CP Commands

***Please note, we are able to see multiple request from engineers related to performing some PUT and POST operation (curl put/post permission has been revoked by service team). However, we would like to inform that some of these command can be executed using `‘aes domain dp node es safe-curl post’` and `‘aes domain dp node es curl cluster settings’`**

```
PUT setting works with “aes domain dp node es curl cluster settings”:
--url "_cluster/settings" --data '{"transient" : {"indices.breaker.total.limit" : "99%"}}'
--url "_cluster/settings" --data '{ "transient" : {"aes.jetty.admission_control.global_jvmmp.limit": "99%"}}'
--url '/_cluster/settings' --data '{"persistent" : {"cluster.blocks.read_only" : false}}'
POST snapshot restore works with “aes domain dp node es safe-curl post”:
--url "_snapshot/cs-automated-enc/<snapshot_id>/_restore" -D $DOM --data '{"indices": "index_1,index_n"}'
```

```
# - low | check di vs dm
aescloudopstools $REGION domain cp cs-domain-manager-status —args "-D $DI"
aescloudopstools $REGION domain cp cs-domain-manager-status —args "-D $DI -ia —all“
# - moderate | retry failed activities on dm
aescloudopstools $REGION domain cp cs-retry-failed-activities —args " -D $DI"
# - low | check vfi
aescloudopstools $REGION domain cp cs-list —args "-D $DI —instances -k rev“
aescloudopstools run $REGION domain cp cs-manage-software-release-config —args " —action list" | awk '{print $1,$2}'
# - low | print snapshots
aescloudopstools $REGION domain cp cs-manage-es-snapshots-read-only —args "-D $DI"
# - low | print shard allocation
aescloudopstools $REGION domain cp es-print-shards-info —args "-D $DI"
# - low | print instances dataNodes
tumbler $REGION aes domain dp get-domain-resources -D $DI | \
jq -r '.ec2Instances[] | select(.nodeType=="dataNode") | .instanceId' | \
awk '{print $1}'
# - low | print instances masterNodes
tumbler $REGION aes domain dp get-domain-resources -D $DI | \
jq -r '.ec2Instances[] | select(.nodeType=="masterNode") | .instanceId' | \
awk '{print $1}'
# - low | max allowed di count
aescloudopstools $REGION generic run —cmd-name "cs-create-domain-instance" —cmd-args "-D $DI —max-allowed-di-count 3“
# - high | check and fix quorum loss
aescloudopstools run $REGION generic run —cmd-name detect-quorum-loss.py —cmd-args "-D $DI"
aescloudopstools $REGION generic run —cmd-name es-master-script.py —cmd-args " -D $DI -A FixQuorumLoss —force" —timeout 3600
# - low broken | list whitelist hourly snapshots
aescloudopstools $REGION domain cp cs-whitelist-client-snapshot —feature es.snapshot.frequent_snapshot_disabled_domains \
--args "—target-stack swift-$REGION-prod —target-client-id -r prod —region $REGION —action list“
# - 2021/04/28 - Commands above for list and disable hourly snapshots is broken since python3
# - workaround is to mwinit —breakglass -o and connect in ssh into a Ressource Manager
./cs-whitelist-client —feature es.snapshot.frequent_snapshot_disabled_domains —target-stack swift-<region>-prod —target-client-id <aws-account-id> -r prod —region <region> —action list
```

# - moderate broken | **disable hourly snapshots**

```
aescloudopstools $REGION domain cp cs-whitelist-client-snapshot —feature es.snapshot.frequent_snapshot_disabled_domains \
--args "—target-stack swift-$REGION-prod —target-client-id 755424219130 -r prod —region $REGION —feature_value —action enable“
# - 2021/04/28 - Commands above for list and disable hourly snapshots is broken since python3
# - workaround is to mwinit —breakglass -o and connect in ssh into a Ressource Manager

./cs-whitelist-client —feature es.snapshot.frequent_snapshot_disabled_domains —target-stack swift-<region>-prod —target-client-id <aws-account-id> -r prod —region <region> —feature_value <domain-name> —action enable
```

