

TASK 3:

> login in aws

> create a vpc

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

```
resource "aws_vpc" "hw" {  
  cidr_block = "192.168.0.0/16"  
  instance_tenancy = "default"
```

```
  tags = {  
    Name = "hw"  
  }  
}
```

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'VPCs' page is active, displaying a table with one VPC: 'hw' with ID 'vpc-02973363e595f38c6', state 'available', and IPv4 CIDR '192.168.0.0/16'. The 'Details' tab for this VPC is selected, showing various attributes:

Attribute	Value
VPC ID	vpc-02973363e595f38c6
State	available
IPv4 CIDR	192.168.0.0/16
IPv6 Pool	-
Network ACL	acl-0196c343525d004ba
DHCP options set	dopt-57758d3c
Owner	410914255776
Tenancy	default
Default VPC	No
IPv6 CIDR	-
DNS resolution	Enabled
DNS hostnames	Disabled
Route table	rtb-0ed36d616f5f035d5

> creating two subnet 1 has auto-launch ip

```
resource "aws_subnet" "hw_subnet-1a" {  
  vpc_id = "${aws_vpc.hw.id}"  
  cidr_block = "192.168.0.0/24"
```

```
    availability_zone = "ap-south-1a"
    map_public_ip_on_launch = true
}
resource "aws_subnet" "hw_subnet-1b" {
    vpc_id    = "${aws_vpc.hw.id}"
    cidr_block = "192.168.1.0/24"
    availability_zone = "ap-south-1b"
}
```

multi hybrid x homework x 10th july x AWS: aws x AWS: aws x ssh - Conve x How to Inst x PHP 7.3.1 x Subnets | V x + -

ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#subnets:sort=AvailabilityZone

aws Services Resource Groups visheshgargavi Mumbai Support

New VPC Experience Tell us what you think

Filter by VPC: vpc-02973...

VPC Dashboard New

Filter by VPC: vpc-02973363e595f38c6 hw Owner: 410914255776

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways New

Egress Only Internet Gateways New

DHCP Options Sets New

Create subnet Actions

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IP	IPv6 CIDR	Av
	subnet-03c940b59379d112e	available	vpc-02973363e595f38c6 ...	192.168.0.0/24	249	-	ap-
	subnet-09fdbb2e1248e3d56	available	vpc-02973363e595f38c6 ...	192.168.1.0/24	250	-	ap-

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

multi hybrid x homework x 10th july x AWS: aws x AWS: aws x ssh - Conve x How to Inst x PHP 7.3.1 x Subnets | V x + -

ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#subnets:sort=AvailabilityZone

aws Services Resource Groups visheshgargavi Mumbai Support

New VPC Experience Tell us what you think

Filter by VPC: vpc-02973...

VPC Dashboard New

Filter by VPC: vpc-02973363e595f38c6 hw Owner: 410914255776

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways New

Egress Only Internet Gateways New

DHCP Options Sets New

Create subnet Actions

Filter by tags and attributes or search by keyword

VPC	IPv4 CIDR	Available IP	IPv6 CIDR	Availability Zone	Availability Zone ID	Route table
c-02973363e595f38c6 ...	192.168.0.0/24	249	-	ap-south-1a	aps1-az1	rtb-01babc1ec000de765 hw_r
c-02973363e595f38c6 ...	192.168.1.0/24	250	-	ap-south-1b	aps1-az3	rtb-0ed36d616f5f035d5

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

multi hybrid X homework X 10th july - X AWS: aws_ X AWS: aws_ X ssh - Conve X How to Inst X PHP 7.3.1 X Modify auti X

ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#ModifyAutoAssignIpSettings:SubnetId=subnet-03c940b59379d112e

Apps New Tab Search Inbox (157) - 2019p... Inbox (1,144) - vish... Inbox (252) - 2019p... Gmail YouTube Maps 29th APRIL - Googl... YAMLLint - The YAM...

aws Services Resource Groups visheshgargavi Mumbai Support

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-03c940b59379d112e

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

* Required Cancel Save

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

Windows taskbar with search bar, taskbar icons, and system tray showing language (ENG IN) and time (1:45 AM 7/11/2020).

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Subnets' page is active, displaying a list of subnets. The first subnet, 'subnet-03c940b59379d112e', is selected. The details for this subnet are shown below the list:

Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Available IPv6
subnet-03c940b59379d112e	available	vpc-02973363e595f38c6 hw	192.168.0.0/24	249	-	-
subnet-09fdbb2e1248e3d56	available	vpc-02973363e595f38c6 hw	192.168.1.0/24	250	-	-

Below the table, the details for the selected subnet are shown:

Subnet ID	State	IPv4 CIDR	IPv6 CIDR	Route Table	Default subnet	Auto-assign IPv6 address	Owner
subnet-03c940b59379d112e	available	192.168.0.0/24	-	rtb-01babc1ec000de765 hw_route_table	No	No	410914255776

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Subnets' page is active, displaying a list of subnets. The second subnet, 'subnet-09fdbb2e1248e3d56', is selected. The details for this subnet are shown below the list:

Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Available IPv6
subnet-03c940b59379d112e	available	vpc-02973363e595f38c6 hw	192.168.0.0/24	249	-	-
subnet-09fdbb2e1248e3d56	available	vpc-02973363e595f38c6 hw	192.168.1.0/24	250	-	-

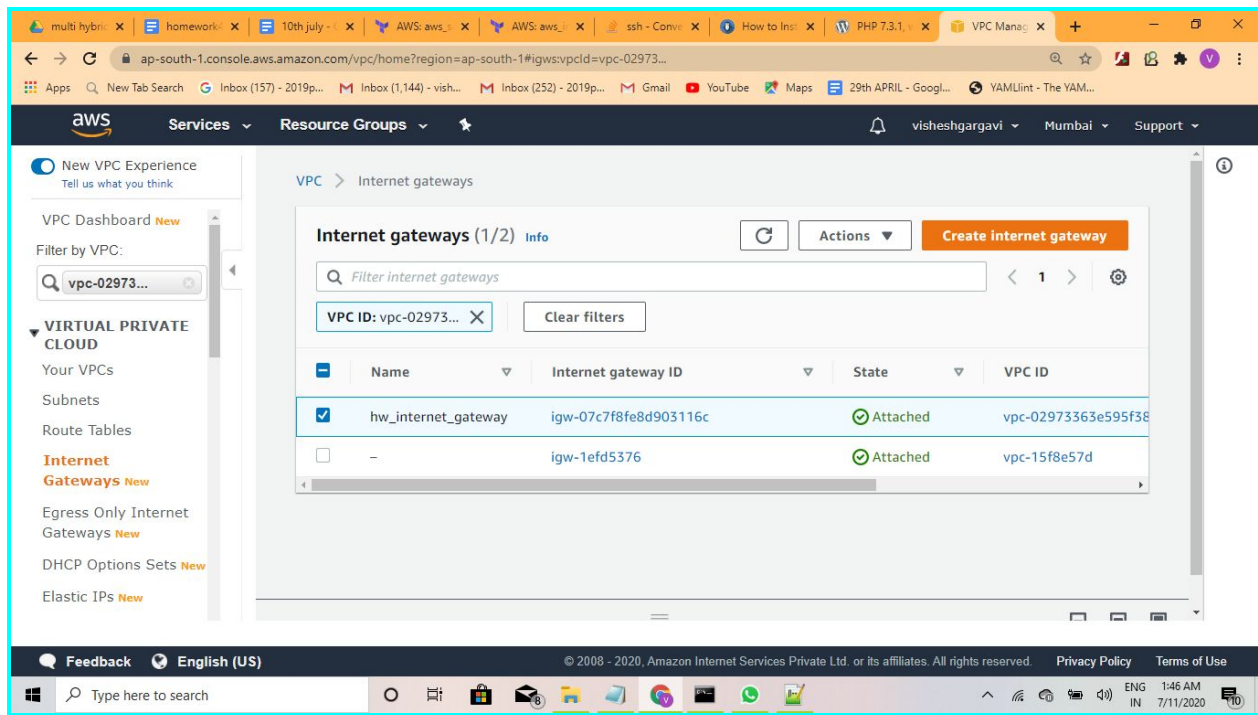
Below the table, the details for the selected subnet are shown:

Subnet ID	State	IPv4 CIDR	IPv6 CIDR	Route Table	Default subnet	Auto-assign IPv6 address	Owner
subnet-09fdbb2e1248e3d56	available	192.168.1.0/24	-	rtb-0ed36d616f5f035d5	No	No	410914255776

> creating an internet gateway for a subnet id in 1a

```
resource "aws_internet_gateway" "hw_internet_gateway" {
  vpc_id = "${aws_vpc.hw.id}"
```

```
tags = {
  Name = "hw_internet_gateway"
```



> creating a route-table

> associating route-table with the internet gateway

```
resource "aws_route_table" "hw_route_table" {
```

```
  vpc_id = "${aws_vpc.hw.id}"
```

```
  route {
```

```
    cidr_block = "0.0.0.0/0"
```

```
    gateway_id = "${aws_internet_gateway.hw_internet_gateway.id}"
```

```
  }
```

```
  tags = {
```

```
    Name = "hw_route_table"
```

```
  }
```

```
}
```


Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
hw_route_t...	rtb-01babc1ec000de765	subnet-03c940b59379d112e	-	No	vpc-029733
	rtb-0ed36d616f5f035d5	-	-	Yes	vpc-029733

Route Table: rtb-01babc1ec000de765

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Route Table ID: rtb-01babc1ec000de765
Main: No
Explicitly Associated with: subnet-03c940b59379d112e
VPC: vpc-02973363e595f38c6 | hw
Owner: 410914255776

