Automatically start ec2 instance [illegible]

==========

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

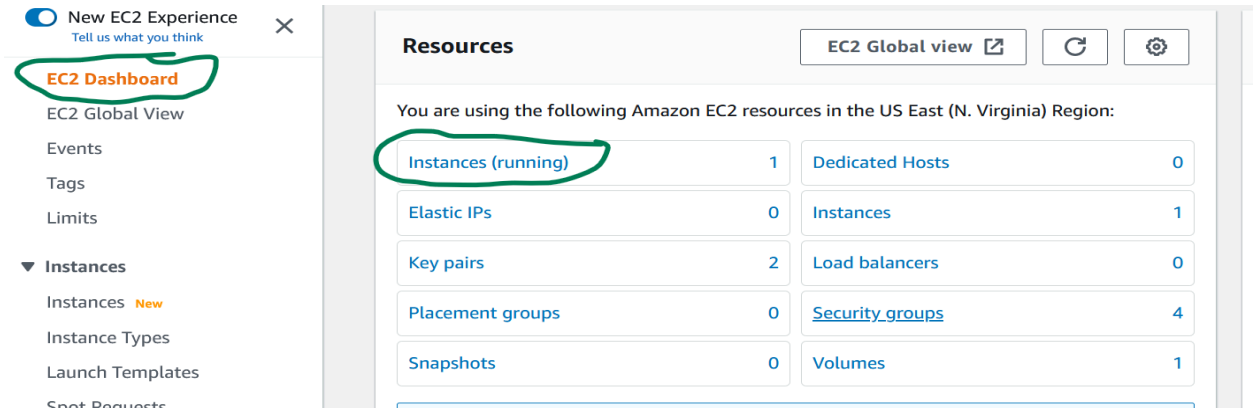
The different types of ami are

==========

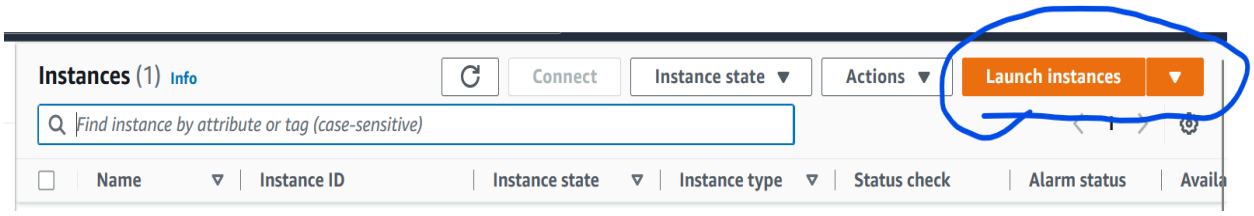
1. Amazon linux
2. Ubuntu and debian
3. Windows
4. Red hat
5. Suse linux
6. macos

First Launch ec2 in instance:
==========

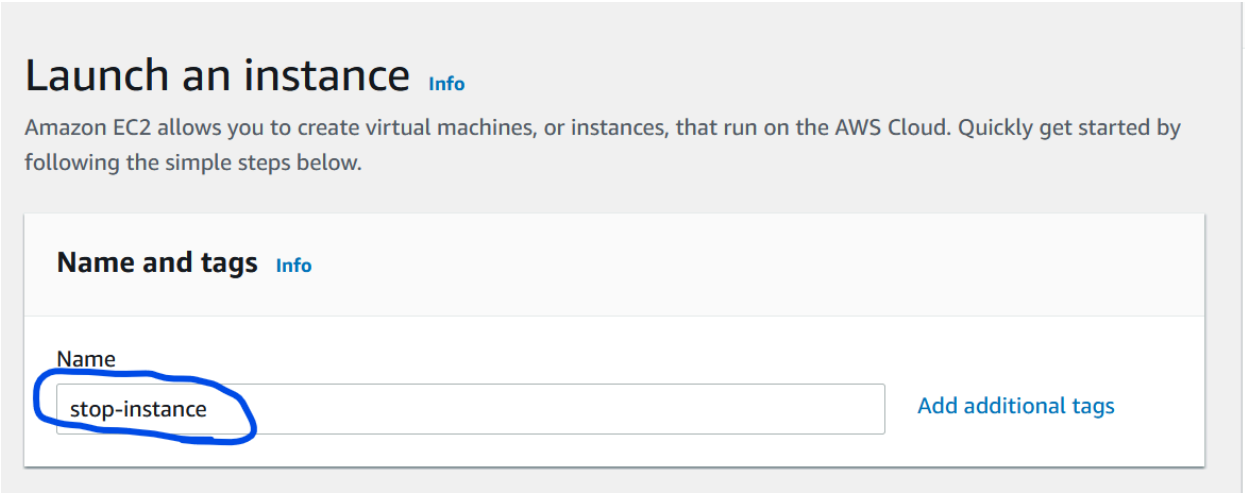
1.Go to the **ec2 dashboard**—->click on instance



