

Home / AWS / Guided Lab / How to create a SQS Queue using Terraform

# How to create a SQS Queue using Terraform

Level: Fundamental

Amazon SQS   Amazon Web Services   Terraform



0h 21m 4s left



End Lab

Open Console

Validation

## Lab Credentials

User Name ⓘ

Whiz\_User\_80425.36890280



Password ⓘ

d46a21cb-7a10-4cc1-ba1e-bfa4f4bcb8f4



Access Key ⓘ

AKIASDQKNNDK4KTNDKPN



Secret Key ⓘ

vbKLweHm48+H+NyIGHGY5EiYu+/SwOxD8sSINleO






## Lab Resources


No Lab Resources Found

## Support Documents

## 1. FAQs and Troubleshooting

### Need help?

-  How to use Hands on Lab
-  Troubleshooting Lab
-  FAQs

[Submit Feedback](#)[Share](#)[Lab Overview](#)[Lab Steps](#)[Lab Validation](#) Cloud Developer Serverless, Infrastructure

# Lab Steps

## Task 1: Sign in to the AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.

2. On the AWS sign-in page,

- Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
- Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in the AWS Console and click on the **Sign in** button

3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1**.

## Task 2: Setup Visual Studio Code

1. Open the Visual Studio code.

2. If you have already installed and using Visual Studio code, open a new window.

3. A new window will open a new file and release notes page (only if you have installed or updated Visual Studio Code recently). Close the Release Notes tab.

4. Open Terminal by selecting View from the Menu bar and choose Terminal.
5. It may take up to 2 minutes to open the terminal window.
6. Once the terminal is ready, let us navigate to the Desktop.

```
cd Desktop
```



7. Create a new folder by running the below command.

```
mkdir task_10095_sqs
```



8. Change your present working directory to use the newly created folder by running the below command:

```
cd task_10095_sqs
```

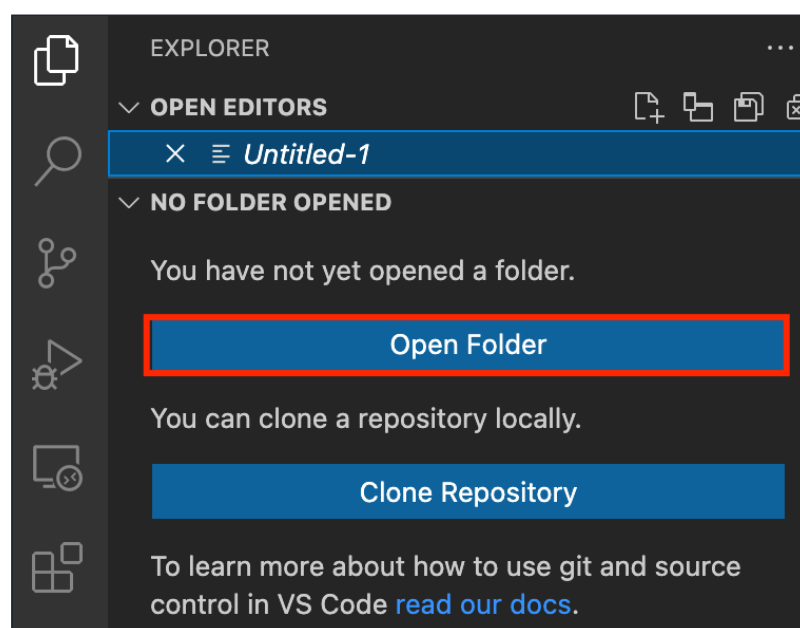


9. Get the location of the present working directory by running the below command:

```
pwd
```



10. Note down the location, as you will open the same in the next steps.
11. Now click on the first icon Explorer present on the left sidebar.
12. Click on the button called Open folder and navigate to the location of folder **task\_10095\_sqs**.



13. (Optional) Click on the Authorize button to allow Visual Studio Code to use the task\_10095\_sqs folder. This will only be asked when you have been using Visual Studio

code for a while as you are allowing a new folder to be accessed by VSC.

14. Visual Studio Code is now ready to use.

### Task 3: Create a variables file

In this task, you will create variable files where you will declare all the global variables with a short description and a default value.

