|  |
| --- |
| **PROJECT** |
| **WEB APPLICATION USING ELASTIC BEANSTALK AND DYNAMO DB** |
|  |
|  |
|  |
|  |
|  |
|  |

**Web Application using Elastic Beanstalk & Dynamo DB**

**Aim**

This project aims at creating a simple web application using Elastic Beanstalk and Dynamo DB

**PREREQUISITES.**

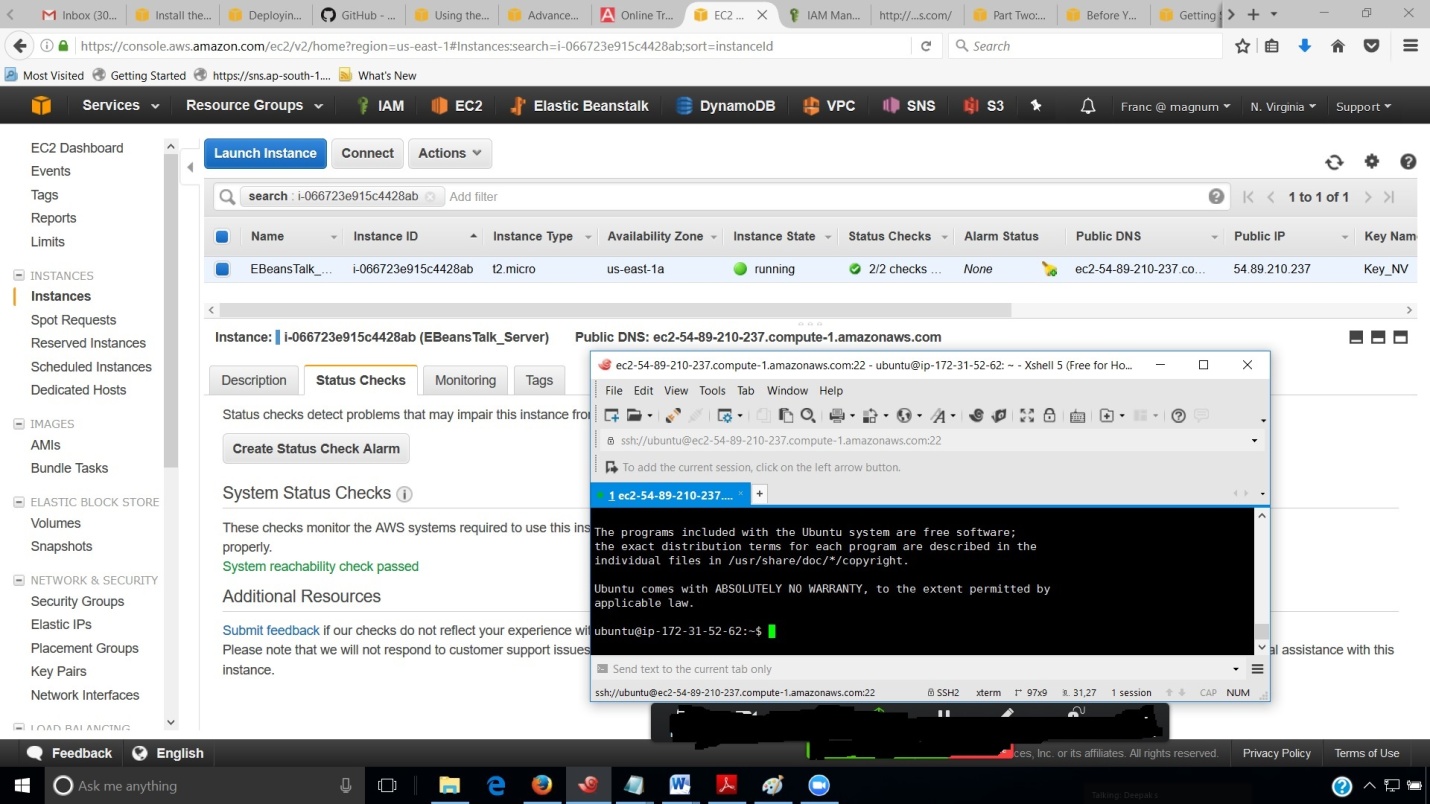
Deploy an application in elastic bean stalk

