

Grau Bioinformàtica
Curs 2024-2025
Distributed systems and web development
Django lab – AWS EC2 Setup

1. Work environment preparation

This tutorial will help you to create the simplest Django web application project and run it in a cloud AWS EC2 environment running Python. We are going to setup a Django configuration in a remote AWS cloud EC2 service.

First we need to setup and connect to an AWS session, open a web browser to manage the sandbox environment.

Check your inbox to find an invitation for AWS Academy courses. You will need to set up a canvas account, then you can login to AWS Academy services by using your e-mail and password:

<https://awsacademy.instructure.com/login/canvas>



Enter the **AWS Academy Learner Lab** course and go to **modules** option in the left menu:

[Home](#)[Collapse All](#)[Modules](#)[Discussions](#)[▼ Course Welcome and Overview](#) [Pre-Course Survey](#) [AWS Academy Learner Lab Student Guide](#)[▼ AWS Academy Learner Lab Compliance and Security](#)[Complete All Items](#) [Learn how to effectively use the Academy Learner Lab](#) [Module Knowledge Check](#)

100 pts | Score at least 70.0

[▼ AWS Academy Learner Lab](#) [Launch AWS Academy Learner Lab](#)

Scroll down to the bottom of the list of modules until you find the AWS Academy Learner Lab and open it. You will see a welcome page to Learner Lab Sandbox environment overview.

Agree with the terms and conditions and wait for your lab environment Lab to be ready. You just need to start it.

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area shows the path: ALLv1-26771 > Modules > Learner Lab > Learner Lab. At the top right are buttons for Start Lab (highlighted with a red circle), End Lab, AWS Details, Readme, Reset, and a close button. A timer shows 03:48. Below the buttons is a message indicating \$0 used of \$100. The central part of the screen displays a terminal window titled 'ddd_v1_w_3dy_1435824@runweb63199:~\$' with the text 'EN-US -'. The main content area is titled 'Learner Lab' and lists various links related to AWS environments and tools.

Learner Lab

- [Environment Overview](#)
- [Environment Navigation](#)
- [Access the AWS Management Console](#)
- [Region restriction](#)
- [Service usage and other restrictions](#)
- [Using the terminal in the browser](#)
- [Running AWS CLI commands](#)
- [Using the AWS SDK for Python](#)
- [Preserving your budget](#)
- [Accessing EC2 Instances](#)
- [SSH Access to EC2 Instances](#)
- [SSH Access from Windows](#)
- [SSH Access from a Mac](#)

Use the Start lab button and wait for AWS infrastructure to be built (some minutes). AWS lab bullet color will change from red to green. You will see that there is a timer set at 4:00 hours.

The screenshot shows the AWS Academy Sandbox Environment interface. The sidebar includes icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area shows the path: ACFv2EN-26396 > Modules > Sandbox > Sandbox Environment. A 'Start Lab' dialog box is open, displaying session details: Region: us-east-1, Lab ID: arn:aws:cloudformation:us-east-1:372825523442:stack/c62179a114735712683788t1w372825523442/583515f0-3d77-11ed-b626-0add2ae7e6b9, Creation Time: 2022-09-26T01:43:51-0700. It also shows the start time (2022-09-26T01:43:52-0700) and remaining session time (03:00:00/180 minutes). A note indicates the lab status is 'in creation'.

When lab is finally started, we need the connection details to configure our connection from a local Linux terminal to AWS Cloud services. For that, press Details menu and look for **AWS details** button, you will see a Credentials window:

The screenshot shows the AWS Cloud9 interface. At the top, there are tabs for Home, Modules, and Discussions. Below these, the AWS Details tab is active, indicated by a red circle. The main content area displays a terminal session output and various configuration details.

```
yK+AbMmkPKDjnQ8Hvcey1l07FKbjZ1KhjCjchuxL+CJnKwtWeJE  
gyV1acfC1iipstfmucV7gWI1gWxTrZNhRmVU1ASFjyzzrxcUD1  
SqB8cGT6KCZwtTAVvs+Qyqjffxw+w6N+Rog5RG91108SDmwCv0M  
7KtueqpsP/VVSWDD44jX5yasLmWKw0aWuyuZzPV1Ioa8dPEu1r  
r/p8KN7wG14BH4nn54mCxu0M1X6ekw+7m2gbwTN6qwmSs/3wBY  
yj0o7ZLwmQYyLeNJ/hPrML7IBhXJPcxoNngG0cYvGJ8bJF83YN6  
rSdIbzRdFSci3k6Tx8dprNQ==
```

Cloud Labs
 Remaining session time: 03:56:30(237 minutes)
 Session started at: 2022-10-04T03:22:35-0700
 Session to end at: 2022-10-04T07:22:35-0700
 Accumulated lab time: 07:59:32 (480 minutes)
 No running instance

SSH key Show Download PEM Download PPK
 AWS SSO Download URL

AWSAccountid	398375560269
Region	us-east-1

The screenshot shows the AWS Cloud9 Credentials dialog box. It contains sections for Cloud Access (AWS CLI, Cloud Labs) and Cloud Labs (Remaining session time: 02:58:19(179 minutes), Session started at: 2022-09-26T01:43:52-0700, Session to end at: 2022-09-26T04:46:32-0700, Accumulated lab time: 06:43:30 (404 minutes)).

AWS CLI:
 Show Download PEM Download PPK
 AWS SSO Download URL

SecretKey	bqCv1K//Pn7yIpqJgC8vEhRsN2uiJS3xh506v4m/
BastionHost	54.89.30.94
Region	us-east-1
AccessKey	AKIAVNTRCATZOHNGFRGR

Now press **AWS CLI: Show** grey button:

The screenshot shows the AWS Cloud9 Credentials dialog box after pressing the 'Show' button for AWS CLI. It displays the generated AWS CLI configuration text:

```
[default]  
aws_access_key_id=ASIAVNTRCATZLGICTMF5  
aws_secret_access_key=vb3HxyesWNgX+aWT6BwWiK1+juZrb9c4tnZclc07  
aws_session_token=FwoGXIVyXdzEF1adB82fgHjB939iFDKxiK+AfSMS/UdNcT0dvmM4r+1D8IxkwaEN8PU58sfvGQxP9aQ3X3pLRCLTK  
oeP3f/nkMvD0oqBRV4VOAaTeqpAMrAY7jmpUZjj4ijZ7AP8/gMC8haAGuz7MZAne5nd//DKuqXJnrAnXWz5Cet1ietasgt4SSDmKbYPiTw1  
CLMaFxSevBPymsA8-3DcLaVZI08DRhWTKEl-aiAFAU6Gd7xpdaFFvComJ4zwCx5GTvp/01qMC04Hezt+lzipWtxXKIoyczFmQYyLbheX5  
XiKhVmZGr4Ij5Y+UkLYzZYsCmhNg7btRSAnZtaU5W+G+9lwJXRiikG==
```

Cloud Labs
 Remaining session time: 02:58:19(179 minutes)
 Session started at: 2022-09-26T01:43:52-0700
 Session to end at: 2022-09-26T04:46:32-0700
 Accumulated lab time: 06:43:30 (404 minutes)

copy all text under AWS CLI: to paste it in a new credentials file:

Go back to your local terminal and create a new `.aws` directory and a new `credentials` file:

```
cd  
mkdir .aws  
gedit .aws/credentials  
<paste AWS CLI text here>
```

paste the text under AWS CLI and save the file and close editor. These are the credentials we will use to connect to AWS console.

You can press the AWS green button to open the AWS Dashboard of cloud services in use.

