



www.linkedin.com/in/prafulpatel16



<https://github.com/prafulpatel16>

Page | 0



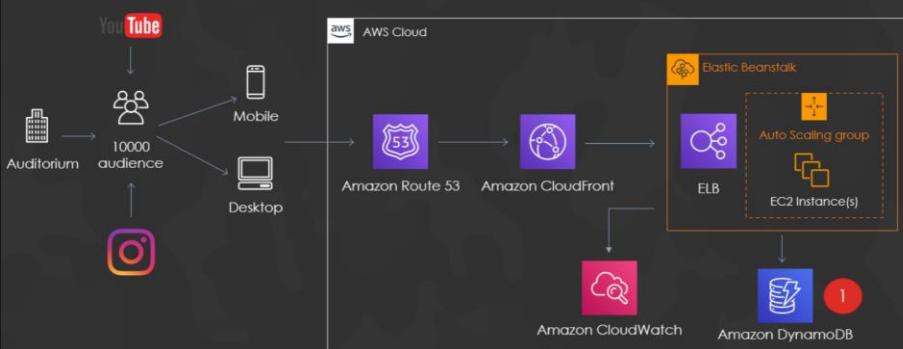
AWS BOOTCAMP PROJECT

PROJECT 3: DEPLOYMENT OF SCALABLE

APPLICATION

IMPLEMENTED BY: PRAFUL PATEL

SCALABLE APPLICATION: THE CLOUD BOOTCAMP CONFERENCE



PRAFUL PATEL

www.linkedin.com/in/prafulpatel16

<https://github.com/prafulpatel16>

At the end of the slide, there is a watermark-like text: 'PROJECT 3: DEPLOYMENT OF SCALABLE APPLICATION'.



Date: Feb 15, 2022

➤ **Project Definition:**

An IT services provider company called **PRAfect Systems Inc.** is engaged into providing software development solutions. Currently, they are going to launch a bootcamp where more than 10000 users may access their web application. So in that case due to heavy load on the web application the application has to auto scale without creating and managing any infrastructure components from scratch.

The plan is to deploy the web application itself without managing any infrastructure headache. The web servers should be able to handle load and database should also be scaled up.

➤ **Project Description:**

Web Servers: Contains python application and packages.

Database Servers: Contains DynamoDB (NoSQL) database.

This project demonstrates an experience of web application deployment through Elastic Beanstalk and store the emails in DynamoDB which is NoSQL scalable database. Also in this project the web application is going to be cached the static and dynamic content using Cloudfront to an Edge locations close to the users.

➤ **Solution:**

This project needs to be deployed on aws using AWS's platform as service called "Elastic Beanstalk" along with public server(EC2 machine) and one db server (dynamoDBdb instance). VPC services should consists more than public subnet and more than one private subnet so that web application machine should be places behind a public server, and behind private server the DB instance should be placed.

For Web server perspective, this server should be from a linux distribution systems. For networking and security perspective, web application server will have to be accessed and allowed publicly available so internet gateway needs to be provisioned and assigned to the public ip. For security concerns and avoiding cyber threats it has to configure some security group and assign and appropriate routes and rules based on requirement. Also web serve should be allowed to access from SSH at port 22 for system administration perspective, and it should allow to be accessible at custom TCP: Port 8080 from everywhere.

For DB server perspective, this needs to be behind a private subnet so it should not allow from external world so certain settings like publicly available should be disabled. Also it has to strictly define some rules from security groups so that no one can access and abuse the db server. DB credentials should be defined while provisioning a db server and pass through safe mechanism.

OS and application packages requirements are provided separately which needs to be provisioned and be ready before go on live session.

➤ **Project Cost Estimation:**

(Note: This cost is Not any actual cost, it's just an estimation based on high level requirement. Price may be vary based on adding and removing services based on requirement.)

Elastic Beanstalk

There is no additional charge for AWS Elastic Beanstalk—you pay only for the AWS resources actually used to store and run your application.

The screenshot shows the AWS Pricing Calculator interface. At the top, there is an 'Estimate summary' section with the following details:

Upfront cost	Monthly cost	Total 12 months cost
15,390.00 USD	3,105.96 USD	52,661.52 USD

A note below states: "Includes upfront cost". To the right, there is a "Getting Started with AWS" sidebar with links to "Contact Us" and "Sign in to the Console".

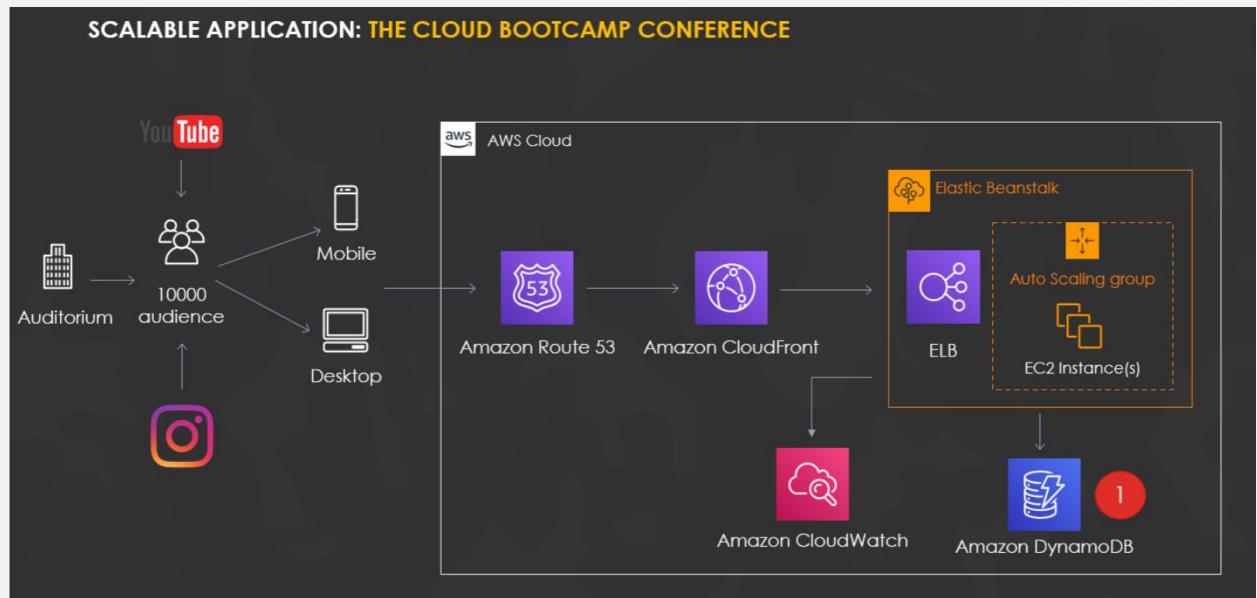
The main content area lists three services with their respective configurations and costs:

- Amazon DynamoDB**: Region: US East (N. Virginia). Description: Table class (Standard), Average item size (all attributes) (100 KB), Write reserved capacity term (1 year), Read reserved capacity term (1 year), Data storage size (5 GB). Monthly: 2,398.01 USD, Upfront: 15,390.00 USD.
- Amazon CloudFront**: Region: US East (N. Virginia). Description: Data transfer out to internet (1024 GB per month), Data transfer out to origin (1024 GB per month), Number of requests (HTTP(S)) (100 per month). Monthly: 107.52 USD.
- Elastic Load Balancing**: Region: US East (N. Virginia). Description: Number of Application Load Balancers (1). Monthly: 600.43 USD.

Tools & Technologies covered:

- AWS public cloud
- Elastic Beanstalk
- Cloudfront
- Auto Scaling
- Cloudwatch
- DynamoDB
- Route 53
- AWS Pricing Calculator
- MobaXterm SSH client

➤ **Architectural Diagram:**



This migration project will be completed in 3 implementation phases.

➤ **Project implementation Phases:**

- ❖ Implementation Phase 1: Create a DynamoDB table and configure Elastic Beanstalk to create an application.
- ❖ Implementation Phase 2: Create Cloudfront distribution.
- ❖ Implementation Phase 3: Perform Load testing and test application auto scale up

➤ **Implementation:**

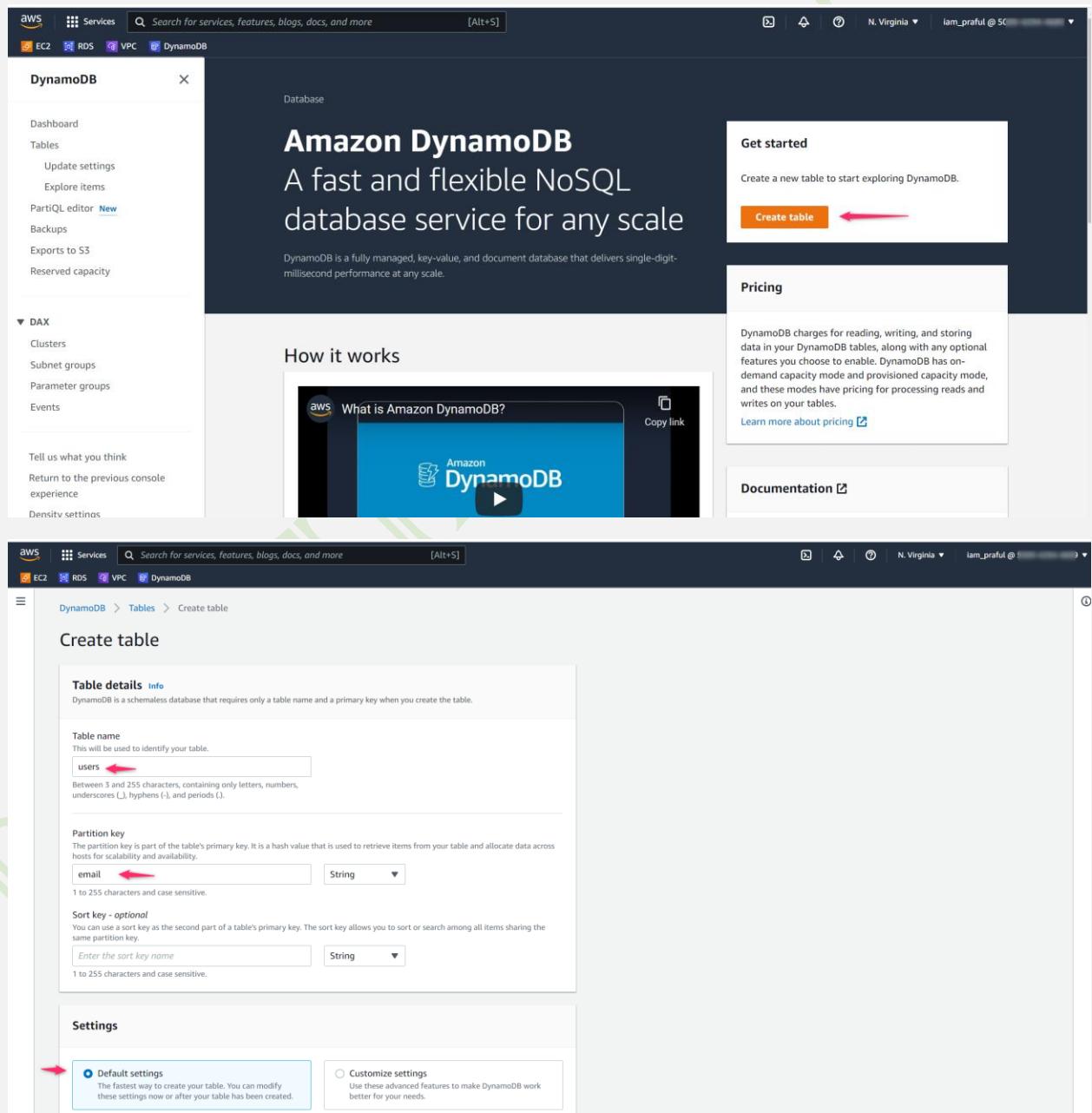
- ❖ **Implementation Phase 1: Create a DynamoDB table and configure Elastic Beanstalk to create an application**
 - ❖ Create DynamoDB table
 - Create a new table
 - Name: users
 - Primary key: email
 - ❖ Configure Elastic Beanstalk
 - ❖ Create application
- ❖ **Implementation Phase 2: Create Cloudfront distribution**
- ❖ **Implementation Phase 3: Perform load testing and verify application auto scale up**

