- Step 5: Produce and consume data
- Step 6: Use Amazon CloudWatch to view Amazon MSK metrics
- Step 7: Delete the AWS resources created for this tutorial

Step 1: Create an MSK Provisioned cluster

In this step of Getting Started Using Amazon MSK, you create an Amazon MSK cluster.

To create an Amazon MSK cluster using the AWS Management Console

1. Sign in to the AWS Management Console, and open the Amazon MSK console at https://console.aws.amazon.com/msk/home?region=us-east-1#/home/.
2. Choose **Create cluster**.
3. For **Creation method**, leave the **Quick create** option selected. The **Quick create** option lets you create a cluster with default settings.
4. For **Cluster name**, enter a descriptive name for your cluster. For example, `MSKTutorialCluster`.
5. For **General cluster properties**, do the following:

   a. For **Cluster type**, choose **Provisioned**.
   b. Choose an **Apache Kafka version** to run on the brokers. Choose **View version compatibility** to see a comparison table.
   c. For **Broker type**, choose either Standard or Express brokers.
   d. Choose a **Broker size**.
6. From the table under **All cluster settings**, copy the values of the following settings and save them because you need them later in this tutorial:

   - VPC
   - Subnets
   - Security groups associated with VPC
7. Choose **Create cluster**.
8. Check the cluster **Status** on the **Cluster summary** page. The status changes from **Creating** to **Active** as Amazon MSK provisions the cluster. When the status is **Active**, you can connect to the cluster. For more information about cluster status, see Understand MSK Provisioned cluster states.

Next Step

[Step 2: Create an IAM role granting access to create topics on the Amazon MSK cluster](#)

Step 2: Create an IAM role granting access to create topics on the Amazon MSK cluster

In this step, you perform two tasks. The first task is to create an IAM policy that grants access to create topics on the cluster and to send data to those topics. The second task is to create an IAM role and associate this policy with it. In a later step, you create a client machine that assumes this role and uses it to create a topic on the cluster and to send data to that topic.

To create an IAM policy that makes it possible to create topics and write to them

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. On the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. In **Policy editor**, choose **JSON**, and then replace the JSON in the editor window with the following JSON.

