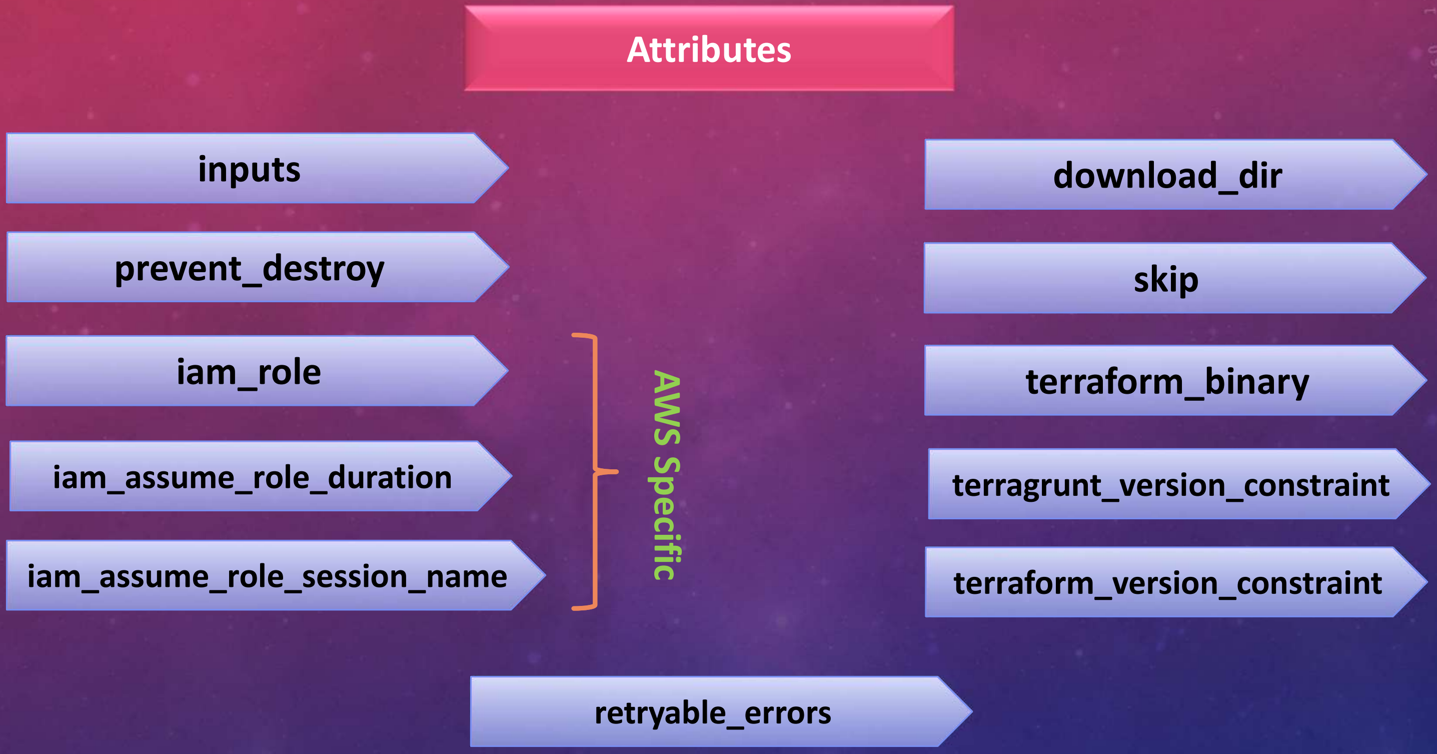


Attributes

- inputs
- prevent_destroy
- iam_role
- iam_assume_role_duration
- iam_assume_role_session_name