- **Solution execution in Action:**

 **Implementation Phase 1 in Action : Create a DynamoDB table and configure Elastic Beanstalk to create an application**

Create a DynamoDB table

1. Create a table: users , primary key: email



The screenshot shows two consecutive screenshots of the AWS DynamoDB console.

Top Screenshot (Dashboard):

- Header: AWS Services Search bar: Search for services, features, blogs, docs, and more [Alt+S]
- Region: N. Virginia iam_praful @ 5C
- Main Content: **Amazon DynamoDB** A fast and flexible NoSQL database service for any scale. Description: DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.
- Left Sidebar: Dashboard, Tables, Update settings, Explore items, PartiQL editor (New), Backups, Exports to S3, Reserved capacity.
- Bottom Sidebar: Tell us what you think, Return to the previous console experience, Density settings.
- Right Sidebar: Get started (Create table button highlighted with a red arrow), Pricing, Documentation.

Bottom Screenshot (Create table wizard):

- Header: AWS Services Search bar: Search for services, features, blogs, docs, and more [Alt+S]
- Region: N. Virginia iam_praful @ 5C
- Path: DynamoDB > Tables > Create table
- Title: Create table
- Form Fields (highlighted with red arrows):
 - Table details**: Table name: users
 - Partition key**: email (String type)
 - Sort key - optional**: Enter the sort key name (String type)
- Settings:
 - Default settings** (radio button selected)
 - Customize settings**

Table created successfully

The screenshot shows the AWS DynamoDB service dashboard. A green success message at the top right states "The users table was created successfully." The main area displays a table named "Tables (1) Info". The table has one item, "users", with the following details:

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Table class
users	Active	email (\$)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)	DynamoDB Standard

Configure Elastic Beanstalk

The screenshot shows the AWS Elastic Beanstalk service dashboard under the Compute section. The main heading is "Amazon Elastic Beanstalk" and "End-to-end web application management.". On the right, there's a "Get started" box with the text "Easily deploy your web application in minutes." and a prominent orange "Create Application" button, which has a red arrow pointing to it.

Create a WebApp

Elastic Beanstalk

Create a web app

Application information

Application name: tcb-conference

Application tags

Platform

Platform: Python

Platform branch: Python 3.7 running on 64bit Amazon Linux 2

Platform version: 3.3.10 (Recommended)

Application code

Web app source code url

<https://tcb-bootcamps.s3.amazonaws.com/bootcamp-aws/en/tcb-conf-app-EN.zip>

Elastic Beanstalk

Platform

Application code

Sample application

Upload your code

Source code origin

Version label: tcb-conference-source

Source code origin: Public S3 URL

Application code tags

Configure more options

Create application

Click on Configure more options

- Presets:
- Configuration presets: High availability

Elastic Beanstalk > Getting started

Configure Tcbconference-env

Presets

Start from a preset that matches your use case or choose *Custom configuration* to unset recommended values and use the service's default values.

Configuration presets

- Single instance (*Free Tier eligible*)
- Single instance (using Spot instance)
- High availability (High availability (using Spot and On-Demand instances))
- Custom configuration

- Software:

-- Environment properties:

--- AWS_REGION = us-east-1

Elastic Beanstalk >

Platform

Python 3.7 running on 64bit Amazon Linux 2/3.3.10

Change platform version

Software

Amazon X-Ray: disabled	Log streaming: disabled (default)	Environment properties: 1
Rotate logs: disabled (default)		PYTHONPATH

Edit

Elastic Beanstalk >

Log streaming

(Standard CloudWatch charges apply.)

Enabled

Retention

7 days

Lifecycle

Keep logs after terminating environment

Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value
PYTHONPATH	/var/app/venv/staging-LQM1lest/bin
AWS_REGION	us-east-1

Cancel Save

- Instances:

-- Root volume type: General Purpose SSD - 8 GB

The screenshot shows the AWS Elastic Beanstalk Instances configuration page. It displays the following settings:

- IMDSv1: disabled
- Root volume size (GB): container default
- Root volume throughput (MiB/s): container default
- Root volume type: container default
- Root volume IOPS: container default
- Security groups: none

The screenshot shows the AWS Elastic Beanstalk Modify instances configuration page. It displays the following settings:

- Root volume (boot device)**
- Root volume type: General Purpose (SSD) (highlighted with a red arrow)
- Size: 8 GB
- IOPS: 100 (Input/output operations per second for a provisioned IOPS (SSD) volume)
- Throughput: 1 MiB/s (The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance)

- Capacity:

- Instances Min: 2
- Instances Max: 4
- Instance type: t2-micro

-- Scaling triggers:

- CPUUtilization
- Unit: Percent
- Period: 1 Min
- Breach duration: 1 Min
- Upper threshold: 50
- Lower threshold: 40

The screenshot shows the AWS Elastic Beanstalk Capacity configuration page. It displays the following settings:

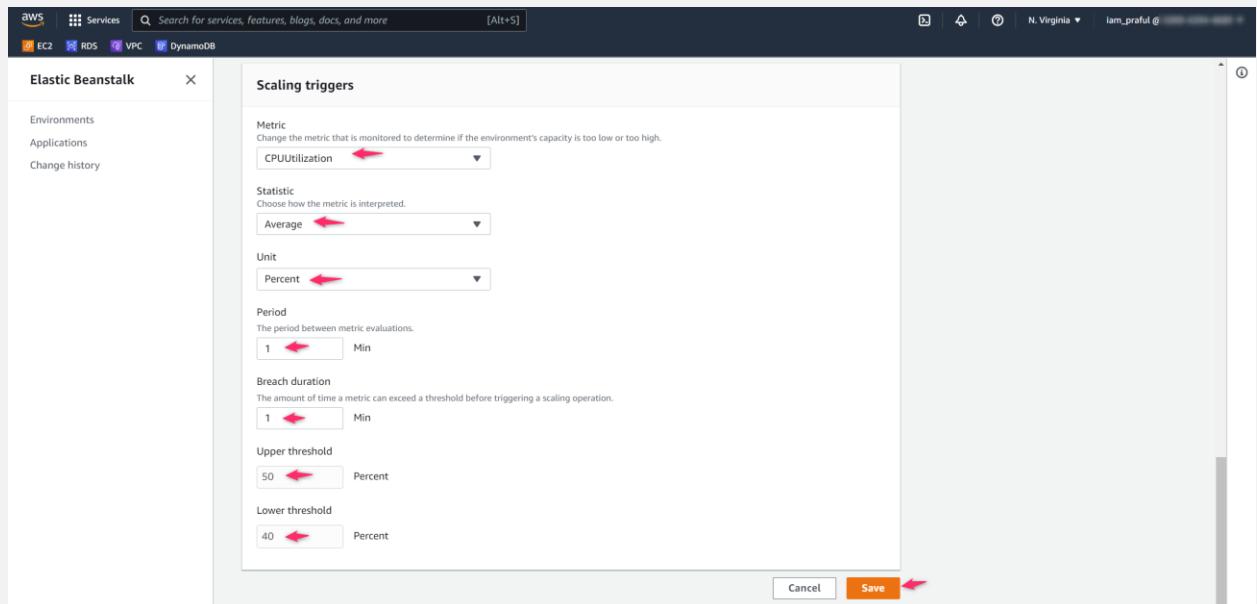
- Environment type: load balancing, auto scaling
- Fleet composition: On-Demand instances
- EC2 image ID: ami-01d52c2dd41814c37
- Availability Zones: Any
- EC2 instance types: t2.micro,t2.small
- Instances: 1-4
- Capacity rebalancing: disabled

The screenshot shows the 'Modify capacity' section of the AWS Elastic Beanstalk console. The 'Auto scaling group' configuration is displayed. Key settings include:

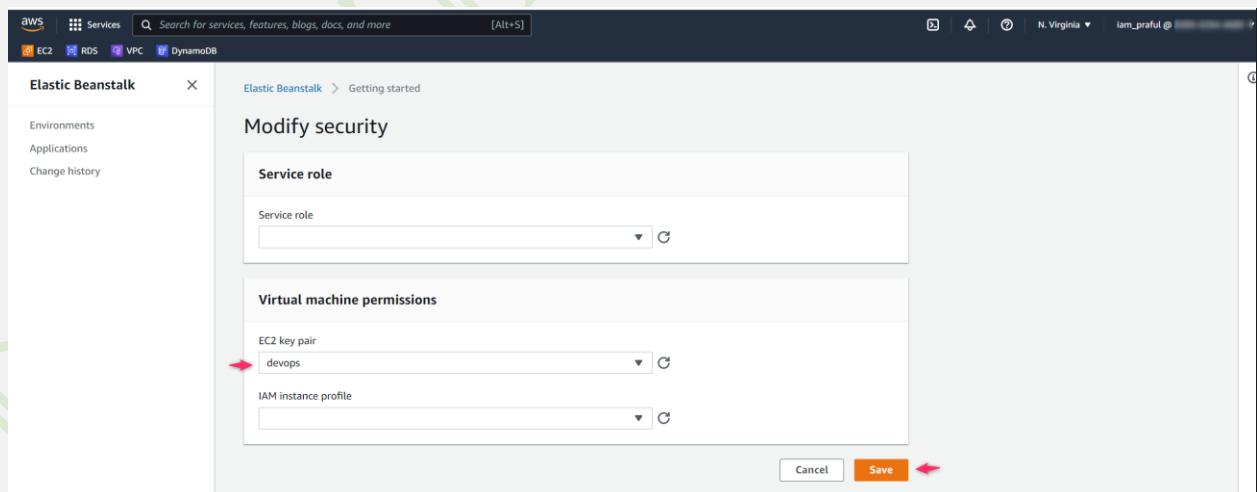
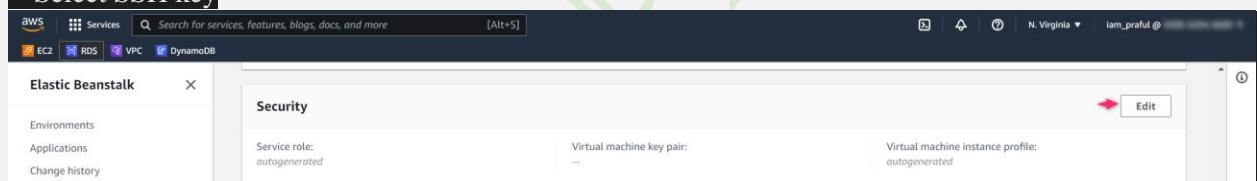
- Environment type:** Load balanced (selected)
- Instances:** Min: 2, Max: 4
- Fleet composition:** On-Demand instances (selected)
- Maximum spot price:** Default - the On-Demand price for each instance type (recommended)
- On-Demand base:** 0

