Establishing Your Cloud Foundation on AWS

AWS Whitepaper

Establishing Your Cloud Foundation on AWS: AWS Whitepaper

Table of Contents

Establishing Your Cloud Foundation on AWS

Publication date: **May 02, 2022** (*Document revisions* (p. 29))

Abstract

The increasing breadth and depth of cloud services makes the cloud a powerful enabler of efficiency, agility, and rapid innovation. However, building a foundational AWS Cloud environment requires decisions across multiple AWS and partner products, services, and solutions. Customers are looking for guidance to help them set up and operate an environment that is compatible with their IT practices, and enables their builders and operators while adhering to governance requirements.

This whitepaper introduces a guided path approach to help customers build and evolve their AWS Cloud environment based on a consolidated set of definitions, use cases, guidance, and automations. The approach includes people, process, and technology considerations of establishing an AWS Cloud environment.

Introduction

The primary business drivers behind moving to the cloud include greater agility, innovation, and scale. When planning a cloud adoption strategy, the number of decisions that you need to make to stand up a production-ready cloud environment is significant. Decisions that are made early on can affect your ability to enhance and/or scale your environment in the future. This complexity has led customers to look for prescriptive guidance across the range of AWS services that can be used to create a foundational environment.

Establishing a cloud foundation on AWS requires guidance tailored to your business needs. Using a capability-based approach (p. 2), you can create an environment to deploy, operate, and govern your workloads. You can also enhance the capabilities to extend your environment as your requirements evolve and you deploy additional workloads to the cloud.

Building a foundational environment on AWS can be done with a standard, prescriptive set of capabilities across different functional areas (p. 3). These capabilities can be used as a structured way to quickly build or expand your AWS Cloud environment, and include use case scenarios and corresponding guidance.

You can adopt and implement capabilities according to your operational and governance needs. As your business requirements mature, the capability-based approach can be used as a mechanism to verify that your cloud environment is ready to support your workloads and scale as needed. This approach enables you to confidently establish your cloud environment for your builders and your business.

Capabilities

To support cloud adoption, AWS recommends that you have a foundational set of capabilities that enable you to deploy, operate, and govern your workloads.

A *capability* includes a definition, use case scenarios, opinionated guidance, and supporting automation to establish and operate a specific part of a cloud environment. Capabilities are components that can help you plan, implement, and operate your cloud environment, and include *people*, *process*, and *technology* considerations. Capabilities are designed to integrate into your overall technology environment.

In addition to technology implementation guidance, capabilities include operational guidance (for instance, notifications, event handling, and remediation, as well as team resource skills and processes) needed to stand up and operate each capability. For an example of what a capability should offer, refer to Appendix A (p. 30).

AWS has defined a set of 30 capabilities that span six categories to help you establish a cloud foundation.

Table 1 - Cloud Foundations capabilities by categories

Governance, Risk, and Compliance (p. 3	**Security (p. 5)**	**Operations (p. 4**	**Infrastructure (p**	**Finance (p. 8)**	**Business Continuity (p. 7)**
Log Storage	Identity Management & Access Control	Developer Experience & Tools	Network Connectivity	Cloud Financial Management	Backup
Governance	Secrets Management	Rollout/ Rollback	Network Security	Resource Inventory Management	Disaster Recovery
Audit & Assessment	Security Incident Response	Logging & Monitoring	Workload Isolation Boundary	Records Management	Support
Tagging	Encryption & Key Management	Sort/Search for Metadata	Template Management		
Service Onboarding	Vulnerability & Threat Management	Patching			
Change Management	Application Security				
Forensics	Data Isolation				
Data De-identification					

Each capability includes stages of maturity that enable you to implement based on where you are in your cloud journey, including your governance and operational requirements. As your cloud environment grows and matures, the *capabilities* can be enhanced to meet your new requirements.

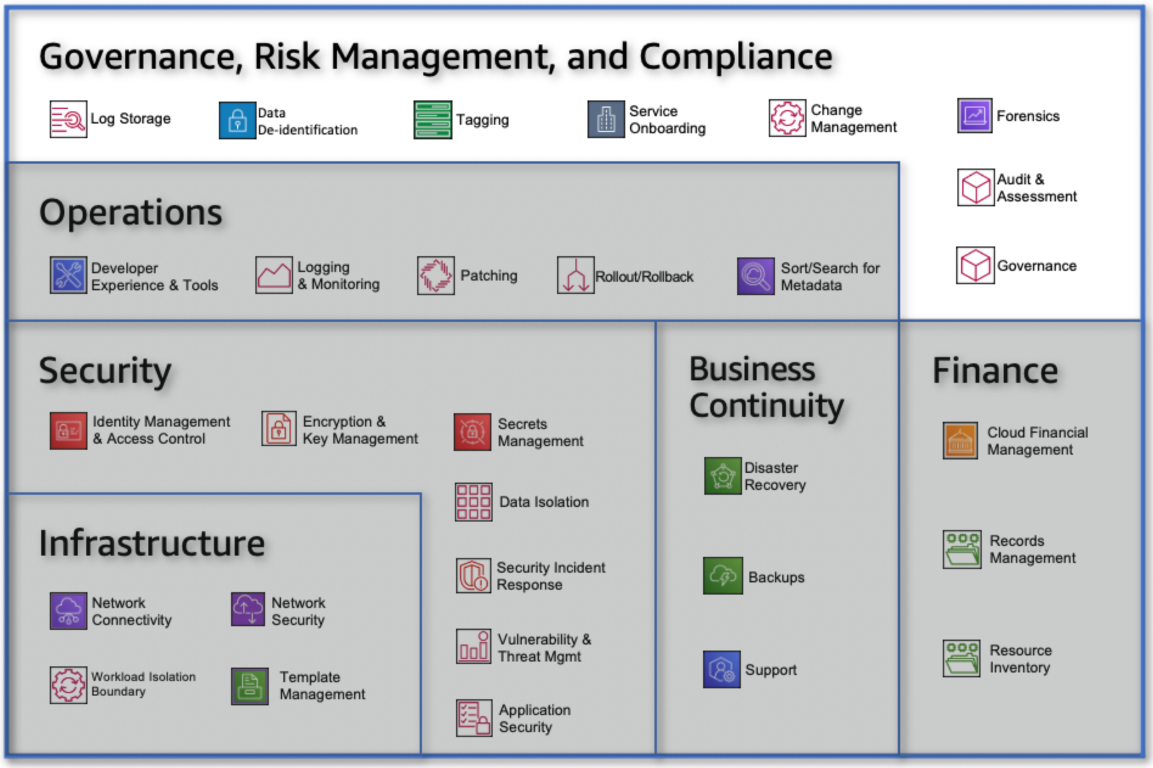
Capabilities definitions

This section includes high-level definitions for each foundational capability organized by their category. For a deeper dive into a specific capability and what it includes, refer to Appendix A (p. 30).

Topics

Governance, Risk Management, and Compliance

Governance, Risk Management, and Compliance (GRC) helps organizations set the foundation for meeting security and compliance requirements and define the overall policies your cloud environment should adhere to. The capabilities within this area help you define what needs to happen, defines your risk appetite, and informs alignment of internal policies.



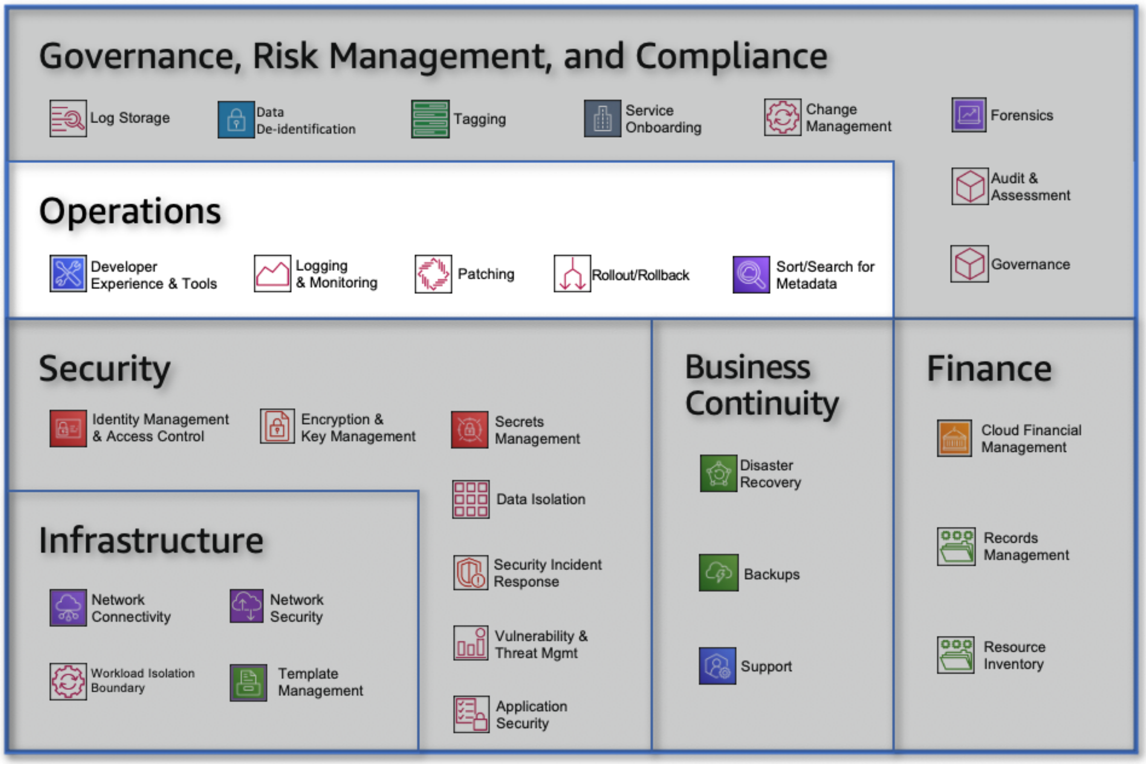
Governance, Risk Management, and Compliance Category

