Global Knowledge®

Expert Reference Series of White Papers

An Introduction to AWS Security

An Introduction to AWS Security

Rich Morrow, Global Knowledge Instructor, Big Data and Cloud Analyst

Introduction

When moving to the public cloud, security is front and center, but many CTOs and CIOs first have to go through an education period in order to even learn the right questions to ask. In that education, they happen across both familiar questions like, "What options exist to encrypt data in transit and at rest?" and, "How does my provider lessen the risk of an internal breach at their facilities?" as well as a whole other new set of questions like, "What is a hypervisor and how does it impact security?" and "How does public cloud impact compliance and regulatory concerns?"

Compiling the long list of "known unknowns" and then crawling through hundreds of pages of formal and informal documentation seeking to answer them can be a Herculean effort consuming weeks of valuable time, and still leaving lots of loose ends at the conclusion. In this white paper, we aim to close that gap quicker and more reliably by explaining the most important aspects of AWS security and what that means to the enterprise.

General Security Best Practices

The great news about security in any public cloud is that it's largely the same as security in an on-premises data center. The two biggest attack vectors any hacker aims for are always Operating System, and Application (Code) Layers, and security in these layers is handled the same on-prem or in cloud. One locks down access via Role Based Access Controls (RBAC), keeps their systems patched, rotates passwords, obfuscates sensitive data, runs static code analysis, and performs periodic penetration and vulnerability scans in order to keep those layers secure.

Those needing to provide additional security by encrypting data on the wire or at rest will find that, in addition to the open source and proprietary tools they already use, public cloud offers several additional capabilities. Many of AWS' services, such as S3, EBS, CloudFront, and others make adding native encryption capabilities as simple as ticking off a checkbox in the web GUI. In addition, AWS provides multiple options for managing encryption keys, ranging from the very high end via Hardware Security Module or "CloudHSM," to their Key Management Service (KMS), to integration with customer-managed external Key Management Infrastructure.

When encrypting data at rest in S3, for example, one could use keys provided via any of the methods above (or even S3's native Server Side Encryption or "SSE" capability), and S3 will automatically encrypt and decrypt data on the fly, keeping the data securely encrypted at rest.

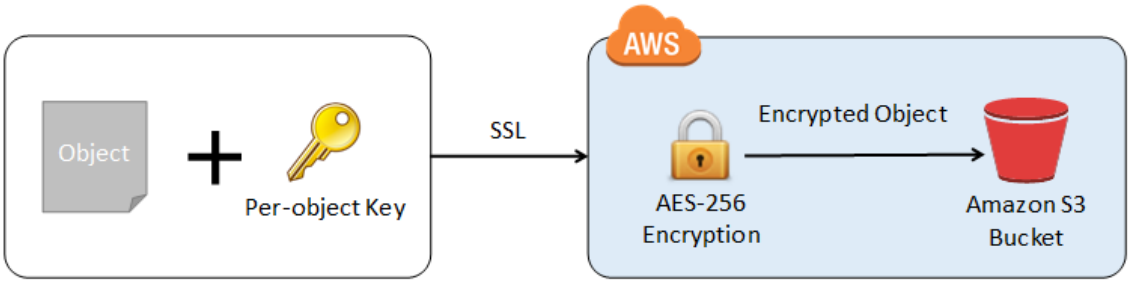


Figure 1: AWS native S3 encryption via SSE

How Cloud Security Differs

Although the list of security similarities between public cloud and on-prem data center is long, the list of differences is perhaps longer. At the top of that list is considerations around the cloud model—the fact that public cloud is a remote resource, allows no physical access, offers only abstractions (in the way of both hypervisors and API calls), and likely requires some data transfer between on-prem and cloud.

In some ways, the differences work in the customer's favor. By virtualizing the underlying hardware (Figure 2), public cloud makes it easy to implement security features like "point and click" firewalls (Security Groups), and traffic filtering (no multicast allowed) at the hypervisor level. This not only makes individual servers easier to secure, it also means that, should a server become compromised (even up to the root level account), the hacker cannot disable the firewall or change the filtering.

