

Automatically start ec2 instance from previous state

Ec2

Under aws resources ec2 belongs to compute.

It's a regional service.

It is not managed by AWS, it means we can manage.

Ec2 purchase plans:

1. On demand
it's the default one.
2. Spot Request
according to the bidding action it means requesting an amount.
3. Reserved instance
before purchasing we can select how much cpu and instance type
in terms of specific years we can buy

In instance we have two types status checks

1. System
 2. instance
- If its a system issue , nothing but infra issue —>to overcome that we can stop and start
If its a instance issue—> checks the app health—> then reboot—>then check the logs

Instance states

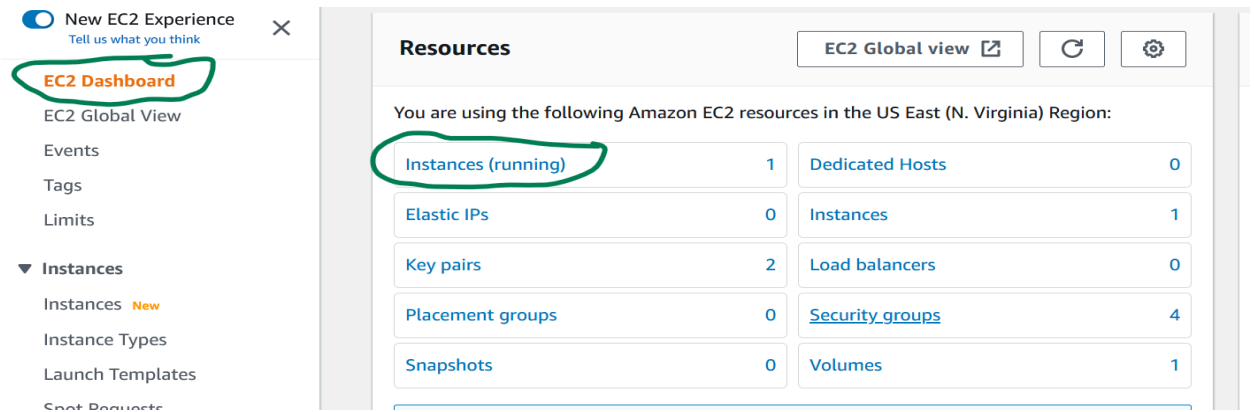
1. Terminated
2. Pending
3. Running
4. Stopped
5. initializing

The different types of ami are

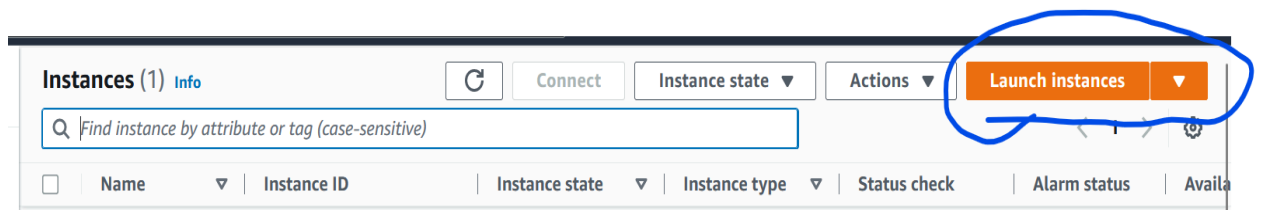
1. Amazon linux
2. Ubuntu and debian
3. Windows
4. Red hat
5. Suse linux
6. macos

