

**National University of Computer & Emerging
Sciences**



**Cloud Computing
Project**

21i-0572 Kissa Zahra

Submission Date: 11/05/2025

Repository link:	2
Live URL	2
Project	2
Architecture diagram of AWS deployment	4
CURD OPERATIONS	6
Create	6
Update	8
Read	9
Delete	11
IAM policy Screenshots	13
EBS	15
EC2	16
RDS	18
S3	19
Security Group	20
AWS deployment and configurations	21
Backend	22
Frontend	24

Repository link:

<https://github.com/kissasium/CloupComputingProject.git>

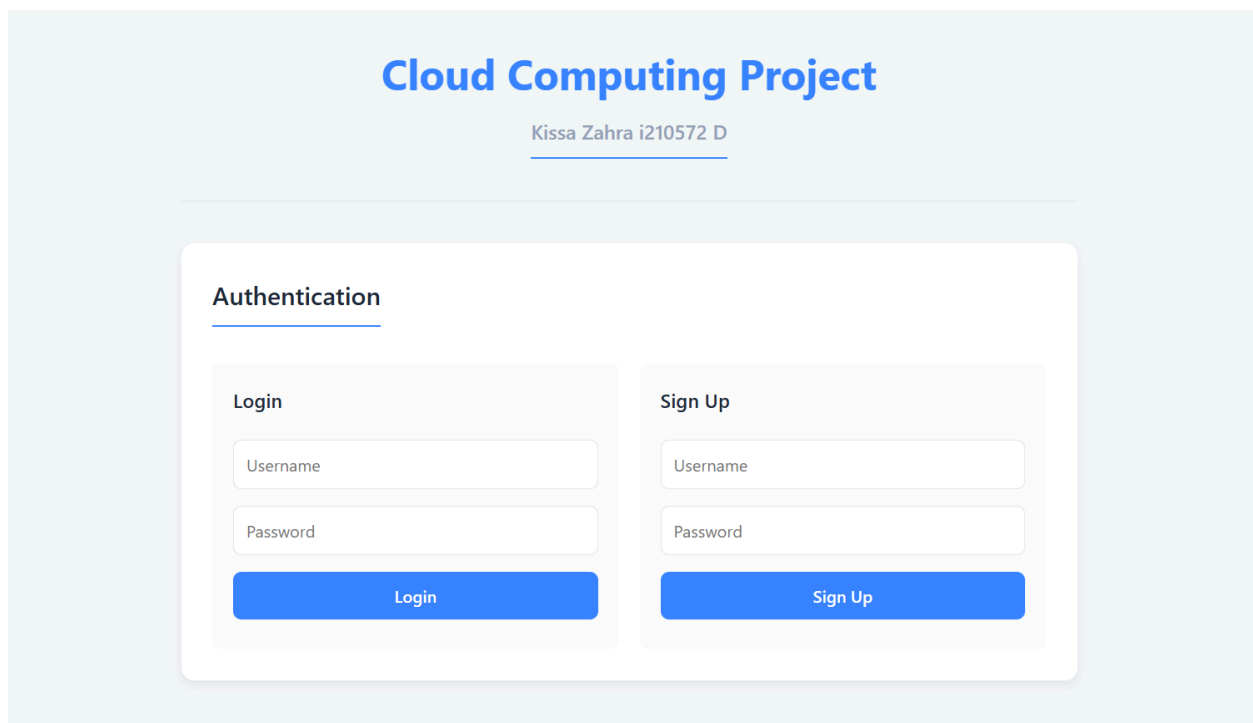
Live URL

Frontend: <http://kissaproject0572.eba-hqs3ana7.ap-southeast-2.elasticbeanstalk.com/>

Backend: <http://3.105.45.13:5000>

"curl http://3.105.45.13:5000"

Project



The screenshot displays the 'Cloud Computing Project' authentication interface. At the top, the title 'Cloud Computing Project' is centered in blue, with the identifier 'Kissa Zahra i210572 D' below it. The main section is titled 'Authentication' and contains two side-by-side forms: 'Login' and 'Sign Up'. Each form has input fields for 'Username' and 'Password', and a corresponding blue button ('Login' or 'Sign Up').

Cloud Computing Project

Kissa Zahra i210572 D

Authentication

Login

Username

Password

Login

Sign Up

Username

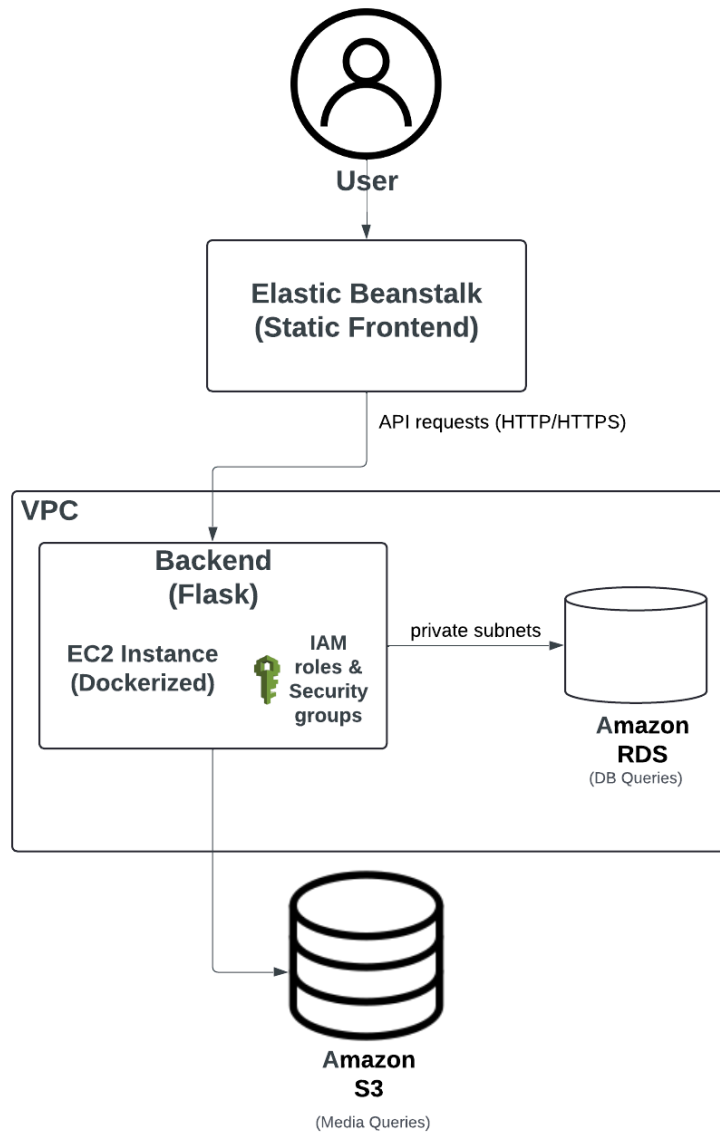
Password

Sign Up

The screenshot displays a web application interface with two main sections. The top section, titled 'Your Items', features a red 'Logout' button in the upper right corner. Below the title is a large text input field containing the placeholder text 'New item content'. To the right of this field is a green 'Add Item' button. At the bottom of this section, a message reads 'No items found. Add your first item above!'. The bottom section, titled 'Upload File', contains a file selection interface with a 'Choose file' button and the text 'No file chosen'. Below this is a prominent blue 'Upload' button.

This project involves deploying a lightweight web application on AWS using a Flask backend and a static HTML/CSS/JavaScript frontend. The backend is containerized and hosted on Amazon EC2 within a VPC, while the frontend is served via Elastic Beanstalk. Amazon RDS (PostgreSQL) handles database operations, and Amazon S3 is used for storing uploaded images. The setup follows AWS best practices for secure, scalable, and modular deployment.

Architecture diagram of AWS deployment



This architecture diagram represents the deployment of a simple web application on AWS using a static HTML frontend and a Flask backend. The design prioritizes simplicity, scalability and security using essential AWS services.

