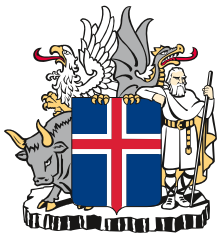


Governance Framework for Amazon Web Services in Iceland

Policies to utilize Amazon Web Services in the Icelandic public sector.

V1.0

31.10.2024

1. General / Inngangur

Eftirfarandi skjal er unnið af fjármála- og efnahagsráðuneytinu og gefið út og samþykkt í október 2024 í útgáfu 1.0. Undirbúningur þess er samvinna margra ríkisaðila í formi vinnuhópa og rýni ásamt aðkomu erlendra ráðgjafa á sviði skýjalausna. Eigandi skjalsins er skrifstofa stjórnunar- og umbóta.

Tilgangur skjalsins er að setja samræmda umgjörð um innkaup, umsjón og notkun AWS skýjaþjónustunnar. Skjalið er liður í innleiðingu Öryggis- og þjónustustefnu um hýsingarumhverfi – stefnu um notkun skýjalausna og aðgerðaáætlun henni.

Með því að útbúa og innleiða samræmda umgjörð um AWS skýjaþjónustuna þar sem sama hönnun er innleidd oft næst að tryggja öryggisstig og stytta þróunar- og afhendingartíma nýrra vara hjá ríkisaðilum. Umgjörðin tryggir auk þess að uppsetningar verði samræmdar, hvort sem innleiðing og rekstur er í höndum miðlægs aðila, ríkisaðilans sjálfs eða er í höndum þjónustuaðila af einkamarkaði.

Mun þetta skjal verða hluti af heildstæðri umgjörð um högun upplýsingatækni ríkisins. Er það birt m.a. á:

- Island.is: Stefnur og skilmálar
- Stjórnarráðið: Verkefni – Upplýsingatæknimál ríkisins

Þar sem umsjónarviðmót skýjaþjónustu AWS er á ensku er meginmál þessa skjals á ensku til að tryggja að hugtakanotkun sé samræmd.

Útgáfa	Lýsing	Dags.
1.0	Fyrsta útgáfa	2024-10-31

In 2021 Iceland defined “Strategic Cloud Policies” for using and promoting public cloud services in Icelandic government and public sector organisations.

In 2022, The Icelandic Ministry of Finance and Economic Affairs launched an initiative to define the principles for utilization of Microsoft Azure services. The main target for the Governance Framework is to have a ready framework and policies that can be utilized to shorten implementation time, reduce cost, standardize implementation and implement required cost and security controls of cloud platforms/projects deployed in Azure.

In 2023, the Icelandic Ministry of Finance and Economics Affairs launched an initiative to define similar principles for Amazon Web Services (AWS) as the public cloud service provider (hyperscaler) used by a variety of institutions in Iceland. The same policies shall govern the usage of any platform to ensure a baseline of security and compliance as well as operational effectiveness and cost control.

The “Strategic Cloud Policies” document is further referred to in this document as **Strategic Cloud Policies**.

1.1. Reasons for defining a Governance Framework

The reasons why the governance framework is defined are:

1. **Ease-of-use**

   Unifying the policies means that there are common, repeatable procedures which in turn lead to faster implementation of services, when there is a clear understanding on how and why to implement the services. When developed further, these common practicalities can be transformed in to Infra-as-Code and used with automated DevOps tools.

2. **Visibility**

   Increasing visibility to the AWS usage through chosen procurement models enables the Icelandic institutions to have a clear understanding to their costs and also grants visibility to the overall AWS usage for the entire Icelandic governmental agencies.

3. **Safety**

   When agreed policies are put into action, they enable compliance auditing features and – if enforced – leads into ensured compliance in the environment for security, privacy and cost.

4. **Centralized assistance**

   Common policies enable solution partners to better assist each institution to use the AWS services efficiently as well as enabling inhouse teams to deploy and use the services faster.

2. Terminology

The following terms are used across this document:

- AWS Account – a logical container for AWS resources
- Root user – unique administrative user on an AWS account
- AWS Organization – a collection of AWS accounts which provides a hierarchical tree-like structure for centrally managing AWS accounts
- Organizational Unit (OU) - a logical grouping of AWS accounts or other Organizational Units within an AWS organization
- Service Control Policy (SCP) - AWS organization policy defining restrictive actions on AWS account(s)
- Resource – AWS resource (such as Virtual Machine) which can be deployed to an AWS account
- Region – geographic location which consists of multiple availability zones
- Availability Zone – a logical group of one or more physical data centers. Each Availability Zone is physically separated from each other
- Landing Zone – set of AWS accounts providing supportive functionality to all AWS accounts within an AWS organization
- CI/CD - Continous Integration & Continous Deployment pipeline, used for automation of building software and deploying changes to the target environment(s)
- Control Tower – AWS service that can be used to deploy, configure and manage a landing zone
- CfCT – Customizations for Control Tower, one of the available tools in Control Tower for centrally deploying AWS resources / configurations to AWS accounts in an AWS organization
- Consolidated Billing – combining multiple AWS account's bills into one single AWS bill
- AWS Backup – AWS service for taking and storing backups and managing their lifecycle
- AWS CloudFormation – AWS service which can be used to deploy AWS resources using templates defined in YAML/JSON
- AWS CloudFormation StackSet – way to deploy a single AWS CloudFormation template to multiple target AWS accounts/regions
- AWS Config – AWS service that can be used to record and query the configuration state of your AWS resources
- CloudTrail – audit trail log consisting of AWS API calls made against an AWS account
- AWS Direct Connect – AWS service for connecting data center infrastructure to AWS networks
- AWS IAM Identity Center – AWS service used for centrally managing users, groups and their permissions in an AWS organization

- IAM Identity Center Permission Set – Policy defining what an IAM Identity Center user/group is allowed/denied to perform on a target AWS account
- AWS IAM – Used to manage users/groups/roles/policies within a single AWS account
- AWS IAM user – user created in AWS IAM service which exists on a single AWS account
- AWS IAM role – role created in AWS IAM service which can be assumed either by users or AWS services for pre-defined AWS privileges
- AWS IAM Policy – policy created in AWS IAM service defining a set of allowed/denied AWS actions
- AWS IAM group – group of AWS IAM users
- AWS Route53 – AWS service for registering domain names and managing DNS settings
- AWS System's Manager – AWS service for configuration management and automation
- AWS System's Manager Session Manager – service providing remote access to virtual machines using AWS Systems Manager SSM Agent installed on the target systems
- AWS Security Hub – AWS service providing a centralized view of the security state of your AWS accounts
- AWS VPC – AWS service for managing the network settings of an AWS account/region

3. Procurement

In the **Strategic Cloud Policies**, the following statement is made:

> ***Use trusted purchase channels and be cost conscious***
>
> *Purchase cloud services from selected and shared public channels, and from cloud vendors that have been selected through a formal process. Be cost conscious and use public funds responsibly – only buy what you need at any given time. Allocate costs fairly in the cloud native way – always pay your own use – do not piggyback on others' costs.*

Procurement of services from hyperscalers such as Amazon Web Services (AWS) are in many ways different from conventional hosting services. The procurement methods described in this document are specific to AWS services and do not apply to any other services or service providers.

Conventional hosting and professional services related to conventional hosting and cloud services shall be procured through the Dynamics Purchasing Systems (DPS) provided by Fjársýslan (Central Financial Services).

A centralized tender will be put in place in Q1 2025 that covers AWS so that individual institutions will be able to provision accounts and organizations from a central agreement.

Data Security and Environmental Integrity:

Assurance of data security and environmental robustness is paramount. Contractual provisions devoid of intermediaries empower the government with singular authority over access to its cloud environment.

Cost Control and Oversight:

A joint contractual framework guarantees meticulous cost control and oversight. The financial advantages stemming from a direct business relationship translate into more favourable terms, thereby optimizing fiscal prudence.

Enhanced Operational Efficiency:

The imperative lies in ensuring that cloud services are procured through consistent channels, leveraging features such as self-service, variable usage and flexibility. This strategic approach ensures optimal utility for government entities.

3.1. Aspects of contracting with AWS

When contracting with AWS several aspects need to be considered. The contractual relationship governing the delivery of the service itself, the support provided and the billing for the consumed resources.

Contractual requirements:

Terms and conditions including data protection agreements and terms need to be directly confirmed between the organization using the service and AWS. This can be done directly or using an Enterprise Agreements (EA) governing the contractual terms and conditions (T&C's) of the relationship signed by the organization or an entity charged with the responsibility for the entire public sector.

Support agreements:

Organizations can if needed procure operational services from 3rd parties through a separate procurement process to operate their organizations within AWS. Such 3rd parties act on the delegated authority of the organization. Support agreements with AWS can be added as needed in addition to such services or if the organization is qualified to operate its organization on internal resources.

Billing relationships:

Billing shall always be direct between the organization and AWS. AWS can invoice collectively for each organization (multiple accounts) in a single invoice both through issued invoices and transfers. Multiple organizations can be collectively cost-monitored if needed based on the requirements set forth in this document.

