Solution

Before proceeding forward and execute its steps, let's check the steps we're going to perform, you can directly jump to any topic for your reference if you want:

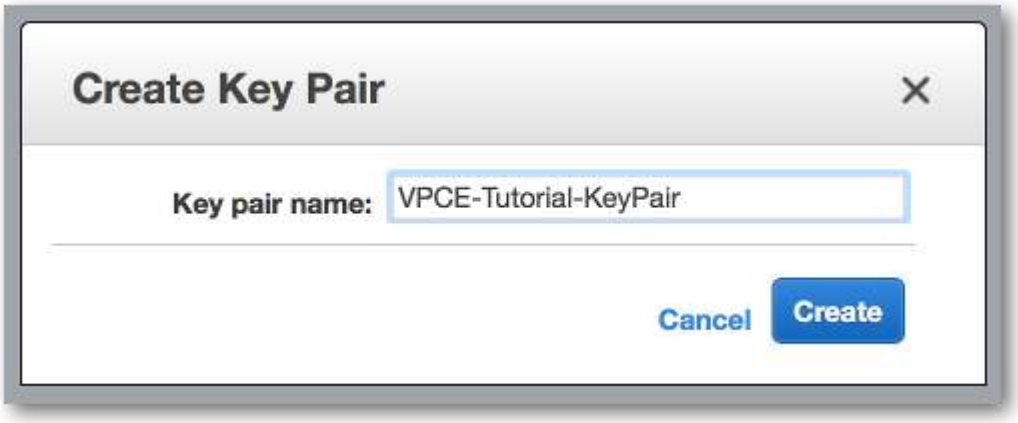
Now let's get started with its steps.

Step 1: Create an Amazon EC2 Key Pair

A key pair is used to log in to an Amazon EC2 instance. It consists of a public key that's used to encrypt your login information, and a private key that's used to decrypt it. When you create a key pair, you download a copy of the private key. Later in this tutorial, you use the key pair to log in to an Amazon EC2 instance. To log in, you specify the name of the key pair, and you provide the private key.

To create the key pair

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation menu on the left, find the **Network & Security** section. Then, choose **Key Pairs**.
3. Choose **Create Key Pair**.
4. In the **Create Key Pair** window, for **Key pair name**, type **VPCE-Tutorial-KeyPair**. Then, choose **Create**.



5. The private key file is automatically downloaded by your browser. Save it in a safe place. Amazon EC2 gives the file an extension of .pem.
6. (Optional) If you're using an SSH client on a Mac or Linux computer to connect to your instance, use the chmod command to set the permissions of your private key file so that only you can read it:
    a. Open a terminal and navigate to the directory that contains the private key:

    ```
    $ cd /filepath_to_private_key/
    ```

    b. Set the permissions by using the following command:

    ```
    $ chmod 400 VPCE-Tutorial-KeyPair.pem
    ```

Step 2: Create the AWS Resources

To set up the infrastructure that supports this tutorial, you use an AWS CloudFormation *template*. A template is a file that acts as a blueprint for building AWS resources, such as Amazon EC2 instances and Amazon SNS topics. The template for this tutorial is provided on GitHub for you to download.

