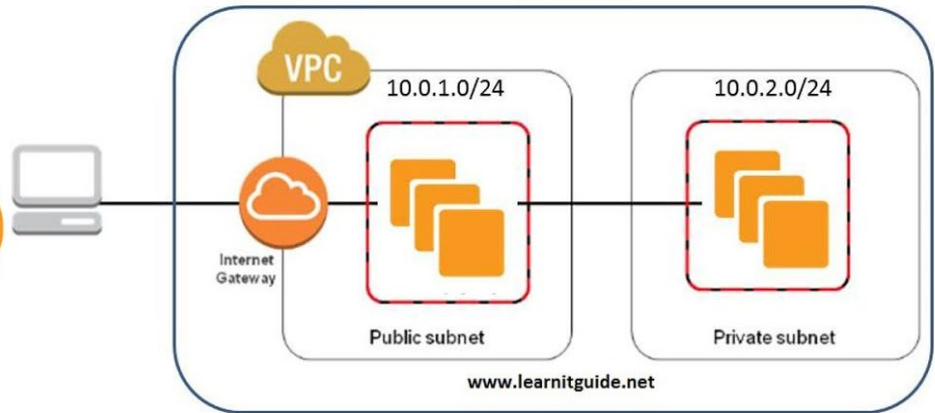


AWS VPC

10.0.0.0/16



HOW TO CREATE NEW VPC SUBNETS, INTERNET GATEWAY



Terraform



Amazon Machine Image (AMI)



Amazon EC2



VPC NAT gateway

Task 4

Perform task-3 with an additional feature to be added that is NAT Gateway to provide the internet access to instances running in the private subnet.

Performing the following steps:


1. Write an Infrastructure as code using terraform, which automatically create a VPC.
2. In that VPC we have to create 2 subnets:
 1. public subnet [Accessible for Public World!]
 2. private subnet [Restricted for Public World!]
3. Create a public facing internet gateway for connect our VPC/Network to the internet world and attach this gateway to our VPC.
4. Create a routing table for Internet gateway so that instance can connect to outside world, update and associate it with public subnet.
5. Create a NAT gateway for connect our VPC/Network to the internet world and attach this gateway to our VPC in the public network
6. Update the routing table of the private subnet, so that to access the internet it uses the nat gateway created in the public subnet
7. Launch an ec2 instance which has Wordpress setup already having the security group allowing port 80 sothat our client can connect to our wordpress site. Also attach the key to instance for further login into it.
8. Launch an ec2 instance which has MYSQL setup already with security group allowing port 3306 in private subnet so that our wordpress vm can connect with the same. Also attach the key with the same.

Note: Wordpress instance has to be part of public subnet so that our client can connect our site.

mysql instance has to be part of private subnet so that outside world can't connect to it.

Addition of auto ip assign and auto dns name assignment option to be enabled.

