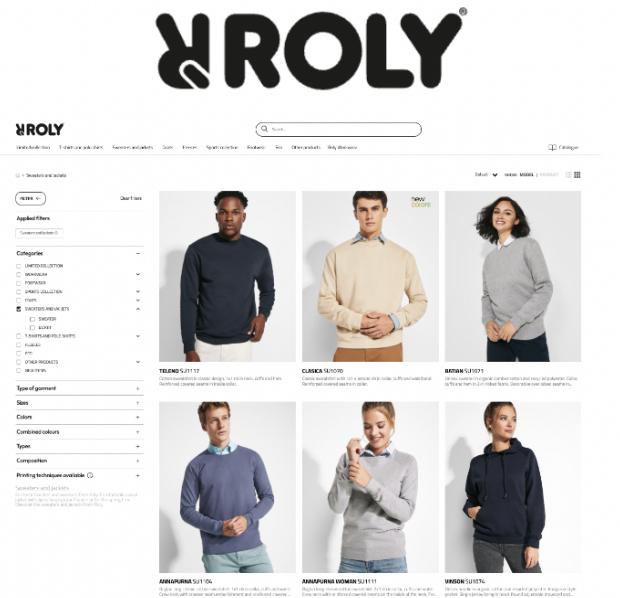


Project Implementation

Ecommerce Cloud Service Implementation - Case Roly



Content

Background Information.....	3
Business Case and Problem.....	4
Recommend Solution to the problem.....	5
Procurement details.....	6
Cloud System Design.....	6
Cloud Service and Implementation.....	7
Website Demonstration.....	18
Security Plan.....	22
Cost Analysis.....	23
Success Criteria and Evaluation.....	24
Conclusion.....	26
Appendix 1 (Lambda).....	
Appendix 2 (SageMaker).....	

Background Information

Roly is an apparel manufacturing company in Europe. T-shirts, polo shirts, sweatshirts, jackets, jeans, and other apparel items are among the many products they are renowned for making. Roly offers shipping services to customers in the USA and Europe and sells its items primarily online. The brand is well-known in the European market, and it aggressively broadens the reach of its online services while also enhancing the infrastructure of its operations.

We decided to analyze the online retail market due to its underlying potential. In recent years, the online market has grown significantly. This trend has been fueled by a number of causes, including the expansion of e-commerce sites like Amazon, Alibaba, and eBay, the expansion of high-speed internet access, and the widespread usage of mobile devices.

Global e-commerce sales will increase by 27.6% to \$4.28 trillion in 2020, according to a forecast by Digital Commerce 360. Additionally, the COVID-19 epidemic has hastened the expansion of internet retail as more customers have opted to shop online rather than in-store.

The online market has experienced significant growth in the past few years, with the number of online shoppers in the USA reaching 268 million in 2022. Projections indicate that this number will continue to grow at an annual rate of 14.6% from 2021 to 2028, with an estimated value of \$58.74 trillion by 2028. These trends suggest that the online retail market will remain a significant force in the global economy in the coming years.

A Marist poll conducted in the USA found that 76% of U.S. adults were online shoppers. Within this group, 62% were considered regular online shoppers. This high level of online shopping activity suggests that consumers are increasingly turning to online platforms for their purchasing needs. The convenience, wider reach, and better pricing options offered by online shopping are likely driving this trend. As the online market continues to grow, businesses that have a strong online presence and effective e-commerce strategies are poised to benefit from the increasing demand for online shopping.

Business Case and Problem

As an E-commerce business, Roly's choice of a web hosting provider is critical to the success of online business. After careful evaluation of our current web host, SiteGround, we have identified several problems that are impacting our business's efficiency, flexibility, and resilience. To address these issues, we propose a move to Amazon Web Services (AWS).

The first problem we have identified with SiteGround is cost inefficiency. Our current web host does not offer tailored plans that meet our specific business needs, resulting in unnecessary costs. We believe that AWS offers a more cost-effective solution that can better match our business requirements. By moving to AWS, we can take advantage of their pay-as-you-go pricing model, which allows us to only pay for the services we need and use.

The second problem is low flexibility. Our current online infrastructure requires third-party applications for analysis and predictions, and data flow is manual. This approach is both time-consuming and prone to errors, which is affecting our ability to make timely and accurate business decisions. AWS offers a more flexible solution that can integrate with our existing applications and automate data flow. With AWS, we can access a wide range of tools and services that can enhance our analytics and enable us to make data-driven decisions that support our business growth.

The third problem is low resilience and fault-tolerance. SiteGround does not provide an option to keep the website on when computational power is exceeded, causing a website crash. This is particularly concerning for an E-commerce business where website uptime is critical to our revenue generation. AWS provides a more resilient and fault-tolerant solution through their auto-scaling feature that can adjust our website's computational resources to match traffic demands. This ensures that our website stays up and running, even during peak traffic periods.

In conclusion, we believe that moving to AWS is a strategic decision that can address the problems we have identified with our current web hosting provider. AWS offers a cost-effective, flexible, and resilient solution that can support our business growth and enable us to meet the demands of our customers. We recommend that we make the move to AWS as soon as possible to avoid further losses caused by our current hosting provider's shortcomings.

Recommended Solutions to the Problem

We have developed some potential solutions to the business problem of Roly using AWS service.

1. Cost effective solutions: AWS service is a pay-as-you-go service that can let us manage our expenses by managing the service amount we use. This is more appropriate for Roly's needs since Roly is an emerging company that is currently expanding their business and is uncertain about the usage of the service.
2. Consolidated Software: Roly is currently using SiteGround which does not provide a wide scope of analytical services for their clients. AWS offers an array of integrated analytical softwares for E-commerce metrics. This is a solution to the problem that SiteGround is lacking the variety of software and service options.
3. Resistance: AWS is a comparatively better solution to provide high availability and fault tolerance. Since the current technology Siteground is more susceptible to site outages, we consider using AWS is a better option.
4. Configurability: AWS is more flexible and can customize the cloud access control while SiteGround only offers smaller control.
5. Scalability: Siteground does not have flexible scaling plans while if we use AWS service, we can obtain a wide range of services that can help scale applications.

Procurement Details

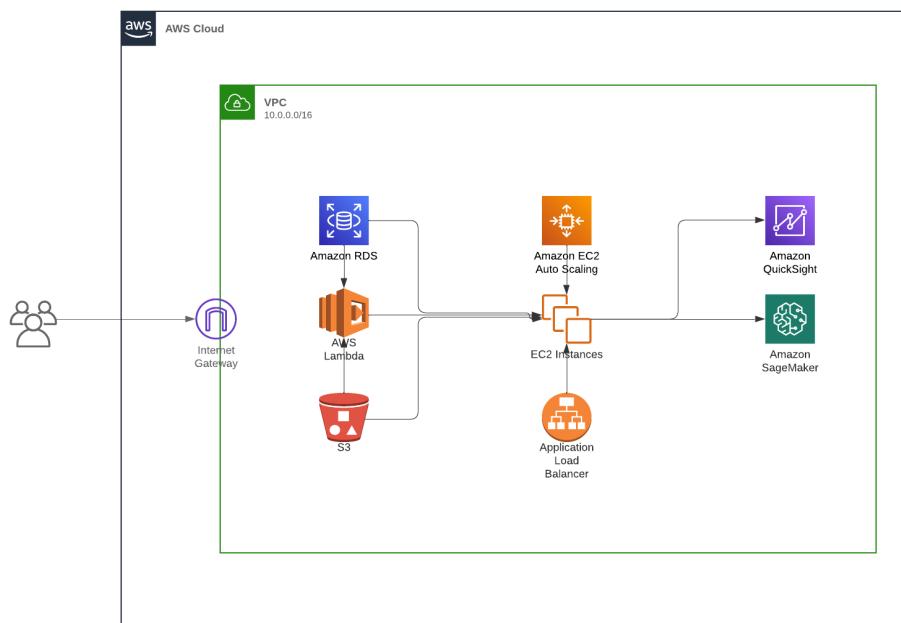
In this section we will elaborate the procurement details of our project.

1. Roly Database: Importing 2 GB of clothing products and data with API Call on Roly DB
2. API: API calling on inventory quantity for real-time updates to merge with products table.
3. Data Wrangling: Processing schema and data adaptation to WooCommerce Database
4. WooCommerce Database and API: Using Python Library to post and put products on WooCommerce Database

We explained more details of the procurement process in the appendix 1.

Cloud System Design

We designed our cloud system with main service, connecting to each other: EC2, RDS, Lambda, S3 and VPC. Besides the fundamental design, we also added other AWS services to enable our system to be more efficient, useful and flexible. For example, EC2 Auto Scaling, Load Balancer, QuickSight and SageMaker. Below is a chart showing our proposed cloud system design.



Cloud Service and Implementation

In this section, we will look into details of our selected cloud services and elaborate the implementation results.

EC2

For EC2, we launched a Linux-based Amazon EC2 instance, choosing T3.Large and the default settings of the EC2. To run a web application, web servers need to run all the time, and we reserve instances to save on running costs. We kept the default of the availability zone of us-east-1b. For security, we created a roly.pem key and created a web security group. We will go through the details further about the security groups in the VPC part later.

Settings of our EC2 (woocommerce_t3large)

Instance: i-0c7abe926dec00860 (woocommerce_t3large)		
▼ Instance summary Info		
Instance ID i-0c7abe926dec00860 (woocommerce_t3large)	Public IPv4 address 54.83.94.200 open address	Private IPv4 addresses 10.0.2.117
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-83-94-200.compute-1.amazonaws.com open address
Hostname type IP name: ip-10-0-2-117.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-2-117.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t3.large	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 54.83.94.200 [Public IP]	VPC ID vpc-0ab88c40da193fd3c (project-vpc)	Subnet ID subnet-07eb57980543901ed (project-subnet-public2)
IAM Role -		Auto Scaling Group name -

VPC

We created a VPC named project-VPC and set up the subnets and security groups accordingly. For the security group, we created a new Web-security group and planned to use this group to protect most of our cloud system. For the inbound and outbound rules, we add new rules of SSH, HTTP and HTTPS.

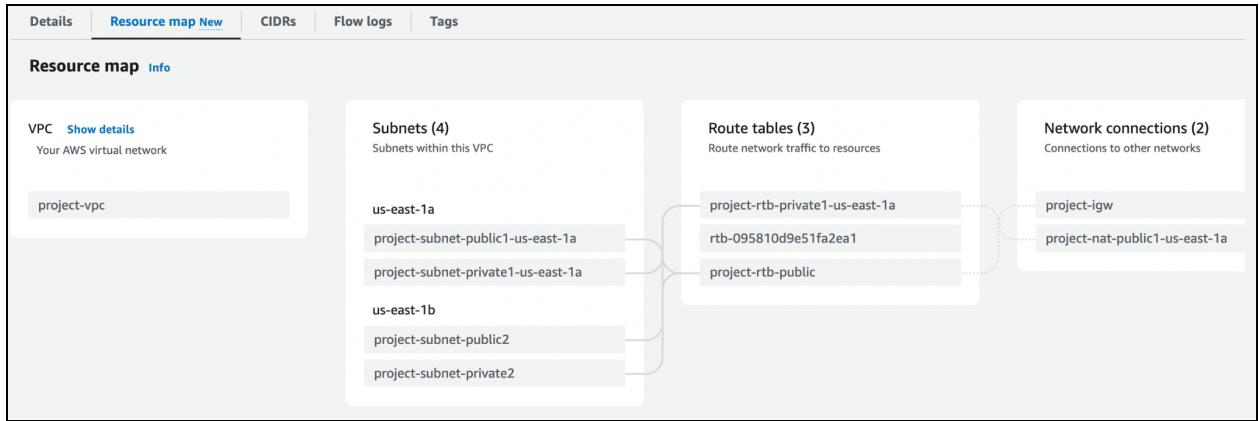
Settings of our VPC (project-vpc)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
<input checked="" type="checkbox"/> project-vpc	vpc-0ab88c40da193fd3c	Available	10.0.0.0/16	-	dopt-0a372f5de4c7c
<input type="checkbox"/> -	vpc-0b89b648b0b6c6cd0	Available	172.31.0.0/16	-	dopt-0a372f5de4c7c

vpc-0ab88c40da193fd3c / project-vpc

Details	Resource map New	CIDRs	Flow logs	Tags
Details				
VPC ID <input type="button" value="vpc-0ab88c40da193fd3c"/>	State Available	DNS hostnames Enabled	DNS resolution Enabled	
Tenancy Default	DHCP option set <input type="button" value="dopt-0a372f5de4c7cce7"/>	Main route table <input type="button" value="rtb-095810d9e51fa2ea1"/>	Main network ACL <input type="button" value="acl-0d8f6b2b4ea6d596e"/>	
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -	
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID <input type="button" value="008993324755"/>		

Resource map of our VPC(project-vpc)



Settings of our security groups (Web Security Group)

Details			
Security group name <input type="button" value="Web Security Group"/>	Security group ID <input type="button" value="sg-001a7d7acd230a12a"/>	Description <input type="button" value="Enabl HTTP Access"/>	VPC ID <input type="button" value="vpc-0ab88c40da193fd3c"/>
Owner <input type="button" value="008993324755"/>	Inbound rules count 3 Permission entries	Outbound rules count 4 Permission entries	

Inbound rules of our security groups(Web Security Group) - HTTPS, HTTP and SSH

Inbound rules (3)								
	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	
<input type="checkbox"/>	-	sgr-0b0cdee28d75817...	IPv4	HTTPS	TCP	443	0.0.0.0/0	
<input type="checkbox"/>	-	sgr-03579e251ab671...	IPv4	SSH	TCP	22	0.0.0.0/0	
<input type="checkbox"/>	-	sgr-0b93c477f84b9347f	IPv4	HTTP	TCP	80	0.0.0.0/0	

Outbound rules of our security groups(Web Security Group) - HTTPS, HTTP and SSH

Outbound rules (4)								
	Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	
<input type="checkbox"/>	-	sgr-05bf7b61f71b82f81	IPv4	SSH	TCP	22	0.0.0.0/0	
<input type="checkbox"/>	-	sgr-0849e392a1ba023...	IPv4	HTTP	TCP	80	0.0.0.0/0	
<input type="checkbox"/>	-	sgr-0d01433b949aa4cff	IPv4	HTTPS	TCP	443	0.0.0.0/0	
<input type="checkbox"/>	-	sgr-0313c2bead60b82...	IPv4	All traffic	All	All	0.0.0.0/0	

RDS

RDS Provides cost-efficient and resizable capacity, while managing time-consuming database administration tasks. We made a decision between RDS and Aurora. Comparing to Amazon Aurora, Amazon RDS is, compatible to wider range of tools, allowing access to wider range of database engines, including MySQL, PostgreSQL

We created an RDS SQL Database named roly and kept most of the settings defaulted. Also, we used nano editor and created a roly.py file on the Mac terminal to see if the roly database is working.

Settings of our RDS (SQL database - roly)

EndPoint : roly.cspvnuqpnova2.us-east-1.rds.amazonaws.com

DB identifier	CPU	Status	Class
roly	2.12%	Available	db.t3.micro

Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-1a

Connectivity & security	Monitoring	Logs & events	Configuration	Maintenance & backups	Tags									
Connectivity & security <table border="1"> <thead> <tr> <th>Endpoint & port</th> <th>Networking</th> <th>Security</th> </tr> </thead> <tbody> <tr> <td>Endpoint roly.cspvnuqpnova2.us-east-1.rds.amazonaws.com</td> <td>Availability Zone us-east-1a</td> <td>VPC security groups ec2-rds-1 (sg-01a6bd51c157b1207) Active</td> </tr> <tr> <td>Port 3306</td> <td>VPC vpc-0b89b648b0b6c6cd0 Subnet group default-vpc-0b89b648b0b6c6cd0</td> <td>default (sg-0e0b0be4f8b88a41b) Active rds-ec2-1 (sg-03b08d5db5be0edc8) Active launch-wizard-1 (sg-0c9e5539449ce1ca7) Active</td> </tr> </tbody> </table>						Endpoint & port	Networking	Security	Endpoint roly.cspvnuqpnova2.us-east-1.rds.amazonaws.com	Availability Zone us-east-1a	VPC security groups ec2-rds-1 (sg-01a6bd51c157b1207) Active	Port 3306	VPC vpc-0b89b648b0b6c6cd0 Subnet group default-vpc-0b89b648b0b6c6cd0	default (sg-0e0b0be4f8b88a41b) Active rds-ec2-1 (sg-03b08d5db5be0edc8) Active launch-wizard-1 (sg-0c9e5539449ce1ca7) Active
Endpoint & port	Networking	Security												
Endpoint roly.cspvnuqpnova2.us-east-1.rds.amazonaws.com	Availability Zone us-east-1a	VPC security groups ec2-rds-1 (sg-01a6bd51c157b1207) Active												
Port 3306	VPC vpc-0b89b648b0b6c6cd0 Subnet group default-vpc-0b89b648b0b6c6cd0	default (sg-0e0b0be4f8b88a41b) Active rds-ec2-1 (sg-03b08d5db5be0edc8) Active launch-wizard-1 (sg-0c9e5539449ce1ca7) Active												

Checking the availability of our database by creating a python file using nano editor

```
cloud_computing_final — ec2-user@ip-172-31-82-59:~ — ssh -i roly.pem ec2-user@ec2-3-84
GNU nano 2.9.8                               roly.py

import mysql.connector

mydb = mysql.connector.connect(host="roly.cspvnuqpnova2.us-east-1.rds.amazonaws.com", user= "ketcp")
mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM roly")

# Fetch all the rows as a list of tuples
result = mycursor.fetchall()

for row in result:
    print(row)

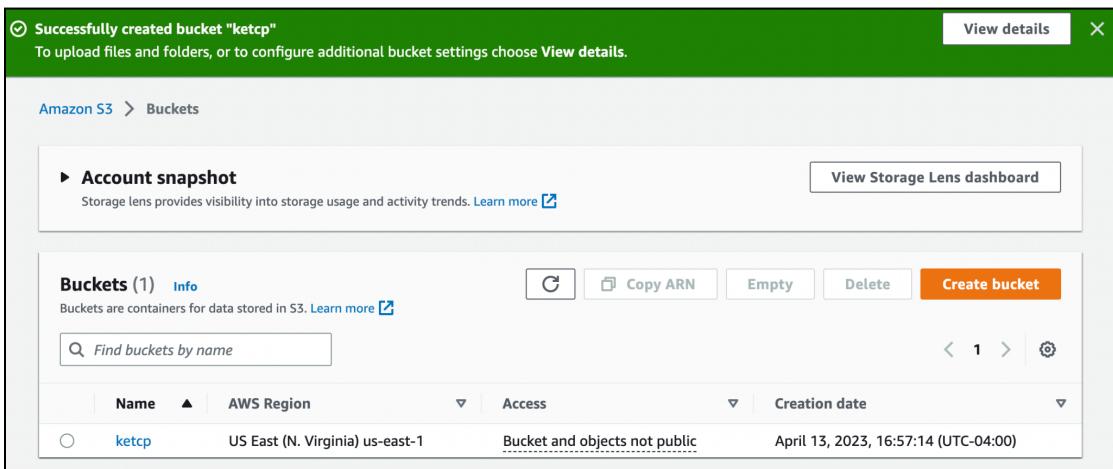
mydb.commit()
mydb.close()
```

S3

For storing large files, compared to EFS, S3 is a better option due to its scalability and low cost per GB of storage. We set up an Amazon S3 bucket to store our product images and other media files. We later configured the WooCommerce store to use Amazon S3 for storing and retrieving media files.

We set up a bucket named ketchup with default settings. We then used the mac terminal to ssh EC2 and S3 and upload the files to S3 bucket ketcp.

