

# **EC006-S04 - AWS Serverful Deployment - Application Deployment**

EC006 - AWS Cloud Services and Infrastructure

# Activity

AWS Serverful Deployment - Application Deployment

# Activity Output

```
ubuntu@ip-172-31-16-215: ~  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-16-215:~$
```

# Activity: Instructions

---

- Review the topics discussed in the previous session.
- Answer the quiz form about the AWS Serverful Deployment - Application Deployment

# Deploy Applications in an EC2 Instance

AWS Serverful Deployment - Application Deployment

# Deploy Applications in an EC2 Instance

---

Follow these steps to setup the Node.js API on the server

1. Clone the Node.js + MongoDB API project into the directory with the command **sudo git clone** <https://github.com/aaronwht/EC2>
2. In your FileZilla, you need to locate your cloned project and upload our file called App.js to our server. See sample on the next slide

# Deploy Applications in an EC2 Instance

Status: Connected to 34.222.1.54  
Status: Retrieving directory listing...  
Status: Listing directory /home/ec2-user  
Status: Directory listing of "/home/ec2-user" successful

Local site: C:\Users\USER\Downloads\EC2\ Remote site: /home/ec2-user

Local site tree:  
Downloads  
  asp.net-core-6-.net-6-project-shopping-cart-main  
  caps3  
  capstone2-babila-v2  
  cart-sample  
  complete-e-commerce-aspnet-mvc-application-master  
  DrinkAndGo-master  
  EC2

Remote site tree:  
home  
  ec2-user

| Filename   | Filesize | Filetype          | Last modified     |
|------------|----------|-------------------|-------------------|
| ..         |          |                   |                   |
| .git       |          | File folder       | 22/02/2023 1:1... |
| build      |          | File folder       | 22/02/2023 1:1... |
| .gitignore | 27       | Git Ignore Sou... | 22/02/2023 1:1... |
| a...       |          |                   | 22/02/2023 1:1... |
| n...       |          |                   | 22/02/2023 1:1... |
| p...       |          |                   | 22/02/2023 1:1... |
| p...       |          |                   | 22/02/2023 1:1... |
| R...       |          | Sou...            | 22/02/2023 1:1... |
| s...       |          | ou...             | 22/02/2023 1:1... |

Context menu options:  
Upload  
Add files to queue  
Open  
Edit  
Create directory  
Create directory and enter it  
Refresh  
Delete  
Rename

| Filename      | Filesize | Filetype    | Last modified | Permissi... | Owner/Gr...  |
|---------------|----------|-------------|---------------|-------------|--------------|
| ..            |          |             |               |             |              |
| .nvmm         |          | File folder | 22/02/2023... | drwxrwx...  | ec2-user ... |
| .ssh          |          | File folder | 22/02/2023... | drwx-----   | ec2-user ... |
| .bash_history | 466      | BASH_HI...  | 22/02/2023... | -rw-----    | ec2-user ... |
| .bash_logout  | 18       | Bash Lo...  | 15/07/2020... | -rw-r--r--  | ec2-user ... |
| .bash_profile | 193      | Bash Pro... | 15/07/2020... | -rw-r--r--  | ec2-user ... |
| .bashrc       | 428      | Bash RC ... | 22/02/2023... | -rw-r--r--  | ec2-user ... |

4 files and 2 directories. Total size: 1,105 bytes

| Size | Priority | Status |
|------|----------|--------|
|------|----------|--------|

# Deploy Applications in an EC2 Instance

| Filename ^        | Filesize | Filetype          | Last modified     |
|-------------------|----------|-------------------|-------------------|
| ..                |          |                   |                   |
| .git              |          | File folder       | 22/02/2023 1:1... |
| build             |          | File folder       | 22/02/2023 1:1... |
| .gitignore        | 27       | Git Ignore Sou... | 22/02/2023 1:1... |
| app.js            | 881      | JavaScript Sou... | 22/02/2023 1:1... |
| mongo.js          | 660      | JavaScript Sou... | 22/02/2023 1:1... |
| package-lock.json | 19,913   | JSON File         | 22/02/2023 1:1... |
| package.json      | 517      | JSON File         | 22/02/2023 1:1... |
| README.md         | 1,897    | Markdown Sou...   | 22/02/2023 1:1... |
| server.js         | 1,080    | JavaScript Sou... | 22/02/2023 1:1... |

