Terraform

In Commander, you can view cloud infrastructure deployed by Terraform.

Terraform allows you to define Infrastructure as Code (IaC) to provision and manage cloud resources, infrastructure, and services.

Backends and state

When resources are deployed, Terraform creates a state file and stores it in a backend. To see deployed resources and their configuration in Commander, you can connect to the following types of backends:

- **Cloud**: State is stored in Terraform Cloud and represents a workspace.
- **Open Source (OSS) - AWS S3**: State is stored in a bucket in Amazon S3.
- **Open Source (OSS) - Azure Blob Storage**: State is stored in a Blob container in Azure Blob storage.

Note: Any JSON file that contains a Terraform version is considered a valid state file and is retrieved from OSS - AWS S3 and OSS - Azure Blob Storage backends.

Commander retrieves the state file from a backend and stores the state data, including raw data from the state file. The state file contains information about the Terraform version and the version of the current state, the deployed resources and their configuration, including all parameters, IDs, dependencies, and outputs.

From Commander, you can view the raw state file and also download the raw JSON output of the state file.

Cloud

For Cloud backends, you can also see the following properties:

- Terraform variables that define the parameters of a workspace configuration.
- Environment variables used by providers for credentials and other data.

Note: Sensitive values are hidden.

OSS - AWS S3

For OSS - AWS S3 backends, you can also see the following properties:

- Name of the AWS region that contains the S3 bucket.
- Name of the S3 bucket.
- File path for the state file, including the name of the state file and the workspace environment.

OSS - Azure Blob Storage

For OSS - Azure Blob Storage backends, you can also see the following properties:

- Name of the container where the state files are stored.
- File path for the bucket's location.

- File path for the state file, including the name of the state file and the workspace environment.

Resources

For each backend state, you can see the collection of infrastructure resources and their configuration, which includes the resource names, the resource types, and the provider. You can select a resource from the list to see the resource block (instances) from the state file.

The frequency at which the backend state is retrieved from Terraform is based on the synchronization period that is set when you add the related Terraform account. To retrieve the latest snapshot of the managed resources, you can manually synchronize with the Terraform backend. For more information, see Synchronize State Backend.

Related topics

- Programmatic Access to Terraform Accounts
- Manage Terraform Accounts
- Synchronize State Backend

Programmatic Access to Terraform Accounts

To manage your Terraform resources in Commander, make sure you have configured programmatic read and write access to the accounts you want to manage.

Terraform Cloud

To manage resources for a Terraform Cloud account, you need:

1. A user account that belongs to a team that has been granted the appropriate permissions to a workspace.
2. An API token.

   For more information, see API Tokens in the documentation for Terraform Cloud.

Terraform OSS - AWS S3

To manage resources for an OSS - AWS S3 account, you need:

1. The name of the S3 bucket where the state files are stored.
2. The region of the bucket’s location (because a single IAM role may have access to multiple bucket locations).
3. An IAM role with a policy that provides access to the S3 bucket with permissions `s3:ListBucket` and `s3:GetObject`.

For more information, see Writing IAM Policies: How to Grant Access to an Amazon S3 Bucket in the documentation for AWS.

Terraform OSS - Azure Blob Storage

To manage resources for an OSS - Azure Blob Storage account, you need:

1. The connection string for the Azure Blob Storage account.
2. The name of the container where the state files are stored.

For more information, see View account access keys in the documentation for Microsoft Azure.

Related topics

- Terraform
- Manage Terraform Accounts
- Synchronize State Backend

Manage Terraform Accounts

To see deployed resources and their configuration in Commander, you can connect to the following types of backends by adding them as accounts:

- **Cloud**: State is stored in Terraform Cloud and represents a workspace.
- **Open Source (OSS) - AWS S3**: State is stored in a bucket in Amazon S3.
- **Open Source (OSS) - Azure Blob Storage**: State is stored in a Blob container in Azure Blob storage.

> **Note:** To manage your Terraform resources in Commander, make sure you have configured programmatic read and write access to the accounts you want to manage. For more information, see Programmatic Access to Terraform Accounts.

Add a Terraform Cloud account

Access:	Views > Inventory
Available to:	Commander Roles of Superuser and Enterprise Admin Administrator Access Rights

1. Select the **Terraform** tab.
2. Select **Manage Accounts**.
3. On the Manage Accounts page, select **Add Account**.
4. In the Add Terraform Account dialog, enter the following:
   a. **Type**: Select **Cloud**.
   b. **Name**: A user friendly name for the account. Used for display purposes only.
   c. **API Token**: The authentication token used to connect to the Terraform Cloud API.
   d. **Update Frequency**: How often to update state data to match the Terraform backend.

      By default, the update frequency is set to 60 minutes.

5. Select **OK**.

Add a Terraform Open Source (OSS) account

Access:	Views > Inventory
Available to:	Commander Roles of Superuser and Enterprise Admin Administrator Access Rights

1. Select the **Terraform** tab.
2. Select **Manage Accounts**.
3. On the Manage Accounts page, click **Add Account**.
4. In the Add Terraform Account dialog, do one of the following:
    a. To connect to an AWS S3 bucket, enter the following:
        - **Type**: Select **OSS - AWS S3**.
        - **Name**: A user friendly name for the account. Used for display purposes only.
        - **Access Key ID**: The access key ID from your AWS credentials.
        - **Secret Access Key**: The secret access key from your AWS credentials.
        - **Region**: The name of the AWS region that contains the S3 bucket. For example, "us_west_2".
        - **Bucket**: The name of the S3 bucket where the state files are stored.
        - **Update Frequency**: How often to update state data to match the Terraform backend.

          By default, the update frequency is set to 60 minutes.

    b. To connect to an Azure Blob Storage account, enter the following:
        - **Type**: Select **OSS - Azure Blob Storage**.
        - **Name**: A user friendly name for the account. Used for display purposes only.
        - **Connection String**: The connection string for the Azure Blob storage account.
        - **Container**: The name of the container where the state files are stored.
        - **Update Frequency**: How often to update state data to match the Terraform backend.

          By default, the update frequency is set to 60 minutes.
5. Select **OK**.

Related topics

- Terraform
- Programmatic Access to Terraform Accounts
- Synchronize State Backend

Synchronize State Backend

Accounts are configured to automatically keep Commander users up-to-date with resource changes performed by Terraform. To retrieve the latest snapshot of the managed resources, you can manually synchronize with the Terraform backend.

Synchronize all backends

To retrieve the latest snapshot of the managed resources for a Terraform account, you can manually synchronize all backends.

Access:	Views > Inventory
Available to:	Commander Role of Superuser and Enterprise Admin Administrator Access Rights

1. Select the **Terraform** tab.
2. Select **Manage Accounts**.
3. From the Manage Accounts page, select an account.
4. Select **Actions**, then **Synchronize**.

Synchronize single backend

To retrieve the latest snapshot of the managed resources for a Terraform backend, you can manually synchronize a single backend.

Access:	Views > Inventory
Available to:	Commander Roles of Superuser and Enterprise Admin Administrator Access Rights

1. Select the **Terraform** tab.
2. On the **Backends** tab, select a backend or multiple.
3. Select **Actions**, then **Synchronize**.

Related topics

- Terraform
- Programmatic Access to Terraform Accounts
- Manage Terraform Accounts