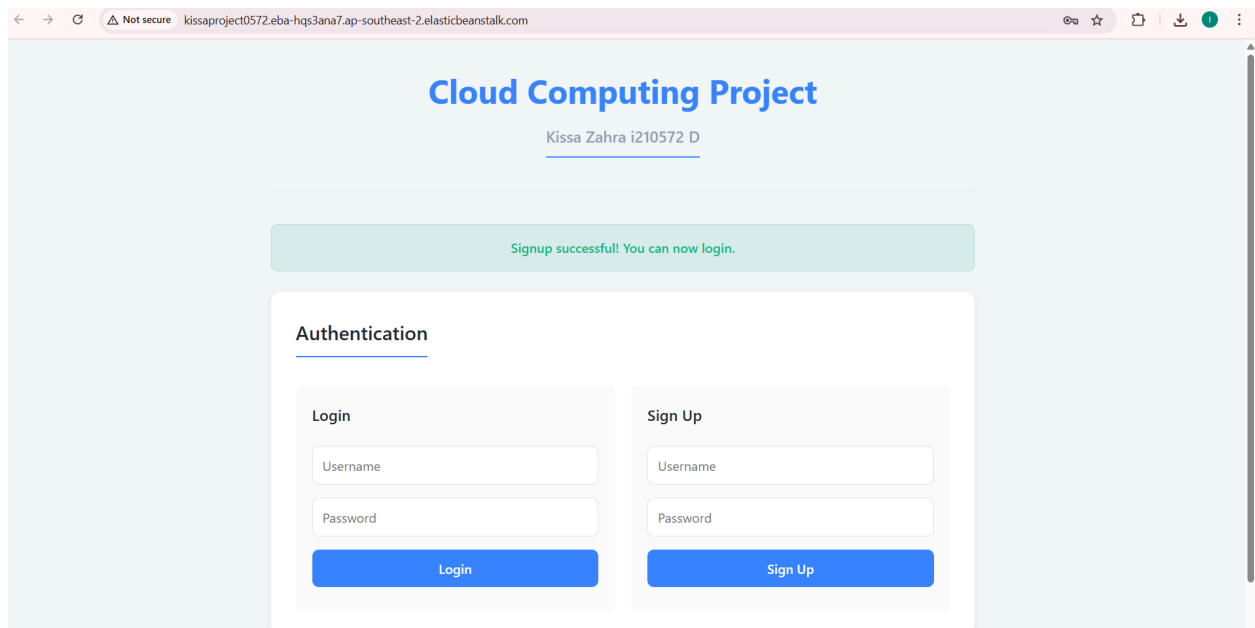
1. The frontend (a static index.html file) is deployed on AWS Elastic Beanstalk which provides managed hosting for the web interface with automatic provisioning and load balancing.
2. The backend is a Flask application (app.py) running inside a Docker container on an Amazon EC2 instance. This EC2 instance is part of a Virtual Private Cloud (VPC).
3. The EC2 instance has:
 - An IAM Role attached to securely access AWS services like S3 and RDS.
 - A Security Group that controls inbound and outbound traffic to protect the instance.
 - An Elastic Block Store (EBS) volume attached to persist data and container storage.
4. Amazon RDS is deployed within the same VPC to store application data securely and is accessible only to the EC2 instance over the private network.
5. Amazon S3 is used to store uploaded files.

Flow Direction

User -> Elastic Beanstalk -> EC2 -> RDS/S3

CURD OPERATIONS

Create



The screenshot shows a web browser window with the address bar displaying "kissaproject0572.eba-hqs3ana7.ap-southeast-2.elasticbeanstalk.com". The page title is "Cloud Computing Project" and the user is logged in as "Kissa Zahra i210572 D". A green message box states "Signup successful! You can now login." Below this, the "Authentication" section contains two forms: "Login" and "Sign Up". Both forms have "Username" and "Password" input fields and a corresponding action button ("Login" or "Sign Up").

Cloud Computing Project

Kissa Zahra i210572 D

Signup successful! You can now login.

Authentication

Login

Username

Password

Login

Sign Up

Username

Password

Sign Up

Login successfully!!

Logout

Your Items

New item content

Add Item

Hi my name is kissa

Edit

Delete

Upload File

Choose file No file chosen

Update

[Logout](#)

Your Items

Add Item

Hi my name is kissa

[Edit](#) [Delete](#)

Upload File

Choose file

No file chosen

[Logout](#)

Your Items

Add Item

[Save](#) [Cancel](#)

Upload File

Your Items

New item content

Add Item

Hi my name is kissa, heheheh

Edit

Delete

Upload File

Choose file No file chosen

Read

Logout

Your Items

New item content

Add Item

Hi my name is kissa, heheheh

Save

Cancel

Upload File

Read anything!!

All files are stored in S3 and all the textual data is stored in RDS

Delete

Your Items

New item content

Add Item

Hi my name is kissa, heheheh

Edit

Delete

hehhdhds

Edit

Delete

Upload File

Logout

Your Items

New item content

Add Item

hehhdhds

Edit

Delete

Upload File

Choose file

pic4.png

Deleted successfully!

IAM policy Screenshots

IAM user Creation

Name: kissa0572

Password: kissa-0572

☰ IAM > Users > Create user ⓘ 🔍

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Step 4

Retrieve password

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

ⓘ Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Giving this user permissions (Attach and managed policies for that user):

☰ IAM > Users > Create user ⓘ 🔍

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Step 4

Retrieve password

User name

kissa0572

Console password type

Custom password

Require password reset

Yes

Permissions summary

< 1 >

Name ⓘ	Type	Used as
AdministratorAccess-AWSElasticBeanstalk	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy
IAMFullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

CloudShell

Feedback

IAM > Users > kissa0572

Info

Delete

Summary

ARN

arn:aws:iam::381492186725:user:kissa0572

Console access

Enabled without MFA

Access key 1

Create access key

Created

May 11, 2025, 18:13 (UTC+05:00)

Last console sign-in

Never

Permissions

Groups

Tags

Security credentials

Last Accessed

Permissions policies (5)

Remove

Add permissions

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type

IAM > Users > kissa0572 > Create access key

Info

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code

You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Third-party service

IAM > Users > kissa0572

Info

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

CloudShell

Feedback

Access keys (1)

Create access key

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

AKIAVRUVUZZS4VVC6A3C

Actions

Description

kissakey0572

Status

Active

Last used

None

Created

Now

Last used region

N/A

Last used service

N/A

SSH public keys for AWS CodeCommit (0)

Actions

Upload SSH public key

User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys

14

Creating an access key for my IAM user that i will put in my .env file to access S3 and RDS

EBS

Installing Elastic Beanstalk CLI so we can use elastic beanstalk to deploy our frontend

```
Select Administrator: Command Prompt

*****
4. Installing EBCLI
*****
Collecting awsebcli
  Using cached awsebcli-3.23.3-py3-none-any.whl
Collecting boto3<2,>=1.35.0 (from awsebcli)
  Using cached boto3-1.38.13-py3-none-any.whl.metadata (5.7 kB)
Collecting cements<2.10.14 (from awsebcli)
  Using cached cements-2.10.14-py3-none-any.whl.metadata (3.5 kB)
Collecting colorama<0.5,>=0.4.6 (from awsebcli)
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting pathspec<0.12.1 (from awsebcli)
  Using cached pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
Collecting python-dateutil<3.0.0,>=2.1 (from awsebcli)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting requests<3,>=2.31 (from awsebcli)
  Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting setuptools<20.0 (from awsebcli)
  Using cached setuptools-80.4.0-py3-none-any.whl.metadata (6.5 kB)
Collecting semantic_version<2.11,>=2.10.0 (from awsebcli)
  Using cached semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting termcolor<3,>=2.4.0 (from awsebcli)
  Using cached termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
Collecting wcwidth<0.3,>=0.2.13 (from awsebcli)
  Using cached wcwidth-0.2.13-py2.py3-none-any.whl.metadata (14 kB)
Collecting PyYAML<6.1,>=5.3.1 (from awsebcli)
  Using cached PyYAML-6.0.2-cp312-cp312-win_amd64.whl.metadata (2.1 kB)
Collecting urllib3<2,>=1.26.5 (from awsebcli)
  Using cached urllib3-1.26.20-py2.py3-none-any.whl.metadata (50 kB)
Collecting packaging<25.0,>=24.2 (from awsebcli)
  Using cached packaging-24.2-py3-none-any.whl.metadata (3.2 kB)
Collecting blessed<=1.20.0 (from awsebcli)
  Using cached blessed-1.21.0-py2.py3-none-any.whl.metadata (13 kB)
Collecting pythonnet<4,>=3.0.5 (from awsebcli)
  Using cached pythonnet-3.0.5-py3-none-any.whl.metadata (6.6 kB)
Collecting pypiwin32==223 (from awsebcli)
  Using cached pypiwin32-223-py3-none-any.whl.metadata (236 bytes)
Collecting pywin32>=223 (from pypiwin32==223->awsebcli)
  Using cached pywin32-2.5.0-py3-none-any.whl.metadata (6.1 kB)
```

The screenshot shows the AWS Elastic Beanstalk console's 'Configure environment' wizard. On the left, a sidebar lists six steps: Step 1: Configure environment (selected), Step 2: Configure service access, Step 3: optional: Set up networking, database, and tags, Step 4: optional: Configure instance traffic and scaling, Step 5: optional: Configure updates, monitoring, and logging, and Step 6: Review. The main content area is titled 'Configure environment' and contains three sections: 'Environment tier' with 'Web server environment' selected, 'Application information' with 'Application name' set to 'kissafontend0572', and 'Environment information' with 'Environment name' set to 'Kissafontend0572-env'. The bottom of the screen shows the 'CloudShell' interface with a 'Feedback' button and copyright information for Amazon Web Services.

Step 1

Configure environment

Step 2

Configure service access

Step 3 - optional

Set up networking, database, and tags

Step 4 - optional

Configure instance traffic and scaling

Step 5 - optional

Configure updates, monitoring, and logging

Step 6

Review

Configure service access

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

☐ Create and use new service role
☒ Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

kissakeypair0572

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

View permission details

Cancel

Skip to review

Previous

Next

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

EC2

EC2

Instances

Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name

kissabackend0572

Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...[read more](#)

ami-0822a7a2356687b0f

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750

Cancel

Launch instance

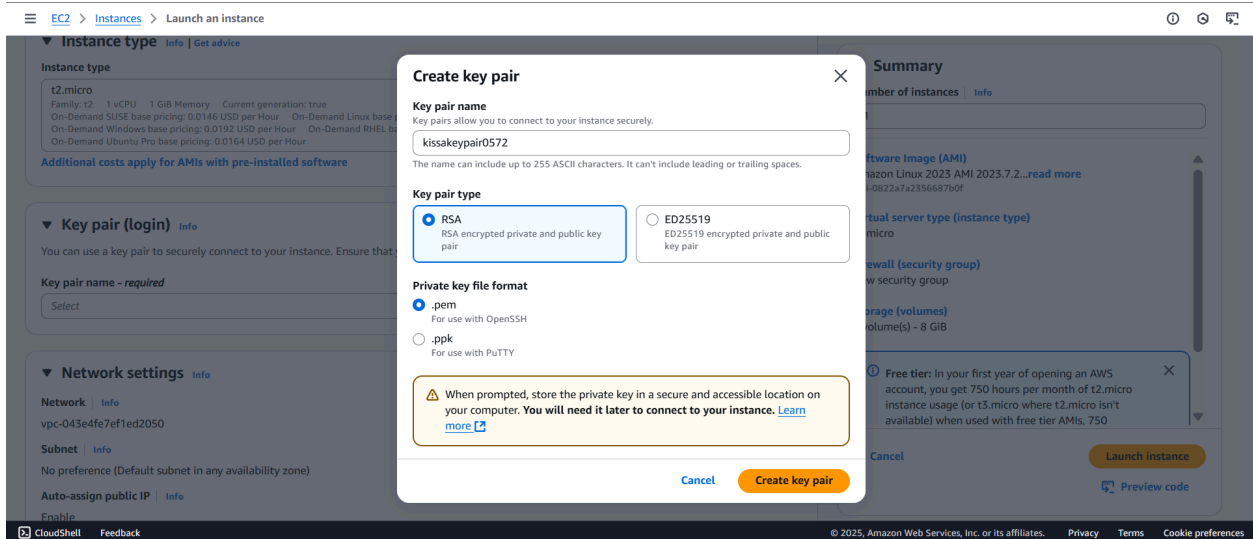
Preview code

CloudShell

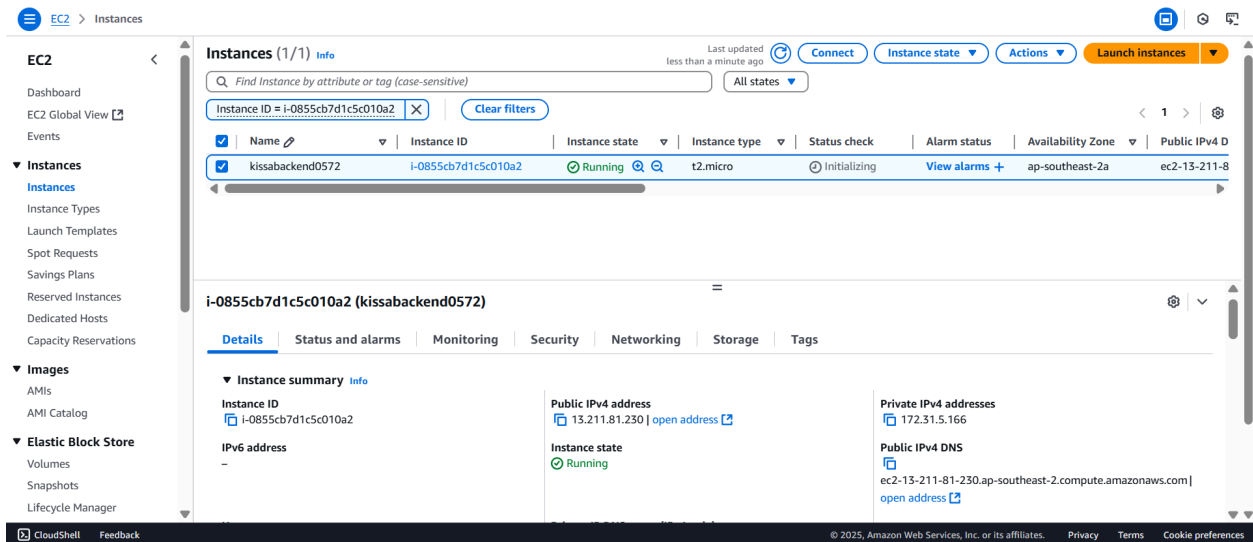
Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

16



Generating a key pair. Later we will use the key pair .pem file to ssh inside the instance terminal, so we can deploy our containerised application.



Creating an EC2 instance to deploy our backend.

