By the very nature of the phrase “AWS Shared Responsibility Model,” we can see that security implementation on the AWS Cloud is not the sole responsibility of any one player, but is shared between AWS and you, the customer.

The AWS Shared Responsibility Model dictates which security controls are AWS’s responsibility, and which are yours. In short, you decide how you want your resources to sit ‘in’ the cloud (in other words, how much access you choose to give to and from your resources), while AWS guarantees the global security ‘of’ the Cloud (i.e., the underlying network and hardware they provide to host and connect your resources).

In my experience, a solid understanding of the AWS Shared Responsibility Model makes it easier to build and maintain a highly secure and reliable environment. Without knowing where I needed to step in and take control of data security, I was never able to properly define just how secure my environment really was.

AWS treats security as a major priority…and so should you

Security is AWS’s number-one priority in every sense. It’s an area into which AWS pours huge capital and energy and devotes near-constant attention.

There’s a reason for this. After speaking with my business contacts in various sectors, it seems that security is still one of the main reasons corporations are reluctant to adopt a cloud presence. Overcoming this hesitation requires AWS to be at the very top of security excellence and governance.

Having served over a million customers in the past month alone, AWS’s most stringent security standards are already being used for audit purposes by the most security-sensitive customers around. Facing so many requirements, AWS is certified and compliant across a huge range of security standards, including PCI DSS, ISO, and SOC.

AWS Services are deployed and distributed in exactly the same way throughout their entire global infrastructure. This means a single user accessing a simple S3 bucket for document backups is covered by the same strict security standards as the largest and most demanding corporations.

AWS Shared Responsibility Models

To help provide a clear definition of the boundaries of responsibility, AWS has devised 3 main models, each representing where AWS and customer responsibilities start and end:

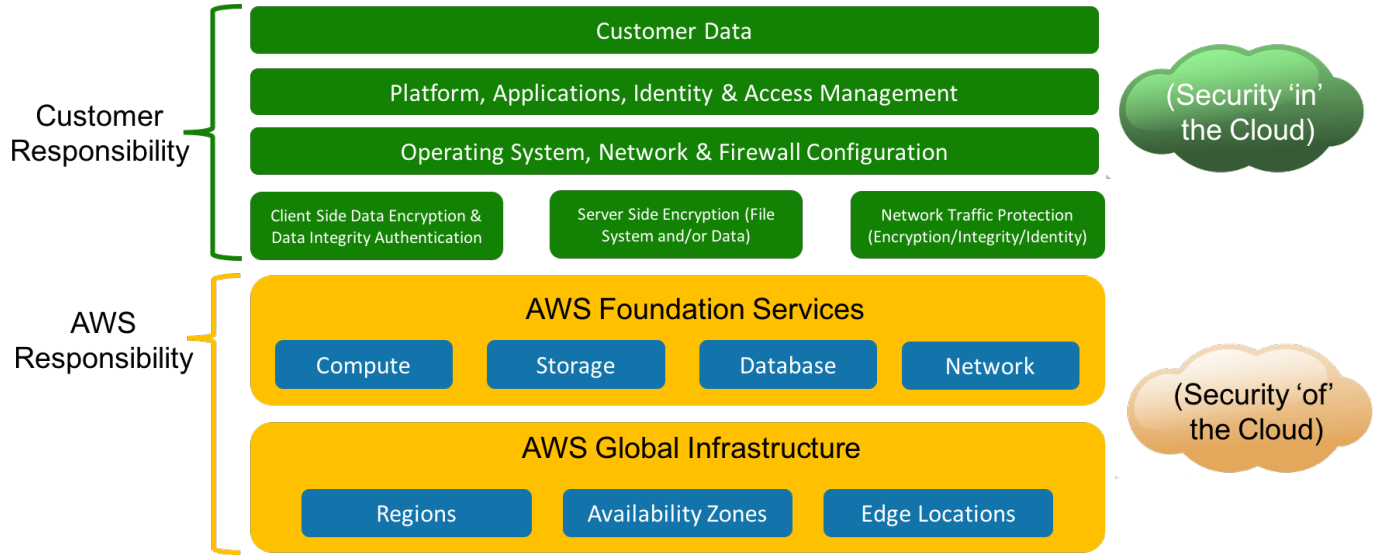
- Shared Responsibility Model for **Infrastructure Services**
- Shared Responsibility Model for **Container Services**
- Shared Responsibility Model for **Abstract Services**

