

# **Course Code and Name: CSI0805 Cloud Computing**

## **PRACTICAL 1**

### **Screenshots:**

	Name	Date modified	Type	Size
persona	📁 .settings	05-08-2010 09:40	File folder	
	📁 classes	14-12-2022 01:15	File folder	
	📁 config	14-12-2022 01:15	File folder	
	📁 jars	14-12-2022 01:15	File folder	
	📁 javadoc	14-12-2022 01:15	File folder	
	📁 resources	14-12-2022 01:15	File folder	
	📁 source	05-08-2010 09:40	File folder	
	📁 test	05-08-2010 09:40	File folder	
	📄 .classpath	14-12-2022 01:15	CLASSPATH File	1 KB
	📄 .project	14-12-2022 01:15	PROJECT File	1 KB
	📄 readme	14-12-2022 01:15	Text Document	1 KB
	%B run	14-12-2022 01:15	Windows Batch File	1 KB



Main Configuration Data Center Configuration Advanced

Simulation Duration: 60.0 min ▾

User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	2	60	100	3	9	1000	100

Add New Remove

Application Deployment Configuration: Service Broker Policy: Closest Data Center ▾

Data Center	# VMs	Image Size	Memory	BW
DC1	5	10000	512	1000

Add New Remove

Cancel Load Configuration Save Configuration Done

## Configure Simulation

Main Configuration Data Center Configuration Advanced

Data Centers:

Name	Region	Arch	OS	VMM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC1	0x86	Linux	Xen		0.1	0.05	0.1	0.1	2

Add New

Remove

Cancel

Load Configuration

Save Configuration

Done

Main Configuration Data Center Configuration Advanced

User grouping factor in User Bases:  
(Equivalent to number of simultaneous users from a single user base)

10

Request grouping factor in Data Centers:  
(Equivalent to number of simultaneous requests a single application server instance can support)

10

Executable instruction length per request:  
(bytes)

100

Load balancing policy across VM's in a single Data Center:

Round Robin

Cancel

Load Configuration

Save Configuration

Done

 Java Update Checker

Java Upd

A new ve  
installed.  
Click here



## Configure Internet Characteristics

Use this screen to configure the Internet characteristics.

### Delay Matrix

The transmission delay between regions. Units in milliseconds

Region\Region	0	1	2	3	4	5
0	25	100	150	250	250	100
1	100	25	250	500	350	200
2	150	250	25	150	150	200
3	250	500	150	25	500	500
4	250	350	150	500	25	500
5	100	200	200	500	500	25

### Bandwidth Matrix

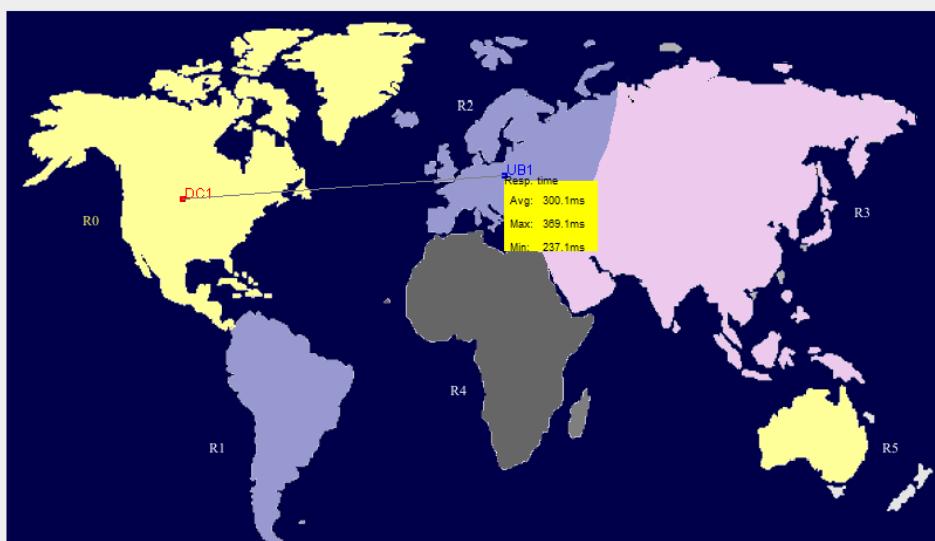
The available bandwidth between regions for the simulated application. Units in Mbps

Region\Region	0	1	2	3	4	5
0	2,000	1,000	1,000	1,000	1,000	1,000
1	1,000	800	1,000	1,000	1,000	1,000
2	1,000	1,000	2,500	1,000	1,000	1,000
3	1,000	1,000	1,000	1,500	1,000	1,000
4	1,000	1,000	1,000	1,000	500	1,000
5	1,000	1,000	1,000	1,000	1,000	2,000

