

# CLOUD COMPUTING

University of Massachusetts Lowell



## AWS PROJECT

(IAM\_EC2\_S3\_RDS\_ROUTE53\_CLOUDFRONT\_L  
AMBDA\_CLOUDWATCH\_ELASTIC IP\_SES)

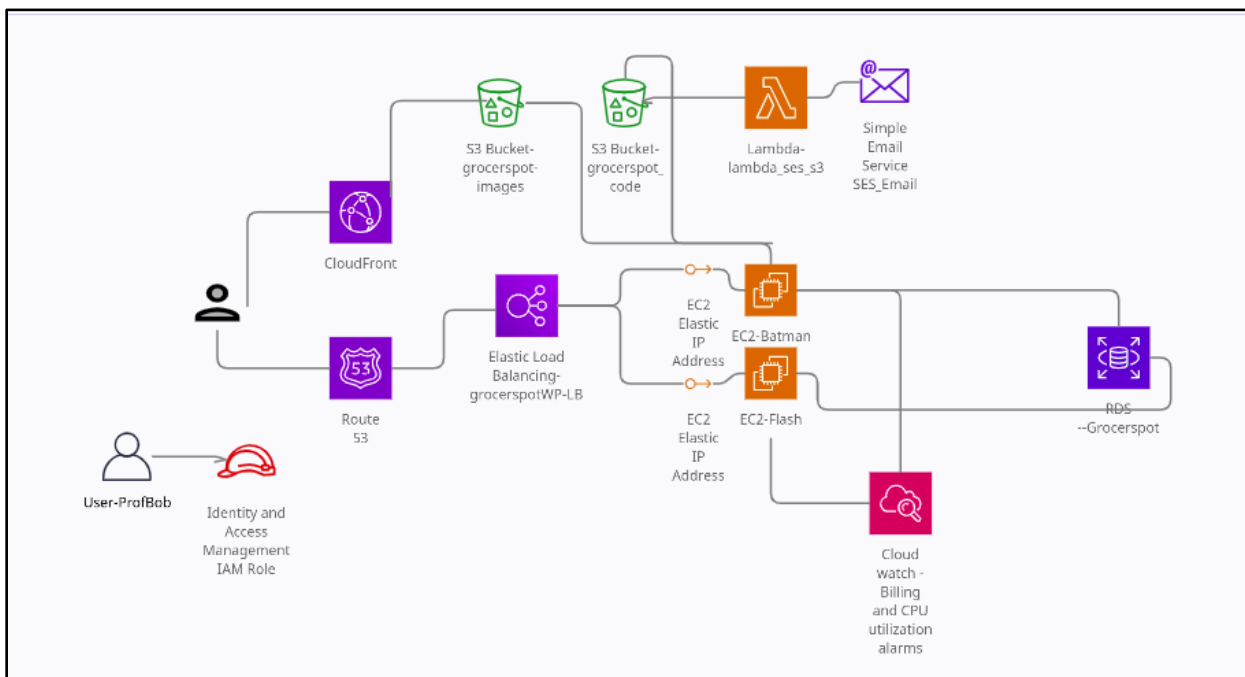
THARUN MADASU

[tharun\\_madasu@student.uml.edu](mailto:tharun_madasu@student.uml.edu)

## Introduction:

AWS is one of the most broadly adopted cloud platform by many organizations across the globe. We get to have end-to-end services for the infrastructure which I'm going to use in my project. I'll be mostly using free tier services which are offered by AWS which I had provided me ample hands-on experience. Moreover, the course lab exercises and reading assignments have done a great deed while working on my project. I relatively had very less cloud exposure and the only cloud hands-on all I had was my Azure and Google Projects. Using various free tier services( Both Tier 1 and Tier 2) of AWS, my main aim is to deploy a website on an EC2 instance that was registered to a domain name with the help of Route53 with hosted zone records to ELB. Two S3 buckets are used in this project one to store the media files of my website and another bucket to store the entire code repository of my website. As part of the Database implementation, I had chosen RDS for this project. The EC2 instance, FLASH was created as part of my future work in which I intend to make active with failover of my primary VM, BATMAN. Lambda(Tier 1 new) I configured a trigger function with SES. Finally, with the cloudwatch, Billing Alarm and CPU\_Utilization alarms are created.

Below is my architectural diagram of this project –

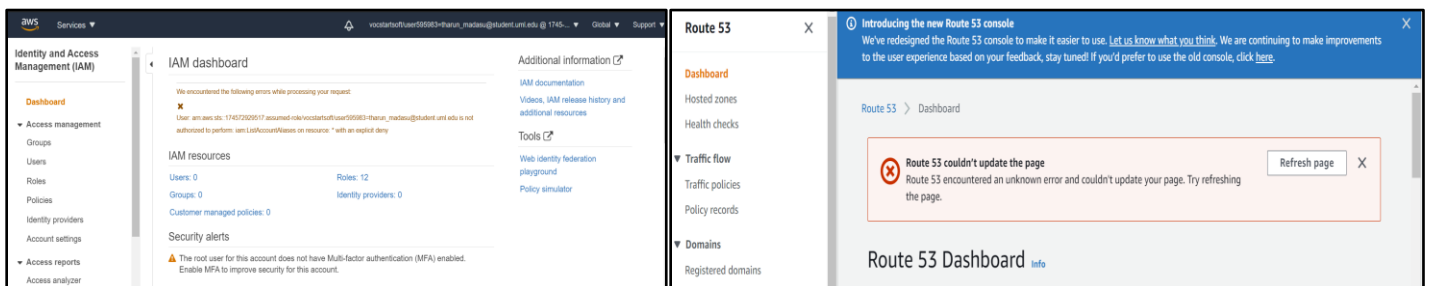
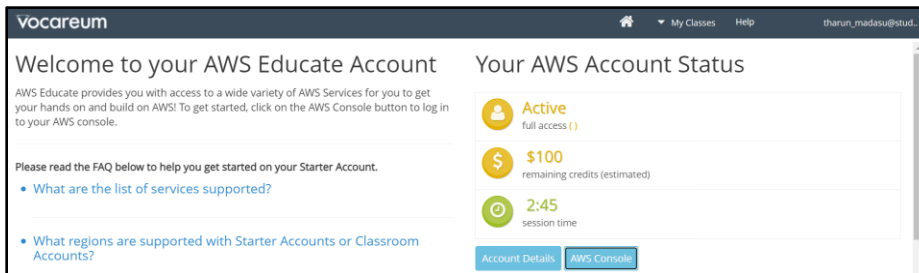


List of AWS services used in this Project –

List of AWS Services	
Tier 1	Tier 2
IAM	Elastic IP
CloudFront	Cloud Watch
EC2	Simple Email Service
S3	
RDS	
Route53	
Lambda	