# - moderate | **disable daily snapshots**

```
aescloudopstools $REGION generic run —cmd-name cs-override-es-domain-config —cmd-args "-D $DI —snapshot-hour -1“
```

# - moderate | **disable daily snapshots**

```
aescloudopstools $REGION generic run —cmd-name cs-override-es-domain-config —cmd-args "-D $DI —snapshot-hour XX“
```

# - moderate | **delete stuck snapshot**

```
tumbler $REGION aes domain dp node es curl delete -D $DI -u "_snapshot/cs-automated-er
```

# - high 2p | **delete di**

```
aescloudopstools run $REGION domain cp cs-delete-domain-instance —args "-D $DI -i <ident>"
```

DP Commands

_CLUSTER/SETTINGS COMMANDS

# - low | **_cluster/settings all with defaults**

```
tumbler $REGION aes domain dp node es curl get -D $DI -u '_cluster/settings?include_defaults=true&pretty'
```

# - low | **check routing allocation require**

```
tumbler $REGION aes domain dp node es curl get -D $DI -u '*/_settings?pretty' | \
grep -A 10 -B 10 "routing" | \
grep -A 10 -B 10 "allocation" | \
grep -A 10 -B 10 "require"
```

# - low | **checking voting_config_exclusion**

```
tumbler $REGION aes domain dp node es curl get -D $DI -u '_cluster/settings?include_defaults=true'
tumbler $REGION aes domain dp node es curl get -D $DI -u '_cluster/state?filter_path=metadata.cluster_coordination.voting_config_exclusions&pretty' \
--data '{"metadata" : {"cluster_coordination" : {"voting_config_exclusions" : [ ]}}}'
```

# - moderate | **speed shards alloc**

```
tumbler $REGION aes domain dp node es curl put -D $DI -u "_cluster/settings?pretty" \
--data '{
"transient": {
"indices.recovery.max_concurrent_file_chunks": 5,
"cluster.routing.allocation.node_concurrent_recoveries": 5,
"indices.recovery.max_bytes_per_sec": "500mb"
}}'
```

# - moderate | **_cluster/settings search.max_open_scroll_context**

```
tumbler $REGION aes domain dp node es curl get -D $DI —url "_cluster/settings?pretty&include_defaults&flat_settings=true" \
| grep max_open_scroll_context
```

```
tumbler $REGION aes domain dp node es curl put -D $DI —url "/_cluster/settings?flat_settings=true" \
--data '{"persistent" : {"search.max_open_scroll_context": 3000},"transient": {"search.max_open_scroll_context": 3000}}'
```

# - moderate | _cluster/settings max_shards_per_node

```
tumbler $REGION aes domain dp node es curl put -D $DI -u '_cluster/settings' \
--data '{"transient": {"cluster.max_shards_per_node":1010}}'
```

# - moderate | remove 7.x read-only

```
tumbler $REGION aes domain dp node es curl put -D $DI -u '_cluster/settings?pretty' \
--data '{"persistent": {"cluster.blocks.read_only": false}}'
```

# - high | es version < 7.x change minimum_master_nodes

```
tumbler $REGION aes domain dp node es curl put -D $DI -u '_cluster/settings?pretty' \
--data '{"transient": {"discovery.zen.minimum_master_nodes": 1}}'
```

# - high | es version > 7.x discovery.zen.minimum_master_nodes

```
tumbler $REGION aes domain dp node es curl post -D $DI -u '_cluster/voting_config_exclusions?node_ids=' —data '{}'
tumbler $REGION aes domain dp node es curl get -D $DI -u '_cluster/state?filter_path=metadata.cluster_coordination'
tumbler $REGION aes domain dp node es curl delete -D $DI -u '_cluster/voting_config_exclusions?wait_for_removal=false'
```

# - high | set routing allocation di exclusion (old command, permission revoked)

```
tumbler $REGION aes domain dp node es curl put -u '_cluster/settings?pretty' -D $DI \
--data '{"transient": {"cluster.routing.allocation.exclude.di_number": ""}}'
tumbler $REGION aes domain dp node es curl put -u '_cluster/settings?pretty' -D $DI \
--data '{"transient": {"cluster.routing.allocation.exclude.di_number": "7,6"}}'
```

