

# How to connect to NYSE Kafka cluster in PROD Env

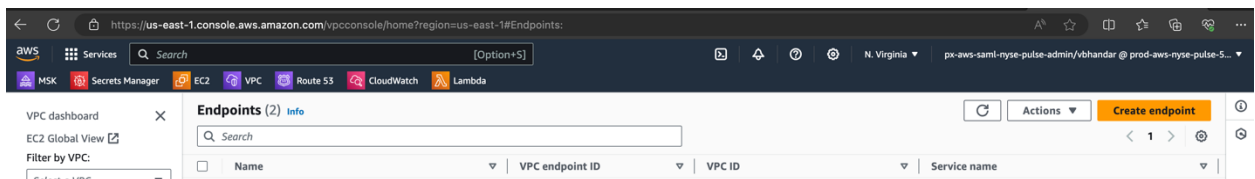
## Prerequisites:

- A VPC with 3 subnets, based on the AWS region where consumer is setting up the client,  

<b>Region:</b> US East (us-east-1)	<b>AZ IDs:</b> use1-az2, use1-az4 and use1-az6
------------------------------------	--
- Access to create VPC endpoints in us-east-1 region or ap-east-1.
- Customer has provided their info to NYSE and entered their role ARN and have Pulse Kafka username/password generated by NYSE RAAMS system handy.

## Steps to create VPC Endpoints:

1. Once logged in to your AWS account (Use the same role to login to AWS account which has been provided to NYSE), navigate to [Endpoints | VPC Console \(amazon.com\)](#) and click on Create Endpoint button



2. On Next page Click on “PrivateLink Ready partner Services” radio button and enter one of the endpoint service names depending on your AWS region, as shown below. Please note your region and Availability Zone IDs will vary depending on the region.

Region	Endpoint Service Name	Availability Zones	Name Tag	Port
US East 1 (us-east-1)	com.amazonaws.vpce.us-east-1.vpce-svc-0fb7d261249ecaed9	use1-az2, use1-az4 and use1-az6	broker.prod.use1.bqt.pulse.nyse	9094

Once the appropriate Service Name is entered, click on “Verify service”.

**Endpoint settings**  
Specify a name and select the type of endpoint.

**Name tag** - optional  
Creates a tag with a key of 'Name' and a value that you specify. Tags help you find and manage your endpoint.

broker.prod.use1.bqt.pulse.nyse

**Type** [Info](#)  
Select a category

☐ **AWS services**  
Connect to services provided by Amazon with an interface endpoint, or a Gateway endpoint.

☒ **PrivateLink Ready partner services**  
Connect to SaaS services which have AWS Service Ready designation with an interface endpoint. Uses AWS PrivateLink.

☐ **AWS Marketplace services**  
Connect to SaaS services that you have purchased through AWS Marketplace with an interface endpoint.

☐ **EC2 Instance Connect Endpoint**  
An elastic network interface that allow you to connect to resources in a private subnet.

☐ **Resources - New**  
Connect to resources like Amazon Relational Database Services (RDS) with a Resource endpoint. Uses AWS PrivateLink.

☐ **Service networks - New**  
Connect to VPC Lattice service networks with a Service network endpoint. Uses AWS PrivateLink.

☐ **Endpoint services that use NLBs and GWLBs**  
Find services shared with you by service name. Connect to a Network LoadBalancer (NLB) service with an interface endpoint or to a Gateway LoadBalancer (GWLB) service with a Gateway Load Balancer endpoint.

**Service settings**

**Pre-existing subscription required**  
Third-party services offered over AWS PrivateLink and validated by AWS for following best practices as part of the [PrivateLink Service Ready program](#). [\[?\]](#)

**Service name**  
com.amazonaws.vpce.us-east-1.vpce-svc-0fb7d261249ecaed9 [Verify service](#)

**Service Region**  
☐ **Enable Cross Region endpoint** [Info](#)  
Connect to cross Region enabled services.

☒ **Service name verified.**

**Network settings**  
Select the VPC in which to create the endpoint

**VPC**  
Create the VPC endpoint in the VPC in the same AWS Region from which you will access a resource.

vpc-033b030ec749b9f9c (PULSE-Dev-Prod-Consumer-VPC) [\[?\]](#)

**Additional settings**

**DNS name**  
☐ **Enable DNS name** [Info](#)  
Associates a private hosted zone with the VPC that contains a record set that enables you to leverage Amazon's private network connectivity to the service while making requests to the service's default public endpoint DNS name. To use this feature, ensure that the attributes 'Enable DNS hostnames' and 'Enable DNS support' are enabled for your VPC.

