Using AWS in the Context of NHS Cloud Security Guidance

AWS Whitepaper

Using AWS in the Context of NHS Cloud Security Guidance: AWS Whitepaper

Table of Contents

Using AWS in the Context of NHS Cloud Security Guidance

Publication date: **September 29, 2021** (*Document history* (p. 28))

Guidance was issued in early 2018 on the use of hyperscale cloud services by UK public sector healthcare organisations and their business partners. The documents comprising the guidance include detailed risk management activities for such organisations to undertake, comprising mostly technical measures appropriate to the level of security required. This whitepaper provides advice corresponding specifically to the measures described, to accelerate organisational alignment with the guidance.

Introduction

The explicit guidance on the secure use of hyperscale cloud services was published in January 2018 by four key UK Public Sector Health bodies: NHS Digital, the Department of Health and Social Care, NHS England, and NHS Improvement. That guidance built on the foundation of the National Cyber-Security Centre's 14 Cloud Security Principles, and adopts the NCSC's philosophy of devolving risk management to Information Asset Owners, taking a risk-based approach to managing information security in the cloud.

The guidance also draws a clear delineation between the security of the cloud infrastructure and services delivered from it, and the workloads deployed to that infrastructure. The expectations on organisations using the guidance are that they:

- Quantify the information security risks involved for their workloads.
- Satisfy themselves that the cloud provider they use implements the required controls to manage those risks.
- Adopt the appropriate customer-usable controls for that purpose.

This whitepaper explains how to achieve the latter when using Amazon Web Services (AWS) for cloud infrastructure.

Achieving compliance

Achieving compliance falls into two main areas:

- Classifying the workload being deployed to AWS
- Implementing the class-appropriate controls

Classify workload

The set of documents comprising the Cloud Security Guidance includes the Health and Social Care Cloud Security – Good Practice Guide (shortened to the *Good Practice Guide* in this paper), which describes five different classes of risk into which a given workload may fall. The primary determinants of the applicable class are:

- The nature and volume of the data it processes
- The duration for which they are stored
- The data availability requirements (specifically as defined by NHS Digital's Service Classification in *Appendix B* of the Good Practice Guide)

The Good Practice Guide breaks down classifying a system into two stages:

- Understanding the data
- Assessing the risk

Understanding the data entails gathering the characteristics listed above that determine how the workload is classified.

To assess the risk, the NHS Guidance provides the NHS Digital Data Risk Model, which computes the risk classification.

There is an element of subjectivity and discretion in the inputs to the model, which potentially affects the workload's classification. When providing the inputs, bear in mind that the higher the classification, the more controls required to manage the risk, and hence the greater the potential cost and complexity involved, which can increase operational risk rather than reduce it. It is important to achieve an appropriate balance between the risks to which the workload is subject and the costs of managing them effectively. If deciding between the deployment of a workload on-premises and cloud, it is equally important to compare the risks for both target environments.

The risks in question sensibly cover not only the technical and organisational measures associated with managing risk, but a host of other considerations, such as public perception, lock-in risk, data repatriation, system complexity, data sovereignty, fair processing, documentation, and contracts. Though important, these are beyond the scope of this whitepaper. AWS recommends that customers who have concerns of this nature engage with their AWS account team to address these.

Implement controls

Appendix A: Detailed Advice and Guidance of the Good Practice Guide describes in detail both the security controls that AWS customers should require of a cloud provider and the controls that they should

implement when consuming that provider's services – AWS, in this case. These follow the structure of the NCSC's 14 Cyber-Security Principles, examining each in turn and detailing provider requirements under the heading *The Cloud Provider should:* and customer responsibilities under *The Service User should:*. For the remainder of this whitepaper, the AWS customer is synonymous with the *Service User*. The guidance in the Good Practice Guide recognises the concept of the Shared Responsibility Model for security in the cloud, which apportions responsibility for the security of element of the cloud and its use to the party most appropriate to manage it. In summary, AWS is responsible for the security *of* the cloud, while customers are responsible for security *in* the cloud.

This section provides prescriptive guidance on how to make concrete the required controls in AWS, specifically. It is intended to be read in conjunction with the companion AWS whitepaper *Using AWS in the Context of NCSC UK's Cloud Security Principles* (which explains how AWS fulfils its responsibility for the security of the cloud) and document "Security Controls Mapping - Health and Social Care Cloud Security" (derived directly from the guidance, and obtainable on request. To request the document, contact Compliance Support).

> **Note**
> Not all of the controls described in this section are necessarily required for a given system being deployed to AWS; those required depend on the system's Risk Classification. Refer to *Appendix A: Detailed Advice and Guidance* of the Good Practice Guide for authoritative information on which controls to apply.

Overall security governance - AWS Landing Zones

Relevant to most of the Principles covered by the Good Practice Guide, a Landing Zone is a solution available from AWS that automatically creates an environment consisting of a set of related AWS accounts configured in such a way as to establish security (and cost-related) guardrails for AWS usage by a wide variety of teams with minimum friction. The environment includes the foundations of identity management, logging and monitoring, governance, security, and network design, the specifics of which may be implemented using decisions made in examining each of the principles covered below. For more information about the solution itself, see the AWS Landing Zone page.

Principles

Principle 1: Data in transit protection

User data transiting networks should be adequately protected against tampering and eavesdropping.

The Service User should utilise strong cryptography as defined by NIST SP800-57 to encrypt communications between the Cloud and the End-user.

Applicable risk classes: III-V

There may be multiple channels of data communication between end users and the system deployed to AWS. These may be divided into two categories: those accessing the system itself as deployed into an Amazon Virtual Private Cloud (Amazon VPC), and those accessing AWS APIs outside of that VPC.

For the first category, two different controls are applicable: an IPsec VPN and a Direct Connect link to the VPC.

- **IPsec VPN** — An IPsec VPN connection connects a customer's VPC to another network designated by the customer. IPsec is a protocol suite for securing IP communications by authenticating and encrypting each IP packet of a data stream. Amazon VPC customers can create an IPsec VPN connection to their VPC by first establishing an internet key exchange (IKE) security association between their Amazon VPC, VPN gateway, and another network gateway using a pre-shared key as the authenticator.

  Upon establishment, IKE negotiates an ephemeral key to secure future IKE messages. An IKE security association cannot be established unless there is complete agreement among the parameters, including authentication (such as SHA-1) and encryption (such as AES 128-bit).

  Next, using the IKE ephemeral key, keys are exchanged between the VPN gateway and customer gateway to form an IPsec association. Traffic between gateways is encrypted and decrypted using this security association. IKE automatically rotates the ephemeral keys used to encrypt traffic within the IPsec security association on a regular basis to ensure confidentiality of communications.

  For steps describing how to establish a VPN connection between a customer environment and an Amazon VPC, see the AWS Site-to-Site VPN User Guide.
- **Direct Connect link** — AWS Direct Connect (DX) is a direct logical connection between the customer's environment from which end users are accessing the system and the VPC which it is deployed. Because this is an entirely private link, the risk of data in transit being intercepted is greatly reduced. However, to minimise it, and implement the guidance, it is possible to establish a VPN within that link. For steps, see this AWS Direct Connect Support article.

