**Learn about IaC**

* [Infrastructure as Code](https://www.hashicorp.com/resources/what-is-infrastructure-as-code) introduction video
* [Infrastructure as Code in a Private or Public Cloud](https://www.hashicorp.com/blog/infrastructure-as-code-in-a-private-or-public-cloud/) blog post
* [Introduction to IaC](https://www.terraform.io/intro/index.html#infrastructure-as-code) documentation
* [Terraform Use Cases](https://www.terraform.io/intro/use-cases.html#multi-cloud-deployment) documentation

## Manage infrastructure

* [Get Started with Terraform](https://learn.hashicorp.com/collections/terraform/aws-get-started) (complete all tutorials)

Study these additional resources to learn more about Terraform key concepts, like providers, provisioners, and state.

* [Providers](https://www.terraform.io/docs/language/providers/configuration.html) documentation
* [Purpose of Terraform State](https://www.terraform.io/docs/language/state/purpose.html) documentation
* [Terraform Settings](https://www.terraform.io/docs/language/settings/index.html) documentation
* [Provision Infrastructure Deployed with Terraform](https://learn.hashicorp.com/collections/terraform/provision) Learn tutorials
* [Provisioners](https://www.terraform.io/docs/language/resources/provisioners/syntax.html#provisioners-are-a-last-resort) documentation
* [Manage Resources in Terraform State](https://learn.hashicorp.com/tutorials/terraform/state-cli) Learn tutorial
* [Use Refresh-Only Mode to Sync Terraform State](https://learn.hashicorp.com/tutorials/terraform/refresh) Learn tutorial
* [Lock and Upgrade Provider Versions](https://learn.hashicorp.com/tutorials/terraform/provider-versioning) Learn tutorial
* [Perform CRUD Operations with Providers](https://learn.hashicorp.com/tutorials/terraform/provider-use) Learn tutorial

## Master the workflow

* [The Core Terraform Workflow](https://www.terraform.io/guides/core-workflow.html) documentation
* [Initialize a Terraform working directory with init](https://www.terraform.io/docs/cli/commands/init.html) documentation
* [Validate a Terraform configuration with validate](https://www.terraform.io/docs/cli/commands/validate.html) documentation
* [Generate and review an execution plan for Terraform with plan](https://www.terraform.io/docs/cli/commands/plan.html) documentation
* [Execute changes to infrastructure with Terraform with apply](https://www.terraform.io/docs/cli/commands/apply.html) documentation
* [Destroy Terraform managed infrastructure with destroy](https://www.terraform.io/docs/cli/commands/destroy.html) documentation

## Learn more subcommands

* [Troubleshoot Terraform](https://learn.hashicorp.com/tutorials/terraform/troubleshooting-workflow#format-the-configuration) Learn tutorial
* [Formatting configuration with fmt](https://www.terraform.io/docs/cli/commands/fmt.html) documentation
* [Tainting resources with taint](https://www.terraform.io/docs/cli/commands/taint.html) documentation
* [Managing state with state](https://www.terraform.io/docs/cli/commands/state/index.html) documentation
* [Using local workspaces with workspace](https://www.terraform.io/docs/language/state/workspaces.html) documentation
* [Refactor Monolithic Terraform Configuration](https://learn.hashicorp.com/tutorials/terraform/organize-configuration#create-a-dev-workspace) Learn tutorial
* [Importing existing resources with import](https://www.terraform.io/docs/cli/commands/import.html) documentation
* [Import Terraform Configuration](https://learn.hashicorp.com/tutorials/terraform/state-import) Learn tutorial

## Use and create modules

* [Organize Configuration with Modules](https://learn.hashicorp.com/collections/terraform/modules) (complete all tutorials)

Review these additional resources to learn about modules and associated concepts in more depth.

* [Finding and using modules](https://www.terraform.io/docs/registry/modules/use.html) documentation
* [Module versioning](https://www.terraform.io/docs/language/modules/syntax.html#version) documentation
* [Input Variables](https://www.terraform.io/docs/language/values/variables.html) documentation
* [Input Variables](https://learn.hashicorp.com/tutorials/terraform/variables) Learn tutorial
* [Output Values](https://www.terraform.io/docs/language/values/outputs.html) documentation
* [Output Values](https://learn.hashicorp.com/tutorials/terraform/outputs) Learn tutorial
* [Calling a child module](https://www.terraform.io/docs/language/modules/syntax.html) documentation

## Read and write configuration

Terraform uses its own configuration language to determine the goal state for the infrastructure it manages. The below resources describe some of the features of Terraform's configuration language.

