Assessment 1 – Output:

**VPC and Dynamic Subnets with availability Zone**

chasrini@WKMZTA5DFE41 project % terraform plan

**module.a1-network.data.aws\_availability\_zones.available: Reading...**

**module.a1-network.data.aws\_availability\_zones.available: Read complete after 1s [id=us-east-1]**

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.a1-network.aws\_subnet.vpc-a1-subnets[0]** will be created

+ resource "aws\_subnet" "vpc-a1-subnets" {

+ 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"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

}

**# module.a1-network.aws\_subnet.vpc-a1-subnets[1]** will be created

+ resource "aws\_subnet" "vpc-a1-subnets" {

+ 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"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

}

**# module.a1-network.aws\_vpc.vpc-a1** will be created

+ resource "aws\_vpc" "vpc-a1" {

+ arn = (known after apply)

+ 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\_dns\_hostnames = (known after apply)

+ enable\_dns\_support = true

+ enable\_network\_address\_usage\_metrics = (known after apply)

+ id = (known after apply)

+ instance\_tenancy = "default"

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

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

+ ipv6\_cidr\_block\_network\_border\_group = (known after apply)

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

+ owner\_id = (known after apply)

+ tags = {

+ "Environment" = "dev"

+ "Name" = "my-vpc"

}

+ tags\_all = {

+ "Environment" = "dev"

+ "Name" = "my-vpc"

}

}

**Plan:** 3 to add, 0 to change, 0 to destroy.

**Terraform Workspaces**

chasrini@WKMZTA5DFE41 project % terraform workspace list

default

\* dev

prod

test

Only one subnet for dev workspace (output below)

hasrini@WKMZTA5DFE41 project % terraform workspace select dev

Switched to workspace "dev".

chasrini@WKMZTA5DFE41 project % terraform plan

module.a1-network.data.aws\_availability\_zones.available: Reading...

module.a1-network.data.aws\_availability\_zones.available: Read complete after 2s [id=us-east-1]

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.a1-network.aws\_subnet.vpc-a1-subnets[0] will be created

+ resource "aws\_subnet" "vpc-a1-subnets" {

+ 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"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

}

# module.a1-network.aws\_vpc.vpc-a1 will be created

+ resource "aws\_vpc" "vpc-a1" {

+ arn = (known after apply)

+ 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\_dns\_hostnames = (known after apply)

+ enable\_dns\_support = true

+ enable\_network\_address\_usage\_metrics = (known after apply)

+ id = (known after apply)

+ instance\_tenancy = "default"

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

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

+ ipv6\_cidr\_block\_network\_border\_group = (known after apply)

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

+ owner\_id = (known after apply)

+ tags = {

+ "Environment" = "dev"

+ "Name" = "my-vpc"

}

+ tags\_all = {

+ "Environment" = "dev"

+ "Name" = "my-vpc"

}

}

Plan: 2 to add, 0 to change, 0 to destroy.

──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply"

now.

chasrini@WKMZTA5DFE41 project %

Two subnets for dev workspace (output below)

chasrini@WKMZTA5DFE41 project % terraform workspace select prod

Switched to workspace "prod".

chasrini@WKMZTA5DFE41 project % terraform plan

**module.a1-network.data.aws\_availability\_zones.available: Reading...**

**module.a1-network.data.aws\_availability\_zones.available: Read complete after 2s [id=us-east-1]**

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.a1-network.aws\_subnet.vpc-a1-subnets[0]** will be created

+ resource "aws\_subnet" "vpc-a1-subnets" {

+ 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"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

}

**# module.a1-network.aws\_subnet.vpc-a1-subnets[1]** will be created

+ resource "aws\_subnet" "vpc-a1-subnets" {

+ 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"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

}

**# module.a1-network.aws\_vpc.vpc-a1** will be created

+ resource "aws\_vpc" "vpc-a1" {

+ arn = (known after apply)

+ 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\_dns\_hostnames = (known after apply)

+ enable\_dns\_support = true

+ enable\_network\_address\_usage\_metrics = (known after apply)

+ id = (known after apply)

+ instance\_tenancy = "default"

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

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

+ ipv6\_cidr\_block\_network\_border\_group = (known after apply)

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

+ owner\_id = (known after apply)

+ tags = {

+ "Environment" = "prod"

+ "Name" = "my-vpc"

}

+ tags\_all = {

+ "Environment" = "prod"

+ "Name" = "my-vpc"

}

}

**Plan:** 3 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

chasrini@WKMZTA5DFE41 project %

Terraform output

chasrini@WKMZTA5DFE41 project % terraform output

vpc\_id = "vpc-0cfd71f818f8f4066"

chasrini@WKMZTA5DFE41 project % terraform output vpc\_id

"vpc-0cfd71f818f8f4066"

**Assessment 2**

terraform plan

Acquiring state lock. This may take a few moments...

module.a2-network.data.aws\_availability\_zones.available: Reading...

module.a2-network.data.aws\_availability\_zones.available: Read complete after 1s [id=us-east-1]

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.myec2s.aws\_instance.appserver will be created

+ resource "aws\_instance" "appserver" {

+ ami = "ami-0230bd60aa48260c6"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

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

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

+ disable\_api\_stop = (known after apply)

+ disable\_api\_termination = (known after apply)

+ ebs\_optimized = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ host\_resource\_group\_arn = (known after apply)

+ iam\_instance\_profile = (known after apply)

+ id = (known after apply)

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

+ instance\_lifecycle = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.micro"

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

+ ipv6\_addresses = (known after apply)

+ key\_name = (known after apply)

+ monitoring = (known after apply)

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ placement\_partition\_number = (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

+ spot\_instance\_request\_id = (known after apply)

+ subnet\_id = (known after apply)

+ tags = {

+ "Environment" = "prod"

+ "Name" = "appserver"

+ "Terraform" = "true"

}

+ tags\_all = {

+ "Environment" = "prod"

+ "Name" = "appserver"

+ "Terraform" = "true"

}

+ tenancy = (known after apply)

+ user\_data = (known after apply)

+ user\_data\_base64 = (known after apply)

+ user\_data\_replace\_on\_change = false

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

}

# module.myec2s.aws\_instance.dbserver will be created

+ resource "aws\_instance" "dbserver" {

+ ami = "ami-0230bd60aa48260c6"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

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

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

+ disable\_api\_stop = (known after apply)

+ disable\_api\_termination = (known after apply)

+ ebs\_optimized = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ host\_resource\_group\_arn = (known after apply)

+ iam\_instance\_profile = (known after apply)

+ id = (known after apply)

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

+ instance\_lifecycle = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.micro"

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

+ ipv6\_addresses = (known after apply)

+ key\_name = (known after apply)

+ monitoring = (known after apply)

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ placement\_partition\_number = (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

+ spot\_instance\_request\_id = (known after apply)

+ subnet\_id = (known after apply)

+ tags = {

+ "Environment" = "prod"

+ "Name" = "dbserver"

+ "Terraform" = "true"

}

+ tags\_all = {

+ "Environment" = "prod"

+ "Name" = "dbserver"

+ "Terraform" = "true"

}

+ tenancy = (known after apply)

+ user\_data = (known after apply)

+ user\_data\_base64 = (known after apply)

+ user\_data\_replace\_on\_change = false

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

}

# module.myec2s.aws\_instance.webserver will be created

+ resource "aws\_instance" "webserver" {

+ ami = "ami-0230bd60aa48260c6"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

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

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

+ disable\_api\_stop = (known after apply)

+ disable\_api\_termination = (known after apply)

+ ebs\_optimized = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ host\_resource\_group\_arn = (known after apply)

+ iam\_instance\_profile = (known after apply)

+ id = (known after apply)

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

+ instance\_lifecycle = (known after apply)

+ instance\_state = (known after apply)

+ instance\_type = "t2.micro"

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

+ ipv6\_addresses = (known after apply)

+ key\_name = (known after apply)

+ monitoring = (known after apply)

+ outpost\_arn = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ placement\_partition\_number = (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

+ spot\_instance\_request\_id = (known after apply)

+ subnet\_id = (known after apply)

+ tags = {

+ "Environment" = "prod"

+ "Name" = "webserver"

+ "Terraform" = "true"

}

+ tags\_all = {

+ "Environment" = "prod"

+ "Name" = "webserver"

+ "Terraform" = "true"

}

+ tenancy = (known after apply)

+ user\_data = (known after apply)

+ user\_data\_base64 = (known after apply)

+ user\_data\_replace\_on\_change = false

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

}

# module.myec2s.aws\_security\_group.sg\_mytraffic will be created

+ resource "aws\_security\_group" "sg\_mytraffic" {

+ arn = (known after apply)

+ description = "Managed by Terraform"

+ egress = (known after apply)

+ id = (known after apply)

+ ingress = [

+ {

+ cidr\_blocks = [

+ "0.0.0.0/0",

]

+ description = ""

+ from\_port = 80

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 80

},

]

+ name = "sg\_allow\_webserver\_traffic"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_db\_subnet\_group.database[0] will be created

+ resource "aws\_db\_subnet\_group" "database" {

+ arn = (known after apply)

+ description = "Database subnet group for a2-vpc"

+ id = (known after apply)

+ name = "a2-vpc"

+ name\_prefix = (known after apply)

+ subnet\_ids = (known after apply)

+ supported\_network\_types = (known after apply)

+ tags = {

+ "Name" = "a2-vpc"

}

+ tags\_all = {

+ "Name" = "a2-vpc"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_default\_network\_acl.this[0] will be created

+ resource "aws\_default\_network\_acl" "this" {

+ arn = (known after apply)

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

+ id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-default"

}

+ tags\_all = {

+ "Name" = "a2-vpc-default"

}

+ vpc\_id = (known after apply)

+ egress {

+ action = "allow"

+ from\_port = 0

+ ipv6\_cidr\_block = "::/0"

+ protocol = "-1"

+ rule\_no = 101

+ to\_port = 0

}

+ egress {

+ action = "allow"

+ cidr\_block = "0.0.0.0/0"

+ from\_port = 0

+ protocol = "-1"

+ rule\_no = 100

+ to\_port = 0

}

+ ingress {

+ action = "allow"

+ from\_port = 0

+ ipv6\_cidr\_block = "::/0"

+ protocol = "-1"

+ rule\_no = 101

+ to\_port = 0

}

+ ingress {

+ action = "allow"

+ cidr\_block = "0.0.0.0/0"

+ from\_port = 0

+ protocol = "-1"

+ rule\_no = 100

+ to\_port = 0

}

}

# module.a2-network.module.vpc.aws\_default\_route\_table.default[0] will be created

+ resource "aws\_default\_route\_table" "default" {

+ arn = (known after apply)

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

+ id = (known after apply)

+ owner\_id = (known after apply)

+ route = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-default"

}

+ tags\_all = {

+ "Name" = "a2-vpc-default"

}

+ vpc\_id = (known after apply)

+ timeouts {

+ create = "5m"

+ update = "5m"

}

}

# module.a2-network.module.vpc.aws\_default\_security\_group.this[0] will be created

+ resource "aws\_default\_security\_group" "this" {

+ arn = (known after apply)

+ description = (known after apply)

+ egress = (known after apply)

+ id = (known after apply)

+ ingress = (known after apply)

+ name = (known after apply)

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "a2-vpc-default"

}

+ tags\_all = {

+ "Name" = "a2-vpc-default"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_internet\_gateway.this[0] will be created

+ resource "aws\_internet\_gateway" "this" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "a2-vpc"

}

+ tags\_all = {

+ "Name" = "a2-vpc"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route.public\_internet\_gateway[0] will be created

+ resource "aws\_route" "public\_internet\_gateway" {

+ destination\_cidr\_block = "0.0.0.0/0"

+ gateway\_id = (known after apply)

+ id = (known after apply)

+ instance\_id = (known after apply)

+ instance\_owner\_id = (known after apply)

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

+ origin = (known after apply)

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

+ state = (known after apply)

+ timeouts {

+ create = "5m"

}

}

# module.a2-network.module.vpc.aws\_route\_table.private[0] will be created

+ resource "aws\_route\_table" "private" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ propagating\_vgws = (known after apply)

+ route = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-private"

}

+ tags\_all = {

+ "Name" = "a2-vpc-private"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table.public[0] will be created

+ resource "aws\_route\_table" "public" {

+ arn = (known after apply)

+ id = (known after apply)

+ owner\_id = (known after apply)

+ propagating\_vgws = (known after apply)

+ route = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-public"

}

+ tags\_all = {

+ "Name" = "a2-vpc-public"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table\_association.database[0] will be created

+ resource "aws\_route\_table\_association" "database" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table\_association.database[1] will be created

+ resource "aws\_route\_table\_association" "database" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table\_association.private[0] will be created

+ resource "aws\_route\_table\_association" "private" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table\_association.private[1] will be created

+ resource "aws\_route\_table\_association" "private" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table\_association.public[0] will be created

+ resource "aws\_route\_table\_association" "public" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_route\_table\_association.public[1] will be created

+ resource "aws\_route\_table\_association" "public" {

+ id = (known after apply)

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

+ subnet\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_subnet.database[0] will be created

+ resource "aws\_subnet" "database" {

+ arn = (known after apply)

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

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

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

+ cidr\_block = "172.20.21.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-db-us-east-1a"

}

+ tags\_all = {

+ "Name" = "a2-vpc-db-us-east-1a"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_subnet.database[1] will be created

+ resource "aws\_subnet" "database" {

+ arn = (known after apply)

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

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

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

+ cidr\_block = "172.20.22.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-db-us-east-1b"

}

+ tags\_all = {

+ "Name" = "a2-vpc-db-us-east-1b"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_subnet.private[0] will be created

+ resource "aws\_subnet" "private" {

+ arn = (known after apply)

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

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

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

+ cidr\_block = "172.20.11.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-private-us-east-1a"

}

+ tags\_all = {

+ "Name" = "a2-vpc-private-us-east-1a"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_subnet.private[1] will be created

+ resource "aws\_subnet" "private" {

+ arn = (known after apply)

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

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

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

+ cidr\_block = "172.20.12.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-private-us-east-1b"

}

+ tags\_all = {

+ "Name" = "a2-vpc-private-us-east-1b"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_subnet.public[0] will be created

+ resource "aws\_subnet" "public" {

+ arn = (known after apply)

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

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

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

+ cidr\_block = "172.20.1.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-public-us-east-1a"

}

+ tags\_all = {

+ "Name" = "a2-vpc-public-us-east-1a"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_subnet.public[1] will be created

+ resource "aws\_subnet" "public" {

+ arn = (known after apply)

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

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

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

+ cidr\_block = "172.20.2.0/24"

+ enable\_dns64 = false

+ enable\_resource\_name\_dns\_a\_record\_on\_launch = false

+ enable\_resource\_name\_dns\_aaaa\_record\_on\_launch = false

+ id = (known after apply)

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

+ ipv6\_native = false

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

+ owner\_id = (known after apply)

+ private\_dns\_hostname\_type\_on\_launch = (known after apply)

+ tags = {

+ "Name" = "a2-vpc-public-us-east-1b"

}

+ tags\_all = {

+ "Name" = "a2-vpc-public-us-east-1b"

}

+ vpc\_id = (known after apply)

}

# module.a2-network.module.vpc.aws\_vpc.this[0] will be created

+ resource "aws\_vpc" "this" {

+ arn = (known after apply)

+ cidr\_block = "172.20.0.0/16"

+ 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\_dns\_hostnames = true

+ enable\_dns\_support = true

+ enable\_network\_address\_usage\_metrics = (known after apply)

+ id = (known after apply)

+ instance\_tenancy = "default"

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

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

+ ipv6\_cidr\_block\_network\_border\_group = (known after apply)

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

+ owner\_id = (known after apply)

+ tags = {

+ "Name" = "a2-vpc"

}

+ tags\_all = {

+ "Name" = "a2-vpc"

}

}

Plan: 25 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ vpc\_id = (known after apply)

──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply"

now.

Releasing state lock. This may take a few moments...

chasrini@WKMZTA5DFE41 project %