

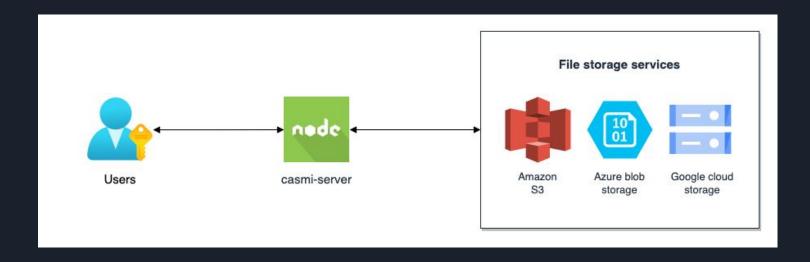
Cloud Agnostic Storage Management Interface

Submission to EXL™ Hackathon

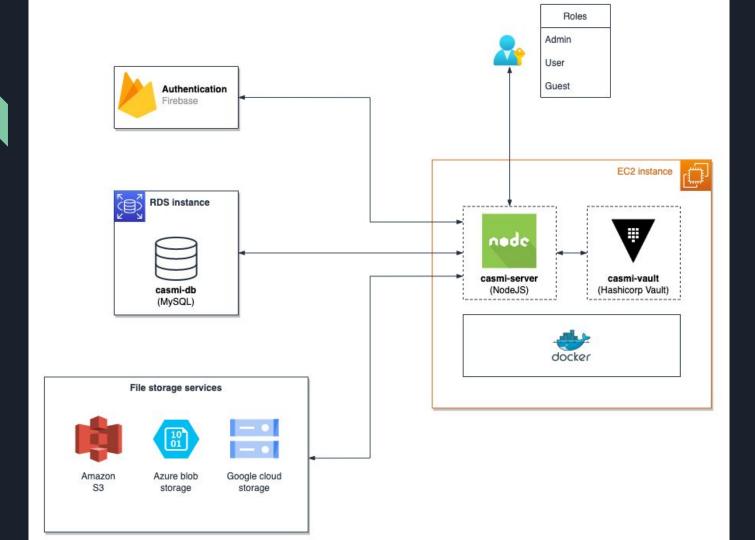
By Rahul Suresh

ABOUT

CASMI (Cloud Agnostic Storage Management Interface) is a NodeJS solution that facilitates cloud-agnostic file management. With access-controlled APIs, the CASMI server offers a standard interface for interacting with file storage services from various cloud providers.



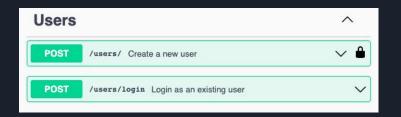
ARCHITECTURE

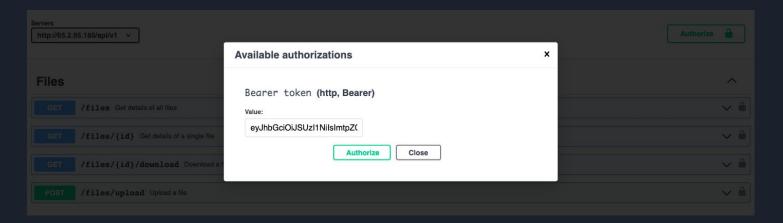


MODULES

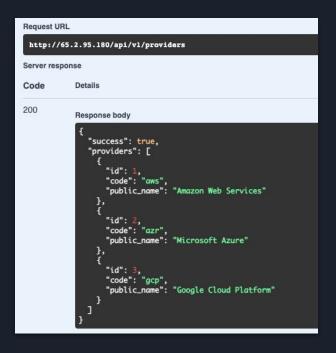
USERS

CASMI uses <u>Firebase Authentication</u> for user authentication and authorization. Every API enforces role-based access, with each user assigned one of the "admin," "user," or "guest" roles.





PROVIDERS



Currently, CASMI supports shared APIs to interact with 3 major cloud providers:







This support can easily be extended to additional cloud service providers, as shown in the next slide.

Adding new providers

1: Create utilities

[Reference]

2: Export utilities

[Reference]

3: Add provider using API

[Reference]

```
provider.utils.js ×

src > providers > utils > s provider.utils.js > ...

1     const createStore = () => {};

2     const getDownloadStream = () => {};

3     const getSignedUrl = () => {};

4     const putFile = () => {};

5     module.exports = { createStore, getDownloadStream, getSignedUrl, putFile };

7
```

```
const provider_utils = require("./utils/provider.utils");

const getUtils = (providerCode) => {
    switch (providerCode) {
        case "": // 3 letter provider code
        return provider_utils;
    }
};

module.exports = { getUtils };
```

