

Tintri VMstore for Dev/Test Environment

December 2015

Contents

Executive Summary	1
Snapshots	2
Use Cases	2
Tintri VMstore Snapshot Benefits	2
Clones	3
Use cases	3
Tintri VMstore Clones Benefits	4
SyncVM	5
Use Case #1 – Refresh Disk(s)	6
Use Case #2 – Restore VM	8
Use Case #3 – File Level Restore	8
Automation	9
PowerShell Example #1	9
PowerShell Example #2	9
Conclusion	10

Executive Summary

This paper explores how Tintri solutions and products can be utilized for Development and Test environments (Dev/Test). Tintri's portfolio of products include VMstore, SyncVM™, Snapshot, Clones, Automation ToolKit, and APIs. It is the assumption of this paper that a VMware environment with Tintri storage is already in place. The target audience looking to leverage these environments are developers and engineers for development and Quality Assurance (QA).

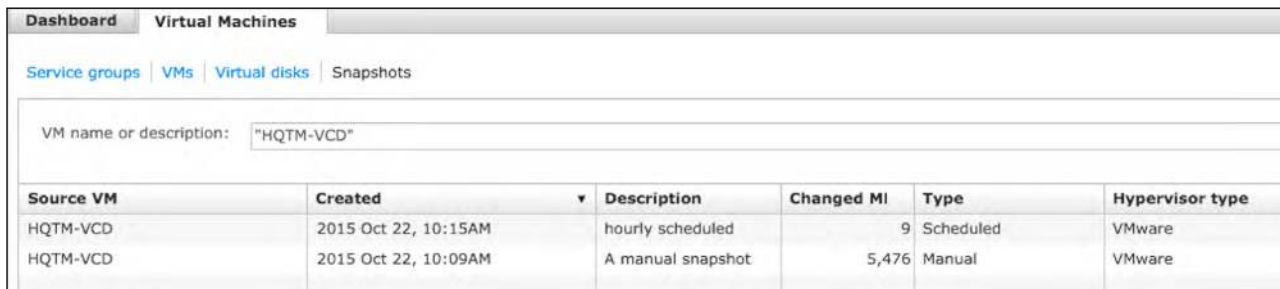
There are many challenges within an organization when it comes to development. First is the software development life cycle (SDLC). Agile and Waterfall are the typical approaches used. The Agile approach allows developers to focus on a particular section of a project, make frequent changes, and enables frequent testing during the life cycle. The waterfall approach goes in phases and completion of one phase leads to the next phase. Unlike the Agile approach, the Waterfall is less flexible and requires nailing down the scope and project requirements up front.

The classic argument between developers and IT revolves around who owns and manages the infrastructure. Developers want an infrastructure that allows them to deploy environments without limitation of scalability, performance issues, and can easily be automated. IT want to keep developers under control making sure they don't consume more resources than needed. Virtualization has helped alleviate most of these issues by allowing these groups to co-exist on the same infrastructure. Features such as resource pools help guarantee performance for production workloads from Dev/Test workloads. VLANs ensured network isolation on physical switches, which propagate down to virtual switches. Virtual machine snapshots and clones help speed up the build time of environments. Virtualization is only part of the equation, yet is has traditionally been the easiest to employ. Storage plays a big role here as well, but needs to work at the same level as the virtual machines.

Traditional storage is not built from the ground up to handle virtualization, which means it is less effective and more cumbersome to manage from those environments. These systems require more tuning and troubleshooting. Creating and maintaining multiple LUNs for separate workloads causes lots of pain points to keep track of. Tintri VMstore alleviates these pain points by supporting multiple of hypervisors and workloads on the same VMstore without contention or worrying about creating or managing LUNs. A Tintri VMstore also removes other limitations and pain points that revolve around traditional VMware vSphere snapshot and clones. VMware snapshots are limited by longevity, restricted to keeping a minimal number based on best practices, and a lack of space savings. Tintri provides administrator flexibility and gives time back from having to monitor and maintain these environments. Everything is supplied to the administrator in the VMstore UI, at a per VM level. Tintri is able to treat each VMs as first class citizen by efficiently creating snapshots and clones. There is also a vSphere plugin that can be used from the vSphere Web client to manage the Tintri VMstore providing the same information from the VMstore UI. The following information will allow you to understand how a Tintri VMstore allows you to maximize your space savings, but also have more flexibility and time savings through features like Sync VM.

