

PURE VALIDATED DESIGN

Increase SQL Server Resilience with Hybrid Cloud

Featuring FlashArray™ with ActiveCluster™ and Pure Cloud Block Store™ for Azure





Contents

Executive Summary	4
Introduction	4
Solution	4
Overview.....	4
Solution Benefits.....	5
Terminology	5
Technology Overview	6
Pure FlashArray//X90 Storage	6
Purity for FlashArray (Purity//FA 6)	7
Pure Cloud Block Store	7
Purity CloudSnap	8
Pure FlashArray//C40 Storage	8
Pure Storage SSMS extension	8
Pure1®	9
VMware ESXi Cluster	10
Technical Solution Design	10
Software Versions	10
Architecture	11
ActiveCluster	12
Uniform and Non-Uniform Host Access	13
VMware ESXi 7.0U1	15
SQL Server Failover Cluster Instance (SQLFCI).....	15
VMware Virtual Machine Disk Layout.....	16
Azure Virtual Machine Disk Layout	18
Data Protection	18
Data Migration.....	22
Failure Scenarios	22
Deployment Guide.....	24
Configure the FlashArray	24
Configure ActiveCluster	24
Configure ESXi Cluster	25
Configure SQL VMs	26
Configure Pure Cloud Block Store	35
Configure Async Replication to Pure Cloud Block Store and FlashArray//C.....	39
Configure Azure SQL VMs	42
Creating and Configuring a 2-Node SQL Server FCI using the Azure Portal	42
Configure CloudSnap to Azure	49
Configure VSS Backup of SQL Server Databases	50



Monitoring	54
Monitoring using FlashArray GUI Dashboard.....	54
Plan and Monitor with Pure1.....	56
Monitoring Using Windows Admin Center	62
Monitoring Using Pure Storage Plugin for the vSphere Client.....	63
Conclusion	64
Additional Documentation	64
Product Support	64
About the Authors	65
Document Updates	65
Document Revisions	65



Executive Summary

This guide focuses on a Microsoft SQL Server business-resilience hybrid-cloud solution. Enterprises require business continuity regardless of where the data exists. The solution provides business continuity across on-premises data centers and disaster recovery with the public cloud. This guide outlines how to ensure data is available wherever and whenever needed. This solution will be based on VMware vSphere 7.0U1 but will not dictate a specific bare-metal hardware implementation.

A Pure Validated Design (PVD) means that Pure has integrated and validated our leading-edge storage technology with an industry-leading application solution to simplify deployment, reduce risk, and free IT resources for business-critical tasks. The PVD process validates a solution and provides design consideration and deployment best practices to accelerate deployment. The PVD process assures the chosen technologies form an integrated solution to address critical business objectives. This document provides design consideration and deployment best practices for the PVD using Pure Storage® FlashArray//X for high-performance SQL Server with business continuity across two data centers: FlashArray//C for capacity-optimized and economical database backup, and Pure Cloud Block Store for disaster recovery with Microsoft Azure cloud.

Introduction

Solution Overview

The production SQL Failover Cluster Instances are configured in Windows Server 2019 Virtual Machines hosted on a stretched vSphere Cluster distributed between two on-premises data centers. This uniform ActiveCluster configuration enables ESXi hosts through MPIO to switch paths in the case of a FlashArray failure. The SQL user data itself is protected by application-consistent snapshots taken at an interval as required to meet the customer SLA for the Recovery Point Objective. These snapshots replicated from the FlashArray systems participate in synchronous ActiveCluster via asynchronous replication to the Pure Cloud Block Store instance in Azure and the FlashArray//C40.

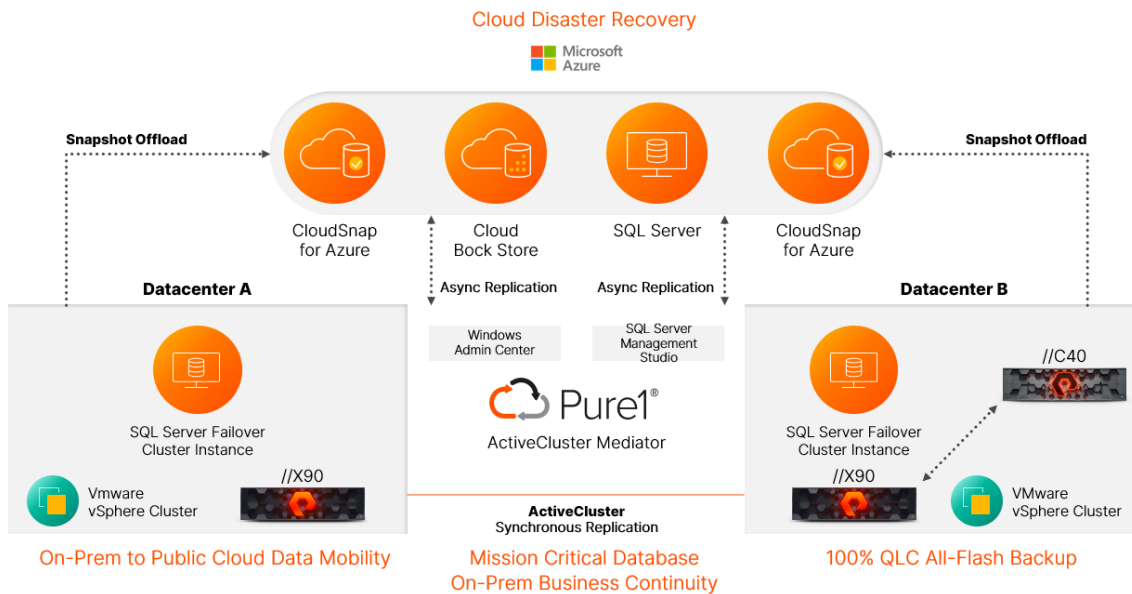


Figure 1. Pure Storage solution overview.



Test/Dev environments running in Azure virtual machines can connect to the asynchronous copy of the production data, which is refreshed at a set interval. Additional Azure VMs can be quickly brought online through automation in the event of a disaster in the on-premises ActiveCluster to provide availability to the production databases.

The FlashArray can offload snapshots of Pure volumes to external storage targets in addition to the asynchronous replication connections. Potential offload targets include the Pure Storage FlashBlade®, third-party NFS appliances, generic Linux servers providing NFS services, and cloud storage targets in the Microsoft Azure, Amazon Web Services, or Google Cloud platforms. This design implements offload to blob storage in Microsoft Azure.

Solution Benefits

This reference architecture combines high availability features from both Pure Storage and Microsoft infrastructure solutions. It's designed to leverage them in an architecture that spans multiple on-premises locations as well as the cloud. The primary benefit of implementing this architecture is the ability to survive a wide variety of failure scenarios while keeping SQL Server workloads available without interruption to business.

Pure Storage ActiveCluster provides for recovery point objective (RPO) and recovery time objective (RTO) of zero with its ease of implementation and synchronous replication capability. The Pure1 hosted mediator gives customers an easy way to deploy the mediator at a third site, utilizing an independent failure domain from the primary data center locations without having to own or lease a third physical site.

Microsoft Server Failover Clustering implemented in conjunction with VMware vSphere virtualization provides a means to ensure high availability and scalability of the clustered SQL Server role, with several safeguards to ensure that compute resources are available and optimized.

Microsoft Azure and Pure Cloud Block Store are leveraged to expand data availability outside the physical data center's bounds. The combination provides an even greater degree of resilience and flexibility and enables seamless data mobility for other hybrid cloud use cases, including dev/test and cloud migration.

Terminology

This document uses certain abbreviations in place of a full feature or service name, including the following:

- **AC:** ActiveCluster
- **AD:** Active Directory
- **FlashArray SSMS:** FlashArray SQL Server Management Studio extension
- **MPIO:** Multipath input output
- **NVMe:** Non-Volatile Memory Express
- **NVMe-oF:** Non-Volatile Memory Express over Fabrics
- **Pgroup:** Protection Group
- **pRDM:** Physical Raw Device Mapping
- **PVD:** Pure Validated Design
- **RPO:** recovery point objective
- **RTO:** recovery time objective
- **RTT:** Round-Trip-Time
- **SCM:** storage-class memory
- **SLA:** service-level agreement
- **SQLFCI:** SQL Server Failover Cluster Instance
- **SSMS:** SQL Server Management Studio
- **VSS:** Volume Shadow Copy Service
- **WAC:** Windows Admin Center
- **WSFC:** Windows Server Failover Cluster



Technology Overview

Pure FlashArray//X90 Storage

FlashArray is the world’s first 100% all-flash end-to-end NVMe and NVMe-oF array, ideal for the most demanding enterprise performance requirements. It’s part of a Modern Data Experience™, delivering breakthroughs in speed, simplicity, flexibility, and consolidation. FlashArray is ideal for departmental to large-scale enterprise shared-storage deployments, high performance, and mission-critical applications. In a world of fast, pervasive networking, ubiquitous flash memory, and evolving scale-out application architecture, Pure Storage’s FlashArray provides customers with both networked and direct-attached storage in a single, shared architecture. With latency as low as 150 μs, FlashArray brings new performance levels to mission-critical business applications and databases.



Figure 2. FlashArray//X90.

From entry-level to enterprise workloads, FlashArray//X lets your organization accelerate your most critical applications. FlashArray//X delivers breakthroughs in performance, simplicity, and consolidation. It’s ideal both for enterprise applications such as Microsoft SQL Server and cloud-native, web-scale applications. The FlashArray//X70 and //X90 support optional DirectMemory™ Cache, which uses Intel Optane storage class memory (SCM) to run database workloads at near-DRAM speeds. If extreme performance is a top priority, your organization can rely on FlashArray//X to deliver the low latency and high throughput end users demand.

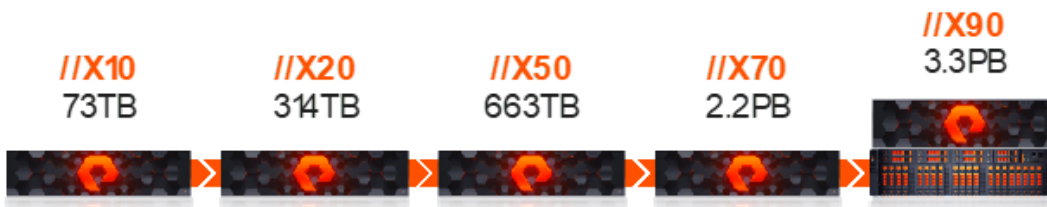


Figure 3. FlashArray delivers low latency and high throughput.



Purity for FlashArray (Purity//FA 6)

The Pure Storage Purity operating environment is the software-defined engine of Pure Storage FlashArray™. Purity is the driver that enables Pure FlashArray products, powering FlashArray//X to deliver comprehensive data services for your performance-sensitive data-center applications, and FlashArray//C for your capacity-oriented applications. Purity’s core technologies provide the speed, agility, and intelligence needed to simplify everything in your production environment. Its features set the pace for next-generation shared accelerated storage, from enterprise data services for all workloads to proven FlashArray 99.9999% availability and 10:1 total efficiency. And with the Pure Evergreen™ ownership model, your Pure as-a-Service™ subscription includes new array features and improvements to Purity via non-disruptive upgrades. Purity implements communication protocols and delivers a rich set of data services across all Pure FlashArray systems. Features including ActiveCluster for business continuity and disaster recovery, QoS, vVols, NVMe-oF, Snap to NFS, Purity CloudSnap™, DirectMemory Cache, and EncryptReduce™ are all examples of valuable new features provided with non-disruptive Purity upgrades. All Purity storage services, APIs, and advanced data services are built-in and included with every array. These technologies drive the next-generation performance and industry-leading resiliency of Pure solutions.

Pure Cloud Block Store

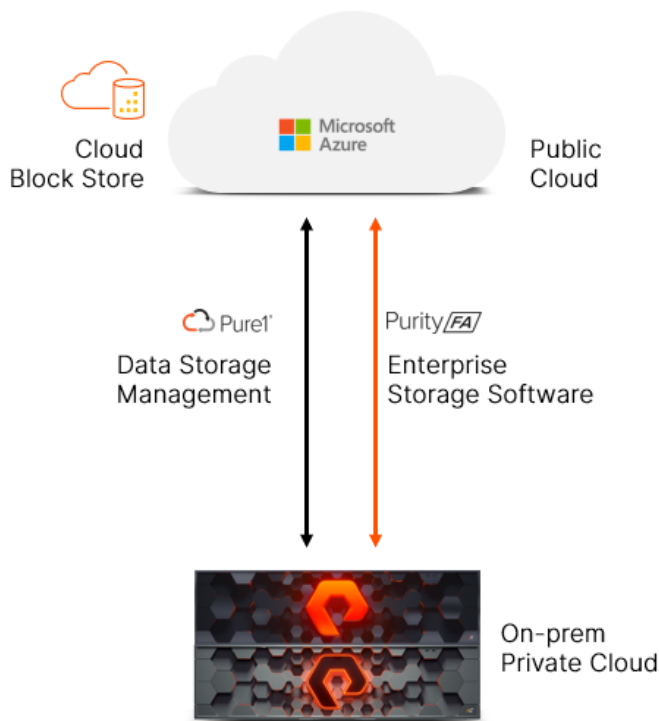


Figure 4. Pure Cloud Block store, AWS, and Azure.

Pure Cloud Block Store enhances the data resiliency in the cloud. When you deploy mission-critical applications, you need to ensure that your applications and data are resilient to single points of failure. For the ultimate level of data protection and business continuity, you can replicate data between on-premises and cloud, cloud availability zones and regions, and even between cloud providers. Read more about our [Pure Cloud Block Store](#).



Purity CloudSnap

CloudSnap provides a highly cost-effective solution for archive use cases that you can leverage in DR use cases. FlashArray customers can use CloudSnap to send snapshots of their volumes to the Azure Blob store, Amazon S3, or Google Cloud. These immutable snapshots are self-contained with the meta-data needed to restore onto any other FlashArray or Pure Cloud Block Store instance.

In a DR event where the primary site is inaccessible, customers can deploy a new Cloud Block Store instance on-demand and restore their CloudSnap snapshots. Once the CloudSnap snapshots are fully restored on the Pure Cloud Block Store instance, customers can attach the volumes to the application VMs and resume application services. This DR alternative provides a lower cost option for customers who have a higher RTO/RPO tolerance. Since volumes are restored from the local cloud storage, the RTO will largely depend on the amount of data that you need to restore.

Pure FlashArray//C40 Storage

Pure Storage FlashArray//C lets you consolidate workloads and simplify storage with consistent all-flash performance at a lower TCO than hybrid storage.



Figure 5. Pure FlashArray//C40 Storage.

Most business-critical applications today run on high-performance all-flash storage arrays, but capacity-oriented applications still suffer from management complexity, inconsistent performance, and lack of modern data services of hybrid flash and legacy disk systems. FlashArray//C provides a 100% NVMe all-QLC foundation for capacity-oriented applications, test and development workloads, [multi-site disaster recovery](#), and [data protection](#) at hybrid storage economics. Scale [up to 7.3 PB](#) effective storage in just three- to nine-rack units. Maximize results and flexibility for high-capacity applications on-premises and easily connect to the cloud. With [Pure Evergreen](#), you can upgrade performance, capacity, and features over time without disruption. With Pure, get storage as a service (STaaS) for on-premises and public cloud. Efficiently operate a hybrid cloud by unifying your environment with one subscription and one set of storage services.

Pure Storage SSMS extension

The Pure Storage FlashArray Management Extension for Microsoft SQL Server Management Studio (FlashArray SSMS Extension) is a database backup and restore extension for Microsoft SQL Server Management Studio (SSMS). The extension supports creating application consistent snapshots using the Pure Storage Volume Shadow Copy Service (VSS) hardware provider. The extension can be used from the SSMS GUI or automated using the included Pure Storage Backup SDK Windows PowerShell module. Database Administrators (DBAs) can manage backup and restore operations on local and remote databases from a centralized SSMS deployment.

The FlashArray SSMS Extension will appear in the SQL Server Management as a top-level menu item and an individual database context menu that is accessible by right-clicking on the database.

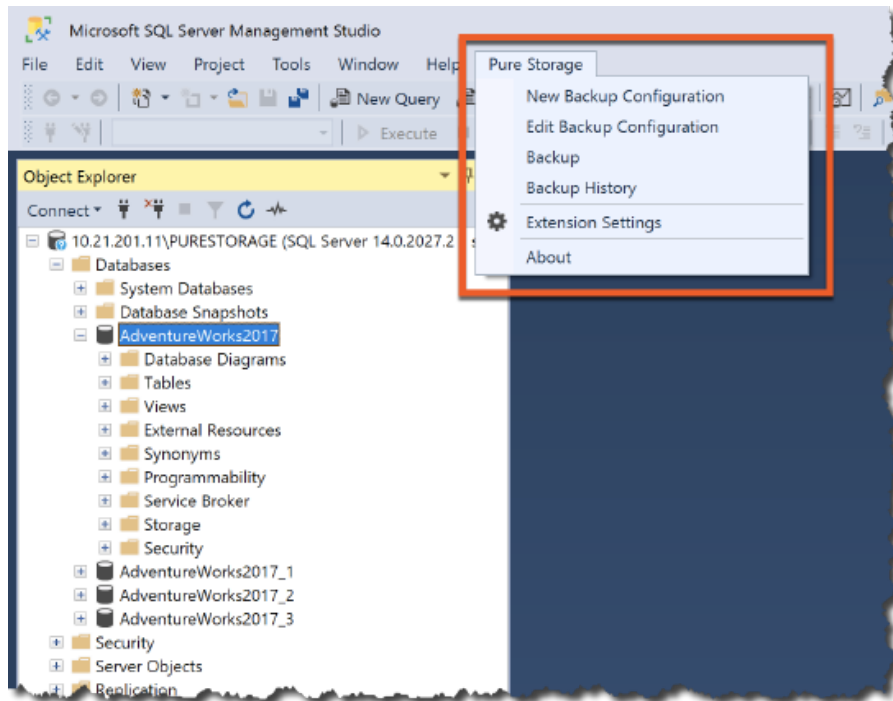


Figure 6. Pure Storage in the Microsoft SQL Server Management Studio.

Pure1®

The Pure1 cloud-based management, analytics, and support platform, expands the self-managing, plug-and-play design of Pure all-flash arrays with Pure1 Meta™ machine learning predictive analytics and continuous scanning to enable an effortless, worry-free data platform.

Monitor: In the cloud IT operating model installing and deploying management software is an oxymoron: you simply log in. Pure1 is SaaS-based, allowing you to monitor your Pure array from any browser or the Pure1 Mobile App – with nothing extra to purchase, deploy, or maintain. From a single dashboard, you can see all your arrays, with full visibility on the health, performance, and capacity of your storage.

Plan: The Pure1 workload planner delivers accurate performance and capacity forecasting for up to 12 months in the future. This gives you complete visibility into the growth of your applications. We give you advanced capabilities for new application deployment planning, so you can see exactly what you need for performance and capacity to support your business. With simulations you can easily clone, grow, or migrate workloads and then see exactly how that will impact your data center.

Act: Not only does Pure1 help you monitor your arrays and plan for the future, it also makes it easy to act on those decisions. The Service Catalog in Pure1 lets you easily subscribe to the data services you need, all from one location. We make it easy for you to automate too, with the Pure1 API letting you take your analysis and monitoring to any of your other tools.

Support: Pure1 doesn't stop at helping you plan for the future. Meta, the foundation of Pure1, makes sure your appliances are healthy, up-to-date, and free from issues. Pure1 Meta tracks the health of all installed Pure appliances and fingerprints known



issues. Pure then proactively fixes over 70% of all known issues. It's just one of the reasons our world-class Pure Support team gets the industry's highest NPS ratings, and a key component in our proven FlashArray 99.9999% availability.

VMware ESXi Cluster

Pure Storage supports VMware vSphere Storage APIs Array Integration (VAAI) to provide:

- Block zero feature to accelerate virtual disk initialization
- Full copy to offload virtual machine copying and deployment
- Space reclamation to ensure efficient use of capacity
- Hardware-assisted locking to enable large-capacity datastores with an enhanced scale of simultaneous operations

In addition to general integration with VMware virtual machines, Pure Storage also has several packaged solutions available on the VMware marketplace, including:

- [FlashArray vSphere Client Plugin](#) gives native management of FlashArray storage directly within the vSphere client.
- [Pure1 VM Analytics](#), which provides visibility of the I/O path from VM to FlashArray.
- [FlashArray Storage Replication Adapter](#) integrates with VMware vCenter Site Recovery Manager (SRM) to provide native array replication and protection policies for simplified disaster recovery and automated data protection.
- [Pure Storage Management Pack for VMware vRealize Operations](#), which provides a comprehensive data-rich view of your entire storage array, including dependency hierarchical maps, heat maps for capacity and performance, and dashboards/reporting.

As a [VMware Advanced Technology Alliance Partner](#), Pure Storage works closely with VMware to help customers maximize the synergy between their virtual and physical infrastructure. Pure supports [VMFS datastores](#), [Virtual Volumes \(vVol\)](#), and [Raw Device Mappings \(RDM\)](#) in both virtual (vRDM) and physical (pRDM) compatibility modes. In the deployment section, we discuss why we chose pRDM for the user databases. Refer to the release notes of your specific version of Purity or with your local account team to determine any limitations that may apply.

Technical Solution Design

Software Versions

Table 1 details the specific versions of software used to create this validated design.

Software	Version
Purity	6.1.2
Windows Server	2019 Datacenter Version 1809
SQL Server	2019 Version 15.0.2000.5
VMware ESXi	7.0U1
VMware vCenter Server	7.0U1

Table 1. Software used to create this Pure Validated Design.

Architecture

- FlashArray//X90R2 using Purity 6.1.1 in an ActiveCluster configuration utilizing Ethernet ports for replication in a uniform configuration. VMware ESXi 7.0U1 was configured on (4) Windows Server 2019 nodes split with (2) nodes in each site. (4) Windows VMs, one per ESXi node, are configured in a Microsoft SQL Server 2019 Always On Failover Cluster Instance.
- Pure Cloud Block Store for Microsoft Azure was configured as an asynchronous replication target for user SQL Server databases.
- For the Pure Cloud Block Store instance, a V20MU-R1 instance size was deployed, and all networking was within the same resource group and Virtual Network (vNet) in Azure.
- The on-premises Pure Storage FlashArray//C40 is configured as a second asynchronous replication target for user SQL Server databases.
- Pure CloudSnap for Microsoft Azure is configured as an offload target on the FlashArray//X90R2 arrays.
- Azure Virtual Machines utilizing a load balancer for SQL FCI.
- SQL Server Management Studio Extension enables Application Consistent snapshots both on-premises and in Azure.
- Pure Storage Management Extension for Windows Admin Center configured for Resource Management of on-premises FlashArrays and Pure Cloud Block Store for Azure instances.

