

Hands-on #11: AWS EC2 & S3 : Word count & webapp

Name: TarunKumar Kanakala

Task 1: Spark on EC2

Objective:

Deploy Apache Spark on an AWS EC2 instance and execute a sample job to validate the deployment.

Steps Followed:

1. EC2 Instance Setup:

- Launched Amazon Linux 2 EC2 instance (t2.medium)
- Installed Java, Python3, pip, and PySpark
- Configured security group to allow SSH and Spark UI ports (e.g., 8080)
- Increased `/tmp` size to avoid memory-related Spark issues

2. Spark Installation:

- Downloaded and extracted Spark
- Set environment variables for Spark and Hadoop
- Started Spark master and worker

3. Sample Job Execution:

- Uploaded a sample text file to S3
- Created `word_count.py` script using PySpark to perform word count on S3 file
- Ran the job with:

```
spark-submit word_count.py
```

- Verified job status and output written back to S3

Screenshots Taken:

- EC2 instance setup screen

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

44.211.167.30

Private IPv4 address

172.26.6.23

Public IPv6 address

2600:1f18:4747:7e00:928f:f16f:4181:166b

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](#)

- ## Terminal output

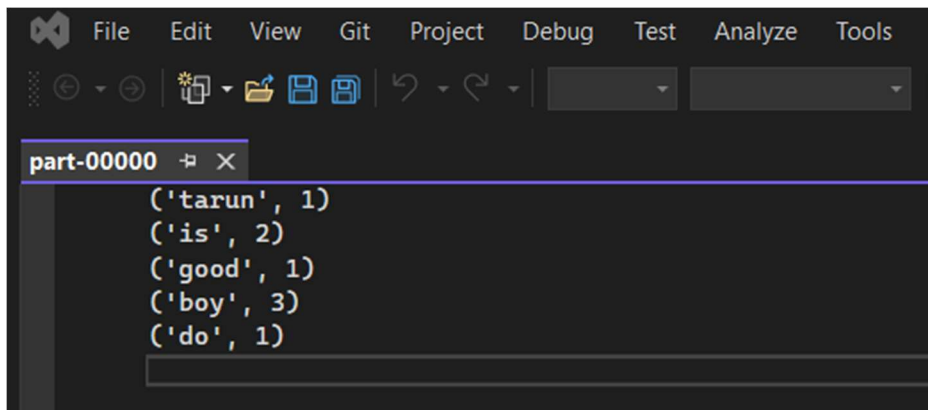
The screenshot shows the AWS Management Console interface for the Amazon S3 bucket 'wordcount2cloud'. The left sidebar contains navigation links for Amazon S3, Buckets, and wordcount2cloud. The main content area displays the bucket's details, including a table of objects. The table lists three objects: 'input.txt' (47.0 B), 'output_folder/' (0 B), and 'word_count.py' (986.0 B). The console also shows the 'Storage Lens' section and the 'Feature spotlight' for AWS Marketplace for S3.

Name	Type	Last modified	Size	Storage class
input.txt	txt	April 15, 2025, 14:06:33 (UTC-04:00)	47.0 B	Standard
output_folder/	Folder	-	-	-
word_count.py	py	April 15, 2025, 14:32:01 (UTC-04:00)	986.0 B	Standard

- ## Input files loaded in the S3 bucket

```
tarun is good boy  
ram is boy  
boy do nothing
```

- Output files from S3 bucket



The screenshot shows a code editor window with a dark theme. The title bar at the top reads "part-00000". The editor contains a JSON array of objects, each with a word and a count. The text is as follows:

```
[  
  ('tarun', 1)  
  ('is', 2)  
  ('good', 1)  
  ('boy', 3)  
  ('do', 1)  
]
```

Task 2: Docker Web Application

Objective:

Create a Docker container for a simple web application (Node.js), push to Docker Hub, and deploy on EC2.

Steps Followed:

1. Node.js App Creation (Locally):

- o Created `server.js` with Express serving "Hello, World!"
- o Added `package.json` and `Dockerfile`

2. Docker Image Build and Push:

- o Built Docker image locally:

```
docker buildx build --platform linux/amd64 -t
tarun0307/webserver:latest --push .
```

- o Verified and tested locally using Docker Desktop
- o Pushed image to Docker Hub

3. EC2 Deployment:

- o Installed Docker on EC2 instance
- o Pulled Docker image:

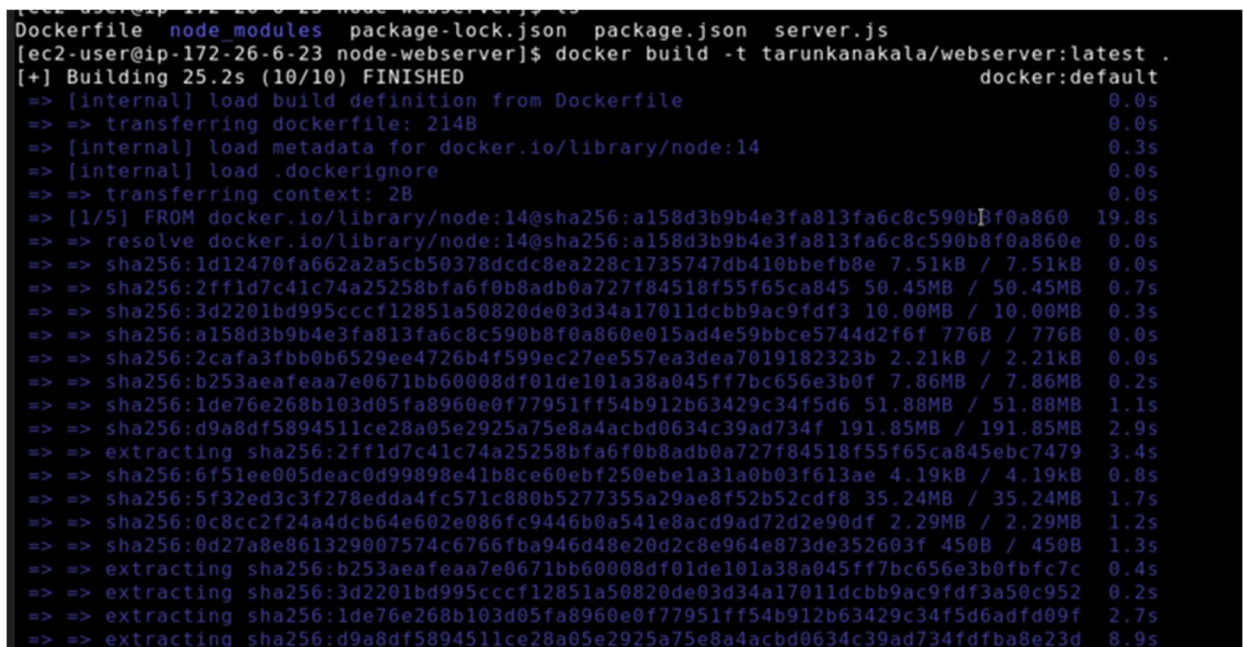
```
docker pull tarun0307/webserver:latest
```

- o Ran the container:

```
docker run -d -p 80:3000 tarun0307/webserver:latest
```

- o Opened URL: <http://18.232.162.132/> to view the running application

Screenshots Taken:



```
tarun0307@ip-172-26-6-23 node-webserver]$ docker build -t tarun0307/webserver:latest .
[+] Building 25.2s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 214B                                0.0s
=> [internal] load metadata for docker.io/library/node:14         0.3s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860 19.8s
=> => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e 0.0s
=> => sha256:1d12470fa662a2a5cb50378dcd8ea228c1735747db410bbefb8e 7.51kB / 7.51kB 0.0s
=> => sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845 50.45MB / 50.45MB 0.7s
=> => sha256:3d2201bd995ccccf12851a50820de03d34a17011dcbb9ac9fdf3 10.00MB / 10.00MB 0.3s
=> => sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6f 776B / 776B 0.0s
=> => sha256:2cfa3fbb0b6529ee4726b4f599ec27ee557ea3dea7019182323b 2.21kB / 2.21kB 0.0s
=> => sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0f 7.86MB / 7.86MB 0.2s
=> => sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6 51.88MB / 51.88MB 1.1s
=> => sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734f 191.85MB / 191.85MB 2.9s
=> => extracting sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc7479 3.4s
=> => sha256:6f51ee005deac0d99898e41b8ce60ebf250ebela31a0b03f613ae 4.19kB / 4.19kB 0.8s
=> => sha256:5f32ed3c3f278edda4fc571c880b5277355a29ae8f52b52cdf8 35.24MB / 35.24MB 1.7s
=> => sha256:0c8cc2f24a4dcb64e602e086fc9446b0a541e8acd9ad72d2e90df 2.29MB / 2.29MB 1.2s
=> => sha256:0d27a8e861329007574c6766fba946d48e20d2c8e964e873de352603f 450B / 450B 1.3s
=> => extracting sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbc7c 0.4s
=> => extracting sha256:3d2201bd995ccccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952 0.2s
=> => extracting sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f 2.7s
=> => extracting sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d 8.9s
```

```
[ec2-user@ip-172-26-6-23 node-webserver]$  
[ec2-user@ip-172-26-6-23 node-webserver]$ docker push tarun0307/webserver:latest  
The push refers to repository [docker.io/tarun0307/webserver]  
ac1b78d02e29: Pushed  
135ad34c0901: Pushed  
4cc38df274ec: Pushed  
d131e6bc2886: Pushed  
0d5f5a015e5d: Pushed  
3c777d951de2: Pushed  
f8a91dd5fc84: Pushed  
cb81227abde5: Pushed  
e01a454893a9: Pushed  
c45660adde37: Pushed  
fe0fb3ab4a0f: Pushed  
f1186e5061f2: Pushed  
b2dba7477754: Pushed  
latest: digest: sha256:fac13eba08657fc5b590b470dac42fa8a4eefeebd233f95dac56cbc2e79513bc size: 3050  
[ec2-user@ip-172-26-6-23 node-webserver]$  
[ec2-user@ip-172-26-6-23 node-webserver]$ docker pull tarun0307/webserver:latest  
latest: Pulling from tarun0307/webserver
```

- Docker Hub push confirmation
- EC2 Docker run
- Web browser showing the app running



Web Application URL : <http://0/18.232.162.132/>