Vertiv™ Avocent® MP1000 Management Platform and Vertiv™ Avocent® MP1000VA Management Platform Virtual Appliance

High Availability (HA) Technical Note

DECEMBER 2024

Technical Note Section Outline

1. Overview

NOTE: At this time, the former Vertiv™ Avocent® ADX platform is transitioning into the Vertiv™ Avocent® DSView™ solution. During this transition, there may temporarily still be references to "ADX" within product-related features and documentation.

The Vertiv™ Avocent® MP1000 Management Platform and the Vertiv™ Avocent® MP1000VA Management Platform Virtual Appliance support High Availability (HA), which provides server redundancy and enables you to reduce downtime through data synchronization and replication on a maximum of three nodes within a "cluster" of appliances. A cluster contains a Primary node that replicates its data to one or two Standby nodes. Standby nodes are promoted to Primary mode if any system service fails or can be promoted manually to allow for maintenance operations such as firmware upgrades.

This technical note provides guidance for various high-level HA activities. Instructions for other basic HA operations can be found in the Vertiv™ Avocent® MP1000 Management Platform User Guide, which can be found on the product page under the *Documents & Downloads* tab.

IMPORTANT NOTE**: Some operations detailed in this technical note assume that you are already familiar with API operations within the Vertiv™ Avocent® DSView™ solution and have basic knowledge of Python programming language. Additional information on the APIs within the Vertiv™ Avocent® DSView™ solution is available in the Vertiv™ Avocent® DSView™ Solution API Guide (see the Session Management section) available here: API Guide.**

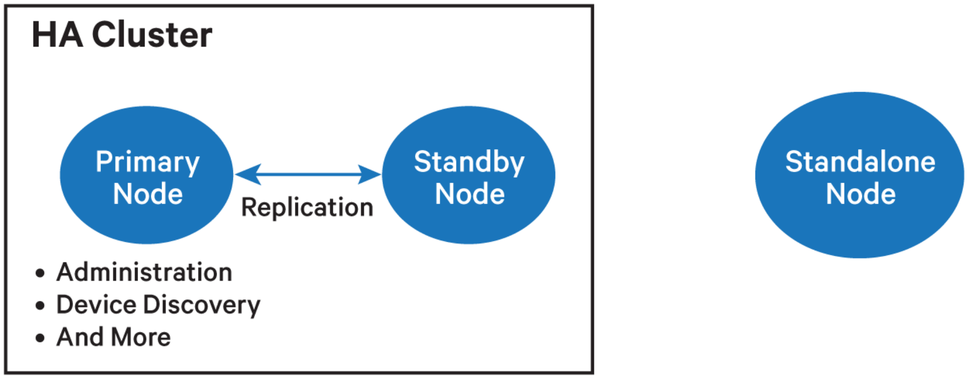
2. Terminology

For your convenience, a list of helpful HA terminology is provided below:

- **Primary Node** – The master node in an HA cluster from which all end-user operations should take place, such as HA administration, target discovery, and authentication provider configuration changes.
- **Standby Node** – The hot standby node; replication flows between the Primary and the Standby while system events flow from the Standby back to the Primary.
- **Standalone Node** – A node that is not part of an HA cluster.
- **HA Cluster** – A High Availability cluster consists of one Primary node which replicates its data to one or two Standby nodes. A maximum of three nodes is permitted for a single cluster.

The following figure highlights the relationship of the nodes, including the Primary and Standby nodes that operate within an HA cluster and the Standalone node that operates independently.

