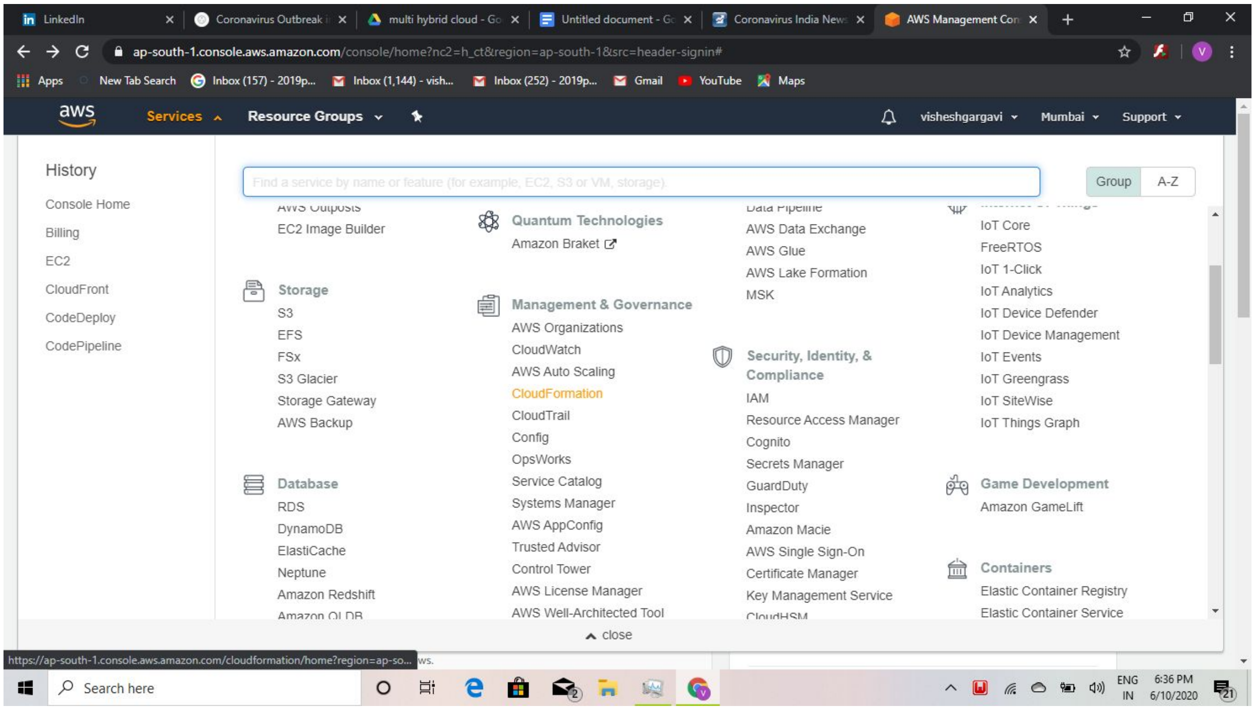
10th june:



Python works on imperative while json works on declarative

<u>“””Idempotent”””</u>

```
provider "aws" {
  region   = "ap-south-1"
  profile  = "myvishesh"
}

resource "aws_instance" "myin" {
  ami           = "ami-0447a12f28fddb066"
  instance_type = "t2.micro"
  key_name      = "mykey1111.pem"
  availability_zone = "ap-south-1a"
  security_groups = [ "launch-wizard-1" ]

  tags = {
    Name = "myin"
  }
}
resource "aws_ebs_volume" "myebs1" {
  availability_zone = "${aws_instance.myin.availability_zone}"
  size              = 1
  tags = {
    Name = "myebs1"
  }
}
resource "aws_volume_attachment" "myattach" {
 device_name = "/dev/sdf"
 volume_id = "${aws_ebs_volume.myebs1.id}"
 instance_id = "${aws_instance.myin.id}"
}
output "myoutputip" {
 value=aws_instance.myin.public_ip
}
C:\Users\user\Desktop\terraform\mytest>terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_ebs_volume.myebs1 will be created
  + resource "aws_ebs_volume" "myebs1" {
      + arn               = (known after apply)
      + availability_zone = "ap-south-1a"
      + encrypted         = (known after apply)
      + id                = (known after apply)
      + iops              = (known after apply)
      + kms_key_id        = (known after apply)
      + size              = 1
      + snapshot_id       = (known after apply)
      + tags              = {
          + "Name" = "myebs1"
        }
      + type              = (known after apply)
    }

  # aws_instance.myin will be created
  + resource "aws_instance" "myin" {
      + ami                          = "ami-0447a12f28fddb066"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = "ap-south-1a"
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = "mykey1111.pem"
      + network_interface_id         = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = [
          + "launch-wizard-1",
        ]
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name" = "myin"
        }
      + tenancy                      = (known after apply)
      + volume_tags                  = (known after apply)
      + vpc_security_group_ids       = (known after apply)

      + ebs_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + snapshot_id           = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }

      + ephemeral_block_device {
```

```
          + device_name  = (known after apply)
          + no_device    = (known after apply)
          + virtual_name = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

  # aws_volume_attachment.myattach will be created
  + resource "aws_volume_attachment" "myattach" {
      + device_name = "/dev/sdf"
      + id          = (known after apply)
      + instance_id = (known after apply)
      + volume_id   = (known after apply)
    }

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


Warning: Interpolation-only expressions are deprecated

  on ec2.tf line 18, in resource "aws_ebs_volume" "myebs1":
```

```
  18:   availability_zone = "${aws_instance.myin.availability_zone}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

(and 2 more similar warnings elsewhere)

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_instance.myin: Creating...
aws_instance.myin: Still creating... [10s elapsed]
aws_instance.myin: Still creating... [20s elapsed]
aws_instance.myin: Creation complete after 24s [id=i-0c4c9255108c4abfc]
aws_ebs_volume.myebs1: Creating...
aws_ebs_volume.myebs1: Still creating... [10s elapsed]
aws_ebs_volume.myebs1: Creation complete after 12s [id=vol-04307feb0450f1d83]
aws_volume_attachment.myattach: Creating...
aws_volume_attachment.myattach: Still creating... [10s elapsed]
aws_volume_attachment.myattach: Still creating... [20s elapsed]
aws_volume_attachment.myattach: Creation complete after 22s [id=vai-2074347430]

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

Outputs:

myoutputip = 13.126.159.113
C:\Users\user\Desktop\terraform\mytest>cd ..

C:\Users\user\Desktop\terraform>mkdir variable

C:\Users\user\Desktop\terraform>cd variable
C:\Users\user\Desktop\terraform\variable>dir
```

```
 Volume in drive C is vishesh
 Volume Serial Number is 1CF6-F84B

 Directory of C:\Users\user\Desktop\terraform\variable

06/10/2020  07:44 PM    <DIR>          .
06/10/2020  07:44 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  181,393,702,912 bytes free

C:\Users\user\Desktop\terraform\variable>terraform apply

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

Outputs:

myvalue = LinuxWorld

C:\Users\user\Desktop\terraform\variable>terraform 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


Destroy complete! Resources: 0 destroyed.

C:\Users\user\Desktop\terraform\variable>terraform apply

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

Outputs:

myvalue = LinuxWorld
```

```
variable "x" {
 type = string
 default = "LinuxWorld"
}

output "myvalue" {
 value = ${var.x}
}

**************************************************************************************************************
provider "aws" {
  region   = "ap-south-1"
  profile  = "myvishesh"
}


variable "mykey"{
 type = string
// default = "mykey1111.pem"
}

resource "aws_instance" "myin" {
  ami           = "ami-0447a12f28fddb066"
  instance_type = "t2.micro"
  key_name      = var.mykey
  availability_zone = "ap-south-1a"
  security_groups = [ "launch-wizard-1" ]

  tags = {
    Name = "myin"
  }
}
resource "aws_ebs_volume" "myebs1" {
  availability_zone = "${aws_instance.myin.availability_zone}"
  size              = 1

  tags = {
    Name = "myebs1"
  }
}
resource "aws_volume_attachment" "myattach" {
 device_name = "/dev/sdf"
 volume_id = "${aws_ebs_volume.myebs1.id}"
 instance_id = "${aws_instance.myin.id}"
}
output "myoutputip" {
 value=aws_instance.myin.public_ip
}
C:\Users\user\Desktop\terraform\mytest>terraform apply
```

```
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_ebs_volume.myebs1 will be created
  + resource "aws_ebs_volume" "myebs1" {
      + arn               = (known after apply)
      + availability_zone = "ap-south-1a"
      + encrypted         = (known after apply)
      + id                = (known after apply)
      + iops              = (known after apply)
      + kms_key_id        = (known after apply)
      + size              = 1
      + snapshot_id       = (known after apply)
      + tags              = {
          + "Name" = "myebs1"
        }
      + type              = (known after apply)
    }

  # aws_instance.myin will be created
  + resource "aws_instance" "myin" {
      + ami                          = "ami-0447a12f28fddb066"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = "ap-south-1a"
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = "mykey1111.pem"
      + network_interface_id         = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = [
          + "launch-wizard-1",
        ]
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name" = "myin"
        }
      + tenancy                      = (known after apply)
      + volume_tags                  = (known after apply)
      + vpc_security_group_ids       = (known after apply)

      + ebs_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + snapshot_id           = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }

      + ephemeral_block_device {
          + device_name  = (known after apply)
          + no_device    = (known after apply)
          + virtual_name = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
```

```
          + network_interface_id  = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

  # aws_volume_attachment.myattach will be created
  + resource "aws_volume_attachment" "myattach" {
      + device_name = "/dev/sdf"
      + id          = (known after apply)
      + instance_id = (known after apply)
      + volume_id   = (known after apply)
    }

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


Warning: Interpolation-only expressions are deprecated

  on ec2.tf line 23, in resource "aws_ebs_volume" "myebs1":
  23:   availability_zone = "${aws_instance.myin.availability_zone}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

(and 2 more similar warnings elsewhere)
```

```
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_instance.myin: Creating...
aws_instance.myin: Still creating... [10s elapsed]
aws_instance.myin: Still creating... [20s elapsed]
aws_instance.myin: Creation complete after 24s [id=i-04c7ebd6c2d844050]
aws_ebs_volume.myebs1: Creating...
aws_ebs_volume.myebs1: Still creating... [10s elapsed]
aws_ebs_volume.myebs1: Creation complete after 11s [id=vol-0324ef7df45a5d676]
aws_volume_attachment.myattach: Creating...
aws_volume_attachment.myattach: Still creating... [10s elapsed]
aws_volume_attachment.myattach: Still creating... [20s elapsed]
aws_volume_attachment.myattach: Creation complete after 23s [id=vai-3758746420]

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

Outputs:

myoutputip = 13.233.224.162
```