Selected 1 file. Total size: 881 bytes

| Filename ^    | Filesize | Filetype    | Last modifi... | Permissi... | Owner/Gr...  |
|---------------|----------|-------------|----------------|-------------|--------------|
| ..            |          |             |                |             |              |
| .nvm          |          | File folder | 22/02/2023...  | drwxrwx...  | ec2-user ... |
| .ssh          |          | File folder | 22/02/2023...  | drwx-----   | ec2-user ... |
| .bash_history | 466      | BASH_HI...  | 22/02/2023...  | -rw-----    | ec2-user ... |
| .bash_logout  | 18       | Bash Lo...  | 15/07/2020...  | -rw-r--r--  | ec2-user ... |
| .bash_profile | 193      | Bash Pro... | 15/07/2020...  | -rw-r--r--  | ec2-user ... |
| .bashrc       | 428      | Bash RC ... | 22/02/2023...  | -rw-r--r--  | ec2-user ... |
| app.js        | 881      | JavaScri... | 22/02/2023...  | -rw-rw-r--  | ec2-user ... |

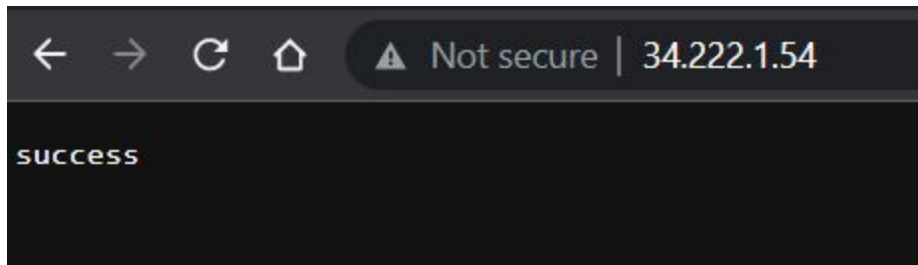
Selected 1 file. Total size: 881 bytes



# Deploy Applications in an EC2 Instance

You can try the command **ls** in your terminal and it should show the App.js file

Run the app by saying **node app** in your terminal and go to your IP address in the web browser



# Deploy Applications in an EC2 Instance

3. After testing our app, **ctrl + c** to exit our running server.

Run the following command

```
cd /etc/yum.repos.d  
sudo touch mongo-org-5.0.repo  
sudo su  
vi mongo-org-5.0.repo
```

# Deploy Applications in an EC2 Instance

Paste the code, this should be at the top

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

And **shift + esc + :** and type **wq!**

# Deploy Applications in an EC2 Instance

```
ec2-user@ip-172-31-19-57:/etc/yum.repos.d
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
~
~
~
~
~
~
~
~
:wq!
```

# Deploy Applications in an EC2 Instance

4. Run this command **cat mongo-org-5.0.repo** to view the file

```
[root@ip-172-31-19-57 yum.repos.d]# [root@ip-172-31-19-57 yum.repos.d]# cat mongo-org-5.0.repo
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

5. Run the command

**Windows** : **yum install -y mongodb-org**

**MAC** : **sudo yum install -y mongodb-org**

# Deploy Applications in an EC2 Instance

6. Configure our simple react application.

Go to your build folder directory and install our with the below command:

**`npx create-react-app .`**

This command will install our react inside our build folder

If you're creating a project from scratch you can do something like:

**`npx create-react-app your-app-name`**

***Note:*** You have to move your `index.html` outside build folder before setting up your react project and we will use it for later.

# Deploy Applications in an EC2 Instance

7. Optional: Remove unnecessary files and boilerplate

Go to **src** folder delete these files: **index.css**, **reportWebVitals.js**

And on **App.js** remove the whole div in your return function.

Remove items in **index.js** the line

```
import './index.css';
```

```
import reportWebVitals from './reportWebVitals';
```

```
reportWebVitals();
```

# Deploy Applications in an EC2 Instance

Add a message and try to run with the command **npm start**



Hello World



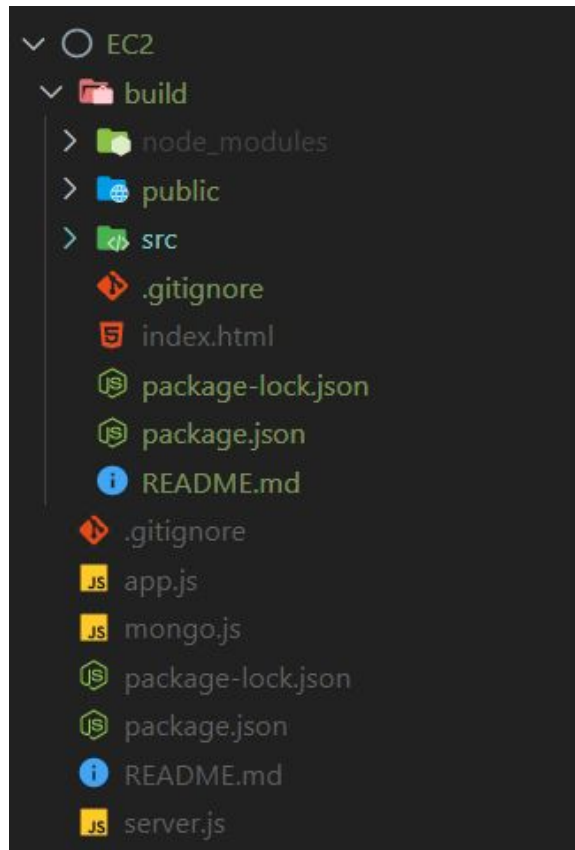
# Deploy Applications in an EC2 Instance

8. Move your index.html to your react app

And delete the folder called

**node\_modules**

For now we just assume that we have a react app but what we're doing is accessing the file index.html. It's gonna be a different implementation if you're retrieving these data inside your react application.



# Deploy Applications in an EC2 Instance

9. Upload the following files to our server make sure you're on ec2-user directory

| Filename ^        | Filesize | Filetype          | Last modified     |
|-------------------|----------|-------------------|-------------------|
| ..                |          |                   |                   |
| .git              |          | File folder       | 22/02/2023 1:1... |
| build             |          | File folder       | 22/02/2023 1:1... |
| .gitignore        | 27       | Git Ignore Sou... | 22/02/2023 1:1... |
| app.js            | 881      | JavaScript Sou... | 22/02/2023 1:1... |
| mongo.js          | 660      | JavaScript Sou... | 22/02/2023 1:1... |
| package-lock.json | 19,913   | JSON File         | 22/02/2023 1:1... |
| package.json      | 517      | JSON File         | 22/02/2023 1:1... |
| README.md         | 1,897    | Markdown Sou...   | 22/02/2023 1:1... |
| server.js         | 1,080    | JavaScript Sou... | 22/02/2023 1:1... |

| Filename ^    | Filesize | Filetype    | Last modifi... | Permissi... | Owner/Gr..   |
|---------------|----------|-------------|----------------|-------------|--------------|
| .ssh          |          | File folder | 22/02/2023...  | drwx-----   | ec2-user ... |
| build         |          | File folder | 22/02/2023...  | drwxrwx...  | ec2-user ... |
| .bash_history | 2,818    | BASH_HI...  | 22/02/2023...  | -rw-----    | ec2-user ... |
| .bash_logout  | 18       | Bash Lo...  | 15/07/2020...  | -rw-r--r--  | ec2-user ... |
| .bash_profile | 193      | Bash Pro... | 15/07/2020...  | -rw-r--r--  | ec2-user ... |
| .bashrc       | 428      | Bash RC ... | 22/02/2023...  | -rw-r--r--  | ec2-user ... |
| .viminfo      | 4,999    | VIMINFO...  | 22/02/2023...  | -rw-----    | ec2-user ... |
| app.js        | 881      | JavaScri... | 22/02/2023...  | -rw-rw-r--  | ec2-user ... |
| mongo.js      | 660      | JavaScri... | 22/02/2023...  | -rw-rw-r--  | ec2-user ... |
| server.js     | 1,080    | JavaScri... | 22/02/2023...  | -rw-rw-r--  | ec2-user ... |

# Deploy Applications in an EC2 Instance

10. Let's create folders we need

**cd /** (going back to root directory)

**mkdir data > cd data**

**mkdir db > cd /home/ec2-user**

**ls** (you should have app.js build mongo.js server.js)

**service mongod start**

```
[root@ip-172-31-19-57 ec2-user]# service mongod start
Redirecting to /bin/systemctl start mongod.service
```

# Deploy Applications in an EC2 Instance

## 11. Install mongoose and express

Go back to the ec2 directory, just to make sure you can run below command:

**exit** & **exit** to disconnect and connect again to your instance

**npm install mongoose express**

**If** you get node: /lib64/...`GLIBC\_2.XX` not found error

Run command **nvm install 16** to make sure you're using that version

And type again **npm install mongoose express**

# Deploy Applications in an EC2 Instance

12. Install **npm install pm2 -g** as this will restart our node service if anything might happen.

13. Let's enter mongo shell console by typing **mongo**. Type the following command to access our database, configure user and insert records.

**use mern**

```
db.createUser({ user: "my_user", pwd: "my_pwd", roles: ["dbOwner"] })
```

```
db.members.insert({ firstName: "Bill", lastName: "Smith" })
```

```
db.members.insert({ firstName: "Bob", lastName: "Smith" })
```

# Test Applications in EC2 Instances

# Test Applications in EC2 Instances

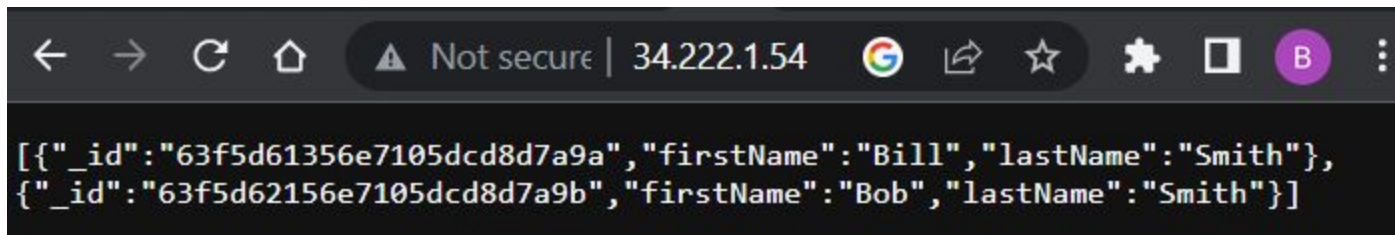
---

Enter the hostname of your AWS EC2 instance in a browser to access and test your new MERN stack application.

The hostname is the "Public DNS (IPv4)" property located on the instance description tab in the AWS Console.

# Test Applications in EC2 Instances

1. After inserting record in our database, **exit** the shell and run the command **node mongo** and go to your web browser and enter your IPv4 address.



```
[{"_id": "63f5d61356e7105dcd8d7a9a", "firstName": "Bill", "lastName": "Smith"}, {"_id": "63f5d62156e7105dcd8d7a9b", "firstName": "Bob", "lastName": "Smith"}]
```



# Test Applications in EC2 Instances

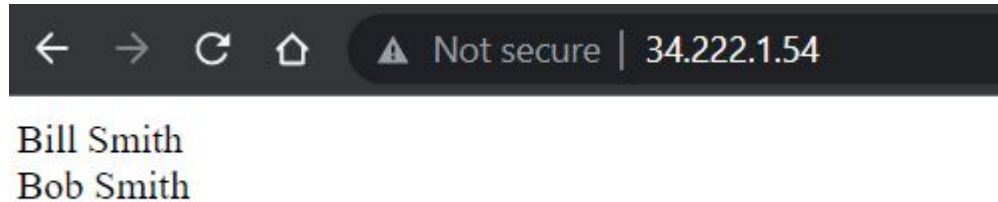
2. Let's run our server.js with this command **pm2 start server.js**

```
[ec2-user@ip-172-31-19-57 ~]$ pm2 start server.js  
[PM2] Starting /home/ec2-user/server.js in fork_mode (1 instance)  
[PM2] Done.
```

| id | name   | namespace | version | mode | pid   | uptime | 🔄 | status | cpu | mem    | user     | watching |
|----|--------|-----------|---------|------|-------|--------|---|--------|-----|--------|----------|----------|
| 0  | server | default   | N/A     | fork | 19559 | 0s     | 0 | online | 0%  | 25.7mb | ec2-user | disabled |

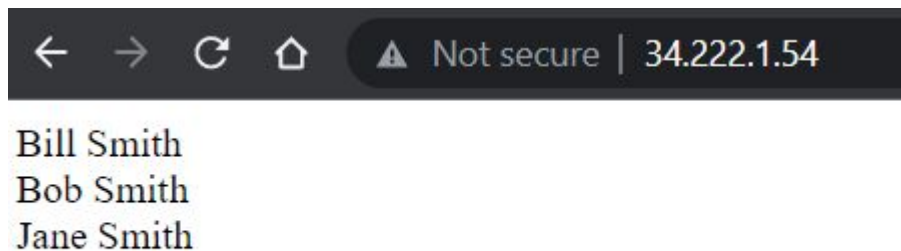
# Test Applications in EC2 Instances

Refresh your web browser and you should see the inserted records.



# Test Applications in EC2 Instances

3. Exit and connect, go to mongo shell, use db mern, add another record and refresh your browser.



# Quiz Form

Tools to Interact with AWS Services

Click on this [link](#) to navigate to the quiz form for the session on Introduction to AWS and EC2.

Make sure to **tick the checkbox and record your email with your response.**