Cyber Security Architecture & Engineering:

Cloud Services Best Practices

NBCUniversal
Operations & Technology
CYBER SECURITY

Table of Contents

NBCUniversal
Operations & Technology
CYBER SECURITY

Objective

Cloud Services Best Practices

The following cloud security best practices were created to utilize cloud services in the most secure way possible. This document is aimed at the following cloud providers:

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

This document was made to be a resource for users using the above cloud providers in the most secure way possible. This is **not** a policy document, but rather a consolidation of recommendations from cloud service vendors when using their various services.

NBCUniversal's Cloud Security Principle

Cloud Services Best Practices

Zero Trust

The Zero Trust Approach was established for businesses to understand the concept of **"never trust, always verify"**. With the increase of compromises to large corporations in recent years, there's a large assumption that once businesses enter contracts with public cloud vendors or other cloud providers, they no longer feel responsible for protecting the data that's stored. The Zero-Trust Approach establishes that businesses are thinking about their security model when protecting their data.

Some actions that businesses can take to uphold this approach are:

- Not have implicit connectivity between systems in any zone
- **Endpoint:**
  - Layer network and endpoint protections together; encrypt traffic to non-network endpoints
  - Ingest user and traffic data from firewalls into a network security management tool
  - Integrate endpoint protections with a firewall; use MFA for scalability and to minimize exposure from critical applications

- Determine what needs to be protected inside your cloud service
- Understand what types of data is being stored in your cloud service
- Set up security access controls

NBCUniversal
Operations & Technology
CYBER SECURITY

- Consider vendor-neutral encryption models
    - helps ensure enterprise compliance with corporate and security mandates and gives companies full control over unauthorized access
    - Not using a 3rd party to perform cryptographic operations on data
- Monitor your hybrid cloud environment
- Stay up-to-date with security capabilities of your cloud providers & vendors

When these actions start to take place and boundaries are set, there's better protection of critical data hosted in cloud from unauthorized applications or users, as well as reducing exposure of vulnerable systems. In addition, businesses prevent the movement of malware throughout the network.

Networking Principles

In addition to Zero Trust, network principles include corporate direct connect/ExpressRoute and external (internet) zones. It's recommended that Corp/Cloud Connection (C3) zone may not have direct internet routes, and have Layer 7 firewalls between C3 and on-premise corporate network*. With external zones, it's recommended to have a Palo Alto VM-Series firewall on all Internet gateways as well as IP Peering with C3 zones (maintaining Zero Trust).

*When mentioning firewalls, consider the use of a WAF. A Web Application Firewall (WAF) protect web servers and hosted web applications against attacks in the application layer via HTTPS. WAFs are designed to protect a portion of your network traffic, and provide virtual patching for weaknesses. A WAF is more customizable, and can be tailored to the specific design of a customer's web application.

It's strongly suggested to use Cloud Flare's WAF. Click here to learn more about Cloud Flare, and how to configure it.

Evident.io

Evident.io is NBCUniversal's chosen cloud security management tool that automates and optimizes security vulnerability and risk management for AWS & Azure cloud environments. The objective of this tool is to increase visibility into cloud environments to provide continuous global view of cloud security risk.

Currently, public cloud accounts across NBCU are developed and configured with minimal security oversight and controls; increasing the risk of accidental or malicious changes which can lead to cyber-attacks that can compromise critical company data. Having Evident.io configured will enforce policy compliance monitoring, management, and alerting—with consistent enforcement of security policy to reduce the risk of accidental or malicious

changes and attacks. The following are some of the benefits of having Evident.io in place:

- Increase security monitoring and detection; have continuous monitoring of accounts that detect policy and compliance violations as they happen while correlating configuration changes with CloudTrail log events to determine who, when, and how risks were introduced to an environment

- Have complete visibility of the organizational structure that provides rollup visibility to cyber team, and allow business teams to have complete access to all their accounts in one place for ease when onboarding new accounts

- Automatically receive real-time alerts when changes are made, or set custom alert intervals with integration with SIEM and cyber orchestration platform

- Have zero impact deployment by having read-only access so operational risks are not introduced (funded by cyber security to drive adoption)

- Uses read-only and STS authenticated API calls; Security Token Service (STS) is a service that authenticates users by validating credentials and issuing security tokens across different formats

- Checks for unused permissions and IAM policies; see **IAM section** for best practices regarding this

To begin integrating Evident.io into everyday business operations, work with your business area information security officer to onboard any existing cloud accounts into Evident.io using step-by-step instructions. Also, use the ESP Dashboard to determine what risks should be remediated as well as set up accounts to send alerts—via email or external system—when new changes are made that don't follow security policies. When changes are made, determine when remediation actions should be escalated to your BA ISO or Cyber Security. Continue to onboard any new cloud accounts when they are created, as well generate customized reports when they are needed.

Identity & Access Management

Cloud Services Best Practices

The following best practices are recommended to ensure that users have the appropriate levels of permission to access the resources.

IAM Users

Ensure users are required to create strong passwords and passwords are rotated periodically. You can use the password policy to define password requirements, such as minimum length, whether it requires special characters, and how frequently it must be rotated.

Create groups that relate to job functions instead of defining permissions for individual IAM users. Assign IAM users to those groups.

Grant only the minimum permissions required to perform necessary tasks, and only those tasks. For example, you should carefully consider your use case before granting read access to everyone because this allows anyone to access the data. You should never grant write access to everyone because this allows anyone to add and delete objects.