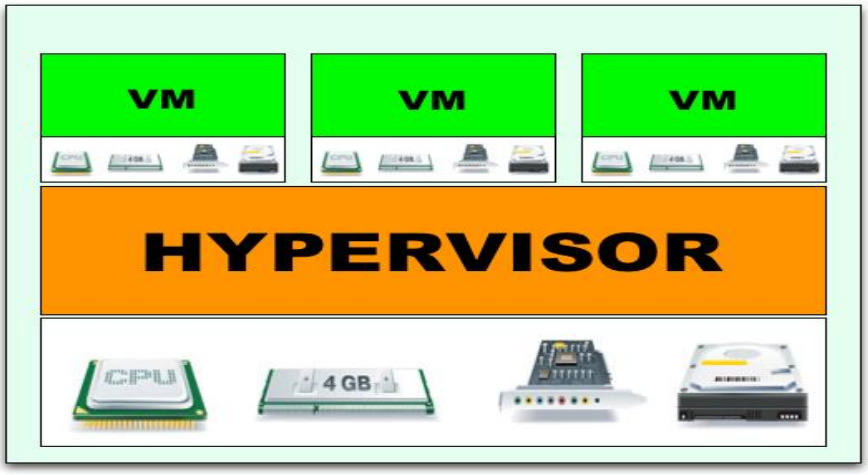


Figure 2: Hypervisor control of underlying hardware

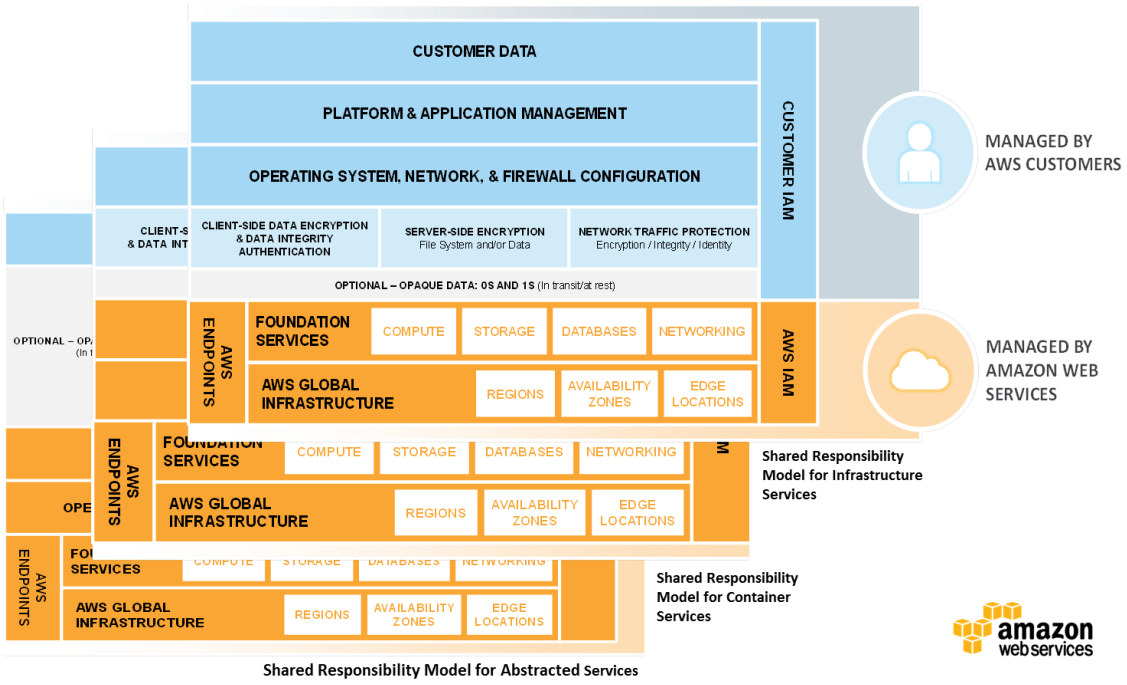
AWS' entire value proposition has always revolved around the premise that they remove the "undifferentiated heavy lifting" of an application, allowing the customer to focus on the business problem. Since security is among the most undifferentiated heavy lifting that any business will do, that benefit is especially pronounced here. By leveraging the efficiency of AWS' security experts, and the security tools they expose, customers need only learn the tools and address the cloud-specific concerns (remote systems access, hypervisor implications, key management) in order to build and deploy secure products.

