# TFLint Terraform linter

**START HERE**

**`tflint --init`**
Install plugins

**`tflint --enable-rule=`***`<rule>`*
Turn a rule on

**`tflint`**
Lint this directory

**`tflint --fix`**
Auto-fix issues

**`tflint --recursive`**
Lint every module dir

**`tflint --chdir=`***`<dir>`*
Lint another dir

**`tflint --format=compact`**
One line per issue

**`tflint --config=`***`<file>`*
Use a config file

**.tflint.hcl** plugins and rules → **`tflint --init`** → **Plugins** rulesets on disk → **`tflint`** → **Rules** run on your .tf files → **`exit code`** → **Issues** error / warning / notice

## CONCEPTS

TFLint checks Terraform code for mistakes terraform validate misses: bad instance types, deprecated syntax, unused declarations.

The terraform ruleset is bundled and on by default; cloud rulesets (aws, azurerm, google) are plugins you must declare and install.

Run tflint from the module directory, or pass --chdir: it reads .tf files and config from there, not from parent dirs.

## SETUP

**`brew install terraform-linters/tap/tflint`**
Install with Homebrew

**`tflint --version`**
Show version and loaded plugins

**`tflint --init`**
Install plugins from the config

**`tflint --recursive --init`**
Install plugins for all directories

**`docker run --rm -v $(pwd):/data -t ghcr.io/terraform-linters/tflint`**
Run the container image

## SETUP (cont.)

**`export GITHUB_TOKEN=`***`<token>`*
Avoid the --init rate limit
*Unauthenticated --init is limited to 60 GitHub requests per hour per IP.*

## RUNNING

**`tflint`**
Lint the current directory

**`tflint --chdir=`***`<dir>`*
Lint a different directory
*Config is then read from <dir>/.tflint.hcl; path.cwd stays the original directory.*

**`tflint --recursive`**
Lint each directory below here

**`tflint --filter=`***`<file>`*
Only report issues in this file or glob

**`tflint --fix`**
Apply automatic fixes

**`tflint --call-module-type=`***`<all|local|none>`*
Choose which module calls to inspect
*Default is local; all also downloads remote modules.*

**`tflint --help`**
List all options

## OUTPUT AND EXIT CODES

**`tflint --format=compact`**
One line per issue

## OUTPUT AND EXIT CODES (cont.)

**`tflint --format=json`**
Machine-readable output

**`tflint --format=sarif`**
SARIF for code scanning

**`tflint --no-color`**
Plain output

**`tflint --force`**
Exit 0 even if issues are found
*Default exit codes: 0 clean, 1 error, 2 issues found.*

**`tflint --minimum-failure-severity=`***`<error|warning|notice>`*
Fail only at this severity or worse

**Exit codes**

| Code | Meaning |
|---|---|
| 0 | No issues |
| 1 | Error while running |
| 2 | Issues found |

## RULES

**`tflint --enable-rule=`***`<rule>`*
Enable a rule for this run

**`tflint --disable-rule=`***`<rule>`*
Disable a rule for this run

**`tflint --only=`***`<rule>`*
Run just this rule

## RULES (cont.)

**Rule priority, highest first**

| Source | Where |
|---|---|
| 1. --only | CLI |
| 2. --enable-rule / --disable-rule | CLI |
| 3. rule blocks | Config file |
| 4. preset | terraform plugin config |
| 5. disabled_by_default | Config file |

## CONFIG FILE

**`tflint --config=`***`<file>`*
Use this config file

**`tflint --config=$(realpath .tflint.hcl) --recursive`**
Share one config across directories
*With --recursive, relative config paths resolve inside each directory.*

**Config lookup order**

| Order | File |
|---|---|
| 1 | `--config` |
| 2 | `$TFLINT_CONFIG_FILE` |
| 3 | `./.tflint.hcl` |
| 4 | `./.tflint.json` |
| 5 | `~/.tflint.hcl` |
| 6 | `~/.tflint.json` |

## CONFIG FILE (cont.)

**plugin and rule blocks**

```
plugin "aws" {
  enabled = true
  version = "0.4.0"
  source  = "github.com/terraform-linters/tflint-ruleset-aws"
}

rule "terraform_typed_variables" {
  enabled = true
}
```

**config block**

```
config {
  format              = "compact"
  call_module_type    = "local"
  disabled_by_default = false
  varfile             = ["dev.tfvars"]
  variables           = ["env=dev"]
}
```

Plugin version must be exact and without a leading v; constraints like >= 0.3 are not supported.

## IGNORING ISSUES

**Ignore one line**

```
# tflint-ignore: <rule>
instance_type = "t1.2xlarge"
```

**Ignore with a reason**

```
# tflint-ignore: <rule> # too new for TFLint
instance_type = "t1.2xlarge"
```

**Ignore a whole file**

```
# tflint-ignore-file: <rule>
```

## IGNORING ISSUES (cont.)

tflint-ignore-file only works at the top of a file; inside a block it is an error. In .tf.json files use a "//" property.

## MODULES AND VARIABLES

**`tflint --var-file=`***`<file>`*
Load a tfvars file
*terraform.tfvars and *.auto.tfvars load automatically.*

**`tflint --var='`***`<name>`*`=`*`<value>`*`'`**
Set a variable

**`tflint --ignore-module=`***`<source>`*
Skip a module by source
*Use the module source, not its name in your code.*

## RECIPES

**First run in a repo**

```
cat > .tflint.hcl <<'EOF'
plugin "terraform" {
  enabled = true
  preset  = "recommended"
}
EOF
tflint --init
tflint --recursive
```

**CI check with a report**

```
tflint --init
tflint --recursive --format=sarif > tflint.sarif
# exit code 2 fails the job
```