

AWS Advance | Project 3

LAB:- Provision EC2 instance with Lambda:

Lab Details :-

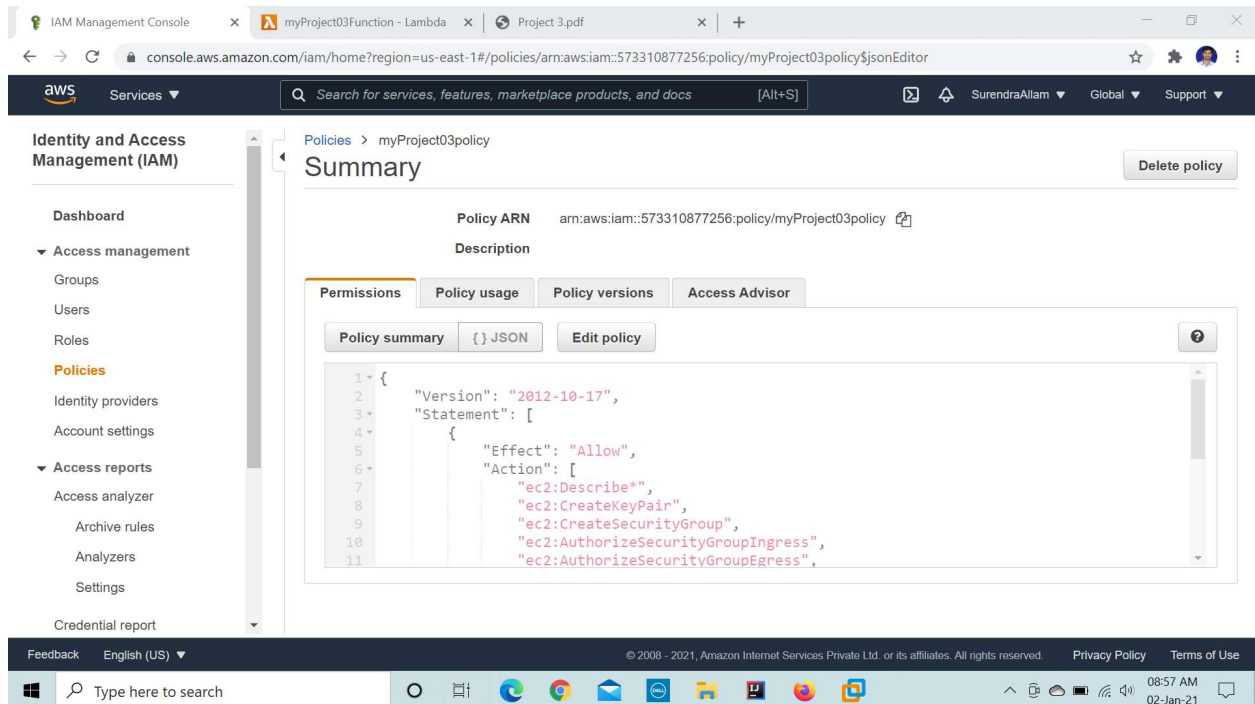
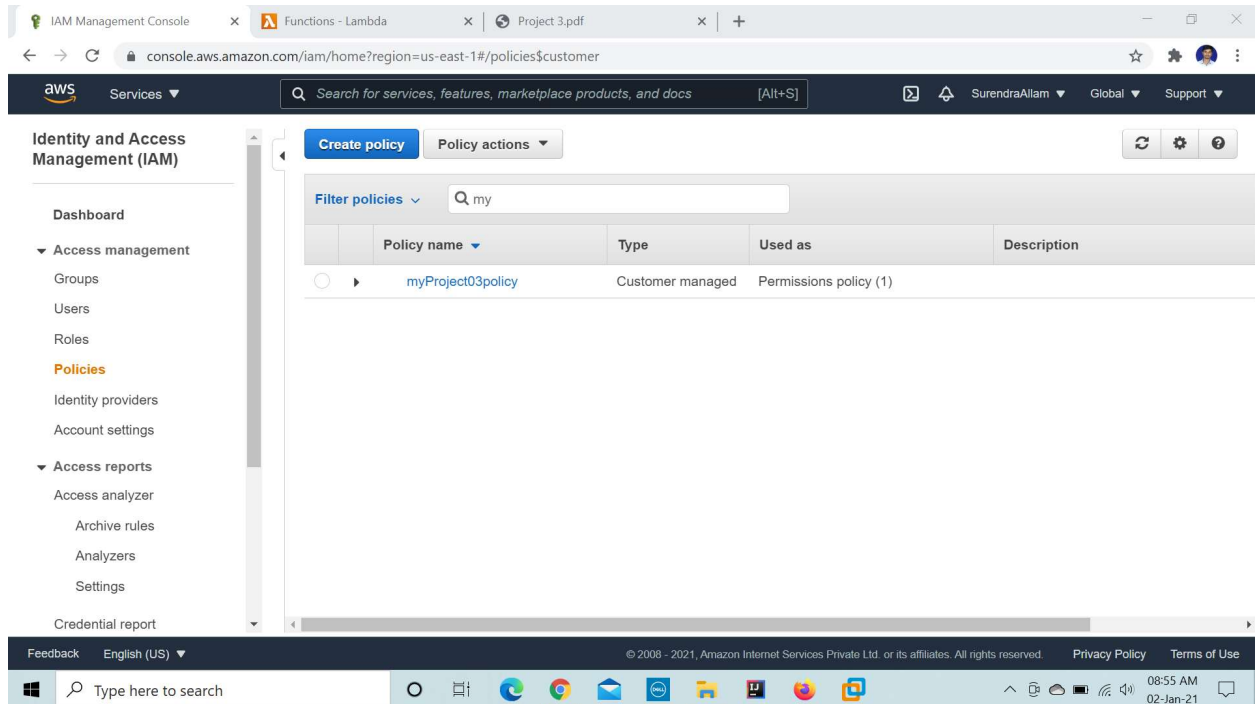
- 1. This lab walks through the steps to Launch an EC2 Instance with Lambda.**
- 2. Practice using Lambda Function.**
- 3. Duration: 1 hour**
- 4. AWS Region: US East (N. Virginia) us-east-1**