AWS is also well aware that "inside jobs" account for possibly the vast majority of security breaches (some estimates say as high as 75 percent of all breaches), and they have put strong systems in place to make it nearly impossible for AWS employees to siphon customer data. AWS strictly enforces a "need-based" security model in their data centers, meaning they allow no outside access whatsoever, and have a clear logical and physical separation of duties.

Employees with logical access (those working in the Network Operations Center or "NOC," who can see that, for example Netflix' data is in server xyz), have no physical access to the data center assets (like servers, disks, switches, etc.), and vice versa. In addition, the dynamic nature of AWS workloads (with data constantly moving from machine to machine) means that in order for an "inside job" to be conducted by AWS employees, at least two individuals would A) need to both have broken moral compasses; B) need to meet each other; and C) coordinate a data breach in a short window when the physical location is known. The chances of an internal employee stealing sensitive data are hundreds, thousands, or maybe even millions of times more likely.

The Shared Security Model

AWS likes to remind customers that strong security in any public cloud requires that both the cloud provider and the customer each bear responsibility for parts of the model. The delineation between AWS responsibilities and customers is largely at the hypervisor level, with AWS being responsible for all elements below (hypervisor itself, global infrastructure, API endpoints, providing secure services), and the customer bearing responsibility for all elements above (firewalling, data protection, operating system, application, etc.).



Sharing Security Responsibility for AWS Services

Figure 3: AWS Shared Responsibility Model around Security

Public cloud, and the agility benefits it offers, allows for some interesting and strong security measures that never existed previously. What used to be an old joke amongst security professionals—"We can make your system very secure … by blocking off all access to it"—becomes a powerful reality in public cloud. Because it's easy to replicate production environments (via services like CloudFormation), operations resources can easily replicate, debug, and fix production issues outside of production. Given this capability, many customers even block SSH or RDP (remote login) access to production systems, or run completely "disconnected" HPC workloads inside of the Virtual Private Cloud (VPC). No VPC is connected to the outside world until one attaches a gateway.

In addition to the EC2 (VM) level security, it's imperative that customers leverage the Identity and Access Management (IAM) service in order to granularly control (via groups, users, and server roles) who has access to what. We'll discuss IAM in detail next.

AWS' Role in Security

AWS takes its role in security extremely seriously, and has successfully managed to create the seemingly impossible—strong security that is not overbearingly difficult to use. In addition to the many practices discussed so far, AWS also ensures that a customer's first login to an EC2 instance can only be performed by the customer themselves.

AWS accomplishes this via "EC2 Keypairs," creating the private key in Javascript in the user's browser (or, in Linux via the openssh library), and embedding the public half in the EC2 instance at launch time. Linux logins require submission of that private key (which only the user has local), and Windows logins leverage that private key to unlock the "Administrator" password. Once a customer logs into a server, they can, of course, turn off this key pair authentication, but at the very least AWS ensures that first login is as secure as it can be.

Given AWS' well-thought-out, flexible, and robust security practices and systems, it's no wonder that they have managed to achieve nearly every industrial compliance certification available. AWS has acquired PCI DSS Level 1, SOC 1,2,3, ISO 9001, HIPAA, FedRAMP, FIPS, and many other certifications, greatly smoothing the path of customers achieving the same certifications.

Your Role in Security

No matter how solid the foundation, a house built of straw on top if it, will easily blow over if/when a big bad wolf huffs and puffs. Achieving strong security is every bit as much the responsibility of the customer as it is of AWS, and in the most important layers—OS and application—the customer bears full responsibility.

Job number one of any customer in AWS should be to patch their base AMI (the software half of an EC2 instance), bringing all OS level (security, functionality) packages up to date, and then keeping those packages up to date. To achieve that patching across the potentially dozens or hundreds of EC2 instances in a customer's systems pretty much necessitates the usage of a config management tool like Puppet or Chef. To that end, AWS also provides another service, OpsWorks, which is a managed deployment of Chef built specifically for EC2.

At the application (customer written code) layer, security inside of AWS functions is nearly identical to a customer-hosted solution. Any savvy company will know that ensuring security here is a combination of many things—developer training, security reviews, static code analysis, and ongoing penetration and vulnerability scanning and mitigation. Although many organizations may currently only use a subset of those tools and processes, the increased agility of the cloud (and the increased number of code pushes, rollbacks, etc that follow) makes usage of all a highly recommended best practice.

