Lab8. Kafka UI 설치하기

※UI for Apache Kafka

-https://github.com/provectus/kafka-ui

1. API 서버 역할의 EC2 인스턴스 생성

1)인스턴스 목록에서 [인스턴스 시작] 버튼 클릭

2)[인스턴스 시작] 페이지에서

-[이름] : {계정}-kafka-ui-ec2

-[애플리케이션 및 OS 이미지] > [Quick Start] > Ubuntu

-Ubuntu Server 22.04 LTS (HVM), SSD Volume Type, 64비트(x86)

-[인스턴스 유형] : t2.micro

-[키 페어(로그인)] > [새 키 페어 생성]

-[키 페어 생성] 창에서,

--[키 페어 이름] : {계정}-kafka-ui-ec2-key

--나머지는 기본값 그대로 사용

--[키 페어 생성] 버튼 클릭

--적당한 위치에 pem 파일 다운로드

-[네트워크 설정] > [편집] 버튼 클릭

-[VPC] : {계정}-datalake-vpc

-[서브넷] : {계정}-datalake-subnet-2a

-[퍼블릭 IP 자동 할당] : 활성화

-[방화벽(보안 그룹)] : 보안 그룹 생성

-[보안 그룹 이름] : {계정}-kafka-ui-sg

-[설명] : Security group for Kafka UI Server

-[스토리지 구성] : 30 GiB, gp2

-[인스턴스 시작] 버튼 클릭

2. UI Server 사용 준비

1)방금 생성한 API EC2 서버의 [상태 검사]가 "2/2개 검사 통과"가 되면 Tabby를 통해 SSH 연결한다.

-Tabby 설정창에서 [New profile] 버튼 클릭

-[Name] : Kafka-ui-server

-[Group] : kafka

-[Host] : {계정}-kafka-ui-ec2의 퍼블릭 IPv4 DNS 값

-[Username] : ubuntu

-[Authentication method] : Key

-[Private keys] > [Add a private key] 버튼 클릭하여 {계정}-kafka-ui-ec2-key.pem 파일 연결

-[Save] 버튼 클릭

-[Profiles] 목록에서 "kafka-ui-server" 연결

2)API 서버에 필요한 Docker 설치 전

```
$ sudo apt update
```

-Set up the repository

```
$ sudo apt-get install ca-certificates curl gnupg
```

-Add Docker's official GPG key

```
$ sudo install -m 0755 -d /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

-Set up the stable repository

```
echo "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

3)Install Docker Engine

-Repository update

```
$ sudo apt update
```

-Install latest version of Docker engine and containerd

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

-Docker Service 상태 확인

```
$ sudo systemctl status docker
```

-hello-world Container Image 실행하기

```
$ sudo docker run hello-world
```

4)Docker-compose 설치

```
$ sudo apt install docker-compose
```

3. kafka-ui 설치하기

1)kafka-ui 디렉토리 생성

```
$ mkdir kafka-ui
```

2)해당 디렉토리로 이동

```
$ cd kafka-ui
```

3)Docker image pull

```
$ sudo docker pull provectuslabs/kafka-ui
```

4)docker-compose.yml 파일 생성
$ nano docker-compose.yml

4. kafka-ui 실행하기
1)Kafka UI Server의 보안그룹에 8080 포트 인바운드 규칙 추가
-{계정}-kafka-ui-ec2 인스턴스의 상세페이지에서 [보안] 탭 클릭
-[보안 그룹]에 있는 {계정}-kafka-ui-sg 링크 클릭
-{계정}-kafka-ui-sg 페이지에서 [인바운드 규칙] 탭 선택
-[인바운드 규칙 편집] 버튼 클릭
-[인바운드 규칙 편집] 페이지에서 [규칙 추가] 버튼 클릭
-[유형] : 사용자 지정 TCP
-[포트 범위] : 8080
-[소스] : Anywhere IPv4(0.0.0.0/0)
-[규칙 저장] 버튼 클릭

