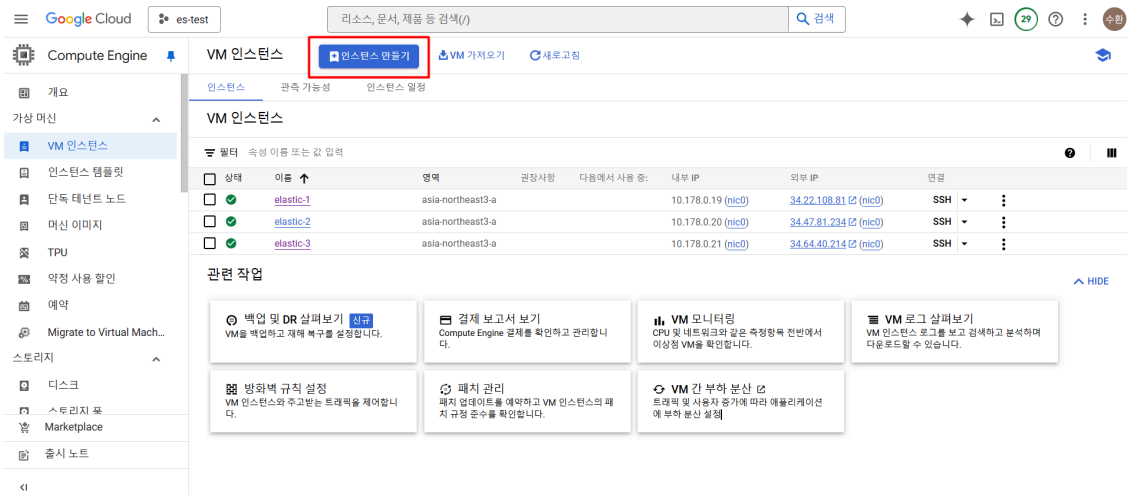
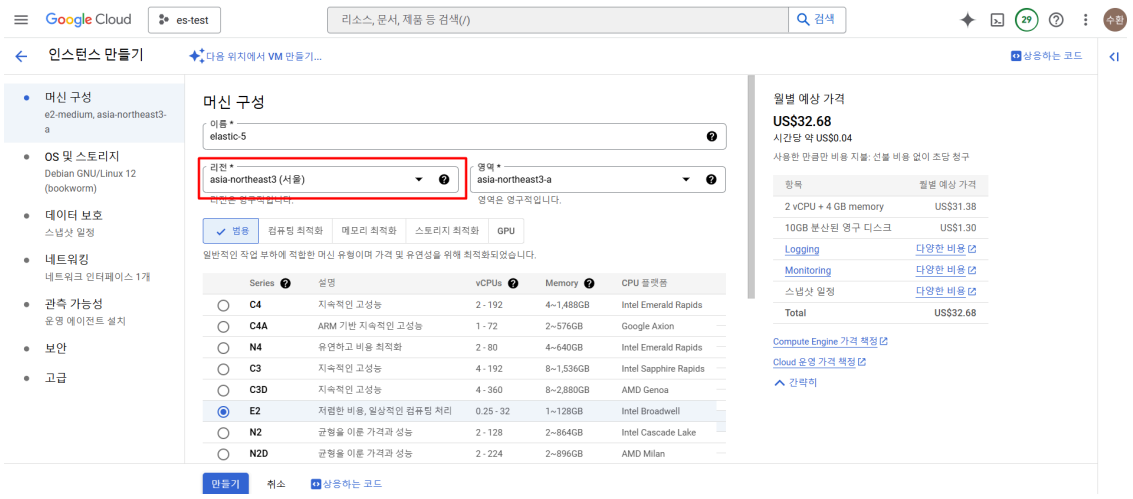
Elasticsearch 기본세팅

GCP 인스턴스 생성

https://cloud.google.com/

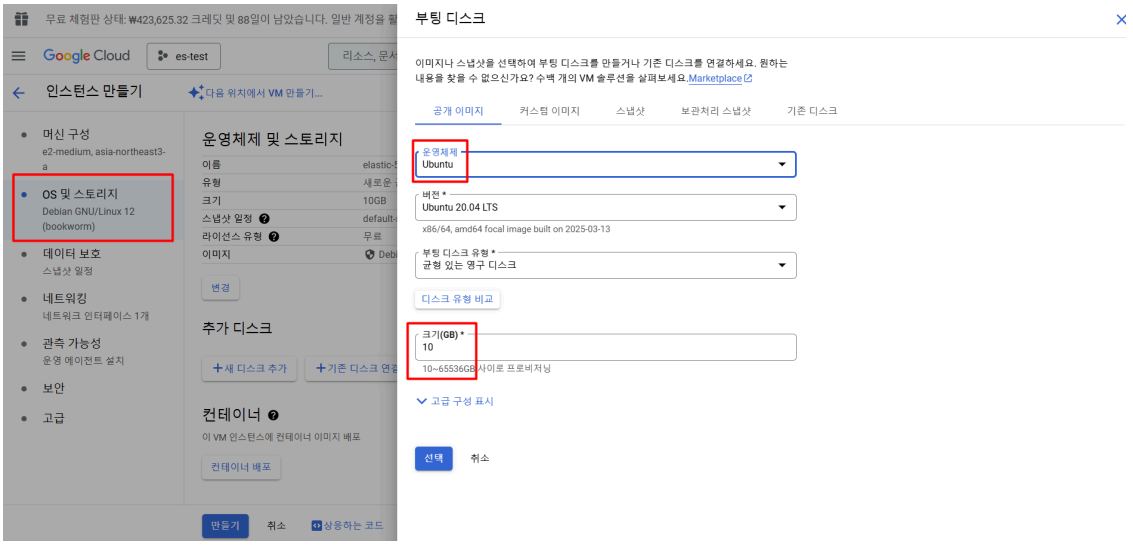
처음 계정을 만들면 $300(430,000원) 90일간 주기 때문에 무료로 어느정도 사용할 수 있다.



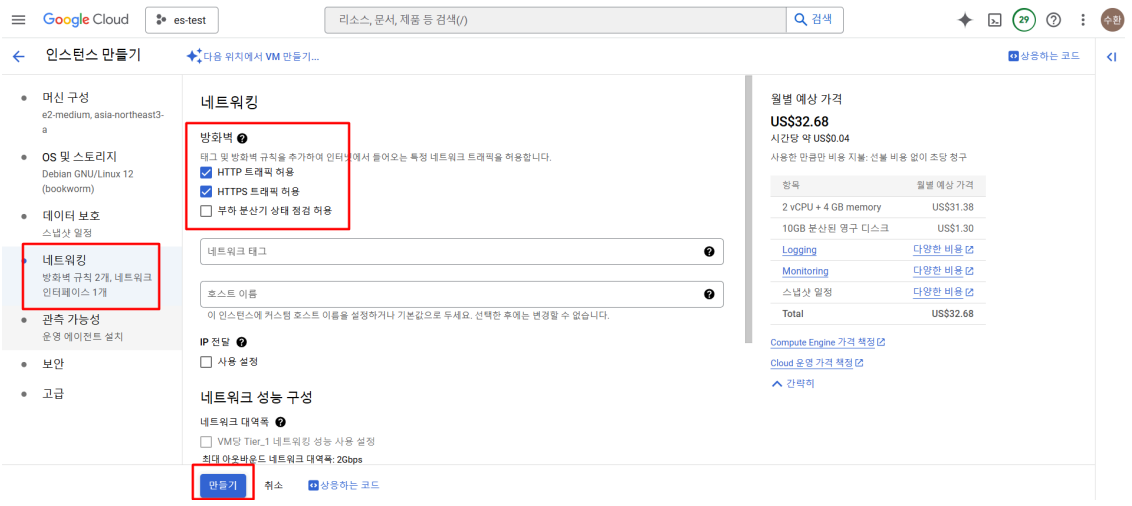


지역을 서울로 맞춰주고

영역은 a, b, c 중에서 원하는거 선택하면 된다.



운영체제도 원하는 거 선택하면 된다. 나는 우분투가 익숙해서 우분투로 생성하였다.

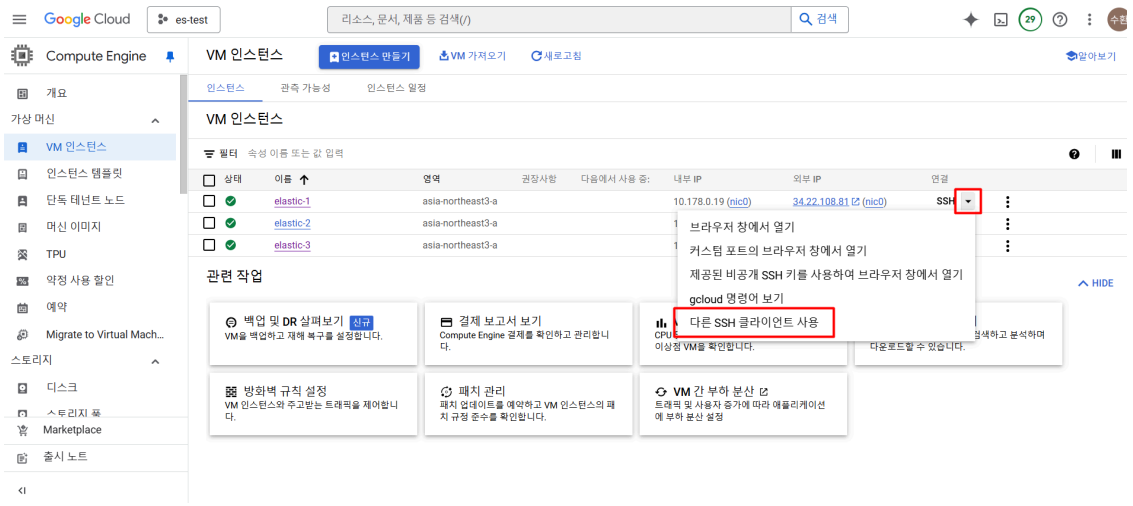


http, https 를 허용한다.

참고로 Elasticsearch 8버전부터는 기본적으로 TLS 를 지원한다(http 통신말고 https 통신을 기본으로 한다.)

이전 버전에는 이것을 다 설정해줬어야 했는데 default 세팅으로 변경되었다.

