- ECE&ES故障排查总结
  - 7.x集群规模达切磁盘io差的节点掉线后可能进进出出不稳定状态
  - 集群GC严重，甚至掉节点
  - 分析slowlog
  - 断路器
  - 集群索引删除保护
  - 索引读写优化
  - 分片迁移的优化：适当提升分片迁移速度，可根据实际情况使用
  - ece allocator节点标签消失（重启会有此状况）
  - ece cluster 删除残留
  - ece 同一个es集群节点迁移是主机级别迁移
  - 集群yellow
  - ece 2.6~2.10的ssl证书有效期只有一年，不升级到2.11.1，只能每年更换
  - snapshot中途停止导致大量的pendingtask
  - ece系统本身的es集群security变readonly导致cloudui登录失败
  - 索引数据的keyvalue互换导致集群不稳定，各种相应慢
  - 集群创建索引有分片落不下
  - 自定义分词设置不对，导致写超时
  - logging-and-metrics集群没有数据
  - 数据分布均匀情况下某些数据节点cpu消耗正常
  - snapshot删除失败，一直abort
  - 集群扩容可能触发整个集群rolling
  - ece的essnapshot设置不成功
  - es7.7.1指标采集不到
  - ece的ccs集群关联集群加减导致客户接入端报查询错误
  - ece的hot warm架构不支持后期转换
  - ilm策略配置错误，比如不正确的alias，导致大量的pendingtask，集群停止响应
  - shard reroute时集群indexing和search指标异常
  - bulk过大写入失败
  - zk节点进进出出，容器重启无效
- ESS运维和开发常用api
  - 运维常用api
    - 集群状态信息：节点数 分片数 未分片数 initialzing_tasks relocating_shards pending_tasks
    - 集群为何非green
    - 是否有queue、reject
    - node内存 cpu load信息
    - 每节点分片数 磁盘用量
    - 分片恢复状态
    - 索引字段mapping
    - snapshort仓库 备份进度 备份结果 删除备份
    - 集群默认设置
    - 各节点scroll执行的相关信息
    - es链接超时问题
    - es线程信息
    - 统计容器内存cache
  - 常用操作

        - 索引分片迁移 重新分配 取消分配
        - es index因磁盘空间read_only后操作
        - 取消查询任务
        - es集群移除相关节点、ip、zone，需要注意，有可能和分配策略的某些选项冲突导致无法正常完成
        - 开启自动reblance（默认：只管数量均衡，primary不能保证均衡）
        - 调整索引在每节点的分片数
        - 调整索引默认fields数量限制（默认1000）
        - 更改每个节点同时恢复的分片数量及最大数据流量
        - ece扩容节点个数参数配置
        - 创建policy及template
        - 删除索引ilm策略
        - 创建初始化ilm索引
        - 手动触发ilm索引rollover
        - 开启slowlog
        - ece rolling 重启
        - 节点常用cpu限制调整（ECE环境2.6.2前版本）
        - 通过门户新创建后调整审计日志的记录类型减少审计日志数量
    - ece常用api（具体body可参照官网各版本文档）
- ECE配置: 存储 内存 cpu 处理器
    - Memory
        - 以下是一些使弹性部署和ECE系统服务在主机上平稳运行的例子：
    - CPU quotas
    - Processors setting
        - 下表概述了根据上述公式计算元素搜索线程池时已分配的处理器：
    - Storage
- ece调整metrics-and-logging系统集群的配置
    - 调整索引清理策略
- ECE升级
    - 准备工作
    - 升级检查项
    - 升级到2.5.2
    - 升级到2.7.2
    - 升级到2.10.1
    - 升级到2.11.2
    - 脚本参考
        - 2.11.2安装包准备
        - 升级脚本修改：elastic-cloud-enterprise.sh
        - 证书有效期处理：因为2.6~2.10版本证书有效期只有一年，只能手动处理