First Launch ec2 in instance: =====

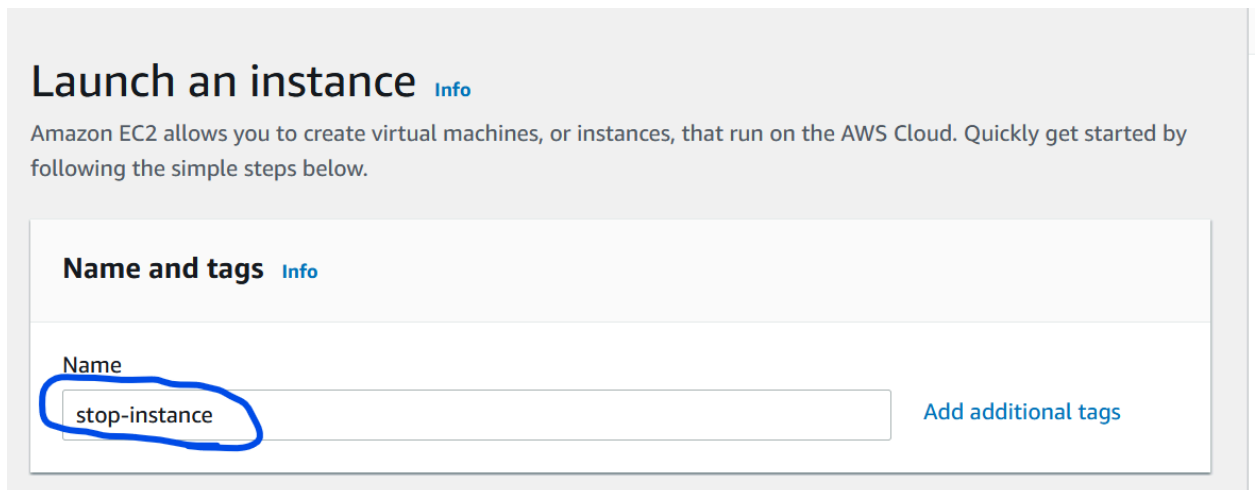
1. Go to the **ec2 dashboard**—>click on instance



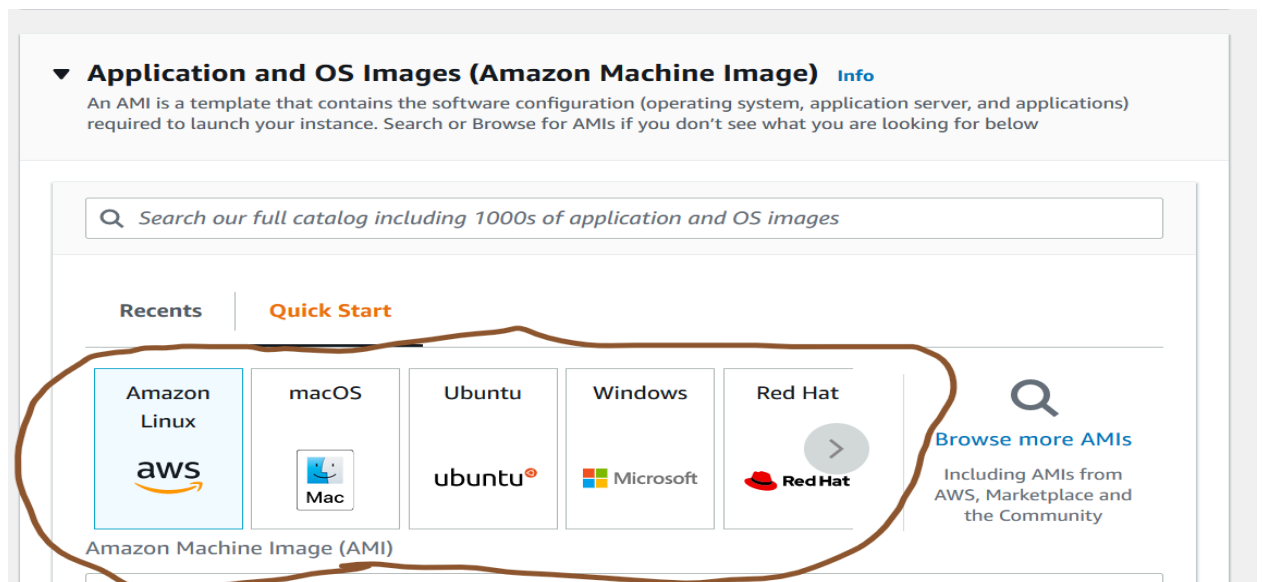
2. Next click on **launch instance**



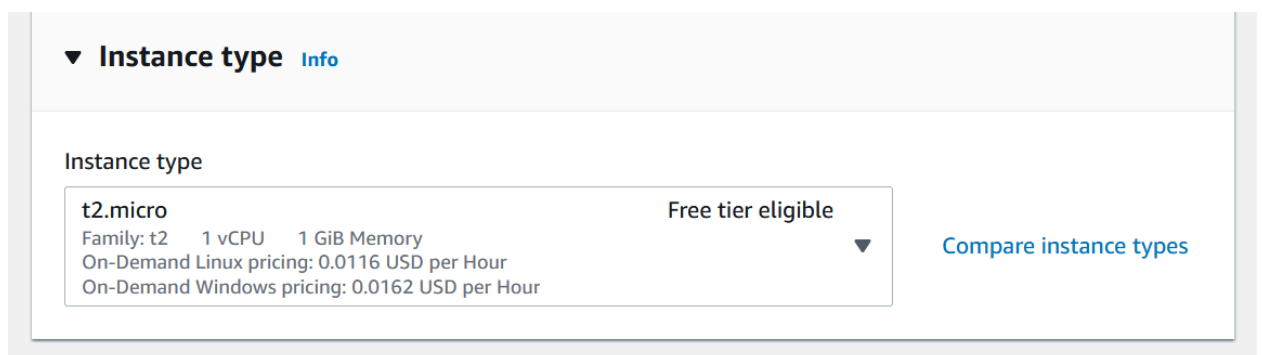
3. Give proper name of the instance



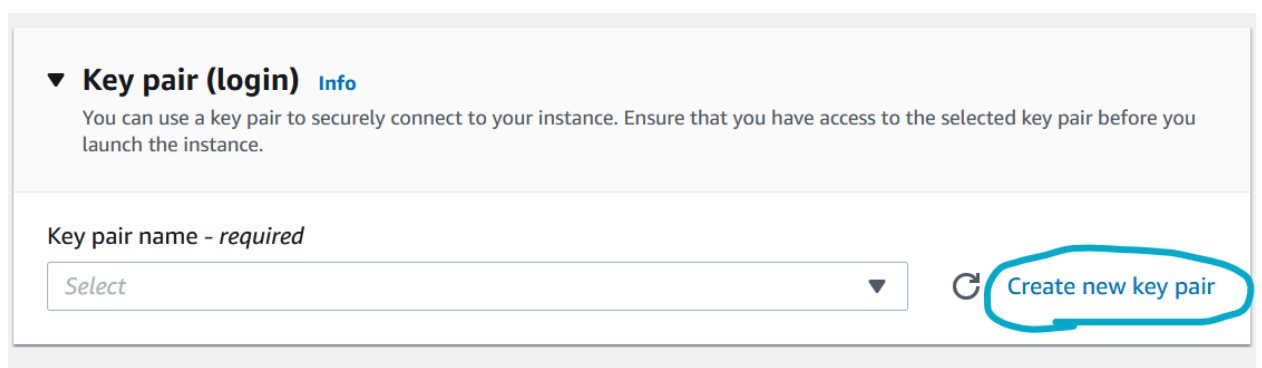
4. here we can select ami depending upon the requirement



5. after that we can select the **family type**




6. here we can select key pair, here we can create new **key pair** and select it



7. here Click on edit in network settings—> here we can select custom or default vpc—> here we can select the subnet


▼ Network settings [Info](#)


VPC - required [Info](#)

vpc-074047d18174e56f7 (default) ▼ 

172.31.0.0/16

Subnet [Info](#)

No preference ▼ 

[Create new subnet](#) 

Auto-assign public IP [Info](#)

Enable ▼

8. Click on security groups—> Depending upon the application we can pass sg rules

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

launch-wizard-4

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . _ - / () # , @ [] + = & ; {} ! \$ *

