환경 구성

수동 설치

- Ubuntu에 수동 설치
  - ✓ 사전 준비
    - sudo apt update && sudo apt upgrade -y
    - sudo apt install wget apt-transport-https gnupg -y

수동 설치

- Ubuntu에 수동 설치
  - ✓ Elastic Search 설치
    - Elastic 공식 GPG 키 및 저장소 추가
      - sudo apt update
      - sudo apt install wget gnupg2 apt-transport-https -y
      - # Elastic GPG 키 등록
      - wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --dearmor | sudo tee /usr/share/keyrings/elasticsearch-keyring.gpg > /dev/null
      - echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list
    - Elasticsearch 설치
      - sudo apt update
      - sudo apt install elasticsearch –y
    - Elasticsearch 서비스 실행
      - sudo systemctl enable elasticsearch
      - sudo systemctl start elasticsearch
    - Elasticsearch 비밀번호 생성
      - sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic

수동 설치

- ❖ Ubuntu에 수동 설치
  - ✓ Elastic Search 설치
    - Elastic Search 확인: curl -k https://localhost:9200 -u elastic:비밀번호


```
{
  "name" : "server",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "OmgIR7aYRtuWTrsbwTaB6w",
  "version" : {
    "number" : "8.19.3",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "1fde05a4d63448377eceb8fd3d51ce16ca3f02a9",
    "build_date" : "2025-08-26T02:35:34.366492370Z",
    "build_snapshot" : false,
    "lucene_version" : "9.12.2",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

수동 설치

- ❖ Ubuntu에 수동 설치
  - ✓ Log Stash 설치
    - sudo apt install logstash -y
    - sudo systemctl enable logstash
    - sudo systemctl start logstash
    - JDK 설치: sudo apt install openjdk-17-jdk -y
    - 확인: dpkg -l | grep logstash

      ii logstash 1:8.19.3-1 arm64 An extensible logging pipeline
    - PATH 수정: export PATH=$PATH:/usr/share/logstash/bin
    - 확인: logstash --version

      Using bundled JDK: /usr/share/logstash/jdk

      logstash 8.19.3

수동 설치

- ❖ Ubuntu에 수동 설치
  - ✓ Kibana 설치
    - sudo apt install kibana -y
    - sudo systemctl enable kibana
    - sudo systemctl start kibana
    - 확인: curl -I http://localhost:5601

```
HTTP/1.1 200 OK
x-content-type-options: nosniff
referrer-policy: strict-origin-when-cross-origin
permissions-policy: camera=(), display-capture=(), fullscreen=(self), geolocation=(),
microphone=(), web-share=()
cross-origin-opener-policy: same-origin
content-security-policy: script-src 'report-sample' 'self'; worker-src 'report-sample'
'self' blob:; style-src 'report-sample' 'self' 'unsafe-inline'
content-security-policy-report-only: form-action 'report-sample' 'self'; object-src
'report-sample' 'none'
kbn-name: elk
```

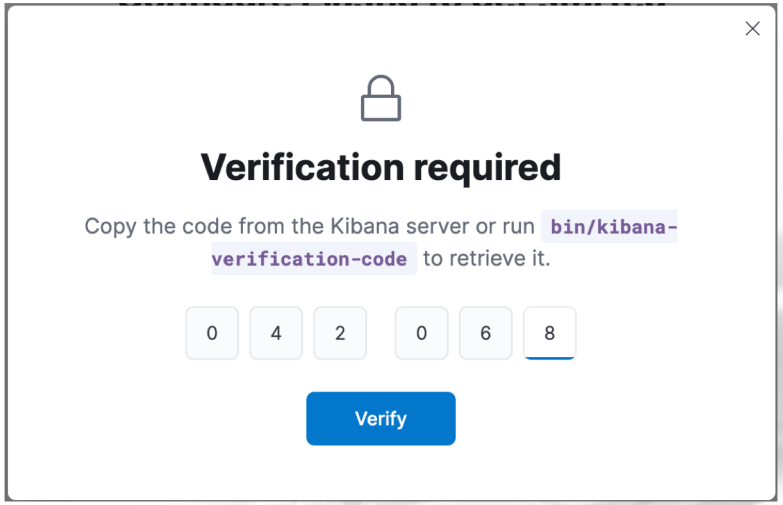