Figure 2.1 HA Cluster and Nodes



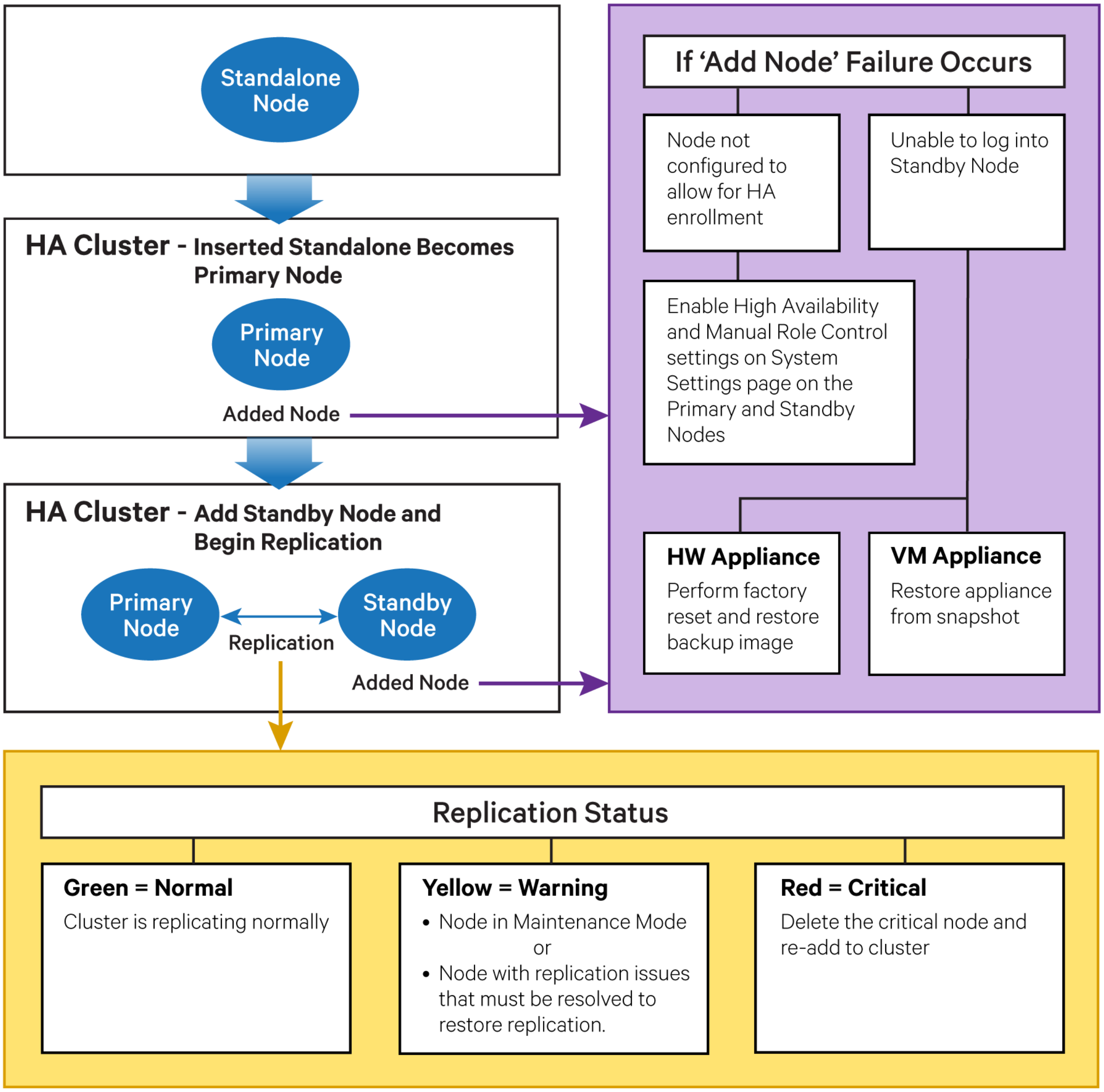
3. Setting Up an HA Cluster

This section contains critical information about setting up an HA cluster, including the following:

- Platform Types
- Setup Things to Know
- Setup Failure Conditions and Resolutions
- Licensing

The following figure explains the process of setting up an HA cluster. The information described in this figure will be explained in more depth later in this section. Instructions for setting up an HA cluster can be found in the Vertiv™ Avocent® MP1000 Management Platform User Guide located on the product page under the *Documents & Downloads* tab.