ECE&ES故障排查总结

7.x集群规模达切磁盘io差的节点掉线后可能进进出出不稳定状态

```
原因：磁盘io不给力，元数据更新超时
方案：调整超时时间
    cluster.follower_lag.timeout: "600s"
    cluster.join.timeout: "600s"
    cluster.publish.timeout: "600s"
```

集群GC严重，甚至掉节点

```
取消一些search
    POST _tasks/_cancel?action=*search
```

分析slowlog

```
打开slowlog
    PUT /_all/_settings
    {
        "index.serach.slowlog.threshold.query.warn": "1s",
        "index.serach.slowlog.threshold.query.info": "500ms",
        "index.serach.slowlog.threshold.fetch.warn": "1s",
        "index.serach.slowlog.threshold.fetch.info": "500ms",
        "index.serach.slowlog.level": "info",
    }
关闭slowlog
    PUT /_all/_settings
    {
        "index.serach.slowlog.threshold.query.warn": null,
        "index.serach.slowlog.threshold.query.info": null,
        "index.serach.slowlog.threshold.fetch.warn": null,
        "index.serach.slowlog.threshold.fetch.info": null,
        "index.serach.slowlog.level": null,
    }
https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-slow.html
```

断路器

```
PUT _cluster/settings
{
    "persistent": {
        "indices.breaker.request.limit": "30%",
        "search.default_search_timeout": "10s",
        "search.max_buckets": 100000
    }
```

```
}
https://www.elastic.co/guide/en/elasticsearch/reference/current/circurit-breaker.html
```

集群索引删除保护

```
PUT _cluster/settings
{
    "persistent": {
        "action.destructive_requires_name": true
    }
}
```

索引读写优化

```
PUT index_name
{
    "settings": {
        "index.number_of_shards": "3",
        "index.refresh_interval": "30s",
        "index.routing.allocation.total_per_node": "3",
        "index.translog.durability": "async",
        "index.translog.sync_interval": "10s",
        "index.unassigned.node_left.delayed_timeout": "30m"
    }
}
"index.number_of_shards": 合理设置为节点倍数，但也不能太大，因为维护shards元数据也需要资源，过多也影响性能
"index.refresh_interval": 提升读写性能，但降低数据的实时性（为数据更新成功到数据可查的时间差）
"index.routing.allocation.total_per_node": 根据索引分片数和节点数计算一个合理值，可减少数据分布不均匀情况
"index.translog.durability": 改成异步可以提升性能，但有丢失数据风险
"index.unassigned.node_left.delayed_timeout": 数据节点掉线的时间，减少不必要的分片滚动，但如果分片设置少会带来一定风险
```

分片迁移的优化：适当提升分片迁移速度，可根据实际情况使用

```
PUT _cluster/settings
{
    "persistent": {
        "cluster.routing.allocation.node_concurrent_recoveries": "10",
        "cluster.routing.allocation.node_initial_primaries_recoveries": "30",
        "cluster.routing.allocation.cluster_concurrent_rebalance": "10",
```

```
        "indices.recovery.max_bytes_per_sec": "100mb"
    }
}
https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-cluster.html
```

ece allocator节点标签消失（重启会有此状况）

```
原因：已知bug，未部署过服务的节点可能会出现此类现象
处理方法：重启docker服务
预防：新加入allocator节点后，创建测试集群占坑
```

ece cluster 删除残留

```
原因：已知bug
处理：
    # admin console cluster
    DELETE cluster-ece-region/cluster/{clusterid}
    DELETE v1-elasticsearch-clusters-ece-region/elasticsearch-cluster/{clusterid}
```

ece 同一个es集群节点迁移是主机级别迁移

```
原因：默认把es实例从一台主机迁移到一台主机
处理：利用api

curl -XPOST http://{hostname}/api/v1/clusters/elasticsearch/{clusterid}/instances/_move -H "Context-Type: application/json" -d '
{
    "plan_configuration": {
        "move_instances": [
            {
                "from": "instanceid",
                "to": ["instanceid"]
            }
        ]
    }
}
'
```

