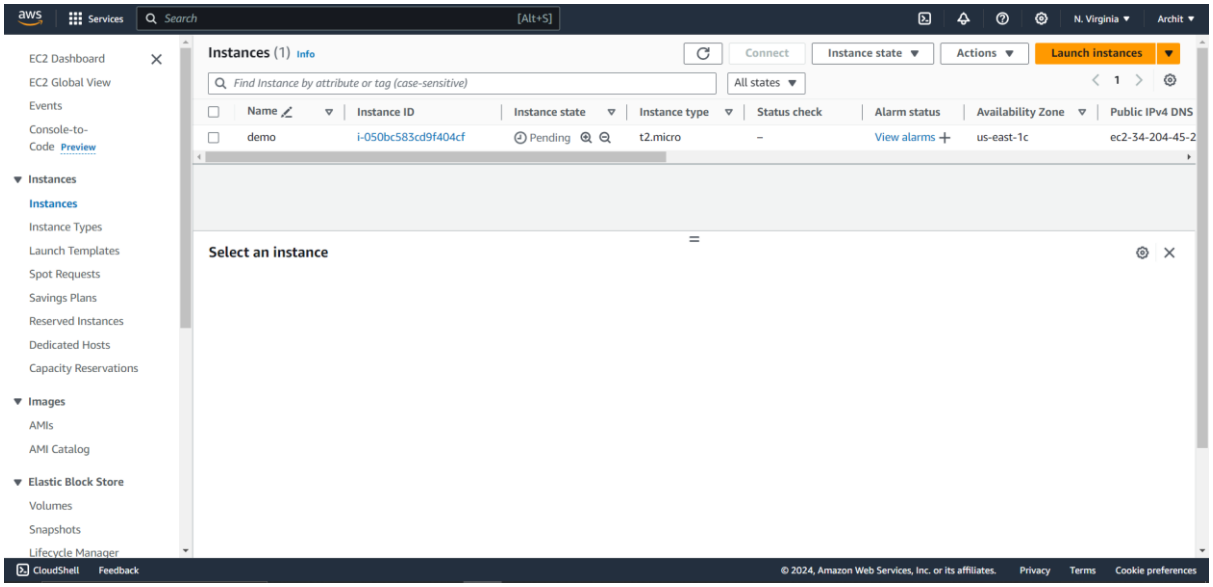
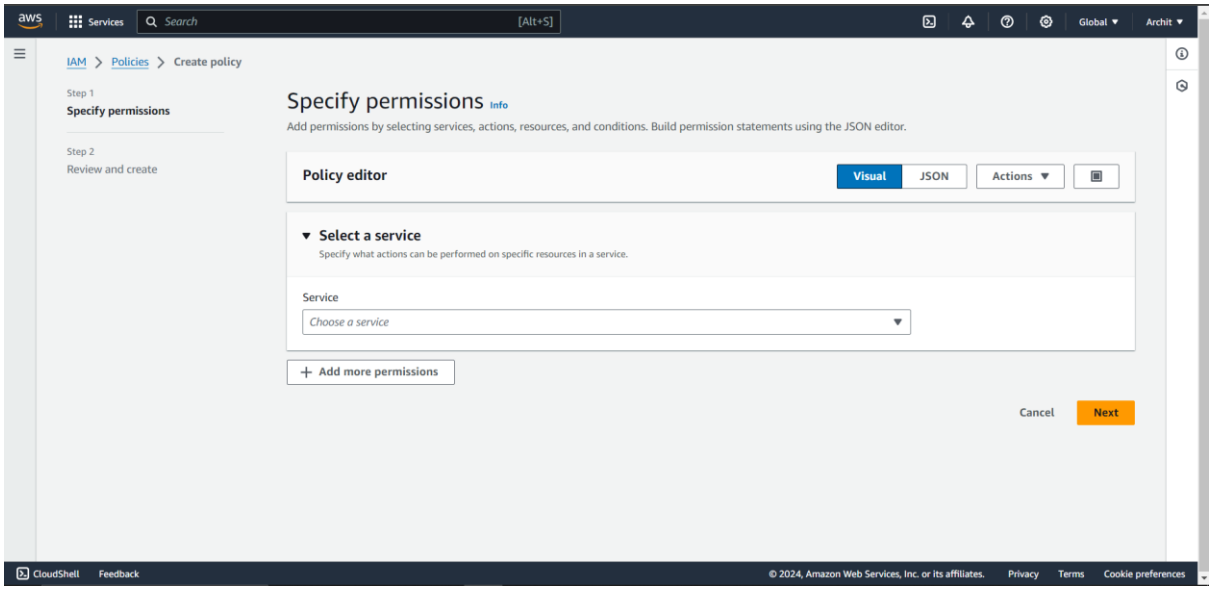
Creating schedules for stopping Instances in Lambda functions and CloudWatch

