Handson2

Thursday, 2 September 2021 23.19

# Hands-on Terraform-02 : Terraform Variables, Conditionals, Loops, Data Sources.

Purpose of the this hands-on training is to give students the knowledge of variables, conditionals, loops and data sources in Terraform.

## Learning Outcomes

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

- Use variables, conditionals, loops and data sources with Terraform

### Variables

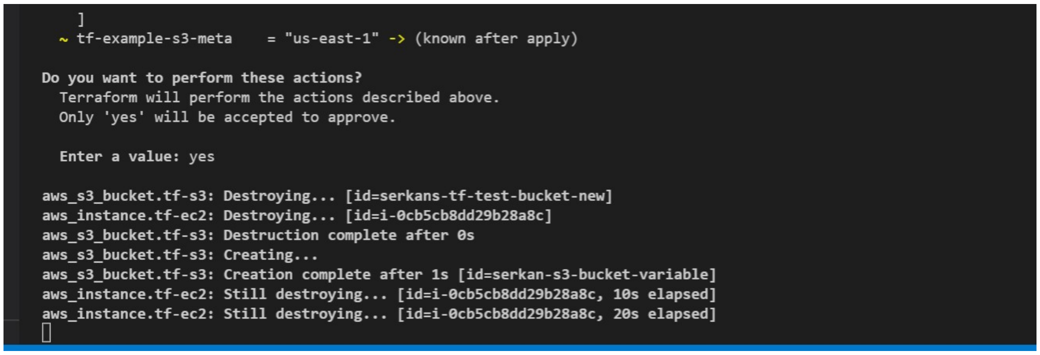
- Make the changes in the `main.tf` file.

```bash
provider "aws" {
  region  = "us-east-1"
}
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "3.38.0"
    }
  }
}
variable "ec2-name" {
  default = "oliver-ec2"
}
variable "ec2-type" {
  default = "t2.micro"
}
variable "ec2-ami" {
  default = "ami-0742b4e673072066f"
}
resource "aws_instance" "tf-ec2" {
  ami           = var.ec2-ami
  instance_type = var.ec2-type
  key_name      = "mk"
  tags = {
    Name = "${var.ec2-name}-💻🎯🎉"
  }
}
variable "s3-bucket-name" {
  default = "oliver-s3-bucket-variable-addwhateveryouwant"
}
resource "aws_s3_bucket" "tf-s3" {
  bucket = var.s3-bucket-name
  acl    = "private"
}
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" {
  value = aws_s3_bucket.tf-s3[*]
}
```

MAIN.tF MIZI SOYLE DUZENLEDIK

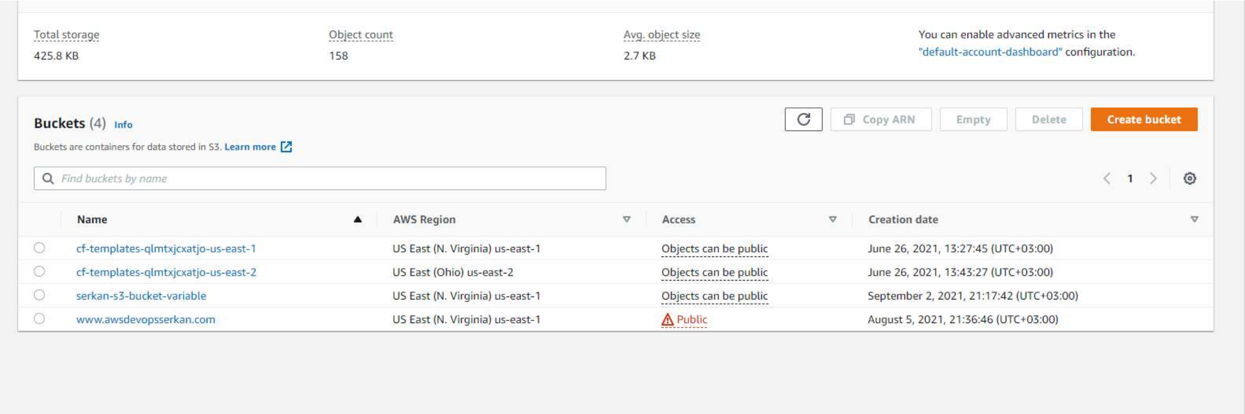
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "3.56.0"
    }
  }
}
provider "aws" {
  region  = "us-east-1"
  # Configuration options
}
variable "ec2-name" {
  default = "serkan-ec2"
}
variable "ec2-instance-type" {
  default = "t2.micro"

}
variable "ec2-ami" {
  default = "ami-0c2b8ca1dad447f8a"

}
resource "aws_instance" "tf-ec2" {     #resource blogu
    ami = var.ec2-ami
    instance_type = var.ec2-instance-type
    key_name = "ec2_key"

    tags = {
      "Name" = "${var.ec2-name}-:)"
    }

}
variable "s3-bucket-name" {
  default = "serkan-s3-bucket-variable"

}
resource "aws_s3_bucket" "tf-s3" {
  bucket = var.s3-bucket-name
  acl = "private"
}
  output "tf-example-public-
ip" {  #bu ismi biz veriyoruz public ip yi isteyecegiz
    value = aws_instance.tf-ec2.public_ip
  }
  output "tf-example-s3-meta" { #s3 e ait metadatayi isteyeciz
    value = aws_s3_bucket.tf-s3.region
  }
  output "tf-example-private-ip" {
    value = aws_instance.tf-ec2.private_ip
  }
  output "tf-example-s3" {
    value = aws_s3_bucket.tf-s3[*]
  }
```

YENI MAKINE KURUYOR