   In the following example, replace *region* with the code of the AWS Region where you created your cluster. Replace *Account-ID* with your account ID. Replace *MSKTutorialCluster* with the name of your cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-
ID:cluster/MSKTutorialCluster/7d7131e1-25c5-4e9a-9ac5-ea85bee4da11-14"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:WriteData",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:topic/MSKTutorialCluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:region:Account-ID:group/MSKTutorialCluster/*"
            ]
        }
    ]
}
```

For instructions about how to write secure policies, see the section called “IAM access control”.

5. Choose **Next**.

6. On the **Review and create** page, do the following:

   a. For **Policy name**, enter a descriptive name, such as **msk-tutorial-policy**.

   b. In **Permissions defined in this policy**, review and/or edit the permissions defined in your policy.

   c. (Optional) To help identify, organize, or search for the policy, choose **Add new tag** to add tags as key-value pairs. For example, add a tag to your policy with the key-value pair of **Environment** and **Test**.

      For more information about using tags, see Tags for AWS Identity and Access Management resources in the *IAM User Guide*.

7. Choose **Create policy**.

To create an IAM role and attach the policy to it

1. On the navigation pane, choose **Roles**, and then choose **Create role**.
2. On the **Select trusted entity** page, do the following:

   a. For **Trusted entity type**, choose **AWS service**.

   b. For **Service or use case**, choose **EC2**.

   c. Under **Use case**, choose **EC2**.
3. Choose **Next**.
4. On the **Add permissions** page, do the following:

   a. In the search box under **Permissions policies**, enter the name of the policy that you previously created for this tutorial. Then, choose the box to the left of the policy name.

   b. (Optional) Set a permissions boundary. This is an advanced feature that is available for service roles, but not service-linked roles. For information about setting a permissions boundary, see Creating roles and attaching policies (console) in the *IAM User Guide*.
5. Choose **Next**.
6. On the **Name, review, and create** page, do the following:

   a. For **Role name**, enter a descriptive name, such as **`msk-tutorial-role`**.

   > ⚠ **Important**
   >
   > When you name a role, note the following:
   >
   > - Role names must be unique within your AWS account, and can't be made unique by case.
   >
   >   For example, don't create roles named both **`PRODROLE`** and **`prodrole`**. When a role name is used in a policy or as part of an ARN, the role name is case sensitive, however when a role name appears to customers in the console, such as during the sign-in process, the role name is case insensitive.
   > - You can't edit the name of the role after it's created because other entities might reference the role.

   b. (Optional) For **Description**, enter a description for the role.

c. (Optional) To edit the use cases and permissions for the role, in **Step 1: Select trusted entities** or **Step 2: Add permissions** sections, choose **Edit**.

d. (Optional) To help identify, organize, or search for the role, choose **Add new tag** to add tags as key-value pairs. For example, add a tag to your role with the key-value pair of **`ProductManager`** and **`John`**.

   For more information about using tags, see Tags for AWS Identity and Access Management resources in the *IAM User Guide*.

7. Review the role, and then choose **Create role**.

Next Step

Step 3: Create a client machine

Step 3: Create a client machine

In this step of Get Started Using Amazon MSK, you create a client machine. You use this client machine to create a topic that produces and consumes data. For simplicity, you'll create this client machine in the VPC that is associated with the MSK cluster so that the client can easily connect to the cluster.

To create a client machine

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch instances**.
3. Enter a **Name** for your client machine, such as **`MSKTutorialClient`**.
4. Leave **Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type** selected for **Amazon Machine Image (AMI) type**.
5. Leave the **t2.micro** instance type selected.
6. Under **Key pair (login)**, choose **Create a new key pair**. Enter **`MSKKeyPair`** for **Key pair name**, and then choose **Download Key Pair**. Alternatively, you can use an existing key pair.
7. Expand the **Advanced details** section and choose the IAM role that you created in Step 2: Create an IAM role.
8. Choose **Launch instance**.
9. Choose **View Instances**. Then, in the **Security Groups** column, choose the security group that is associated with your new instance. Copy the ID of the security group, and save it for later.

10. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
11. In the navigation pane, choose **Security Groups**. Find the security group whose ID you saved in the section called "Create a cluster".
12. In the **Inbound Rules** tab, choose **Edit inbound rules**.
13. Choose **Add rule**.
14. In the new rule, choose **All traffic** in the **Type** column. In the second field in the **Source** column, select the security group of your client machine. This is the group whose name you saved after you launched the client machine instance.
15. Choose **Save rules**. Now the cluster's security group can accept traffic that comes from the client machine's security group.

Next Step

Step 4: Create a topic in the Amazon MSK cluster

Step 4: Create a topic in the Amazon MSK cluster

In this step of Getting Started Using Amazon MSK, you install Apache Kafka client libraries and tools on the client machine, and then you create a topic.

> ⚠ **Warning**
>
> Apache Kafka version numbers used in this tutorial are examples only. We recommend that you use the same version of the client as your MSK cluster version. An older client version may be missing certain features and critical bug fixes.

To find the version of your MSK cluster

1. Go to https://eu-west-2.console.aws.amazon.com/msk/
2. Select the MSK cluster.
3. Note the version of Apache Kafka used on the cluster.
4. Replace instances of Amazon MSK version numbers in this tutorial with the version obtained in Step 3.

To create a topic on the client machine

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**. Then select the check box beside the name of the client machine that you created in Step 3: Create a client machine.
3. Choose **Actions**, and then choose **Connect**. Follow the instructions in the console to connect to your client machine.
4. Install Java on the client machine by running the following command:

   ```
   sudo yum -y install java-11
   ```

5. Run the following command to download Apache Kafka.

   ```
   wget https://archive.apache.org/dist/kafka/{YOUR MSK VERSION}/kafka_2.13-{YOUR MSK VERSION}.tgz
   ```

   > **Note**
   >
   > If you want to use a mirror site other than the one used in this command, you can choose a different one on the Apache website.

6. Run the following command in the directory where you downloaded the TAR file in the previous step.

   ```
   tar -xzf kafka_2.13-{YOUR MSK VERSION}.tgz
   ```

7. Go to the `kafka_2.13-{YOUR MSK VERSION}/libs` directory, then run the following command to download the Amazon MSK IAM JAR file. The Amazon MSK IAM JAR makes it possible for the client machine to access the cluster.

   ```
   wget https://github.com/aws/aws-msk-iam-auth/releases/download/v2.3.0/aws-msk-iam-auth-2.3.0-all.jar
   ```

   Using this command, you can also download the latest version of `aws-msk-iam-auth-*-all.jar`.

8. Go to the `kafka_2.13-{YOUR MSK VERSION}/config` directory. Copy the following property settings and paste them into a new file. Name the file **`client.properties`** and save it.

   ```
   security.protocol=SASL_SSL
   sasl.mechanism=AWS_MSK_IAM
   sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required;
   sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
   ```

9. Open the Amazon MSK console at https://console.aws.amazon.com/msk/.

10. Wait for the status of your cluster to become **Active**. This might take several minutes. After the status becomes **Active**, choose the cluster name. This takes you to a page containing the cluster summary.

11. Choose **View client information**.

12. Copy the connection string for the private endpoint.

    You will get three endpoints for each of the brokers. You only need one broker endpoint for the following step.

13. Run the following command, replacing *`BootstrapServerString`* with one of the broker endpoints that you obtained in the previous step.

    ```
    <path-to-your-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-server
     BootstrapServerString --command-config client.properties --replication-factor 3 --
    partitions 1 --topic MSKTutorialTopic
    ```

    If the command succeeds, you see the following message: `Created topic MSKTutorialTopic.`

Next Step

[Step 5: Produce and consume data]()

Step 5: Produce and consume data

In this step of [Get Started Using Amazon MSK](), you produce and consume data.

To produce and consume messages

1. Run the following command to start a console producer. Replace *BootstrapServerString* with the plaintext connection string that you obtained in Create a topic. For instructions on how to retrieve this connection string, see Getting the bootstrap brokers for an Amazon MSK cluster.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-producer.sh --
   broker-list BootstrapServerString --producer.config client.properties --
   topic MSKTutorialTopic
   ```