Done

Cancel

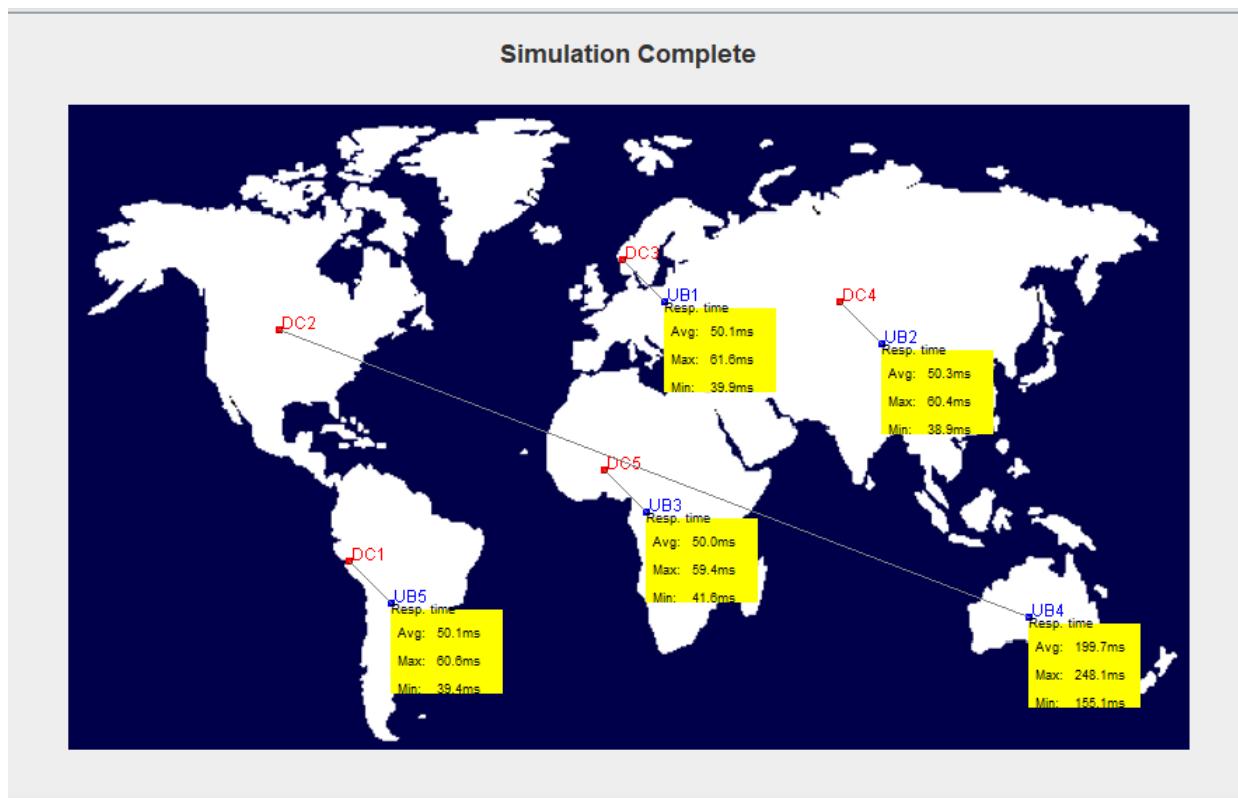
### Simulation Complete



**Course Code and Name:** CSI0805 Cloud Computing

## **PRACTICAL 2**

### **Screenshots:**



## Overall Response Time Summary

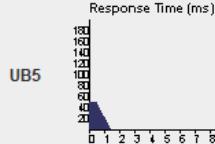
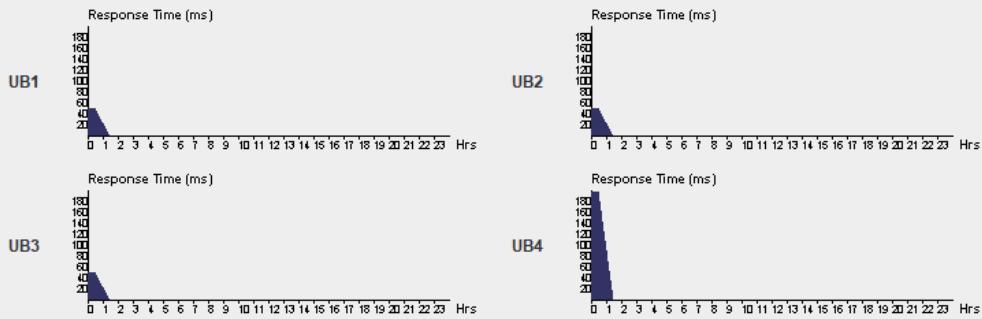
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	116.84	38.86	248.13
Data Center Processing Time:	0.32	0.02	0.88

[Export Results](#)

### Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.143	39.859	61.607
UB2	50.32	38.863	60.357
UB3	50.028	41.637	59.383
UB4	199.677	155.14	248.134
UB5	50.094	39.384	60.634

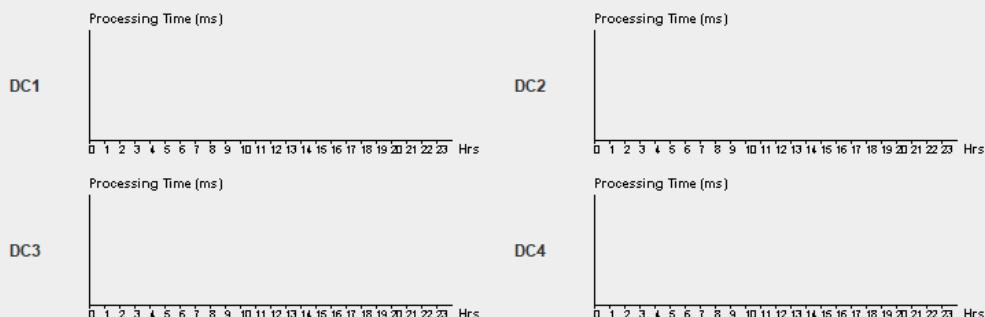
### User Base Hourly Average Response Times



### Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.471	0.018	0.882
DC2	0.125	0.019	0.139
DC3	0.47	0.024	0.854
DC4	0.461	0.016	0.857
DC5	0.488	0.04	0.87

### Data Center Hourly Average Processing Times





## Cost

Total Virtual Machine Cost: \$2.51

Total Data Transfer Cost: \$0.43

**Grand Total :** \$2.94

Data Center	VM Cost	Data Transfer Cost	Total
DC2	0.502	0.208	0.709
DC1	0.502	0.033	0.534
DC4	0.502	0.065	0.567
DC3	0.502	0.107	0.608
DC5	0.502	0.02	0.522

```
DC2-Broker finalizing, submitted cloudlets=2061 processing cloudlets=0 ,allRequestsProcessed=19715
DC1-Broker finalizing, submitted cloudlets=652 processing cloudlets=0 ,allRequestsProcessed=6241
DC4-Broker finalizing, submitted cloudlets=649 processing cloudlets=0 ,allRequestsProcessed=6187
DC5-Broker finalizing, submitted cloudlets=203 processing cloudlets=0 ,allRequestsProcessed=1933
DC3-Broker finalizing, submitted cloudlets=1059 processing cloudlets=0 ,allRequestsProcessed=10136
Simulation completed.
***** Vm allocations in DC1
0->262
1->262
2->262
3->262
4->262
***** Vm allocations in DC2
0->826
1->826
2->826
3->825
4->825
***** Vm allocations in DC3
0->425
1->425
2->425
3->425
4->424
***** Vm allocations in DC4
0->261
1->261
2->261
3->261
4->260
***** Vm allocations in DC5
0->83
1->83
2->82
3->82
4->82
```

```
*****Datacenter: DC1*****
User id      Debt
6            5128
*****
*****Datacenter: DC2*****
User id      Debt
8            5128
*****
*****Datacenter: DC3*****
User id      Debt
10           5128
*****
*****Datacenter: DC4*****
User id      Debt
12           5128
*****
*****Datacenter: DC5*****
User id      Debt
14           5128
*****
Simulation finished at 3611620.0
```

**Course Code and Name:** CSI0805 Cloud Computing

## **PRACTICAL 3**

### **Compute Services:**

The word "compute" in cloud computing refers to ideas and things associated with software computation. It is a general phrase that refers to the computing power, memory, networking, storage, and other resources necessary for any application to succeed computationally.

Computing services include all information technology and computer systems (including software, application service provider services, hosted computing services, information technology and telecommunication hardware and other equipment), whether or not they are used to transmit, store, maintain, organise, present, generate, process, or analyse data and information.

For example, Amazon Elastic Compute Cloud (EC2), Amazon Elastic Container Service (ECS), Amazon Elastic Container Service for Kubernetes (EKS), Amazon Lightsail, AWS Batch, and AWS Lambda are just a few of the cost-effective and flexible computing services that AWS offers to meet the needs of your business. While AWS has complete control over some services, such Amazon EC2, you have considerable influence over the underlying resources for other services.

Type of compute services:

Compare AWS compute services  
Amazon EC2  
Amazon EC2 Auto Scaling  
Amazon EC2 Image Builder  
Amazon Lightsail  
AWS App Runner  
AWS Batch  
AWS Elastic Beanstalk  
AWS Fargate  
AWS Lambda  
AWS Serverless Application Repository  
AWS Outposts  
AWS Wavelength  
VMware Cloud on AWS

Benefits of Compute Services:

Cost Savings  
Security  
Flexibility  
Mobility  
Insight  
Increased Collaboration  
Quality Control  
Disaster Recovery  
Loss Prevention  
Automatic Software Updates  
Competitive Edge  
Sustainability

## **Containers:**

Containers are software packages that come with everything needed to execute in any environment. Containers virtualize the operating system in this manner, enabling them to run anywhere, including on a developer's own laptop as well as on a public cloud or a private data centre. Every aspect of Google, including Gmail, YouTube, and Search, runs in containers. Our development teams can work quickly, deliver software effectively, and reach previously unimaginable scales thanks to containerization.

Containers are software packages that come with everything needed to execute in any environment. Containers virtualize the operating system in this manner, enabling them to run anywhere, including on a developer's own laptop as well as on a public cloud or a private data centre. Every aspect of Google, including Gmail, YouTube, and Search, runs in containers. Our development teams can work quickly, deliver software effectively, and reach previously unimaginable scales thanks to containerization.

However, a VM is essentially a standalone machine with its own operating system, so it takes a lot longer to start up and run than a container. VMs images, which are used to create new VMs, are heavier than container images and more difficult to automate.

In the cloud, the most common scenario is running containers on top of compute instances, which are technically virtual machines. Cloud providers are now offering the ability to run containers directly on their bare metal servers, without VMs as an intermediary, a model known as “container instances”.

## Use case of containers:

Containers are becoming increasingly important in cloud environments. Many organizations are considering containers as an alternative to virtual machines (VMs), which were traditionally the preferred option for large-scale enterprise workloads.