集群yellow

```
检查集群状态：GET _cluster/health?v
检查未分配原因：GET _cat/allocation/explain
查看分配进度：GET _cat/recovery?active_only=true?v
重试分配失败的分片：POST _cluster/reroute?retry_failed
```

ece 2.6~2.10的ssl证书有效期只有一年，不升级到2.11.1，只能每年更换

```
file:///C:/Users/free/Documents/WeChat%20Files/wxid_pde84ty10c5d21/FileStorage/File/2021-12/ECE%20Certificate%20Rotation%20_%20Elastic%20Support.mhtml
```

插件更新： 原因：es节点原地更新插件，改动过的插件配置文件不会自动覆盖 处理：始终用grow and shrink 的集群变更方式更新插件

snapshot中途停止导致大量的pendingtask

```
原因：obs链接终端导致，而且不是简单的断网导致的原因
处理：重启主节点
```

ece系统本身的es集群security变readonly导致cloudui登录失败

```
原因：security集群给的内存较小，内存磁盘又是1:1，容易被自己的服务日志撑爆，磁盘超过95%
处理：去掉readonly标记，预防是启动初始化时进行优化手动调整磁盘内存比
PUT _all/_settings
{
    "index.blocks.read_only_allow_delete": false
}

deployments -> advance edit -> Data
"overrides": {
    "quota": {
        "fs_multiplier": 32
    }
}
```

索引数据的keyvalue互换导致集群不稳定，各种相应慢

```
愿意：key过长
处理：删除对应的索引，或者对其进行存储优化
```

集群创建索引有分片落不下

```
原因：total_shards_per_node限制
处理：调整为 shards_count / node_count + 1
```

自定义分词设置不对，导致写超时

```
不要滥用正则表达式，效率过低
```

logging-and-metrics集群没有数据

```
原因：2.6版本的filebeats账号权限bug
处理：deployments -> advance edit -> Data -> "found-internal-monitor" 添加
superuser 角色
```

数据分布均匀情况下某些数据节点cpu消耗正常

```
原因：比较多，gc，pendingtask，热点等等
处理：尝试重启，抓取慢日志等
```

snapshot删除失败，一直abort

```
找到有问题的snapshot节点，重启
    _cat/tasks
    _nodes/hot_threads?threads=100&ignore_idle_threads=false
如果不行，考虑重启master
如果还不行，重启所有节点
```

集群扩容可能触发整个集群rolling

```
原因：未知
```

ece的essnapshot设置不成功

```
原因：过程是异步的，需要等一段时间
如果长时间不能解决，重新设置一遍
```

es7.7.1指标采集不到

原因：版本缺陷，兼容性有问题

ece的ccs集群关联集群加减导致客户接入端报查询错误

原因：skip_unavailable被自动重置了

ece的hot warm架构不支持后期转换

避免这个场景，会降低稳定性，部分行为不可控制

ilm策略配置错误，比如不正确的alias，导致大量的pendingtask，集群停止响应

原因：无法自动rollover，如果同时集群还繁忙，会出现此类问题
处理：elasticsearch.yml中关闭ilm，重启集群；索引中去掉ilm；重新开启ilm并正确配置；

shard reroute时集群indexing和search指标异常

正常现象

bulk过大写入失败

调整参数：http.max_content_length，默认100M

zk节点进进出出，容器重启无效

原因：因某些原因导致文件损坏或磁盘内存等问题，导致出错
处理：

```
    docker stop frc-runners-runner
    docker inspect --format '{{ range .Mounts }}{{ .Source }}{{ end }}' frc-
zookeeper-servers-zookeeper | grep --color=auto "zookeeper/data"
/mnt/data/elastic/172.16.0.30/services/zookeeper/logs
/mnt/data/elastic/172.16.0.30/services/zookeeper/managed
/mnt/data/elastic/172.16.0.30/services/zookeeper/data
```

```
/mnt/data/elastic/172.16.0.30/services/zookeeper
    docker stop frc-zookeeper-servers-zookeeper
    cp -R /mnt/data/elastic/172.16.0.30/services/zookeeper/data
/mnt/data/elastic/zk_data_bak
    rm -R /mnt/data/elastic/172.16.0.30/services/zookeeper/data/version-?/
    docker start frc-runners-runner
    echo mntr | nc 172.16.0.30 zk-port
    check zk monitor
```

ESS运维和开发常用api

运维常用api

集群状态信息：节点数 分片数 未分片数 initialzing_tasks relocating_shards pending_tasks

```
GET _cluster/health
GET _cat/indices?v&health=yellow
GET _cat/indices?v&health=read
```

集群为何非green

```
GET _cluster/allocation/explain?pretty

GET _cat/pending_tasks?v
```

是否有queue、reject

```
GET _cat/thread_pool?v
```

node内存 cpu load信息

```
GET _cat/nodes?v
```

每节点分片数 磁盘用量

```
GET _cat/allocation?v
```

分片恢复状态

```
GET _cat/recovery?active_only
```

索引字段mapping

```
GET indexname/_mapping?filter_path=**.fieldname
```

snapshort仓库 备份进度 备份结果 删除备份

```
GET _snapshot/_all
GET _snapshot/_status?human
GET _snapshot/<respository>/snapshotname
DELETE _snapshot/<respository>/snapshotname
```

集群默认设置

```
GET _cluster/settings?flat_settings&include_defaults
```

各节点scroll执行的相关信息

```
GET /_nodes/stats/indices/search
```

es链接超时问题

```
    timeout 30s,  try catch 重试三次，能确保连得上
```

es线程信息

```
GET /_nodes/hot_threads
```

统计容器内存cache

```
cat /sys/fs/cgroup/memory/memory.stat | grep cache
```

常用操作

索引分片迁移 重新分配 取消分配

```
    POST _cluster/reroute?retry_failed

    POST _cluster/reroute
    {
        "commands" : [
            { "move" : { "index" : "ops", "shard" : 1, "from_node" :
"es_node_one", "to_node" : "es_node_two" } },
      { "cancel" : { "index" : "ops", "shard" : 0, "node" : "es_node_one" } },
            { "allocate" : { "index" : "ops", "shard" : 0, "node" : "es_node_two"
} }
        ]
  }
```

es index因磁盘空间read_only后操作

```
    PUT indexname/_settings
    {
        "index.blocks.read_only_allow_delete": "false"
    }
```

取消查询任务

```
    POST _tasks/_cancel?actions="search"
```

es集群移除相关节点、ip、zone，需要注意，有可能和分配策略的某些选项冲突导致无法正常完成

```
    PUT _cluster/settings
    {
        "persistent": {
            "cluster.routing.allocation.exclude._name/ip/zone":
"node01,node02,node03..."
        }
    }
```

开启自动reblance（默认：只管数量均衡，primary不能保证均衡）

```
PUT _cluster/settings
{
    "persistent": {
        "cluster.routing.rebalance.enable": "all"
    }
}
```

调整索引在每节点的分片数

```
PUT indexname/_settings
{
    "index.routing.allocation.total_shards_per_node": "2"
}
```

调整索引默认fields数量限制（默认1000）

```
PUT indexname/_settings
{
    "index.mapping.total_fields_limit": 1000
}
```

更改每个节点同时恢复的分片数量及最大数据流量

```
PUT _cluster/settings
{
    "persistent": {
        "cluster.routing.allocation.node_concurrent_recoveries": "6",
        "indices.recovery.max_bytes_per_node": "20mb"
    }
}
```

ece扩容节点个数参数配置

```
“node_count_per_zone": 2
"memory_per_node": 4096
```

创建policy及template

```
policy&template
GET _template/templatename
PUT _template/templatename
{
    "order": 1,
    "index_patterns": [
        "metricbeat-*", ......
    ],
    "settings": {
        "index": {
            "livecycle": {
                "name": "templatename"
            }
        }
    }
}

GET _ilm/policy/templatename

PUT _ilm/policy/templatename
{
    "policy": {
        "phases": {
            "min_age": "7d",
            "actions": {
                "delete": {}
            }
        }
    }
}
```

