Auzere EventHub(KAFKA) With Actor

Azure EventHub .Net Core Akka.net .

EventHub , KAFKA KAFKA-SASS

KAFKA Azure .

- net core 3.1
- akka.net
- Confluent.Kafka

AkkaStream With EventHub AKKAStream

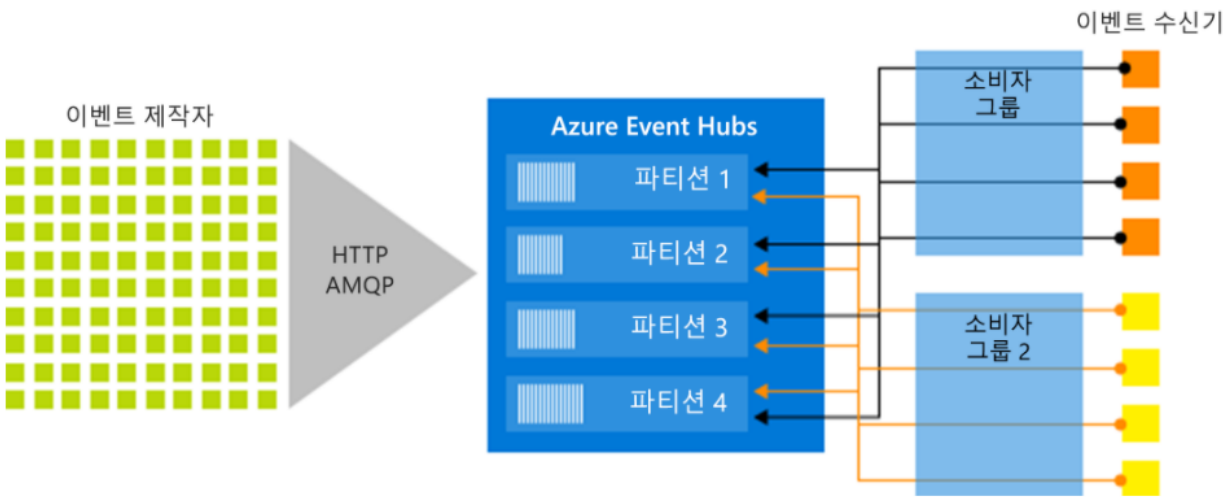
Akka.net EventHub(Kafka)

Kafka , .

- Event Hub
-
-
  - (Topic)
-
-
-
- Kafka -
  -
- SAS
  -
  - EventHub Kafka
- Consumer
- API EventHub
- Kafka
-

Event Hub

Event Hubs PaaS(Platform-as-a-Service) . Apache Kafka Event Hubs , PaaS Kafka .