2. Creation of a new EC2 instance for our Django application

Go to Launch AWS Academy Learner Lab page and check that AWS services state is green:

The screenshot shows the AWS Academy Learner Lab interface. At the top, there's a navigation bar with 'Home', 'Modules', and 'Discussions'. On the right, there are buttons for 'Start Lab', 'End Lab', 'AWS Details', 'Readme', 'Reset', and a close button. Below this, it says 'Used \$0 of \$100'. In the center, there's a terminal window with a command like 'ddd_v1_w_3dy_1435824@runweb63277:~\$' followed by a long string of characters. To the right of the terminal, there's a 'Cloud Labs' section showing session details: 'Remaining session time: 03:56:30(237 minutes)', 'Session started at: 2022-10-04T03:22:35-0700', and 'Session to end at: 2022-10-04T07:22:35-0700'. It also shows 'Accumulated lab time: 07:59:32 (480 minutes)'. Below this, there are buttons for 'SSH key' (with 'Show', 'Download PEM', 'Download PPK'), 'AWS SSO' (with 'Download URL'), and a large text box containing 'AWSAccountId' (398375560269) and 'Region' (us-east-1). A red circle highlights the 'AWS' status indicator in the top right corner of the main interface.

Click on AWS green state to launch AWS services console home

The screenshot shows the AWS Services Console Home page. At the top, there's a navigation bar with 'AWS' (highlighted), 'Services' (highlighted), a search bar ('Search'), and other options like '[Option+S]', a refresh icon, a gear icon, and 'N. Virginia'. Below this, the main area has a 'Console Home' title and a 'Recently visited' section with links to 'EC2' and 'IAM'. To the right, there's a 'Applications' section showing 'No applications' and a 'Create application' button. Further down, there are sections for 'Welcome to AWS' (with a 'Getting started with AWS' link and a rocket ship icon), 'AWS Health' (showing 'Open issues 0' and 'Past 7 days'), and 'Cost and usage' (showing 'Current month costs \$0.03' and a progress bar for 'Forecasted month end costs'). A 'View all services' button is at the bottom left, and a 'Go to myApplications' button is at the bottom right.

Now search for EC2 service and click on EC2 logo to create a new EC2 instance.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Global View, Events, Console-to-Code, Instances, Images, Elastic Block Store, and Network & Security. The main area has a 'Resources' summary table and a 'Launch instance' section. The 'Launch instance' section contains a 'Launch instance' button, a 'Migrate a server' button, and a note about launching in the US East (N. Virginia) Region. To the right, there's an 'Account attributes' section with settings for Default VPC, Settings, Data protection and security, Zones, EC2 Serial Console, Default credit specification, and EC2 console preferences. Below that is an 'Explore AWS' section with a note about Amazon GuardDuty Malware Protection, a link to 10 things to do to reduce AWS costs, and a note about getting 40% better price performance.

Click on **Launch instance** button to start the creation of our new cloud session:

Give a name to the EC2 instance, in our case we are naming it: **django server 1**

The screenshot shows the 'Launch an instance' wizard. Step 1: Name and tags. It shows a 'Name' field with 'django server 1' and an 'Add additional tags' button. Step 2: Application and OS Images (Amazon Machine Image). It shows a search bar with 'Search our full catalog including 1000s of application and OS images', a 'Recents' section with 'Quick Start' selected, and a grid of OS icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. To the right, there's a 'Summary' section with details like Number of instances (1), Software Image (AMI) (Amazon Linux 2023 AMI 2023.5.2...), Virtual server type (instance type) (t2.micro), Firewall (security group) (New security group), Storage (volumes) (1 volume(s) - 8 GiB), and a note about the Free tier. At the bottom are 'Cancel', 'Launch instance', and 'Review commands' buttons.

