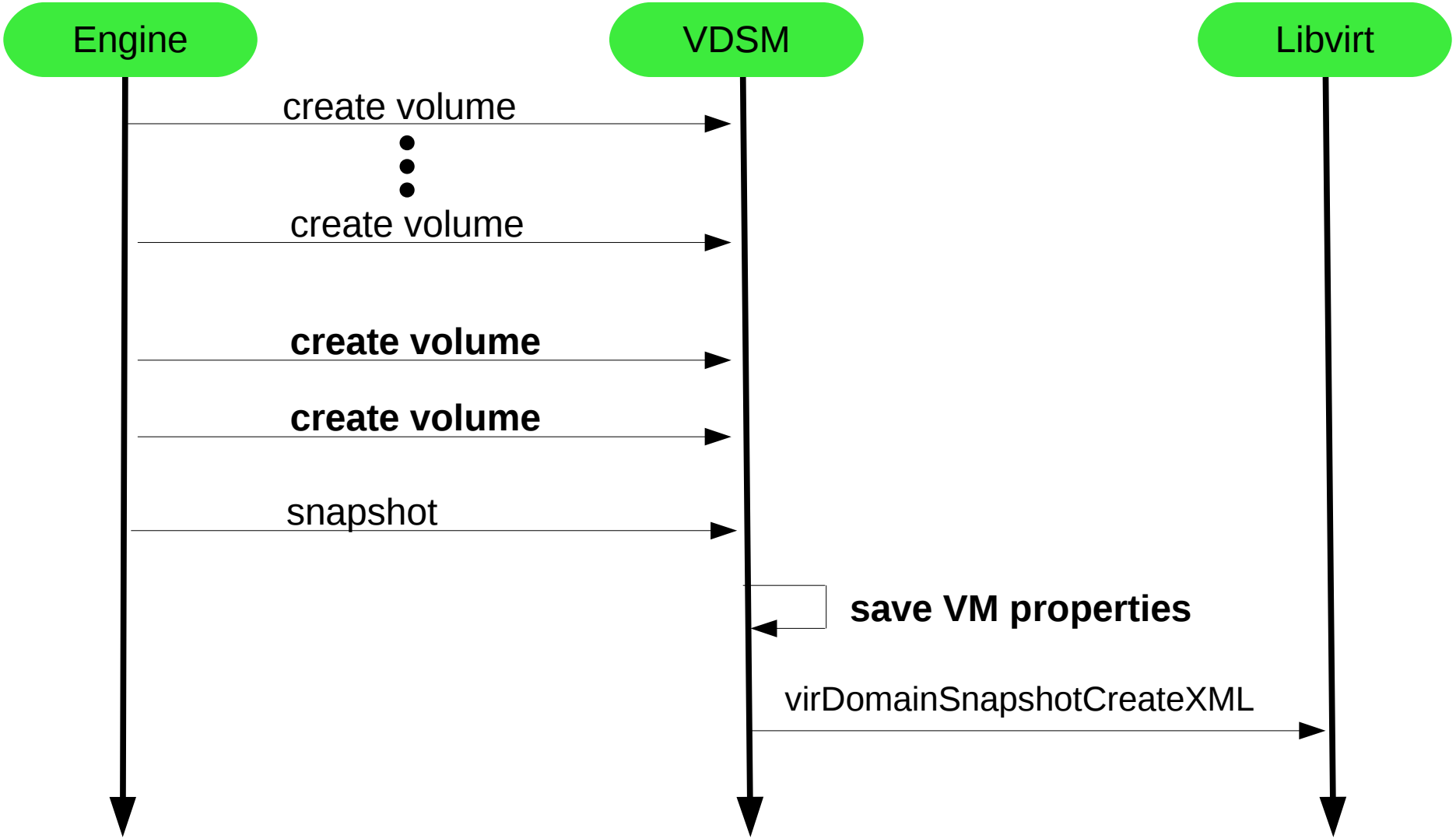
- External disks snapshot
- xml
  - Volume to switch to for each disk
- flags
  - VIR_DOMAIN_SNAPSHOT_CREATE_DISK_ONLY
  - VIR_DOMAIN_SNAPSHOT_CREATE_REUSE_EXT
  - VIR_DOMAIN_SNAPSHOT_CREATE_NO_METADATA
  - VIR_DOMAIN_SNAPSHOT_CREATE_QUIESCE

RAM Snapshots

oVirt

- Disks snapshot with memory state
- When running VM that was reverted to RAM snapshot
  - The saved state (memory + disks) is restored
  - TCP connections might time out
- Unlike hibernation the VM remains active

Creating RAM Snapshots



RAM snapshots in libvirt

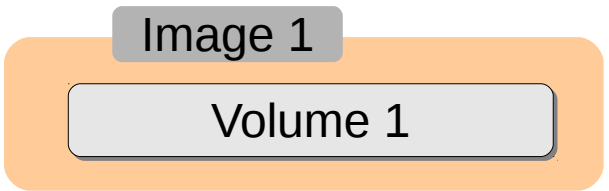
oVirt

- System checkpoint
  - External disks snapshot
  - **External memory state**
- xml
  - Volume to switch to for each disk
  - **Volume to save the memory state in**
- flags
  - **VIR_DOMAIN_SNAPSHOT_CREATE_LIVE**
  - VIR_DOMAIN_SNAPSHOT_CREATE_REUSE_EXT
  - VIR_DOMAIN_SNAPSHOT_CREATE_NO_METADATA