* [Resources](https://www.terraform.io/docs/language/resources/index.html) describe infrastructure objects
* [Data sources](https://www.terraform.io/docs/language/data-sources/index.html) let Terraform fetch and compute data
* [Query Data Sources](https://learn.hashicorp.com/tutorials/terraform/data-sources) Learn tutorial guides you through using data sources
* [Resource addressing](https://www.terraform.io/docs/cli/state/resource-addressing.html) lets you refer to specific resources
* [Named values](https://www.terraform.io/docs/language/expressions/references.html) let you reference values
* [Create Resource Dependencies](https://learn.hashicorp.com/tutorials/terraform/dependencies) Learn tutorial guides you through managing related infrastructure using implicit and explicit dependencies
* [Terraforms Resource Graph](https://www.terraform.io/docs/internals/graph.html) ensures proper order of operations
* [Complex types](https://www.terraform.io/docs/language/expressions/type-constraints.html#complex-types) let you validate user-provided values
* [Built in functions](https://www.terraform.io/docs/language/functions/index.html) help transform and combine values
* [Perform Dynamic Operations with Functions](https://learn.hashicorp.com/tutorials/terraform/functions) Learn tutorial walks you through using Terraform functions
* [Create Dynamic Expressions](https://learn.hashicorp.com/tutorials/terraform/expressions) Learn tutorial shows you how to create more complex expressions
* [Dynamic blocks](https://www.terraform.io/docs/language/expressions/dynamic-blocks.html) allow you to construct nested expressions within certain configuration blocks

Because Terraform manages your infrastructure, it sometimes needs access to sensitive data. You can inject sensitive data into Terraform configuration using Vault. Review the following resources to learn about secure secrets injection.

* [Inject secrets into Terraform using the Vault provider](https://learn.hashicorp.com/tutorials/terraform/secrets-vault) Learn tutorial
* [Vault Provider for Terraform](https://www.terraform.io/docs/providers/vault/index.html) documentation

## Manage state

Terraform uses state to keep track of the infrastructure it manages. To use Terraform effectively, you have to keep your state accurate and secure. Read the following resources to learn about managing Terraform's state and storage backends.

State management:

* [State locking](https://www.terraform.io/docs/language/state/locking.html) documentation
* [Sensitive data in state](https://www.terraform.io/docs/language/state/sensitive-data.html) documentation
* [Reconcile state and real resources with refresh](https://www.terraform.io/docs/cli/commands/refresh.html) documentation
* [Manage Resource Drift](https://learn.hashicorp.com/tutorials/terraform/resource-drift) Learn tutorial

Backend management:

* [Login to Terraform Cloud from the CLI](https://learn.hashicorp.com/tutorials/terraform/cloud-login) Learn tutorial
* [Backends overview](https://www.terraform.io/docs/language/settings/backends/index.html) documentation
* [Local backend](https://www.terraform.io/docs/language/settings/backends/local.html) documentation
* [Backend types](https://www.terraform.io/docs/language/settings/backends/index.html) documentation
* [How to configure a backend](https://www.terraform.io/docs/language/settings/backends/configuration.html) documentation
* [Migrate State to Terraform Cloud](https://learn.hashicorp.com/tutorials/terraform/cloud-migrate#set-up-the-remote-backend) Learn tutorial

## Debug in Terraform

Terraform has a verbose logging mode to help you debug. Review the [debugging documentation](https://www.terraform.io/docs/internals/debugging.html), which will help you demonstrate that you know when, why, and how to enable verbose logging

## Understand Terraform Cloud and Enterprise

The Terraform CLI focuses on solving the technical challenges of managing IaC. When you collaborate with a team on Infrastructure as Code, new organizational challenges come up. Terraform Cloud and Enterprise focus on solving these organizational challenges. The below resources will help you understand when you would want to consider using Terraform Cloud or Enterprise, and the problems they solve.

* [Terraform Cloud overview](https://www.terraform.io/docs/cloud/index.html) documentation
* [Understanding Workspaces and Modules](https://www.hashicorp.com/resources/terraform-enterprise-understanding-workspaces-and-modules) resource
* [CLI workspaces](https://www.terraform.io/docs/language/state/workspaces.html) documentation
* [The UI- and VCS-driven Run Workflow](https://www.terraform.io/docs/cloud/run/ui.html) documentation
* [Terraform Cloud workspaces documentation](https://www.terraform.io/docs/cloud/workspaces/index.html) documentation
* [Use Modules from the Registry](https://learn.hashicorp.com/tutorials/terraform/module-use) Learn tutorial
* [Module registry](https://www.terraform.io/docs/cloud/registry/index.html) documentation
* [Install the Sentinel CLI](https://learn.hashicorp.com/tutorials/terraform/sentinel-install) Learn tutorial
* [Sentinel Policy as Code](https://www.terraform.io/docs/cloud/sentinel/index.html) documentation
* [Feature comparison](https://www.hashicorp.com/products/terraform/pricing/) pricing page (scroll down for feature matrix)