On-Premises Business Continuity Architecture

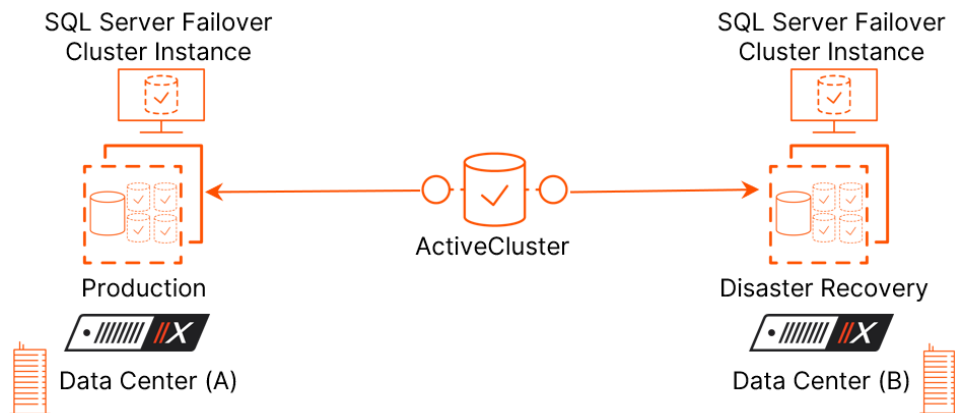


Figure 7. On-Premises Business Continuity Architecture

- Two data center (A & B) implementations using ActiveCluster replicating using Ethernet (IP).
- Four ESXi 7.0U1 Servers in a cluster; (2) in each data center A & B.
- Two SQL FCI, one per node in Data Center A. We will document how easy it is to have more than one SQL FCI per node, and if appropriate how you can have SQL FCI on nodes in both sites. We will build one SQL FCI per node in Data Center A.
- TPCC databases that have been created with [HammerDB](#).

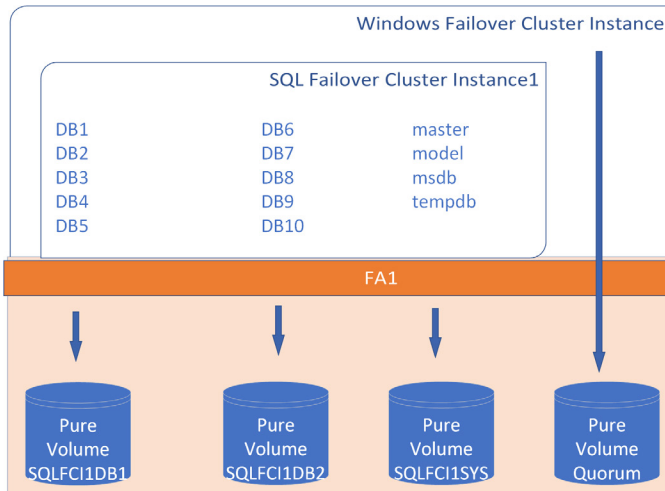


Figure 8. Failover cluster instances.

- Four virtual machines, one VM per physical ESXi host. The VM System Disk will be a VMDK on a VMFS datastore. The VMs will not utilize site failover by leveraging VMware [anti-affinity](#), and the SQL Instances instead will failover as appropriate. The Failover Cluster Quorum and SQL FCI disks will be Raw Device Mapping volumes operating in physical compatibility mode (pRDM).
 - Each SQL FCI will have 10 DBs split between two pRDM. The SQL FCI system databases will be isolated on their own pRDM to encourage easy application-consistent snapshot and replication of just the user databases. This is a common database layout, and we will discuss why we have this layout and how changing the number of volumes will impact data protection in the Deployment section.

FlashArray1 Prod				FlashArray2 DR			
POD1							
BOOTVMFS	WINQUORUM	SQLFCI1DB1	SQLFCI1DB2	SQLFCI1SYS	SQLFCI2DB1	SQLFCI2DB2	SQLFCI2SYS
SQLVM1Boot.vmdk	Quorum files	DB1	DB6	master	DB11	DB16	master
SQLVM2Boot.vmdk		DB2	DB7	model	DB12	DB17	model
SQLVM3Boot.vmdk		DB3	DB8	msdb	DB13	DB18	msdb
SQLVM4Boot.vmdk		DB4	DB9	tempdb	DB14	DB19	tempdb
		DB5	DB10		DB15	DB20	

Figure 9. FlashArray database layout.

ActiveCluster

Components

Purity ActiveCluster comprises the following three core components:

Pure1 Cloud Mediator: A required solution component provided for free in Pure1 and used to determine which array will continue data services should a replication outage occur in the environment. (also known as “split brain”). As an alternative, you can install a local mediator on a virtual machine.

Active-Active Clustered FlashArrays: Utilize synchronous replication to maintain a copy of data on each array that presents as a single consistent copy to hosts that are attached to either, or both, arrays.

Stretched Storage Containers: Management containers that collect storage objects such as volumes into groups that are stretched between two arrays.

Uniform and Non-Uniform Host Access

Hosts can be configured to see just the array local to it, which is uniform access, or both arrays which are non-uniform access.

Uniform Access

You can use a uniform storage access model in environments where there is host-to-array connectivity of either Fibre Channel or Ethernet (for iSCSI), and array-to-array Ethernet connectivity, between the two sites. When deployed in this way, a host has access to the same volume through both the local and the remote array. The solution supports connecting arrays with up to 11ms of round-trip time (RTT) latency between the arrays.

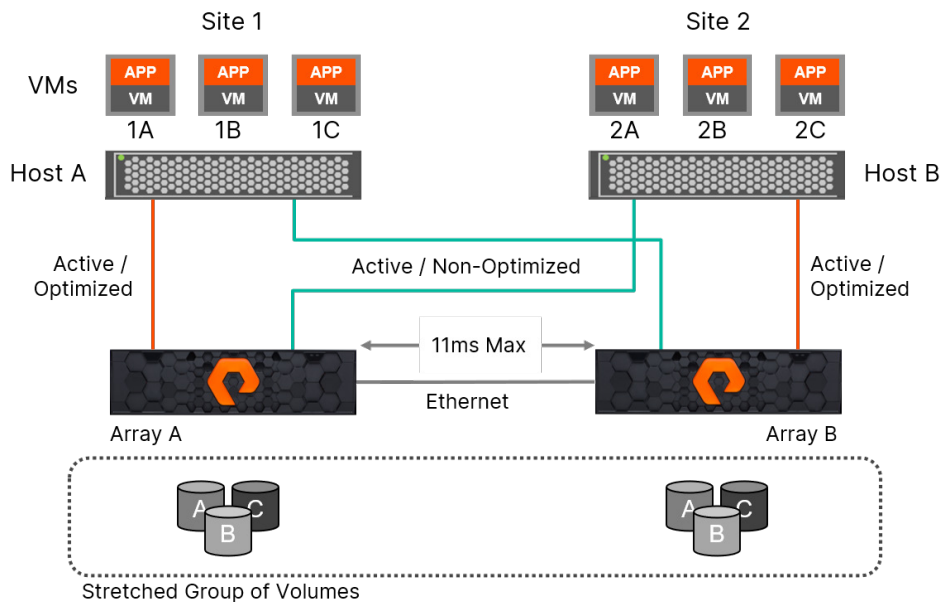


Figure 10. Uniform Access RTT latency maximum increased to 11ms with Purity 5.1.2.

Figure 10 represents the logical paths that exist between the hosts and arrays, and the replication connection between the two arrays in a uniform access model. Because a uniform storage access model allows all hosts—regardless of site location—to access both arrays, the paths will have different latency characteristics. Paths from hosts to the local array will have lower latency; paths from each local host to the remote array will have higher latency. For the best performance in an active-active synchronous replication environment, prevent the hosts from using paths that access the remote array unless necessary. For example, in Figure 10, if VM 2A were to perform a write to volume A over the host side connection to array A, that write would incur twice the latency of the inter-site link, once for each traverse of the network. The write would experience 5ms of latency for the trip from host B to array A plus 5ms of latency while the array asynchronously sends the write back to array B.

There are no such management headaches with Purity ActiveCluster. ActiveCluster makes use of Asymmetric Logical Unit Access (ALUA) to expose paths to local hosts as active/optimized paths. It exposes paths to remote hosts as active/non-optimized. There are two advantages to the ActiveCluster implementation.

- ActiveCluster volumes in stretched pods are read/write on both arrays. There is no such thing as a passive volume that cannot service both reads and writes.
- The optimized path is defined on a per host-to-volume connection basis using a preferred-array option on the host object in each FlashArray. This ensures that the host will have a local optimized path to that volume regardless of the VM or



application on which it's running. Ensure that hosts in the uniform configuration have the preferred path set on the FlashArrays in both sites.

With ActiveCluster you may configure truly active-active data centers. You don't have to care on what site or host a VM runs. The VM will always have the same performance regardless of the site. While VM 1A is running on host A and accessing volume A, it will use only the local optimized paths as shown in the image.

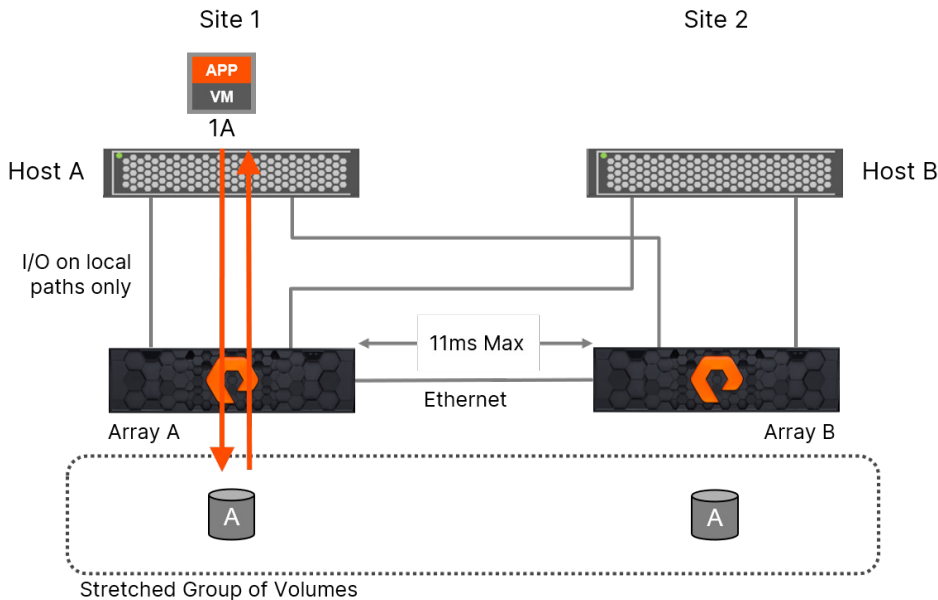


Figure 11. Site 1 Server reads are on the optimized path within the site on Array A.

If the VM or application is switched to a host in the other site with the data left in place, it will use only local paths in the other site. There is no need to adjust path priorities or migrate data to a different volume to ensure local optimized access.

Non-Uniform Access

A non-uniform storage access model is used in environments where there is host-to-array connectivity of either Fibre Channel or Ethernet (for iSCSI) locally only within the same site. Ethernet connectivity for the array-to-array replication interconnect must still exist between the two sites. When deployed in this way, each host has access to a volume only through the local array and not the remote array. The solution supports connecting arrays with up to 11ms of round-trip-time (RTT) latency between the arrays. Hosts will distribute I/O across all paths to only the storage because only the local Active/Optimized paths are available (Figure 12).

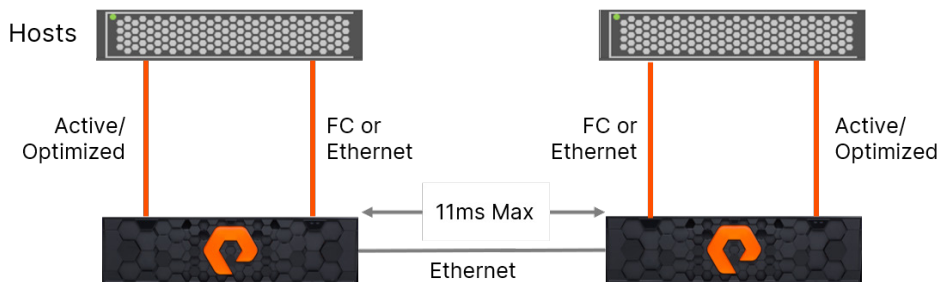


Figure 12. In a non-uniform configuration, the hosts only see the local storage.



VMware ESXi 7.0U1

This design utilized ESXi 7.0U1 and vSphere 7.0U1 to stretch an ESXi cluster across two data centers utilizing Pure ActiveCluster in a uniform configuration. The subnet is stretched across both data centers, and each ESXi host can communicate with both FlashArrays. Each Host is assigned a preferred array, to prevent an ESXi host from directly reading and writing to the remote FlashArray for best performance. This configuration provides additional availability because the SQL FCI will not failover in the event of a local FlashArray outage. The VM will switch to the non-preferred path and continue operating against the remote array. Environments that cannot stretch the network subnet or VLAN would need to isolate VMs in each physical site or use external orchestration to change the networking configuration should the VM failover to another site. See [SQL Server Multi-Subnet Clustering](#) for details on how to enable a SQL FCI to failover to a different subnet.

In this design one SQL FCI VM was placed on each ESXi node, and 2 SQL FCI are deployed to the 4 VM WSFC. Both SQL FCIs are online in 'Datacenter A' and have one Preferred Owner in each site. Using vSphere HA and enabling [Host Monitoring](#) will ensure that an ESXi failure will allow VMs to failover. Enable [anti-affinity](#) to prevent more than 1 VM in each site from being brought online on the same ESXi node unless there are multiple failures. In a site outage, each SQL FCI will failover to a different VM in the other site. As the solution scales it utilizes both [anti-affinity](#) and [DRS](#) to ensure enough compute is available for all VMs that may be brought online on a particular node during both site failures, node failures, and multiple failures as required by the service level agreement (SLA).

ActiveCluster supports moving a running VM from one site to another. This option is less common because of networking complexity, and the longer Recovery Time Objective (RTO) that is caused by having to boot the entire VM. It is more common and far quicker to simply move the SQL Server role to another host in the cluster which will take ownership of the shared storage resources assigned to the role and bring SQL Server online.

The number of VMs participating in a WSFC is only limited by the operating system limits, but Microsoft recommends [smaller node sizes](#). Ensure that the number of VMs can handle the number of deployed SQL FCIs while meeting the service level agreement (SLA) for failures and performance. For example, if the SLA requires no performance impact during a site failure and N-2 in the surviving site, then all SQL FCIs need to be able to run in the VMs that can run in either site, in the event of both a remote site failure and 2 hypervisor failures. Ensure that there is enough compute to handle the SQL workload to meet the business requirements.

SQL Server Failover Cluster Instance (SQLFCI)

Microsoft Windows Server Failover Clustering (WSFC) provides cluster-based monitoring of clustered roles running on Windows Server hosts, even across multiple subnets. In the event of a failure of the storage, network, or host, the remaining Servers coordinate to restart the affected clustered roles on surviving hosts. Through a variety of network and storage-based heart-beating, the servers can detect and respond to various failure or isolation events to provide the fastest automated recovery of clustered roles. Virtual machines (VMs) that are configured as highly available show up as roles in Failover Cluster Manager and are protected from server issues because the VM will be failed over to another node in the cluster.

Multiple identical SQL Servers can be configured as an Always On Failover Cluster Instance, which is a SQL Server role in WSFC. This cluster role runs in a WSFC resource group with one or more WSFC nodes. When a WSFC starts, a [quorum is established](#), and then cluster resources can be brought online. When the SQL FCI starts, one of the nodes assumes ownership and brings the SQL Server instance online. When a failover occurs, automatic or manual, the SQL FCI will automatically start on another node of the cluster.



There are many available strategies for providing high availability to the SQL Server instance. In this configuration, each virtual machine was pinned to a particular ESXi node with the Possible Owners setting, and the VM is prevented from failing over. In the event of an outage at the physical server or storage layer, the VM would go down, and the SQL FCI would failover to a surviving node in seconds to minutes. It is much faster for the SQL role to failover to another node, than to wait for a VM to failover and completely boot before the SQL service can start. Larger deployments, where more than one SQL FCI VM is placed on each ESXi node can use a balance of [Preferred Owners, and Possible Owners](#) in the Failover Cluster Manager to provide a balance of server and site resiliency. An example would be to place all the SQL VMs in site 1 as Preferred Owners and then add those Preferred Owners and to a larger set of Possible Owners to include nodes in site 2. Careful selection of nodes to include in site 2 would reduce the reliance on automatic VM resource balancing tools such as [DRS](#) and VM [anti-affinity](#) settings. Another popular option is to provision an active/passive model where the SQL FCI runs in one site, and the entire site fails over manually through either continuous (Pure ActiveDR), or synchronous (Pure ActiveCluster) storage replication. Strategies that involve moving the VM to another site with a different subnet usually require orchestration to facilitate the network changes. With SQL Server, additional network IP Address resources can be added to the SQL Server role in an 'OR' dependency to enable the SQL instance to failover to nodes that are in a different subnet. See Microsoft's [SQL Server Multi-Subnet Clustering](#) document for more details.

VMware Virtual Machine Disk Layout

Most of the past storage designs were influenced by performance and capacity limitations of legacy hardware. Spinning hard disk latency and throughput was impacted by both the workload and RAID type. FlashArray is orders of magnitude faster than legacy SANs and carving up a SAN into many RAID arrays and carefully separating the Operating System, Page File, TempDB, User Databases, and User transaction logs are no longer necessary to provide adequate Performance. There are many ways to provision storage for a SQL FCI for [multi-host clustering](#) on a Pure FlashArray such as; iSCSI in the VM, VMDK in a Data Store, physical raw device mapping (pRDM), and virtual volume (vVol). The type of clustered disk, whether it is single or multi-host clustering, and the version of ESXi all influence what VMware supports. This multi-host clustering solution has deployed VMware ESXi 7.0U1 and VMware supports vVol, pRDM, and for the first time VMDK disks. The Pure Storage ActiveCluster solution supports both VMDK and pRDM on Purity 6.1.1 and the decision was made to implement pRDM so that Microsoft VSS application-consistent snapshots of the SQL Server databases could be implemented with the Pure Storage SQL Server Management Studio extension with no additional license costs. Customers that already have a backup application in place must carefully evaluate the supported disk type for the version of SQL Server, VMware and Purity deployed.

The operating system, paging file, and SQL Server binaries for all VMs were placed in a VMDK on a VMware VMFS datastore on the BOOTVMFS Pure Volume in the Pod. Designs that asynchronously replicate the VM such as Pure's asynchronous pgroup replication or ActiveDR, usually exclude transient data such as the paging file and SQL TempDB.

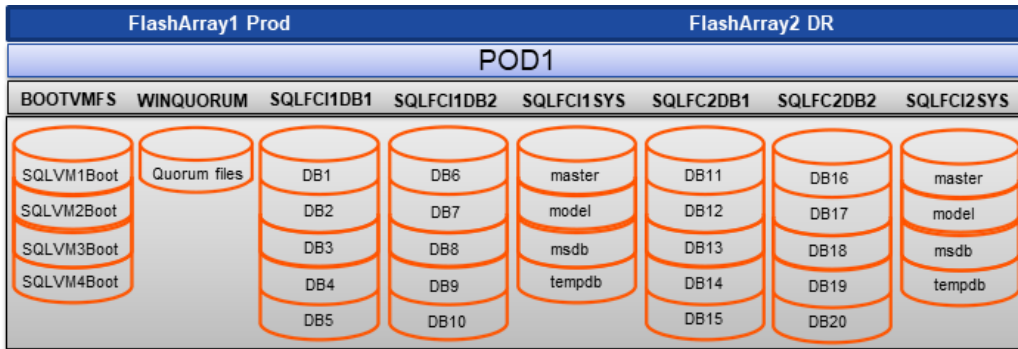


Figure 13. The layout of OS file, binaries, and SQL Server database data files to disk.

The following Pure Volumes were exposed to the Host Group, added as pRDM disks to the four VM’s in the WSFC, and then added as cluster disk resources.

The WSFC Disk Witness is a small disk that contains some cluster configuration information and is used to add a vote to the quorum of the cluster which is designed to prevent split-brain scenarios where a subset of nodes cannot communicate with each other. This design leverages the Node Majority with Witness, and this disk is the witness, using the WINQUORUM Pure Volume. Other quorum configuration options can be found on the [configure and manage quorum](#) page.

Each SQL FCI is provisioned with three Pure volumes for the system and user databases. The system disk is the selected destination during the installation of the SQL FCI and by default is where the system databases including TempDB are found. These Pure Volumes are SQLFCI1SYS and SQLFCI2SYS each owned by the respective SQL FCI. If excessive writes are written to a SQL FCI’s TempDB, it would be recommended to isolate the TempDB to a Pure volume that is not in the Pod, and thus not synchronously replicated. This will cut down on TempDB write latency, and since it is recreated at instance startup, it is not necessary to replicate this database to the other data-center. The cost of this performance increase is more disks to manage as a TempDB disk would have to be allocated outside of the synchronously replicated POD, to each Hypervisor. [TempDB on local disk](#) for SQL FCI has been supported since SQL Server 2012.

The Pure Volumes SQLFCI1DB1 and SQLFCI1DB2 were provisioned for the first SQL FCI, and SQLFCI2DB1 and SQLFCI2DB2 were provisioned for the second SQL FCI and each hold five of the user databases. There are no performance considerations on the FlashArray side that force a particular disk layout. All the user databases, including the system databases, and their transaction logs could have been placed on a single Pure Volume. VMware has performance considerations due to the Queue Depth placed on disks, and their SCSI controllers. For design considerations that will influence the number of pRDM disks and the number of PVSCSI controllers see VMware’s KB: [Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values](#). Application consistent backup through VSS and the Pure Hardware Provider is volume-based, meaning the entire Pure Volume is involved in a snapshot. That means that a VSS restore is unlikely to be implemented if more than 1 database is co-located on the same volume since VSS restore will overwrite the live volume with the chosen point-in-time snapshot. If a SQL FCI has many databases, isolating each one to a separate Pure Volume adds to the management cost of the solution as there would be many more disks to manage. Disaster recovery options for the replication of non-production environment user database volumes are covered later in this document. It is a best practice to isolate the user databases from the system databases.



Azure Virtual Machine Disk Layout

The SQL FCI running on Azure Virtual Machines has connected storage from both Azure Shared Premium SSD Disk and an instance of the Pure Cloud Block Store. The Azure SQL FCI was configured with similar parameters and settings for the disk layout as the previous section on the VMware Virtual Machine Disk Layout. As VMware disks and Azure Managed Disks have differences in selection and configuration, not all parameters would apply to all environments.

- Each storage instance contained the SQL Server database data files and transaction logs for each database test.
- The Premium SSD shared disk was configured to a P50 disk SKU size with maxShares = 2 as there are two nodes in the cluster.
- Each virtual machine is connected via iSCSI to the Azure Pure Cloud Block Store instance and the on-premises FlashArrays.
- The volumes were formatted as NTFS with a 64k allocation unit size, sized at either 1 or 2TB in size, and configured as an available Cluster Disk.

Data Protection

Pure FlashArray Level

The Pure FlashArray has several inherent features designed to protect the integrity of user data and provide [proven high availability](#) of that data.

RAID-HA

Pure Storage's proprietary RAID scheme is used to protect against three flash failure modes: device failure, bit errors, performance variability. RAID-HA is an adaptive form of RAID that provides a minimum of dual parity protection, self-heals when an SSD has failed or is removed, automatically increases the parity protection on highly reduced data sets, and performs priority-based RAID reconstruction that begins with the least protected data residing in the RAID set.

Pure FlashArray Snapshots

Snapshots of live data have become an expected storage array feature. Users employ them for everything from creating stable backups of moving data to starting points for disaster recovery, testing, and development. Over time, snapshot technology has evolved to share common content with parent volumes and to produce independent writable clone volumes. Clones are particularly useful in that they enable realistic test and development scenarios.