1. microservices
2. DevOps
3. hybrid and multi cloud
4. application modernization

## Types of containers:

Dry storage containers

Flat rack containers

Refrigerated containers

Special purpose containers

## Benefits of containers:

Separation of responsibility

Application isolation

More consistent operation

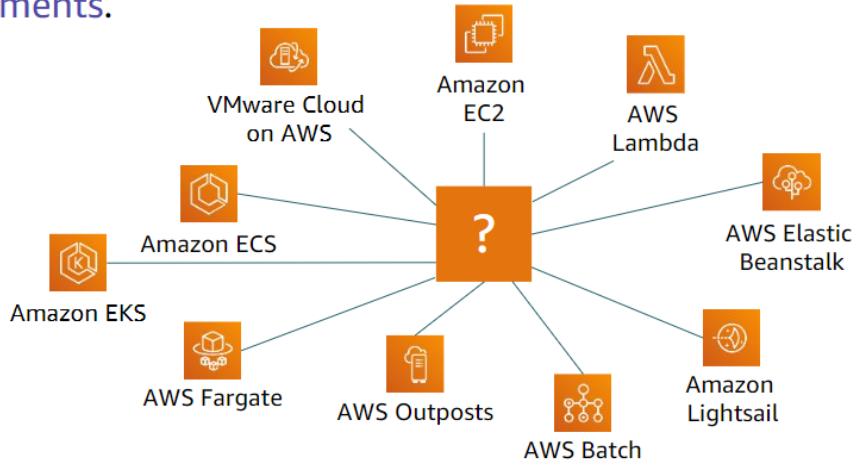
Greater efficiency

Increased portability

## Screenshots:

### Choosing a service

The service you select depends on your business goals and technology requirements.



aws

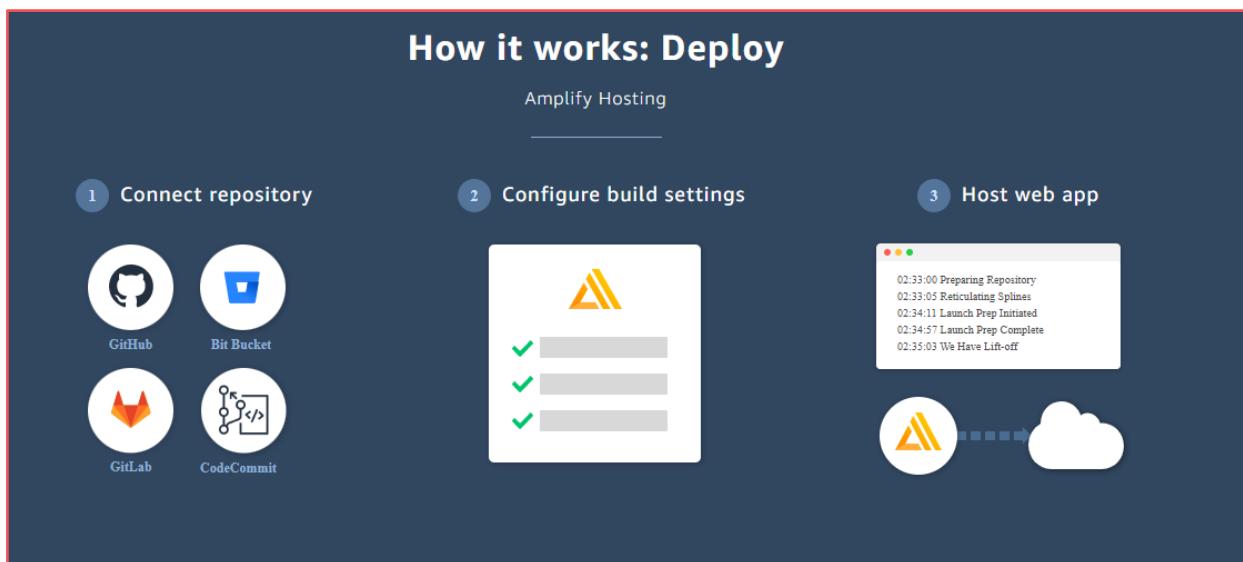
# **Course Code and Name: CSI0805 Cloud Computing**

## **PRACTICAL 4**

**Step-1:** Hosting a website is a 3-step process.

**Step-2:** We have access to domain management, Notifications, access control, monitoring, redirects settings, etc.

### **Screenshots:**



**AWS Amplify**

### Manual deploy

Manually upload objects to deploy your app. You can choose to drag and drop the artifacts directly, pull a zip from an existing S3 bucket or any other URL.

**Start a manual deployment**

**App name**  
Give this app a name or we will generate a default for you

**Environment name**  
Give this resource a meaningful environment name, like dev, test, or prod, or we will generate a default for you

**Method**

- Drag and drop 
- Amazon S3 
- Any URL 

 **index.zip**

**Cancel** **Previous** **Save and deploy**

**AWS Amplify**

All apps > Guesser

### Guesser

The app homepage lists all deployed frontend and backend environments.

**Hosting environments** **Backend environments**

This tab lists all connected branches, select a branch to view build details.

**test**

Deployment successfully completed.  
  
 Domain: <https://test.dj40psabq5ps.amplifyapp.com> Last deployment: 10/1/2023, 9:51:11 pm

Drag and drop your project's build output directory or zip file here to update your app, or, choose another method.

