AWS Deployment Documentation

AWS Deployment Documentation

This document provides detailed instructions on deploying the PDF Ingestion & Search API on AWS, as well as details on the current deployment setup using Render for the API and Northflank for Meilisearch. The API, built with Node.js, Express, MongoDB, and Meilisearch, supports ingesting parsed PDF data (paragraphs, images, and tables) and searching it securely via a REST API with JWT authentication.

Overview

The deployment process ensures the API is scalable, secure, and optimized for handling un-structured PDF data (paragraphs, images, tables) as required for the search functionality. The AWS deployment uses Elastic Beanstalk for hosting the Node.js application, Amazon RDS for MongoDB, and AWS Secrets Manager for secure configuration management. For the current deployment, Render hosts the Node.js API, and Northflank hosts the Meilisearch instance.

Deploying on AWS

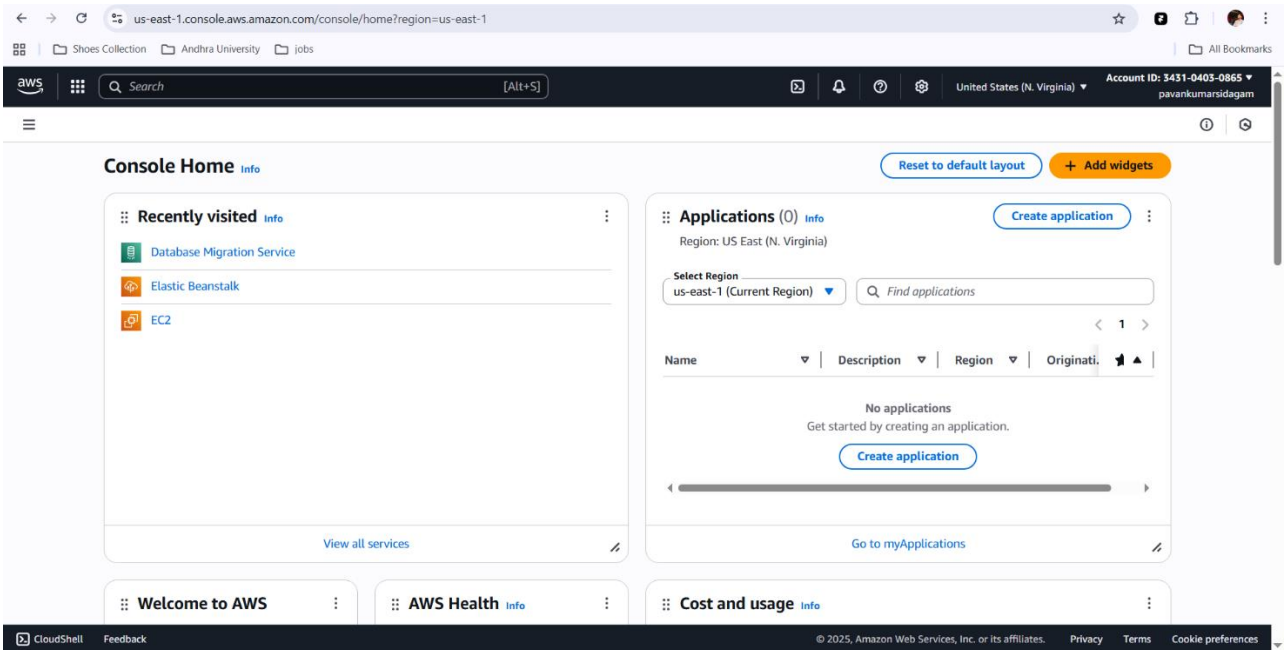
AWS Architecture

The AWS deployment uses the following services:

- **AWS Elastic Beanstalk**: Hosts the Node.js/Express API, providing automatic scaling and load balancing.
- **Amazon RDS (MongoDB-compatible)**: Stores parsed PDF data and metadata (alternatively, MongoDB Atlas can be used for simplicity).
- **AWS Secrets Manager**: Securely stores environment variables like MONGO_URI, MEILI_MASTER_KEY, and API_SECRET_KEY.
- **Amazon API Gateway (optional)**: Exposes the API endpoints with additional security and throttling.
- **Meilisearch**: Hosted externally (e.g., Northflank) or locally on an EC2 instance for search functionality.

Prerequisites

- AWS account with administrative access.
- AWS CLI installed and configured (aws configure).
- Node.js application code from the repository (https://github.com/pavankumarsidagam/ pdf-search-pipeline.git).
- Docker Desktop for local testing (optional).
- Basic familiarity with AWS services.



Deployment Steps

1. Prepare the Application

1. **Clone the Repository**:

```
git clone https://github.com/pavankumarsidagam/pdf-search-pipeline.git cd
pdf-search-pipeline
```

2. **Update Environment Variables**:
   - Create a .env file based on .env.example:

```
# MongoDB (RDS or Atlas)
MONGO_URI=mongodb:// <rds-endpoint >:27017/ pdf-search - pipeline #
OR for MongoDB Atlas
# MONGO_URI=mongodb+srv://<your-db-url> #
Meilisearch (Northflank or local EC2)
MEILI_URL=https:// meili--meilisearch --6y7qslcbcmz5 .code.run
MEILI_MASTER_KEY=F121523A
API_SECRET_KEY=F121523A PORT=5000
```

   - For AWS deployment, these variables will be managed via AWS Secrets Manager (see step 4).
3. **Package the Application**:
   - Ensure package.json includes all dependencies (e.g., express, mongoose, meilisearch, jsonwebtoken).
   - Run npm install to generate node_modules.

2. Set Up Amazon RDS for MongoDB

1. **Create an RDS Instance**:
   - Navigate to the AWS RDS console.
   - Create a new database with a MongoDB-compatible engine (e.g., DocumentDB) or use MongoDB Atlas for simplicity.
   - Note the endpoint (e.g., mongodb://<rds-endpoint>:27017).
2. **Configure Security**:

- Set up a security group to allow inbound traffic on port 27017 from the Elastic Beanstalk environment.
- Store the database credentials in AWS Secrets Manager (see step 4).

3. Deploy the Node.js Application with Elastic Beanstalk

1. Initialize Elastic Beanstalk:

```
npm install -g awsebcli
```

- Install the Elastic Beanstalk CLI:
- Run eb init in the project directory:

```
eb init -p "Node.js 20 running on 64bit Amazon Linux 2023" pdf-search-
    pipeline --region us-east-1
```

2. Create Environment:

- Create an Elastic Beanstalk environment:

```
eb create pdf-search-pipeline-env --single
```

- This deploys a single-instance environment for simplicity (use --elb-type application for load-balanced setups in production).

3. Deploy the Application:

- Deploy the code:

```
eb deploy
```

- Access the application at the provided Elastic Beanstalk URL (e.g., http://pdf-search-pipeline- us-east-1.elasticbeanstalk.com).

4. Configure Environment Variables with AWS Secrets Manager

1. Create a Secret:

- In the AWS Secrets Manager console, create a new secret with the following key-value pairs:

```
{
  "MONGO_URI":   "mongodb://<rds-endpoint>:27017/pdf-search-pipeline ",
  "MEILI_URL": "https://meili--meilisearch--6y7qslcbcmz5.code.run ",
  "MEILI_MASTER_KEY": "F121523A", "
  API_SECRET_KEY": "F121523A",
  "PORT": "5000"
}
```

2. Link Secret to Elastic Beanstalk:

- In the Elastic Beanstalk console, go to the environment's "Configuration" > "Soft- ware" section.
- Add environment variables referencing the secret (e.g., MONGO_URI, MEILI_URL) or configure the application to fetch secrets using the AWS SDK.

5. Configure Meilisearch

- **Option 1: Use Northflank Meilisearch (Current Setup):**
  - The project uses a Meilisearch instance hosted on Northflank (https://meili--meilisearch--6y7q code.run).

- No additional setup is required; ensure the MEILI_URL and MEILI_MASTER_KEY are set in Secrets Manager.

- **Option 2: Host Meilisearch on AWS EC2**:
  1. Launch an EC2 instance (e.g., t2.micro with Amazon Linux 2).
  2. Install Docker:

```
sudo yum update -y
sudo amazon-linux-extras install docker
sudo service docker start
sudo usermod -a -G docker ec2-user
```

  3. Run Meilisearch:

```
docker run -d -p 7700:7700 -e MEILI_MASTER_KEY=F121523A getmeili/
    meilisearch:v1.16
```

  4. Update MEILI_URL to the EC2 public IP (e.g., http://<ec2-public-ip>:7700) in Secrets Manager.
  5. Configure security group to allow inbound traffic on port 7700.

Current Deployment: Render and Northflank

Render Deployment (API)

The Node.js/Express API is currently deployed on Render, a platform that simplifies hosting web applications.

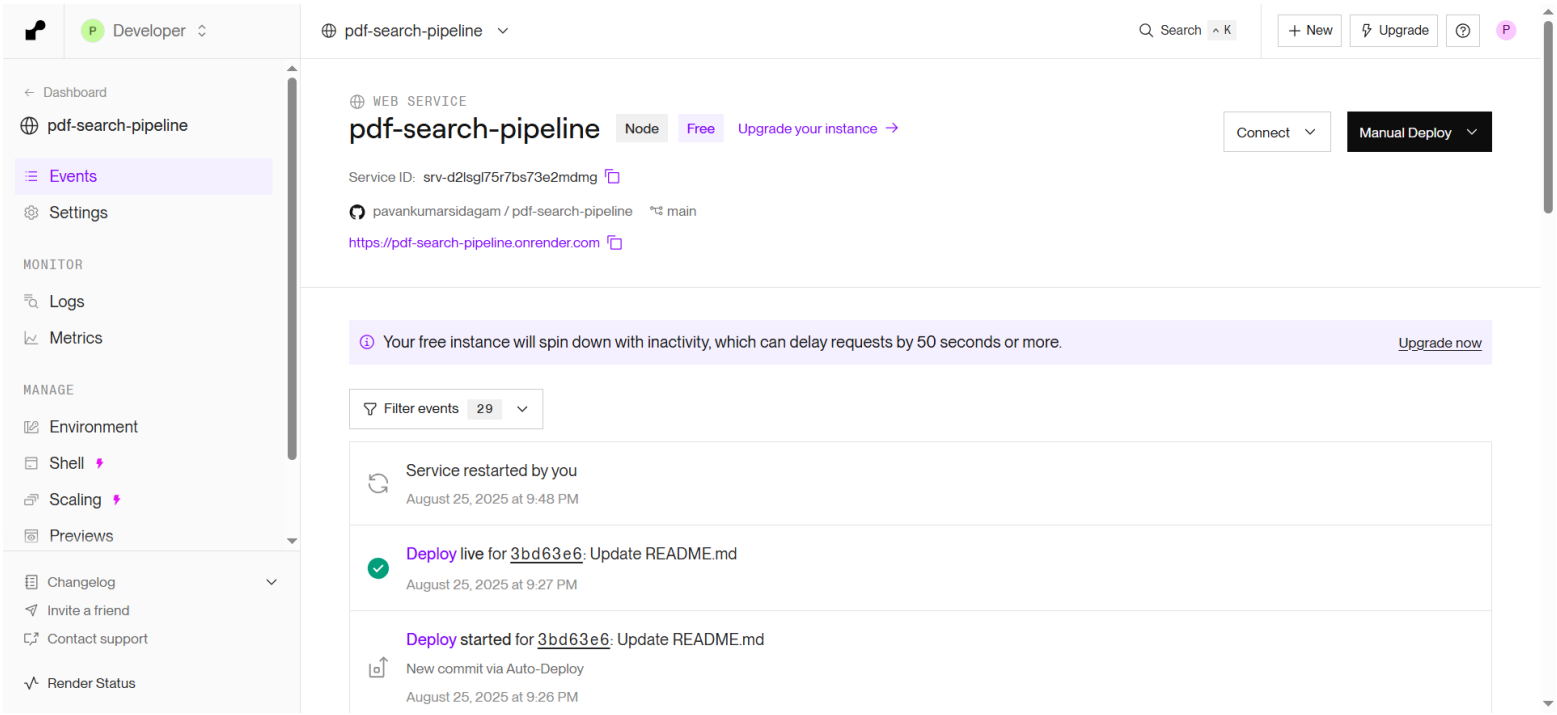
How It Was Deployed

1. **Create a Render Account**:
   - Sign up at https://render.com and create a new web service.
2. **Link Repository**:
   - Connect the GitHub repository (https://github.com/pavankumarsidagam/pdf-search-pipeline git) to Render.
   - Select Node.js as the runtime environment.
3. **Configure Environment Variables**:
   - In the Render dashboard, add the following environment variables:

```
MONGO_URI=mongodb+srv:// <your-db-url>
MEILI_URL=https:// meili--meilisearch --6y7qslcbcmz5.code.run
MEILI_MASTER_KEY=F121523A
API_SECRET_KEY=F121523A
PORT=5000
```

   - Used MongoDB Atlas for the database to avoid managing a database instance.
4. **Deploy**:
   - Trigger a deployment from the Render dashboard or set up automatic deploys on Git push.
   - Access the API at the provided Render URL (e.g., https://pdf-search-pipeline. onrender.com).

Screenshot:



Northflank Deployment (Meilisearch)

The Meilisearch instance is hosted on Northflank, a platform for deploying and managing containerized services.

