

# A Raspberry Pi-based Private Cloud System for Remote Data Access

Pratik Kumar singh<sup>1</sup>, Prakhar Tibrewal<sup>2</sup>, P j Mohammed shoaib<sup>3</sup>, Naveen<sup>4</sup> Padmavathi M<sup>5</sup>, Dr. T.C.Manjunath<sup>6</sup>

<sup>1,2,3,4</sup>UG BE Dept. of ECE, Dayananda Sagar College of Engineering, Bangalore, Karnataka

<sup>5</sup>Asst. Prof., ECE Dept., Dayananda Sagar College of Engineering, Bangalore, Karnataka

<sup>6</sup>Professor & Head, Electronics & Communication Engg. Dept., Dayananda Sagar College of Engg., Bangalore, Karnataka

**Abstract—** The cloud storages are becoming expensive these days and can be expected to increase more and more. The cloud storages gives certain amount of the storage and then start asking the users to pay for the subscription model. Using Microprocessors and web development, it can offer you a better solution to have a personalised

**Private cloud storage.** A private cloud storage is a storage solution that is hosted on your own network, rather than on a third-party service. This means that your data is stored in a more secure location and you have more control over it. Microprocessors can be used to create a private storage and which will be only accessible to the particular users only. Microprocessors like raspberry pi and the hard drive like SD cards, HDDs are been used to make a private cloud storage. We are installing a NAS server in the raspberry pi and create the website which is used to access the private cloud storage. NAS servers can be secured using a variety of methods, such as encryption and authentication. This helps to protect data from unauthorized access. NAS servers can be secured using a variety of methods, such as encryption and authentication. This helps to protect data from unauthorized access.

**Keywords** -Network Attached Storage (NAS)

## I. INTRODUCTION

Now everything is done over the internet. The ease of access from anywhere and anytime makes people prefer to secure and store their files and information online. However, cloud providers are constantly increasing their prices. With the help of a Microprocessor, Wi-Fi Modules, and Hard Drive we could build and provide private cloud for anyone for free! This technology can replace Google Drive,

iCloud or Drop Box. This project is aimed at reducing the power consumption, cost, remote access of your data, providing a comprehensive layer of security, as well as authentication for the data being read or written without compromising on the portability of data.

will be utilizing the power of Microprocessors in the most optimized manner. Microprocessor are general purpose electronics processing devices which can be programmed to execute a number of tasks. They have compact size and low-power consumption, thus becomes ideal for being a portable option as well.

Microprocessors operate over a speed of over 3 GHZ, which makes it ideal for reliable speed of the system. The system will be a plug-and-play and can be used for professional and personal usage as well. We could extend this idea of storing to computation and cloud computing. The system can run on an internal network that is accessible anytime, no matter your internet connection. Unlike cloud storage, which is more vulnerable to third parties and prying eyes, this storage is completely restricted from anyone other than authorized personnel.

The system will have a firewall, load balancer and backup strategy on the software to ensure the safety and reliability of the system. The software will be agnostic in nature and thus in case of failure of hardware components will not affect the user with a heavy toil. Even upgrading the computational or storage capacity of the system will be way easier than any other cloud. Each cloud provider has a monthly package to charge for the services, while this system will be a one-time investment.

At the end we would have built a secure Private Cloud Storage system that allows users to store and access their personal data for free.

Each of these issues contribute to lack of efficiency and potential loss of income if customers cannot wait for a data outage to be corrected. Additionally, when it comes to data storage, small businesses find themselves faced with other storage-related needs such as:

- Lower cost options
- Ease of operation (many small businesses do not have IT staff)
- Ease of data backup (and it's always accessible when you need it)
- Growth capability

When compared as per the perspective of scalability, reliability and ease of access, both of the options have stood the benchmark values. There are two main aspects that we found to differentiate a Cloud Storage and a NAS

## II LITREATURE SURVEY

to create their own NAS servers using Raspberry Pi, the clinching detail is that this is the first successful attempt at providing remote access of data to clients along with an extra layer of security in the form of Virtual Private Network for any sort of paramount data over several networks. With the help of Dynamic DNS allocation [1], we have managed to keep a constant external IP address of the router so that clients can access the Samba Server in turn having access to the data hosted by the external drive. The reduction in the total power consumption without any compromise on the security, portability and back up of data can classify our model as the most preferable of the ones which already exists [2]. Here, we take advantage of the ease of usage of NAS servers and add a few tweaks to make data storage and accessibility both user friendly as well as preventing loss of data. The Raspberry Pi serves as the host to the external storage devices. We have also enabled RAID backup of data present in the external storage device to prevent loss of data. Implementing the idea of protecting data by using a secondary storage device ensures the maintaining of the confidentiality and integrity of the data stored on the external drive which is mounted to the Raspberry Pi

