

Cloud Automation with Tintri

Using the Tintri vRealize Orchestrator Plugin

Revision History

Version	Date	Description	Author
1.0	09/20/2016	Initial Release	Adam Cavaliere Nick Colyer Tomer Hagay

Table 1 - Revision history

Contents

Executive Summary	1
Introduction.....	1
Tools for Private and Hybrid Cloud Deployment.....	1
Cloud Management Platform/Service Catalog.....	2
Orchestration.....	2
Configuration Management.....	3
Choosing Technologies to Enable Cloud Automation.....	3
Tintri Storage Arrays and Cloud Automation.....	3
Tintri REST APIs.....	3
VMware vRealize Automation and vRealize Orchestrator.....	3
The Tintri vRealize Orchestrator Plugin.....	4
Tintri Snapshot Use Case.....	4
Executing the Snapshot Workflow.....	4
Snapshot Workflow Examples.....	5
Self-Service Example.....	5
Customization Example.....	6
Tintri Protect VM Use Case.....	7
Executing the Protect VM Workflow.....	7
Tintri Replication Use Case.....	8
Executing the Replicate VM Workflow.....	9
Replicate VM Workflow Examples.....	10
Customization Example.....	10
Self-Service Example 1: Running the Workflow During Blueprint Provisioning.....	11
Self-Service Example 2: Enabling Replication as a Day 2 Operation.....	13
Tintri SyncVM Use Case.....	14
Executing the SyncVM Refresh Virtual Disks Workflow.....	15
SyncVM Workflow Examples.....	16
Customization Example: Guided-Refresh Disks.....	16
Self-Service Example: Running the SyncVM Workflow During Blueprint Provisioning.....	18
Self-Service Example 2: Running SyncVM as a Day 2 Operation.....	19
Tintri QoS Use Case.....	20
Tintri File Restore Use Case.....	22
Executing the Restore Files Workflow.....	22
Conclusion.....	22

Executive Summary

Many enterprise IT teams are deploying private clouds to allow on-premises infrastructure to offer the streamlined consumption model, improved agility, and economics of the public cloud. Enterprises need to simplify and automate services available from existing IT infrastructure to achieve this goal.

IT infrastructure can be simplified through virtual machine (VM) awareness. Tintri provides VM-aware storage that frees IT from having to worry about and orchestrate the complexities of LUNs, zoning, masking, and other storage specifics. Because all Tintri capabilities are provided at VM granularity, Tintri VMstore storage arrays, add significant value in terms of allowing users to easily protect and replicate individual VMs.

VMware vRealize Automation has become a popular cloud platform supporting self-service for private and hybrid cloud deployments. VMware vRealize Orchestrator simplifies the process of creating fully custom workflows. Tintri vRealize Orchestrator plugin facilitates the integration and use of Tintri storage in vRealize environments. It provides a variety of pre-defined workflows for common Tintri tasks. This white paper explains how you can use this plugin to accomplish important storage tasks including snapshot for VM protection, replication for disaster recovery, VM sync to update datasets in development and test environments, and QoS to manage performance service levels across large numbers of VMs.

Introduction

The cloud has had a profound effect on the IT landscape. Enterprises are creating private cloud infrastructures to deliver a similar user experience as the public cloud, delivering the benefits of greater business agility and lower IT costs. Many enterprises are combining both private cloud and public cloud resources in a hybrid cloud model that allows them to take advantage of the predictable performance and costs of on-premises infrastructure while being able to utilize the public cloud for special projects, bursts of activity that exceed on-premises capacity, and other special needs.

Creating a private cloud requires some fundamental changes to IT:

- Simplify the underlying infrastructure
- Develop a simple methodology to provide IT as a Service
- Improve quality via repeatable, standardized deployments that reduce human error
- Improve speed of service delivery via automation
- Adapt and change with evolving business needs

This paper discusses how you can take advantage of Tintri VM-aware storage capabilities in a cloud environment. Specifically, using Tintri storage in combination with vRealize Automation and vRealize Orchestrator results in a highly efficient private cloud solution. By utilizing Tintri clones, snapshots, and replication, and integrating those capabilities into a self-service portal, the promise of the private cloud—lower costs and faster time to market—can be achieved. Tintri provides a plugin for vRealize Orchestrator that automates common Tintri workflows for use within vRealize Orchestrator. The second half of this paper focuses on common Tintri use cases.

Tools for Private and Hybrid Cloud Deployment

There are three fundamental tools to help you achieve your private and hybrid cloud goals:

- Cloud Management Platform or Service Catalog
- Orchestration
- Configuration Management

Cloud Management Platform/Service Catalog

The Cloud Management Platform or Service Catalog is the entry point through which developers and Enterprise IT users get the services they need. Showback or Chargeback are often implemented here, as well as all approval policies necessary to ensure governance. A decision matrix guides users through the process of requesting the correct service without them having to know details of the underlying infrastructure such as specifics of networking, storage, and servers. The Cloud Management Platform also adds a layer of multi-tenancy, which allows shared infrastructure resources to be consumed by multiple users and organizations. Multi-tenancy at the Cloud Management layer also enables additional advanced features such as quota management, which could be managed at the individual tenant level.

Orchestration

The orchestration layer is the glue that connects many of the disparate systems together. The orchestration layer ensures that all the tasks necessary to deliver a requested service, such as provisioning a new VM, are completed. It helps to think of the process of orchestration as building a task library.

Task Library of Services	
Server Name	Join AD
Request IP	Update Backup
Update DNS	Update Alerting
Clone Template	Create Snapshot

Figure 1—The orchestration process results in a library of tasks that are used to create more complicated workflows.

Each task is a piece of automation that enforces the standards and options required for your data center. These tasks can then be joined into an orchestration workflow.

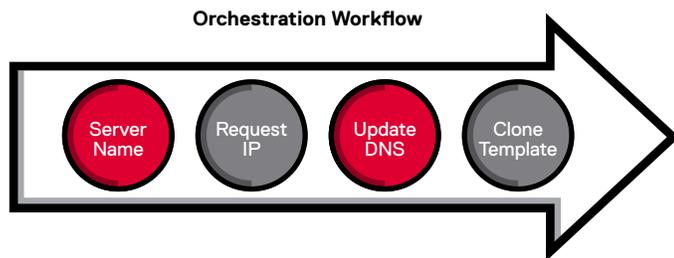


Figure 2—An orchestration workflow combines multiple tasks from your task library into the sequence needed to accomplish a more complex task such as provisioning a new VM.

Figure 3 shows a useful process flow for creating new services as part of your service catalog:

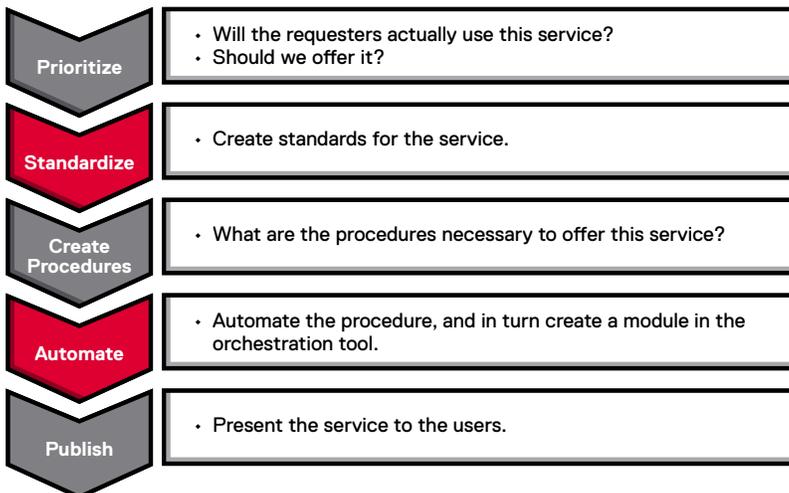


Figure 3—Process flow for service creation

Configuration Management

Configuration management software tracks all of the configuration items required for an IT system such as a server or a VM. In configuration management, you identify the modules that will be installed in each OS container, or external modules that will define configurations, as well as ensure they are enforced. Enforcement is a critical step as it changes the way you think about troubleshooting and making configuration changes. Configurations are held in a central repository and pushed down to nodes (servers, devices, etc.). If someone changes a configuration manually, when the configuration management agent runs, it will set it back to the desired state.

Choosing Technologies to Enable Cloud Automation

As mentioned in the introduction, simplifying the infrastructure can make the tasks of orchestration and automation far easier. Choosing infrastructure options that let you focus on the VM level is often a good way to facilitate automation because the VM is likely to be central to many of the configuration requirements.

