

[StreamNative Managed Cloud](#) >

QuickStart - StreamNative Cloud

This QuickStart takes you through the steps for provisioning a cluster and building a simple Pulsar client with Java.

Note

This QuickStart assumes you are familiar with the [basics concepts of Apache Pulsar](#).

Prerequisites

Ensure you have the following installed and configured before starting:

- [Java 17](#)
- [Maven](#)

Sign up

Note

If you have an email account configured for using Single Sign-On (SSO) with StreamNative Cloud, use that email address and password when signing up.

To sign up, navigate to the [StreamNative Cloud Console login page](#). Follow the prompts to create an account. After you click **Finish**, you might have to wait briefly for your first organization to be created. After your new organization is created, continue on to creating your first instance and cluster.

Create a Pulsar instance and cluster

You first need to create an [instance](#) and a [cluster](#).

1. In the upper-right corner of the StreamNative Cloud Console, click your Profile and select **Organizations** to list your created organizations.
2. Click the name of your first organization. In the figure below, the organization name is `org-2`.

Name	Namespace	Status	Actions
↔ org-2	o-2p5g6	Ready	invite teammates create instance

Records per page: 10 ▾ 1-1 of 1

3. On the **Instances** card of the **Dashboard** page, click **New**.
4. Click **Deploy Hosted** to start the instance creation process.
5. On the **Instance Configuration** page, enter a name for your instance, select an infrastructure pool, and select the multi Availability Zone (AZ). The instance name starts with a lowercase letter, contains any combination of lowercase letters (a-z), numbers (0-9), and hyphens (-), and must be 4-10 characters.
6. Click **Cluster Location** to start the cluster creation process.
7. On the **Cluster Location** page, enter a name for your cluster, select the cluster location, and then click **Cluster Size**. The cluster name starts with a lowercase letter, contains any combination of lowercase letters (a-z), numbers (0-9), and hyphens (-), and must be 4-10 characters.
8. On the **Cluster Size** page, configure the cluster, and then click **Payment**.
 - On the **Basic** tab, select custom sizing options.
 - On the **Advanced** tab, in the **Features area**, enable the cluster features you want on your cluster.
9. If needed, on the **Payment** page, in the **Create Payment Method** box, enter a valid credit card number, and then click **Create Payment Method**.

10. Click **Finish**.

The cluster page displays, showing the cluster creation process. The cluster is ready for use after all components have been successfully deployed.

Create a service account

After your cluster is created, you need to create a service account.

1. On the left navigation pane, click **Service Accounts**.
2. Click **Create Service Account**.
3. Enter a name for the service account, and then click **Confirm**.

You should also download the key file to your local machine while you are on the Service Account page.

You will point to the key file when you connect to the cluster.

- In the row of the service account you want to use, in the **Key File** column, click the **Download** icon to download the key file to your local directory.

Authorize the service account

After you have created the service account and downloaded the key file, you need to authorize the service account. For more information, see [Authorization and ACL](#).

1. On the left navigation pane, in the **Admin** section, click **Tenants/Namespaces**.
2. Select the **Public** tenant, then select the **Default** namespace under the tenant.
3. Select the **POLICY** tab.
4. In the **Authorization** area, click **ADD ROLE**, and select the name of the service account you just created in the previous section.
5. In the **Authorization** area, on the drop-down menu below the service name you just added, select the **consume** and **produce** roles. The roles are added to your service account.

You can now continue on to building a Java app, connecting to the cluster, and producing and consuming messages.

Build a Java application

Note

This QuickStart provides you with an example Java app to get you up and running with consuming and producing messages. This is a simple example and is not intended for production environments.

You'll first need to create a Maven project and add the required dependencies.

Create a Maven project

On a terminal window run the following command:

```
mvn archetype:generate -DgroupId=io.streamnative.examples.oauth2 -DartifactId=sncloud-pulsar
cd sncloud-pulsar
```

Add required dependencies

1. In a text editor, open the `pom.xml` file.
2. In the `properties` block, add the following:

```
<pulsar.version>2.11.0</pulsar.version>
```

3. Update the following properties to use Java 1.8:

```
<maven.compiler.target>1.8</maven.compiler.target>
<maven.compiler.source>1.8</maven.compiler.source>
```

4. In the `dependencies` block, add the following:

```
<dependency>
  <groupId>org.apache.pulsar</groupId>
  <artifactId>pulsar-client</artifactId>
  <version>${pulsar.version}</version>
</dependency>
```

Create a producer/consumer

1. Create a `SNCloudProducer` and a `SNCloudConsumer` class in `io.streamnative.examples.oauth2` under the `src/main/java` directory.
2. Return to the StreamNative Cloud Console, and in the **Admin** section of the left navigation pane, click **Pulsar Clients**.
3. On the **Code libraries** tab, follow the setup wizard to get the sample codes for your producer and consumer.
 - a. Keep **Java** selected as the client library and click **Next**.
 - b. Select the service account you created and click **Next**.
 - c. Keep **OAuth2** selected as the authentication type and click **Next**.
 - d. Keep **Maven** selected and click **Next**.
 - e. Select the target tenant, namespace, topic, and subscription, and copy the auto-generated sample codes.
4. Return to your text editor, and paste the sample codes you copied respectively for the producer and consumer in the `SNCloudProducer` and `SNCloudConsumer` classes.
5. In the `String credentialsUrl` parameter, replace `YOUR-KEY-FILE-PATH` with the absolute file path of the key file you downloaded from the Service Account page when you created your service account. There should be three slashes after the word `file`. For example,

```
String credentialsUrl = "file:///Users/yung/.sn/pulsar-client-config/o-2nqre-my-servi
```

Run the clients to produce and consume your first message

1. On a terminal window, run the following command:

```
cd snccloud-pulsar
mvn compile exec:java -Dexec.mainClass=io.streamnative.examples.oauth2.SNCloudConsume
```

2. Open a second terminal window, and run the following command:

```
cd snccloud-pulsar
mvn compile exec:java -Dexec.mainClass=io.streamnative.examples.oauth2.SNCloudProduce
```

3. Return to the first terminal window. You should see the following:

```
Received message hello, Pulsar!
```

You have now produced and consumed your first message.

Next steps

- After you have successfully set up your StreamNative Cloud Console and connected to the cluster, you can learn more about working with StreamNative Cloud by reading through [Console basics](#).
- For basic information about Pulsar, take the FREE [Pulsar Introduction course](#).
- If you want to take a deep dive into Pulsar with hands-on labs and practice go to [StreamNative Academy](#).

Previous

← Concepts

Next

Setup Pulsar Clusters →