Program 3: backup/restore

David Woo

Introduction:

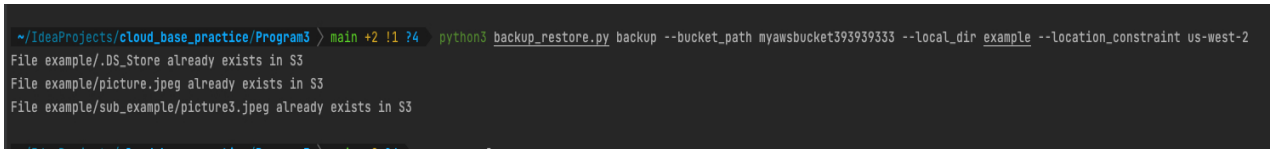
The program is designed to recursively traverse a local directory and back up its files to the cloud. The program also supports restoring from the cloud to a local directory. The cloud storage service used in the program is Amazon S3, and the program uses the Boto3 library to interact with S3.In summary, the program is designed to provide a simple and efficient way to back up and restore files to/from the cloud using Amazon S3.

Design:

The program takes two main arguments: the local directory to backup/restore and the S3 bucket to backup/restore to/from. The program also takes an optional argument to specify the S3 bucket directory where the backup should be stored.

- To backup a local directory to the cloud
  Bash:

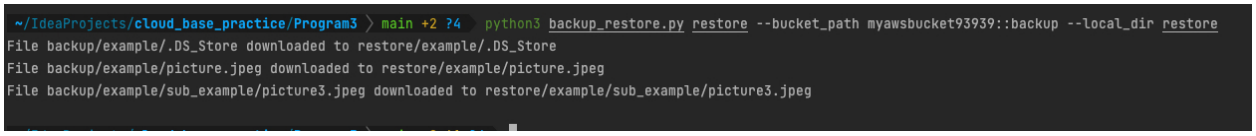
  Python3 backup_restore.py backup –bucket_path [bucket-name]::[directory-name] –local_dir [local-directory-name] –location_constraint [AWS Region]
  - Us-east-1: boto3 doesn't support this region



- To restore cloud storage to local

  Bash:

  Python3 backup_restore.py restore –bucket_path [bucket-name]::[directory-name] –local_dir [local-directory-name]

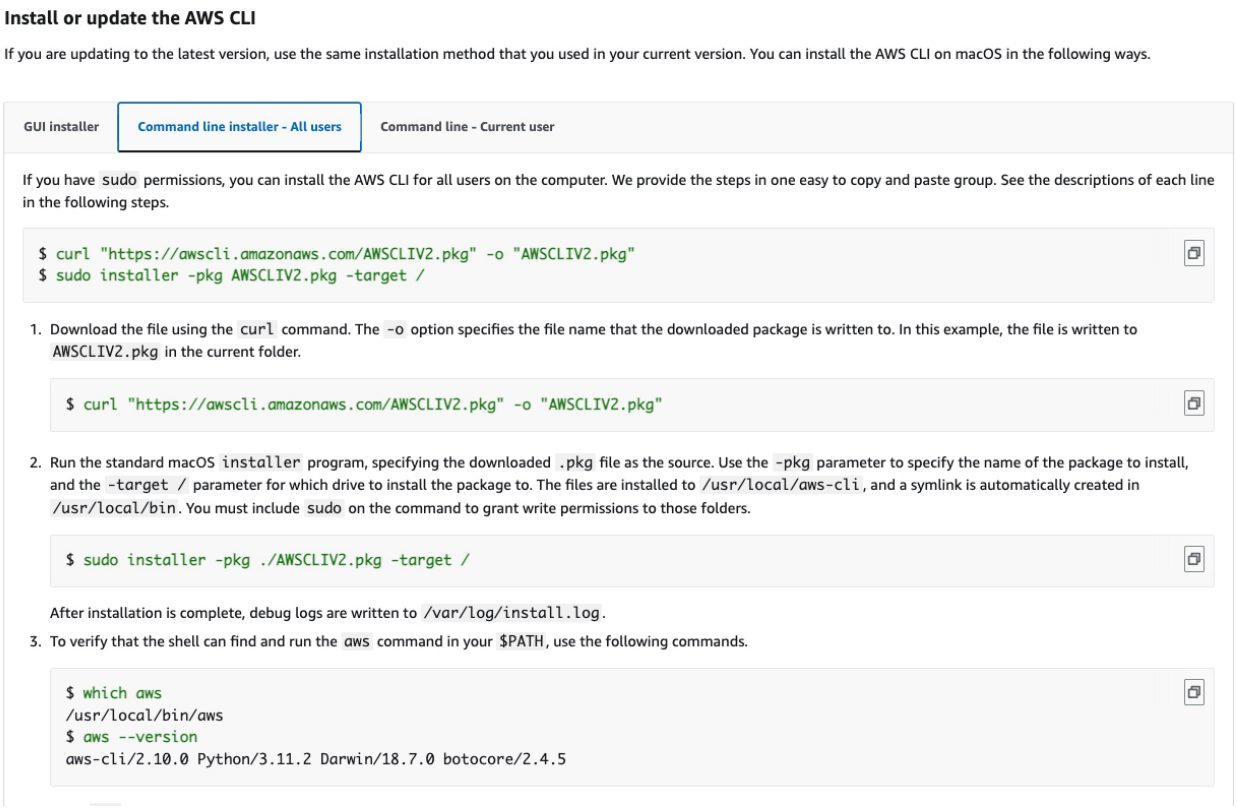


Usage: To use the application, follow these steps:

1. Boto3 set up

   Pip install boto3

2. Install or update the AWS CLI



Install or update the AWS CLI

If you are updating to the latest version, use the same installation method that you used in your current version. You can install the AWS CLI on macOS in the following ways.

GUI installer | Command line installer - All users | Command line - Current user

If you have `sudo` permissions, you can install the AWS CLI for all users on the computer. We provide the steps in one easy to copy and paste group. See the descriptions of each line in the following steps.

