Automate IaC documentations with terraform-docs

Khosrow Moossavi (@khos2ow)

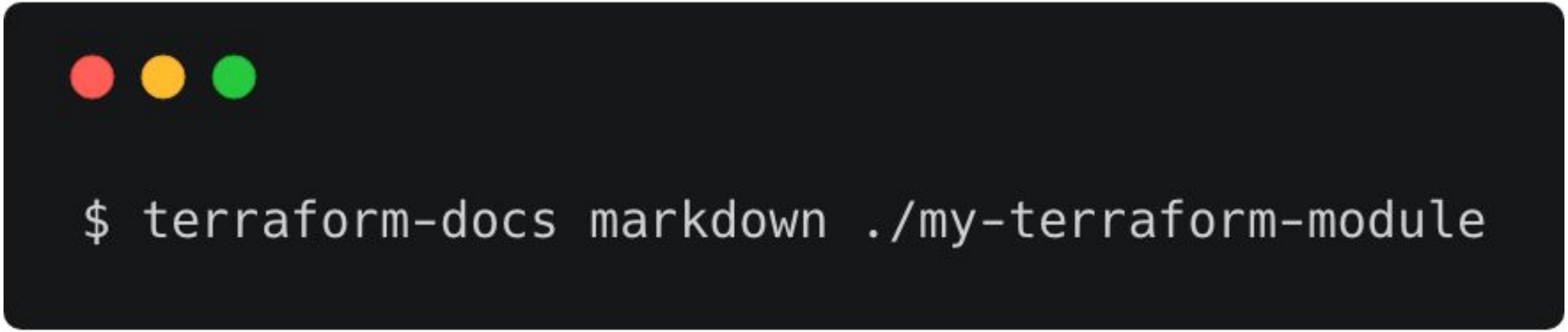
FOSDEM'21

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
```

- asciidoc (document and table)
- json
- markdown (document and table)
- pretty
- tfvars (hcl and json)
- toml
- xml
- yaml

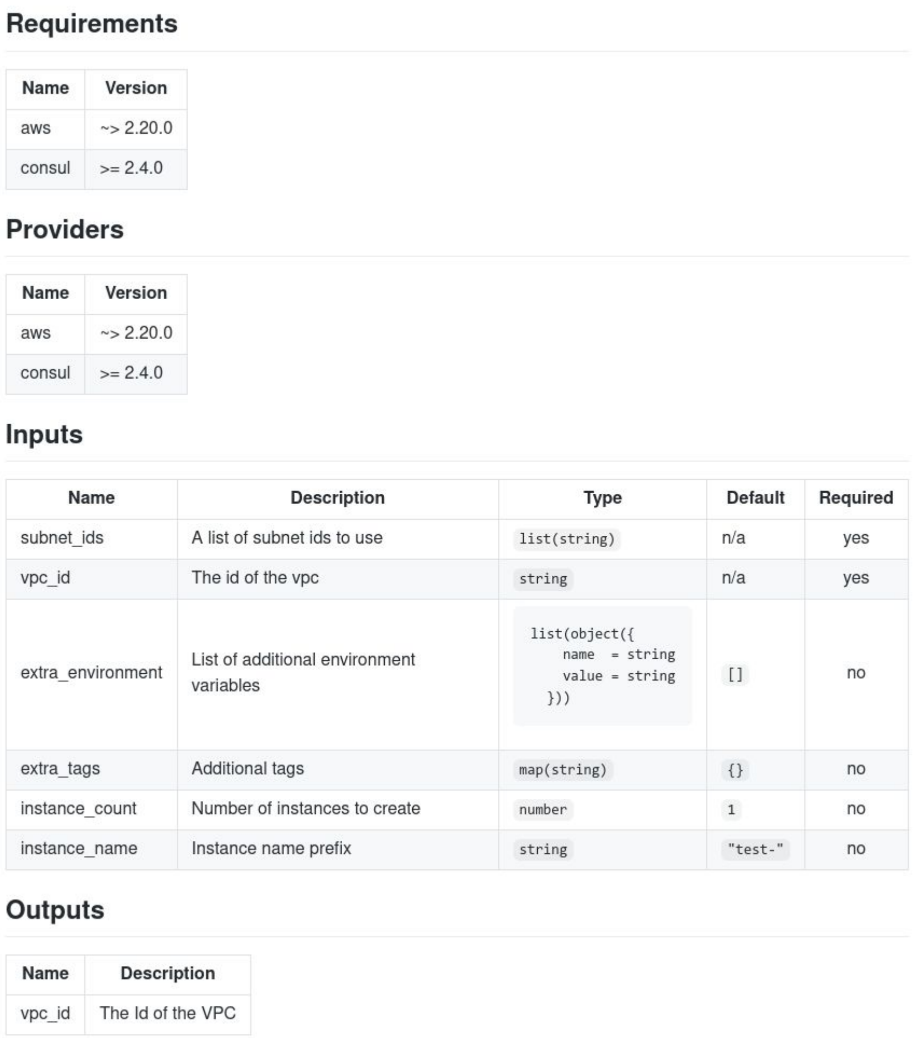
https://github.com/terraform-docs/terraform-docs/blob/master/docs/FORMATS_GUIDE.md

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



Requirements

Name	Version
aws	~> 2.20.0
consul	>= 2.4.0

Providers

Name	Version
aws	~> 2.20.0
consul	>= 2.4.0