For privileged users (users who are allowed access to sensitive resources or API operations), enable multi-factor authentication (MFA). MFA should always be enabled on root. With MFA, users have a device that generates a unique authentication code (a one-time password, or OTP). To authenticate, users must enter both their normal credentials (such as username and password) and their OTP.

Remove IAM user credentials that are not needed. Periodically check for user credentials that are not used based on last sign-in.

IAM Accounts

Never create account root user access keys. Instead, create an IAM user for yourself that has administrative privileges.

Never share passwords or access keys with anyone. Also, don't share accounts and passwords between administrators. Create separate accounts for each individual service (one account for production, one for development, one for testing, etc.).

Use logging features to determine actions users have taken in your account and resources used. See the **logging section** below for best practices regarding this.

IAM Roles

When needing to delegate access within or between accounts or to services, use IAM roles.

A role is a set of permissions that grant access to actions and resources. An IAM role is similar to a user in that it is an identity with permission policies that determine what an identity can and cannot do. However, a role does not have any credentials (password or access keys) associated with it.

An IAM user can assume a role to temporarily take on different permissions for a specific task.

Roles also can be assumed by services to perform actions in your account on your behalf. When you setup most service environments, you must define a role for the service to assume. This service role must include all the permissions required for the service to access the resources it needs. Service roles provide access only within your account and cannot be used to grant access to services in other accounts.

In AWS, use trust policies to define who can assume a role.

Note that roles in the sense of IAM roles that can be assumed don't exist in Azure. Instead, Azure has the concept of service principals, which is an identity assigned to applications that will be used to assume and gain access to Azure resources. See below for more detailed comparisons of IAM between the cloud providers.

Resource-based Access (AWS)

When creating a permissions policy to restrict access to a resource in AWS, you can choose an identity-based policy (attached to user, group, or role) or a resource-based policy.

Resource-based policies are attached to a resource as opposed to an IAM user, group, or role. With resource-based policies, you can specify who has access to the resource and what actions they can perform on it. Resource-based policies are only available for certain services and can provide advantages in ease of use depending on the service.

Whether you use identity-based or resourced-based policies, give access only to entities you trust and to give only the minimum amount of access necessary.

Use Policy Conditions for Extra Security

Define additional conditions under which IAM policies allow access to a resource. Some examples include:

- Specify range of allowable IP addresses that a request must come from
- Ensure requests are only within specified date range and time
- Require use of SSL or MFA

AWS, Microsoft Azure, and Google Cloud all provide IAM tools. However, the tools for each provider vary. For Amazon Web Services (AWS), there are 3 main frameworks related to IAM:

- AWS IAM Roles and Policies
  - Create and configure users and groups
- AWS Organizations
  - Configure multiple AWS accounts with same identity and access control policies
- AWS Directory Service
  - For using native Active Directory tools
  - An implementation of Microsoft's Active Directory
  - Useful when you are using Active Directory to manage other parts of infrastructure

For Microsoft Azure, the following are related to their IAM:

- Active Directory provides default foundation for identity and permissions management in Azure
- Offers web-based interface for Active Directory, where you can configure permissions for users and groups

For Google Cloud Platform, the following are related to their IAM:

- Offers an Active Directory connection called Google Cloud Directory Sync (GCDS)
  - Performs one-way synchronization
- Offers two different types of IAM accounts:
  - Google accounts (people)
  - Service accounts (applications)
- Google Group
  - A named collection of Google and service accounts
- Organization
  - The root node of GCP resource hierarchy
- G Suite Domain
  - Virtual group of all the Google accounts that have been created in an organization's G Suite account
- Cloud Identity Domain
  - Similar to G Suite Domain in that it's a virtual group of all Google accounts in an organization, but users don't have access to G Suite
  - Needs GCDS in order to work properly (or else would need to recreate Active directory users and groups manually in GCP)
  - Can implement SSO & MFA (but not for users of G Suite Domain)

Encryption

Cloud Services Best Practices

Encryption of data is crucial in preventing accidental informational disclosure, ensuring that data integrity is not compromised, and restoring data via backup in the event of a system failure or disaster. Because of this, data can be exposed to risks while at rest and in transit and requires encryption in both states.

Encryption can be done by either the client or the server, and there are significant differences between these two configurations.

NBCU Data Classification

NBCU uses the following data classification:

- Tier 1 Company Public
- Tier 2 Company Internal
- Tier 3 Company Confidential
- Tier 4 Company Restricted

For more details on NBCU classification levels, visit http://www.nbcunow.com/nbcu-cyber-security/data-classification-diagram.

Server-side Encryption vs Client-side Encryption

With server-side encryption, data is not encrypted until it reaches the object storage service. Server-side encryption limits the complexity of the environment while maintaining the isolation of your data, but it still has risks because it relies on trust in the server to keep the data private. If the server memory contents are exposed, the data becomes vulnerable.

With client-side encryption, users encrypt their own data with their own keys. Users' data and keys are never revealed to the server—only encrypted data is revealed. Data is encrypted before it leaves devices/networks. Client-side encryption is ideal for sensitive data, but it is important to consider the needs of your specific applications and decide what data is sensitive enough to implement client-side encryption for. Client-side encryption limits the risk of outside access to your information.

AWS, Microsoft Azure, and Google Cloud all offer both server-side encryption and client-side encryption.

Use server-side encryption for NBCU data classification levels Tier 2 to Tier 4.