# - high | set routing allocation di exclusion (new available command, need 2PR from Service Team)

```
tumbler <REGION> aes domain dp node cluster-routing-allocation —domain-identifier <DOMAIN-IDENTIFIER> —di <DI-NUMBER> —instance-id <INSTANCE-ID> —approver-id <APPROVER-ID>
```

This wrapper is a command which is used to exclude a DI number in a cluster, it come with below Guardrails:

- Cluster has to be in Blue/Green deployment
- Total number if DI in the cluster should be at least 2
- Latest DI should not be excluded

Please note, due to existing bug, you need to provide a healthy instance id in this command so that you can run this command successfully. Service team engineer has awarded this bug and will fix this later.

# - high | remove routing allocation zone exclusion

```
tumbler $REGION aes domain dp node es curl put -u '_cluster/settings?pretty' -D $DI \
--data '{"transient": {"cluster.routing.allocation.awareness.attributes": ""}}'

tumbler $REGION aes domain dp node es curl put -u '_cluster/settings?pretty' -D $DI \
--data '{"transient": {"cluster.routing.allocation.awareness.attributes": "zone"}}'
```

HEALTH & ALLOCATION COMMANDS

# - LOW | _CLUSTER/HEALTH

```
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_CLUSTER/HEALTH?PRETTY'
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_CLUSTER/HEALTH?PRETTY&LOCAL=TRUE'
```

# - low | _cluster/allocation

```
tumbler $REGION aes domain dp node es curl get -D $DI -u '_cluster/allocation/explain?pretty'
tumbler $REGION aes domain dp node es curl post -D $DI -u '_cluster/allocation/explain?pretty' \
--data '{"index": ".opendistro-ism-config","shard": 0,"primary": true}
```

# - low | retry failed allocation

```
tumbler run $REGION aes domain dp node es curl cluster reroute-retry-failed -D $DI —submit
tumbler $REGION aes domain dp node es curl post -D $DI -u "/_cluster/reroute?retry_failed=true" —data '{}'
```

# - low | change disk watermark settings

```
tumbler $REGION aes domain dp node es curl put -D $DI —url "localhost:9200/_cluster/settings?" —data '{ "persistent" : {
"cluster.routing.allocation.disk.watermark.low" : "25gb", "cluster.routing.allocation.disk.watermark.high": "15gb",
"cluster.routing.allocation.disk.watermark.flood_stage" : "5gb"}}'
```

SNAPSHOT RELATED COMMANDS

# - LOW | CHECK SNAPSHOT

```
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_CLUSTER/STATE?PRETTY' | GREP
'SNAPSHOT_DELETIONS' -A 5
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_TASKS?PRETTY'
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_SNAPSHOT/CS-AUTOMATED/_ALL?
PRETTY&ERROR_TRACE=TRUECURL'
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_CLUSTER/STATE?PRETTY' | GREP -A 1 '"STATE"
: "FAILED"' | GREP NODE | SORT | UNIQ -C
TUMBLER $REGION AES DOMAIN DP NODE ES CURL GET -D $DI -U '_CLUSTER/STATE?PRETTY'| GREP -A 1 '"STATE"
: "ABORTED"' | GREP NODE | SORT | UNIQ -C
```

# - moderate | take / delete snapshot

```
tumbler $REGION aes domain dp node es curl put -D $DI -u '_snapshot/cs-automated/' —data '{}'
tumbler $REGION aes domain dp node es curl delete $DI -u '_snapshot/cs-automated-enc/'
```

# - high | list red indices + restore them from snapshots

```
tumbler $REGION aes domain dp node es curl get -D $DI —url '_cat/indices?v&health=red'
aescloudopstools run $REGION generic run —cmd-name="cs-manage-es-snapshots" —cmd-args="-D $DI —action restore —
repository-name $RESPOSITORY —snapshot-name $LAST_SUCCESSFUL_SNAPSHOT —restore-indices $RED_INDICES“
```

# - high | restore-index (with 2PR from Service Team)

```
tumbler run <REGION> aes domain dp node restore-index —domain-identifier <DOMAIN-IDENTIFIER> —source-index-name
<INDEX-NAME> —instance-id <INSTANCE-ID> —approver-id <APPROVER-ID>
```

This wrapper is used to delete an index from a node and restore the index from the latest successful snapshot, it come with below Guardrails:

- If there is node drop
- If the index is an internal index only
- If the index is a red index only
- If the index is present in the latest successful snapshot

Please note, due to existing bug, you need to provide a healthy instance id in this command so that you can run this command successfully. Service team engineer has awarded this bug and will fix this later.

