Instructions & Tasks

You work as a system administrator for a company that wants to use Elasticsearch with Kibana to store and analyze some log data. You are being asked to prepare the Elasticsearch cluster for the log data by creating some indexes. Logs are considered time-series data, and we typically care most about the most recent logs. We need to make sure the data we care most about is allocated to our fastest, or "hot," nodes. The data we care less about can be allocated to the smaller "warm" nodes, which won't be indexed or searched as often. We also need to be able to search the data using aliases such as "this\_week" or "last\_week". Aliases make it easy to search the data you care about because you don't have to know specific index names.

You have a pre-configured, 3-node Elasticsearch cluster with Kibana already set up and running. You will need to use either the command line curlcommand or Kibana's console tool to interact with Elasticsearch's APIs to create the following indexes:

| **Name** | **Alias** | **Primary Shards** | **Replica Shards** | **Allocation** |
| --- | --- | --- | --- | --- |
| logs-01 | this\_week | 3 | 1 | hot |
| logs-02 | last\_week | 3 | 1 | warm |

Your 3-node Elasticsearch cluster is already up and running on the master node, along with a Kibana instance. You can perform the instructions above with either Kibana's console tool or curl on the command line. If you want to use Kibana's console tool for this activity, you will need to perform Task 1. Otherwise, skip Task 1 and continue to Task 2.

help

check\_box\_outline\_blank

**OPTIONAL: Set up a remote tunnel to Kibana on the master node.**

**NOTE: This task only needs to be performed if you're using the Kibana UI to interact with Elasticsearch.**

Set Up a Remote Tunnel

1. Open a new terminal window and use ssh to log in to the master node as cloud\_user with port forwarding.

ssh cloud\_user@your\_public\_ip -L 5601:localhost:5601

Open the Kibana Console Tool

1. In your local web browser, go to http://localhost:5601.
2. In Kibana, navigate to **Dev Tools** in the side navigation bar.
3. Select the **Console** tool (it should be the default tool that loads).

help

check\_box\_outline\_blank

**Create the `logs-01` index.**

Option 1: Kibana Console Tool

Create logs-01

1. To create the logs-01 index, use the Console to send the following request to Elasticsearch:
2. PUT /logs-01
3. {
4. "aliases": {
5. "this\_week": {}
6. },
7. "settings": {
8. "number\_of\_shards": 3,
9. "number\_of\_replicas": 1,
10. "index.routing.allocation.require.temp": "hot"
11. }

}

Option 2: Command line curl

Create logs-01

1. To create the logs-01 index, execute the following from the command line of one of the nodes:
2. curl -XPUT "http://localhost:9200/logs-01" -H 'Content-Type: application/json' -d'
3. {
4. "aliases": {
5. "this\_week": {}
6. },
7. "settings": {
8. "number\_of\_shards": 3,
9. "number\_of\_replicas": 1,
10. "index.routing.allocation.require.temp": "hot"
11. }

}'

help

check\_box\_outline\_blank

**Create the `logs-02` index.**

Option 1: Kibana Console Tool

Create logs-02

1. To create the logs-02 index, use the Console to send the following request to Elasticsearch:
2. PUT /logs-02
3. {
4. "aliases": {
5. "last\_week": {}
6. },
7. "settings": {
8. "number\_of\_shards": 3,
9. "number\_of\_replicas": 1,
10. "index.routing.allocation.require.temp": "warm"
11. }

}

Option 2: Command line curl

Create logs-02

1. To create the logs-02 index, execute the following from the command line of one of the nodes:
2. curl -XPUT "http://localhost:9200/logs-02" -H 'Content-Type: application/json' -d'
3. {
4. "aliases": {
5. "last\_week": {}
6. },
7. "settings": {
8. "number\_of\_shards": 3,
9. "number\_of\_replicas": 1,
10. "index.routing.allocation.require.temp": "warm"
11. }

}'