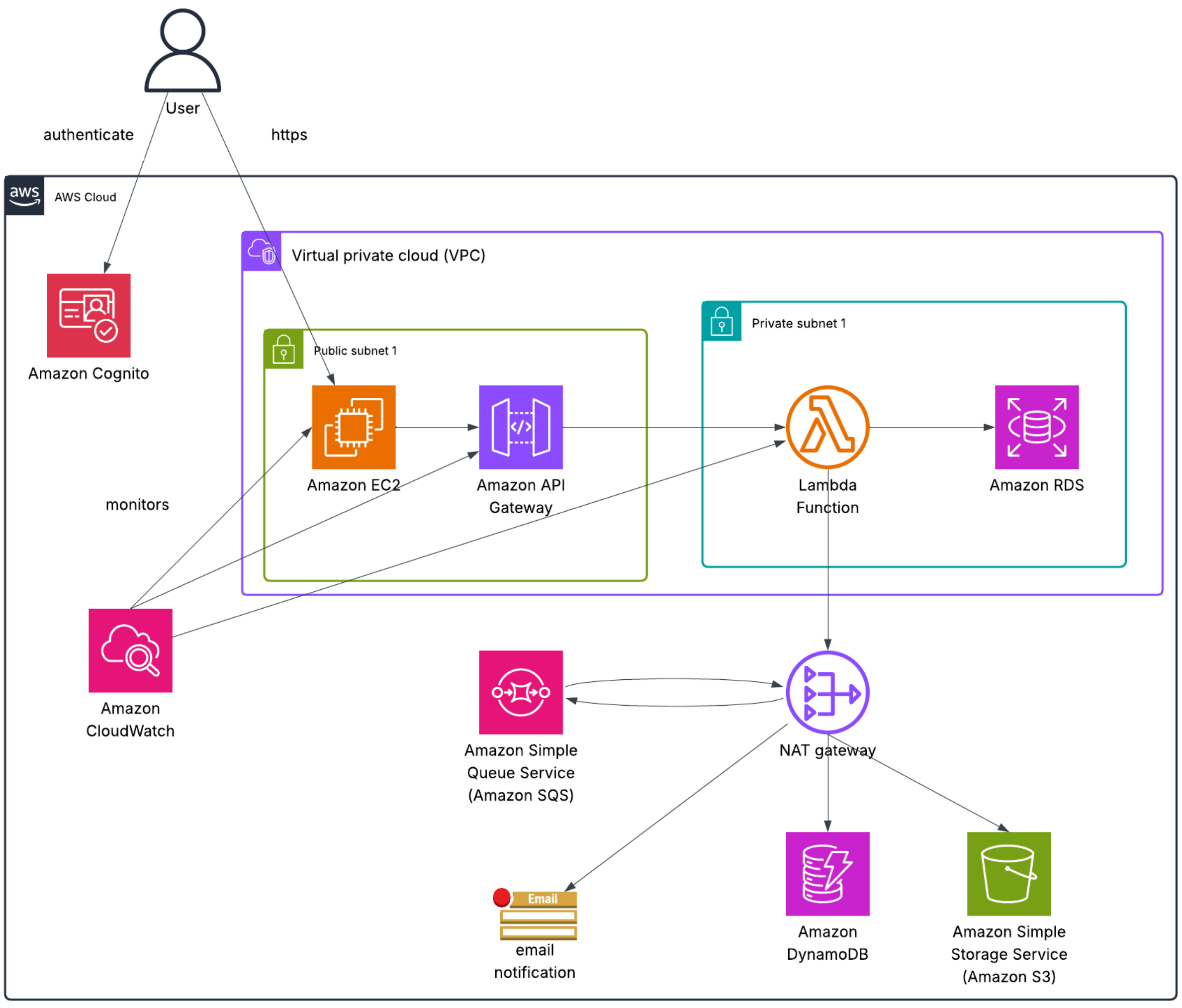
Task Management System on AWS - Documentation

