PURE VALIDATED DESIGN

Pure FlashBlade with Vertica in Eon Mode for Analytics

Pure Validated Design

Table of Contents

Executive Summary

In today's innovation-driven economy, analytics has become a major competitive differentiator. Companies must ingest and analyze a wide range of data types, often in real-time, to respond to business and customer needs. The sheer volume of data and speed required to do this can present daunting challenges. To meet these challenges, Vertica and Pure Storage have partnered to design, validate, and deliver a high-performance, simple, and massively scalable unified data analytics solution.

A Pure Validated Design (PVD) means that Pure has integrated and validated its leading-edge storage technology with an industry-leading application solution platform to simplify deployment, reduce risk, and free up 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 FlashBlade with Vertica in Eon Mode for Analytics.

A Vertica in Eon Mode on Pure FlashBlade solution is especially well-suited for multifaceted enterprise application workloads having different SLAs and workload profiles. This analytical solution brings together the leading big data analytics platform with an industry-best all-flash storage array to deliver speed, scalability, and reliability for mission-critical analytics. Vertica in Eon Mode on Pure FlashBlade combines the power of a high-performance, massively parallel processing SQL platform and query engine with advanced analytics and machine learning with the speed of all-flash storage emulating a cloud object store. In cloud environments, object storage provides a fast, cost-effective means of storing massive data volumes. However, many organizations have chosen not to migrate to the public cloud due to unintended costs, technical complexities, loss of control, governance, and more. The Vertica in Eon Mode on Pure Storage analytical solution brings all the capabilities and benefits of public cloud-optimized solutions to on-premises environments, enabling customers to unlock the true potential of their data to deliver timely insights, drive business improvements and sales conversions, and sharpen market focus.

Introduction

This document describes the benefits of implementing a Vertica in Eon Mode analytics solution using Pure Storage FlashBlade. It provides validated design considerations, sizing guidelines, deployment specifics, and configuration best practices for a fast, simple, and highly scalable unified data analytics solution.

Solution Overview

This Pure Validated Design solution consists of Vertica in Eon Mode deployed with a single Amazon S3 compatible bucket created on FlashBlade. Vertica in Eon Mode on Pure Storage FlashBlade is the industry's first analytical database solution to separate compute and storage for on-premises environments and leverage fast object storage for an on-premises cloud solution.

The Vertica in Eon Mode on FlashBlade integrates the Vertica columnar database application, compute, storage, and network into an enterprise-class data analytics solution. This solution brings the power of FlashBlade storage to the Vertica Massively Parallel Processing (MPP) architecture and delivers the fastest data search capacity in the industry, in an easy to deploy and manage platform. Vertica in Eon Mode separates compute and storage components to give organizations greater flexibility and operational efficiency. Organizations can easily extend their analytical capability in a modular way and achieve linear gains in capacity and performance by scaling compute or storage components separately. The high-level architecture is designed with scalability and growth in mind (Figure 1).

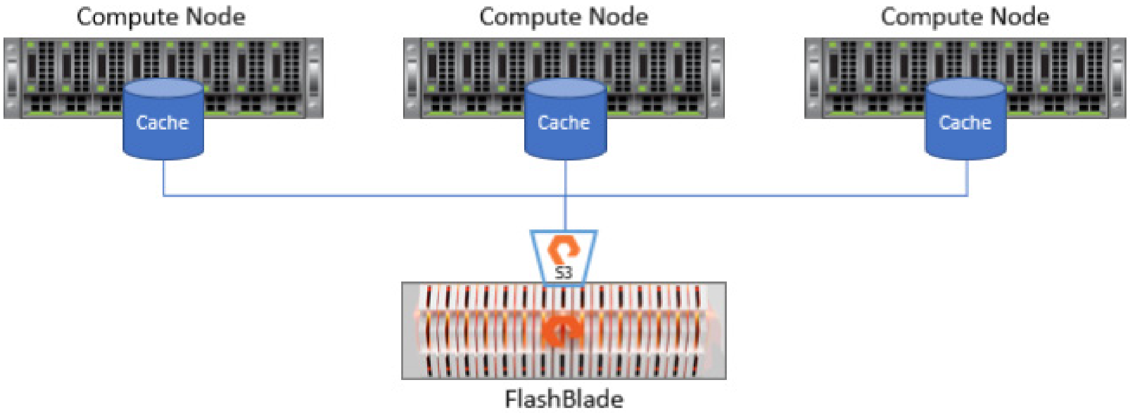


Figure 1. Pure FlashBlade Storage with Vertica in Eon Mode High-Level Architecture

Solution Benefits

Vertica in Eon Mode is built on a read-optimized relational SQL database designed to handle an organization's most challenging analytic workload. Vertica in Eon Mode is optimized to address variable workloads and isolate them from one another by using disaggregated compute and storage resources. The durable copy of the database resides in a communal, Amazon S3 cloud compatible object store provided by Pure FlashBlade. Vertica in Eon Mode also improves performance by using a "Depot," an on-node disk cache storage location, as cache-like transient storage to provide a copy of the data that is local to each node and therefore faster to access. The Vertica in Eon Mode on FlashBlade solution offers rapid data ingestion from multiple sources, advanced data protection with elastic scale, speed, and simple management

Benefits of the Vertica in Eon Mode on FlashBlade solution include performance, scalability, operational simplicity, and resilience.

Performance

Pure Storage and Vertica jointly collaborated to optimize access patterns of the Vertica databases to take advantage of the multi-dimensional performance and scalability of FlashBlade storage. Optimization includes loading data in such a manner that it can be queried intelligently and quickly. Testing demonstrated that the Vertica in Eon Mode on FlashBlade took 54 minutes to load 10TB of data vs. 71 minutes for DAS-based systems.

Unlike the architectures of Oracle, SQL Server, and other relational databases, the Vertica MPP architecture stores table data in columnar form, rather than in rows. This optimizes data loads and accelerates queries. Vertica reads only the columns referenced by any query, instead of scanning the whole table as row-oriented databases must do. This dramatically accelerates query processing by reducing disk I/O, especially for datasets with many sparse columns. In addition, Vertica in Eon Mode uses the "Depot," an on-node disk cache storage location, to accelerate queries. The combination of this data caching on the hosts and high-speed low latency storage with the Pure FlashBlade Amazon S3 compatible object store provides faster data loading and better query performance.

Scalability

With the Vertica in Eon Mode on FlashBlade solution, organizations can easily build a robust analytics solution, starting with the smallest possible FlashBlade configuration at 65TB usable capacity (7x 17TB blades), and can scale the storage capacity as high as 4.7PB usable.

Vertica columnar compression combined with Pure FlashBlade communal storage and FlashBlade in-line compression provides storage capacity that is twice as efficient as a similarly configured Vertica Enterprise solution on DAS. It does this by eliminating multiple copies of data in at least two locations, while with Vertica in Eon Mode the data is stored in communal storage only once.

Vertica in Eon Mode offers a variety of advantages, including performance, for workloads that benefit from sub clustering, frequent rebalancing, and other operational elements of database management. Vertica in Eon Mode, when configured with Pure FlashBlade storage, delivers a very powerful high-performance analytics platform leveraging communal storage. For highly variable workloads, Vertica in Eon Mode with Pure FlashBlade reduced the time required for database rebalancing from 74 minutes to less than 5 minutes based on testing of an 8-node cluster scaled to 16 nodes. The result is a 15x improvement in the time to scale the cluster environment. This increase from 8 to 16 nodes also resulted in an almost 50% reduction in query runtime. The data supporting these findings is included in the design validation section of this document.

Operational Simplicity

An application as critical as data analytics will require high SLAs from IT. Therefore, the data analytics solution must be designed to provide efficiencies in operation and maintenance. The disaggregated architecture of the Vertica in Eon Mode on FlashBlade solution enables IT to maintain compute nodes in a minimal state, making maintenance operations on these nodes fast and simple. As analytics demands grow, administrators can expand storage without having to add or remove nodes. Clusters can be expanded or contracted without having to rebalance them, which will save time and resources. FlashBlade is self-healing, self-tuning, simple to operate, and easy to integrate into automated workflows. IT engineers can manage petabytes of storage with a minimal time commitment.

Resilience

The Vertica in Eon Mode on FlashBlade solution is resilient to failures and errors. FlashBlade is highly reliable, highly available, and self-healing. Vertica is engineered to be always consistent on disk and self-healing as well. The combination of Vertica and FlashBlade enables accelerated workflows for backup and recovery, and space-efficient rapid cloning to enable isolated work environments for development and test teams.

Technology Overview

Vertica in Eon Mode on FlashBlade is a set of loosely coupled, multi-core clustered x86 hosts running Linux, and communicating with each other over redundant high-speed LAN interconnects. The Linux hosts access FlashBlade storage over a segregated TCP/IP data network. Figure 2 provides a detailed architecture design for the platform.

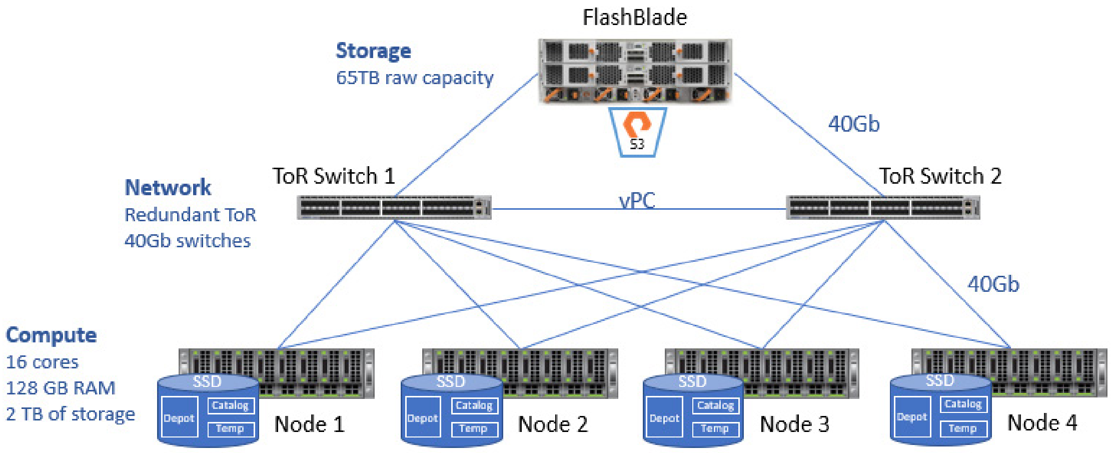


Figure 2. Pure FlashBlade Storage with Vertica in Eon Mode Detailed Architecture

Compute Resources

Physical or virtual multi-core servers are clustered together to provide the compute resources. Vertica recommends using a homogeneous hardware configuration for each Vertica subcluster with each node in the cluster configured with the same number of cores, the same amount of memory, and running the same version of the operating system. Each server runs a supported x86_64 Linux Operating System and each is configured with the required number of processor cores and memory required to handle the workload of the Vertica in Eon Mode database instance. Figure 3 shows server architecture options.

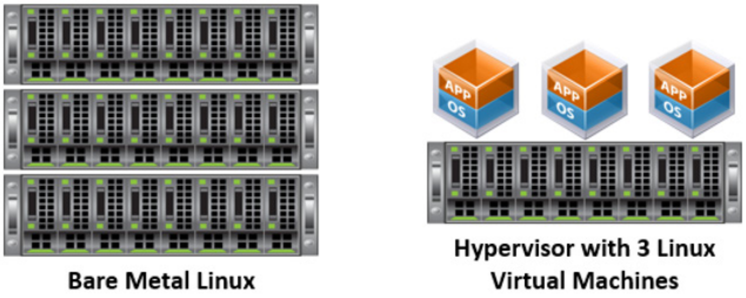


Figure 3. Pure FlashBlade Storage with Vertica in Eon Mode Server Options

Pure FlashBlade Storage

FlashBlade is an all-flash system, delivering a unified fast file and object storage platform optimized for storing, processing, and protecting unstructured data that addresses the requirements of modern applications. The FlashBlade storage layer in the Vertica in Eon Mode on FlashBlade solution brings superior performance to the functionality of the Vertica unified data analytics. FlashBlade is used as an object store and also simplifies storage expansion, with seamless growth up to multiple petabytes and non-disruptive upgrades.

Chassis

Each FlashBlade chassis can be configured with up to 15 blades for processing data operations and storage (Figure 4). A full FlashBlade system configuration consists of up to 10 self-contained rack-mounted chassis. For reliability, each chassis is equipped with redundant power supplies and cooling fans.

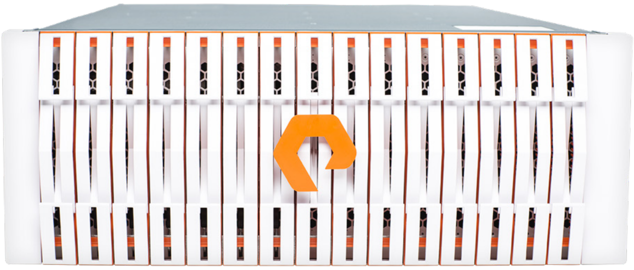


Figure 4. Pure FlashBlade Chassis Front View

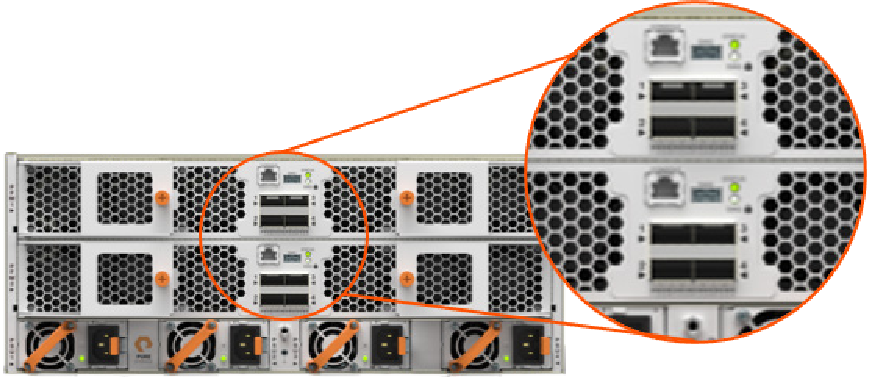


Figure 5. Pure FlashBlade Chassis Rear View - On-Board Fabric Modules

External Fabric Modules

For FlashBlade configurations with more than 15 blades, the rack-mounted chassis (Figure 5) are interconnected by high-speed links to two external fabric modules (XFM). At the rear of each chassis are two on-board fabric modules (Figure 6) for interconnecting the blades, other chassis, and clients using TCP/IP over high-speed Ethernet. Both fabric modules are interconnected, and each contains a control processor and Ethernet switch ASIC.

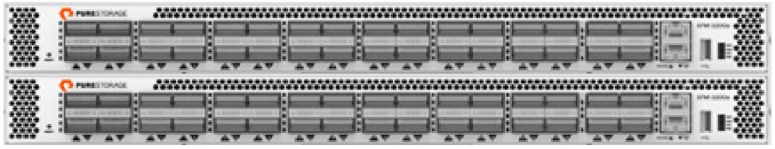


Figure 6. Pure FlashBlade External Fabric Modules (XFM)

Blade

Each blade is a self-contained compute module equipped with processors, communication interfaces, and either 17TB or 52TB of flash memory for persistent data storage. Each blade can be hot-plugged into the system to add capacity and performance (Figure 7).