Kibana Commands

# - moderate | kibana fix index

```
tumbler $REGION aes domain dp node es curl post -D $DI -u '_aliases' \
--data '{"actions" : [{ "remove" : { "index" : ".kibana_1", "alias" : ".kibana" } },{ "add" : { "index" : ".kibana_2", "alias" : ".kibana" }
}]}'
tumbler $REGION aes domain dp node es curl post -D $DI -u '_reindex' \
--data '{"source": {"index": ".kibana_1"},"dest": {"index": ".kibana_2"}}'
```

# - low | check kibana api

```
tumbler $REGION aes domain dp node shell -b "curl" -a "-s localhost:9200/_plugin/kibana/app/kibana" -D $DI
tumbler $REGION aes domain dp node shell -b "curl" -a "-s localhost:5601/api/status" -D $DI | jq -r ".status.overall"
```

BOUNCE COMMANDS

# - HIGH | TUMBLER BOUNCE ES

```
TUMBLER $REGION AES DOMAIN DP NODE ES DEACTIVATE -D $DI -I $I
TUMBLER $REGION AES DOMAIN DP NODE ES ACTIVATE -D $DI -I $I
```

# - high | tumbler bounce kibana

```
tumbler $REGION aes domain dp node kibana deactivate -D $DI -i $I
tumbler $REGION aes domain dp node kibana activate -D $DI -i $I
```

# - high | tumbler bounce sdpdagent

```
tumbler $REGION aes domain dp node sdpdagent deactivate-agent -D $DI —pip $IP
tumbler $REGION aes domain dp node sdpdagent activate-agent -D $DI —pip $IP
```

OTHER DP COMMANDS

# - low | shell find file

```
tumbler $REGION aes domain dp node shell -D $DI -b "find" -a " / -name *es_bootstrap*" -i i-0afe4672b284ac30d
```

# - high 2p | < 7.x seed master recovery

```
aescloudopstools run $REGION domain cp cs-control-auto-scaling —args \
"-D $DI —suspend —asg-number -i “
tumbler run $REGION aes domain dp node haproxy deactivate -D $DI -i $I
tumbler $REGION aes domain dp node shell -D $DI -b "curl" -a \
"-s -XPOST localhost:9200/__aes/cluster/bootstrap -d '{\"initial_master_nodes\": [\"IP.IP.IP.IP\"]}' -H \"Content-Type:
application/json\"" -i $I/html
- - - -
tumbler $REGION aes domain dp node shell -D $DI -b "curl" -a \
"-s -XPOST localhost:9200/__aes/shutdown -d '{\"StayDown\": \"false\"}' -H \"Content-Type: application/json\"" -i $I
OR
tumbler $REGION aes domain dp node shell -b 'ps' -a ' -aux' -D $DI -i $I | grep 'bootstrap'
tumbler $REGION aes domain dp node shell -b 'kill' -a '' -D $DI -i $I
- - - -
tumbler aes domain dp node es curl put -u "localhost:9200/_cluster/settings" -D —data '{"persistent":
{"cluster.blocks.read_only":"true"}}'
tumbler run $REGION aes domain dp node haproxy activate -D $DI -i $I
aescloudopstools run $REGION domain cp cs-control-auto-scaling —args "-D $DI —resume —asg-number -i “
tumbler $REGION aes domain dp node es curl get -D $DI -u '_cluster/settings?include_defaults=true&&pretty&&flat_settings' |
grep "aes.jetty.cp_events.upgrade_status"
```

