AWS Prescriptive Guidance

Setting up a secure and scalable multi-account AWS environment

AWS Prescriptive Guidance: Setting up a secure and scalable multi-account AWS environment

Table of Contents

Setting up a secure and scalable multi-account AWS environment

Nivas Durairaj, Senior Business Development Manager, AWS Control Tower

March 2020

Organizations have to balance their builders' needs to stay agile while they provide governance at scale. Establishing the foundational standards gives you the ability to enable, provision, and operate your environment for both business agility and governance at scale.

A successful cloud adoption starts with a secure cloud-based environment that includes:

- An AWS environment with a multi-account architecture
- An initial security baseline
- Identity and access management
- Governance
- Data security
- Network design
- Logging

We refer to an environment that has these features as a *landing zone*. This guide helps you set up a secure and scalable landing zone that can support a production implementation for your migration.

What is a landing zone?

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. Building a landing zone involves technical and business decisions to be made across account structure, networking, security, and access management in accordance with your organization's growth and business goals for the future.

When you start to use AWS at scale, you can look to AWS for prescriptive guidance and an approach for establishing your environment. AWS best practices in this area center around the need to isolate resources and workloads into multiple AWS accounts (resource containers) for isolation and scope of impact reductions. The next section explains why you want to use multiple accounts.

The multi-account framework

While there is no one-size-fits-all answer for how many AWS accounts you should have, we recommend that you create more than one AWS account. Multiple accounts provide the highest level of resource and security isolation. Consider creating additional AWS accounts if you answer yes to any of the following questions:

- Does your business require administrative isolation between workloads?
- Does your business require limited visibility and discoverability of workloads?
- Does your business require isolation to minimize the blast radius?
- Does your business require strong isolation of recovery and/or auditing data?

Why one account isn't enough

Further reasons why a single account might not be enough:

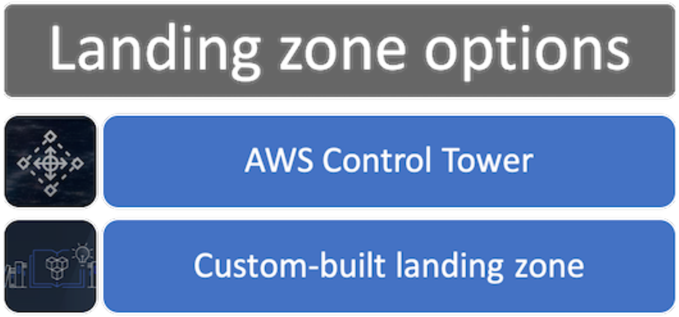
- **Security controls** – Different applications might have different security profiles, requiring different control policies and mechanisms around them. It's easier to talk to an auditor and point to a single account hosting the Payment Card Industry (PCI) workload.
- **Isolation** – An account is a unit of security protection. Potential risks and security threats should be contained within an account without affecting others. There could be different security needs that require you to isolate one account from one another, whether due to multiple teams or a different security profile.
- **Data isolation** – Isolating data stores to an account limits the number of people that can access and manage that data store. This contains exposure to highly private data and helps with General Data Protection Regulation (GDPR) compliance.
- **Many teams** – Different teams have their different responsibilities and resource needs. They should not over-step one another in the same account.

- **Business process** – Different business units or products might have different purposes and processes. You should establish different accounts to serve business-specific needs.
- **Billing** – An account is the only true way to separate items at a billing level, including things like transfer charges. Multiple accounts help separate items at a billing level across business units, functional teams, or individual users.
- **Limit allocation** – Limits are per account. Separating workloads into different accounts prevents them from consuming limits or potentially overprovisioning resources and then preventing other applications from working as intended.

Building a landing zone

You have a few options for creating your landing zone on AWS. You can choose a managed service to orchestrate your environment or work with a partner to build your own. AWS offers AWS Control Tower, a managed service. We recommend new customers start off with AWS Control Tower. However, it is important to understand the differences and capabilities of each approach so you can make the best decision for your organization.