1. To create a variable file, expand the folder **task\_10095\_sqs** and click on the **New File** icon to add the file.
2. Name the file as **variables.tf** and press **Enter** to save it.
3. **Note:** Don't change the location of the new file, keep it default, i.e. inside the **task\_10095\_sqs** folder.
4. Paste the below contents in **variables.tf** file.

```
variable "access_key" {  
    description = "Access key to AWS console"  
}  
variable "secret_key" {  
    description = "Secret key to AWS console"  
}  
variable "region" {  
    description = "AWS region"  
}
```



5. In the above content, you are declaring a variable called, access\_key, secret\_key, and region with a short description of all 3.
6. After pasting the above contents, save the file by pressing **ctrl + S**.
7. Now expand the folder **task\_10095\_sqs** and click on the **New File** icon to add the file.
8. Name the file as **terraform.tfvars** and press **Enter** to save it.
9. Paste the below content into the **terraform.tfvars** file.


```
region = "us-east-1"  
access_key = "<YOUR AWS CONSOLE ACCESS ID>"  
secret_key = "<YOUR AWS CONSOLE SECRET KEY>"
```



10. In the above code, you are defining the dynamic values of variables declared earlier.
11. Replace the values of access\_key and secret\_key by copying from the lab page.



12. After replacing the values of `access_key` and `secret_key`, save the file by pressing **Ctrl + S**.



```
variables.tf  terraform.tfvars X
terraform.tfvars > secret_key
1  region = "us-east-1"
2
3  access_key = "[REDACTED]"
4
5  secret_key = "[REDACTED]"
```

#### Task 4: Create an SQS queue in main.tf file

In this task, you will create a **main.tf** file where you will add details of the provider and resources.

1. To create a **main.tf** file, expand the folder **task\_10095\_sqs**, and click on the **New File** icon to add the file.
2. Name the file as **main.tf** and press **Enter** to save it.
3. Paste the below content into the **main.tf** file.

```
provider "aws" {
  region      = var.region
  access_key  = var.access_key
  secret_key  = var.secret_key
}
```



4. In the above code, you are defining the provider as **aws**.
5. Next, we want to tell Terraform to create a **FIFO** and **Standard** SQS Queue
6. To create a **FIFO** queue Paste the below content into the **main.tf** file after the provider.

```
##### Creating FIFO Queue #####
resource "aws_sqs_queue" "queue" {
  name = "MyWhizQueue.fifo"
  fifo_queue = true
  content_based_deduplication = true
}
```



7. Now to create a **Standard** queue add another block of code just below the **FIFO** queue code into the **main.tf** file.





```
##### Creating Standard Queue #####
resource "aws_sqs_queue" "queue2" {
  name = "MyWhizQueue"
}
```

```
main.tf > resource "aws_sqs_queue" "queue2"
1  provider "aws" {
2      region      = "${var.region}"
3      access_key  = "${var.access_key}"
4      secret_key  = "${var.secret_key}"
5  }
6  ##### Creating FIFO Queue #####
7  resource "aws_sqs_queue" "queue" {
8      name          = "MyWhizQueue.fifo"
9      fifo_queue    = true
10     content_based_deduplication = true
11 }
12 ##### Creating Standard Queue #####
13 resource "aws_sqs_queue" "queue2" {
14     name          = "MyWhizQueue"
15 }
```

8. Save the file by pressing **Ctrl + S**.

## Task 5: Create an Output file

In this task, you will create an **output.tf** file where you will add details of the provider and resources.

1. To create an **output.tf** file, expand the folder **task\_10095\_sqs**, and click on the **New File** icon to add the file.
2. Name the file as **output.tf** and press **Enter** to save it.
3. Paste the below content into the **output.tf** file.

```
output "sqs-queue-arn" {
  value = aws_sqs_queue.queue.arn
}
```



4. In the above code, we will extract the sqs queue arn to confirm that they are created.

## Task 6: Confirm the installation of Terraform by checking the version

1. In the Visual Studio Code, open Terminal by selecting **View** from the Menu bar and choose **Terminal**.
2. If you are not in the newly created folder change your present working directory by running the below command.

```
cd task_10095_sqs
```



3. To confirm the installation of Terraform, run the below command to check the version:

```
terraform version
```



4. If you are getting output as command not found: terraform, this means that terraform is not installed on your system, To install terraform follow the official guide link provided in the Prerequisite section above.