Inputs

Name	Description	Type	Default	Required
subnet_ids	A list of subnet ids to use	`list(string)`	n/a	yes
vpc_id	The id of the vpc	`string`	n/a	yes

extra_environment	List of additional environment variables	`list(object({`
		`    name  = string`
		`    value = string`
		`  }))`
			`[]`	no

extra_tags	Additional tags	`map(string)`	`{}`	no
instance_count	Number of instances to create	`number`	`1`	no
instance_name	Instance name prefix	`string`	`"test-"`	no

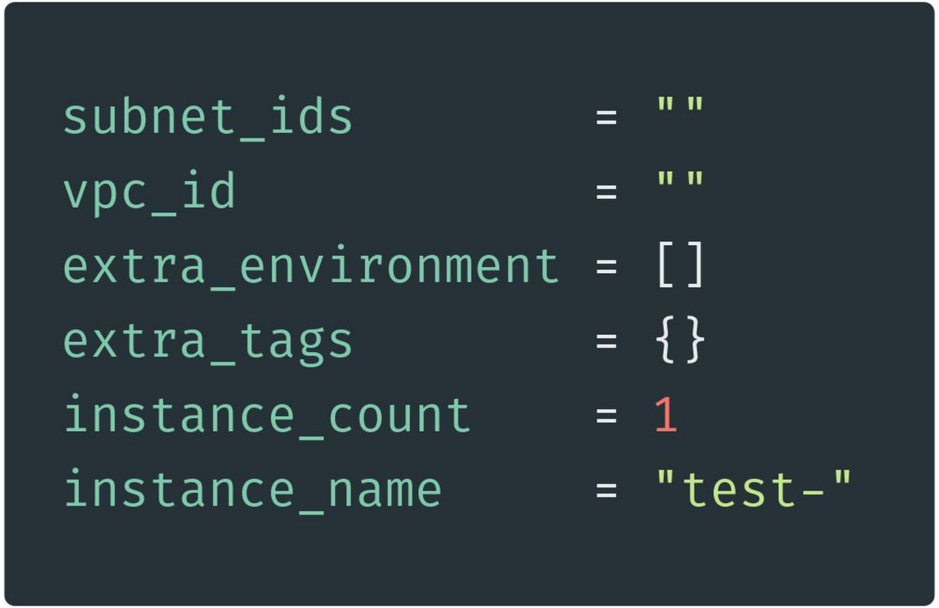
Outputs

Name	Description
vpc_id	The Id of the VPC

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-"
```