2. Enter any message that you want, and press **Enter**. Repeat this step two or three times. Every time you enter a line and press **Enter**, that line is sent to your Apache Kafka cluster as a separate message.
3. Keep the connection to the client machine open, and then open a second, separate connection to that machine in a new window.
4. In the following command, replace *BootstrapServerString* with the plaintext connection string that you saved earlier. Then, to create a console consumer, run the following command with your second connection to the client machine.

   ```
   <path-to-your-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-
   server BootstrapServerString --consumer.config client.properties --
   topic MSKTutorialTopic --from-beginning
   ```

   You start seeing the messages you entered earlier when you used the console producer command.
5. Enter more messages in the producer window, and watch them appear in the consumer window.

Next Step

Step 6: Use Amazon CloudWatch to view Amazon MSK metrics

Step 6: Use Amazon CloudWatch to view Amazon MSK metrics

In this step of Getting Started Using Amazon MSK, you look at the Amazon MSK metrics in Amazon CloudWatch.

To view Amazon MSK metrics in CloudWatch

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Choose the **All metrics** tab, and then choose **AWS/Kafka**.
4. To view broker-level metrics, choose **Broker ID, Cluster Name**. For cluster-level metrics, choose **Cluster Name**.
5. (Optional) In the graph pane, select a statistic and a time period, and then create a CloudWatch alarm using these settings.

Next Step

Step 7: Delete the AWS resources created for this tutorial

Step 7: Delete the AWS resources created for this tutorial

In the final step of Getting Started Using Amazon MSK, you delete the MSK cluster and the client machine that you created for this tutorial.

To delete the resources using the AWS Management Console

1. Open the Amazon MSK console at https://console.aws.amazon.com/msk/.
2. Choose the name of your cluster. For example, **MSKTutorialCluster**.
3. Choose **Actions**, then choose **Delete**.
4. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
5. Choose the instance that you created for your client machine, for example, **`MSKTutorialClient`**.
6. Choose **Instance state**, then choose **Terminate instance**.

To delete the IAM policy and role

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. On the navigation pane, choose **Roles**.
3. In the search box, enter the name of the IAM role that you created for this tutorial.
4. Choose the role. Then choose **Delete role**, and confirm the deletion.
5. On the navigation pane, choose **Policies**.

6. In the search box, enter the name of the policy that you created for this tutorial.
7. Choose the policy to open its summary page. On the policy's **Summary** page, choose **Delete policy**.
8. Choose **Delete**.

Amazon MSK: How it works

Amazon MSK is a fully managed Apache Kafka service that makes it easy to build and run applications that use Apache Kafka to process streaming data. This guide provides information to help developers understand how Amazon MSK works and how to use it effectively in their applications.

At a high level, Amazon MSK provides a fully managed Apache Kafka cluster that is provisioned and operated by AWS. This means that you don't have to worry about provisioning EC2 instances, configuring network settings, managing Kafka brokers, or performing ongoing maintenance tasks. Instead, you can focus on building your application and let Amazon MSK handle the infrastructure. Amazon MSK automatically provisions the necessary compute, storage, and network resources, and provides features like automatic scaling, high availability, and failover to ensure that your Kafka cluster is reliable and highly available. This guide covers the key components of Amazon MSK and how you can use it to build streaming data applications.

Manage your Provisioned cluster

An Amazon MSK cluster is the primary Amazon MSK resource that you can create in your account. The topics in this section describe how to perform common Amazon MSK operations. For a list of all the operations that you can perform on an MSK cluster, see the following:

- The AWS Management Console
- The Amazon MSK API Reference
- The Amazon MSK CLI Command Reference

Topics