

# MLflow Remote Experiment Tracking with AWS EC2 Instance

This project demonstrates how to configure and use MLflow for remote experiment tracking on an AWS EC2 instance. It provides an end-to-end guide for setting up an MLflow tracking server on EC2, connecting your ML projects to it, and managing experiment results remotely in a cloud environment.

## MLFLOW On AWS

### MLflow on AWS Setup:

1. Login to AWS console.
2. Create IAM user with AdministratorAccess
3. Export the credentials in your AWS CLI by running "aws configure"
4. Create a s3 bucket
5. Create EC2 machine (Ubuntu) & add Security groups 5000 port

Run the following command on EC2 machine

```
sudo apt update

sudo apt install python3-pip

sudo apt install pipenv

sudo apt install virtualenv

mkdir mlflow

cd mlflow

pipenv install mlflow

pipenv install awscli

pipenv install boto3

pipenv shell
```

### Then set aws credentials

```
aws configure
```

## Finally

```
mlflow server -h 0.0.0.0 --default-artifact-root s3://mlflowuritracking1  
mlflow server --host 0.0.0.0 --port 5000 --default-artifact-root s3://mlflowuritracking1
```

## open Public IPv4 DNS to the port 5000

### set uri in your local terminal and in your code

<<<<< HEAD

### set uri in your local terminal and in your code

```
export MLFLOW_TRACKING_URI=http://ec2-56-228-81-180.eu-north-1.compute.amazonaws.com:5000/
```

## Start of readme content

=====

```
export MLFLOW_TRACKING_URI=http://ec2-56-228-81-180.eu-north-1.compute.amazonaws.com:5000/
```

## (NEW) Export Username and Password for MLflow Authentication

If your MLflow server is configured with basic authentication, you can set the username and password as environment variables for your experiment scripts to use.

(If not configured, these variables will have no effect.)

```
export MLFLOW_TRACKING_USERNAME=dummyuser  
export MLFLOW_TRACKING_PASSWORD=dummypassword
```

In your Python code, MLflow will pick these up automatically if your version supports HTTP basic auth environment variables.

## Features

- **Remote Experiment Tracking:** Centralize your ML experiment logs and artifacts on a cloud-based MLflow server.
- **AWS EC2 Deployment:** Step-by-step instructions for setting up an EC2 instance as your MLflow tracking server.
- **Secure Connection:** Guidance on networking, security group setup, and authentication for safe remote tracking.
- **Example Usage:** Example scripts to log experiments remotely from your local machine or another server.

## Contents

- `mlflow_server_setup/`: Scripts and guides to set up MLflow on AWS EC2.
- `examples/`: Code examples for logging experiments to the remote MLflow server.
- `requirements.txt`: Python dependencies for running MLflow and sample experiments.

## Getting Started

---

### Prerequisites

---

- AWS account with permissions to launch EC2 instances.
- Basic knowledge of AWS EC2, SSH, and Python.
- [MLflow](#) installed locally (`pip install mlflow`).

### 1. Launch and Set Up EC2 Instance

---

1. Launch an EC2 instance (recommended: Ubuntu 20.04, t2.micro or larger).
2. Open ports **5000** (MLflow default) and **22** (SSH) in your Security Group.
3. SSH into your instance:

```
sh ssh -i your-key.pem ubuntu@<EC2_PUBLIC_DNS>
```

### 2. Install MLflow and Dependencies on EC2

---

```
sudo apt update
sudo apt install python3-pip -y
pip3 install mlflow boto3
```

### 3. Start MLflow Tracking Server

---

```
mlflow server \
    --host 0.0.0.0 \
    --port 5000 \
    --backend-store-uri sqlite:///mlflow.db \
    --default-artifact-root s3://your-bucket/mlflow/
```

- For local artifact storage, use a local path in `--default-artifact-root`.

### 4. Configure Your Local ML Project

---

Set the MLflow tracking URI to your EC2 instance:

```
import mlflow
mlflow.set_tracking_uri("http://<EC2_PUBLIC_DNS>:5000")
```

### 5. (Optional) Set Username and Password for Remote MLflow Server

---

If your MLflow server requires basic authentication, set the following environment variables before running your code:

```
export MLFLOW_TRACKING_USERNAME=dummyuser
export MLFLOW_TRACKING_PASSWORD=dummypassword
```

## 6. Example: Logging an Experiment

---

See `examples/remote_tracking_example.py` for a full example.

## Security Notes

---

- Restrict access to port 5000 to trusted IPs only.
- For production, consider using HTTPS and authentication (see [MLflow docs](#)).

## License

---

This project is licensed under the MIT License.

## References

---

- [MLflow Tracking](#)
  - [AWS EC2 Documentation](#)
- 

*Happy Experiment Tracking!*

cd2e5a387541b45dccd3d5fe0d109d3611166562