Leave instance type in t2.micro

Select Key pair (login): vockey

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID	Username	
64-bit (x86) ▼	uefi-preferred	ami-0ebfd941b bafe70c6	ec2-user	

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

vockey

[Create new key pair](#)

In network settings, leave parameters as they are, and **select Allow HTTP traffic from the internet**

▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-0053b3ea724138a0b

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of **free tier allowance**

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

X

Leave the rest of the parameters as they are, and click on "Launch instance"

Configure storage Info Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

Advanced details Info

virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance Review commands

After few seconds, you should have your new instance created

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-03b2418547fb1f819)

▶ Launch log

Click on instances to see a summary of your current cloud servers you have running

Instances (1) Info

Last updated 10 minutes ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
django server 1	i-03b2418547fb1f819	Running	t2.micro	Initializing	View alarms	us-east-1b

Select an instance

Now click on Instance ID (the blue number) to check the summary of our new running instance

EC2 Dashboard X

EC2 Global View

Events

Console-to-Code [Preview](#)

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations [New](#)

Images

- AMIs
- AMI Catalog

Elastic Block Store

- Volumes
- Snapshots
- Lifecycle Manager

Network & Security

- Security Groups

EC2 > Instances > i-03b2418547fb1f819

Instance summary for i-03b2418547fb1f819 (django server 1) [Info](#)

Updated less than a minute ago

Actions	
C	Connect
Instance state	Actions
Updated less than a minute ago	
Instance ID i-03b2418547fb1f819 (django server 1)	Public IPv4 address 98.82.202.103 open address
IPv6 address -	Instance state Running
Hostname type IP name: ip-172-31-44-216.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-44-216.ec2.internal
Answer private resource DNS name IPv4 (A) Auto-assigned IP address 98.82.202.103 [Public IP]	Instance type t2.micro
IAM Role -	VPC ID vpc-0053b3ea724138a0b
IMDSv2 Required	Subnet ID subnet-06fd532903ce34663
	Instance ARN arn:aws:ec2:us-east-1:487499254270:instance/i-03b2418547fb1f819
	Elastic IP addresses -
	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.
	Learn more
	Auto Scaling Group name -

We need to create an inbound rule to allow web traffic coming in. Click on the **Security** tab, then in **Security groups**

EC2 > Instances > i-03b2418547fb1f819

Instance summary for i-03b2418547fb1f819 (django server 1) [Info](#)

Updated 36 minutes ago

Actions	
C	Connect
Instance state	Actions
Updated 36 minutes ago	
Instance ID i-03b2418547fb1f819 (django server 1)	Public IPv4 address 98.82.202.103 open address
IPv6 address -	Instance state Running
Hostname type IP name: ip-172-31-44-216.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-44-216.ec2.internal
Answer private resource DNS name IPv4 (A) Auto-assigned IP address 98.82.202.103 [Public IP]	Instance type t2.micro
IAM Role -	VPC ID vpc-0053b3ea724138a0b
IMDSv2 Required	Subnet ID subnet-06fd532903ce34663
	Instance ARN arn:aws:ec2:us-east-1:487499254270:instance/i-03b2418547fb1f819
	Elastic IP addresses -
	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
	Auto Scaling Group name -

Details | **Status and alarms** | **Monitoring** | **Security** | **Networking** | **Storage** | **Tags**

Security details

IAM Role -	Owner ID 487499254270	Launch time Wed Sep 18 2024 18:52:23 GMT+0200 (hora de verano de Europa central)
Security groups	sg-095572e61437ff310 (launch-wizard-2)	

Inbound rules

Filter rules						
Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-0a3b97012692df8d	22	TCP	0.0.0.0/0	launch-wizard-2	-
-	sgr-0be077dbe12b747e0	80	TCP	0.0.0.0/0	launch-wizard-2	-

Click on Edit inbound rules box

The screenshot shows the AWS Security Groups page for a specific security group. At the top, it displays the security group name 'sg-095572e61437ff310 - launch-wizard-2'. Below this, there's a 'Details' section with fields for Security group name, Security group ID, Description, Owner, Inbound rules count, Outbound rules count, and VPC ID. Under the 'Inbound rules' tab, there's a table showing two existing rules: one for SSH (TCP port 22) and one for HTTP (TCP port 80). There are buttons for 'Edit inbound rules', 'Manage tags', and a search bar.