# - low | tail logs

```
tumbler $REGION aes domain dp node shell -D $DI -b "tail" —args "-n 1000 /apollo/env/swift-*-prod-
*Kibana*/var/output/logs/kibana.log" -i $I
tumbler $REGION aes domain dp node shell -D $DI -b "tail" —args "-n 1000 /apollo/env/swift-*-prod-
*Kibana*/var/output/logs/kibana.log"
```

# - low | check pid of es process

```
tumbler $REGION aes domain dp node shell -D $DI -b 'ps' -a ' -wwef' -i $I | grep -e "bootstrap"
```

# - moderate | check pid of cloudwatch and bounce

```
tumbler $REGION aes domain dp node shell -b 'ps' -a ' -aux' -D $DI -i $I | grep CloudWatch | grep -v grep | awk '{print $2}'
tumbler $REGION aes domain dp node shell -b 'kill' -a '' -D $DI -i $I
```

# - moderate | tumbler captures diagnostic dumps (light weight and doesn’t take full heap dumps)

```
tumbler $REGION aes domain dp node es profilers thread -D $DI -i $I && sleep 5 && \
tumbler $REGION aes domain dp node es profilers thread -D $DI -i $I && \
tumbler $REGION aes domain dp node diagnose -D $DI -i $I
```

~~# - moderate | increase the disk space -~~ Avoid using this tool. Modifications made by this tool hinders the inplace updates on customer cluster. Recommend the customer to perform inplace volume update.

~~tumbler $REGION aes domain dp node modify-data-volume -D $DI -i —volume-size~~

# - low | **grepping ES logs**

```
tumbler $REGION aes domain dp node shell -D $DI -b "grep" -a " Exception /apollo/env/swift-*-prod-*ES*/var/output/logs/elasticsearch.log" -i $I
```

# - high | **reboot instance**

```
tumbler —bypass $REGION aes domain dp node reboot-instance "-D $DI —instance-id $I"
```

# - moderate | **closing index (permission revoked)**

```
tumbler $REGION aes domain dp node es curl post -D $DI -u "/_close" —data '{}'
```

# - high | **new closing index (need 2PR from Service Team)**

```
tumbler run <REGION> aes domain dp node close-index —domain-identifier <DOMAIN-IDENTIFIER> —snapshot-id <SNAPSHOT-ID> —source-index-name <INDEX-NAME> —instance-id <INSTANCE-ID> —approver-id <APPROVER-ID>
```

This wrapper is used to close an index, it come with below Guardrails:

- The index is red
- Index must be present in the snapshot whose id has been provided

Please note, due to existing bug, you need to provide a healthy instance id in this command so that you can run this command successfully. Service team engineer has awarded this bug and will fix this later.

# - moderate | **opening index**

```
tumbler $REGION aes domain dp node es curl post -D $DI -u "/_open" —data '{}'
```

# - moderate | **increase number of instances per domain**

```
aescloudopstools run $REGION domain cp cs-update-limits —args "-D $DI —es-max-instance-count "
```

# - low | **check ES version upgrade history**

```
aescloudopstools $REGION domain cp es-print-upgrade-history —args "-D $DI"
```

# - low | **check software version**

```
aescloudopstools $REGION domain cp cs-list —args "-D $DI —instances -p" | grep "SoftwareVersion"
```

# - low | **list VPC resources**

```
aescloudopstools $REGION domain cp es-manage-vpc-interface —args "list-resources -D $DI"
```

# - low | **disable admission control settings**

```
tumbler $REGION aes domain dp node es curl put -D $DI —url "localhost:9200/_cluster/settings?" —data '{ "persistent" : { "aes.jetty.admission_control.global_cpu_usage.enable" : "false" }, "transient" : { "aes.jetty.admission_control.global_cpu_usage.enable" : "false" } }'
```

# - moderate | **recover deleted domain**

```
aescloudopstools $REGION domain cp cs-recover-deleted-domain —args "-D $DI"
```

