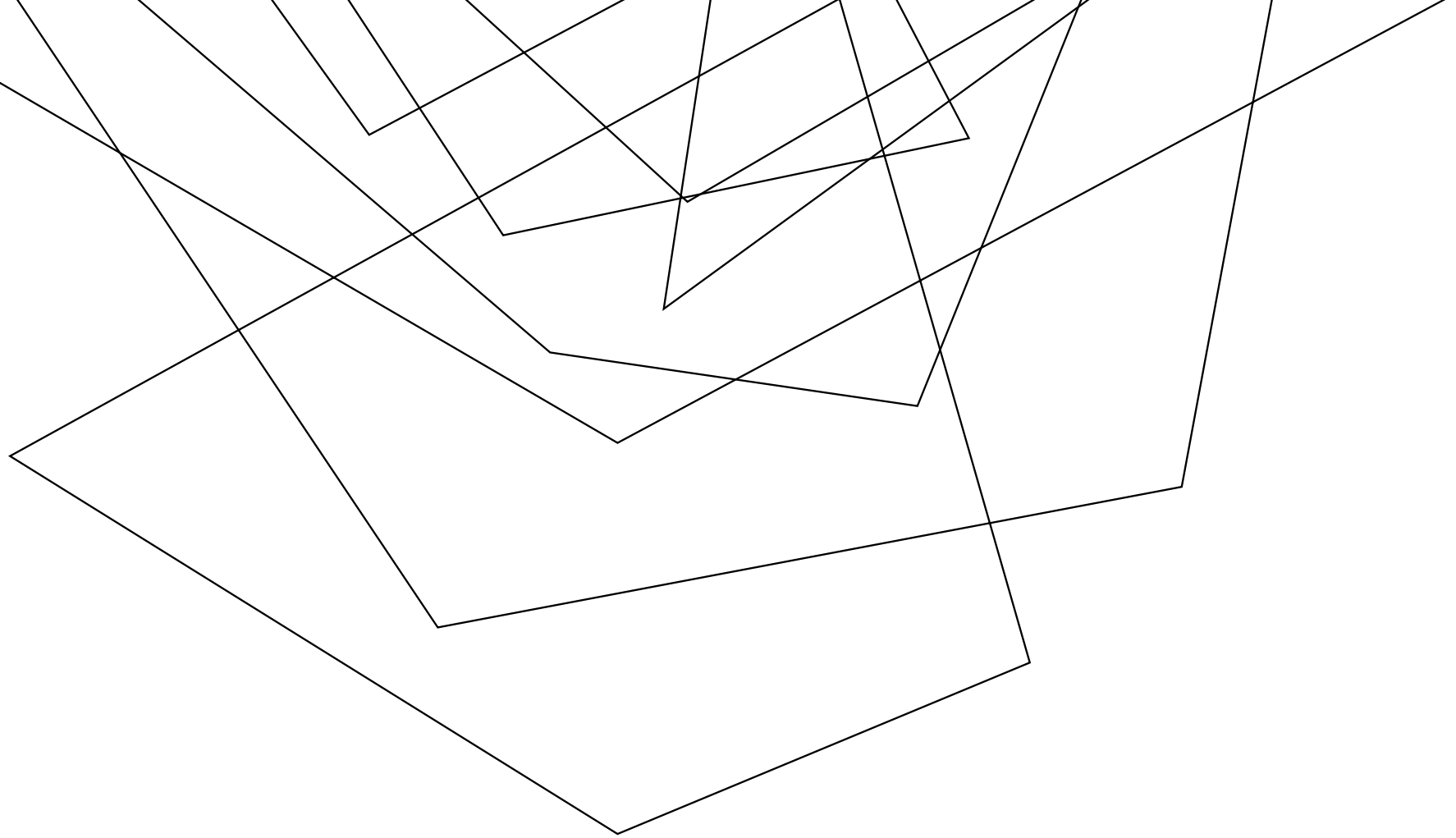


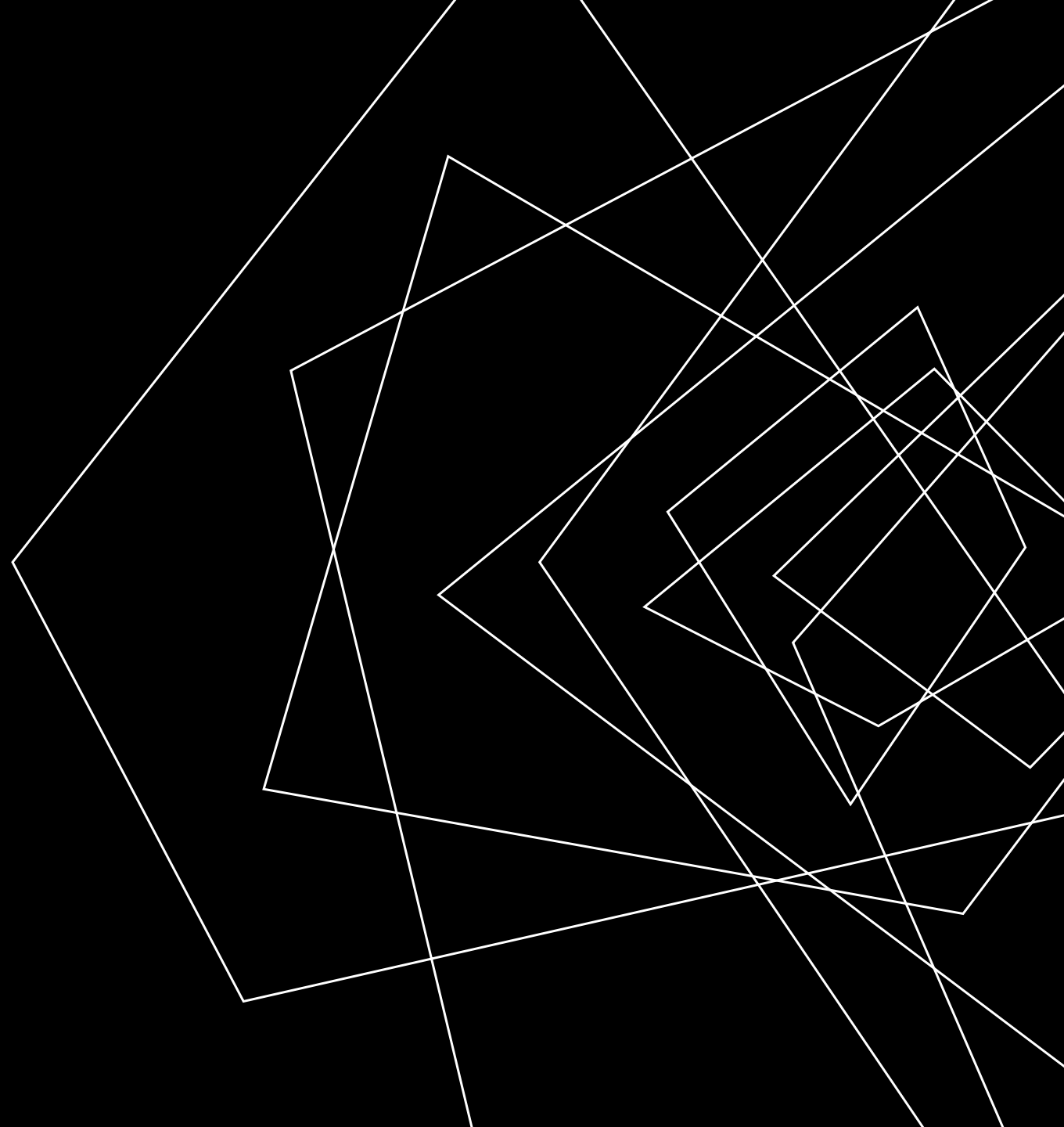
TO DO APPLICATION

Szymon Rećko, Hubert Ruczyński, Mateusz Sperkowski

01.06.2023

ABOUT THE APP

TO DO App is an online website hosted on the AWS servers that enables the user to create and log in to their own accounts, and create their workspace with easy to manage TO DO list. Additionally, the user can easily filter the tasks by their completion state or the due date.



FUNCTIONAL REQUIREMENTS

RETRIEVAL OF ACCOMPLISHED TASKS

Allow users to retrieve all the tasks in their to-do list.

CUSTOMIZABLE TASKS CREATION

Allow users to create a new task in their to-do list with a title, description, status, and due date.

POSSIBLE UPDATES

Allow users to update an existing task's title, description, status, and due date.

MONITORING TASKS STATES

Allow users to mark a task as TO DO, in progress, or completed, and remove it from the to-do list.