Now Add a new Custom TCP rule with the following parameters,

Type: Custom TCP

Port range: 8080

Source: Anywhere-IPv4, 0.0.0.0/0

and then press **Save rules**

The screenshot shows the 'Edit inbound rules' configuration page. It lists three existing rules and allows adding a new one. The new rule is being added with the following parameters:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0a3b979012692df8d	SSH	TCP	22	Custom	0.0.0.0/0
sgr-0be077dbe12b74e0	HTTP	TCP	80	Custom	0.0.0.0/0
-	Custom TCP	TCP	8000	Anywhere-IPv4	0.0.0.0/0

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." There are 'Cancel', 'Preview changes', and 'Save rules' buttons at the bottom right.

Go back to EC2 main page, click on Instances running, and click again in the instance ID.

To open a console connection to the running instance, we must click on **connect box**

Instance summary for i-03b2418547fb1f819 (django server 1) Info		
C Connect Instance state Actions		
Updated less than a minute ago		
Instance ID i-03b2418547fb1f819 (django server 1)	Public IPv4 address 98.82.202.103 open address	Private IPv4 addresses 172.31.44.216
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-98-82-202-103.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-44-216.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-44-216.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 98.82.202.103 [Public IP]	VPC ID vpc-0053b3ea724138a0b	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-06fd532903ce34663	
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:487499254270:instance/i-03b2418547fb1f819	

And then click on connect yellow button to open an ssh session to our EC2 instance, we should see a welcome console message:

```
'      #
`-\_ ####
--\_###\ \
-- \##|
--   '#/
--     V~'__->
--- ._.
--- /`-/
--- /m/`-
ec2-user@ip-172-31-44-216 ~]$
```

3. Django Installation

First, we need to install Python requirements for Django in our local system.

```
sudo yum install python-is-python3 -y  
  
sudo yum install pip -y  
  
pip install virtualenv
```

Second step: prepare a new Python application virtual environment. In this case, it will be named **polls-env**:

```
virtualenv polls-env  
  
source polls-env/bin/activate  
  
pip install django==2.2
```

Finally, we create a new local Django application using the templates

```
django-admin startproject mysite  
  
cd mysite/mysite  
  
nano settings.py
```

Edit ALLOWED_HOSTS LINE as shown in this example:

```
Django settings for mysite project.  
Generated by 'django-admin startproject' using Django 2.2.  
For more information on this file, see  
https://docs.djangoproject.com/en/2.2/topics/settings/  
For the full list of settings and their values, see  
https://docs.djangoproject.com/en/2.2/ref/settings/  
"""  
import os  
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)  
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))  
# Quick-start development settings - unsuitable for production  
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/  
# SECURITY WARNING: keep the secret key used in production secret!  
SECRET_KEY = ')644--hbsmcj)tbvgxihlmb&=4w)_@ctyxo$ma&r+qb$5v$o=t'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['*']
```

Save the file with control+X and confirm saving the changes

```
cd /home/ec2-user/mysite

python manage.py runserver 0:8080
Django version 2.2, using settings 'mysite.settings'
Starting development server at http://0:8080/
Quit the server with CONTROL-C.

[07/Sep/2018 20:14:09] "GET / HTTP/1.1" 200 16348
```

Go back to Instance summary page and click on **Public IPv4 address:** open address

EC2 > Instances > i-03b2418547fb1f819	
Instance summary for i-03b2418547fb1f819 (django server 1) Info	
Updated 5 minutes ago	
Instance ID	Public IPv4 address
i-03b2418547fb1f819 (django server 1)	98.82.202.103 open address
IPv6 address	Instance state
-	Running
Hostname type	Private IP DNS name (IPv4 only)
IP name: ip-172-31-44-216.ec2.internal	ip-172-31-44-216.ec2.internal
Answer private resource DNS name	Instance type
IPv4 (A)	t2.micro
Auto-assigned IP address	VPC ID
98.82.202.103 [Public IP]	vpc-0053b3ea724138a0b
IAM Role	Subnet ID
-	subnet-06fd532903ce34663
IMDSv2	Instance ARN
Required	arn:aws:ec2:us-east-1:487499254270:instance/i-03b2418547fb1f819

A new web browser connection to your instance will be opened.