Create a table using DynamoDB

**STEPS**

1. Launch Ubuntu Instance 14.04 LTS -amd64-server-20160714

(ami-d732f0b7)



ssh [ubuntu@ec2-54-158-76-97.compute-1.amazonaws.com](mailto:ubuntu@ec2-54-158-76-97.compute-1.amazonaws.com)

ssh into the machine by Xshell5 and type the following commands below

**Install the Elastic Beanstalk Command Line Interface (EB CLI) along with Sample Application**

var/lib/apache2/site

1. sudo apt-get update -y

2. sudo apt-get upgrade -y

3. sudo apt-get install python-pip

4. pip install --upgrade --user awsebcli

5. cd ~/.local/bin/

6. ls -a ~

7. export PATH=~/.local/bin:$PATH

8. source ~/.profile

9. eb --version

10. pip install --upgrade --user awsebcli

11. cd

11. cd /home/ubuntu/

12. ls /// To check pip it will show get-pip.py

13. sudo apt-get install git

14. git clone https://github.com/awslabs/eb-node-express-sample.git

15. ls /// you can see the sample git app listed.

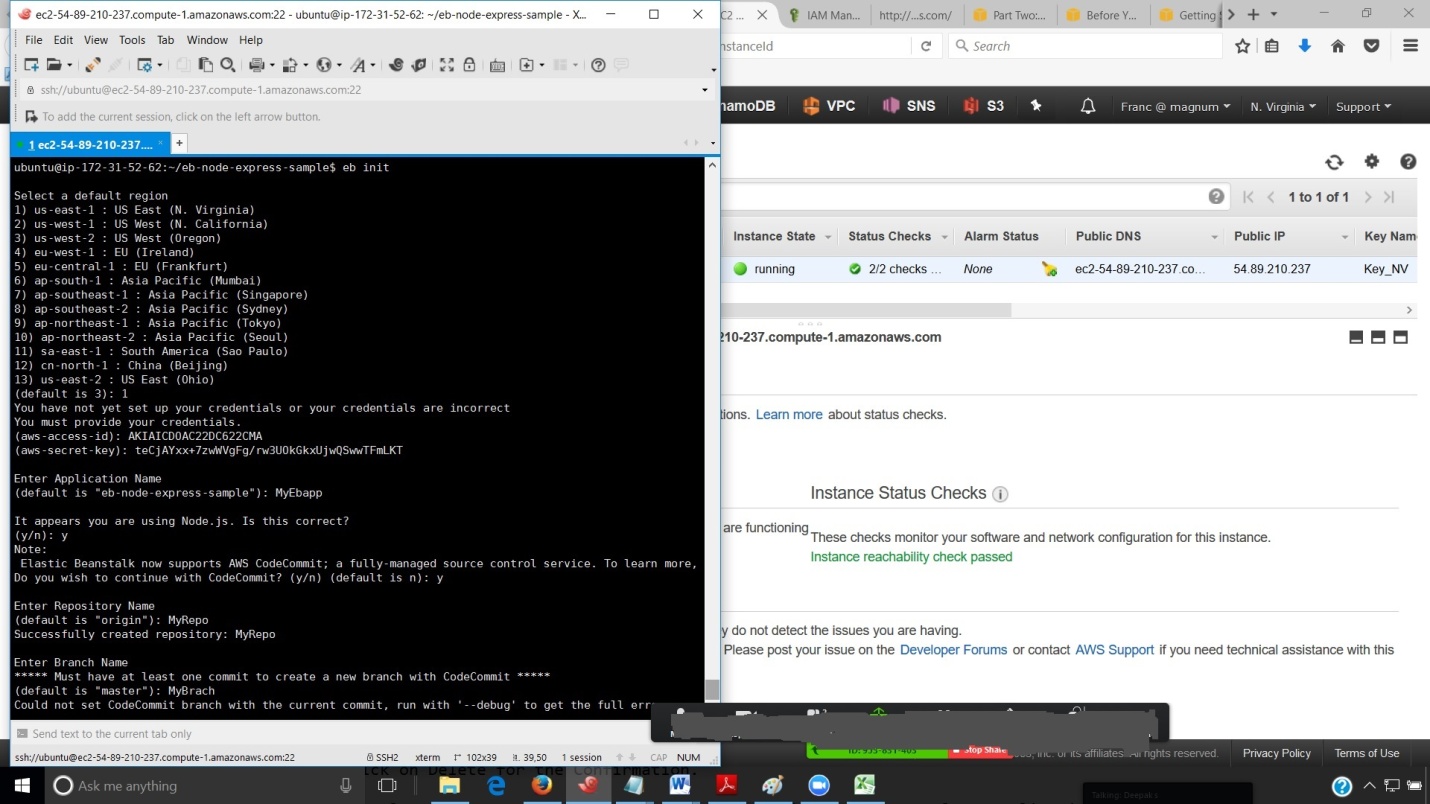
17. cd eb-node-express-sample/

18. ls

19. eb --version /// eb version is 3.8.1( Python 3.4.3 )