**DNS record IP type**  
☒ **IPv4**  
☐ IPv6  
☐ Dualstack  
☐ Service defined

**Subnets** (3) [Info](#)

<input type="checkbox"/>	Availability Zone	Subnet ID	Designate IP addresses	IPv4 address	IPv6 address
<input type="checkbox"/>	us-east-1a (use1-az6)	Select a subnet	<input type="checkbox"/>		
<input type="checkbox"/>	us-east-1c (use1-az2)	Select a subnet	<input type="checkbox"/>		
<input type="checkbox"/>	us-east-1d (use1-az4)	Select a subnet	<input type="checkbox"/>		

## Note:

When you create endpoint from ap-east-1 (Hong Kong) region, please choose "Enable Service Region endpoint" and select "us-east-1" as a region, as shown below.

**Service settings**

**Service name**  
com.amazonaws.vpce.us-east-1.vpce-svc-0fb7d261249ecaed9 [Verify service](#)

**Service Region**  
☒ **Enable Cross Region endpoint** [Info](#)  
Connect to cross Region enabled services.

US East (N. Virginia) (us-east-1)

☒ **Service name verified.**

☒ **You are creating an Endpoint that connects to another Service Region**

- Once Service Name is verified, pick your VPC and subnets and click on Create endpoint. This endpoint DNS name will be referred to as broker.prod.use1.bqt.pulse.nyse in step#6 below.
- Repeat steps 2 and 3 for three more times using following service names based on the region where you are creating the endpoints. Your AWS region and Availability Zone IDs will vary depending on the region:

Region	Endpoint Service Name	Availability Zones	Name Tag	Port
US East (us-east-1)	com.amazonaws.vpce.us-east-1.vpce-svc-065db063a68f6c80f	use1-az2	broker1.prod.use1.bqt.pulse.nyse	9094
US East (us-east-1)	com.amazonaws.vpce.us-east-1.vpce-svc-00f6fe24081f3c61f	use1-az4	broker2.prod.use1.bqt.pulse.nyse	9094
US East (us-east-1)	com.amazonaws.vpce.us-east-1.vpce-svc-0b58442c9f5ee8ba4	use1-az6	broker3.prod.use1.bqt.pulse.nyse	9094

*Please note VPC endpoint DNS names from 4 will be referred to in step number 6 below.*

- Navigate to Route 53 and create a private hosted zone named **bqt.pulse.nyse** and pick the appropriate VPC in which the endpoints were created.
- Once hosted zone is created click on it and create 4 different records of type CNAME namely using the VPC endpoint DNS names from step # 4:

broker.prod.use1.bqt.pulse.nyse
broker1.prod.use1.bqt.pulse.nyse
broker2.prod.use1.bqt.pulse.nyse
broker3.prod.use1.bqt.pulse.nyse

## Define simple record

×

**Record name**
Info

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to `blog.example.com`, enter `blog`. If you leave this field blank, the default record name is the name of the domain.

Keep blank to create a record for the root domain.

**Record type**
Info

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

▼

Choose when routing traffic to some Elastic Beanstalk environments or to Amazon RDS database instances.

**Value/Route traffic to**
Info

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

▼

Enter multiple values on separate lines.

**TTL (seconds)**
Info

The amount of time, in seconds, that DNS resolvers and web browsers cache the settings in this record. ("TTL" means "time to live.")

Recommended values: 60 to 172800 (two days)

Cancel

7. Send an email to `Dev-NYSE-Cloud@ice.com` and try a library of your choice to test kafka connectivity using `broker.prod.use1.bqt.pulse.nyse:9094` as the hostname and credentials shared with you.
8. Some sample libraries that you can use are:
  - a. <https://github.com/confluentinc/librdkafka> : C++
  - b. <https://kafka.js.org/> : nodejs
  - c. <https://docs.confluent.io/kafka-clients/python/current/overview.html> : python

9. To parse the messages from the stream, download the [proto](#) file and compile it using protoc command for the tech stack your system uses:

C/C++	Python	Java	Javascript
<pre>protoc --proto_path=src -- cpp_out=\$DEST/gen/foo. proto \$SRC/ bqt_cloud_streaming.proto</pre>	<pre>protoc -I=\$SRC_DIR -- python_out=\$DST_DIR \$SRC_DIR/ bqt_cloud_streaming.proto</pre>	<pre>protoc -I=\$SRC_DIR -- java_out=\$DST_DIR \$SRC_DIR/ bqt_cloud_streaming.proto</pre>	<pre>protoc --js_out=gen bqt_cloud_streaming. proto</pre>

10. Once you include the generated binary in your application, you can use the auto generated methods from above to deserialize the kafka message. Tutorials are in <https://protobuf.dev/getting-started/>
- 
11. Detailed client specifications for kafka are available [here](#).