

Code Syntax

terraform fmt	Formats Terraform configuration files to a canonical format.
terraform validate	Validates the syntax and configuration of Terraform files.

Initializing Terraform

terraform init	Initializes the working directory containing Terraform configuration files.
terraform init -upgrade	Updates provider plugins to the latest versions.

Provisioning Infrastructure

terraform plan	Generates an execution plan showing what Terraform will do.
terraform apply	Applies the changes required to reach the desired state.
terraform destroy	Destroy all remote objects managed.

Miscellaneous

terraform version	Displays the Terraform version information.
terraform console	Opens an interactive console to evaluate Terraform expressions.
terraform graph	Generates a visual representation of Terraform resources and dependencies.

Azure Service Principal

```
provider "azurerm" {
  features {}

  subscription_id = "<subscription_id>"
  client_id       = "<application_id>"
  client_secret   = "<secret_value>"
  tenant_id      = "<tenant_id>"
}
```

Azure Backend

```
terraform {
  backend "azurerm" {
    resource_group_name = "<resource_group_name>"
    storage_account_name = "<storage_account_name>"
    container_name = "<container_name>"

    key = "<state_file_key>"
    access_key = "<storage_account_access_key>"
  }
}
```

Azure Required Provider

```
terraform {
  required_providers {
    azurerm = {
      source = "hashicorp/azurerm"
      version = "~> x.xx.x"
    }
  }
}
```

Valid Operators:

= : Allows only one exact version number.
 != : Excludes an exact version number.
 >, >=, <, <=: Comparisons against a specified version.
 ~>: Allows only the rightmost version component to increment.

Terraform State

terraform state list	Lists all resources in the Terraform state.
terraform state show <resource>	Shows the attributes of a specific resource in the Terraform state.
terraform state rm <resource>	Removes a resource from the Terraform state.
terraform state mv <old-resource> <new-resource>	Moves a resource within the Terraform state.



By [williamarobinson](#)

Published 10th April, 2024.

Last updated 10th April, 2024.

Page 1 of 2.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

Terraform State Backend

<code>terraform init -backend--config=<config></code>	Initializes Terraform with a specific backend configuration.
<code>terraform remote config</code>	Configures remote state storage.
<code>terraform remote pull</code>	Retrieves the current state from the remote backend.
<code>terraform remote push</code>	Uploads the local state to the remote backend.

Resource Manipulation (cont)

```
> destroy = false
}
}
```

Command-Line Flags

<code>-upgrade</code>	Opt to upgrade modules and plugins
<code>-reconfigure</code>	Disregards any existing configuration and state.
<code>-input=true -false</code>	Ask for input if necessary. If false, will error if input was required.
<code>-no-color</code>	Disable text coloring & formatting in the output.
<code>-var 'NAME=VALUE'</code>	Sets a value for a single input variable.
<code>-var-file=FILENAME</code>	Sets values for variables as TFVARS.
<code>-target=ADDRESS</code>	Focus its efforts only on specific resource instance.

Resource Manipulation

```
moved {
  from = azurer_m_resource_group_name -
ple.rg01
  to = azurer_m_resource_group_name ple.rg02
}
import {
  to = azurer_m_resource_group_name ple.rg03
  id = /subscription/ <subscription_id> / resource_group / <resource_group_name>
}
removed {
  from = azurer_m_resource_group_name -
ple.rg02
  lifecycle {
```



By [williamarobinson](#)

Published 10th April, 2024.

Last updated 10th April, 2024.

Page 2 of 2.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>