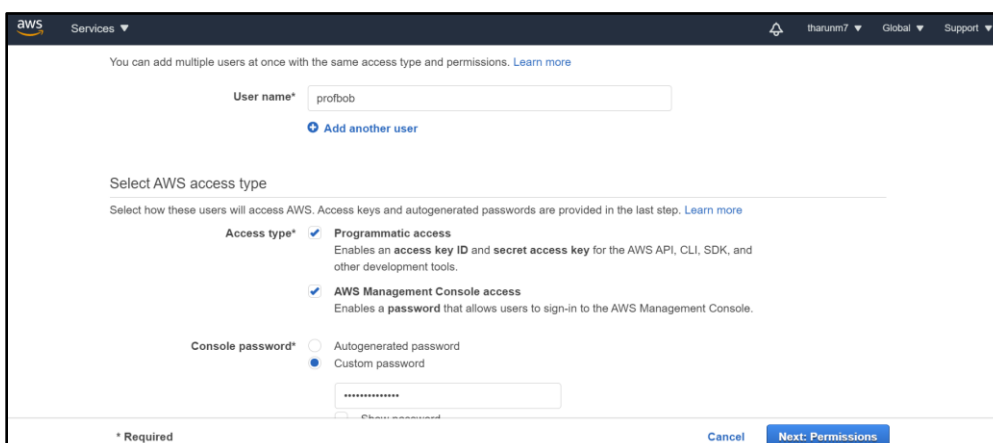
## AWS EDUCATE:

As part of my hands-on with AWS Cloud Services at first, I started with an AWS Educate account. I initially thought that I could transfer my \$100 free credits from my AWS Educate account to an AWS account. Only after contact with the customer representatives, I found out that these two are two different entities entirely and the credits cannot be transferred. And you do have quite some limitations on the AWS Educate account. A couple of them are IAM and Route53 services. It is highly advisable to get the credit code from the professor and start with your project early. I had received the credit code from the professor, and I was eligible for \$200 credit.

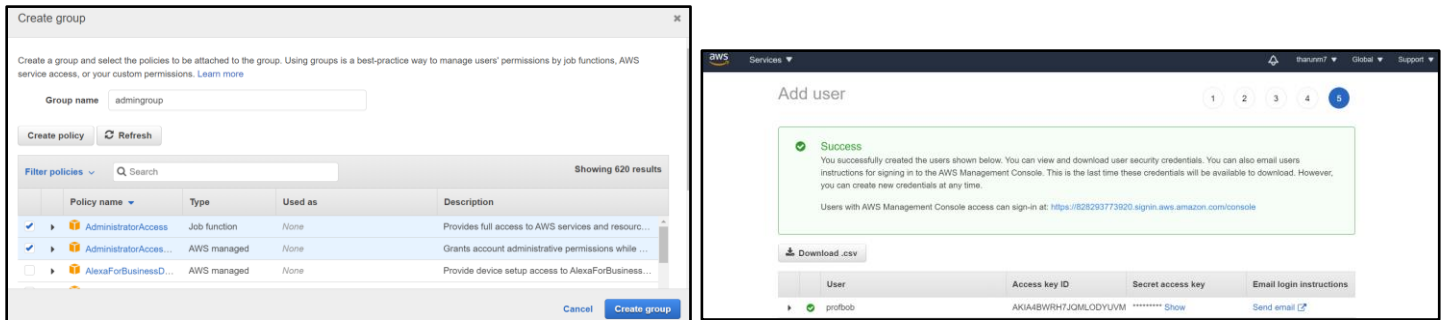


## Identity and Access Management (IAM)

One of the project requirements is to set up an account for Professor Bob with administrative access. From the Services option, I navigated to the IAM dashboard and there were many options. Users, Groups, Roles, Policies, etc. From the user's option, I provided the username and password as mentioned in the AWS\_project\_description document. For the access types, I check both the options – Programmatic access and AWS Management Console access. Since the Professor had already given me a password I unchecked the option to create a new password on sign-in. I did observe that I could add multiple users and add them to different groups with different policy setups. This is fascinating and comes in handy for the IT team when their organization opts for AWS.



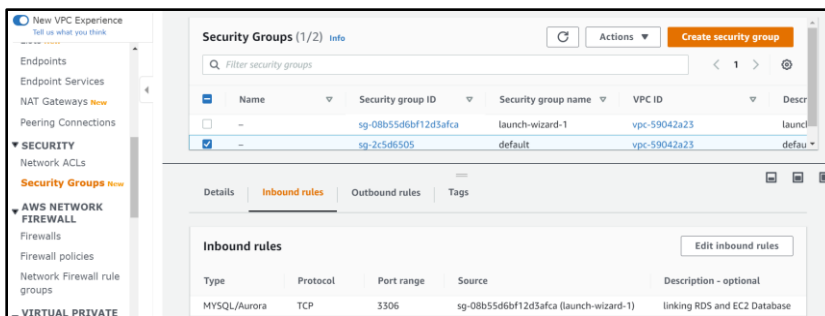
Onto the next step, I created a group 'admingroup' with policies that provide full access to AWS Services and resources along with account administrative permissions. Then I've created a Tag for the IAM user - AWSProject and created the user. Here's one thing that I appreciate as an AWS user is that I could share these details in an email. I've created one more user – adminTM and added it to the same IAM group as above.



Here's the URL for Professor to login into the account: <https://828293773920.signin.aws.amazon.com/console>

## AWS RDS:

Since my database will be on a separate instance and not be on the same instance (BATMAN) where my WordPress Site will be installed, they will not compete for resources. As per my future work, I wanted to have a back-up instance so, even in that case, I could connect to a single MySQL instance on AWS RDS which will allow me to scale my site. Also, RDS supports automated security patch updates and backups which helps in the smooth administration of my database. So for these reasons I chose to go with AWS RDS. Before jumping on creating my RDS, I navigated to the security groups in VPC service and wanted to make sure that RDS MYSQL/Aurora port: 3306 is opened to the security group, launch-wizard-1. To do that I clicked on the add rule, and from type, I searched for 'MySQL' and in the source, I searched 'sg' and selected the required security group.

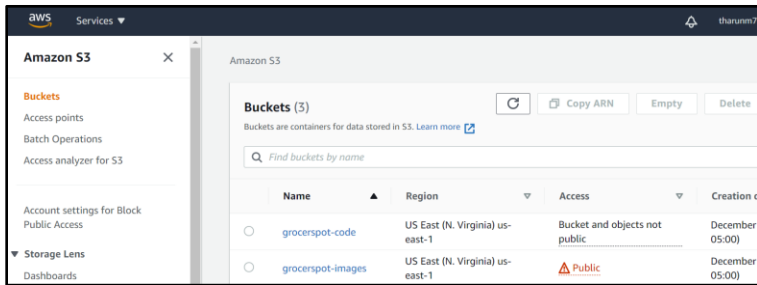


Coming on to the RDS, I wanted to have a database for my website, and for that, I chose the AWS RDS service. After clicking on creating a database, I chose the standard DB creation method and MySQL as the Engine for my DB and the version was left as the default one and 'Free Tier' as the template type. FOR the DB instance size, I have chosen DB.t2.micro and from the storage option, I unchecked the auto-scaling feature as I will not be needing it for this project. I left the default options for VPC and Subnets. The DB name and password will be the same as grocerspot.

