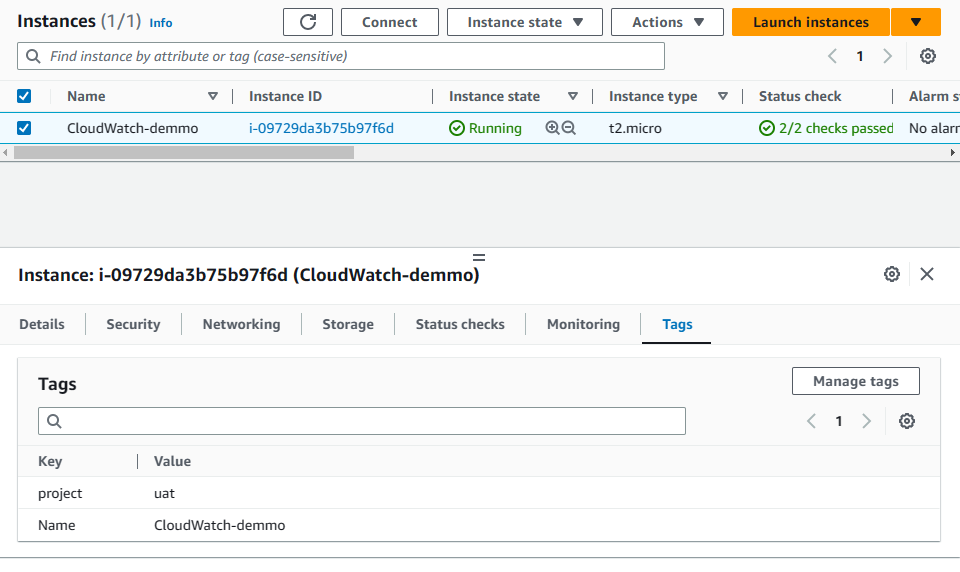
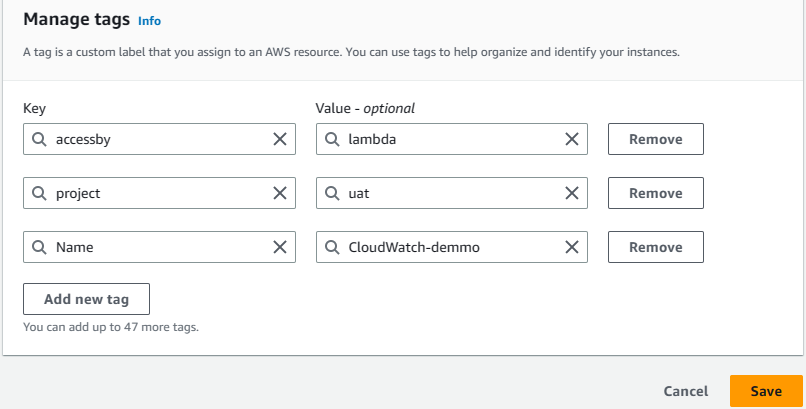
1. Navigate to EC2>Instances
2. Select instance and navigate to Tags tab.



1. Click to ‘Manage Tags’ and create a new tag Key= ‘accessby’, Value= ‘lambda’. Or anything you want. And save it.



1. Navigate to IAM>Policies and create a policy using this json:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"ec2:DescribeInstances",

"ec2:Start\*",

"ec2:Stop\*",

"ec2:DescribeInstanceStatus"

],

"Resource": "\*"

},

{

"Sid": "VisualEditor1",

"Effect": "Allow",

"Action": [

"logs:CreateLogStream",

"logs:CreateLogGroup",

"logs:PutLogEvents"

],

"Resource": "\*"

}

]

}

1. Create a role with selecting ‘Lambda’ as Common use cases and select policy which we created previous step.
2. Create a lambda function and in python using this code:

import os

import boto3

import logging

DEFAULT\_TAGS = 'tag:accessby=lambda'

ec2\_resource = boto3.resource('ec2')

ec2\_client = boto3.client('ec2')

def lambda\_handler(event, context):

