Next JS is an open-source React-based framework that provides a hybrid static/server-side rendered model. Using Next JS data fetching, you can render your content differently based on your application's use case and build APIs using API Routes.

Launch supports the following features of Next JS out of the box:

- API Routes
- SSR (Server Side Rendering)
- ISR (Incremental Static Regeneration)

The fastest way to get started with Launch is to clone and deploy a sample Contentstack NextJS Starter project. Launch allows you to create a project by importing the website code from GitHub.
Prerequisites

1. Contentstack account
2. Access to Launch for your organization (Note: Only the Organization Admin/Owner has permissions to create Launch projects)
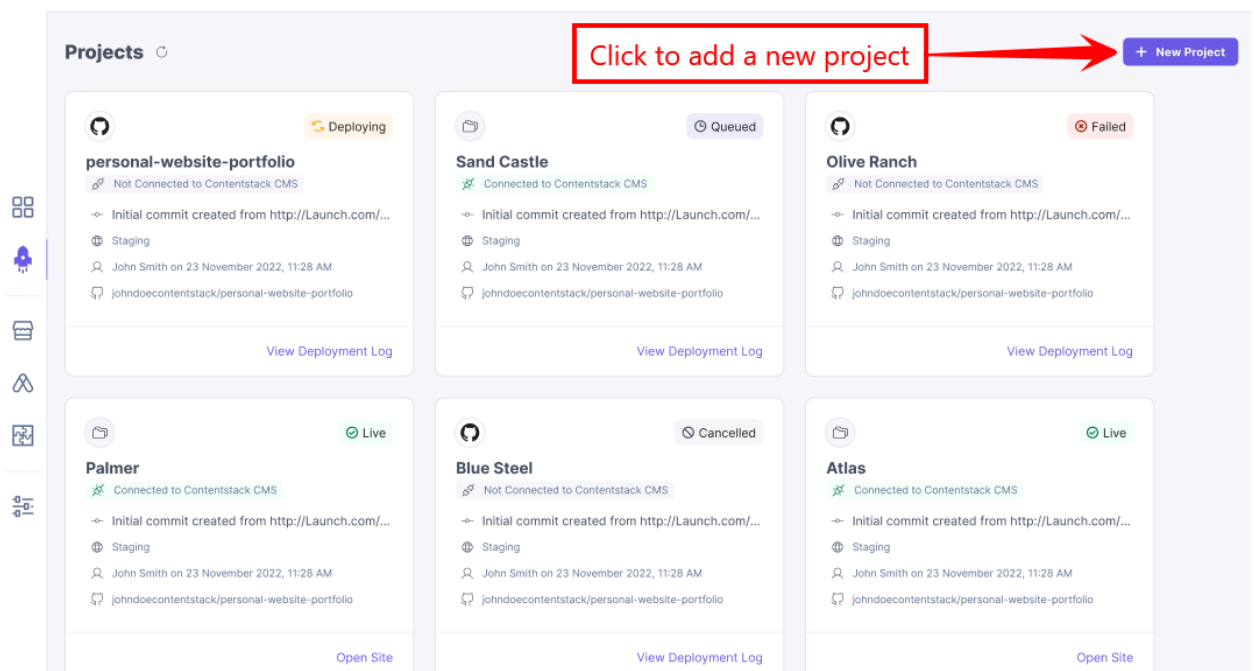3. GitHub account