```
      ]
    ~ tf-example-s3-meta    = "us-east-1" -> (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

  aws_s3_bucket.tf-s3: Destroying... [id=serkans-tf-test-bucket-new]
  aws_instance.tf-ec2: Destroying... [id=i-0cb5cb8dd29b28a8c]
  aws_s3_bucket.tf-s3: Destruction complete after 0s
  aws_s3_bucket.tf-s3: Creating...
  aws_s3_bucket.tf-s3: Creation complete after 1s [id=serkan-s3-bucket-variable]
  aws_instance.tf-ec2: Still destroying... [id=i-0cb5cb8dd29b28a8c, 10s elapsed]
  aws_instance.tf-ec2: Still destroying... [id=i-0cb5cb8dd29b28a8c, 20s elapsed]
```

Bucket kuruyor

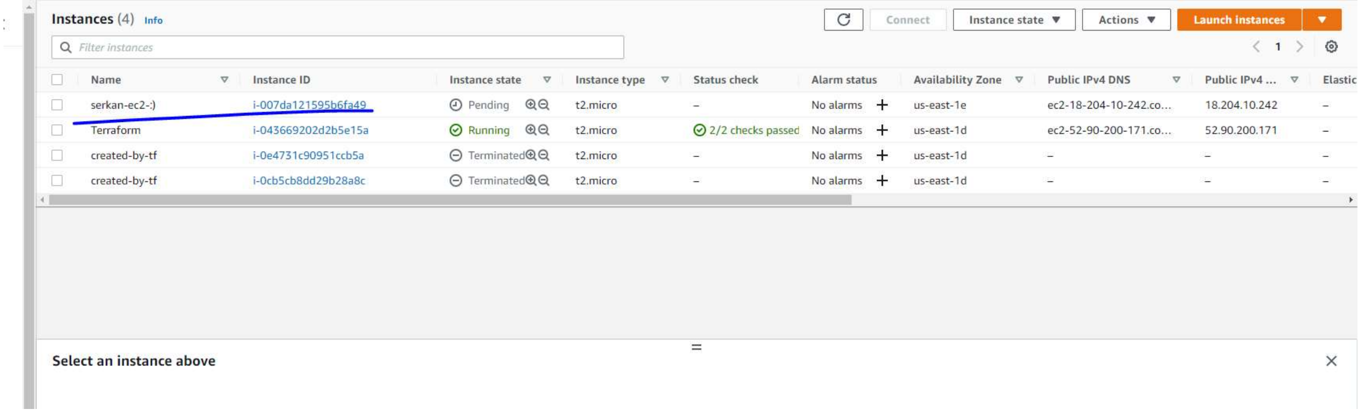


Total storage	Object count	Avg. object size	
425.8 KB	158	2.7 KB	You can enable advanced metrics in the "default-account-dashboard" configuration.

Buckets (4) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	cf-templates-qlmtxjcxatjo-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-qlmtxjcxatjo-us-east-2	US East (Ohio) us-east-2	Objects can be public	June 26, 2021, 13:43:27 (UTC+03:00)
○	serkan-s3-bucket-variable	US East (N. Virginia) us-east-1	Objects can be public	September 2, 2021, 21:17:42 (UTC+03:00)
○	www.awsdevopsserkan.com	US East (N. Virginia) us-east-1	⚠ Public	August 5, 2021, 21:36:46 (UTC+03:00)



Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
□	serkan-ec2-:)	i-007da121595b6fa49	Pending	t2.micro	–	No alarms	us-east-1e	ec2-18-204-10-242.co...	18.204.10.242	–
□	Terraform	i-043669202d2b5e15a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-52-90-200-171.co...	52.90.200.171	–
□	created-by-tf	i-0e4731c90951ccb5a	Terminated	t2.micro	–	No alarms	us-east-1d	–	–	–
□	created-by-tf	i-0cb5cb8dd29b28a8c	Terminated	t2.micro	–	No alarms	us-east-1d	–	–	–

Select an instance above

```bash
terraform apply
```

Sadece onceki sayfada olusturdugumuz variable kisimlarini kesip yeni olusturavagimiz variables.tf dosyasina ekliyoruz
Burda olmasi da sonucu degistirmyecek
State tf uzantili dosyalara bakacak

- Create a file name `variables.tf`. Take the variables from `main.tf` file and paste into "variables.tf".

```bash
terraform validate
terraform fmt
terraform apply
```

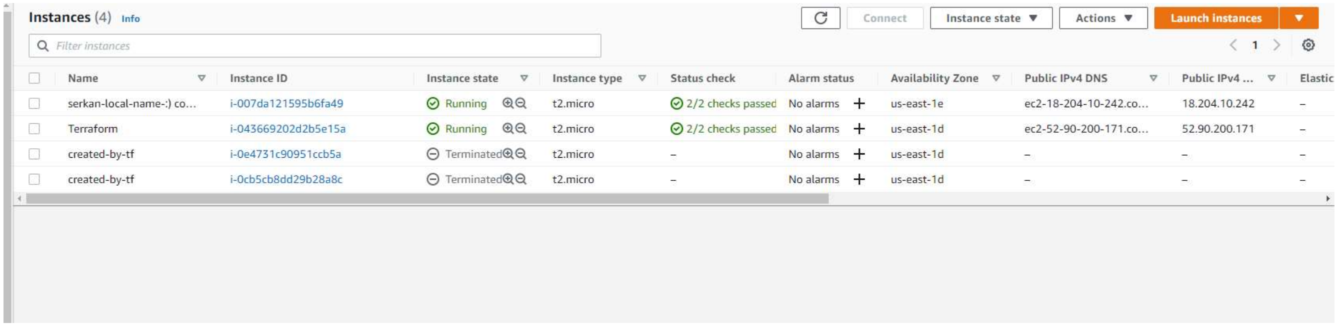
- Comment the variable of `ec2-name` with "ctrl+k+c" in vscode. (comment out = ctrl+k+u) Then make the changes in the `main.tf` file.

