Post-deployment configuration tasks

Note: You only need to read this section if you have decided to deploy the solution manually.

Once the stacks are deployed successfully, you'll need to carry out some manual configuration tasks. These ensure Sandbox Studio integrates with your organisation's identity provider, DNS, and that other application settings are initialised.

At a high level, you will:

1. **Set up a SAML 2.0 application** in IAM Identity Center, and assign Sandbox Studio groups to it.
2. **Configure DNS (optional)** for a custom domain.
3. **Update AWS AppConfig settings** (IdP settings, web app URL, access portal, email address).
4. **Store the IdP certificate** in AWS Secrets Manager.
5. Add **initial users** to Sandbox Studio **groups** in IAM Identity Center**.**

- Create an IAM Identity Center application
- Add initial users
- Update AWS AppConfig
- Update AWS Secrets Manager
- Logging into the web UI
- Setup a custom domain (Optional)

Create an IAM Identity Center application

1. Login to the AWS console and open IAM Identity Center.
2. Navigate to **Applications** → **Add application**.
3. Select **I have an application I want to setup** and chose **SAML 2.0**.
4. Enter the following details
   - **Display name**: `Sandbox Studio` (or your preferred name)
   - **Description**: e.g. `Sandbox Studio allows users to access AWS sandbox accounts`
   - Leave **Application start URL** and **Relay state** blank.
   - **Application Metadata**
     - Select **Manually type your metadata values**
     - **Application ACS URL** will be `https://<your-app-url>/api/auth/login/callback` (for now, use the **CloudFrontDistributionUrl**; if you later add a custom domain, come back and update this)
     - **Audience (Entity ID)**: `SandboxStudio`
     - **Submit**.
5. From the list of applications, choose the SAML application you just set up.
6. Click **Actions** → **Edit attribute mappings**.
7. Enter the following attributes:

User attribute in the application	Maps to this value...	Format
Subject	${user:email}	emailAddress
ID	${user:AD_GUID}	unspecified

8. **Save changes**.
9. On the application, page click **Assign users or groups**.
10. Assign the **three groups** created by the `SandboxStudio-IDC` stack (Admin / Manager / User) to this application.
11. **Done**.

You have now successfully set up a custom IAM Identity Center Application.

Extract application details

Before proceeding to the next step, you will need to extract the following information which will be used in subsequent steps.

1. Click **Actions** → **Edit configuration**.

2. Take note of:
   - **IAM Identity Center sign-in URL**
   - **IAM Identity Center sign-out URL**
   - Download the **IAM Identity Center Certificate**
3. Also take note of the:
   1. **Web App URL** - this will be the same URL as the **Application ACS URL** in the previous step **without** the `/api/auth/login/callback` part.
   2. **Audience (Entity ID)** from the previous step.
   3. **AWS Access Portal URL** - this is always `https://<IdentityStoreId>.awsapps.com/start`

Keep these details handy as you will need them in one of the upcoming steps.

Add initial users

The IDC CloudFormation deployment creates three default groups in IAM Identity Center (you can customise their names when launching the `SandboxStudio-IDC` stack):

- **Admins Group** — members of this group have *full access* to Sandbox Studio. They can configure settings, manage budgets, control permissions, and generally administer the platform. You should place only a small number of trusted users here.
- **Managers Group** — managers can create and manage sandboxes for their teams but do not have full platform-wide administrative rights.
- **Users Group** — standard users can request and use sandboxes but cannot configure or administer Sandbox Studio itself.

To set up your initial administrators:

1. Open the IAM Identity Center console.
2. Go to **Groups**, then select the Admins group (for example, `MySs_SsAdminsGroup`).
3. Choose **Add users**, search for the user accounts you want to designate as administrators, and assign them to this group.
4. Repeat the same process for Managers and Users if you want to prepare those groups now.

Update AWS AppConfig

AWS AppConfig is used by Sandbox Studio to store its runtime configuration. You will need to update this configuration after the CloudFormation stacks have been deployed so that Sandbox Studio knows how to authenticate users and where to route traffic.

If AppConfig is not updated correctly, users will not be able to log in or send/receive notifications.

1. Open AWS AppConfig
   - In the **Hub account**, go to the AWS Console.
   - Navigate to AWS AppConfig under **Systems Manager**.
2. **Locate the Sandbox Studio configuration profile**
   - The **SandboxStudio-Data** stack creates an AppConfig application and configuration profile.
   - Use the stack outputs to identify the:
     - **Application ID**
     - **Environment ID**
     - **Configuration Profile ID**
3. **Edit the configuration**
   Update the following fields with values from your environment:

Setting	Description
IdP Sign In URL	The login URL from your Identity Center SAML application.
IdP Sign Out URL	The logout URL from your Identity Center SAML application.
IDP Audience	The SAML audience used when previously setting up the IAM Identity Center Application.
Web App URL	The URL for users to access Sandbox Studio (CloudFront URL or your custom DNS).
AWS Access Portal URL	The IAM Identity Center portal URL.
Notification Email	The “From” address Sandbox Studio uses to send emails (must be verified in SES).

4. **Deploy the configuration**
   - Save the updated configuration.
   - Create a new hosted configuration version.
   - Deploy the configuration to the **Sandbox Studio environment**.

You're application config should look like the YAML configuration shown below.

Note: you should only update the **auth** and **notification** attributes and leave other attributes in place.

```
...
auth:
  idpSignInUrl: https://portal.sso.<region>.amazonaws.com/saml/assertion/<id>
  idpSignOutUrl: https://portal.sso.<region>.amazonaws.com/saml/logout/<id>
  idpAudience: SandboxStudio
  awsAccessPortalUrl: https://d-<id>.awsapps.com/start
  webAppUrl: https://<id>.cloudfront.net
  sessionDurationInMinutes: 60
notification:
  emailFrom: sandboxstudio@example.com
...
```

Update AWS Secrets Manager

AWS Secrets Manager is used to store the SAML Identity Provider (IdP) certificate securely. The SandboxStudio-API stack creates a secret for this purpose. You must update it with the correct certificate from your Identity Center application.

If the certificate is missing or incorrect, Sandbox Studio will not be able to validate SAML assertions, and user login will fail.

1. **Get the secret ARN**
   - Check the outputs of the **SandboxStudio-API** CloudFormation stack.
   - Look for the output key **IdpCertArn**.
2. **Retrieve the IdP certificate**
   - Open the **IAM Identity Center application** you created for Sandbox Studio.
   - Download the **SAML metadata XML** or copy the signing certificate directly.
   - Ensure it is in **PEM format** (starts with `-----BEGIN CERTIFICATE-----`).
3. **Update the secret**
   - In the **Hub account**, open **AWS Secrets Manager**.
   - Find the secret with the ARN from step 1.
   - Edit the secret value.
   - Paste in the IdP certificate.
4. **Save and test**
   - Save the new secret value.
   - Restart the login flow in Sandbox Studio to confirm that SAML authentication works.

Logging into the web UI

Once you have completed the installation of Sandbox Studio, you can log into the web user interface (UI).

Finding the Login URL

The login page is hosted behind an **Amazon CloudFront distribution** that was created during installation. To find the URL:

1. Sign in to the **AWS Management Console** for your **Hub account**.
2. Navigate to **CloudFormation** and open the stack created for Sandbox Studio.
3. Go to the **Outputs** tab.
4. Look for the output parameter named **CloudFrontDistributionUrl**.
5. The value of this parameter is the **login URL** for your Sandbox Studio environment.

Example:

`https://d123example.cloudfront.net`

Use this URL in your browser to access the Sandbox Studio login page.

What Happens Next

- You will be redirected to the **AWS IAM Identity Center (SSO)** sign-in page.
- Log in using your corporate or assigned credentials.
- Once authenticated, you will land on the Sandbox Studio home page (User, Manager, or Administrator view depending on your role).

Setup a custom domain (Optional)

By default, Sandbox Studio is deployed behind an AWS CloudFront distribution. Users can access it using the **CloudFront distribution URL** that is output from the `SandboxStudio-API` stack.

However, in most organisations you will want to provide a more user-friendly, branded domain name (e.g. `sandbox.example.com`). This requires setting up a **custom domain** in CloudFront and updating your **DNS provider** to route traffic to Sandbox Studio.

---

1. Retrieve CloudFront distribution details

- Go to the AWS Console in the **Hub account**.
- Navigate to **CloudFront**.
- Find the distribution created by the `SandboxStudio-API` stack.
- From the stack outputs, note:
  - **CloudFrontDistributionUrl** (e.g. `d12345abcdef.cloudfront.net`)
  - **CloudFrontDistributionId** (used if you need to update settings later)

---

2. Choose your custom domain

Decide on the domain name that will be used for Sandbox Studio. Examples:

- `sandbox.yourcompany.com`
- `studiosandbox.example.org`

Make sure this domain is one you control in your DNS provider (such as **Route 53**, Cloudflare, or another registrar).

---

3. Update CloudFront distribution with Alternate Domain Name (CNAME)

- In the **CloudFront distribution settings**, add your chosen domain under **Alternate Domain Names (CNAMEs)**.
- If you're using the AWS Console:
  1. Open your distribution → **Settings** → **General** → **Alternate Domain Names**.
  2. Click **Edit** and add your custom domain name.

>

> CloudFront requires an **SSL/TLS certificate** for custom domains.

---

4. Provision an SSL/TLS certificate in ACM

- Go to the **AWS Certificate Manager (ACM)** in the **us-east-1 region** (required for CloudFront).
- Request a certificate for your custom domain (e.g. `sandbox.example.com`).
- Validate the certificate using DNS (preferred) or email validation.
- Once validated, return to your CloudFront distribution and attach this ACM certificate under **Custom SSL Certificate**.

---

5. Update your DNS provider

- In your DNS provider (e.g. Route 53), create a **CNAME record**:
  - **Name**: your custom domain (e.g. `sandbox.example.com`)
  - **Value**: the CloudFront distribution URL (e.g. `d12345abcdef.cloudfront.net`)
- Save the record.

> It may take up to 30 minutes (or more depending on TTL settings) for DNS changes to propagate.

---

6. Update the ACS URL in Identity Center

Since the login flow depends on the correct **Assertion Consumer Service (ACS) URL**, you must update the Identity Center SAML application configuration:

- Open **IAM Identity Center** in the management account.
- Find the Sandbox Studio custom application.
- Update the ACS URL to:
  `https://<your-custom-domain>/api/auth/login/callback`

Example:
`https://sandbox.example.com/api/auth/login/callback`

This ensures SAML assertions are posted to the correct URL.

---

7. Update the Web App URL in Sandbox Studio

In your **Sandbox Studio** environment:

- Go to "**Settings**" > "**Advanced Settings**" and scroll to "**Authentication Settings**"
- Update the "**Web App URL**" value to your new domain (With no trailing slash)

Example:

`https://sandbox.example.com`

- You should now be able to access (and login) to your Sandbox Studio using the new domain.

Why This Matters

- Using a custom domain makes Sandbox Studio easier for users to remember and access.
- It allows branding (e.g., using your company’s domain).
- Ensures smoother authentication flows by aligning the SAML ACS URL with the URL that users actually log in through.