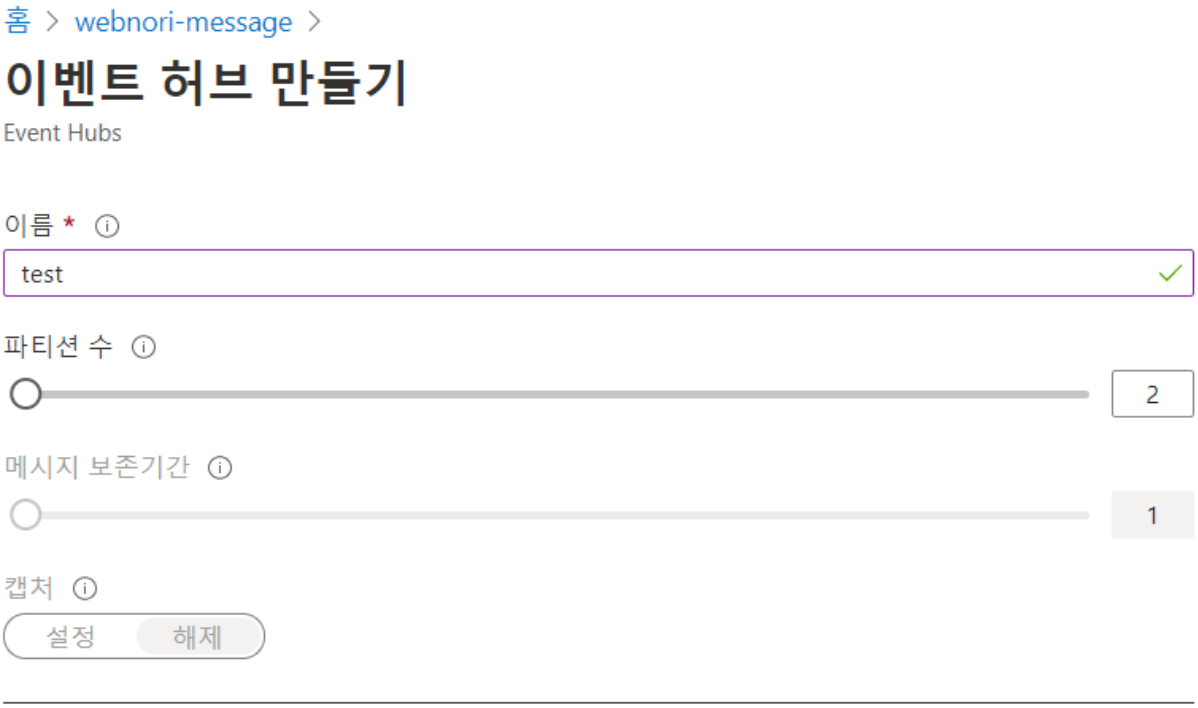
- : . HTTPS AMQP 1.0 Apache Kafka 1.0 .

- : .
- : (,). . .
- : Event Hubs .
- : . Event Hubs AMQP 1.0 . Event Hubs . Kafka Kafka 1.0 .

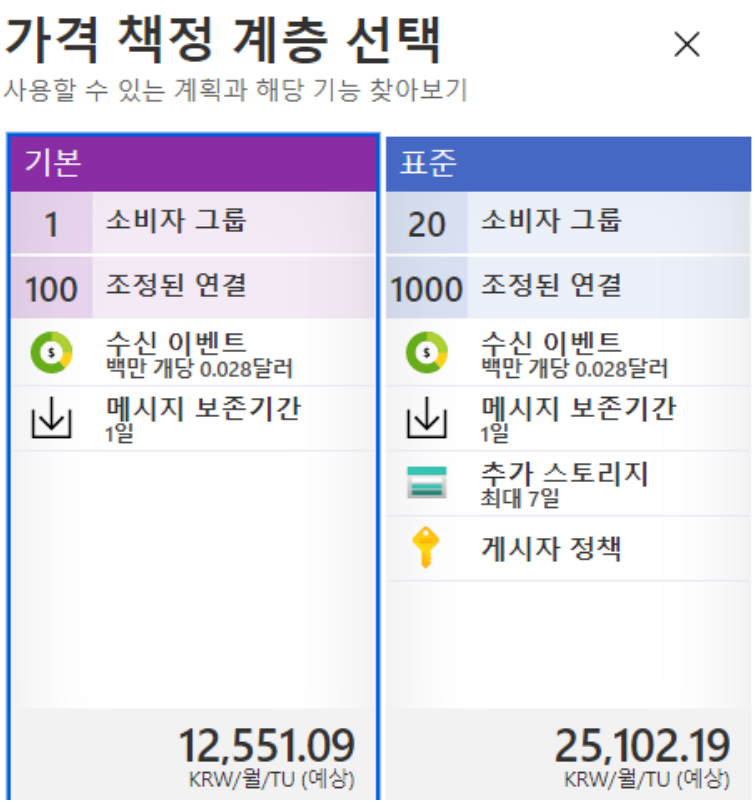
Kafka 및 Event Hubs의 개념적 매핑

Kafka 개념	Event Hubs 클러스터
클러스터	네임스페이스
항목	이벤트 허브
파티션	파티션
소비자 그룹	소비자 그룹
Offset	Offset

(Topic)



name , kafka topic , "akka100" . ( )

