AWS NETWORKING – WHAT YOU NEED TO KNOW

For any non-tech folks reading this, AWS (Amazon Web Services) is a remote computing service. It provides a cloud-computing Infrastructure-as-a-Service (IaaS) over the Internet, along with storage, bandwidth, and customized support for application programming interfaces (API).

Amazon packages AWS with scalable and virtually unlimited computing, storage, and bandwidth resources. It's based on a pay-as-you-go subscription pricing model.

Some Common Misconceptions About AWS and IaaS

As strange as it sounds, some people complain that IaaS it either too expensive or too cheap (it's complicated!). Others think cloud-computing isn't secure. Still, others believe that you don't need to know anything about IT to work in the Cloud. And finally (again it's complicated), some believe the Cloud is more reliable than traditional IT infrastructures, while others believe it's less reliable.

Let's clear up these misconceptions.

"The Cloud is less expensive in some situations and more expensive in others." There's a lot of gray areas here. There are soft savings. For example, if you can access an application in the Cloud faster than on premises, how do you quantify this savings? The speed of deployment using the Cloud is instantaneous as opposed to setting up a net-new infrastructure where you need to process purchase orders and buy and install equipment.

Using cloud computing will be less expensive if your organization is using a more agile approach in your development cycles than if you run production up in the Cloud – at least in the long run. This goes for any cloud deployment, not just AWS.

"The Cloud is less secure." The reality is that it has the potential to be as secure or less secure than any on-premises data center. Amazon does a very good job of securing their end of things, as do other cloud providers like Azure or Google. However, there is a due diligence on your side. There are things you must do like set up the proper firewall policies. Some industries employ workloads that aren't meant to be used in the Cloud.

So, the answer here is that it all depends on your needs and the security you apply on your end. You should consider using the Cloud and not let security be the reason that holds you back. Have a conversation with your cloud provider to flesh out the details for your particular requirements and what their capabilities are. There are a lot of security options in the Cloud.

"You don't need to know anything about IT to work in the Cloud." If you want to just use an application in the Cloud, this is true. It's easy to do this. However, if you want to set up an IT infrastructure in the Cloud, you need to know a lot about IT. Don't try this on your own. Bring in an IT professional to help.

"The Cloud is more reliable (or less reliable) than traditional IT infrastructures." When it comes to AWS in general, they have very specific Service Level Agreements (SLAs). You must be careful to read the fine print on these. And, if you develop an application that's poorly built, with no resiliency, when Amazon patches AWS and your app goes down, (sorry) this is your fault, not Amazon's.

Regions & Availability Zones (AZs) For AWS

There are 18 geographic locations (or Regions) where AWS has data centers. If you want to put an IT infrastructure in the Cloud, it needs to go into one of these Regions.

Amazon tries to add a few Regions every year. They will probably have 20 by the end of 2018.

- A Region is a geographic area. These days, a Region has at least 3 Availability Zones (AZ)s
- An AZ is a highly independent infrastructure cluster within a specific Region (a building or buildings)
- Your AZs may not line up with someone else's AZs

Think of an AZ as a cluster of data centers (see below). Each AZ can be a single building or a few buildings. But they will all be separate from one another, with individual power sources, separate fiber paths, etc. So, if one goes down, the Region is still fully functioning. There's very low latency between the AZs in AWS. You can expect 1 to 2 milliseconds between two AZs. And with three, possibly 3 to 4 milliseconds.

AWS Common Items – EC2

Basic Compute Unit – (EC2) Elastic Compute Cloud
Be aware that many services in AWS are really an EC2 "under the covers." Most of the items you'll work within AWS are EC2s. Consider how you purchase, deploy, and consume infrastructures today. Consider a cloud approach when doing so.

One of the reasons businesses enjoy a cloud-computing perspective is a flexibility of how they can gain ways to buy their compute cycle. AWS provides four main ones:

1. On-Demand: You can pick the server you want and pay by the hour (or second) with no commitment or terms.

2. Reserved: You can pick the server you want and then choose a commitment term from 1 to 3 years. You can also choose how you want to pay (upfront, per month, etc.).

