

CLOUD COMPUTING (CS G527)

MINI PROJECT REPORT

GROUP – 5

ID	Name	Email
2022H1030067H	S Shashank	h20221030067@hyderabad.bits-pilani.ac.in
2022H1030099H	Ajinkya Medhekar	h20221030099@hyderabad.bits-pilani.ac.in
2022H1030074H	Utsav Seth	h20221030074@hyderabad.bits-pilani.ac.in

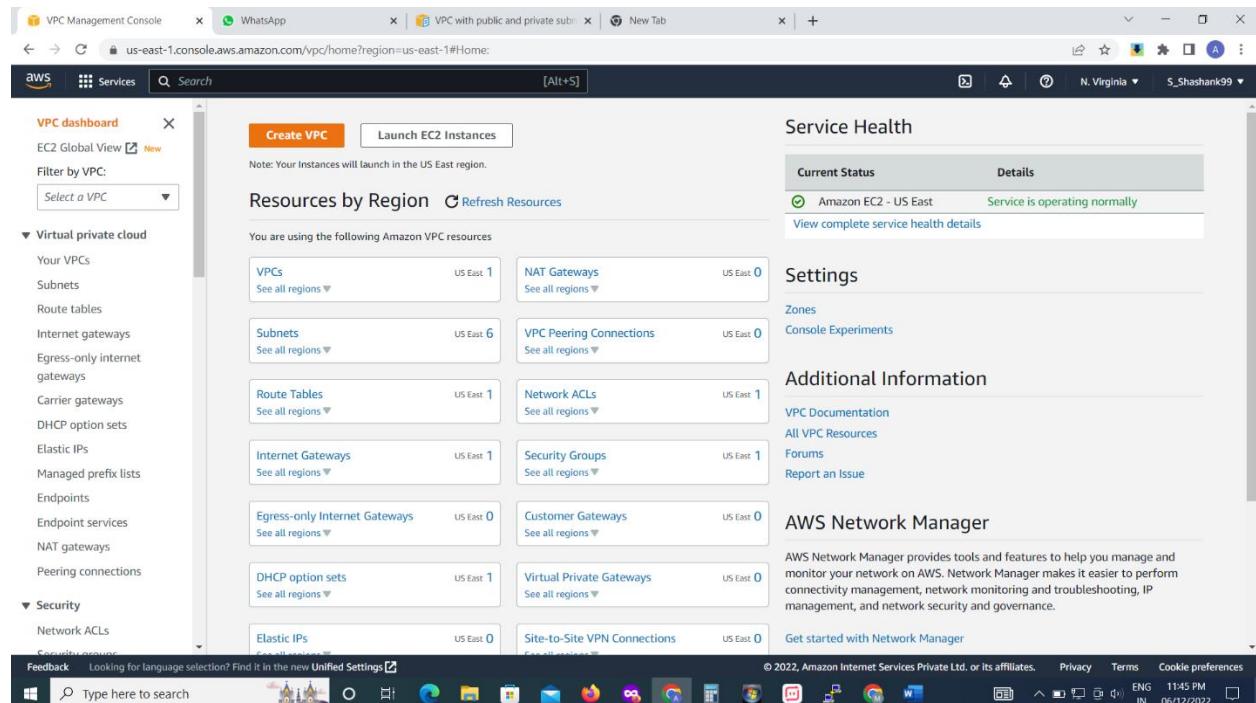
Phase 1:

We created a web application using MERN stack with React acting as front-end and MongoDB Atlas as database server.

Phase 2:

Step 1: Creating VPC

- Login to AWS console. Go to VPC console and select Create VPC.



- Provide VPC name to your new VPC. Select VPC and more option and select the Auto-generate option and click on Create to create your VPC.

- Created VPC can be viewed in Your VPCs tab.

The screenshot shows the AWS VPC Management Console. On the left, there's a navigation sidebar with options like 'VPC dashboard', 'EC2 Global View', 'Virtual private cloud', 'Security', and 'Feedback'. The main area displays 'Your VPCs (1/2)' with a table. One row is selected for 'project-vpc'. The table columns include Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, and DHCP option set. Below the table, there's a detailed view for 'vpc-05c95f98e05c88ac6 / project-vpc' with tabs for Details, CIDRs, Flow logs, and Tags. The 'Details' tab is active, showing information such as VPC ID, State (Available), and DNS resolution.

Step 2: Create Security Groups for your VPC which will be used later for your Web Server instance and Database Server instance and enable auto-assign public IPv4 address for the public subnet you will use for the Web Server from the VPC created.

- Under VPC console scroll and go to Security Groups tab and create New Security Group for your Web Server EC2 instance and add appropriate Inbound Rules.

The screenshot shows the AWS EC2 Management Console. The top navigation bar includes tabs for EC2 Manager, WhatsApp, Google Cloud, Install and Deploy, Install Monitor, Configuration, Deployments, How to do it, and node.js. The main content area is titled 'Inbound rules' and lists several security group rule entries. Each entry includes fields for Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. The 'Delete' button is visible next to each rule. At the bottom of the list, there's a 'Add rule' button.

- Under VPC console scroll and go to Security Groups tab and create New Security Group for your Database Server EC2 instance and add appropriate Inbound Rules. For source add the security group created for Web Server.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0397aa0a183bfde5b	All traffic	All	All	Custom	<input type="text" value="sg-02b8efd8cc02984e3"/> X

[Add rule](#) [Cancel](#) [Preview changes](#) [Save rules](#)

- Go to Subnets tab in VPC console and select the public subnet in which you will create your Web Server. Go to Actions and select Edit Subnet settings.