Command prompt :-

 Command Prompt

```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>CD DESKTOP

C:\Users\Asus\Desktop>CD terraformcodefiles

C:\Users\Asus\Desktop\terraformcodefiles>notepad task-4.tf

C:\Users\Asus\Desktop\terraformcodefiles>
```

```
Command Prompt

C:\Users\Asus\Desktop\terraformcodefiles>terraform init

Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.70"

Warning: Interpolation-only expressions are deprecated

  on task3.tf line 15, in resource "aws_subnet" "public":
  15:   vpc_id      = "${aws_vpc.myvpc.id}"

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 6 more similar warnings elsewhere)

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.
```

```
Command Prompt

Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd desktop

C:\Users\Asus\Desktop>cd terraformcodefiles

C:\Users\Asus\Desktop\terraformcodefiles>notepad task-4.tf

C:\Users\Asus\Desktop\terraformcodefiles>terraform init

Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.70"

Warning: Interpolation-only expressions are deprecated

  on task3.tf line 15, in resource "aws_subnet" "public":
  15:   vpc_id      = "${aws_vpc.myvpc.id}"

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 6 more similar warnings elsewhere)

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,
```

```
*task-4.tf - Notepad

File Edit Format View Help

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

resource "aws_vpc" "myvpc" {
  cidr_block      = "192.168.0.0/16"
  instance_tenancy = "default"
  enable_dns_hostnames = true

  tags = {
    Name = "pratik_vpc"
  }
}

resource "aws_subnet" "public" {
  vpc_id      = "${aws_vpc.myvpc.id}"
  cidr_block = "192.168.0.0/24"
  availability_zone = "ap-south-1a"

  tags = {
    Name = "pratikpublicsubnet"
  }
}

resource "aws_subnet" "private" {
  vpc_id      = "${aws_vpc.myvpc.id}"
  cidr_block = "192.168.1.0/24"
  availability_zone = "ap-south-1b"

  tags = {
    Name = "pratikprivatesubnet"
  }
}

resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.myvpc.id}"

  tags = {
    Name = "pratikIG"
  }
}
```



```

C:\Users\Asus\desktop\terraformcodefiles>terraform apply --auto-approve
aws_eip.nat: Creating...
aws_vpc.myvpc: Creating...
aws_eip.nat: Creation complete after 1s [id=eipalloc-028a742f22363ebfb]
aws_vpc.myvpc: Creation complete after 7s [id=vpc-0a79d675648ea2d6b]
aws_internet_gateway.gw: Creating...
aws_subnet.private: Creating...
aws_subnet.public: Creating...
aws_security_group.webserver: Creating...
aws_subnet.private: Creation complete after 3s [id=subnet-0af2a448d8b9e4da5]
aws_subnet.public: Creation complete after 3s [id=subnet-0af2a448d8b9e4da5]
aws_internet_gateway.gw: Creation complete after 4s [id=igw-0137bfbddd3fa50fd]
aws_nat_gateway.nat-gw: Creating...
aws_route_table.forig: Creating...
aws_security_group.webserver: Creation complete after 6s [id=sg-0722d0c2b42ab7ae1]
aws_security_group.database: Creating...
aws_instance.wordpress: Creating...
aws_route_table.forig: Creation complete after 3s [id=rtb-0a8164f5a6b26c707]
aws_route_table_association.asstopublic: Creating...
aws_route_table_association.asstopublic: Creation complete after 0s [id=rtbassoc-08d1f90e974589031]
aws_security_group.database: Creation complete after 6s [id=sg-0f1ea0acaebc9dec6]
aws_instance.mysql: Creating...
aws_nat_gateway.nat-gw: Still creating... [10s elapsed]
aws_instance.wordpress: Still creating... [10s elapsed]
aws_instance.mysql: Still creating... [10s elapsed]
aws_nat_gateway.nat-gw: Still creating... [20s elapsed]
aws_instance.wordpress: Still creating... [20s elapsed]
aws_instance.mysql: Still creating... [20s elapsed]
aws_instance.wordpress: Creation complete after 26s [id=i-0206404abd8804ecb]
aws_nat_gateway.nat-gw: Still creating... [30s elapsed]
aws_instance.mysql: Creation complete after 26s [id=i-00ecf790bfd60ef58]
aws_nat_gateway.nat-gw: Still creating... [40s elapsed]

```

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 10 more similar warnings elsewhere)

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.

1.login in aws and create a vpc

```

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

resource "aws_vpc" "myvpc" {
  cidr_block     = "192.168.0.0/16"
  instance_tenancy = "default"
  enable_dns_hostnames = true

```

```
tags = {
  Name = "pratik_vpc"
}
```

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (Pratik, Mumbai, Support). The left sidebar shows the 'VIRTUAL PRIVATE CLOUD' section with 'Your VPCs' selected. The main content area displays the details for the VPC 'pratik_vpc' (ID: vpc-0a79d675648ea2d6b). The VPC is in an 'available' state with an IPv4 CIDR of 192.168.0.0/16. Below the table, the 'Description' tab is active, showing various attributes like VPC ID, State, IPv4 CIDR, IPv6 Pool, Network ACL, DHCP options set, Owner, Tenancy, Default VPC, IPv6 CIDR, DNS resolution, DNS hostnames, and Route table.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL
pratik_vpc	vpc-0a79d675648ea2d6b	available	192.168.0.0/16	-	dopt-0490a668700b92dcb	rtb-05b1e4d44ef116204	acl-07a43e1d72f1b66e3

