Amazon OpenSearch L2 tooling

This page aims to be a cheat sheet with useful commands that can be ran with L2 level

- Stop Elasticsearch process in a node (requires 2PR)

```
tumbler <region> aes domain dp node es deactivate --domain-identifier=<domain-id> --instance-id=<instance-id> --approver-id=alias
```

For more info on this command: https://api.us-east-1.tumbler.prod.search-services.aws.a2z.com:8443/explorer/aes/domain/dp/node/es/deactivate

- Preactivate Elasticsearch process in a node (requires 2PR)

```
tumbler <region> aes domain dp node es preactivate --domain-identifier=<domain-id> --instance-id=<instance-id> --approver-id=alias
```

For more info on this command: https://api.us-east-1.tumbler.prod.search-services.aws.a2z.com:8443/explorer/aes/domain/dp/node/es/preactivate

- Start Elasticsearch process in a node (requires 2PR)

```
tumbler <region> aes domain dp node es activate --domain-identifier=<domain-id> --instance-id=<instance-id> --approver-id=alias
```

For more info on this command: https://api.us-east-1.tumbler.prod.search-services.aws.a2z.com:8443/explorer/aes/domain/dp/node/es/activate

- Stop Kibana (requires 2PR)

```
tumbler <region> aes domain dp node kibana deactivate -D <domain-d> -i <instance-id>
```

- Start Kibana (requires 2PR)

```
tumbler <region> aes domain dp node kibana activate -D <domain-d> -i <instance-id>
```

- Recover deleted domain (recommended to run by L2 ONLY when cluster stuck on deletion)

```
aescloudopstools <region> domain cp cs-recover-deleted-domain --args '-D <domain-id>' --approver-id alias.
```

- Recover domain

```
aescloudopstools <region> domain cp cs-recover-domain --args '-D <domain-id>' --approver-id alias
```

- Force CDI (requires 2PR) **(unfortunately, permission revoked)**

```
aescloudopstools $REGION domain cp cs-create-domain-instance --args '-D $DOM'--approver-id alias
```

***- Increase DI count to 3 (requires 2PR)* (unfortunately, permission revoked)**

`aescloudopstools $REGION domain cp cs-create-domain-instance --args '-D $DOM --max-allowed-di-count 3 --ticket-id <TT id>' --approver-id alias`

***- Replace node (requires 2PR)* (unfortunately, permission revoked)**

`tumbler <region> aes domain dp node terminate-instance --skip-diagnose -D <domain-d> -i <instance-id> --approver-id alias`

- Disable hourly snapshots (need 2PR from L3) and modify daily snapshot time.

In order for us to access these commands, we would need to follow a contingent security mechanism which is implemented by creating a ticket to the below mentioned CTI:
C: AWS
T: OpenSearch Tool Usage
I: Customer Issue
To ease this activity of creating a ticket and following the specified format, we have created a ticket template which can be found below:
https://t.corp.amazon.com/create/templates/76128b53-02ee-42b2-9e23-ed39e55af773
The Command usage have been described below:
Command to Disable hourly snapshots:
This command will disable hourly snapshots but keep daily snapshots for a set of Elasticsearch domains. However, this command is a bit primitive and fully erases the previous configuration. For this reason, you will need to perform TWO steps when disabling the hourly snapshots for an Elasticsearch domain.
1. First step: Retrieving **WHAT OPENSEARCH DOMAINS ALREADY DO HAVE DISABLED SNAPSHOTS** for the region in which the domain you want to configure is located:
`aescloudopstools <REGION> domain cp cs-whitelist-client-snapshot --feature es.snapshot.frequent_snapshot_disabled_domains --args "--target-stack swift-<REGION>-prod --target-client-id <ACCOUNT-ID> -r prod --region <REGION> --action list --ticket-id <Ticket-Id>"`
- Example output:

```
INFO Executing (...)
{
    "swift-eu-west-1-prod": {
        "123456789012": {
            "es.snapshot.frequent_snapshot_disabled_domains": "DOMAIN_1,DOMAIN_2",
            "updated_by": "(...)",
            "updated_utc": "(...)"
        }
    }
}SUCCESS
```

Please note that the command returned a list WITH THE DOMAINS THAT DO NOT HAVE HOURLY SNAPSHOTS ENABLED FOR THAT REGION AND AWS ACCOUNT ID.
NOTE: If you want to re-enable hourly snapshots make sure to swap the "—action" parameter to "disable".
2. Second step: adding **THE FULL LIST OF OPENSEARCH DOMAINS THAT SHOULD HAVE NO HOURLY SNAPSHOTS FOR THAT REGION**, using a comma separated list (after run below command, you can re-run above command to double check).
`aescloudopstools <REGION> domain cp cs-whitelist-client-snapshot --feature es.snapshot.frequent_snapshot_disabled_domains --args "--target-stack swift-<REGION>-`