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
$ sudo installer -pkg AWSCLIV2.pkg -target /
```

1. Download the file using the `curl` command. The `-o` option specifies the file name that the downloaded package is written to. In this example, the file is written to `AWSCLIV2.pkg` in the current folder.

   ```
   $ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
   ```

2. Run the standard macOS `installer` program, specifying the downloaded `.pkg` file as the source. Use the `-pkg` parameter to specify the name of the package to install, and the `-target /` parameter for which drive to install the package to. The files are installed to `/usr/local/aws-cli`, and a symlink is automatically created in `/usr/local/bin`. You must include `sudo` on the command to grant write permissions to those folders.

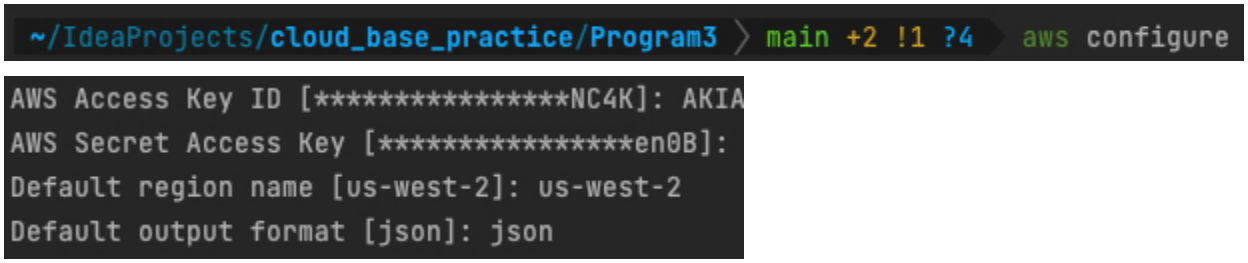
   ```