important

```
C:\Users\user\Desktop\terraform\mytest>terraform apply
var.mykey
  Enter a value: mykey.pem


An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_ebs_volume.myebs1 will be created
  + resource "aws_ebs_volume" "myebs1" {
      + arn               = (known after apply)
      + availability_zone = "ap-south-1a"
      + encrypted         = (known after apply)
      + id                = (known after apply)
      + iops              = (known after apply)
```

```
    + kms_key_id        = (known after apply)
    + size              = 1
    + snapshot_id       = (known after apply)
    + tags              = {
        + "Name" = "myebs1"
      }
    + type              = (known after apply)
  }

  # aws_instance.myin will be created
  + resource "aws_instance" "myin" {
      + ami                          = "ami-0447a12f28fddb066"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = "ap-south-1a"
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = "mykey.pem"
      + network_interface_id         = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = [
          + "launch-wizard-1",
        ]
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name" = "myin"
        }
      + tenancy                      = (known after apply)
```

```
+ volume_tags                  = (known after apply)
+ vpc_security_group_ids       = (known after apply)

+ ebs_block_device {
    + delete_on_termination = (known after apply)
    + device_name           = (known after apply)
    + encrypted             = (known after apply)
    + iops                  = (known after apply)
    + kms_key_id            = (known after apply)
    + snapshot_id           = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (known after apply)
  }

+ ephemeral_block_device {
    + device_name  = (known after apply)
    + no_device    = (known after apply)
    + virtual_name = (known after apply)
  }

+ metadata_options {
    + http_endpoint               = (known after apply)
    + http_put_response_hop_limit = (known after apply)
    + http_tokens                 = (known after apply)
  }

+ network_interface {
    + delete_on_termination = (known after apply)
    + device_index          = (known after apply)
    + network_interface_id  = (known after apply)
  }

+ root_block_device {
    + delete_on_termination = (known after apply)
    + device_name           = (known after apply)
    + encrypted             = (known after apply)
    + iops                  = (known after apply)
    + kms_key_id            = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (known after apply)
  }
```

```
    }

  # aws_volume_attachment.myattach will be created
  + resource "aws_volume_attachment" "myattach" {
      + device_name = "/dev/sdf"
      + id          = (known after apply)
      + instance_id = (known after apply)
      + volume_id   = (known after apply)
    }

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


Warning: Interpolation-only expressions are deprecated

  on ec2.tf line 24, in resource "aws_ebs_volume" "myebs1":
  24:   availability_zone = "${aws_instance.myin.availability_zone}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

(and 2 more similar warnings elsewhere)

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_instance.myin: Creating...

Error: Error launching source instance: InvalidKeyPair.NotFound: The key pair 'mykey.pem'
does not exist
        status code: 400, request id: 3c0fa500-9a07-44e2-9e4e-3825530135ca

  on ec2.tf line 12, in resource "aws_instance" "myin":
```

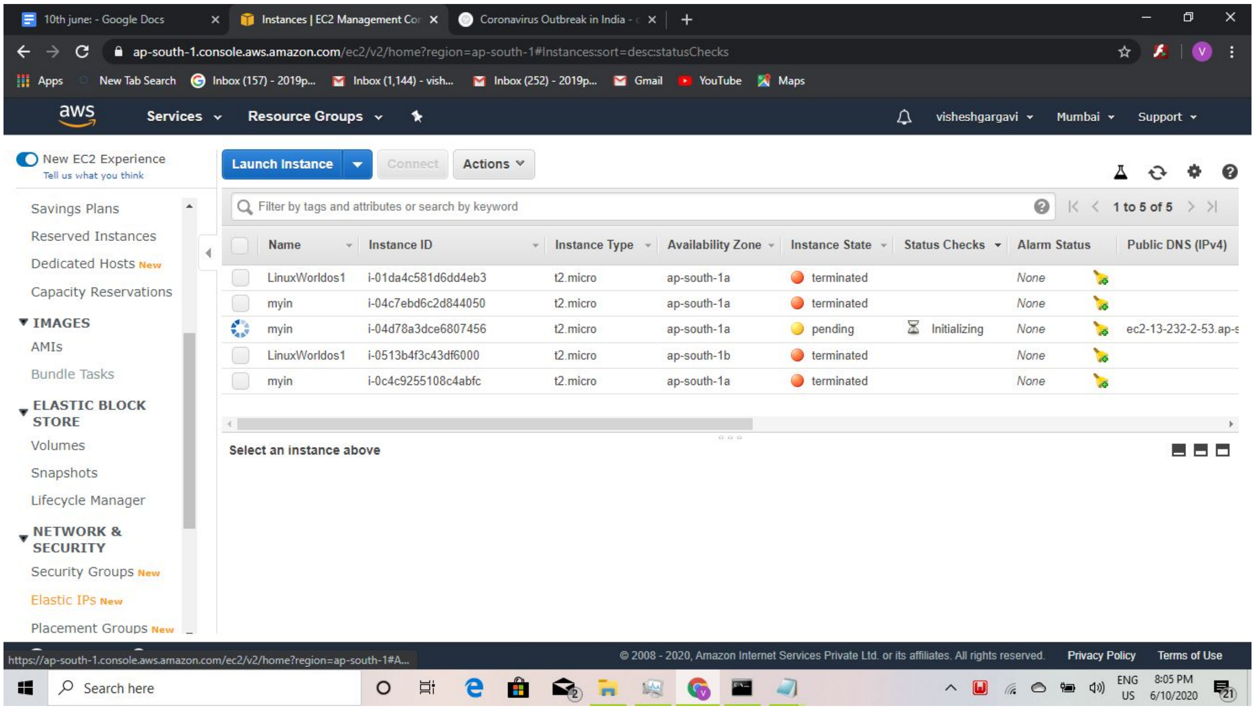
12: resource "aws_instance" "myin" {

****

ELASTIC IP(EIP)

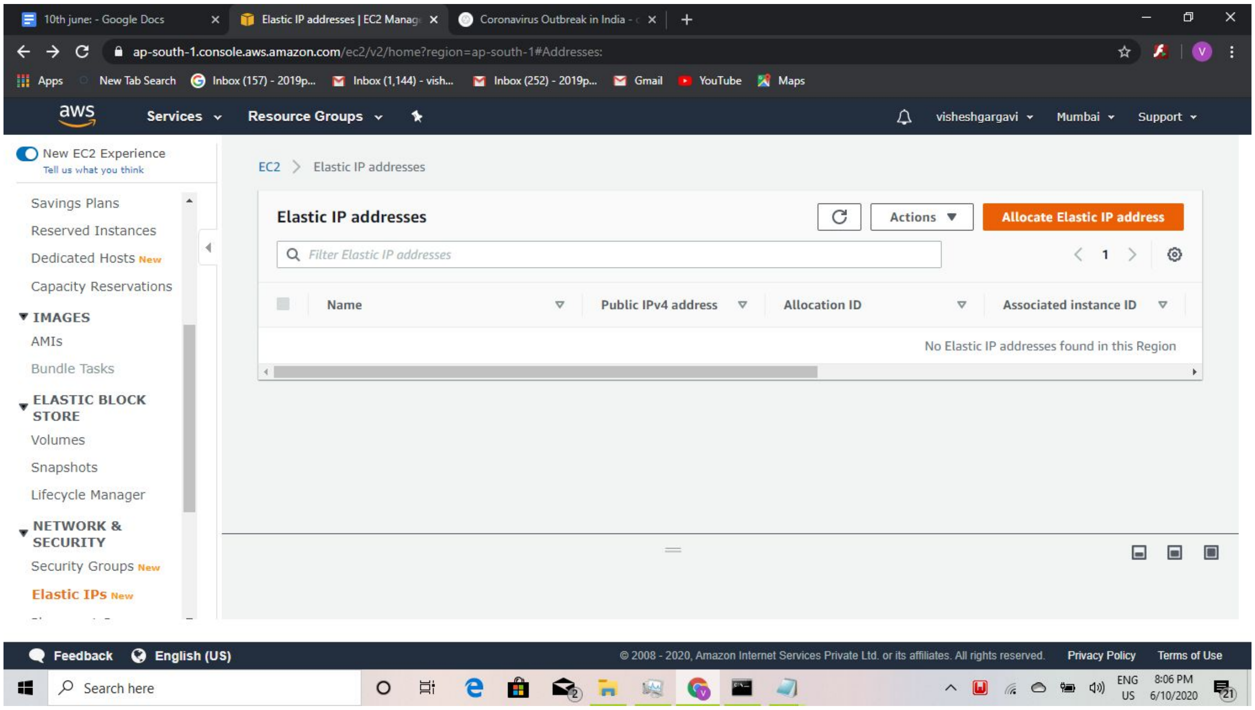
>they charge u per hourly

>static public ip



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
LinuxWorldos1	i-01da4c581d6dd4eb3	t2.micro	ap-south-1a	terminated		None	
myin	i-04c7ebd6c2d844050	t2.micro	ap-south-1a	terminated		None	
myin	i-04d78a3dce6807456	t2.micro	ap-south-1a	pending	Initializing	None	ec2-13-232-2-53.ap-s
LinuxWorldos1	i-0513b4f3c43df6000	t2.micro	ap-south-1b	terminated		None	
myin	i-0c4c9255108c4abfc	t2.micro	ap-south-1a	terminated		None	

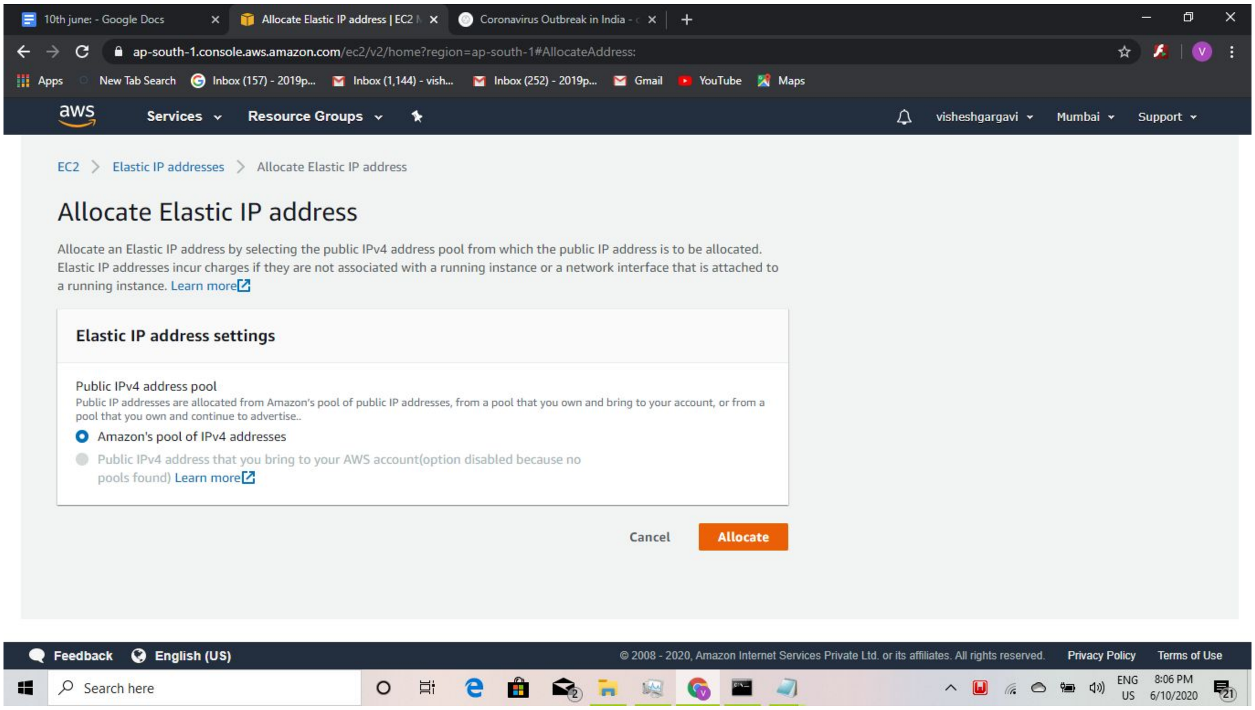
Select an instance above



Elastic IP addresses

Name	Public IPv4 address	Allocation ID	Associated instance ID

No Elastic IP addresses found in this Region



EC2 > Elastic IP addresses > Allocate Elastic IP address

Allocate Elastic IP address

Allocate an Elastic IP address by selecting the public IPv4 address pool from which the public IP address is to be allocated. Elastic IP addresses incur charges if they are not associated with a running instance or a network interface that is attached to a running instance. Learn more

Elastic IP address settings

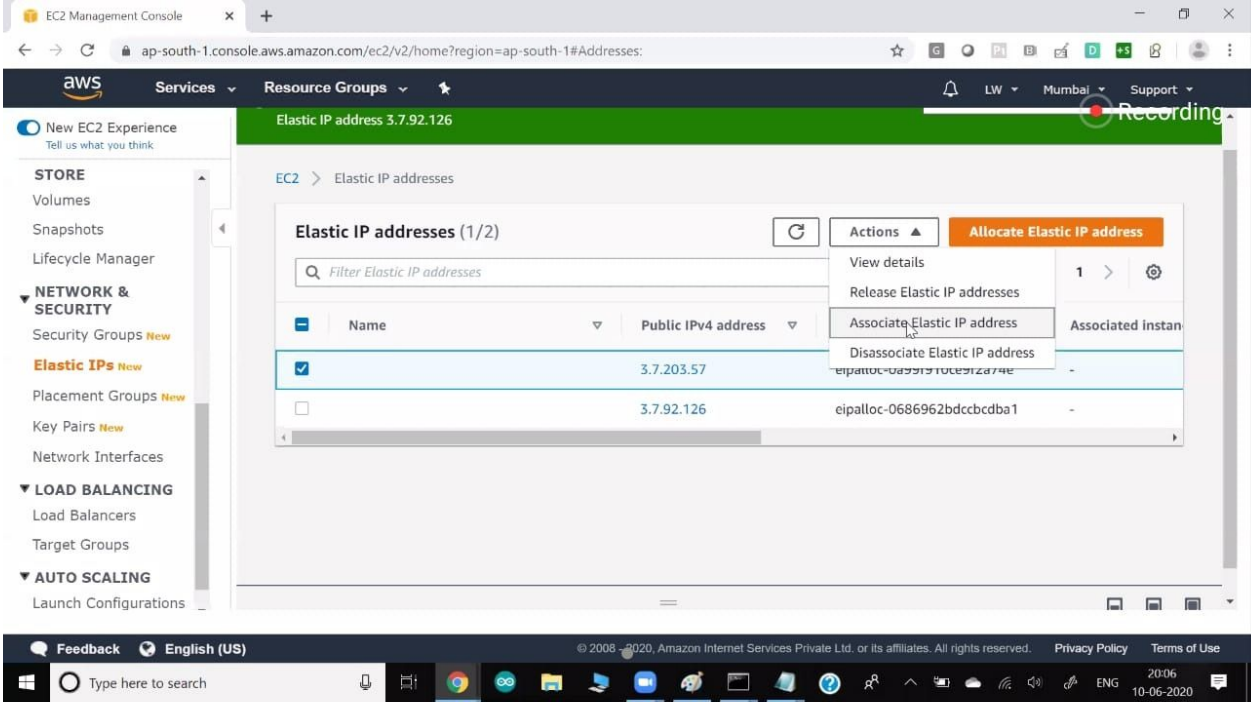
Public IPv4 address pool

Public IP addresses are allocated from Amazon's pool of public IP addresses, from a pool that you own and bring to your account, or from a pool that you own and continue to advertise..