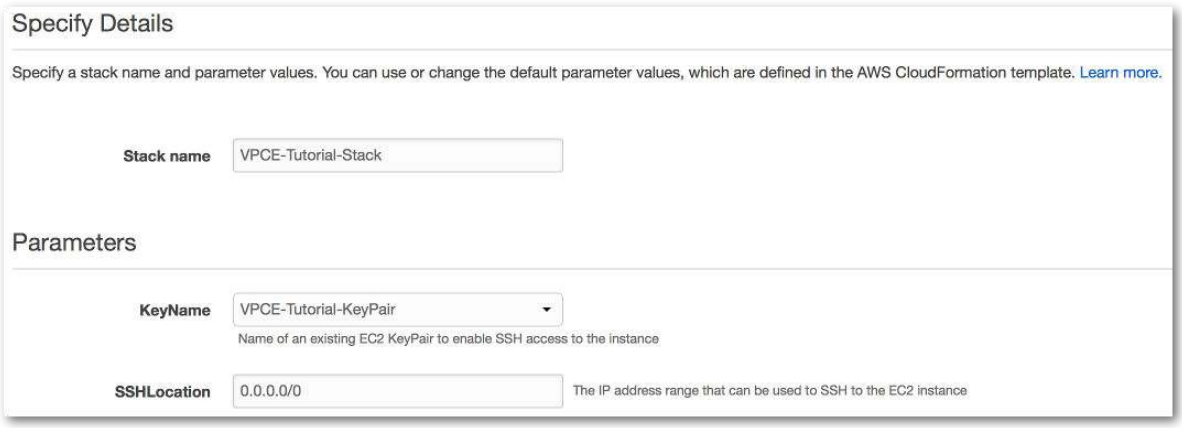
You provide the template to AWS CloudFormation, and AWS CloudFormation provisions the resources that you need as a *stack* in your AWS account. A stack is a collection of resources that you manage as a single unit. When you finish the tutorial, you can use AWS CloudFormation to delete all of the resources in the stack at once. These resources don't remain in your AWS account, unless you want them to.

The stack for this tutorial includes the following resources:

- A VPC and the associated networking resources, including a subnet, a security group, an internet gateway, and a route table.
- An Amazon EC2 instance that's launched into the subnet in the VPC.
- An Amazon SNS topic.
- Two AWS Lambda functions. These functions receive messages that are published to the Amazon SNS topic, and they log events in CloudWatch Logs.
- Amazon CloudWatch metrics and logs.
- An IAM role that allows the Amazon EC2 instance to use Amazon SNS, and an IAM role that allows the Lambda functions to write to CloudWatch logs.

To create the AWS resources

1. Download the template file from their Github site
2. Sign in to the AWS Management Console
3. Choose **Create Stack**.
4. On the **Select Template** page, choose **Upload a template to Amazon S3**, select the file, and choose **Next**.
5. On the **Specify Details** page, specify stack and key names:
    a. For **Stack name**, type **VPCE-Tutorial-Stack.**
    b. For **KeyName**, choose **VPCE-Tutorial-KeyPair**.
    c. For **SSHLocation**, keep the default value of **0.0.0.0/0**.



    d. Choose **Next**.
6. On the **Options** page, keep all of the default values, and choose **Next**.
7. On the **Review** page, verify the stack details.
8. Under **Capabilities**, select the check box that acknowledges that AWS CloudFormation might create IAM resources with custom names.
9. Choose **Create**.

The AWS CloudFormation console opens the **Stacks** page. The VPCE-Tutorial-Stack has a status of **CREATE_IN_PROGRESS**. In a few minutes, after the creation process completes, the status changes to **CREATE_COMPLETE**.

Tip

Choose the **Refresh** button to see the latest stack status.

Step 3: Confirm That Your Amazon EC2 Instance Lacks Internet Access

The Amazon EC2 instance that was launched in your VPC in the previous step lacks internet access. It disallows outbound traffic, and it's unable to publish messages to Amazon SNS. Verify this by logging in to the instance. Then, attempt to connect to a public endpoint, and attempt to message Amazon SNS.

At this point in the tutorial, the publish attempt fails. In a later step, after you create a VPC endpoint for Amazon SNS, your publish attempt succeeds.

To connect to your Amazon EC2 instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation menu on the left, find the **Instances** section. Then, choose **Instances**.
3. In the list of instances, select **VPCE-Tutorial-EC2Instance**.
4. Copy the hostname that's provided in the **Public DNS (IPv4)** column.

5. Open a terminal. From the directory that contains the key pair, connect to the instance by using the following command, where *instance-hostname* is the hostname that you copied from the Amazon EC2 console:

```
$ ssh -i VPCE-Tutorial-KeyPair.pem ec2-user@instance-hostname
```

To verify that the instance lacks internet connectivity

- In your terminal, attempt to connect to any public endpoint, such as amazon.com:

```
$ ping amazon.com
```

Because the connection attempt fails, you can cancel at any time (Ctrl + C on Windows or Command + C on macOS).