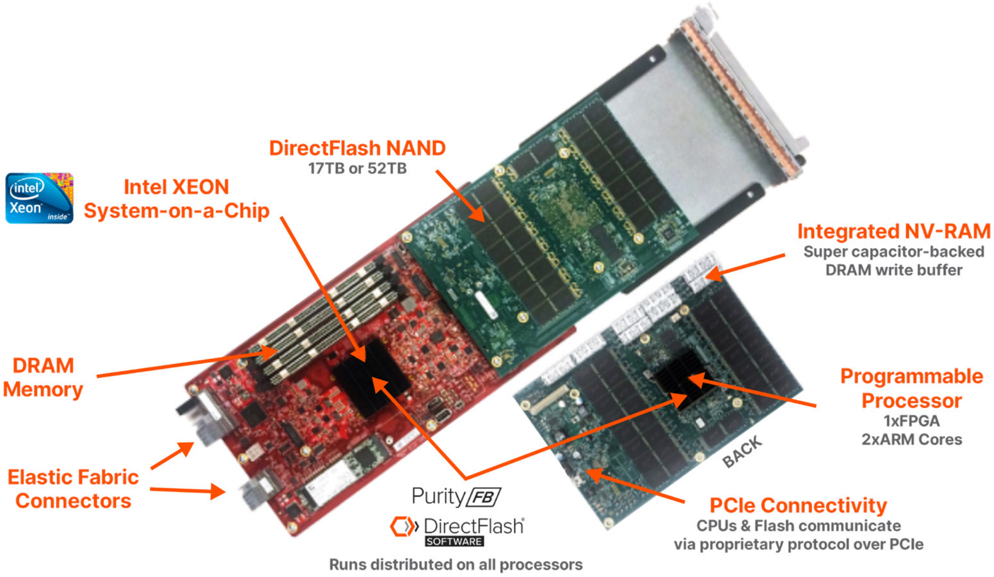
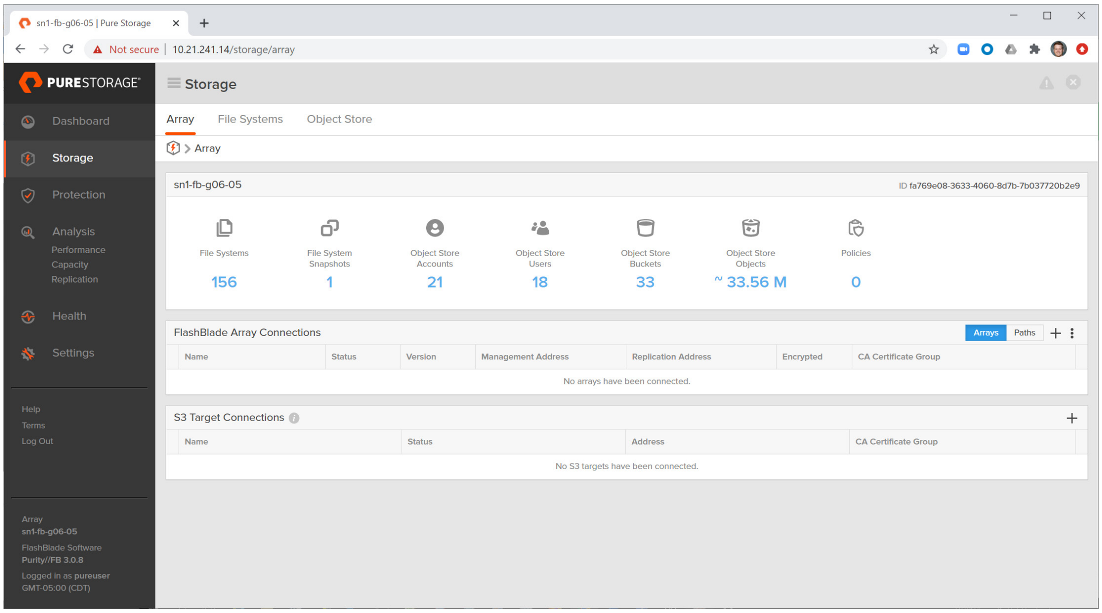


Figure 7. Pure FlashBlade Assembly

Purity//FB

Purity//FB is the FlashBlade operating system that runs on fabric modules (Figure 8). It minimizes workload-balancing problems by distributing client operation requests among the blades on FlashBlade storage. It is the heart of FlashBlade and is architected on a massively distributed key-value database for limitless scale and performance, delivering enterprise-class data services and management with simplicity. NFS file and S3 object protocols are native to the Purity//FB software stack.



sn1-fb-g06-05 ID fa769e08-3633-4060-8d7b-7b037720b2e9

File Systems	File System Snapshots	Object Store Accounts	Object Store Users	Object Store Buckets	Object Store Objects	Policies
156	1	21	18	33	~ 33.56 M	0

FlashBlade Array Connections

Name	Status	Version	Management Address	Replication Address	Encrypted	CA Certificate Group

No arrays have been connected.

S3 Target Connections

Name	Status	Address	CA Certificate Group

No S3 targets have been connected.

Array
sn1-fb-g06-05
FlashBlade Software
Purity//FB 3.0.8
Logged in as pureuser
GMT-05:00 (CDT)

Figure 8. Purity//FB Dashboard

Network Resources

The network considerations for Vertica in Eon Mode with Pure Storage are the same as any enterprise IT infrastructure solution: availability, performance, and extensibility. The nodes in the solution can attach to any compatible TCP/IPv4 or TCP/IPv6 network infrastructure, with Vertica recommending a minimum network speed of 10 GbE. For Vertica in Eon Mode with Pure Storage network topologies, Vertica recommends dual switches to eliminate a single switch as a single point of failure.

Vertica in Eon Mode

Vertica is a columnar-focused analytics database that enables faster query performance than relational databases. Vertica was introduced as a high-speed SQL Analytical Database, with tightly coupled storage. When used as Vertica in Eon Mode, functionally it is equivalent to Vertica in Enterprise Mode, sold with the same license – but it separates compute from storage. Vertica in Eon Mode uses Pure FlashBlade storage as a single communal storage location for all data and the catalog (metadata). This communal storage is the database's centralized storage location, shared among the database nodes. Within communal storage, data is divided into portions called shards. Compute nodes access central storage by subscribing to particular shards, with subscriptions balanced among the compute nodes. Shard subscriptions are dynamically updated as nodes join and leave the cluster.

Compute nodes are responsible for all data processing and computations. A compute node stores a copy of frequently queried data and a portion of the catalog that represents the shards to which that node subscribes. Each compute node subscribes to one or more shards. Recently accessed data for subscribed shards is kept in the Depot, a cache-like component (Figure 9). This intermediate layer of data storage enables faster access to this copy of the data that is local to the node. Data that is frequently used in queries takes priority in the Depot. If the data for a query is not in the Depot, then Vertica will read the data from communal storage. The Depot improves query performance by preventing some queries from making the longer trip to communal storage.

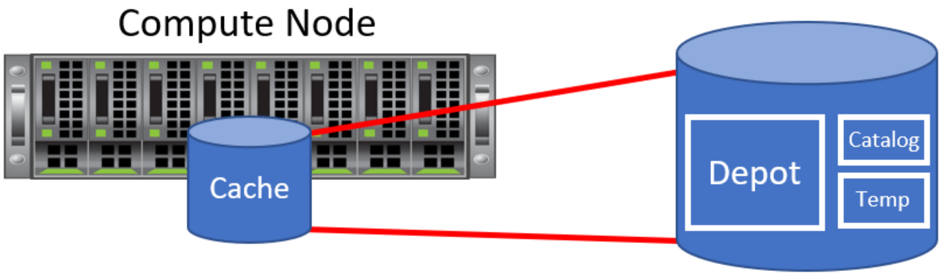
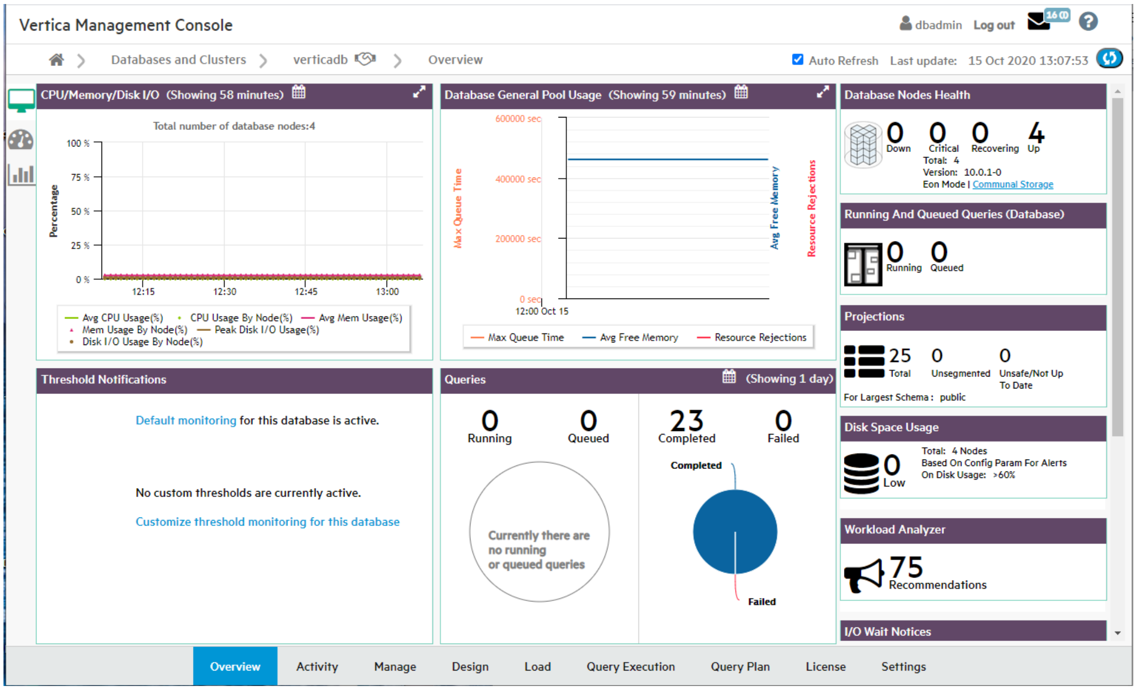


Figure 9. Vertica in Eon Mode Cache (the Depot).

Vertica Management Console

The Vertical Management Console (Figure 10) is the Vertica browser-based management and monitoring tool used to operate, provision, and deploy a Vertica in Eon Mode database. It is a graphical user interface that provides a unified view of the Vertica database operational state. It provides a simple-to-use graphical interface that allows administrators to create, configure, manage, and monitor your Vertica databases and associated clusters.



Default monitoring for this database is active.

No custom thresholds are currently active.

Customize threshold monitoring for this database

Currently there are no running or queued queries

Figure 10. Vertica Management Console

Technical Solution Design

Vertica is a unified analytics data warehouse, based on a massively scalable architecture with the broadest set of analytical functions spanning event and time series, pattern matching, geospatial, and end-to-end in-database machine learning. Vertica in Eon Mode provides the added benefit of separating compute from storage, thereby enabling simplified management for variable workloads with the flexibility to apply specific compute resources to shared storage for different business use cases. Vertica in Eon Mode uses a single communal storage location for all data and the catalog (metadata). Communal storage is the database's centralized storage location, shared among the database nodes. Communal storage is based on an object store, and in this solution, the object store is provided by Pure Storage FlashBlade using S3 protocol and emulating Amazon S3 cloud storage. Vertica relies on the FlashBlade object store to maintain the durable copy of the data. Vertica in Eon Mode maintains a local disk cache of data called the Depot for faster access. Vertica allows bulk loading of the data from multiple sources. Vertica in Eon Mode is optimized to address variable workloads and isolate them from one another with disaggregated compute and storage resources.

Pure Storage FlashBlade is the ideal communal storage location for Vertica in Eon Mode. Providing high throughput and fast time to first byte, FlashBlade can host multiple file systems and multi-tenant object stores for thousands of clients. Since each blade assembly is a self-contained compute module (equipped with processors, communication interfaces, and either 17TB or 52TB of flash memory for persistent data storage), FlashBlade provides a highly scalable solution to meet growing storage demands and enables IT administrators to improve productivity and consolidate silos by supporting multiple data sources on one storage platform.

Compute Layer

Vertica with Pure FlashBlade storage must have a minimum of three nodes in the Eon Mode database cluster. The required server is a commodity x86-64 architecture with a recommended minimum configuration based on this PVD:

- 16 cores
- 128GB
- 2TB of local storage

Vertica requires a minimum of 8GB of memory per physical CPU core in each server. However, in high-performance applications, 12GB–16GB of memory per physical core is recommended. The memory should be at least DDR3-1600 (preferably DDR4-2133 or better) and should be appropriately distributed across all memory channels in the server.

The number of compute servers required for the solution correlates directly with the amount of raw data expected to be loaded within the Vertica in Eon Mode database. For every 10TB of raw records stored in the database, Vertica recommends that you add a compute node to the cluster. For example, 50TB of raw capacity should have at least five compute nodes.

FlashBlade Storage Layer

FlashBlade storage is an all-flash data hub comprised of a chassis that house fabric modules and blade data processing storage units that all run the Purity operating environment. Fabric modules provide a scale-out software-defined fabric that allows near-linear scale for storage in the cluster. They are also responsible for inter-chassis connectivity as well as host access. The blades are powerful are elastic data processing storage units that can be scale within a chassis and by adding additional chassis. The platform runs Purity, the massively distributed software for near limitless scale. The FlashBlade storage system is designed for simplicity with a self-tuning architecture and native S3 support, which means there are no storage design considerations for the Vertica in Eon Mode on FlashBlade solution.

Vertica in Eon Mode Design Considerations

Vertica in Eon Mode separates compute and storage which enables rapid scaling and shrinking of clusters in response to a variable workload. Vertica in Eon Mode decouples the cluster size from the data volume and lets administrators size storage based on compute needs. A Vertica administrator designing a production cluster running in Vertica in Eon Mode must make important decisions about database design to meet business needs.

Before reviewing the design considerations, it is important to understand the functional aspects of the Vertica in Eon Mode database as compared to other relational databases.

Considerations for designing the Vertica in Eon Mode database include:

- **Working Data Size:** Working data size is the amount of data on which most queries will operate. More on this in the sizing section.
- **Depot Location:** To get the fastest response time for frequently executed queries, provision a Depot large enough to hold the working data set. When loading data, Vertica must write uncommitted data files into the Depot before uploading to communal storage. Vertica will evict files from the Depot to make space for new files if the free space in the Depot is not sufficient.

- **Data Location:** Data location is used for data files that belong to temporary tables and temporary data from sort operators that spill to disk. When loading data into Vertica, the sort operator may spill to disk and, depending on the size of the load, Vertica may do the sort in multiple merge phases if temporary data written to disk runs out of space.
- **Catalog Location:** The catalog size depends on the number of database objects per shard and the number of shard subscriptions per node.
- **Local Disk Sizing Recommendation:** Vertica recommends a minimum local storage capacity of 2TB per node, out of which 60% must be reserved for the Depot and the other 40% can be shared between the catalog and data location. Since this storage will be used for Depot cache and spill space, we recommend selecting storage technologies (like Optane or NVMe SSD) that in aggregate across all nodes will be substantially faster than the FlashBlade.
- **Concurrency and Throughput Scaling:** In Vertica in Eon Mode, the database achieves elastic throughput scaling by adding more nodes and creating a sub-cluster. To create a sub-cluster, administrators must define a fault group where the number of nodes is a factor of the shard count.

Sizing

When sizing Vertica in Eon Mode with FlashBlade storage, administrators should size for both the physical infrastructure and the Vertica in Eon Mode cluster. The server infrastructure must have the capacity to handle the required database capacities. The Vertica in Eon Mode cluster must be configured to support the capacities for optimum queries and reporting.

Infrastructure Sizing Considerations

For optimal performance, Vertica recommends running processors configured with two-socket servers with 8- to 14-core CPUs, clocked at or above 2.6 GHz.

Vertica requires a minimum of 8GB of memory per physical CPU core in each server. For Vertica in Eon Mode, 12GB–16GB of memory per physical core is recommended. The memory should be at least DDR3-1600 (preferably DDR4-2133) and should be appropriately distributed across all memory channels in the server.

Vertica requires local storage to have a minimum read/write speed of 40 MB/s per physical core of the CPU. For optimal performance, 60–80MB/s per physical core is recommended.

Nodes should have a minimum of 10Gbs networking bandwidth for data traffic and 1Gbs for management traffic.

Cluster Sizing Considerations

Several factors must be considered when sizing the Vertica in Eon Mode with Pure FlashBlade storage.

- **Data volume:** Understanding the total raw data volume for the cluster is very important. If there is an existing data warehouse, calculate the raw size from that. If the analytical database is going to be constructed from multiple data marts, calculate the sum of each data mart to get an idea of the total capacity required for the Pure FlashBlade storage. The objective is to get an approximation, not a hard number, as the necessary data volume will increase over time.
- **Compression:** If there are previously attained compression numbers, apply those. If not, estimate with a 2:1 compression ratio. The combination of data volume and the compression ratio will give an initial sizing capacity following the formula below:

  Initial database size = data volume / compression ratio

- **Data growth:** Determine the daily volume ingestion rate (i.e., the amount of data loaded in the database each day) as well as the organization's data retention policy. Then estimate the total data volume required for the cluster following the formula below:

  Total volume capacity required = (ingest_rate × retention_period) + initial database size

Settings and Tuning

The Pure FlashBlade Tuned for Everything storage architecture comes tuned for any workload. The Tuned for Everything architecture benefits include the ability to:

- Mix databases and applications without worrying about unpredictable noisy neighbors destroying SLA's;
- Avoid bottlenecks and consolidate silos on a platform architected to handle a wide range of workloads at once;
- Use the same storage for all stages of the data pipeline to avoid time-consuming copies between silos;
- Stage data on the same FlashBlade and use NFS mounts from all nodes for faster ON ANY NODE loads;
- Ingest data in large batches to utilize more efficient IO patterns; and
- Stop stranding capacity by consolidating multiple teams, projects, and applications on the same storage platform.

You can use Vertica in Eon Mode subclusters to isolate variable workloads. Each subcluster can be set up and tuned for very different workloads without affecting others. Subclusters can be tuned up or tuned down based on needs. Vertica supports a few settings at each subcluster level, including Depot pinning. Depot pinning tells Vertica what table and partitions to keep in the Depot. Pinned tables and partitions remain resident in the Depot and are only evicted as a last resort. Unpinned files are evicted first and only when another pinned file comes in will a pinned file be evicted for the immediate query being executed.

Scaling

The Vertica in Eon Mode on Pure FlashBlade storage is easy to scale. Benefits include:

- Freedom to scale compute and storage at different rates to meet evolving business conditions and needs.
- Ability to host many applications on the same cluster without worrying about exhausting storage while stranding CPU.
- Ability to burst additional compute capacity in minutes when needed.
- Flexibility of the cloud, with the safety and predictability of on-premises deployments.
- Reduced data silos and access to the consolidated data from on-premises and multiple competitive cloud providers.

Adjusting the size of the cluster does not interrupt analytic workloads, which allows compute and storage to be scaled separately.

Design Validation

The design validation provides evidence of the solution's integrity. It demonstrates the benefits through a series of test cases comparing Vertica in Eon Mode with Vertica in Enterprise Mode.

Scalability and Capacity

Vertica is a column-oriented SQL database that stores the data in a compressed format to optimize storage capacity and performance. In a Vertica in Enterprise Mode deployment of the Vertica database, data is replicated to more than one node to protect from node failures. The number of replicas is defined by the **k-safety** factor of a database. The default k-safety factor is 1 (two copies of the data) but can be set higher than 1 to protect against more than one simultaneous node failure.

While a higher k-safety factor can improve node reliability in the Enterprise Mode deployment model, it also increases storage capacity and write performance requirements. By contrast, the disaggregated storage and compute architecture of the Vertica in Eon Mode stores only one copy of the data in the S3 communal storage, regardless of the k-safety factor setting.

Figure 11 shows the capacity savings with Vertica in Eon Mode by comparing the storage consumed by an identical 10TB database for Vertica in Eon and Vertica in Enterprise Modes with a default k-safety factor of '1'. The 10TB (Decision Support style) database is compressed to 4TB using Vertica's advanced columnar compression. Vertica in Enterprise Mode requires two copies with k=1, so the total amount of storage used was 8TB. FlashBlade further compressed the Vertica in Eon Mode data to 3.8TB (~5%) and only needed a single copy.

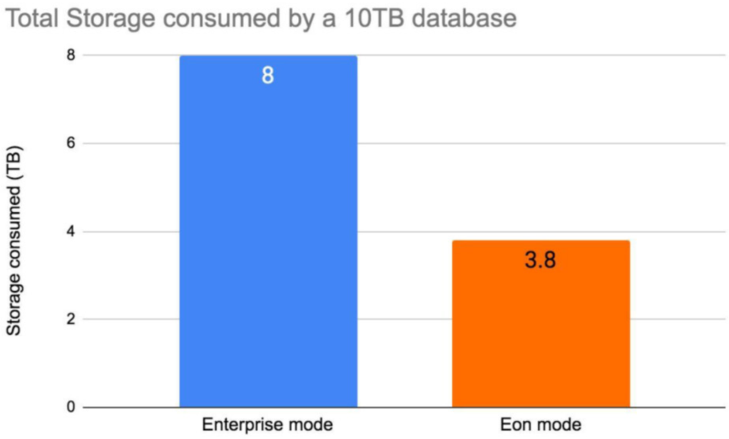


Figure 11. Capacity savings with Vertica in Eon Mode

Operational Efficiency

The disaggregated compute and storage architecture provide operational efficiency by eliminating the complexity that results from node proliferation at scale, thereby simplifying both scaling and patching. Figure 12 shows the time in minutes that it took to scale a Vertica database cluster from 8 compute nodes to 16 nodes using a live 10TB database.

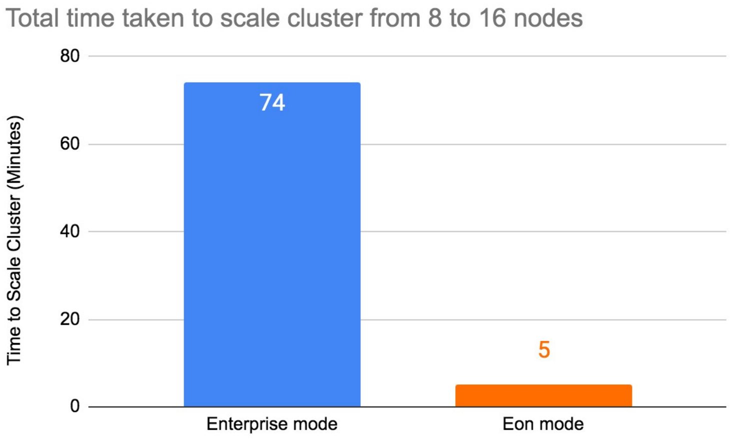


Figure 12. Vertica in Enterprise Mode vs. Vertica in Eon Mode operational efficiency

When scaling in Vertica in Enterprise Mode, data must be redistributed after new nodes are added. In testing on a database that was online but not running any queries, the database rebalance operation initiated after adding eight additional nodes took 74 minutes to complete. By contrast, in the Vertica in Eon Mode deployment, virtually redistributing an identically sized database took only five minutes. With Vertica in Eon Mode, this becomes a simple logical mapping operation without any real data movement. The result is a 15x improvement in the time to scale the cluster environment.

It is expected that the time to redistribute data will scale with the size of the database, k-safety factor, and ongoing database loads and queries. For production databases, rebalancing data is also likely to impact query and data loading performance. Many enterprises use Vertica to run databases that are hundreds (or thousands) of terabytes in size and have a k-safety factor greater than 1. Since rebalancing a database across nodes would impact performance, the Vertica in Eon Mode deployment provides benefits to both administrative time and user performance.

Increasing the number of nodes from 8 to 16 results in an almost 50% reduction in query runtime. The result underscores the flexibility and advantages of using Vertica in Eon Mode with Pure Storage FlashBlade. Importantly, note that the time required to rebalance a database grows with larger database sizes in the Enterprise Mode deployment model, whereas it remains constant with Eon Mode deployments, making it easier for the system to scale up and down to meet business needs.

Database Performance: Vertica in Enterprise Mode vs. Vertica in Eon Mode

FlashBlade storage is optimized for high-bandwidth applications and well-suited for Vertica's ROS (Read-Optimized-Store) data format. Testing indicates that the performance of Vertica in Eon Mode databases is either on par or better than the performance in Enterprise Mode for identical clusters.

For a typical ETL operation database load time was tested and the write Depot setting was varied to quantify its impact on database loads. When the UseDepotForWrites parameter is enabled, data is written into the Depot before getting flushed into S3 storage. Enterprise Mode took the longest time (71 minutes) to ingest the 10TB of data, as the database must write two copies of data, at minimum, to protect from node failures. Increasing the k-safety factor can further increase the data load times of Enterprise Mode databases. In a test environment in Eon Mode, the same 10TB data was loaded in 54 minutes with write Depot enabled and 43 minutes with write Depot disabled. This is subject to change based on the specific environment and data being loaded. Figure 13 compares the data load time for a 10TB Vertica database in both Enterprise and Eon Modes.

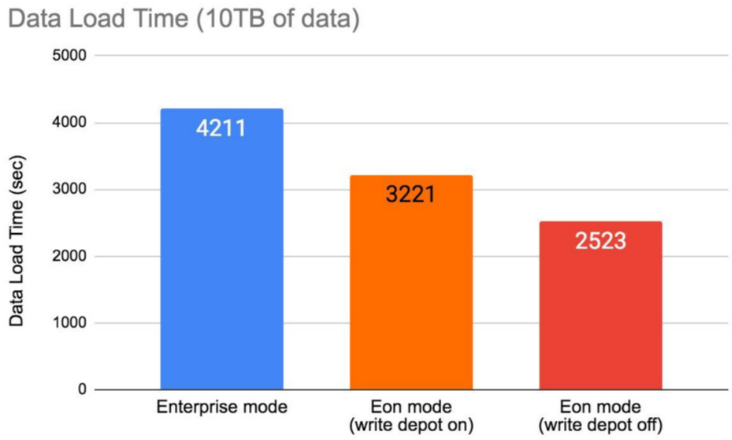


Figure 13. Data load times – Enterprise and Eon Mode

For a database query performance test concurrent database access was mimicked by running the same set of queries from four concurrent user sessions. Additionally, the read Depot setting in the Eon Mode tests was toggled to verify its impact on performance. The test results show that Vertica in Eon Mode with read Depot enabled delivered slightly better performance than Vertica in Enterprise Mode. The impact of read Depot can vary significantly based on data access patterns, database size, and Depot storage technologies. Experiment with this setting to determine the right value for a given environment. Figure 14 shows the database query run time on a 10TB Vertica in Enterprise Mode database and compares it to an identically sized database deployed in Eon Mode.

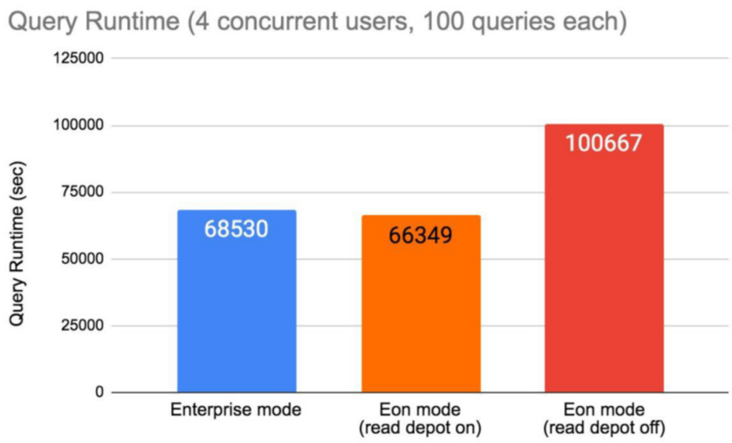


Figure 14. Database query performance – Vertica in Enterprise vs. Vertica in Eon Mode

Deployment

This section of the document describes the best practices for deployment of Vertica in Eon Mode on Pure FlashBlade storage. It provides guidelines for installing and configuring the hardware and software and describes the configuration options to implement an optimal on-premises analytical platform solution.

Deployment Requirements

Vertica in Eon Mode analytical platform must be installed on dedicated physical and/or virtual servers and Pure FlashBlade hardware. The physical and/or virtual servers should be dedicated to Vertica. Pure FlashBlade is designed as a Data Hub and can simultaneously support a broad range of applications and can be shared. It's common to use FlashBlade for hosting related analytics applications like Kafka, Spark, GitLab, Docker Registry, JupyterHub, etc.

Compute Requirements

The required server is a commodity x86-64 architecture with a recommended minimum configuration of 16 cores, 128GB RAM, and 2TB of local protected storage. Vertica requires a minimum read/write speed of 40 MB/s per physical core of the CPU. However, for best performance, plan on 60–80MB/s per physical core.

Storage Requirements

Vertica supports communal storage on Pure Storage FlashBlade version 3.0.0 and later.

Network Requirements

The corporate network and compute node architecture determines the type of networking used within the Vertica platform. Physical servers connect to a Top-of-Rack (ToR) switch for traditional three-layer network topology or connect to a leaf switch in a Leaf/Spine topology. Vertica in Eon Mode nodes connect to one or more network switches, forming a cluster. The physical network infrastructure enables I/O operations between the Vertica nodes and the FlashBlade storage objects and provides connectivity to applications and end-users who require access to the analytical database. Figure 15 shows the network architecture of Vertica nodes with FlashBlade storage.

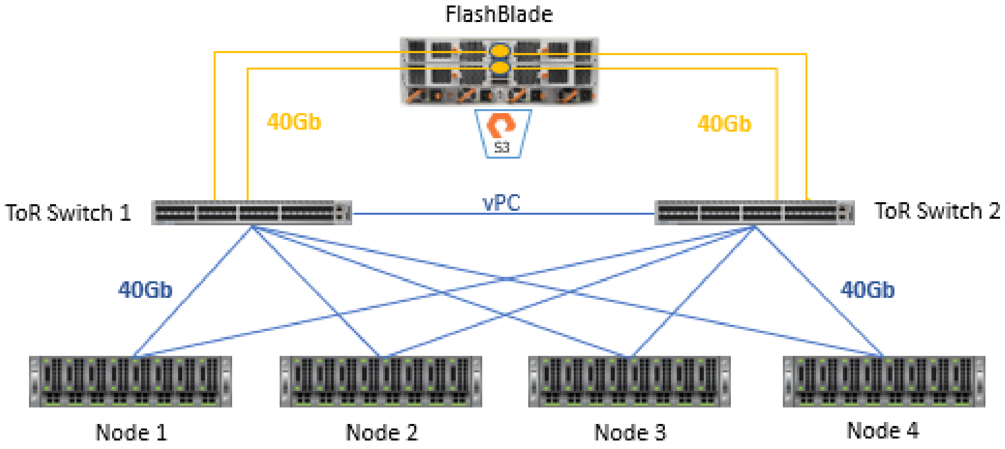


Figure 15. Vertica Nodes and FlashBlade Network Architecture

When a physical server is used for bare metal deployments, they require connectivity to physical switches and require the Linux OS to properly configure the IP addressing for the Network Interface Cards (NIC). When used within a virtual environment, the physical node running the hypervisor has the physical NICs configured for the appropriate subnet, but the virtual machines must be configured to use virtual distributed switches with each virtual machine configured with the appropriate number and speed of the virtual NICs (vNICs). The best practice is to use Virtual LAN (VLAN) IDs within the data center network to segregate Vertica in Eon Mode with FlashBlade network traffic. The ideal network would consist of three VLANs: management, private network for cross-node traffic, and storage network.

Linux OS Deployment

For physical hardware deployments, a Linux Operating System must be installed on the bare metal servers. For virtual compute infrastructure environments, the Hypervisor must be installed on the bare metal and a Linux OS guest must be installed on the required number of servers to meet the performance requirements of the solution. Installing the Linux on bare metal or as a virtual machine is outside the scope of this PVD guide. Please follow Linux vendor instructions for installation. Supported operating systems include:

- Red Hat Enterprise Linux versions 6.6, 6.7, 6.8, 6.9, 7.0, 7.4
- CentOS versions 6.6, 6.7, 6.8, 6.9, 7.0, 7.4
- SUSE Linux Enterprise Server Version 12 SP2
- Oracle Enterprise Linux versions 6.7, 6.8, 6.9, 7.3
- Debian Linux versions 7.6, 7.7, 8.5, 8.9
- Ubuntu Version 14.04 LTS

Linux OS File System Format

Choose the storage format type based on deployment requirements. Vertica supports the following storage format types where applicable:

- ext3 (older format and not recommended for new deployments)
- ext4
- XFS

Linux Storage Requirements for Vertica in Eon Mode

To prepare the Linux OS for the Vertica installation, create and specify directories in which the installation will store the catalog and data files (physical schema). These will be requested during the install and configuration of the database.

In addition to actual data stored in the database, Vertica requires disk space for several data reorganization operations, such as merge out and managing nodes in the cluster. For best results, Vertica recommends that disk utilization per node be no more than sixty percent for a K-Safe=1 database to allow such operations to proceed. In addition, disk space is temporarily required by certain query execution operators, such as hash joins and sorts, in the case when they cannot be completed in memory (RAM). Such operators might be encountered during queries, recovery, refreshing projections, and so on.

The amount of disk space needed (known as temp space) depends on the nature of the queries, amount of data managed by each node, node memory size node, and the number of concurrent users on the system. By default, any unused disk space on the data disk can be used as a temp space. However, Vertica recommends provisioning temp space separate from data disk space. An example of how to compute storage is allocated: 200GB for the boot disk, 2TB of SSD for Depot, merge, catalog, and temp space.

Deployment Guide

Vertica in Eon Mode analytical platform must be installed on dedicated physical and/or virtual servers and Pure FlashBlade hardware. Although the FlashBlade can be shared with other applications, Vertica requires the servers and Pure FlashBlade hardware to be dedicated to the analytical database.

Install Pure FlashBlade Storage

Pure offers three installation options to install a Pure FlashBlade array at a customer location. Customers can choose to install the FlashBlade themselves, Pure can assist with a remote install, or customers can choose to have Pure or an authorized partner perform a complete white-glove installation. This guide assumes the successful deployment of Pure FlashBlade.

Configure the Pure FlashBlade Storage

Prerequisites

The following are required to use a Pure Storage FlashBlade appliance as a communal storage location for a Vertica in Eon Mode database:

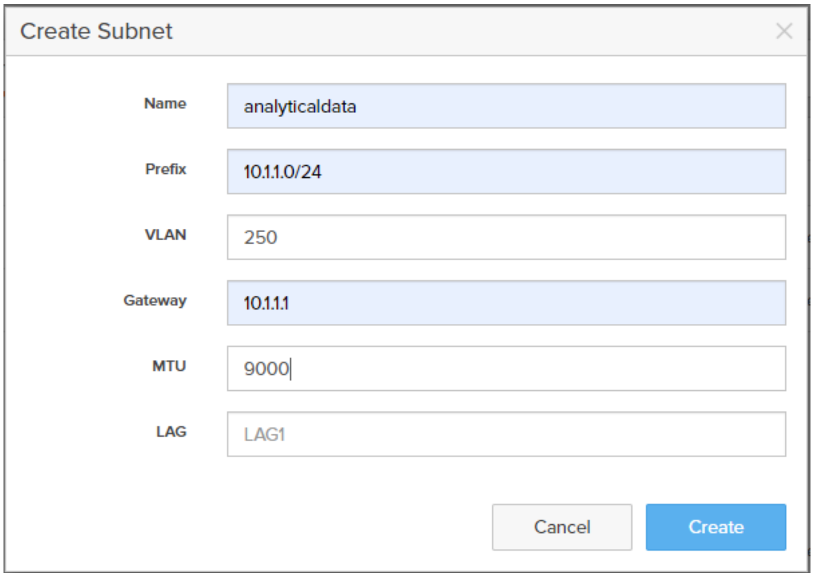
- Each FlashBlade Storage system will require at least two IP address: one for management, and another for storage access. The storage access IP address should be on the same subnet and VLAN as at least one compute node network interface. All nodes in the Vertica cluster must be able to access these IP addresses.

- Firewalls between the FlashBlade storage and the compute nodes must be properly configured to allow access between the systems.
- Network Service ports must be open for proper cluster communications. The default ports that must be open with the host's services file at 80 or 443 to access the bucket
- Identify VLANs to segregate data and management traffic.
- Access to communal data requires bucket access and secret keys for user read and write access.

Configure the Pure FlashBlade Network

Create a subnet for the analytical database data traffic (Figure 16).

1. In the Purity//FB GUI, select Settings > Network.
2. In the Subnets list, click the Add (+) button in the Subnets title bar. The Create Subnet window appears.
3. In the Name field, type the name of the subnet.
4. In the Prefix field, type the IP address of the network prefix and prefix length in the form `ddd.ddd.ddd.ddd/dd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx/xxx` for IPv6.
5. In the VLAN field, specify the VLAN ID to which the subnet is configured. Valid VLAN ID numbers are between 1 and 4094.
6. In the Gateway field, type the IP address of the gateway through which the Vertica Database data communicates with the network in the form `ddd.ddd.ddd.ddd` for IPv4, or `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx` for IPv6
7. Use jumbo frames for the maximum transmission unit (MTU) of the data network interface. In the MTU field, specify 9000 for the value (Jumbo Frames are recommended for Vertica deployments). If the MTU is not specified during subnet creation, the value defaults to 1500.
8. Click Create.



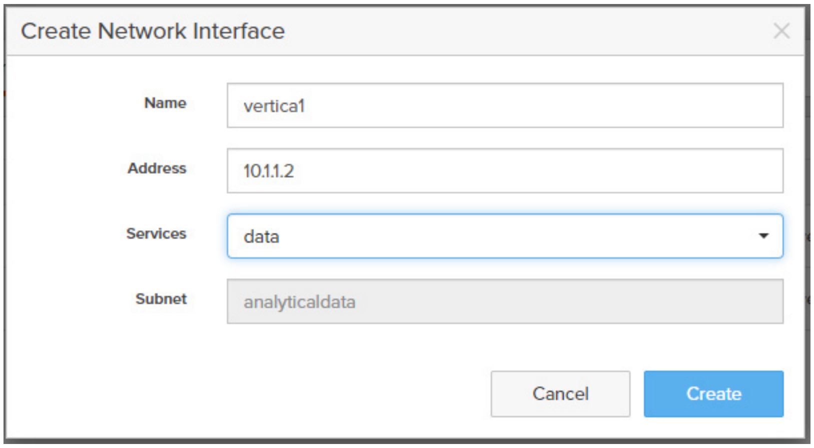
Create Subnet

Name	analyticaldata
Prefix	10.1.1.0/24
VLAN	250
Gateway	10.1.1.1
MTU	9000
LAG	LAG1

Figure 16. Create Subnet window

Create a network interface on the analytical database data traffic subnet (Figure 17).

1. Select Settings > Network.
2. In the Subnets list, find the subnet with the correct network prefix, VLAN ID, and gateway. The data interface will be attached to this subnet for file export purposes. Create a subnet if none of the existing ones meet the requirements.
3. Click the Add interface button (+) belonging to the subnet to which the data interface will be attached. The Create Network Interface window appears.
4. In the Name field, type the name of the data interface.
5. In the Address field, type the IP address to be associated with the data interface.
6. In the Services field, leave the service type as data.
7. In the Subnet field, leave the subnet name as the default.
8. Click Create.



Create Network Interface

Name: vertica1

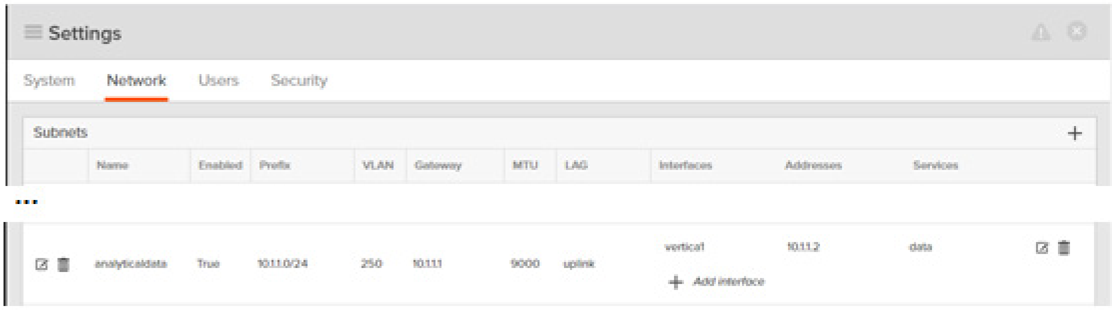
Address: 10.1.1.2

Services: data

Subnet: analyticaldata

Figure 17. Create Network Interface window

9. View the new subnet in the Subnets list (Figure 18).



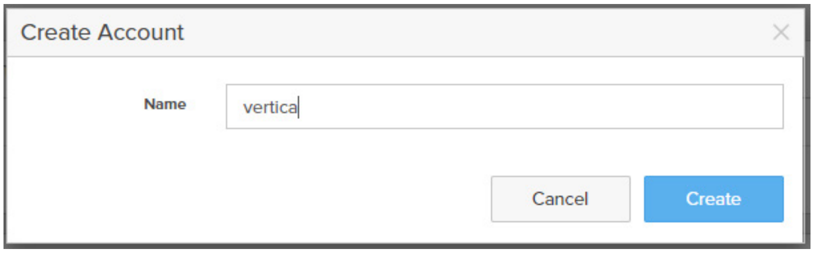
	Name	Enabled	Prefix	VLAN	Gateway	MTU	LAG	Interfaces	Addresses	Services
	analyticaldata	True	10.1.1.0/24	250	10.1.1.1	9000	uplink	vertica1 + Add interface	10.1.1.2	data

Figure 18. List Subnets

Provision Object Bucket for Vertica Database

Before installing the Vertica binaries, create a bucket for the Vertica in Eon Mode database data. Buckets are organized into accounts.

1. To create a new account:
   a. In the Accounts section of the Storage > Object Store page, click the Add (+) button. The Create Account window appears (Figure 19).
   b. Enter the new account's name `vertica` in the Name field.
   c. Click Create.

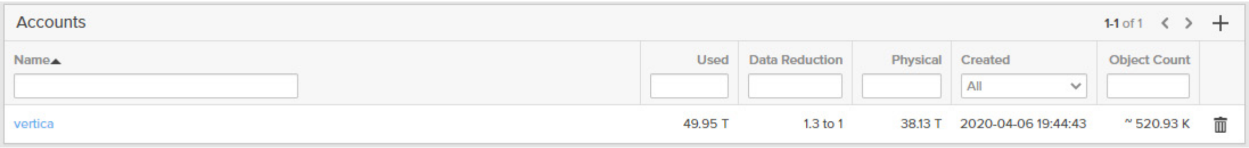


Create Account

Name: vertica

Figure 19. The Create Account window

   d. View the Vertica account listed in the Accounts pane. Click the account name to open its details (Figure 20).

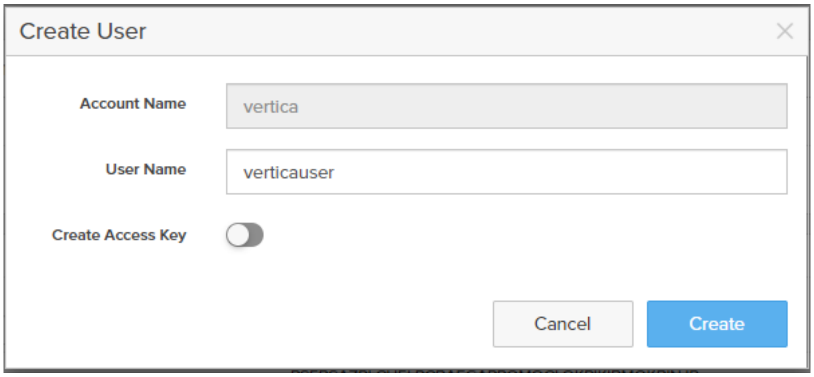


Accounts

Name▲	Used	Data Reduction	Physical	Created	Object Count
				All	
vertica	49.95 T	1.3 to 1	38.13 T	2020-04-06 19:44:43	~ 520.93 K

Figure 20. Accounts pane

2. Within the account, create a user, which will have an access key associated with it. To create a new user:
   a. From the Storage > Object Store page, click the account to add a new user.
   b. In the Users section, click Add add (+) button. The Create User window appears (Figure 21).
   c. Enter the `dbadmin` name in the User Name field.
   d. Do not enable the Create Access Key option; this will happen in a later step.
   e. Click Create.



Create User

Account Name: vertica

User Name: verticauser

Create Access Key: ○

Figure 21. Create User window

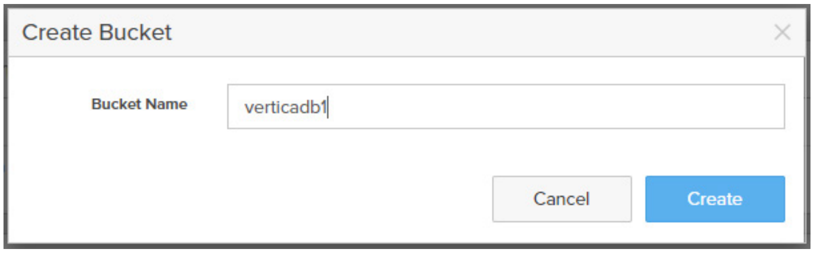
3. Within the same account, create a bucket to store the objects Vertica writes. To create a bucket:

a. From the Storage > Object Store page, click the account to add a bucket.

b. From the Buckets window, click the Add (+) button. The Create Bucket window appears (Figure 22).

c. Enter a name for the bucket. For this example, the bucket has the name verticadb1.

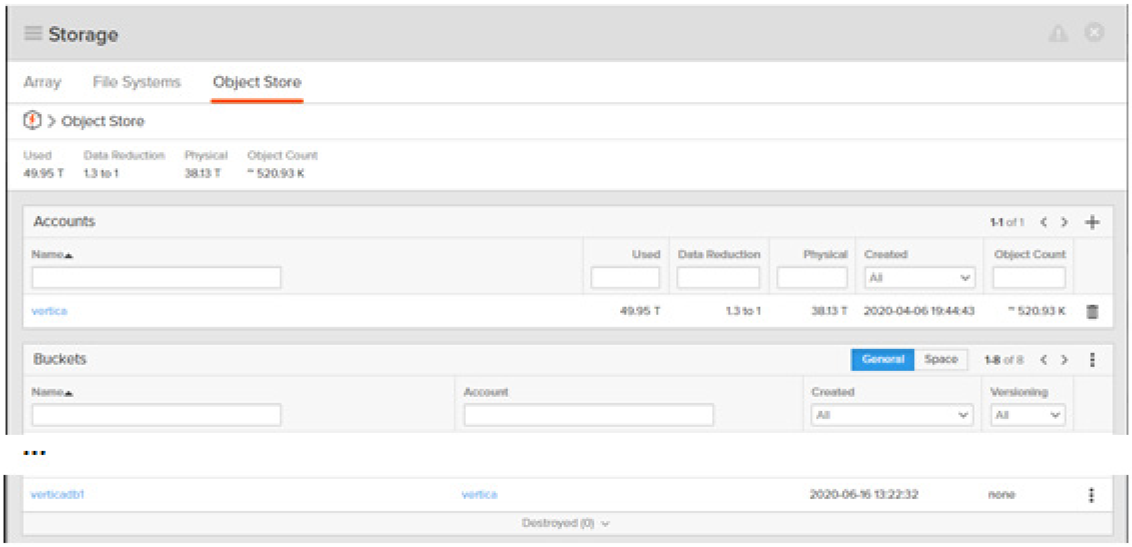
d. Click Create.



