CSDO1010 Assignment 8

Paul Krznaric

# Repo:

<https://github.com/PaulKrznaric/08.Lab8-Prometheus-and-Grafana.git>

# Proof:

## Terraform Apply

> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

# module.compute.aws\_instance.k8s-master will be created

+ resource "aws\_instance" "k8s-master" {

+ ami = (sensitive)

+ arn = (known after apply)

+ associate\_public\_ip\_address = true

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_initiated\_shutdown\_behavior = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.medium"

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = "k8s"

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ primary\_network\_interface\_id = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

+ public\_ip = (known after apply)

+ secondary\_private\_ips = (known after apply)

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ tags = {

+ "Name" = "k8s\_master\_tf"

}

+ tags\_all = {

+ "Name" = "k8s\_master\_tf"

}

+ tenancy = (known after apply)

+ vpc\_security\_group\_ids = (known after apply)

+ capacity\_reservation\_specification {

+ capacity\_reservation\_preference = (known after apply)

+ capacity\_reservation\_target {

+ capacity\_reservation\_id = (known after apply)

}

}

+ ebs\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ snapshot\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

+ enclave\_options {

+ enabled = (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

+ metadata\_options {

+ http\_endpoint = (known after apply)

+ http\_put\_response\_hop\_limit = (known after apply)

+ http\_tokens = (known after apply)

}

+ network\_interface {

+ delete\_on\_termination = (known after apply)

+ device\_index = (known after apply)

+ network\_interface\_id = (known after apply)

}

+ root\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

}

# module.compute.aws\_instance.k8s-node will be created

+ resource "aws\_instance" "k8s-node" {

+ ami = (sensitive)

+ arn = (known after apply)

+ associate\_public\_ip\_address = true

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_initiated\_shutdown\_behavior = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.medium"

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = "k8s"

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ primary\_network\_interface\_id = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

+ public\_ip = (known after apply)

+ secondary\_private\_ips = (known after apply)

+ security\_groups = (known after apply)

+ source\_dest\_check = true

+ subnet\_id = (known after apply)

+ tags = {

+ "Name" = "k8s\_node\_tf"

}

+ tags\_all = {

+ "Name" = "k8s\_node\_tf"

}

+ tenancy = (known after apply)

+ vpc\_security\_group\_ids = (known after apply)

+ capacity\_reservation\_specification {

+ capacity\_reservation\_preference = (known after apply)

+ capacity\_reservation\_target {

+ capacity\_reservation\_id = (known after apply)

}

}

+ ebs\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ snapshot\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

+ enclave\_options {

+ enabled = (known after apply)

}

+ ephemeral\_block\_device {

+ device\_name = (known after apply)

+ no\_device = (known after apply)

+ virtual\_name = (known after apply)

}

+ metadata\_options {

+ http\_endpoint = (known after apply)

+ http\_put\_response\_hop\_limit = (known after apply)

+ http\_tokens = (known after apply)

}

+ network\_interface {

+ delete\_on\_termination = (known after apply)

+ device\_index = (known after apply)

+ network\_interface\_id = (known after apply)

}

+ root\_block\_device {

+ delete\_on\_termination = (known after apply)

+ device\_name = (known after apply)

+ encrypted = (known after apply)

+ iops = (known after apply)

+ kms\_key\_id = (known after apply)

+ tags = (known after apply)

+ throughput = (known after apply)

+ volume\_id = (known after apply)

+ volume\_size = (known after apply)

+ volume\_type = (known after apply)

}

}

# module.compute.aws\_key\_pair.aws-key will be created

+ resource "aws\_key\_pair" "aws-key" {

+ arn = (known after apply)

+ fingerprint = (known after apply)

+ id = (known after apply)

+ key\_name = "k8s"

+ key\_pair\_id = (known after apply)

+ public\_key = "ssh-rsa  paulkrznaric@Pauls-MBP.local"

+ tags\_all = (known after apply)

}

# module.vpc.aws\_internet\_gateway.tf\_igw will be created

+ resource "aws\_internet\_gateway" "tf\_igw" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "Terraform-Gateway"