Tintri Storage Arrays and Cloud Automation

Tintri VMstore storage arrays are purpose-built for virtualization and the cloud. IT administrators with working knowledge of virtualization can easily deploy Tintri storage without specialized storage knowledge. When deploying Tintri storage, there are no prerequisite operations such as LUN provisioning, HBA compatibility checks, or FC LUN zoning operations. From a VMware administrator's point of view, the entire Tintri VMstore is presented as a single datastore.

Tintri VMstore storage delivers extreme performance, VM density, and a wide variety of powerful data management features, seamlessly integrated with vSphere. Examples of data management functionality include snapshots, clones, instant bottleneck visualization, and automatic virtual disk alignment. Tintri VMstore extends and simplifies the management of virtual machines (VMs) through intrinsic VM-awareness that reaches from the top of the computing stack all the way down into the storage system.

Because Tintri VMstore systems let you focus at the VM level, automating tasks such as replication policies is much simpler. When automating storage policies for a VM, they are executed natively on the storage. The operational overhead of these tasks is minimal as is the effort required to automate them.

To summarize Tintri benefits:

- Tintri makes storage simple, so you can spend more time automating and less time worrying about storage details such as LUNs.
- Tintri gives you a VM-focused view, as well as a forward-thinking vision to integrate with public cloud services where possible.
- Tintri facilitates automation via plugins and REST APIs

Tintri REST APIs

All Tintri functions are presented via an external RESTful API. You don't need to focus on disk, you can think purely about what you want to achieve:

- I want to offer Tintri Snapshots On Demand
- I want to offer VM replication On Demand
- I want to sync Data between two machines for my DevOps team.

With the Tintri REST API, any automation tool can invoke Tintri-specific functions.

VMware vRealize Automation and vRealize Orchestrator

VMware vRealize Automation is a cloud automation software platform providing a self-service portal with a unified service catalog, and providing multi-vendor virtual, physical, and public cloud support. With vRealize Automation you can offer infrastructure as a service (IaaS) or higher-level services as your requirements dictate.

VMware vRealize Orchestrator is one of the most widely used orchestration tools in the industry. It is designed for use with both VMware vRealize Automation and vCloud. It allows you to easily automate complex workflows and includes an extensive library of prebuilt tasks for common administrative actions. It provides an SDK that supports the creation of specific plugins.

The Tintri vRealize Orchestrator Plugin

Tintri has created a plugin for vRealize Orchestrator that automates many common Tintri tasks. Think of these tasks as building blocks that you can then assemble into your desired workflows. Combine this with vRealize Automation, and you can offer self-service for provisioning of services, or you can offer a combination of Day 2 operational services. The Tintri plugin is beneficial for both Enterprises and Service Providers, enabling your customers to access services that previously would have required a ticket and days or weeks of waiting.

Example use cases are explained in more detail in the remaining sections of this document. The following use cases are covered:

- Snapshot
- Protect VM
- Replication
- Sync VM
- Quality of Service (QoS)

Tintri Snapshot Use Case

The Tintri snapshot capability is extremely useful for protecting VMs. An app developer can take a snapshot of a VM before an application push, or a Windows server engineer could snapshot hundreds of servers before they get patched. Tintri snapshots operate on the VM itself, make very efficient use of data, and do not impose any performance overhead. Using the vRealize Orchestrator Snapshot Workflow, you can offer this capability as part of your service catalog or as part of a larger workflow.

Executing the Snapshot Workflow

The vRealize Orchestrator Snapshot Workflow is ready to use out of the box and requires a minimal amount of input to execute.

In the first step, the user selects the virtual machine from vCenter.

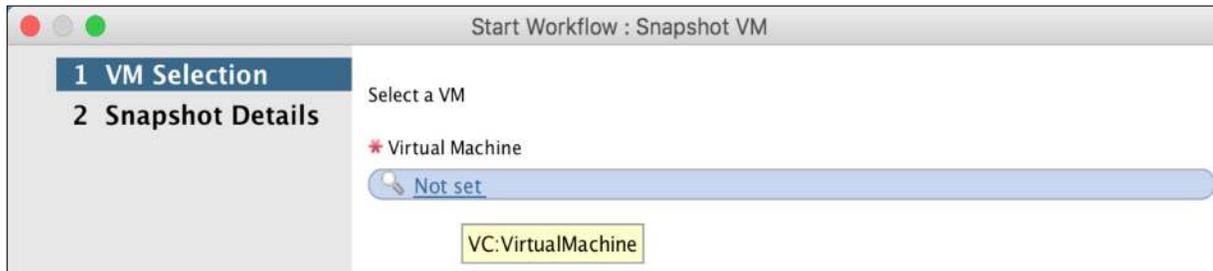


Figure 4— How to use the vRealize Orchestrator workflow

In the second step, the user enters some additional information about the snapshot:

- Snapshot name: name of snapshot as it will appear on the Tintri VMstore
- Snapshot type:
 - Crash-consistent
 - VM-consistent
- Retention Minutes: number of minutes to keep the snapshot

Note: Tintri snapshots integrate with vSphere to allow the creation of VM-consistent snapshots. Before the snapshot is created on the storage array, the VM is first quiesced and stabilized. With crash-consistent snapshots, the VM is not quiesced. You can find application-specific information on choosing VM-consistent versus crash-consistent snapshots at tintri.com/resources. The Tintri best practice guides you'll find there often give guidance on this setting for specific applications.

Figure 5—Snapshot details

You should verify that the Snapshot Workflow executes successfully against a test VM. Then you can allow users to access and use it “as is” or create a workflow that incorporates the Snapshot Workflow. Both options are illustrated in the examples that follow.

Snapshot Workflow Examples

Self-Service Example

Figure 4 shows the Actions menu in vRealize Automation:

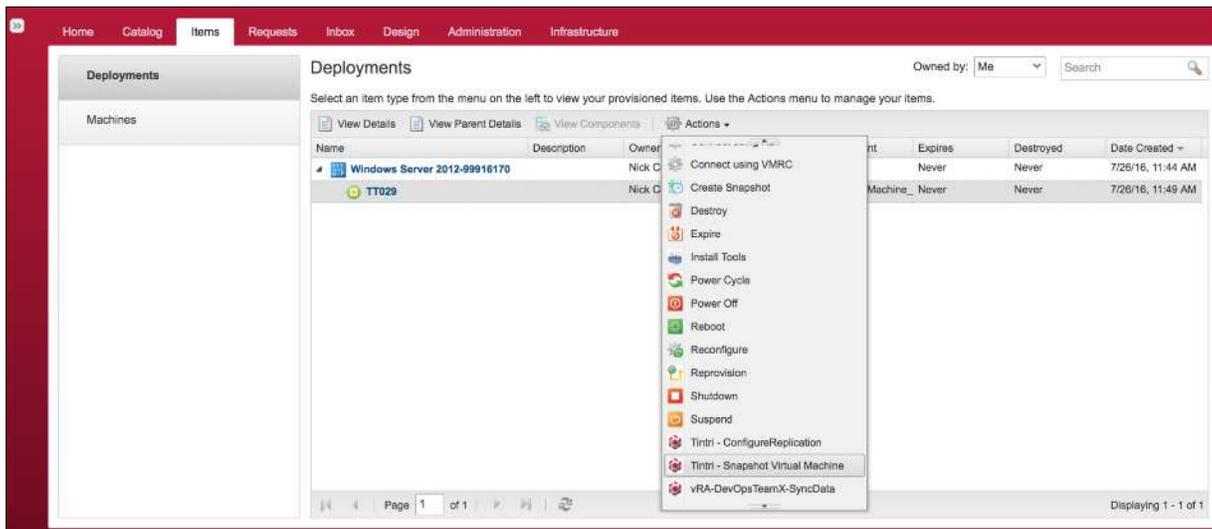


Figure 6—Self-Service example

By entitling the Tintri-Snapshot Virtual Machine action, you give users self-service access to Tintri snapshots on demand. Users can take a snapshot without having to make a request to IT and waiting for someone to get back to them.

Customization Example

The next example assumes you have decided not to prompt the user for the snapshot type and only use the “CRASH_CONSISTENT” option, since it is more efficient, significantly faster, and also addresses most use cases.

To do this, you create a new workflow called “D2-Tintri Snapshot” that includes the “Snapshot VM” workflow.