  For the second category, accessing AWS public APIs outside of a VPC, the primary applicable controls for data in transit protection are the TLS-secured API endpoints, accessible via the AWS web-based console, the AWS Command Line Interface (AWS CLI) or software development kits (SDKs) for a variety of programming languages. These may only be accessed with appropriate authentication (see Section 9.1: Authentication of [admin] users to management interfaces and support channels (p. 14) of this document for more detail).

  While these public APIs are very secure, thanks to their encryption with TLS, customers also have the option of configuring their AWS environments to access these solely from within their VPCs using VPC endpoints. For a detailed explanation and associated directions on how to use these, see VPC endpoints.

Principle 2: Asset protection and resilience

User data, and the assets storing or processing it, should be protected against physical tampering, loss, damage, or seizure.

Section 2.1: Physical location and legal jurisdiction

The Service User should only use Cloud Infrastructures to store and process data that are physically located within the UK, European Economic Area (EEA), a country deemed adequate by the European Commission, or in the US where covered by Privacy Shield.

Applicable risk classes: All

- **AWS Regions** — An AWS Region is a geographical locality from which services are delivered, and in which the underlying physical infrastructure to support these exists. The vast majority of these are fully automated, provided on a self-service basis to customers, and are the locations where systems deployed to AWS actually run. It is entirely up to customers which of these Regions they choose to use; data in AWS stay in the Region they are stored, and are moved only in automated response to customer requests, via the AWS Management Console, API, or Command Line Interface.

  At the time of writing, AWS has Regions in the UK, Ireland, Frankfurt, Paris, and Stockholm. AWS is certified under Privacy Shield, so customers subject to this requirement may also use Regions in the United States: Virginia, California, Oregon, and Ohio. For an up-to-date list of Regions, see Global Infrastructure. AWS is covered under Amazon.com's certification, confirmation of which may be found at https://www.privacyshield.gov/list.

  The Service User should review the Cloud Provider's terms and conditions (T&Cs) to ensure they are compliant with the Data Protection Act (DPA) and the General Data Protection Regulation (GDPR).

- **Compliance** — AWS Terms and Conditions are compliant with the UK's G-Cloud Framework, the Data Protection Act, and the EU's General Data Protection Regulation. For more information about compliance specifically, see AWS Compliance.

Section 2.2: Data centre security

This is entirely AWS responsibility; other than to satisfy themselves of AWS having fulfilled this, there is no action for customers to take in support of this section of Principle 2. AWS provides a wide range of third-party compliance reports that customers can use for assurance purposes. For details, see AWS Compliance Programs.

Section 2.3: Data at rest protection

The Service User should ensure that the encryption is appropriately configured when you implement the system on your chosen cloud provider.

Encryption

Customers have the option of ensuring that all data stored in AWS are encrypted. Precisely how this is done varies from one storage service to another. A basic introduction on how to go about this is provided in the paragraphs below. For more detail, see the Security Learning page.

- Amazon Elastic Block Store (Amazon EBS) presents storage as disk volumes to Amazon Elastic Compute Cloud (Amazon EC2) virtual machine instances. To encrypt these volumes, customers need only to check a box or set the appropriate API or command line parameter to enable this. Snapshots of those volumes are also automatically encrypted. The encryption key is unique to the volume in question, and is in turn encrypted by a primary key, protected by the AWS Key Management Service (AWS KMS). For details on how to implement this, see Amazon EBS encryption.
- Amazon Relational Database Service (Amazon RDS) is a service for customers that require managed relational databases, and offers various encryption options, depending on the RDBMS engine chosen. For details, see Encrypting Amazon RDS resources.
- Amazon DynamoDB is the AWS NoSQL database service, enabling unstructured and semi-structured data to be written and retrieved in a flexible fashion. To encrypt data stored in DynamoDB, see Amazon DynamoDB Encryption at Rest.
- Amazon Simple Storage Service (Amazon S3) is the AWS object storage service, available in the form of configurable *buckets* (customer-named containers for those objects). Customers have the option of

specifying that specific objects or entire buckets be encrypted. For details, see Protecting data using encryption.

- Amazon S3 Glacier is an archive storage service. Data stored in Amazon S3 Glacier are encrypted as standard, using the 256-bit Advanced Encryption Standard (AES-256).
- Amazon Elastic File System (Amazon EFS) is a shared file system, presented over the NFS protocol. To encrypt data stored on this service, see Data encryption in Amazon EFS.
- AWS Storage Gateway presents network-accessible volumes or virtual tape libraries to client applications, storing the data for these in Amazon S3 in encrypted form as standard. Customers have the choice over whether to use keys managed by AWS Key Management Service (AWS KMS) to encrypt data, or Amazon S3's Server-Side Encryption (SSE) keys. For more details, see Data encryption using AWS KMS.
- AWS Snowball transfers large volumes of data between on-premises environments and AWS. Customers may use the AWS Snowball service, which provides a robust portable appliance with 80 TB or more of storage to which customers can copy data for transfer to their AWS environment via physical shipment. Data is written to a Snowball with AES-256 encryption as standard, and the keys are never stored on the device. For more detail, see Security in AWS Snowball.

Section 2.4: Data sanitisation

> The process of provisioning, migrating, and deprovisioning resources should not result in unauthorised access to user data. The NHS recommends explicit overwriting of storage before reallocation.

There are no obligations on the Service User in this section, since this activity falls to AWS under the Shared Responsibility Model for Security. Details of the standards to which AWS adheres when performing this are available in the AWS Artifact portal.

Section 2.5: Equipment disposal

> Once equipment used to deliver a service reaches the end of its useful life, it should be disposed of in a way which does not compromise the security of the service, or user data stored in the service. The NHS recommends that a recognised standard for equipment disposal is followed.

There are no obligations on the Service User in this section either.

Section 2.6: Physical resilience and availability

> Services have varying levels of resilience, which will affect their ability to operate normally in the event of failures, incidents, or attacks. A service without guarantees of availability may become unavailable, potentially for prolonged periods, regardless of the impact on customer business.
> The NHS recommends that the service provider commits to a Service Level Agreement (SLA) AND analysis of the service design.

A growing number of services on AWS have Service Level Agreements (SLAs), details of which are specific to each service. It is not feasible to provide an exhaustive list here, but following are links to these details for a subset of services:

- Amazon EC2 Service Level Agreement
- Amazon S3 Service Level Agreement
- AWS Direct Connect Service Level Agreement
- AWS Lambda Service Level Agreement
- Amazon Neptune Service Level Agreement
- Amazon RDS Service Level Agreement

The Service User should design for failure. Solutions should be architected for cloud such that they are resilient regardless of the underlying cloud infrastructure.

Applicable risk classes: All

- **AWS Well-Architected Framework** — AWS offers the Well-Architected Framework to design and implement workloads in the cloud using AWS best practices. The Framework entails five Pillars, likened to aspects of the workload to consider when evaluating it. One of these is the Reliability Pillar, which provides detailed guidance as to how to design and implement robust systems in the AWS Cloud.

  The accompanying AWS Well-Architected Tool is now available to customers in the AWS Management Console to enable them to evaluate their workloads against best practices, with the help of with a Well-Architected Partner, if desired.

  In addition, when deploying third-party software into AWS, customers should follow vendor-recommended High Availability (HA) architecture practices.

The Service User should use at least one Availability Zone/Data Centre.

