

CS6620 Fundamentals of Cloud Computing

E-Commerce SaaS Web Application -- InstaShop

Team Members:

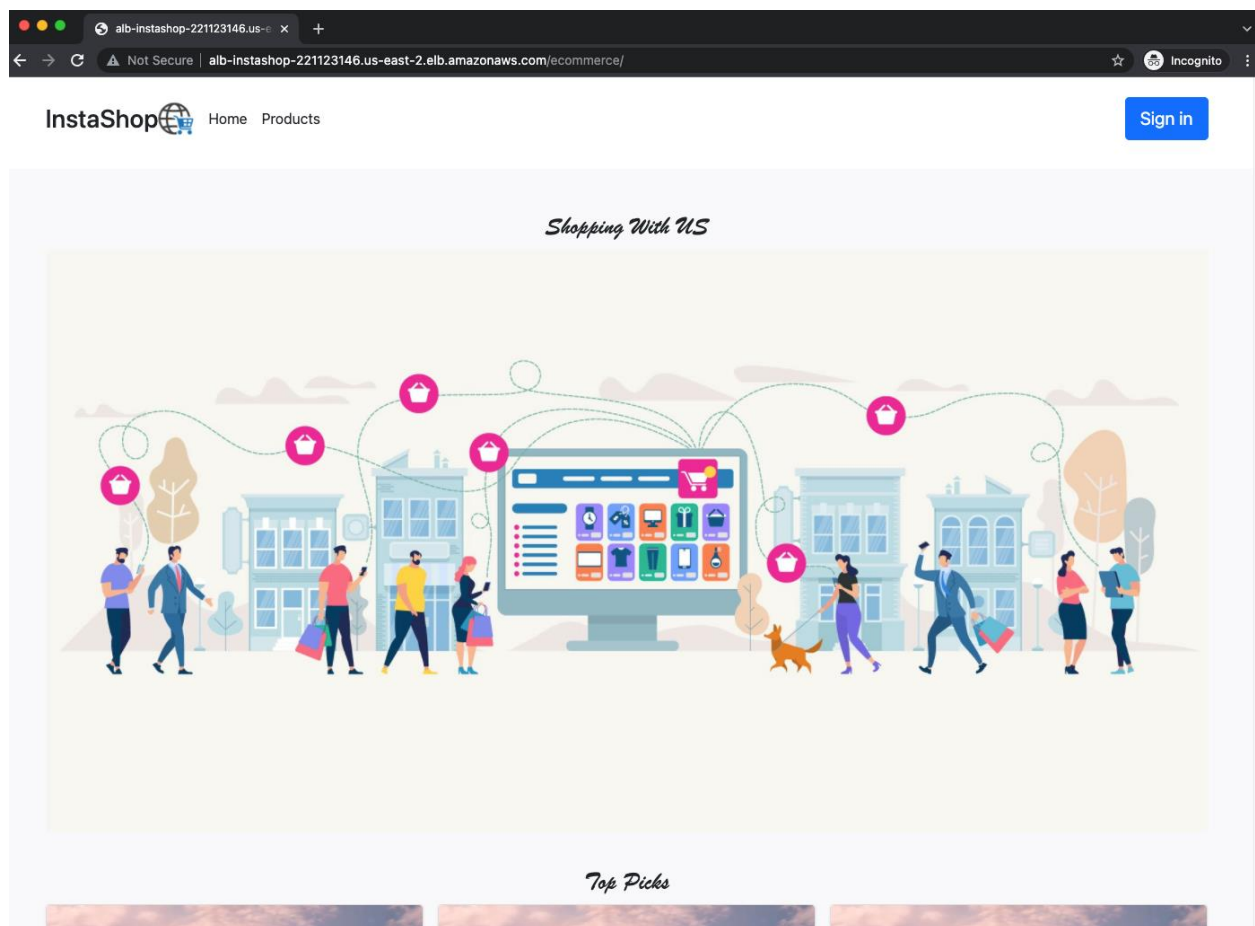
Guan Wang

Yu Feng

Qichen An

Wenhao Ge

https://github.com/singinzrain/cloud_computing_ecommerce



Project Description:

For years, on-premises solutions were synonymous with ecommerce. These solutions were installed on-site on company-owned servers. They promised innovation and customization — often by adding on hardware. This is how many companies built high-growth digital businesses, but they proved rigid. Over time, more IT resources were needed to support legacy systems. What started as a strength — the ability to customize — became a weakness. Years of one-off customizations made platforms less agile and inflexible. Unfortunately, ecommerce is the last place any company can afford to lose out. Businesses need to meet ever-changing customer expectations, security threats, and regulatory requirements. Ecommerce technology is a significant area of innovation. Difficult and costly product upgrades were another big challenge with legacy on-premises solutions. Even after investing considerable time, money, and resources, upgrades added little business value.

In the mid-2000s, cloud-based SaaS ecommerce platforms became available. In the following years, many ecommerce vendors shifted to a cloud strategy. And, new entrants emerged. A SaaS e-commerce platform is an e-commerce software delivered as a cloud-based system that can be accessed through all types of web browsers. Cloud-based SaaS ecommerce platforms could provide Scalability, Flexibility, Maintenance, Updates and Security, which push SaaS-based e-commerce to come to the forefront as exactly the solution businesses need.

Therefore, under such a background, our team collaborated on the final project (SaaS Ecommerce web application) by designing, implementing, and thoroughly testing on AWS. And finally, we ultimately finished the project – InstaShop.

InstaShop is a SaaS e-commerce website. Nowadays, the Internet has changed the traditional financial model. Along with it, there are various applications that facilitate people's lives. For example, delivery software has changed the traditional restaurant supply model and e-commerce platforms have changed the offline sales model. The design of InstaShop is to help people stay at home but could buy a variety of snacks, fruits, groceries, take-out, household appliances and so on. In an ideal world, we'll have our own delivery people and would partner with UberEats, Ikea, Walmart and so on. But for Instashop under the concept of CS6620 cloud engineering, our team will only do an e-commerce web application on the web side for the following use cases.

Use Case:

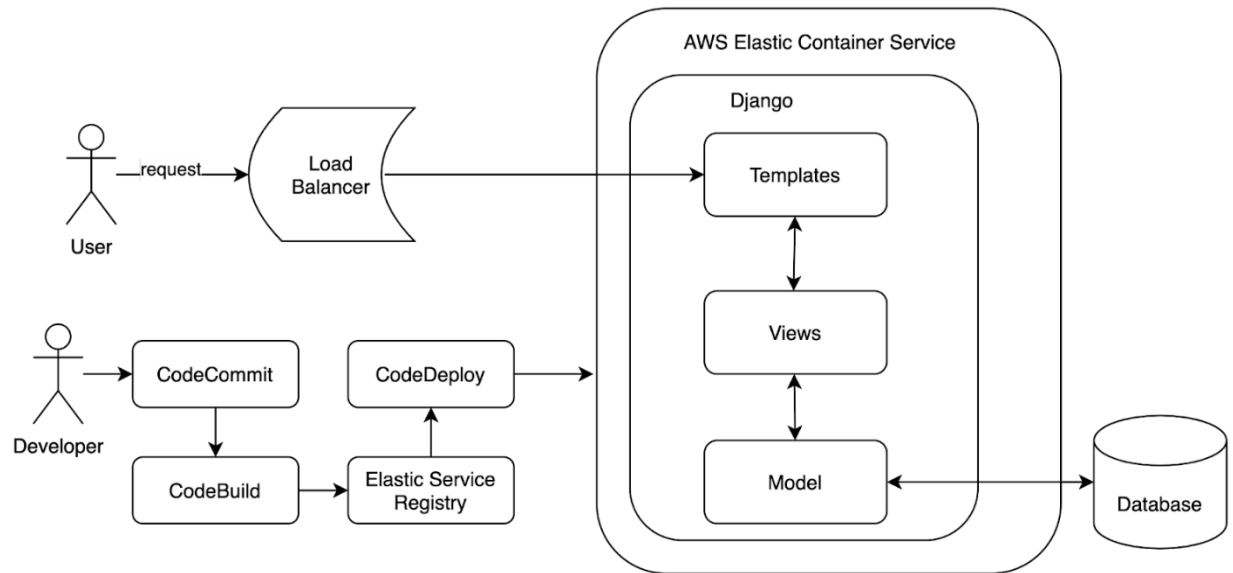
As customers go to our website, they will first see our home page which welcomes our customers and informs them with top picked products. They can view all the products as going to the products page. For each listed product, they can click on it which will direct them to a product detail page. From the navigation bar, they can sign up and login with their accounts. After they login, they can add products into the shopping cart, edit the shopping cart and then check out their orders. They can revisit their orders in the order history page.