Governance, Risk Management, and Compliance capabilities include:

- **Tagging** enables you to group sets of resources by assigning metadata to cloud resources for a variety of purposes. These purposes include access control (such as ABAC), cost reporting, and automation (such as patching for select tagged instances). Tagging can also be used to create new resource constructs for visibility or control (such as grouping together resources that make up a microservice, application, or workload). Tagging is fundamental to providing enterprise-level visibility and control.
- **Log storage** enables you to securely collect and store environment logs centrally within tamper resistant storage. This capability enables you to later evaluate, monitor, alert, and audit access and actions performed on your AWS resources and events.
- **Forensics** involve the analysis of log data and evidentially-captured images of potentially compromised resources, to determine whether a compromise occurred (and if so, how). Outcomes of root cause analysis resulting from forensic investigations are typically used to produce and motivate the application of preventative measures.
- **Service Onboarding** is the ability to review and approve AWS services for use based on consideration of internal, compliance, and regulatory requirements. This capability includes risk assessment, documentation, implementation patterns, and the change communication aspects of service consumption.
- **Data De-Identification** is the ability to anonymize subsets of data and information as they are stored and processed to reduce their sensitivity (for example, national ID numbers, trade data, healthcare information), and when required, preserving the underlying data format. This capability also includes the ability to tokenize data (such as credit card numbers, physical address, health care records) to reduce the need to access the underlying sensitive data.
- **Governance** is the ability to implement executive board policies that your AWS Cloud environment must adhere to. This policy includes the rules for your environment, defines risks, and informs alignment of internal policies. As your cloud foundation matures, a portion of this capability is embedded in all other capabilities to ensure adherence to governance requirements.
- **Audit & Assessment** is the gathering and organizing of documentary evidence to enable internal or independent assessment of your cloud environment, and activities within it, against standards (including information about who accessed what, when, and from where, and what changes happened). This capability allows you to validate assertions that all changes were performed in accordance with policy and via appropriate workflow mechanisms.
- **Change Management** enables you to deploy planned alterations to all configurable items that are in an environment within the defined scope, such as production and test. An approved change is an action which alters resource configuration that is implemented with a minimized and accepted risk to existing IT infrastructure.

Operations

Enable your developers and operations teams to innovate faster, while ensuring the quality of application and infrastructure updates. The capabilities within this area enable you to build, deploy, and operate, workloads with ease in the cloud with developer experience and tools capabilities.



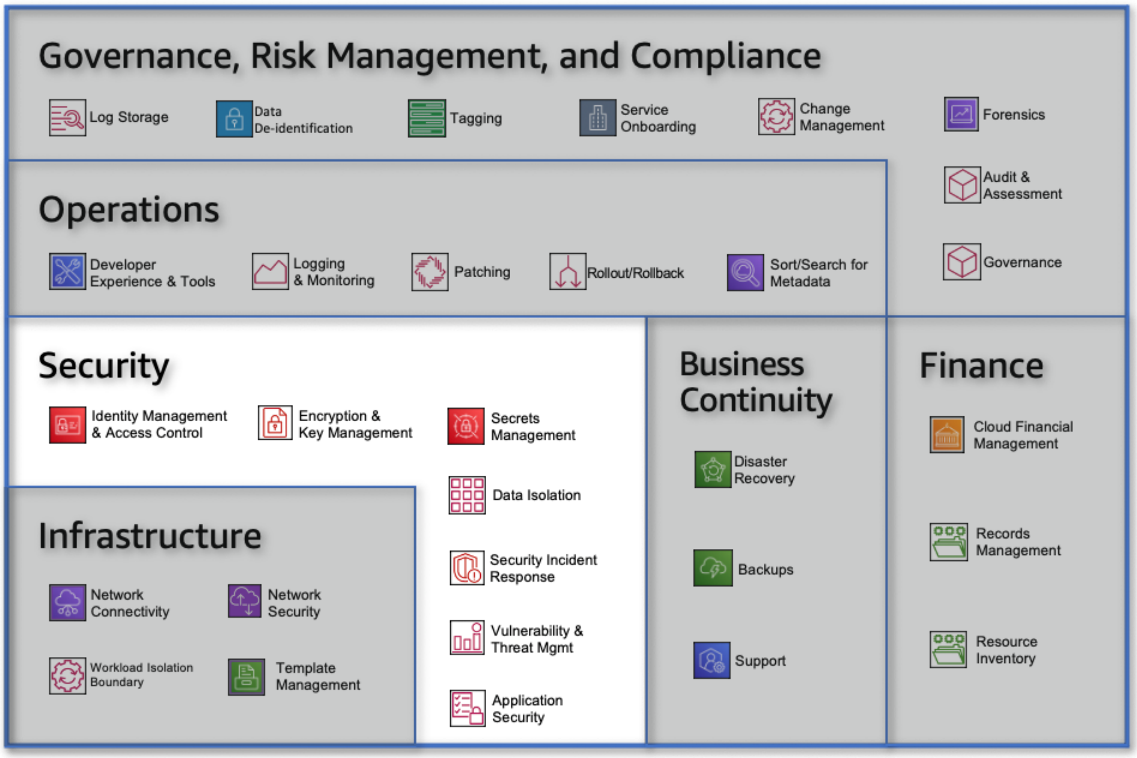
Operations Category

Operations capabilities include:

- **Rollout/Rollback** is the defined strategy to roll out application or configuration changes to the environment, or roll back these changes in case of failure. Application and configuration changes can include updated permissions, new policies, new or updated network configuration, new version of the application, or updated software development kits. These configuration changes can also include modifications to the orchestration framework that deploy these changes.
- **Logging & Monitoring** is the ability to gather and aggregate security and operational data about system and application activities, including near-real-time analysis of data to identify anomalies, indicators of compromise, performance issues, and configuration changes.
- **Metadata Sorting/Searching** is the ability to search and filter based on metadata applied to tagged resources within your environment. These resources can be accounts, or resources within these accounts.
- **Patching** is the ability to deploy sets of changes to update, fix, and/or enhance the operation and security properties of infrastructure and workloads. This includes addressing security vulnerabilities, bug fixes, and other related work. The scope of patching includes operating systems, applications, and any relevant software systems.
- **Developer Experience and Tools** is the ability to provide the tools and processes required for developers to build and deploy workloads easily to the cloud. This capability spans from storing code, to building workflows, to promoting workloads from test to production environment.

Security

Create a secure, high-performing, and resilient foundation for your cloud environment. The capabilities within this area enable you to design and implement security policies and controls across different levels of the stack to protect your resources from external or internal vulnerabilities and threats. They ensure confidentiality, availability, integrity, and usability, while providing priorities and advice to assist with remediation.



