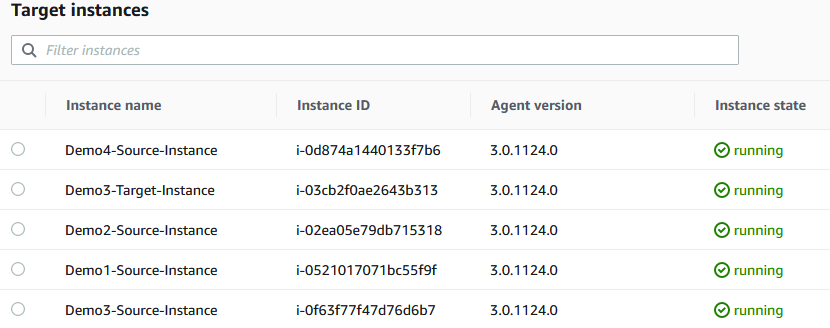
**Chasing Packets Instructor Cheat Sheet**

1. The environment utilizes 6 VPC’s but the default VPC’s per account quota is 5.
   1. Delete the default VPC
   2. But in a service quota request for a total of 6 VPC’s. it generally takes less than an hour to get this service limit increased.
2. Do not deploy the CloudFormation template until 1a and 1b are complete.
3. Copy the template YAML from the following link - <https://github.com/GH-JeffJ/Arch-7-Assets/blob/main/TIPs-Chasing-Packets.yaml>
   1. If you receive an error deploying by uploading the template (happened to me once), select to deploy using designer and paste the yaml in vs. uploading the template.
   2. Deploy the template from us-east-1 (Northern Virginia).
   3. Internal IP addresses for all ec2 instances are provided in the Outputs of the CloudFormation Template
   4. Instances are named by scenario and contain source or target in the name. For Demo purposes, we will always log in to the source instance and attempt to communicate to the target instance to prove/disprove the problem and solution identified through the VPC Reachability Analyzer.
4. Once the infrastructure is deployed, go to Session Manager and confirm you have five instances available to establish a session with. This is not required but is a good sanity check to make sure the instances used for troubleshooting as part of the lesson are accessible.  
   
5. The session presents four different scenarios each with isolated and separate infrastructure
   1. Demo1 – A single VPC with a single subnet and two ec2 instances that cannot communicate with one another over port 80.
   2. Demo2 – A single VPC with two public and one private subnet and two ec2 instances that cannot communicate with one another over port 80
   3. Demo3 – Two VPC’s with one public subnet and one ec2 instance per VPC. The VPC’s are connected using a peering connection and the two ec2 instances are unable to communicate with each other over port 80
   4. Demo4 – Two VPC’s, a total of three subnets (one public and two private), and a single ec2 instance. In this demo, the VPC’s are connected through a Transit Gateway. The VPC with the ec2 instance has no internet connectivity within its VPC and utilizes the NAT Gateway and Internet Gateway in the “Shared Services” VPC via the Transit Gateway to reach the internet.
6. What is broke in each scenario and what is the recommended approach to running the scenario?
   1. Demo1 **(One security group)** – This is a single subnet so not much we can break other than a Security Group.
      1. Log in to Demo1-Source using Session Manager and try to ping and/or curl the IP of the target instance.
      2. Ask what is required for a ping to work (ICMP) and ask what we need to check in order to verify connectivity.
      3. Discuss how long does it take to go to each ec2 instance, look up the subnet, look up the security group, compare rules to see if something is missing while going through the process.
      4. Go to the VPC Reachability Analyzer and analyze the path from the Demo1-Source to the Demo1-Target.
         1. Fix the security group issue and re-run the path analysis
         2. Switch back over to Demo1-Source and run a ping and curl again against the instance. Rather than timing out, curl should reject the connection. This confirms port 80 is available but has no content to return. This is because the instance does not have internet access and was unable to pull down the package from GitHub (if anyone asks the question).
   2. Demo2 **(One Network ACL and one SG)** – We have multiple subnets in this scenario so we will introduce an error in a Network ACL and an error in a Security Group.
      1. Log in to Demo2-Source using Session Manager to ping/curl/traceroute to Demo2-Target and see what we can learn.
         1. Discuss the various areas where we can have a fault and how we could attempt to isolate using ping, ifconfig to compare source and destination IP addresses (are they in the same subnet or different subnets).
      2. From the AWS perspective, how many different objects do we need to check? Two security groups, two or more subnets, multiple route tables, all while forming a mental picture of the traffic flow.
      3. As with 6.a.iv, run Reachability Analyzer from Demo2-Source to Demo2-Target, fix any errors. You can also execute a test from the Demo2-target instance to the eni of Demo2-natgw and from Demo2-target to Demo2-igw. Demo2-igw will give you some errors implying that the instance needs to have a public IP **(hint: This is because the route to the natgw is not established yet…)**. Re-run the analysis once the issues are remediated for all three analysis and prove success using Session Manager to test from Demo2-Source to Demo2-Target.
      4. An issue should be identified with the target security group and the network ACL associated with the Demo2-private1-subnet.

**Note:** Since the network ACL did not have an inbound rule, the target instance was unable to install software during bootstrapping. This will result in curl responding on port 80 but refusing the connection like Demo1.

* 1. Demo3 **(Two NACL’s and one SG)** – We have two VPC’s and one subnet per VPC in this demo connected through a peering connection.
     1. Log in to Demo3-Source using Session Manager ping/curl/traceroute to Demo3-Target to see what we can learn. **Also, curl the Public IP address of Demo3-Target.**
        1. You should receive a timeout when curl’ing the private IP address and see a success for the public IP address.
        2. Open up a web browser and paste in the public IP address and the page should load.
        3. Prior to troubleshooting any further, ask the audience for their thoughts.
     2. Discuss how much time it would take to work through the various VPC’s, SG’s, Subnets, Route Tables, NACL’s to isolate the problem all while you are running an analysis through the VPC Reachability Analyzer.
        1. You should see that some NACL’s are configured incorrectly and a route is missing.
        2. Fix those issues by linking directly to the service from the analyzer and re-run the path analysis once the errors are corrected.
        3. Confirm success by running a curl from Demo3-Source to Demo3-Target.
  2. Demo4 **(Nothing broke)** – In this final demo, we are replicating a shared VPC model where VPC’s can only access the internet through a singular VPC.
     1. In this Demo, VPC-a contains a public and private subnet with an IGW and NatGW in the public subnet and the private subnet is attached to a TGW.
     2. VPC-b contains our only ec2 instance and a private subnet. In order to reach the internet, the private subnet is attached to the TGW.
     3. **~~At the present time, the VPC Reachability Analyzer cannot analyze transitive routes through a TGW. At best we can map this out using two path analysis’ – One path from the ec2 instance to the TGW and another path from the TGW to the IGW.~~**

The VPC Reachability Analyzer now functions with TGW but only from inside out. This means that during Demo4, you can run an analysis from the Source Instance to the IGW but the reverse path is not available.

* + 1. ~~For this reason, this environment contains no errors but we can demo the inverse where we introduce various errors in to the environment and observe them through the Reachability Analyzer. Also, breaking and re-establishing routes through the TGW takes a little bit of time to recover.~~

The Demo 4 environment is still built without any errors. After you run an analysis from the instance to the IGW, you will have an excellent talking point regarding all the areas that could have errors and how the VPC Reachability Analyzer is beneficial in defining those areas. Poll the audience and see where we should break the architecture and visualize the error(s) by re-running the analysis. If you inject an error that prevents an instance from talking to the NatGW, expect to see errors stating that the instance doesn’t have a public IP and therefore cant use an IGW.

* + 1. ~~The service team was able to pull a full trace from one of my tests so we can see what a full trace looks like even if the analyzer is unable to do so (included in the presentation).~~
    2. Reference Architecture - https://aws.amazon.com/blogs/networking-and-content-delivery/creating-a-single-internet-exit-point-from-multiple-vpcs-using-aws-transit-gateway/

1. Time will likely be available to demo other faults, feel free to ask the audience what they would like to break in the environment so we can visualize errors through the Reachability Analyzer.

Please reach out to Jeff Jepsen – [jljepsen@amazon.com](mailto:jljepsen@amazon.com) if you have any questions.