But this setup works for all devices in a particular LAN only. We use Port Forwarding and VPN, in this case, to make the data remotely accessible across any network. Apart from the remote access given, crucial data can be provided with sufficient security using the

virtual private network created [8, 10]. Making use of the fact that the Raspberry Pi draws very little power to host the external drives, this setup can be classified as low power compared to the conventional serverclient method [5]

## III PROPOSED METHODOLOGY

### 1) Creating a NAS Server

A Network Attached Storage (NAS) server is a dedicated device or software solution that provides centralized file storage and sharing capabilities over a network. It allows multiple users or devices to access files simultaneously and offers features like data backup, remote access, media streaming, and file synchronization.

- To install a NAS server on a Raspberry Pi, you'll need to follow a few steps. First, ensure that your Raspberry Pi is properly set up by connecting peripherals such as a keyboard, mouse, and monitor. Insert an SD card with a compatible operating system, such as Raspbian or Raspberry Pi OS. Boot up the Raspberry Pi and ensure it is connected to the network.
- Next, you'll need to install the necessary packages for setting up the NAS server. This typically involves updating the package lists and installing packages like Samba and Samba Common Bin. Samba is a widely used software suite that enables file sharing and printer sharing between devices.
- Once the installation is complete, you'll need to configure the Samba software. This involves editing the Samba configuration file and customizing it according to your requirements. You can specify shared folders, access permissions, and other settings in this file. For example, you can define a shared folder called "SharedFolder" with a specific path, set it as writable, and restrict guest access.
- After configuring Samba, you'll need to create a user account that will be used to access the shared folders on the NAS server. This user account will require a password. The user account and password will be used

for authentication when accessing the NAS server from other devices within the network.

- To apply the configuration changes, you'll need to restart the Samba service on the Raspberry Pi. This will ensure that the new configuration is loaded and the NAS server is ready for use.
- Once the NAS server is set up and running, you can access it from other devices within the network. Simply enter the network address (IP address) of the Raspberry Pi NAS server in a file manager or file explorer on the other devices. You'll be prompted to enter the username and password you created earlier to authenticate and access the shared folders.
- It's important to note that the above steps provide a general overview of installing a NAS server on a Raspberry Pi. The specific details and commands may vary depending on the operating system you are using and your specific requirements. It's always recommended to refer to official documentation or online tutorials for detailed instructions tailored to your setup.

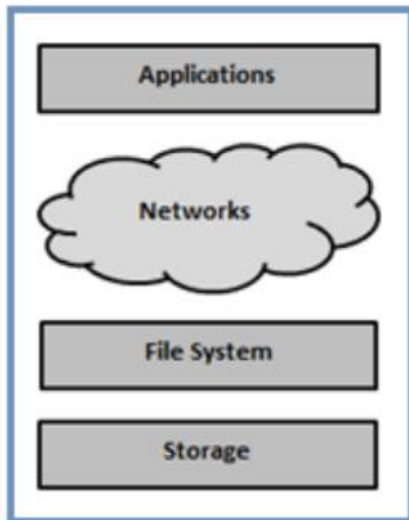


Fig 1. Nas server

## 2) creating a website

In front-end, Using html, css, javascript we create it.

Create a folder and create 3 files named as index.html,

Style.css, script.js.

In index.html file ,we create the structure of the website ,we have created a website with 3-4 pages ,First we create the main page with describing about website and login/sign up option at the corner of it.By clicking on login page , it opens the login page where user can login with their credentials and then can access to their private storage . CSS is used to style the login and page style ,which looks nice and easy to understand. Javascript is used to create how the login page works .

In back end part,there is a node js server with 3 end points ,the first is the main page(/), the other is the login page(/login) and the other is the main storage page(/storage).