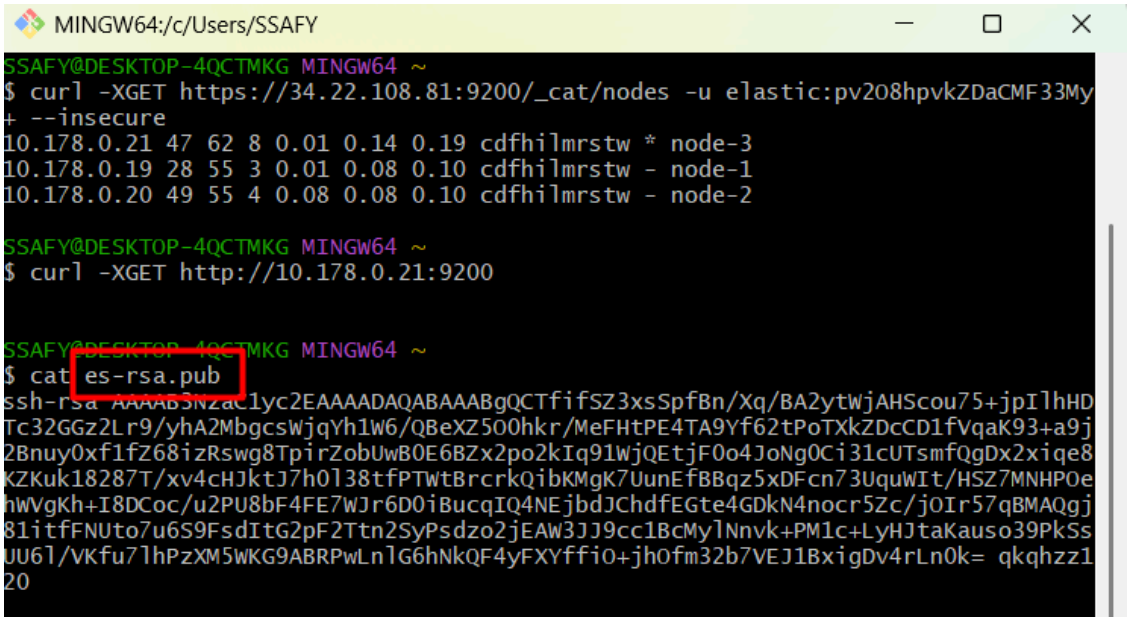
http 통신이 안된다. 참고하자



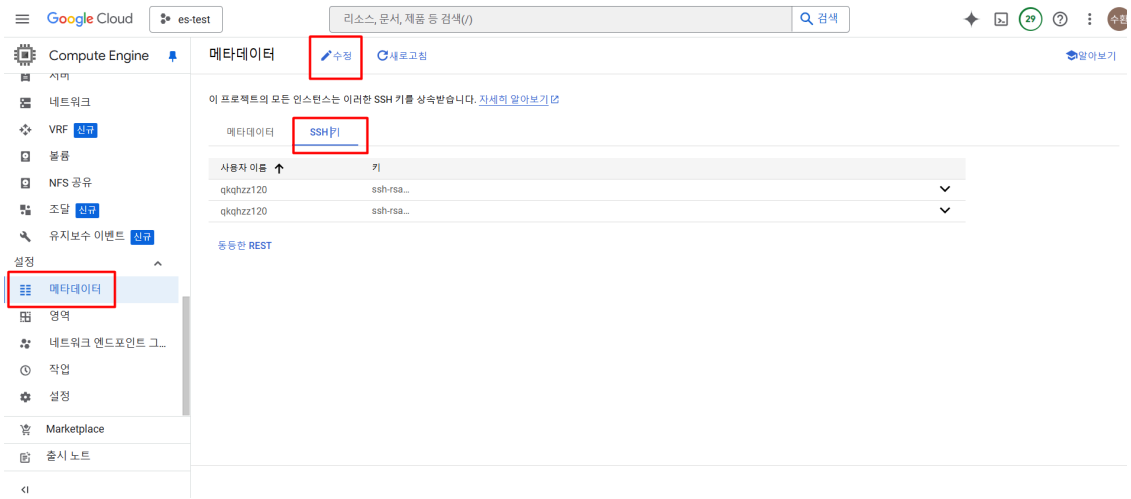
웹으로 터미널 열어서 사용하지 않고

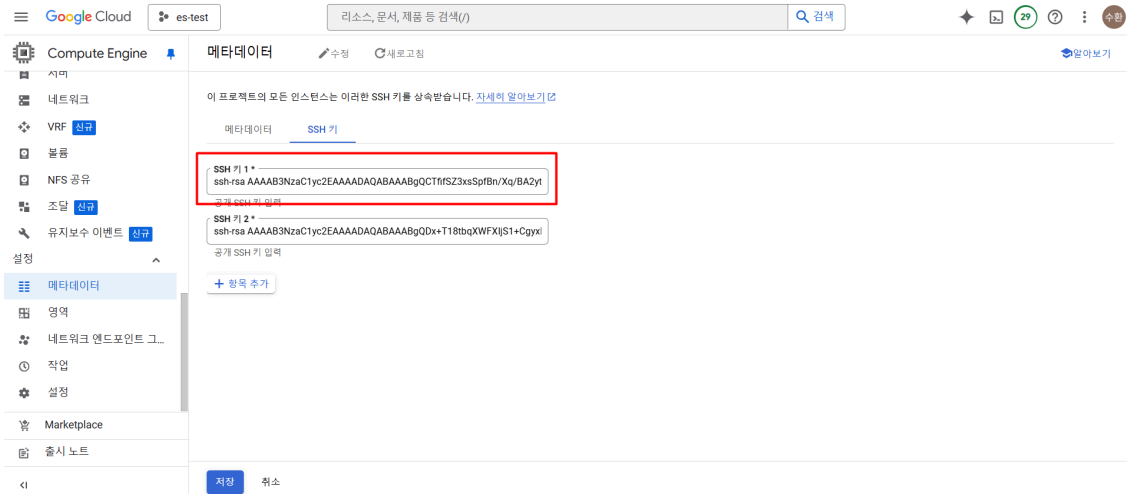
로컬에서 터미널로 사용하기 위해서 하는 설정

운영체제에 맞게 선택해서 절차대로 하면 된다.



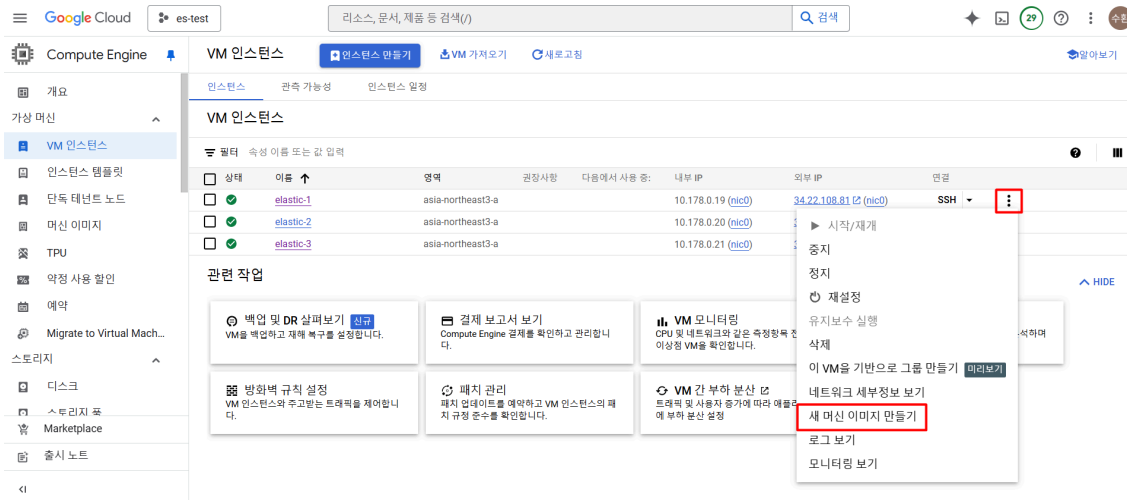
이 값을 복사해서 ssh 키에 추가해줘야 한다.



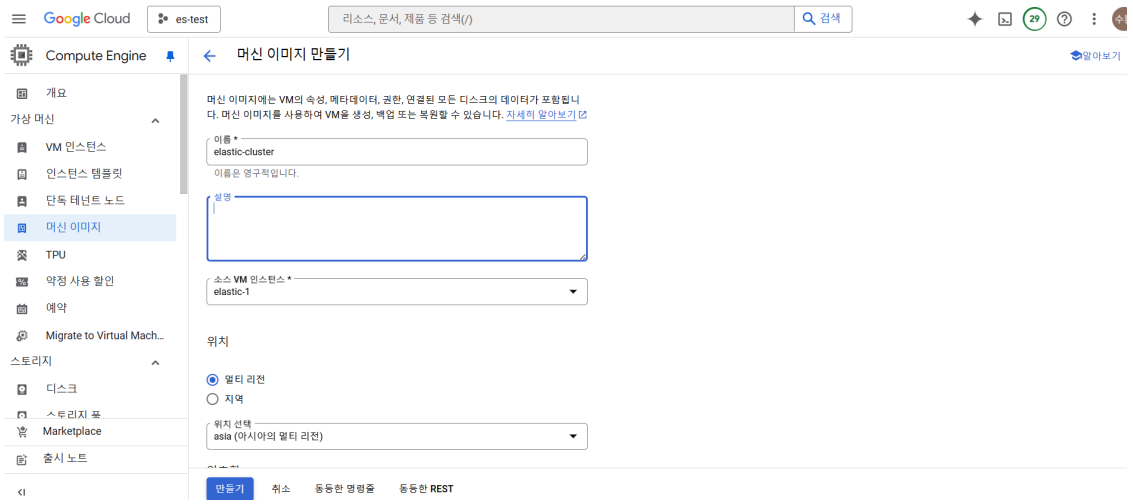


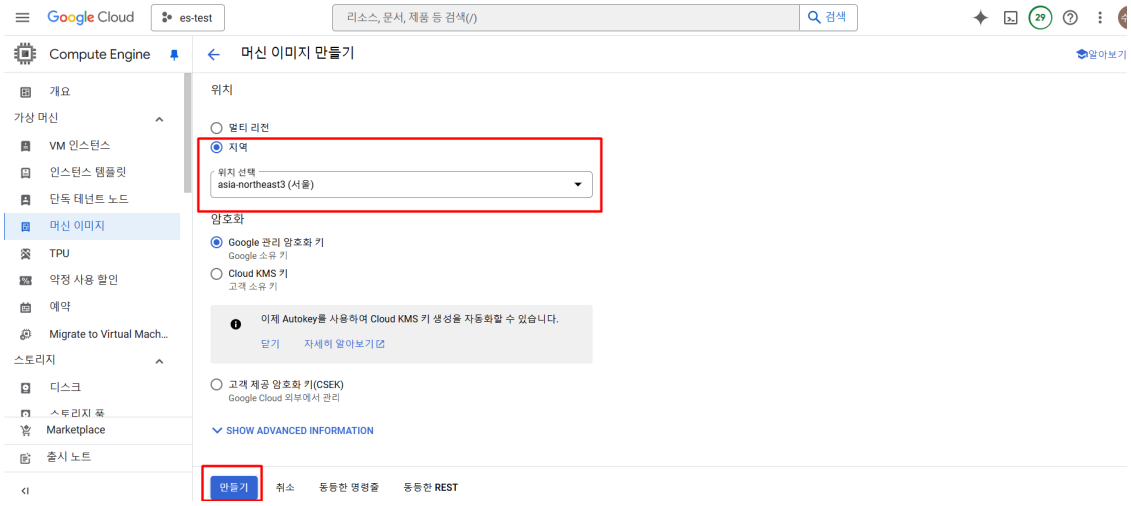
그냥 복붙하면 된다.

