ilgea@LAPTOP-H7GFBQRB~/GIT/terraform-projeler/deneme:$ terraform plan

data.aws\_vpc.default: Reading...

data.aws\_ami.amazon-linux-2: Reading...

data.aws\_ami.amazon-linux-2: Read complete after 1s [id=ami-09d3b3274b6c5d4aa]

data.aws\_vpc.default: Read complete after 1s [id=vpc-00a91f9e85797580d]

data.aws\_subnets.subnet: Reading...

data.aws\_subnets.subnet: Read complete after 0s [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:

# aws\_autoscaling\_group.WebserverASG will be created

+ resource "aws\_autoscaling\_group" "WebserverASG" {

+ arn = (known after apply)

+ availability\_zones = (known after apply)

+ default\_cooldown = (known after apply)

+ desired\_capacity = 2

+ force\_delete = false

+ force\_delete\_warm\_pool = false

+ health\_check\_grace\_period = 300

+ health\_check\_type = "ELB"

+ id = (known after apply)

+ max\_size = 3

+ metrics\_granularity = "1Minute"

+ min\_size = 1

+ name = "phonebook-asg"

+ name\_prefix = (known after apply)

+ protect\_from\_scale\_in = false

+ service\_linked\_role\_arn = (known after apply)

+ target\_group\_arns = (known after apply)

+ vpc\_zone\_identifier = [

+ "subnet-036892f92bdc3dc1a",

+ "subnet-095b8202438f5378a",

+ "subnet-0a10a901b2fb7c029",

+ "subnet-0b414da436134e0d7",

+ "subnet-0bcab4dc5c83246e2",

+ "subnet-0ed2cb564211ad3dc",

]

+ wait\_for\_capacity\_timeout = "10m"

+ launch\_template {

+ id = (known after apply)

+ name = (known after apply)

+ version = (known after apply)

}

}

# aws\_db\_instance.rds will be created

+ resource "aws\_db\_instance" "rds" {

+ address = (known after apply)

+ allocated\_storage = 20

+ apply\_immediately = (known after apply)

+ arn = (known after apply)

+ auto\_minor\_version\_upgrade = true

+ availability\_zone = (known after apply)

+ backup\_retention\_period = (known after apply)

+ backup\_window = (known after apply)

+ ca\_cert\_identifier = (known after apply)

+ character\_set\_name = (known after apply)

+ copy\_tags\_to\_snapshot = false

+ db\_name = "phonebook"

+ db\_subnet\_group\_name = (known after apply)

+ delete\_automated\_backups = true

+ endpoint = (known after apply)

+ engine = "mysql"

+ engine\_version = "8.0.25"

+ engine\_version\_actual = (known after apply)

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

+ id = (known after apply)

+ identifier = "phonebook-app-db"

+ identifier\_prefix = (known after apply)

+ instance\_class = "db.t2.micro"

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

+ latest\_restorable\_time = (known after apply)

+ license\_model = (known after apply)

+ maintenance\_window = (known after apply)

+ monitoring\_interval = 0

+ monitoring\_role\_arn = (known after apply)

+ multi\_az = (known after apply)

+ name = (known after apply)

+ nchar\_character\_set\_name = (known after apply)

+ network\_type = (known after apply)

+ option\_group\_name = (known after apply)

+ parameter\_group\_name = (known after apply)

+ password = (sensitive value)

+ performance\_insights\_enabled = false

+ performance\_insights\_kms\_key\_id = (known after apply)

+ performance\_insights\_retention\_period = (known after apply)

+ port = 3306

+ publicly\_accessible = false

+ replica\_mode = (known after apply)

+ replicas = (known after apply)

+ resource\_id = (known after apply)

+ skip\_final\_snapshot = true

+ snapshot\_identifier = (known after apply)

+ status = (known after apply)

+ storage\_type = (known after apply)

+ tags\_all = (known after apply)

+ timezone = (known after apply)

+ username = "admin"

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

}

# aws\_launch\_template.my-temp will be created

+ resource "aws\_launch\_template" "my-temp" {

+ arn = (known after apply)

+ default\_version = (known after apply)

+ id = (known after apply)

+ image\_id = "ami-09d3b3274b6c5d4aa"

+ instance\_type = "t2.micro"

+ key\_name = "firstkey"

+ latest\_version = (known after apply)

+ name = "phonebook-lt"

+ name\_prefix = (known after apply)

+ tags\_all = (known after apply)

+ user\_data = (known after apply)

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

+ metadata\_options {

+ http\_endpoint = (known after apply)

+ http\_protocol\_ipv6 = (known after apply)

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

+ http\_tokens = (known after apply)

+ instance\_metadata\_tags = (known after apply)

}

+ placement {}

+ tag\_specifications {

+ resource\_type = "instance"

+ tags = {

+ "Name" = "Web server of Phonebook App"

}

}

}

# aws\_lb.alb will be created

+ resource "aws\_lb" "alb" {

+ arn = (known after apply)

+ arn\_suffix = (known after apply)

+ desync\_mitigation\_mode = "defensive"

+ dns\_name = (known after apply)

+ drop\_invalid\_header\_fields = false

+ enable\_deletion\_protection = false

+ enable\_http2 = true

+ enable\_waf\_fail\_open = false

+ id = (known after apply)

+ idle\_timeout = 60

+ internal = false

+ ip\_address\_type = "ipv4"

+ load\_balancer\_type = "application"

+ name = "phonebook-app"

+ preserve\_host\_header = false

+ security\_groups = (known after apply)

+ subnets = [

+ "subnet-036892f92bdc3dc1a",

+ "subnet-095b8202438f5378a",

+ "subnet-0a10a901b2fb7c029",

+ "subnet-0b414da436134e0d7",

+ "subnet-0bcab4dc5c83246e2",

+ "subnet-0ed2cb564211ad3dc",

]

+ tags\_all = (known after apply)

+ vpc\_id = (known after apply)

+ zone\_id = (known after apply)

+ subnet\_mapping {

+ allocation\_id = (known after apply)

+ ipv6\_address = (known after apply)

+ outpost\_id = (known after apply)

+ private\_ipv4\_address = (known after apply)

+ subnet\_id = (known after apply)

}

}

# aws\_lb\_listener.listener\_http will be created

+ resource "aws\_lb\_listener" "listener\_http" {

+ arn = (known after apply)

+ id = (known after apply)

+ load\_balancer\_arn = (known after apply)

+ port = 80

+ protocol = "HTTP"

+ ssl\_policy = (known after apply)

+ tags\_all = (known after apply)

+ default\_action {

+ order = (known after apply)

+ target\_group\_arn = (known after apply)

+ type = "forward"

}

}

# aws\_lb\_target\_group.turgut will be created

+ resource "aws\_lb\_target\_group" "turgut" {

+ arn = (known after apply)

+ arn\_suffix = (known after apply)

+ connection\_termination = false

+ deregistration\_delay = "300"

+ id = (known after apply)

+ ip\_address\_type = (known after apply)

+ lambda\_multi\_value\_headers\_enabled = false

+ load\_balancing\_algorithm\_type = (known after apply)

+ name = "phonebook-tg"

+ port = 80

+ preserve\_client\_ip = (known after apply)

+ protocol = "HTTP"

+ protocol\_version = (known after apply)

+ proxy\_protocol\_v2 = false

+ slow\_start = 0

+ tags\_all = (known after apply)

+ target\_type = "instance"

+ vpc\_id = "vpc-00a91f9e85797580d"

+ health\_check {

+ enabled = true

+ healthy\_threshold = 2

+ interval = 10

+ matcher = (known after apply)

+ path = (known after apply)

+ port = "traffic-port"

+ protocol = "HTTP"

+ timeout = (known after apply)

+ unhealthy\_threshold = 3

}

+ stickiness {

+ cookie\_duration = (known after apply)

+ cookie\_name = (known after apply)

+ enabled = (known after apply)

+ type = (known after apply)

}

}

# aws\_security\_group.alb-sg will be created

+ resource "aws\_security\_group" "alb-sg" {

+ arn = (known after apply)

+ description = "Managed by Terraform"

+ 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 = 80

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 80

},

]

+ name = "ALBSecurityGroup"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "ALBSecurityGroup"

}

+ tags\_all = {

+ "Name" = "ALBSecurityGroup"

}

+ vpc\_id = "vpc-00a91f9e85797580d"

}

# aws\_security\_group.db-sec-grp will be created

+ resource "aws\_security\_group" "db-sec-grp" {

+ arn = (known after apply)

+ description = "Managed by Terraform"

+ 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 = []

+ description = ""

+ from\_port = 3306

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = (known after apply)

+ self = false

+ to\_port = 3306

},

]

+ name = "rds-sec-grp"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "rds-sec-grp"

}

+ tags\_all = {

+ "Name" = "rds-sec-grp"

}

+ vpc\_id = "vpc-00a91f9e85797580d"

}

# aws\_security\_group.ec2-sec-grp will be created

+ resource "aws\_security\_group" "ec2-sec-grp" {

+ arn = (known after apply)

+ description = "Managed by Terraform"

+ 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 = 22

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = []

+ self = false

+ to\_port = 22

},

+ {

+ cidr\_blocks = []

+ description = ""

+ from\_port = 80

+ ipv6\_cidr\_blocks = []

+ prefix\_list\_ids = []

+ protocol = "tcp"

+ security\_groups = (known after apply)

+ self = false

+ to\_port = 80

},

]

+ name = "webserver-sec-grp"

+ name\_prefix = (known after apply)

+ owner\_id = (known after apply)

+ revoke\_rules\_on\_delete = false

+ tags = {

+ "Name" = "webserver-sec-grp"

}

+ tags\_all = {

+ "Name" = "webserver-sec-grp"

}

+ vpc\_id = "vpc-00a91f9e85797580d"

}

Plan: 9 to add, 0 to change, 0 to destroy.

Changes to Outputs:

+ lb-dns-name = (known after apply)

+ rds-endpoint = (known after apply)