StreamNative Cloud

for Kafka®

StreamNative Cloud for Kafka® Quick Start v1.0

This Quick Start will walk you through the process of getting started with StreamNative Cloud for Kafka®. The procedures outlined in this document are valid only for the beta version of this feature. You need to sign up for the beta program, even if you have an existing StreamNative Cloud account.

Fill out the StreamNative Cloud for Kafka® beta form. You'll get a pop-up confirmation that we received your request. A member of the StreamNative team will then contact you directly, either by email or via phone call, to confirm your participation in the beta and to begin your beta account configuration. Note that there might be a delay in receiving the beta account creation contact, depending on the availability of our team to set up your account.

To get started with StreamNative Cloud for Kafka®, follow these steps.

Step 1: Navigate to your cloud account

- After you have received communication from StreamNative confirming your acceptance into the beta program, navigate to the StreamNative Cloud Manager login page: https://console.streamnative.cloud/.

Step 2: Create an Organization

For the beta, you should create a new organization to ensure it has the correct configuration for enabling Kafka.

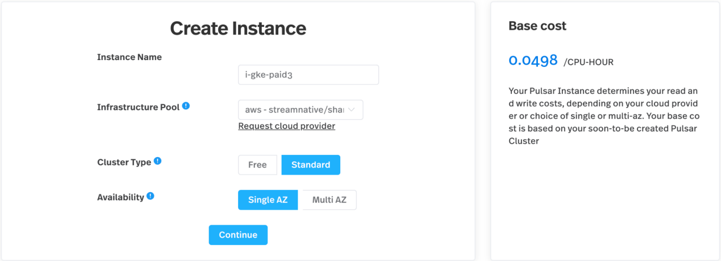
1. Click **Create organization**.
2. Enter a name for the organization and then click **Confirm**. An organization name must be less than 12 characters and can contain any combination of lowercase letters (a-z), numbers (0-9), and hyphens (-).

NOTE: You might have to wait briefly for the organization to be created. After your new organization is created, proceed to creating an instance.

Step 3: Create an Instance

For best testing performance, create a standard cluster. Speak with your StreamNative team contact if you would like to perform testing with a standard cluster. This step assumes you are creating a standard cluster.

1. Click the organization that you want to create an instance for, and then click **Create instance**.



Create Instance

Instance Name: i-gke-paid3

Infrastructure Pool: aws - streamnative/shai

Cluster Type: Free [✓ Standard]