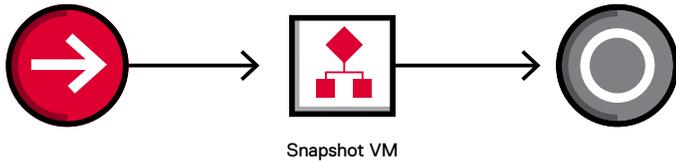


Figure 7—D2-Tintri Snapshot workflow

First you preset the general attribute for “SnapshotType” to “CRASH_CONSISTENT”

Name	Type	Value	Description
SnapshotType	string	CRASH_CONSISTENT	Type of snapshot to be created (PreSet)

Figure 8—Configuring the SnapshotType attribute

By providing the user inputs for the virtual machine and snapshot name, your workflow presentation now appears as follows:

Start Workflow : D2-Tintri Snapshot VM

Common parameters

Virtual Machine
Not set

Name of snapshot to be created

Finally, from the Design section of vRealize Automation, simply create a new advanced service to enable user access to the newly created workflow:

Home Catalog Items Requests Inbox Design Administration Infrastructure

Design

Custom Resources
Resource Mappings
XaaS Blueprints
Resource Actions

New Resource Action

Workflow Input Resource Details Form

Select a Workflow

- PowerShell
- SNMP
- SNMP Samples
- SOAP
- SQL
- SSH
- Tagging
- Tintri
 - Configuration
 - Day 2 Operations Examples
 - D2-Tintri Snapshot VM**
 - Virtual Machine
 - Tintri-REST Workflows
 - TinTriTech
 - Troubleshooting
 - VAPI

Selected Workflow

Name: D2-Tintri Snapshot VM

Description:

Input parameters:

Name	Type
VM	VC:VirtualMachine
SnapshotName	string

Displaying 1 - 2 of 2

Output parameters:

Name	Type
No parameters	

< Back Next > Finish Cancel

Figure 9—vRealize advanced service

Tintri Protect VM Use Case

The ability to set protection at a per-VM level is built into the Tintri VMstore and utilizes the same snapshotting mechanisms as described in the previous section. By creating a per-VM protection schedule, applications can be protected based on their needs without having to conform to the capabilities and limited schedules of traditional storage. Because Tintri is able to have up to 128 snapshots per VM without a performance penalty, the applications can be protected with very low RPOs and also utilize a deeper tree for longer-term data protection.

The schedules can be set to snapshot as frequently as every 15 minutes with retention periods ranging from hourly to quarterly. The system utilizes this retention schedule to create a rolling protection window that offers a “first in, first out” (FIFO) mechanism for snapshot retention. This results in zero management of snapshots.

Executing the Protect VM Workflow

To start, a user selects the VM they want to set a protection schedule on. They can then decide the frequency of protection they'd like to set. When a user selects “yes” for the different schedule options, they will be presented with a series of sub options, as shown in Figure 10 below. In this example, every schedule option has been selected:

Start Workflow : Protect VM

- ✓ 1 VM Details
- ✓ 2 Protect with Snapshot Sc...
 - Use system default Yes No
 - Hourly Schedule Yes No
 - Daily schedule Yes No
 - Weekly Schedule Yes No
 - Monthly schedule Yes No
 - Quarterly Schedule Yes No
- ✓ 2a Snapshot Type
- ✓ 2b Hourly
- ✓ 2c Daily
- ✓ 2d Weekly
- ✓ 2e Monthly
- ✓ 2f Quarterly

Figure 10—Protect VM workflow

Each schedule option allows for specific configuration settings to be set. Figure 11 below is a full example of the options that can be set for an hourly snapshot.

* Starts with per minutes
5.0

* Every Minutes
15 minutes

* Weekly Days
Array [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, ...]

* keep local hours
6.0

* Keep remote hours
4

* Snapshot Type
Crash_consistent

Figure 11—Protect VM workflow, complete options

The following table explains the options for configuring hourly snapshots.

Configuration Setting		Details
Hourly Snapshots	Starts with per minutes	The time snapshots will execute. Entering “5” would result in a snapshot taken five minutes past the hour.
	Every Minutes	The frequency of snapshots.
	Weekly Days	The days of the week the snapshot schedule will run.
	Keep Local Hours	Local snapshot expiration.
	Keep Remote Hours	Replicated snapshot expiration.
	Snapshot Type	Either Crash_Consistent or VM_Consistent

Table 2—Protect VM “Hourly Snapshots” configuration options

The other scheduled options provide many of the same settings. The table below outlines the options that are unique.

Configuration Setting		Details
Daily / Weekly Snapshots	Daily at	Sets a specific time for the snapshot to occur. This is entered by specifying the time in these formats: 1:30am or 5:30pm. Multiple times can be set throughout the day by separating with a comma.
Monthly / Quarterly Snapshots	Days	This allows for the date to be set for the snapshot. This allows for a snapshot to happen on the first of the month or any other significant organizational date. There are multiple ways to trigger this, for example: 1st, 15th, first day, and last day.

Table 3—Additional configuration options for Daily/Weekly and Monthly/Quarterly snapshot schedules

Tintri Replication Use Case

Many IT departments go through a lot of pain creating a disaster recovery (DR) plan. Application Teams request virtual machines, install their applications, and as they go live, they need to have replication enabled to ensure the virtual machines are available at the DR site. Historically this meant making sure the virtual machine was located on a specific LUN, as well as routinely checking to make sure Storage vMotion wasn’t used to move it onto a “non-DR” LUN. Tintri solves some of these problems by being VM aware and allowing replication to be enabled specifically on a virtual machine. Tintri automation capabilities enable even richer options.

Executing the Replicate VM Workflow

The first step in replicating a VM is to make sure it has a snapshot schedule set. Enable replication directly via the Tintri vRealize Orchestrator plugin, which can be customized to present the replication options you want. After installing the plugin, the “Replicate VM” workflow is made available.

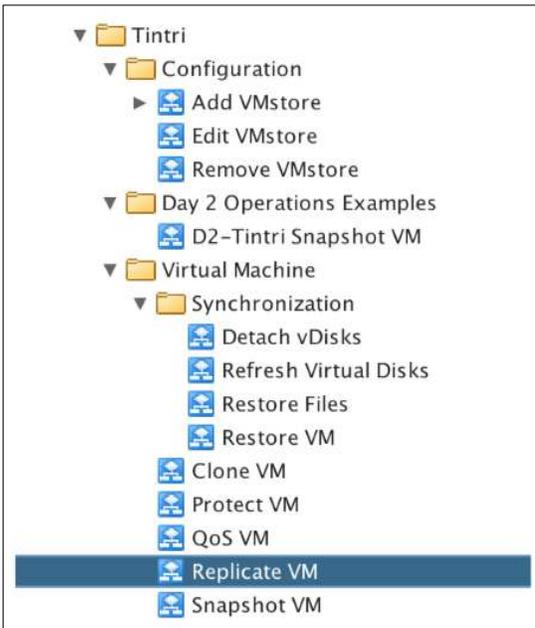


Figure 12—Replicate VM workflow

Running this workflow presents the user with a number of options for replication:

A screenshot of the 'Replicate VM' workflow options dialog box. It contains several settings: 'Use system default' with radio buttons for 'Yes' and 'No' (selected); 'Protect by replicating snapshots' with radio buttons for 'Yes' (selected) and 'No'; '* Replicate to Ip Address' with a dropdown menu and a red 'X' icon; 'Snapshot and replicate every minute' with radio buttons for 'Yes' and 'No' (selected); 'RPO' with a text input field containing '1 hour'; 'Alert threshold hours to be set for per hour Snapshot' with a dropdown menu showing '15 minutes'; and 'Is replication to be paused' with radio buttons for 'Yes' and 'No' (selected).

Figure 13—Replicate VM workflow options

In addition to choosing the target Tintri VMstore to replicate to, the user can fine-tune the RPO. This allows snapshot replication to occur hourly or down to the minute.

The plugin is designed to be flexible, and just like the Snapshot Workflow, it can be embedded inside a customized workflow to provide simple and clear options to the requester.

Replicate VM Workflow Examples

Customization Example

In this example, you have decided that you wish to offer the user three options for replication.

- **Platinum:** Replication occurs every 15 minutes
- **Gold:** Replication occurs every hour
- **Silver:** Replication occurs twice a day

To do this, you create a new workflow called “Guided-Replicate VM” that includes the “Replicate VM” workflow.