수동 설치

- ❖ Ubuntu에 수동 설치
  - ✓ 방화벽 설정
    - sudo ufw allow 5601/tcp # Kibana
    - sudo ufw allow 9200/tcp # Elasticsearch
    - sudo ufw allow 5044/tcp # Logstash Beats 입력

수동 설치

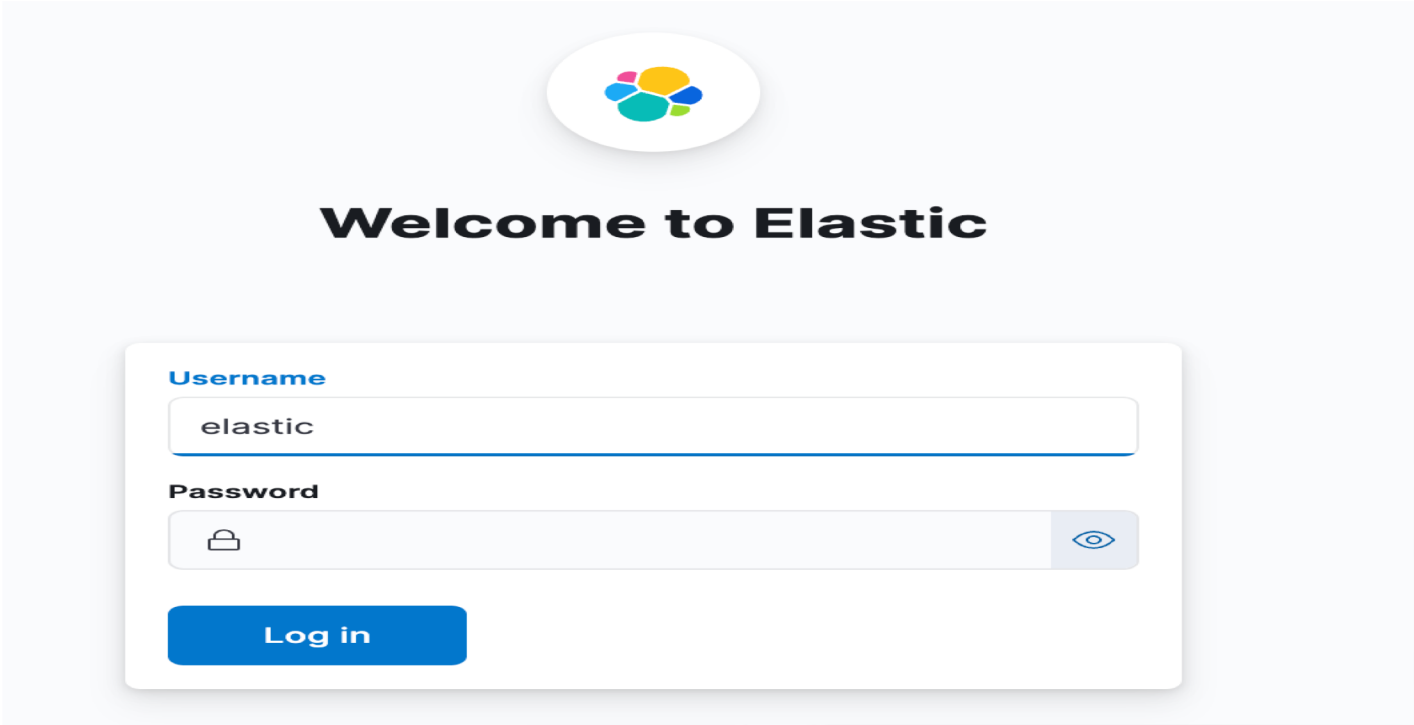
- ❖ Ubuntu에 수동 설치
  - ✓ Kibana 설치
    - 외부 접속 허용: /etc/kibana/kibana.yml 파일에 추가

      server.host: "0.0.0.0"
    - 토큰 생성: sudo /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
    - Kibana 웹 화면(http://IP:5601)에서 토큰 등록
      - ◆ 검증 코드 확인: sudo /usr/share/kibana/bin/kibana-verification-code



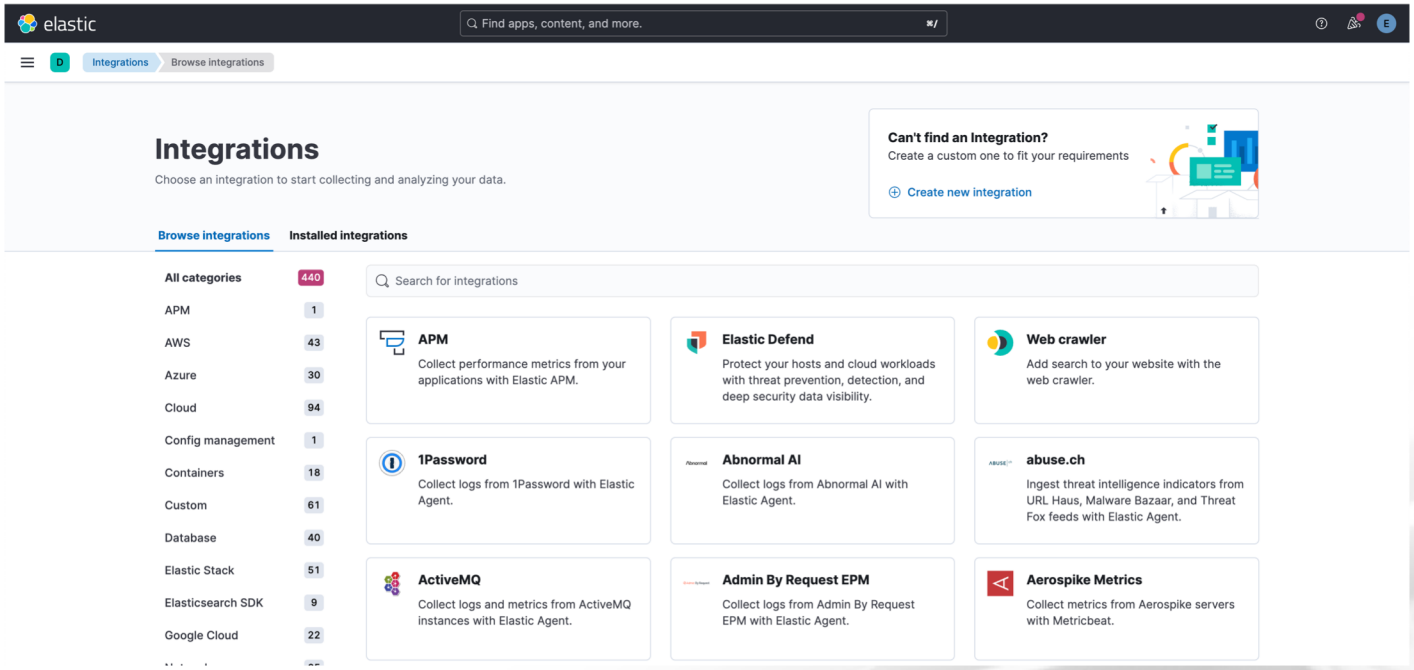
수동 설치

- ❖ Ubuntu에 수동 설치
  - ✓ Kibana 설치
    - ● 로그인



수동 설치

- ❖ Ubuntu에 수동 설치
  - ✓ Kibana 설치
    - ● 확인

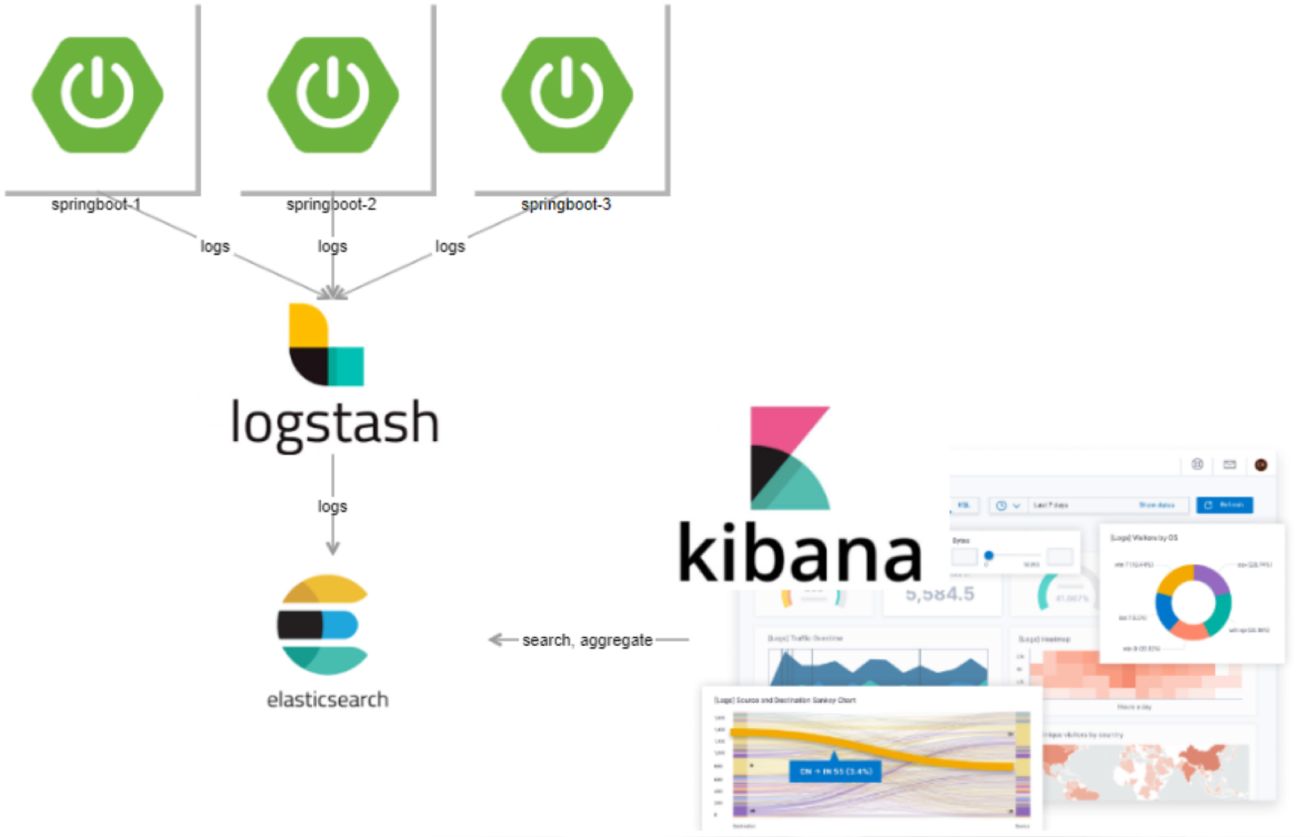


수동 설치

- Ubuntu에 수동 설치
  - ✓ 비트 설치
    - sudo apt install filebeat
    - sudo apt install metricbeat

Spring Boot Log

❖ 개요



Spring Boot Log

- ElasticSearch 설정
  - ✓ ElasticSearch 설정
    - 설정 파일 수정 /etc/elasticsearch/elasticsearch.yml

      xpack.security.enabled: false
    - 서비스 재시작: sudo systemctl restart elasticsearch

Spring Boot Log

- ElasticSearch 설정
  - ✓ logstash 설정 파일: /etc/logstash/conf.d/logstash.conf