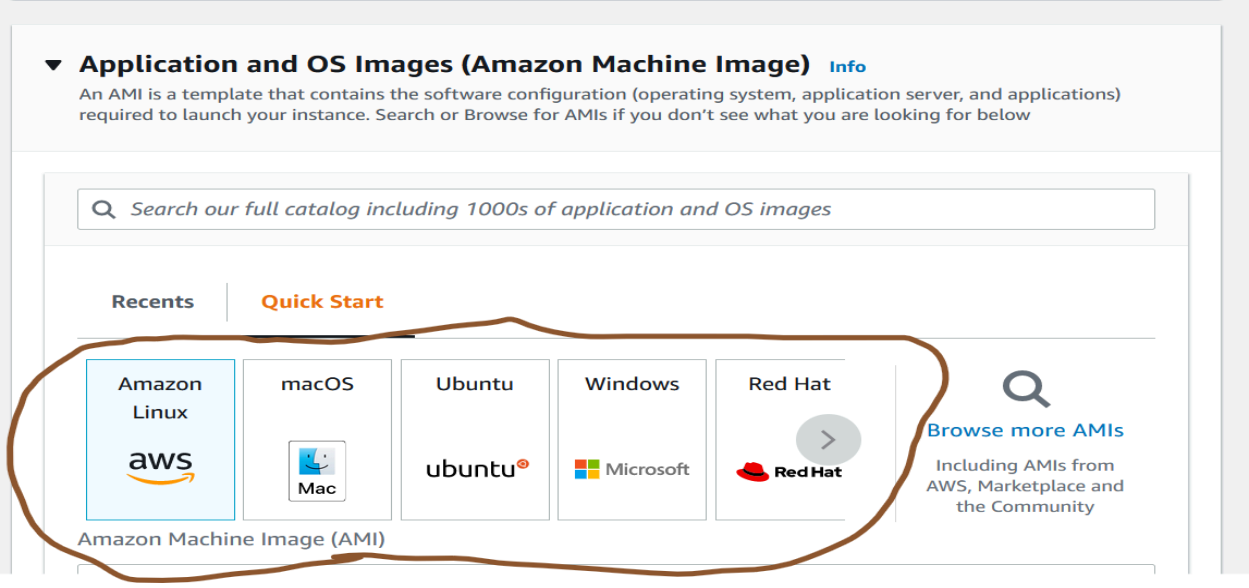
2. Next click on **launch instance**



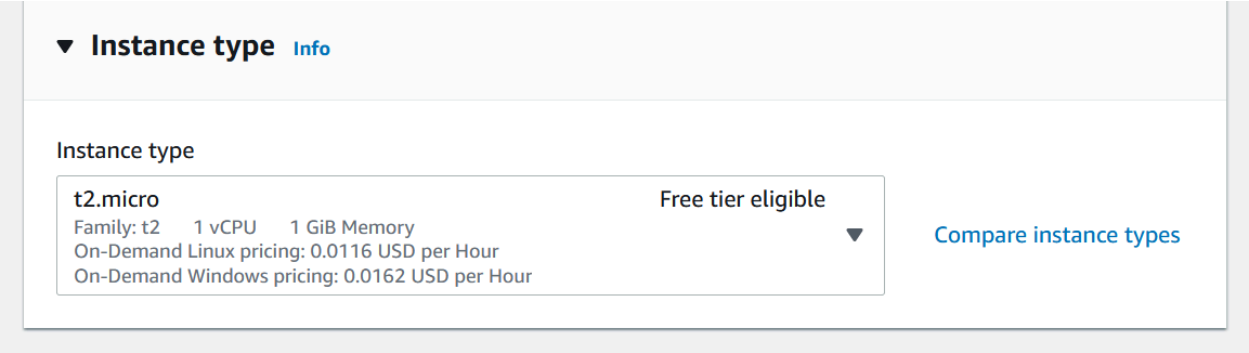
3. Give proper name of the instance



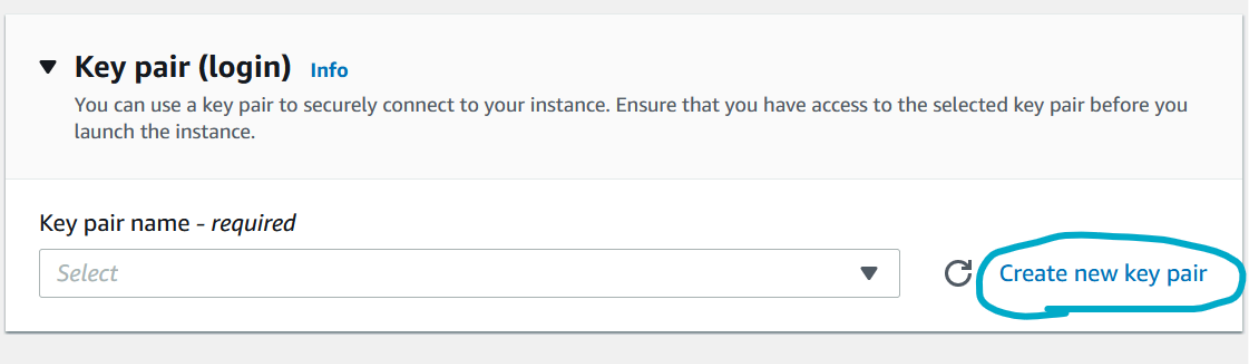
4.here we can select ami depending upon the requirement



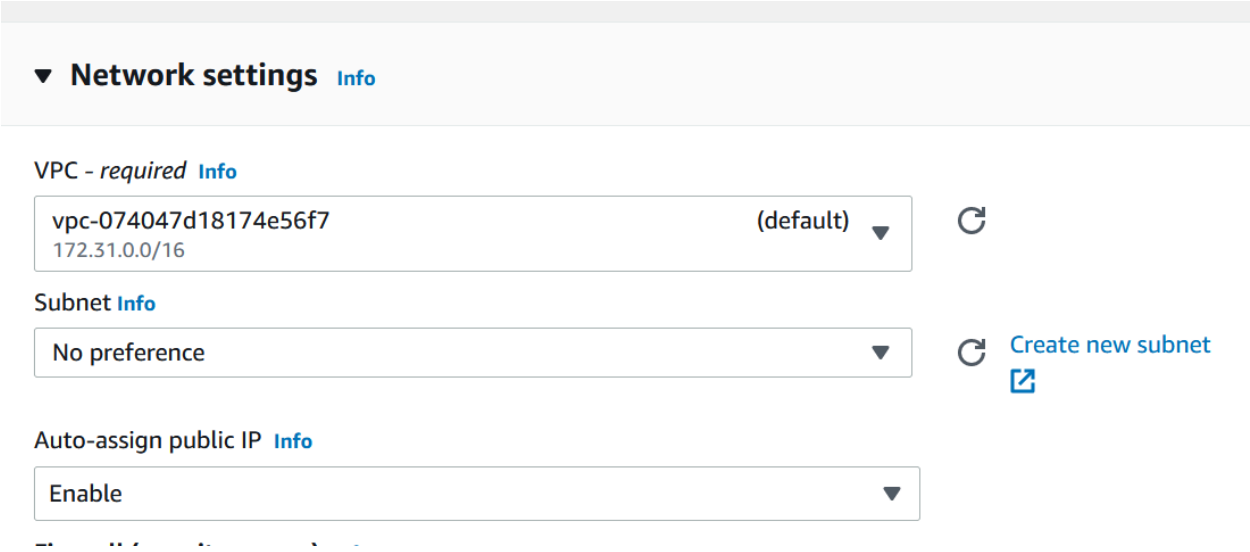
5.after that we can select the **family type**



