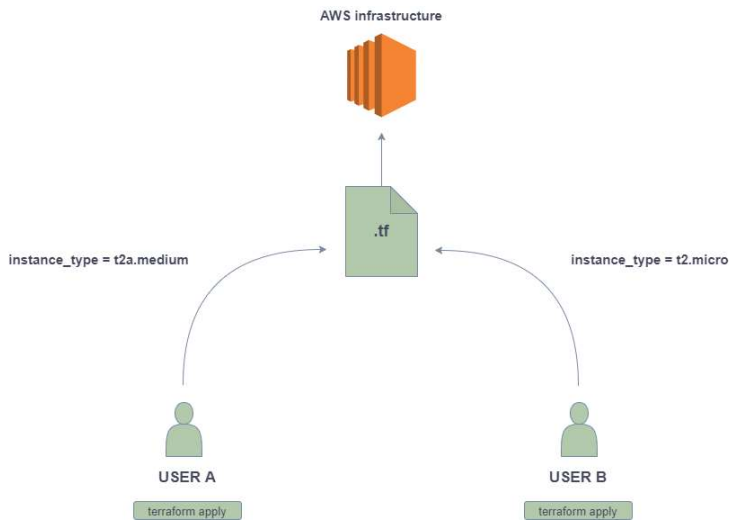


Handson 3

Saturday, 4 September 2021 11.06



Backend olarak kullanmak için bucket ve dynamodb kurarak başlatmaya çalıştık bunları kendi terraform config file'i ile tanımlayacağız.
Benim bucketimiz backend olarak kullanılabileceğiz

Normal hayatta force destroy kullanmak riskli kazayla silinmesini engellemek için başka bir argüman kullanacağız

Hands-on Terraform-03 : Terraform Remote Backend and Modules.

Purpose of this hands-on training is to give students the knowledge of remote backend and modules in Terraform.

Learning Outcomes

At the end of this hands-on training, students will be able to;

- Create a remote backend and use modules.

Terraform Remote State (Remote backend)

- A `backend` in Terraform determines how the `tfstate` file is loaded/stored and how an operation such as `apply` is executed. This abstraction enables non-local file state storage, remote execution, etc. By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to.

- Go to the AWS console and attach `DynamoDBFullAccess` policy to the existing role.

![[state-locking]](state-locking.png)

- Create a new folder named `s3-backend` and a file named `backend.tf`.

```
```\ntxt\n  s3-backend\n    └─ backend.tf\n  terraform-aws\n    ├── oliver.tfvars\n    ├── main.tf\n    └─ variables.tf\n```\n
```

- Go to the `s3-`

`backend` folder and create a file named `backend.tf`. Add the followings.

```
```\nbash\n  cd .. && mkdir s3-backend && cd s3-backend && touch backend.tf\n```\n\n```\nbash\n  provider "aws" {\n    region = "us-east-1"\n  }\n  resource "aws_s3_bucket" "tf-remote-state" {\n
```

```

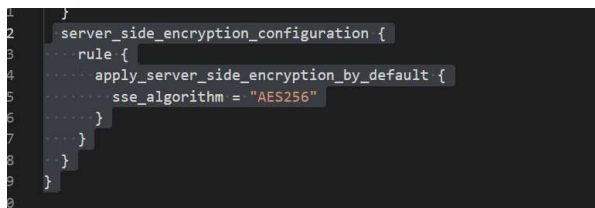
bucket = "tf-remote-s3-bucket-oliver-changehere"

force_destroy = true
versioning {
  enabled = true
}
server_side_encryption_configuration {
  rule {
    apply_server_side_encryption_by_default {
      sse_algorithm = "AES256"
    }
  }
}
}
}
resource "aws_dynamodb_table" "tf-remote-state_lock" {
  hash_key = "LockID"
  name = "tf-s3-app-lock"
  attribute {
    name = "LockID"
    type = "S"
  }
  billing_mode = "PAY_PER_REQUEST"
}
...

- Run the commands below.
```bash
terraform init
terraform apply
```

- We have created a S3 bucket and a Dynamodb table. Now associate S3 bucket with the Dynamodb table.
- Go to the `main.tf` file make the changes.
```bash
terraform {
 required_providers {
 aws = {
 source = "hashicorp/aws"
 version = "3.56.0"
 }
 }
 backend "s3" {
 bucket = "tf-remote-s3-bucket-oliver-changehere"
 key = "env/dev/tf-remote-backend.tfstate"
 region = "us-east-1"
 dynamodb_table = "tf-s3-app-lock"
 encrypt = true
 }
}

```



```

1 }
2 server_side_encryption_configuration {
3 rule {
4 apply_server_side_encryption_by_default {
5 sse_algorithm = "AES256"
6 }
7 }
8 }
9 }

```

Kriptolöamayı dokumantasyondan alıyoruz

Simdi yukardaki komutlarla dynomo db yi olusturuyoruz

`

```

20
21 resource "aws_dynamodb_table" "tf-remote-state-lock" {
22 hash_key = "LockID"
23 name = "tf-s3-app-lock"
24 attribute {
25 name = "LockID"
26 type = "S"
27 }
28 billing_mode = "PAY_PER_REQUEST"
29 }
30
31

```

Hask key de LockId yazmak zorunlu

Statelock ozelligi icin onemli

Dynamodb kullanarak stateimiz uzerinde ayni anda baskasi da islem yapabiliyor, iki tarafta ayni anda terradorm apply a basiaınca eger lock ozelligi yoksa yanlış hatalı vb işlemler yapılabilir statelock ozelligi bu durumlarda onemli. Bu işlemi yaptığımız zaman ilk önce terraform applya basanın işlemini yapıyor sonra sıradaki diğer kişinin işlemini yapıyor. İşlem yaparken diğerinin yaptığı işlemi beklemeye aliyor kilitliyor.

Önce rolümüze dynamo db ekleyeceğiz

The screenshot shows the AWS IAM console for the role 'Terraform\_ec2\_role'. The 'Summary' tab is active, displaying the role's ARN, description, instance profile ARNs, path, creation time, last activity, and maximum session duration. The 'Permissions' tab is also shown, indicating that 4 policies are applied. The policies listed are AmazonEC2FullAccess, IAMFullAccess, AmazonS3FullAccess, and AmazonDynamoDBFullAccess, all of which are AWS managed policies. The permissions boundary is not set.

Go to the `terraform-aws` directory and run the commands below. First try to terraform apply command.