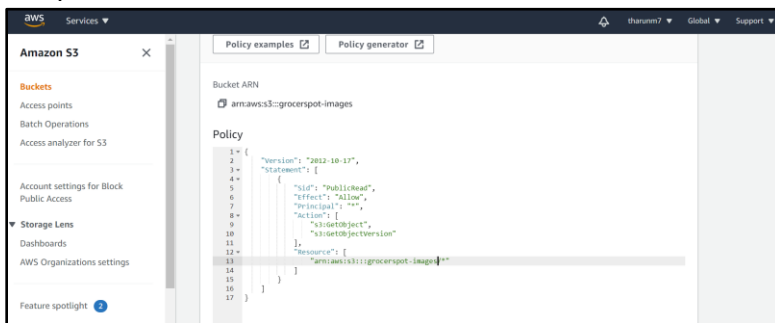
## AWS S3:

A new S3 bucket, grocerspot-images has been created for all the media that contains on my website. The other S3 bucket is grocerspot-code. I could lose this EC2 Instance (BATMAN) at any time. So what I want is a full copy of my website in my S3 Bucket, grocerspot-code. So, even if I did lose my EC2 Instance, I can have an Auto Scaling Group with a bunch of EC2 Instances as soon as they boot, they will pull down the code from the S3 Bucket. I

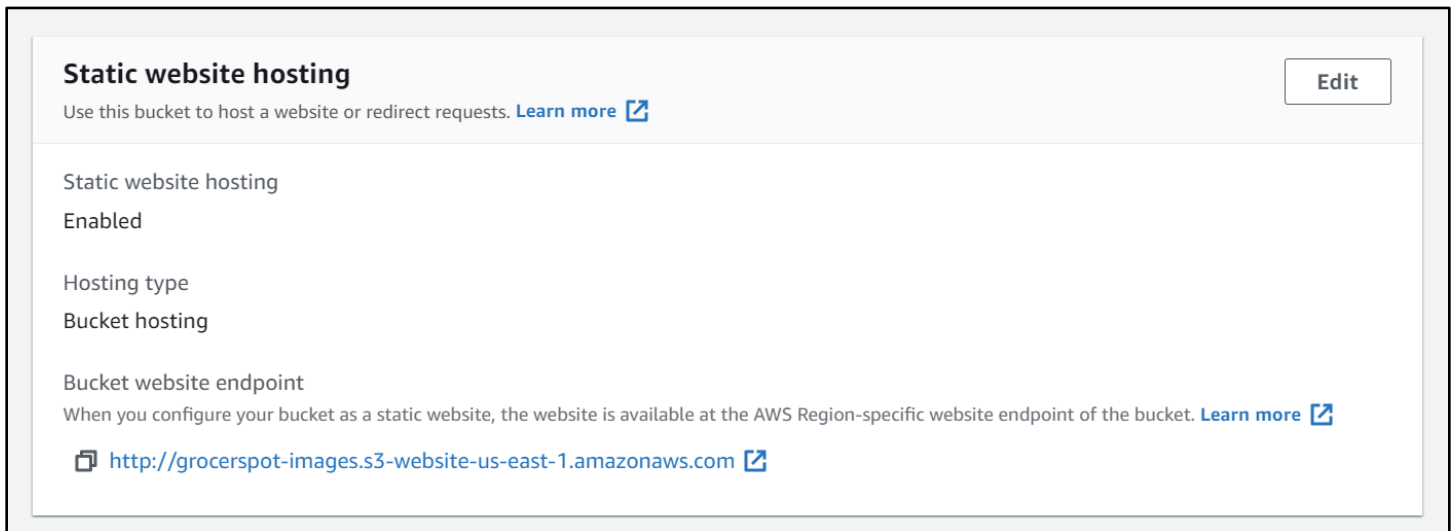
have blocked all the public access to this bucket and disabled object lock, encryption, and versioning of this bucket.



Also, I wanted to make the images in my S3 bucket – grocerspot-images, publicly available on my website. To do that, I had to edit the policy such that it grants read-only permission to an anonymous user. Many policy templates are available in the below-mentioned link. I have also enabled static-website hosting for this bucket.



<https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-2>



## AWS CloudFront:

From CloudFront, I've selected a web-based distribution and the origin domain name will be similar to the S3 bucket which was created before, and the rest of the settings and options were left as is and the distribution is created. So, what I wanted to do is that every time I upload an image to my WordPress site, that that file is also stored in S3 for redundancy. Eventually, I'm going to force CloudFront to serve those files using the CloudFront distribution rather than using the images on my EC2 Instance because then the site will load a bit faster.

**Create Distribution**

Origin Settings

Origin Domain Name:

Origin Path:

Enable Origin Shield: ☐ Yes ☒ No

Origin ID:

Restrict Bucket Access: ☐ Yes ☒ No

Origin Connection Attempts:

Origin Connection Timeout:

Origin Custom Headers: Header Name Value

**CloudFront Distributions**

Create Distribution Distribution Settings Delete Enable Disable

Viewing: Any Delivery Method Any State

Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State
Web	E16IHAA3A66F0N	d1cud5ffb69ohz	-	grocespot	-	In Pro	Enable

```
ubuntu@BATMAN: /var/www/html
GNU nano 4.8
Options +FollowSymLinks
RewriteEngine on
rewriterule ^wp-content/uploads/(.*)$ http://d1cud5ffb69ohz.cloudfront.net/$1 [r=301,nc]
```

So from above, I have changed the URL rewrite rule in the .htaccess file which essentially allows my website to serve my images out of CloudFront instead of serving the images out of EC2. So, from the CloudFront dashboard, I got the domain name and changed the previous URL to my CloudFront URL. And later when I sync my /var/www/html/ directory to update this URL path.

## AWS EC2:

As part of my project, I wanted to have 2 EC2 instances. From the services option, I could see EC2 as the first service listed down. I selected it and then I chose the option to launch the instance. Both my instances were launched with the help of the AWS management console. I have selected Ubuntu-20 image with t2.micro as the instance type with 1GB memory and 1vCPU and 8GB SSD storage. I have named one of the instances as 'BATMAN' and the other as 'FLASH' I have an unchecked deletion on the termination option while creating the instance. I have also created an inbound traffic rule for HTTP port 80. Under tags, I have this EC2 instance under AWSProject. Apart from these I left all other options as default and created the instance. Also, I added both these instances into the same security group and to make things simple I used the same key-pair.

**Instances (2/2)**

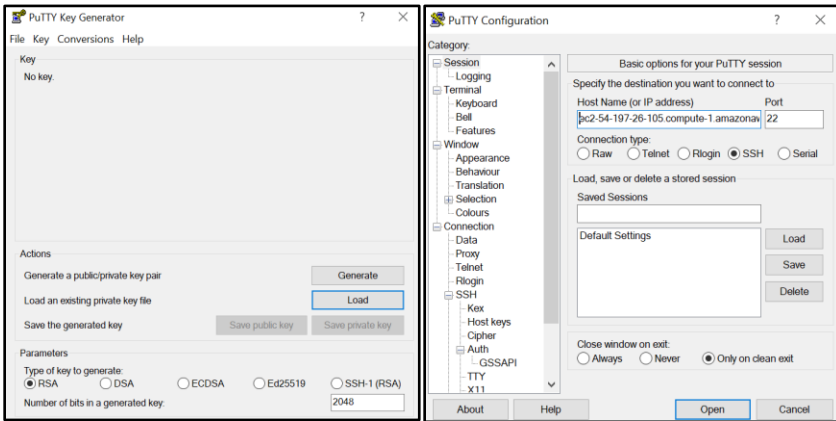
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avi
BATMAN	i-06198e04e49f5d0ef	Running	t2.micro	2/2 checks ...	No alarms +	us-
FLASH	i-06512c36aa1530c37	Running	t2.micro	2/2 checks ...	No alarms +	us-

