Cloudera Runtime 7.3.1

Virtual Private Clusters and Cloudera SDX

Date published: 2024-07-31
Date modified: 2024-07-31

https://docs.cloudera.com/

Legal Notice

Contents

Introduction to Virtual Private Clusters and Cloudera SDX

A Virtual Private Cluster uses the Cloudera Shared Data Experience (SDX) to simplify deployment of both on-premise and cloud-based applications and enable workloads running in different clusters to securely and flexibly share data.

The architecture of a Virtual Private Cluster provides many advantages for deploying workloads and sharing data among applications, including a shared catalog, unified security, consistent governance, and data lifecycle management.

In traditional cluster deployments, a Regular cluster contains storage nodes, compute nodes, and other services such as metadata services and security services that are collocated in a single cluster. This traditional architecture provides many advantages where computational services such as Impala and YARN can access collocated data sources such as HDFS or Hive.

With Virtual Private Clusters and the SDX framework, a new type of cluster is available in Cloudera Manager called a Compute cluster. A Compute cluster runs computational services such as Hive Execution Service, Spark, or YARN but you configure these services to access data hosted in another Regular cluster, called the Base cluster. Using this architecture, you can separate compute and storage resources in a variety of ways to flexibly maximize resources.

Advantages of Separating Compute and Data Resources

Separating compute and data resources in a Virtual Private Cluster has important advantages for many workloads.

Architectures that separate compute resources from data resources can provide many advantages for a CDP deployment:

- More options for deploying computational and storage resources
  - You can selectively deploy resources using on-premise servers, containers, virtual machines, or cloud resources that are tailored for the workload. When you configure a Compute cluster, you can provision hardware that is more appropriate for computational workloads while the Base cluster can use hardware that emphasizes storage capacity. Cloudera recommends that each cluster use similar hardware.
  - Software resources can be optimized to best use computational and storage resources.
- Ephemeral clusters

  When deploying clusters on cloud infrastructure, having separate clusters for compute and storage allows you to temporarily shut down the compute clusters and avoid unnecessary expense -- while still leaving the data available to other applications.
- Workload Isolation
  - Compute clusters can help to resolve resource conflicts among users accessing the cluster. Longer running or resource intensive workloads can be isolated to run in dedicated compute clusters that do not interfere with other workloads.
  - Resources can be grouped into clusters that allow IT to allocate costs to the teams that use the resources.

Related Information

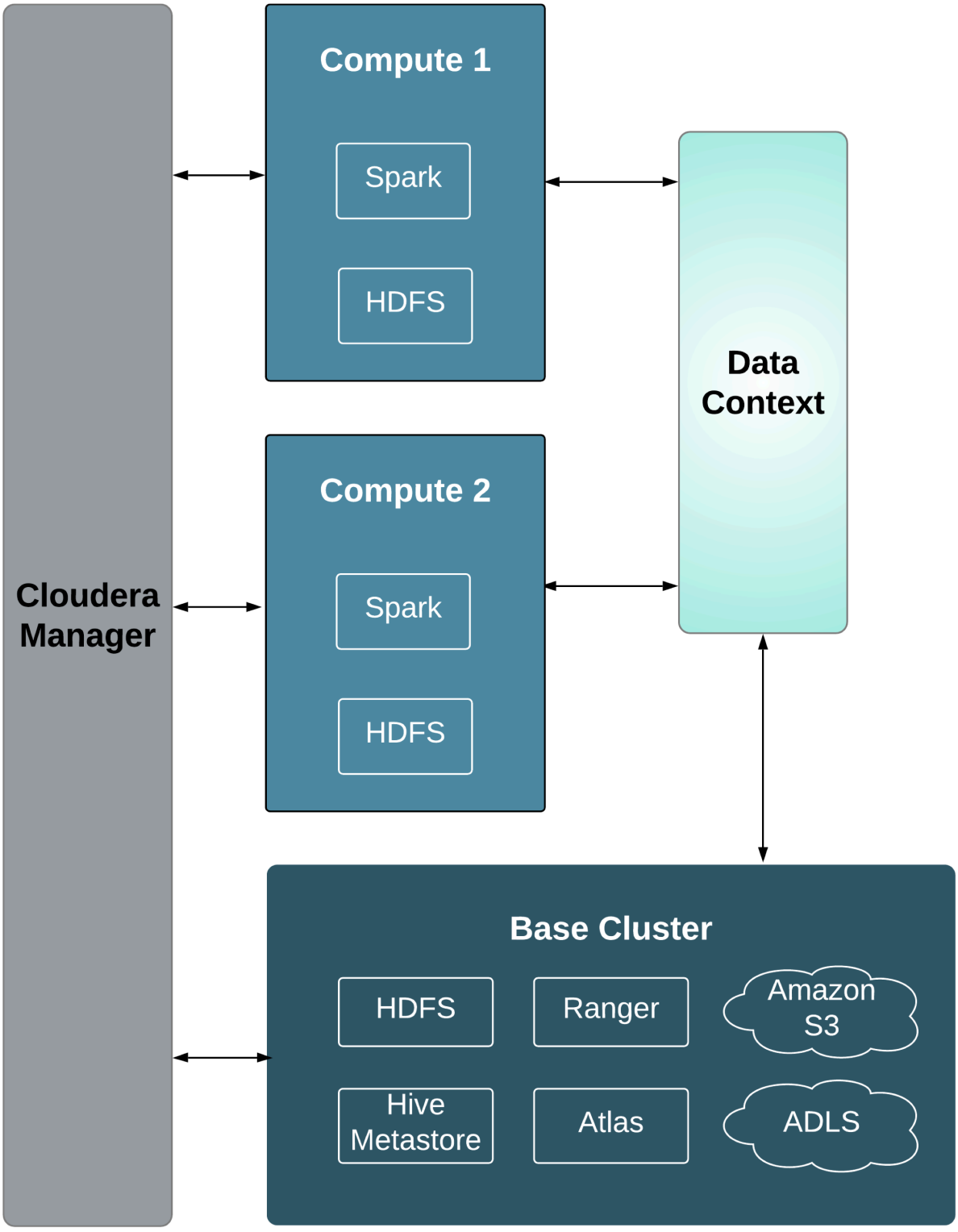
Compatibility Considerations for Virtual Private Clusters

Architecture

If you are creating Virtual Private Clusters, it is important to understand the architecture of compute clusters and how they related to Data contexts.

A Compute cluster is configured with compute resources such as Spark or Hive Execution. Workloads running on these clusters access data by connecting to a Data Context for the Base cluster. A Data Context is a connector to

a Regular cluster that is designated as the Base cluster. The Data context defines the data, metadata and security services deployed in the Base cluster that are required to access the data. Both the Compute cluster and Base cluster are managed by the same instance of Cloudera Manager. A Base cluster can contain any Cloudera Runtime services -- but only HDFS, Hive, Atlas, Ranger, Amazon S3, and Microsoft ADLS can be shared using the data context.



Either the Core Configuration service or the HDFS service is required on Compute clusters as temporary, persistent space. If you are using Hive-on-Tez in the Compute Cluster, the HDFS service is required. A compute cluster uses these services to store temporary files used in multi-stage MapReduce jobs. In addition, the following services may deployed as needed:

- Hive Execution Service (This service supplies the HiveServer2 role only.)
- Hue
- Kafka
- Spark 2
- Oozie (only when Hue is available, and is a requirement for Hue)
- YARN
- HDFS
- Stub DFS (Stub DFS replaces Core Settings and requires the Storage Operations role.)

The functionality of a Virtual Private cluster is a subset of the functionality available in Regular clusters, and the versions of Cloudera Runtime that you can use are limited. For more information, see Compatibility Considerations for Virtual Private Clusters on page 6.

Related Information

Networking Considerations for Virtual Private Clusters

Performance Trade Offs

Learning about the types of workloads appropriate for Virtual Private Clusters can help you decide if this architecture is appropriate for your needs.

Throughput

Because data will be accessed over network connections to other clusters, this architecture may not be appropriate for workloads that scan large amounts of data. These types of workloads may run better on Regular clusters where compute and storage are collocated and features such as Impala short-circuit reads can provide improved performance.

You can evaluate the networking performance using the Network Performance Inspector.

Ephemeral Clusters

For deployments where the Compute clusters are shut down or suspended when they are not needed, cluster services that collect historical data do not collect data when the Compute clusters are off-line, and the history is not available to users. This affects services such as the Spark History Server and the YARN JobHistory Server. When the Compute cluster restarts, the previous history will be available.

Compatibility Considerations for Virtual Private Clusters

Related Information

Networking Considerations for Virtual Private Clusters

CDH and Cloudera Runtime Version Compatibility

The Cloudera Runtime version of the Compute cluster must match the major.minor version of the Base cluster. Support for additional combinations of Compute and Base cluster versions may be added at a later time. The following Cloudera Runtime and CDH versions are supported for creating Virtual Private Clusters:

- CDH 5.15
- CDH 5.16
- CDH 6.0

- CDH 6.1
- CDH 6.2
- CDH 6.3
- Cloudera Runtime 7.0.3
- Cloudera Runtime 7.1.1 and higher

Licensing Requirements

A valid CDP Private Cloud Base Edition license is required to use Virtual Private Clusters. You cannot access this feature with a CDP Private Cloud Base Edition Trial license.

Components

- SOLR – Not supported on Compute clusters
- Kudu – Not supported on Compute clusters.
- HDFS
  - Either the Core Configuration service or the HDFS service is required on Compute clusters as temporary, persistent space. If you are using Hive-on-Tez in the Compute Cluster, the HDFS service is required. The intent is to use this for Hive temporary queries and is also recommended for multi-stage Spark ETL jobs.
  - Cloudera recommends a minimum storage of 1TB per host, configured as the HDFS DataNode storage directory.
  - The Base cluster must have an HDFS service.
  - Isilon is not supported on the Base cluster.
  - S3 and ADLS connectors are supported only on the Base cluster; Compute clusters will use the supplied S3 or ADLS credentials from their associated Base cluster
  - The HDFS service on the Base cluster must be configured for high availability.
  - Cloudera highly recommends enabling high availability for the HDFS service on the Compute cluster, but it is not required.
  - The Base and compute nameservice names must be distinct.
  - The following configurations for the local HDFS service on a compute cluster must match the configurations on the Base cluster. This enables services on the Compute cluster to correctly access services on the Base cluster:
    - Hadoop RPC protection
    - Data Transfer protection
    - Enable Data Transfer Encryption
    - Kerberos Configurations
    - TLS/SSL Configuration (only when the cluster is not using Auto-TLS). See Security on page 8.
  - Do not override the nameservice configuration in the Compute cluster by using Advanced Configuration Snippets.
  - The HDFS path to compute cluster home directories use the following structure: /mc/<*cluster_id*>/fs/user

    You can find the <*cluster_id*> by clicking the cluster name in the Cloudera Manager Admin Console. The URL displayed by your browser includes the cluster id. For example, in the URL below, the cluster ID is 1.

    ```
    http://myco-1.prod.com:7180/cmf/clusters/1/status
    ```
- Backup and Disaster Recovery (BDR) – not supported when the source cluster is a Compute cluster and the target cluster is running a version of Cloudera Manager lower than 6.2.
- YARN and MapReduce
  - If both MapReduce (MR1, Deprecated in CM6) and YARN (MR2) are available on the Base cluster, dependent services in the Compute cluster such as Hive Execution Service will use MR1 because of the way service dependencies are handled in Cloudera Manager. To use YARN, you can update the configuration to make these Compute cluster services depend on YARN (MR2) before using YARN in your applications.

- Impala – Not supported on Compute clusters
- Hue
  - Only one Hue service instance is supported on a Compute cluster.
  - The Hue service on Compute clusters will not share user-specific query history with the Hue service on other Compute clusters or Hue services on the Base cluster
  - Hue examples may not install correctly due to differing permissions for creating tables and inserting data. You can work around this problem by deleting the sample tables and then re-adding them.
  - If you add Hue to a Compute cluster after the cluster has already been created, you will need to manually configure any dependencies on other services (such as Hive or Hive Execution Service) in the Compute cluster.
- Hive Execution Service

  The newly introduced “Hive execution service” is only supported on Compute clusters, and is not supported on Base or Regular clusters.

  To enable Hue to run Hive queries on a Compute cluster, you must install the Hive Execution Service on the Compute cluster.

Compute Cluster Services

Only the following services can be installed on Compute clusters:

- Hive Execution Service (This service supplies the HiveServer2 role only.)
- Hue
- Kafka
- Spark 2
- Oozie (only when Hue is available, and is a requirement for Hue)
- YARN
- HDFS
- Stub DFS (Stub DFS replaces Core Settings and requires the Storage Operations role.)

Cloudera Navigator Support

Navigator lineage and metadata, and Navigator KMS are not supported on Compute clusters.

Cloudera Manager Permissions

Cluster administrators who are authorized to only see Base or Compute clusters can only see and administer these clusters, but cannot create, delete, or manage Data Contexts. Data context creation and deletion is only allowed with the Full Administrator use role or for unrestricted Cluster Administrators.

Security

- KMS
  - Base cluster:
    - Hadoop KMS is not supported.
    - KeyTrustee KMS is supported on Base cluster.
  - Compute cluster: Any type of KMS is not supported.
- Authentication/User directory
  - Users should be identically configured on hosts on compute and Base clusters, as if the nodes are part of the same cluster. This includes Linux local users, LDAP, Active Directory, or other third-party user directory integrations.

- Kerberos
  - If a Base cluster has Kerberos installed, then the Compute and Base clusters must both use Kerberos in the same Kerberos realm. The Cloudera Manager Admin Console helps facilitate this configuration in the cluster creation process to ensure compatible Kerberos configuration.
- TLS
  - If the Base cluster has TLS configured for cluster services, Compute cluster services must also have TLS configured for services that access those Base cluster services.
  - Cloudera strongly recommends enabling Auto-TLS to ensure that services on Base and Compute clusters uniformly use TLS for communication.
  - If you have configured TLS, but are not using Auto-TLS, note the following:
    - You must create an identical configuration on the Compute cluster hosts before you add them to a Compute cluster using Cloudera Manager. Copy all files located in the directories specified by the following configuration properties, from the Base clusters to the Compute cluster hosts:
      - hadoop.security.group.mapping.ldap.ssl.keystore
      - ssl.server.keystore.location
      - ssl.client.truststore.location
    - Cloudera Manager will copy the following configurations to the compute cluster when you create the Compute cluster.
      - hadoop.security.group.mapping.ldap.use.ssl
      - hadoop.security.group.mapping.ldap.ssl.keystore
      - hadoop.security.group.mapping.ldap.ssl.keystore.password
      - hadoop.ssl.enabled
      - ssl.server.keystore.location
      - ssl.server.keystore.password
      - ssl.server.keystore.keypassword
      - ssl.client.truststore.location
      - ssl.client.truststore.password

Networking

Workloads running on Compute clusters will communicate heavily with hosts on the Base cluster; Customers should have network monitoring in place for networking hardware (such as switches including top-of-rack, spine/leaf routers and so on) to track and adjust bandwidth between racks hosting hosts utilized in Compute and Base clusters.

You can also use the Cloudera Manager Network Performance Inspector to evaluate your network.

For more information on how to set up networking, see Networking Considerations for Virtual Private Clusters on page 9.