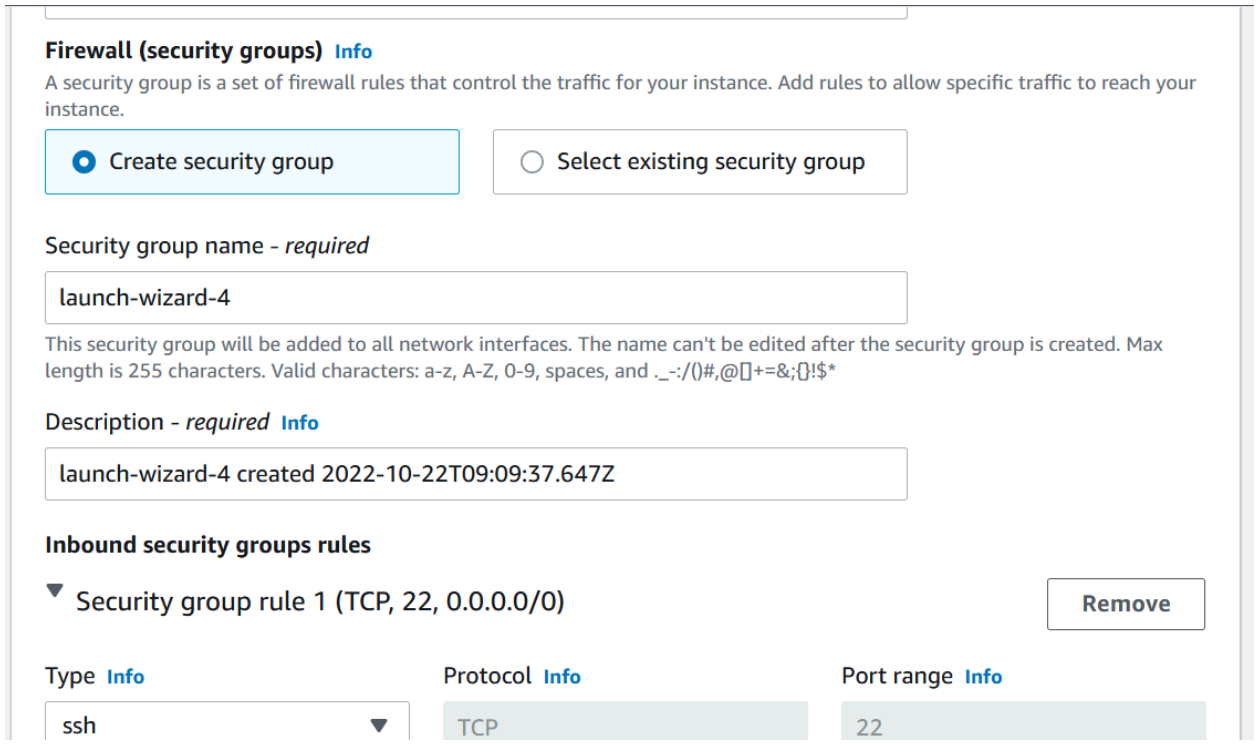
6.here we can select key pair, here we can create new **key pair** and select it



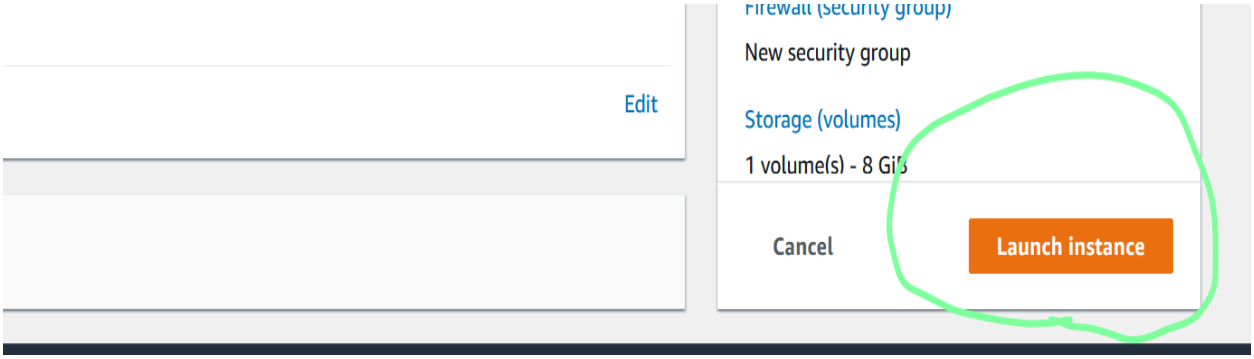
7.here Click on edit in network settings—>here we can select custom or default vpc—> here we can select the subnet



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



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

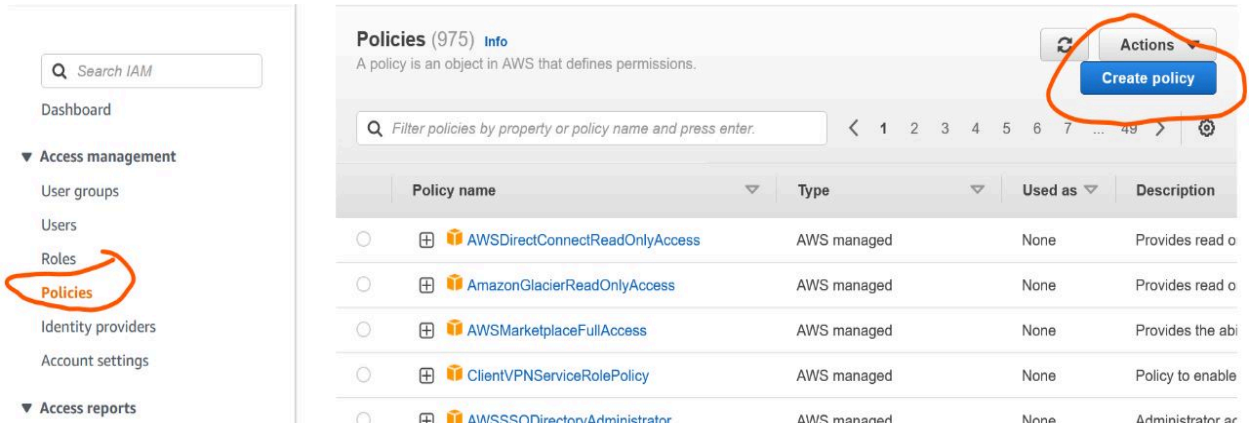
Custom policy
This policies are managed by the users