    """

        Function that start and stop ec2 instances schedule and with specific tags<br/>

        :param event: Input event, that should contain action and tags parameters, where tags is a list of comma separates key/value tags.<br/>

        :param context: Lambda context.<br/>

        :return: nothing

    """

    tags = get\_tags(event['tags'] if 'tags' in event else DEFAULT\_TAGS)

    instances = get\_instances\_by\_tags(tags)

    if not instances:

        print('No instances available with this tags')

    else:

        if event['action'] == 'start':

            ec2\_client.start\_instances(InstanceIds=instances)

            print('Starting instances.')

        elif event['action'] == 'stop':

            ec2\_client.stop\_instances(InstanceIds=instances)

            print('Stopping instances.')

        else:

            print('No instances availables with this tags')

def get\_tags(tags):

    """

        Method that split comma separated tags and return a formed tags filter<br/>

        :param tags: Comma separated string with the tags values.<br/>

        :return: tags structure

    """

    final\_tags = []

    split\_tags = tags.split(",")

    for tag in split\_tags:

        values = tag.split('=')

        final\_tags.append({

            'Name': values[0],

            'Values': [values[1]]

        })

    return final\_tags

def get\_instances\_by\_tags(tags):

    """

        Method that filter all ec2 instances and return only the instances with specific tags<br/>

        :param tags: Filter structure with tag values.<br/>

        :return: list of ec2 instances

    """

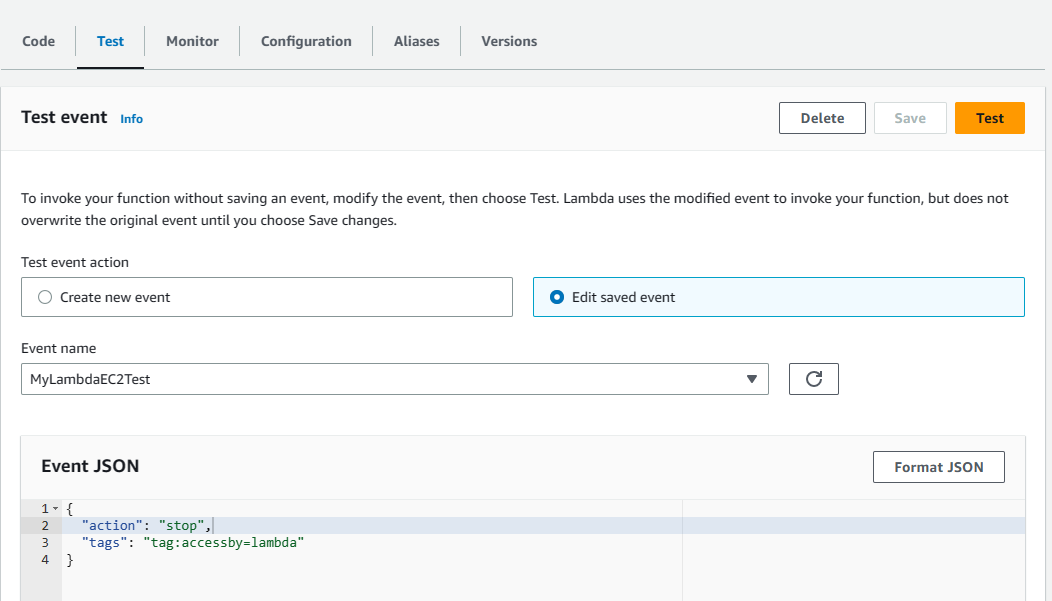
    response = ec2\_resource.instances.filter(Filters=tags)

    for instance in response:

    intance\_ids = [instance.id for instance in response]

    return intance\_ids

1. If your instance tag is another name provide your tag in json. Or if you set default tag just pass action name. Test function like this:



1. It will stop all instance with tag: ‘tag:accessby=lambda’.
2. You can start instance by changing action property as ‘start’.
3. Navigate to CloudWatch>Rules and create a role like this:

