AWS Project Report: Word Count with PySpark & Node.js Docker Deployment

Web App URL: http://3.89.209.39/
Docker Hub Repo URL: https://hub.docker.com/r/saiharsha027/node-webserver
Github: https://github.com/saiharsha27/word-count-spark-hands-on-11

Task 1: Word Count on AWS Lightsail using PySpark

Objective

To set up Apache Spark on a Lightsail instance and run a PySpark job to perform word count on a text file stored in an S3 bucket.

1. Lightsail Setup

- Instance: Amazon Linux 2 (1 GB RAM, 1 vCPU)
- Connect using SSH

2. Install Required Software

```
sudo yum update -y
sudo yum install java-11 -y
export JAVA_HOME=$(dirname $(dirname $(readlink -f $(which java))))
sudo mount -o remount,size=2G /tmp
sudo yum install python3-pip -y
pip3 install pyspark
```

3. PySpark Word Count Script (`word_count.py`)

```
from pyspark.sql import SparkSession

AWS_ACCESS_KEY_ID = 'YOUR_ACCESS_KEY'
AWS_SECRET_ACCESS_KEY = 'YOUR_SECRET_KEY'
S3_INPUT = 's3a://your-bucket/input_file.txt'
S3_OUTPUT = 's3a://your-bucket/output_folder/'

spark = SparkSession.builder \\\\
    .appName("WordCount") \\\\
    .config("spark.jars.packages",
"org.apache.hadoop:hadoop-aws:3.3.1,com.amazonaws:aws-java-sdk-bundle:1.11.901") \\\\
    .getOrCreate()
```

```
hadoop_conf = spark.sparkContext._jsc.hadoopConfiguration()
hadoop_conf.set("fs.s3a.access.key", AWS_ACCESS_KEY_ID)
hadoop_conf.set("fs.s3a.secret.key", AWS_SECRET_ACCESS_KEY)

text_file = spark.sparkContext.textFile(S3_INPUT)
counts = text_file.flatMap(lambda line: line.split()) \\\\
            .map(lambda word: (word, 1)) \\\\
            .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile(S3_OUTPUT)
spark.stop()
```