1. Architecture Diagram

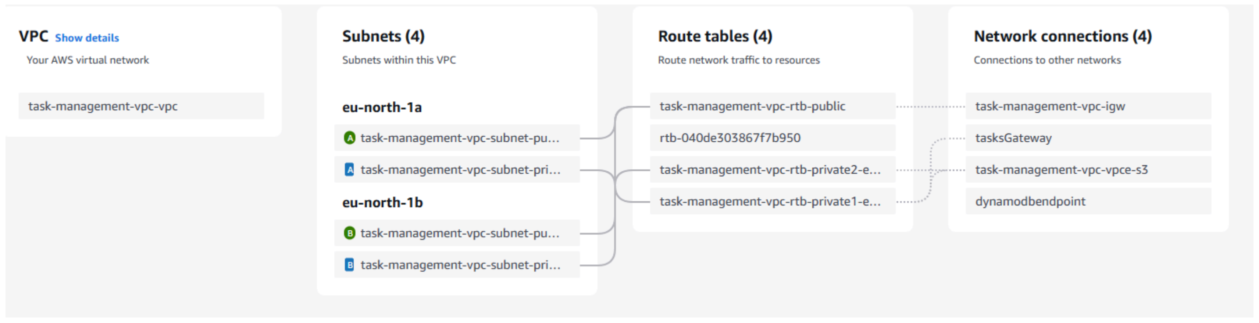
Below is a visual representations of the AWS services used in the Task Management System and their interactions:



2. Setup Guide

Step-by-Step Deployment

1. **Configure vpc :** Create public and private subnets


VPC Show details
Your AWS virtual network
task-management-vpc-vpc
Subnets (4)
Subnets within this VPC
eu-north-1a
A task-management-vpc-subnet-pu...
A task-management-vpc-subnet-pri...
eu-north-1b
B task-management-vpc-subnet-pu...
B task-management-vpc-subnet-pri...
Route tables (4)
Route network traffic to resources
task-management-vpc-rtb-public
rtb-040de303867f7b950
task-management-vpc-rtb-private2-e...
task-management-vpc-rtb-private1-e...
Network connections (4)
Connections to other networks
task-management-vpc-igw
tasksGateway
task-management-vpc-vpce-s3
dynamodbendpoint

1. Configure Amazon Cognito for User Authentication

1. Go to **AWS Cognito** → **Manage User Pools** → **Create a User Pool**.
2. Set up:
   - **Pool name**: `UserPool`
   - **Attributes**: Enable `Email` for sign-in.
   - **Password Policy**
     - i. **Password minimum length: 8 character(s)**
     - ii. **Temporary passwords set by administrators expire in 7 day(s)**
     - iii. **Allow reuse of previous passwords**
     - iv. **Password requirements**
       1. **Contains at least 1 number**
       2. **Contains at least 1 special character**
       3. **Contains at least 1 uppercase letter**
       4. **Contains at least 1 lowercase letter**
   - **App Client**: Create a new app client (`TaskManagementApp`).

2. Set Up RDS (Relational Database)

1. Go to **Amazon RDS** → **Create Database**.
2. Choose **MySQL/PostgreSQL**.
3. Configure:
   - **DB Instance Identifier**: `task-management-db`

- **Master Username & Password**: Set securely.
- **Public Access**: **No** (for security, use VPC).

Put in the private subnet

4. Create a database named `task-management-db`.

3. Set Up DynamoDB (NoSQL Database)

1. Go to **DynamoDB** → **Create Table**.
2. Configure:
   - **Table Name**: `TaskMetadata`
   - **Partition Key**: `taskId` (String)
   - Sortkey: userId

