

Deploy a Web App on Nginx Server using AWS App Runner

Overview

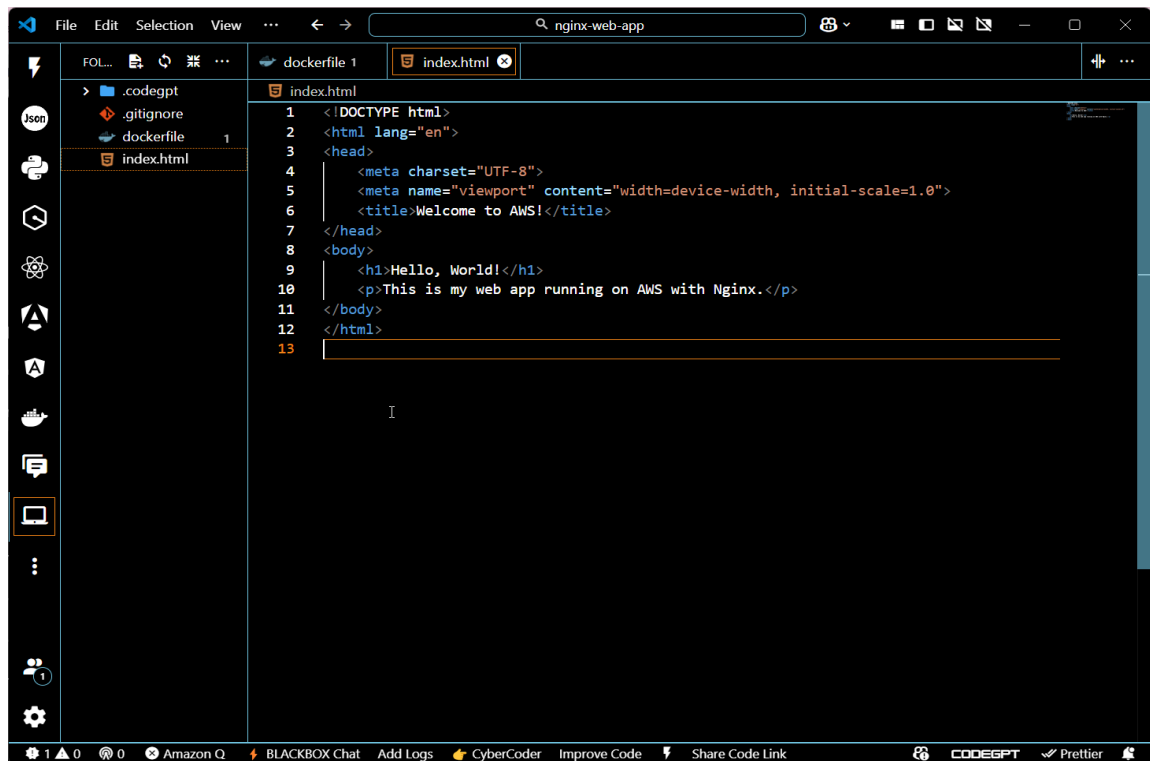
In this project, you will learn how to deploy a sample containerized application on a Nginx server using AWS App Runner.

AWS App Runner is a fully managed service that makes it easy for developers to quickly deploy containerized web applications and APIs, at scale and with no prior infrastructure experience required. Start with your source code or a container image. App Runner automatically builds and deploys the web application and load balances traffic with encryption. App Runner also scales up or down automatically to meet your traffic needs.

What you will accomplish

In this project, you will:

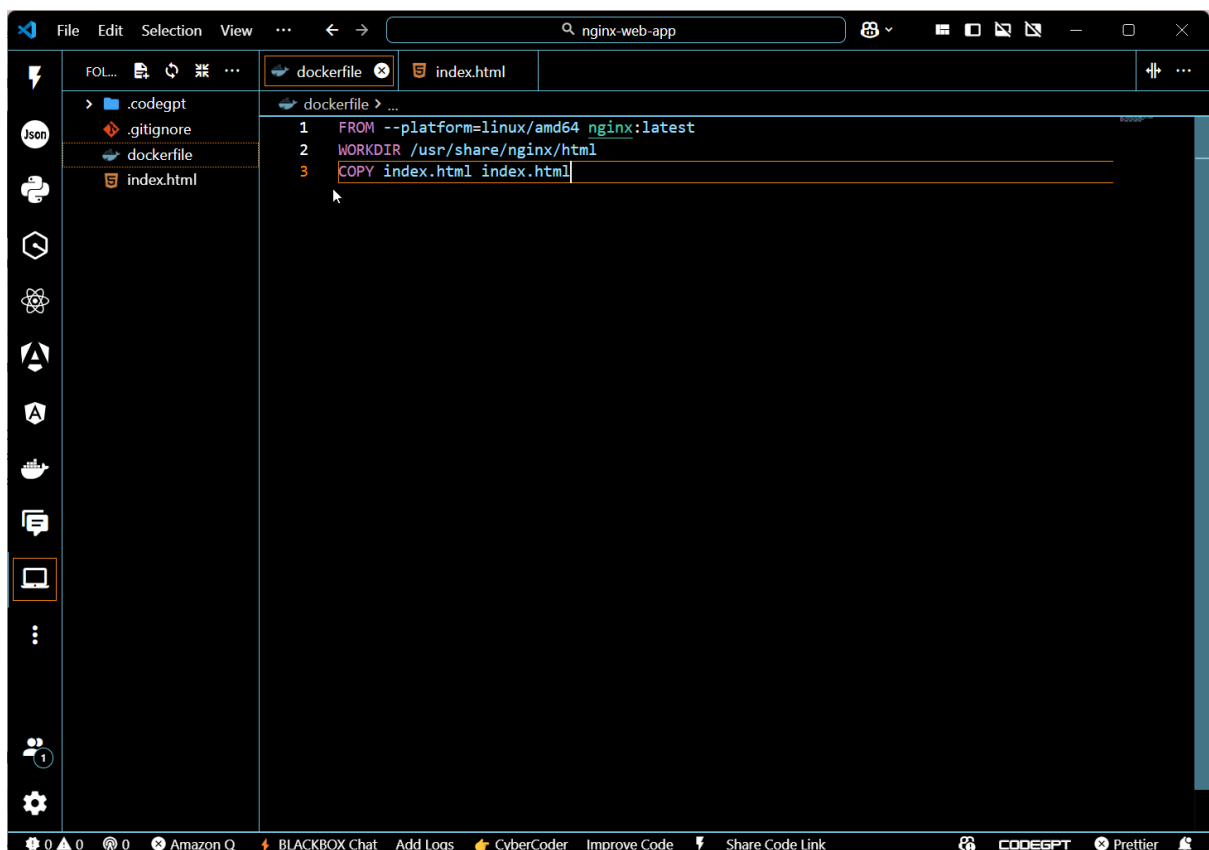
- Create a container image for your web app
- Push the image to Amazon Elastic Container Registry
- Create an AWS App Runner service
- Clean up your resources
- **STEP 1 CREATE A CONTAINER IMAGE FOR YOUR WEB APP**
 - **Set Up Your Web App Locally**
 - Let's assume you have a basic web app (like an `index.html` file).
LIKE THIS



The screenshot shows the Visual Studio Code editor with the 'index.html' file open. The file contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Welcome to AWS!</title>
7 </head>
8 <body>
9   <h1>Hello, World!</h1>
10  <p>This is my web app running on AWS with Nginx.</p>
11 </body>
12 </html>
13
```

Create another file named *Dockerfile*, and update it



The screenshot shows the Visual Studio Code editor with the 'Dockerfile' file open. The file contains the following Dockerfile content:

```
1 FROM --platform=linux/amd64 nginx:latest
2 WORKDIR /usr/share/nginx/html
3 COPY index.html index.html
```

- Then login into aws linux 2 instances and then follow bellow steps

- cmd