Change the address to **http://<leave your IP address here>:8080**

In our example: <http://98.82.202.103:8080> to check the new Django created site:



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



[Django Documentation](#)
Topics, references, & how-to's



[Tutorial: A Polling App](#)
Get started with Django

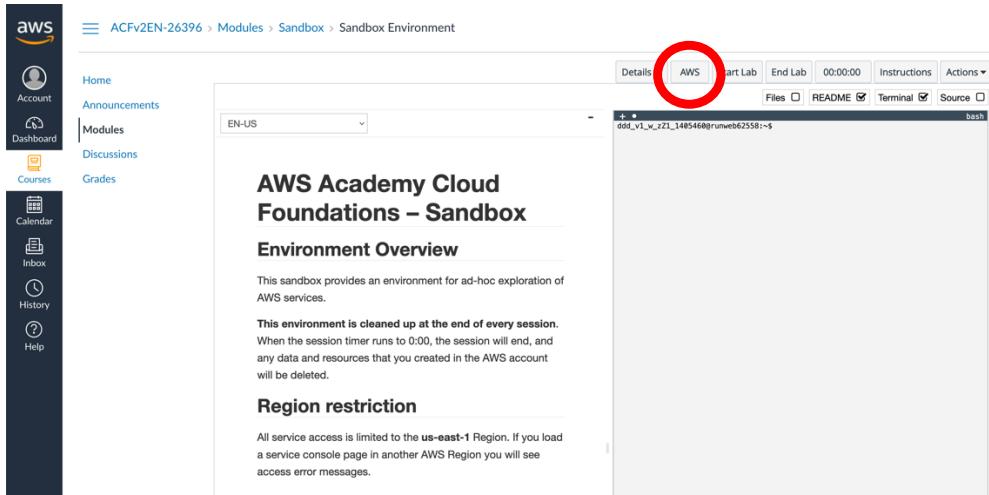


[Django Community](#)
Connect, get help, or contribute

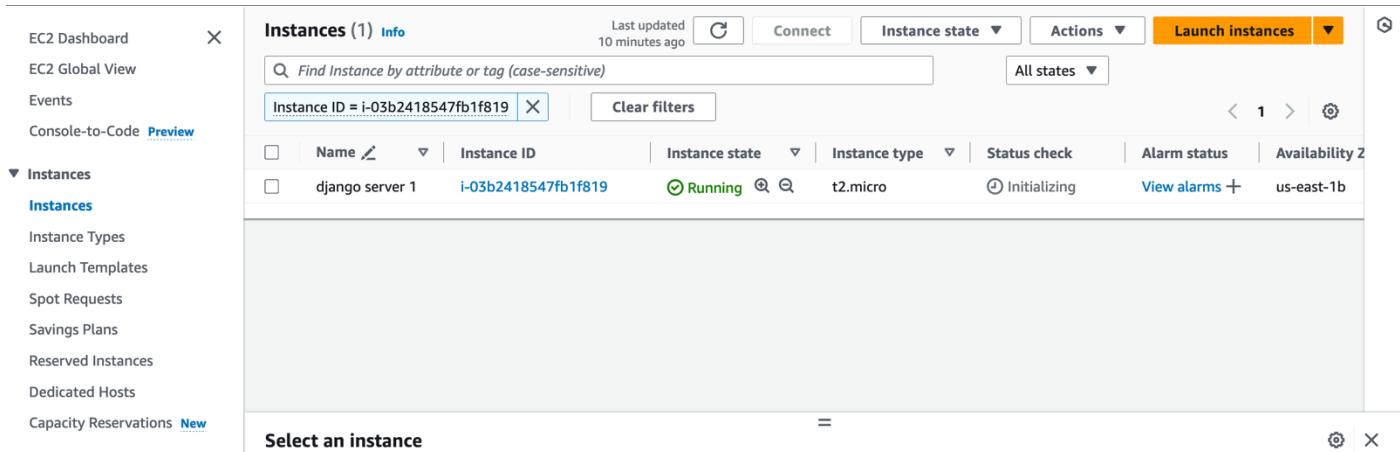
use `Ctrl+C` in terminal to stop the web server in the console to continue with the next tutorial.