Encryption at Rest

Encryption at rest provides data protection for stored data. The purpose of encryption at rest is to prevent attackers from accessing unencrypted

data by ensuring data is encrypted at disk-level. If an attacker were to obtain a hard drive with encrypted data, the attacker would not readily compromise the data.

At rest, models use a key hierarchy made up of the following keys:

- Data Encryption Key (DEK)—a symmetric key used to encrypt a partition or block of data
- Key Encryption Key (KEK, sometimes KSK)—an asymmetric key used to encrypt DEKs

This is known as envelope encryption. Envelope encryption is the practice of encrypting plaintext data with a data key, and then encrypting the data key under another key.

Encryption in Transit

Encrypting data at transit involves data actively moving from one location to another. TLS is required to protect data when traveling between cloud services and end users.

When data is traversing the public network, it should be protected from disclosure through encryption. Along with limiting the data, applications and administrative access to public cloud services should be managed. To manage, the following is recommended:

- Use HTTPS with server certificate authentication
- Offload HTTPS processing on load balancing to minimize impact on web servers

AWS, Microsoft Azure, and Google Cloud all offer encryption in transit using TLS, and their implementations do not differ greatly.

Encryption in AWS vs. Microsoft Azure vs. Google Cloud

The following sections will focus on encryption for Amazon S3, Azure Blob Storage, and Google Cloud Storage, but it is important to enable encryption for whichever specific cloud service is being used.

AWS has the most options regarding encryption services and key management offerings. Users can enable server-side encryption (SSE) by default for all objects uploaded to S3. For both server-side and client-side encryption, AWS utilizes AES-256 encryption algorithm.

For SSE, Amazon S3 supports:

- Amazon S3-managed keys
  - Both KEKs and DEKs stored and managed by S3 service
  - Rotates master key automatically and does not change the ARN or alias of key, only cryptographic material
- AWS Key Management Service (KMS) managed keys
  - Data Key is encrypted by CMK (customer master keys)
  - Allows both manual and automatic CMK rotation

        - Automatic key rotation does not change ARN or alias of key, only cryptographic material
        - Manual key rotation changes ARN and alias of the key
    - Customer-provided keys
        - Burden of key management completely placed on user
        - Amazon S3 handles encryption/decryption process, but customer provides encryption keys (must be AES-256 symmetric key)
        - Doesn't store the keys
            - Stores a salted HMAC of the keys for authentication
        - Will reject any request over HTTP (must be HTTPS)
            - Key should be invalidated and rotated if used over HTTP

For client-side encryption, Amazon S3 supports:

- Using an AWS KMS-managed customer master key
- Using a client-side master key
- Note that client-side encryption may not be compatible with other services such as AWS Macie and network optimizations or firewalls that inspect traffic
    - SSE-S3 and SSE-KMS work better in these scenarios

From a key management perspective, AWS offers:

- CloudHSM for customer with their own key management software
    - Provides hardware security modules in the cloud
        - A hardware security module (HSM) is a computing device that processes cryptographic operations and provides secure storage for cryptographic keys
        - Provides ability to set cryptographic lifecycle
- Amazon S3-managed keys
- AWS KMS

For more detailed information on AWS Security Best Practices, visit https://nbcuni.sharepoint.com/sites/CybersecurityEngineeringArchitecture/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2Fsites%2FCybersecurityEngineeringArchitecture%2FShared%20Documents%2FOperating%20Standards%2FCloud%20Standards&FolderCTID=0x012000

For **Microsoft Azure**, users can enable SSE by default for all objects uploaded to Azure Blob Storage. In Azure, SSE is called Storage Service Encryption when it pertains to blob storage. Azure, like AWS, also utilizes AES-256 algorithm.

Storage Service Encryption supports using a KEK that uses either:

- Managed by the storage service itself, using Microsoft's internal key management infrastructure
- Customer managed and stored in Microsoft Key Vault, the Azure key management service

For client-side encryption, Azure supplies a storage client library. With this option, users have following options:

- Storing and managing their own KEKs
- Using Azure Key Vault.

From a key management perspective, Azure offers the following 2 options for SSE:

- All keys generated and stored by Azure Blog Storage service itself
  - Microsoft handles key storage and management with no customer involvement
- CEKS are generated and stored within Azure Key Vault
  - KEKs are stored within Azure Key Vault but managed by customer

From a key management perspective, Azure offers the following options for client-side encryption:

- CEKs are generated by Azure storage client library
  - KEKs are stored within Azure Key Vault but managed by customer
- CEKs are generated by Azure storage client library
  - KEKs are generated, stored, and managed by customer using own key management infrastructure
- Both CEK and KEK are generated, stored, and managed by customer using own cryptographic system
  - Azure Blob Storage is unaware that it is storing encrypted data

Google Cloud Storage performs SSE by default on all uploaded objects.

Google Cloud Storage supports SSE with 2 options:

- Keys generated and stored in Google's KMS
- Customer-supplied encryption key
  - User generates own AES-256 encryption key
  - Like AWS, GCP also provides CloudHSM

Google Cloud allows client-side encryption but does not currently offer specific integrations. User is responsible for generating encryption keys and encrypting the data prior to upload. Google Cloud Storage Encryption by Default option leverages Google's internal KMS.

Cloud storage allows wide storage and retrieval of any amount of data at any time. Storage allows for range of usage: storing data for disaster recovery, distributing data into objects, direct download/upload, etc.

