Using Netkit, implement the network depicted in the figure below. In doing so, comply with the specifications and achieve the goal indicated below the figure.

**Specifications:**
- Internal IPv4 routing within **AS2** must be implemented using RIP.
- **AS2** announces the network prefix indicated in the dark grey box.
- When required, a node must be enabled to act as an IPv6 router by using the command specified in the box alongside.
- No router announces IPv6 subnets in RIP or BGP.
- No router announces the default route 0/0.
- No router implements customer-provider routing policies.
- No router implements filters that drop BGP announcements.

**USEFUL COMMANDS** (arguments in square brackets are optional):
- Assign the IPv6 address `ipv6addr/mask` to interface `if`.
  `ifconfig if up
  ifconfig if add ipv6addr/mask`
- Enable a network node to act as an IPv6 router:
  `echo 1 > /proc/sys/net/ipv6/conf/all/forwarding`
- Add a static route towards `ipv6addr/[mask]`:
  `route -A inet6 add ipv6addr[/mask] [gw nexthop] [dev interface]`
- Setup of an IPv6-in-IPv4 tunnel called `tunnelName` between `ipv4LocalAddr` and `ipv4RemoteAddr` (note: this setup must be applied to both tunnel endpoints):
  `ip tunnel add tunnelName mode sit remote ipv4RemoteAddr local ipv4LocalAddr ttl 10`
  `ifconfig tunnelName up`
  `ifconfig tunnelName add ipv6LocalAddr` route `-A inet6 add ipv6RemoteAddr dev tunnelName`

**Goal:**
It must be possible to ping **pc2** from **pc1** (and vice versa) in IPv6 using command `ping6`. 
Using Netkit, implement the network depicted in the figure below. In doing so, comply with the specifications and achieve the goal indicated below the figure.

**Specifications:**
- Internal IPv6 routing within **AS20** must be implemented using static routes.
- When required, a node must be enabled to act as an IPv6 router by using the command specified in the box alongside.
- No router announces IPv6 subnets in RIP or BGP.
- No router announces the default route 0/0.
- No router implements customer-provider routing policies.
- No router implements filters that drop BGP announcements.

**Goal:**
It must be possible to ping **pc2** from **pc1** (and vice versa) in IPv6 using command **ping6**.

**USEFUL COMMANDS** (arguments in square brackets are optional):
- Assign the IPv6 address **ipv6addr/mask** to interface **if**:
  - `ifconfig if up
  - ifconfig if add ipv6addr/mask`
- Enable a network node to act as an IPv6 router:
  - `echo 1 >/proc/sys/net/ipv6/conf/all/forwarding`
- Add a static route towards **ipv6addr[/mask]**:
  - `route -A inet6 add ipv6addr[/mask] [gw nexthop] [dev interface]`
- Setup of an IPv6-in-IPv4 tunnel called **tunnelName** between **ipv4LocalAddr** and **ipv4RemoteAddr** (note: this setup must be applied to both tunnel endpoints):
  - `ip tunnel add tunnelName mode sit remote ipv4RemoteAddr local ipv4LocalAddr ttl 10
  - ifconfig tunnelName up
  - ifconfig tunnelName add ipv6LocalAddr
  - route -A inet6 add ipv6RemoteAddr dev tunnelName`
Using Netkit, implement the network depicted in the figure below. In doing so, comply with the specifications and achieve the goal indicated below the figure.

**Specifications:**
- Internal IPv4 routing within **AS300** must be implemented using RIP.
- **AS300** routers must not use **redistribute bgp**.
- When required, a node must be enabled to act as an IPv6 router by using the command specified in the box alongside.
- No router announces IPv6 subnets in RIP or BGP.
- No router announces the default route 0/0.
- No router implements customer-provider routing policies.
- No router implements filters that drop BGP announcements.

**USEFUL COMMANDS** (arguments in square brackets are optional):
- Assign the IPv6 address `ipv6addr/mask` to interface `if`:
  - `ifconfig if up`
  - `ifconfig if add ipv6addr/mask`
- Enable a network node to act as an IPv6 router:
  - `echo 1 \>/proc/sys/net/ipv6/conf/all/forwarding`
- Add a static route towards `ipv6addr[/mask]`:
  - `route \-A inet6 add ipv6addr[/mask] [gw nexthop] [dev interface]`
- Setup of an IPv6-in-IPv4 tunnel called `tunnelName` between `ipv4LocalAddr` and `ipv4RemoteAddr` (note: this setup must be applied to both tunnel endpoints):
  - `ip tunnel add tunnelName mode sit remote ipv4RemoteAddr local ipv4LocalAddr ttl 10`
  - `ifconfig tunnelName up`
  - `ifconfig tunnelName add ipv6LocalAddr dev tunnelName`

**Goal:**
It must be possible to ping `pc2` from `pc1` (and vice versa) in IPv6 using command `ping6`.
Using Netkit, implement the network depicted in the figure below. In doing so, comply with the specifications and achieve the goal indicated below the figure.

Specifications:
- Internal IPv4 routing within **AS1000** and **AS2000** must be implemented using RIP.
- **AS1000** and **AS2000** announce the network prefix in the dark grey box.
- When required, a node must be enabled to act as an IPv6 router by using the command specified in the box alongside.
- No router announces IPv6 subnets in RIP or BGP.
- No router announces the default route 0/0.
- No router implements customer-provider routing policies.
- No router implements filters that drop BGP announcements.

Goal:
It must be possible to ping **pc2** from **pc1** (and vice versa) in IPv6 using command **ping6**.

**USEFUL COMMANDS** (arguments in square brackets are optional):
- Assign the IPv6 address `ipv6addr/mask` to interface `if`:
  - `ifconfig if up`
  - `ifconfig if add ipv6addr/mask`
- Enable a network node to act as an IPv6 router:
  - `echo 1 > /proc/sys/net/ipv6/conf/all/forwarding`
- Add a static route towards `ipv6addr[/mask]`:
  - `route -A inet6 add ipv6addr[/mask] [gw nexthop] [dev interface]`
- Setup of an IPv6-in-IPv4 tunnel called `tunnelName` between `ipv4LocalAddr` and `ipv4RemoteAddr` (note: this setup must be applied to both tunnel endpoints):
  - `ip tunnel add tunnelName mode sit remote ipv4RemoteAddr local ipv4LocalAddr ttl 10`
  - `ifconfig tunnelName up`
  - `ifconfig tunnelName add ipv6LocalAddr`
  - `route -A inet6 add ipv6RemoteAddr dev tunnelName`