}

+ tags\_all = {

+ "Name" = "Terraform-Gateway"

}

+ vpc\_id = (known after apply)

}

# module.vpc.aws\_route\_table.tf\_public\_route will be created

+ resource "aws\_route\_table" "tf\_public\_route" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ propagating\_vgws = (known after apply)

+ route = [

+ {

+ carrier\_gateway\_id = ""

+ cidr\_block = "0.0.0.0/0"

+ destination\_prefix\_list\_id = ""

+ egress\_only\_gateway\_id = ""

+ gateway\_id = (known after apply)

+ instance\_id = ""

+ ipv6\_cidr\_block = ""

+ local\_gateway\_id = ""

+ nat\_gateway\_id = ""

+ network\_interface\_id = ""

+ transit\_gateway\_id = ""

+ vpc\_endpoint\_id = ""

+ vpc\_peering\_connection\_id = ""

},

]

+ tags = {

+ "Name" = "Terraform-Public-RouteTable"

}

+ tags\_all = {

+ "Name" = "Terraform-Public-RouteTable"

}

+ vpc\_id = (known after apply)

}

# module.vpc.aws\_route\_table\_association.tf\_public1\_assoc will be created

+ resource "aws\_route\_table\_association" "tf\_public1\_assoc" {

+ id = (known after apply)

+ route\_table\_id = (known after apply)

+ subnet\_id = (known after apply)

}

# module.vpc.aws\_route\_table\_association.tf\_public2\_assoc will be created

+ resource "aws\_route\_table\_association" "tf\_public2\_assoc" {

+ id = (known after apply)

+ route\_table\_id = (known after apply)

+ subnet\_id = (known after apply)

}

# module.vpc.aws\_security\_group.tf\_public\_sg will be created

+ resource "aws\_security\_group" "tf\_public\_sg" {

+ arn = (known after apply)

+ description = "Used for access to the public instances"

+ egress = [

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = ""

+ from\_port = 0

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "-1"

+ security\_groups = []

+ self = false

+ to\_port = 0

},

]

+ id = (known after apply)

+ ingress = [

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = ""

+ from\_port = 0

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "-1"

+ security\_groups = []

+ self = false

+ to\_port = 0

},

]

+ name = "tf\_public\_sg"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "Terraform-SecurityGroup"

}

+ tags\_all = {

+ "Name" = "Terraform-SecurityGroup"

}

+ vpc\_id = (known after apply)

}

# module.vpc.aws\_subnet.tf\_public\_subnet1 will be created

+ resource "aws\_subnet" "tf\_public\_subnet1" {

+ arn = (known after apply)

+ assign\_ipv6\_address\_on\_creation = false

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

+ availability\_zone\_id = (known after apply)

+ cidr\_block = "10.0.1.0/24"

+ id = (known after apply)

+ ipv6\_cidr\_block\_association\_id = (known after apply)

+ map\_public\_ip\_on\_launch = false

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "Terraform-Subnet1"

}

+ tags\_all = {

+ "Name" = "Terraform-Subnet1"

}

+ vpc\_id = (known after apply)

}

# module.vpc.aws\_subnet.tf\_public\_subnet2 will be created

+ resource "aws\_subnet" "tf\_public\_subnet2" {

+ arn = (known after apply)

+ assign\_ipv6\_address\_on\_creation = false

+ availability\_zone = "us-east-1b"

+ availability\_zone\_id = (known after apply)

+ cidr\_block = "10.0.2.0/24"

+ id = (known after apply)

+ ipv6\_cidr\_block\_association\_id = (known after apply)

+ map\_public\_ip\_on\_launch = false

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "Terraform-Subnet2"

}

+ tags\_all = {

+ "Name" = "Terraform-Subnet2"

}

+ vpc\_id = (known after apply)

}

# module.vpc.aws\_vpc.tf\_vpc will be created

+ resource "aws\_vpc" "tf\_vpc" {

+ arn = (known after apply)

+ assign\_generated\_ipv6\_cidr\_block = false

+ cidr\_block = "10.0.0.0/16"

+ default\_network\_acl\_id = (known after apply)