FlashArray snapshots are unique because:

- They are immutable. Administrators can destroy snapshots and eradicate their content, but if a snapshot exists, its content never changes.
- They are only visible to array administrators. Clients cannot mount them as read-only devices. They are administration objects, not a special type of read-only volume.

FlashArray snapshots are the basis for cloned volumes. Array administrators create clones by copying snapshots to new or existing volumes. Snapshot immutability means that every volume created from a given snapshot starts with identical content. For example, all the tests that use volumes that are created from a particular snapshot will start out with identical contents.



FlashArray volumes are organized into protection groups (pgroups) for snapshot purposes. Pgroup snapshots are atomic—all volumes in a group are snapped at the same logical instant. This allows administrators to place data sets on volumes solely according to application requirements. Making a pgroup of an application's volumes guarantees that snapshots are atomic for all its data.

Array administrators manage snapshots of entire pgroups. It is not possible, for example, to destroy a snapshot of a single volume in a multivolume pgroup. One can, however, clone a single volume, or copy a snapshot over its parent volume to roll back the volume's contents. The latter can be used to recover from data corruption. By rolling back volume contents to a snapshot that predates a data-corrupting event, integrity can be restored. Valid operations can then be replayed to recover work lost on account of the event.

FlashArray snapshots are fast. Taking one amount to create a few persistent data structures, so the elapsed time is comparable to that of a typical client write (less than a millisecond). Administrators can therefore use snapshots liberally, for example, to create frequent application recovery points, or to perform incremental backups while applications operate on live data. These facilities may be constrained, or even unavailable, with other snapshot technologies.

FlashArray administrators can use the GUI, SDKs, or CLI to create ad hoc snapshots at any time, for example, to meet unforeseen data protection requirements, such as might precede hardware or application upgrades. More commonly, administrators schedule snapshots to be taken automatically at regular intervals. An array keeps scheduled snapshots for a retention period, after which it automatically destroys them. Once a snapshot schedule has been created, the array manages application data recovery points automatically.

By default, arrays do not destroy ad hoc snapshots automatically; unless retention is specified at creation time, they exist until explicitly destroyed. Destroying an ad hoc snapshot initiates a 24-hour eradication delay grace period during which it can be recovered. An administrator can override the grace period, and eradicate a snapshot, for example, to initiate free space reclamation if necessary.

Finally, snapshots protect against erroneous destruction of volumes. Destroying a volume implicitly creates a snapshot with the usual 24-hour eradication delay period during recovery is possible. After 24 hours the implicit snapshot is automatically eradicated, and storage unique to it becomes eligible for reclamation. An administrator can terminate the 24-hour period by explicitly eradicating the volume, which eradicates the implicit snapshot.

Pure ActiveCluster

Pure Storage introduces complete business continuity that requires no third site, no extra hardware, licenses, or fees, and no more than a few minutes to set up. Purity ActiveCluster is a multi-site, Active/Active stretch cluster that makes the highest levels of availability easy and affordable for everyone.

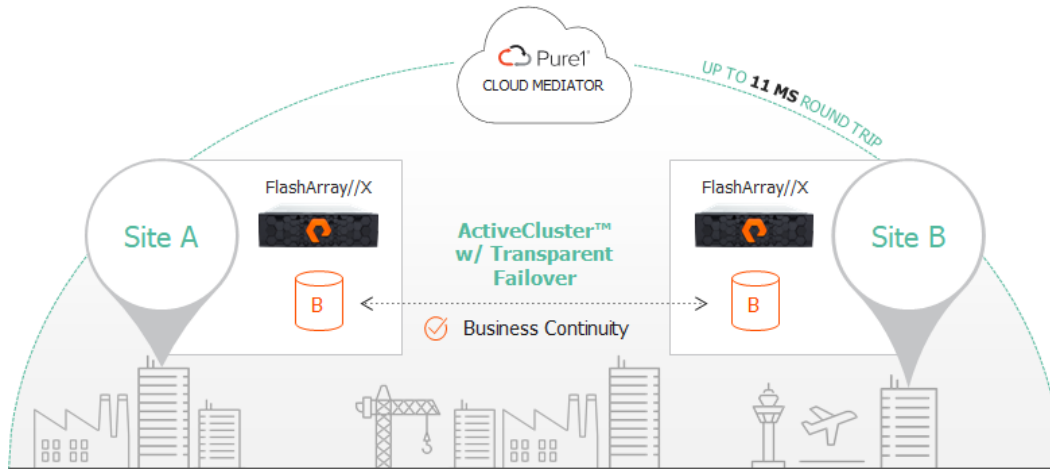


Figure 14. Pure ActiveCluster.

Active/Active for All: Purity ActiveCluster brings business continuity to mainstream Tier 1 deployments. Now, organizations can effortlessly enable zero RPO and zero RTO between two FlashArrays with true Active/Active synchronous replication for transparent failover. Unlike “Active/Active” solutions that are Active/Passive at the volume level, ActiveCluster serves reads and writes on a given volume from both sites simultaneously. You don’t have to worry about the complexity of managing VM or database instance affinity to a site. Moreover, application latency is optimized with reads served locally. ActiveCluster maintains storage efficiency with inline compression on the wire and by performing XCOPY operations, ZeroSnap snapshots, and clones efficiently across arrays.

No Third Site, No Gateways: Purity ActiveCluster is built-in to FlashArray, and comes with the Pure1 Cloud Mediator, a SaaS-based quorum witness, that eliminates the need for a third site. Failovers happen safely, automatically, and transparently. If an array or replication link fails, the Cloud Mediator ensures your hosts simply access data on the other array.

An End to Complexity and Expense: To date, conventional synchronous replication and stretch cluster solutions have been complex and extraordinarily costly, with setups lasting days to weeks and involving hundreds of pages of manuals. License fees, external gateways, and the frequent need for outside professional services added to the pain and expense. With ActiveCluster, we’ve eliminated these headaches and delivered true business continuity for all.

Set Up in Minutes: Purity ActiveCluster uses the same simple and easy storage management model as the rest of FlashArray. To enable ActiveCluster, we added just one new command. It takes only four short steps to set up: Connect the arrays, create a stretched pod, create a volume, and connect the hosts. That’s it!

Your Apps Stay Up: Your mission-critical apps – like SAP, Oracle, and Microsoft SQL Server on physical or virtualized environments such as VMware or Hyper-V – deployed on Purity ActiveCluster run seamlessly even when disaster strikes.

Bullet-Proof Availability Everywhere: With ActiveCluster, failures, and recovery are now transparent across your environment, array, data center, or network. Resync is automatic: when a link is restored, arrays automatically resynchronize using dedupe-aware async, and access to volumes is automatically restored when resync is complete



Asynchronous Replication

Pure Storage enables customers to cost-effectively deploy all-flash storage solutions in environments where traditional storage arrays can no longer meet the growing performance and efficiency requirements of today's workloads. These workloads are often the most important in an enterprise environment, requiring data protection for business continuity and disaster recovery. Most all-flash storage systems require the use of host-based replication solutions that involve complex configuration, time-consuming recovery processes, and expensive licenses that increase the overall cost of the storage solution resulting in unexpected additional costs.

The choice to implement an all-flash storage solution should not require compromises, should not add complexity to the environment, and should not increase the overall cost of the solution. To meet these needs Pure Storage provides [Asynchronous Replication](#): a license-free, array-based, data reduction-aware asynchronous replication solution that's simple to configure and manage while providing robust capabilities for enterprise workloads. Asynchronous replication is a snapshot-based asynchronous replication solution that leverages space-efficient snapshots to replicate point-in-time consistent copies of multiple LUNs. Asynchronous combines snapshots and replication into a single solution allowing you to create multiple recovery points on the replication target. The replication target can be a connected FlashArray, or it can be an Offload Target. Offload Targets can be an NFS Server, or AWS, Azure, or Google Cloud endpoint. Data is always transferred in a compressed format and written to the SSD at the target array in compressed format. Purity always stores compressed data on the SSD. Whenever data is read by a host, such as with the FC or iSCSI protocol, that data is always decompressed when sent in response to a host read request. In Asynchronous replication data always remains compressed. Compressed data is read from the source array SSDs and transferred over the wire in a compressed format where it remains compressed and is written to the destination SSDs.

This design includes asynchronous replication to a Cloud Block Storage instance in Azure, a FlashArray //C40, and CloudSnap to Azure blob storage. If you want to replicate the same volumes to different targets but require using a different schedule or retention policy per target array, then you must create a different protection group for each target that requires a different schedule or retention. Use a pgroup naming convention that helps to identify the target array in the name of the pgroup to make it easier to differentiate the pgroups in the user interface.

Application Consistent VSS Snapshot

Microsoft SQL DBAs are familiar with SQL Server Management Studio (SSMS) and the Pure Storage FlashArray Management Extension enables application-consistent snapshots of SQL Server databases. The SSMS extension leverages the Pure Storage Backup SDK PowerShell module for easy automation, and the Pure Storage VSS Hardware Provider to create VSS snapshots of volumes containing SQL Server databases.

The user databases have been isolated to two Pure volumes so that they can be replicated to Pure Cloud Block Store for test/dev, and to both a FlashArray//C40 and CloudSnap for long-term retention. Customers that rely on things that do not reside in user databases such as jobs, linked servers, and non-contained logins will need to replicate the entire VM, or they can be backed up independently in SSMS and then restored to the replicated environment.

Crash Consistent Snapshot

Pure FlashArray excels at quickly creating space-efficient crash-consistent snapshots. When a database or group of databases must be consistent, a pgroup can be utilized to ensure that the snapshots of all the volumes in the pgroup are atomic. Common use cases that do not require application-consistent snapshots include database cloning for test/dev, reporting, and even alternate database recovery. See [Cloning SQL Server Databases using FlashArray](#) for more details.



Data Migration

After the initial setup, data is synchronously replicated between the FlashArrays participating in ActiveCluster using Ethernet (IP). The user database volume snapshots are also replicated to three targets at a set interval: CloudSnap to Azure blob storage, a FlashArray //C40, and Pure Cloud Block Store in Azure. Pure snapshots are immutable and can only be used as a source to copy from. The target can be the original volume, a new volume, or overwrite an existing volume.

On the two FlashArrays that are receiving asynchronously replicated copies of the user database snapshots (Pure Cloud Block Store Azure and FlashArray//C40), user database snapshots can quickly be copied to volumes on the respective FlashArrays and be presented to SQL Servers in seconds. There are many use cases for this functionality; two of the most common are Disaster Recovery and enabling test/dev workflows outside of the production storage.

Snapshots are also asynchronously replicated to an offload target, CloudSnap for Azure, using Azure blob storage. Offloaded snapshots are often used to retain snapshots for a longer duration than the source FlashArray. These snapshots can be easily rehydrated to the source FlashArray or pushed to another FlashArray with a simple command. This means that the user database snapshots can be restored to the production ActiveCluster, pushed to a new FlashArray, or pushed to a new Pure Cloud Block Store instance in AWS or Azure.

Failure Scenarios

This reference architecture is designed to tolerate many different types of equipment and network failures while still preserving access to SQL Server data. This section will go over several of the potential failures and detail how the architecture is designed to deal with them.

Failure of a Single Array Controller

Pure FlashArrays are designed with two redundant controllers per array. If one controller should fail, access to data will be maintained through the surviving controller, and the failed controller can be replaced non-disruptively.

Failure of an Entire Array

The loss of an entire storage array is precisely the situation that ActiveCluster and Array Replication are intended to mitigate. Because data is synchronously replicated between arrays that reside in separate failure domains, or Availability Zones and Regions in the case of Pure Cloud Block Store, and hosts have volumes connected through data paths to both arrays simultaneously, the loss of an entire array can be handled non-disruptively to the hosts. Note that during normal operation, a host will prefer paths to the local array. If that array fails, the host will start to use paths to the remote array. While this is non-disruptive, the host may experience increased latency while accessing the remote array. To mitigate this, it may be desirable to fail the host workloads over to the remote location as well so that all data access remains local.

Loss of Connection to the Mediator

If one array loses connection to the mediator but the arrays can still communicate with one another, both arrays will generate alerts and replication continues normally. If the array loses connection with both the mediator as well as its peer, it is then considered isolated. The array which retains contact with the mediator will continue to serve data for the ActiveCluster pod.

If both arrays lose connection to the mediator, both arrays will stay online if they maintain a connection with one another through the replication network. If the replication network fails, a pre-elected array will stay online to continue serving data for the ActiveCluster pod.



Loss of an ESXi Host

If an ESXi host should fail, the VMware High Availability feature will attempt to restart the virtual machines previously running on that host on another host in the cluster. Cluster affinity rules have been defined so that it will prefer a host in the same location as the failed host. Meanwhile, the Windows Failover Cluster will have failed the SQL Server roles to another of the VMs used in the failover cluster.

Loss of a Node in the Failover Cluster

In this design, each of the nodes in the Failover Cluster is a virtual machine. If one of those virtual machines should fail for some reason, Windows Failover Clustering will move the SQL Server role over to a surviving member of the Failover Cluster.

Intermittent Loss of the VM Network between Sites

Microsoft Clustering by default will failover cluster resources if a node is unavailable. When stretching a cluster to multiple-sites intermittent issues could cause unnecessary SQL FCI failovers. Microsoft's Tuning Failover Cluster Network Thresholds has more details on tuning in a stretched environment.

Microsoft Failover Clustering: Physical Disk Cluster Resource Default Timeouts

By default, cluster disk resources that fail will only attempt to restart one time 0.5 seconds later. If the storage fabric has intermittent spikes in latency, this can cause cluster resources to failover unnecessarily. The [Microsoft Platform Guide](#) has guidance to change the cluster disk timeouts to provide more resiliency in a stretched cluster environment.

Loss of Replication Links Between ActiveCluster Member Arrays

Each array is configured to use multiple redundant IP interfaces for synchronous replication between arrays configured as an ActiveCluster. If one of these links is lost, replication will continue via the remaining links. If connectivity between the two peer arrays is lost entirely, they will both perform a race to the mediator. The array that contacts the mediator first is declared the winner and continues to serve data for the ActiveCluster pod.

Loss of an Entire Site

If an entire site should fail, perhaps due to a natural disaster or another significant incident that impacts the entire site, this would entail losing both one half of the ActiveCluster as well as one half of the Windows Failover Cluster. In this scenario, hosts at the remote site would continue accessing data through the local ActiveCluster member array. Windows Failover Clustering would bring up the SQL Server roles from the hosts which were lost on the surviving hosts in the surviving location.

Loss of Both Sites

While losing both sites would represent a rare and catastrophic loss of both on-premises compute and storage resources, SQL Server data would still be available in Microsoft Azure. The Pure Cloud Block Store instance will contain data as recent as the last asynchronous replication update. Volumes on the Pure Cloud Block Store could then be attached via iSCSI to Azure Virtual Machines running SQL Server.

If Pure Cloud Block Store was not initially configured, the data which was offloaded to CloudSnap for Azure can be used. A Pure Cloud Block Store array can be instantiated in minutes, and the data from CloudSnap restored to this instance.



Deployment Guide

The following sections provide detailed deployment details for the hardware, software, and cloud solutions described above. While manual configuration procedures are provided where needed, this document focuses on solution automation by also providing links to script and template repositories. The primary repository is [here](#).

Configure the FlashArray

In a uniform ActiveCluster deployment, each host, and the host group need to be created on both FlashArrays, because each ESXi host will be connected to both FlashArrays. The personality should be set to ESXi in the host object in the Purity GUI. The [FlashArray User Guide](#) can be used for detailed configuration steps.

1. Ensure the Server Fibre Channel initiator ports, and the FlashArray Fibre Channel target ports are zoned and connected.
2. Create the Host objects and set the Personality to ESXi. In the Purity UI, select the Host, then under Details, Set Personality.
3. On each Host object select Host Ports and then for Fibre Channel, select Configure WWNs. Select the appropriate FC port addresses for each host. The Port Address can be found with “Get-InitiatorPort” in PowerShell on each host.
4. Create a Host Group object and then add all the Hosts to the Host Group.
5. On only one FlashArray, create the volumes, because in the Configure ActiveCluster deployment step, these Volumes will be added to a Pod that will then be stretched across both FlashArrays. Adjust the number and size of the Volumes to meet your requirements.
 - a. 1TB for the VMFS datastore for the boot VMDK files of the SQL FCI VMs.
 - b. 10GB for the WSFC disk witness. Microsoft [recommends at least 512MB in size](#).
 - c. 20GB for each SQL FCI disk. Microsoft [recommends at least 8030MB in size](#).
 - d. 1TB for each user database disk. In our design, we place 5 databases on each user database volume.

FlashArray1 Prod				FlashArray2 DR			
POD1							
BOOTVMFS	WINQUORUM	SQLFCI1DB1	SQLFCI1DB2	SQLFCI1SYS	SQLFCI2DB1	SQLFCI2DB2	SQLFCI2SYS
SQLVM1Boot.vmdk	Quorum files	DB1	DB6	master	DB11	DB16	master
SQLVM2Boot.vmdk		DB2	DB7	model	DB12	DB17	model
SQLVM3Boot.vmdk		DB3	DB8	msdb	DB13	DB18	msdb
SQLVM4Boot.vmdk		DB4	DB9	tempdb	DB14	DB19	tempdb
		DB5	DB10		DB15	DB20	

6. On each FlashArray, select the Host Group, and add all the Volumes in the Pod.

Configure ActiveCluster

While ActiveCluster is a powerful tool that provides for business continuity of customer data through a variety of failure scenarios, deploying it is a very simple process. Please refer to the release notes of your specific version of Purity or with your local account team to determine any limitations that may apply. The high-level steps to deploy ActiveCluster are:

1. Ensure the Ethernet Ports used for ActiveCluster replication are assigned the Replication service and are configured with IP addresses that can communicate between the FlashArrays.



2. Add a connect FlashArray by getting a connection key on the target FlashArray, and creating a synchronous replication connection between the two FlashArrays
3. Create a pod.
4. Add the existing volumes to the Pod and/or create new volumes in the pod.
5. Stretch the pod between the two FlashArrays.
6. In this uniform configuration, set the Preferred Array to the FlashArray that is local to each ESXi host object in the Purity GUI. In the Purity GUI select the Host, then under Details, Add Preferred Arrays.

Detailed deployment steps can be found in the web guide entitled [Implementing vSphere Metro Storage Cluster with ActiveCluster](#).

Quickly Check That the Connectivity Between the Flasharrays Is Balanced

1. Login to the Purity GUI on each FlashArray.
2. Select Health, then Connections, and view the connected Array Connections.

sn1-x90r2-e06-31		● connected		Ethernet (IP)			
Remote Array		Local Array		CT0		CT1	
		ETH4: 10.21.190.113	ETH5: 10.21.190.114	ETH4: 10.21.190.115	ETH5: 10.21.190.116		
CT0	ETH4: 10.21.190.123						
	ETH5: 10.21.190.124						
CT1	ETH4: 10.21.190.125						
	ETH5: 10.21.190.126						

Configure ESXi Cluster

The VMware Native Multipathing plugin will automatically be configured for Pure Storage Volumes with Round Robin and an I/O operation limit of 1. For more information see [ESXi Host Configuration](#) in the [VMware Platform Guide](#).

A remote Syslog server is important for troubleshooting ESXi issues and is not enabled by default. For more information see [ESXi Host Configuration](#) in the [VMware Platform Guide](#).



Ensure NTP is configured for the same time servers on both the [ESXi hosts](#) and the FlashArrays. See the [FlashArray User Guide](#) for a walkthrough.

1. Login to the Purity GUI on each FlashArray.
2. Select Settings, and then System.
3. Under Array Time, the NTP settings can be edited and tested.

Quickly check that the connectivity between the ESXi hosts and the FlashArrays is balanced. This can be found in the [ESXi Host Configuration](#) in the [VMware Platform Guide](#).

1. Login to the Purity GUI on each FlashArray.
2. Select Health, then Connections, and view the connected Host Connections.

Hardware Alerts Connections Network

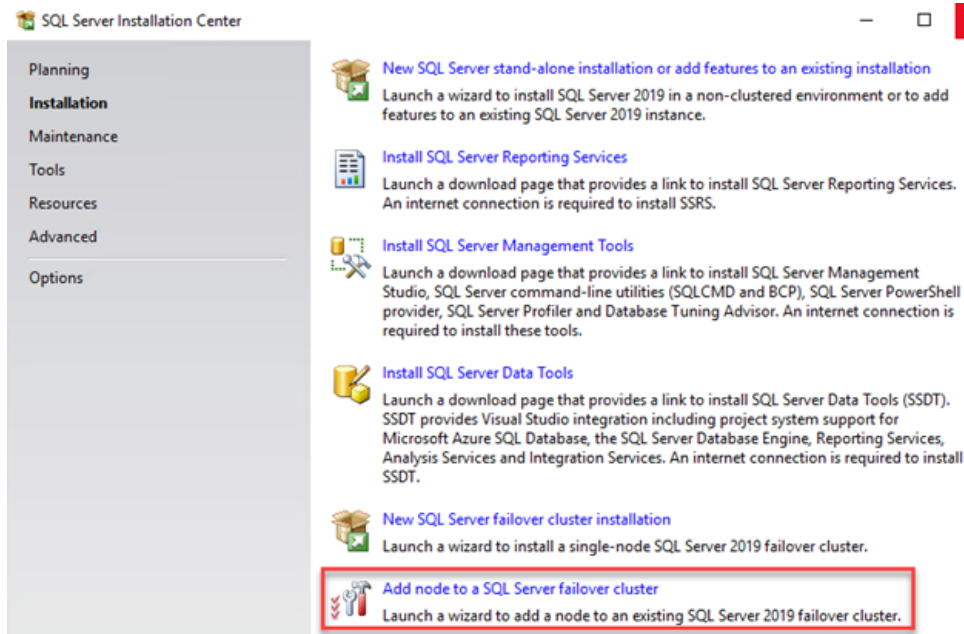
Host	# WWN	# IQN	# NQN	Paths	CT0	CT1
sn1-r720-e06-01	2	0	0	None	None	None
sn1-r720-e06-03	2	0	0	None	None	None
sn1-r720-e06-05	2	0	0	Redundant	1 1	1 1
sn1-r720-e06-07	2	0	0	Redundant	1 1	1 1
sn1-r720-e06-09	2	0	0	Redundant	1 1	1 1
sn1-r720-e06-11	2	0	0	Redundant	1 1	1 1

Configure SQL VMs

Create a Two-Node WSFC and SQL FCI

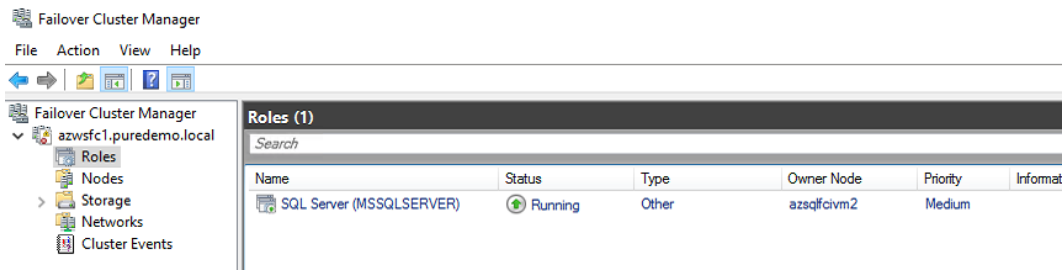
1. Be sure to meet the [prerequisites that are stated for WSFC installations](#). You should at a minimum ensure that:
 - a. You have a unique static IP address (recommended) and name for each cluster node.
 - b. You have a static IP address (recommended) and name for the cluster.
 - c. Both VMs have close to or identical Windows Updates applied.
 - d. You have access to Active Directory administrative consoles and DNS.
2. Using the vSphere client, create two virtual machines and install Windows Server 2019 on each of them.
 - a. Allocate CPU and memory resources as needed to suit your workload
 - b. Set the SCSI controller type to VMware Paravirtual and ensure that SCSI bus sharing is set to physical mode.
3. Join the VMs to the Active Directory domain.
4. Install the Windows Failover Cluster feature on each of the VMs.
5. Create the WSFC. Open the Failover Cluster Manager on the first node and follow this [Microsoft documentation](#) for detailed installation instructions.
6. Create the shared quorum disk witness.

