EC2 IP Address: 3.132.208.92

3.132.208.92

ECR Repository URI: 586794458740.dkr.ecr.us-east-2.amazonaws.com/django\_docker\_aws

Github Repository: <https://github.com/HasanMorshedBhuiyan/django_docker_aws.git>

Workshop2 Repository: 586794458740.dkr.ecr.us-east-2.amazonaws.com/workshop2

aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 586794458740.dkr.ecr.us-east-2.amazonaws.com

**EC2 Ipv4 DNS:** ec2-3-132-208-92.us-east-2.compute.amazonaws.com

**‘**ec2-3-132-208-92.us-east-2.compute.amazonaws.com **‘, '0.0.0.0', 'localhost', '127.0.0.1'**

**nctutorials Endpoint:** nctutorials.cn4ssqccgkcr.us-east-2.rds.amazonaws.com

Port: 5432

Git repository: https://github.com/HasanMorshedBhuiyan/workshop2.git

echo "# workshop2" >> README.md

git init

git add .

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/HasanMorshedBhuiyan/workshop2.git

git push -u origin main

**how to remove one repository**

To remove (delete) a repository on GitHub, follow these steps:

1. **Navigate to the Repository:**
   * Go to GitHub and sign in to your account.
   * Navigate to the repository you want to delete.
2. **Access Repository Settings:**
   * Click on the "Settings" tab located in the repository navigation bar.
3. **Scroll to the Danger Zone:**
   * Scroll down to the bottom of the settings page until you find the "Danger Zone."
4. **Delete the Repository:**
   * In the "Danger Zone," click on the "Delete this repository" button.
   * A confirmation dialog will appear. You will need to type the repository name to confirm the deletion.
5. **Confirm Deletion:**
   * Type the repository name as prompted and click on the "I understand the consequences, delete this repository" button.

Make sure you really want to delete the repository as this action is irreversible.

Here is a visual guide from GitHub's documentation: [Deleting a repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/deleting-a-repository).

ssh -i nucamp-private-key.pem ubuntu@ec2-3-132-208-92.us-east-2.compute.amazonaws.com

git clone <https://github.com/HasanMorshedBhuiyan/workshop2.git>

<http://ec2-3-132-208-92.us-east-2.compute.amazonaws.com:8000>

**http://**ec2-3-132-208-92.us-east-2.compute.amazonaws.com**:8000/api/tutorials/**

## Push commands for django\_docker\_aws

* macOS / Linux

**Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see**[**Getting Started with Amazon ECR**](https://docs.aws.amazon.com/AmazonECR/latest/userguide/getting-started-cli.html)**.**

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](https://docs.aws.amazon.com/AmazonECR/latest/userguide/Registries.html#registry_auth).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

**aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 586794458740.dkr.ecr.us-east-2.amazonaws.com**

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

1. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html). You can skip this step if your image is already built:

**docker build -t django\_docker\_aws .**

1. After the build completes, tag your image so you can push the image to this repository:

**docker tag django\_docker\_aws:latest 586794458740.dkr.ecr.us-east-2.amazonaws.com/django\_docker\_aws:latest**

1. Run the following command to push this image to your newly created AWS repository:

**docker push 586794458740.dkr.ecr.us-east-2.amazonaws.com/django\_docker\_aws:latest**

## Push commands for django\_docker\_aws

* Windows

**Make sure that you have the latest version of the AWS TOOLS for PowerShell and Docker installed. For more information, see**[**Getting Started with Amazon ECR**](https://docs.aws.amazon.com/AmazonECR/latest/userguide/getting-started-cli.html)**.**

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](https://docs.aws.amazon.com/AmazonECR/latest/userguide/Registries.html#registry_auth).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS TOOLS for PowerShell:

**(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin 586794458740.dkr.ecr.us-east-2.amazonaws.com**

Note: If you receive an error using the AWS TOOLS for PowerShell, make sure that you have the latest version of the AWS TOOLS for PowerShell and Docker installed.

1. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html). You can skip this step if your image is already built:

**docker build -t django\_docker\_aws .**

1. After the build completes, tag your image so you can push the image to this repository:

**docker tag django\_docker\_aws:latest 586794458740.dkr.ecr.us-east-2.amazonaws.com/django\_docker\_aws:latest**

1. Run the following command to push this image to your newly created AWS repository:

**docker push 586794458740.dkr.ecr.us-east-2.amazonaws.com/django\_docker\_aws:latest**