Plan your deployment

This section describes the Regions, cost, security, and other considerations prior to deploying the solution.

Prerequisite Skills and Specialised Knowledge

Overview

This solution requires foundational knowledge of AWS and specific AWS services. The level of expertise needed depends on the user's role and responsibilities within the deployment.

General Requirements (All Users)

AWS Fundamentals

Users deploying or utilising this solution should have:

- Basic familiarity with AWS services and the AWS Management Console
- Understanding of AWS accounts and regions
- Knowledge of IAM basics and user permissions

Administrator Requirements (Installation and Setup)

Administrators responsible for deploying and configuring this solution require specialized knowledge in the following AWS services:

AWS Organizations

- Understanding of organizational structure and how to manage multiple AWS accounts
- Ability to navigate the Organizations console
- Knowledge of service control policies (SCPs) and their impact on deployments

AWS Identity Center (formerly AWS SSO)

- Configuration and management of Identity Center
- Creating and assigning permission sets
- Managing user and group access across AWS accounts
- Understanding federated access patterns

AWS CloudShell

- Launching and using CloudShell from the AWS Management Console
- Executing CLI commands and scripts within the CloudShell environment
- Basic troubleshooting of CloudShell connectivity and permissions
- Familiarity with AWS CLI commands for automating deployment tasks

CloudWatch

All users analyzing solution outputs should be comfortable with:

- Navigating CloudWatch dashboards and log groups
- Viewing and filtering CloudWatch Logs
- Understanding basic log analysis and interpreting log messages
- Creating simple CloudWatch queries and metrics

Summary

For End Users: AWS fundamentals
For Administrators: AWS fundamentals + Organizations + Identity Center + CloudShell + AWS CLI basics

Installation Prerequisites

Before installing Sandbox Studio, it is important to confirm that the required prerequisites are in place. Most enterprise organisations that already run a multi-account AWS environment will typically have these prerequisites met. However, it is still essential to verify them before starting the installation to avoid any delays or configuration issues later in the process.

1. AWS Organisations

Ensure you have enabled AWS Organisations in your AWS environment before you deploy Sandbox Studio.

> “ AWS Organisations helps you centrally manage and govern your environment as you grow and scale your AWS resources. [1]

> **Why Sandbox Studio needs it**: Sandbox Studio creates and manages sandbox accounts dynamically. AWS Organisations provides the framework to programmatically create new accounts, apply consistent policies, and maintain governance across all sandbox environments.

Please refer to this link to learn how to use AWS Organisations:

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_tutorials_basic.html

---

2. Service Control Policies (SCPs)

Ensure you have enabled Service Control Polices within your AWS Organisation.

> “ Service control policies (SCPs) are a type of organisation policy that you can use to manage permissions in your organisation. SCPs offer central control over the maximum available permissions for the IAM users and IAM roles in your organisation. SCPs help you to ensure your accounts stay within your organisation’s access control guidelines. SCPs are available only in an organisation that has all features enabled. SCPs aren't available if your organisation has enabled only the consolidated billing features. For instructions on enabling SCPs, see Enabling a policy type. [1]

>

> **Why Sandbox Studio needs it**: SCPs allow setting up guardrails and security boundaries for sandbox accounts, preventing users from accessing restricted services or regions and maintain a safe experimentation environment.

Refer to this page to learn how to enable SCPs:

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies.html

---

3. AWS IAM Identity Center

Ensure you have enabled AWS IAM Identity Center in your AWS Organisation.

Note: Sandbox Studio requires an Organization instance of IAM Identity Center to be configured in your AWS environment.

> IAM Identity Center is built on top of AWS Identity and Access Management (IAM) to simplify access management to multiple AWS accounts, AWS applications, and other SAML-enabled cloud applications. In IAM Identity Center, you create, or connect, your workforce users for use across AWS. You can choose to manage access just to your AWS accounts, just to your cloud applications, or to both. You can create users directly in IAM Identity Center, or you can bring them from your existing workforce directory. With IAM Identity Center, you get a unified administration experience to define, customize, and assign fine-grained access. Your workforce users get a user portal to access their assigned AWS accounts or cloud applications. [1]

> **Why Sandbox Studio needs it**: Users need seamless access to their assigned sandbox accounts. Identity Center provides single sign-on capabilities and centralised user management, allowing Sandbox Studio to grant and revoke access to sandbox environments automatically.

Refer to this page to learn how to enable IAM Identity Center:

https://docs.aws.amazon.com/singlesignon/latest/userguide/enable-identity-center.html

---

4. Resource Access Manager (RAM)

Enable resource sharing in your AWS organisation using AWS Resource Access Manager (RAM).

>

> AWS Resource Access Manager (AWS RAM) helps you securely share your resources across AWS accounts, within your organization or organizational units (OUs) in AWS Organizations, and with IAM roles and IAM users for supported resource types. You can use AWS RAM to share resources with other AWS accounts. This eliminates the need to provision and manage resources in every account. When you share a resource with another account, that account is granted access to the resource and any policies and permissions in that account apply to the shared resource. [1]

Why Sandbox Studio needs it: Sandbox Studio needs to share common resources (like SSM Parameters) across multiple sandbox accounts efficiently, reducing duplication and management overhead.

Refer to this page to learn how to enable Resources Sharing within your AWS organisation: https://docs.aws.amazon.com/ram/latest/userguide/getting-started-sharing.html#getting-started-sharing-orgs

---

5. CloudFormation StackSets

Ensure you have activated trusted access for CloudFormation Stack sets.

> AWS CloudFormation StackSets extends the capability of stacks by allowing you to create, update, or delete stacks across multiple accounts and AWS Regions with a single operation. Using an administrator account, you define and manage a CloudFormation template, and use the template as the basis for provisioning stacks into selected target accounts across specified AWS Regions. [1]

Why Sandbox Studio needs it: Sandbox Studio uses StackSets to deploy consistent infrastructure templates across multiple sandbox accounts simultaneously, enabling standardised environment provisioning and updates.

Refer to this page to learn how to activate trusted access for StackSets with AWS Organisations: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacksets-orgs-activate-trusted-access.html

---

6. Cost Explorer

Ensure that you have enabled Cost Explorer in your organisation management account.

> AWS Cost Explorer has an easy-to-use interface that lets you visualise, understand, and manage your AWS costs and usage over time. [1]

> **Why Sandbox Studio needs it**: Sandbox Studio requires cost monitoring to track spending across sandbox accounts, implement cost controls, generate usage reports, and trigger cleanup actions when cost thresholds are exceeded.

Refer to this page to learn how to enable Cost Explorer:

https://docs.aws.amazon.com/cost-management/latest/userguide/ce-enable.html

---

7. Lambda Concurrency Limit

Ensure that your AWS Lambda concurrency limit is adequate; most accounts default to 1000 (which is usually more than sufficient), but in new accounts this limit may be set to 10, in which case you should raise a service quota request to increase it via AWS Service Quotas.

Note: This limit increase should be applied to the **Hub Account**.

> Concurrency is the number of in-flight requests that your AWS Lambda function is handling at the same time. For each concurrent request, Lambda provisions a separate instance of your execution environment. As your functions receive more requests, Lambda automatically handles scaling the number of execution environments until you reach your account's concurrency limit. By default, Lambda provides your account with a total concurrency limit of 1,000 concurrent executions across all functions in an AWS Region. To support your specific account needs, you can request a quota increase and configure function-level concurrency controls so that your critical functions don't experience throttling. [1]
>
> If the Applied quota value is less than 1000, select the **Request quota increase** button to request an increase to this value to at least 1000 before deploying the solution. [2]

Refer to this page to learn how to request the quota increase for the concurrency limit.

https://repost.aws/knowledge-center/lambda-concurrency-limit-increase

---

Choosing your region(s)

When setting up Sandbox Studio, choosing the correct AWS Regions is an important step. The regions you select determine where the solution is deployed, which regions users can access, and how accounts are cleaned up.

---

1. Identify Your Home Region

In an AWS Organisation setup, **IAM Identity Center (IDC)** is enabled in one specific Region. This Region becomes your **home Region** and must be used for deploying Sandbox Studio:

- **Organisation Management Account** – deploy into this Region.
- **Hub Account** – deploy into this same Region.

> All core solution stacks (AccountPool, IDC, Network, Data, SES, Compute, API) must be deployed in the same home region.

---

2. Select Managed Regions

During installation, you specify which AWS Regions Sandbox Studio will **manage**. This has two main effects:

a. Service Control Policies (SCPs)

- SCPs are applied at the **Organisational Unit (OU)** level.
- They restrict users to the Regions you whitelist.
- Users cannot deploy resources into Regions outside of this list.
- Some AWS services (e.g. **IAM**, **CloudFront**) are considered *global services* and are not restricted by SCPs.

b. Account Clean-Up

- When a sandbox account expires, a clean-up job is triggered.
- This job scans only the whitelisted Regions.
- More Regions = longer scan and clean-up time.
- Limiting Regions speeds up recycling while maintaining governance.

Best Practices

- **Keep your managed Regions list small** – choose only the Regions your teams genuinely need.

- **Consider compliance requirements** – some organisations must restrict usage to specific Regions (e.g. EU-only).
- **Balance flexibility with efficiency** – more Regions provide flexibility but increase clean-up time.

Available Regions

Sandbox Studio on AWS is available in the following AWS Regions. Learn more about enabling regions.

Region Name	Region Code
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Jakarta)	ap-southeast-3
Asia Pacific (Melbourne)	ap-southeast-4
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Zurich)	eu-central-2
Europe (Stockholm)	eu-north-1
Europe (Milan)	eu-south-1
Europe (Spain)	eu-south-2
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3

Region Name	Region Code
Middle East (UAE)	me-central-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1

Choosing the hub account

Sandbox Studio requires multiple AWS accounts to function. These accounts follow a **hub-and-spoke model**, where a central **hub account** manages a pool of **sandbox accounts**. The **organisation management account** also plays a key role, as certain AWS services can only be controlled from this top-level account.

There are three types of AWS accounts involved:

Organisation Management Account

- This is the top-level account in an AWS Organisation.
- It is automatically created when you set up the organisation.
- It has full administrative control over all member accounts.
- AWS recommends avoiding workloads in this account; instead, use it primarily for governance and management (e.g. running AWS Control Tower).

Why Sandbox Studio needs this account

Sandbox Studio requires limited components to be deployed into the organisation management account because:

1. **Organisational Units (OUs) and Service Control Policies (SCPs):**
   Only the management account can create and manage OUs and SCPs. Sandbox Studio uses these to organise accounts and enforce guardrails. Accounts are automatically moved between OUs during their lifecycle (e.g. from “Active” to “Cleanup”).
2. **Identity setup:**
   The initial set of IAM Identity Center roles and groups for Sandbox Studio is created in the management account. These are used for authentication and authorisation of users.
3. **Cost management:**
   The management account provides access to consolidated billing and Cost Explorer data. Sandbox Studio uses this to query costs for sandbox accounts in bulk, reducing overhead compared to querying accounts individually.

> **Important:** Two of the seven CloudFormation stacks **must** be installed in the organisation management account. See CloudFormation templates section for more details.

Hub Account

- A dedicated member account used to host most of the Sandbox Studio solution.
- Acts as the **central hub** that manages sandbox accounts (the “spokes”).
- Runs shared infrastructure, automation, and orchestration services.

Benefits of using a hub account

- Keeps the organisation management account clean and reserved for governance only.
- Separates operational workloads from core AWS Organisation functions.
- Provides a secure, centralised place for automation and account lifecycle management.

Sandbox Accounts

- These are the accounts actually handed out to users.
- They are recycled and reused through an automated lifecycle.
- Administrators create a pool of accounts, then onboard them into Sandbox Studio.
- Once onboarded, the accounts are considered **managed** by Sandbox Studio.

The system controls their lifecycle by:

- Assigning them to OUs (e.g. “Active” or “Cleanup”).
- Applying SCPs and guardrails.
- Resetting and recycling them after a lease expires.

Note: Sandbox accounts are intended for non-production use. If your users are looking for ways to provision production ready accounts, consider alternative solutions.

Deployment Options

You have two choices when deploying Sandbox Studio:

1. **Deploy everything into the organisation management account**
   - Simplifies the setup.
   - Not recommended by AWS, as it mixes governance functions with workloads.
2. **Split the deployment between management and hub accounts (recommended)**
   - Management account runs only the required governance components.
   - Hub account runs the main solution.
   - Provides better alignment with AWS best practices for multi-account security and governance.

Understand running costs

Running Sandbox Studio does involve some ongoing AWS costs, but these are generally modest and reflect the standard services needed to keep things running securely and reliably. You can think of them as the “behind-the-scenes” charges for the hub account that coordinates all of your sandbox activity, plus whatever your team chooses to spend in the sandbox accounts themselves.

There are three areas to be aware of:

1. **Hub account running costs**
2. **Sandbox account usage costs**
3. **Sandbox Studio licensing**

---

1. Hub Account Running Costs

The **hub account** is where Sandbox Studio itself lives. It runs the background services that make the platform work—things like APIs, databases, and networking.

Some typical monthly costs you might see:

a) Core compute services

These are the serverless AWS services that power the application — **Lambda**, **API Gateway**, **Step Functions**, **CloudFront**, **Amazon S3**, **KMS,** and **SES**.

- **Typical spend → USD $30 – $60 per month.**

b) Web Application Firewall (WAF)

Helps protect your Sandbox Studio web interface from unwanted traffic.

- **Typical spend → USD $10–$12 per month.**

c) AWS Cost Explorer API

Used to fetch the latest spend data from your sandbox accounts (so you can see usage and enforce limits). Sandbox Studio checks once an hour, which works out to:

- **USD $7.20 per month.**

d) Database (Amazon RDS)

Stores all the information about accounts, budgets, and leases. The default setup uses a lightweight **t4g.micro PostgreSQL instance** to keep things cost-effective.

- **Typical spend → USD $35–$45 per month.**

> You can upgrade the database for extra reliability (e.g. Multi-AZ, larger instance, automated backups), but that will naturally add to the monthly cost.

e) Networking (VPC, NAT Gateways, VPC Endpoints)

Provides secure private networking for the database and functions. By default, this includes 2 NAT gateways and 4 VPC endpoints.

- **Typical spend → about USD $125 per month.**

> If you want to reduce network costs, you can customise the networking — for example, by using a NAT instance, routing traffic through a shared networking account, or dropping VPC endpoints in favour of internet access.

See: AWS services in this solution for the full list of services involved.

---

2. Sandbox Account Usage Costs

Each sandbox account has its own AWS bill, which depends entirely on how people use it.

- You control this by **setting budgets and thresholds** in account templates.
- Sandbox Studio automatically enforces these budgets, but be aware that AWS Cost Explorer data can be delayed by up to 8 hours.

> This means actual spend might go slightly over the set budget before the system notices. To stay safe, we recommend setting your budget a little below your maximum acceptable spend.

In short, sandbox account costs are **your choice**—you decide the budgets, and Sandbox Studio helps keep them under control.

---

3. Sandbox Studio Licensing

Licensing is straightforward:

- **Free Tier**: Manage up to 3 AWS accounts at no cost.
- **AWS Marketplace Upgrade**: If you want to manage more than 3 accounts, you can upgrade directly through AWS Marketplace.
- **Education Discounts**: Heavily reduced rates are available for educational use — contact us for details.

See: Free Tier and Upgrading.

Creating sandbox accounts

Sandbox Studio works by managing a **pool of AWS accounts**. These accounts are pre-provisioned by your organisation and then handed out to users as sandboxes when requested. Sandbox Studio does not create new AWS accounts itself; instead, it manages the lifecycle of accounts that you provide.

Account Pool and Lifecycle

When a user requests a sandbox:

1. An AWS account is **allocated from the pool**.
2. Sandbox Studio applies the correct policies, budgets, and permissions.
3. The user is granted access to the account.
4. Sandbox Studio continuously monitors usage, including:
   - **Duration** (how long the account has been leased)
   - **Costs** (how much has been spent)

When a lease expires or a budget limit is reached:

- The account is **revoked from the user**.
- All resources in the account are cleaned up using the configured **cleaner settings** (by default, AWS Nuke is used).
- The account is returned to the pool for future use (**recycled**).

Provisioning New Accounts

Sandbox Studio does not provision AWS accounts directly. It is the responsibility of **administrators** to create new accounts before onboarding them into Sandbox Studio.

You can use any existing organisational process to provision accounts, including:

- **AWS Control Tower**
- **Landing Zone Accelerator**
- **Terraform or other automation tools**
- **Manual account creation in AWS Organisations**

> **Note:** Sandbox Studio is agnostic of how you provision new AWS accounts. It does not dictate how you create accounts; it only requires that the accounts are onboarded to be managed by Sandbox Studio.

Onboarding Accounts

Before Sandbox Studio can manage accounts, they must be **onboarded**. Onboarding ensures Sandbox Studio can take full lifecycle control of the account.

Onboarding involves:

1. **Moving the account** into the designated **Sandbox OU** within AWS Organisations.
   - Sandbox Studio configures this OU during installation.
   - It applies guardrails and policies to all accounts inside it.
2. **Registering the account** inside the Sandbox Studio console.
   - Use the **AWS Accounts** page in the administrator view.
   - Select the account to onboard and confirm management by Sandbox Studio.

Once onboarded, the account becomes fully managed. Sandbox Studio will:

- Assign and track leases
- Monitor budgets and thresholds
- Clean and recycle the account at the end of each lease

Capacity Planning

As an **IT administrator**, you are responsible for ensuring there are enough accounts in the pool to meet demand. Consider:

- **Number of active users** – how many developers, students, or testers will need accounts at once.
- **Expected workloads** – training, hackathons, or workshops may need dozens of accounts at short notice.
- **Recycling time** – accounts are not available again until after cleanup completes.

Best practice is to provision slightly more accounts than your peak expected demand to avoid user delays.

External identity provider setup (Optional)

Many organisations, particularly those running a multi-account AWS environment, use **AWS IAM Identity Center** with an external identity provider such as **Microsoft Active Directory, Microsoft Entra ID, or Okta**. This allows centralised identity management, where one platform governs access across multiple enterprise systems.

If your organisation uses an external identity platform (for example, Entra), you will need to align its group setup with **Sandbox Studio's IAM Identity Center groups**.

Default Groups in IAM Identity Center

When you install Sandbox Studio, the solution automatically provisions **three groups** in IAM Identity Center. These groups control access based on role type:

1. Administrators

Responsible for configuring and maintaining Sandbox Studio. Administrators are responsible for:

- Setting global policies (e.g. maximum budgets and cleanup rules).
- Provisioning new sandbox accounts and monitoring the sandbox account pool.
- Overseeing security and governance settings.

2. Managers

Oversee day-to-day sandbox usage within a department or team. Managers are responsible for

- Approving or rejecting sandbox requests within their team/department.
- Creating and managing account templates including budgets, pre-provisioned resources and permissions.
- Tracking spending and activity for supervised accounts.

3. Sandbox Users

Login to sandbox accounts and use them for development, testing, training, or experimentation.

Group Naming

The **default names** created by Sandbox Studio are:

- `<namespace>_SsAdminsGroup`
- `<namespace>_SsManagersGroup`
- `<namespace>_SsUsersGroup`

You can change these names during installation.

> **Important:** You must create groups in your external identity platform (e.g. Entra, Okta) with the **exact same names** you configure in Sandbox Studio.

Linking External Identity Provider Groups

1. **Create Groups in Your Identity Platform**
   - Create groups in Entra/Okta/AD that match the IAM Identity Center group names.
   - Example: If your namespace is `Acme`, create `Acme_SsAdminsGroup`, `Acme_SsManagersGroup`, and `Acme_SsUsersGroup`.
2. **Assign Users to Groups in Your Identity Platform**
   - Add users to the relevant group based on their role.
   - Example: Developers should be added to the `SsUsersGroup`, team leads to `SsManagersGroup`, and central admins to `SsAdminsGroup`.
3. **Synchronisation with IAM Identity Center**
   - IAM Identity Center automatically syncs external groups.
   - Once a user is added to the external group, they will inherit the corresponding **Sandbox Studio role and permissions**.