**Add environment** **Choose files**

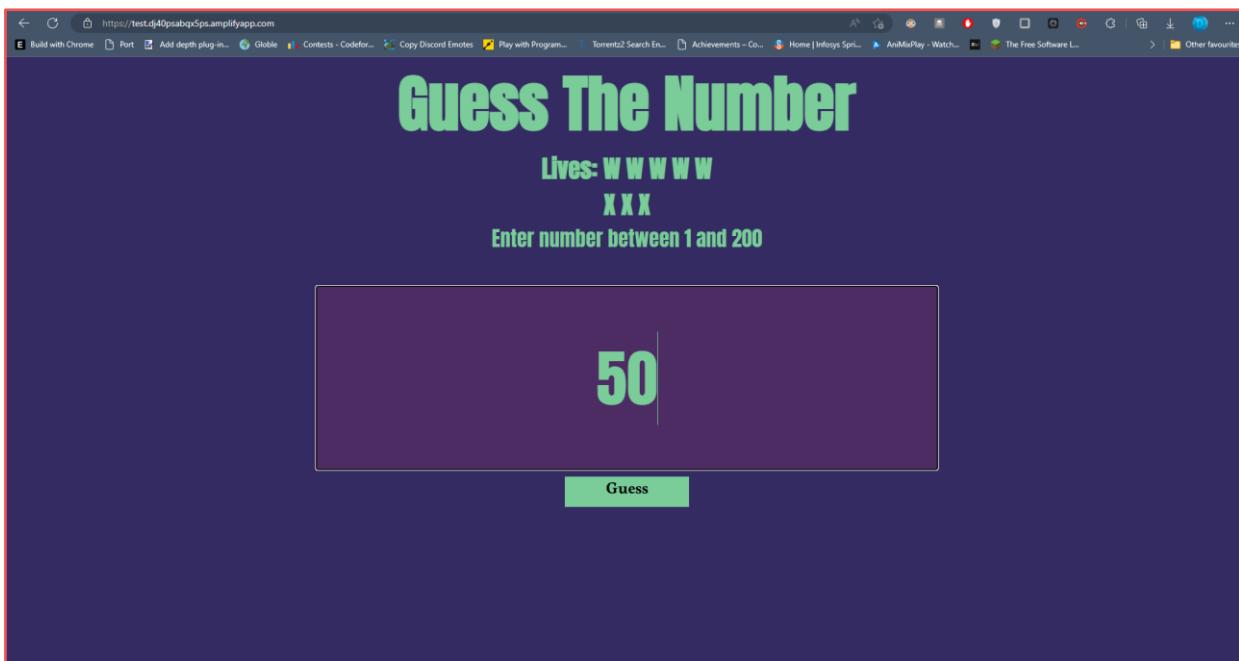
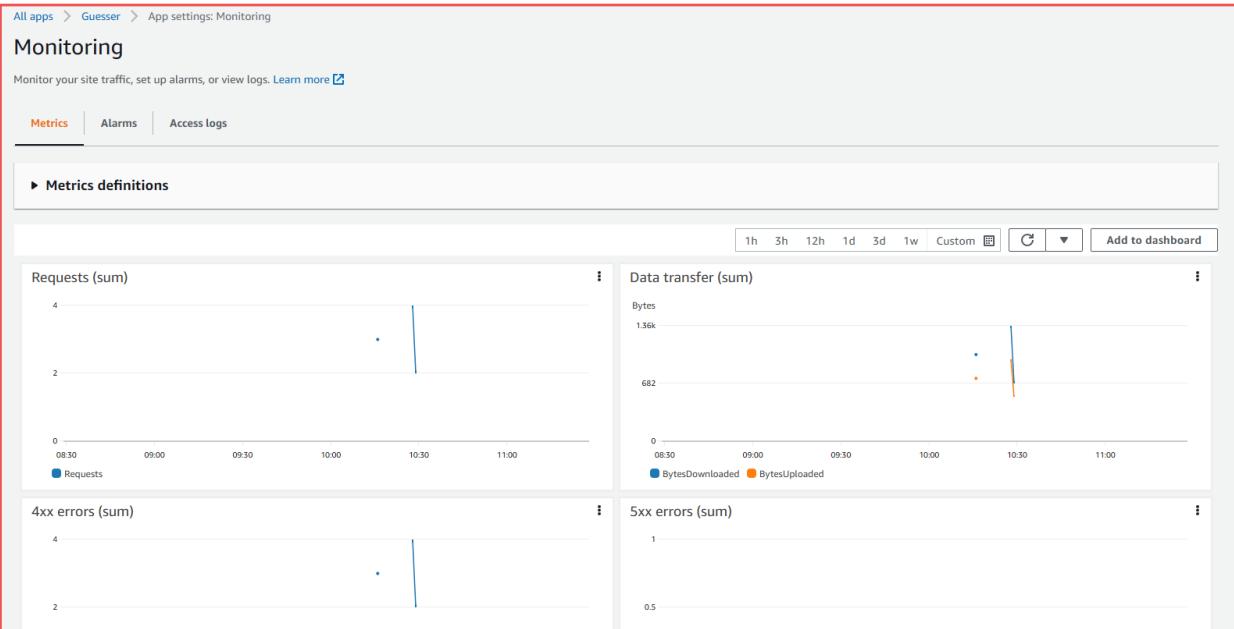
All apps > Guesser > App settings: General

### Guesser

**App settings**

App name Guesser	Platform Web
Production environment URL <a href="https://test.dj40psabq5ps.amplifyapp.com">https://test.dj40psabq5ps.amplifyapp.com</a>	App ARN am:aws:amplify:ap-south-1:844332920379:apps/dj40psabq5ps
Service role	Created at 26/12/2022, 9:51:07 pm
Framework	Updated at 26/12/2022, 9:51:07 pm

**Edit** **Delete app**



# Course Code and Name: CSI0805 Cloud Computing

## PRACTICAL 5

### Screenshots:

#### Load balancer description:

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar has sections for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Feedback. The main content area shows a load balancer named 'ts01'. The 'Basic Configuration' tab is selected, displaying details such as Name (ts01), DNS name (ts01-1679829667.ap-south-1.elb.amazonaws.com), Type (Classic (Migrate Now)), Scheme (internet-facing), and Availability Zones (subnet-0167cd2422f7e9701 - ap-south-1a, subnet-028d87d1afbd7e621 - ap-south-1b, subnet-04571cae08f777610 - ap-south-1c). The 'Port Configuration' tab shows port 80 (HTTP) forwarding to 443 (HTTPS) with Backend Authentication: Disabled and Stickiness: Disabled. There is also an 'Edit stickiness' button. The 'Security' section is partially visible at the bottom.

# Load balancer instances:

The screenshot shows the AWS EC2 Management console with the 'Load balancers' section selected. The main view displays the 'Instances' tab for a load balancer named 'ts01'. A message indicates 'There are no instances registered to this load balancer'. Below this, there is a table titled 'Edit Availability Zones' showing three availability zones: ap-south-1a, ap-south-1b, and ap-south-1c, each associated with a specific subnet ID and CIDR range.

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
ap-south-1a	subnet-0167cd2422f7e9701	172.31.32.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
ap-south-1b	subnet-028d87d1afbd7e621	172.31.0.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
ap-south-1c	subnet-04571cac08f777610	172.31.16.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer

# Load balancer health check:

The screenshot shows the AWS EC2 Management console with the 'Load balancers' section selected. The main view displays the 'Health check' tab for a load balancer named 'ts01'. The configuration includes a Ping Target of 'HTTP:80', a Timeout of '5 seconds', an Interval of '30 seconds', an Unhealthy threshold of '2', and a Healthy threshold of '10'. There is a 'Edit Health Check' button available for modification.

# Load balancer listeners:

The screenshot shows the AWS EC2 Management console with the URL [ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LBDetails:clbName=ts01](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LBDetails:clbName=ts01). The left sidebar shows the New EC2 Experience interface with sections like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-options Instances, Instance Types), Images (with sub-options AMIs, AMI Catalog), and Feedback. The main content area is titled 'EC2 | Load balancers | ts01' and 'Load balancer: ts01'. It has tabs for Description, Instances, Health check, **Listeners**, Monitoring, Tags, and Migration. The 'Listeners' tab is selected. A table shows the configuration for the load balancer:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Cipher	SSL Certificate
HTTP	80	HTTPS	443	N/A	N/A

An 'Edit' button is located below the table.

# Load balancer monitoring:

The screenshot shows the same AWS EC2 Management console URL as the previous screenshot. The left sidebar is identical. The main content area is titled 'EC2 | Load balancers | ts01' and 'Load balancer: ts01'. The 'Monitoring' tab is selected. It includes a section for 'Manage alarms in CloudWatch' and a 'CloudWatch metrics' section. The 'Showing data for: Last Hour' dropdown is set to 'Last Hour'. Below this, it says 'Below are your CloudWatch metrics for the selected resources (a maximum of 10). Click on a graph to see an expanded view. All times shown are in UTC.' There are six line graphs: 'Unhealthy Hosts Count', 'Healthy Hosts Count', 'Average Latency Milliseconds', 'Requests Count', 'HTTP 5XXs Count', and 'ELB 5XXs Count'. Each graph shows data from 09:00 to 09:30 on 1/16.

## Load balancer tags:

The screenshot shows the AWS EC2 Management console. The URL is [ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LBDetails:clbName=ts01](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LBDetails:clbName=ts01). The page title is "EC2 | Load balancers | ts01". The "Tags" tab is selected. A table shows one tag: "Key" is "ts" and "Value" is "tirth".

## Load balancer migration:

The screenshot shows the same AWS EC2 Management console URL as the previous screenshot. The "Migration" tab is selected. A message says "Migrate this Classic Load Balancer to a next generation load balancer. See Comparison of Elastic Load Balancing Products." Below it is a blue button labeled "Launch ALB Migration Wizard".

# **Course Code and Name: CSI0805 Cloud Computing**

## **PRACTICAL 6**

### **Screenshots:**

#### **AWS auto scaling:**

The screenshot shows the AWS Auto Scaling Groups page. The navigation bar at the top includes the AWS logo, a services menu, a search bar, and user information for 'nandan shah'. The main content area shows the path 'EC2 > Auto Scaling groups > MyGroup'. Below this, there's a 'MyGroup' heading and a navigation bar with tabs: Details (selected), Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. On the left, a sidebar lists EC2-related services like EC2 Dashboard, Global View, Events, Tags, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), and Images. The central 'Group details' section contains the following information:

Group details	
Desired capacity	Auto Scaling group name
1	MyGroup
Minimum capacity	Date created
0	Thu Jan 19 2023 15:19:13 GMT+0530 (India Standard Time)
Maximum capacity	Amazon Resource Name (ARN)
1	arn:aws:autoscaling:ap-northeast-1:243587954769:autoScalingGroup:731d95fd-4f54-4278-ab1e-012fc5c46d88:autoScalingGroupName/MyGroup

AWS Services Search [Alt+S] Tokyo nandan shah

New EC2 Experience Tell us what you think X

Auto Scale your Amazon EC2 Instances Ahead of Demand Explore how the new predictive scaling policy of EC2 Auto Scaling helps you improve availability for your applications. Learn More X ⓘ

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
MyGroup	MyGroup	1	-	1	0

AWS Services Search [Alt+S] Tokyo nandan shah

New EC2 Experience Tell us what you think X

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations

Launch configuration

Launch configuration MyGroup	AMI ID ami-0bba69335379e17f8	Security groups <a href="#">sg-0955fc034c59d8d3c</a>
Instance type t2.micro	Key pair name key2	Create time Thu Jan 19 2023 15:19:13 GMT+0530 (India Standard Time)
Storage (volumes) -		

[View details in the launch configuration console](#)

Network

Availability Zones ap-northeast-1a	Subnet ID subnet-05b9e785646ad1cff
---------------------------------------	---------------------------------------

Load balancing

Load balancer target groups -	Classic Load Balancers
----------------------------------	------------------------

Health check type: EC2

Health check grace period: 0

### Advanced configurations

Instance scale-in protection: Not protected from scale in	Termination policies: Default	Maximum instance lifetime: -	Default instance warmup: Disabled
Placement group: -	Suspended processes: -	Default cooldown: 300	Service-linked role: arn:aws:iam::243587954769:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling

### Tags (0)

Key	Value	Tag new instances
-----	-------	-------------------

New EC2 Experience Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs
- AMI Catalog

### Activity notifications (0)

No notifications are currently specified

Create notification

### Activity history (1)

Status	Description	Cause	Start time
Successful	Attaching an existing EC2 instance: i-0e5a605b0f0fa5156d	At 2023-01-19T09:49:14Z an instance was added in response to user request. Keeping the capacity at the new 1.	2023 January 19, 03:19:14 PM +05:30

AWS Services Search [Alt+S] Tokyo nandan shah

New EC2 Experience Tell us what you think.

EC2 > Auto Scaling groups > MyGroup

## MyGroup

Details Activity Automatic scaling Instance management Monitoring Instance refresh

### Instances (1)

Filter instances

Instance ID	Lifecycle	Instance Type	Weighted Capacity	Launch Configuration	Availability Zone	Health Status	Protect From Termination
i-0e5a605b00fa5156d	InService	t2.micro	-		ap-northe...	Healthy	

### Lifecycle hooks (0) Info

Filter lifecycle hooks

Name	Lifecycle transition	Default result	Heartbeat timeout	Notification targets	Role ARN

No lifecycle hooks are currently configured.

Lifecycle hooks help you perform custom actions on instances as they launch and before they terminate.

Create lifecycle hook

Actions

C

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

## Monitoring:

AWS Services Search [Alt+S] Tokyo nandan shah

New EC2 Experience Tell us what you think.

EC2 > Auto Scaling groups > MyGroup

## MyGroup

Details Activity Automatic scaling Instance management Monitoring Instance refresh

### CloudWatch monitoring details

Auto Scaling EC2

All times shown are in UTC.

View all CloudWatch metrics

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

CPU Utilization (Percent) Disk Reads (Bytes) Disk Read Operations (Operations/Second) Disk Writes (Bytes)

07:03 10:02

# Course Code and Name: CSI0805 Cloud Computing

## PRACTICAL 7

### Screenshots:

#### Public and private bucket:

The screenshot shows the AWS S3 Management Console interface. The left sidebar has sections for Buckets, Storage Lens, and Feature spotlight. The main area displays an Account snapshot and a list of Buckets. There are two buckets listed: 'tirthshah' and 'tss01'. The 'tirthshah' bucket is marked as 'not public' and was created on January 23, 2023. The 'tss01' bucket is marked as 'Public' and was created on January 16, 2023.

Name	AWS Region	Access	Creation date
tirthshah	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	January 23, 2023, 14:39:42 (UTC+05:30)
tss01	Asia Pacific (Mumbai) ap-south-1	Public	January 16, 2023, 14:25:55 (UTC+05:30)

# Publicly accessible bucket:

The screenshot shows the AWS S3 console interface. The left sidebar is titled "Amazon S3" and includes sections for Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards, AWS Organizations settings), Feature spotlight, and AWS Marketplace for S3. The main content area is titled "tss01" and shows the "Info" tab. A red banner at the top of the main content area says "Publicly accessible". Below this, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The "Objects" tab is selected, showing a table with one object: "tirth.html" (html type, 13.3 KB, Standard storage class). There are buttons for Actions (Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, Upload), a search bar for "Find objects by prefix", and navigation controls (back, forward, search, refresh).

# Public bucket permission:

The screenshot shows the AWS S3 console interface, similar to the previous one but with different tabs selected. The left sidebar is identical. The main content area is titled "tss01" and shows the "Permissions" tab. The "Permissions overview" section shows "Access" set to "Public". The "Block public access (bucket settings)" section has "Block all public access" turned "Off". The "Bucket policy" section indicates that the bucket policy is empty. There are buttons for Edit and Delete in the Bucket policy section. The footer of the page includes standard links like Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

## Public bucket policy:

The screenshot shows the AWS S3 console for a bucket named 'tss01'. In the left sidebar, under 'Bucket policy', there is a JSON code editor containing the following policy:

```
{ "Version": "2012-10-17", "Statement": [ { "Sid": "PublicReadGetObject", "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "s3:GetObject", "Resource": "arn:aws:s3:::tss01/*" } ] }
```

Below the policy, there is a section titled 'Object Ownership' with the following details:

Object Ownership [Info](#)  
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Object Ownership  
Bucket owner preferred  
ACLs are enabled and can be used to grant access to this bucket and its objects. If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

## Public bucket access control list:

The screenshot shows the AWS S3 console for a bucket named 'tss01'. In the left sidebar, under 'Access control list (ACL)', there is a table showing grants:

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: <a href="#">http://acs.amazonaws.com/groups/global/AllUsers</a>	List, Write	Read, Write
Everyone (public access) Group: <a href="#">http://acs.amazonaws.com/groups/global/AllUsers</a>	-	-
Authenticated users group (anyone with an AWS account) Group: <a href="#">http://acs.amazonaws.com/groups/global/AuthenticatedUsers</a>	-	-
S3 log delivery group Group: <a href="#">http://acs.amazonaws.com/groups/s3/LogDelivery</a>	-	-

Below the table, there is a section titled 'Cross-origin resource sharing (CORS)' with the following details:

Cross-origin resource sharing (CORS)  
The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. Learn more [\[?\]](#)

# Private bucket permission:

The screenshot shows the AWS S3 console interface for a bucket named 'tirthshah'. The 'Permissions' tab is selected. Under 'Permissions overview', it states 'Bucket and objects not public'. In the 'Block public access (bucket settings)' section, 'Block all public access' is set to 'On'. A note indicates that public access is blocked because Block Public Access settings are turned on. Under 'Bucket policy', there is a note stating 'Public access is blocked because Block Public Access settings are turned on for this bucket'. The bottom of the page includes standard AWS footer links.