RAM snapshot usages

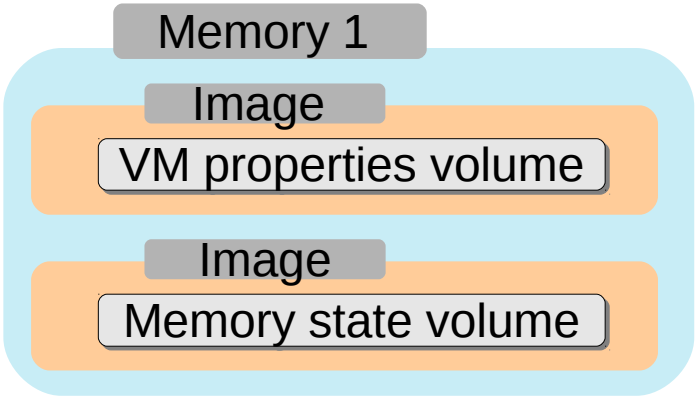
- Create snapshot with memory
- Preview snapshot with memory
- Commit to snapshot with memory
- Stateless VM with initial memory

Snapshot Name	Volumes	Memory
Active VM	Volume 1	



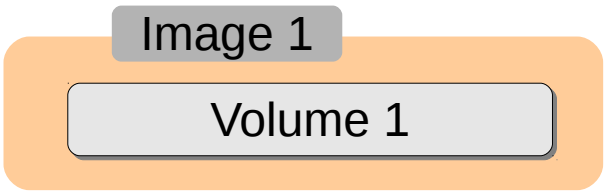
RAM snapshot usages

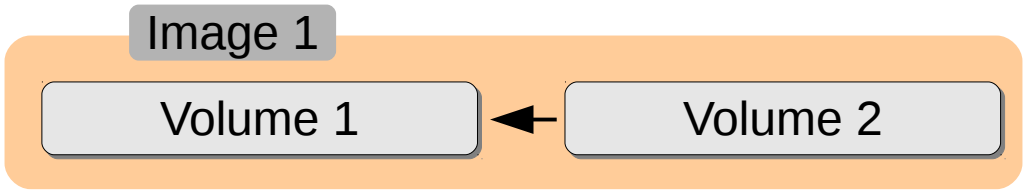
- **Create snapshot with memory**
- Preview snapshot with memory
- Commit to snapshot with memory
- Stateless VM with initial memory



Snapshot Name	Volumes	Memory
Active VM	Volume 1	

Snapshot Name	Volumes	Memory
Active VM	Volume 2	
Snapshot 1	Volume 1	Memory 1



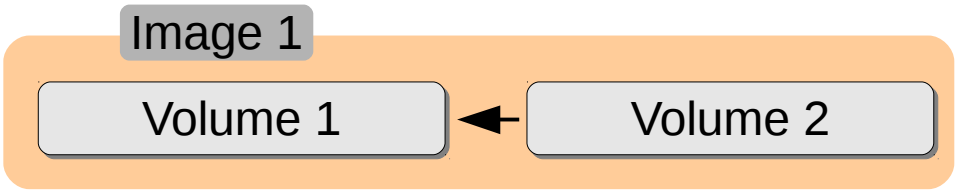


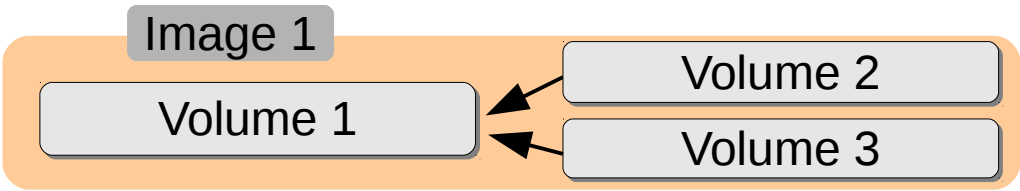
RAM snapshot usages

- Create snapshot with memory
- **Preview snapshot with memory**
- Commit to snapshot with memory
- Stateless VM with initial memory

Snapshot Name	Volumes	Memory
Active VM	Volume 2	
Snapshot 1	Volume 1	Memory 1

Snapshot Name	Volumes	Memory
Active VM	Volume 3	Memory 1
Previous Active VM	Volume 2	
Snapshot 1	Volume 1	Memory 1

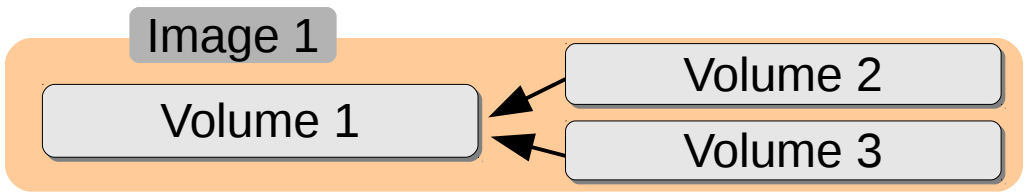




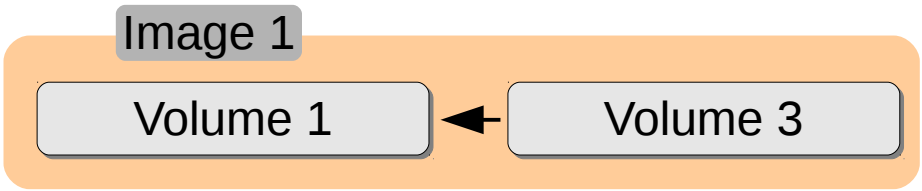
RAM snapshot usages

- Create snapshot with memory
- Preview snapshot with memory
- **Commit to snapshot with memory**
- Stateless VM with initial memory