Filter by tags and attributes or search by keyword

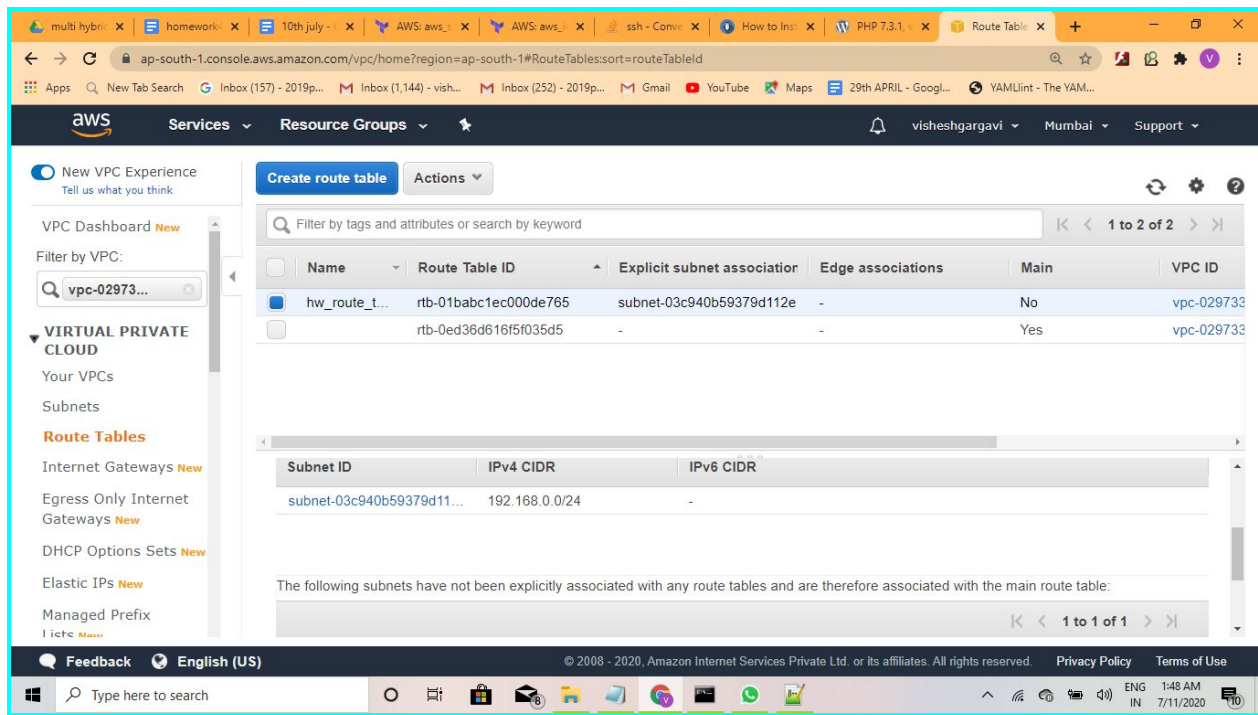
Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
hw_route_t...	rtb-01babc1ec000de765	subnet-03c940b59379d112e	-	No	vpc-029733
	rtb-0ed36d616f5f035d5	-	-	Yes	vpc-029733

View: All routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	active	No
0.0.0.0/0	igw-07c7f8fe8d903116c	active	No

> associating route table with subnet

```
resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.hw_subnet-1a.id
  route_table_id = "${aws_route_table.hw_route_table.id}"
}
```



> creating the security group with ingress(ssh,http and icmpv4 protocol)

> myweb

```
resource "aws_security_group" "myweb" {
  name      = "myweb"
  description = "Allow ssh http and icmp"
  vpc_id    = "${aws_vpc.hw.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"]
  }
```

```
  ingress {
```



```
    description = "ICMP-IPv4"  
    from_port   = 0  
    to_port     = 0  
    protocol    = "-1"  
    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 = "myweb"  
  }  
}
```

multi hybrid x homework x 10th july x AWS: aws x AWS: aws x ssh - Conve x How to Inst x PHP 7.3.1 x VPC Mana x

ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#SecurityGroups:vpc-id=vpc-02973363e595f38c6

aws Services Resource Groups visheshgargavi Mumbai Support

New VPC Experience Tell us what you think

VPC Dashboard New

Filter by VPC:

Q vpc-02973...

vpc-02973363e595f38c6
hw
Owner: 410914255776

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways New

Egress Only Internet Gateways New

DHCP Options Sets New

Security Groups (5) Info

Filter security groups

VPC ID: vpc-02973363e595f38c6 X Clear filters

	Name	Security group ID	Security group name	VPC ID
<input type="checkbox"/>	myweb	sg-00403fe0ed551bd3c	myweb	vpc-02973363e595f38c6
<input type="checkbox"/>	mybastion	sg-087485e43647d1d62	mybastion	vpc-02973363e595f38c6
<input type="checkbox"/>	mysql	sg-0b51375c9bd222be6	mysql	vpc-02973363e595f38c6
<input type="checkbox"/>	mysqlallow	sg-0c0b663c8bf40207c	mysqlallow	vpc-02973363e595f38c6
<input type="checkbox"/>	-	sg-0d13995efbac315ee	default	vpc-02973363e595f38c6

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

multi hybrid x homework x 10th july x AWS: aws x AWS: aws x ssh - Conve x How to Inst x PHP 7.3.1 x VPC Mana x

ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#SecurityGroup:groupId=sg-00403fe0ed551bd3c

aws Services Resource Groups visheshgargavi Mumbai Support

New VPC Experience Tell us what you think

VPC Dashboard New

Filter by VPC:

Q vpc-02973...