You have successfully changed subnet settings:

- Enable auto-assign public IPv4 address

Actions	Create subnet
View details	
Create flow log	
Edit subnet settings	
Edit IPv6 CIDRs	
Edit network ACL association	
Edit route table association	
Edit CIDR reservations	
Share subnet	
Manage tags	
Delete subnet	

- Select the Enable auto-assign public IPv4 address and save changes.

Subnet

Subnet ID	Name
subnet-004b24d9991559c84	project-subnet-public2-us-east-1b

Auto-assign IP settings

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

Enable auto-assign public IPv4 address

Enable auto-assign customer-owned IPv4 address

Resource-based name (RBN) settings

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

Enable resource name DNS A record on launch

Enable resource name DNS AAAA record on launch

Step 3: Create Internet Gateway (IGW) for your VPC from the respective tab.

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A table lists two entries:

Name	Internet gateway ID	State	VPC ID	Owner
project-igw	igw-04b4ecc50f407d10	Attached	vpc-05c95f98e05c88ac6 project-vpc	489471630092
-	igw-0ad060a05e11a6476	Attached	vpc-0baf5294eed68d74e	489471630092

Below the table, a detailed view for 'igw-04b4ecc50f407d10 / project-igw' is shown, including its details and state.

Step 4: Configure Route Tables

- Go to Route Tables tab and select the public route table for your VPC and select Edit routes.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A table lists several route tables:

Name	Route table ID	Explicit subnet associations	Edge associations
project-rtb-private2-us-east-1b	rtb-00336f65070402206	subnet-012139779679...	-
-	rtb-0ba48b08c06536cb	-	-
project-rtb-private1-us-east-1a	rtb-0834b8a07886d311c	subnet-0150ac36ce895...	-
-	rtb-03b4579fae6d921b2	-	-
project-rtb-public	rtb-036bd9097218e076a	2 subnets	-

A context menu is open over the 'project-rtb-public' row, listing options like 'View details', 'Edit subnet associations', 'Edit edge associations', etc.

- Add destination as 0.0.0.0/0 and target as your IGW created for VPC and click on Save changes.

VPC > Route tables > rtb-036bd9097218e076a > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	Q local	Active	No
Q 0.0.0.0/0	Q igw-04b4ecc50f407d10	Active	No

Add route

Cancel Preview Save changes

Step 5: Create public EC2 instance for your Web Server

- Go to EC2 dashboard from your AWS console and click on Launch Instance to create your Web Server EC2 instance.

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S]

New EC2 Experience Tell us what you think

EC2 Dashboard

- EC2 Global View
- Events
- Tags
- Limits
- Instances
 - Instances New
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances New
 - Dedicated Hosts
 - Scheduled Instances
 - Capacity Reservations
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	0	Load balancers	0
Placement groups	0	Security groups	4	Snapshots	0
Volumes	0				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Account attributes

- Supported platforms
 - VPC
 - Default VPC
 - vpc-0ba5f294eed68d74e
- Settings
 - EBS encryption
 - Zones
 - EC2 Serial Console
 - Default credit specification
 - Console experiments

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

AWS Health Dashboard

Region: US East (N. Virginia)

Status: This service is operating normally

Explore AWS

Amazon GuardDuty Malware Protection

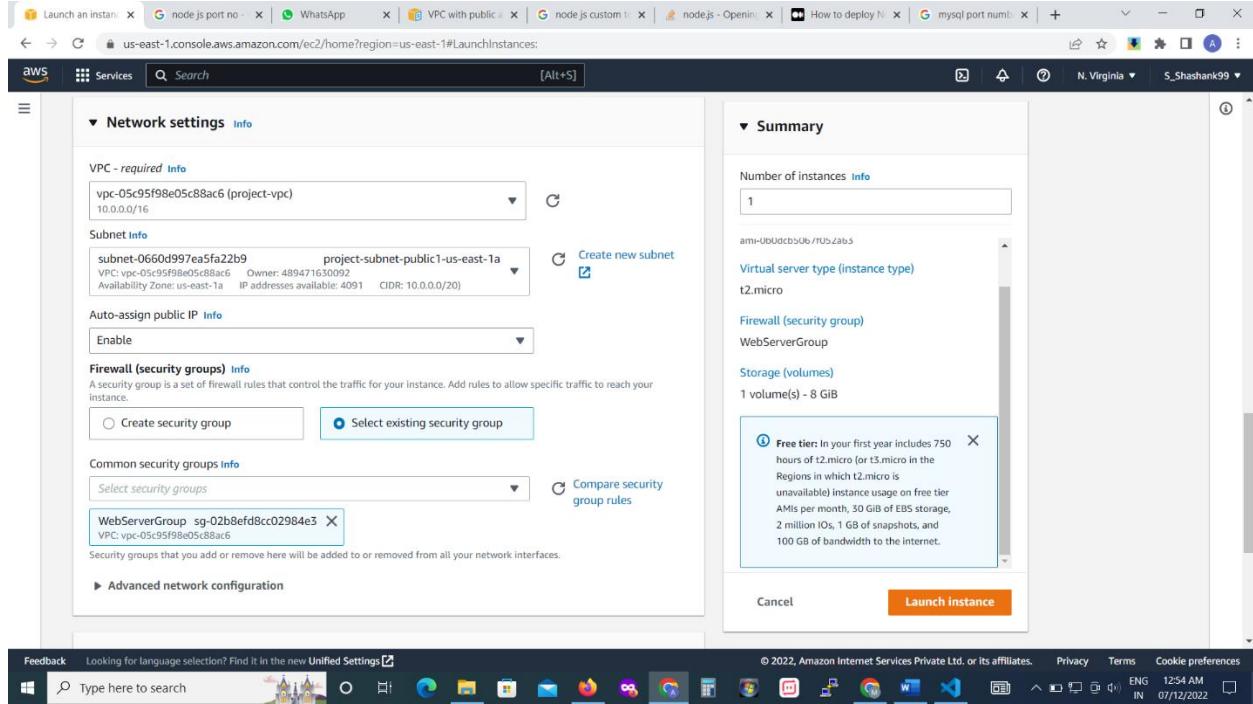
Enable Best Price-Performance with AWS Graviton2