Snapshots

Snapshot capabilities for virtual machines are native in vCenter server. However, these VMware native snapshots have risks and limitations that do not apply to Tintri snapshots. While VMware-based snapshots support chain lengths up to 32 snapshots deep, while, VMware's best practice ([KB1025279](#)) recommends limiting these to 2-3 per VM and encourage users **NOT** to allow snapshots to exist for long periods of time. In addition to these risks that require constant monitoring to find and eliminate VMware snapshots, they also take up unnecessary storage space. This space requires more storage since there is no native space savings built in. Tintri's VMstore snapshot capabilities remove the concern around space, monitoring, and flexibility for snapshots.



The screenshot shows the Tintri UI interface for managing VM snapshots. At the top, there are navigation tabs for 'Dashboard' and 'Virtual Machines'. Below that, there are sub-tabs for 'Service groups', 'VMs', 'Virtual disks', and 'Snapshots'. A search bar labeled 'VM name or description:' contains the text 'HQTM-VCD'. Below the search bar is a table with the following columns: 'Source VM', 'Created', 'Description', 'Changed MI', 'Type', and 'Hypervisor type'. The table contains two rows of data.

Source VM	Created	Description	Changed MI	Type	Hypervisor type
HQTM-VCD	2015 Oct 22, 10:15AM	hourly scheduled	9	Scheduled	VMware
HQTM-VCD	2015 Oct 22, 10:09AM	A manual snapshot	5,476	Manual	VMware

Figure 1: example of VM snapshot in Tintri UI.

Use Cases

Dev/Test environments need to be agile due to the frequent changes that take place. Managing multiple snapshots has risks and limitations in the case of VMware vSphere. The space needs for these snapshots can be extensive and costly. Virtual Machines (VMs) created for Dev/Test can use Tintri snapshots prior to testing. Examples can be the following:

- Installing a new application or testing an existing application functionality.
- Applying a new update or patch.
- Provisioning new Dev/Test virtual machine(s).

Tintri VMstore Snapshot Benefits

- **Per-VM Architecture.** The ability to snapshot specific VMs, rather than the entire logical unit number (LUN). Each VMstore has a limit of up to **128 per individual virtual machine (VM)**.
- **Per-VM Visibility.** VMs with VMware snapshots are also easily identified, along with additional information such as space consumption. This information allows VMware-based snapshots to be easily discovered and removed.
- **Space Savings.** Tintri snapshots are space efficient, point in time copies of VMs.
- **Performance.** There is no performance impact when taking a Tintri snapshot.
- **Flexibility.** A Tintri snapshot can be created manually or scheduled. A snapshot can also be used for cloning, to easily create a new VM resembling the point-in-time snapshot of another VM. This is beneficial when troubleshooting.
- **Configuration Changes.** Tintri snapshots have no constraints to configuration changes such as adding extending existing disks or adding new ones unlike VMware based snapshots, which do not allow changed to be made after a snapshot is taken.
- **Manageability.** VMstore UI displays all snapshot information per-VM. Tintri Global Center (TGC) has the capability to create snapshot service groups.

Clones

Like snapshots, clones also provide the capability for point-in-time copies, but are standalone VMs. Clones are copies of virtual machines in their current state, which can be used independently from the parent VM they were created from. They have their own unique MAC address and UUID from the parent VM. Also changes made to a clone do not affect the parent VM. When it comes to provisioning VMs the process can be lengthy and time consuming. Cloning is an option for rapid deployment. Saving the time and effort needed to install the operating system, updates, and configuration. Clones in VMware can be scheduled, but can only be created one at a time from vCenter server. In addition, VMware clones do not have any logic behind space efficiency, meaning the clone will consume the same amount of space as the parent VM on storage. A Tintri clone takes place at the individual VM level, making efficient use of metadata. This can only be achieved by having core data paths from the ground up to work efficiently with features such as snapshots and clones. Clones can be created instantaneously regardless of the size of the VM, and they benefit from compression and dedupe.

Use cases

Building and configuring an environment is time consuming. Developers need to have the ability to build several instances and quickly. Virtual Machines (VMs) created for Dev/Test can use Tintri clones prior to testing. Examples can be the following:

- Exact copies of production useful for troubleshooting.
- Testing updates prior to implementing changes in production.
- Rapid provisioning suitable for virtual desktops or labs.

