

TERRAFORM FILES

➤ **TERRAFORM FILES & DIRECTORIES:**

FILE EXTENSIONS:

- Code in the Terraform language is stored in plain text files with the **.tf** file extension.
- There is also a **JSON-based variant of the language** that is named with the **.tf.json** file extension.
- Files containing Terraform code are often called **configuration files**.
- Configuration files must always use UTF-8 encoding, and by convention usually use **Unix-style line endings (LF)** rather than **Windows-style line endings (CRLF)**, though both are accepted.

DIRECTORIES AND MODULES:

- A module is a collection of **.tf** and/or **.tf.json** files kept together in a directory.
- A Terraform module only consists of the top-level configuration files in a directory; nested directories are treated as completely separate modules, and are not automatically included in the configuration.
- Terraform evaluates all of the configuration files in a module, effectively treating the entire module as a single document.

ROOT MODULE:

- Terraform always runs in the context of a single *root module*. A complete ***Terraform configuration*** consists of a root module and the tree of child modules.
 - In Terraform CLI, the root module is the working directory where terraform is invoked.
 - In Terraform Cloud and Terraform Enterprise, the root module for a workspace default to the top level of the configuration directory, but the workspace settings can specify a subdirectory to use instead.

OVERRIDE FILES:

- Terraform normally loads all of the **.tf** and **.tf.json** files within a directory and expects each one to define a distinct set of configuration objects. If two files attempt to define the same object, Terraform **returns an error**.
- In some rare cases, it is convenient to be able to override specific portions of an existing configuration object in a separate file.
 - For example, a human-edited configuration file in the Terraform language native syntax could be partially overridden using a programmatically-generated file in JSON syntax.
- For these rare situations, Terraform has special handling of any configuration file whose name ends in **_override.tf** or **_override.tf.json**. This special handling also applies to a file named literally **override.tf** or **override.tf.json**.

DEPENDENCY LOCK FILE:

- A Terraform configuration may refer to two different kinds of external dependency that come from outside of its own codebase:
 - **Providers**, which are plugins for Terraform that extend it with support for interacting with various external systems.
 - **Modules**, which allow splitting out groups of Terraform configuration constructs (written in the Terraform language) into reusable abstractions.
- Both of these dependency types can be published and updated independently from Terraform itself and from the configurations that depend on them. For that reason, Terraform must determine which versions of those dependencies are potentially compatible with the current configuration and which versions are currently selected for use.

LOCK FILE LOCATION:

- The dependency lock file is a file that belongs to the configuration as a whole, rather than to each separate module in the configuration. For that reason Terraform creates it and expects to find it in your current working directory when you run Terraform, which is also the directory containing the **.tf** files for the **root module** of your configuration.

- The lock file is always named **.terraform.lock.hcl**, and this name is intended to signify that it is a lock file for various items that Terraform caches in the **.terraform** subdirectory of your working directory.
- Terraform automatically creates or updates the dependency lock file each time you run the **terraform init** command.

TEST FILES:

- Specific Terraform commands, such as **test**, **init**, and **validate**, load Terraform test files for your configuration.

FILE EXTENSION:

- use the file extensions **.tftest.hcl** and **.tftest.json**.
- Terraform loads all test files within your root configuration directory.