Steps for Execution
Follow the steps given below to deploy your first Launch project using GitHub:
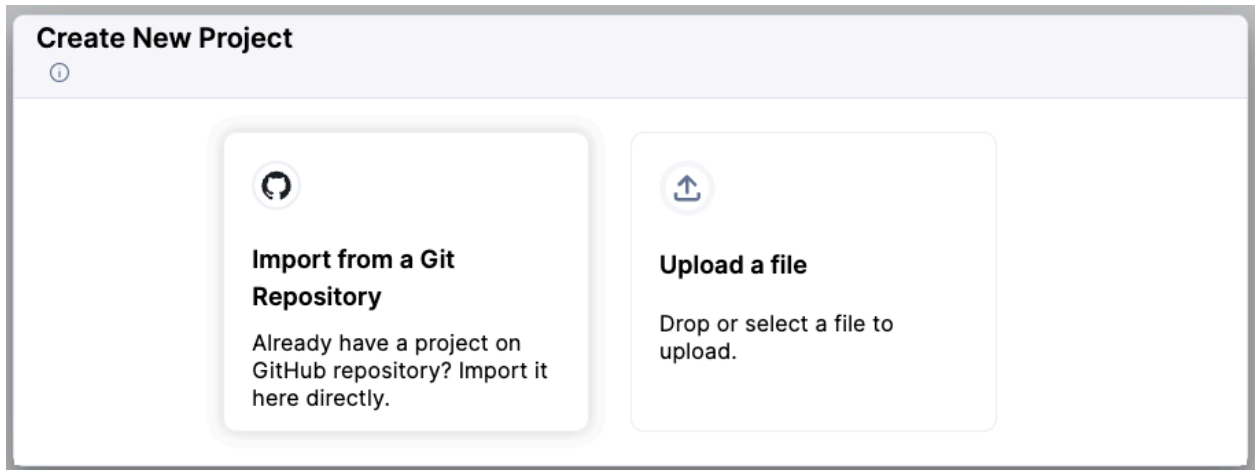
1. Log into Contentstack and click the