

NodeJS ile AWS Üzerinden Uygulama Ayağa Kaldırma

- REST API protokolleri ile nodejs kullanarak uygulama geliştirilmesi
- Bu uygulamayı AWS ekosisteminde ayağa kaldırıp, güvenlik tedbirlerinin koyulması

Assignments

1. Temel bir öğrenci ve öğretim görevlisi görüntüleme platformu için API geliştirin, bu platformda öğrenciler, öğretim görevlileri ve dersler bulunacak.
2. Bu platformun testlerini, Postman uygulamasını kullanarak yapabilirsiniz.
3. Bu API servisinizi, bir docker container'ı içerisinde çalıştırmanız bekleniyor.
4. Bu kurduğunuz docker container'ı, AWS ekosistemi içerisinde tercih ettiğiniz metot ile ayağa kaldırıp tercih ettiğiniz kişiler tarafından erişilebilir kılın.
 - a. Port'lerinizi açma konusunda araştırma yapın(Security grouplardan yapıyorsunuz)
5. Yaptığınız tüm çalışmalarda, endüstri standartı best practice'leri kullanmanız bekleniyor. Özellikle AWS'nin best practice'lerini, kendi dökümantasyonlarında uygun aratmaları yaparak bulabilirsiniz.
6. NodeJS ve mongoose dışında herhangi bir teknoloji sınırlaması bulunmamakta. Tek şart AWS ekosisteminden çıkmamanız.
7. Opsiyonel olarak, API'nize çeşitli güvenlik protokolleri ekleyebilirsiniz(JWT gibi), veya HTTPS bağlantı protokolüne bağlayabilirsiniz(SSL sertifikası için herhangi bir ücret ödemeyin(Domain gibi.)).

- when I run the project in EC2 ubuntu

```

=> [node_app 5/8] RUN npm install 0.8s
=> [node_app 6/6] COPY . . 0.1s
=> [node_app] exporting to image 0.6s
=> => exporting layers 0.6s
=> => writing image sha256:41d1a7aed93776732956acf581a08580d5ff93f4f19a9126ec3977227255bb6a 0.0s
=> => naming to docker.io/library/mongodb-docker-app-node_app 0.0s
[+] Running 3/3
✓ Network mongodb-docker-app_my-network Created 0.1s
✓ Container mongodb-docker-app-node_app-1 Created 0.1s
✓ Container mongodb-docker-app-mongodb-1 Created 0.1s
Attaching to mongodb-docker-app-mongodb-1, mongodb-docker-app-node_app-1
mongodb-docker-app-node_app-1 | > deneme@1.0.0 start /src
mongodb-docker-app-node_app-1 | > node index.js
mongodb-docker-app-node_app-1 |
mongodb-docker-app-node_app-1 | {"t":{"$date":"2023-08-22T18:13:14.917+00:00"},"s":"I", "c":"CONTROL", "id":23285,

```

- EC2 machine

The screenshot shows the AWS Management Console interface for an EC2 instance named 'MyServer' (ID: i-084c9043e5a10ae93). The instance is in a 'Running' state. The 'Instance summary' section provides key details:

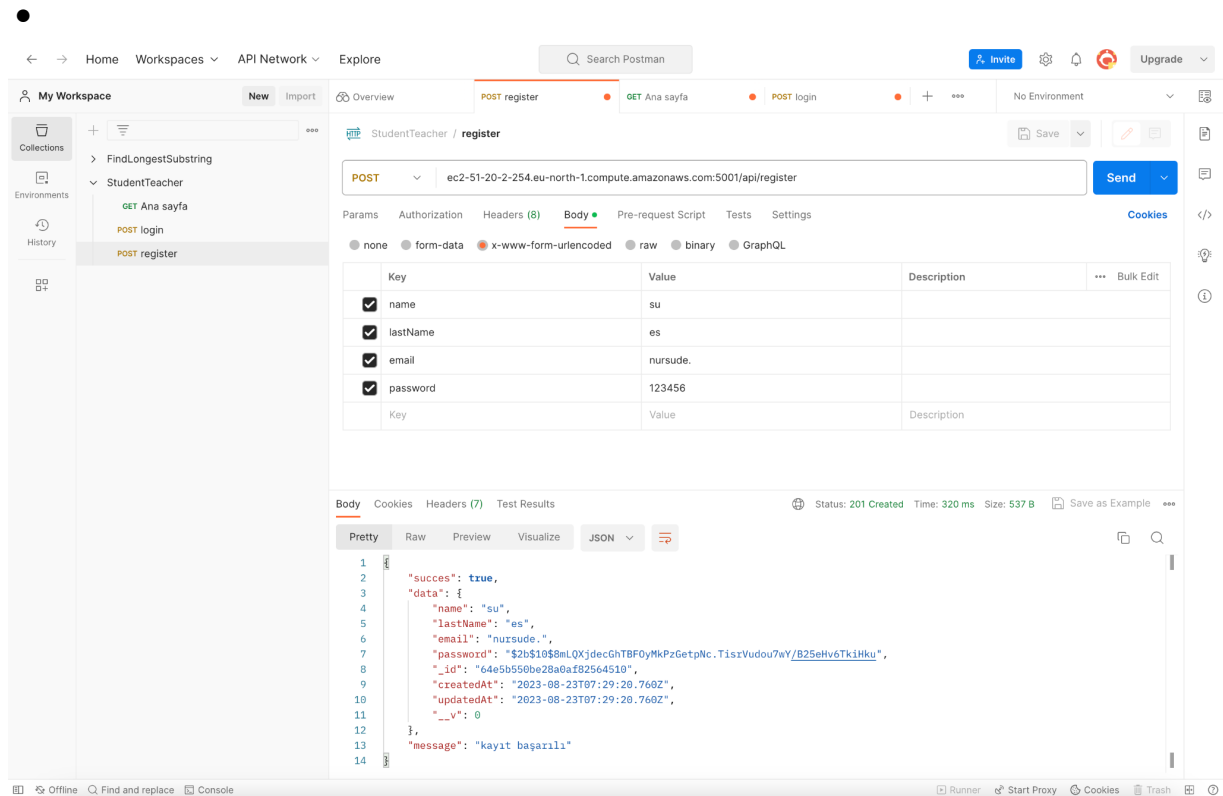
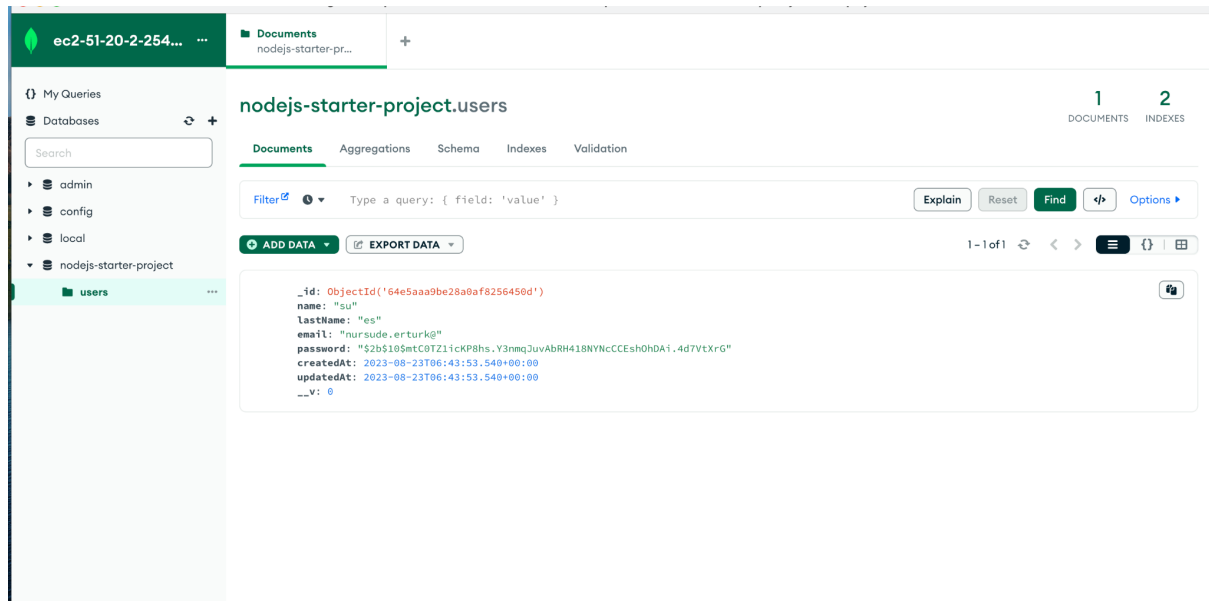
- Instance ID:** i-084c9043e5a10ae93 (MyServer)
- Public IPv4 address:** 51.20.2.254
- Private IPv4 addresses:** 172.31.39.231
- Instance state:** Running
- Public IPv4 DNS:** ec2-51-20-2-254.eu-north-1.compute.amazonaws.com
- Private IP DNS name (IPv4 only):** ip-172-31-39-231.eu-north-1.compute.internal
- Instance type:** t3.micro
- VPC ID:** vpc-0910721b526a1bd32
- Auto-assigned IP address:** 51.20.2.254 [Public IP]

The console also shows tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. A footer note mentions 'Opt-in to AWS Compute Optimizer for recommendation'.

- EC2 machine Inbound rules

Instance: i-084c9043e5a10ae93 (MyServer)					
Filter rules					
Name	Security group rule ID	Port range	Protocol	Source	Security group
-	sgr-0a692286d13e58b81	443	TCP	::/0	MyServerSec
-	sgr-0a3d803fa2b5c3685	80	TCP	0.0.0.0/0	MyServerSec
-	sgr-09f61478abd3b8bcb	27017	TCP	0.0.0.0/0	MyServerSec
-	sgr-05c41a0a7d26824db	80	TCP	::/0	MyServerSec
-	sgr-0e15530adf5788912	5001	TCP	0.0.0.0/0	MyServerSec
-	sgr-0c65a2aae26604cd	443	TCP	0.0.0.0/0	MyServerSec
-	sgr-09ce3ee141bfdb333	22	TCP	0.0.0.0/0	MyServerSec
Outbound rules					
Filter rules					

- Saving data to MongoDB Compass



←

→

Home

Workspaces

API Network

Explore

Search Postman

invite

Upgrade

My Workspace

NewImport

OverviewPOST registerGET Ana sayfaPOST login+...No Environment

Collections

+FindLongestSubstring

StudentTeacher

GET Ana sayfa

POST login

POST register

StudentTeacher / login

Save

Send

POST

ec2-51-20-2-254.eu-north-1.compute.amazonaws.com:5001/api/login

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Key	Value	Description	Bulk Edit
email	nursude.		
password	12345		
Key	Value	Description	

Body

Cookies

Headers (7)

Test Results

Status: 200 OKTime: 184 msSize: 274 BSave as Example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

"email": "nursude.",

"password": "12345"

Online

Find and replace

Console

Runner

Start Proxy

Cookies

Trash