The following table describes the different AWS procurement models and their main differences. There are four ways to procure the AWS consumption:

- Direct models:
    - Pay-as-you-Go: month-to-month agreement model with standard AWS terms and conditions
    - Enterprise Agreement: long-term agreement model
- Partner-led models with differences shown in the below table:
    - Solution provider account model (SPAM)
    - End customer account model (ECAM).

Feature	Customer Agreement / Click-through	ECAM	SPAM	Enterprise Agreement
Procurement	Direct agreement	Through partner	Through partner	Direct agreement
Terms and conditions	AWS standard T&C's	AWS standard T&C's or EA adjusted	Service Provider	AWS, adjustment possible
Ownership	Customer has full ownership	Customers own their linked AWS accounts. The partner owns the master account but not the linked accounts	The partner owns both the master account and the linked accounts	Customer has full ownership
Account security	Customer is responsible for the account security	Customers are responsible for the account security on the linked accounts, SP is responsible of the security of the master account	The partner is responsible of the account security.	Customer is responsible for the account security
Administrative access	Customers have full administrative access	Customers have full administrative access to the linked accounts and may have limited administrative access to the master account	Customers have full administrative access to the linked accounts and may have limited administrative access to the master account	Customers have full administrative access
Billing and payment	Direct: Credit card (transaction per account) or Invoice/Transfer (transaction per organization)	Through partner	Through partner	Direct: Invoice/Transfer
Support	Direct	Through partner (Partner-led) or direct (Resold)	Through partner (Partner-led) or direct (Resold)	Direct
Agreement suitability (AWS recommendation)	Available for all	Recommended for Private sector, where T&C's from AWS are preferred	Default for Public sector	Applicable when a minimum amount of consumption is achieved
Transferability of accounts (workloads) between organizations/service providers	Yes	Yes	Yes (some resources might be shared per reseller)	Yes
Agreement availability to Iceland	NO, standard unchangeable T&C.	YES, with the proper procurement model.	NO, partner/reseller ownership of account structure not suitable.	No, direct contracts do not

3.2. Account structures

AWS has two main structures for account ownership for reseller: The SPAM and ECAM models.

SPAM: Service Provider Account Model

Reseller through solution providers (reseller accounts). All usage of AWS services is governed by the Service Providers (resellers) agreement.

Individual service providers can create their own terms and conditions. Email accounts used for management accounts and all linked accounts are owned by the service provider.

- This model does not give the organization full ownership and control over their organization.
- This model creates increased vendor-lockin as some aspects of the organization could be owned by the service provider.

ECAM: End customer account model:

Under this model the customer agreement (between the organization and AWS) governs all usage of AWS services. Additional terms can apply within the solution providers agreement regarding fees, payment, pricing and taxing.

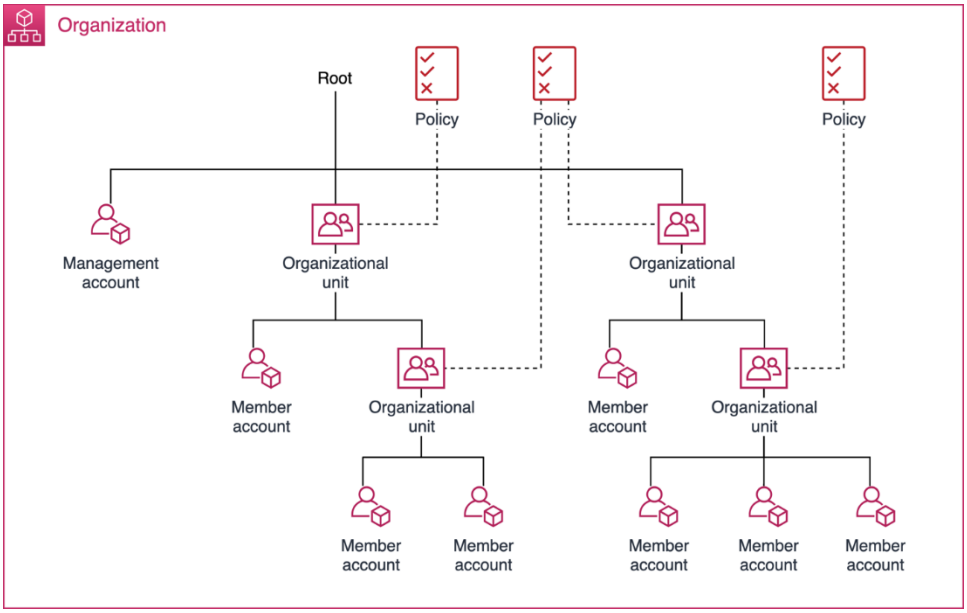
- This model creates an indirect billing relationship with AWS and gives the solution provider (reseller) access to usage data and the ability to modify terms and fees.
- This model does not give the organization complete ownership of their organizational structure within AWS but gives complete ownership of the accounts containing the customer workloads.

3.3. Procurement policies

Purpose	POLICY
Vendor transferability	Resources and configuration (including account structures) shall be owned by the organization and not tied to the reseller or any other intermediary.
Billing relationships	Billing shall be possible from the vendor to individual institutions, collected for a group of institutions or for the entire contractual relationships.
Cost monitoring	A central entity shall have access to detailed cost monitoring information for monitoring, analytics and reporting purposes.
Data access	Access to data within resources shall only be granted by the institution and not the reseller or hyperscaler.

4. Management Architecture

Management Architecture in AWS consists of Organizations, Organizational Units, Accounts and Resources, which create a nested hierarchy. The whole hierarchy of all of these should be utilized to gain all advantages. A basic Management Architecture can be created by hand or with services provided by AWS, out of which a Control Tower based solution is the one that should be favoured.



Example of a well-architected organization structure

4.1. Accounts

The process for creating accounts can be initiated in two different ways; as standalone accounts or through an existing organization. When no organization exists, standalone is the only viable option. In all other cases standalone accounts should be avoided.

In AWS, account ownership is mainly tied to the email address that was given when it was created. Ownership verification is done through email, so it is paramount for the account creator to have control of the email inbox. Each account owner email address should also be unique, since they can only be tied to one AWS account at a time. This can be handled with, for example, email groups or using a plus sign followed by a unique identifier in the email address.

A common naming convention should be used across all accounts. The recommended way is described in the following table.

Section	Description	Reason
Orra-code	3-5 letter code for institution that links to the Orri central ERP system.	**Required** – for cost allocation purposes.
Project/ Purpose	A shortened description/name of the project being deployed.	**Required** – to ensure boundaries between projects This can also be the AWS centralized services
Sub-project/ Purpose	A further separation of purpose of the account.	**Optional** – if a project or a set of projects require shared centralized services that are not organization wide.
Environment	Designation of prod, test, qa, dev	**Recommended** – if not specified the assumption is that the account contains “prod” resources. Centralized services do not require this designation.

Examples:

- Ríkissaksóknari, sakaskrá, raunumhverfi: RSAK-sakaskra-prod
- Þjóðskrá Íslands, einstaklingsskrá, dev: THS-einstaklingar-dev
- Fjármálráðuneyti, Stafrænt Ísland, raun: FJR-island-is-prod
- Network resources used by multiple accounts for Fjármálaráðuneyti, Stafrænt Ísland, raun: FJR-island.is-networking

4.1.1. Root Credentials

Every AWS account has a root user associated with it, which can be used to login to that account. Root user is the most powerful user of the account and should be used only for very few specific operations that cannot be completed without it. The best practices around Root users dictate that every time an AWS account is created, the Root user credentials of the account should be stored in a secure way, for example, by securely printing them out & storing them in a safe.

The Root user should also have multi-factor authentication enabled and the MFA device stored securely. Without the MFA enforcement, an AWS account could be compromised by someone having access to the email box associated with the account's email address. The Root user credentials and the MFA should be tested regularly.

A combination of virtual and physical MFA tokens is recommended. If hardware MFA is used for any root user, a virtual MFA should be defined as a backup since hardware MFA's might stop working.

Root users should replicate the account naming convention, the use of additional addresses per mailbox or the ‘+ sign method’ where available is

recommended. Institutions may want to add ‘aws’ to the email address to designate its purpose within their email address structure.

Examples:

- Ministry of Justice, útgáfukerfi, dev: aws-dmr+utgafa-dev@dmr.is
- Ministry of Finance, Ísland.is, prod: aws-fjr+island.is-prod@fjr.is
- Þjóðskrá, einstaklingsskrá, dev: aws-ths+einstaklingar-dev@skra.is

4.2. Organizations

An AWS organization represents the top level of Management Architecture under which accounts are placed. An organization is a security boundary that allows maintaining consistency across accounts grouped by, for example, institution or a multi-team project. This way it’s possible to centrally apply policies and service-level configurations across multiple accounts. Certain resources can also be shared across accounts within an organization.