VPC: vpc-0a79d675648ea2d6b	
Description	<p>VPC ID: vpc-0a79d675648ea2d6b</p> <p>State: available</p> <p>IPv4 CIDR: 192.168.0.0/16</p> <p>IPv6 Pool: -</p> <p>Network ACL: acl-07a43e1d72f1b66e3</p> <p>DHCP options set: dopt-0490a668700b92dcb</p> <p>Owner: 810445783252</p> <p>Tenancy: default</p> <p>Default VPC: No</p> <p>IPv6 CIDR: -</p> <p>DNS resolution: Enabled</p> <p>DNS hostnames: Disabled</p> <p>Route table: rtb-05b1e4d44ef116204</p>

2. creating two subnet 1 and has auto-launch ip

```
resource "aws_subnet" "public" {
  vpc_id   = "${aws_vpc.myvpc.id}"
  cidr_block = "192.168.0.0/24"
  availability_zone = "ap-south-1a"
```

```
tags = {
  Name = "pratikpublicsubnet"
}
```

```
resource "aws_subnet" "private" {
  vpc_id   = "${aws_vpc.myvpc.id}"
  cidr_block = "192.168.1.0/24"
  availability_zone = "ap-south-1b"
```

```
tags = {
  Name = "pratikprivatesubnet"
}
```

New VPC Experience

Tell us what you think

Create subnet

Actions



VPC Dashboard

Filter by VPC:

Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Managed Prefix

Filter by tags and attributes or search by keyword

1 to 2 of 2

	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
<input type="checkbox"/>	pratikprivat...	subnet-03a4c91adbdc9faae	available	vpc-0a79d675648ea2d6b ...	192.168.1.0/24	250	-
<input checked="" type="checkbox"/>	pratikpublic...	subnet-0af2a448d8b9e4da5	available	vpc-0a79d675648ea2d6b ...	192.168.0.0/24	250	-

VPC	vpc-0a79d675648ea2d6b pratik_vpc	IPv4 CIDR	192.168.0.0/24
Available IPv4 Addresses	250	IPv6 CIDR	-
Availability Zone	ap-south-1a (aps1-az1)	Route Table	rtb-0a8164f5a6b26c707 IGroutetable
Network ACL	acl-07a43e1d72f1b66e3	Default subnet	No
Auto-assign public IPv4 address	Yes	Auto-assign IPv6 address	No

Subnets > Modify auto-assign IP settings

Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.

Subnet ID subnet-0af2a448d8b9e4da5

Auto-assign IPv4 ☒ Enable auto-assign public IPv4 address

* Required

Cancel

Save

Subnets

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
pratikprivat...	subnet-03a4c91adbdc9faae	available	vpc-0a79d675648ea2d6b ...	192.168.1.0/24	250	-
pratikpublic...	subnet-0af2a448d8b9e4da5	available	vpc-0a79d675648ea2d6b ...	192.168.0.0/24	250	-

Description

Subnet ID: subnet-03a4c91adbdc9faae
 VPC: vpc-0a79d675648ea2d6b | pratik_vpc
 Available IPv4 Addresses: 250
 Availability Zone: ap-south-1b (aps1-az3)
 State: available
 IPv4 CIDR: 192.168.1.0/24
 IPv6 CIDR: -
 Route Table: rtb-05b1e4d44ef116204

3. creating an internet gateway for a subnet id in south-1a

```
resource "aws_internet_gateway" "gw" {
  vpc_id = "${aws_vpc.myvpc.id}"

  tags = {
    Name = "pratikIG"
  }
}
```

Internet gateway igw-0137bfbddd3fa50fd successfully attached to vpc-0a79d675648ea2d6b

igw-0137bfbddd3fa50fd / pratikIG

Details

Internet gateway ID: igw-0137bfbddd3fa50fd
 State: Attached

aws Services Resource Groups

New VPC Experience
Tell us what you think

VPC Dashboard **New**

Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways **New****
- Egress Only Internet Gateways **New**
- DHCP Options Sets **New**
- Elastic IPs **New**
- Managed Prefix Lists **New**

VPC > Internet gateways

Internet gateways (1/1) Info

Filter internet gateways

<input checked="" type="checkbox"/>	Name	Internet gateway ID	State	VPC ID
<input checked="" type="checkbox"/>	pratikIG	igw-0137bfbddd3fa50fd	Attached	vpc-0a79d675648ea2d6b

