Cloudasta Migration

GCP Setup Script - Execution Guide

How to execute cloudasta-migration-setup.sh
on Google Cloud Shell and macOS

Version 1.3 | February 2026 | Cloudasta Migration Team

Table of Contents

1. Overview

The **cloudasta-migration-setup.sh** script automates the GCP configuration needed for every CloudM Migrate project. Instead of manually clicking through the GCP Console, the script handles project creation, API enablement, service account setup, and key generation in a single run.

What the script automates:

- Creates a GCP project named **Cloudasta-Migration**
- Enables all 13 required APIs (Service Usage, IAP, Org Policy, Admin SDK, Gmail, Calendar, Drive, People, Tasks, Forms, Groups Migration, Chat, Groups Settings)
- Creates a service account named **cloudasta-migration** with Owner role
- Enables domain-wide delegation on the GCP side (via IAM PUT API)
- Configures the OAuth consent screen (Internal)
- Downloads the JSON key and renames it to **<domain>-serviceaccount.json**
- Auto-resolves the **iam.disableServiceAccountKeyCreation** org policy error if encountered
- Saves all output to a persistent log file (**~/cloudasta-migration-<timestamp>.log**) so logs survive terminal disconnections

What requires manual completion:

- **Domain-Wide Delegation** - Adding the Client ID and OAuth scopes in Google Admin Console
- **Chat API Configuration** - Setting app name, avatar, and visibility in GCP Console
- **Drive SDK** - Verifying it is enabled in Google Admin Console

NOTE: The script provides ready-to-paste values and direct links for all manual steps. It also saves a DWD helper text file with the Client ID and scopes for easy reference.

2. Prerequisites

Requirement	Cloud Shell	macOS
gcloud CLI	Pre-installed	Must install manually
Authentication	Pre-authenticated	Run: gcloud auth login
curl	Pre-installed	Pre-installed on macOS
Google Account	Super Admin required	Super Admin required
GCP Terms of Service	Must be accepted (script checks)	Must be accepted (script checks)
Billing Account (optional)	Needed for some APIs	Needed for some APIs

Access the Script

The script is hosted on GitHub. You have two options to access it:

- **Option A (Recommended — Cloud Shell):** Click the **Open in Cloud Shell** link below. This opens Google Cloud Shell with the script automatically downloaded — no manual upload needed:

```
https://shell.cloud.google.com/cloudshell/editor?cloudshell_git_repo=https://github.com/hozefa1997/cloudm-gcp-setup.git&cloudshell_git_branch=main&cloudshell_workspace=.&cloudshell_tutorial=TUTORIAL.md
```

- **Option B (macOS / manual):** Download the script from the GitHub repository or Google Drive shared folder and run it locally.

Information you will need to have ready:

- **Destination domain name** (e.g., example.com) - used for key file naming
- **Admin email address** (e.g., admin@example.com) - your super admin account
- **GCP Organization ID** (optional) - found at GCP Console > IAM & Admin > Settings
- **Billing Account ID** (optional) - found at GCP Console > Billing

3. Option A: Running on Google Cloud Shell

TIP: Google Cloud Shell is the recommended way to run this script. It comes with gcloud pre-installed and pre-authenticated - no setup needed on your Mac.

Step 1: Open the Script in Cloud Shell (One Click)

Click the **Open in Cloud Shell** link from Section 2 (or the README on GitHub). Cloud Shell opens in your browser and automatically clones the script repository into your home directory. A guided tutorial panel also appears on the right side.

If you prefer to open Cloud Shell manually, navigate to `https://shell.cloud.google.com` and upload the script using the three-dot menu > Upload.

Step 2: Run the Script

In the Cloud Shell terminal, navigate to the cloned repo folder and run:

```
cd cloudm-gcp-setup && chmod +x cloudasta-migration-setup.sh &&
./cloudasta-migration-setup.sh
```

Step 4: Follow the Prompts

The script will ask for the domain name, admin email, and optional settings. See Section 5 for a full walkthrough of each prompt.

Step 5: Download the Key File

When the key is created, the script will offer to download it to your local machine. Type **y** and your browser will show a download prompt for the JSON key file.

If the automatic download does not work, you can manually download: click the three-dot menu in Cloud Shell > **Download file** > enter the full file path shown by the script.

NOTE: Cloud Shell sessions time out after 20 minutes of inactivity. The script typically completes in 3-5 minutes, but if the org policy retry loop is triggered, it can take up to 8 minutes. Keep the tab active.