Applicable risk classes: I-II

This recommendation applies to workloads with relatively low availability requirements, for which the trade-off of uptime and cost is biased towards the latter. Examples include Development and Test workloads. AWS recommends that customers use the principles of Infrastructure as Code to at least be able to reproduce to another Availability Zone resources deployed to an unusable one in the event of failure.

The Service User should have resilient network links to the Availability Zone/Data Centre.

Applicable risk classes: I-II

Since the impact of connectivity failure to the chosen AWS Region would be to render the workloads in question inaccessible, it is highly advisable to establish resilient connectivity. For workloads tolerant to network performance variations, this may be achieved very cost-effectively over existing internet connections, using an AWS Site-to-Site VPN.

Those requiring more consistent network performance, but which are still cost-constrained may make use of an AWS Direct Connect link, with a VPN connection as a fallback. Finally, it is possible to make use of a Direct Connect Link Aggregation Group (LAG) using multiple connections for both added resilience and consistent performance even in the event of the failure of one connection.

For more details, see the Building a Scalable and Secure Multi-VPC AWS Network Infrastructure whitepaper.

The Service User should use multiple Availability Zones/Data Centres.

Applicable risk class: III

In cases where availability is more important, the impact of services failing in a given Availability Zone is correspondingly greater, the measures appropriate to managing the risk are more extensive. Every AWS Region includes at least two Availability Zones, located in physically separate locations with independent risk profiles, but geographically near enough to support synchronous data replication. This affords customers the flexibility to deploy workloads with higher availability requirements to multiple Availability Zones without potential compromise to data integrity.

For more information, see the Well-Architected Framework Reliability Pillar and AWS Global Infrastructure.

The Service User should have resilient network links to each Availability Zone/Data Centre.

Applicable risk class: III

Whether using an AWS Site-to-Site VPN (which is configured by default with two tunnels) or a Direct Connect Link Aggregation Group with multiple connections, resilient connectivity to an AWS Region is automatically available to all Availability Zones in that Region. All that is required to make use of it is to deploy resources to more than one Availability Zone.

The Service User should use different cloud vendors or multiple Regions from the same vendor.

Applicable risk classes: IV-V

This recommendation addresses the possibility of an entire Region becoming unavailable for a time, stipulating the use of other cloud service providers or additional AWS Regions to manage this risk. While the first of these is beyond the scope of this whitepaper, the second is covered in some depth in the *Multi-Region Scenarios* section in the Reliability Pillar of the AWS Well-Architected Framework. It covers a range of Recovery Time and Recovery Point availability goals, discussing the implications of each in terms of both uptime and cost.

The Service User should have resilient network links to each Region / vendor.

Applicable risk classes: IV-V

This is achievable in two ways:

- By using the Direct Connect gateway, which enables fault-tolerant connectivity to multiple AWS Regions from one resilient set of Direct Connect links; and
- By establishing separate Direct Connect links to multiple AWS Regions using different Points of Presence, peering the VPCs in these Regions with one another to facilitate data synchronisation. See the Multiple Region Multi-VPC Connectivity AWS Solution Brief for more details.

The Service User should ensure their system has DDoS protection. This may be provided by the Cloud vendor or a third party.

Applicable risk classes: IV-V

For Distributed Denial of Service (DDoS) protection, customers may use AWS Shield, a managed service to help prevent DDoS attacks and minimise their impact. It is available at two tiers: AWS Shield Standard (protecting against all known layer 3 and 4 attacks), and AWS Shield Advanced (providing protection against application layer attacks and associated charge spikes for a number of AWS services, and access to the AWS DDoS Response Team). There is no charge for the Standard tier of the service.

Principle 3: Separation between users

A malicious or compromised user of the service should not be able to affect the service or data of another.

Applicable risk classes: All

The Service User Should undertake end-to-end penetration testing of the solution.

- **Separation of customer environments** — AWS provides robust boundaries between different customer accounts and the resources they contain. The means by which this is accomplished are detailed in the SOC 2 reports for the relevant services, available to customers under nondisclosure agreement (NDA).

  AWS Customers may also benefit from the advanced isolation properties of the Nitro System, including the Nitro Security Chip, hardware EBS processing, hardware support for the software-defined network inside each VPC, and hardware support for local storage. The Nitro system builds hardware implementations of components that are typically found in software-based virtualization technology, offloading the work from the processors used by customers, increasing performance, and raising the bar for security.

- **Penetration testing** — AWS customers can perform penetration testing as a risk identification measure, but for certain services, they must notify AWS before doing so. Otherwise, the test may be regarded by the automated threat response systems as a genuine attack, which may result in action that has a negative impact on customer systems.

  For details of current policy and procedures, see Penetration Testing.

The Service User should implement a GPG13 compliant Protective Monitoring solution.

AWS offers various monitoring services enabling customers to implement protective and holistic solutions in line with businesses risks and expectations and tailored for specific applications or system risks.

These solutions can cover monitoring phases such as event collection, alerting and notification, compliance, management and operational reporting, incident response, and forensics, in addition to monitoring infrastructure-related controls such as integrity protection and retention.

A high-level overview of the Native AWS Security-Logging Capabilities can be found on the associated AWS Answers page.

Amazon GuardDuty is a threat detection service that continuously monitors for malicious or unauthorised behaviour to help customers protect customer AWS accounts and workloads.

AWS Security Hub provides a comprehensive view of high-priority security alerts and compliance status across AWS accounts.

The services above can be combined with Amazon Simple Notification Service (Amazon SNS) to receive notifications, and with AWS Lambda and AWS System Manager for automated alert responses.

For more information on protective monitoring, see Section 5.3: Protective monitoring (p. 12) and Section 5.4: Incident management (p. 13) in this document.

Principle 4: Governance framework

The service provider should have a security governance framework which coordinates and directs its management of the service and information within it. Any technical controls deployed outside of this framework will be fundamentally undermined.

The responsibility for fulfilling this requirement falls solely on AWS; the customer need do no more than satisfy themselves that AWS does so.

Principle 5: Operational security

The service needs to be operated and managed securely in order to impede, detect, or prevent attacks. Good operational security should not require complex, bureaucratic, time consuming, or expensive processes.

Section 5.1: Configuration and change management

You should ensure that changes to the system have been properly tested and authorised. Changes should not unexpectedly alter security properties.

AWS offers its customers several methods to help configure and manage infrastructure deployed on the AWS Cloud. These methods range from AWS managed configuration management services to third-party AWS Partner Network (APN) products.

There are several best practices to consider when managing infrastructure configuration. First, it is important to understand the types of resources to manage, and their different characteristics that must be accounted for in a configuration management system. For example, the configuration and

orchestration of AWS resources (such as security groups, Auto Scaling groups, Elastic Load Balancing load balancers, and so on) is very different from that of operating system and application stack changes.

It is also important to recognise that manual configuration of distributed systems is time-consuming and error-prone, and can lead to inconsistently-configured systems. Therefore, the ability to automate, monitor, and track configuration changes is a key component of any configuration management solution.