클러스터 구성 노드 바인딩

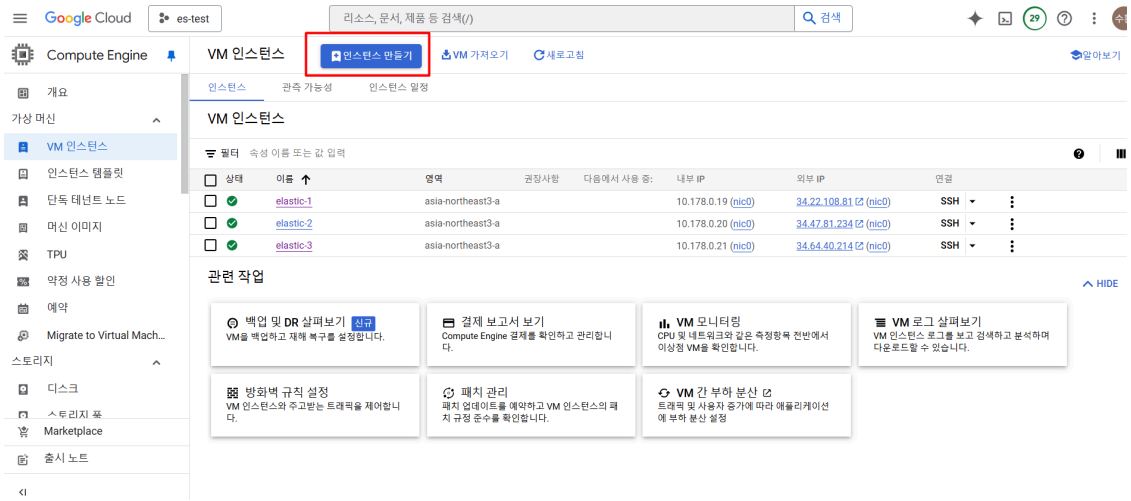


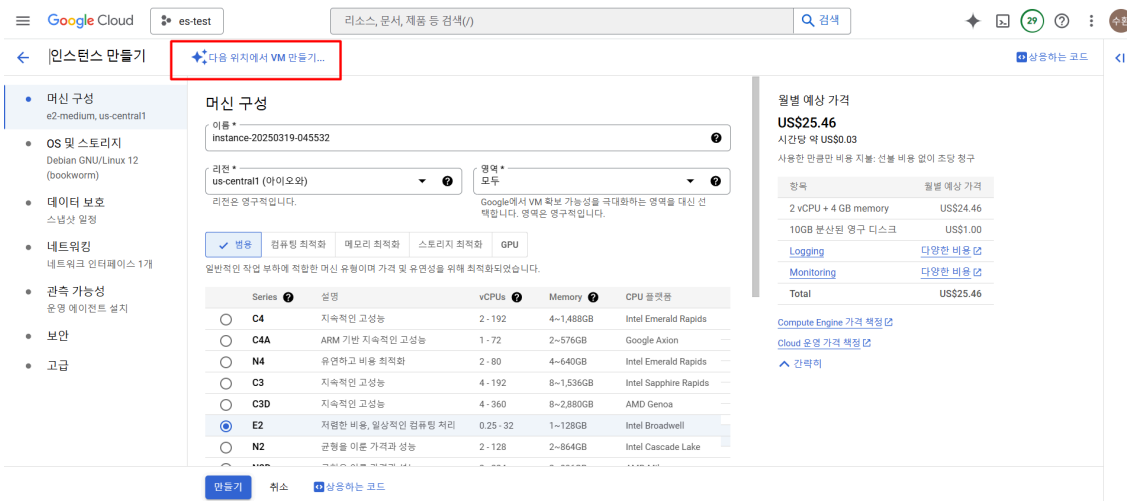
elastic-1 (제일 처음에 만든 인스턴스) 를 이미지로 만들어서 똑같이 2, 3을 만들기 위함

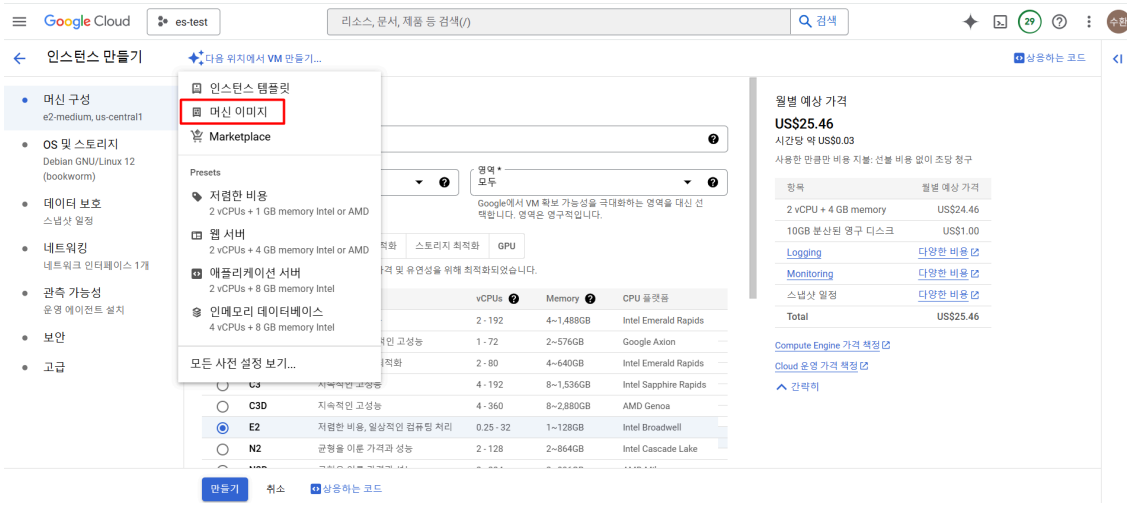


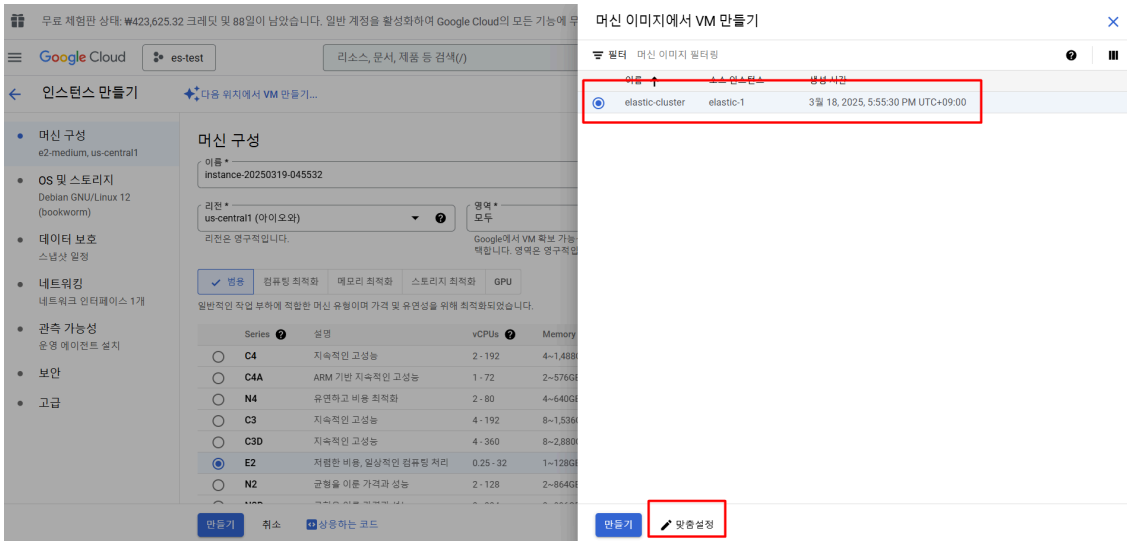


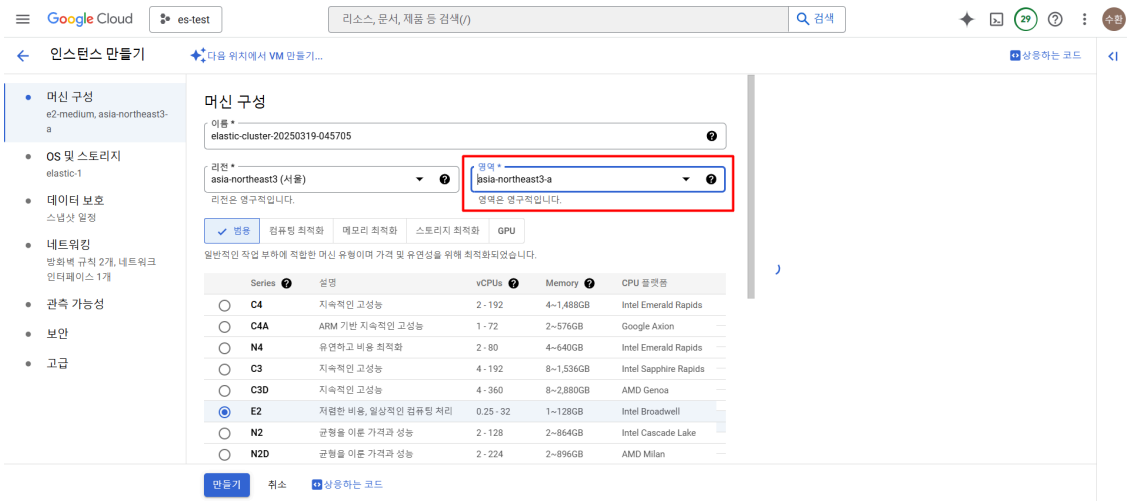
지역을 서울로 해주고 이미지를 만든다.