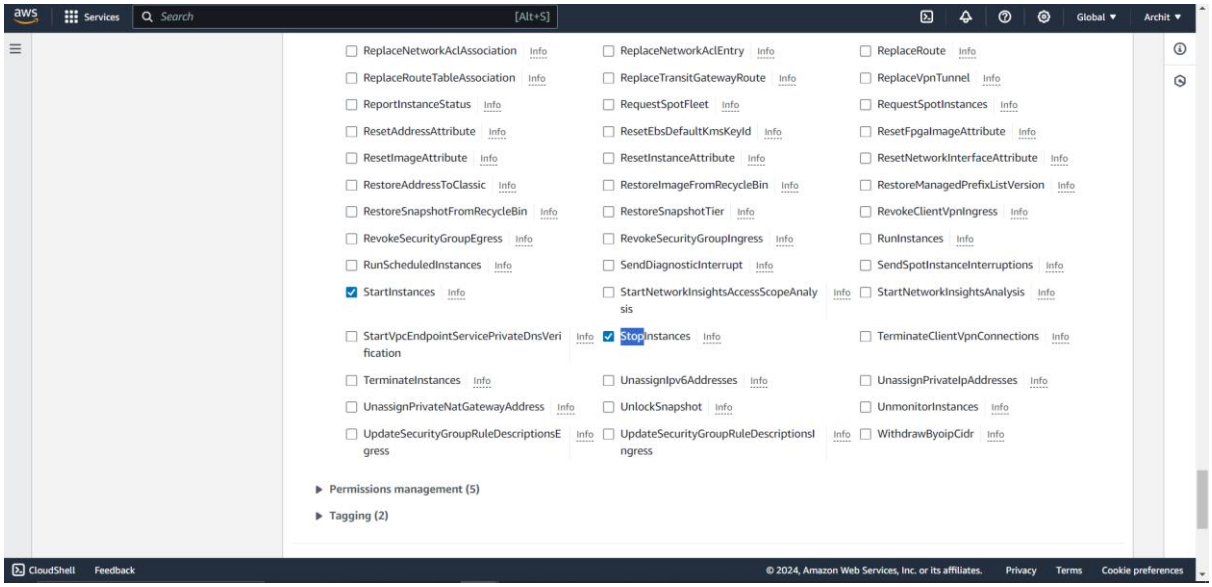
Step 1- Create & Launch an EC2 Instance

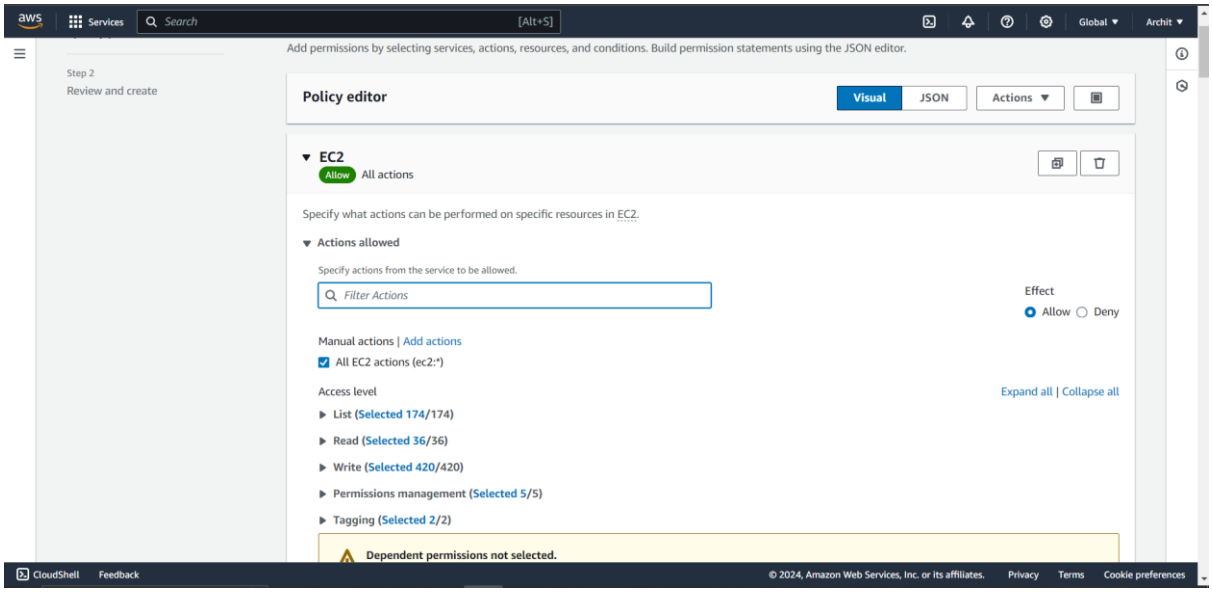


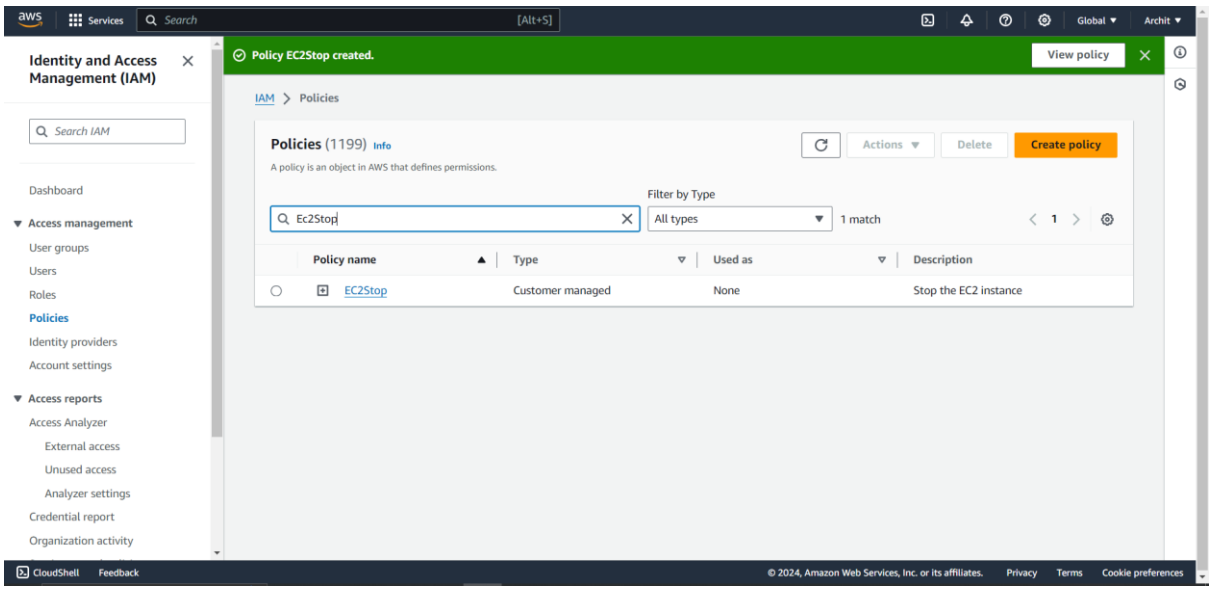
Step 2 -Go to IAM roles & Create the policies