The screenshot shows the 'Modify capacity' section of the AWS Elastic Beanstalk console. The 'Auto scaling group' configuration is displayed. Key settings include:

- On-Demand base:** 0
- On-Demand above base:** 70 %
- Capacity rebalancing:** Turn on capacity rebalancing (disabled)
- Processor:** x86 (selected)
- Instance types:** t2.micro (selected)
- AMI ID:** ami-01d52c2dd41814c37
- Availability Zones:** Any
- Placement:** Choose Availability Zones (AZs) --



- Security:
-- Select SSH key



- Networking:
-- VPC: Default:
-- Add IP Public

-- Select all subnets

aws Services Search for services, features, blogs, docs, and more [Alt+S]

EC2 RDS VPC DynamoDB

Elastic Beanstalk

environments Applications Change history

enabled Wed:19:00 UTC

Notifications

Email address:

Network

This environment is not part of a VPC.

Edit

aws Services Search for services, features, blogs, docs, and more [Alt+S]

EC2 RDS VPC DynamoDB

Elastic Beanstalk > Getting started

Elastic Beanstalk

environments Applications Change history

Modify network

Virtual private cloud (VPC)

VPC Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-24205a59 (172.31.0.0/16) | default vpc (default)

Create custom VPC

Load balancer settings

Assign your load balancer to a subnet in each Availability Zone (AZ) in which your application runs. For a publicly accessible application, set Visibility to Public and choose public subnets.

Visibility

Make your load balancer internal if your application serves requests only from connected VPCs. Public load balancers serve requests from the Internet.

Public

Load balancer subnets

<input checked="" type="checkbox"/> Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> us-east-1a	subnet-f718e6bb	172.31.16.0/20	-
<input checked="" type="checkbox"/> us-east-1b	subnet-a7143ff8	172.31.32.0/20	-
<input checked="" type="checkbox"/> us-east-1c	subnet-23cee045	172.31.0.0/20	-
<input checked="" type="checkbox"/> us-east-1d	subnet-0a8cb92b	172.31.80.0/20	-
<input checked="" type="checkbox"/> us-east-1e	subnet-328cfe03	172.31.48.0/20	-
<input checked="" type="checkbox"/> us-east-1f	subnet-cf05e3ce	172.31.64.0/20	-

aws Services Search for services, features, blogs, docs, and more [Alt+S]

EC2 RDS VPC DynamoDB

Elastic Beanstalk

environments Applications Change history

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

Instance subnets

<input checked="" type="checkbox"/> Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> us-east-1d	subnet-0a8cb92b	172.31.80.0/20	-
<input checked="" type="checkbox"/> us-east-1e	subnet-328cfe03	172.31.48.0/20	-
<input checked="" type="checkbox"/> us-east-1f	subnet-cf05e3ce	172.31.64.0/20	-

Elastic Beanstalk - Create App

Database settings

Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> us-east-1a	subnet-f718e6bb	172.31.16.0/20	-
<input checked="" type="checkbox"/> us-east-1b	subnet-a7143ff8	172.31.32.0/20	-
<input checked="" type="checkbox"/> us-east-1c	subnet-23cee045	172.31.0.0/20	-
<input checked="" type="checkbox"/> us-east-1d	subnet-0a8cb92b	172.31.80.0/20	-
<input checked="" type="checkbox"/> us-east-1e	subnet-328cfe03	172.31.48.0/20	-
<input checked="" type="checkbox"/> us-east-1f	subnet-cf03e5ce	172.31.64.0/20	-

Database subnets

Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/> us-east-1a	subnet-f718e6bb	172.31.16.0/20	-
<input type="checkbox"/> us-east-1b	subnet-a7143ff8	172.31.32.0/20	-
<input type="checkbox"/> us-east-1c	subnet-23cee045	172.31.0.0/20	-
<input type="checkbox"/> us-east-1d	subnet-0a8cb92b	172.31.80.0/20	-
<input type="checkbox"/> us-east-1e	subnet-328cfe03	172.31.48.0/20	-
<input type="checkbox"/> us-east-1f	subnet-cf03e5ce	172.31.64.0/20	-

Notifications

Email address:

Network

VPC: `vpc-24205a59 (172.31.0.0/16) | default vpc (default)`

Load balancer visibility: Public

Load balancer subnets: `subnet-23cee045, subnet-328cfe03, subnet-0a8cb92b, subnet-cf03e5ce, subnet-f718e6bb, subnet-a7143ff8`

Associate public IP address: enabled

Instance subnets: `subnet-23cee045, subnet-328cfe03, subnet-0a8cb92b, subnet-cf03e5ce, subnet-f718e6bb, subnet-a7143ff8`

Database

Engine:

Instance class:

Storage (GB):

Tags

Tags: none

Warning

An RDS database must have subnets selected in at least two Availability Zones. Configure the database subnets in the Network section.

Elastic Beanstalk - Environments

Tcbconference-env

Creating Tcbconference-env

This will take a few minutes..

9:14pm createEnvironment is starting

The screenshots illustrate the deployment process of a scalable application using AWS Elastic Beanstalk.

Screenshot 1: Creating a New Environment

This screenshot shows the "Creating Tcbconference-env" step in the Elastic Beanstalk console. A pink arrow points to the status message "createEnvironment is starting".

Screenshot 2: All Environments

This screenshot shows the "All environments" list. It includes columns for Environment name, Health, Application name, Date created, Last modified, URL, Running versions, Platform, Platform state, and Tier name. A pink arrow points to the "Platform" column for the Tcbconference-env row, which shows "Python 3.7 running on 64bit Amazon Linux 2".

Screenshot 3: Detailed Environment View

This screenshot provides a detailed view of the Tcbconference-env environment. It shows the health status as "Ok" (indicated by a green checkmark icon), the running version (tcb-conference-source), and the platform (Python 3.7 running on 64bit Amazon Linux 2). A pink arrow points to the "Health" section. Another pink arrow points to the "Recent events" table, which lists the following log entries:

Time	Type	Details
2022-02-14 21:18:28 UTC-0600	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 22 seconds ago and took 3 minutes.
2022-02-14 21:18:28 UTC-0600	INFO	Added instances [i-026abc5548078075, i-04278bb196748d085] to your environment.
2022-02-14 21:18:10 UTC-0600	INFO	Successfully launched environment: Tcbconference-env
2022-02-14 21:18:08 UTC-0600	INFO	Application available at Tcbconference-env.eba-sz8pqjmf.us-east-1.elasticbeanstalk.com.
2022-02-14 21:17:37 UTC-0600	INFO	Instance deployment completed successfully.

Verify that 2 EC2 instance has started

The screenshot shows the AWS EC2 Instances page with two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
Tcbconference...	i-026abcc5548078075	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-50-231-27.com...	3.90.231.27	-
Tcbconference...	i-04278bb196748d085	Running	t2.micro	Initializing	No alarms	us-east-1d	ec2-18-204-205-175.co...	18.204.205.175	-

Open an application in web browser

The screenshot shows the AWS Elastic Beanstalk Environment page for 'Tcbconference-env'. It displays the application URL: Tcbconference-env.eba-s28pqjmf.us-east-1.elasticbeanstalk.com. The environment status is shown as 'Ok' with a green checkmark icon.

Recent events:

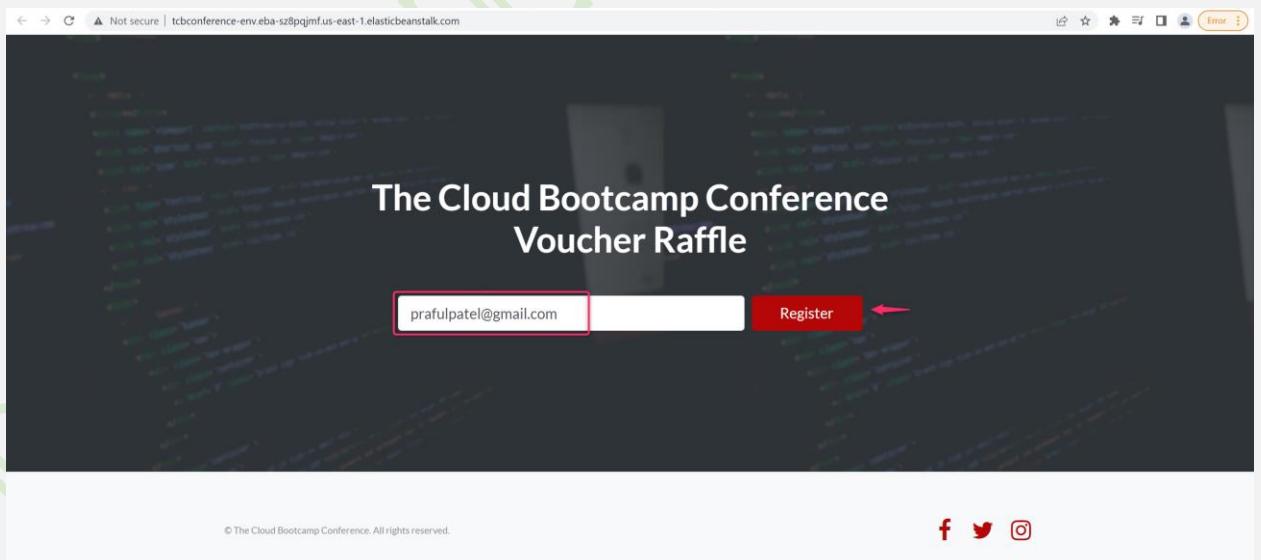
Time	Type	Details
2022-02-14 21:18:28 UTC-0600	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 22 seconds ago and took 3 minutes.
2022-02-14 21:18:28 UTC-0600	INFO	Added instances [i-026abcc5548078075, i-04278bb196748d085] to your environment.
2022-02-14 21:18:10 UTC-0600	INFO	Successfully launched environment: Tcbconference-env
2022-02-14 21:18:08 UTC-0600	INFO	Application available at Tcbconference-env.eba-s28pqjmf.us-east-1.elasticbeanstalk.com.



