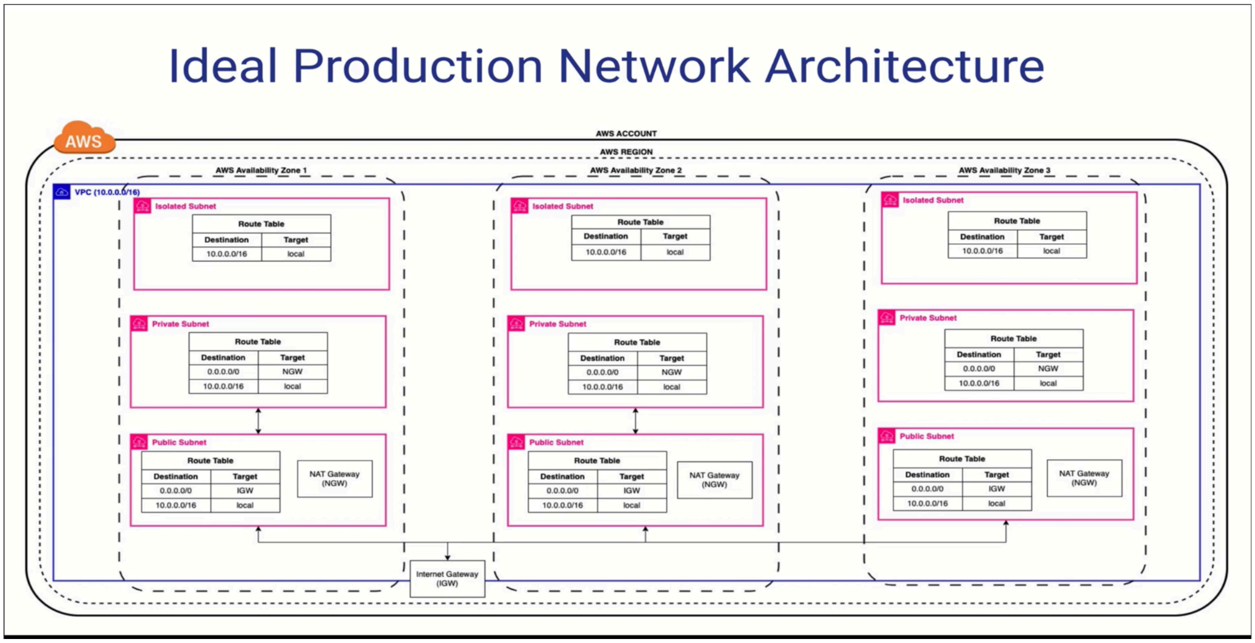
Step 1: Create an Amazon VPC followed by a MSK cluster

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
2. Choose Create cluster.
3. Select creation method provisioned and custom
4. Choose Create cluster.
5. 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, we can connect to the cluster.



Isolated Subnet

Route Table	
Destination	Target
10.0.0.0/16	local

Private Subnet

Route Table	
Destination	Target
0.0.0.0/0	NGW
10.0.0.0/16	local

Public Subnet

Route Table	
Destination	Target
0.0.0.0/0	IGW
10.0.0.0/16	local

Create a VPC with two private subnets and one public subnet. Place the MSK cluster in the private subnets.

Step 2: Create an IAM role

- Create a policy that allows access to the MSK cluster and perform actions
- Attach the policy to a role
- Attach the role to the client ec2 instance (The one in the next step)

```
{
    "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/*"
            ]
        },
        {
            "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/*"
            ]
        }
    ]
}
```

Step 3: Create a client machine

- Create an ec2 instance in the same VPC but in the public subnet
- Create a new security group, allow SSH port in the rules to access the instance
- Copy the security group ID of this security group. In the inbound rules of the MSK security group, choose All traffic in the Type column. In the second field in the Source column, select the security group of the ec2 instance

Step 4: Create a topic

- sudo apt install openjdk-11-jdk -y

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

- Go to the `kafka_2.13-{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/v1.1.1/aws-msk-iam-auth-1.1.1-all.jar
- Go to the `kafka_2.13-{MSK VERSION}/bin` directory. Copy the following 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

1. Open the Amazon MSK console
2. Choose View client information.
3. Copy the connection string for the private endpoint (IAM)
4. Run the following command, replacing `BootstrapServerString` with one of the broker endpoints that we obtained in the previous step.
5. `<path-to-kafka-installation>/bin/kafka-topics.sh --create --bootstrap-server BootstrapServerString --command-config client.properties --replication-factor 3 --partitions 1 --topic MSKDemoTopic`
6. If the command succeeds, we see the following message: `Created topic MSKTutorialTopic.`

Step 5: 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

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

2. Enter any message, and press Enter. Repeat this step two or three times. Every time we enter a line and press Enter, that line is sent to the 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 we saved earlier. Then, to create a console consumer, run the following command with the second connection to the client machine.

5.
   ```
   <path-to-kafka-installation>/bin/kafka-console-consumer.sh --bootstrap-server BootstrapServerString --consumer.config client.properties --topic MSKDemoTopic --from-beginning
   ```
6. We start seeing the messages we entered earlier when we used the console producer command.
7. Enter more messages in the producer window, and watch them appear in the consumer wind