

1 Create an IAM policy and execution role for your Lambda function

1. Create an IAM policy using the JSON policy editor. Copy and paste the following JSON policy document into the policy editor

IAM Policy Start/Stop

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

2. Create an IAM role for Lambda.



When attaching a permissions policy to Lambda, make sure that you choose the IAM policy you just created.


2 Create Lambda functions that stop and start your EC2 instances

1. In the AWS Lambda console, choose **Create function**.
2. Choose **Author from scratch**.
3. Under **Basic information**, add the following:
 - For **Function name**, enter a name that identifies it as the function used to stop your EC2 instances. For example, "StopEC2Instances".
 - For **Runtime**, choose **Python 3.8**.
 - Under **Permissions**, expand **Choose or create an execution role**.
 - Under **Execution role**, choose **Use an existing role**.
 - Under **Existing role**, choose the IAM role that you created.
4. Choose **Create function**.
5. Under **Function code**, copy and paste the following code into the editor pane in the code editor (**lambda_function**). This code stops the EC2 instances that you identify.


Example Stop Function

```
import boto3
region = 'INSERT CORRECT REGION HERE'
instances = ['INSERT INSTANCE ID HERE']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

 For **region**, replace "us-west-1" with the AWS Region that your instances are in. For **instances**, replace the example EC2 instance IDs with the IDs of the specific instances that you want to stop and start.

6. Under **Basic settings**, set **Timeout** to 10 seconds.

 Configure the Lambda function settings as needed for your use case. For example, if you want to stop and start multiple instances, you might need a different value for **Timeout** and **Memory**.

7. Choose **Deploy**.
8. Repeat steps 1-7 to create another function. Do the following differently so that this function starts your EC2 instances:
In step 3, enter a different **Function name** than the one you used before. For example, "StartEC2Instances".
In step 5, copy and paste the following code into the editor pane in the code editor (**lambda_function**):

Example Start Function

```
import boto3
region = 'INSERT REGION HERE'
instances = ['INSERT INSTANCE ID HERE']
ec2 = boto3.client('ec2', region_name=region)


def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```



For **region** and **instances**, use the same values that you used for the code to stop your EC2 instances.

3 Test your Lambda functions

1. In the AWS Lambda console, choose **Functions**.
2. Choose one of the functions that you created.
3. Choose **Test**.
4. In the **Configure test event** dialog box, choose **Create new test event**.
5. Enter an **Event name**. Then, choose **Create**.


 You don't need to change the JSON code for the test event—the function doesn't use it.

6. Choose **Test** to run the function.
7. Repeat steps 1-6 for the other function that you created.

Tip: You can check the status of your EC2 instances before and after testing to confirm that your functions work as expected.

4 Create CloudWatch Events rules that trigger your Lambda functions

1. Open the Amazon CloudWatch console.
2. In the left navigation pane, under **Events**, choose **Rules**.
3. Choose **Create rule**.
4. Under **Event Source**, choose **Schedule**.
5. Do either of the following:
 - For **Fixed rate of**, enter an interval of time in minutes, hours, or days.
 - For **Cron expression**, enter an expression that tells Lambda when to stop your instances. For information on expression syntax, see Schedule expressions for rules.

 Cron expressions are evaluated in UTC. Make sure that you adjust the expression for your preferred time zone.

6. Under **Targets**, choose **Add target**.
7. Choose **Lambda function**.
8. For **Function**, choose the function that stops your EC2 instances.
9. Choose **Configure details**.
10. Under **Rule definition**, do the following:
 - For **Name**, enter a name to identify the rule, such as "StopEC2Instances".
(Optional) For **Description**, describe your rule. For example, "Stops EC2 instances every night at 10 PM."
 - For **State**, choose the **Enabled** check box.
11. Choose **Create rule**.
12. Repeat steps 1-11 to create a rule to start your EC2 instances. Do the following differently:
In step 5, for **Cron expression**, enter an expression that tells Lambda when to start your instances.
In step 8, for **Function**, choose the function that starts your EC2 instances.
In step 10, under **Rule definition**, enter a **Name**, such as "StartEC2Instances".
(Optional) Enter a **Description**, such as "Starts EC2 instances every morning at 7 AM."