2)docker-compose로 kafka-ui 실행하기
$ sudo docker-compose up kafka-ui
Creating network "kafka-ui_default" with the default driver
Creating kafka-ui ... done
Attaching to kafka-ui
kafka-ui | 07:16:59,520 |-INFO in ch.qos.logback.classic.LoggerContext[default] - This is logback-classic version 1.4.6
kafka-ui | 07:16:59,658 |-INFO in ch.qos.logback.classic.LoggerContext[default] - Could NOT find resource [logback-test.xml]
kafka-ui | 07:16:59,666 |-INFO in ch.qos.logback.classic.LoggerContext[default] - Could NOT find resource [logback.xml]
kafka-ui | 07:16:59,683 |-INFO in ch.qos.logback.classic.BasicConfigurator@52e6fdee - Setting up default configuration.
kafka-ui | 07:17:01,399 |-INFO in ch.qos.logback.core.joran.spi.ConfigurationWatchList@6c80d78a - URL [jar:file:/kafka-ui-api.jar!/BOOT-INF/classes!/logback-spring.xml] is not of type file
kafka-ui | 07:17:01,600 |-INFO in ch.qos.logback.core.model.processor.AppenderModelHandler - Processing appender named [STDOUT]
kafka-ui | 07:17:01,600 |-INFO in ch.qos.logback.core.model.processor.AppenderModelHandler - About to instantiate appender of type [ch.qos.logback.core.ConsoleAppender]
kafka-ui | 07:17:01,657 |-WARN in ch.qos.logback.core.ConsoleAppender[STDOUT] - This appender no longer admits a layout as a sub-component, set an encoder instead.
kafka-ui | 07:17:01,657 |-WARN in ch.qos.logback.core.ConsoleAppender[STDOUT] - To ensure compatibility, wrapping your layout in LayoutWrappingEncoder.
kafka-ui | 07:17:01,657 |-WARN in ch.qos.logback.core.ConsoleAppender[STDOUT] - See also http://logback.qos.ch/codes.html#layoutInsteadOfEncoder for details
kafka-ui | 07:17:01,662 |-INFO in ch.qos.logback.classic.model.processor.RootLoggerModelHandler - Setting level of ROOT logger to INFO
kafka-ui | 07:17:01,663 |-INFO in ch.qos.logback.classic.jul.LevelChangePropagator@62150f9e - Propagating INFO level on Logger[ROOT] onto the JUL framework
kafka-ui | 07:17:01,664 |-INFO in ch.qos.logback.core.model.processor.AppenderRefModelHandler - Attaching appender named [STDOUT] to Logger[ROOT]
kafka-ui | 07:17:01,664 |-INFO in ch.qos.logback.core.model.processor.DefaultProcessor@1a451d4d - End of configuration.
kafka-ui | 07:17:01,665 |-INFO in org.springframework.boot.logging.logback.SpringBootJoranConfigurator@7fa98a66 - Registering current configuration as safe fallback point
kafka-ui |
kafka-ui | _ _ ____ __ _ _ _ __ __ _
kafka-ui | | | | |_ _| / _|___ _ _ /_\ _ _ __ __ _ __| |_ ___ | |/ /__ _ / _| |_____
kafka-ui | | |_| || | | _/ _ | '_| / _ \| '_ / _` / _| ' \/ -_) | ' </ _` | _| / / _`|
kafka-ui | ___/|___| |_| ___|_| /_/ _| .____,___|_||____| |_|___,_|_| |____,|
kafka-ui | |_|
kafka-ui |
kafka-ui | 2023-05-10 07:17:02,018 INFO [main] c.p.k.u.KafkaUiApplication: Starting KafkaUiApplication using Java 17.0.6 with PID 1 (/kafka-ui-api.jar started by kafkaui in /)