## IV. CODING TECHNOLOGY

Technologies used in creating a website to access the nas storage are

In the **Front-End Technologies:**

### **HTML (Hypertext Markup**

**Language):** HTML is used for structuring the website's content and defining the elements and layout of web pages. Sure, here is a paragraph version of the explanation of HTML:

- HTML, or HyperText Markup Language, is the standard markup language for creating web pages. It is a text-based language that uses tags to identify the different elements of a web page, such as headings, paragraphs, lists, images, and links. HTML elements can be nested, which means that one element can be contained within another element. HTML can be used to create a wide variety of web pages, from simple static pages to complex interactive pages.
- Here are some of the basic HTML elements:

- <html> - The <html> element is the root element of an HTML document.
- <head> - The <head> element contains the metadata for an HTML document, such as the title, author, and keywords.
- <body> - The <body> element contains the content of an HTML document, such as text, images, and links.
- <h1> - The <h1> element is used to create a heading.
- <p> - The <p> element is used to create a paragraph.
- <img> - The <img> element is used to insert an image into an HTML document.
- <a> - The <a> element is used to create a link.
- Here are some of the basic HTML attributes:
- id - The id attribute is used to identify an element uniquely.
- class - The class attribute is used to group elements together.
- style - The style attribute is used to style an element.
- href - The href attribute is used to specify the URL of a link.

**CSS (Cascading Style Sheets):** CSS is used for styling the website's appearance, including colors, fonts, layouts, and responsiveness. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is used to control the appearance of web pages. It can be used to change the font, size, color, and position of text; the size, shape, and color of borders; and the background color and image of a web page. CSS can also be used to create animations and other effects.

CSS is a powerful tool that can be used to create a wide variety of visual effects. It is easy to learn and use, and it can be used to make web pages look more professional and appealing.

## Basics of CSS

CSS is a declarative language, which means that you tell the browser what you want the page to look like, and the browser figures out how to do it. CSS is made up of rules, which are made up of selectors and declarations.

Selectors are used to identify the elements that you want to style. Declarations are used to specify the properties of the elements that you have selected.

For example, the following CSS rule will change the font of all paragraphs on a web page to Arial:

CSS properties

CSS has a wide variety of properties that can be used to control the appearance of web pages. Some of the most common properties include:

- font-family: The font family of the element.
- font-size: The size of the font.
- color: The color of the text.
- background-color: The background color of the element.
- border: The border of the element.
- margin: The margin of the element.
- padding: The padding of the element.

**JavaScript:** JavaScript is used for implementing interactivity and dynamic behavior on the website, such as form validation, file uploads, and user interactions

JavaScript is a high-level, interpreted programming language that is predominantly used for web development. It plays a crucial role in creating interactive and dynamic websites. Here's an overview of JavaScript and its importance in website development:

1. **Client-Side Scripting:** JavaScript primarily runs on the client-side, meaning it is executed within the user's web browser. It allows web developers to add interactivity and responsiveness to web pages, enhancing the user experience. Unlike HTML and CSS, which handle the structure and presentation of a website, JavaScript enables dynamic content and real-time updates.
2. **Interactivity:** JavaScript enables developers to create interactive elements on web pages. It can handle user actions such as button clicks, form submissions, and mouse movements, and respond accordingly. This

interactivity enhances user engagement, enables user input validation, and enables the creation of dynamic user interfaces.

3. **DOM Manipulation:** The Document Object Model (DOM) represents the structure of a web page as a tree-like structure. JavaScript provides powerful methods and APIs to access and manipulate the DOM. Developers can dynamically modify the content, styles, and attributes of HTML elements, enabling the creation of dynamic and responsive websites.
4. **Event Handling:** JavaScript allows developers to respond to various events triggered by user actions or system events. Events such as mouse clicks, keyboard inputs, and page loading can be handled using JavaScript. This capability enables the creation of interactive features like dropdown menus, sliders, and form validation.
5. **Asynchronous Operations:** JavaScript supports asynchronous programming through techniques like callbacks, promises, and `async/await`. This allows web developers to perform tasks such as making AJAX requests to servers, fetching data from APIs, and handling time-consuming operations without blocking the execution of other scripts. Asynchronous operations enhance the responsiveness of web applications and improve overall performance.
6. **Third-Party Libraries and Frameworks:** JavaScript has a vast ecosystem of libraries and frameworks that streamline web development. Popular libraries like jQuery, React, and AngularJS provide pre-built components, simplified APIs, and additional functionality to simplify common tasks and speed up development.
7. **Browser Compatibility:** JavaScript is supported by all modern web browsers, making it a widely adopted language for web development. It ensures cross-browser compatibility, allowing websites to function consistently across different browsers and platforms.
8. **Enhanced User Experience:** JavaScript's capabilities enable developers to create rich and interactive user experiences. Features

like image sliders, form validation, real-time updates, and smooth animations can be easily implemented using JavaScript. These interactive elements make websites more engaging, user-friendly, and visually appealing.

9. **Integration with Server-Side Technologies:** JavaScript can also be used on the server-side with platforms like Node.js, allowing developers to use the same language for both client-side and server-side development. This facilitates seamless communication and data exchange between the client and server, enabling efficient and scalable web applications.