© The Cloud Bootcamp Conference. All rights reserved.



- Once the deploy completes, access the application and simulate a record insertion



© The Cloud Bootcamp Conference. All rights reserved.



- Upon inserting email address it gives an error



- Investigate the error in the logs

Elastic Beanstalk > Environments > Tcbconference-env > Logs

Logs

Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)

Log file	Time	EC2 instance	Type
Click Request Logs to request and review log files for all your servers.			

Elastic Beanstalk > Environments > Tcbconference-env > Logs

Logs

Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)

Log file	Time	EC2 instance	Type
Click Request Logs to request and review log files for all your servers.			

Request Logs ▾ Refresh

Last 100 Lines Full Logs

Elastic Beanstalk > Environments > Tcbconference-env > Logs

Logs

Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)

Log file	Time	EC2 instance	Type
Download	2022-02-14T21:31:31-06:00	i-026abcc5548078075	Last 100 Lines
Download	2022-02-14T21:31:31-06:00	i-04278bb196748d085	Last 100 Lines

Click on download and observe the log for an error logs

```

2022/02/15 03:31:31.030716 [INFO] reading event message file
2022/02/15 03:31:31.030811 [INFO] no env enviror Info file found, skip loading env tier info.
2022/02/15 03:31:31.030869 [INFO] Engine received EB command cfn-hup-exec

2022/02/15 03:31:31.134961 [INFO] Running command /bin/sh -c /opt/aws/bin/cfn-get-metadata -s arn:aws:cloudformation:us-east-1:500942944689:stack/awseb-e-nfsunwyummk-stack/59c9f710-8e0d-11ec-95e4-12645090582f -r AWSEBAutoScalingGroup -region us-east-1
2022/02/15 03:31:31.446055 [INFO] Running command /bin/sh -c /opt/aws/bin/cfn-get-metadata -s arn:aws:cloudformation:us-east-1:500942944689:stack/awseb-e-nfsunwyummk-stack/59c9f710-8e0d-11ec-95e4-12645090582f -r AWSEBBeanstalkMetadata
2022/02/15 03:31:31.758533 [INFO] checking whether command tail-log is applicable to this instance...
2022/02/15 03:31:31.758547 [INFO] this command is applicable to the instance, thus instance should execute command
2022/02/15 03:31:31.758551 [INFO] Engine command: (tail-log)

2022/02/15 03:31:31.758601 [INFO] Executing instruction: GetTaillogs
2022/02/15 03:31:31.758606 [INFO] Tail Logs...
2022/02/15 03:31:31.758859 [INFO] Running command /bin/sh -c tail -n 1000 /var/log/eb-engine.log

-----
/var/log/web.stdout.log
-----
Feb 15 03:27:34 ip-172-31-40-148 web: [2022-02-15 03:17:37 +0000] [3478] [INFO] Starting unicorn 2.0.0
Feb 15 03:27:34 ip-172-31-40-148 web: [2022-02-15 03:17:37 +0000] [3478] [INFO] Listening at: http://127.0.0.1:8000 (3478)
Feb 15 03:27:34 ip-172-31-40-148 web: [2022-02-15 03:17:37 +0000] [3478] [INFO] Using worker: gthread
Feb 15 03:27:34 ip-172-31-40-148 web: [2022-02-15 03:17:37 +0000] [3515] [INFO] Booting worker with pid: 3515
Feb 15 03:27:34 ip-172-31-40-148 web: [2022-02-15 03:27:34,884] [ERROR] in app: Exception on / [POST]
Feb 15 03:27:34 ip-172-31-40-148 web: /var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/flask/app.py", line 2447, in wsgi_app
Feb 15 03:27:34 ip-172-31-40-148 web: response = self.full_dispatch_request()
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/flask/app.py", line 1952, in full_dispatch_request
Feb 15 03:27:34 ip-172-31-40-148 web: rv = self.handle_user_exception(e)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/flask/app.py", line 1821, in handle_user_exception
Feb 15 03:27:34 ip-172-31-40-148 web: reraise(exc_type, exc_value, tb)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/flask/_compat.py", line 39, in reraise
Feb 15 03:27:34 ip-172-31-40-148 web: raise value
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/flask/app.py", line 1958, in full_dispatch_request
Feb 15 03:27:34 ip-172-31-40-148 web: self.preprocess_request()
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/flask/app.py", line 1936, in dispatch_request
Feb 15 03:27:34 ip-172-31-40-148 web: return self.view_functions[rule.endpoint](**req.view_args)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/current/application.py", line 27, in index
Feb 15 03:27:34 ip-172-31-40-148 web: user_resp = self.user_form(user_email.data)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/current/application.py", line 17, in put_user
Feb 15 03:27:34 ip-172-31-40-148 web: File "email": email
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/boto3/resources/factory.py", line 520, in do_action
Feb 15 03:27:34 ip-172-31-40-148 web: response = action(self, **kwargs)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/boto3/resources/action.py", line 83, in __call__
Feb 15 03:27:34 ip-172-31-40-148 web: self._make_api_call(operation_name)(*args, **params)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/botocore/client.py", line 337, in _api_call
Feb 15 03:27:34 ip-172-31-40-148 web: return self._make_api_call(operation_name, kwargs)
Feb 15 03:27:34 ip-172-31-40-148 web: File "/var/app/venv/staging-LQ1Mlest/lib/python3.7/site-packages/botocore/client.py", line 656, in _make_api_call
Feb 15 03:27:34 ip-172-31-40-148 web: raise error_class(parsed_response, operation_name)
Feb 15 03:27:34 ip-172-31-40-148 web: botocore.exceptions.ClientError: An error occurred (AccessDeniedException) when calling the PutItem operation: User: arn:aws:sts::500942944689:assumed-role/aws-elasticbeanstalk-ec2-role/I-026abc54ab978879 is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:500942944689:table/users
-----
/var/log/web-hooks.log
-----

```

- Add the permission AmazonDynamoDBFullAccess on the EC2 associate role in IAM.

Create and IAM role

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with navigation links like 'AWS Services', 'Search for services, features, blogs, docs, and more', and a dropdown for 'N. Virginia'. The main area displays the configuration for the 'Elastic Beanstalk' environment 'Tcbconference-env'. Under the 'Configuration' tab, several sections are visible: 'Capacity' (Max: 2, Metric: CPUUtilization, Min: 2, Period: 1, Placement: Scale down increment: -1, Scale up increment: 1, Scaling cooldown: 360 seconds, Statistic: Average, Unit: Percent, Upper threshold: 50), 'Load balancer' (Listeners: 1, Load balancer type: application, Processes: 1, Rules: 0, Shared: false, Store logs: disabled), 'Rolling updates and deployments' (Batch size: 100%, Command timeout: 600, Deployment policy: All at once, Healthy threshold: Ok, Ignore health check: disabled, Rolling updates: disabled), 'Security' (EC2 key pair: devogs, IAM instance profile: aws-elasticbeanstalk-ec2-role, Service role: arn:aws:iam::500942944689:role/aws-elasticbeanstalk-service-role), 'Monitoring' (CloudWatch Custom Metrics-Environment: CloudWatch Custom Metrics-Instance: Health event log streaming: disabled, Ignore HTTP 4xx: disabled, Ignore load balancer 4xx: disabled, System Enhanced), 'Managed updates' (Instance replacement: disabled, Managed updates: enabled, Update level: Minor and patch, Weekly update window: Wed:19:00), 'Notifications' (Email: --), 'Network' (Instance subnets: subnet-23ce0d45, subnet-328cf05, subnet-0a8cb92b, subnet-cf03e3cc, subnet-f718e6bb, subnet-a7143ff8, Public IP address: enabled, VPC: vpc-24205a59, Visibility: public), and 'Database' (Edit button). There are also 'Edit' buttons for each of the configuration sections.

IAM dashboard

Security recommendations

- Root user has MFA (Green checkmark)
- Add MFA for yourself (Red warning icon)
- Your user, iam_praful, does not have any active access keys that have been unused for more than a year. (Green checkmark)

AWS Account

Account ID: 500942944689
Account Alias: 500942944689 Create
Sign-in URL for IAM users in this account: https://500942944689.signin.amazonaws.com/sole

IAM resources

User groups	Users	Roles	Policies	Identity providers
8	9	16	4	0

What's new (View all) (Updates for features in IAM)

- IAM Access Analyzer helps you generate fine-grained policies that specify the required actions for more than 50 services. 6 months ago
- IAM Access Analyzer helps you generate IAM policies based on access activity found in your organization trail. 6 months ago
- IAM Access Analyzer adds new policy checks to help validate conditions during IAM policy authoring. 8 months ago
- AWS Amplify announces support for IAM permissions boundaries on Amplify-generated IAM roles. 8 months ago

Quick Links

- My security credentials
- Tools
- Policy simulator
- Web identity federation playground

Search the role and click on it

Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.

Roles (16) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities
aws-elasticbeanstalk-ec2-role	AWS Service: ec2

Click on attach policies – attach “DynamodbFullAccess” policy to EC2 role

Summary

Creation date: February 14, 2022, 21:13 (UTC-06:00)	ARN: arn:aws:iam::500942944689:role/aws-elasticbeanstalk-ec2-role	Instance profile ARN: arn:aws:iam::500942944689:instance-profile/aws-elasticbeanstalk-ec2-role
Last activity: 22 minutes ago	Maximum session duration: 1 hour	

Permissions

Permissions policies (3): You can attach up to 10 managed policies.

Policy name	Type	Description
AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server environment access to upload log files to Amazon S3.
AWSElasticBeanstalkMulticontainerDoc...	AWS managed	Provide the instances in your multicontainer Docker environment access to use the Amazon EC2 Container Service to manage container dep...
AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker environment access to upload log files to Amazon S3, to use Amazon SQS to monitor your application's ...

Add permissions

- Attach policies (Red arrow)
- Create inline policy

Attach policy to aws-elasticbeanstalk-ec2-role

Current permissions policies (3)

Other permissions policies (Selected 1/727)

Filter policies by property or policy name and press enter

"dynamodb" X Clear filters

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon Dyn...
<input type="checkbox"/> AWSLambdaDynamoDBExecutionRole	AWS managed	Provides list and read access to Dyr...
<input type="checkbox"/> AmazonDynamoDBReadOnlyAccess	AWS managed	Provides read only access to Amaz...
<input type="checkbox"/> AWSLambdaInvocation-DynamoDB	AWS managed	Provides read access to DynamoDB ...