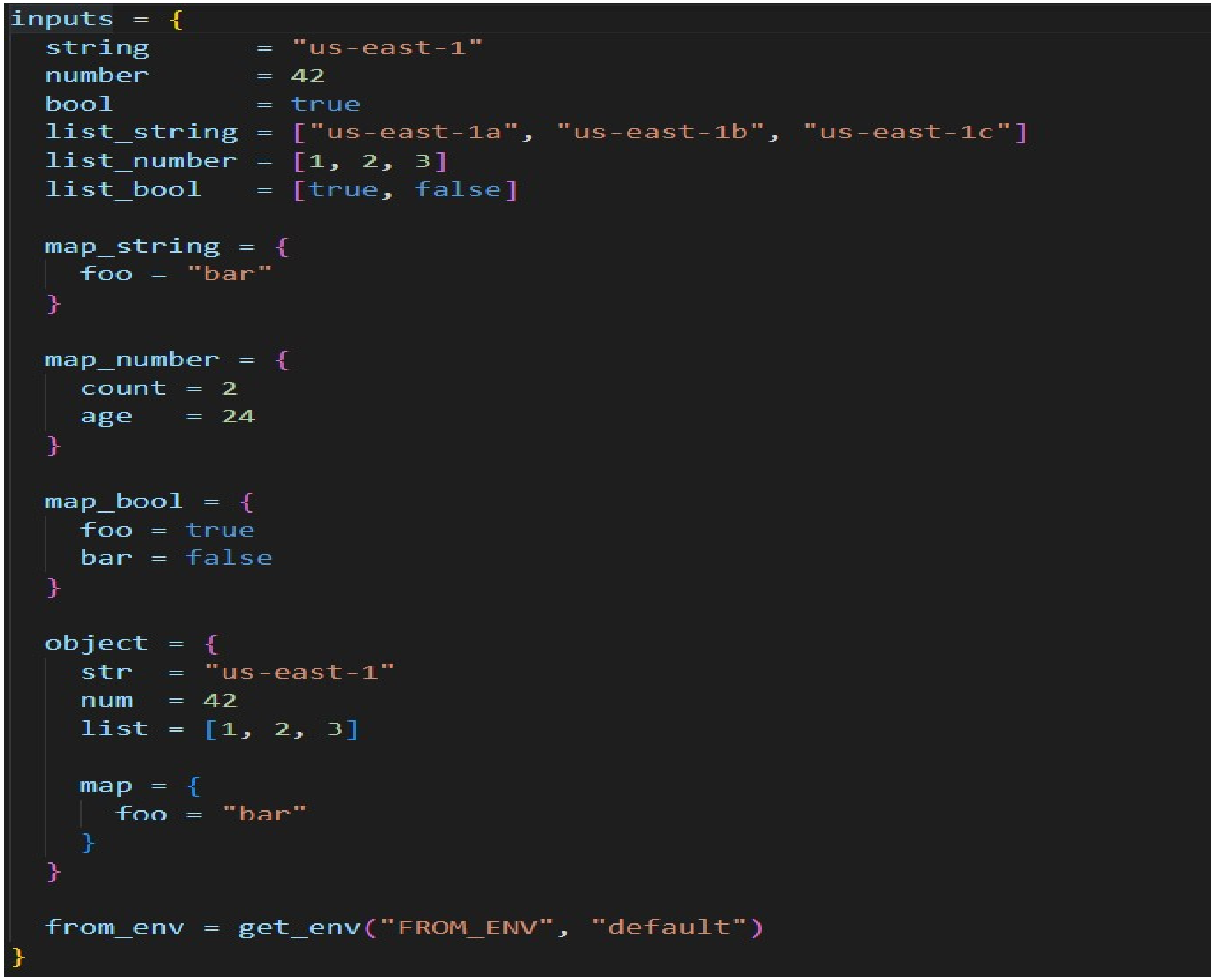
AWS Specific

- download_dir
- skip
- terraform_binary
- terragrunt_version_constraint
- terraform_version_constraint

- retryable_errors

Inputs attribute

- To pass inputs variables and values to Terraform resources
- We define inputs in map format
- Map details of Inputs attributes are passed as Environment variables to Terraform
- For example: TF_VAR_instance_type with value in JSON encoded format, similar to Terraform environment variables
- Limitation: type information is lost when variables are passed from Terragrunt to Terraform variables
- For which we have to define variables block with proper type attribute in Terraform variables file
- Inputs attribute accepts String, Integer, Boolean, list of strings, Integers, Boolean values, map of string, number, Boolean, object and from_env



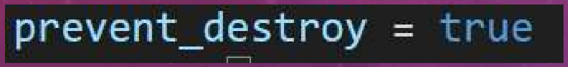
```
inputs = {
  string      = "us-east-1"
  number      = 42
  bool        = true
  list_string = ["us-east-1a", "us-east-1b", "us-east-1c"]
  list_number = [1, 2, 3]
  list_bool   = [true, false]

  map_string = {
    foo = "bar"
  }

  map_number = {
    count = 2
    age   = 24
  }

  map_bool = {
    foo = true
    bar = false
  }

  object = {
    str  = "us-east-1"
    num  = 42
    list = [1, 2, 3]

    map = {
      foo = "bar"
    }
  }

  from_env = get_env("FROM_ENV", "default")
}
```

Download_dir attribute

- To specify the download path for terragrunt, instead of default location/path
- Default path Terragrunt save terraform configuration files in working directory under: “.terragrunt-cache”
- Order that terragrunt follows:
  - --terragrunt-download-dir passed in terragrunt CLI
  - TERRAGRUNT_DOWNLOAD defined through Environment variable
  - Terragrunt.hcl defined with download_dir attribute
  - Module directory
  - Include block in terragrunt.hcl