Key Features:

- **Home page:** welcome customers and inform them with top 3 picked products
- **Product page:** showing all the products
- **Product details page:** showing more detail information for the product along with reviews
- **Shopping cart page:** showing a list of customer wanted products and the customer can increase or decrease the product count. The customer can also delete a product from the shopping cart
- **Check out page:** asking customers to fill out needed information and allowing them to check-out
- **History orders:** showing a list of history orders
- **Sign In/Sign Up:** allowing customers to create their account and login

Please note, we mention a search product feature in our project proposal. But, we think Sign In/Sign Up is a more important feature. Thus, we decided to implement the Sign In/Sign Up feature rather than the search product feature.

Architecture overview diagram (AOD) and design description



For a high-level view, users will send out requests which would be sent through a load balancer and directed to an AWS instance. In the AWS instance, our web application will be implemented with Django where requests would go through related templates, views, and models.

From the perspective of developers, AWS Codepipeline is leveraged. The code update will be pushed to CodeCommit repo, then CodeBuild is triggered to build a docker image and the image will be pushed to Elastic Service Registry. Finally, CodeDeploy is triggered to start the container from the image and deploy the service to ECS cluster.

Cloud Features:

For the implementation, our group use following AWS features:

1. AWS Elastic Container Service
 We deploy our containerized application service through ECS.
2. Load Balancer
 Route incoming end-users' traffic to applications based in the AWS cloud.
3. Amazon RDS
 We use AWS RDS to store our structured data.
4. Complete CI/CD
 Automate build and deployment with AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline.
5. AWS Lambda
 In our deployment process, we tried to use zappa to deploy our web app which involves AWS Lambda.

Deployment Process

Create Dockerfile for the application

```
FROM python:3
```

```
ENV PYTHONDONTWRITEBYTECODE=1
```

```
ENV PYTHONUNBUFFERED=1
```

```
WORKDIR /code
```

```
COPY requirements.txt /code/
```

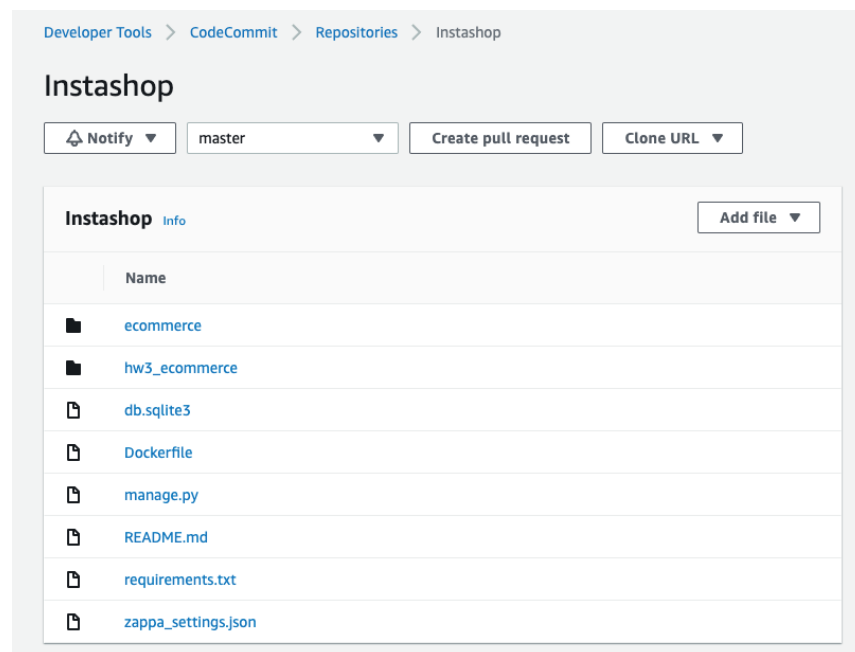
```
RUN pip install -r requirements.txt
```

```
COPY . /code/
```

```
EXPOSE 8000
```

```
CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

Create CodeCommit and upload the code files



Create Codebuild and make the source as the Instashop from CodeCommit, make sure the codebuild environment image is privileged

Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

Instashop

Reference type

Choose the source version reference type that contains your source code.

☒ Branch

☐ Git tag

☐ Commit ID

Branch

Choose a branch that contains the code to build.

master

Commit ID - optional

Choose a commit ID. This can shorten the duration of your build.

Source version [Info](#)

refs/heads/master

c2ec4638 remove submodule

► Additional configuration

Git clone depth, Git submodules

Add buildspec.yaml

```
version: 0.2
```

```
phases:
```

```
pre_build:
```

```
  commands:
```

```
    - echo Logging in to Amazon ECR...
```

```
    - $(aws ecr get-login --no-include-email --region $AWS_DEFAULT_REGION)
```

```
build:
```

```
  commands:
```

```
    - echo Build started on `date`
```

- echo Building the Docker image...

- docker build -t instashop:1 .

- docker tag instashop:1 632547974315.dkr.ecr.us-east-2.amazonaws.com/yufengcloud

post_build:

commands:

- echo Build completed on `date`

- echo Pushing the Docker image...

- docker push 632547974315.dkr.ecr.us-east-2.amazonaws.com/yufengcloud

artifacts:

files: imagedefinition.json

Add imagedefinition.json



```
[
{
  "name": "instashop",
  "imageUri": "632547974315.dkr.ecr.us-east-2.amazonaws.com/yufengcloud:latest"
}
]
```

Add policy to the codebuild role

[Roles](#) > [codebuild-Instashop-build-service-role](#)

Summary

Delete role

Role ARN	arn:aws:iam::632547974315:role/service-role/codebuild-Instashop-build-service-role 
Role description	Edit
Instance Profile ARNs	
Path	/service-role/
Creation time	2021-12-09 15:38 PST
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions

Trust relationships

Tags


Access Advisor

Revoke sessions

▼ Permissions policies (2 policies applied)

Attach policies

[+ Add inline policy](#)

Policy name ▼	Policy type ▼	
▶  AmazonEC2ContainerRegistryFullAccess	AWS managed policy	✕
▶ CodeBuildBasePolicy-Instashop-build-us-...	Managed policy	✕

Successfully built image

[Developer Tools](#) > [CodeBuild](#) > [Build projects](#) > [Instashop-build](#) > [Instashop-build:9940a50e-c467-42a8-b829-01215a40eba5](#)

Instashop-build:9940a50e-c467-42a8-b829-01215a40eba5

[Stop build](#) [Retry build](#)

Build status

Status	Initiator	Build ARN
✓ Succeeded	project2-yu	arn:aws:codebuild:us-east-2:632547974315:build/Instashop-build:9940a50e-c467-42a8-b829-01215a40eba5
Resolved source version	Start time	End time
03fdeef9d418c70b5f30b459aa778235e933de73	Dec 9, 2021 4:48 PM (UTC-8:00)	Dec 9, 2021 4:50 PM (UTC-8:00)
Build number		
7		

Verify the image in ECR

[Amazon ECR](#) > [Repositories](#) > [yufengcloud](#)

yufengcloud

[View push commands](#) [Edit](#)

Images (12) [Refresh](#) [Delete](#) [Scan](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Image tag	Pushed at ▼	Size (MB) ▼	Image URI	Digest	Scan status
<input type="checkbox"/>	latest	December 09, 2021, 16:50:07 (UTC-08)	365.71	Copy URI	sha256:279c9f03...	-

Create task

Task Definitions > instashop-task > 1

Task Definition: instashop-task:1

View detailed information for your task definition. To modify the task definition, you need to create a new revision and then make the required changes to the task definition

Create new revision

Actions

BuilderJSONTags

Task definition name

instashop-task

Task role

None

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#)

Network mode

awsvpc

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

Operating system family

Linux

Compatibilities

EC2, FARGATE

Requires compatibilities

FARGATE

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the `ecsTaskExecutionRole` already, we can create one for you.

Task execution role

ecsTaskExecutionRole

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 or External launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (MiB)

512

Task CPU (unit)

256

Task memory maximum allocation for container memory reservation

0512 shared of 512 MiB

Task CPU maximum allocation for containers

0256 shared of 256 CPU units

Task Placement

Constraint

No constraints

Container definitions

Container Na...	Image	CPU Units	GPU	Inference Acc...	Hard/Soft memory limits (MiB)	Essential
instashop	632547974315...	0			--/--	true

Volumes

Create ALB

Create Load BalancerActions

Filter by tags and attributes or search by keyword

<<1 to 1 of 1>>

Name	DNS name	State	VPC ID
alb-instashop	alb-instashop-221123146.us-east-2.elb.amazonaws.com	Active	vpc-dfb9d2b4

Load balancer: alb-instashop

DescriptionListenersMonitoringIntegrated servicesTags

Basic Configuration

Name

alb-instashop

ARN

arn:aws:elasticloadbalancing:us-east-2:632547974315:loadbalancer/app/alb-instashop/4ff7b7b8df468a2a

DNS name

alb-instashop-221123146.us-east-2.elb.amazonaws.com
(A Record)

State

Active

Type

application

Scheme

internet-facing

IP address type

ipv4
Edit IP address type

VPC

vpc-dfb9d2b4

Availability Zones

subnet-0d467641 - us-east-2c

Create Cluster

Clusters > instashop-cluster

Cluster : instashop-cluster

Update Cluster

Delete Cluster

Get a detailed view of the resources on your cluster.

Cluster ARNarn:aws:ecs:us-east-2:632547974315:cluster/instashop-cluster

StatusACTIVE

Registered container instances0

Pending tasks count0 Fargate, 0 EC2, 0 External

Running tasks count2 Fargate, 0 EC2, 0 External

Active service count1 Fargate, 0 EC2, 0 External

Draining service count0 Fargate, 0 EC2, 0 External

ServicesTasksECS InstancesMetricsScheduled TasksTagsCapacity Providers

Run new TaskStopStop AllActions

Last updated on December 9, 2021 10:11:22 PM (0m ago)

Desired task status:RunningStopped

Filter in this page

Launch typeALL

< 1-2 > Page size50

	Task	Task defini...	Container i...	Last status...	Desired st...	Started at ...	Started By ...	Group	Launch typ...	Platform v...
<input type="checkbox"/>	716bb69d2f...	instashop-t...	--	RUNNING	RUNNING	2021-12-09...	ecs-svc/08...	service:inst...	FARGATE	1.4.0
<input type="checkbox"/>	eba5d2010...	instashop-t...	--	RUNNING	RUNNING	2021-12-09...	ecs-svc/08...	service:inst...	FARGATE	1.4.0

Clusters > instashop-cluster

Cluster : instashop-cluster

Update Cluster

Delete Cluster

Get a detailed view of the resources on your cluster.

Cluster ARNarn:aws:ecs:us-east-2:632547974315:cluster/instashop-cluster

StatusACTIVE

Registered container instances0

Pending tasks count0 Fargate, 0 EC2, 0 External

Running tasks count2 Fargate, 0 EC2, 0 External

Active service count1 Fargate, 0 EC2, 0 External

Draining service count0 Fargate, 0 EC2, 0 External

ServicesTasksECS InstancesMetricsScheduled TasksTagsCapacity Providers

CreateUpdateDeleteActions

Last updated on December 9, 2021 10:11:03 PM (0m ago)

Filter in this page

Launch typeALL

Service typeALL

< 1-1 >

	Service Name	Status	Service ty...	Task Defi...	Desired t...	Running t...	Launch ty...	Platform ...
<input type="checkbox"/>	instashop-service	ACTIVE	REPLICA	instashop-...	2	2	FARGATE	LATEST(1...

Create CodePipeline and execute successfully

Developer Tools

CodePipeline

▶ Source • CodeCommit

▶ Artifacts • CodeArtifact

▶ Build • CodeBuild

▶ Deploy • CodeDeploy

▼ Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

▶ Settings

Q Go to resource

Feedback

Developer Tools

CodePipeline

▶ Source • CodeCommit

▶ Artifacts • CodeArtifact

▶ Build • CodeBuild

▶ Deploy • CodeDeploy

▼ Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

▶ Settings

Q Go to resource

Feedback

Developer Tools > CodePipeline > Pipelines > instashop-pipeline

instashop-pipeline

Notify

Edit

Stop execution

Clone pipeline

Release change

Source

Succeeded

Pipeline execution ID: 82d9facd-79df-43f7-b969-5f79f9afe59ce

Source

1

AWS CodeCommit

Succeeded

- 14 minutes ago

be590d23

be590d23 Source: update allowed host

Disable transition

Build

Succeeded

Pipeline execution ID: 82d9facd-79df-43f7-b969-5f79f9afe59ce

Build

1

AWS CodeBuild

Succeeded

- 12 minutes ago

Details

be590d23 Source: update allowed host

Disable transition

Deploy

Succeeded

Pipeline execution ID: 82d9facd-79df-43f7-b969-5f79f9afe59ce

Deploy

1

Amazon ECS

Succeeded

- 11 minutes ago

Details

be590d23 Source: update allowed host

✓

✓

✓

AWS Relational Database Service

The screenshot shows the AWS RDS console for an instance named 'database-1'. The left sidebar contains navigation links for Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, Recommendations, and Certificate update. The main content area displays the instance summary and connectivity & security details.

Summary

DB identifier	CPU	Status	Class
database-1	5.25%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-2b

Connectivity & security

Endpoint & port	Networking	Security
Endpoint: database-1.ctbryzefccu.us-east-2.rds.amazonaws.com	Availability Zone: us-east-2b	VPC security groups: default (sg-a8d463e5) Active
Port: 3306	VPC: vpc-1c761c77	Publicly accessible: Yes
	Subnet group: default	Certificate authority: rds-ca-2019
	Subnets: subnet-5f724113, subnet-472de53a, subnet-84b527ef	Certificate authority date: August 22, 2024, 10:08 (UTC+10:08)

Security group rules (3)

Security group	Type	Rule
default (sg-a8d463e5)	EC2 Security Group - Inbound	sg-a8d463e5