Cancel Attach policies

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Roles (1)

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

aws-elasticbeanstalk-ec2-role

Policy has been successfully attached to role

aws-elasticbeanstalk-ec2-role

Summary

Creation date: February 14, 2022, 21:13 (UTC-06:00)

Last activity: 23 minutes ago

ARN: arn:aws:iam::500942944689:role/aws-elasticbeanstalk-ec2-role

Maximum session duration: 1 hour

Instance profile ARN: arn:aws:iam::500942944689:instance-profile/aws-elasticbeanstalk-ec2-role

Permissions (4)

Permissions policies (4)

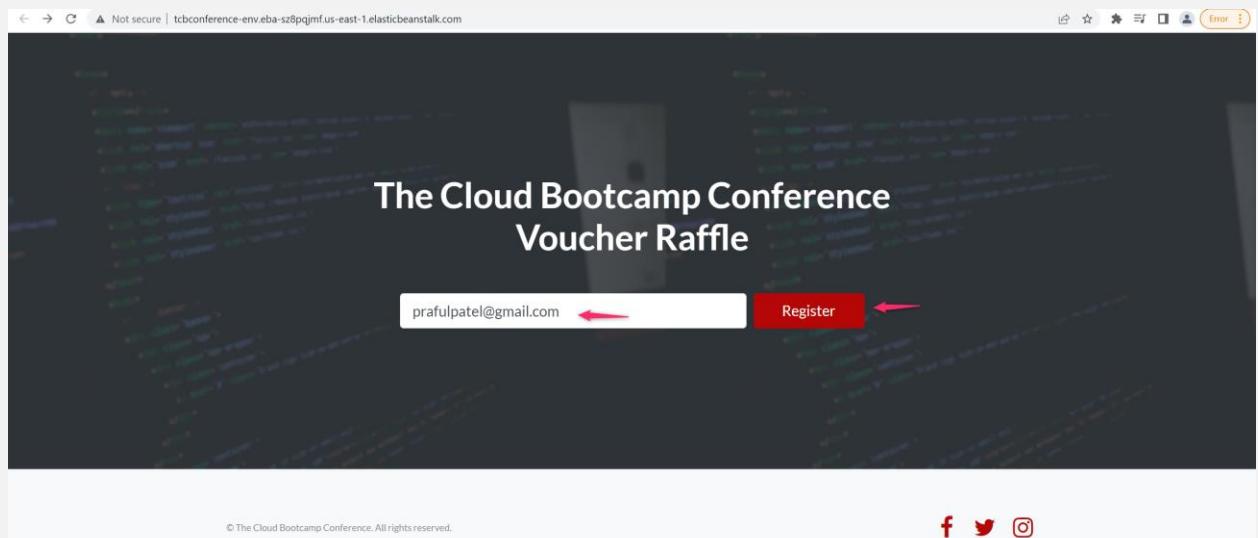
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

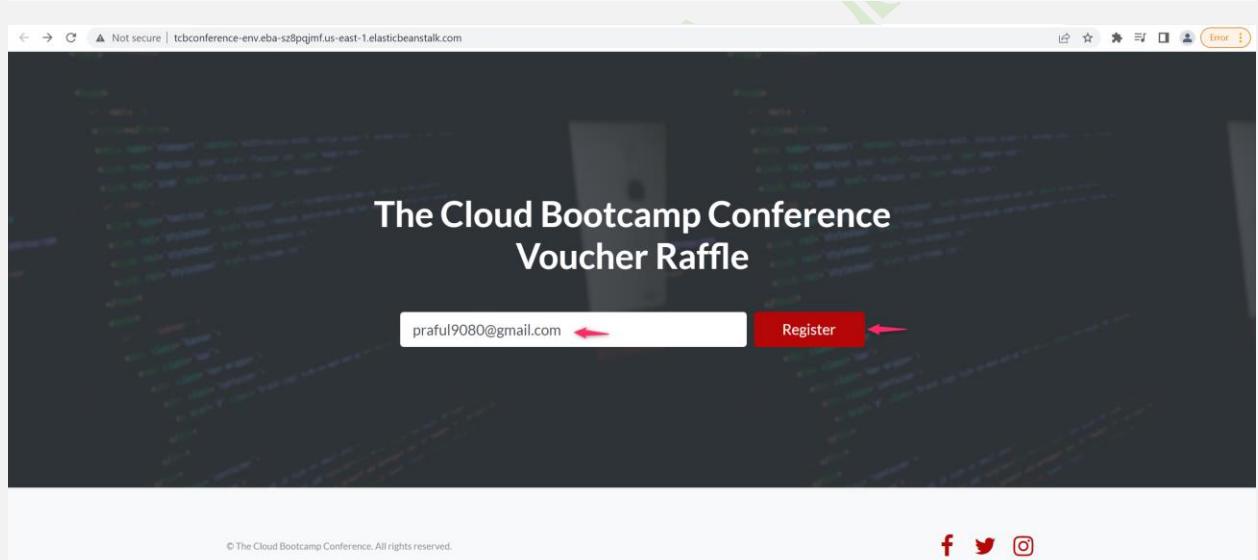
Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon DynamoDB via the...
<input type="checkbox"/> AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server environm...
<input type="checkbox"/> AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicontainer Docke...
<input type="checkbox"/> AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker environment ...

Simulate Remove Add permissions

- Try to insert the registry once again. It should be sucessful.



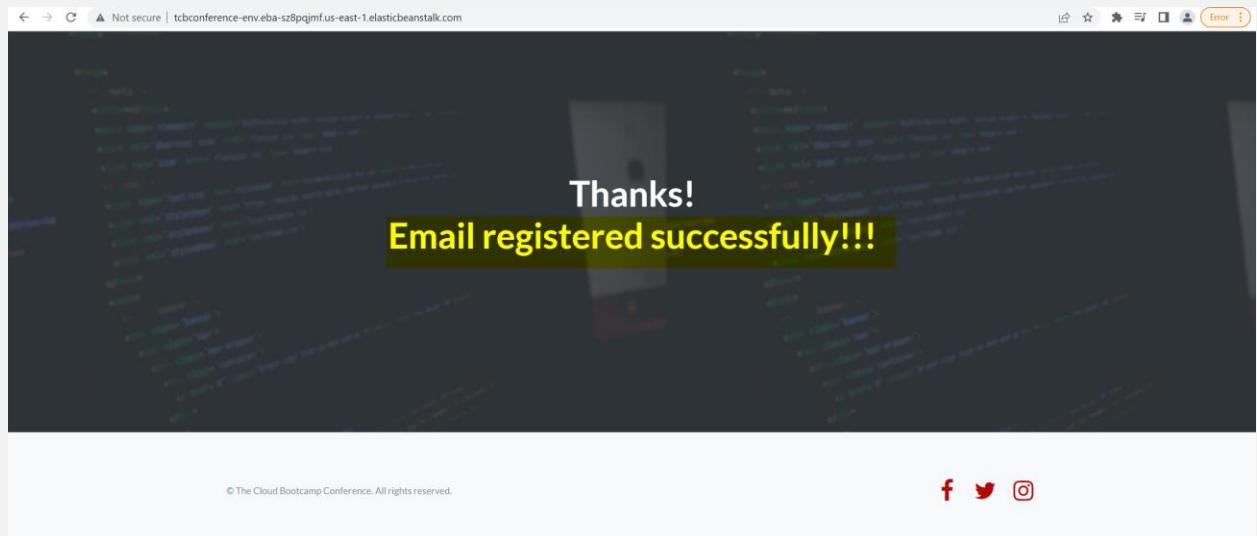
© The Cloud Bootcamp Conference. All rights reserved.



© The Cloud Bootcamp Conference. All rights reserved.



Data has added successfully from frontend web application

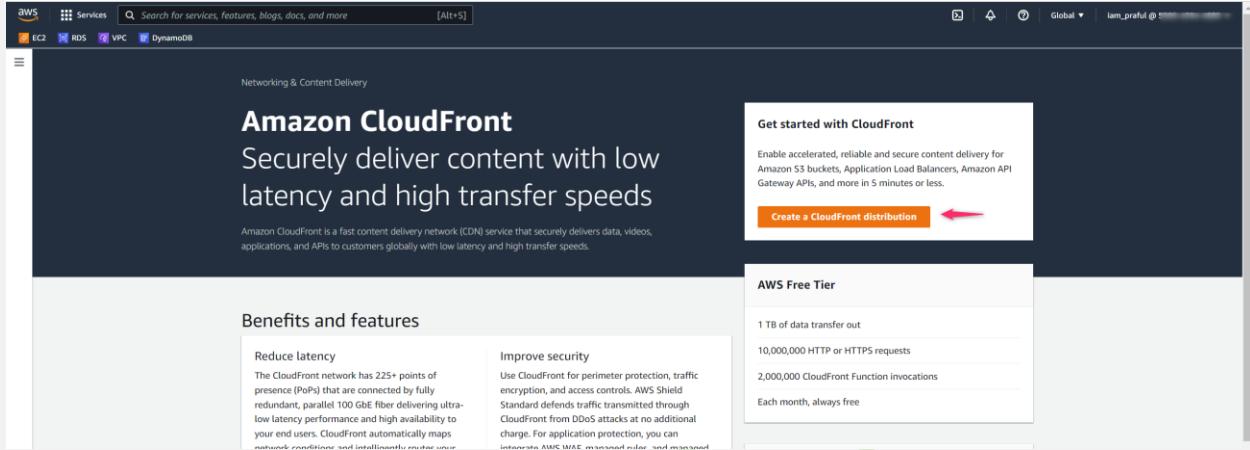


Verify that data has been added successfully to the backend database

The image contains two screenshots of the AWS DynamoDB console, both titled "DynamoDB > Items > users".

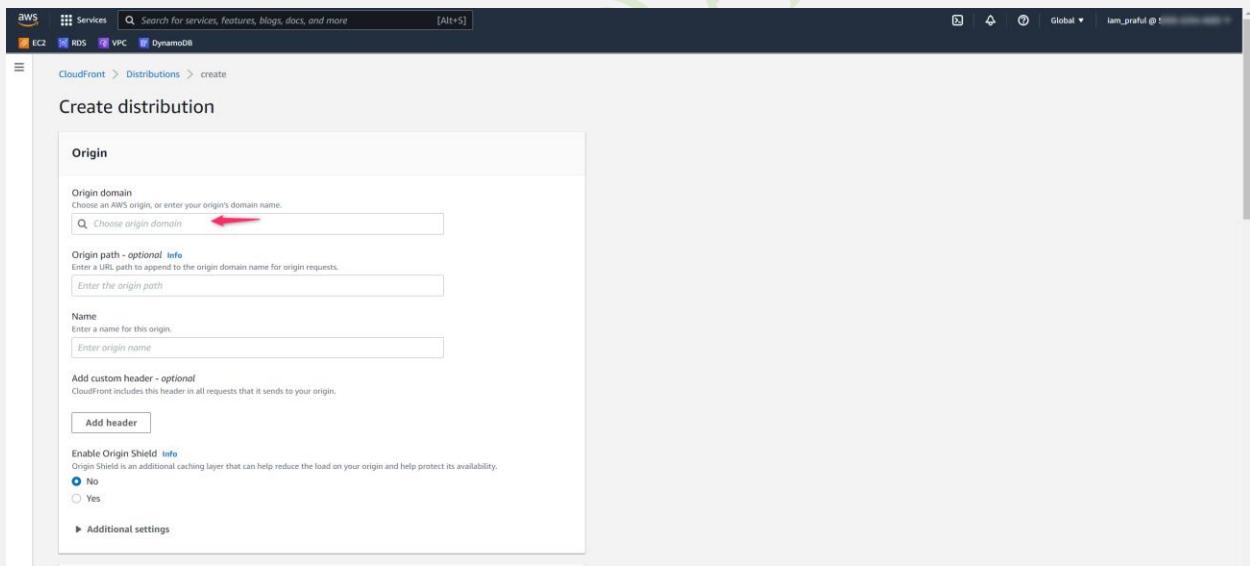
Screenshot 1: Shows the "Tables (1)" section with a single entry: "users". In the "Items returned (1)" section, there is one item listed: "email" with the value "prafulpatel@gmail.com". A red arrow points to this email address.