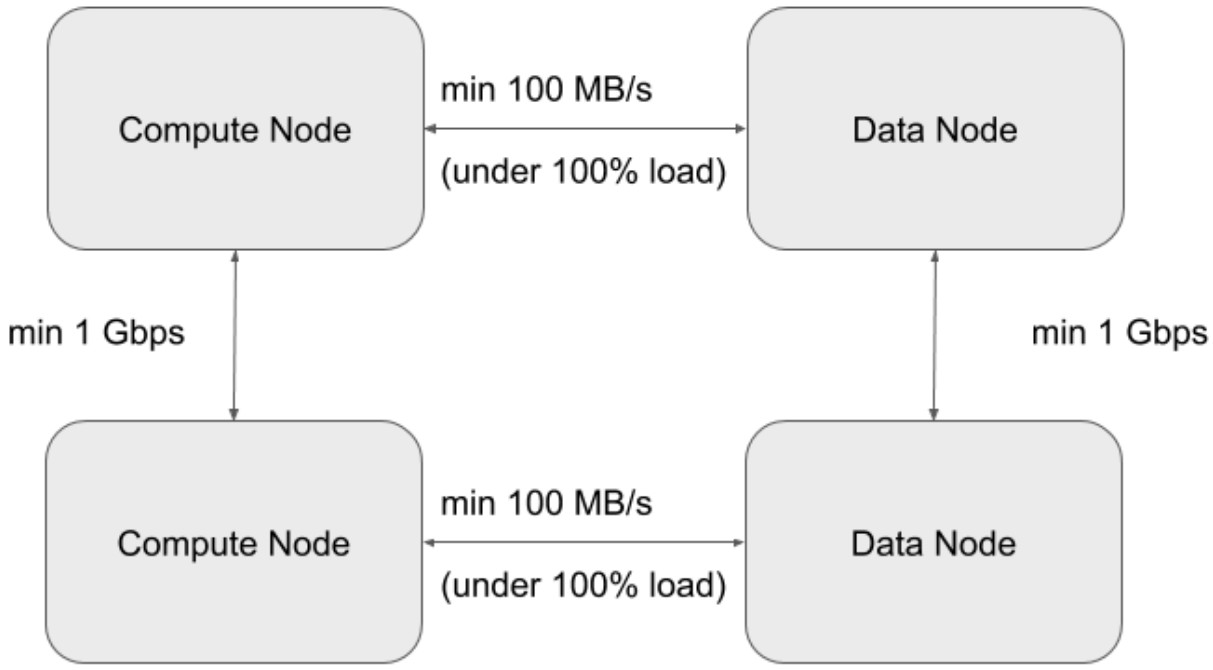
Networking Considerations for Virtual Private Clusters

Minimum Network Performance Requirements

A Virtual Private Cluster deployment has the following requirements for network performance:

- A worst case IO throughput of 100MB/s , i.e., 1 Gbp/s of network throughput (sustained) between any compute node and any storage node. To achieve the worst case, we will measure throughput when all computenodes are reading/writing from the storage nodes at the same time. This type of parallel execution is typical of big data applications.
- A worst case network bandwidth of 1Gbps between any two workload cluster Nodes or any two base cluster Nodes.

The following picture summarizes these requirements:



Sizing and designing the Network Topology

Minimum and Recommended Performance Characteristics

This section can be used to derive network throughput requirements, following the minimum network throughputs outlined in the table below.

Note: For the purposes of this guide, we are going to use the following terms:

- North-South (NS) traffic patterns indicate network traffic between Compute and Storage tiers.
- East-West (EW) traffic patterns indicate internal network traffic within the storage or compute clusters.

Tier	Minimum Per-Node Storage IO throughput (MB/s)	Minimum per-Node network throughput (Gbps) (EW + NS)	Recommended per-Node Storage IO throughput (MB/s)	Recommended per-Node network throughput (Gbps) (EW + NS)
Compute (This can be virtualized or bare-metal)	100	2	200	4
Storage (This can be virtualized or bare-metal)	400	8	800	16

Attention: The storage backend will predicate the maximum number of compute Nodes connecting to it. The backend is not just about capacity, but also throughput. Since the backend will be HDFS DataNodes, proper sizing of the backend nodes (or Nodes) is required. The following guidelines will help size the backend accordingly:

- Assuming SATA drives are being used for storage, each drive can generate about 100MB/s of throughput in bare-metal clusters and should expect to generate 70-80MB/s in virtualized clusters with directly attached storage.
- Each drive requires one physical CPU core. So a node with 12 drives should have at least 12 cores.
- The network bandwidth should be planned with 2x NS traffic in mind. For instance, if a node has 12 drives, then NS traffic expected should be 1200MB/s (1.2GB/s), which is ~ 10 Gbps in network throughput. Plan for 20 Gbps to address EW traffic.
- The inter-networking between the compute layer and the storage layer needs to factor in how much throughput will flow in the NS direction. The section below articulates the considerations for network design and illustrates the impact of having various degrees of oversubscription on the overall achievable throughput of the clusters.

Network Topology Considerations

The preferred network topology is spine-leaf with as close to 1:1 over subscription between leaf and spine switches, ideally aiming for no over subscription. This is so we can ensure full line-rate between any combination of storage and compute nodes. Since this architecture calls for disaggregation of compute and storage, the network design must be more aggressive in order to ensure best performance.

There are two aspects to the minimum network throughput required, which will also determine the compute to storage nodes ratio.

- The network throughput and IO throughput capabilities of the storage backend.
- The network throughput and network over subscription between compute and storage tiers (NS).

Stepping through an example scenario can help to better understand this. Assuming that this is a greenfield setup, with compute nodes and storage nodes both being virtual machines (VMs):

- Factor that the total network bandwidth is shared between EW and NS traffic patterns. So for 1 Gbps of NS traffic, we should plan for 1 Gbps of EW traffic as well.
- Network oversubscription between compute and storage tiers is 1:1
- Backend comprises of 5 nodes (VMs), each with 8 SATA drives.
    - This implies that the ideal IO throughput (for planning) of the entire backend would be ~ 4GB/s, which is ~ 800MB/s per Node. which is the equivalent of ~ 32 Gbps network throughput for the cluster, and ~ 7 Gbps per node for NS traffic pattern
    - Factoring in EW + NS we would need 14 Gbps per Node to handle the 800MB/s IO throughput per Node.
- Compute Cluster would then ideally have the following:
    - 5 hypervisor nodes each with 7 Gbps NS + 7 gpbs EW = 14 Gbps total network throughput per hypervisor.
    - This scenario can handle ~ 6 Nodes with minimum throughput (100MB/s) provided they are adequately sized in terms of CPU and memory, in order to saturate the backend pipe (6 x 100 MB/s x 5 = 3000 MB/s). Each Node should have ~ 2 Gbps network bandwidth to accommodate NS + EW traffic patterns.
    - If we consider the recommended per Node throughput of 200MB/s then, it would be 3 such Nodes per hypervisor (3 x 200 MB/s x 5 = 3000 MB/s) Each Node should have ~ 4 Gbps network bandwidth to accommodate NS + EW traffic patterns.

Assume a Compute to Storage Node ratio - 4:1. This will vary depending on in-depth sizing and a more definitive sizing will be predicated by a good understanding of the workloads that are being intended to run on said infrastructure.

The two tables below illustrate the sizing exercise through a scenario that involves a storage cluster of 50 Nodes, and following the 4:1 assumption, 200 compute nodes.

- 50 Storage nodes

- 200 Compute nodes (4:1 ratio)

Table 1: Storage-Compute Node Level Sizing

Tier	Per node IO (MB/s)	Per node NS network (mbps)	per node EW (mbps)	Num of Nodes	Cluster IO (MB/s)	Cluster NS (Network) (mbps)	NS oversubscription
HDFS	400	3200	3200	50	20000	160000	1
Compute Node	100	800	800	200	20000	160000	1
Compute Node	100	800	800	200	10000	80000	2
Compute Node	100	800	800	200	6667	53333	3
Compute Node	100	800	800	200	5000	40000	4

Table 2: Storage and Compute Hypervisor level sizing

Tier	Per node IO (MB/s)	Per node NS network (mbps)	per node EW (mbps)	Num of Nodes	Hypervisor Oversubscription
Compute Hypervisor	600	4800	4800	33	6
Compute Hypervisor	400	3200	3200	50	4
Compute Hypervisor	300	2400	2400	67	3
Storage Hypervisor	1200	9600	9600	17	3
Storage Hypervisor	800	6400	6400	25	2
Storage Hypervisor	400	3200	3200	50	1

One can see that the table above provides the means to ascertain the capacity of the hardware for each tier of the private cloud, given different consolidation ratios and different throughput requirements.

Physical Network Topology

The best network topology for Hadoop clusters is spine-leaf. Each rack of hardware has its own leaf switch and each leaf is connected to every spine switch. Ideally we would not like to have any oversubscription between spine and leaf. That would result in having full line rate between any two nodes in the cluster.

The choice of switches, bandwidth and so on would of course be predicated on the calculations in the previous section.

If the Storage nodes and the Workload nodes happen to reside in clearly separate racks, then it is necessary to ensure that between the workload rack switches and the Storage rack switches, there is at least as much uplink bandwidth as the theoretical max offered by the Storage cluster. In other words, the sum of the bandwidth of all workload-only racks needs to be equivalent to the sum of the bandwidth of all storage-only racks.

For instance, taking the example from the previous section, we should have at least 60 Gbps uplink between the Storage cluster and the workload cluster nodes.

If we are to build the desired network topology based on the sizing exercise from above, we would need the following.

Layer	Network Per-port Bandwidth	Number of Ports required	Notes
Workload	25	52	Per sizing, we needed 20 Gbps per node. However, to simplify the design, let us pick single 25 Gbps ports instead of bonded pair of 10Gbps per node.
Storage	25	43	
Total Ports		95	

Assuming all the nodes are 2 RU in form factor, we would require 5 x 42 RU racks to house this entire set up.

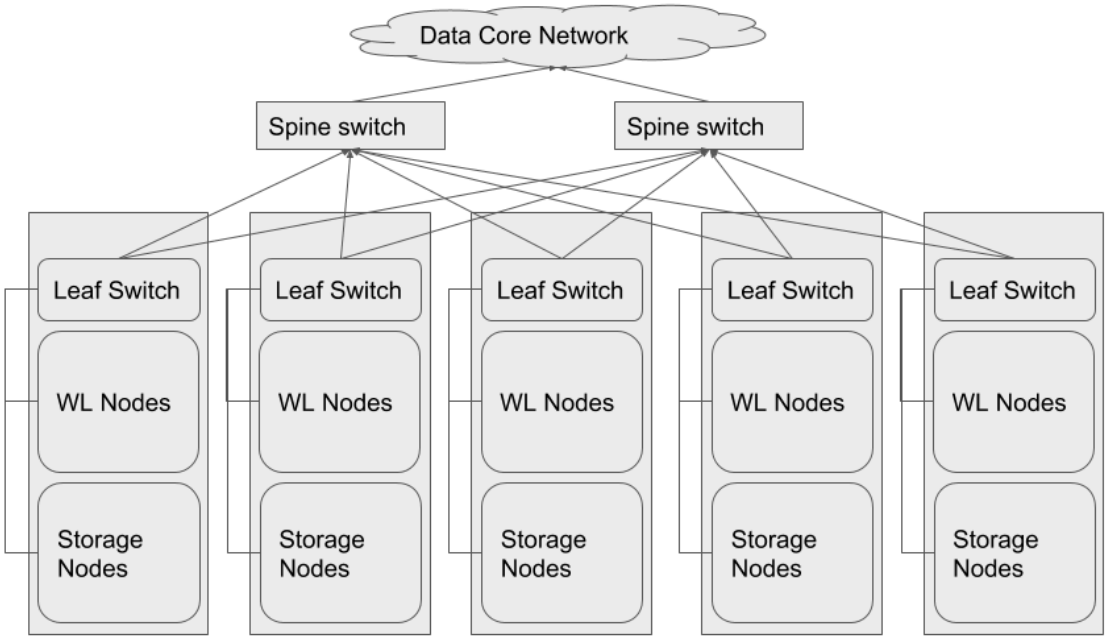
If we place the nodes from each layer as evenly distributed across the 5 Racks as we can, we would end up with following configuration.

Rack1	Rack2	Rack3	Rack4	Rack5
Spine (20 x 100 Gbps uplink + 2 x 100 Gbps to Core)			Spine (20 x 100 Gbps uplink + 2 x 100 Gbps to Core)	
ToR (18 x 25Gbps + 8 x 100 Gbps uplink)	ToR (20 x 25 Gbps + 8 x 100 Gbps uplink)	ToR (20 x 25 Gbps + 8 x 100 Gbps uplink)	ToR (18 x 25 Gbps + 8 x 100 Gbps uplink)	ToR (20 x 25 Gbps + 8 x 100 Gbps uplink)
10 x WL	11 x WL	11 x WL	10 x WL	11 x WL
8 x Storage	9 x Storage	9 x Storage	8 x Storage	9 x Storage

So that implies that we would have to choose ToR (Top of Rack) switches that have at least 20 x 25 Gbps ports and 8 x 100 Gbps uplink ports. Also the Spine switches would need at least 22 x 100 Gbps ports.

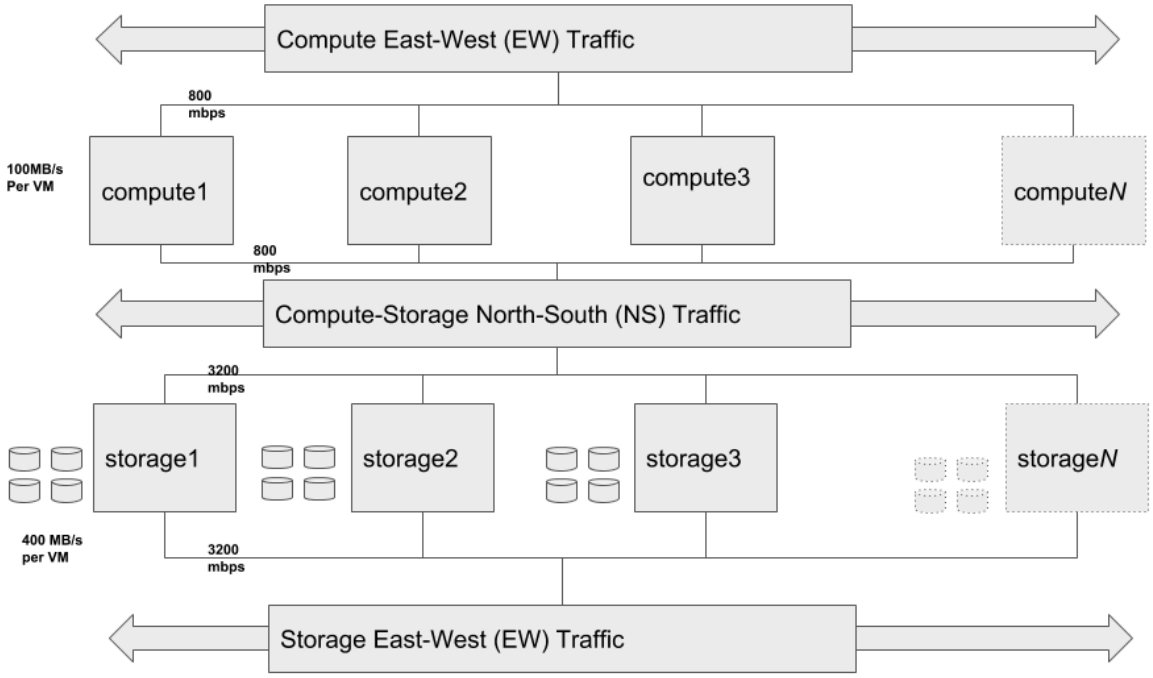
Using eight 100 Gbps uplinks from the leaf switches would result in almost 1:1 (1.125:1) oversubscription ratio between leaves (upto 20 x 25 Gbps ports) and the spine (4 x 100 Gbps per spine switch).

Mixing the Workload and Storage nodes in the way shown below, will help localize some traffic to each leaf and thereby reduce the pressure of N-S traffic (between Workload and Storage clusters) patterns.



Note: For sake of clarity, the spine switches have been shown outside the racks

The following diagram illustrates the logical topology at a virtual machine level.



The Storage E-W, Compute N-S and Compute E-W components shown above are not separate networks, but are the same network with different traffic patterns, which have been broken out in order clearly denote the different traffic patterns.