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

VPC > Security Groups > sg-00403fe0ed551bd3c - myweb

sg-00403fe0ed551bd3c - myweb

Delete security group Copy to new security group

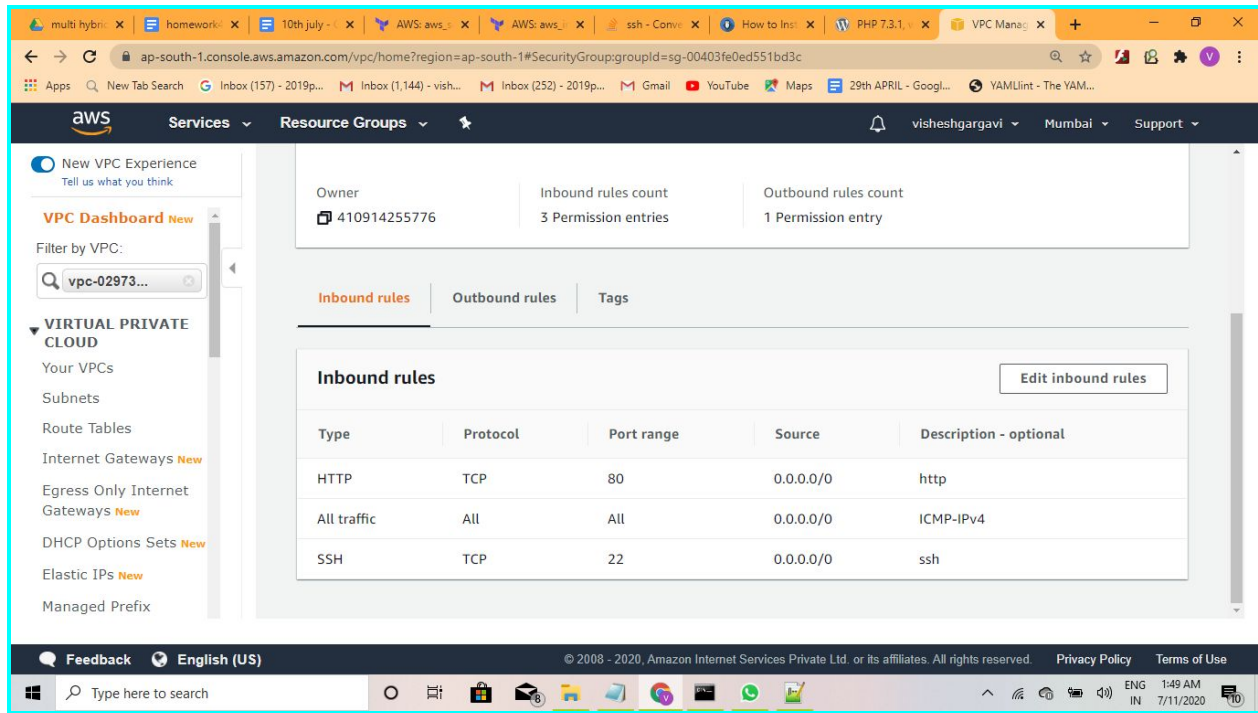
Details

Security group name	Security group ID	Description	VPC ID
myweb	sg-00403fe0ed551bd3c	Allow ssh http and icmp	vpc-02973363e595f38c6
Owner	Inbound rules count	Outbound rules count	
410914255776	3 Permission entries	1 Permission entry	

Inbound rules Outbound rules Tags

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



> creating a subnet group with MYSQL protocol and value of security_id(myweb)
 > mysql

```
resource "aws_security_group" "mysql" {
  name      = "mysql"
  description = "Allow sql"
  vpc_id    = "${aws_vpc.hw.id}"
```

```
  ingress {
    description = "MYSQL"
    security_groups = [ "${aws_security_group.myweb.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 = "mysql"

```
}  
}  
}
```

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile for 'visheshgargavi' in 'Mumbai'. The left sidebar contains a 'VPC Dashboard' with a search filter 'vpc-02973...' and a list of VPCs. The main content area displays the details for a security group named 'sg-0b51375c9bd222be6 - mysql'. The details table includes the following information:

Details			
Security group name mysql	Security group ID sg-0b51375c9bd222be6	Description Allow sql	VPC ID vpc-02973363e595f38c6
Owner 410914255776	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Below the details table, there are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. The 'Inbound rules' tab is currently selected.

This screenshot shows the 'Inbound rules' tab for the same security group. It displays a table of inbound rules with the following data:

Type	Protocol	Port range	Source	Description - optional
MYSQL/Aurora	TCP	3306	sg-00403fe0ed551bd3c (myweb)	MYSQL

An 'Edit inbound rules' button is located at the top right of the inbound rules table.

> creating a security group with ssh protocol

> bastion

```
resource "aws_security_group" "mybastion" {  
  name      = "mybastion"  
  description = "Allow ssh for bastion"  
  vpc_id    = "${aws_vpc.hw.id}"  
  
  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 = "mybastion"  
  }  
}
```


The image displays two screenshots of the AWS Management Console, specifically the 'Security Groups' page for a security group named 'sg-087485e43647d1d62 - mybastion' in the 'ap-south-1' region. The top screenshot shows the 'Details' tab, which includes fields for the security group name, ID, description, VPC ID, owner, and rule counts. The bottom screenshot shows the 'Inbound rules' tab, which contains a table of rules.

Details Tab:

Field	Value
Security group name	mybastion
Security group ID	sg-087485e43647d1d62
Description	Allow ssh for bastion
VPC ID	vpc-02973363e595f38c6
Owner	410914255776
Inbound rules count	1 Permission entry
Outbound rules count	1 Permission entry

Inbound rules Tab:

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	0	0.0.0.0/0	ssh

> creating a subnet group with ssh protocol and value as security_id(mybastion)
> mysqlallow

```
resource "aws_security_group" "mysqlallow" {  
  name = "mysqlallow"
```

```
description = "ssh allow to the mysql"
```

```
vpc_id      = "${aws_vpc.hw.id}"
```

```
ingress {
```

```
  description = "ssh"
```

```
  security_groups=[ "${aws_security_group.mybastion.id}" ]
```

```
  from_port = 22
```

```
  to_port   = 22
```

```
  protocol  = "tcp"
```

```
}
```

```
egress {
```

```
  from_port = 0
```

```
  to_port   = 0
```

```
  protocol  = "-1"
```

```
  cidr_blocks = ["0.0.0.0/0"]
```

```
}
```

```
tags = {
```

```
  Name = "mysqlallow"
```

```
}
```

```
}
```

The screenshot shows the AWS Management Console interface. The breadcrumb navigation indicates the path: VPC > Security Groups > sg-0c0b663c8bf40207c - mysqlallow. The main heading is 'sg-0c0b663c8bf40207c - mysqlallow'. Below this, there are buttons for 'Delete security group' and 'Copy to new security group'. The 'Details' section contains a table with the following information:

Security group name mysqlallow	Security group ID sg-0c0b663c8bf40207c	Description ssh allow to the mysql	VPC ID vpc-02973363e595f38c6
Owner 410914255776	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

At the bottom of the details section, there are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. The left sidebar shows the 'VIRTUAL PRIVATE CLOUD' section with options like 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways', and 'Egress Only Internet Gateways'.

This screenshot shows the 'Inbound rules' tab for the security group 'mysqlallow'. It displays a table with one rule:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	sg-087485e43647d1d62 (mybastion)	ssh

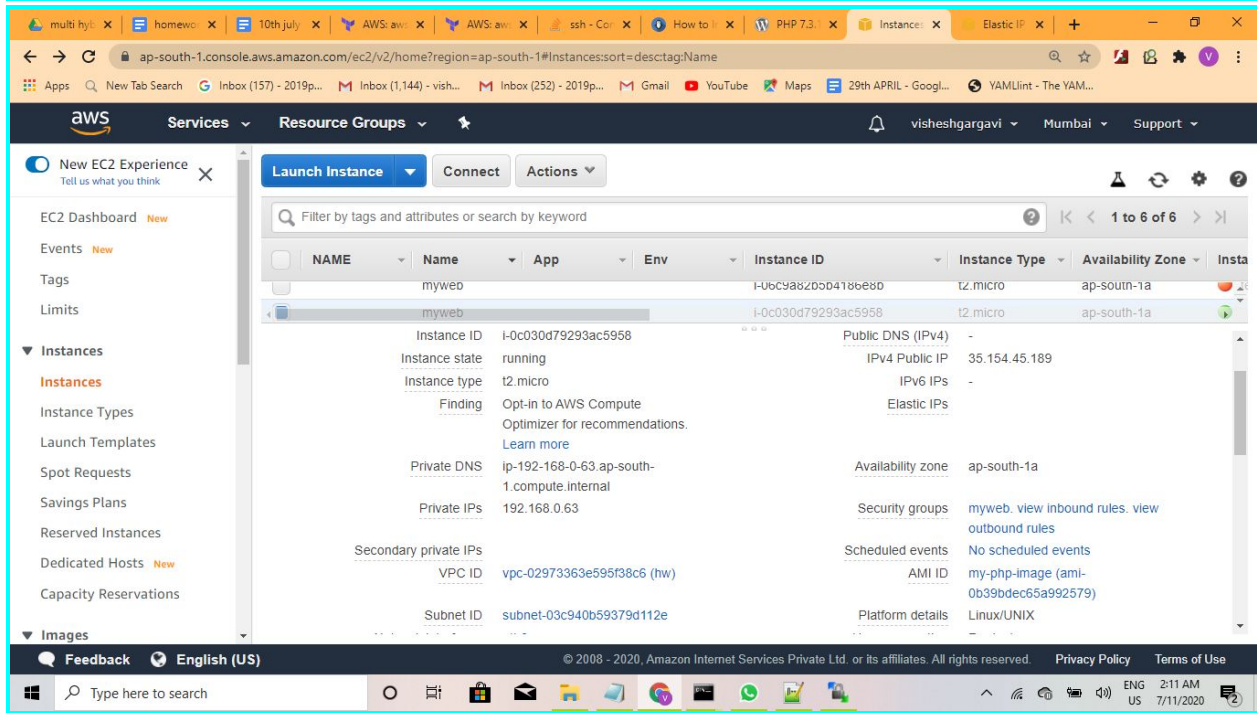
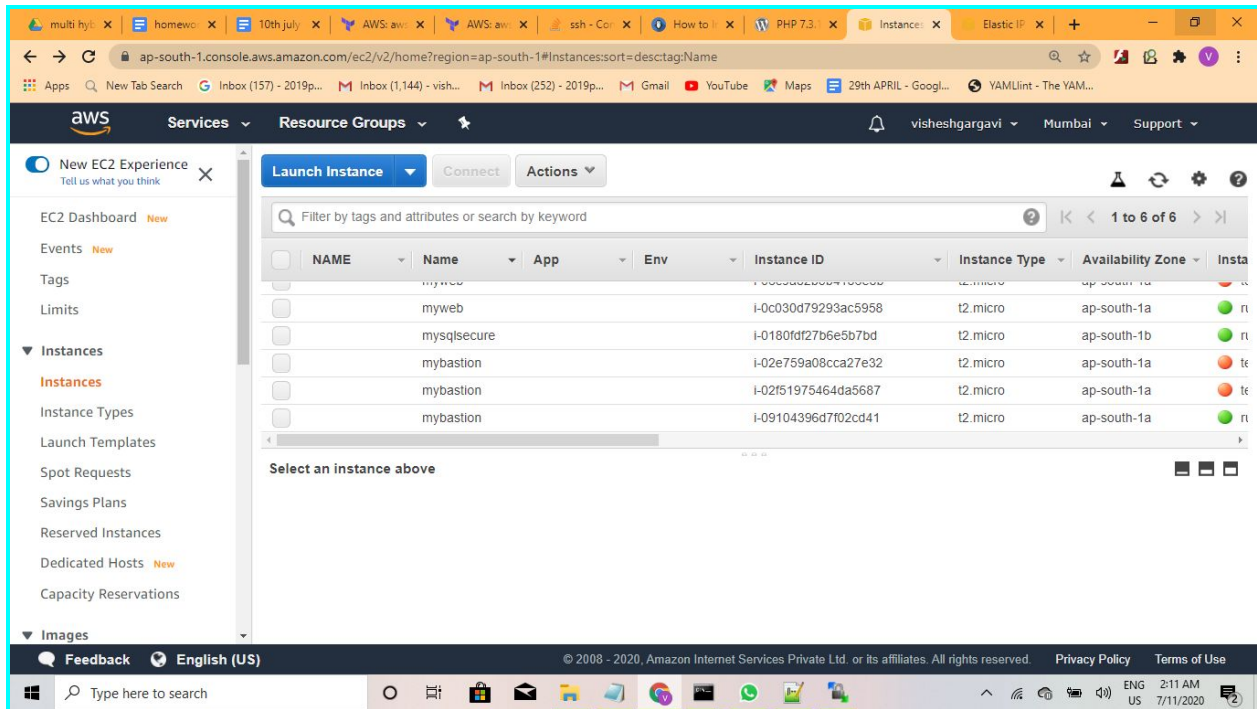
There is an 'Edit inbound rules' button to the right of the table. The interface is consistent with the previous screenshot, showing the same navigation and sidebar.

> launching the instance with the rhel image in the 1a region and attaching the security group myweb
> enabling the public-ip

resource "aws_instance" "mywordpress" {