## Task 7: Apply terraform configurations

1. Initialize Terraform by running the below command,

```
terraform init
```



```
PS C:\Users\██████\Desktop\task_10095> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.39.0...
- Installed hashicorp/aws v4.39.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

**Note:** terraform init will check for all the plugin dependencies and download them required, this will be used for creating a deployment plan

2. To generate the action plans run the below command,

```
terraform plan
```



```
Plan: 2 to add, 0 to change, 0 to destroy.
```

#### Changes to Outputs:

```
+ sqs-queue-arn = (known after apply)
+ sqs-queue2-arn = (known after apply)
```

3. To create all the resources declared in the main.tf configuration file, run the below command,

```
terraform apply
```



4. Approve the creation of all the resources by entering **yes**.
5. IDs of all the resources created by Terraform will be visible there.

```
Enter a value: yes
aws_sqs_queue.queue: Creating...
aws_sqs_queue.queue2: Creating...
aws_sqs_queue.queue: Still creating... [10s elapsed]
aws_sqs_queue.queue2: Still creating... [10s elapsed]
aws_sqs_queue.queue2: Still creating... [20s elapsed]
aws_sqs_queue.queue: Still creating... [20s elapsed]
aws_sqs_queue.queue: Still creating... [30s elapsed]
aws_sqs_queue.queue2: Still creating... [30s elapsed]
aws_sqs_queue.queue: Creation complete after 31s [id=https://sqs.us-east-1.amazonaws.com/461815519783/MyWhizQueue.fifo]
aws_sqs_queue.queue2: Creation complete after 31s [id=https://sqs.us-east-1.amazonaws.com/461815519783/MyWhizQueue]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