+ default\_route\_table\_id = (known after apply)

+ default\_security\_group\_id = (known after apply)

+ dhcp\_options\_id = (known after apply)

+ enable\_classiclink = (known after apply)

+ enable\_classiclink\_dns\_support = (known after apply)

+ enable\_dns\_hostnames = true

+ enable\_dns\_support = true

+ id = (known after apply)

+ instance\_tenancy = "default"

+ ipv6\_association\_id = (known after apply)

+ ipv6\_cidr\_block = (known after apply)

+ main\_route\_table\_id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "Terraform-VPC"

}

+ tags\_all = {

+ "Name" = "Terraform-VPC"

}

}

Plan: 11 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ k8s-master-Public-IP = (known after apply)

+ k8s-node-Public-IP = (known after apply)

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

module.vpc.aws\_vpc.tf\_vpc: Creating...

module.compute.aws\_key\_pair.aws-key: Creating...

module.compute.aws\_key\_pair.aws-key: Creation complete after 0s [id=k8s]

module.vpc.aws\_vpc.tf\_vpc: Still creating... [10s elapsed]

module.vpc.aws\_vpc.tf\_vpc: Creation complete after 13s [id=vpc-0e2fceb6a04b0208e]

module.vpc.aws\_internet\_gateway.tf\_igw: Creating...

module.vpc.aws\_subnet.tf\_public\_subnet1: Creating...

module.vpc.aws\_subnet.tf\_public\_subnet2: Creating...

module.vpc.aws\_security\_group.tf\_public\_sg: Creating...

module.vpc.aws\_subnet.tf\_public\_subnet2: Creation complete after 1s [id=subnet-0578e0fa769f4bbdd]

module.vpc.aws\_subnet.tf\_public\_subnet1: Creation complete after 2s [id=subnet-092fd600ef6725c95]

module.vpc.aws\_internet\_gateway.tf\_igw: Creation complete after 3s [id=igw-0c644e9109b3ffbdb]

module.vpc.aws\_route\_table.tf\_public\_route: Creating...

module.vpc.aws\_route\_table.tf\_public\_route: Creation complete after 1s [id=rtb-0d260ef547f98330e]

module.vpc.aws\_route\_table\_association.tf\_public2\_assoc: Creating...

module.vpc.aws\_route\_table\_association.tf\_public1\_assoc: Creating...

module.vpc.aws\_route\_table\_association.tf\_public2\_assoc: Creation complete after 0s [id=rtbassoc-03a8951deff1fc8b1]

module.vpc.aws\_security\_group.tf\_public\_sg: Creation complete after 4s [id=sg-0bf916fa85bacb5e1]

module.compute.aws\_instance.k8s-master: Creating...

module.compute.aws\_instance.k8s-node: Creating...

module.vpc.aws\_route\_table\_association.tf\_public1\_assoc: Creation complete after 0s [id=rtbassoc-0c3131607be9fa33f]

module.compute.aws\_instance.k8s-node: Still creating... [10s elapsed]

module.compute.aws\_instance.k8s-master: Still creating... [10s elapsed]

module.compute.aws\_instance.k8s-master: Still creating... [20s elapsed]

module.compute.aws\_instance.k8s-node: Still creating... [20s elapsed]

module.compute.aws\_instance.k8s-master: Creation complete after 25s [id=i-06b786ace473a2623]

module.compute.aws\_instance.k8s-node: Still creating... [30s elapsed]

module.compute.aws\_instance.k8s-node: Still creating... [40s elapsed]

module.compute.aws\_instance.k8s-node: Still creating... [50s elapsed]

module.compute.aws\_instance.k8s-node: Creation complete after 55s [id=i-0ee766f322b982ae7]

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.

Outputs:

k8s-master-Public-IP = "54.82.122.90"

k8s-node-Public-IP = "3.86.63.147"

Text

Description automatically generated

## Helm Version:

Graphical user interface, text

Description automatically generated

## Website:

### Master:

Graphical user interface, text, application

Description automatically generated

### Worker:

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

## First Dashboard

### Master:

Graphical user interface, application

Description automatically generated

### Worker:

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence