

Account template creation - Launch Settings

Using launch templates, managers can define what resources need to be set up on new accounts creation. For instance, create S3 buckets, deploy Cloud Formation templates, etc.

Sandbox Studio uses [AWS Codebuild](#) to set up resources.

Home > Account Templates > New Template > Template Details

Add a new Account Template

Give your users a new way to access a temporary AWS account.

- Step 1 Basic Details
- Step 2 Budget
- Step 3 Lease Duration
- Step 4 Account Permissions
- Step 5 Roles & Access
- Step 6 **Launch Settings**
- Step 7 Review

Launch Settings

Run setup before account access [More info](#) 1
Run scripts to prepare the account before it is given to the user.

Download files from S3 2
OPTIONAL Download files from an S3 bucket before scripts run (for example, configs or scripts). Files are saved to /tmp. [Configure S3 access.](#)
e.g. s3://my-bucket/setup-files/

Pre-Launch Script 3
OPTIONAL Runs first to prepare the environment (for example, install tools or set up dependencies).
1

Launch Script 4
Runs to set up the account before access is given to the user.
1

Cancel < Back Next >

Launch Settings (1)

Select "**Run setup before account access**" to enable launch templates. By ticking this item, you need to provide the script to create resources on account creation.

If no resource needs to be created on account creation, untick this box to provide an "empty" account to users.

Download files from S3 (2)

In some scenario, you may need to download files from S3 to create your resources. For example, download a Cloudformation template, download softwares, etc.

You need to provide an S3 path to a folder or a bucket (not to the object directly).

Download files from S3

OPTIONAL Download files from an S3 bucket before scripts run (for example, configs or scripts). Files are saved to /tmp. [Configure S3 access.](#)

s3://account-templates-resources-479547938758/



Sandbox Studio will run a `sync` command to fetch the folders and objects from S3 into the `/tmp` folder of the Codebuild instance.

Sandbox Studio does not require your bucket to be public! Follow the next steps to learn how to configure S3 access

Sandbox Studio requires your bucket to be accessible to the `LaunchTemplateExternalAccessRole` created in your environment.

You can click on "Configure S3 access" to display the S3 bucket policy to apply on your S3 bucket:

[Configure S3 access.](#)

S3 Bucket Policy Configuration

Add this policy to your S3 bucket to allow Sandbox Studio to download files during launch. Replace **your-bucket-name** with your S3 bucket name.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSandboxStudioAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<account-id>:role/LaunchTemplateExternalAccessRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::your-bucket-name",
        "arn:aws:s3::your-bucket-name/*"
      ]
    }
  ]
}
```

Copy to Clipboard

Close

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSandboxStudioAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<hub-account-number>:role/LaunchTemplateExternalAccessRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::your-bucket-name",
        "arn:aws:s3::your-bucket-name/*"
      ]
    }
  ]
}
```

Pre-Launch script (3)

Pre-Launch script allows you to run a script before your actual launch script to, for example, install tools, packages, update the environment, etc.

The environment used to create the resources is based on **Amazon Linux 2023 (x86_64)**. ([aws/codebuild/amazonlinux-x86_64-standard:5.0](https://aws.amazon.com/codebuild/images/amazonlinux-x86_64-standard:5.0))

Launch script (4)

Launch script is your actual script to create resources in your account.

The environment used to create the resources is based on **Amazon Linux 2023 (x86_64)**. ([aws/codebuild/amazonlinux-x86_64-standard:5.0](https://aws.amazon.com/codebuild/images/amazonlinux-x86_64-standard:5.0))

Please refer to following section to return environment details to the Sandbox Studio interface.

Example of script to create a S3 bucket:

```
#!/bin/bash
BUCKET_NAME="sample-bucket-$(date +%s)" # Unique bucket name from current timestamp
REGION="us-east-1"

aws s3 mb --region $REGION s3://$BUCKET_NAME
```

How to set "Environment details" in Sandbox Studio?

Sandbox Studio allows you to display environment details from the launch template script. You can, for example, return URLs, passwords, environment variables, etc.

One interesting use case is to provide access to a VSCode environment: You can create a VSCode environment to your users and return the URL to the user. They don't need to login to the account and can directly access their VSCode with the information provided by the install script.

To set the environment details, you can use the pre-defined `set-sandbox-output` command:

```
VSCODE_URL="https://example.com"
PASSWORD="SuperStrongPassword"

set-sandbox-output --name "VSCodeServerUrl" --value "$VSCODE_URL"
set-sandbox-output --name "Password" --value "$PASSWORD" --is-secret
```

The tool takes 2 parameters (name and value).

For secret strings (i.e.: Password, API keys, etc.), add the parameter `is-secret`

Environment Details (2)

Detail

Value

VSCodeServerUrl

<https://example.com> 

Password

..... 

Secrets variables are saved in AWS Secrets Manager and only displayed when the user clicks on the Display icon.

Having completed all the fields on the Launch Settings page as needed, click on **Next** to [Account template creation - Review](#)

Revision #9

Created 2025-07-22 18:16:41 UTC by Winston

Updated 2026-04-04 10:45:34 UTC by Paul