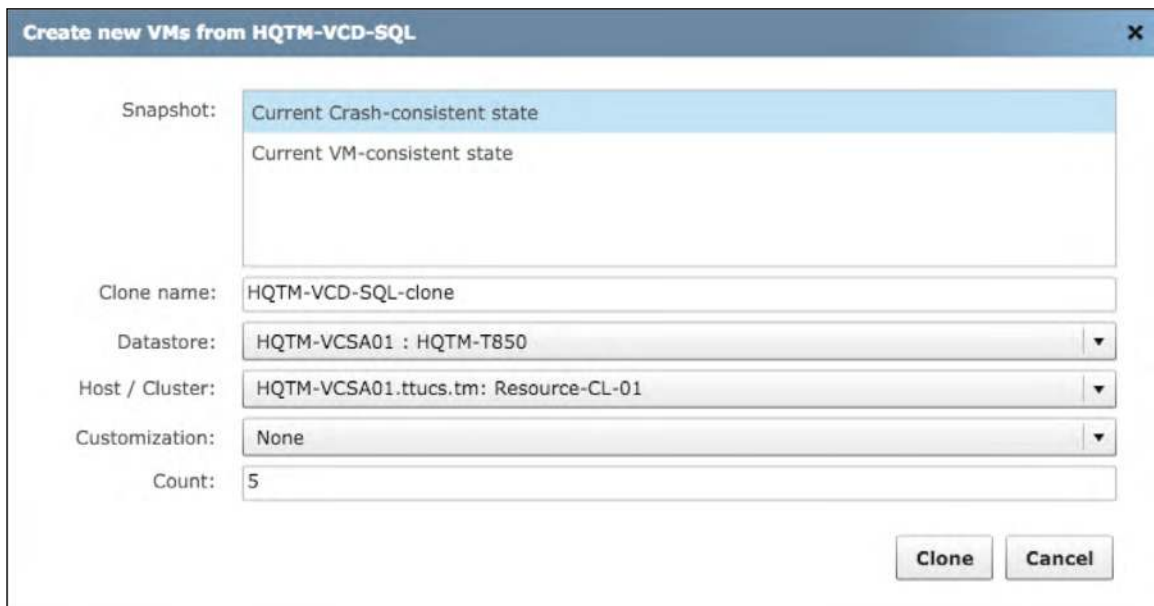


Figure 2: Tintri's VM clone creation.

- **Per-VM.** Cloning takes place at the VM level.
- **Space Savings.** In the case of VMware's clone a copy of the VM's virtual disk files is created to match the total number of times the VM is clone. This is very time-consuming and space-inefficient. Tintri clones wouldn't use any additional space; per-VM Tintri clones only consume additional space as they are modified.
- **Performance.** There is no performance impact when creating a Tintri clone.
- **Flexibility.** A Tintri clone(s) can be created manually or scheduled.
- **Manageability.** VMstore UI displays all clone information per-VM.

VM	IOPS	Cloned from	Clone dedup factor
W7	0	0.0 Win7 Apr 23 4:48PM	3.2x
Win7-CL2-1	0	0.0 Win7 Apr 23 4:48PM	1.9x
Win7CL-2	313	5.1 Win7 Apr 23 4:48PM	2.0x
Win7CL-1	310	5.1 Win7 Apr 23 4:48PM	2.1x
UCSVM1	0	0.0 UCSVM1 Sep 29 11:41AM	1.8x
UCSVM3	0	0.0 UCSVM1 Aug 13 5:18PM	4.2x
DC-SQL1	1	0.01 SQL2 Aug 14 1:03PM	1.0x
SQL12Clus2	0	0.0 SQL12Clus Jun 19 2:39PM	2.3x

Figure 3: Tintri VMstore UI columns display Clone dedup and parent VM.

SyncVM

Mirroring a production environment in a Dev/Test environment is usually a challenge. To go even further it's the data management component that makes it a challenge. Figure 4 below illustrates two types of environments. The one on the left utilizes scripts to maintain the data between each environment, requiring a lot of maintenance and work. One the right uses SyncVM, which is only a couple of mouse clicks use and no maintenance work required like scripts.