Bucket Name: verticadb1

Figure 22. Create Bucket window

4. Verify there are a user and an empty bucket within the object store account just created (Figure 23).



Used	Data Reduction	Physical	Object Count
49.95 T	1.3 to 1	38.13 T	~ 520.93 K

Name	Used	Data Reduction	Physical	Created	Object Count
				All	
vertica	49.95 T	1.3 to 1	38.13 T	2020-04-06 19:44:43	~ 520.93 K

Name	Account	Created	Versioning
		All	All
verticadb1	vertica	2020-06-16 13:22:32	none

Figure 23. New User and Bucket

Optional: Add Trusted Certification Authority (CA) Certificate

If there is a requirement to use TLS to secure the Amazon S3 connection between Vertica clients and the FlashBlade object store, update the FlashBlade with a certificate generated by a trusted certification authority (CA). To import a CA certificate:

1. Select **Settings** > **System**.

2. Click the **More Options** button from the **SSL Certificate** panel.

3. Click Import Certificate.

4. Complete or modify the following fields:
   a. **Certificate** - Click **Choose File** and select the signed certificate. Verify the certificate is PEM formatted (Base64 encoded); and that it includes the " -----BEGIN CERTIFICATE----- " and " -----END CERTIFICATE----- " lines.
   b. **Private Key** - Click **Choose File** and select the private key.
   c. **Intermediate Certificate** - (Optional) Click **Choose File** and select the intermediate certificate.
   d. **Key Passphrase** - (Optional) If the private key is encrypted with a passphrase, enter the passphrase.
5. Click **Import**. The page will refresh in several seconds.

Generate Access Key for Object User on FlashBlade

Access Keys are used to sign the requests sent to FlashBlade S3 Object Bucket storage. Like the Username/Password pair, used to access a management console, Access Key Id and Secret Access Key are used for programmatic (API) access to S3 Object services. For the Vertica in Eon Mode FlashBlade storage, manage Access Keys in Purity//FB GUI.

Create an access key for the object user created on FlashBlade.

1. Open the Purity//FB GUI.
2. From the **Storage > Object Store** page, click the Vertica account to which the new user was recently added.
3. To generate an access Key ID and a Secret Access Key navigate to the Users section, click the **More Options > Create** access key button on the same row as the user for which to create an access key (Figure 24).

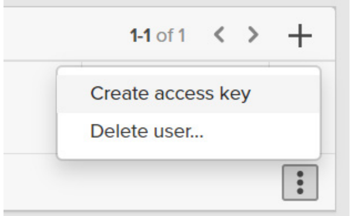
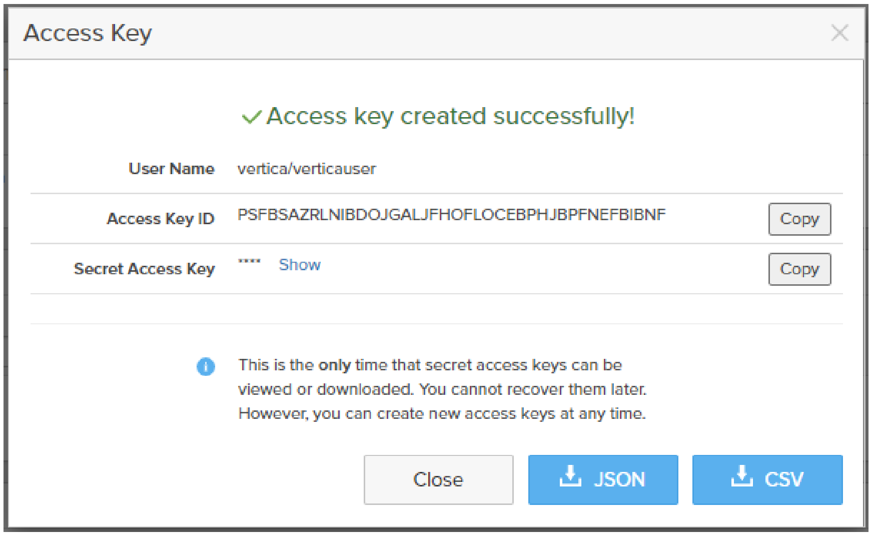


Figure 24. More Options menu

A confirmation pop-up window will display (Figure 25).


Access Key

✓ Access key created successfully!

User Name vertica/verticauser

Access Key ID PSFBSAZRLNIBDOJGALJFHOFLOCEBPHJBPFNEFBIBNF

Secret Access Key **** Show

This is the **only** time that secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Figure 25. Access Key confirmation pop up window

Download the Vertica Server Package

Navigate to the Vertica software page and download the version of Vertica that aligns with the version of Linux running on the compute server. To download and install the Vertica server package:

1. Use a Web browser to log in to Vertica https://www.vertica.com/register/.
2. Click the Download tab and download the Vertica server package to an Administration Host. Running the Administration Tools on the Administration Host, whenever possible, is strongly recommended.

Install Vertica in Eon Mode Database

The installation steps are the same ones followed to install Vertica in Enterprise Mode. The difference between Eon Mode and Enterprise Mode on-premises databases is how the database is created, not how Vertica software is installed. During the Vertica software installation, the user `dbadmin` was created.

To install Vertica, use the `install_vertica` script to verify the nodes are correctly configured and to install the Vertica binaries on all the cluster nodes.

1. To install, log into the Administration Host as root (or log in as another user and switch to root). Root privileges are only required to perform the install, after Vertica is installed, root privileges are no longer needed.

   ```
   $ su - root
   password: root-password
   #
   ```

2. As root installing an RPM:

   ```
   # rpm -Uvh vertica-x.x.x-x.x.rpm
   ```

   Vertica is installed under `/opt/vertica`.

3. Install Vertica on the other nodes in the cluster. Run the following command on the Administration Host:

   ```
   # cd /opt/vertica/sbin/
   #./install_vertica -s host_list -r rpm_package
   ```
   -s is the hostname/IP of all the nodes of the cluster, including the one on which Vertica was just previously installed.
   -r is the path of the Vertica package

Prerequisites for Creating Vertica in Eon Mode Database

Before creating the Vertica in Eon Mode on-premises database, create an authorization file that admintools will use to authenticate with the FlashBlade storage. The `auth_params.conf` file contains the secret key to access the bucket containing the Eon Mode database's data. This information is sensitive and can be used to access the raw data in the database. Be sure this file is not readable by unauthorized users. After creating the database, delete this file.

1. Connect to the Administration Host where admintools have been installed.
2. Use a text editor to create a file, named `auth_params.conf`
   The location of this file is not important, as long as it is readable by the dbadmin Linux user assigned to create the database.

3. Add the following lines to the file:

```
awsauth = FlasbBlade_Access_Key:FlashBlade_Secret_Key
awsendpoint = FlashBladeIP:FlashBladePort
```

4. Use the Access Key ID and Secret Key generated earlier. It is not necessary to supply a port number in the awsendpoint setting if the default port for the connection between Vertica and the FlashBlade (80 for an unencrypted connection or 443 for an encrypted connection) was used.

   If not using TLS encryption for the connection between Vertica and the FlashBlade, add the following line to the file:

```
awsenablehttps = 0
```

5. Save the file and exit the editor.

This example `auth_params.conf` file is for an unencrypted connection between the Vertica cluster and a FlashBlade storage at IP address 192.168.10.99 using the standard port 80.

```
awsauth = PIWHSNDGSHVRPIQ:339068001+e904816E02E5fe9103f8MQOEAEHFFVPKBAAL
awsendpoint = 192.168.10.99
awsenablehttps = 0
```

Choose a Depot Path on All Nodes

Choose or create a directory on each node for the Depot storage path.

The directory supplied for the Depot storage path parameter must:

- Have the same path on all nodes in the cluster (i.e. `/home/dbadmin/Depot`).
- Be readable and writable by the `dbadmin` user.
- Have sufficient storage. By default, Vertica uses 60% of the filesystem space containing the directory for Depot storage. It is possible to limit the size of the Depot by using the `--Depot-size` argument in the `create_db` command.

The `admintools create_db` tool will create the Depot path automatically if it does not exist.

Create the Vertica in Eon Mode On-Premises Database

Use the `admintools create_db` tool to create the database. The best practice is to create your Eon Mode databases using additional prefix paths in the bucket (e.g., `s3://vertica/prod/dw01/` instead of `s3://vertica/dw01/`). This makes it simple to use Fast CopyObject to clone the database into additional prefixes (e.g., test, dev, etc.).

```
admintools -t create_db –help
```

The following example demonstrates creating a three-node database named verticadb, specifying the Depot will be stored in the home directory of the dbadmin user.

```
$ admintools -t create_db -x auth_params.conf \
  --communal-storage-location=s3://verticadbbucket/prod/verticadb \
  --Depot-path=/home/dbadmin/Depot --shard-count=6 \
  -s vnode01,vnode02,vnode03 -d verticadb -p 'DBPasswordHere'
```

Argument	Description
`-x`	The path to the `auth_params.conf` file.
`--communal-storage-location`	The S3 URL for the bucket on the FlashBlade appliance (usually, this is s3://bucketname).
`--Depot-path`	The absolute path to store the Depot on the nodes in the cluster.
`--shard-count`	The number of shards for the database. This is an integer number that is usually either a multiple of the number of nodes in the cluster or an even divider.
`-s`	A comma-separated list of the nodes in the database.
`-d`	The name for the database.
`-l`	The absolute path to the Vertica license file to apply to the new database.
`-p`	The password for the new database.
`--Depot-size`	The maximum size for the Depot. Defaults to 60% of the filesystem containing the Depot path.

Validate Database Compliance

Login as `dbadmin`, connect to the database, and check its status. Verify the nodes are up and are in compliance. From the output, verify the node requested is up and that the Compliance Status displays the database complies with the raw data size.

```
[dbadmin@sn1-r720-g03-27 ~]$ vsql -U dbadmin
Welcome to vsql, the Vertica Analytic Database interactive terminal.
Type:  \h or \? for help with vsql commands
       \g or terminate with semicolon to execute query
       \q to quit

dbadmin=> SELECT node_name, node_state FROM nodes ORDER BY 1;
      node_name       | node_state
----------------------+------------
 v_verticadb_node0001 | UP
 v_verticadb_node0002 | UP
 v_verticadb_node0003 | UP
 v_verticadb_node0004 | UP
 (4 rows)
dbadmin=> SELECT node_name, user_name, client_os FROM v_monitor.sessions;
      node_name       | user_name |             client_os
----------------------+-----------+------------------------------------
 v_verticadb_node0002 | dbadmin   | Linux 3.10.0-693.el7.x86_64 x86_64
 v_verticadb_node0001 | dbadmin   | Linux 3.10.0-693.el7.x86_64 amd64
 v_verticadb_node0001 | dbadmin   | Linux 3.10.0-693.el7.x86_64 amd64
 v_verticadb_node0001 | dbadmin   | Linux 3.10.0-693.el7.x86_64 amd64
 v_verticadb_node0001 | dbadmin   | Linux 3.10.0-693.el7.x86_64 amd64
 v_verticadb_node0001 | dbadmin   | Linux 3.10.0-693.el7.x86_64 amd64
 v_verticadb_node0001 | dbadmin   | Linux 3.10.0-693.el7.x86_64 amd64
(7 rows)
dbadmin=> SELECT GET_COMPLIANCE_STATUS();
                                GET_COMPLIANCE_STATUS
------------------------------------------------------------------------------------
  No size-compliance concerns for an Unlimited license
 License End Date: 2025-10-26
 Days Remaining: 1836.77
(1 row)
dbadmin=>
```

Upgrade

Pre-Upgrade Process

The Vertica installer checks the target platform as it runs and stops whenever it determines the platform fails to meet an installation requirement. Before updating the server package on a cluster, manually verify the platform meets all hardware and software requirements.

Before performing an upgrade to the Vertica in Eon Mode database take the following precautionary steps to help ensure the upgrade process succeeds.

1. Perform a full database backup so that the current version can be restored if the upgrade is unsuccessful.
2. Perform a backup of all grants.
3. Verify platform requirements for the new version.
4. Identify and remove unsupported projections. In all post-9.0 versions of Vertica, support has been removed for projection buddies with different SELECT and ORDER BY clauses. Support also has been removed for pre-join and range segmentation projections. If the upgrade encounters unsupported projections, it is liable to fail.
5. Check catalog storage space.

   Verify that there is enough RAM available to run the upgrade. The upgrade requires approximately three times the amount of memory the database catalog uses.

   Catalog memory usage on all nodes can be calculated by querying system table `RESOURCE_POOL_STATUS:`

   ```
   => SELECT node_name, pool_name, memory_size_kb FROM resource_pool_status WHERE pool_name =
   'metadata';
   ```

By default, the installer stops on all warnings. Administrators can configure the level where the installer stops installation, through the installation parameter --failure-threshold. If the threshold is set to FAIL, the installer ignores warnings and stops only on failures.

After completing these tasks, shut down the database gracefully.

Upgrade Process

1. On each host where additional packages are installed, uninstall them (R language pack).

   If this step is omitted and additional packages are not uninstalled, the Vertica server package fails to install.

2. Log in as root or sudo and use one of the following commands to run the `# rpm -e vertica-R-lang` RPM package installer:

   If installing an RPM as root: `# rpm -Uvh pathname`

   If installing an RPM as sudo: `$ sudo rpm -Uvh pathname`

3. On the same node on which the RPM was installed, run `update_vertica` as root or `sudo`. This installs the RPM on all the hosts in the cluster.

```
# /opt/vertica/sbin/update_vertica --rpm /home/dbadmin/vertica-latest.x86_64.RHEL6.rpm --dba-user
dbadmin
```

The following requirements and restrictions apply:

- The DBADMIN user must be able to read the RPM file when upgrading.
- Use the same options used when the database was last installed or upgraded. These options can be found in `/opt/vertica/config/admintools.conf`, on the `install_opts` line.
- Omit the `--hosts/-s host-list` parameter. The upgrade script automatically identifies cluster hosts.
- If the root user is not in `/etc/sudoers`, an error appears. The installer reports this issue with S0311.

Note: If there are omitted options, default settings are restored. If any options have been changed, the upgrade script uses the new settings to reconfigure the cluster. This can cause issues with the upgraded database.

4. Start the database. The start-up scripts analyze the database and perform necessary data and catalog updates for the new version. If Vertica issues a warning stating that one or more packages cannot be installed, run the `admintools --force-reinstall` option to force the reinstallation of the packages.
5. When the upgrade is complete, the database automatically restarts.
6. Manually restart any nodes that fail to start up.
7. Perform another database backup.

Expanding Installation

Before expanding the Vertica in Eon Mode Database perform a full backup of the environment. Do not use the Management Console to add hosts to the Vertica in Eon Mode with Pure FlashBlade storage environment. However, after the hosts are added to the cluster, the Management Console can be used to add the hosts to a database as nodes.

Adding Hosts to a Cluster

Adding a host to a cluster can be performed using the `updates_vertica` script. The duration to expand the cluster is dependent on the average in-memory size of catalogs across all cluster nodes.

1. From one of the existing cluster hosts, run the `update_vertica` script with a minimum of the --add-hosts host(s) parameter (where host(s) is the hostname or IP address of the system(s) that is being added to the cluster) and the `--rpm` or `--deb parameter`:
   ```
   # /opt/vertica/sbin/update_vertica --hosts node01,node02,node03 --rpm /home/dbadmin/vertica-
   10.0.1.x86_64.RHEL6.rpm --dba-user dbadmin
   ```
2. Be sure to provide the same options that were used when originally installing the cluster. The `update_vertica` script uses all the same options as `install_vertica` and:
   - Installs the Vertica RPM on the new host.
   - Performs post-installation checks, including RPM version and N-way network connectivity checks.
   - Modifies spread to encompass the larger cluster.

- Configures the Administration Tools to work with the larger cluster.

3. If a package is specified with `--rpm`, and that package is newer than the one currently installed on the existing cluster, then Vertica installs the new package on the existing cluster hosts before the newly-added hosts.
4. Use the same command line parameters for the database administrator username, password, and directory path as were used when the cluster was originally installed. Alternatively, create a properties file to save the parameters during install and then re-use it on subsequent install and update operations.

   If installing using sudo, the database administrator user (`dbadmin`) must already exist on the hosts being added and must be configured with passwords and home directory paths identical to the existing hosts. Vertica sets up `passwordless ssh` from existing hosts to the new hosts, if needed.
5. The maximum number of spread daemons supported in point-to-point communication and broadcast traffic is 80. It is possible to have more than 80 nodes by using large cluster mode, which does not install a spread daemon on each node.

Adding Nodes from Subcluster

There will be conditions when an administrator needs to add new nodes to an existing subcluster. This is done to scale the database to respond to changing analytic needs. Add nodes from subcluster using either the Administration Tools or the Management Console. The `admintools` command line can be used to add nodes and preserve the specific order in which the nodes were added.

Add new nodes to a subcluster, using the `db_add_node` command of `admintools`. This is done to add nodes to a subcluster to meet additional workloads. The nodes being added to the subcluster must already be part of the cluster.

Monitoring

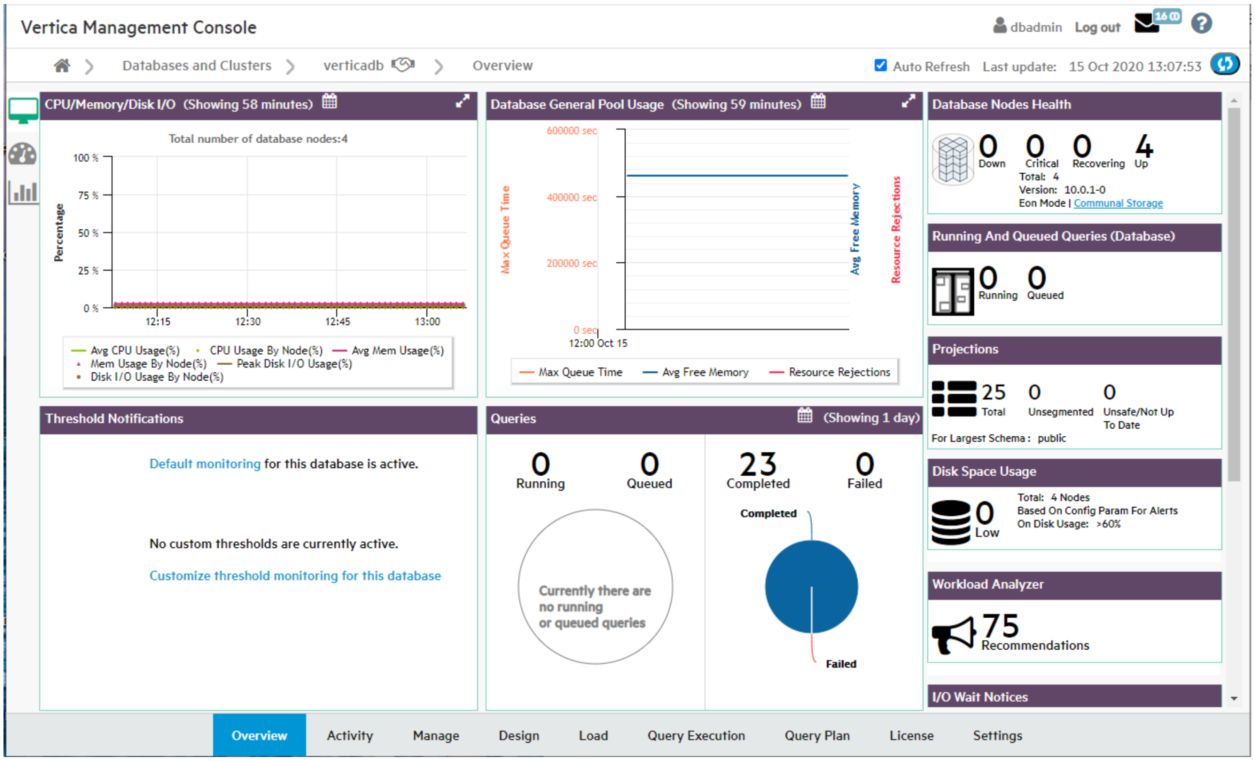
Vertica in Eon Mode monitoring is the tracking of infrastructure resources as well as the Vertica database. Monitoring is done to verify the operational state of the hardware. The optimized configuration state of the environment and maintain a high performance and highly available analytical infrastructure.

Monitoring Verica Database in Eon Mode

Monitoring Cluster using the Management Console Dashboard
Use the Management Console to monitor the health of the Vertica databases and clusters. On the Databases and Clusters page, click the database, to go to its Overview page.

Perform the following tasks from the Overview page:

1. View Quick Stats to get instant alerts and information about the cluster's status.
2. View Status Summary that provides a general overview of the status of the cluster (as shown in preceding Figure 26).
3. Analyze System Health using a comprehensive summary of the system resource usage and node information, with configurable statistics that allow administrators to specify acceptable ranges of resource usage.
4. Use Query Synopsis to monitor system query activity and resource pool usage.



Default monitoring for this database is active.

No custom thresholds are currently active.

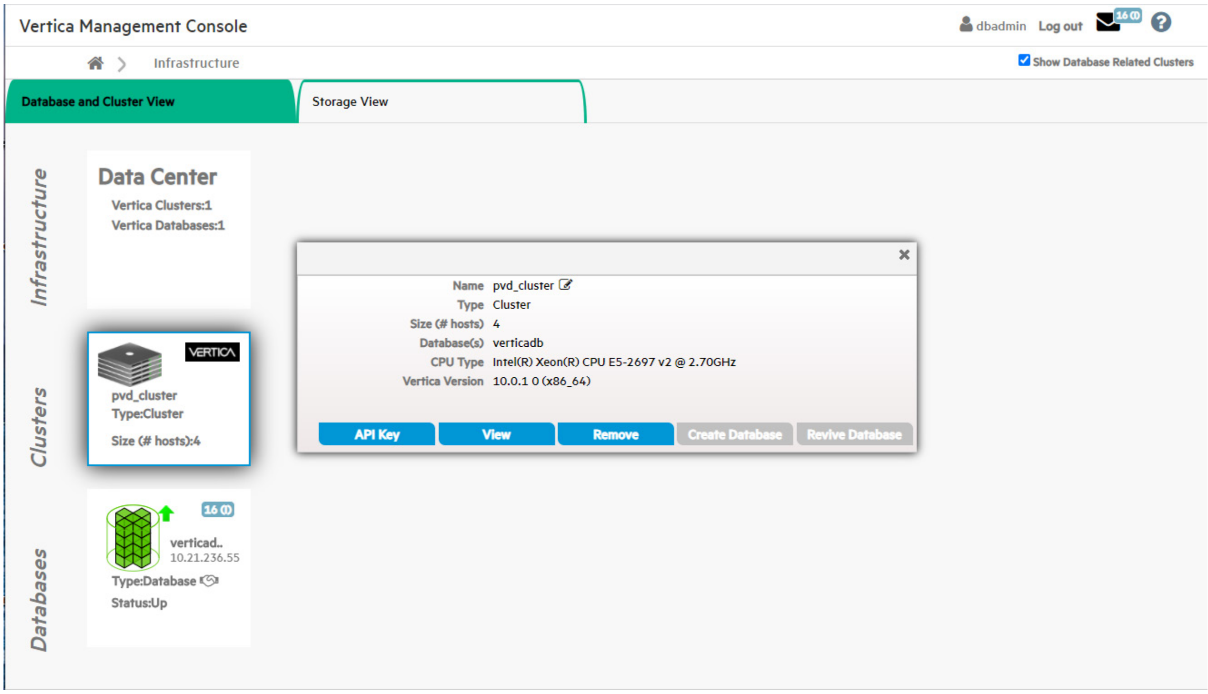
Customize threshold monitoring for this database

Currently there are no running or queued queries

Figure 26. Vertica Management Console Dashboard

Monitoring Existing Infrastructure with Management Console

Click the Infrastructure button on the Home page to see the Databases and Clusters page. Then click the cluster of interest to view the health of the nodes in that cluster and the key information associated with the cluster such as Vertica version, number of hosts, CPU type, last updated date, and node list (Figure 27).



Name pvd_cluster
Type Cluster
Size (# hosts) 4
Database(s) verticadb
CPU Type Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz
Vertica Version 10.0.1 0 (x86_64)

Figure 27. Vertica Management Console - Infrastructure summary

For Vertica in Eon Mode databases, the Status Summary and Query Synopsis pages allow admins to display information for the entire database. If subclusters are defined, admins can also display information for a specific subcluster or node, or the nodes not assigned to a subcluster.

Select Status Summary icon to display the Monitoring Cluster nodes, CPU/memory, and performance.

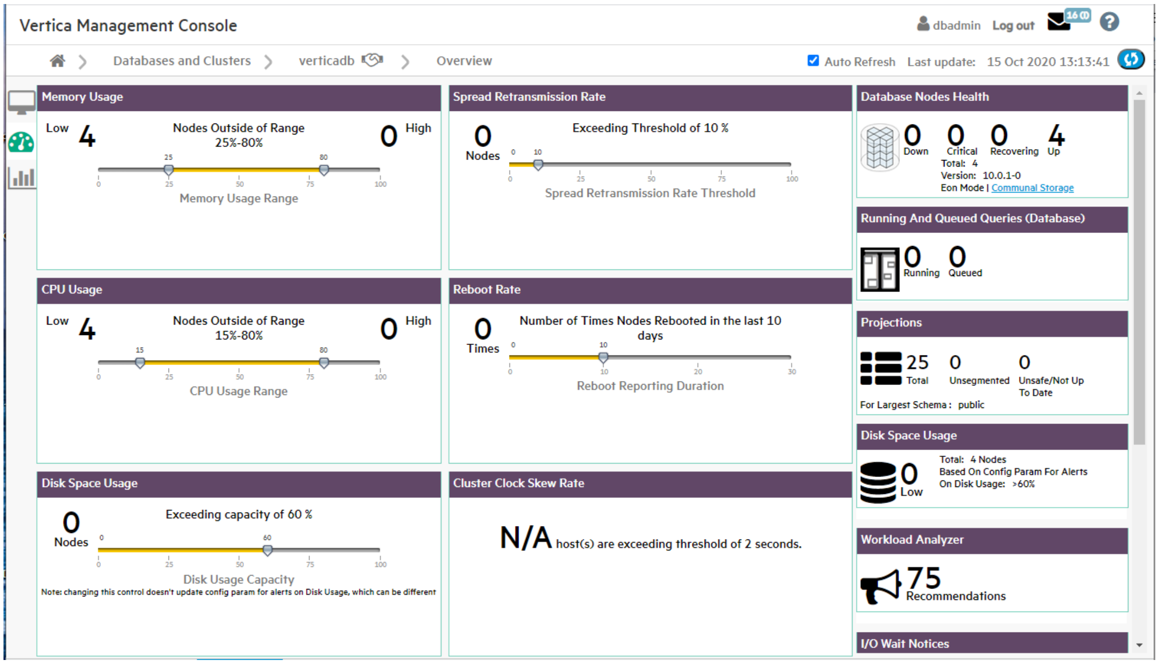

Vertica Management Console
dbadmin Log out
Databases and Clusters > verticadb > Overview
☑ Auto Refresh Last update: 15 Oct 2020 13:13:41
Memory Usage
Low 4 Nodes Outside of Range 25%-80% 0 High
Memory Usage Range
Spread Retransmission Rate
0 Nodes Exceeding Threshold of 10 %
Spread Retransmission Rate Threshold
Database Nodes Health
0 Down 0 Critical 0 Recovering 4 Up
Total: 4
Version: 10.0.1-0
Eon Mode | Communal Storage
Running And Queued Queries (Database)
0 Running 0 Queued
CPU Usage
Low 4 Nodes Outside of Range 15%-80% 0 High
CPU Usage Range
Reboot Rate
0 Times Number of Times Nodes Rebooted in the last 10 days
Reboot Reporting Duration
Projections
25 Total 0 Unsegmented 0 Unsafe/Not Up To Date
For Largest Schema : public
Disk Space Usage
0 Nodes Exceeding capacity of 60 %
Disk Usage Capacity
Note: changing this control doesn't update config param for alerts on Disk Usage, which can be different
Cluster Clock Skew Rate
N/A host(s) are exceeding threshold of 2 seconds.
Disk Space Usage
0 Low Total: 4 Nodes Based On Config Param For Alerts On Disk Usage: >60%
Workload Analyzer
75 Recommendations
I/O Wait Notices

Figure 28. Vertica Management Console - System Status

Select the Query Synopsis icon to view Query Statistics (Figure 29).