Settings of our S3 buckets (ketcp)



Syncing photos to S3 from wordpress database using codes below

```
aws s3 sync /var/www/html/wp-content/uploads s3://ketcp/wp-content/uploads
```

Objects successfully synced and uploaded to S3 bucket

Objects (10)						
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more						
<input type="checkbox"/>	<input type="checkbox"/> Copy S3 URI	<input type="checkbox"/> Copy URL	<input type="checkbox"/> Download	<input type="checkbox"/> Open	<input type="checkbox"/> Delete	<input type="checkbox"/> Actions ▾
<input type="checkbox"/>	<input type="checkbox"/> Upload					
<input type="text"/> Find objects by prefix						
<input type="checkbox"/>	Name	Type	Last modified		Size	Storage class
<input type="checkbox"/>	<input type="checkbox"/> wc-logs/	Folder	-		-	-
<input type="checkbox"/>	<input type="checkbox"/> woocommerce_uploads/	Folder	-		-	-
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-100x100.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		2.3 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-1024x1024.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		90.0 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-150x150.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		4.1 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-300x300.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		12.3 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-324x324.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		27.5 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-416x416.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		45.0 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder-768x768.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		57.3 KB	Standard
<input type="checkbox"/>	<input type="checkbox"/> woocommerce-placeholder.png	png	April 19, 2023, 14:49:24 (UTC-04:00)		100.2 KB	Standard

Amazon S3 > Buckets > ketcp > wp-content/ > uploads/ > 2023/ > 04/

Objects Properties

To enable sorting in the table below, use the search to reduce the size of the list to 999 objects or fewer.

Objects (999+)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 Inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
0201_193_2_1-100x100.jpeg	jpeg	April 30, 2023, 23:23:57 (UTC-04:00)	3.4 KB	Standard
0201_193_2_1-150x150.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	6.2 KB	Standard
0201_193_2_1-300x300.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	17.9 KB	Standard
0201_193_2_1-324x324.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	19.4 KB	Standard
0201_193_2_1-416x416.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	28.2 KB	Standard
0201_193_2_1.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	99.5 KB	Standard
0201_194_1_1-100x100.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	2.8 KB	Standard
0201_194_1_1-150x150.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	4.9 KB	Standard
0201_194_1_1-240x300.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	11.3 KB	Standard
0201_194_1_1-324x324.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	15.5 KB	Standard
0201_194_1_1-512x512.jpeg	jpeg	April 30, 2023, 23:23:58 (UTC-04:00)	26.1 KB	Standard

Lambda

Lambda is a useful service to improve our EC2 computing capability. For example, it is easy to compute data as it enters or moves through the cloud. Also, it is free for us to use on a certain scale. We used Lambda to add custom logic to AWS resources which are elaborated below. Also, **we attached the Lambda codes in the appendix 1 which can be found in the last part of this report.**

1. Back end processing
 - a. Process data in the backend
 - b. Manage inventory management
 - c. Product order processing
 - d. Helps reduce website's workload
2. Customer insight developing
 - a. Process data in the backend
 - b. Manage inventory management
 - c. Product order processing
 - d. Helps reduce website's workload
3. Website Features
 - a. Provides email notifications to customers for product information
 - b. Notifies shipping updates and promotional offers

Auto Scaling

We always aim to optimize utilization and cost efficiencies when consuming AWS services. When demand fluctuates, AWS Auto Scaling will automatically add more instances or remove any excess resource capacity to avoid overspending.

Settings of our auto scaling group (ProjectGroup)

The screenshot shows the AWS Auto Scaling Groups interface. At the top, it displays the 'Auto Scaling group: Project Auto Scaling Group'. Below this, there's a 'Load balancing' section with an 'Edit' button. Under 'Load balancer target groups', it lists 'ProjectGroup' under 'Classic Load Balancers'. In the main list, 'Project Auto Scaling Group' is selected, showing its details: Launch template/configuration is 'ProjectConfig', Desired capacity is 2, Minimum capacity is 2, Maximum capacity is 6, and Status is '-'.

Auto Scaling group name	Desired capacity	Status	Amazon Resource Name (ARN)
Project Auto Scaling Group	2	-	arn:aws:autoscaling:us-east-1:008993324755:autoScalingGroup:53b10020-0b28-43a1-a498-00bdd88874a9:autoScalingGroupName/Project Auto Scaling Group
Date created	Minimum capacity		
Sat Apr 15 2023 20:54:59 GMT-0400 (Eastern Daylight Time)	2		
	Maximum capacity		
	6		

Elastic Load Balancer

We chose using Elastic Load Balancer to automatically distribute incoming traffic across multiple EC2 instances, containers, and IP addresses, in one or more Availability Zones. Also, we planned to use this service for monitoring the health of registered targets, and routes traffic only to the healthy targets.

Settings of our Load balancer (ProjectELB)

Load balancers (1/1)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

<input checked="" type="checkbox"/>	Name	DNS name	Status	VPC ID	Availability Z.
<input checked="" type="checkbox"/>	ProjectELB	ProjectELB-1380830547.us-east-1.elb.amazonaws.com	Active	vpc-0ab88c40da193fd3c	2 Availability Z.

Load balancer: ProjectELB

Details

<input checked="" type="checkbox"/> arn:aws:elasticloadbalancing:us-east-1:008993324755:loadbalancer/app/ProjectELB/b93d474a77d878de			
Load balancer type	DNS name	Status	VPC
Application	ProjectELB-1380830547.us-east-1.elb.amazonaws.com (A Record)	Active	vpc-0ab88c40da193fd3c

Load balancer: ProjectELB

IP address type	Scheme	Availability Zones	Hosted zone
IPv4	Internet-facing	subnet-07eb57980543901ed us-east-1b (use1-az4) subnet-0715c57a7122218f1 us-east-1a (use1-az2)	Z35SXDOTRQ7X7K

Settings of our Target groups in EC2 (ProjectGroup)

EC2 > Target groups

Target groups (1/1) Info

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type
<input checked="" type="checkbox"/>	ProjectGroup	arn:aws:elasticloadbalanci...	80	HTTP	Instance

Target group: ProjectGroup

<input checked="" type="checkbox"/> arn:aws:elasticloadbalancing:us-east-1:008993324755:targetgroup/ProjectGroup/a1e202c564d5039e			
Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0ab88c40da193fd3c
IP address type	Load balancer		
IPv4	ProjectELB		

Settings of our Alarm

The screenshot shows the AWS CloudWatch Alarms interface. At the top, there are buttons for 'Hide Auto Scaling alarms' (unchecked), 'Clear selection' (disabled), 'Create composite alarm' (disabled), 'Actions' (dropdown), and 'Create alarm' (button). Below this is a search bar and filters for 'Any state', 'Any type', and 'Any actions ...'. A table lists two alarms:

Name	State	Last state update	Conditions	Actions
TargetTracking-Project Auto Scaling Group- AlarmLow-c11832d9- c5f4-4acb-a19d- 551657c26eea	⚠ In alarm	2023-04-29 04:03:25	CPUUtilization < 35 for 15 datapoints within 15 minutes	↻ Actions ↻
TargetTracking-Project Auto Scaling Group- AlarmHigh-b8401aac- f6e6-4079-a6d1- 7bac95fee7c4	🕒 OK	2023-04-27 04:07:25	CPUUtilization > 50 for 3 datapoints within 3 minutes	↻ Actions ↻

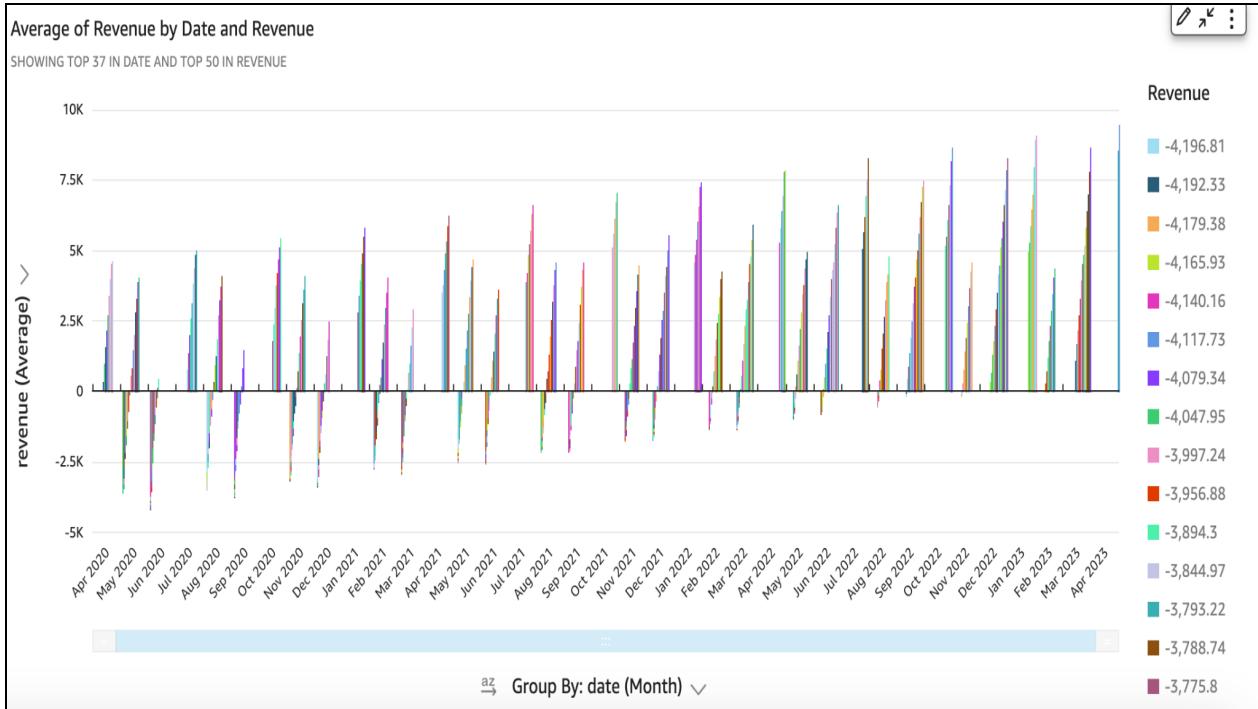
SageMaker

SageMaker provides capability to build Machine learning models. Its capability to quickly connect to training data and select the best algorithm for application, while managing all of the underlying infrastructure.

The code for Sagemaker is attached at the end of this report in appendix 2.

QuickSight

We connected AWS QuickSight to our simulated orders dataset created from SageMaker, and we then developed visualizations and dashboards to analyze our orders data. This is a convenient tool to develop business strategies because it provides interactive data visualizations and analytics capabilities with high customization.



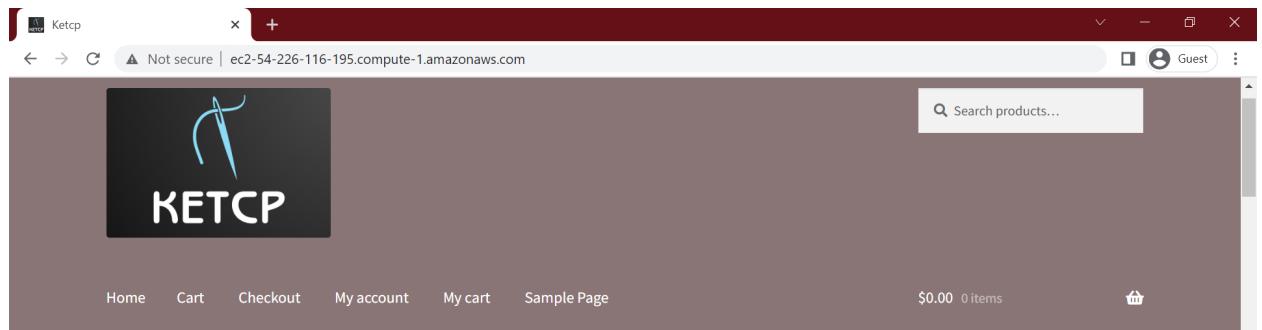
Website Demonstration

With Lambda implementation and wordpress utilization, we designed our backend and frontend of our webpage. In the webpage, customers can browse the items of roly, select different categories and add items to the shopping cart to check out.

Link of the screen video recording:

https://drive.google.com/file/d/1vh8h3zdpvHcjD8a2aD3nZAvhLKNigY7g/view?usp=share_link

Screenshots:



Shop



What are you looking for?

Search



New Arrivals!



Ketcp Not secure | ec2-54-226-116-195.compute-1.amazonaws.com Guest :

New Arrivals!

Accessories (2) Baby (17) Men (173)

Bermuda Shorts

Women (45)

Cart – Ketcp Not secure | ec2-54-226-116-195.compute-1.amazonaws.com/cart/ Guest :

Home > Cart

Cart

Product	Price	Quantity	Subtotal
DAYTONA T-SHIRT NAVY BLUE	\$25.00	1	\$25.00

Coupon code Apply coupon

What are you looking for? Search

New Arrivals!

Bermuda Shorts

Cart totals

Subtotal	\$25.00
----------	---------

Cart – Ketcp x +

Not secure | ec2-54-226-116-195.compute-1.amazonaws.com/cart/ Guest ::

Coupon code Apply coupon Update cart

Cart totals

Subtotal	\$25.00
Shipping	Free shipping
Shipping to NY.	
Change address	
Total	\$25.00

Proceed to checkout →

 **Bermuda Shorts**

 **Dry-Fit Women T-Shirt**

Checkout – Ketcp x +

Not secure | ec2-54-226-116-195.compute-1.amazonaws.com/checkout/ Guest ::

Home > Checkout

Checkout

Have a coupon? [Click here to enter your code](#)

Billing details

First name *	Last name *
Country / Region *	
United States (US)	
Street address *	
House number and street name	
Apartment, suite, unit, etc. (optional)	
Town / City *	
State *	
New York	
ZIP Code *	
Phone *	
Email address *	

Your order

Product	Subtotal
DAYTONA T-SHIRT NAVY BLUE = 1	\$25.00
Subtotal	\$25.00
Shipping	Free shipping
Total	\$25.00

What are you looking for?

New Arrivals!

 **Bermuda Shorts**

 **Dry-Fit Women T-Shirt**

 **SPRING 2023**

Sorry, it seems that there are no available payment methods for your state. Please contact us if you require assistance or wish to make alternate arrangements.

Your personal data will be used to process your order, support your experience throughout this website, and for other purposes described in our privacy policy.

Place order

Security Plan

Security plan for managing access and securing resources is crucial for every cloud architecture. In our cloud system, we have several security measurements listed below.

1. Key Pairs on local : roly.pem

We use it to securely authenticate and connect to an EC2 instance. We keep private keys secure, stored on our local machine or a secure key management system.

2. KMS to manage keys to support encryption

We manage keys that are used to encrypt data. By using KMS, we can ensure that sensitive data is protected and can only be accessed by authorized users.

3. IAM role Setting : IAM policy, IAM group

We created IAM roles and checked the IAM policies. We plan to assign different permissions to different roles to maintain the security levels. We also created IAM policies for S3 to make our S3 secured under limited access and encrypted.

4. CloudTrail

We set up CloudTrail to log, monitor, and retain account activity within your AWS environment. We can also track user activity, API usage, and changes to your resources. This helps us to identify security threats, troubleshoot operational issues, and ensure compliance with regulatory requirements.

Settings of our CloudTrail (apan5450_trailname)

apan5450_trailname			
General details			
Trail logging <input checked="" type="checkbox"/> Logging	Trail log location ketcp/AWSLogs/008993324755	Log file validation Disabled	SNS notification delivery Disabled
Trail name apan5450_trailname	Last log file delivered April 21, 2023, 17:00:51 (UTC-04:00)	Last file validation delivered -	Last SNS notification -
Multi-region trail Yes	Log file SSE-KMS encryption Not enabled		
Apply trail to my organization Not enabled			

5. ACL in AWS Shield

ACL is a list of rules that define the traffic flow that is allowed in and out of a subnet. We checked the rules for our system by setting up the Web ACLs

Settings of our ACL in AWS Shield

The screenshot shows the AWS WAF Web ACLs interface. At the top, a green success message states: "Success You successfully created the web ACL ketcp." Below this, the navigation bar shows "AWS WAF > Web ACLs". The main area has tabs for "Web ACLs" (selected) and "Info". A search bar is present. The table lists one item:

Name	Description	ID
ketcp	-	04722808-65a1-4d8b-87a3-1ee833eefc5b

Employing the above security measurements, we aim to secure our data effectively and efficiently.

Cost Analysis

For cost analysis, we analyzed two different perspectives, as shown below;

1. Comparing AWS service to a second cloud service provider

Next, we compared AWS to a second cloud service provider. There are several cloud service providers in the market, we should always use cloud service calculators to choose the right provider for our use. In this case, we compared using Microsoft Azure for three years or using AWS for three years. Although there would not be an exact same service as AWS and Azure, we can still find similar products and compare the difference. Below is a chart that shows the total cost of the two cloud service providers and the details.

We found that AWS is cheaper after adding up all the service costs that we will use.

Service Type	AWS Service	AWS cost	Azure Service	Azure cost
computing	EC2	\$822	Virtual Machine Scale Sets	\$6,300
computing	Lambda	\$0	Function	\$0
database	RDS	\$10,908	Azure SQL DB	\$13,464
Network	VPC	\$0	Virtual Network	\$144
storage	S3	\$18	Storage	\$53
Cloud Service	CloudFront	\$0	CDN	\$93
Cloud Service	Elastic Load Balancing	\$2,063	Load Balancer	\$0
Cloud Service	Auto Scaling	\$0		
Total		\$13,811		\$0
				\$20,053

2. Conducting a ten-year cost projection using AWS service

Lastly, we would like to have a better picture of the long-term expenses of AWS. Using the AWS price calculator, we can calculate the approximate total cost for running the AWS service for ten years. Taking a closer look at the chart below, we can see that there is only one CAPEX which is EC2 and the remaining services are OPEX and mostly pay-as-you-go, calculating from the amount we use per month. The projecting expense on AWS service for 10 years is \$46,407. Most importantly, this is even cheaper than the on-premises one year plan.

Service Type	AWS Service	AWS price	AWS price	AWS cost	AWS cost
Service Type	AWS Service	CAPEX	OPEX per month	OPEX in10years	Total
computing	EC2	\$2,892	\$0	\$0	\$2,892
computing	Lambda	\$0	\$0	\$0	\$0
database	RDS	\$0	\$303	\$36,360	\$36,360
Network	VPC	\$0	\$0	\$0	\$0
storage	S3	\$0	\$0.49	\$59	\$59
Cloud Service	CloudFront	\$0	\$0	\$0	\$0
Cloud Service	Elastic Load Balancing	\$0	\$57	\$6,877	\$6,877
Cloud Service	Auto Scaling	\$0	\$0	\$0	\$0
Cloud Service	sagemaker	\$0	\$10	\$352	\$352
Cloud Service	quicksight	\$0	\$26	\$936	\$936
Total		\$2,892	\$397	\$44,584	\$47,476

Success Criteria

In this part, we will walk through our eight success criteria and do evaluation based on the current cloud architecture designed.

Success Criteria	Criteria Details	Weight	Evaluation
Retrieve Products Information	Real-time updates from Roly API GET requests can be used to monitor inventory levels and the access to the dataset	12.25%	Fully marked. We successfully uploaded all our photos and products onto AWS and used wordpress to implement our website.
Data Processing and Storage	Set up an Amazon RDS instance for MySQL and establish connectivity between it and the EC2 instance.	12.25%	Fully marked. We successfully set up an RDS of MySQL and connected to EC2.

AWS Elastic Load Balancing and Auto Scaling	Configure an Elastic Load Balancer to distribute traffic to multiple EC2 instances and improve scalability.	12.25%	Fully marked. We successfully set up Elastic Load Balancer and Auto Scaling.
Configure Amazon S3	Set up an Amazon S3 bucket to store product images and configure WooCommerce to use Amazon S3.	12.25%	Fully marked. We successfully set up S3 and check its availability by using scp to copy and sync files from local.
Create AWS Lambda Functions	Develop AWS Lambda functions to perform additional operations, like processing orders, and handling payment gateway integration.	12.25%	Fully marked. We successfully built Lambda to process data in the backend and to reduce website's workload.
AWS QuickSight	Connect AWS QuickSight to the WooCommerce database and create visualizations and dashboards	12.25%	Fully marked. We successfully created Quicksight and created visualizations with our own datasets.
Amazon SageMaker	Develop machine learning models to predict future sales based on historical data	12.25%	Fully marked. We successfully used the LSTM model to predict revenue over time. Also, we simulated orders using a cosine function and constant positive trends for the last three years.
Functionality and Performance	Develop an interactive dashboard with customizable filters for products details to help customers make informed decisions	12.25%	Fully marked. We successfully built a website prototype with products and product details to provide customers a user friendly webpage.