Description - required [Info](#)

launch-wizard-4 created 2022-10-22T09:09:37.647Z

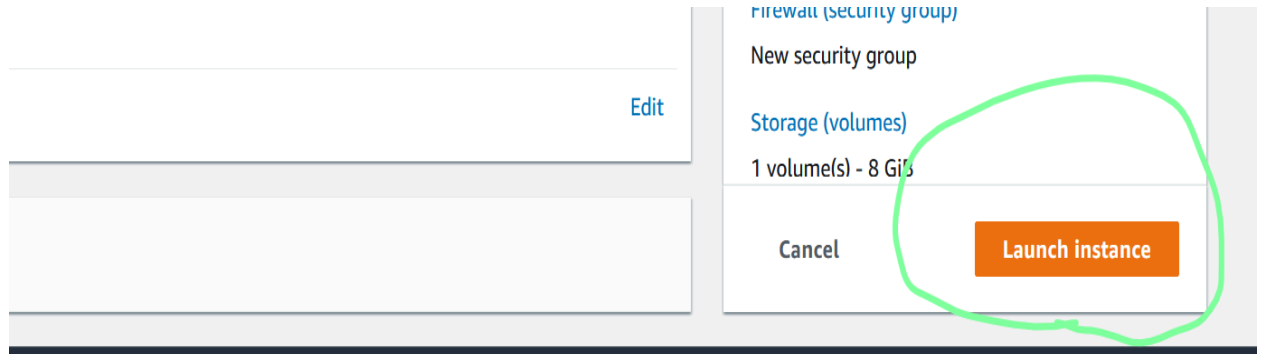
Inbound security groups rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type Info	Protocol Info	Port range Info
ssh ▼	TCP	22

9.click on launch instance ec2 will be created



Create policy

How to create policy

=====

Policy other name is permissions.
Policies are attached to Roles and Users.
File format for policy is JSON Language
Policies are three types

Managed police

This policies created and managed by AWS

Custom policy

This policies are managed by the users

Inline policy

we will assign the policy for the single user and we can't re use this policy again to another user

Policy format Syntax:

```
{
```

```
  Action:Resource/service
```

```
  Effect:Allow/deny
```

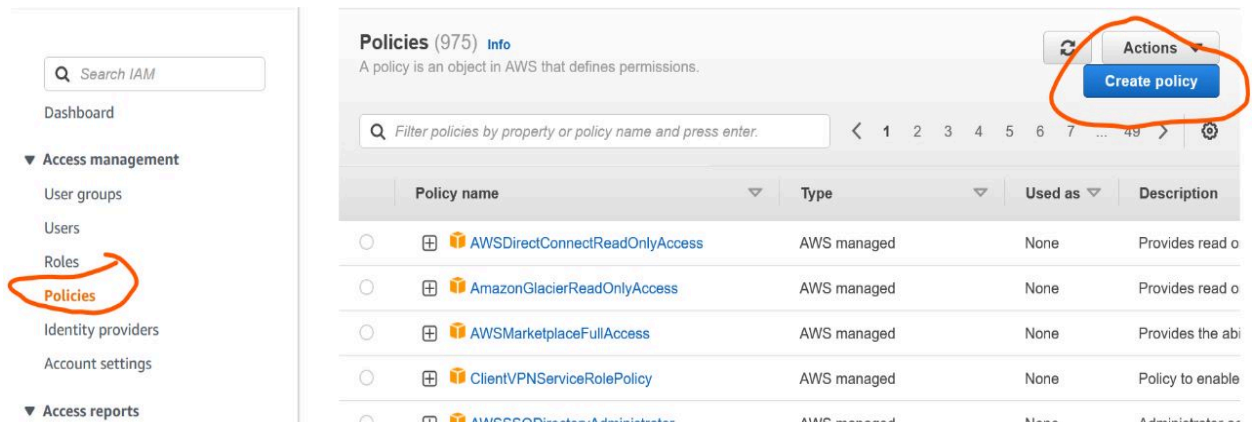
```
  Resource:
```

```
  Condition:
```

```
  Principal:
```

```
}
```

1. Click on policies—> and then click on create policy



Policies (975) Info

A policy is an object in AWS that defines permissions.

Filter policies by property or policy name and press enter.

	Policy name	Type	Used as	Description
<input type="radio"/>	AWSDirectConnectReadOnlyAccess	AWS managed	None	Provides read o
<input type="radio"/>	AmazonGlacierReadOnlyAccess	AWS managed	None	Provides read o
<input type="radio"/>	AWSMarketplaceFullAccess	AWS managed	None	Provides the abi
<input type="radio"/>	ClientVPNServiceRolePolicy	AWS managed	None	Policy to enable
<input type="radio"/>	AWSSSOAdministrator	AWS managed	None	Administrator ar

2. here we can select service (here select service type EC2), action (StopInstances, start Instances), resource and conditions

Expand all | Collapse all

Select a service

Service

Choose a service

Actions

Choose a service before defining actions

Resources

Choose actions before applying resources

Request conditions

Choose actions before specifying conditions

[Clone](#) [Remove](#)

[Add additional permissions](#)

3. Here we can give a policy name and tags for that policy—> then click on create policy.

Review policy

Name*

Use alphanumeric and '+,=, @, _' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+,=, @, _' characters.

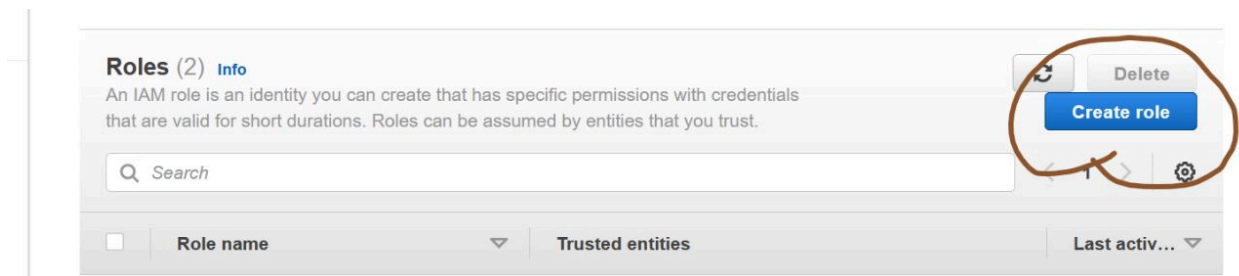
Next create role

How to create role

=====

The purpose of the role is used to connect from one service to another service

1. Click on roles---->next click on create role



2. Select the trusted entity type

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

3. Select the use case for example in my case IAM using Lambda

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.

☐ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

4. Here we can add permissions/policies for that role here i am created policy already so i am chose custom policy

3

Add permissions [Info](#)

Permissions policies (768) [Info](#)

Choose one or more policies to attach to your new role.

Q Filter policies by property or policy name and press enter. < 1 2 3 4 5 6 7 ... 39 > ⚙

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	+ AWSDirectConnect...	AWS m...	Provides read only access to AWS Direct Connect via the AWS...
<input type="checkbox"/>	+ AmazonGlacierRea...	AWS m...	Provides read only access to Amazon Glacier via the AWS Ma...
<input type="checkbox"/>	+ AWSMarketplaceFu...	AWS m...	Provides the ability to subscribe and unsubscribe to AWS Mark...
<input type="checkbox"/>	+ AWSSSODirectoryA...	AWS m...	Administrator access for SSO Directory
<input type="checkbox"/>	+ AWSIoT1ClickRead...	AWS m...	Provides read only access to AWS IoT 1-Click.

5. After that we can give Role name and tags for that Role

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+=,._@-' characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+=,._@-' characters.

After that create lambda function

How to create lambda function:

=====

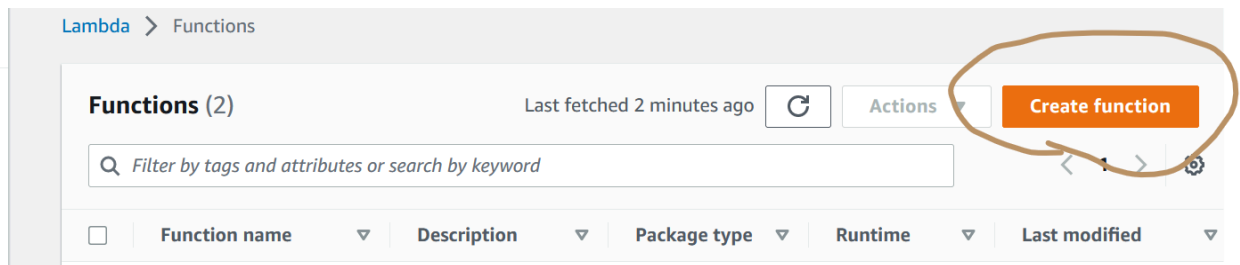
Lambda is a managed service

Lambda is a serverless service

Lambda is created with or without vpc

Lambda is used to bring up the light weight machine which will only run for a minimum of less time and goes to the excited state

1.first go to lambda service—— and then click on create function.