To verify that the instance lacks connectivity to Amazon SNS

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/.
2. In the navigation menu on the left, choose **Topics**.
3. On the **Topics** page, copy the Amazon Resource Name (ARN) for the topic **VPCE-Tutorial-Topic**.
4. In your terminal, attempt to publish a message to the topic:

```
$ aws sns publish --region aws-region --topic-arn sns-topic-arn --message "Hello"
```

Because the publish attempt fails, you can cancel at any time.

Step 4: Create an Amazon VPC Endpoint for Amazon SNS

To connect the VPC to Amazon SNS, you define an interface VPC endpoint. After you add the endpoint, you can log in to the Amazon EC2 instance in your VPC, and from there you can use the Amazon SNS API. You can publish messages to the topic, and the messages are published privately. They stay within the AWS network, and they don't travel the public internet.

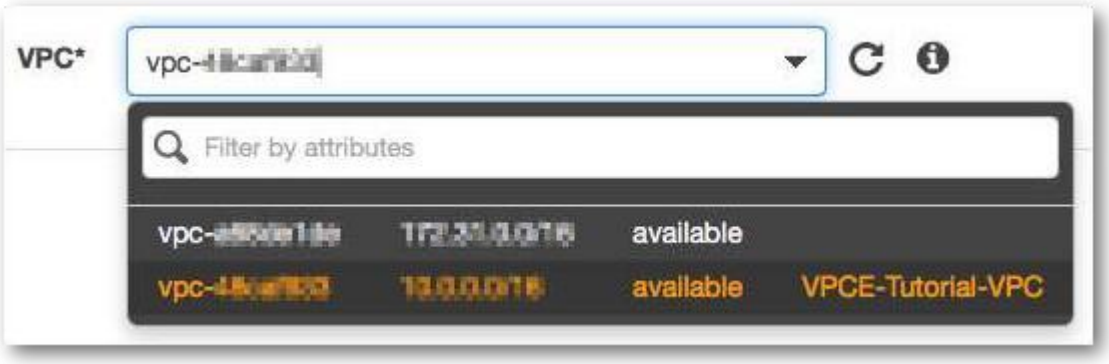
Note that the instance still lacks access to other AWS services and endpoints on the internet.

To create the endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation menu on the left, choose **Endpoints**.
3. Choose **Create Endpoint**.
4. On the **Create Endpoint** page, for **Service category**, keep the default choice of **AWS services**.
5. For **Service Name**, choose the service name for Amazon SNS.

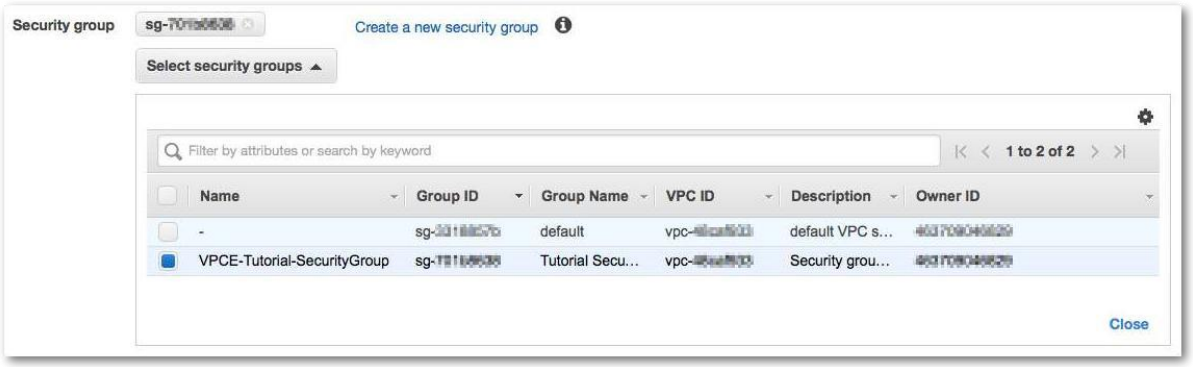
The service names vary based on the chosen region. For example, if you chose US East (N. Virginia), the service name is **com.amazonaws.*us-east-1*.sns**.