- a. Create a volume on the FlashArray which is at least 512MB in size.
 - b. Connect the newly created volume to the ESXi cluster
 - c. Add the volume as a new RDM disk to the first of the two VMs comprising the WSFC, being sure to connect it to the paravirtual SCSI interface.
 - d. On the second VM in the WSFC, add an Existing Hard Disk and select the VMDK file which represents the RDM disk added to the first VM.
7. Now configure the quorum disk on both cluster nodes.
- a. On each VM, open Administrative Tools - Computer Management - Disk Management.
 - b. You should receive a prompt to online and set the partition table for the new disk. Choose MBR or GPT.
 - c. Right-click on the disk and select Create Simple Volume. Create the volume, do not assign a drive letter, and format it as NTFS. You may also set the allocation block size to 64k which is recommended for SQL Server workloads.
 - d. Open the Failover Cluster Manager tool and add the disk as a quorum witness.
 - e. Click on the cluster name and in the Actions pane, select More Actions - Configure Cluster Quorum Settings...
 - f. Click *Next* - Select the *quorum witness* - *Configure a disk witness* - check the available disk - *Next* - *Finish*.
8. Install SQL Server 2019 on the first node of the cluster using the “Advanced cluster preparation” option in the Advanced menu of the SQL Server installer. As part of the installation, the SQL installer will validate the cluster. Make sure no critical errors exist.
9. On the second node, install SQL Server using the “Add node to a SQL Server failover cluster” option in the Installation menu of the installer.





- Once both nodes are successfully installed, there should now be a new *Role* in the Failover Cluster Manager called *SQL Server (MSSQLSERVER)*.



- Click on the “Dependencies” tab and add the Cluster Disk, specifying “AND” as the dependency type. The SQL Server FCI is now complete and ready for use.

Adding Shared Storage to the SQL FCI

- Create a volume on the Pure array and connect it to the host group representing the vSphere cluster. Make note of the LUN ID and volume serial number, which can be used to identify the storage device within vSphere.
- Using the vSphere client, locate the newly created volume within the storage devices section. If the device is not seen, rescan the storage adapters. Repeat for each ESXi host in the cluster.

Storage Devices

Name	LUN	Type	Capacity	Datastore	Operational	Hardware Accelera...	Drive Type	Transport
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e00013e7)	1	disk	1.50 TB	DS1	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a93707ecb5a1f7bd4d46a0001442)	1	disk	300.00 GB	Not Consumed	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e0001fbbf)	249	disk	1.00 TB	Not Consumed	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e0001fbbd)	250	disk	1.00 TB	Not Consumed	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e00017d4)	251	disk	1.00 TB	Not Consumed	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e00017d3)	252	disk	20.00 GB	Not Consumed	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e00013ea)	253	disk	1.00 TB	ISO	Attached	Supported	Flash	Fibre Channel
PURE Fibre Channel Disk (naa.624a937094f7d35083a14c8e00013e9)	254	disk	123.00 GB	Not Consumed	Attached	Supported	Flash	Fibre Channel



3. Edit the settings for the first VM in the failover cluster and select “RDM Disk” under “Add New Device”

Edit Settings | SQLFCI1VM1

Virtual Hardware | VM Options

ADD NEW DEVICE ▾

> CPU	8	▾
> Memory	16	GB ▾
> Hard disk 1	90	GB ▾
> Hard disk 2	20	GB ▾
> Hard disk 3	1024	GB ▾
> SCSI controller 0	LSI Logic SAS	
> SCSI controller 1	VMware Paravirtual	
> Network adapter 1	VM Network ▾	
> CD/DVD drive 1	Datastore ISO File ▾	
> USB xHCI controller	USB 3.1	
> Video card	Specify custom settings ▾	
> Security Devices	Not Configured	
VMCI device		
SATA controller 0	AHCI	
> Other	Additional Hardware	

Disks, Drives and Storage

- Hard Disk
- Existing Hard Disk
- RDM Disk**
- Host USB Device
- CD/DVD Drive

Controllers

- NVMe Controller
- SATA Controller
- SCSI Controller
- USB Controller

Other Devices

- PCI Device
- Watchdog Timer
- Precision Clock
- Serial Port

Network

- Network Adapter

CANCEL OK

4. Select the storage device which was previously identified.

Select Target LUN

Name	Path ID	LUN	Capacity	Drive Type	Hardware Accelerati...
PURE Fibre Cha...	/vmfs/devices/di...	254	123.00 GB	Non-flash	Supported
PURE Fibre Cha...	/vmfs/devices/di...	250	1.00 TB	Non-flash	Supported
PURE Fibre Cha...	/vmfs/devices/di...	249	1.00 TB	Non-flash	Supported

3 items

CANCEL OK

5. Edit the settings for the new hard disk and under “Virtual Device Node”, select the SCSI controller which corresponds to the Paravirtual adapter. Review and verify the settings and click “OK” to continue.



Edit Settings | SQLFCI1VM1

Virtual Hardware | VM Options

ADD NEW DEVICE ▾

> CPU	8	
> Memory	16	GB
> Hard disk 1	90	GB
> Hard disk 2	20	GB
> Hard disk 3	1024	GB
▼ New Hard disk *	1024	GB
VM storage policy	Datastore Default	
Location	Store with the virtual machine	
Sharing	Unspecified	
Physical LUN	/vmfs/devices/disks/naa.624a937094f7d35083a14c8e0001bbd	
Compatibility Mode	Physical	
Shares	Normal	1000
Limit - IOPs	Unlimited	
Virtual Device Node	SCSI controller 0	SCSI(0:1) New Hard disk
> SCSI controller 0	IDE 0 IDE 1 SCSI controller 0 SCSI controller 1 SATA controller 0	
> SCSI controller 1		
> Network adapter 1	VM Network	<input checked="" type="checkbox"/> Connected
> CD/DVD drive 1	Datastore ISO File	<input checked="" type="checkbox"/> Connected
> USB xHCI controller	USB 3.1	
> Video card	Specify custom settings	
> Security Devices	Not Configured	
VMCI device		

CANCEL OK

- Edit the settings for the second VM in the failover cluster and select "Existing Hard Disk" under "Add New Device". Browse to the location of the VMDK associated with the RDM created for the first VM and select that VMDK.

Select File

Datstores	Contents	Information
▼ DS1 > .sdd.sf > vSphere-HA SQLFCI1VM1 > SQLFCI1VM2 > vCLS (1) > vCLS (2) > vCLS (3) > VMware vCenter Server > W2019-Template > ISO > vmimages	SQLFCI1VM1.vmdk SQLFCI1VM1_1.vmdk SQLFCI1VM1_2.vmdk SQLFCI1VM1_3.vmdk	Name: SQLFCI1VM1_3.vmdk Size: 1 TB Modified: 12/08/2020, 1:33:01 PM Encrypted: No

File Type: Compatible Virtual Disks (*.vmdk, *.disk, *.raw)

CANCEL OK



- Edit the settings for the new hard disk and under "Virtual Device Node", select the SCSI controller which corresponds to the Paravirtual adapter. Review and verify the settings and click "OK" to continue.

Edit Settings | SQLFCI1VM2
✕

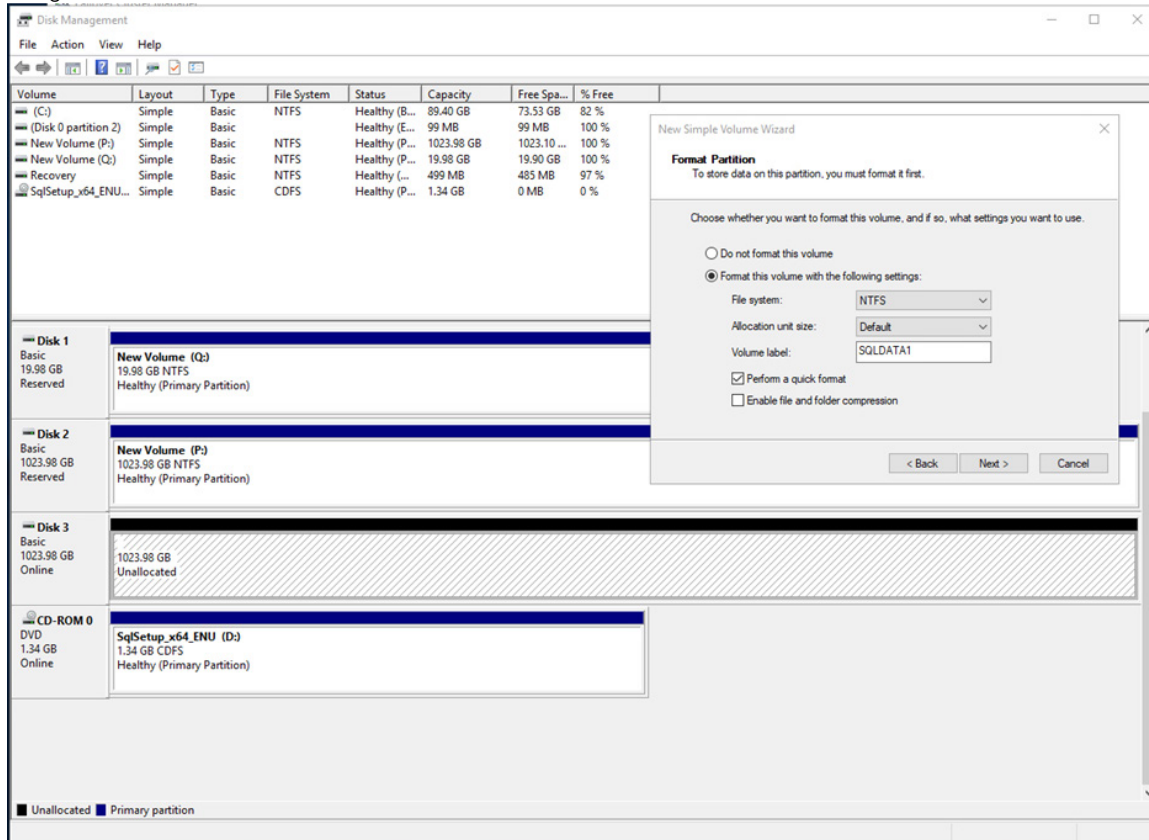
Virtual Hardware
VM Options

[ADD NEW DEVICE ▾](#)

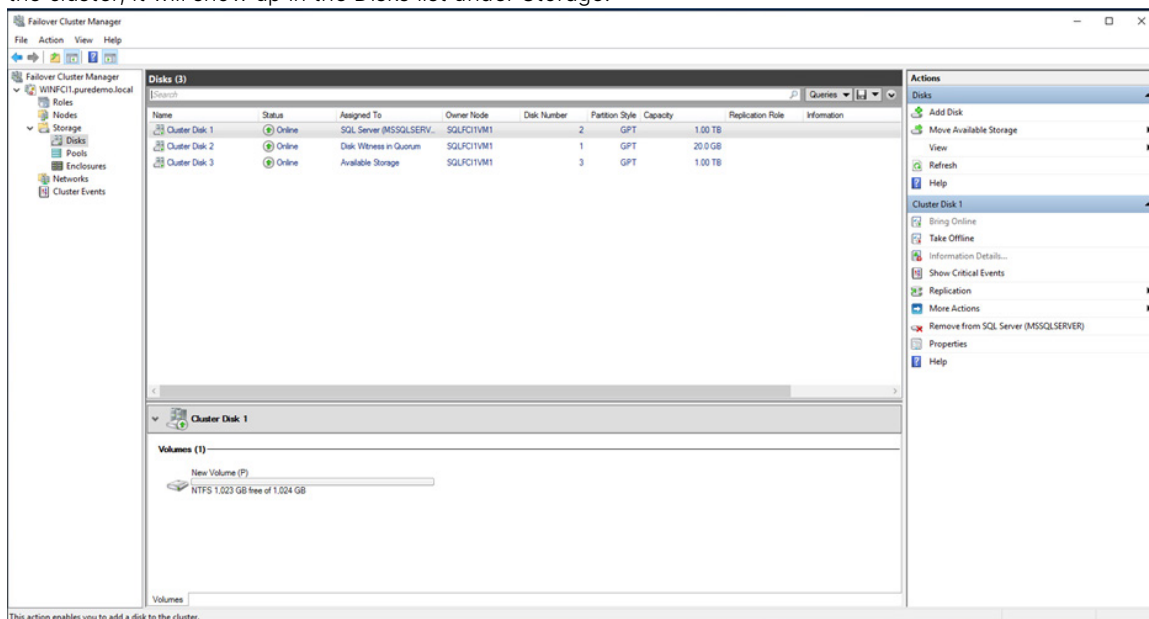
> CPU	8	▼	ⓘ
> Memory	16	GB	▼
> Hard disk 1	90	GB	▼
> Hard disk 2	20	GB	▼
> Hard disk 3	1024	GB	▼
▼ New Hard disk *	1024	GB	▼
VM storage policy	Datastore Default ▼		
Location	Store with the virtual machine ▼		
Sharing	Unspecified ▼		
Shares	Normal ▼	1000	
Limit - IOPs	Unlimited ▼		
Virtual Device Node	SCSI controller 1 ▼	SCSI(1:2) New Hard disk ▼	
> SCSI controller 0	LSI Logic SAS		
> SCSI controller 1	VMware Paravirtual		
> Network adapter 1	VM Network ▼	<input checked="" type="checkbox"/> Connected	
> CD/DVD drive 1	Datastore ISO File ▼	<input checked="" type="checkbox"/> Connected	
> USB xHCI controller	USB 3.1		
> Video card	Specify custom settings ▼		
> Security Devices	Not Configured		
VMCI device			
SATA controller 0	AHCI		
> Other	Additional Hardware		

CANCEL OK

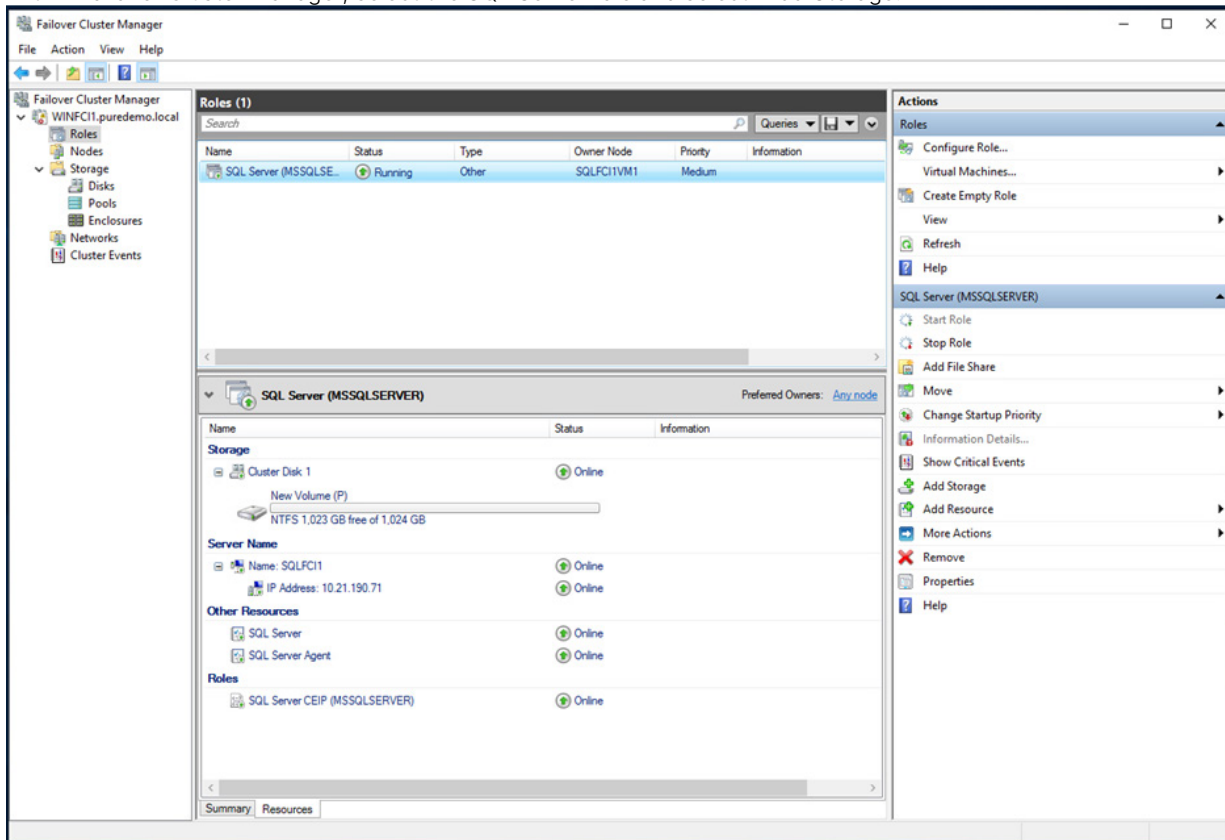
- Open Disk Management on the first node of the failover cluster. Bring the new disk online, initialize it, create a partition, and assign a drive letter.



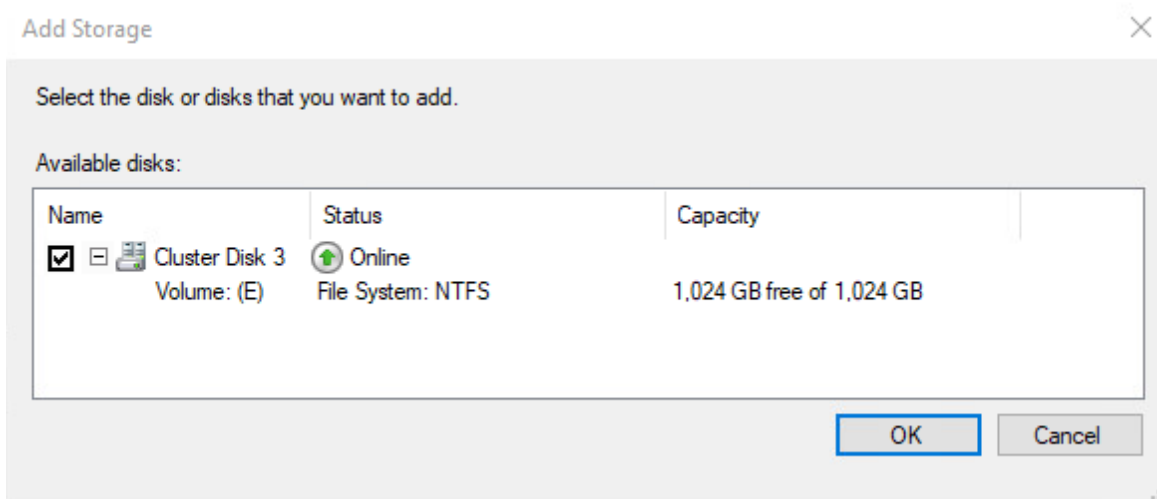
- Within Failover Cluster Manager, select Add Disk and add the newly created volume to the cluster. After adding the disk to the cluster, it will show up in the Disks list under Storage.



10. Within Failover Cluster Manager, select the SQL Server role and select "Add Storage."



11. Select the Cluster Disk which was created in the previous steps and click OK.

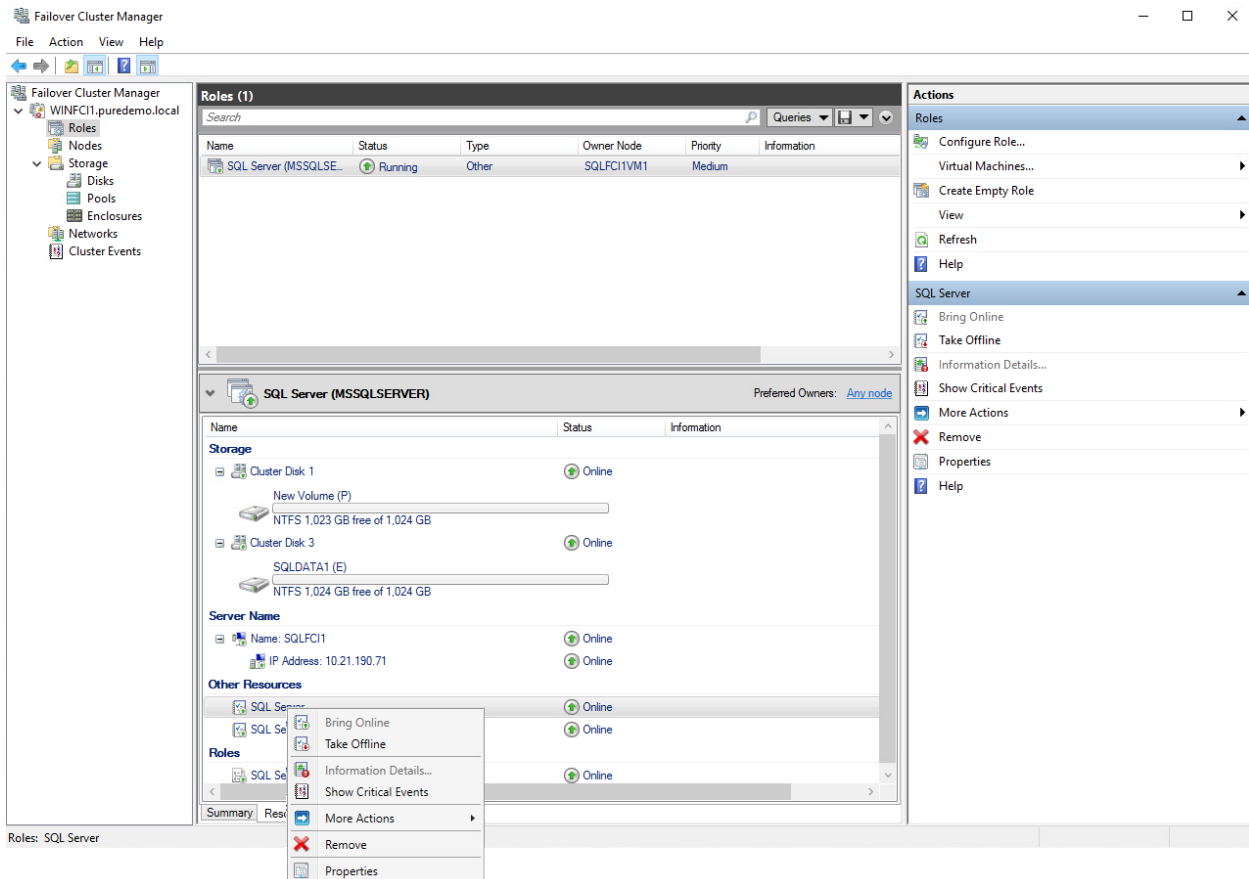


- The Cluster Disk must be added as a dependency to the SQL Server role. Until this is done, trying to create databases on this Cluster Disk will fail with an error like the following:

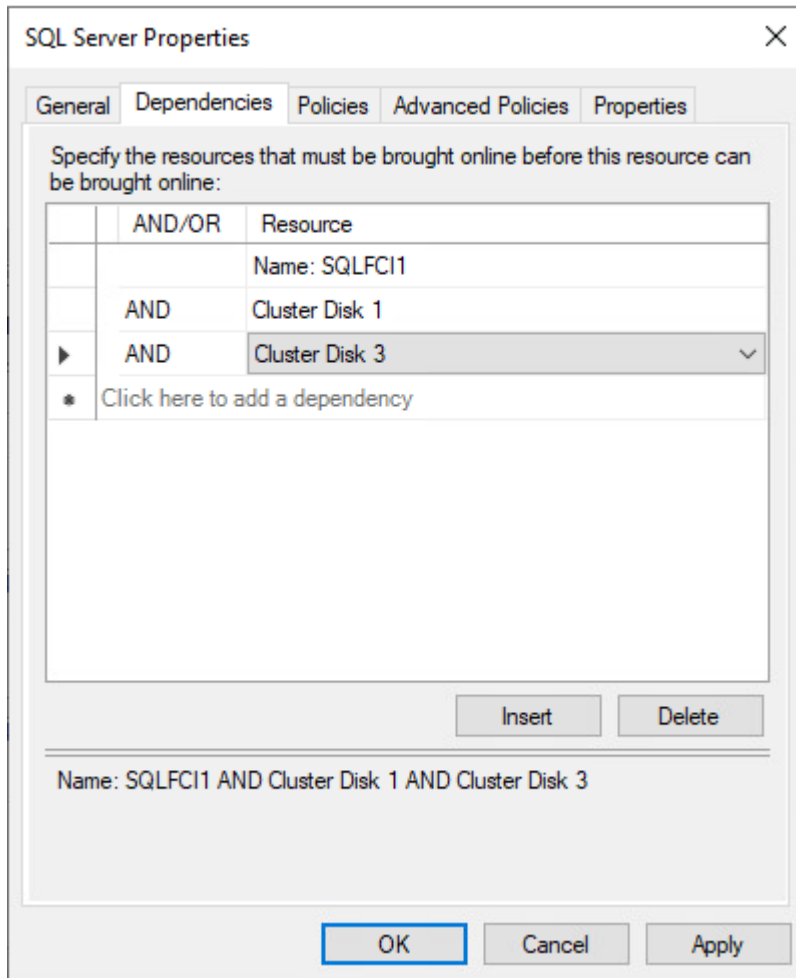
```

Administrator: Windows PowerShell - SQLCMD
PS C:\Users\administrator.PUREDEMO> sqlcmd
1> use master;
2> go
Changed database context to 'master'.
1> create database puredemo
2> on
3> ( NAME = puredemo_dat,
4> FILENAME = 'E:\MSSQLDATA\puredemo.mdf',
5> SIZE = 10,
6> MAXSIZE = 50,
7> FILEGROWTH = 5 )
8> LOG ON
9> ( NAME = puredemo_log,
10> FILENAME = 'E:\MSSQLLOG\puredemo.ldf',
11> SIZE = 5MB,
12> MAXSIZE = 25MB,
13> FILEGROWTH = 5MB );
14> GO
Msg 5184, Level 16, State 2, Server SQLFCI1, Line 1
Cannot use file 'E:\MSSQLDATA\puredemo.mdf' for clustered server. Only formatted files on which the cluster resource of the server has a dependency can be used.
Either the disk resource containing the file is not present in the cluster group or the cluster resource of the Sql Server does not have a dependency on it.
Msg 1802, Level 16, State 1, Server SQLFCI1, Line 1
CREATE DATABASE failed. Some file names listed could not be created. Check related errors.
1>
    
```

- Right-click the SQL Server entry under “Other Resources” and select “Properties.”



- Click on the “Dependencies” tab and add the Cluster Disk, specifying “AND” as the dependency type.



- Repeat as necessary to add additional disks.

Configure Pure Cloud Block Store

The configuration of the Pure Cloud Block Store for Azure is very similar to configuring a FlashArray on-premises in your data center, with some features and services that are either altered to work with the Public Cloud or removed due to not being necessary or adapted for that environment.

The installation of Pure Cloud Block Store is done through the Azure Marketplace by use of an Azure Resource Manager (ARM) template. Before and as part of the ARM process, some specific prerequisites and parameters must be created and defined to ensure a successful deployment. Please consult the [Pure Cloud Block Store Deployment and Configuration Guide for Azure](#) for more detailed configuration and pre-requisite information that should be configured during or before the deployment such as shown in the screenshot below. [The deployment could also be done using a PowerShell or Azure CLI script for automation.](#)



Create Pure Cloud Block Store™ (Product Deployment)

Basics Network Tags JIT Configuration Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource group * ⓘ
[Create new](#)

Instance details

Region * ⓘ

Cloud Block Store Model * ⓘ

Zone * ⓘ

Array Name * ⓘ

Company Domain Name * ⓘ

License Key * ⓘ

Alert Recipients ⓘ

SSH Key for 'pureuser' login ⓘ

Managed Application Details

Provide a name for your managed application, and its managed resource group. Your application's managed resource group holds all the resources that are required by the managed application which the consumer has limited access to.

Application Name *

Managed Resource Group * ⓘ ✓

[Review + create](#) [< Previous](#) [Next : Network >](#)

Figure 15. Pure Cloud Block Store Deployment

Automating the Pure Cloud Block Store Deployment

You could automate the deployment of the Pure Cloud Block Store by using the Azure CLI and our supplied template files. You can find the information on how to do this deployment in the [Pure Cloud Block Store Deployment Guide for Azure using Azure CLI](#) document, and the script and template file examples are located in [this GitHub repository](#) for your use.



Post-Deployment Configuration

1. Determine the management floating IP address of the Pure Pure Cloud Block Store instance by clicking on *All Services - Managed Applications* - and selecting the name of your Pure Cloud Block Store deployed managed application. Once that Azure portal blade is visible, then select *Parameters and Outputs* and identify the *floatingManagementIP* field as shown in the figure below.

fasolndbsqlpvdmappp | Parameters and Outputs
Managed application

Search (Ctrl+/) «

Overview
Activity log
Access control (IAM)
Tags

Settings

- Parameters and Outputs
- JIT Configuration (preview)
- JIT Access (preview)
- Authorizations
- Identity
- Properties
- Locks

Monitoring

- Alerts

Automation

- Tasks (preview)
- Export template

Support + troubleshooting

- New support request

_artifactsLocation	String	
_artifactsLocationSasToken	SecureString	*****
alwaysEmpty	Object	[object Object]

17 outputs

Name	Type	output
controllerimage	String	/subscriptions/
controllerimages	Object	[object Object]
directory	Object	[object Object]
ct0Name	String	azsqlpvcbs-ct0
ct1Name	String	azsqlpvcbs-ct1
ct0IPConfig	Object	[object Object]
ct1IPConfig	Object	[object Object]
floatingManagementIP	String	172.21.10.4
identityName	String	
identityRoleAssignmentId	String	
keyVaultName	String	
managementEndpointCT0	String	172.21.10.7
managementEndpointCT1	String	172.21.10.9
replicationEndpointCT0	String	172.21.10.5
replicationEndpointCT1	String	172.21.10.11
iSCSIEndpointCT0	String	172.21.10.8
iSCSIEndpointCT1	String	172.21.10.10

2. Open a browser and connect to the array via HTTPS. HTTPS is required for connectivity to the Pure Cloud Block Store UI in a public cloud environment.
3. Click on the Settings link on the left, then Network. Verify that the network configuration is accurate for your environment and that iSCSI and replication ports have been defined.



Settings ? ×

System **Network** Access Software

Fibre Channel ^

Name▲	Enabled	WWN	Speed	Services

Ethernet ^

Name▲	Enabled	Type	Subnet	Address	Mask	Gateway	MTU	MAC	Speed	Services
ct0.eth0	true	physical	-	172.21.10.6	255.255.255.0	172.21.10.1	1500		10 Gb/s	system
ct0.eth1	true	physical	-	172.21.10.5	255.255.255.0	172.21.10.1	1500		10 Gb/s	replication
ct0.eth2	true	physical	-	172.21.10.8	255.255.255.0	172.21.10.1	1500		10 Gb/s	iscsi
ct0.eth3	true	physical	-	172.21.10.7	255.255.255.0	172.21.10.1	1500		10 Gb/s	management
ct1.eth0	true	physical	-	172.21.10.12	255.255.255.0	172.21.10.1	1500		10 Gb/s	system
ct1.eth1	true	physical	-	172.21.10.11	255.255.255.0	172.21.10.1	1500		10 Gb/s	replication
ct1.eth2	true	physical	-	172.21.10.10	255.255.255.0	172.21.10.1	1500		10 Gb/s	iscsi
ct1.eth3	true	physical	-	172.21.10.9	255.255.255.0	172.21.10.1	1500		10 Gb/s	management
vir3	false	vif	-	-	-	-	1500		10 Gb/s	management

4. You should also verify the iSCSI IQNs are valid under the Health - Array Ports section.

Array Ports ^

FC Port	Name

Ethernet Port	Name
CT0.ETH2	iqn.2010-06.com.purestorage:flasharray.5091434e6f30b8c5
CT1.ETH2	iqn.2010-06.com.purestorage:flasharray.5091434e6f30b8c5

You should now complete the next steps to connect the Pure Cloud Block Store Array to the on-premises array for replication which is detailed in the following section of this PVD. You may also connect any on-premises or Azure hosts to the Pure Cloud Block Store using iSCSI connections. Note that on-premises hosts may be subject to additional IO latency when connecting to Pure Cloud Block Store via iSCSI, and egress charges may be incurred. An example of how that would appear when the hosts are connected is shown in the UI screenshot below.

Health ? ×

Hardware Alerts **Connections** Network

Host Connections ^ 16 of 6

Host▲	# WWN	# IQN	# NQN	Paths	CT0	CT1
azsqfclvm1	0	1	0	● Redundant		
azsqfclvm2	0	1	0	● Redundant		
sqfclvm1	0	1	0	● Redundant		
sqfclvm2	0	1	0	● Redundant		
sqfcl2vm1	0	1	0	● Redundant		
sqfcl2vm2	0	1	0	● Redundant		



Configure Async Replication to Pure Cloud Block Store and FlashArray//C

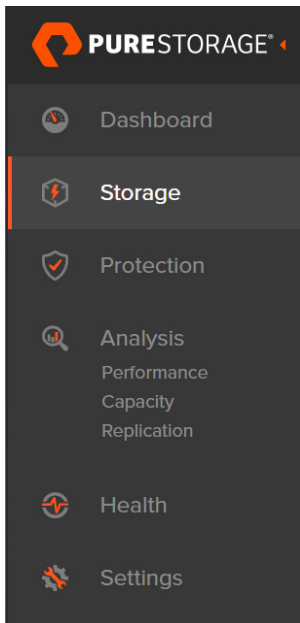
The following procedure is the same regardless of whether the target is a FlashArray or Pure Cloud Block Store. If configuring both as asynchronous replication targets, complete the following seven steps to establish replication connections for both. Then, add both targets when following the next set of steps to create a protection group.

To automate this configuration, please visit [our script and template repository](#).

Establish Replication Connections

First, create a connection between each of the on-premises FlashArrays and the target array by executing the following steps:

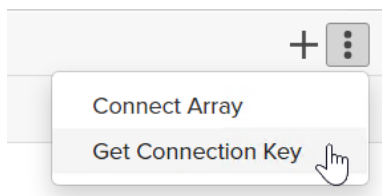
1. Log in to the web interface for both the FlashArray as well as the target array and click on “Storage” in the left-hand menu.



2. Note the Array Connections pane. Any connections which have been previously configured will display here.

Array Connections									
Name	Status	Type	Version	Management Address	Replication Transport	Replication Address	Throttled		
sn1-x90r2-e06-28	connected	async-replication	6.1.0	-	Ethernet (IP)	10.21190.113 10.21190.114 10.21190.115 10.21190.116	False		

3. On the target array, click on the ellipsis in the upper right-hand corner of the Array Connections pane and select “Get Connection Key”. Copy the connection key.



4. On the on-premises array, click the plus sign in the upper right-hand corner of the Array Connections pane to launch the Connect Array wizard.



- Enter the management address of the target array, select Async Replication, enter the connection key obtained in step 3, select Ethernet as the replication transport, and click “Connect”.

Connect Array
✕

Management Address

Type

Connection Key ⓘ

Replication Transport

Replication Address

- Verify that the connection appears in the Array Connections pane on both arrays.

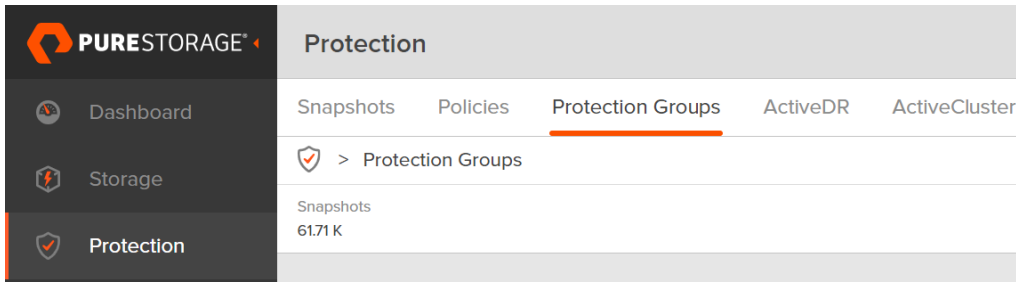
Array Connections				
Array	Status	Replication Transport		
sn1-x90r2-e06-28	● connected	Ethernet (IP)		
Remote Array	Local Array	CT0	CT1	
		ETH1: 172.21.10.5	ETH1: 172.21.10.11	
CT0	ETH4: 10.21.190.113	🟢	🟢	
	ETH5: 10.21.190.114	🟢	🟢	
sn1-x90r2-e06-31	● connected	Ethernet (IP)		
Remote Array	Local Array	CT0	CT1	
		ETH1: 172.21.10.5	ETH1: 172.21.10.11	
CT0	ETH4: 10.21.190.123	🟢	🟢	
	ETH5: 10.21.190.124	🟢	🟢	

- Repeat steps 4-6 on the second on-premises FlashArray within the ActiveCluster.

Create the Protection Group

Now that connections have been established, configure which volumes should be asynchronously replicated to the target array(s).

- In the web interface for the first FlashArray, select “Protection” from the left-hand menu pane and navigate to the “Protection Groups” section.



- In the “Source Protection Groups” pane, click the plus sign in the upper right-hand corner to open the “Create Protection Group” wizard. Select the ActiveCluster pod, provide a name for the protection group, and click “Create”.

Create Protection Group ✕

Pod

Name

- Click on the newly created protection group to bring up its properties.

Members 12 of 2

Name

- sqlfci1::SQLFC1DATA1 ✕
- sqlfci1::SQLFC1DATA2 ✕

Targets 12 of 2

Name	Allowed	
sn1-c40/3-e02-36	True	✕
sqlpvdcb1	True	✕

Snapshot Schedule ✎

Enabled: False
 Create a snapshot on source every 1 hours
 Retain all snapshots on source for 1 days
 then retain 4 snapshots per day for 7 more days

Replication Schedule ✎

Enabled: False
 Replicate a snapshot to targets every 4 hours
 Retain all snapshots on targets for 1 days
 then retain 4 snapshots per day for 7 more days

- In the “Members” pane, add the volumes to be replicated.
- In the “Targets” pane, select the remote arrays to which the volumes will be replicated. If you will be replicating asynchronously to multiple destination arrays, add each of the targets here.
- In the “Snapshot Schedule” pane, configure your desired frequency and retention policies for local snapshot recovery points.
- In the “Replication Schedule” pane, configure your desired replication schedule and remote snapshot retention.
- Optionally, create an immediate ad-hoc snapshot by clicking the plus sign in the upper right-hand corner of the “Protection Group Snapshots” pane.



Configure Azure SQL VMs

The creation and configuration of the 2-Node Windows Server Failover Cluster (WSFC) and the SQL Server Failover Cluster Instances (SQLFCI) can be done in the Azure portal UI manually, by using Azure Resource Manager (ARM) templates from the [Azure Marketplace](#) or [Quick Start Templates](#), or by using automation scripting. Pure supplies [several example scripts in our repository](#) that are available for you to adapt to your environments.

Windows Clustering Requirements in Azure

One of the primary requirements of creating a WSFC is that the VMs must be joined to an Active Directory domain. For Windows Server clusters created in the cloud, this is usually an AD domain that is present on-premises and extended into the Azure Cloud by installing a Windows Server VM, adding the Active Directory role, and promoting it to a domain controller for the on-premises domain. It is not recommended best practice to have Azure resources validate against AD using only an on-premises domain controller. However, if you cannot create this hybrid integration, you do have other options to accomplish this as well. Here are two such options:

- Utilize a separate AD domain that is hosted on Azure VMs as standalone domain controllers. This domain could then be configured for a trust relationship with the on-premises domain.
- Use Azure Active Directory services (AD-DS) to create a new domain in your Azure tenant and use the features of that service to establish the domain. You can find more information on AD-DS and configurations in [this Microsoft documentation](#).

The SQL user databases were replicated to Pure Cloud Block Store from the ActiveCluster and presented to the SQL FCI as clustered disk resources so that the FlashArray SSMS extension could be leveraged to quickly clone any databases in Azure. Cluster Shared Volumes could be used with crash-consistent snapshots if application-consistent snapshots are not a requirement.

Creating and Configuring a 2-Node SQL Server FCI using the Azure Portal

Create the WSFC and SQL FCI

1. The same [prerequisites that are stated for on-premises WSFC installations](#) are valid for instances that are created in Azure as well. You should at a minimum ensure that:
 - a. You have a unique static IP address (recommended) and name for each cluster node.
 - b. You have a static IP address (recommended) and name for the cluster.
 - c. Both VMs have close to or identical Windows Updates applied.
 - d. You have access to Active Directory administrative consoles and DNS.
2. You must decide in what region the VMs will be served from, as well as if you will want these VMs located in an Availability Set or an Availability Zone. This placement is required to utilize the UltraSSD managed disks, which are only available in select regions, as the location to hold the database data files. For this example, we will be creating an [Availability Zone](#). Availability Zones can only be specified when creating the VM and not after creation. In contrast, Availability Sets are pre-created resources in Azure. Please consult [this Azure documentation](#) before proceeding to understand the requirements and options.
3. Create a new or use an existing Active Directory domain controller or AD-DS in Azure (see above).



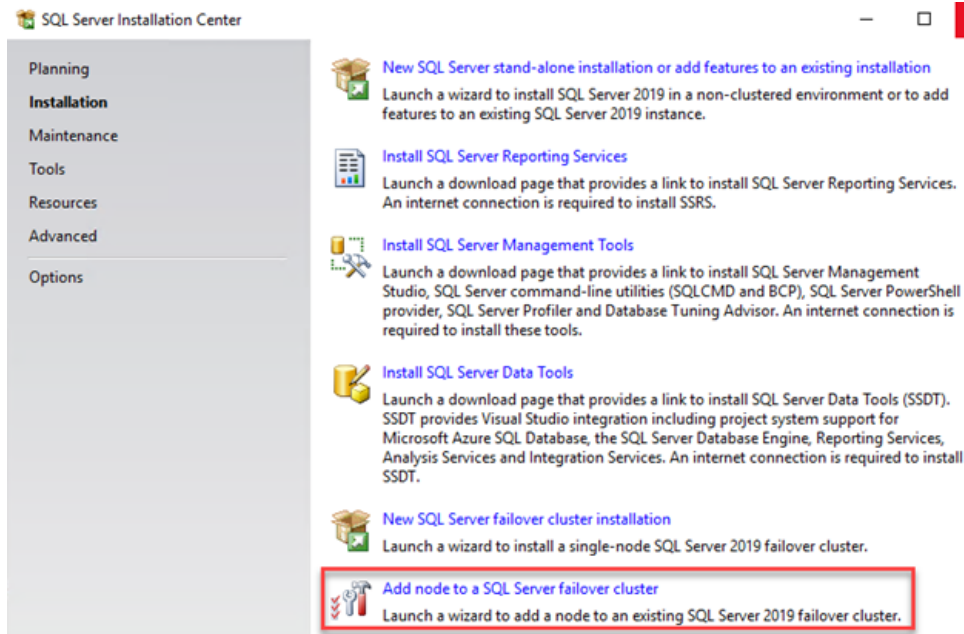
4. In the Azure Portal, create (2) Windows Server 2019 virtual machines by using the [Azure Marketplace](#) or the [Azure Quick Start](#) templates.
 - a. It is recommended to size the VMs to allow for Windows and SQL Server overhead.
 - b. Keep the default of a single OS disk of type Premium SSD.
 - c. For an Availability Zone, you must check the *High Availability* option and select a zone that is listed for the region you have selected. Remember this zone selection.

Availability options ⓘ Availability zone

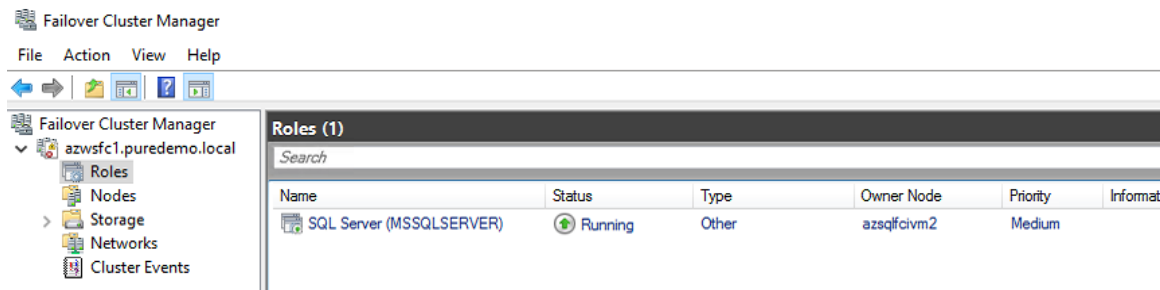
Availability zone * ⓘ

- d. Both VMs should be in the same resource group and connected to the same virtual network (vNet).
 - e. Do not assign a Public IP Address (PIP) if it is not necessary. If you must have internet access to the cluster nodes, Pure recommends enabling [Just-in-Time \(JiT\) access](#) to ensure a secure access environment.
5. Once the VMs are created, join them to the Active Directory domain.
6. Depending on which image you selected to deploy from the Marketplace, Azure creates a default Windows OS disk size. If that size is acceptable to you, then continue. If not, extend the System volume on the VM before continuing to a larger size by following [this documentation](#).
7. Install the Windows Failover Cluster and MPIO features on each server.
8. Configure MPIO and iSCSI for the VMs. You can reference these [Microsoft Platform Guide documents](#) for step-by-step configurations.
9. Create the WSFC. Open the Failover Cluster Manager on the first node and follow this [Microsoft documentation](#) for detailed installation instructions.
10. Choose a type of quorum witness to create for the cluster. In this example, we will create an [Azure Premium Shared Managed Disk](#) to be the witness.
11. Create the shared disk. Please refer to these [complete instructions](#) on creating an Azure Shared Disk. You must ensure that this disk is created in the same region, resource group, and availability zone as the VMs that will use it.
12. Once the disk is created, you can now add it to both cluster nodes.
 - a. Within the properties of each node, click on *Disks*.
 - b. Under *Data Disks*, click on *Attach existing disk* and select the shared disk you previously created. Do this for each node.
 - c. On each VM, open Administrative Tools - Computer Management - Disk Management.
 - d. You should receive a prompt to set the partition table for the new disk. Choose MBR or GPT.
 - e. Right-click on the disk and select *Create Simple Volume*. Create the volume, do not assign a drive letter, and format it as NTFS. You may also set the *allocation block size* to 64k which is recommended for SQL Server workloads.
 - f. Open the Failover Cluster Manager tool and add the disk as a quorum witness.
 - g. Click on the cluster name and in the Actions pane, select More Actions - Configure Cluster Quorum Settings...
 - h. Click Next - Select the quorum witness - Configure a disk witness - check the available disk - Next - Finish.