**Instance summary for i-06198e04e49f5d0ef (BATMAN)**

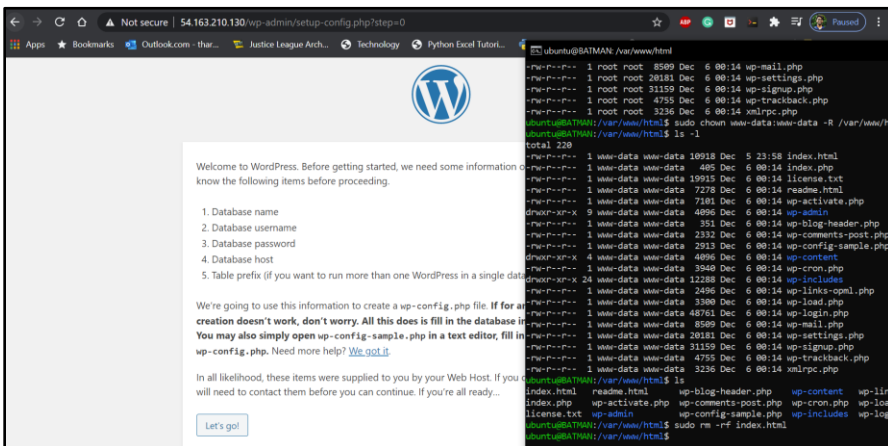
Instance ID	Public IPv4 address	Private IPv4 addresses
i-06198e04e49f5d0ef (BATMAN)	35.172.190.43   open address	172.31.26.57
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-35-172-190-43.compute-1.amazonaws.com   open address	ip-172-31-26-57.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-59042a23
AWS Compute Optimizer finding	IAM Role	Subnet ID
Opt-in to AWS Compute Optimizer for recommendations.   Learn more	-	subnet-e618baab

Port range	Protocol	Source	Security groups
3306	TCP	sg-08b55d6bf12d3afca	default
80	TCP	0.0.0.0/0	launch-wizard-1
80	TCP	::/0	launch-wizard-1
22	TCP	73.114.64.153/32	launch-wizard-1

To connect to my instances, I opted for PuTTY. But the downloaded key pair(groceries-site-key) was in .pem and PuTTY needs .ppk format. Therefore, using the PuTTY key generator I have loaded the .pem file and generated a .ppk file along with a passphrase for the key. I've inserted my public IPv4 DNS as the hostname and in the SSH option, I've loaded the .ppk file that I have just generated. The PuTTY window opens asking for the username, and I've just given the default one which is 'ubuntu'. After successful login into my instances, I updated them and installed a web server in this case Apache.

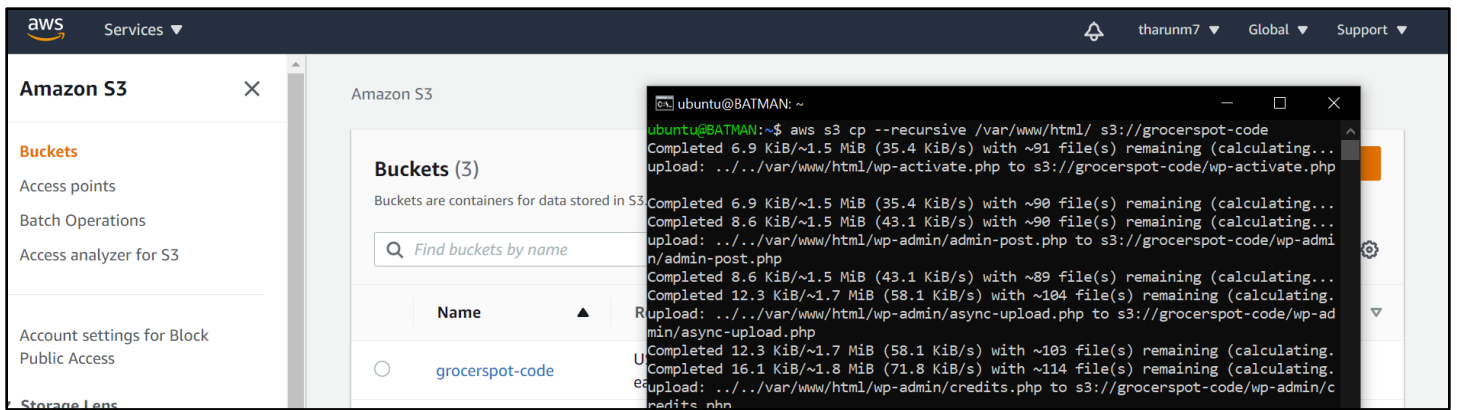


Now, WordPress has been installed on the webserver. I have changed the database host from 'localhost' to the endpoint of my RDS(grocerspot.cf1cvxkhdwc4.us-east-1.rds.amazonaws.com). Also, initially while launching the instance, I didn't have any specific IAM role for it. So, I have created an IAM under EC2 with S3 full access as the policy and I have updated the IAM role of my BATMAN instance from Actions→Security→Modify IAM role. I have done this because anytime if an image is uploaded into my WordPress site, that file should also store in my S3 bucket for redundancy. And eventually what I'm going to force CloudFront to serve those files using the CloudFront distribution, rather than using the images on my EC2 Instance resulting in loading the website a bit faster. As you can see now from my CLI, there were two S3 buckets listed out and grocerspot-images is the one I created earlier for this project. So, the next thing I want to do add a bit of redundancy. So I'll copy the entire directory in another bucket 'grocerspot-code'.



```
ubuntu@BATMAN:~$ aws s3 ls
2020-12-05 21:27:50 grocerspot-images
2020-12-05 22:46:12 grocerspot-veg-bucket
ubuntu@BATMAN:~$ aws s3 cp --recursive /var/www/html/wp-content/uploads s3://grocerspot-images
upload: ../../var/www/html/wp-content/uploads/2020/12/Head-Image-1024x438.jpg to s3://grocerspot-images/2020/12/Head-Image-1024x438.jpg
upload: ../../var/www/html/wp-content/uploads/2020/12/Head-Image.jpg to s3://grocerspot-images/2020/12/Head-Image.jpg
upload: ../../var/www/html/wp-content/uploads/2020/12/Head-Image-1200x513.jpg to s3://grocerspot-images/2020/12/Head-Image-1200x513.jpg
upload: ../../var/www/html/wp-content/uploads/2020/12/Head-Image-300x128.jpg to s3://grocerspot-images/2020/12/Head-Image-300x128.jpg
upload: ../../var/www/html/wp-content/uploads/2020/12/Head-Image-150x150.jpg to s3://grocerspot-images/2020/12/Head-Image-150x150.jpg
upload: ../../var/www/html/wp-content/uploads/2020/12/Head-Image-768x328.jpg to s3://grocerspot-images/2020/12/Head-Image-768x328.jpg
ubuntu@BATMAN:~$
```

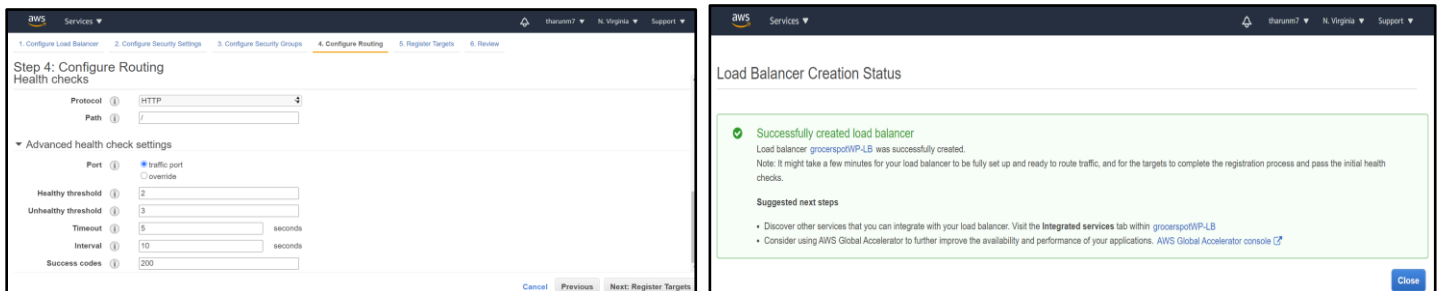




With that, all the images on my website had now been stored in my S3 bucket, grocerspot-images in folder 2020.

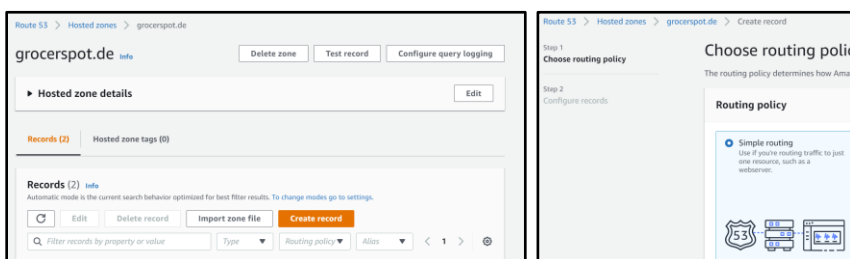
## Load Balancer:

From the EC2 service dashboard, you can find the load balancer option. I'm going with the application load balancer for this project which will be internet-facing and named it 'grocerspotWP-LB'. I've checked all the boxes for availability zones and opening up only for HTTP port. For the security group, I have chosen my existing security group, i.e. 'launch-wizard-1'. Under configuring routing options, I have created a new target group, WPInstances, and the type as Instances. From advanced health settings, make sure that the timeout should be lesser than the interval value. I have registered my BATMAN Instance and successfully created the load balancer. It changed its state from provisioning to active in just a couple of seconds.



## AWS Route53:

As part of my lab exercise, I had registered a Domain Name (It's \$9 though) because it is not accepting the domain names that I created used the freenom website. Check the availability of the domain name and, select one. The domain will be registered to your AWS account after the payment is done. After that under DNS management, I selected the hosted zone option and I wanted to create a record with a simple routing policy. I have defined the simple record as to show in below snap with respect to my load balancer. Now, the domain is pointing towards our load balancer.





Define simple record

Value/Route traffic to

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

Alias to Application and Classic Load Balancer

US East (N. Virginia) [us-east-1]

Record type

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A - Routes traffic to an IPv4 address and some AWS resources

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Evaluate target health

Select Yes if you want Route 53 to use this record to respond to DNS queries only if the specified AWS resource is healthy.

☒ Yes

Cancel

Define simple record

<input type="checkbox"/>	Record name	Type	Routing policy	Difference	Alias	Value/Route traffic to
<input type="checkbox"/>	grocerspot.de	A	Simple	-	Yes	dualstack.grocerspotwp-lb-1501583248.us-east-1.elb.amazonaws.com.

I now need to place my BATMAN Instance in a target group. So, from my EC2 dashboard, I chose the Target Group option and selected my instance and added it as pending at first, and then registered it in the target group. I could see the state as 'healthy' for my instance after a couple of refreshes.

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

Target Groups

Auto Scaling

Launch Configurations

Auto Scaling Groups

Group details

Targets

Monitoring

Tags

Registered targets (0)

Instance ID

Name

Port

Zone

Status

Status details

No registered targets

You have not registered targets to this group yet

Register targets

Targets (1)

Remove all pending

Remove	Status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
<input type="checkbox"/>	Pending	i-06198e04e49f5d0ef	BATMAN	80	running	default, launch-wizard-1	us-east-1a	subnet-e618baab

1 pending

Cancel

Register pending targets

After that, I tried to open my website, Et Voila....!!!

GROCERSPOT - Getting Groceries

Not secure | grocerspot.de

ABP

Paused

Getting Groceries Just Got Simple...

Sample Page

Search

GROCERSPOT

By Tharun Madasu

December 6, 2020

No Comments

We deliver farm fresh fruits and vegetables to our customers in and around the Greater Boston Area.


54.163.210.130/index.php/2020/12/06/grocerspot/


## LAMBDA & Simple Email Service:


Lambda is an event-driven, serverless computing platform provided by Amazon. Lambda is used to run code in response to events, such as changes to data in an Amazon Simple Storage Service (Amazon S3) bucket or dynamo DB. I aim to receive an email notification when an event occurs in an S3 bucket. To achieve this, I need to implement a lambda function to trigger a response if there's any addition or deletion of a file(s) in my S3 bucket. Firstly, I have created a role using IAM for lambda as a use case and giving it full access to AmazonSESFullAccess and CloudWatchLogsFullAccess as its policies. Lastly, I gave a role name and created a new role for the lambda function. Now, I have to create a new lambda function and assign this role.


Create role

Select type of trusted entity

 **AWS service**  
EC2, Lambda and others

 **Another AWS account**  
Belonging to you or 3rd party

 **Web identity**  
Cognito or any OpenID provider



Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

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.



Role name\*

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



Role description

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

Trusted entities AWS service: lambda.amazonaws.com

Policies  [AmazonSESFullAccess](#)  [CloudWatchLogsFullAccess](#)

Lambda > Functions

Functions (1) Last fetched 10 seconds ago  Actions  [Create function](#)

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

Runtime [Info](#)  
Choose the language to use to write your function.  

Python 3.8

Permissions [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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.



After selecting the create function, I had to give it a name and I did choose the language as Python, however, there were many languages available, to write my function. From role selection, I have selected the one which I have just created. I left everything to default in the settings and went ahead and created the function. The code below responds to the event and sends an email using the SES service.

```

1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5
6     for i in event["Records"]:
7         action = i["eventName"]
8         ip = i["requestParameters"]["sourceIPAddress"]
9         bucket_name = i["s3"]["bucket"]["name"]
10        object = i["s3"]["object"]["key"]
11
12        client = boto3.client("ses")
13
14
15        subject = str(action) + 'Event from ' + bucket_name
16        body = """
17        <br>
18        This email is to notify you regarding {} event.
19        The object {} is deleted.
20        Source IP: {}
21        """.format(action, object, ip)
22
23        message = {"Subject": {"Data": subject}, "Body": {"Html": {"Data": body}}}
24
25        response = client.send_email(Source = "tharun_nadasu@student.unl.edu", Destination = {"ToAddresses": ["tharun_nadasu@student.unl.edu"]})
26
27        # TODO implement
28        return {
29            'statusCode': 200,
30            'body': json.dumps('Hello from Lambda!')}
31
32
33

