IMPLEMENTING A PRIVATE CLOUD WITH VMWARE VREALIZE AND DELL EMC XTREMIO X2

Abstract

This Reference Architecture describes the integration capabilities and benefits of Dell EMC XtremIO X2 all-flash storage array for creating a private cloud solution with a VMware vSphere infrastructure, using VMware vRealize Orchestrator and vRealize Automation, and the XtremIO plugin for vRealize Orchestrator.

April 2018
# Contents

Abstract ................................................................................................................................. 1
Executive Summary ............................................................................................................... 5
Business Case ...................................................................................................................... 5

| Audience and Whitepaper Content | 6 |

Solution Overview ............................................................................................................. 7

| Dell EMC XtremIO X2 | 7 |
| VMware vRealize Orchestrator | 9 |
| Workflow Development and Management | 10 |
| Plug-in SDK | 10 |
| Integration | 11 |
| VMware vRealize Automation | 11 |
| Key Features | 12 |

| Hosts and Networks | 14 |
| Compute Hosts: Dell PowerEdge Servers | 14 |
| Compute Integration – Dell OpenManage | 14 |
| Storage Networks | 15 |
| Other Software Components | 16 |

Dell EMC XtremIO X2 for Private Cloud Environments .................................................... 17

| XtremIO X2 Overview and Architecture | 17 |
| Multi-Dimensional Scaling | 18 |
| Inline Data Reduction | 19 |
| Data Deduplication | 19 |
| Data Compression | 20 |
| Thin Provisioning | 20 |

| Copy Data Services | 21 |
| Integrated Copy Data Management | 21 |
| XtremIO Virtual Copies | 21 |

| Native Replication | 22 |
| VMware APIs for Array Integration (VAAI) | 23 |

| Other System Features | 25 |
| XtremIO Data Protection | 25 |
| Write Boost | 25 |

| Data at Rest Encryption | 26 |
| XtremIO Management Server | 27 |

XtremIO Plugin for VMware vRealize Orchestrator ......................................................... 32
Executive Summary

This white paper reviews the integration of VMware vSphere virtual infrastructure with the vRealize Orchestrator (vRO) and vRealize Automation (vRA) tools when using Dell EMC XtremIO X2 all-flash storage array as the storage backend of the environment. It discusses the benefits of using XtremIO X2 for virtualized and cloud environments and describes its integration in such a solution using the XtremIO plugin for VMware vRealize Orchestrator, and how it can be leveraged for automation with the VMware vRealize Automation tool. XtremIO storage-related, built-in and customized vRO workflows and vRA services will be showcased in this white paper as a proof of concept for XtremIO’s integration in the solution, together with examples of how to use and work with these private cloud tools.

Business Case

Private cloud solutions are emerging all over in the Information Technology (IT) world to provide IT departments with the ability to design automated workflows that mirror the organization’s IT procedures (simple or complex), and to provide a way to offload the operation of some of those procedures to their customers (application teams) for quicker deployment and provisioning of computer resources and services in the organization. One such private cloud solution is VMware’s vRealize Suite. The Suite equips IT departments with the vRealize Orchestrator (vRO) tool that enables using built-in workflows and designing new ones that suit the organization’s processes, and the vRealize Automation (vRA) tool that integrates with vRO and enables the automation and offloading of processes as services to the customers. The Suite also includes vRealize Operations Manager (that collects performance and capacity data and monitors the infrastructure) and vRealize Log Insight (that provides real-time log administration across all of the cloud’s components), but the focus in this paper will be on the automation and orchestration tools.

With vRealize Orchestrator, clients can easily design and deploy scalable workflows to automate simple or complex IT processes, which can significantly simplify and accelerate IT operations in the organization. By using this tool, IT departments can reduce their operating expenses and be less susceptible to user errors, thus improving their service delivery efficiency and operational management. vRO enables customers to create their own workflows to accommodate IT operations in their environment, and leverage built-in workflows and actions based on SOAP, REST operations and PowerShell scripts. Using plug-ins and external packages, customers can automate the management and operational tasks of third-party applications, tools and infrastructure, further utilizing vRO and simplifying IT processes across all components of their datacenters.

The vRealize Automation tool provides IT automation through the creation and management of personalized infrastructure, applications and custom IT services across multi-vendor and multi-cloud environments. It accelerates the end-to-end delivery of those applications and services and ensures that clients receive the right size of resources at the appropriate service level for the jobs they need to perform. Those otherwise manual, time-consuming procedures will now become automatic, thus reducing the operational cost of managing the environment and making IT service delivery processes more responsive to business needs. vRA presents a self-service interface that can rapidly deliver building blocks to developers and application teams. With vRA you can streamline and automate the lifecycle management of an environment, from initial deployment and on-going rebalance of resources to the retirement and reclamation of those resources. A tight integration with vRealize Orchestrator makes coupling these two tools invaluable for the organization’s automation ambitions, as vRA can publish and combine vRO workflows as services to clients.
To complement a private cloud solution, IT departments must choose a storage system that will be fast and efficient for cloud environments and one that integrates well with the chosen private cloud software. Dell EMC’s XtremIO X2 storage system fits those needs entirely. The proficient new-generation all-flash array is a perfect fit for cloud environments. Its unique scale-out architecture enables servicing dynamic amounts of data at each scale, and its Data Reduction abilities (such as thin provisioning, deduplication and compression) reduce the physical storage needed for the logical data by at least a few orders of magnitude, allowing customers to save in both space and storage costs. Being a 100% flash-based technology, XtremIO was built specifically to utilize its flash disks at an optimal level, which allows delivering ultra-high performance for the cloud infrastructure at a very low latency, for both FC and iSCSI connections. XtremIO’s Copy Data Services also present a great benefit for cloud environments, as entire projects and tenants in the cloud can be copied to new test, development and analytics projects for almost no extra space in the storage system and with no performance degradation to either environment. XtremIO comes with an easy-to-use user interface to provide storage administrators a quick and convenient way of setting up enterprise class storage environments, provisioning storage to client hosts and applications, and monitoring performance. With the XtremIO plugin for vRealize Orchestrator (and other plugins for VSI and for vRealize Operations Manager and Log Insight), XtremIO integrates with VMware’s cloud solution to provide the customer with all of its storage needs, and enables management and automation of services through both the vRO and vRA tools.

**Audience and Whitepaper Content**

This Whitepaper is intended for those involved in adopting, deploying and managing a private cloud environment using VMware vRealize Orchestrator and vRealize Automation tools together with Dell EMC’s XtremIO as the environment’s storage system. To help you navigate your way through this paper, use the following index:

- For an overview of the private cloud solution, including hardware and software recommended components – see the **Solution Overview** chapter.
- For an overview of Dell EMC XtremIO X2 Storage Array, including its benefits for private cloud environments – see the **Dell EMC XtremIO X2 for Private Cloud Environments** chapter.
- For an overview of the VMware vRealize Orchestrator tool, the XtremIO Plugin for vRO, a guide for how to create and design Actions and Workflows and how to run them – see the **XtremIO Plugin for VMware vRealize Orchestrator** chapter.
- For an overview of the VMware vRealize Automation tool, a guide for how to create and provision XaaS Resources and Blueprints and how you can integrate your XtremIO storage with vRA – see the **Integrating vRA with vRO and the XtremIO Plugin** chapter.
- For a quick guide on how to initially configure vRO against your vCenter and XtremIO instances – see **Appendix B – vRO Initial Configuration**.
- For a quick guide on how to initially configure vRA against your vCenter and vRO instances – see **Appendix C – vRA Initial Configuration**.
Solution Overview

The solution used for this reference architecture was comprised of several software and hardware elements, which will be described in this section. From a hardware perspective, we are using XtremIO X2, Dell EMC’s new-generation all-flash array, as our storage provider, and Dell PowerEdge servers as the hosts for our virtual environment. We are also using Brocade and Mellanox switches for FC and Ethernet connectivity accordingly. As for software, we are using VMware vSphere as the virtualization infrastructure with the EMC VSI plugin, and the vRealize Suite as our private cloud infrastructure, which includes vRealize Log Insight, vRealize Operations Manager, vRealize Orchestrator and vRealize Automation. In this chapter, we mainly elaborate on our XtremIO storage system and the vRO and vRA tools.

Dell EMC XtremIO X2

Dell EMC’s XtremIO is an enterprise-class scalable all-flash storage array that provides rich data services with high performance. It is designed from the ground up to unlock flash technology’s full performance potential by uniquely leveraging the characteristics of SSDs, and uses advanced inline data reduction methods to reduce the physical data that must be stored on the disks.

XtremIO’s storage system uses industry-standard components and proprietary intelligent software to deliver unparalleled levels of performance and achieve consistent low latency for up to millions of IOPS. It comes with a simple, easy-to-use interface for storage administrators and fits a wide variety of use cases for customers who need a fast and efficient storage system for their datacenters. XtremIO requires very little planning before setting-up, and storage provisioning can be initiated quickly after installation.

XtremIO leverages flash to deliver value across multiple dimensions:

- Performance (consistent low-latency and up to millions of IOPS).
- Scalability (using a scale-out and scale-up architecture).
- Storage efficiency (using data reduction techniques such as deduplication, compression and thin-provisioning).
- Data Protection (with a proprietary flash-optimized algorithm named XDP).
- Environment Consolidation (using XtremIO Virtual Copies or VMware's XCOPY).

XtremIO X2 is the newest generation of Dell EMC’s advanced all-flash storage system, and adds enhancements and flexibility in several aspects to customers. The new features include more flexibility in the storage system in the form of scaling-up (in addition to already existing scale-out capabilities), higher performance due to a more powerful hardware and a new write boost addition to the I/O flow, improved data availability with the use of NVRAM for protecting data, and highly intuitive management of the storage system provided by the new web-based UI.

Figure 1. XtremIO X2 in Comparison to X1
Figure 2 shows XtremIO X2’s incredible performance for an intensive live VMware production environment. We can see extremely high IOPS (~1.6M) stats handled by XtremIO X2 storage array with latency mostly below 1msec. In addition, we can see an impressive data reduction factor of 6.6:1 (2.8:1 for deduplication and 2.4:1 for compression) which lowers the physical footprint of the data.

In our environment we are using an XtremIO cluster with a single X2-S X-Brick (see the XtremIO X2 Overview and Architecture section for the differences between X-Brick types) as the storage array for the solution. The X-Brick contains 36 400GB SSD drives which, after leaving capacity for parity calculations and other needs, provides about 11.21TB of physical capacity. The logical capacity of this cluster however is much higher, as XtremIO’s data reduction techniques increases the data that can be stored on the array by potentially a few hundred percent, especially for virtual and cloud environments, which have high deduplication ratios and that allow the storage array to save plenty of space with thin provisioning. 36 SSDs are half the amount that can fit in a single X-Brick, meaning that in terms of capacity, we can double the size of this X-Brick (scale-up) before adding more X-Bricks to the cluster for additional capacity and performance (scale-out).

We will further review XtremIO X2 architecture, capabilities and benefits for the cloud in the Dell EMC XtremIO X2 for Private Cloud Environments chapter.
VMware vRealize Orchestrator

VMware vRealize Orchestrator (vRO) is the orchestrator tool included in VMware’s vRealize Suite which is designed to simplify the automation of IT tasks for customers. It seeks to improve service delivery efficiency, operational management and IT agility in an organization. It is built with an open and flexible architecture which system administrators and IT operations staff can use to streamline tasks and integrate functions with third-party software through workflows. Performance and scalability within the vRO enables it to execute hundreds to thousands of concurrent workflows on single or distributed architectures.

We are using VMware vRealize Orchestrator version 7.3 in our cloud solution.
Workflow Development and Management

vRO enables users to create their own workflows through the designer menus using an intuitive, easy-to-use, drag-and-drop workflow creator. Workflow designers can use existing Actions and Workflows, the tool’s scripting engine (based on JavaScript) and imported Actions and Workflows from third-party plugins to produce their workflows. Any task that can be scripted or automated using a set of inputs (i.e. network configurations, generating work order tickets, updating a CMDB, initiate system backups, etc.) can be designed as a Workflow on vRO and then used for faster deployment and provisioning, and to reduce user errors during operation. Every step in every workflow is saved in a content database which retains the state and context perimeters of an environment for the on-going activity of vRO in that environment.

Figure 4. Designing a Workflow in vRO

Plug-in SDK

The vRO Plug-in Software Development Kit (SDK) jump-starts new vRO plug-in developers and enables advanced developers to integrate key SDK features into their existing development environments. It provides an Eclipse add-on for plug-in development.
Integration

VMware vRealize Orchestrator can be integrated with various technologies to allow designing and running workflows on VMware-based tools and third-party applications that are part of a customer’s cloud infrastructure. The tool provides 100% coverage of vSphere APIs and is integrated with other vRealize solutions such as vRealize Automation and vRealize Operations. Using plug-ins and workflows published on VMware Solution Exchange [14] (a marketplace of extensible solution plug-ins) users can integrate vRO with other components in their cloud infrastructure such as storage, network, databases, Active Directory, etc. The orchestrator also provides a REST API to enable other applications to integrate with the tool and design and execute their own workflows.

Running vRO Workflows through vSphere vCenter

vRealize Orchestrator can be registered in an integrated vSphere vCenter instance as an extension which allows managing and running its workflows through the vSphere web client. This provides increased interoperability between the vRO and vCenter, and supplies a way to execute the complex operations that are designed in the Orchestrator directly through the vSphere UI. Running vRO Workflows through the vCenter will be demonstrated later in this paper.

![Figure 5. vSphere vCenter vRO Extension](image)

VMware vRealize Automation

VMware vRealize Automation (vRA) enables IT teams to remove process inefficiencies in the organization through the use of end-to-end automation that helps IT better serve DevOps and application teams. It delivers IT Automation across a hybrid cloud from Day-One service provisioning through Day-Two operational capabilities, thus helping IT teams automate core IT processes, speed up infrastructure delivery and get the most out of both hardware and human resources. With vRA, clients, it can automate the provisioning of all infrastructure components, including compute, network, storage and security resources. vRA provides management of infrastructure resources through their entire lifecycle – from initial deployment through resizing, reclamation and retirement. Customers can extend automation with vRA across other IT ecosystems by integrating with prevailing third-party tools through a full spectrum of extensibility options.
VMware vRealize Automation gives IT teams the ability to provide developers with easy access to traditional and cloud native application resources through a rich set of self-service capabilities while simultaneously supporting the need for developers to use the tool chain of their choice. vRA provides developers with access to infrastructure services through catalogs, API or CLI and gives them free use of their own resource pools. It also supports out-of-the-box integration with Puppet, leveraging a framework for integration configuration management tools to simplify the delivery and management of second generation application stacks. With vRA, customers can discover and manage container hosts and containers through an embedded container management platform that supports VMware vSphere Integrated Containers and Docker Containers, and can use a mix mode of applications blueprints that contain both virtual machines and containers.

We are using VMware vRealize Automation version 7.3 in our cloud solution.

**Key Features**

vRA provides a variety of features to adopters, mainly including the following:

- **Self Service** - Via a unified IT service catalog (see Figure 7), vRA delivers infrastructure, container, application and custom services (XaaS) to clients, who can choose and deploy them themselves with a click of a button, skipping tedious requests and approvals from IT teams. Policy-based governance ensures the right service level meets specific business needs, and the automation of the entire process accelerates IT service delivery all around.

- **Unified Blueprint Model** - Customers are presented with several methods and resources for designing their desired automated IT processes. The web application’s drag-and-drop Design Canvas helps streamlining and visualizing the design process by assembling applications from pre-built components. Blueprints can be presented by code, allowing exporting, importing and editing automation blueprints as text, and can also be designed and managed entirely via API calls. An entire marketplace of VMware and partner-provided blueprints for customers interested in leveraging existing blueprints for several technologies is available on VMware Solution Exchange [14].

- **Multi-Vendor, Hybrid Cloud Deployments** - Customers are given the flexibility to choose the right cloud platform and location that meets their business needs, and are handled with consistent governance and control across hybrid cloud deployments in vRealize Automation.
- **Extensible Automation Platform** - vRA comes with an extensible platform that enables customization and extensibility at multiple levels across IT ecosystems. It enables to design and automate the delivery of any IT service through service orchestration, and can also make use of VMware and partner-provided integration solutions that are available through VMware Solution Exchange \[14\].