```

```bash
- cd ../terraform-aws
```
```bash
terraform apply
terraform init
terraform apply

```

Sirasiyla islemleri yapiyoruz.s3 backend klasoru icinden terminal acip

```
31
TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE
- Installed hashicorp/aws v3.57.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-88-43 s3-backend]$
```

```
28   billing_mode = "PAY_PER_REQUEST"
29   }
30 }
31

}
}

+ versioning {
+   enabled      = true
+   mfa_delete   = false
+ }
}

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.
[ec2-user@ip-172-31-88-43 s3-backend]$
```

```
TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE
+ enabled      = true
+ mfa_delete   = false
}
}

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

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

aws_s3_bucket.tf-remote-state: Creating...
aws_dynamodb_table.tf-remote-state-lock: Creating...
aws_s3_bucket.tf-remote-state: Creation complete after 1s [id=tf-remote-s3-bucket-serkans]
```

Bucket olusuyor

...

Apps Gmail Lists Questions and... Your first Python pr... 20KN_1576883390... Clarusway-dev-750... AWS Management... Python conditional... Home | CodeWars 5. Data Structures... Python Tutorial for... Opintopolku - haku... Reading list

Services Search for services, features, marketplace products, and docs [Alt+S] Admin_user @ ser007 Global Support

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Account snapshot

Last updated: Sep 3, 2021 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)

[View Storage Lens dashboard](#)

Total storage	Object count	Avg. object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
425.8 KB	158	2.7 KB	

Buckets (4) Info

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	AWS Region	Access	Creation date
<input type="radio"/>	cf-templates-qlmtjxcxatjo-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	June 26, 2021, 13:27:45 (UTC+03:00)
<input type="radio"/>	cf-templates-qlmtjxcxatjo-us-east-2	US East (Ohio) us-east-2	Objects can be public	June 26, 2021, 13:43:27 (UTC+03:00)
<input type="radio"/>	tf-remote-s3-bucket-serkans	US East (N. Virginia) us-east-1	Objects can be public	September 4, 2021, 11:29:40 (UTC+03:00)
<input type="radio"/>	www.awsdevopssekan.com	US East (N. Virginia) us-east-1	Public	August 5, 2021, 21:36:46 (UTC+03:00)

Dyn amo db miz de olustu

Services Search for services, features, marketplace products, and docs [Alt+S] Admin_user @ ser007 N. Virginia Support

DynamoDB

Dashboard

Tables

- Items [New](#)
- PartiQL editor [New](#)
- Backups
- Exports to S3 [New](#)
- Reserved capacity

DAX

- Clusters
- Subnet groups
- Parameter groups
- Events

Tell us what you think

Tables (1) Info

Find tables by table name

Any table tag

	Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
<input type="checkbox"/>	tf-s3-app-lock	Active	LockID (String)	-	0	On-demand	On-demand

Daha onceki main.tf deki kaynaklari tekrar olusturuyoruz
Terraform.apply diyoruz

```
main.tf
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "tf-ec2" {
  ami           = "ami-023464620e0b2a980"
  instance_type = "t2.micro"
  subnet_id     = "subnet-0a1b2c3d"
  vpc_security_group_ids = ["sg-0a1b2c3d"]
  user_data     = <<-EOF
    #!/bin/bash
    echo "Hello, World!" > /tmp/hello.txt
  EOF
  tags = {
    Name = "tf-ec2"
  }
}

resource "aws_s3_bucket" "tf-s3" {
  bucket = "tf-s3"
  acl    = "public-read"
  tags = {
    Name = "tf-s3"
  }
}

resource "aws_iam_user" "new_users" {
  name = "tf-ec2"
  password = "tf-ec2"
  tags = {
    Name = "tf-ec2"
  }
}

output "tf-example-public-ip" {
  value = aws_instance.tf-ec2.public_ip
}

output "tf-example-private-ip" {
  value = aws_instance.tf-ec2.private_ip
}

output "tf-example-s3-meta" {
  value = aws_s3_bucket.tf-s3.region
}
```

```
[ec2-user@ip-172-31-88-43 terraform-aws]$ terraform apply
aws_s3_bucket.tf-s3["ramazan"]: Refreshing state... [id=example-s3-bucket-ramazan]
aws_s3_bucket.tf-s3["serkan"]: Refreshing state... [id=example-s3-bucket-serkan]
aws_iam_user.new_users["hamit"]: Refreshing state... [id=hamit]
aws_iam_user.new_users["ramazan"]: Refreshing state... [id=ramazan]
aws_iam_user.new_users["serkan"]: Refreshing state... [id=serkan]
aws_s3_bucket.tf-s3["hamit"]: Refreshing state... [id=example-s3-bucket-hamit]
aws_instance.tf-ec2: Refreshing state... [id=i-023464620e0b2a980]
```

Bucketlar tekrar geldi

Name	AWS Region	Access	Creation date
cf-templates-qlmtjxcxatjo-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	June 26, 2021, 13:27:45 (UTC+03:00)
cf-templates-qlmtjxcxatjo-us-east-2	US East (Ohio) us-east-2	Objects can be public	June 26, 2021, 13:43:27 (UTC+03:00)
example-s3-bucket-hamit	US East (N. Virginia) us-east-1	Objects can be public	September 4, 2021, 11:32:48 (UTC+03:00)
example-s3-bucket-ramazan	US East (N. Virginia) us-east-1	Objects can be public	September 4, 2021, 11:32:48 (UTC+03:00)
example-s3-bucket-serkan	US East (N. Virginia) us-east-1	Objects can be public	September 4, 2021, 11:32:48 (UTC+03:00)
tf-remote-s3-bucket-serkans	US East (N. Virginia) us-east-1	Objects can be public	September 4, 2021, 11:29:40 (UTC+03:00)
www.awsdevopsserkan.com	US East (N. Virginia) us-east-1	Public	August 5, 2021, 21:36:46 (UTC+03:00)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Terraform	i-043669202d2b5e15a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-3-92-205-95.compute-1.ar
serkan-local-name-come from locals	i-023464620e0b2a980	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-100-26-184-242.compute
Terraform-Learning	i-08dc7c507c04d5cc5	Stopped	t2.micro	-	No alarms	us-east-1d	-