The services that customers should consider using in particular are:

- AWS Service Catalog enables customer cloud administrators to make available an approved set of AWS services and associated configurations to be consumed. This is a preventative control to help avoid introducing security vulnerabilities through insecure deployment of services not approved for organisational use by certain stakeholder groups.
- AWS CloudFormation enables AWS resources to be described in declarative code form known as a template, with an associated development, testing, deployment, and retirement lifecycle, this simplifies the process of creating secure infrastructure by preventing security misconfigurations arising from manual actions, and ensuring consistent, repeatable deployment of initial resources and subsequent updates. This service supports the principle of immutable infrastructure, in which deployed resources are replaced with newer versions rather than updated, which benefits from security reviews built into the template development lifecycle.

Applicable risk classes: All

> The Service User should maintain an accurate inventory of the assets which make up the service, along with their configurations and dependencies.

AWS recommends using the following combination of configuration management tools to maintain an accurate inventory of services:

- AWS Config with AWS Config Rules or an AWS Config Partner to provide a detailed, visual, and searchable inventory of AWS resources, configuration history, and resource configuration compliance.
- AWS CloudFormation or a third-party AWS–resource orchestration tool to manage AWS resource provisioning, update, and termination.
- AWS OpsWorks or a third-party server configuration management tool to manage operating system and application stack configuration changes (preferably abiding by the principle of infrastructure as code).

For more detailed guidance and resources, see Infrastructure Configuration Management on the AWS Answers site.

> The Service User should ensure changes to the service are assessed for potential security impact, and the implementation of changes are managed and tracked through to completion.

Customer can conduct a security impact assessment in much the same way as for existing infrastructure. With AWS though, the application of automation should reduce the time and effort to conduct this assessment.

There are two elements to this assessment:

- The security impact of changes to the AWS environment, including the AWS services and resources used in the solution. AWS recommends adopting a Security by Design (SbD) approach to operating the environment, as described in the *Introduction AWS Security by Design* whitepaper. The result is an automated environment enabling assurance, governance, security, and compliance capabilities. This provides a reliable implementation of what was previously written in policies, standards and regulations. It minimises the effort involved in carrying out a security impact assessment, as the majority of resources will be deployed with a known, and minimised, security impact.
- The security of hosts and applications, such as operating systems, databases, and applications. The impact assessments already used today can be applied when using AWS. AWS also provides Amazon

Inspector, which is an automated security assessment service that helps improve the security and compliance of applications deployed on Amazon EC2.

Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, it produces a detailed list of security findings prioritised by level of severity. These findings can be reviewed directly or as part of detailed assessment reports which are available via the Amazon Inspector console or API. This makes it easy to build these into a DevOps process, decentralising and automating vulnerability assessments, and empowering development and operations teams to make security assessment an integral part of the deployment process.

Section 5.2: Vulnerability management

You should identify and mitigate security issues in constituent components.

Applicable risk classes: All

The Service User should undertake patching or vulnerability management for the guest operating system and application components, within the NCSC best practice timescales set out below:
a. 'Critical' patches should be deployed within 24 hours.
b. 'Important' patches should be deployed within 2 weeks of a patch becoming available.
c. 'Other' patches deployed within 8 weeks of a patch becoming available.

AWS provides facilities to patch the operating systems that customers run in their AWS environments. Where applicable, AWS will also undertake patch management on the customer's behalf for certain managed services.

AWS Systems Manager Patch Manager automates the process of patching managed EC2 instances with security-related updates. Patches for non-security updates can also be deployed to Linux-based instances. Fleets of EC2 instances and on-premises servers and virtual machines (VMs) can also be patched using this service. Applicable operating systems include supported versions of Microsoft Windows, Ubuntu Server, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Amazon Linux, and Amazon Linux 2. Instances can be scanned to establish which patches are missing, if any, and optionally, all missing patches can then be automatically installed.

While customer-operated applications deployed to AWS can be patched and updated using a conventional manual approach, there are additional benefits to automation, such as the ability to perform blue/green deployments for greater application availability during patching as well as upgrades. Customers can choose to host patching solutions provided by AWS Partners and third-party solutions with which they are already familiar. Some customers establish repositories within their AWS environment that automatically fetch patches from software vendors; this limits the number of internal instances (and other servers) connecting to external repositories.

Note that for both EC2 and application-patching AWS recommends testing the patches in a separate test environment, and also putting mechanisms in place to roll back patches in the event that they cause issues with the system being updated.

The recommended approach from AWS to patching for maximum security risk control is to make use of the principle of immutable infrastructure: namely, that instead of patching running EC2 instances, customers should launch replacement instances from Amazon Machine Images (AMIs) to which the required patches have already been deployed, and destroy the unpatched ones. This minimised the probability of "configuration drift" and deviation from desired patch status.

AWS offers a substantial and growing number of services for which the line of security responsibility with the customer is drawn at a higher level than the hypervisor. Amazon Relational Database Service (Amazon RDS) is one example of such a service. AWS performs automated patching of the database engine, underlying operating system, and certain plugins. Customers configure a time window in which

the RDS service can take the database offline, perform the update, then bring it back online. This reduces the required customer effort for managing the security of their databases.

Another example is AWS Lambda, a service that enables customers to deploy very granular application functions comprising microservices architectures or other application capabilities to the cloud: not only does AWS carry out regular automated patching of the operating system, but also the containers in which the functions run and the language runtimes they use.

In this case, customer patching responsibility covers any third-party libraries used in the code for their Lambda functions. An additional security benefit of using Lambda to run code is that functions have a finite maximum lifetime (15 minutes at the time of writing), so even if a function is compromised during run, the window of opportunity for an attacker to exploit this is greatly reduced.

Applicable risk classes: III-V

> The Service User should undertake regular (minimum yearly) penetration testing.
> The Service User should ensure that the penetration test is well scoped such that 'security vulnerabilities in the Operating system and components above' are fully tested.

Penetration testing can be performed as it would on systems deployed to non-AWS environments, such as customers' own premises. Note, however that the AWS Acceptable Use Policy describes permitted and prohibited behaviour on AWS and includes descriptions of prohibited security violations and network abuse. Because penetration testing and other simulated events are frequently indistinguishable from these activities, AWS has established a security testing policy that customers need to abide by when conducting penetration tests and vulnerability scans to or originating from the AWS environment.

Section 5.3: Protective monitoring

You should put measures in place to detect attacks and unauthorised activity on the service.

Applicable risk classes: III-V

> The Service User should put in place monitoring solutions to identify attacks against their applications or software.

AWS provides services that provide monitoring of resources deployed to customer AWS environments and service usage to help identify attacks against applications. Attacks may come from either external or internal sources and may be malicious or accidental.

Attacks from external sources come in forms such as Distributed Denial of Service (DDoS). AWS provides flexible infrastructure and services that help customers implement strong DDoS mitigations and create highly available application architectures that follow AWS Best Practices for DDoS Resiliency. These include services such as Amazon Route 53, Amazon CloudFront, Elastic Load Balancing, and AWS WAF to control and absorb traffic, and deflect unwanted requests. These services integrate with AWS Shield, a managed DDoS protection service that provides always-on detection and automatic inline mitigations to safeguard web applications running on AWS. For more details, see the *AWS Best Practices for DDoS Resiliency* whitepaper.

AWS also provides attack identification with Amazon GuardDuty, a managed threat detection service that continuously monitors for malicious or unauthorised behavior to help customers protect their AWS accounts and workloads. It monitors for activity such as unusual API calls or potentially unauthorised deployments that indicate a possible account compromise. GuardDuty also detects potentially compromised instances or reconnaissance by attackers.