4. Cleaning the environment and closing the Sandbox session

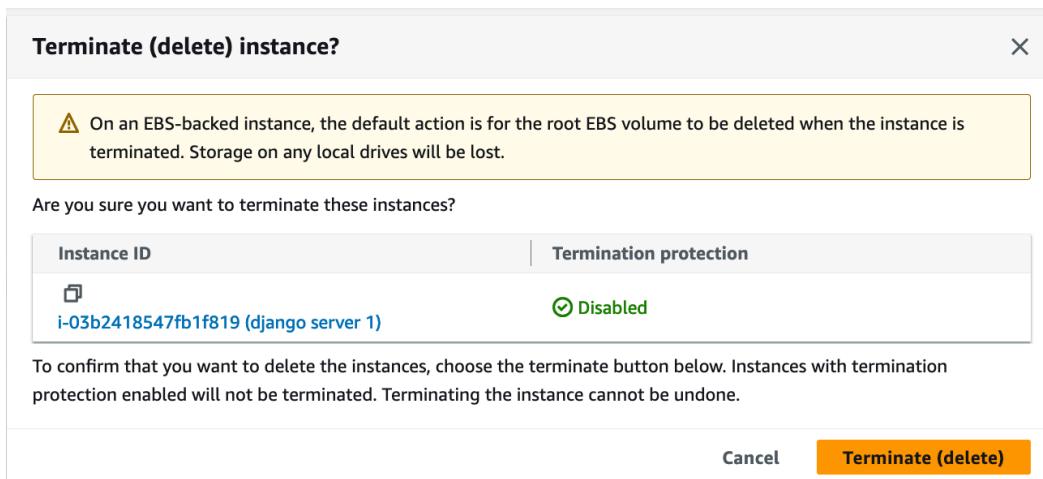
When you need to close the lab session, enter AWS Console using the AWS button in Sandbox window.



Search for EC2 Dashboard and select django server running instance:



Click on **Instance state** box and select **Terminate (delete)** instance



Click on **Terminate(delete)** and go back to Launch AWS Academy Learner Lab. Now, you need to press **End Lab in Sandbox web** application and logout of AWS Academy portal.

AWS Start Lab End Lab AWS Details Readme Reset

Used \$0 of \$100 03:30

Home Modules Discussions

Cloud Access

AWS CLI:
Copy and paste the following into ~/.aws/credentials

```
[default]
aws_access_key_id=ASIAYJUC65AW3PM5GTPT
aws_secret_access_key=KXxI1Q6/QSQ9RcNRmn2LR0M6wVh2YRbvQUiEqlw
t
aws_session_token=FwoGZXIvYXdzECMaDLQqwo8KWgK0/cpkviK9ATUMyjs
M/HajzEmYK3hSYnr6JwzooiLCe76Iucxg2EN+hbt/tnzdY3W8KSAYggBa0CcV
D31a7dmvsfzdJ5bkD8Lp3pvwy+6qc/rMh3MTxptQcQELi6W+E9RvHvENVW1ut
hMxd09jjxJEYWptJqlCc1XhjymDD5+81cNCobbyBA1mEUMYVaLBhUrG1Z7Rom
d4gp6so+x2UtCvRNhREKorySYy2o/0IXYe6VUmF61QDjbb01j/3Pt16HPvIa7
qZijm80uoBjItAFafN5DS7hIqhtRMr5LR6TKeDH0010Y5VJs9A7yvw78dz1q
gK1hSTe0EqVv
```