Redistributing Workload After Active /Active Leader Failover

Redistributing Workload After Active/Active Leader Failover

When one of the nodes(leader/follower) in the ESC Active/Active setup fails, ESC equally distributes the ownership of the loads from the failed node to the available healthy nodes. When the failed node is available again, ESC does not redistribute the configurations automatically from the overloaded nodes to the nodes that are back from failed state to active state.

With a large number of deployments, the nodes with loads are likely to experience performance issues. ESC provides a new REST API that can be called on demand to redistribute the configurations from the overloaded nodes to the available nodes based on ownership transfer and ownership policies.

When the rebalancing API is triggered, ESC performs the following functions to transfer the deployments from one node to another:

- Unloading the deployment with VMs in memory, unset monitor from the current owner
- Assigning new owner based on the ownership policy
- Updating ownership record
- Notifying the new owner

Rebalancing API to Redistribute Ownership

When the failed node is available again in the Active/Active setup, ESC does not redistribute the ownership from the overloaded nodes to the healthy node. You can manually trigger rebalancing API to redistribute the ownerships among available nodes.

Use the following URL to trigger the rebalancing API:

```
http://localhost:8080/ESCManager/internal/ownership/rebalance
```

When rebalancing API is triggered, ESC uses an optimal distribution approach to find the overloaded nodes. ESC identifies the configurations to be transferred and calls remotely to unload the configurations in the current owner and reload remotely in the new owners of the earmarked configurations.

The following example shows how rebalancing is done:

```
URL: http://localhost:8080/ESCManager/internal/ownership/rebalance
Method: PUT
Input: None
{
    "esc1-uuid": --> esc node uuid
    {
        "previous_configs_count": 300, --> number of configs before rebalance
        "requested_unload_count": 100, --> number of configs earmarked for unload
        "accepted_unload_count": 100, --> number of configs actually unloaded
        "current_configs_count": 200 --> number of configs after rebalance
    },
    "esc2-uuid":
    {
        "previous_configs_count": 300,
        "requested_unload_count": 100,
        "accepted_unload_count": 90,
        "current_configs_count": 210
    },
    "esc3-uuid":
    {
        "previous_configs_count": 0, --> new esc node, no configs yet
        "requested_unload_count": 0, --> previous config count is less than optimal count
per esc, so no unload requested
        "accepted_unload_count": 0,
        "current_configs_count": 190--> 100 from esc1 and 90 from esc2 got loaded
    }
}
Exception: {"error": {"error_code":500, "error_message": "Internal Error"}}
```

You can trigger the rebalancing API multiple times until you receive the following message:

```
No configs found for rebalance.
```

You will receive an error code/message when one of the following instances occurs.

- During the ongoing transfer of ownership due to a failover.
- If there are less than two healthy ESC nodes.
- If the pause ownership is enabled.
- If ESC is standalone.