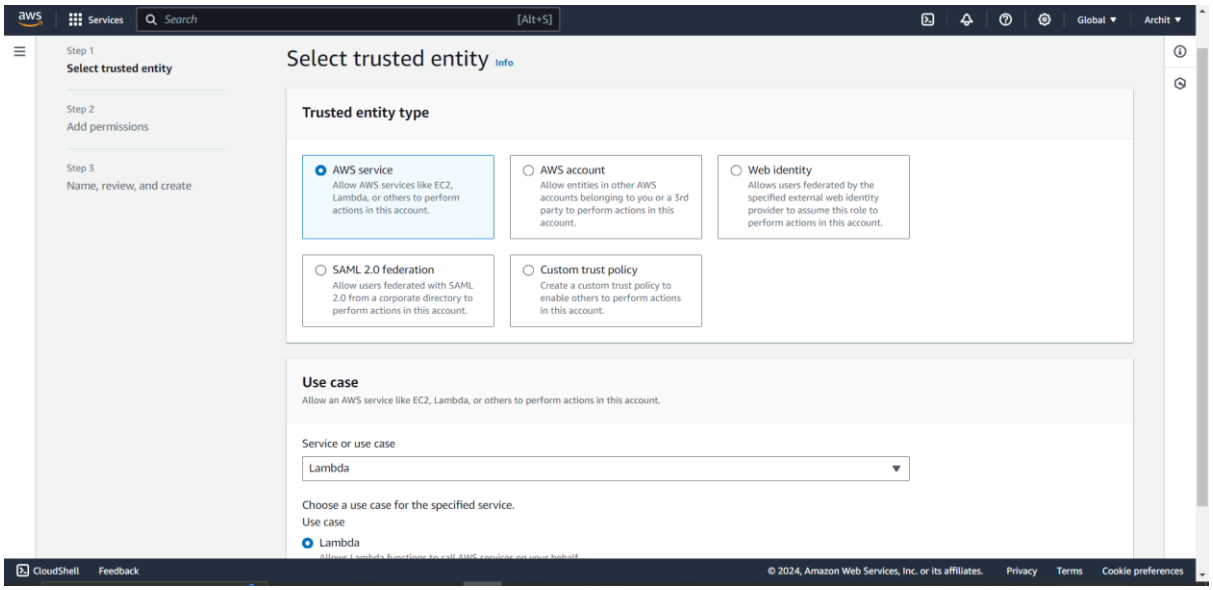
Step 3- Select ec2 @all ec2 actions

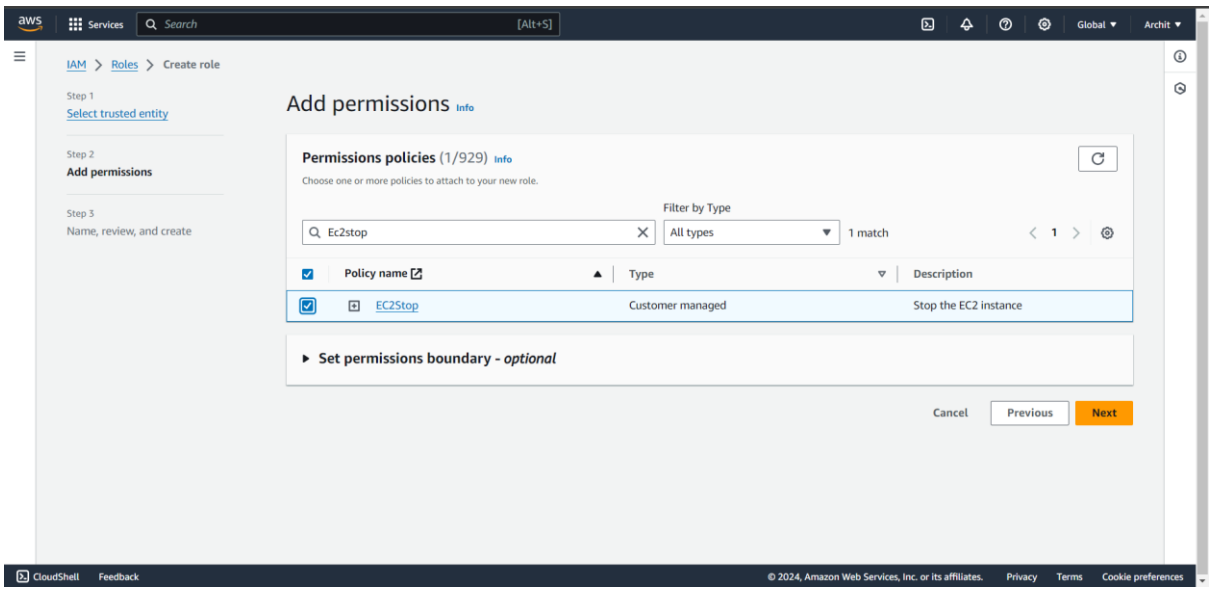

ReplaceNetworkAclAssociation
ReplaceNetworkAclEntry
ReplaceRoute
ReplaceRouteTableAssociation
ReplaceTransitGatewayRoute
ReplaceVpnTunnel
ReportInstanceStatus
RequestSpotFleet
RequestSpotInstances
ResetAddressAttribute
ResetEbsDefaultKmsKeyId
ResetFpgaImageAttribute
ResetImageAttribute
ResetInstanceAttribute
ResetNetworkInterfaceAttribute
RestoreAddressToClassic
RestoreImageFromRecycleBin
RestoreManagedPrefixListVersion
RestoreSnapshotFromRecycleBin
RestoreSnapshotTier
RevokeClientVpnIngress
RevokeSecurityGroupEgress
RevokeSecurityGroupIngress
RunInstances
RunScheduledInstances
SendDiagnosticInterrupt
SendSpotInstanceInterruptions
StartInstances
StartNetworkInsightsAccessScopeAnalysis
StartNetworkInsightsAnalysis
StartVpcEndpointServicePrivateDnsVerification
StopInstances
