Journal of Engineering Sciences
Vol11,Issue2,FEB/2020
ISSN NO:0377-9254

AN APPROACH TO AMAZON VIRTUAL PRIVATE CLOUD

[1]H. SAILAKSHMI, M.Tech, CSE Dept.
[2]M.Siva Ganesh, [2]Assistant Professor, CSE Dept.
AKRG COLLEGE OF ENGINEERING & TECHNOLOGY
Nallajerla, West Godavari, Andhra Pradesh-534112

ABSTRACT

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

You can easily customize the network configuration for your VPC. For example, you can create a public facing subnet for your web servers that has access to the Internet, and place your backend systems, such as databases or application servers, you can create a private facing subnet with no Internet access. You can leverage multiple layers of security including security groups to help control access to EC2 instances in each subnet.

Additionally, you can create a hardware virtual private network (VPN) connection between your corporate data center and your VPC and leverage the AWS Cloud as an extension of your corporate data center.

1.INTRODUCTION

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate inyour own data center, with the benefits of using the scalable infrastructure of AWS. Amazon VPC provides a web-based user interface, the Amazon VPC console. If you've signed up for anAWS account, you can access the Amazon VPC console by signing into the AWS Management Console and choosing VPC. If you prefer to use a command line interface, you have the following options:AWS Command Line Interface (AWS CLI)Provides commands for a broad set of AWS services, and is supported on Windows, macOS, and Linux/Unix. To get started, see AWS Command Line Interface User Guide. For more information about the commands for Amazon VPC, see ec2.AWS Tools for Windows PowerShell Provides commands for a broad set of AWS services for those who script in the PowerShell environment. To get started, see AWS Tools for Windows PowerShell User Guide. Amazon VPC provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named Action. For more information, see Actions in the AmazonEC2 API Reference. To build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS,AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automatically take care of tasks such as cryptographically signing your requests, retrying requests, and handling error responses. For more information, see AWS SDKs and Tools.

What Is Amazon VPC?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual networkthat you've defined. This virtual network closely resembles a traditional network that you'd operate inyour own data center, with the benefits of using the scalable infrastructure of AWS.

Getting Started with IPv4 for Amazon VPC

In this exercise, you'll create a VPC with IPv4 CIDR block, a subnet with an IPv4 CIDR block, and launcha public-facing instance into your subnet. Your instance will be able to communicate with the Internet,and you'll be able to access your instance from your local computer using SSH (if it's a Linux instance) orRemote Desktop (if it's a Windows instance). In your real world environment, you can use this scenario tocreate a public-facing web server; for example, to host a blog. This exercise is intended to help you set

up your own nondefault VPC quickly. If you alreadyhave a default VPC and you want to get started launching instances into it (and not creating orconfiguring a new VPC), see Launching an EC2 Instance into Your Default VPC. If you want toget started setting up a nondefault VPC that supports IPv6, see Getting Started with IPv6 forAmazon VPC.To complete this exercise, you'll do the following:

• Create a nondefault VPC with a single public subnet. Subnets enable you to group instances basedon your security and operational needs. A public subnet is a subnet that has access to the Internetthrough an Internet gateway.
• Create a security group for your instance that allows traffic only through specific ports.
• Launch an Amazon EC2 instance into your subnet.
• Associate an Elastic IP address with your instance. This allows your instance to access the Internet .Before you can use Amazon VPC for the first time, you must sign up for Amazon Web Services (AWS).When you sign up, your AWS account is automatically signed up for all services in AWS, including Amazon VPC. I

Amazon VPC Limits

There are limits to the number of Amazon VPC components that you can provision. You can request an increase for some of these limits.

