Cloud Management Console Cloud Connector Security White Paper

October 2023

Author:
Manjunath Shanbhag
manjunathns@in.ibm.com

Table of Contents

Overview

The HMC Cloud Connector is a Linux service. When you start Cloud Connector, it pushes data into the Cloud Management Console (CMC) database. Cloud Connector utilizes a one-way push model that initiates all outbound communication. This is not to be confused with the one-way computer communication, which sends a message without waiting for a response. Though the typical request-response style is used in Cloud Connector, it is never the responder but always the requestor. For the automatic network-based configuration, where Cloud Connector pulls the configuration file from the cloud database, HTTPS is used; and for application data flow (push) between Cloud Connector and CMC Data Ingestion Node, TCP with SSL is used.

Cloud Connector is preinstalled in HMC and can be started by using the command with the key mentioned in **CMC** > **Settings** > **Cloud Connector** > **Management**. Cloud Connector needs to connect to multiple end points, and hence needs an outbound connection to the IPs and the ports that are mentioned in **CMC** > **Settings** > **Cloud Connector** > **Management**. If the HMCs do not have an outbound connection to the endpoints, then proxies can be used for the connections.

What are Outbound Connections?

Cloud Connector needs to connect to multiple end points, and it hence needs an outbound connection to the IPs and the ports. There are multiple ways to make outbound connections:

- A - HMC Cloud Connector to CMC Cloud Portal Server
- B - HMC Cloud Connector to CMC Cloud Database (Configuration/Certificates)
- C - HMC Cloud Connector to CMC Cloud Data Ingestion Node

A - HMC Cloud Connector to CMC Cloud Portal Server

- With the user provided key, Cloud Connector establishes trust with the Cloud Portal Server. Cloud Connector pushes the user provided key to a verification endpoint of the Cloud Portal Server.
- After the key is successfully verified, CMC Cloud Portal Server returns the credentials for pulling the Cloud Connector configuration file and the SSL certificates.

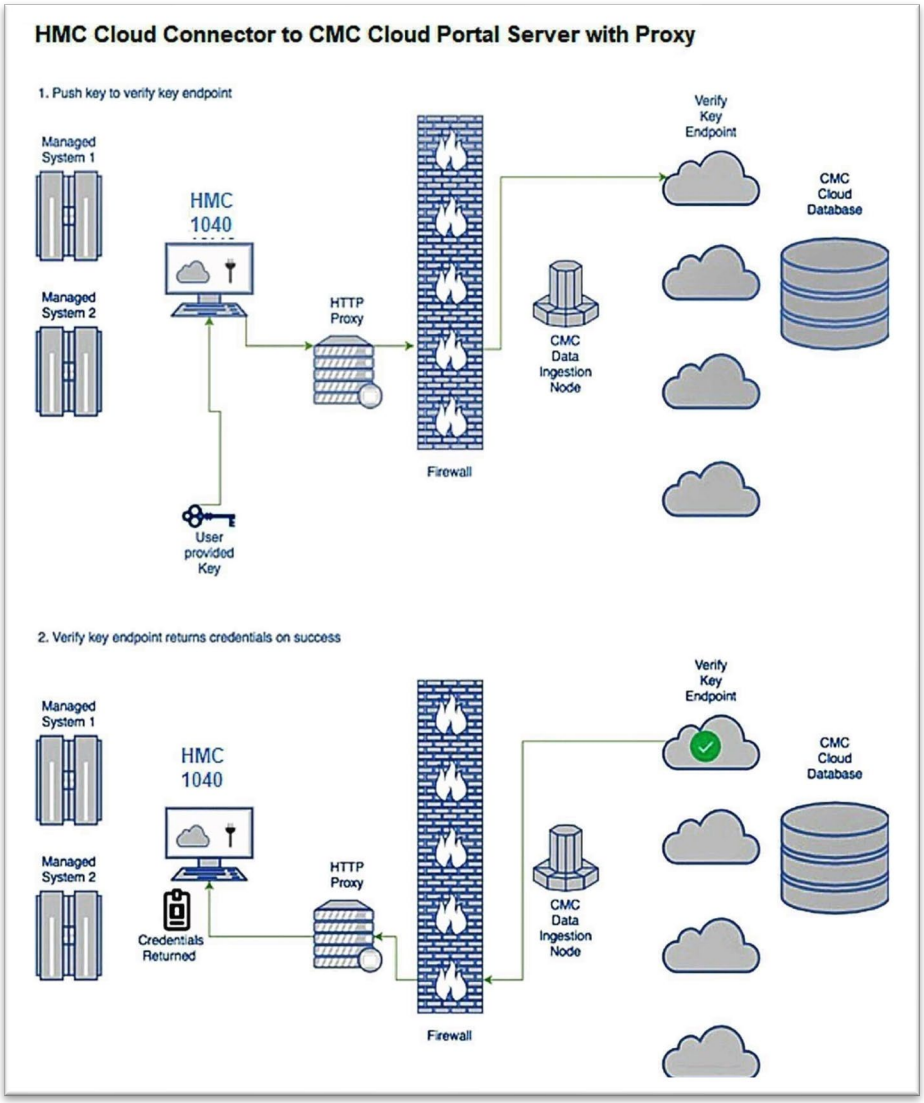

Figure 1 A - HMC Cloud Connector to CMC Cloud Portal Server

B - HMC Cloud Connector to CMC Cloud Database (Configuration/Certificates)

- With the credentials from Figure 1 A - HMC Cloud Connector to CMC Cloud Portal Server, the connection between Cloud Connector and CMC Cloud database is established either by using Cloudant's credentials or by authenticating the database against IBM Cloud's Identity and Access management (IAM). The endpoint for IAM is iam.cloud.ibm.com. Starting with CMC 1.17.0, by default, this connection is made by using IAM authentication. After this connection is established, Cloud Connector pulls the customer specific Cloud Connector configuration file from the CMC Cloud configuration database.

- With the credentials from the configuration file that is collected by using the connection that is explained in Figure 2 B1 – Cloud Connector to CMC Cloud Database, Cloud Connector pulls the SSL certificates and the key from the CMC Cloud Certificate database to secure the application data flow pipeline (Figure 4 C - Cloud Connector to CMC Cloud Data Ingestion Node).

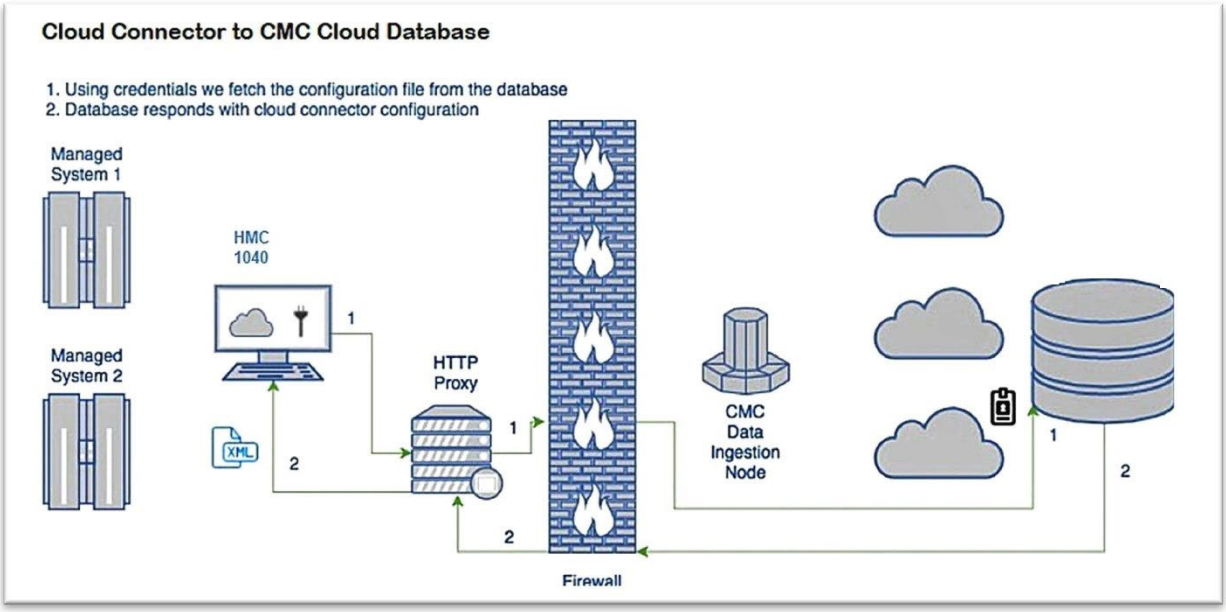

Figure 2 B1 – Cloud Connector to CMC Cloud Database

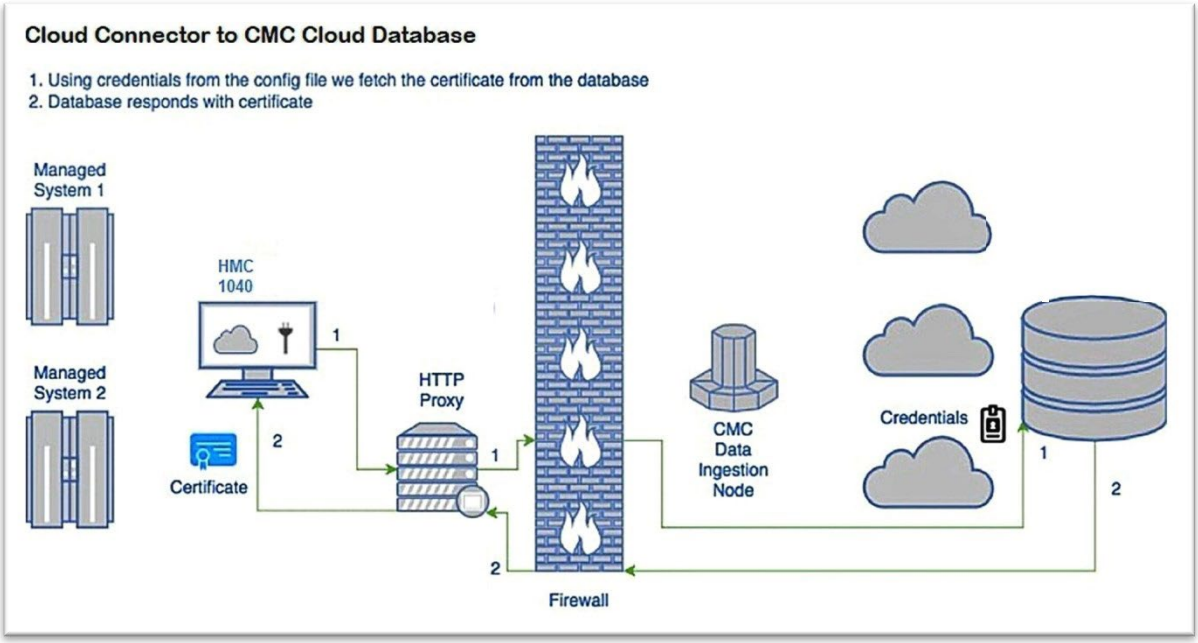

Figure 3 B2 – Cloud Connector to CMC Cloud Database

C - HMC Cloud Connector to CMC Cloud Data Ingestion Node

- Given the SSL certificates from the connection that is explained in Figure 3 B2 – Cloud Connector to CMC Cloud Database, a secure channel is created between Cloud Connector and CMC Cloud Data Ingestion Node, through which the cloud application data is pushed, such as Inventory data, Performance data, Logging data, and data related to any CMC application that will be provided in the future.
- Starting with HCM 9.1.941.0, you can start Cloud Connector only with the HTTP proxy option. If you start Cloud Connector with only HTTP proxy, then it uses HTTP proxy to establish connection between HMC and Ingestion Node as explained in Figure 4 C - Cloud Connector to CMC Cloud Data Ingestion Node). This option removes the dependency on SOCKS5 proxy, which was mandatory in the previous versions of HMC. The option that is explained in Figure 5 D - Cloud Connector to CMC Cloud Data Ingestion is still supported in the current versions of the HMCs, when Cloud Connector is started with both HTTP and SOCKS5 proxy options.

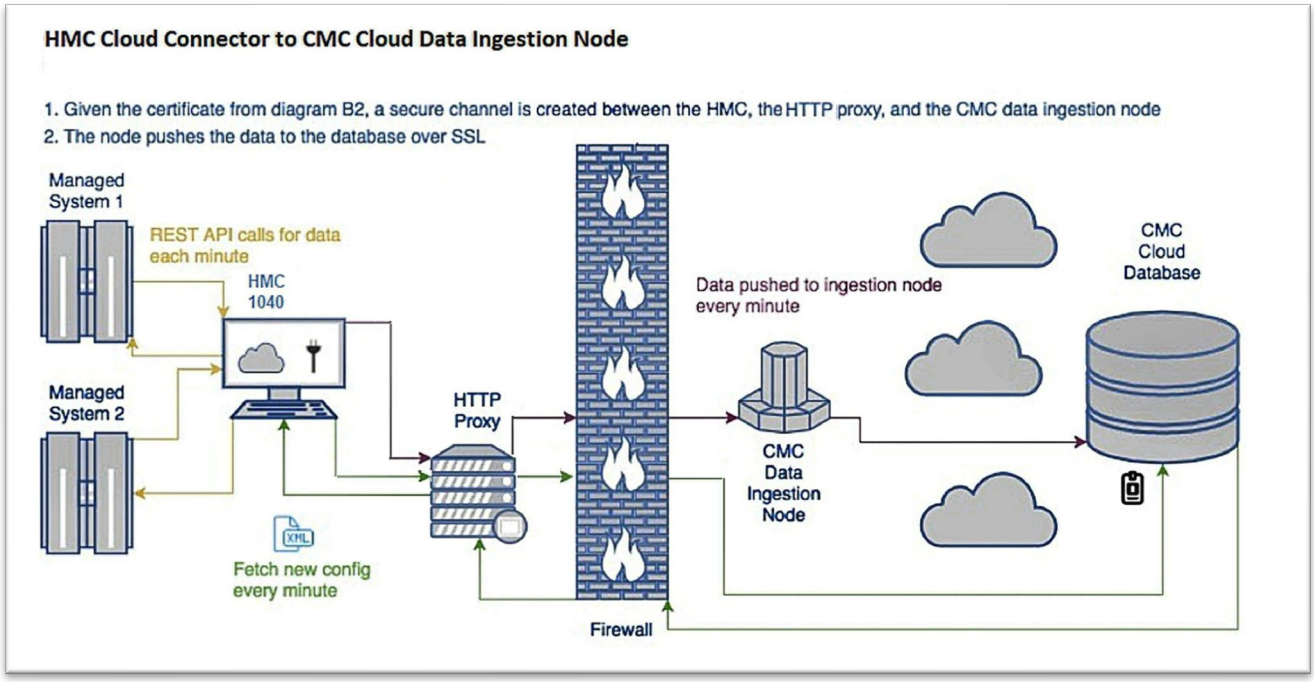

Figure 4 C - Cloud Connector to CMC Cloud Data Ingestion Node

- You can also connect HMC and Ingestion Node by using SOCKS5 proxy. SOCKS5 works with all HMC versions. Figure 5 D - Cloud Connector to CMC Cloud Data Ingestion shows how to connect HMC Cloud Connector to CMC Cloud Data Ingestion Node.

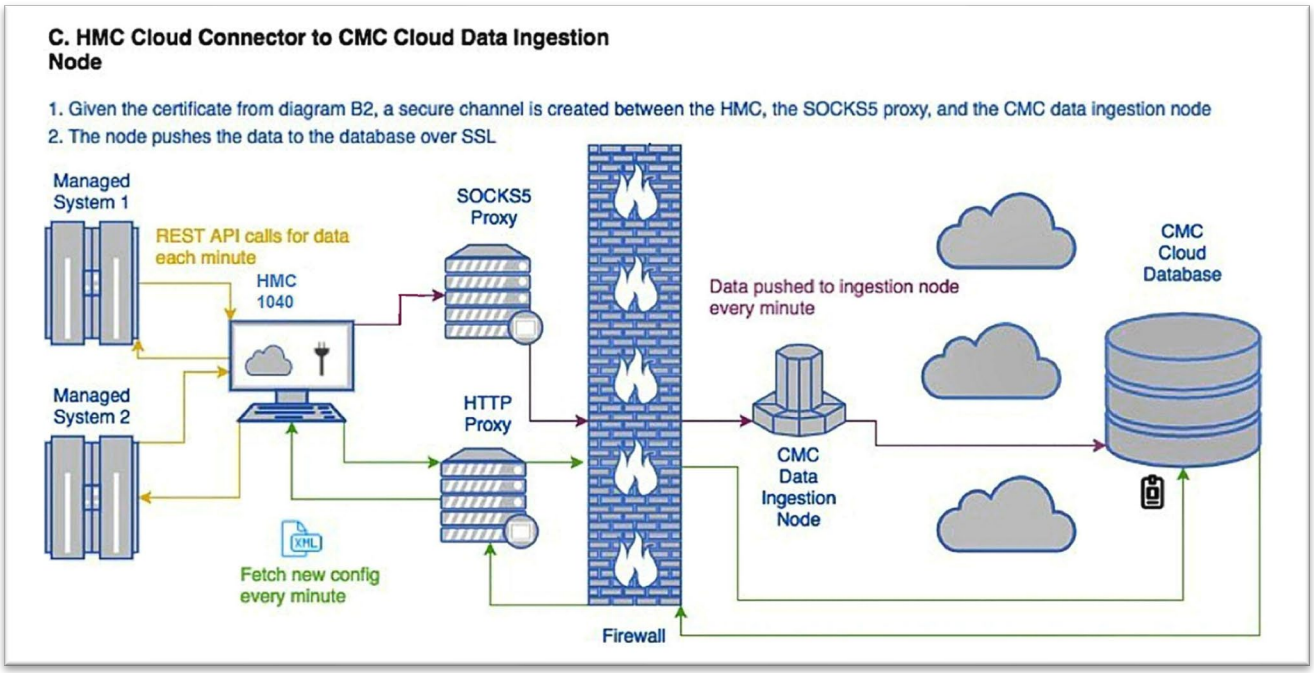

Figure 5 D - Cloud Connector to CMC Cloud Data Ingestion

How are Outbound Connections Secure?

The outbound connections that Cloud Connector makes are secure, network driven, and automatic.

A. HMC Cloud Connector to CMC Cloud Portal Server

The startup key for the HMC based Cloud Connector is used to establish a valid connection between Cloud Connector and the CMC Cloud Portal Server, and between Cloud Connector and the configuration database. Once a valid connection is established to the Cloud Portal Server, the credentials are returned to Cloud Connector. This enables a dynamic configuration and reconfiguration. To establish this connection, a security test is executed to assert that the startup key provided is valid. The test begins with a GET request from Cloud Connector to Cloud Portal Server, which will return a cross-site request forgery (XSRF) header. This XSRF header, along with a portion of the decoded key are then posted to the same endpoint of Cloud Portal Server. If the key is valid, Cloud Portal Server responds with a set of encoded credentials that give Cloud Connector the access to a database that contains the Cloud Connector configuration file.

All communication from Cloud Connector to Cloud Portal Server is secured by using the Transport Layer Security version 1.2 protocol (TLSv1.2) and the SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384 default cipher suite.

The following cipher suites are enabled for the first phase of the communication with the Cloud Portal Server:

```
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,


SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
SSL_RSA_WITH_AES_256_CBC_SHA256,
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384,
SSL_DHE_RSA_WITH_AES_256_CBC_SHA256,
SSL_DHE_DSS_WITH_AES_256_CBC_SHA256,
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA, SSL_RSA_WITH_AES_256_CBC_SHA,


SSL_ECDH_RSA_WITH_AES_256_CBC_SHA,
SSL_DHE_RSA_WITH_AES_256_CBC_SHA,
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
SSL_RSA_WITH_AES_128_CBC_SHA256,
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,


SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256,
SSL_DHE_RSA_WITH_AES_128_CBC_SHA256,
SSL_DHE_DSS_WITH_AES_128_CBC_SHA256,
SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
```

```
SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
SSL_RSA_WITH_AES_256_GCM_SHA384,

SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,
SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384,
SSL_DHE_DSS_WITH_AES_256_GCM_SHA384,
SSL_DHE_RSA_WITH_AES_256_GCM_SHA384,
SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256,

SSL_RSA_WITH_AES_128_GCM_SHA256,
SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,
SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256,
SSL_DHE_RSA_WITH_AES_128_GCM_SHA256,

SSL_DHE_DSS_WITH_AES_128_GCM_SHA256,
```

B. HMC Cloud Connector to CMC Cloud Database (Configuration/Certificates)

After the first phase of the communication with the Cloud Portal Server is completed, a secure SSL connection is established between Cloud Connector and the cloud database to fetch the configuration file. This configuration file contains the list of cloud applications that you have enabled, the data to push for these applications, and the data to filter. Filtered data is the data that is not sent to the Cloud Portal Server. Data can be filtered by using Managed System blocklisting and by using the data filter option to filter the system and the IP address of the partition and the IP address of Cloud Data Ingestion Node. The configuration file also provides the credentials for fetching an SSL certificate and the key pair that is used in the communication between Cloud Connector and Cloud Data Ingestion Node. However, the underlying network location and the mechanism that is used to fetch the certificates is the same. In short, an SSL connection is established, and the data is returned to Cloud Connector. Every minute, Cloud Connector checks for and fetches a new configuration, if it has changed.

All communication from Cloud Connector to the cloud database is secured by using the Transport Layer Security version 1.2 protocol (TLSv1.2).

C. HMC Cloud Connector to CMC Cloud Data Ingestion Node

After Cloud Connector is configured by using the automated configuration process, it collects data and pushes that data to Data Ingestion Node. This channel is secured by using SSL with mutual authentication that is done by using the certificate and the key that is mentioned in B. HMC Cloud Connector to CMC Cloud Database (Configuration/Certificates). Using mutual

authentication ensures that Cloud Connector only sends data to the trusted data ingestion nodes. The certificate and key are stored on the HMC filesystem but are only accessible by the root user.

How to Fetch Data from HMC API?

The final phase for data communication does not present a security risk but is worth discussing. Cloud Connector uses the HMC REST API to fetch inventory and performance data. New inventory data is fetched every minute and saved to the filesystem for delivery to the data ingestion node. The data must be saved to the filesystem because the data shipper code relies on this to keep track of what has or has not been successfully sent. For security, the files are only accessible to the root user as they contain all the inventory that is associated to that HMC. If no new inventory data is present (that is, the inventory is the same) nothing is saved and shipped. The performance data is saved and shipped every minute unless Performance Collection is disabled.

To connect to the HMC API, a special login procedure was built that only works for API calls that originate from the HMC itself to its own localhost or the endpoint 127.0.0.1. Once the login query is completed, the HMC API server saves a login token (session cookie) to the HMC filesystem so that the connector can use it to make queries to the API without the need to re-authenticate. This token may expire. If it does, a new one is generated and saved to the filesystem using the same mechanism. The token is only usable for queries to `localhost`. In other words, the token cannot be hijacked and used for queries against the actual HMC hostname or the IP from a remote server. Thus, it is only useful for the services that run on the HMC. The saved token is only accessible to the root user. TLS version 1.2 is used to fetch data from the HMC API. Since the requests initiate from the HMC themselves, this is an added security measure. In short, no man-in-the-middle attack is viable for requests coming from the HMC to the HMC.

Some information, such as logging data, is not available through HMC REST APIs. Such data is saved in the HMC's filesystem, accessible only by the root user. Cloud Connector reads the data from such files, filters, and pushes the data to the ingestion node.

The application built in Cloud Management Console uses data from CMC Cloud Database. The application needs data, such as inventory, performance metrics, logging information, and so on, based on its use cases. This data is being pushed by Cloud Connector that runs in the HMC, which uses different REST API to fetch this information, or the data saved by HMC in the filesystem.

HMC provides rich set of REST APIs (HMC REST APIs), which provides information of the partition, systems, IO, performance metrics, and so on. But Cloud Connector retrieves and pushes attributes required for the applications that are running in Cloud Management Console. The *Inventory attributes pushed by Cloud Connector* and *Performance metrics attributes pushed by Cloud Connector* tables list the attributes that are pushed by Cloud Connector.

How to Mask CMC Attributes?

By enabling the attribute masking feature, you can ensure that sensitive data does not leave your data centre. After enabling this feature, Cloud Connector masks sensitive data and sends the masked data to the CMC server, and the CMC UI displays these masked values of the resource attributes on all the CMC pages and apps.

When Attribute Masking is enabled, the CMC APIs also contain masked data in their response.

To enable Attribute Masking, click **Settings** > **Cloud Connector** > **Cloud Connector Management**, scroll down to the end of the page, and then set **Attribute Masking** to **On**.

The attribute masking feature is available with HMC V10.2.1040.0 and later only. Data from earlier HMC versions is not masked and will continue to be displayed unmasked even when Attribute Masking is enabled.

Important:

- If a system is connected to multiple HMCs, for the attribute masking feature to work, the version of all the connected HMCs must be HMC V10.2.1040.0 or later.
- After you enable Attribute Masking, to see the masked values on the UI, wait for a minimum of five minutes, then reload CMC in the browser.
- The historical data that was collected before you enabled Attribute Masking remains unmasked. For example, the values for the Partition Lifecycle record, which were collected in the Logging application before you enabled Attribute Masking, are not masked on the UI.

Table 1: Inventory attributes pushed by Cloud Connector

Resource	Attribute Name	Masked
ManagedSystem	SystemName	√
	State	
	SystemFirmware	
	SystemLocation	√
	Description	√
	SystemType	
	MachineTypeModelAndSerialNumber.MachineType	
	MachineTypeModelAndSerialNumber.Model	
	MachineTypeModelAndSerialNumber.SerialNumber	
	Hostname	√
	PrimaryIPAddress	√
	UUID	
	RemainingHoursInMeteredPoolAuthPeriod	
	PowerEnterprisePoolID	
	ProcessorThrottling	

Resource	Attribute Name	Masked
	AssociatedSystemProcessorConfiguration.ConfigurableSystemProcessorUnits	
	AssociatedSystemProcessorConfiguration.InstalledSystemProcessorUnits	
	AssociatedSystemProcessorConfiguration.CurrentAvailableSystemProcessorUnits	
	AssociatedSystemMemoryConfiguration.InstalledSystemMemory	
	AssociatedSystemMemoryConfiguration.ConfigurableSystemMemory	
	AssociatedSystemMemoryConfiguration.CurrentAvailableSystemMemory	
	AssociatedSystemMemoryConfiguration.HugePageSize	
	AssociatedSystemMemoryConfiguration.MemoryUsedByHypervisor	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.BusGroupingRequired	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.Description	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.FeatureCodes	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IOUnitPhysicalLocation	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCAdapterID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCIClass	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCIDeviceID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCISubsystemDeviceID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCIManufacturerID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCIRevisionID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCIVendorID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.PCISubsystemVendorID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.AlternateLoadSourceAttached	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.ConsoleCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.DirectOperationsConsoleCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.IOP	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.IOPInfoStale	

Resource	Attribute Name	Masked
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.IOPoolID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.LANConsoleCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.LoadSourceAttached	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.LoadSourceCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.OperationsConsoleAttached	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIBMiIOSlot.OperationsConsoleCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.AdapterID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.Description	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.DeviceName	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.DynamicReconfigurationConnectorName	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.PhysicalLocation	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.UniqueDeviceID	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.LogicalPartitionAssignmentCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.RelatedIOAdapter.IOAdapter.DynamicPartitionAssignmentCapable	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.SlotDynamicReconfigurationConnectorIndex	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.SlotDynamicReconfigurationConnectorName	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.SlotPhysicalLocationCode	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.SRIOVCapableDevice	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.SRIOVCapableSlot	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.SRIOVLogicalPortsLimit	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.ParentDynamicReconfigurationConnectorIndex	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.ParentName	

Resource	Attribute Name	Masked
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PCIDeviceId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PCIVendorId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PCISubsystemDeviceId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PCISubsystemVendorId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PCIRevisionId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.ProgrammingInterfaceClass	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PCIClassCode	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.DeviceType	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PrimaryDeviceFunction	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.SerialNumber	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.FruNumber	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.PartNumber	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.CCIN	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.SlotChildId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.ParentSlotChildId	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.Size	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.SizeMetric	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.NumEnclosureBays	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.LocationCode	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.MicroCodeVersion	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.WWPN	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.WWNN	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.MacAddressValue	
	AssociatedSystemIOConfiguration.IOSlots.IOSlot.IORDevices.IORDevice.Description	

Resource	Attribute Name	Masked
	PowerSupplies.PowerSupply.LocationCode	
	PowerSupplies.PowerSupply.FruNumber	
	PowerSupplies.PowerSupply.SerialNumber	
	PowerSupplies.PowerSupply.State	
	PowerSupplies.PowerSupply.Health	
	PowerSupplies.PowerSupply.Description	
	PowerSupplies.PowerSupply.MemberId	
	FANs.FAN.LocationCode	
	FANs.FAN.FruNumber	
	FANs.FAN.SerialNumber	
	FANs.FAN.State	
	FANs.FAN.Health	
	FANs.FAN.Description	
	FANs.FAN.MemberId	
Shared ProcessorPool	AssignedPartitionsLinks	
	CurrentReservedProcessingUnits	
	MaximumProcessingUnits	
	PoolID	
	AvailableProcUnits	
	PoolName	√
VirtualSwitch	SwitchID	
	SwitchMode	
	SwitchName	√

Resource	Attribute Name	Masked
SharedMemory Pool	CurrentPoolMemory	
LogicalPartition	PartitionID	
	PartitionName	√
	PartitionState	
	PartitionType	
	ResourceMonitoringControlState	
	ResourceMonitoringIPAddress	√
	PartitionUUID	
	AssociatedManagedSystemLink	
	PowerVMManagementCapable	
	OperatingSystemVersion	
	OperatingSystemType	
	Description	√
	PartitionMemoryConfiguration.SharedMemoryEnabled	
	PartitionMemoryConfiguration.CurrentMemory	
	PartitionProcessorConfiguration.CurrentHasDedicatedProcessors	
	PartitionProcessorConfiguration.CurrentDedicatedProcessorConfiguration.CurrentProcessors	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.AllocatedVirtualProcessors	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.CurrentProcessingUnits	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.RuntimeProcessingUnits	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.CurrentSharedProcessorPoolID	
	PartitionID	

Resource	Attribute Name	Masked
VirtualIOServer	PartitionName	√
	PartitionState	
	PartitionType	
	PartitionUUID	
	ResourceMonitoringControlState	
	ResourceMonitoringIPAddress	√
	AssociatedManagedSystemLink	
	PowerVMManagementCapable	
	OperatingSystemVersion	
	OperatingSystemType	
	Description	√
	PartitionMemoryConfiguration.SharedMemoryEnabled	
	PartitionMemoryConfiguration.CurrentMemory	
	PartitionProcessorConfiguration.CurrentHasDedicatedProcessors	
	PartitionProcessorConfiguration.CurrentDedicatedProcessorConfiguration.CurrentProcessors	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.AllocatedVirtualProcessors	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.CurrentProcessingUnits	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.RuntimeProcessingUnits	
	PartitionProcessorConfiguration.CurrentSharedProcessorConfiguration.CurrentSharedProcessorPoolID	
Management Console	ManagementConsoleName	√
	MachineTypeModelAndSerialNumber.MachineType	
	MachineTypeModelAndSerialNumber.Model	
	MachineTypeModelAndSerialNumber.SerialNumber	

Resource	Attribute Name	Masked
	ManagedSystemsLinks	
	NetworkInterfaces.ManagementConsoleNetworkInterface.InterfaceName	
	NetworkInterfaces.ManagementConsoleNetworkInterface.NetworkAddress	√
	VersionInfo.BuildLevel	
	VersionInfo.Maintenance	
	VersionInfo.Minor	
	VersionInfo.Release	
	VersionInfo.Version	
	VersionInfo.ServicePackName	
	ProcConfiguration.NumberOfProcessors	
	ProcConfiguration.ModelName	
	ProcConfiguration.Architecture	
	MemConfiguration.TotalMemory	
	MemConfiguration.TotalSwapMemory	
	UVMID	
	IPAddress	√
Cluster	UUID	
	ClusterName	√
	ClusterID	
	RepositoryDisk.PhysicalVolume.Description	√
	RepositoryDisk.PhysicalVolume.UniqueDeviceID	
	RepositoryDisk.PhysicalVolume.VolumeCapacity	
	RepositoryDisk.PhysicalVolume.VolumeName	√
	RepositoryDisk.PhysicalVolume.VolumeState	
	RepositoryDisk.PhysicalVolume.IsFibreChannelBacked	

Resource	Attribute Name	Masked
	RepositoryDisk.PhysicalVolume.StorageLabel	√
	ClusterSharedStoragePoolLink	
	Node.Node.HostName	√
	Node.Node.PartitionID	
	Node.Node.State	
	Node.Node.VirtualIOServerLevel	
	Node.Node.VirtualIOServerLink	
	Node.Node.MachineTypeModelAndSerialNumber.MachineType	
	Node.Node.MachineTypeModelAndSerialNumber.Model	
	Node.Node.MachineTypeModelAndSerialNumber.SerialNumber	
	ClusterCapabilities.IsTierCapable	
	ClusterCapabilities.IsTierMirrorCapable	
SharedStorage Pool	UUID	
	MultiDataTierConfigured	
	MultiFailureGroupConfigured	
	Capacity	
	FreeSpace	
	TotalLogicalUnitSize	
	StoragePoolName	√
	UniqueDeviceID	
	AssociatedClusterLink	
	AssociatedTiersLinks	
	PhysicalVolumes.PhysicalVolume.Description	√
	PhysicalVolumes.PhysicalVolume.UniqueDeviceID	
	PhysicalVolumes.PhysicalVolume.VolumeCapacity	

Resource	Attribute Name	Masked
	PhysicalVolumes.PhysicalVolume.VolumeName	√
	PhysicalVolumes.PhysicalVolume.VolumeState	
	PhysicalVolumes.PhysicalVolume.IsFibreChannelBacked	
	PhysicalVolumes.PhysicalVolume.StorageLabel	√
Tier	UUID	
	Name	√
	UniqueDeviceID	
	Type	
	IsDefault	
	FreeSpace	
	OverCommitSpace	
	Capacity	
	FreeSpaceThreshold	
	OverCommitSpaceThreshold	
	TotalLogicalUnitSize	
	MirrorState	
	AssociatedSharedStoragePoolLink	
	FailureGroups.FailureGroup.Name	√
	FailureGroups.FailureGroup.UniqueDeviceID	
	FailureGroups.FailureGroup.Capacity	
	FailureGroups.FailureGroup.State	
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.Description	√
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.UniqueDeviceID	
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.VolumeCapacity	
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.VolumeName	√

