Pragmatic Security for Cloud and Hybrid Networks

Version 1.0
October 5, 2015

Securosis, L.L.C. info@securosis.com www.securosis.com

Author's Note

The content in this report was *developed independently of any licensees*. It is based on material originally posted on the Securosis blog and on GitHub, but has been enhanced, reviewed, and professionally edited.

Special thanks to John Moltz for writing and editing support, and Chris Pepper for editing and content support.

Thanks to commenters and contributors on the blog and GitHub for suggestions and review:

jjarava
ejgrossman

This report is licensed by Algosec:

AlgoSec simplifies, automates and orchestrates security policy management to enable enterprise organizations and service providers to manage security at the speed of business. Over 1,500 of the world's leading organizations, including 20 of the Fortune 50, rely on AlgoSec to optimize the network security policy throughout its lifecycle, to accelerate application delivery while ensuring security and compliance. AlgoSec is committed to the success of each and every customer, and provides the industry's only money-back guarantee.

Visit us at algosec.com.

Copyright

Table of Contents

Introduction

Over the last few decades we have been refining our approach to network security. Find the boxes, find the wires connecting them, drop a few security boxes between them in the right spots, and move on. Sure, we continue to advance the state of the art in exactly what those security boxes do, and we constantly improve how we design networks and plug everything together, but overall change has been incremental. How we think about network security doesn't change – just some of the particulars.

Until you move to the cloud.

While many of the fundamentals still apply, cloud computing releases us from the physical limitations of those boxes and wires by fully abstracting the network from the underlying resources. We move into entirely virtual networks, controlled by software and APIs, with very different rules. Things may look the same on the surface, but dig a little deeper and you quickly realize that network security for cloud computing requires a different mindset, different tools, and new fundamentals.

Many of which change every time you switch cloud providers.

The challenge of cloud computing and network security

Cloud networks don't run magically on pixie dust, rainbows, and unicorns – they rely on the same old physical network components we are used to. The key difference is that cloud customers never access the 'real' network or hardware. Instead they work inside virtual constructs – that's the nature of the cloud.

Cloud computing uses *virtual networks* by default. The network your servers and resources see is abstracted from the underlying physical resources. When you server gets IP address 10.0.0.12, that isn't really an IP address on routing hardware – it's a virtual IP address on a virtual network. Everything is handled in software, and most of these virtual networks are Software Defined Networks (SDN). We will explore SDN in depth in the next section.

These networks vary across cloud providers, but they are all fundamentally different from traditional networks in a few key ways:

- **Virtual networks don't provide the same visibility as physical networks** because packets don't move around the same way. We can't plug a box into the network to grab all the traffic – there is no location which all traffic traverses, and much of the traffic is wrapped and encrypted anyway.
- **Cloud networks are managed via Application Programming Interfaces** – not by logging in and provisioning hardware the old-fashioned way. A developer has the power to stand up an entire class B network, completely destroy an entire subnet, or add a network interface to a server and bridge to an entirely different subnet on a different cloud account, all within minutes with a few API calls.

- **Cloud networks change faster than physical networks, and constantly**. It isn't unusual for a cloud application to launch and destroy dozens of servers in an hour – faster than traditional security and network tools can track – or even build and destroy entire networks just for testing.
- **Cloud networks look like traditional networks, but aren't.** Cloud providers tend to give you things that look like routing tables and firewalls, but don't work quite like classic routing tables and firewalls. It is important to know the differences.

Don't worry – the changes make a lot of sense once you start digging in, and most of them provide *better* security that's *more accessible* than on a physical network… so long as you know how to manage them.

The role of hybrid networks

A hybrid network bridges your existing network to your cloud provider. If, for example, you want to connect a cloud application to your existing database, you can connect your physical network to the virtual network in your cloud.

Hybrid networks are extremely common, especially as traditional enterprises begin migrating to cloud computing and need to mix and match resources instead of building everything from scratch. One popular example is setting up big data analytics at your cloud provider, where you only pay for processing and storage time, so you don't need to buy a bunch of servers you will only use once a quarter.

But hybrid networks complicate management, both in your data center and in the cloud. Each side uses a different basic configuration and security controls, so the challenge is to maintain *consistency* across both, even though the tools you use – such as your nifty next generation firewall – might not work the same (if at all) in both environments.

This paper explains how cloud network security is different, and how to pragmatically manage it for both pure cloud and hybrid cloud networks. We start with some background material and Cloud Networking 101, then move into cloud network security controls, and specific recommendations on how to use them. It is written for readers with a basic background in networking, but if you made it this far you'll be fine.

Cloud Networking 101

There is no canonical cloud networking stack – every cloud service provider uses its own mix of technologies to wire everything up. Some of these use known standards, tech, and frameworks, while others are completely proprietary and so secret that you as a customer never know exactly what is going on under the hood.

Building cloud scale networks is insanely complex, and the different providers clearly see networking capabilities as a competitive differentiator.

Instead of trying to describe all possible options we will keep things relatively high-level, and focus on common building blocks we see with some consistency across the different platforms.

Types of Cloud Networks

When you shop providers, cloud networks fit roughly into two buckets:

- **Software Defined Networks (SDN)** that fully decouple the virtual network from the underlying physical networking and routing.
- **VLAN-based Networks** that still rely on the underlying network for routing, lacking the full customization of SDN.

Most providers today offer full SDNs of different flavors, so we'll focus more on those, but we do still encounter some VLAN architectures which we need to cover at a high level.

Software Defined Networks

As we mentioned, Software Defined Networks are a form of virtual networking that (usually) takes advantage of special features in routing hardware to fully abstract the virtual network you see from the underlying physical network. To your instance (virtual server) everything looks like a normal network. But instead of connecting to a normal network interface it connects to a virtual network interface which handles everything in software.

SDNs don't work the same as physical networks (or even older virtual networks). For example, in an SDN you can create two networks that use the same address spaces and run on the same physical hardware but never see each other. You can create an entirely new subnet, not by adding hardware, but with a single API call that 'creates' the subnet in software.

How do they work? Ask your cloud provider. Amazon Web Services, for example, intercepts every packet, wraps it and tags it, and uses a custom mapping service to figure out where to actually send the packet over the physical network, with multiple security checks to ensure no customer ever sees anyone else's packets.

(Amazon has a video with great details). Your instance never sees the real underlying network, and AWS skips a lot of normal networking (including ARP requests/caching) within the SDN itself.

SDN enables you to take all your networking hardware, abstract it, pool it together, and then allocate it however you want. On some cloud providers, for example, you can allocate an entire class B network with multiple subnets, routed to the Internet behind NAT, in just a few minutes. Different cloud providers use different underlying technologies; just to complicate things they all offer different ways of managing their networks.

Why make things so complicated? Because SDN actually makes management of cloud networks much *easier*, while allowing cloud providers to offer customers a ton of flexibility to craft the virtual networks they need for different situations. The providers do the heavy lifting, and you as the consumer work in a simplified environment. Additionally SDN handles issues unique to the cloud, such as provisioning network resources faster than existing hardware can handle configuration changes (a very real problem), and multiple customers needing the same private IP address ranges to integrate optimally with existing applications.

Virtual LANs (VLANs)

Although they do not offer the same flexibility as SDNs, a few providers still rely on VLANs. Customers must evaluate their own needs, but VLAN-based cloud services should be considered outdated compared to SDN-based services.

VLANs let you create segmentation on the network, and can isolate and filter traffic, in effect just cutting off your own slice of the network rather than creating your own virtual environment. This means you can't do SDN-level things like creating two networks on the same hardware with the same address range.

- VLANs offer less flexibility. You can create segmentation on the network, and isolate and filter traffic, but cannot do SDN-level things like create two networks on the same hardware with the same address range.
- VLANs are built into standard networking hardware, which is why that's where many people start. No special software needed.
- Customers don't get to control their own addresses and routing very well.
- VLANs cannot be trusted for security segmentation.

VLANs are built into standard networking hardware, so they are an easier starting point for cloud computing. But customers on VLANs don't get to control their addresses and routing very well, and they scale and perform terribly when you plop a cloud on top of them. They are mostly being phased out of cloud computing due to these limitations.

Defining and Managing Cloud Networks

We like to think of one big cloud out there, but there is more than one kind of cloud network, and several technologies that support them. Each provides different features and customization options. Management

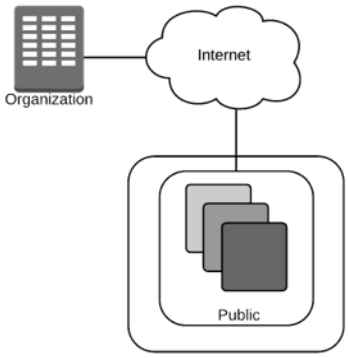
varies between vendors but they exhibit certain common characteristics. Different providers use different terminology, so we have tried to pick terms that will make sense when you look at actual offerings.

Cloud Network Architectures

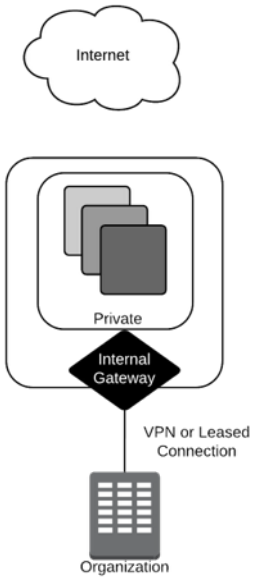
You need to understand the types of cloud network architectures, and the different technologies that enable them, to fit your needs to the right solution.

There are two basic types of cloud network architectures.

- **Public cloud networks** are Internet facing. You connect to your instances/servers via the public Internet with no special routing; every instance has a public IP address.
- **Private cloud networks** (also called "virtual private clouds") use private IP addresses like you would on a LAN. You need a back-end connection such as a VPN to connect to your instances. Most providers allow you to pick your own address ranges so you can use these private networks as extensions of your existing network. If you need to bridge traffic to the Internet you route it back through your data center or use Network Address Translation to a public network segment, similar to the way home networks use NAT to bridge to the Internet.



Simplified Public Cloud
Note that all communications traverses the Internet



Simplified (Virtual) Private Cloud
This time, communications are only via encrypted VPN or a leased line

These are enabled and supported by the following technologies:

- **Internet Connectivity:** An Internet gateway hooks your cloud network to the Internet. You don't normally manage it directly – your cloud provider does it for you.
- **Internal Gateways** connect your existing datacenter to your private cloud network. These are often VPN based, but instead of managing the VPN server yourself, your cloud provider handles it – you just manage the configuration. Some providers also support direct connections through partner broadband network providers which route directly between your data center and the private cloud network, instead of running a VPN over leased lines.
- **Virtual Private Networks:** Instead of using your cloud provider's, you can always set up your own, assuming you can bridge your private and public networks at your cloud provider. This kind of setup is very common, especially if you don't want to directly connect your data center and cloud but still want a private segment with access for users, developers, and administrators.

Cloud providers all break up their physical infrastructure differently. Typically they have different data centers (which might actually be a collection of multiple data centers clumped together) in different regions. A *region or location* is the physical location of the data center(s), while a *zone* is a sub-section of that region used for ensuring availability. These are for:

- **Performance:** Physical proximity can improve the performance of applications which pass large amounts of traffic.
- **Regulatory Requirements**: Geographical flexibility for your data stores can help you satisfy legal and regulatory requirements for data residency.
- **Disaster Recovery and Availability:** Most providers charge for some or all network traffic between across regions or locations, which can make disaster recovery expensive. That's why they provide local 'zones' to break individual regions into independent pieces – each with its own network, power, and so forth. A problem might take out one zone in a region, but shouldn't affect any others, giving customers a way to build resiliency without needing to span continents or oceans. Fortunately you typically don't pay for network traffic between zones *within* a region.

Managing Cloud Networks

Managing these networks depends on all the components listed above. Each vendor has its own set of tools, based on common principles.

- Everything is managed via APIs, which are typically RESTful (REpresentational State Transfer based).
- You can fully define and change everything remotely via APIs, and most changes are nearly instant.
- Cloud platforms also have web UIs, which are simply front ends for the same APIs you code to, but they tend to automate a lot of the heavy lifting for you.
- Key for security is protecting these management interfaces, because otherwise someone could completely reconfigure your network while sitting at a hipster coffee shop, making them by definition evil (fortunately you can usually spot them by these elite hackers by their ski masks, according to our clip art library).

Hybrid Cloud Architectures

As mentioned earlier, your data center may be connected to the cloud. Sometimes you need more resources you don't want on the public Internet. This is common for established companies which aren't starting from scratch, and need to mix and match resources.

There are two ways to accomplish this.

- **VPN Connections:** You connect to the cloud via a dedicated VPN, which is nearly always hardware-based and hooked into your local routers to span traffic to the cloud. The cloud provider handles their side of the VPN, but you still need to configure some of it. All traffic goes over the Internet but is isolated.

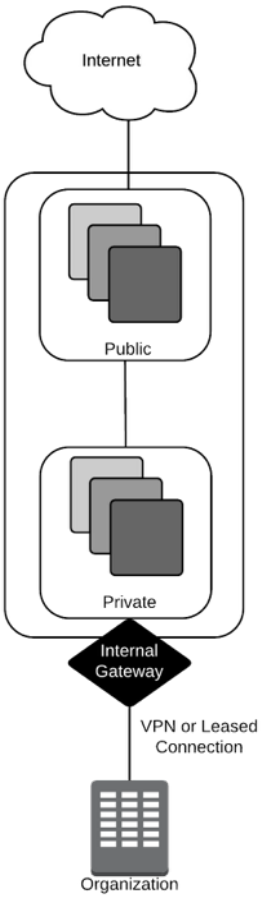
- **Direct Network Connections:** These are typically set up over leased lines. They aren't necessarily more secure, and are much more expensive, but can reduce latency.

Routing Challenges

Cloud services offer remarkable flexibility, but they also require substantial customization and pose their own security challenges.

Nearly every Infrastructure as a Service provider supports *auto scaling*, which is one of the single most important benefits of cloud computing. You can define your own rules for when to add or remove server instances. For example you could set a rule to add servers when you hit 80% CPU load, and terminate them when load drops… of course you need to architect appropriately for this kind of behavior.

This creates application elasticity: your resources can automatically adapt based on demand, instead of needing to leave extra servers running all the time *just in case* demand increases. Consumption now aligns with demand, unlike traditional architectures which leave a lot of hardware sitting around unused in case of high demand. This is the heart of IaaS. This is what you're paying for, and how you can save money.



Simplified Hybrid Cloud

Such flexibility creates complexity. You won't necessarily know the exact IP addresses of all your servers – they can appear and disappear within minutes. You may even design in complexity when you design for availability – by creating rules to keep multiple instances in multiple subnets across multiple zones available, in case one of them drops out. Within those virtual subnets you might have multiple different types of instances with different security requirements. This is common in cloud computing.

Fewer static routes, highly dynamic addressing, and servers that can exist for less than an hour… this all challenges security. It requires new ways of thinking, which is what the rest of this paper will focus on.

Our goal is to start getting you comfortable with how different cloud networks can be. On the surface, depending on your provider, you may still be managing subnets, routing tables, and ACLs. But underneath these are now probably database entries implemented in software, not the hardware you might be used to.

Network Security Controls

Now that we covered the basics of cloud networks, it's time to focus on available security controls. Keep in mind that this all varies between providers, and that cloud computing is rapidly evolving, with new capabilities appearing constantly. These fundamentals give you the background to get started, but you will need to learn the ins and outs of whatever platforms you work with.

What Cloud Providers Give You

Not to sound like a broken record (those round things your parents listened to… no, not the small shiny ones with lasers — nevermind!), but all providers are different. The following options are relatively common across providers, but not quite ubiquitous.

- **Perimeter security** is traditional network security that the provider manages completely, invisibly to their customers. Firewalls, IPS, etc. are used to protect the provider's infrastructure. The customer doesn't control any of it.

  Pro: It's free, effective, and always there.
  Con: You don't control any of it, and it's only useful for stopping background attacks.

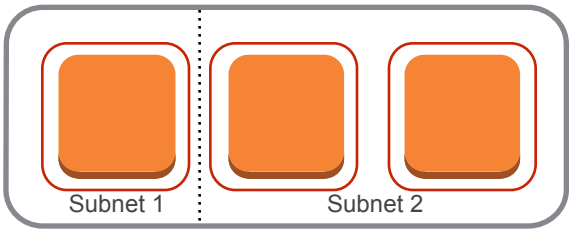
A note on monitoring:
None of the major providers offers packet-level network monitoring, and many don't offer any network monitoring at all. If you need it, consider host agents and virtual appliances.