TerminateClientVpnConnections
TerminateInstances
UnassignIpv6Addresses
UnassignPrivateIpAddresses
UnassignPrivateNatGatewayAddress
UnlockSnapshot
UnmonitorInstances
UpdateSecurityGroupRuleDescriptionsEgress
UpdateSecurityGroupRuleDescriptionsIngress
WithdrawByoipCidr
Permissions management (5)
Tagging (2)

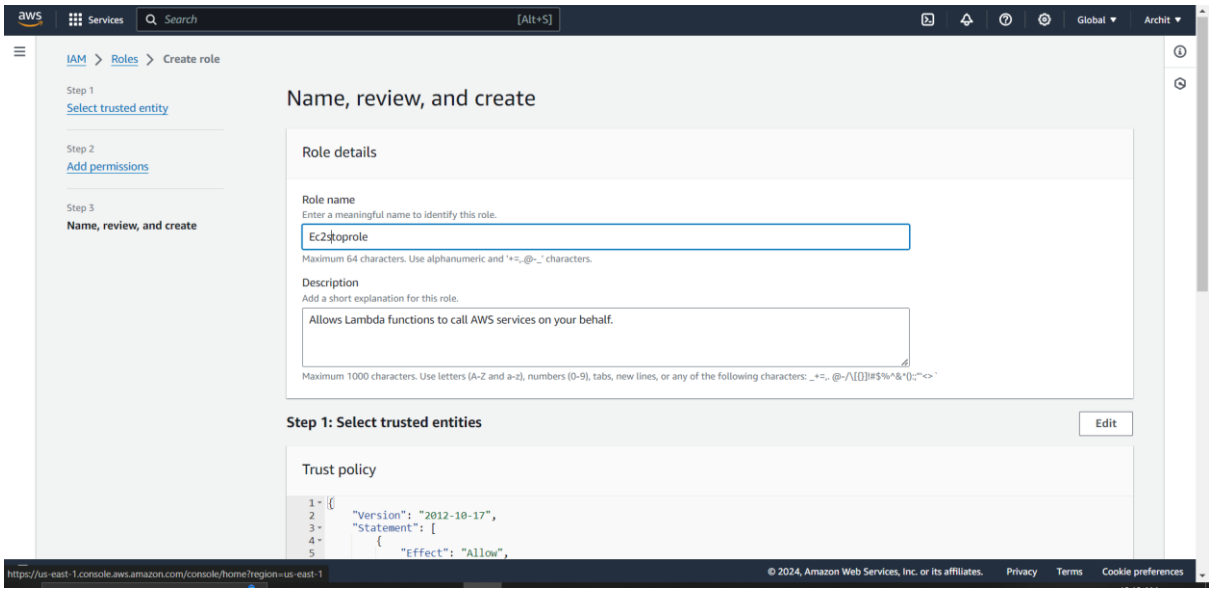

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Step 2
Review and create
Policy editor
Visual
JSON
Actions
EC2
Allow
All actions
Specify what actions can be performed on specific resources in EC2.
Actions allowed
Specify actions from the service to be allowed.
Filter Actions
Effect
Allow
Deny
Manual actions | Add actions
All EC2 actions (ec2:*)
Access level
Expand all | Collapse all
List (Selected 174/174)
Read (Selected 36/36)
Write (Selected 420/420)
Permissions management (Selected 5/5)
Tagging (Selected 2/2)
Dependent permissions not selected.


Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Policy EC2Stop created.
View policy
IAM > Policies
Policies (1199) Info
A policy is an object in AWS that defines permissions.
Actions
Delete
Create policy
Filter by Type
Ec2Stop
All types
1 match
Policy name
Type
Used as
Description
EC2Stop
Customer managed
None
Stop the EC2 instance

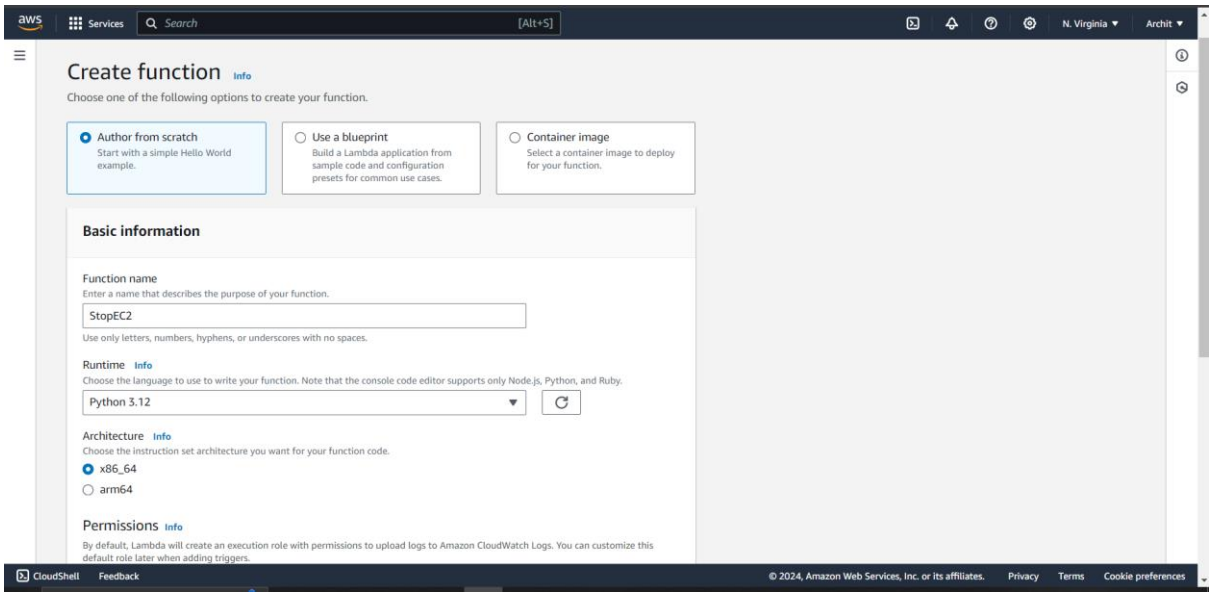
Step 4- Create Role & Attach the Policy to it