```
mkdir nginx-web-app
```

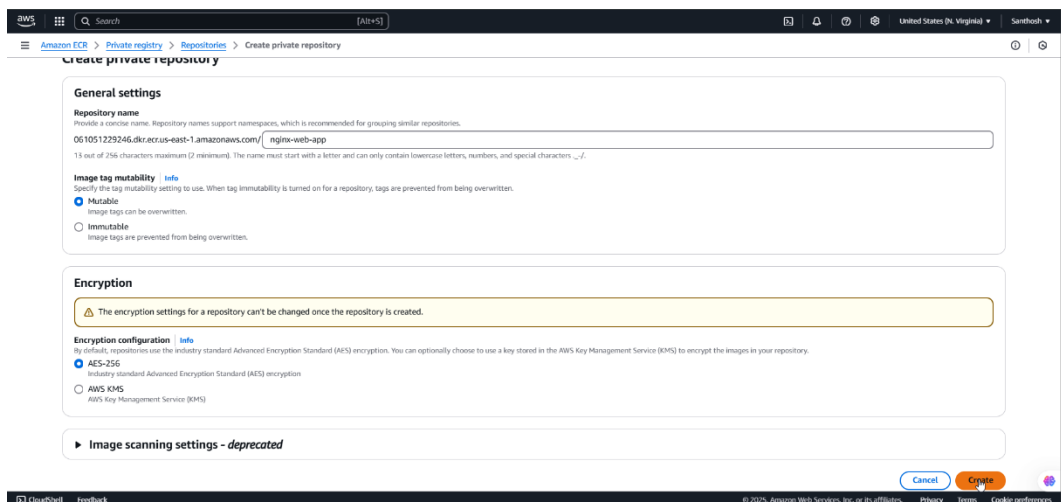
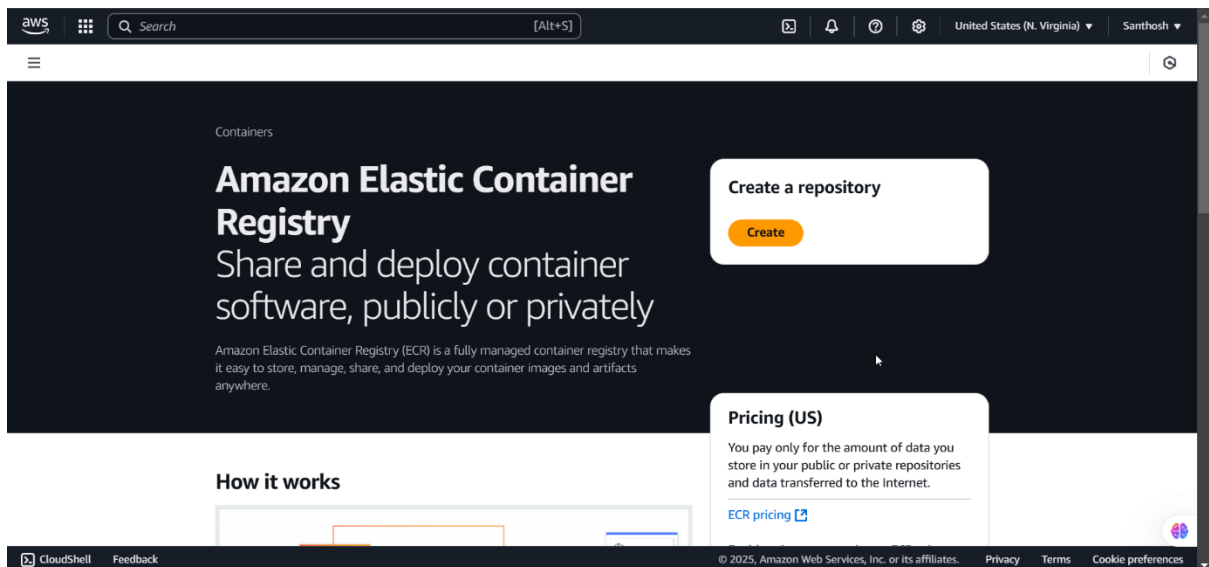
```
cd nginx-web-app
```

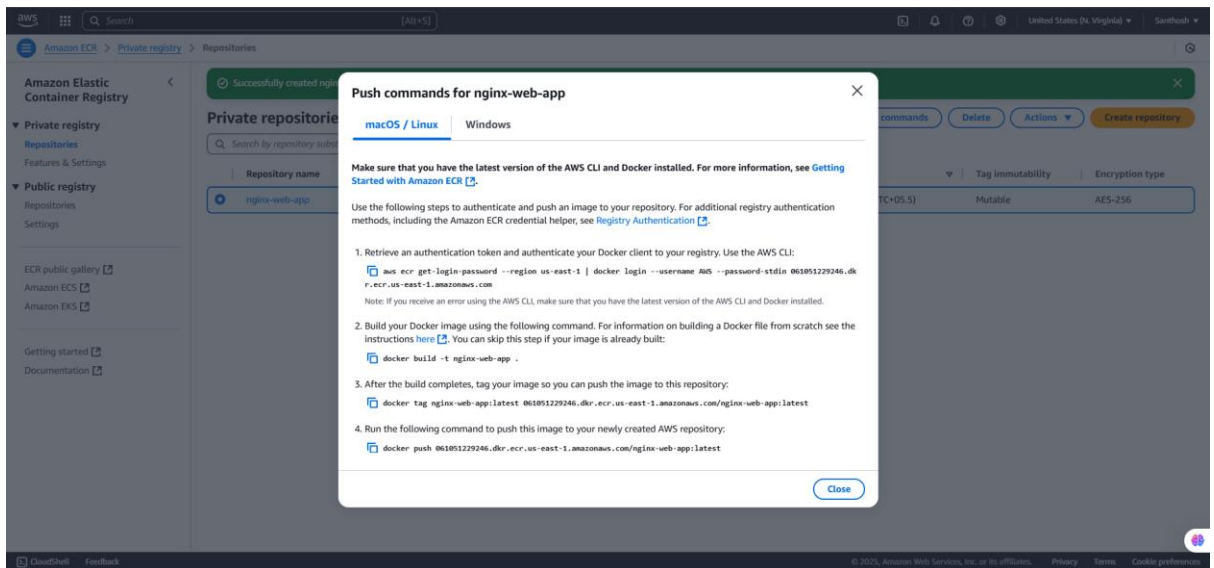
,then move the my local file into ec2 instance using filezilla (ftp tool) or github clone

Create the container images >>> **docker build -t nginx-web-app .**

STEP 2> Push the image to Amazon Elastic registry

Create

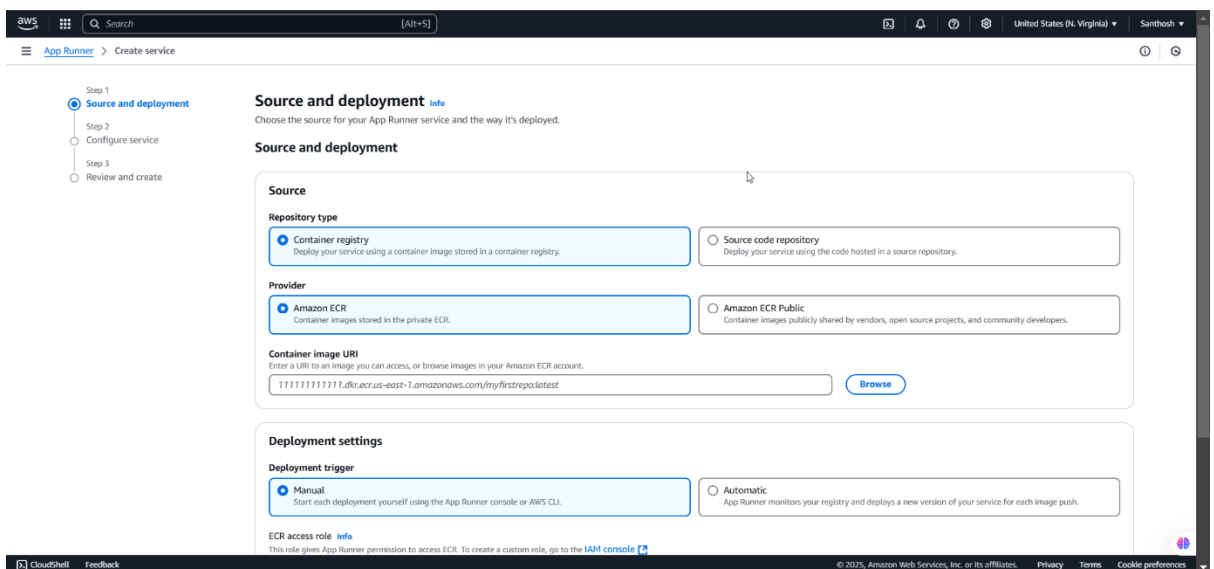




Then need install cli in ec2 server

NEED ADD IAM ROLE IN(Amazon\$SMRoleForInstancesQuickSetup) Search this and add this permission (AmazonEC2ContainerRegistryFullAccess.)

Step 3 create an AWS App Runner service



In the Deployment settings section, for ECR access role, select Create new service role, and choose Next.

On the Configure service page, for Service name enter nginx-web-app-service, and change the Port to 80. Leave the rest as default, and select Next.

Step 1 Source and deployment
Step 2 **Configure service**
Step 3 Review and create

Configure service

Service settings

Service name
nginx-web-app-service

Virtual CPU: 1 vCPU
Virtual memory: 2 GB

Runtime environment variables - optional