By taking a look at each of these models, we will be able to clearly see the differences. Let’s start by looking at the first model, based on an infrastructure that includes services such as EC2. Then, we’ll look at how the level of responsibility shifts as we move into containers and abstract services.

For more information on the differences between container and abstract services within AWS, please see our course AWS Security Best Practices: Abstract and Container Services.

AWS Shared Responsibility Model: Cloud Security

Shared Responsibility Model: Infrastructure Services



As we said, AWS is responsible for what is known as Security 'of' the cloud. This covers their global infrastructure elements including Regions, Availability Zones, and Edge Locations, and the foundations of their Compute, Storage, Database, and Network services.

AWS owns and controls access to their data centers where your customer data resides. This covers physical access to all hardware and networking components and any additional data center facilities including generators, uninterruptible power supply (UPS) systems, power distribution units (PDUs), computer room air conditioning (CRAC) units, and fire suppression systems. Some of the security compliance controls mentioned previously are based upon this physical access entry and control. Essentially, AWS is responsible for the components that make up the cloud, any data put 'into' the cloud then becomes your responsibility.

With the basic Cloud infrastructure secured and maintained by AWS, the responsibility for what goes into the cloud falls on you. This covers both client and server side encryption and network traffic protection, security of the operating system, network, and firewall configuration, followed by application security and identity and access management.

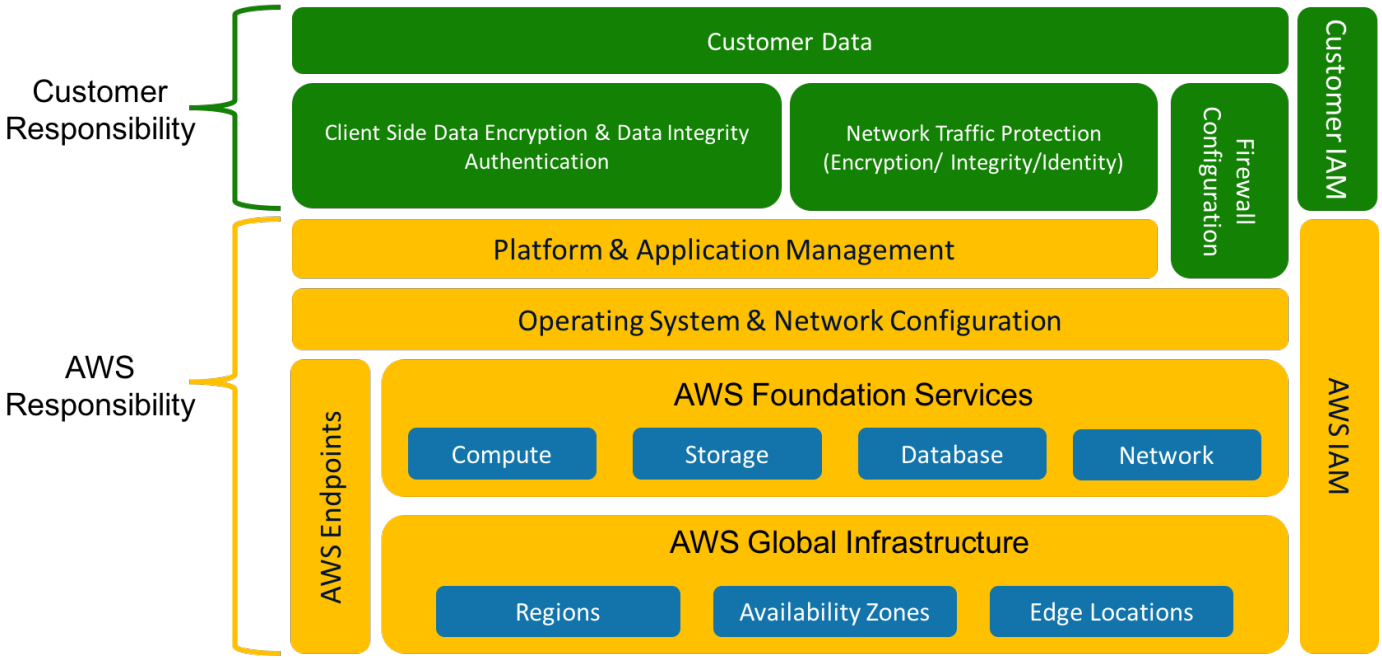
How much of this additional security you wish to implement is entirely your decision. What you choose may depend on the nature of your business or on existing controls that you may already have in place. I recommend tightening security as much as possible to minimize exposure to external threats that could compromise your environment. The important point to remember is that, while AWS provides many powerful security controls, how and when to apply them is not AWS's responsibility.