# Private access control list:

The screenshot shows the AWS S3 console interface for the same 'tirthshah' bucket. The 'Object Ownership' section shows 'Bucket owner enforced'. In the 'Access control list (ACL)' section, it notes that the bucket owner has enforced settings applied for object ownership. The ACL table lists four entries:

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: 0a35055fdf6b0ad21c455d82b54368860e5ae704b654fd5aa5722ce046ce5	List, Write	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	-	-
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-
S3 log delivery group Group: http://acs.amazonaws.com/groups/s3/LogDelivery	-	-

At the bottom, the 'Cross-origin resource sharing (CORS)' section is present but empty. The page includes standard AWS footer links.

## Private bucket website:

The screenshot shows a web browser window with the URL <https://tirthshah.s3.ap-south-1.amazonaws.com/tirth.html>. The page displays an XML error document indicating access denial.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>S0YVC368A88H5C63</RequestId>
<HostId>sPwQ7QVog1Gj3S2+JorrQ+OHFwxImpjuMNvqc3zsH9YEGzdzLk2d8AvQUGZMGKF3Kh556xOU9TvJO/FkcUAfQA==</HostId>
</Error>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

# **Course Code and Name: CSI0805 Cloud Computing**

## **PRACTICAL 8**

**Aim:** To understand the PAAS model of the IBM Cloud and run a web application onto the already available readymade environment (with all the necessary available platforms).

### **AWS elastic beanstalk:**

Amazon Elastic Beanstalk service offered by AWS that makes it easy to deploy, run, and scale web applications and services. It supports several programming languages, including Java, .NET, PHP, Node.js, Python, Ruby, and Go. With Elastic Beanstalk, we simply upload our application, and the service takes care of provisioning the resources and the infrastructure required to run our application. It also automatically handles tasks such as capacity provisioning, load balancing, automatic scaling, and application health monitoring. Elastic Beanstalk is a great option for those who want to focus on writing code and don't want to spend time managing infrastructure. Basically, Beanstalk is much recommended instead of using EC2, in which we have to select all the configurations.

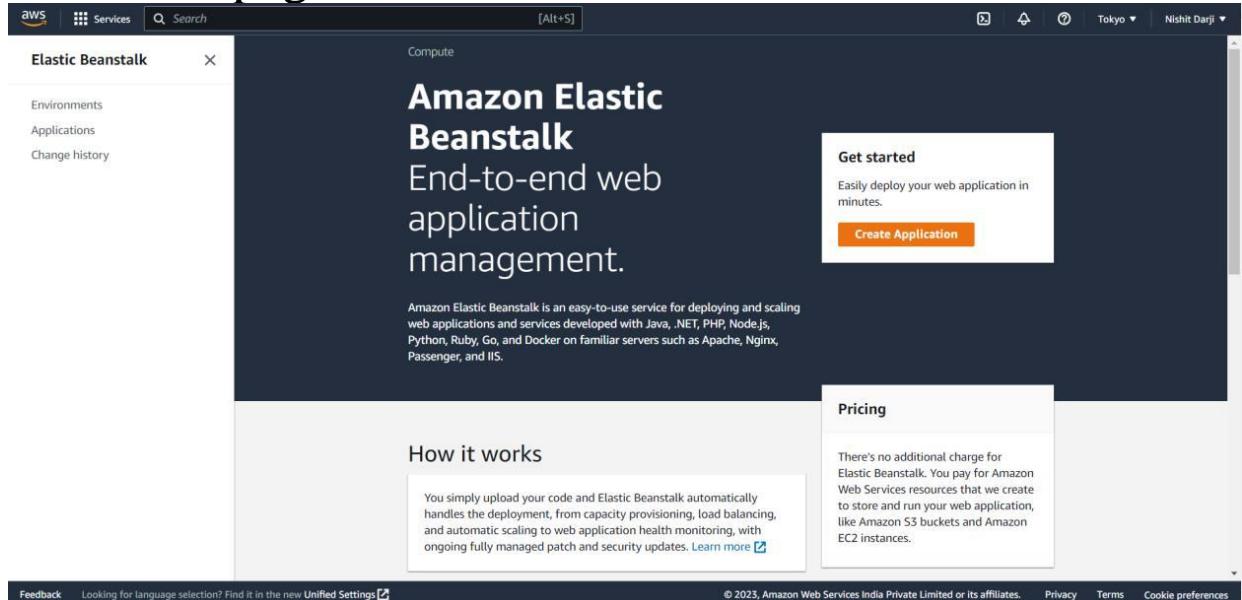
Beanstalk provides several features. Some of them are listed below:

- ***Environment management:*** Elastic Beanstalk allows you to create and manage multiple environments for your application, such as development, staging, and production. You can easily switch between environments and make updates without affecting other environments.

- **Automated deployment:** Elastic Beanstalk supports several methods for deploying your application, including uploading a .zip file or using Git or GitHub. You can also configure automatic deployments whenever you push code to a specific branch in your repository.
- **Scaling:** Elastic Beanstalk automatically scales your application based on demand, so you don't have to manually provision and manage the resources required to run your application.
- **Monitoring and logging:** Elastic Beanstalk provides detailed metrics and logs for your application, so you can monitor its performance and troubleshoot issues.
- **Integration with other AWS services:** Elastic Beanstalk integrates with other AWS services, such as RDS, S3, and CloudWatch. This allows you to use the power of these services to enhance the functionality of your application.

## Screenshots:

Beanstalk page:



# Create a web app:

Elastic Beanstalk

Environments Applications Change history

Elastic Beanstalk > Getting started

## Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. [Learn more](#)

### Application information

Application name

MyApplication

Up to 100 Unicode characters, not including forward slash (/).

### Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value

Add tag Remove tag

50 remaining

### Platform

Platform

Java

Platform branch

Corretto 17 running on 64bit Amazon Linux 2

Platform version

3.4.3 (Recommended)

### Application code

Sample application  
Get started right away with sample code.

Upload your code  
Upload a source bundle from your computer or copy one from Amazon S3.

Cancel Configure more options Create application

**i** Creating Myapplication-env  
This will take a few minutes. ...

```
10:19pm Successfully launched environment: Myapplication-env
10:19pm Application available at Myapplication-env.eba-b7vp9pak.ap-northeast-1.elasticbeanstalk.com.
10:19pm Instance deployment completed successfully.
10:19pm Instance deployment successfully used commands in the 'Procfile' to start your application.
10:19pm Instance deployment successfully built your application using commands in the 'Buildfile'.
10:18pm Added instance [i-0f6507b702c0de79b] to your environment.
10:17pm Created Load Balancer listener named:
arn:aws:elasticloadbalancing:ap-northeast-1:644518341255:listener/app/awseb-AWSEB-1QLLK6WNT185J/3de1f0f34deb80b6/c47aa9713d8483a0
10:17pm Created load balancer named:
arn:aws:elasticloadbalancing:ap-northeast-1:644518341255:loadbalancer/app/awseb-AWSEB-1QLLK6WNT185J/3de1f0f34deb80b6
10:16pm Created CloudWatch alarm named:
awseb-e-cu7jmanyi-stack-AWSEBCloudwatchAlarmHigh-9ZJK4UD4UN4T
10:16pm Created CloudWatch alarm named:
awseb-e-cu7jmanyi-stack-AWSEBCloudwatchAlarmLow-1BALOOL0KMMIK
10:16pm Created Auto Scaling group policy named:
arn:aws:autoscaling:ap-northeast-1:644518341255:scalingPolicy:34c17027-50b3-462a-9ff-3296a7337dd3:autoScalingGroupName/awseb-e-cu7jmanyi-stack-AWSEBAutoScalingGroup-
```

**Myapplication-env**

Myapplication-env.eba-b7vp9pak.ap-northeast-1.elasticbeanstalk.com  (e-cu7jmanyi)  
Application name: **MyApplication**

 Refresh

Actions ▾



Ok

 Causes**Running version**

Sample Application

 Upload and deploy**Platform**Corretto 17 running on 64bit  
Amazon Linux 2/3.4.3 Change**Recent events** Show all

&lt; 1 &gt;

Time	Type	Details
2023-01-28 22:19:46 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 15 seconds ago and took 5 minutes.
2023-01-28 22:19:28 UTC+0530	INFO	Successfully launched environment: Myapplication-env
2023-01-28 22:19:28 UTC+0530	INFO	Application available at Myapplication-env.eba-b7vp9pak.ap-northeast-1.elasticbeanstalk.com.
2023-01-28 22:19:12 UTC+0530	INFO	Instance deployment completed successfully.
2023-01-28 22:19:09 UTC+0530	INFO	Instance deployment successfully built your application using commands in the 'Buildfile'.