2. SYSTEM ANALYSIS

2.1 PROPOSED SYSTEM:

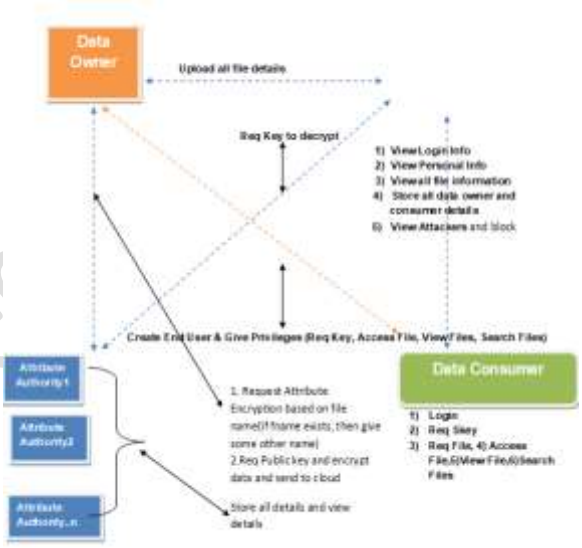
- Amazon Virtual Private Cloud (Amazon VPC) provides a logically isolated section of the AWS Cloud where you can launch AWS resources ina virtual network that you define.
- Amazon VPC have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- Ituses both IPV4 andIPV6 in VPC for secure and easy access to resources and applications.

ADVANTAGES:

Amazon Virtual Private Cloud provides features that you can use to increase and monitor the security for your virtual private cloud (VPC):

- Security groups –Act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level
- Network access control lists (ACLs) – Act as a firewall for associated subnets, controlling inbound and outbound traffic at the subnet level
- Flow logs – Capture information about the IP Traffic going to and from network interfaces in your VPC.

System Architecture:



3. LITERATURE SURVEY

Today the world of computation is moving towards pay-as-per use model due to the numerous benefits provided by this model. Hence cloud computing services are predicted as the best option for future computational world. Cloud computing is compared with other previous synonymous existing services and underlying technologies viz. utility computing, services provided via the internet using the web browsers, Infrastructure as a Service (IaaS), Platform as a Sevice (PaaS) and Software as a Service (SaaS),Grid Computing, and data centers etc. The reasons for the advances in cloud computing mainly include the recent advances in Internet backbone, high performance, and scalable infrastructure in the web technologies and data centers. As a consequence of the

inherent advantages and the enabling technological support systems, cloud computing is sure to become the key computing paradigm within the next few years. It may redefine and revolutionize the IT world with its efficient, easy to use, and economical new model

Cloud Computing is the fastest growing technology due to continuous growth of internet users. It has three major features as follows:

- Pay as peruse: Usually, business associations have to setup their own IT infrastructure to meet their business requirements of storage, network, services etc. which requires huge capital investment. Through cloud, they can get these infrastructures as a service, which saves huge upfront investment and have to pay for only what they use.
- Flexible Allocation: The resources are allocated only temporarily to a user and not permanently. The resources are allocated to user on demand from shared pool and returned back to pool after completion of its use. Then they can be re-allocated
- Virtualization: It provides resources in a scalable manner which makes it cost effective and hence becomes very critical to cloud computing.

It has three characteristics which makes it ideal for cloud computing-partitioning, isolation and encapsulation Cloud based services can be divided into three types:

(i) Infrastructure as a Service (IaaS) (ii) Platform as a Service (PaaS) and (iii) Software as a Service (SaaS).

SPORC: Group Collaboration using Untrusted Cloud Resources

Cloud-based services are an attractive deployment model for user-facing applications like word processing and calendaring. Unlike desktop applications, cloud services allow multiple users to edit shared state concurrently and in real-time, while being scalable, highly available, and globally accessible. Unfortunately, these benefits come at the cost of fully trusting cloud providers with potentially sensitive and important data. To overcome this strict tradeoff, we present SPORC, a generic framework for building a wide variety of collaborative applications with untrusted servers. In SPORC, a server observes only encrypted data and cannot deviate from correct execution without being detected. SPORC allows concurrent, low-latency editing of shared state, permits disconnected operation, and supports dynamic access control even in the presence of concurrency. We demonstrate SPORC's flexibility through two prototype applications: a causally-consistent key-value store and a browser-based collaborative text editor.

Conceptually, SPORC illustrates the complementary benefits of operational transformation (OT) and fork* consistency. The former allows SPORC clients to execute concurrent operations without locking and to resolve any resulting conflicts automatically. The latter prevents a misbehaving server from equivocating about the order of operations unless it is willing to fork clients into disjoint sets.

Disadvantages

1. The data integrity is proving only based on the filename and not on the file blocks.
2. The attackers details are not dynamic instead its maintaining the log files to store the attacker details and viewing using data mining concepts which is time consuming job and less security.

4. IMPLEMENTATION

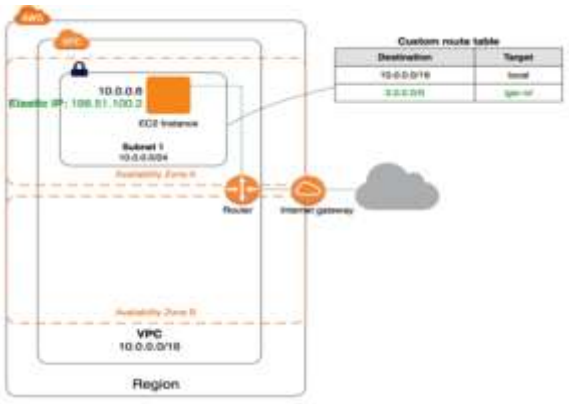
Scenario 1: VPC with a Single Public Subnet

The configuration for this scenario includes a virtual private cloud (VPC) with a single public subnet, andan Internet gateway to enable communication over the Internet. We recommend this configuration if youneed to run a single-tier, public-facing web application, such as a blog or a simple website.

This scenario can also be optionally configured for IPv6—you can use the VPC wizard to create a VPC and subnet with associated IPv6 CIDR blocks. Instances launched into the public subnet can receive IPv6addresses, and communicate using IPv6. For more information about IPv4 and IPv6 addressing, see IP Addressing in Your VPC (p. 104).

Overview

The following diagram shows the key components of the configuration for this scenario



The configuration for this scenario includes the following:

• A virtual private cloud (VPC) with a size /16 IPv4 CIDR block (example: 10.0.0.0/16). This provides65,536 private IPv4 addresses.

• A subnet with a size /24 IPv4 CIDR block (example: 10.0.0.0/24). This provides 256 private IPv4addresses.

• An Internet gateway. This connects the VPC to the Internet and to other AWS services.• An instance with a private IPv4 address in the subnet range (example: 10.0.0.6), which enables the instance to communicate with other instances in the VPC, and an Elastic IPv4 address (example:198.51.100.2), which is a public IPv4 address that enables the instance to be reached from the Internet.

• A custom route table associated with the subnet. The route table entries enable instances in the subnet to use IPv4 to communicate with other instances in the VPC, and to communicate directly over the Internet. A subnet that's associated with a route table that has a route to an Internet gateway is known as a public subnet.

Overview for IPv6

You can optionally enable IPv6 for this scenario. In addition to the components listed above, the configuration includes the following:

• A size /56 IPv6 CIDR block associated with the VPC (example: 2001:db8:1234:1a00::/56). Amazon automatically assigns the CIDR; you cannot choose the range yourself.

• A size /64 IPv6 CIDR block associated with the public subnet (example: 2001:db8:1234:1a00::/64). You can choose the range for your subnet from the range allocated to the VPC. You cannot choose the size of the subnet IPv6 CIDR block.

• An IPv6 address assigned to the instance from the subnet range (example: 2001:db8:1234:1a00::123).

• Route table entries in the custom route table that enable instances in the VPC to use IPv6 to communicate with each other, and directly over the Internet.

Routing

Your VPC has an implied router (shown in the configuration diagram above). In this scenario, the VPC wizard creates a custom route table that routes all traffic destined for an address outside the VPC to the Internet gateway, and associates this route table with the subnet.

The following table shows the route table for the example in the configuration diagram above. The first entry is the default entry for local IPv4 routing in the VPC; this entry enables the instances in this VPC to communicate with each other. The second entry routes all other IPv4 subnet traffic to the Internet gateway (for example, igw-1a2b3c4d).

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Routing for IPv6

If you associate an IPv6 CIDR block with your VPC and subnet, your route table must include separate routes for IPv6 traffic. The following table shows the custom route table for this scenario if you choose to enable IPv6 communication in your VPC. The second entry is the default route that's automatically added for local routing in the VPC over IPv6. The fourth entry routes all other IPv6 subnet traffic to the Internet gateway.

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	igw-id
::/0	igw-id

Security

AWS provides two features that you can use to increase security in your VPC: security groups and

network ACLs. Security groups control inbound and outbound traffic for your instances, and network ACLs control inbound and outbound traffic for your subnets. In most cases, security groups can meet your needs; however, you can also use network ACLs if you want an additional layer of security for your VPC. For more information, see Security.

For this scenario, you use a security group but not a network ACL. If you'd like to use a network ACL, see Recommended Rules for Scenario 1.

The following are the inbound and outbound rules for IPv4 traffic for the Web Server SG security group.

Inbound Source Protocol Port RangeComments0.0.0.0/0TCP80Allow inbound HTTP access to the web servers from any IPv4 address.0.0.0.0/0TCP443Allow inbound HTTPS access to the web servers from any IPv4 address Public IPv4 address range of your network TCP22(Linux instances) Allow inbound SSH access from your network overIPv4. You can get the public IPv4address of your local computer using a service such as http://checkip.amazonaws.com or https://checkip.amazonaws.com. If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers. Public IPv4 address range of your networkTCP3389(Windows instances) Allow inbound RDP access from your network overIPv4.The security group ID (sg-xxxxxxxx)All All(Optional) Allow inbound traffic from other instances associated with this security group. This rule is automatically added to the default security group for the VPC; for any custom security group you create, you must manually add the rule to allow this type of communication.

Outbound (Optional) Destination Protocol Port RangeComments0.0.0.0/0AllAllDefault rule to allow all outbound access to any IPv4 address. If you want your web server to initiate outbound traffic, for example, to get software updates, you can keep the default outbound rule. Otherwise, you can remove this rule.

Security for IPv6

If you associate an IPv6 CIDR block with your VPC and subnet, you must add separate rules to your security group to control inbound and outbound IPv6 traffic for your web server instance. In this scenario, the web server will be able to receive all Internet traffic over IPv6, and SSH or RDP traffic from your local network over IPv6.

The following are the IPv6-specific rules for the Web Server SG security group (which are in addition to the rules listed above).

Inbound Source Protocol Port Range Comments::/0TCP80Allow inbound HTTP access to the web servers from any IPv6 address.::/0TCP443Allow inbound HTTPS access to the web servers from any IPv6 address.IPv6 address range of yournetworkTCP22(Linux instances) Allow inbound SSH access over IPv6 from your network.IPv6 address range of yournetworkTCP3389(Windows instances) Allow inbound RDP access over IPv6 from your network Outbound (Optional)Destination Protocol Port Range Comments::/0AllAllDefault rule to allow all outbound access to any IPv6 address. If you want your web server to initiate outbound traffic, for example, to get software updates, you can keep the default outbound rule. Otherwise, you can remove this rule.

Implementing Scenario 1

To implement scenario 1, create a VPC using the VPC wizard, create and configure the Web Server SG security group, and then launch an instance into your VPC.

These procedures include optional steps for enabling and configuring IPv6 communication for your VPC. You do not have to perform these steps if you do not want to use IPv6 in your VPC.

To create a VPC

1.Open the Amazon VPC console

2.On the dashboard, choose Launch VPC Wizard.

3.Choose the first option, VPC with a Single Public Subnet, and then choose Select.

4.(Optional) You can name your VPC and subnet to help you to identify them later in the console. You can specify your own IPv4 CIDR block ranges for the VPC

and subnet, or you can keep the default values (10.0.0.0/16 and 10.0.0.0/24 respectively).

5.(Optional, IPv6-only) For IPv6 CIDR block, choose Amazon-provided IPv6 CIDR block. For Public subnet's IPv6 CIDR, choose Specify a custom IPv6 CIDR and specify the hexadecimal pair value for your subnet, or keep the default value (00).

6.Keep the rest of the default settings, and choose Create VPC.

To create the Web Server SG security group

1.Open the Amazon VPC console

2.In the navigation pane, choose Security Groups.

3.Choose Create Security Group.

4.Provide a name and description for the security group. In this topic, the name Web Server SG is used as an example. Select the ID of your VPC from the VPC menu, and then choose Yes, Create.

5.Select the Web Server SG security group that you just created. The details pane include a tab for information about the security group, plus tabs for working with its inbound rules and outbound rules.

6.On the Inbound Rules tab, choose Edit, and then do the following:

• Select HTTP from the Type list, and enter 0.0.0.0/0 in the Source field.

• Choose Add another rule, then select HTTPS from the Type list, and enter 0.0.0.0/0 in the Source field.

• Choose Add another rule, then select SSH (for Linux) or RDP (for Windows) from the Type list. Enter your network's public IP address range in the Source field. (If you don't know this address range, you can use 0.0.0.0/0 for testing purposes; in production, you authorize only a specific IP address or range of addresses to access your instance.)

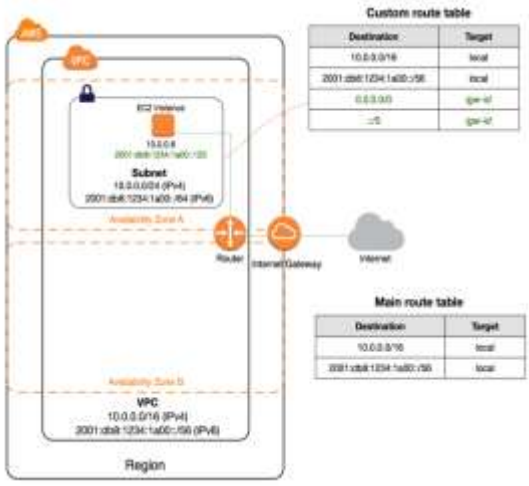
• (Optional) Choose Add another rule, then select ALL traffic from the Type list. In the Source field, enter the ID of the Web Server SG security group.

• (Optional, IPv6-only) Choose Add another rule, select HTTP from the Type list, and enter ::/0 in the Source field.• (Optional, IPv6-only) Choose Add another rule, select HTTPS from the Type list, and enter ::/0in the Source field.

• (Optional, IPv6-only) Choose Add another rule, select SSH (for Linux) or RDP (for Windows) from the Type list. Enter your network's IPv6 address range in the Source field. (If you don't know this address range, you can use ::/0 for testing purposes; in production, you authorize only a specificIPv6 address or range of addresses to access your instance.)

7.Choose Save.

8.(Optional) On the Outbound Rules tab, choose Edit. Locate the default rule that enables all outbound traffic, choose Remove, and then choose Save.



To launch an instance into the VPC

1.Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2.From the dashboard, choose Launch Instance.

3.Follow the directions in the wizard. Choose an AMI, choose an instance type.

4.On the Configure Instance Details page, select the VPC that you created in step 1 from the Network list, and then specify a subnet.

5.(Optional) By default, instances launched into a non default VPC are not assigned a public IPv4address. To be able to connect to your instance, you can assign a public IPv4 address now, or allocate an Elastic IP address and assign it to your instance after it's launched. To assign a public IPv4 address now, ensure that you select Enable from the Auto-assign Public IP list. Note You can only use the auto-assign public IP feature for a single, new network interface with the device index of eth0. For more information, see Assigning a Public IPv4 Address During Instance Launch.

6.(Optional, IPv6-only) You can auto-assign an IPv6 address to your instance from the subnet range .For Auto-assign IPv6 IP, choose Enable.

7.On the next two pages of the wizard, you can configure storage for your instance, and add tags. On

the Configure Security Group page, select the Select an existing security group option, and select the Web Server SG security group that you created in step 2. Choose Review and Launch.

8. Review the settings that you've chosen. Make any changes that you need, and then choose Launch to choose a key pair and launch your instance.

9.If you did not assign a public IPv4 address to your instance in step 5, you will not be able to connect to it over IPv4. Assign an Elastic IP address to the instance:

a. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

b. In the navigation pane, choose Elastic IPs.

c. Choose Allocate new address.

d. Choose Allocate .Note If your account supports EC2-Classic, first choose VPC.

e. Select the Elastic IP address from the list, choose Actions, and then choose Associate address.

f. Select the instance to associate the address with, and then choose Associate.

5. CONCLUSION

This paper proposes a semi-anonymous attribute-based privilege control scheme Anony Control and a fully-anonymous attribute-based privilege control scheme Anony Control-F to address the user privacy problem in a cloud storage server. Using multiple authorities in the cloud computing system, our proposed schemes achieve not only fine-grained privilege control but also identity anonymity while conducting privilege control based on users' identity information. More importantly, our system can tolerate up to N□2 authority compromise, which is highly preferable especially in Internet-based cloud computing environment. We also conducted detailed security and performance analysis which shows that Anony Control both secure and efficient for cloud storage system. The Anony Control-F directly inherits the security of the Anony Control and thus is equivalently secure as it, but extra communication overhead is incurred during the 1-out-of-n oblivious transfer. One of the promising future works is to introduce the efficient user revocation mechanism on top of our anonymous ABE. Supporting user revocation is an important issue in the real application, and this is a great challenge in the application of ABE schemes. Making our

REFERENCES

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in CRYPTO. Springer, 1985, pp. 47–53.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in EUROCRYPT. Springer, 2005, pp. 457–473.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in CCS. ACM, 2006, pp. 89–98.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in S&P. IEEE, 2007, pp. 321–334.

[5] M. Chase, "Multi-authority attribute based encryption," in TCC. Springer, 2007, pp. 515 534.

[6] M. Chase and S. S. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in CCS. ACM, 2009, pp. 121– 130.

[7] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," Information Sciences, vol. 180, no. 13, pp. 2618–2632, 2010.

[8] V. Bo˘zovi´c, D. Socek, R. Steinwandt, and V. I. Vill´anyi, "Multiauthority attribute-based encryption with honest-but-curious central authority," IJCM, vol. 89, no.3, pp. 268–283, 2012.

[9] F. Li, Y. Rahulamathavan, M. Rajarajan, and R.-W. Phan, "Low complexity multi-authority attribute based encryption scheme for mobile cloud computing," in SOSE. IEEE, 2013, pp. 573–577.

[10] K. Yang, X. Jia, K. Ren, and B. Zhang, "Dac-macs: Effective data access control for multi-authority cloud storage systems," in INFOCOM. IEEE, 2013, pp. 2895–2903.

[11] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in EUROCRYPT. Springer, 2011, pp. 568–588.

[12] S. M¨uller, S. Katzenbeisser, and C. Eckert, "On multi-authority ciphertext-policy attribute-based encryption," Bulletin of the Korean Mathematical Society, vol. 46, no. 4, pp. 803–819, 2009.