● Amazon's pool of IPv4 addresses

○ Public IPv4 address that you bring to your AWS account(option disabled because no pools found) Learn more

Cancel   **Allocate**



Elastic IP address 3.7.92.126

EC2 > Elastic IP addresses

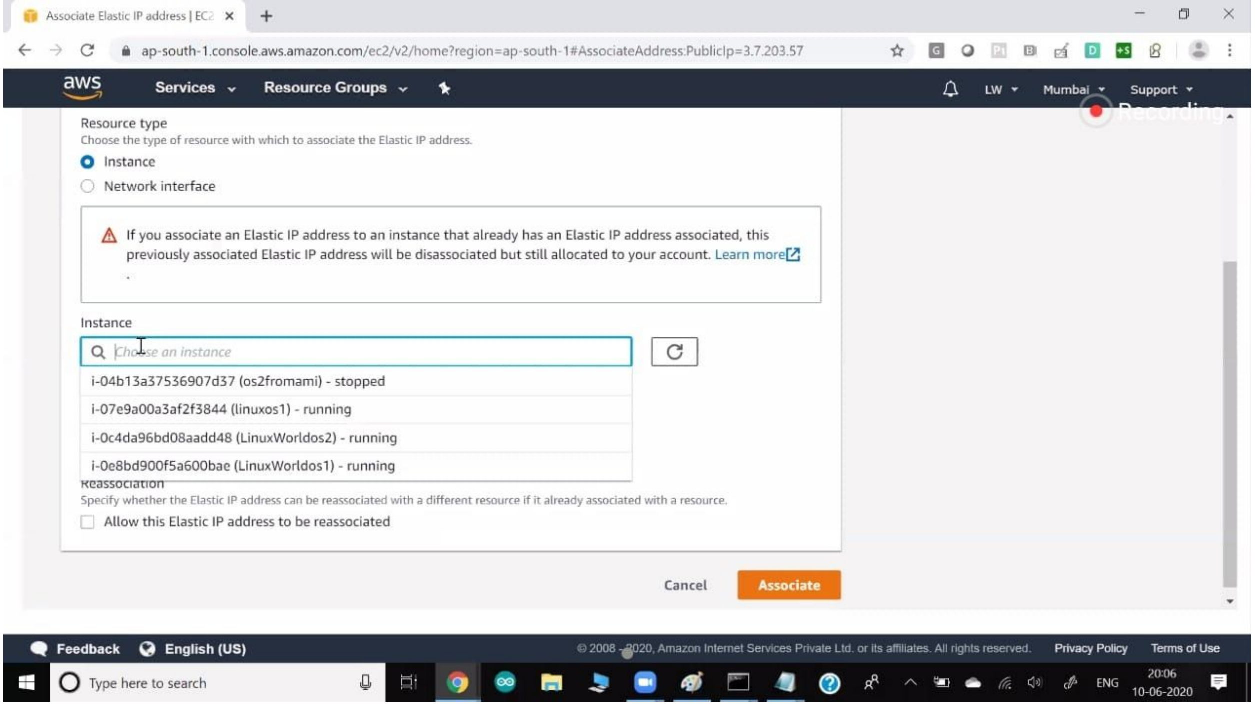
Elastic IP addresses (1/2)

Actions ▲   **Allocate Elastic IP address**

Filter Elastic IP addresses

- View details
- Release Elastic IP addresses
- Associate Elastic IP address
- Disassociate Elastic IP address

	Name	Public IPv4 address		Associated instan
☑		3.7.203.57	eipalloc-0a99f910ce9f2a74e	-
□		3.7.92.126	eipalloc-0686962bdccbcdba1	-



Associate Elastic IP address | EC2
ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#AssociateAddress:PublicIp=3.7.203.57
aws
Services
Resource Groups
LW
Mumbai
Support
Recording

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more

.

Instance

Choose an instance

i-04b13a37536907d37 (os2fromami) - stopped

i-07e9a00a3af2f3844 (linuxos1) - running

i-0c4da96bd08aadd48 (LinuxWorldos2) - running

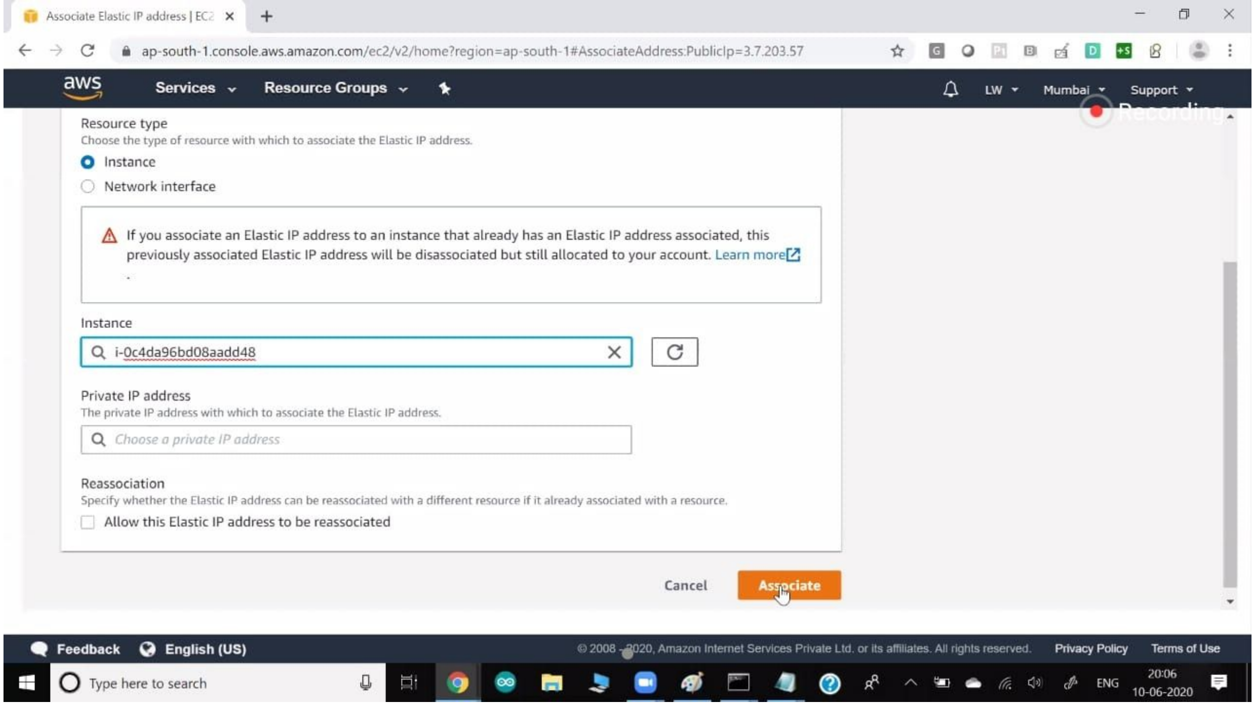
i-0e8bd900f5a600bae (LinuxWorldos1) - running

Reassociation

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

□ Allow this Elastic IP address to be reassociated

Cancel
Associate
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
ENG
20:06
10-06-2020



Associate Elastic IP address | EC2
ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#AssociateAddress:PublicIp=3.7.203.57
aws
Services
Resource Groups
LW
Mumbai
Support
Recording

Resource type

Choose the type of resource with which to associate the Elastic IP address.

● Instance

○ Network interface

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more

.

Instance

i-0c4da96bd08aadd48

Private IP address

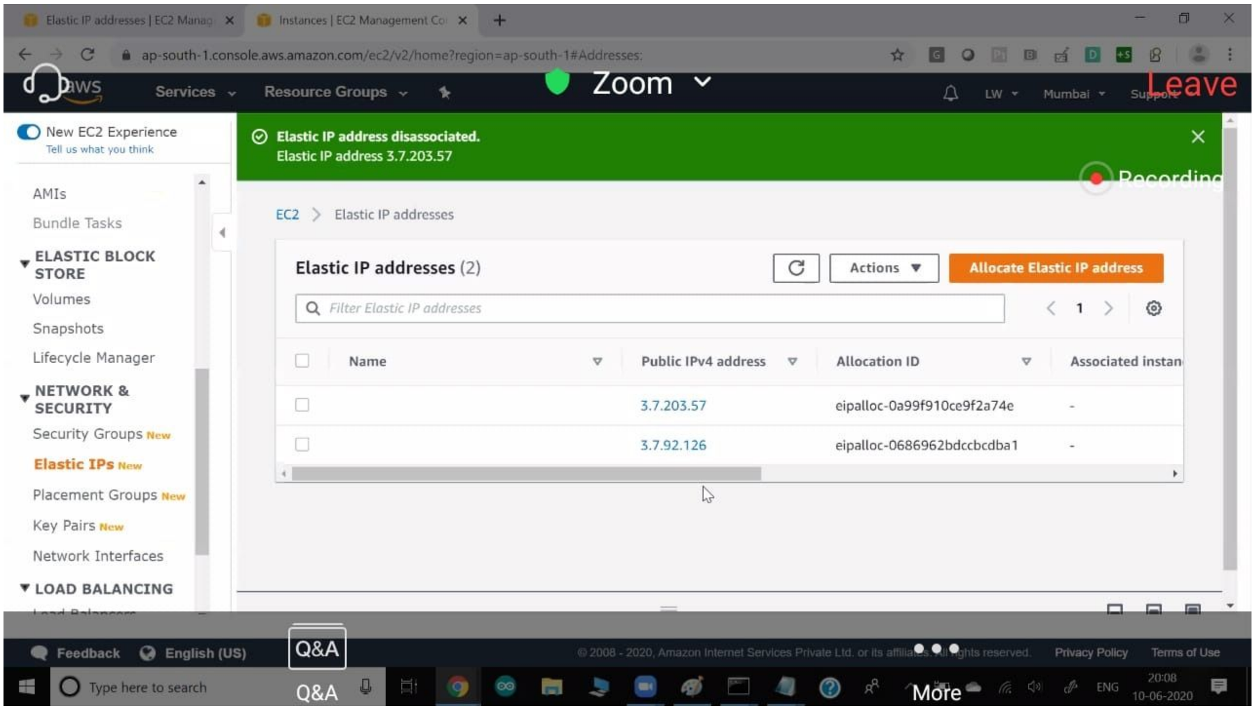
The private IP address with which to associate the Elastic IP address.