Figure 3.1 Setting Up an HA Cluster



Standalone Node
HA Cluster - Inserted Standalone Becomes Primary Node
Primary Node
Added Node
HA Cluster - Add Standby Node and Begin Replication
Primary Node
Standby Node
Replication
Added Node
If 'Add Node' Failure Occurs
Node not configured to allow for HA enrollment
Unable to log into Standby Node
Enable High Availability and Manual Role Control settings on System Settings page on the Primary and Standby Nodes
HW Appliance
Perform factory reset and restore backup image
VM Appliance
Restore appliance from snapshot
Replication Status
Green = Normal
Cluster is replicating normally
Yellow = Warning
• Node in Maintenance Mode
or
• Node with replication issues that must be resolved to restore replication.
Red = Critical
Delete the critical node and re-add to cluster

Platform Types

To begin the setup process, you should choose the platform type on which you will create the cluster. There are currently two options for setting up an HA cluster with the Vertiv™ Avocent® MP1000 Management Platform:

- On a Vertiv™ Avocent® MP1000 hardware appliance with 2 to 3 nodes
- On a virtual machine using 2 to 3 instances, one instance per node

NOTE: It is recommended that clusters not contain a heterogeneous mix of node types due to wide variances in the performance of the Vertiv™ Avocent® MP1000 Management Platform Virtual Appliance nodes.

After choosing the platform to use (hardware or virtual machine), you will select which appliance to configure as the Primary node. For detailed instructions on the setup process, refer to the Vertiv™ Avocent® MP1000 Management Platform User Guide. To configure your system with HA, you must have the appropriate licenses activated and installed. For a guide to understanding the licensing process for High Availability, see the Licensing section of this document.

Setup Things to Know

Before adding nodes to the cluster:

- Each node in the cluster must have a fixed IP address. It is acceptable for the hosts to have a fixed address in your DHCP table. Otherwise, each node must be set to use a static IP address.
- Each node must use the same firmware version.
- It is recommended that:
    - For virtual machines, you create a snapshot image of the node.
    - For hardware appliances, you perform a backup.

After adding nodes to the cluster:

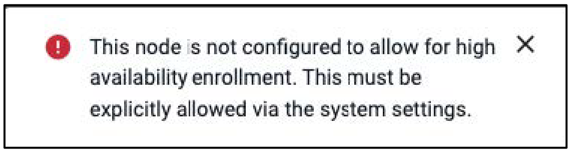
- After adding a Standby node to the cluster, it is strongly recommended to allow the cluster to synchronize all database tables before performing any activities on the cluster, such as adding another node or initiating a failover. When adding a node, the Add Node dialogue box will complete in roughly 30-40 seconds; however, the synchronization of the database can take 3-5 minutes depending on varying factors such as the number of targets, users, and events.
- The Management – High Availability screen in the web UI will show the current state of each node in the cluster and an overall health indicator at the top. The health indicator reflects the replication status of the cluster. Nodes that display a green (healthy) status are undergoing the replication process normally. Nodes that display a red (critical) status are not undergoing replication. A yellow (warning) status can indicate either of the following:
    - The node is in Maintenance Mode for firmware upgrade or other maintenance activities that require the node to be out of lineup for failover.
    - The node is experiencing replication issues that need to be resolved for replication to be restored and the node to become eligible for promotion to Primary in the event of failover.
- After the initial synchronization, all database changes are synchronized immediately to all nodes, except for the following:
    - SNMP target changes (newly discovered or deleted) are synchronized on a regular seven-minute interval.
    - Service Processors (newly discovered or deleted) are synchronized on a regular five-minute interval.

  Therefore, if the Primary node goes down immediately after discovering an SNMP or Service Processor (SP) device, then the SNMP or SP device may not be able to synchronize properly. In this case, the device that was unable to synchronize must be re-discovered on the new Primary node.
- An error will occur if you try to perform configuration changes to devices, users, credential profiles, and other related activities. This error occurs because the permissions to perform such operations are disabled on Standby nodes. Any target or node configuration changes (except for HA Maintenance Mode) should only be performed on the current Primary node, and the changes will be replicated to the Standby nodes. However, certain recovery operations can be performed on Standby nodes, which are discussed more in-depth in the Troubleshooting Examples section of this document.

Setup Failure Conditions and Resolutions

There have been reported instances where the operation of adding a node fails. The operation could have failed due to either of the two following circumstances. See the resolution for these issues below.

- After entering the node IP address and credentials, you receive the following error:



This node is not configured to allow for high availability enrollment. This must be explicitly allowed via the system settings.