# Configurations:

Software	Environment properties: GRADLE_HOME, M2, M2_HOME, PORT Log streaming: disabled Rotate logs: disabled X-Ray daemon: disabled
Instances	EC2 security groups: awseb-e-cu7jmanyi-stack-AWSEBSecurityGroup-NSCDF4DX015H IMDSv1: disabled IOPS: container default Monitoring interval: 5 minute Root volume type: container default Size: container default Throughput: container default
Capacity	AMI ID: ami-0f00fdb2c7f257839 Availability Zones: Any Breach duration: 5 Capacity rebalancing: disabled Environment type: load balancing, auto scaling Instance types: t2.micro,t2.small Lower threshold: 2000000 Max: 4 Metric: NetworkOut Min: 1 Period: 5 Placement: Scale down increment: -1 Scale up increment: 1 Scaling cooldown: 360 seconds Statistic: Average Unit: Bytes Upper threshold: 6000000
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: -- IAM instance profile: aws-elasticbeanstalk-ec2-role Service role: arn:aws:iam::644518341255: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

## 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](#)

[Request Logs ▾](#) [Refresh](#)

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

## Health status:

Elastic Beanstalk > Environments > Myapplication-env > Health

**Enhanced health overview**

Instances: 1 Total, 1 Ok

[Learn more](#) about enhanced health.

Auto refresh (12 s)

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses
Overall	Ok	N/A	N/A	0.1	100%
i-0f6507b702c0de79b	Ok	7 minutes	1	0.1	1

## Monitoring:

**Overview**

Period [1 hour ▾](#) [Edit](#) [C](#)

<b>0.9</b> Healthy Host Count	<b>3.0</b> Target Response Time <i>in milliseconds</i>	<b>2.0</b> Sum Requests	<b>10.7%</b> CPU Utilization	<b>8MB</b> Max Network In	<b>288KB</b> Max Network Out
----------------------------------	--	----------------------------	---------------------------------	------------------------------	---------------------------------

**Monitoring**

Time Range [3 hours ▾](#) [Period](#) [1 minute ▾](#) [Edit](#) [C](#)

Environment Health by health codes

Info

Ok

1/28 22:14 1/28 22:15 1/28 22:16 1/28 22:17 1/28 22:18 1/28 22:19 1/28 22:20 1/28 22:21 1/28 22:22

Target Response Time in seconds

0.0040  
0.0030  
0.0020  
0.0010  
0.0000

1/28 22:21

Sum Requests by count

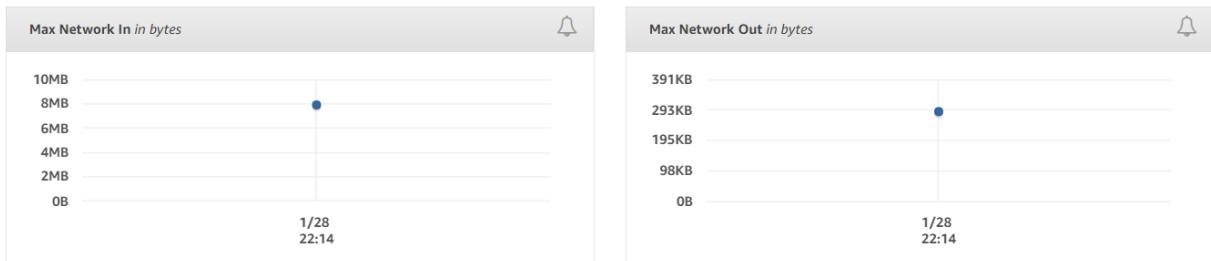
2.5  
2.0  
1.5  
1.0  
0.5  
0.0

1/28 22:17 1/28 22:18 1/28 22:19 1/28 22:20 1/28 22:21 1/28 22:22

CPU Utilization in percent

12.5  
10.0  
7.5  
5.0  
2.5  
0.0

1/28 22:14



## Events:

Events			
Severity	Time	Type	Details
TRACE	2023-01-28 22:19:46 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 15 seconds ago and took 5 minutes.
	2023-01-28 22:19:28 UTC+0530	INFO	Successfully launched environment: Myapplication-env
	2023-01-28 22:19:28 UTC+0530	INFO	Application available at Myapplication-env.eba-b7vp9pak.ap-northeast-1.elasticbeanstalk.com.
	2023-01-28 22:19:12 UTC+0530	INFO	Instance deployment completed successfully.
	2023-01-28 22:19:09 UTC+0530	INFO	Instance deployment successfully built your application using commands in the 'Buildfile'.

## Application version:

Application versions					
	Version label	Description	Date created	Source	Deployed to
<input type="checkbox"/>	Sample Application		2023-01-28T22:14:28+05:30	Sample Application	Myapplication-env

## All application:

All applications				
<input type="checkbox"/> Filter results matching the display values				
Application name	Environments	Date created	Last modified	ARN
<input type="radio"/> MyApplication	Myapplication-env	2023-01-28 22:14:01 UTC+0530	2023-01-28 22:14:01 UTC+0530	arn:aws:elasticbeanstalk:ap-northeast-1:644518341255:application/MyApplication

## Status:

Application 'MyApplication' environments										<a href="#">Create a new environment</a>
Environment name	Health	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name		
Myapplication-env	<span>Ok</span>	2023-01-28 22:14:30 UTC+0530	2023-01-28 22:19:28 UTC+0530	Myapplication-env.eba-b7vp9pak.ap-northeast-1.elasticbeanstalk.com	Sample Application	Corretto 17 running on 64bit Amazon Linux 2	<span>Supported</span>	WebServer		

## Webapp:

The screenshot shows a web browser window displaying the AWS Elastic Beanstalk 'Congratulations' page. The page is split into two main sections: a light blue left section and a dark blue right section.

**Left Section (Light Blue):**

- Congratulations** (Large, bold, blue text)
- Your first AWS Elastic Beanstalk Corretto application is now running on your own dedicated environment in the AWS Cloud (Text)
- This environment is launched with Elastic Beanstalk Corretto Platform (Text)

**Right Section (Dark Blue):**

- What's Next?** (Section title)
- [AWS Elastic Beanstalk overview](#)
  - [AWS Elastic Beanstalk concepts](#)

**Course Code and Name:** CSI0805 Cloud Computing

## **PRACTICAL 9**

**Aim:** To work with the IAM (Identity Access Management) of the AWS (Amazon Web Services) Cloud and to enables the access management mechanism in AWS for Cloud security. Also, create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

### **Identity and access management:**

AWS Identity and Access Management (IAM) is a web service that helps us securely control access to AWS resources. With IAM, we can centrally manage permissions that control which AWS resources users can access. We use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Whenever we create an AWS account, we begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that we used to create the account. But AWS strongly recommend that we should not use the root user for our everyday tasks. We should safeguard our root user credentials and use them to perform the tasks that only the root user can perform.

Below are the benefits of using IAM:

- Security: IAM provides secure access control to AWS resources, helping to reduce the risk of unauthorized access to sensitive data.

