ModelScope × UPH × PCL × MetaC · Campus AI Co-Creation Competition GUIDE

How to Bind Alibaba Cloud, Configure Your AI Coding Tool & Publish on ModelScope Studio

- Bind Alibaba Cloud International → Get Access Token → Configure Claude Code / OpenCode → Code Locally → Publish to Studio → Submit on the Activity Page
- Audience: developers participating in the Prompt to App – AI Application Development Project for the first time

Before You Start

- Register / log in with a ModelScope account at `modelscope.ai` (free — email sign-up).
- **Magic Cubes:** during the activity period every account gets **500 Magic Cubes per day** — **250** from the daily-login bonus plus **250** gifted extra to competition participants. All of them **expire at the end of the same day**.
- Prepare an Alibaba Cloud International account (you can sign up for one during the binding flow if you don't have it yet).
- Install Node.js 18 or later locally — required to install the AI coding tools.
- Have a deployable application project ready (Gradio / Streamlit / static page / Docker all work).
- Prompt to App – AI Application Development Project only approves four models — see the Approved Models table in STEP C and configure one of them.
- Keep your access token safe: never commit it to a public repository.

Workflow Overview

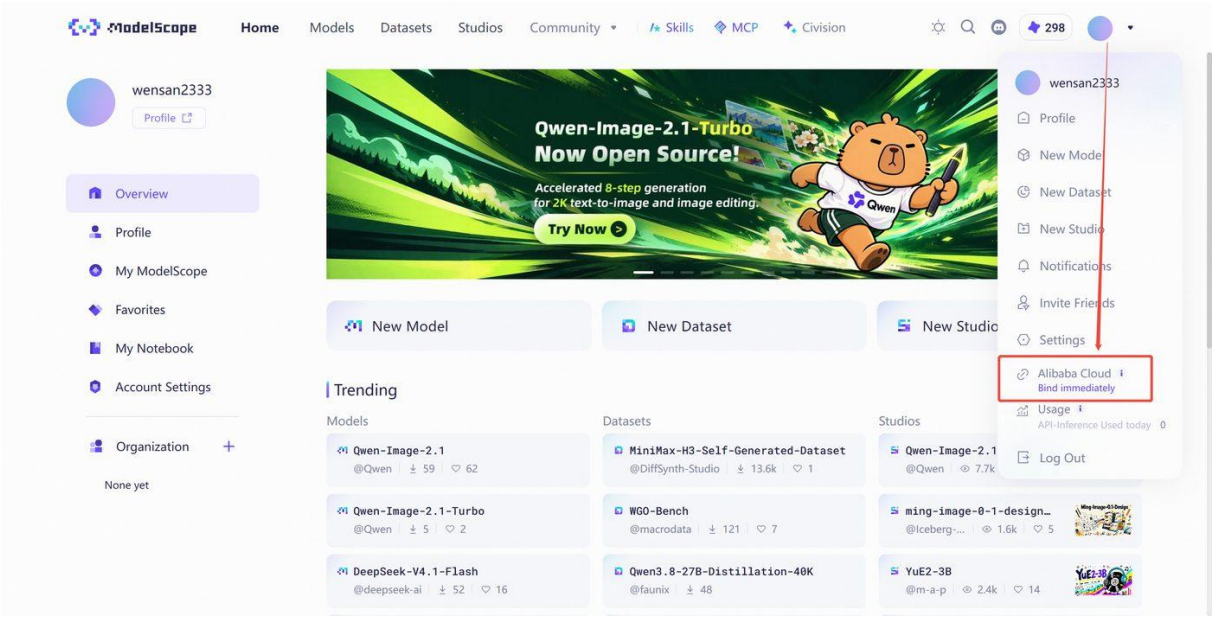
Milestone	Action	Outcome
1. Bind	Bind your Alibaba Cloud International account	Account linked, free cloud resources unlocked
2. Get Token	Get your own access token	Credential for API-Inference calls
3. Configure	Configure Claude Code / Codex / OpenCode	One tool ready with an approved model
4. Code	Code locally with AI	An application that runs correctly locally
5. Publish	Publish to Studio	A public link to your published project
6. Submit	Submit on the competition activity page	Entry submitted with all required materials

STEP A · Bind Your Alibaba Cloud International Account

Where: modelscope.ai — click your avatar in the top-right corner.

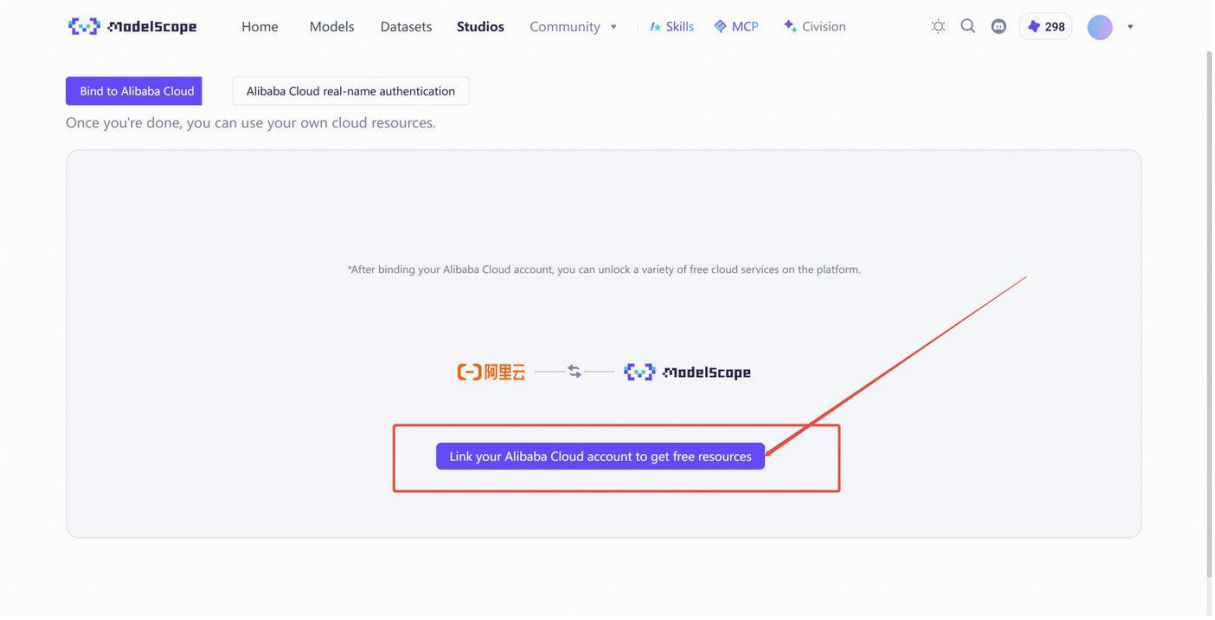
Binding an Alibaba Cloud account is a prerequisite for every later step: once bound, you unlock a range of free cloud resources on ModelScope (including the API-Inference free quota and free CPU resources for Studio).

1. Sign in to ModelScope, click your avatar in the top-right corner, and select "Alibaba Cloud · Bind immediately" in the dropdown menu.



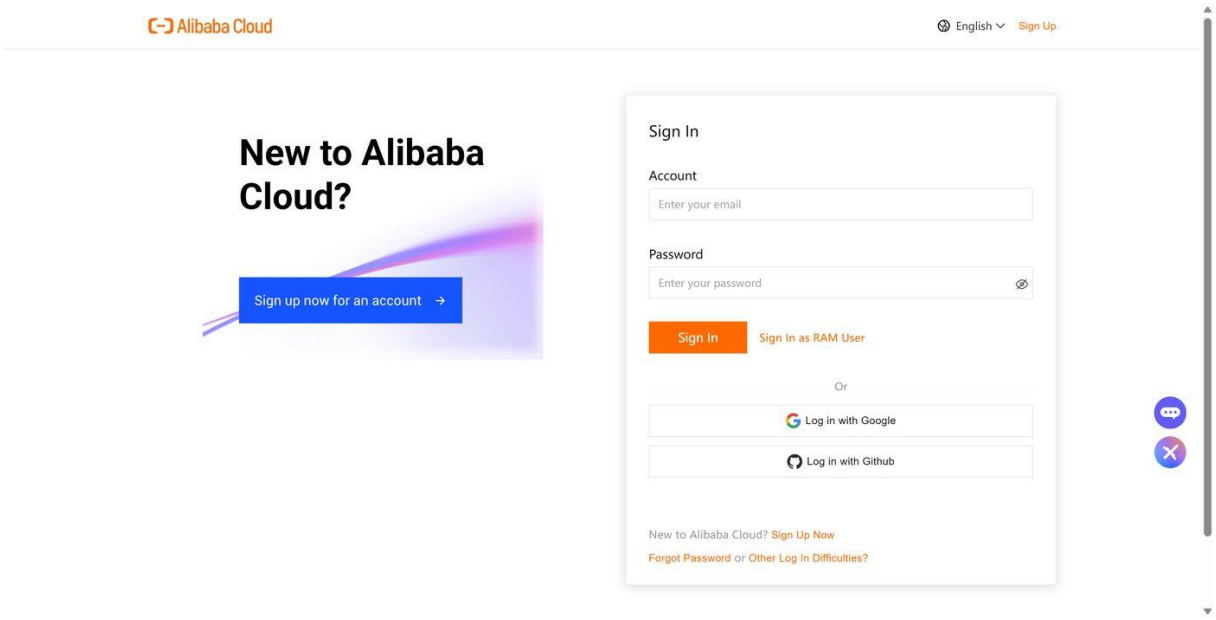
Avatar dropdown — the Alibaba Cloud binding entry

2. On the binding page, click "Link your Alibaba Cloud account to get free resources".



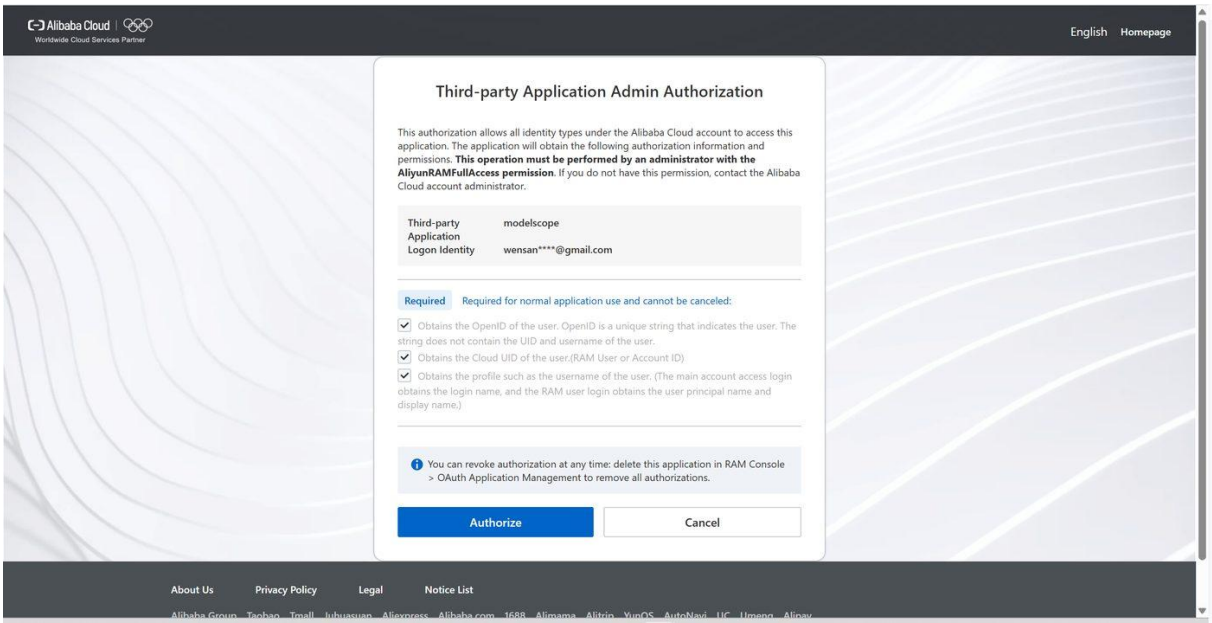
The binding page

3. You will be redirected to the Alibaba Cloud International sign-in page. Sign in directly if you already have an account (email + password, Google, and GitHub sign-in are supported); otherwise click "Sign up now for an account" to register.



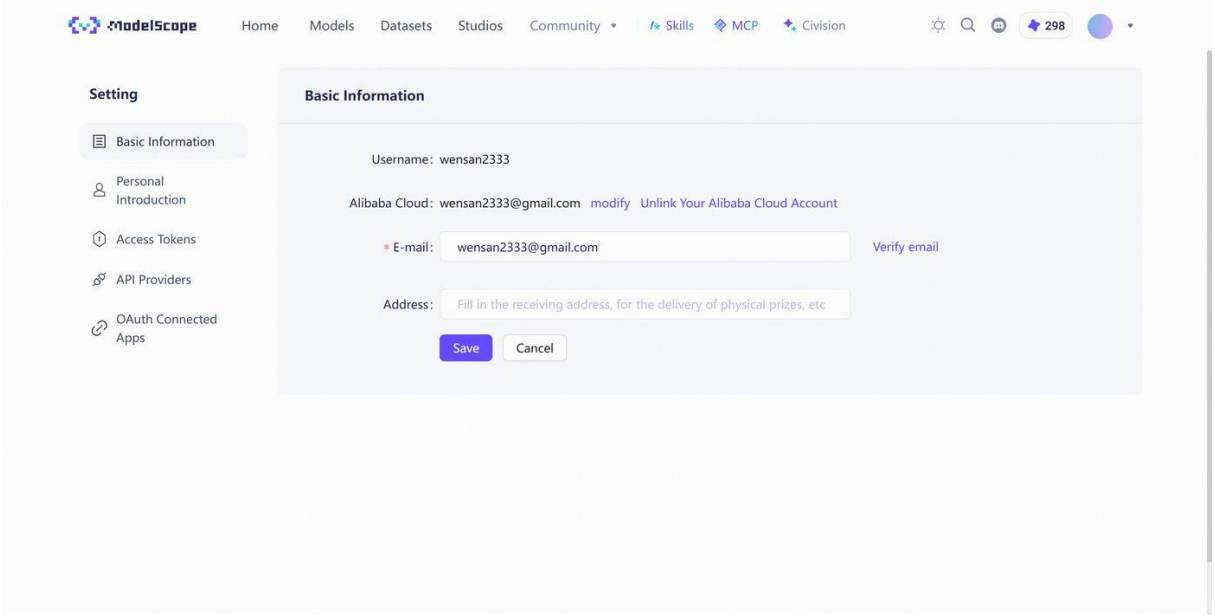
The Alibaba Cloud International sign-in page

4. After signing in, you'll land on the Third-party Application Admin Authorization page. Keep all three permissions (OpenID, UID, Profile) checked and click "Authorize" to complete the authorization.



The Alibaba Cloud authorization page

5. You'll be returned to the ModelScope settings page automatically. If the bound Alibaba Cloud email appears under Basic Information, the binding is complete.



Binding complete — the Alibaba Cloud email appears under Basic Information

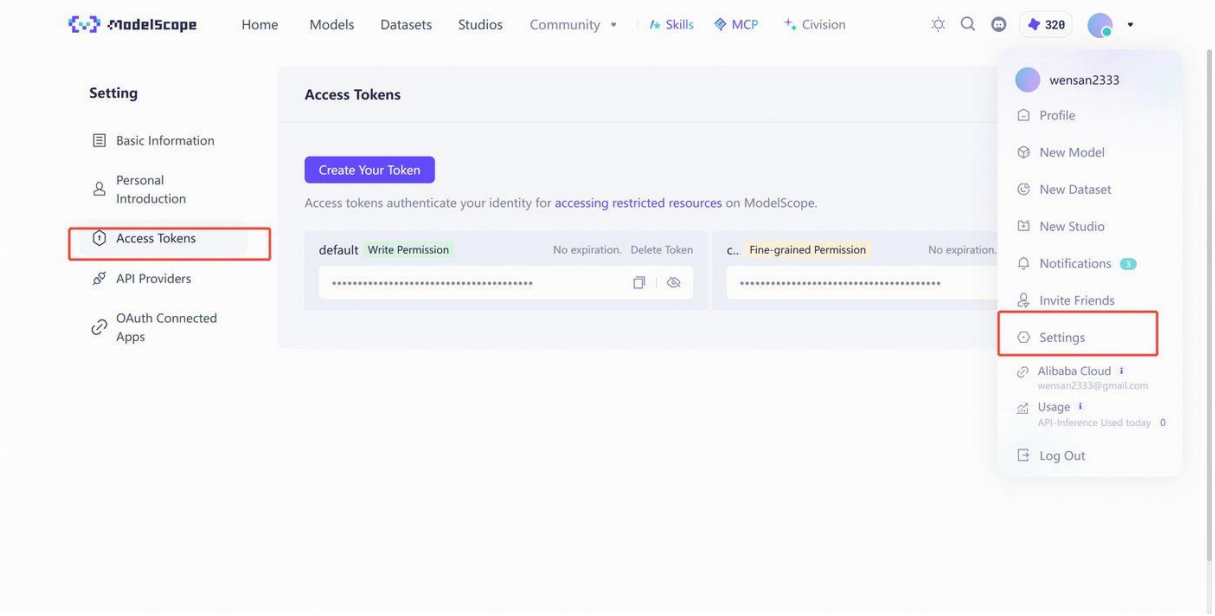
> ⚠ The authorization page states that this action must be performed by an administrator with the AliyunRAMFullAccess permission. If you don't have it, contact your Alibaba Cloud account administrator to complete the authorization.

STEP B · Get Your ModelScope Access Token

Do this right after STEP A — you need your own access token before configuring any coding tool in STEP C.

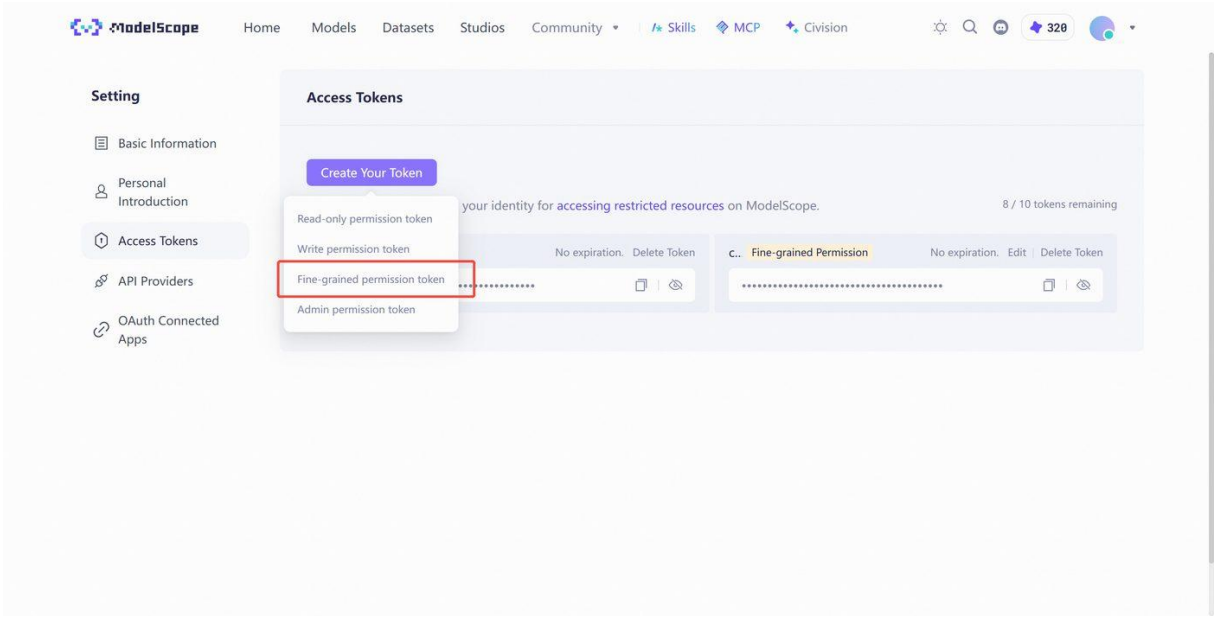
> ⚠ Order matters: bind your Alibaba Cloud International account first (STEP A), then create your own access token here. The coding tools in STEP C cannot authenticate without it.