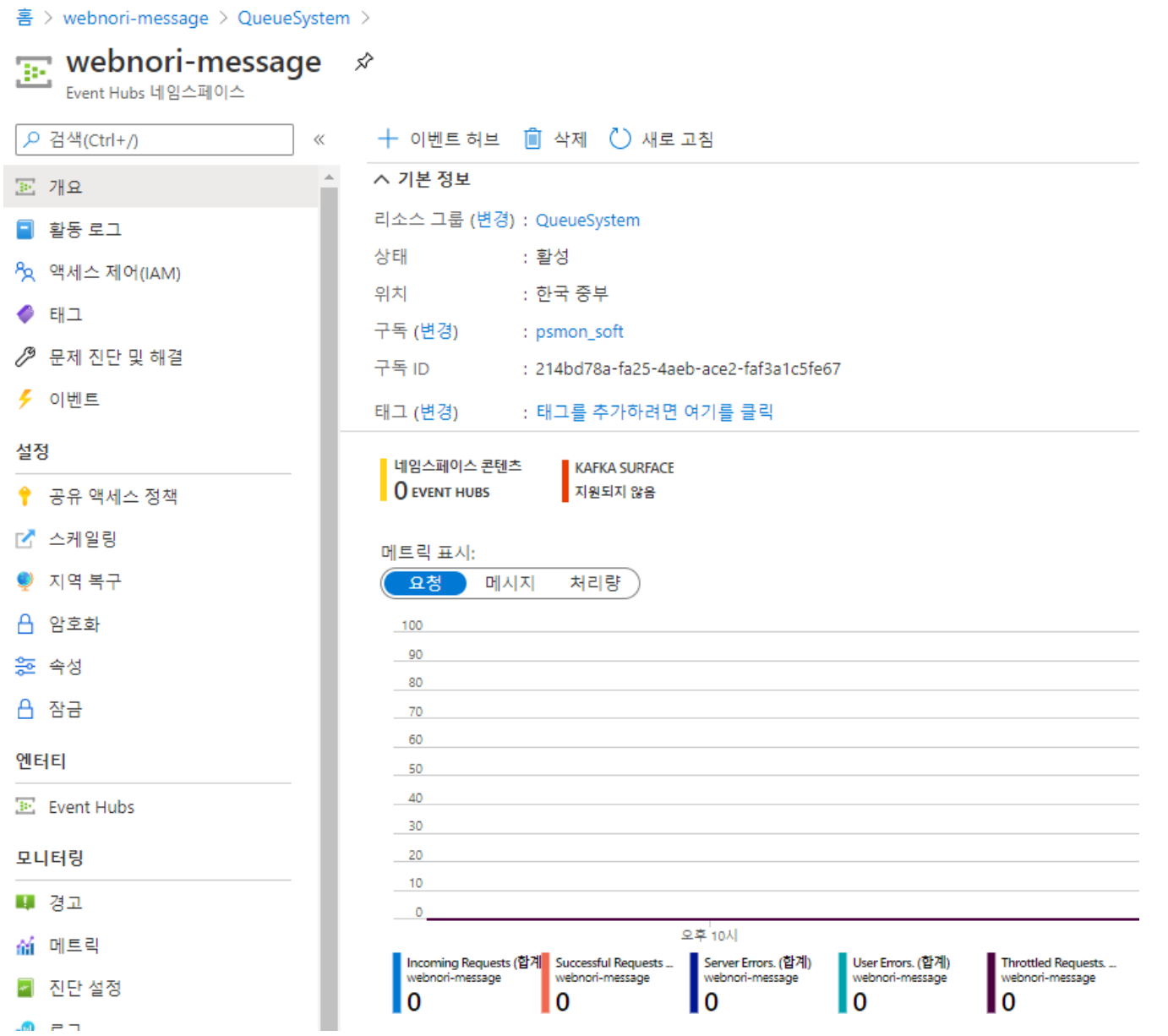


Kafka . (, Kafka 1 2)

```
BootstrapServers = "webnori-kafka.servicebus.windows.net:9093"
```

Kafka Brokers .

Port 9093 SSL . (KAFKA : SaslSsl , Kafka)



KAFKASUFACE .

