PDF Utility Microservices

Video Link: [Video Link | Please increase volume]

Github Link: [Github Link | PDF Utility]

Introduction:

This report provides step-by-step instructions for implementing a microservice-based PDF utility application. The application comprises four distinct components deployed across separate VirtualBox virtual machines (VMs):

- **VM1:** API Gateway
- **VM2:** PDF Merger Microservice
- **VM3:** PDF-to-Image Converter Microservice
- **VM4:** Front End (React Application)

The report details the installation and configuration process—from setting up VirtualBox and creating the VMs through configuring the network to deploying the microservices and front end.

Installation of VirtualBox and Creation of Multiple VMs

1. Install VirtualBox

- **Download and Install:**
  - Go to the VirtualBox website and download the latest version for your host operating system (Windows).
  - Follow the installation wizard to install VirtualBox.

2. Create the Virtual Machines

For this project, four VMs will be created on VirtualBox.

VM Creation Steps (Repeat for Each VM or Clone VM):

1. **Open VirtualBox Manager:**
   Launch VirtualBox on your host machine.
2. **Create a New VM:**

    - Click the **"New"** button.
    - **Name:** Provide a name (e.g., VM1-API-Gateway, VM2-PDF-Merger, VM3-PDF-Converter, VM4-Front-End).
    - **Type & Version:**
        - Select **Linux** as the type.
        - Choose **Ubuntu (64-bit)** as the version.
    - **Memory Allocation:**
        - We have allocated 3048 MB RAM for each VM
    - **Hard Disk:**
        - We have allocated 25 GB of Storage to each VM.
3. **Install Ubuntu on Each VM:**
    - Download an Ubuntu ISO from Ubuntu's official website.
    - Mount the ISO in each VM's settings (under the Storage tab) and start the VM.
    - Follow the Ubuntu installation wizard on each VM.

Configuration of Network Settings to Connect the VMs

1. Setting Up the Network in VirtualBox

For communication between VMs, use a dual-adapter configuration:

Adapter 1 (NAT):

- **Purpose:** Provides internet access for package installation and updates.
- **Configuration:**
    - In each VM's **Settings > Network**, set **Adapter 1** to **NAT**.

Adapter 2 (Host-Only):

- **Purpose:** Facilitates predictable, isolated inter-VM communication.
- **Configuration:**
    - In each VM's **Settings > Network**, enable **Adapter 2** and set it to **Host-Only Adapter**.
    - Select the host-only network and disable DHCP.
    - **Assign Static IPs** using Netplan

2. Configuring Static IP Addresses with Netplan

For each VM, edit the Netplan configuration file (commonly in /etc/netplan/, e.g., 99-static-enp0s8.yaml) to assign a static IP on the Host-Only adapter (typically named enp0s8):

Example for VM1 (API Gateway):

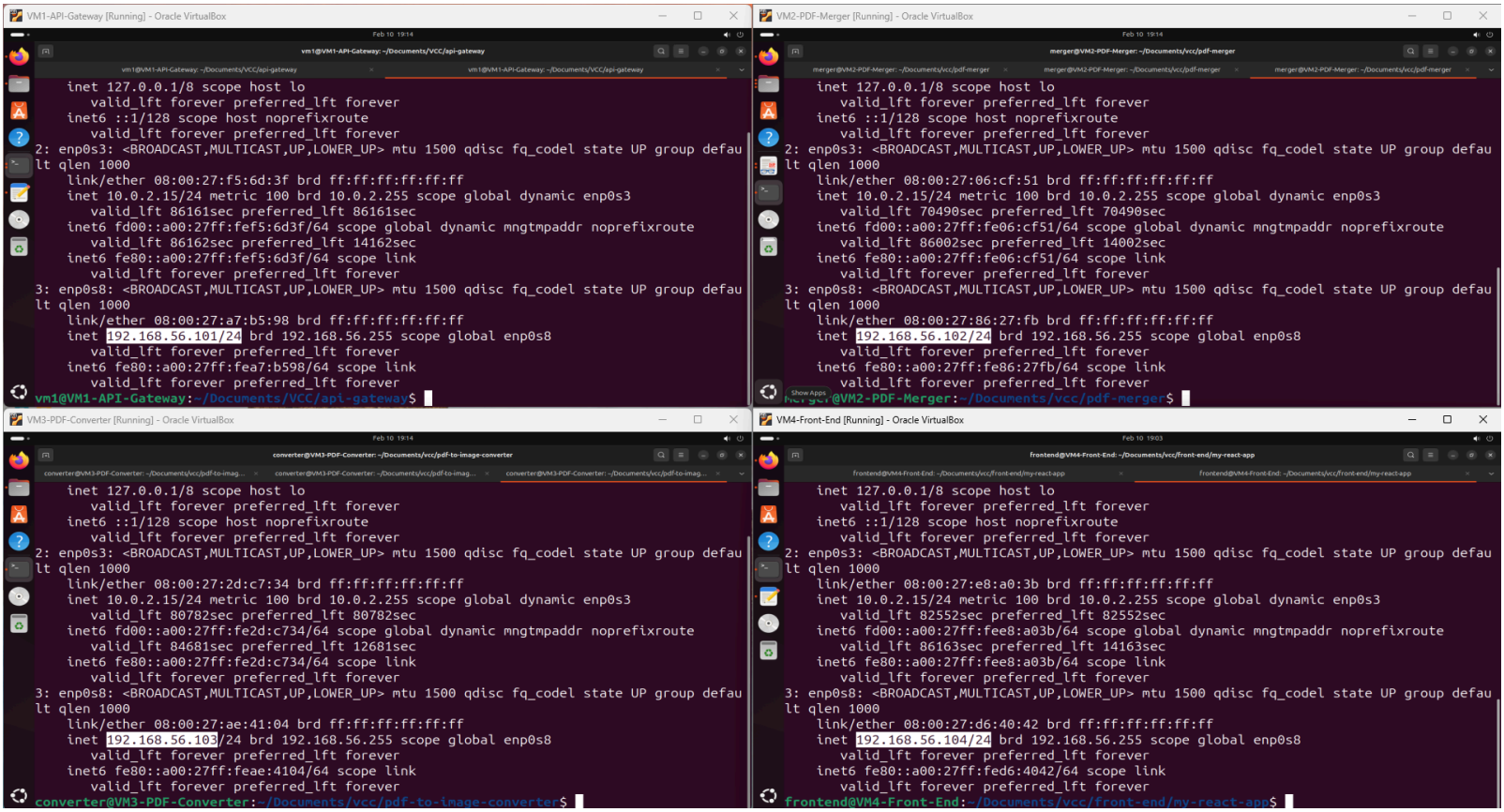
```yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.56.101/24
```

For the Other VMs:

- **VM2 (PDF Merger):** Use 192.168.56.102/24
- **VM3 (PDF-to-Image Converter):** Use 192.168.56.103/24
- **VM4 (Front End):** Use 192.168.56.104/24