6. For **VPC**, choose the VPC that has the name **VPCE-Tutorial-VPC**.

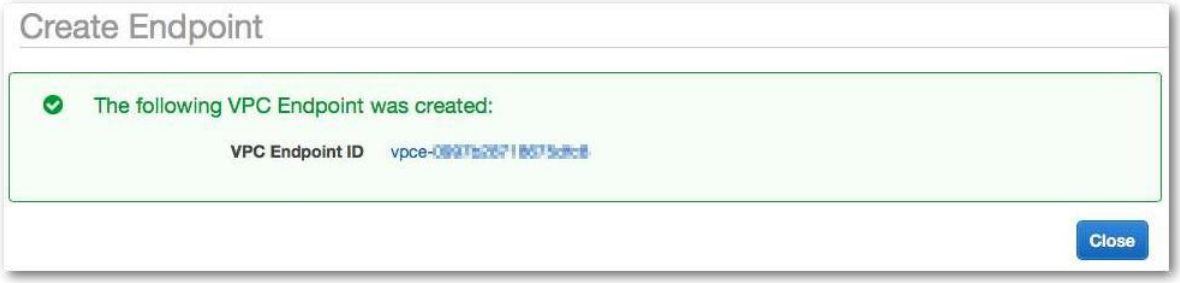


7. For **Subnets**, select the subnet that has *VPCE-Tutorial-Subnet* in the subnet ID.

8. For **Enable Private DNS Name**, select **Enable for this endpoint**.
9. For **Security group**, choose **Select security group**, and select the one named **VPCE-Tutorial-SecurityGroup**.

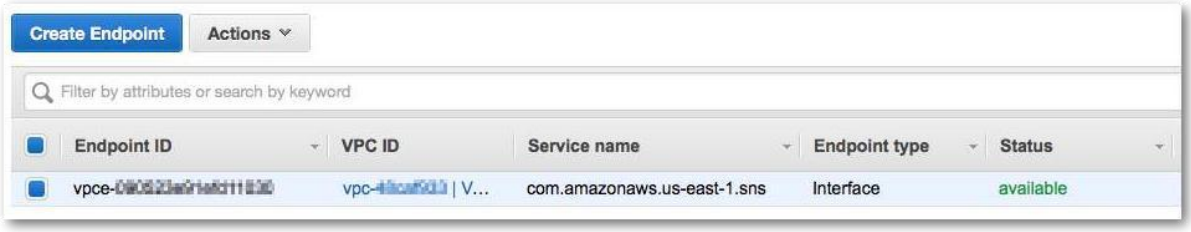


10. Choose **Create endpoint**. The Amazon VPC console confirms that a VPC endpoint was created.



11. Choose **Close**.

    The Amazon VPC console opens the **Endpoints** page. The new endpoint has a status of **pending**. In a few minutes, after the creation process completes, the status changes to**available**.



Step 5: Publish a Message to Your Amazon SNS Topic

Now that your VPC includes an endpoint for Amazon SNS, you can log in to the Amazon EC2 instance and publish messages to the topic.

To publish a message

1. If your terminal is no longer connected to your Amazon EC2 instance, connect again:

```
$ ssh -i VPCE-Tutorial-KeyPair.pem ec2-user@instance-hostname
```

2. Run the same command that you did previously to publish a message to your Amazon SNS topic. This time, the publish attempt succeeds, and Amazon SNS returns a message ID:

```
$ aws sns publish --region aws-region --topic-arn sns-topic-arn --message "Hello"
    {
      "MessageId": "5b111270-d169-5be6-9042-410dfc9e86de"
    }
```