(, . - .)

, KAFKA .

Kafka , .

Kafka Suface , Kafka .

: https://docs.microsoft.com/ko-kr/azure/event-hubs/apache-kafka-developer-guide

KafkaClient , Akka.net .

(Akka.net Stream Kafka SaSSL . - .)

Kafka -

, Auzre .

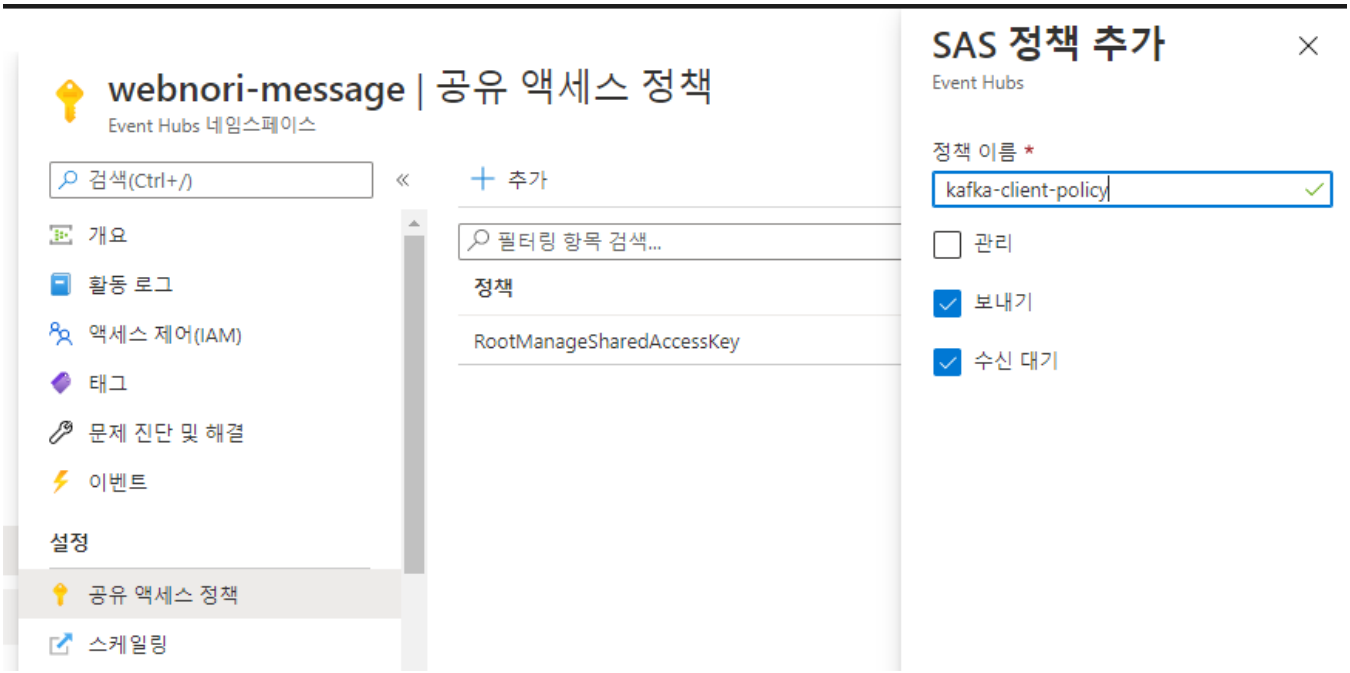
AkkaDotModule AKKA.net KAFKA

Akka.net , .

https://www.nuget.org/packages/AkkaDotModule.Webnori/ - 1.0.9

Nuget , .