0137bfbddd3fa50fd | pratik_vpc

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

4. creating a route-table > associating route-table with the internet gateway

```
resource "aws_route_table" "forig" {
  vpc_id = "${aws_vpc.myvpc.id}"

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = "${aws_internet_gateway.gw.id}"
  }

  tags = {
    Name = "IGroutetable"
  }
}
```

aws Services Resource Groups

New VPC Experience
Tell us what you think

VPC Dashboard **New**

Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD

- Your VPCs
- Subnets
- Route Tables**
- Internet Gateways **New**
- Egress Only Internet Gateways **New**
- DHCP Options Sets **New**
- Elastic IPs **New**
- Managed Prefix Lists **New**

Create route table Actions

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
<input type="checkbox"/>		rtb-05b1e4d44ef116204	-	-	Yes	vpc-0a79d675648ea2d6b
<input type="checkbox"/>	IGroutetable	rtb-0a8164f5a6b26c707	subnet-0af2a448d8b9e4da5	-	No	vpc-0a79d675648ea2d6b
<input checked="" type="checkbox"/>	for_Database	rtb-0fb65b74c0bf3a899	subnet-03a4c91adbdc9faae	-	No	vpc-0a79d675648ea2d6b

Route Table: rtb-0fb65b74c0bf3a899

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Route Table ID	rtb-0fb65b74c0bf3a899	Main	No
Explicitly Associated with	subnet-03a4c91adbdc9faae	VPC	vpc-0a79d675648ea2d6b pratik_vpc

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

5. Associating route table with subnet

```
resource "aws_route_table_association" "asstopublic" {
  subnet_id    = aws_subnet.public.id
  route_table_id = aws_route_table.forig.id
}
```

The screenshot shows the AWS Management Console interface for the 'Route Tables' section. The left sidebar contains navigation links for VPC Dashboard, Virtual Private Cloud, Your VPCs, Subnets, Route Tables (selected), Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, and Endpoint Services. The main content area displays a list of route tables. The 'for_Database' route table is selected, showing its details. Below the list, the 'Routes' tab is active, displaying a table of routes for the selected route table.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
	rtb-05b1e4d44ef116204	-	-	Yes	vpc-0a79d675648ea2d6b ...
IGroutetable	rtb-0a8164f5a6b26c707	subnet-0af2a448d8b9e4da5	-	No	vpc-0a79d675648ea2d6b ...
for_Database	rtb-0fb65b74c0bf3a899	subnet-03a4c91adbdc9faae	-	No	vpc-0a79d675648ea2d6b ...

Route Table: rtb-0fb65b74c0bf3a899

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View: All routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	active	No
0.0.0.0/0	nat-0a8b8161fbd29e5f3	active	No

6. Creating the security group with ingress(ssh,http and icmpv4 protocol) - mywebserver_sg

```
resource "aws_security_group" "webserver" {
  name      = "for_wordpress"
  description = "Allow http,ssh"
  vpc_id    = "${aws_vpc.myvpc.id}"
}
```

```
ingress {
  description = "HTTP"
  from_port   = 80
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}
```

```
ingress {
  description = "SSH"
  from_port   = 22
  to_port     = 22
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}
```

```
egress {
  from_port = 0
}
```

```
to_port    = 0
protocol   = "-1"
cidr_blocks = ["0.0.0.0/0"]
}

tags = {
  Name = "mywebserver_sg"
}
```

aws

Services

Resource Groups

Pratik

Mumbai

Support

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Inbound security group rules successfully modified on security group (sg-0722d0c2b42ab7ae1 | for_wordpress)

Details

EC2

Security Groups

Security Groups (1/3)

Info

Filter security groups

1

Create security group

	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-01ccdb25a34428c28	default	vpc-0a79d675648ea2d6b	default VPC security
<input checked="" type="checkbox"/>	mywebserver_sg	sg-0722d0c2b42ab7ae1	for_wordpress	vpc-0a79d675648ea2d6b	Allow http,ssh
<input type="checkbox"/>	mydatabase_sg	sg-0f1ea0acaebc9dec6	for_MYSQL	vpc-0a79d675648ea2d6b	Allow ssh and MYSQL

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	::/0	-
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-
All ICMP - IPv4	ICMP	All	0.0.0.0/0	-

Feedback English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws

Services

Resource Groups

Pratik

Mumbai

Support

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

EC2

Security Groups

Security Groups (1/5)

Info

Filter security groups

1

Create security group

Feedback English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

6.1 Launching NAT Gateway:-

```
resource "aws_eip" "nat" {
  vpc=true
}

resource "aws_nat_gateway" "nat-gw" {
  allocation_id = "${aws_eip.nat.id}"
  subnet_id     = "${aws_subnet.public.id}"
  depends_on = [aws_internet_gateway.gw]

  tags = {
    Name = "GateWay NAT"
  }
}

resource "aws_route_table" "forprivate" {
  vpc_id = "${aws_vpc.myvpc.id}"

  route {
    cidr_block = "0.0.0.0/0"
    nat_gateway_id = "${aws_nat_gateway.nat-gw.id}"
  }
}
```

```
tags = {
  Name = "for_Database"
}

resource "aws_route_table_association" "nat" {
  subnet_id      = aws_subnet.private.id
  route_table_id = aws_route_table.forprivate.id
}
```

The screenshot displays the AWS Management Console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (Pratik, Mumbai, Support). The left-hand navigation pane shows categories like 'VIRTUAL PRIVATE CLOUD' and 'SECURITY', with 'NAT Gateways' highlighted under VPC. The main content area features a 'Create NAT Gateway' button and a table listing existing NAT Gateways. One gateway, 'GateWay NAT', is shown with an 'available' status. Below the table, the 'Details' tab for the selected NAT Gateway is active, displaying its ID, status, Elastic IP Address, Private IP Address, Network Interface ID, and Subnet.

Name	NAT Gateway ID	Status	Status Message	Elastic IP Address	Private IP Address	Network Interface ID	VPC
GateWay NAT	nat-0a8b8161bd29e5f3	available	-	3.7.7.49	192.168.0.130	eni-08642d17692b1c46e	vpc-0a79d675

Property	Value
NAT Gateway ID	nat-0a8b8161bd29e5f3
Status	available
Status Message	-
Elastic IP Address	3.7.7.49
Private IP Address	192.168.0.130
Network Interface ID	eni-08642d17692b1c46e
Subnet	subnet-0af2a448d8b9e4da5
VPC	vpc-0a79d675648ea2d6b pratik_vpc

aws

Services

Resource Groups

Pratik

Mumbai

Support

New VPC Experience

Tell us what you think

VPC Dashboard

New

Filter by VPC:

Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

New

Egress Only Internet Gateways

New

DHCP Options Sets

New

Elastic IPs

New

Managed Prefix Lists

New

Create route table

Actions

Filter by tags and attributes or search by keyword

1 to 3 of 3

	Name	Route Table ID	Explicit subnet associatio	Edge associations	Main	VPC ID
<input type="checkbox"/>		rtb-05b1e4d44ef116204	-	-	Yes	vpc-0a79c
<input type="checkbox"/>	IGroutetable	rtb-0a8164f5a6b26c707	subnet-0af2a448d8b9e4da5	-	No	vpc-0a79c
<input checked="" type="checkbox"/>	for_Database	rtb-0fb65b74c0bf3a899	subnet-03a4c91adbdc9faae	-	No	vpc-0a79c

Route Table: rtb-0fb65b74c0bf3a899

Summary

Routes

Subnet Associations

Edge Associations

Route Propagation

Tags

Route Table ID

rtb-0fb65b74c0bf3a899

Main

No

Explicitly Associated with

subnet-03a4c91adbdc9faae

VPC

vpc-0a79d675648ea2d6b | pratik_vpc

Feedback

English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

aws

Services

Resource Groups

Pratik

Mumbai

Support

New VPC Experience

Tell us what you think

VPC Dashboard

New

Filter by VPC:

Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

New

Egress Only Internet Gateways

New

DHCP Options Sets

New

Elastic IPs

New

Managed Prefix Lists

New

Endpoints

Endpoint Services

Create route table

Actions

Filter by tags and attributes or search by keyword

1 to 3 of 3

	Name	Route Table ID	Explicit subnet associatio	Edge associations	Main	VPC ID
<input type="checkbox"/>		rtb-05b1e4d44ef116204	-	-	Yes	vpc-0a79d675648ea2d6b ...
<input type="checkbox"/>	IGroutetable	rtb-0a8164f5a6b26c707	subnet-0af2a448d8b9e4da5	-	No	vpc-0a79d675648ea2d6b ...
<input checked="" type="checkbox"/>	for_Database	rtb-0fb65b74c0bf3a899	subnet-03a4c91adbdc9faae	-	No	vpc-0a79d675648ea2d6b ...