RDS

[Aurora and RDS](#) > Create database

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

kissadb0572

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

☐ **Managed in AWS Secrets Manager - most secure**
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ **Self managed**
Create your own password or have RDS create a password that you manage.

☐ **Auto generate password**
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Password strength [Weak](#)

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / * @

[Confirm master password](#) [Info](#)

PostgreSQL >

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- The most Oracle-compatible open-source database.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Choose Postgresql

[Aurora and RDS](#) > Databases

Aurora and RDS <

Dashboard

Databases

Query editor

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations [New](#)

Events

Event subscriptions

Databases (2)

Group resources [C](#) [Modify](#) [Actions](#) [Create database](#)

Filter by databases

	DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
<input type="radio"/>	db45230603	Available	Instance	PostgreSQL	ap-southe...	db.t4g.micro	
<input type="radio"/>	kissadb0572	Creating	Instance	PostgreSQL	ap-southe...	db.t4g.micro	

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully DB created!!

Aurora and RDS > Databases > kissadb0572

Aurora and RDS

- Dashboard
- Databases**
- Query editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations [New](#)
- Events
- Event subscriptions

Connectivity & security

Endpoint & port

Endpoint
kissadb0572.c7a06co0a7wo.ap-south-east-2.rds.amazonaws.com

Port
5432

Networking

Availability Zone
ap-southeast-2c

VPC
vpc-043e4fe7ef1ed2050

Subnet group
default-vpc-043e4fe7ef1ed2050

Subnets
subnet-03931d815dbf7cae8
subnet-01dcaadc15f2f90d6
subnet-0bcab5d6979a49849

Network type
IPv4

Security

VPC security groups
default (sg-097e045978ab21533)
Active

Publicly accessible
No

Certificate authority
rds-ca-rsa2048-g1

Certificate authority date
May 25, 2061, 02:42 (UTC+05:00)

DB instance certificate expiration date
May 11, 2026, 18:18 (UTC+05:00)

Connected compute resources (0)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Created url

S3

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Sydney) ap-southeast-2

Bucket name [Info](#)
kissabucket0572

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

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.

☒ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon S3 > Buckets

Successfully created bucket "kissabucket0572"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Account snapshot - updated every 24 hours All AWS Regions [View Storage Lens dashboard](#)
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. [Learn more](#)

General purpose buckets | Directory buckets

General purpose buckets (3) All AWS Regions [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
bucket45230603	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	May 10, 2025, 17:37:14 (UTC+05:00)
elasticbeanstalk-ap-southeast-2-381492186725	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	May 11, 2025, 16:28:55 (UTC+05:00)
kissabucket0572	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	May 11, 2025, 18:17:56 (UTC+05:00)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Image uploaded from frontend successfully stored in AWS S3 bucket.

Security Group

Security group created, it will be used for the EC2 instance

aws Search [Alt+S]

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

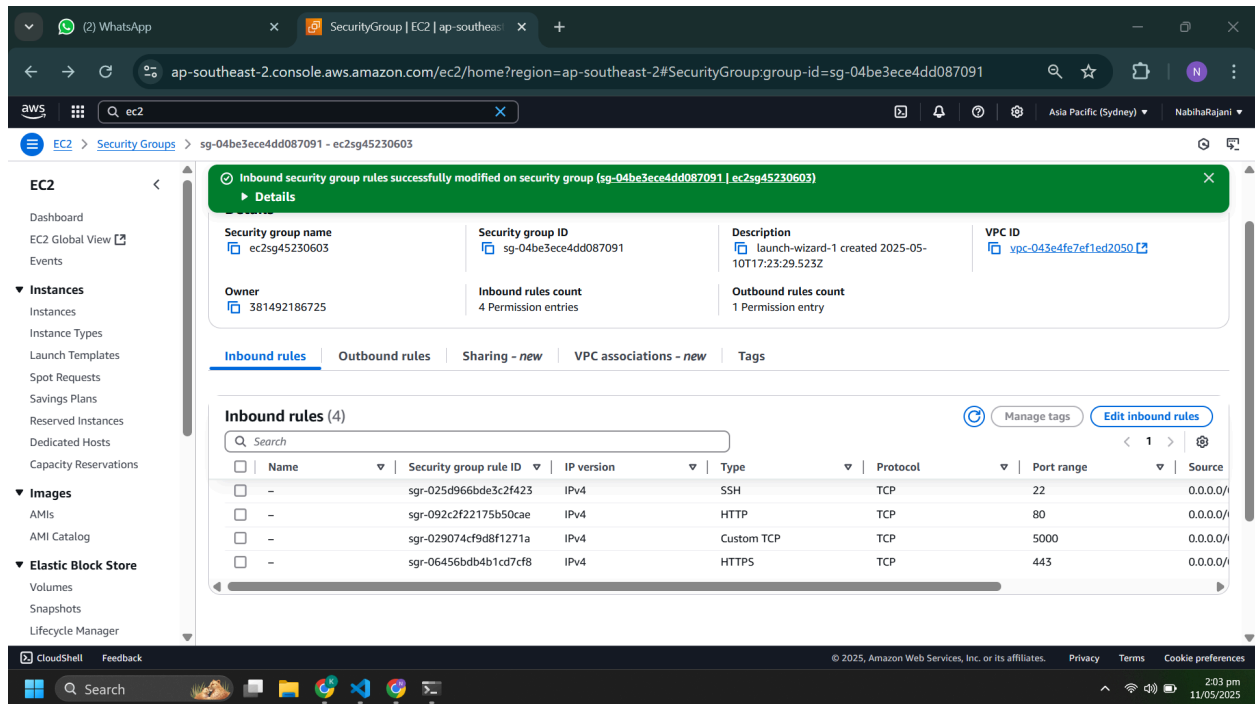
Security group name Info
pgsg45230603
Name cannot be edited after creation.

Description Info
Allows SSH access to developers

VPC Info
vpc-043e4fe7ef1ed2050

Inbound rules Info

Windows taskbar: Search, 8:22 pm, 10/05/2025



Set up **Security Groups** to manage inbound and outbound traffic for EC2 and other AWS services.

Allow traffic on ports 80 (HTTP), 443 (HTTPS), and 5000 (Flask backend). Ensure that only trusted sources can access the EC2 instance and other resources.

AWS deployment and configurations

Backend

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
kissabackend0572 [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...[read more](#)
ami-0822a7a2356687b0f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750

[Cancel](#) [Launch instance](#) [Preview code](#)

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
kissakeypair0572
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

[Cancel](#) [Create key pair](#)

Instance type [Info](#) [Get advice](#)

Instance type
t2.micro
Family: t2, 1 xCPU, 1 GiB Memory, Current generation: true
On-Demand SUSE base pricing: 0.0146 USD per Hour On-Demand Linux base pricing: 0.0146 USD per Hour
On-Demand Windows base pricing: 0.0192 USD per Hour On-Demand RHEL base pricing: 0.0192 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0164 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have the private key file on your computer.

Key pair name - required
Select

Network settings [Info](#)

Network [Info](#)
vpc-043e4fe7ef1ed2050

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...[read more](#)
ami-0822a7a2356687b0f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750

[Cancel](#) [Launch instance](#) [Preview code](#)

Generating a key pair. Later we will use the key pair .pem file to ssh inside the instance terminal, so we can deploy our containerised application.

Frontend

Installing ebcli so we can use elastic beanstalk to deploy our frontend

```
Select Administrator: Command Prompt

*****
4. Installing EBCLI
*****
Collecting awsebcli
  Using cached awsebcli-3.23.3-py3-none-any.whl
Collecting botocore<2,>=1.35.0 (from awsebcli)
  Using cached botocore-1.38.13-py3-none-any.whl.metadata (5.7 kB)
Collecting cement==2.10.14 (from awsebcli)
  Using cached cement-2.10.14-py3-none-any.whl.metadata (3.5 kB)
Collecting colorama<0.5,>=0.4.6 (from awsebcli)
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting pathspec==0.12.1 (from awsebcli)
  Using cached pathspec-0.12.1-py3-none-any.whl.metadata (21 kB)
Collecting python-dateutil<3.0.0,>=2.1 (from awsebcli)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting requests<3,>=2.31 (from awsebcli)
  Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting setuptools==20.0 (from awsebcli)
  Using cached setuptools-80.4.0-py3-none-any.whl.metadata (6.5 kB)
Collecting semantic_version<2.11,>=2.10.0 (from awsebcli)
  Using cached semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting termcolor<3,>=2.4.0 (from awsebcli)
  Using cached termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
Collecting wcwidth<0.3,>=0.2.13 (from awsebcli)
  Using cached wcwidth-0.2.13-py2.py3-none-any.whl.metadata (14 kB)
Collecting PyYAML<6.1,>=5.3.1 (from awsebcli)
  Using cached PyYAML-6.0.2-cp312-cp312-win_amd64.whl.metadata (2.1 kB)
Collecting urllib3<2,>=1.26.5 (from awsebcli)
  Using cached urllib3-1.26.20-py2.py3-none-any.whl.metadata (50 kB)
Collecting packaging<25.0,>=24.2 (from awsebcli)
  Using cached packaging-24.2-py3-none-any.whl.metadata (3.2 kB)
Collecting blessed==1.20.0 (from awsebcli)
  Using cached blessed-1.21.0-py2.py3-none-any.whl.metadata (13 kB)
Collecting pythonnet<4,>=3.0.5 (from awsebcli)
  Using cached pythonnet-3.0.5-py3-none-any.whl.metadata (6.6 kB)
Collecting pypiwin32==223 (from awsebcli)
  Using cached pypiwin32-223-py3-none-any.whl.metadata (236 bytes)
Collecting pywin32>=223 (from pypiwin32==223->awsebcli)
  Using cached termcolor-2.5.0-py3-none-any.whl.metadata (6.1 kB)
```