Screenshot 2: Shows the "Tables (1)" section with a single entry: "users". In the "Items returned (2)" section, there are two items listed: "email" with the value "prafulpatel@gmail.com" and "email" with the value "praful9080@gmail.com". Two red arrows point to these email addresses.

 **Implementation Phase 2 in Action: Create Cloudfront distribution**


The screenshot shows the AWS CloudFront landing page. At the top, there's a search bar and navigation links for EC2, RDS, VPC, and DynamoDB. Below the header, it says "Networking & Content Delivery" and features the "Amazon CloudFront" logo with the tagline "Securely deliver content with low latency and high transfer speeds". A brief description of CloudFront as a fast content delivery network follows. To the right, there's a "Get started with CloudFront" section with a "Create a CloudFront distribution" button, which has a red arrow pointing to it. Another section titled "AWS Free Tier" lists benefits like 1 TB of data transfer out, 10,000,000 HTTP or HTTPS requests, 2,000,000 CloudFront Function invocations, and "Each month, always free".

Search elastic load balancer



The screenshot shows the "Create distribution" step of the CloudFront wizard. It's titled "Origin" and includes fields for "Origin domain" (with a red arrow pointing to the "Choose origin domain" input field), "Origin path - optional", "Name", and "Add custom header - optional". There's also a section for "Enable Origin Shield" with radio buttons for "No" (selected) and "Yes". A "Additional settings" link is at the bottom.

The screenshot shows the 'Create distribution' page in the AWS CloudFront console. In the 'Origin' section, under 'Origin domain', the search bar contains 'choose origin domain'. Below it, a list of origins includes 'Amazon S3' with several items like 'cf-templates-nj3lg3tqkg9-us-east-1.s3.amazonaws.com' and 'cf-templates-nj3lg3tqkg9-us-east-2.s3.amazonaws.com'. Under 'Elastic load balancer', the item 'awseb-AWSEB-V5THJP6DV77H' is selected, indicated by a red arrow. Other sections like 'Mediastore container' and 'Mediapackage container' show 'No origins available'.

The screenshot shows the 'Create distribution' page in the AWS CloudFront console. In the 'Origin' section, under 'Protocol', the 'HTTP only' option is selected, indicated by a red arrow. Other options like 'HTTPS only' and 'Match viewer' are available. The 'HTTP port' is set to 80 and 'HTTPS port' is set to 443. Under 'Minimum origin SSL protocol', 'TLSv1.2' is selected. The 'Origin path - optional' field is empty. The 'Name' field contains 'awseb-awseb-v5thjp6dv77h-1772143995.us-east-1.elb.amazonaws.com'.

Default cache behavior

Path pattern [Info](#)

Default (*)

Compress objects automatically [Info](#)

- No
- Yes

Viewer

Viewer protocol policy

- HTTP and HTTPS
- Redirect HTTP to HTTPS
- HTTPS only

Allowed HTTP methods

- GET, HEAD
- GET, HEAD, OPTIONS
- GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods

GET and HEAD methods are cached by default.

- OPTIONS

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

- No
- Yes

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

- Cache policy and origin request policy (recommended)
- Legacy cache settings

Price class: Info

Choose the price class associated with the maximum price that you want to pay.

- Use all edge locations (best performance)
- Use only North America and Europe
- Use North America, Europe, Asia, Middle East, and Africa

AWS WAF web ACL - optional

Choose the web ACL in AWS WAF to associate with this distribution.

Choose web ACL

Alternate domain name (CNAME) - optional

Add the custom domain names that you use in URLs for the files served by this distribution.

Add item

To add a list of alternative domain names, use the bulk editor.

Custom SSL certificate - optional

Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

Choose certificate

Request certificate

Supported HTTP versions

Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.

- HTTP/2

Default root object - optional

The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

Standard logging

Get logs of viewer requests delivered to an Amazon S3 bucket.

- Off
- On

IPv6

- Off
- On

Description - optional

E1FPBT7G06SGVT

General

Distribution domain name: d3m8rtymkr92ya.cloudfront.net

ARN: arn:aws:cloudfront:500942944689:distribution/E1FPBT7G06SGVT

Last modified: Deploying

Description: -

Price class: Use all edge locations (best performance)

Supported HTTP versions: HTTP/2, HTTP/1.1, HTTP/1.0

AWS WAF: -

Alternate domain names: -

Standard logging: Off

Cookie logging: Off

Default root object: -

Cloudfront deployed

Distributions (1) Info

ID	Description	Domain name	Alternate domain names	Origins	Status	Last modified
E1FPBT7G06SGVT	-	d3m8rtymkr92ya.cloudfr...	-	awseb-awseb-v5thjp6dv77h-17; Enabled	Enabled	February 15, 2022 at 4:14:...

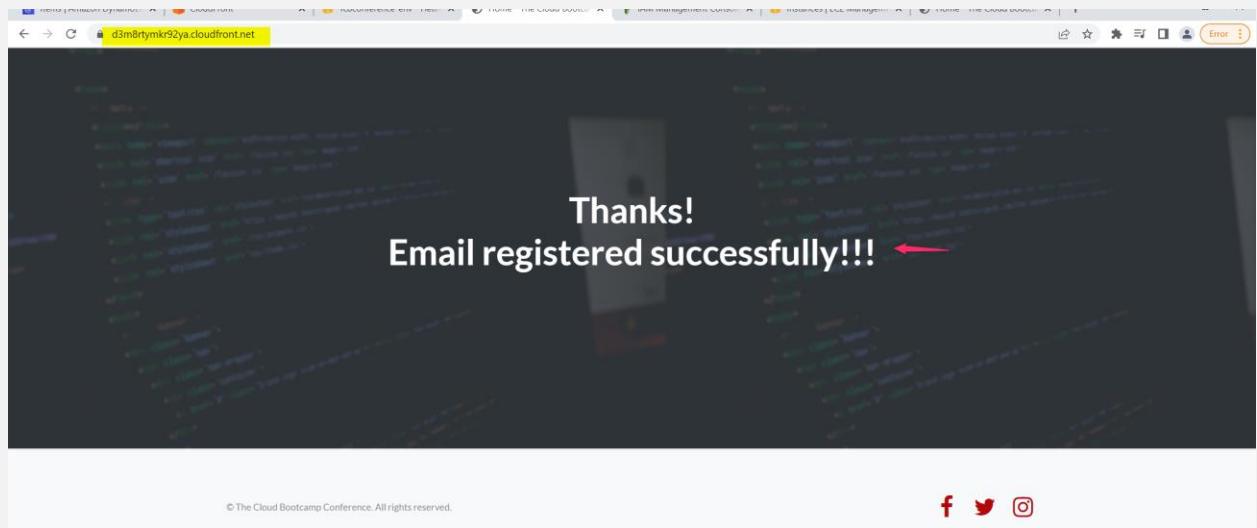
Verify that cloudfront URL is able to access and enter some data

The Cloud Bootcamp Conference Voucher Raffle

praful_cloudfronttest@gmail.com

Register

© The Cloud Bootcamp Conference. All rights reserved.



⊕ Implementation Phase 3 in Action : Perform load test and verify application auto scale up

Steps to run during the "dry-run & cutover":

Perform load testing.

- Install the stress tool to perform the load testing

```
sudo amazon-linux-extras install epel -y  
sudo yum install stress -y
```

```
stress -c 4
```

- Explore the resources created by the AWS Elastic Beanstalk (EC2, ELB, Auto Scaling Group, Cloud Watch resources) and also the auto scaling process

- Once you finish exploring it, please remove the Elastic Beanstalk application, Elastic Beanstalk environment , disable and delete the CloudFront distribution and finally delete the DynamoDB users table

Situation before scaling and load test

Elastic Beanstalk Configuration (Screenshot 1)

Elastic Beanstalk Health Overview (Screenshot 2)

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses
i-026abcc5548078075	Ok	1 hour	1	0.1	1	0
i-04278bb196748d085	Ok	1 hour	1	0.1	1	0

EC2 Instances (Screenshot 3)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP
Tcbconference...	i-026abcc5548078075	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-90-231-27.com...	3.90.231.27
Tcbconference...	i-04278bb196748d085	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-18-204-205-175.co...	18.204.205.175

The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a navigation sidebar with various EC2-related options like Dashboard, Global View, Instances, Images, Elastic Block Store, and Network & Security. The main area displays 'Auto Scaling groups (1)'. A search bar at the top says 'Search your Auto Scaling groups'. Below it is a table with columns: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. One row is highlighted: 'awseb-e-nfsuwymml | AWSEBEC2LaunchTemplate_brrQLaTj43F | 2 | - | 2 | 2 | us-east-1a, us-east-1b, u...'. At the bottom right of the table, there are 'Edit' and 'Delete' buttons, and a 'Create an Auto Scaling group' button.

This screenshot shows the 'Instances management' tab for the 'awseb-e-nfsuwymml' Auto Scaling group. The table lists two instances: 'i-026abcc5548078075' and 'i-04278bb196748d085', both in 'InService' state with 't2.micro' instance type. The 'Health status' column shows 'Healthy' for both. The table has columns: Instance ID, Lifecycle, Instance type, Weighted capacity, Launch template/configurati..., Availability Zone, and Health status. At the bottom, there are 'Actions' and 'Create lifecycle hook' buttons.

