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

---

Revision #25
Created 2025-07-30 11:34:41 UTC by Paul
Updated 2025-10-16 09:39:41 UTC by Paul