On the vulnerability testing side, a particular tool that can be of use is Netflix' "Security Monkey" (one of the latest recruits in the Simian Army project)—a componentized security and vulnerability testing framework custom designed to test entire stacks of resources in a customer's AWS account. The widely publicized resiliency testing "Chaos Monkey" is yet another tool from Netflix' Open Source Software (OSS) Center that many customers find valuable.

As a now-defunct company called Codespaces recently discovered, controlling access to your AWS account, itself, is paramount to any another customer-side security practice. New customers to AWS should immediately use the IAM service to create individual sub-accounts, and then use only those sub-accounts going forward. Customers should then strip the access and secret keys (API signing credentials) from the AWS root account, apply Multi-Factor Authentication (MFA) to it, copy the password onto a USB drive, and lock both the MFA keyfob and the USB drive in a company safe. The root account is quite literally the key to a customer's entire AWS kingdom—allowing full access to delete all services and all data in all regions.

VPC Security Walkthrough

When deploying individual services in AWS, customers should also strive to ensure they are deploying those services, whenever possible, in the VPC. AWS' VPC is a free use service (VPC costs nothing, the assets launched inside are charged at normal rates) that wraps additional security, network topology control, and NAT/PAT capabilities around select services such as EC2, ELB, RDS, Redshift, and others. Although not all services can be launched in the VPC, the major components of most stacks (the load balancer, web, app, and database tiers) can, and should always be launched only in the VPC.

The VPC allows customers to select and use their own private address spaces, in customer-defined subnets, with ability to control routes between the subnets. Customers can also leverage Network Access Control Lists or NACLs that perform Security-Group-like ingress/egress control over the traffic in and out of each subnet.

These powerful features, (in combination with additional gateway, VPN, and dedicated instance possibilities), make the VPC a foundational design component that offers much improved security for only slightly increased architectural complexity. VPC is a must for any company, and it's one of the first services any user of AWS should learn in-depth.

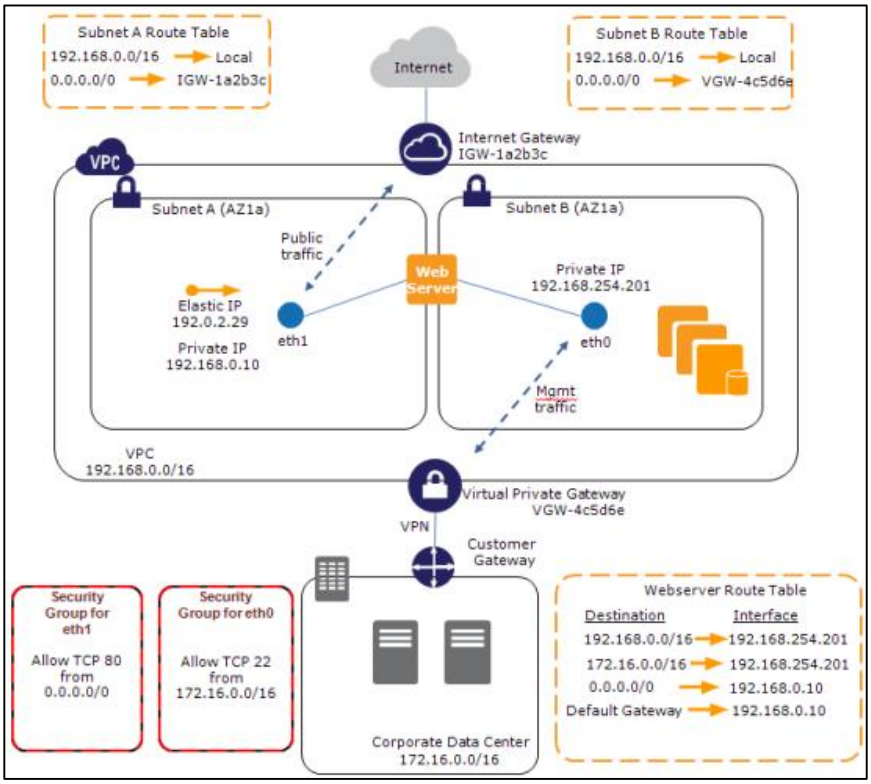


Figure 4: Complex VPC Architecture, showing subnets, route tables, and gateways

IAM Usage

Probably the first service any AWS user should familiarize themselves with is IAM—the mechanism used to control who can perform what service actions inside of a customer's AWS account. IAM allows creation of familiar Role Based Access Control (RBAC) Users and Groups, but also allows creation of Roles—a unique mechanism used primarily to control access to services from applications running inside of an EC2 instance.

IAM uses JSON-based Policy Documents (PDs) to define, on an API-call level basis, who can create, modify, delete, and otherwise control AWS services such as EC2, VPC, NACLs, etc. IAM is a "default deny" mechanism, meaning

that for an IAM user to be granted access to an API call on a particular service, either their User PD, or one of their Group PDs must contain an explicit ALLOW statement, and no other PD can contain an explicit DENY against that API call (DENY always overrides any number of ALLOWs).

Many services—such as EC2, S3, and others—also offer the ability for IAM to lock control down by individual resource or resource type, as well as by tag (tags are key/value metadata about a resource such as "ENV=PRODUCTION"). For example, one could attach a PD to the "newbies" group, which allows creation of only small, inexpensive EC2 instances like the T2.micro family, while specifying an explicit deny for creation of all other instance types. Or, alternately, one could attach a PD to the "testers" group, which only allows deletion of EC2 instances tagged with "ENV=TEST."

IAM is an extremely flexible, useful tool with many possibilities, and several well-established best practices—some of which have been discussed above. In the arms race for IT security, where hackers and enterprises constantly battle to create new methods to thwart the other, IAM is one tool for which the enemy has no countermeasure. Used properly, it can allow hundreds or even millions of internal and external users to access your sensitive data and services in a secure, auditable fashion.

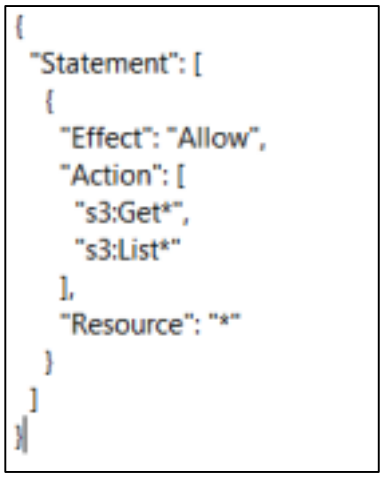


Figure 5: Example IAM JSON Policy Document

Other AWS Security Services

No paper on AWS security would be complete without at the very least giving a hat tip to a handful of other AWS-provided services. Although we don't have the space to do their explanations justice, customers should additionally explore CloudTrail (API call audit logging), AWS Config (service inventory and config history), Service Catalog (approved service catalog creation and distribution), IAM STS (useful for employee/mobile federation), Directory Service (managed AD/LDAP), Cognito (mobile identity management), and Direct Connect (dedicated, secure leased 1/10Gb lines).

Conclusion

AWS is an incredibly rich ecosystem of services and tools, some of which have security aspects baked in (like S3 SSE), and others that provide overarching security capabilities (like IAM and VPC) that apply to many services. With regard to data storage, operating system, and applications, security functions largely the same in the cloud or on-prem. Customers can and should continue to follow best practices that have served them well in their own data centers. But to ensure the best security possible, customers need to be aware of the shared responsibility model of AWS security, and pay particular attention to their role, especially with regards to EC2 and IAM.

Of the security tools AWS provides, IAM and VPC are probably the best places to start the journey, and customers would be well advised to strictly adhere to established best practices for each.

Those who know security well will all tell you the same two statements apply anywhere:

1. Security is a journey, not a destination.
2. Security isn't about making a system 100 percent secure, it's about making it so difficult to hack, that the hackers move on to easier targets.

In short, make sure you're not the weakest gazelle, and at the very least, take weekly and monthly actions to ensure you stay at the front of the pack, well ahead of predators.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

- Architecting on AWS
- Advanced Architecting on AWS
- Systems Operations on AWS
- DevOps Engineering on AWS
- Developing on AWS
- Big Data on AWS

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Rich Morrow is a 20-year open-source technology veteran who enjoys coding and teaching as much as writing and speaking. His current passions are cloud technologies (mainly AWS and Google Cloud Platform) and big data (Hadoop and NoSQL), and he spends about half of his work life traveling around the country training the Fortune 500 on their use and utility. He leads the Denver-Boulder cloud computing group, as well as quicloud, a cloud and big data consultancy firm.