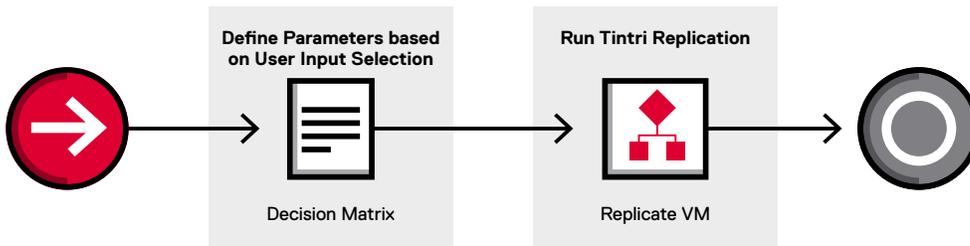


Figure 14—Guided-Replicate VM workflow

In this custom workflow we use a vRealize Orchestrator Scriptable Task to create a decision matrix which updates the other variables to be passed into the “Replicate VM” workflow provided by the plugin.

The tables on the following page show the defined schedules for each option.

Service Selection	Hourly	Hourly Per Minutes	Hourly Every Minutes	Hourly Days	Hourly Keep Local	Hourly Keep Remote	Hourly Snapshot Consistent
Platinum	Yes	5	15	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	6	4	Crash_consistent
Gold	Yes	5	60	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	8	6	Crash_consistent

Table 4—Protection schedule: Platinum and Gold

Service Selection	Daily	Daily Time	Daily Days	Daily Keep Local	Daily Keep Remote	Daily Snapshot Consistent
Silver	Yes	6:00am, 6:00pm	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday	3	3	Crash_consistent

Table 5—Protection schedule: Silver

Disaster Recovery Site Required	Variable “SourceIPAddress”
Denver	10.10.250.15
New York	10.20.250.25

Table 6—Additional parameters for DR site(s)

The end result of creating this decision matrix is the ability to customize the original workflow into a menu such as this:

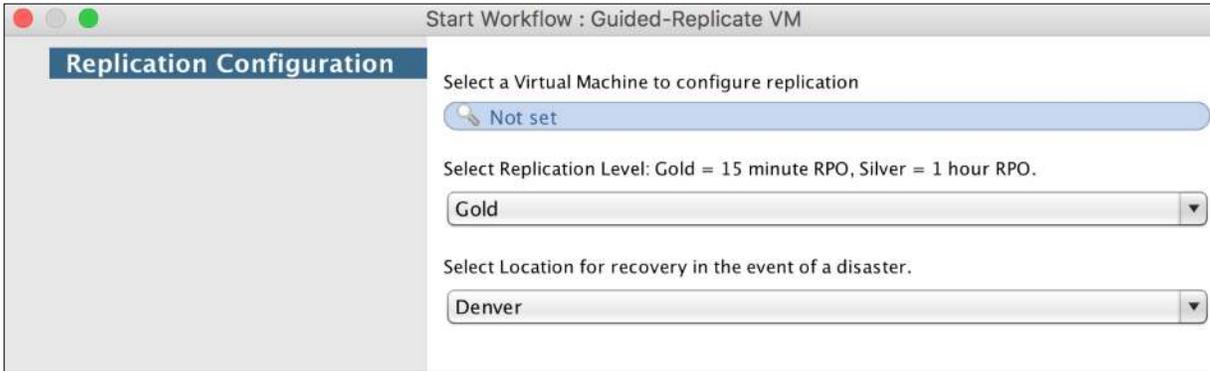


Figure 15—Customize the replication configuration of the original workflow

Self-Service Example 1: Running the Workflow During Blueprint Provisioning

Taking this customization a step further, it is possible to add the workflow to the Self-Service Portal in vRealize Automation using event subscriptions. Users may configure replication options at the same time they are provisioning a new virtual machine.

Step 1: Create Properties in vRealize Automation to match up with the vRealize Orchestrator inputs

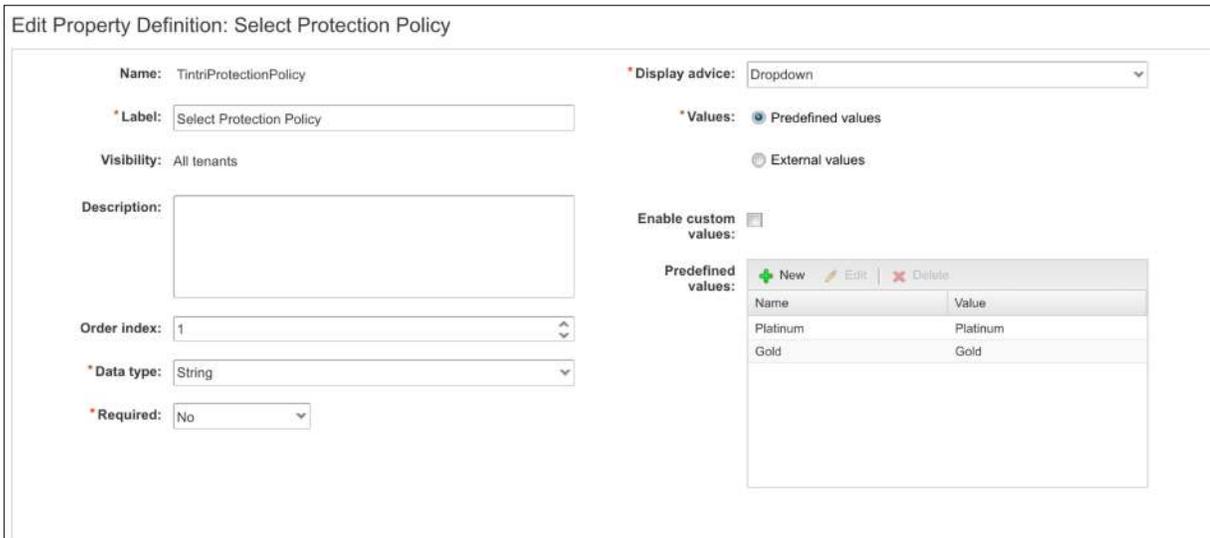


Figure 16— Step 1: Create Properties

Step 2: Create a Property Group to associate with the vRealize Automation Blueprint

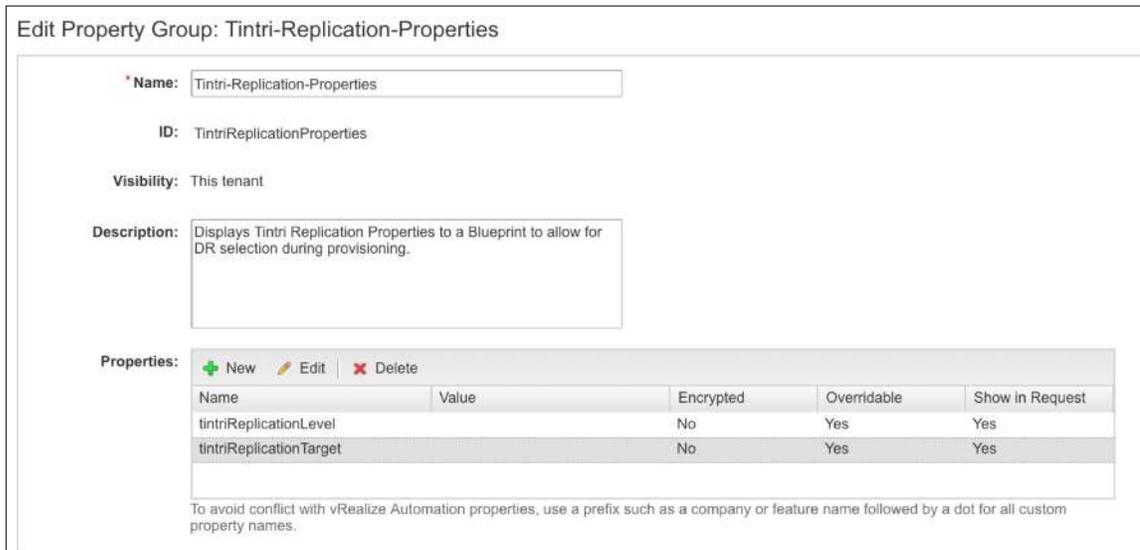


Figure 17—Step 2: Create a Property Group

Step 3: Add the Property Group to your vRealize Automation Blueprint

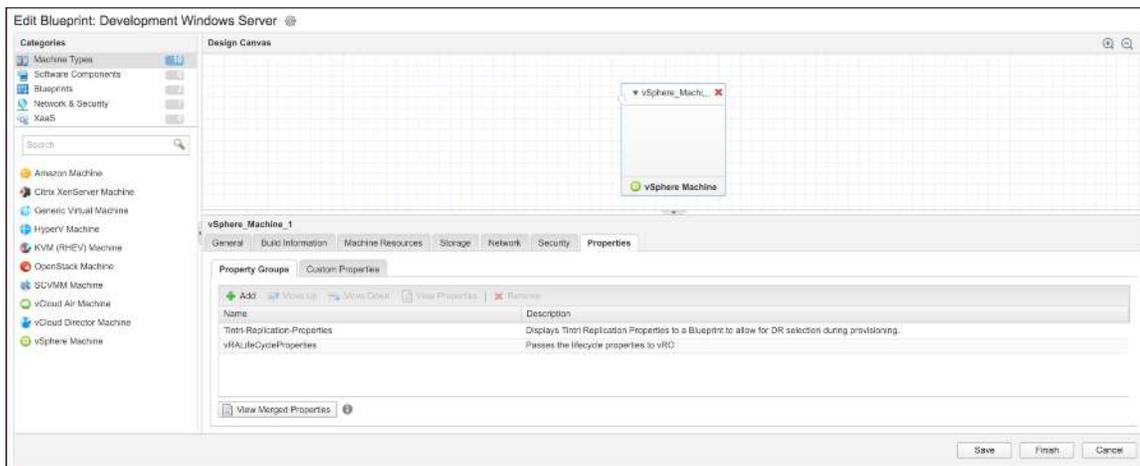


Figure 18—Step 3: Add the Property Group to your vRealize Automation Blueprint

Step 4: Add customized Tintri workflow to the vRealize Orchestrator workflow

This is intended to be executed at post-provisioning time. (This is the workflow that executes after vRealize Automation has completed building the VM and it is online and on the network.)