1. Sign in to ModelScope, click your avatar in the top-right corner and select "Settings".



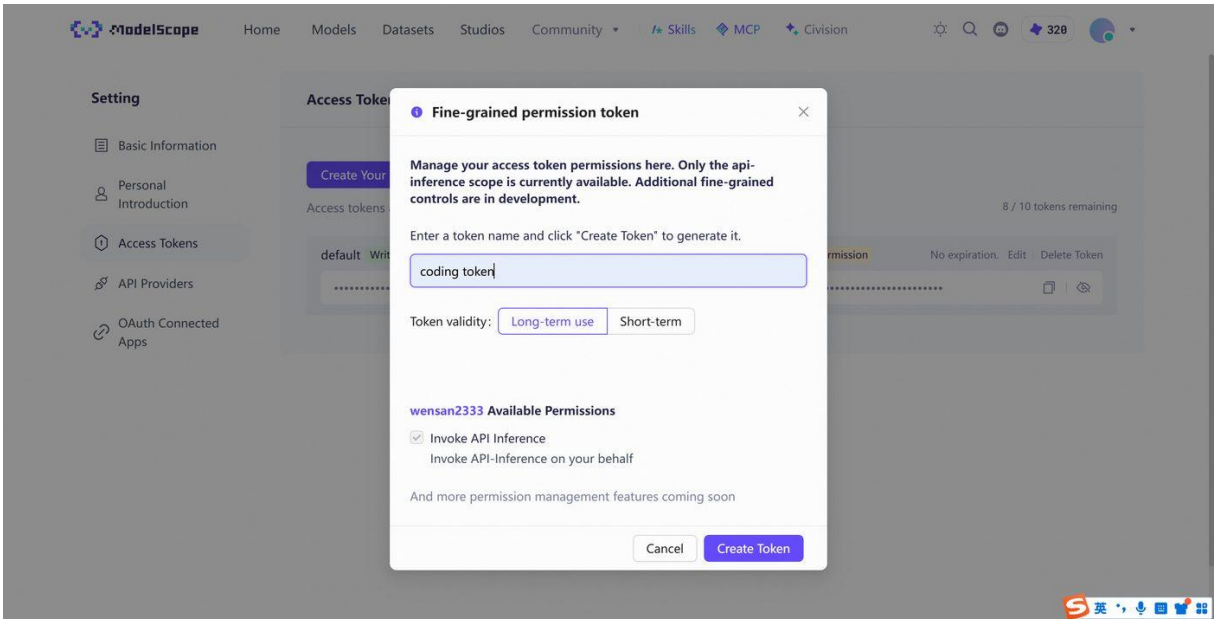
Avatar menu — Settings

2. In the left menu select "Access Tokens", click "Create Your Token", and choose "Fine-grained permission token" from the dropdown.



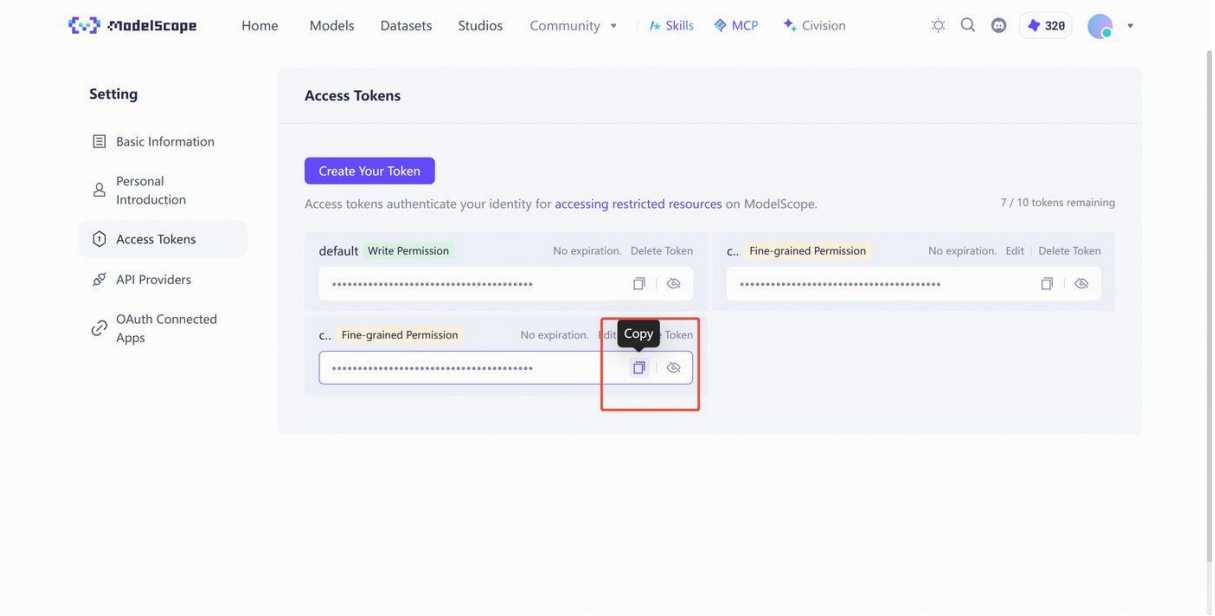
Access Tokens — choose the fine-grained permission token

3. In the popup, enter a token name (e.g. "coding token"), keep "Long-term use" selected, check "Invoke API Inference", then click "Create Token".



The token creation dialog

4. The new token appears in the list — click "Copy" to copy it. This is your access token, referred to below as `your_modelscope_token`.



Copy the token from the list

5. This one token is shared by Claude Code, Codex, and OpenCode. Keep it safe and never commit it to a public repository.

> The access token is the only credential for calling ModelScope API-Inference. If it leaks, delete it on the token management page and create a new one right away.

STEP C · Configure Your AI Coding Tool (Pick One)

There are two tools. Pick one. Both use the same access token from STEP 2 and the same four approved models.

How to open a terminal. Mac: press Command + Space, type Terminal, then press Enter. Windows: press the Windows key, type PowerShell, then press Enter.

About file paths. In this guide, ~ means your home folder. On Mac, this is /Users/your-name. On Windows, this is C:\Users\your-name, and PowerShell calls it $HOME.