```

Event notifications (1)
Edit
Delete
Create event notification

Name	Event types	Filters	Destination type	Destination
<h3>Create event notification</h3> <p>The notification configuration identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications. <a href="#">Learn more</a></p> <div> <h4>General configuration</h4> <p>Event name</p> <input type="text" value="Lambda_Event"/> <p>Event name can contain up to 255 characters.</p> <p>Prefix - optional</p> <p>Limit the notifications to objects with key starting with specified characters.</p> <input type="text" value="images/"/> <p>Suffix - optional</p> <p>Limit the notifications to objects with key ending with specified characters.</p> <input type="text" value="jpg"/> </div> <div> <input checked="" type="checkbox"/> All object create events  s3:ObjectCreated:* <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Put s3:ObjectCreated:Put</li> <li><input checked="" type="checkbox"/> Post s3:ObjectCreated:Post</li> <li><input checked="" type="checkbox"/> Copy s3:ObjectCreated:Copy</li> <li><input checked="" type="checkbox"/> Multipart upload completed s3:ObjectCreated:CompleteMultipartUpload</li> </ul> <input checked="" type="checkbox"/> All object delete events  s3:ObjectRemoved:* <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Permanently deleted s3:ObjectRemoved:Delete</li> <li><input checked="" type="checkbox"/> Delete marker created s3:ObjectRemoved:DeleteMarkerCreated</li> </ul> </div> <div> <p><b>Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. <a href="#">Learn more</a></b></p> <p>Destination</p> <p>Choose a destination to publish the event. <a href="#">Learn more</a></p> <p><input checked="" type="radio"/> Lambda function Run a Lambda function script based on S3 events.</p> <p><input type="radio"/> SNS topic Send notifications to email, SMS, or an HTTP endpoint.</p> <p><input type="radio"/> SQS queue Send notifications to an SQS queue to be ready by a server.</p> <p>Specify Lambda function</p> <p><input checked="" type="radio"/> Choose from your Lambda functions</p> <p><input type="radio"/> Enter Lambda function ARN</p> <p>Lambda function</p> <input type="text" value="Lambda_SES_event"/> </div>				

Next, I have configured event notification in the S3 bucket – Groceries\_images. Next, I have selected events that trigger the lambda function. Next, I have assigned our lambda function as a destination to this event, so on upload or delete of file on S3 the event triggers lambda function, from there lambda executes a code to receive the event information and configured with SES to send an email. SES is a cloud-based email sending service designed to help digital marketers and application developers send marketing, notification, and transnational emails. For lambda to send the email provided in the code, the email address must be registered with SES. Now to register my email id with SES, I have entered it and clicked on verify my email address. As soon as, I have received the verification email, I clicked the link to do the needful. And, once the email is verified my lambda function can trigger/generate a notification on any upload and delete of file in the S3 bucket.

Code line in lambda function:

```

message = {"Subject": {"Data": subject}, "Body": {"Html": {"Data": body}}}
response = client.send_email(Source = "tharun_nadasu@student.unl.edu", Destination = {"ToAddresses": ["tharun_nadasu@student.unl.edu"]})

```

After my email is successfully verified, I have tested my trigger function by deleting a file in the s3 bucket and it caused the trigger and I received an email for my delete event on s3 bucket.

## Amazon Web Services – Email Address Verification Request in region US East (N. Virginia)



Amazon Web Services <no-reply-aws@amazon.com>  
To: Madasu, Tharun

Reply

Reply All

Dear Amazon Web Services Customer,

We have received a request to authorize this email address for use with Amazon SES and Amazon Pinpoint in region US East (N. Virginia). If you requested this verification, please go to the link below to confirm that you are authorized to use this email address:

[https://urldefense.com/v3/\\_https://email-verification.us-east-1.amazonaws.com/?Context=828293773920&X-Amz-Date=20201206T225418Z&Identity.IdentityName=tharun\\_madasu\\*40student.uml.edu&X-Amz-Algorithm=AWS4-HMAC-SHA256&Identity.IdentityType=EmailAddress&X-Amz-SignedHeaders=Content-Type&Credential=AKIAJR7UYJEP5GNMLX6A\\*2F20201206\\*2Fus-east-1\\*2Fses\\*2Faws4\\_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=7e3449b0656b1994699835d8d239f1f2606e0072999b75c6aef33442e6f6c34d\\_\\_JSUIJSU!!lrdRII43zQIKI\\_c9VySM568J6WL4Ee1O0S88uTyJw7fesuwDCrC3iObZ6uwK1Vxv2la7R](https://urldefense.com/v3/_https://email-verification.us-east-1.amazonaws.com/?Context=828293773920&X-Amz-Date=20201206T225418Z&Identity.IdentityName=tharun_madasu*40student.uml.edu&X-Amz-Algorithm=AWS4-HMAC-SHA256&Identity.IdentityType=EmailAddress&X-Amz-SignedHeaders=Content-Type&Credential=AKIAJR7UYJEP5GNMLX6A*2F20201206*2Fus-east-1*2Fses*2Faws4_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=7e3449b0656b1994699835d8d239f1f2606e0072999b75c6aef33442e6f6c34d__JSUIJSU!!lrdRII43zQIKI_c9VySM568J6WL4Ee1O0S88uTyJw7fesuwDCrC3iObZ6uwK1Vxv2la7R)

Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original verification request.

If you did NOT request to verify this email address, do not click on the link. Please note that many times, the situation isn't a phishing attempt, but either a misunderstanding of how to use someone setting up email-sending capabilities on your behalf as part of a legitimate service, but without having fully communicated the procedure first. If you are still concerned, please notify us at [aws-email-domain-verification@amazon.com](mailto:aws-email-domain-verification@amazon.com) and let us know in the forward that you did not request the verification.

## ObjectCreated:PutEvent from grocerspot-images



tharun\_madasu@student.uml.edu  
To: Madasu, Tharun

Reply

Reply All

This email is to notify you regarding ObjectCreated:Put event. The object Groceries.jpeg is deleted. Source IP: 173.48.192.8

I have also used cloudwatch logs to debug my lambda function code by clicking on 'view logs in cloudwatch'. I could find out the error in my code which caused no email trigger at first.

Lambda > Functions > Lambda\_SES\_event

ARN: arn:aws:lambda:us-east-1:828293773920:function:Lambda\_SES\_event

Lambda\_SES\_event

Throttle Qualifiers Actions Select a test event Test

Configuration Permissions Monitoring

CloudWatch metrics

View traces in X-Ray View logs in CloudWatch View Lambda Insights

The metrics shown are for the unqualified function only. To view metrics for a specific function version or alias, choose a qualifier.

Add to dashboard 1h 3h 12h 1d 3d 1w custom

Invocations	Duration	Error count and success rate (%)
1	1	1 100

```
2020-12-06T18:47:55.230-05:00 START RequestId: 51d47d42-ebe5-489d-b8eb-f375f74d5e48 Version: $LATEST
2020-12-06T18:47:55.439-05:00 [ERROR] ClientError: An error occurred (InvalidParameterValue) when calling the SendEmail operation: Missing final 'domain'
[ERROR] ClientError: An error occurred (InvalidParameterValue) when calling the SendEmail operation: Missing final 'domain'
Traceback (most recent call last):
  File "/var/task/lambda_function.py", line 25, in lambda_handler
    response = client.send_email(Source = "sender-email", Destination =
    {
      "ToAddresses": [
        "tharun_madasu@student.uml.edu"
      ]
    },
    Message = message)
  File "/var/runtime/botocore/client.py", line 357, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "/var/runtime/botocore/client.py", line 676, in _make_api_call
    raise error_class(parsed_response, operation_name)
No newer events at this moment. Auto retry paused. Resume
```

## Elastic IP:

Usually, each EC2 instance is associated with a unique public ipv4 to access/connect over the internet and a private ipv4 address is used to connect within the home/private network and it is used to ssh into ec2 instances. Furthermore, the associated public address on the EC2 instance changes whenever you restart the instance. So, if you intend to launch a website on an instance make sure that you never restart/reboot the instance or use elastic IP. So, the Elastic IP address provides a static IPv4 address for the instance which can mask the failure or software by assigning the address to another instance. Since there is no Elastic IP associated with the EC2 instance by default, considering the above-mentioned reasons I decided to assign an elastic IP to my EC2 instance.