```bash
locals {
  instance-name = "oliver-local-name"
}
resource "aws_instance" "tf-ec2" {
  ami           = var.ec2-ami
  instance_type = var.ec2-type
  key_name      = "mk"
  tags = {
    Name = "${local.instance-name}-come from locals"
  }
}
```

- A `local` value assigns a name to an expression, so you can use it multiple times within a module without repeating it.
- Run the command `terraform plan`

```bash
terraform plan
```

~~~
```
- Run the command `terraform apply` again. Check the EC2 instance's Name tag column.
```bash
terraform apply
```
- Go to the `variables.tf` file and comment the s3 bucket name variable's default value.
```tf
variable "s3-bucket-name" {
#   default     = "oliver-new-s3-bucket-addwhateveryouwant"
}
```
~~~

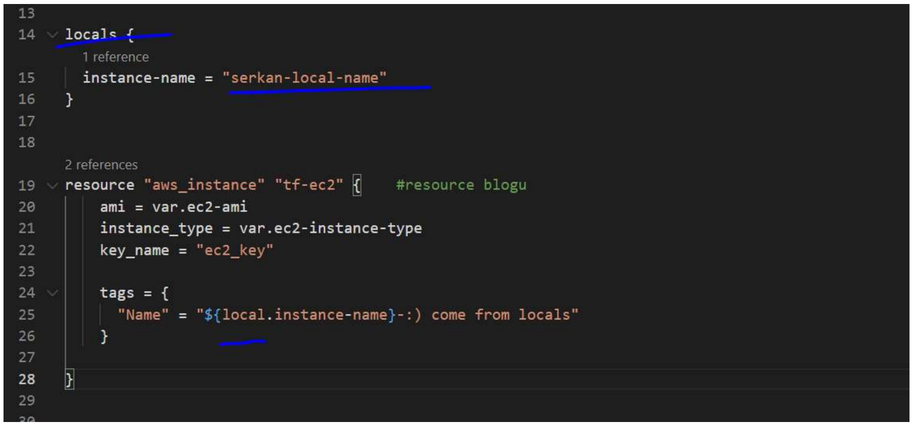


Instances (4) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
☐	serkan-local-name-:) co...	i-007da121595b6fa49	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1e	ec2-18-204-10-242.co...	18.204.10.242	–
☐	Terraform	i-043669202d2b5e15a	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1d	ec2-52-90-200-171.co...	52.90.200.171	–
☐	created-by-tf	i-0e4731c90951ccb5a	⊖ Terminated	t2.micro	–	No alarms +	us-east-1d	–	–	–
☐	created-by-tf	i-0cb5cb8dd29b28a8c	⊖ Terminated	t2.micro	–	No alarms +	us-east-1d	–	–	–

Isim degistirdi m,akineyi destroy etmedi

Local ile de variable a benzer sekilde atamalar yapabiliyoruz



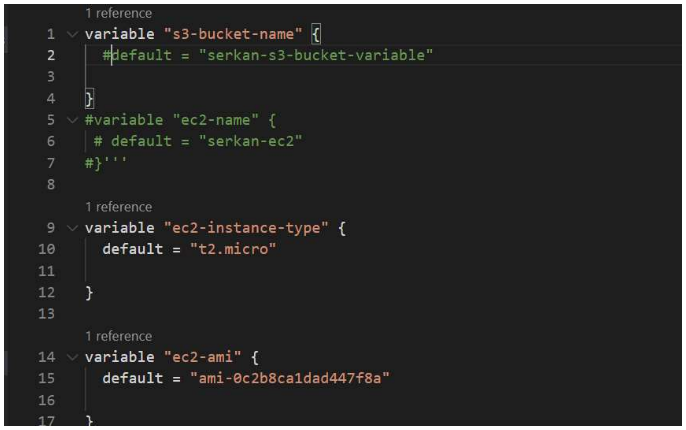
```
locals {
  1 reference
  instance-name = "serkan-local-name"
}


2 references
resource "aws_instance" "tf-ec2" {     #resource blogu
    ami = var.ec2-ami
    instance_type = var.ec2-instance-type
    key_name = "ec2_key"

    tags = {
      "Name" = "${local.instance-name}-:) come from locals"
    }

}

```

Variable daha genel herkes icin hazirlanmis
Locals da daha ozel daha fazla kullanilabilecek bir durumda kullanilmak icin yapilmis.

ctrl+k+c

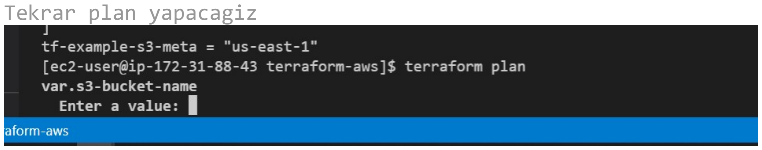
From <https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ>

toplu olarak yorum satiri



```
1 reference
variable "s3-bucket-name" {
  #default = "serkan-s3-bucket-variable"

}
#variable "ec2-name" {
 # default = "serkan-ec2"
#}'''

1 reference
variable "ec2-instance-type" {
  default = "t2.micro"

}

1 reference
variable "ec2-ami" {
  default = "ami-0c2b8ca1dad447f8a"

}
```

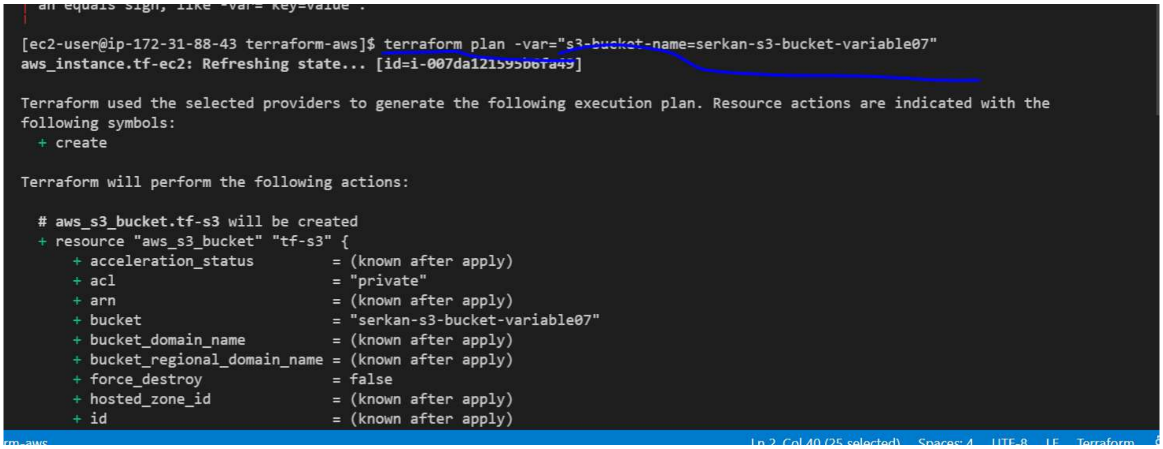
Bucket name # satri yaptik
Tekrar plan yapacagiz



```
tf-example-s3-meta = "us-east-1"
[ec2-user@ip-172-31-88-43 terraform-aws]$ terraform plan
var.s3-bucket-name
  Enter a value: █
```

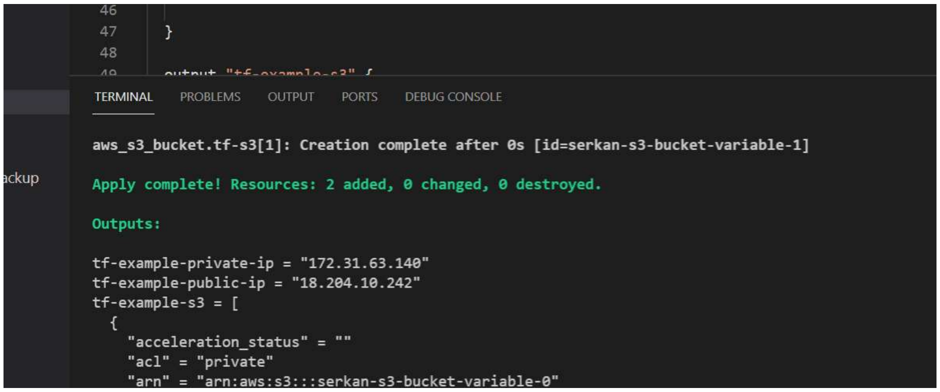
Variable da defaultu gormedigindsen ismi bana soruyor

```bash
terraform plan
```
- You can define variables with `-var` command
```bash
terraform plan -var="s3-bucket-name=oliver-new-s3-bucket-2"
```
- Create a file name `oliver.tfvars`. Add the followings.
```bash
s3-bucket-name = "oliver-s3-bucket-newest"
```
- Run the command belov.



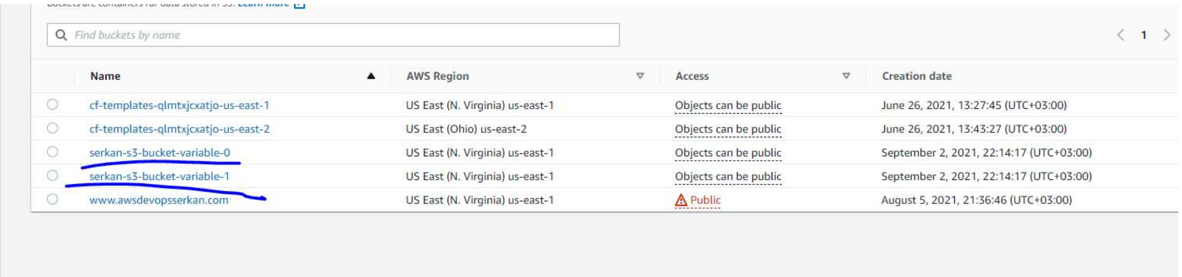
```
  an equals sign, like -var= key=value .

[ec2-user@ip-172-31-88-43 terraform-aws]$ terraform plan -var="s3-bucket-name=serkan-s3-bucket-variable07"
aws_instance.tf-ec2: Refreshing state... [id=i-007da121595b6fa49]

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_s3_bucket.tf-s3 will be created
  + resource "aws_s3_bucket" "tf-s3" {
      + acceleration_status         = (known after apply)
      + acl                         = "private"
      + arn                         = (known after apply)
      + bucket                      = "serkan-s3-bucket-variable07"
      + bucket_domain_name          = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
```

Variable olmadigi icin boyle yap diyoruz



```
}

output "tf-example-s3" {
```

```
aws_s3_bucket.tf-s3[1]: Creation complete after 0s [id=serkan-s3-bucket-variable-1]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

tf-example-private-ip = "172.31.63.140"
tf-example-public-ip = "18.204.10.242"
tf-example-s3 = [
  {
    "acceleration_status" = ""
    "acl" = "private"
    "arn" = "arn:aws:s3:::serkan-s3-bucket-variable-0"
```



	Name	AWS Region	Access	Creation date
○	cf-templates-qlmtxjcxatjo-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-qlmtxjcxatjo-us-east-2	US East (Ohio) us-east-2	Objects can be public	June 26, 2021, 13:43:27 (UTC+03:00)
○	serkan-s3-bucket-variable-0	US East (N. Virginia) us-east-1	Objects can be public	September 2, 2021, 22:14:17 (UTC+03:00)
○	serkan-s3-bucket-variable-1	US East (N. Virginia) us-east-1	Objects can be public	September 2, 2021, 22:14:17 (UTC+03:00)
○	www.awsdevopsserkan.com	US East (N. Virginia) us-east-1	⚠ Public	August 5, 2021, 21:36:46 (UTC+03:00)

```bash
terraform plan --var-file="oliver.tfvars"
```

- Go to the `variables.tf` file and comment out the s3 bucket name variable's default value..

```tf
variable "s3-bucket-name" {
  default     = "oliver-new-s3-bucket"
}
```

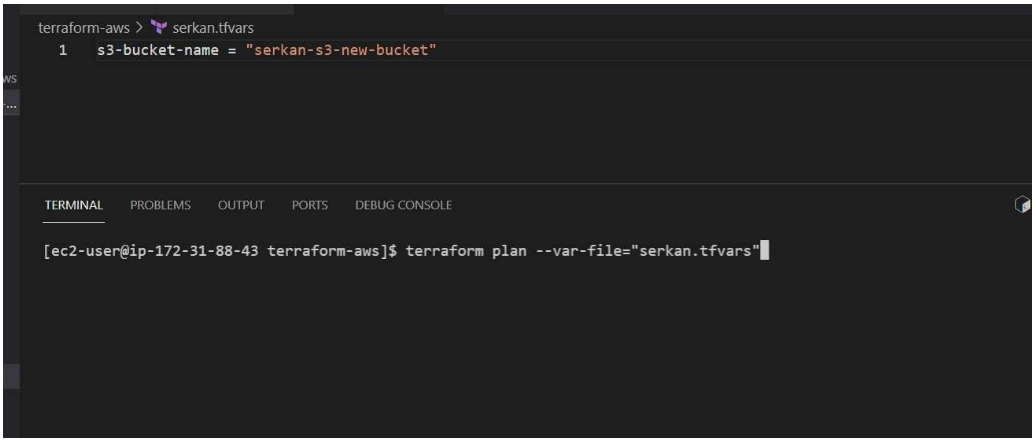
- Run terraform apply --var-file="oliver.tfvars" command.

```bash
terraform apply --var-file="oliver.tfvars"

Variablelari tanimlama yeri



```

- Run terraform apply command.

```bash
terraform apply
```

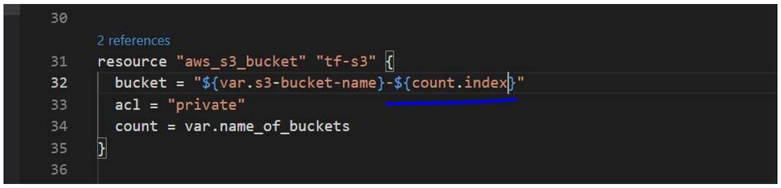
- Change the name of oliver.tfvars to terraform.tfvars.
- Run terraform apply command.



```
s3-bucket-name = "serkan-s3-new-bucket"
```

```
[ec2-user@ip-172-31-88-43 terraform-aws]$ terraform plan --var-file="serkan.tfvars"
```

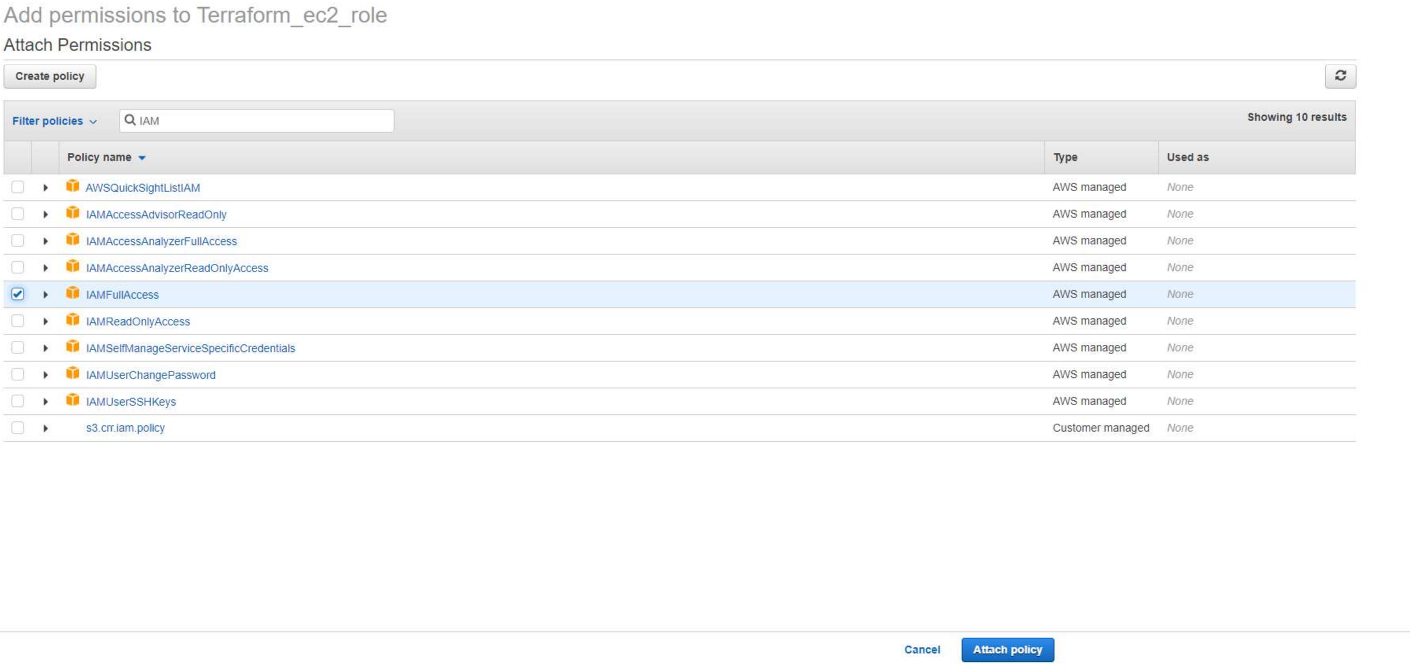
### Conditionals and Loops
- Count and count.index
- Go to the `variables.tf` file and create a new variable.
```bash
variable "num_of_buckets" {
  default = 2
}
```
- Go to the `main.tf` file, make the changes in order.
```bash
resource "aws_s3_bucket" "tf-s3" {
  bucket = "${var.s3-bucket-name}-${count.index}"
  acl    = "private"
  count = var.num_of_buckets
}
```
```bash
terraform plan
```
```bash
terraform apply
```
- Check the S3 buckets from console.
- Conditional Expressions.
- Go to the `main.tf` file, make the changes in order.
```bash
resource "aws_s3_bucket" "tf-s3" {
  bucket = "${var.s3-bucket-name}-${count.index}"
  acl    = "private"
  # count = var.num_of_buckets
  count = var.num_of_buckets != 0 ? var.num_of_buckets : 3
}
```



```
resource "aws_s3_bucket" "tf-s3" {
  bucket = "${var.s3-bucket-name}-${count.index}"
  acl = "private"
  count = var.name_of_buckets
}
```

