ICT & Infra S3 S/NO week 4: AWS VPC Sample web application

Date: Sep 2020
Version 1.0

Class: I3 CB01

Student numbers: 4642295, 4216709, 4961854

Student names: Ryan Smith, Edris Rahimi, Heiko Morales

Introduction

S/NO: In this exercises you will learn:

- how to create initial secure VPC web application design in AWS
- make a connection to S3 bucket from a public EC2 Instance
- study sample AWS Powershell scripts for this design and create similar Ansible/Teffaform scripts.

How to deliver your assignments?

Fill in this document with required information. Answer questions and upload the document to Canvas at most one week after the assignment is given.

Assignment 1: Create initial VPC web application setup

- Follow the demo from the class and create web-vpc and shared-vpc with corresponding EC2 instances.
- Demonstrate successful ssh-access to web-pub instance.
- Demonstrate successful “ping 8.8.8.8” from web-pub instance.
- Explain routing rules via “route” command at web-pub instance

We created a VPC and assigning tags to it. We then created a subnet for our VPC and configured the subnet settings. After that we created an internet gateway and attached it to our VPC. After that we used ssh to connect to an ec2 instance and pinged 8.8.8.8. The response was successful

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

- (•) VPC only
- ( ) VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

web-group14

IPv4 CIDR block Info

- (•) IPv4 CIDR manual input
- ( ) IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

IPv6 CIDR block Info

- (•) No IPv6 CIDR block
- ( ) IPAM-allocated IPv6 CIDR block
- ( ) Amazon-provided IPv6 CIDR block
- ( ) IPv6 CIDR owned by me

Tenancy Info

Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	web-group14	Remove

Add new tag

You can add 49 more tags.

Cancel Create VPC

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-00e0230a2642bf855 (web-group14)

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

web-subnet

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Europe (Frankfurt) / eu-central-1a

IPv4 CIDR block Info

0.0.0.0/24

▼ Tags - *optional*

Key	Value - *optional*	
Name	web-subnet	Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel Create subnet

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

IG-group14

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	IG-group14	Remove

Add new tag

You can add 49 more tags.

Cancel | Create internet gateway

Attach to VPC (igw-0348c337fb05abfe6) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

vpc-00e0230a2642bf855

▶ AWS Command Line Interface command

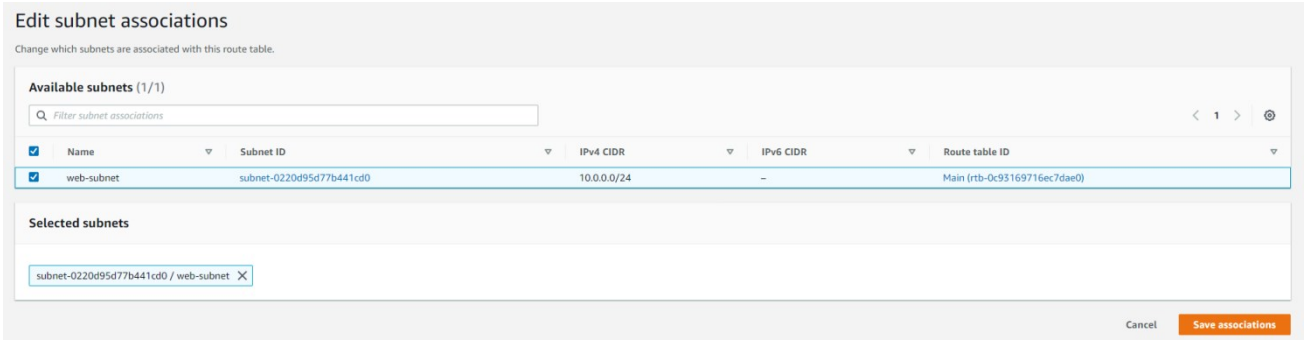
Cancel | Attach internet gateway

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	Active	No	
0.0.0.0/0	igw-0348c337fb05abfe6	Active	No	Remove

Add route

Cancel | Preview | Save changes




```
heiko@DESKTOP-IDN3683:~$ ssh -i .ssh/heiko_ubuntu_key.pem ubuntu@ec2-18-184-82-172.eu-central-1.compute.amazonaws.com
The authenticity of host 'ec2-18-184-82-172.eu-central-1.compute.amazonaws.com (18.184.82.172)' can't be established.
ECDSA key fingerprint is SHA256:2d4br+0sRv4XzomqfZKfV9nqPXpv+vNp1gZKiGI4C64.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-18-184-82-172.eu-central-1.compute.amazonaws.com,18.184.82.172' (ECDSA) to the list of k
nown hosts.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Oct  9 17:30:40 UTC 2022

  System load:  0.19              Processes:             101
  Usage of /:   19.6% of 7.57GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for eth0: 10.0.8.11
  Swap usage:   0%

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-8-11:~$
```

```
ubuntu@ip-10-0-8-11:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=1.25 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=2.17 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=1.40 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=1.31 ms
^C^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.251/1.533/2.171/0.371 ms
```