20. eb init

21. Select Region



22. Provide your Credentials ( Access ID and Secret Key)

23. Select Default (2) for creating Application and give an application name.

24. Give yes for node.js

25. Give yes setup SSH for instance

26. Select your Keypair

/// eb create --help

27. eb create signup-staging -i t2.micro -c signup-staging-launch //// ( Launches in the Producation. Must Delete immediately after use. )

/// when yoiu fire this command it will show first upload complete if success.

/// Then create environment is starting.

/// will also show name of Load balancer.

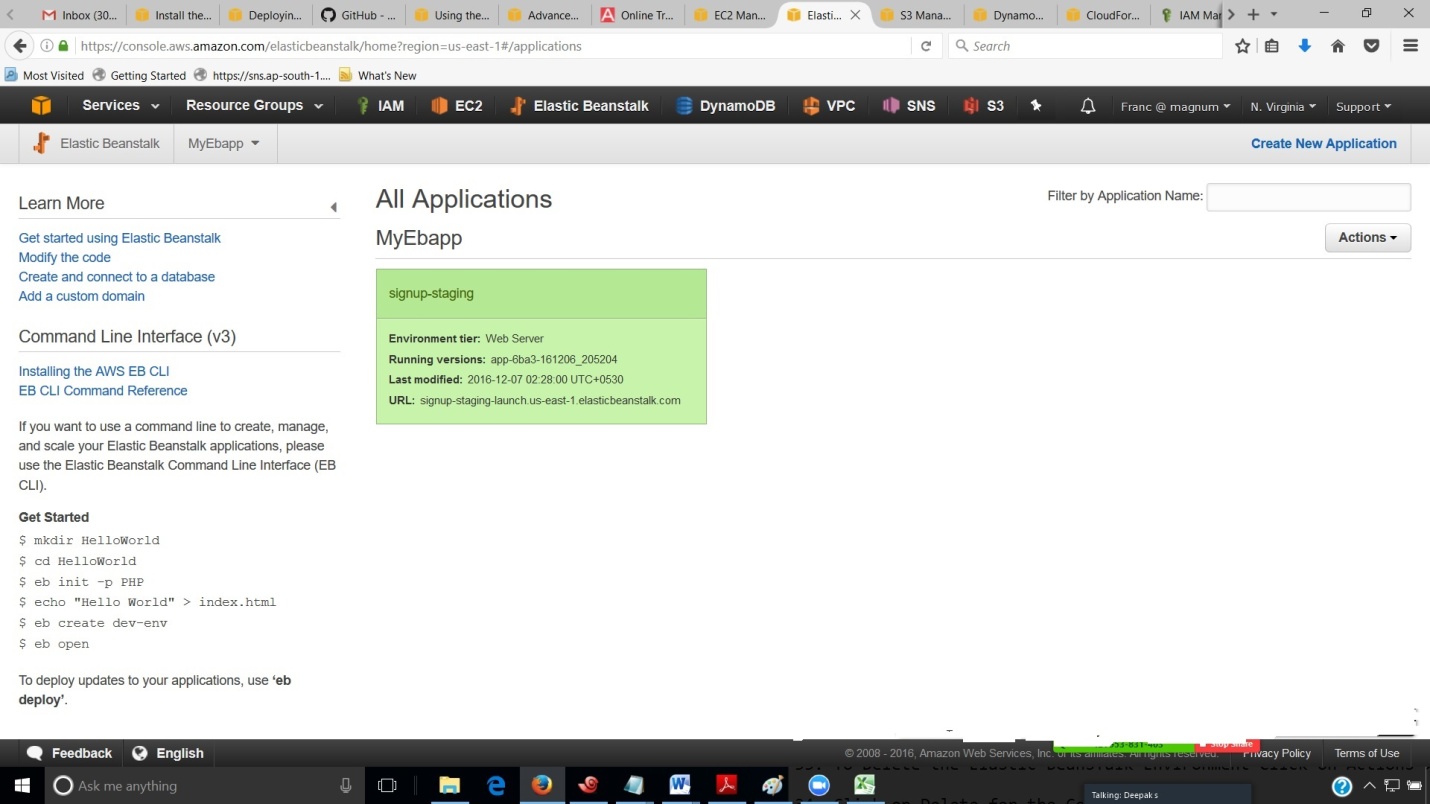
28. go to Elastic BeansTalk Dashboard. It will show signup-staging.