  - To resolve this issue, perform these steps on both the Primary and the node that you are adding: Go to the *Administration – System Settings – High Availability* page. Under the High Availability section, ensure the High Availability and Manual Role Control settings are enabled (ON) and click *Save*, if the settings have not already been saved.

- Under rare conditions (more prevalent on virtual appliances separated by slower networks), a node may fail to be added and leave the new node in a state where the user cannot log into the node while it is still in Standalone mode.

  - To resolve this issue on a hardware appliance: Perform a factory reset. Then, update to the expected firmware version and restore the backup image you made as recommended in the Setup Things to Know section of this document.

  - To resolve this issue on a virtual appliance: Restore the appliance from the snapshot you took as recommended in the Setup Things to Know section of this document.

Licensing

NOTE: For information on obtaining and installing the new format license keys, refer to the Vertiv™ Avocent® MP1000 Management Platform User Guide. Prior release licenses (legacy licenses) will remain valid and compatible until they expire or are removed.

Starting at firmware version 3.69.6, the Vertiv™ Avocent® DSView™ solution will support a new licensing system. Legacy licenses will still be honored, but moving forward, the product family will be using the new Thales-based licensing system. One of the significant changes of this transition is that HA cluster licenses only need to be entered on the Primary node, rather than each node in the cluster. No licenses need to be installed on Standby nodes. The total number of HA licenses needed is determined by the total number of nodes within a cluster.

The following table describes the guidelines for legacy licenses.

Table 3.1 Legacy Licensing Guidelines

CLUSTER SIZE	PRIMARY LICENSES	NODE LICENSES
1 Primary, 1 Standby	1 Base license 1 or more Target licenses 1 HA license	1 HA license
1 Primary, 2 Standby	1 Base license 1 or more Target licenses 1 HA license	1 HA license per node, installed on each node

The following table describes the guidelines for new licenses going forward.

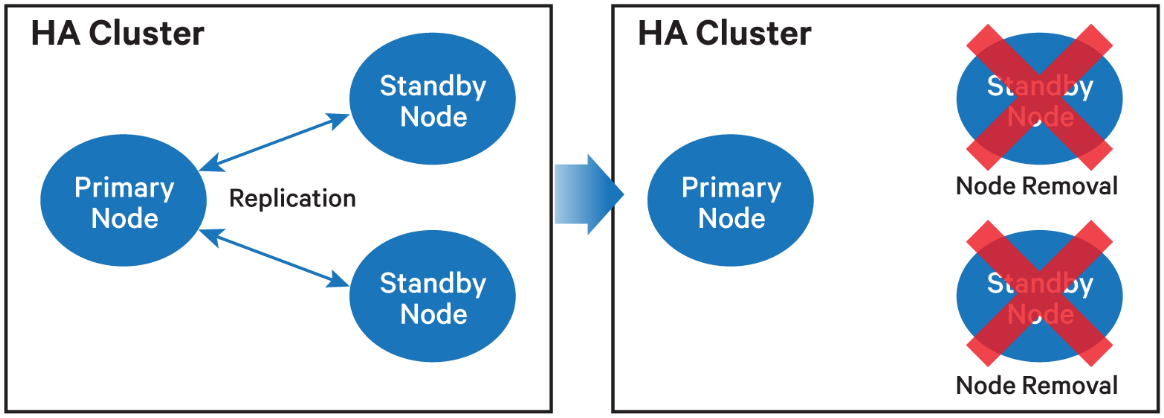
Table 3.2 New Licensing Guidelines

CLUSTER SIZE	PRIMARY LICENSES	NODE LICENSES
1 Primary, 1 Standby	1 Base license 1 or more Target licenses 1 HA license per node (2 total), installed on the Primary	None
1 Primary, 2 Standby	1 Base license 1 or more Target licenses 1 HA license per node (3 total), installed on the Primary	None

4. Breaking Down an HA Cluster

Breaking down an HA cluster is intended to remove one or more nodes from an HA cluster with the intent of re-purposing the node to a non-HA role or for rebuilding the cluster. The following figure shows the cluster breakdown process. Refer to the procedure in this section to break down an HA cluster.