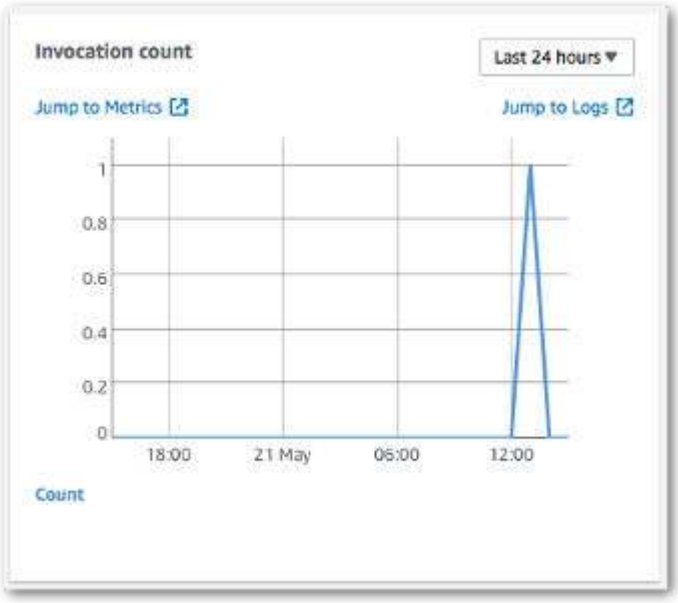
Step 6: Verify Your Message Deliveries

When the Amazon SNS topic receives a message, it fans out the message by sending it to the two subscribing Lambda functions. When these functions receive the message, they log the event to CloudWatch logs. To verify that your message delivery succeeded, check that the functions were invoked, and check that the CloudWatch logs were updated.

To verify that the Lambda functions were invoked

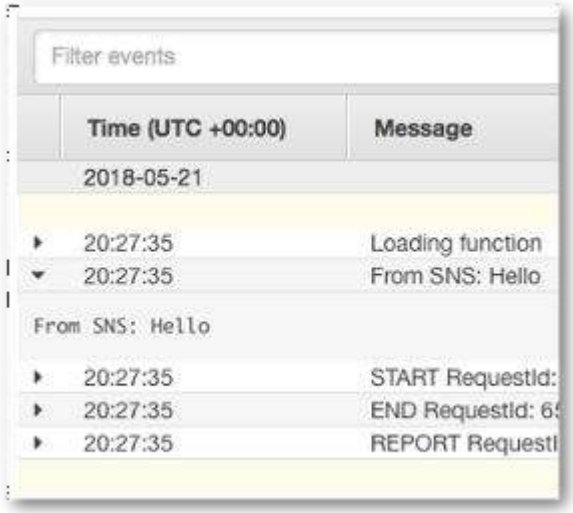
1. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
2. On the **Functions** page, choose **VPCE-Tutorial-Lambda-1**.
3. Choose **Monitoring**.
4. Check the **Invocation count** graph. This graph shows the number of times that the Lambda function has been run.

   The invocation count matches the number of times you published a message to the topic.



To verify that the CloudWatch logs were updated

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation menu on the left, choose **Logs**.
3. Check the logs that were written by the Lambda functions:
    a. Choose the **/aws/lambda/VPCE-Tutorial-Lambda-1/** log group.
    b. Choose the log stream.
    c. Check that the log includes the entry From SNS: Hello.



    d. Choose **Log Groups** at the top of the console to return the **Log Groups** page. Then, repeat the preceding steps for the /aws/lambda/VPCE-Tutorial-Lambda-2/ log group.

Congratulations! By adding an endpoint for Amazon SNS to a VPC, you were able to publish a message to a topic from within the network that's managed by the VPC. The message was published privately without being exposed to the public internet.

Step 7: Clean Up

Unless you want to retain the resources that you created for this tutorial, you can delete them now. By deleting AWS resources that you're no longer using, you prevent unnecessary charges to your AWS account.

First, delete your VPC endpoint by using the Amazon VPC console. Then, delete the other resources that you created by deleting the stack in the AWS CloudFormation console. When you delete a stack, AWS CloudFormation removes the stack's resources from your AWS account.

To delete your VPC endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation menu on the left, choose **Endpoints**.
3. Select the endpoint that you created.
4. Choose **Actions**, and then choose **Delete Endpoint**.
5. In the **Delete Endpoint** window, choose **Yes, Delete**.

   The endpoint status changes to **deleting**. When the deletion completes, the endpoint is removed from the page.

To delete your AWS CloudFormation stack

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. Select the stack **VPCE-Tutorial-Stack**.
3. Choose **Actions**, and then choose **Delete Stack**.
4. In the **Delete Stack** window, choose **Yes, Delete**.

   The stack status changes to **DELETE_IN_PROGRESS**. When the deletion completes, the stack is removed from the page.