Automate IaC documentations with terraform-docs

Khosrow Moossavi (@khos2ow)
What is terraform-docs
and why it’s needed!
where to get terraform-docs

- GitHub Releases (github.com/terraform-docs/terraform-docs/releases)
- Official Docker image (quay.io/terraform-docs/terraform-docs)
- Homebrew (brew install terraform-docs)
- Chocolatey (choco install terraform-docs)
- Scoop (scoop install terraform-docs)
$ terraform-docs markdown ./my-terraform-module
markdown table

➔ most popular format
➔ perfect for README

Example:

```
$ terraform-docs markdown table .
```

https://github.com/terraform-docs/terraform-docs/blob/master/docs/formats/markdown-table.md
tfvars hcl

→ prepopulate terraform.tfvars

Example:

```
$ terraform-docs tfvars hcl . > terraform.tfvars
```

```
subnet_ids = ""
vpc_id = ""
extra_environment = []
extra_tags = {}
instance_count = 1
instance_name = "test-
```
.terraform-docs.yml

- since v0.10.0
- check-in config with the code
- unified user experience

Example:

```
$ terraform-docs -c .terraform-docs.yml .
```

https://github.com/terraform-docs/terraform-docs/blob/master/docs/CONFIG_FILE.md
New Features!
GitHub Action

- commit changes back to PR
- fail CI if changes detected

```
name: Build
on:
push:
  branches-ignore:
    - main
jobs:
  build:
    name: Documentations
    runs-on: ubuntu-latest
    steps:
    - name: Checkout
      uses: actions/checkout@v2
      with:
        ref: ${{ github.event.pull_request.head.ref }}
    - name: Regenerate README.md
      uses: terraform-docs/gh-actions@v0.4.0
      with:
        working-dir: .
        output-file: README.md
        config-file: .terraform-docs.yml
        git-push: true
        git-push-sign-off: true
```

https://github.com/terraform-docs/gh-actions/
https://github.com/khos2ow/fosdem-tfdocs-demo/
Plugin SDK

➔ write your own formatter!
➔ needs .terraform-docs.yml

```go
import (  
  "github.com/terraform-docs/plugin-sdk/plugin"
  "github.com/terraform-docs/terraform-docs/pkg/print"
  "github.com/terraform-docs/terraform-docs/pkg/tfconf"
  "github.com/terraform-docs/terraform-docs/pkg/tmpl"
)

func main() {  
  plugin.Serve(&plugin.ServeOpts{    
    Name:  "foo",
    Version:  "0.1.0",
    Engine:  &engine{},
  })
}

type engine struct {}

func (e *engine) Print(m *tfconf.Module, s *print.Settings) (string, error) {  
  // define sections' template and order as you wish!
  tt := tmpl.NewTemplate( ... )

  rendered, err := e.template.Render(m)
  if err != nil {    
    return "", err
  }
  return rendered, nil
}
```

https://github.com/terraform-docs/plugin-sdk/
https://github.com/terraform-docs/tfdocs-format-template/
Thanks!

Any questions?

https://github.com/terraform-docs/
@khos2ow