Security Category

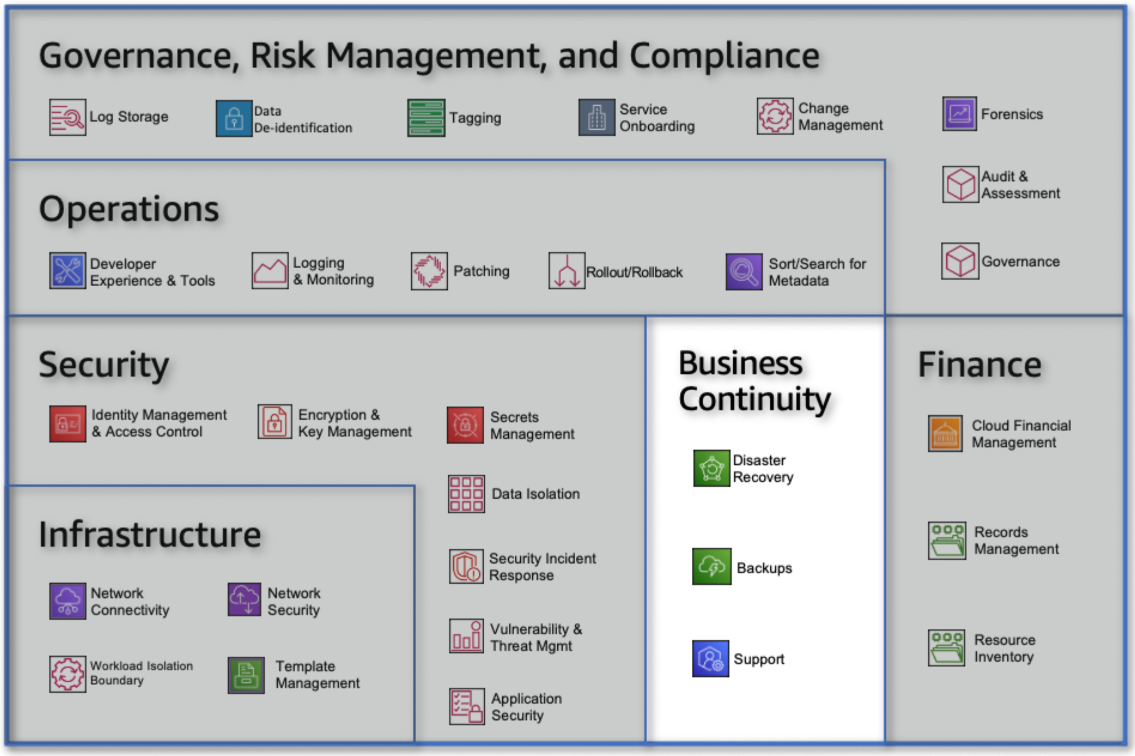
Security capabilities include:

- **Identity Management & Access Control** enables your teams to efficiently build and centrally manage the access to your cloud platform environment. The capability enables you to structure your organization, organize your accounts, and set up access to your environment based on a least-privilege model.
- **Data Isolation** enables you to limit access to data at rest and in transit so that data is only accessible to appropriate, authorized entities. This capability also includes the ability to detect misuse and/or unauthorized access, leak, and theft of data.
- **Application Security** encompasses the protection of application software, and the detection of anomalous behavior in the context of the applications' interactions with clients. Threats to be addressed include unauthorized access, privilege escalation, and other application-level threats typically characterized in threat frameworks.
- **Encryption and Key Management** refers to the ability to centrally manage encryption keys for different workloads, and the ability to encrypt data at rest and in transit. Access to keys is provided based on least privilege, and usage is monitored to report any anomalies. This capability also includes different patterns of rotation based on requirements.
- **Secrets Management** applies to managing *secrets* (access credentials) such as passwords, access keys, other API keys, X.509, or SSH private keys. This capability incudes storage, access control, access logging, revocation, and rotation aspects for managing secrets.
- **Security Incident Response** enables you to respond to a security incident. Based on decisions specified in policy, the response involves characterizing the nature of the incident and making changes (which may involve activities including restoration of operational status, identification and remediation of root cause, and gathering evidence pursuant to civil or criminal prosecution).

- **Vulnerability & Threat Management** is the ability to identify vulnerabilities that can affect the environment (availability, performance, or security). This capability enables you to assess the impact and scope (such as blast radius) of vulnerabilities and threats, and address/remediate them.

Business Continuity

Resilience is critical, it affects the quality of service your users experience. The capabilities within this area enable you to have a strategy in place to continue operations during a time of inefficiency or crisis, including Disaster Recovery, Backups and Support. Having this in place can help avoid downtime during outages or unprecedented situations.



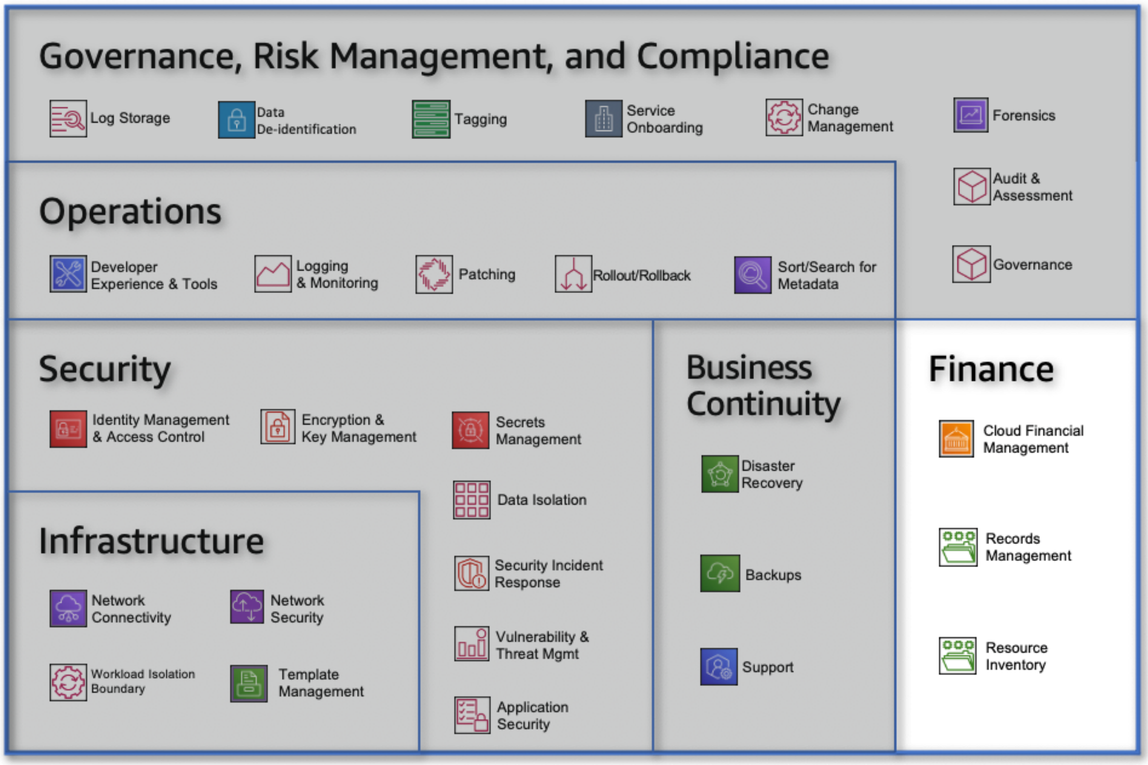
Business Continuity Category

Business Continuity capabilities include:

- **Backups** is the ability to make reliable copy of data in a reliable way for retrieval as needed to meet business and security goals, Recovery Point Objective (RPO), and Recovery Time Objective (RTO). Content to be backed up includes: orchestration framework data and configuration, application data, logs, and customer data.
- **Disaster Recovery** involves the use of automated mechanisms to resume processing of transactions hosted in one physical environment, in a different physical environment in the event that the physical environment where the transactions were originally being processed becomes unexpectedly unavailable.
- **Support** is the ability to troubleshoot an environment, ask questions, submit tickets, integrate into existing ticketing systems, and escalate issues to an appropriate entity for a timely response depending on criticality and support level. Support may also require granting ability to access relevant resources to perform troubleshooting and remediation activities.

Finance

The capabilities within this area enable you to establish and enhance your existing finance processes to be cloud ready in order to establish and operate with cost transparency, control, planning, and optimization. Additionally, manage your records and resource inventory while meeting compliance and regulatory needs.



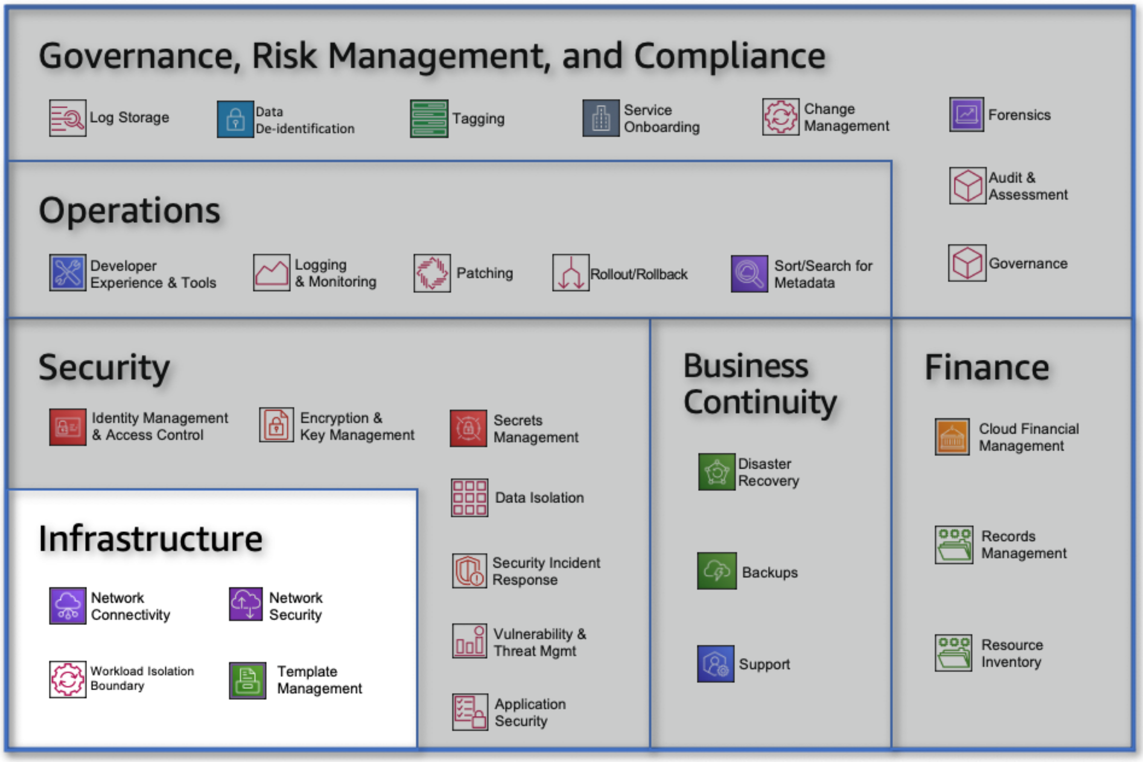
Finance Category

Finance capabilities include:

- **Cloud Financial Management** provides the ability to manage and optimize your variable expense for cloud services. This capability includes near real-time visibility as well as cost and usage analysis to support decision making (e.g., spend dashboards, optimization, spend limits, chargeback, anomaly detection and response). This capability also includes budget and forecasting, to enable you to have a defined cost optimized architecture for your workloads, to select right pricing model, attributing cost of resources to the relevant teams. This enables you to track, notify, and apply cost optimization techniques across your environment and resources. Expense information is centrally managed and consumed, and access to critical stakeholders can be provided for targeted visibility and decision making.
- **Resource Inventory Management** enables visibility and configuration of cloud-based resources that make up an IT-level service or workload. Resources are tracked in the environment along with associated configurations via a system of record (e.g., CMDB for ITSM-managed environments) to enable a larger IT-level system of record for visibility and configuration management of all software, hardware, and firmware resources in the cloud environment.
- **Records Management** enables you to set retention of data according to your internal policies and regulatory requirements, including how to transition data to archive before it is deleted. This data can include financial records, transactional data, audit logs, business records, personally identifiable information (PII), or other data subject to retention policies.

Infrastructure

The capabilities within this area enable you to design, build, and manage a secure and highly available cloud infrastructure. Use practices such as Network Security to design and implement security policies and controls across different levels of the networking stack, and Workload Isolation Boundary to isolate environments that contain your newly migrated workloads. If you are migrating apps from on premises or building them natively in the cloud, the infrastructure that you build on should be both secure and reliable.



Infrastructure Category

Infrastructure capabilities include:

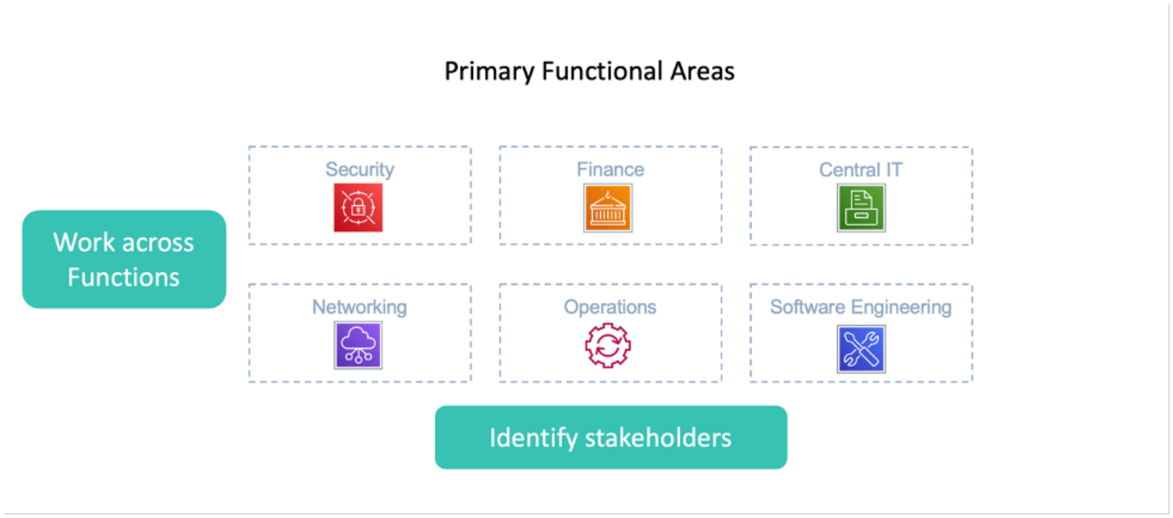
- **Network Security** enables you to design and implement security policies and controls across different levels of the networking stack to protect your resources from external or internal threats to ensure confidentiality, availability, integrity, and usability. This capability includes prevention, detection, and blocking of anomalous network traffic based on monitoring of ingress/egress and lateral data movement.
- **Network Connectivity** enables you to design, build, and manage a secure and highly available network cloud infrastructure. This capability provides best practices and resources to automate network infrastructure build, configuration, and expansion.
- **Template Management** is the ability to create and group reusable templates in a central repository to quickly deploy, manage, and update infrastructure, schemas, golden images, and resources across the environment. This capability includes the necessary processes to create, test, update, and validate the templates when required. These templates are pre-approved implementation patterns using already approved and onboarded AWS services, and are ready to be used by different teams based on requirements.
- **Workload Isolation Boundary** enables you to create and manage isolated environments to contain your newly created or migrated workloads. This approach reduces blast radius of vulnerabilities

and threats, and eases the complexity of compliance by providing mechanisms to isolate access to resources.

Working with the capabilities

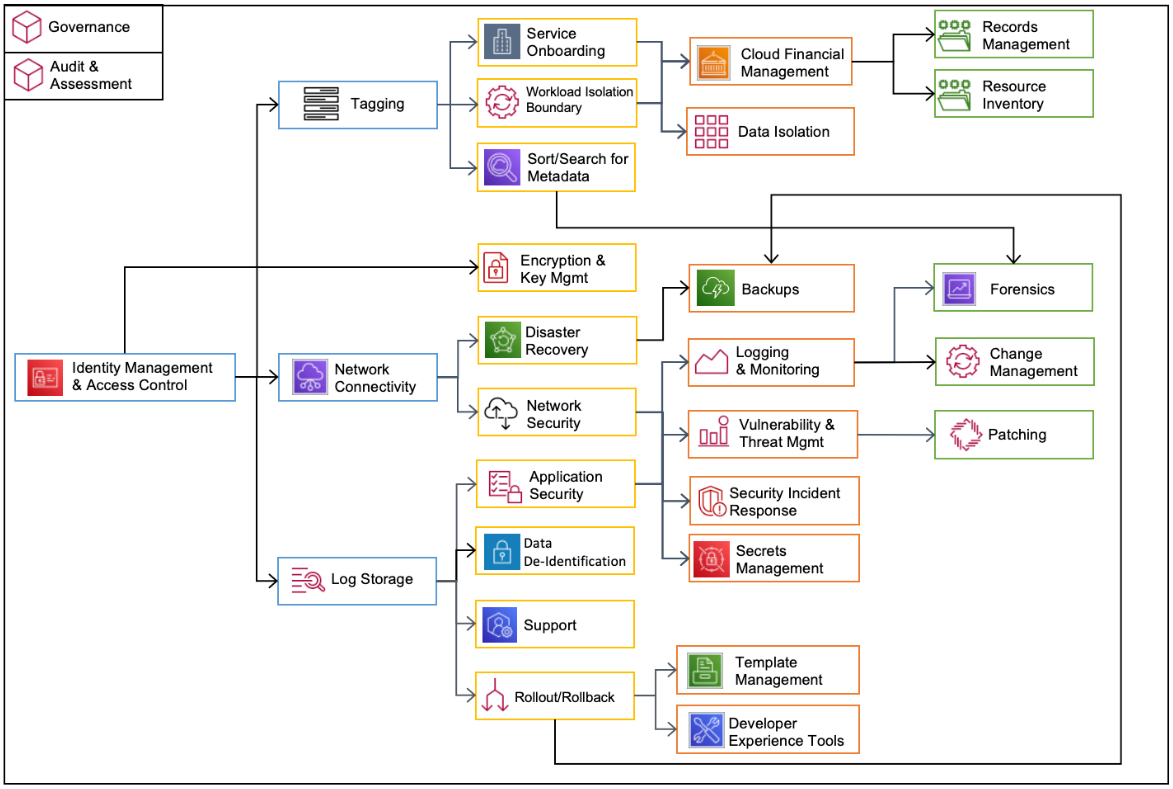
Each organization’s cloud adoption journey is unique. To successfully build your cloud environment, you need to understand your organization’s current state, the target state, and the transition required to achieve the target state. As you work on your plan to establish your environment, these capabilities can help you drive the conversation and decisions across relevant stakeholders (identified by the functional areas for each capability).