Conclusion

Employing the AWS services, Roly can not only build a more flexible infrastructure for the ecommerce but also save capital cost and operational costs. Moreover, AWS provides secured system services which is important when Roly keeps growing its e-commerce scope. In our implementation, we found the efficiency of using EC2, RDS and S3 to operate the infrastructure. Also using Lambda, Auto-Scaling, Elastic Load Balancer, Sagemaker, and QuickSight enables Roly to perform additional features. In this way, Roly develops a more effective and efficient system to run e-commerce.

Appendix 1

EC2 & LAMBDA SCREENSHOTS

EC2

The screenshot shows the AWS EC2 instance configuration page. At the top, there's a section titled "Summary" with a dropdown arrow. Below it, the "Number of instances" is set to 1. The "Software Image (AMI)" is listed as "Amazon Linux 2 Kernel 5.10 AMI...read more" with the ID "ami-069aabeee6f53e7bf". The "Virtual server type (instance type)" is "t2.micro". Under "Storage (volumes)", it says "1 volume(s) - 30 GiB". A callout box in the bottom right corner provides information about the free tier: "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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet." There is a close button "X" in the top right corner of the callout box.

! "#\$%&%(`#\$*)#%" +%" - . (%/#)0#)%

```
$ sudo yum update -y
```

1. Install the lamp-mariadb10.2-php7.2 and php7.2 Amazon Linux Extras repositories to get the latest versions of the LAMP MariaDB and PHP packages for Amazon Linux 2.

```
$ sudo amazon-linux-extras install -y lamp-mariadb10.2-
php7.2 php7.2
```

If you receive an error stating sudo: amazon-linux-extras: command not found, then your instance was not launched with an Amazon Linux 2 AMI (perhaps you are using the Amazon Linux AMI instead). You can view your version of Amazon Linux using the following command.

```
$ cat /etc/system-release
```

To set up a LAMP web server on Amazon Linux AMI , see [Install LAMP on Amazon Linux](#).

2. Now that your instance is current, you can install the Apache web server, MariaDB, and PHP software packages.

Use the yum install command to install multiple software packages and all related dependencies at the same time.

```
$ sudo yum install -y httpd mariadb-server
```

You can view the current versions of these packages using the following command:

```
yum info package_name
```

3. Start the Apache web server.

```
$ sudo systemctl start httpd
```

4. Use the systemctl command to configure the Apache web server to start at each system boot.

```
$ sudo systemctl enable httpd
```

You can verify that httpd is on by running the following command:

```
$ sudo systemctl is-enabled httpd
```

```
$ sudo chkconfig httpd on
```

Step 2: Set basic file permissions

Add a `www` group:

```
$ sudo groupadd www
```

Add the `ec2-user` to the `www` group:

```
$ sudo usermod -a -G www ec2-user
```

Now, log out and then log back in again so that the `ec2-user` picks up the new permissions:

```
$ sudo exit
```

```
$ ssh -i MyKeyName.pem ec2-user@public-IP-address
```

Now, change the ownership of the web root. Your website's root folder is located in `/var/www`, so that's the directory whose permissions you'll be modifying. First, change the group ownership of `/var/www`:

```
$ sudo chown -R root:www /var/www
```

Then, change directory permissions for `/var/www` and its subdirectories to add write permissions for the `www` group:

```
$ sudo chmod 2775 /var/www
```

```
$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

Finally, recursively change the file permissions of `/var/www` and its subdirectories to add group write permissions:

```
$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Step 3: Set up the MySQL server

The next task at hand is to set up the MySQL Server that will host your website's database:

```
$ sudo service mysqld start
```

N*+\$)0\$%&'()@GRW9H

```
$ sudo mysql_secure_installation
```

N*+@GRW9)0\$%&'('&08\$)0%)=80\$)=&8)'%F)?8/)8,+)+E8\$?0=:/&08\$)1+&'0(%H

- A/+%\$&M*\$+\$)'%F+1)?8//+\$&)/88&)-'%%J8/1)K1+?:(&0%)\$8)-'%%J8/1L
- NG->M'\$1)&*+\$+\$&*&8)E/+&+)'%+E:/+)-'%%J8/1)?8/G8:/)88&):%+/
- NG+-)0\$)G8:/)-'%%J8/1)'\$1)&*+\$)A+\$&%
- X.+\$&+/)(&*+)-'%%J8/1)&8)E8\$?0/,)0&)'\$1)&*+\$)A+\$&%
- NG->M'\$1)-/+%\$&*&8)/+,8>+')\$8\$G,8%):%+)%
- NG->M'\$1)-/+%\$&*&8)10%'2(+)/+,8&+)/88&)(8=0\$
- NG->M'\$1)-/+%\$&*&8)/+,8>+)&*+)&+%%&)1'&'2'%+
- NG->M'\$1)-/+%\$&*&8)/+(8'1)&*+)1'&'2'%+)-/0>0(+=+%)&'2(+

V0\$'((GI)G8:)J'\$&)&8)E&FE8\$?8&)+((()@GRW9)&8)%&'&):&8,'&0E'((G)J*+\$)&*+)%+/>/)288&%):-H

```
$ sudo chkconfig mysqld on
```

Y'*88Z)M8:)8J)*'>+')?)\$E&08\$0\$=)96@A)J+2)%+/>+/"")6((()&*'&T%)(+?&)&8)18)0%)
0\$%&'(')\$1)%+&):-)Y8/1A/+%%I)'\$1)&*+\$)G8:T((2+)=881)&8)=8"

Step 4: Download WordPress

N8)0\$%&'((Y8/1A/+%%I):\$1->E8,,'\$1")V0/%&I)\$'>0='&+2'E&F)&8)GE4
:\$%+[%)*8,+)+10/+E&8/GI)'\$1)&*+\$)18J\$(8'1)Y8/1A/+%%&)&8)0&H

```
$ cd /home/ec2-user
```

```
$ wget https://wordpress.org/latest.tar.gz
```

O8JI):\$70-)&*+)?0(+H

```
$ tar -xzf latest.tar.gz
```

M8:T((8J)*'>+')J8/1-/+%\$0/+E&8/G)&*+'&)E8\$&'0\$%')((8?)&*+)&YA)?0(+%"")R0\$E+)
J+2%0&+T%)/88&)+E&8/GI)'\$1)&*+\$)10\$&,(\$G8:T((*)'>+)&8),8>+)&*+)&YA)?0(+%)
&*+/-)H

```
$ mv wordpress/* /var/www/html/
```

ust to confirm, you can move to the `/var/www/html/` directory, run an `ls` command to list the directory's contents, and verify that the WordPress files and folders are there:

```
$ cd /var/www/html  
$ ls
```

Step 5: Set up the MySQL database

Next, create a database user for WordPress to use. Start MySQL:

```
$ mysql -u root -p
```

Enter the password you created in the last tutorial for the MySQL root user and press enter.

```
mysql> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED  
BY 'your_password';
```

Next, create a database for WordPress to use:

```
mysql> CREATE DATABASE `wordpress-db`;
```

And then grant full database privileges to the wordpress user you created:

```
mysql> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO  
"wordpressuser"@"localhost";
```

Finally, flush MySQL privileges and exit the MySQL client:

```
mysql> FLUSH PRIVILEGES;  
mysql> exit
```

Step 6: Set up WordPress' wp-config.php file

First, navigate back to your website directory:

```
$ cd /var/www/html
```

The standard WordPress installation includes a sample `wp-config-sample.php` file, so copy that to the live `wp-config.php` file:

```
$ cp wp-config-sample.php wp-config.php
```

Now you can use the `nano` editor to configure WordPress:

```
$ sudo nano wp-config.php
```

This will open the file editor, and you should replace the details below with the database and user credentials you created in the steps above:

```
define('DB_NAME', 'wordpress-db');  
define('DB_USER', 'wordpress-user');  
define('DB_PASSWORD', 'your_password');
```

Step 7: Modify your Apache configuration to allow WordPress to use permalinks

For WordPress to properly manage your URL structure, you'll need to override some Apache default settings. First, pull up the Apache's `httpd.conf` file in the `nano` editor:

```
$ sudo nano /etc/httpd/conf/httpd.conf
```

Now, find the section that starts with `<Directory "/var/www/html">` and make some adjustments. In particular, change the `AllowOverride None` to read `AllowOverride All`.

Make sure you **only** made this change in the `<Directory "/var/www/html">` section, and then save the file and exit:

- Press **ctrl+x**
 - Type **Y** and then press **enter**
-

Step 8: Further refine file permissions

You just need to make a few final permission changes so that WordPress can function as intended. Since WordPress uses the `apache` user to perform certain

functions, make sure that apache is granted the proper permissions. First, add apache to the www group:

```
$ sudo usermod -a -G www apache
```

Next, change the file ownership of /var/www and its contents to the apache user:

```
$ sudo chown -R apache /var/www
```

Then, change the group ownership of /var/www and its contents to the www group:

```
$ sudo chgrp -R www /var/www
```

Next, change the directory permission of /var/www and its subdirectories to add group write permissions for current and future directories:

```
$ sudo chmod 2775 /var/www
```

```
$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

Finally, recursively change the file permissions of /var/www and its subdirectories to include group write permissions:

```
$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now, all you need to do is restart Apache, and you're good to go:

```
$ sudo service httpd restart
```

```

[cristianleo@Cristians-Laptop Key_Pair % ssh -i "roly.pem" ec2-user@ec2-54-196-217-252.compute-1.amazonaws.com
The authenticity of host 'ec2-54-196-217-252.compute-1.amazonaws.com (54.196.217.252)' can't be established.
ED25519 key fingerprint is SHA256:CeiGTzlc7Gp+kPFx0brAx0EBGhxqzwYwa9+Cgy29NNs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-196-217-252.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

--| --|_
_| (   /   Amazon Linux 2 AMI
---|\---|_]

https://aws.amazon.com/amazon-linux-2/
[[ec2-user@ip-10-0-2-212 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package glibc.x86_64 0:2.26-62.amzn2 will be updated
--> Package glibc.x86_64 0:2.26-63.amzn2 will be an update
--> Package glibc-all-langpacks.x86_64 0:2.26-62.amzn2 will be updated
--> Package glibc-all-langpacks.x86_64 0:2.26-63.amzn2 will be an update
--> Package glibc-common.x86_64 0:2.26-62.amzn2 will be updated
--> Package glibc-common.x86_64 0:2.26-63.amzn2 will be an update
--> Package glibc-locale-source.x86_64 0:2.26-62.amzn2 will be updated
--> Package glibc-locale-source.x86_64 0:2.26-63.amzn2 will be an update
--> Package glibc-minimal-langpack.x86_64 0:2.26-62.amzn2 will be updated
--> Package glibc-minimal-langpack.x86_64 0:2.26-63.amzn2 will be an update
--> Package kernel.x86_64 0:5.10.177-158.645.amzn2 will be installed
--> Package libcrypt.x86_64 0:2.26-62.amzn2 will be updated
--> Package libcrypt.x86_64 0:2.26-63.amzn2 will be an update
--> Package tzdata.noarch 0:2022g-1.amzn2.0.2 will be updated
--> Package tzdata.noarch 0:2023c-1.amzn2.0.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository        Size
=====
Installing:
kernel          x86_64    5.10.177-158.645.amzn2    amzn2extra-kernel-5.10  33 M
Updating:
glibc            x86_64    2.26-63.amzn2          amzn2-core        3.3 M
glibc-all-langpacks x86_64    2.26-63.amzn2          amzn2-core        7.0 M
glibc-common     x86_64    2.26-63.amzn2          amzn2-core        774 k
glibc-locale-source x86_64    2.26-63.amzn2          amzn2-core        3.2 M
glibc-minimal-langpack x86_64    2.26-63.amzn2          amzn2-core        33 k
libcrypt          x86_64    2.26-63.amzn2          amzn2-core        53 k
tzdata           noarch    2023c-1.amzn2.0.1       amzn2-core        482 k

Transaction Summary
=====
Install 1 Package
Upgrade 7 Packages

Total download size: 48 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/8): glibc-2.26-63.amzn2.x86_64.rpm                                | 3.3 MB  00:00:00
(2/8): glibc-common-2.26-63.amzn2.x86_64.rpm                            | 774 kB  00:00:00
(3/8): glibc-locale-source-2.26-63.amzn2.x86_64.rpm                     | 3.2 MB  00:00:00
(4/8): glibc-all-langpacks-2.26-63.amzn2.x86_64.rpm                   | 7.0 MB  00:00:00
(5/8): glibc-minimal-langpack-2.26-63.amzn2.x86_64.rpm                 | 33 kB  00:00:00
(6/8): libcrypt-2.26-63.amzn2.x86_64.rpm                               | 53 kB  00:00:00
(7/8): tzdata-2023c-1.amzn2.0.1.noarch.rpm                           | 482 kB  00:00:00
(8/8): kernel-5.10.177-158.645.amzn2.x86_64.rpm                      | 33 MB  00:00:00

Total                                         36 MB/s | 48 MB  00:00:01

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : tzdata-2023c-1.amzn2.0.1.noarch                                1/15
  Updating : glibc-common-2.26-63.amzn2.x86_64                             2/15
  Updating : glibc-2.26-63.amzn2.x86_64                                    3/15

```

```

Updating : tzdata-2023c-1.amzn2.0.1.noarch 1/15
Updating : glibc-common-2.26-63.amzn2.x86_64 2/15
Updating : glibc-2.26-63.amzn2.x86_64 3/15
Updating : glibc-minimal-langpack-2.26-63.amzn2.x86_64 4/15
Updating : glibc-locale-source-2.26-63.amzn2.x86_64 5/15
Updating : glibc-all-langpacks-2.26-63.amzn2.x86_64 6/15
Updating : libcrypt-2.26-63.amzn2.x86_64 7/15
Installing : kernel-5.10.177-158.645.amzn2.x86_64 8/15
Cleanup : glibc-all-langpacks-2.26-62.amzn2.x86_64 9/15
Cleanup : glibc-locale-source-2.26-62.amzn2.x86_64 10/15
Cleanup : libcrypt-2.26-62.amzn2.x86_64 11/15
Cleanup : glibc-2.26-62.amzn2.x86_64 12/15
Cleanup : glibc-minimal-langpack-2.26-62.amzn2.x86_64 13/15
Cleanup : glibc-common-2.26-62.amzn2.x86_64 14/15
Cleanup : tzdata-2022g-1.amzn2.0.2.noarch 15/15
Verifying : glibc-minimal-langpack-2.26-63.amzn2.x86_64 1/15
Verifying : glibc-locale-source-2.26-63.amzn2.x86_64 2/15
Verifying : kernel-5.10.177-158.645.amzn2.x86_64 3/15
Verifying : glibc-2.26-63.amzn2.x86_64 4/15
Verifying : glibc-common-2.26-63.amzn2.x86_64 5/15
Verifying : glibc-all-langpacks-2.26-63.amzn2.x86_64 6/15
Verifying : tzdata-2023c-1.amzn2.0.1.noarch 7/15
Verifying : libcrypt-2.26-63.amzn2.x86_64 8/15
Verifying : tzdata-2022g-1.amzn2.0.2.noarch 9/15
Verifying : glibc-all-langpacks-2.26-62.amzn2.x86_64 10/15
Verifying : libcrypt-2.26-62.amzn2.x86_64 11/15
Verifying : glibc-minimal-langpack-2.26-62.amzn2.x86_64 12/15
Verifying : glibc-2.26-62.amzn2.x86_64 13/15
Verifying : glibc-common-2.26-62.amzn2.x86_64 14/15
Verifying : glibc-locale-source-2.26-62.amzn2.x86_64 15/15

Installed:
  kernel.x86_64 0:5.10.177-158.645.amzn2

Updated:
  glibc.x86_64 0:2.26-63.amzn2           glibc-all-langpacks.x86_64 0:2.26-63.amzn2
  glibc-common.x86_64 0:2.26-63.amzn2    glibc-locale-source.x86_64 0:2.26-63.amzn2
  glibc-minimal-langpack.x86_64 0:2.26-63.amzn2   libcrypt.x86_64 0:2.26-63.amzn2
  tzdata.noarch 0:2023c-1.amzn2.0.1

Complete!
[[ec2-user@ip-10-0-2-212 ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
Topic lamp-mariadb10.2-php7.2 has end-of-support date of 2020-11-30
Topic php7.2 has end-of-support date of 2020-11-30
Installing php-pdo, php-mysqlnd, php-fpm, php-cli, php-json, mariadb
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Existing lock /var/run/yum.pid: another copy is running as pid 6340.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: yum
    Memory : 178 M RSS (395 MB VSZ)
    Started: Mon Apr 24 21:20:55 2023 - 00:07 ago
    State  : Running, pid: 6340
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-lamp-mariadb10.2-php7.2 amzn2extra-php7.2
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         | 3.7 kB  00:00:00
amzn2extra-docker                                | 3.0 kB  00:00:00
amzn2extra-kernel-5.10                            | 3.0 kB  00:00:00
amzn2extra-lamp-mariadb10.2-php7.2                | 3.0 kB  00:00:00
amzn2extra-php7.2                                | 3.0 kB  00:00:00
(1/11): amzn2-core/2/x86_64/group_gz            | 2.5 kB  00:00:00
(2/11): amzn2-core/2/x86_64/updateinfo          | 591 kB  00:00:00
(3/11): amzn2extra-docker/2/x86_64/updateinfo   | 9.1 kB  00:00:00
(4/11): amzn2extra-docker/2/x86_64/primary_db    | 107 kB  00:00:00
(5/11): amzn2extra-lamp-mariadb10.2-php7.2/2/x86_64/updateinfo | 76 B   00:00:00
(6/11): amzn2extra-php7.2/2/x86_64/updateinfo    | 76 B   00:00:00
(7/11): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 27 kB  00:00:00
(8/11): amzn2extra-kernel-5.10/2/x86_64/primary_db | 17 MB  00:00:00
(9/11): amzn2extra-php7.2/2/x86_64/primary_db     | 580 kB  00:00:00
(10/11): amzn2extra-lamp-mariadb10.2-php7.2/2/x86_64/primary_db | 506 kB  00:00:00
(11/11): amzn2-core/2/x86_64/primary_db          | 71 MB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 3:10.2.38-1.amzn2.0.1 will be installed