default (sg-a8d463e5)	CIDR/IP - Inbound	73.231.38.204/32
default (sg-a8d463e5)	CIDR/IP - Outbound	0.0.0.0/0

The screenshot shows the AWS RDS console for the 'database-1' instance, displaying the 'Security group rules (3)', 'Replication (1)', and 'Proxies (0)' sections.

Security group rules (3)

Security group	Type	Rule
default (sg-a8d463e5)	EC2 Security Group - Inbound	sg-a8d463e5
default (sg-a8d463e5)	CIDR/IP - Inbound	73.231.38.204/32
default (sg-a8d463e5)	CIDR/IP - Outbound	0.0.0.0/0

Replication (1)

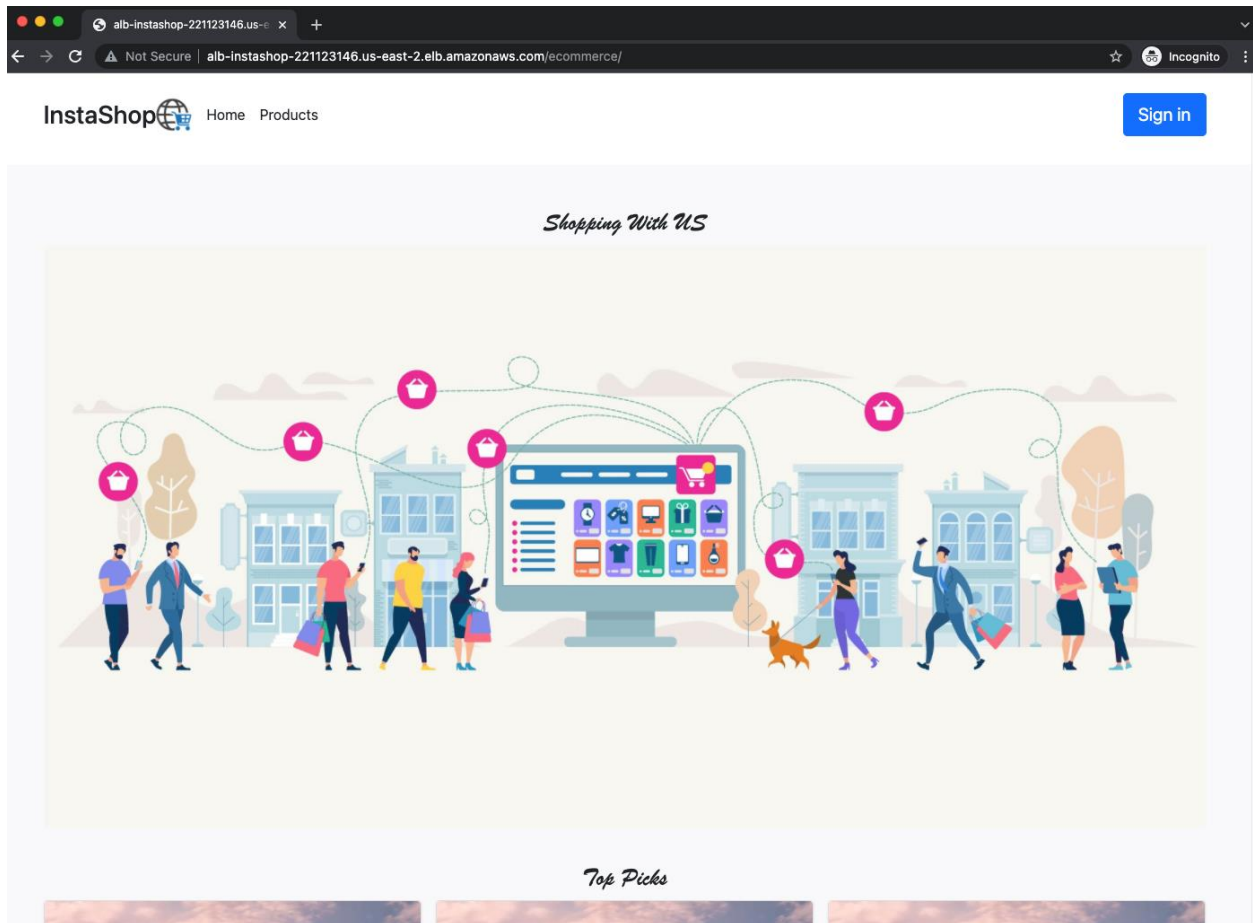
DB instance	Role	Region & AZ	Replication source	Replication state	Lag
database-1	Instance	us-east-2b	-	-	-

Proxies (0)

No proxies
You don't have any proxies.
[Create proxy](#)

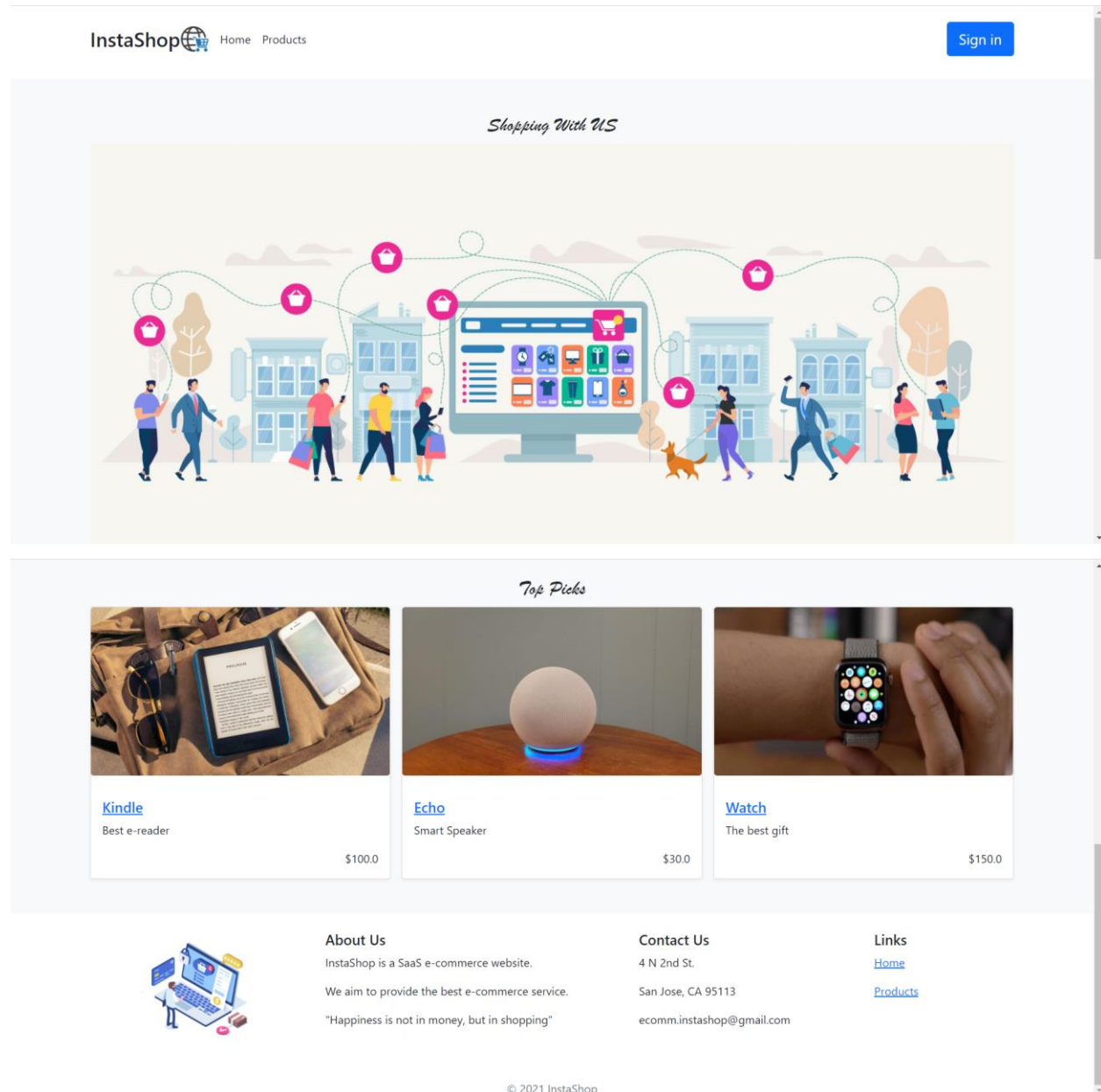
Verify the deployed web app is working

<http://alb-instashop-221123146.us-east-2.elb.amazonaws.com/ecommerce/>



UI & HCI

Home Page (without user logged in)



Sign In Page

Welcome to InstaShop!

Please sign in

<input type="text"/>
<input type="password"/>

[Sign in](#)

[New customer? Sign up here.](#)

© 2017–2021

Sign Up Page

Welcome to InstaShop!

Please sign up

<input type="text"/>
<input type="password"/>
<input type="password"/>

[Sign up](#)

[Already have an account? Sign in here.](#)

© 2017–2021

Here is an example, we create a guest for this system.

Welcome to InstaShop!

Please sign up

Username	ecommerceGuest
Password	*****
Confirm Password	*****

Sign up

[Already have an account? Sign in here.](#)

© 2017–2021

Welcome to InstaShop!

Please sign in

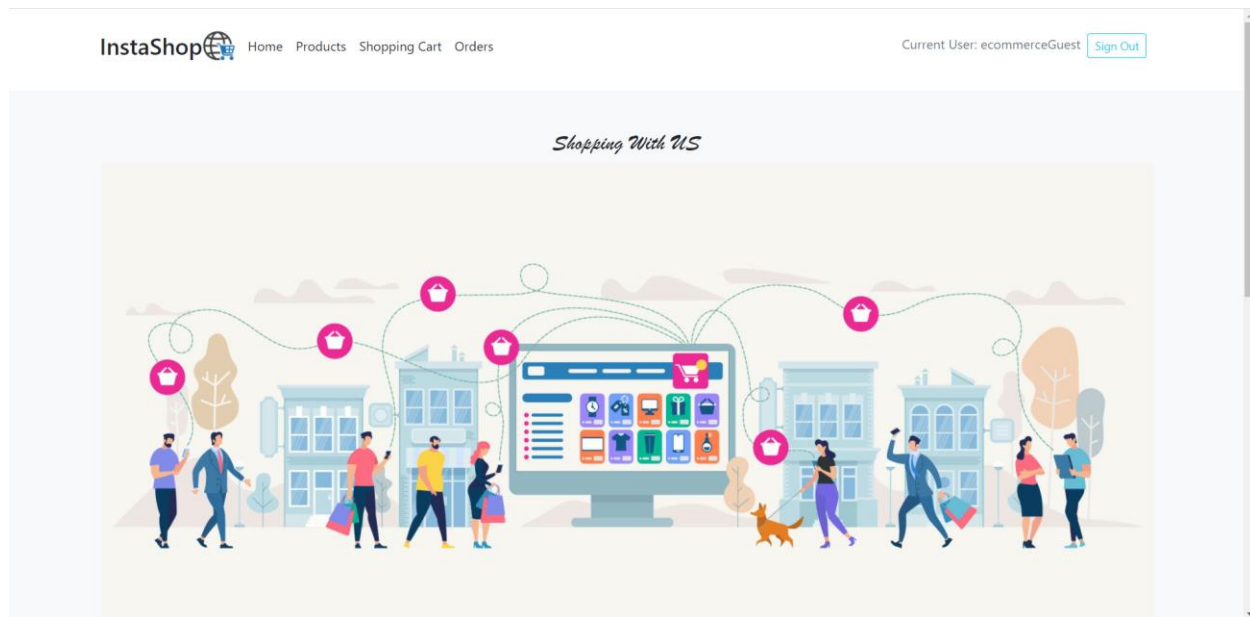
Username	ecommerceGuest
Password	*****

Sign in

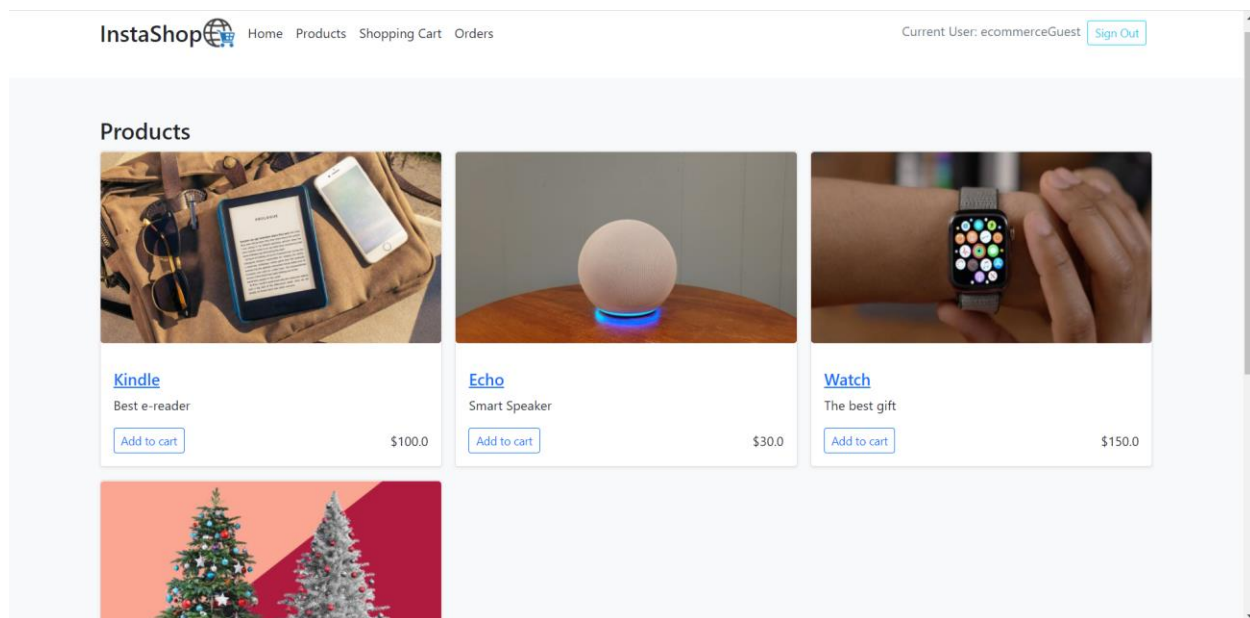
[New customer? Sign up here.](#)

© 2017–2021

Home Page with user logged in



Products Page



Add to cart


\$100.0

Add to cart

\$30.0

Add to cart


\$150.0



Artificial Tree
Special holiday memories!

Add to cart

\$299.0



About Us
InstaShop is a SaaS e-commerce website.
We aim to provide the best e-commerce service.
"Happiness is not in money, but in shopping"

Contact Us
4 N 2nd St.
San Jose, CA 95113
ecomm.instashop@gmail.com

Links
[Home](#)
[Products](#)
[Shopping Cart](#)


© 2021 InstaShop

Product Details Page

InstaShop

[Home](#)
[Products](#)
[Shopping Cart](#)
[Orders](#)

Current User: ecommerceGuest [Sign Out](#)



Kindle
Best e-reader

Price	\$100.0
Total	\$100.0

Your story starts here

"Go beyond a book"


Holds thousands of books
Fast downloads
At-a-tap controls

"Massive selection"

Finding new stories has never been easier with kindle.
Enjoy access to Kindle Exclusive titles you won't find anywhere else.

"Love it!!!"

Changing life is not an exaggeration. Books are friends.
Love the way they look! Love the way they feel! Love the way they smell!



Echo
Smart Speaker

Price	\$30.0
Total	\$30.0

Meet Echo

"Alexa, play top hits"


Echo combines premium sound, a built-in Zigbee smart home hub, and a temperature sensor. Powerful speakers deliver clear highs, dynamic mids, and deep bass for rich,

"Designed to protect your privacy"

Alexa and Echo devices are built with multiple layers of privacy protection. Echo smart speaker has

"Alexa is happy to help!"

Make your life easier at home. Set timers, ask questions, add items to lists, and create calendar events and reminders. Check the traffic and weather, or play the



Watch
The best gift

Price	\$150.0
Total	\$150.0

The ultimate device for life

"Full screen ahead"

Apple Watch can do what your other devices can't because it's on your wrist. When you wear it, you get a fitness partner that measures all the ways you move,

"Track all the ways you're active"

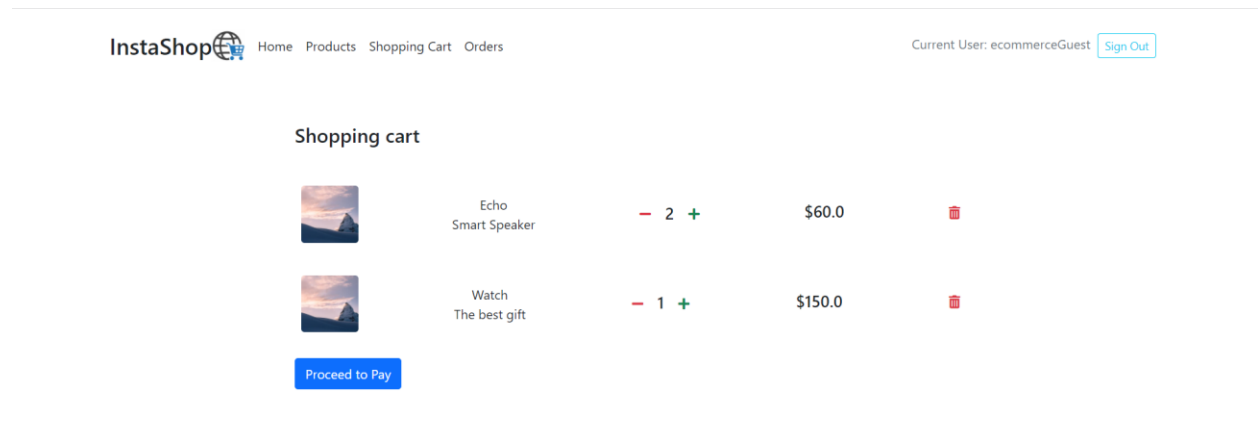
Activity rings show your daily activity. Make it your goal to close them every day. To keep you motivated there are

"Stay in touch with just a tap"

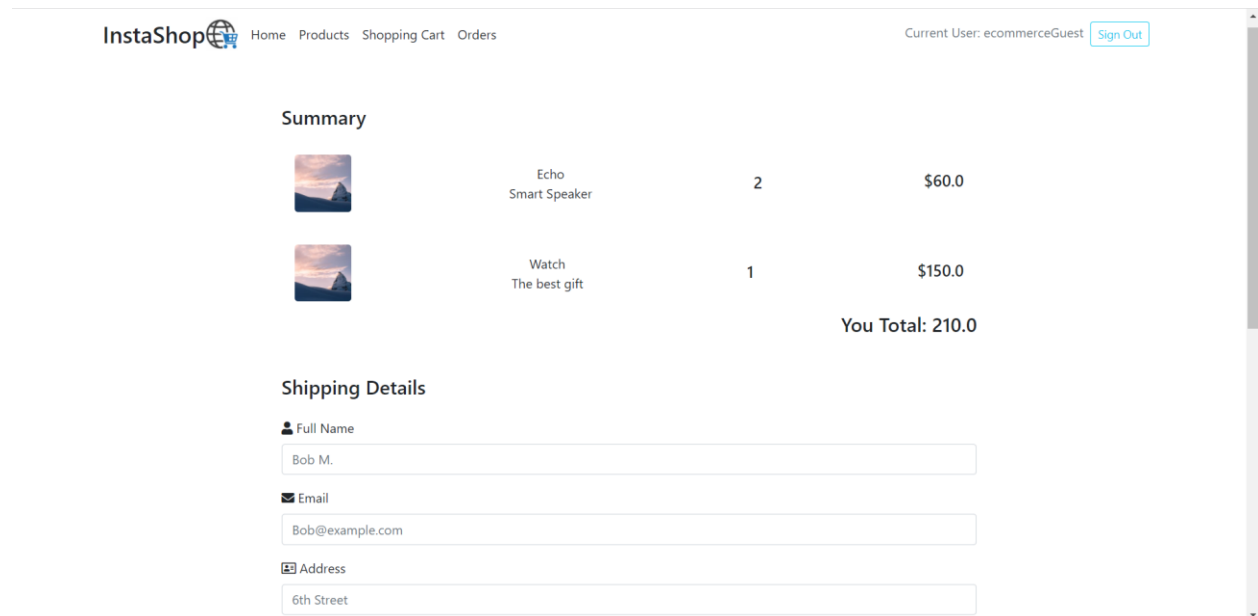
Text, make calls, and listen to music with ease. And with available cellular, you can do it all without your phone. It

\$150.0

Then, you can see the products are already in the shopping cart.



After click 'Proceed to Pay' button, you will be redirected to the checkout page. The checkout page will show you the summary of the order, and require the users filling shipping details and payment information.



6th Street

City

San Jose

State

CA

Zip

95112

Payment Information

Name on Card

Bob M.

Credit card number

1111-2222-3333-4444

Exp Month

September

Checkout

Then, you can see what you have ordered in the order page!

InstaShop
Home Products Shopping Cart Orders
Current User: ecommerceGuest Sign Out

Orders

Order Date/Time	Order Number	Order Total	Status	Details
2021-12-10	20	\$210.0	Preparing Shipment	Products (amount) Echo: 2 Watch: 1

Showing 1 to 1 of 1 entries



About Us

InstaShop is a SaaS e-commerce website.
We aim to provide the best e-commerce service.
"Happiness is not in money, but in shopping"

Contact Us


4 N 2nd St.
San Jose, CA 95113
ecomm.instashop@gmail.com

Links

[Home](#)
[Products](#)
[Shopping Cart](#)


The order page contains the information including order time, order ID, total and the products details for what you have bought.

Enjoy your shopping trip at InstaShop! Add more items to your shopping cart!

InstaShop

[Home](#) [Products](#) [Shopping Cart](#) [Orders](#)


Current User: ecommerceGuest [Sign Out](#)



Orders

Order Date/Time	Order Number	Order Total	Status	Details
2021-12-10	20	\$210.0	Preparing Shipment	Products (amount) Echo: 2 Watch: 1
2021-12-10	21	\$100.0	Preparing Shipment	Products (amount) Kindle: 1
2021-12-10	22	\$1247.0	Preparing Shipment	Products (amount) Artificial Tree: 3 Watch: 1 Kindle: 2

Showing 1 to 3 of 3 entries



About Us

InstaShop is a SaaS e-commerce website.

We aim to provide the best e-commerce experience.

Contact Us

4 N 2nd St.

San Jose, CA 95113

Links

[Home](#)

[Products](#)

Test Plan & Expected results

Test case 1: test the Sign up feature.

expected result: Users can sign up by using their username and password

Test case 2: test the sign in feature

expected result: Users can sign in by using the credentials that they signed up

Test case 3: click 'Home', 'Products', 'Shopping Cart', 'Orders' buttons

expected result: These web pages load correctly.

Test case 4: At home page, click each product of top picks

Expected result: Each product information is displayed on the new page and web page is loaded correctly

Test case 5: Click 'Products', add each product to the cart

Expected result: all these products can be added to the cart.

Test case 6: Click 'Shopping Cart'

expected result: shopping cart should have the products that customers chose in the 'Products' page. Shopping cart should have the product name, product quantity and the total price of each product. In addition, customers can click the '+' and '-' to increase and decrease the quantity of product.

Test case 7: Click the 'proceed to pay' button on the checkout section. Fill out the shipping details and payment information

Expected result: the checkout feature should work properly. Customers click 'checkout' button, then this order should be placed successfully

Take away

1. We learned to use Django to develop a web application
2. We learned to use Zappa commands to deploy the web application
3. We learned to use Elastic Beanstalk to deploy the web application
4. We learned to use containerization, amazon rds and build CI/CD pipeline to deploy the web application

Additional effort

Besides the above-mentioned success deployment, we also tried other approaches to deploy the web app. For instance, we tried to use zappa and Elastic Beanstalk to deploy the web app. We really learned a lot via the overall deployment process. Thus, we also want to mention it here as additional effort.

Use Elastic Beanstalk to deploy the web app:

Use 'eb create' command to create the elastic Beanstalk environment, ec2 instance and the S3 bucket.