Instance summary for i-06198e04e49f5d0ef (BATMAN) info

Updated less than a minute ago

Connect

Instance state ▼

Instance ID

i-06198e04e49f5d0ef (BATMAN)

Public IPv4 address

54.163.210.130 | open address

Private IPv4 addresses

172.31.26.57

Instance state

Running

Public IPv4 DNS

ec2-54-163-210-130.compute-1.amazonaws.com | open address

Private IPv4 DNS

ip-172-31-26-57.ec2.internal

Instance type

t2.micro

Elastic IP addresses

~

VPC ID

vpc-59042a23

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role

S3forWP

Subnet ID

subnet-e618baab

Elastic IP addresses

Filter Elastic IP addresses

1

⌕

Name	Allocated IPv4 add...	Type	Allocation ID	Associated in
No Elastic IP addresses four				

Elastic IP address settings

Network Border Group

A Network Border Group is a logical group of Zones from where public IPv4 addresses are advertised. Set this parameter to limit the IPv4 address to the Zones in Network Border Group.

us-east-1

Public IPv4 address pool

Public IP addresses are allocated from Amazon's pool of public IP addresses, from a pool that you own and bring to your account, or from a pool that you own and continue to advertise.

Amazon's pool of IPv4 addresses

Public IPv4 address that you bring to your AWS account(option disabled because no pools found) | Learn more

Customer owned pool of IPv4 addresses(option disabled because no customer owned pools found) | Learn more

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. | Learn more

Create accelerator

Cancel

Allocate

From the EC2 dashboard, from the list of available options, I found the 'Elastic IPs' in Network and Security group. From the following page, I chose "Allocate Elastic IP address" and since my EC2 instance was in us-east-1, I selected the network border group to be the same. I like my public IPv4 address to be assigned from Amazon's address pool of IPv4 addresses and selected allocate.

I was redirected to the main page of Elastic IP and I could see that my elastic IP address was allocated successfully. So to assign this allocated address to my EC2 instance, under the actions menu I chose the option 'associate Elastic IP Address'. Under the resource type, I selected Instance and from the list of instances, I have selected 'BATMAN' since my website was launched on this instance. Finally, the Elastic IP was successfully associated with my instance and I could also see my public and my elastic IP address for my instance are now the same.

Elastic IP address: 54.81.213.82

Resource type

Choose the type of resource with which to associate the Elastic IP address.

Instance

Network Interface

⚠ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. | Learn more

Instance

Choose an Instance

i-06198e04e49f5d0ef (BATMAN) - running

i-06512c36aa1530c37 (FLASH) - stopped

i-0fa3e1c8ee604b990 (SUPERMAN) - running

Reassociation

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

Allow this Elastic IP address to be reassociated

Cancel

Associate

Instance summary for i-06198e04e49f5d0ef (BATMAN) info

Updated less than a minute ago

Connect

Instance state ▼

Instance ID

i-06198e04e49f5d0ef (BATMAN)

Public IPv4 address

54.81.213.82 | open address

Private IPv4 addresses

172.31.26.57

Instance state

Running

Public IPv4 DNS

ec2-54-81-213-82.compute-1.amazonaws.com | open address

Private IPv4 DNS

ip-172-31-26-57.ec2.internal

Instance type

t2.micro

Elastic IP addresses

54.81.213.82 [Public IP]

VPC ID

vpc-59042a23

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role

S3forWP

Subnet ID

subnet-e618baab

Not secure | grocerspot.de

Apps

Bookmarks

Outlook.com - thar...

Justice League Arch...

Technology

Python Excel Tutor...

Welcome to Python...

Kairos | Focusing th...

Other bookmarks

Getting Groceries Just Got Simple...

Sample Page

Search

GROCPERSPOT

By Tharun Madasu | December 6, 2020 | No Comments

We deliver farm fresh fruits and vegetables to our customers in and around the Greater Boston Area.

MERRY CHRISTMAS AND HAPPY NEW YEAR...!!!!

Type here to search

12:26 AM 12/10/2020

CloudWatch monitors the AWS resources and the applications that we run in AWS in real-time which basically helps us to gain wide visibility into the application performance, operational health, and resource utilization.

Select metric

Untitled graph

1h 3h 12h 1d 3d 1w 1y

1

0.5

0

03:3003:4504:0004:1504:3004:4505:0005:1505:3005:45

Your CloudWatch graph is empty.  
Select some metrics to appear here.

All metrics

Graphed metrics

Graph options

Source

EC2

113 Metrics

Lambda

16 Metrics

Logs

4 Metrics

113 Metrics

By Image (AMI) Id

Per-Instance Metrics

Aggregated by Instance Type

<input type="checkbox"/>	BATMAN	i-0813804e-49f5d0ef	CPUUsageCreditsCharged
<input type="checkbox"/>	BATMAN	i-0813804e-49f5d0ef	NetworkPacketsIn
<input type="checkbox"/>	BATMAN	i-0813804e-49f5d0ef	NetworkPacketsOut
<input type="checkbox"/>	BATMAN	i-0813804e-49f5d0ef	CPUUtilization
<input type="checkbox"/>	BATMAN	i-0813804e-49f5d0ef	NetworkIn
<input type="checkbox"/>	BATMAN	i-0813804e-49f5d0ef	NetworkOut

Period

5 minutes

Conditions

Threshold type

☒ Static

Use a value as a threshold

☐ Anomaly detection

Use a band as a threshold

Whenever CPUUtilization is...

Define the alarm condition.

☐ Greater

> threshold

☒ Greater/Equal

>= threshold

☐ Lower/Equal

<= threshold

☐ Lower

< threshold

than...

Define the threshold value.

5

Must be a number

Additional configuration

So, if my threshold case becomes TRUE then the alarm is triggered and I would be receiving a notification email specifying the same. In order to achieve the threshold, I had accessed my website from various devices simultaneously and I was lucky enough to achieve the threshold and my CPU utilization on instance reached beyond the threshold and the was alarm which initiated the SNS topic and sent an email to my verified email address.



Name CPU_Utilization_For_Instance_Batman	State OK	Namespace AWS/EC2	Datapoints to alarm 1 out of 1
Type Metric alarm	Threshold CPUUtilization >= 5 for 1 datapoints within 5 minutes	Metric name CPUUtilization	Missing data treatment Treat missing data as missing
Description Your BATMAN instance has exceeded your CPU Threshold	Last change 2020-12-07 02:18:02	InstanceId i-06198e04e49f5d0ef	Percentiles with low samples evaluate
		Instance name BATMAN	ARN arn:aws:cloudwatch:us-east-1:828293773920:alarm:CPU_Utilization_For_Instance_Batman
		Statistic Average	
		Period 5 minutes	

ALARM: "CPU\_Utilization\_For\_Instance\_Batman" in US East (N. Virginia)

AWS Notifications <no-reply@sns.amazonaws.com>  
To: Madasu, Tharun

You forwarded this message on 12/6/2020 9:00 PM

You are receiving this email because your Amazon CloudWatch Alarm "CPU\_Utilization\_For\_Instance\_Batman" in the US East (N. Virginia) region has entered the ALARM state, because "out of the last 1 datapoints [5.9999999999999997 (07/12/20 02:07:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Monday 02:12:02 UTC".