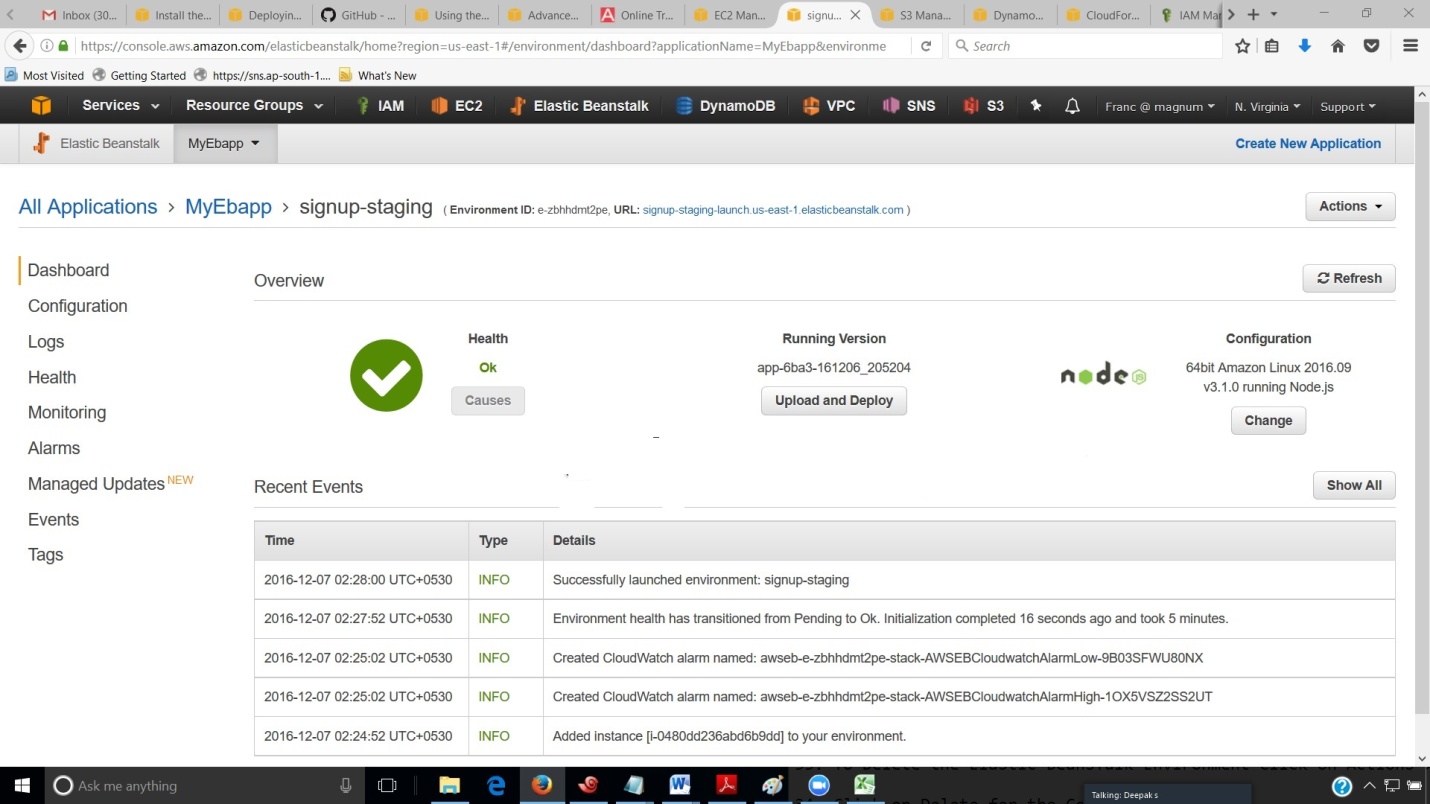
29. After sometime you can see in the Xshell that

" Successfully launched the environment: signup-staging."

30. Go to Elastic BeansTalk Dashboard and Refresh the URL.

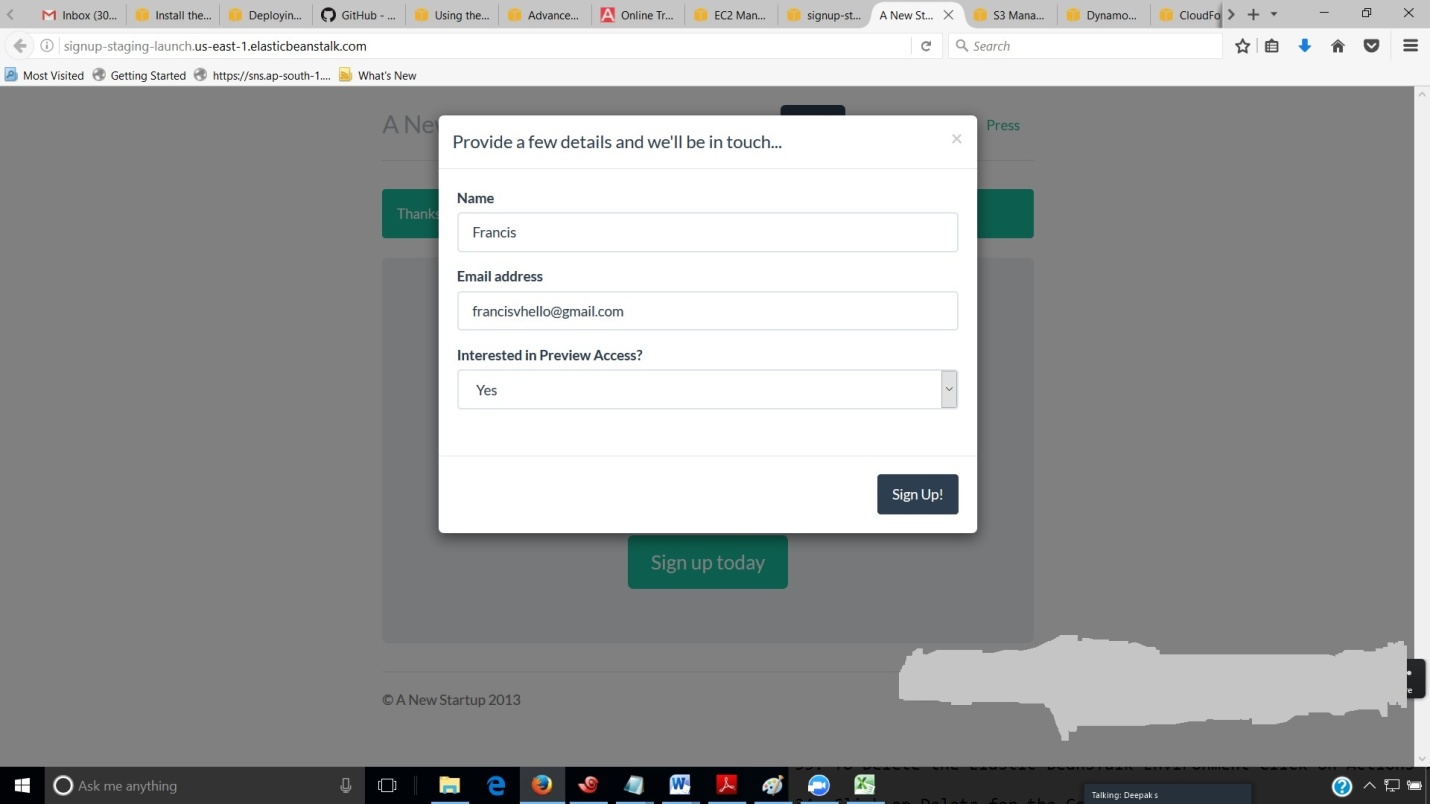
The color of the box Signup-staging will change from gray to green if it is available status

.



31. Click on the URL path given on the box Signup-staging and you can see the details of it.

32. Copy the URL at the Top ( Environment ID and open in the browser.) to get the sample application as below



32. Right now, the Amazon SNS topic is configured with a placeholder email for notifications. You will update the configuration soon, but in the meantime you can verify the DynamoDB table and Amazon SNS topic in the AWS Management Console.