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

- Select Amazon Linux 2 AMI and instance type t2.micro.
- Create a key pair for your Web Server EC2 instance.

- Edit Network Settings and add created VPC information and select existing Security Groups option and select the Security Group created for your Web Server in your VPC. Enable Auto-assign public IP.
- Click on Launch Instance to create the Web Server EC2 instance.



Step 6: Create private EC2 instance for your Database Server

- Go to EC2 dashboard from your AWS console and click on Launch Instance to create your Web Server EC2 instance.

The screenshot shows the AWS EC2 Management Console dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays 'Resources' and 'Account attributes'. Under 'Resources', it shows 1 instance (running), 1 key pair, 0 placement groups, 4 security groups, and 1 volume. The 'Account attributes' section includes links for Supported platforms (VPC), Default VPC (vpc-0baf5294eed68d74e), Settings, EBS encryption, Zones, EC2 Serial Console, Default credit specification, and Console experiments.

- Select Amazon Linux 2 AMI and instance type t2.micro.
- Create a key pair for your Database Server EC2 instance.
- Edit Network Settings and add created VPC information and select existing Security Groups option and select the Security Group created for your Database Server in your VPC. Disable Auto-assign public IP to create a private EC2 instance.
- Click on Launch Instance to create the Database Server EC2 instance.

The screenshot shows the 'Launch instance' wizard. Step 1: Network settings. It requires a VPC (selected: project-vpc) and a subnet (selected: project-subnet-private1-us-east-1a). The 'Auto-assign public IP' option is disabled. Under Firewall (security groups), the 'Select existing security group' option is selected, and a dropdown shows 'DatabaseServerGroup sg-0f807632ef2c08e2a'. A note states: 'A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.' Step 2: Summary. It shows 1 instance, the selected AMI (Amazon Linux 2 Kernel 5.10 AMI...), instance type t2.micro, and a note about the Free tier. Buttons for 'Cancel' and 'Launch instance' are at the bottom.

Step 7: Create NAT gateway in public subnet to connect public EC2 instance to private EC2 instance in private subnet.

The screenshot shows the AWS VPC dashboard with the 'NAT gateways' section selected. On the right, there is a large orange button labeled 'Create NAT gateway'. The left sidebar lists various VPC-related services like EC2 Global View, Subnets, Route tables, and NAT gateways.

- Give a name for your NAT Gateway and select the public subnet in which public EC2 instance was created earlier.
- Assign Elastic IP address and add appropriate tags.

The screenshot shows the 'Create NAT gateway' wizard. Step 1: 'NAT gateway settings'. It asks for a name ('nat-gateway'), selects a public subnet ('subnet-0660d997ea5fa22b9 (project-subnet-public1-us-east-1a)'), and sets the connectivity type to 'Public'. An elastic IP ('eipalloc-068699455d8b207b2') is assigned. Step 2: 'Tags', where a single tag 'nat-gateway' is added. The status bar at the bottom indicates the process is 54% complete.

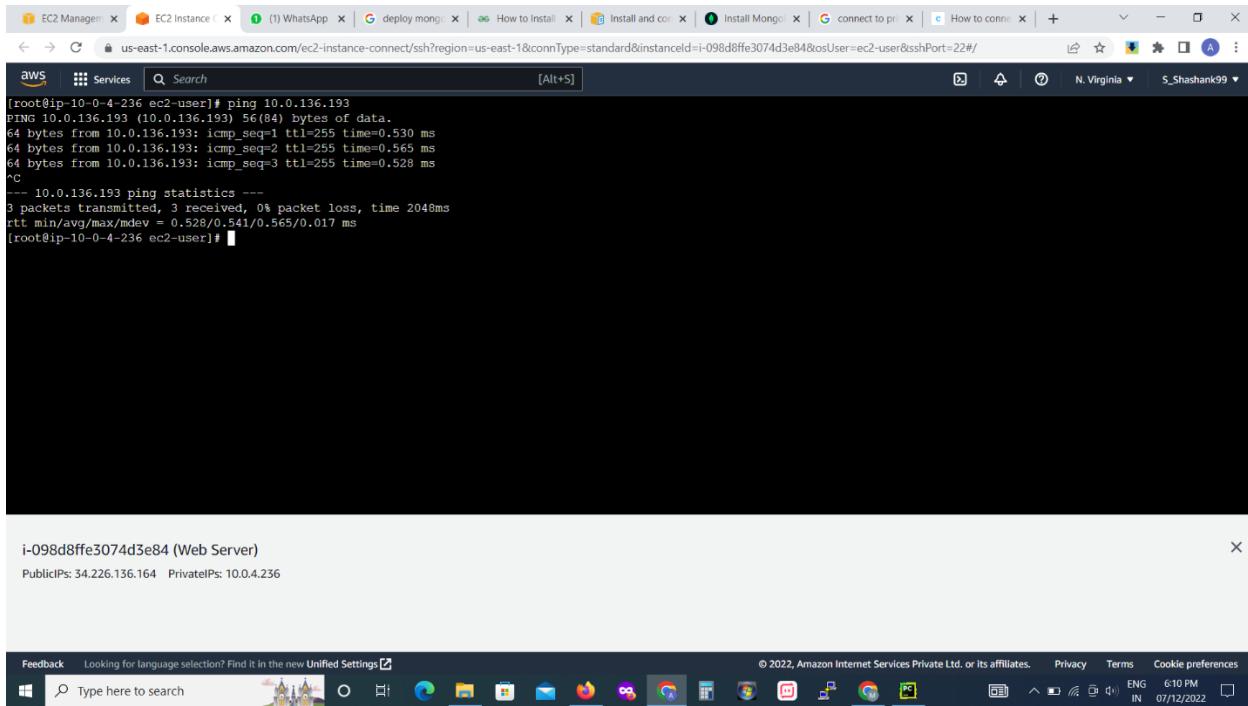
The screenshot shows the AWS VPC Management console. On the left, there's a navigation sidebar with sections like 'Your VPCs', 'Subnets', 'Route tables', etc. The main content area displays a success message: 'NAT gateway nat-03a5384dc01eeef94 | nat-gateway was created successfully.' Below this, it shows the details of the newly created NAT gateway, including its ID, ARN, connectivity type (Public), state (Pending), and subnet information. The 'Monitoring' tab is selected at the bottom.

