Terraform

template

-IAAS

-Automation tool

hashcorp-maintaining the terraform

TERRAFORM WORKFLOW

-fundamental bloc

-variable block

-online/calling block.

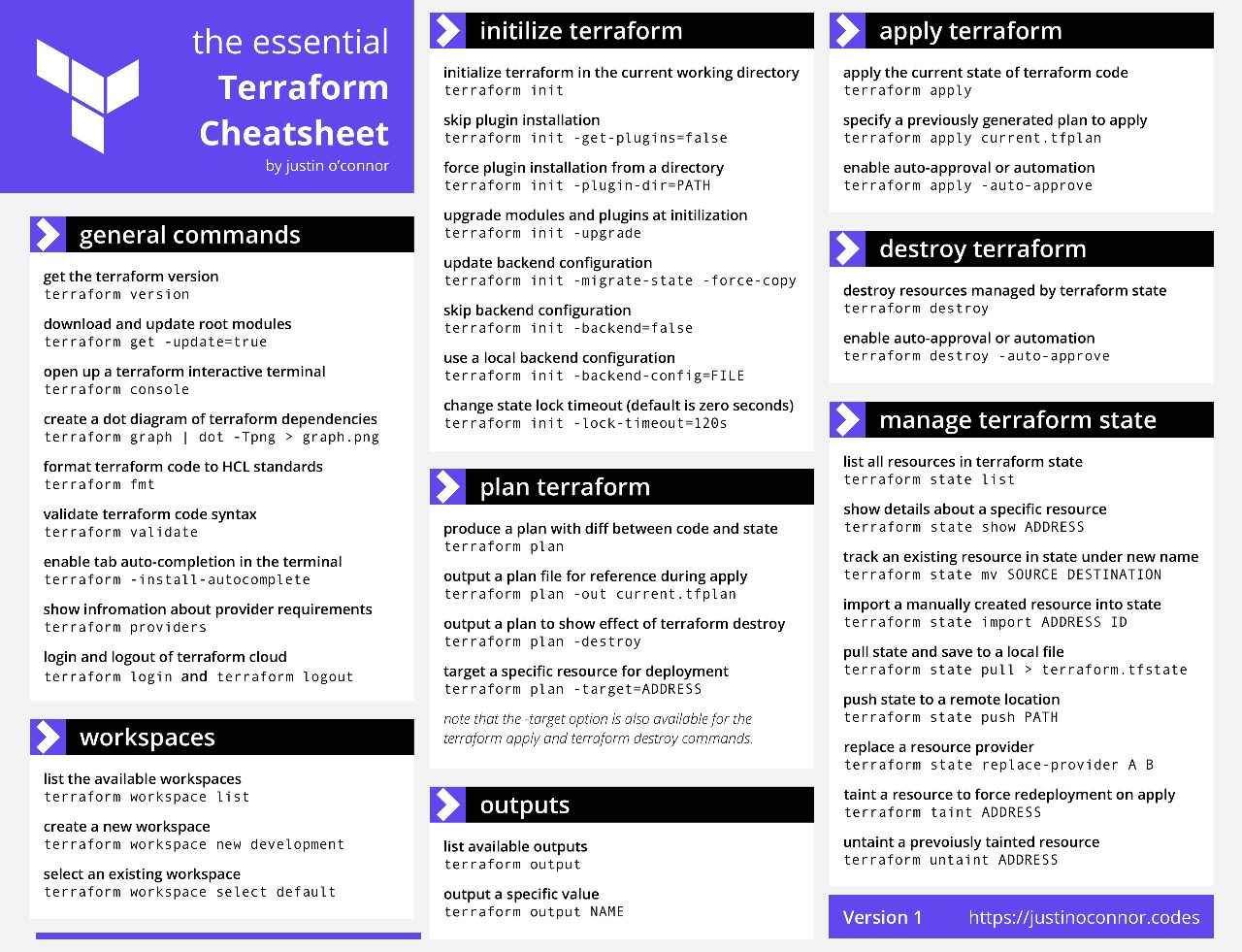
#terraform init

#terraform validate

#terraform plan

#terraform apply

#terraform destroy



CODE:

region = "us-east-1"

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "~> 5.0"

}

}

}

# Configure the AWS Provider

provider "aws" {

region = "us-east-1a"

resource "aws\_vpc" "myvpc" {

cidr\_block = "10.0.0.0/16"

tags = {

Name = "demovpc"

  }

}

# Create a VPC

resource "aws\_vpc" "example" {

cidr\_block = "10.0.0.0/21"

}

# Configure the AWS Provider

provider "aws" {

version = "~> 5.0"

region = "us-east-1b"

resource "aws\_subnet" "pubsub" {

vpc\_id = aws\_vpc.myvpc.id

cidr\_block = "10.0.1.0/24"

availability\_zone = "us-east-1a"

tags = {

Name = "sn1"

  }

}

}

# Create a VPC

resource "aws\_vpc" "example" {

cidr\_block = "10.0.0.0/22"

resource "aws\_internet\_gateway" "tfigw" {

vpc\_id = aws\_vpc.myvpc.id

tags = {

Name = "tfigw"esource "aws\_route\_table" "tfprirt" {

vpc\_id = aws\_vpc.myvpc.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_nat\_gateway.tfnat.id

}

tags = {

Name = "tfprivateroute"

  }

}

  }

}

resource "aws\_route\_table" "tfpubrt" {

vpc\_id = aws\_vpc.myvpc.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.tfigw.id

}

tags = {

Name = "tfpublicroute"

  }

}

resource "aws\_route\_table\_association" "pubsn1" {

subnet\_id = aws\_subnet.pubsub.id

route\_table\_id = aws\_route\_table.tfpubrt.id

}

resource "aws\_route\_table\_association" "pubsn2" {

subnet\_id = aws\_subnet.pub\_sub.id

route\_table\_id = aws\_route\_table.tfpubrt.id

}

resource "aws\_eip" "tfeip" {

domain = "vpc"

}

resource "aws\_nat\_gateway" "tfnat" {

allocation\_id = aws\_eip.tfeip.id

subnet\_id = aws\_subnet.pub\_sub.id

tags = {

Name = "gw NAT"

  }

}

}

resource "aws\_route\_table\_association" "prisn3" {

subnet\_id = aws\_subnet.prisub.id

route\_table\_id = aws\_route\_table.tfprirt.id

}

resource "aws\_route\_table\_association" "prisn4" {

subnet\_id = aws\_subnet.pri\_sub.id

route\_table\_id = aws\_route\_table.tfprirt.id

}

resource "aws\_security\_group" "allow\_tfsg" {

name = "allow\_tfsg"

description = "Allow TLS inbound traffic"

vpc\_id = aws\_vpc.myvpc.id

ingress {

description = "HTTPS "

from\_port = 443

to\_port = 443

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

description = "HTTP "

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

description = "SSH"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

tags = {

Name = "TfsecurityGroup"

  }

}

resource "aws\_instance" "pub\_ins" {

ami = "ami-0fc5d935ebf8bc3bc"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.pub\_sub.id

vpc\_security\_group\_ids = [aws\_security\_group.allow\_tfsg.id]

key\_name = "David"

associate\_public\_ip\_address   =  "true"

}

resource "aws\_instance" "pri\_ins" {

ami = "ami-0fc5d935ebf8bc3bc"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.prisub.id

vpc\_security\_group\_ids = [aws\_security\_group.allow\_tfsg.id]

key\_name      = "David"

}