Launch icon from the left navigation panel, as shown below:
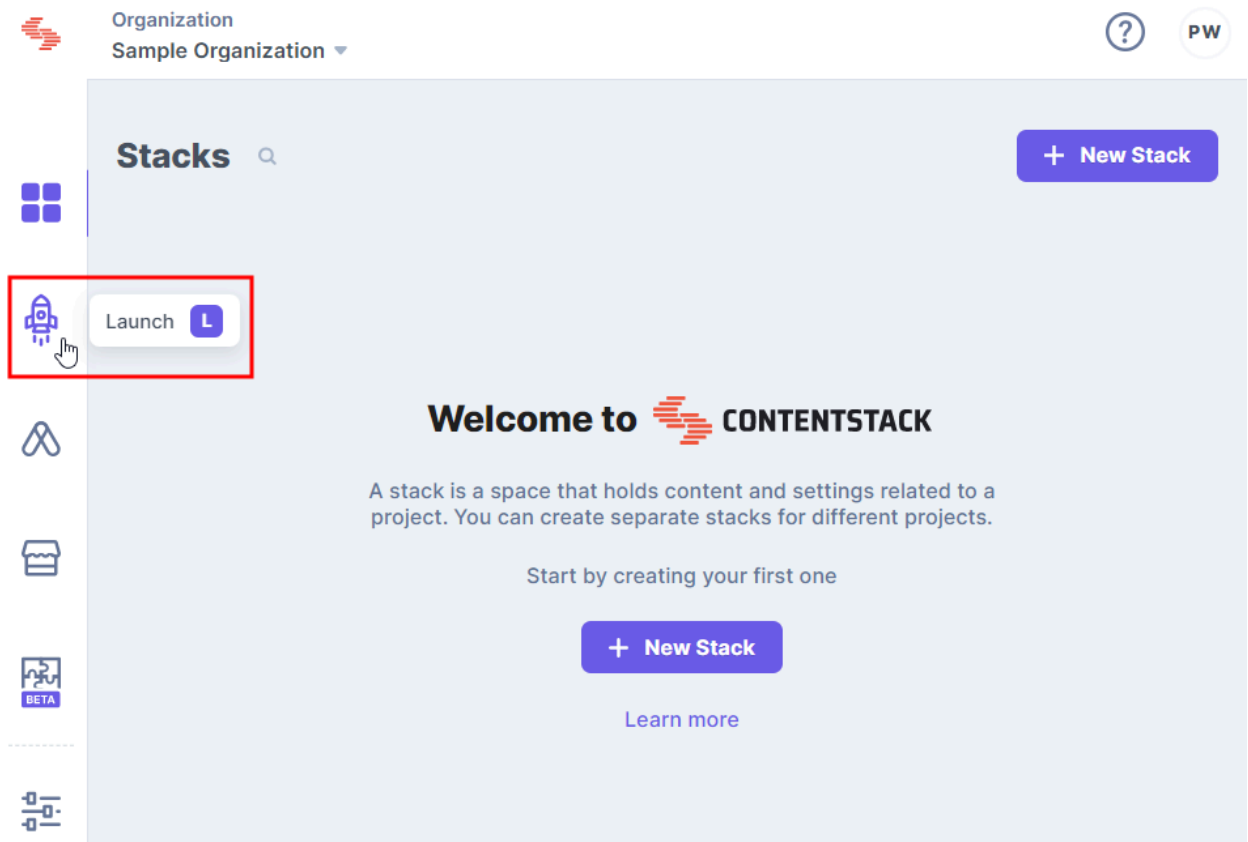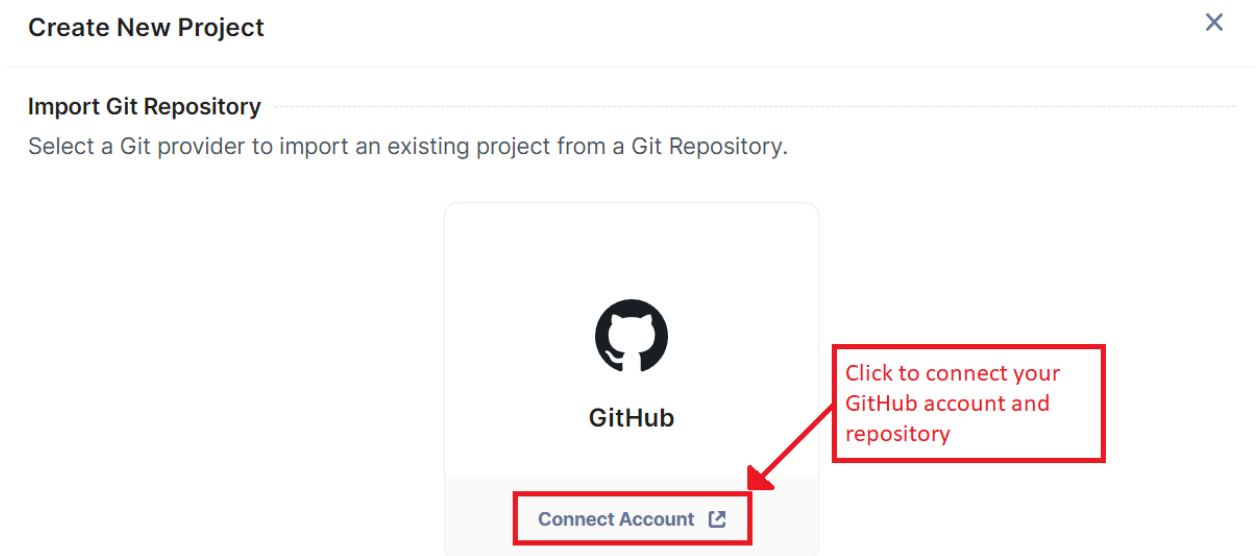
2. Click the + New Project button:

3. In the Create New Project modal, click Import from a Git Repository. (Note: You can also choose to [upload a .zip file to create your project](#).)

Create New Project
ⓘ

Import from a Git Repository

Already have a project on GitHub repository? Import it here directly.

Upload a file

Drop or select a file to upload.

4. Click GitHub:

Create New Project   ✕

Import Git Repository

Select a Git provider to import an existing project from a Git Repository.

GitHub

Connect Account ↗

Click to connect your GitHub account and repository

This will open a modal window to authenticate with your GitHub account.

5. Enter your credentials and log in:

Sign in to **GitHub**
to continue to **Contentstack Launch**

Username or email address

Password   Forgot password?

Sign in

New to GitHub? Create an account.

6. In the Repository Access section, select All repositories.
   Note: If you want to choose specific repositories, pick the Only select
   repositories option and choose the required repository.

Repository access

○ **All repositories**
   This applies to all current *and* future repositories owned by the resource owner.
   Also includes public repositories (read-only).

○ **Only select repositories**
   Select at least one repository.
   Also includes public repositories (read-only).

Save   Cancel

7. Click the Save button
   Note: If you are accessing GitHub through Contentstack for the first time,
   you must authorize Contentstack by clicking the "Install & Authorize" button
   after selecting All repositories.

Install & Authorize Contentstack Launch

Install & Authorize on your personal account

⦿ **All repositories**

This applies to all current *and* future repositories owned by the resource owner. Also includes public repositories (read-only).

○ **Only select repositories**

Select at least one repository. Also includes public repositories (read-only).

with these permissions:

✓ **Read** access to metadata and pull requests

✓ **Read** and **write** access to administration, code, commit statuses, repository hooks, and workflows

Click to install and authorize

Install & Authorize   Cancel

Next: you'll be redirected to https://app.contentstack.com/connected-apps/callback

Once you've connected your GitHub account, you'll be redirected back to Launch.

8. In the Create New Project modal, add the following details.
    1. Repository (Mandatory): Select your Git repository.
    2. Git Branch (Mandatory): By default, master or main is selected as the branch. You can choose another branch from the dropdown.
    3. Project Name (Mandatory): This is auto-populated when you select the repository, but you can edit it as needed.
    4. Environment Name (Mandatory): The environment name is automatically populated as Default, but can be changed. Note that this is specific to Launch and not directly related to any environments in your stack.
    5. Build and Output Settings (Mandatory): The fields in this section get auto-populated based on the detected framework. (If no framework is detected, you can specify your own Build Command and Output Directory.)
    6. Environment Variables (Optional): Enter the key and value of one or more environment variables. You can also use the "Connect and Import Variables" option to automatically import API-related variables from a specific stack and delivery token.

← **Create New Project**   ✕

Repository

TestUser101/contentstack-nextjs-starter-app   ⌄

Git Branch*

master   ⌄

Detected framework: NextJs

Project Name*

contentstack-nextjs-starter-app

Environment Name*

Default

Build and Output Settings

Framework Preset*

NEXTJS   ⌄

Build Command   **Output Directory***

npm run build   ./.next

Environment Variables

Cancel   ← Back   ☁ Deploy

← Create New Project   ✕

Build and Output Settings

Framework Preset*

NEXTJS   ⌄

Build Command   Output Directory*

npm run build   ./.next

Environment Variables

┌─────────────────────────────────────┐
│ 🔴 Connect and Import Variables   │
└─────────────────────────────────────┘

Key Value Edit   **Bulk Edit**

Key and Value*
CONTENTSTACK_API_HOST=api.domain.io

CONTENTSTACK_CDN=cdn.domain.com

CONTENTSTACK_API_KEY=vn3   33h

CONTENTSTACK_DELIVERY_TOKEN=ljnv   d08

CONTENTSTACK_ENVIRONMENT=production

Cancel   ← Back   ☁ Deploy

Note: Click Back to revert the changes and re-enter details.

9. Once ready, click Deploy to save and deploy the project.
   You'll see the project's status update in real time, from Queued to Deploying
   to Deployed. Note: depending on the size and complexity of your specific
   application, it may take several minutes to build and deploy your site.
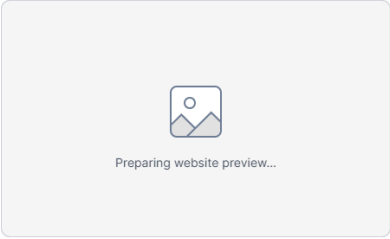
Upon successful deployment, you will see the following screen:

You can preview the deployed website by copying the URL from the URL section under Deployment Information and opening it in a browser.

If you want to deploy the changes from any other commits, you can click the Redeploy button and choose the required commit.

Note: The log details of the current deployment can be found under the Logs section.