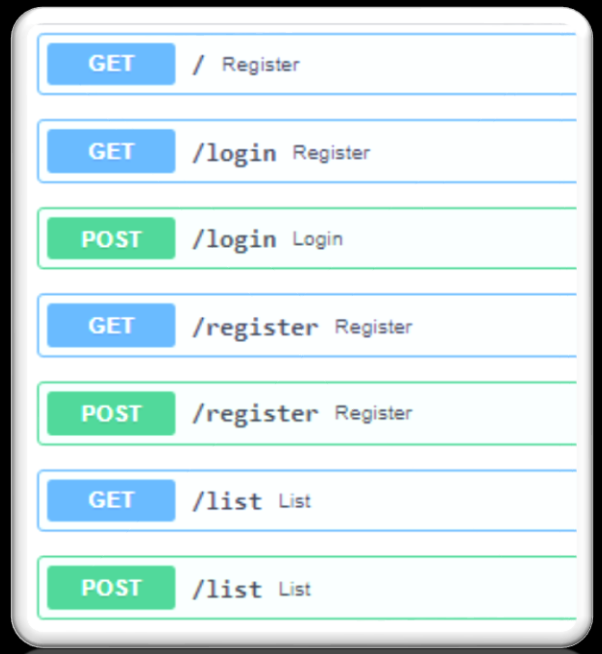
TASKS FILTERING

Provide an option for users to filter tasks by their status (in progress, completed, or incomplete) and due date.

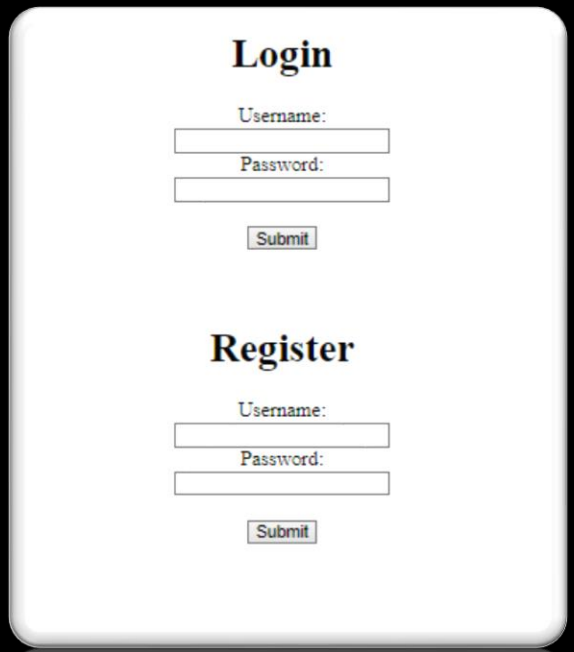
Registration and login

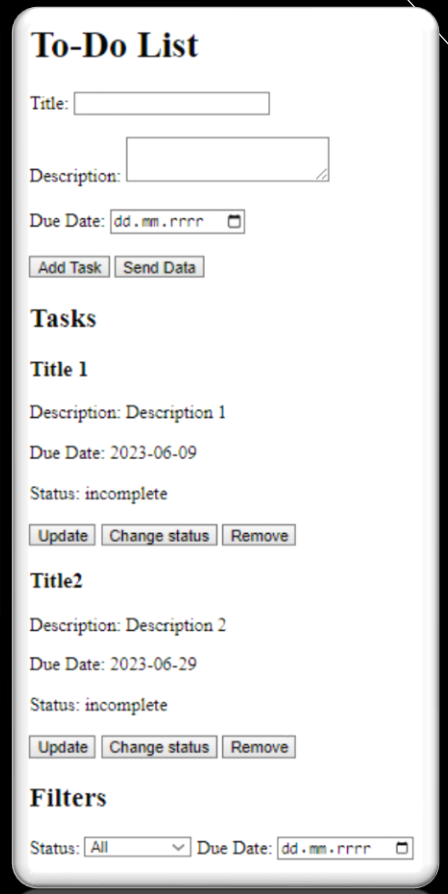
Allow users to create their personal accounts and safely log in to the system.