Add environment variables in plain text or reference them from [Secrets Manager](#) and [SSM Parameter Store](#). Update IAM Policies using the IAM Policy template given below to securely reference secrets and configurations as environment variables.
No environment variables have been configured.
[Add environment variable](#)
You can add up to 50 items.

IAM policy templates for secrets

Port
Your service uses this TCP port.
80

Additional configuration

On the Review and create page, review all inputs, and choose Create & deploy.

It will take several minutes for the service to be deployed. You can view the event logs for progress.

Successfully deployed to nginx-web-app-service

Running

Incoming traffic
Public endpoint
Default domain
<https://dtpga7xzt.us-east-1.awsapprunner.com>

Source
061051229246.dkr.ecr.us-east-1.amazonaws.com/nginx-web-app-latest

Logs Activity Metrics Observability Configuration Custom domains

App Runner event logs

View logs of events in the lifecycle of your App Runner service.

Search logs

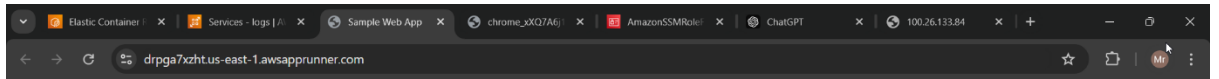
```
1 01-21-2025 05:23:10 PM [AppRunner] Deployment with ID : d17fa939ecd14432bcb1ac83998dfc26 started. Triggering event : SERVICE_CREATE