# - moderate | **change value of server.maxPayloadBytes (for domain version lower than OS 2.19)**

STEP 1: Checking existing max payload

```
for i in tumbler $REGION aes domain dp get-domain-resources -D $DI | jq -r '.ec2Instances[] | select(.nodeType=="dataNode") | .instanceId'
do{
echo $i{
tumbler $REGION aes domain dp node shell -b su -a 'root -c "cat /apollo/env/swift-us-east-1-prod-OSD_2_9AMI-ES2/var/kibana/config/opensearch_dashboards.yml"' -D $DI -i $i | grep maxPayload
done
```

STEP 2: Deactivate kibana

```
yes | tumbler —bypass $REGION aes domain dp deactivate-kibana -D $DI
```

STEP 3: Changed max payload

```
for i in tumbler $REGION aes domain dp get-domain-resources -D $DI | jq -r '.ec2Instances[] | select(.nodeType=="dataNode") | .instanceId'
do
echo $i
```

```
tumbler $REGION aes domain dp node shell -b "su" —args " root -c \"sed -i 's/server.maxPayloadBytes: 5242880/server.maxPayloadBytes: 31457280/' /apollo/env/swift-us-east-1-prod-OSD_2_9AMI-ES2/var/kibana/config/opensearch_dashboards.yml\"" -D $DI —instance-id $i
done
```

STEP 4: Activated kibana nodes

```
yes l tumbler —bypass $REGION aes domain dp activate-kibana -D $DI
```

STEP 5: Checked status of all nodes

```
tumbler $REGION aes domain dp get-domain-resources -D $DI l jq -r '.ec2Instances[] l select(.nodeType=="dataNode") l .instanceId' l \ awk '{print $1}' l xargs -I {} sh -c 'echo {};
tumbler $REGION aes domain dp node shell -b \ "curl" -a "-s localhost:5601/api/status" \ -D $DI -i {} l jq -r ".status.overall.state"'
```

STEP 6: Verified all maxPayloadBytes were set to the requested settings

```
for i in tumbler $REGION aes domain dp get-domain-resources -D $DI | jq -r '.ec2Instances[] | select(.nodeType=="dataNode") | .instanceId'
do
echo $i
tumbler $REGION aes domain dp node shell -b su -a 'root -c "cat /apollo/env/swift-us-east-1-prod-OSD_2_9AMI-ES2/var/kibana/config/opensearch_dashboards.yml"' -D $DI -i $i l grep maxPayload
done
```

***Please note,** for domain version **OS 2.19 and higher**, we implement a customer requested feature which is persisting server.maxpayloadbytes in opensearch_dashboards.yml behind a feature flag for account level. After adding this feature flag, cx are able to request increase the server.maxpayloadbytes and the change will not be wiped out after B/G or node replacement. In this, to update **server.maxPayloadBytes for OS 2.19 and higher**, we need to follow new SOP. However, SupOps engineers do not have permission for new SOP yet, if that is the case, then please contact Service Team Frontend oncall for help.

For more information:

https://quip-amazon.com/O0zTAHCuF0Xl

https://w.amazon.com/bin/view/Search/A9/Swift/Ops/Procedures/Kibana#Hserver.maxPayloadBytes28EnableFeatureFlag2Cstopmanuallyupdatingconfig29

# - moderate l restore domain from snapshot of deleted domain

```
aescloudopstools run generic run —cmd-name es-tombstone-restore-snapshot —cmd-args "—deleted-domain-ident —validate-deleted-domain —second-user= cp-on-call“
```

```
aescloudopstools run generic run —cmd-name es-tombstone-restore-snapshot —cmd-args "—deleted-domain-ident —new-domain-ident —validate-new-domain —second-user= cp-on-call“
```

```
aescloudopstools run generic run —cmd-name es-tombstone-restore-snapshot —cmd-args "—deleted-domain-ident —new-domain-ident —restore —second-user= cp-on-call“
```

```
aescloudopstools run generic run —cmd-name es-tombstone-restore-snapshot —cmd-args "—deleted-domain-ident —new-domain-ident —delete-restore-repo —second-user= cp-on-call“
```

SOP:

https://w.amazon.com/bin/view/Search/A9/Swift/Ops/Procedures/AccidentalDeleteRecovery/#HProceduretorestoresnapshotfromdeleteddomain