After editing, apply the configuration with:

```
sudo netplan apply
```

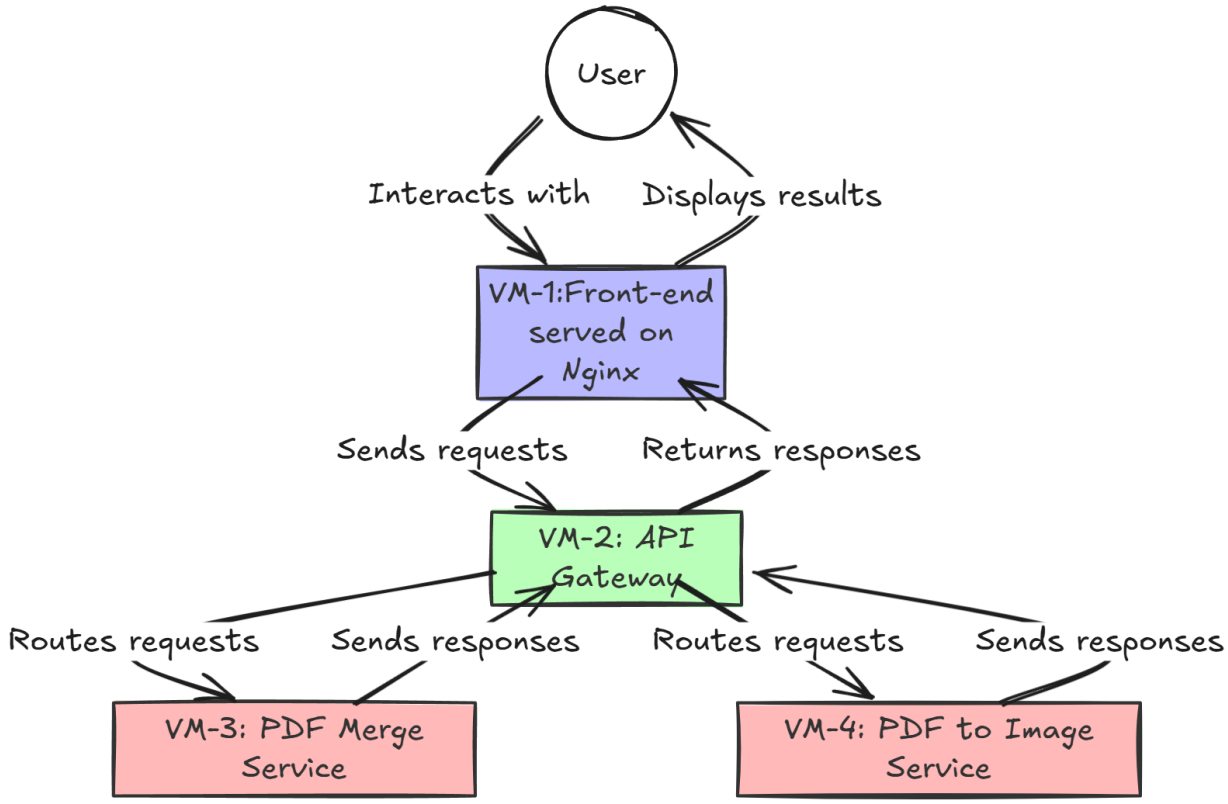


3. Testing Connectivity

Use the ping command on each VM to verify that they can communicate with each other using their Host-Only IPs. For example, on VM1:

```
ping 192.168.56.102
ping 192.168.56.103
ping 192.168.56.104
```

The architecture of the proposed PDF Utility Application



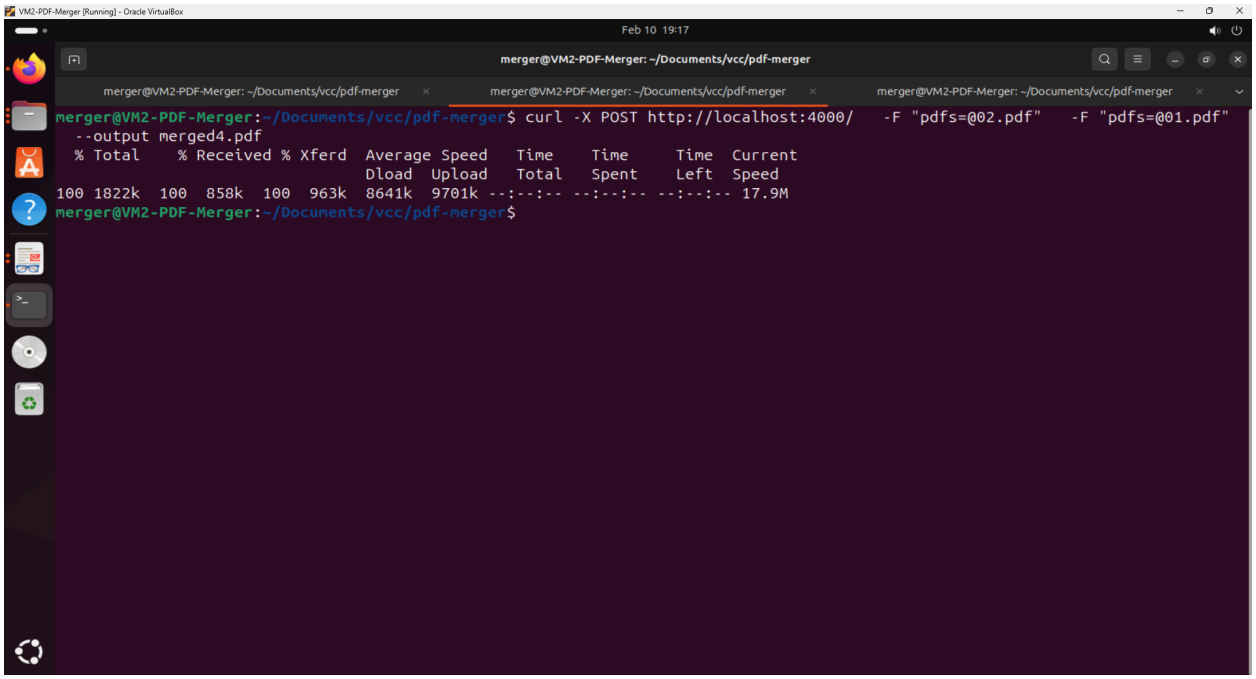
Deployment of the Microservice Application Across the VMs