删除索引ilm策略

```
POST indexname/_ilm/remove
```

创建初始化ilm索引

```
PUT /%3C{indexname}-%7Bnow%2Fm%7Byyyy.MM.dd.mm%7CAsia%2FShanghai%7D%7D-
000001%3E
{
    "alias": {
        "indexalias": {
            "is_write_index": true
        }
```

```
        }
    }
```

手动触发ilm索引rollover

```
    POST indexalias/_rollover
```

开启slowlog

```
    PUT indname/_settings
    {
        "index.search.slowlog.threshold.query.debug": "1s",
        "index.search.slowlog.threshold.query.info": "3s",
        "index.search.slowlog.threshold.query.warn": "10s",
        "index.indexing.slowlog.threshold.query.debug": "1s",
        "index.indexing.slowlog.threshold.query.info": "5s",
        "index.search.slowlog.level": "debug"
    }
```

ece rolling 重启

```
    修改集群内容：node.attr.box_type1: isas
    然后选择：rolling inline size进行重启
    ”strategy": {
        "rolling": {
            "allow_rolling_resize": true
        }
    }
```

节点常用cpu限制调整（ECE环境2.6.2前版本）

```
    打开限制：deployment -> 集群名 -> opetations -> cpu hard limit setting -> turn
off
    调整cpu比例：advance edit ->
        "resources": {
            "cpu": {
                "hard_limit": true,
                "boost": true,
                "factor": 4
            }
        }
```

通过门户新创建后调整审计日志的记录类型减少审计日志数量

```
PUT _cluster/settings
{
    "persistent": {

        "xpack.security.audit.logfile.events.include": "connection_denied"
    }
}
```

ece常用api（具体body可参照官网各版本文档）

```
获取集群列表
    GET /api/v1/clusters/elasticsearch

获取集群详情
    GET /api/v1/clusters/elasticsearch/{clusterid}

创建集群
    POST /api/v1/clusters/elasticsearch

变更集群
    POST /api/v1/clusters/elasticsearch/{clusterid}/plan

重启集群
    POST /api/v1/clusters/elasticsearch/{clusterid}/_restart

停止集群
    POST /api/v1/clusters/elasticsearch/{clusterid}/_shutdown

释放集群
    DELETE /api/v1/clusters/elasticsearch/{clusterid}

重命名集群
    PATCH /api/v1/clusters/elasticsearch/{clusterid}/metadata/settings

变更kibana
    POST /api/v1/clusters/kibana/{clusterid}/plan

创建data节点规格
    POST /api/v1 /platform/configuration/instances/DATA_10

创建master节点规格
    POST /api/v1 /platform/configuration/instances/master_1

创建kibana节点规格
    POST /api/v1 /platform/configuration/instances/kibana_1
```

ECE配置: 存储 内存 cpu 处理器

在ECE中，每个主机都是一个runner。根据平台的大小，runner可以有一个或多个角色：协调员、主管、代理和分配器。在规划ECE安装的容量时，您必须正确地调整所有角色的容量大小。然而，分配器角色值得特别注意，因为它承载着元素搜索、Kibana、APM、企业搜索节点和相关服务。
本节重点介绍分配器角色，并解释如何在内存、CPU、处理器设置和存储方面规划其容量。

Memory

应该根据数据量来计划部署规模。内存是部署的主要缩放单元。其他单元，如CPU和磁盘，都与内存大小成正比。分配器可用的内存称为容量。
在安装过程中，分配器的容量默认为主机物理内存的85%，而其余的则预留给ECE系统服务。
要调整分配器容量，请使用新值重新安装ECE。如果无法在主机上重新安装，在服务器添加更多物理内存后，使用ECEAPI：

```
    curl -X PUT \

http(s)://<ece_admin_url:port>/api/v1/platform/infrastructure/allocators/<allocator_id>/settings \
    -H “Authorization: ApiKey $ECE_API_KEY” \
    -H 'Content-Type: application/json' \
    -d '{"capacity":<Capacity_Value_in_MB>}'
```

有关如何使用API密钥进行身份验证的详细信息，请参阅从命令行访问API一节。
无论如何使用此API，CPU配额都会使用在安装时指定的内存。

以下是一些使弹性部署和ECE系统服务在主机上平稳运行的例子：

如果运行器有多个角色（分配器、协调器、主管或代理），则应保留28GB的主机内存。例如，在具有256GBRAM的主机上，228GB适合于部署使用。
如果运行器只有分配器角色，则应该保留12GB的主机内存。例如，在具有256GBRAM的主机上，244GB适合于部署使用

CPU quotas

ECE使用CPU配额将分配器主机的共享分配给在它上运行的实例。若要计算CPU配额，请使用以下公式：

```
CPU quota = DeploymentRAM / HostCapacity

Consider a 32GB deployment hosted on a 128GB allocator.
If you use the default system service reservation, the CPU quota is 29%:
CPUquota = 32 / 128 * 0.85 = 29%
```

If you use 12GB Allocator system service reservation, the CPU quota is 28%:
CPUquota = 32 / (128 - 12) = 28%
这些百分比表示给定100ms周期内可用CPU资源总数百分比的上限。

Processors setting

除了CPU配额外，处理器的设置也起着相关的作用。
已分配的处理器设置来自元素搜索，并负责计算线程池。CPU配额定义了分配给实例的分配器的总CPU资源的百分比，而分配的处理器定义了如何在元素搜索中计算线程池，从而定义了实例可以处理多少并发搜索和索引请求。换句话说，CPU比率定义了单个任务完成的速度，而处理器设置定义了可以同时完成多少个不同的任务。
从运行在ECE2.7.0或更高版本上的元素搜索7.9.2版本开始，我们依靠元素搜索和-XX：激活进程或计算JVM设置来自动检测已分配的处理器。
在ECE和弹性搜索的早期版本中，使用元素搜索处理器设置，根据以下公式配置所分配的处理器：
Math.min(16,Math.max(2,(16*instanceCapacity*1.0/1024/64).toInt))

下表概述了根据上述公式计算元素搜索线程池时已分配的处理器：

instance size	vCPU
1024	2
2048	2
4096	2
8192	2
16384	4
32768	8
65536	16

该表还提供了在ECE和弹性搜索的较新版本中可以自动检测到的值的粗略指示。

Storage

ECE有特定的存储硬件先决条件。磁盘空间被系统日志、容器开销和部署数据所占用。
选择磁盘配额的主要因素是部署数据，即来自元素搜索、Kibana和APM节点的数据。最大部分的数据被元素搜索节点消耗。
ECE使用XFS强制执行特定的磁盘空间配额，以控制在分配器上运行的部署节点的磁盘消耗。
若要计算磁盘配额，请使用以下公式：
    Diskquota = ICmultiplier * Deployment RAM
        ICmultiplier是在ECE环境中定义的实例配置的磁盘倍增器。
        数据的默认乘数。默认值为32，用于热节点。数据的默认乘数。高存储空间是64，用于热节点和冷节点。冻结数据的FS乘数是80，用于冻结的节点。

您可以在不同的级别上更改磁盘乘法器的值： 在ECE级别上，请参见编辑实例配置。 在实例级别，登录到云UI，并按照以下方式继续操作： 在部署概览页面中，找到所需的实例，并打开实例菜单。 选择“覆盖磁盘配

额"。 根据您的需要调整磁盘配额。 覆盖只在实例容器的生命周期中持续存在。如果创建了一个新的容器，例如在grow_and_shrink计划或空出操作期间，该配额将被重置为其默认值。要以持久的方式增加存储比率，请编辑实例配置。