AWS organizations can be managed either by hand or with Control Tower. The latter is recommended. When doing so, accounts can be created through automation and can be automatically provisioned to conform with chosen security standards. To achieve this, using for example Customizations for AWS Control Tower (CfCT) is a good idea.

Consolidated billing should always be utilized, so that all charges from all accounts under an organization are combined into a single bill.

When designing and implementing security boundaries for an organization, the management account should always be treated differently compared to member accounts. Access to the management account should be limited to essential personnel only. Furthermore, the management account should only be used for tasks that require the management account. No workloads should be located there.

4.3. Organizational Units

Organizational Units are a construct of AWS Organizations used to group accounts or other OUs together to administer them as a single unit. OUs can exist under other OUs, creating a tree-like structure. The main purpose of using OUs is to apply Service Control Policies to multiple accounts at once. Once an SCP is applied to an OU, all accounts and sub-OUs will inherit it automatically.

OUs should be named in a logical and descriptive manner. Each OU name must be unique to the Root or parent OU that it resides in. It is technically possible to give the same name to two sub-OU’s that reside under different parent OU’s. However, doing so is not recommended.

Similar accounts and workloads should be grouped based on function, business purpose and/or ownership. A deciding factor for the structure is what kind of SCPs you plan to create and use. For example, if there are accounts that only contain CI/CD services and the usage of all other services are to be restricted with an SCP, then it’s a good idea to create an OU for those accounts specifically. As a continued example, if there are multiple teams with their own CI/CD accounts, then it’s a good idea to create sub-OUs for each team under the parent CI/CD OU. Although unnecessarily deep OU

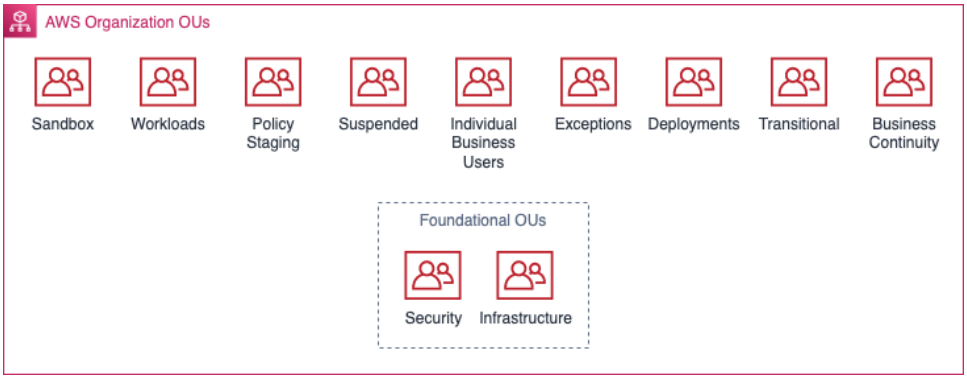
hierarchies should be avoided, it's a good idea to futureproof the OU structure right from the beginning.

OUs can also be used to create a structure for sharing automations across accounts. These can include, for example, CloudFormation StackSets or Customizations for AWS Control Tower (CfCT). In these cases, certain automations or IaC templates would be automatically run on each account that belongs to an OU.

By default, AWS recommends having the following parent OUs under each organization:

- Security
- Infrastructure
- Sandbox
- Workloads
- Policy Staging
- Suspended
- Individual Business Users
- Exceptions
- Deployments
- Transitional
- Business Continuity

Not all of these are needed in most cases and should be taken as a frame of reference. The chosen OU structure should always be based on the use case. Having OUs that have no accounts under them serves no purpose, and they can be created later if required.



Default OUs as recommended by AWS

When AWS services are to be shared across multiple accounts in an organization, care should be taken when deciding where to create and manage each service. Recommended locations for shared services in their respective OU's and/or accounts are detailed in the table below. Note that the services and service types should be seen as examples and actual architectural choices should be based on what is needed.

Organization level **recommendations** for centralized service placement:

OU / Account	Services	Placement
Management account	AWS Control Tower, AWS IAM Identity Center, AWS Systems Manager, AWS Artifact, IAM access advisor	Centralized
Security OU – Security Tooling account	AWS CloudTrail, AWS Security Hub, Amazon GuardDuty, AWS Config, Amazon Security Lake, Amazon Macie, AWS IAM Access Analyzer, AWS Firewall Manager, Amazon EventBridge, Amazon Detective, AWS Audit Manager, AWS Artifact, AWS KMS, AWS Private CA, Amazon Inspector	Centralized
Security OU – Log Archive account	Amazon S3 buckets for central log storage, Amazon Security Lake	Centralized
Infrastructure OU – Network account	AWS PrivateLink, AWS Transit Gateway, AWS Network Firewall, Network Access Analyzer, AWS RAM, AWS Verified Access, AWS WAF, AWS Shield	Centralized or distributed
Infrastructure OU – Shared Services account	AWS Systems Manager, AWS Managed Microsoft AD, IAM Identity Center (delegated)	Centralized or distributed
Workloads OU – Application account(s)	Amazon EC2, Application Load Balancers, AWS Private CA, Amazon Inspector, Amazon Systems Manager, Amazon Aurora, Amazon S3, AWS KMS, AWS CloudHSM, AWS Secrets Manager, Amazon Cognito	Distributed

4.3.1. Workload OUs

Organizational Units containing workloads or workload-oriented resources should be planned with extra care. A workload is a discrete collection of applications, cloud resources, and data that you manage. It can be either an off-the-shelf application or your own custom application, usually consisting of infrastructure resources, data, and the application itself.

There are several ways to arrange the hierarchy for workload OU's and accounts. The most important factor that should be considered is whether there are multiple institutions and/or multiple teams operating in the accounts within the same organization. Another factor to be considered is who has ownership of the organization. This determines the need for institution and project OU’s. If the organization is owned and managed by a single institution, then there is no need for the institution OU. This structure applies to workload OU’s and accounts. For shared/central services accounts and OU’s see the previous chapter.

A common naming convention should be used across all OUs. The recommended way is described in the following table.

Section	Description	Reason
Orra-code	3-5 letter code for institution that links to the Orri central ERP system.	**Required** – for cost allocation purposes.
Workloads	All workload accounts should be within a workloads OU's	**Required** – to separate projects from infrastructure used by all accounts.
Project/Purpose	A short description of project or purpose (same as in the account name requirement)	**Required** – to separate projects within the environment.
Environment	Designation of prod, qa, test, dev	**Recommended** – is needed if different teams have different access to accounts within each environment. Or if different SCPs are to be applied to different environments.

For example, exmpl-project-1-prod placed under the "Workloads" OU would be valid according to the naming convention.

4.4. Service Control Policies

Service Control Policies are a type of policy that can be applied to OUs or accounts. If applied to an OU, all sub-OUs and accounts under that OU will inherit the SCP automatically. SCP's offer centralized control over the maximum available permissions for accounts in an organization. They can be seen as guardrails, limits or boundaries and help ensure that all accounts stay within an organization's access control guidelines. SCPs cannot be used to grant permissions.

Great care should be taken when testing SCP's. Testing should never be done on the Root of an organization or an OU/account that contains critical resources or workloads. Therefore, it is recommended to have an OU that is designated for testing SCPs. While the AWS recommendation is a whole separate OU for this (Policy Staging), in real world scenarios it is often better to use an existing non-production OU that contains an account or accounts with resources in it.

Common examples of recommended SCPs to be forced across the whole organization include:

- Deny the usage of all other regions than the ones that are needed.
- Deny the usage of Root credentials.
- Deny modifications to CloudTrail configurations.
- Deny modifications to billing configurations.
- Deny modifications to admin level IAM roles.

4.5. Management Architecture Policies

SCOPE	POLICY
Organization	Control Tower should always be used to manage organizations, unless an equivalent Landing Zone has been created using other methods.
Organization	Production and non-production workloads should be separated with accounts.
Organization	Management account should only be used for tasks that require the management account. Avoid mixing workload and management accounts.
Organization	A tagging strategy should be planned and enforced across all organization resources.
Accounts	All accounts should belong to an organization and OU, excluding the Root account.
Accounts	Accounts and owner email addresses should follow a pre-determined naming convention.
Accounts	Account contact phone number should be kept updated.
Root credentials	Root user credentials for all accounts should be stored in a secure manner.
Root credentials	The process for using the Root user should be documented.
Root credentials	MFA should be enabled for the Root user. It can be either a physical or virtual token. If physical, a virtual backup should be made.
SCPs	Regions to be used should be pre-determined and using any other regions should be denied with SCPs.

4.6. Account Structures

4.6.1. Landing Zones

A landing zone provides essential services for all the AWS accounts in an AWS organization. A basic Control Tower based landing zone provides 3 account setup out-of-the-box:

- Management account
- Audit account
- Log archive account

The management account is the most critical account of the whole AWS organization. Excessive permissions on the management account can be used to compromise the whole AWS organization and all the accounts in it. The management account works as the consolidated billing account for the organization and can be used to create and manage accounts within the AWS organization using the Control Tower service. By default, the management account also contains the AWS IAM Identity Center, which can provide users access to all other AWS accounts within the AWS organization. The IAM Identity Center can also be delegated to a separate AWS account within the

AWS organization, so that personnel responsible of AWS user management don't have to be given permissions on the management account.