13. Install SQL Server 2019 on the first node of the cluster using the “Advanced cluster preparation” option in the Advanced menu of the SQL Server installer. As part of the installation, the SQL installer will validate the cluster. Make sure no critical errors exist. Also, ensure to use the default SQL Instance name of MSSQLSERVER when prompted.
14. On the second node, install SQL Server using the “Add node to a SQL Server failover cluster” option in the Installation menu of the installer.



15. Once both nodes are successfully installed, there should now be a new *Role* in the Failover Cluster Manager called *SQL Server (MSSQLSERVER)*.



16. To connect the VMs to the Pure Cloud Block Store instance, you follow the same configuration steps as you would follow if it were VMs and FlashArray on-premises using iSCSI. These steps include the following and detailed instructions can found in our Microsoft Platform Guide - Quick Setup Steps article repository:
 - a. Installing the MPIO features on the VMs.
 - b. Configuring the iSCSI service in MPIO on the VMs and obtaining the IQNs for each VM.
 - c. Creating the Hosts on the Pure Cloud Block Store instance and adding their IQN ports.



17. Create a Host Group by clicking on the “+” icon located under **Storage - Hosts - Host Groups**

Create Host Group

Name

Create Multiple...
Cancel
Create

18. Add the Hosts to a Host Group by clicking on the newly created Host Group and selecting the “+” icon under Member Hosts and select Add. On the next screen, select the boxes for each host and click on Add.

Member Hosts

Name	Interface	Size	Volumes	
				Add Remove Download CSV

No hosts found.

19. Create a volume by clicking on Storage - Volumes and selecting the “+” icon under Volumes.

Create Volume ✕

Pod or Volume Group

Name

Provisioned Size T ▼

QoS Configuration (Optional) ▼

Create Multiple...
Cancel
Create

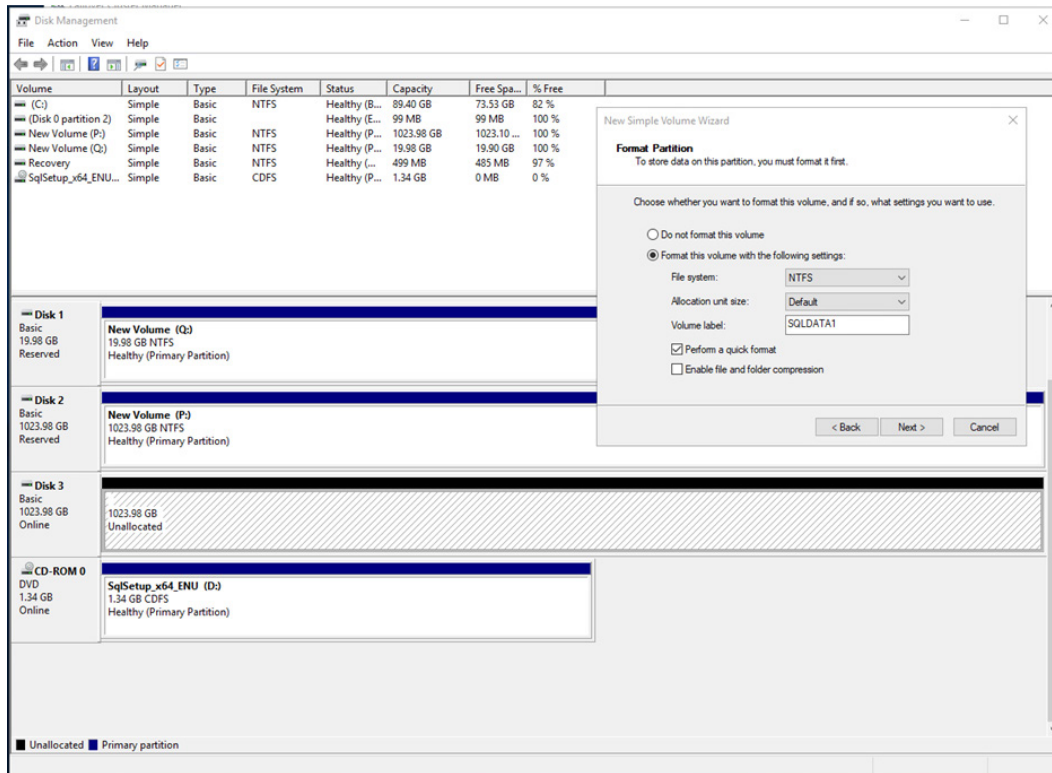
20. Connect the volume to the Host Group by clicking on the newly created Volume and clicking on the “+” icon under Connected Host Groups and click on Connect. In the next screen, click the boxes next to the newly created Host Group name and click Connect.

Connected Host Groups

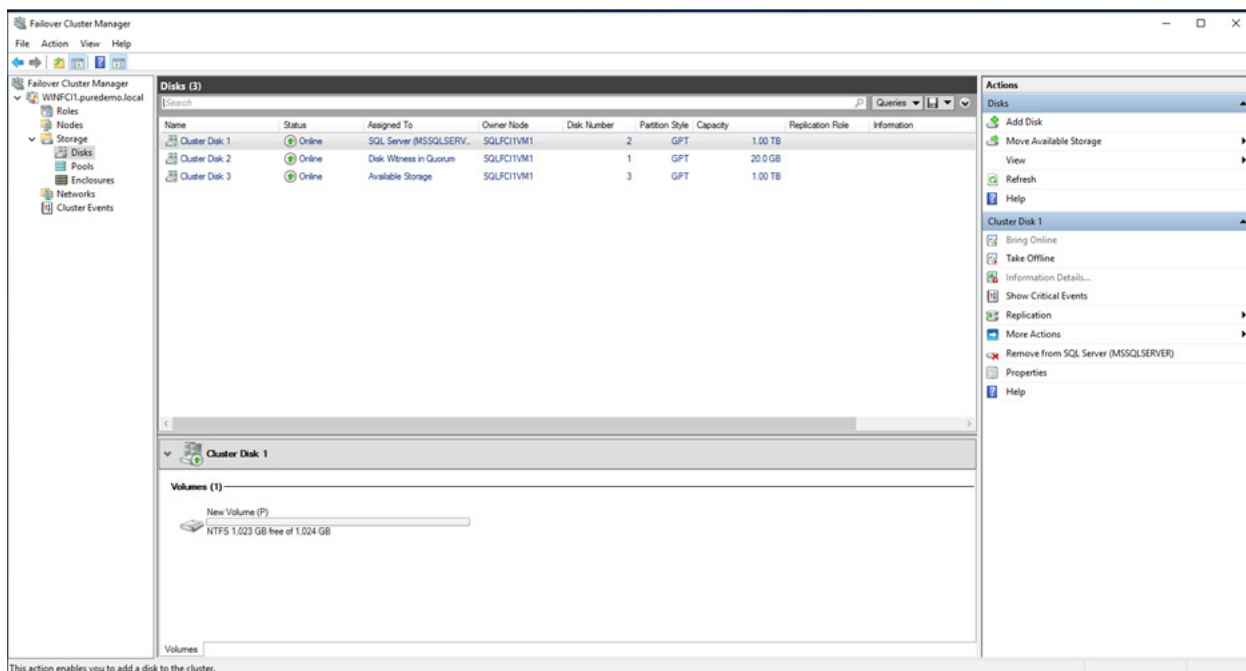
Name	
	Connect... Disconnect... Show Remote Connections Download CSV

No host groups found.

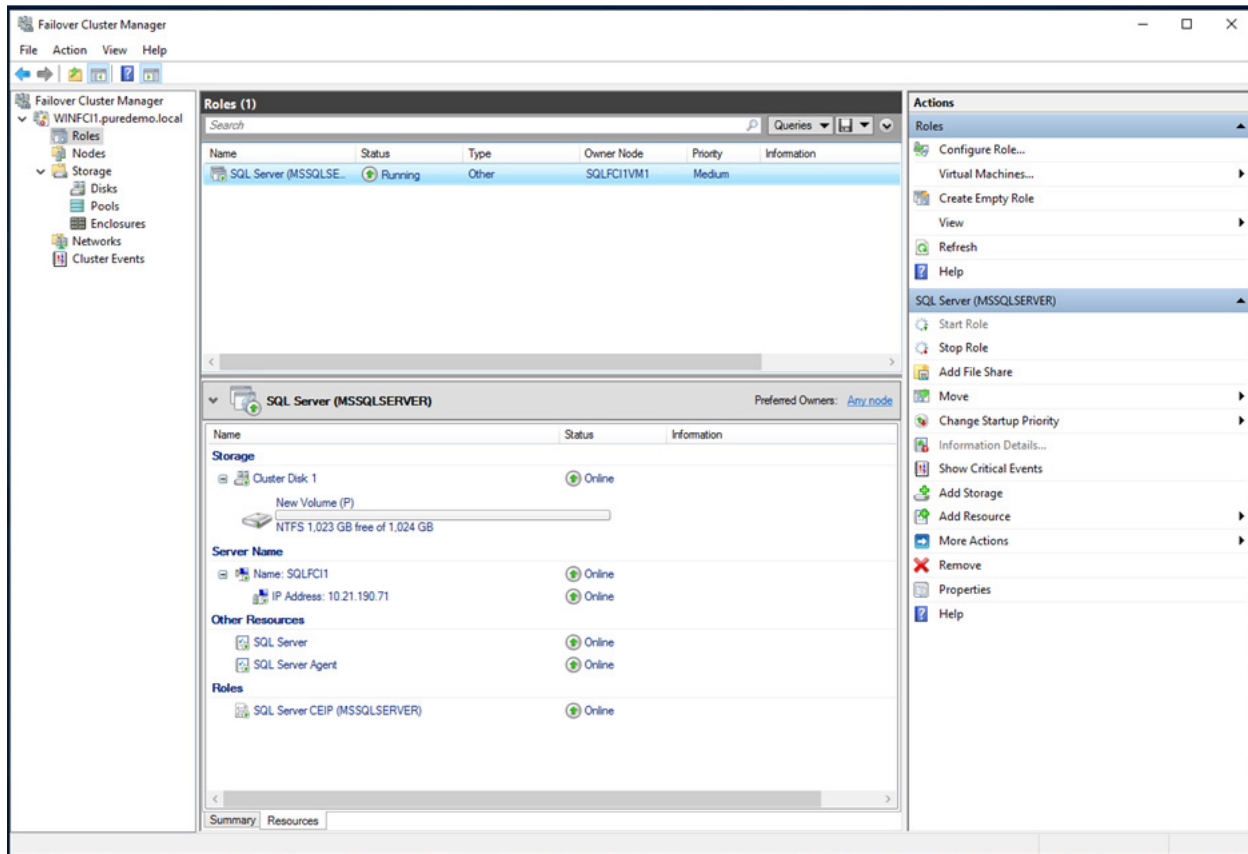
- On the VMs, open on the first node of the failover cluster. Bring the new disk online, initialize it, create a partition, and assign a drive letter.



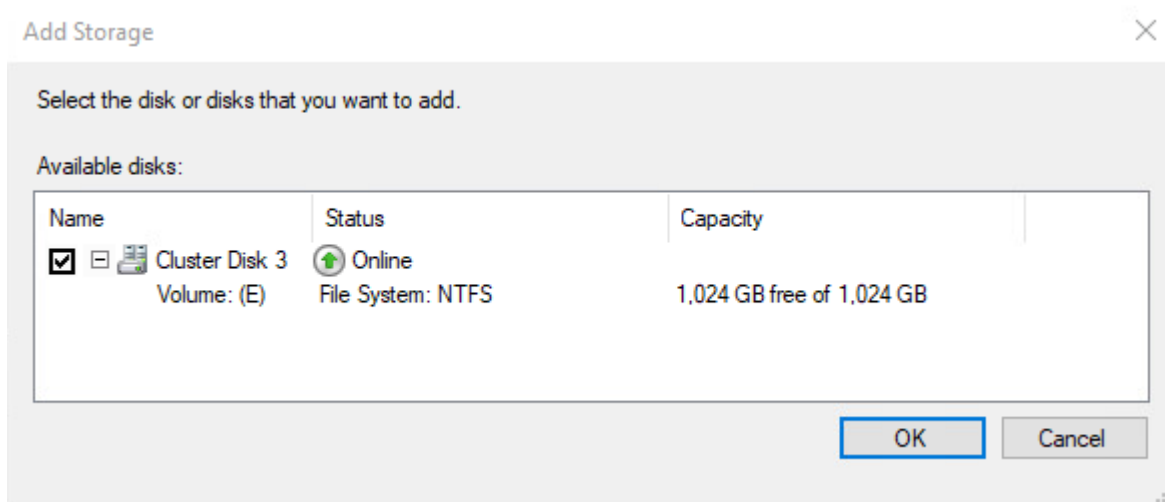
- Within Failover Cluster Manager, select Add Disk and add the newly created volume to the cluster. After adding the disk to the cluster, it will show up in the Disks list under Storage.



23. Within Failover Cluster Manager, select the SQL Server role and select "Add Storage."



24. Select the Cluster Disk which was created in the previous steps and click OK.

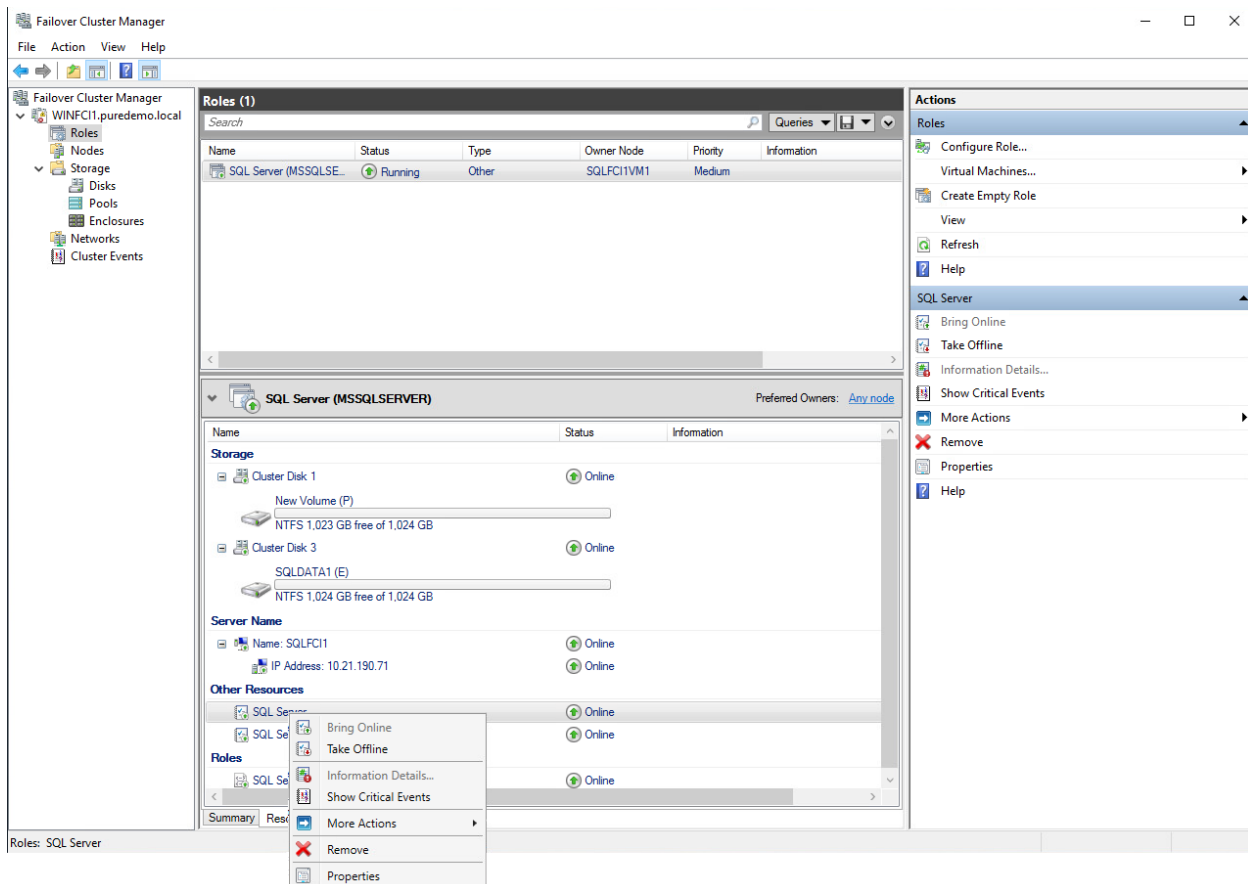


25. The Cluster Disk must be added as a dependency to the SQL Server role. Until this is done, trying to create databases on this Cluster Disk will fail with an error like the following:

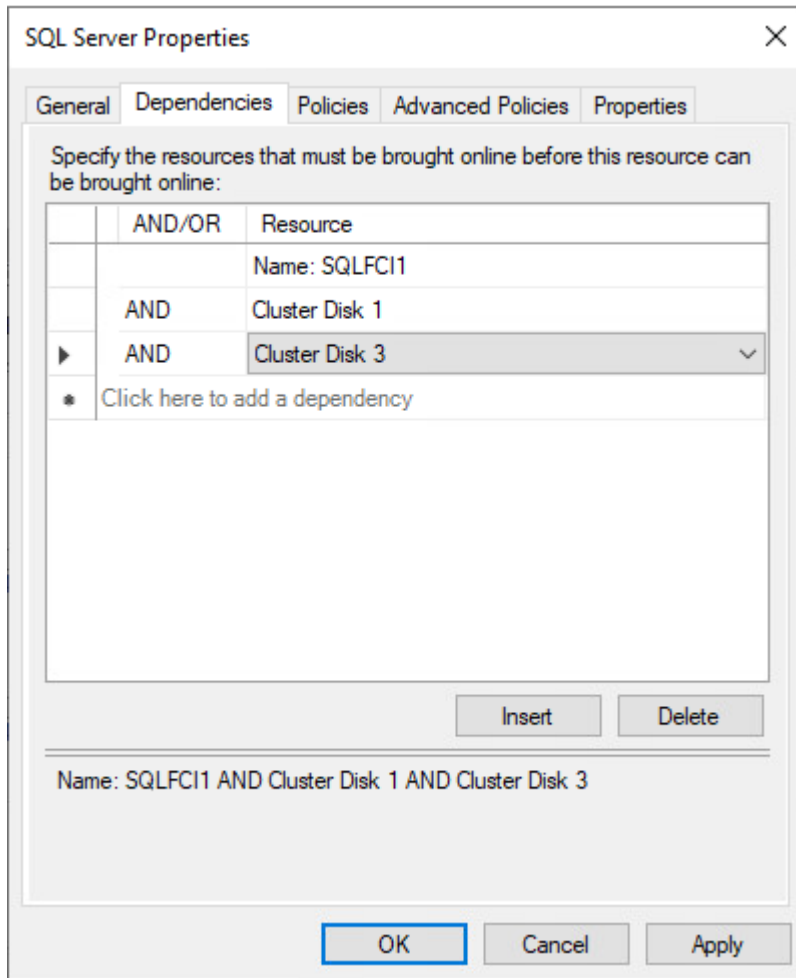
```

Administrator: Windows PowerShell - SQLCMD
PS C:\Users\administrator.PUREDEMO> sqlcmd
1> use master;
2> go
Changed database context to 'master'.
1> create database puredemo
2> on
3> ( NAME = puredemo_dat,
4> FILENAME = 'E:\MSSQLDATA\puredemo.mdf',
5> SIZE = 10,
6> MAXSIZE = 50,
7> FILEGROWTH = 5 )
8> LOG ON
9> ( NAME = puredemo_log,
10> FILENAME = 'E:\MSSQLLOG\puredemo.ldf',
11> SIZE = 5MB,
12> MAXSIZE = 25MB,
13> FILEGROWTH = 5MB ) ;
14> GO
Msg 5184, Level 16, State 2, Server SQLFC11, Line 1
Cannot use file 'E:\MSSQLDATA\puredemo.mdf' for clustered server. Only formatted files on which the cluster resource of the server has a dependency can be used.
Either the disk resource containing the file is not present in the cluster group or the cluster resource of the Sql Server does not have a dependency on it.
Msg 1802, Level 16, State 1, Server SQLFC11, Line 1
CREATE DATABASE failed. Some file names listed could not be created. Check related errors.
1>
    
```

26. Right-click the SQL Server entry under “Other Resources” and select “Properties.”



27. Click on the “Dependencies” tab and add the Cluster Disk, specifying “AND” as the dependency type.



SQL Server Properties

General Dependencies Policies Advanced Policies Properties

Specify the resources that must be brought online before this resource can be brought online:

AND/OR	Resource
	Name: SQLFC11
AND	Cluster Disk 1
AND	Cluster Disk 3
* Click here to add a dependency	

Insert Delete

Name: SQLFC11 AND Cluster Disk 1 AND Cluster Disk 3

OK Cancel Apply

28. Repeat as necessary to add additional disks.

The SQL Server FCI is now complete and ready for use.

Configure CloudSnap to Azure

The creation and configuration via the array UI of an Offload Target that points to an Azure Storage Blob Account can be done by following the instructions located in the [Purity CloudSnap Best Practices Guide](#) under the section Creating an Offload Target in Azure. As this procedure could also be automated, we have supplied an [example script in this script repository](#) that can be modified to suit the intended environment.

Note: CloudSnap cannot be used to offload data directly from inside an ActiveCluster stretched pod. To use CloudSnap with ActiveCluster, you must create a Protection Group which exists outside of the ActiveCluster pod and configure this Protection Group to use CloudSnap to Azure as a target. Volumes within the ActiveCluster pod must be cloned, and the clones then added to this Protection Group to be replicated to CloudSnap for Azure. A script is available in the software repository accompanying this paper which completely automates this process.



Configure VSS Backup of SQL Server Databases

The Pure Storage FlashArray Management Extension for Microsoft SQL Server Management Studio, or SSMS extension, should only be installed on one management server. Whenever an application-consistent VSS snapshot is taken, information about the snapshot set is written to a metadata cab file. These cab files are used to perform a restore or mount operation. Each SQL server is added to the extension as a new Computer Credential. Each time an operation takes place, the versions of the SSMS extension are checked, and if something is out of date or missing, the current version is pushed to that SQL Server. This means that once you install the SSMS extension to the management server, anytime you invoke a backup, everything will automatically be installed and kept up to date.

The procedures shown below could also be automated using PowerShell and PowerCLI. You will find an example script [in our script repository](#).

If the Azure and on-premises networks are linked through a VPN or ExpressRoute, only one management server with SSMS is required.

