Welcome to your new pre-configured MERN stack server!

This AMI includes everything you need to start developing with MongoDB or MySQL, Express, React, and Node.js (MERN).

You'll also have VS Code Server for remote development and Nginx as a reverse proxy.

Usage instructions:

1. Launch the product via 1-click. **Please wait until** the instance passes **all** status checks and is running. You can connect using your Amazon private key and '**ubuntu**' login via your SSH client.

To update software, use:

```
sudo apt update && sudo apt upgrade -y
```

Suggested Steps to Get Start!

Step 1: Enable and Configure Nginx

1. Open the Nginx configuration file:

   ```
   sudo nano /etc/nginx/sites-available/mern
   ```

2. Find the line that says: “server_name” your-server-ip;

   Replace your-server-ip with your actual **Instance Public IP Address.**

3. Save and exit (CTRL+X, then Y, then ENTER).

4. Restart Nginx to apply changes:

   ```
   sudo systemctl restart nginx
   ```

◆ Step 2: Choose Your Database (MongoDB or MySQL)

1. Navigate to the backend directory:

   ```
   cd /opt/mern/backend
   ```

2. Edit the **environment file**:

```
nano .env
```

3. **Choose Your Database**:

   - **To Use MongoDB**:
     ```
     USE_MONGO=true
     USE_MYSQL=false
     ```

   - **To Use MySQL**:

     ```
     USE_MONGO=false
     USE_MYSQL=true
     ```

4. **Set Your Own Password** in .env under:

   ```
   MYSQL_USER=root
   MYSQL_PASSWORD=yourpassword
   MYSQL_DATABASE=mern_db
   ```

5. **Save and exit** (CTRL+X, then Y, then ENTER).

📂 Step 3: Understand Your MERN Directories

The project is structured as follows:

```
cd /opt/mern
ls -la
```

You will see:

```
backend/   # Express.js Backend (API Server)
frontend/  # React.js Frontend
```

- **Backend:** Contains API logic, Express routes, and database connections.
- **Frontend:** Contains React UI.

🖥️ Step 4: Start React & Express

Start the backend and frontend using **PM2**:

```
pm2 start /opt/mern/backend/index.js --name backend
pm2 start "npm start" --name frontend --cwd /opt/mern/frontend
pm2 save
```

Verify that both are running:

```
pm2 list
```

🌐 Step 5: Test Your Backend

In your browser, go to:

http://your-public-ip/api

Ex: http://36.323.32/api

Expected Output: "MERN Backend Running"

If the API does not respond, check logs:

```
pm2 logs backend
```

📝 Step 6: Access VS Code Server (Remote Code Editing)

- Find your **VS Code login password**:

  ```
  nano ~/.config/code-server/config.yaml
  ```

- Restart VS Code Server if needed or if you change the password

  ```
  sudo systemctl restart code-server@$USER
  ```

- Open in browser:

  http://your-public-ip:8085

  Ex: http://36.323.32:8085

📡 Step 7: Check Running Services

Use these commands to check if everything is running:

```
pm2 list                                   # Check backend & frontend status
sudo systemctl status nginx                 # Check Nginx
sudo systemctl status mongod               # Check MongoDB
sudo systemctl status mysql                # Check MySQL (if used)
sudo systemctl status code-server@$USER    # Check VS Code Server
```

💻 Step 8: Start Coding!

1. Navigate to the correct directory:

   ```
   cd /opt/mern/frontend        # Edit React files
   cd /opt/mern/backend         # Edit Express API files
   ```

2. Open **VS Code Server** and start coding remotely:

   http://your-public-ip:8085

📖 Additional Resources

- Learn **React.js**: https://react.dev/
- Learn **Express.js**: https://expressjs.com/
- Learn **MongoDB**: https://www.mongodb.com/docs/
- Learn **MySQL**: https://dev.mysql.com/doc/
- Learn **PM2 Process Manager**: https://pm2.keymetrics.io/

🎉 Congratulations!

Your MERN Stack development server is now fully configured and ready for use. 🚀
Happy coding! 🎨💻

AWS Data

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized_keys & /home/ubuntu/.ssh/authorized_keys
- Monitor the health:
  - Navigate to your Amazon EC2 console and verify that you're in the correct region.
  - Choose Instance and select your launched instance.
  - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

Extra Information: (Optional)

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html