- Configure private Route table for the NAT Gateway by adding destination as 0.0.0.0/0 and target as your NAT Gateway.

The screenshot shows the 'Edit routes' page for a specific route table. It lists two existing routes: one for destination pl-63a5400a targeting vpce-0fc4d6e2cfcd05e9a, and another for destination 10.0.0.0/16 targeting 'local'. A new route is being added for destination 0.0.0.0/0, with the target set to nat-03a5384dc01eeef94. The 'Save changes' button is visible at the bottom right.

Step 8: Connect to Web Server EC2 instance and install default packages.

Step 9: Check connectivity to private Database Server EC2 instance.



Step 10: Connect to your private Database Server EC2 instance.

- Create and save private key of Database Server.(Here file name given is db-server.pem)
 - nano db-server.pem
- Change permissions of the private key file
 - chmod 400 db-server.pem
- Use SSH to connect to the private server
 - ssh ec2-user@<private_ip of db-server> -i db-server.pem
- This will connect your public EC2 web server to your private Database Server.

Step 11: Install MongoDB on your Database Server

- Refer official MongoDB site for installation steps
<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-amazon/>
- Configure the package management system(yum).
 - sudo nano /etc/yum.repos.d/mongodb-org-6.0.repo
- Add below lines to the repo file and save it.

[mongodb-org-6.0]

name=MongoDB Repository

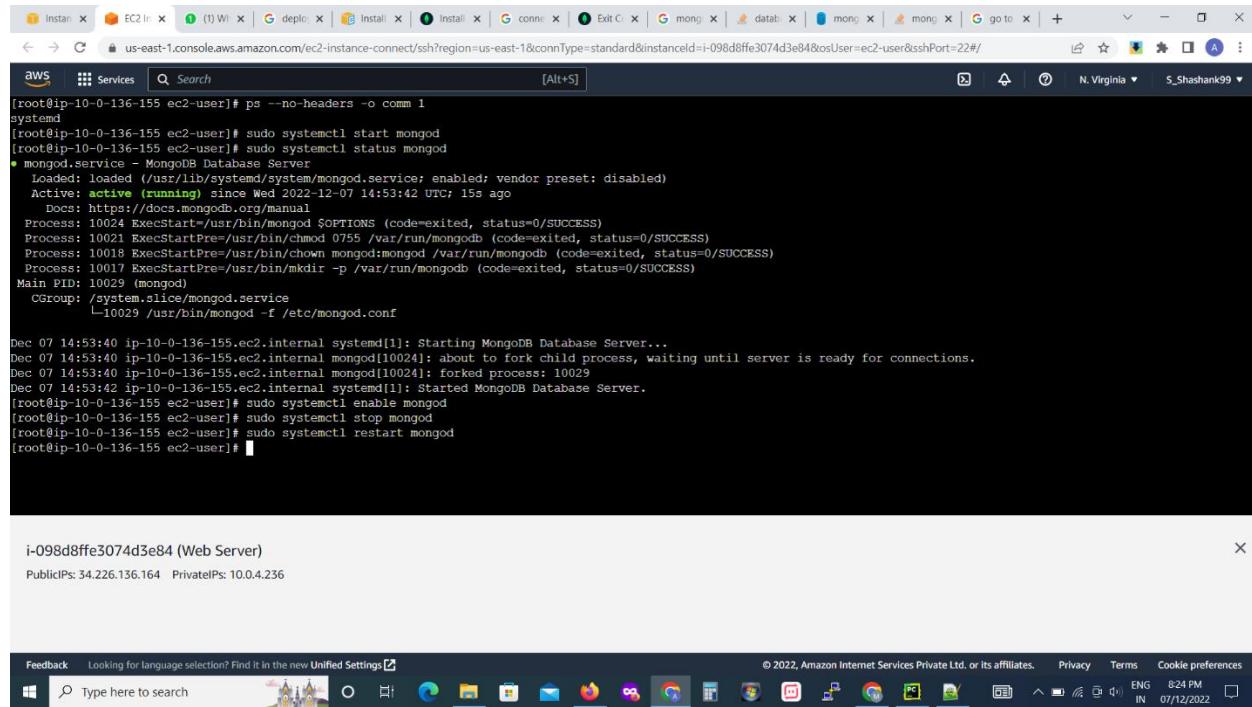
baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/6.0/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-6.0.asc