kafka-ui | 2023-05-10 07:17:02,021 DEBUG [main] c.p.k.u.KafkaUiApplication: Running with Spring Boot v3.0.5, Spring v6.0.7
kafka-ui | 2023-05-10 07:17:02,022 INFO [main] c.p.k.u.KafkaUiApplication: No active profile set, falling back to 1 default profile: "default"
kafka-ui | 2023-05-10 07:17:09,607 DEBUG [main] c.p.k.u.s.SerdesInitializer: Configuring serdes for cluster henry-msk-cluster
kafka-ui | 2023-05-10 07:17:11,107 INFO [main] o.s.b.a.e.w.EndpointLinksResolver: Exposing 2 endpoint(s) beneath base path '/actuator'
kafka-ui | 2023-05-10 07:17:11,422 INFO [main] o.h.v.i.u.Version: HV000001: Hibernate Validator 8.0.0.Final
kafka-ui | 2023-05-10 07:17:11,946 INFO [main] o.s.b.a.s.r.ReactiveUserDetailsServiceAutoConfiguration:
kafka-ui |
kafka-ui | Using generated security password: 4f873927-738a-4686-93f9-8b4517296e25
kafka-ui |
kafka-ui | 2023-05-10 07:17:12,341 WARN [main] c.p.k.u.c.a.DisabledAuthSecurityConfig: Authentication is disabled. Access will be unrestricted.
kafka-ui | 2023-05-10 07:17:13,577 INFO [main] o.s.b.w.e.n.NettyWebServer: Netty started on port 8080
kafka-ui | 2023-05-10 07:17:13,636 INFO [main] c.p.k.u.KafkaUiApplication: Started KafkaUiApplication in 13.397 seconds (process running for 15.434)
kafka-ui | 2023-05-10 07:17:13,705 DEBUG [parallel-1] c.p.k.u.s.ClustersStatisticsScheduler: Start getting metrics for kafkaCluster: henry-msk-cluster
kafka-ui | 2023-05-10 07:17:13,746 INFO [parallel-1] o.a.k.c.a.AdminClientConfig: AdminClientConfig values:
kafka-ui | bootstrap.servers = [b-2.henrymskcluster.4ka9rz.c4.kafka.ap-northeast-2.amazonaws.com:9092, b-1.henrymskcluster.4ka9rz.c4.kafka.ap-northeast-2.amazonaws.com:9092]
kafka-ui | client.dns.lookup = use_all_dns_ips
kafka-ui | client.id = kafka-ui-admin-1683703033-1
kafka-ui | connections.max.idle.ms = 300000

```
kafka-ui  |  default.api.timeout.ms = 60000
kafka-ui  |  metadata.max.age.ms = 300000
kafka-ui  |  metric.reporters = []
kafka-ui  |  metrics.num.samples = 2
kafka-ui  |  metrics.recording.level = INFO
kafka-ui  |  metrics.sample.window.ms = 30000
kafka-ui  |  receive.buffer.bytes = 65536
kafka-ui  |  reconnect.backoff.max.ms = 1000
kafka-ui  |  reconnect.backoff.ms = 50
kafka-ui  |  request.timeout.ms = 30000
kafka-ui  |  retries = 2147483647
kafka-ui  |  retry.backoff.ms = 100
kafka-ui  |  sasl.client.callback.handler.class = null
kafka-ui  |  sasl.jaas.config = null
kafka-ui  |  sasl.kerberos.kinit.cmd = /usr/bin/kinit
kafka-ui  |  sasl.kerberos.min.time.before.relogin = 60000
kafka-ui  |  sasl.kerberos.service.name = null
kafka-ui  |  sasl.kerberos.ticket.renew.jitter = 0.05
kafka-ui  |  sasl.kerberos.ticket.renew.window.factor = 0.8