   $ sudo installer -pkg ./AWSCLIV2.pkg -target /
   ```

   After installation is complete, debug logs are written to `/var/log/install.log`.

3. To verify that the shell can find and run the `aws` command in your `$PATH`, use the following commands.

   ```
   $ which aws
   /usr/local/bin/aws
   $ aws --version
   aws-cli/2.10.0 Python/3.11.2 Darwin/18.7.0 botocore/2.4.5
   ```

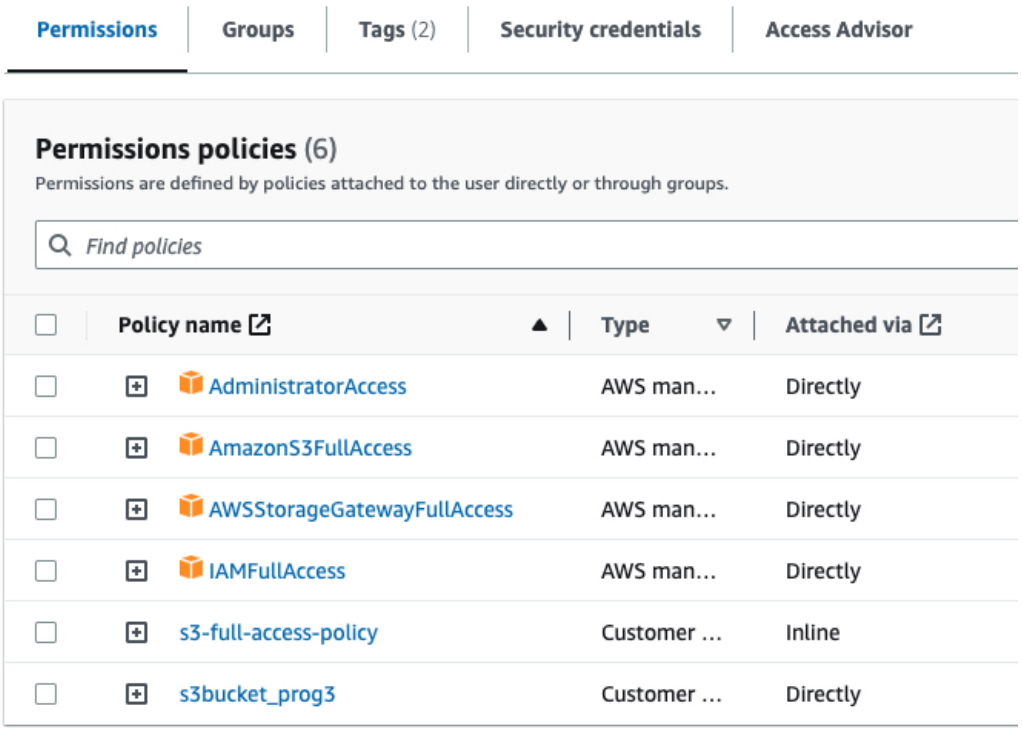
3. Aws configure



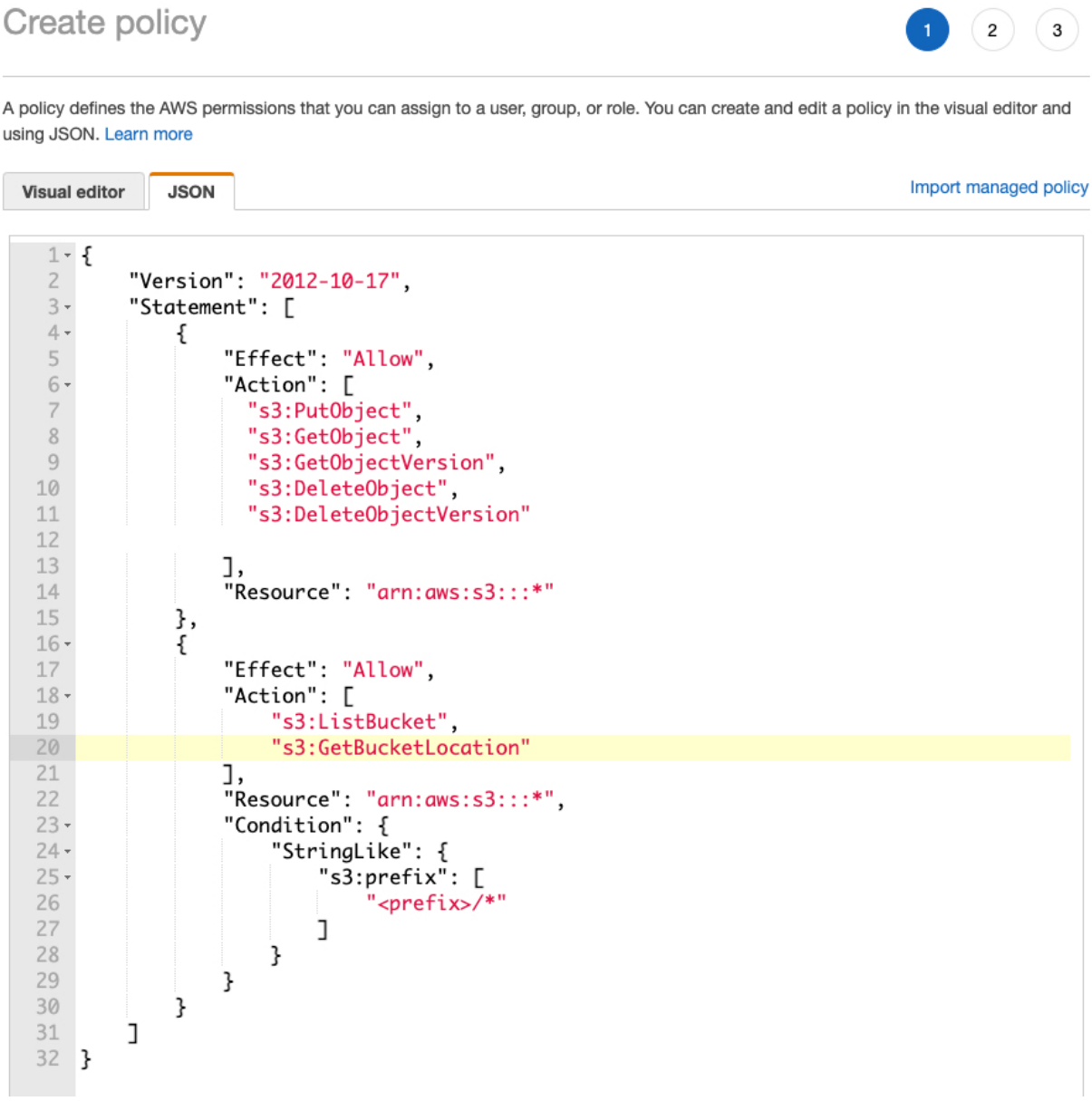
```
~/IdeaProjects/cloud_base_practice/Program3 〉main +2 !1 ?4  aws configure
AWS Access Key ID [****************NC4K]: AKIA
AWS Secret Access Key [****************en0B]:
Default region name [us-west-2]: us-west-2
Default output format [json]: json
```

4. IAM configure

Add all permissions policies

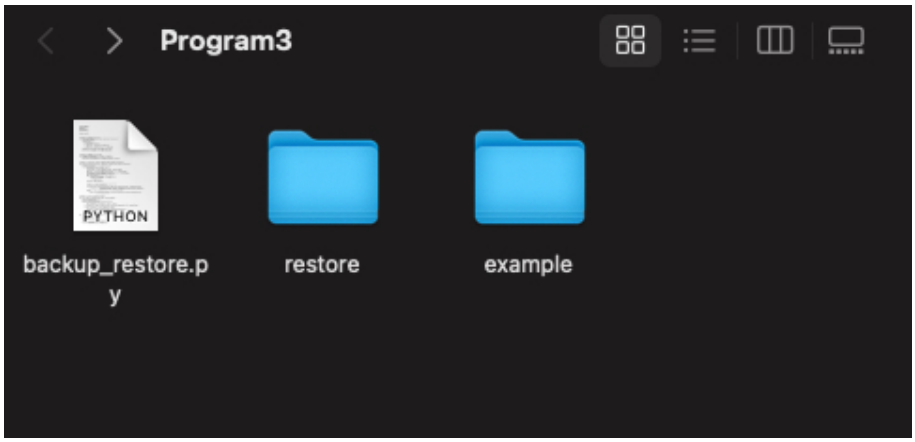


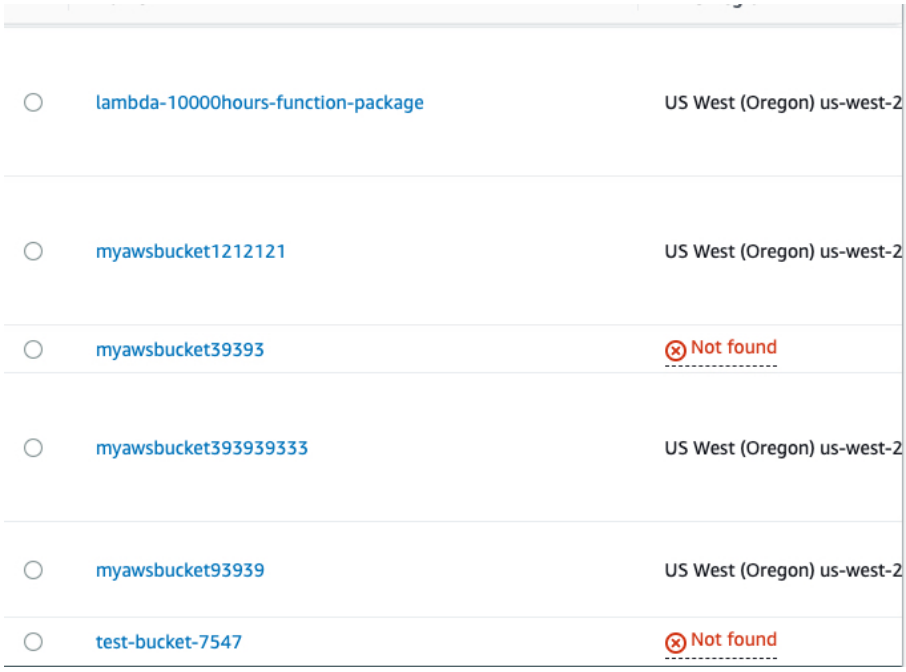
For s3bucket_prog3

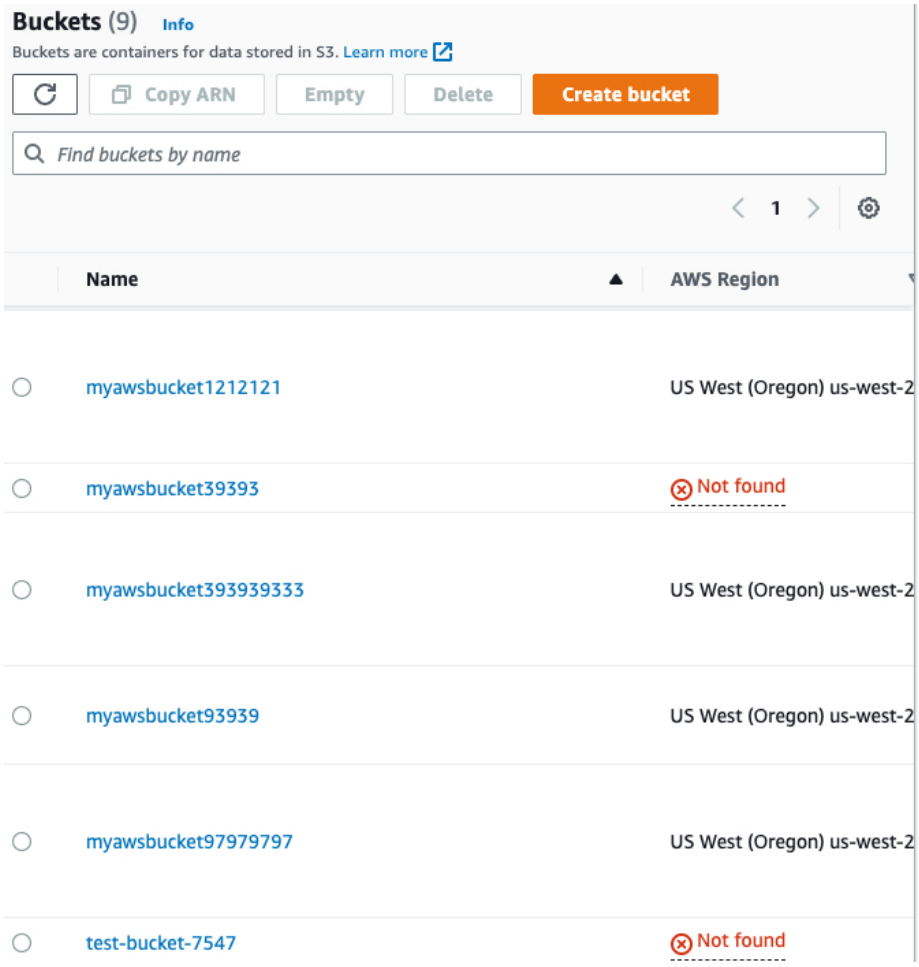


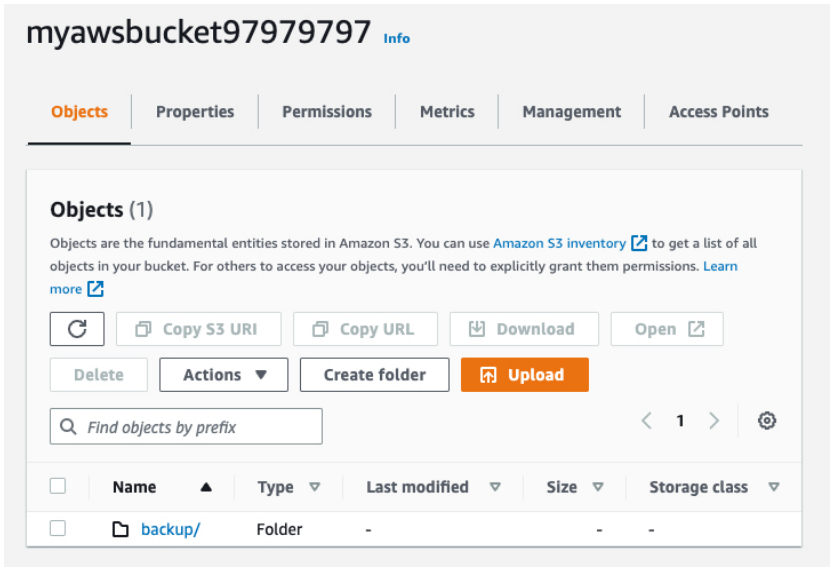
resource:
https://docs.snowflake.com/en/user-guide/data-load-s3-config-aws-iam-role

Result:


Program3
PYTHON
backup_restore.py
restore
example


lambda-10000hours-function-package
US West (Oregon) us-west-2
myawsbucket1212121
US West (Oregon) us-west-2
myawsbucket39393
Not found
myawsbucket393939333
US West (Oregon) us-west-2
myawsbucket93939
US West (Oregon) us-west-2
test-bucket-7547
Not found


Buckets (9) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
myawsbucket1212121
US West (Oregon) us-west-2
myawsbucket39393
Not found
myawsbucket393939333
US West (Oregon) us-west-2
myawsbucket93939
US West (Oregon) us-west-2
myawsbucket97979797
US West (Oregon) us-west-2
test-bucket-7547
Not found


myawsbucket97979797 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
backup/
Folder
-
-
-

- Backup:
    - Initial backup:

```
~/IdeaProjects/cloud_base_practice/Program3 〉main +2 !1 ?4  python3 backup_restore.py backup --bucket_path myawsbucket97979797 --local_dir example --location_constraint us-west-2
File backup/example/.DS_Store uploaded to S3
File picture.jpeg does not exist in the cloud
File backup/example/picture.jpeg uploaded to S3
File picture3.jpeg does not exist in the cloud
File backup/example/sub_example/picture3.jpeg uploaded to S3
```

    - Duplicated backup:

```
~/IdeaProjects/cloud_base_practice/Program3 〉main +2 !1 ?4  python3 backup_restore.py backup --bucket_path myawsbucket97979797 --local_dir example --location_constraint us-west-2
File example/.DS_Store already exists in S3
File example/picture.jpeg already exists in S3
File example/sub_example/picture3.jpeg already exists in S3
```

- Restore:

```
~/IdeaProjects/cloud_base_practice/Program3 〉main +2 ?4  python3 backup_restore.py restore --bucket_path myawsbucket93939::backup --local_dir restore
File backup/example/.DS_Store downloaded to restore/example/.DS_Store
File backup/example/picture.jpeg downloaded to restore/example/picture.jpeg
File backup/example/sub_example/picture3.jpeg downloaded to restore/example/sub_example/picture3.jpeg
```