Step 1

Configure environment

Step 2

Configure service access

Step 3 - optional

Set up networking, database, and tags

Step 4 - optional

Configure instance traffic and scaling

Step 5 - optional

Configure updates, monitoring, and logging

Step 6

Review

Configure environment

Environment tier

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

☒ Web server environment
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

☐ Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information

Application name

kissafrend0572

Maximum length of 100 characters.

Application tags (optional)

Environment information

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Kissafrend0572-env

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 1

Configure environment

Step 2

Configure service access

Step 3 - optional

Set up networking, database, and tags

Step 4 - optional

Configure instance traffic and scaling

Step 5 - optional

Configure updates, monitoring, and logging

Step 6

Review

Configure service access

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

☐ Create and use new service role

☒ Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

kissakeypair0572

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

Cancel

Skip to review

Previous

Next

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

It caused problems so i moved to CLI 😞

```
Administrator: C:\WINDOWS\system32\cmd.exe - CALL C:\Users\rajan\ebcli-virtual-env\Scripts\eb.exe init -p static
Uploading kissaapp0572/app-250511_193039170356.zip to S3. This may take a while.
Upload Complete.
Environment details for: kissaproject0572
  Application name: kissaapp0572
  Region: ap-southeast-2
  Deployed Version: app-250511_193039170356
  Environment ID: e-a39rxs2qcp
  Platform: arn:aws:elasticbeanstalk:ap-southeast-2::platform/Python 3.12 running on 64bit Amazon Linux 2023/4.5.1
  Tier: WebServer-Standard-1.0
  CNAME: UNKNOWN
  Updated: 2025-05-11 14:30:44.502000+00:00
Printing Status:
2025-05-11 14:30:43 INFO createEnvironment is starting.
2025-05-11 14:30:44 INFO Using elasticbeanstalk-ap-southeast-2-381492186725 as Amazon S3 storage bucket for environment data.
2025-05-11 14:31:05 INFO Created security group named: awseb-e-a39rxs2qcp-stack-AWSEBSecurityGroup-CBu4LE58ZHMJ
2025-05-11 14:31:21 INFO Created EIP: 54.206.202.217
2025-05-11 14:31:37 INFO Waiting for EC2 instances to launch. This may take a few minutes.
2025-05-11 14:32:00 INFO Instance deployment used the commands in your 'Procfile' to initiate startup of your application.
2025-05-11 14:32:04 INFO Instance deployment completed successfully.
2025-05-11 14:32:38 INFO Application available at kissaproject0572.eba-hqs3ana7.ap-southeast-2.elasticbeanstalk.com.
2025-05-11 14:32:38 INFO Successfully launched environment: kissaproject0572

(.ebcli-virtual-env) >eb deploy
Creating application version archive "app-250511_193457742375".
Uploading kissaapp0572/app-250511_193457742375.zip to S3. This may take a while.
Upload Complete.
2025-05-11 14:35:00 INFO Environment update is starting.
2025-05-11 14:35:05 INFO Deploying new version to instance(s).
2025-05-11 14:35:08 INFO Instance deployment used the commands in your 'Procfile' to initiate startup of your application.
2025-05-11 14:35:13 INFO Instance deployment completed successfully.
2025-05-11 14:35:18 INFO New application version was deployed to running EC2 instances.
2025-05-11 14:35:18 INFO Environment update completed successfully.

(.ebcli-virtual-env) >
```

Elastic Beanstalk

Applications

Environments

Change history

▼ Recent environments

Kissafrentend0572-env

Environments (4) Info

Q kissaproject0572

X

2 matches

Environment name

Health

Applica...

Platform

Domain

Runnin...

Tier na...

D

kissaproject0572

Ok

kissaapp0...

Python 3...

kissaproject0572.eba-hqs3ana...

app-2505...

WebServer

M

kissaproject0572 (terminated)

Unknown

kissaapp0...

Python 3...

kissaproject0572.eba-hqs3ana...

app-2505...

WebServer

M

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

LESSS GOO DONE!!

26