Go to one of EC2 instance and access one of public ip

Instance summary for i-026abcc5548078075 (Tcbconference-env)

Instance ID	i-026abcc5548078075 (Tcbconference-env)	Public IPv4 address	3.90.231.27 [open address]
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-40-148.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-40-148.ec2.internal
Instance type	t2.micro	Elastic IP addresses	-
AWS Compute Optimizer finding	(Opt-in to AWS Compute Optimizer for recommendations. Learn more)	IAM Role	aws-elasticbeanstalk-ec2-role

Details **Security** **Networking** **Storage** **Status checks** **Monitoring** **Tags**

Instance details

Platform	AMI ID	Monitoring
Linux/UNIX (Inferred)	ami-01d52c2dd41814c37	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	aws-elasticbeanstalk-amzn-2.0.20220121.64bit-eb_python37_amazon_linux_2-hvm-2022-01-30T22-29	Disabled

Login to EC2 server

```

• MobaXterm Personal Edition v21.3 •
(SSH client, X server and network tools)

> SSH session to ec2-user@3.90.231.27
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✘ (disabled or not supported by server)

> For more info, ctrl+click on help or visit our website.

[ec2-user@ip-172-31-40-148 ~]$
```

Amazon Linux 2 AMI

This EC2 instance is managed by AWS Elastic Beanstalk. Changes made via SSH WILL BE LOST if the instance is replaced by auto-scaling. For more information on customizing your Elastic Beanstalk environment, see our documentation here: <http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html>

- Install the stress tool to perform the load testing

```
sudo amazon-linux-extras install epel -y
```

```
sudo yum install stress -y
```

```
(2/3): epel/x86_64/updateinfo
(3/3): epel/x86_64/primary_db
218 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package stress.x86_64 0:1.0.4-16.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version       Repository      Size
=====
stress            x86_64   1.0.4-16.el7   epel           39 k

Transaction Summary
=====
Install 1 Package

Total download size: 39 k
Installed size: 94 k
Downloading packages:
warning: /var/cache/yum/x86_64/2/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Public key for stress-1.0.4-16.el7.x86_64.rpm is not installed
stress-1.0.4-16.el7.x86_64.rpm
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x352c64e5:
  Userid : "Fedora EPEL (7) <epel@fedoraproject.org>"
  Fingerprint: 91e9 7d7c 4a5e 96f1 7f3e 888f 6a2f aeaf 352c 64e5
  Package : epel-release-7-11.noarch (@amzn2extra-epel)
  From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stress-1.0.4-16.el7.x86_64
  Verifying   : stress-1.0.4-16.el7.x86_64
                                                               1/1
                                                               1/1
Installed:
  stress.x86_64 0:1.0.4-16.el7
Complete!
[ec2-user@ip-172-31-40-148 ~]$
```

Increase load on one of the EC2 server using stress command

```
stress -c 4
```

```
[ec2-user@ip-172-31-40-148 ~]$ stress -c 4
stress: info: [31410] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
```

6. 172.16.1.159 (praful) 8. 172.16.1.133 (praful) 11. 3.90.231.27 (ec2-user) 12. 3.90.231.27 (ec2-user) (1)

```
top - 05:11:45 up 1:56, 2 users, load average: 1.77, 0.44, 0.15
Tasks: 106 total, 6 running, 62 sleeping, 0 stopped, 0 zombie
%Cpu(s):100.0 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1005828 total, 209480 free, 233504 used, 562844 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 610044 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
31411 ec2-user 20 0 7572 96 0 R 24.9 0.0 0:08.68 stress
31412 ec2-user 20 0 7572 96 0 R 24.9 0.0 0:08.67 stress
31413 ec2-user 20 0 7572 96 0 R 24.9 0.0 0:08.68 stress
31414 ec2-user 20 0 7572 96 0 R 24.9 0.0 0:08.68 stress
3515 webapp 20 0 956144 45824 8544 S 0.3 4.6 0:03.20 gunicorn
1 root 20 0 125764 5732 3976 S 0.0 0.6 0:02.55 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/u:0:0H
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
7 root 20 0 0 0 0 S 0.0 0.0 0:00.18 ksoftirqd/0
8 root 20 0 0 0 0 I 0.0 0.0 0:00.42 rcu_sched
9 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_bh
10 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
11 root rt 0 0 0 0 S 0.0 0.0 0:00.01 watchdog/0
12 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
15 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
16 root 20 0 0 0 0 I 0.0 0.0 0:00.01 kworker/u30:1
191 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
192 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom_reaper
193 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback
195 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kcompactd0
196 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd
197 root 39 19 0 0 0 S 0.0 0.0 0:00.00 khugepaged
198 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 crypto
199 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd
201 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd
543 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xen-balloon
554 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 md
557 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 edac-poller
562 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 watchdogd
703 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kaudited
709 root 20 0 0 0 0 S 0.0 0.0 0:00.03 kswapd0
799 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 xfsalloc
800 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 xfs_mru_cache
854 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kthrotld
865 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xenbus
```

CPU has gone up to 100%

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
31411	ec2-user	20	0	7572	96	0 R	25.0	0.0	0.0	1:04.70	stress
31412	ec2-user	20	0	7572	96	0 R	25.0	0.0	0.0	1:04.70	stress
31413	ec2-user	20	0	7572	96	0 R	25.0	0.0	0.0	1:04.70	stress
31414	ec2-user	20	0	7572	96	0 R	24.7	0.0	0.0	1:04.70	stress
1	root	20	0	125764	5732	3976 S	0.0	0.6	0.0	0:02.55	systemd
2	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.18	ksoftirqd/0
8	root	20	0	0	0	0 I	0.0	0.0	0.0	0:00.42	rcu_sched
9	root	20	0	0	0	0 I	0.0	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0 S	0.0	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0 S	0.0	0.0	0.0	0:00.01	watchdog/0
12	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	cpuhp/0
14	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	kdevtmpfs
15	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	netns
16	root	20	0	0	0	0 I	0.0	0.0	0.0	0:00.01	kworker/u30:1
191	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	khungtaskd
192	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	oom_reaper
193	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	writelock
195	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	Kcompactd0
196	root	25	5	0	0	0 S	0.0	0.0	0.0	0:00.00	ksmd
197	root	39	19	0	0	0 S	0.0	0.0	0.0	0:00.00	khugepaged
198	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	crypto
199	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	Kintegrityd
201	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	kblockd
543	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	xen-balloon
554	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	md
557	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	edac-poller
562	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	watchdogd
703	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.01	kauitd
709	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.03	kswapd0
799	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	xfsalloc
800	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	xfs_mru_cache
854	root	0	-20	0	0	0 I	0.0	0.0	0.0	0:00.00	kthrotld
865	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.00	xenbus
866	root	20	0	0	0	0 S	0.0	0.0	0.0	0:00.02	xenwatch

EC2 instance has considered as unhealthy and started to de register

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses
Overall	Warning	N/A	N/A	0.4	100%	0.0%
i-04278bb196748d085	Ok	2 hours	1	0.2	2	0
i-026abcc5548078075	Degraded	2 hours	1	0.2	2	0

All environments

Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
Tcbconference-env	Warning	tcb-conference	2022-02-14 21:14:02 UTC-0600	2022-02-14 21:31:36 UTC-0600	Tcbconference-env.eba-sz8pqjmif.us-east-1.elasticbeanstalk.com	tcb-conference-source	Python 3.7 running on 64bit Amazon Linux 2	Supported	WebServer

Events

Time	Type	Details
2022-02-14 23:12:23 UTC-0600	WARN	Environment health has transitioned from Ok to Warning. 1 out of 2 instances are impacted. See instance health for details.
2022-02-14 21:46:30 UTC-0600	INFO	Deleted log fragments for this environment.
2022-02-14 21:37:27 UTC-0600	INFO	Environment health has transitioned from Warning to Ok.
2022-02-14 21:32:27 UTC-0600	WARN	Environment health has transitioned from Ok to Warning. One or more TargetGroups associated with the environment are in a reduced health state - awseb-AWSEB-400H004Y5OSB - Warning
2022-02-14 21:51:36 UTC-0600	INFO	Pulled logs for environment instances.
2022-02-14 21:31:31 UTC-0600	INFO	Instance deployment completed successfully.
2022-02-14 21:31:29 UTC-0600	INFO	requestEnvironmentInfo is starting.

New EC 2 instance has started registering to auto scaling group and added to load balancer target

EC2 > Auto Scaling groups > awseb-e-nsfuwyummk-stack-AWSEBAutoScalingGroup-F3E2MFU72314

Details | Activity | Automatic scaling | **Instance management** | Monitoring | Instance refresh

Instances (3)

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configurati...	Availability Zone	Health status
i-026abcc5548078075	InService	t2.micro	-	AWSEBEC2LaunchTemplate_brrQlLaTj	us-east-1b	Healthy
i-04278bb196748d085	InService	t2.micro	-	AWSEBEC2LaunchTemplate_brrQlLaTj	us-east-1d	Healthy
i-0e91c98748d790c28	Pending	t2.micro	-	AWSEBEC2LaunchTemplate_brrQlLaTj	us-east-1f	Healthy

Lifecycle hooks (0) Info

No lifecycle hooks are currently configured.

Warm pool Info

No warm pool currently configured.

EC2 > Instances > awseb-e-nsfuwyummk-stack-AWSEBAutoScalingGroup-F3E2MFU72314

Instances (3) Info

Search: Instance state = running

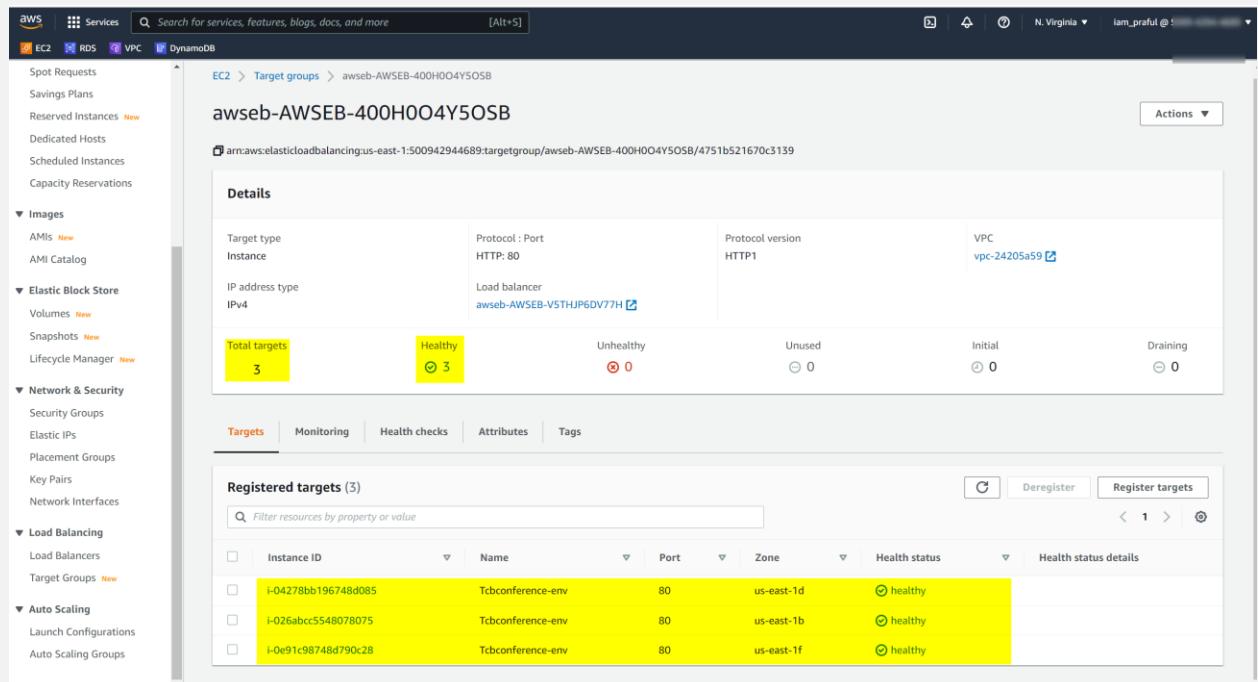
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Tcbconference...	i-0e91c98748d790c28	Running	t2.micro	Initializing	No alarms	us-east-1f	ec2-3-238-173-47.com...	3.238.173.47
Tcbconference...	i-026abcc5548078075	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-90-231-27.com...	3.90.231.27
Tcbconference...	i-04278bb196748d085	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-18-204-205-175.co...	18.204.205.171