Availability: [✓ Single AZ] Multi AZ

Base cost

0.0498 /CPU-HOUR

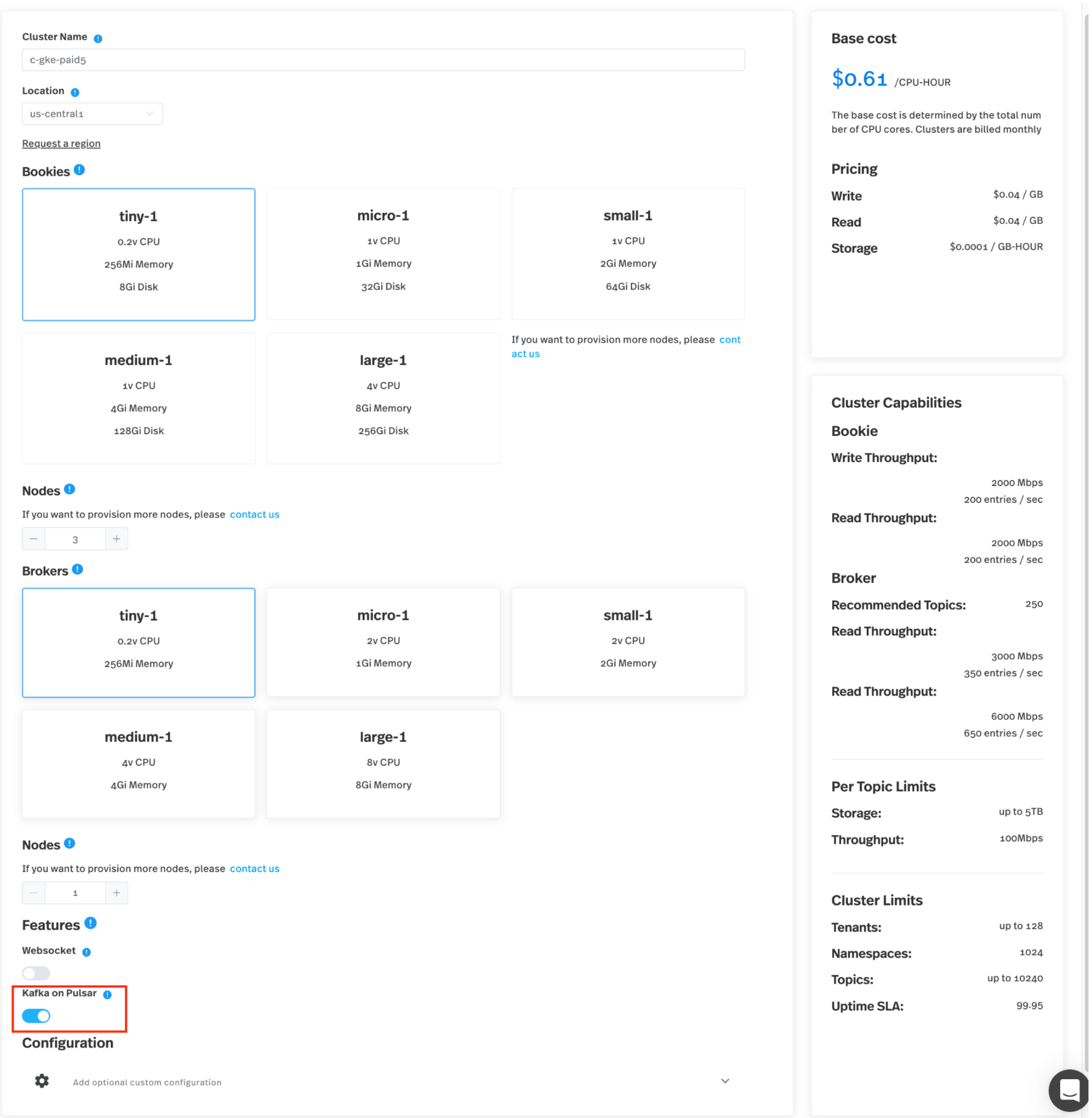
Your Pulsar Instance determines your read an d write costs, depending on your cloud provid er or choice of single or multi-az. Your base co st is based on your soon-to-be created Pulsar Cluster

2. Configure the instance, and then click **Continue**.
3. In the **Create Payment Method** box, enter a valid credit card number, and then click **Create Payment Method**.
4. Click **Continue**.

Next, create a cluster that has KoP enabled as outlined in the following step.

Step 4: Create a KoP Enabled Cluster

1. On the left navigation pane, in the **Admin** area, click **Pulsar Clusters**.
2. Configure the cluster. Make sure to toggle the **Kafka on Pulsar** option as shown in the illustration below.



Cluster Name

c-gke-paid5

Location

us-central1

Bookies

tiny-1	micro-1	small-1
0.2v CPU	1v CPU	1v CPU
256Mi Memory	1Gi Memory	2Gi Memory
8Gi Disk	32Gi Disk	64Gi Disk

medium-1	large-1
1v CPU	4v CPU
4Gi Memory	8Gi Memory
128Gi Disk	256Gi Disk

If you want to provision more nodes, please contact us

Nodes

If you want to provision more nodes, please contact us

3

Brokers

tiny-1	micro-1	small-1
0.2v CPU	2v CPU	2v CPU
256Mi Memory	1Gi Memory	2Gi Memory

medium-1	large-1
4v CPU	8v CPU
4Gi Memory	8Gi Memory

Nodes

If you want to provision more nodes, please contact us

1

Features

Websocket

Kafka on Pulsar

Configuration

Add optional custom configuration

Base cost

$0.61 /CPU-HOUR

The base cost is determined by the total number of CPU cores. Clusters are billed monthly

Pricing

Write	$0.04 / GB
Read	$0.04 / GB
Storage	$0.0001 / GB-HOUR

Cluster Capabilities

Bookie

Write Throughput:	2000 Mbps 200 entries / sec
Read Throughput:	2000 Mbps 200 entries / sec

Broker

Recommended Topics:	250
Read Throughput:	3000 Mbps 350 entries / sec
Read Throughput:	6000 Mbps 650 entries / sec

Per Topic Limits

Storage:	up to 5TB
Throughput:	100Mbps

Cluster Limits

Tenants:	up to 128
Namespaces:	1024
Topics:	up to 10240
Uptime SLA:	99.95

3. Click **Add Cluster**.

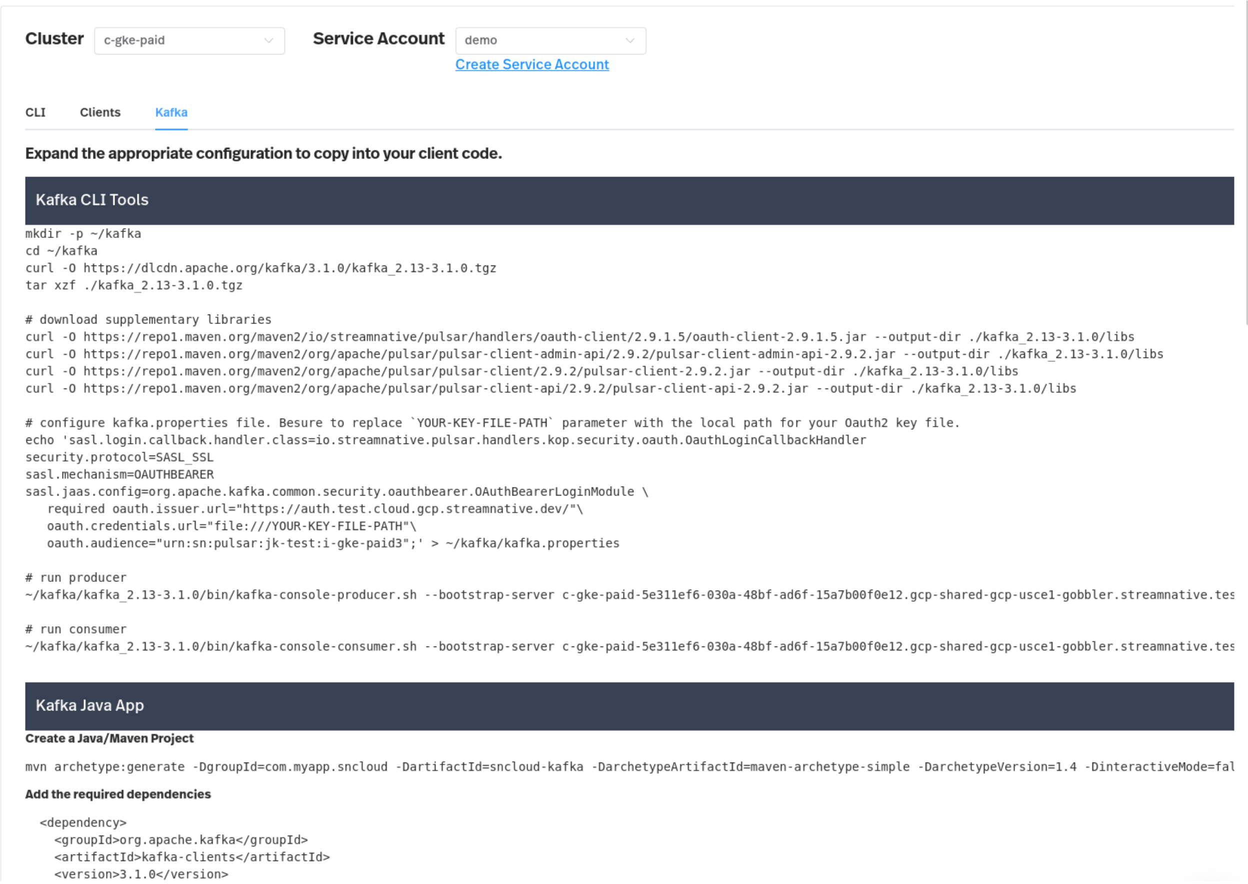
Step 5: Create a Service Account

1. On the left navigation pane, in the **Admin** area, click **Service Accounts**.
2. Click **Create Service Account**, enter a name in the name field, and click **Confirm**. After creating a Service Account, you need to add it to a Tenant/Namespace.
3. On the left navigation pane, in the **Admin** area, click **Tenants/Namespaces**.
4. In the row for the **Public** tenant, in the **Admin Roles** column, click the arrow next to **Admin Roles** to select the Service Account you just created.

You can now use the Service Account for connecting with a client.

Step 6: Validate your connection to the cluster

1. On the left navigation pane, click **Clients**
2. In the primary tab bar, select **Kafka**



Expand the appropriate configuration to copy into your client code.

Kafka CLI Tools

```
mkdir -p ~/kafka
cd ~/kafka
curl -O https://dlcdn.apache.org/kafka/3.1.0/kafka_2.13-3.1.0.tgz
tar xzf ./kafka_2.13-3.1.0.tgz

# download supplementary libraries
curl -O https://repo1.maven.org/maven2/io/streamnative/pulsar/handlers/oauth-client/2.9.1.5/oauth-client-2.9.1.5.jar --output-dir ./kafka_2.13-3.1.0/libs