2. Here we can give proper name

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

3.here we can chose the language

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture [Info](#)

4. here i am passing role the role information is passed on above

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

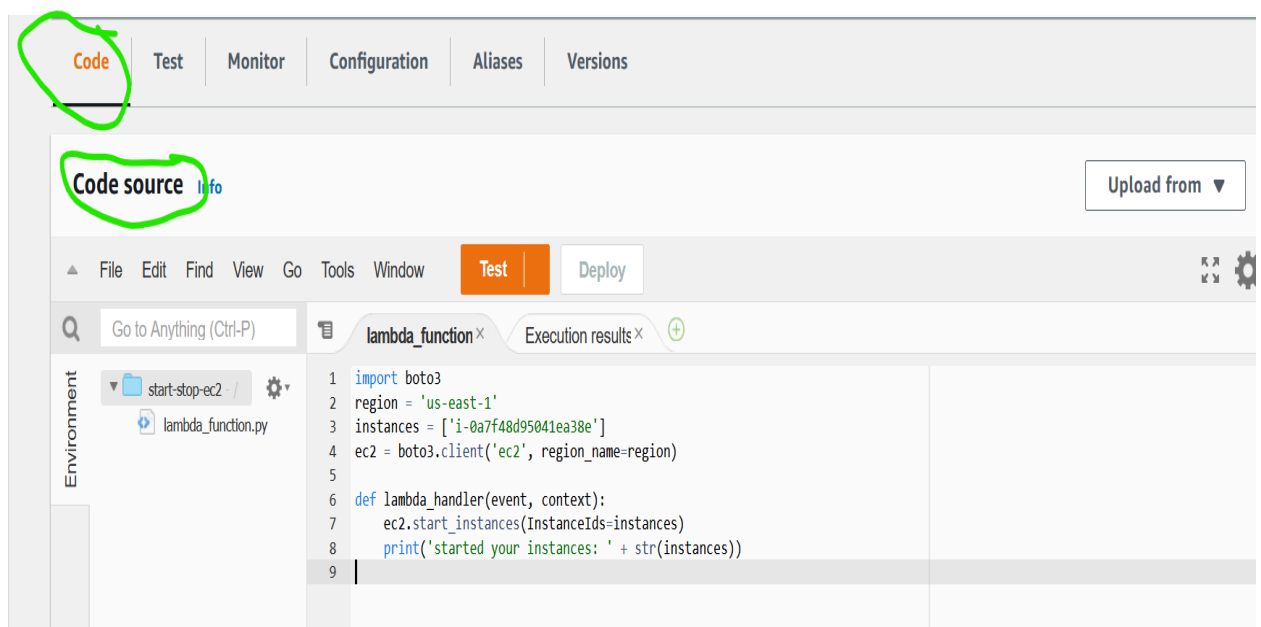
☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