first row is the ip for the DHCP

second row is the subnet ip

```
ubuntu@ip-10-0-8-11:~$ ip route
default via 10.0.0.1 dev eth0 proto dhcp src 10.0.8.11 metric 100
10.0.0.0/20 dev eth0 proto kernel scope link src 10.0.8.11
10.0.0.1 dev eth0 proto dhcp scope link src 10.0.8.11 metric 100
```

Assignment 2: Make a connection from web-pub instance to s3 bucket

- Follow the guidelines: https://aws.amazon.com/premiumsupport/knowledge-center/ec2-instance-access-s3-bucket/

We created an s3 bucket and an ec2 instance. We connected to the instance by SSH and installed aws tools on the instance. We then went back to the bucket and applied the policies that would allow the ec2 instance to connect to the bucket. When that was done we went back to the instance and ran the command “aws s3 ls s3://heikoweb”

```
  },
  {
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::heikoweb/*"
  },
  {
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::heikoweb"
  }
]
```

```
ubuntu@ip-10-0-8-11:~$ aws s3 ls s3://heikoweb
                           PRE css/
                           PRE img/
2022-09-16 10:50:08        994 main.html
```

Assignment 3: Create Ansible/Terraform scripts for initial VPC web application setup

- Install latest Windows 7 Powershell Tool
- Unpack “PS_scripts.zip” archive
- Study and experiment with “PS_scripts.zip/Test” scripts. You don’t need to run these scripts, but you may use for inspiration.

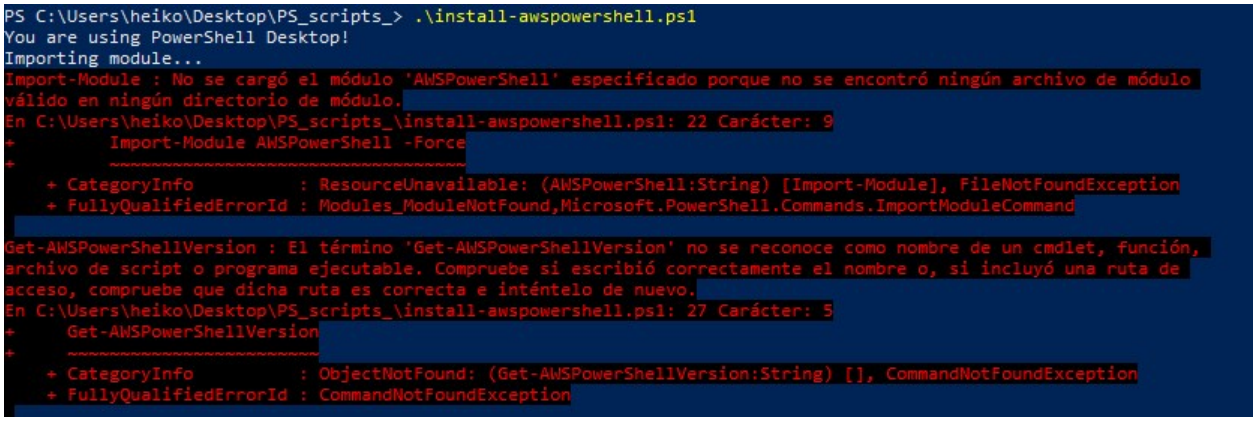
The main task is to compose the Ansible/Terraform scripts with all necessary configuration as in the Assignment 1 for public web-pub and private database

To run these sample scripts you may need to:

- Run “Install-awspowershel.ps1” to install AWS Powershell tools.
- Update “credentials.ps1”.
- Run “TEST/vpc-creation.ps1” to create sample web-pub instance with all necessary configurations in AWS.

On this task we ran into errors every time we tried to run the script. After an hour of troubleshooting we could not find the cause.

due to an unexpected error which we were unable to resolve.