Figure 4.1 Breaking Down an HA Cluster



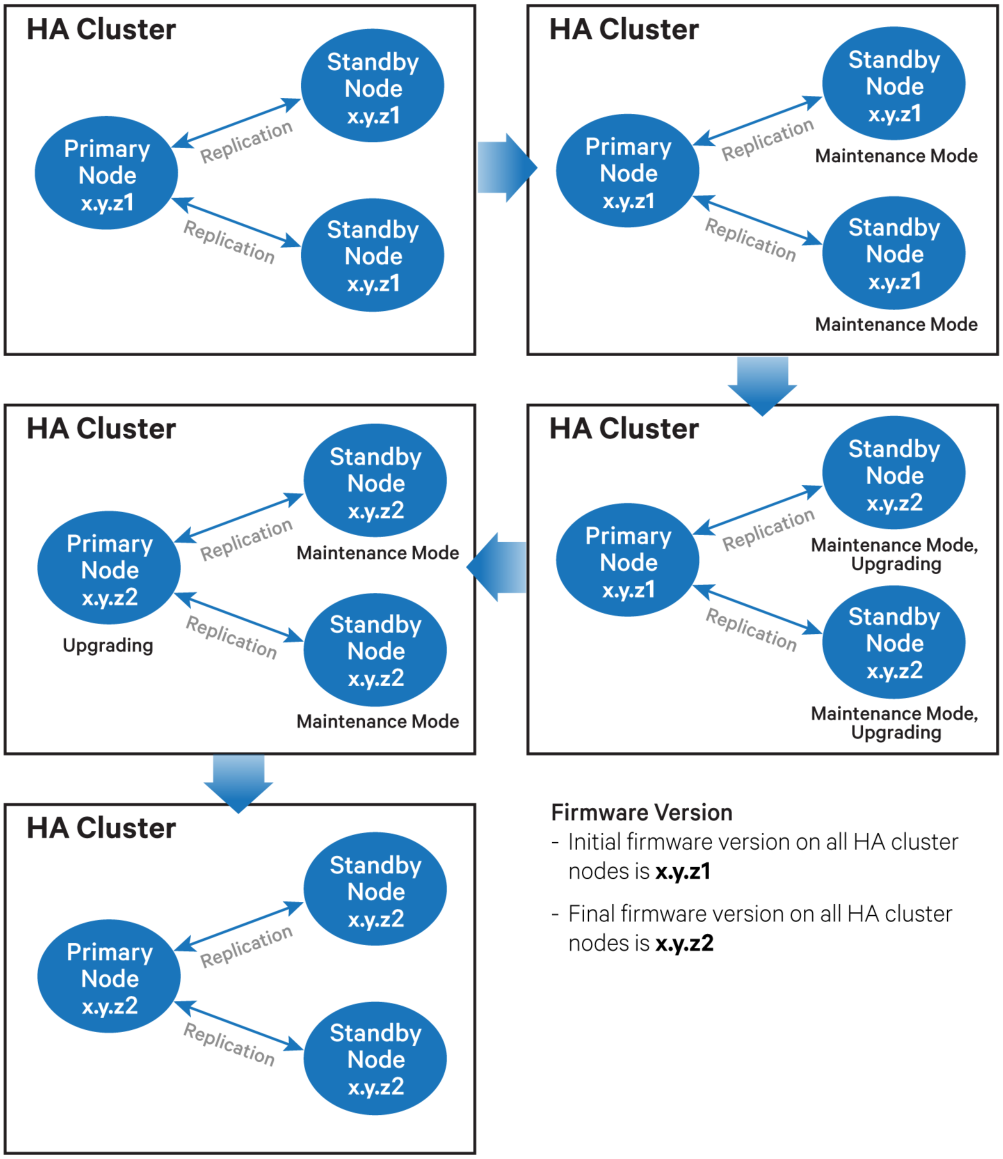
To break down an HA cluster:

1. From the Primary node, go to the *Management – High Availability* screen.
2. Starting with an existing Standby node, click the vertical ellipsis on the right-hand side of the row and click *Remove from cluster*. The node should disappear from the list of nodes on the screen. Wait approximately two minutes before continuing to the next step to ensure the node has been fully removed. During this time, the database will be cleared from the node and all user login information will be removed. The default admin password will be re-established, and the node will return to factory settings.
3. The cluster status as shown on the Management – High Availability screen should remain green for the remaining nodes.
4. Repeat the same procedure for the next Standby node.