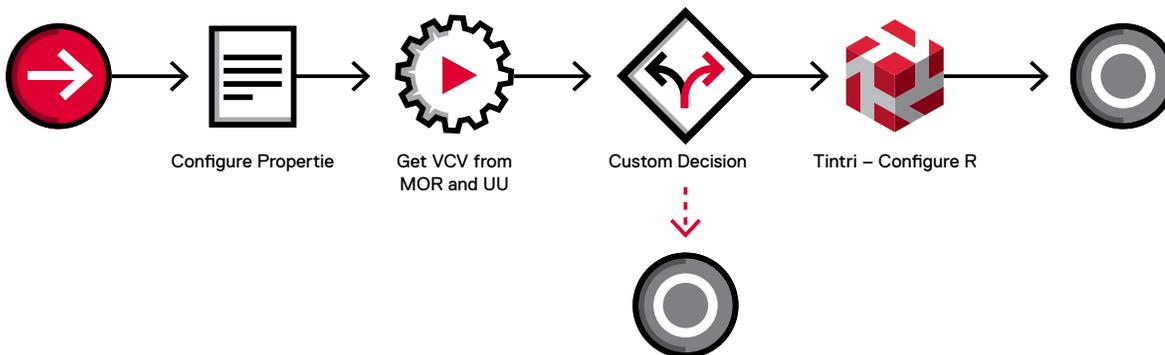


Figure 19—Step 4: Add Customized Tintri Workflow

Step 5: Add a vRealize Automation event subscription

If not already completed, associate your post-provisioning workflow with relevant blueprints.

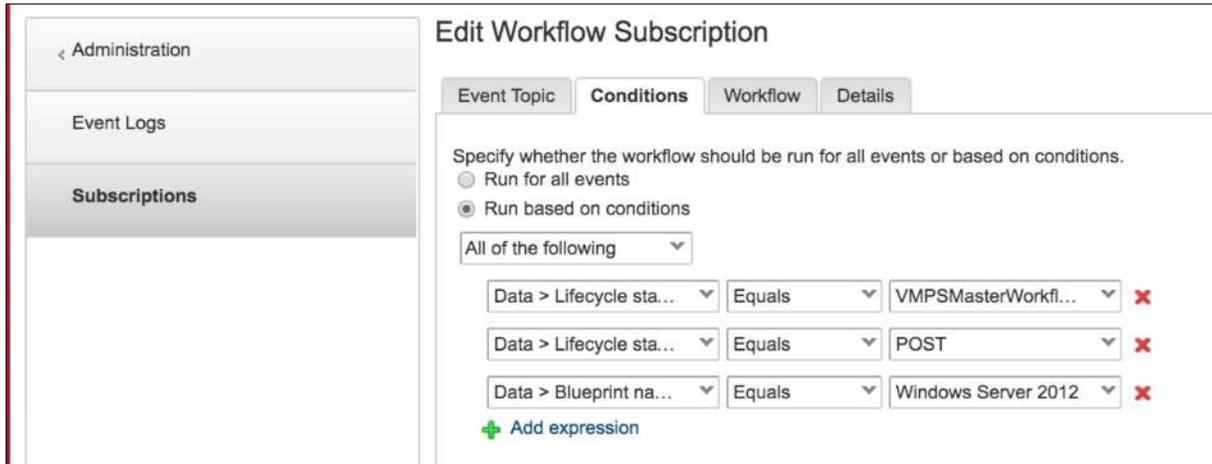


Figure 20—Step 5: Add an event subscription

The end result is a Catalog Item with the Tintri options visible to the requestor. These then get passed to the vRealize Orchestrator workflow during post build for execution.

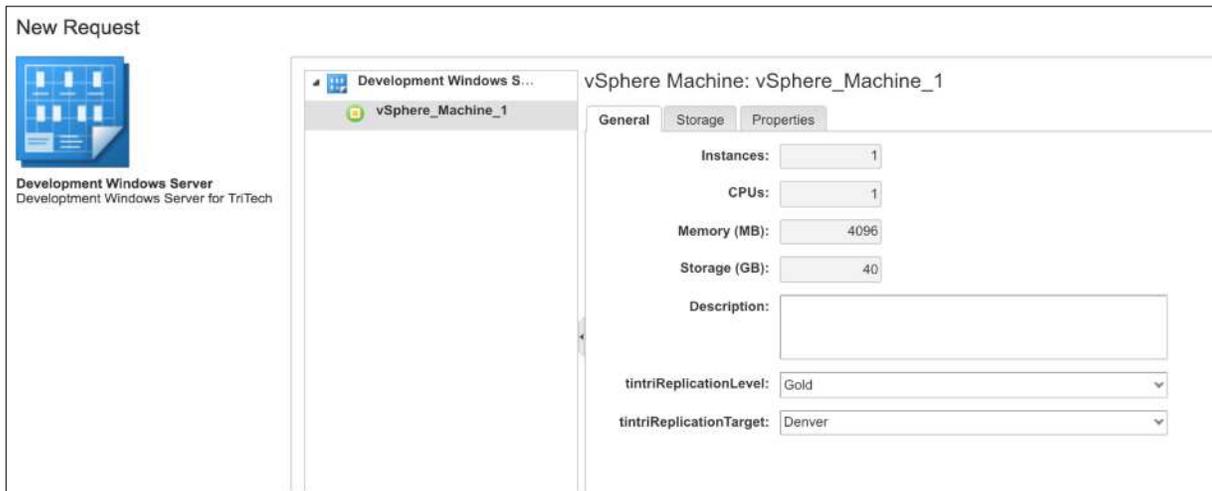


Figure 21—Step 6: The result

Self-Service Example 2: Enabling Replication as a Day 2 Operation

There are often cases where a service may not have had replication enabled during build and you'd prefer to offer options to enable and disable it as needed. This could also be tied back to a chargeback system. The approach is the same as that for Snapshot Day 2 Operations.

Step 1. Add the customized workflow as a resource action in vRealize Automation



Figure 22—Add customized workflow as a resource action

Step 2. Publish and entitle the action

Step 3. The end result is an action available in the items list for entitled users.