```
ami      = "ami-052c08d70def0ac63"
instance_type = "t2.micro"
key_name  = "mykey1111.pem"
availability_zone = "ap-south-1a"
subnet_id = "${aws_subnet.hw_subnet-1a.id}"
security_groups = [ "${aws_security_group.myweb.id}" ]
user_data = <<-EOF
    #!/bin/bash
    sudo yum install dnf install php-mysqlnd php-fpm httpd tar curl php-json -y
    systemctl start httpd
    systemctl enable httpd
    curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
    tar xf wordpress.tar.gz
    cp -r wordpress /var/www/html
    chown -R apache:apache /var/www/html/wordpress
    chcon -t httpd_sys_rw_content_t /var/www/html/wordpress -R
EOF
```

```
tags = {
    Name = "mywordpress"
}
}
```



> **launching the instance with the rhel image in the region 1b and attaching the security group mysql and mysqlallow**
 > **not enabling the public-ip**

```
resource "aws_instance" "mysqlsecure" {
  ami = "ami-042c08d40def0ac62"
```



```

instance_type = "t2.micro"
key_name      = "mykey1111.pem"
availability_zone = "ap-south-1b"
subnet_id = "${aws_subnet.hw_subnet-1b.id}"
security_groups = [ "${aws_security_group.mysql.id}" ,
"${aws_security_group.mysqlallow.id}" ]
user_data = <<-EOF
    #!/bin/bash
    sudo yum install @mysql -y
    systemctl start mysql
    systemctl enable mysql