Audit account (also known as "Security Tooling" account) works as a dedicated account for building security / compliance visibility and monitoring for all AWS accounts in the AWS organization.

Log Archive account is used as the centralized CloudTrail (AWS level audit logs) storage for all accounts in the AWS organization. It's important to store the CloudTrail logs in a centralized way so that operators on the workload AWS accounts cannot erase the audit logs to hide their tracks.

The basic 3 account landing zone setup can be extended after the initial Control Tower setup to serve the needs of the AWS organization as required.

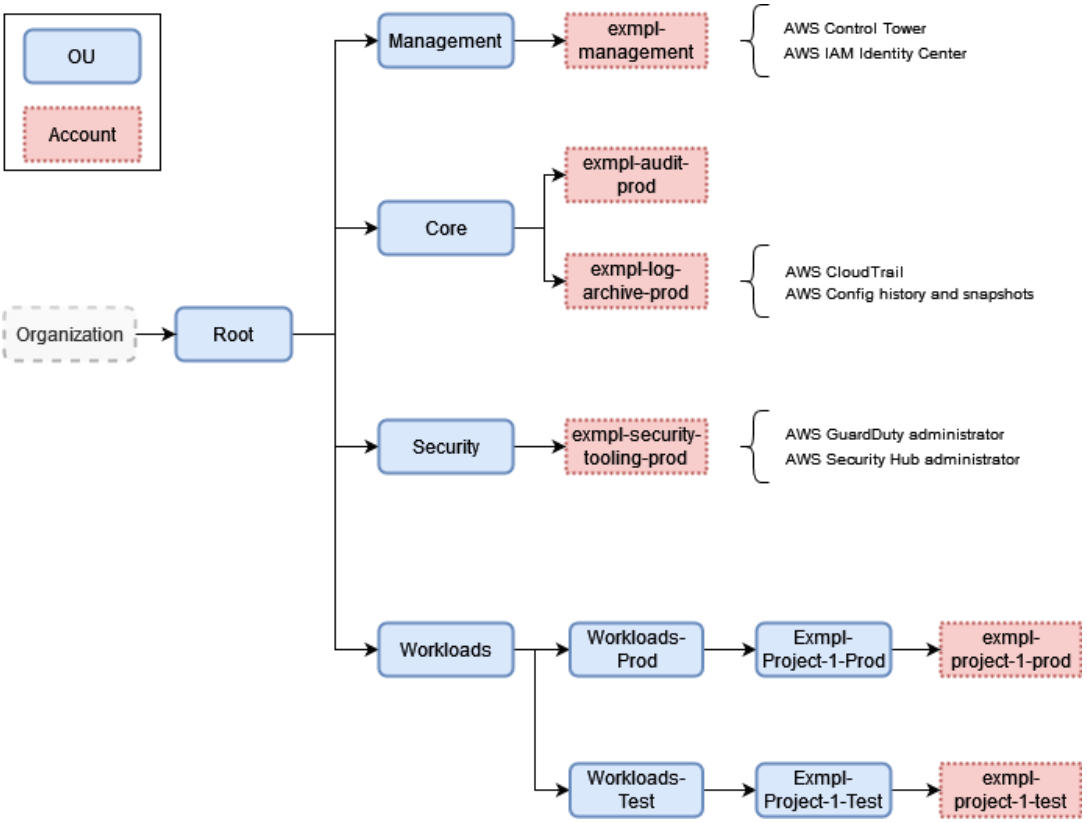
4.6.2. Multi-account environments

Accounts are the abstraction level that should be used whenever possible to isolate workloads and other resources. It is highly recommended to always utilize a multi-account strategy, which should be based on function, business purpose and/or ownership. The account architecture that's best suited for each scenario should always be determined case by case and it should be a conscious decision based on guidelines and best practices.

There are rare cases when using a single account is acceptable, but these should only be used for temporary purposes such as POC's and are never sufficient for an actual production workload. In these cases, it is worth noting that an existing account can be imported into an organization later should the POC go into further development or production use.

The multi-account strategies outlined in this document are largely based on a Control Tower based landing zone setup. While using Control Tower is the recommended way, similar results can be achieved without it.

All the examples below are designed on the assumption that the owner of the organization is an institution. In such case, no institution level OU is needed. However, if multiple institutions reside in the same organization, then a parent OU for each institution is recommended for workload OUs.

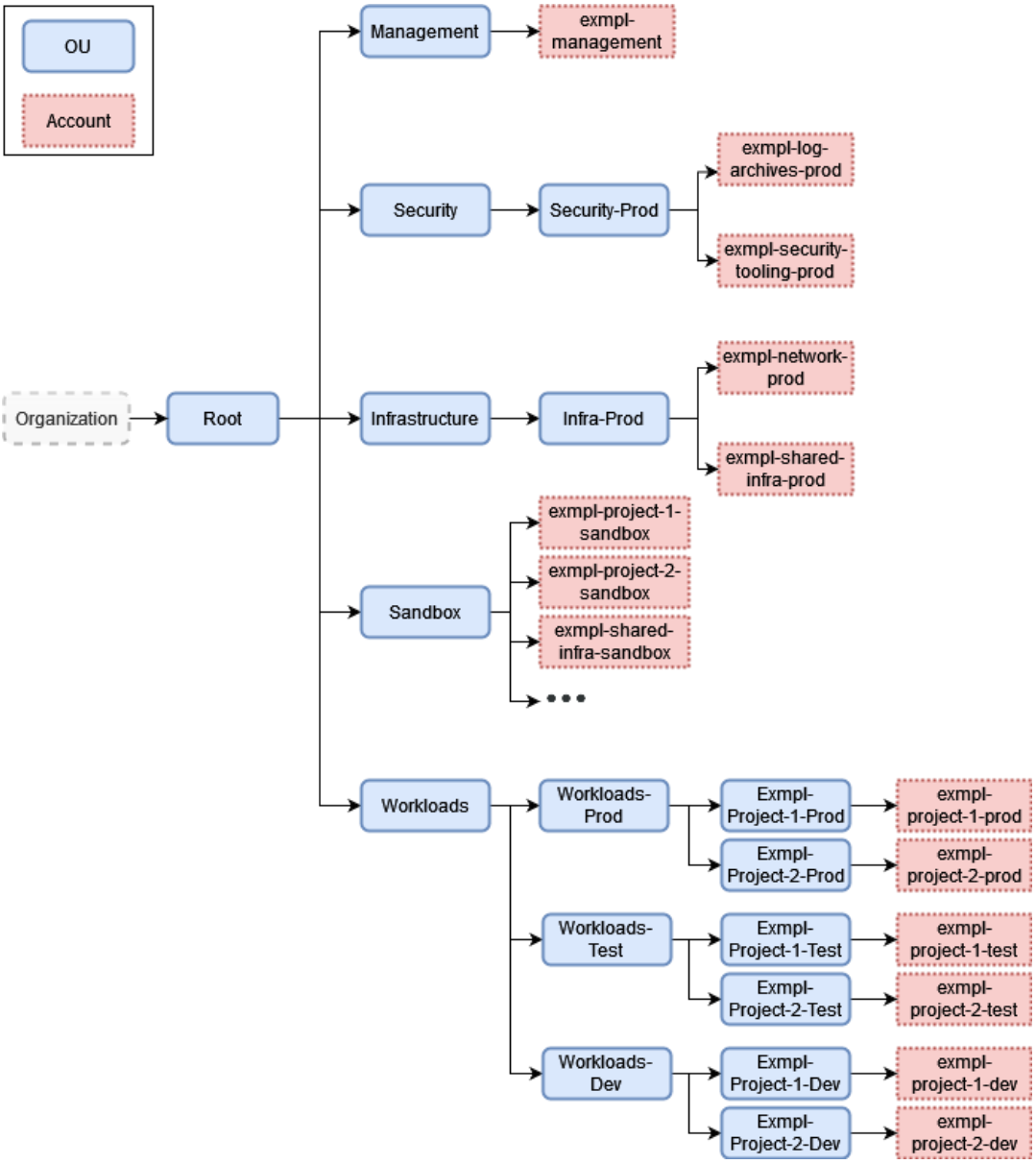


A basic Control Tower based OU and account structure (orra code "exmpl")

In the above example, a Security OU and account as well as Audit and Log archive accounts are created by Control Tower by default. Control Tower itself and IAM Identity Center are managed in the management account, CloudTrail is isolated into the Audit and Log archive accounts, and shared security services are placed in the Security Tooling account. This represents a basic Control Tower account architecture.

Workloads are and should always be in their own respective accounts. In this example, they have been divided into testing and production accounts, which each are located under their own respective OUs. Chapter *4.3.1 Workload OUs* contains more details on how to structure workload OUs and accounts.

An expanded version of the above architecture can be seen below.



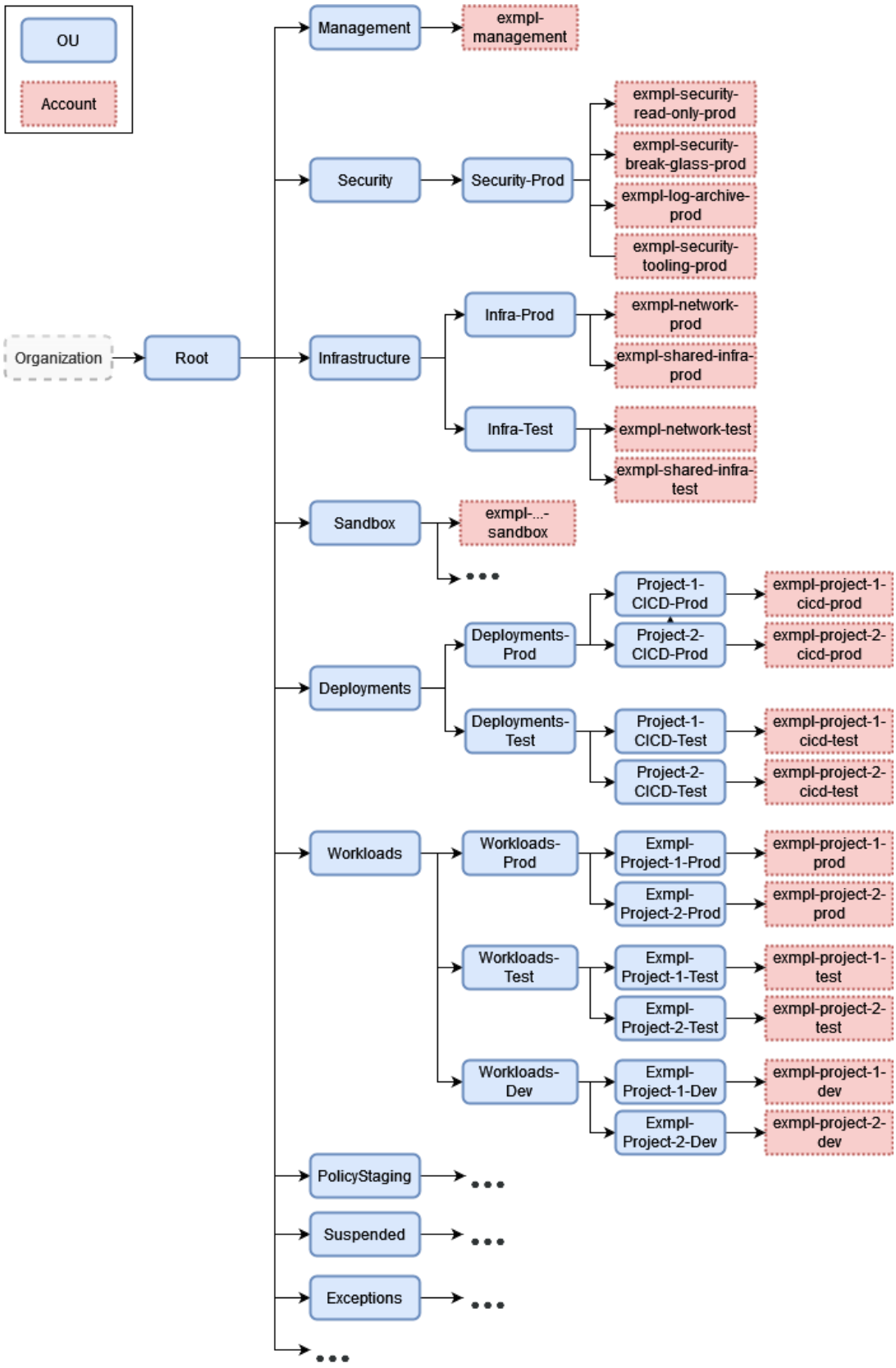
Expanded Control Tower based OU and account structure (orra code "exmpl")

In the above example, an infrastructure OU, network account and shared infrastructure account have been added. The network account could contain, for example, AWS Transit Gateway, AWS Direct Connect or other shared networking services.

Workloads have been divided into their respective OU's and accounts.

Additionally, an OU for sandbox accounts have been added, which contains accounts for quick testing that can be wiped clean often.

An even more expanded version of the above architecture can be seen below.



Further expanded Control Tower based OU and account structure (orra code "exmpl")

Many things have been added in the above example. For one, shared infra services have been divided into production and testing environments so that they can be developed in a more efficient and secure manner. Additionally, a

Deployments OU has been added, which contains the production and test accounts for the CI/CD services of each project.

Several other OU's have also been added, such as PolicyStaging OU for testing SCP's, Suspended OU for accounts that have been put out of use but haven't been deleted yet and Exceptions OU for accounts that don't fall under the category of any other OU.

It's important to remember that the above architectures serve as examples and major emphasis should be placed on what is really needed.

5. Usage policies

It is important to have and follow a set of policies regarding the usage of Amazon Web Services. These policies have effect on – for example – manageability and costs. This section describes the usage policies set by Iceland.

5.1. Cost management

In the **Strategic Cloud Policies**, the following statement is made:

- *Strategic cloud policy 4.5: Make purchasing and ordering of general cloud platform services (capacity, technical services) easy and quick. Utilize cloud elasticity* ***to optimize the entire life cycle costs of the service****.*
- *Strategic cloud policy 4.6:* ***Costs of common governmental cloud services are allocated according to utilization.***
- *Strategic cloud policy 5.4: Utilize the elasticity of cloud services using iterative and experimentative development model. Publish and test often,* ***start small and expand according to growing needs****.*
- *Strategic cloud policy 6.3:* ***Measure and manage*** *your cloud suitability, continuity, security and* ***costs on a day-to-day basis****.*

The policies for Cost Management are stated in the following table.

SCOPE	POLICY
Ownership	The Resource owners and ultimately the Account owner is responsible of cost management procedures in the environment
Tool for cost management	AWS Billing and Cost Management - service is used for cost management activities. Account and/or Organization owners must have rights to see costs related to their Accounts. Additional external cost management (FinOps) tools can be used if needed.
Period of use	Resources must be kept running only when they are needed. Automated startup/shutdown processes are recommended for test and development workloads.
Review of used Resources	Resources must be reviewed regularly, in minimum once per three months. Unneeded resources must be removed.
Resource sizing	Begin with a minimum setup required for the use. Compare different kinds of resource types and product prices prior to implementation.
Budgets	AWS Budgets should be utilized to automatically monitor and predict account costs. Organization owners are responsible for setting up budgets and cost alerting mechanisms.
Reservations	Use AWS Savings Plans and Reserved Instances as applicable. For example, with AWS Savings Plans it is possible to achieve savings of up to 72% compared to the On-Demand model.

5.2. Region

Choice of Region when using AWS services is important due to many reasons. The factors to consider are the following:

1. Latency: using a Region that's closer to end-user will have a substantial impact on user experience due to lower network latency.
2. High Availability: having business-critical resources in multiple Regions offers better SLA for the application.
3. Data sovereignty: keeping data inside EU/ETA borders is important for non-public data.
4. Cost optimization: AWS resources and services have different prices across different Regions. Also, if you have interconnected resources in multiple Regions, traffic between them will cost more than if they are in a single Region.
5. Service and features: new AWS services and features are deployed to lesser used Regions gradually. Not all of them may yet contain all the required services.

It is worth noting that some AWS services are non-regional, and the above criteria do not apply to them.

The policies for used Regions are stated in the following table.

SCOPE	POLICY
Region	Primary Region to be used: Ireland (eu-west-1) Secondary Region for HA purposes: Frankfurt (eu-central-1), Stockholm (eu-north-1)

5.3. Naming conventions

Naming conventions offer administrative users' easy access to such information as:

- Owning Institution
- Resource type
- Associated application
- Environment (prod/test/dev)

This in turn leads to less human errors, when the administrative users can see more easily which resources they are managing.

The generic naming conventions are stated in the following table.

COMPONENT	NAMING CONVENTION
Institution	Abbreviation (Orrakóði) of the institution owning the Subscription, Resource Group or Resource. Up to 5 characters long. Examples: *lsh* = National Hospital *fjr* = Ministry of Financial and Economical affairs Full list of Orra-codes shall be made available.
Application or service name	Name of the application, workload or service. Examples: *sharepoint* *analytics*
Environment	Environment abbreviation: *prod* = production *dev* = development *test* = test *qa* = quality assurance
Running number (optional)	For example 001, 002 or 003.

Note! Names are given in lowercase letters and are separated by hyphens (-).

So, an EC2 virtual machine owned by National Hospital, running a productional MySQL instance, may have the name of:

lsh-mysql-prod-001

5.4. Tagging

Tagging is a mechanism, where resources are labelled with metadata (key – value pair). Tagging can be used for:

- Operations management purposes – tagging SLA, business criticality etc.
- Resource management – ownerships, environments, applications etc.
- Cost management – for example cost allocation
- Classification of data – what confidentiality level is related to the workloads
- Measuring compliancy with the policies set by the organization
- Automation, such as start/stop procedures.

Tags are, in essence, there to help govern and manage the environment. They also help with filtering views – for example in Cost Explorer– enabling persons to see only the relevant workloads.

The policies for Tagging are stated in the following table.

TAG NAME	FORMAT AND/OR *EXAMPLE VALUE*	DESCRIPTION	NECESSITY
oApplicationName	Text *sharepoint*	Application name	Mandatory
isgov-Env	*prod* *dev* *qa* *test*	Environment information	Mandatory
isgov-DataClassification	*Open* *Protected* *Specially protected* *Restricted*	Data classification	Mandatory
isgov-OrID	Text Examoke: FJR, RSAK, DMR	Organisation ID	Mandatory
isgov-Owner	IAM Identity Center group (or Azure Active Directory group) that owns the resource	Resource or application owner	Mandatory
isgov-ReviewedDate	Text in date format *2023-10-20*	Date, which tells on when the last review has been done for the resource	Mandatory
isgov-Criticality	Business Critical Critical Non-Critical	Business criticality of the Resource or application	Optional

TAG NAME	FORMAT AND/OR EXAMPLE VALUE	DESCRIPTION	NECESSITY
isgov-SLA	*Gold* *Silver* *Bronze*	Service Level Agreement for the workload / application	Optional
isgov-CostCenter	Number (Viðfang) as specified in the Orri ERP system.	Cost center / project responsible of the costs related to the Resource or application	Optional
isgov-OpsTeam	Text Example: operations@island.is	Team or partner operating the Resource or application	Optional
isgov-EndDate	Text in date format 2024-01-22	The assumed end date for the use of the Resource.	Optional
isgov-Requester	E-mail Jon.jonsson@Stofnun.is	The person who has requested the Resource	Optional

Note! The format of the tag and the value is important. Use the precise letter case (upper or lower) format as given above. The “isgov” - prefix is used for easy identification of tags within the scope of this framework.

5.5. Resource removals

Removing unneeded resources must be ensured to enable cost savings. However, continuity must also be ensured – when resources are removed, accidents may occur. AWS offers certain protections – which are defined in this policy area – for these kinds of situations.

The policies related to Resource Removals are stated in the following table.

SCOPE	POLICY
Generic principle	When Resources are removed, all the related Resources – that are no longer being used – must be removed as well. Examples of such Resources include Elastic IP’s and storage services.
Deletion protection	Deletion of vital resources should be prohibited. More of deletion protection in chapter 5.7.4 *Deletion protection*.
Removal of productional environments	Prior to removing productional Resources, the Resources must be shut down, stopped or otherwise sealed from the environment for seven (7) days. This ensures that the Resources are not being used for some other use. Use *EndDate* (see: *5.4* Tagging*)* to tag the period, when the Resource may be removed.
Removal of snapshots and backups	Special care must be taken when removing EBS/RDS snapshots and backups.

5.6. IAM practices

Everything in AWS is governed by AWS IAM Identity Center and AWS IAM.

AWS IAM Identity Center is used to centrally manage users and their access levels to AWS accounts within the AWS organization. You can also connect **one** external identity provider to AWS IAM Identity Center (such as Azure Active Directory) and federate the users and groups into AWS IAM Identity Center. AWS IAM Identity Center uses Permission Sets to define what permission users and groups have on the target AWS accounts. AWS IAM Identity Center is located on the AWS organization's management account by default, but the administration of it can be delegated to a separate AWS account. This helps segregating the responsibilities of the AWS organization's user management and taking care of the organization management account itself.

AWS IAM is used to mainly manage service-to-service access within a single AWS account or across trusted AWS Accounts. Access management is done using IAM roles and IAM policies. Roles are entities that can be "assumed" by a user or a service (AWS service or external service, such as Github Actions) and IAM policies determine what AWS actions are allowed by the role. AWS IAM can also be used to create users and groups to the AWS account directly, but this practise should be avoided, and the users and groups should be centrally managed using AWS IAM Identity Center. Preventing the creation of the AWS IAM users (with for example Security Control Policies) should be considered.

The policies related to IAM practices are stated in the following tables. There are separate tables for users federated from Azure Active Directory and for using IAM Identity Center in stand-alone – mode.

Using IAM Identity Center in stand-alone mode

SCOPE	POLICY
Granting Rights	Rights are granted to groups, not individual users
Granting Rights	There should be at least two (2), but preferably three (3) users with administrator rights to AWS organization's management account.
Granting Rights	Least privilege must be applied. Grant only the rights needed.
Remote management of Virtual Machines	AWS System manager's session manager should be used for remote management of Virtual Machines. Session manager can be used with a bastion host for tunneling connections to non-virtual machine resources, such as AWS managed databases.

Using IAM Identity Center with users federated from Azure Active Directory

SCOPE	POLICY
Administrative accounts	Administrative accounts are separated from “office accounts”
Location of Administrative accounts and groups	Administrative accounts with permissions to project AWS accounts can be synced accounts (from on-premises Active Directory). Administrative groups with access to AWS organization management account are cloud-only.
Granting Rights	Rights are granted to groups.
Granting Rights	There should be at least two (2), but preferably three (3) users with administrator rights to the AWS organization’s management account. Federated Groups are not used for AWS organization management account administrators, so that a user with group management rights in Azure AD cannot grant themselves rights for the AWS organization management account.
Granting Rights	Least privilege must be applied. Grant only the rights needed.
Remote management of Virtual Machines	Use AWS System manager’s session manager for remote management of Virtual Machines. Session manager can be used with a bastion host for tunneling connections to non-virtual machine resources, such as AWS managed databases.

5.7. Continuity

The **Strategic Cloud Policies** state the following:

- *Strategic cloud policy 4.3: The **continuity and availability requirements** of the processes are achieved by **developing a cloud native high availability architecture together with SLAs in the cloud contracts**.*
- *Strategic cloud policy 5.3: **The data in cloud services and platforms must always be easily transferable to other platforms or systems**. **Continuity must be ensured in all cases based on the business continuity needs**.*
- *Strategic cloud policy 6.2: **Constantly monitor your services**. Create technical capabilities to provide **real-time insight on your environment’s health**.*

While continuity is a larger subject altogether – ranging from day-to-day operations into business continuity in disaster scenarios – in the Governance Framework we cover the following four subjects:

1. Monitoring: how applications, services and resources are being monitored
2. Deletion protection: how accidental or malicious removal or alteration of resources is prevented

3. High availability: how redundancy is considered when building up solutions
4. Backup and recovery: how to ensure recovery of data and services in case of faults or accidental/malicious removals.

The policies related to Continuity are stated in the following chapters.

5.7.1. Business continuity in AWS

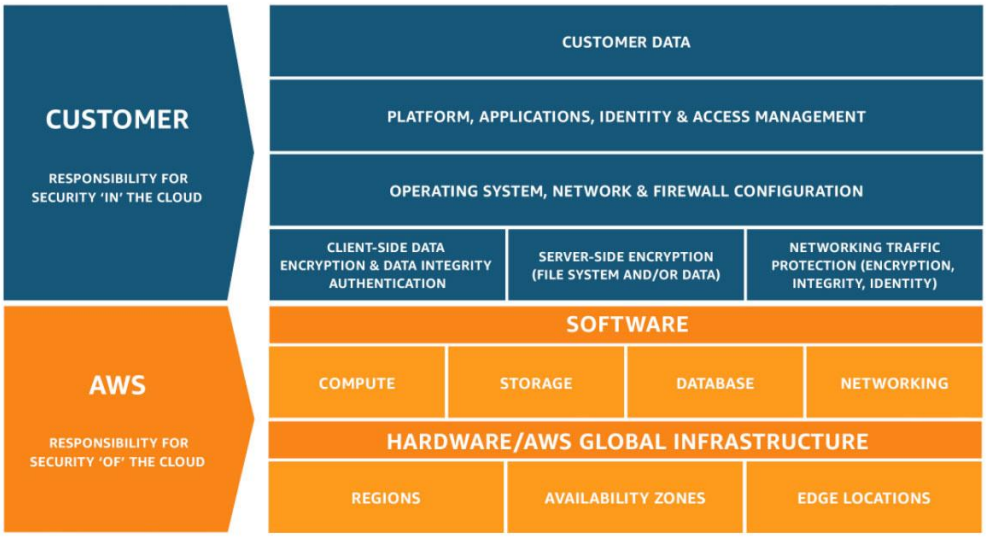
Business continuity, i.e. the continuity of the applications and solutions created for "business purposes" (in this case an example could be a patient registry used by the National Hospital) typically have recovery time objective (*RTO*) and recovery point objective (*RPO*) requirements. These requirements work as a base for designing continuity solutions, which may include:

- High availability – e.g. clustering the environment, so that it is not dependent on a single point of failure
- Disaster Recovery – both a technical solution to return the system and its data to use and the surrounding process and guides on how the system is returned to use. Technologies may include such as:
    - Backups – recovering workloads, applications and data from backups
    - Cold site – recovering the system using a secondary site and backups
    - Replication – recovering the system using a passive, nearly identical and up-to-date environment.