View this alarm in the AWS Management Console:  
[https://urldefense.com/v3/\\_https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarms&alarm=CPU\\_Utilization\\_For\\_Instance\\_Batman\\_\\_\\_jw!lr0R43e2NnAp3URtn9aVl\\_28rL2pxRr\\_88IQ867UUDuTO13ca6S2\\_1XG68cmg5KLOhm2PHFFdmg5](https://urldefense.com/v3/_https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarms&alarm=CPU_Utilization_For_Instance_Batman___jw!lr0R43e2NnAp3URtn9aVl_28rL2pxRr_88IQ867UUDuTO13ca6S2_1XG68cmg5KLOhm2PHFFdmg5)

Alarm Details:

- Name: CPU\_Utilization\_For\_Instance\_Batman
- Description: Your BATMAN instance has exceeded your CPU Threshold
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [5.9999999999999997 (07/12/20 02:07:00)] was greater than or equal to the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Monday 07 December, 2020 02:12:02 UTC
- AWS Account: 828293773920
- Alarm Arn: arn:aws:cloudwatch:us-east-1:828293773920:alarm:CPU\_Utilization\_For\_Instance\_Batman

Threshold:  
The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 5.0 for 300 seconds.

**Billing Alarm** – I wanted to keep track of my available credits hence I chose to create a billing alarm for every \$1. Following the same procedure as in creating a CPU utilization alarm, I followed similar steps to create a billing alarm. So firstly, I had to create a new topic as 'Billing\_Alarm' and had given my email id to get the alarm notification whenever the threshold is crossed. I did get an email to confirm my subscription for this specific alarm.

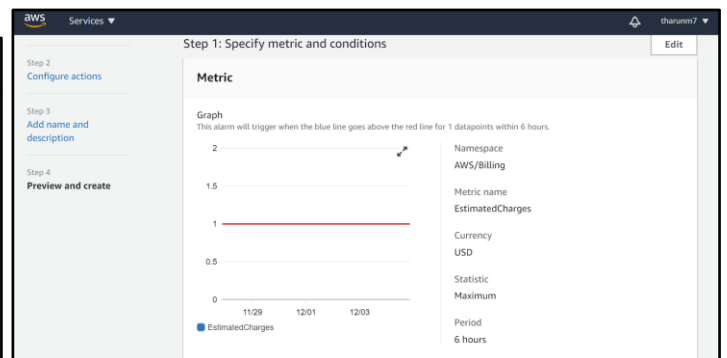
amazon web services Simple Notification Service

**Subscription confirmed!**

You have subscribed tharun\_madasu@student.uml.edu to the topic: **Billing\_Alarm**.

Your subscription's id is:  
arn:aws:sns:us-east-1:828293773920:Billing\_Alarm:b9f1ba43-deb7-4711-9459-652efe0ffc3d

If it was not your intention to subscribe, [click here to unsubscribe](#).



As part of my project, I had to use Route 53 to which I had incurred a charge of \$1.12 for a period of 3 weeks which was relatively economical. Therefore, as soon as I created the billing alarm, I got an immediate email notification that my threshold has been crossed. Now that my billing alarm is functioning perfectly fine I've updated the threshold with a slightly higher value.

ALARM: "Billing\_Alarm" in US East (N. Virginia)

AWS Notifications <no-reply@sns.amazonaws.com>  
To: Madasu, Tharun

You are receiving this email because your estimated charges are greater than the limit you set for the alarm "Billing\_Alarm" in AWS Account 828293773920.

The alarm limit you set was \$ 1.00 USD. Your total estimated charges accrued for this billing period are currently \$ 1.12 USD as of Saturday 05 December, 2020 02:35:24 UTC. The actual charges you will be billed in this statement period may differ from the charges shown on this notification. For more information, view your estimated bill at: [https://urldefense.com/v3/\\_https://us-east-1.console.aws.amazon.com/billing/home\\*/bill?year=2020&month=12\\_\\_\\_jw!lr0R43e2NnAp3URtn9aVl\\_28rL2pxRr\\_88IQ867UUDuTO13ca6S2\\_1XG68cmg5KLOhm2PHFFdmg5](https://urldefense.com/v3/_https://us-east-1.console.aws.amazon.com/billing/home*/bill?year=2020&month=12___jw!lr0R43e2NnAp3URtn9aVl_28rL2pxRr_88IQ867UUDuTO13ca6S2_1XG68cmg5KLOhm2PHFFdmg5)

than... Define the threshold value.

50 USD

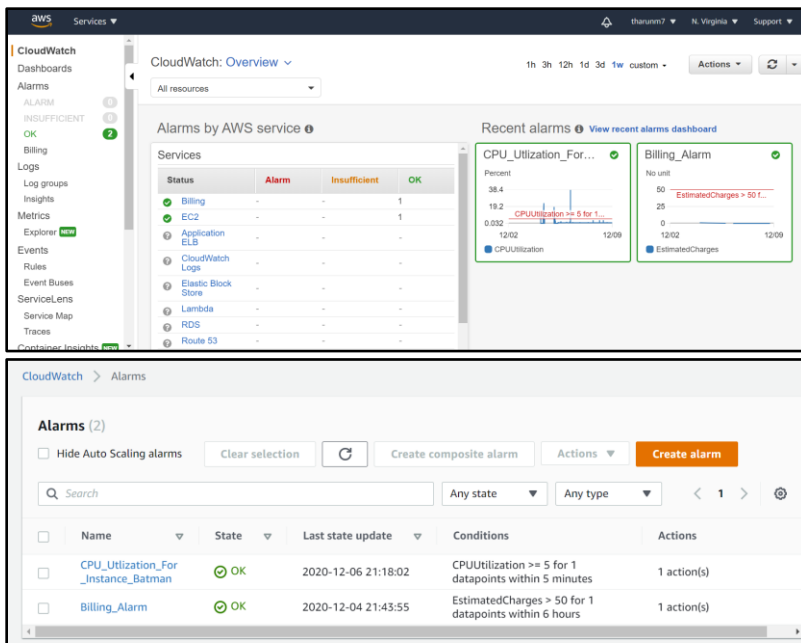
Must be a number

► Additional configuration

Cancel Next Update alarm

Currently, this is how my CloudWatch dashboard looks like –





## Conclusion:

First things first, it was a great learning and hands-on experience. The main goal of this project has been achieved. I got loads of help from AWS docs and the internet as well. Nonetheless, my lab exercise and weekly reading assignments did help me in getting things started with the project. There were a plethora of services available in the free tier, so designing, developing, and implementing these services in my project with AWS UI was a phenomenally satisfying hands-on experience in this course. I felt AWS has a lot of services to offer when compared to Azure. Nonetheless, I thoroughly enjoyed brainstorming the issues that I faced during working on these projects. Coming to my future work in my AWS project, which is to use my back-up instance (FLASH) and deploy my website and serve user requests when my current operational instance (BATMAN) has a failover, i.e. FLASH strikes if BATMAN fails.