Lab Tasks :-

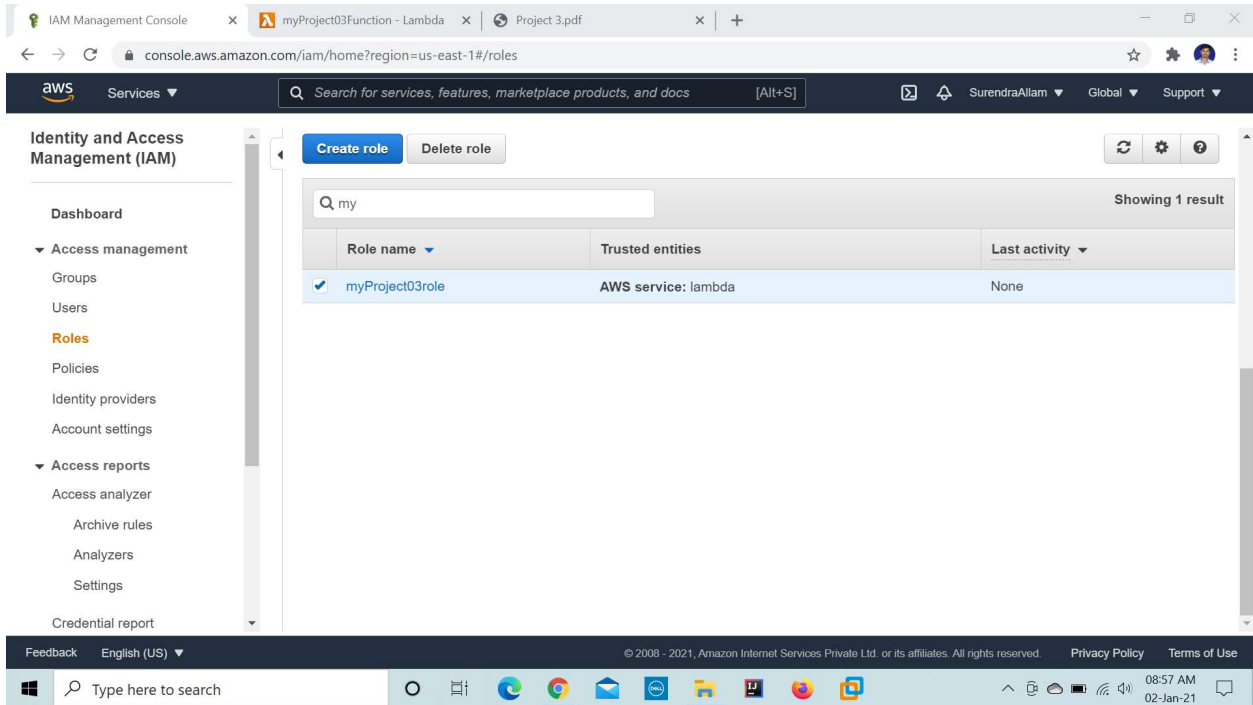
- 1. Log into the AWS Management Console.**
 - 2. Create an IAM Policy and an IAM Role.**
 - 3. Create a lambda function.**
 - 4. Configure a test event.**
 - 5. Trigger the lambda function manually using the test event.**
- Test the new EC2 instance.**

Steps:-

1. Creating an IAM Policy.

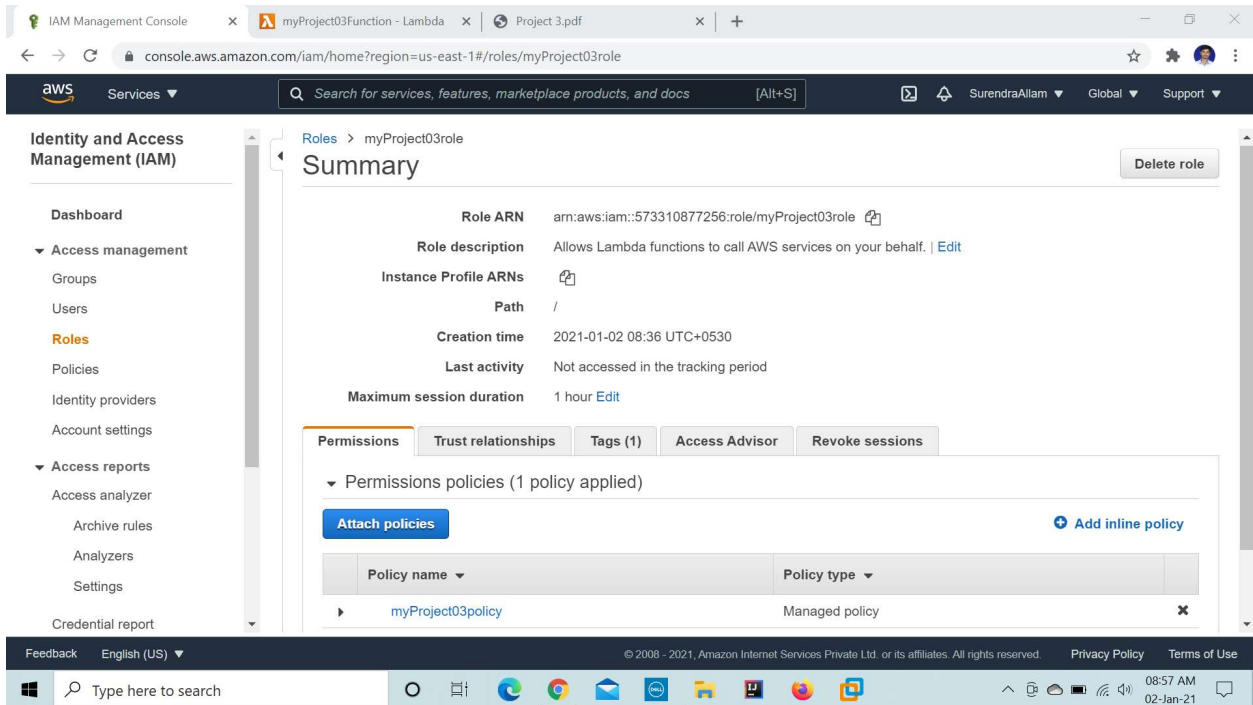


2. Create an IAM Role.



The screenshot shows the AWS IAM Management Console. The left sidebar contains the navigation menu with 'Roles' highlighted. The main content area shows a search for 'my' with 'Showing 1 result'. The table below lists the role 'myProject03role' with trusted entities 'AWS service: lambda' and no last activity.

Role name	Trusted entities	Last activity
myProject03role	AWS service: lambda	None



The screenshot shows the 'Summary' page for the 'myProject03role'. The page displays various details about the role, including its ARN, description, instance profile ARNs, path, creation time, last activity, and maximum session duration. The 'Permissions' tab is selected, showing that one policy is applied: 'myProject03policy'.

Role ARN arn:aws:iam::573310877256:role/myProject03role

Role description Allows Lambda functions to call AWS services on your behalf. [Edit](#)

Instance Profile ARNs

Path /

Creation time 2021-01-02 08:36 UTC+0530

Last activity Not accessed in the tracking period

Maximum session duration 1 hour [Edit](#)

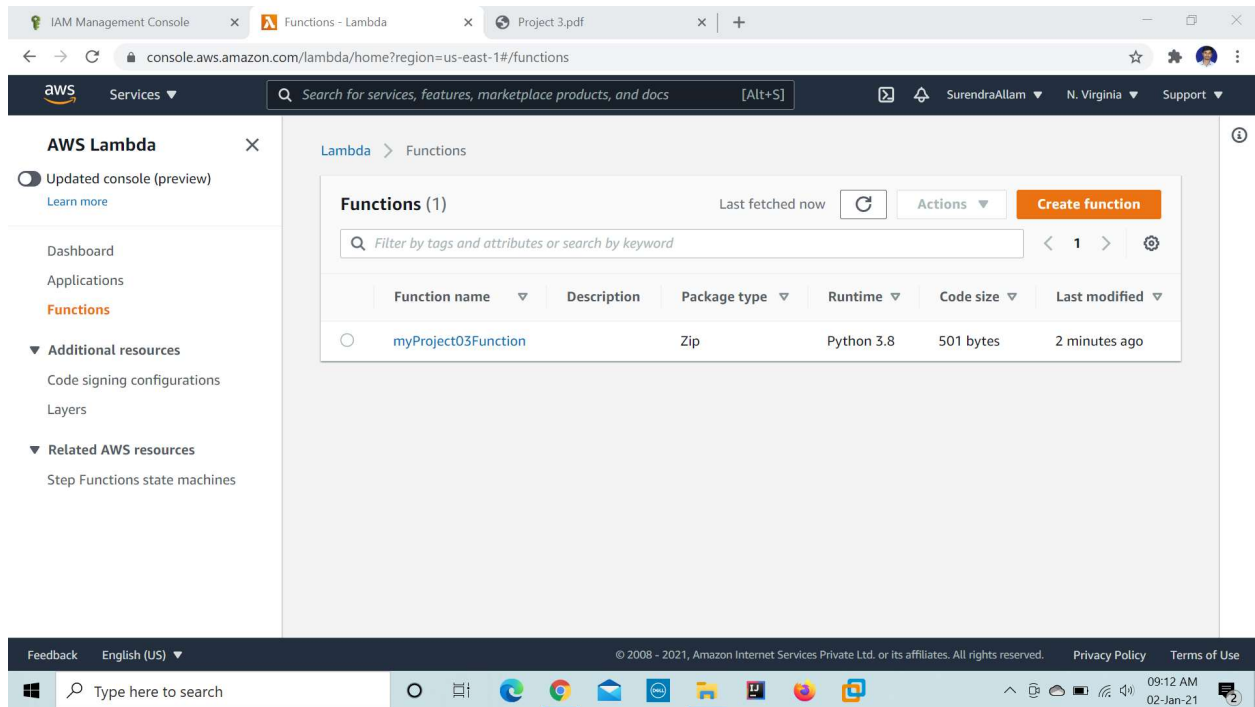
Permissions | Trust relationships | Tags (1) | Access Advisor | Revoke sessions

Permissions policies (1 policy applied)

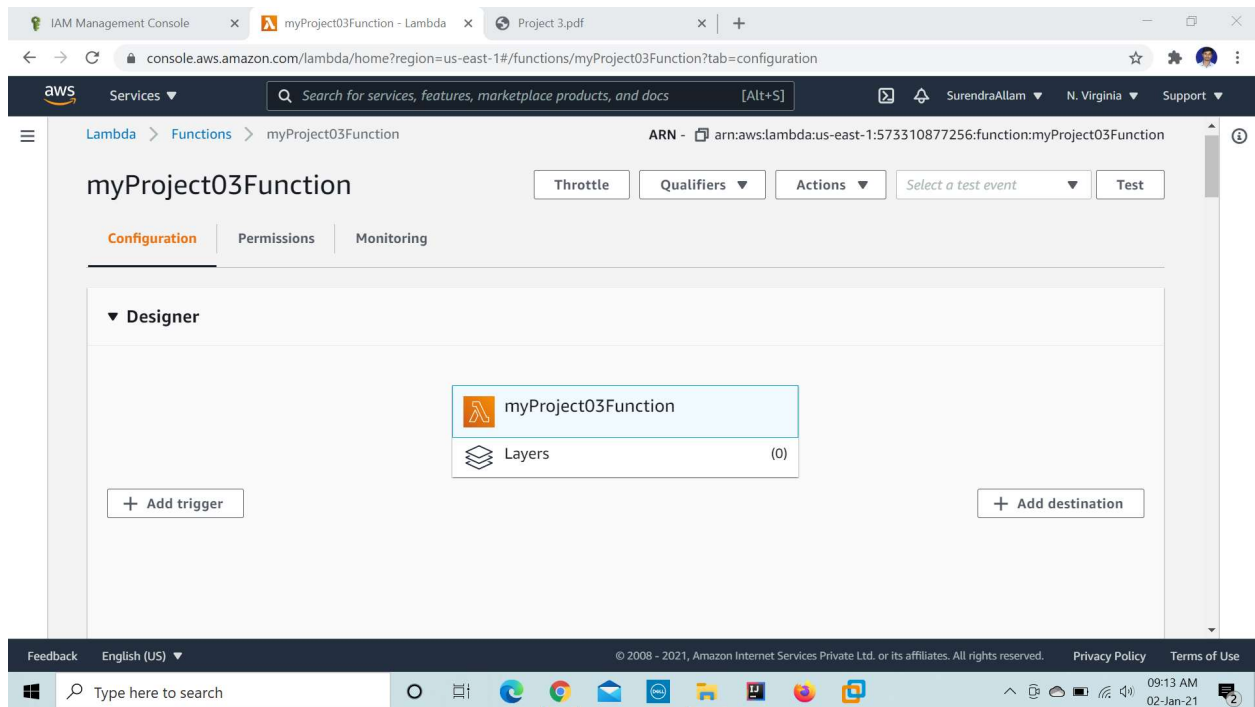
[Attach policies](#) [Add inline policy](#)

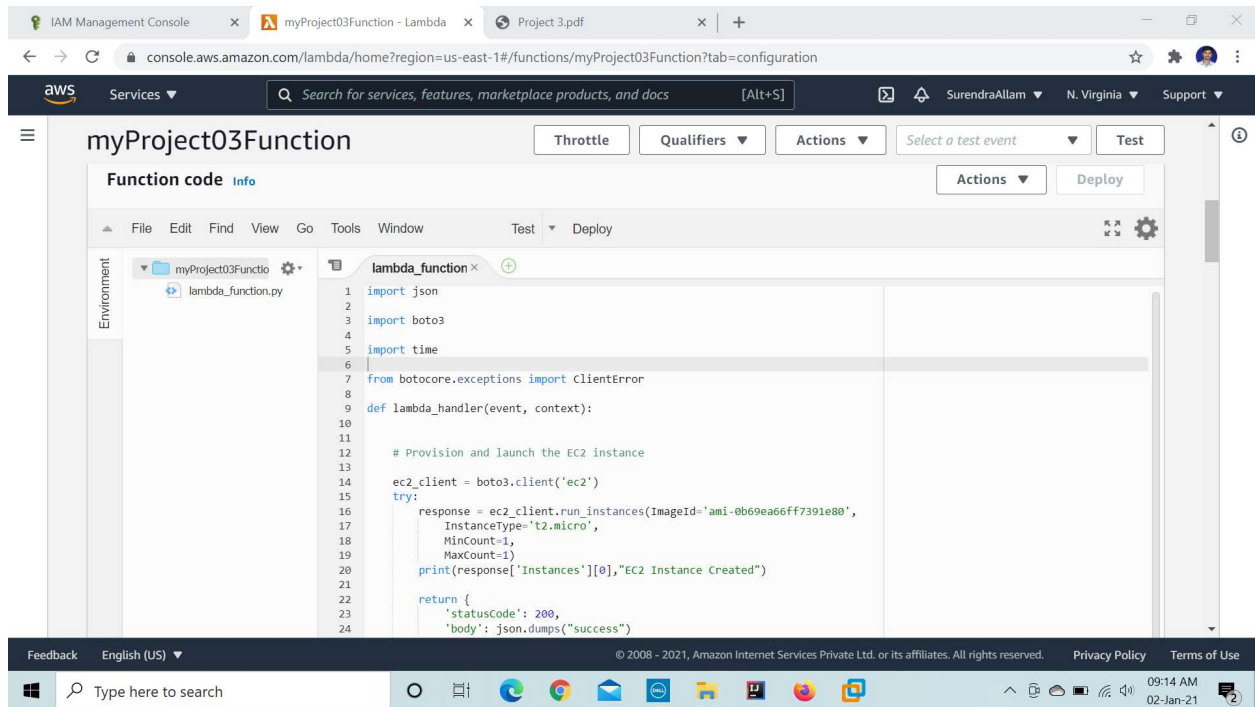
Policy name	Policy type
myProject03policy	Managed policy

3. Create a Lambda Function.

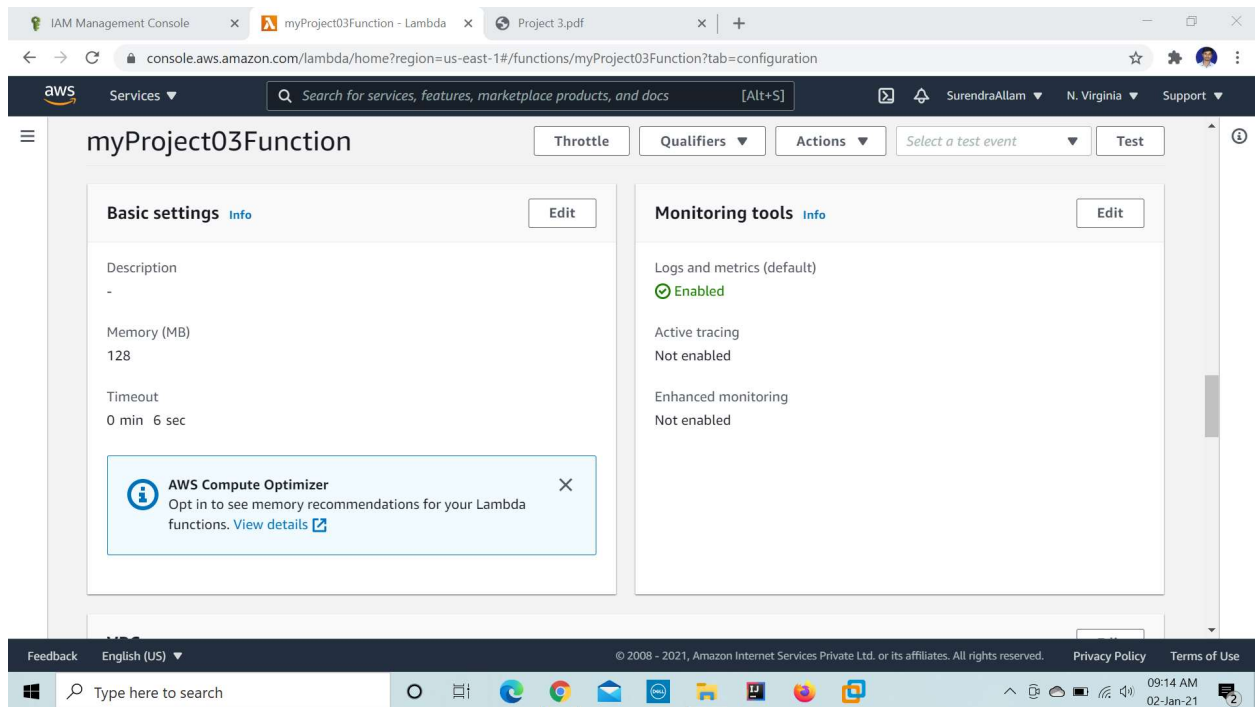


Function code Configuration:

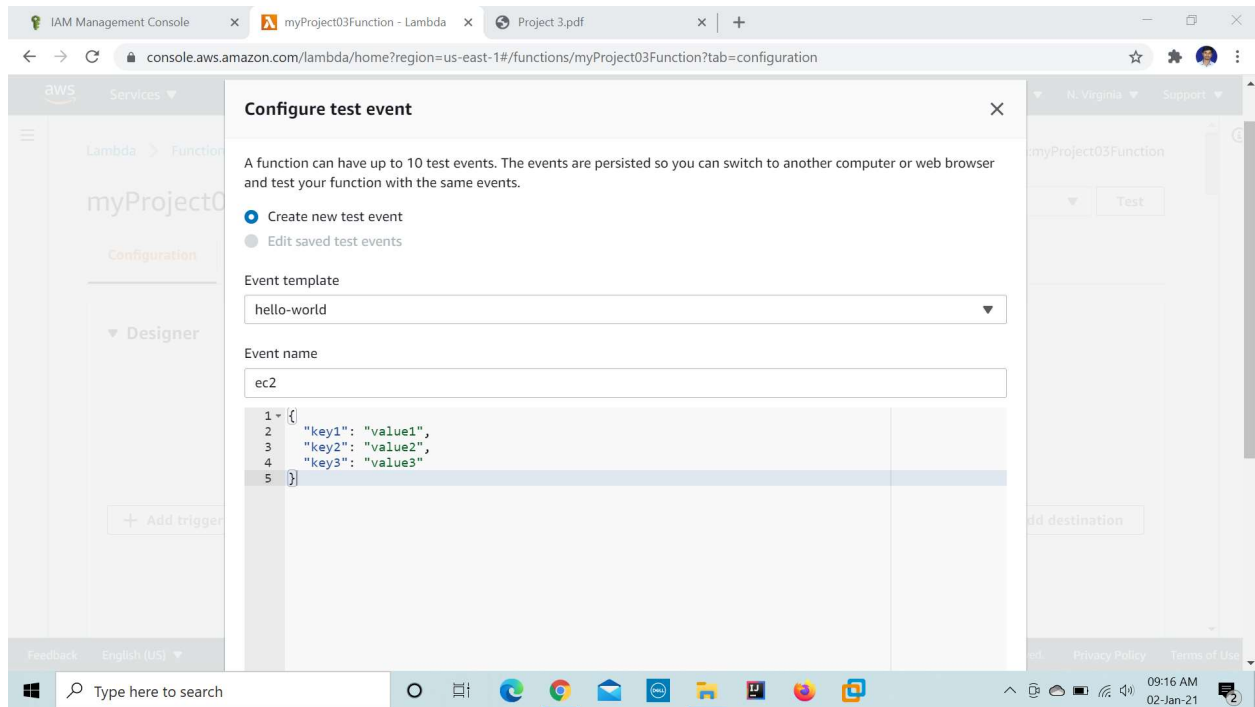




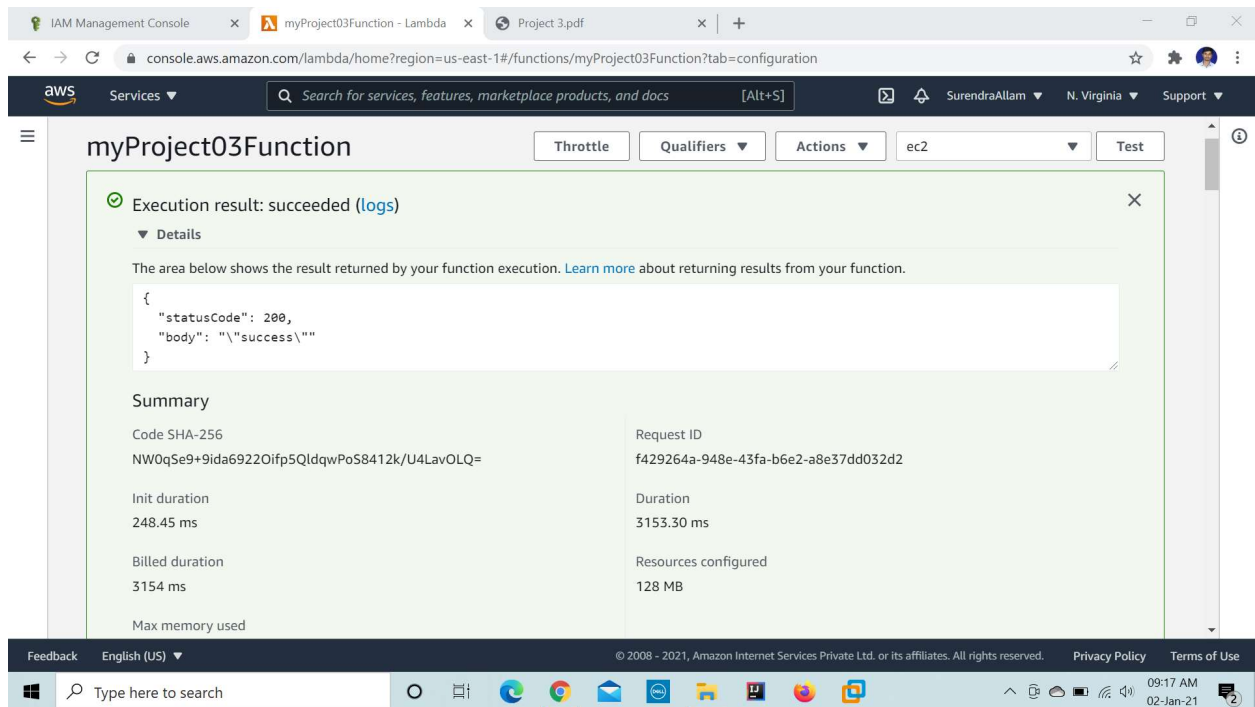
4. Set Timeout settings.



5. Configure a Test Event.



6. Trigger the lambda manually using this simple test. Test the new EC2 Instance.



Private IP of EC2 Instance: 172.31.22.69

The screenshot shows the AWS Lambda console for the function **myProject03Function**. The 'Log output' section displays a log entry from CloudWatch. The log entry is a JSON object containing various instance details, including the private IP address.

```
START RequestId: f429264a-948e-43fa-b6e2-a8e37dd032d2 Version: $LATEST
{"AmiLaunchIndex": 0, "ImageId": "ami-0b69ea66ff7391e80", "InstanceId": "i-02e5d31811d35003e", "InstanceType": "t2.micro",
"LaunchTime": "2021-01-23T03:47:08.000Z", "Monitoring": {"State": "disabled"}, "Placement": {"AvailabilityZone": "us-east-1c", "GroupName": "", "Tenancy": "default"}, "PrivateDnsName": "ip-172-31-22-69.ec2.internal",
"PrivateIpAddress": "172.31.22.69", "ProductCodes": [], "PublicDnsName": "", "State": {"Code": 0, "Name": "pending"},
"StateTransitionReason": "", "SubnetId": "subnet-a6b6f9eb", "VpcId": "vpc-d6b77cab", "Architecture": "x86_64",
"BlockDeviceMappings": [], "ClientToken": "127e4a34-ff16-4279-b24d-9391172ec8bf", "EbsOptimized": false, "EnaSupport": true,
"Hypervisor": "xen", "NetworkInterfaces": [{"Attachment": {"AttachTime": "2021-01-23T03:47:08.000Z", "AttachmentId": "eni-attach-092cd5d4a404db2bb", "DeleteOnTermination": true, "DeviceIndex": 0, "Status": "attaching", "Description": "", "Groups": [{"GroupName": "default", "GroupId": "sg-05120736"}], "Ipv6Addresses": []},
"InterfaceId": "eni-attach-092cd5d4a404db2bb"}], "Status": "pending", "Tags": []}
```

7. Check that the EC2 instance launched successfully. EC2 Dashboard

The screenshot shows the AWS EC2 console. The 'Instances (1)' list shows one instance in the 'Running' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
-	i-02e5d31811d35003e	Running	t2.micro	2/2 checks ...	No alarms	us-east-1

Instance Details:

Private IP: 172.31.22.69

Here also Same Private IP address reflecting so EC2 instance launched successfully.

The screenshot displays the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and user information (SurendraAllam, N. Virginia, Support). The left sidebar shows the navigation menu with categories like EC2 Dashboard, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, and Images. The main content area shows the 'Instance details' for the instance ID i-02e5d31811d35003e. The instance is in a 'Running' state. The summary table lists various attributes:

Instance summary for i-02e5d31811d35003e		
Instance ID	Public IPv4 address	Private IPv4 addresses
i-02e5d31811d35003e	34.227.207.146 open address	172.31.22.69
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-34-227-207-146.compute-1.amazonaws.com open address	ip-172-31-22-69.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-d6b77cab
AWS Compute Optimizer finding	IAM Role	Subnet ID
Opt-in to AWS Compute Optimizer for recommendations. Learn more	-	subnet-a6b6f9eb

The bottom of the screenshot shows the Windows taskbar with the search bar and various application icons. The system clock indicates the time is 09:21 AM on 02-Jan-21.