Bucket uniq oldugundan isim burda sira nosu ver diyoruz

`



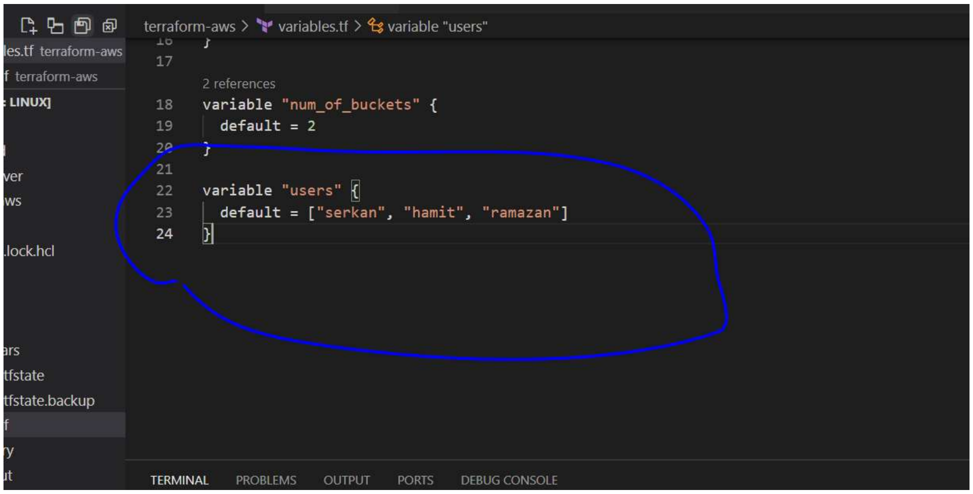
Add permissions to Terraform_ec2_role

Attach Permissions

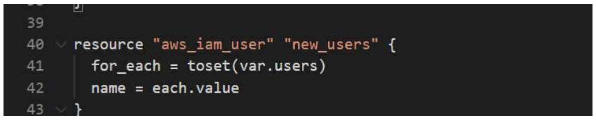
🔍 IAM

Showing 10 results

		Policy name ▾	Type	Used as
□	▸	AWSQuickSightListIAM	AWS managed	None
□	▸	IAMAccessAdvisorReadOnly	AWS managed	None
□	▸	IAMAccessAnalyzerFullAccess	AWS managed	None
□	▸	IAMAccessAnalyzerReadOnlyAccess	AWS managed	None
☑	▸	IAMFullAccess	AWS managed	None
□	▸	IAMReadOnlyAccess	AWS managed	None
□	▸	IAMSelfManageServiceSpecificCredentials	AWS managed	None
□	▸	IAMUserChangePassword	AWS managed	None
□	▸	IAMUserSSHKeys	AWS managed	None
□	▸	s3.crr.iam.policy	Customer managed	None



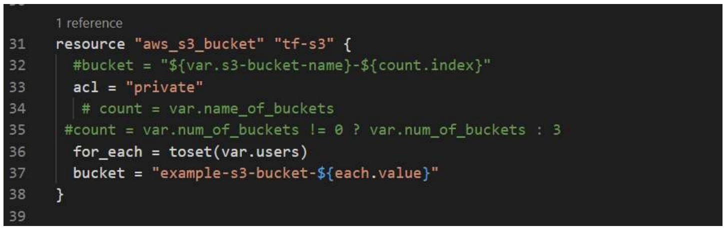
```
}

2 references
variable "num_of_buckets" {
  default = 2
}

variable "users" {
  default = ["serkan", "hamit", "ramazan"]
}
```



```
}

resource "aws_iam_user" "new_users" {
  for_each = toset(var.users)
  name = each.value
}
```

Uc tane user var

Hepsine de bucket olustur dedik



```
1 reference
resource "aws_s3_bucket" "tf-s3" {
  #bucket = "${var.s3-bucket-name}-${count.index}"
  acl = "private"
   # count = var.name_of_buckets
 #count = var.num_of_buckets != 0 ? var.num_of_buckets : 3
  for_each = toset(var.users)
  bucket = "example-s3-bucket-${each.value}"
}
```

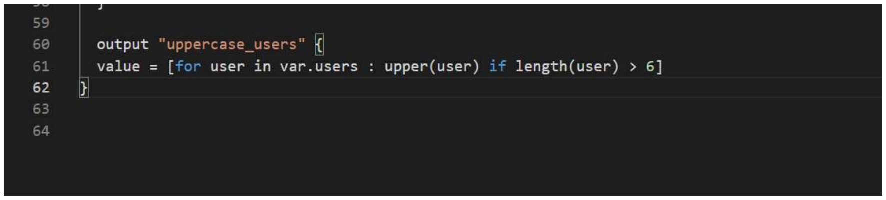
```bash
terraform plan
```

- Functions.
- Go to the `variables.tf` file again and add a new variable.

```bash
variable "users" {
  default = ["spring", "micheal", "oliver"]
}
```

- Go to the `main.tf` file make the changes. Change the IAM role and add IAMFullAccess policy.

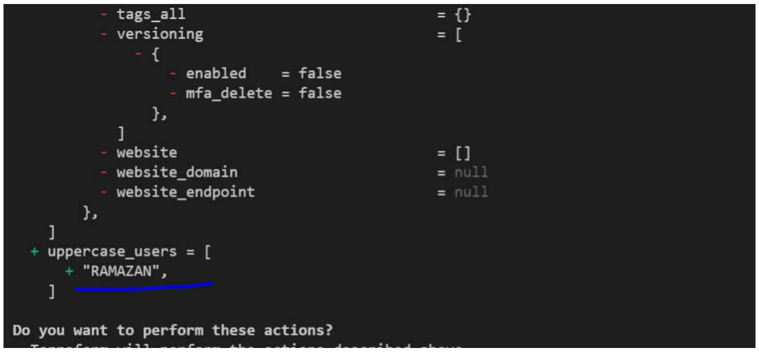
```bash
resource "aws_s3_bucket" "tf-s3" {
  # bucket = "var.s3-bucket-name.${count.index}"
  acl = "private"
  # count = var.num_of_buckets
  # count = var.num_of_buckets != 0 ? var.num_of_buckets : 1
  for_each = toset(var.users)
  bucket   = "example-s3-bucket-${each.value}"
}
resource "aws_iam_user" "new_users" {
  for_each = toset(var.users)
  name = each.value
}
output "uppercase_users" {
  value = [for user in var.users : upper(user) if length(user) > 6]
}
```



```
output "uppercase_users" {
value = [for user in var.users : upper(user) if length(user) > 6]
}
```

Burda da bir dongu var
Userlari var.users da dondur lenghti 6 dan buyukse upper yap

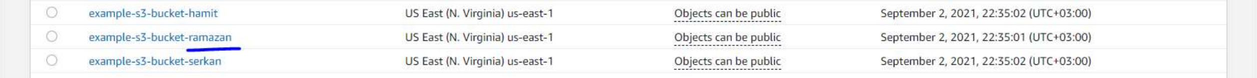
Uc isim vermsitik 6 dan buyuk olanin ismini dondurdu



```
          - tags_all                                = {}
          - versioning                              = [
              - {
                  - enabled    = false
                  - mfa_delete = false
                },
            ]
          - website                                 = []
          - website_domain                          = null
          - website_endpoint                        = null
        },
    ]
  + uppercase_users = [
      + "RAMAZAN",
    ]

Do you want to perform these actions?
```

```bash
terraform apply
```

- Go to the AWS console (IAM and S3) and check the resources.



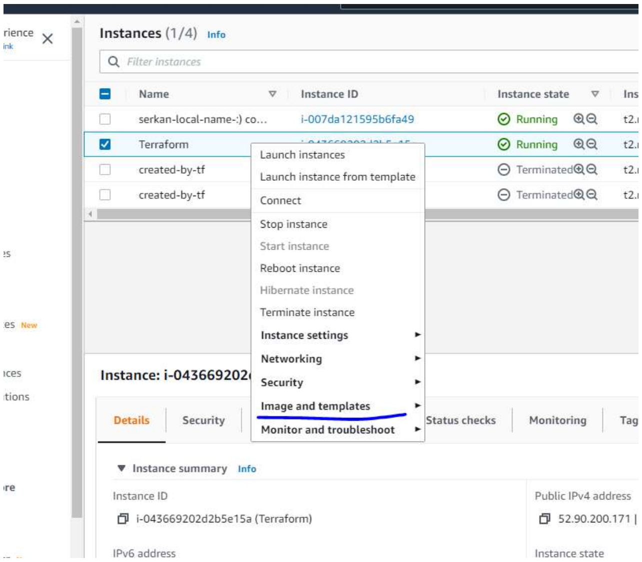
○	example-s3-bucket-hamit	US East (N. Virginia) us-east-1	Objects can be public	September 2, 2021, 22:35:02 (UTC+03:00)
○	example-s3-bucket-ramazan	US East (N. Virginia) us-east-1	Objects can be public	September 2, 2021, 22:35:01 (UTC+03:00)
○	example-s3-bucket-serkan	US East (N. Virginia) us-east-1	Objects can be public	September 2, 2021, 22:35:02 (UTC+03:00)

Kondolda da olusutugunu goruyoruz

### Terraform Data Sources

- `Data sources` allow data to be fetched or computed for use elsewhere in Terraform configuration.
- Go to the `AWS console and create an image` from your EC2. Select your instance

and from actions click image and templates and then give a name for ami `my-ami` and click create.



Instances (1/4) Info

Filter instances

☐	Name	Instance ID	Instance state	Ins
☐	serkan-local-name-:) co...	i-007da121595b6fa49	Running	t2.
☑	Terraform		Running	t2.
☐	created-by-tf		Terminated	t2.
☐	created-by-tf		Terminated	t2.

Instance: i-043669202

Details | Security | Status checks | Monitoring | Tag

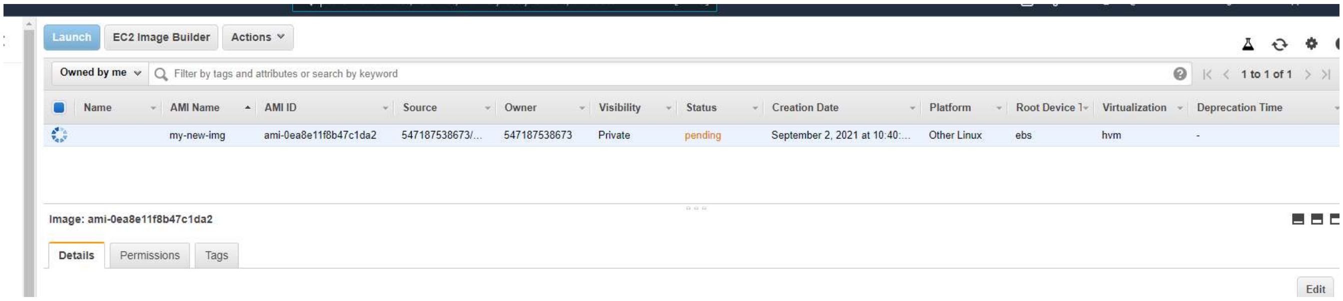
Instance summary Info

Instance ID: i-043669202d2b5e15a (Terraform)

Public IPv4 address: 52.90.200.171 |

IPv6 address

Instance state



Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform	Root Device T	Virtualization	Deprecation Time
	my-new-img	ami-0ea8e11f8b47c1da2	547187538673/...	547187538673	Private	pending	September 2, 2021 at 10:40:...	Other Linux	ebs	hvm	-

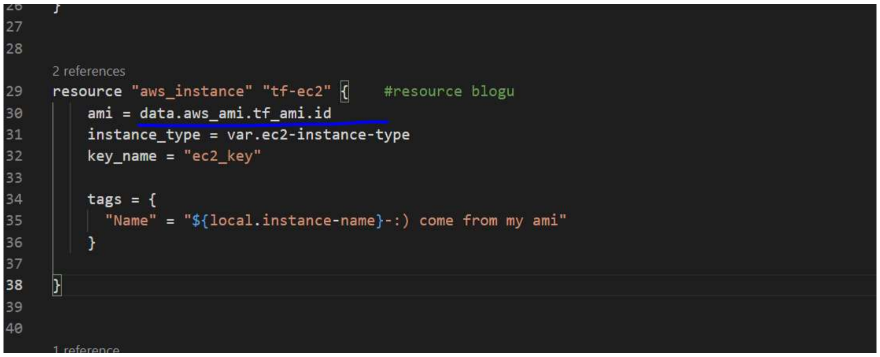
Image: ami-0ea8e11f8b47c1da2

# It will take some time. go to the next steps.

- Go to the `variables.tf` file and comment the variable `ec2-ami`.
- Go to the `main.tf` file make the changes in order.

```bash
data "aws_ami" "tf_ami" {
  most_recent      = true
  owners           = ["self"]
  filter {
    name = "virtualization-type"
    values = ["hvm"]
  }
}
resource "aws_instance" "tf-ec2" {
  ami           = data.aws_ami.tf_ami.id
  instance_type = var.ec2-type
  key_name      = "mk"
  tags = {
    Name = "${local.instance-name}-this is from my-ami"
  }
}
```

Burda datadan cejkecegi icin amiyi o sekilde degistirdik
Ayrica variables.tf de de ami kismini # satirina aldik



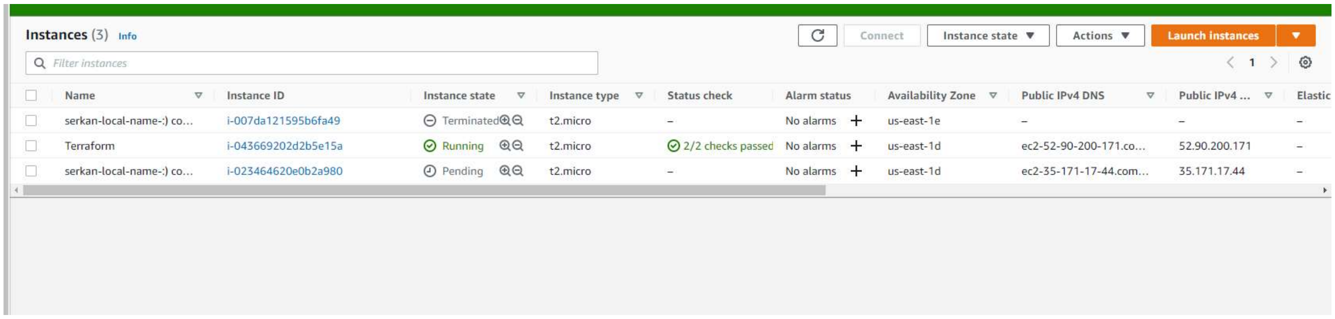
```
}


2 references
resource "aws_instance" "tf-ec2" {     #resource blogu
    ami = data.aws_ami.tf_ami.id
    instance_type = var.ec2-instance-type
    key_name = "ec2_key"

    tags = {
      "Name" = "${local.instance-name}-:) come from my ami"
    }

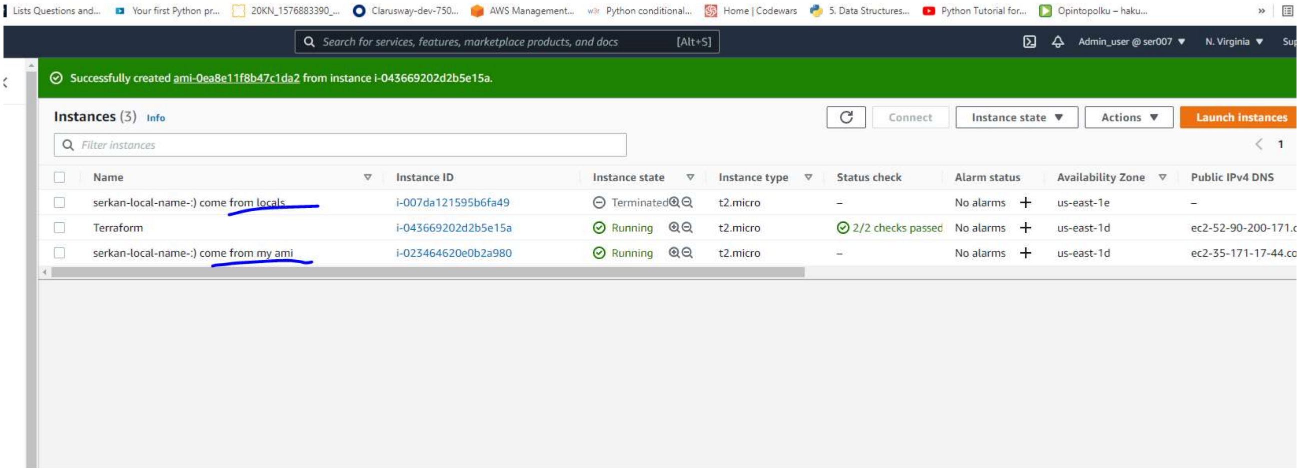
}


1 reference
```



Instances (3) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
□	serkan-local-name-:) co...	i-007da121595b6fa49	Terminated	t2.micro	–	No alarms +	us-east-1e	–	–	–
□	Terraform	i-043669202d2b5e15a	Running	t2.micro	2/2 checks passed	No alarms +	us-east-1d	ec2-52-90-200-171.co...	52.90.200.171	–
□	serkan-local-name-:) co...	i-023464620e0b2a980	Pending	t2.micro	–	No alarms +	us-east-1d	ec2-35-171-17-44.com...	35.171.17.44	–



Successfully created ami-0ea8e11f8b47c1da2 from instance i-043669202d2b5e15a.

Instances (3) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	serkan-local-name-:) come from locals	i-007da121595b6fa49	Terminated	t2.micro	–	No alarms +	us-east-1e	–
□	Terraform	i-043669202d2b5e15a	Running	t2.micro	2/2 checks passed	No alarms +	us-east-1d	ec2-52-90-200-171.c
□	serkan-local-name-:) come from my ami	i-023464620e0b2a980	Running	t2.micro	–	No alarms +	us-east-1d	ec2-35-171-17-44.co

~~~~
Ami den dolayi eski ec2 destroy oldu yenisi olusmaya basladi


```
```bash
terraform plan
```
```bash
terraform apply
```
- Check EC2 instance's ami id.
- You can see which data sources can be used with a resource in the documentation
 of terraform. For example EBS snapshot.
```bash
terraform destroy
```
~~~~