```

```

--> Processing Dependency: mariadb-libs(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-common(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-10.2.38-1.amzn2.0.1.x86_64
--> Package php-cli.x86_64 0:7.2.34-1.amzn2 will be installed
--> Processing Dependency: php-common(x86-64) = 7.2.34-1.amzn2 for package: php-cli-7.2.34-1.amzn2.x86_64
--> Package php-fpm.x86_64 0:7.2.34-1.amzn2 will be installed
--> Package php-json.x86_64 0:7.2.34-1.amzn2 will be installed
--> Package php-mysqlnd.x86_64 0:7.2.34-1.amzn2 will be installed
--> Package php-pdo.x86_64 0:7.2.34-1.amzn2 will be installed
--> Running transaction check
--> Package mariadb-common.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Processing Dependency: /etc/my.cnf for package: 3:mariadb-common-10.2.38-1.amzn2.0.1.x86_64
--> Package mariadb-libs.x86_64 1:5.5.68-1.amzn2 will be updated
--> Package mariadb-libs.x86_64 3:10.2.38-1.amzn2.0.1 will be an update
--> Package php-common.x86_64 0:7.2.34-1.amzn2 will be installed
--> Processing Dependency: libzip.so.5()(64bit) for package: php-common-7.2.34-1.amzn2.x86_64
--> Running transaction check
--> Package libzip.x86_64 0:1.3.2-1.amzn2.0.1 will be installed
--> Package mariadb-config.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Package mariadb-libs.x86_64 1:5.5.68-1.amzn2 will be updated
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package          Arch      Version           Repository      Size
=====
Installing:
mariadb          x86_64   3:10.2.38-1.amzn2.0.1    amzn2extra-lamp-mariadb10.2-php7.2   6.1 M
php-cli          x86_64   7.2.34-1.amzn2        amzn2extra-php7.2                  4.4 M
php-fpm          x86_64   7.2.34-1.amzn2        amzn2extra-php7.2                  1.5 M
php-json         x86_64   7.2.34-1.amzn2        amzn2extra-php7.2                  71 k
php-mysqlnd      x86_64   7.2.34-1.amzn2        amzn2extra-php7.2                  238 k
php-pdo          x86_64   7.2.34-1.amzn2        amzn2extra-php7.2                  132 k
Installing for dependencies:
libzip            x86_64   1.3.2-1.amzn2.0.1     amzn2-core                           62 k
mariadb-common   x86_64   3:10.2.38-1.amzn2.0.1   amzn2extra-lamp-mariadb10.2-php7.2   58 k
mariadb-config   x86_64   3:10.2.38-1.amzn2.0.1   amzn2extra-lamp-mariadb10.2-php7.2   34 k
php-common       x86_64   7.2.34-1.amzn2        amzn2extra-php7.2                  1.1 M
Updating for dependencies:
mariadb-libs     x86_64   3:10.2.38-1.amzn2.0.1   amzn2extra-lamp-mariadb10.2-php7.2   154 k

Transaction Summary
=====
Install 6 Packages (+4 Dependent packages)
Upgrade   ( 1 Dependent package)

Total download size: 14 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/11): libzip-1.3.2-1.amzn2.0.1.x86_64.rpm                                | 62 kB  00:00:00
(2/11): mariadb-common-10.2.38-1.amzn2.0.1.x86_64.rpm                         | 58 kB  00:00:00
(3/11): mariadb-config-10.2.38-1.amzn2.0.1.x86_64.rpm                         | 34 kB  00:00:00
(4/11): mariadb-libs-10.2.38-1.amzn2.0.1.x86_64.rpm                          | 154 kB 00:00:00
(5/11): mariadb-10.2.38-1.amzn2.0.1.x86_64.rpm                            | 6.1 MB 00:00:00
(6/11): php-common-7.2.34-1.amzn2.x86_64.rpm                             | 1.1 MB 00:00:00
(7/11): php-fpm-7.2.34-1.amzn2.x86_64.rpm                            | 1.5 MB 00:00:00
(8/11): php-cli-7.2.34-1.amzn2.x86_64.rpm                           | 4.4 MB 00:00:00
(9/11): php-json-7.2.34-1.amzn2.x86_64.rpm                          | 71 kB  00:00:00
(10/11): php-mysqlnd-7.2.34-1.amzn2.x86_64.rpm                         | 238 kB 00:00:00
(11/11): php-pdo-7.2.34-1.amzn2.x86_64.rpm                           | 132 kB 00:00:00
=====
Total                                         27 MB/s | 14 MB 00:00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libzip-1.3.2-1.amzn2.0.1.x86_64                                1/12
  Installing : php-common-7.2.34-1.amzn2.x86_64                               2/12
  Installing : php-json-7.2.34-1.amzn2.x86_64                               3/12
  Installing : php-pdo-7.2.34-1.amzn2.x86_64                               4/12
  Installing : 3:mariadb-config-10.2.38-1.amzn2.0.1.x86_64                   5/12
  Installing : 3:mariadb-common-10.2.38-1.amzn2.0.1.x86_64                   6/12
  Updating   : 3:mariadb-libs-10.2.38-1.amzn2.0.1.x86_64                     7/12
  Installing : 3:mariadb-10.2.38-1.amzn2.0.1.x86_64                        8/12
  Installing : php-mysqlnd-7.2.34-1.amzn2.x86_64                                9/12

```

```
Installing : php-mysqlnd-7.2.34-1.amzn2.x86_64          9/12
Installing : php-fpm-7.2.34-1.amzn2.x86_64           10/12
Installing : php-cli-7.2.34-1.amzn2.x86_64           11/12
Cleanup   : 1:mariadb-libs-5.5.68-1.amzn2.x86_64        12/12
Verifying  : php-fpm-7.2.34-1.amzn2.x86_64           1/12
Verifying  : php-cli-7.2.34-1.amzn2.x86_64           2/12
Verifying  : 3:mariadb-common-10.2.38-1.amzn2.0.1.x86_64 3/12
Verifying  : 3:mariadb-libs-10.2.38-1.amzn2.0.1.x86_64 4/12
Verifying  : php-json-7.2.34-1.amzn2.x86_64           5/12
Verifying  : 3:mariadb-config-10.2.38-1.amzn2.0.1.x86_64 6/12
Verifying  : libzip-1.3.2-1.amzn2.0.1.x86_64           7/12
Verifying  : php-mysqlnd-7.2.34-1.amzn2.x86_64           8/12
Verifying  : php-pdo-7.2.34-1.amzn2.x86_64           9/12
Verifying  : 3:mariadb-10.2.38-1.amzn2.0.1.x86_64        10/12
Verifying  : php-common-7.2.34-1.amzn2.x86_64           11/12
Verifying  : 1:mariadb-libs-5.5.68-1.amzn2.x86_64        12/12

Installed:
mariadb.x86_64 3:10.2.38-1.amzn2.0.1      php-cli.x86_64 0:7.2.34-1.amzn2      php-fpm.x86_64 0:7.2.34-1.amzn2
php-json.x86_64 0:7.2.34-1.amzn2      php-mysqlnd.x86_64 0:7.2.34-1.amzn2      php-pdo.x86_64 0:7.2.34-1.amzn2

Dependency Installed:
libzip.x86_64 0:1.3.2-1.amzn2.0.1      mariadb-common.x86_64 3:10.2.38-1.amzn2.0.1
mariadb-config.x86_64 3:10.2.38-1.amzn2.0.1      php-common.x86_64 0:7.2.34-1.amzn2

Dependency Updated:
mariadb-libs.x86_64 3:10.2.38-1.amzn2.0.1

Complete!
 0 ansible2           available  \
    [ =2.4.2  =2.4.6  =2.8  =stable ]
 2 httpd_modules       available  [ =1.0  =stable ]
 3 memcached1.5        available  \
    [ =1.5.1  =1.5.16  =1.5.17 ]
 6 postgresql10         available  [ =10  =stable ]
 9 R3.4                 available  [ =3.4.3  =stable ]
10 rust1               available  \
    [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0
      =stable ]
15 *php7.2=latest      enabled  \
    [ =7.2.0  =7.2.4  =7.2.5  =7.2.8  =7.2.11  =7.2.13  =7.2.14
      =7.2.16  =7.2.17  =7.2.19  =7.2.21  =7.2.22  =7.2.23
      =7.2.24  =7.2.26  =stable ]
17 *lamp+mariadb10.2-php7.2=latest enabled  \
    [ =10.2.10_7.2.0  =10.2.10_7.2.4  =10.2.10_7.2.5
      =10.2.10_7.2.8  =10.2.10_7.2.11  =10.2.10_7.2.13
      =10.2.10_7.2.14  =10.2.10_7.2.16  =10.2.10_7.2.17
      =10.2.10_7.2.19  =10.2.10_7.2.22  =10.2.10_7.2.23
      =10.2.10_7.2.24  =stable ]
18 libreoffice          available  \
    [ =5.0.6.2_15  =5.3.6.1  =stable ]
19 gimp                 available  [ =2.8.22 ]
20 docker=latest        enabled  \
    [ =17.12.1  =18.03.1  =18.06.1  =18.09.9  =stable ]
21 mate-desktop1.x       available  \
    [ =1.19.0  =1.20.0  =stable ]
22 GraphicsMagick1.3    available  \
    [ =1.3.29  =1.3.32  =1.3.34  =stable ]
23 tomcat8.5            available  \
    [ =8.5.31  =8.5.32  =8.5.38  =8.5.40  =8.5.42  =8.5.50
      =stable ]
24 epel                  available  [ =7.11  =stable ]
25 testing               available  [ =1.0  =stable ]
26 ecs                  available  [ =stable ]
27 corretto8             available  \
    [ =1.8.0_192  =1.8.0_202  =1.8.0_212  =1.8.0_222  =1.8.0_232
      =1.8.0_242  =stable ]
29 golang1.11            available  \
    [ =1.11.3  =1.11.11  =1.11.13  =stable ]
30 squid                 available  [ =4  =stable ]
32 lustre2.10            available  \
    [ =2.10.5  =2.10.8  =stable ]
33 java-openjdk11         available  [ =11  =stable ]
34 lynis                 available  [ =stable ]
36 BCC                  available  [ =0.x  =stable ]
37 mono                 available  [ =5.x  =stable ]
```

```

41 postgresql11           available   [ =11 =stable ]
43 livepatch               available   [ =stable ]
44 python3.8                available   [ =stable ]
45 haproxy2                 available   [ =stable ]
46 collectd                 available   [ =stable ]
47 aws-nitro-enclaves-cli  available   [ =stable ]
48 R4                      available   [ =stable ]
49 kernel-5.4                available   [ =stable ]
50 selinux-ng               available   [ =stable ]
51 php8.0                   available   [ =stable ]
52 tomcat9                  available   [ =stable ]
53 unbound1.13              available   [ =stable ]
54 mariadb10.5              available   [ =stable ]
55 kernel-5.10=latest       enabled     [ =stable ]
56 redis6                   available   [ =stable ]
57 ruby3.0                  available   [ =stable ]
58 postgresql12              available   [ =stable ]
59 postgresql13              available   [ =stable ]
60 mock2                    available   [ =stable ]
61 dnsmasq2.85              available   [ =stable ]
62 kernel-5.15              available   [ =stable ]
63 postgresql14              available   [ =stable ]
64 firefox                  available   [ =stable ]
65 lustre                   available   [ =stable ]
66 php8.1                   available   [ =stable ]
67 awscli1                  available   [ =stable ]
68 php8.2                   available   [ =stable ]
69 dnsmasq                  available   [ =stable ]
70 unbound1.17              available   [ =stable ]
71 golang1.19               available   [ =stable ]
* Extra topic has reached end of support.
[[ec2-user@ip-10-0-2-212 ~]$ cat /etc/system-release
Amazon Linux release 2 (Karoo)
[[ec2-user@ip-10-0-2-212 ~]$ sudo yum install -y httpd mariadb-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.56-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.56-1.amzn2 for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.56-1.amzn2 for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.56-1.amzn2.x86_64
-->> Package mariadb-server.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Processing Dependency: mariadb-tokudb-engine(x86-64) = 3:10.2.38-1.amzn2.x86_64 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-server-utils(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-rocksdb-engine(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-gssapi-server(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-errmsg(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-cracklib-password-check(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: mariadb-backup(x86-64) = 3:10.2.38-1.amzn2.0.1 for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: bison for package: 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64
--> Running transaction check
-->> Package apr.x86_64 0:1.7.2-1.amzn2 will be installed
-->> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64
-->> Package bison.x86_64 0:3.0.4-6.amzn2.0.2 will be installed
--> Processing Dependency: m4 >= 1.4 for package: bison-3.0.4-6.amzn2.0.2.x86_64
-->> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.56-1.amzn2 will be installed
-->> Package httpd-tools.x86_64 0:2.4.56-1.amzn2 will be installed
-->> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
-->> Package mariadb-backup.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
-->> Package mariadb-cracklib-password-check.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
-->> Package mariadb-errmsg.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
-->> Package mariadb-gssapi-server.x86_64 3:10.2.38-1.amzn2.0.1 will be installed

```

```

--> Package mariadb-rocksdb-engine.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Package mariadb-server-utils.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Data::Dumper) for package: 3:mariadb-server-utils-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(DBI) for package: 3:mariadb-server-utils-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(DBD) for package: 3:mariadb-server-utils-10.2.38-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(DBD::mysql) for package: 3:mariadb-server-utils-10.2.38-1.amzn2.0.1.x86_64
--> Package mariadb-tokudb-engine.x86_64 3:10.2.38-1.amzn2.0.1 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Package m4.x86_64 0:1.4.16-10.amzn2.0.2 will be installed
--> Package perl-DBD-MySQL.x86_64 0:4.023-6.amzn2 will be installed
--> Package perl-DBI.x86_64 0:1.627-4.amzn2.0.2 will be installed
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Package perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-PlRPC.noarch 0:0.2020-14.amzn2 will be installed
--> Processing Dependency: perl(Net::Daemon) >= 0.13 for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Test) for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Log) for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Compress::Zlib) for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Running transaction check
--> Package perl-IO-Compress.noarch 0:2.061-2.amzn2 will be installed
--> Processing Dependency: perl(Compress::Raw::Zlib) >= 2.061 for package: perl-IO-Compress-2.061-2.amzn2.noarch
--> Processing Dependency: perl(Compress::Raw::Bzip2) >= 2.061 for package: perl-IO-Compress-2.061-2.amzn2.noarch
--> Package perl-Net-Daemon.noarch 0:0.48-5.amzn2 will be installed
--> Running transaction check
--> Package perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.amzn2.0.2 will be installed
--> Package perl-Compress-Raw-Zlib.x86_64 1:2.061-4.amzn2.0.2 will be installed
--> Finished Dependency Resolution

```

Dependencies Resolved

Package	Arch	Version	Repository	Size
<hr/>				
Installing:				
httpd	x86_64	2.4.56-1.amzn2	amzn2-core	1.4 M
mariadb-server	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	17 M
Installing for dependencies:				
apr	x86_64	1.7.2-1.amzn2	amzn2-core	130 k
apr-util	x86_64	1.6.3-1.amzn2.0.1	amzn2-core	101 k
apr-util-bdb	x86_64	1.6.3-1.amzn2.0.1	amzn2-core	22 k
bison	x86_64	3.0.4-6.amzn2.0.2	amzn2-core	674 k
generic-logos-httdp	noarch	18.0.0-4.amzn2	amzn2-core	19 k
httpd-filesystem	noarch	2.4.56-1.amzn2	amzn2-core	24 k
httpd-tools	x86_64	2.4.56-1.amzn2	amzn2-core	88 k
m4	x86_64	1.4.16-10.amzn2.0.2	amzn2-core	256 k
mailcap	noarch	2.1.41-2.amzn2	amzn2-core	31 k
mariadb-backup	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	5.9 M
mariadb-cracklib-password-check	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	37 k
mariadb-errmsg	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	222 k
mariadb-gssapi-server	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	39 k
mariadb-rocksdb-engine	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	5.5 M
mariadb-server-utils	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	1.6 M
mariadb-tokudb-engine	x86_64	3:10.2.38-1.amzn2.0.1	amzn2extra-lamp-mariadb10.2-php7.2	833 k
mod_http2	x86_64	1.15.19-1.amzn2.0.1	amzn2-core	149 k
perl-Compress-Raw-Bzip2	x86_64	2.061-3.amzn2.0.2	amzn2-core	32 k
perl-Compress-Raw-Zlib	x86_64	1:2.061-4.amzn2.0.2	amzn2-core	58 k
perl-DBD-MySQL	x86_64	4.023-6.amzn2	amzn2-core	141 k
perl-DBI	x86_64	1.627-4.amzn2.0.2	amzn2-core	804 k
perl-Data-Dumper	x86_64	2.145-3.amzn2.0.2	amzn2-core	48 k
perl-IO-Compress	noarch	2.061-2.amzn2	amzn2-core	260 k
perl-Net-Daemon	noarch	0.48-5.amzn2	amzn2-core	51 k
perl-PlRPC	noarch	0.2020-14.amzn2	amzn2-core	36 k

Transaction Summary

Install 2 Packages (+25 Dependent packages)

```

Total download size: 36 M
Installed size: 150 M
Downloading packages:
(1/27): apr-1.7.2-1.amzn2.x86_64.rpm | 130 kB  00:00:00
(2/27): apr-util-1.6.3-1.amzn2.0.1.x86_64.rpm | 101 kB  00:00:00

```

```

Installing : perl-PlRPC-0.2020-14.amzn2.noarch 17/27
Installing : perl-DBI-1.627-4.amzn2.0.2.x86_64 18/27
Installing : perl-DBD-MySQL-4.023-6.amzn2.x86_64 19/27
Installing : 3:mariadb-errmsg-10.2.38-1.amzn2.0.1.x86_64 20/27
Installing : 3:mariadb-rocksdb-engine-10.2.38-1.amzn2.0.1.x86_64 21/27
Installing : 3:mariadb-cracklib-password-check-10.2.38-1.amzn2.0.1.x86_64 22/27
Installing : 3:mariadb-gssapi-server-10.2.38-1.amzn2.0.1.x86_64 23/27
Installing : 3:mariadb-tokudb-engine-10.2.38-1.amzn2.0.1.x86_64 24/27
Installing : 3:mariadb-server-utils-10.2.38-1.amzn2.0.1.x86_64 25/27
Installing : 3:mariadb-backup-10.2.38-1.amzn2.0.1.x86_64 26/27
Installing : 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64 27/27
Verifying  : 3:mariadb-errmsg-10.2.38-1.amzn2.0.1.x86_64 1/27
Verifying  : 1:perl-Compress-Raw-Zlib-2.061-4.amzn2.0.2.x86_64 2/27
Verifying  : mailcap-2.1.41-2.amzn2.noarch 3/27
Verifying  : 3:mariadb-rocksdb-engine-10.2.38-1.amzn2.0.1.x86_64 4/27
Verifying  : 3:mariadb-cracklib-password-check-10.2.38-1.amzn2.0.1.x86_64 5/27
Verifying  : generic-logos-httdp-18.0.0-4.amzn2.noarch 6/27
Verifying  : perl-Compress-Raw-Bzip2-2.061-3.amzn2.0.2.x86_64 7/27
Verifying  : perl-DBI-1.627-4.amzn2.0.2.x86_64 8/27
Verifying  : perl-PlRPC-0.2020-14.amzn2.noarch 9/27
Verifying  : 3:mariadb-gssapi-server-10.2.38-1.amzn2.0.1.x86_64 10/27
Verifying  : 3:mariadb-tokudb-engine-10.2.38-1.amzn2.0.1.x86_64 11/27
Verifying  : httpd-2.4.56-1.amzn2.x86_64 12/27
Verifying  : apr-1.7.2-1.amzn2.x86_64 13/27
Verifying  : perl-DBD-MySQL-4.023-6.amzn2.x86_64 14/27
Verifying  : apr-util-1.6.3-1.amzn2.0.1.x86_64 15/27
Verifying  : perl-Data-Dumper-2.145-3.amzn2.0.2.x86_64 16/27
Verifying  : httpd-filesystem-2.4.56-1.amzn2.noarch 17/27
Verifying  : 3:mariadb-server-10.2.38-1.amzn2.0.1.x86_64 18/27
Verifying  : bison-3.0.4-6.amzn2.0.2.x86_64 19/27
Verifying  : m4-1.4.16-10.amzn2.0.2.x86_64 20/27
Verifying  : httpd-tools-2.4.56-1.amzn2.x86_64 21/27
Verifying  : 3:mariadb-server-utils-10.2.38-1.amzn2.0.1.x86_64 22/27
Verifying  : perl-IO-Compress-2.061-2.amzn2.noarch 23/27
Verifying  : perl-Net-Daemon-0.48-5.amzn2.noarch 24/27
Verifying  : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 25/27
Verifying  : 3:mariadb-backup-10.2.38-1.amzn2.0.1.x86_64 26/27
Verifying  : mod_http2-1.15.19-1.amzn2.0.1.x86_64 27/27

Installed:
httpd.x86_64 0:2.4.56-1.amzn2          mariadb-server.x86_64 3:10.2.38-1.amzn2.0.1

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2           apr-util.x86_64 0:1.6.3-1.amzn2.0.1
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1   bison.x86_64 0:3.0.4-6.amzn2.0.2
generic-logos-httdp.noarch 0:18.0.0-4.amzn2   httpd-filesystem.noarch 0:2.4.56-1.amzn2
httpd-tools.x86_64 0:2.4.56-1.amzn2       m4.x86_64 0:1.4.16-10.amzn2.0.2
mailcap.noarch 0:2.1.41-2.amzn2         mariadb-backup.x86_64 3:10.2.38-1.amzn2.0.1
mariadb-cracklib-password-check.x86_64 3:10.2.38-1.amzn2.0.1   mariadb-errmsg.x86_64 3:10.2.38-1.amzn2.0.1
mariadb-gssapi-server.x86_64 3:10.2.38-1.amzn2.0.1   mariadb-rocksdb-engine.x86_64 3:10.2.38-1.amzn2.0.1
mariadb-server-utils.x86_64 3:10.2.38-1.amzn2.0.1   mariadb-tokudb-engine.x86_64 3:10.2.38-1.amzn2.0.1
mod_http2.x86_64 0:1.15.19-1.amzn2.0.1     perl-Compress-Raw-Zlib.x86_64 0:2.061-3.amzn2.0.2
perl-Compress-Raw-Zlib.x86_64 1:2.061-4.amzn2.0.2   perl-DBD-MySQL.x86_64 0:4.023-6.amzn2
perl-DBI.x86_64 0:1.627-4.amzn2.0.2       perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2
perl-IO-Compress.noarch 0:2.061-2.amzn2   perl-Net-Daemon.noarch 0:0.48-5.amzn2
perl-PlRPC.noarch 0:0.2020-14.amzn2

Complete!
[[ec2-user@ip-10-0-2-212 ~]$ sudo systemctl start httpd ]]
[[ec2-user@ip-10-0-2-212 ~]$ sudo systemctl enable httpd ]]
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service. ]
[[ec2-user@ip-10-0-2-212 ~]$ sudo systemctl is-enabled httpd ]]
enabled
[[ec2-user@ip-10-0-2-212 ~]$ sudo yum install -y httpd24 php56 mysql55-server php56-mysqld ]]
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No package httpd24 available.
No package php56 available.
No package mysql55-server available.
No package php56-mysqld available.
Error: Nothing to do
[[ec2-user@ip-10-0-2-212 ~]$ sudo chkconfig httpd on ]]
Note: Forwarding request to 'systemctl enable httpd.service'.
[[ec2-user@ip-10-0-2-212 ~]$ sudo groupadd www ]]
[[ec2-user@ip-10-0-2-212 ~]$ sudo usermod -a -G www ec2-user ]]
[[ec2-user@ip-10-0-2-212 ~]$ ]]
```

```
[https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-2-212 ~]$ sudo chown -R root:www /var/www
[ec2-user@ip-10-0-2-212 ~]$
[ec2-user@ip-10-0-2-212 ~]$ sudo chmod 2775 /var/www
[ec2-user@ip-10-0-2-212 ~]$
[ec2-user@ip-10-0-2-212 ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-10-0-2-212 ~]$
[ec2-user@ip-10-0-2-212 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-10-0-2-212 ~]$
[ec2-user@ip-10-0-2-212 ~]$ sudo service mysqld start
Redirecting to /bin/systemctl start mysqld.service
Failed to start mysqld.service: Unit not found.
[ec2-user@ip-10-0-2-212 ~]$ sudo systemctl start mariadb
[ec2-user@ip-10-0-2-212 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] n
[ ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
[production environment.

Remove anonymous users? [Y/n] n
... skipping.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

