Task 1: Understanding DNS

Objective: Learn how DNS resolves domain names into IP addresses.

Steps:

1. Open a terminal or command prompt on your computer.
2. Use the `nslookup` command to find the IP address of a website:
   - Example: `nslookup www.example.com`
3. Record the IP address that is returned.
4. Try the same with a few different domain names (e.g., `www.google.com`, `www.amazon.com`).
5. Research what DNS servers your computer is using. On Windows, use the command `ipconfig /all`. On macOS/Linux, use `cat /etc/resolv.conf`.

Question: What is the role of a DNS server in resolving domain names? How does this help you access websites?

Task 2: Basic HTTP and HTTPS Requests

Objective: Understand the difference between HTTP and HTTPS by making simple requests.

Steps:

1. Open a browser and enter the URL `http://www.example.com` and `https://www.example.com`.
2. Notice the difference in the browser's URL bar (HTTP vs. HTTPS).
3. Use an online tool like https://www.httpstatus.io/ to check the response status code of a few websites.
   - Example: Try `https://www.example.com` and note the HTTP status code (e.g., 200 OK, 301 Moved Permanently).

Questions:

- What is the main difference between HTTP and HTTPS in terms of security?
- Why does a website use HTTPS instead of HTTP, and what role does SSL/TLS play?

Task 3: Understanding IP Addressing

Objective: Get familiar with IP addresses, IPv4, and IPv6.

Steps:

1. Open a terminal or command prompt.
2. Run the command to check your device's IP address:
   - On Windows: `ipconfig`
   - On macOS/Linux: `ifconfig` or `ip a`
3. Write down your IP address (it should be something like `192.168.1.x` for IPv4 or a longer string for IPv6).
4. Visit a website like https://whatismyipaddress.com to find your public IP address.

Question: What is the difference between IPv4 and IPv6? Why is IPv6 necessary?

Task 4: Exploring Subnets

Objective: Understand the concept of subnets and how to calculate subnet masks.

Steps:

1. Read about subnetting (you can find tutorials online like this one).
2. Given an IP address `192.168.1.0` with a subnet mask of `255.255.255.0`, calculate the range of available IP addresses in that subnet.
3. Use an online subnet calculator to verify your answer (e.g., this tool).

Question: How does subnetting help improve network security and efficiency?

Task 5: Testing a REST API Request

Objective: Make a basic HTTP request to a REST API using a tool like `curl` or Postman.

Steps:

1. Open Postman or use `curl` in your terminal.
2. Try making a `GET` request to a public API. For example:
   - `https://jsonplaceholder.typicode.com/users`
3. Review the JSON response returned from the server. This should include data about users.
4. Experiment with different HTTP methods:
   - **GET**: Retrieve data.
   - **POST**: Create new data (use Postman to create a new user).
   - **PUT**: Update data.
   - **DELETE**: Delete a resource.

Question: How does REST differ from traditional web services? What are the key HTTP methods used in REST?

Task 6: REST API Authentication

Objective: Learn the basics of different authentication methods used in REST APIs.

Steps:

1. Explore an API that requires authentication, such as the GitHub API.
2. Go to GitHub's Authentication documentation.
3. Create a GitHub account (if you don't already have one), and generate a **Personal Access Token** for authentication.
4. Use Postman or `curl` to make a request to the GitHub API with your token:
   - Example: `curl -H "Authorization: token YOUR_ACCESS_TOKEN" https://api.github.com/user`
5. Compare Basic Authentication with OAuth. Research the steps involved in generating an OAuth token.

Question: How does OAuth differ from Basic Authentication in terms of security? What is the advantage of using tokens like JWT for authentication?

Task 7: Understanding and Setting Up DHCP

Objective: Learn about DHCP and how it assigns IP addresses dynamically.

Steps:

1. If you're working within a local network (like at home), log into your router's admin panel and find the DHCP settings.
2. Ensure DHCP is enabled on the router.
3. Disconnect and reconnect a device (e.g., laptop or phone) to the network and observe the IP address it is assigned.
4. Take note of the IP address assigned to the device and check the range defined by the DHCP settings.

Question: What is the role of DHCP in a network? How does it help manage IP addresses dynamically?

Task 8: Testing Cookies in Web Browsers

Objective: Understand how cookies work in web browsers.

Steps:

1. Open your browser and go to a website that uses cookies (almost all websites do).
2. Press `F12` (or right-click and select "Inspect") to open the browser's developer tools.
3. Go to the "Application" tab (in Chrome, for example) and find the "Cookies" section.
4. Refresh the page and observe the cookies being set by the website.
5. Modify a cookie value and refresh the page to see if any changes occur (if you understand the specific cookie's purpose).

Question: How do cookies help in session management? What are the security risks associated with cookies, and how can they be mitigated (e.g., using `Secure` and `HttpOnly` flags)?

Task 9: Exploring Routing

Objective: Understand how routing works in networks.

Steps:

1. Use the `traceroute` (or `tracert` on Windows) command to trace the path your request takes from your computer to a website (e.g., `traceroute www.example.com`).
2. Record the different hops along the path and identify any timeouts or errors that occur.
3. Research how routing tables are used to direct traffic between networks.

Question: What happens if a routing issue occurs (e.g., a routing table error)? How do routers determine the best path for data?

Task 10: VLAN Basics

Objective: Learn about Virtual Local Area Networks (VLANs).

Steps:

1. If you have access to a network switch with VLAN support (or a simulation tool like Cisco Packet Tracer), create two VLANs:
   - VLAN 10 (for IT staff).
   - VLAN 20 (for HR staff).
2. Assign different IP subnets to each VLAN (e.g., VLAN 10 = `192.168.10.0/24`, VLAN 20 = `192.168.20.0/24`).
3. Test communication between devices within the same VLAN and between devices on different VLANs.

Question: How do VLANs help in network segmentation? Why is VLAN tagging important in modern networks?

Task 11: Understanding Cloud Service Models (IaaS, PaaS, SaaS)

Objective: Learn the differences between the three main cloud service models: IaaS, PaaS, and SaaS.
Steps:

1. Research the definitions of **IaaS**, **PaaS**, and **SaaS**.
2. For each model, identify at least two real-world examples of services (e.g., AWS EC2 for IaaS, Google App Engine for PaaS, Microsoft Office 365 for SaaS).
3. Write a brief description of the level of control users have in each model.
4. Compare the use cases for each service model. Which model is best for developers, which is best for businesses, and which is best for end users?
5. Create a table summarizing the differences between IaaS, PaaS, and SaaS (e.g., control, management responsibilities, common use cases).
   Question: What are the key differences between IaaS, PaaS, and SaaS, and in what scenarios is each service model most appropriate?

Task 12: Choosing a Cloud Deployment Model (Public, Private, Hybrid)

Objective: Understand the different cloud deployment models and how to choose the best one for an organization.
Steps:

1. Research the three primary cloud deployment models: **Public**, **Private**, and **Hybrid**.
2. Write a brief explanation of each model, highlighting the key characteristics and differences.
3. Develop a scenario for a business (e.g., a small business, a large corporation, or an enterprise with sensitive data needs).
4. Based on the business scenario, choose the most suitable cloud deployment model and explain why you chose it (consider cost, security, and scalability).
5. Create a decision matrix that compares the pros and cons of each deployment model for this business.
   Question: How do you decide whether to use a public, private, or hybrid cloud deployment model? What factors impact your decision?

Task 13: Configuring Auto-Scaling for a Web Application on AWS

Objective: Learn how to set up auto-scaling for a web application on AWS.
Steps:

1. Launch a simple web application on AWS using **EC2 instances** (e.g., an Apache server or a basic PHP application).
2. Set up an **Auto Scaling Group** in AWS to automatically scale your application based on traffic (use metrics like CPU utilization or incoming traffic).
3. Set scaling thresholds (e.g., scale up when CPU utilization reaches 70%, scale down when it drops below 30%).
4. Monitor the performance of your application as traffic increases and decreases to verify that auto-scaling is working as expected.
5. Record the number of instances that are launched or terminated as part of the scaling process.
   Question: How does auto-scaling improve the performance and cost efficiency of cloud-based applications? How does AWS handle scaling automatically?