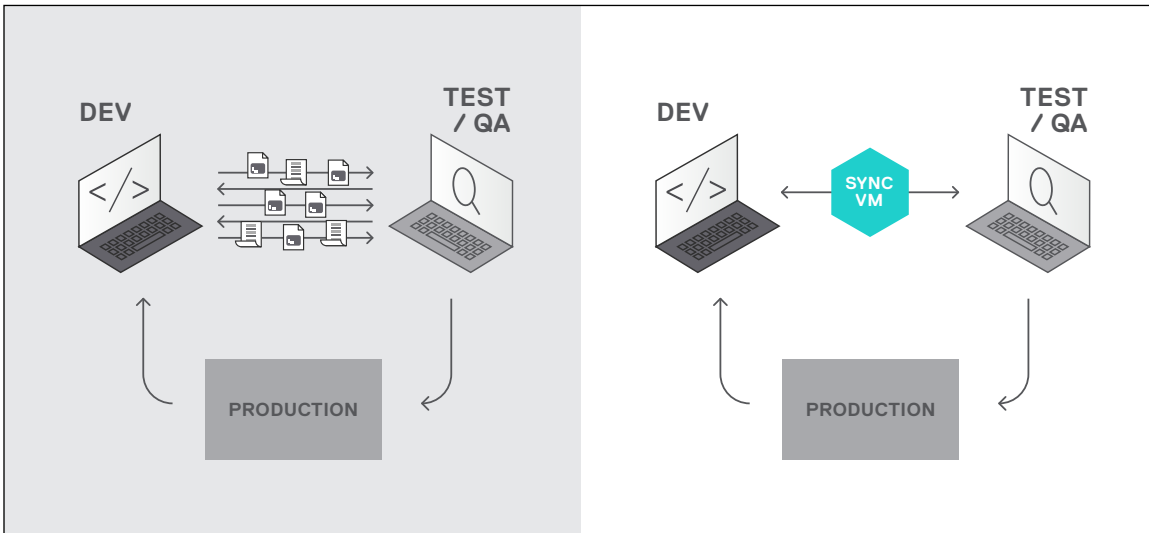


Figure 4: (Left) Utilizes multiple scripts to maintain data between the environment. (Right) SyncVM only requires a couple of clicks.

Maintaining an environment with scripts requires a lot of time, up keep, and intervention from an administration standpoint. As the environment changes so will the scripts and keeping up with the maintenance is the challenge. Scripts also may not be able to move between other environments, not to mention the clutter that comes from having to keep several scripts available or up to date. SyncVM eliminates the need for all these dependencies and guess work. It provides a dynamic and scalable way of refreshing your data between environments. This can also be useful for other environments such as labs, kiosks, demos, and file servers.

SyncVM solves these issues by focusing on the data management. Physical data is not moved, instead it leverages snapshots and cloning. One of the value adds is the ability to move between different points in time, referred to as time travel. This is not possible with traditional snapshots. A crash-consistent snapshot is taken prior to the data synchronization and is selectable from the snapshot dropdown list. This snapshot is referred to as a “**SyncVM safety snapshot**”, a built-in default preventive measure. This safety snapshot can be very useful when used with **File Level Restores (FLR)**, covered later in this document.

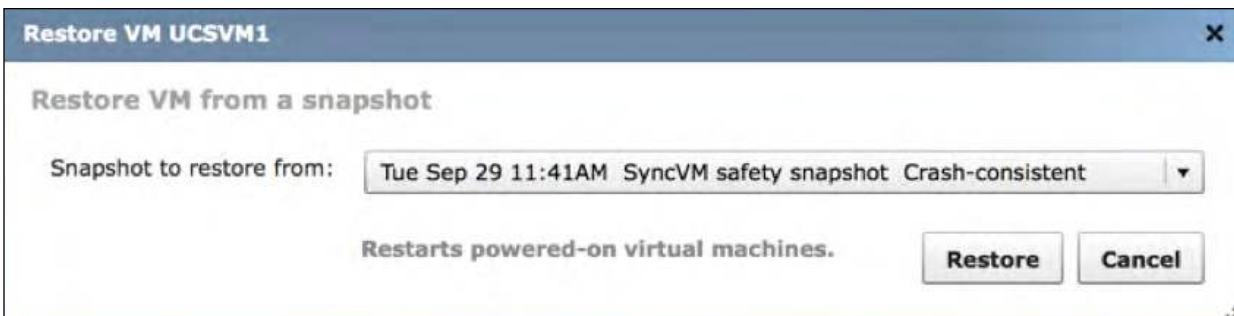


Figure 5: SyncVM safety snapshot taken initially can be used for restore.