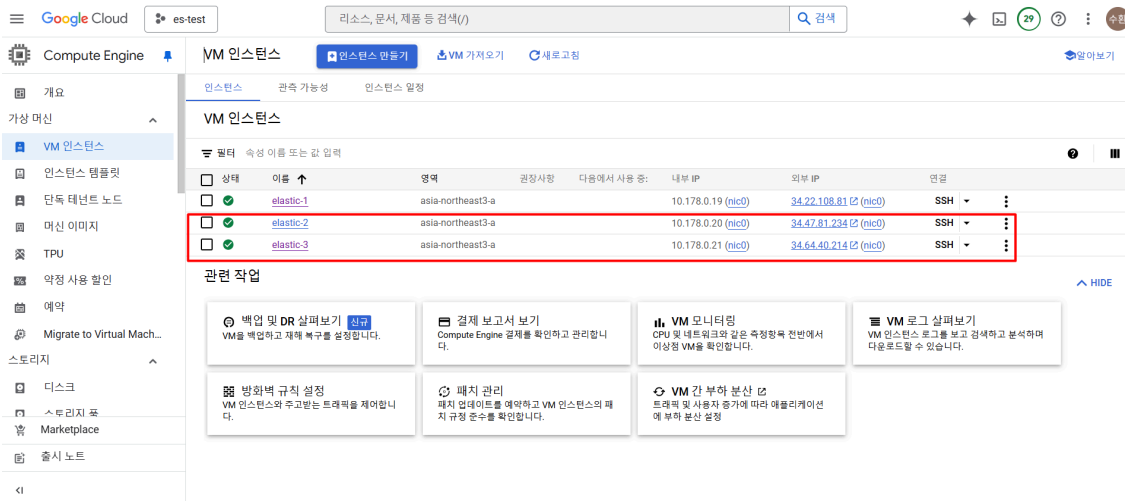








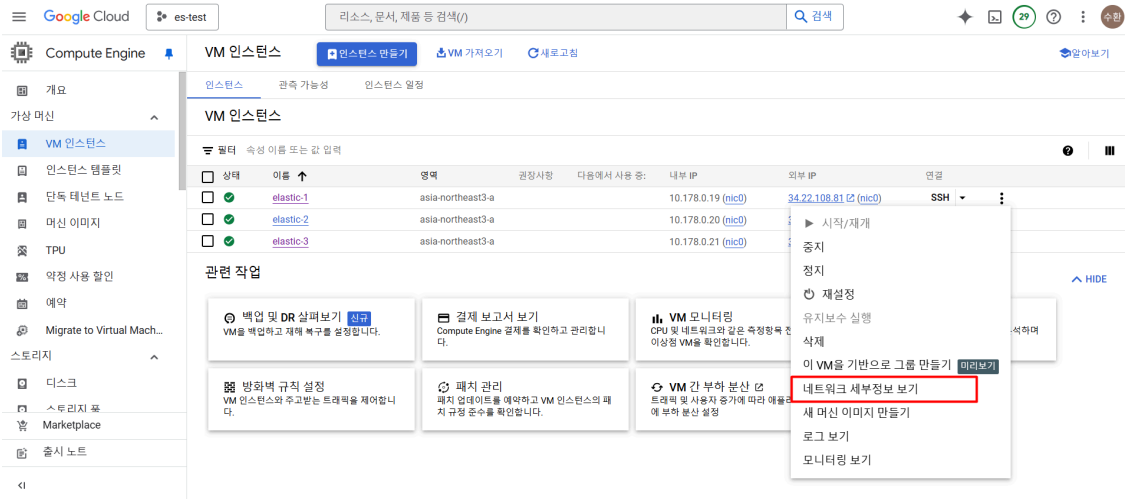
제일 처음 생성했던 인스턴스의 지역과 영역을 똑같이 해줘야 한다.

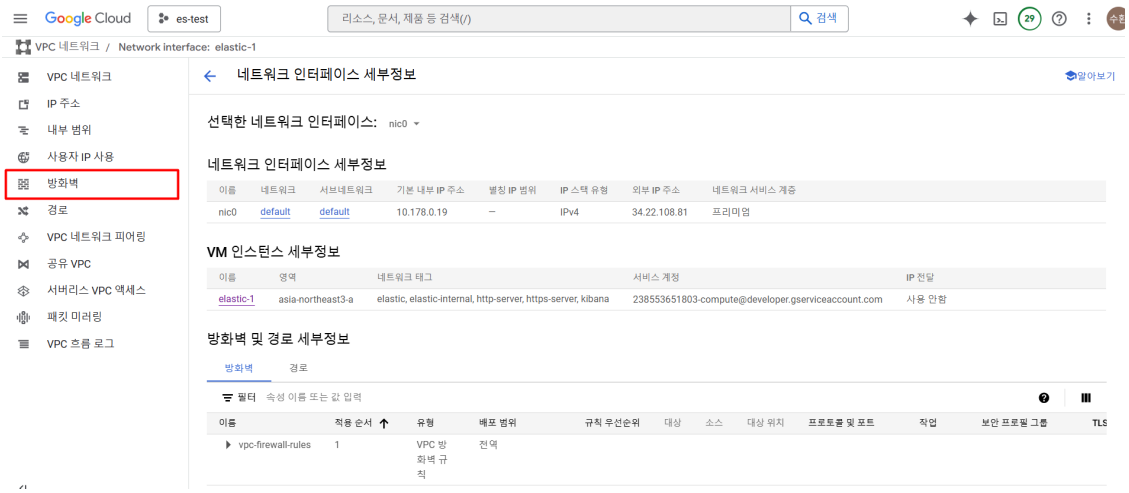


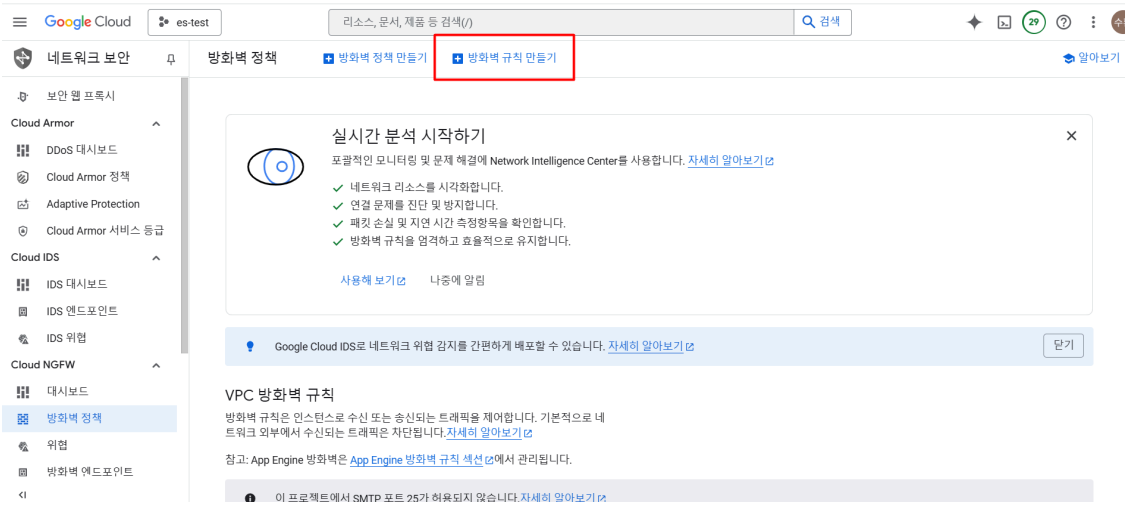
2, 3을 똑같은 방식으로 만든다.

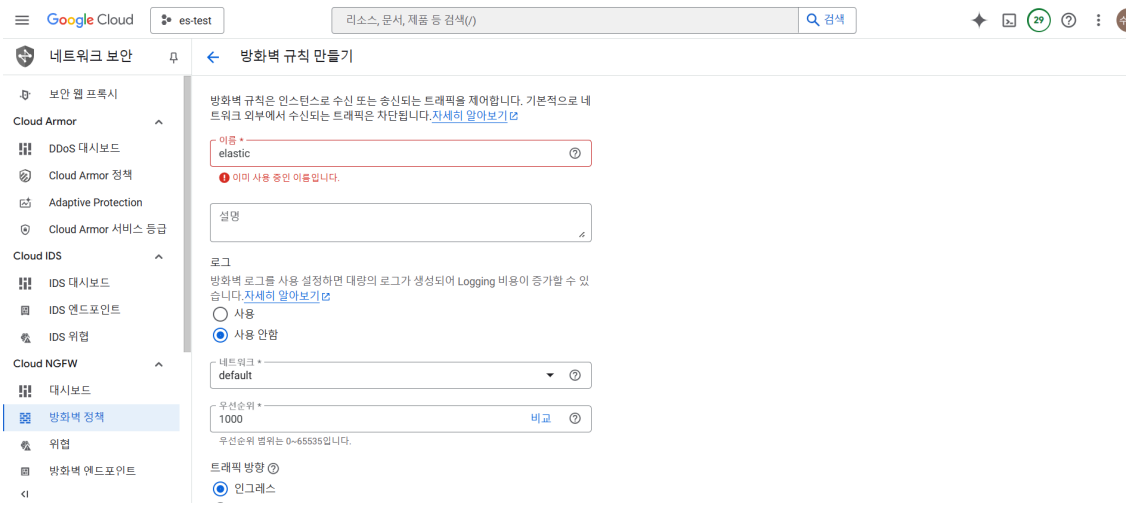
방화벽 설정

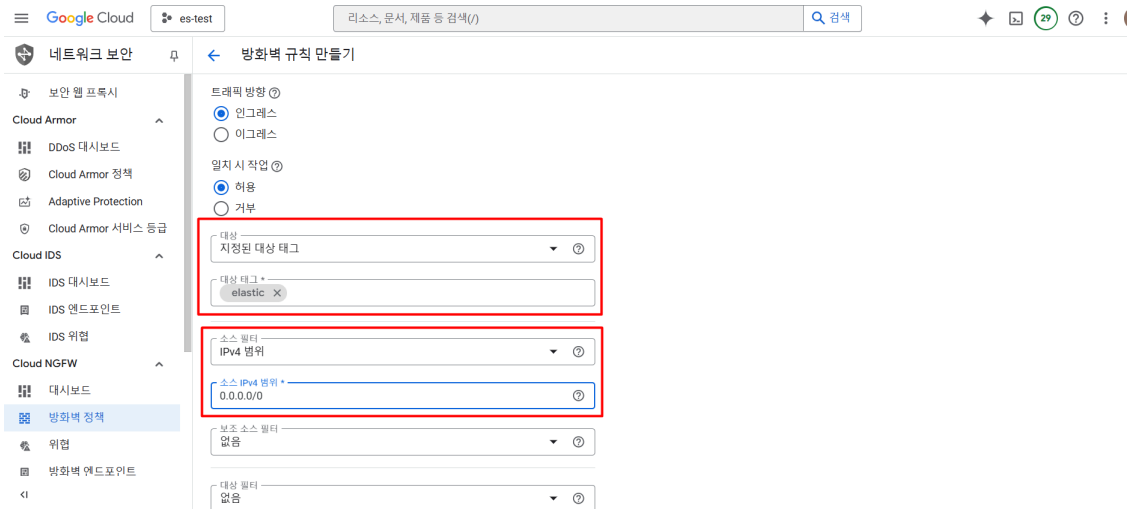
로컬에서 elasticsearch에 접근하게 하기 위한 방화벽 설정

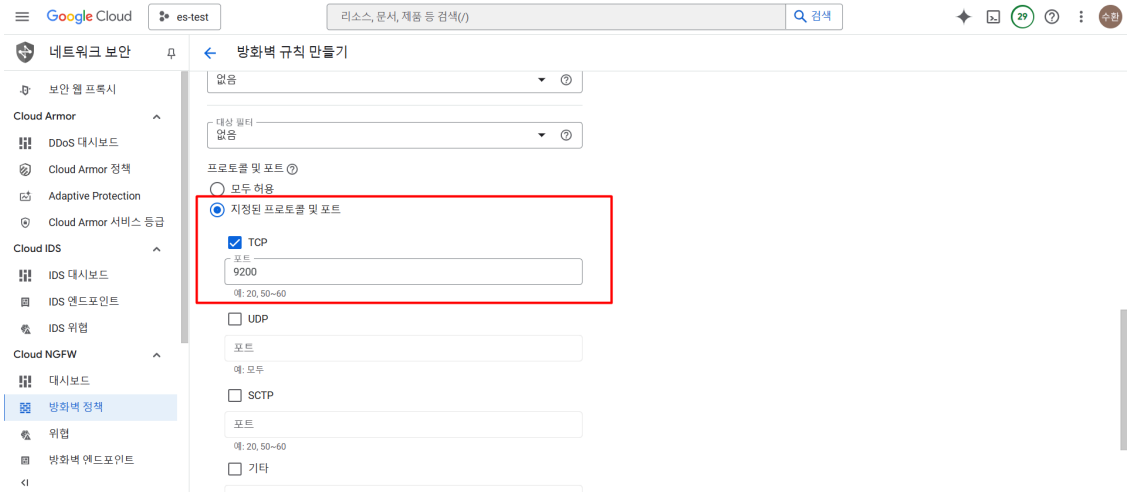



Google Cloud
es-test
리소스, 문서, 제품 등 검색(/)
검색
VPC 네트워크 / Network interface: elastic-1
VPC 네트워크
IP 주소
내부 범위
사용자 IP 사용
방화벽
경로
VPC 네트워크 피어링
공유 VPC
서버리스 VPC 액세스
패킷 미러링
VPC 흐름 로그
네트워크 인터페이스 세부정보
알아보기
선택한 네트워크 인터페이스: nic0
네트워크 인터페이스 세부정보
이름 네트워크 서브네트워크 기본 내부 IP 주소 별칭 IP 범위 IP 스택 유형 외부 IP 주소 네트워크 서비스 계층
nic0 default default 10.178.0.19 — IPv4 34.22.108.81 프리미엄
VM 인스턴스 세부정보
이름 영역 네트워크 태그 서비스 계정 IP 전달
elastic-1 asia-northeast3-a elastic, elastic-internal, http-server, https-server, kibana 238553651803-compute@developer.gserviceaccount.com 사용 안함
방화벽 및 경로 세부정보
방화벽
경로
필터 속성 이름 또는 값 입력
이름 적용 순서 유형 배포 범위 규칙 우선순위 대상 소스 대상 위치 프로토콜 및 포트 작업 보안 프로필 그룹 TLS
vpc-firewall-rules 1 VPC 방화벽 규칙 전역

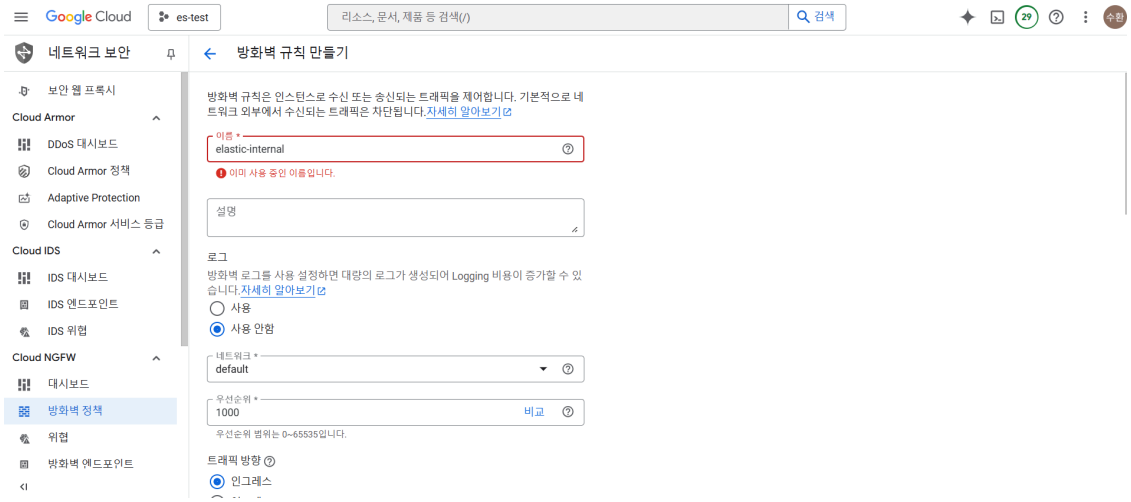

Google Cloud
es-test
리소스, 문서, 제품 등 검색(/)
검색
네트워크 보안
방화벽 정책
방화벽 정책 만들기
방화벽 규칙 만들기
알아보기
보안 웹 프록시
Cloud Armor
DDoS 대시보드
Cloud Armor 정책
Adaptive Protection
Cloud Armor 서비스 등급
Cloud IDS
IDS 대시보드
IDS 엔드포인트
IDS 위협
Cloud NGFW
대시보드
방화벽 정책
위협
방화벽 엔드포인트
실시간 분석 시작하기
포괄적인 모니터링 및 문제 해결에 Network Intelligence Center를 사용합니다. 자세히 알아보기
네트워크 리소스를 시각화합니다.
연결 문제를 진단 및 방지합니다.
패킷 손실 및 지연 시간 측정항목을 확인합니다.
방화벽 규칙을 염격하고 효율적으로 유지합니다.
사용해 보기
나중에 알림
Google Cloud IDS로 네트워크 위협 감지를 간편하게 배포할 수 있습니다. 자세히 알아보기
닫기
VPC 방화벽 규칙
방화벽 규칙은 인스턴스로 수신 또는 송신되는 트래픽을 제어합니다. 기본적으로 네트워크 외부에서 수신되는 트래픽은 차단됩니다.자세히 알아보기
참고: App Engine 방화벽은 App Engine 방화벽 규칙 섹션에서 관리됩니다.
이 프로젝트에서 SMTP 포트 25가 허용되지 않습니다.자세히 알아보기

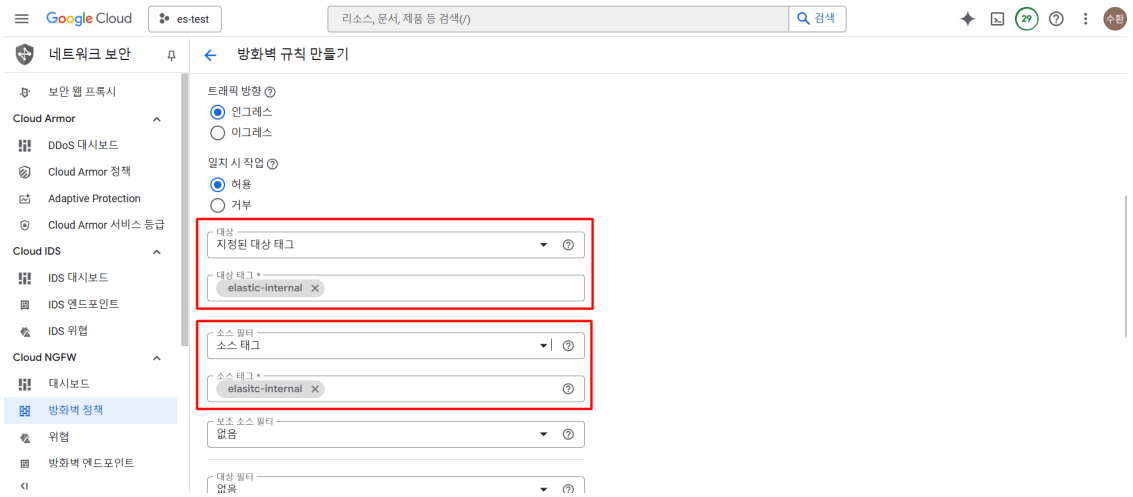

Google Cloud
es-test
리소스, 문서, 제품 등 검색(/)
검색
네트워크 보안
방화벽 규칙 만들기
보안 웹 프록시
Cloud Armor
DDoS 대시보드
Cloud Armor 정책
Adaptive Protection
Cloud Armor 서비스 등급
Cloud IDS
IDS 대시보드
IDS 엔드포인트
IDS 위협
Cloud NGFW
대시보드
방화벽 정책
위협
방화벽 엔드포인트
방화벽 규칙은 인스턴스로 수신 또는 송신되는 트래픽을 제어합니다. 기본적으로 네트워크 외부에서 수신되는 트래픽은 차단됩니다.자세히 알아보기
이름 *
elastic
이미 사용 중인 이름입니다.
설명
로그
방화벽 로그를 사용 설정하면 대량의 로그가 생성되어 Logging 비용이 증가할 수 있습니다.자세히 알아보기
사용
사용 안함
네트워크 *
default
우선순위 *
1000
비교
우선순위 범위는 0~65535입니다.
트래픽 방향
인그레스

