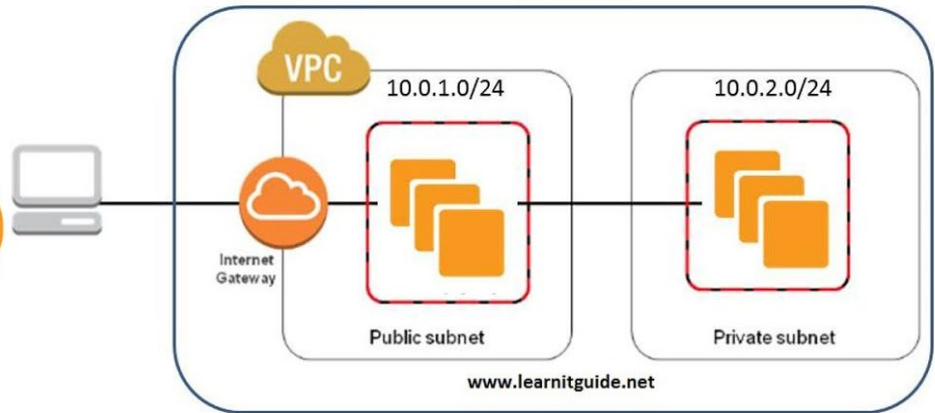


# AWS VPC

10.0.0.0/16



## HOW TO CREATE NEW VPC SUBNETS, INTERNET GATEWAY



**Terraform**



Amazon Machine Image (AMI)



Amazon EC2

# Task 3

Statement: We have to create a web portal for our company with all the security as much as possible.

So, we use Wordpress software with dedicated database server.

Database should not be accessible from the outside world for security purposes.

We only need to public the WordPress to clients.

So here are the steps for proper understanding!

Steps:-

- 1) Write a Infrastructure as code using terraform, which automatically create a VPC.
- 2) In that VPC we have to create 2 subnets:
  - a) public subnet [ Accessible for Public World! ]
  - b) 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) Launch an ec2 instance which has Wordpress setup already having the security group allowing port 80 so that our client can connect to our wordpress site.  
Also attach the key to instance for further login into it.
- 6) 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.

Don't forgot to add 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>terraformcodefiles
'terraformcodefiles' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Asus\Desktop>cd terraformcodefiles

C:\Users\Asus\Desktop\terraformcodefiles>notepad task3.tf
```

Command Prompt

1: provider "aws" {  
  
A default (non-aliased) provider configuration for "aws" was already given at  
ec2.tf:1,1-15. If multiple configurations are required, set the "alias"  
argument for alternative configurations.  
  
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.

\*task3.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"  
 }  
}

Type here to search

ENG 12:31 AM  
IN 14/07/2020

```
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.
```

```
Warning: Interpolation-only expressions are deprecated

   on task.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)

Apply complete! Resources: 10 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 for the VPC Dashboard. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (Pratik, Mumbai, Support). The left sidebar lists various services under 'VIRTUAL PRIVATE CLOUD' and 'SECURITY'. The main content area displays a table of VPCs with columns: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, Main Route table, and Main Network ACL. The 'pratik\_vpc' is highlighted. Below the table, the detailed view for VPC 'vpc-0a79d675648ea2d6b' is shown, including tabs for Description, CIDR Blocks, Flow Logs, and Tags. The 'Description' tab is active, showing details 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.

## 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"  
  }  
}
```



## 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

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 subnet Actions

Filter by tags and attributes or search by keyword

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 Flow Logs Route Table Network ACL Tags Sharing

Subnet ID subnet-03a4c91adbdc9faae State available

VPC vpc-0a79d675648ea2d6b | pratik\_vpc

Available IPv4 Addresses 250 IPv6 CIDR -

Availability Zone ap-south-1b (aps1-az3) Route Table rtb-05b1e4d44ef116204

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

### 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"
  }
}
```

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**

VPC > ... > igw-0137bfbddd3fa50fd

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

igw-0137bfbddd3fa50fd / pratikIG Actions

Details Info

Internet gateway ID  
igw-0137bfbddd3fa50fd

State  
Attached

Feedback English (US) Privacy Policy Terms of Use

The screenshot displays the AWS Management Console interface for the 'Internet gateways' section. The left-hand navigation pane includes options like 'New VPC Experience', 'VPC Dashboard', and 'VIRTUAL PRIVATE CLOUD'. The main area shows a list of internet gateways. A single gateway, 'pratikIG', is listed with an 'Attached' status. Below the list, a detailed view of the selected gateway shows its ID and the VPC it is associated with, 'pratik\_vpc'.

#### 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"
  }
}
```



**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
}
```

**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

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

Feedback

English (US)

Inbound security group rules successfully modified on security group (sg-0722d0c2b42ab7ae1 | for\_wordpress)

Details

EC2 > Security Groups

Security Groups (1/3)

Info

Actions

Create security group

Filter security groups

< 1 >

	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

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

-

© 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 Security Groups. The left-hand navigation pane includes categories like 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. At the top, there's a search bar labeled 'Filter security groups' and a 'Create security group' button. Below this is a table with columns: Name, Security group ID, Security group name, and VPC ID. One security group is listed: 'mywebserver\_sg' with ID 'sg-0722d0c2b42ab7ae1' and VPC ID 'vpc-0a79d675648ea2d6b'. Below the table, a detailed view of the security group rules is shown, including Type, Protocol, Port range, Source, and Description - optional. The rules listed are: HTTP/TCP on port 80 from 0.0.0.0/0, HTTP/TCP on port 80 from ::/0, SSH/TCP on port 22 from 0.0.0.0/0, and SSH/TCP on port 22 from ::/0.

## 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's 'Security Groups' page. The left-hand navigation pane includes categories like 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'Elastic Block Store', and 'Network & Security'. Under 'Network & Security', 'Security Groups' is highlighted. The main panel shows 'Security Groups (1/5)' with a search bar and a table of existing 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 group are listed in a table with columns: Type, Protocol, Port range, Source, and Description - optional. A rule is shown 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-000cbce3e1b8999ebd"
  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"
}
```

```
}
```





## USER'S BLOG!

Just another WordPress site

Activate Windows  
Go to Settings to activate Windows.



Manage

**Warning:** Interpolation-only expressions are deprecated

```
on task.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)

Destroy complete! Resources: 10 destroyed.