```

Thanks for using MariaDB!
[ec2-user@ip-10-0-2-212 ~]$ sudo service mysqld start
Redirecting to /bin/systemctl start mysqld.service
Failed to start mysqld.service: Unit not found.
[ec2-user@ip-10-0-2-212 ~]$ sudo chkconfig mysqld on
error reading information on service mysqld: No such file or directory
[ec2-user@ip-10-0-2-212 ~]$ 
[ec2-user@ip-10-0-2-212 ~]$ cd /home/ec2-user
[ec2-user@ip-10-0-2-212 ~]$ 
[ec2-user@ip-10-0-2-212 ~]$ wget https://wordpress.org/latest.tar.gz
--2023-04-24 21:26:21-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23018887 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[=====] 23,018,887 31.3MB/s   in 0.7s

2023-04-24 21:26:22 (31.3 MB/s) - 'latest.tar.gz' saved [23018887/23018887]

[ec2-user@ip-10-0-2-212 ~]$ 
[ec2-user@ip-10-0-2-212 ~]$ tar -xzf latest.tar.gz
[ec2-user@ip-10-0-2-212 ~]$ mv wordpress/* /var/www/html/
[ec2-user@ip-10-0-2-212 ~]$ 
[ec2-user@ip-10-0-2-212 ~]$ cd /var/www/html
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ ls
index.php      wp-activate.php    wp-comments-post.php  wp-cron.php        wp-load.php    wp-settings.php    xmlrpc.php
license.txt    wp-admin          wp-config-sample.php  wp-includes       wp-login.php   wp-signup.php
readme.html    wp-blog-header.php wp-content         wp-links-opml.php  wp-mail.php   wp-trackback.php
[ec2-user@ip-10-0-2-212 html]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.2.38-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'apan5450';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE DATABASE `wordpress-db`;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpressuser"@"localhost";
ERROR 1133 (28000): Can't find any matching row in the user table
MariaDB [(none)]>
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

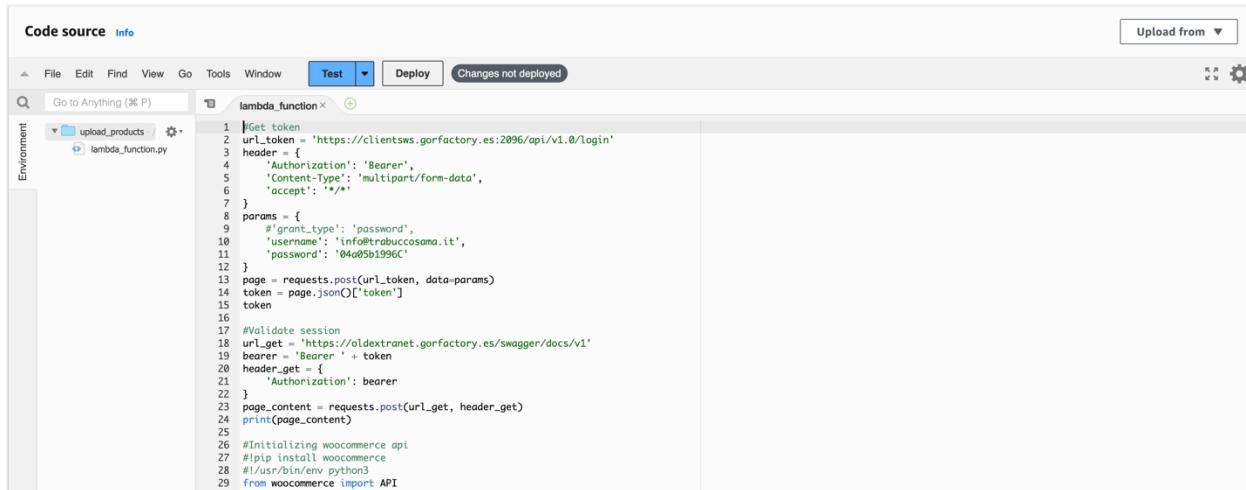
MariaDB [(none)]> cd /var/www/html
-> exit
-> Ctrl-C -- exit!
Aborted
[ec2-user@ip-10-0-2-212 html]$ cd /var/www/html
[ec2-user@ip-10-0-2-212 html]$ cp wp-config-sample.php wp-config.php
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ sudo nano wp-config.php
[ec2-user@ip-10-0-2-212 html]$ sudo nano /etc/httpd/conf/httpd.conf
[ec2-user@ip-10-0-2-212 html]$ sudo usermod -a -G www apache
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ sudo chown -R apache /var/www
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ sudo chgrp -R www /var/www
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ sudo chmod 2775 /var/www
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ sudo chgrp -R www /var/www
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ sudo chmod 2775 /var/www
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ 
[ec2-user@ip-10-0-2-212 html]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-10-0-2-212 html]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-10-0-2-212 html]$ 
```

LAMBDA

The code is a script that imports products from a supplier's API (Roly) and uploads them to our woocommerce store.

1. It first sends a POST request to obtain a token from an API using the provided username and password.
2. It then sends a POST request with the token to a different API to validate the session.
3. It then initializes the woocommerce API with the provided credentials and parameters.
4. It calls a function to get product details from a supplier API for a specific brand.
5. It loops through the list of products obtained from the supplier API and prepares each product to be uploaded to the woocommerce store as a variation of a parent product.
6. It creates a list of variations, images, and stocks for each product.
7. It appends each variation to the list of variations.
8. If the last product has been processed, it prepares a dictionary to post the parent product to the woocommerce store with the attributes of the variations added to the product dictionary.

Limitation: This code is has been tested on a Woocommerce store with HTTPS access. However, for Lab constraints we are not allowed to download a SSL certificate and move the website from HTTP to HTTPS access.



The screenshot shows a Lambda function editor interface. The top navigation bar includes 'Code source' and 'Info' tabs, followed by 'Test' (which is currently selected), 'Deploy', and 'Changes not deployed'. Below the navigation is a search bar labeled 'Go to Anything (⌘ P)' and a file tree on the left containing 'upload_products' and 'lambda_function.py'. The main area displays the Python code for the 'lambda_function.py' file:

```
1 #Get token
2 url_token = 'https://clientsws.gorfactory.es:2096/api/v1.0/login'
3 header = {
4     'Authorization': 'Bearer',
5     'Content-Type': 'multipart/form-data',
6     'accept': '*/*'
7 }
8 params = {
9     #'grant_type': 'password',
10    'username': 'info@trabuccosama.it',
11    'password': '04a0b01996C'
12 }
13 page = requests.post(url_token, data=params)
14 token = page.json()['token']
15 token
16
17 #Validate session
18 url_get = 'https://oldextranet.gorfactory.es/swagger/docs/v1'
19 bearer = 'Bearer ' + token
20 header_get = {
21     'Authorization': bearer
22 }
23 page_content = requests.get(url_get, header_get)
24 print(page_content)
25
26 #Initializing woocommerce api
27 #!pip install woocommerce
28 #!/usr/bin/env python3
29 from woocommerce import API
```

```

Environment upload_products | λ * Environment upload_products | λ * Environment upload_products | λ *
λ lambda_function.py λ lambda_function.py λ lambda_function.py
30 wcapic = API(
31     url="http://ec2-3-208-22-74.compute-1.amazonaws.com",
32     consumer_key="ck_7ec96e5ac03ced629973b833b6041f9f51de8",
33     consumer_secret="cs_7d6488580be7b9184d609906d1a006363acc166",
34     output="JSON",
35     wp_api=True,
36     version="wc/v1",
37     query_string_auth=True # Force Basic Authentication as query string true and using under HTTPS
38 )
39 )
40 wcapic
41
42 def get_roly_products(brand='roly_stamina'):
43     url_model = 'https://clientsws.gorfactory.es:2096/api/v1.0/item/getcatalog'
44     header_roly_products = {
45         'accept': '*/*',
46         'Authorization': 'bearer'
47     }
48
49     params = {
50         'long': 'it-IT',
51         'brand': brand
52     }
53     model_request = requests.get(url=url_model, headers=header_roly_products, params=params)
54     model = model_request.json()['item']
55     return model
56
57 products = get_roly_products('roly')
58
59 #Uploading new products
60 """
61 Iterating over each product in the ids list, getting the item from the supplier and posting them on woocommerce.
62 To know the number of variations I created a dictionary in order to match them for further update
63
64 n_products = len(products)
65 #n_loaded = 1
66 print("Initializing the importation of {} products".format(n_products))
67
68 variations = []
69 images = []
70 stocks = []
71 attributes_color = set()
72 attributes_size = set()
73 for i in range(len(products)):
74     name = products[i]['modelname']
75     parent_code = products[i]['modelcode']
76     child_code = products[i]['itemcode']
77     try:
78         stock = get_quantity(child_code)
79     except:
80         stock = 0
81     stocks.append(stock)
82     child_img = products[i]['productimage']
83     color = products[i]['colorname']
84     size = products[i]['sizeone']
85     single_price = products[i]['single']
86     #box_price = product['price box']
87     #pack_qty = product['qt bag']
88     #pack_price = product['price bag']
89     #single_price = product['price single']
90     attributes_color.add(color)
91     attributes_size.add(size)
92     images.append({'src': str(child_img)})
93
94 var = {
95     'sku': str(child_code),
96     #'parent_id': variation['PARENT_SKU'],
97     'name': str(name),
98     'type': 'variation',
99     '#regular_price': str(single_price),
100    '#price': str(single_price),
101    'image': {'src': str(child_img)},
102    'manage_stock': True,
103    'stock_quantity': str(stock),
104    'attributes': [
105        {
106            'id': 2,
107            'option': str(color)
108        },
109        {
110            'id': 3,
111            'option': str(size) #possible error in case some products don't have color or size
112        }
113    ]
114
115
116 # if the cycle for a new product just started then append the variation
117 if len(variations) == 0:
118     variations.append(var)
119
120 # if the current product is the last product in the list, then upload the product
121 elif i+1 == len(products):
122     if sum(stocks) > 0:
123         stock_status = 'instock'
124     else:
125         stock_status = 'outofstock'
126
127 categories = [{"name": category} for category in products[i]['categories'].split(',')]
128
129 product_to_post = {
130     #'id': product.json()['models'][0]['SKU'],
131     'name': str(products[i]['modelname']),
132     'type': 'variable',
133     'description': str(products[i]['description']),
134     'sku': str(products[i]['modelcode']),
135     '#regular_price': str(round(df_merged.loc[row-1]['price single'], 2)),
136     '#price': str(round(df_merged.loc[row-1]['price single'], 2)),
137     'manage_stock': True,
138     'stock_quantity': int(sum(stocks)),
139     'stock_status': stock_status,
140     'categories': categories,
141     'images': images,
142     'attributes': [
143         {
144             "id": 2,
145             "visible": True,
146             "variation": True,
147             "options": list(attributes_color)
148         },
149         {
150             "id": 3,
151             "visible": True,
152             "variation": True,
153             "options": list(attributes_size)
154         }
155     ],
156     #'variations': variations
157 }
158 #display(product_to_post)
159 response = wcapic.post("products", product_to_post)
160

```

```

Environment upload_products lambda_function.py
159     print('{}) . {} uploaded. {}% Completed.'.format(response, parent_code, round(i/len(products)*100, 2)))
160     if response.status_code == 201:
161         parent_id = wcapi.get('products').json()[0]['id']
162         for variation in variations:
163             data = {"update": variation}
164             response_var = wcapi.post("products/{}/variations".format(parent_id), data)
165             print('{}) . {} variation uploaded'.format(response_var, variation['sku']))
166             variations = []
167             images = []
168             stock = []
169             attributes_color = set()
170             attributes_size = set()
171             variations.append(var)
172
173     # if the first two conditions don't match and the parent code of the current product is the same of this product
174     # then append the variation and continue
175     elif products[i-1]['modelcode'] == parent_code:
176         variations.append(var)
177
178     else:
179         if sum(stocks) > 0:
180             stock_status = 'instock'
181         else:
182             stock_status = 'outofstock'
183
184         categories = [{"name": category} for category in products[i-1]['categories'].split(',')]
185
186         product_to_post = {
187             "id": product.json()['models'][0]['SKU'],
188             "name": str(products[i-1]['modelname']),
189             "description": str(products[i-1]['description']),
190             "regular_price": str(round(products[i-1]['price single'], 2)),
191             "price": str(round(df_merged.iloc[row-1]['price single'], 2)),
192             "manage_stock": True,
193             "stock_quantity": int(sum(stocks)),
194             "stock_status": stock_status,
195             "categories": categories,
196             "images": images,
197             "attributes": [
198                 {
199                     "id": 2,
200                     "visible": True,
201                     "variation": True,
202                     "options": list(attributes_color)
203                 },
204                 {
205                     "id": 3,
206                     "visible": True,
207                     "variation": True,
208                     "options": list(attributes_size)
209                 }
210             ],
211             "variations": variations
212         }
213
214         #display(product_to_post)
215
216     #display(product_to_post)

Environment lambda_function.x
1    lambda_function.x
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

```

Appendix 2

poly_predictions

April 24, 2023

1 Simulate Orders

This code generate revenue data.

First, a date range is generated using the Pandas date_range() function, starting from April 1st, 2020 and ending on April 21st, 2023, with a frequency of 1 day.

Then, a cyclical revenue signal is generated using a sine wave with a period of 90. The np.arange() function generates an array with the length of the date range, and this array is used to calculate the sine wave using the np.sin() function.

A linear trend is added to the cyclical signal by creating a linearly spaced array from 0 to 1 with the same length as the date range, and multiplying it by 5000. The amplitude of the cyclical signal is set to 4500. The revenue is calculated by multiplying the cyclical signal by the amplitude and adding the trend.

A Pandas DataFrame is created with two columns: date and revenue. The date column contains the date range generated previously and the revenue column contains the revenue signal generated.

Finally, the set_index() function can be used to set the date column as the index of the DataFrame. This line of code is commented out in this example, so the default integer index will be used. The head() function is used to display the first few rows of the DataFrame.

```
[11]: import pandas as pd
import numpy as np

# Generate a date range for 30 days
date_range = pd.date_range(start='2020-04-01', end='2023-04-21', freq='D')

# Generate a cyclical revenue signal using a sine wave
period = 90 # set the period of the cyclical signal
x = np.arange(len(date_range))
signal = np.sin(2*np.pi*x/period)

# Add a linear trend to the cyclical signal
trend = np.linspace(0, 1, len(date_range))
amplitude = 4500 # set the amplitude of the cyclical signal
revenue = signal * amplitude + trend * 5000

# Create a DataFrame with the date and revenue columns
```

```

orders = pd.DataFrame({'date': date_range, 'revenue': revenue})

# Set the date column as the index of the DataFrame
#orders.set_index('date', inplace=True)

# Display the DataFrame
orders.head()

```

[11]:

	date	revenue
0	2020-04-01	0.000000
1	2020-04-02	318.388437
2	2020-04-03	635.247564
3	2020-04-04	949.055523
4	2020-04-05	1258.305321

[13]:

[13]: 1116

[15]:

```

def extract_year(col):
    year = col.year
    return int(year)

def extract_month(col):
    month = col.month
    return int(month)

def extract_day(col):
    day = col.day
    return int(day)

orders['Year'] = orders['date'].apply(extract_year)
orders['Month'] = orders['date'].apply(extract_month)
orders['Day'] = orders['date'].apply(extract_day)
orders.head()

```

[15]:

	date	revenue	Year	Month	Day
0	2020-04-01	0.000000	2020	4	1
1	2020-04-02	318.388437	2020	4	2
2	2020-04-03	635.247564	2020	4	3
3	2020-04-04	949.055523	2020	4	4
4	2020-04-05	1258.305321	2020	4	5

[17]:

```

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
orders.iloc[:, 2:] = scaler.fit_transform(orders.iloc[:, 2:])
orders.head()

```

```
[17]:      date    revenue    Year    Month    Day
0 2020-04-01    0.000000 -1.353935 -0.722075 -1.667229
1 2020-04-02   318.388437 -1.353935 -0.722075 -1.553283
2 2020-04-03   635.247564 -1.353935 -0.722075 -1.439337
3 2020-04-04  949.055523 -1.353935 -0.722075 -1.325391
4 2020-04-05  1258.305321 -1.353935 -0.722075 -1.211444
```

```
[19]: !pip install tensorflow
```

```
Looking in indexes: https://pypi.org/simple,
https://pip.repos.neuron.amazonaws.com
Collecting tensorflow
  Downloading tensorflow-2.12.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(585.9 MB)
      585.9/585.9 MB
  545.3 kB/s eta 0:00:0000:0100:01
Collecting astunparse>=1.6.0
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.32.0-cp310-cp310-manylinux_2_12_x86
_64.manylinux2010_x86_64.whl (2.4 MB)
      2.4/2.4 MB
  58.4 MB/s eta 0:00:0000:01
Collecting tensorflow-estimator<2.13,>=2.12.0
  Downloading tensorflow_estimator-2.12.0-py2.py3-none-any.whl (440 kB)
      440.7/440.7 kB
  10.4 MB/s eta 0:00:0000:01
Requirement already satisfied: setuptools in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (65.6.3)
Collecting gast<=0.4.0,>=0.2.1
  Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Collecting opt-einsum>=2.3.2
  Downloading opt_einsum-3.3.0-py3-none-any.whl (65 kB)
      65.5/65.5 kB
  6.6 MB/s eta 0:00:00
Requirement already satisfied: packaging in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (21.3)
Requirement already satisfied: typing-extensions>=3.6.6 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (4.4.0)
Requirement already satisfied: numpy<1.24,>=1.22 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (1.22.3)
Requirement already satisfied:
```

```
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (3.20.3)
Collecting termcolor>=1.1.0
    Downloading termcolor-2.2.0-py3-none-any.whl (6.6 kB)
Requirement already satisfied: six>=1.12.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (1.16.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (1.14.1)
Collecting libclang>=13.0.0
    Downloading libclang-16.0.0-py2.py3-none-manylinux2010_x86_64.whl (22.9 MB)
                                                22.9/22.9 MB
23.6 MB/s eta 0:00:000:0100:01
Collecting flatbuffers>=2.0
    Downloading flatbuffers-23.3.3-py2.py3-none-any.whl (26 kB)
Collecting jax>=0.3.15
    Downloading jax-0.4.8.tar.gz (1.2 MB)
                                                1.2/1.2 MB
46.8 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Collecting absl-py>=1.0.0
    Downloading absl_py-1.4.0-py3-none-any.whl (126 kB)
                                                126.5/126.5 kB
28.4 MB/s eta 0:00:00
Collecting grpcio<2.0,>=1.24.3
    Downloading
grpcio-1.54.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.1
MB)
                                                5.1/5.1 MB
27.2 MB/s eta 0:00:000:0100:01
Collecting tensorboard<2.13,>=2.12
    Downloading tensorboard-2.12.2-py3-none-any.whl (5.6 MB)
                                                5.6/5.6 MB
70.7 MB/s eta 0:00:00ta 0:00:01
Requirement already satisfied: google-pasta>=0.1.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorflow) (3.7.0)
Collecting keras<2.13,>=2.12.0
    Downloading keras-2.12.0-py2.py3-none-any.whl (1.7 MB)
                                                1.7/1.7 MB
64.3 MB/s eta 0:00:00
```

```
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
astunparse>=1.6.0->tensorflow) (0.38.4)
Collecting ml-dtypes>=0.0.3
    Downloading
ml_dtypes-0.1.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (190
kB)
    190.6/190.6 kB
33.3 MB/s eta 0:00:00
Requirement already satisfied: scipy>=1.7 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
jax>=0.3.15->tensorflow) (1.10.0)
Requirement already satisfied: werkzeug>=1.0.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorboard<2.13,>=2.12->tensorflow) (2.2.2)
Collecting google-auth-oauthlib<1.1,>=0.5
    Downloading google_auth_oauthlib-1.0.0-py2.py3-none-any.whl (18 kB)
Collecting google-auth<3,>=1.6.3
    Downloading google_auth-2.17.3-py2.py3-none-any.whl (178 kB)
    178.2/178.2 kB
30.7 MB/s eta 0:00:00
Collecting tensorboard-plugin-wit>=1.6.0
    Downloading tensorboard_plugin_wit-1.8.1-py3-none-any.whl (781 kB)
    781.3/781.3 kB
59.9 MB/s eta 0:00:00
Collecting markdown>=2.6.8
    Downloading Markdown-3.4.3-py3-none-any.whl (93 kB)
    93.9/93.9 kB
20.0 MB/s eta 0:00:00
Requirement already satisfied: requests<3,>=2.21.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
tensorboard<2.13,>=2.12->tensorflow) (2.28.1)
Collecting tensorboard-data-server<0.8.0,>=0.7.0
    Downloading tensorboard_data_server-0.7.0-py3-none-manylinux2014_x86_64.whl
(6.6 kB)
    6.6/6.6 kB
59.6 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
packaging->tensorflow) (3.0.9)
Collecting pyasn1-modules>=0.2.1
    Downloading pyasn1_modules-0.3.0-py2.py3-none-any.whl (181 kB)
    181.3/181.3 kB
25.2 MB/s eta 0:00:00
Requirement already satisfied: rsa<5,>=3.1.4 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow) (4.7.2)
Collecting cachetools<6.0,>=2.0.0
```

```
  Downloading cachetools-5.3.0-py3-none-any.whl (9.3 kB)
Collecting requests-oauthlib>=0.7.0
    Downloading requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: idna<4,>=2.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow) (1.26.8)
Requirement already satisfied: charset-normalizer<3,>=2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow) (2.1.1)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
werkzeug>=1.0.1->tensorboard<2.13,>=2.12->tensorflow) (2.1.1)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow) (0.4.8)
Collecting oauthlib>=3.0.0
    Downloading oauthlib-3.2.2-py3-none-any.whl (151 kB)
          151.7/151.7 kB
24.0 MB/s eta 0:00:00
Building wheels for collected packages: jax
  Building wheel for jax (pyproject.toml) ... done
  Created wheel for jax: filename=jax-0.4.8-py3-none-any.whl size=1439678
sha256=98c9afe33e7accba4da4e9c8735db7da1a6a99d5f29f04bd41a725f99ad55c2
  Stored in directory: /home/ec2-user/.cache/pip/wheels/d6/f6/d5/63686989c723075
de411cbc630ca12f4241a8436e411e38d6a
Successfully built jax
Installing collected packages: tensorboard-plugin-wit, libclang, flatbuffers,
termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorboard-data-
server, pyasn1-modules, opt-einsum, oauthlib, ml-dtypes, markdown, keras,
grpcio, gast, cachetools, astunparse, absl-py, requests-oauthlib, jax, google-
auth, google-auth-oauthlib, tensorboard, tensorflow
Successfully installed absl-py-1.4.0 astunparse-1.6.3 cachetools-5.3.0
flatbuffers-23.3.3 gast-0.4.0 google-auth-2.17.3 google-auth-oauthlib-1.0.0
grpcio-1.54.0 jax-0.4.8 keras-2.12.0 libclang-16.0.0 markdown-3.4.3 ml-
dtypes-0.1.0 oauthlib-3.2.2 opt-einsum-3.3.0 pyasn1-modules-0.3.0 requests-
oauthlib-1.3.1 tensorboard-2.12.2 tensorboard-data-server-0.7.0 tensorboard-
plugin-wit-1.8.1 tensorflow-2.12.0 tensorflow-estimator-2.12.0 tensorflow-io-
gcs-filesystem-0.32.0 termcolor-2.2.0
```

```
[36]: import tensorflow as tf
tf.keras.backend.set_image_data_format("channels_last")

[47]: from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout

# Prepare the data
X = orders.drop(['revenue', 'date'], axis=1).values
print(X.shape)
y = orders['revenue'].values

# Reshape X to match LSTM input shape
X = X.reshape((X.shape[0], 1, X.shape[1]))
print(X.shape)

# Split the data into training and testing sets
split_idx = int(len(X) * 0.7)
X_train, X_test = X[:split_idx], X[split_idx:]
y_train, y_test = y[:split_idx], y[split_idx:]

# Define the LSTM model
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(X.
    ↪shape[1], X.shape[2])))
for n in range(1):
    model.add(LSTM(20, activation='relu', return_sequences=False))
    model.add(Dropout(0.1))
model.add(Dense(20, activation='relu'))
model.add(Dense(1))

# Compile the model with an appropriate learning rate and metric
opt = Adam(learning_rate=0.01)
model.compile(optimizer=opt, loss='mean_absolute_error', metrics=['mae'])

# Define early stopping criteria
early_stopping = EarlyStopping(monitor='val_loss', patience=100)

# Train the model with early stopping
history = model.fit(X_train, y_train, epochs=1000, verbose=1, ↪
    ↪validation_data=(X_test, y_test), callbacks=[early_stopping])

# Use the best model for predictions
y_pred = model.predict(X_test)
x_pred = model.predict(X_train)
```

```

# Calculate the MAE
mae = np.mean(np.abs(y_pred - y_test))

# Print the MAE
print("MAE: ", mae)

# Save the model
model.save("best_model.h5")

```

```

(1116, 3)
(1116, 1, 3)
Epoch 1/1000
25/25 [=====] - 3s 21ms/step - loss: 3187.0127 - mae:
3187.0127 - val_loss: 4413.6470 - val_mae: 4413.6470
Epoch 2/1000
25/25 [=====] - 0s 5ms/step - loss: 3160.6541 - mae:
3160.6541 - val_loss: 4161.4043 - val_mae: 4161.4043
Epoch 3/1000
25/25 [=====] - 0s 4ms/step - loss: 2902.0676 - mae:
2902.0676 - val_loss: 3421.8918 - val_mae: 3421.8918
Epoch 4/1000
25/25 [=====] - 0s 5ms/step - loss: 2774.3706 - mae:
2774.3706 - val_loss: 3199.8247 - val_mae: 3199.8247
Epoch 5/1000
25/25 [=====] - 0s 5ms/step - loss: 2715.3911 - mae:
2715.3911 - val_loss: 3164.0825 - val_mae: 3164.0825
Epoch 6/1000
25/25 [=====] - 0s 5ms/step - loss: 2689.6177 - mae:
2689.6177 - val_loss: 3111.6880 - val_mae: 3111.6880
Epoch 7/1000
25/25 [=====] - 0s 5ms/step - loss: 2692.5852 - mae:
2692.5852 - val_loss: 3031.4836 - val_mae: 3031.4836
Epoch 8/1000
25/25 [=====] - 0s 5ms/step - loss: 2680.9309 - mae:
2680.9309 - val_loss: 3044.9465 - val_mae: 3044.9465
Epoch 9/1000
25/25 [=====] - 0s 5ms/step - loss: 2665.0479 - mae:
2665.0479 - val_loss: 2940.0857 - val_mae: 2940.0857
Epoch 10/1000
25/25 [=====] - 0s 5ms/step - loss: 2643.4309 - mae:
2643.4309 - val_loss: 2934.1272 - val_mae: 2934.1272
Epoch 11/1000
25/25 [=====] - 0s 4ms/step - loss: 2647.8293 - mae:
2647.8293 - val_loss: 2820.0803 - val_mae: 2820.0803
Epoch 12/1000
25/25 [=====] - 0s 5ms/step - loss: 2615.1606 - mae:
2615.1606 - val_loss: 2800.3699 - val_mae: 2800.3699

```

Epoch 13/1000
25/25 [=====] - 0s 5ms/step - loss: 2598.6968 - mae:
2598.6968 - val_loss: 2810.8860 - val_mae: 2810.8860
Epoch 14/1000
25/25 [=====] - 0s 4ms/step - loss: 2584.5769 - mae:
2584.5769 - val_loss: 2768.5293 - val_mae: 2768.5293
Epoch 15/1000
25/25 [=====] - 0s 5ms/step - loss: 2587.2712 - mae:
2587.2712 - val_loss: 2765.8809 - val_mae: 2765.8809
Epoch 16/1000
25/25 [=====] - 0s 5ms/step - loss: 2576.3984 - mae:
2576.3984 - val_loss: 2764.2021 - val_mae: 2764.2021
Epoch 17/1000
25/25 [=====] - 0s 5ms/step - loss: 2576.3093 - mae:
2576.3093 - val_loss: 2765.3164 - val_mae: 2765.3164
Epoch 18/1000
25/25 [=====] - 0s 4ms/step - loss: 2577.4087 - mae:
2577.4087 - val_loss: 2752.1279 - val_mae: 2752.1279
Epoch 19/1000
25/25 [=====] - 0s 5ms/step - loss: 2562.4138 - mae:
2562.4138 - val_loss: 2772.5764 - val_mae: 2772.5764
Epoch 20/1000
25/25 [=====] - 0s 5ms/step - loss: 2561.7163 - mae:
2561.7163 - val_loss: 2771.3264 - val_mae: 2771.3264
Epoch 21/1000
25/25 [=====] - 0s 5ms/step - loss: 2563.2634 - mae:
2563.2634 - val_loss: 2762.5344 - val_mae: 2762.5344
Epoch 22/1000
25/25 [=====] - 0s 4ms/step - loss: 2550.5598 - mae:
2550.5598 - val_loss: 2803.4932 - val_mae: 2803.4932
Epoch 23/1000
25/25 [=====] - 0s 5ms/step - loss: 2561.4678 - mae:
2561.4678 - val_loss: 2737.2183 - val_mae: 2737.2183
Epoch 24/1000
25/25 [=====] - 0s 5ms/step - loss: 2545.2654 - mae:
2545.2654 - val_loss: 2759.9668 - val_mae: 2759.9668
Epoch 25/1000
25/25 [=====] - 0s 4ms/step - loss: 2540.6602 - mae:
2540.6602 - val_loss: 2809.4529 - val_mae: 2809.4529
Epoch 26/1000
25/25 [=====] - 0s 4ms/step - loss: 2561.4250 - mae:
2561.4250 - val_loss: 2736.5935 - val_mae: 2736.5935
Epoch 27/1000
25/25 [=====] - 0s 5ms/step - loss: 2529.0051 - mae:
2529.0051 - val_loss: 2794.4338 - val_mae: 2794.4338
Epoch 28/1000
25/25 [=====] - 0s 4ms/step - loss: 2521.8989 - mae:
2521.8989 - val_loss: 2752.1201 - val_mae: 2752.1201

Epoch 29/1000
25/25 [=====] - 0s 5ms/step - loss: 2503.8047 - mae:
2503.8047 - val_loss: 2831.8240 - val_mae: 2831.8240
Epoch 30/1000
25/25 [=====] - 0s 4ms/step - loss: 2534.4661 - mae:
2534.4661 - val_loss: 2811.2820 - val_mae: 2811.2820
Epoch 31/1000
25/25 [=====] - 0s 4ms/step - loss: 2509.0544 - mae:
2509.0544 - val_loss: 2808.3757 - val_mae: 2808.3757
Epoch 32/1000
25/25 [=====] - 0s 5ms/step - loss: 2491.3923 - mae:
2491.3923 - val_loss: 2827.2493 - val_mae: 2827.2493
Epoch 33/1000
25/25 [=====] - 0s 4ms/step - loss: 2493.0801 - mae:
2493.0801 - val_loss: 2768.8452 - val_mae: 2768.8452
Epoch 34/1000
25/25 [=====] - 0s 4ms/step - loss: 2511.9927 - mae:
2511.9927 - val_loss: 2872.6040 - val_mae: 2872.6040
Epoch 35/1000
25/25 [=====] - 0s 4ms/step - loss: 2529.4104 - mae:
2529.4104 - val_loss: 2830.7678 - val_mae: 2830.7678
Epoch 36/1000
25/25 [=====] - 0s 4ms/step - loss: 2470.9441 - mae:
2470.9441 - val_loss: 2888.0159 - val_mae: 2888.0159
Epoch 37/1000
25/25 [=====] - 0s 5ms/step - loss: 2442.2812 - mae:
2442.2812 - val_loss: 2980.9006 - val_mae: 2980.9006
Epoch 38/1000
25/25 [=====] - 0s 5ms/step - loss: 2452.9102 - mae:
2452.9102 - val_loss: 2898.4644 - val_mae: 2898.4644
Epoch 39/1000
25/25 [=====] - 0s 4ms/step - loss: 2428.8018 - mae:
2428.8018 - val_loss: 2898.3879 - val_mae: 2898.3879
Epoch 40/1000
25/25 [=====] - 0s 4ms/step - loss: 2448.5254 - mae:
2448.5254 - val_loss: 2863.8438 - val_mae: 2863.8438
Epoch 41/1000
25/25 [=====] - 0s 4ms/step - loss: 2410.4912 - mae:
2410.4912 - val_loss: 2843.7312 - val_mae: 2843.7312
Epoch 42/1000
25/25 [=====] - 0s 5ms/step - loss: 2416.3347 - mae:
2416.3347 - val_loss: 2863.9224 - val_mae: 2863.9224
Epoch 43/1000
25/25 [=====] - 0s 4ms/step - loss: 2413.7300 - mae:
2413.7300 - val_loss: 2912.7371 - val_mae: 2912.7371
Epoch 44/1000
25/25 [=====] - 0s 4ms/step - loss: 2435.1040 - mae:
2435.1040 - val_loss: 2884.4097 - val_mae: 2884.4097

Epoch 45/1000
25/25 [=====] - 0s 4ms/step - loss: 2395.4343 - mae:
2395.4343 - val_loss: 2980.7700 - val_mae: 2980.7700
Epoch 46/1000
25/25 [=====] - 0s 5ms/step - loss: 2385.0610 - mae:
2385.0610 - val_loss: 2861.5256 - val_mae: 2861.5256
Epoch 47/1000
25/25 [=====] - 0s 4ms/step - loss: 2376.0044 - mae:
2376.0044 - val_loss: 2876.5320 - val_mae: 2876.5320
Epoch 48/1000
25/25 [=====] - 0s 4ms/step - loss: 2361.3120 - mae:
2361.3120 - val_loss: 2949.0706 - val_mae: 2949.0706
Epoch 49/1000
25/25 [=====] - 0s 4ms/step - loss: 2403.2373 - mae:
2403.2373 - val_loss: 2853.3428 - val_mae: 2853.3428
Epoch 50/1000
25/25 [=====] - 0s 4ms/step - loss: 2400.9639 - mae:
2400.9639 - val_loss: 2863.2615 - val_mae: 2863.2615
Epoch 51/1000
25/25 [=====] - 0s 5ms/step - loss: 2319.9050 - mae:
2319.9050 - val_loss: 2839.1223 - val_mae: 2839.1223
Epoch 52/1000
25/25 [=====] - 0s 7ms/step - loss: 2316.9026 - mae:
2316.9026 - val_loss: 2959.4053 - val_mae: 2959.4053
Epoch 53/1000
25/25 [=====] - 0s 7ms/step - loss: 2303.5339 - mae:
2303.5339 - val_loss: 2912.4807 - val_mae: 2912.4807
Epoch 54/1000
25/25 [=====] - 0s 4ms/step - loss: 2303.8904 - mae:
2303.8904 - val_loss: 2889.0823 - val_mae: 2889.0823
Epoch 55/1000
25/25 [=====] - 0s 5ms/step - loss: 2293.3638 - mae:
2293.3638 - val_loss: 2957.8042 - val_mae: 2957.8042
Epoch 56/1000
25/25 [=====] - 0s 4ms/step - loss: 2353.4756 - mae:
2353.4756 - val_loss: 2905.2205 - val_mae: 2905.2205
Epoch 57/1000
25/25 [=====] - 0s 5ms/step - loss: 2298.2507 - mae:
2298.2507 - val_loss: 2869.1040 - val_mae: 2869.1040
Epoch 58/1000
25/25 [=====] - 0s 4ms/step - loss: 2289.6836 - mae:
2289.6836 - val_loss: 2861.9143 - val_mae: 2861.9143
Epoch 59/1000
25/25 [=====] - 0s 5ms/step - loss: 2255.9365 - mae:
2255.9365 - val_loss: 2896.8975 - val_mae: 2896.8975
Epoch 60/1000
25/25 [=====] - 0s 4ms/step - loss: 2274.1704 - mae:
2274.1704 - val_loss: 2886.5532 - val_mae: 2886.5532

```
Epoch 61/1000
25/25 [=====] - 0s 5ms/step - loss: 2275.2666 - mae: 2275.2666 - val_loss: 2837.5320 - val_mae: 2837.5320
Epoch 62/1000
25/25 [=====] - 0s 5ms/step - loss: 2249.0552 - mae: 2249.0552 - val_loss: 2799.5347 - val_mae: 2799.5347
Epoch 63/1000
25/25 [=====] - 0s 5ms/step - loss: 2234.0862 - mae: 2234.0862 - val_loss: 2890.1541 - val_mae: 2890.1541
Epoch 64/1000
25/25 [=====] - 0s 4ms/step - loss: 2251.9873 - mae: 2251.9873 - val_loss: 2818.4675 - val_mae: 2818.4675
Epoch 65/1000
25/25 [=====] - 0s 4ms/step - loss: 2171.9800 - mae: 2171.9800 - val_loss: 2776.5850 - val_mae: 2776.5850
Epoch 66/1000
25/25 [=====] - 0s 4ms/step - loss: 2178.3027 - mae: 2178.3027 - val_loss: 2804.8018 - val_mae: 2804.8018
Epoch 67/1000
25/25 [=====] - 0s 5ms/step - loss: 2187.4343 - mae: 2187.4343 - val_loss: 2774.1855 - val_mae: 2774.1855
Epoch 68/1000
25/25 [=====] - 0s 4ms/step - loss: 2181.9377 - mae: 2181.9377 - val_loss: 2888.4941 - val_mae: 2888.4941
Epoch 69/1000
25/25 [=====] - 0s 5ms/step - loss: 2216.5886 - mae: 2216.5886 - val_loss: 2869.6392 - val_mae: 2869.6392
Epoch 70/1000
25/25 [=====] - 0s 4ms/step - loss: 2221.7039 - mae: 2221.7039 - val_loss: 2935.1331 - val_mae: 2935.1331
Epoch 71/1000
25/25 [=====] - 0s 5ms/step - loss: 2187.4490 - mae: 2187.4490 - val_loss: 2932.2705 - val_mae: 2932.2705
Epoch 72/1000
25/25 [=====] - 0s 5ms/step - loss: 2160.2656 - mae: 2160.2656 - val_loss: 2830.2046 - val_mae: 2830.2046
Epoch 73/1000
25/25 [=====] - 0s 4ms/step - loss: 2131.8911 - mae: 2131.8911 - val_loss: 2800.6213 - val_mae: 2800.6213
Epoch 74/1000
25/25 [=====] - 0s 4ms/step - loss: 2134.6387 - mae: 2134.6387 - val_loss: 2756.5435 - val_mae: 2756.5435
Epoch 75/1000
25/25 [=====] - 0s 4ms/step - loss: 2114.9287 - mae: 2114.9287 - val_loss: 2893.7869 - val_mae: 2893.7869
Epoch 76/1000
25/25 [=====] - 0s 5ms/step - loss: 2133.7087 - mae: 2133.7087 - val_loss: 2901.9966 - val_mae: 2901.9966
```