Figure 7. VMware vRealize Automation Self-Service Catalog Example
Hosts and Networks

Compute Hosts: Dell PowerEdge Servers

We have homogenous clusters of ESX servers for hosting the cloud environment and virtual appliances, all comprised of Dell’s PowerEdge FC630 servers that have powerful compute resources for handling medium-to-large scale private cloud environments. We recommend separating the ESX clusters such that the virtual machines and services created on the private cloud reside on one cluster and virtual appliances and services created for the private cloud (vRealize appliances, plugins, vCenter Server etc.) reside on a different cluster.

Table 1 lists ESX hosts details at our environment.

Table 1. ESX Hosts Used for our Cloud Solution

<table>
<thead>
<tr>
<th>PROPERTIES</th>
<th>DELL POWEREDGE FC630 HOSTS (NUMBERS ARE PER HOST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cores</td>
<td>36 CPUs x 2.10GHz</td>
</tr>
<tr>
<td>Processor type</td>
<td>Intel Xeon CPU E5-2695 v4 @ 2.10GHz</td>
</tr>
<tr>
<td>Processor Sockets</td>
<td>2</td>
</tr>
<tr>
<td>Cores per socket</td>
<td>18</td>
</tr>
<tr>
<td>Logical processors</td>
<td>72</td>
</tr>
<tr>
<td>Memory</td>
<td>524 GB</td>
</tr>
<tr>
<td>Ethernet NICs</td>
<td>4</td>
</tr>
<tr>
<td>Ethernet NICs type</td>
<td>QLogic 57840 10Gb</td>
</tr>
<tr>
<td>iSCSI NICs</td>
<td>4</td>
</tr>
<tr>
<td>iSCSI NICs type</td>
<td>QLogic 57840 10Gb</td>
</tr>
<tr>
<td>FC adapters</td>
<td>4</td>
</tr>
<tr>
<td>FC adapters type</td>
<td>QLE2742 Dual Port 32Gb</td>
</tr>
<tr>
<td>On-board SAS controller</td>
<td>1</td>
</tr>
</tbody>
</table>

The sizing of your environment (i.e. numbers of ESX hosts and clusters) is not discussed in this paper and should fit the size of your private cloud and allow scalability.

Compute Integration – Dell OpenManage

Dell OpenManage is a program providing simplicity and automation of hardware management tasks and monitoring for both Dell and other vendors’ hardware systems. Among its capabilities are:

- Rapid deployment of PowerEdge servers, operating systems, and agent-free updates.
- Maintenance of policy-based configuration profiles.
- Streamlined template-driven network setup and management for Dell Modular Infrastructure.
- Providing a “geographic view” of Dell-related hardware.

Dell OpenManage can integrate with VMware vCenter using the OpenManage Integration for VMware vCenter (OMIVV), which provides VMware vCenter with the ability to manage a datacenter's entire server infrastructure, both physical and virtual. It can assist with monitoring the physical environment, sending system alerts to the user, and rolling out firmware updates to an ESXi cluster. The integration is more profitable and contains more features when using Dell PowerEdge servers as the ESX hosts of the VMware environment.
Figure 8 shows an example of a cluster's hardware information provided by the OpenManage Integration for VMware vCenter.

Storage Networks

Our environment is configured to be able to use both FC and iSCSI connectivity between the physical hosts and storage system. We are using Brocade switches for FC connectivity and Mellanox switches for iSCSI connectivity, with both networks fully redundant. Connection between the storage and the servers is done via multiple paths. Table 2 details the hardware and configuration of the storage network’s switches.

<table>
<thead>
<tr>
<th>HARDWARE</th>
<th>CONFIGURATION</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brocade 6510 SAN switch</td>
<td>64 ports of 32 Gbps FC switches</td>
<td>2 switches per site, dual FC fabric configuration.</td>
</tr>
<tr>
<td>Mellanox MSX1016 10GbE</td>
<td>64 ports of 10 Gbps Ethernet switches</td>
<td>2 switches per site, Infrastructure Ethernet switches.</td>
</tr>
</tbody>
</table>
Other Software Components

As mentioned, our solution includes other software components that we will not focus on in this white paper. Applications, plug-ins and appliances that take part in our private cloud infrastructure are not a must, but are recommended as part of the private cloud solution for better interoperability within the infrastructure. Table 3 lists these software components and their use in the environment.

<table>
<thead>
<tr>
<th>SOFTWARE</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCenter Server VM 6.5 update 1d</td>
<td>The server that runs vCenter and the associated vCenter services.</td>
</tr>
<tr>
<td>EMC Solutions Integration Service (SIS) 7.2 VM and EMC Virtual Storage Integrator (VSI) for VMware vSphere 7.2</td>
<td>EMC SIS provides unique storage integration capabilities between VMware vSphere 6.5 and Dell EMC XtremIO X2 (XMS 6.0.0 and above). The EMC VSI plugin is used for the interaction between VMware vSphere and the XtremIO storage system and can be registered via EMC SIS.</td>
</tr>
<tr>
<td>vRealize Operations Manager (vROPS) 6.6 VM with the EMC Storage Analytics (ESA) plugin 4.4</td>
<td>VMware vROPS collects performance and capacity data from monitored software and hardware resources in the virtual infrastructure. The ESA package enables the collection of analytical data from Dell EMC resources, specifically XtremIO storage, through vROPS.</td>
</tr>
<tr>
<td>vRealize Log Insight 4.5 VM with the Dell EMC XtremIO X2 Content Pack</td>
<td>VMware vRealize Log Insight provides real-time log administration for environments that span across physical, virtual and private cloud infrastructure. The Dell EMC XtremIO X2 Content Pack integrated into vRealize Log Insight to provide predefined dashboards and user-defined fields specifically for log administration of XtremIO arrays.</td>
</tr>
<tr>
<td>vSphere ESXi 6.5 update 1d</td>
<td>VMware’s newest-version hypervisor.</td>
</tr>
</tbody>
</table>
Dell EMC XtremIO X2 for Private Cloud Environments

XtremIO X2 storage system serves many use cases in the IT world, due to its high performance and advanced abilities. One major use case is for virtualized environments and cloud computing. In this section we will detail XtremIO X2’s architecture and some of its key advantages for private cloud environments, such as those implemented by VMware’s vRealize tools, together with other important features of the all-flash storage array.

XtremIO X2 Overview and Architecture

An XtremIO X2 Storage Array is comprised of building blocks called X-Bricks. Each X-Brick is composed of two storage controllers (SCs) – containing the CPUs, RAM and the ports that connect them to the storage network – and one Disk Array Enclosure (DAE) – that can hold up to 72 SSDs to store data (18 minimum). X-Bricks can be clustered together to create a stronger (high-performant) and larger (high capacity) storage array.

There are three types of X2 X-Bricks: X2-S, X2-R and X2-T. The X2-S type is for environments whose storage needs are more I/O intensive than capacity intensive, as they use smaller SSDs and less RAM. X2-R X-Bricks clusters are made for the more capacity intensive environments, with bigger disks, more RAM and a bigger expansion potential for future code releases. The X2-T type is for customers in need for midrange storage only, as they allow only a single X-Brick and can hold a maximum of 36 SSDs. The different X-Brick types cannot be mixed together in a single system, so the decision of which type is suitable for a specific environment should be made in advance (X2-T clusters can be expanded to X2-R clusters, but with additional cost).

XtremIO architecture is based on a metadata-centric content-aware system, which helps to efficiently streamline data operations without requiring any movement of data post-write for any maintenance reason (data protection, data reduction, data encryption, etc. – all done inline). The system lays out the data uniformly across all SSDs in all X-Bricks using unique fingerprints of the incoming data and controls data access using metadata tables. This contributes to an extremely balanced system across all X-Bricks in terms of compute power, storage bandwidth and capacity.

Using the same unique fingerprints, XtremIO is equipped with exceptional always-on, in-line data deduplication abilities, which highly benefits virtualized and cloud environments. Together with its data compression and thin provisioning capabilities (both also in-line and always-on), it achieves incomparable data reduction rates.

System operation is controlled by storage administrators via a stand-alone dedicated Linux-based server called the XtremIO Management Server (XMS). A new intuitive user interface is used to manage and monitor the storage cluster and its performance. The XMS can be either a physical or a virtual server and can manage multiple XtremIO clusters (up to 8 clusters at the time of writing this document).

With its intelligent architecture, XtremIO provides a storage system that is easy to set-up, needs zero tuning by the client, and does not require complex capacity or data protection planning, as the system handles it on its own.
Multi-Dimensional Scaling

Cloud environments tend to be very dynamic in size and performance needs. Many configurations start small and increase in size as their use increases over time. Such environments need a storage system that can start small and grow in both size and performance according to its needs.

XtremIO’s storage system is built in a way that satisfies this use case perfectly, and now with much more flexibility in terms of scaling. An XtremIO X2 cluster can be scaled-out as well as scaled-up, according to the customer’s scaling needs (more capacity, higher performance or both).

Scale-out is implemented by adding X-Bricks to an existing cluster. The addition of an X-Brick to an existing cluster linearly increases its compute power, overall bandwidth and potential capacity, as each X-Brick that is added to the cluster brings with it its own CPUs, RAM and FC/iSCSI ports to service the clients of the environment, together with an additional DAE and SSDs to increase the capacity provided by the cluster. Adding an X-Brick to scale-out an XtremIO cluster is intended for environments that grow in both capacity and performance needs. An XtremIO cluster can start with any number of X-Bricks that fits the environment’s initial needs and can currently grow to up to 4 X-Bricks (for both X2-S and X2-R). Future code upgrades of XtremIO X2 will allow growing to up to 8 X-Bricks for X2-R arrays.

![Figure 10. XtremIO X2 Scale Out Capabilities – Single to Multiple X-Bricks in One Cluster](image)

Scale-up of an XtremIO cluster is implemented by adding SSDs to existing X-Bricks in the cluster. Adding SSDs to scale-up an XtremIO cluster is intended for environments that grow in capacity needs but have no need for extra performance. Any X-Brick can be initially installed with 18 or more SSDs, and can be extended to up to 72 SSDs in increments of 6, according to the environment’s capacity needs.
SSDs are 400GB per drive for X2-S clusters and 1.92TB per drive for X2-R clusters. Future releases will allow customers to populate their X2-R clusters with 3.84TB sized drives, doubling the physical capacity available in their XtremIO clusters.

Inline Data Reduction

XtremIO has tremendous benefits for private cloud environments in terms of storage administration and flexibility. One of its most important features for the cloud is its superb Data Reduction capabilities which allows for a substantially higher logical capacity than the physical flash storage that it uses, and provides storage administrators with a much more flexible approach of provisioning storage to the cloud’s tenants.

XtremIO's unique Data Reduction features are inline and always-on. This means that no configuration is needed to apply them, and that the storage array uses no post-write processes in their implementation, thus maintaining consistent performance at all times without producing redundant writes to the flash media after the data is initially written. This increases SSD endurance and eliminates performance degradation. The features providing XtremIO’s Data Reduction are described in the following sections.

Data Deduplication

XtremIO’s Inline Data Deduplication removes duplicate I/O blocks from a stream of incoming data prior to it being written to the flash media. The deduplication is global, meaning no duplicate blocks are written if they already exist anywhere in the storage cluster.

Deduplication in XtremIO is performed using fingerprinting of blocks. For every data block, the array fingerprints the data with a unique identifier and stores it in the cluster’s mapping table. When a new block is written to the array, the array first checks if the block already exists in the system’s mapping table using its fingerprint. If it is found, it will not be written again. The fingerprints are also used for uniform distribution of data blocks across the array, which provides inherent load balancing for performance and enhances flash wear-level efficiency, since the data never needs to be rewritten or rebalanced.

XtremIO uses a content-aware, globally deduplicated Unified Data Cache for highly efficient data deduplication. The system's unique content-aware storage architecture enables achieving a substantially larger cache size with a small DRAM allocation. Therefore, XtremIO is the ideal solution for difficult data access patterns that are common in private cloud environments.

XtremIO has excellent data deduplication ratios, especially for virtual and private cloud environments. With it, SSD usage is smarter, flash longevity is maximized, the logical storage capacity is multiplied, and total cost of ownership is reduced.
Data Compression

Inline Data Compression is the compression of data prior to it being written to the flash media. XtremIO automatically compresses data after all duplications are removed, ensuring that the compression is performed only on unique data blocks. Compression rates depend on the type of data written.

Data Compression complements data deduplication in many cases, and saves storage capacity by storing only unique data block in the most efficient manner. In a private cloud environment deduplication dramatically reduces the required capacity of recurring services, while compression reduces specific user data. As a result, less physical capacity is required to store data and an increased number of cloud services can be managed by a single X-Brick, increasing the storage array's efficiency and dramatically reducing the $/GB cost of storage, even when compared to hybrid storage systems.

We can see the benefits and capacity savings for the deduplication-compression combination in Figure 12.

In the above example, the twelve data blocks written by the host are first deduplicated to four data blocks, providing a 3:1 data deduplication ratio. Next is the data compression process, as each of the four data blocks are then compressed by an average compression ratio of 2:1, resulting in a total data reduction ratio of 6:1.

Thin Provisioning

XtremIO storage is natively thin provisioned, using a small internal block size. All volumes in the system are thin provisioned, meaning that the system consumes capacity only when it is needed. Storage space is never pre-allocated before writing.

Because of XtremIO's content-aware architecture, blocks can be stored at any location in the system (when the metadata is used to refer to their location) and the data is written only when unique blocks are received. Therefore, as opposed to disk-oriented architecture, no space creeping or garbage collection is necessary on XtremIO, volume fragmentation does not occur in the array and no defragmentation utilities are needed.

XtremIO’s content-aware architecture and Thin Provisioning feature enables consistent performance and data management across the entire life cycle of a volume, regardless of the system capacity utilization or client’s write patterns.

This characteristic allows frequent manual and automatic reclaiming of unused space directly from VMFS datastores and virtual machines, which has the following benefits:

- The allocated disks can be used optimally, and the actual space reports are more accurate.
- Snapshots are more efficient since blocks that are no longer needed are not protected by additional snapshots.

Figure 12. Data Deduplication and Data Compression Demonstrated
Copy Data Services

One of the most usable technologies XtremIO offers is its agile Copy Data Services. XtremIO elevates snapshots and clones beyond simple data protection, as its Copy Data Services allow the instant creation of full-size, full-performance volume copies to use for non-production needs. Whether customers need to create multiple instances of their data for test, development, analytics or other purposes, XtremIO provides a way to efficiently deploy all of the application’s instances on a single array with no loss of performance and minimal space consumption. This ability means no planning in advance of the number of copies a tenant in the cloud would need throughout its lifetime – whenever a new copy of a client’s application is needed, with its entire data, it can be generated quickly with XtremIO with no performance or capacity ramifications on the rest of the cloud. This ability gives application and infrastructure teams breakthrough workflows and business process agility by eliminating time-wasting, performance-sapping, capacity-hungry brute force copies, and provides flexibility to work on more added-value projects and innovations.

Integrated Copy Data Management

XtremIO pioneered the concept of integrated Copy Data Management (iCDM) – the ability to consolidate both primary data and its associated copies on the same scale-out all-flash array for unprecedented agility and efficiency. XtremIO is one of a kind in its consolidation capabilities, providing organizations with a new level of agility and self-service for on-demand procedures, while consistently maintaining delivery of all performance SLAs.

Consolidation of primary data and its copies in the same array has numerous benefits:

- It can make development and testing activities up to 50% faster, creating copies of production code quickly for development and testing purposes, and then refreshing the output back into production for the full cycle of code upgrades in the same array. This dramatically reduces complexity and infrastructure needs, as well as development risks, and increases the quality of the product.
- Production data can be extracted and pushed to all downstream analytics applications on-demand as a simple in-memory operation. Copies of the data are high performance and receive the same SLA as production copies without compromising production SLAs. XtremIO offers this on-demand procedure for both self-service and automated workflows for application and infrastructure teams.
- Operations such as patches, upgrades and tuning tests can be made quickly using copies of production data. Diagnosing problems of applications and databases can be done using these copies, and applying the changes back to production can be done by refreshing copies back. The same goes for testing new technologies and combining them in production environments.
- iCDM can also be used for data protection purposes, as it enables creating many copies for recovery at low point-in-time intervals. Application integration and orchestration policies can be set to auto-manage data protection using different SLAs.

XtremIO Virtual Copies

XtremIO uses its own implementation of snapshots called XtremIO Virtual Copies (XVCs) for all iCDM purposes. XVCs are created by capturing the state of the data in a volume at a particular point in time, and retaining that capture as long as it is needed regardless of the state of the source volume (even when deleted). They allow both read-only and read-write access types, and can be taken either from a source volume or another Virtual Copy.

XtremIO’s Virtual Copy technology is implemented by leveraging the content-aware capabilities of the system that is optimized for SSDs. This allows efficient copy creation that can sustain high performance, while maximizing the media endurance.

When creating a Virtual Copy, the system only generates a pointer to the ancestor metadata of the actual data in the system, making the operation very quick. This operation does not have any impact on the system and does not consume any capacity at the point of creation, unlike traditional snapshots, which may need to reserve space or copy the metadata for each snapshot. Virtual Copies capacity consumption occurs only when changes are made to any copy of the data.

The system supports the creation of Virtual Copies on a single volume or a set of volumes. All Virtual Copies of the volumes in the set are cross-consistent and contain the exact same point-in-time.
Virtual Copy deletions are lightweight and proportional only to the amount of changed blocks between the entities. The system uses its content-aware capabilities to handle copy deletions. Each data block has a counter that indicates the number of references that block has in the system, and when that counter hits zero, the block can be deleted.

XVCs are used for all iCDM purposes and provide volume snapshot capabilities such as:

- Snapshot Sets and Consistency Groups (CG) – for managing snapshots of one or more volumes as a single entity.
- Protection Copies – copies for data protection and recovery purposes (can be created periodically with the Protection Scheduler).
- Repurposing Copies – copies created for alternating purposes such as test, development and analytics (with the ability to refresh a repurpose copy with updated data from the source, or to refresh a production copy with new developed data or code).

**Native Replication**

Native Replication is a new feature in XtremIO X2 (presented at version 6.1) which allows customers to replicate their data between different XtremIO clusters (in potentially separate sites) to protect their data in cases of cluster failures or unavailability of an entire site. XtremIO replication is currently Asynchronous only.

The unique content-based architecture of XtremIO that is discussed in the Inline Data Reduction and Copy Data Services sections is also leveraged for data replication, contributing to a faster data replication with low bandwidth consumption.

As shown in Figure 13, with this implementation, when a new block “D”, whose content already exists in both the source and destination arrays, is written to a new logical address in the system, it will not need to be replicated to the destination array. Similarly, the block is not written again to the source array, due to the system’s in-line deduplication. Only the block’s metadata is written and replicated. Furthermore, the unique blocks that are being written and replicated are saved and sent only after compression.

![Figure 13. XtremIO Native Replication – Only Unique Blocks are Actually Replicated](image)
XtremIO replication is based on a snapshot shipping method. Snapshots at the source are created at a frequency derived from the RPO (Recovery Point Objective) settings, and only the changes between the last 2 snapshot-sets are transferred to the destination, further increasing the replication efficiency. The snapshots are kept in the array for point-in-time (PIT) recoveries based on a Retention Policy. Up to 3 different granularity levels of time-periods and snapshot count can be configured in a Retention Policy to fit the customer's needs. Currently, the maximum snapshot-sets allowed per protection session is 500.

![XtremIO Native Replication Retention Policy](image)

XtremIO supports all failover scenarios including “Planned Migration” (sync and failover), “Failover Test” and “Disaster Recovery” to any point-in-time at the target. The failover step is instantaneous and requires no metadata copy and no log roll-forward or roll-back. All it does is refresh the Volume with the chosen PIT snapshot, which essentially means updating pointers, just like refreshing a Volume from a snapshot on a stand-alone cluster. This contributes to a near-zero Recovery Time Objective (RTO), regardless of the selected PIT to recover from. When performing a failover, the replication direction is reversed (but only initiated after confirmation from the user).

**VMware APIs for Array Integration (VAAI)**

VAAI was first introduced as VMware's improvements to host-based VM cloning. It offloads the workload of cloning a VM to the storage array, making cloning much more efficient. Instead of copying all blocks of a VM from and back to the array for the creation of a new cloned VM, the application lets the array do it internally, utilizing the array's features and saving host and network resources that are no longer involved in the actual cloning of data. This procedure of offloading the operation to the storage array is backed by the XCOPY (extended copy) command to the array, which is used when cloning large amounts of complex data.

XtremIO is fully VAAI compliant, allowing the array to communicate directly with vSphere and providing accelerated storage vMotion, VM provisioning, and thin provisioning functionalities. In addition, XtremIO's VAAI integration improves XCOPY efficiency even further by making the whole operation metadata driven. Due to its inline data reduction features and in-memory metadata, no actual data blocks are copied during an XCOPY command and the system only creates new pointers to existing data, all carried out inside the Storage Controllers' memory. Therefore, in addition to saving host and network resources, the operation also does not consume storage resources, leaving no impact on the system's performance, as opposed to other implementations of VAAI and the XCOPY command.
Figure 15 illustrates the XCOPY operation when performed against an XtremIO storage array, and shows the efficiency in metadata-based cloning.

The XtremIO features for VAAI support include:

- Zero Blocks / Write Same – Used for zeroing-out disk regions and provides accelerated volume formatting.
- Clone Blocks / Full Copy / XCOPY – Used for copying or migrating data within the same physical array, an almost instantaneous operation on XtremIO due to its metadata-driven operations.
- Record Based Locking / Atomic Test & Set (ATS) – Used during creation and the locking of files on a VMFS volumes, as is needed during power-down and power-up of VMs.
- Block Delete / Unmap / Trim – Used for reclamation of unused space using the SCSI UNMAP feature.
Other System Features

We list here additional built-in features that are provided and offered by the XtremIO X2 Storage Array and require no special license. The architecture and implementation of these features is unique to XtremIO and is designed around the capabilities and limitations of flash media.

XtremIO Data Protection

XtremIO Data Protection (XDP) provides a "self-healing" double-parity data protection with very high efficiency to the storage system. It requires very little capacity overhead and metadata space, and does not require dedicated spare drives for rebuilds. Instead, XDP leverages the "hot space" concept, where any free space available in the array can be utilized for failed drive reconstructions. The system always reserves sufficient distributed capacity for performing at least a single drive rebuild. In the rare case of a double SSD failure, the second drive will be rebuilt only if there is enough space to rebuild the second drive as well, or when one of the failed SSDs is replaced.

The XDP algorithm provides:

- N+2 drives protection.
- Capacity overhead of only 5.5%-11% (depends on the number of disks in the protection group).
- 60% more write-efficient than RAID1.
- Superior flash endurance to any RAID algorithm, due to the smaller number of writes and even distribution of data.
- Automatic rebuilds that are faster than traditional RAID algorithms.

As shown in Figure 16, XDP uses a variation of N+2 row and diagonal parity which provides protection from two simultaneous SSD errors. An X-Brick DAE may contain up to 72 SSDs organized in two Data Protection Groups (DPGs). XDP is managed independently on the DPG level. A DPG of 36 SSDs will result in capacity overhead of only 5.5% for its data protection needs.

![Figure 16. N+2 Row and Diagonal Parity](image)

Write Boost

In the new X2 storage array, the write flow algorithm has been improved to significantly enhance the array's performance, smartly handling the rise in compute power and disk speeds and taking into account common applications' I/O patterns and block sizes. The commit of a write operation to the host is now asynchronous to the actual writing of the blocks to the disk, and is sent after the changes are written to the local and remote NVRAMs for protection. The changes are written to the disk only later, at a time that best optimizes the system's activity. In addition to the shortened procedure from write to commit, the new algorithm addresses an issue relevant to many applications and clients: a high percentage of small I/Os creating load on the storage system and influencing latency, especially on bigger I/O blocks.
Examining customers’ applications and I/O patterns, it was found that many I/Os from common applications come in small blocks, under 16K pages, creating high loads on the storage array. Figure 17 shows the block size histogram from the entire XtremIO install base. The percentage of blocks smaller than 16KB is highly evident. The new algorithm solves this issue by aggregating small writes to bigger blocks in the array before writing them to disk. This way, the small writes less demanding on the system, which is now more capable of handling bigger I/Os faster. The test results for the improved algorithm were amazing: The improvement in latency for several cases is around 400% and allows XtremIO X2 to address application requirements with a latency of 0.5 msec or less.

Data at Rest Encryption

Data at Rest Encryption (DARE) provides a solution to securing critical data even when the media is removed from the array, for customers in need of such security. XtremIO arrays utilize a high-performance inline encryption technique to ensure that all data stored on the array is unusable if the SSD media is removed. This prevents unauthorized access in the event of theft or loss during transport, and makes it possible to return/replace failed components containing sensitive data. DARE has been established as a mandatory requirement in several industries, such as health care, banking, and government institutions.

At the heart of XtremIO’s DARE solution lays the use of the Self-Encrypting Drive (SED) technology. An SED has a dedicated hardware which is used to encrypt and decrypt data as it is written to or read from the drive. Offloading the encryption task to the SSDs enables XtremIO to maintain the same software architecture whether encryption is enabled or disabled on the array. All XtremIO’s features and services (including Inline Data Reduction, XtremIO Data Protection, XtremIO Virtual Copies, etc.) are available on an encrypted cluster as well as on a non-encrypted cluster, and performance is not impacted when using encryption.
A unique Data Encryption Key (DEK) is created during the drive manufacturing process, and does not leave the drive at any time. The DEK can be erased or changed, rendering its current data unreadable forever. To ensure that only authorized hosts can access the data on the SED, the DEK is protected by an Authentication Key (AK) that resides on the Storage Controller. Without the AK, the DEK is encrypted and cannot be used to encrypt or decrypt data.

Figure 18. Data at Rest Encryption in XtremIO

Other features of XtremIO X2 include:

- High Availability (no single points of failure)
- Non-Disruptive Upgrade and Expansion
- RecoverPoint integration (for replications to local or remote arrays)

**XtremIO Management Server**

The XtremIO Management Server (XMS) is the component that manages XtremIO clusters. It is pre-installed with the CLI, GUI and RESTful API interfaces, and can be installed on a dedicated physical server or a VMware virtual machine. A single XMS can manage up to 8 XtremIO clusters.

The XMS manages the cluster through the management ports on the Storage Controllers of the first X-Brick in the cluster, and uses a standard TCP/IP connection to communicate with them. It is not part of the XtremIO data path, thus can be disconnected from an XtremIO cluster without jeopardizing usual I/O tasks. A failure on the XMS only affects monitoring and configuration activities, such as creating and attaching volumes. A virtual XMS is naturally less vulnerable to such failures.
The GUI is based on a new Web User Interface (WebUI), which is accessible via any browser, and provides easy-to-use tools for performing most system operations (certain management operations must be performed using the CLI). The WebUI contains the following sections:

- **Dashboard** – A section which presents main overview pages of the managed cluster which include: Health, Performance (shown in Figure 19) and Capacity (shown in Figure 20).

- **Notifications** – A section which shows Events, Alerts and Advisories regarding the system’s health and operations.

- **Configuration** – A section in which the logical entities in the storage system can be viewed and configured. Entities include: Volumes, Consistency Groups, Snapshot Sets, Initiator Groups and Initiators.

- **Data Protection** – A section where you can configure data protection settings using XtremIO replication. Tabs here are: Overview, Protected Entities, Sessions, Retention Policies and Paired Clusters.

- **Reports** (shown in Figure 21) – A section showing graphs and statistics of different aspects of the system’s activities, mainly related to the system’s performance and resource utilization. The different reports are divided into the following categories:
  - **Performance** – Including Overview, Performance, Latency (shown in Figure 22), Cumulative IOs and BW, Block Size and Remote Protection BW reports
  - **Health** – Including CPU Utilization, SSD Balance and Endurance reports
  - **Capacity** – Including Savings, Capacity and Remote Protection Efficiency reports
  - **User Defined** – A section where you can create your own custom reports

- **Hardware** – A section providing a cluster overview, and showing X-Bricks visual illustrations and hardware information and connectivity – Front (shown in Figure 23) and Back.

- **Inventory** – A section presenting the storage environment components including detailed information. The components shown include: XMS, Clusters, X-Bricks, Storage Controllers, Local Disks, Storage Controller PSUs, XEnvs, Data Protection Groups, SSDs, DAEs, DAE Controllers, DAE PSUs, DAE Row Controllers, InfiniBand Switches and NVRAMs.

![Figure 19. XtremIO WebUI – Dashboard – Performance Panel](image-url)
Figure 20. XtremIO WebUI – Dashboard – Capacity Panel

Figure 21. XtremIO WebUI – Reports
Figure 22. XtremIO WebUI – Reports – Latency Window

Figure 23. XtremIO WebUI – Hardware – Front Panel
As mentioned, other interfaces are available for users to monitor and manage their XtremIO clusters through the XMS server, including:

- **Command Line Interface (CLI)** – Can be used for everything the GUI is used for and more.
- **RESTful API** – Pre-installed in the system and allows HTTP-based commands to manage the clusters.
- **PowerShell API Module** – Can be used to administer XtremIO clusters for those who prefer the Windows PowerShell console.
XtremIO Plugin for VMware vRealize Orchestrator

The XtremIO Plugin for vRealize Orchestrator facilitates the automation and orchestration of tasks that involve the XtremIO Storage Array, by providing vRO with access to XtremIO-specific management workflows.

The plugin provides rich built-in Actions and Workflows that can be used individually by vRealize users, or can be utilized to design and deploy custom workflows which automate more complex IT tasks involving XtremIO with VMs, databases, backups or other desired infrastructures that integrate with vRealize Orchestrator.

The plugin provides the following benefits to vRO customers:

- **Agility** – accelerated deployment of storage infrastructure resources through automated workflows.
- **Simplicity** – reduced complexity of datastore provisioning, VM backup and restore, and other storage-related procedures.
- **Efficiency** – improved virtual data center operations with seamless access to XtremIO Storage Array-based services.

To use the built-in Actions and Workflows in the plugin, the XtremIO Plugin must be imported to the vRO instance. See the Dell EMC XtremIO Plugin for VMware vRealize Orchestrator 2.0.0 Installation and Configuration Guide[^7] for instructions on how to install the plugin. A link to download the plugin can be found in the References chapter[^9].

[^7]: Dell EMC XtremIO Plugin for VMware vRealize Orchestrator 2.0.0 Installation and Configuration Guide
[^9]: References chapter
Built-in Actions

The XtremIO plugin for vRealize Orchestrator adds a rich variety of Actions to vRO. These Actions are used by the built-in Workflows imported by the plugin, and can also be used by customers to design their own workflows for their environments. The imported Actions can be found under the `com.emc.XtremIO.util` namespace in the Actions section in the Design area (see Figure 25). There are well over a hundred Actions added by the plugin for use by vRO customers who want to design their own workflows.

![Figure 25. XtremIO X2 Actions for VMware vRealize Orchestrator](vmware_vRealize_Orchestrator.png)

Built-in Workflows

The plugin provides basic and high-level Workflows. Basic Workflows allow the management of specific objects of XtremIO functionality, such as Consistency Groups, Initiator Groups, Protection Schedulers, Snapshot Sets or Volumes. An example of a basic Workflow can be a function such as creating or mapping a Volume. High-level Workflows combine basic XtremIO and VMware operations to achieve a higher level of functionality. For example, a high-level Workflow can be a procedure that creates an XtremIO Volume, creates an Initiator Group for several ESX hosts, and maps the Volume to that Initiator Group. A well-designed high-level Workflow would request any input necessary for the Workflow to run prior to the execution of the entire sequence of operations.

There are about 120 Workflows in the XtremIO Plugin for vRO that can be ran directly or used to design new Workflows. When the XtremIO package is imported to the vRO a new folder will be created under the “Library” folder in the Workflows section called **XtremIO**, inside which there are a set of folders dividing the imported XtremIO Workflows to sub-categories. There are several folders containing basic Workflows of different object types (Consistency Groups, Snapshot Sets, etc.) and a couple of folders containing high-level Workflows. The added folders are listed in Table 4.
<table>
<thead>
<tr>
<th>FOLDER</th>
<th>WORKFLOWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>XtremIO CG Management</td>
<td>Workflows performing basic operations on XtremIO Consistency Groups.</td>
</tr>
<tr>
<td>XtremIO Cluster Management</td>
<td>Workflows performing basic operations on entire XtremIO Clusters.</td>
</tr>
<tr>
<td>XtremIO IG Management</td>
<td>Workflows performing basic operations on XtremIO Initiator Groups.</td>
</tr>
<tr>
<td>XtremIO Performance Management</td>
<td>Workflows producing XtremIO Performance reports.</td>
</tr>
<tr>
<td>XtremIO Protection Scheduler</td>
<td>Workflows performing basic operations on XtremIO Protection Schedulers.</td>
</tr>
<tr>
<td>XtremIO RecoverPoint Management</td>
<td>Workflows performing basic operations on RecoverPoint entities interfacing with XtremIO storage.</td>
</tr>
<tr>
<td>XtremIO Snapshot Set Management</td>
<td>Workflows performing basic operations on XtremIO Snapshot Sets.</td>
</tr>
<tr>
<td>XtremIO Storage Management</td>
<td>Workflows performing high-level operations on Datastore and VMs residing on XtremIO storage.</td>
</tr>
<tr>
<td>XtremIO Tag Management</td>
<td>Workflows performing basic operations on XtremIO Tags.</td>
</tr>
<tr>
<td>XtremIO VMware Storage Management</td>
<td>Workflows performing high-level operations on VMware entities (Datastores, Hosts, vCenters and VMs) residing on XtremIO storage.</td>
</tr>
<tr>
<td>XtremIO Volume Management</td>
<td>Workflows performing basic operations on XtremIO Volumes.</td>
</tr>
<tr>
<td>XtremIO XMS Management</td>
<td>Workflows performing basic operations on XtremIO Management Servers.</td>
</tr>
</tbody>
</table>

The full list of XtremIO Workflows added by the vRO plugin is provided in Appendix A – Built-in Workflows by the XtremIO Plugin for vRO and in the Dell EMC XtremIO Plugin for VMware vRealize Orchestrator 2.0.0 Workflows User Guide.  

![Figure 26. XtremIO X2 Workflows for VMware vRealize Orchestrator](image)

34 | Implementing a Private Cloud with VMware vRealize and Dell EMC XtremIO X2
© 2018 Dell Inc. or its subsidiaries.
The next two sections (Building a Custom Action and Designing a New Workflow) are for the more experienced users, interested in developing their own procedures in vRO for orchestration purposes.

**Building a Custom Action**

VMware vRealize Orchestrator provides users with the ability to build their own Actions, which they would later use in new Workflows they want to design. A new custom Action should be built when a certain function will be used more than once in our Workflows. It should usually be a discrete and simple operation to be used on an infrastructure that is integrated with the vRO. For example, in the `XtremIO.util` module, we can find the `tagAnObject` Action, which gives a specific Tag to a requested object in the storage system (Volume, Initiator, etc.), or the `getCGforVolume` which receives a Volume name and returns the Consistency Groups in which it is located.

**Creating a New Module for Custom Actions**

To build a new Action a user must have “Design” permissions. Building Actions and designing Workflows are performed via the Design area, selected in the top left scroll-down list of the vRO interface.

First create a Module to store the new Actions. Right click on the parent entity (user @ vRO_instance) and select the New module… option. We will name the new Module “com.emc.XtremIO.custom”:

![Creating a New Module for Custom Actions](image)
Building an Action from Scratch

There are two main methods for creating new Actions. The first one is creating a new action from scratch using the **Add action…** menu option in the new Module, and start scripting. In our example, we are creating an Action named "getVMbackingDatastoresExceptHardDisk1":

![Creating a New Action](image)

After creating a new Action, the vRO interface will automatically open a new window to start scripting our Action. We will first set our inputs and output, and then write the code.
To set the inputs, click the Add parameter button (1) in the Scripting tab and set the parameter Name (2), Type (3) and Description (4). We can use the “Filter” text box in step (3) to find the desired type for the parameter, which in our case is VC:VirtualMachine. We can set multiple parameters for an Action, but our example requires only one:

![Figure 29. Setting an Input Parameter for a Custom Action](image)

To set the output, click the Return type item (set to void by default) and select the type to be returned. In our case, we want to return an array of Datastores, so we select Array of instead of Type in the Select a type... box, and choose the VC:Datastore type:

![Figure 30. Setting the Output Type for a Custom Action](image)

The next step is to write the code for the Action. vRO is based on javascript as its scripting language, so one must know basic javascript to develop code in vRO, or review code used by other existing Actions. To help users develop their code, vRO offers assistance as it presents object hierarchies and possible variables, functions and existing Actions to use in the code in the left pane of the Scripting tab.
After learning the hierarchy of the VC:VirtualMachine object and testing some of the code, we completed writing our Action and click the **Save and close** button to move to our next task. The Action’s input, output and code are all presented in Figure 31.

**Figure 31. New “getVMbackingDatastoresExceptHardDisk1” Custom Action**

### Duplicating an Action

Another option for creating a new Action is to modify an existing Action. Many packages (built-in or available on the web) containing thousands of Actions exist for vRO and contain code that may be similar to what we want to create ourselves. We can use that existing code as a base for our own Actions.
To find an appropriate Action to be used as the basis for our purpose, we can view an Action’s **Description** or browse through its code for comments on what it does. We can also use the **Search for** tool in the upper right corner of the vRO interface to search for specific terms or text that might help us find what we are looking for:

![Search for an Action](image)

**Figure 32. Searching for an Action**

To duplicate an existing Action, select the **Duplicate action...** menu option when right clicking the Action we wish to use. Then choose a name for the new Action and the Module we wish to store it in and click **Submit**:

![Duplicating an Action](image)

**Figure 33. Duplicating an Action**
We are interested in creating an Action that receives an NAA name (of some XtremIO Volume) as an input and returns the XMS that manages that Volume. To do this, we can duplicate the `getVolumeNamesForNaaNames` Action (which is available in the `com.emc.XtremIO.util` Action Module) that accepts an array of NAA names of XtremIO Volumes and returns a json string that contains information about them. We will duplicate that Action to a new Action called `getXMSForNaaName`, place it in our `com.emc.XtremIO.custom` Action Module and modify its input and code to match our requirements, just as we edited the Action that we created from scratch.

**Custom XtremIO Sample Actions**

In the next step, we created a few more Actions for the Workflows we want to design. All of the Actions created are summarized in Table 5 and shown in Figure 34.

<table>
<thead>
<tr>
<th>ACTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getDatastoreVolumesNaaNames</code></td>
<td>Gets a Datastore and returns its underlying Volumes' NAA names.</td>
</tr>
<tr>
<td><code>getVMbackingDatastoresExceptHardDisk1</code></td>
<td>Gets a VM and returns the backing Datastores of its disks unrelated to the operating system disk.</td>
</tr>
<tr>
<td><code>getVolumeNameForNaaName</code></td>
<td>Gets an NAA name of an XtremIO Volume and returns the XtremIO Volume name.</td>
</tr>
<tr>
<td><code>getXMSForNaaName</code></td>
<td>Gets an NAA name of an XtremIO Volume and returns the XMS that manages it.</td>
</tr>
<tr>
<td><code>getXtremIOClusterForNaaName</code></td>
<td>Gets an NAA name of an XtremIO Volume and returns the XtremIO Cluster that holds it.</td>
</tr>
</tbody>
</table>

Figure 34. Custom XtremIO Actions

**Designing a New Workflow**

VMware vRO equips users with an editor to design new Workflows that would help them run recurring internal IT procedures quickly and easily. New Workflows should be designed for complex IT procedures that involve several operations (Actions, other Workflows, user interactions, scriptable objects, etc.) and often even several infrastructures, and would make running them via a single Workflow much more effective.
Adding a Folder for New Workflow

To create new Workflows for an environment, start by adding a new folder in the Workflow section. To do that, right click the parent entity \textit{(user @ vRO\_instance)} and select the \texttt{Add folder...} option. We will name the new folder \textit{"Custom XtremIO Workflows"}.

![Adding a New Folder for Custom Workflows](image)

Figure 35. Adding a New Folder for Custom Workflows
Duplicating a Workflow

As with Actions, we can duplicate existing Workflows that perform procedures similar to what we desire and modify them to fit our needs. The process is the same – right click a Workflow and select Duplicate workflow..., choose a name and a folder and click Submit:

![Duplicating a Workflow](image)

In our example, we are not duplicating Workflows, but creating them from scratch.
Designing a Workflow from Scratch

To create a new Workflow, right-click the folder in which you want it to be located and select **New workflow**:

![Creating a New Workflow](image)

After providing a name, the design interface will automatically open to allow us to start designing our Workflow.

The vRO window for designing Workflows contains several important tabs, which will be explained later in this section. We will start with the **Schema** tab, which is the main tab where we design the flow of our Workflow procedure.

**Figure 38** shows the **Schema** tab in its initial state. Each Workflow has a beginning point and an ending point (sometimes multiple potential ending points). To this we begin adding elements that will construct our Workflow.

![Workflow Design Schema Tab Starting Point](image)
We want to design a Workflow that would replicate a "Production" environment to a new "Test/Dev" environment. Starting with a "Production" VM folder in our VC that contains "Production" VMs with primary storage (operating system disks) and secondary storage (application disks), the new workflow will do the following:

1. Create a new "Test/Dev" VM folder.
2. Clone the operating systems of each "Production" VM to create new "Test/Dev" VMs in the new folder.
3. Copy the secondary storage of each VM in the "Production" folder to the new VMs in the "Test/Dev" folder.

This workflow can be used to replicate a "Production" environment with all of its current application data to a new "Test/Dev" environment for any purpose (e.g. new environment for developers, analytics, patch testing, etc.). Consolidating production data and its copies on the same infrastructure is one of XtremIO’s top benefits for customers (see the Copy Data Services section), as it has the ability to copy entire Volumes quickly with minimal space consumption and no performance degradation for production copies. We are aiming to utilize this XtremIO unique feature in our Workflow.

**Adding Elements to a Workflow**

We start with adding the Create virtual machine folder Workflow that will create the new "Test/Dev" VM folder. To add a Workflow element to the schema, we can either select it in the All Workflows list on the left, or add a generic Workflow element from the Generic list and search for the Workflow we want to use. To add an element, we need to drag it from one of the lists on the left into the schema:

![Adding a Workflow to the Schema](image.png)
Binding an Element’s Inputs and Outputs

The *Create virtual machine folder* Workflow, much like any other Workflow or Action, has inputs and outputs. Each input of a Workflow or Action element needs to be bound to some variable or value. Non-mandatory inputs can be assigned a *null* value. Outputs can be left unbound, depending on whether or not they will be used next. The bound variables can be either Inputs or Outputs of the super-Workflow (the one we are designing) or general Attributes, which are the local parameters of the Workflow, and can either be fixed or calculated using other variables in the Workflow.

When highlighting the element we want to examine, we can see its inputs and outputs together with their description in the IN and OUT tabs in the lower panel of the window:
We have several ways to bind the inputs and outputs of an element we are using in our Workflow. One way is by using the **Setup...** button that appears at the top right corner of the schema pane when highlighting an element whose inputs and outputs we want to bind:

![Setup button](image)

Clicking the **Setup...** button will open up the **Promote Workflow Input/Output Parameters** window in which we can create and bind new parameters (or just bind existing parameters) to the inputs and outputs of the Workflow element. In our case, as shown in Figure 42 we chose to create new input parameters named “DestinationParentVMFolder” and “NewEnvironmentName” and bind them to the Workflow element’s inputs, and a new local variable named “DestinationVMFolder” and bind it to the Workflow element’s single output. We are binding the Workflow element’s output to a local variable because we want to use it later in other tasks of our Workflow.

![Promote Workflow Input/Output Parameters](image)
After clicking **Promote** we can see in the **Visual Binding** tab below the schema that the Workflow element's inputs and output are bound to the variables we configured:

With this we have completed the first step in designing our new Workflow! We continue adding elements to the Workflow in the same way we did with the first element to complete all the functionalities the Workflow will perform. We are using **Workflow elements**, **Action elements** and **Scriptable tasks**. Scriptable tasks are javascript codes we develop to be used once in the new Workflow. Otherwise, it would be better to create Actions for them.

**“Create Test/Dev Environment” Sample Workflow**

*Figure 44* shows the complete Workflow schema, and *Table 6* lists its elements and their purpose in the flow.
### Table 6. “Create Test/Dev Environment” Workflow Element List

<table>
<thead>
<tr>
<th>ELEMENT NAME</th>
<th>ELEMENT TYPE</th>
<th>ELEMENT SOURCE</th>
<th>ELEMENT PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create virtual machine folder</td>
<td>Workflow</td>
<td>vRO base package</td>
<td>Creates a new Virtual Machine folder in which to store the VMs of the new “Test/Dev” environment.</td>
</tr>
<tr>
<td>getAllVirtualMachinesByFolder</td>
<td>Action</td>
<td>vRO base package</td>
<td>Provides the “Production” environment VMs with the information related to the contents of the environment to be replicated.</td>
</tr>
<tr>
<td>Prepare new VMs info</td>
<td>Scriptable task</td>
<td>Self-developed</td>
<td>Prepares the inputs for the Clone virtual machine, no customization Foreach Workflow.</td>
</tr>
<tr>
<td>Clone virtual machine, no customization</td>
<td>Foreach Workflow</td>
<td>vRO base package</td>
<td>Clones operating system templates to create the VMs of the “Test/Dev” environment according to the operating systems of the VMs in the “Production” environment.</td>
</tr>
<tr>
<td>Prepare VM Clone Storage info</td>
<td>Scriptable task</td>
<td>Self-developed</td>
<td>Prepares the inputs for the VM Clone Storage Workflow.</td>
</tr>
<tr>
<td>VM Clone Storage</td>
<td>Workflow</td>
<td>XtremIO Plugin</td>
<td>Clones the secondary storage of the VMs in a specified “Production” VM folder to VMs in a “Test/Dev” VM folder (the VM names’ prefixes in the “Test Dev” folder must match the VM names in the “Production” folder).</td>
</tr>
<tr>
<td>Rename Datastores and prepare Volume Rename info</td>
<td>Scriptable task</td>
<td>Self-developed</td>
<td>Renames the Datastore copies of the “Test/Dev” environment to meaningful names and prepares the inputs for the Volume Rename Foreach Workflow.</td>
</tr>
<tr>
<td>Volume Rename</td>
<td>Foreach Workflow</td>
<td>XtremIO Workflow package</td>
<td>Renames the XtremIO Volume copies of the “Test/Dev” environment to meaningful names.</td>
</tr>
<tr>
<td>Relay Output</td>
<td>Scriptable task</td>
<td>Self-developed</td>
<td>Relays the output of the Workflow.</td>
</tr>
</tbody>
</table>
Foreach Workflow Elements

In the Workflow we designed we are using **Foreach Workflow elements**. A Foreach Workflow element is usable when we wish to run a workflow multiple times with several iterations, sending different variables for each iteration. Let’s take a look at how we set the inputs and outputs of a Foreach element:

![Foreach Workflow diagram](image_url)

*Figure 45. Foreach Clone virtual machine, no customization Workflow Element Input Parameters*
We are passing 3 parameters as the **Array(s) to be traversed** for the Foreach **Clone virtual machine, no customization** Workflow element. Each iteration of the Workflow will use the next value in each array, while the other parameters remain the same through each iteration. The three arrays that are traversed are bound to: **vm(s)** – the source VMs of the clone operations (in our case it is the templates used to create the new VMs); **name(s)** – the names for the new VMs; **host(s)** – the destination ESX Hosts for the new VMs.

![Image: Foreach Clone virtual machine, no customization Workflow Element Output Parameters](image)

**Figure 46.** Foreach **Clone virtual machine, no customization** Workflow Element Output Parameters

The output of the original **Clone virtual machine, no customization** Workflow is a single Virtual Machine, but since we are running several iterations, the output of the entire Foreach Workflow element is an array of Virtual Machines, one for each iteration of the Workflow. Output is forwarded to the **CreatedVMs** parameter.
See Figure 47 for the Visual Binding of the Foreach Clone virtual machine, no customization Workflow element:

![Figure 47. Foreach Clone virtual machine, no customization Workflow Element Visual Binding](image)

The second Foreach Workflow element of our Workflow is the XtremIO’s imported Workflow Volume Rename. Two parameters are passed as Array(s) to this Foreach element: volName(s) – the names of the Volumes we want to rename; newName(s) – new names for the Volumes. The output of this Foreach element is an array with the new Volumes’ names.
Using (Custom) Actions in the Workflow

As described above, we created a few custom Actions which we said will be used in our Workflow. Those Actions are all used in the Prepare VM Clone Storage info Scriptable task shown in Figure 48.

```javascript
// get Source and Destination VM Folders Path
SourceVMFolderPath = System.getModule("com.emc.XtremIO.util").getFolderPath(SourceVMFolder);
DestinationVMFolderPath = System.getModule("com.emc.XtremIO.util").getFolderPath(DestinationVMFolder);

SourceVolumeNameList = new Array();
var SourceDatastoreIDList = new Array();
SourceDatastoreList = new Array();
var NAANames = null;

// for every VM in the Source
for (var i in SourceVMs)
{
  var vm = SourceVMs[i];
  System.debug("For VM: " + vm.name);
  // get the backing Datastore of every secondary storage
  var datastores = System.getModule("com.emc.XtremIO.custom").getVMbackingDatastoresExceptHardDisk(vm);
  // for every such Datastore
  for (var j in datastores)
  {
    var datastore = datastores[j];
    System.debug("datastore: " + datastore.name);
    var datastoreID = datastore.id;
    // check if it was already added to the list of Datastores to be copied
    if ( SourceDatastoreIDList.indexOf(datastoreID) == -1 )
    {
      System.debug("The datastore is not yet in the array and will be added");
      // add it to the list of Datastores to be copied
      SourceDatastoreIDList.push(datastoreID);
      SourceDatastoreList.push(datastore);
      // get NAA Names of all the Volumes comprising the Datastore
      NAANames = System.getModule("com.emc.XtremIO.custom").getVolumeNAANames(datastore);
      // for every NAA Name
      for (var k in NAANames)
      {
        System.debug("The datastore contains NAA name: " + NAANames[k]);
        // find the XtremIO Volume of that NAA
        var volumeName = System.getModule("com.emc.XtremIO.custom").getVolumeNameForNaaName(NAANames[k]);
        System.debug("The next volume matches the NAA and will be added to the list: " + volumeName);
        // add that Volume to the list of Volumes to be copied
        SourceVolumeNameList.push(volumeName);
      }
    }
  }
}

// get XMS Server and XtremIO-Cluster of the Volumes
XMSServer = System.getModule("com.emc.XtremIO.custom").getXMSForNaaName(NAANames[0]);
clusterName = System.getModule("com.emc.XtremIO.custom").getXtremIOClusterForName(NAANames[0]);

// get the ESX Cluster of the Source Environment
primaryCluster = System.getModule("com.vmware.library.vc.cluster").getResourceCtxtVm(SourceVMs[0]);
```

Figure 48. The Prepare VM Clone Storage info Scriptable Task

All of the new Actions we have created (listed in Table 5) are in use in this Scriptable task.

We can see that it is possible to use Actions as discrete elements in a Workflow or in code using the `System.getModule` vRO function, specifying the Action we want to use and the Module in which it is located.
vRO Design Interface Overview

We will now have a look at other important tabs in the Workflow design window at the vRO interface.

The first tab is the **General** tab, in which we can modify the Name, Version and Description of our Workflow, along with other configurations. We can also configure and modify the Workflow’s Attributes, which are the local variables in use during its execution.

![Create Test/Dev Environment General Tab](image-url)

Figure 49.  *Create Test/Dev Environment General Tab*
Next, we have the **Inputs** and **Outputs** tabs, where we can configure and modify the Workflow’s inputs and outputs.

![Create Test/Dev Environment Inputs Tab](image1)

![Create Test/Dev Environment Outputs Tab](image2)

We are already familiar with the **Schema** tab, so we move forward to the **Presentation** tab. This tab is where we can configure what will be presented to the user when trying to execute the Workflow. It is also where we can configure the inputs for the execution. We do that by adding **Properties** to the input parameters and by adding **Steps** to the Presentation itself.

By adding **Steps** to the presentation, we can divide the inputs required by the user to phases and determine which Step will be presented in which phase. Two Steps are used in our Workflow – the **Source Environment** Step and the **New Test/Dev Environment** Step. With this, we divide the input presentation to “Source Environment” inputs and “New Test/Dev Environment” inputs.
Let’s take a look at the VC input parameter’s presentation, shown in Figure 52, to learn about some of the Properties we can give to inputs.

In our example, we have 3 properties for the VC parameter:

1. Mandatory input – Yes
2. Default value – GetAction("com.emc.XtremIO.util","onevCenterServer").call()
3. Predefined list of elements – GetAction("com.emc.XtremIO.util","findvCenterServers").call()

The Mandatory input property defines whether or not this parameter is required for the execution of the Workflow; the Default value property sets what will be the default value of the parameter; and the Predefined list of elements property sets a list of values from which the user will need to choose the value of that input. We can see that for the last two properties we are using existing Actions – the first returns a single value, which will be the default value of the input, and the second returns an array, which will consist of a list of elements from which the user will choose the desired input.
We can add other properties by clicking the **Add property...** button and choosing and configuring a new property.

**Figure 53. Create Test/Dev Environment Presentation Tab – VC Parameter**

Possible properties include:

- **Show parameter input** and **Hide parameter input** properties – Defining when an input parameter will be shown to the user executing the workflow.
- **Data binding** property – Setting a parameter as a function of another parameter or value.
- **Custom validation** property – Setting up rules for valid values to be entered by the user for that input parameter.
Next we have the **Parameters References** tab, in which we can see all the variables of the Workflow (Inputs, Outputs and Attributes) and find where they are used and how.

![Create Test/Dev Environment Presentation Tab – VC Parameter](image)

For instance, we can see that the `XMSServer` Attribute is an output in the *Prepare VM Clone Storage info* Scriptable task, and is an input in the *VM Clone Storage* Workflow element and the *Volume Rename* Foreach Workflow element.

The other tabs of the Workflow design interface are less relevant for the designing of the Workflow, but have other important information: the **Workflow Tokens** tab – lists every execution instance of the Workflow, including execution time, whether it succeeded or not (and if not, where did it fail) and by whom it was run; the **Events** tab – details the history of the Workflow’s versions – when was it created/imported, modified, etc.; the **Permissions** tab – where we can define the permissions for the Workflow (who can change it, view it, execute it, etc.).

For more information on designing Workflows, see the vRealize Orchestrator Documentation in VMware Docs\(^{11}\) and other related vRO pages.
“Delete Test/Dev Environment” Sample Workflow

In addition to the Create Test/Dev Environment Workflow, we have also designed the Delete Test/Dev Environment Workflow, which is meant for erasing an environment set up by the Create Test/Dev Environment Workflow. See Figure 55 for a complete schema of the Delete Test/Dev Environment Workflow and Table 7 for a list of its elements and their purpose in the flow.

**Table 7. “Delete Test/Dev Environment” Workflow Element List**

<table>
<thead>
<tr>
<th>ELEMENT NAME</th>
<th>ELEMENT TYPE</th>
<th>ELEMENT SOURCE</th>
<th>ELEMENT PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAllVirtualMachinesByFolder</td>
<td>Action</td>
<td>vRO base package</td>
<td>Provides the “Test/Dev” environment VMs with the information related to the contents of the environment to be deleted.</td>
</tr>
<tr>
<td>Prepare Datastore Delete</td>
<td>Scriptable task</td>
<td>Self-developed</td>
<td>Prepares the inputs for the Datastore Delete Storage Foreach Workflow.</td>
</tr>
<tr>
<td>Storage info</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datastore Delete</td>
<td>Foreach Workflow</td>
<td>XtremIO Plugin</td>
<td>Deletes the Datastores of the secondary storage of all the “Test/Dev” VMs (after deleting the VMs themselves and unmounting the Datastores) and deletes the underlying XtremIO Volumes of those Datastores.</td>
</tr>
<tr>
<td>virtual machine folder</td>
<td>Workflow</td>
<td>vRO base package</td>
<td>Deletes the Virtual Machine folder that was created for the “Test/Dev” environment being deleted.</td>
</tr>
<tr>
<td>Relay Output</td>
<td>Scriptable task</td>
<td>Self-developed</td>
<td>Relays the output of the Workflow.</td>
</tr>
</tbody>
</table>

*These Workflow examples were designed without exception or error handling. They were designed simply to show how a procedure which will be run repeatedly can be orchestrated in vRO. When designing your own Workflows for production environments, including error handling is recommended.*
Executing Workflows

Now for the essence of vRO – executing the Workflows. There are a few ways to run vRO Workflows which will be reviewed in this section.

For the purpose of this example, we deployed a VMware Datacenter with the following VM folders:

1. **Management Servers** – Holds all Virtual Machines needed for managing the cloud infrastructure (which include the VC Server, the vRO Server, etc.). The “Management Servers” are hosted on a designated ESX Cluster named **PROD-MGMT** and reside on a separate Datastore.

2. **Production** – Holds our environment’s “Production” VMs. We deployed 3 VMs here, named **App_Server**, **Database** and **Web_Server**. The “Production” environment is hosted on a designated ESX Cluster named **PROD-SERVERS** and the VMs’ operating system disks reside on the **OS_Datastore**.

3. **Templates** – Holds the operating system templates used in our environment. We created two templates – **Windows-2016** and **CentOS-7.3**. The templates reside on the **OS_Datastore**.

4. **Test-Dev** – Will hold our environment’s “Test/Dev” VMs. The “Test/Dev” environment will be hosted on a designated ESX Cluster named **TEST-DEV-SERVERS** and its VM operating system disks will reside on the **OS_Datastore**.

The deployment can be viewed in Figure 56.

![Figure 56. Our VMware Deployment](image)

Figure 56. Our VMware Deployment
Executing Workflows via the vRO

Executing Workflows is quite an easy task once we have a comprehensible Workflow.

To run a Workflow via the vRO, simply navigate to the Workflow you want to run in the Workflows section, right click on it and select Start Workflow. Next, enter all required information for the Workflow and click Submit:

Figure 57. Running a Workflow via the vRO
We are running the *Datastore Expose Storage* Workflow that is a part of the XtremIO Workflow Package for vRO, to provision storage in our “Production” environment.

![Running the Datastore Expose Storage Workflow from the vRO Interface – Input Data Sources Step](image)

![Running the Datastore Expose Storage Workflow from the vRO Interface – Volume Information Step](image)
During the Workflow’s execution, we can monitor the tasks as they are running, and the logs generated in the vRO interface.
After the Workflow is completed successfully, we will see one of the ending points of the Workflow marked green (✔) and the ✔ sign next to the matching task on the right.

If the Workflow was not successful, we will see one of the “error” signs marked red (❗) and the ❗ sign next to the matching task.

Figure 62. Running the *Datastore Expose Storage* Workflow – A Successful Execution

If the Workflow was not successful, we will see one of the “error” signs marked red (❗) and the ❗ sign next to the matching task.
We can see the new Datastore in our vSphere Web Client and its underlying XtremIO Volume in the XMS. We can identify the relationship between Datastore and XtremIO Volume using the NAA Identifier, marked in the figures below.

Figure 63. A New Datastore Created Using the *Datastore Expose Storage* Workflow

Figure 64. A New XtremIO Volume Created Using the *Datastore Expose Storage* Workflow
Next, we will run the VM Add VMDK Workflow to add disks to our “Production” VMs from the Datastore that we just provisioned. We will show an example for the App_Server VM and do the same for the Database and Web_Server VMs.

Figure 65. Running the VM Add VMDK Workflow from the vRO Interface
After running the Workflow, we can see that our App_Server now has Hard Disk 2 provisioned from the new Production_Datastore.

Figure 66. A New Hard Disk Provisioned to a VM Using the VM Add VMDK Workflow

We added disks of different sizes to our Database and Web_Server VMs using the same Workflow.
Executing Workflows via the vSphere Web Client

vRO Workflows can also be run via the vSphere Client. In order to run Workflows through the vSphere Client, we must first register vRO as an extension of our vCenter Server. For more information on how this may be done, see Appendix B – vRO Initial Configuration in the Registering a vRO as an Extension to your vCenter Server subsection.

Adding Workflows to vSphere Client Menus

After the vRO instance has been registered as an extension of the vCenter, Workflows can be added to the vSphere menus via the vRealize Orchestrator menu.
In the vRealize Orchestrator menu, under vRO Home, go to Manage → Context Actions to see the Workflows configured for the vSphere Client (some Workflows exist there by default). To add a Workflow to the vSphere menus, click the Add (+) button. In the Add window, choose the Workflows you want to add to the vSphere Client from your vRO instance on the left pane (1), click the Add button for the Workflows you want to add (2), and select the object types you want to associate with the Workflows (3). In our case, we are adding our two new designed Workflows – Create Test/Dev Environment and Delete Test/Dev Environment and associating them with the Folder object.

Figure 68. Adding Workflows to the vSphere Client Menus

After adding the Workflows to the menus, be sure to logout from the vSphere Web Client and login again for the changes to take effect.
Executing Workflows via the vSphere Client

After adding and associating Workflows to vSphere objects, they can be run by simply right-clicking one of those objects via the vSphere Client.

![Figure 69. Running a Workflow via the vSphere Client](image-url)
The figures below show how we run the *Create Test/Dev Environment* Workflow designed.

Figure 70. Running the *Create Test/Dev Environment* Workflow from the vSphere Client – *Source Environment* Step

Figure 71. Running the *Create Test/Dev Environment* Workflow from the vSphere Client – *New Test/Dev Environment* Step
The Workflow has two steps:

- **Source Environment** step – receives the vCenter Server instance and a source “Production” VM Folder as inputs.
- **New Test/Dev Environment** step – receives the parent VM Folder for the destination environment, the destination ESX Cluster for the new environment’s VMs, and a name for the new environment as inputs.

After entering all the details, we click **Finish** for the Workflow to start running.

While the Workflow is running, we can open the vRO interface and monitor the Workflow execution and the logs created just as was done when running the Workflow through the vRO.

Next, we will review the results of the Workflow just run in the vSphere Client and XMS:

We created an environment named **Analytics** that is a replica of our “Production” environment. We can see the new VMs in a new VM Folder under the **Test-Dev** Folder. We can also see that those VMs are hosted on the **TEST-DEV SERVERS** ESX Cluster. For each VM we will see that its operating system disk (Hard disk 1) resides under the **OS_Datastore** and that its application data (Hard disk 2) resides under **Production_Datastore.Analytics** Datastore that was created by the Workflow. We will also review our new Datastore and identify its XtremIO Volume using the NAA Identifier.

![Figure 72. Create Test/Dev Environment Workflow Results – A New VM folder with New VMs with Secondary Storage from a New Datastore](image-url)
Figure 73. *Create Test/Dev Environment* Workflow Results – The New VMs are Hosted on the Selected ESX Cluster

Figure 74. *Create Test/Dev Environment* Workflow Results – A New Datastore
Create Test/Dev Environment Workflow Results – A New XtremIO Volume which is a Copy of the “Production” Volume

We can see our new Volume in the XMS with an NAA Identifier that matches that of the new Datastore that was created. In the Volume Snapshot Groups tab below, we can see that the new Volume is a copy of our “Production” Volume, thus containing “Production” data. Due to XtremIO’s unique iCDM capabilities (as discussed in the Copy Data Services section), we can see that the new Volume consumes almost no extra space on the storage array, as the Logical Space In Use for the entire Snapshot Group remains approximately the same as it was prior to the creation of the new Volume.

On to the next chapter – using our XtremIO vRO Workflows to design vRA services.
Integrating vRA with vRO and the XtremIO Plugin

Now that we have overviewed the vRealize Orchestrator and learned how it can be used to design and execute general IT tasks as well as storage-related tasks (using the XtremIO Plugin), let’s have a look at how we can use vRO in the vRealize Automation tool.

As mentioned, vRA equips IT teams with an engine that provides self-service capabilities to DevOps and application teams through the use of end-to-end automation of IT tasks. By itself, vRA provides Infrastructure-as-a-Service (IaaS) capabilities to adopters. But when combined with vRO, vRA can leverage the vRO Inventory, Actions and Workflows to grant Anything-as-a-Service (XaaS) capabilities to the framework. In this chapter, we will show XaaS abilities from a storage perspective, using some of the Workflows we exhibited (and designed) in XtremIO Plugin for VMware vRealize Orchestrator.

Initial Configuration

Some basic configurations are required before we can start formulating services for consumers. We will not detail every step of those basic configurations, but we will show how we configured them in our private cloud in Appendix C – vRA Initial Configuration.

Creating Custom Resources

We start the implementation of our XaaS design by creating a Custom Resource. We want to create a Datastore-as-a-Service Catalog Item for users responsible for provisioning storage to ESX clusters, so we will create a VMware Datastore Custom Resource. Every Resource we create can be owned by a privileged consumer, and manipulated through the Items section, once appropriate Blueprint and Actions are created.

To create a new Custom Resource, we go to Design ➔ XaaS ➔ Custom Resources in the vRA interface and click + New.
In the **Resource type** tab, we select an **Orchestrator type** and a **Name** for the new Resource. We can optionally specify a **Description** and a **Version** of that Resource. In the Orchestrator type-box we are only allowed to enter types known by our connected vRO instance (its text box is in fact a search box for vRO known types).

**VMware Datastore - New Resource**

```
<table>
<thead>
<tr>
<th>Resource type</th>
<th>Details Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Orchestrator type: VC:Datastore</td>
<td></td>
</tr>
<tr>
<td>* Name: VMware Datastore</td>
<td></td>
</tr>
<tr>
<td>Description: A VMFS Datastore</td>
<td></td>
</tr>
<tr>
<td>Version: 0.0.0</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 77. New Custom Resource – Resource Type Tab

Clicking **Next >** opens the Details Form tab. The **Details Form** tab specifies all the properties the new Resource will have in vRA. By default, it holds all the properties available in vRO, but several options are available for modifying it. For instance, we can remove information that is irrelevant for our consumers or rearrange the form order. Additionally, we can add fields and even use vRO Actions to add extra information we want to present to consumers.
In our case, we removed information that is imported from the vRO object but is of no use for our consumers, like the Orchestrator ID of the vRO or the Datastore’s unique locator. We also added the **Storage info** section, which includes fields that use custom vRO Actions we developed, to add information about the XtremIO Volumes that comprise the Datastore and their respective XtremIO Cluster name and XMS info.

![Diagram](image)

**Figure 78. New Custom Resource – Details Form Tab**

**Figure 78** shows an example of the **XtremIO Volumes** field that we added, which uses a vRO Action to return the XtremIO Volumes that comprise the Datastore. The different options for editing the Form are marked in this figure: adding new fields (1); changing the position of a field (2); removing a field (3); and editing a field value (4).

When we finish modifying the Details Form we click **Finish** to create the new Datastore Resource.
Designing XaaS Blueprints

After creating our Resource, we move forward to designing a Blueprint that will provision that Resource.

In the Design section under XaaS we go to XaaS Blueprints and click + New to design a new XaaS Blueprint.

![Designing a New XaaS Blueprint](image)

The first step in designing a new XaaS Blueprint is choosing a Workflow that will provision it. Every Workflow in our connected vRO can be chosen for a XaaS Blueprint, but if we want that Blueprint to provision a specific Resource, it must have that Resource type as one of its outputs. We have several Workflows imported by the XtremIO Plugin in our vRO that create and return a Datastore, but we will choose the Datastore Expose Storage Workflow, which creates a new XtremIO Volume, maps it to an ESXi Host or Cluster, and creates a VMFS Datastore on it.
We select a Workflow in the **Workflow** tab and click **Next >**.

**Figure 80. New XaaS Blueprint – Workflow Tab**
In the **General** tab we fill in the desired information (we can just use the information that is imported from the vRO Workflow we used, or use our own info) and click **Next >**. We are not making this Blueprint available as a component in the design canvas since we will be using it as a standalone Blueprint only. But we will show an example of using a Blueprint in the design canvas for a different Blueprint we will design later on. Notice that the **Component Lifecycle** tab disappears when we select the **Make available as a component in the design canvas** check box. We will elaborate on that in our next design.

**Provision New Datastores - New Blueprint**

<table>
<thead>
<tr>
<th>Workflow</th>
<th>General</th>
<th>Blueprint Form</th>
<th>Provisioned Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> Provision New Datastores</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong> Provision new VMFS Datastores to a selected vCenter Cluster. This service also creates and maps the underlying XtremIO Volumes of the Datastores</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Version:</strong> 0.0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Provisioning Workflow:</strong> Datastore Expose Storage</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 81.** New XaaS Blueprint – General Tab
The **Blueprint Form** tab is where we design the Form that will be shown to the vRA user when requesting to run the Catalog Item. It is also where we set and map the vRO Workflow inputs to the vRA Blueprint parameters. By default, all of the Workflow inputs are imported to the Blueprint Form, but we can modify them as we wish, just as we did with the Resource’s Details Form. We can remove optional Workflow inputs we do not want to be a part of the Blueprint Form, or preset values for inputs we do not want the consumer to decide on, such as the XtremIO storage array on which to create the Volume, or the vCenter Server on which to provision the Datastore. Additionally, we can create standards for other inputs. For instance, in our case, we only want to provision storage to entire vCenter Clusters, and not to single ESXi hosts, so we only enable the **vCenter Cluster** field. We can use the Blueprint Form tab to create new input fields, rearrange them in the form, configure Constraints or use external methods (vRO Actions) to set input values, and modify the Request Form of the XaaS Catalog Item as we desire.

We manipulated our form in several ways, including the methods mentioned above (removing inputs, presetting inputs, adding inputs and using vRO Actions to configure them). In Figure 82 we show an example of the **New volume name** parameter, which uses a vRO Action to set the XtremIO Volume name to a string of the form `<VC_Name>_<Datastore_Name>`. The Blueprint Form has similar modifying options as the Resource’s Details Form we reviewed earlier, but with a few additional options that are more typical for input fields, like the **Default value** option, or the **Required** option. It is recommended to refer to the vRA documentation for its variety of options to configure the Blueprint Form and Workflow inputs in the method most suitable for your needs and desires.
After completing our work on the Blueprint Form we click Next > and get to the Provisioned Resource tab. This is where we set the type of Resource that will be provisioned by this Blueprint to the consumer. Some Blueprints may require no Resource to be provisioned such as those which just send reports or rescan some resource in the cloud (like rescanning an ESXi Host). But if we are creating a Resource in our Blueprint which we want the consumer to continue to manage after provisioning, we must select the correct resource type. The drop-down list will only present us with outputs of the Workflow that have a corresponding Resource type created in the Custom Resources page. Here we select the datastores output and now when run, our Blueprint will provision VMware Datastores items.

![Provision New Datastores - New Blueprint](image)

After choosing the Provisioned Resource we click Finish to complete our Blueprint. We continue by publishing the Blueprint to make it available to our Catalog by clicking the Publish button for our new Blueprint in the XaaS Blueprint page.

![Publishing a XaaS Blueprint](image)
Enabling a XaaS Blueprint as a New Catalog Item

After publishing a XaaS Blueprint we need to assign it to a Service for it to be available in our Catalog. This is done through the **Administration** section. Under **Catalog Management → Catalog Items** we can find the newly published Blueprint. We select it and click **Configure**.

![Figure 85. Configuring a Catalog Item](image)

In the **General** tab in the **Configure Catalog Item** window we can choose a desired **Icon** for the Catalog Item and assign it to a **Service**. We are assigning our Blueprint to the **XtremIO** Service we created in advance. The new Catalog Item will have the **Entitlements** of the Service it is assigned to. We then click **OK** to complete the Catalog Item configuration.

![Figure 86. Configure Catalog Item Wizard](image)
Now we can browse to our **Catalog** and see our new Service Item.

![Service Catalog with the “Provision New Datastores” Item](image)

**Figure 87. Service Catalog with the “Provision New Datastores” Item**

**Creating Resource Actions**

After creating a Resource and designing a Blueprint that provisions it, we can add the desired consumer Actions to the Resource. For instance, in our case, once a consumer provisions a Datastore, we would like to allow them to also delete or expand it. Let’s start with creating the “Delete” Action.

In the **Design** section, under **XaaS** we go to **Resource Actions** and click **+ New** to create a new Resource Action.

![Creating a New Resource Action](image)

**Figure 88. Creating a New Resource Action**
Creating Resource Actions are very similar to designing XaaS Blueprints. The process starts with choosing a vRO Workflow which will perform the desired Action, but this time we need to select a Workflow that receives the Resource on which we want to perform the Action as an input. Here we are choosing the **Datastore Delete Storage Workflow**, which unmounts and deletes the Datastore, unmaps the underlying XtremIO Volumes from the ESXi Hosts of the Cluster, and deletes the Volumes from the XtremIO Array. After choosing a Workflow we click **Next >**.

Next, in the **Input Resource** tab we choose the type of Resource on which the Action will be performed, and map it to the correct input of the vRO Workflow. We choose our previously created **VMware Datastore Resource type** and the **datastore Input parameter** and click **Next >**.
Next, in the **Details** tab, we set the **Name**, **Description** and **Version** of the Action and select whether the consumer needs to specify a reason for requesting the action (check the **Hide catalog request information page** for No). In the **Type** field we select what type of Action we are creating: if it is deleting our Resource we will check the **Disposal** option; if it creates a new Resource we should check the **Provisioning** option. An Action can also be both a Disposal and a Provisioning Action or neither of them, depending on the Action we are creating. For example, copying a Datastore to a new Datastore is an example of a Provisioning Action; expanding a Datastore is an example of neither. The Provisioned Resource of the Action can also be of a different type, for example, an Action of booting a new VM on top of the Datastore can provision a Virtual Machine Resource type. Our Action is a Disposal Action, since it deletes our Datastore, so we will check **Disposal**. We can also create criteria regarding when this Action is available to a Resource, but in our case, we will set it to **Always available**.

![Figure 91. New Resource Action – Details Tab](image)

---

85 | Implementing a Private Cloud with VMware vRealize and Dell EMC XtremIO X2
© 2018 Dell Inc. or its subsidiaries.
After clicking **Next >** we get to the **Form** tab, which is where we design the Form that will be presented to the vRA user when requesting to run that Action, as was done in the Blueprint Form tab of the New XaaS Blueprint configuration wizard. We are making the Action as simple as possible for the user, presetting all the parameters in advance so that the user will simply need to confirm the request to run the Action. We also added a field which asks the user to confirm the name of the Datastore to be deleted. That field is highlighted in Figure 92.

**Delete Datastore - New Resource Action**

![Figure 92. New Resource Action – Form Tab](image)

After completing the Action Form, we click **Finish** to create the new Action. We will then publish the Action by clicking the **Publish** button for the new Action in the **Resource Actions** page.

![Figure 93. Publishing a Resource Action](image)
Enabling a Resource Action for Provisioned Items

After publishing a Resource Action, we still need to enable it as an entitled Action to run on a provisioned item. This is done through the Administration section. Under Catalog Management → Entitlements we choose our previously created Entitlement and click Edit:

![Entitlements](image)

We then need to navigate to the Items & Approvals tab which is where we configure Entitled Services, Items and Actions. In the Entitled Actions column, we can either click the + button to browse for Actions or simply use the Search text box to find the Action we want to entitle. We add the desired Action and click Finish to enable the changes.

![Enabling a Resource Action](image)
To choose a new Icon for the Action, go to the **Actions** page under **Administration → Catalog Management**, find the Action for which you want to change the icon, and click **Configure**.

![Configure an Action](image)

**Figure 96. Configuring an Action**

In the **Configure Action** wizard we choose an **Icon** and click **Finish**.

![Configure Action Wizard](image)

**Figure 97. Configure Action Wizard**

We have added an **Expand Datastore** Resource Action using the same steps used to create the **Delete Datastore** Resource Action with the **Datastore Expand Workflow** from the **XtremIO VMware Storage Management** Workflow folder imported by the XtremIO Plugin. The main difference was that the **Type** of the Action was neither Disposal nor Provisioning.
Implementing a Private Cloud with VMware vRealize and Dell EMC XtremIO X2

© 2018 Dell Inc. or its subsidiaries.

Running Catalog Items and Resource Actions

Now that we completed configuring our Datastore-as-a-Service Resource, Blueprint and Actions, we can see how it will look to our consumers when they run it, and what is performed in the background when running the Blueprint and Actions.

Running a Catalog Item

To run a Catalog Item, go to the Catalog section and click Request on the Item you wish to run.

The Form we have designed in the Blueprint Form of the XaaS Blueprint appears. We fill in the desired information and click Submit.

New Request

Figure 98. Requesting a Catalog Item

Figure 99. Requesting a Catalog Item
If everything entered in the Request is valid, we will get a message saying **The request has been submitted successfully.** That does not mean that the action we requested was completed, but only that our request was submitted. It is still pending approval of an administrator, provided that we created an Approval Policy (which was not done in our deployment example), and the successful execution of the vRO Workflow.

![Request](image1.png)

**Figure 100. Successfully Submitting a Request**

Once the request is submitted, we will be able to see it in the **Requests** section in the vRA interface, and can view its details when selecting it and clicking the **View Details** button. Common Request statuses are: **Pending Approval, In Progress, Failed and Successful.**

![Requests](image2.png)

**Figure 101. vRA Requests Section**

While the request is in the **In Progress** status, we know that it was approved and we will be able to see the vRO Workflow running in the vRO interface, just as we saw in the **Executing Workflows** section. When the request state changes to **Successful** it indicates that the vRO Workflow completed successfully. We can now review the **Items** section and find our newly provisioned resource.

![Items](image3.png)

**Figure 102. vRA Items Section**
To view the Item details, we choose it and click **View Details**. We will then see the Datastore details in the form which we designed in the Creating Custom Resources step.

On the right, we will see the Actions that we can perform on that resource. These are the Resource Actions we created in the Creating Resource Actions section. The Resource Actions can be initiated from this page or from the main Items page. We will run these next, but let’s first see how our new Datastore looks from a vSphere and XtremIO perspective.
Running a Resource Action

After a Resource has been provisioned, we can run our created Resource Actions on it. To run an Action on a provisioned Resource, go to the Items section, choose a Resource, click Actions and select the Action you want to run.

The Action Form which we designed will open and prompt us with the required inputs for running it, if any. We are running the Expand Datastore Action, so we will be prompted to enter the new size of the Datastore. We insert our input and click Submit:
A Resource Action acts the same as a Catalog Request: it will appear in the Requests section with its status, run the matching vRO Workflow once approved, and change its status to *Successful* if the Workflow completes successfully. In our case, the Action will not add or remove any Resource from the Items section, but only change its details once completed.

The XtremIO Volume size will of course change accordingly.

The *Delete Datastore* Action acts similarly, except that it will delete the Datastore and XtremIO Volume and will remove the Datastore Item from the Items section.
Using an XaaS Blueprint in the Design Canvas

XaaS Blueprints can be used as part of bigger Blueprints that carry out composite procedures at a higher level. We build composite Blueprints when we want to create a procedure that runs several Workflows and can create multiple resources of various types. In this section, we will demonstrate such a composite Blueprint.

Imagine a process in which for every new developer hired to your organization, you need to create several resources in your infrastructure; for example, adding a user for the new developer in your Active Directory and creating a test/dev environment that copies your production environment for your developer to work on. With vRA you can create a composite Blueprint that creates all of these resources and allows you to manage them according to the developer’s needs, and all on a self-service basis.

For this we first need to create the Custom Resources we want to allocate in our composite Blueprint. In our case, those are: (1) an Active Directory User; (2) a Test/Dev Environment (which we will represent in our solution using the VM Folder type). Then, we need to design the XaaS Blueprints of those two Resources, this time with selecting the Make available as a component in the design canvas check box in the General tab of the New Blueprint wizard. Let’s see how that looks for our second XaaS Blueprint – Create Test/Dev Environment.

![Create Test/Dev Environment - New Blueprint](image)

Figure 110. Designing an XaaS Blueprint to be available in the Design Canvas – General Tab

When the option to make the XaaS Blueprint available as a component in the design canvas is selected, the wizard will present the Component Lifecycle tab for completion.

We are using the Create Test/Dev Environment Workflow that we developed in the vRO chapter in the Designing a New Workflow section. We preset almost every input in the Blueprint Form tab and leave only a name to be entered for the environment. In the Provisioned Resource tab, we select the provisioned resource to be of VM Folder type, since it is also the input required when running the disposal Workflow – Delete Test/Dev Environment.
In the Component Lifecycle tab, we design the lifecycle behavior of the XaaS Blueprint when used in a composite blueprint.

Create Test/Dev Environment - New Blueprint

![Diagram of the Component Lifecycle tab]

In the **Component Lifecycle** tab, we design the lifecycle behavior of the XaaS Blueprint when used in a composite blueprint that you configure in the design canvas. Necessary lifecycle actions include:

- **Scalable**: [ ]
- **Provisioning workflow**: Create Test/Dev Environment
- **Update workflow**: [ ]
- **Destroy workflow**: [ ]
- **Category**: XaaS

The **Scalable** check box will define whether the consumer can change the number of instances of the XaaS Blueprint as part of a scale-in or scale-out operation. We only allow one Test/Dev Environment per deployment, so we will uncheck this option. The **Provisioning workflow** is that which we selected at the first tab of the wizard. The **Update workflow** is where we can choose the vRO Workflow which will run during update operations, which also relates to scale-in or scale-out operations. We have no update procedure for our environment, so we leave this field empty. The **Destroy workflow** is where we choose the vRO Workflow which will run when we want to destroy this resource. When clicking the button next to the Destroy Workflow field, we are prompted to choose a Workflow from our vRO instance. We choose this Workflow to be the **Delete Test/Dev Environment** Workflow that we developed in the previous chapter. We finish by choosing a **Category** to which the new Blueprint will be assigned when using the design canvas. If we want to make the new XaaS Blueprint also available separately in the Catalog, we need to assign it to a service through the **Administration → Catalog Management → Catalog Items** menu, just as we did in the **Enabling a XaaS Blueprint as a New Catalog Item** subsection.

See Table 8 for a summary of the components of our two new designed XaaS Blueprints.

### Table 8. XaaS Blueprints Designed for “New Developer” Composite Blueprint

<table>
<thead>
<tr>
<th>BLUEPRINT</th>
<th>PROVISIONING WORKFLOW (INCLUDING LOCATION IN VRO)</th>
<th>PROVISIONED RESOURCE VRO TYPE</th>
<th>SCALABLE?</th>
<th>DESTROY WORKFLOW (INCLUDING LOCATION IN VRO)</th>
<th>CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an AD user</td>
<td>/Library/Microsoft/Active Directory/User/Create a user with a password in a group</td>
<td>AD:User</td>
<td>No</td>
<td>/Library/Microsoft/Active Directory/User/Destroy a user</td>
<td>XaaS</td>
</tr>
<tr>
<td>Create Test/Dev Environment</td>
<td>/Custom XtremIO Workflows/Create Test/Dev Environment (self-created)</td>
<td>VC:VmFolder</td>
<td>No</td>
<td>/Custom XtremIO Workflows/Delete Test/Dev Environment (self-created)</td>
<td>XaaS</td>
</tr>
</tbody>
</table>
Now we move forward to designing a composite Blueprint. To create a new composite Blueprint, go to the Design section and then to Blueprints, and click + New. Enter a name for the new Blueprint (an ID will be automatically entered) and click OK (we are leaving the other fields of this wizard empty, but they can be set as needed).

Figure 112. Creating a New Composite Blueprint for Design
The **New Blueprint** window appears containing the **Design Canvas** in which the designed XaaS Blueprints will be placed. On the left side of the **Category** section we select the **XaaS** Category to find the new designed XaaS Blueprints as shown below. To add a XaaS Blueprint as a component in the composite Blueprint, simply drag it to the Design Canvas.

![Design Canvas](image)

**Figure 113.  Adding a XaaS Component to a Composite Blueprint**
Once a component is added to the canvas, a wizard for configuring it will open. In our examples, we will have two tabs to configure for each XaaS Blueprint added – a **General** tab and a tab to further modify the XaaS Blueprint Form.

For the Create an AD user XaaS Blueprint, no further changes are made.
Next, we continue and add the **Create Test/Dev Environment** XaaS Blueprint.

In the **Create Test/Dev Environment** tab of the **Create_Test_Dev_Environment** component, we set the value for the **Name for the new environment** to be the name of the AD user we are creating as part of the procedure. We do this by selecting the text box, switching to **Advanced Settings...** in the right section, clicking the **Value** drop down list, choosing **Field**, and then clicking the **Define Field Values** button. We will then see the **Define Field Values** box where we will choose the **User name** field of the **Create_an_AD_User** Blueprint, and this binds the two parameters together. Now each new Test/Dev environment will be named after the new developer that was added to our team!
We click Finish to complete the Blueprint and make sure to publish it, so it will be available in the Catalog Management.

Figure 117. Publishing a Composite Blueprint
After publishing the Blueprint, we add it to a Service through the Catalog Management, so it would be available to run through the Catalog (we added the Blueprint to the XtremIO service). We will run the new Blueprint and see its results.

We fill in the name of the new AD user and click Submit.
The new request, once approved, will run the two provisioning Workflows of the Blueprint – *Create a user with a password in a group* and *Create Test/Dev Environment*, and will return the provisioned resources – a new AD User and a new VM Folder representing the new Test/Dev Environment on which the new developer will work.

**Figure 121. The New Developer Deployment**

**Figure 122. The New AD User Item**
We can also see the new resources provisioned in the infrastructure itself – a new user in our Active Directory, the new Virtual Machines in our vSphere instance and the new XtremIO Volume which is the underlying storage of the Datastore that holds the secondary disks of our new Test/Dev Environment.