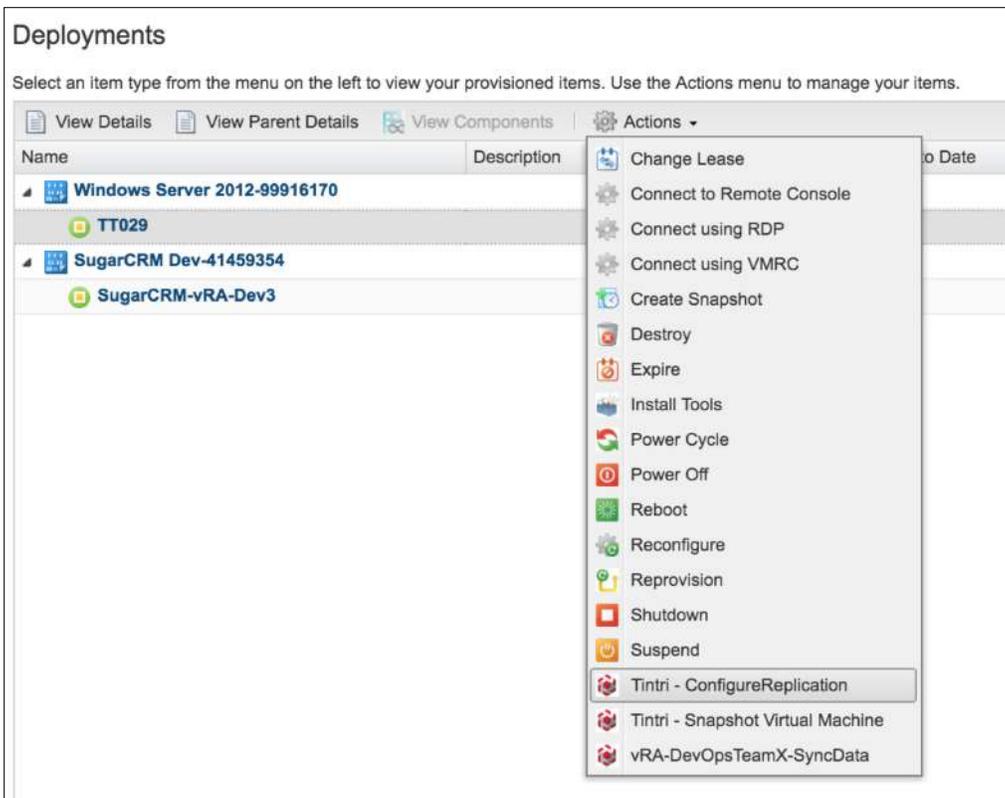


Figure 23—Deployments

Tintri SyncVM Use Case

SyncVM is a powerful feature which allows you to refresh disks from a source virtual machine to a target virtual machine. When you combine this with automation, it can empower your development groups, as well as support ad hoc requests for copying data between virtual machines.

Think about the use case below, where a developer has provisioned a new environment that contains an instance of SQL Server. The SQL Server is deployed, but it doesn't contain the latest set of data. By freshening the non-OS disks, you can ensure the developer has the latest set of data available in under a minute, without any impact to the SQL Server foundation or the production system.

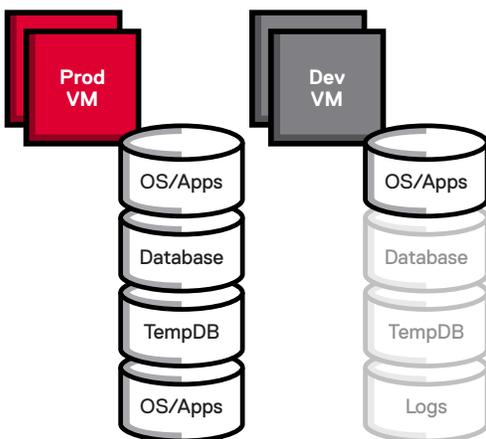


Figure 24—Refreshing non-OS disks

A significant benefit of this capability is that it utilizes the same techniques as Tintri's cloning capability for a whole VM. This means that

every clone has a zero byte cost and every copy of data made incurs no additional storage space usage, enabling broader utilization of the data. Every developer can have their own copy of the data and, with automation, they can do ad hoc updates as needed.

When performing a SyncVM operation, the Tintri VMstore synchronizes data from snapshots of the source VM to elements (vDisk, complete VM) of the destination VM. The Tintri vRealize Orchestrator Plugin contains a number of pre-built workflows associated with SyncVM which are located under the synchronization workflow folder.



Figure 25—Pre-built workflows associated with SyncVM

Executing the SyncVM Refresh Virtual Disks Workflow

Running Refresh Virtual Disks presents the user with options similar to those in the Tintri UI.

Step 1. Select the Target VM



Figure 26—Selecting the Target VM

Step 2. Select the Source VM and Snapshot

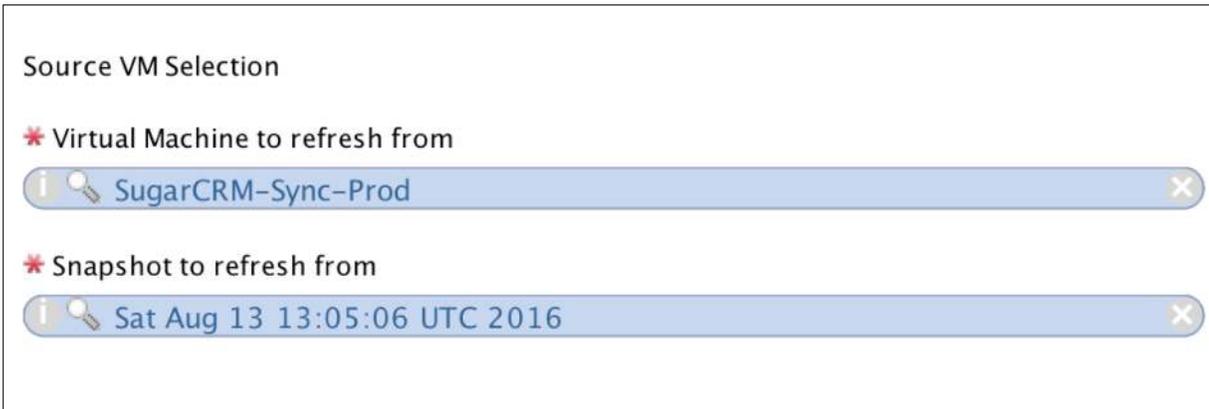


Figure 27—Selecting the Source VM and Snapshot

Step 3. Once Target and Source VMs are selected, a user must determine which disks they wish to overwrite in the Target VM, with

disks from the Source VM.

For example, a user could choose to overwrite SCSI 0:1 in the Target VM with SCSI 0:1 from the Source VM.

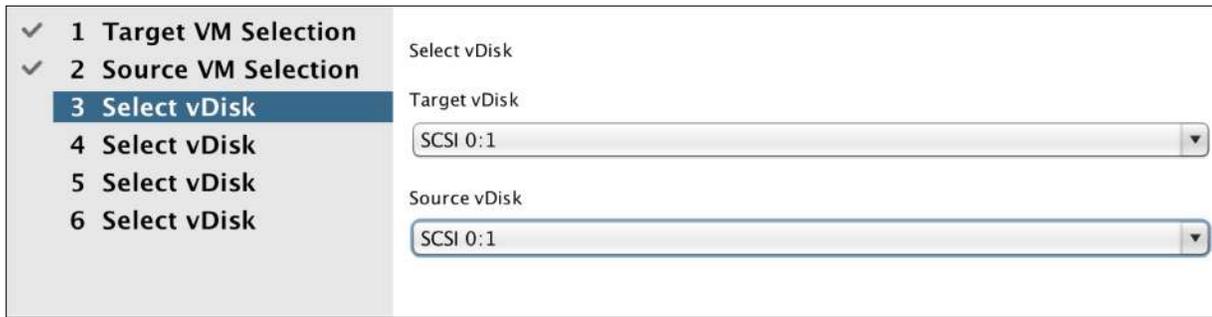


Figure 28—Choosing the Target vDisks to overwrite

Once executed, the Target VM will be powered off and the disks refreshed. This in itself is useful, but can be taken a step further using vRealize Orchestrator.

SyncVM Workflow Examples

Customization Example: Guided-Refresh Disks

The following example will use the pre-built workflow without having the user select the specific snapshot. Instead, the workflow will take an immediate snapshot to capture the latest data from the Source VM. Since the SCSI disks are known, they are preconfigured and the user does not need to select the specific SCSI disks. As illustrated in the screenshot below, only SCSI IDs SCSI 0:1, SCSI 0:2, and SCSI 0:3 on the Target VM are refreshed with those from the source VM.

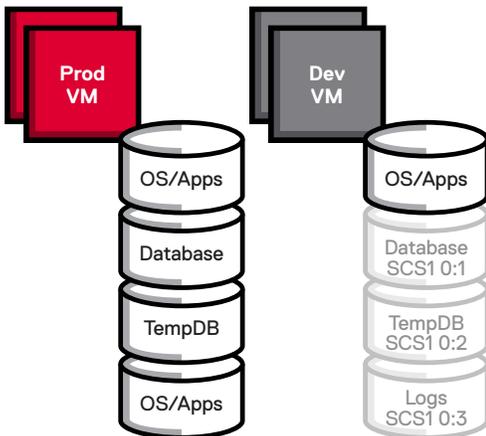


Figure 29—Immediate snapshot with pre-built workflow

In addition, the workflow executes a graceful shutdown of the Target VM when SyncVM occurs and also sends a notification email to the requester when the disk refresh has completed.

To do this, a wrapper workflow is built, which contains the Tintri “Refresh Disks” workflow as well as the additional modules to complete

the use case. The end result is a workflow that looks like this:

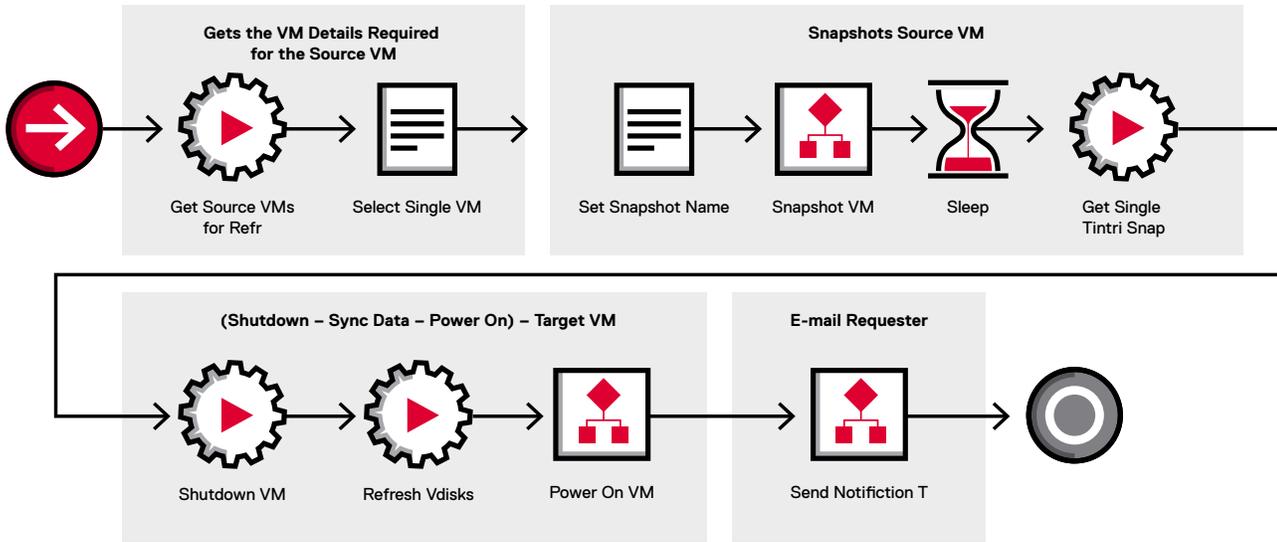


Figure 30—vRealize Orchestrator workflow with SyncVM plus immediate snapshot and "graceful" shutdown of Target VM

The steps necessary to create this workflow are as follows:

1. Get the VM details required for the SourceVM
 - a. **Tintri Action:** GetSourceVMsForRefresh — This is an action provided by Tintri as part of the vRealize Orchestrator plugin.
 - b. **Scriptable Task:** SelectSingleVM — A script created to automatically match the specific name of the VM selected. Because the VM name is known, the user doesn't have to select it from the potential Source VMs.
2. Snapshot Source VM
 - a. **Scriptable Task:** SetSnapshotName — A script to automatically generate a new and unique snapshot name.
 - b. **Tintri Workflow:** SnapshotVM — This is the same workflow discussed in the Snapshot use case earlier. The workflow simply makes use of it to initiate the on-demand snapshot as part of SyncVM.
 - c. **Sleep** — A five-second sleep to allow for the tiny amount of time it takes the VMstore to update the snapshot list before vRealize Orchestrator searches for it.
 - d. **Custom Action:** GetSingleTintriSnapshotFromVcVm — A custom action created for retrieving an individual Tintri Snapshot. This searches for the specific snapshot just created and makes sure it exists before SyncVM continues.
3. Shutdown — Sync Data — Power On Target VM
 - a. **vRealize Orchestrator Pre Built Action:** ShutdownVMandForce — This action initiates a graceful shutdown of the OS, and if after two minutes it does not shut down, initiates a hard power off. A choice can be made whether or not to use the force option depending on the use case.
 - b. **Tintri Action:** RefreshVDisks — This is the Tintri Refresh Disks action provided by the Tintri vRealize Orchestrator plugin.
 - c. **vRealize Orchestrator Pre-Built Workflow:** PoweronVMandWait — This powers on the Target VM and waits for VMware Tools to be started so that you can be sure the VM is online and running successfully.
4. Email Requester
 - a. **vRealize Orchestrator Pre-Built Workflow:** SendNotification — This workflow comes with vRealize Orchestrator and is used for sending an email to a requester.

The user inputs have now been drastically reduced to the following five:

- Source VM
- Target VM
- Email address for email notification
- Target VM Disks
- Source VM Disks

Running the workflow now presents the user with a much more concise menu of options that need to be filled in.

Figure 31—Menu of options

As with the replication use case, you can make SyncVM functionality available in two major places within vRealize Automation.

1. During the initial service request
2. During Day 2 operations so that the user can execute the workflow whenever it's required

Self-Service Example: Running the SyncVM Workflow During Blueprint Provisioning

To use the customized workflow, it first needs to be associated with the specific set of VMs a developer may be using. This is because the workflow includes a specific set of SCSI disks that matches the implementation of a developer VM.

Step 1. Wrap the workflow

This requires taking the customized workflow created in the previous section and adding it to a new workflow.

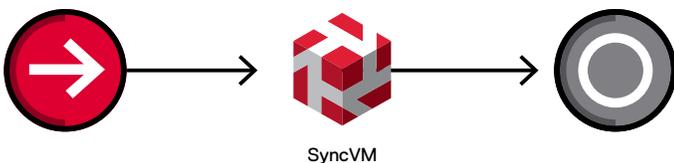


Figure 32—Wrap the workflow

Step 2. Set attributes

Instead of prompting the user for the disk configuration, you now set these as static general attributes. This must match the exact disks that must be synced between the source and target VM. In addition, you also hard set the Source VM to the specific VM you want to sync data from.

Name	Type	Value
<input type="checkbox"/> sourceVM	VC:VirtualMachine	SugarCRM-Sync-Prod
<input type="checkbox"/> targetvDisks	Array/string	Array [No Change , SCSI 0:1 , SCSI 0:2 , SCSI 0:3]
<input type="checkbox"/> sourcevDisks	Array/string	Array [SCSI 0:0 , SCSI 0:1 , SCSI 0:2 , SCSI 0:3]

Figure 33—Set attributes

Step 3. Add new workflow to vRealize automation

This piece is no different than the replication example and can be found in the vRealize documentation.

- a. Create vRealize Automation Properties
- b. Create vRealize Automation Property Group
- c. Add Property Group to relevant vRealize Automation Blueprints
- d. Create vRealize Automation Event Subscription to utilize the SyncVM workflow during post-build (vRA Machine. Provisioned)

The result is a blueprint which has a SyncVM option during build.

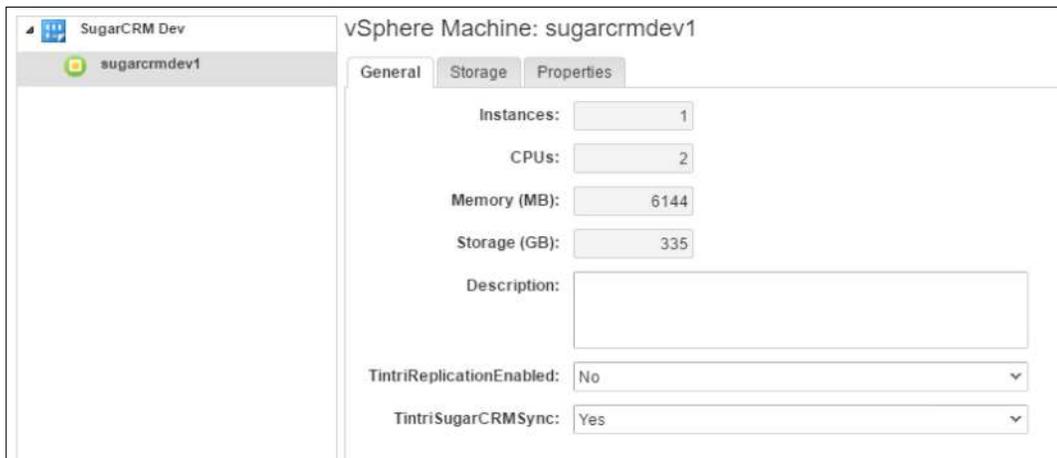


Figure 34—Blueprint with SyncVM option

Self-Service Example 2: Running SyncVM as a Day 2 Operation

Again, like the replication use case above, vRealize Orchestrator workflows can quickly be turned into Day 2 Operations.