AWS Shared Responsibility Model: Cloud Security

Shared Responsibility Model: Container Services

Examples of AWS container services include:

- AWS Relational Database Service (RDS)
- AWS Elastic Map Reduce (EMR)
- AWS Elastic Beanstalk



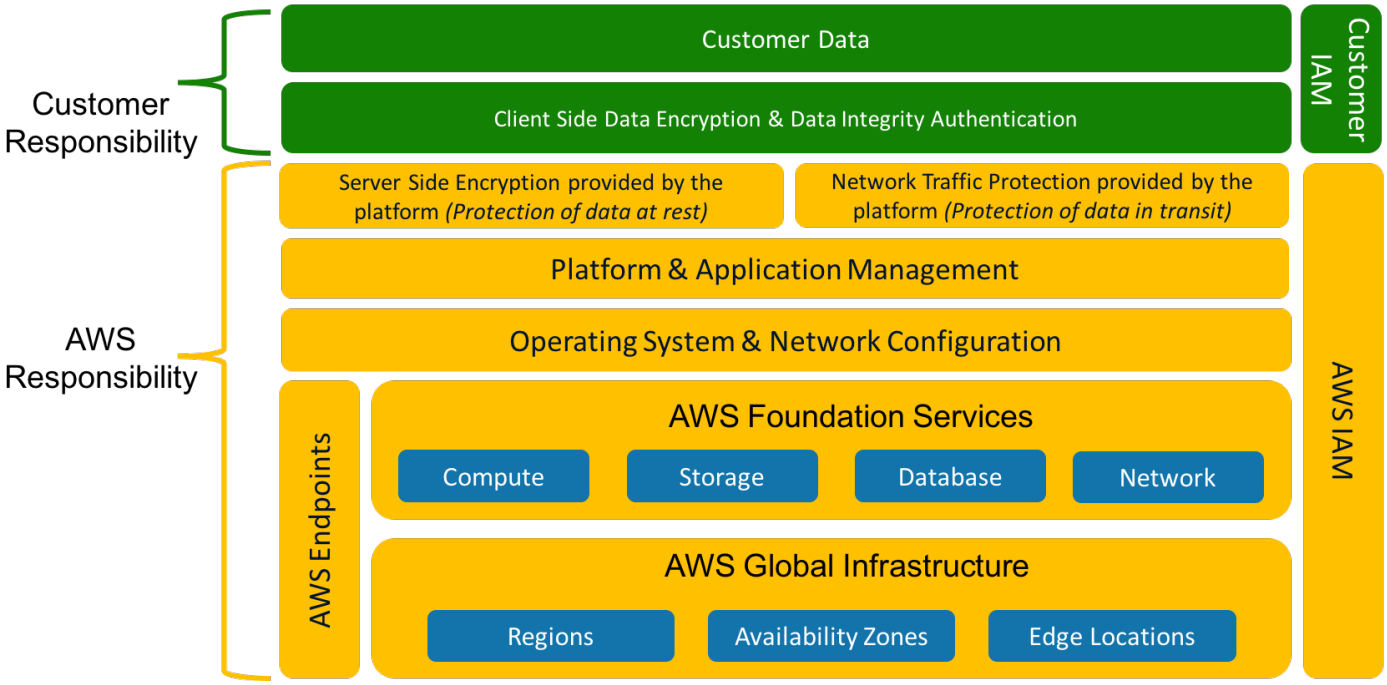
Straight away, we can see that both Platform and Application management along with any operating system or system and network configuration has shifted to being the responsibility of AWS and is no longer down to us as the customer to manage. This is a huge difference from that of infrastructure-based services. However, not all responsibility has shifted. You should note that firewall configuration remains the responsibility of the end user, which integrates at the platform and application management level. For example, RDS utilizes security groups, which you would be responsible for configuring and implementing.

Shared Responsibility Model: Abstract Services

Examples of abstract services include:

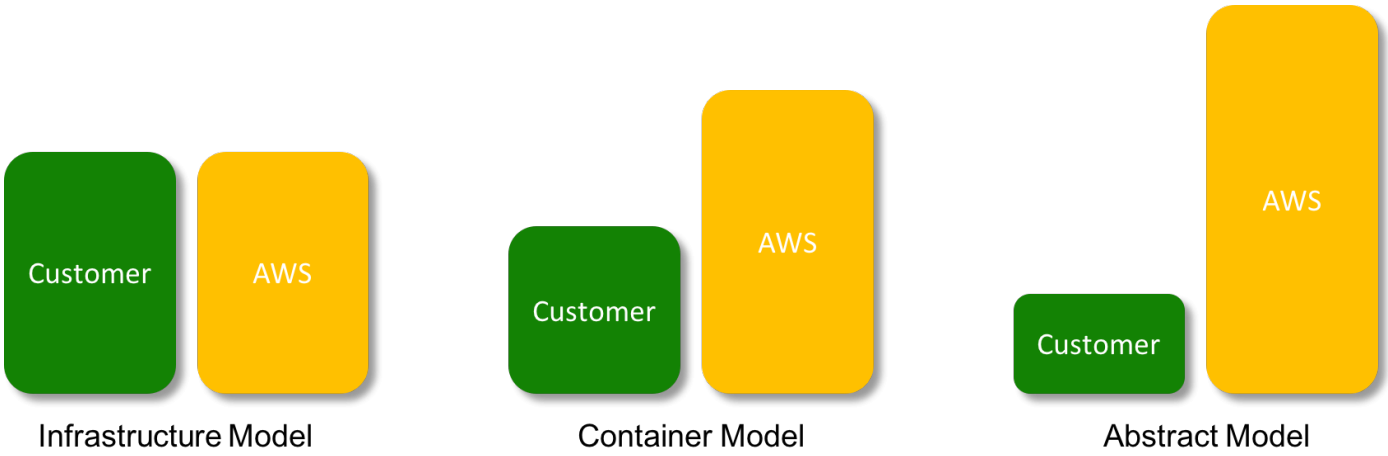
- Simple Storage Service (S3)
- DynamoDB
- Amazon Glacier
- SQS

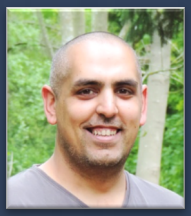
AWS Shared Responsibility Model: Cloud Security



You will notice that even more responsibility has been shifted to AWS, specifically Network Traffic protection, which AWS will manage via the platform protecting all data in transit using AWS's own network. You are also responsible for using IAM tools to apply the correct permissions both at the platform (such as S3 Bucket policies) and IAM user/group level.

As we progress through each of these models, it's clear that the level of control and responsibility shifts more toward AWS than to the customer.





Stuart Scott
AWS Content Lead

@Stuart_A_Scott
https://uk.linkedin.com/in/stuartanthonyscott