As far as VMware's vCenter is concerned there is no impact on performance, existing snapshots, storage, and / or virtual machine reconfiguration. This is all handled at the Tintri VMstore level.

Do: Use for Databases, Exchange, and SharePoint.

DON'T: Have VMware-based snapshots and/or Linked Clones.

Flexibility is key with SyncVM. The capabilities can be invoked at either the VM or vDisk level. Another thing to note is SyncVM is application agnostic, since the focus is primarily on the data. SyncVM can be useful not only in DEV/Test environments, but others such as labs, server refreshes, and demos to name a few. Let's explore a few use cases and how SyncVM can be beneficial.

Use Case #1 – Refresh Disk(s)

Databases are usually complex depending on the application or configuration. Having to mirror this setup or requesting test data can also be time consuming. For example, a SQL database test environment can be refreshed from the source production data to a test bed (manual database seed). This can be done at the VM level, creating an exact copy of the production database in test. Now any changes can be made and tested without effecting production and was done with a single click of button. The other option is refreshing only part of the database from the vDisk level.

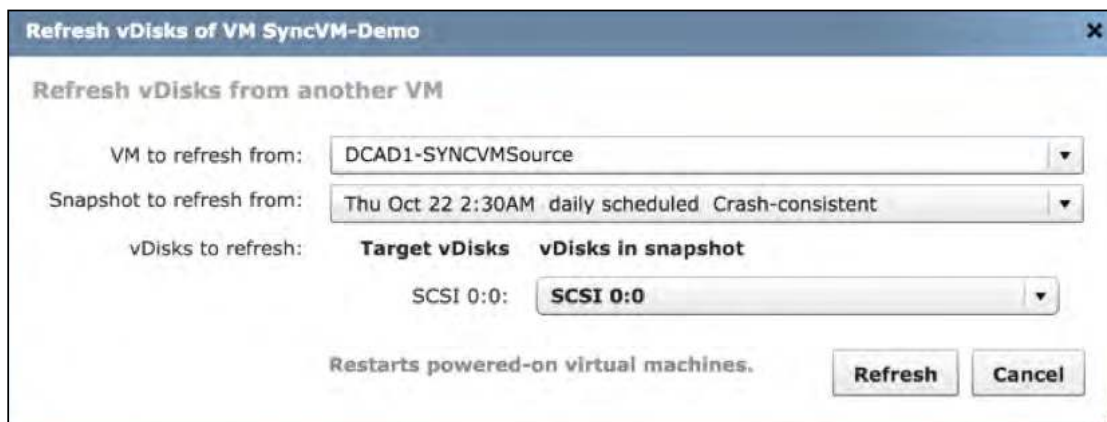


Figure 6: SyncVM refresh at the vDisk level.

Most SQL database(s) have individual disks for data (could have multiple disks), log(s), TempDB, and Backup. SyncVM can be used in this scenario to refresh the individual database files. Along the same line as the database refresh restores are also possible.

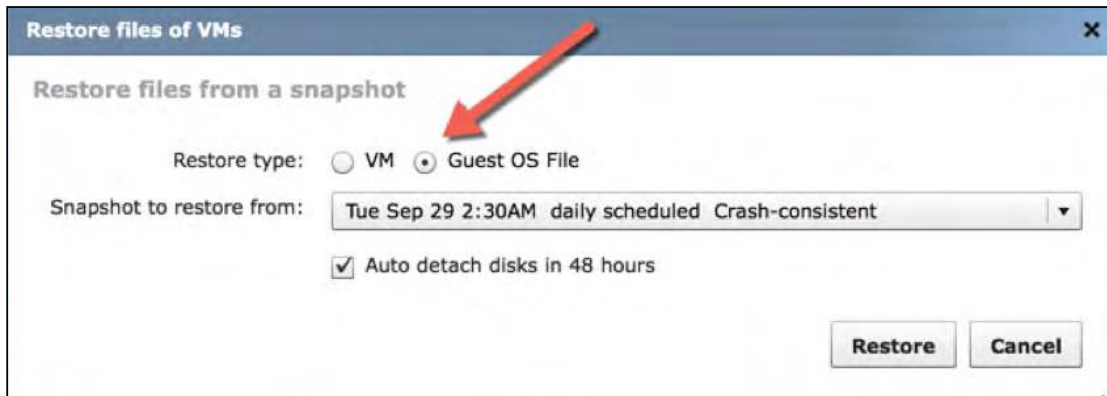


Figure 7: SyncVM provides the option to restore the entire VM or individual files.