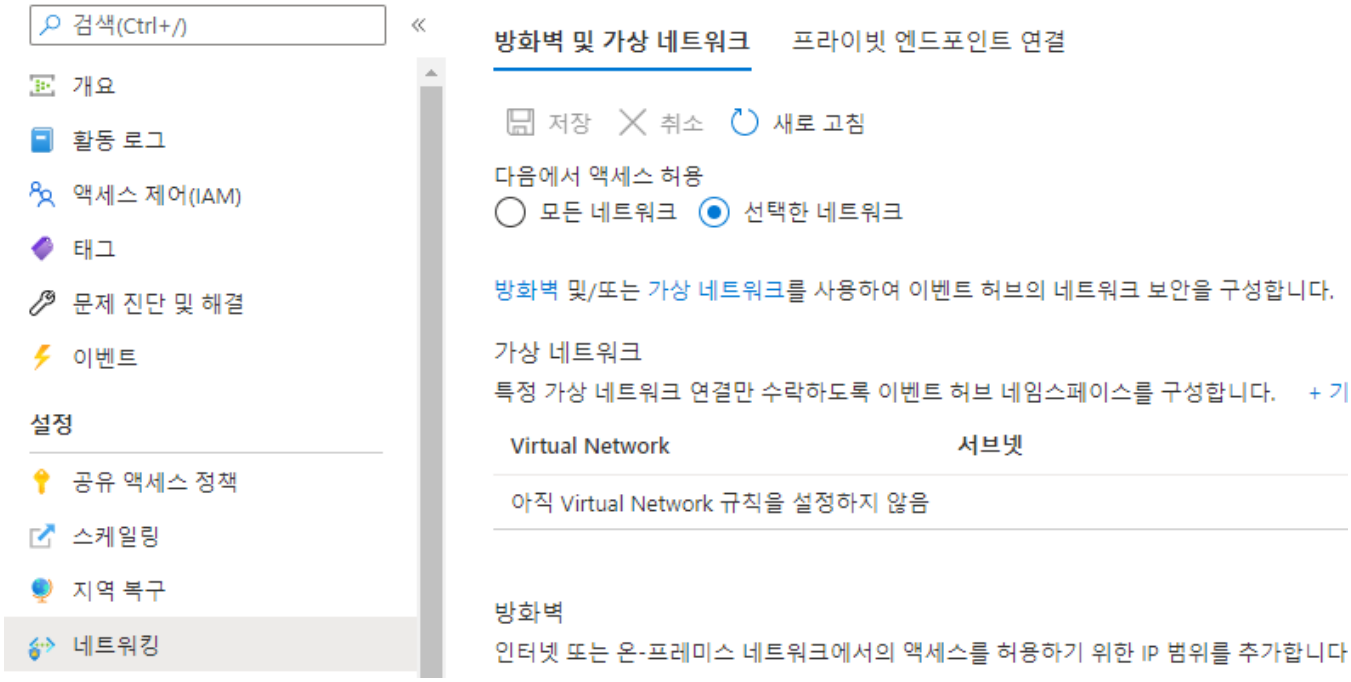
SAS




```
            var producerAkkaOption = new ProducerAkkaOption()
            {
                BootstrapServers = "webnori-kafka.servicebus.windows.net:9093",
                ProducerName = "webnori-kafka",
                SecuritOption = new KafkaSecurityOption()
                {
                    SecurityProtocol = SecurityProtocol.SaslSsl,
                    SaslMechanism = SaslMechanism.Plain,
                    SaslUsername = "$ConnectionString",
                    SaslPassword = "Endpoint=sb://webnori-kafka.servicebus.windows.net/;
SharedAccessKeyName=kafka-client;SharedAccessKey=PfL0qRUm50AXZHRXLiVfnatIRI3OqAh+dT6Owsqrd2M=",
                    SslCaLocation = "./cacert.pem"
                }
            };
```

, SaslPassword .

SSL Global Root : https://github.com/psmon/AkkaDotModule/blob/master/AkkaDotBootApi/cacert.pem



, IP Base .

IP , String .

(2 , - , .)

Off.