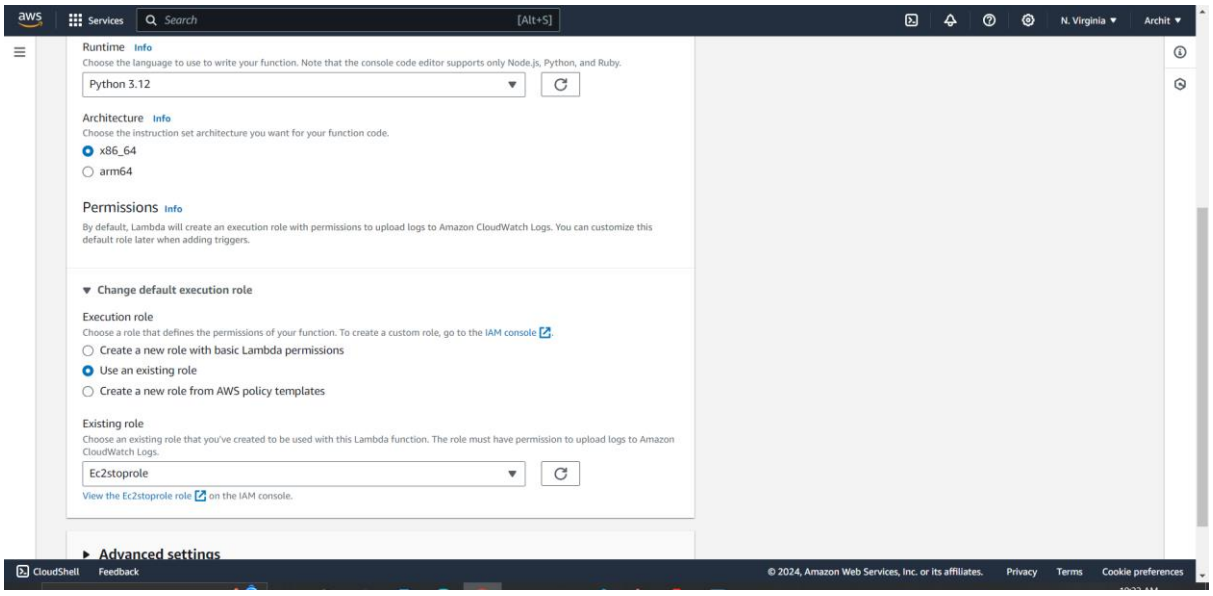






Step 5- go to lambda

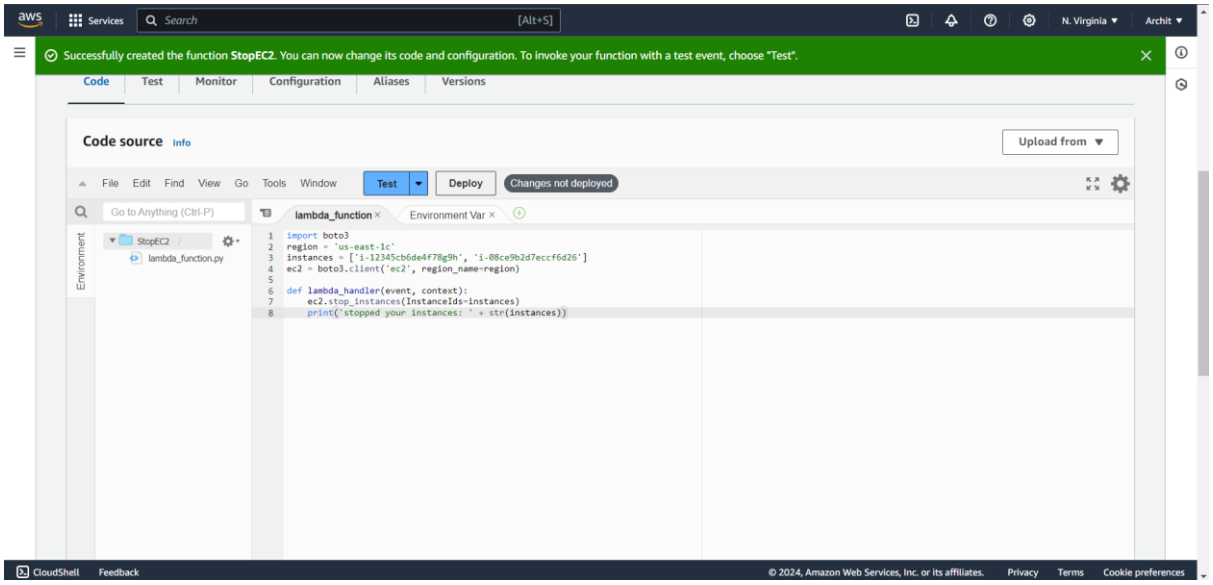




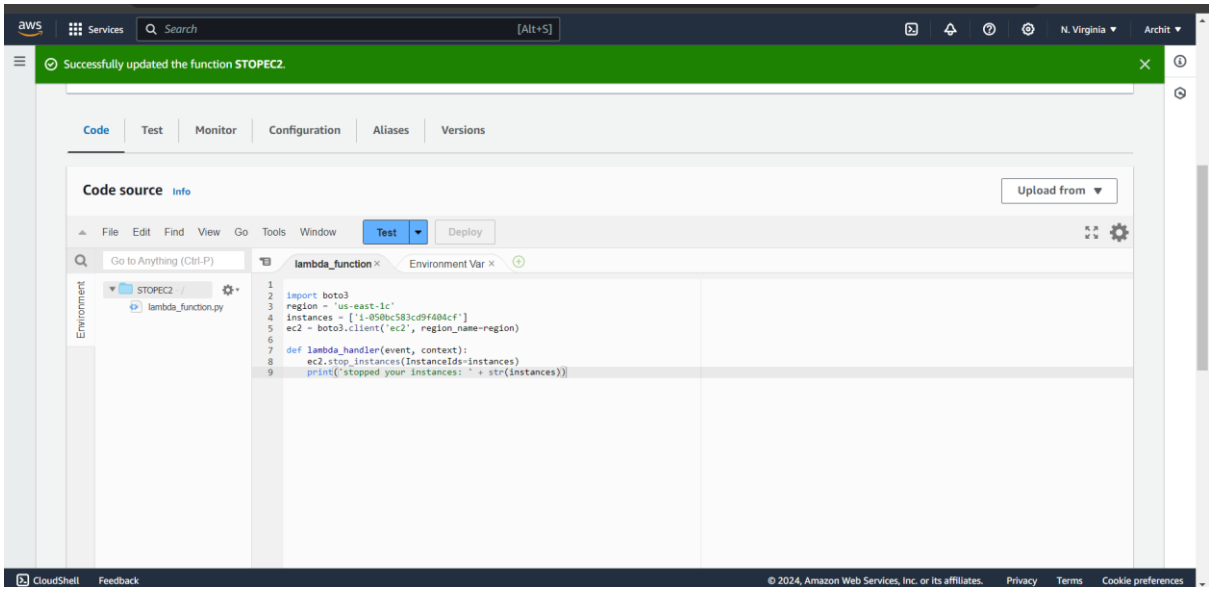
Step 5- write code **(**change region & instance id in code)

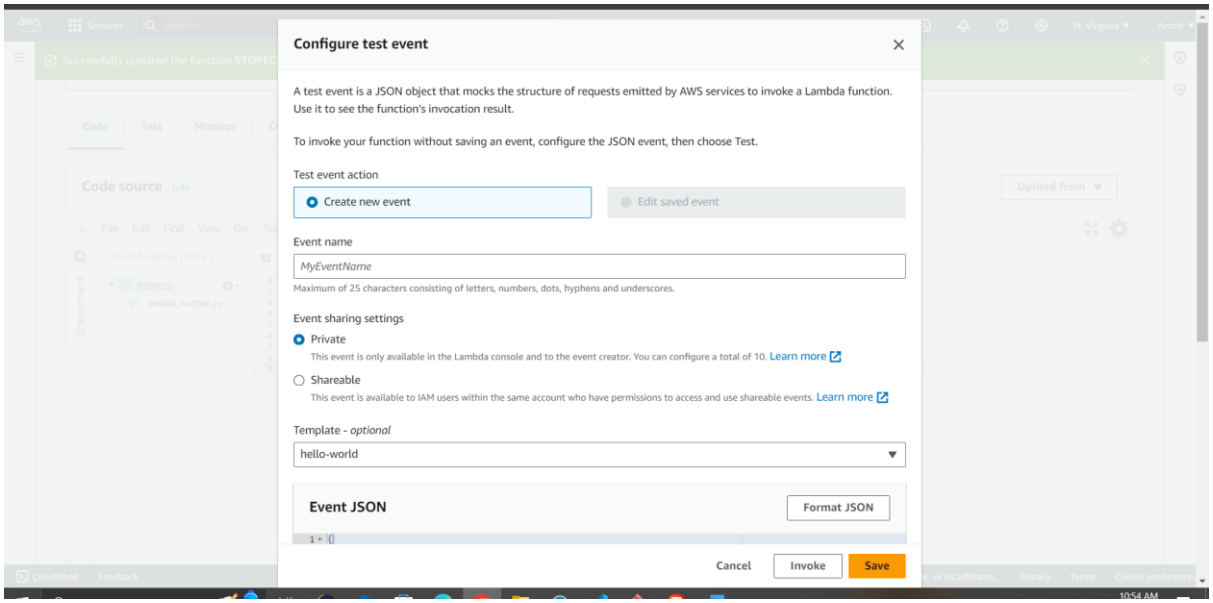
```
import boto3
region = 'us-east-1c'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

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



Step 8- Deploy & Create test event




Configure test event
A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.
To invoke your function without saving an event, configure the JSON event, then choose Test.
Test event action
Create new event
Edit saved event
Event name
MyEventName
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more
Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more
Template - optional
hello-world
Event JSON
Format JSON
Cancel
Invoke
Save