```
(venv) wenhaoge@erhaos-MBP cloud_computing_ecommerce % eb create django-dev-final-project
Creating application version archive 'app-8840-211207-214831'.
Uploading django-final-project/app-8840-211207-214831.zip to S3. This may take a while.
Upload Complete.
Application django-final-project has been created.
Environment details for: django-dev-final-project
Application name: django-final-project
Region: us-east-2
Deployed Version: app-8840-211207-214831
Environment ID: e-5y6bsr9c4m
Platform: arn:aws:elasticbeanstalk:us-east-2::platform/Python 3.7 running on 64bit Amazon Linux 2/3.3.8
Tier: WebServer-Standard-1.0
CNAME: UNKNOWN
Updated: 2021-12-08 05:48:36.460000+00:00
Printing Status:
2021-12-08 05:48:35 INFO createEnvironment is starting.
2021-12-08 05:48:36 INFO Using elasticbeanstalk-us-east-2-808150127179 as Amazon S3 storage bucket for environment data.
2021-12-08 05:48:36 INFO Created security group named: sg-0b0a5e6aed2a7c3b7
2021-12-08 05:48:55 INFO Created load balancer named: awseb-e-5y6bsr9c4m-stack-AWSEBLoadBalancer
2021-12-08 05:49:11 INFO Created security group named: awseb-e-5y6bsr9c4m-stack-AWSEBSecurityGroup-VPT8CEJKYNGF
2021-12-08 05:49:11 INFO Created Auto Scaling launch configuration named: awseb-e-5y6bsr9c4m-stack-AWSEBAutoScalingLaunchConfiguration-11VATQSHWHL8G
2021-12-08 05:50:29 INFO Created Auto Scaling group named: awseb-e-5y6bsr9c4m-stack-AWSEBAutoScalingGroup-11VCQW6GCO1W0
2021-12-08 05:50:29 INFO Waiting for EC2 instances to launch. This may take a few minutes.
2021-12-08 05:50:29 INFO Created Auto Scaling policy named: arn:aws:autoscaling:us-east-2:808150127179:scalingPolicy:08493faf-f93a-4561-bf5a-2c0663d444c1:autoScalingGroup-11VCQW6GCO1W0:policyName/awseb-e-5y6bsr9c4m-stack-AWSEBAutoScalingScaleDownPolicy-1T2MEDF41VM51
2021-12-08 05:50:29 INFO Created Auto Scaling policy named: arn:aws:autoscaling:us-east-2:808150127179:scalingPolicy:cd2f8eb6-002c-41ce-b597-dd21dc15c005:autoScalingGroup-11VCQW6GCO1W0:policyName/awseb-e-5y6bsr9c4m-stack-AWSEBAutoScalingScaleUpPolicy-1U3NND3C6AADI
2021-12-08 05:50:44 INFO Created CloudWatch alarm named: awseb-e-5y6bsr9c4m-stack-AWSEBCloudWatchAlarmHigh-YPL9W1NRQ4SA
2021-12-08 05:50:44 INFO Created CloudWatch alarm named: awseb-e-5y6bsr9c4m-stack-AWSEBCloudWatchAlarmLow-1P1QITM935SN
2021-12-08 05:50:48 INFO Instance deployment successfully generated a 'Profile'.
2021-12-08 05:50:50 INFO Instance deployment completed successfully.
2021-12-08 05:51:53 INFO Successfully launched environment: django-dev-final-project
The above exception was the direct cause of the following exception:
(venv) wenhaoge@erhaos-MBP cloud_computing_ecommerce % eb status
Environment details for: django-dev-final-project
Application name: django-final-project
Region: us-east-2
Deployed Version: app-8840-211207-214831
Environment ID: e-5y6bsr9c4m
Platform: arn:aws:elasticbeanstalk:us-east-2::platform/Python 3.7 running on 64bit Amazon Linux 2/3.3.8
Tier: WebServer-Standard-1.0
CNAME: django-dev-final-project.elb.pnxnnyu.us-east-2.elasticbeanstalk.com
```

Use 'eb deploy' command to deploy the new changes to the environment. Then use the 'eb open' command to open the URL in the browser. But the URL did not open.

