22-Aug-2024

Internship Day - 5 Report:

Domain and Hosting Charges?

A domain is the web address (e.g., www.example.com) that users type into a browser to access a website. Hosting refers to the service that stores your website files and makes them accessible on the internet. These are separate services, and each incurs its own charges. Domain registration typically involves an annual fee, while hosting services can be billed monthly or annually based on the type of plan chosen.

Knowledge of Server, SQL, and PHP?

A server is a computer or system that provides resources, data, services, or programs to other computers, known as clients, over a network. SQL (Structured Query Language) is a standard language for managing and manipulating databases. PHP (Hypertext Preprocessor) is a server-side scripting language used for web development, allowing developers to create dynamic web pages. Understanding these technologies is crucial for full-stack development, enabling seamless interaction between the front-end and back-end of a web application.

IIS Server?

IIS (Internet Information Services) is a web server created by Microsoft for hosting websites and applications on Windows servers. It supports various protocols, including HTTP, HTTPS, FTP, and more. IIS is specifically designed for Windows environments, offering features like security, logging, and application management, making it a popular choice for enterprises that use Microsoft technologies.

XAMPP, LAMP, and WAMP?

- **XAMPP**: A cross-platform web server solution stack package that includes Apache, MySQL, PHP, and Perl, designed for easy installation and use on multiple operating systems.

- **LAMP**: A stack consisting of Linux, Apache, MySQL, and PHP, commonly used for developing and deploying web applications on Linux servers.

- **WAMP**: A Windows-based stack that includes Windows, Apache, MySQL, and PHP, providing a similar development environment for Windows users.

Requirements for Developing a PHP Website?

To develop a PHP website, you typically need:

- A web server (like Apache or Nginx) to serve your PHP files.
- A database (like MySQL) to store and manage your data.
- A PHP interpreter to process the PHP code and generate dynamic content.
- A development environment (like XAMPP, LAMP, or WAMP) to set up these components easily on your local machine.

Need for Apache Server?

Apache is one of the most widely used web servers in the world, known for its flexibility, power, and community support. It serves web content to users by processing HTTP requests and delivering HTML pages, images, and other resources. Apache supports various modules, allowing for extended functionality like URL rewriting, authentication, and security features, making it a critical component for hosting dynamic websites and applications.

Introduction of Elastic Ip:

An Elastic IP in AWS is a static public IPv4 address that you can allocate to your AWS account. It allows you to maintain a consistent IP address for your resources, even if you stop or restart your EC2 instances. This is particularly useful for applications that require a stable endpoint for users to connect to.

Uses of Elastic IP:

- **Static IP Addressing:** Unlike standard public IP addresses that change when an instance is stopped or restarted, an Elastic IP remains constant, ensuring that your applications can always be accessed at the same IP address.

- **Failover Capability:** If an instance fails or needs to be replaced, you can quickly remap the Elastic IP to another instance, minimizing downtime and maintaining service continuity.

- **Consistent Endpoint:** Elastic IPs serve as stable identifiers for your cloud resources, which is beneficial for configuring external services, such as DNS records or firewall rules.

- **Dynamic Allocation:** You can programmatically associate and disassociate Elastic IPs as needed, allowing you to adapt to changing demands and scale your infrastructure efficiently.

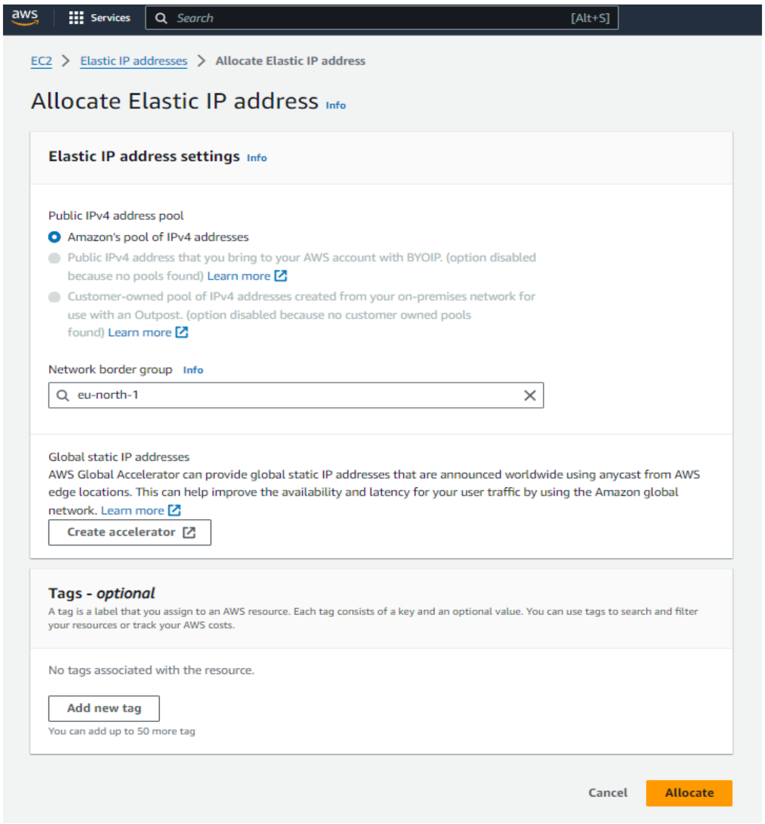
Steps to Create and Associate an Elastic IP with a Launch Instance:

Open the Amazon EC2 Console:

- Navigate to the Amazon EC2 console.

Allocate an Elastic IP Address:

- In the navigation pane, choose Network & Security and then select Elastic IPs.
- Click on Allocate Elastic IP address.
- (Optional) Choose the Network border group if you want to specify where the Elastic IP will be allocated.
- Click Allocate to create the Elastic IP.



Allocate Elastic IP address

Elastic IP address settings

Public IPv4 address pool

● Amazon's pool of IPv4 addresses

○ Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) Learn more

○ Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) Learn more

Network border group: eu-north-1

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. Learn more

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

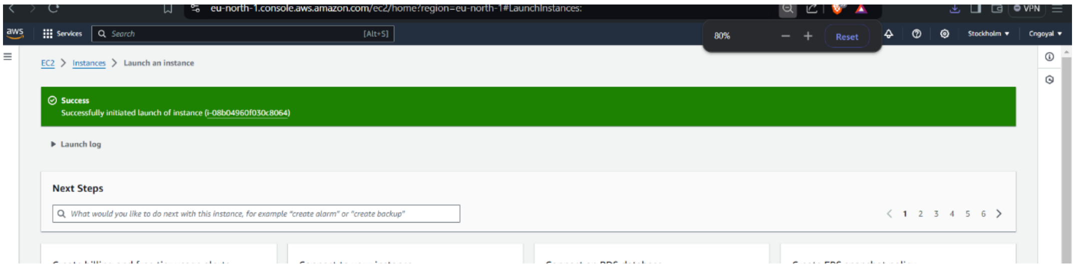
No tags associated with the resource.

You can add up to 50 more tag

Launch an EC2 Instance:

- In the EC2 console, click on Instances and then Launch Instance.
- Follow the prompts to select an Amazon Machine Image (AMI), instance type, and configure instance details.

- Ensure that the instance is in a public subnet to allow internet access.

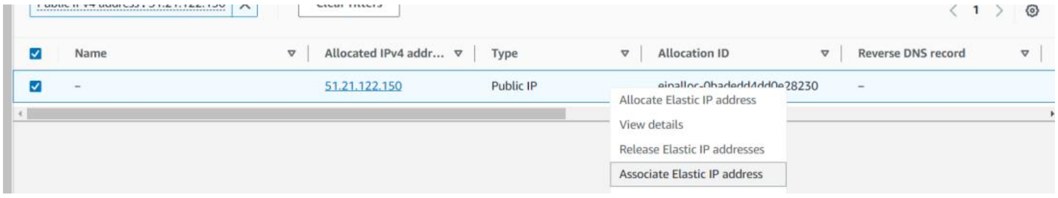

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchInstances:
aws
Services
Search
[Alt+S]
80%
Reset
Stockholm
Cngoyal
EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-08b04960f030c8064)

Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
1 2 3 4 5 6

Associate the Elastic IP with the Instance:

- After launching the instance, go back to the Elastic IPs section in the EC2 console.
- Select the Elastic IP address you allocated.
- Click on Actions and choose Associate Elastic IP address.
- For Resource type, select Instance and then choose the instance you just launched.
- Click Associate to link the Elastic IP with your instance.



☑	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record
☑	–	51.21.122.150	Public IP	eipalloc-0badedd4dd0e28230	–

1
Allocate Elastic IP address
View details
Release Elastic IP addresses
Associate Elastic IP address

Verify the Association:

- Once associated, you can check the instance details to confirm that the Elastic IP is now linked to your instance.
- You can also test connectivity by accessing the instance using the Elastic IP address.

Benefits of Associating an Elastic IP:

- **Persistent Connectivity:** Your applications can maintain a consistent IP address, even if the underlying instance changes.

- **Easy Failover:** In case of instance failure, you can quickly remap the Elastic IP to a new instance, ensuring minimal downtime.

- **Simplified DNS Management:** You can point your domain names to the Elastic IP, making it easier to manage your web applications.

23-July-2024

Internship Day - 6 Report:

What is the mean of 400 in chmod ?

In the context of Unix and Linux file permissions, the chmod command is used to change the permissions of a file or directory. The permissions are represented using a three-digit octal number, where each digit corresponds to a different set of permissions.

Understanding chmod 400

- 400 is an octal representation of file permissions.
- Each digit represents permissions for the owner, group, and others, respectively.

Here's the breakdown of 400:

1. **First Digit (4):**

    - This represents the permissions for the owner of the file.
    - The value 4 corresponds to read permission. This means the owner can read the file but cannot write to it or execute it.
    - In binary, this is represented as 100, which means:
        1. Read (4): Yes
        2. Write (2): No
        3. Execute (1): No

2. **Second Digit (0):**

    - This represents the permissions for the group.
    - The value 0 means no permissions are granted to the group. In binary, this is represented as 000.

3. **Third Digit (0):**

    - This represents the permissions for others (everyone else).
    - Similarly, the value 0 means no permissions are granted to others. In binary, this is also represented as 000.

Summary of chmod 400

When you set permissions to 400 using chmod, you are effectively allowing only the owner of the file to read it, while the group and others have no permissions at all.

This is a common setting for sensitive files (like private keys or configuration files) that should only be accessible by the user who owns them.

Introduction of Filezilla?

FileZilla is a popular, open-source file transfer protocol (FTP) client that allows users to transfer files between a local computer and a remote server. It supports multiple file transfer protocols, including FTP, SFTP (SSH File Transfer Protocol), and FTPS (FTP Secure).

FileZilla is widely used by web developers, system administrators, and anyone needing to upload, download, or manage files on a server. The interface is user-friendly, providing an easy drag-and-drop system for file management, and it includes features like:

- **Cross-platform compatibility:** Works on Windows, macOS, and Linux.
- **File transfer resume:** Allows users to resume file transfers if they get interrupted.
- **Support for large files:** Handles files larger than 4GB.
- **Directory comparison:** Highlights differences between local and remote directories.
- **Remote file editing:** Allows users to edit files on the server without downloading them first.

FileZilla also has a server counterpart, FileZilla Server, which enables users to set up their own FTP server for secure file sharing.

Windows Installation

1. Download FileZilla Client from filezilla-project.org (https://filezilla-project.org/) for Windows.
2. Run the downloaded `.exe` installer file.
3. Accept the license agreement in the installation wizard.
4. Choose installation options and select the installation location.
5. Click **Install** to complete the installation.

Linux Installation (Ubuntu/Debian-based)

1. Update your package list with `sudo apt update`.
2. Install FileZilla using `sudo apt install filezilla’.
3. Verify the installation by typing `filezilla` in the terminal.

Setting Up an Nginx Web Server on an Amazon EC2 Instance and Uploading Files via FileZilla

Step-by-Step Guide

1. **Create an EC2 Instance:**
   - Log in to the AWS Management Console, navigate to EC2, and click on "Launch Instance" to create a new instance using an Amazon Machine Image (AMI) like Ubuntu or Amazon Linux.

2. **Configure Instance Details:**
   - Choose instance type (e.g., t2.micro for free tier eligibility), configure instance details, and click "Next" until you reach the "Review and Launch" section.

3. **Set Security Group:**
   - Create a new security group or select an existing one, ensuring to allow inbound traffic on ports 22 (SSH) and 80 (HTTP) for web access.

4. **Launch the Instance:**
   - Review your settings and click "Launch," then select or create a new key pair (download the .pem file) to access the instance.

5. **Access the Instance via SSH:**
   - Open a terminal and use the command ssh -i /path/to/your-key.pem ec2-user@your-instance-public-ip (replace with your key path and instance public IP) to connect.

6. **Update Package Repository:**
   - Run sudo apt update (for Ubuntu) or sudo yum update (for Amazon Linux) to ensure your package repository is up to date.

7. **Install Nginx:**
   - Execute sudo apt install nginx (for Ubuntu) or sudo yum install nginx (for Amazon Linux) to install the Nginx web server.

8. **Start Nginx Service:**
   - Start Nginx with sudo systemctl start nginx and enable it to start on boot with sudo systemctl enable nginx.

9. **Verify Nginx Installation:**
   - Open a web browser and navigate to http://your-instance-public-ip to see the Nginx welcome page.

10. **Prepare for File Upload:**
    - Ensure your instance's security group allows inbound traffic on port 21 (FTP) or use SFTP (port 22) for secure file transfer.

11. **Open FileZilla:**
    - Launch FileZilla on your local machine and go to Site Manager (File > Site Manager).

12. **Configure FileZilla Connection:**
    - Create a new site with Host as your instance's public IP, Protocol as SFTP, and enter the username (ec2-user for Amazon Linux or ubuntu for Ubuntu) and the path to your private key in the settings.

13. **Connect to the Instance:**
    - Click "Connect" in FileZilla to establish a connection to your EC2 instance.

14. **Navigate to the Web Directory:**
    - In FileZilla, navigate to the web root directory (usually /var/www/html for Nginx) in the right panel.

15. **Upload Your File:**
    - Drag and drop your website files from the left panel (local) to the right panel (remote) to upload them to the Nginx web directory.

16. **Verify File Upload:**
    - Go back to your web browser and navigate to http://your-instance-public-ip/your-file.html to ensure the uploaded file is accessible.

By following these steps, you will have successfully created an EC2 instance, installed Nginx, and uploaded files using FileZilla.

24-July-2024

Internship Day - 7 Report:

Overview of File Permissions and User Management in Linux

Understanding File Permissions in Linux

File Permissions:

In Linux, each file and directory has associated permissions that determine who can read, write, or execute them. Permissions are divided into three categories:

- **Owner**: The user who owns the file.
- **Group**: A group of users who share permissions.
- **Others**: All other users on the system.

Permission Types:

- **Read (r):** Permission to read the file or list the directory.
- **Write (w):** Permission to modify the file or add/remove files in a directory.
- **Execute (x):** Permission to execute a file or access a directory.

Checking File Permissions

- Use the command **ls -l** to list files and their permissions in a directory.
  Example Output: -rw-r--r-- 1 user group 0 Oct 1 12:00 file.txt

  1. The first character indicates the type (e.g., - for a file, d for a directory).
  2. The next three characters represent owner permissions (read, write, execute).
  3. The following three represent group permissions.
  4. The last three represent permissions for others.

- Use **ls -ltr** to list files in long format sorted by modification time (oldest first).

Creating and Managing Users

1. **Create a User:**

- **Command:** sudo useradd -m -s /bin/bash cgoyal
  1. -m: Creates a home directory for the user.
  2. **-s /bin/bash**: Sets the default shell to bash.

- **Set a password**: sudo passwd cgoyal

2. **View User Information:**

- **Command**: cat /etc/passwd
- Displays user account information, including username, user ID, group ID, home directory, and shell.

3. **Delete a User:**
    - **Command**: sudo userdel ram
    - **To remove the user and their home directory:** sudo userdel -r ram

4. **Create Another User:**

- **Command**: sudo useradd -m -s /bin/bash chintu
- **Set a password:** sudo passwd chintu
- **Switch to the new user**: su chintu (enter the password for chintu).

File and Directory Management

1. **Create a Directory:**
    - **Command:** sudo mkdir demo_1 (creates a new directory named demo_1).

2. **Remove a Directory:**
    - Command: sudo rmdir demo_1 (removes an empty directory).

3. **Create a File:**
    - **Command**: touch demo_1.txt (creates an empty file named demo_1.txt).

4. **Write to a File:**
    - **Command**: echo "my name" > cgoyal.txt (writes "my name" to cgoyal.txt).
    - To enter content interactively, use cat > cgoyal.txt and press Ctrl + D to exit.

5. **Edit a File:**
    - Use a text editor like vi: vi cgoyal.txt (opens cgoyal.txt for editing).

6. **Remove Files:**
    1) **Command**: sudo rm *.txt (removes all .txt files in the current directory).

Day – 25 July, 2024

Internship Day - 8 Report:

Managing AWS EC2 Instances: Snapshots and Volumes

Definitions:

Volume: A volume in AWS is a durable block storage device that can be attached to an EC2 instance. It is used to store data persistently, even when the instance is stopped or terminated. Volumes are typically used for storing the operating system, application data, and other files that need to be retained.

Snapshot: A snapshot is a point-in-time backup of an EBS (Elastic Block Store) volume. It captures the data stored in the volume at a specific moment, allowing you to restore the volume to that state later. Snapshots are stored in Amazon S3 and can be used to create new volumes or to back up existing volumes.

Steps to Manage EC2 Instances, Volumes, and Snapshots:

1. Create an EC2 Instance: Launch a new instance in the AWS cloud.

2. Access the Instance: Connect to the instance via terminal using SSH.

3. Start Apache HTTP Server: Run the command to start and enable the Apache web server (httpd).

4. Change Ownership: Use the chown command to change file ownership on the instance.

5. Upload Website: Use FileZilla to upload your website files to the instance.

6. Create a Snapshot: In the EC2 dashboard, select the instance, go to the snapshot section, and create a snapshot of the volume.

7. Monitor Snapshot Status: Wait for the snapshot creation to complete and check its status.

8. Create Volume from Snapshot: Use the snapshot to create a new volume.

9. Attach Volume to New Instance: Select the new instance and attach the newly created volume, giving it a name and disabling the automatic IP option.

10. Detach Volume from Running Instance: If the volume is in use, stop the instance, detach the volume, and then attach it to the new instance.

11. Start the New Instance: Run the new instance with the attached volume to ensure the backup website runs successfully.

By following these steps, you can effectively manage your EC2 instances, create backups using snapshots, and utilize volumes for persistent storage.

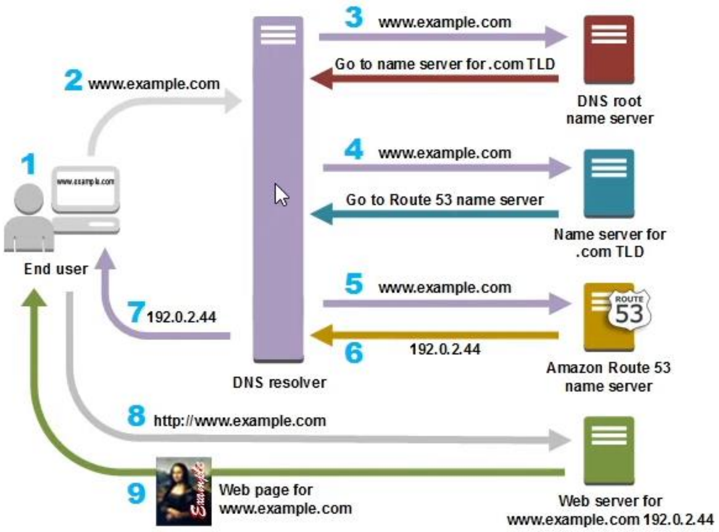
26-July-2024

Internship Day - 9 Report:

Setting Up Amazon Route 53 for Domain and DNS Management

What is Amazon Route 53?

Amazon Route 53 is a scalable and highly available Domain Name System (DNS) web service designed to provide reliable routing of end users to Internet applications by translating human-readable domain names (like www.example.com) into IP addresses. It helps improve the speed and reliability of applications by directing traffic to the nearest endpoint.



Why Use Route 53?

1. **DNS Management:** Route 53 simplifies the management of DNS records and provides a user-friendly interface for creating and managing DNS configurations.

2. **Traffic Routing:** It offers various routing policies (like latency-based routing, geolocation routing, etc.) to direct user traffic efficiently.

3. **Health Checks:** Route 53 can monitor the health of your application endpoints and route traffic away from unhealthy instances.

Types of Routing Policies in Route 53:

- Simple Routing: Routes traffic to a single resource (e.g., an IP address).

- Weighted Routing: Distributes traffic across multiple resources based on assigned weights.

- Latency-based Routing: Directs traffic to the region with the lowest latency for the user.

- Geolocation Routing: Routes traffic based on the geographic location of users.

- Failover Routing: Automatically redirects traffic to a backup resource if the primary resource fails.

- Multivalue Answer Routing: Returns multiple IP addresses in response to DNS queries, enabling load balancing.

Steps to Connect an IP Address to a Domain Using Route 53:

1) **Access Route 53:** Go to the AWS Management Console and search for Route 53 in the services menu.

2) **Create a Hosted Zone:** Click on "Create Hosted Zone," enter the domain name you purchased, and provide an optional description.

3) **Select Type:** Choose "Public" for the hosted zone type.

4) **Create Record Set**: Click "Create Record Set" to add DNS records.

5) **Root Domain Record:** Leave the root domain empty, enter the IP address in the value box, and click "Create Record."

6) **Create CNAME Record:** For the www subdomain, create a new record with the name "www," select CNAME as the type, and enter your domain name (e.g., chandan.com) as the value.

7) **Update Domain Registrar**: Go to the domain registrar where you purchased the domain, navigate to the DNS management section, and assign the AWS Route 53 name servers (NS) to your domain.

8) **Final Steps:** Confirm all settings are correct, and the domain should now point to your server.