Several best practices for storage, including **IAM**, **encryption**, **logging/monitoring**, etc. are covered in detail in previous and later sections. The following are additional best practices to allow the data to be stored in the most secure way possible.

Public vs. Private vs. Hybrid Storage

There are 3 types of cloud storage models: public, private, and hybrid. Public storage is owned and operated by a third-party cloud provider (such as AWS, Azure, or Google Cloud) and delivered over the internet. Private storage is housed on-premises. Hybrid storage has characteristics of both public and private storage.

Public storage is the most economical solution while private storage allows full control over security. Hybrid storage allows flexibility in designing access to data and maintaining control of sensitive assets.

Storage in AWS vs. Microsoft Azure vs. Google Cloud

There are many cloud storage options in AWS, Microsoft Azure, and Google Cloud.

Block storage is persistent disk storage used in conjunction with cloud-based VMs. Each provider splits their block storage offerings into two categories:

- Traditional magnetic spinning hard-drive disks
- Solid state disks
    - More expensive but better performance

AWS's block storage product is Elastic Block Store, Azure's block storage product is Managed Disks, and Google's block storage product is Persistent Disks. AWS and Google offer a 99.95% availability SLA, and Azure offers a 99.99% availability SLA. Google Cloud also offers the highest IOPs (disk speed) and gives customers the most choice in the size of block storage volumes.

Consider block storage if the files being stored are over 1TB, or when data must be accessible quickly and requires long-term persistence. By doing this, the upload time is significantly decreased. Store data in backup so information is not lost if compromised happens. For a more in-depth understanding of block storage in AWS S3 Storage, please refer to Cybersecurity Senior Engineer Steve Nicholls' document found here.

Object storage is used to store bundled data, such as files, with corresponding metadata. Cloud providers classify object storage based on how often the customer expects access to it. "Hot" storage is data that needs to be instantly accessible, and "cold" storage is data that rarely needs to be accessed. Cold storage is less expensive.

Amazon's object storage product is S3, Azure's is Azure Storage (Blobs), and Google's is Google Cloud Storage.

File Storage is data stored as a single piece of information inside a folder. AWS's file storage product is Elastic File System and Azure's is Azure File Storage. Google does not have native file storage but offers FUSE adapter, which allows users to mount files from Google Cloud Storage buckets and convert them into a file system. Elastic File System has no size limit for files, and Azure File Storage has a limit of 5TB per file and 500TB per account.

Storage Best Practices

Cloud providers follow a shared responsibility model for security. This means that they provide secure infrastructure and cloud services, but their customers are responsible for secure operating systems, platforms, and data.

Be stringent about who has read and write access to data. Follow IAM best practices listed above. If your use case requires more granular control, you can create your own Access Control Lists (ACLs) with custom permissions and/or enable MFA Delete, which requires users to authenticate with MFA before deleting data.

Storage security begins well before the storage level. Starting at the network level, you can build a VPC topology that isolates parts of the infrastructure through the use of subnets, security groups, and routing controls. Note that a caveat to routing is that a route that is more specific than the VPC CIDR block cannot be added to the VPC's route tables. For example, if VPC CIDR is 10.0.0.0/16 then route tables cannot contain the route 10.0.0.0/24. In this scenario, use security groups or NACLs to allow or deny.

You can also build a web application firewall to protect from SQL injection and other vulnerabilities in your application. An exhaustive list of all network security features is beyond the scope of this document but enabling available security features may be instrumental in building defense in depth.

Cloud storage requests refer to buckets and objects by their names. The following are a few best practices when it comes to naming:

- Choose bucket and object names that are difficult to guess
- Avoid the use of sensitive information as part of bucket or object names. This includes:
  - User IDs
  - Email addresses
  - Project names
  - Project numbers
  - Any PII Data

Enable IAM storage policies such as AWS S3's bucket policy.

Implement encryption for file storage and block storage. See **encryption** section above for more details.

Periodically version/create snapshots to store previous storage versions as backups.

Enable security services such as Amazon Macie.

Enable a logging solution to track what actions have been performed against your data.

Enable services that allow you to trigger alerts for specific events. See **logging** section below for more details.

Databases

Cloud Services Best Practices

A database is a collection of information that is organized with a well-defined data format in an efficient manner for creating, retrieving, updating, and deleting data.

Relational vs. Non-Relational Databases

There are two main types of databases: relational and non-relational. A relational database organizes structured data into defined columns whereas a non-relational database stores data in a single document file.

Databases in AWS vs. Microsoft Azure vs. Google Cloud

AWS, Microsoft Azure, and Google Cloud all provide a variety of database services and support relational databases as well as NoSQL databases.

	AWS	Microsoft Azure	Google Cloud
Types of storage (Relational)	Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server	SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL	Google Cloud SQL for MySQL, Google Cloud SQL for PostgreSQL
NoSQL databases provided	DynamoDB, SimpleDB	DocumentDB	Cloud Datastore, Bigtable
Additional database-related services	Database Migration Service, DynamoDB Streams, Redshift, Elasticache	Table Storage, SQL Data Warehouse, Azure Redis Cache	Big Query, Memcache

Database Best Practices

The following are some database security best practices for the cloud:

Use access and role management to restrict access to the database. Enable dynamic data masking, making data selectively available based on a user's authorization level.

Have firewall rules in place to prevent any database access except through rules specified by an associated security group.

Uninstall and/or disable any features or services that do not need to be used that may have access to the database (follow **IAM** best practices for more details).

View stored sensitive data as “toxic waste.” Periodically check for storage of sensitive data that does not need to be retained.

To prevent overloads, performance constraints, and capacity issues, scale database instance up when approaching storage capacity limits. Have buffer in storage and memory to accommodate increases in demand from applications.

To protect database at an application-level, deploy a web application firewall to prevent common attacks such as SQL injections. Use secure coding best practices such as prepared statements.

Use recommended best practice encryption algorithms to protect sensitive data (Follow **encryption** best practices below for more details).

Utilize logging services that track what database actions have been performed. Enable services that allow you to trigger alerts for specific events (follow **logging** best practices for more details).

Regularly test and monitor backup system to ensure reliable disaster recovery. Keep an encrypted copy of your database on backup. This allows you to recover any deleted or changed file in the case something goes wrong.

Compute (VM)

Cloud Services Best Practices

Virtual machines (or instances) are the safest way to run an alternative operating system such as Linux or Windows. When using instances, best practice would be to define the minimum set of privileges each server needs to perform its function. Restrict server access from both the network and on the instance; creating processes to control changes to server configuration baselines.

IaaS vs. SaaS vs. PaaS

When thinking of responsibilities of a service provider in relation to that of its users, there are three service options:

-**Infrastructure as a Service (IaaS)** gives users automated and scalable hardware environments with flexibility and control. With IaaS, users are given an environment to host their applications on virtual servers while the underlying hardware is outsourced to the given cloud provider.

-**Software as a Service (SaaS)** allows applications to be hosted in the cloud, while being accessible through the internet. With SaaS, users aren't restricted to one location when accessing applications; via the internet, applications are accessible for various devices, and are able to scale their storage and service without need to install new software or hardware.

-**Platform as a Service (PaaS)** provides a framework for quickly developing and deploying applications through automating infrastructure and management. With PaaS, developers can collaborate on applications in various locations; using the internet to develop an application simultaneously.

IaaS helps build the infrastructure, PaaS helps developers build custom applications, while SaaS are the applications that are utilizing these environments. All three services are needed to host, build, deploy, and maintain applications in the cloud.

Below are the transcriptions from CSA's Security Guidance for Critical Areas of Focus in Cloud Computing V3.0:

"**Infrastructure as a Service (IaaS)** includes the entire infrastructure resource stack form the facilities to the hardware platforms that reside in them. It incorporates the capability to abstract resources (or not), as well as deliver physical and logical connectivity to those resources. Ultimately, IaaS provides a set of logical Application Programming Interfaces (API's), which allows management and other forms of interaction with the infrastructure by consumers.

Platform as a Service (PaaS) facilitate deployment of applications without the cost and complexity of buying and managing the

underlying hardware and software and provisioning hosting capabilities. This proves all the facilities require to support the complete life cycle of building and delivering web applications and services entirely available from the Internet. PaaS sits on top of IaaS and adds and additional layer of integration with application development frameworks, middleware capabilities, and functions such as database, messaging, and queueing. These services allow developers to build applications on the platform with programming language and tools that are supported by the stack.

Software as a Service (SaaS) in turn is built upon the underlying IaaS and PaaS stacks and provides a self-contained operating environment that is used to deliver the entire user experience, include the content, its presentation, the application(s), and management capabilities. Generally, SaaS provides the most integrated functionality built directly into the offering, with least consumer extensibility, and a relatively high level of integrated security (or the providers bears responsibility for security, at the very least).

The Key takeaway for security architecture is that the lower down the stack the cloud service provider stops, the more security capabilities and management consumer are responsible for implanting and managing themselves."

Manage Operating Systems in Instances

OS-level security restricts access to the operating system. In general, disabling the root API access key and secret keys, rotating all keys regularly, enabling MFA for all users (root required at minimum), keeping VMs up-to-date, and deploying a firewall for internet inbound and outbound traffic are all highly advised. Additional OS-level security can be applied through deploying best practices for authentication, least privilege, and auditing.

- **Authentication:** password protect .pem files and use a bastion host with agent forwarding to connect to instances – do not store keys on the instance. Credentials should be rotated regularly and keys should be deleted from the authorized key file when a user no longer requires access. A centralized directory service, such as AWS Managed Directory, can be used to manage Windows and Linux instances with group policy enforcement and SSO. Access can be further restricted by limiting the accepted IP range and protocol – do not allow open access to instances (0.0.0.0/0 for all traffic), especially if directly connected to the internet.
- **Least privilege:** ensure users, roles, and groups adhere to the best practice of least privilege. Avoid creating policies with wildcards. For example, to allow an AWS resource to get an item from a S3 bucket use "Action": "s3:GetObject" instead of "Action": "s3:*". **Never** use wildcards for the principal as this will allow anyone to assume the policy.
- **Auditing:** log access to instances, resource performance, and IAM with services such as Evident.io. Automatic

alerts should be configured on violation of security policies and include remediation steps.

Temporary Storage

Virtual machines tend to contain one temporary disk on each VM. The data on these disks don't remain through the VM's lifecycle events. Instead, the data is stored on the host operating system running the software while the data for persistent disks are in storage. It's recommended not to use the temporary disk for data that should be persistent; persistent meaning data that is written to disks that will be available through reboots, start/stop, and other lifecycle events. Persistent disks are located independently from VM instances, so they can move to keep data even after the instance is deleted. In this case, best practice is to ensure the temporary disk is not being used incorrectly. To ensure this, take actions to cause the temporary disk to be reset as a part of testing procedures; with the easier method being to change the size of the virtual machine. Consider the following:

- Configure the virtual machine as planned
- Change the VM size, and then return to the VM to ensure everything is working properly

VM in AWS vs. Microsoft Azure vs. Google Cloud

The following section takes a deeper dive into virtual machines for Amazon Elastic Compute Cloud, Azure and Google Cloud's virtual machine practices.