Instance: i-023464620e0b2a980 (serkan-local-name-) come from my ami

Instance ID	Public IPv4 address	Private IPv4 addresses
i-023464620e0b2a980	ec2-100-26-184-242.compute-1.amazonaws.com	10.0.0.1

Ec2 muz da acildi

S3 un versionlama ozelligi var, her backup yaptigimizda statefile da yenileniyor
Statelock ozelligi de dynamo db ile saglaniyor

Son olarak bir tane bucketimizin arn sini output olarak gostermesini istedik,
tarraform apply yaptik.

```
81
82   output "s3-arn-2" {
83     value = aws_s3_bucket.tf-s3["serkan"].arn
84   }
}
```

TERMINAL PROBLEMS OUTPUT PORTS

TERMINAL

Only 'yes' will be accepted to approve.

Enter a value: yes

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

Outputs:

s3-arn-2 = "arn:aws:s3:::example-s3-bucket-serkan"
tf-example-private-ip = "172.31.94.54"
tf-example-public-ip = "100.26.184.242"
upper = [
 "RAMAZAN",
]
[ec2-user@ip-172-31-88-43 terraform-aws]\$

- Because of using S3 bucket for backend, run `terraform init` again. It will ask you to copy the existing tfstate file to s3. yes.

- Go to the `main.tf` file add the followings.

```
``bash
output "s3-arn-1" {
  value = aws_s3_bucket.tf-s3["fredo"].arn
}
``
```

Elde uc tane bucket oldugu icin ["...isim"] belirtmemiz oneli

iki tane terminal actik ayni main.tf de ikisinde sirasiyla entera bastik ama digerinde islem yaptigi icin state lock devreye girdi ve uyarı verdi

```
Operation: OperationTypePlan
Who:      ec2-user@ip-172-31-88-43.ec2.internal
Version:  1.0.5
Created:   2021-09-04 09:20:22.226809353 +0000 UTC
Info:

Terraform acquires a state lock to protect the state from
being written
by multiple users at the same time. Please resolve the issue
above and try
again. For most commands, you can disable locking with the
"-lock=false"
flag, but this is not recommended.

[ec2-user@ip-172-31-88-43 terraform-aws]$
```

```
aws_iam_user.new_users["hamit"]: Refreshing state... [id=hamit]
aws_instance.tf-ec2: Refreshing state... [id=i-023464620e0b2a98
0]

Changes to Outputs:
+ s3-arn-3 = "arn:aws:s3:::example-s3-bucket-ramazan"

You can apply this plan to save these new output values to the
Terraform state, without changing any real infrastructure.

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.

[ec2-user@ip-172-31-88-43 terraform-aws]$
```

Digerindeki islem bitince beklemede olanda islem yapabiliyoruz

```
configuration and found no differences, so no changes are
needed.

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

Outputs:

s3-arn-1 = "arn:aws:s3:::example-s3-bucket-hamit"
s3-arn-2 = "arn:aws:s3:::example-s3-bucket-serkan"
tf-example-private-ip = "172.31.94.54"
tf-example-public-ip = "100.26.184.242"
upper = [  
  "RAMAZAN",  
]  
[ec2-user@ip-172-31-88-43 terraform-aws]$
```

```
``bash
terraform apply
``
```

- Go to the AWS console and check the S3 bucket, tfstate file. tfstate file is copied from local to S3 backend.

- Go to the `main.tf` file make the changes (add another output).

```
``bash
```



```

output "s3-arn-2" {
  value = aws_s3_bucket.tf-s3["santino"].arn
}

```

- Open a new terminal. Write `terraform apply` in the both terminal. Try to run the command in both terminals at the same time.
- We do not get an error in the terminal that we run `terraform apply` command for the first time, but we get an error in the terminal we run later.
- Now you can try to run the same command with the second terminal. Check the Dynamo DB table and items.
- Destroy all resources.

```

```bash
terraform destroy
terraform destroy

```

```

run "terraform apply" now.
[ec2-user@ip-172-31-88-43 terraform-aws]$ terraform destroy

```

Yapip kaynaklari siliyoruz

**Buckets (4)** Info Refresh Copy ARN Empty Delete Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

Name	AWS Region	Access	Creation date
<input type="radio"/> cf-templates-qlmtjxcxatjo-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	June 26, 2021, 13:27:45 (UTC+03:00)
<input type="radio"/> cf-templates-qlmtjxcxatjo-us-east-2	US East (Ohio) us-east-2	Objects can be public	June 26, 2021, 13:43:27 (UTC+03:00)
<input type="radio"/> tf-remote-s3-bucket-serkans	US East (N. Virginia) us-east-1	Objects can be public	September 4, 2021, 11:29:40 (UTC+03:00)
<input type="radio"/> www.awsdevopsserkans.com	US East (N. Virginia) us-east-1	Public	August 5, 2021, 21:36:46 (UTC+03:00)

bucketlar gitmis

Instance dea destroy ediliyor

**Instances (1/3)** Info Refresh Connect Instance state Actions Launch instances

Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/> Terraform	i-043669202d2b5e15a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-3-92-205-95.compute-1.amazonaws.com
<input checked="" type="checkbox"/> serkan-local-name-come from locals	i-023464620e0b2a980	Shutting-down	t2.micro	-	No alarms	us-east-1d	ec2-100-26-184-242.compute-1.amazonaws.com
<input type="checkbox"/> Terraform-Learning	i-08dc7c507c04d5cc5	Stopped	t2.micro	-	No alarms	us-east-1d	-

**Instance: i-023464620e0b2a980 (serkan-local-name-come from locals)**

Instance ID i-023464620e0b2a980 (serkan-local-name-come from locals)	Public IPv4 address 100.26.184.242 <a href="#">open address</a>	Private IPv4 addresses 172.31.94.54
-------------------------------------------------------------------------	--------------------------------------------------------------------	----------------------------------------

Backend de olusturulanlari da orada terminal acip destroy ediyoruz

```

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

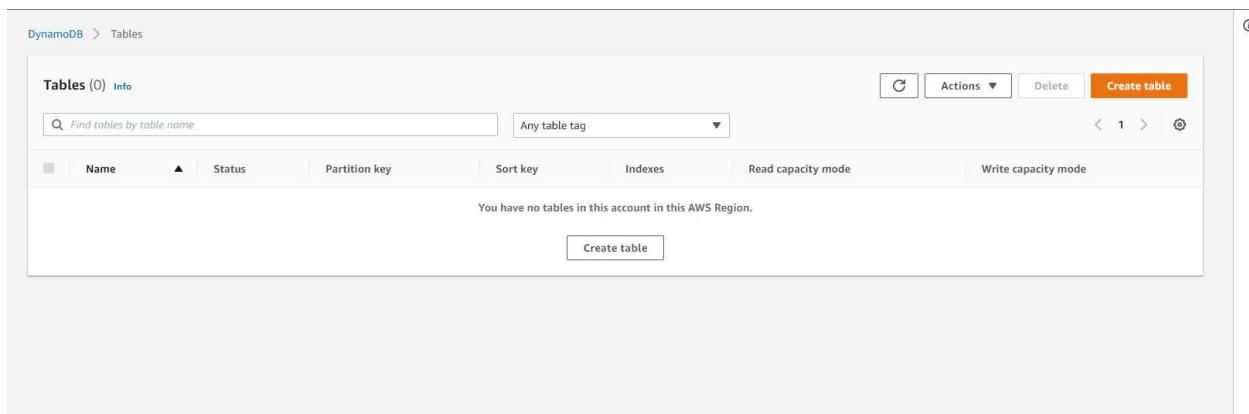
aws_s3_bucket.tf-remote-state: Destroying... [id=tf-remote-s3-bucket-serkans]
aws_dynamodb_table.tf-remote-state-lock: Destroying... [id=tf-s3-app-lock]
aws_s3_bucket.tf-remote-state: Destruction complete after 1s
aws_dynamodb_table.tf-remote-state-lock: Destruction complete after 4s

Destroy complete! Resources: 2 destroyed.
[ec2-user@ip-172-31-88-43 s3-backend]$

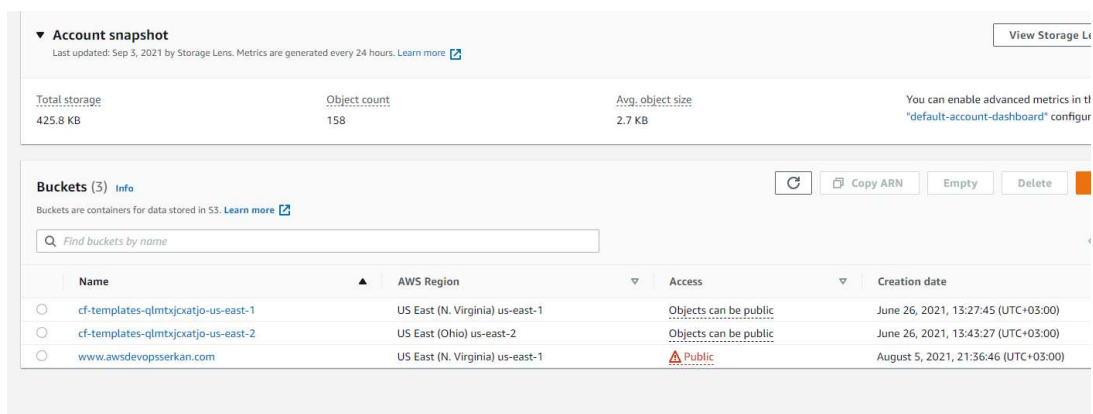
```

Dynamodb gitt

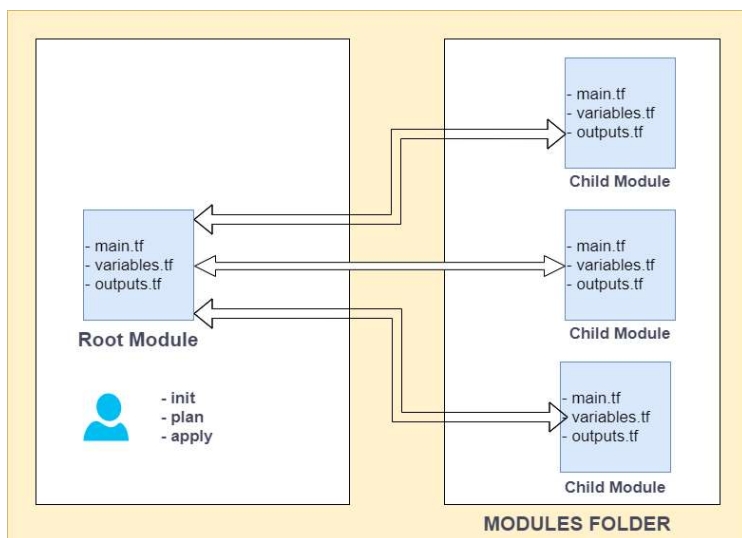




Backend bucket da gitti

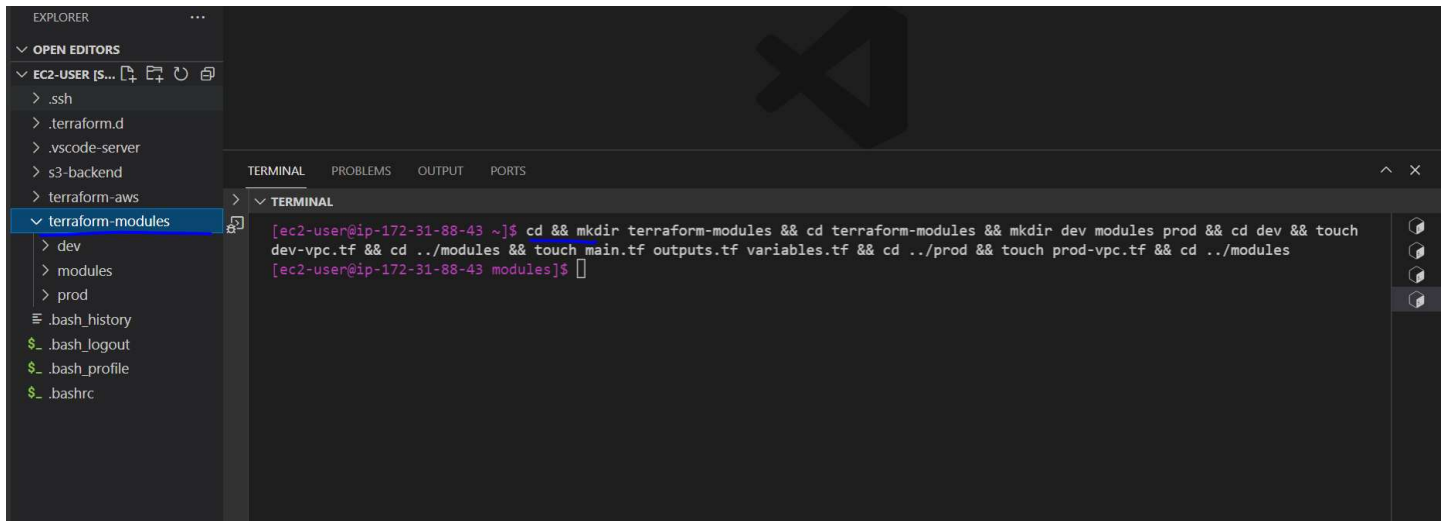


...



### ### Terraform Modules