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

.terraform-docs.yml

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

```
Example:

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

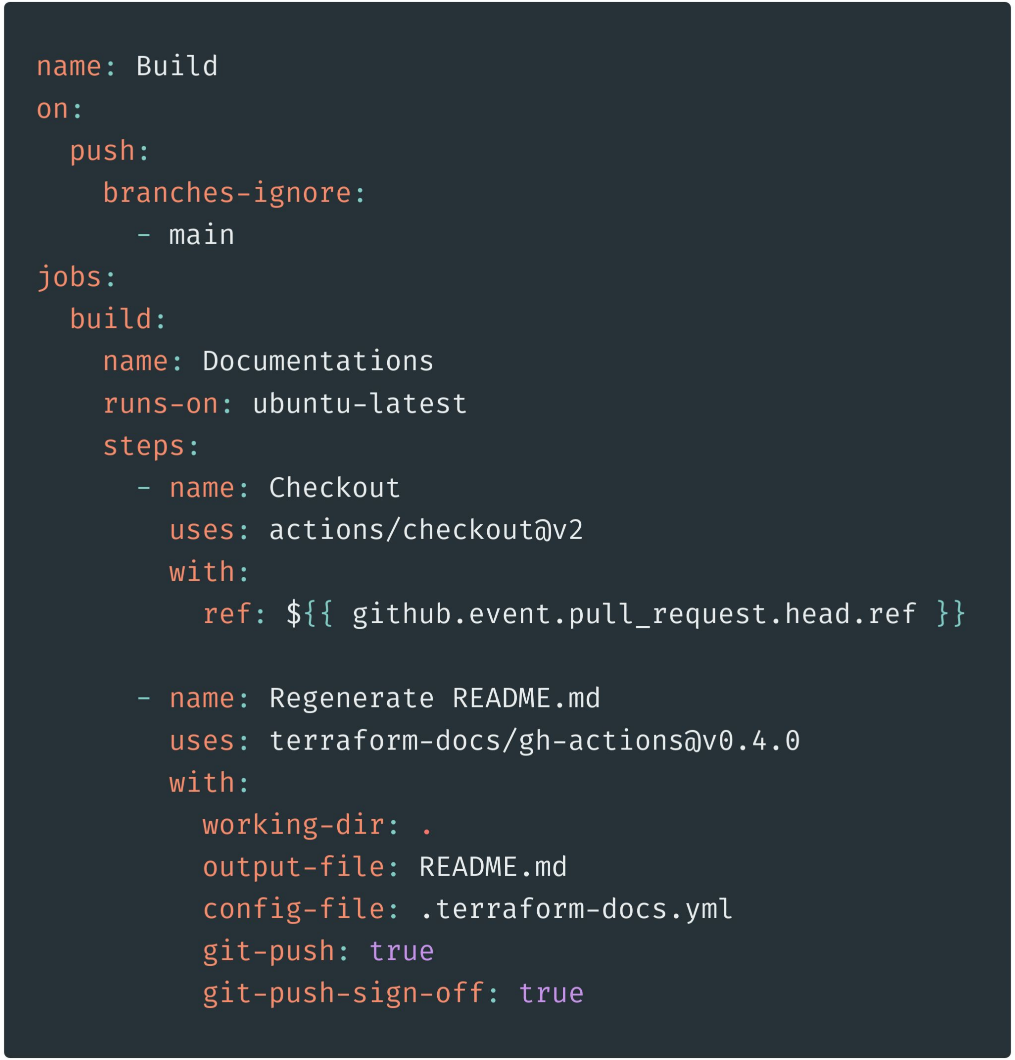
https://github.com/terraform-docs/terraform-docs/blob/master/docs/CONFIG_FILE.md

```yaml
formatter: <FORMATTER_NAME>
header-from: main.tf

sections:
  hide-all: false
  hide:
    - header
    - inputs
    - outputs
    - providers
    - requirements
  show-all: true
  show:
    - header
    - inputs
    - outputs
    - providers
    - requirements

output-values:
  enabled: false
  from: ""