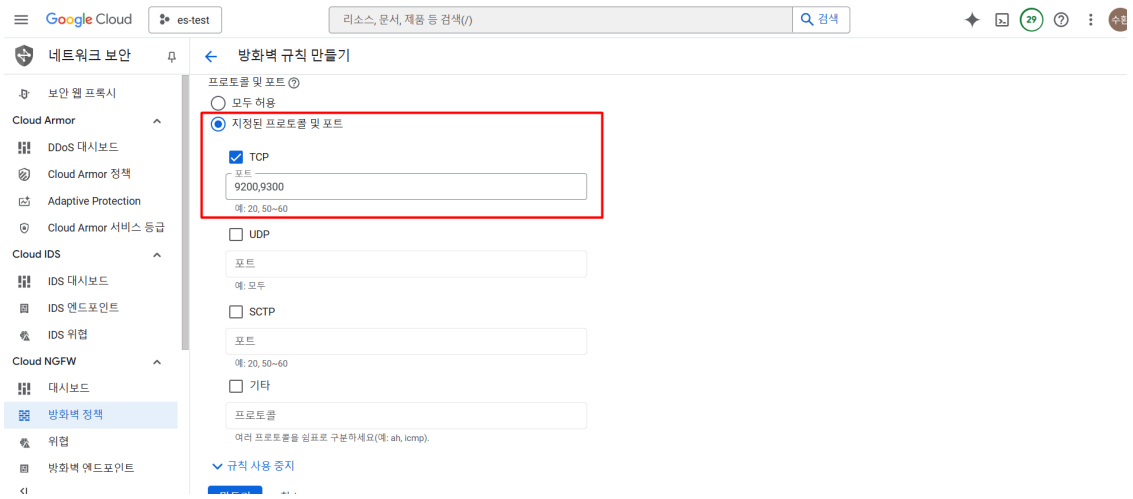




Elasticsearch 노드들 간 통신 방화벽 설정



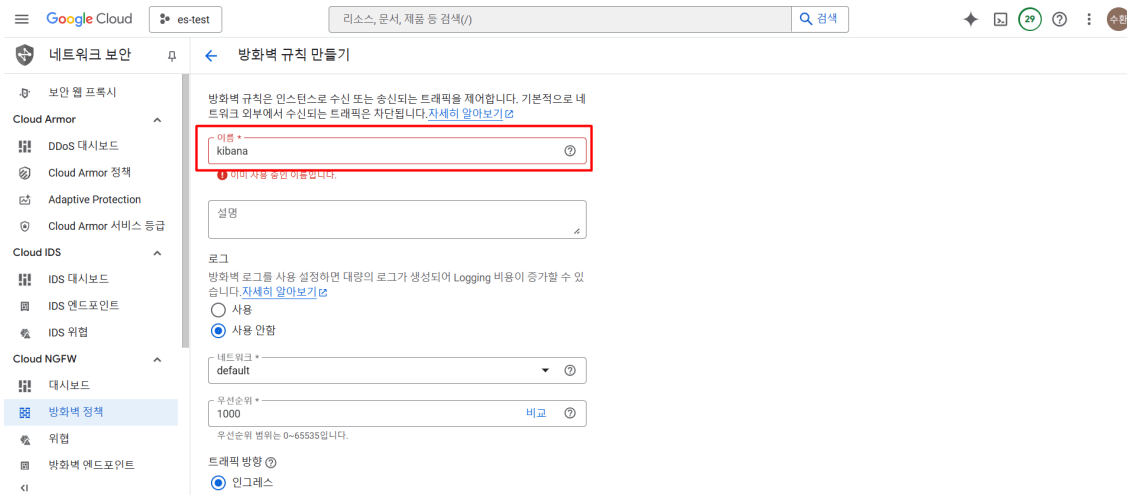


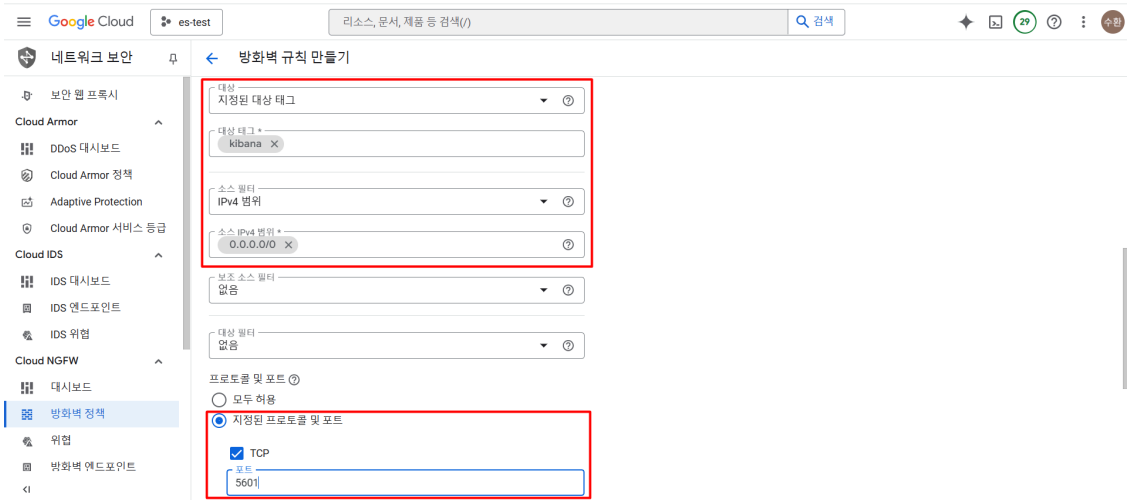


9200: REST API & HTTP 통신

9300: 클러스터 노드 간 통신

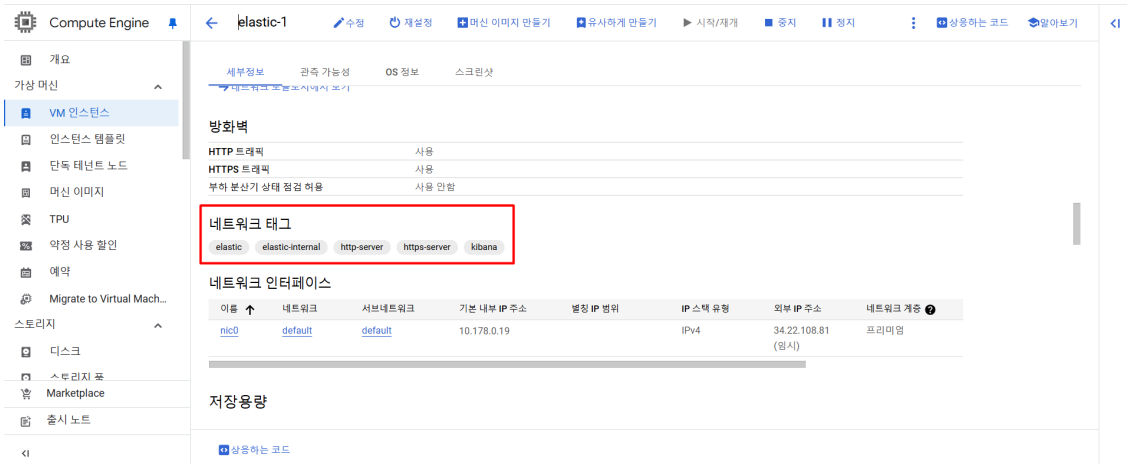
Kibana 방화벽 설정



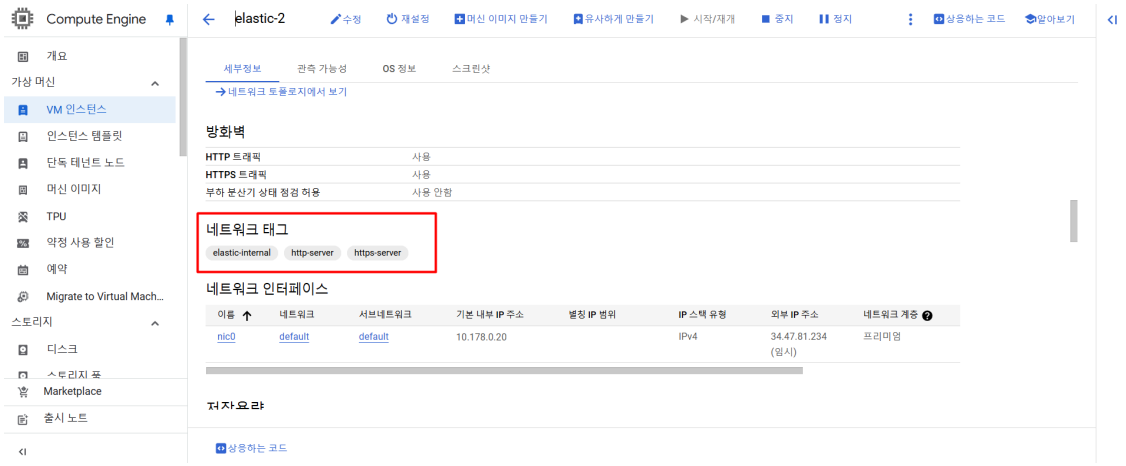


kibana는 기본적으로 5601 포트를 사용한다.

