*Architecture development*

Architecture Development Part 1: Virtual Machine

Configuration/Testing

As a part of this project, I have created Virtual Machines for various hospitals.

I have selected 5 hospitals Aspirus, Portage health, BCMH, MGH, UPHIE.

For these hospitals I have prepared VM’s with my IP address and secured them.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hospital | HIE** | **OS Compatible with HAPI-FHIR?** | **OS Compatible with Open EHR?** | **IP Address** | **Successfully Pinged the Other 4 VMs?**  **Yes or No** |
| Aspirus | yes | yes | 192.168.17.42 | Yes |
| Portage Health | yes | yes | 192.168.17.43 | Yes |
| BCMH | Yes | Yes | 192.168.17.44 | Yes |
| MGH | yes | yes | 192.168.17.45 | Yes |
| UPHIE | yes | yes | 192.168.17.46 | Yes |

Architecture Development Part 2: Installation, Configuration, and

Security of OpenEMR

For this part, I have installed and configured EHR as OpenEMR in each virtual machine.

OpenEMR gives cost-effective solutions for small to medium practices and low resource

settings due to its open-source licensing. Its customizability is a significant advantage,

accommodating various healthcare settings and needs.

**Web page screenshot of the hospital successful installation of OpenEMR.**

**Aspirus**

A screenshot of a computer

AI-generated content may be incorrect.

Portage Health  
A screenshot of a computer

AI-generated content may be incorrect.

BCMH

A screenshot of a computer

AI-generated content may be incorrect.

MGH

A screenshot of a computer

AI-generated content may be incorrect.

**Steps and commands you used to secure OpenEMR**

1. Updated and upgraded Ubuntu Server by using following commands in terminal

sudo apt-get update

sudo apt-get upgrade

2. Enabled automatic security updates by installing unattended upgrades package by

using ‘sudo apt-get install unattended-upgrades’ in terminal.

Enabled automatic updates by running following command in terminal

sudo dpkg-reconfigure --priority=low unattended-upgraded.

3.Installed 'ufw' (Uncomplicated Firewall) package by sudo apt-get install ufw

And Allowed HTTP, HTTPS, and SSH traffic by following commands

sudo ufw allow http

sudo ufw allow https

sudo ufw allow ssh

Enabled the firewall by sudo ufw enable

4. Secured Apache by - sudo nano /etc/apache2/conf-available/security.conf

Modified the following lines to increase security

ServerTokens Prod

ServerSignature Off

TraceEnable

Header set X-Content-Type-Options: "nosniff"

Header set X-Frame-Options: "sameorigin"

Header set X-XSS-Protection: "1; mode=block"

Header set X-Robots-Tag: "none"

Header set X-Download-Options: "noopen"

Header set X-Permitted-Cross-Domain-Policies: "none"

Saved and exit the file by using Ctrl+X, Y, Enter.

Enabled the new security headers by using following command

sudo a2enconf security

Restarting Apache by -- sudo systemctl restart apache2

5.Restarted Apache by -- sudo systemctl restart apache2

Secured OpenEMR By running all above mentioned commands in all VMs.

Aspirus  
A screenshot of a computer

AI-generated content may be incorrect.

Portage Health  
A screenshot of a computer

AI-generated content may be incorrect.

BCMH  
A screenshot of a computer

AI-generated content may be incorrect.

MGH

A screenshot of a computer

AI-generated content may be incorrect.

Architecture Development Part 3

A.1. Aspirus json screen shot

A screenshot of a computer

AI-generated content may be incorrect.

2. Portage Health json screen shot

A screenshot of a computer

AI-generated content may be incorrect.

3. BCMH json screen shot

A screenshot of a computer

AI-generated content may be incorrect.

4. MGH json screen shot

A screenshot of a computer

AI-generated content may be incorrect.

B.

|  |  |  |
| --- | --- | --- |
| Hospital | % Used of Total Served  Hospital Population | Amount of COVID Patients Created |
| Aspirus | 0.4% | 20 |
| Portage Health | 0.5% | 45 |
| BCMH | 2% | 140 |
| MGH | 6% | 1200 |

C. I didn't find anything difficult in this assignment because all the steps were clearly mentioned, and I followed them as instructed. However, my files were downloaded into the Home section instead of the Downloads folder, which was a bit tricky, I appreciate the instructor and the TA for me guiding me with the obstacle I faced. Thank you!

Architecture Development Part 4

Hapi-FHIR is an open-source implementation of the FHIR standard, enabling seamless data

exchange in healthcare. It supports interoperability, customization, and collaboration but requires

investment in training and infrastructure. Despite challenges, it’s use in public health surveillance enhances early disease detection and response through real-time data sharing and standardized information exchange, leading to evidence-based policies and improved public health outcomes.

A screenshot of a computer

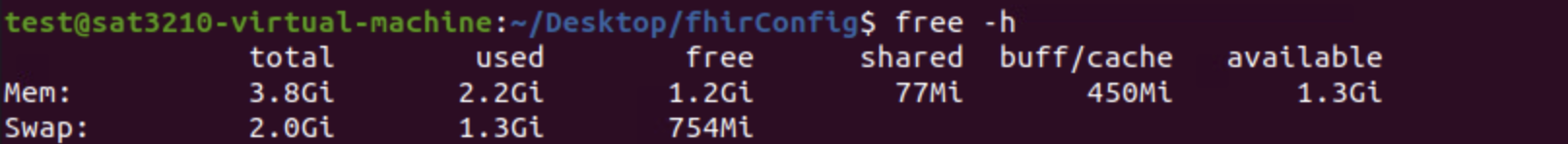
AI-generated content may be incorrect.

2. What command could you use to check the total, used, and free memory in system if you

thought the HAPI-FHIR server was not running properly due to lack of memory?

To check memory usage on an Ubuntu server, we can use:

free -h



3. You've updated the firewall rules on your Ubuntu server and want to check if the port (say

8080) used by the HAPI-FHIR server is open. What command would you use?

To check if port is open on server after updating the firewall rules, we can use the following

command:

sudo ufw status



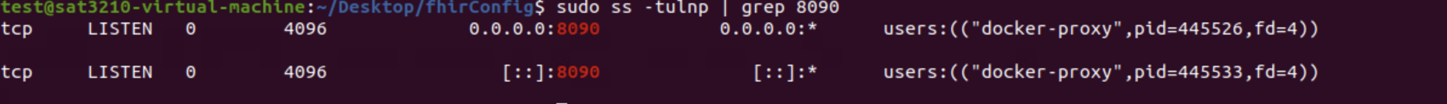
4.If the HAPI-FHIR server fails to start, you want to ensure that there are no other processes

occupying the default port (8080). What command should you use to check if the port 8080 is

currently in use?

To check if the port 8080 is currently in use we can run following command:

sudo ss -tulnp | grep 8090



Architecture Development Part 5

Screenshot

*A screenshot of a computer

AI-generated content may be incorrect.*

1.What is the process to create a new FHIR resource using Postman

and HAPI FHIR server?

Process to create a new FHIR resource using Postman and HAPI FHIR server

1. Install Postman.

2. Set up a Postman environment with variables for HAPI FHIR server's base URL and any

necessary authentication credentials.

3. Determine the type of FHIR resource want to create like Patient, Observation, Medication

Request etc.

4. Create a JSON object that adheres to the FHIR standard for the chosen resource type. Include

required attributes and any optional attributes as needed.

5. Compose POST Request by Opening Postman and create a new request. Set the request type to

POST and enter the endpoint URL for creating the desired resource.

6. Set Headers

7. Attach Request Body - Paste the JSON representation of the FHIR resource into the request

body section in Postman.

8. Send Request

9. Review the response from the server. If successful, it should return a status code of 201

Created along with the newly created resource in the response body. If there are errors, the server

will provide an appropriate status code and error message.

10. Handling errors

2.What is the process to manage authenFcaFon when using Postman

with a HAPI FHIR server?

1. In Postman, select the appropriate authentication method based on the authentication mechanism

supported by the HAPI FHIR server. Common methods include Basic Auth, OAuth 2.0, API Key,

etc.

2. Provide the necessary authentication credentials or tokens as required by the selected

authentication method.

• If using Basic Auth, enter the username and password.

• If using OAuth 2.0, configure the authorization URL, access token URL, client ID, client

secret, and any scopes required for accessing the FHIR resources.

• If using API Key, specify the API key value.

3. Apply Authentication to Requests: For each request send to the HAPI FHIR server, ensure that

the authentication method is applied.

• In Postman, we can set authentication at the collection level or at the request level.

4. Before sending requests to the HAPI FHIR server, verify that the authentication setup is correct

by sending a test request.

• Check the response to ensure that the server accepts the authentication credentials and returns

the expected data.

5. Handle Token Expiry and Refresh

3.How can you handle error responses from the HAPI FHIR server when

using Postman?

1. After sending a request to the HAPI FHIR server, examine the status code of the response in

Postman. This code indicates whether the request was successful or encountered an error.

2. Review the response body returned by the server for additional details about the error. The body

may contain information such as error messages, error codes, and additional context. Correct the

request based on the error message.

3. If the response status code indicates a client error (e.g., 400 Bad Request, 401 Unauthorized),

identify the cause of the error based on the information provided in the response body.

- Modify the request parameters or payload as needed and resend the request to address the error.

4. Implement Error Handling Logic

- Depending on requirements, we may implement specific error handling logic in application or

test suite.

- For automated testing, we can write scripts in Postman to automatically analyze error responses

and take appropriate actions based on the encountered errors.

5. Use the Console in Postman for debugging if needed.