The functional areas within each capability are there to help identify owners and stakeholders that will be responsible and accountable for each capability. Each capability has a single, primary functional area, which indicates the owner accountable for the capability. However, most capabilities are also relevant to other functional areas, which indicate the stakeholders responsible for providing inputs, and help to make decisions within a capability.



Primary functional areas

The following graph shows a path that you can follow when planning your environment. It’s based on dependencies between capabilities, and can be used to create a project plan for the implementation of capabilities in your environment. In addition to the dependencies shown (via the arrows), some capabilities apply to the overall environment (for example, Governance, and Audit & Assessment).



Capability dependency guided path

The following foundational capabilities based on AWS best practices and guidance, can help you get started with building your environment.

Topics

Identity Management & Access Control capability

The Identity Management & Access Control (IMAC) capability will help you build and monitor IAM permissions in your environment. These capabilities will enable you to structure your organization, organize your resources within defined isolated groups following the principal of least privilege (PoLP). The following guidance will help your team develop a framework to manage your environment and provide access to your services.

Category: Security

Functional Areas:

- Security (Primary Functional Area)
- Operations
- Central IT
- Software Engineering

Personas:

- **Cloud Team** - the team(s) who make cloud available to customers.
- **Identity Management Team** – the members of the cloud subject matter expert (SME) team responsible for Identity Management and Access control in the cloud.
- **Information Security Team** - the members of the Information Security team responsible for security in the cloud.
- **Consumer** - everyone who needs to access the cloud platform.

Scenarios:

- **CF2 - IMAC1: Create a structure to represent and organize identities and roles in their environment**
- **CF2 - IMAC2: Enable preventive access controls across the environment**
- **CF2 - IMAC3: Establish a single point of management for Access and Authorization for the Cloud environment**
- **CF2 - IMAC4: Manage the lifecycle of identities**
- **CF2 - IMAC5: Enforce Multi-Factor Authentication on user access as needed**
- **CF2 - IMAC6: Implement Least Privilege permissions**
- **CF2 - IMAC7: Implement zero trust principles**
- **CF2 - IMAC8: Implement Data Perimeter**
- **CF2 - IMAC9: Use Infrastructure as Code templates for Identity policies, roles, and preventive controls**

Topics

Guidance

When you are building your environment, access to your platform, your resources, and your applications needs to be established. First to build the environment, and second, to operate the environment through the established capabilities and services you build on it.

As you structure your environment, you want to delegate administrative tasks to different teams, and separate responsibilities across different functions. For example, implementing security tooling, managing the network, or creating central repositories. Different teams may be responsible, and granting access to administer and consume this resources and services you are building within the environment.

Access to your environment should be secured to all users, regardless of the function they will be responsible for. Enabling a form of Multi-Factor Authentication (MFA) for every user is a requirement in order to meet a minimum-security standard.

Structuring the environment

Once your roles have been defined and you have decided what services you will start using, you need to structure your environment in a way that allows you to assign and separate the responsibilities previously described. It's recommended that you start small where you can, and separate the security functions, workload environments (separating production from the rest of the environments), and sandbox environments. ***You can achieve this by using a mechanism to create isolated group of resources from each other.***

You can group workloads with a common business purpose in a distinct isolated group of resources. The enables you to align the ownership and decision making with the isolated group of resources and avoid dependencies and conflicts with how workloads in other isolated group of resources are secured and managed.

Workloads often have distinct security profiles that require separate control policies and mechanisms to support them, you can **apply distinct security controls by environment.**

When you limit sensitive data stores to an account that is built to manage it, you can more easily **constrain the number of people and processes that can access and manage the data** store. This approach simplifies the process of achieving least privilege access.

In the early stages of a workload’s lifecycle, you can help **promote innovation** by providing your builders with separate isolated group of resources in support of experimentation, development, and early testing.

Organizations often have **multiple IT operating models** or ways in which they divide responsibilities among parts of the organization to deliver their application workloads and platform capabilities.

Additionally, creating isolated group will help you organized your resources **based on their function**, and **share them across these the** different isolated groups when needed. Restrictions can also be applied across isolated group of resources that perform a similar action **applying common policies.**

Defining functions and responsibilities to manage your environment

Federated access grants you the ability to efficiently manage the access to the environment, and should be established and operated centrally. The benefits of managing your identities and controlling access to your environment centrally, allows you to quickly create, update, and delete the permissions and policies you need to meet your business requirements. From granting or revoking permissions to specific users or roles, or by establishing preventative controls on your overall environment, your security teams can manage access to the environment from one place.

The responsibilities to perform certain actions of the environment need to be separated. Granting permissions to perform only the necessary actions to specific roles, and users, depending on the purpose of the service being established, achieving a least privileged access model. You also need to ensure that you group different users by their job family to access different tools with a different set of permissions, and that regardless of the user, there may be some preventative controls needed to set centrally to prevent access or modification to certain resources and areas of your environment.

It is recommended that you establish Multi-Factor Authentication (MFA) for every role that has access to your environment, a minimum requirement is to establish MFA for administrator roles. This adds an extra layer of protection on top of your username and password. Additionally, it is very important that every ***root user*** has MFA enabled. For access to your organizational account, the root user MFA needs to be enabled, and the access key and password should be stored separately.

When establishing your environment, you need to set the following functions in your environment, these fuctions will enable you to manage your overall infrastructure:

- The **Organization administrator role** manages the overall environment, creates isolated group of resources, sets guardrails, and delegates the administration of different services to the appropriate isolated group of resources and teams.
- The **Network administrator** manages the network. This function will have access to create and configure network topologies, DNS, VPNs, and build network security across your environment. Network administrators are responsible for securing the network and distributing resources workloads.
- The **Directory Services administrator** manages access to the environment. This function creates, updates, deletes, assigns, and removes access from different users to the environment.
- The **Security administrator** manages security services and tools across your environment, ensuring all your services and workloads are running on a secured infrastructure. This function has access to the environment to remediate any possible security threat.
- The **Billing administrator** manages the spend of your environment and creates budgets and alerts based on the forecasting of your expenses. This function is also responsible to pay the invoice for your environment.
- The **Read Only Security** this function is used to monitor the environment. Security administrators can use this function to oversee the environment in real time and interact with the different security tools, without having full access to the environment
- The **Security Audit** is an exclusively read only function intended to grant access to external or internal auditors that need to examine the environment.
- The **Log Storage administrator** manages the Log Storage in your environment. This function creates or updates the resources needed to security and immutably store your logs, and manages the environment when changes need to be applied.
- The **Shared Services administrator** manages all the shared services across the environment. For example, this function is used to set up a central DNS or a central Template Management function.
- The **Support** function will have access to read only permissions for the infrastructure and not the data within each of the isolated group of resources in the environment. This will allow the cloud team to help troubleshooting the environment, and the workload infrastructure when deployed, helping the team solve any issues with the environment or internal workflows. When necessary, it can be used to escalate to find resolution to your cloud service provider.

These functions with the appropriate permissions and boundaries should be assigned to the user groups which job family needs to perform tasks related to the roles above. This will allow them to access, monitor, operate, update, and secure the environment as needed to meet your business requirements. These functions represent responsibilities within your cloud environment, and multiple functions can be assumed by the same team or person.