```
Epoch 77/1000
25/25 [=====] - 0s 4ms/step - loss: 2064.3057 - mae: 2064.3057 - val_loss: 2908.8398 - val_mae: 2908.8398
Epoch 78/1000
25/25 [=====] - 0s 5ms/step - loss: 2119.8147 - mae: 2119.8147 - val_loss: 2833.0486 - val_mae: 2833.0486
Epoch 79/1000
25/25 [=====] - 0s 5ms/step - loss: 2089.0107 - mae: 2089.0107 - val_loss: 2963.2605 - val_mae: 2963.2605
Epoch 80/1000
25/25 [=====] - 0s 4ms/step - loss: 2075.9951 - mae: 2075.9951 - val_loss: 2846.8494 - val_mae: 2846.8494
Epoch 81/1000
25/25 [=====] - 0s 5ms/step - loss: 2107.1548 - mae: 2107.1548 - val_loss: 2868.1975 - val_mae: 2868.1975
Epoch 82/1000
25/25 [=====] - 0s 4ms/step - loss: 2086.2512 - mae: 2086.2512 - val_loss: 2855.6924 - val_mae: 2855.6924
Epoch 83/1000
25/25 [=====] - 0s 5ms/step - loss: 2047.8617 - mae: 2047.8617 - val_loss: 2913.2756 - val_mae: 2913.2756
Epoch 84/1000
25/25 [=====] - 0s 5ms/step - loss: 2015.1642 - mae: 2015.1642 - val_loss: 3060.1355 - val_mae: 3060.1355
Epoch 85/1000
25/25 [=====] - 0s 5ms/step - loss: 2017.0696 - mae: 2017.0696 - val_loss: 2928.9744 - val_mae: 2928.9744
Epoch 86/1000
25/25 [=====] - 0s 4ms/step - loss: 1987.0231 - mae: 1987.0231 - val_loss: 2665.2551 - val_mae: 2665.2551
Epoch 87/1000
25/25 [=====] - 0s 4ms/step - loss: 2009.8175 - mae: 2009.8175 - val_loss: 2953.2021 - val_mae: 2953.2021
Epoch 88/1000
25/25 [=====] - 0s 4ms/step - loss: 1935.3560 - mae: 1935.3560 - val_loss: 3023.0957 - val_mae: 3023.0957
Epoch 89/1000
25/25 [=====] - 0s 5ms/step - loss: 1910.2598 - mae: 1910.2598 - val_loss: 2982.0461 - val_mae: 2982.0461
Epoch 90/1000
25/25 [=====] - 0s 5ms/step - loss: 1894.5238 - mae: 1894.5238 - val_loss: 3013.0952 - val_mae: 3013.0952
Epoch 91/1000
25/25 [=====] - 0s 4ms/step - loss: 1820.0415 - mae: 1820.0415 - val_loss: 3065.6318 - val_mae: 3065.6318
Epoch 92/1000
25/25 [=====] - 0s 4ms/step - loss: 1692.3718 - mae: 1692.3718 - val_loss: 3105.8777 - val_mae: 3105.8777
```

```
Epoch 93/1000
25/25 [=====] - 0s 4ms/step - loss: 1687.3849 - mae: 1687.3849 - val_loss: 3073.6726 - val_mae: 3073.6726
Epoch 94/1000
25/25 [=====] - 0s 4ms/step - loss: 1649.9507 - mae: 1649.9507 - val_loss: 3032.7683 - val_mae: 3032.7683
Epoch 95/1000
25/25 [=====] - 0s 4ms/step - loss: 1617.0844 - mae: 1617.0844 - val_loss: 3040.3652 - val_mae: 3040.3652
Epoch 96/1000
25/25 [=====] - 0s 5ms/step - loss: 1526.1986 - mae: 1526.1986 - val_loss: 2884.1860 - val_mae: 2884.1860
Epoch 97/1000
25/25 [=====] - 0s 5ms/step - loss: 1475.2465 - mae: 1475.2465 - val_loss: 2986.2148 - val_mae: 2986.2148
Epoch 98/1000
25/25 [=====] - 0s 5ms/step - loss: 1389.3700 - mae: 1389.3700 - val_loss: 2767.6641 - val_mae: 2767.6641
Epoch 99/1000
25/25 [=====] - 0s 4ms/step - loss: 1292.0634 - mae: 1292.0634 - val_loss: 2670.8467 - val_mae: 2670.8467
Epoch 100/1000
25/25 [=====] - 0s 4ms/step - loss: 1330.7054 - mae: 1330.7054 - val_loss: 2478.8667 - val_mae: 2478.8667
Epoch 101/1000
25/25 [=====] - 0s 5ms/step - loss: 1273.1825 - mae: 1273.1825 - val_loss: 2589.4688 - val_mae: 2589.4688
Epoch 102/1000
25/25 [=====] - 0s 5ms/step - loss: 1228.6327 - mae: 1228.6327 - val_loss: 2427.5818 - val_mae: 2427.5818
Epoch 103/1000
25/25 [=====] - 0s 5ms/step - loss: 1203.0112 - mae: 1203.0112 - val_loss: 2464.9172 - val_mae: 2464.9172
Epoch 104/1000
25/25 [=====] - 0s 4ms/step - loss: 1103.9841 - mae: 1103.9841 - val_loss: 2735.6602 - val_mae: 2735.6602
Epoch 105/1000
25/25 [=====] - 0s 5ms/step - loss: 1091.6460 - mae: 1091.6460 - val_loss: 2408.3088 - val_mae: 2408.3088
Epoch 106/1000
25/25 [=====] - 0s 4ms/step - loss: 1121.0469 - mae: 1121.0469 - val_loss: 2142.9976 - val_mae: 2142.9976
Epoch 107/1000
25/25 [=====] - 0s 5ms/step - loss: 1086.8893 - mae: 1086.8893 - val_loss: 2199.8726 - val_mae: 2199.8726
Epoch 108/1000
25/25 [=====] - 0s 4ms/step - loss: 1100.4587 - mae: 1100.4587 - val_loss: 2096.1235 - val_mae: 2096.1235
```

```
Epoch 109/1000
25/25 [=====] - 0s 5ms/step - loss: 1011.6379 - mae:
1011.6379 - val_loss: 2250.9226 - val_mae: 2250.9226
Epoch 110/1000
25/25 [=====] - 0s 5ms/step - loss: 978.1317 - mae:
978.1317 - val_loss: 2220.0276 - val_mae: 2220.0276
Epoch 111/1000
25/25 [=====] - 0s 7ms/step - loss: 1050.4979 - mae:
1050.4979 - val_loss: 2355.2876 - val_mae: 2355.2876
Epoch 112/1000
25/25 [=====] - 0s 5ms/step - loss: 983.5723 - mae:
983.5723 - val_loss: 1924.5582 - val_mae: 1924.5582
Epoch 113/1000
25/25 [=====] - 0s 4ms/step - loss: 977.3777 - mae:
977.3777 - val_loss: 2020.8748 - val_mae: 2020.8748
Epoch 114/1000
25/25 [=====] - 0s 5ms/step - loss: 946.2823 - mae:
946.2823 - val_loss: 2089.2734 - val_mae: 2089.2734
Epoch 115/1000
25/25 [=====] - 0s 5ms/step - loss: 944.4133 - mae:
944.4133 - val_loss: 2195.8997 - val_mae: 2195.8997
Epoch 116/1000
25/25 [=====] - 0s 6ms/step - loss: 944.7653 - mae:
944.7653 - val_loss: 2060.7434 - val_mae: 2060.7434
Epoch 117/1000
25/25 [=====] - 0s 4ms/step - loss: 902.8862 - mae:
902.8862 - val_loss: 1942.3929 - val_mae: 1942.3929
Epoch 118/1000
25/25 [=====] - 0s 4ms/step - loss: 867.2382 - mae:
867.2382 - val_loss: 2091.3518 - val_mae: 2091.3518
Epoch 119/1000
25/25 [=====] - 0s 4ms/step - loss: 870.9985 - mae:
870.9985 - val_loss: 2196.9653 - val_mae: 2196.9653
Epoch 120/1000
25/25 [=====] - 0s 5ms/step - loss: 831.9000 - mae:
831.9000 - val_loss: 1825.6235 - val_mae: 1825.6235
Epoch 121/1000
25/25 [=====] - 0s 5ms/step - loss: 871.1178 - mae:
871.1178 - val_loss: 1817.5306 - val_mae: 1817.5306
Epoch 122/1000
25/25 [=====] - 0s 5ms/step - loss: 846.0820 - mae:
846.0820 - val_loss: 2032.9177 - val_mae: 2032.9177
Epoch 123/1000
25/25 [=====] - 0s 5ms/step - loss: 863.2646 - mae:
863.2646 - val_loss: 1922.2468 - val_mae: 1922.2468
Epoch 124/1000
25/25 [=====] - 0s 5ms/step - loss: 847.1711 - mae:
847.1711 - val_loss: 1965.2504 - val_mae: 1965.2504
```

```
Epoch 125/1000
25/25 [=====] - 0s 4ms/step - loss: 803.0074 - mae: 803.0074 - val_loss: 1902.4244 - val_mae: 1902.4244
Epoch 126/1000
25/25 [=====] - 0s 4ms/step - loss: 791.0256 - mae: 791.0256 - val_loss: 1964.4724 - val_mae: 1964.4724
Epoch 127/1000
25/25 [=====] - 0s 4ms/step - loss: 759.7612 - mae: 759.7612 - val_loss: 2080.0720 - val_mae: 2080.0720
Epoch 128/1000
25/25 [=====] - 0s 4ms/step - loss: 860.1254 - mae: 860.1254 - val_loss: 1867.0042 - val_mae: 1867.0042
Epoch 129/1000
25/25 [=====] - 0s 4ms/step - loss: 845.1813 - mae: 845.1813 - val_loss: 1685.2444 - val_mae: 1685.2444
Epoch 130/1000
25/25 [=====] - 0s 4ms/step - loss: 773.3199 - mae: 773.3199 - val_loss: 1819.2012 - val_mae: 1819.2012
Epoch 131/1000
25/25 [=====] - 0s 5ms/step - loss: 713.7340 - mae: 713.7340 - val_loss: 1910.3679 - val_mae: 1910.3679
Epoch 132/1000
25/25 [=====] - 0s 5ms/step - loss: 718.0511 - mae: 718.0511 - val_loss: 1934.1572 - val_mae: 1934.1572
Epoch 133/1000
25/25 [=====] - 0s 4ms/step - loss: 711.9941 - mae: 711.9941 - val_loss: 1966.4541 - val_mae: 1966.4541
Epoch 134/1000
25/25 [=====] - 0s 7ms/step - loss: 744.9487 - mae: 744.9487 - val_loss: 2049.6523 - val_mae: 2049.6523
Epoch 135/1000
25/25 [=====] - 0s 5ms/step - loss: 654.2192 - mae: 654.2192 - val_loss: 2156.4348 - val_mae: 2156.4348
Epoch 136/1000
25/25 [=====] - 0s 7ms/step - loss: 671.4880 - mae: 671.4880 - val_loss: 1866.4731 - val_mae: 1866.4731
Epoch 137/1000
25/25 [=====] - 0s 5ms/step - loss: 675.8079 - mae: 675.8079 - val_loss: 1959.0061 - val_mae: 1959.0061
Epoch 138/1000
25/25 [=====] - 0s 5ms/step - loss: 687.6906 - mae: 687.6906 - val_loss: 1948.4662 - val_mae: 1948.4662
Epoch 139/1000
25/25 [=====] - 0s 7ms/step - loss: 686.2166 - mae: 686.2166 - val_loss: 1785.4526 - val_mae: 1785.4526
Epoch 140/1000
25/25 [=====] - 0s 7ms/step - loss: 678.5342 - mae: 678.5342 - val_loss: 2057.1902 - val_mae: 2057.1902
```

```
Epoch 141/1000
25/25 [=====] - 0s 8ms/step - loss: 644.1688 - mae: 644.1688 - val_loss: 2115.3521 - val_mae: 2115.3521
Epoch 142/1000
25/25 [=====] - 0s 7ms/step - loss: 655.5674 - mae: 655.5674 - val_loss: 1822.4484 - val_mae: 1822.4484
Epoch 143/1000
25/25 [=====] - 0s 6ms/step - loss: 739.4688 - mae: 739.4688 - val_loss: 1784.9291 - val_mae: 1784.9291
Epoch 144/1000
25/25 [=====] - 0s 7ms/step - loss: 691.7798 - mae: 691.7798 - val_loss: 1682.4136 - val_mae: 1682.4136
Epoch 145/1000
25/25 [=====] - 0s 7ms/step - loss: 637.9730 - mae: 637.9730 - val_loss: 1858.5972 - val_mae: 1858.5972
Epoch 146/1000
25/25 [=====] - 0s 7ms/step - loss: 632.6911 - mae: 632.6911 - val_loss: 1985.2662 - val_mae: 1985.2662
Epoch 147/1000
25/25 [=====] - 0s 7ms/step - loss: 626.3307 - mae: 626.3307 - val_loss: 1836.1711 - val_mae: 1836.1711
Epoch 148/1000
25/25 [=====] - 0s 7ms/step - loss: 644.4196 - mae: 644.4196 - val_loss: 2007.2631 - val_mae: 2007.2631
Epoch 149/1000
25/25 [=====] - 0s 7ms/step - loss: 615.3616 - mae: 615.3616 - val_loss: 2059.7339 - val_mae: 2059.7339
Epoch 150/1000
25/25 [=====] - 0s 7ms/step - loss: 611.2243 - mae: 611.2243 - val_loss: 1952.5465 - val_mae: 1952.5465
Epoch 151/1000
25/25 [=====] - 0s 7ms/step - loss: 605.7150 - mae: 605.7150 - val_loss: 1959.7889 - val_mae: 1959.7889
Epoch 152/1000
25/25 [=====] - 0s 7ms/step - loss: 612.0366 - mae: 612.0366 - val_loss: 1899.4919 - val_mae: 1899.4919
Epoch 153/1000
25/25 [=====] - 0s 7ms/step - loss: 612.2933 - mae: 612.2933 - val_loss: 2088.9446 - val_mae: 2088.9446
Epoch 154/1000
25/25 [=====] - 0s 7ms/step - loss: 658.9139 - mae: 658.9139 - val_loss: 2065.7791 - val_mae: 2065.7791
Epoch 155/1000
25/25 [=====] - 0s 7ms/step - loss: 614.0594 - mae: 614.0594 - val_loss: 1913.0367 - val_mae: 1913.0367
Epoch 156/1000
25/25 [=====] - 0s 7ms/step - loss: 661.3784 - mae: 661.3784 - val_loss: 1999.3821 - val_mae: 1999.3821
```

```
Epoch 157/1000
25/25 [=====] - 0s 5ms/step - loss: 597.4754 - mae: 597.4754 - val_loss: 2058.9397 - val_mae: 2058.9397
Epoch 158/1000
25/25 [=====] - 0s 6ms/step - loss: 626.3558 - mae: 626.3558 - val_loss: 2061.6245 - val_mae: 2061.6245
Epoch 159/1000
25/25 [=====] - 0s 7ms/step - loss: 575.5283 - mae: 575.5283 - val_loss: 2220.3738 - val_mae: 2220.3738
Epoch 160/1000
25/25 [=====] - 0s 5ms/step - loss: 572.3146 - mae: 572.3146 - val_loss: 2137.6165 - val_mae: 2137.6165
Epoch 161/1000
25/25 [=====] - 0s 7ms/step - loss: 611.0471 - mae: 611.0471 - val_loss: 2083.5503 - val_mae: 2083.5503
Epoch 162/1000
25/25 [=====] - 0s 5ms/step - loss: 661.8671 - mae: 661.8671 - val_loss: 2151.7476 - val_mae: 2151.7476
Epoch 163/1000
25/25 [=====] - 0s 5ms/step - loss: 596.3352 - mae: 596.3352 - val_loss: 2102.1804 - val_mae: 2102.1804
Epoch 164/1000
25/25 [=====] - 0s 7ms/step - loss: 488.4964 - mae: 488.4964 - val_loss: 2038.7419 - val_mae: 2038.7419
Epoch 165/1000
25/25 [=====] - 0s 5ms/step - loss: 549.7538 - mae: 549.7538 - val_loss: 2048.0308 - val_mae: 2048.0308
Epoch 166/1000
25/25 [=====] - 0s 5ms/step - loss: 599.3995 - mae: 599.3995 - val_loss: 2128.0105 - val_mae: 2128.0105
Epoch 167/1000
25/25 [=====] - 0s 5ms/step - loss: 583.7462 - mae: 583.7462 - val_loss: 2212.2749 - val_mae: 2212.2749
Epoch 168/1000
25/25 [=====] - 0s 6ms/step - loss: 551.5560 - mae: 551.5560 - val_loss: 2072.8662 - val_mae: 2072.8662
Epoch 169/1000
25/25 [=====] - 0s 5ms/step - loss: 530.0535 - mae: 530.0535 - val_loss: 2184.0195 - val_mae: 2184.0195
Epoch 170/1000
25/25 [=====] - 0s 5ms/step - loss: 577.5148 - mae: 577.5148 - val_loss: 2139.6995 - val_mae: 2139.6995
Epoch 171/1000
25/25 [=====] - 0s 5ms/step - loss: 493.0445 - mae: 493.0445 - val_loss: 2222.5195 - val_mae: 2222.5195
Epoch 172/1000
25/25 [=====] - 0s 4ms/step - loss: 507.0913 - mae: 507.0913 - val_loss: 2252.1001 - val_mae: 2252.1001
```