AWS

Amazon Elastic Compute Cloud (EC2) is AWS's service that allows users to create and run instances in the cloud with either a Windows or Linux operating system. Some best practices that Amazon EC2 provides in relation to instances include:

- Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles (please see the I&AM section for recommended practices for securing roles and users)
- Regularly patch, update, and secure the operation system and applications on the used instance; for more information on managing software on Linux instance, click here
- Launch instances into a VPC (virtual private cloud); some benefits to this include:
  -Able to define the network your instance will sit in; select its IP address range, create subnets, configure route table, network gateways, and security settings

-Able to connect the instance in the VPC to the internet
-Control the outbound traffic from the instance; control the inbound traffic to the instance
-Add a layer of access controls to the instance in the form of network ACLs
-For more benefits and their explanation, refer to AWS's document here

- Use instance metadata and custom resource tags to track and identify AWS resources
- Regularly backup EBS volumes, and create an Amazon Machine Image from the instance to save the configuration as a template for future instances
- Deploy critical components of applications across multiple Availability Zones; replicating data when appropriate.

AWS Config enables a user to assess, audit, and evaluate the configurations of their AWS resources. AWS Config monitors and records AWS resource configurations. With AWS Config users can record changes to patching and compliance statuses, and maintain a history of all changes to this data over time-being able to use it for auditing and compliance needs.

Some best practices for AWS Config include:

- Enable AWS Config in all accounts across regions to ensure that ensure configurations comply with best practices when they're audited
- Record configuration changes to ALL resource types to ensure that users have comprehensive configuration audits in place
- Ensure that users have a secure Amazon S3 buckets to collect the configuration history and snapshots
- Specify an Amazon S3 bucket from another account for centralized management of history files and snapshots

To know more best practices, click here.

AWS Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. AWS Inspector assesses application for vulnerabilities and produces a list of security findings prioritized by level of security. AWS Inspector consists of an agent installed in the OS of a user's EC2 instance, and uses telemetry from the agent and AWS configuration to assess instances for security exposures and vulnerabilities.

Google Cloud

When creating virtual machines on Google Cloud Platform (GCP), each instance belongs to a GCP console project. When creating the instance, the user can specify the zone, operating system, and

machine type of that instance. By default, each instance has a small root persistent disk that contains the operating system. When an instance is deleted, it's deleted from the project. Click here for a more detailed explanation when creating an instance in GCP.

When managing Linux instances in Google Cloud, using OS Login allows users to associate SSH keys to their Google (or G Suite) account, and manage admin or non-admin access to instance through IAM roles. Another option is to manage SSH keys in project or instance metadata; granting admin access to instances with metadata access that don't use OS Login.

When managing Windows instance in Google Cloud, users can create a password for a windows server instance. Google Cloud utilizes Compute Engine, that requires a user to generate a new Windows password for that instance before a user can connect to it. For documentation on how to create those passwords, click here.

Azure

Azure virtual machines can be used in the following examples:

- Development and testing; create a computer with specific configurations required to code and test an application
- Applications in the cloud
- Connecting an organization's network in the azure virtual network

According to Azure documentation, there are various design considerations when building out an application infrastructure in Azure. Some of those include:

- The names of application resources
- The location where the resources are stored
- The size of VM
- The maximum number of VMs that can be created
- The operating systems that the VM runs
- The configuration of the VM after it starts
- The related resources that the VM needs

Logging

Cloud Services Best Practices

Each cloud service provides log files to give insight into how the service is operating. When working with these log files, it's best to enable audit and access logging capabilities wherever they are available. Analyze these logs, and then create reports to configure the appropriate alerts to get the most out of the log data.

It's also recommended to use meaningful names to organize log data when consolidating logs into a single location. Restrict the usage of owner roles for projects and log data; grant the least minimum permissions required to do this.

Create standards within the code itself to be enforced; have log information be provided in the contextual parts rather than plain full-text lines.

Also, **do not** log data across accounts- for example, creating logs from instance running in account A and store the logs in S3 bucket in account B. Users might not have access to specific account therefore visibility will be lost.

Centralized Logging

The following are general best practices when implementing a centralized logging management solution:

Define log retention requirements and lifecycle policies early on, and plan to move those log files to cost-efficient storage locations; incorporating tools and features that automate the enforcement of these lifecycle policies. A good starting point for this is to store compressed copies of audit, firewall, and alert logs for at least 60 days. Depending on the need and requirements for your service, will let you know if the retention requirements need to be adjusted. Begin thinking about a cost-efficient storage location that includes built-in lifecycle capabilities that meet your retention requirements (i.e. storage buckets or objects).

Consider additional tasks, costs, dependencies associated with managing and maintaining its components, before implementing a custom-built solution. Consider automating the installation and configuration of log shipping agents to consistently capture system and application logs and support dynamic scaling of instances. A best practice is to choose a solution that integrates with both on-premise and your cloud service's workloads; implementing a log management solution that provides visibility across all operating environments.