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

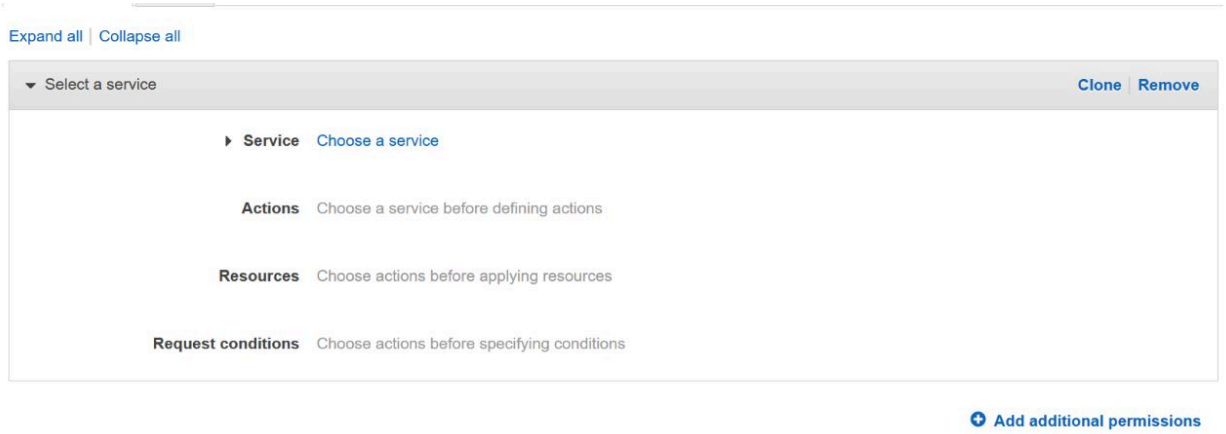
Policy format Syntax:

```
{

 Action:Resource/service
 Effect:Allow/deny
 Resource:
 Condition:
 Principal:

}
```

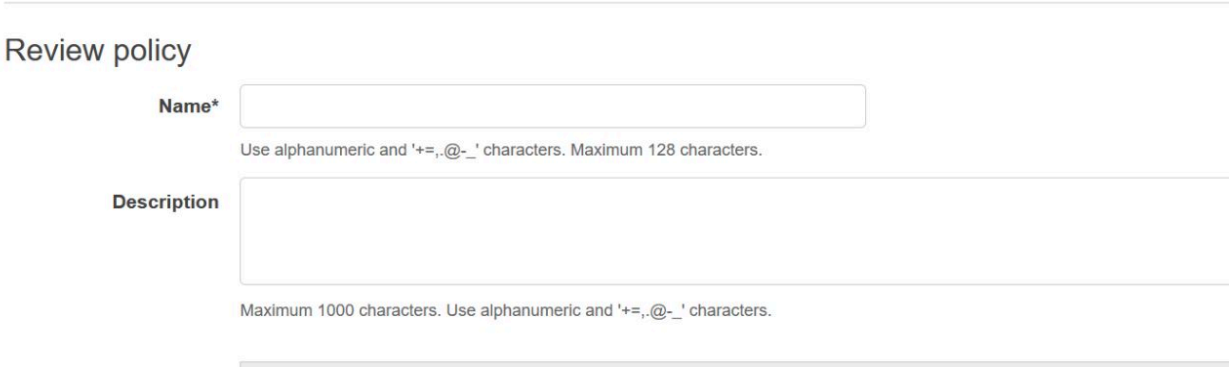
1. Click on policies—> and then click on create policy



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



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



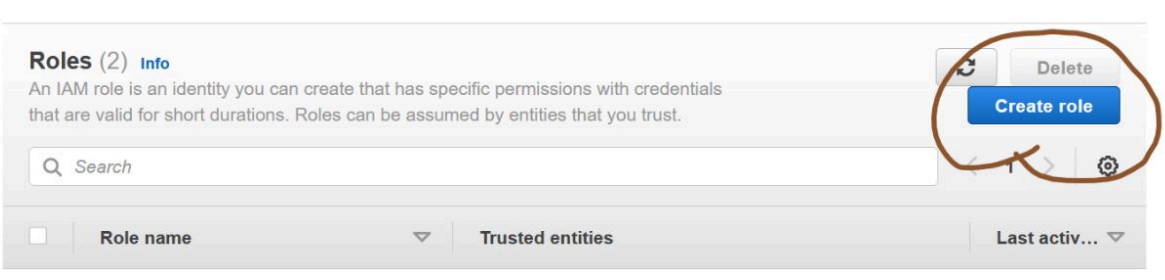
Next create role

How to create role
==========

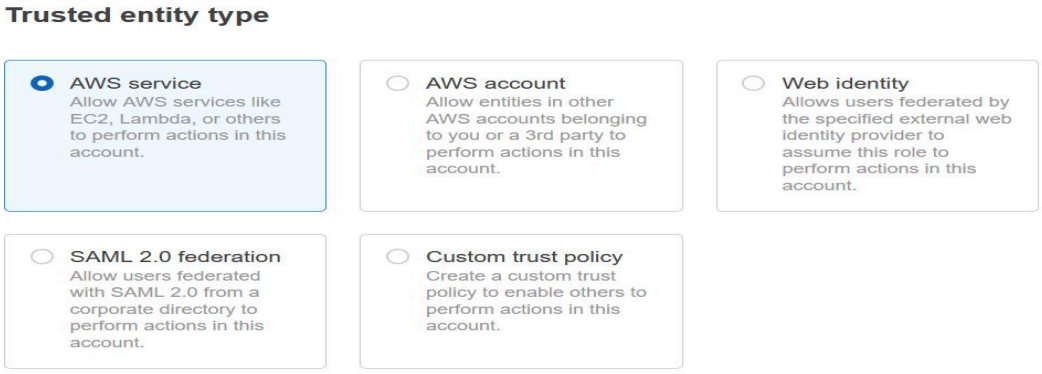
The purpose of the role is used to connect from one service to another service

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

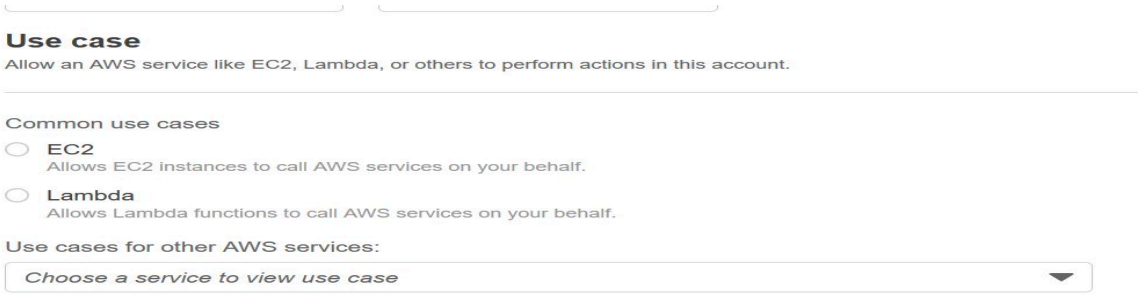
.