If installing a tool fails. Mac: if you see a permission error (EACCES), put sudo at the start of the install command and type your Mac password. Windows: if PowerShell says "running scripts is disabled on this system", run Set-ExecutionPolicy -Scope CurrentUser RemoteSigned, type Y, press Enter, and try again.

Method A · Claude Code

A good choice if you have used Claude Code before.

> This guide assumes you have already installed Claude Code. To check, open a terminal and run claude --version. If you see a version number, Claude Code is installed. If you see an error such as

> "command not found" (Mac) or "is not recognized" (Windows), install it first with npm install -g @anthropic-ai/claude-code, then run the check again.

Claude Code reads its settings from the file ~/.claude/settings.json.

Open the settings file.

On Mac, run these commands in Terminal. The file opens in nano, a simple text editor inside Terminal.

```
mkdir -p ~/.claude
nano ~/.claude/settings.json
```

- `ANTHROPIC_AUTH_TOKEN` — the access token from https://modelscope.ai/my/access/token
- `ANTHROPIC_MODEL` — defaults to zai-org/GLM-5.2; switch between the four approved models by changing this field only: zai-org/GLM-5.2, deepseek-ai/DeepSeek-V4-Pro-0813, stepfun-ai/Step-3.5-Flash, Qwen/Qwen3.8-27B
- `API_TIMEOUT_MS` — relax the timeout to keep long-running tasks from being interrupted

On Windows, run these commands in PowerShell. The file opens in Notepad. If Notepad asks to create a new file, click Yes.

```
mkdir $HOME\.claude -Force
notepad $HOME\.claude\settings.json
```

Add the settings.

Paste the text below into the file. Replace your_modelscope_token with your token from STEP 2. If the file already has other settings, keep them and only add the env part.

```
{
  "env": {
    "ANTHROPIC_AUTH_TOKEN": "your_modelscope_token",
    "ANTHROPIC_BASE_URL": "https://api-inference.modelscope.ai",
    "ANTHROPIC_MODEL": "zai-org/GLM-5.2",
    "API_TIMEOUT_MS": "3000000"
  }
}
```

Save the file.

In nano (Mac): press Control + O, press Enter, then Control + X. In Notepad (Windows): press Ctrl + S, then close Notepad.

Test the setup.

Open a new Terminal (Mac) or PowerShell (Windows) window, type claude, and press Enter. Send a test message, for example "write a Python script that prints Hello World". If you get a reply, the setup works.

Method B · OpenCode

A good choice if you want a simple, light setup.

> This guide assumes you have already installed OpenCode. To check, open a terminal and run opencode --version. If you see a version number, OpenCode is installed. If you see an error such as "command not found" (Mac) or "is not recognized" (Windows), install it first with npm install -g opencode-ai, then run the check again.

OpenCode reads its settings from the file ~/.config/opencode/opencode.json. The settings below already include all four approved models.

Open the settings file.

On Mac, run these commands in Terminal. The file opens in nano, a simple text editor inside Terminal.

```
mkdir -p ~/.config/opencode
nano ~/.config/opencode/opencode.json
```

On Windows, run these commands in PowerShell. The file opens in Notepad. If Notepad asks to create a new file, click Yes.

```
mkdir $HOME\.config\opencode -Force
notepad $HOME\.config\opencode\opencode.json
```

Add the settings.

Paste the text below into the file. Replace your_modelscope_token with your token from STEP 2 and keep the quotes. Keep this file private and do not share it.

```
{
  "$schema": "https://opencode.ai/config.json",
  "model": "modelscope_ai/zai-org/GLM-5.2",
  "provider": {
    "modelscope_ai": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "ModelScope International",
```

```
      "options": {
        "baseURL": "https://api-inference.modelscope.ai/v1",
        "apiKey": "{env:MODELSCOPE_API_KEY}"
      },
      "models": {
        "zai-org/GLM-5.2": {
          "name": "GLM-5.2"
        },
        "deepseek-ai/DeepSeek-V4-Pro-0813": {
          "name": "DeepSeek V4 Pro"
        },
        "stepfun-ai/Step-3.5-Flash": {
          "name": "Step 3.5 Flash"
        },
        "Qwen/Qwen3.8-27B": {
          "name": "Qwen3.8 27B"
        }
      }
    }
  }
}
```

The model line sets the default model (modelscope_ai/zai-org/GLM-5.2). Later, inside OpenCode, you can type /models to switch to any of the four approved models.

Save the file.

In nano (Mac): press Control + O, press Enter, then Control + X. In Notepad (Windows): press Ctrl + S, then close Notepad.

Test the setup.

Open a new Terminal (Mac) or PowerShell (Windows) window, type opencode, and press Enter. Send a test message, for example "write a Python script that prints Hello World". If you get a reply, the setup works.

Approved Models for Prompt to App – AI Application Development Project

Prompt to App – AI Application Development Project only approves the following four models. Pick one when configuring your AI coding tool above.

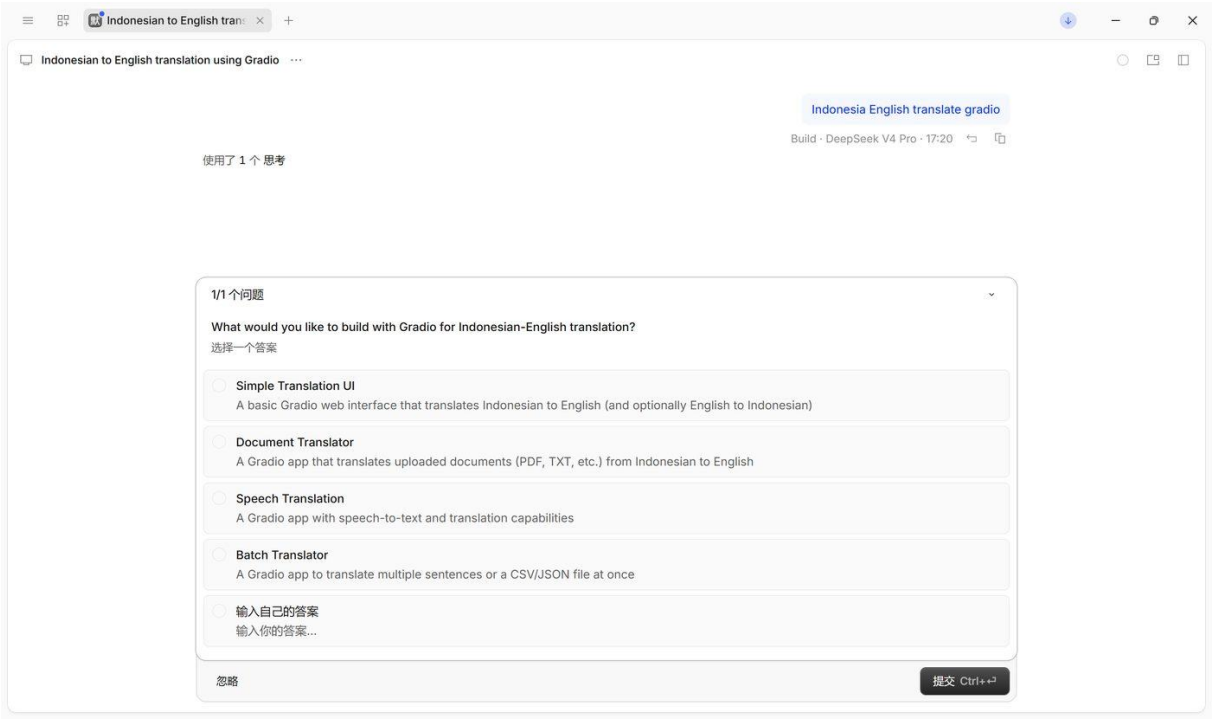
Model	Model ID (use in config)	Model page
GLM-5.2	`zai-org/GLM-5.2`	`modelscope.ai/models/zai-org/GLM-5.2`
DeepSeek-V4-Pro-0813	`deepseek-ai/DeepSeek-V4-Pro-0813`	`modelscope.ai/models/deepseek-ai/DeepSeek-V4-Pro-0813`
Step-3.5-Flash	`stepfun-ai/Step-3.5-Flash`	`modelscope.ai/models/stepfun-ai/Step-3.5-Flash`
Qwen3.8-27B	`Qwen/Qwen3.8-27B`	`modelscope.ai/models/Qwen/Qwen3.8-27B`

> ⚠ Configuring any other model may cause your submission to be rejected.

STEP D · Code Locally

Where: your application project directory.

1. Go to your application project directory and launch the tool you configured (`claude`, `opencode`, or `codex`). The example below uses `opencode`.
2. Send a simple command first to verify connectivity (e.g. "write a script that prints the names of all .py files in the current directory") and confirm you get a normal result.
3. Once connectivity is confirmed, start real development. Note that Studio deployment supports four forms — Gradio, Streamlit, static pages, and Docker — so organize the project around the framework you plan to publish with.



Example — an OpenCode session: the model in use (DeepSeek V4 Pro, one of the four approved models) is shown at the top right, and the agent asks clarifying questions as it builds

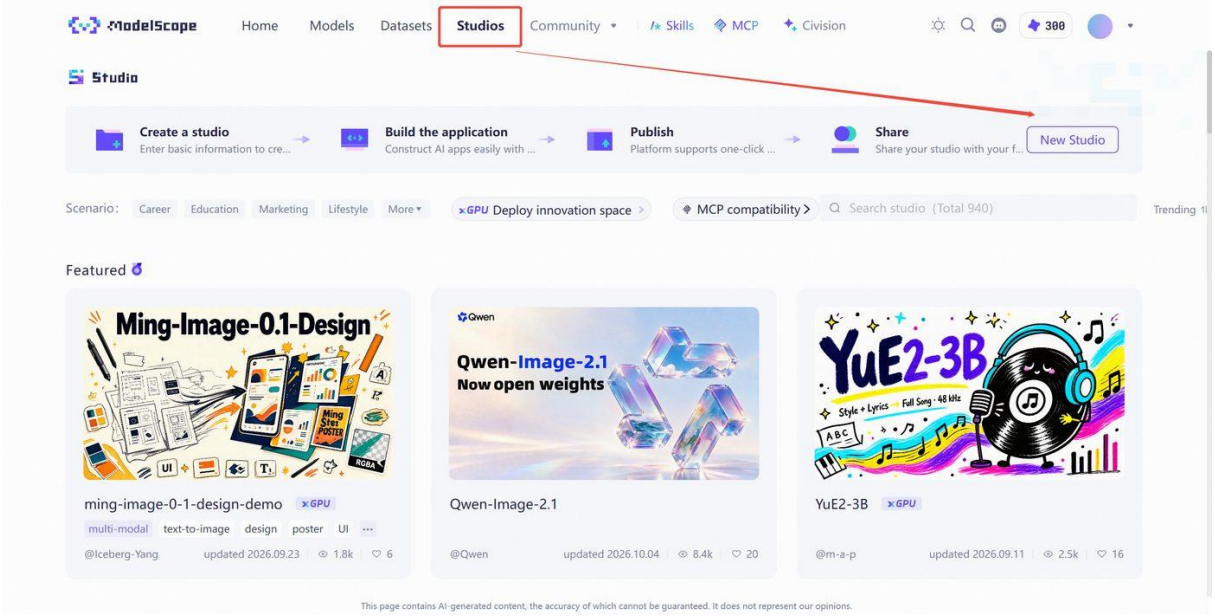
4. Run the app locally to confirm it works, write the dependencies to `requirements.txt`, and get ready to publish.

> 💡 Commit early and commit often. Only files uploaded to Studio will be deployed — keep the project directory clean and self-contained.

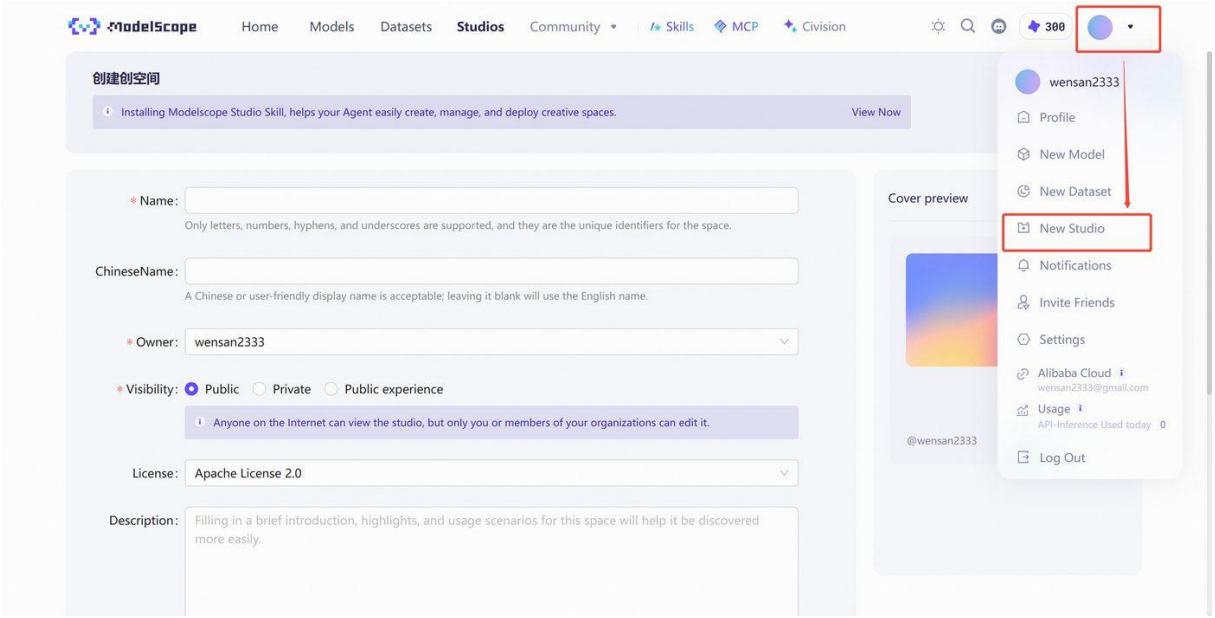
STEP E · Publish Your Project to Studio

Where: modelscope.ai — Studios → New Studio.

1. Go to the creation page: click "Studios" → "New Studio" in the top navigation of ModelScope, or click your avatar in the top-right corner → "New Studio".



The Studios entry in the top navigation



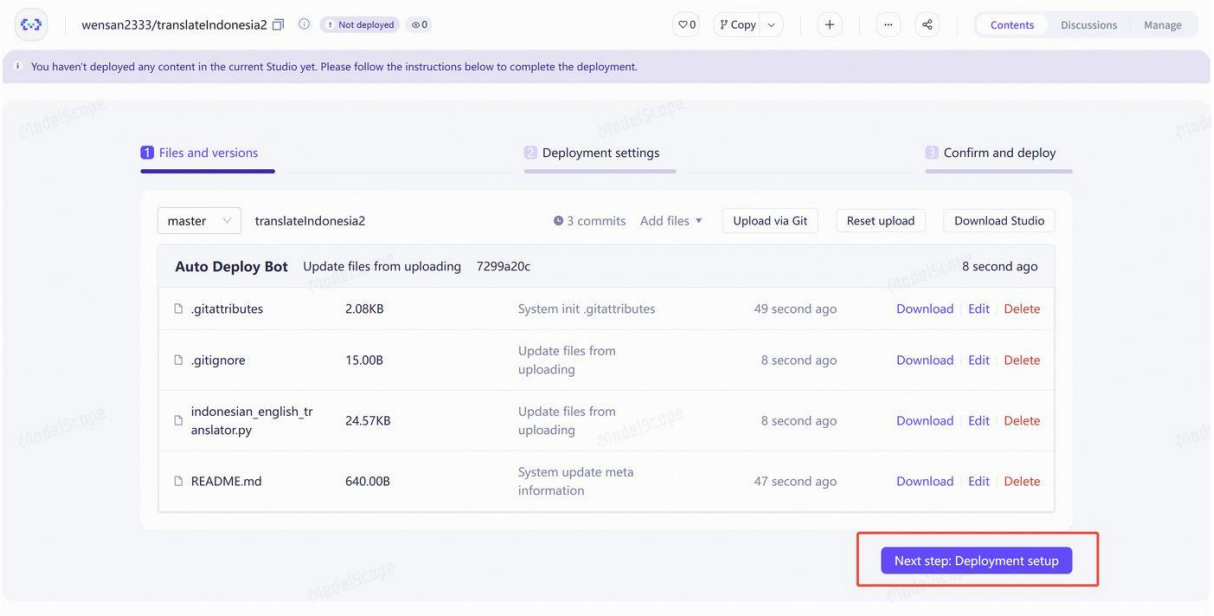
The New Studio creation form

2. Fill in the New Studio form:

   - **Name** — the unique identifier of the space; only letters, numbers, hyphens, and underscores are supported (required).
   - **ChineseName** — a friendly Chinese display name; may be left empty (falls back to Name).
   - **Visibility** — Public (visible to everyone on the internet) / Private (only you or organization members can edit) / Public experience.
   - **License** — Apache License 2.0 by default.

- **Description / Cover Image / Tag** — optional, but they help your project get discovered.Click "Create" when everything looks right.

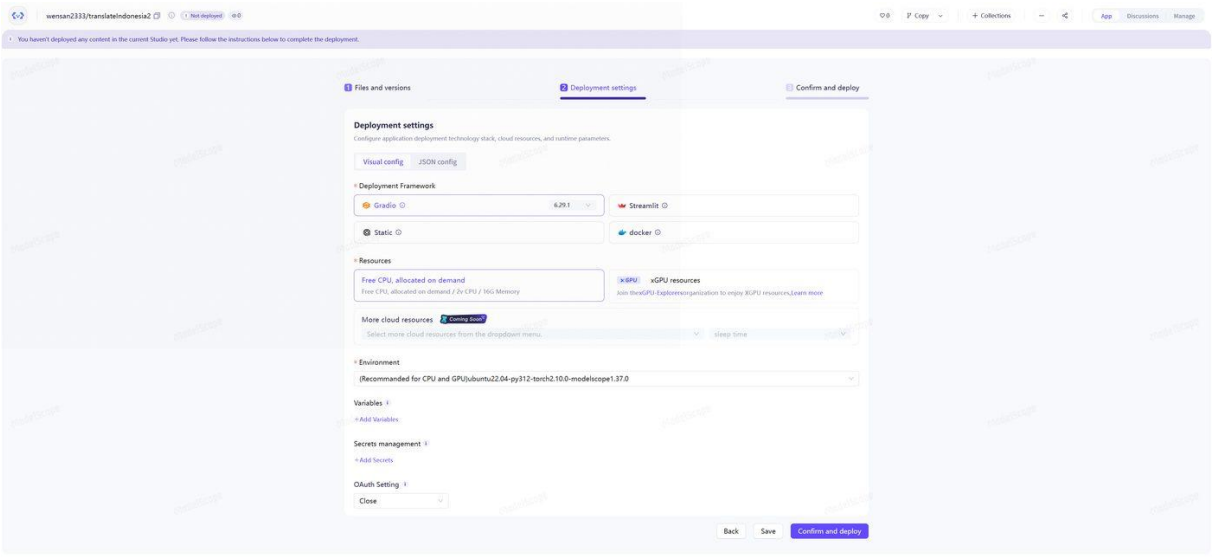
3. Upload files: on the space's Contents page, drag files or the whole project folder into the upload area (.py, .json, .html, etc. are supported), or click "Upload via Git" to import from a Git repository.



Files uploaded on the Contents page

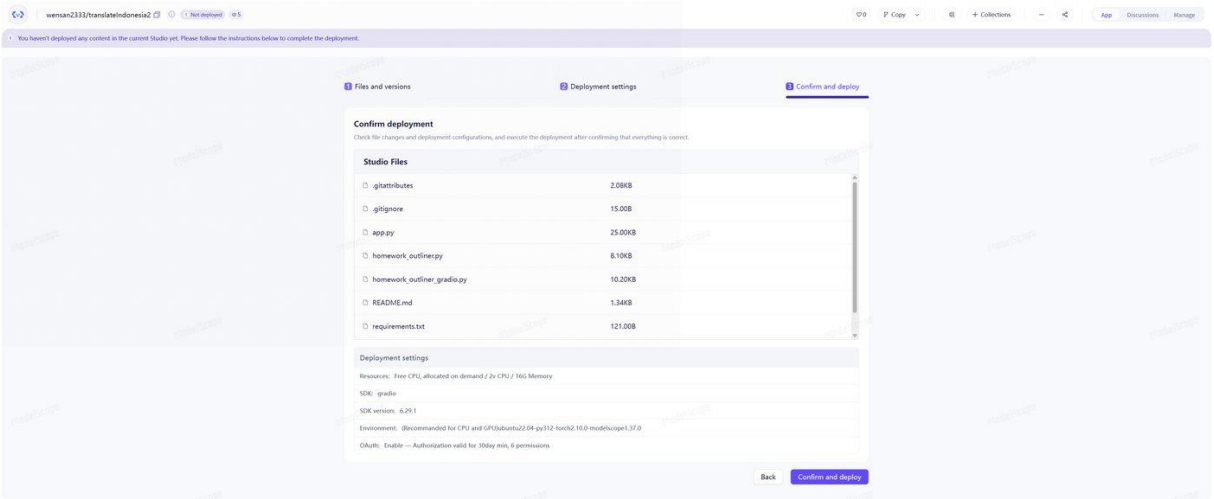
4. Once the files are ready, click "Next step: Deployment setup" to enter the deployment configuration.
5. Configure the deployment parameters:

- **Deployment Framework** — choose Gradio / Streamlit / Static / Docker according to your application type.
- **Resources** — Free CPU allocated on demand (free CPU, allocated on demand) or xGPU resources.
- **Environment** — choose a runtime environment (e.g. Python 3.10).
- **Variables / Secrets / OAuth Setting** — add as needed.Click "Save", then click "Confirm and deploy".



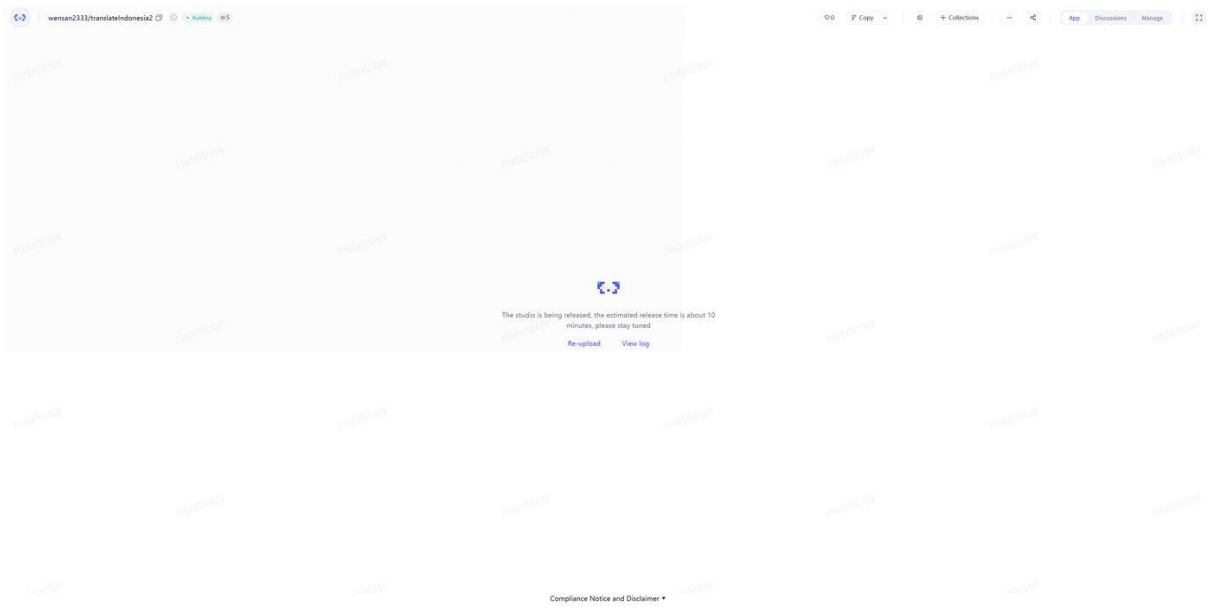
The deployment settings page

6. Confirm the deployment: review the file list and deployment settings (Resources, SDK, Environment, etc.), then click "Confirm and deploy".



Review and confirm the deployment

7. Wait for publishing: the system estimates about 10 minutes to complete; while waiting you can click "View log" to check the deployment logs.



Publishing — about 10 minutes

8. Get the public link: once publishing is complete, the space status turns to running. The URL in your browser's address bar is the public link to your project, in the format `https://modelscope.ai/studios/<username>/<space-name>`. You can also copy the link from the space's share entry and share it with others directly.

> 💡 Keep the Studio public until the competition ends — judges must be able to open the link. Do not delete the space or switch it to private after submitting.

STEP F · Submit on the Competition Activity Page

Where: the official Prompt to App – AI Application Development Project competition activity page — submit your work link here.

1. Open the official competition activity page: `[activity page link – provided by the organizer]`.
2. Find the submission entry (**Submit Works**) and fill in the required information:
   - **Your name, email, school, and contact info** (team members if applicable);
   - **Work title** — identical to the title used on Studio;
   - **Studio share link(s)** — copied from STEP E (multiple links allowed if you published more than one space);
   - **Your coding work** — a zip archive of your project (e.g. the `.py` files), uploaded as an attachment;
   - **Key prompts used for coding** — just the 2–3 most important ones;

- **Key screenshots of the AI coding agent** — they must show the model used, the key prompts, and the tuning / refinement process;
- **Other models used outside the official four?** If yes, list them and upload screenshots of their usage.
- Submit your work by **[4 November] at [16.00 - 23.59] WIB** through the Google Form provided. Please keep a screenshot of your submission confirmation as proof. **Google Form Submission Link:** https://forms.gle/4mTU9bN7a6vxnbji6

⚠ The agent screenshots must clearly show the model name — one of GLM-5.2, DeepSeek-V4-Pro-0813, Step-3.5-Flash, Qwen3.8-27B. Screenshots without a visible model, key prompts, or the refinement process may be counted as incomplete.

💡 Prepare the submission materials while your project is being deployed in STEP E — the zip archive, the conversation-rounds note, the key prompts, and the screenshots can all be collected in advance.

Prompt to App – AI Application Development Project · Student Guide · Only the four approved models (GLM-5.2, DeepSeek-V4-Pro-0813, Step-3.5-Flash, Qwen3.8-27B) are allowed for Prompt to App – AI Application Development Project submissions · Replace the bracketed placeholders with the official activity link and deadline before distribution.