> ***RTO** states the timeframe in which the system must be functional. For example, an RTO of 24 hours would state that the system is operational latest 24 hours from failure.*
>
> ***RPO** states the accepted loss of data in time. As an example, RTO of 4 hours would state that losing the last 4 hours of data from failure would be acceptable.*

When it comes to AWS (and cloud services in general), it is vital to understand that the customer is responsible for ensuring the continuity of their business applications and data (i.e. continuity *in* the cloud). In turn, the cloud service provider is responsible for the existence and functionality of the services that they provide (i.e. continuity *of* the cloud). Therefore, it is vital to plan the business continuity of each application and data according to the customer requirements.

AWS illustrates the areas and division of responsibility with their Shared Responsibility Model, pictured below (source: https://aws.amazon.com/compliance/shared-responsibility-model/). Although the model talks about security, for all intents and purposes it applies to continuity as well.



5.7.2. Monitoring

SCOPE	POLICY
Generic policies	All productional environments and their Resources are monitored. Each Institution is accountable for organizing the monitoring capabilities. Organization owners may offer this service to the Institutions in each Organization.
Location of CloudTrail logs	CloudTrail logs for each account in an organization should be stored in a separate AWS account in a centralized manner. For example, when using Control Tower, an account called Log Archive is designed for this purpose.
Access to CloudTrail logs	Access to centralized CloudTrail logs should be limited to a minimum. Alteration or deletion of logs should be denied completely.
Application monitoring	Each system owner is responsible of their applications' monitoring capabilities. Amazon CloudWatch may be utilized for application monitoring when applicable.
Connection to Incident Management process	Connection to Incident Management process and possible ITSM tooling is done on a later phase, not currently available.

5.7.3. Centralized compliance checks

AWS Security Hub's Security Standards can be used to monitor the AWS resource compliance against various security standards, such as:

- AWS Foundational Security Best Practices v1.0.0
- CIS AWS Foundations Benchmark v1.4.0
- NIST Special Publication 800-53 Revision 5

Security Hub's Security Standards provides a centralized view of the compliance status against enabled Security Standards of all accounts in the AWS organization. Individual accounts can also check their own compliance

score from Security Hub. Individual compliance checks can be disabled from the standard if needed.

Suggested Security Hub Security Standards for the AWS organization are AWS Foundational Security Best Practices and CIS AWS Foundations Benchmark.

In addition to Security Hub's Security Standards, AWS Config Conformance Packs can be utilized for compliance checks. AWS Config Conformance packs (or custom conformance packs) can be integrated with Security Hub.

5.7.4. Deletion protection

SCOPE	POLICY
Generic policies	Deletion of production resources must be denied by default on IAM level.
Recycle Bin	Each system owner is responsible of defining Recycle Bin policies for their account. Recycle bin feature helps preventing accidental deletion of EBS volumes and AMI images. Recycle Bin policy can be defined to target AWS Regions or resource tags.
How to lock	Productional Resources are recommended to be locked with the "CanNotDelete" mechanism. "CanNotDelete" can be implemented with resource tags combined with AWS IAM Identity Center Permission Sets. Some AWS services have implemented deletion protection mechanisms of their own; where service specific deletion protection mechanism exists, it can be used. Note that some resources have "Modification requires replacement" on CloudFormation level which will block such resource updates. The "CanNotDelete" - policy is recommended for staging environment for that reason as well when the application goes from active development into maintenance mode.

5.7.5. High Availability

SCOPE	POLICY
Redundancy (multiple availability zones, multiple regions)	Each system owner can decide of the implementation of redundancy according to the business requirements. (see: 5.7.1: To be considered: business continuity in AWS)
Geographical distribution	Each system owner can decide of the implementation of geographical distribution according to business requirements. The location restrictions stated in *5.2* must be considered.

5.7.6. Backup and Recovery

Each system owner is responsible of defining the backup and recovery mechanisms based on RPO/RTO needs and a risk-based evaluation. The following table describes the policies and procedures if there are no specific requirements for the system.

SCOPE	POLICY
Generic policies	The native service AWS Backup is used for backups. Default backup policies (see the next table) are used, but application specific deviations may be defined, based on business requirements.
High impact data or services	For data and services that are of high importance to the society as a whole additional backups shall be used that are isolated from the cloud service provider (AWS) to mitigate systemic risks of data loss and applications an additional 3rd party backup tool must be considered to enable backups to a separate location/environment outside of AWS. Guidance for when this applies can be found in Iceland's Data Classification Policy. When considering data loss both the integrity and availability of the data shall be considered.
Backup lifecycle management	AWS Backup service manages the lifecycle of backups performed by it. Each system owner is responsible of the lifecycle management of manually performed backups.
Tooling	3rd party backup tool is used for the backup of society-critical **Virtual Machines and databases running on virtual machines.** AWS Backup and database level backups is used for other **Virtual Machines and databases running on virtual machines**. Built-in backup functionality is used for **PaaS-databases** (Snapshots on AWS RDS and Amazon Aurora).
Disaster recovery	To prevent potential malicious entity from deleting the backups, the backup destination can be another AWS account or an alternative location. If the backups are copied to another AWS account, consider using a different AWS region for the backup copies for additional redundancy.

Default backup policies are defined in the following table. These may be changed according to business needs.

SCOPE	POLICY
Productional environments	Daily backups: retention time 7 days Weekly backups: retention time 4 weeks Monthly backups: retention time 12 months Application-specific deviations may be defined.
Test and quality assurance environments	Daily backups: retention time 7 days Weekly and monthly backups: no retention requirements Application-specific deviations may be defined.
Development environments	Daily, weekly and monthly backups: no retention requirements Application-specific deviations may be defined.

5.7.7. Data continuity

SCOPE	POLICY
Delete protection	IAM policies can be utilized to prevent users from deleting certain types of resources on production accounts, e.g. deny rds:DeleteDBSnapshot.
Versioning	Versioning can be enabled for S3 buckets if seen fit. System owner is responsible for defining and enabling this feature. Versioning is required for data classification levels 3 and 4 (see: *Data Security Classification of the Icelandic government, 05/2023*)

5.8. Networking

Networks are a major building block in the public cloud as well. Through them you can segment application workloads and add security layers and enable access to end-users.

The policies related to Networking are stated in the following table.

SCOPE	POLICY
Connections to the on-premises networks	Connections to the on-premises networks are done through a VPN connectivity over the Internet when necessary. AWS Direct Connect may also be used when necessary. AWS Direct Connect and site-to-site VPN connectivity should be terminated to a centralized networking hub when possible.
Network topology	Currently, the network architecture is implemented as individual virtual networks. Making sure the network ranges don't overlap with each other or with on-prem is still essential in case hub-and-spoke architecture will be implemented in the future. Target setup per AWS organization would be to have a

	hub-and-spoke network topology, but it is not currently implemented. In the hub-and-spoke topology the virtual networks (vpc) are connected to each other through a centralized virtual network and a router/firewall located there.
Name resolution	Use Route53 for name resolution in AWS. If the AWS environment is connected to on-premises data center and resolution of internal DNS addresses is required, setup Route53 for hybrid DNS.
IP addressing	Organization Operator and the Institutions are responsible of the IP network allocations for each system/need.
Virtual network connectivity	Virtual private networks are connected to each other through VPC Peering when needed, unless a hub-and-spoke topology is in place. If a hub-and-spoke topology is in place, all communications must flow through the hub network.
Use of PaaS services	PaaS services are primarily used through VPC Endpoints.

Policies specific to network security are stated in the following table.

SCOPE	POLICY
Generic policies	All traffic inbound and outbound should be denied by default. Unsecured HTTP must not be used for publishing services externally. All external websites and other HTTP-based communications must use HTTPS.
Use of Network Security Groups	When possible, Security Groups shall be utilized for protection and ease of management of multi-layer application environments (such as application workloads and database environments). Security Groups should be preferred instead of Network Access Control Lists. Security Groups are tied to elastic network interfaces of AWS resources. Security Groups have an allow all outbound traffic rule by default which must be removed. Presence of rule allowing all outbound traffic can be monitored using AWS Config.
Publication of applications	Applications are published by default using services such as Application Load Balancer, AWS CloudFront and AWS API Gateway.
Virtual networks	Create a virtual network (VPC) per AWS account. Use subnets and network security groups to segment applications and workloads from each other in a virtual network. If there is a need to segment on a virtual network level, additional virtual networks can be created on the AWS account.
Use segmentation on	Segment the virtual networks into subnets as needed. As

subnet level	an example, segment shared database resources to a separate group of subnets, applications to their own private subnets and load balancers that need to be reachable from the Internet to the public subnets. Use multiple subnets of the same type and spread them out to different Availability Zones for high availability.
Enable AWS Web Application Firewall	Use this with HTTPS traffic
Enforce HTTPS-only communication	Ensures the user-to-app internal traffic is encrypted.
Use CloudFront, WAF and Shield Advanced for DDoS protection of critical workloads	AWS CloudFront should be used to provide resilience for web applications. Use CloudFront with AWS Web Application Firewall to provide DDoS mitigation capabilities. AWS Shield Advanced should also be considered on the AWS organization level depending on the probability of DDoS attacks and the criticality of availability. 3rd party DDoS protection/CDN can also be considered if CloudFront does not meet the requirements.

5.9. Security

Security is also a wide area, but in this Governance Framework the following six key risk areas addressed by Zero Trust Framework are defined:

1. **Identity:** Automate risk detection and remediation. Secure access to resources with strong authentication
2. **Endpoints:** Defend larger attack surface created by the growing number of endpoints using integrated approach to management
3. **Data:** Classify, label and protect data across cloud and on-premises environments to help prevent inappropriate sharing and reduce insider risks
4. **Apps:** Institutions must find the right balance of providing access while maintaining control to protect critical data accessed via applications and APIs.
5. **Infrastructure:** Protect hybrid infrastructure, including on-premises and cloud environments, with more efficient and automated management
6. **Network:** Reduce perimeter-based security vulnerabilities. Instead of believing everything behind corporate firewall is safe, Zero Trust strategy assumes breaches are inevitable.

The following sections cover the Identity and Infrastructure, while Network security is covered in chapter *5.8*. Endpoints. Data and apps are not part of this Governance Framework.

5.9.1. Identity

Standalone AWS IAM Identity Center Policies

POLICY	DESCRIPTION
Use IAM Identity Center Groups	Use IAM Identity Center Groups to handle permissions if possible.

AWS IAM Identity Center with Azure AD federation Policies

POLICY	DESCRIPTION
Use Azure AD Groups	Use AAD Groups to handle permissions if possible.
Block external guest access	Don't allow invitations to be sent to any domain. For external consultants and partners an identity must be created for the Azure Tenant in question.

Generic Identity related policies

POLICY	DESCRIPTION
Roll out AWS IAM Identity center MFA	Multi-Factor Authentication helps safeguard access to data and applications, providing another layer of security by using a second form of authentication. Organizations must enable multifactor authentication (MFA) for all IAM Identity Center users.
Block legacy authentication	One of the most common attack vectors for malicious actors is to use stolen/replayed credentials against legacy protocols.
Use Permission Sets for user/group Access Management	AWS IAM Identity Center permission sets provide the mechanism for determining what level of access is granted to each user/group. Permission set grants temporary access to the destination account and the session duration can be adjusted in the permission set's settings. Groups (federated or local) should be utilized effectively with permission sets to grant users the desired level of access to AWS accounts and resources in them.

5.9.2. Infrastructure

POLICY	DESCRIPTION
Enable AWS Config	Enable AWS Config for infrastructure compliance monitoring.
Govern how resources are deployed	You can e.g., only grant power user access for CI/CD pipelines.
Application audit logs	Make sure your applications produce meaningful audit logs and the audit logs are stored both on the AWS account that produces them and on a centralized AWS account where users on the producer account cannot

	overwrite them.
Use AWS Config to ensure logs are compliant for every resource	You can monitor resource's compliance to logging policy using AWS Config rules.

5.10. DevOps

The **Strategic Cloud Policies** state the following:

- Strategic cloud policy 5.2: Leverage a wide range of the technical capabilities of the chosen cloud environment. **Use native automation tools and value add services.**
- Strategic cloud policy 5.4: Utilize the elasticity of cloud services **using iterative and experimentative development model**. Publish and test often, start small and expand according to growing needs.
- Strategic cloud policy 6.1: **Fully automate your services**. Leverage automation tools to **scale your services based on demand and automate changes to your environment**.

DevOps enables development, IT operations, quality engineering, and security to coordinate and collaborate to produce better, more reliable products. With DevOps culture along with DevOps practices and tools, teams gain the ability to increase confidence in the applications they build and achieve business goals faster.

The policies related to DevOps are stated in the following table.

POLICY	DESCRIPTION
Limit access to projects and repos	Reduce the risk of leaking sensitive information and deploying insecure code
Approval Gates	You can add members to Approval Gates when needed so their approval is needed when e.g., deploying to production
Secure secrets	Use the combination of CI/CD tool's secret variables, AWS System's Manager Parameter Store and AWS Secrets Manager to secure secrets. AWS System's Manager Parameter Store should be preferred over AWS Secrets Manager unless the secret in question needs to be automatically rotated periodically.
Use code reviewing	Require at least one reviewer outside of the original requester
Use PR's (Pull Request)	Deny that code can be directly merged to production branch
Disallow completion of a PR from the requester	Don't allow the original pull requester to approve their own PR's
Always use different credentials for dev, test and production environ-ments	Make environment specific AWS IAM OIDC roles/ credentials

Use YAML pipelines	Manage pipeline definitions with configuration files. This allows using version control which "UI based pipelines" sometimes don't
Don't store secrets in pipeline variables	Never use hard coded secrets straight from the configuration files or version-controlled code. Use CI/CD's Secret Variables.
Enable Audit logging	If possible, enable logging for the CI/CD

5.10.1. Infrastructure as Code

Infrastructure as code (IaC) is the process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

The policies related to Infrastructure as Code are stated in the following table.

POLICY	DESCRIPTION
Use Infrastructure as Code when possible	Infrastructure resources should be built, maintained and updated using an IaC tool whenever possible. These tools include, but are not limited to, AWS CDK, AWS Cloudformation and Terraform by HashiCorp.
Resource state should be centralized	IaC tools function based on resource state. These state files should always be stored in a central manner, for example in AWS S3.
Use a version control system	IaC templates, configurations and files should always be stored in a version control system such as Git.

6. Organizational policies

There are different kind of ways to achieve AWS organization wide governance strategies, such as:

- Baseline recommendations
- Policy compliance monitoring
- (AWS) Policy enforcement
- Cross-Organization enforcement
- Automated enforcement.

The most concrete way to have Organizational policies in AWS is by using Service Control Policies (SCP), which are applied at the Organization level. See: Service Control Policies.

It is to be noted and understood, that the purpose of a policy enforcement is not to make the creation of new services hard, but to ensure the compliancy of the environment.

The following table describes the minimum Service Control Policies to be set in each AWS Organization. Although not all listed policies are required, the

nonrequired ones serve as a recommendation for policies to consider. Each Organization Operator and individual Institution may add policies according to their needs.

POLICY	DESCRIPTION	Required
Allowed locations	Specify the regions where resources can be deployed (see: chapter *5.2*). Use to enforce your geo-compliance requirements. Excludes resources that use the 'global' region.	Yes
Prevent leaving the organization	Prevent member accounts from leaving the organization. This is done to prevent going around SCPs.	Yes
Root user usage	Deny service access for root user or block using the root user completely. Other security measures for securing Root credentials should still be followed (see: Root Credentials).	Yes
Modifying Account and Billing settings	Prevent users or roles from modifying the account and billing settings, either as an API command or through the console. Additional policies may apply from the capacity reseller if applicable.	Yes
Escalation of privileges	Prevent escalation of privileges. Privilege escalation refers to the ability of a bad actor to use stealthy permissions to elevate permission levels and compromise security. To prevent this, users should be denied from using administrative IAM actions. Administrative actions should be restricted to delegated IAM admins	Yes
Allowed resource types	Specify the resource types that can be deployed in the organization.	No
Not allowed resource types	Restrict which resource types can be deployed across the organization. Limiting resource types can reduce the complexity and attack surface of the environment while also helping to manage costs. Compliance results are only shown for non-compliant resources.	Yes
Allowed services for workload OU's	Allow only approved services for the workloads OU's. New services are rarely introduced in production environments, so only the known ones should be allowed.	No
MFA for resource deletion in production OU's	Require MFA for resource deletion in Production OU's. This should be done to protect accidental or malicious deletion of production resources.	No
S3 object encryption in production OU's	Ensure that S3 objects are encrypted when uploaded to buckets in production OU's.	No
Adding public accessibility to VPC's	Deny the ability to make a VPC accessible from the Internet that isn't already	No
Deleting VPC flow logs	Prevent users from deleting Amazon VPC flow logs.	Yes
Disabling CloudTrail	Prevent CloudTrail logs from being disabled.	Yes
Disabling GuardDuty	Prevent GuardDuty from being disabled or disrupted.	No
Disabling CloudWatch	Prevent CloudWatch from being disabled and CloudWatch Event collection from being disrupted.	No
Disabling AWS Config	Prevent AWS Config from being disabled and the rules of AWS Config from being changed.	No