Resource	Attribute Name	Masked
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.VolumeState	
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.IsFibreChannelBacked	
	FailureGroups.FailureGroup.PhysicalVolumes.PhysicalVolume.StorageLabel	√

Table 2: Performance metrics attributes pushed by Cloud Connector

Metrics Resource	Attribute Name	Masked
ManagedSystemPreferences	SystemName	√
	SystemUUID	
	IsAggregationEnabled	
	SystemMTMS	
	IsEnergyMonitorEnabled	
	ConsoleMTMS	
SharedStoragePoolPreferences	ClusterName	√
	PoolName	√
	PoolId	
	ClusterId	
	SSPUUID	
	IsAggregationEnabled	
	ConsoleMTMS	
ManagedSystemMetrics	AssignedMemToLpars	
	ConsoleMTMS	
	HEAPort.DRCIndex	
	HEAPort.ID	
	HEAPort.PhysicalLocation	

Metrics Resource	Attribute Name	Masked
	HEAPort.ReceivedBytes	
	HEAPort.ReceivedPackets	
	HEAPort.SentBytes	
	HEAPort.SentPackets	
	Network.ReceivedBytes	
	Network.ReceivedPackets	
	Network.SentBytes	
	Network.SentPackets	
	SharedMemoryPool.AssignedMemToLpars	
	SharedMemoryPool.AssignedMemToSysFirmware	
	SharedMemoryPool.TotalMemory	
	SharedProcessorPool.AssignedProcUnits	
	SharedProcessorPool.AvailableProcUnits	
	SharedProcessorPool.ConfiguredProcUnits	
	SharedProcessorPool.ID	
	SharedProcessorPool.Name	√
	SharedProcessorPool.UtilizedProcUnits	
	SRIOVPort.DRCIndex	
	SRIOVPort.ID	
	SRIOVPort.PhysicalLocation	
	SRIOVPort.ReceivedBytes	
	SRIOVPort.ReceivedPackets	
	SRIOVPort.SentBytes	
	SRIOVPort.SentPackets	
	Storage.NumOfReads	

Metrics Resource	Attribute Name	Masked
	Storage.NumOfWrites	
	Storage.ReadBytes	
	Storage.WriteBytes	
	SystemAvailableMemory	
	SystemAvailableProcUnits	
	SystemConfigurabelProcUnits	
	SystemConfigurableMemory	
	SystemFWAssignedMem	
	SystemMTMS	
	SystemName	√
	SystemTotalMemory	
	SystemTotalProcUnits	
	SystemUtilizedProcUnits	
	SystemUUID	
LogicalPartitionMetrics	ConsoleMTMS	
	EntitledProcunits	
	ID	
	LPARUUID	
	MemMode	
	Name	√
	Network.ReceivedBytes	
	Network.ReceivedPackets	
	Network.SentBytes	
	Network.SentPackets	

Metrics Resource	Attribute Name	Masked
	ProcMode	
	SRIOVLogicalPortMetrics.DRCIndex	
	SRIOVLogicalPortMetrics.LocationCode	
	SRIOVLogicalPortMetrics.PhysicalPortDRCIndex	
	SRIOVLogicalPortMetrics.PhysicalPortID	
	SRIOVLogicalPortMetrics.ReceivedBytes	
	SRIOVLogicalPortMetrics.ReceivedPackets	
	SRIOVLogicalPortMetrics.SentBytes	
	SRIOVLogicalPortMetrics.SentPackets	
	State	
	Storage.NumOfReads	
	Storage.NumOfWrites	
	Storage.ReadBytes	
	Storage.WriteBytes	
	SystemName	√
	SystemUUID	
	UtilizedProcUnits	
	VFCAdapterMetrics.LocationCode	
	VFCAdapterMetrics.NumOfReads	
	VFCAdapterMetrics.NumOfWrites	
	VFCAdapterMetrics.ReadBytes	
	VFCAdapterMetrics.VIOSID	
	VFCAdapterMetrics.WriteBytes	
	VFCAdapterMetrics.WriteBytes	
	VirtualEthMetrics.IsPVID	

Metrics Resource	Attribute Name	Masked
	VirtualEthMetrics.LocationCode	
	VirtualEthMetrics.ReceivedBytes	
	VirtualEthMetrics.ReceivedPackets	
	VirtualEthMetrics.SentBytes	
	VirtualEthMetrics.SentPackets	
	VirtualEthMetrics.VLANId	
	VirtualEthMetrics.VSwitchID	
	VSCSIAadapterMetrics.LocationCode	
	VSCSIAadapterMetrics.NumOfReads	
	VSCSIAadapterMetrics.NumOfWrites	
	VSCSIAadapterMetrics.ReadBytes	
	VSCSIAadapterMetrics.VIOSID	
VirtualIOServerMetrics	AssignedMemory	
	ConsoleMTMS	
	EntitledProcUnits	
	FCAdapterMetrics.ID	
	FCAdapterMetrics.LocationCode	
	FCAdapterMetrics.NumOfReads	
	FCAdapterMetrics.NumOfWrites	
	FCAdapterMetrics.ReadBytes	
	FCAdapterMetrics.WriteBytes	
	ID	
	Name	√
	Network.ReceivedBytes	

Metrics Resource	Attribute Name	Masked
	Network.ReceivedPackets	
	Network.SentBytes	
	Network.SentPackets	
	NetworkAdapterMetrics.ID	
	NetworkAdapterMetrics.LocaionCode	
	NetworkAdapterMetrics.ReceivedBytes	
	NetworkAdapterMetrics.ReceivedPackets	
	NetworkAdapterMetrics.SentBytes	
	NetworkAdapterMetrics.SentPackets	
	SEA.BridgedAdapters	
	SEA.LocaionCode	
	SEA.ReceivedBytes	
	SEA.ReceivedPackets	
	SEA.SentBytes	
	SEA.SentPackets	
	SRIOVLogicalPortMetrics.DRCIndex	
	SRIOVLogicalPortMetrics.LocationCode	
	SRIOVLogicalPortMetrics.PhysicalPortDRCIndex	
	SRIOVLogicalPortMetrics.PhysicalPortID	
	SRIOVLogicalPortMetrics.ReceivedBytes	
	SRIOVLogicalPortMetrics.ReceivedPackets	
	SRIOVLogicalPortMetrics.SentBytes	
	SRIOVLogicalPortMetrics.SentPackets	
	State	
	Storage.NumOfReads	

Metrics Resource	Attribute Name	Masked
	Storage.NumOfWrites	
	Storage.ReadBytes	
	Storage.WriteBytes	
	SystemName	√
	SystemUUID	
	SystemUUID	
	UtilizedMemory	
	VirtualEthMetrics.IsPVID	
	VirtualEthMetrics.LocationCode	
	VirtualEthMetrics.ReceivedBytes	
	VirtualEthMetrics.ReceivedPackets	
	VirtualEthMetrics.SentBytes	
	VirtualEthMetrics.SentPackets	
	VirtualEthMetrics.VLANId	
	VirtualEthMetrics.VSwitchID	
SharedStoragePoolMetrics	ReadBytes	
	WriteBytes	
	NodeMetrics.ReadBytes	
	NodeMetrics.WriteBytes	
	NodeMetrics.VIOSName	√
	NodeMetrics.ID	
	NodeMetrics.SystemMTMS	
	NodeMetrics.TierMetrics.ReadBytes	
	NodeMetrics.TierMetrics.WriteBytes	

Metrics Resource	Attribute Name	Masked
	NodeMetrics.TierMetrics.TierName	√
	ClusterName	√
	PoolName	√
	PoolID	
	ClusterID	
	ConsoleMTMS	

The attributes that are required by the Logging application are computed and stored in a file in the local file system of HMC. Cloud Connector reads these attributes and pushes it to the CMC Cloud database. The Logging application requires the following attributes:

Table 3: Attributes required by the Logging application

Attributes	Masked
SourceCEC	√
DestinationCEC	√
SourceHMCIP	√
DestinationHMCIP	√
OperationResult	
PartitionName	√
HMCUser	√
OperationType	
NumOfConcurrentOperations	
IsRemoteOperation	
ErrorCodes	
AbortSide	
SPPName	√
SourceCECName	√
DestinationCECName	√
SourcePrimaryMSPName	√

Attributes	Masked
SourceSecondaryMSPName	√
DestinationPrimaryMSPName	√
DestinationSecondaryMSPName	√
TotalMemoryTransferData	
ConsoleMTMS	
StartSuspendDelta	
SuspendResumeDelta	
ResumeCompleteDelta	
EstimatedLineSpeedInMbits	
AverageResumeLatencyInMillis	
BytesSentDuringSpeculative	
BytesSentDuringSuspend	
BytesSentDuringResume	
BytesDirtyAtResume	
BytesDemandPaged	
MigrationDurationInTB	
NumOfLPARVirtualAdapters	
LparConfigValidationTime	
LparCleanUpTime	

Proxies

For all outbound connections, proxies can be configured to provide added security. For the connections between Cloud Connector and the Cloud Portal Server, and between Cloud Connector and the database, an HTTP proxy can be used. For HMC l 9.1.941.0 and earlier, only the Basic Authentication is supported. With HMC 9.1.941.1, Cloud Connector supports Kerberos, LDAP, and Digest-md5 based proxy server authentication apart from Basic Authentication. While starting Cloud Connector, an attribute to be used for the proxy connection can be specified on the authentication type. The default authentication used is Basic Authentication. The credentials, which are used for the initial HTTP CONNECT request from Cloud Connector to the proxy by using Basic Authentication, will not be encrypted. Since this connection occurs behind the firewall, it does not present a security concern. After the SSL tunnel is established, all data leaving the proxy will be properly encrypted by using the mechanisms described in Outbound Connections.

For HMC 9.1.941.0 and earlier, for the connection between Cloud Connector and the data ingestion node, a SOCKS5 proxy must be configured. Again, only Basic Authentication is supported. With HMC 9.1.941.1, Cloud Connector can connect to the ingestion node by using only the HTTP proxy. For HMC 9.1.941.0 and later, it is not mandatory to use the SOCKS5 proxy to connect the ingestion node.

While starting HMC, when a proxy is configured by using Basic Authentication credentials, the password that you provide is encrypted by using a shared secret between the HMC CLI code and Cloud Connector process. From this secret, a key is generated by using Password Based Encryption with the MD5 and DES algorithms. The key and a salt are used to encrypt the password. The encrypted password is then saved to the filesystem where it is only accessible by the root user. Once Cloud Connector process begins running as root, it reads the encrypted key and decrypts it to construct the initial HTTP CONNECT request to the proxy server. For the SOCKS5 server, the password is also encrypted in the initial phase where it is transferred from the HMC CLI to Cloud Connector process. However, due to the constraints provided by our data shipper process, the SOCKS5 proxy password is stored in clear text in the data shipper configuration file. For added security, the file is only accessible to the root user. When Basic Authentication is used for the initial HTTP CONNECT request with cleartext, there is not an urgent need for the password to be encrypted at rest since a motivated attacker could simply capture the network packets and discover the credentials. However, since all this occurs behind the firewall in a secured data centre, the threat of an attacker having access to the HMC is minimal. If it does occur, greater risks than the discovery of the proxy password surely exist.