Task: Networking Setup and Configuration
Description:
Configure networking components such as routers, switches, firewalls, and load balancers to ensure reliable communication and security within the infrastructure.
Test Cases:
1. Router Configuration:
Test Case 1: Configure routing protocols (e.g., OSPF, BGP) on routers to enable dynamic routing.
Expected Outcome: Routers should be able to exchange routing information and dynamically update routing tables.

Router Configuration:

- Access the router's configuration interface (usually via a web browser or SSH).

- Configure routing protocols such as OSPF or BGP by specifying neighbors, areas, and authentication if needed.

OSPF Configuration:

1. Access Router Configuration Mode:

```
enable
configure terminal
```

2.Enable OSPF:

```
router ospf [process-id]
```

3.Define OSPF Router ID (optional):

```
router-id [router-id]
```

4.Assign OSPF Areas:

```
network [network-address] [wildcard-mask] area [area-id]
```

5.Enable OSPF on Interfaces:

```
interface [interface-type] [interface-number]
ip ospf [process-id] area [area-id]
```

6. Exit Configuration Mode and Save Configuration:

```
end
write memory
```

BGP Configuration:
1.Access Router Configuration Mode:
Enable
configure terminal

2.Enable BGP:
router bgp [AS-number]

3.Establish Neighbors (Peers):
neighbor [neighbor-ip-address] remote-as [remote-AS-number]

4.Specify Networks to Advertise:
network [network-address] mask [network-mask]

5.Configure BGP Timers (optional):
timers bgp [keepalive-time] [hold-time]

6.Exit Configuration Mode and Save Configuration:
End
write memory

- Verify the router's interfaces are enabled for routing and participating in the chosen routing protocol.
- Ensure proper subnetting and network addressing to facilitate routing.
- Monitor routing tables for dynamic updates and troubleshoot any connectivity issues.

2. Firewall Rules Setup:
Test Case 2: Define and implement firewall rules to restrict incoming and outgoing traffic based on security policies.
Expected Outcome: Traffic should be filtered according to the defined firewall rules, allowing only authorized communication.
Firewall Rules Setup:

Access Firewall Configuration Mode:
enable
configure terminal

Define Access Control Lists (ACLs):

- Define ACLs to permit or deny specific traffic based on criteria such as source IP, destination IP, port numbers, and protocols.

Example:
access-list <acl_number> permit tcp any host <server_ip> eq <port_number>
access-list <acl_number> deny ip any any

Apply ACLs to Interfaces:

- Apply the access control lists to specific interfaces where traffic filtering is required.

Example:

interface GigabitEthernet0/0
ip access-group <acl_number> in

Enable Firewall Features:

- Enable specific firewall features such as stateful packet inspection, intrusion prevention, or application layer filtering based on your security requirements.

Example:
firewall
firewall module <module_number> <feature>
Inspect Traffic:

- Configure the firewall to inspect traffic based on protocols and applications to ensure that only authorized traffic is allowed.

Example:

class-map inspection_default
match default-inspection-traffic

Apply NAT (Network Address Translation) if Required:

- If you're using NAT, configure it to translate internal/private IP addresses to external/public IP addresses and vice versa.

Example:
nat (inside,outside) source dynamic any interface

- Exit Configuration Mode and Save Configuration:

end
write memory

- Define inbound and outbound firewall rules based on security policies, considering factors like source/destination IP, port numbers, and protocols.
- Implement rules to allow necessary traffic (e.g., HTTP, HTTPS, SSH) and deny unauthorized access.

- Test the firewall rules by simulating different types of traffic and ensuring they are enforced correctly.
- Regularly review and update firewall rules to adapt to changing security requirements.

3. Load Balancer Configuration:
Test Case 3: Configure a load balancer to distribute incoming traffic across multiple servers in a backend pool.
Expected Outcome: The load balancer should evenly distribute traffic among the backend servers, ensuring high availability and scalability.

Load Balancer Configuration:

- Access the load balancer's configuration interface (often via a web browser or command line interface).
-
- Install and Configure Load Balancer:
    - Install NGINX or any other load balancer software on a dedicated server or virtual machine.
    - Configure NGINX to act as a reverse proxy to distribute traffic to the backend servers.
- Define Backend Servers:
    - Identify the backend servers that will receive traffic from the load balancer.
    - Ensure that the backend servers are properly configured and running the necessary services (e.g., web servers, application servers).
- Configure NGINX as a Load Balancer:
    - Edit the NGINX configuration file (usually located at /etc/nginx/nginx.conf or /etc/nginx/sites-available/default).
    - Define an upstream block that specifies the backend servers:

```
upstream backend {
 server backend1.example.com;
 server backend2.example.com;
 server backend3.example.com;
}
```

Here, backend1.example.com, backend2.example.com, etc., are placeholders for the actual IP addresses or domain names of your backend servers.

- Define a backend pool consisting of multiple servers that will receive traffic.

- Configure load balancing algorithms (e.g., round-robin, least connections) to distribute incoming requests evenly among the backend servers.

Example

```
server {
   listen 80;
   location / {
      proxy_pass http://backend;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header X-Forwarded-Proto $scheme;
   }
}
```

- Set up health checks to monitor the availability and health of backend servers, removing unhealthy servers from the pool if necessary.

Test the Load Balancer:

- After configuring NGINX, test the load balancer by accessing the website or service through the load balancer's IP address or domain name.
- Monitor the traffic distribution among the backend servers to ensure that the load balancer is functioning correctly.
- Monitor and Scale as Needed:
    - Monitor the performance of the backend servers and the load balancer.
    - Scale the backend server infrastructure horizontally by adding more servers to the backend pool as traffic increases.
- Implement High Availability:
    - Configure redundancy and failover mechanisms to ensure high availability of the load balancer itself.
    - Consider using multiple load balancer instances in an active-passive or active-active configuration.