- Execute the following command to install latest stable version of MongoDB
 - sudo yum install -y mongodb-org
- Check which init system your platform uses and proceed further accordingly.
- Execute following command to check init system
 - ps --no-headers -o comm 1
- It returned systemd so follow systemd related further steps.
- To start MongoDB execute following command
 - sudo systemctl start mongod
- This will start MongoDB to verify whether MongoDB has started successfully execute following command
 - sudo systemctl status mongod
- To ensure MongoDB will start following a system reboot run following command
 - sudo systemctl enable mongod
- You can now begin using MongoDB by using mongosh command.



The screenshot shows a terminal session on an AWS EC2 instance. The user is root and is performing the following commands:

```

[root@ip-10-0-136-155 ec2-user]# ps --no-headers -o comm 1
systemd
[root@ip-10-0-136-155 ec2-user]# sudo systemctl start mongod
[root@ip-10-0-136-155 ec2-user]# sudo systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-12-07 14:53:42 UTC; 15s ago
     Docs: https://docs.mongodb.org/manual
  Process: 10024 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 10021 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 10018 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 10017 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 10029 (mongod)
   Group: /system.slice/mongod.service
         └─10029 /usr/bin/mongod -f /etc/mongod.conf

Dec 07 14:53:40 ip-10-0-136-155.ec2.internal systemd[1]: Starting MongoDB Database Server...
Dec 07 14:53:40 ip-10-0-136-155.ec2.internal mongod[10024]: about to fork child process, waiting until server is ready for connections.
Dec 07 14:53:40 ip-10-0-136-155.ec2.internal mongod[10024]: forked process: 10029
Dec 07 14:53:42 ip-10-0-136-155.ec2.internal Systemd[1]: Started MongoDB Database Server.
[root@ip-10-0-136-155 ec2-user]# sudo systemctl enable mongod
[root@ip-10-0-136-155 ec2-user]# sudo systemctl stop mongod
[root@ip-10-0-136-155 ec2-user]# sudo systemctl restart mongod
[root@ip-10-0-136-155 ec2-user]#

```

Below the terminal, a tooltip displays the instance details:

i-098d8ffe3074d3e84 (Web Server)
PublicIPs: 34.226.136.164 PrivateIPs: 10.0.4.236

At the bottom, the Windows taskbar shows various open applications including a browser window for the AWS console.

Step 12: Configure bindIp in the mongod.conf file. (Encountered problems)

- By default MongoDB server mongod only accepts loopback connections from IP address 127.0.0.1 (localhost). To allow connection from anywhere else in your VPC we need to edit the bindIp value.
- Run the following command to access the mongod.conf file
 - sudo nano /etc/mongod.conf
- Edit the mongod.conf file and look for following lines

```
# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1 # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.
```

- Modify the bindIp line so it looks like the following (Edit according to solution provided following line will give error as mongod.service failed)