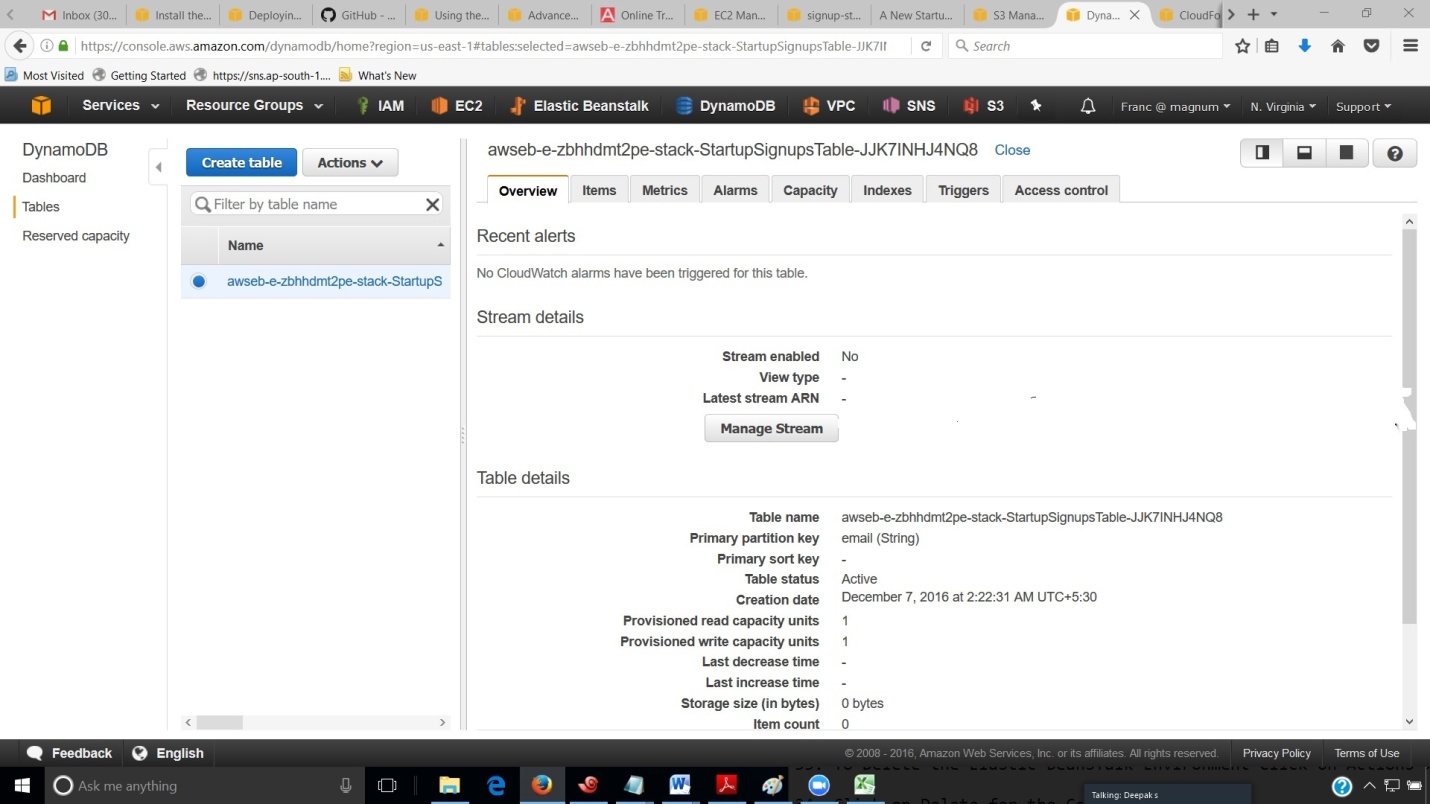
**To view the table**

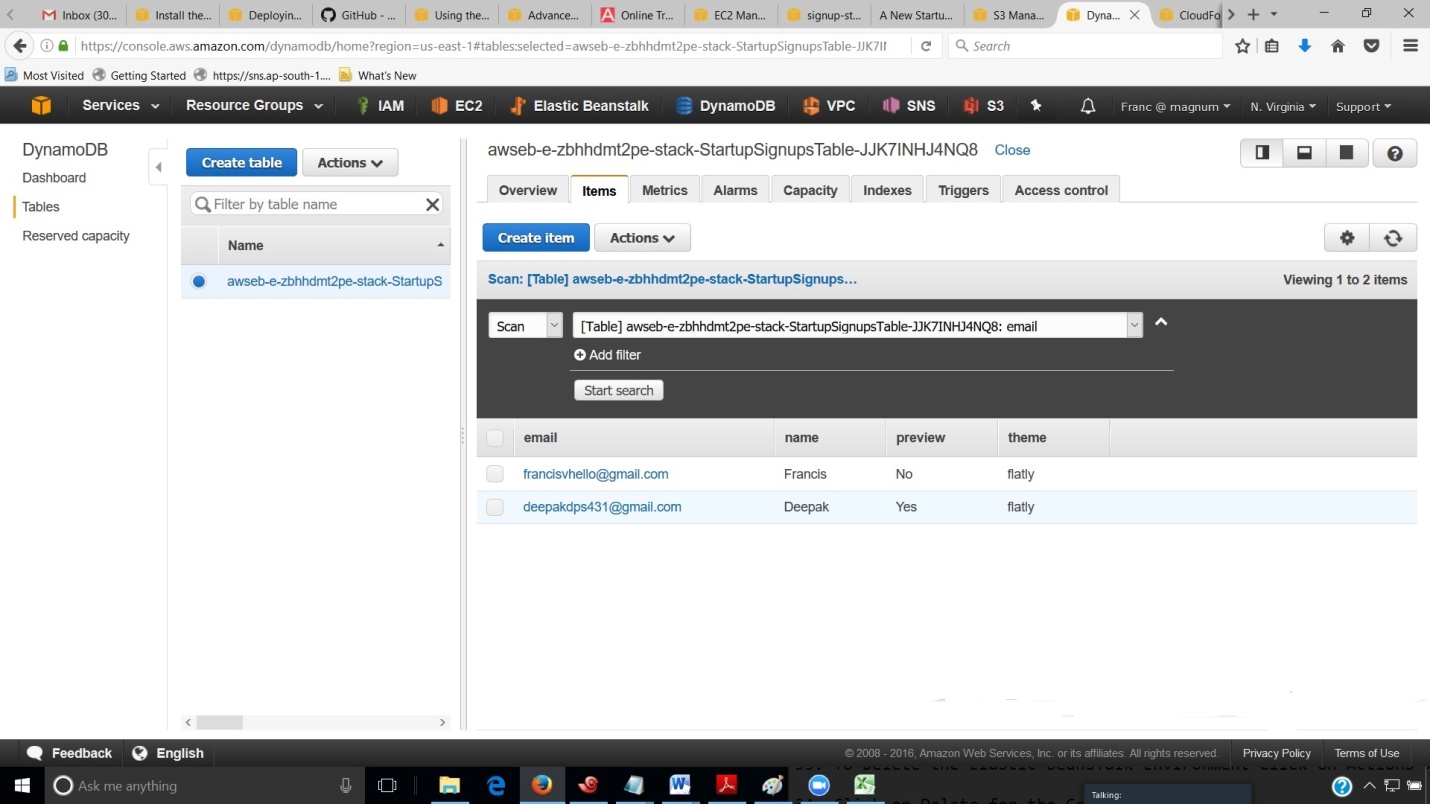
1.Open the Tables page in the DynamoDB console.

2.Find the table that the application created. The name starts with awseb and contains StartupSignupsTable.

3.Select the table, choose Items, and then choose Start search to view all items in the table.