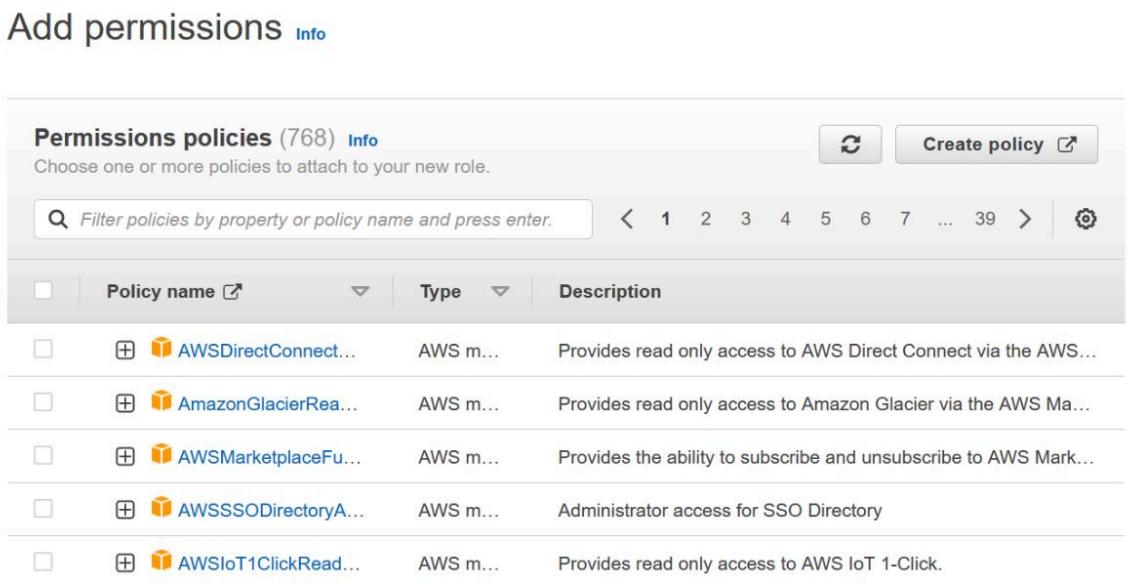
2.Select the trusted entity type



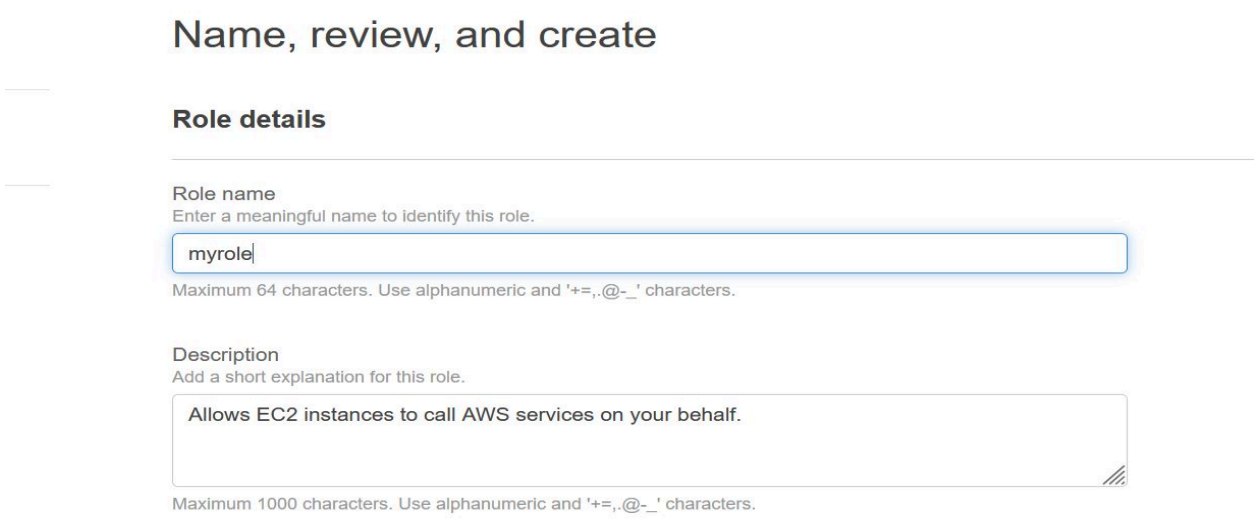
3. Select the use case for example in my case IAM using Lambda



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



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

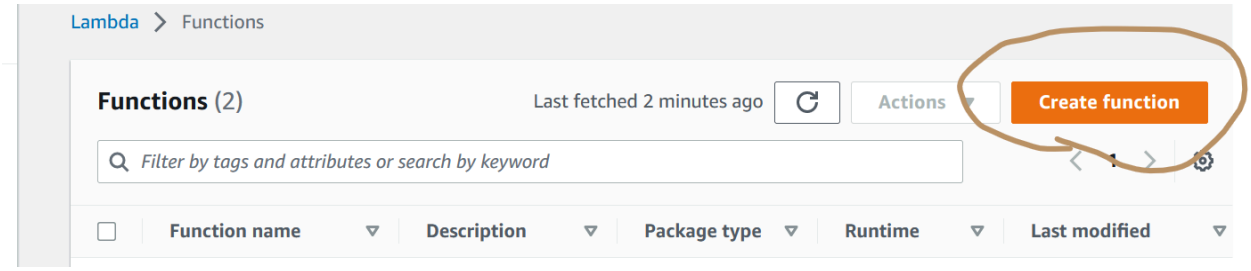


After that create lambda function

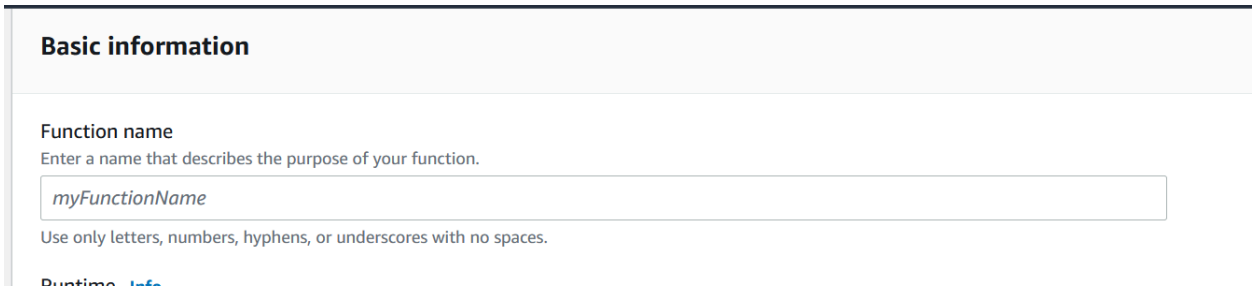
How to create lambda function:
==========