Route Table: rtb-0fb65b74c0bf3a899

Summary

Routes

Subnet Associations

Edge Associations

Route Propagation

Tags

Edit routes

View

All routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	active	No
0.0.0.0/0	nat-0a8b8161fbd29e5f3	active	No

Feedback

English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

The screenshot displays the AWS Management Console interface for Elastic IP addresses. The left-hand navigation pane includes sections for 'New VPC Experience', 'VIRTUAL PRIVATE CLOUD', and 'Elastic IPs'. The main content area is titled 'Elastic IP addresses (1/1)' and features a table with columns for Name, Public IPv4 address, Allocation ID, and Associated i. A single entry is shown with the IP address 3.7.7.49 and Allocation ID eipalloc-028a742f22363ebfb. Below the table, the 'Summary' tab is active, displaying the IP address 3.7.7.49. The footer of the console shows the AWS logo, navigation links, and copyright information.

7. Creating a subnet group with MYSQL protocol and value of security_id(myweb) - mydatabase_sg

```
resource "aws_security_group" "database" {
  name      = "for_MYSQL"
  description = "Allow ssh and MYSQL"
  vpc_id    = "${aws_vpc.myvpc.id}"

  ingress {
    description = "MYSQL"
    security_groups = [aws_security_group.webserver.id]
    from_port     = 3306
    to_port       = 3306
    protocol      = "tcp"
  }

  egress {
    from_port = 0
    to_port   = 0
    protocol  = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "mydatabase_sg"
  }
}
```

The screenshot displays the AWS Management Console interface for Security Groups. The left-hand navigation pane includes categories such as Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, Elastic Block Store, and Network & Security. The 'Security Groups' link under Network & Security is highlighted. The main panel shows the 'Security Groups (1/5)' page with a search bar and a table of existing security groups. The table has columns for Name, Security group ID, Security group name, and VPC ID. The group 'mydatabase_sg' with ID 'sg-0f1ea0acaebc9dec6' is selected. Below the table, the rules for this security group are detailed in a table with columns: Type, Protocol, Port range, Source, and Description - optional. A rule is listed for 'MYSQL/Aurora' using 'TCP' on port '3306' from the source 'sg-0f1ea0acaebc9dec6 (for_MYSQL)'.

8. Launching the instance

```
resource "aws_instance" "wordpress" {
  ami          = "ami-000cbce3e1b899ebd"
  instance_type = "t2.micro"
  associate_public_ip_address = true
  subnet_id = aws_subnet.public.id
  vpc_security_group_ids = [aws_security_group.webserver.id]
  key_name = "mykey111"
```

```
tags = {
  Name = "wordpress"
}
```

```
}

resource "aws_instance" "mysql" {
  ami          = "ami-0019ac6129392a0f2"
  instance_type = "t2.micro"
  subnet_id = aws_subnet.private.id
  vpc_security_group_ids = [aws_security_group.database.id]
  key_name = "mykey111"
```

```
tags = {
  Name = "mysql"
}
```

```
}
```

aws Services Resource Groups

New EC2 Experience

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

NAME	Name	App	Environment	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
<input checked="" type="checkbox"/>	wordpress			i-0206404abd8804e...	t2.micro	ap-south-1a	running	2/2 checks ...	None	ec2-13-235-95-78.ap-s...
<input type="checkbox"/>	mysql			i-00ecf790bfd60ef58	t2.micro	ap-south-1b	running	2/2 checks ...	None	
<input type="checkbox"/>				i-04c002a509590af03	t2.micro	ap-south-1a	terminated		None	
<input type="checkbox"/>				i-069c929c824d6134b	t2.micro	ap-south-1a	terminated		None	
<input type="checkbox"/>				i-0e4918db9e62c13...	t2.micro	ap-south-1b	terminated		None	

Instance: i-0206404abd8804ecb (wordpress) Public DNS: ec2-13-235-95-78.ap-south-1.compute.amazonaws.com

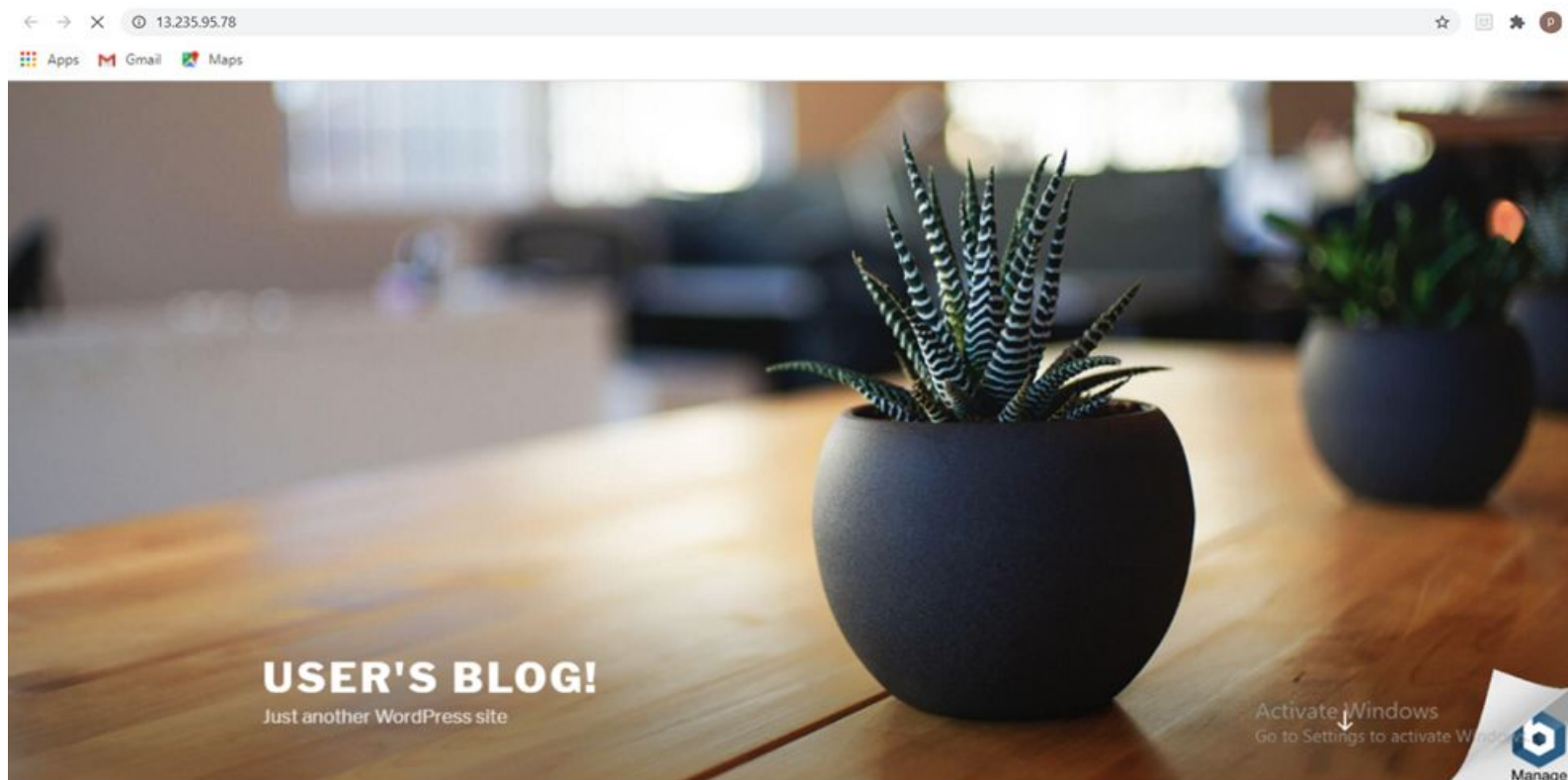
Description Status Checks Monitoring Tags

Instance ID: i-0206404abd8804ecb
 Instance state: running
 Instance type: t2.micro
 Finding: Opt-in to AWS Compute Optimizer for recommendations. Learn more

Public DNS (IPv4): ec2-13-235-95-78.ap-south-1.compute.amazonaws.com
 IPv4 Public IP: 13.235.95.78
 IPv6 IPs: -
 Elastic IPs: -

Feedback English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use



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 10 more similar warnings elsewhere)

Destroy complete! Resources: 14 destroyed.

THANKS FOR READING!!!