`prod --target-client-id <ACCOUNT-ID> -r prod --region <REGION> --feature_value <DomainName> --action enable --ticket-id <Ticket-Id>"`
Example:
`aescloudopstools dub domain cp cs-whitelist-client-snapshot --feature es.snapshot.frequent_snapshot_disabled_domains --args "--target-stack swift-eu-west-1-prod --target-client-id 123456789012 -r prod --region eu-west-1 --feature_value es_domain_1,es_domain_2,es_domain_avocado --action enable --ticket-id V1234567890"`
*Please note, for `swift-<REGION>-prod` information here, the region need to be writen as "us-east-1" format but not "IAD".
*Please note, how the domains have been added as a list. This change might take a while to kick in, so the next snapshot(s) might be taken after the change was pushed.
***Please note**: The `--ticket-id` in above command is for the new TT you just created and while you are running the command, you need to make sure the TT is in Open status.
To get more info about the different nomenclatures for regions you can visit [4]: https://w.amazon.com/bin/view/AWS/Regions/
Command to modify snapshot hour:
`aescloudopstools run <REGION> domain cp cs-override-es-domain-config-snapshot-hour --args "-D <DI> --ticket-id <Ticket-Id>" --snapshot-hour <Value>`
***Please note**: The —ticket-id in above command is for the new TT you just created and while you are running the command, you need to make sure the TT is in Open status.

- List snapshot disabled domains (same as above, need to create TT to get temporary permission)

`aescloudopstools $REGION domain cp cs-whitelist-client-snapshot --feature es.snapshot.frequent_snapshot_disabled_domains --args "--target-stack swift-$REGION-prod --target-client-id <aws-account-id> -r prod --region $REGION --action list"`

- Change the update-recovery-settings (requires 2PR)

`tumbler $REGION aes domain dp node es curl cluster update-recovery-settings -D $DI --concurrent-recoveries 10 --concurrent-rebalance 10 --indices-recoveries 60mb --primary-recoveries 10`
Note: Please change the int number based on your use case
concurrent-recoveries = cluster.routing.allocation.node_concurrent_recoveries
indices-recoveries = indices.recovery.max_bytes_per_sec
primary_recoveries = cluster.routing.allocation.node_initial_primaries_recoveries
concurrent-rebalance = cluster.routing.allocation.cluster_concurrent_rebalance

- Mandatory Software Release (requires 2PR) - Apply release on a domain with scheduled time

`aescloudopstools run $region domain cp sdpds-apply-release \ --args="-a apply -D $domainIdent --request-source OPERATOR \ --scheduled-time $scheduleTimeInMillis --force-schedule --second-user $approverAlias" --approver-id $approverAlias`

- Cloudwatch agent restart (requires 2PR)

The AWS Elasticsearch/Opensearch managed service has some mechanisms to upkeep the cloudwatch agent health. However, there are some edge cases where the agent needs to be restarted manually. In order to do so, you will need to use the tumbler CLI in the Ops Host that you configured in the introduction. You can find more info on tumbler in [5]. There are two ways to restart the Cloudwatch agent: the automated and the manual

When Cloudwatch isn't reporting metrics for a node, use the automated approach. It will first check if the Cloudwatch agent is running, and it will start it only if the Cloudwatch agent is not generating any logs. As explained in the CWAgentUnhealthy

SOP[6], there are three steps to perform this operation:

- **Step 1:** `tumbler <region> aes domain cw restart -D <domain-id>` command will check count of cw.agent.log* files in /apollo/env/swift-*-LogPusherAMI-ES2/var/output/logs across all the nodes from a domain. **If count is greater than 0**, it will return status of that **CloudWatch agent for that node** as healthy else unhealthy.
- **Step 2:** To Restart CloudWatch agent (kill CloudWatch process id) operator need to pass `--submit` flag to above command. `tumbler <region> aes domain cw restart -D <domain-id> --submit` will kill CloudWatch agent processes across domain if CloudWatch agent is unhealthy.
- **Step 3:** To verify if CloudWatch restarted successfully wait for a minute and then check status by running command (`tumbler <region> aes domain cw restart -D <domain-id>`) mentioned in Step 1.

In some edge cases, it might be required to restart the Cloudwatch agent manually. For example, https://issues.amazon.com/BeagleRock-965. In such cases, **please synchronize with the customers so they do not get surprised/alarmed if some specific nodes stops sending metrics**.

- **Step 1:** Stop the Cloudwatch agent in the problematic node tumbler `tumbler <region> aes domain cw stop -D <domain-id> -i <instance_id>`. You can find instance IDs in Domain Resources.
- **Step 2:** This command takes a while to run. Please wait until it finishes and confirm from the output that the execution was successful.
- **Step 3:** Make sure that the agent is active for that very same instance id. `tumbler <region> aes domain cw start -D <domain-id> -i <instance_id>`. The supervisor process should bring up the Cloudwatch agent as soon as it stops working, but always double check by running the command specified in this step.

- Deactivate kibana, reindex-and-delete .kibana index, activate kibana (with 2PR)

```
tumbler $REGION aes domain dp deactivate-kibana -D $DOM
tumbler $REGION aes domain dp node kibana reindex-and-delete -D $DOM --source-index-name <.kibana_index_name>
tumbler $REGION aes domain dp activate-kibana -D $DOM
```