

This is a fantastic conceptual question. It touches on exactly how "Cloud Deployment" actually works under the hood.

You might be thinking: *"If Modal runs the code in the cloud, why does my laptop need the libraries?"*

Here is the simple answer: **Your laptop is the Architect, and Modal is the Construction Site.**

## 1. The "Blueprint" Check

Before your computer sends your code to Modal, it reads through every single file to make sure the "blueprints" make sense.

When it sees this line:

```
from google.oauth2 import id_token
```

Your local Python asks: *"Wait, what is `google.oauth2`? I don't know what that is. Is this a typo? Is this a file you forgot to create?"*

Because it can't find the definition, it panics and crashes **before** it even attempts to send the code to the cloud. It's like trying to email a file that doesn't exist on your desktop—your computer stops you immediately.

## 2. The "Pickle" Process (Technical Reason)

When you run `modal deploy`, Modal doesn't just copy-paste your text files. It uses a Python tool called `cloudpickle`.

- It literally "freezes" your functions and variables into a package.
- To freeze them correctly, it needs to inspect them locally to see what dependencies they have.
- If the library isn't installed locally, it can't inspect it, so it can't package it.

## 3. The Two Environments

You actually have **two** computers involved now:

Environment	Role	Needs Library?	How it gets it
Your Laptop	<b>The Packer:</b> Reads code, checks for errors, packages it up.	YES	You run <code>pip install ...</code>
Modal Cloud	<b>The Runner:</b> Unpacks code, executes it, serves users.	YES	You wrote <code>.pip_install(...)</code> in <code>common.py</code>

## 🚀 The Solution

You need to satisfy your local Python so it lets the code pass through.

Run this locally:

```
pip install google-auth requests
```

Then deploy, and your "Packer" will finally be happy enough to send the suitcase to the "Runner."