Create a vault

This guide helps you create your first Skyflow vault. You can use Studio or APIs to access the Quickstart vault, create a vault with a template, or create a custom vault.

Prerequisites

Studio | API

- Sign in to your Skyflow account. If you don't have an account, sign up for a free trial.
- A bearer token to authenticate API calls. For a short-lived token, use the following process. To generate tokens from service accounts, see Authenticate.
  1. In Studio, click your account icon and choose **Generate API Bearer Token**.
  2. Click **Generate Token**.
- A device with the following tools available:
  - A terminal that can run bash commands
  - `curl`
  - jq 1.6 or greater
- Skyflow account, vault, and workspace details:
  1. In Studio, click **vault menu icon > View vault details**.
  2. Note your **Account ID**, **Vault ID**, and **Vault URL** values.
- Your environment's Management API URL:
  - Trial or Production: `https://manage.skyflowapis.com`
  - Staging: `https://manage.skyflowapis-preview.com`
- Set environment variables for your account and vault details:

```bash
export MANAGEMENT_URL=$MANAGEMENT_URL
export ACCOUNT_ID=$ACCOUNT_ID
export TOKEN=$TOKEN
export WORKSPACE_ID=$WORKSPACE_ID
export VAULT_NAME=$VAULT_NAME
export SCHEMA=$SCHEMA
export TAGS=$TAGS
```

Create a Quickstart vault

The Quickstart vault is a template that any Skyflow account can access. It is designed to help you get started with Skyflow APIs.

If you have a trial account, a Quickstart vault is automatically created for you. If you don't see the Quickstart vault, you can create one from the Vault Dashboard.

1. Click **Add Vault**.
2. Click **Start with a template**, then click **Quickstart**.
3. Click **Create**.

The Quickstart vault uses a simple schema with two tables, `credit_cards` and `persons`, and populates the tables with the applicable records.

Start with a template

Skyflow offers pre-built vault templates based on popular use cases that you can use as a starting point for your vault. For example, the Payments vault template stores data about credit cards, credit scores, and customer PII.

The following table details the available templates.

Template	Edit schema	Data	Table count	Relational	Tables
Quickstart	Yes*	Yes	2	No	Credit Card, Persons
Payment	Yes	No	7	Yes	Consumers, Alloy kyc, Cards, Transactions, Bank Accounts, Financial Service Providers, Merchants
PIIData	Yes	No	1	No	PII fields
CustomerIdentity	Yes	No	4	Yes	Persons, Identifiers, Contacts, Organizations
Plaid	Yes	No	14	No	Accounts, Numbers SCH, Liabilities Mortgage, Liabilities Student, Holdings, Liabilities APRS, Balances, Owners Email, Owners Names, Owners Phone Numbers, Owners Addresses, Users, Transactions, Credentials

Note: You can't change fields containing data.

Studio | API

To create a vault using a template, call List Vault Templates to retrieve the available templates.

```bash
curl -s -X GET "$MANAGEMENT_URL/v1/vault-templates" \
-H "X-SKYFLOW-ACCOUNT-ID: $ACCOUNT_ID" \
-H "Accept: application/json" \
-H "Authorization: Bearer $TOKEN"
```

The response returns a list of templates.

Update your environment variables to include the `templateID`:

```bash
export TEMPLATE_ID=$TEMPLATE_ID
```

Run the following command to create a vault with a template:

```bash
curl -s -X POST "$MANAGEMENT_URL/v1/vaults" \
-H "X-SKYFLOW-ACCOUNT-ID: $ACCOUNT_ID" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "name": "'"$VAULT_NAME"'",
  "description": "$DESCRIPTION",
  "templateID": "'"$TEMPLATE_ID"'",
  "workspaceID": "'"$WORKSPACE_ID"'"
}'
```

The response returns the `vaultID`.

Create a vault schema

You can create a vault by uploading a vault schema directly to Skyflow. Schema files must be in JSON format and can't exceed 15 MB. Visit the list of Vault settings to set your schema accordingly.

The following example is a sample schema:

```bash
{
  "name": "simpleVaultExample",
  "description": "A vault with 1 table",
  "vaultSchema": {
    "schemas": [
      {
        "name": "table_1",
        "fields": [
          {
            "name": "skyflow_id",
            "datatype": "DT_STRING"
          },
          {
            "name": "age",
            "datatype": "DT_INT32"
          },
          {
            "name": "ssn",
            "datatype": "DT_STRING",
            "tags": [
              {
                "name": "skyflow.options.replace_pattern",
                "values": [
                  "XXX${1}XX${2}${3}"
                ]
              },
              {
                "name": "skyflow.options.format_preserving_regex",
                "values": [
                  "^[0-9]{3}-[0-9]{2}-([0-9]{4})$"
                ]
              },
              {
                "name": "skyflow.options.default_dlp_policy",
                "values": [
                  "REDACT"
                ]
              },
              {
                "name": "skyflow.options.operation",
                "values": [
                  "EXACT_MATCH"
                ]
              },
              {
                "name": "skyflow.options.find_pattern",
                "values": [
                  "^[0-9]{3}([- ])?[0-9]{2}([- ])?([0-9]{4})$"
                ]
              },
              {
                "name": "skyflow.options.default_token_policy",
                "values": [
                  "FORMAT_PRESERVING_TOKEN"
                ]
              },
              {
                "name": "skyflow.validation.regular_exp",
                "values": [
                  "^$|^([0-9]{3}-?[0-9]{2}-?[0-9]{4})$"
                ]
              }
            ]
          },
          {
            "name": "marital_status",
            "datatype": "DT_STRING",
            "tags": [
              {
                "name": "skyflow.validation.predefinedvalues",
                "values": [
                  "UNSPECIFIED_MARITAL_STATUS",
                  "ANNULLED",
                  "DIVORCED",
                  "SEPARATED",
                  "MARRIED",
                  "UNMARRIED",
                  "WIDOWED"
                ]
              },
              {
                "name": "skyflow.options.default_token_policy",
                "values": [
                  "RANDOM_TOKEN"
                ]
              },
              {
                "name": "skyflow.options.default_dlp_policy",
                "values": [
                  "REDACT"
                ]
              },
              {
                "name": "skyflow.options.operation",
                "values": [
                  "EXACT_MATCH"
                ]
              }
            ]
          }
        ],
        "childrenSchemas": [
          {
            "name": "name",
            "description": "",
            "fields": [
              {
                "name": "first_name",
                "datatype": "DT_STRING",
                "tags": [
                  {
                    "name": "skyflow.options.default_token_policy",
                    "values": [
                      "RANDOM_TOKEN"
                    ]
                  },
                  {
                    "name": "skyflow.options.operation",
                    "values": [
                      "EXACT_MATCH"
                    ]
                  }
                ]
              },
              {
                "name": "last_name",
                "datatype": "DT_STRING",
                "tags": []
              }
            ]
          }
        ]
      }
    ]
  },
  "workspaceID": "z10198d5553411def9f2360c809gt3yx"
}
```

Create a custom vault

To create a custom vault, you can start from scratch or upload your schema file directly to Skyflow.

Note: You cannot use spaces and underscores in the Vault Name.

Studio | API

Run the following command to upload your schema:

```bash
curl -s -X POST "$MANAGEMENT_URL/v1/vaults" \
-H "X-SKYFLOW-ACCOUNT-ID: $ACCOUNT_ID" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "name": "'"$VAULT_NAME"'",
  "vaultSchema": {
    "schema": "'"$SCHEMA"'",
    "tags": "'"$TAGS"'"
  }
}'
```

Edit the vault schema

Like a database schema, the vault schema specifies the tables and columns for storing data and their respective data types. The Skyflow vault schema includes extra functionalities that detail privacy-preserving techniques for each column. When creating a vault, it generates a default single-table setup that you can rename. Every vault must have at least one table, and all tables contain a permanent "skyflow_id" column that you cannot alter.

Note: The applicable tables display when you use a template or upload your schema.

If you want to edit your schema after creating the vault, you can return to the schema editing mode by completing the following steps.

Note: When editing a vault's schema, some operations are inactive if there is data in the column. For example, you can't rename or change a column's data type if it has data. You can always add new columns.

Studio | API

When you create a vault or edit a vault's schema, there are various settings (represented as tags in the Management API) that define field behaviors.