- **Security groups:** Think of a group as a tag you can apply to a network interface/instance (or certain other cloud objects, such as databases and load balancers) that applies an associated set of network security rules. Security groups combine the best of network and host firewalls, because you get policies that can follow individual servers (or even network interfaces) like a host firewall, but you manage them like network firewalls – with protection applied no matter what is running inside. You get the granularity of a host firewall with the manageability of a network firewall. They are critical to auto scaling. You are spreading assets all over your virtual network, and instances appear and disappear on demand, so you cannot build security rules on IP addresses. Here's an example: You can create a "database" security group that only allows access to one specific database port, from instances inside a "web server" security group; only the web servers in that group can talk to the database servers in *this* "database" group. Unlike a network firewall, the database servers can't talk to each other, because they aren't in the web server group (remember, these rules are applied on a per-server basis, not at subnet boundaries, although that is also an option sometimes). As new databases pop up the right security is applied as long as they have the right tag. Unlike host firewalls you don't need to log into servers to make changes,

so everything is much easier to manage. Not all providers use this term, but the concept of security rules as a policy set you can apply to instances is relatively consistent.

Security groups vary between providers. Amazon, for example, is default deny and only supports allow rules. Microsoft Azure, however, supports rules that more closely resemble traditional firewalls, with both allow and block options.

Pro: They are free and work hand in hand with auto scaling and default deny. They are very granular, but also very easy to manage. They are the core of cloud network security.
Con: They are usually allow rules only (you can't explicitly deny), basic firewalling only, and you cannot manage them using most of your familiar tools.



Type	Protocol	Port Range	Source
SSH	TCP	22	68.2.174.98/32
HTTP	TCP	80	0.0.0.0/0

This example shows a screenshot of an Amazon Security Group applied to 3 instances in two subnets. Everything is blocked except the listed rules. Notice that the instances in the same subnet can't communicate. Only SSH access from a single IP address, and global web access, are allowed.

- **ACLs (Access Control Lists):** While security groups work at a per-instance (or per-object) level, ACLs restrict communications between subnets in your virtual network. Not all providers offer them, and they are more to handle legacy network configurations (when you need a restriction that matches what you might have in your existing data center) than 'modern' cloud architectures (which typically ignore or avoid them). In some cases you can use them to get around limitations of security groups, depending on your provider.

  Pro: ACLs can isolate traffic between virtual network segments and can create both allow and deny rules.
  Con: They're not great for auto scaling and don't apply to specific instances. You lose some powerful granularity.

By default nearly all cloud providers launch your assets with default-deny on all inbound traffic. Some might automatically open a management port from your current location (based on IP address), but that's about it.

Some providers use the term ACL to describe what we called a security group. Sorry! We know it's confusing, but blame vendors – not your friendly neighborhood analysts.

Commercial Options

There are a number of add-ons you can buy through your cloud provider, or buy and run yourself.

- **Physical security appliances:** The provider will provision an old-school piece of hardware to protect your assets. These are mostly seen at VLAN-based providers, and pretty antiquated. They may also be used in private (on-premise) clouds, where you control and run the network yourself, but that's out of scope for this research.

  Pro: They're expensive, but they're something you are used to managing. They keep your existing vendor happy? Look, it's really all cons on this one unless you're a cloud provider, in which case this paper isn't for you.
  Con: Cost can be a concern because they these use resources like any other virtual server, constrain your architectures, and may not play well with auto scaling and other cloud features.

- **Virtual appliances** are a virtual machine version of your friendly neighborhood security appliance, and must be configured and tuned for the cloud platform you are working on. They can provide more advanced security – such as IPS, WAF, NGFW – than cloud providers typically offer. They are also useful for capturing network traffic, which providers tend not to support directly.

  Pro: They offer more advanced network security, and can be managed the same as on-premise versions of these tools.
  Con: Cost – they use resources like any other virtual server, limiting your architecture options. They may not play well with auto scaling and other cloud features.

- **Host security agents** are software agents you build into images, which run in your instances and provide network security. This could include IDS, IPS, or other features beyond basic firewalling. We recommend lightweight agents with remote management. The agents (and management platform) need to be designed for use in cloud computing, because auto scaling and portability break traditional tools.

  Pro: Like virtual appliances, host security agents can offer features missing from your cloud provider. With a good management system they can be extremely flexible, and usually include capabilities beyond network security. They are a great option for monitoring network traffic.
  Con: You need to make sure they are installed and run in all your instances, and they are not free. They also won't work well if you get one that isn't designed for the cloud.

To review, your network security controls, no matter what your provider calls them, nearly always fall into 5 buckets:

- Perimeter security the provider puts in place, which you never see or control.
- Software firewalls built into the cloud platform (security groups) that protect cloud assets such as instances, offer basic firewalling, and are designed for auto scaling and other cloud-specific uses.

- Lower-level Access Control Lists for controlling access into and out of the subnets in your virtual cloud network.
- Virtual appliances to add expanded features from your familiar network security tools, such as IDS/IPS, WAF, and NGFW.
- Host security agents to embed in your instances.

Advanced Options on the Horizon

We know some niche vendors already offer more advanced network security built into their platforms, such as IPS, and we exspect major vendors to eventually offer similar options. We don't recommend picking a cloud provider based on these, but you may get more options in the future.

Building Your Cloud Network Security Program

There is no single 'best' way to secure a cloud or hybrid network. Cloud computing is moving faster than any other technology in decades, with providers constantly struggling to out-innovate each other with new capabilities. You cannot lock yourself into any single architecture, but need to build out a program capable of handling diverse and dynamic needs.

There are four major focus areas when building out this program.

- Start by understanding the *key considerations* for the cloud platform and application you are working with.
- Design your *network and application architecture* for security.
- Design your *network security architecture*, including any necessary additional security tools and management components.
- *Manage security operations* for your cloud deployments – including everything from staffing to automation.

Understand Key Considerations

Building applications in the cloud is decidedly not the same as building them on traditional infrastructure. Sure, you can do it, but the odds are high that something will break. Badly – as in "update the resume" breakage. To see the full benefit of cloud computing, applications must be designed specifically for the cloud – including security controls.

For network security this means you need to keep a few key things in mind before you start mapping out security controls.

- **Provider-specific limitations or advantages:** All providers are different. Nothing is standard, and don't expect it to ever become standard. One provider's security group is another's ACL. Some allow more granular management. There may be limits on the number of security rules available. A provider might offer both `allow` and `deny` rules, or `allow` only. Take the time to learn the ins and outs of your provider's capabilities. They all offer plenty of documentation and training, and in our experience most organizations limit themselves to no more than one to three infrastructure providers, keeping the problem manageable.

- **Application needs:** Applications, especially those using the newer architectures we will mention in a moment, often have different needs than applications deployed on traditional infrastructure. For example application components in your private network segment may still need Internet access to connect to a cloud component – such as storage, a message bus, or a database. These needs directly affect architectural decisions – both security and otherwise.

- **New architectures:** Cloud applications use different design patterns than apps on traditional infrastructure. For example, as previously mentioned, components are typically distributed across diverse network locations for resiliency, and tied tightly to cloud-based load balancers. Early cloud applications often emulated traditional architectures, but modern cloud applications make extensive use of advanced cloud features – particularly Platform as a Service, which may be highly specific to a particular cloud provider. Cloud-based databases, message queues, notification systems, storage, containers, and application platforms are all now common due to cost, performance, and agility benefits. You often cannot even control the network security of these services, which are instead fully managed by the cloud provider. Continuous deployment, DevOps, and immutable servers are the norm rather than exceptions. On the upside, used properly these architectures and patterns are far more secure, cost effective, resilient, and agile than building everything yourself, but you really need to understand how they work.

- **Elasticity and a high rate of change are standard in the cloud:** Beyond auto scaling, cloud applications tend to alter the infrastructure around them to maximize the benefits of cloud computing. For example one of the best ways to update a cloud application is not to patch servers, but instead to create an entirely new installation of the app, based on a revised template, running in parallel – and then to switch traffic over from the current version. This breaks familiar security approaches, including relying on IP addresses (for server identification, vulnerability scanning, and logging). Server names and addresses are largely meaningless, and controls which aren't adapted for the cloud are liable to be useless.