Choose a private IP address

Reassociation

Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

□ Allow this Elastic IP address to be reassociated

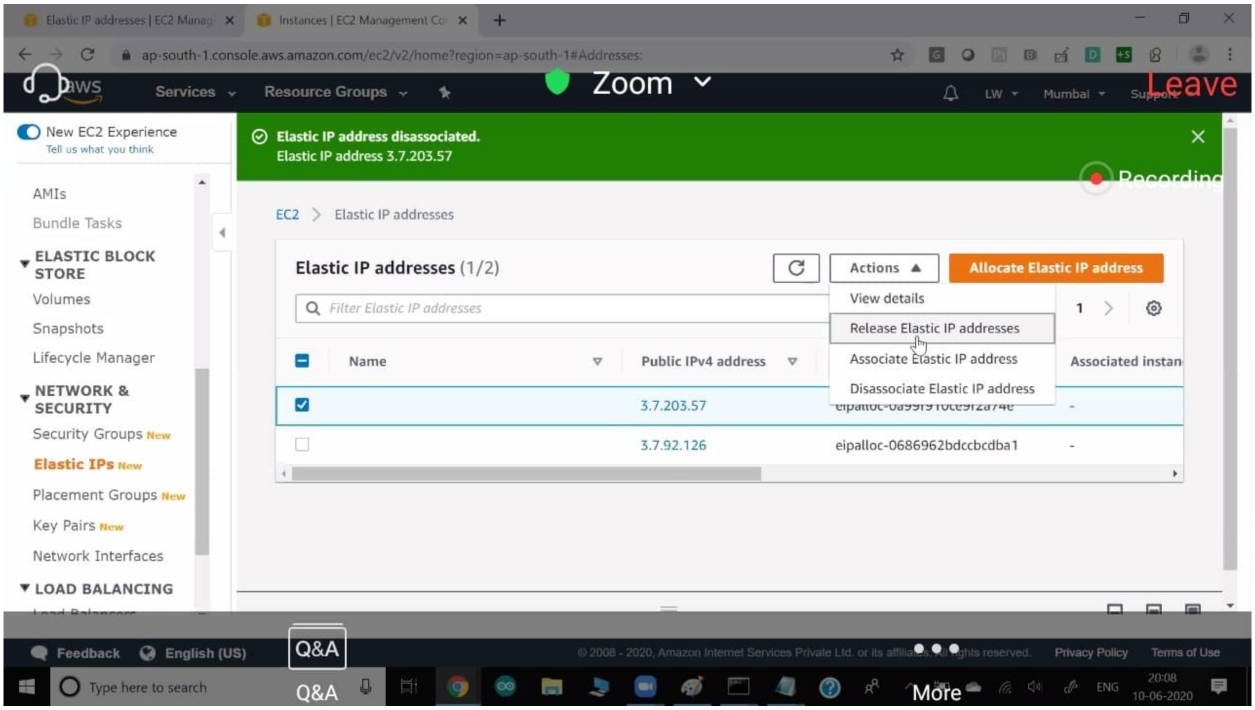


Elastic IP address disassociated.
Elastic IP address 3.7.203.57

EC2 > Elastic IP addresses

Elastic IP addresses (2)

☐	Name	Public IPv4 address	Allocation ID	Associated instan
☐		3.7.203.57	eipalloc-0a99f910ce9f2a74e	-
☐		3.7.92.126	eipalloc-0686962bdccbcdba1	-

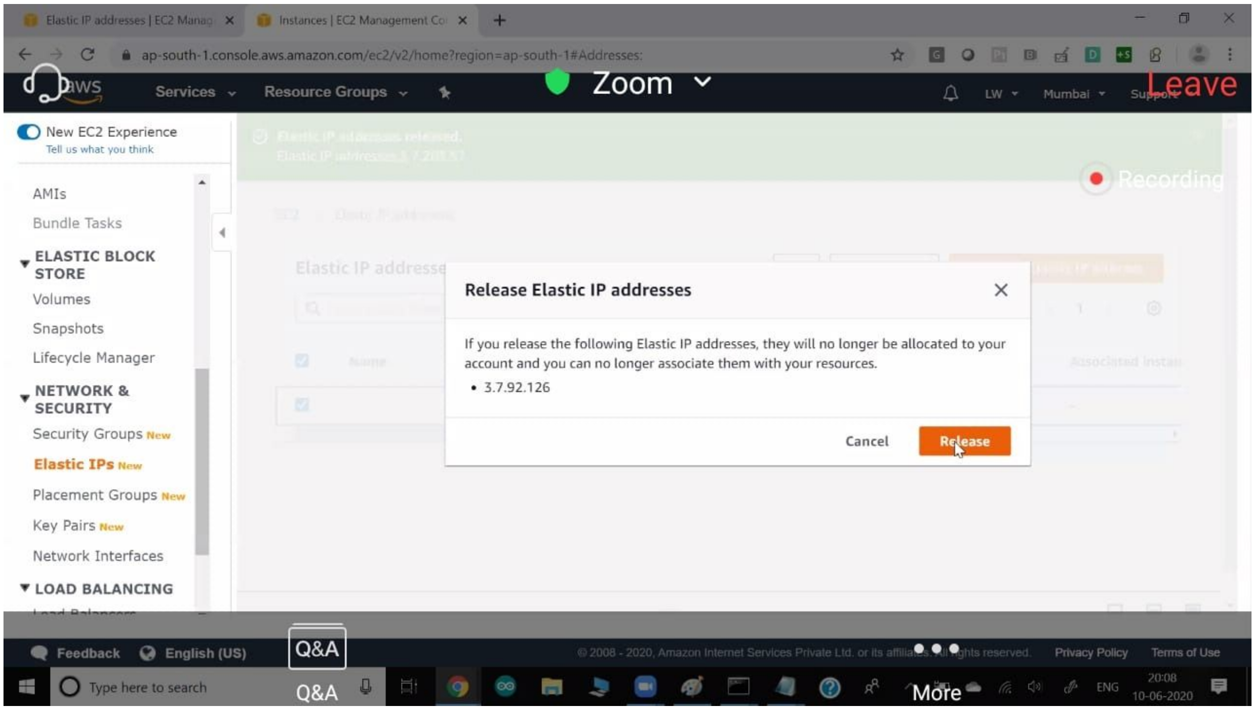


Elastic IP address disassociated.
Elastic IP address 3.7.203.57

EC2 > Elastic IP addresses

Elastic IP addresses (1/2)

☐	Name	Public IPv4 address	Allocation ID	Associated instan
☑		3.7.203.57	eipalloc-0a99f910ce9f2a74e	-
☐		3.7.92.126	eipalloc-0686962bdccbcdba1	-

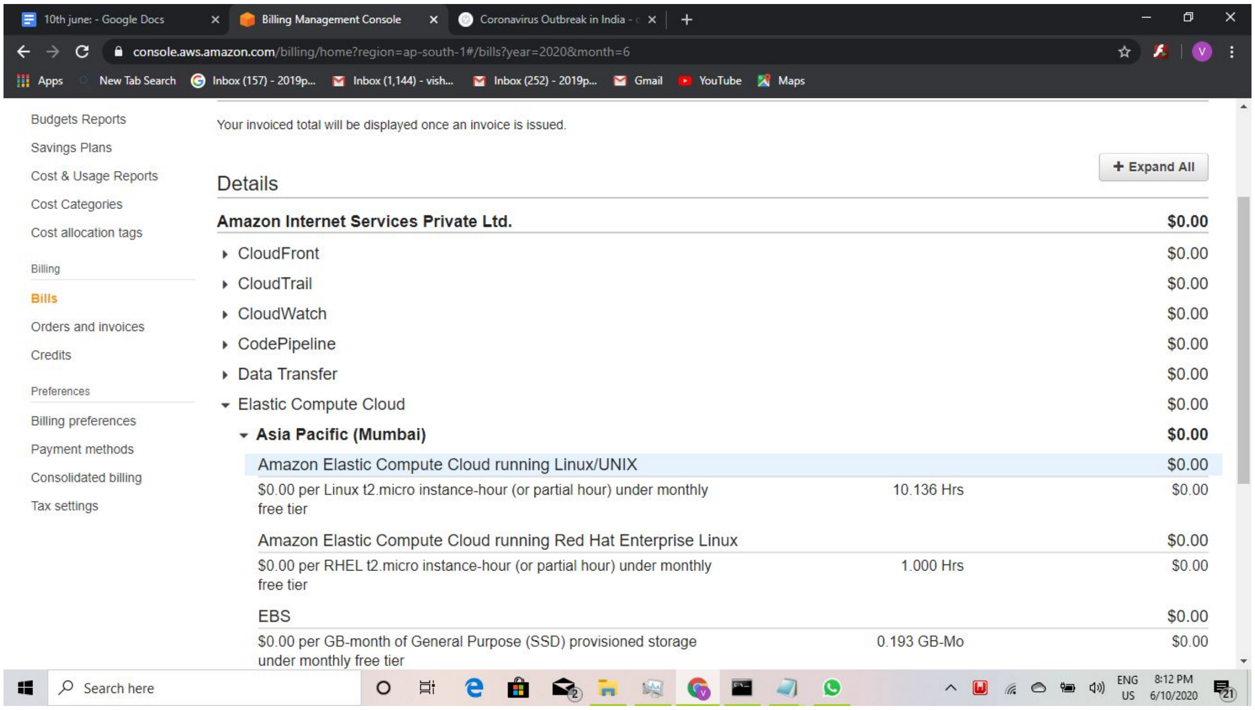


Release Elastic IP addresses

If you release the following Elastic IP addresses, they will no longer be allocated to your account and you can no longer associate them with your resources.

- 3.7.92.126

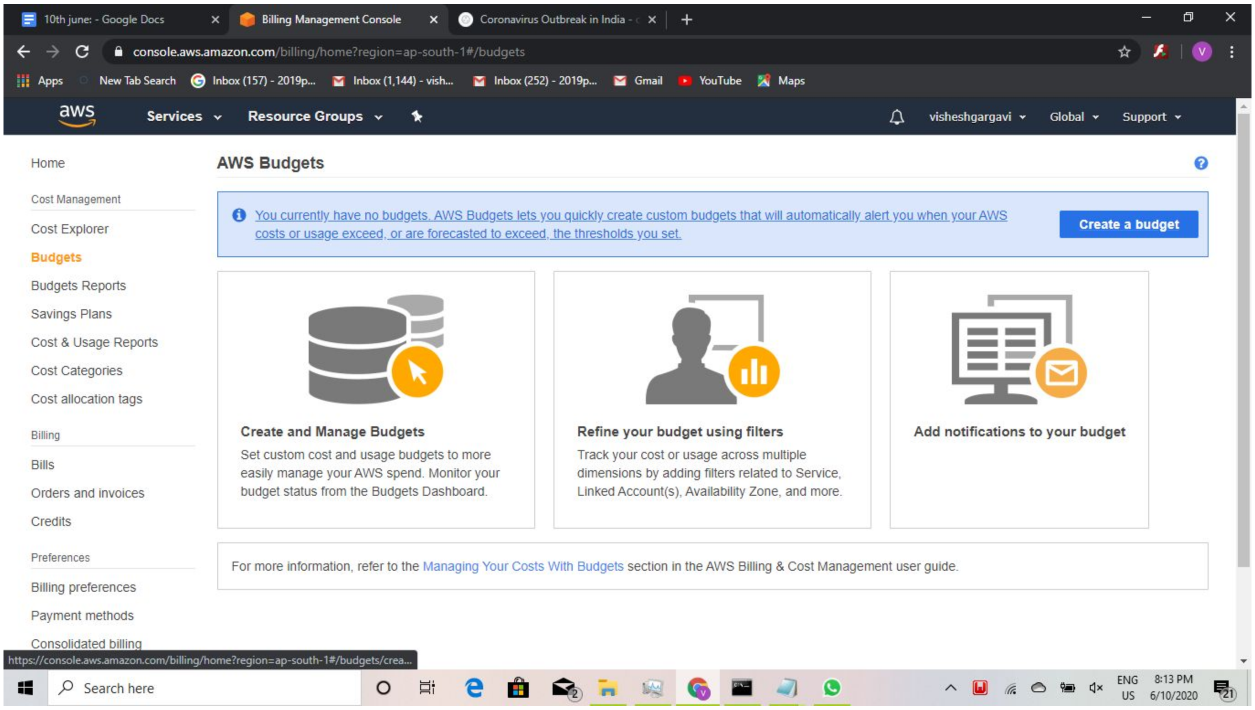
Cancel Release



Your invoiced total will be displayed once an invoice is issued.

Details

Amazon Internet Services Private Ltd.		**$0.00**
CloudFront		$0.00
CloudTrail		$0.00
CloudWatch		$0.00
CodePipeline		$0.00
Data Transfer		$0.00
Elastic Compute Cloud		$0.00
Asia Pacific (Mumbai)		**$0.00**
Amazon Elastic Compute Cloud running Linux/UNIX		$0.00
$0.00 per Linux t2.micro instance-hour (or partial hour) under monthly free tier	10.136 Hrs	$0.00
Amazon Elastic Compute Cloud running Red Hat Enterprise Linux		$0.00
$0.00 per RHEL t2.micro instance-hour (or partial hour) under monthly free tier	1.000 Hrs	$0.00
EBS		$0.00
$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier	0.193 GB-Mo	$0.00



AWS Budgets

You currently have no budgets. AWS Budgets lets you quickly create custom budgets that will automatically alert you when your AWS costs or usage exceed, or are forecasted to exceed, the thresholds you set.

Create and Manage Budgets

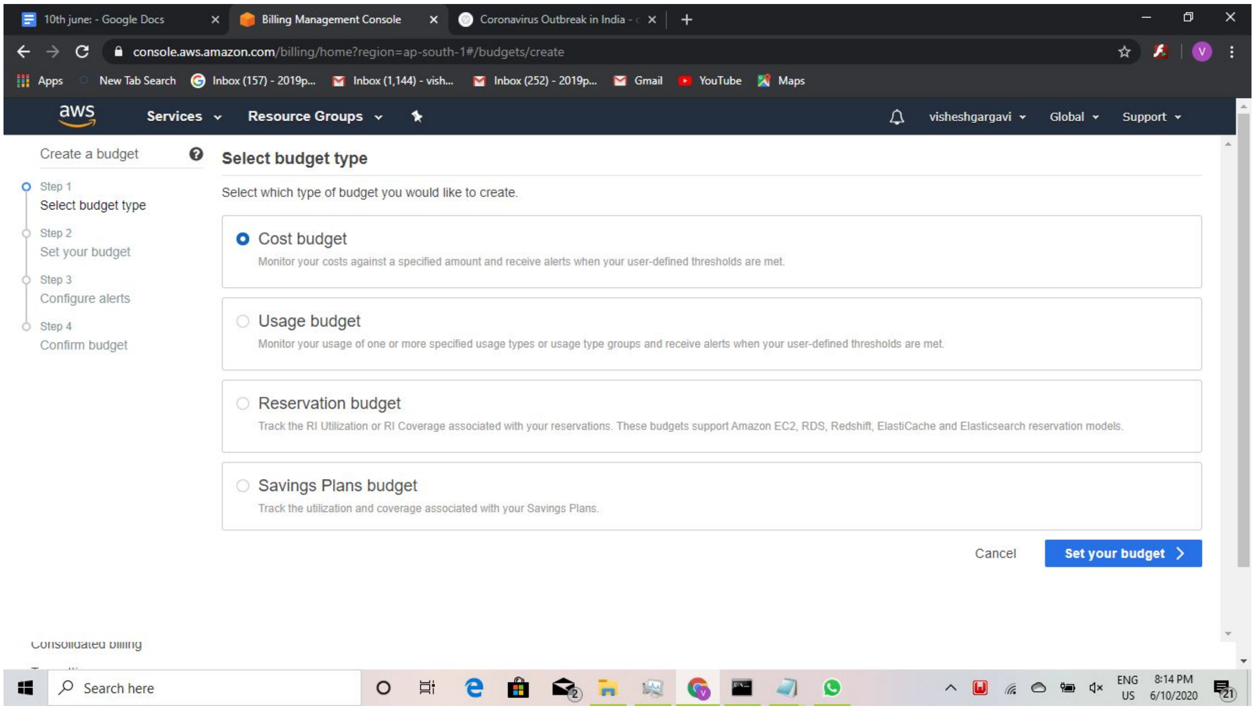
Set custom cost and usage budgets to more easily manage your AWS spend. Monitor your budget status from the Budgets Dashboard.

Refine your budget using filters

Track your cost or usage across multiple dimensions by adding filters related to Service, Linked Account(s), Availability Zone, and more.

Add notifications to your budget

For more information, refer to the Managing Your Costs With Budgets section in the AWS Billing & Cost Management user guide.



Select budget type

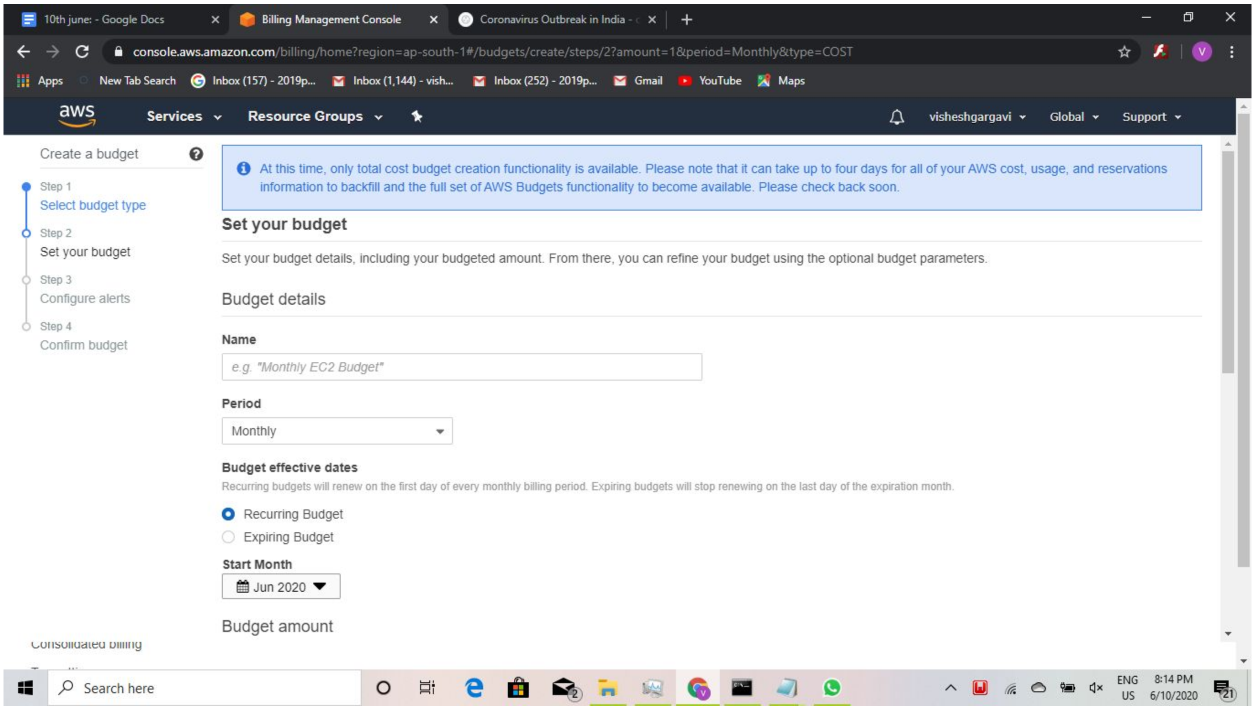
Select which type of budget you would like to create.

● Cost budget
Monitor your costs against a specified amount and receive alerts when your user-defined thresholds are met.

○ Usage budget
Monitor your usage of one or more specified usage types or usage type groups and receive alerts when your user-defined thresholds are met.

○ Reservation budget
Track the RI Utilization or RI Coverage associated with your reservations. These budgets support Amazon EC2, RDS, Redshift, ElastiCache and Elasticsearch reservation models.

○ Savings Plans budget
Track the utilization and coverage associated with your Savings Plans.



Create a budget

Step 1
Select budget type

Step 2
Set your budget

Step 3
Configure alerts

Step 4
Confirm budget

At this time, only total cost budget creation functionality is available. Please note that it can take up to four days for all of your AWS cost, usage, and reservations information to backfill and the full set of AWS Budgets functionality to become available. Please check back soon.

Set your budget

Set your budget details, including your budgeted amount. From there, you can refine your budget using the optional budget parameters.

Budget details

Name

e.g. "Monthly EC2 Budget"

Period

Monthly

Budget effective dates

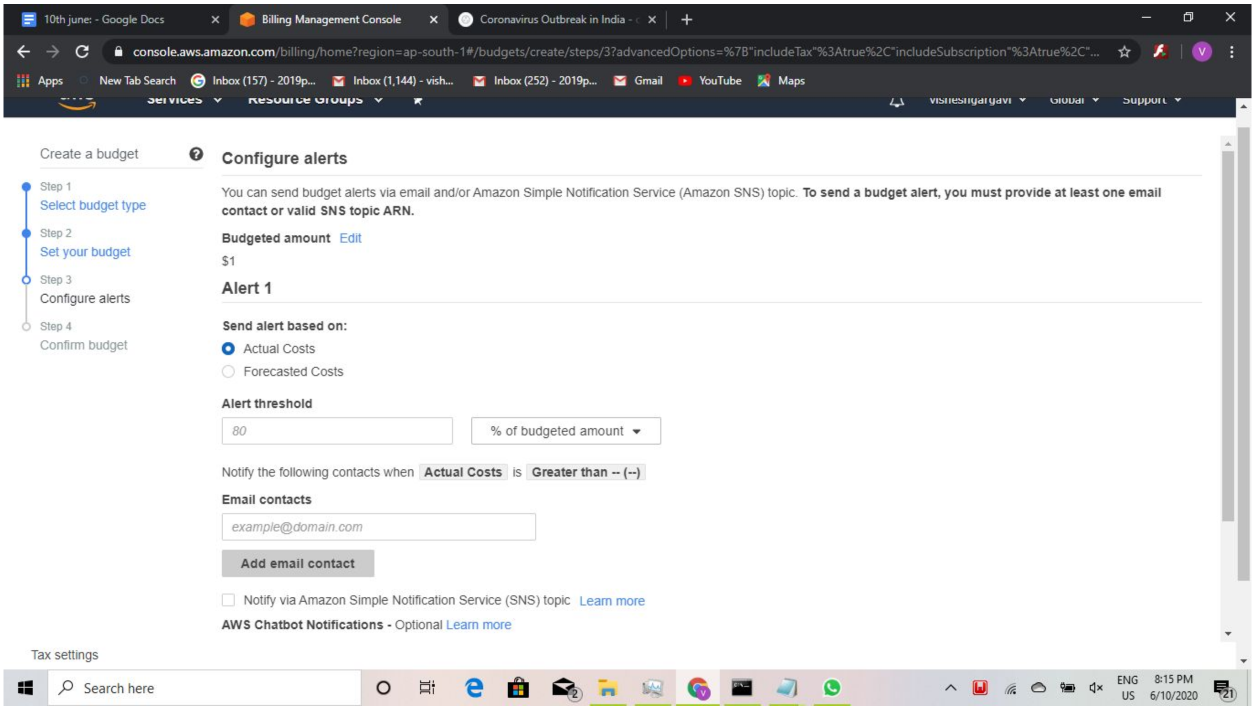
Recurring budgets will renew on the first day of every monthly billing period. Expiring budgets will stop renewing on the last day of the expiration month.

● Recurring Budget
○ Expiring Budget

Start Month

Jun 2020

Budget amount



Create a budget

Step 1
Select budget type

Step 2
Set your budget

Step 3
Configure alerts

Step 4
Confirm budget

Configure alerts

You can send budget alerts via email and/or Amazon Simple Notification Service (Amazon SNS) topic. **To send a budget alert, you must provide at least one email contact or valid SNS topic ARN.**

Budgeted amount Edit

$1

Alert 1

Send alert based on:

● Actual Costs
○ Forecasted Costs

Alert threshold

80 | % of budgeted amount

Notify the following contacts when **Actual Costs** is **Greater than -- (--)**

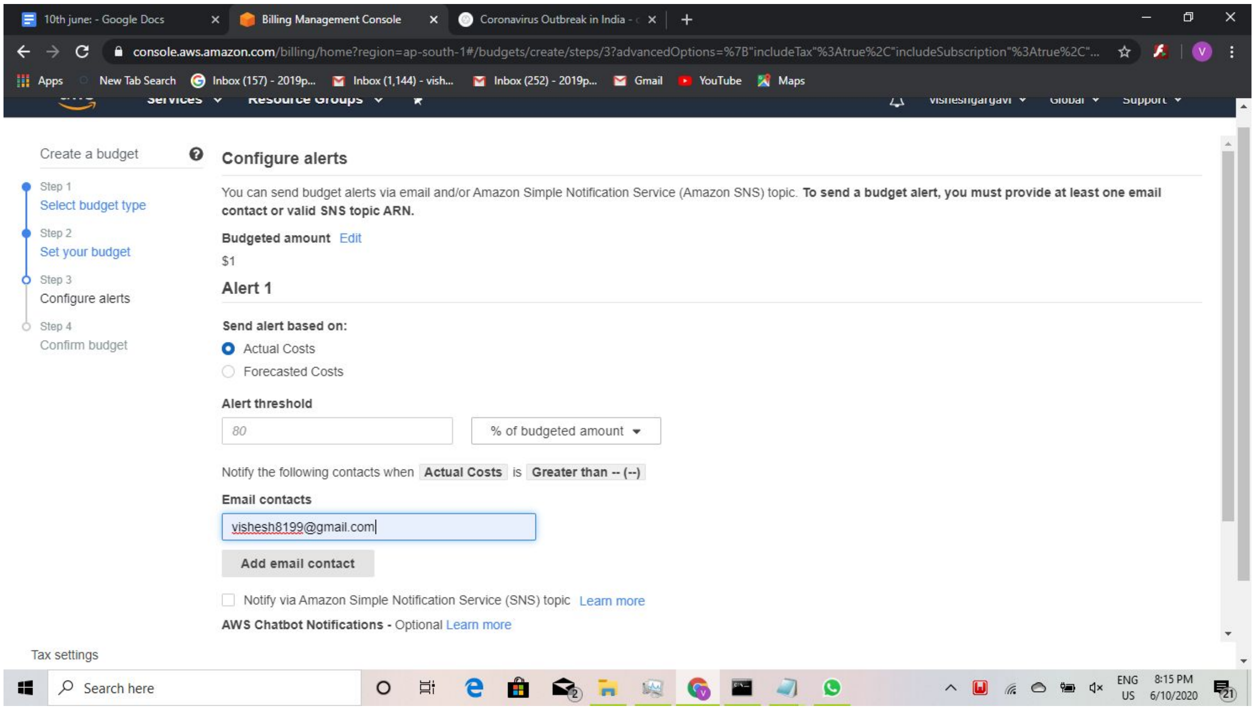
Email contacts

example@domain.com

Add email contact

☐ Notify via Amazon Simple Notification Service (SNS) topic Learn more

AWS Chatbot Notifications - Optional Learn more



Configure alerts

You can send budget alerts via email and/or Amazon Simple Notification Service (Amazon SNS) topic. **To send a budget alert, you must provide at least one email contact or valid SNS topic ARN.**

Budgeted amount Edit

$1

Alert 1

Send alert based on:

● Actual Costs

○ Forecasted Costs

Alert threshold

80 | % of budgeted amount

Notify the following contacts when **Actual Costs** is **Greater than -- (--)**

Email contacts

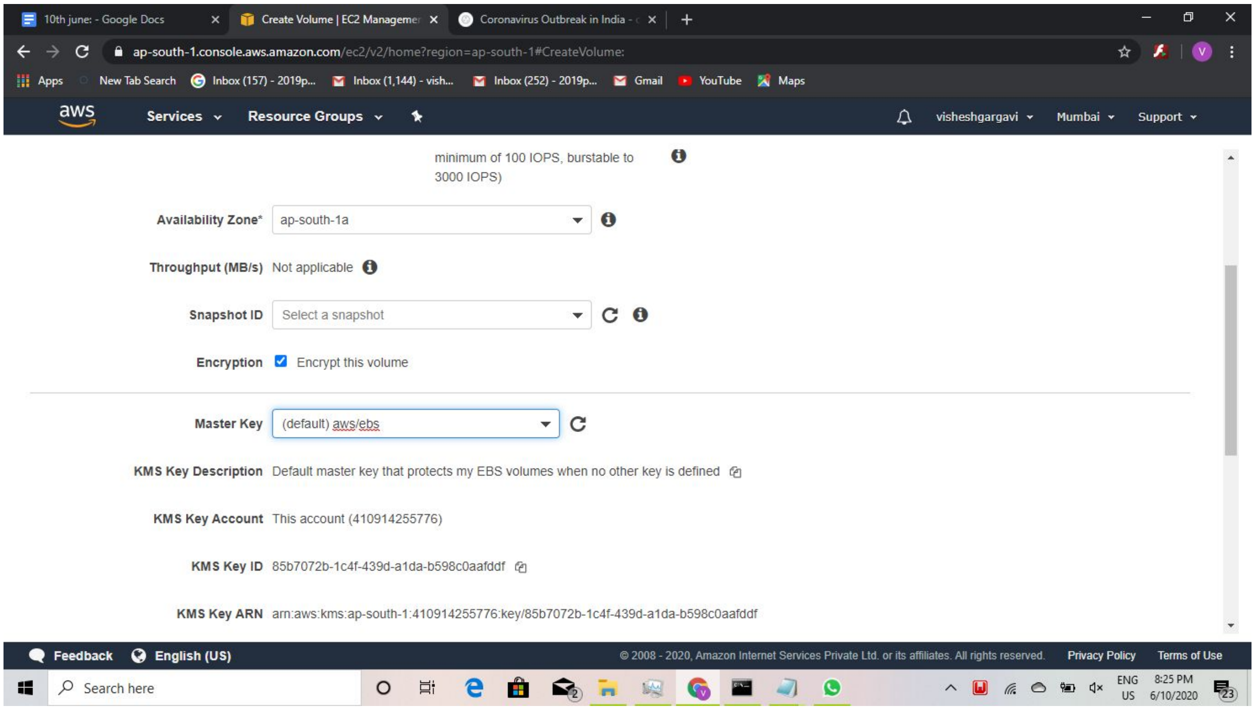
vishesh8199@gmail.com

Add email contact

□ Notify via Amazon Simple Notification Service (SNS) topic Learn more

AWS Chatbot Notifications - Optional Learn more

encrypt(lock) the hardisk: data encryption



minimum of 100 IOPS, burstable to 3000 IOPS)

Availability Zone* ap-south-1a

Throughput (MB/s) Not applicable

Snapshot ID Select a snapshot

Encryption ☑ Encrypt this volume

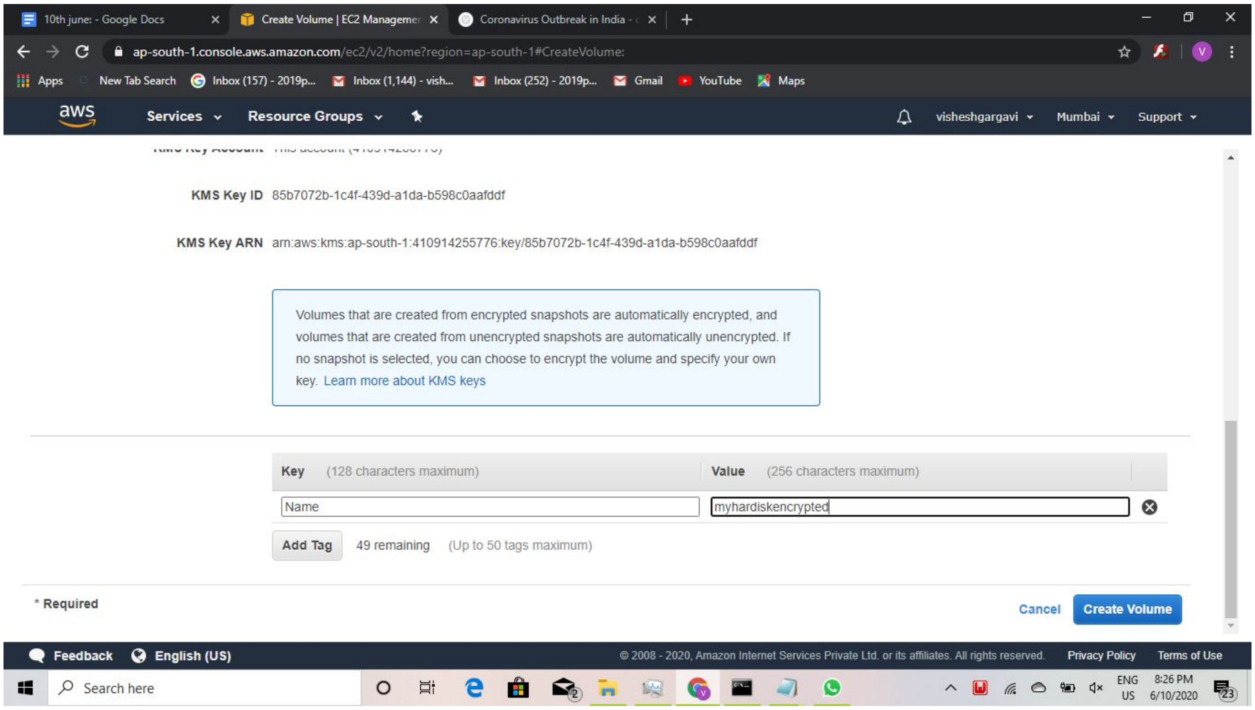
Master Key (default) aws/ebs

KMS Key Description Default master key that protects my EBS volumes when no other key is defined

KMS Key Account This account (410914255776)

KMS Key ID 85b7072b-1c4f-439d-a1da-b598c0aafddf

KMS Key ARN arn:aws:kms:ap-south-1:410914255776:key/85b7072b-1c4f-439d-a1da-b598c0aafddf



KMS Key ID 85b7072b-1c4f-439d-a1da-b598c0aafddf

KMS Key ARN arn:aws:kms:ap-south-1:410914255776:key/85b7072b-1c4f-439d-a1da-b598c0aafddf

Volumes that are created from encrypted snapshots are automatically encrypted, and volumes that are created from unencrypted snapshots are automatically unencrypted. If no snapshot is selected, you can choose to encrypt the volume and specify your own key. Learn more about KMS keys

Key (128 characters maximum)	Value (256 characters maximum)
Name	myhardiskencrypted

Add Tag 49 remaining (Up to 50 tags maximum)

* Required

```
C:\Users\user\Desktop\cloud>ssh -l ec2-user 13.233.112.182 -i mykey1111.pem

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-40-100 ~]$ sudo su
[root@ip-172-31-40-100 ec2-user]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 0D7E094F-1C63-43F2-AC82-3316E992075B

Device       Start      End  Sectors Size Type
/dev/xvda1    4096 16777182 16773087   8G Linux filesystem
/dev/xvda128  2048     4095     2048   1M BIOS boot
```

```
Partition table entries are not in disk order.


Disk /dev/xvdf: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@ip-172-31-40-100 ec2-user]# fdisk /dev/xvdf1

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

fdisk: cannot open /dev/xvdf1: No such file or directory
[root@ip-172-31-40-100 ec2-user]# fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x5a23eac2.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151):

Created a new partition 1 of type 'Linux' and of size 1023 MiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@ip-172-31-40-100 ec2-user]# ldisk
```

```
bash: ldisk: command not found
[root@ip-172-31-40-100 ec2-user]#
[root@ip-172-31-40-100 ec2-user]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdf    202:80   0    1G  0 disk
└─xvdf1 202:81   0 1023M  0 part
[root@ip-172-31-40-100 ec2-user]#
```