To return the latest vault schema, run the following command:

```bash
curl -s -X GET "$MANAGEMENT_URL/v1/vaults/$VAULT_ID/" \
-H "X-SKYFLOW-ACCOUNT-ID: $ACCOUNT_ID"
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Bearer $TOKEN" \
```

Using the returned `schema` and `tags`, your vault schema accordingly.

Run the following command to update your vault with the new schema:

```bash
curl -s -X PATCH "$MANAGEMENT_URL/v1/vaults" \
-H "X-SKYFLOW-ACCOUNT-ID: $ACCOUNT_ID" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d'{
  "vaultSchema": {
    "schemas": "'"$SCHEMAS"'",
    "tags": "'"$TAGS"'"
  }
}'
```

Note: When you edit the vault schema, the system disables some operations if the column contains data. For example, you can't rename or change a column's data type if it has data. However, you can always add new columns.

Reserved keywords

When you create or edit a vault, you need to follow Skyflow's rules for column and table names.

No capital letters or special characters: Use lowercase alphanumeric characters (a-z, 0-9), underscores (_), and hyphens (-) for column and table names.

No SQL keywords: Use of SQL keywords as column or table names is forbidden. SQL reserves keywords for specific purposes in the language, and using them as identifiers can lead to syntax errors and unexpected behavior. Refer to the SQL documentation or a reliable SQL reference guide to familiarize yourself with the reserved keywords list.

No policy keywords: When you create your vault, you set roles and policies for specific data access and security purposes. In addition to SQL keywords, you can't use policy-specific keywords as column or table names.

Add columns with basic data types

When adding columns, you can choose between two data types:

Skyflow Data Types: Common PII elements defined by Skyflow for your convenience

Basic Data Types: Standard database types like integers and strings

Let's start by adding a basic data type column to the table. Click **New Column**, select the **Basic Data Types** tab, and pick a data type, like string.

Now you can configure the settings for the new column:

- On the **General tab**, you can add information about the column, like the name, description, column group, data type, and any regex validations that apply to the column.
- On the **Tokens tab**, you can specify which type of non-sensitive tokens you want to generate for values in the column.
- On the **Redaction tab**, you can choose how this column should be redacted by default. You also have the option to specify a masking format for the column.
- On the **Encryption tab**, you can configure column-level encryption and which encrypted operations you want to enable for the column.
  - If you use column-level encryption, you must enable certain encrypted operations before performing them on that column.

Encrypted operation	Enables
Exact match	=
Aggregation	AVG, COUNT, MAX, MIN, SUM
Comparison	>, <, ORDER BY

  - If you don't use column-level encryption, you can perform all operations on the column. Note that substring matching with the `LIKE` and `ILIKE` requires additional configuration.

  Note: You can't change encryption settings after you insert data into the column.

When you're done, click **Create column** to add it to your schema.

Add columns with Skyflow data types

Now, let's create a column with a Skyflow data type. Click **New column**, select the **Skyflow Data Types** tab, and choose **Social Security Number**.

Skyflow data types pre-configure field settings, such as the data validation on the General tab and the masking format on the Redaction tab. You can alter these settings, including the column name, then click Create column to add it to your schema.

When you're done building your schema, click **Save**.

Configure access controls

To configure access to your vault, click **Access** in the side navigation.

The Access section has three tabs: **Roles**, **People**, and **Service accounts**.

- *People* (users) and *service accounts* are two types of identities that can access your vault: People are human accounts and service accounts are for machine access (For example, if an application backend wants to access the vault).
- *Roles* define what and how each identity can access aspects of your vault. By default, there are three roles defined for a vault: Vault Owner, Vault Editor, and Vault Viewer. Each of these roles has attached *policies* that specify the role's permissions.

The table below summarizes the permissions for each role:

	Vault Viewer	Vault Editor	Vault Owner
Read records with the default redaction level	✅	✅	✅
Create, update, & delete records		✅	✅
Read records in plain text			✅
Create, update, & delete service accounts			✅
Create, update, & delete roles & policies			✅

You can also define custom roles and policies. See Data governance.

Next steps

Learn more about vault settings, explore what Skyflow can do, or learn how to authenticate with Skyflow.