DO: See [Tintri's SQL Best Practice guide](#).

Isolation is also important for keeping a separation between production and Dev/Test environments. Using VLANs or keeping a VM offline to provide this separation can ensure the Dev/Test environment will not affect production. SyncVM's refresh feature can be used to alongside the isolated of environments. Creating a Dev/Test environment can take a while in creating the VMs. This can be done by either manually building out a separate environment or cloning a current one. Keeping that test bed of data up to date can be a challenge with traditional storage. Scripts are used to copy over data from production to test, some also rely on backups. While these options will get the job done, they require lots of time and intervention to keep that data up to date. SyncVM removes these constraints of time and up keep. Now Dev/Test environments can be build using Tintri's cloning and SyncVM to refresh the data.

Multi-tier applications include Web Servers, Application Servers, and Database servers. These environments can be complex, especially when spanning multiple security zones. Refreshing the data on these servers, rather than using "golden images", is a simple approach. Testing updates, patches, bug fixes, and troubleshooting are ways these applications or Dev/Test environment can benefit from SyncVM.

Troubleshooting is always hard to do especially in production environments. Having tools like SyncVM we can now build isolated environments that can be used to further investigate. Troubleshooting can not only can benefit from the refresh feature, but from the file level restore feature as well. The ability to pull logs from another system to compare instead of restoring an entire VM.

Use Case #2 – Restore VM

Environments such as labs, demos, QA, and Kiosks normally need to be consistently cleaned up. SyncVM can be used to cleanup and refresh these environments to their original state. This option also allows flexibility to bring a VM back from catastrophes such as viruses and upgrades or patches that have affected the operating system. From the GUI this can be done on a per virtual machine basis, and it utilized the time travel feature. This can be done manually or one can automate the process. Using the Tintri PowerShell toolkit, commands or scripts can be created and scheduled to restore these VMs to their original state. From one source VM you can restore 20 target VMs at a time.

Use Case #3 – File Level Restore

File level restores (FLR) can be beneficial in a multitude of use cases.

- One example is over writing files with newer data and need to recover the older version. This makes it easier to recover and compare files without recovering the entire virtual machine. Beneficial for Dev/Test environment where data is updated and issues occur. This enables easier troubleshooting by being able to look at older version of files.
- Another example for FLR is retention. Scheduling snapshots on a daily, weekly, or monthly can provide the availability to go forward or backward in time leveraging snapshots to get files that are needed for legal reasons. There is also the potential for data mining from a database throughout a period of time.