Customer can also deploy some or all of their existing attack detection tools into AWS or purchase new solutions from the AWS Marketplace or Amazon Partner Network.

An advantage of the cloud is that if an attack is detected, automated action can be taken to reduce or mitigate the impact. For example, if a virus is detected on an EC2 instance, response code can be

automatically run that quarantines the affected instance for forensic analysis and sets up a clean one in its place to ensure continuity of operations.

Section 5.4: Incident management

Ensure you can respond to incidents and recover a secure, available service.

Applicable risk classes: III-V

The Service User should put in place monitoring solutions to identify attacks against their applications or software.

See Section 5.3: Protective monitoring (p. 12).

Applicable risk classes: III-V

The Service User should have an incident management process to rapidly respond to attacks.

An incident is an unplanned interruption to an IT service or reduction in the quality of an IT service. Through tools such as AWS CloudTrail, Amazon CloudWatch, AWS Config, and AWS Config Rules, customers can track, monitor, analyse, and audit events. Customers may also want to bring existing, application-specific event monitoring tools already familiar to them. Finally, AWS provides a Service Health Dashboard that allows customers to subscribe to an RSS feed and be notified of interruptions to each individual service.

If these tools identify an event, which is analysed and qualified as an incident, that "qualifying event" will raise an incident and trigger the incident management process and any appropriate response actions necessary to mitigate the incident. These qualifying events will need to be integrated into existing incident management processes.

Should customers detect a vulnerability or have a security concern regarding AWS, they may want to report it as part of their incident management process. For details, see Vulnerability Reporting.

For further guidance, see Building a cloud-specific incident response plan on the AWS Government, Education, and Nonprofits Blog.

Principle 6: Personnel security

Where service provider personnel have access to your data and systems you need a high degree of confidence in their trustworthiness. Thorough screening, supported by adequate training, reduces the likelihood of accidental or malicious compromise by service provider personnel.

The Service User should ensure IT admin staff are strongly authenticated.

Applicable risk classes: III-V

The AWS Identity and Access Management (AWS IAM) service offers flexible user authentication options, including password policies covering aspects such as required length and complexity, expiry, reuse restrictions, and so on, and the option to use multiple factors. This service is described in more detail under Principle 9.

The Service User should have a suitable auditing solution is in place to record all IT admin access to data and hosting environments.

Applicable risk classes: III-V

The AWS CloudTrail service, described in greater detail in Principle 13, provides the basis for an auditing solution to record such access. It may be configured to capture AWS sign-in and API call events, and access to data stored in Amazon S3 buckets. In addition, the CloudWatch Logs service can be used to

log instance-level data access, such as configuration files, etc. Finally, partner products from the AWS Marketplace can fulfil more specialised requirements.

Principle 7: Secure development

> Services should be designed and developed to identify and mitigate threats to their security. Those which aren't may be vulnerable to security issues which could compromise your data, cause loss of service or enable other malicious activity.

Applicable risk classes: III-V

The requirements of this principle are satisfied entirely by the AWS; the customer bears no responsibility for fulfilling Principle 7.

The fulfilment of this principle is a joint effort between AWS and the customer under the Shared Responsibility Model for Security. AWS goes to great lengths to protect the security of the various services that customers consume (providing security *of* the cloud), and provides customers with a rich set of tools to employ to be secure *in* the cloud. The majority of this whitepaper is devoted to describing the tools available for this, and which aspects of security they are aimed at.

Customer responsibility for secure development in particular extends beyond the AWS and third-party technology used for this, into the processes and methodologies (such as DevSecOps) that govern it. Advice for putting this in place is available in the AWS Cloud Adoption Framework – Security Perspective.

Principle 8: Supply chain security

> The service provider should ensure that its supply chain satisfactorily supports all of the security principles which the service claims to implement.

Applicable risk classes: III-V

There is no customer action necessary to fulfil the requirements of this principle; these are AWS' responsibility.

Principle 9: Secure user management

> Your provider should make the tools available for you to securely manage your use of their service. Management interfaces and procedures are a vital part of the security barrier, preventing unauthorised access and alteration of your resources, applications, and data.

Section 9.1: Authentication of [admin] users to management interfaces and support channels

> In order to maintain a secure service, [admin] users need to be properly authenticated before being allowed to perform management activities, report faults or request changes to the service.
> The Service User should ensure that a list of authorised individuals from your organisation who can use those mechanisms is maintained and regularly reviewed.

To manage the AWS resources deployed, administrators need to authenticate to AWS with appropriate credentials and permissions.

The constructs available in AWS providing context for authentication and authorisation are, in descending order of granularity:

- The AWS Organization (the administrative and security hierarchy for one or more AWS Accounts), which offers Service Control Policies to set the maximum permissions envelope for AWS Accounts (see the AWS Organizations documentation for more information);

- The AWS account;
- The VPC within the AWS account; and
- Individual resources within the AWS account (separated by the dimensions of network – such as subnets – or metadata, using tags).

The Guidance makes only passing reference to authorisation, but it is imperative that this be considered in addition to authentication as part of the overall security strategy, to ensure that the appropriate privileges are assigned to the right individuals, groups, and systems accessing them.

Access to the API that manages and controls each service is governed by AWS Identity and Access Management (IAM). IAM provides customers with the granular controls and features necessary for them to have confidence that authenticated and authorised users can access specified services and interfaces with the appropriate degree of privilege. It enables customers to create multiple users and manage the permissions for each of these users within their AWS Accounts.

An IAM user is an identity (within an AWS account) with unique security credentials that can be used to access AWS Services. To simplify administration, IAM users can be included in Groups for those with the same access requirements. Users can also take on Roles, reflecting a temporary change to their access privileges for specific tasks (see below for more detail about this capability).

These identities and associated permissions are the means for customers to authenticate and authorise individuals in their organisation to perform administrative and other tasks in customer AWS environments. IAM therefore eliminates the need to share passwords or access keys, and makes it easy to enable or disable a user's access as appropriate.

Access privileges are assigned through IAM policies, which express the precise permissions that the policy permits (or denies). These policies are then assignable to Users, Groups and Roles (as well as other kinds of AWS entities). There are several different types of policies, applicable in different contexts. Details are provided in Policy types, but the most important types are introduced here:

- Identity-based policies are those associated with Users, Groups, and Roles.
- Resource-based policies are assigned to AWS resources (such as S3 buckets).
- Permissions boundaries express the outer limits of permissions that any given IAM user or role could *potentially* have, to guard against overly-permissive policies being inadvertently applied to these entities. Note that this does not set permissions limits on Resource-based policies.

API access delegation — AWS supports two very important and powerful use cases with IAM *roles* in combination with IAM *users*: cross-account API access, and delegating API access within an account. These enable better control and simplify access management, by helping customers avoid having to share long-term security credentials for cross-account access or more privileged access within an account.

When an IAM user assumes an IAM role, a set of temporary security credentials is assigned to that user, with the permissions associated with the role. It is these temporary credentials that are then used instead of the user's long-term credentials in calls to AWS service APIs. Users interact with the service with the permissions granted to the IAM role assumed, rather than those assigned to their IAM identity (or Groups it may be a member of). This reduces the potential attack surface area by minimising the number of user credentials created and managed, and the number of passwords users have to remember. For details, see IAM roles.