EventHub Kafka

EventHub(Kafka)

. Kafka SecurityOption SSL .

```
            var producerAkkaOption = new ProducerAkkaOption()
            {
                BootstrapServers = "webnori-kafka.servicebus.windows.net:9093",
                ProducerName = "webnori-kafka",
                SecuritOption = new KafkaSecurityOption()
                {
                    SecurityProtocol = SecurityProtocol.SaslSsl,
                    SaslMechanism = SaslMechanism.Plain,
                    SaslUsername = "$ConnectionString",
                    SaslPassword = "Endpoint=sb://webnori-kafka.servicebus.windows.net/;
SharedAccessKeyName=kafka-client;SharedAccessKey=PfL0qRUm50AXZHRXLiVfnatIRI3OqAh+dT6Owsqrd2M=",
                    SslCaLocation = "./cacert.pem"
                }
            };

            string producerActorName = "producerActor";

            var producerActor= AkkaLoad.RegisterActor(producerActorName /*AkkaLoad  */,
                actorSystem.ActorOf(Props.Create(() =>
                    new ProducerActor(producerAkkaOption)),
                    producerActorName /*AKKA  Path*/
            ));

            producerActor.Tell(new BatchData()
            {
                Data = new KafkaTextMessage()
                {
                    Topic = "akka100",
                    Message = "testData"
                }
            });
```

: , AkkaStream , .

- : https://github.com/psmon/AkkaDotModule/blob/master/AkkaDotBootApi/Test/TestAkka.cs
- : https://github.com/psmon/AkkaDotModule/blob/master/AkkaDotModule/ActorUtils/Confluent/ProducerActor.cs

Consumer

```
            //ConsumerActor
            var consumerAkkaOption = new ConsumerAkkaOption()
            {
                BootstrapServers = "webnori-kafka.servicebus.windows.net:9093",
                Topics = "akka100",
                AutoOffsetReset = AutoOffsetReset.Earliest,
                KafkaGroupId = "akakTestGroup",
                RelayActor = null,  //  ,    (    )
                SecurityOption = new KafkaSecurityOption()
                {
                    SecurityProtocol = SecurityProtocol.SaslSsl,
                    SaslMechanism = SaslMechanism.Plain,
                    SaslUsername = "$ConnectionString",
                    SaslPassword = "Endpoint=sb://webnori-kafka.servicebus.windows.net/;
SharedAccessKeyName=kafka-client;SharedAccessKey=PfL0qRUm50AXZHRXLiVfnatIRI3OqAh+dT6Owsqrd2M=",
                    SslCaLocation = "./cacert.pem"
                }
            };

            string consumerActorName = "consumerActor";
            var consumerActor = AkkaLoad.RegisterActor(consumerActorName /*AkkaLoad  */,
                actorSystem.ActorOf(Props.Create(() =>
                    new ConsumerActor(consumerAkkaOption)),
                    consumerActorName /*AKKA  Path*/
            ));

            // .
            consumerActor.Tell(new ConsumerStart());
```

SubScibe Consumer .

Consumer , ConsumerStart-

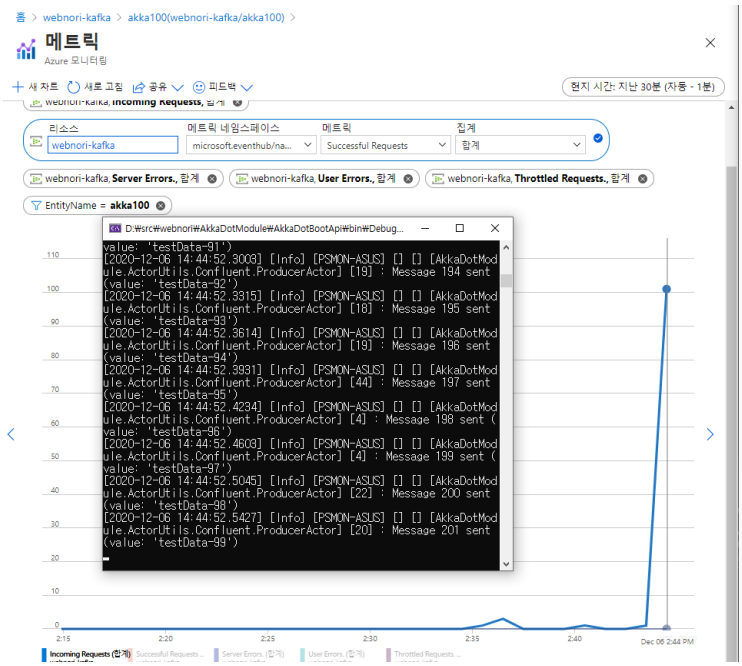
Consumer .

API EventHub

Auzre Kafka ,

, .

.



: https://github.com/psmon/AkkaDotModule/blob/master/AkkaDotBootApi/Controllers/KafkaController.cs

```
<PackageReference Include="Confluent.Kafka" Version="1.5.2" />
```

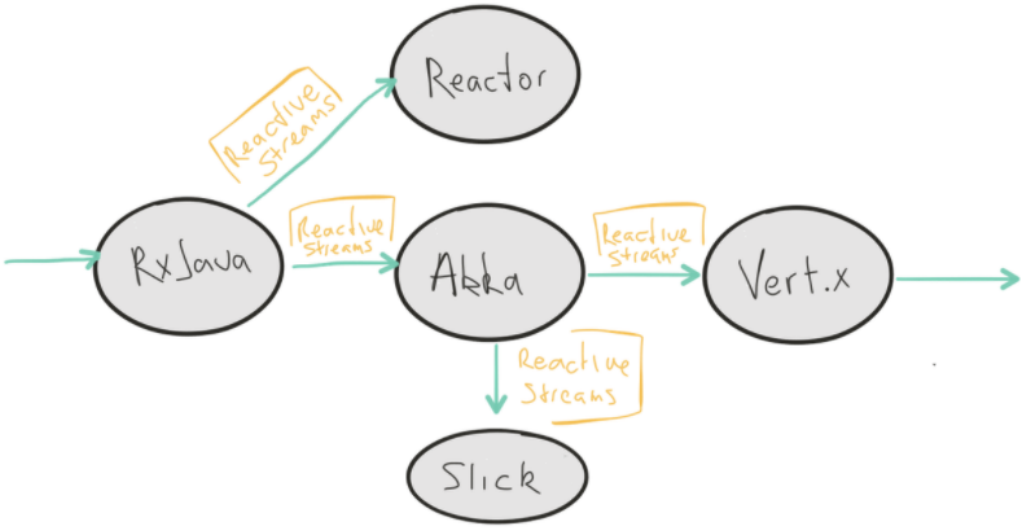
Confluent.Kafka , Actor

EventHub(Kafka) .

Kafka

Reactive Streams 구현들

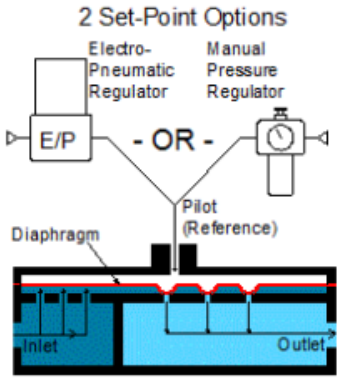
당신만의 스트림 처리 시스템을 만들 준비가 되었다면 라이브러리를 선택할 필요가 있을 것이다. Reactive Streams 명세의 구현은 서로 호환되며 이는 Reactive Streams 호환 라이브러리 선택의 명백한 장점이다. 명세를 준수하는 한 다른 라이브러리들을 사용하는 스트림 처리 시스템들을 통합할 수도 있다.



: http://eyeahs.github.io/blog/2017/01/24/a-journey-into-reactive-streams/

KafkaClient , Akka Stream Akka .

- : KafkaClient , .
- : , .
- ( ) : ,
  , .



기준압력이 수동 또는 전자 기준압력조절기로 입력됩니다.

,

, .

- : https://github.com/psmon/AkkaDotModule/blob/master/AkkaDotBootApi/Test/TestAkka.cs
- AKKA.net Stream : https://getakka.net/articles/streams/buffersandworkingwithrate.html
- MS- : https://docs.microsoft.com/ko-kr/azure/event-hubs/event-hubs-about
- Kafka-Security : https://springboot.cloud/31
- Alpakka.net(AkkaStream) : https://alpakka.getakka.net/documentation/index.html