Snapshot Name	Volumes	Memory
Active VM	Volume 3	Memory 1
Previous Active VM	Volume 2	
Snapshot 1	Volume 1	Memory 1



Snapshot Name	Volumes	Memory
Active VM	Volume 3	Memory 1
Snapshot 1	Volume 1	Memory 1

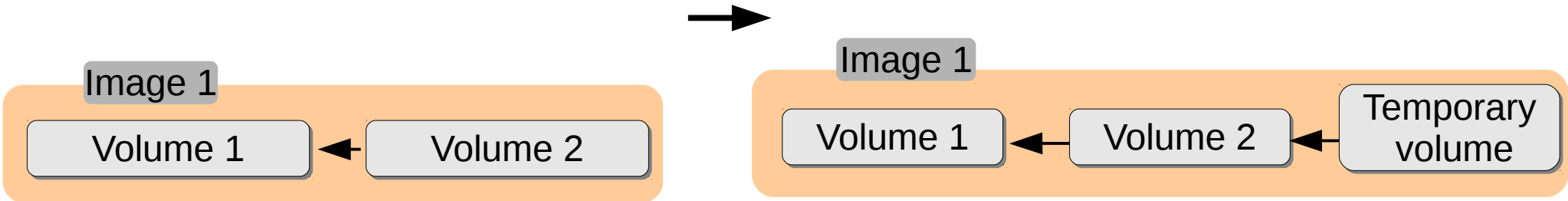


RAM snapshot usages

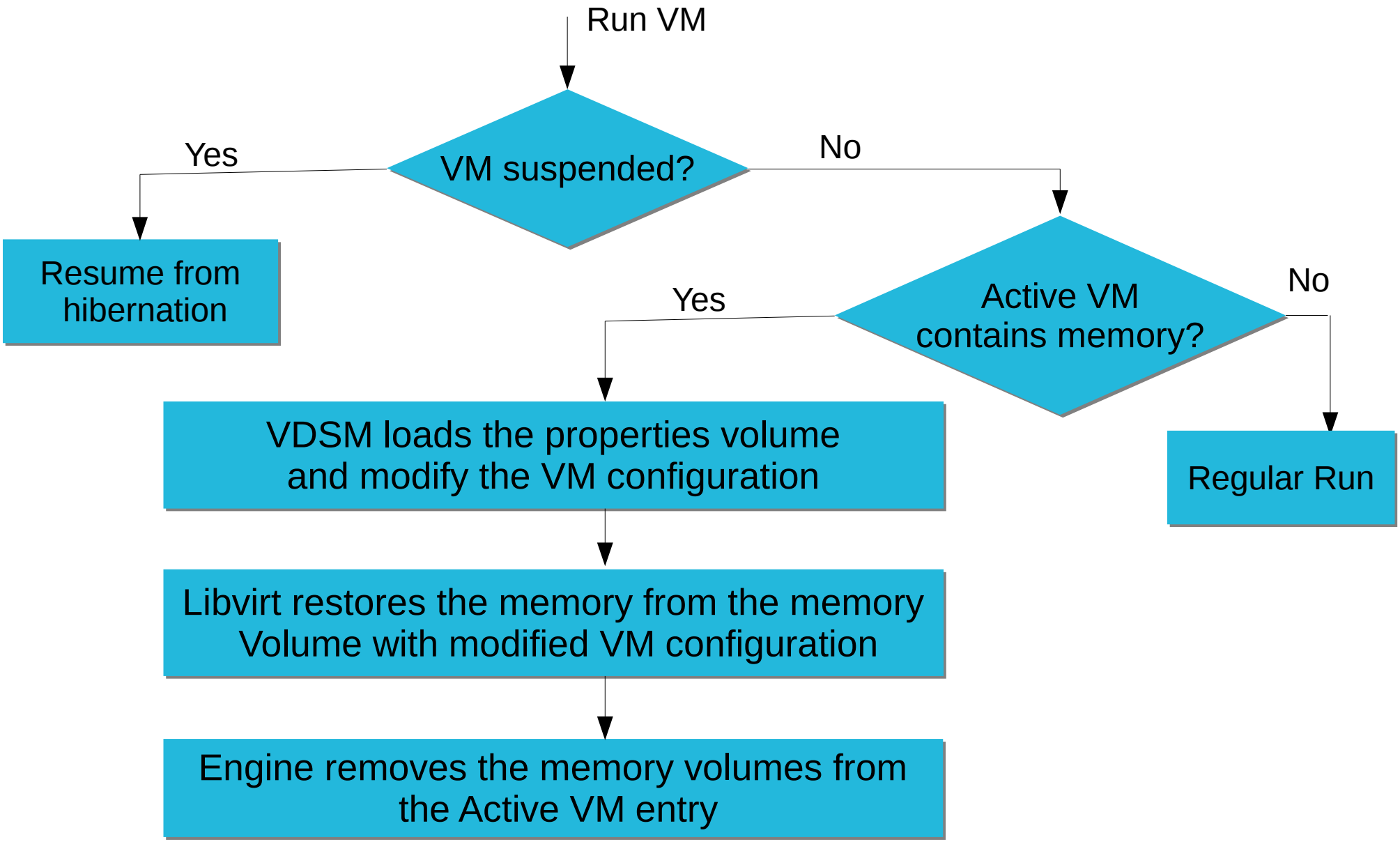
- Create snapshot with memory
- Preview snapshot with memory
- Commit to snapshot with memory
- **Stateless VM with initial memory**

Snapshot Name	Volumes	Memory
Active VM	Volume 2	Memory 1
Snapshot 1	Volume 1	Memory 1

Snapshot Name	Volumes	Memory
Active VM	Temporary volume	Memory 1
Stateless snapshot	Volume 2	Memory 1
Snapshot 1	Volume 1	Memory 1



Run VM – using RAM snapshot

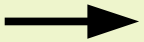


Why do we need to change volumes?

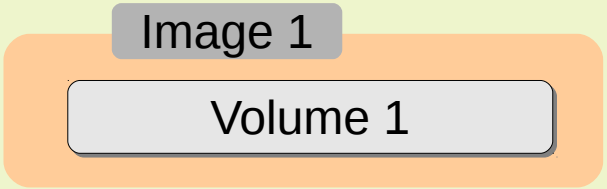
oVirt

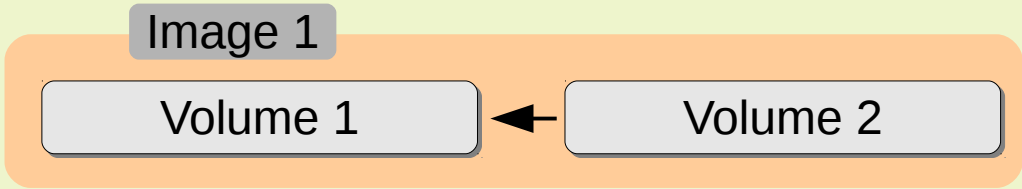
When creating the snapshot:

Snapshot Name	Volumes	Memory
Active VM	Volume 1	



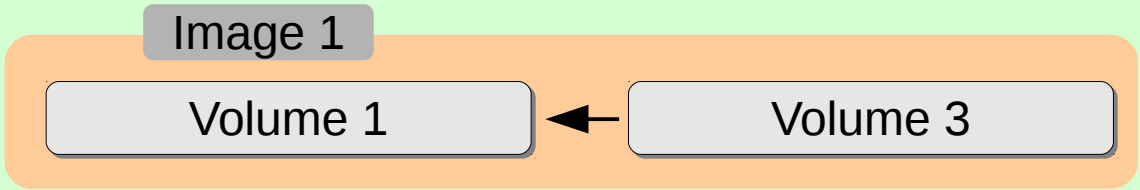
Snapshot Name	Volumes	Memory
Active VM	Volume 2	
Snapshot 1	Volume 1	Memory 1



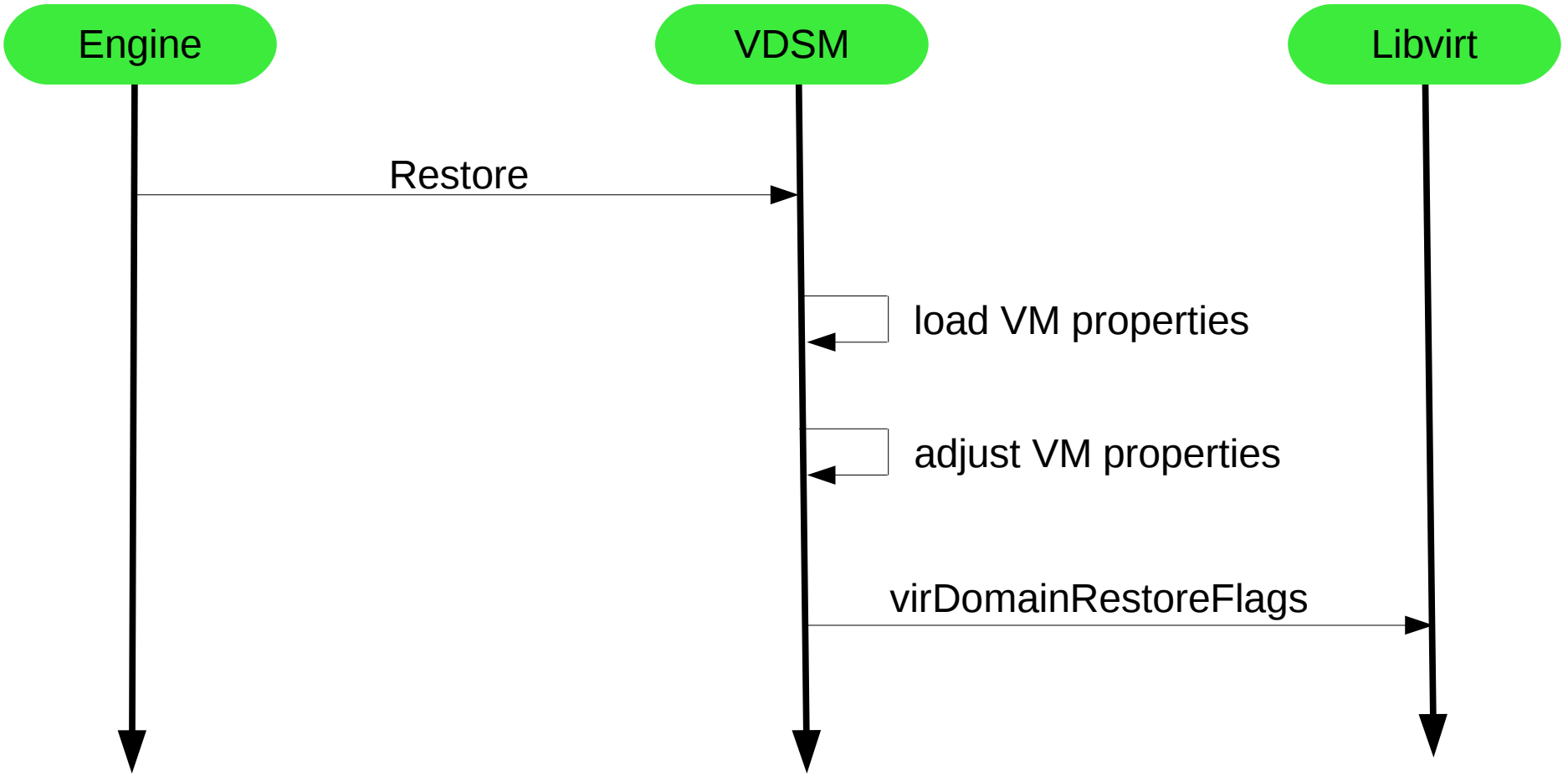


After committing to the snapshot:

Snapshot Name	Volumes	Memory
Active VM	Volume 3	Memory 1
Snapshot 1	Volume 1	Memory 1



Run VM – using RAM snapshot



Snapshot with memory state in libvirt (2) oVirt

virDomainRestoreFlags (VM, memory _state, xml, flags)

- Allows to alter specific portion of the VM configuration
  - Change active volumes
  - Change DC & cluster settings

OS Level: memory size, cpu type, disks, network cards..

Virtualization

Host level: disk volumes, networks..

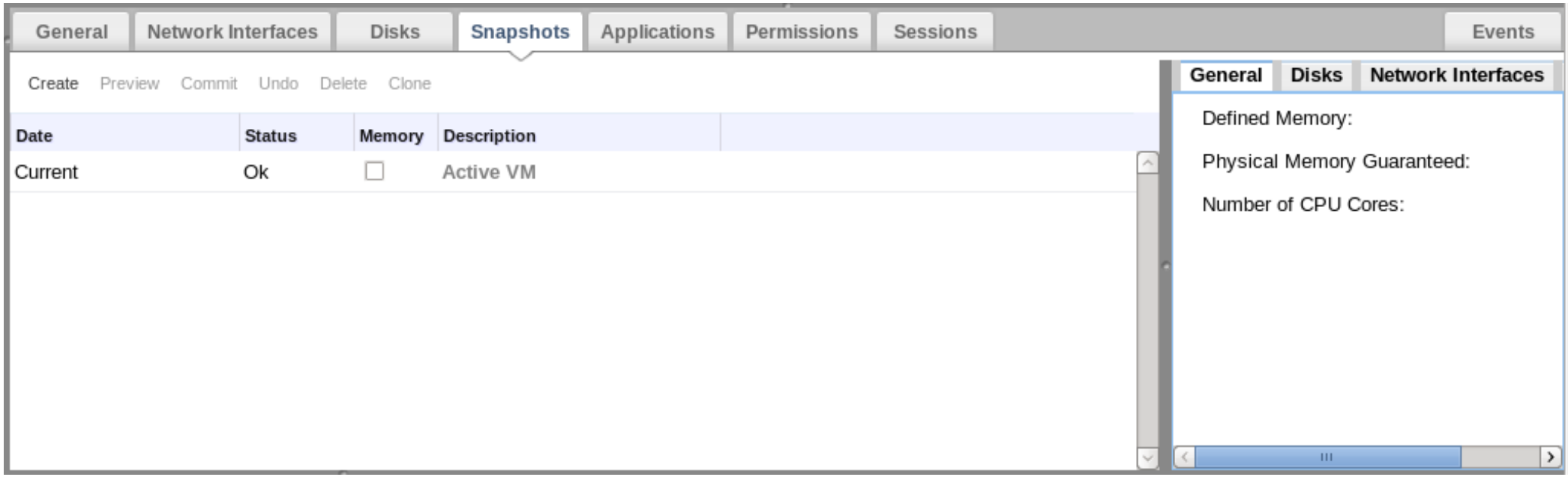
Export/Import VM with RAM snapshots

- RAM snapshots can be exported & imported
  - Except export/import with copy collapse set
- When copy collapse is set, snapshots are dropped
  - Including their memory volumes
  - Except the Active VM entry

Snapshot	Volumes	Memory
Active VM	Volume 3	Memory 2
Snapshot 2	Volume 2	Memory 2
Snapshot 1	Volume 1	Memory 1

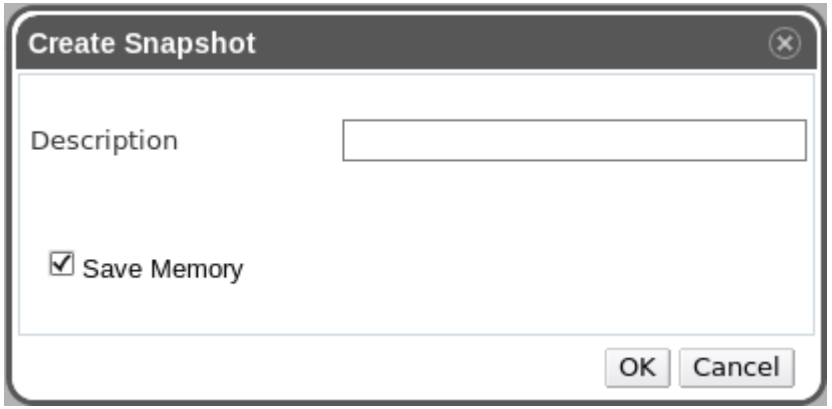
Snapshot	Volumes	Memory
Active VM	Volume’	Memory 2’