Select an instance

The screenshot shows the AWS Elastic Beanstalk Health Overview page. The left sidebar is for 'Elastic Beanstalk' with sections like Environments, Applications, Change history, tcb-conference, Application versions, Saved configurations, Tcbconference-env, Go to environment, Configuration, Logs, Health (which is selected), Monitoring, Alarms, Managed updates, Events, Tags, and Recent environments. The main area is titled 'Enhanced health overview' with the sub-section 'Instances: 3 Total, 1 Degraded, 2 Ok'. It includes a 'Learn more' link and a 'Filter by' dropdown. A checkbox for 'Auto refresh (10 s)' is checked. The table has columns: Instance ID, Status, Running, Deployment ID, Requests/sec, 2xx Responses, and 3xx Responses. The first row, 'Overall', is highlighted in yellow and labeled 'Warning' with the note '1 out of 3 instances are impacted. See instance health for details.' The other two rows show 'Ok' status for instances i-0e91c98748d790c28 and i-04278bb196748d085, and 'Degraded' status for instance i-026abcc5548078075 with the note '100 % of CPU is in use.'

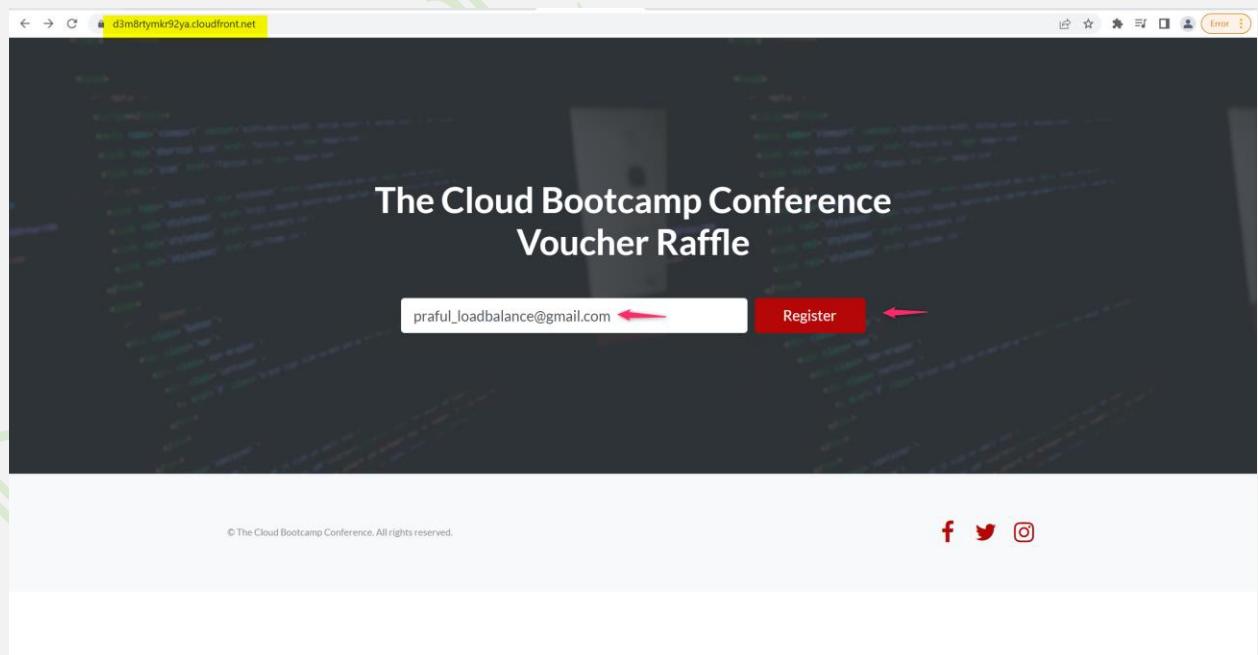
Now 3 EC2 instance has up and running due to load on web application

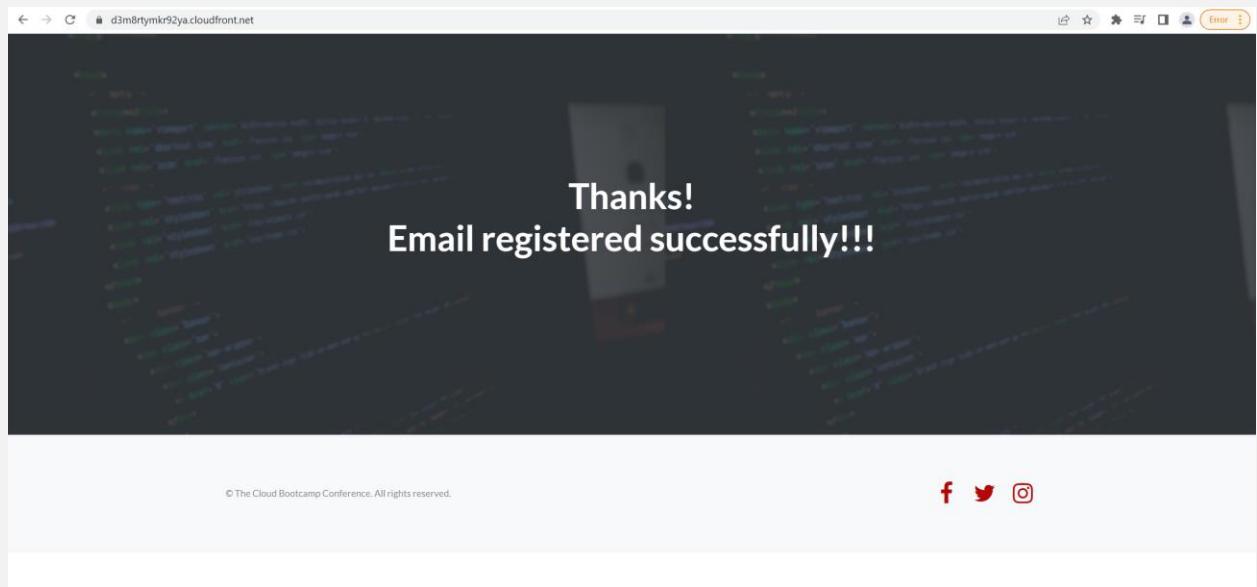
The screenshot shows the AWS EC2 Instances page. The left sidebar includes Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs New, AMI Catalog), Elastic Block Store (Volumes New, Snapshots New, Lifecycle Manager New), and Network & Security. The main area displays 'Instances (3) Info' with a search bar and filters for 'Instance state = running'. The table lists three instances: Tcbconference..., i-0e91c98748d790c28, Tcbconference..., i-026abcc5548078075, and Tcbconference..., i-04278bb196748d085. Each instance is shown as 'Running' with status check counts of 2/2 checks passed and no alarms. The table includes columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 ...



The screenshot shows the AWS CloudFront web app URL: d3m8tymkr92ys.cloudfront.net. The page displays a registration form for a conference voucher raffle. The form includes a text input field containing the email address `praful_loadbalance@gmail.com` and a red "Register" button. Two red arrows point to these elements. At the bottom of the page, there is a copyright notice: "© The Cloud Bootcamp Conference. All rights reserved." and social media icons for Facebook, Twitter, and Instagram.

Let's verify through cloudfront web app url if the load balancer is working correctly





Verify that Data is added successfully to backend database using cloudfront url while loadbalancer is up

The screenshot shows the AWS DynamoDB console. On the left, the navigation pane shows "DynamoDB" selected. Under "Tables", "users" is selected. The main area shows a table named "users" with 5 items returned. The items are listed as follows:

email
praful_cloudfronttest@gmail.com
prafulpatel@gmail.com
praful_testcloudfront@gmail.com
praful9080@gmail.com
praful_leadbalance@gmail.com

Verify that after few minutes the Elastic Beanstalk is back to Healthy status

Elastic Beanstalk > Environments > Tcbconference-env

Tcbconference-env

Health: Ok

Running version: tcb-conference-source

Platform: Python 3.7 running on 64bit Amazon Linux 2/3.5.10

Recent events:

Time	Type	Details
2022-02-14 23:31:22 UTC-0600	INFO	Environment health has transitioned from Warning to Ok.
2022-02-14 23:30:22 UTC-0600	INFO	Removed instance [i-025abcc5548078075] from your environment.
2022-02-14 23:21:22 UTC-0600	INFO	Added instance [i-0e91c98748d790c28] to your environment.
2022-02-14 23:12:23 UTC-0600	WARN	Environment health has transitioned from Ok to Warning. 1 out of 2 instances are impacted. See instance health for details.

Clean up:

Clean up all resources in order to avoid unnecessary charges on the account.

Terminate Environment

All environments

Environment name	Health	Application name	Date created	Last modified	URL	Running versions
Tcbconference-env	Ok	tcb-conference	2022-02-14 21:14:02 UTC-0600	2022-02-14 21:31:36 UTC-0600	Tcbconference-env.eba-sz8pmjmfus-east-1.elasticbeanstalk.com	tcb-conference-source

Actions: Create a new environment

- Load configuration
- Save configuration
- Swap environment URLs
- Clone environment
- Abort current operation
- Restart app server(s)
- Rebuild environment

Terminate environment

Elastic Beanstalk > Environments > Tcbconference-env

On September 2020, Elastic Beanstalk introduced the EnhancedHealthAuthEnabled option. Instances that report enhanced health information will require authorization when this option value is set to 'true'. If you're using an Elastic Beanstalk managed policy for your environment's instance profile, which is the default when using the Elastic Beanstalk console or EB CLI, you can safely enable this option.

If you are using a custom instance profile instead of a managed policy, your environment might show the No Data health status. This happens because the instances aren't authorized for the action that communicates enhanced health data to the service. To authorize the action, add permission to your instance profile according to Enhanced health authorization in the Amazon Elastic Beanstalk Developer Guide.

Elastic Beanstalk is terminating your environment. View Events

Tcbconference-env

Tcbconference-env.eba-sz8pgjinf.us-east-1.elasticbeanstalk.com (e-nisfuwyummk)
Application name: tcb-conference

Health

Running version

Platform

Recent events

Time	Type	Details
2022-02-14 23:40:42 UTC-0600	INFO	terminateEnvironment is starting.