SERVERLESS LAMBDA APP



- Back-end in Python FastAPI
- "Front-end" in JavaScript and HTML
- Database connection using psycopg2
- Password hashing using bcrypt.
- Cookie-based authentication.
- Magnum - adapter for running FastAPI applications in Lambda
- Lambda stores database access parameters in environment variables





NON-FUNCTIONAL REQUIREMENTS

SCALABILITY

Handle an increase in traffic without any performance degradation.

RELIABILITY

Provide a consistent level of service even in the face of hardware or software failures.

SECURITY

Protect against unauthorized access and data breaches.

PERFORMANCE

Low response times available.

COST OPTIMIZATION

Minimize infrastructure costs while still meeting the application's needs.

NON-FUNCTIONAL REQUIREMENTS ASSURANCE

SCALABILITY

Scalable DB instances from 100GiB to 1000GiB.

The Lambda scales up to 400 concurrent workers.

RELIABILITY

Multi-AZ deployment with a Database standby copy in different AZs

The Lambda is available in 2 Azs which makes it resilient to failovers.

SECURITY

Database secured with login and password, available only from instances in the same VPC, and deployed in the private subnet.

Hashed account passwords.

PERFORMANCE

Extremely lightweight implementation with API calls.

Only aggregated read/write operations to and from the database.