sort:
  enabled: true
  by:
    - required
    - type

settings:
  color: true
  escape: true
  indent: 2
  required: true
  sensitive: true
```

New Features!

GitHub Action

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



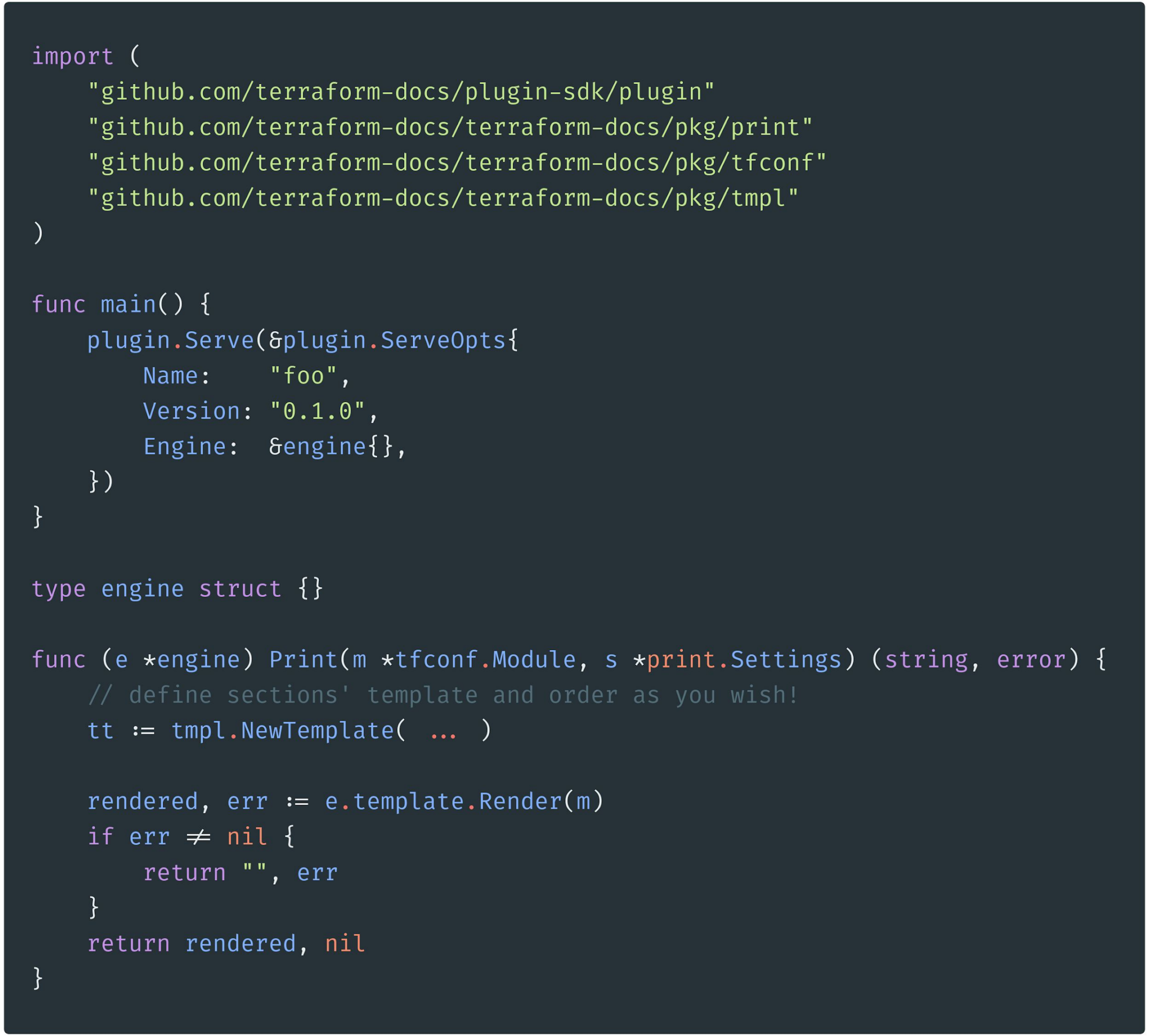
```yaml
name: Build
on:
  push:
    branches-ignore:
      - main
jobs:
  build:
    name: Documentations
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v2
        with:
          ref: ${{ github.event.pull_request.head.ref }}

      - 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

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



```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
}
```

Demo

Thanks!

Any questions?

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

@khos2ow