

# VolumeSnapshot API Enhancement for VDI-per-LUN

## Bug Reference

The Jira issue associated with this design spec

## Branch

4.8

## Introduction

## Purpose

Currently managed storage backends like SolidFire store their snapshots locally on the same storage. This is contrary to Cloudstack's view of a snapshot which resides on secondary storage. With the introduction of Storage Snapshot API, we want to make sure Volume snapshots are implemented the same on managed storages (i.e. backup of snapshot to secondary storage).

## References

- *relevant links*

## Document History

## Glossary

## Feature Specifications

- **put a summary or a brief description of the feature in question**
- **list what is deliberately not supported or what the feature will not offer - to clear any prospective ambiguities**
- **list all open items or unresolved issues the developer is unable to decide about without further discussion**
- **quality risks (test guidelines)**
  - functional
  - non functional: performance, scalability, stability, overload scenarios, etc
  - corner cases and boundary conditions
  - negative usage scenarios
- **specify supportability characteristics:**
  - what new logging (or at least the important one) is introduced
  - how to debug and troubleshoot
  - what are the audit events
  - list JMX interfaces
  - graceful failure and recovery scenarios
  - possible fallback or work around route if feature does not work as expected, if those workarounds do exist ofcourse.
  - if feature depends other run-time environment related requirements, provide sanity check list for support people to run
- **explain configuration characteristics:**
  - configuration parameters or files introduced/changed
  - branding parameters or files introduced/changed
  - highlight parameters for performance tweaking
  - highlight how installation/upgrade scenarios change
- **deployment requirements (fresh install vs. upgrade) if any**
- **system requirements: memory, CPU, disk space, etc**
- **interoperability and compatibility requirements:**
  - OS
  - xenserver, hypervisors
  - storage, networks, other
- **list localization and internationalization specifications**
- **explain the impact and possible upgrade/migration solution introduced by the feature**
- **explain performance & scalability implications when feature is used from small scale to large scale**
- **explain security specifications**
  - list your evaluation of possible security attacks against the feature and the answers in your design\* \*
- **explain marketing specifications**
- **explain levels or types of users communities of this feature (e.g. admin, user, etc)**

## Use cases

*put the relevant use case/stories to explain how the feature is going to be used/work*

## Architecture and Design description

- discussion of alternatives amongst design ideas, their resources/time tradeoffs and limitations. Explain why a certain design idea is chosen over others

- highlight architectural patterns being used (queues, async/sync, state machines, etc)
- talk about main algorithms used
- explain what components are being changed and what the dependent components are
- regarding database: talk about tables being added/modified
- performance implications: what are the improvements or risks introduced to capacity, response time, resources usage and other relevant KPIs
- preferably show class diagrams, sequence diagrams and state diagrams
- if possible, publish signatures of all methods classes and interfaces implement, and the explain the object information of different classes

## Web Services APIs

*list changes to existing web services APIs and new APIs introduced with signatures and throughout documentation*

## UI flow

- *either demonstrate it visually here or link to relevant mockups*

## IP Clearance

- what dependencies will you be adding to the project?
- are you expecting to include any code developed outside the Apache CloudStack project?

## Usage Impact

- Are there any entities being created that require usage reporting for billing purposes?
- Does this change any existing entities for which usage is being tracked already?

## Appendix

**Appendix A:**

**Appendix B:**