sqs-queue-arn = "arn:aws:sqs:us-east-1:461815519783:MyWhizQueue.fifo"
sqs-queue2-arn = "arn:aws:sqs:us-east-1:461815519783:MyWhizQueue"
```

## Task 8: Check the resources in the AWS Console

1. Make sure you are in the **US East (N. Virginia) us-east-1** Region.
2. Navigate to **SQS** by clicking on **Services** on the top, then click on **SQS** in the **Application Integration** section.
3. You can see two queues are created successfully.



Queues (2)							
<input type="text" value="Search queues by prefix"/>							
	Name ▲	Type ▼	Created ▼	Messages available ▼	Messages in flight ▼	Encryption ▼	
<input type="radio"/>	MyWhizQueue	Standard	11/16/2022, 20:28:09 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	
<input type="radio"/>	MyWhizQueue.fifo	FIFO	11/16/2022, 20:28:09 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	

## Task 9: Send and receive messages

In this task, we'll be sending a message from our FIFO and Standard queue.

1. **Select** the FIFO queue and click on the **Send and Receive Messages** button.
2. On the Send and Receive Messages page,
  - Message body: Enter **Hello!! From Whizlabs**
  - Message group ID: Enter **group123**
  - Message deduplication ID: Enter **id123**
  - Now Click on the **Send message** button.

Send message [Info](#)

Clear content

Send message

Message body

Enter the message to send to the queue.

Hello!! From Whizlabs

Message group ID

The tag that specifies that a message belongs to a specific message group.

group123

Message deduplication ID - *Optional*

The token used for deduplication of messages within the deduplication interval.

id123

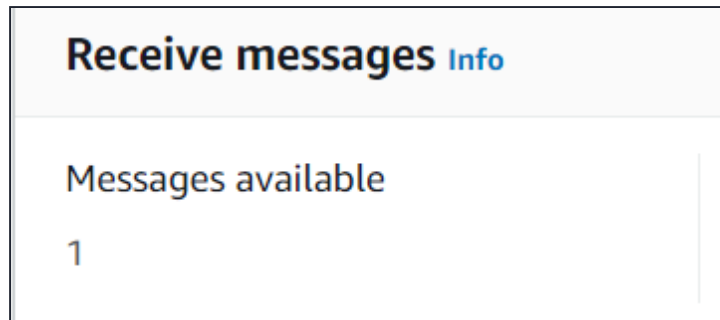
▶ Message attributes - *Optional* [Info](#)

3. Once you have sent a message you get an acknowledgment.

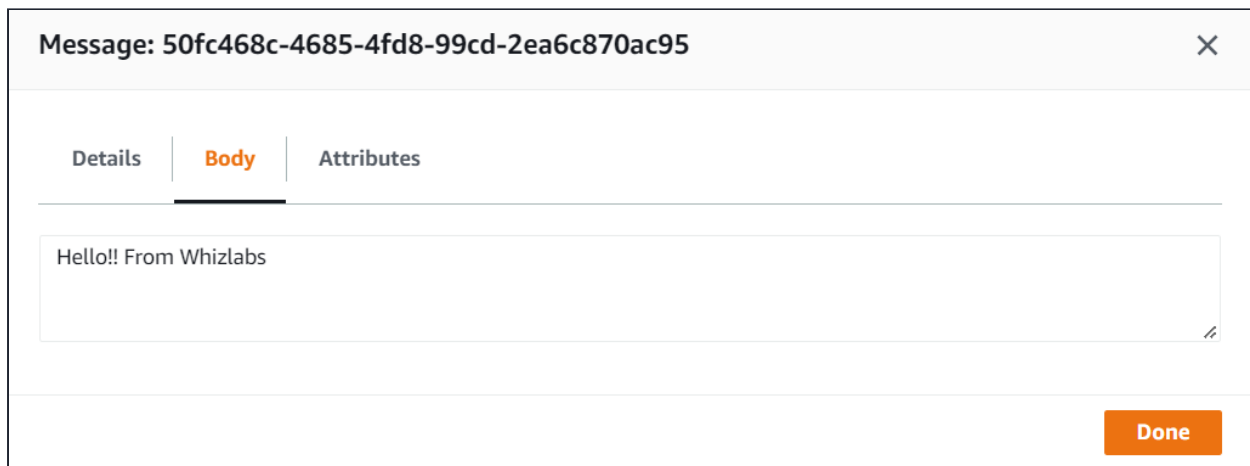
Your message has been sent and is ready to be received.

View details

4. After sending the message, scroll down and we can see that the Message available changes to **1**.



5. Similarly, send a few more messages with different Message bodies, Message group IDs, and Message deduplication.
6. After sending the messages, the Message available changes to the number of messages sent.
7. Now, to get all the messages we have sent, click on **Poll for messages** in the Receive messages section.
8. Then Click on the message to see it's body:



9. Hence, we have successfully sent and received messages in the console.
10. Similarly, we'll try with the Standard queue for sending the message.
11. Select the Standard queue and Click on the **Send and receive messages** button.

Queues (2)								
<input type="text" value="Search queues by prefix"/>			<a href="#">Send and receive messages</a>			<a href="#">Actions</a>		<a href="#">Create queue</a>
Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication		
<input checked="" type="radio"/> <a href="#">MyWhizQueue</a>	Standard	11/16/2022, 20:28:09 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	-		
<input type="radio"/> <a href="#">MyWhizQueue.fifo</a>	FIFO	11/16/2022, 20:28:09 GMT+5:30	0	0	Amazon SQS key (SSE-SQS)	Enabled		

## 12. In the Send and Receive Messages page

- Message body: Enter **Message from Standard Queue**
- Now Click on the **Send message** button.

Send message [Info](#)

Clear content

Send message

Message body

Enter the message to send to the queue.

Delivery delay [Info](#)

Seconds

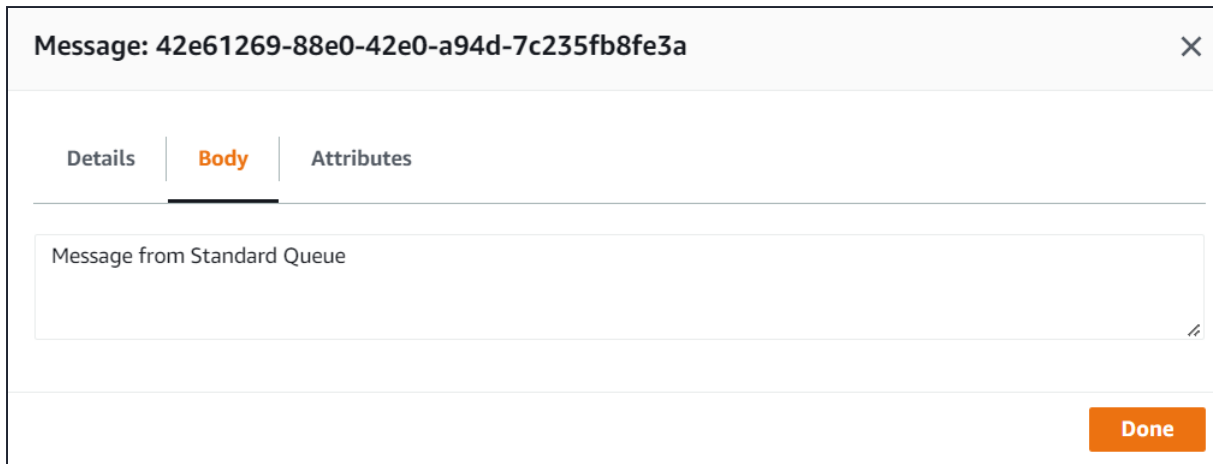
Should be between 0 seconds and 15 minutes.

## 13. Once you have sent a message you get an acknowledgment.

14. After sending the message, scroll down and we can see that the Message available changes to **1**.

15. Now, to get all the messages we have sent, click on **Poll for messages** in the Receive messages section.

16. Then click on the message to see it's body:

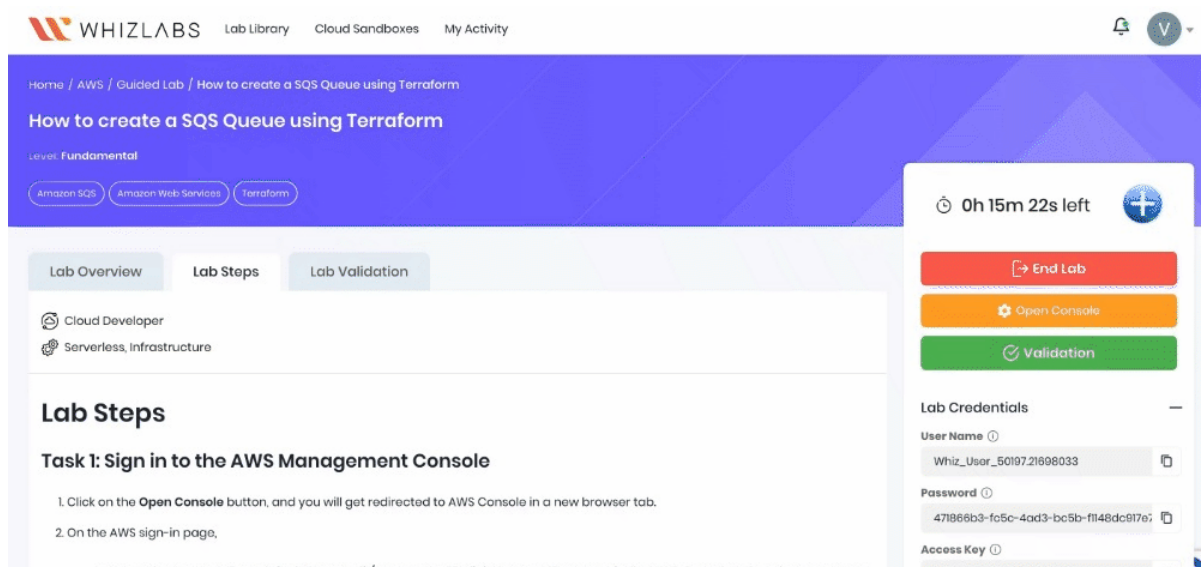


## Do you know ?

The maximum size for a single message in SQS is 256 KB for Standard Queues and 256 KB for FIFO Queues.

### Task 10: Validation of the Lab

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :



### Task 11: Delete AWS Resources

1. To delete the resources, open Terminal again.

2. Run the below command to delete all the resources.

```
terraform destroy
```



3. Enter **yes** to confirm the deletion. You can see the **Destroy complete!** message.

```
Do you really want to destroy all resources?  
Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.  
  
Enter a value: yes  
  
aws_sqs_queue.queue2: Destroying... [id=https://sqs.us-east-1.amazonaws.com/461815519783/MyWhizQueue]  
aws_sqs_queue.queue: Destroying... [id=https://sqs.us-east-1.amazonaws.com/461815519783/MyWhizQueue.fifo]  
aws_sqs_queue.queue2: Destruction complete after 8s  
aws_sqs_queue.queue: Destruction complete after 8s  
  
Destroy complete! Resources: 2 destroyed.
```

## Completion and Conclusion

- You have successfully set up the Visual Studio Code editor.
- You have successfully created variables.tf and terraform.tfvars files.
- You have successfully created FIFO and standard queue using the terraform.
- You have successfully executed the terraform configuration commands to create the resources.
- You have successfully checked all the resources created by opening the Console.
- You have successfully Sent and Received messages
- You have deleted all the resources.

## End Lab

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs dashboard.

[About Us](#) [Subscription](#) [Instructions and Guidelines](#) [FAQ's](#) [Contact Us](#)



© 2023, Whizlabs Software Pvt. Ltd.