```
-Create folders name `terraform-
modules`, `modules`, `dev`, `prod` directories in the home directory and files as
below.
```bash
cd && mkdir terraform-modules && cd terraform-
modules && mkdir dev modules prod && cd dev && touch dev-
vpc.tf && cd ../modules && touch main.tf outputs.tf variables.tf && cd ../prod &&
touch prod-vpc.tf && cd ../modules
```



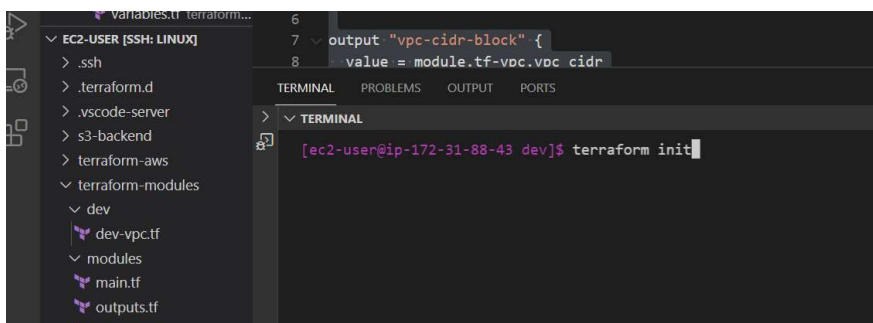
BURDA İKİ TANE AYRI VPC olusturacagiz

Developerlar dev de production prod da calisiyor

```
Sudo ``
```txt
terraform-modules
├── dev
│ └── dev-vpc.tf
├── modules
│ ├── main.tf
│ ├── outputs.tf
│ └── variables.tf
└── prod
 └── prod-vpc.tf
...

```

MODULE de mantik benim configaraoyonumu git surdan al diyoruz



```
![terraform modules](terraform-modules.png)
- Go to the `modules/main.tf` file, add the followings.
```bash
provider "aws" {
  region = "us-east-1"
}

resource "aws_vpc" "module_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
    Name = "terraform-vpc-${var.environment}"
  }
}

resource "aws_subnet" "public_subnet" {
  cidr_block = var.public_subnet_cidr
  vpc_id = aws_vpc.module_vpc.id
}

```

```

tags = {
  Name = "terraform-public-subnet-${var.environment}"
}
}
resource "aws_subnet" "private_subnet" {
  cidr_block = var.private_subnet_cidr
  vpc_id = aws_vpc.module_vpc.id
  tags = {
    Name = "terraform-private-subnet-${var.environment}"
  }
}
...
- Go to the `modules/variables.tf` file, add the followings.
```bash
variable "environment" {
 default = "oliver"
}
variable "vpc_cidr_block" {
 default = "10.0.0.0/16"
 description = "this is our vpc cidr block"
}
variable "public_subnet_cidr" {
 default = "10.0.1.0/24"
 description = "this is our public subnet cidr block"
}
variable "private_subnet_cidr" {
 default = "10.0.2.0/24"
 description = "this is our private subnet cidr block"
}
...
- Go to the `modules/outputs.tf` file, add the followings.
```bash
output "vpc_id" {
  value = aws_vpc.module_vpc.id
}
output "vpc_cidr" {
  value = aws_vpc.module_vpc.cidr_block
}
output "public_subnet_cidr" {
  value = aws_subnet.public_subnet.cidr_block
}
output "private_subnet_cidr" {
  value = aws_subnet.private_subnet.cidr_block
}
...
- Go to the `dev/dev-vpc.tf` file, add the followings.
```bash
module "tf-vpc" {
 source = "../modules"
 environment = "DEV"
}
output "vpc-cidr-block" {
 value = module.tf-vpc.vpc_cidr
}
...
- Go to the `prod/prod-vpc.tf` file, add the followings.
```bash
module "tf-vpc" {
  source = "../modules"
  environment = "PROD"
}
output "vpc-cidr-block" {
  value = module.tf-vpc.vpc_cidr
}
...
- Go to the `dev` folder and run the command below.
```bash
terraform init
terraform apply

```

Your VPCs (4) [Info](#)

Filter VPCs

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)	IPv6 pool	DHCP options set
aws_capstone-VPC	vpc-0515554f6b9fee688	Available	90.90.0.0/16	-	-	dopt-2c340d56
default-vpc	vpc-d9e37fa4	Available	172.31.0.0/16	-	-	dopt-2c340d56
clarus-vpc-a	vpc-09e43c478719d2b31	Available	10.7.0.0/16	-	-	dopt-2c340d56
terraform-vpc-DEV	vpc-030e1fee59c7eabe6	Available	10.0.0.0/16	-	-	dopt-2c340d56

Select a VPC above

VPC miz oludtu

Subnets (14) [Info](#)

Filter subnets

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
terraform-public-subnet-DEV	subnet-08a5572a28a84c9c1	Available	vpc-030e1fee59c7eabe6   terr...	10.0.1.0/24	-
terraform-private-subnet-DEV	subnet-03c7bed8d45f01845	Available	vpc-030e1fee59c7eabe6   terr...	10.0.2.0/24	-
clarus-az1c-public-subnet	subnet-0a1c0b0a47e8424b3	Available	vpc-09e43c478719d2b31   cla...	10.7.0.0/24	-
clarus-az1c-private-subnet	subnet-049f9967efe55af1b	Available	vpc-09e43c478719d2b31   cla...	10.7.8.0/24	-
clarus-az1b-public-subnet	subnet-0e9647e10645de75c	Available	vpc-09e43c478719d2b31   cla...	10.7.4.0/24	-

Subnetler olustu

- Go to the AWS console and check the VPC and subnets.
  - Go to the `prod` folder and run the command below.
- ```
bash
terraform init
terraform apply
```
- Go to the AWS console and check the VPC and subnets.

```

> .ssh
> .terraform.d
> .vscode-server
> s3-backend
> terraform-aws
> terraform-modules
  > dev
    > .terraform
    > .terraform.lock.hcl
    > dev-vpc.tf
    > terraform.tfstate
  > modules
    > main.tf
    > outputs.tf
    > variables.tf
  > prod
    > .terraform
    > .terraform.lock.hcl
    > prod-vpc.tf
    > terraform.tfstate
    > .bash_history
    > OUTLINE

```

```

}

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

Changes to Outputs:
  + vpc-cidr-block = "10.0.0.0/16"

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.tf-vpc.aws_vpc.module_vpc: Creating...
module.tf-vpc.aws_vpc.module_vpc: Creation complete after 1s [id=vpc-0753abc1add177106]
module.tf-vpc.aws_subnet.private_subnet: Creating...
module.tf-vpc.aws_subnet.public_subnet: Creation complete after 1s [id=subnet-0a8d4be6f723609d2]
module.tf-vpc.aws_subnet.private_subnet: Creation complete after 1s [id=subnet-07ac69fe9559dc16c]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

vpc-cidr-block = "10.0.0.0/16"
[ec2-user@ip-172-31-88-43 prod]$

```

Init ve apply yaptik
Simdi konmsolda prod icin de subnet ve vpc olustugunu gorecegiz

| Subnets (16) Info | | | | | | | |
|--------------------------|-------------------------------|--------------------------|-----------|---------------------------------|-------------|-----------|--|
| Filter subnets | | | | | | | |
| | Name | Subnet ID | State | VPC | IPv4 CIDR | IPv6 CIDR | |
| <input type="checkbox"/> | terraform-public-subnet-PROD | subnet-0a8d4be6f723609d2 | Available | vpc-0753abc1add177106 terr... | 10.0.1.0/24 | - | |
| <input type="checkbox"/> | terraform-public-subnet-DEV | subnet-08a5572a28a84c9c1 | Available | vpc-030e1fee59c7eabe6 terr... | 10.0.1.0/24 | - | |
| <input type="checkbox"/> | terraform-private-subnet-PROD | subnet-07ac69fe9559dc16c | Available | vpc-0753abc1add177106 terr... | 10.0.2.0/24 | - | |
| <input type="checkbox"/> | terraform-private-subnet-DEV | subnet-03c7bed8d45f01845 | Available | vpc-030e1fee59c7eabe6 terr... | 10.0.2.0/24 | - | |
| <input type="checkbox"/> | clarus-az1c-public-subnet | subnet-0a1c0b0a47e8424b3 | Available | vpc-09e43c478719d2b31 cla... | 10.7.7.0/24 | - | |

#

| Your VPCs (5) Info | | | | | | | |
|--------------------------|--------------------|-----------------------|-----------|---------------|----------------------------------|-----------|--------------|
| Filter VPCs | | | | | | | |
| | Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR (Network border group) | IPv6 pool | DHCP options |
| <input type="checkbox"/> | aws_capstone-VPC | vpc-0515554f6b9fee688 | Available | 90.90.0.0/16 | - | - | dop |
| <input type="checkbox"/> | default-vpc | vpc-d9e37fa4 | Available | 172.31.0.0/16 | - | - | dop |
| <input type="checkbox"/> | terraform-vpc-PROD | vpc-0753abc1add177106 | Available | 10.0.0.0/16 | - | - | dop |
| <input type="checkbox"/> | clarus-vpc-a | vpc-09e43c478719d2b31 | Available | 10.7.0.0/16 | - | - | dop |
| <input type="checkbox"/> | terraform-vpc-DEV | vpc-030e1fee59c7eabe6 | Available | 10.0.0.0/16 | - | - | dop |

```

TERMINAL
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

private_subnet_cidr = "10.0.2.0/24"
public_subnet_cidr = "10.0.1.0/24"
vpc_cidr = "10.0.0.0/16"
vpc_id = "vpc-04b6b719e9be7e1ec"
[ec2-user@ip-172-31-88-43 modules]$

```

Modules içinde terraform init ve apply yapınca bu sefer default olarak serkan ismiyle vpc olussun ve subnetlerolussun demistik onlar olusuyor

| Your VPCs (5) Info | | | | | | | |
|--------------------------|----------------------|-----------------------|-----------|---------------|----------------------------------|-----------|---------------|
| Filter VPCs | | | | | | | |
| | Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR (Network border group) | IPv6 pool | DHCP options |
| <input type="checkbox"/> | terraform-vpc-serkan | vpc-04b6b719e9be7e1ec | Available | 10.0.0.0/16 | - | - | dopt-2c340d56 |
| <input type="checkbox"/> | aws_capstone-VPC | vpc-0515554f6b9fee688 | Available | 90.90.0.0/16 | - | - | dopt-2c340d56 |
| <input type="checkbox"/> | default-vpc | vpc-d9e37fa4 | Available | 172.31.0.0/16 | - | - | dopt-2c340d56 |
| <input type="checkbox"/> | clarus-vpc-a | vpc-09e43c478719d2b31 | Available | 10.7.0.0/16 | - | - | dopt-2c340d56 |

| Subnets (16) Info | | | | | | | |
|--------------------------|---------------------------------|--------------------------|-----------|---------------------------------|-------------|-----------|--|
| Filter subnets | | | | | | | |
| | Name | Subnet ID | State | VPC | IPv4 CIDR | IPv6 CIDR | |
| <input type="checkbox"/> | terraform-public-subnet-serkan | subnet-0004d85a99d05cf29 | Available | vpc-04b6b719e9be7e1ec terr... | 10.0.1.0/24 | - | |
| <input type="checkbox"/> | terraform-public-subnet-DEV | subnet-08a5572a28a84c9c1 | Available | vpc-030e1fee59c7eabe6 terr... | 10.0.1.0/24 | - | |
| <input type="checkbox"/> | terraform-private-subnet-serkan | subnet-0194628406618b610 | Available | vpc-04b6b719e9be7e1ec terr... | 10.0.2.0/24 | - | |
| <input type="checkbox"/> | terraform-private-subnet-DEV | subnet-03c7bed8d45f01845 | Available | vpc-030e1fee59c7eabe6 terr... | 10.0.2.0/24 | - | |
| <input type="checkbox"/> | clarus-az1c-public-subnet | subnet-0a1c0b0a47e8424b3 | Available | vpc-09e43c478719d2b31 cla... | 10.7.7.0/24 | - | |
| <input type="checkbox"/> | clarus-az1c-private-subnet | subnet-049f9967efe55af1b | Available | vpc-09e43c478719d2b31 cla... | 10.7.8.0/24 | - | |
| <input type="checkbox"/> | clarus-az1b-public-subnet | subnet-0e9647e10645de75c | Available | vpc-09e43c478719d2b31 cla... | 10.7.4.0/24 | - | |
| <input type="checkbox"/> | clarus-az1b-private-subnet | subnet-03d721df393d322fd | Available | vpc-09e43c478719d2b31 cla... | 10.7.5.0/24 | - | |
| <input type="checkbox"/> | clarus-az1a-public-subnet | subnet-06771acce792a0b0 | Available | vpc-09e43c478719d2b31 cla... | 10.7.1.0/24 | - | |

En fazla 5 tane vpc acabiliyorsunuz

Simdi ekstradan bir sec grup da resource olarak main.tf de olusturuyoruz

```
resource "aws_security_group" "vpc-tf-sec-gr" {
  name = "vpc-tf-sec-gr"
  vpc_id = aws_vpc.module_vpc.id
  ingress {
    from_port = 22
    protocol = "tcp"
    to_port = 22
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port = 0
    protocol = "-1"
    to_port = 0
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ>>

STATE TERRAFORMUN EN ONEMLI UNSURU,
BIR DEGISIKLIK VEYA BIR SEY YAPILMASIICIN STATE DE OLMASI LAZIM

Destroy

The `terraform destroy` command terminates resources defined in your Terraform configuration. This command is the reverse of `terraform apply` in that it terminates all the resources specified by the configuration. It does not destroy resources running elsewhere that are not described in the current configuration.

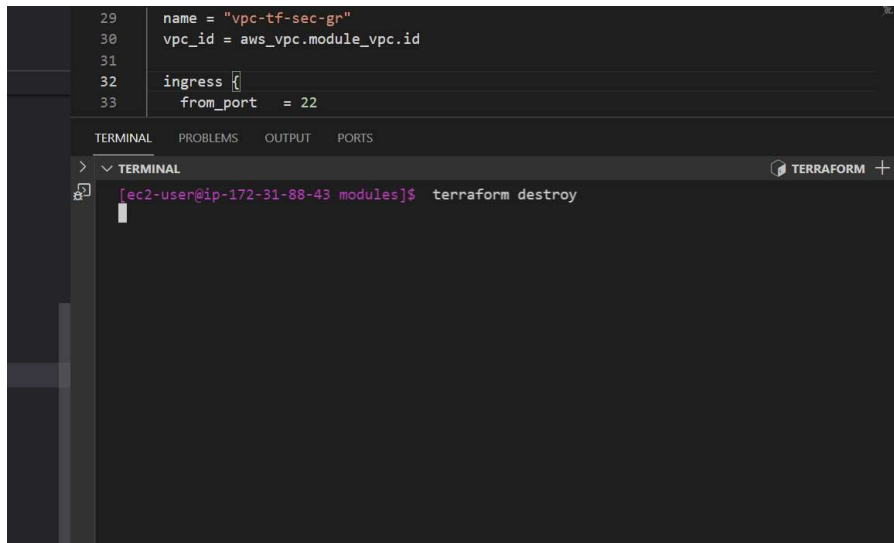
- Go to the `prod` and `dev` folders and run the command below.

```
```bash
```

```
terraform destroy -auto-approve
```

```
```
```

- Visit the EC2, S3, DynamoDB, IAM, VPC console to see all the resources are deleted if not delete manually. Because all tf configuration files are related each other so sometimes terraform destroy will not work. Don't forget to deregister ami and delete the snapshot (when you creating an ami, AWS create a snapshot of your instance) of your instance. Finally delete your terraform EC2 instance.



```
> TERMINAL
Changes to Outputs:
- private_subnet_cidr = "10.0.2.0/24" -> null
- public_subnet_cidr = "10.0.1.0/24" -> null
- vpc_cidr = "10.0.0.0/16" -> null
- vpc_id = "vpc-04b6b719e9be7e1ec" -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_subnet.private_subnet: Destroying... [id=subnet-0194628406618b610]
aws_security_group.vpc-tf-sec-gr: Destroying... [id=sg-08afa81d6bb830211]
aws_subnet.public_subnet: Destroying... [id=subnet-0004d85a99d05cf29]
aws_security_group.vpc-tf-sec-gr: Destruction complete after 0s
aws_subnet.public_subnet: Destruction complete after 0s
aws_subnet.private_subnet: Destruction complete after 1s
aws_vpc.module_vpc: Destroying... [id=vpc-04b6b719e9be7e1ec]
aws_vpc.module_vpc: Destruction complete after 0s

Destroy complete! Resources: 4 destroyed.
[ec2-user@ip-172-31-88-43 modules]$
```

Ayrı ayrı destroy yapmaliyiz

```
TERMINAL PROBLEMS OUTPUT PORTS
> TERMINAL
[ec2-user@ip-172-31-88-43 dev]$ terraform apply
provider.aws.region
The region where AWS operations will take place. Examples
are us-east-1, us-west-2, etc.

Enter a value: us-east-1

Error: Unsupported attribute

on dev-vpc.tf line 14, in resource "aws_instance" "tf-module-ec2":
14:     vpc_security_group_ids = [module.tf-vpc.aws_security_group.vpc-tf-sec-gr.id]
    module.tf-vpc is a object, known only after apply

This object does not have an attribute named "aws_security_group".

[ec2-user@ip-172-31-88-43 dev]$ terraform destroy
provider.aws.region
The region where AWS operations will take place. Examples
are us-east-1, us-west-2, etc.

Enter a value: 
```

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

Changes to Outputs:
- vpc-cidr-block = "10.0.0.0/16" -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

module.tf-vpc.aws_subnet.public_subnet: Destroying... [id=subnet-08a5572a28a84c9c1]
module.tf-vpc.aws_subnet.private_subnet: Destroying... [id=subnet-03c7bed8d45f01845]
module.tf-vpc.aws_subnet.public_subnet: Destruction complete after 1s
module.tf-vpc.aws_subnet.private_subnet: Destruction complete after 1s
module.tf-vpc.aws_vpc.module_vpc: Destroying... [id=vpc-030e1fee59c7eabe6]
module.tf-vpc.aws_vpc.module_vpc: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
[ec2-user@ip-172-31-88-43 dev]$
```

Vpc ler destroy olmus

Filter VPCs

<

1

>

| <input type="checkbox"/> | Name | VPC ID | State | IPv4 CIDR | IPv6 CIDR (Network border group) | IPv6 pool | DHCP options set |
|--------------------------|------------------|-----------------------|-----------|---------------|----------------------------------|-----------|------------------|
| <input type="checkbox"/> | aws_capstone-VPC | vpc-0515554f6b9fee688 | Available | 90.90.0.0/16 | - | - | dopt-2c340d56 |
| <input type="checkbox"/> | default-vpc | vpc-d9e37fa4 | Available | 172.31.0.0/16 | - | - | dopt-2c340d56 |
| <input type="checkbox"/> | clarus-vpc-a | vpc-09e43c478719d2b31 | Available | 10.7.0.0/16 | - | - | dopt-2c340d56 |

SUBNETLER de destroy olmus

Subnets (12) Info

Filter subnets

Actions

Create subnet

<

1

>

| <input type="checkbox"/> | Name | Subnet ID | State | VPC | IPv4 CIDR | IPv6 CIDR | Available IPv4 addresses | Avail |
|--------------------------|------------------------|--------------------------|-----------|--------------------------------|----------------|-----------|--------------------------|-------|
| <input type="checkbox"/> | clarus-az1c-public... | subnet-0a1c0b0a47e8424b3 | Available | vpc-09e43c478719d2b31 cla... | 10.7.7.0/24 | - | 251 | us-ea |
| <input type="checkbox"/> | clarus-az1c-private... | subnet-049f9967efe55af1b | Available | vpc-09e43c478719d2b31 cla... | 10.7.8.0/24 | - | 251 | us-ea |
| <input type="checkbox"/> | clarus-az1b-public... | subnet-0e9647e10645de75c | Available | vpc-09e43c478719d2b31 cla... | 10.7.4.0/24 | - | 251 | us-ea |
| <input type="checkbox"/> | clarus-az1b-private... | subnet-03d721df393d322fd | Available | vpc-09e43c478719d2b31 cla... | 10.7.5.0/24 | - | 251 | us-ea |
| <input type="checkbox"/> | clarus-az1a-public... | subnet-06771acceb792a0b0 | Available | vpc-09e43c478719d2b31 cla... | 10.7.1.0/24 | - | 251 | us-ea |
| <input type="checkbox"/> | clarus-az1a-private... | subnet-007f968dea82c83ef | Available | vpc-09e43c478719d2b31 cla... | 10.7.2.0/24 | - | 251 | us-ea |
| <input type="checkbox"/> | - | subnet-37d4ee39 | Available | vpc-d9e37fa4 default-vpc | 172.31.64.0/20 | - | 4091 | us-ea |
| <input type="checkbox"/> | - | subnet-28db4c19 | Available | vpc-d9e37fa4 default-vpc | 172.31.48.0/20 | - | 4091 | us-ea |
| <input type="checkbox"/> | - | subnet-573e7076 | Available | vpc-d9e37fa4 default-vpc | 172.31.80.0/20 | - | 4089 | us-ea |