Log Archive

Use secure, durable storage for log files to ensure log files are not accidentally lost, stolen, or tampered with. Also create and implement

log lifecycle policies for storing, aggregating, analyzing, archiving, and deleting log data. Use cloud provider's log solution to log data in durable storage; easy to quickly send both rotated and non-rotated log data off a host and into the log service. With this, you're able to access the raw log data when you need it. For AWS, Amazon CloudWatch Logs is used to monitor, store, and access files from Amazon EC2 instances among other AWS services.

In addition, AWS has VPC Flow Logs- a feature that enables a user to capture information about the IP traffic going to and from network interfaces in the user's VPC. Flow logs can be beneficial when troubleshooting (i.e. why specific traffic is not reaching an instance) as well as a security tool to monitor the traffic that is reaching a user's instance.

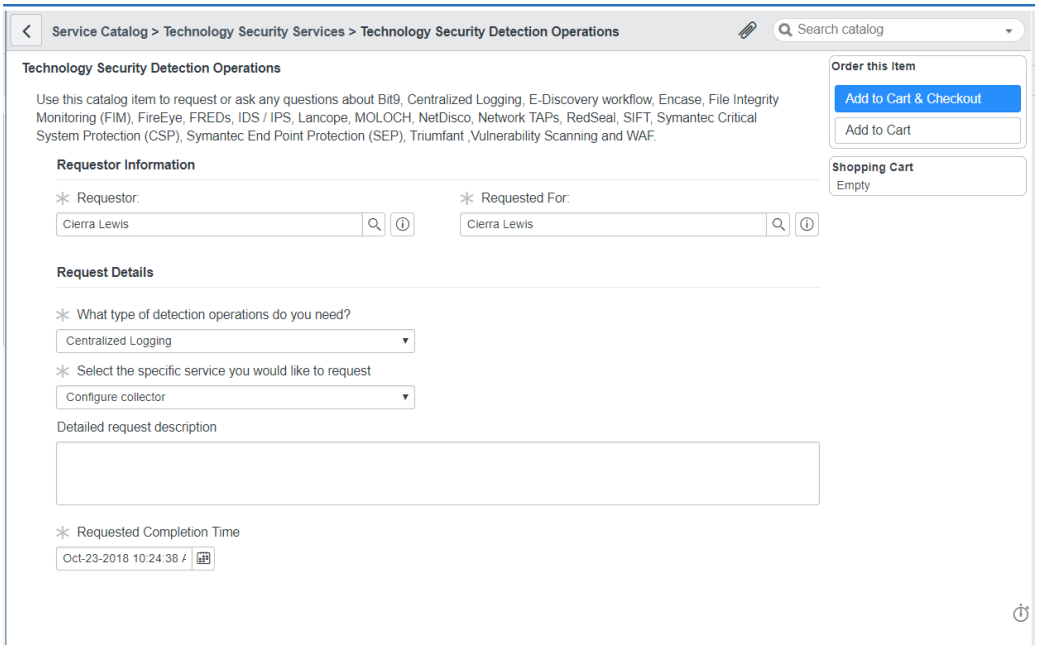
For Azure, while there are various types of logs that collect metadata to maintain an application's performance, Analytics Log is Azure's service that collects and analyzes data that's generated by resources in one's cloud and on-premise environment.

Finally, Google Cloud's tool, Stackdriver Logging, allows a user to store, search, analyze, monitory, and alert on log data from Google Cloud Platform and AWS.

Splunk

While each cloud provider provides their own logging solution, Splunk is NBCUniversal's chosen logging application that captures, indexes, and correlates real-time data in a repository that can be generated into graphs, reports, alerts, and dashboards for visualization into one's applications and services.

The current process to integrate logs into Splunk starts with a ServiceNow request. In the Service Catalog, select the 'Technology Security Detection Operations' option. From there in the 1st drop-down selector Centralized Logging; which will then populate the 2nd drop-down: Configure Collector. In the detail section, specify the application and host information of the appliances, such as IP and hostname, as well as where it's located, Azure of AWS for example.



From there, a meeting would need to be initiated between the Splunk Team and the application owner to discuss the integration method and what will need to be configured on the application side. For more explanation, contact Ilya Beskin, Willis Hy, or the Cyber Security Platforms' team at cyber.platforms@nbcuni.com.

References

Cloud Services Best Practices

Amazon Web Services

- https://aws.amazon.com/answers/logging/centralized-logging/
- https://aws.amazon.com/answers/logging/aws-native-security-logging-capabilities/
- https://d1.awsstatic.com/whitepapers/Security/AWS_Security_Whitepaper.pdf
- https://aws.amazon.com/answers/security/aws-securing-windows-instances/
- https://aws.amazon.com/answers/security/aws-securing-ec2-instances/
- https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
- https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html
- https://docs.aws.amazon.com/aws-technical-content/latest/efs-encrypted-file-systems/introduction.html
- https://aws.amazon.com/blogs/security/how-to-protect-data-at-rest-with-amazon-ec2-instance-store-encryption/
- https://docs.aws.amazon.com/aws-technical-content/latest/efs-encrypted-file-systems/encryption-of-data-in-transit.html
- https://d1.awsstatic.com/whitepapers/Security/Security_Database_Services_Whitepaper.pdf
- https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html
- https://d1.awsstatic.com/whitepapers/Security/AWS_Security_Best_Practices.pdf
- https://aws.amazon.com/premiumsupport/knowledge-center/secure-s3-resources/
- https://docs.aws.amazon.com/aws-technical-content/latest/cost-optimization-right-sizing/cost-optimization-right-sizing.pdf
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id.html
- https://aws.amazon.com/iam/faqs/
- https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access_resource-based-policies.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_controlling.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_compare-resource-policies.html
- https://aws.amazon.com/blogs/security/control-access-to-aws-resources-by-using-the-aws-organization-of-iam-principals/
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-services-that-work-with-iam.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_identity-vs-resource.html