6. Under **Code**, **Code source**, copy and paste the following code. This code stops the EC2 instances



Example function code—stopping EC2 instances

```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h']
ec2 = boto3.client('ec2', region_name=region)
def lambda_handler(event, context):

    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

Repeat above steps to create another function.

enter a different Function name you used before For example, "StartEC2Instances".
copy and paste the following code.

Example function code—starting EC2 instances

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

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

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

Test your Lambda functions

=====

In the **Lambda Function**, choose **Functions**.
Choose one of the functions that you created.

Select the **Code** tab.

In the **Code source** section, select **Test**.

In the **Configure test event** dialog box, choose **Create new test event**.

Enter an **Event name**. Then, choose **Create**.

Choose **Test** to run the function.

Repeat steps for the other function that you created.

Note: after you can check the status of ec2 instances

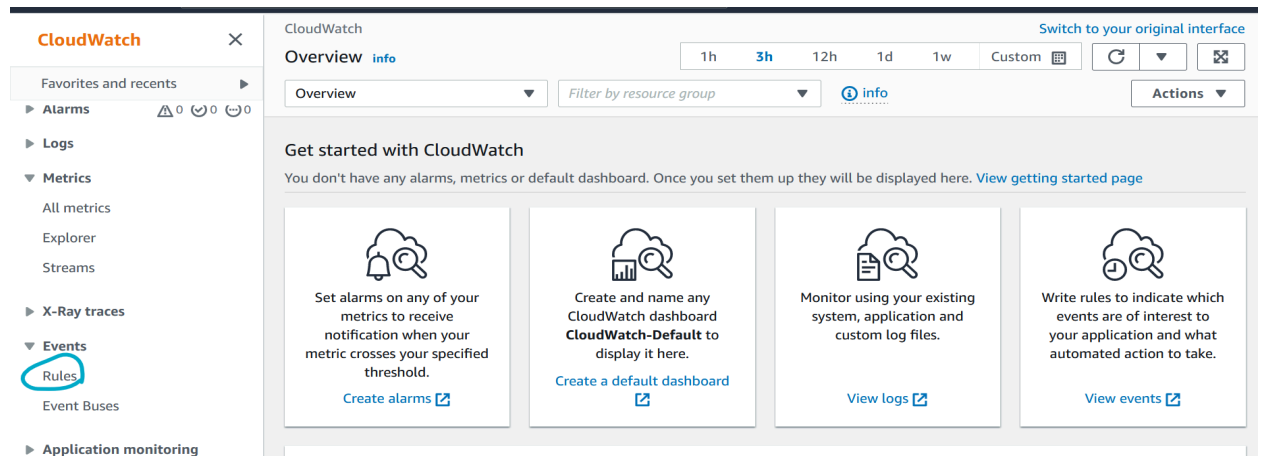
How to set time in cloud watch

=====

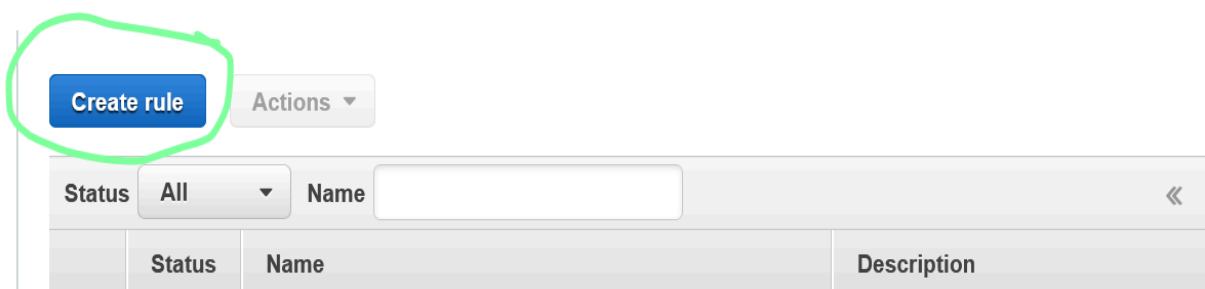
It is a monitoring and observability service

To monitor the metrics like cpu,disc

1.first go to **cloud watch** service—and go to **events**——and select **rules**



2. Click on **create rule**



3. here we select schedule

Create rules to invoke targets based on events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern ⓘ ☒ Schedule ⓘ

☒ Fixed rate of

☐ Cron expression

[Learn more](#) about CloudWatch Events schedules.

► Show sample event(s)

1. **Fixed rate:** enter an interval of time in minutes, hours, days.

Cron expression: enter a time that tells Lambda when to stop your instances.

Here cron expression are in UTC so to change your preferred time zone

\

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern ⓘ ☒ Schedule ⓘ

☐ Fixed rate of

☒ Cron expression

[Learn more](#) about CloudWatch Events schedules.

5. here we select the targets and then click configure details in my case i am chose lambda

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function*

start-stop-ec2

▸ Configure version/alias

▸ Configure input

+ Add target*

Cancel

Configure details

6. Here give proper name and click create role, role will be created

Step 2: Configure rule details

Rule definition

Name*

Description

State

☒ Enabled

7. Repeat above steps to create a rule to start your EC2 instances.

In non-business hours we don't want to run applications that we make automatically to stop and start using lambda function. By using this we can optimize the cost.