### Things to know:

- What is Border Gateway Protocols (BGP)?
  - iBGP?
  - eBGP
- What are Autonomous Systems (AS)?
- What is the decision process for AS paths? (See slide 23)

### For breakdown of the commands used in this section - refer to:
[https://netref.soe.ucsc.edu/osnl/13](https://netref.soe.ucsc.edu/osnl/13)

### Show Command:
```
show ip bgp
```

### Not covered in section:

“Additional fun will be had in sections with community-strings and path modification…”

Using community:

Hint:
You can **Deny** a path just as you can **permit** a path.
# Topology Setup:

## Assign IP to all interface

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
</tr>
</thead>
</table>
| conf t  
int fa0/0  
ip add 10.0.1.1  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.2.1  
255.255.255.0  
no shut  
int fa2/0  
ip add 10.0.3.1  
255.255.255.0  
no shut  
end | conf t  
int fa0/0  
ip add 10.0.1.2  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.4.2  
255.255.255.0  
no shut  
end | conf t  
int fa0/0  
ip add 10.0.2.3  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.5.3  
255.255.255.0  
no shut  
end | conf t  
int fa0/0  
ip add 10.0.3.4  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.7.4  
255.255.255.0  
no shut  
end |

<table>
<thead>
<tr>
<th>R5</th>
<th>R6</th>
<th>R7</th>
</tr>
</thead>
</table>
| conf t  
int fa0/0  
ip add 10.0.4.5  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.5.5  
255.255.255.0  
no shut  
int fa2/0  
ip add 10.0.8.5  
255.255.255.0  
no shut  
end | conf t  
int fa0/0  
ip add 10.0.6.6  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.7.6  
255.255.255.0  
no shut  
int fa2/0  
ip add 10.0.9.6  
255.255.255.0  
no shut  
end | conf t  
int fa0/0  
ip add 10.0.8.7  
255.255.255.0  
no shut  
int fa1/0  
ip add 10.0.9.7  
255.255.255.0  
no shut  
int fa2/0  
ip add 10.0.10.7  
255.255.255.0  
no shut  
end |

### ip <address>/<prefix> <default gateway>

<table>
<thead>
<tr>
<th>VPC1</th>
<th>VPC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip 10.0.0.11/24 10.0.0.1</td>
<td>ip 10.0.10.12/24 10.0.10.7</td>
</tr>
</tbody>
</table>
### BGP Setup:

<table>
<thead>
<tr>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
</tr>
</thead>
</table>
| conf t  
router bgp 100  
neighbor 10.0.1.2 remote-as 200  
neighbor 10.0.2.3 remote-as 300  
neighbor 10.0.3.4 remote-as 400  
network 10.0.0.0 mask 255.255.255.0 | conf t  
router bgp 200  
neighbor 10.0.1.1 remote-as 100  
neighbor 10.0.4.5 remote-as 500 | conf t  
router bgp 300  
neighbor 10.0.2.1 remote-as 100  
neighbor 10.0.5.5 remote-as 500  
neighbor 10.0.6.6 remote-as 600 | conf t  
router bgp 400  
neighbor 10.0.3.1 remote-as 100  
neighbor 10.0.7.6 remote-as 600 |

<table>
<thead>
<tr>
<th>R5</th>
<th>R6</th>
<th>R7</th>
</tr>
</thead>
</table>
| conf t  
router bgp 500  
neighbor 10.0.4.2 remote-as 200  
neighbor 10.0.5.3 remote-as 300  
neighbor 10.0.8.7 remote-as 700 | conf t  
router bgp 600  
neighbor 10.0.6.3 remote-as 300  
neighbor 10.0.7.4 remote-as 400  
neighbor 10.0.9.7 remote-as 700 | conf t  
router bgp 700  
neighbor 10.0.8.5 remote-as 500  
neighbor 10.0.9.6 remote-as 600  
network 10.0.10.0 mask 255.255.255.0 |

BGP is set up. You will need to wait a few minutes for routes to propagate, and use **clear ip bgp * or clear bgp *** to restart bgp process and update route tables.

### Check BGP table

```
R1# show ip bgp
BGP table version is 3, local router ID is 10.0.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network     Next Hop      Metric LocPrf Weight Path
*> 10.0.0.0/24  0.0.0.0        0     32768 i
  10.0.3.4        0       400 600 700 i
  10.0.2.3        0       300 500 700 i
*> 10.0.1.2        0        200 500 700 i
R1#
```
Discussion:
What are there blanks below network 10.0.10.0? What does this output mean?

The network states the different subnet that are connected. There are blanks under 10.0.10.0 because the routes listed below it are all paths to the 10.0.0.10 network

‘*’ denotes preferred path

Issue a traceroute from VPC1 to VPC2 to check if the route was taken

Modifying routes!

1. Let's start with having the route from C1 (VPC1) to C2 (VPC2) go through R2 and then R5 to R7, while the route from C2 to C1 goes to R6, R4, then R1. We will be modifying the local preferences of these paths.

<table>
<thead>
<tr>
<th>R1</th>
<th>R7</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf t route-map C1-C2-through-R2 permit 10 set local-pref 50</td>
<td>conf t route-map C2-C1-through-R6 permit 10 set local-pref 50</td>
</tr>
<tr>
<td>exit</td>
<td>exit</td>
</tr>
<tr>
<td>route-map bogus permit 10 set local-pref 1 exit</td>
<td>route-map bogus permit 10 set local-pref 1 exit</td>
</tr>
<tr>
<td>router bgp 100 neighbor 10.0.1.2 route-map C1-C2-through-R2 in</td>
<td>router bgp 700 neighbor 10.0.9.6 route-map C2-C1-through-R6 in</td>
</tr>
<tr>
<td>neighbor 10.0.2.3 route-map bogus in neighbor 10.0.3.4 route-map</td>
<td>neighbor 10.0.8.5 route-map bogus in end</td>
</tr>
<tr>
<td>bogus in end clear bgp *</td>
<td>clear bgp *</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.8/24</td>
<td>0.0.0.0</td>
<td>32768</td>
<td>0</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>10.0.10.0/24</td>
<td>10.0.1.2</td>
<td>50</td>
<td>0</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>10.0.2.3</td>
<td>10.0.3.4</td>
<td>1</td>
<td>0</td>
<td>300</td>
<td>500</td>
</tr>
<tr>
<td>10.0.3.4</td>
<td></td>
<td>1</td>
<td>0</td>
<td>400</td>
<td>500</td>
</tr>
</tbody>
</table>
Great! But let's say that we actually want our traffic to pass through R3. We can modify R2 and R4's path length.

set [community aa:nn | ip NEXTHOP] local-preference # | as-path prepend AS-LIST

<table>
<thead>
<tr>
<th>R2</th>
<th>R4</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf t</td>
<td>conf t</td>
</tr>
<tr>
<td>route-map extend permit 10</td>
<td>route-map extend permit 10</td>
</tr>
<tr>
<td>set as-path prepend 200 200 200 200</td>
<td>set as-path prepend 400 400 400 400</td>
</tr>
<tr>
<td>exit</td>
<td>exit</td>
</tr>
<tr>
<td>router bgp 200</td>
<td>router bgp 400</td>
</tr>
<tr>
<td>neighbor 10.0.1.1 route-map extend out</td>
<td>neighbor 10.0.3.1 route-map extend out</td>
</tr>
<tr>
<td>end</td>
<td>end</td>
</tr>
<tr>
<td>clear bgp *</td>
<td>clear bgp *</td>
</tr>
</tbody>
</table>

At this point, notice how the local-preference value is still overwriting the path length for path selection. So we need to undo local pref for R1.

R1

conf t
router bgp 100
no neighbor 10.0.2.3 route-map bogus in
no neighbor 10.0.3.4 route-map bogus in
no neighbor 10.0.1.2 route-map C1-C2-through-R2 in
end
clear bgp *

The shorter path is now the best path without the bias of Local-pref settings.