## 2. Back-End Technologies:

- **Node js :**

Node.js is a powerful open-source JavaScript runtime environment that allows developers to build scalable and high-performance web applications. It has gained significant popularity in web development due to its unique features and benefits. Here's an overview of Node.js and its importance in web development:

1. **JavaScript Everywhere:** Node.js enables developers to use JavaScript not only on the client-side but also on the server-side. This unification of programming languages allows developers to use the same language and codebase for both front-end and back-end development, resulting in improved productivity, code reusability, and reduced development time.
2. **Non-Blocking, Asynchronous Architecture:** One of the key strengths of Node.js is its non-blocking, event-driven architecture. It uses an event loop to handle multiple client requests simultaneously without blocking the execution of other operations. This asynchronous nature makes Node.js highly efficient and scalable, as it can handle a large number of concurrent connections and perform I/O operations efficiently, resulting in fast and responsive web applications.
3. **Scalability and Performance:** Node.js is known for its excellent scalability and performance characteristics. Its event-driven architecture, coupled with non-blocking I/O

operations, enables handling a large number of simultaneous connections with minimal system resources. This makes Node.js ideal for building real-time applications, chat systems, streaming services, and applications requiring high concurrency.

4. **Vibrant Ecosystem and Package Manager:** Node.js has a vast ecosystem of open-source libraries and packages available through its package manager, npm (Node Package Manager). This extensive collection of modules provides developers with ready-to-use solutions for various functionalities, such as database integration, web frameworks, authentication, testing, and more. This thriving ecosystem accelerates development and reduces the need for building everything from scratch.
5. **Community Support and Developer Productivity:** Node.js has a large and active community of developers, contributing to its growth and providing valuable resources, documentation, and support. The community-driven nature of Node.js ensures continuous improvements, frequent updates, and the availability of numerous resources, tutorials, and forums. This support network enhances developer productivity and fosters collaboration within the community.
6. **Microservices and APIs:** Node.js is well-suited for building microservices architectures and APIs. Its lightweight and modular nature enable developers to create independent and scalable microservices that can communicate with each other through APIs. This approach allows for better code organization, scalability, and flexibility in building complex applications.
7. **Real-time Applications:** Node.js excels in developing real-time applications that require instant data updates and bidirectional communication between clients and servers. With the help of frameworks like Socket.io, developers can build real-time chat applications, collaborative tools, live dashboards, and multiplayer games with ease.
8. **Integration with Front-end Frameworks:** Node.js seamlessly integrates with popular

front-end frameworks like React, Angular, and Vue.js. This enables developers to create full-stack JavaScript applications, where the same language, tooling, and ecosystem are used on both the front-end and back-end. This synergy results in better code maintainability, improved development workflow, and faster application performance.

9. **Cross-Platform Compatibility:** Node.js is cross-platform, running on major operating systems like Windows, macOS, and Linux. This flexibility allows developers to deploy Node.js applications on various environments, making it an ideal choice for building applications that need to run on different platforms.

## V. CONCLUSIONS

The Private Cloud Storage project is a novel solution that provides users with a secure and reliable way to store their data remotely. Using Raspberry Pi, the project offers a self-hosted cloud storage solution that can be stored or access it from anywhere using website and Mobile Application Overall, the Private Cloud Storage project is a valuable contribution to the field of cloud storage solutions. It demonstrates the potential of using low-cost hardware to build sophisticated solutions that can offer an alternative to centralized cloud storage solutions. The project provides valuable insights into the development and deployment of cloud storage solutions while emphasizing the importance of data privacy and security in the current digital age.

## REFERENCE

- [1] Amit Rakesh Shrivastava, Jayant Gadge, Home Server and NAS using Raspberry Pi Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8126184>
- [2] Mehdi Sookhak, Abdullah Gani, Muhammad Khurram Khan and Rajkumar Buyya, "Dynamic remote data auditin/g for securing big data storage in cloud computing", Information Sciences, Vol. 380, 20 February 2017, pp. 101- 116

[3] Mohiuddin Ahmed, Abu Sina Md. Raju Chowdhury, Mustaq Ahmed and Md. Mahmudul Hasan Rafce, “An Advanced Survey on Cloud Computing and State-of-the-art Research Issues”, International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012, ISSN: 1694-0814, pp. 201-20

7 [4] Heng Liao, “Storage Area Networks”, Technology White Paper, Issue 1: April, 2003 PMC-2022178

[5] “Eltechs Guide to NAS Servers”. Available: <https://eltechs.com/raspberry-pi-nas-guide>