Another important IAM capability to use is Policy Conditions — source IP address, timestamp, tags, and so on, apply to all services, and others are service-specific, such as attributes in a DynamoDB table, AWS resource Names, and so on.

The Security best practices in IAM page of the IAM documentation provides detailed advice for this and other aspects of that service.

For authorization, tag-based access-control provides a powerful and flexible means of controlling permissions in the AWS environment; see the *AWS Tagging Strategies* whitepaper and the Controlling access to and for IAM users and roles using tags section of the IAM documentation for additional guidance.

For certain services (such as S3), the API also covers some or all aspects of its function, as well as its configuration and management; hence there are service-specific access control mechanisms. The following list covers just a small subset of those available (as new services are released regularly), so customers should refer to the documentation specific to each of the services under consideration:

- **Amazon EC2** — In addition to the IAM permissions and roles assignable to EC2 instances, inbound remote terminal connections to them using Secure Sockets Shell (SSH) may be authenticated using SSH key pairs for EC2. For details, see Amazon EC2 key pairs and Linux instances.

  The access control capabilities of operating systems available for EC2 are extensive, but fall outside the scope of this whitepaper, so customers are encouraged to consult the available third-party documentation for these to manage access at that level.

  The recommended approach from AWS, however, is to employ the AWS Systems Manager service to reduce or eliminate altogether any manual human access to EC2 instances. Features in this service include (but are not restricted to):
  - **Run Command and Automation**, which enables management operations to be undertaken on instances without the need for inbound terminal access
  - **Inventory**, which collects data about instances and software installed on them; Distributor, which deploys applications and other software to them
  - **Patch Manager**, a feature to deliver operating system and application software patches, and report on patching status
  - **State Manager**, to enable consistent instance configuration

  For more details, see the AWS Systems Manager documentation.
- **Amazon EBS** — Volumes attached to EC2 instances are secured using the mechanisms available in the operating system those instances are running; this is outside of this whitepaper's scope.

  Detached volumes are secured using IAM, as described previously.
- **Amazon S3** — Access control relies on a combination of Resource Policies (which in turn use AWS IAM service – see above for details) and Bucket Policies to achieve this. For details, see Identity and access management in Amazon S3.
- **Amazon RDS** — For the majority of RDBMS engines available in this service, access to the data stored in RDS is secured through the mechanisms specific to those engines. As an application-level consideration, this falls outside of the scope of this whitepaper. See the engine-specific documentation to achieve this.

  At the time of writing, the exceptions to this are RDS MySQL and RDS PostgreSQL, for which customers can choose to manage authentication using IAM. See IAM database authentication for MySQL and PostgreSQL for details.
- **Amazon DynamoDB** — Control of access to data stored in this service uses AWS IAM. For details, see Overview of Managing Access Permissions to Your Amazon DynamoDB Resources.
- **Amazon EMR** — There being several components to this service, access to each of these is controlled in a way appropriate to that component. Inbound remote terminal connections are made to cluster instances using Secure Sockets Shell (SSH), and authentication can use either Kerberos or SSH key pairs for EC2, as described in Use an Amazon EC2 key pair for SSH credentials. Access to the EMR File System (EMRFS) is controlled using IAM Roles, as described in Configure IAM roles for EMRFS requests to Amazon S3.
- **AWS Internet of Things (IoT) Core** — Access to the service itself and its associated device shadows is controlled using IAM. The devices themselves may authenticate to the IoT broker using IAM credentials, X.509 certificates, Amazon Cognito identities, or custom tokens.

- **Machine learning (ML) services** — This category of services (including Amazon SageMaker, Amazon Comprehend Medical, Amazon Forecast, Amazon Lex, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Translate, Amazon Transcribe, and so on) generally has access controlled using IAM, as previously described.

  End user computing services — These include Amazon WorkSpaces and Amazon AppStream 2.0 (complementary client application delivery services), Amazon WorkDocs and Amazon WorkLink (a service enabling secure remote access to non-internet-facing corporate resources, whether in the cloud or on-premises). The first three of these are also secured using IAM, as described in Identity and access management for WorkSpaces, Identity and Access Management for Amazon AppStream 2.0, and Prerequisites for Amazon WorkDocs.

  Access to Amazon WorkLink is managed by identity federation with a third-party Identity Provider such as Active Directory Federation Services, Ping One Federation Services, or Okta, as described in Configure your identity provider (IdP). For Amazon AppStream 2.0, see Single Sign-on Access (SAML 2.0).

  It is also possible to control access to Amazon AppStream 2.0 via User Pools. For details, see AppStream 2.0 User Pools.

  The Service User should use 2FA to obtain access to the system.

Applicable risk classes: III-V

If using AWS IAM as the identity provider for AWS administrative access, additional factors (such as hardware or software tokens) may be registered against each identity in the IAM service.

The Service User should configure logging of access attempts.

The AWS CloudTrail service, described in detail in Principle 13: Audit information for users, monitors sign-in attempts to the AWS Management Console and access attempts to the AWS APIs. Customers may set up CloudTrail logs to fulfil this requirement.

Section 9.2: Separation and access control within management interfaces

Many cloud services are managed via web applications or APIs. These interfaces are a key part of the service's security. If [admin] users are not adequately separated within management interfaces, one [admin] user may be able to affect the service, or modify the data of another. The Service User should ensure that authorised individuals from your organisation who can use those mechanisms are managed by the 'principle of least privilege', typically using a RBAC mechanism.

AWS IAM enables customers to implement security best practices, such as the principle of least privilege, by granting unique credentials to each user accessing resources in customers' AWS accounts and restricting their permissions to only those AWS services and resources required for them to perform their jobs. AWS IAM is secure by default; new users have no access to AWS until permissions are explicitly granted.

AWS IAM enables customers to minimise the use of AWS account credentials. After AWS IAM users have been created, all interactions with AWS Services and resources should occur using IAM user security credentials.

An important complementary means of implementing the principle of least privilege is through the AWS Systems Manager Service. This provides a set of management tools that reduces further the need to access AWS resources (such as EC2 instances) directly, and enables actions performed in the course of that access to be logged in companion AWS services such as CloudTrail and CloudWatch Logs for audit purposes.

Principle 10: End user identity and authentication

All access to service interfaces should be constrained to authenticated and authorised [end user] individuals.

Applicable risk classes: III-V

- **Two factor authentication** — If required, the customer may configure identities to authenticate using additional factors.
- **Identity federation with your existing identity provider** — If configuring federation between an existing identity provider and AWS IAM, the identity provider's two-factor authentication will operate independently of AWS, so the only AWS-specific task the customer is required to undertake is the federation itself.

Principle 11: External interface protection

> All access to service interfaces should be constrained to authenticated and authorised individuals.
> The Service User should ensure their system has AWS WAF protection. This may be provided by the cloud vendor or a third party.

Applicable risk classes: III-V

To fulfil this aspect of the Guidance, customers may avail themselves of three complementary AWS services: AWS WAF (Web Application Firewall, in conjunction with the Amazon CloudFront content distribution service, Amazon API Gateway, or an Application Load Balancer), AWS Shield, and AWS Firewall Manager.

AWS WAF helps protect applications and web service components against application-level attacks, using a customer-configurable ruleset. Request types, source IP addresses and other request properties may be configured as the criteria for allow lists or deny lists which can even be updated automatically. More specialised requirements in this category may be met by a relevant product from the AWS Marketplace. For details on how to use this service, see AWS WAF.

AWS Shield is a Distributed Denial of Service (DDoS) protection service for preventing internet-facing web applications from being overwhelmed with requests intending to compromise their availability. AWS Shield is offered at two levels: AWS Shield Standard (for which there is no additional charge) and AWS Shield Advanced. For more information, see AWS Shield.

AWS Firewall Manager provides customers with a single point of management for all AWS WAF-protected resources across their AWS accounts, enabling AWS WAF and Shield Advanced operations to be streamlined. See AWS Firewall Manager for more information.

> The Service User should ensure that the implemented design protects data by ensuring it is at least two "firewall" hops from the external network, architected in such a way that the compromise of one firewall will not affect the other.

Applicable risk classes: III-V

AWS network security includes security groups, which are logical firewalls that may be applied to single instances or sets of instances. Because of how the security group mechanism operates, this threat is less of a risk in the AWS environment, but should customers choose to implement this part of the guidance, it can be achieved by deploying one or more instances running virtual firewalls to the Virtual Private Cloud as the gatekeeper for all in- and out-bound traffic, in conjunction with security groups.

> The Service User should correctly implement firewall rulesets using the "Deny All" First and then Add Exceptions principle.

Applicable risk classes: All

- **Security group and network access control list (network ACL) rules** — These two security features implement this principle as standard; customers may add exceptions based on source IP address, range or Security Group; and source TCP port (or range). Security Groups may then be applied to individual or sets of virtual instances, thus controlling the destination addresses to which they apply. Network ACLs operate on a similar basis, except that they apply to subnets rather than instances, and are stateless in nature.

  For more details, see Internetwork traffic privacy in Amazon VPC in the *Amazon Virtual Private Cloud User Guide*.

Additional applicable security controls

- **Amazon Virtual Private Cloud (Amazon VPC)** gives customers with a secure logical section of the AWS Cloud, which provides private subnets for their instances and other resources. VPCs created by customers do not have internet access by default; an Internet Gateway component must be configured to enable this.
- **NAT Gateway** is an optional component of the VPC that enables instances deployed to subnets with no direct in- or out-bound access from and to networks outside the VPC to access the internet (for say, downloading patches).
- **Virtual Private Network** helps customers restrict access to their AWS resources in a VPC to a corporate network. Customers may set up an IPsec VPN over the internet between these using the optional Customer Gateway component of the VPC.
- **AWS Direct Connect (DX)** is a dedicated remote link to AWS that is private to the customer.

Principle 12: Secure service administration

> Systems used for administration of a cloud service will have highly privileged access to that service. Their compromise would have significant impact, including the means to bypass security controls and steal or manipulate large volumes of data.
> The methods used by the service provider's administrators to manage the operational service should be designed to mitigate any risk of exploitation that could undermine the security of the service. If this principle is not implemented, an attacker may have the means to bypass security controls and steal or manipulate large volumes of data.

Satisfying the requirements of this principle requires no action on the part of the customer; they are fulfilled by AWS under the Shared Responsibility Model for Security.

Equally, customers remain responsible for conducting securely the administration of the resources and systems they have chosen to deploy to the AWS Cloud. This whitepaper's purpose is to provide specific advice to assist customers to achieve this.

Principle 13: Audit information for users

> You should be provided with the audit records needed to monitor access to your service and the data held within it. The type of audit information available to you will have a direct impact on your ability to detect and respond to inappropriate or malicious activity within reasonable timescales.
> The Service User should use the audit data as part of an effective pro-active monitoring regime.

Applicable risk classes: All

AWS offers a service called CloudTrail that provides audit records for AWS customers, presenting audit information in the form of log files to a specified storage location (specifically, a nominated Amazon

S3 bucket). The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.

CloudTrail provides a history of AWS API calls for customer accounts, including those made via the AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services (such as AWS CloudFormation) that invoke those APIs on a customer's behalf. The AWS API call history captured by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

The log file objects written to Amazon S3 are granted full control to the bucket owner. The bucket owner thus has full control over whether to share the logs with any other parties. This feature provides AWS customers with a mechanism for investigating service misuse or security incidents.

For more details on AWS CloudTrail and further information on audit records, see AWS CloudTrail.

The other service relevant to this purpose is Amazon CloudWatch Logs, which enables events occurring on EC2 instances (under customer management in the Shared Responsibility Model for Security) to be written to log files in AWS for analysis (and response, if required, through the companion Amazon CloudWatch Events feature). This service is also used for longer-term storage of CloudTrail records.

Principle 14: Secure use of the service

> The security of cloud services and the data held within them can be undermined if you use the service poorly. Consequently, you will have certain responsibilities when using the service in order for your data to be adequately protected.

Cloud computing introduces a significant shift in how technology is obtained, used, and managed, and each organisation's cloud adoption journey is unique. To successfully achieve cloud adoption, customers need to understand their organization's current state, the target state, and the transition required to achieve that target state. Knowing this will help set goals and create workstreams that will enable the organisation to thrive in the cloud.

As workstreams are implemented, organisations can take advantage of the AWS Cloud Adoption Framework (AWS CAF) to understand how to communicate dependencies between different stakeholders. The AWS CAF structure can also help ensure that the strategies and plans across organisations are complete and aligned to business goals and outcomes.

Maintaining Security and Compliance is an ongoing effort, due to changing application requirements, operating system patches, configurations updates, and so on, all creating a "configuration drift" away from the ideal established at the implementation, in addition to the changing threat landscape and compliance regime.

AWS offers customers the opportunity to build adaptive and highly resilient security programmes for their workloads, much more effectively than is possible using traditional infrastructure.

As per the AWS Shared Responsibility Model, and due to the self-service nature of cloud computing, a successful security programme starts from the preparation phase, before moving into design, implementation, operation, and then continuous improvement.

AWS offers customers many resources to help and guide them throughout the entire process. In addition to the AWS Cloud Adoption Framework mentioned above, customers can take advantage of the following resources:

- The AWS Well-Architected Framework;
- The associated Security Pillar;
- A variety of AWS security best practices; and
- AWS Architecture Center to common questions on a variety of topics, including account management, big data, networking, and security. The AWS Architecture Center outlines AWS best practices and

provides prescriptive architectural guidance, as well as automated solutions deployable to an AWS account in very short timescales.

These frameworks and best practices, in addition to more in-depth consultancy available from AWS Professional Services or AWS Partners, can assist customers in all phases of their cloud journey and align with the *Secure use of the service* Principle of the Good Practice Guide.

Advice for specific Guidance points:

The Service User should use a security-hardened primary operating system image to build guest servers.

Applicable risk classes: All

Instance-based services

Services that expose virtual instances to customers, such as Amazon EC2, Amazon Elastic Container Service (Amazon ECS), and so on, enable those instances to be completely controlled by the customer, and by default, provide full root access or administrative control over accounts, services, and applications on them. However, AWS recommends that customers implement a base set of security best practices, including disabling password-only access, and utilising some form of multi-factor authentication to gain access to instances (or at a minimum certificate-based SSH Version 2 access). Additionally, customers should employ a privilege escalation mechanism, with per-user logging. The publicly available Center for Internet Security (CIS) Benchmarks provide comprehensive advice on what can be done, how to do it, and how to audit it.