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

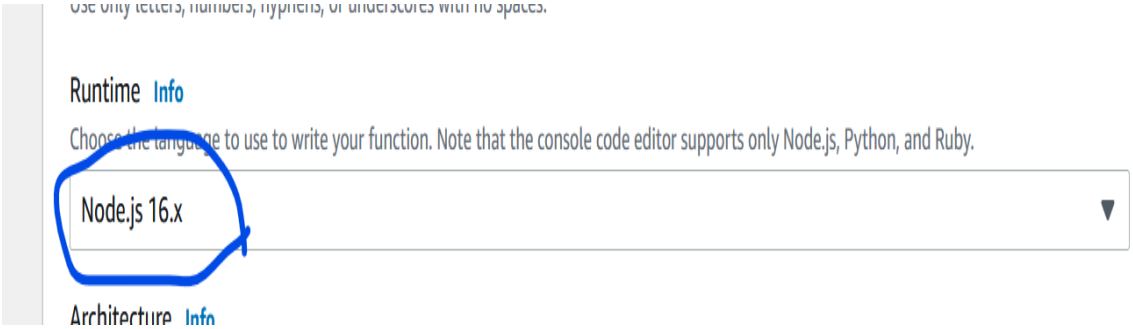
1.first go to lambda service—- and then click on create function.



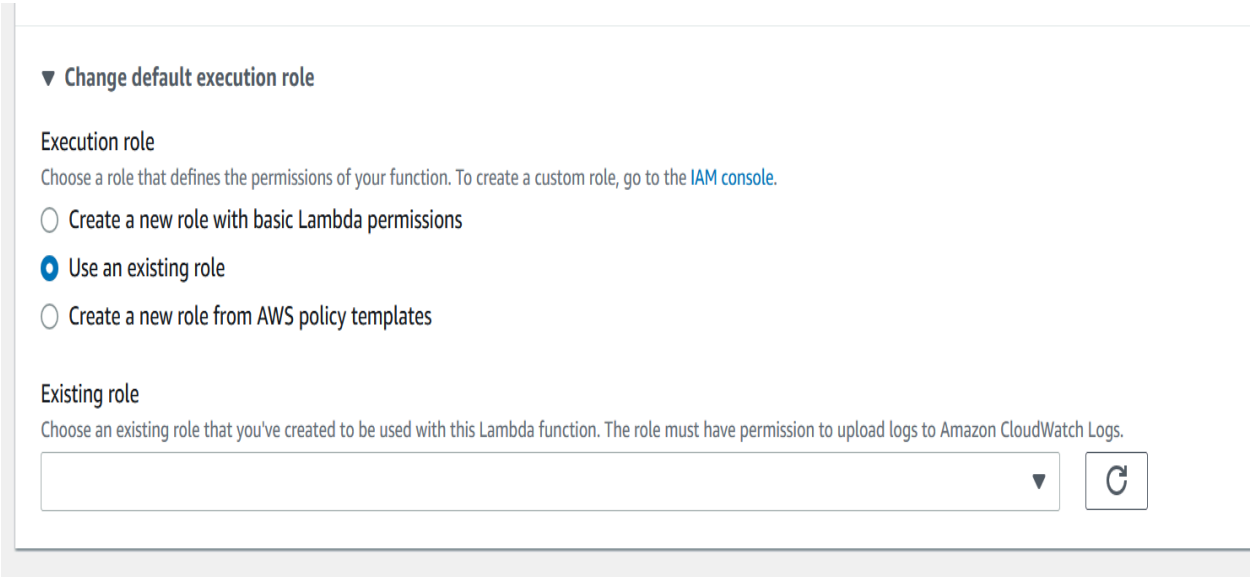
2. Here we can give proper name



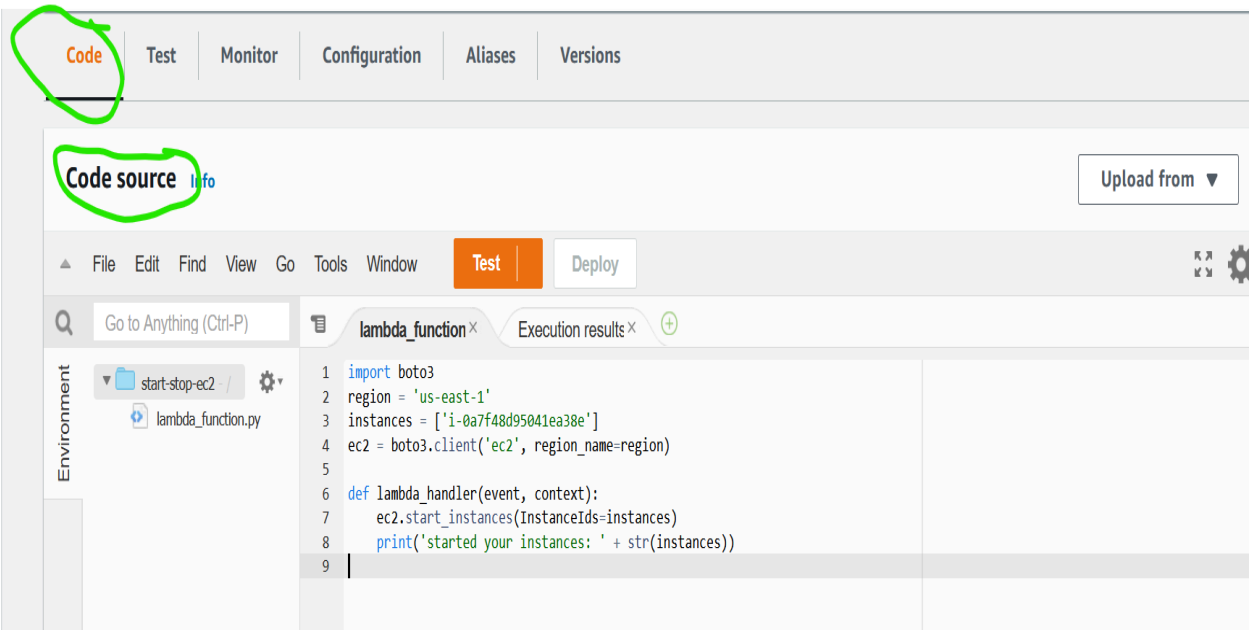
3.here we can chose the language



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



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.