```
Epoch 173/1000
25/25 [=====] - 0s 6ms/step - loss: 614.5447 - mae: 614.5447 - val_loss: 2281.1658 - val_mae: 2281.1658
Epoch 174/1000
25/25 [=====] - 0s 4ms/step - loss: 586.9212 - mae: 586.9212 - val_loss: 2182.4148 - val_mae: 2182.4148
Epoch 175/1000
25/25 [=====] - 0s 4ms/step - loss: 479.6490 - mae: 479.6490 - val_loss: 2228.2644 - val_mae: 2228.2644
Epoch 176/1000
25/25 [=====] - 0s 4ms/step - loss: 586.1513 - mae: 586.1513 - val_loss: 2578.5901 - val_mae: 2578.5901
Epoch 177/1000
25/25 [=====] - 0s 4ms/step - loss: 506.2101 - mae: 506.2101 - val_loss: 2451.1123 - val_mae: 2451.1123
Epoch 178/1000
25/25 [=====] - 0s 4ms/step - loss: 489.1354 - mae: 489.1354 - val_loss: 2239.1165 - val_mae: 2239.1165
Epoch 179/1000
25/25 [=====] - 0s 5ms/step - loss: 552.8019 - mae: 552.8019 - val_loss: 2222.4529 - val_mae: 2222.4529
Epoch 180/1000
25/25 [=====] - 0s 5ms/step - loss: 560.7542 - mae: 560.7542 - val_loss: 2260.6572 - val_mae: 2260.6572
Epoch 181/1000
25/25 [=====] - 0s 5ms/step - loss: 540.5606 - mae: 540.5606 - val_loss: 2111.6484 - val_mae: 2111.6484
Epoch 182/1000
25/25 [=====] - 0s 5ms/step - loss: 460.5381 - mae: 460.5381 - val_loss: 2339.7642 - val_mae: 2339.7642
Epoch 183/1000
25/25 [=====] - 0s 4ms/step - loss: 456.7904 - mae: 456.7904 - val_loss: 2297.9302 - val_mae: 2297.9302
Epoch 184/1000
25/25 [=====] - 0s 5ms/step - loss: 520.4563 - mae: 520.4563 - val_loss: 2197.8210 - val_mae: 2197.8210
Epoch 185/1000
25/25 [=====] - 0s 4ms/step - loss: 531.6687 - mae: 531.6687 - val_loss: 2310.5962 - val_mae: 2310.5962
Epoch 186/1000
25/25 [=====] - 0s 5ms/step - loss: 594.7181 - mae: 594.7181 - val_loss: 2275.1023 - val_mae: 2275.1023
Epoch 187/1000
25/25 [=====] - 0s 5ms/step - loss: 528.2746 - mae: 528.2746 - val_loss: 2463.9478 - val_mae: 2463.9478
Epoch 188/1000
25/25 [=====] - 0s 4ms/step - loss: 518.3746 - mae: 518.3746 - val_loss: 2420.0764 - val_mae: 2420.0764
```

Epoch 189/1000
25/25 [=====] - 0s 5ms/step - loss: 430.1458 - mae:
430.1458 - val_loss: 2136.8840 - val_mae: 2136.8840
Epoch 190/1000
25/25 [=====] - 0s 5ms/step - loss: 540.5804 - mae:
540.5804 - val_loss: 2176.4490 - val_mae: 2176.4490
Epoch 191/1000
25/25 [=====] - 0s 5ms/step - loss: 476.8987 - mae:
476.8987 - val_loss: 2206.8311 - val_mae: 2206.8311
Epoch 192/1000
25/25 [=====] - 0s 4ms/step - loss: 492.5732 - mae:
492.5732 - val_loss: 2275.3835 - val_mae: 2275.3835
Epoch 193/1000
25/25 [=====] - 0s 4ms/step - loss: 468.8599 - mae:
468.8599 - val_loss: 2429.2346 - val_mae: 2429.2346
Epoch 194/1000
25/25 [=====] - 0s 4ms/step - loss: 498.7211 - mae:
498.7211 - val_loss: 2169.6880 - val_mae: 2169.6880
Epoch 195/1000
25/25 [=====] - 0s 5ms/step - loss: 561.2704 - mae:
561.2704 - val_loss: 2239.1616 - val_mae: 2239.1616
Epoch 196/1000
25/25 [=====] - 0s 5ms/step - loss: 540.5581 - mae:
540.5581 - val_loss: 2495.8892 - val_mae: 2495.8892
Epoch 197/1000
25/25 [=====] - 0s 5ms/step - loss: 532.4470 - mae:
532.4470 - val_loss: 2361.7996 - val_mae: 2361.7996
Epoch 198/1000
25/25 [=====] - 0s 4ms/step - loss: 538.4196 - mae:
538.4196 - val_loss: 2240.7322 - val_mae: 2240.7322
Epoch 199/1000
25/25 [=====] - 0s 5ms/step - loss: 510.6451 - mae:
510.6451 - val_loss: 2367.5100 - val_mae: 2367.5100
Epoch 200/1000
25/25 [=====] - 0s 5ms/step - loss: 490.5682 - mae:
490.5682 - val_loss: 2439.4355 - val_mae: 2439.4355
Epoch 201/1000
25/25 [=====] - 0s 5ms/step - loss: 488.9948 - mae:
488.9948 - val_loss: 2318.5144 - val_mae: 2318.5144
Epoch 202/1000
25/25 [=====] - 0s 4ms/step - loss: 459.4173 - mae:
459.4173 - val_loss: 2429.2600 - val_mae: 2429.2600
Epoch 203/1000
25/25 [=====] - 0s 5ms/step - loss: 495.6776 - mae:
495.6776 - val_loss: 2375.2151 - val_mae: 2375.2151
Epoch 204/1000
25/25 [=====] - 0s 4ms/step - loss: 452.0133 - mae:
452.0133 - val_loss: 2679.3718 - val_mae: 2679.3718

```
Epoch 205/1000
25/25 [=====] - 0s 4ms/step - loss: 496.5165 - mae: 496.5165 - val_loss: 2547.4392 - val_mae: 2547.4392
Epoch 206/1000
25/25 [=====] - 0s 5ms/step - loss: 494.8383 - mae: 494.8383 - val_loss: 2425.0483 - val_mae: 2425.0483
Epoch 207/1000
25/25 [=====] - 0s 5ms/step - loss: 492.7928 - mae: 492.7928 - val_loss: 2433.6045 - val_mae: 2433.6045
Epoch 208/1000
25/25 [=====] - 0s 4ms/step - loss: 450.2238 - mae: 450.2238 - val_loss: 2419.1843 - val_mae: 2419.1843
Epoch 209/1000
25/25 [=====] - 0s 5ms/step - loss: 451.3049 - mae: 451.3049 - val_loss: 2406.2292 - val_mae: 2406.2292
Epoch 210/1000
25/25 [=====] - 0s 4ms/step - loss: 467.8473 - mae: 467.8473 - val_loss: 2593.2605 - val_mae: 2593.2605
Epoch 211/1000
25/25 [=====] - 0s 4ms/step - loss: 459.6328 - mae: 459.6328 - val_loss: 2412.7722 - val_mae: 2412.7722
Epoch 212/1000
25/25 [=====] - 0s 4ms/step - loss: 489.8440 - mae: 489.8440 - val_loss: 2429.6475 - val_mae: 2429.6475
Epoch 213/1000
25/25 [=====] - 0s 5ms/step - loss: 449.1306 - mae: 449.1306 - val_loss: 2796.8035 - val_mae: 2796.8035
Epoch 214/1000
25/25 [=====] - 0s 4ms/step - loss: 465.1569 - mae: 465.1569 - val_loss: 2672.4524 - val_mae: 2672.4524
Epoch 215/1000
25/25 [=====] - 0s 4ms/step - loss: 456.0001 - mae: 456.0001 - val_loss: 2609.1169 - val_mae: 2609.1169
Epoch 216/1000
25/25 [=====] - 0s 4ms/step - loss: 417.0891 - mae: 417.0891 - val_loss: 2677.1714 - val_mae: 2677.1714
Epoch 217/1000
25/25 [=====] - 0s 5ms/step - loss: 525.1398 - mae: 525.1398 - val_loss: 2491.0596 - val_mae: 2491.0596
Epoch 218/1000
25/25 [=====] - 0s 4ms/step - loss: 431.7803 - mae: 431.7803 - val_loss: 2643.9204 - val_mae: 2643.9204
Epoch 219/1000
25/25 [=====] - 0s 4ms/step - loss: 495.3534 - mae: 495.3534 - val_loss: 2666.1025 - val_mae: 2666.1025
Epoch 220/1000
25/25 [=====] - 0s 5ms/step - loss: 477.7967 - mae: 477.7967 - val_loss: 2425.5862 - val_mae: 2425.5862
```

```
Epoch 221/1000
25/25 [=====] - 0s 5ms/step - loss: 487.2490 - mae:
487.2490 - val_loss: 2499.2686 - val_mae: 2499.2686
Epoch 222/1000
25/25 [=====] - 0s 5ms/step - loss: 468.1622 - mae:
468.1622 - val_loss: 2533.7156 - val_mae: 2533.7156
Epoch 223/1000
25/25 [=====] - 0s 4ms/step - loss: 444.3773 - mae:
444.3773 - val_loss: 2635.3855 - val_mae: 2635.3855
Epoch 224/1000
25/25 [=====] - 0s 4ms/step - loss: 487.7523 - mae:
487.7523 - val_loss: 2537.1697 - val_mae: 2537.1697
Epoch 225/1000
25/25 [=====] - 0s 4ms/step - loss: 471.5941 - mae:
471.5941 - val_loss: 2584.5361 - val_mae: 2584.5361
Epoch 226/1000
25/25 [=====] - 0s 5ms/step - loss: 475.7403 - mae:
475.7403 - val_loss: 2493.8438 - val_mae: 2493.8438
Epoch 227/1000
25/25 [=====] - 0s 5ms/step - loss: 451.3504 - mae:
451.3504 - val_loss: 2585.3105 - val_mae: 2585.3105
Epoch 228/1000
25/25 [=====] - 0s 4ms/step - loss: 498.9634 - mae:
498.9634 - val_loss: 2508.7664 - val_mae: 2508.7664
Epoch 229/1000
25/25 [=====] - 0s 4ms/step - loss: 452.4822 - mae:
452.4822 - val_loss: 2620.5178 - val_mae: 2620.5178
Epoch 230/1000
25/25 [=====] - 0s 4ms/step - loss: 447.8397 - mae:
447.8397 - val_loss: 2614.0549 - val_mae: 2614.0549
Epoch 231/1000
25/25 [=====] - 0s 5ms/step - loss: 464.0759 - mae:
464.0759 - val_loss: 2527.1384 - val_mae: 2527.1384
Epoch 232/1000
25/25 [=====] - 0s 5ms/step - loss: 459.8112 - mae:
459.8112 - val_loss: 2499.4116 - val_mae: 2499.4116
Epoch 233/1000
25/25 [=====] - 0s 5ms/step - loss: 404.1570 - mae:
404.1570 - val_loss: 2536.9143 - val_mae: 2536.9143
Epoch 234/1000
25/25 [=====] - 0s 4ms/step - loss: 472.0116 - mae:
472.0116 - val_loss: 2430.9028 - val_mae: 2430.9028
Epoch 235/1000
25/25 [=====] - 0s 4ms/step - loss: 459.8469 - mae:
459.8469 - val_loss: 2610.7046 - val_mae: 2610.7046
Epoch 236/1000
25/25 [=====] - 0s 4ms/step - loss: 438.0642 - mae:
438.0642 - val_loss: 2574.7134 - val_mae: 2574.7134
```

```
Epoch 237/1000
25/25 [=====] - 0s 5ms/step - loss: 470.5535 - mae: 470.5535 - val_loss: 2487.0881 - val_mae: 2487.0881
Epoch 238/1000
25/25 [=====] - 0s 4ms/step - loss: 449.1167 - mae: 449.1167 - val_loss: 2682.6277 - val_mae: 2682.6277
Epoch 239/1000
25/25 [=====] - 0s 5ms/step - loss: 445.7893 - mae: 445.7893 - val_loss: 2553.2366 - val_mae: 2553.2366
Epoch 240/1000
25/25 [=====] - 0s 5ms/step - loss: 409.6593 - mae: 409.6593 - val_loss: 2791.0779 - val_mae: 2791.0779
Epoch 241/1000
25/25 [=====] - 0s 4ms/step - loss: 484.8275 - mae: 484.8275 - val_loss: 2700.4849 - val_mae: 2700.4849
Epoch 242/1000
25/25 [=====] - 0s 4ms/step - loss: 470.8033 - mae: 470.8033 - val_loss: 2667.4761 - val_mae: 2667.4761
Epoch 243/1000
25/25 [=====] - 0s 5ms/step - loss: 448.6470 - mae: 448.6470 - val_loss: 2612.9871 - val_mae: 2612.9871
Epoch 244/1000
25/25 [=====] - 0s 5ms/step - loss: 413.2486 - mae: 413.2486 - val_loss: 2783.1177 - val_mae: 2783.1177
11/11 [=====] - 0s 2ms/step
25/25 [=====] - 0s 1ms/step
MAE: 3778.0048267525367
```

```
[48]: x_pred = [x[0] for x in x_pred]
y_pred = [x[0] for x in y_pred]
y_pred[:5]
```

```
[48]: [-521.4989, -632.7398, -757.0157, -919.6214, -1135.9291]
```

```
[49]: display(len(orders['date'][split_idx:]))
display(len(y_pred))
display(len(orders['revenue'][split_idx:]))
```

```
335
```

```
335
```

```
335
```

```
[50]: import matplotlib.pyplot as plt
import matplotlib.dates as mdates

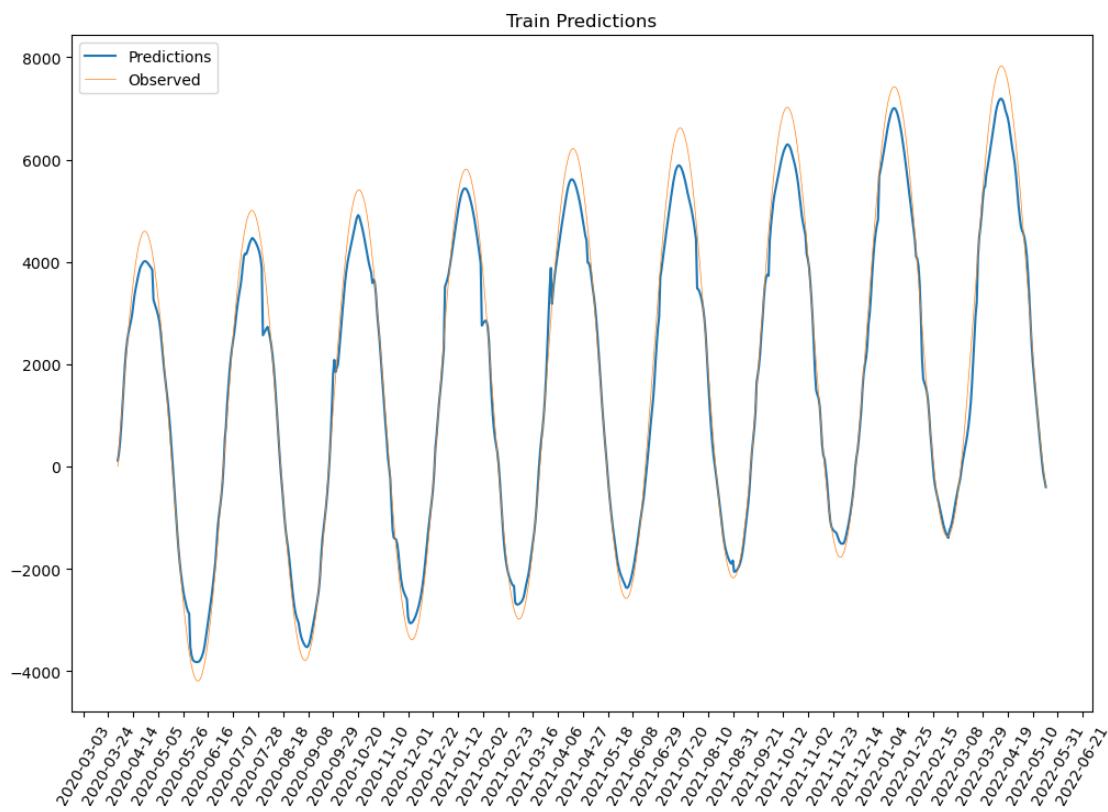
preds = pd.DataFrame({
```

```

'Date': orders['date'][:split_idx],
'Predictions': x_pred,
'Observed': orders['revenue'][:split_idx]
})
preds.to_csv('train_pred.csv')

fig, ax = plt.subplots(figsize=(12,8))
plt.title('Train Predictions')
ax.plot('Date', 'Predictions', data=preds, label='Predictions')
ax.plot('Date', 'Observed', data=preds, label='Observed', linewidth=0.5) # set linewidth to 0.5 for the Observed line
plt.xticks(rotation=60)
ax.xaxis.set_major_locator(mdates.WeekdayLocator(interval=3))
ax.legend()
plt.show()

```

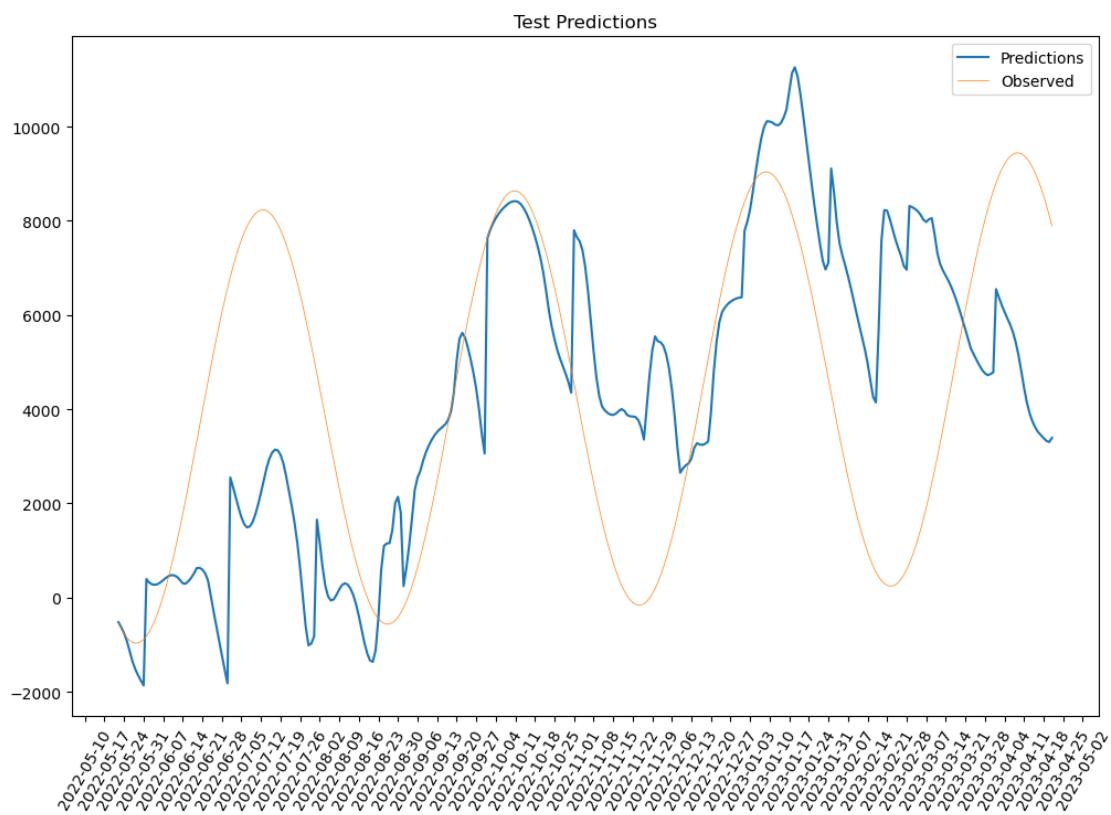


```
[51]: import matplotlib.pyplot as plt
preds = pd.DataFrame({
    'Date': orders['date'][split_idx:],
    'Predictions': y_pred,
    'Observed': orders['revenue'][split_idx:]}
```

```

})
fig, ax = plt.subplots(figsize=(12,8))
plt.title('Test Predictions')
ax.plot('Date', 'Predictions', data=preds, label='Predictions')
ax.plot('Date', 'Observed', data=preds, label='Observed', linewidth=0.5) # set linewidth to 0.5 for the Observed line
plt.xticks(rotation=60)
ax.xaxis.set_major_locator(mdates.WeekdayLocator(interval=1))
ax.legend()
plt.show()

```



```

[54]: # Predict next 2 months
# Prepare the data
from datetime import datetime, timedelta
new_data = pd.DataFrame(orders).tail(60)

def next_2months(date):
    date += timedelta(days=60)
    date = date.strftime('%Y-%m-%d')
    date = pd.to_datetime(date)

```

```

    return date

def extract_year(col):
    #col = datetime.strptime(col, '%Y-%m-%d')
    year = col.year
    return int(year)

def extract_month(col):
    #col = datetime.strptime(col, '%Y-%m-%d')
    month = col.month
    return int(month)

def extract_day(col):
    #col = datetime.strptime(col, '%Y-%m-%d')
    day = col.day
    return int(day)

#new_data['Date'] = pd.to_datetime(new_data['Date'])
new_data['date'] = new_data['date'].apply(next_2months)
new_data['Year'] = new_data['date'].apply(extract_year)
new_data['Month'] = new_data['date'].apply(extract_month)
new_data['Day'] = new_data['date'].apply(extract_day)
new_data.iloc[:, 2:] = scaler.transform(new_data.iloc[:, 2:])
new_data.head()

X = new_data.drop(['revenue', 'date'], axis=1).values
y = new_data['revenue'].tail(60).values

#Reshape X to match LSTM input shape
X = X.reshape((X.shape[0], 1, X.shape[1]))
X.shape
new_pred = model.predict(X)
new_pred = [x[0] for x in new_pred]
new_pred[:5]

```

2/2 [=====] - 0s 4ms/step

[54]: [3484.6653, 3540.3296, 3564.2766, 3564.1687, 3552.3599]

```

[55]: import matplotlib.pyplot as plt
preds = pd.DataFrame({
    'Date': new_data['date'],
    'Predictions': new_pred
})
preds.to_csv('new_preds.csv')

fig, ax = plt.subplots(figsize=(12,8))

```

```
plt.title('Next 2 Predictions')
ax.plot('Date', 'Predictions', data=preds, label='Predictions')
plt.xticks(rotation=60)
ax.xaxis.set_major_locator(mdates.WeekdayLocator(interval=1))
ax.legend()
plt.show()
```

