

2. Elasticsearch 클러스터링

1. 목표

- 서로 다른 서버들을 클러스터로 묶는 실습 진행.
- AMI를 활용해 인스턴스를 복제하고 클러스터링 설정.

2. 클러스터링 구성 과정

1) 기존 인스턴스를 이미지로 생성

1. 기존 인스턴스를 ****AMI(Image)****로 생성.
2. 새로 생성된 인스턴스들의 보안 그룹은 기존 인스턴스의 보안 그룹을 동일하게 설정.
3. 인스턴스 개수를 2개로 설정하여 총 3개 노드로 클러스터 구성.

2) **/etc/hosts** 파일 수정

- 각 노드의 프라이빗 IP 주소와 별명을 설정:

```
bash
sudo vim /etc/hoss
```

```
yaml
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
<노드1 프라이빗 IP> es-1
<노드2 프라이빗 IP> es-2
<노드3 프라이빗 IP> es-3
```

- 각 노드에서 **/etc/hosts** 를 동일하게 설정.

3) 보안 그룹 설정

- **인바운드 규칙**에서 포트 열기:
 - 9200 (HTTP 통신)
 - 9300 (노드 간 통신)
- 보안 그룹 수정 시 동일 보안 그룹을 사용하는 모든 인스턴스에 적용됨.

4) **elasticsearch.yml** 파일 수정

- 각 노드의 설정 파일 수정:

```
bash
sudo vim /path/to/elasticsearch-7.10.2/config/elasticsearch.yml
```

- 수정 내용:

```
yaml
cluster.name: elastic-cluster
node.name: node-n # 각 노드마다 고유한 이름 설정 (예: node-1, node-2, node-3)
network.host: ["_local_", "_site_"]
discovery.seed_hosts: ["es-1", "es-2", "es-3"]
cluster.initial_master_nodes: ["node-1", "node-2", "node-3"]
```

5) Elasticsearch 실행

- 각 노드에서 Elasticsearch 실행:

```
bash
/path/to/elasticsearch-7.10.2/bin/elasticsearch &
```

- 실행 후 다른 노드와 연결 상태 확인:
 - `curl {각 노드의 퍼블릭 IP}:9200/_cat/nodes`
 - 모든 노드가 검색되면 정상적으로 클러스터가 구성된 것.

3. 클러스터 구성 결과

- 정상적인 실행 결과:

```
bash
curl 52.79.240.80:9200/_cat/nodes
```

```
plaintext
172.31.44.216 32 47 -1 0.05 0.08 0.05 cdhilmrstw - node-3
172.31.43.128 14 47 -1 0.08 0.13 0.05 cdhilmrstw * node-2
172.31.44.111 38 50 -1 0.12 0.13 0.06 cdhilmrstw - node-1
```

- 각 노드가 9200 포트를 통해 HTTP 통신, 9300 포트를 통해 클러스터 연결.

4. 문제 해결 (Trouble Shooting)

1) 메모리 설정 문제

- 오류 메시지:

```
plaintext
max virtual memory areas vm.max_map_count [65530] is too
low, increase to at least [262144]
```

- 해결 방법:

```
bash
sudo vim /etc/sysctl.conf
```

설정 추가:

```
plaintext
vm.max_map_count=262144
```

설정 적용:

```
bash
sudo sysctl -p
```

2) 데이터 초기화

- 클러스터 문제가 지속될 경우, 기존 데이터 삭제 후 다시 실행:

```
bash
rm -rf /home/ubuntu/elasticsearch-7.10.2/data/*
```

5. 검증

- 각 노드의 상태 확인:

```
bash
curl {각 노드의 퍼블릭 IP}:9200
```

모든 노드가 정상적으로 클러스터링된 상태 확인 완료.