To achieve this, you will need to build isolation boundaries to limit the access from each group to the services and tools needed to build, deploy, and operate. In some cases, these groups will need to work together to establish a connection to consume services between the boundaries isolating the resources, to create a cohesive environment that is secure and scalable, and provide access to these services to workloads or other foundational services within the environment.

Federated access

Federation is a common approach to building access control systems which manage users centrally within a central Identity Providers (IdP) and govern their access to multiple applications and services acting as Service Providers (SP).

If you already manage user identities outside of AWS, you can use IAM *identity providers* instead of creating IAM users in your AWS account. With an identity provider (IdP), you can manage your user identities outside of AWS and give these external user identities permissions to use AWS resources in your account. This is useful if your organization already has its own identity system, such as a corporate user directory or via an external Identity Provider, where you can manage the credentials, (such as active directory) and use the same identity store to grant access to AWS with the existing user lifecycle and passwords policies automatically enforced (such as if a user leaves the company, they can simply delete the user's corporate identity, which then also revokes access to your environment). It is also useful if you are creating a mobile app or web application that requires access to AWS resources.

When you use an identity provider (IdP), you don't have to create custom sign-in code or manage your own user identities. The IdP provides that for you. Your external users sign in through a well-known IdP, such as a log in with Amazon, Facebook, or Google. You can give those external identities permissions to use AWS resources in your account. Identity providers help keep your AWS account secure because you don't have to distribute or embed long-term security credentials, such as access keys, in your application.

To streamline the administration of user access in AWS, organizations can utilize a federated solution with an external directory, allowing them to minimize administrative overhead. Benefits of this approach include leveraging existing passwords and password policies, roles, and groups.

Delegating the administration of different services

As your environment grows, the complexity of managing your environment will increase with the amount of isolated group of resources you have. In order to effectively manage your environment, you will need to have a level of automation that matches the complexity of your environment.

Each team in your organization will assume different roles to manage different aspects of the environment. Delegating the administration of the services that the team will be managing in your environment will allow them to establish boundaries for data and security separately, they can build or deploy the necessary automation they need to operate and oversee the environment as they require, without affecting other teams or environments. For example, delegate the management of security services to your security team, delegate the infrastructure deployments, and template management to your Operations and Central IT teams, and delegate the management of your identity to your Identity or Security team.

Establishing preventive controls across your environment

Only granting permissions does not guarantee that our environment is fully secured. To ensure that only the services that are intended to be used by the assigned roles, you need to limit what actions can be performed in your overall environment. For example, to limit the modification of the logs that are being stored in your Log Storage.

To prevent anyone from deleting or modifying these logs by mistake, you need to enable preventive controls to restrict the deletion on the logs in your Log Storage. On AWS this can be accomplished by applying service control policies to your accounts. These policies allow you to limit certain actions within a specific account, but you can also use them to prevent access to services completely, or limit the actions in your environment in specific regions that are not approved for use in your environment.

Log Storage capability

The Log Storage capability enables you to collect and store your environment logs centrally and securely in tamper resistant storage. This will enable you to evaluate, monitor, alert, and audit access and actions performed on your cloud resources and objects.

Functional Areas:

- Security (Primary Functional Area)
- Operations
- Central IT

Personas:

- **Cloud Team** - the team(s) who make AWS available to customers.
- **Security Team** - the members of the Cloud Team responsible for security in AWS.
- **Consumer** - entity within the company that consumes the logs stored within the log storage.

Dependencies: Identity Management & Access Control capability (p. 12)

Scenarios:

- **CF1 - LOG1: Centrally store logs to build a single source of truth**
- **CF1 - LOG2: Logs need to be immutable**
- **CF1 - LOG3: Build a Secure and resilient log storage**
- **CF1 - LOG4: Establish a lifecycle for your logs**
- **CF1 - LOGA5: Store new logs into the log storage**
- **CF1 - LOGA6: Grant read access to the logs**
- **CF1 - LOGA7: Create alerts for your log storage**

Topics

Guidance

The Log Storage capability primary mapping is to the Security Functional Area. The **Security team** should be responsible for implementing this capability according to your governance requirements.

Having a separated Log Storage allows you to establish a secure location where the logs become the source of truth for the actions and events happening in your environment relevant to security and operations. For example access to different accounts, or infrastructure updates.

Log Storage must be tamper resistant and encrypted, and only accessed by controlled, automated, and monitored mechanisms, based on least privilege access by role. Controls need to be implemented around the Log Storage to protect the integrity and availability of the logs and their management process.

Benefits of centralized Logs

As your environment grows and scales with your business needs, creating a single location to aggregate all the logs across your environment helps simplify the analysis and monitoring of the logs. Additionally, it makes easier to isolate access to the environment logs centrally, controlling who is able to consume the logs. Allowing you to create different dashboards and tools for your logging capabilities later on.

When your environment scales and distributes across multiple resources and accounts, there are some benefits that centralizing your logs in one place may bring to your environment.

Create a single location for your logs

The logs from your environment are contained within one place. This enables you to monitor your environment centrally and simplifies your operations. It also creates a single source of truth across your resources, security, and operations logs that your operations and security teams can use as an input for monitoring tools.

Securing your logs

When the logs in your environment are stored in a central location it is easier for you to establish controls to access, isolation, and disaster recovery plans for resiliency. It's recommended that you set controls to protect your environment, to build a tamper resistant environment where your logs will be stored. These controls can restrict permissions within your environment, and monitor actions. You should also restrict human access to your Log Storage, use automations to run your processes and deploy changes to your Log Storage, and set alarms that alert you when anomalies and/or unauthorized actions are attempted within your Log Storage environment.

Protecting your logs with centralized controls

When all of your logs are stored in the same isolated resource group, you can set up centralized controls to protect the Log environment, to controll the access and actions that can be done within the environment. Additionally, setting up different monitoring capabilities so you can react proactively when certain events happen. There are two major level of controls you can set you protect your environment:

- **Preventive Controls** are passive controls which enable you to prevent actions on the environment. They can help you limit the number of actions, roles, services, and regions within your Log Storage, protecting it from changes to configuration, and restricting access and permissions.
- **Detective controls** are implemented to actively monitor the environment. This allows you to create **alerts** based on unwanted or unexpected actions taken within the environment. Optionally, you can trigger **remediation actions** that can automatically mitigate the risks within the Log Storage environment.

What kind of logs can I store?

When you start building your Log Storage, start by storing your security and operations logs across your environment, and be prepared to store different kind of logs (such as Network logs, Access logs, Financial logs, DNS logs, Inventory Records, or Change Management Records). In order to minimize and maximize efficiency, each type of log needs a different strategy, based on the different requirements for each log, the frequency they are delivered, the quantity of logs that are delivered, as well as their retention policies for each kind of log.

Audit logs

Audit logs include everything that you are doing in your environment to ensure you are complying with the policies set for your resources, and your customers. These logs include access logs and action logs that affect customer data or production environments. Additionally, these logs can be subject to a certain compliance requirements depending on the kind of information being stored, and the policy may require that they are stored for a longer period of time, usually more than 12 months. However, based on your policy, and how often these logs are accessed they can be moved to a cheaper storage tier, or they can be archived.

Metric logs

Metric logs help you keep track of resources in your environment. You can create a metric that allows you to measure the quantity of a given resource or event within your environment. These logs can be used to generate events based on different thresholds. The lifecycle for metric logs is medium, on average of 30-60 days. Then, they can be moved to a more economic storage option, archived or deleted, based on your policy requirements.

Configuration logs

Configuration logs help you keep track changes in your environment, and see what and when configurations are being deployed across your infrastructure and applications. The logs can be classified based on your environment, and different policies and life cycles may affect different environments. Based on policy, the lifecycle for configuration logs can vary based on events, time, or use case, and the logs can be kept for 6-12 months. Then, they can be moved to a more economic storage option, archived or deleted.

Networking Logs

Networking logs give you an overview of what is happening on your network. Due to the nature of this logs, and the amount and frequency they are generated, you may choose to not store the logs as long as necessary to be able to analyze traffic anomalies. The lifecycle for networking logs is usually short, on average 2-6 days based on the amount of network traffic. Then, they can be moved to a more economic storage option, archived or deleted, based on policy.

Tagging capability

Tagging enables you to group sets of resources by assigning metadata to AWS resources for a variety of purposes, such as access control, Cloud Financial Management, and automation (such as patching for select tagged instances). Tagging can also be used to create new resource constructs for visibility or control (such as grouping together resources that make up a micro-service, application, or workload). Tagging is fundamental to providing enterprise-level visibility and control.

Functional Areas:

- Central IT (Primary Functional Area)
- Finance
- Security
- Software Engineering

Personas:

- **Cloud Team** - the team(s) who make AWS available to customers.
- **Security Team** - the members of the Cloud Team responsible for security in AWS.
- **Finance Team** - the members of the Finance Team responsible for reporting, allocating and forecasting cloud costs.