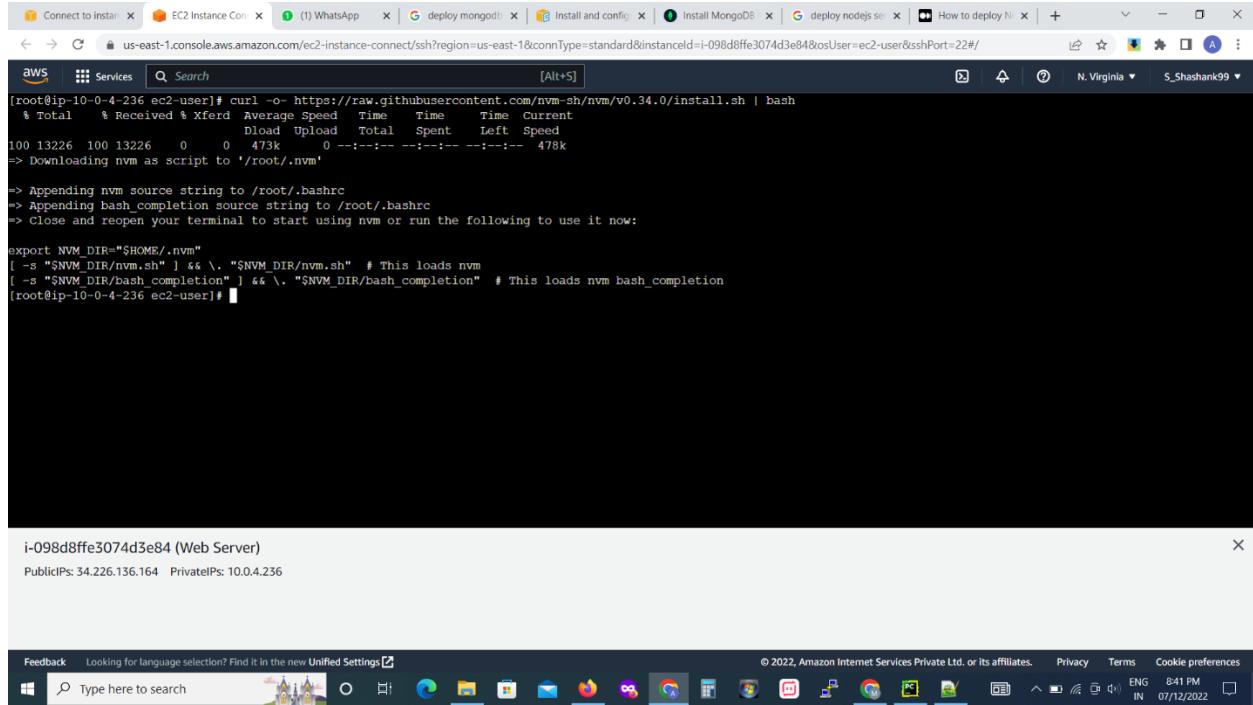
bindIp: public-dns-name

- Replace public-dns-name with actual public DNS name
- Save the mongod.conf file and restart mongod.
 - sudo service mongod restart
- After executing above line mongod.service failed message was displayed with an exit code referring to an un-recoverable error. So we had to terminate the private EC2 instance and recreate the instance by following same steps.
- After recreating the private EC2 Database Server and installing MongoDB the bindIp field should be edited as follows:
 - Following is the solution for bindIp problem. Maximum editing of this value in the mongod.conf file is 1(one).
 - bindIp: <private_ip of web server>
- Save the mongod.conf file and restart mongod.
 - sudo service mongod restart
- The modification is done successfully and mongod.service won't fail. If failure occurred terminate current instance again and repeat the installation steps with editing the bindIp in same way as shown in solution.
- After successfully restarting mongod you can execute MongoDB queries from your Web Server.

Step 13: Installing NodeJS and required dependencies on Web Server.

- Connect to your public EC2 Web Server instance.
- To install node version manager (nvm) run following command
 - curl -o- <https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh> | bash

- Activate nvm by typing the following command
 - . ~/.nvm/nvm.sh
- Use nvm to install latest stable version of Node.js
 - nvm install 16
- To test that node and npm are properly installed run following commands.
 - node -v or node --version
 - npm -v or npm --version



The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several pinned icons: Connect to instance, EC2 Instance Con..., WhatsApp, deploy mongoDB, Install and config, Install MongoDB, deploy nodejs serv, How to deploy N..., and a search bar. Below the taskbar is a system tray with icons for battery, signal, volume, and date/time (8:41 PM, 07/12/2022). The main area of the screen is a terminal window titled 'aws' with the title bar 'Services'. The terminal content is as follows:

```
[root@ip-10-0-4-236 ec2-user]# curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100 13226  100 13226    0     0  473k  0:--:--:--:--:--:--:--: 478k
--> Downloading nvm as script to '/root/.nvm'

--> Appending nvm source string to /root/.bashrc
--> Appending bash completion source string to /root/.bashrc
--> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[root@ip-10-0-4-236 ec2-user]#
```

Below the terminal window, a tooltip displays the instance information: i-098d8ffe3074d3e84 (Web Server) and Public IPs: 34.226.136.164 Private IPs: 10.0.4.236.

```
Connect to instance | EC2 Instance Connect | WhatsApp | deploy mongoDB | Install and config | Install MongoDB | deploy node.js | How to deploy Node.js | + N. Virginia | S_Shashank99 | ...
```

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-098d8ffe3074d3e84&osUser=ec2-user&sshPort=22/#

aws Services Search [Alt+S]

```
[root@ip-10-0-4-236 ec2-user]# curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
100 13226 100 13226 0 0 473k 0 --:--:--:--:--:-- 478k
=> Downloading nvm as script to '/root/.nvm'.
```

=> Appending nvm source string to /root/.bashrc
=> Appending bash_completion source string to /root/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && . "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[root@ip-10-0-4-236 ec2-user]# . ~/.nvm/nvm.sh
[root@ip-10-0-4-236 ec2-user]# nvm install node
Downloading and installing node v19.2.0...
Downloaded https://nodejs.org/dist/v19.2.0/node-v19.2.0-linux-x64.tar.xz... 100.0%
Computing checksum with sha256sum
Checksums matched!
node: /lib64/libm.so.6: version 'GLIBC_2.27' not found (required by node)
node: /lib64/libc.so.6: version 'GLIBC_2.28' not found (required by node)
nvm is not compatible with the npm config "prefix" option: currently set to ""
Run `nvm use --delete-prefix v19.2.0` to unset it.
[root@ip-10-0-4-236 ec2-user]#
```

i-098d8ffe3074d3e84 (Web Server)

Public IPs: 34.226.136.164 Private IPs: 10.0.4.236

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

ENG 8:42 PM IN 07/12/2022

Step 14: Installing Git and clone repository from GitHub

- To install git run following command
 - sudo yum install git -y
 - To verify if git has been properly installed run following command
 - git –version
 - To clone the code repository from GitHub run following command
 - git clone <https://github.com/Aji925/project-healer.git>
 - This will create a folder with name project-healer which will contain your code files.

Step 15: Installing required dependencies for the Node.js application cloned from GitHub.

- To install node dependencies and any other project dependencies run following set of commands.
 - cd project-healer
 - npm install 16
 - npm install
 - npm run client-install
 - npm run server-install
 - This will install all the required dependencies.