kafka-ui  |  sasl.login.callback.handler.class = null
kafka-ui  |  sasl.login.class = null
kafka-ui  |  sasl.login.connect.timeout.ms = null
kafka-ui  |  sasl.login.read.timeout.ms = null
kafka-ui  |  sasl.login.refresh.buffer.seconds = 300
kafka-ui  |  sasl.login.refresh.min.period.seconds = 60
kafka-ui  |  sasl.login.refresh.window.factor = 0.8
kafka-ui  |  sasl.login.refresh.window.jitter = 0.05
kafka-ui  |  sasl.login.retry.backoff.max.ms = 10000
kafka-ui  |  sasl.login.retry.backoff.ms = 100
kafka-ui  |  sasl.mechanism = GSSAPI
kafka-ui  |  sasl.oauthbearer.clock.skew.seconds = 30
kafka-ui  |  sasl.oauthbearer.expected.audience = null
kafka-ui  |  sasl.oauthbearer.expected.issuer = null
kafka-ui  |  sasl.oauthbearer.jwks.endpoint.refresh.ms = 3600000
kafka-ui  |  sasl.oauthbearer.jwks.endpoint.retry.backoff.max.ms = 10000
kafka-ui  |  sasl.oauthbearer.jwks.endpoint.retry.backoff.ms = 100
kafka-ui  |  sasl.oauthbearer.jwks.endpoint.url = null
kafka-ui  |  sasl.oauthbearer.scope.claim.name = scope
kafka-ui  |  sasl.oauthbearer.sub.claim.name = sub
kafka-ui  |  sasl.oauthbearer.token.endpoint.url = null
kafka-ui  |  security.protocol = PLAINTEXT
kafka-ui  |  security.providers = null
kafka-ui  |  send.buffer.bytes = 131072
kafka-ui  |  socket.connection.setup.timeout.max.ms = 30000
kafka-ui  |  socket.connection.setup.timeout.ms = 10000
kafka-ui  |  ssl.cipher.suites = null
kafka-ui  |  ssl.enabled.protocols = [TLSv1.2, TLSv1.3]
kafka-ui  |  ssl.endpoint.identification.algorithm = https
kafka-ui  |  ssl.engine.factory.class = null
kafka-ui  |  ssl.key.password = null
kafka-ui  |  ssl.keymanager.algorithm = SunX509
kafka-ui  |  ssl.keystore.certificate.chain = null
kafka-ui  |  ssl.keystore.key = null
kafka-ui  |  ssl.keystore.location = null
kafka-ui  |  ssl.keystore.password = null
kafka-ui  |  ssl.keystore.type = JKS
kafka-ui  |  ssl.protocol = TLSv1.3
kafka-ui  |  ssl.provider = null
kafka-ui  |  ssl.secure.random.implementation = null
kafka-ui  |  ssl.trustmanager.algorithm = PKIX
kafka-ui  |  ssl.truststore.certificates = null
kafka-ui  |  ssl.truststore.location = null
kafka-ui  |  ssl.truststore.password = null
kafka-ui  |  ssl.truststore.type = JKS
kafka-ui  |
kafka-ui  | 2023-05-10 07:17:13,940 INFO  [parallel-1] o.a.k.c.u.AppInfoParser: Kafka version: 3.3.1
kafka-ui  | 2023-05-10 07:17:13,945 INFO  [parallel-1] o.a.k.c.u.AppInfoParser: Kafka commitId: e23c59d00e687ff5
kafka-ui  | 2023-05-10 07:17:13,946 INFO  [parallel-1] o.a.k.c.u.AppInfoParser: Kafka startTimeMs: 1683703033938
kafka-ui  | 2023-05-10 07:17:15,002 DEBUG [parallel-1] c.p.k.u.s.ClustersStatisticsScheduler: Metrics updated for cluster:
henry-msk-cluster
```

3)웹브라우저로 방문하기

-http://{Kafka UI Server의 퍼블릭 IPv4 DNS 값}:8080/

5. [Lab7 API를 통한 Kafka 실습] 실행해서 Kafka-ui로 확인하기