---

ece调整metrics-and-logging系统集群的配置

各索引的分片数：
    创建对应索引的新模板，内容如下：
        将新模板的优先级设置为高于老模板
        包含索引pattern
        将shards参数设为3
    一定不能带mappings，否则动态类型部分可能会出错，导致beats无法写入数据
    碰到allocator-metricbeats因为数据类型问题，无法写入数据，导致后果：ece界面无法查看内存 cpu等的使用量

调整索引清理策略

需要注意下：索引pattern不能带版本号
    要在firstnode或者director节点执行
    elastic-cloud-enterprise.sh set-logging-and-metrics-policy
    NAME
    bash elastic-cloud-enterprise.sh set-logging-and-metrics-policy Set the retention period for logging and metrics indices

    SYNOPSIS
    bash elastic-cloud-enterprise.sh set-logging-and-metrics-policy
    [--host-docker-host "PATH_NAME/docker.sock"] [--print] [--pattern PATTERN]
    [--days DAYS] [--secrets PATH_TO_SECRETS_FILE]
    [--host-storage-path PATH_NAME]
    [[--]help]
    REQUIRED PRIVILEGES
    To run this script, a user must be part of the docker group. The host that you run this script on must be the first host that you installed Elastic Cloud Enterprise on or a host that holds the director role.

    DESCRIPTION
    Set the retention period for the logging and metrics indices that allow you to monitor your Elastic Cloud Enterprise installation.

    PARAMETERS
    --host-docker-host /PATH_NAME/docker.sock
    Specifies the location of the Docker socket used to communicate with the Docker daemon. Defaults to /var/run/docker.sock.
    --print
    Prints the policy that is currently in effect. Cannot be specified together

```
with --pattern and --days.
    --pattern PATTERN
    Specifies the index pattern for which a new retention period will be set, such
as cluster-logs-*. If specified, --days must also be specified.
    --days DAYS
    Specifies the number of days that indices matching the given pattern are kept.
Defaults to one day for patterns that match metrics indices and seven days for
patterns that match logging indices. Specifying 0 or -1 disables index curation
and results in indices being retained indefinitely. If specified, --pattern must
also be specified.
    --secrets PATH_TO_SECRETS_FILE
    Specifies a path to a file with secrets. If not specified, attempts to use
HOST_STORAGE_PATH/bootstrap-state/bootstrap-secrets.json, where HOST_STORAGE_PATH
is the host storage path used by the Elastic Cloud Enterprise installation.
    --host-storage-path PATH_NAME
    Specifies the host storage path used by the Elastic Cloud Enterprise
installation. Defaults to /mnt/data/elastic. Used for determining the default
locations of the secrets file and log files.
    WARNING: This path cannot be the exact XFS volume mount point, it must be the
subdirectory of it. By default, the XFS volume should be mounted at /mnt/data.
    Example
    Set the retention period for the cluster-logs-* index pattern to 14 days:
    bash elastic-cloud-enterprise.sh set-logging-and-metrics-policy --pattern
cluster-logs-* --days 14
```

ECE升级

版本升级有个过程，本次记录从2.3.1升级到2.11的过程

准备工作

1 各版本的ece包：2.3 -> 2.5 -> 2.7 -> 2.10 -> 2.11
2 将ece包导入私有仓库
3 将ece包拉取到各个节点
4 准备好升级版本的安装脚本，修改对应的版本及私有仓库url

升级检查项

每次升级前都检查下，包括但不限于如下项目。
1 剩余内存
2 剩余磁盘空间

```
3   当前ece版本，license，证书有效期
4   当前es集群各版本
```

升级到2.5.2

```
1   执行升级脚本
2   重新导入license
3   系统集群要升级到6.8.13
    注意：如果logging-and-metrics过大，先禁用快照，否则备份时间会比较久
        另外升级前清理不需要的数据，否则迁移分片时间会比较长
        如果启用kibana，则kibana需要单独升级
```