3. Spot: eBay for Compute! Bid on what you're willing to pay for the resources you need but be aware you must have a lot of flexibility.

4. Dedicated Hosts: Get dedicated hardware for your instances so you avoid multi-tenancy. This can help with licensing for some solutions such as Oracle.

Identity Access Management (IAM)

This is global service controlling access based on users' accounts. It doesn't reside in any given Region. It's similar in approach to Cisco's RBAC (Role Based Access Control) where you set up users and roles. You don't have to set up accounts and permissions per Region.

Be aware that if you create an account and don't assign a role, users won't be able to do anything! No one can do anything in AWS until they have an account in IAM.

IAM also supports multi-factor authentication.

S3 Simple Storage Service

This is a global object-based storage service. You define permissions in "buckets." It's important to be aware of permissions. Never open an S3 bucket to the public! Anyone can read or write to it if you do this.

S3 comes in three versions:

1. S3 (this is the default)
2. S3-IA (infrequent access)
3. S3-1z-IA (infrequent access only in a single availability zone)

You don't want to store information in IA versions. It's less expensive, but if you need access to it all the time, this won't work for you. It's also not an archiving solution. Amazon has an archiving solution called Glacier. (However, it takes 3 to 5 hours to retrieve something from Glacier.) If you go with the less expensive S3-IA or S3-iz-IA you'll be giving up some reliability and performance.

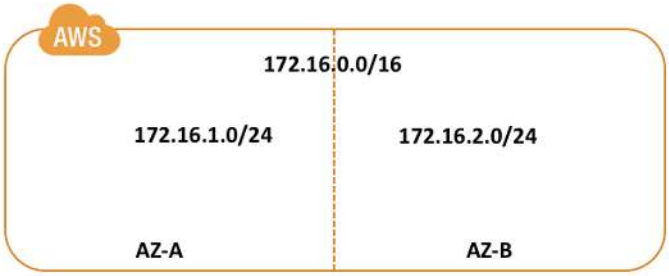
Be aware that once versioning is on, it's never really off again. This can rack up expenses if you're not careful. If you have items that are very large and updated frequently, you shouldn't turn on versioning. It would be better to use snapshotting or something similar (unless you really have a reason to do so).

S3 is for object storage only – It's not for storing your operating system (OS). Block storage is called EBS. Use this for your OS.

AWS Network Basics – VPC Virtual Private Cloud

This is the basic network component in AWS. It's the building block for other services. When you create an IAM it makes a default VPC in every Region. Think of it as your private data center in the Cloud.

- A VPC is akin to your specific Domain Controller (DC) in the Cloud. It's self-contained by default.
- The VPC is a Region-based construct. If you build out a VPC in Oregon and you need to extend it into Virginia, you'll have to build another one in Virginia. (You could build a VPN tunnel between the two if you wanted. But the VPC itself is Region-specific.) Remember that it's tied to your IAM.
- You can use the default VPC in a Region (it's the most commonly used). But you can't/shouldn't delete the default VPC. (An Amazon architect recommended that you not use a default for anything within a VPC.)
- The VPC contains a Classless Inter-Domain Routing (CIDR) block, which is then broken into subnets to allow AZ access.
- There is a VPC router. (You have little ability to interact with it.) You can view it and add static routes, but that's about it.



AWS Network Connectivity – Route Limits

You should be aware of any scalability limits. You cannot have more than 100 routes in a VPC. This means in the VGW (Private Virtual Gateway) routing table OR the VPC routing table.

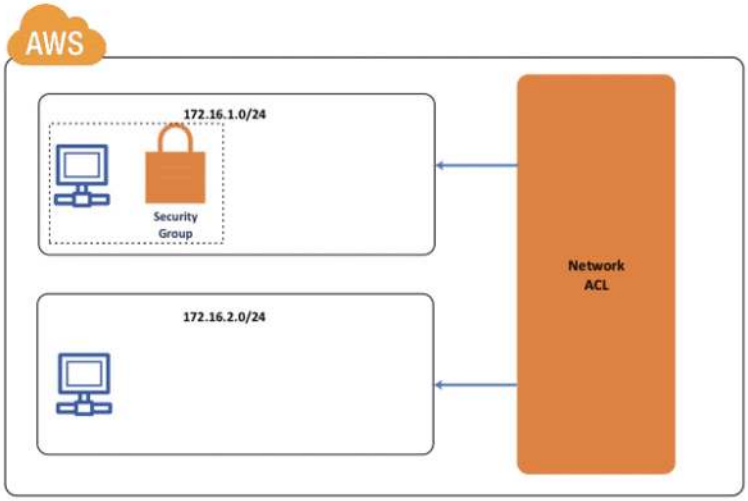
If you try to enter 101 static routes on a VPC routing table, it will block you. If you advertise 101 BGP (Border Gateway Protocol) routes to a VGW, it FAILS the BGP neighbor relationship! By default, you can't see the VGW routing table. This can be changed by turning on "Propagation."

AWS Network Basics – Security Objects

At the VPC level there are two basic security objects:

1. Network ACL's (NACL): These are like a normal ACL (Access Control List) except they are written against a VPC Subnet.
2. Security Groups (SG): These behave more like a FW (Firewall). They are stateful, however they are written for an instance in a VPC. By default, traffic is allowed out of a VPC, but blocked into a VPC. For the NACL you must do this by directional rules.

A security group evaluates all rules before allowing or denying traffic. A NACL evaluates in numerical order and stops after a match. The SG will evaluate all rules. Remember that an object is placed in an SG, where a NACL is placed against a subnet.



AWS Network Connectivity – VGW (Private Virtual Gateway)

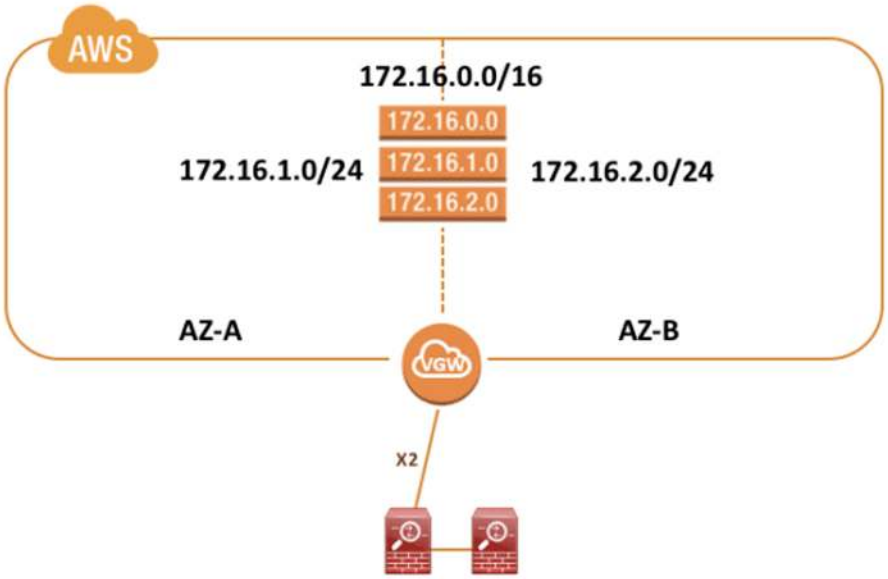
There are two main methods for connecting to your VPC. You could VPN to it or create a private circuit to it. Both of these require a Virtual Gateway in the VPC.

AWS refers to your CPE (Customer Premises Equipment) as a Customer Gateway (CGW). When you connect a CGW to a VPC, it's typically done through a VGW.

If you choose the VPN route (most common) they will build the VPN configuration for you based on your gear. While you might need to make small tweaks, this is actually quite good. Amazon's templates are very accurate and their comments are helpful.

Be aware that with VPN's on AWS, they have very aggressive timeout values. Use an IP-SLA policy if you'd like to keep the tunnel up. Direct Connect circuits don't have the same issue.

From a routing perspective, the VGW supports BGP (Border Gateway Protocol) and Static Routing.



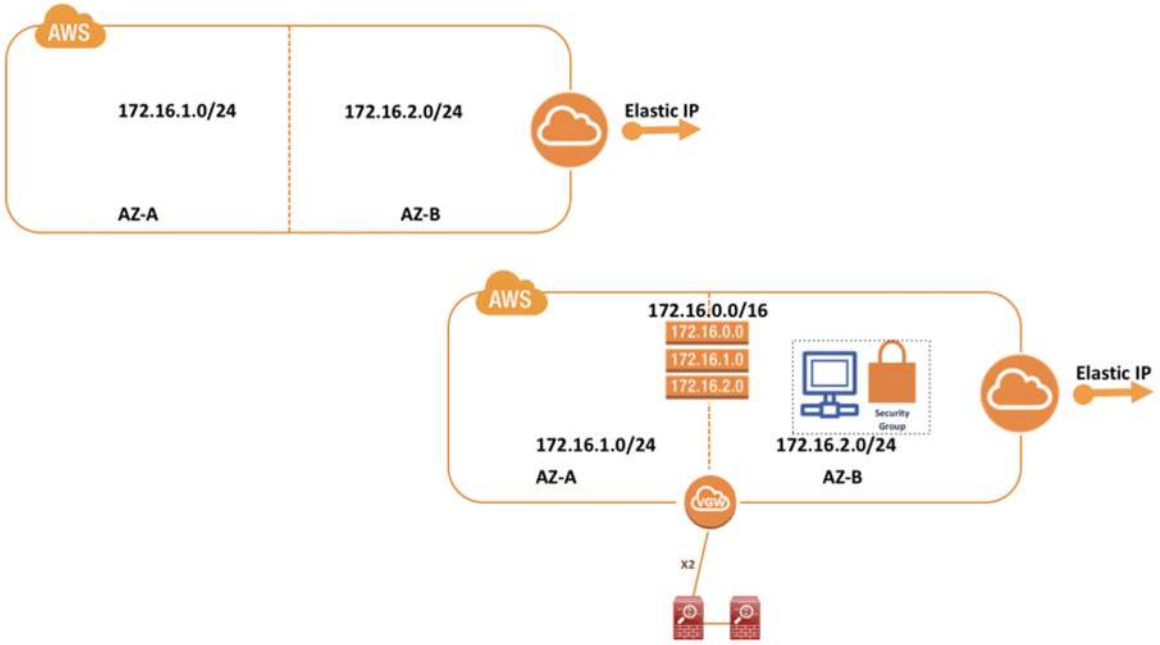
IGW (Internet Gateway)

What the VGW does for private communications, the IGW does for public communications.

The IGW connects to the AWS Internet backbone. While this allows you to get to the Internet, it also lets you get to other resources in your Region or in other Regions.

To utilize an IGW, you will need to configure an Elastic IP. Or you can use a NAT (Network Address Translation) Gateway if you want to do a NAT overload.

Note: All of this isn't required. If you don't want your VPCs to get to the Internet, don't put in the IGWs. Just force-default to come back to the VGW to your on-premise and let it go back to the Internet in this regard. Which way you want to go depends on what problem you're trying to solve.



AWS Tips, Tricks & Thoughts

When you connect to resources outside your VPC, it often requires you to hit the AWS Internet backbone. This means you may need an IGW and an Elastic IP. Carefully consider your applications and how they do HA. Will you use AZs or another method?

Things like IGWs and VGWs are not routers. They are more like routing services. They might look like traditional routers but looks can be deceiving! Be very aware of scalability limits on items in AWS. For example, VPC routing tables scale to a max of 100 routes.

Many network services look like network devices, so don't be fooled!

Need Help? **Contact the AWS Experts at LA Networks. You can reach us by phone at (818) 333-4880 or by email at info@la-networks.com**