Dependencies: Identity Management & Access Control capability (p. 12)

Scenarios:

- **CF23 - TAG1: Define your tagging requirements and strategy**
- **CF23 - TAG2: Build a tagging dictionary**
- **CF23 - TAG3: Define and build a tag deployment mechanism**
- **CF23 - TAG4: Define an enforcement process for your tagging strategy**

Topics

Guidance

Tagging is the act of assigning metadata to the different resources in your AWS environment. As metadata, tags allow you to assign additional labels to these resources for you to identify them according you your business needs. It's recommended that you define a tagging strategy for your environment, this will enable you to confidently and efficiently identify resources across your environment and teams.

Having a separated Log Storage allows you to establish a secure location where the logs become the source of truth for the actions and events happening in your environment relevant to security and operations. For example access to different accounts, or infrastructure updates.

It is important to define a strategy to tag your resources as soon as possible when establishing your Cloud Foundation on AWS, this will enable you to find resources and environments quickly, as your overall environment expands and matures. When defining your tagging strategy, you need to determine the right tags that will help you gather all of the information you will need in your environment for the following scenarios:

Tags for workload and ownership

You can use tags to help you organize and display the resources that are owned by the same team or developer, as well as the resources that belong to the same workload across your environment. These tags can also help you identify what resources within a workload belong to a specific software development life cycle (SDLC).

Tags for cloud financial management

You can create filter views and reports for the resources associated with a specific tag, which will enable you to create budgets and forecast your spend based on specific tags. Being able to control how much you are spending on AWS and related in your environment, can help you reduce your costs in the long term.

One way to you can achieve this is by utilizing Cost Allocation Tags. These tags allow you to track your AWS costs on a detailed level. After you activate Cost Allocation Tags, AWS uses the Cost Allocation Tags to organize your resource costs on your cost allocation report to make it easier for you to categorize and track your AWS costs. AWS provides two types of cost allocation tags, `AWS generated tags` and `user-defined tags`.

Tags for regulatory scope definition and security risk management

Some of the resources in your environment may handle confidential information, personal information, or data that is subject to a specific compliance framework. Assigning tags to identify these resources allows you to ensure that the correct access controls and security mechanisms are established and working as intended.

Tags for operations and automation

When your resources are identifiable through tags, you can filter resources during your automated infrastructure activities. For example, when deploying, updating, or deleting resources within your infrastructure. Additionally, you can use tags to stop or start an entire fleet of resources according to your business needs.

Tags for operational support and disaster recovery

You can use tags to identify the kind of support a group of resources may need, and as part of your incident management process. Tags can be assigned to resources when they are isolated, or when they are on standby before deleting them or archiving them. This can help your support teams to identify the resources within a workload that need to be worked on. Tags can also be used to identify the frequency your resources need to be backed up, and where the backup copies need to go or where to restore the backups.

Tags for Attribute-based access control

In addition to role-based access control (RBAC), tagging your resources enables you to define and enhance the security of your resources in the environment. You can limit access to certain resources for roles in different environments, and you can also use tags to grant a temporary elevated access to certain resources. For more information, refer to the What is ABAC for AWS? documentation.

Authorization-based access control (ABAC) is not supported for all services. For information on what services support tags refer to, the service table. In the table, locate the service and check the **Authorization based on tags** column. You can also select the service name for additional documentation on authorization and access control for the service.

Choosing tags for your environment

Across the different kind of tags, you have to define for your environment, you need to decide what tags will be the **mandatory tags** or **discretionary tags**. Additionally, you need to define what resources need to tagged, and define detection and enforcement mechanisms to ensure all the required resources have the mandatory tags. When building an environment with multiple accounts, every account in the environment should have the mandatory tags that allow you to identify what the purpose of the account is for and who is responsible for the resources in that account.

> **Note**
> When building your tag strategy personally identifiable information (PII) should not be used to label your resources, as tags are not encrypted and are visible across your environment. Codify these values, so you can identify owners internally.

Mandatory tags are the set of tags that every resource should have, regardless of purpose. These tags will enable you to identify the necessary metadata to identify the resource.

The list of recommended **mandatory tags** includes:

- **Owner**- This tag indicates who is the owner and main user of the resource, this can be a team or an individual.

> **Note**
> The Owner is not always the user who created the resource.

- **Business Unit** - This tag identifies the business unit the resource belongs to.
- **SDLC Stage** - This tag indicates if the resources are being used for Production or for non-Production. (For example, development, test, or sandbox.)
- **Cost Center** - This tag specifies the budget or account that will be used to pay for the spend associated with the tag.
- **Financial Owner** - This tag identifies who is responsible for the costs associated with the resource tagged with a specific tag.
- **Compliance Requirement** - This tag identifies the resources that are subject to a specific compliance framework. (For example, PCI-DSS or HIPAA).

Discretionary tags are the set of tags that must be defined as part of your tagging strategy, so they are available to be assigned to resources that need them (for example, temporary elevation of permissions, or data sensitivity).

The list of recommended **discretionary tags** includes:

- **Workload ID/Name** - This tag indicates if the resource belongs to a specific workload. The value can be the workload ID or name.
- **Data Sensitivity** - This tag indicates if the resource contains, stores, or uses any type of special or sensitive data.
- **Environment version** - This tag indicates the version of the environment, in case the same workload has more than one environment associated.
- **Workload type** - This tag indicates the type of workload the resources belong to. Some workload types examples are confidential, internal, or critical.
- **Backup** - This tag indicates if the resource needs to be backed up based on the type of workload and the data that it manages.
- **SLA level** - This tag indicates SLA Requirements.
- **Lifespan** - This tag indicates the lifetime of the resources of the workload. If exceeded, these resources should be reviewed, replaced, or isolated.

Tagging Standards

As you define your tagging strategy, a naming convention needs to be established for the different tags across your environment ensuring a standard, and making it easier for the tagged resources to be identified. Tags enable you to identify resources, and having no more than 50 tags per resource will allow you to keep your tag strategy manageable in your environment. The following are examples for tag key names and values:

```
example-company:owner = SecOps
```

```
example-company:cost-center = 5432
```

```
example-company:financial-owner = Security
```

Resources that need to be tagged

There are resources that always need to be tagged in your environment, because it is critical to have the identifiable information about the resources at all times. The resources in this category are meant to be persistent, and in some occasions, they act as resource containers for other resources. These resources include, but are not limited to, accounts, critical workloads, and shared infrastructure. For these resources, you should aim to tag 100% of the resources which will allow you to identify the spend, access, ownership, and permissions for the tagged resources.

However, as operational complexity increases, and the level of automation to manage tags becomes more demanding, you may choose to not tag certain types of resources that are ephemeral. These resources should run within a resource container that is properly tagged to allow you to identify and trace what happened within that environment, but enforcing the tags on these types of resources may not be necessary if they do not belong to critical workloads or applications.

Enforcing tagging

Because of the importance of tagging and the level of complexity, it's recommended to automate the tagging process when possible. This will reduce the human error that can be introduced when tagging critical resources, and will minimize the number of resources that are not identifiable due to the lack of tags. When possible, creating tag policies in your organization can help you ensure that the tags assigned to resources have the correct value assigned.

Additionally, automation needs to be established in the environment to discover resources with missing tags or resources that are not compliant with the established tagging strategy. Once the resources have been identified, a report including these resources on the environment needs to be sent to the relevant stakeholders, to evaluate and make a decision to remediate the situation, if needed.

Based on the results of this report, if a situation where persistent resources that are identified as non-compliant or have missing tags is given, it should be remediated immediately, by assigning a default pre-defined tag value defined as part of your tagging strategy, or if pre-defined tag doesn't exist deleting the non-compliant resources.

For example, when deploying resources using infrastructure as code, resources that don't have the required tags assigned, should not be allowed to be created. This remediation should come from preventive controls. For more information, refer to Establishing preventive controls across your environment (p. 16).

Build a tagging dictionary

As you prepare to set up tagging across your environment, creating a reference where the different tag values each resource must have associated becomes important. Developers and cloud architects need to have access to the required tags for each of the resources they will be creating in their environments, as well as the available values for each of the tags created.

It's recommended that you define and build a tagging dictionary where all these values are available for developers, cloud architects, and environment operators. In order to add, update, or remove values for each of the tags included in the tagging dictionary, you need to establishes processes where all the relevant stakeholders can provide their inputs, when a tag becomes standard. This will ensure that all relevant stakeholders involved in the definition of the tags in your environment are aware of any changes they need to provision and deploy across their resources.