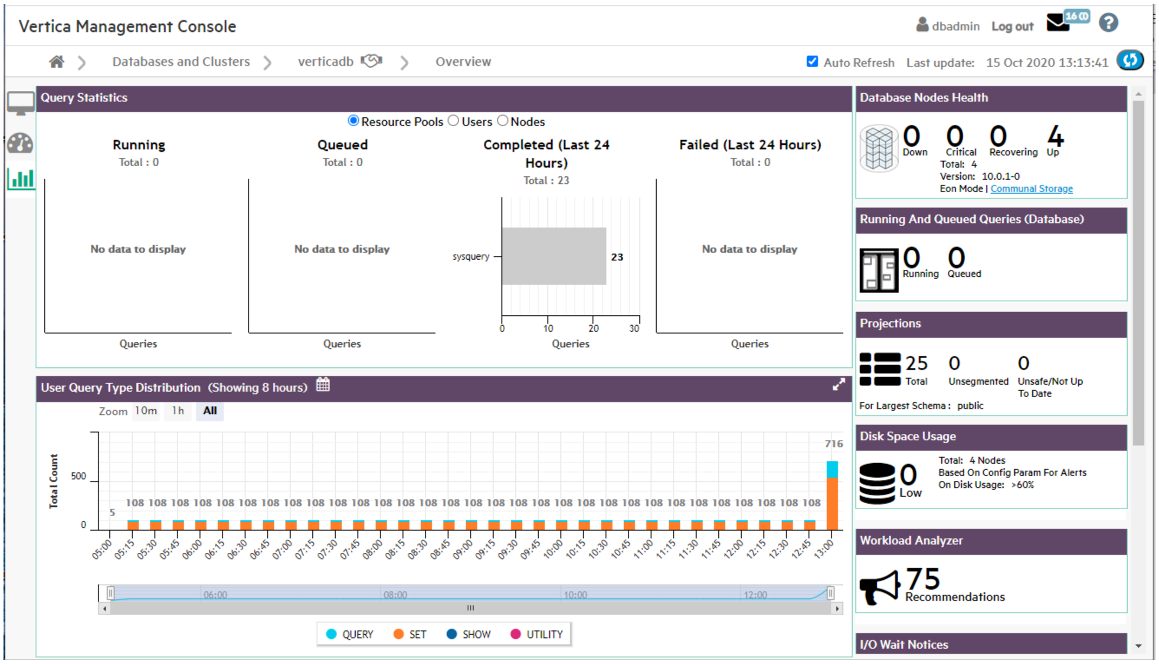
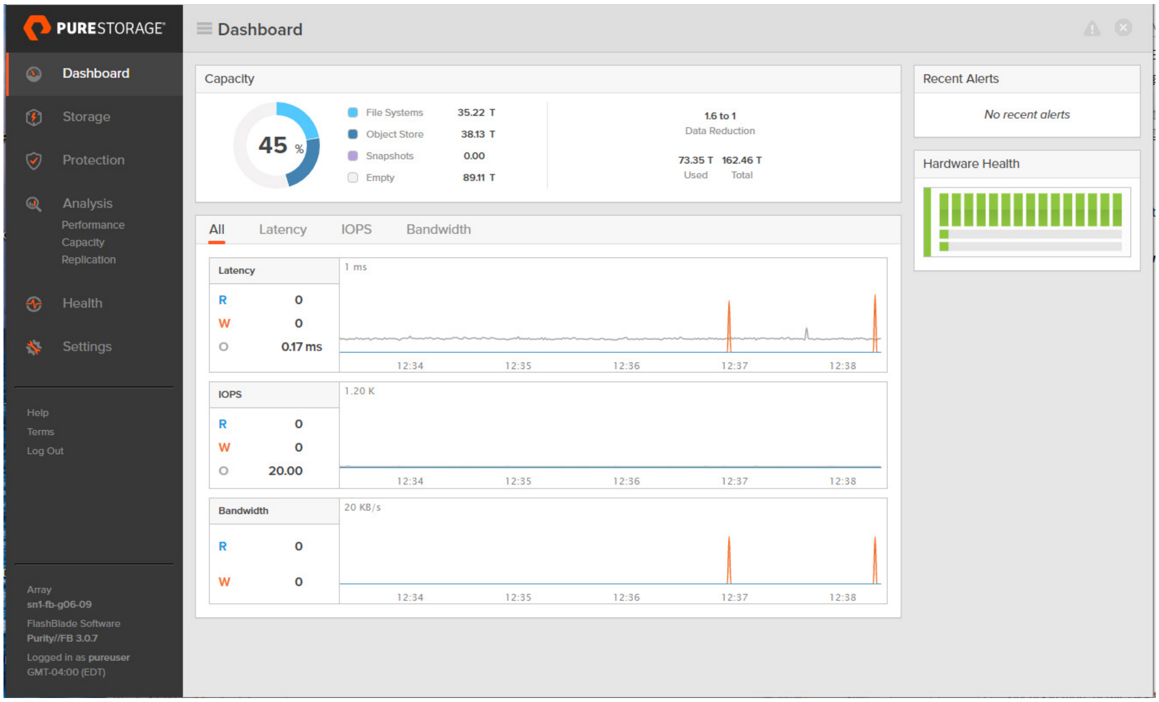


Figure 29. Vertica Management Console - Query Synopsis

Monitoring Pure FlashBlade

Monitoring Pure FlashBlade storage using the Purity//FB GUI Dashboard

Use Purity//FB GUI to monitor the health of the Pure FlashBlade storage system. The Dashboard page displays a running graphical overview of the array's storage capacity, performance, and hardware status as shown in Figure 30.



Capacity

45 %

File Systems	35.22 T
Object Store	38.13 T
Snapshots	0.00
Empty	89.11 T

1.6 to 1
Data Reduction

73.35 T Used
162.46 T Total

Recent Alerts

No recent alerts

Hardware Health

Latency	
R	0
W	0
O	0.17 ms

IOPS	
R	0
W	0
O	20.00

Bandwidth	
R	0
W	0

Array
sn1-fb-g06-09
FlashBlade Software
Purity//FB 3.0.7
Logged in as pureuser
GMT-04:00 (EDT)

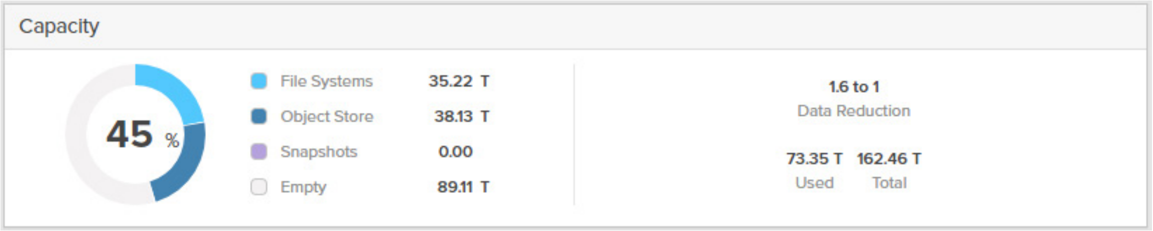
Figure 30. Purity//FB GUI Dashboard

The **Dashboard** page contains the following panels and charts:

- Capacity
- Performance Charts
- Recent Alerts
- Replication Bandwidth (only for replication setups)
- Hardware Health

Monitoring Capacity

The **Capacity** panel displays array size and storage consumption details as shown in Figure 31.



Capacity

45 %

File Systems	35.22 T
Object Store	38.13 T
Snapshots	0.00
Empty	89.11 T

1.6 to 1
Data Reduction

73.35 T Used
162.46 T Total

Figure 31. Dashboard - Capacity Panel

The capacity wheel displays the percentage of array space occupied by file system data. The percentage value in the center of the wheel is calculated as Used capacity/ Total capacity. All capacity values are rounded to two decimal places. The capacity data is broken down into the following components:

File Systems: Amount of space that the written data occupies on the array's file systems after reduction via data compression.

- Object Store: Amount of space that the written data occupies on the array's buckets after reduction via data compression.
- Snapshots: Amount of space that the array's snapshots occupy after data compression.
- Empty: Unused space.
- Used: Amount of space that the written data occupies on the array after reduction via data compression.
- Total: Total usable capacity of the array.
- Data Reduction: Ratio of the size of the written data (i.e. Used) versus the amount of space the data occupies after data compression (i.e. Physical).

Monitoring Performance

The performance panel displays latency, IOPS, and bandwidth values in real-time.

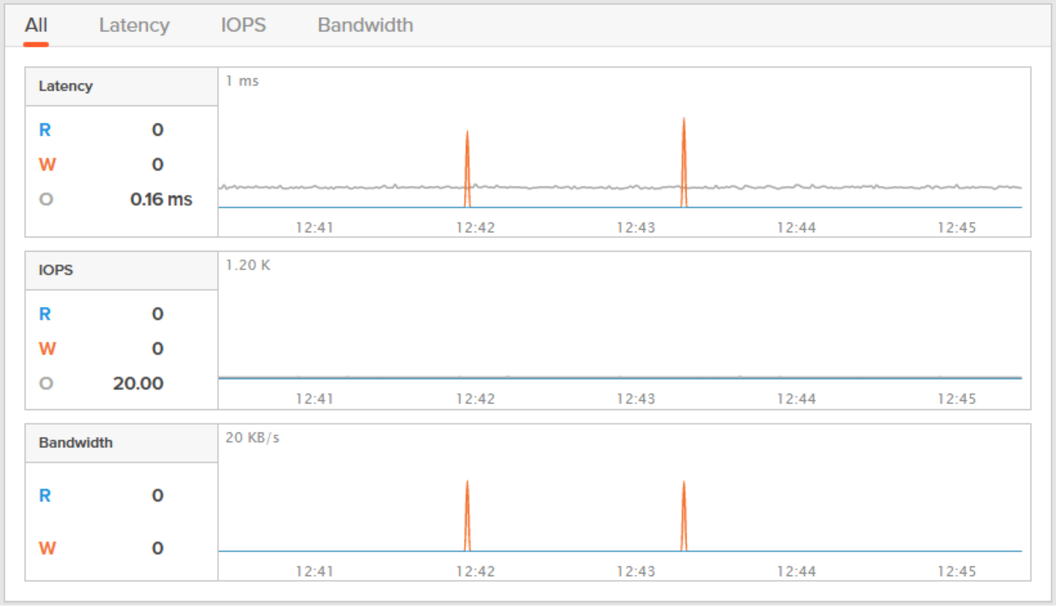


Figure 32. Purity//FB GUI Dashboard - Performance Charts

The performance metrics are displayed along a scrolling graph; incoming data appears along the right side of each graph every second as older data drops off the left side after 5 minutes. Each performance chart includes R, W, and O (if applicable) values, representing the most recent data samples. Hover over any of the charts to display metrics for a specific point in time. The values that appear in the point-in-time tooltips are rounded to two decimal places. The performance panel includes Latency, IOPS, and Bandwidth charts. The All chart displays all three performance charts in one view as shown in Figure 32.

Monitoring Latency

The Latency chart displays the average latency times for various operations.

- Read Latency (R): Average arrival-to-completion time, measured in milliseconds, for a read operation.
- Write Latency (W): Average arrival-to-completion time, measured in milliseconds, for a write operation.
- Other Latency (O): Average arrival-to-completion time, measured in milliseconds, for all other metadata operations.

IOPS

The IOPS (Input/output Operations Per Second) chart displays I/O requests processed per second by the array. This metric counts requests per second, regardless of how much or how little data is transferred in each.

- Read IOPS (R): Number of read requests processed per second.
- Write IOPS (W): Number of write requests processed per second.
- Other IOPS (O): Number of metadata operations processed per second.
- Average IO Size: Average I/O size per request processed. Requests include reads and writes.

Bandwidth

The Bandwidth chart displays the number of bytes transferred per second to and from the array (both file systems and buckets). The data is counted in its expanded form rather than the reduced form stored in the array to truly reflect what is transferred over the storage network. Metadata bandwidth is not included in these numbers.

- Read Bandwidth (R): Number of bytes read per second.
- Write Bandwidth (W): Number of bytes written per second.

By default, the performance charts display performance metrics for the past 5 minutes. To display more than 5 minutes of historical data, select `Analysis > Performance.`

Note about the Performance Charts

The Dashboard and Analysis pages display the same latency, IOPS, and bandwidth performance charts, but the information is presented differently between the two pages.

On the Dashboard page:

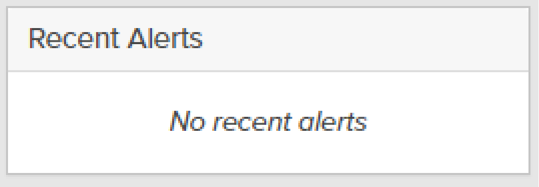
- The performance charts are updated once every second.
- The performance charts display up to 5 minutes of historical data.

On the Analysis page:

- At its shortest range (5m), the performance charts are updated once every second. As the range increases, the update frequency (and resolution) decreases.
- The performance charts display up to 1 year of historical data.
- The performance charts can be filtered to display metrics by protocol.

Monitoring Alerts

The Recent Alerts panel displays a list of alerts that Purity//FB saw that is both flagged and not in the closed state. The list contains recent alerts of all severity levels. If no alerts are logged, the panel displays No recent alerts (Figure 33).



Recent Alerts

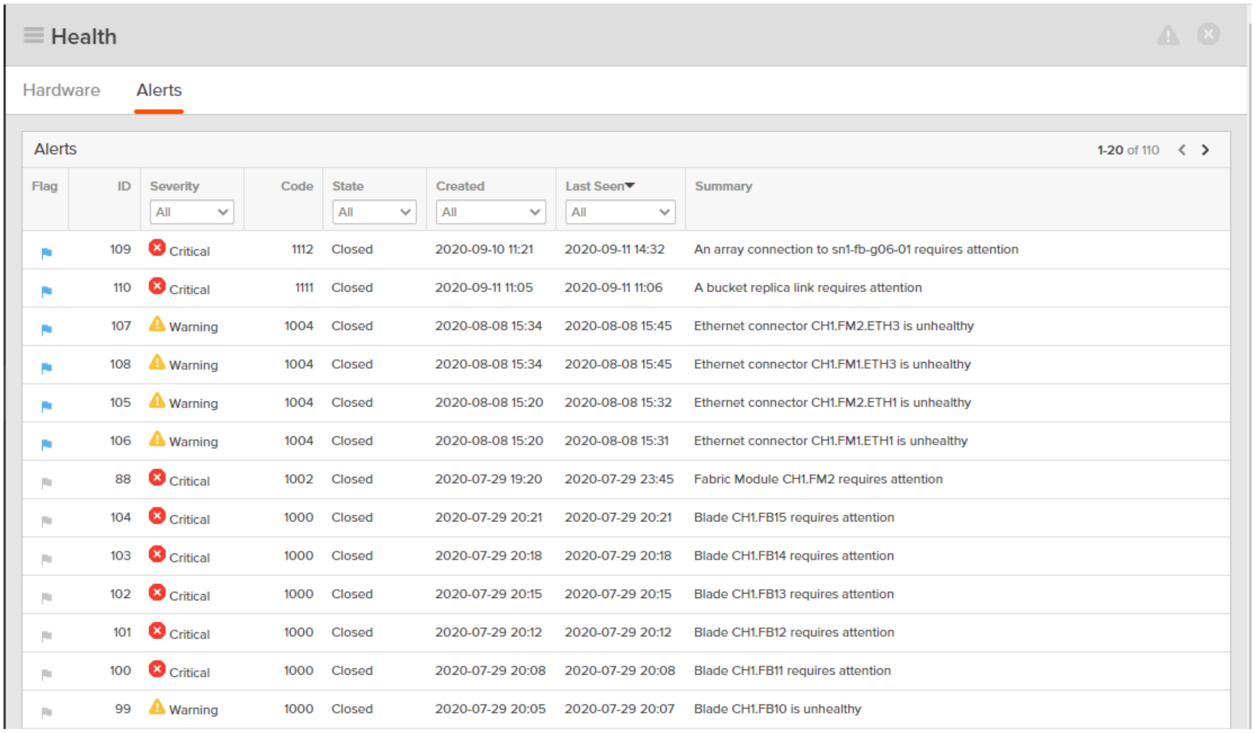
No recent alerts

Figure 33. Purity//FB GUI Dashboard - Recent Alerts Panel

To view the details of an alert, click the alert message.

To remove an alert from the Recent Alerts panel, click the clear flag (X) button. The alert will no longer be displayed in the Recent Alerts panel but will still appear on the Health page.

To view a list of all alerts, including ones that are no longer open, go to the Health page (Figure 34).



Alerts 1-20 of 110

Flag	ID	Severity All	Code	State All	Created All	Last Seen▾ All	Summary
⚑	109	✖ Critical	1112	Closed	2020-09-10 11:21	2020-09-11 14:32	An array connection to sn1-fb-g06-01 requires attention
⚑	110	✖ Critical	1111	Closed	2020-09-11 11:05	2020-09-11 11:06	A bucket replica link requires attention
⚑	107	⚠ Warning	1004	Closed	2020-08-08 15:34	2020-08-08 15:45	Ethernet connector CH1.FM2.ETH3 is unhealthy
⚑	108	⚠ Warning	1004	Closed	2020-08-08 15:34	2020-08-08 15:45	Ethernet connector CH1.FM1.ETH3 is unhealthy
⚑	105	⚠ Warning	1004	Closed	2020-08-08 15:20	2020-08-08 15:32	Ethernet connector CH1.FM2.ETH1 is unhealthy
⚑	106	⚠ Warning	1004	Closed	2020-08-08 15:20	2020-08-08 15:31	Ethernet connector CH1.FM1.ETH1 is unhealthy
⚑	88	✖ Critical	1002	Closed	2020-07-29 19:20	2020-07-29 23:45	Fabric Module CH1.FM2 requires attention
⚑	104	✖ Critical	1000	Closed	2020-07-29 20:21	2020-07-29 20:21	Blade CH1.FB15 requires attention
⚑	103	✖ Critical	1000	Closed	2020-07-29 20:18	2020-07-29 20:18	Blade CH1.FB14 requires attention
⚑	102	✖ Critical	1000	Closed	2020-07-29 20:15	2020-07-29 20:15	Blade CH1.FB13 requires attention
⚑	101	✖ Critical	1000	Closed	2020-07-29 20:12	2020-07-29 20:12	Blade CH1.FB12 requires attention
⚑	100	✖ Critical	1000	Closed	2020-07-29 20:08	2020-07-29 20:08	Blade CH1.FB11 requires attention
⚑	99	⚠ Warning	1000	Closed	2020-07-29 20:05	2020-07-29 20:07	Blade CH1.FB10 is unhealthy

Figure 34. Purity//FB GUI Health page - Alerts

Monitoring Hardware Health

The Hardware Health panel displays the operational state of the FlashBlade array chassis, blades, and fabric modules. Hover over the image to view the component details. Depending on the configuration of your array, the Hardware Health panel displays either a single or multi-chassis FlashBlade graphic. The single-chassis Hardware Health panel is shown in Figure 35.

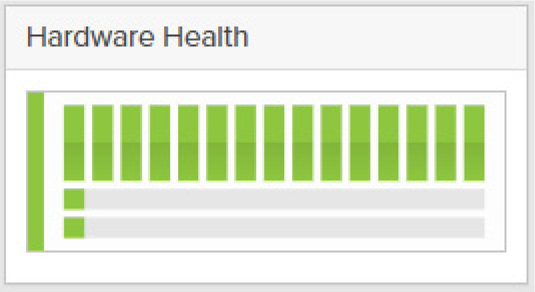


Figure 35. Purity//FB GUI Dashboard - Hardware Health Panel - Single chassis

To analyze the hardware components in more detail, click the Health link, the display is shown below in Figure 36.

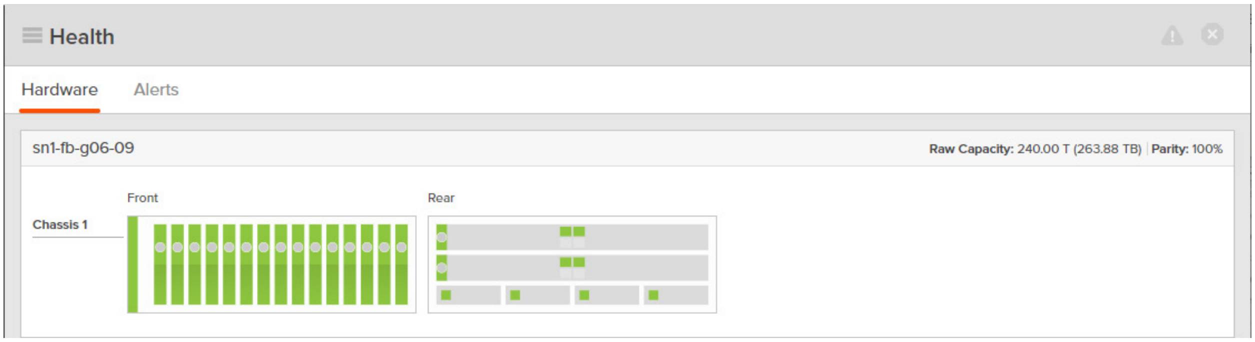


Figure 36. Purity//FB GUI Dashboard Health - Hardware

Conclusion

The Vertica Analytics platform is a powerful tool for high performance, large-scale SQL analytics. Vertica in Eon Mode is an on-premise storage deployment model for Vertica databases that leverages shared fast-object storage. Vertica in Eon Mode significantly improves operational efficiency while maintaining database performance. FlashBlade object storage allows organizations to take advantage of existing TCP/IP networking infrastructure reducing cost and simplifying the actual install. This jointly engineered data analytics solution provides workload isolation and throughput improvement thereby delivering better economics and greater choice for customers by expanding the ability to dynamically manage workloads and simplify operations. The Vertica using Eon Mode on FlashBlade solution delivers speed, cloud-like simplicity, and flexibility in a complete, on-premises, cost-optimized data analytics solution.

Product Support

Pure and Vertica will support their customers following each respective company's normal support process. If Pure needs to engage Vertica, the Pure support team will follow the engagement process in the Micro Focus Customer Support Agreement (CSA). Vertica is a Micro Focus company. Pure offers support services over the phone, by email, and through our web portal.

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 Numbers (International)	support.purestorage.com/pure1/support

Additional Documentation

- Vertica Documentation
- Vertica Linux OS Deployment
- Installing Vertica in Eon Mode on-premises
- Solution Brief: Pure Storage and Vertica
- Pure Storage and Vertica Partnership
- Blog Post: "Why Modern Analytics Need a Modern Infrastructure"
- Upgrading Vertica
- Managing Subclusters in Management Console
- Vertica in Eon Mode Shards and Subscriptions
- Deploying Vertica in Eon Mode on Pure FlashBlade

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

Appendix A: Vertica Predeployment Validation Utilities

Vertica provides several validation utilities that can be used before deploying Vertica to help determine if your hosts and network can properly handle the processing and network traffic required by Vertica. These utilities can also be used if you are encountering performance issues and need to troubleshoot the issue.

After installation, Vertica provides the following scripts in `/opt/vertica/bin`:

- vcpuperf- a CPU performance test used to verify your CPU performance.
- vioperf- an Input/Output test used to verify the speed and consistency of your hard drives.
- vnetperf- a Network test used to test the latency and throughput of your network between hosts.

These utilities can be run at any time but are recommended to validate the environment before running the `install_vertica` script.

CPU Test Utility

The `vcpuperf` utility measures your server's CPU processing speed and compares it against benchmarks for common server CPUs. The utility performs a CPU test and measures the time it takes to complete the test. The lower the number scored on the test, the better the performance of the CPU.

The `vcpuperf` utility also checks the high and low load times to determine if CPU throttling is enabled. If a server's low-load computation time is significantly longer than the high-load computation time, CPU throttling may be enabled. CPU throttling is a power-saving feature. However, CPU throttling can reduce the performance of your server. Vertica recommends disabling CPU throttling to enhance server performance.

Validate CPU resources with `vcpuperf` utility:

```
[root@sn1-r720-g03-27 bin]# vcpuperf

Compiled with: 7.3.1 20180303 (Red Hat 7.3.1-5)
Expected time on Core 2, 2.53GHz: ~9.5s
Expected time on Nehalem, 2.67GHz: ~9.0s
Expected time on Xeon 5670, 2.93GHz: ~8.0s
This machine's time:
  CPU Time: 8.080000s
  Real Time:8.090000s
Some machines automatically throttle the CPU to save power.
This test can be done in <100 microseconds (60-70 on Xeon 5670, 2.93GHz).
Low load times much larger than 100-200us or much larger than the corresponding high load time
indicate low-load throttling, which can adversely affect small query / concurrent performance.
This machine's high load time: 65 microseconds.
This machine's low load time: 75 microseconds
```

Network Test Utility

The `vnetperf` utility allows administrators to measure the network performance of production hosts. It can measure network latency and the throughput for both the TCP and UDP protocols.

> **Note:** This utility introduces a high network load and must not be used on a running Vertica cluster or database performance is degraded.

This utility can detect:

- If throughput is low for all hosts or a particular host
- If latency is high for all hosts or a particular host
- If there are any bottlenecks between one or more hosts or subnets
- Any limit in the number of TCP connections that can be established simultaneously
- A high rate of packet loss on the network.

The latency test measures the latency from the host running the script to the other hosts. Any host that has a particularly high latency should be investigated further.

The throughput tests measure both UDP and TCP throughput. You can specify a rate limit in MB/s to use for these tests or allow the utility to use a range of throughputs to be used.

Validate Network resources with `vnetperf` utility:

```
[root@sn1-r720-g03-27 bin]# vnetperf --hosts 10.21.236.55, 10.21.236.56 latency
The maximum recommended rtt latency is 2 milliseconds. The ideal rtt latency is 200 microseconds or
less. It is recommended that clock skew be kept to under 1 second.
test         | date                    | node             | index | rtt latency (us)  | clock skew (us)
-------------------------------------------------------------------------------------------------------
latency      | 2020-10-16_11:50:38,425 | 10.21.236.55      | 0     | 55                 | 1
latency      | 2020-10-16_11:50:38,425 | 10.21.236.56      | 1     | 258                | -1999
The minimum recommended throughput is 100 MB/s. Ideal throughput is 800 MB/s or more. Note: UDP numbers
may be lower, multiple network switches may reduce performance results.

date                    | test            | limit (MB/s) | node          | MB/s (sent) | MB/s (rec) |
bytes (sent)   | bytes (rec)    | duration (s)
-------------------------------------------------------------------------------------------------------
-----------------------------------------
2020-10-16_11:50:38,439 | udp-throughput | 32           | 10.21.236.55  | 30.5148     | 30.5162    |
160001984      | 160009344      | 5.00052
2020-10-26_11:50:38,439 | udp-throughput | 32           | 10.21.236.56  | 30.5157     | 30.5171    |
160001984      | 160009344      | 5.00037
2020-10-16_11:50:38,439 | udp-throughput | 32           | average       | 30.5163     | 30.5163    |
160001984      | 160001984      | 5.00028
```

```
2020-10-16_11:52:11,562 | tcp-throughput | 2048         | 10.21.236.55  | 1130.26      | 1028.32     |
5931466752     | 5396529152     | 5.00477
2020-10-16_11:52:11,562 | tcp-throughput | 2048         | 10.21.236.56  | 1182.7       | 1046.35     |
6227263488     | 5509316608     | 5.02137
```

Input/Output Test Utility

The `vioperf` utility quickly tests the performance of your host's input and output subsystem. The utility performs the following tests: sequential write, sequential rewrite, sequential read, and skip read (read non-contiguous data blocks). The utility verifies that the host reads the same bytes that it wrote and prints its output to STDOUT. The utility also logs the output to a JSON formatted file.

Validate Storage resources with `vioperf` utility:

```
[root@sn1-r720-g03-27 /]# vioperf /mnt/vertica_shared
WARNING: Vertica suggests allowing 1 open file per MB of memory, minimum of 65536; see 'ulimit -n'.

The minimum required I/O is 20 MB/s read and write per physical processor core on each node, in full
duplex i.e. reading and writing at this rate simultaneously, concurrently on all nodes of the cluster.
The recommended I/O is 40 MB/s per physical core on each node. For example, the I/O rate for a server
node with 2 hyper-threaded six-core CPUs is 240 MB/s required minimum, 480 MB/s recommended. Using
direct io (buffer size=1048576, alignment=512) for directory "/mnt/vertica_shared"
test    | directory             | counter name         | counter value | counter value (10 sec avg) |
counter value/core | counter value/core (10 sec avg) | thread count | %CPU | %IO Wait | elapsed time
(s)| remaining time (s)
----------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
Write    | /mnt/vertica_shared | MB/s                 | 959           | 959                        |
39.9583           | 39.9583                         | 24              | 6    | 0         | 10
| 65
Write    | /mnt/vertica_shared | MB/s                 | 959            | 960                       |
39.9583           | 40                              | 24              | 6    | 0         | 20
| 55
Write    | /mnt/vertica_shared | MB/s                 | 960            | 959                       | 40
| 39.9583                       | 24              | 6    | 0         | 30                | 45
Write    | /mnt/vertica_shared | MB/s                 | 960            | 958                       | 40
| 39.9167                       | 24              | 6    | 0         | 40                | 35
Write    | /mnt/vertica_shared | MB/s                 | 960            | 960                       | 40
| 40                            | 24              | 6    | 0         | 50                | 25
Write    | /mnt/vertica_shared | MB/s                 | 960            | 960                       | 40
| 40                            | 24              | 6    | 0         | 60                | 15
```

```
Write    | /mnt/vertica_shared | MB/s                | 960             | 959                        |
40                | 39.9583                         | 24          | 6  | 0        | 70
| 5
Write    | /mnt/vertica_shared | MB/s                | 960             | 956                        |
40                | 39.8333                         | 24          | 6  | 0        | 75
| 0
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 585+585       | 585+585                    |
24.375+24.375     | 24.375+24.375                   | 24          | 4  | 0        | 10
| 65
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 593+593       | 601+601                    | 2
4.7083+24.7083  | 25.0417+25.0417               | 24          | 4    | 0        | 20              | 55

ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 593+593       | 594+594                    | 2
4.7083+24.7083  | 24.75+24.75                   | 24         | 4     | 0        | 30              | 45
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 592+592       | 593+593                    |
24.6667+24.6667     | 24.7083+24.7083                 | 24          | 4  | 0        | 40
| 35
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 589+589       | 589+589                    |
24.5417+24.5417     | 24.5417+24.5417                 | 24          | 4  | 0        | 50
| 25
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 592+592       | 594+594                    |
24.6667+24.6667     | 24.75+24.75                     | 24          | 4  | 0        | 60
| 15
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 593+593       | 593+593                    |
24.7083+24.7083     | 24.7083+24.7083                 | 24          | 5  | 0        | 70
| 5
ReWrite   | /mnt/vertica_shared | (MB-read+MB-write)/s| 592+592       | 602+602                    |
24.6667+24.6667     | 25.0833+25.0833                 | 24          | 4  | 0        | 75
| 0
Read      | /mnt/vertica_shared | MB/s                | 1037          | 1037                       |
43.2083             | 43.2083                         | 24          | 7  | 0        | 10
| 65
Read      | /mnt/vertica_shared | MB/s                | 1037          | 1051                       |
43.2083             | 43.7917                         | 24          | 7  | 0        | 20
| 55
Read      | /mnt/vertica_shared | MB/s                | 1032          | 1032                       |
43                  | 43                              | 24          | 7  | 0        | 30
| 45
Read      | /mnt/vertica_shared | MB/s                | 1032          | 1032                       |
43                  | 43                              | 24          | 7  | 0        | 40
| 35
```

```
Read      | /mnt/vertica_shared | MB/s                | 1032          | 1032                        |
43                  | 43                             | 24         | 7   | 0         | 50
| 25
Read      | /mnt/vertica_shared | MB/s                | 1032          | 1032                        |
43                  | 43                             | 24         | 7   | 0         | 60
| 15
Read      | /mnt/vertica_shared | MB/s                | 1032          | 1032                        |
43                  | 43                             | 24         | 7   | 0         | 70
| 5
Read      | /mnt/vertica_shared | MB/s                | 1032          | 1056                        |
43                  | 44                             | 24         | 7   | 0         | 75
| 0
SkipRead  | /mnt/vertica_shared | seeks/s             | 41469         | 41469                       |
1727.88             | 1727.88                        | 24         | 1   | 0         | 10
| 65
SkipRead  | /mnt/vertica_shared | seeks/s             | 42149         | 42826                       |
1756.21             | 1784.42                        | 24         | 1   | 0         | 20
| 55
SkipRead  | /mnt/vertica_shared | seeks/s             | 42186         | 42264                       |
1757.75             | 1761                           | 24         | 1   | 0         | 30
| 45
SkipRead  | /mnt/vertica_shared | seeks/s             | 42282         | 42577                       |
1761.75             | 1774.04                        | 24         | 1   | 0         | 40
| 35
SkipRead  | /mnt/vertica_shared | seeks/s             | 42412         | 42923                       |
1767.17             | 1788.46                        | 24         | 2   | 0         | 50
| 25
SkipRead  | /mnt/vertica_shared | seeks/s             | 42472         | 42778                       |
1769.67             | 1782.42                        | 24         | 2   | 0         | 60
| 15
SkipRead  | /mnt/vertica_shared | seeks/s             | 42417         | 42084                       |
1767.38             | 1753.5                         | 24         | 2   | 0         | 70
| 5
SkipRead  | /mnt/vertica_shared | seeks/s             | 42438         | 42723                       |
1768.25             | 1780.12                        | 24         | 1   | 0         | 75
| 0
[root@sn1-r720-g03-27 /]#
```

Document Revisions

Rev #	Description	Date
1.0	Initial Publication	Novemer 2020

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