5. Upgrading Firmware on an HA Cluster

Upgrading firmware on an HA cluster requires you to first upgrade the firmware on the Standby node(s), and then upgrade the firmware on the Primary node. The following figure shows the firmware upgrade process. Refer to the procedure in this section to upgrade the firmware on a cluster.

Figure 5.1 Upgrading Firmware on an HA Cluster



Firmware Version

- Initial firmware version on all HA cluster nodes is **x.y.z1**
- Final firmware version on all HA cluster nodes is **x.y.z2**

To upgrade the firmware on a cluster:

1. Ensure the target firmware version is the same for all nodes.
2. Put all Standby nodes into Maintenance Mode.
    a. From the *Management – High Availability* screen, click the vertical ellipsis on the right-hand side of the row for the Standby node.
    b. Click *Set to Maintenance*. Wait until the status shows that the node has transitioned.
    c. Repeat the previous steps for any other Standby nodes in the cluster, if applicable.
3. Now that you have placed the Standby nodes into Maintenance Mode, log into each of the Standby nodes and perform the firmware update.
4. Perform the firmware update on the Primary node. The Primary does not need to be in Maintenance Mode; however, the Standby nodes should remain in Maintenance Mode while the Primary node's firmware updates.

5. Return to the *Management – High Availability* screen and verify that the Primary is listed and the expected number of Standby nodes are listed and in Maintenance Mode.

6. Move the Standby nodes out of Maintenance Mode.

    a. From the Primary node, click the vertical ellipsis on the right-hand side of the row for the Standby node.

    b. Click *Set to Standby*. Wait a few minutes before proceeding.

    c. Repeat the previous steps for any other Standby nodes in the cluster, if applicable.

7. After a few minutes, verify that the status on the High Availability screen shows all green (healthy). If after three minutes the state of the cluster shows yellow (warning) or red (critical), then proceed to the Troubleshooting Example section of this document.

6. Using HA with Load Balancers

To interface a load balancer with a High Availability system where you have a Primary and a hot Standby node, you must configure the load balancer to intelligently route requests based on the roles of the nodes.

Detecting the Primary Node

1. Health Check Configuration:
    a. Configure the load balancer to perform regular health checks on the `/api/v1/` endpoint of all nodes.
    b. Ensure the health check can parse the JSON response and, specifically, the `ha_role` key.
2. Role Identification:
    a. The load balancer needs to distinguish between the Primary node and the hot Standby node by examining the `ha_role` key.
    b. When the value is "`MODE_SET_PRIMARY`", that node should be set as the primary target for balancing the load.
    c. The node(s) with "`MODE_SET_STANDBY`" should be regarded as the Standby node and no traffic directed to these.
3. Routing Logic:
    a. Configure the routing logic such that all incoming requests are directed to the node identified as "`MODE_SET_PRIMARY`".
    b. Manage failover by switching the request direction to the node that becomes the new primary if the current primary fails or its role changes.
4. Failover Handling:
    a. Implement automatic failover by using the results of the health check to switch primary nodes. This could involve setting up a monitoring alert or action in the load balancer that changes the primary target when it detects the role change.

Setting the Primary Node

To set a new Primary node via the load balancer, you must automate the process of sending a POST request to the cluster's nodes to change the primary role:

1. Determine the target node.
    a. Based on health checks and role detection through the `GET /api/v1/` endpoint, identify the Standby node that should become the new Primary. It should return the status of "ha_role": "MODE_SET_STANDBY". Typically, this can be done by the load balancer's monitoring mechanism that detects a failure or a role change requirement in the current primary.
2. Send the role change request.
    a. Use a script or load balancer's built-in functionality to send a `PATCH` request to the selected node's `/api/v1/haNodes/<node_id>` endpoint.
3. Request payload.
    a. The payload for the `POST` request should explicitly set the `ha_role` to `MODE_SET_PRIMARY`.
4. Automate and verify.
    a. Automate this process using a tool or script that the load balancer can execute when necessary. Ensure this script has the necessary permissions and network access.
    b. After sending the request, verify the success of the role change by checking the `GET /api/v1/` endpoint to confirm that the ha_role has updated to `MODE_SET_PRIMARY`.