Prevent_destroy attribute

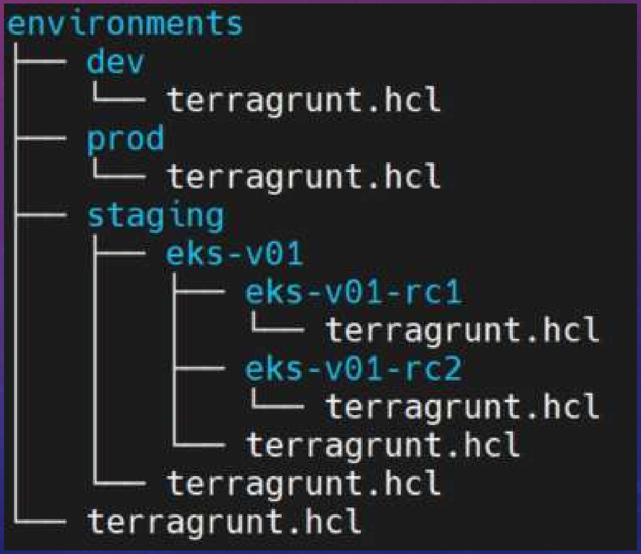
- It protects selected modules from accidental deletion
- While executing commands like #terragrunt destroy or terragrunt destroy-all
- This option will helps to carefully protect for critical workloads like VPC, RDS



```
prevent_destroy = true
```

Skip attribute

- In order to ignore the changes done in the specific module, we use skip attribute
- Terragrunt commands will skip that specific module where we have defined skip attribute with true value
- Skip attribute will be defined in terragrunt.hcl configuration files (skip = true)



```
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_binary

- We have to use terraform_binary attribute, when we have placed terraform binary/package in custom path
- In some cases, we will not store binaries in default $PATH location.
- Order to define this attribute:
  - --terragrunt-tfpath from CLI
  - TERRAGRUNT_TFPATH
  - Terragrunt.hcl in module directory

Terraform_version_constraint

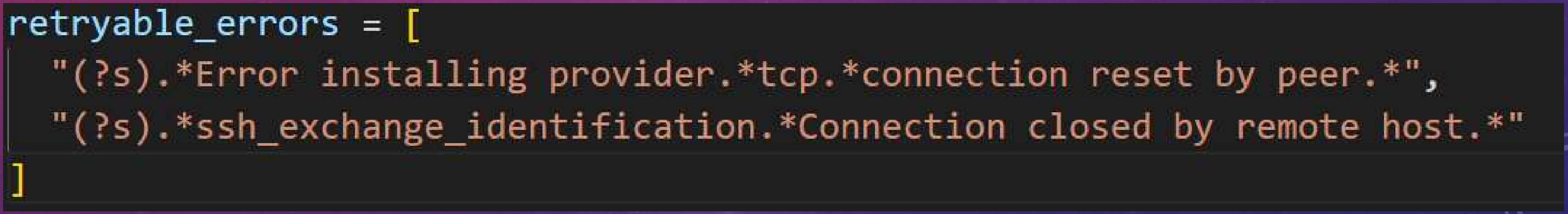
- If we want to use a specific terraform version
- Some cases, we will develop Terraform configuration files on specific version of Terraform which may be not compatible with older versions or any
- Entry in terragrunt configuration file #terraform_version_constraint = “>= 1.2.5”

Terragrunt_version_constraint

- If we want to use a specific terragrunt version
- In some cases, the attributes and blocks etc., we use may be deprecated or removed which will cause errors
- Entry in terragrunt configuration file #terragrunt_version_constraint = “>= 0.38.0”

Retryable_errors

- Terragrunt has a small list of default known errors built-in
- To define custom error messages to the default retryable error to make it more readable
- By default, auto-retry tries a maximum of three times to re-run the command, which pauses for 5 seconds delay in retry
- Parameters to control auto-retry on failure #retry_max_attempts = 5 and #retry_sleep_interval_sec = 60



```
retryable_errors = [
  "(?s).*Error installing provider.*tcp.*connection reset by peer.*",
  "(?s).*ssh_exchange_identification.*Connection closed by remote host.*"
]
```

iam_role

- AWS IAM service provides Role for authentication
- Roles can be assumed by Users (IAM Users and Federated users) and AWS Service to communicate to AWS Services
- Terragrunt/Terraform will use that Role as specified to authenticate to AWS Account
- Argument to define iam_role in terragrunt configuration file
  - iam_role = “arn:aws:iam::ACCOUNT_ID:role/ROLE_NAME”
  - Note: we can pass IAM role in CLI, ENV, CONFIG

iam_assume_role_duration

- AWS IAM assume role duration will provide STS session duration in seconds
- This IAM role will be assumed by Terragrunt to contact AWS Services and perform the task within the duration specified (This value has to be less or equal to the value defined by the AWS Account administrator)
- AWS STS Session default value 1 hour (3600 seconds)
- Argument to define in configuration file
  - iam_assume_role_duration = 14400

iam_assume_role_session_name

- To specify name of the STS session to be assumed by Terragrunt prior to the connection
- Argument to define in terragrunt configuration file
  - iam_assume_role_session_name = terragrunt