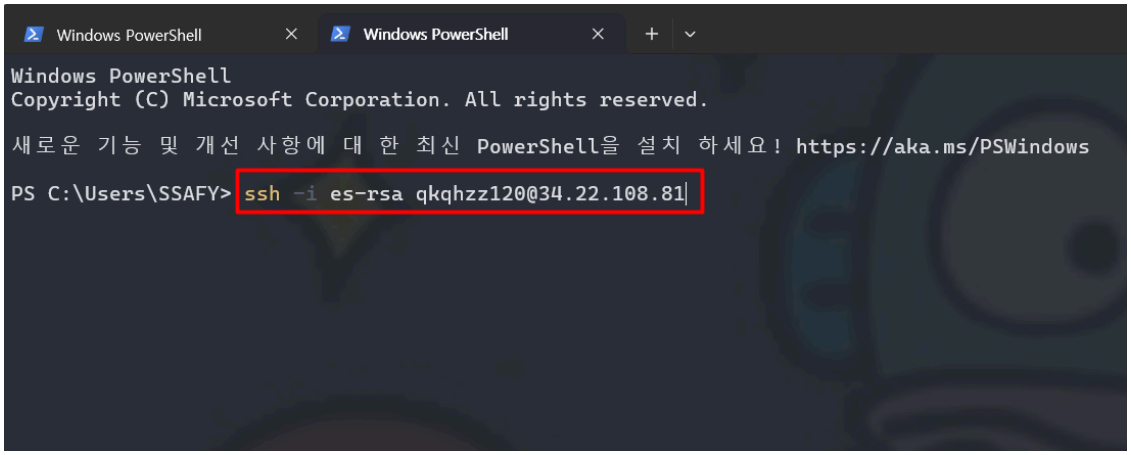
elastic-1 노드의 방화벽 설정



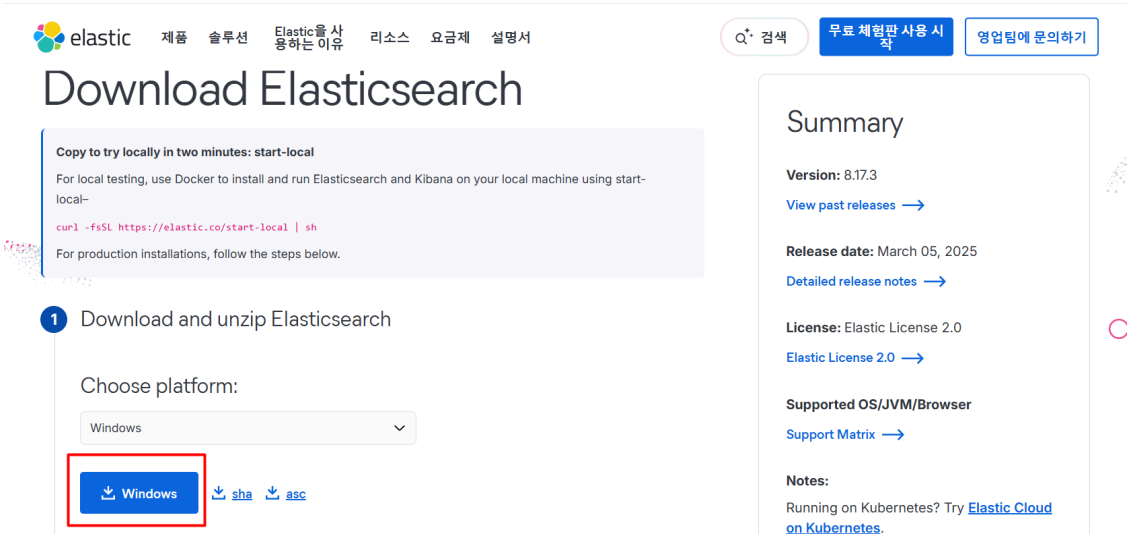
elastic-2, 3 노드의 방화벽 설정

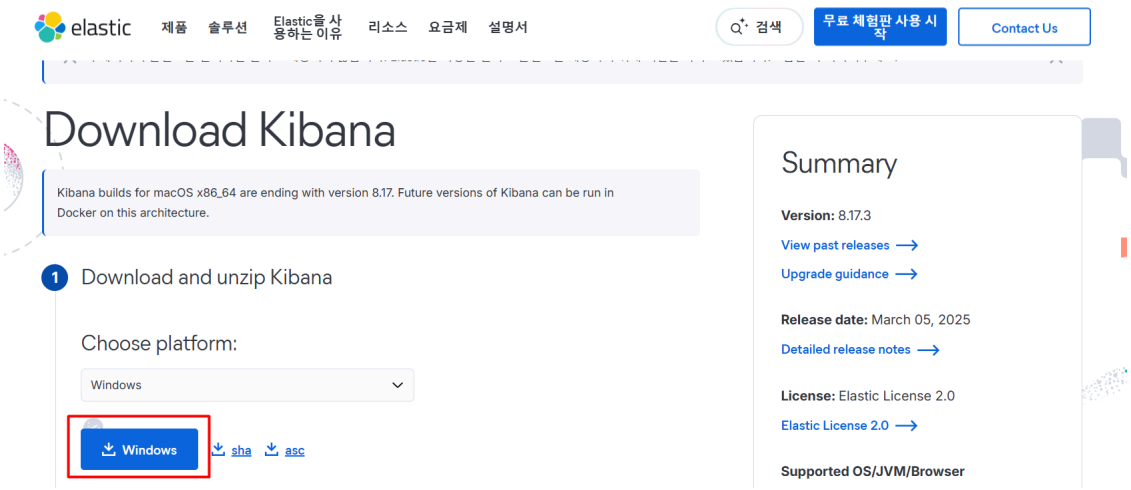


Elasticsearch 설치 및 실행



이전에 만들었던 인증키로 ssh 접속을 한다. (es-rsa는 내가 만든 인증키 이름이다)





우분투에서 파일 설치 링크로 설치하기 위해서 복사해서 설치한다.

`wget 링크` 로 설치하고 `tar xfz 다운받은파일명` 으로 압축을 푼다.

network host 설정과 부트스트랩 체크

```
qkqhzz120@elastic-1:~$ cd elasticsearch-8.17.3/
qkqhzz120@elastic-1:~/elasticsearch-8.17.3$ vi config/elasticsearch.yml
```

아래와 같이 설정해준다.

```
cluster.name: "es-cluster"
node.name: "node-1"
network.host: ["_local_", "_site_"]
discovery.seed_hosts: ["elastic-1", "elastic-2", "elastic-3"]
cluster.initial_master_nodes: ["node-1", "node-2", "node-3"]

# ======================== Elasticsearch Configuration =========
==================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure yo
u
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cl
uster.
```

```
#
# Please consult the documentation for further information on configuratio
n options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------
--
#
# Use a descriptive name for your cluster:
#
#cluster.name: my-application
#
# ------------------------------------ Node ------------------------------
----
#
# Use a descriptive name for the node:
#
#node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ----------------------------------- Paths ------------------------------
---
#
# Path to directory where to store the data (separate multiple locations by c
omma):
#
#path.data: /path/to/data
#
# Path to log files:
#
#path.logs: /path/to/logs
#
# ----------------------------------- Memory -----------------------------
-----
#
```

```
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
#network.host: 192.168.0.1
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
#http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
# For more information, consult the discovery and cluster formation module
```

```
documentation.
#
# ---------------------------------- Various -----------------------------------
#
# Allow wildcard deletion of indices:
#
#action.destructive_requires_name: false

#---------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 18-03-2025 08:26:29
#
# --------------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: true

xpack.security.enrollment.enabled: true

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
```

```
# Additional nodes can still join the cluster later
# cluster.initial_master_nodes: ["elastic-1"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0

#----------------------- END SECURITY AUTO CONFIGURATION ----------
---------------
```

위의 5줄이 추가해야할 부분이다.

인스턴스를 2개 더 만들었기 때문에 미리 3개를 하나의 클러스터로 묶는다고 생각하고 위와 같이 설정한 것이다.

elastic-2, elastic-3 인스턴스는 node.name 만 바꿔주면 된다. (똑같이 나머진 똑같이 설정해야 함)

```
qkqhzz120@elastic-1:~/elasticsearch-8.17.3$ vi config/jvm.options

# 아래 내용을 추가한다.
-Xms512m
-Xmx512m
```

```
sudo su
ulimit -n 65535
su qkqhzz120
```

```
sudo vi /etc/security/limits.conf

# 아래 내용을 추가한다.
qkqhzz120      -      nofile  65535
```

```
sudo sysctl -w vm.max_map_count=262144
```

💡 이 부분이 중요하다.

늘려주지 않으면 실행이 안되는 경우가 많다.

Elasticsearch 실행

```
qkqhzz120@elastic-1:~$ cd elasticsearch-8.17.3/
qkqhzz120@elastic-1:~/elasticsearch-8.17.3$ bin/elasticsearch
```

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------

✅ Elasticsearch security features have been automatically configured!
✅ Authentication is enabled and cluster connections are encrypted.

❌ Unable to auto-generate the password for the elastic built-in superuser.