4. Configure S3 for File Attachments

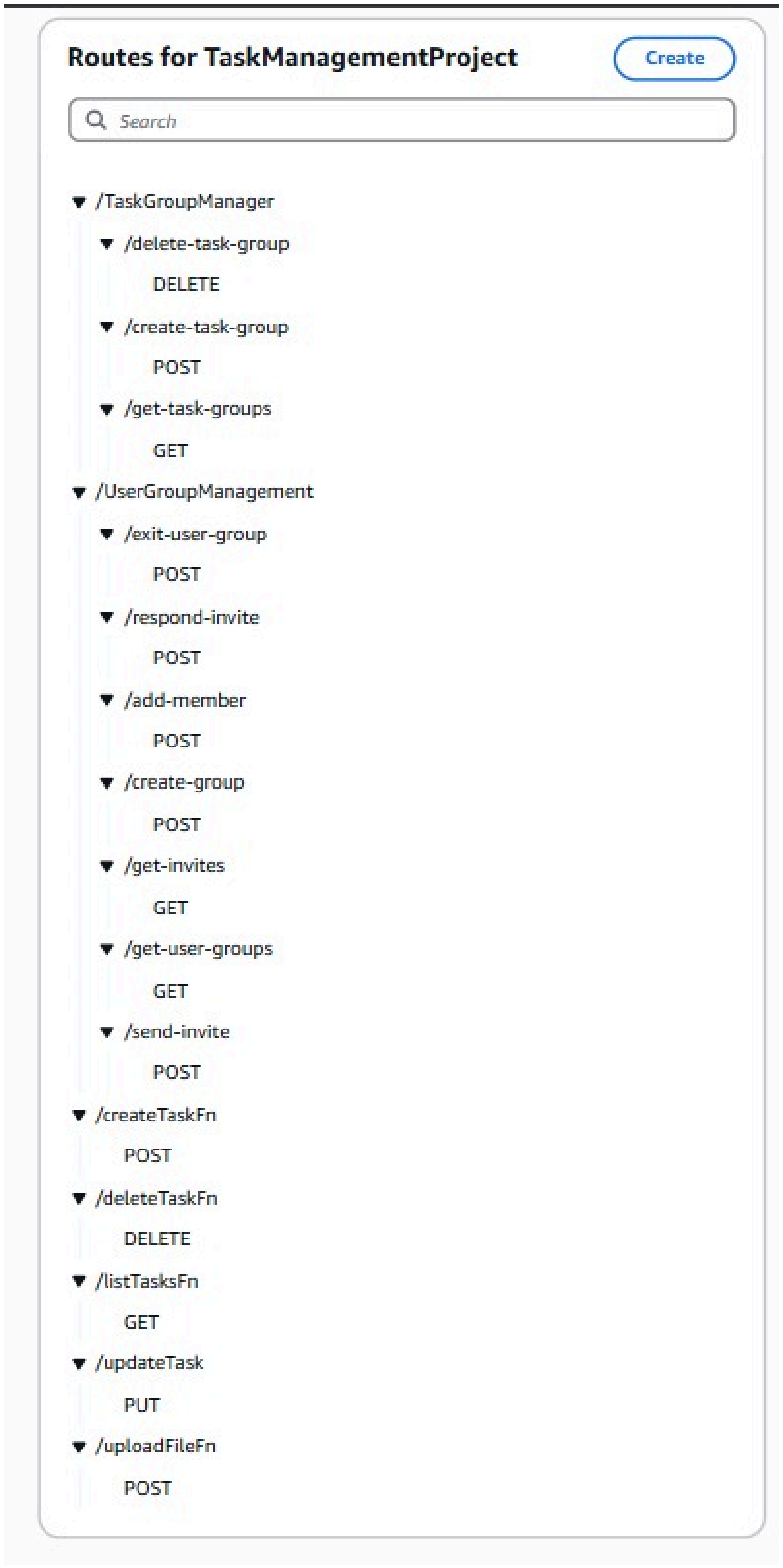
1. Go to **Amazon S3** → **Create Bucket**.
2. Set:
   - **Bucket Name**: `task-management-bucket5228`
   - **Block Public Access**: Enable (for security).
3. Create an **IAM Policy** allowing read/write access to this bucket.

5. Deploy AWS Lambda Functions

1. Go to **AWS Lambda** → **Create Function**.
2. Create functions for:
   - `createTaskFn` (handles task creation)
   - `Update task` (handles task updates)
   - `deleteTaskFn` (handles task deletion)
   - `TaskGroupManager(handles group task)`
   - `Send-task-email-notification (handles notifications)`
   - `listTasksFn (get tasks)`
   - `Send-task-email-notification (send the notification)`
   - `UserGroupManagement (manages user groups)`
   -
3. Attach IAM roles allowing access to **DynamoDB, RDS, S3, and SQS**.