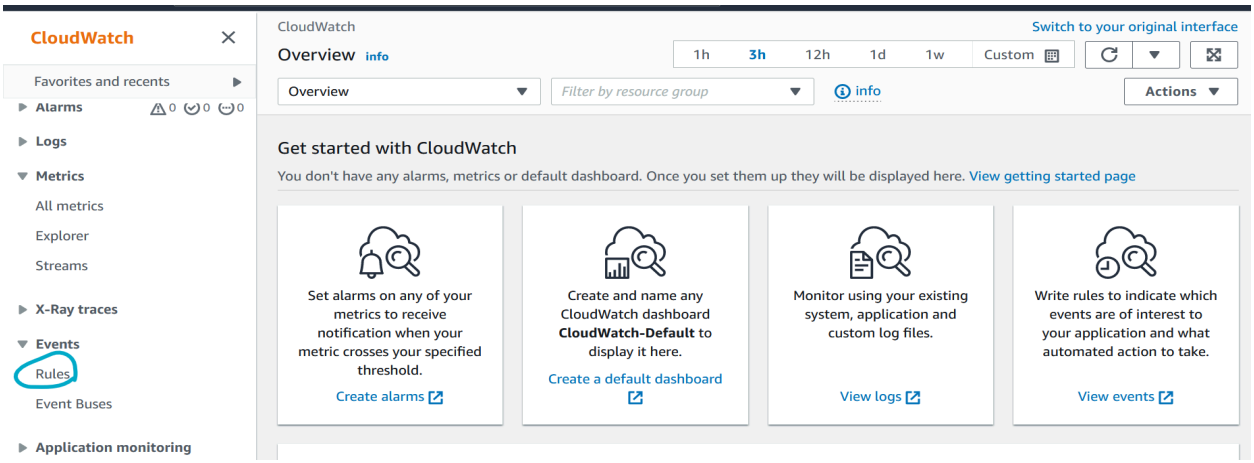
Note: after you can check the status of ec2 instances

How to set time in cloud watch

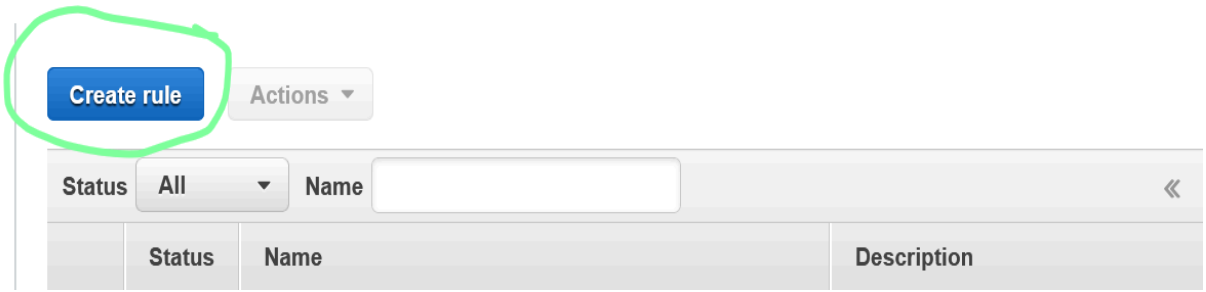
==========

It is a monitoring and observability service
To monitor the metrics like cpu,disc

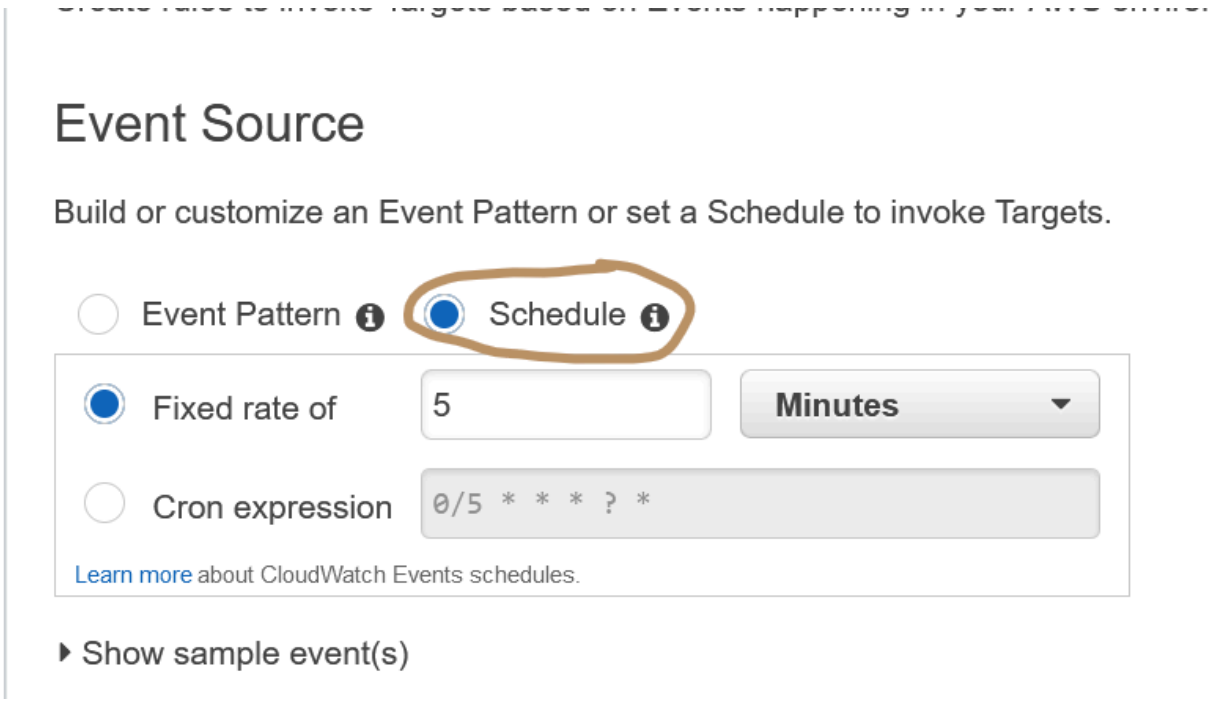
1.first go to **cloud watch** service—and go to **events**—-and select **rules**