Potential Prerequisites


In a normal production environment, vCenter will be provisioned with valid certificates, so when a vCenter credential is added, the SSMS extension will pull the “VMware.PowerCLI” module from the PSGallery. If the management server is not on the Internet, get it directly from [VMware](#). If the vCenter certificate is invalid [administrator action](#) should be performed before the configuration stage. Most of these prerequisites are not required on a domain-joined management server that has Internet access with valid VMware certificates in place.

Configure the SSMS extension

1. Ensure an 18.x version of SSMS is installed.
2. Install the Pure Storage SMS extension.
3. On the Title Bar, select Pure Storage, and then New Backup Configuration.




4. Configuration: Enter a friendly name that implies the name of the SQL instance and database for ease of automation.

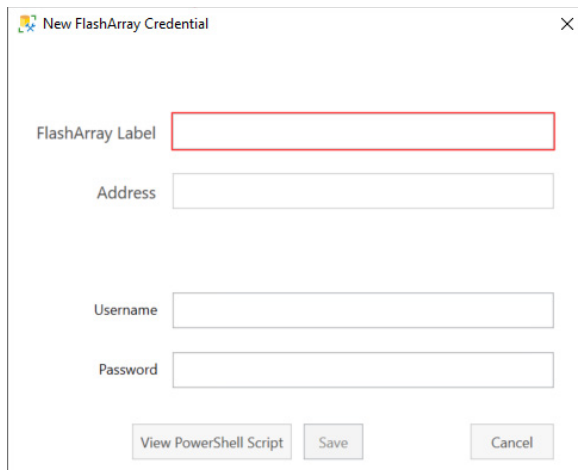
5. Computer: Select the  icon to create a new Computer Credential.

6. Computer Label: Friendly name for the SQL Server or SQL FCI.

7. Address: IP Address, Hostname, or FQDN.


8. Username: Domain\Username or Username on a workstation.

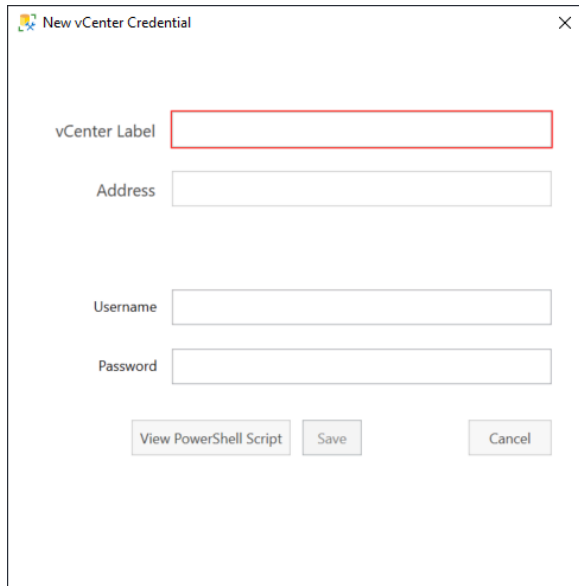
9. Password: Enter the Password. This will be encrypted and only usable by the user who created the credential.
10. Save.
11. Choose to create additional SQL Server Computer Credentials or continue with the Backup Configuration.
12. Database: After creating a computer credential, select that Computer, and the databases will be enumerated on all instances that the credential has permissions for. Create additional computer credentials if other SQL instances require different credentials. Select the database to be backed up. Keep in mind all databases that are co-located on the Pure Volumes of the selected database will also be backed up.
13. FlashArray: Select the  icon to create a new FlashArray Credential. In the case of ActiveCluster, ensure that a credential for both FlashArrays is created.

The screenshot shows a dialog box titled 'New FlashArray Credential' with a close button (X) in the top right corner. It contains four text input fields: 'FlashArray Label', 'Address', 'Username', and 'Password'. Below the fields are three buttons: 'View PowerShell Script', 'Save', and 'Cancel'. The 'FlashArray Label' field is highlighted with a red border.

14. FlashArray Label: Friendly name indicating which FlashArray.
15. Address: The IP Address of the FlashArray.
16. Username: FlashArray user.
17. Password: Password for the FlashArray user.
18. Save.

Note: The Azure SQL FCI VM's can ignore the RDM support steps and skip to step 28.

19. Check the box for VMware® ESXi RDM Support.
20. vCenter: Select the  icon to create a new vCenter Credential.

A screenshot of a "New vCenter Credential" dialog box. The dialog has a title bar with a close button. It contains four text input fields: "vCenter Label", "Address", "Username", and "Password". The "vCenter Label" field is highlighted with a red border. At the bottom, there are three buttons: "View PowerShell Script", "Save", and "Cancel".

21. vCenter Label: Friendly name indicating which vCenter Server.
22. Address: The IP Address of the vCenter Server.
23. Username: vCenter username.
24. Password: vCenter password.
25. Save.
26. VM Name: Name of the standalone VM, or name of the SQLFCI network name, or SQLFCI IP Address resource in the SQL FCI.
27. Metadata location: By default, a folder will be created under the logged-in user. This location can be changed but only the logged-in user will have access to restore or mount snapshots that were created with this user.
28. Type: Select [Full](#) or [Copy only](#).
29. Save.

At this point, the credentials have been created, checked, encrypted, and saved on the management server. Backup jobs can be run manually by right-clicking a configured database, selecting Pure Storage, Backup, and then selecting the Backup button. Automation examples are in the [automation script repository](#).

To restore or mount a backup, right-click a configured database, selecting Pure Storage, then Restore Backup History. To see the global backup history, select Pure Storage on the Title Bar, and then Backup History.

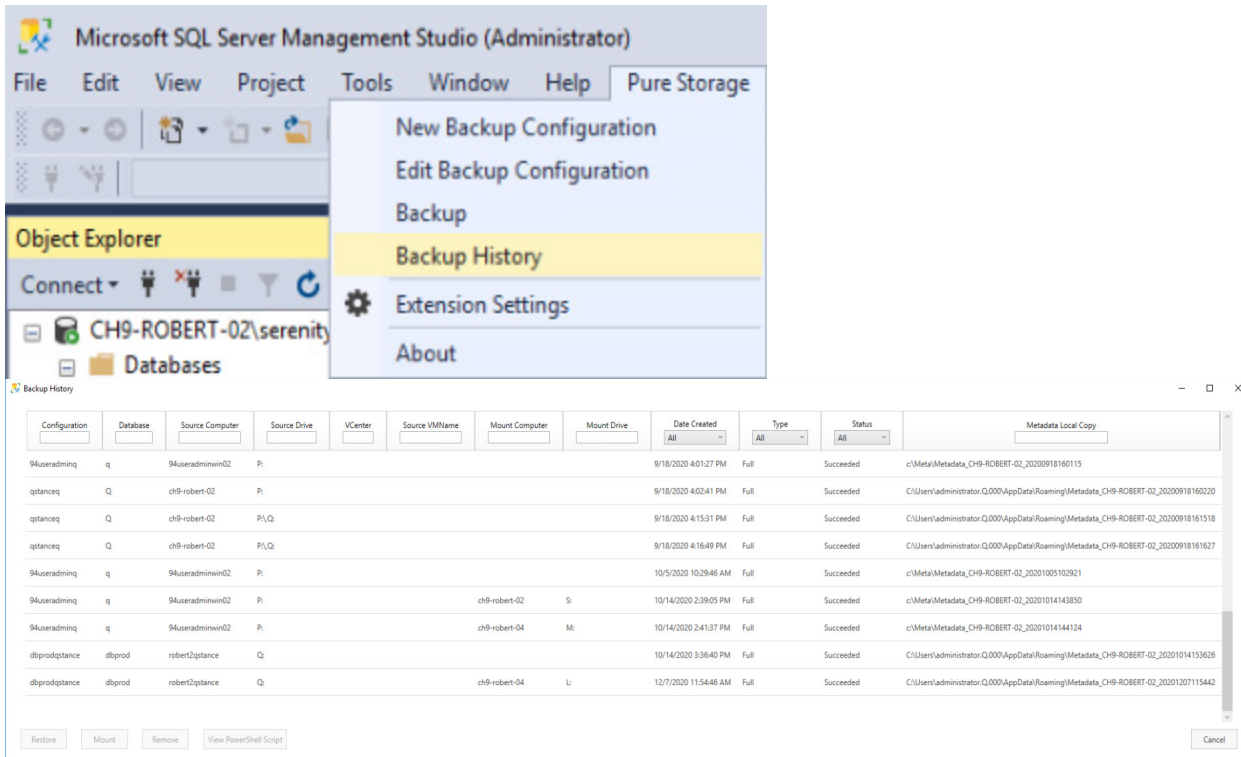


Figure 16. Pure Storage backup history.

The Backup History window enables restore, mount, and dismount operations.

Monitoring

The following sections describe monitoring solutions employed in this guide that have some overlap but, in most cases, monitor different things that may be of use to a particular administrator. The FlashArray GUI Dashboard is the storage-centric view of the hardware health, configuration, and storage utilization. This view can also be seen in the FlashArray CLI. A higher-level fleet view can be obtained in Pure1 with a browser or [REST APIs](#). Windows administrators can add the Pure Storage plugin to the Windows Admin Center and manage servers and storage in the same browser tab. Unique features, such as the ability to see a view of just the servers that have Pure Volumes connected to them in a view, while simultaneously checking for MPIO best practices, where you can fix any non-optimal settings with a click. The Pure Storage Plugin for the vSphere Client, which is simply deployed from the FlashArray GUI, enables VMware administrators to quickly perform simple tasks such as, expand a disk size, or provision a new one. While these are the most common monitoring tools, others can be employed, and each user can decide which tools meet their needs. The FlashArray can be directly interfaced with REST APIs, through the [Python](#), and [PowerShell SDKs](#). 3rd party management tools include [SMI-S](#) and for [FlashStack](#)® deployments, [Cisco Intersight](#).

Monitoring using FlashArray GUI Dashboard

The Pure Storage FlashArray graphical user interface is a simple and intuitive means to manage and monitor the FlashArray. While the breadth of everything that can be done in the GUI is beyond the scope of this document, a few of the more relevant examples are noted here.



Figure 16 shows the Array Connections pane. With this tool, one can tell at a glance the status of the connections used for ActiveCluster replication.

Array Connections					
Array		Status		Replication Transport	
sn1-x90r2-e06-31		● connected		Ethernet (IP)	
Remote Array \ Local Array		CT0		CT1	
		ETH4: 10.21.190.113	ETH5: 10.21.190.114	ETH4: 10.21.190.115	ETH5: 10.21.190.116
CT0	ETH4: 10.21.190.123				
	ETH5: 10.21.190.124				
CT1	ETH4: 10.21.190.125				
	ETH5: 10.21.190.126				

Figure 16. Monitoring Array Connections.

Figure 17 shows the replication traffic for the pods which have been selected. In this instance, we can see that the SQL FCI pod is replicating synchronously.

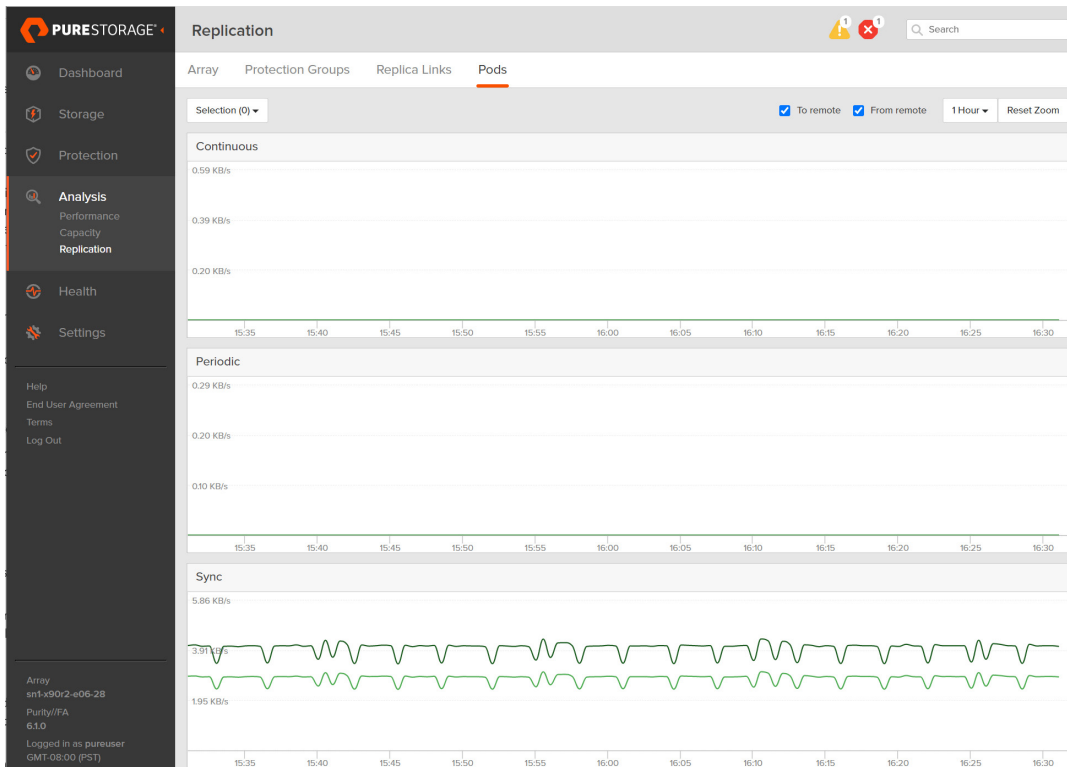


Figure 17. Replication Analysis



Under the network settings view shown in Figure 18, it is easy to quickly identify the Ethernet interfaces which have been designated to be used for replication, along with their speed and status.

Ethernet ^											11-20 of 27
Name	Enabled	Type	Subnet	Address	Mask	Gateway	MTU	MAC	Speed▼	Services	
ct0.eth4	true	physical	-	10.21.190.113	255.255.255.0	10.21.190.1	1500	24:a9:37:02:f8:d9	25 Gb/s	replication	
ct0.eth5	true	physical	-	10.21.190.114	255.255.255.0	10.21.190.1	1500	24:a9:37:02:f8:d8	25 Gb/s	replication	
ct1.eth2	false	physical	-	-	-	-	1500	24:a9:37:03:9c:53	25 Gb/s	iscsi	
ct1.eth3	false	physical	-	-	-	-	1500	24:a9:37:03:9c:52	25 Gb/s	iscsi	
ct1.eth4	true	physical	-	10.21.190.115	255.255.255.0	10.21.190.1	1500	24:a9:37:03:9c:55	25 Gb/s	replication	
ct1.eth5	true	physical	-	10.21.190.116	255.255.255.0	10.21.190.1	1500	24:a9:37:03:9c:54	25 Gb/s	replication	
ct0.eth10	true	physical	-	-	-	-	1500	90:e2:ba:af:9b:95	10 Gb/s	iscsi	
ct0.eth11	true	physical	-	-	-	-	1500	90:e2:ba:af:9b:94	10 Gb/s	iscsi	
ct1.eth10	true	physical	-	-	-	-	1500	90:e2:ba:8a:2b:c1	10 Gb/s	iscsi	
ct1.eth11	true	physical	-	-	-	-	1500	90:e2:ba:8a:2b:c0	10 Gb/s	iscsi	

Figure 18. Network Settings.

For a more in-depth look at what can be managed and monitored from the FlashArray graphical user interface, consult the [FlashArray User Guide](#).

Plan and Monitor with Pure1

While the FlashArray has a great UI and API for monitoring performance and capacity, when you have multiple storage appliances and more in the cloud as with this design, you will want a single place to monitor everything.

[Pure1](#) is that place – an application-aware cloud-based monitoring tool that uses the Meta AI engine to give you lots of details about what your arrays are doing, allows you to plan for future growth, and lets you look across the stack from the array down to individual virtual machines. Pure1 is part of the world-class Pure Support offerings and is available to all Pure customers with a support contract.

There is a lot of information available in Pure1, and we won't cover that all here. The complete guide to Pure1 can be found at the [Pure Support site](#).

Planning

The wealth of data available in Pure1 allows our Meta AI engine to help you see what the future looks like for both performance and capacity of the Pure Storage that is backing your SQL instances. Right from the start, the Workload Planner will give you recommendations on sizing for Microsoft SQL. Meta has analyzed workloads for many common applications including SQL and can give you sizing for typical, small, large, or custom workloads:



Add New Workload



Workload Name: **New SQL**

Select workload type: **MsSQL**

Workload size: **Typical**

Workload Size
By analyzing the deployed workloads across all Pure arrays, we provide recommendations for what range of storage parameters to expect for different scales of workloads. All values are based on hourly averages.
— Typical deployments

Capacity parameters

Size (TiB):

Data Reduction:

Read performance parameters

Avg IO Size (KiB):

IOPS:

Bandwidth (MiB/s):

Write performance parameters

Avg IO Size (KiB):

IOPS:

Bandwidth (MiB/s):

Figure 19. Adding a new workload.

Once you have chosen a new SQL workload, you can then take that and model it against any existing or new FlashArray. This is a powerful tool not just for new workloads, but to see how your existing workloads will change over time.

Pure1 includes a powerful analytic tool that we call **Load**. Load is a calculation based on thousands of data points from each FlashArray that gives you a single number from 0-100 to show how that array is performing. The Workload Planner can show you how your Load and your capacity will change over time. You can simulate scaling volume workloads, cloning workloads, or migrating workloads to another array.

In this example, we've picked an array, *dogfood-chuckwagon*, and selected the *workloads2* volume. We scaled the volume by 5x to see what the projected Load and capacity changes will be on this array.

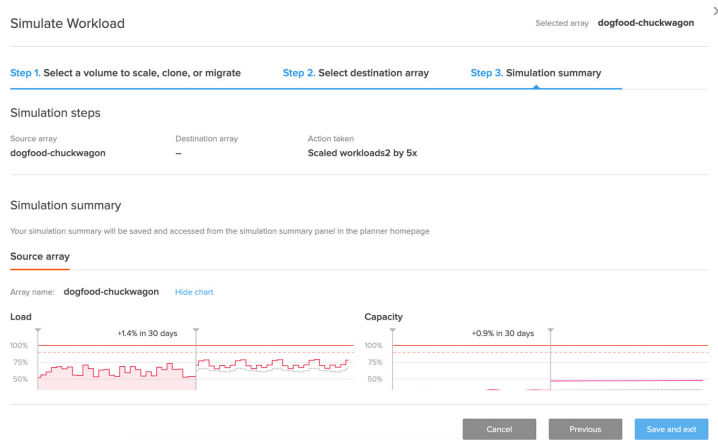


Figure 20. Simulated scaling.



As you can see, scaling this workload does result in an increase in Load and capacity over the next 30 days. The Workload Planner will let you look forward up to 12 months. This is a key tool for your data center planning because it takes the guesswork out of what you will need to support your application and your business growth.

Monitoring

Pure1 gives you a wealth of information about the health, performance, capacity, data protection, and other details of your Pure Storage appliances, whether they are in your data center, a subscription service, or the cloud.

When you first log in to Pure1, you'll see the Appliances screen. This gives you an overview of each Pure Storage appliance in your fleet and can be sorted by many different criteria. Here we are sorting this view by capacity full, but your options include performance metrics, alerts, and even support contract status.

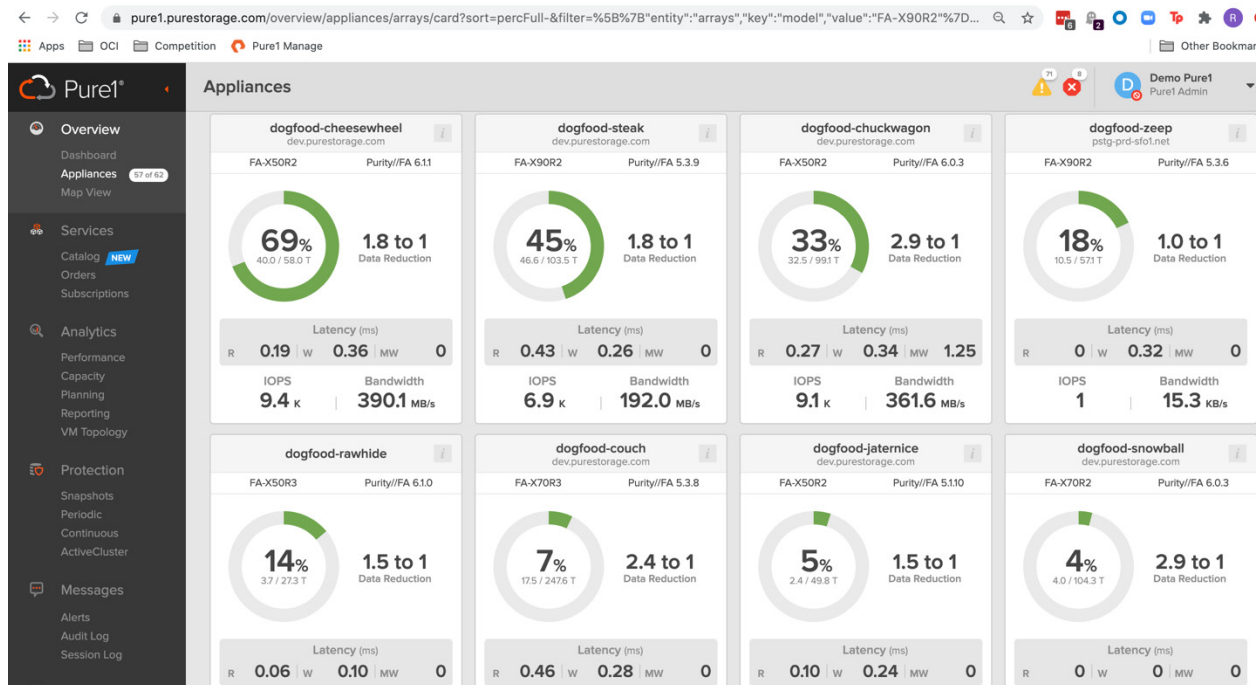


Figure 21. Appliances screen in Pure1.

As you can see this view gives you a great overview of capacity, data reduction, and performance, at one glance letting you see what's going on. You can expand these card views to see even more details about the health and components of each array or show them in a list view.

Above we talked about Load to measure the performance of your FlashArrays. The Performance section of Pure1 is where you can see details of Load as well as IOPS, latency, and bandwidth. These metrics are available for the last 3 hours up to the last 30 days.

When you have many arrays, you may want to narrow your results on any given page down to find the specific array you're looking for. Pure1 has a global filter bar, shown in the example below. Here we have filtered specifically for *dogfood-chuckwagon*, the array we used in the planning example above.