curl -O https://repo1.maven.org/maven2/org/apache/pulsar/pulsar-client-admin-api/2.9.2/pulsar-client-admin-api-2.9.2.jar --output-dir ./kafka_2.13-3.1.0/libs
curl -O https://repo1.maven.org/maven2/org/apache/pulsar/pulsar-client/2.9.2/pulsar-client-2.9.2.jar --output-dir ./kafka_2.13-3.1.0/libs
curl -O https://repo1.maven.org/maven2/org/apache/pulsar/pulsar-client-api/2.9.2/pulsar-client-api-2.9.2.jar --output-dir ./kafka_2.13-3.1.0/libs

# configure kafka.properties file. Besure to replace `YOUR-KEY-FILE-PATH` parameter with the local path for your Oauth2 key file.
echo 'sasl.login.callback.handler.class=io.streamnative.pulsar.handlers.kop.security.oauth.OauthLoginCallbackHandler
security.protocol=SASL_SSL
sasl.mechanism=OAUTHBEARER
sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule \
   required oauth.issuer.url="https://auth.test.cloud.gcp.streamnative.dev/"\
   oauth.credentials.url="file:///YOUR-KEY-FILE-PATH"\
   oauth.audience="urn:sn:pulsar:jk-test:i-gke-paid3";' > ~/kafka/kafka.properties

# run producer
~/kafka/kafka_2.13-3.1.0/bin/kafka-console-producer.sh --bootstrap-server c-gke-paid-5e311ef6-030a-48bf-ad6f-15a7b00f0e12.gcp-shared-gcp-usce1-gobbler.streamnative.tes

# run consumer
~/kafka/kafka_2.13-3.1.0/bin/kafka-console-consumer.sh --bootstrap-server c-gke-paid-5e311ef6-030a-48bf-ad6f-15a7b00f0e12.gcp-shared-gcp-usce1-gobbler.streamnative.tes
```

Kafka Java App

Create a Java/Maven Project

```
mvn archetype:generate -DgroupId=com.myapp.sncloud -DartifactId=sncloud-kafka -DarchetypeArtifactId=maven-archetype-simple -DarchetypeVersion=1.4 -DinteractiveMode=fal
```

Add the required dependencies

```
  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>3.1.0</version>
```

3. Following the instructions for connecting, using either the CLI tools or a Java application