2. Click on **create rule**



3.here we select schedule

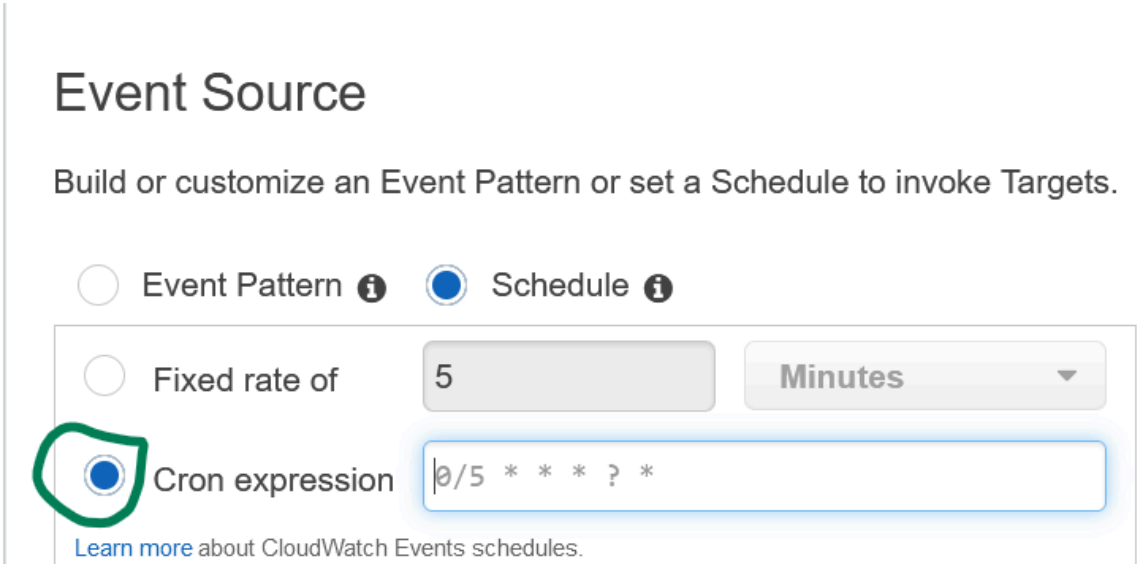


1. **Fixed rate:** enter an interval of time in minutes, hours, days.

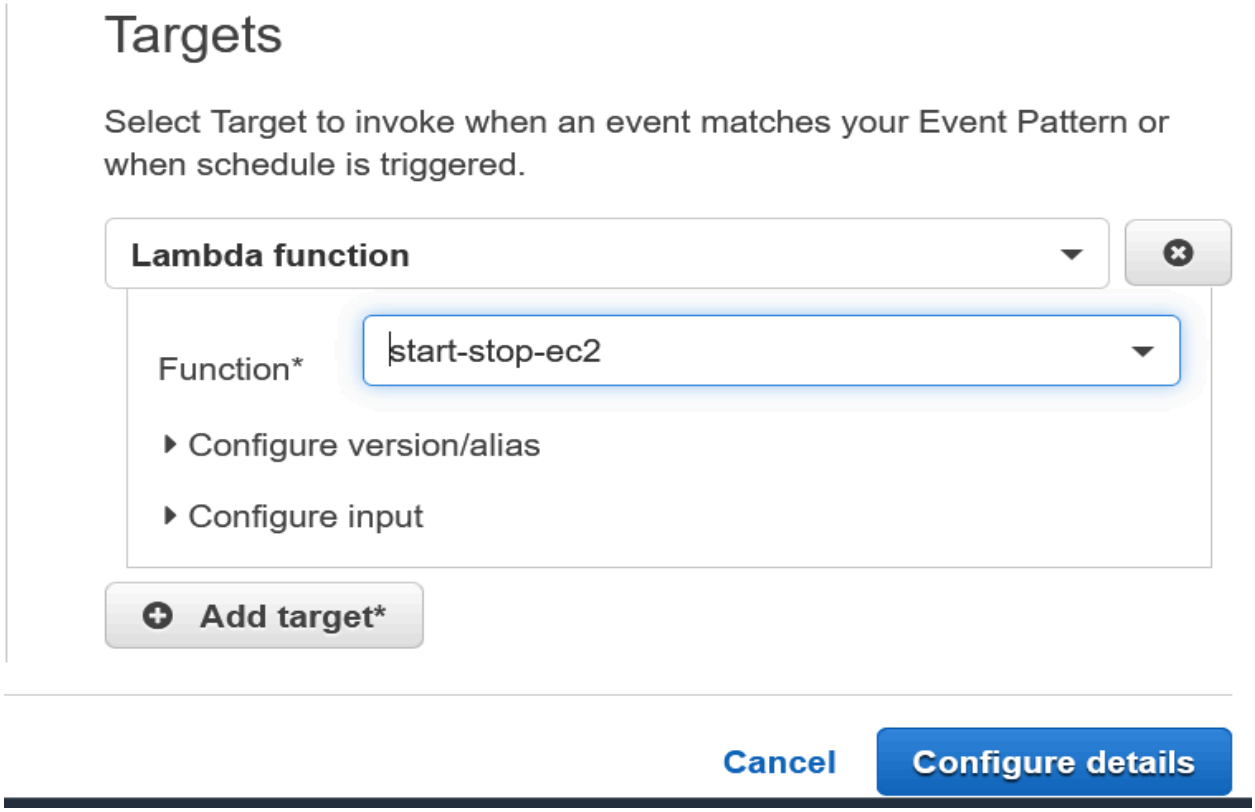
   Cron expression: enter a time that tells Lambda when to stop your instances.

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

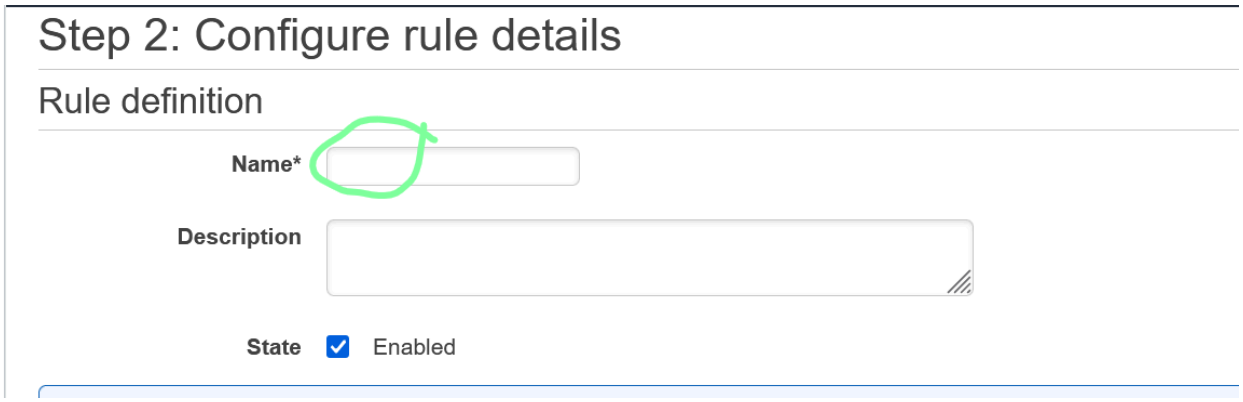
\



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



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



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.