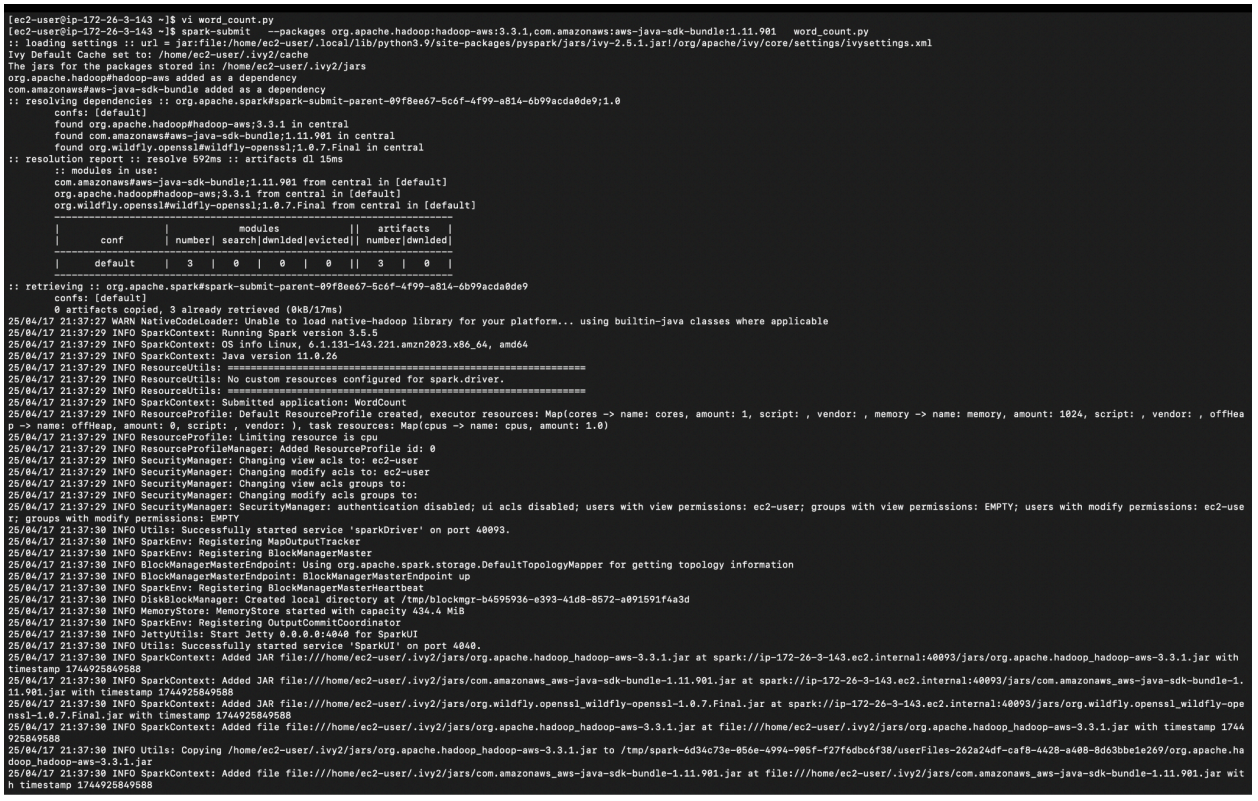
4. Run Spark Job

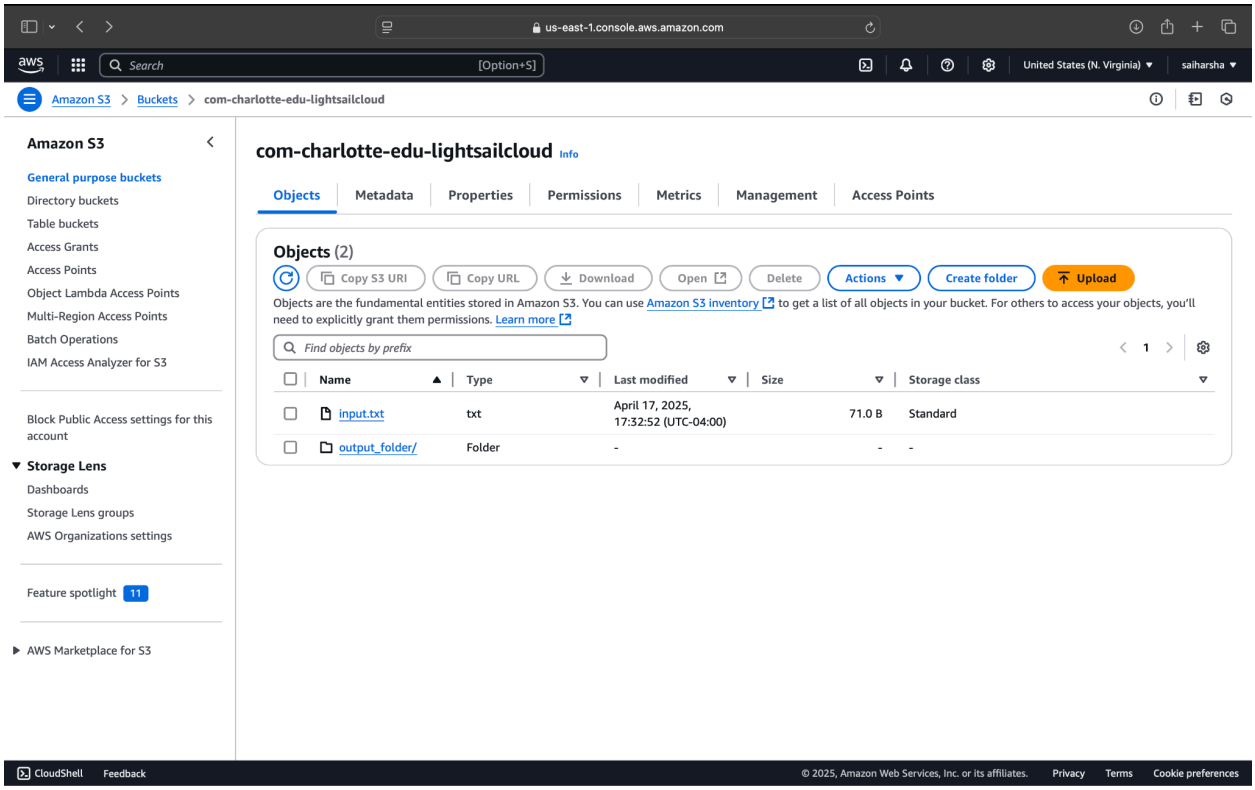
spark-submit word_count.py

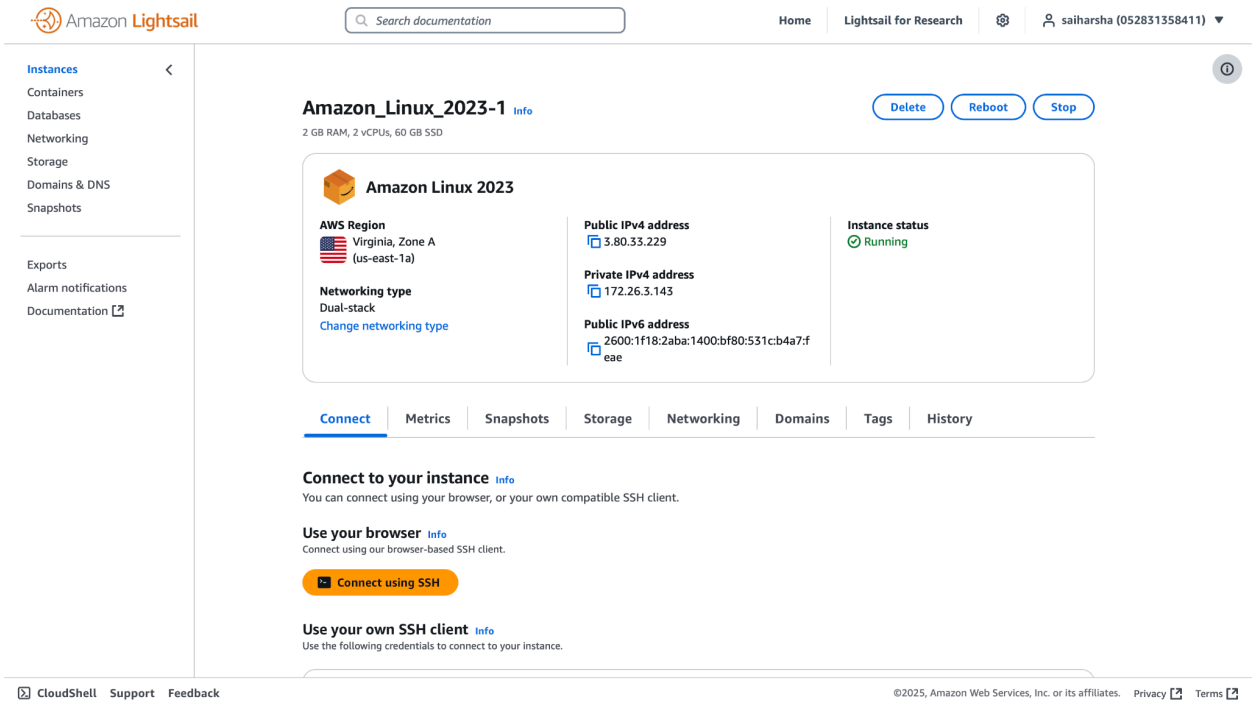
5. Verification

- Output saved to S3 bucket.
- Spark UI accessible at port 4040.

Screenshots Captured




us-east-1.console.aws.amazon.com
Search
[Option+S]
United States (N. Virginia)
saiharsha
Amazon S3 > Buckets > com-charlotte-edu-lightsailcloud
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
com-charlotte-edu-lightsailcloud Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects (2)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
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
Find objects by prefix
Name
Type
Last modified
Size
Storage class
input.txt
txt
April 17, 2025, 17:32:52 (UTC-04:00)
71.0 B
Standard
output_folder/
Folder
-
-
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Amazon Lightsail
Search documentation
Home
Lightsail for Research
saiharsha (052831358411)
Instances
Containers
Databases
Networking
Storage
Domains & DNS
Snapshots
Exports
Alarm notifications
Documentation
Amazon_Linux_2023-1 Info
2 GB RAM, 2 vCPUs, 60 GB SSD
Delete
Reboot
Stop
Amazon Linux 2023
AWS Region
Virginia, Zone A (us-east-1a)
Networking type
Dual-stack
Change networking type
Public IPv4 address
3.80.33.229
Private IPv4 address
172.26.3.143
Public IPv6 address
2600:1f18:2aba:1400:bf80:531c:b4a7:feae
Instance status
Running
Connect
Metrics
Snapshots
Storage
Networking
Domains
Tags
History
Connect to your instance Info
You can connect using your browser, or your own compatible SSH client.