- **Managing and monitoring security changes:** You either need to learn how to manage cloud security using the provider's console and APIs, or choose security tools that integrate directly. This may become especially complex if you need to normalize security between your data center and cloud provider when building a hybrid cloud. Additionally, few cloud providers offer good tools to track security changes over time, so you need to track them yourself or use a third-party tool.

Immutable Servers

An immutable server is one you never change, at least when it's running. Since all instances are based off images, you create a new, updated image and then deploy that in place of the running ones. This is an inherent characteristic of auto scaling, since you already need to design servers to be automatically replaced. Thus you never need to log into an instance, since even patching is done on the image and you swap out the servers. You can disable **all** admin access, including SSH, making your servers very difficult to hack.

Design the Network Architecture

Unlike traditional networks, security is built into cloud networks by default. Go to any major cloud provider, spin up a virtual network, launch a server, and the odds are very high it is already well-defended – with most or all access blocked by default.

Because security and core networking are so intertwined, and *every* cloud application has its own virtual network (or networks), the first step toward security is to work with the application team and design it into the architecture.

Here are specific guidelines and recommendations:

- **Accounts** provide your first layer of segregation. Establish multiple accounts with each cloud provider, each for a different environment (e.g., dev, test, production, logging). This enables you to tailor cloud security controls and minimize administrator access. This isn't a purely network security feature, but affects network security because you can, for example, have tighter controls for environments closer to production data. The rule of thumb for accounts is to consider separate accounts for separate applications, and then separate accounts for a given application when you want to restrict how many people have administrator access. For example a dev account is more open with more administrators, while production is a different account with a much smaller group of admins. Within accounts, don't forget about the physical architecture:
    - **Regions/locations** are often used for resiliency, but may also be incorporated into the architecture for data residency requirements or to reduce network latency to customers. Unlike accounts, we don't normally use locations for security, but you do need to build network security within each location.
    - **Zones** are the cornerstone of cloud application resiliency, especially when tied to auto scaling. You won't use them as a security control, but they affect also security, as they often map directly to subnets. An auto scale group might keep multiple instances of a server in different zones, which are different subnets, so you cannot necessarily rely on subnets and addresses when designing your security.
- **Virtual Networks (Virtual Private Clouds)** are your next layer of security segregation. You can (and will) create and dedicate separate virtual networks for each application (potentially in different accounts), each with its own set of network security controls. This compartmentalization offers tremendous security advantages, but seriously complicates security management. It forces you to rely much more heavily on automation, because manually replicating security controls across accounts and virtual networks within each account takes tremendous discipline and effort. In our experience the security benefits of compartmentalization outweigh the risks created by management complexity – especially because development and operations teams already tend to rely on automation to create, manage, and update environments and applications in the first place. There are a few additional non-security-specific aspects to keep in mind when you design the architecture:

- Within a given virtual network, you can include public and private facing subnets and connect them together. This is similar to DMZ topologies, except that public-facing assets can still be fully restricted from the Internet, and private network assets are all by default walled off from each other. Even more interesting, you can spin up totally isolated private network segments that only connect to other application components through an internal cloud service such as a message queue, and prohibit all server-to-server traffic over the network.
- There is no additional cost to spin up new virtual networks (or if your provider charges for this, it's time to move on), and you can create another with a few clicks or API calls. Some providers even allow you to bridge across virtual networks, assuming they aren't running on the same IP address range. Instead of trying to lump everything into one account and one virtual network, it makes far more sense to use multiple networks for different applications, and even within a given application architecture.
- Within a virtual network you also have complete control over subnets. While they may play a role in your security design, especially as you map out public and private network segments, make sure you also design them to support zones for availability.
- Flat networks aren't flat in the cloud. Everything you deploy in the virtual network is surrounded by its own policy-based firewall which blocks all connections by default, so you don't need to rely on subnets themselves as much for segregation between application components. Public vs. private subnets are one thing, but creating a bunch of smaller subnets to isolate application components quickly leads to diminishing returns.

Hybrid Clouds

You may need *enterprise datacenter connections for hybrid clouds*. These VPN or direct connections route traffic directly from your data center to the cloud and vice-versa. You simply set your routing tables to send traffic to the appropriate destination, and SDN-based virtual networks allow you to set distinct subnet ranges to avoid address conflicts with existing assets.

Whenever possible, we actually recommend avoiding hybrid cloud deployments. It isn't that there is anything wrong with them, but they make it much more difficult to support account and virtual network segregation. For example if you use separate accounts or virtual networks for your different dev/test/prod environments, you will tend to do so using templates to automatically build out your architecture, and they will perfectly mimic each other – down to individual IP addresses. But if you connect them directly to your data center you need to shift to non-overlapping address ranges to avoid conflicts, so they can't be as automated or consistent. (This consistency is a cornerstone of *continuous deployment and DevOps*).

Additionally, hybrid clouds complicate security. We have actually seen them, not infrequently, reduce the overall security level of the cloud, because assets in the datacenter aren't as segregated as on the cloud network, and cloud providers tend to be more secure than most organizations can manage in their own

infrastructure. Instead of cracking your cloud provider, someone only needs to crack a system on your corporate network, and use that to bridge to the cloud.

When should you consider a hybrid deployment? Any time your application architecture requires direct address-based access to an internal asset that isn't Internet-accessible. Alternatively, sometimes you need a cloud asset on a static, non-Internet-routable address – such as an email server or other service that isn't designed to work with auto scaling – which internal things need to connect to. (We *strongly* recommend you minimize these – they don't benefit from cloud computing, so there is usually no good reason to deploy them there). And yes, this means hybrid deployments are extremely common unless you are building everything from scratch. We minimize their use as much as possible, but they are still important.

For security there are a few things to keep in mind when building a hybrid deployment:

- VPN traffic will traverse the Internet. VPNs are very secure, but you need to keep them up-to-date with the latest patches and make sure you use strong, up-to-date certificates.
- Direct connections may reduce latency, but decide whether you trust your network provider or need to encrypt traffic.
- Don't let your infrastructure reduce the security of your cloud. If you mandate multi-factor authentication in the cloud but not on your LAN, that's a loophole. Is your entire LAN connected to the cloud? Could someone compromise a single workstation and then start attacking your cloud through your direct connection? Do you have security groups or other firewall rules to keep your cloud assets as segregated from datacenter assets as from each other? Cloud providers tend to be exceptionally good at security, and everything you deploy in the cloud is isolated by default. Don't allow hybrid connection to become a weak link, and reduce your compartmentalization.
- You may still be able to use multiple accounts and virtual networks for segregation, by routing different datacenter traffic to different accounts and/or virtual networks. But your on-premise VPN hardware or your cloud provider might not support this, so check before designing it into your architecture.
- Cloud and on-premise network security controls may look similar on the surface, but they have deep implementation differences. If you want unified management you need to understand them and be able to harmonize based on security goals – not by trying to force a standard implementation across very different technologies.
- Cloud computing offers many more ways to integrate into your existing operations than you might think. For example instead of using SFTP and setting up public servers to receive data dumps, consider installing your cloud provider's command-line tools and directly transferring data to their object storage service (fully locked down, of course). Now you don't need to maintain the burden of either an Internet-accessible FTP server or a hybrid cloud connection.

It's hard to convey the full breadth and depth of options for building security into your architectures, even without additional security tools. This isn't mere theory – we have a lot of real-world experience with different

architectures creating much higher security levels than can be achieved on traditional infrastructure at any reasonable cost.

Design the Network Security Architecture

At this point you should have a well-segregated environment where effectively every application, and every environment (*e.g.*, dev/test) for every application, is running on its own virtual network. These assets are mostly either in auto scale groups which spread them around zones and subnets for resiliency; or connect to secure cloud services such as databases, message queues, and storage. In our experience these architectures themselves are materially more secure than a typical starting point on traditional infrastructure.

