Long deployment of a Managed Instance

Last updated by | Vitor Tomaz | Nov 16, 2022 at 12:58 PM PST

Contents

Tags: Long creation, long deployment, long provision, provisioning, long time, stuck, create virtual cluster

Most managed instances are deployed within 6 hours (~95%).

- After 6 hours alerts are fired internally for the PG and an ICM is generated.

Internal SLO to provision Managed Instance is 24 hours.

- This is the current estimated goal for the PG.

Note: The information above is currently not publicly available and no official SLA is offered, because of non-deterministic behavior in provisioning.

Step 1) Check Managed Instance deployment request status in CMS

A. If there are no results here
   - Then most likely the deployment request failed
     - To confirm, check failed deployments: [Getting Managed Instance deployment errors in Kusto](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Getting-Managed-Instance-deployment-errors-in-Kusto)
   - Or perhaps the managed instance is already created
     - To confirm, check: [Step 3) Detect stuck Managed Instance deployments in CMS](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Detect-stuck-Managed-Instance-deployments-in-CMS)
     - If there is a result with the *[request_state]* value '*Ready*'
     - Then notify the customer that the MI deployment is complete

B. If there is a result with *[request_total_duration_hours]* less than 6.5h
   - Then notify the customer that this is expected and that they should wait for some additional time

C. Otherwise, proceed with [(Step 2)](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Long-deployment-of-a-Managed-Instance)

Step 2) It is possible that Virtual Cluster is being provisioned for the initial Managed Instance.

Check for the potential provisioning failures with the following CMS query:

Detect Managed Instance provisioning issues in CMS

A. If there are no results here
   a. Then proceed with [(Step 3)](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Long-deployment-of-a-Managed-Instance)
B. If *[virtual_cluster_create_duration_hours]* is less than 6.5h
   - Then this is expected, notify the customer to be patient
C. If *[virtual_cluster_create_duration_hours]* is more than 6.5h
   - Check for an existing ICM (LSI) using the Incidents insight in ASC
   - Create an IcM (CRI)
   - Link both incidents (CRI and LSI)

Step 3) Detect stuck Managed Instance deployments in CMS

A. If there are no results here
   - If there were no results in [(Step 1)](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Long-deployment-of-a-Managed-Instance)
     - Then first confirm with the customer that he initiated the request successfully and can see that it is still ongoing
     - After the confirmation
       - Create an ICM for Managed Instance queue
   - If there were results in [(Step 1)](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Long-deployment-of-a-Managed-Instance) , use those details and
     - Create an ICM for Managed Instance queue
B. If *[managed_instance_state]* is '*Ready*'
   - Then the deployment is complete
     - Notify the customer about completion
C. If there is a result with *[request_total_duration_hours]* less than 6.5h
   - Then this is expected
     - Notify the customer that they should wait for additional time
D. If *[managed_instance_stuck_duration_hours]* is less than 1.5h
   - Then this is expected
     - Notify the customer that they should wait for additional time
E. If *[managed_instance_stuck_duration_hours]* is more than 1.5h
   - Use details from the [(Step 3)](/SQL-Managed-Instance/Troubleshooting-Guides/Create-or-Drop-Resources/Instance/Long-deployment-of-a-Managed-Instance) results and
     - Create an ICM for Managed Instance queue

Troubleshooting :

1. You can check the virtual cluster status from XTS the view is **Global Adhoc CMS Query.xts**

Run the following query :

```
select pc.name ,

pc.subscription_id,

pc.tenant_ring_name,
```

```
pc.state private_cluster_state,

umsr.private_cluster_id,

umsr.target_operation_type,

umsr.create_time,

umsr.concurrency_token,

umsr.is_stable_state,

umsr.is_error_state,

umsr.operation_request_id,

umsr.requested_subscription_id,

umsr.requested_managed_server_name,

umsr.state upsert_managed_server_state

from  upsert_managed_server_requests umsr

left outer join private_clusters pc

on pc.subscription_id = 'Input subscription_id here'
```

2. Http CMS query on region for the MI:

```
select * from private_clusters where subscription_id = 'Input subscription_id here'
```

3. In order to make sure that nothing is blocking the managed instance from being created ask the customer to validate the virtual environment for managed instance by running the attached PowerShell script :

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-configure-vnet-subnet

How good have you found this content?