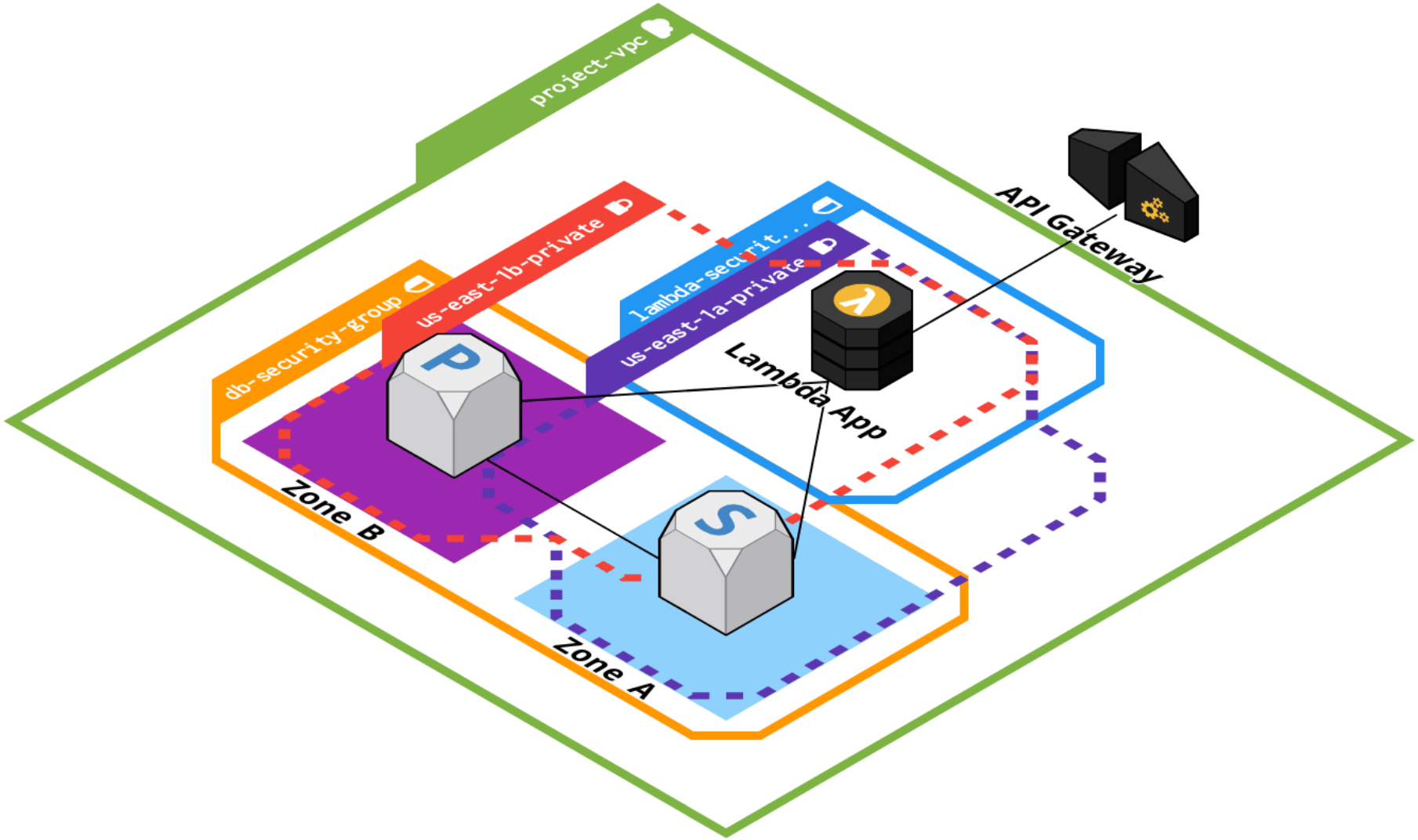
COST OPTIMIZATION

The use of cheap db.t4g.micro instances.

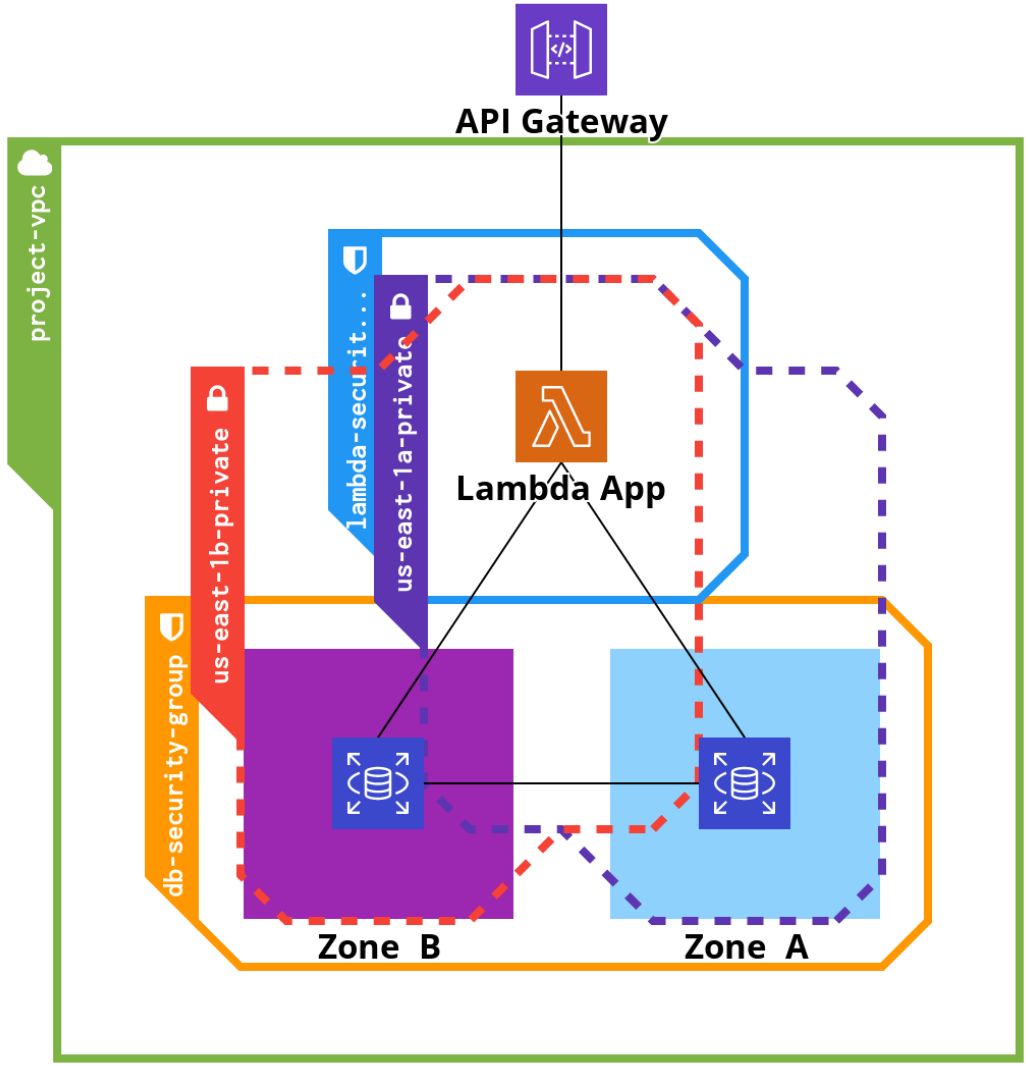
Lack of extensional expenses (ex. 2nd Region).