To “destroy” the New Developer’s vRA resources, we choose the deployment in the Items section, click Actions and select Destroy. At the Destroy Request window, click Submit. This will run the Destroy Workflows of each provisioned resource that we defined when creating their Blueprints. At the end of the Destroy run, the resources will be removed from the infrastructure and from the vRA Items section.
References
6. XtremIO CTO Blog (with product announcements and technology deep dives) – https://xtremio.me/
9. Dell EMC XtremIO Plugin for VMware vRealize Orchestrator 2.0.0 – https://download.emc.com/downloads/DL86266_XtremIO_2.0.0_plugin_for_VMware_vRealize_(vCenter)_Orchestrator.zip?source=OLS
Appendix A – Built-in Workflows by the XtremIO Plugin for vRO

There are about 120 built-in Workflows provided by the XtremIO Plugin for vRO, divided into sub-categories. Here is a full list of the folders imported by the plugin and the Workflows within them.

**XtremIO CG Management Workflows**

This folder contains Workflows performing basic operations on XtremIO Consistency Groups.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG Add Volume</td>
<td>Adds one or more Volumes to a Consistency Group.</td>
</tr>
<tr>
<td>CG Copy</td>
<td>Creates a read-write copy of all Volumes in a selected Consistency Group.</td>
</tr>
<tr>
<td>CG Create</td>
<td>Creates a Consistency Group (with or without a Tag).</td>
</tr>
<tr>
<td>CG Delete</td>
<td>Deletes a Consistency Group.</td>
</tr>
<tr>
<td>CG List</td>
<td>Lists all or some (according to a certain criteria) Consistency Groups in the storage cluster.</td>
</tr>
<tr>
<td>CG Links</td>
<td>Lists all Consistency Group links associated with a Consistency Group.</td>
</tr>
<tr>
<td>CG Volumes</td>
<td>Lists all volumes in a selected Consistency Group.</td>
</tr>
<tr>
<td>CG Protection</td>
<td>Creates a read-only cross-consistent copy of all Volumes in a selected Consistency Group.</td>
</tr>
<tr>
<td>CG Refresh</td>
<td>Refreshes a Consistency Group copy by updating all Volumes in the Consistency Group copy with updated data from their original Volumes.</td>
</tr>
<tr>
<td>CG Remove Volume</td>
<td>Removes one or more Volumes from a Consistency Group.</td>
</tr>
<tr>
<td>CG Rename</td>
<td>Renames a Consistency Group.</td>
</tr>
<tr>
<td>CG Restore</td>
<td>Restores all Volumes in a Consistency Group with data from a selected Consistency Group Snapshot.</td>
</tr>
</tbody>
</table>

**XtremIO Cluster Management Workflows**

This folder contains Workflows performing basic operations on XtremIO Clusters.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster List</td>
<td>Lists all XtremIO Clusters managed by a given XMS.</td>
</tr>
</tbody>
</table>

**XtremIO IG Management Workflows**

This folder contains Workflows performing basic operations on XtremIO Initiator Groups.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IG Add Initiator</td>
<td>Adds one or more Initiator to an Initiator Group.</td>
</tr>
<tr>
<td>IG Create</td>
<td>Creates one or more Initiator Groups (with or without a Tag).</td>
</tr>
<tr>
<td>IG Delete</td>
<td>Deletes one or more Initiator Groups.</td>
</tr>
<tr>
<td>IG List</td>
<td>Lists all or some (according to a certain criteria) Initiator Groups in the storage cluster.</td>
</tr>
<tr>
<td>IG List Initiators</td>
<td>Lists all Initiators in an Initiator Group.</td>
</tr>
<tr>
<td>IG Volumes</td>
<td>Lists all Volumes mapped to a selected Initiator Group.</td>
</tr>
<tr>
<td>IG Remove Initiator</td>
<td>Removes one or more Initiators from an Initiator Group.</td>
</tr>
<tr>
<td>IG Rename</td>
<td>Renames an Initiator Group.</td>
</tr>
<tr>
<td>IG Show</td>
<td>Shows Initiator Group attributes as a formatted string for some or all Initiator Groups.</td>
</tr>
</tbody>
</table>
XtremIO Performance Management Workflows

This folder contains Workflows producing performance reports on the XtremIO infrastructure and related components.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Report</td>
<td>Outputs available Cluster information (in CSV, HTML or JSON formats).</td>
</tr>
<tr>
<td>Datastore Report</td>
<td>Output Datastore performance information for a specific ESXi Host.</td>
</tr>
<tr>
<td>Host Report</td>
<td>Output performance information for ESXi Hosts.</td>
</tr>
<tr>
<td>IG Report</td>
<td>Outputs performance information for Initiator Groups.</td>
</tr>
<tr>
<td>Workflow Usage Report</td>
<td>Returns usage count information for XtremIO Workflows that have been successfully invoked.</td>
</tr>
</tbody>
</table>

XtremIO Protection Scheduler Workflows

This folder contains Workflows performing basic operations on XtremIO Protection Schedulers.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protection Scheduler Create</td>
<td>Creates a Protection Scheduler (with or without a Tag) on a particular object, together with the schedule itself.</td>
</tr>
<tr>
<td>Protection Scheduler Delete</td>
<td>Deletes a Protection Scheduler.</td>
</tr>
<tr>
<td>Protection Scheduler List</td>
<td>Lists all or some (according to a certain criteria) Protection Schedulers in the storage cluster.</td>
</tr>
<tr>
<td>Protection Scheduler Modify</td>
<td>Modifies various settings of a selected Protection Scheduler.</td>
</tr>
<tr>
<td>Protection Scheduler Resume</td>
<td>Resumes a suspended Protection Scheduler.</td>
</tr>
<tr>
<td>Protection Scheduler Suspend</td>
<td>Suspends a Protection Scheduler.</td>
</tr>
</tbody>
</table>

XtremIO RecoverPoint Management Workflows

This folder contains Workflows performing basic operations on a RecoverPoint Appliance protecting XtremIO Volumes (replication-based protection).

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP Add</td>
<td>Adds a RecoverPoint Server to the list of RecoverPoint Servers available in the vRealize Inventory for use by XtremIO RecoverPoint Workflows.</td>
</tr>
<tr>
<td>RP Create CG</td>
<td>Creates and enables a RecoverPoint Consistency Group.</td>
</tr>
<tr>
<td>RP Delete CG</td>
<td>Deletes a RecoverPoint Consistency Group without deleting its storage.</td>
</tr>
<tr>
<td>RP Delete CG Storage</td>
<td>Deletes a RecoverPoint Consistency Group with its storage.</td>
</tr>
<tr>
<td>RP List</td>
<td>Lists all RecoverPoint Servers available in the vRealize inventory.</td>
</tr>
<tr>
<td>RP List CGs</td>
<td>Lists all Consistency Groups in a RecoverPoint Server.</td>
</tr>
<tr>
<td>RP Modify CG</td>
<td>Modifies various settings of a selected RecoverPoint Consistency Group.</td>
</tr>
<tr>
<td>RP Remove</td>
<td>Removes a RecoverPoint Server from the list of RecoverPoint Servers available in the vRealize Inventory.</td>
</tr>
</tbody>
</table>
XtremIO Snapshot Set Management Workflows

This folder contains Workflows performing basic operations on XtremIO Snapshot Sets.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SnapshotSet Copy</td>
<td>Creates a read-write Copy of all the Volumes in a selected Snapshot Set.</td>
</tr>
<tr>
<td>SnapshotSet Delete</td>
<td>Deletes a selected Snapshot Set.</td>
</tr>
<tr>
<td>SnapshotSet List</td>
<td>Lists all or some (according to a certain criteria) Snapshot Sets in the storage cluster.</td>
</tr>
<tr>
<td>SnapshotSet Map</td>
<td>Maps a Snapshot Set to an Initiator Group.</td>
</tr>
<tr>
<td>SnapshotSet Protection</td>
<td>Creates a read-only cross-consistent copy of all the Volumes in a selected Snapshot Set.</td>
</tr>
<tr>
<td>SnapshotSet Refresh</td>
<td>Refreshes a Snapshot Set copy with data from another Snapshot Set copy.</td>
</tr>
<tr>
<td>SnapshotSet Rename</td>
<td>Renames a Snapshot Set.</td>
</tr>
<tr>
<td>SnapshotSet Unmap</td>
<td>Unmaps a Snapshot Set from an Initiator Group it is mapped to.</td>
</tr>
</tbody>
</table>

XtremIO Storage Management Workflows

This folder contains Workflows performing high-level procedures on Datastores and VMs residing on XtremIO storage.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datastore Delete Storage</td>
<td>Deletes a VMFS Datastore and its underlying XtremIO Volumes (also shuts down associated VMs, unmaps the Volumes and Deletes Consistency Groups that remain empty following the Volumes' deletion).</td>
</tr>
<tr>
<td>Datastore Expose Storage</td>
<td>Exposes one or more XtremIO Volumes (creating new Volumes or using a set of existing unmapped Volumes or Volumes that belong to a Consistency Group) to provision a VMFS Datastore to one or more ESXi Hosts in a given vSphere Cluster.</td>
</tr>
<tr>
<td>VM Clone Storage</td>
<td>Copies specified Volumes/Datastores and connects the copied Datastore’s VMDKs to a set of specified VMs (usually for test/dev purposes). This procedure disregards operating system disks and requires the destination VMs to be up and running with an existing operating system (the destination VMs’ prefixes must coincide the source VMs’ names).</td>
</tr>
<tr>
<td>VM Delete Storage</td>
<td>Deletes the Datastores containing application data of a VM and its underlying XtremIO Volumes (also unmaps the Volumes and deletes Consistency Groups that remain empty following the Volumes’ deletion). This procedure can either delete the entire VM or simply unmount the datastore containing application data from all ESXi Hosts it is mapped to.</td>
</tr>
<tr>
<td>VM Expose Storage</td>
<td>Exposes an XtremIO Volume (creating a new Volume or using an existing one) to provision a VMFS Datastore to an ESXi Cluster and provision either a VMDK or RDM to a selected VM.</td>
</tr>
</tbody>
</table>

XtremIO Tag Management Workflows

This folder contains Workflows performing basic operations on XtremIO Tags.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Apply</td>
<td>Applies a Tag to a given XtremIO entity.</td>
</tr>
<tr>
<td>Tag Create</td>
<td>Creates a Tag for a given Tag Type.</td>
</tr>
<tr>
<td>Tag Delete</td>
<td>Deletes one or more Tags.</td>
</tr>
<tr>
<td>Tag List</td>
<td>Lists all Tags for a given Tag Type.</td>
</tr>
<tr>
<td>Tag Remove</td>
<td>Removes a Tag from a given XtremIO entity.</td>
</tr>
<tr>
<td>Tag Rename</td>
<td>Renames a Tag.</td>
</tr>
</tbody>
</table>
**XtremIO VMware Storage Management Workflows**

This folder contains Workflows performing high-level procedures on Datastores, Hosts, vCenters and VMs using XtremIO storage.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datastore Copy</td>
<td>Copies an XtremIO Volume on which a specified Datastore is based.</td>
</tr>
<tr>
<td>Datastore Create</td>
<td>Creates a VMFS Datastore on an existing unused Volume and mounts the Datastore to all Hosts in the Cluster of the specified Host.</td>
</tr>
<tr>
<td>Datastore Delete</td>
<td>Deletes a specified Datastore (after shutting down and deleting VMs utilizing that Datastore for operating system’s disks, disconnecting its VMDKs and unmount it from all ESXi Hosts it is mapped to).</td>
</tr>
<tr>
<td>Datastore Expand</td>
<td>Expands an existing Datastore.</td>
</tr>
<tr>
<td>Datastore List</td>
<td>Lists all Datastores known to a given vCenter Server Instance.</td>
</tr>
<tr>
<td>Datastore Mount</td>
<td>Mounts a specified previously-created Datastore on a selected Host or all Hosts in the ESXi Cluster.</td>
</tr>
<tr>
<td>Datastore Protection</td>
<td>Creates a read-only copy of the underlying XtremIO Volume of a specified Datastore.</td>
</tr>
<tr>
<td>Datastore Reclaim Storage</td>
<td>Reclaims unused space in a specified Datastore and deletes unused VMDK files.</td>
</tr>
<tr>
<td>Datastore Refresh</td>
<td>Refreshes a selected copy of the underlying XtremIO Volume that a Datastore is based on with updated data from the original Volume.</td>
</tr>
<tr>
<td>Datastore Restore</td>
<td>Restores the underlying XtremIO Volume that a Datastore is based on from a previously-taken copy of that Volume.</td>
</tr>
<tr>
<td>Datastore Show</td>
<td>Shows Datastore information (capacity, contained VMs, used and unused VMDK/RDM files, known Hosts, etc.).</td>
</tr>
<tr>
<td>Datastore Show Storage</td>
<td>Shows XtremIO Volumes, Consistency Groups and Snapshot Sets that make up a specified Datastore.</td>
</tr>
<tr>
<td>Datastore Unmount</td>
<td>Unmounts a Datastore from a selected Host or all Hosts it is mounted to.</td>
</tr>
<tr>
<td>Host Add SSH Connection</td>
<td>Adds SSH connection information for a given ESXi Host to the vRealize Inventory.</td>
</tr>
<tr>
<td>Host Conformance</td>
<td>Checks for a single ESXi Host or all ESXi Hosts whether for all XtremIO Volumes that are mapped to the Hosts the vRealize instance have a connection to the matching XMS Servers managing them.</td>
</tr>
<tr>
<td>Host Delete Storage</td>
<td>Deletes a list of selected Volumes mounted to a Host (including unmapping the Volumes from the Hosts and deleting Consistency Groups that remain empty following the Volume deletion).</td>
</tr>
<tr>
<td>Host Expose Storage</td>
<td>Exposes new or existing XtremIO Volumes to an ESXi Host (can also create a Consistency Group for newly-created Volumes or choose an existing Consistency Group to expose all of its Volumes to the Host).</td>
</tr>
<tr>
<td>Host List</td>
<td>Lists the Hosts known to a specified vCenter Server.</td>
</tr>
<tr>
<td>Host Modify Settings</td>
<td>Modifies settings for a given ESXi Host (requires an SSH Host Instance to be setup for that Host in advance).</td>
</tr>
<tr>
<td>Host Remove SSH Connection</td>
<td>Removes an SSH Host configuration entry from the vRealize Inventory.</td>
</tr>
<tr>
<td>Host Rescan</td>
<td>Rescans a single Host or all Hosts for a given vCenter Cluster.</td>
</tr>
<tr>
<td>Host Show</td>
<td>Shows information on a given Host (WWNs, used and unused disks, etc.).</td>
</tr>
<tr>
<td>vCenter Cluster Delete Storage</td>
<td>Deletes all unused XtremIO Volumes for each ESXi Host in a specified vCenter Cluster.</td>
</tr>
<tr>
<td>vCenter Cluster Expose Storage</td>
<td>Exposes new or existing XtremIO Volumes to all ESXi Hosts in a vCenter Cluster.</td>
</tr>
<tr>
<td>vCenter List</td>
<td>Lists vCenter Servers that were registered to the vRealize Inventory.</td>
</tr>
<tr>
<td>vCenter List Clusters</td>
<td>Lists the vCenter Clusters of a specified vCenter Server.</td>
</tr>
<tr>
<td>vCenter Remove</td>
<td>Removes a vCenter Server Instance from the vRealize Inventory.</td>
</tr>
<tr>
<td>vCenter Show</td>
<td>Shows a list of all Hosts of a given vCenter Cluster.</td>
</tr>
<tr>
<td>VM Add VMDK</td>
<td>Provisions new or existing VMKDs/RDMs to a VM.</td>
</tr>
<tr>
<td>VM Delete</td>
<td>Deletes a VM and removes its primary storage (preserving secondary VMDKs and RDMs). This procedure shuts the VM down before deleting it.</td>
</tr>
<tr>
<td>VM List</td>
<td>Lists VMs running on a single Host or all Hosts in a given vCenter Server.</td>
</tr>
</tbody>
</table>
### XtremIO Volume Management Workflows

This folder contains Workflows performing basic operations on XtremIO Volumes.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume Copy</td>
<td>Creates a read-write copy of one or more XtremIO Volumes.</td>
</tr>
<tr>
<td>Volume Create</td>
<td>Creates one or more XtremIO Volumes (with or without a Tag).</td>
</tr>
<tr>
<td>Volume Delete</td>
<td>Deletes one or more Volumes in the storage cluster (Volumes must be unmapped from any Host prior to deletion).</td>
</tr>
<tr>
<td>Volume Expand</td>
<td>Expands an XtremIO Volume.</td>
</tr>
<tr>
<td>Volume List</td>
<td>Lists all or some (according to a certain criteria) Volumes in the storage cluster.</td>
</tr>
<tr>
<td>Volume Map</td>
<td>Maps one or more Volumes to an Initiator Group.</td>
</tr>
<tr>
<td>Volume Modify</td>
<td>Modifies various settings of a selected Volume.</td>
</tr>
<tr>
<td>Volume Protection</td>
<td>Creates a read-only copy of one or more XtremIO Volumes.</td>
</tr>
<tr>
<td>Volume Refresh</td>
<td>Refreshes either a Volume or a Volume copy with data from a different copy of the Volume (either the original Volume or one of its copies).</td>
</tr>
<tr>
<td>Volume Rename</td>
<td>Renames a Volume.</td>
</tr>
<tr>
<td>Volume Restore</td>
<td>Restores a Volume from a previously-taken read-only copy of it.</td>
</tr>
<tr>
<td>Volume Show</td>
<td>Shows information on one or more Volumes (NAAs, capacity and other attributes).</td>
</tr>
<tr>
<td>Volume Unmap</td>
<td>Unmaps one or more Volumes from an Initiator Group.</td>
</tr>
</tbody>
</table>

### XtremIO XMS Management Workflows

This folder contains Workflows performing basic operations on XtremIO Management Servers.

<table>
<thead>
<tr>
<th>WORKFLOW</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMS Add</td>
<td>Adds an XMS to the XMS list in the vRealize Inventory to run XtremIO Workflows on.</td>
</tr>
<tr>
<td>XMS List</td>
<td>Lists the XMS Servers that are present in the vRealize Inventory.</td>
</tr>
<tr>
<td>XMS Remove</td>
<td>Removes an XMS from the XMS list in the vRealize Inventory.</td>
</tr>
<tr>
<td>XMS Show</td>
<td>Shows an XMS’s attributes (such as IP, name, clusters managed, etc.).</td>
</tr>
</tbody>
</table>
Appendix B – vRO Initial Configuration

In this appendix, we present a list of procedures fundamental for initially configuring your vRealize Orchestrator with your vCenter Server and an XtremIO Management Server. These procedures are in fact Workflows you should run on your vRO instance to integrate it in your environment.

Adding a vCenter Instance to your vRO

Adding a vCenter Server instance to your vRO is done using the Add a vCenter Server instance Workflow found in the Library → vCenter → Configuration folder.

Figure 128. Add a vCenter Server Instance Workflow – Set the vCenter Server Instance Properties Step
Figure 129. Add a vCenter Server Instance Workflow – Set the Connection Properties Step

Figure 130. Add a vCenter Server Instance Workflow – Additional Endpoints Step
Registering a vRO as an Extension to your vCenter Server

Registering a vRO as an extension to your vCenter Server is done using the **Register vCenter Orchestrator as a vCenter Server extension** Workflow found in the **Library → vCenter → Configuration** folder.

![Register vCenter Orchestrator as a vCenter Server Extension Workflow](image)

Figure 131. Register vCenter Orchestrator as a vCenter Server Extension Workflow
Adding an XtremIO Management Server to your vRO

Adding an XtremIO Management Server (XMS) to your vRO is done using the XMS Add Workflow found in the Library → XtremIO → XtremIO XMS Management folder imported by the XtremIO Plugin for vRO.

Figure 132. XMS Add Workflow – Connection and Credentials Step

Figure 133. XMS Add Workflow – SSL Step
Appendix C – vRA Initial Configuration

In this appendix we present a list of configurations fundamental for working with your vRealize Automation, including its integration with your VMware infrastructure. Some of these are configurations to be done from within the vRA interface, and some are Workflows you should run on your vRO instance.

Creating Endpoints for your Infrastructure

Creating Endpoints to your vRA is done through the Infrastructure → Endpoints → Endpoints menu. We created endpoints to both our vCenter Server and our vRO instance.

![Figure 134. vSphere (vCenter) Endpoint in vRA](image)

![Figure 135. vRealize Orchestrator Endpoint in vRA](image)
Creating a Fabric Group

Creating a Fabric Group in your vRA is done through the Infrastructure → Endpoints → Fabric Groups menu. We created a single Fabric Group in our vRA.

Figure 136. A Fabric Group in vRA
Creating a Business Group

Creating a Business Group in your vRA is done through the **Administration → Users & Groups → Business Groups** menu. We created a single Business Group in our vRA.

![Figure 137. A Business Group in vRA – General Tab](image)

![Figure 138. A Business Group in vRA – Members Tab](image)
Creating Reservations and Reservation Policies

Creating Reservation Policies in your vRA is done through the Infrastructure → Reservations → Reservation Policies menu. We created a couple of Reservation Policies in our vRA, one per ESX Cluster.

![Reservation Policies in vRA](image1.png)

Creating Reservations in your vRA is done through the Infrastructure → Reservations → Reservations menu. We created one Reservation per ESX Cluster from within in vRA, each with a different Reservation Policy. Below is the PROD-SERVERS Reservation.

![A Reservation in vRA – General Tab](image2.png)
The second Reservation we created for the TEST-DEV-SERVERS cluster is virtually the same, only without the Production_Datastore Resource and with the TEST-DEV-SERVERS Reservation Policy.
Creating a Service

Creating Services in your vRA is done through the **Administration → Catalog Management → Services** menu. We created three Services in our vRA – **Windows**, **Linux** and **XtremIO**. Below is the XtremIO Service.

![Figure 143. A Service in vRA](image)

Creating an Entitlement

Creating Entitlements in your vRA is done through the **Administration → Catalog Management → Entitlements** menu. We created a single Entitlement in our vRA.

![Figure 144. An Entitlement in vRA – General Tab](image)
vRO Configuration

The vRO Configuration in your vRA is comprised of two parts: Server Configuration and Endpoints. Configuring your vRA-connected vRO instance is done through the Administration ➔ vRO Configuration ➔ Server Configuration menu.

Configuring your vRA-connected vRO instance’s Endpoints is done through the Administration ➔ vRO Configuration ➔ Endpoints menu. We created an Endpoint for our vCenter Server and for our Active Directory Server.
Figure 147. vRO vCenter Server Endpoint in vRA – Plug-in Tab

Figure 148. vRO vCenter Server Endpoint in vRA – Endpoint Tab

Figure 149. vRO vCenter Server Endpoint in vRA – Details Tab – Select a vCenter Server Instance Page

Figure 150. vRO vCenter Server Endpoint in vRA – Details Tab – Update the Connection Properties Page
Figure 151. vRO vCenter Server Endpoint in vRA – Details Tab – Update the Log-in Properties Page

Figure 152. vRO Active Directory Endpoint in vRA – Plug-in Tab

Figure 153. vRO Active Directory Endpoint in vRA – Endpoint Tab
Notes:

1. We used port 636 in our Active Directory Endpoint since we are using SSL in our authentication. This is mandatory if you want your vRO instance to be able to manage Active Directory Users.

2. Creating vRO Endpoints through vRA add the specified object in your vRO instance, so you do not need to add them again using the vRO Workflows shown in Appendix B – vRO Initial Configuration.

3. If needed, you can configure other vRO Endpoints through the vRA, including HTTP-REST Endpoints, PowerShell, SQL Plug-in, etc.
Adding vRA Endpoints to vRO

It is recommended to also add vRA Endpoints to your vRO. This is done through vRO Workflows. We are adding both the vRA Host and vRA IaaS Hosts (one Endpoint per Host).

Adding a vRA Host to your vRO is done using the Add a vRA host Workflow found in the Library → vRealize Automation → Configuration folder.

![Add a vRA host Workflow](image)

Figure 155. Add a vRA Host Workflow – Host Properties Step

---

© 2018 Dell Inc. or its subsidiaries.
Figure 156. Add a vRA Host Workflow – User Credentials Step

- Session mode: 
  - Shared Session

- Tenant: 
  - tomer.xtremio

- Authentication username: 
  - gilad@tomer.xtremio

- Authentication password: 
  - ********
Adding a vRA IaaS Host to your vRO is done using the **Add the IaaS host of a vRA host** Workflow found in the **Library → vRealize Automation → Configuration** folder.

![Add the IaaS host of a vRA host Workflow](image)

**Figure 157. Add the IaaS Host of a vRA Host Workflow – Common Parameters Step**
Figure 158. Add the IaaS Host of a vRA Host Workflow – Host Properties Step

Figure 159. Add the IaaS Host of a vRA Host Workflow – Proxy Settings Step
Figure 160. Add the IaaS Host of a vRA Host Workflow – User Credentials Step
### Figure 161. Add the IaaS Host of a vRA Host Workflow – Domain and Workstation Step

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Common parameters</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Add an IaaS host</td>
<td></td>
</tr>
<tr>
<td>2a</td>
<td>Host Properties</td>
<td></td>
</tr>
<tr>
<td>2b</td>
<td>Proxy settings</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Host Authentication</td>
<td></td>
</tr>
<tr>
<td>3a</td>
<td>User Credentials</td>
<td></td>
</tr>
<tr>
<td>3b</td>
<td>Domain and Workstation</td>
<td><code>dc-prod.tomer.xtremio</code></td>
</tr>
</tbody>
</table>
How to Learn More

For a detailed presentation explaining XtremIO X2 Storage Array's capabilities and how XtremIO X2 substantially improves performance, operational efficiency, ease-of-use and total cost of ownership, please contact XtremIO X2 at XtremIO@emc.com. We will schedule a private briefing in person or via a web meeting. XtremIO X2 provides benefits in many environments and mixed workload consolidations, including virtual server, cloud, virtual desktop, database, analytics and business applications.