VM1: API Gateway

1. Set Up the API Gateway:
    - Create a Node.js/Express project in a directory (e.g., ~/api-gateway).
    - Install dependencies: Express, http-proxy-middleware, dotenv, cors.
    - Configure proxy routes to forward /merge and /convert requests to the appropriate microservices (VM2 and VM3).
    - Use environment variables (in a .env file) to specify target URLs:
    - Start the server using npm start or nodemon server.js.

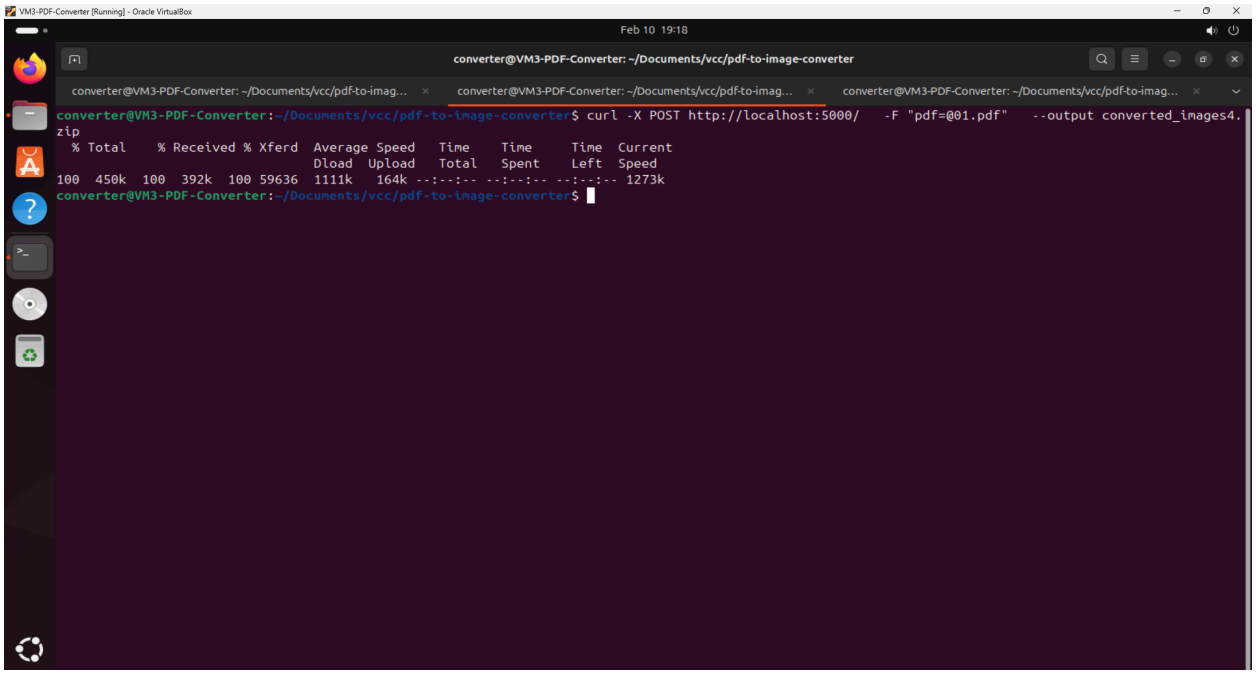
VM2: PDF Merger Microservice

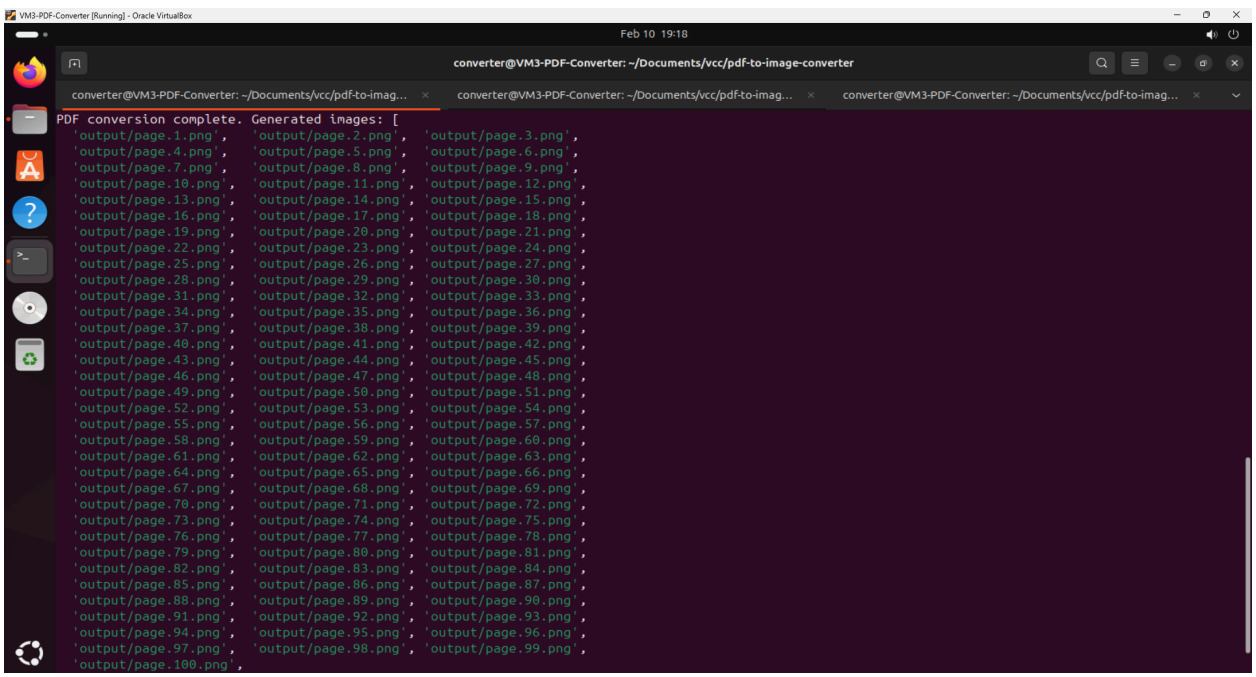
1. Set Up the PDF Merger Service:
    - Create a Node.js/Express project (e.g., in ~/pdf-merger).
    - Install dependencies: Express, multer, pdf-lib.
    - Implement an endpoint (e.g., POST /) that accepts multiple PDF files, merges them using pdf-lib, and returns the merged PDF.
    - Start the service on port 4000.



VM3: PDF-to-Image Converter Microservice

1. Set Up the PDF-to-Image Converter Service:
    - Create a Node.js/Express project (e.g., in ~/pdf-to-image-converter).
    - Install dependencies: Express, multer, pdf2pic, pdf-lib (for page counting if needed), archiver.
    - Implement an endpoint (e.g., POST /) that accepts a PDF file, converts all pages to images, packages them into a ZIP file, and returns the ZIP file.
    - Start the service on port 5000.





VM4: Front End (React Application)

1. Set Up the React App:
    - Create a React application using Create React App in a directory (e.g., ~/my-react-app).
    - Develop a simple one-page UI with tabs for merging PDFs and converting PDFs to images using Material-UI for a modern look.

- Use Axios to send HTTP requests to the API Gateway (base URL configured via a .env file.

2. Deploy the React App:
   - Run npm run build to generate static assets.
   - Install Nginx on VM4 and copy the build files to the default web root (/var/www/html).
   - Front end is accessible via the VM's IP (e.g., http://192.168.56.104).