ℹ️  HTTP CA certificate SHA-256 fingerprint:
  b57cac9cc8fc0cf9e95f7fee734c2450d6d4d128581ffc44268bbbab3e4f4
9aa

ℹ️  Configure Kibana to use this cluster:
• Run Kibana and click the configuration link in the terminal when Kibana st
arts.
• Copy the following enrollment token and paste it into Kibana in your brow
ser (valid for the next 30 minutes):
  eyJ2ZXIiOiI4LjE0LjAiLCJhZHliOlsiMTAuMTc4LjAuMTk6OTIwMCJdLCJmZ
3IiOiJiNTdjYWM5Y2M4ZmMwY2Y5ZTk1ZjdmZWU3MzRjMjQ1MGQ2ZDRkM
TI4NTgxZmZjNDQyNjhiYmJhYjNlNGY0OWFhIiwia2V5IjoiVWlkdnFaVUJxN2
9PeVhXXzlNLXY6c2hxNEtPZjFUQVdsNXdsMlBBSkVhQSJ9

ℹ️ Configure other nodes to join this cluster:
• Copy the following enrollment token and start new Elasticsearch nodes wi
```

```
th `bin/elasticsearch --enrollment-token <token>` (valid for the next 30 min
utes):
  eyJ2ZXIiOiI4LjE0LjAiLCJhZHIiOlsiMTAuMTc4LjAuMTk6OTIwMCJdLCJmZ
3IiOiJiNTdjYWM5Y2M4ZmMwY2Y5ZTk1ZjdmZWU3MzRjMjQ1MGQ2ZDRkM
TI4NTgxZmZjNDQyNjhiYmJhYjNlNGY0OWFhIiwia2V5IjoiVkNkdnFaVUJxN2
9PeVhXXzlNX1c6dzNJd3VMa2dTaTZQam9hQld3eVZ1ZyJ9

  If you're running in Docker, copy the enrollment token and run:
  `docker run -e "ENROLLMENT_TOKEN=<token>" docker.elastic.co/elastic
search/elasticsearch:8.17.3`
```

실행이 되면 위와 같이 나온다. 나는 설정하면서 `rm -rf data` 로 지웠다가 하면서 뭐가 잘못 됐었는지 위에 비밀번호 부분이 안나왔는데 아래 명령어로 비밀번호 생성했다

```
bin/elasticsearch-reset-password -u elastic -i
```

```
curl -XGET "https://34.47.81.234:9200/_cat/nodes" -u elastic:{생성된비밀번호} --insecure
```

생성된 비밀번호(위에는 X로 되어있지만 원래는 V표시 되어서 비밀번호가 나온다)를 넣으면 된다. 나는 명령어로 생성한 비밀번호로 했다.

💡 **참고사항**

비밀번호를 까먹었을 때

`rm -rf data` 를 해서 초기화 해서 재생성해야 한다. 하지만 여태 데이터가 모두 지워지기 때문에 안 좋은 방법이다. 그래서 무조건 어디에 기록해두자.

`bin/elasticsearch-setup-passwords auto` 자동으로 적절한 비밀번호를 생성해줌

`bin/elasticsearch-setup-passwords auto interactive` 내 멋대로 만들 수 있음

Kibana 설정 및 실행

```
vi config/kibana.yml
```

```
server.host: "elastic-1"
server.name: "my-kibana"
elasticsearch.hosts: ["https://elastic-1:9200"]
elasticsearch.username: "kibana_system"
elasticsearch.ssl.certificateAuthorities: ["/home/qkqhzz120/elasticsearch-8.17.3/config/certs/http_ca.crt"]
```

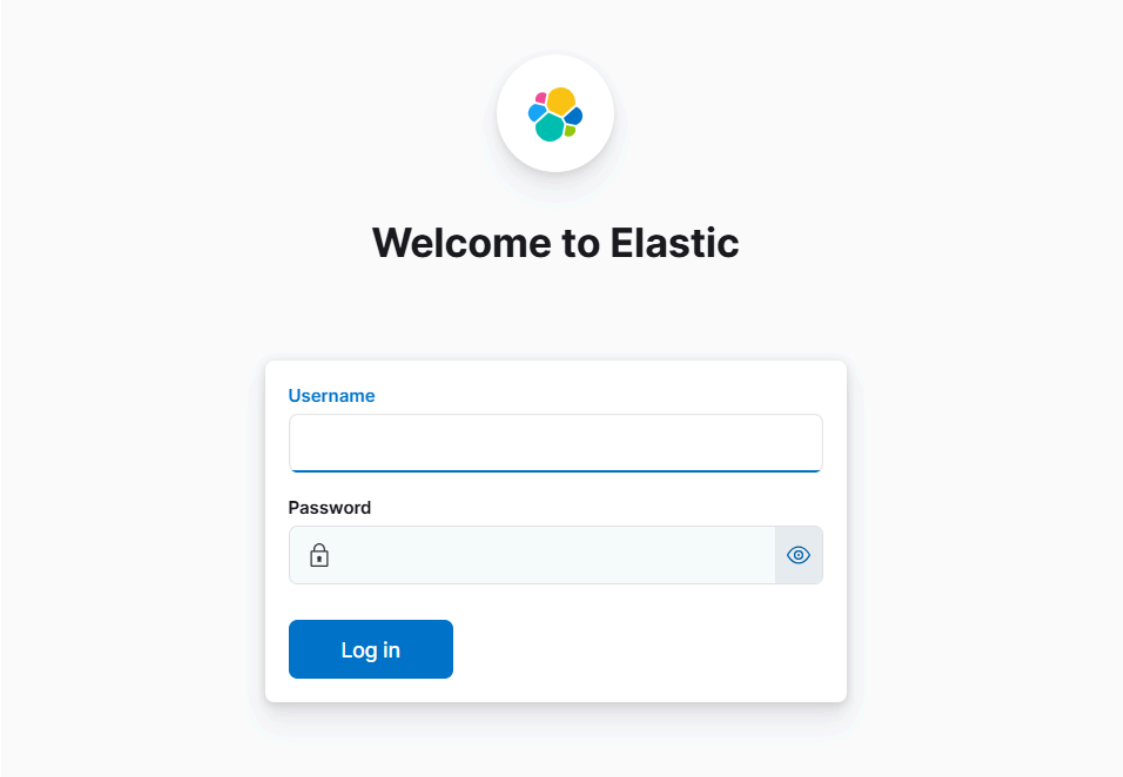
```
bin/kibana-keystore create
# 비밀번호는 kibana_system 의 비밀번호로 해야한다.
bin/kibana-keystore add elasticsearch.password
# 잘 생성 되었는지 확인
kibana-keystore list
```

```
# kibana에서 인증 코드를 적으라고 하면 아래 명령어를 실행해서 적자
bin/kibana-verification-code
```

```
find /home/qkqhzz120/elasticsearch-8.17.3/config/ -name "http_ca.crt"
```

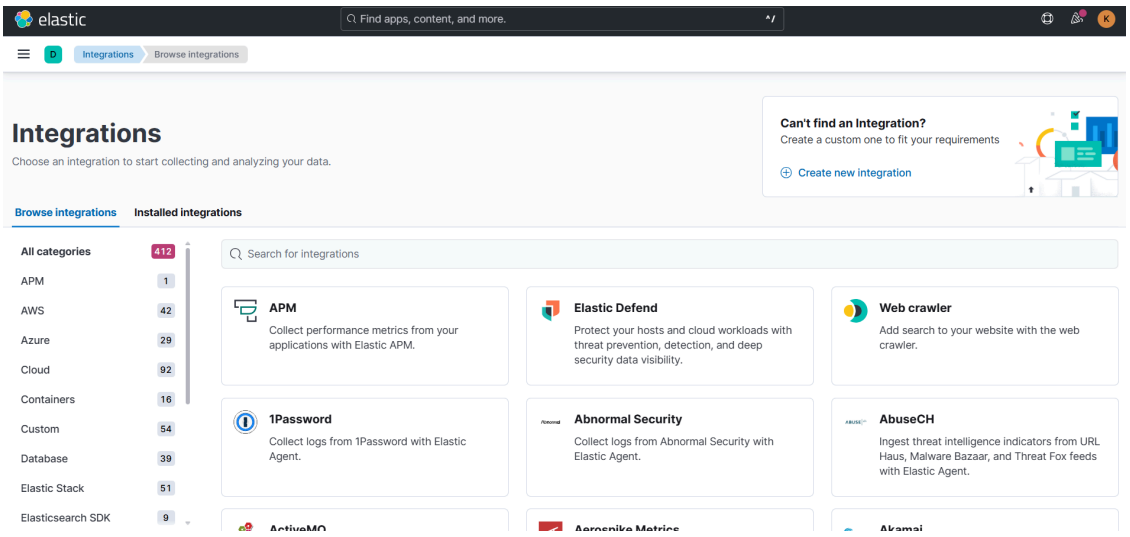
```
find / -name "*.p12" 2>/dev/null

find / -name "http_ca.crt" 2>/dev/null
```

```
curl -XGET https://{호스트이름 또는 내부아이피}:9200 -u elastic:{비밀번호} --insecure
```



초기 아이디: elastic

초기 비밀번호: 처음 생성된 비밀번호(위에 elastic:{비밀번호} 로 쓰인 부분)



Elasticsearch, Kibana 백그라운드(데몬) 실행

elasticsearch

```
echo './bin/elasticsearch -d -p es.pid' > start.sh
echo 'kill `cat es.pid`' > stop.sh
chmod 755 start.sh stop.sh

# 실행
./start.sh
```

kibana

kibana는 -d 옵션이 따로 없다. 백그라운드로 실행할 수 있는 여러가지 방법이 있다.

1. `bin/kibana &` 로 실행

- `ctrl + z` 로 눌러서 종료하지 않고 빠져나오기

2. systemd (시스템데몬)으로 실행

3. service로 실행

4. kibana home directory > src > cli > cli.js로 실행

`./node/bin/node ./src/cli/cli.js`

5. pm2로 실행

kibana가 배포된 node가 아닌 서버안에서 전역으로 사용할 수 있는 node.js를 설치해야 함

아무 node.js 버전을 사용하면 안된다.

딱 그 버전에 맞는 것으로 설치해야 한다.

- 알맞는 버전 정보 확인하는 방법
  - home directory에서 `ls -al`
  - `package.json` 을 통해 확인
- nvm install(github)
  - node는 이전 버전에서 되다가 안되고 그런 경우가 좀 있음 → 의존성에 좀 자주 걸림

    - nvm으로 여러 버전 사용하는 걸 추천
    - bashrc 에 자동추가가 되는건지 기존에 있었던 내용인지 모르겠지만 이 내용대로 하면 에러가 뜬다. nvm git의 설치부분 밑에 코드로 바꿔주자.
    - `source .bashrc`
    - `nvm install {맞는 노드버전}`
- pm2 install
    - `npm install pm2 -g`
- 실행
    - `pm2 start ./kibana-8.17.3/src/cli/cli.js --name kibana`
    - 확인
        - `pm2 list`
- 데몬으로 실행하기
    - kibana home dir 에서 `vi start.sh`
        - `pm2 start ~/kibana-8.17.3/src/cli/cli.js --name kibana` 저장
    - 똑같이 `vi stop.sh`
        - `pm2 stop kibana`
    - 스크립트 파일 실행하면 된다.

종료(pm2 말고 그냥 실행했을 경우)

- `kill` 명령어로 종료하기
    - `ps -ef | grep kibana` 로 하면 안나옴
    - `ps -ef | grep node` 로 찾아야 나옴
        - kibana는 node.js로 실행되고 있어서 그럼