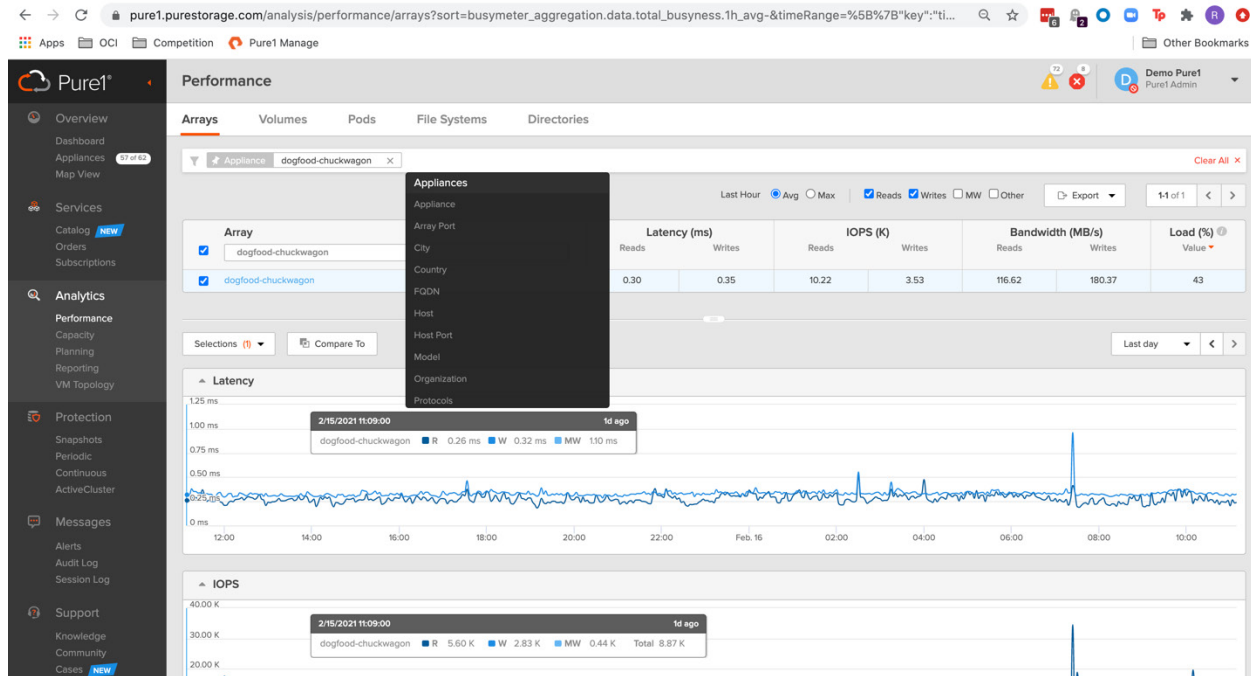


Figure 22. Performance interface.

The performance screen will show you latency, IOPS, bandwidth, and finally, Load:



Figure 23. Load graph for dogfood-chuckwagon.

With multiple arrays selected, you can compare any of these metrics across your fleet to see how different workloads are affecting your storage.

Chances are your SQL environment is virtualized. This is true for many SQL deployments across all types of business environments. We know this is a highly efficient way to operate, but we also know that virtual environments come with their own challenges, particularly when you'd like to see details about how one part affects another.

The good news here is that Pure1 has a great capability for you in VM Analytics. It requires the installation of a collector agent in vCenter, which sends data to Pure1 that Meta uses to analyze and show you all the details of your VMware environment from the FlashArray down to each individual VM, vDisk, or vVol.

Like the other sections in Pure1, VM Analytics can be sorted by performance, capacity, or many other metrics depending on what you are looking at. Host and VM CPU and memory, for example, or datastore capacity. In the example below, we are viewing by Capacity % Used, and we have sorted the Datastore column to show that *steak_prod1* is the highest used capacity datastore in this particular vCenter tier.

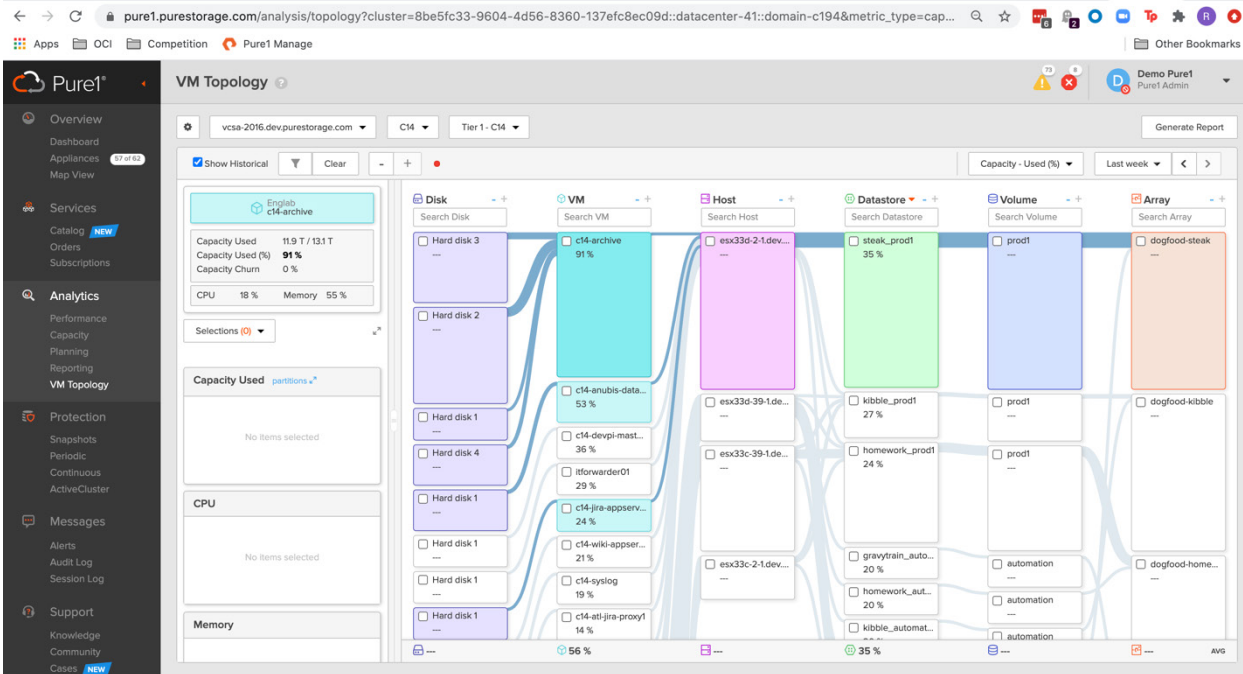


Figure 24. VM analytics with Pure1.

However, you will also notice that the top VM in this datastore is at 91% capacity! VM Analytics allows you to discover this type of information easily. One other metric we can show you is Capacity Churn. This tells you for each VM or Datastore what percentage of change, plus or minus, is happening in your environment, making it easy for you to find unexpected growth.

One last item we'll cover here: data protection. This design covers the use of both snapshots and replication to protect your business-critical data. Pure1 has you covered with monitoring of all data protection across your Pure Storage fleet.

All snapshots taken are visible in Pure1 and you can see at a glance the specific snapshots for any given array or volume. Here we are again looking at dogfood-chuckwagon and the snapshots are taken for the *engineer_vm2* volume. This view will show you not only when the snapshots were taken, but what kind of snapshots they are. In this case, they are protection group snapshots that are replicated to the *dogfood-kong* array.

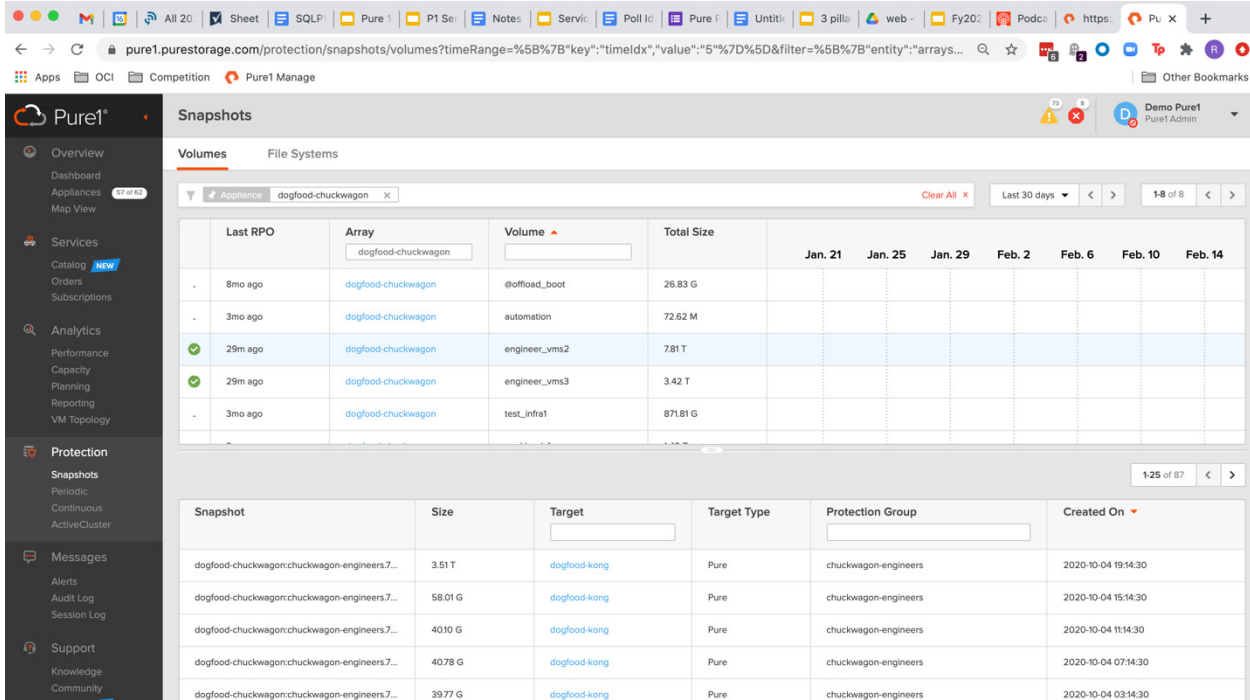


Figure 25. At-a-glance view of Snapshots.

To show you even more details, we can go to the Periodic replication screen, filter again for *dogfood-chuckwagon*, and see the replication relationship between *dogfood-chuckwagon* and *dogfood-kong*:

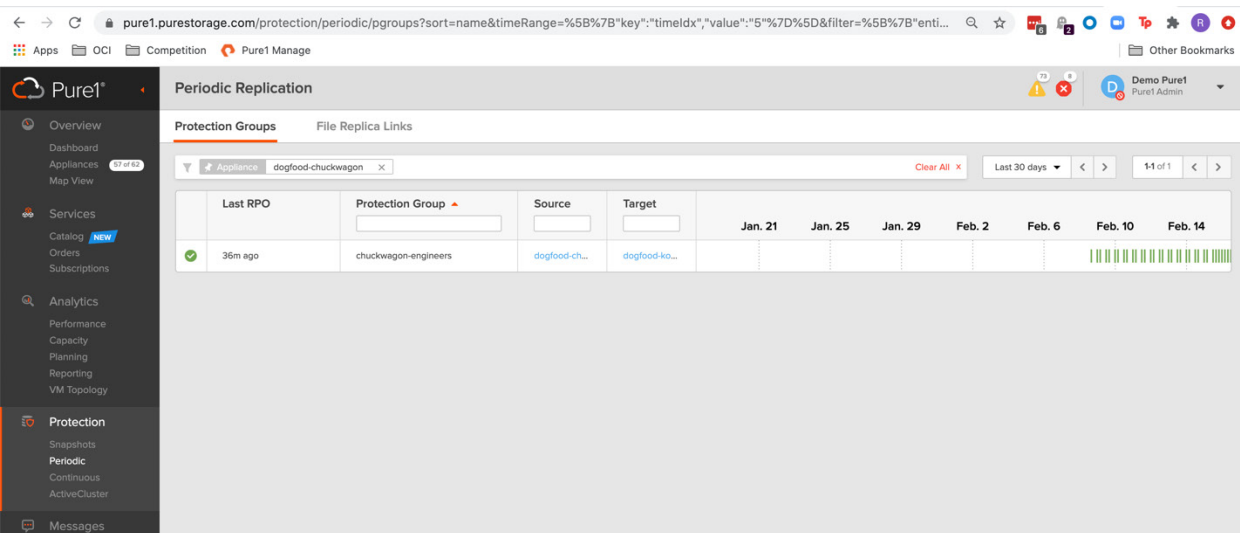


Figure 26. Periodic replication screen.

Here you can see the history up to 30 days of any replication transfer, and if you click on any of the green bars you can get detailed information on what amount of data was transferred and when.

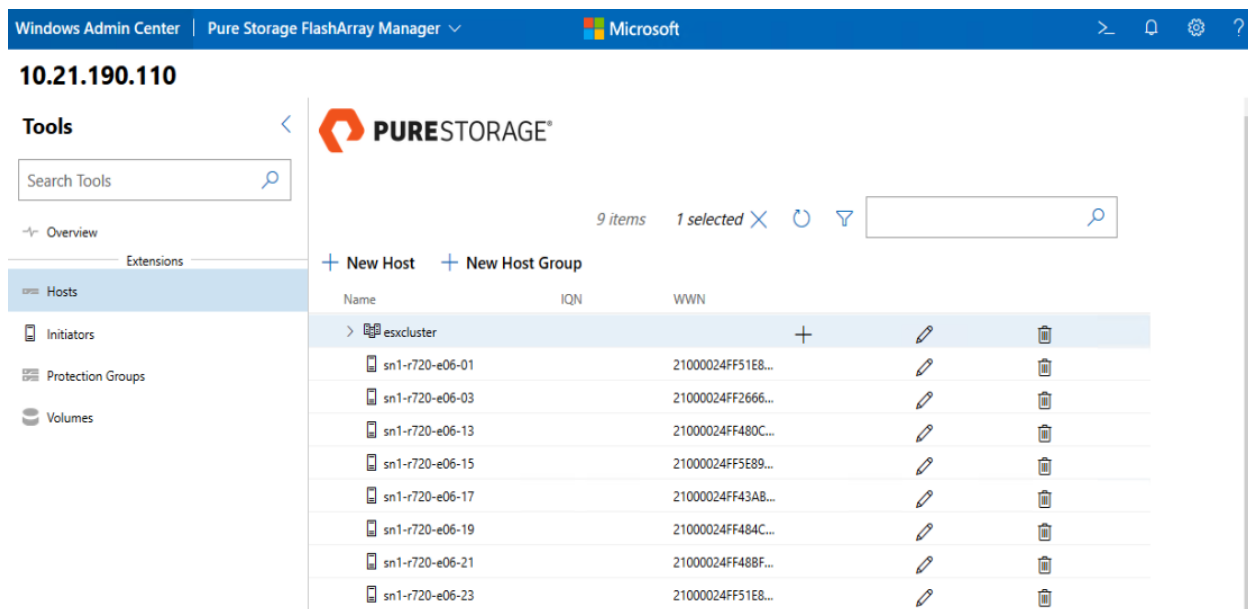
It's worth noting that Pure1 also provides detailed monitoring for ActiveDR, our continuous replication for FlashArray, as well as ActiveCluster.

Pure1 also includes a Support section that lets you open and update cases with our Support team, as well as a convenient calendar tool to schedule upgrades of Purity.

For more information on all the Pure1 capabilities, please do consult the [Pure1 Support Documentation](#).

Monitoring Using Windows Admin Center

Windows Admin Center (WAC) is the browser accessed tool provided by Microsoft for the remote management of infrastructure and services in your environments. The Pure Storage Extension for WAC enables the administrator to configure and monitor both Hosts and Pure FlashArrays. Some of the views and functionality are not available in any other tool, such as the ability to see which servers that have been added to WAC also have a connection to the FlashArray.



The screenshot shows the Windows Admin Center interface for the Pure Storage FlashArray Manager. The address bar indicates the IP address 10.21.190.110. The main content area displays the Pure Storage logo and a list of hosts. The list has columns for Name, IQN, and WWN. The selected host is 'esxcluster', which contains 9 items. The list shows the following data:

Name	IQN	WWN
esxcluster		
sn1-r720-e06-01		21000024FF51E8...
sn1-r720-e06-03		21000024FF2666...
sn1-r720-e06-13		21000024FF480C...
sn1-r720-e06-15		21000024FF5E89...
sn1-r720-e06-17		21000024FF43AB...
sn1-r720-e06-19		21000024FF484C...
sn1-r720-e06-21		21000024FF48BF...
sn1-r720-e06-23		21000024FF51E8...

Figure 27. Pure Storage Extension for Windows Admin Center.

The WAC is very useful for monitoring and managing Windows Servers and their connection to a FlashArray on-premises, in the public cloud using Pure Cloud Block Store, or when running physical Windows Servers. The Pure Storage vSphere plugin should be used for managing volumes connected to an ESXi cluster.

For instructions for installing the WAC on a Server or Windows 10 computer, refer to this [Microsoft documentation](#). The details of how to install and configure the Pure Storage extension can be found in the [Pure Storage FlashArray Solutions Extension for Windows Admin Center](#).



Monitoring Using Pure Storage Plugin for the vSphere Client

The Pure Storage Plugin for the vSphere Client provides information about the Pure Storage FlashArrays directly to the vSphere administrator and is an easy way to configure and provision storage on those FlashArrays directly from within the vSphere client. Additionally, the plugin can connect to Pure1 to enable additional monitoring options, insight, and intelligent provisioning.

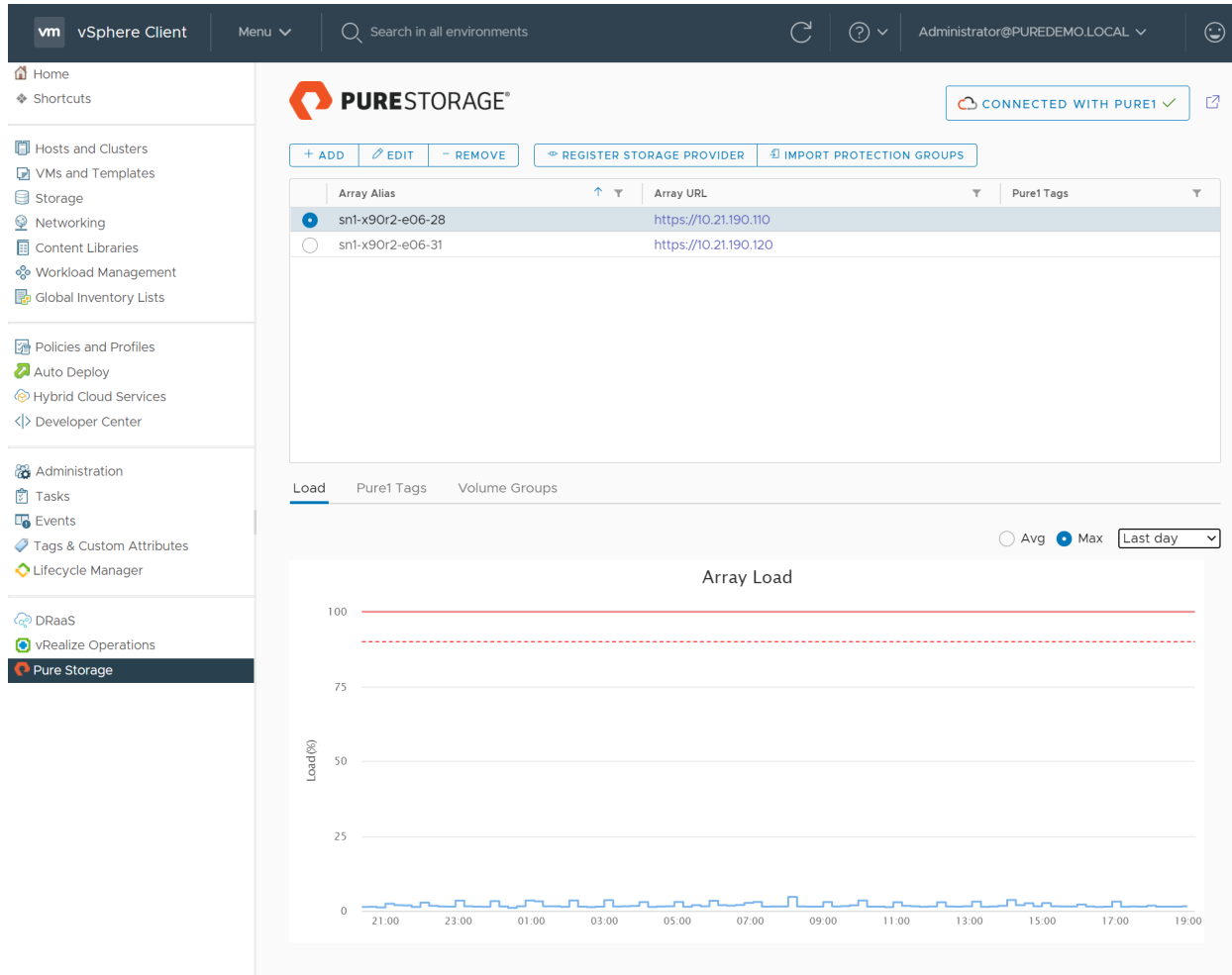


Figure 28. Pure Storage Plugin for the vSphere Client.

Details of how to install, configure, and use the Pure Storage Plugin for the vSphere Client can be found within the [Pure VMware Platform Guide](#), specifically in the [Pure Storage Plugin for the vSphere Client User Guide](#).



Conclusion

The Pure FlashArray platform, on-premises and in the cloud enables storage, hypervisor, and application administrators high performance and the agility to adapt to changes in the business environment. The modular nature of the solutions highlighted in this guide, enables customers to deploy the subset of features necessary to meet their business needs. The features shown are built-in to Purity in the FlashArray and Pure Cloud Block Store and can be easily utilized without external licensing or delays. SQL Server databases can be protected, replicated, and cloned quickly and space efficiently on-premises and in the cloud. Extend the value of this solution with [FlashBlade](#), object storage with high throughput and linear scaling performance, and [Portworx](#)®, Pure Storage's Kubernetes data services platform.

Additional Documentation

- VMware VM Windows Failover Cluster pRDM configuration: [first node](#), [additional nodes](#).
- ActiveCluster with VMware [User Guide](#).
- FlashArray VMware [Best Practices](#).
- Microsoft SQL Server with [ActiveCluster](#).
- [Pure Cloud Block Store for Azure](#).
- Pure Cloud Block Store [Deployment and Configuration Guide for Azure](#)
- Pure Cloud Block Store [Deployment Guide for Azure Using Azure CLI](#)
- Automation Repository – [Scripts and Templates](#).
- Pure Storage OpenConnect [GitHub repository](#)

Product Support

Pure and Microsoft will support their customers following each respective company's normal support process. When the need arises, the Pure and Microsoft support teams will engage each other to collaborate. Pure offers support services over the phone, by email, and through our web portal.

To contact pure storage support:

- Web: pure1.purestorage.com/support
- Email: support@purestorage.com
- Phone (US): +1 (866) 244-7121 or +1 (650) 729-4088
- Phone (international): support.purestorage.com/pure1/support



About the Authors

Mike Nelson, Julian Cates, and Robert Quimbey are Solutions Architects for FlashArray Microsoft Solutions at Pure Storage.

Document Updates

We are always looking to improve the quality of our content and documentation and welcome your feedback. Please send us your comments at pvd-documents@purestorage.com.

Document Revisions

Rev #	Description	Date
1.0	Initial Publication	March 2021

©2021 Pure Storage, the Pure P Logo, and the marks on the Pure Trademark List at <https://www.purestorage.com/legal/productenduserinfo.html> are trademarks of Pure Storage, Inc. Other names are trademarks of their respective owners. Use of Pure Storage Products and Programs are covered by End User Agreements, IP, and other terms, available at: <https://www.purestorage.com/legal/productenduserinfo.html> and <https://www.purestorage.com/patents>.

The Pure Storage products and programs described in this documentation are distributed under a license agreement restricting the use, copying, distribution, and decompilation/reverse engineering of the products. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors, if any. Pure Storage may make improvements and/or changes in the Pure Storage products and/or the programs described in this documentation at any time without notice.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041

purestorage.com

800.379.PURE