升级到2.7.2

```
1   执行升级脚本
2   重新导入license
3   系统集群要升级到6.8.17
    注意：如果logging-and-metrics过大，先禁用快照，否则备份时间会比较久
    kibana需要单独升级
    security没有自动升级，需要手动升级到对应的版本
4   证书要求要在4以上
    curl -X GET -u admin:PASSWORD -k
https://COORDINATOR_HOST:12443/api/v1/platform/license
        {
          "license": {
            "version": 4,
            // other fields
          }
        }
```

升级到2.10.1

```
测试和实际中能正常升级
```

升级到2.11.2

```
测试和实际中能正常升级
```

脚本参考

2.11.2安装包准备

```
docker pull docker.elastic.co/cloud-enterprise/elastic-cloud-enterprise:2.11.2
docker pull docker.elastic.co/cloud-assets/elasticsearch:7.14.0-0
docker pull docker.elastic.co/cloud-assets/kibana:7.14.0-1
docker pull docker.elastic.co/cloud-assets/apm:7.14.0-0
docker pull docker.elastic.co/cloud-assets/enterprise-search:7.14.0-0

docker tag docker.elastic.co/cloud-enterprise/elastic-cloud-enterprise:2.11.2
REGISTRY/cloud-enterprise/elastic-cloud-enterprise:2.11.2
docker tag docker.elastic.co/cloud-assets/elasticsearch:7.14.0-0 REGISTRY/cloud-
assets/elasticsearch:7.14.0-0
docker tag docker.elastic.co/cloud-assets/kibana:7.14.0-1 REGISTRY/cloud-
assets/kibana:7.14.0-1
docker tag docker.elastic.co/cloud-assets/apm:7.14.0-0 REGISTRY/cloud-
assets/apm:7.14.0-0
docker tag docker.elastic.co/cloud-assets/enterprise-search:7.14.0-0
REGISTRY/cloud-assets/enterprise-search:7.14.0-0

docker push REGISTRY/cloud-enterprise/elastic-cloud-enterprise:2.11.2
docker push REGISTRY/cloud-assets/elasticsearch:7.14.0-0
docker push REGISTRY/cloud-assets/kibana:7.14.0-1
docker push REGISTRY/cloud-assets/apm:7.14.0-0
docker push REGISTRY/cloud-assets/enterprise-search:7.14.0-0
```

升级脚本修改：elastic-cloud-enterprise.sh

```
# ECE version
CLOUD_ENTERPRISE_VERSION=2.11.2 # 改为对应的版本
# Default Docker registry
DOCKER_REGISTRY=docker.elastic.co # 改为本地仓库

执行：bash elastic-cloud-enterprise.sh upgrade
```

证书有效期处理：因为2.6~2.10版本证书有效期只有一年，只能手动处理

```
脚本：curl -L -o  /tmp/rotate-certificates.sh https://ela.st/ece-cert-rotation-
script
chmod +x /tmp/rotate-certificates.sh

证书上传在ECE管理界面；

检查：DIRECTOR_ENTRYPOINT=<hostname-of-first-director> COMMAND=check /tmp/rotate-
certificates.sh
执行：DIRECTOR_ENTRYPOINT=<hostname-of-first-director> COMMAND=run /tmp/rotate-
certificates.sh
如果有问题，可以在脚本开始处增加配置：
```

```
    DIRECTOR_ENTRYPOINT=<hostname-of-first-director>
    COMMAND=check

命令参考: COMMAND (default value: run) - supported commands:
    run - Starts or continues a certificate rotation on a host. If a run
previously failed, the run command continues from the last failed step, skipping
steps that were already taken.
    rollback - Reverts or continues reverting changes produced by running a
certificate rotation on a host. If a run previously failed, the rollback command
will revert the steps that were already taken. If a rollback previously failed, it
will continue to to rollback from the last failed step, skipping steps which were
already reverted.
    check - Prints the expiration date of the certificates for that host.
    state - In case a rotation failed, this command prints the steps that were
already taken and the results.
    backup - In case a rotation failed, this command prints the backup taken
before the run started.
    cleanup - This deletes all information related to the certificate rotation on
every host (history of changes and backups). This should only be run once all the
hosts have been successfully rotated.
```