VM with no snapshots

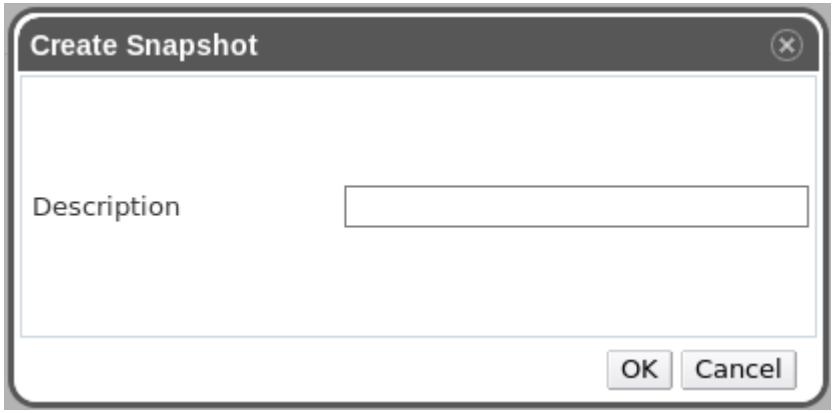


Create snapshot dialog

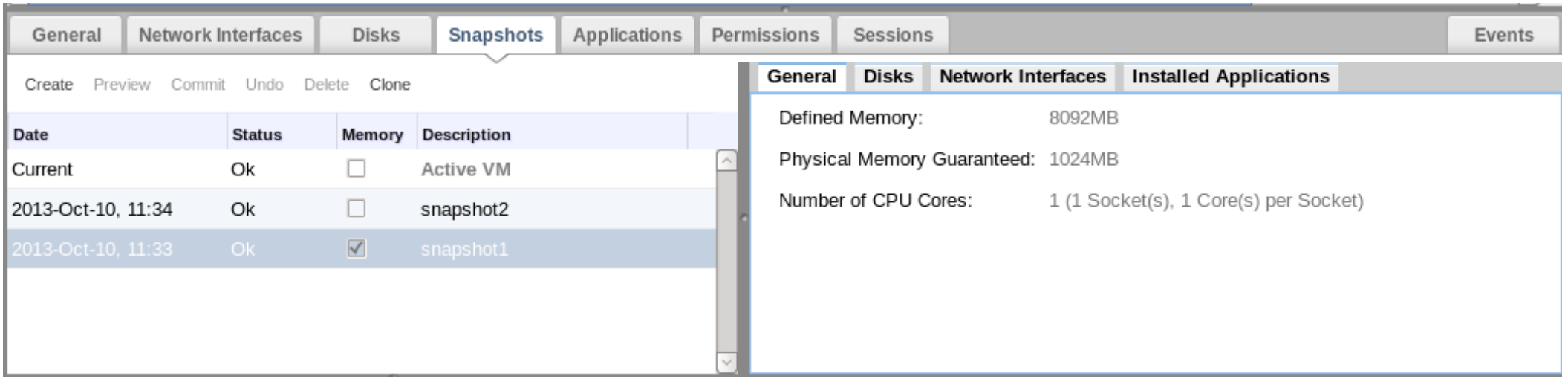
VM is running:

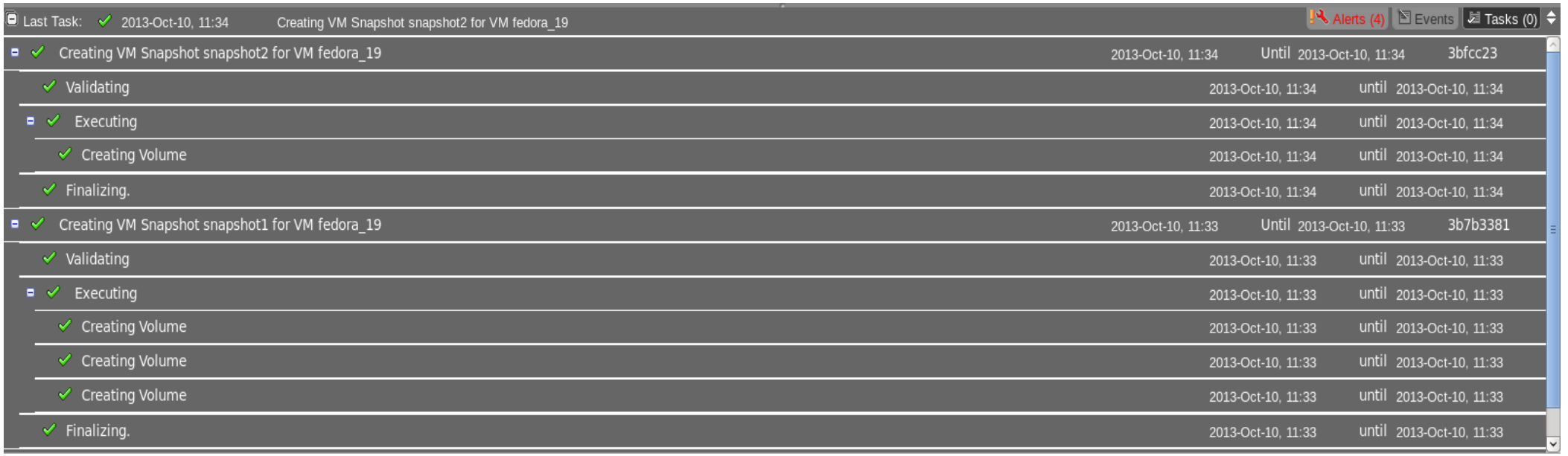


VM is not running:

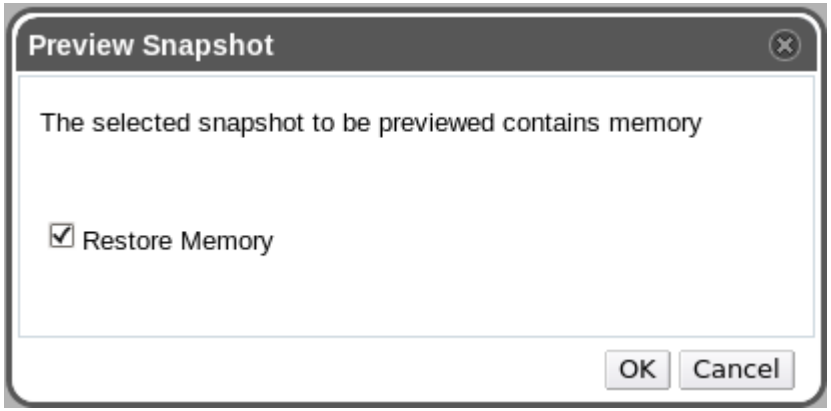


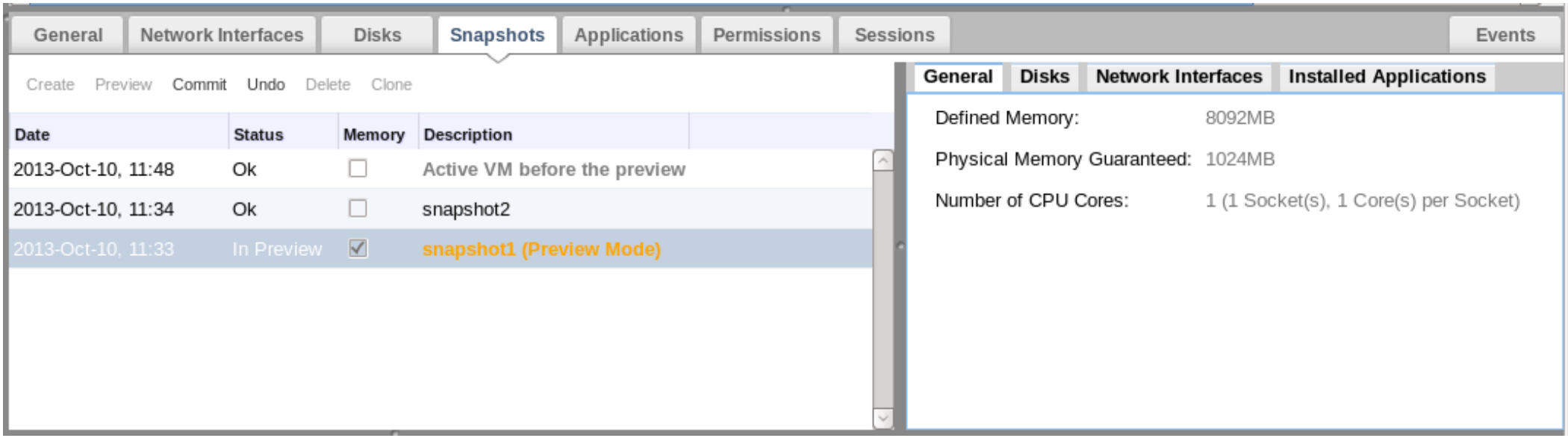
Snapshots creation



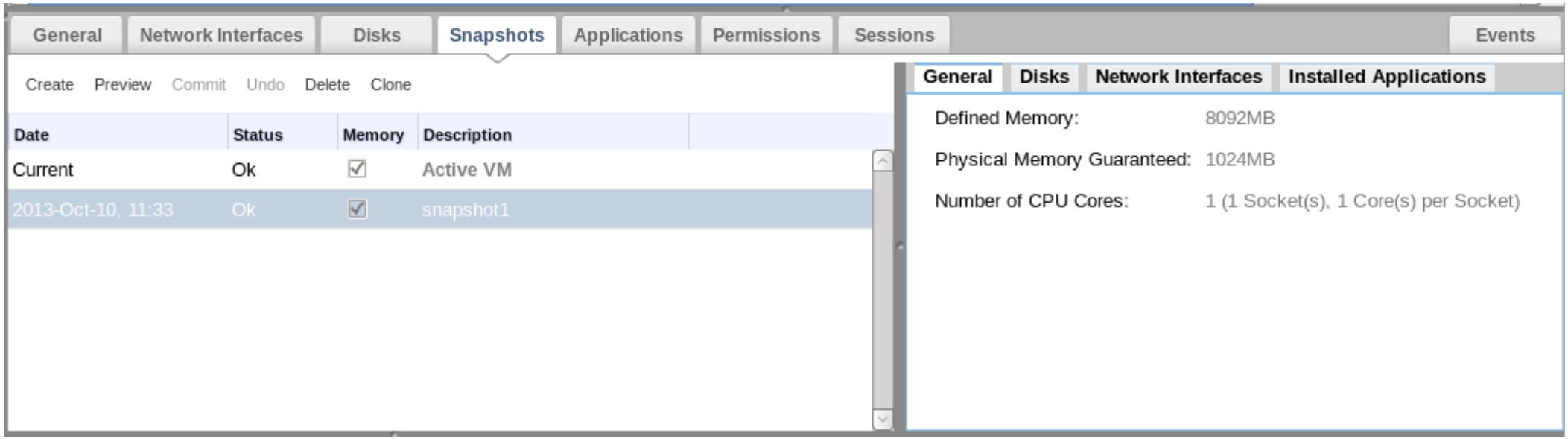


Preview snapshot

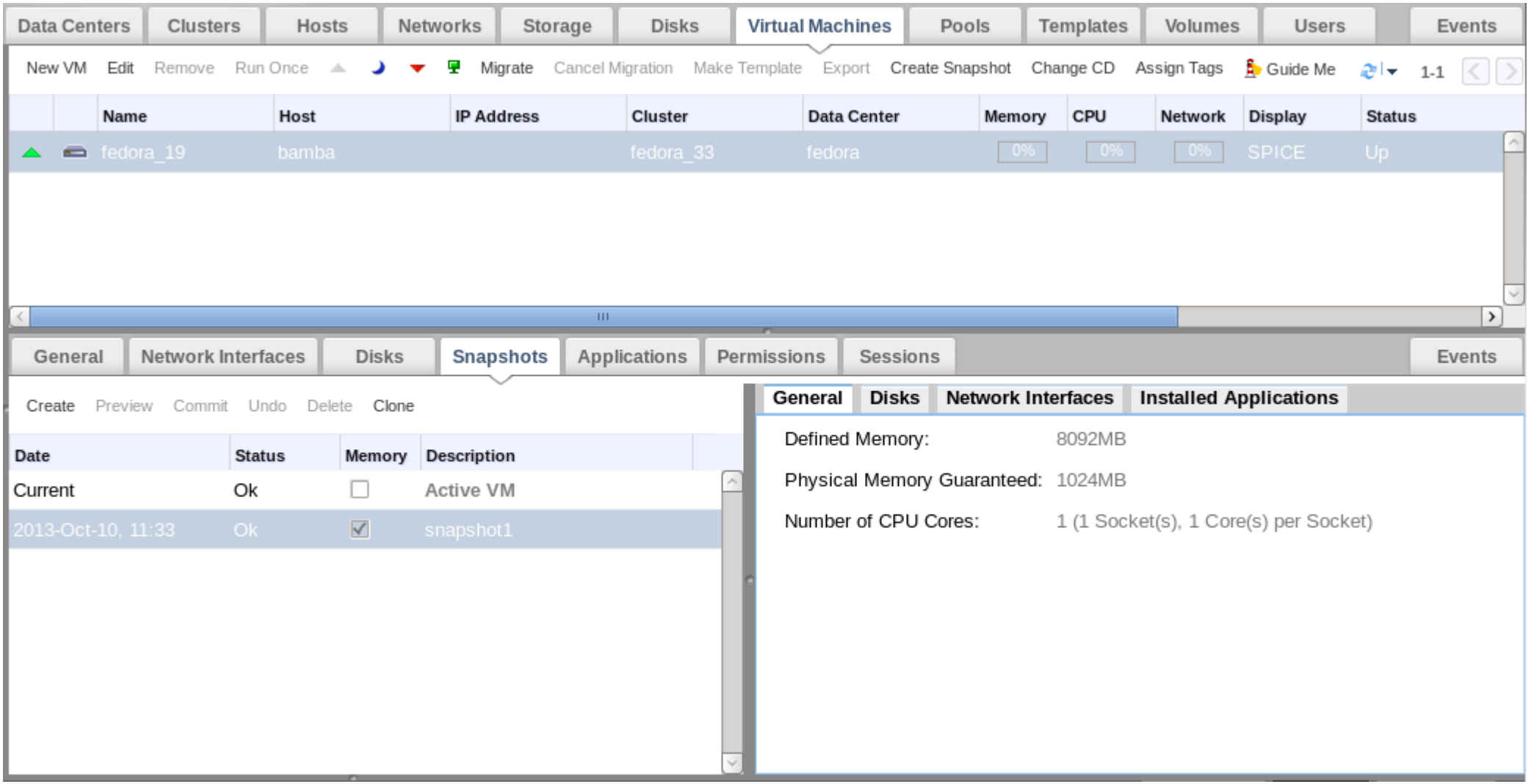




Commit to RAM snapshot



Run the committed VM



Stateful snapshot – REST API

oVirt

```
<snapshot href="/api/vms/19f5f748-9c81-46c2-8f78-3379588981e7/snapshots/534f67ce-5a97-4acb-8e97-f31c0492738d" id="534f67ce-5a97-4acb-8e97-f31c0492738d">
+<actions></actions>
  <description>my_snapshot</description>
  <type>regular</type>
+<vm id="19f5f748-9c81-46c2-8f78-3379588981e7"></vm>
  <date>2013-10-10T13:43:04.681+03:00</date>
  <snapshot_status>ok</snapshot_status>
  <persist_memorystate>true</persist_memorystate>
</snapshot>
```

Stateful snapshot operations – REST API

oVirt

Create:

```
<snapshot>
  <description>my_snapshot</description>
  <persist_memorystate>true</persist_memorystate>
</snapshot>
```

Preview/Restore:

```
<action>
  <restore_memory>true</restore_memory>
</action>
```

THANK YOU !

http://www.ovirt.org/Features/RAM_Snapshots
engine-devel@ovirt.org
vdsm-devel@lists.fedorahosted.org

#ovirt irc.oftc.net

ahadas
ahadas@redhat.com