```
input {
  # Spring Boot 애플리케이션으로부터 TCP를 통해 로그를 받음
  tcp {
    port => 5044
    codec => json_lines
  }
}

filter {
  # 필요한 경우 로그를 가공하는 로직을 추가합니다.
  # 예: 타임스탬프 변환이나 특정 필드 제거
  mutate {
    remove_field => ["port", "host"]
  }
}
```

Spring Boot Log

- Logstash 설정
  - ✓ logstash 설정 파일: /etc/logstash/conf.d/logstash.conf

```
output {
  # 가공된 로그를 Elasticsearch로 전송
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "springboot-elk"
    ssl => false

    # Elasticsearch 보안 설정이 되어 있는 경우 계정 정보 입력
    #user => "elastic"
    #password => "8j05aJ4bncEE8HXzBmHx"
  }

  # 디버깅을 위해 콘솔에도 출력 (설치 초기 권장)
  stdout { codec => rubydebug }
}
```

Spring Boot Log

- 스프링 프로젝트 생성 및 실행 – 로그 생성
  - ✓ spring-web, lombok 의존성을 추가한 프로젝트 생성

Spring Boot Log

❖ 스프링 프로젝트 생성 및 실행 – 로그 생성

✓ 프로젝트에 Controller 추가 - LogController

```
import lombok.extern.slf4j.Slf4j;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

@RestController
@Slf4j
public class LogController {
    @GetMapping("/member/{name}")
    public String sayHello(@PathVariable String name) {
        log.info("logs-1 " + name);
        return "Hello " + name;
    }

    @GetMapping("/board/{name}")
    public String sayGoodBye(@PathVariable String name) {
        log.info("logs-2 " + name);
        return "Board " + name;
    }
}
```

Spring Boot Log

❖ 스프링 프로젝트 생성 및 실행 – 로그 생성

✓ 프로젝트의 Application 클래스 수정

```
import lombok.extern.slf4j.Slf4j;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@Slf4j
@SpringBootApplication
public class ElkApplication implements ApplicationRunner {

    public static void main(String[] args) {
        SpringApplication.run(ElkApplication.class, args);
    }
    @Override
    public void run(ApplicationArguments args) throws Exception {
        int i = 0;
        while(true) {
            Thread.sleep(1000);
            log.info("springboot-elk-01 :: {}", ++i);
        }
    }
}
```

Spring Boot Log

- 스프링 프로젝트 생성 및 실행 – 로그 생성
  - ✓ 로그를 LogStash로 내보내도록 build.gradle 파일에 의존성 추가

    implementation 'net.logstash.logback:logstash-logback-encoder:7.4'

Spring Boot Log

- 스프링 프로젝트 생성 및 실행 – 로그 생성
  - ✓ 로그를 LogStash로 내보내도록 resources 디렉토리에 logback.xml 파일을 추가하고 작성

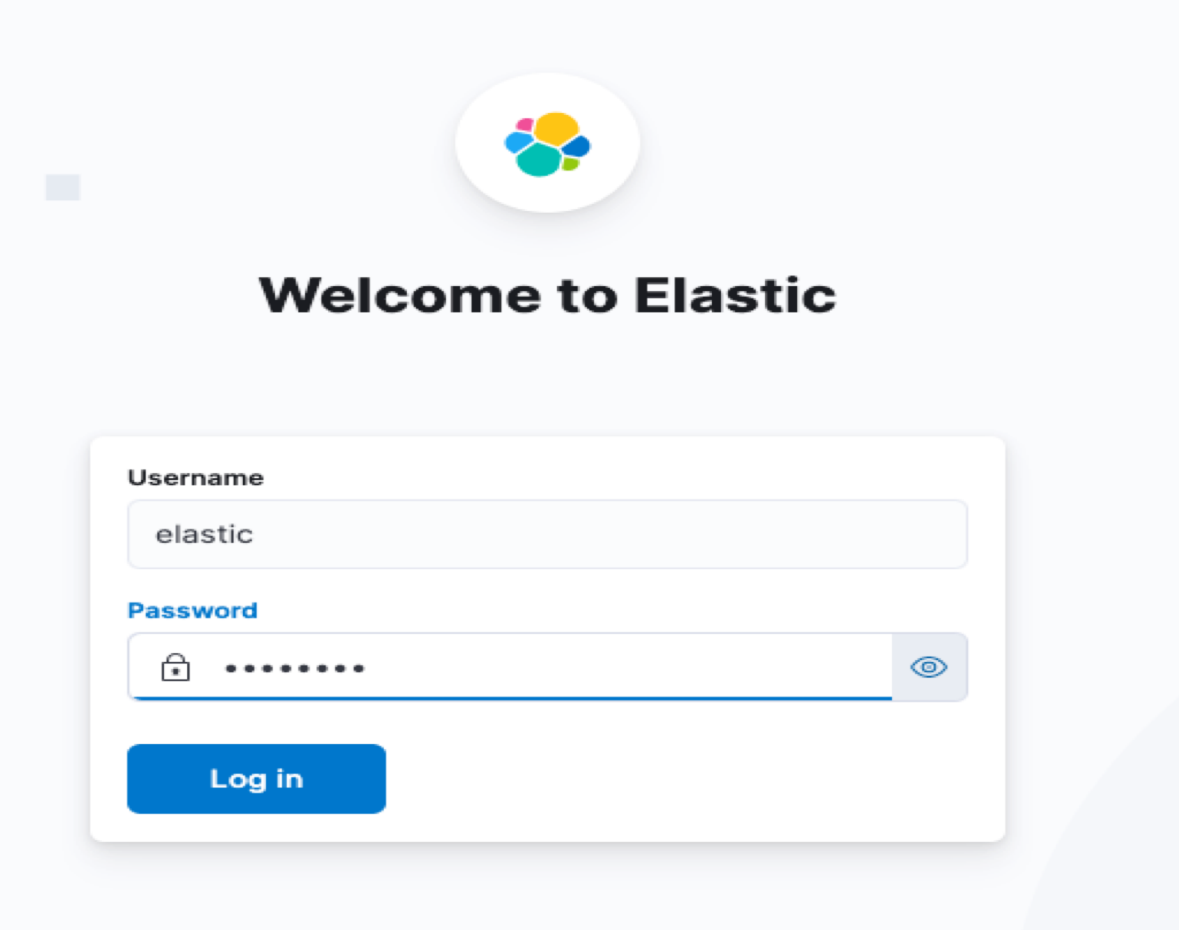
```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
        <encoder class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">
            <pattern>%-5level %d{HH:mm:ss.SSS} [%thread] %logger{36} -
 %msg%n</pattern>
        </encoder>
    </appender>
    <appender name="stash"
    class="net.logstash.logback.appender.LogstashTcpSocketAppender">
        <destination>192.168.202.18:5044</destination>
        <!-- encoder is required -->
        <encoder class="net.logstash.logback.encoder.LogstashEncoder" />
    </appender>
    <root level="INFO">
        <appender-ref ref="console"/>
        <appender-ref ref="stash"/>
    </root>
</configuration>
```

Spring Boot Log

- Spring Boot Log
  - ✓ 서버 실행 후 로그 생성
    - 요청을 여러 개 전송
  - ✓ ElasticSearch에서 로그 전송 확인(Elastic Search 가 제대로 동작 중인지 확인): sudo tail -f /var/log/elasticsearch/elasticsearch.log
  - ✓ LogStash 로그 확인: sudo tail -f /var/log/logstash/logstash-plain.log

Spring Boot Log

- Spring Boot Log
  - ✓ 브라우저에서 http://localhost:5601 로 접속 후 로그인(elastic/changeme)



Spring Boot Log

- Spring Boot Log
  - ✓ 브라우저에서 http://localhost:5601 로 접속 후 로그인(elastic/changeme)
    - kibana 로그인 > Stack Management