For example, if the guest OS is Linux, and the customer wants to enable remote terminal access, then after hardening the instance, the customer should utilise certificate-based SSHv2 for that access, disable remote root login, use command line logging, and use 'sudo' for privilege escalation. Customers should also generate their own key pairs to guarantee that they are unique and not shared with other customers or with AWS.

Virtual instances are launched from an Amazon Machine Image (AMI) selected by the customer. AWS supplies a set of default AMIs for guest operating systems supported on AWS, which customers may harden further and store as custom AMIs to use as the template for launching additional instances for use in their AWS environments.

AWS also supports the use of the Secure Shell (SSH) network protocol to enable secure login to the instances. Authentication for SSH used with AWS is via a public/private key pair to reduce the risk of unauthorised access to the instance. Customers can also connect remotely to a Windows instance with Remote Desktop Protocol (RDP) by using an RDP certificate generated for the instance.

The recommended approach from AWS, however, is not to permit direct remote terminal access to EC2 instances, but rather to make use of the AWS Systems Manager services, which enables remote management to be performed more securely and with centralised oversight, using the appropriate elements of that service. See AWS Systems Manager for more details.

Serverless services

The majority of services on the AWS Cloud platform are serverless, that is, the management of the underlying virtual instances is the responsibility of AWS, including their security (which will entail the use of security-hardened primary images). AWS recommends customers consider using these services where possible, in order to take advantage of the additional security (as well as operational agility and potential for cost reduction) available through this approach.

Examples of such services for generalised compute include AWS Fargate (for managed containers) and AWS Lambda, which integrate with natively with related services such as Amazon API Gateway, Amazon Kinesis (for applications processing streaming data), and so on.

The Service User should utilise integrated security monitoring and policy management facilities to help detect threats and weaknesses, due to poor design or misconfiguration.

Applicable risk classes: All

- **Customer notification mechanisms** — Mechanisms are in place to allow the AWS Support team to be made aware of operational issues that impact the customer experience. A Service Health Dashboard is available and maintained by the AWS Support team to alert customers to any issues that may be of broad impact. The customer-specific instance of this is known as the AWS Health Dashboard, showing a view of operational issues that may affect individual customer accounts. AWS Cloud Security is available to provide customers with security and compliance details about AWS.

  Customers can also subscribe to AWS Support offerings that include direct communication with the AWS support team and proactive alerts to any customer impacting issues.
- **AWS Trusted Advisor** — Some AWS Support plans include access to the AWS Trusted Advisor tool, which offers a one-view snapshot of the service portfolio in use and helps identify common security misconfigurations, suggestions for improving system performance, and underutilised resources.

  Trusted Advisor checks for compliance with the following security recommendations:

- Limited access to common administrative ports to only a small subset of addresses. This includes ports 22 (SSH), 23 (Telnet) 3389 (RDP), and 5500 (VNC).
- Limited access to common database ports. This includes ports 1433 (MSSQL Server), 1434 (MSSQL Monitor), 3306 (MySQL), Oracle (1521), and 5432 (PostgreSQL).
- IAM is configured to help ensure secure access control of AWS resources.
- Multi-factor authentication (MFA) token is enabled to provide two-factor authentication for the root AWS account.

The Service User should undertake annual assessment against a recognised standard such as ISO, CyberEssentials to test the "security monitoring".

Applicable risk classes: III-V

Should the scope of such an assessment include the AWS services used by customer systems, the associated burden may be considerably reduced by referring to the controls detailed in the relevant AWS compliance documentation, such as SOC reports, ISO27001, 27017, and 27018 certifications, among others.

Maintaining compliance

Having achieved compliance with the expected controls for a given risk classification, it is essential to maintain compliance, since changes to workload requirements, operating system patches, changes to configuration, and so on, all give rise to configuration drift away from the ideal established at the implementation stage. Securing a system deployed to the cloud is not a one-off activity; it is an ongoing process. This section describes the approach and the available AWS services to help with this process.

Secure practices

Before considering the tools available to implement secure practices, it is essential to ensure that good security methodology and processes exist and are adhered to within your organization. There is useful advice on what to establish in this regard within the following resources:

- The Cloud Adoption Framework's Security Perspective;
- The AWS Well-Architected Framework's Security Pillar; and
- Other complementary AWS security best practices.

AWS CloudFormation drift detection

AWS resources deployed and managed through the CloudFormation service (described in Section 5.1: Configuration and change management (p. 9) in this document) can benefit from a feature called *drift detection*, which discerns configuration changes to AWS resources outside this service's agency upon request. For more information, see Detecting unmanaged configuration changes to stacks and resources.

AWS Config

Included within the Landing Zone solution, this service tracks configuration settings of AWS resources over time against a desired-state baseline, and raises alerts (and optionally triggers remedial action) when changes are detected.

The service also enables configuration to be audited, in order to demonstrate compliance (or otherwise) against a baseline. See the AWS Config Developer Guide for a detailed description of how to use it.

AWS Systems Manager

To minimise the possibility and impact of unauthorised or erroneous configuration changes being made to the operating systems of Amazon EC2 instances and the applications on them, as well as other AWS resources, use the AWS Systems Manager suite of services.

AWS Security Hub

This service brings together the variety of security tools available within the AWS environment, providing automated compliance checks and aggregated security findings from disparate security tools in a standardised format. For more information, see AWS Security Hub.

Third-party tools

A variety of AWS Partners offer tools that complement the capabilities of AWS Config and provide configuration management for the resources in the cloud, such as software running on Amazon EC2 instances. See the Infrastructure Configuration Management Solution Brief for more information.

Conclusion

This whitepaper has explained the background to securing systems deployed to the AWS Cloud effectively in the context of Healthcare in the UK. It has described the detailed considerations to achieving compliance with the Good Practice Guide for cloud security, beginning with workload risk classification, and going on to show how the Good Practice Guide's Principles apply specifically to many of the AWS services likely to be used by organisations needing to adhere to the Good Practice Guide. To constrain the document to a digestible size, the step-by-step instructions have been omitted. The reader can find step-by-step instructions through the links provided in each section. This paper concludes with concrete suggestions on how to address what can be the most challenging aspect of compliance: maintaining it in a constantly changing environment.

Contributors

The following individuals contributed to this document:

- Angus McAllister, Solutions Architecture Manager for Healthcare, UK Public Sector
- Simon Russell, Solutions Architect for Healthcare, UK Public Sector
- Enrico Massi, Specialist Solutions Architect, Security and Compliance

Further reading

For additional information, see:

- National Cyber Security Centre's 14 Cloud Security Principles
- Health and Social Care Cloud Security – Good Practice Guide
- Using AWS in the Context of NCSC UK's Cloud Security Principles
- AWS Documentation
- AWS Architecture Centre

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Whitepaper updated (p. 28)	Updated for technical accuracy.	September 28, 2021
Initial publication (p. 28)	Whitepaper first published.	July 1, 2019

Note
To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser that you are using.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.