Use your browser Info
Connect using our browser-based SSH client.
Connect using SSH
Use your own SSH client Info
Use the following credentials to connect to your instance.
CloudShell
Support
Feedback
©2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

Task 2: Node.js Web Server with Docker Deployment

Objective

Create a Docker container for a Node.js app, test it locally, push it to Docker Hub, and deploy it on AWS Lightsail.

1. Node.js App Setup

```
mkdir node-webserver && cd node-webserver
npm init -y
npm install express
```

server.js:

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello, World! Running in Docker.');
});

app.listen(port, '0.0.0.0', () => {
  console.log(`Server running at <http://localhost>:${port}`);
});
```

2. Dockerize Application

Dockerfile:

```
FROM node:14
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["node", "server.js"]
```

3. Build and Push Docker Image

docker buildx create --use
docker buildx build --platform linux/amd64 -t saiharsha027/node-webserver:latest --push .

4. Deploy on Lightsail

- Instance: Amazon Linux 2
- Installed Docker:

sudo yum install docker -y
sudo service docker start
sudo usermod -aG docker ec2-user

- Run container:

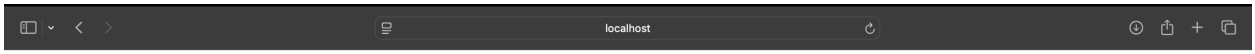
docker pull saiharsha027/node-webserver:latest
docker run -d -p 80:3000 saiharsha027/node-webserver:latest

- Ensure Lightsail firewall allows HTTP (port 80)

5. Verification

- Access via: `http://3.89.209.39/`
- Page should show: **Hello, World! Running in Docker.**

Screenshots Captured



Hello, World! Running in Docker.