```
(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb deploy
Creating application version archive "app-d30e-211207_221202".
Uploading django-final-project/app-d30e-211207_221202.zip to S3. This may take a while.
Upload Complete.
2021-12-08 06:12:06 INFO Environment update is starting.
2021-12-08 06:12:10 INFO Deploying new version to instance(s).
2021-12-08 06:12:13 INFO Instance deployment successfully generated a 'Procfile'.
2021-12-08 06:12:21 INFO Instance deployment completed successfully.
2021-12-08 06:12:27 INFO New application version was deployed to running EC2 instances.
2021-12-08 06:12:27 INFO Environment update completed successfully.
NameError: name 'mysql' is not defined

(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb open
(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb deploy
Creating application version archive "app-d30e-211207_222651".
Uploading django-final-project/app-d30e-211207_222651.zip to S3. This may take a while.
Upload Complete.
2021-12-08 06:26:55 INFO Environment update is starting.
2021-12-08 06:26:58 INFO Deploying new version to instance(s).
2021-12-08 06:27:01 INFO Instance deployment successfully generated a 'Procfile'.
2021-12-08 06:27:09 INFO Instance deployment completed successfully.
2021-12-08 06:27:15 INFO New application version was deployed to running EC2 instances.
2021-12-08 06:27:15 INFO Environment update completed successfully.
File "/Applications/everything/NEU/Fall 2021/CloudComputing/django