1. Add the customized workflow as a resource action in vRealize Automation
2. Publish and entitle the action to specific user groups

Note: It is also important to lock down actions such as these to specific blueprints. If you blanket entitle this resource action to all blueprints, your developers could, for example, get into serious trouble syncing VMs that do not have the same disk configuration.



Figure 35—Turn the vRealize Orchestrator SyncVM workflow into a Day 2 Operation with limited availability

The end result for our Day 2 Operation allows the user to initiate a SyncVM on demand. In this example it is called “vRA-DevOpsTeamX-SyncData.”

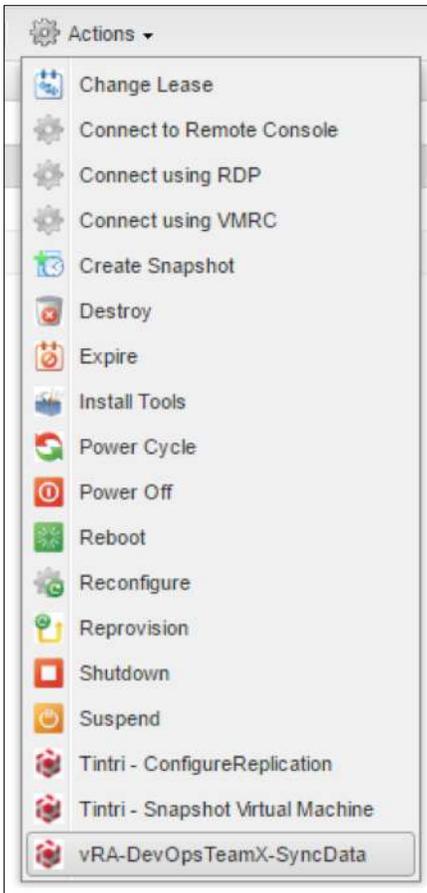


Figure 36—SyncVM workflow added to Actions menu.

Tintri QoS Use Case

Being able to set quality of service at the VM level from a native storage perspective is unique to Tintri. You can guarantee each application its own level of performance and protect others by limiting their performance as well. These capabilities change the way you approach the tiering of storage. No longer do you need to plan out your storage-based tiers on different pools of storage capabilities. You can use the same pool of storage, but have distinct levels of service based on the settings that make the most sense for your organization. When automation is combined with this feature, it opens up capabilities for both enterprises and service providers.

For enterprises, this feature gives you the ability to utilize a self-service portal to offer multiple performance levels. For most workloads, the default setting (no QoS assigned) allows the Tintri array to automatically adjust performance per VM. Workloads which need guarantees or specific limits can be configured to have these assigned per VM. The option to expose QoS through a self-service portal is as simple as a drop down. Contrasting this with traditional storage, VMs are required to be placed in the LUN or volume that matches the required storage tier. This requires decision workflows to determine if there is enough storage on a certain tier or if policies for protection match that tier. Tintri VMstore systems eliminate the need for these workflows and decisions, because QoS and protection configuration are done at the VM level.

Service providers can build service tiers where customer VMs automatically get a specific maximum throughput. This gives you the option of charging for guaranteed IOPS. By utilizing logic at the automation tier, customers can automatically be placed into per-VM configurations

that limit the maximum amount of IOPS. If customers would like to have their VM modified for a higher tier of storage performance, their actual blocks of storage do not need to be migrated to that tier, as they can be instantly adjusted to the level of performance they desire.

Service Level	Maximum IOPs	Minimum IOPs	Cost
Default	0	0	\$
Tier 1	10000	5000	\$\$\$
Tier 2	5000	0	\$\$
Tier 3	2000	0	\$

Table 7—Matrix for self-service Enterprise users

To configure this in vRealize Orchestrator, a simple scriptable task can be utilized to set the variables that will configure these tiers.

The scriptable task is put in front of the main action:

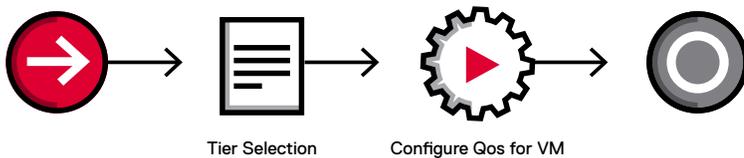


Figure 37—Workflow for configuring QoS

A series of if /else if decisions set the necessary variables, which are then passed into the action:

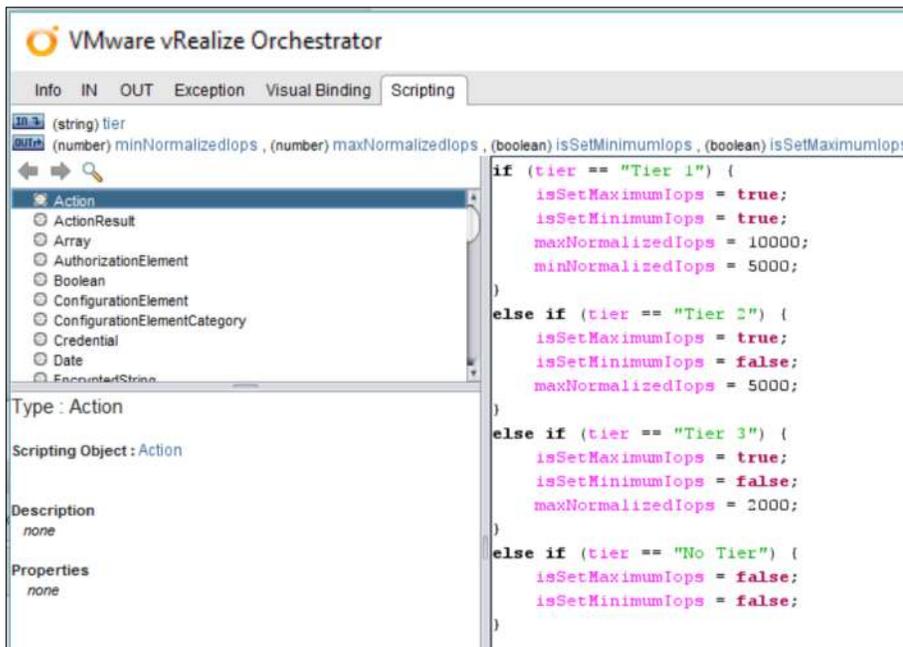


Figure 38—Setting variables for QoS using if and else if statements in vRealize Orchestrator

These tier options match the sets of options in the table above. The one difference is also setting the variables for whether or not the IOPS are being set for Min/Max.

Tintri File Restore Use Case

Being able to snapshot a VM not only allows you to restore an entire VM, but also individual files. Another workflow included in the

plugin is the “Restore Files” workflow.

Executing the Restore Files Workflow

Running this workflow allows the user to first select the individual Virtual Machine they wish to restore files for.



Figure 39—Restore Files workflow: VM Details

Once selected the user is then able to choose a specific Tintri Snapshot to restore files from. If you’ve setup a protection policy as discussed in the earlier use cases, then the user should have a number of points in time to choose from when restoring files.

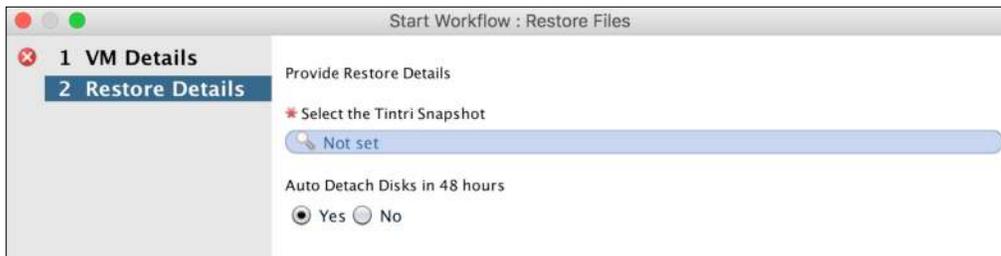


Figure 40—Restore Files workflow: Restore Details

Once the disk is attached it will be presented to the operating system, and the user can also decide to auto-detach the disk in 48 hours if desired.

Combining this workflow with a wrapper workflow, as in other use cases above, allows you to provide self-service file restore to users.

Conclusion

In this paper, we discussed how Tintri VM-aware storage capabilities operate and complement cloud environments. It is clear that using Tintri storage in combination with vRealize Automation and vRealize Orchestrator results in a highly efficient private cloud solution. By utilizing Tintri clones, snapshots, and replication, and integrating those capabilities into a self-service portal, the promise of the private cloud—lower costs and faster time to market—can be achieved. The Tintri plugin for vRealize Orchestrator automates common Tintri workflows for use within vRealize Orchestrator, enabling the creation of advanced and efficient cloud services.