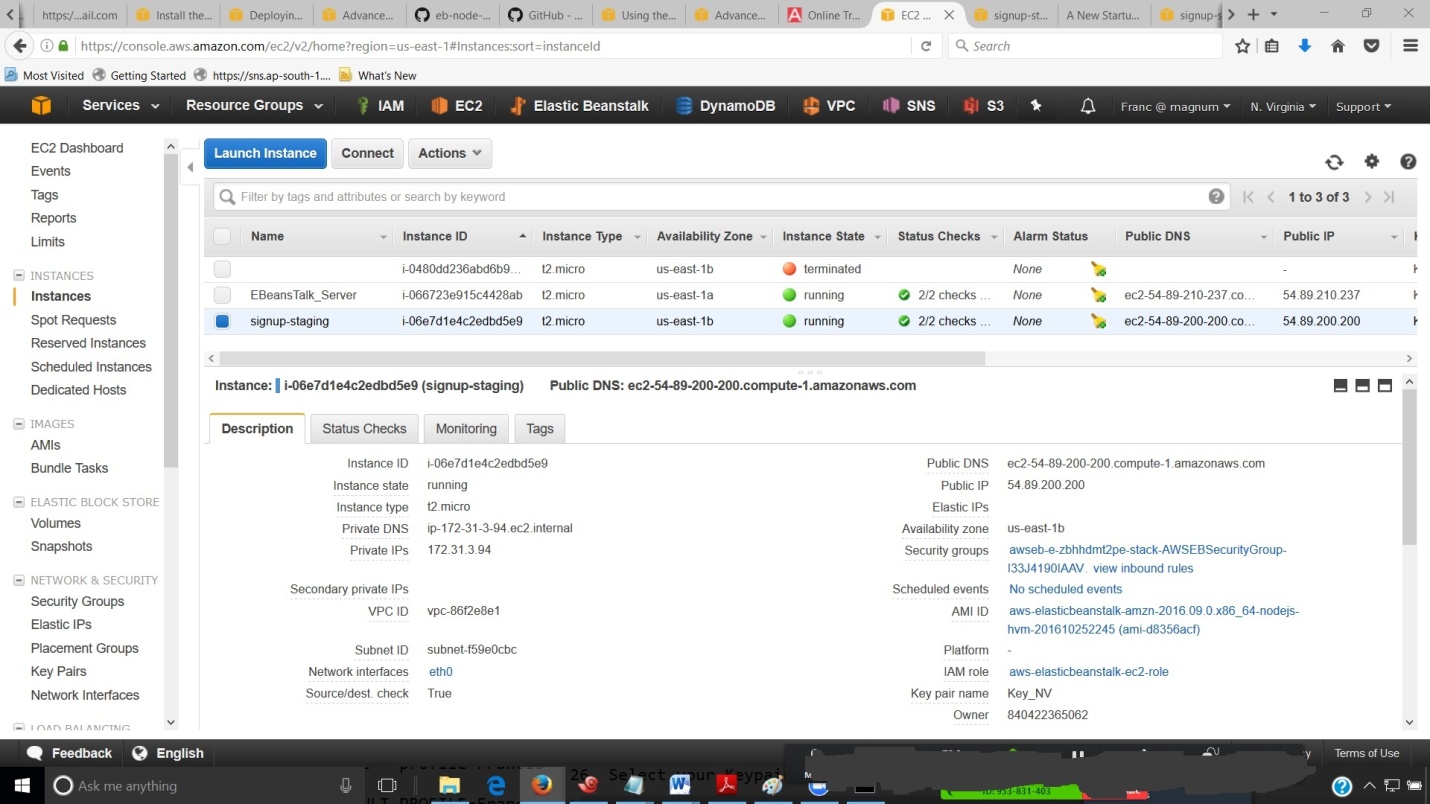
The table contains an entry for every email address submitted on the signup site. In addition to writing to the table, the application sends a message to an Amazon SNS topic that has two subscriptions, one for email notifications to you, and another for an Amazon Simple Queue Service queue that a worker application can read from to process requests and send emails to interested customers.





**Elastic Beanstalk takes about five minutes to create the environment with the following resources:**

EC2 instance – An Amazon Elastic Compute Cloud (Amazon EC2) virtual machine configured to run web apps on the platform that you choose.



Each platform runs a different set of software, configuration files, and scripts to support a specific language version, framework, web container, or combination thereof. Most platforms use either Apache or nginx as a reverse proxy that sits in front of your web app, forwards requests to it, serves static assets, and generates access and error logs.

Instance security group – An Amazon EC2 security group configured to allow ingress on port 80. This resource lets HTTP traffic from the load balancer reach the EC2 instance running your web app. By default, traffic is not allowed on other ports.

Load balancer – An Elastic Load Balancing load balancer configured to distribute requests to the instances running your application. A load balancer also eliminates the need to expose your instances directly to the Internet.

Load balancer security group – An Amazon EC2 security group configured to allow ingress on port 80. This resource lets HTTP traffic from the Internet reach the load balancer. By default, traffic is not allowed on other ports.

Auto Scaling group – An Auto Scaling group configured to replace an instance if it is terminated or becomes unavailable.

Amazon S3 bucket – A storage location for your source code, logs, and other artifacts that are created when you use Elastic Beanstalk.

Amazon CloudWatch alarms – Two Amazon CloudWatch alarms that monitor the load on the instances in your environment and are triggered if the load is too high or too low. When an alarm is triggered, your Auto Scaling group scales up or down in response.

AWS CloudFormation stack – Elastic Beanstalk uses AWS CloudFormation to launch the resources in your environment and propagate configuration changes. The resources are defined in a template that you can view in the AWS CloudFormation management console.

Domain name – A domain name that routes to your web app in the form subdomain.region.elasticbeanstalk.com

**RESULT.**

**Deployed a sample application in elastic beanstalk and connected it to Dynamo DB in baground**