(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb open
(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb deploy
Creating application version archive "app-d30e-211207_222956".
Uploading django-final-project/app-d30e-211207_222956.zip to S3. This may take a while.
Upload Complete.
2021-12-08 06:29:59 INFO Environment update is starting.
2021-12-08 06:30:03 INFO Deploying new version to instance(s).
2021-12-08 06:30:06 INFO Instance deployment successfully generated a 'Procfile'.
2021-12-08 06:30:15 INFO Instance deployment completed successfully.
2021-12-08 06:30:21 INFO New application version was deployed to running EC2 instances.
2021-12-08 06:30:21 INFO Environment update completed successfully.
File "manage.py", line 18, in main
MySQLdb.exceptions.OperationalError: (1049, "Unknown database 'dat

(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb open
(venv) wenhaoge@Wenhaos-MBP cloud_computing_ecommerce % eb deploy
Creating application version archive "app-d30e-211207_225343".
Uploading django-final-project/app-d30e-211207_225343.zip to S3. This may take a while.
```

In the logs, we got the errors in the next picture. Then we added a Procfile in the project directory, and added this line into the file:

```
web: gunicorn --bind :8000 --workers 3 --threads 2
```

In addition, we added this line in django.config file:

```
WSGIPath: hw3_ecommerce.wsgi.application
```

Then we ran 'eb deploy' again, but the problem persisted. When we ran 'eb open', there was no URL open in the browser.

```
Wenhai Ge | x | singinraincloud... | x | github - Pushing to | x | aws login - Google | x | djang... | x | https://elasticbeanstalk-us-east-2-808150102779.us-east-2.amazonaws.com/resources/environments/logs/tail?e-Sy6bs9c4mJ-05462b8e88d950949/tailLogs-1639157860310.txtTX-Amz-Algorithm=... | x | Getting started with... | x | RDS Management C... | x | amazon rds - Google | x | + | Update |
Apps | 23种设计模式全解 | Schedule | CMU.1 | coderQandNew... | LeetCode 分类题解... | CORNELL CS478... | 数据增长DataSino | 来自fb hr的简历资源... | 分类: 系统设计 | ja... | 墨西哥哥手稿补全... | Interview Preparation... | Reading List
2021/12/18 17:37:48.314866 [INFO] Running command /bin/bash -c 'tail -n 100 /var/log/eb-engine.log'

/var/log/web_stdout.log
-----
Dec 10 07:00:58 ip-172-31-2-48 web1 File <frozen importlib__bootstrap>, line 1006, in _get_import
Dec 10 07:00:58 ip-172-31-2-48 web1 File <frozen importlib__bootstrap>, line 983, in find_and_load
Dec 10 07:00:58 ip-172-31-2-48 web1 File <frozen importlib__bootstrap>, line 963, in find_and_load_unlocked
Dec 10 07:00:58 ip-172-31-2-48 web1 ModuleNotFoundError: No module named 'application'
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3542] [INFO] Worker exiting (pid: 3542)
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3535] [INFO] Shutting down: Master
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3535] [INFO] Reason: Worker failed to boot.
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3550] [INFO] Starting unicorn 26.1.0
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3550] [INFO] Listening at: http://127.0.0.1:8000 (3550)
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3550] [INFO] Using worker: gthread
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3557] [INFO] Booting worker with pid: 3557
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3557] [ERROR] Exception in worker process
Dec 10 07:00:58 ip-172-31-2-48 web1 Traceback (most recent call last):
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/arbiter.py", line 589, in spawn_worker
Dec 10 07:00:58 ip-172-31-2-48 web1 worker.init_process()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/workers/gthread.py", line 92, in init_process
Dec 10 07:00:58 ip-172-31-2-48 web1 super().init_process()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/workers/base.py", line 134, in init_process
Dec 10 07:00:58 ip-172-31-2-48 web1 self.load_wsgi()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/workers/base.py", line 146, in load_wsgi
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/app/base.py", line 67, in wsgi
Dec 10 07:00:58 ip-172-31-2-48 web1 self.callable = self.load()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 56, in load
Dec 10 07:00:58 ip-172-31-2-48 web1 return self.load_wsgiapp()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 46, in load_wsgiapp
Dec 10 07:00:58 ip-172-31-2-48 web1 return util.import_app(self.app_uri)
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/util.py", line 359, in import_app
Dec 10 07:00:58 ip-172-31-2-48 web1 mod = importlib.import_module(module)
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/usr/lib64/python3.7/importlib__init__.py", line 127, in import_module
Dec 10 07:00:58 ip-172-31-2-48 web1 return _bootstrap._gcd_import(name[level], package, level)
Dec 10 07:00:58 ip-172-31-2-48 web1 File <frozen importlib__bootstrap>, line 1006, in _get_import
Dec 10 07:00:58 ip-172-31-2-48 web1 File <frozen importlib__bootstrap>, line 983, in find_and_load
Dec 10 07:00:58 ip-172-31-2-48 web1 File <frozen importlib__bootstrap>, line 963, in find_and_load_unlocked
Dec 10 07:00:58 ip-172-31-2-48 web1 ModuleNotFoundError: No module named 'application'
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3557] [INFO] Worker exiting (pid: 3557)
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3550] [INFO] Shutting down: Master
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3550] [INFO] Reason: Worker failed to boot.
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3565] [INFO] Starting unicorn 26.1.0
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3565] [INFO] Listening at: http://127.0.0.1:8000 (3565)
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3565] [INFO] Using worker: gthread
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3572] [INFO] Booting worker with pid: 3572
Dec 10 07:00:58 ip-172-31-2-48 web1 [2021-12-10 07:00:58 +0000] [3572] [ERROR] Exception in worker process
Dec 10 07:00:58 ip-172-31-2-48 web1 Traceback (most recent call last):
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/arbiter.py", line 589, in spawn_worker
Dec 10 07:00:58 ip-172-31-2-48 web1 worker.init_process()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/workers/gthread.py", line 92, in init_process
Dec 10 07:00:58 ip-172-31-2-48 web1 super().init_process()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/workers/base.py", line 134, in init_process
Dec 10 07:00:58 ip-172-31-2-48 web1 self.load_wsgi()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/app/base.py", line 67, in wsgi
Dec 10 07:00:58 ip-172-31-2-48 web1 self.callable = self.load()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 56, in load
Dec 10 07:00:58 ip-172-31-2-48 web1 return self.load_wsgiapp()
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/app/wsgiapp.py", line 46, in load_wsgiapp
Dec 10 07:00:58 ip-172-31-2-48 web1 return util.import_app(self.app_uri)
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/var/app/vener/staging-LQMIest/lib/python3.7/site-packages/gunicorn/util.py", line 359, in import_app
Dec 10 07:00:58 ip-172-31-2-48 web1 mod = importlib.import_module(module)
Dec 10 07:00:58 ip-172-31-2-48 web1 File "/usr/lib64/python3.7/importlib__init__.py", line 127, in import_module
```

Use Zappa to deploy the web app:

We ran ‘zappa init’, ‘zappa update dev’, ‘zappa deploy dev’ to deploy the web app. The deployment is live, and we got a URL. we opened the URL in the browser, then we got this error in the browser.

```
Wenhai Ge | x | aws zappa | x | Operational: | x | zappa-q88 | x | Operational: | x | zappa deplo | x | atadid Do | x | operational: | x | Using Sign | x | How to Buil | x | Announcem | x | CS 6920 | x | + |
Apps | 23种设计模式全解 | Schedule | CMU.1 | coderQandNew... | LeetCode 分类题解... | CORNELL CS478... | 数据增长DataSino | 来自fb hr的简历资源... | 分类: 系统设计 | ja... | 墨西哥哥手稿补全... | Interview Preparation... | Reading List
176cnv96x2 execute-api-us-east-2.amazonaws.com/dev/e-commerce/

OperationalError at /ecommerce/
no such table: ecommerce_user

Request Method: GET
Request URL: https://176cnv96x2.execute-api-us-east-2.amazonaws.com/dev/e-commerce/
Django Version: 2.2.13
Exception Type: OperationalError
Exception Value: no such table: ecommerce_user
Exception Location: /var/task/django/db/backends/sqlite3/base.py in execute, line 383
Python Executable: /usr/bin/python3.7
Python Version: 3.7.12
Python Path: ['/var/task',
               '/opt/python/lib/python3.7/site-packages',
               '/opt/python',
               '/var/task',
               '/var/lang/lib/python3.7.zip',
               '/var/lang/lib/python3.7/lib-dynload',
               '/var/lang/lib/python3.7/site-packages',
               '/opt/python/lib/python3.7/site-packages',
               '/opt/python',
               '/var/task']
Server time: Tue, 7 Dec 2021 00:50:53 +0000

Traceback (switch to copy-and-paste view)
/var/task/django/db/backends/sqlite3/base.py in _execute
384:         return self.cursor.execute(sql, params)
└─ Local vars

/var/task/django/db/backends/sqlite3/base.py in _execute
383:         return Database.Cursor.execute(self, query, params)
└─ Local vars

The above exception (no such table: ecommerce_user) was the direct cause of the following exception:
/var/task/django/core/handlers/exception.py in inner
34:         response = get_response(request)
└─ Local vars

/var/task/django/core/handlers/base.py in _get_response
115:         response = self.process_exception_by_middleware(e, request)
└─ Local vars

/var/task/django/core/handlers/base.py in _get_response
113:         response = wrapped_callback(request, **callback_kwargs)
└─ Local vars

/var/task/ecommerce/views.py in index
11:         default_user = User.objects.first()
└─ Local vars
```

We think the reason is the database was not deployed successfully. Then we took steps to deploy the database.

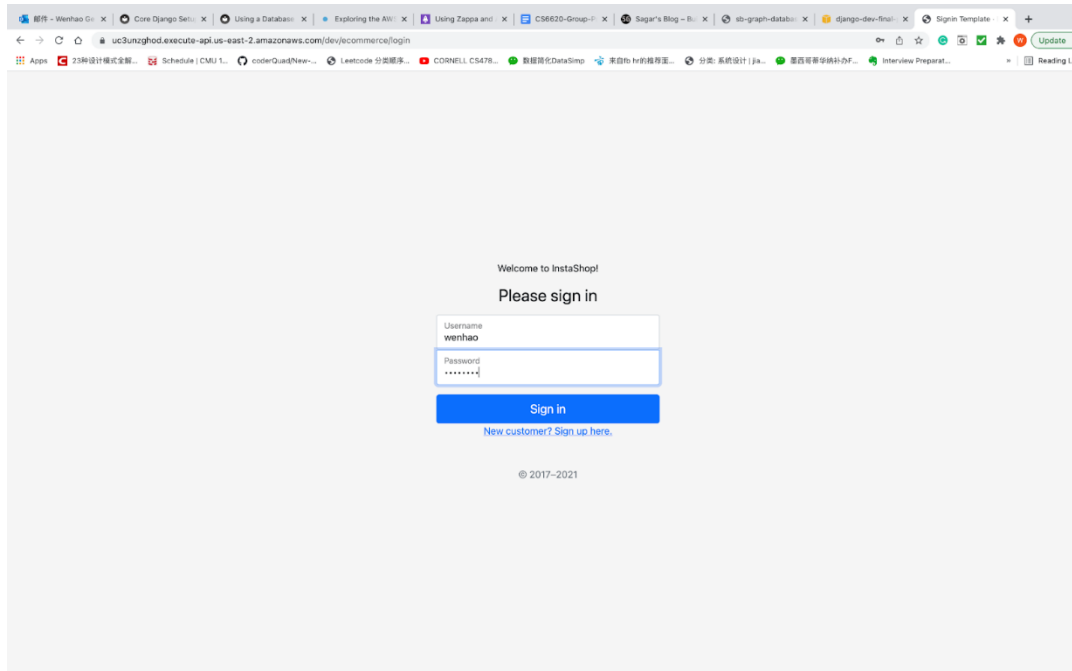
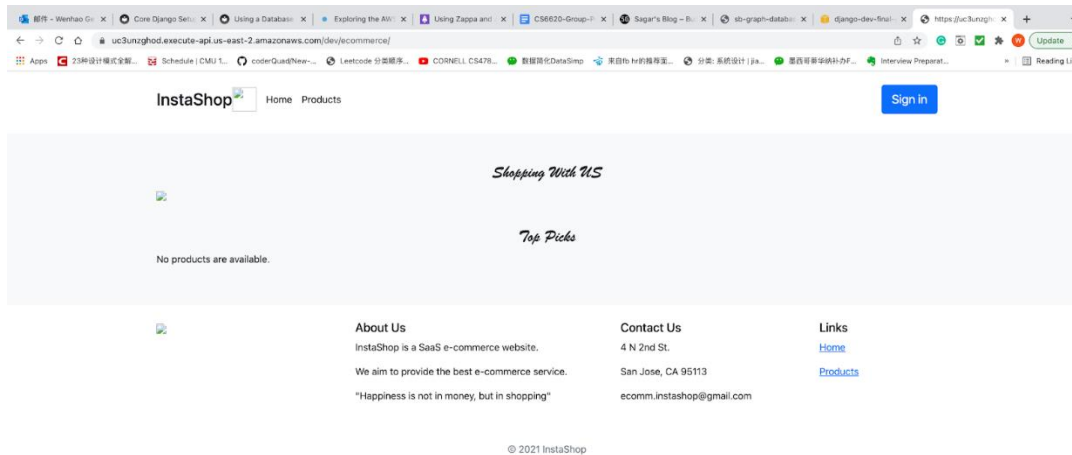
We ran 'pip install django_s3_sqlite' and 'pip install django_s3_storage'. We created a MySQL database using AWS RDS. Then we ran 'python manage.py makemigrations', we got an error: NameError: name '_mysql' is not defined. Our solution is to run 'pip install pymysql' and 'brew install mysql', then this problem was resolved. Later, we met with this kind of error:

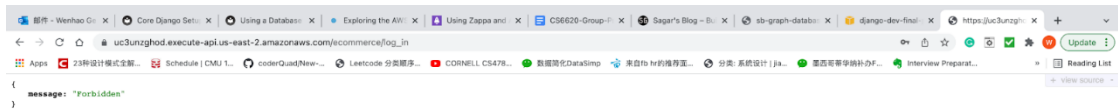
django.db.utils.OperationalError: (1049, "Unknown database"). we fixed this issue too. In addition, when we ran 'python manage.py makemigrations' command, we was met with access denied related issues. we rewrote the correct database name, database username and database password in settings.py file. Then we solved these accesses denied issues.

The deployed URL is

<https://uc3unzghod.execute-api.us-east-2.amazonaws.com/dev><https://uc3unzghod.execute-api.us-east-2.amazonaws.com/dev>

The following images shows that something was not loaded correctly. we clicked the 'Sign in' button, we typed in the username and password, but we were not able to sign in. we saw 'Forbidden' on the screen.





Executive Summary

As an overview of this project, we would thank our teammates, Professor Prasad and our teaching assistants first. Without their help, no individual could have achieved so much in just one semester. Through this project, our team have a clear division of labor of each task. We split the whole task into some group tasks and some light individual work. In summary, the process of the Ecommerce SaaS web application includes designing, implementing UI section, connecting to the database and deployment. This experience would be helpful for our understanding of Cloud Computing and build long-term memory.

For the learning outcome – Concept, CI/CD is the combined practices of continuous integration (CI) and either continuous delivery or continuous deployment (CD). CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications. The process contrasts with traditional methods where all updates were integrated into one large batch before rolling out the newer version. Modern day DevOps practices involve continuous development, continuous testing, continuous integration, continuous deployment and continuous monitoring of software applications throughout its development life cycle. The CI/CD practice, or CI/CD pipeline, forms the backbone of modern-day DevOps operations. And for the view part, we have a better understanding of how Django works (with SQLite other storage service, for example RDS) and how it is deployed.

Through the process of deployment and some programming, we understand how the ECS, S3, Lambda, git, code pipeline, Load balancer work better. Also, lots of effort could be favorable to SQL operation and the relationships between various of services of AWS. Through the comprehensive application of them, we believe, in the future, we can cooperate with team members, friends or colleagues to complete larger projects in a more efficient way using what we learnt in this course. We're very grateful for the experience!