2 01-21-2025 05:23:39 PM [AppRunner] Deployment Artifact: [Repo type: ECR], [Image URL: 061051229246.dkr.ecr.us-east-1.amazonaws.com/nginx-web-app], [Image Tag: latest]
3 01-21-2025 05:23:41 PM [AppRunner] Pulling image 061051229246.dkr.ecr.us-east-1.amazonaws.com/nginx-web-app from ECR repository.
4 01-21-2025 05:23:51 PM [AppRunner] Successfully pulled your application image from ECR.
5 01-21-2025 05:23:51 PM [AppRunner] Provisioning instances and deploying image for publicly accessible service.
6 01-21-2025 05:24:01 PM [AppRunner] Performing health check on protocol 'TCP' [Port: '80'].
7 01-21-2025 05:24:39 PM [AppRunner] Health check is successful. Routing traffic to application.
8 01-21-2025 05:25:01 PM [AppRunner] Successfully routed incoming traffic to application.
9 01-21-2025 05:26:02 PM [AppRunner] Deployment with ID : d17fa939ecd14432bcb1ac83998dfc26 completed successfully.
```

Deployment logs (1)

View logs related to source repository deployments to your App Runner service.

Filter operations

Once the status updates to Running, choose the default domain name URL to view the web app.



Welcome to AWS App Runner!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

Thank you for using AWS App Runner!