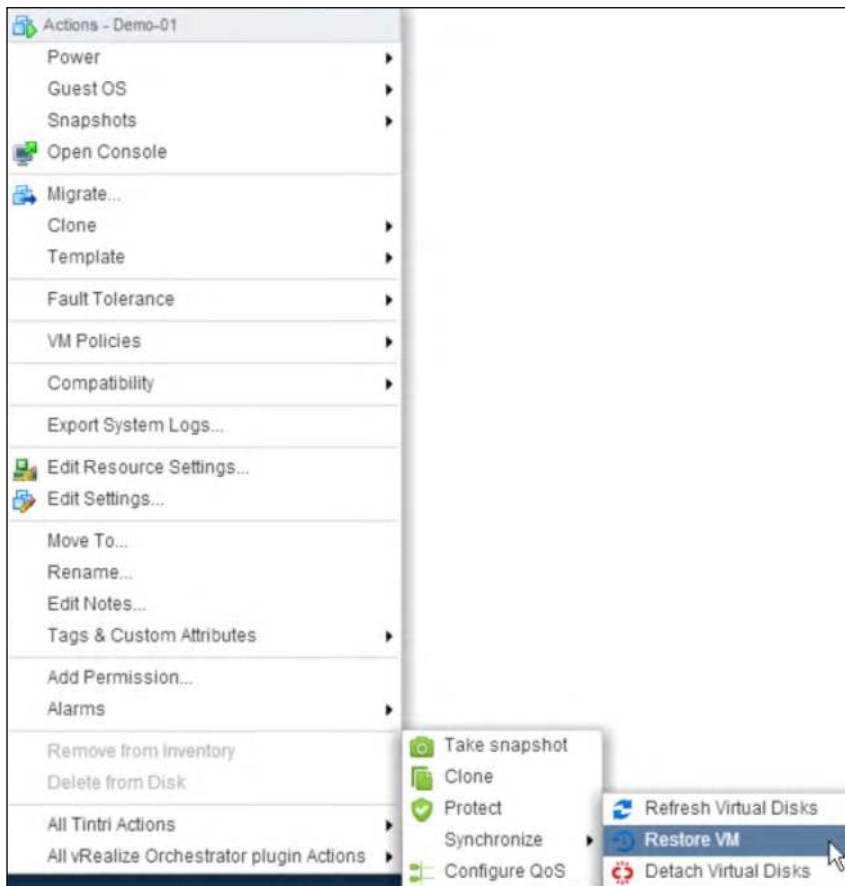


Figure 8: Tintri VMware Web Client Plugin.

Automation

While most operations can be done within the Tintri VMstore UI, Tintri offers options to also automate these operations. If you are new to PowerShell or an expert, the [Tintri Automation ToolKit](#) continues to allow you to use skills or build on them as an added module. The PowerShell toolkit also works with VMware's PowerCLI providing the uniqueness at the per VM level. Using the PowerShell toolkit allows the creation of scripts to automate many functions on the VMstore and / or Tintri's Global Center (TGC). Tintri also offers access to open REST APIs. Like the PowerShell Toolkit, REST APIs will assist in automating various tasks all at the VM level. An example is using the REST APIs to create a custom workflow in VMware vRealize Orchestrator. A few examples are:

- VM Snapshots (delete & restore)
- VM Cloning
- Collecting VM information (disk, statistics, VM)
- Tintri Replication
- SyncVM
- QOS

PowerShell Example #1

In this example we are looking for all VMs that start with "LabVM" and assigning the results to the variable `$vms`.

```
$vms = Get-TintriVM -Name LabVM*
```

Next step is to use the SyncVM command Sync-TintriVM to restore from the latest snapshot for all the VMs that start with "LabVM".

```
$vms | Sync-TintriVM -UseLatestSnapshot
```

PowerShell Example #2

In this example we are looking for a VM named "LabVM" and assigning the results to the variable `$vm`.

```
$vm = Get-TintriVM -Name LabVM
```

Next getting the snapshots associated with the vm named "LabVM" and assigned to the variable `$snaps`.

```
$snaps = Get-TintriVMSanpshot -VM $vm
```

Finally restoring using a specific snapshot.

```
Sync-TintriVM -VM $vm -SourceSnashot $snaps[0]
```

Another example is using the SyncVM command `Sync-TintriVDisk` to restore individual `vDisks`. An example script is available on the [Github](#). This Script is used to refresh the individual vDisk on a daily schedule.

These workflows can also be used as part of a self service portal, such as with VMware's vRealize Automation. More information for Tintri's REST APIs can be found [Here](#). For a list of examples using Tintri's REST APIs can also be found on Tintri's [GitHub page](#).

Conclusion

Tintri VMstore is the perfect storage and enabler for Dev/Test environments. It removes the overhead and complexity that other storage brings to virtualized environments. This allows Tintri, built from the ground up for virtualization to meet the demands of businesses and customers. The ability to reclaim space and provide more density for Dev/Test environment is an advantage. Tintri VMstore also provides ease of management and the ability to troubleshoot & debug issues accurately, therefore saving time. Providing open REST APIs and Automation ToolKit support to build scripts at the VM-level to expands capabilities of the administrator and provides more granularity not available in the user interface (UI). More information of Tintri offerings can be found at Tintri.com



Global HQ

303 Ravendale Dr.
Mountain View, CA 94043
United States
+1 650-810-8200
info@tintri.com

EMEA HQ

Fountain House 10th Fl
130 Fenchurch Street
London EC3M 5DJ
+44 (0) 203 053 0853
emea@tintri.com

APAC HQ

9 Temasek Boulevard
Suntec Tower 2, #09-01
Singapore 038989
+65 6407 1359
apac@tintri.com

Japan HQ

Level 6, Kishimoto Building
2-2-1 Marunouchi, Chiyoda-ku,
Tokyo 100-0005 Japan
+81 (3) 6213-5400
info.japan@tintri.com