Docker Compose로 Elastic Cluster 구성

> 사전 준비 : Docker 설치

💡 !!오탈자 주의!!

1. 필요한 파일 준비

▼ instances.yml 파일

```
instances:
  - name: es01
    dns:
      - es01
      - localhost
    ip:
      - 127.0.0.1

  - name: es02
    dns:
      - es02
      - localhost
    ip:
      - 127.0.0.1

  - name: es03
    dns:
      - es03
      - localhost
    ip:
      - 127.0.0.1

  - name: 'kib01'
    dns:
      - kib01
      - localhost
```

▼ .env 파일

```
COMPOSE_PROJECT_NAME=es
CERTS_DIR=/usr/share/elasticsearch/config/certificates
VERSION=7.15.0
```

▼ create-certs.yml 파일

```
version: '2.2'

services:
  create_certs:
    image: docker.elastic.co/elasticsearch/elasticsearch:${VERSION}
    container_name: create_certs
    command: >
      bash -c '
        yum install -y -q -e 0 unzip;
        if [[ ! -f /certs/bundle.zip ]]; then
          bin/elasticsearch-certutil cert --silent --pem --in config/certificates/instances.yml -out /certs/bundle.zip;
          unzip /certs/bundle.zip -d /certs;
        fi;
        chown -R 1000:0 /certs
      '
    working_dir: /usr/share/elasticsearch
    volumes:
      - certs:/certs
      - .:/usr/share/elasticsearch/config/certificates
    networks:
      - elastic
```

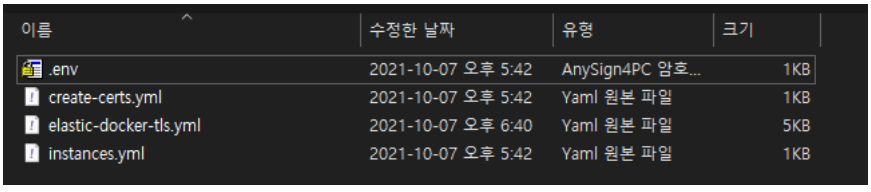
```
volumes:
  certs:
    driver: local

networks:
  elastic:
    driver: bridge
```

▼ elastic-docker-tls.yml 파일

```
version: '2.2'

services:
  es01:
    image: docker.elastic.co/elasticsearch/elasticsearch:${VERSION}
    container_name: es01
    environment:
      - node.name=es01
      - cluster.name=es-docker-cluster
      - discovery.seed_hosts=es02,es03
      - cluster.initial_master_nodes=es01,es02,es03
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - xpack.license.self_generated.type=trial
      - xpack.security.enabled=true
      - xpack.security.http.ssl.enabled=true
      - xpack.security.http.ssl.key=$CERTS_DIR/es01/es01.key
      - xpack.security.http.ssl.certificate_authorities=$CERTS_DIR/ca/ca.crt
      - xpack.security.http.ssl.certificate=$CERTS_DIR/es01/es01.crt
      - xpack.security.transport.ssl.enabled=true
      - xpack.security.transport.ssl.verification_mode=certificate
      - xpack.security.transport.ssl.certificate_authorities=$CERTS_DIR/ca/ca.crt
      - xpack.security.transport.ssl.certificate=$CERTS_DIR/es01/es01.crt
      - xpack.security.transport.ssl.key=$CERTS_DIR/es01/es01.key
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data01:/usr/share/elasticsearch/data
      - certs:$CERTS_DIR
    ports:
      - 9200:9200
    networks:
      - elastic

    healthcheck:
      test: curl --cacert $CERTS_DIR/ca/ca.crt -s https://localhost:9200 >/dev/null; if [[ $$? == 52 ]]; then echo 0; else echo 1;
      interval: 30s
      timeout: 10s
      retries: 5

  es02:
    image: docker.elastic.co/elasticsearch/elasticsearch:${VERSION}
    container_name: es02
    environment:
      - node.name=es02
      - cluster.name=es-docker-cluster
      - discovery.seed_hosts=es01,es03
      - cluster.initial_master_nodes=es01,es02,es03
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - xpack.license.self_generated.type=trial
      - xpack.security.enabled=true
      - xpack.security.http.ssl.enabled=true
      - xpack.security.http.ssl.key=$CERTS_DIR/es02/es02.key
      - xpack.security.http.ssl.certificate_authorities=$CERTS_DIR/ca/ca.crt
      - xpack.security.http.ssl.certificate=$CERTS_DIR/es02/es02.crt
      - xpack.security.transport.ssl.enabled=true
      - xpack.security.transport.ssl.verification_mode=certificate
      - xpack.security.transport.ssl.certificate_authorities=$CERTS_DIR/ca/ca.crt
      - xpack.security.transport.ssl.certificate=$CERTS_DIR/es02/es02.crt
      - xpack.security.transport.ssl.key=$CERTS_DIR/es02/es02.key
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data02:/usr/share/elasticsearch/data
      - certs:$CERTS_DIR
    networks:
      - elastic
```

```
  es03:
    image: docker.elastic.co/elasticsearch/elasticsearch:${VERSION}
    container_name: es03
    environment:
      - node.name=es03
      - cluster.name=es-docker-cluster
      - discovery.seed_hosts=es01,es02
      - cluster.initial_master_nodes=es01,es02,es03
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - xpack.license.self_generated.type=trial
      - xpack.security.enabled=true
      - xpack.security.http.ssl.enabled=true
      - xpack.security.http.ssl.key=$CERTS_DIR/es03/es03.key
      - xpack.security.http.ssl.certificate_authorities=$CERTS_DIR/ca/ca.crt
      - xpack.security.http.ssl.certificate=$CERTS_DIR/es03/es03.crt
      - xpack.security.transport.ssl.enabled=true
      - xpack.security.transport.ssl.verification_mode=certificate
      - xpack.security.transport.ssl.certificate_authorities=$CERTS_DIR/ca/ca.crt
      - xpack.security.transport.ssl.certificate=$CERTS_DIR/es03/es03.crt
      - xpack.security.transport.ssl.key=$CERTS_DIR/es03/es03.key
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data03:/usr/share/elasticsearch/data
      - certs:$CERTS_DIR
    networks:
      - elastic
  kib01:
    image: docker.elastic.co/kibana/kibana:${VERSION}
    container_name: kib01
    depends_on: {"es01": {"condition": "service_healthy"}}
    ports:
      - 5601:5601
    environment:
      SERVERNAME: localhost
      ELASTICSEARCH_URL: https://es01:9200
      ELASTICSEARCH_HOSTS: https://es01:9200
      ELASTICSEARCH_USERNAME: kibana_system
      ELASTICSEARCH_PASSWORD: CHANGEME
      ELASTICSEARCH_SSL_CERTIFICATEAUTHORITIES: $CERTS_DIR/ca/ca.crt
      SERVER_SSL_ENABLED: "true"
      SERVER_SSL_KEY: $CERTS_DIR/kib01/kib01.key
      SERVER_SSL_CERTIFICATE: $CERTS_DIR/kib01/kib01.crt
    volumes:
      - certs:$CERTS_DIR
    networks:
      - elastic
volumes:
  data01:
    driver: local
  data02:
    driver: local
  data03:
    driver: local
  certs:
    driver: local

networks:
  elastic:
    driver: bridge
```

- 폴더 내 파일 예시



2. vm.max_map_count 값 변경

- Powershell에서 다음과 같은 명령어 입력
- !재부팅 할때마다 실행!

```
> wsl -d docker-desktop
sysctl -w vm.max_map_count=262144
```

3. 인증서 생성을 위한 create-certs 컨테이너 실행

- Powershell에서 다음과 같은 명령어 입력 (create-certs.yml 파일이 있는 곳으로 이동 후 실행)

```
> docker-compose -f create-certs.yml run --rm create_certs
```

4. Elastic Cluster 실행

- Powershell에서 다음과 같은 명령어 입력 (elastic-docker-tls.yml 파일이 있는 곳으로 이동 후 실행)

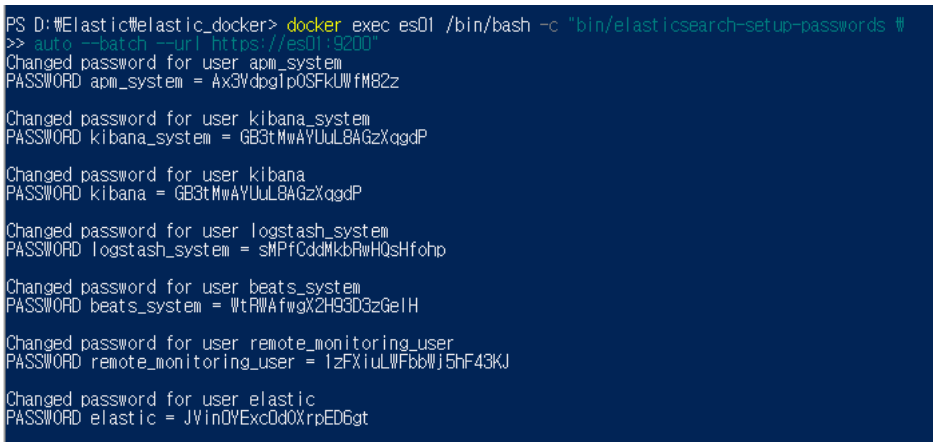
```
> docker-compose -f elastic-docker-tls.yml up -d
```

5. Password 자동생성

- Powershell에서 다음과 같은 명령어 입력

```
docker exec es01 /bin/bash -c "bin/elasticsearch-setup-passwords \
auto --batch --url https://es01:9200"
```

- 명령어 실행 결과 주요 Elastic의 system user의 비밀번호가 아래와 같이 자동 생성



6. Elastic Cluster 설정 파일 수정

- 생성된 비밀번호로 elastic-docker-tls.yml 파일의 kibana_system의 비밀번호 수정

```
kib01:
    image: docker.elastic.co/kibana/kibana:${VERSION}
    container_name: kib01
    depends_on: {"es01": {"condition": "service_healthy"}}
    ports:
      - 5601:5601
    environment:
      SERVERNAME: localhost
      ELASTICSEARCH_URL: https://es01:9200
      ELASTICSEARCH_HOSTS: https://es01:9200
```

```
      ELASTICSEARCH_USERNAME: kibana_system
      *ELASTICSEARCH_PASSWORD: GB3tMwAYUuL8AGzXqgdP
      ELASTICSEARCH_SSL_CERTIFICATEAUTHORITIES: $CERTS_DIR/ca/ca.crt
      SERVER_SSL_ENABLED: "true"
      SERVER_SSL_KEY: $CERTS_DIR/kib01/kib01.key
      SERVER_SSL_CERTIFICATE: $CERTS_DIR/kib01/kib01.crt
    volumes:
      - certs:$CERTS_DIR
    networks:
      - elastic
```

7. Elastic Cluster 재시작

- Powershell에서 다음과 같은 명령어 입력

```
> docker-compose stop
> docker-compose -f elastic-docker-tls.yml up -d
```

8. Kibana 접속

- `https://localhost:5601` 접속
- userid : elastic
- passwd : 생성된 비밀번호 참조

9. Docker Container 삭제(Volume 포함 삭제)

- Volume, Network를 포함한 Elasticsearch Cluster Container 삭제 명령어(모든 데이터 삭제 주의)

```
> docker-compose -f elastic-docker-tls.yml down -v
```

참고자료

- https://www.elastic.co/guide/en/elastic-stack-get-started/7.15/get-started-docker.html#get-started-docker-tls
- https://stackoverflow.com/questions/42111566/elasticsearch-in-windows-docker-image-vm-max-map-count