- Scalability: IAM enables us to manage access for a growing number of AWS users and resources, making it easy to scale our security as our organization grows.

3 | Page

- Flexibility: IAM allows us to define and apply fine-grained permissions for AWS resources, giving us the flexibility to control who can access what.
- Compliance: IAM helps us meet compliance requirements by providing a centralized way to manage access and track changes to AWS resources.
- Cost Savings: IAM helps us avoid unnecessary costs by allowing us to easily revoke access to unused AWS resources, reducing the risk of unintended charges.
- Improved Collaboration: IAM enables us to share AWS resources with other users and organizations, making it easier to collaborate on projects and scale our work.
- Integration: IAM integrates with other AWS services, allowing us to manage access to resources across our entire AWS infrastructure.

## Screenshots:

### IAM dashboard:

The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with navigation links like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Access reports', and 'Create a new user'. The main area has sections for 'Security recommendations' (with items like 'Add MFA for root user', 'Root user has no active access keys', and 'Update your access permissions for AWS Billing, Cost Management, and Account consoles'), 'IAM resources' (showing 0 User groups, 0 Users, 8 Roles, 0 Policies, and 0 Identity providers), and 'AWS Account' (with details like Account ID: 644518341255, Account Alias: 644518341255, and Sign-in URL: https://644518341255.signin.aws.amazon.com/console). There's also a 'Quick Links' section for 'My security credentials' and 'Tools' for 'Policy simulator'.

### Create a new user:

The screenshot shows the 'Create a new user' wizard. It starts with a 'User details' step where you can enter the user name (e.g., 'User-1'). You can enable console access (checked by default) and choose between autogenerated or custom passwords. A password field ('User1aws@73') includes validation rules: at least 8 characters, three character types (uppercase, lowercase, numbers, symbols), and a hyphen. You can also show the password. Below this, there's a note about users creating their own password and a link to learn more about programmatic access.

## Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

**Permissions options**

- Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1/1036)**  
Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	0

## Review:

### Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

**User details**

User name	Console password type	Require password reset
User-1	Custom password	No

**Permissions summary**

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

## User created successfully:

**User created successfully**  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

Step 4: Retrieve password

**Retrieve password**

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

**Console sign-in details**

Console sign-in URL: <https://644518341255.signin.aws.amazon.com/console>

Email sign-in instructions

User name: User-1

Console password: \*\*\*\*\* Show

Download .csv file | Return to users list

# Signing in as a user:



## Sign in as IAM user

Account ID (12 digits) or account alias

644518341255

IAM user name

User-1

Password

\*\*\*\*\*

Remember this account

**Sign in**

[Sign in using root user email](#)

[Forgot password?](#)

## SageMaker Fridays

Join SageMaker Fridays for live coding, demos, and more

[Register now >](#)



English ▾

[Terms of Use](#) [Privacy Policy](#) © 1996-2023, Amazon Web Services, Inc. or its affiliates.

AWS Services Search [Alt+S] Tokyo User-1 @ 6445-1834-1255 ⓘ

Console Home [Info](#) [Reset to default layout](#) [+ Add widgets](#)

① Introducing the new Managed instances, Ops summary, and Patch compliance widgets. Find them at the bottom of your Console Home or click Add widgets to access the widget library. X

Recently visited [Info](#)

- S3
- EC2
- Elastic Beanstalk

[View all services](#)

Welcome to AWS

Getting started with AWS [Get Started](#)  
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification [Get Certified](#)  
Learn from AWS experts and advance your skills and knowledge.

What's new with AWS? [Check it out](#)  
Discover new AWS services, features, and Regions.

# IAM resources:



## Creating user groups:

IAM > User groups

**User groups (0) Info**  
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Filter User groups by property or group name and press enter

Group name	Users	Permissions	Creation time
No resources to display			

IAM > User groups > Create user group

### Create user group

#### Name the group

User group name  
Enter a meaningful name to identify this group.

Maximum 128 characters. Use alphanumeric and '+-, @\_-' characters.

#### Add users to the group - *Optional* (Selected 1/1) Info

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS. A user can belong to up to 10 groups.

User-1

#### Attach permissions policies - *Optional* (Selected 1/808) Info

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

9 matches

< 1 >

AWS managed

Provides access to manag...

AWS managed

Provides full access to all...

AWS managed

Policy used by QuickSigh...

AWS managed

Provides read only acces...

AWS managed

Provides full access to An...

## Group created:

The screenshot shows the 'User groups' page in the AWS IAM console. A green header bar at the top indicates that a group has been created. The main area displays a table of user groups. The first row in the table is 'FirstGroup', which has 1 user assigned and 1 attached policy ('AmazonS3FullAccess').

Group name	Users	Permissions	Creation time
FirstGroup	1	Defined	2023-01-29 (2 minutes ago)

## Creating another user and adding in group:

The screenshot shows the 'User details' page for creating a new user. The 'User name' field is set to 'User-2'. Under 'Console password', the 'Custom password' option is selected, and the password 'User2aws@73' is entered. The 'Show password' checkbox is checked. The 'Users must create a new password at next sign-in (recommended)' checkbox is unchecked. A note at the bottom states that users automatically get the IAMUserChangePassword policy.

## Adding in group:

### Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

The screenshot shows the 'Permissions options' section where the 'Add user to group' option is selected. Below it, the 'User groups' table shows the 'FirstGroup' assigned to the user 'User-2'. The table includes columns for Group name, Users, Attached policies, and Created.

Group name	Users	Attached policies	Created
FirstGroup	1	AmazonS3FullAccess	2023-01-29 (2 minutes ago)

# Review:

## Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details		
User name	Console password type	Require password reset
User-2	Custom password	No

Permissions summary			
Name	Type	Used as	
FirstGroup	Group	Permissions group	

**⌚ User created successfully**  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1  
Specify user details

Step 2  
Set permissions

Step 3  
Review and create

Step 4  
Retrieve password

**Retrieve password**

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details	
Console sign-in URL	<a href="https://644518341255.signin.aws.amazon.com/console">https://644518341255.signin.aws.amazon.com/console</a>
User name	User-2
Console password	***** <a href="#">Show</a>

[Email sign-in instructions](#)

[Download .csv file](#) [Return to users list](#)

# Group updated:

IAM > User groups > FirstGroup

FirstGroup

[Delete](#)

**Summary**

[Edit](#)

User group name	Creation time	ARN
FirstGroup	January 29, 2023, 18:12 (UTC+05:30)	<a href="#">arn:aws:iam::644518341255:group/FirstGroup</a>

[Users](#)

[Permissions](#)

[Access Advisor](#)

### Users in this group (2)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.



[Remove users](#)

[Add users](#)

[Search](#)

[<](#) [1](#) [>](#)



	User name	Groups	Last activity	Creation time
<input type="checkbox"/>	User-2	1	None	Now
<input type="checkbox"/>	User-1	1	10 minutes ago	13 minutes ago

# Checking if user can access ec2:

The screenshot shows the AWS EC2 console with the following details:

- Resources:** Instances (running), Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, Volumes. All items show an "API Error".
- Account attributes:** Supported platforms, An error occurred (An error occurred retrieving supported platforms), An error occurred (An error occurred checking for a default VPC).
- Launch instance:** To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
- Service health:** AWS Health Dashboard.

No

# User can access s3:

The screenshot shows the AWS S3 console with the following details:

- Buckets:** Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3.
- Block Public Access settings for this account.**
- Storage Lens:** Dashboards, AWS Organizations settings.
- Feature spotlight:**
- AWS Marketplace for S3.**
- Amazon S3 > Buckets:**
  - Account snapshot:** Last updated: Jan 28, 2023 by Storage Lens. Metrics are generated every 24 hours. View Storage Lens dashboard.
  - Buckets (3):** elasticbeanstalk-ap-northeast-1-644518341255, firstbucket952, newbucket54.