TIP: The script automatically saves all output to a log file at `~/cloudasta-migration-<timestamp>.log`. If your Cloud Shell session disconnects or refreshes, reconnect and run `cat ~/cloudasta-migration-*.log` to view the full log. The log file persists in your Cloud Shell home directory.

4. Option B: Running on macOS (Local Terminal)

Step 1: Install Google Cloud SDK (if not installed)

If you do not already have gcloud installed, install it using Homebrew:

```
brew install --cask google-cloud-sdk
```

Alternatively, download from: https://cloud.google.com/sdk/docs/install

Step 2: Authenticate with Google Cloud

Open Terminal and authenticate with your Super Admin Google account:

```
gcloud auth login
```

This opens a browser window. Sign in with your admin account and grant permissions. Once authenticated, return to the Terminal.

Step 3: Clone the Repository and Run

Open Terminal and clone the GitHub repository:

```
git clone https://github.com/hozefa1997/cloudm-gcp-setup.git
```

Navigate to the folder and run the script:

```
cd cloudm-gcp-setup && chmod +x cloudasta-migration-setup.sh &&
./cloudasta-migration-setup.sh
```

Step 4: Follow the Prompts

The script will prompt you for the domain name, admin email, and optional settings. The key file will be saved to the directory you specify (defaults to your current directory).

TIP: On macOS, the script automatically uses the **open** command to launch browser links for manual steps (DWD, Chat API config). It also copies the Client ID to your clipboard using **pbcopy** for easy pasting.

5. Script Prompts Walkthrough

The script is interactive. Here is what you will see at each prompt and what to enter:

Prompt: "Domain name"

Example input: `example.com`

The Google Workspace domain you are setting up for migration (source or destination). Used for naming the key file.

Prompt: "Admin email address"

Example input: `admin@example.com`

The super admin email for this domain. Used for OAuth consent, Chat API visibility, and DWD setup.

Prompt: "GCP Organization ID"

Example input: `(press Enter to skip)`

Optional. Found at GCP Console > IAM & Admin > Settings. Needed if you want the project under a specific org. Also used for auto-resolving the key creation policy error.

Prompt: "Billing Account ID"

Example input: `(press Enter to skip)`

Optional. Required if APIs fail to enable due to missing billing. Found at GCP Console > Billing.

Prompt: "Directory to save key file"

Example input: `(press Enter for current directory)`

Where the JSON key file will be saved. Defaults to the current working directory.

Prompt: "Continue? (y/n)"

Example input: `y`

Confirmation after seeing the configuration summary. Review all values before proceeding.

Output file naming convention: The key file is automatically renamed to `<domainname>-serviceaccount.json`. For example, if the domain is example.com, the file becomes `example.com-serviceaccount.json`. This convention works for both source and destination tenants.

6. Manual Steps After Script Completion

After the script completes, three manual steps remain. The script displays all the values you need and offers to open the relevant pages in your browser.

6A. Domain-Wide Delegation (Google Admin Console)

The script generates a **one-click DWD link** that opens the Google Admin Console with the Client ID and all 36 OAuth scopes pre-populated. Simply click the link and then click **Authorize**.

1. Click the **one-click DWD link** shown by the script in the terminal (the link contains the Client ID and all scopes as URL parameters)
2. The Admin Console DWD page opens with the Client ID and scopes already filled in
3. Click **Authorize**

TIP: If the one-click link does not work (e.g., URL too long for your browser), use the manual fallback: go to Admin Console > Security > API controls > Manage Domain Wide Delegation > Add new, then paste the Client ID and scopes from the DWD helper file (`dwd-setup-<domain>.txt`) saved alongside the key file.

IMPORTANT: The 36 scopes include 6 People API profile scopes (user.addresses.read, user.birthday.read, user.emails.read, user.gender.read, user.organization.read, user.phonenumbers.read) required for CloudM Contacts migration. Missing these scopes causes 'unauthorized_client' errors on Contacts system checks.

6B. Google Chat API Configuration (GCP Console)

1. Open the Chat API configuration link shown by the script
2. Set **App Name** to: **Cloudasta-Migration**
3. Set **Avatar URL** to: the Cloudasta logo URL shown by the script
4. Set **Description** to: **CloudM Migrate** (max 40 characters — do not exceed this limit)
5. **Untick** "Build this Chat app as a Workspace add-on"
6. **Enable** "Log errors to Logging"
7. Under **Functionality**: **Enable** "Join spaces and group conversations"
8. Under **Connection settings**: set connection type to **HTTP endpoint URL**
9. Set **HTTP endpoint URL** to: **https://<domainname>** (use the domain name entered during script execution)
10. Set **Authentication Audience** to: select **Project Number** from the dropdown
11. Under **Visibility**: add admin email address (or a Google Group — see tip below)
12. Under **App Status**: set to **LIVE - available to users**
13. Click **Save**

TIP: Create a Google Group (e.g., migration-users@domain.com) containing all users being migrated, then add that group under Visibility instead of individual emails. This makes the Chat app available to all migrating users at once.

TIP: If CloudM shows an informational message about Chat configuration even after completing these steps, navigate to **mail.google.com** with the admin account and click on the **Chat** section in the left sidebar. This activates Chat for the account and resolves the CloudM system check message.

6C. Enable Drive SDK (Google Admin Console)

1. Go to **https://admin.google.com**
2. Navigate to **Apps > Google Workspace > Drive and Docs**
3. Click **Features and Applications**
4. Enable **Allow users to access Google Drive with the Drive SDK API**

7. Troubleshooting

GCP Terms of Service Not Accepted

Script says: "ACTION REQUIRED: Accept GCP Terms of Service"

This occurs on fresh GCP environments. The script detects this automatically and opens the GCP Console for you. Accept the terms in your browser, then press Enter in the terminal to continue.

Service Account Key Creation is Disabled (Org Policy)

Error: iam.disableServiceAccountKeyCreation policy is enforced

The script will offer to auto-resolve this. It detects the Organization ID, grants you the Organization Policy Administrator role, disables the policy, then retries key creation with a progressive backoff (15s, 30s, 60s, 120s). If all 4 retries fail, wait a few more minutes and run the key creation command shown by the script. After the key is created, the script offers to re-enable the policy for security.

APIs Fail to Enable - Billing Required

Warning: Could not enable - billing account required

Some APIs require a billing account linked to the project. The script shows a direct link to link billing. After linking, either re-run the script with the billing account ID or manually enable the failed APIs in the GCP Console.

Project Creation Quota Exceeded

Error: Project creation quota exceeded

GCP limits each account to approximately 25 projects. Delete unused projects at console.cloud.google.com/cloud-resource-manager. The script also lets you enter an existing project ID instead.

Access Token Expired

Error: Access token expired. Please re-authenticate

This can happen if the script takes a long time (e.g., during org policy retries). Run `gcloud auth login` to re-authenticate, then re-run the script.

Project Not Accessible

Error: Project exists but is not accessible. Check your permissions.

Ensure you are signed in as a Super Admin with permissions on the project. Run `gcloud auth list` to verify the active account.

Cloud Shell Disconnected / Logs Lost

Terminal refreshed and all output is gone

The script automatically saves all output to `~/cloudasta-migration-<timestamp>.log`. Reconnect to Cloud Shell and run `cat ~/cloudasta-migration-*.log` to view the full log. The log file persists in your Cloud Shell home directory even after disconnections.

All APIs Failed to Enable

Warning: 12 of 12 API(s) failed to enable

This usually means billing is not linked or you lack the Editor/Owner role on the project. The script now shows detailed error output per API. Check if errors mention 'billing' or 'PERMISSION_DENIED'. Link a billing account at GCP Console > Billing, then re-run the script or enable APIs manually.

Starting over: If something went wrong and you want a clean slate, delete the project and re-run the script: `gcloud projects delete <project-id>`

8. Quick Reference Card

Essential Commands

Action	Command
Authenticate	`gcloud auth login`
Run the script	`./cloudasta-migration-setup.sh`
Make executable	`chmod +x cloudasta-migration-setup.sh`
List projects	`gcloud projects list`
Delete project	`gcloud projects delete <project-id>`
Check active account	`gcloud auth list`
Manual key creation	`gcloud iam service-accounts keys create key.json --iam-account=SA_EMAIL`
View script logs	`cat ~/cloudasta-migration-*.log`

Key URLs

Page	URL
Google Cloud Console	`https://console.cloud.google.com`
Google Cloud Shell	`https://shell.cloud.google.com`
Admin Console	`https://admin.google.com`
DWD Page	`https://admin.google.com/ac/owl/domainwidedelegation`
Billing	`https://console.cloud.google.com/billing`
Resource Manager	`https://console.cloud.google.com/cloud-resource-manager`

Naming Conventions

Item	Value
GCP Project Name	Cloudasta-Migration
Service Account Name	cloudasta-migration
Chat API App Name	Cloudasta-Migration
Key File Name	`<domainname>-serviceaccount.json`
DWD Helper File	`dwd-setup-<domainname>.txt`