Creation and Implementation of a Virtual Private Cloud

Kaden M Marshall

University of Missouri – Computer System Administration 3850

Table of Contents

[Abstract 3](#_Toc101780129)

[Creation and Implementation of a Virtual Private Cloud 4](#_Toc101780130)

[[Heading 1] 4](#_Toc101780131)

[References 7](#_Toc101780132)

[Footnotes 8](#_Toc101780133)

[Tables 9](#_Toc101780134)

[Figures 10](#_Toc101780135)

# A screenshot of a computer Description automatically generatedStarting the Instances and IP AddressesGraphical user interface, text Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidenceAbove we can see all three instances up and running. We can also see the individual public IP addresses for each instance a public IP of 18.217.217.192 was assigned to the remote Microsoft Server, 3.142.211.14 was assigned to the RedHat instance and 3.26.206.245 was assigned to the Bitnami WordPress instance.

A screenshot of a computer

Description automatically generatedIn the screenshots to the right and below, we can also see the private IP addresses assigned to the individual instances. An IP of 192.168.100.36 was assigned to the Windows Server, an IP of 192.168.100.51 was assigned to the Bitnami WordPress instance, and an IP address of 192.168.100.38 was assigned to the RedHat instance

The reason for these specific IP addresses is because we created a VPC or a Virtual Private Cloud. We launched of these instances within our own cloud network on Amazon’s servers.

In these next screenshots you will see our security groups for our instances. We need to use the TCP protocols for all of them. For our RedHat instance we Graphical user interface, website

Description automatically generatedneeded HTTP, HTTPS, SSH, and SFTP ports to be opened. That means the inbound ports of 80,22,443 needed to be open. For the Windows Server we had to allow the RDP and HTTP to pass through. That means that ports 80 and 3389 must be open. For Bitnami we needed to open HTTP, SSH, HTTPS. This means that ports 80,22,443 must be open. All Graphical user interface, website

Description automatically generatedthe instances also required an additional protocol called echo request to allow us to ping between devices. Without it the other instances would receive the ping but would not respond. In addition to this, our Windows Server had the firewall closed to allow response when pinged, so we had to enable a protocol on there as well.

A screenshot of a computer

Description automatically generatedGraphical user interface, website

Description automatically generatedOur VPC is named vpc-sysadmin and our IPV$4 CIDR is 192.168.100.32/27 our subnet is named subnet-sysadmin and is attached to the VPC. The gateway is named gateway-sysadmin and is also attached to our VPC. We also have our routes setup up under the name routes-sysadmin. This is so we can ping between all the other devices.

A screenshot of a computer

Description automatically generated with medium confidenceText

Description automatically generatedAttached here are the pings between all different systems. I attempted to ping from our RedHat instance to our WordPress, from Windows Server to WordPress and RedHat to Windows. (The ping for RedHat to Windows and WordPress is on the same screenshot)

Graphical user interface, application

Description automatically generated

Also, as you can see from the RedHat terminal screenshots we have a user account created on our instance named usr\_redhat. We can login directly into the server using ssh with this account. We also created an account called usr\_ftp which will be displayed later.

For our Windows Server we can login using our administrator username and password generated by our private key. After I logged in, I opened the fire wall to allow incoming connections then created Graphical user interface, text

Description automatically generateda new user account called usr\_windows and added it to the administrator group, giving it administrator permissions.

The next thing I did is create a new user account for our WordPress on their GUI. The new user was called usr\_wp.

A screenshot of a computer

Description automatically generated with medium confidenceWhile on our Windows client I downloaded WinSCP this way I could transfer files via sftp. It was here that I uploaded the template of my website to the /var/www/html/ directory and was able to display the template by typing in the ip address of my RedHat Ec2 instance.

Once that was completed, I uploaded that file to my WordPress root folder which allowed me to display the template whenever I typed in the IP address of my WordPress instance. I was able to access this on both my windows server and Redhat instance.

The usr\_sftp user I created comes in to play here because now I was able to transfer and upload files from one instance to another.

I also attempted to create a mysql database. After downloading the necessary requirements, I created a database called final\_project with a table contained inside of it called information. Once I had inserted all of the information into the table I backed it up an transferred it to my WordPress instance root folder.

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generatedGraphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generated

1