

Documentation: Simplified Explanation of the DSL Workflow**1. What is the DSL Configuration File (cloudStorage.mp2)?**

The cloudStorage.mp2 file is a simple way to define cloud storage setups for different providers (AWS, Azure, GCP). Think of it as a blueprint where you specify details like the name of the storage, the cloud provider, the region, and access permissions.

Example Configuration:

```
storage "myBucket" {  
    provider = AWS  
    region = "us-west-2"  
    accessControl = Private  
}
```

```
storage "logsContainer" {  
    provider = Azure  
    region = "east-us"  
    accessControl = Public  
}
```

```
storage "backupBucket" {  
    provider = GCP  
    region = "us-central1"  
    accessControl = Private  
}
```

Key Details You Define:

- **Name:** What you want to call the storage (e.g., myBucket).

- **Provider:** Which cloud provider (AWS, Azure, or GCP) the storage belongs to.
 - **Region:** The geographical region for the storage.
 - **AccessControl:** Whether it should be Public or Private.
-

2. What Does the Groovy Code Do?

The Groovy code (DslGeneratorGroovy.groovy) reads the configuration file and turns it into cloud-specific commands that you can run directly in the terminal.

How It Works:

1. **Reads Your Configuration:**
 - It reads each storage block from the file.
 - Extracts details like the name, provider, region, and access type.
2. **Processes Each Storage Setup:**
 - Goes through each storage and figures out what commands are needed based on the provider.
3. **Generates Commands Based on Provider:**
 - **For AWS:**
 - `aws s3api create-bucket --bucket <name> --region <region>`
 - `aws s3api put-bucket-acl --bucket <name> --acl <accessControl>`
 - **For Azure:**
 - `az storage container create --name <name> --public-access <accessControl>`
 - **For GCP:**
 - `gsutil mb -l <region> gs://<name>`
 - `gsutil acl set <accessControl> gs://<name>`
 - **If Provider Is Unknown:** It adds a warning, so you know something's off.
4. **Creates the Final Output:**
 - Combines all the generated commands into one file or output that you can use directly.

Example Output:

```
aws s3api create-bucket --bucket myBucket --region us-west-2
```

```
aws s3api put-bucket-acl --bucket myBucket --acl private
```

```
az storage container create --name logsContainer --public-access public --region east-us
gsutil mb -l us-central1 gs://backupBucket
gsutil acl set private gs://backupBucket
```

3. Why is This Useful?

1. **Saves Time:** Instead of manually writing commands, you define what you need, and the code does the rest.
 2. **Reduces Errors:** Generates accurate commands based on your input, avoiding typos or mistakes.
 3. **Works Across Providers:** Handles AWS, Azure, and GCP without requiring you to know the specifics of each.
 4. **User-Friendly:** Even if you're not a programmer, you can write simple configurations.
 5. **Customizable:** You can easily add more features or providers if needed.
 6. **Consistency:** Ensures all your setups follow the same format and standards.
-

4. How Do You Use It?

- **Step 1:** Write your configurations in the cloudStorage.mp2 file. Example:
 - storage "myBucket" {
 - provider = AWS
 - region = "us-west-2"
 - accessControl = Private
 - }
 - **Step 2:** Run the Groovy script.
 - **Step 3:** Use the generated commands to set up your cloud storage resources.
-

5. Why Should You Care?

This tool simplifies cloud storage management. Instead of memorizing or looking up commands for different platforms, you just describe what you want, and it gives you ready-to-use commands. It's like having a personal assistant for cloud setups!