6. Set Up API Gateway

1. Go to **API Gateway** → **Create API** (HTTP API).



Routes for TaskManagementProject

```
▼ /TaskGroupManager
    ▼ /delete-task-group
        DELETE
    ▼ /create-task-group
        POST
    ▼ /get-task-groups
        GET
▼ /UserGroupManagement
    ▼ /exit-user-group
        POST
    ▼ /respond-invite
        POST
    ▼ /add-member
        POST
    ▼ /create-group
        POST
    ▼ /get-invites
        GET
    ▼ /get-user-groups
        GET
    ▼ /send-invite
        POST
▼ /createTaskFn
    POST
▼ /deleteTaskFn
    DELETE
▼ /listTasksFn
    GET
▼ /updateTask
    PUT
▼ /uploadFileFn
    POST
```

2. Define endpoints:
3. Deploy the API to a stage (e.g., `prod`).

7. Configure SQS for Notifications

1. Go to **Amazon SQS** → **Create Queue**.
2. Set:
   - **Queue Name**: `task-updates-queue`
   - **Type**: Standard Queue
3. Modify Lambda function: updateTask to push notifications to this queue.
4. This queue triggers the lambda function: send-task-email-notification to send email notification

8. Set Up CloudWatch for Monitoring

1. Go to **CloudWatch** → **Logs** → **Create Log Group**.
2. Set up **Alarms** for:
   - High Lambda errors
   - API Gateway latency
   - EC2 CPU utilization

9. Deploy Web Application on EC2

1. Go to **EC2** → **Launch Instance**.
2. Choose an **Amazon Linux/Ubuntu AMI**.
3. Configure **Security Group** to allow HTTP/HTTPS.
4. Deploy the frontend code (Reactr) on this instance.

Deployment Steps

1. Upload Zip: Sent the TaskHub.zip file to EC2 using scp.
2. Connect to EC2: Used ssh with the .pem key to connect.
3. Install Apache: Installed and started Apache server.
4. Unzip and Move Build: Unzipped the React build and moved it to /var/www/myapp/frontend.
5. Set Permissions: Gave Apache user access to the frontend files.
6. Enable Proxy Modules: Enabled Apache proxy settings for backend API forwarding.
7. Create SSL Certificate: Generated a self-signed SSL cert and key.
8. Apache Config File: Created /etc/httpd/conf.d/myapp.conf to
   a. Set up HTTPS (port 443)
   b. Serve frontend from /frontend

    c. Proxy /api to the backend
9. Restart Apache: Restarted the server to apply everything.

10. Configure IAM Roles

1. Go to **IAM** → **Roles**.
2. Create roles for:
    - **EC2** (to access S3, DynamoDB, Cognito)
    - **Lambda** (to access RDS, SQS, DynamoDB)

11. Configure SES to send emails

1. Verify sender email
2. Verify recipient emails in sandbox

3. User Manual

How to Use the Task Management System

1. Sign Up / Login

- Visit the web application URL.
- Click **Sign Up** if new, or **Login** if existing.
- Enter **Email & Password** (managed by Cognito).

2. Create a User group and feel free to invite other users by email to collaborate with you on the group

- Type their email in the textbox
- Select the group you would like to invite them to and click invite
- They will receive the invitation and can choose to accept or decline

3.Create task groups within the user groups to organize tasks within each user group

- Click create new task group
- Choose a descriptive name for your task group

4. Create a Task

- Click **"Add Task"**.
- Fill in:
  - Title
  - Description
- Optionally **upload a file** (stored in S3).
- Click **Save**.

5. Update Task Status

- Open a task.
- Change **Status** (e.g., "In Progress" → "Completed").
- Click **Update**.

6. Delete a Task

- Open the task.
- Click **Delete**

7. View Tasks

- The dashboard lists all tasks per task group per user group.

8. Notifications

- When a task is updated, an **email notification** is sent (via SQS).

9. Leave User group

- Click Exit group and if you are the last member of the group it gets deleted

10. Delete task group

- Click delete task group and all tasks within will be deleted