```
[root@ip-10-0-4-236 project-healer]# npm install
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile

added 86 packages, and audited 87 packages in 7s

9 vulnerabilities (5 moderate, 3 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

npm notice
npm notice New major version of npm available! 8.19.2 -> 9.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.2.0
npm notice Run npm install -g npm@9.2.0 to update!
npm notice

[root@ip-10-0-4-236 project-healer]#
```

i-098d8ffe3074d3e84 (Web Server)

PublicIPs: 34.226.136.164 PrivateIPs: 10.0.4.236



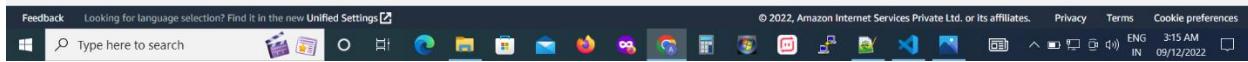
```
[root@ip-10-0-4-236 project-healer]# npm run client-install
> project-healer@1.0.0 client-install
> cd client && npm install

npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile

npm deprecated w3c-hr-time@1.0.1: Use your platform's native performance.now() and performance.timeOrigin.
npm deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm deprecated stable@0.1.0: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm deprecated source-map-url@0.4.0: See https://github.com/lydell/source-map-url#deprecated
npm deprecated set-value@2.0.0: Critical bug fixed in v3.0.1, please upgrade to the latest version.
npm deprecated uidv3@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm deprecated request-promise-native@0.1.0: request-promise-native has been deprecated because it extends the now deprecated request package, see https://github.com/request/request/issues/3142
npm deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm deprecated same4@1.0.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and newer ECMAScript syntax/features added
npm deprecated source-map-resolve@0.5.2: See https://github.com/lydell/source-map-resolve#deprecated
npm deprecated querystring@0.2.0: The querystring API is considered Legacy, new code should use the URLSearchParams API instead.
npm deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142
npm deprecated mixin-deep@1.3.1: Critical bug fixed in v2.0.1, please upgrade to the latest version.
npm deprecated left-pad@1.3.0: use String.prototype.padStart()
```

i-098d8ffe3074d3e84 (Web Server)

PublicIPs: 34.226.136.164 PrivateIPs: 10.0.4.236



```

npm WARN deprecated core-js@3.0.1: core-js<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.
npm WARN deprecated core-js-pure@3.1.3: core-js-pure@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js-pure.
npm WARN deprecated core-js@2.6.9: core-js@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.
npm WARN deprecated @material-ui/core@4.0.0-rc.0: Material UI v4 doesn't receive active development since September 2021. See the guide https://mui.com/material-ui/migration-v4/ to upgrade to v5.
npm WARN deprecated @material-ui/icons@4.0.0-rc.0: You can now upgrade to @mui/icons. See the guide: https://mui.com/guides/migration-v4/
added 1494 packages, and audited 1495 packages in 2m

3 packages are looking for funding
  run `npm fund` for details

94 vulnerabilities (6 low, 25 moderate, 47 high, 16 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
[root@ip-10-0-4-236 project-healer]#

```

i-098d8ffe3074d3e84 (Web Server)
PublicIPs: 34.226.136.164 PrivateIPs: 10.0.4.236

```

Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences
ENG 3:15 AM IN 09/12/2022

aws Services Search [Alt+S]
[root@ip-10-0-4-236 project-healer]# npm run server-install

> project-healer@1.0.0 server-install
> cd server && npm install

npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated url@0.1.0: Please see https://github.com/lydell/url#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x fewer dependencies
added 325 packages, and audited 326 packages in 13s

10 packages are looking for funding
  run `npm fund` for details

9 vulnerabilities (1 low, 5 moderate, 3 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:

i-098d8ffe3074d3e84 (Web Server)
PublicIPs: 34.226.136.164 PrivateIPs: 10.0.4.236

```

Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences
ENG 3:16 AM IN 09/12/2022

Step 16: Modifying Database URL in application for connecting it to private EC2 Database Server instead of MongoDB Atlas

- Run the following command in the project-healer directory to edit the server.js file present in server directory.

- sudo nano server/server.js
- Find the url used for MongoDB connection

```

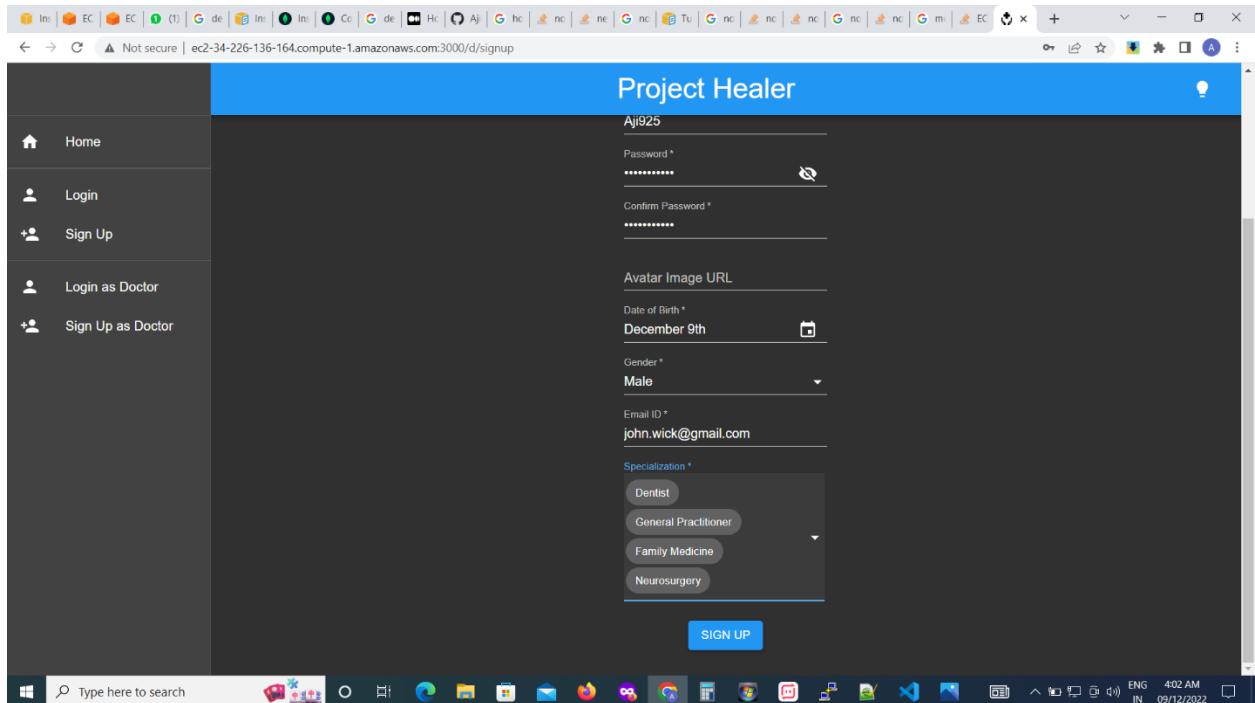
15
16 const url = process.env.DATABASE_URL || "mongodb://10.0.129.153:27017/local";
17 mongoose.connect(url, { useNewUrlParser: true, useUnifiedTopology: true });
18 mongoose.set("useFindAndModify", false);
19 mongoose.set("useCreateIndex", true);
20

```

- Edit the url (from above Fig line 16) and add mongodb url for our private Database Server (from above Fig the string “mongodb://10.0.129.153:27017/local”). The added url string will be of form:
 - “mongodb://<private_ip of db-server>:27017/local”
- This will allow the application server to send database queries to our private EC2 Database Server.

Step 17: Running the NodeJS application server and accessing it on browser

- To run our application server, execute following command inside the project-healer directory.
 - npm run dev
- This will compile the application code and run the application server. Once the application server is running, you can access the application via the following URL
 - <public-dns-name of web-server>:3000/
- Some screenshots showing the application running are given below.



Not secure | ec2-3-92-234-77.compute-1.amazonaws.com:3000/u/signup

Project Healer

Sign Up

First Name *

Last Name *

Username *

Password *

Confirm Password *

Avatar Image URL

Date of Birth *

Gender *

Email ID *

SIGN UP

Not secure | ec2-3-92-234-77.compute-1.amazonaws.com:3000/d/appointments

Project Healer

A Dr. Ajinkya Medhekar

Booked by Utsav Seth

Patient Name: Utsav Seth
Preferred Date: 10th December 2022
Preferred Time: 3:13 PM
Status: Under Approval

Description

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

DECLINE **ACCEPT**

The screenshot shows a web browser window with a blue header bar containing the text "Project Healer". On the left, there is a dark sidebar with a user profile picture and the name "Dr. Ajinkya Medhekar". The sidebar includes links for "Home", "Dashboard", "My Appointments", "Post An Article", and "Logout". The main content area displays a modal box titled "Booked by Utsav Seth". Inside the modal, the patient's name is listed as "Utsav Seth", with details: "Preferred Date: 10th December 2022", "Preferred Time: 3:13 PM", and "Status: Accepted". Below this, there is a section titled "Description" with placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.". At the bottom of the modal is a red "DECLINE" button. A green notification bar at the bottom of the page indicates: "Appointment was successfully accepted!" with a checkmark icon.

The screenshot shows a web browser window with a blue header bar containing the text "Project Healer". On the left, there is a dark sidebar with a user profile picture and the name "Dr. Ajinkya Medhekar". The sidebar includes links for "Home", "Dashboard", "My Appointments", "Post An Article", and "Logout". The main content area features a large white box with the text "WRITE. TEACH. HEAL." in bold capital letters. Below this, a sub-header reads: "Project Healer Health Articles is a medium for doctors and health specialists to inculcate helpful health tips and guidance to millions of patients around the world. Knowledge is the first step towards healing!". A form is displayed for posting an article. It has fields for "Title *", "Description", and "Content *". The "Title" field contains "Panadol is a double edged sword". The "Description" field contains: "Paracetamol is a good analgesic and antipyretic. Its side effects are low compared with other drugs, but may lead to severe hepatic failure when overdosing." The "Content" field contains: "Paracetamol is a good analgesic and antipyretic. Its side effects are low compared with other drugs, but may lead to severe hepatic failure when overdosing. The mother may find the same medicine but with different commercial names. **Tilanol** in the UK: **Beradol**, **Anadin** France: **Davilghan**, **Dolebran**, **Everlian** Saudi Arabia: Countries, **Fefadol**, **Benadol** Egypt: **Baramol**, **Abimol**, **Cital**, **Tempora**, **Timbalal**, **Kalpol** ... The scientific ingredient under the trade name above, the box outside and in the leaflet Also drug. "Dr. Mahmoud Fikri Consultant Emergency Medicine International Medical Center, Jeddah. At the bottom right of the form is a blue "SUBMIT" button. The taskbar at the bottom of the screen shows various application icons.

Project Healer

ec2-54-208-48-195.compute-1.amazonaws.com:3000

- Home
- Login
- Sign Up
- Login as Doctor
- Sign Up as Doctor

Diseases and Conditions

Find Doctor Online

Health Articles

About Us