Lack of redundant features (ex. RDS proxy).

PROJECT DIAGRAM



PROJECT DIAGRAM



COST ESTIMATION

LAMBDA

App execution.

Estimated 5 million requests per month.

API GATEWAY

Connection with the Application.

Estimated 5 million requests per month.

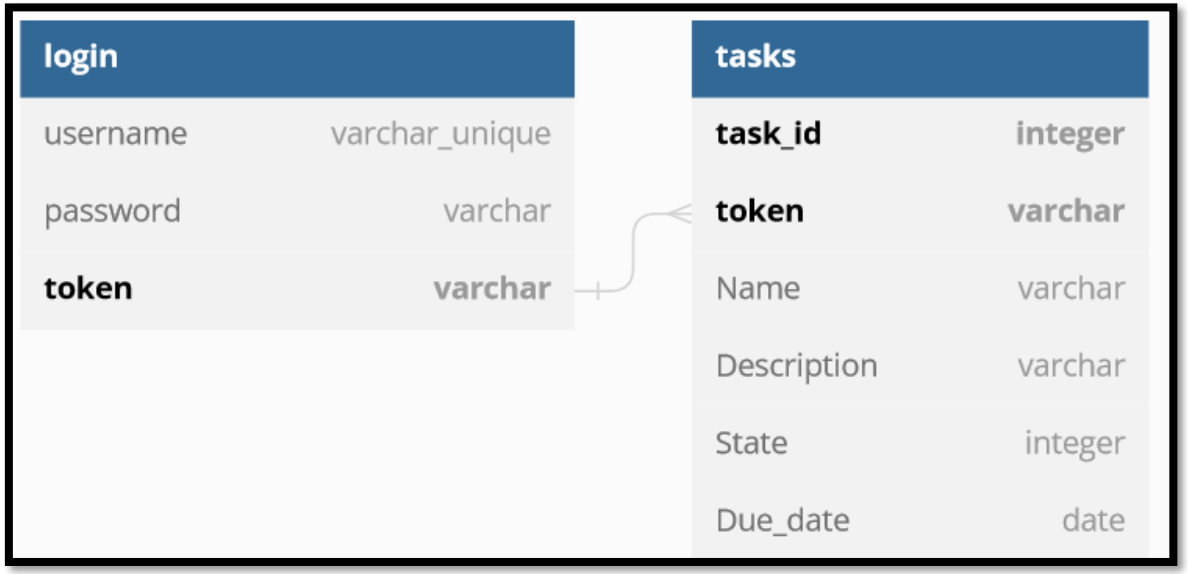
RDS

Primary PostgreSQL Database and its Standby in different Availability Zone.

DATABASE – POSTGRES SQL

SECURITY

- SQL injection protection,
- Available only from AWS VPC, with no public access,
- Database security group managing the access,
- Master admin account,
- Deletion protection,
- Hashed passwords using bcrypt.



BACKUP & DISASTER RECOVERY

DB MULTI-AZ DEPLOYMENT

With the Multi-AZ deployment, we can ensure that the users won't lose their data, as the standby DB will be always available in separate availability zone.

AUTOMATIC SWITCH

Thanks to the Multi-AZ technology, when the failover happens the application will automatically change to the standby instance with the current data.

RTO

According to the AWS documentation, the mean time between switching to the standby instance takes up to 1 minute.

TECHNICAL TIPS & TRICKS

PostgreSQL DB creation

During the early development select one AZ, and no backup to limit the costs, it’s easily applicable later on.

During the development make it public, so you can easily connect from pgAdmin.

Don’t use the RDS proxy, it enhances the costs greatly for every instance.

Lambda App

Develop the app using VPCs Endpoint Gateway, so the AWS services are easily accessible.

Firstly, develop the app, and later manage the cloud, you can see if you break something.

Create one Lambda with multiple endpoints for small applications, it’s less work while adding new features.

Cloudcraft diagrams

It’s a useful tool to use during the planning phase.

Greatly helps with the cost estimation.

Reminds about placing the services inside VPC, subnets, and so on.

SUMMARY

Cloud architecture is very beneficial for medium and large-size applications, as it easily enables upscaling and doesn’t bring any upfront costs, however, even the smallest machines like databases or EC2s are extremely costly, thus making small apps more pricey than they should be.

THANKS FOR ATTENTION