Options for landing zones on AWS:

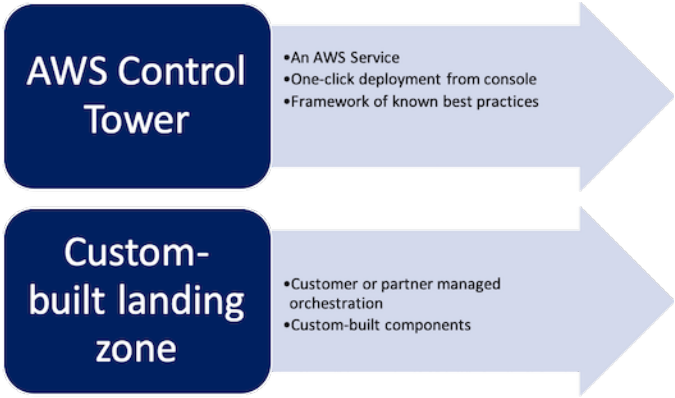


Landing zone options

AWS Control Tower

Custom-built landing zone

Delivery mechanism:



AWS Control Tower

- An AWS Service
- One-click deployment from console
- Framework of known best practices

Custom-built landing zone

- Customer or partner managed orchestration
- Custom-built components

Benefits and trade-offs for each approach:

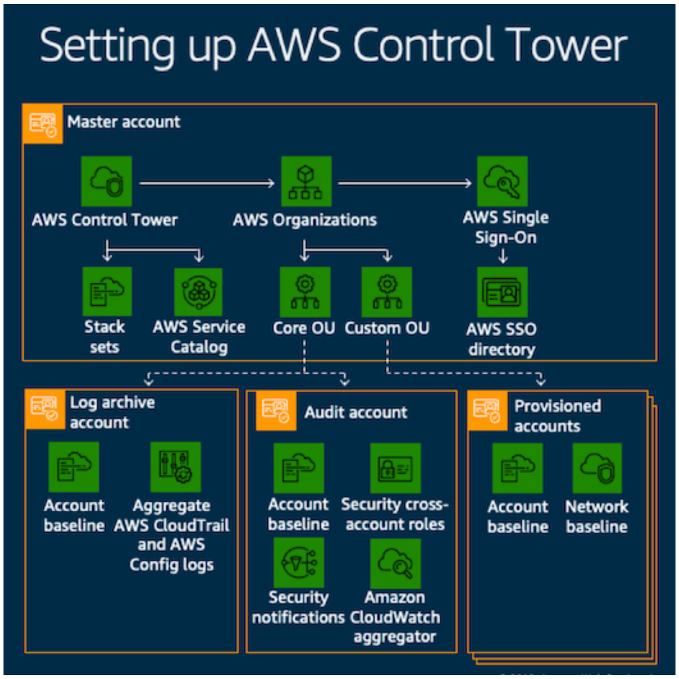
Solution	Benefits	Trade-offs
AWS Control Tower	• Fully managed service • AWS-provided guardrails and compliance policies applied by default • Central dashboard for monitoring and compliance status • Account factory for provisioning new accounts	• Extensibility and customization are provided by the Customizations for AWS Control Tower solution. • AWS Control Tower is supported in the AWS Regions shown in the AWS Regional services list.

Solution	Benefits	Trade-offs
AWS Organizations with a customer or partner built custom solution	Custom-built solution	• Customer or partner owns all development and coding. • Customer or partner is responsible for integration and implementation.

All multi-account environment offerings are powered by AWS Organizations. AWS Organizations provides the underlying infrastructure and capabilities for you to build and manage your AWS environment. With AWS Organizations, you can take the multi-account strategy guidance provided by AWS and customize your environment yourself to best fit your business needs. If you are an existing customer and you're happy with your current AWS Organizations implementation, you should continue to operate your current AWS environment.

AWS Control Tower

AWS Control Tower runs as an AWS managed service. When you're looking for a pre-packaged environment solution out of the box, you can use AWS Control Tower for prescriptive guidance and a fully managed environment. The service sets up a landing zone based on multi-account best practices, centralizes identity and access management, and establishes pre-configured governance rules for security and compliance.



AWS Control Tower automates the setup of a new landing zone using best practices, blueprints for identity, federated access, and account structure. Some of the blueprints implemented on AWS Control Tower include:

- A multi-account environment using AWS Organizations
- Identity management using AWS Single Sign-On (AWS SSO) default directory
- Centralized logging from AWS CloudTrail, and AWS Config stored in Amazon Simple Storage Service (Amazon S3)

- Cross-account security audits using AWS Identity and Access Management (IAM) and AWS SSO

Guardrails are high-level rules that provide ongoing governance for your overall AWS environment. Guardrails can be both preventive or detective. Preventive guardrails are implemented using service control policies (SCPs), which are a part of AWS Organizations. Detective guardrails are implemented using AWS Config Rules and AWS Lambda functions. Examples of AWS Control Tower guardrails include:

- Disallow creation of access keys for the root user
- Disallow internet connection through RDP
- Disallow public write access to S3 buckets
- Disallow Amazon Elastic Block Store (Amazon EBS) volumes that are unattached to an Amazon Elastic Compute Cloud (Amazon EC2) instance

> **Note**
> AWS Control Tower is a starting point for a landing zone. You need to determine your strategy for networking, access management, and security based on your unique requirements as you build out your landing zone.

Custom-built landing zone

You can choose to build your own customized landing zone solution. In this case, you have to implement the baseline environment to get started with identity and access management, governance, data security, network design, and logging. We recommend this approach if you want to build all of your environment components from scratch, or if you have requirements that only a custom solution can support. You must have enough expertise in AWS to manage, upgrade, maintain, and operate the solution once it's deployed.

However, before you move forward with a customized landing zone design, we recommend that you consider AWS Control Tower first. AWS Control Tower has been customized and used by many customers across industries to successfully deploy workloads on AWS. If AWS Control Tower does not meet your needs for customization, try AWS Landing Zone. This is a landing zone implementation based on AWS CloudFormation.

Recommended approach

We recommend that all new landing zones start with AWS Control Tower. AWS Control Tower helps you build out an initial prescriptive landing zone configuration, use out-of-the-box guardrails and blueprints, and create new accounts using AWS Control Tower account factory.

If you require custom guardrails and blueprints, see Customizations for AWS Control Tower for customizing your AWS Control Tower landing zone. This reference implementation integrates with AWS Control Tower lifecycle events and notifications feature to push landing zone customizations in response to applicable AWS Control Tower lifecycle events.

If you are an existing AWS Control Tower customer, you have both native AWS Control Tower lifecycle events and the reference implementation for customization available to support your customization needs. All you need to do is deploy the reference implementation's AWS CloudFormation template into your existing AWS Control Tower account.

Next steps

You should now have a better understanding of landing zones on AWS. If you have more questions or would like to get started on building your landing zone, please get in touch with your AWS account team for assistance.

For additional review and more information, please look at some of the documents and videos linked in the following sections.

Videos

- Architecting security and governance across your landing zone
- What is AWS Control Tower?

Documents

- AWS Control Tower
- AWS Control Tower Guardrails
- Customizations for AWS Control Tower
- AWS Organizations
- AWS Organizations Service Control Policies
- Account Factory
- AWS CloudFormation
- AWS Service Catalog

AWS Prescriptive Guidance glossary

Migration terms (p. 8) | Modernization terms (p. 11)

Migration terms

The following are commonly used terms in migration-related strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. This migration scenario is specific to VMware Cloud on AWS, which supports virtual machine (VM) compatibility and workload portability between your on-premises environment and AWS. You can use the VMware Cloud Foundation technologies from your on-premises data centers when you migrate your infrastructure to VMware Cloud on AWS. Example: Relocate the hypervisor hosting your Oracle database to VMware Cloud on AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there’s no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

configuration management database (CMDB)

A database that contains information about a company's hardware and software products, configurations, and inter-dependencies. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program execution guide.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogenous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the CloudEndure Migration Factory guide in this content set.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines and assigns roles and responsibilities in a project. For example, you can create a RACI to define security control ownership or to identify roles and responsibilities for specific tasks in a migration project.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

Modernization terms

The following are commonly used terms in modernization-related strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilites. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization’s applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application’s features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

polyglot persistence

Independently choosing a microservice’s data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization’s capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development. For more information, see the Two-pizza team section of the Introduction to DevOps on AWS whitepaper.

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

update-history-change	update-history-description	update-history-date
Initial publication (p. 13)	—	March 16, 2020