7. Troubleshooting Examples

Table 7.1 HA Troubleshooting Examples

ISSUE	RESOLUTION	STEPS TO RESOLVE
Both the Primary and Standby nodes report they are in "standby" mode, causing potential management issues.	Manually direct the Primary node to assume the "primary" role and restore normal operations.	1. Verify the current states. a. Use `GET /api/v1/` on both nodes to confirm that both report `{"ha_role": "MODE_SET_STANDBY"}`. This would indicate that both nodes are in standby mode. 2. Check the network and system health. a. Ensure both nodes are healthy and that network connectivity between the nodes and client systems is functioning. b. Verify there are no ongoing issues with the underlying server infrastructure. 3. Identify the potential Primary node. a. Decide which node should be the primary based on past configurations, stability, or other operational criteria. 4. Manually set the Primary node. a. In the web UI, go to the *Management – High Availability* screen and set the desired node to Primary via the vertical ellipsis on the right-hand side of the row. b. In the API, on the node identified as the potential primary: i. Execute a `POST` request to `/api/v1/haNodes/<nodeid>` to change its role to `MODE_SET_PRIMARY`. To get the nodeid, refer to API Call Examples.
Two nodes report as Primary (for longer than a few minutes after transitioning the designated Primary node).	Manually force the node which should be the Standby to transition to standby mode.	1. Log into the CLI of the node which should be in standby mode. 2. Go to the diagnostics menu and select the Restart Service option. 3. Enter the number associated with the system-management service. 4. When this service comes back up, it will check the cluster state and determine that there is an existing Primary node. At this point, it will designate the current node as a Standby node.
The Standby node will not remain outside of Maintenance Mode. This issue suggests that there may be services experiencing issues.	Manually remove the node from the cluster, diagnose, and add back to the cluster.	1. From the web UI, select the node that is experiencing issues and click the vertical ellipsis on the right-hand side of the row. 2. Click *Remove from cluster*. Allow the node to remove itself from the cluster. If the node is unable to receive the command, perform the same operation from the node itself instead. 3. Diagnose the node by looking for the following: • Full disk • Services exiting and restarting • Other service issues, but these would be determined based on data from the logs 4. Once resolved, add the node back to the cluster as described in the Vertiv™ Avocent® MP1000 Management Platform User Guide.

8. API Call Examples

This section provides API call examples to perform basic HA procedures. These examples assume knowledge of CURL commands and jQuery.

NOTE: The following examples use insecure connections. The `--insecure` flag in `curl` is used to allow connections to SSL sites without certificates being validated. When you use this flag, `curl` will not verify the authenticity of the SSL certificate presented by the server, which means it will not check if the certificate is self-signed or if it matches the requested hostname.

Example 1: Get JWT Token

This API call would be part of a script that first retrieves the JWT for use when retrieving the node ID. This example uses a host of 10.207.15.180.

```
# get JWT
JWT=$(curl --location --insecure --globoff "https://10.207.15.180/api/v1/userSessions" \
    --header 'Content-Type: application/json' \
    --data '{
    "username": "'"$USER"'",
    "password": "'"$PASS"'"
}' | jq -r '.jwt')
```

Example 2: Get HA Cluster Node ID

This API call returns the UUID of the node that can be used to call to set the Primary node.

```
  NODE_ID=$(curl --location --insecure --globoff GET "https://10.207.15.180/api/v1/clusters" \

    --header 'Content-Type: application/json' \

    --header "Authorization: Bearer $JWT"    \

    | jq '.clusters[] |.nodes[] | select(.node_ipv4 == "'"10.207.15.180"'") |.node_id' | tr -d '"')
```

Example 3: Set Primary Node

This API call uses the $NODE_ID, which is the UUID (NODE_ID) returned in Example 2 above.

```
curl --location --insecure --globoff POST "https://10.207.15.180/api/v1/haNodes/$NODE_ID/ha-mode" \

        --header 'Content-Type: application/json' \

        --header "Authorization: Bearer $jwt" \

        --data '{

            "set_mode": "MODE_SET_PRIMARY"

        }'
```