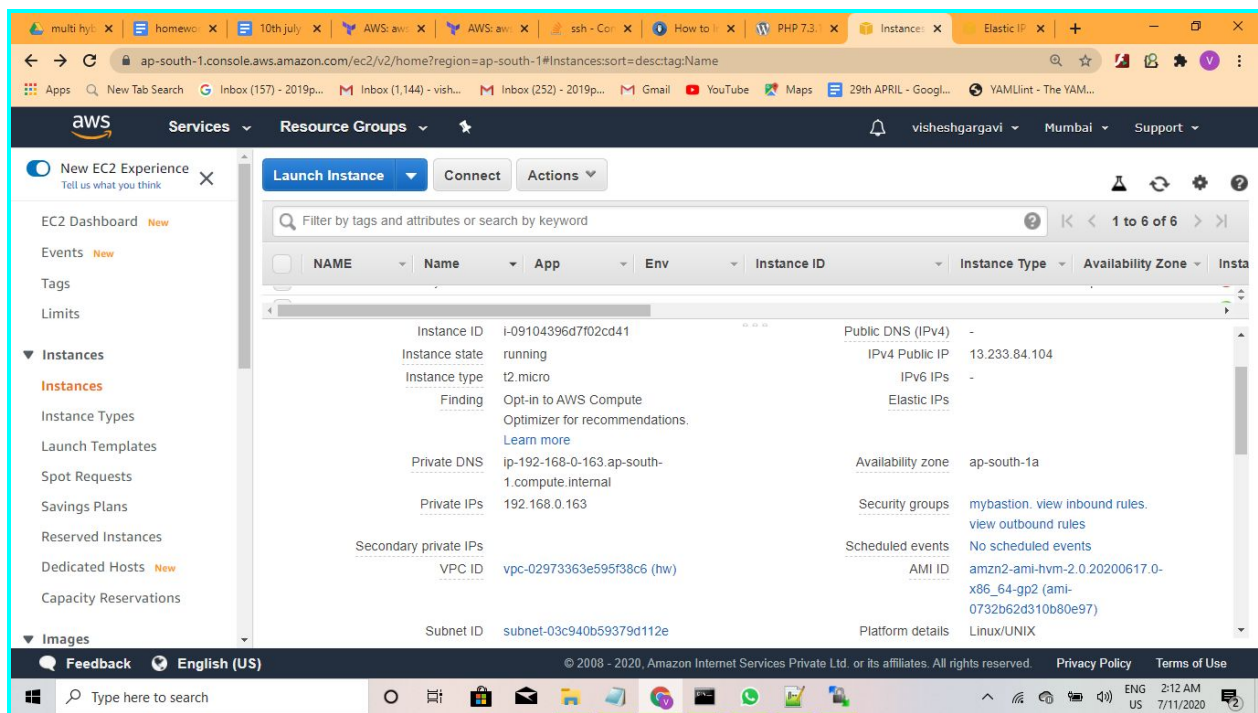
EOF

```

```

tags = {
    Name = "mysqlsecure"
}

```

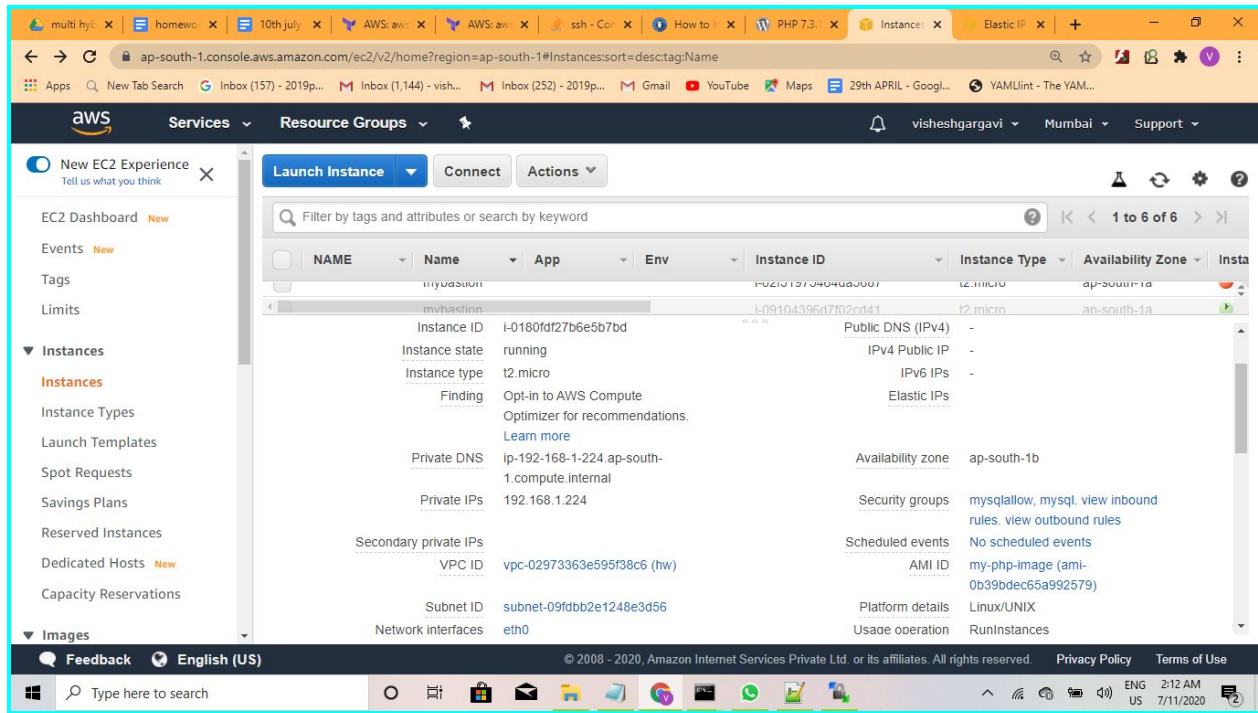


> launching the instance with the rhel image in the region 1a and attaching the security group mybastion

> enabling the public-ip

