

Terragrunt Configuration file



Terragrunt Configuration file

- Terragrunt configuration is defined in a file “terragrunt.hcl”
- The data written inside this file “terragrunt.hcl” uses the same HCL syntax as Terraform
- Terragrunt supports JSON-serialized HCL defined in “terragrunt.hcl.json”
- Terragrunt identifies “<filename>.hcl” files from the path specified in order:
 - If CLI option defined, --terragrunt-config
 - Environment variable, TERRAGRUNT_CONFIG
 - Current working directory, if terragrunt.hcl file exists
 - Current working directory, if terraform.hcl.json file exists
 - Throws error, if not found any one of the above specified approach
- We can rewrite the hcl configuration files to a canonical format of terragrunt using command `#terragrunt hclfmt`
- Above command will check for hcl files in recursive order as well
- In order to preview the format of the hcl files, instead of overwriting them. Use command `#terragrunt hclfmt --terragrunt-check`
- For more info [Doc](#)

Sample terragrunt configuration file "terragrunt.hcl"

```
terraform {  
  # Deploy version @v0.1.0 on target provider  
  source = "git::git@github.com:sudheerdemo/modules.git//app?ref=v0.1.0"  
}  
  
generate "backend" {  
  path = "aws_s3_backend.tf"  
  if_exists = "overwrite_terragrunt"  
  contents = <<EOF  
terraform {  
  backend "s3" {  
    bucket      = "dev-env-statefiles"  
    key         = "${path_relative_to_include()}/terraform.tfstate"  
    region      = "us-east-1"  
    encrypt     = true  
    dynamodb_table = "dev-env-statefilelock"  
  }  
}  
EOF  
}  
  
inputs = {  
  inst_count = 3  
  inst_type  = "t2.large"  
  az_name    = "us-east-1a"  
  vpc_id     = dependency.vpc.outputs.vpc_id  
}
```

Terraform block

Generate block

Inputs attribute

```
environments
├── dev
│   └── terragrunt.hcl
├── prod
│   └── terragrunt.hcl
├── staging
│   ├── eks-v01
│   │   ├── eks-v01-rc1
│   │   │   └── terragrunt.hcl
│   │   ├── eks-v01-rc2
│   │   │   └── terragrunt.hcl
│   │   └── terragrunt.hcl
│   └── terragrunt.hcl
└── terragrunt.hcl
```

#terraform hclfmt

Terragrunt will pick configuration files from working directory and sub-folders as well in recursive order

environments/terragrunt.hcl

environments/dev/terragrunt.hcl

environments/prod/terragrunt.hcl

environments/staging/terragrunt.hcl

environments/staging/eks-v01/terragrunt.hcl

environments/staging/eks-v01/eks-v01-rc1/terragrunt.hcl

environments/staging/eks-v01/eks-v01-rc2/terragrunt.hcl