Using Netkit, implement the network depicted in the figure and described below (you can use the following as a checklist).

**Routing**
- Remember to set up a default route where required.
- AS100 is a transit AS whose internal routing is implemented by using RIP; AS100r1 and AS100r3 redistribute BGP and loopback interface addresses in RIP.
- Routing within AS4 is implemented by using OSPF (all interfaces belong to area 0.0.0.0); interface costs on as4r1 and as4r3 are assigned as indicated.
- No routers announce 0.0.0.0/0 or IPv6 subnets.
- All peering LANs are announced. AS1, AS3, AS4, and AS20 also announce their own subnets (in gray).
- AS100’s border routers establish an iBGP peering using their loopback interfaces (use `update-source IPADDRESS`).
- AS10 is a customer of AS1 and AS2 and, as such, forbids all transit traffic.
- AS20 is a customer of AS100 and a peer of AS4. As such, it forbids all transit traffic and prefers using link A.
- AS4 is a customer of AS1, a peer of AS20, and a provider of AS3. As such, it forbids any transit traffic between AS1 and AS20, by suitably applying the following two access-lists to outgoing announcements:

```bash
ip as-path access-list announcementsForAS20 deny AS1
ip as-path access-list announcementsForAS20 permit *
neighbor AS1 neighborIP filter-list announcementsForAS20 out
deny AS1 neighborIP filter-list announcementsForAS20 out

ip as-path access-list announcementsForAS20 deny AS3
ip as-path access-list announcementsForAS20 permit *
neighbor AS3 neighborIP filter-list announcementsForAS20 out
deny AS3 neighborIP filter-list announcementsForAS20 out
```

**BGP**
- IPv6 forwarding:
  - Redistribution BGP in an IGP:
    - `redistribute bgp route-map routemapName` route-map `route-mapName` permit 10
    - `match ip next-hop prefix-listName prefix-listName` ip prefix-list `prefix-listName` permit next-hop/32
  - Setting up (adding) an IPv6 address:
    - `ip interface up` ipconfig `INTERFACE` `interface address` /`NETMASK`
  - Configuring a static IPv6 route:
    - `ip route` route `A inet6 add` `INET6` /`NETMASK` `DEV` `DEV`
  - Creating an IPv6-in-IPv4 tunnel:
    - `ip tunnel add tunname mode sit remote REMOTEIP4 local LOCALIP4 ttl 10`
    - `ipconfig tunname up` ipconfig `TUNNAM` `TUNNAM` `TUNNAM`
  - IPv6 TraceRoute:
    - `traceroute -n 1 IPv6Address`
  - Enabling bind on IPv6 (in NAME.CONFIG):`options {-listen-on-v6 {::0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0;}};

**DNS**
- Web1, web2, and web3 are Web servers that run Apache and serve a default web page, with different contents for each server.
- as4r1 is a layer 4 web switch with VIP 4.0.0.43, with a round robin policy implemented by the following configuration:
  - `iptables -t nat -A PREROUTING -d 4.0.0.43 -m statistic --mode nh --every 2 --jump DNAT --to-destination 192.168.0.2`
  - `iptables -t nat -A PREROUTING -d 4.0.0.43 --jump DNAT --to-destination 192.168.0.3`
- as10r1 is the local name server for pc1 (reached over IPv6 – enable bind on IPv6); as20r1 is the local name server for pc2.
- root-ns-1 and root-ns-2 are root name servers with anycast address 5.5.5.5; as4r2 is the authority for `com` (pick one of its IP addresses as name server address); as4r3 is the authority for shopping.com (pick one of its IP addresses as name server address).
- `p1c1.shop.com` is mapped to pc1’s IPv6 address; `p2c2.shop.com` is mapped to pc2’s IPv4 and IPv6 addresses; on name `www.shop.com` is implemented a round-robin load balancing between AS4’s farm (4.0.0.43) and AS3’s farm (3.0.0.2).

**IPv6**
- Enable IPv6 forwarding on network devices that act as IPv6 routers.
- IPv6 routing is implemented by static routes.
- An IPv6-in-IPv4 tunnel is established between as10r1’s interface eth2 and as20r1’s interface eth1.