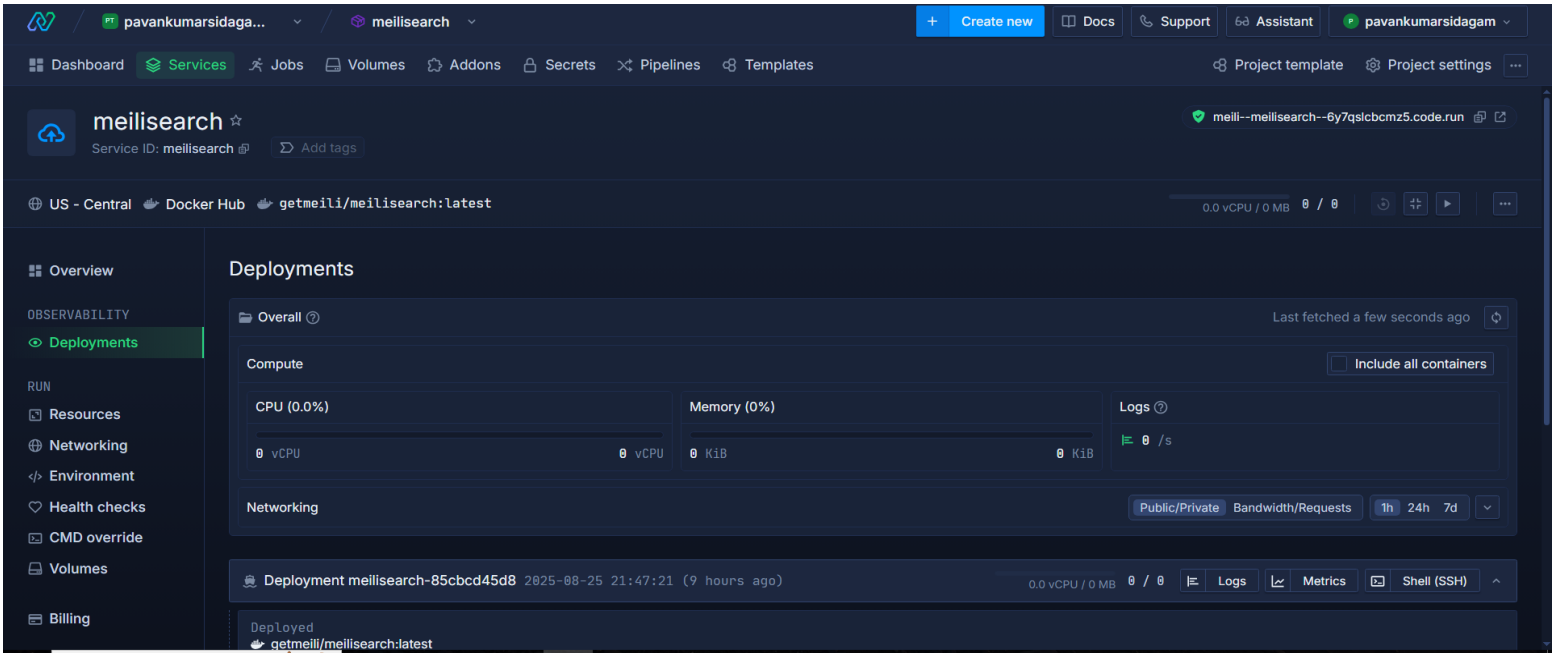
How It Was Deployed

1. **Create a Northflank Account**:
   - Sign up at https://northflank.com.
2. **Create a Meilisearch Service**:
   - In the Northflank dashboard, create a new service using the Meilisearch Docker image (getmeili/meilisearch@lastest).
   - Configure the master key:

     ```
     MEILI_MASTER_KEY=F121523A
     ```

   - Expose port 7700 for external access.
3. **Deploy and Access**:
   - Deploy the service and note the public URL (e.g., https://meili--meilisearch--6y7qslcbcmz5. code.run).
   - Update the API's .env file with this URL and the master key.
4. **Verify Meilisearch**:
   - Check https://meili--meilisearch--6y7qslcbcmz5.code.run/health to ensure the instance is running ({"status": "available"}).

Screenshot:


pavankumarsidaga...
meilisearch
Create new
Docs
Support
Assistant
pavankumarsidagam
Dashboard
Services
Jobs
Volumes
Addons
Secrets
Pipelines
Templates
Project template
Project settings
meilisearch
Service ID: meilisearch
Add tags
meili--meilisearch--6y7qslcbcmz5.code.run
US - Central
Docker Hub
getmeili/meilisearch:latest
0.0 vCPU / 0 MB
0 / 0
Overview
OBSERVABILITY
Deployments
RUN
Resources
Networking
Environment
Health checks
CMD override
Volumes
Billing
Deployments
Overall
Last fetched a few seconds ago
Compute
Include all containers
CPU (0.0%)
0 vCPU
0 vCPU
Memory (0%)
0 KiB
0 KiB
Logs
0 /s
Networking
Public/Private
Bandwidth/Requests
1h
24h
7d
Deployment meilisearch-85cbcd45d8
2025-08-25 21:47:21 (9 hours ago)
0.0 vCPU / 0 MB
0 / 0
Logs
Metrics
Shell (SSH)
Deployed
getmeili/meilisearch:latest