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_terms-and-concepts.html
- https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html
- https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html
- https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html
- https://docs.aws.amazon.com/macie/latest/userguide/macie-integration.html#macie-encrypted-objects
- https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingKMSEncryption.html
- https://aws.amazon.com/blogs/mt/aws-config-best-practices/
- https://docs.aws.amazon.com/cloudhsm/latest/userguide/introduction.html
-

Microsoft Azure

- https://docs.microsoft.com/en-us/azure/security/azure-database-security-best-practices
- https://docs.microsoft.com/en-us/azure/security/azure-security-identity-management-best-practices
- https://cloud.google.com/security/encryption-at-rest/default-encryption/
- https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis
- https://azure.microsoft.com/en-us/product-categories/identity/
- https://docs.microsoft.com/en-us/azure/security/azure-security-encryption-atrest
- https://docs.microsoft.com/en-us/azure/security/security-azure-encryption-overview#encryption-of-data-in-transit
- https://cloudarchitectmusings.com/2018/03/09/data-encryption-in-the-cloud-part-4-aws-azure-and-google-cloud/
- https://azure.microsoft.com/en-us/overview/what-are-private-public-hybrid-clouds/
- https://azure.microsoft.com/en-us/blog/virtual-machines-best-practices-single-vms-temporary-storage-and-uploaded-disks/
- https://docs.microsoft.com/en-us/azure/virtual-machines/windows/overview
- https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals

- https://docs.microsoft.com/en-us/azure/role-based-access-control/

Google Cloud Platform

- https://cloud.google.com/storage/docs/best-practices
- https://cloud.google.com/compute/docs/disks/
- https://cloudplatform.googleblog.com/2018/03/best-practices-for-working-with-Google-Cloud-Audit-Logging.html
- https://cloudplatform.googleblog.com/2018/04/best-practices-for-securing-your-Google-Cloud-databases.html
- https://medium.com/@doctusoft/how-to-make-your-google-cloud-platform-project-more-secure-iam-245dcf05b18f
- https://cloud.google.com/iam/docs/overview
- https://cloud.google.com/security/encryption-at-rest/
- https://cloud.google.com/security/encryption-in-transit/
- https://cloud.google.com/compute/docs/instances/apply-sizing-recommendations-for-instances
- https://cloud.google.com/compute/docs/instances/
- https://cloud.google.com/compute/docs/instances/apply-sizing-recommendations-for-instances
- https://cloud.google.com/iam/docs/understanding-roles
- https://cloud.google.com/iam/docs/resource-hierarchy-access-control
- https://cloud.google.com/iam/docs/overview
- https://support.google.com/a/answer/106368?hl=en
-

Multiple/Miscellaneous

- https://www.cloudberrylab.com/blog/managing-iam-permissions-in-the-cloud-aws-vs-microsoft-azure-vs-google-cloud/
- https://stackify.com/microsoft-azure-vs-amazon-web-services-vs-google-compute-comparison/?utm_referrer=https%3A%2F%2Fwww.google.com%2F
- https://www.symantec.com/connect/blogs/data-stored-clouds-server-side-encryption-enough
- https://www.redhat.com/en/topics/data-storage/file-block-object-storage
- https://www.networkworld.com/article/3191520/cloud-computing/deep-dive-on-aws-vs-azure-vs-google-cloud-storage-options.html
- https://support.cloudflare.com/hc/en-us/articles/115000223771-How-do-I-configure-the-WAF-
-

Zero Trust

- https://www.securityroundtable.org/zero-trust-approach-can-make-cloud-secure/
- **Image:** https://ecs.co.uk/wp-content/uploads/2017/12/Encryption-1024x574.jpg
- https://www.paloaltonetworks.com/cyberpedia/extending-zero-trust-to-the-endpoint

Evident.io

- https://nbcuni-my.sharepoint.com/:p:/r/personal/206566812_tfayd_com/_layouts/15/Doc.aspx?action=edit&sourcedoc={129d1695-8dfe-4bb9-9627-cbbe939e81ef}

SaaS vs IaaS vs PaaS

- https://www.hostingadvice.com/how-to/iaas-vs-paas-vs-saas/
- https://www.interoute.com/what-iaas
- https://www.interoute.com/what-saas
- https://www.interoute.com/what-paas

**Revisions to this document where made on the following date and description of change:*

Date	Description	Change Owner
08/09/18	Go more technical on each section; enlist Charles to help with document	Cierra Lewis
09/10/18	Combine PAM w/ IAM; detail process to add logs into Splunk; move Cloud Security Principle to 1st Section; IaaS vs. SaaS vs. PaaS should be included	Cierra Lewis/Charles Lin
09/25/18	Added more detail to the following sections: IAM, Encryption, Storage, Evident.io, Managed OS in Instances, Logging	Cierra Lewis/Charles Lin
10/24/18	Expanded on document to include use of CloudFlare, VPC Flow Logs, AWS Config & AWS Inspector	Cierra Lewis
11/06/18	Expanded on Data Classifications, updated IAM Section, updated KEKs section,	Charles Lin