Now it's time to layer on the additional security controls we covered earlier under *Cloud Networking 101*. Instead of repeating the pros and cons, here are some direct recommendations about when to use each option:

- **Security groups:** These should be used and set to deny by default. Only open up the absolute minimum access needed. Cloud services allow you to right-size resources far more easily than on your own hardware, so we find most organizations tend to deploy far fewer services on each instance, which directly translates to fewer network ports open per instance. Many cloud deployments we have evaluated use only a good base architecture and security groups for network security.
- **ACLs:** These mostly make sense in hybrid deployments, when you need to closely match or restrict communications between the data center and the cloud. Security groups are usually a better choice, and we only recommend falling back to ACLs or subnet-level firewalling when you cannot achieve your security objectives otherwise.
- **Virtual Appliances:** Whenever you need capabilities beyond basic firewalls, this is where you are likely to end up. But we find host agents often make more sense when they offer the same capabilities, because virtual appliances become costly bottlenecks which restrict cloud architecture options. Don't deploy one merely because you have a checkbox requirement for a particular tool – ensure it makes sense first. Over time we do see them becoming more "cloud friendly", but when we rip into requirements on projects, we often find there are better, more cloud-appropriate ways to meet the same security objectives.
- **Host security agents** are often a better option than virtual appliances because they don't restrict virtual networking architectural options. But you need to ensure you have a way to deploy them consistently. Make sure you pick cloud-specific tools designed to work with features such as auto scaling. These tools are particularly useful to cover network monitoring gaps, meet IDS/IPS requirements, and satisfy all your normal host security needs.

Of course you will need some way of managing these controls, even if you stick to only capabilities and features offered by your cloud provider.

Security groups and ACLs are managed via API or your cloud provider's console. They use the same *management plane* as the rest of the cloud, but this won't necessarily integrate out of the box with the way you manage things internally. You can't track them across multiple accounts and virtual networks unless you use a purpose-built tool or write your own code. We will talk about specific techniques for management in the next section, but make sure you plan out how to manage these controls when you design your architecture.

Platform as a Service introduces its own set of security differences. For example in some situations you still define security groups and/or ACLs for the PaaS (as with a cloud load balancer); but in other cases access to the platform is only via API, and may require an outbound public Internet connection, even from a private network segment. PaaS also tends to rely more on DNS rather than IP addresses, to help the cloud provider maintain flexibility. We can't give you any hard and fast rules here. Understand what's required to connect to the platform, and then ensure your architecture allows those connections. When you can manage security, treat it like any other cluster of servers, and stick with the minimum privileges possible.

We cannot cover anything near every option for every cloud in a relatively short (believe it or not) paper like this, but for the most part once you understand these fundamentals and the core differences of working in software-defined environments, it becomes much easier to adapt to new tools and technologies.

Especially once you realize that you start by integrating security into the architecture, instead of trying to layer it on after the fact.

Manage Cloud (and Hybrid) Network Security Operations

Building in security is one thing, but keeping it up to date over time is an entirely different – and harder – problem. Not only do applications and deployments change over time, but cloud providers have this pesky habit of "innovating" for "competitive advantage". Someday things might slow down, but it definitely won't be within the lifespan of this particular research.

Here are some suggestions on managing cloud network security for the long haul.

Organization and Staffing

It's a good idea to make sure you have cloud experts on your network security team, people trained for the platforms you support. They don't need to be new people, and depending on your scale this doesn't need to be their full-time focus, but you definitely need the skills. We suggest you build your team with both security architects (to help design) and operators (to implement and fix).

Cloud projects occur outside the constraints of your data center, including normal operations, so you might need to make organizational changes so security is engaged in projects. A security representative should be assigned and integrated into each cloud project. Think about how things normally work – someone starts a new project and security gets called when they need access or firewall rule changes. With cloud computing, network security isn't blocking anything (unless they need access to an on-premise resource), and entire projects can happen without security or ops being directly involved. You need to adapt policies and organizational structure to minimize this risk. For example work with procurement to require a security evaluation and consultation before any new cloud account is opened.

Because so much of cloud network security relies on architecture, it isn't just important to have a security architect on the team – it is *essential* they be engaged in projects early. It goes without saying that this needs to be a collaborative role. Don't merely write up some pre-approved architectures and then try to force everyone to work within those constraints. You'll lose that fight before you even know it started.

Discovery

We hinted at this in the section above: one of the first challenges is to *find* all the cloud projects, and then keep finding new ones as they pop up over time. You need to enumerate your existing cloud network security controls. Here are a couple ways we have seen clients successfully keep tabs on cloud computing:

- If your critical assets (such as a customer database) are well locked down, you can use this to control cloud projects. If they want access to the data/application/whatever, they need to meet your security requirements.
- Procurement and Accounting are your next best options. At some point someone needs to pay the (cloud) piper, and you can work with Accounting to identify payments to cloud providers and tie them back to the teams involved. Just make sure you differentiate between credit card charges to Amazon for office supplies, and the one to replicate your entire datacenter up into AWS.
- Hybrid connections to your data center are pretty easy to track using established process. Unless you let random employees plug in VPN routers.
- Lastly, you could try setting a policy that says "don't cloud without telling us". I mean, if you trust your people and all. It could work. Maybe. It's probably good to have to keep the auditors happy, anyway.

The next discovery challenge is to figure out how the cloud networks are architected and secured:

- First, always start with the project team. Sit down with them and perform an architecture and implementation review.
- It's a young market, but there are some assessment tools that can help. Especially to analyze security groups and network security and compare against best practices.
- You can use your cloud provider's console in many cases, but most of them don't provide a good overall network view. If you don't have a tool to help, you can use scripts and API calls to pull down the raw configuration and analyze it manually.

Integrating with Development

In the broadest sense there are two kinds of cloud deployments: applications you build and run in the cloud (or hybrid), and core infrastructure (like file and mail servers) you transition to the cloud. Developers play the central role in the former, but they are also often involved in the latter.

The cloud is essentially *software defined everything*. We build and manage all kinds of cloud deployments using code. Even if you start by merely transitioning a few servers into virtual machines at a cloud provider, you will inevitably end up defining and managing much of your environment in code.

This is an incredible opportunity for security. Instead of sitting outside the organization and trying to protect things by building external walls, we gain much greater ability to manage security using the exact same tools Development and Operations use to define, build, and run infrastructure and services. Here are a few key ways to integrate with development and ensure security is integrated:

- Create a handbook of design patterns for the cloud providers you support, including security controls and general requirements. Keep adding new patterns as you work on new projects. Then make this library available to business units and development teams so they know which architectures already have general approval from Security.
- A cloud security architect is essential, and this person or team should engage early with development teams to help build security into every initial design. We hate to have to say it, but the role really needs to be *collaborative*. Lay down the law with a bunch of requirements that interfere with the project's execution, and you definitely won't be invited back to the table.
- A lot of security can be automated and templated by working with development. For example monitoring and automation code can be deployed on projects without the project team having to develop them from scratch. Even integrating third party tools can often be managed programmatically.

Policy Enforcement

Change is constant in cloud computing. The foundational concept is dynamic adjustment of capacity (and configuration) to meet changing demands. When we say "enforce policies" we mean that, for a given project, once you design the security you are able to keep it consistent. Just because clouds change all the time doesn't mean it's okay to let a developer drop all the firewalls by mistake.

The key policy enforcement difference between traditional networking and the cloud is that in traditional infrastructure security has exclusive control over firewalls and other security tools. In the cloud anyone with sufficient authorization in the cloud platform (management plane) can make those changes. Even applications can reconfigure their own infrastructure. That's why you need to rely on automation to detect and manage change.

You lose the single point of control. Heck, your own developers can create entire networks from their desktops. Remember when someone occasionally plugged in their own wireless router or file server? It's a bit like that, but more like building their own datacenter over lunch. Here are some techniques for managing these changes:

- Use access controls to limit who can change what on a given cloud project. It is typical to allow developers a lot of freedom in the dev environment, but lock down any network security changes in production, using your cloud provider's IAM features.
- To the greatest extent possible, try to use cloud provider specific templates to define your infrastructure. These files contain a programmatic description of your environment, including complete network and network security configurations. You load them into a cloud platform and it builds the environment for you. This is a very common way to deploy cloud applications, and essential in organizations using DevOps to enforce consistency.

- When this isn't possible you will need to use a tool or manually pull the network architecture and configuration (including security) and document them. This is your baseline.
- Then you need to automate change monitoring using a tool or the features of your cloud and/or network security provider:
    - Cloud platforms are slowly adding monitoring and alerting on security changes, but these capabilities are still new and often manual. This is where cloud-specific training and staffing can really pay off, and there are also third-party tools to monitor these changes for you.
    - When you use virtual appliances or host security you don't rely on your cloud provider, so you may be able to hook change management and policy enforcement into your existing approaches. These are security-specific tools, so unlike cloud provider features the security team will often have exclusive access and be responsible for making changes themselves.
- Did we mention automation? We will talk about it more in a minute because it's the only way to maintain cloud security.

Normalizing On-Premise and Cloud Security

Organizations have a lot of security requirements for very good reasons, and need to ensure those controls are consistently applied. We all have developed a tremendous amount of network security experience over decades running our own networks, which is still relevant when moving into the cloud. The challenge is to carry over the requirements and experience without assuming everything is the same in the cloud, or letting old patterns prevent us from taking full advantage of cloud computing.

- Start by translating whatever rules sets you have on-premise into a comparable version for the cloud. This takes a few steps:
    - Figure out which rules should still apply, and what new rules you need. For example a policy to deny all ssh traffic from the Internet won't work if that's how you manage public cloud servers. Instead a policy that limits `ssh` access to your corporate CIDR block makes more sense. Another example is the common restriction that back-end servers shouldn't have any Internet access at all, which may need updating if they need to connect to PaaS components of their own architecture.
    - Then adapt your policies into enforceable rulesets. For example security groups and ACLs work differently, so how you enforce them changes. Instead of setting subnet-based policies with a ton of rules, tie security group policies to instances by function. We once encountered a client who tried to recreate very complex firewall rulesets into security groups, exceeding their provider's rule count limit. Instead we recommended a set of policies for different categories of instances.
    - Watch out for policies like "deny all traffic from this IP range". Those can be very difficult to enforce using cloud-native tools, and if you really have those requirements you will likely need a network security virtual appliance or host security agent. In many projects we find you can

resolve the same level of risk with smarter architectural decisions (such as using immutable servers, which we will describe in a moment).

  - Don't just drop in a virtual appliance because you are used to it and know how to build its rules. Always start with what your cloud provider offers, then layer on additional tools as needed.

  - If you migrate existing applications to the cloud the process is a bit more complex. You need to evaluate existing security controls, discover and analyze application dependencies and network requirements, and then translate them for a cloud deployment, taking into account all the differences we have been discussing.

- Once you translate the rules, normalize operations. This means having a consistent process to deploy, manage, and monitor your network security over time. Fully covering this is beyond to scope of this research, as it depends on how you manage network security operations today. Just remember that you are trying to blend what you do now with the cloud project's requirements, not simply enforce existing processes under an entirely new operating model.

We hate to say it, but we will: this is a process of transition. We find customers who start on a project-by-project basis are more successful because they can learn as they go and build up a repository of knowledge and experience.

Automation and Immutable Network Security

Cloud security automation isn't merely fodder for another paper – it's an entirely new body of knowledge we are all just beginning to build.

Any organization that moves to the cloud in any significant way learns quickly that automation is the only way to survive. How else can you manage multiple copies of a single project in different environments – never mind dozens or hundreds of different projects, each running in their own sets of cloud accounts across multiple providers?

Then keep all those projects compliant with regulatory requirements and your internal security policies.

Yeah, it's like that.

Fortunately this isn't an insoluble problem. Every day we see more examples of companies successfully using the cloud at scale, and staying secure and compliant. Today they largely build their own libraries of tools and scripts to continually monitor and enforce changes. We also see some emerging tools to help with this management, and expect to see many more in the near future.

A core developing concept tied to automation is immutable security, and we have used it ourselves.

One of the core problems in security is managing change. We design something, build in security, deploy it, validate that security, and lock everything down. This inevitably drifts as it's patched, updated, improved, and otherwise modified. Immutable security leverages automation, DevOps techniques, and inherent cloud characteristics to break this cycle. To be honest, it's really just DevOps applied to security, and all the principles are in wide use already.

For example an *immutable server* is one that is never logged into or changed in production. If you remember back to auto scaling, we deploy servers based on standard images. Changing one of those servers after deployment doesn't make sense, because those changes wouldn't be in the image, so new versions launched by auto scaling wouldn't include them. Instead DevOps creates a new image with all the changes, then alters the auto scale group rules to deploy new instances based on the new image, and optionally prunes off the older versions.

In other words no more patching and no more logging into servers. You take a new known-good state, and completely replace what is in production.

Think about how this applies to network security. We can build templates to automatically deploy entire environments at our cloud providers. We can write network security policies, then override any changes automatically, even across multiple cloud accounts. This pushes the security effort earlier into design and development, and enables much more consistent enforcement in operations. And we use the exact same toolchain as Development and Operations to deploy our security controls, rather than trying to build our own on the side and overlay enforcement afterwards.

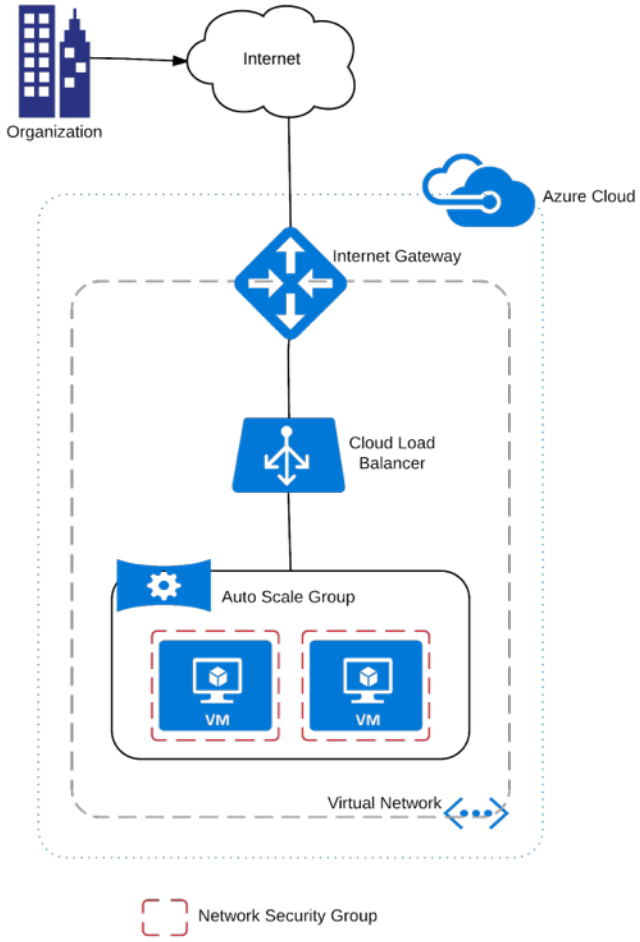
This might seem like an aside, but these automation principles are the cornerstone of real-world cloud security, especially at scale. This is a capability we *never* have with traditional infrastructure, where we cannot simply stamp out new environments automatically and need to hand-configure everything.

Design Examples

To finish off this research it's time to show what some of this looks like. Here are some practical design patterns based on projects we have worked on. The examples are specific to Amazon Web Services and Microsoft Azure, rather than generic templates. Generic patterns are less detailed and harder to explain, and we would rather you understand what these look like in the real world.

Basic Public Network on Microsoft Azure

This is a simplified example of a public network on Azure. All the components run on Azure, with nothing in the enterprise data center and no VPN connections. Management of all assets is over the Internet. We can't show all the pieces and configuration settings in this diagram, so here are some specifics:

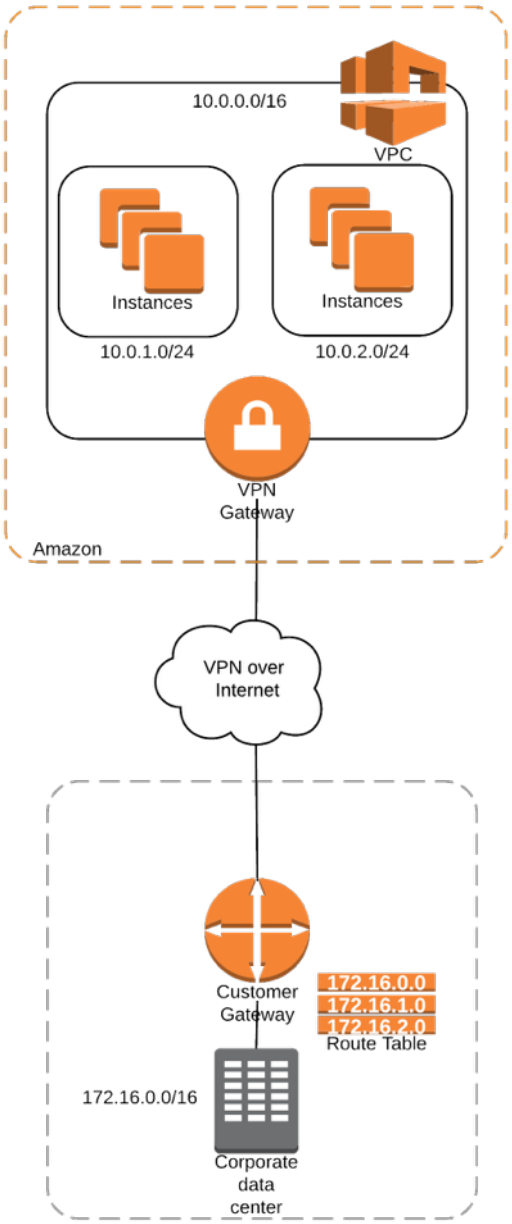


- The Internet Gateway is set in Azure by default (you don't need to do anything). Azure also sets up default *service endpoints,* which are routed management ports to manage your instances (think of it as a combination of firewall and port forwarding). These connections are direct to each instance and don't run through the load balancer. They will (should) be limited to only your current IP address, so the ports are closed to the rest of the world. In this example we have a single public facing subnet.
- Each instance gets a public IP address and domain name, but you can't access anything that isn't opened up with a defined service endpoint. Think of the endpoint as port forwarding, which it pretty much is.
- The service endpoint can point to the load balancer, which in turn is tied to the auto scale group. You set rules on instance health, performance, and availability; the load balancer and auto scale group provision and deprovision servers as needed and handle routing. The IP addresses of the instances change as these updates take place.
- *Network Security Groups (NSGs)* restrict access to each instance. In Azure you can also apply them to subnets. In this case we apply them on a per-server basis. Traffic is restricted to whatever services are being provided by the application, and denied between instances on the same subnet. Azure allows such internal traffic by default, unlike Amazon.
- NSGs can also restrict traffic to the instances, locking it down to only from the load balancer, disabling direct Internet access. Ideally you never need to log into the servers because they are in an auto scale group, so you can also disable all the management/administration ports.

There is more to do, but this pattern produces a hardened server with no administrative traffic, protected with both Azure's default protections and Network Security Groups. Note that on Azure you are often much better off using their PaaS offerings such as web servers, instead of manually building infrastructure like this.

Basic Private Network on Amazon Web Services

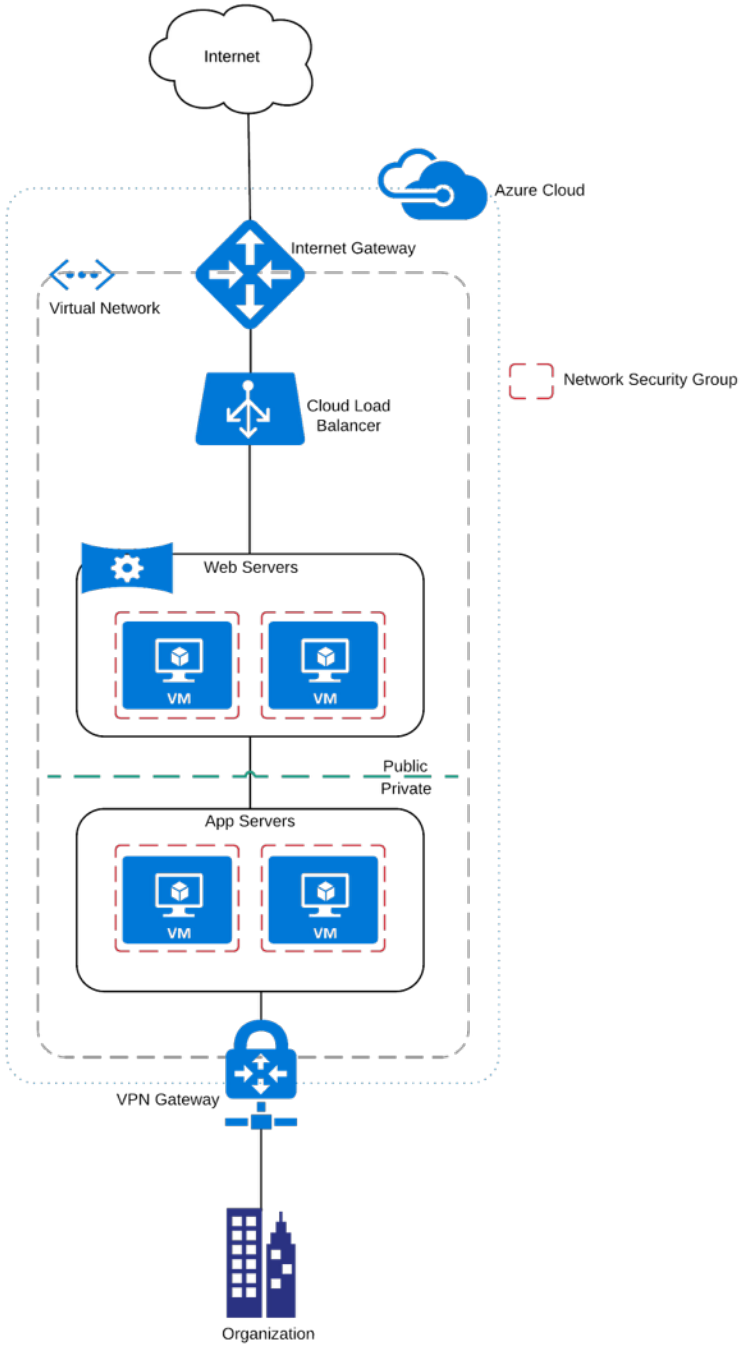
Amazon works a bit differently than Azure (okay – much differently). This example is a Virtual Private Cloud (VPC, their name for a virtual network) that is completely private, without any Internet routing, connected to a data center through a VPN connection.



- This shows a class B network with two smaller subnets. In AWS you would place each subnet in a different *Availability Zone* (what we called a 'zone') for resilience in case one goes down – these are separate physical data centers.
- You configure the VPN gateway through the AWS console or API, and then configure the client side of the VPN connection on your own hardware. Amazon maintains the VPN gateway in AWS; you don't directly touch or maintain it, but you do need to maintain everything on your side of the connection (and it needs to be a hardware VPN).
- You adjust the routing table on your internal network to send all traffic for the 10.0.0.0/16 network over the VPN connection to AWS. This is why it's called a 'virtual' private cloud. Instances can't see the Internet, but you have that gateway that's Internet accessible.
- You also need to set your virtual routing table in AWS to send Internet traffic back through your corporate network if you want any of your assets to access the Internet for things like software updates. Sometimes you do, sometimes you don't – we don't judge.
- By default instances are protected with a Security Group that denies all inbound traffic and allows all outbound traffic. Unlike in Azure, instances on the same subnet can't talk to each other. You cannot connect to them through the corporate network until you open them up. AWS Security Groups offer `allow` rules only. You cannot explicitly deny traffic – only open up allowed traffic. In Azure you create Service Endpoints to explicitly route traffic, then use network security groups to allow or deny on top of them (within the virtual network). AWS uses security groups for both functions – opening a security group allows traffic through the private IP (or public IP if it is public facing).
- Our example uses no ACLs but you could put an ACL in place to block the two subnets from talking to each other. ACLs in AWS are there by default but allow all traffic. An ACL in AWS is not stateful, so you need to create rules for all bidirectional traffic. ACLs in AWS work better as a deny mechanism.
- A public network on AWS looks relatively similar to our Azure sample (because we designed them that way). The key differences are how security groups and service endpoints function.

Hybrid Cloud on Azure

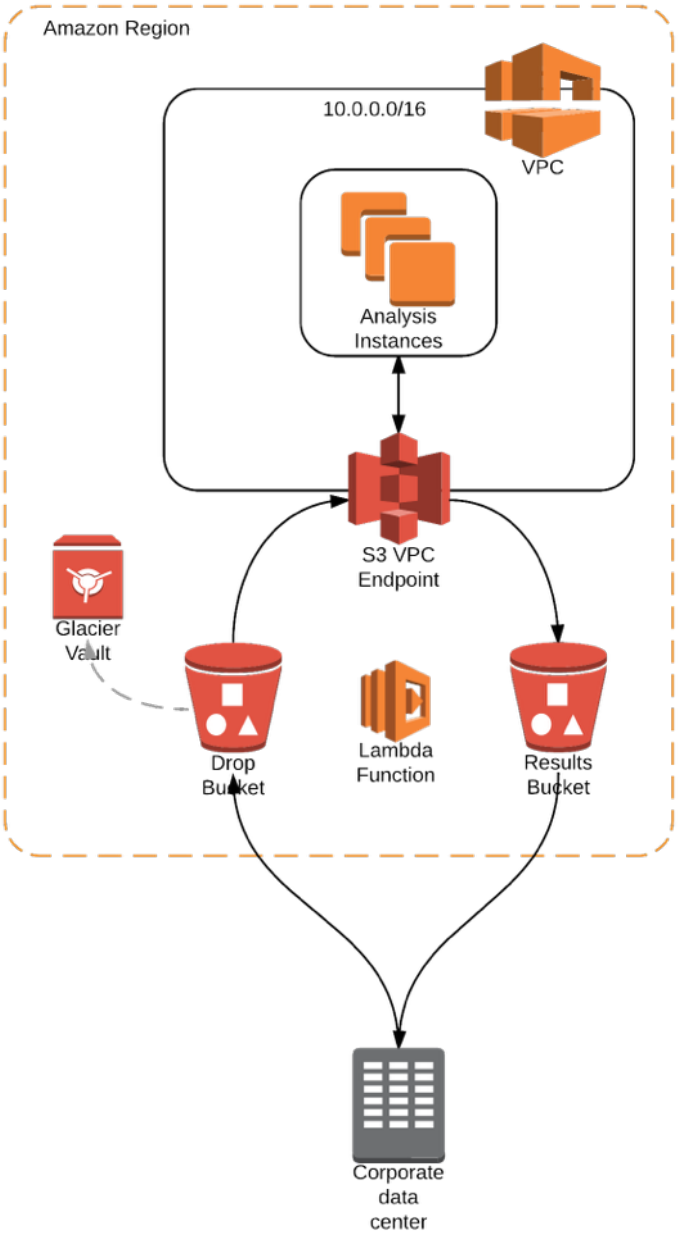
This builds on our previous examples. In this case the web servers and app servers are separated, with app servers on a private subnet. We already explained the components in our other examples, so there is only a little to add:



- The key security control here is a Network Security Group to allow access to the app servers **only** from the web servers, and only to the specific port and protocol required.
- The NSG should be applied to each instance, not to the subnets, to prevent a “flat network” and block peer traffic that could be used in an attack.
- The app servers can connect to your data center, and that is where you route all Internet traffic. That gives you just as much control over Internet traffic as with virtual machines in your own data center.
- You will want to restrict traffic from your organization’s network to the instances (via the NSGs) so you don’t become the weak link for an attack.

A Cloud Native Data Analytics Architecture

Our last example shows how to use some of the latest features of Amazon Web Services to create a new cloud-native design for big data transfers and analytics.



- In this example we have is a private subnet in AWS, without either Internet access or a connection to the enterprise data center. Images will be created in another account or VPC, and there will be no manual logins.

- When an analytics job is triggered, a server in the data center takes the data and sends it to Amazon S3, their object storage service, using command-line tools or custom code. This is an encrypted connection by default, but you could also encrypt the data using the AWS Key Management Service or any other encryption tool. We have clients using both options.
- The S3 bucket in AWS is tightly restricted to either only the IP address of the sending server, a set of AWS IAM credentials, or both. AWS manages S3 security so you don't worry about network attacks, merely enabling access. S3 isn't like a public FTP server – if you lock it down (easy to do) it isn't visible except from authorized sources.
- A service called AWS Lambda monitors the S3 bucket. Lambda is a container for event-driven code running inside Amazon that can trigger based on internal things, including a new file appearing in an S3 bucket. You only pay for Lambda when your code is executing, so there is no cost to have it wait for events.
- When a new file appears the Lambda function triggers and launches analysis instances based on a standard image. The analysis instances run in a private subnet, with security group settings that block **all** inbound access.
- When the analysis instances launch, the Lambda code sends them the S3 location of the data to analyze. The instances connect to S3 through something known as a *VPC Endpoint*, which is totally different from an Azure service endpoint. A VPC endpoint allows instances in a totally private subnet to talk to S3 without Internet access (which was required until recently). As of this writing only S3 offers a VPC endpoint, but we know Amazon is working on endpoints for additional services such as their Simple Queue Service (we suspect AWS hasn't confirmed exactly which services are next on the list).
- The instances boot, grab the data, then do their work. When they are done they go through the S3 VPC endpoint to drop their results into a second S3 bucket.
- The first bucket only allows writes from the data center, and reads from the private subnet. The second bucket reverses that and only allows reads from the data center and writes from the subnet. Everything is a one-way closed loop.
- The instance can then trigger another Lambda function to send a notification back to your on-premise data center or application that the job is complete, and code in the data center can grab the results. There are several ways to do this – for example the results could go into a database, instead.
- Once everything is complete Lambda moves the original data into *Glacier*, Amazon's super-cheap long-term archival storage. In this scenario it is of course encrypted. (For this network-focused research we are skipping over most of the encryption options for this architecture, but they aren't overly difficult).

Think about what we have described: the analysis servers have no Internet access, spin up only as needed, and can only read in new data and write out results. They automatically terminate when finished, so there is no persistent data sitting unused on a server or in memory. All Internet-facing components are native Amazon services, so we don't need to maintain their network security. Everything is extremely cost-effective,

even for very large data sets, because we only process when we need it; big data sets are always stored in the cheapest option possible, and automatically shifted around to minimize storage costs. The system is event-driven so if you load 5 jobs at once, it runs all 5 at the same time without any waiting or slowdown, and if there are no jobs the components are just programmatic templates, in the absolute most cost-effective state.

This example does skip some options that would improve resiliency in exchange for network security. For example we would normally recommend using Simple Queue Service to manage the jobs (Lambda would send them over), because SQS handles situations such as an instance failing partway through processing. But this is security research, not availability focused.

Embracing the Future

This research isn't the tip of the iceberg; it's more like the first itty bitty little ice crystal on top of an iceberg, which stretches down into a deep ocean trench. But if you remember the following principles, you will be fine as you dig into securing your own cloud and hybrid deployments:

- The biggest differences between cloud and traditional networks is the combination of abstraction (virtualization) and automation. Things look the same but don't function the same.
- Everything is managed in software, providing tremendous flexibility, and enabling you to manage network security using the exact same tools that Development and Operations use to manage their pieces of the puzzle.
- You can achieve tremendous security through architecture. Virtual networks (and multiple cloud accounts) support incredible degrees of compartmentalization, where every project has its own dedicated network or networks.
- Security groups enhance that by providing the granularity of host firewalls, without relying on operating systems. They provide better manageability than even most network firewalls.
- Platform as a Service and cloud-provider-specific services open up entirely new architectural options. Don't try to build things the way you always have. Actually, if you find yourself doing that, you should probably rethink your decision to use the cloud.

Don't be intimidated by cloud computing, but don't think you can or should implement network security the way you always have. Your skills and experiences are still important, now as a base to build on as you learn all the new options available within the cloud.

About the Analyst

Rich Mogull, Analyst/CEO

Rich has twenty years experience in information security, physical security, and risk management. He specializes in cloud security, data security, emerging security technologies, and security management. Rich is the primary developer of the Cloud Security Alliance CCSK training program. Prior to founding Securosis, Rich was a Research Vice President at Gartner on the security team where he also served as research co-chair for the Gartner Security Summit. Prior to his seven years at Gartner, Rich worked as an independent consultant, web application developer, software development manager at the University of Colorado, and systems and network administrator. Rich is the Security Editor of TidBITS, on the advisory board of DevOps.com, and a frequent contributor to publications ranging from Information Security Magazine to Macworld. He is a frequent industry speaker at events including the RSA Security Conference, Black Hat, and DefCon, and has spoken on every continent except Antarctica (where he is happy to speak for free — assuming travel is covered).

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services. Our services include:

- **The Securosis Nexus**: The Securosis Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics that tells you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at <https://nexus.securosis.com/>.
- **Primary research publishing**: We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements and conform to our Totally Transparent Research policy.
- **Research products and strategic advisory services for end users**: Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessment.
- **Retainer services for vendors**: Although we will accept briefings from anyone, some vendors opt for a tighter, ongoing relationship. We offer a number of flexible retainer packages. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. Even with paid clients, we maintain our strict objectivity and confidentiality requirements. More information on our retainer services (PDF) is available.
- **External speaking and editorial**: Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.
- **Other expert services**: Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis. For more information about Securosis, visit our website: <http://securosis.com/>.