Defining tags for Attribute-based access control

As part of your tagging dictionary, you should define certain tags that can be used to access specific environments resources based on the tags attached to a role at a certain time. These tags can be used for a temporary escalation of privileges or for deploying changes through infrastructure as code that other identities may not have access otherwise.

It’s recommended that you define and build a tagging dictionary where all these values are available for developers, cloud architects, and environment operators. In order to add, update, or remove values for each of the tags included in the tagging dictionary, you need to establishes processes where all the relevant stakeholders can provide their inputs, when a tag becomes standard. This will ensure that all relevant stakeholders involved in the definition of the tags in your environment are aware of any changes they need to provision and deploy across their resources.

Next steps

If you are still exploring the cloud, AWS recommends that you deploy a few proof-of-concepts (POCs) to demonstrate business value to your stakeholders. **If you are ready to start building a cloud environment** to host your workloads on the cloud, this set of defined capabilities can help you build your foundational cloud environment. Before getting started with your cloud adoption, AWS recommends that you complete the following activities, and reach out to your account team for more information:

- Review the list of capabilities and create a timeline for implementing capabilities, accounting for any dependencies.
- Identify the stakeholders in your organization that are responsible for each capability.
- Create an implementation plan and a timeline to build your cloud environment.

As your requirements change, to help you grow your presence in the AWS Cloud, you can use the defined capabilities to build your own approach using your own tools.

Conclusion

This whitepaper introduces a capability-based approach to establishing the foundation for your AWS environment, and helps you identify the relevant stakeholders needed to make important decisions along your journey. The defined *capabilities* in this paper are based on current AWS best practices, and the experience of thousands of customers that have built their foundational environment on the AWS Cloud.

Contributors

Contributors to this document include:

- Alex Torres, Sr. Solutions Architect, Amazon Web Services
- Eamonn Faherty, Principal Solutions Developer, Amazon Web Services
- Fabian Labat, Sr. Solutions Architect, Amazon Web Services
- Glenn Dasmalchi, Sr. Manager Tech Leader, Amazon Web Services
- Sam Elmalak, Tech Leader, Amazon Web Services
- George Rolston, Sr. Solutions Architect, Amazon Web Services
- David Rowe, Sr. Solutions Architect, Amazon Web Services
- Brandy Smith, Sr. Solutions Architect, Amazon Web Services

Further reading

For additional information, refer to:

- AWS Architecture Center
- AWS Whitepapers & Guides

Document revisions

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

update-history-change	update-history-description	update-history-date
Whitepaper updated (p. 29)	Whitepaper updated to include capabilities guidance	May 2, 2022
Initial publication (p. 29)	Whitepaper published	November 17, 2021

Appendix A: Capability structure and example

Capability structure

Definition

The definition includes a high-level description of what the capability will help you enable in your cloud environment.

Scenarios

Scenarios are a set of use cases that expand the capability definition, and detail what parts of your environment the guidance included in the capability solves. Each capability provides a baseline, which establishes the minimum requirement for the capability, and can be expanded and customized to add additional scenarios based on your requirements or as your business needs mature and your AWS presence grows.

Guidance

This section outlines prescriptive recommendations for how the capability should be built in your environment to implement the included scenarios. It also includes responsible stakeholders, and a description of how the capability will work in the overall environment. Additionally, this section includes the people and recommended skillsets necessary to successfully establish the capability in your environment.

Implementation guidance

Each capability provides prescriptive, opinionated AWS guidance to establish the capability in your environment. Runbooks are included to help you operate the capability efficiently in your environment using AWS services.

Capability example - Log Storage

Definition

The Log Storage capability enables you to securely collect and store your environment logs centrally within an immutable storage. This will enable you to evaluate, monitor, alert, and audit access and actions performed on your AWS resources and objects.

Scenarios

- Your cloud team wants to log *individual user access* to resources, and what systems are accessed and actions taken (*Individual user access* also includes access by system administrators and system operators).
- Your cloud team wants to set controls to prevent modification of the related logs.

- Your cloud team wants to set controls to prevent unauthorized access to logs.
- Your cloud team wants to generate logs that can show if inappropriate or unusual activity has occurred.
- Your cloud team wants to store logs in near real-time for resiliency during a determined period of time (matching your governance requirements).
- Your cloud team wants the stored logs to be encrypted at rest.

Guidance

The Log Storage capability primary mapping is to the **Security Functional Area**. This means the **Security team** should be responsible for implementing this capability.

When establishing your capability, the builders owning the implementation will need to receive inputs from the owners of additional functional areas to ensure the proper interlock of the functions in the cloud environment. The list of secondary functional areas required are:

- Operations
- Central IT

Having a separated Log Storage allows you to establish a secure location where the logs become the source of truth to show what is happening in your environment related to security and operations. As your environment expands to accommodate your business needs, centrally aggregating the information will enable you to later build monitoring and observability capabilities, to monitor in near real-time what is happening across your environment.

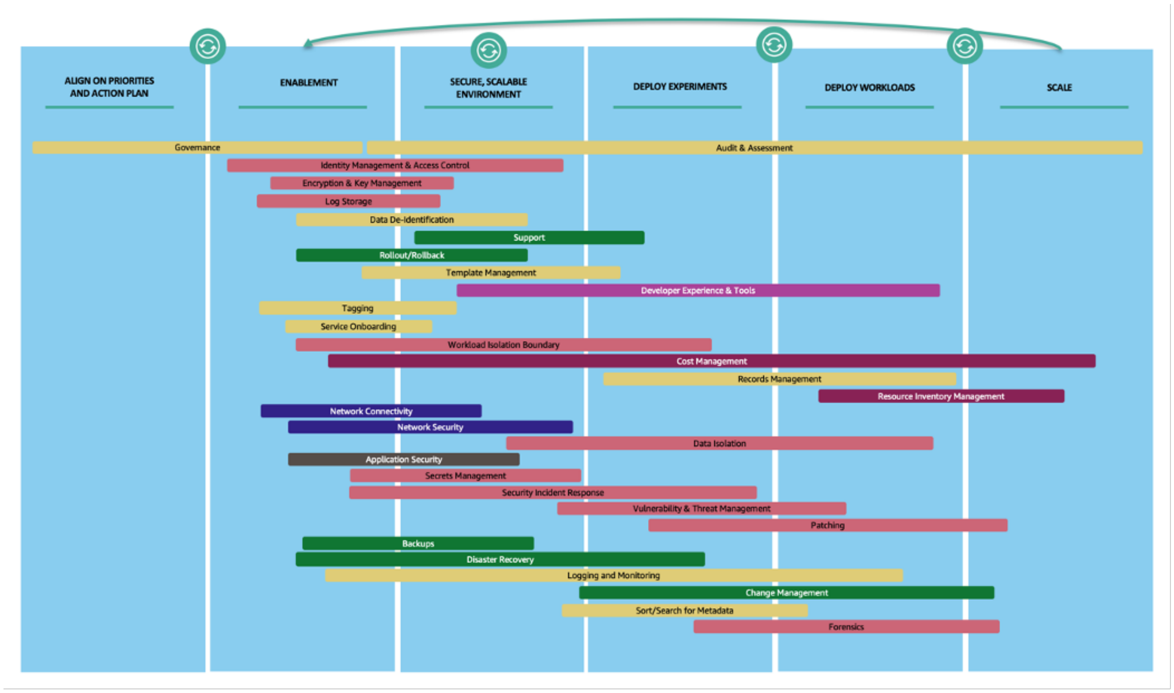
The Log Storage must be secured, built for resilience, to avoid tampering with the logs, and only accessed by controlled, automated, and monitored mechanisms, based on least privilege access by role. The following controls need to be implemented around the Log Storage to protect the integrity and availability of the logs and their management process. The logs delivered to Log Storage should be encrypted, and the encryption key access and permissions should also be based on least privilege permissions.

- **Detective controls** should be implemented to alert and remediate the collection of permissions used on the log storage, and to actively monitor access to the logs within the Log Storage.
- **Preventive controls** should be implemented to protect from changes to your configuration and access in your Log Storage, and restricting permissions on your Log Storage.

The Log Storage should also have retention policies, establishing a lifecycle for your logs based on your governance and data retention policy requirements (for example, automatically archiving infrequent access or delete the logs over time to reduce the cost while meeting retention requirements).

Appendix B: Sample timeline

In this section you can find a sample timeline that includes all 30 capabilities that are needed to meet your requirements when establishing a foundational cloud environment on AWS. Enable your teams to work with the capabilities, and start building an environment which initially allows you to deploy experiments and then workloads. As you scale or your business needs evolve, you can assess your established capabilities, and enhance them as necessary to meet your requirements.



Sample timeline of capabilities

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*.