```
resource "aws_instance" "mybastion" {  
  ami           = "ami-0732b62d310b80e97"  
  instance_type = "t2.micro"  
  key_name      = "mykey1111.pem"  
  availability_zone = "ap-south-1a"  
  subnet_id     = "${aws_subnet.hw_subnet-1a.id}"  
  security_groups = [ "${aws_security_group.mybastion.id}" ]  
}
```

```
tags = {  
  Name = "mybastion"  
}
```



>> creating an elastic ip for allowing the NAT CONNECTIVITY

> creating a nat gateway and associating the nat_gateway with the elastic_ip

> associating the nat_gateway with the route table

```
resource "aws_eip" "hw_eip" {  
  vpc = true  
  
  instance = "${aws_instance.mysql.id}"
```

```

    associate_with_private_ip = "10.0.0.12"
    depends_on      = ["aws_internet_gateway.hw_internet_gateway"]
}

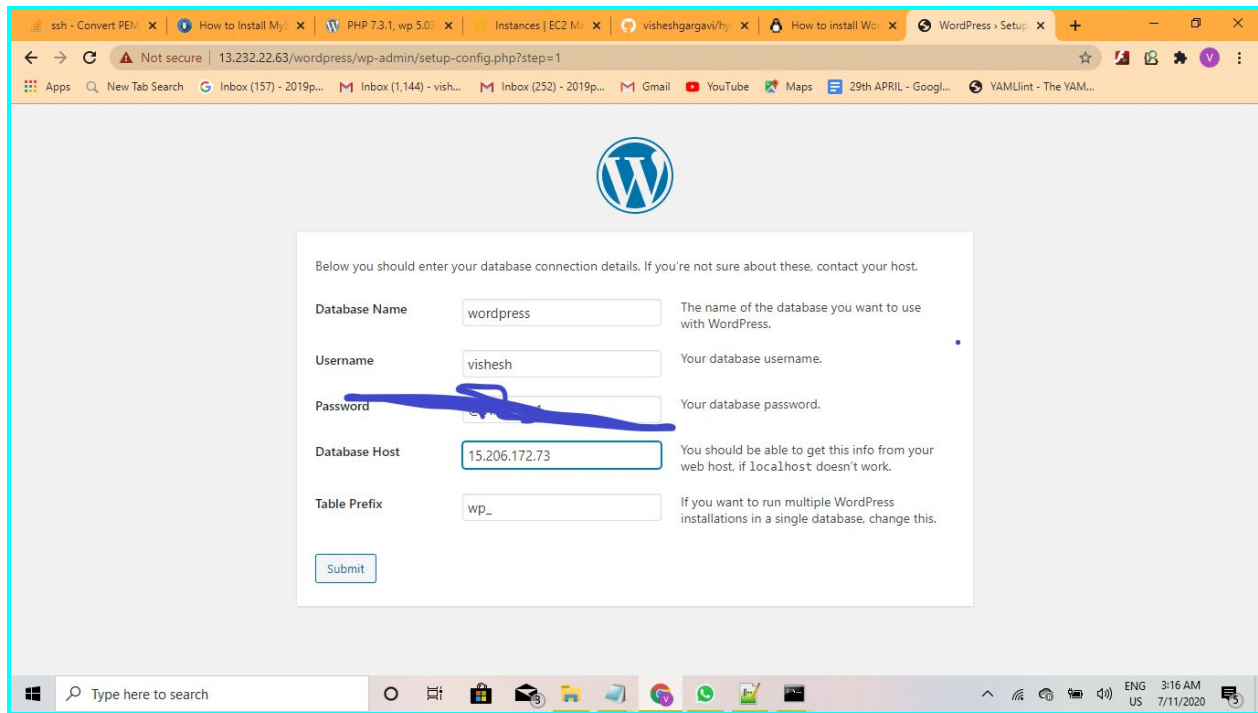
resource "aws_nat_gateway" "hw_nat_gateway" {
    allocation_id = "${aws_eip.hw_eip.id}"
    subnet_id     = "${aws_subnet.hw_subnet-1b.id}"

    tags = {
        Name = "hw_nat_gateway"
    }
}

resource "aws_route_table" "hw_route_table2" {
    vpc_id = "${aws_vpc.hw.id}"

    route {
        cidr_block = "0.0.0.0/0"
        nat_gateway_id = "${aws_nat_gateway.hw_nat_gateway.id}"
    }
}

```



Highly recommend to use pre-created image to avoid network disturbances

Run this few commands on the mysqlserver console

```
mysql_secure_installation
```

```
mysql -u root -p
```

```
mysql> CREATE DATABASE wordpress;
```

```
mysql> CREATE USER `admin`@`localhost` IDENTIFIED BY 'pass';
```

```
mysql> GRANT ALL ON wordpress.* TO `admin`@`localhost`;
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> exit
```