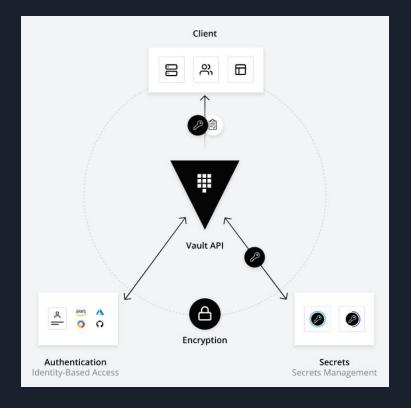
POST /providers Add a new provider

Allowed access: [admin]

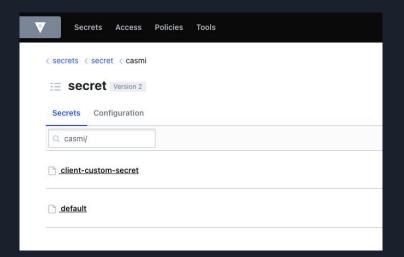
SECRETS

Provider secrets are securely maintained using <u>HashiCorp Vault</u>, which is an identity-based secrets and encryption management system.

"Vault provides encryption services that are gated by authentication and authorization methods. Using Vault's UI, CLI, or HTTP API, access to secrets and other sensitive data can be securely stored and managed, tightly controlled (restricted), and auditable."



SECRETS



The casmi-server is preconfigured to use default provider secrets, but custom secrets can be stored and used from the vault.

```
casmi-kv-policy

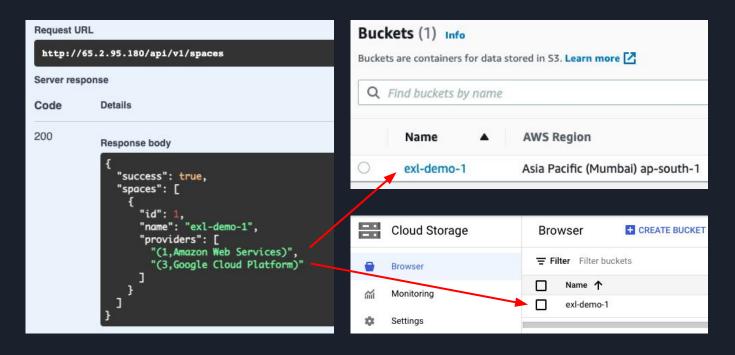
Policy (hcl format)

path "secret/data/casmi/*" {
   capabilities = [ "create", "read" ]
}
```

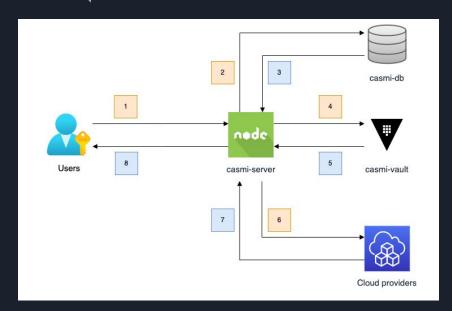
Restricted roles and policies govern who can access casmi-vault and its secrets.

SPACES

Spaces are analogous to projects. They can be configured to use single/multiple cloud providers.



SPACES



```
const getSignedUrl = async (vaultKey, storeName, sourceKey) => {
   try {
      const vaultLoginResponse = await vault.approleLogin(appRoleConfig);
      vault.token = vaultLoginResponse.auth.client_token;
      const { data } = await vault.read(vaultKey);
      const storageClient = getStorageClient(data.data.AWS);
      const params = { Bucket: storeName, Key: sourceKey, Body: "" };
      const command = new GetObjectCommand(params);
      const signedUrl = await getPresignedUrl(storageClient, command, {
            expiresIn: process.env.SIGNED_URL_EXPIRY_SECONDS,
      });
```

Depending on how the space is configured, default OR custom secrets are extracted from the casmi-vault prior to any operations with cloud providers.

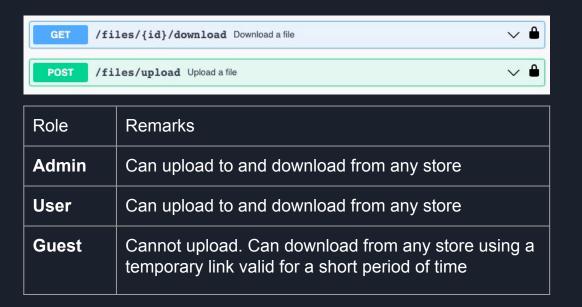
STORES

```
Request URL
 http://65.2.95.180/api/v1/stores
Server response
Code
            Details
200
            Response body
               "success": true,
               "stores": [
                  "id": 1,
                  "file_count": 0,
                   "space_id": 1,
                   "space": "exl-demo-1",
                   "provider_id": 1,
                   "provider": "Amazon Web Services"
                   "id": 2,
                   "file_count": 0,
                   "space_id": 1,
                   "space": "exl-demo-1",
                   "provider_id": 3,
                   "provider": "Google Cloud Platform"
```

Storage is provided by AWS in "buckets," whereas Azure uses "containers." As a standard term for file storage containers used by different providers, CASMI refers to them as "stores."

FOLDERS & FILES

Store items can be either files or folders. Using common protected APIs, files can be downloaded from and uploaded to any CASMI store.



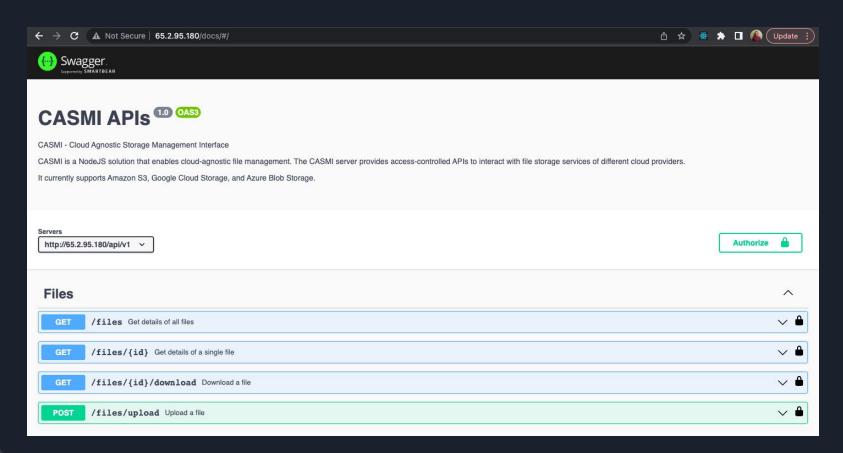
FOLDERS & FILES

The casmi-db keeps track of the entire folder-file structure across providers for quick retrieval.

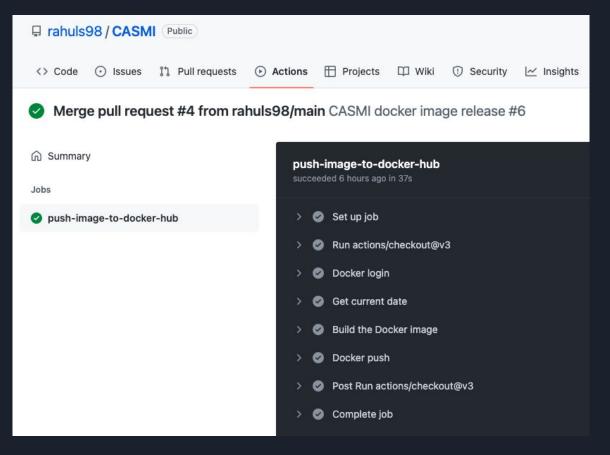
```
Request URL
 http://65.2.95.180/api/v1/folders
Server response
Code
            Details
200
            Response body
                   "id": 3,
                   "provider_key": "/",
                    "store_id": 3,
                   "space_id": 2.
                   "space": "exl-demo-2",
                    "provider_id": 2,
                    "provider": "Microsoft Azure"
                   "provider_key": "/system-design/kafka",
                   "store_id": 2,
                   "space_id": 1,
                   "space": "exl-demo-1",
                   "provider_id": 3,
"provider": "Google Cloud Platform"
                    "provider_key": "/system-design",
                   "store_id": 3,
                   "space_id": 2,
                   "space": "exl-demo-2",
                   "provider_id": 2,
```

```
Request URL
 http://65.2.95.180/api/v1/files
Server response
Code
            Details
200
            Response body
                    "space": "exl-demo-1",
                   "provider_id": 1,
                   "provider": "Amazon Web Services"
                   "name": "kafka-data-streaming.pdf",
                   "provider_key": "system-design/kafka/kafka-data-streaming.pdf",
                   "folder_id": 4,
                   "store_id": 2.
                   "space_id": 1,
                   "space": "exl-demo-1",
                   "provider_id": 3,
                   "provider": "Google Cloud Platform"
                   "name": "frontend-system-design.pdf",
                   "provider_key": "system-design/frontend-system-design.pdf",
                   "folder_id": 5,
                   "store_id": 3,
                   "space_id": 2,
                   "space": "exl-demo-2",
```

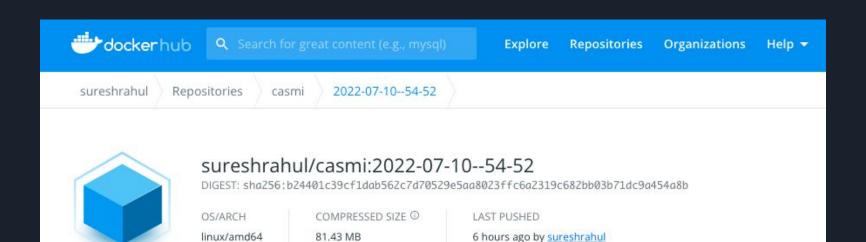
ADDITIONAL FEATURES



Swagger API documentation



Repository set up with Continuous Integration (CI)



Docker container image for quick deployment

THANK YOU