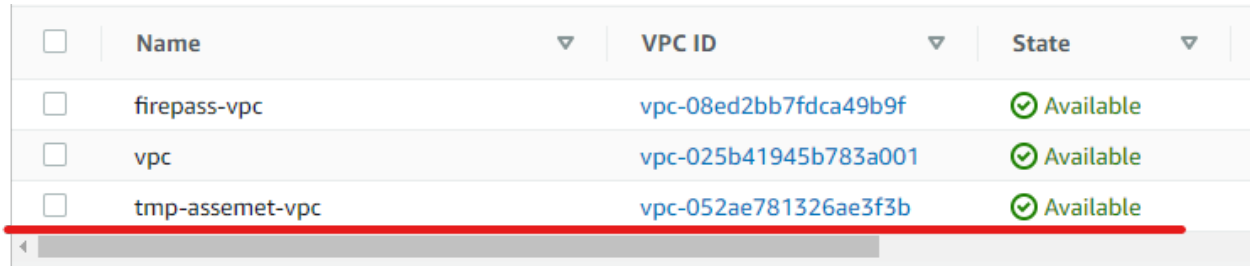
We decided to transfer the script to a terraform script in order to run it and learn how to generate a VPC, public subnet for internet connectivity, internet gateway and web server instance.

```
provider "aws" {
  region = "us-east-1"
}

# week4 VPC
# https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/vpc
resource "aws_vpc" " tmp-assemet-vpc" {
  cidr_block = "10.0.0.0/18"

  tags = {
    Name = "tmp-assemet-vpc"
  }
}

# Public Subnet with Default Route to Internet Gateway
#
https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/subne
t
resource "aws_subnet" " tmp-assemet-subnet" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.0.0/24"

  tags = {
    Name = " tmp-assemet-subnet"
  }
}

# Main Internal Gateway for VPC
#
https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/inter
net_gateway
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id
```

```
  tags = {
    Name = "Main IGW"
  }
}


# Route Table for Public Subnet
#
https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/route
_table
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.main.id

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

  tags = {
    Name = "Public Route Table"
  }
}

# Association between Public Subnet and Public Route Table
#
https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/route
_table_association
resource "aws_route_table_association" "public" {
  subnet_id      = aws_subnet.public.id
  route_table_id = aws_route_table.public.id
}

resource "aws_security_group" "webpage" {
    name = "webpage security group"
    vpc_id = aws_vpc.main.id

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

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

    ingress {
```

```
        from_port = 443
        to_port = 443
        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"]
    }
}

data "aws_ami" "ubuntu" {
    most_recent = true

    filter {
        name = "name"
        values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
    }

    filter {
        name = "virtualization-type"
        values = ["hvm"]
    }

    owners = ["099720109477"]
}

resource "aws_instance" "terraform" {
    ami = data.aws_ami.ubuntu.id
    subnet_id = local.subnet_id
    instance_type = "t2.micro"
    associate_public_ip_address = true
    security_groups = [aws_security_group.webpage.id]
    key_name = local.key_name
    tags = {
        Name = "terraform"
    }
}
```

VPC created:



Name	VPC ID	State
firepass-vpc	vpc-08ed2bb7fdca49b9f	Available
vpc	vpc-025b41945b783a001	Available
tmp-assemet-vpc	vpc-052ae781326ae3f3b	Available

Subnet creation:

☐	firepass-subnet-public2-eu-central-1b	subnet-0ae357abaf727b3b9	Available	vpc-08ed2bb7fdca49b9f | firepass-vpc
☐	–	subnet-01f1b55723562c262	Available	vpc-025b41945b783a001 | vpc
☐	firepass-subnet-public1-eu-central-1a	subnet-0161e10e80db60267	Available	vpc-08ed2bb7fdca49b9f | firepass-vpc
☐	firepass-subnet-private2-eu-central-1b	subnet-010ba03deabb67f6d	Available	vpc-08ed2bb7fdca49b9f | firepass-vpc
☐	tmp-assemet-subnet-public1-eu-central-1a	subnet-057b7ec86f8cd918b	Available	vpc-052ae781326ae3f3b | tmp-assem
☐	–	subnet-00f1f8cbdf712f9f8	Available	vpc-025b41945b783a001 | vpc
☐	firepass-subnet-private1-eu-central-1a	subnet-05768807648a2872c	Available	vpc-08ed2bb7fdca49b9f | firepass-vpc
☐	–	subnet-098c6f0ed18ccb993	Available	vpc-025b41945b783a001 | vpc

Ec2 instance:

Instance summary for i-0339cc9e2a70decb5 (tmp_assement) Info
Updated less than a minute ago

Connect | Instance state | Actions

Instance ID i-0339cc9e2a70decb5 (tmp_assement)	Public IPv4 address 35.158.122.213 | open address	Private IPv4 addresses 10.0.1.252
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-35-158-122-213.eu-central-1.compute.amazonaws.com | open address
Hostname type IP name: ip-10-0-1-252.eu-central-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-1-252.eu-central-1.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 35.158.122.213 [Public IP]	VPC ID vpc-052ae781326ae3f3b (tmp-assemet-vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-057b7ec86f8cd918b (tmp-assemet-subnet-public1-eu-central-1a)	Auto Scaling Group name –

Web server runing: