CHAPTER 01

ABOUT THE COMPANY

1.1 Overview

Fundamentals is an EdTech company that provides computer coding classes, academic project assistance, and IT industry partnerships to students. Our goal is to empower students to innovate, create and collaborate with the aim to build a better future. We are committed to helping our students develop the skills necessary to succeed in the tech industry and to providing them with the support and resources they need to achieve their goals.

1.2 Services

- **Computer Coding Classes**: Fundamentals offers a variety of coding classes to students of all ages and skill levels. Our courses cover a range of programming languages, including Python, JavaScript, and Ruby, and are designed to provide a hands-on learning experience. We also offer coding classes for specific applications, such as web development and game design.
- **Academic Project Assistance**: We provide assistance to students in completing their academic projects, including research assistance and project development support. Our team of experts is available to help students with any questions they may have, from brainstorming ideas to finalizing their projects.
- **IT Industry Partnerships**: Fundamentals has partnerships with various IT companies, providing students with internship opportunities and job placements upon graduation. We work closely with our industry partners to ensure that our students have access to the latest tools and technologies and are well-prepared for careers in the tech industry.

The company is aimed at providing the recent technologies used in real-time to make students industry ready after completion of their graduation. We believe that education is a fundamental right and everyone should have access to quality higher education. With this view in mind, we strive to create opportunities for those who have genuine aspiration and honest intention to seek high quality education and great academic and industry experience.

We believe in creating an inclusive environment where everyone is welcome and encouraged to participate. We welcome students from all backgrounds and believe that diversity is essential to creativity and innovation.

1.3 Projects

We pride on providing some of the important projects like Face mask recognition, Sound recognition, Face recognition using Artificial Intelligence. Also Electrical and Electronics projects like "Kisan bandhu” an early warning system for the detection of wild animals around villages, "Mookadhvani" a hand gesture to voice conversion system for the specially able people, "Virtual drumstick" a drumstick fit with motion sensors and capable of producing sound virtually without the need of actual drums, "Trap It" an AI based camera trap capable of capturing pictures when motion is detected and can identify objects in the captured image and Motorized headlamp cluster actuator for 2 wheel vehicles.

1.4 Values

At Fundamentals, we value:

- **Innovation:** We strive to foster creativity and innovation among our students, encouraging them to explore new ideas and solutions. We believe that innovation is essential to success in the tech industry and encourage our students to think outside the box.
- **Collaboration:** We believe that collaboration is essential to success and encourage our students to work together to achieve their goals. We encourage our students to share their ideas and to learn from one another.
- **Excellence:** We set high standards for ourselves and our students, with the aim of achieving excellence in all that we do. We believe that excellence is essential to success in the tech industry and encourage our students to strive for excellence in their work.
- **Inclusivity:** We believe in creating an inclusive environment where everyone is welcome and encouraged to participate. We welcome students from all backgrounds and believe that diversity is essential to creativity and innovation.

1.5 Team

Our team is comprised of experienced professionals in the fields of education and technology. We are passionate about providing students with the tools and skills necessary to succeed in the tech industry. Our instructors are experts in their respective fields and have years of experience teaching coding and other technical skills to students of all ages. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows. This reduces inefficiencies and saves time.

1.6 Location

Fundamentals is located in Chikkamagaluru. We offer online classes as well, so students can take our courses from anywhere in the world.

1.7 Contact

To learn more about Fundamentals, or contact us at our office opposite to AIT Girls' hostel, by-pass road Chikamagalur, ph : 9900792744. We look forward to hearing from you.

1.8 Organization of the report

Project report is organized as follows:

Chapter 1: About the company, it presents brief overview if the company Fundamentals.

Chapter 2: Introduction about the devops, it presents brief overview if the devops, how devops will work, what are the benefits of devops.

Chapter 3: Task Performed, it presents the Introduction, motivation of the project, problem statement, objective, scope of the project.

Chapter 4: Implementation, this section presents the introduction to the plugins and brief description .

Chapter 5: Results and analysis, this section presents snapshots of each module and performance analysis.

Chapter 6: Conclusion and future enhancements, this section presents the conclusions and future enhancements

1.9 Summary

This chapter describes about the company. Section 1.1 presents the brief overview about the company. Section 1.2 describes services of company. Section 1.3 describes projects of the company. Section 1.4 values of the company. Section 1.5 team of the company. Section 1.6 Location of the Company Section 1.7 describes the Contact of the company. Section 1.8 describes the Organization of report.

CHAPTER 02

ABOUT THE DEVOPS

2.1 Overview of the DevOps

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

DevOps is a software development methodology that emphasizes collaboration and communication between development and operations teams to improve the software delivery process. It aims to bridge the gap between software development and IT operations and to enable more efficient and effective delivery of software applications.

The primary goal of DevOps is to increase the speed and agility of software development and deployment, while also improving the quality of the software delivered. DevOps achieves this by breaking down silos between development and operations teams, encouraging the adoption of automation, and creating a culture of continuous improvement and learning.

DevOps practices include continuous integration and continuous delivery (CI/CD), infrastructure as code (IAC), monitoring and logging, and agile development methodologies. DevOps also relies heavily on tools and technologies such as Docker, Kubernetes, Ansible, Jenkins, and Git. Some benefits of DevOps include faster time-to-market, increased efficiency and productivity, better collaboration and communication between teams, and improved customer satisfaction. DevOps is a continuously evolving field, with new tools and practices being developed all the time. As such, it requires a willingness to embrace change and a commitment to ongoing learning and improvement.

2.1.1 How DevOps work

Under a DevOps model, development and operations teams are no longer "siloed." Sometimes, these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function. In some DevOps models, quality assurance and security teams may also become more tightly integrated with development and operations and throughout the application lifecycle.

When security is the focus of everyone on a DevOps team, this is sometimes referred to as DevOps. These teams use practices to automate processes that historically have been manual and slow. They use a technology stack and tooling which help them operate and evolve applications quickly and reliably. These tools also help engineers independently accomplish tasks (for example, deploying code or provisioning infrastructure) that normally would have required help from other teams, and this further increases a team's velocity.

DevOps is a set of practices and methodologies that aims to increase collaboration and communication between software development and operations teams to enable the rapid delivery of high-quality software products.

2.2 Benefits of DevOps

1) Speed

Move at high velocity so you can innovate for customers faster, adapt to changing markets better, and grow more efficient at driving business results. The DevOps model enables your developers and operations teams to achieve these results. For example, micro services and continuous delivery let teams take ownership of services and then release updates to them quicker.

2) Rapid Delivery

Increase the frequency and pace of releases so you can innovate and improve your product faster. The quicker you can release new features and fix bugs, the faster you can respond to your customers' needs and build competitive advantage. Continuous integration and continuous delivery are practices that automate the software released deploy.

3) Reliability

Ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining a positive experience for end users. Use practices like continuous integration and continuous delivery to test that each change is functional and safe. Monitoring and logging practices help you stay informed of performance in real-time.

4) Scale

Operate and manage your infrastructure and development processes at scale. Automation and consistency help you manage complex or changing systems efficiently and with reduced risk. For example, infrastructure as code helps you manage your development, testing, and production environments in a repeatable and more efficient manner.

5) Improved Collaboration

Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows. This reduces inefficiencies and saves time (e.g. reduced handover periods between developers and operations, writing code that takes into account the environment in which it is run).

6) Security

Move quickly while retaining control and preserving compliance. You can adopt a DevOps model without sacrificing security by using automated compliance policies, fine grained controls, and configuration management techniques. For example, using infrastructure as code and policy as code, you can define and then track compliance at scale. . It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

7) Faster time to market

DevOps practices can help teams to release software faster by automating the deployment process, enabling continuous integration and delivery, and reducing the time it takes to detect and fix bugs.

8) Improved collaboration

By breaking down the silos between development and operations teams, DevOps encourages collaboration and communication between them. This can result in better-quality software, fewer errors, and more efficient processes.

9) Increased efficiency

DevOps practices help to streamline processes and eliminate bottlenecks in software development, resulting in faster delivery times, improved resource utilization, and reduced costs.

10) Greater flexibility

DevOps enables teams to respond quickly to changing customer needs and market demands, allowing them to pivot and adapt their software products as required.

11) Better quality software

By automating testing, monitoring, and feedback loops, DevOps can help teams to detect and fix errors and issues more quickly, resulting in higher-quality software.

2.2.1 Importance of DevOps

Software and the Internet have transformed the world and its industries, from shopping to entertainment to banking. Software no longer merely supports a business; rather it becomes an integral component of every part of a business. Companies interact with their customers through software delivered as online services or applications and on all sorts of devices. They also use software to increase operational efficiencies by transforming every part of the value chain, such as logistics, communications, and operations. In a similar way that physical goods companies transformed how they design, build, and deliver products using industrial automation throughout the 20th century, companies in today’s world must transform how they build and deliver software.

2.2.2 DevOps Cultural Philosophy Transitioning to DevOps

It requires a change in culture and mindset. At its simplest, DevOps is about removing the barriers between two traditionally siloed teams, development and operations. In some organizations, there may not even be separate development and operations teams; engineers may do both. With DevOps, the two teams work together to optimize both the productivity of developers and the reliability of operations. They strive to communicate frequently, increase efficiencies, and improve the quality of services they provide to customers. They take full ownership for their services, often beyond where their stated roles or titles have traditionally been scoped by thinking about the end customer’s needs and how they can contribute to solving those needs.

Quality assurance and security teams may also become tightly integrated with these teams. Organizations using a DevOps model, regardless of their organizational structure, have teams that view the entire development and infrastructure lifecycle as part of their responsibilities. The main goal is to develop a sustainable infrastructure for specific applications and ensure high scalability and integration in the modern-day software development process.

2.3 Functionality of DevOps

• **Collaboration:** DevOps emphasizes the need for collaboration between software development and IT operations teams. This collaboration ensures that all stakeholders are aligned on the goals of the project, and that communication channels are open throughout the development process.

• **Automation:** DevOps seeks to automate as much of the software development lifecycle as possible, from testing to deployment to monitoring. This automation ensures that processes are repeatable and consistent, reducing the risk of errors and enabling faster delivery of software applications.

• **Continuous delivery**: DevOps prioritizes the continuous delivery of software and applications, with frequent releases that provide value to end-users. This approach ensures that feedback is received early in the development process, enabling teams to make adjustments and improvements as needed.

• **Monitoring and feedback:** DevOps teams use a variety of tools and processes to monitor software applications in production, ensuring that any issues are quickly identified and resolved. This feedback loop enables continuous improvement and ensures that end-users receive a high-quality experience.

2.4 DevOps Practices

The following are DevOps best practices:

• **Continuous Integration** - Continuous integration is a software development practice where developers regularly merge their code changes into a central repository. The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates. • Continuous Delivery – Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after thebuild stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that passed through a standardized test processes.

• **Microservices** - The microservices architecture is a design approach to build a single application as a set of small services. Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism, typically an HTTPbased application programming interface (API). Microservices are built around business capabilities; each service is scoped to a single purpose. You can use different frameworks or programming languages to write microservices and deploy them independently, as a single service, or as a group of services.

AWS Services – Amazon Elastic Container Service, AWS Lambda

• **Infrastructure as Code -** Infrastructure as code is a practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration. The cloud's API-driven model enables developers and system administrators to interact with infrastructure programmatically, and at scale, instead of needing to manually set up and configure resources. Thus, engineers can interface with infrastructure using code-based tools and treat infrastructure in a manner similar to how they treat application code. Because they are defined by code, infrastructure and servers can quickly be deployed using standardized patterns, updated with the latest patches and versions, or duplicated in repeatable ways AWS Services - AWS CloudFormation, AWS OpsWorks.

• **Configuration Management -** Developers and system administrators use code to automate operating system and host configuration, operational tasks, and more. The use of code makes configuration changes repeatable and standardized. It frees developers and system's administrators from manually configuring operating systems, system applications, or server software. AWS Services - AWS Systems Manager.

• **Policy as Code** - With infrastructure and its configuration codified with the cloud, organizations can monitor and enforce compliance dynamically and at scale. Infrastructure that is described by code can thus be tracked, validated, and reconfigured in an automated way. This makes it easier for organizations to govern changes over resources and ensure that security measures are properly enforced in a distributed manner This allows teams within an organization to move at higher velocity since noncompliant resources can be automatically flagged for further investigation or even automatically brought back into compliance. AWS Service - AWS Config.

• **Monitoring and Logging -** Organizations monitor metrics and logs to see how application and infrastructure performance impacts the experience of their product's end user. By capturing, categorizing, and then analyzing data and logs generated by applications and infrastructure, organizations understand how changes or updates impact users, shedding insights into the root causes of problems or unexpected changes. Active monitoring becomes increasingly important as services must be available 24/7 and as application and infrastructure update frequency increases.

Creating alerts or performing real-time analysis of this data also helps organizations more proactively monitor their services. AWS Services – Amazon CloudWatch, AWS X-Ray, AWS CloudTrail.

- **Communication and Collaboration -** Increased communication and collaboration in an organization is one of the key cultural aspects of DevOps. The use of DevOps tooling and automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations. Building on top of that, these teams set strong cultural norms around information sharing and facilitating communication through the use of chat applications, issue or project tracking systems, and wikis. This helps speed up communication across developers, operations, and even other teams like marketing or sales, allowing all parts of the organization to align more closely on goals and projects.

AWS Services – AWS Elastic Beanstalk, AWS CodeCommit.

- **Improved Collaboration**: Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Teams set strong cultural norms around information sharing and facilitating communication through the use of chat applications. Developers and system administrators use code to automate operating system and host configuration, operational tasks, and more. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflow.

2.5 Organizational structure

- **Centralized DevOps Team:** In this structure, a centralized DevOps team is responsible for implementing DevOps practices across the organization. This team typically works with other departments and teams to implement and maintain DevOps processes, tools, and methodologies.

- **Decentralized DevOps Teams:** In this structure, DevOps responsibilities are distributed across various teams within the organization. Each team is responsible for implementing DevOps practices within their respective areas, with some coordination from a central DevOps team or leadership group.

- **Hybrid DevOps Teams**: This structure combines elements of both centralized and decentralized DevOps teams. A central DevOps team is responsible for defining and enforcing DevOps standards and practices, while individual teams are responsible for implementing and maintaining those practices within their respective areas.

• DevOps as a Service: Some organizations may choose to outsource DevOps responsibilities to a third-party provider, such as a cloud provider or DevOps consultancy. In this model, the provider is responsible for implementing and maintaining DevOps processes and tools, while the organization focuses on their core business activities.

2.6 Technologies and Skills in DevOps

DevOps is a practice that requires a combination of technologies and skills to ensure the efficient and reliable delivery of software. Some of the technologies and skills commonly used in DevOps include:

- **Continuous Integration and Continuous Delivery (CI/CD):** CI/CD is the process of automating the building, testing, and deployment of software. Tools like Jenkins, CircleCI, and Travis CI can be used to automate these processes.

- **Infrastructure as Code (IaC):** IaC is the practice of defining infrastructure and configuration settings as code, which can be version-controlled and automated. Tools like Terraform, Ansible, and Chef can be used for IaC.

- **Cloud computing platforms:** Cloud computing platforms like AWS, Azure, and Google Cloud provide the infrastructure and services necessary for modern software development and deployment.

- **Containerization:** Containerization allows for applications to be packaged with their dependencies and run in a consistent environment. Docker is a popular tool for containerization.

- **Monitoring and Logging:** Monitoring and logging tools like Prometheus, Grafana, and ELK stack are used to monitor the performance and health of the application and infrastructure.

- **Collaboration and communication:** Collaboration and communication skills are essential in DevOps. Tools like Slack, JIRA, and Confluence can be used for collaboration and communication.

- **Agile and Lean methodologies:** DevOps is heavily influenced by Agile and Lean methodologies, which emphasize collaboration, iterative development, and continuous improvement.

- **Automation and scripting:** Automation and scripting skills are necessary for automating repetitive tasks and building automation pipelines. Scripting languages like Python, Bash, and PowerShell are commonly used in DevOps.

2.6.1 Git Hub

GitHub is a web-based platform that provides developers with a powerful suite of tools and services for version control and collaborative software development. Launched in 2008, GitHub quickly became the go-to platform for millions of developers worldwide, offering an intuitive user interface, robust features, and seamless integration with other development tools.

One of the key features of GitHub is its version control system, which enables developers to track changes to their code over time, collaborate with other developers, and manage complex software projects. This allows developers to work more efficiently and effectively, reducing the risk of errors and conflicts in the code. With GitHub's version control system, developers can create branches of code, make changes, and merge those changes back into the main codebase, all while keeping a detailed record of every change made.

Another important feature of GitHub is its support for open-source software development. GitHub provides a centralized platform where developers can share their code with others, collaborate on projects, and contribute to open- source software projects. This has helped to fuel a vibrant and active community of developers who work together to create new software solutions, fix bugs, and improve existing code. This allows developers to work more efficiently and effectively, reducing the risk of errors and conflicts in the code.

Finally, GitHub provides a range of tools and services that streamline the software development process, including code review, bug tracking, and project management. With GitHub, developers can easily manage their code repositories, track issues and bugs, and collaborate with other developers in real-time. This makes it easier for teams to work together and deliver high-quality software faster and with fewer errors.

Overall, GitHub has revolutionized the way developers work by providing a powerful suite of tools and services that enable version control, collaboration, and project management. With its intuitive user interface, robust features, and support for open-source development, GitHub has become an essential tool for developers around the world.

2.6.2 Microsoft Azure

Microsoft Azure is Microsoft's cloud computing platform, providing a wide variety of services you can use without purchasing and provisioning your own hardware. Azure enables the rapid development of solutions and provides the resources to accomplish tasks that may not be feasible in an on-premises environment. Azure's compute, storage, network, and application services allow you to focus on building great solutions without the need to worry about how the physical infrastructure is assembled.

Microsoft provides support for public, private, and hybrid clouds. Microsoft Azure, the focus of this book, is a public cloud. Microsoft Azure Stack is an add-on to Windows Server 2016 that allows you to deploy many core Azure services in your own datacenter and provides a self-service portal experience to your users. Getting started with Microsoft Azure integrate these into a hybrid cloud through the use of a virtual private network.

Microsoft has deployed Azure datacenters in over 22 regions around the globe from Melbourne to Amsterdam and Sao Paulo to Singapore. Additionally, Microsoft has an arrangement with 21Vianet, making Azure available in two regions in China. Microsoft has also announced the deployment of Azure. Getting started with Microsoft Azure another eight regions. Only the largest global enterprises are able to deploy datacenters in this manner, so using Azure makes it easy for enterprises of any size to deploy their services close to their customers, wherever they are in the world. And you can do that without ever leaving your office.

Azure Virtual Machines, the Azure IaaS offering, is a popular choice when migrating services to Azure because it enables the “lift and shift” model for migration. You can configure a VM similar to the infrastructure currently running your services in your datacenter and migrate your software to the new VM. You might need to make tweaks, such as URLs to other services. Getting started with Microsoft Azure storage, but many applications can be migrated in this manner.

One of the key advantages of Azure is its scalability. Azure allows users to scale their resources up or down as needed, without the need for upfront investment in hardware or infrastructure. This means that business can easily scale their operations to meet changing demand, without worrying about hardware limitations or capacity constraints.

2.7 Summary

This chapter describes about the department in the company. Section 2.1 presents the overview of department in the company. Section 2.2 describes the benefits of devops. Section 2.3 follows the functionality of the company. Section 2.4 talks about the practice of devops. Section 2.5 briefs out organizational structure. Section 2.6 explains technologies and skills used in the company department.

CHAPTER 03

AMAZON WEB SERVICES

3.1 Introduction about AWS

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers including the fastest-growing startups, largest enterprises, and leading government agencies are using AWS to lower costs, become more agile, and innovate faster.

AWS offers a comprehensive set of cloud-based services, including compute, storage, database, analytics, machine learning, networking, security, and more. These services can be used to build, deploy, and manage a wide range of applications, from simple web applications to complex enterprise-level systems.

One of the key advantages of AWS is its scalability. AWS allows users to scale their resources up or down as needed, without the need for upfront investment in hardware or infrastructure. This means that businesses can easily scale their operations to meet changing demand, without worrying about hardware limitations or capacity constraints.

AWS also offers high levels of security and reliability. AWS has a range of security features and tools that help businesses to protect their data and applications, including encryption, identity and access management, and network security. Additionally, AWS has a global infrastructure that provides high levels of availability and redundancy, ensuring that applications are always accessible and that data is always safe and secure.

Amazon Web Services (AWS) is a cloud computing platform provided by Amazon.com, offering a wide range of cloud-based computing services such as storage, databases, analytics, machine learning, networking, security, and many more. AWS was launched in 2006 and has since become one of the most widely used cloud computing platforms in the world, serving millions of customers from startups to enterprises across various industries.

AWS provides its users with a range of services to help them build and manage their applications, including computing power, storage, and databases. With AWS, users can easily scale their applications up or down as per their requirements and only pay for what they use. This enables businesses to run their applications more efficiently while saving costs on infrastructure and maintenance. AWS also offers a comprehensive set of tools and services to manage and monitor their applications and ensure that they are running smoothly.

3.1.1 Cloud computing using AWS

Cloud computing is the on-demand delivery of IT resources over the Internet with pay- as-we-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS). Cloud infrastructure consists of servers, storage devices, network, cloud management software, deployment software, and platform virtualization.

Cloud computing using AWS (Amazon Web Services) involves utilizing a wide range of cloud-based services and solutions provided by Amazon to build, deploy, and manage various types of applications and infrastructure. AWS provides an extensive range of cloud-based services and solutions, including computing, storage, databases, analytics, machine learning, networking, security, and more.

3.1.2 Characteristics of Cloud computing

- **On Demand Self Service**: Cloud Computing allows the users to use web services and resources on demand. One can logon to a website at any time and use them.
- **Broad Network Access**: Since cloud computing is completely web based, it can be accessed from anywhere 00000and at any time.
- **Resource Pooling:** Cloud computing allows multiple tenants to share a pool of resources. One can share single physical instance of hardware, database and basic infrastructure.
- **Rapid Elasticity:** It is very easy to scale the resources vertically or horizontally at any time. Scaling of resources means the ability of resources to deal with increasing or decreasing demand. The resources being used by customers at any given point of time are automatically monitored.
- **Measured Service**: In this service cloud provider controls and monitors all the aspects of cloud service. Resource optimization, billing, and capacity planning etc. depend on it.

3.1.3 Advantages of Cloud Computing

- **Trade capital expense for variable expense –** Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources, and pay only for how much you consume.

- **Benefit from massive economies of scale** – By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay as-you-go prices.

- **Stop guessing capacity** – Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often end sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little capacity as you need, and scale up and down as required with only a few minutes' notice.

- **Increase speed and agility –** In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organization.

- **Stop spending money running and maintaining data centers** – Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers.

- **Go global in minutes** – Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide lower latency and a better experience for your customers at minimal cost.

3.2 AWS Management Console

AWS Management Console is a web application for managing Amazon Web Services. AWS Management Console consists of list of various services to choose from. It also provides all information related to our account like billing. This console provides an inbuilt user interface to perform AWS tasks like working with Amazon S3 buckets, launching and connecting to Amazon EC2 instances, setting Amazon CloudWatch alarms, etc.

3.2.1 Region

AWS Region is a separate geographic area where we cluster data centers. Each AWS Region is completely independent. AWS Regions are separate geographic areas that AWS uses to house its infrastructure. These are distributed around the world so that customers can choose a region closest to them in order to host their cloud infrastructure there. The closer your region is to you, the better, so that you can reduce network latency as much as possible for your end-users. You want to near the data centers for fast service. Geographical area where AWS has setup the physical resources. The instance created for specific region in confined to that region. As of November 2019, there are 23 AWS regions. Regions having most of the services.

- **Americas**: US East (N. Virginia), US West (N. California)
- **Asia Pacific**: Singapore, Sydney,
- **EU:** Frankfurt, Ireland

3.3 Availability Zones

Each AWS Region consists of multiple, isolated, and physically separate AZ's within a geographic area. An AWS Availability Zone (AZ) is the logical building block that makes up an AWS Region. There are currently 69 AZs, which are isolated locations-data centerswithin a region. Each region has multiple AZs and when you design your infrastructure to have backups of data in other AZs you are building a very efficient model of resiliency, i.e. a core concept of cloud computing.

An Availability Zone (AZ) is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region. AZs give customers the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center.

Availability zones: Each AWS Region consists of multiple, isolated, and physically separate AZ's within a geographic area**.**

Local zone: A Local zone is an extension of an AWS Region that is geographically close to your users, for low latency communication**.**

Edge location: its data center, where end users access services located at AWS to reduce latency numbers on the map – each AWS reasons consists of isolated, multiple physically separated resources. This number will specify the availability resources. These zones are required for fault tolerance.

3.4 Services of AWS

Amazon Web services (AWS) is a cloud computing platform that offers a wide range of services and resources to help individuals, businesses, and organizations to build and manage their applications and infrastructure in the cloud. Some of the most popular AWS services include:

1. Compute Services: These services provide computing resources to run applications in the cloud. Some examples of AWS compute services are Amazon Elastic Compute Cloud (EC2), Amazon Elastic Kubernetes Service (EKS), AWS Lambda, Amazon Elastic Container Service (ECS), Light sail.

2. Storage Services: AWS provides various storage services to store and retrieve data from the cloud. Some of the popular AWS storage services are Amazon Simple Storage Service(S3), Amazon Elastic Block Store (EBS), Amazon Elastic File System (EFS), and Amazon Glacier.

3. Database Services: AWS provides managed database services that are easy to deploy, operate, and scale. Some popular AWS database services are Amazon Relational Database Service(RDS), Amazon DynamoDB, Amazon Aurora, Amazon Document DB and Amazon Neptune.

4. Security Services: AWS provides various security services to protect the applications and data stored in the cloud. Some popular AWS security services are AWS Identity and Access Management(IAM), AWS Certificate Manager, AWS Shield, and AWS Web Application Firewall(WAF).

5. Networking and Content Delivery Services: AWS provides services to help businesses connect their resources and content delivery networks to deliver content to users with low latency and high data transfer rates. Some popular AWS networking services are Virtual Private Cloud (VPC), AWS Direct Connect, Elastic Load Balancing(ELB), and Amazon Route S3, Amazon CloudFront.

6. Analytics Services: AWS provides services to analyze and process data at scale. Some popular AWS analytics services are Amazon Kinesis, Amazon Redshift, and Amazon Athena, Amazon QuickSight.

7. Machine Learning Services: AWS provides services to build and deploy machine learning models at scale. Some popular AWS machine learning services are Amazon SageMaker, Amazon Comprehend, and Amazon Rekognition, Amazon Transcribe.

8. Management and Governance Services: AWS provides services to help businesses manage their infrastructure and resources in the cloud efficiently. Some popular AWS management and governance services are AWS CloudFomation, AWS CloudTrial, AWS Config, AWS System Manager.

9. Internet of Things(IoT) Services: AWS provides services to several IoT resources to build, deploy and manage such as AWS IoT Core, AWS IoT analytics, and AWS IoT Device Defender.

10. Developer Services: AWS provides several services for developer tools such as AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, and AWS CodeDeploy.

3.5 Tools of AWS

Amazon Web services(AWS) is a comprehensive cloud computing platform that offers a wide range of tools and services to support various use cases.

Here are some of the key tools supported by AWS:

1. Amazon Elastic Cloud Compute(EC2): A web service that provides resizable compute capacity in the cloud.

2. Amazon Simple Storage Service(S3): An object storage service that offers industry- leading scalability, data availability, security, and performance.

3. Amazon Relational Database Service(RDS): A web services that makes it easier to set up, operate, and scale a relational database in the cloud.

4. Amazon DynamoDB: A fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.

5. Amazon CloudFront: A content deliver network (CDN) that securely delivers data, videos, applications and APIs to customers globally.

6. AWS Lambda: A serverless computing services that runs our code in response to events and automatically manages the compute resources for us.

7. Amazon Route S3: A highly available and scalable Domain Name System(DNS) web service that translates domain names into IP addresses.

8. Amazon Elastic Beanstalk: A fully managed service that makes it easy to deploy and scale web applications.

9. Amazon Simple Queue Service(SQS): A fully managed message queuing service that enables to decouple and scale microservices, distributed systems, and serverless applications.

10. Amazon Elastic Container Service(ECS): A fully managed container orchestration service that enables to run scale containerized application on AWS.

11. Amazon Elastic Kubernetes Service(EKS): A fully managed Kubernetes service that makes it easy to deploy, and scale containerized applications using Kubernetes on AWS.

12. Amazon CloudWatch: A monitoring service for AWS resources and the applications run on AWS.

13. Amazon CloudFormation: A service that helps to model and set up AWs resources so that one can spend less time managing those resources and more time focusing on application.

14. AWS CloudTrail: A service that enables governance, compiliance, operational auditing, and risk auditing of AWS account.

3.6 Summary

This chapter describes about Amazon Web Services. Section 3.1 provides basic introduction about AWS. Section 3.1.1 describes cloud computing using AWS. Section 3.1.2 provides characteristics of cloud computing. Section 3.1.3 provides advantages of cloud computing. Section 3.2 points out AWS management console. Section 3.2.1 explains the different regions. Section 3.3 describes availability zones. Section 3.4 describes the different services supported by AWS. Section 3.5 explains tools of AWS.

CHAPTER 04

TASK PERFORMED

4.1 Introduction

AWS is one of the leading cloud computing platforms available today. It provides a wide range of services that can help businesses and individuals manage their IT infrastructure more efficiently and effectively. An Amazon VPC is an isolated portion of the AWS cloud. A virtual private cloud (VPC) is an on-demand configurable pool of shared resources allocated within a public cloud environment, providing a certain level of isolation between the different organizations (denoted as users hereafter) using the resources. The isolation between one VPC user and all other users of the same cloud (other VPC users as well as other public cloud users) is achieved normally through the allocation of a private IP subnet and a virtual communication construct (such as a VLAN or a set of encrypted communication channels) per user.

Amazon VPC to create a virtual network topology for your Amazon EC2 resources. User has complete control over their virtual networking environment, including selection of their own IP address range, creation of subnets, and configuration of route tables and network gateways. Users can create a public-facing subnet for their webservers that has access to the Internet, and place their backend systems such as databases or application servers in a private-facing subnet with no Internet access.

VPC, providing isolation within the cloud, is accompanied by a virtual private network (VPN) function (again, allocated per VPC user) that secures, by means of authentication and encryption, the remote access of the organization to its VPC resources. With the introduction of the described isolation levels, an organization using this service is in effect working on a 'virtually private' cloud (that is, as if the cloud infrastructure is not shared with other users), and hence the name VPC.

A VPC is a virtual network environment that is logically isolated from other networks in the AWS cloud, as well as from the internet. It allows customers to launch Amazon Elastic Compute Cloud (EC2) instances, Amazon Relational Database Service (RDS) instances, and other AWS resources in a virtual network that is dedicated to their organization.

In order to establish a connection between a VPC and an organization's on-premises network, AWS provides several options, including a VPN connection, Direct Connect, or AWS Private Link. These options enable customers to securely connect their VPCs with their on-premises networks and access resources securely and efficiently.

In addition to providing secure connectivity, VPCs also offer several other benefits, including greater control over network configurations, the ability to customize network topologies, and improved security and isolation. Customers can configure VPCs to meet their specific networking requirements and can scale their networks as needed to accommodate changing business needs.

4.2 Motivation

- **Isolation:** VPCs allow customers to create a private network within AWS that is completely isolated from the public internet. This provides a more secure environment for running applications and storing data.

- **Efficiency:** Using AWS services such as AMI, S3, KMS, RDS, and Route 53 can help businesses and individuals manage their IT infrastructure more efficiently.

- **Customization:** With VPCs, customers have complete control over the network topology, including IP addressing, subnets, and routing tables. This enables them to create a network that meets their specific requirements.

- **Security:** VPCs allow customers to define security rules using network access control lists (ACLs) and security groups. These rules can be used to restrict access to resources and protect against unauthorized access.

- **Connectivity:** VPCs can be connected to on-premises data centers or other cloud environments using VPN or Direct Connect. This enables customers to build hybrid architectures that extend their on-premises networks into the cloud.

- **Scalability**: AWS services are designed to be scalable, meaning they can grow with your business or project. This provides a sense of security and stability, knowing that your infrastructure can handle whatever comes your way.

- **Flexibility:** AWS services are highly flexible, meaning they can be tailored to meet your specific needs.

- **Expertise:** AWS services are designed and managed by some of the top experts in the industry. By using these services, you can tap into this expertise and gain a deeper understanding of the latest best practices and techniques for managing IT infrastructure.

4.3 Problem Statement

Aim: To gain agility as well as additional security

Input: Create private (VPC) or public virtual interfaces for AWS.

Processing:

- The instances run in a private, isolated section of the AWS cloud with direct access to the Internet.
- Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to our instances.
- The instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet.
- Only users can connect this private subnet to there corporate data center via an IPsec Virtual Private Network (VPN) tunnel.
- If user enable ssh from the WebServerSG to the DBServerSG they will be able to login from the Server in the Public subnet to the server in the Private subnet.
- It establish private connectivity between AWS and data center, office, or colocation environment.

Output: Creation of multiple public and private networks.

4.4 Objectives

Objectives of the project are:

- To provide high security at the instance and subnet level.
- To reduce downtime and keep the application and workloads available every moment.
- To Create Hybrid Connection.
- To Secure monitor connection, screen traffic and restrict instance access inside virtual network.
- To facilitate the transfer of data.
- To remove the need for an internet gateway or NAT instance to provide S3 access.
- To gain Simple Configuration with multiple security controls.

- To maintain a Bandwidth which is not impacted by the NAT instance.

4.6 Scope of the Project

Scope of Project are:

- Amazon VPC enables to build a virtual network in the AWS cloud - no VPNs, hardware, or physical datacenters required.
- A VPC can span all Availability Zones globally.
- The scope of the VPC connection in AWS depends on the type of connection establish, which can range from connecting two VPCs within the same region to establishing a dedicated, high-speed connection between the VPC .
- Define own network space, and control how network and the Amazon EC2 resources inside the network are exposed to the Internet.
- Leverage the enhanced security options in Amazon VPC to provide more granular access to and from the Amazon EC2 instances in a virtual network.

4.7 Summary

Chapter 4 discusses the task performed section. Section 4.1 gives an introduction to the project topic. The motivation for the project is defined in section 4.2. The problem statement is defined in section 4.3. Section 4.4 describes the objectives of the project. Section 4.5 describes the scope of the project.

CHAPTER 05

REFLECTION NOTES

5.1 Introduction

Amazon VPC provides multiple network connectivity options for you to leverage depending on your current network designs and requirements. These connectivity options include leveraging either the internet or an AWS Direct Connect connection as the network backbone and terminating the connection into either AWS or user-managed network endpoints. Additionally, with AWS, you can choose how network routing is delivered between Amazon VPC and your networks, leveraging either AWS or user-managed network equipment and routes.

Amazon Virtual Private Cloud (Amazon VPC) lets customers provision a private, isolated section of the Amazon Web Services (AWS) Cloud where they can launch AWS resources in a virtual network using customer-defined IP address ranges. Amazon VPC provides customers with several options for connecting their AWS virtual networks with other remote networks. This document describes several common network connectivity options available to our customers. These include connectivity options for integrating remote customer networks with Amazon VPCs and connecting multiple Amazon VPCs into a contiguous virtual network.

To connect VPCs, AWS provides several options, including VPC peering and VPN connections. VPC peering is a networking connection between two VPCs that enables traffic to flow between them as if they were on the same network. VPN connections, on the other hand, use IPsec tunnels to create a secure connection between VPCs and remote networks or data centers.

To set up a VPC connection in AWS, users need to create a VPC and choose the appropriate connectivity option based on their requirements. They then need to configure the network settings, security groups, and routing tables to allow traffic to flow between the VPCs. Finally, they can test the connectivity and troubleshoot any issues that arise.

5.2 Software Installation and Configuration

5.2.1 VPCs And IP Addresses

When you create your VPC, you specify its set of IP addresses with CIDR notation. Classless Inter-Domain Routing (CIDR) notation is a simplified way to show a specific range of IP addresses, Example: 10.0.0.0/16 = all IPs from 10.0.0.0 to 10.0.255.255.

Amazon Virtual Private Cloud (VPC) is a networking service that allows you to create a private, isolated section of the AWS cloud. Within a VPC, you can launch AWS resources such as EC2 instances, RDS databases, and Lambda functions.

Every VPC is assigned a range of IP addresses in the form of a CIDR block. This CIDR block determines the range of IP addresses that can be assigned to resources within the VPC. When you launch an EC2 instance or any other resource in a VPC, you can specify the IP address to be assigned to that resource.

To connect your VPC to other networks or resources, you can create a VPC connection, which is a logical connection between your VPC and another network. There are several types of VPC connections available in AWS, including VPC peering, VPN connections, and AWS Direct Connect.

In a VPC connection, you must ensure that the IP addresses used in the VPC do not overlap with the IP addresses used in the other network. This is to avoid IP address conflicts that can cause network connectivity issues. You can achieve this by carefully selecting the CIDR block for your VPC and the IP addresses used in the other network.

5.2.2 IPs and CIDR

AWS VPCs can use CIDR ranges between /16 and /28. For every one step a CIDR range increases, the total number of IPs is cut in half.

In Amazon Web Services (AWS), a Virtual Private Cloud (VPC) allows you to create a private network within the AWS infrastructure. When setting up a VPC, you need to define IP addresses for the VPC and its subnets.

Internet Protocol (IP) addresses are numerical identifiers assigned to devices that are connected to a network. In AWS, you can use either IPv4 or IPv6 addresses for your VPC. IPv4 addresses are the most commonly used and are represented as four sets of digits separated by periods, for example, 10.0.0.0.

Classless Inter-Domain Routing (CIDR) notation is used to specify the IP address range for a VPC and its subnets. CIDR notation is represented as an IP address followed by a forward slash (/) and a number that indicates the number of bits in the network mask. For example, a CIDR notation of 10.0.0.0/16 specifies that the VPC has an IP address range from 10.0.0.0 to 10.0.255.255, with 16 bits in the network mask.

When connecting a VPC to another network, such as an on-premises network or another VPC, you need to define IP addresses and CIDR ranges for the connected networks to ensure that they can communicate with each other. AWS provides various options for connecting VPCs, including Virtual Private Network (VPN) connections and Direct Connect connections.

5.2.3 Subnets

Subnets are segments or partitions of a network, divided by CIDR range. A VPC with CIDR /22 includes 1,024 total IPs In every subnet, the first four and last one IP addresses are reserved for AWS use.

Amazon Virtual Private Cloud (VPC) allows you to create a logically isolated section within the AWS Cloud where you can launch AWS resources in a virtual network that you define.

Subnets are a way to divide a VPC into smaller networks, which can be used to control network traffic flow, apply security measures, and improve the overall network performance.

In the context of VPC connections in AWS, subnets can be used to partition the resources in your VPC into isolated groups, which can then be connected to other VPCs or on-premises networks through VPC peering, VPN, or AWS Direct Connect.

Each subnet in a VPC is associated with a specific availability zone (AZ) and has its own IP address range. By creating subnets in different AZs, you can achieve high availability and fault tolerance for your applications and services.

5.2.4 Public subnets

Include a routing table entry to an Internet gateway to support inbound/outbound access to the public Internet.

In Amazon Web Services (AWS), Virtual Private Cloud (VPC) is a service that allows you to create a virtual network infrastructure in the cloud. A VPC consists of a set of subnets, which are logical partitions of the VPC's IP address range.

A public subnet in a VPC is a subnet that has a route to the Internet via an Internet Gateway. Instances in a public subnet can have public IP addresses, and they can communicate with the Internet directly.

When you create a VPC, you can choose to create one or more public subnets. You can also connect your VPC to other networks, such as on-premises data centers or other VPCs, using Virtual Private Network (VPN) or AWS Direct Connect.

5.2.5 Private subnets

Do not have a routing table entry to an Internet gateway and are not directly accessible from the public Internet. Typically use a "jump box" (NAT/proxy/bastion host) to support restricted, outbound-only public Internet access.

In Amazon Web Services (AWS), a Virtual Private Cloud (VPC) is a virtual network that can be customized to suit the needs of an organization. Within a VPC, a private subnet is a subset of the VPC's IP address range that is not directly accessible from the internet. Private subnets can be used to isolate resources that should not be publicly accessible, such as databases, application servers, and other internal systems.

When setting up a VPC connection in AWS, private subnets can be used to enhance security by limiting access to resources. This can be achieved by configuring security groups and network access control lists (NACLs) to restrict inbound and outbound traffic to and from the private subnet. Additionally, private subnets can be used to improve performance by enabling resources to communicate with each other more quickly and efficiently, without the need to traverse the public internet.

5.2.6 Route Tables

In Amazon Web Services (AWS), a Virtual Private Cloud (VPC) is a virtual network that can be customized to suit the needs of an organization. Within a VPC, a private subnet is a subset of the VPC's IP address range that is not directly accessible from the internet. Private subnets can be used to isolate resources that should not be publicly accessible, such as databases, application servers, and other internal systems.

When setting up a VPC connection in AWS, private subnets can be used to enhance security by limiting access to resources. This can be achieved by configuring security groups and network access control lists (NACLs) to restrict inbound and outbound traffic to and from the private subnet. Additionally, private subnets can be used to improve performance by enabling resources to communicate with each other more quickly and efficiently, without the need to traverse the public internet.

5.2.7 Network ACLs

NACL (Network Access Control List) is a security feature in AWS (Amazon Web Services) that acts as a firewall for controlling traffic at the subnet level.

When establishing a VPC (Virtual Private Cloud) connection in AWS, NACL plays a critical role in controlling inbound and outbound traffic between subnets. It operates on a rule basis, where each rule contains a specific set of conditions for allowing or denying traffic.

NACL is stateless, meaning that it evaluates each packet independently, whereas Security Groups are stateful and can evaluate traffic flow as a whole. NACL provides an added layer of security to the VPC connection, ensuring that traffic only flows between subnets according to predefined rules.

It is essential to configure NACL correctly to ensure that it does not inadvertently block legitimate traffic. NACL can be associated with subnets, and rules can be configured to allow or deny traffic based on IP addresses, protocols, ports, and other conditions.

5.2.8 Internet gateways

An Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It, therefore, acts as a gateway to the Internet for resources in your VPC.

When you create a VPC in Amazon Web Services (AWS), by default, instances in that VPC are not accessible from the internet. If you want your VPC instances to communicate with the Internet or allow Internet users to connect to your instances, you need to attach an Internet Gateway to your VPC.

To set up a VPC with an Internet Gateway in AWS, you need to create a VPC, subnets, route tables, and then attach an Internet Gateway to your VPC. After the Internet Gateway is attached, you need to configure the route tables to direct internet-bound traffic to the Internet Gateway. This allows instances within the VPC to access the Internet, and public users to access resources within the VPC.

It does not cause availability risks or bandwidth constraints on your network traffic. An Internet Gateway is a network component in AWS that allows traffic to flow in and out of the Virtual Private Cloud (VPC) from the public Internet. It is a gateway that is connected to the VPC to route traffic to and from the internet.

5.3 System Architecture

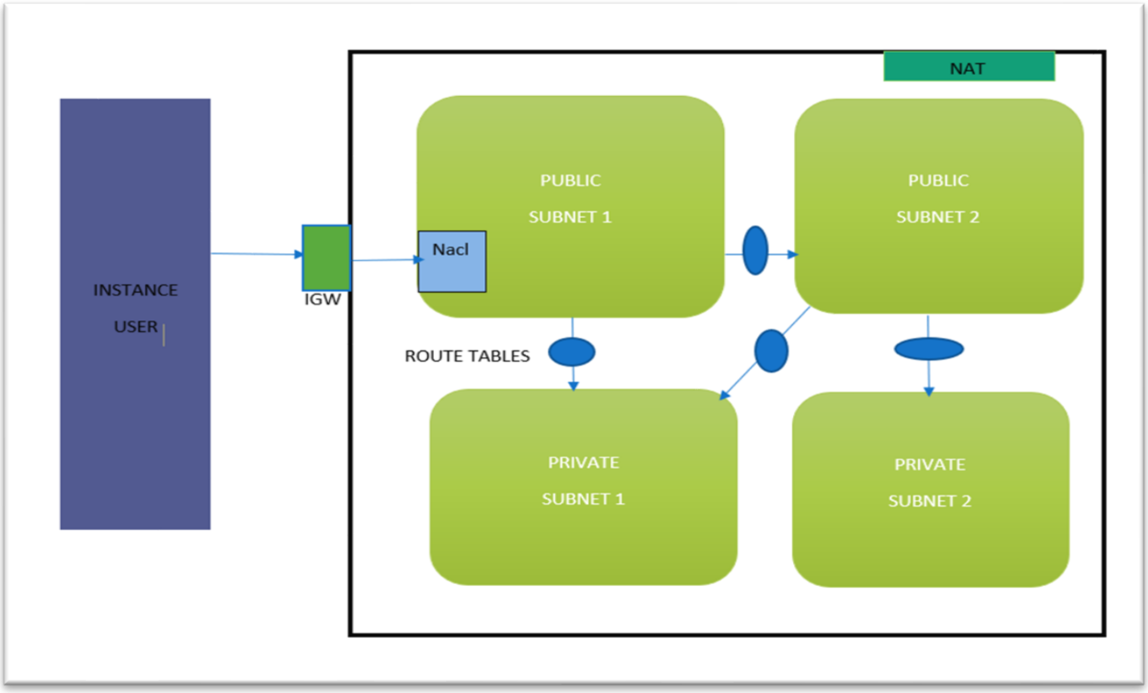


Figure 5.3: System Architecture of VPC Connection

Figure 5.3 shows System Architecture of VPC. Virtual Private Cloud (VPC)is a highly available architecture that spans two to four Availability Zones. A VPC configured with public and private subnets, according to AWS best practices, to provide a virtual network on AWS. The VPC provides Domain Name System (DNS) resolution. In the public subnets Managed network address translation (NAT) gateways to allow outbound internet access for resources in the private subnets. Dedicated custom network access control lists (ACLs) for each Availability Zone. Public subnets all use the same internet gateway as the sole route to communicate with the internet. In the private subnets dedicated custom network ACLs for each Availability Zone. For each private subnet configured to control the flow of traffic within and outside the VPC.

Amazon Web Services (AWS) is a leading cloud service provider that offers a wide range of services to businesses of all sizes. One of the key features of AWS is its Virtual Private Cloud (VPC) service, which enables customers to create and manage isolated virtual networks within the AWS cloud. A VPC is a virtual network environment that is logically isolated from other networks in the AWS cloud, as well as from the internet. It allows customers to launch Amazon Elastic Compute Cloud (EC2) instances, Amazon Relational Database Service (RDS) instances, and other AWS resources in a virtual network that is dedicated to their organization.

In order to establish a connection between a VPC and an organization's on-premises network, AWS provides several options, including a VPN connection, Direct Connect, or AWS PrivateLink. These options enable customers to securely connect their VPCs with their on-premises networks and access resources securely and efficiently. In addition to providing secure connectivity, VPCs also offer several other benefits, including greater control over network configurations, the ability to customize network topologies, and improved security and isolation. Customers can configure VPCs to meet their specific networking requirements and can scale their networks as needed to accommodate changing business needs.

5.4 Algorithm

5.4.1 Create An AWS Learning Account

- To get to the AWS Learning account, log in to https://aws.amazon.com/console/
- Click on Sign In to the Console
- Provide the email address used for signing in the text box Email address of your AWS account
- Provide the a IAM username and Password used while signing in. The default IAM user could be root.
- Click on Sign In•
- Alternatively, you can also sign in using root credentials using the Sign-in using root account credentials• link on the login page.
- If you do not have an AWS account, you can create one by clicking on Create a new AWS account and filling in your details that will be used for signup.

5.4.2 Create An Virtual Private Cloud

- In the AWS Management Console, on the search bar, type VPC, from the drop-down click on VPC.
- In the navigation pane on the left, click Your VPCs on the VPC dashboard.
- Click Create VPC.
- In the Create VPC window use the following:
- Name tag: myawslabVPC IPv4 CIDR block: 10.0.0.0/16
- Ensure Tenancy: Default
- Click Yes, Create

5.4.3 Creation of Public Subnet

- Create a VPC
- Create an Internet Gateway (IGW)
- Create a public subnet
- Choose an Availability Zone
- Create a Route Table
- Launch instances

5.4.4 Creation of Private Subnet

- Create a VPC
- Create a private subnet
- Choose an Availability Zone
- Create a Route Table
- Launch instances

5.4.5 Creation of IGW

- Click on the "Internet Gateways" option in the left-hand navigation pane.
- Click on the "Create Internet Gateway" button.
- Give a new IGW a name and click on the "Create" button.
- Once IGW is created, select it from the list of available IGWs.
- Click on the "Attach to VPC" button and select the VPC to attach the IGW.
- Click on the "Attach" button to complete the attachment process.
- Once the IGW is attached to the VPC.

5.4.6 Creation of NACL

- Open the Amazon VPC console
- In the navigation pane, choose "Network ACLs".
- Choose "Create network ACL".
- In the "Create Network ACL" wizard, give a name to the new NACL and select the VPC where to create it.
- Choose "Create".

- Once the NACL is created, select it from the list and choose "Edit inbound rules" or "Edit outbound rules" to add rules to control traffic.
- Add a new inbound or outbound rule by clicking "Add Rule".
- Specify the rule number, the protocol (TCP, UDP, etc.), the port range, the source or destination IP address, and the action to allow or deny traffic.
- Repeat steps 7 and 8 to add more rules as needed.
- Choose "Save rules".
- associate the NACL with one or more subnets in VPC to control traffic in and out of those subnets. To associate an NACL with a subnet:
- In the VPC console, choose "Subnets".
- Select the subnet to associate the NACL with.
- Choose "Actions", then "Edit subnet associations".
- Select the NACL from the list of available NACLs.
- Choose "Save".

5.4.7 Creation of NACL

- Open the Amazon VPC console.
- Choose "NAT Gateways" in the navigation pane.
- Choose "Create NAT Gateway".
- On the "Create NAT Gateway" page, select the VPC to create a NAT gateway.
- Choose an availability zone for the NAT gateway.
- Select an existing Elastic IP address, or create a new one.
- Choose "Create a new EIP" or "Use existing EIP".
- Select the subnet(s) to create a NAT gateway.
- Choose "Create NAT Gateway".
- Wait for the NAT gateway to be created.
- Once the NAT gateway is created, you must update the route table for the private subnet(s) to use the NAT gateway as the default route.

5.4.8 Creation of The Route table

- In the navigation pane on the left, click Route Tables
- Select the route table that is associated with main AWS lab VPC
- Click the Routes tab.

- Click the Subnets Associations tab.
- Click the Create Route Table.
- VPC: Select my aws lab VPC from the drop down
- Click Yes, Create.

5.4.9 Connecting the Route table To Public Subnets And IGW

- Select the Route Table we just created.
- Click on the Routes tab.
- Click on Add another route.
- Click the Destination field and enter 0.0.0.0/0
- Click the Target field. Auto-complete will display the name of the InternetGateway that you created earlier.
- Click the Internet Gateway that you created (myawslabIGW)
- Click Save
- Click Subnet Associations.
- Click Edit.
- Select myawslabPublic subnet
- Click Save.

5.4.10 Connecting Route to Private Subnet And NAT

- Select the Route Table we just created.
- Click on the Routes tab.
- Click on Add another route.
- Click the Destination field and enter 0.0.0.0/0
- Click the Target field. Auto-complete will display the name of the NAT that you created earlier.
- Click the Internet Gateway that you created (myawslabIGW)
- Click Save
- Click Subnet Associations.
- Click Edit.
- Select myawslabPrivate subnet
- Click Save.

5.5 Summary

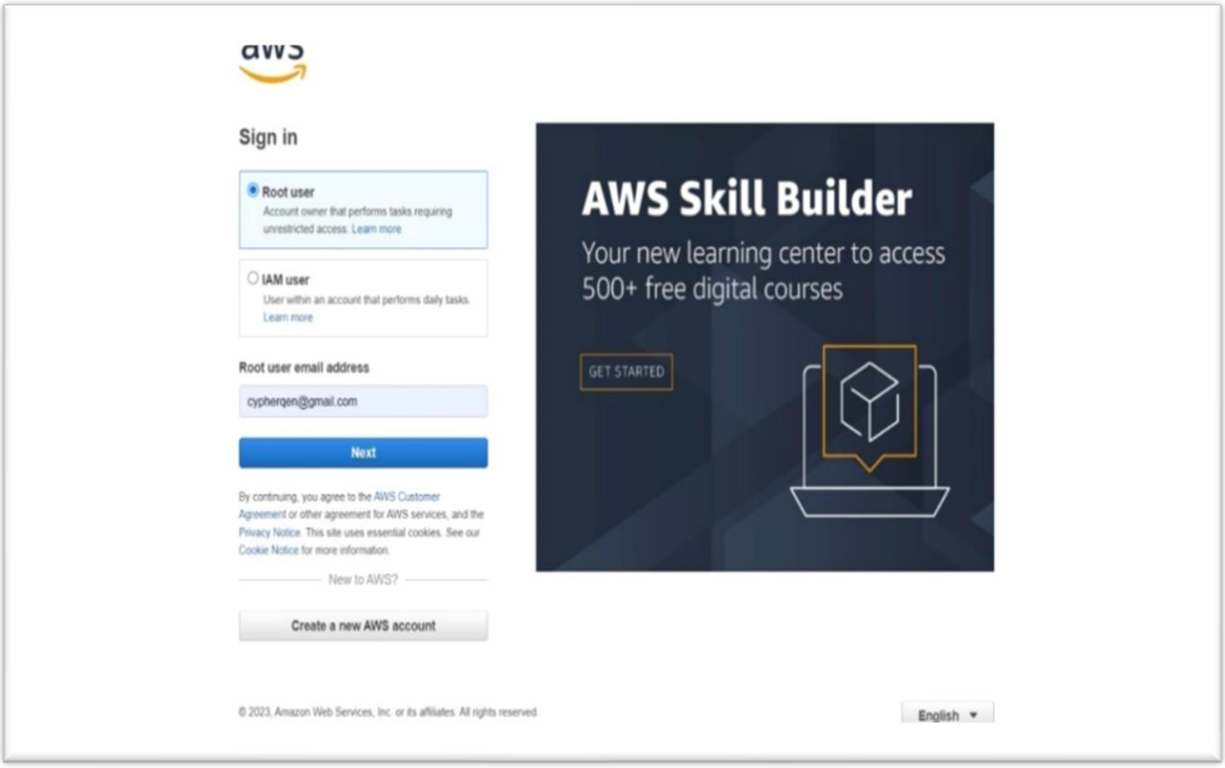
Chapter 5 gives the overall view of the reflection notes. Whereas section 5.1 gives the introduction of the reflection notes. Then section 5.2 explains the software installation and configuration. Section 5.3 gives an overview of the system architecture. Section 5.4 gives the details of the algorithm implemented in the project.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Results

6.1.1 AWS Login Page



Sign in

● Root user
Account owner that performs tasks requiring unrestricted access. Learn more

○ IAM user
User within an account that performs daily tasks. Learn more

Root user email address: cypherqen@gmail.com

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

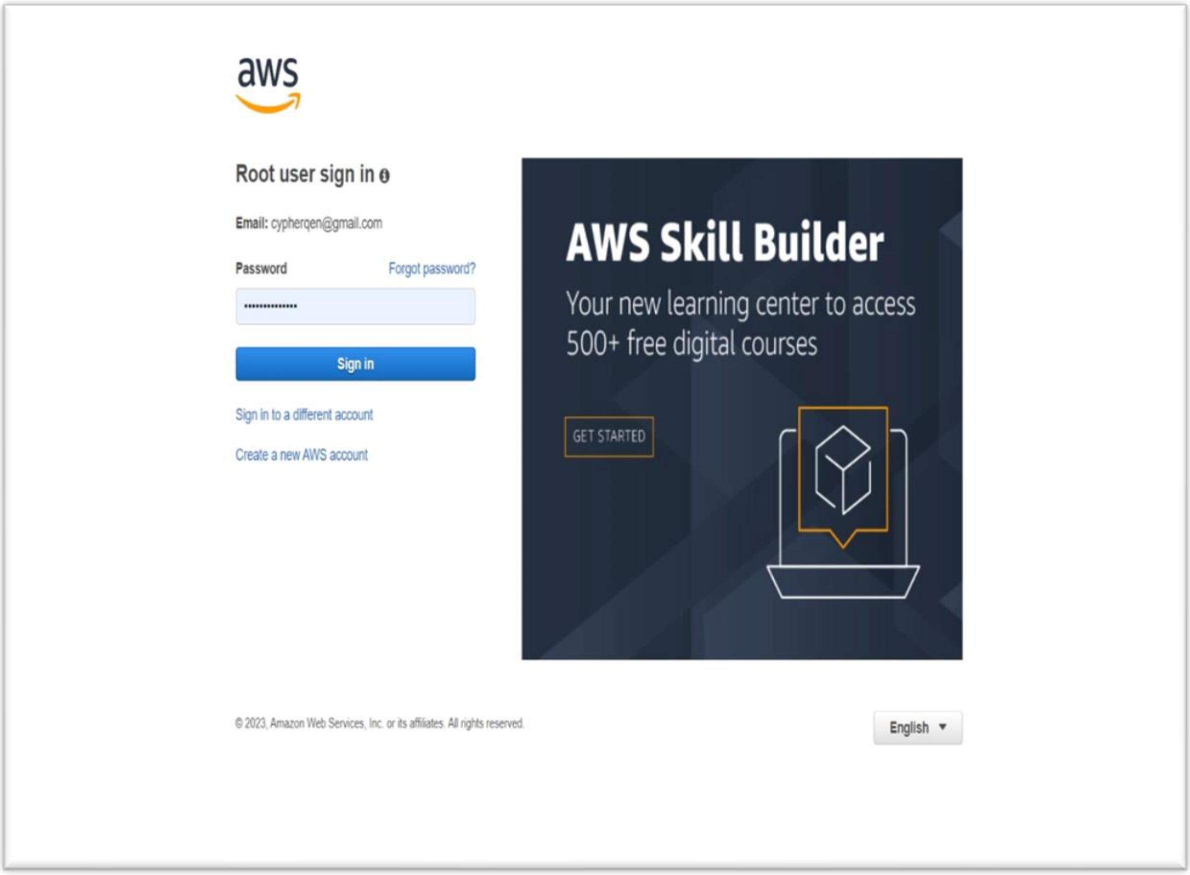
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Snapshot 6.1: AWS login page

Snapshot 6.1 tells about login page to create VPC connection. Amazon Virtual Private Cloud is a commercial cloud computing service that provides users with a virtual private cloud. A logically isolated section of Amazon Web Services Cloud. 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 like EC2 instance Databases.

The AWS login page is the portal where users can securely access their Amazon Web Services (AWS) account. The login page is accessible through the AWS website and requires users to enter their email address or AWS account ID and password to gain access. Once logged in, users can manage their cloud resources, configure security settings, and access other AWS services. The login page also provides options for users to reset their password or create a new account if they are new to AWS. Security features such as multi-factor authentication can also be enabled to provide an additional layer of protection to the login process.

6.1.2 Root User Sign in Page



Root user sign in

Email: cypherqen@gmail.com

Password

••••••••••••••

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws
Forgot password?
Sign in
Sign in to a different account
Create a new AWS account
AWS Skill Builder
Your new learning center to access 500+ free digital courses
GET STARTED
English ▾

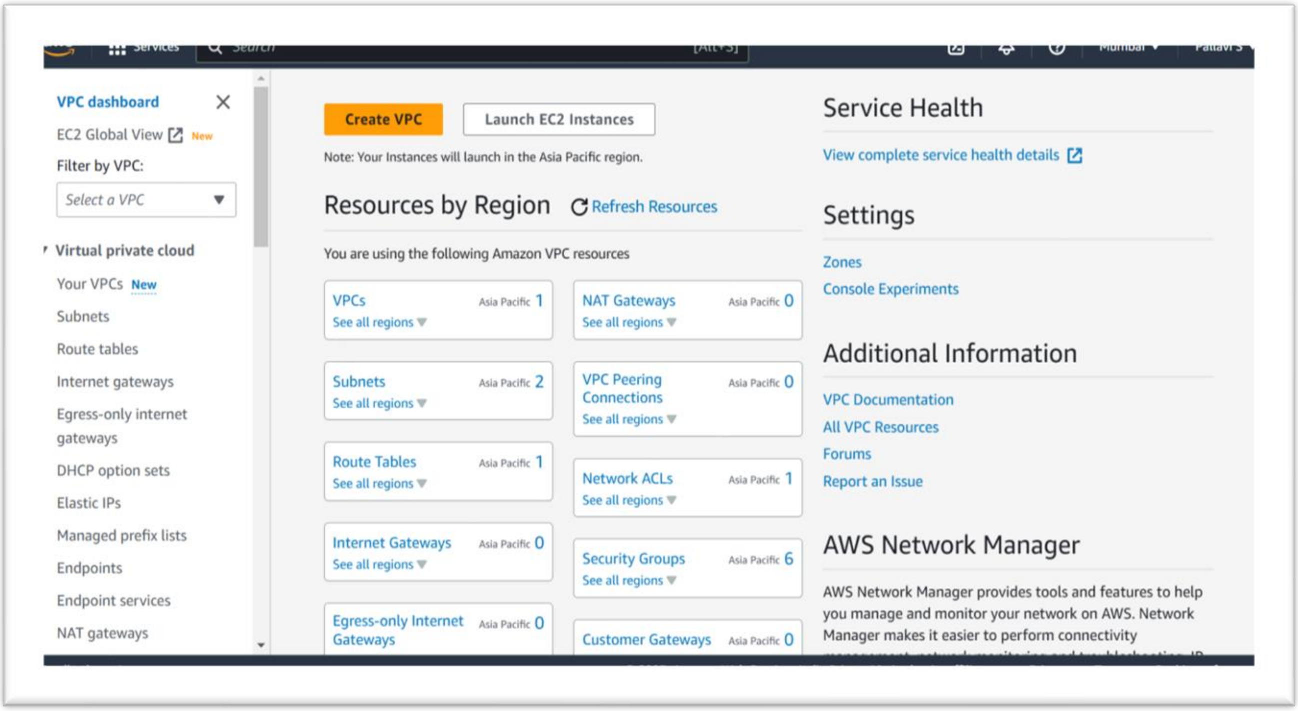
Snapshot 6.2: Root user sign in page

Snapshot 6.2 briefs about the root user sign in page of AWS. After the login page, the sign in page will be visible where user have to give the required credentials. Unique password is preferable. A sign-in page is a webpage that allows users to enter their credentials, such as a username and password, to access a secure system or service. It is commonly used by websites, online applications, and other digital services to authenticate users and ensure that only authorized users can access protected information or perform certain actions.

A typical sign-in page usually includes a form where users can enter their login credentials. The form may also include additional features such as password strength indicators, password reset options, and multi-factor authentication. The sign-in page may also display legal disclaimers, privacy policies, or other terms of service.

To use a sign-in page, users typically navigate to the URL or click on a link provided by the service they want to access. Once on the sign-in page, users enter their login credentials and click the "Sign In" or "Log In" button to access the protected system or service.

6.1.3 AWS Menu Page


VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Create VPC
Launch EC2 Instances
Resources by Region
Refresh Resources
VPCs Asia Pacific 1 See all regions
NAT Gateways Asia Pacific 0 See all regions
Subnets Asia Pacific 2 See all regions
VPC Peering Connections Asia Pacific 0 See all regions
Route Tables Asia Pacific 1 See all regions
Network ACLs Asia Pacific 1 See all regions
Internet Gateways Asia Pacific 0 See all regions
Security Groups Asia Pacific 6 See all regions
Egress-only Internet Gateways Asia Pacific 0
Customer Gateways Asia Pacific 0
Service Health
View complete service health details
Settings
Zones
Console Experiments
Additional Information
VPC Documentation
All VPC Resources
Forums
Report an Issue
AWS Network Manager

Note: Your Instances will launch in the Asia Pacific region.

You are using the following Amazon VPC resources

AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity

Snapshot 6.3 Menu page

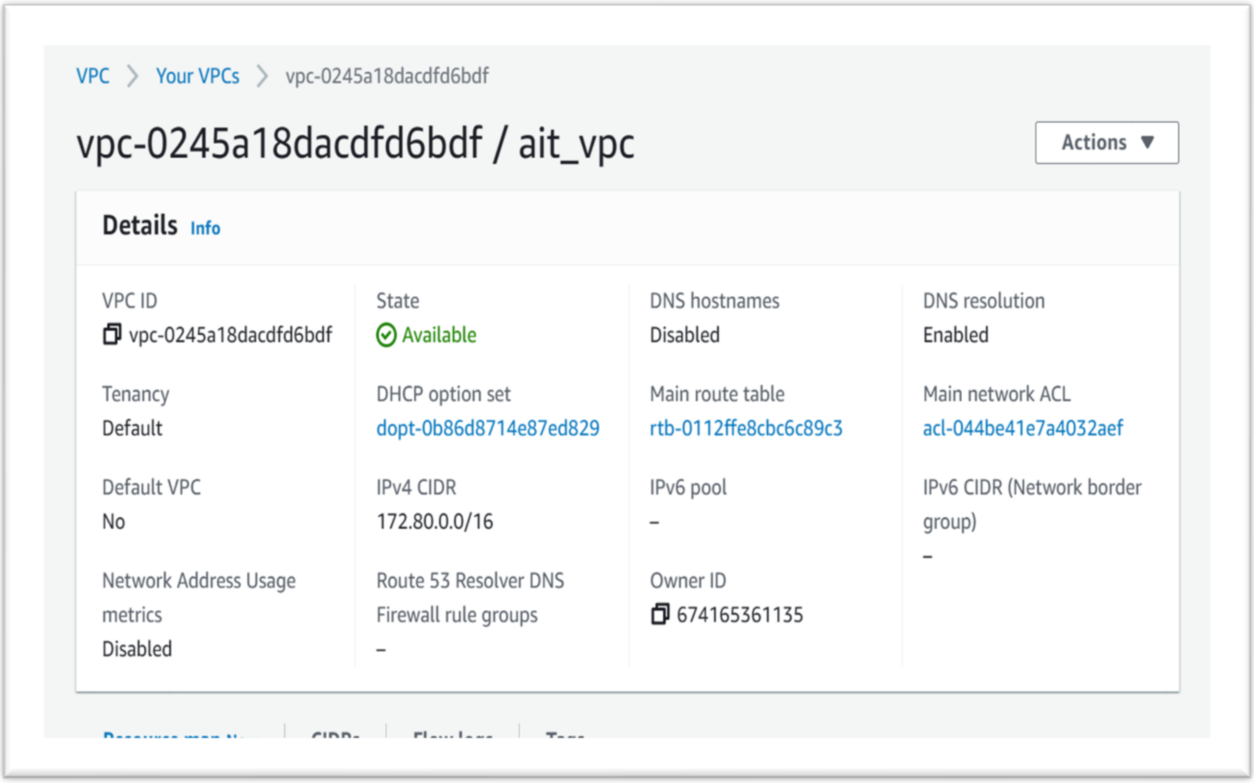
In Snapshot 6.3, it shows how to create menu page. Then log in to Amazon account, just search for VPC in the search bar and we'll see the following screen. There click on the CREATE VPC button to get our virtual private cloud connection. A menu page in an AWS VPC (Virtual Private Cloud) is a graphical interface that provides users with easy access to various resources and services within the VPC environment. The menu page typically presents a list of available services, tools, and features that can be used to manage the VPC. The menu is organized into sections that correspond to different aspects of the VPC, including networking, security, instances, storage, monitoring, and administration.

One of the key sections on the menu page is Networking, which allows users to configure and manage the VPC's network settings. Users can set up and manage subnets, create and modify route tables, and configure internet gateways to provide connectivity to the internet. They can also set up and manage virtual private gateways to connect the VPC to on-premises networks.

The Security section of the menu page provides tools for managing the security settings of the VPC. Users can create and manage security groups and network access control lists (NACLs) to control traffic in and out of the VPC. They can also set up and manage virtual private network (VPN) connections and AWS Direct Connect to provide secure access to on-premises resources.

The Storage section of the menu page provides tools for managing the storage resources of the VPC, including Elastic Block Store (EBS) volumes, S3 buckets, and Glacier archives. Users can create, manage, and delete storage resources, as well as configure storage settings and access policies.

6.1.4 Viewing of Created VPC



VPC > Your VPCs > vpc-0245a18dacdfd6bdf

vpc-0245a18dacdfd6bdf / ait_vpc

Details Info

VPC ID vpc-0245a18dacdfd6bdf	State ✓ Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0b86d8714e87ed829	Main route table rtb-0112ffe8cbc6c89c3	Main network ACL acl-044be41e7a4032aef
Default VPC No	IPv4 CIDR 172.80.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 674165361135	

Snapshot 6.4 Viewing of created VPC

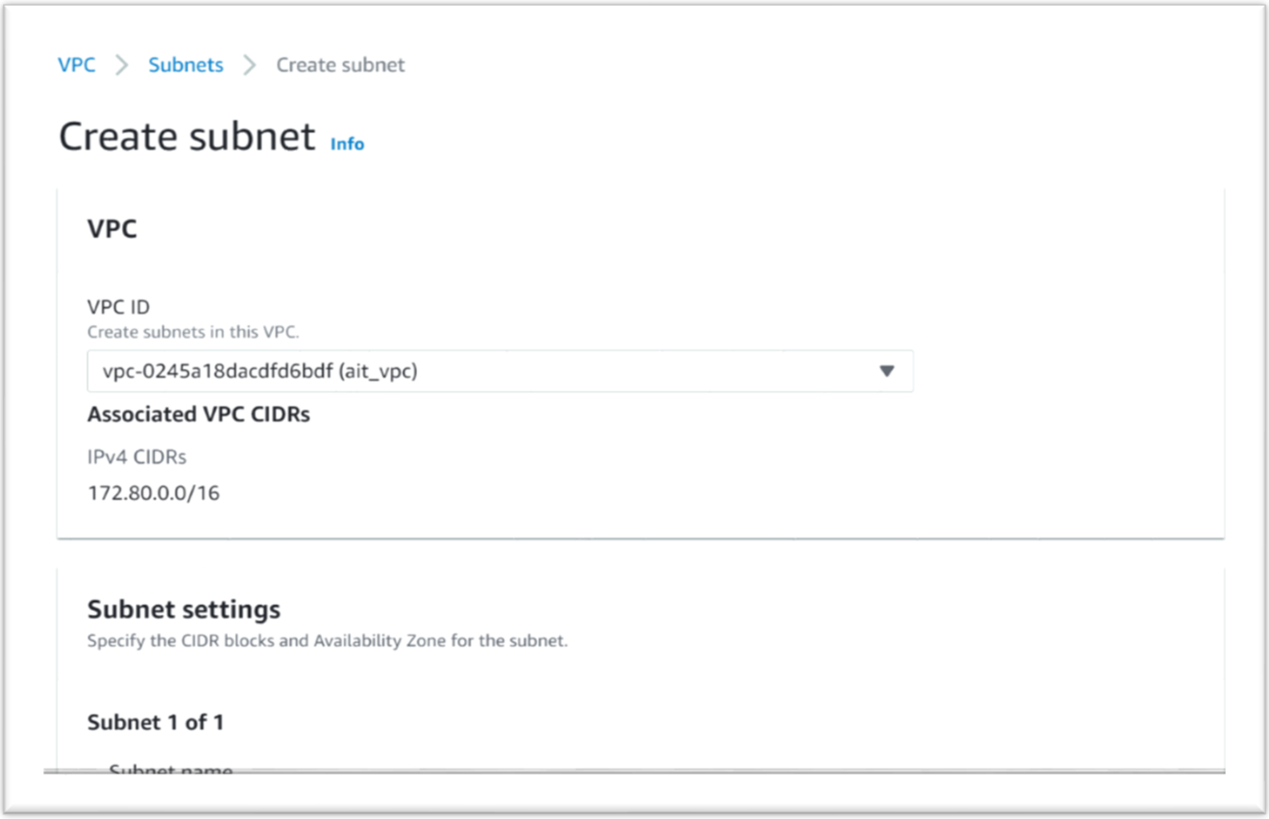
Snapshot 6.4 view about the creating the VPC in AWS. Created VPC can see in the above figure with information such as VPC id, state of availability, DNS hostname, IPV4 CIDRA, Owner ID etc. When you create a VPC (Virtual Private Cloud) in AWS (Amazon Web Services), you are essentially creating your own private network in the cloud. A VPC allows you to define a virtual network topology that is isolated from other VPCs and the public internet. With a VPC, you can launch AWS resources like EC2 instances, RDS databases, and Elastic Load Balancers within your own virtual network.

To create a VPC in AWS, you must define a CIDR (Classless Inter-Domain Routing) block that represents the IP address range of your VPC. You can also configure additional settings like the number of subnets and the size of the subnets within your VPC. Once your VPC is created, you can launch resources within your VPC and control access to them using security groups and network access control lists (NACLs).

When you create a VPC in AWS, you have complete control over the network topology and security of your VPC. You can customize your VPC by adding and configuring subnets, internet gateways, virtual private gateways, and other network resources. You can also use AWS services like Route 53 and AWS Private Link to securely connect your VPC to other AWS services and on-premises resources.

Overall, creating a VPC in AWS provides you with a powerful way to manage and secure your cloud resources. With a VPC, it can isolate your resources from the public internet and other VPCs, and you can define your own network topology and security policies. This makes it easier to manage and secure your cloud resources, while also providing greater flexibility and scalability.

6.1.5 Creation of Subnet



VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0245a18dacdfd6bdf (ait_vpc) ▼

Associated VPC CIDRs

IPv4 CIDRs
172.80.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

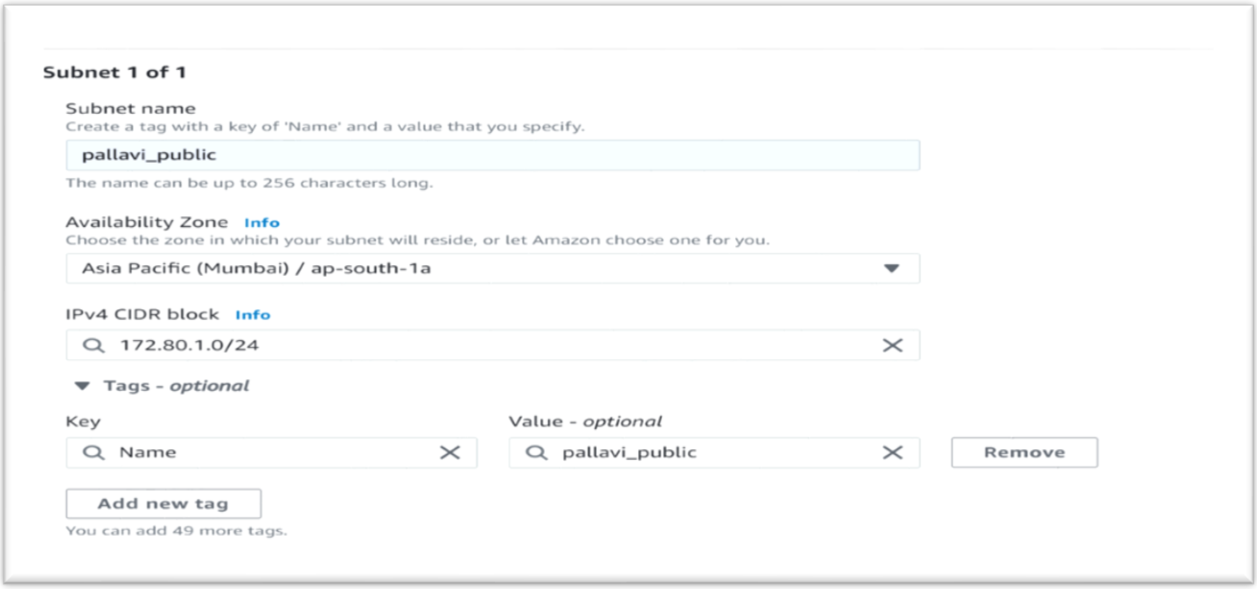
Subnet 1 of 1

Subnet name

Snapshot 6.5: Creating the subnet

Snapshot 6.5 shows the creation of the subnet. A subnet is a range of IP addresses in your VPC. Each VPC network consists of one or more IP address range called subnets. Subnets are regional resources, and have IP address ranges associated with them. Subnets make networks more efficient.

6.1.6 Creating the Public subnet with unique IP address



Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pallavi_public
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.
Asia Pacific (Mumbai) / ap-south-1a

IPv4 CIDR block Info
172.80.1.0/24

▼ Tags - *optional*

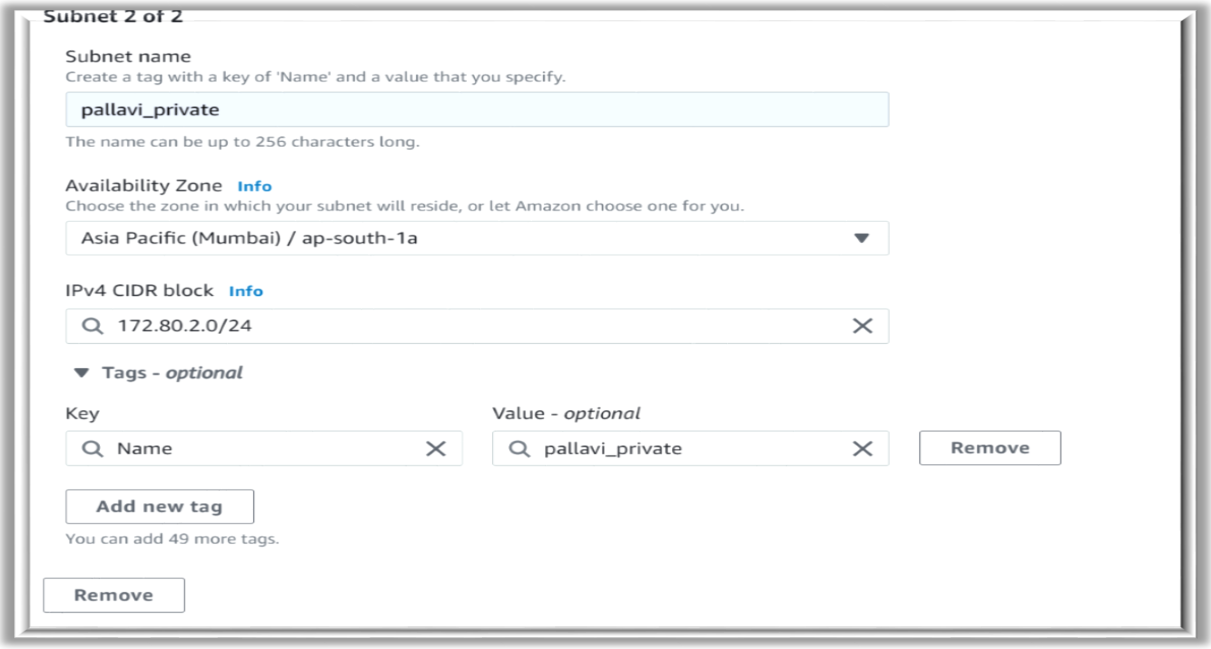
Key	Value - *optional*	
Name	pallavi_public	Remove

Add new tag
You can add 49 more tags.

Snapshot 6.6: Creating the Public subnet with unique IP address

In Snapshot 6.6 shows how the subnets are created. The public subnet can set up security and routing so that the web servers can communicate with the database servers. The instances in the public subnet can send outbound traffic directly to the internet. A public subnet is a subnet that's associated with a route table that has a route to an internet gateway.

6.1.7 Creating the Private Subnet with unique IP address



Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pallavi_private
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.
Asia Pacific (Mumbai) / ap-south-1a

IPv4 CIDR block Info
172.80.2.0/24

▼ Tags - *optional*

Key	Value - *optional*	
Name	pallavi_private	Remove

Add new tag
You can add 49 more tags.

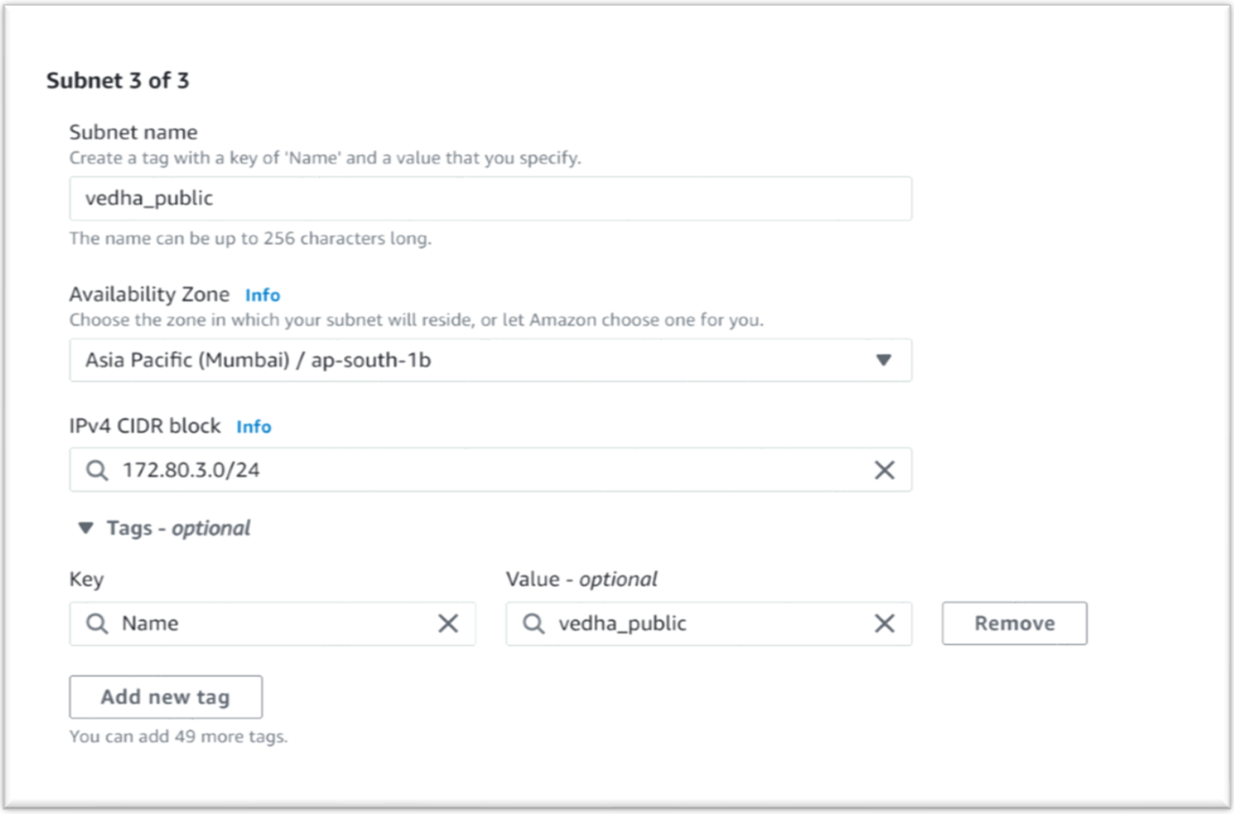
Remove

Snapshot 6.7: creating the Private Subnet with unique IP address

Snapshot 6.7 shows creating private subnet. Private Subnet is a range of Ip address which cannot connect to the internet. A private subnet is a subnet that is associated with a route table that doesn't have a route to an internet gateway. The instances in the private subnet can access the internet by using a network address translation (NAT) gateway that resides in the public subnet.

When setting up a VPC connection in AWS, private subnets can be used to enhance security by limiting access to resources. This can be achieved by configuring security groups and network access control lists (NACLs) to restrict inbound and outbound traffic to and from the private subnet. Additionally, private subnets can be used to improve performance by enabling resources to communicate with each other more quickly and efficiently, without the need to traverse the public internet.

6.1.8 Creating the Public subnet with Unique IP address



Subnet 3 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
vedha_public
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.
Asia Pacific (Mumbai) / ap-south-1b

IPv4 CIDR block Info
172.80.3.0/24

▼ Tags - *optional*

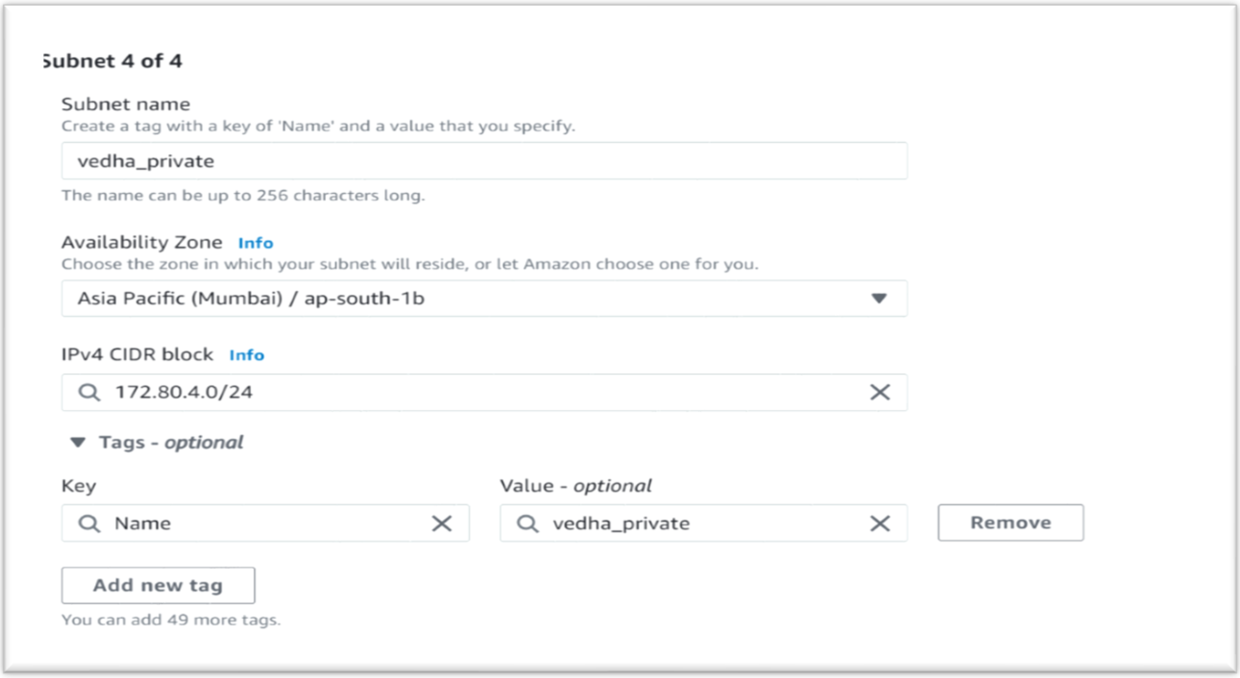
Key	Value - *optional*	
Name	vedha_public	Remove

Add new tag
You can add 49 more tags.

Snapshot 6.8: Creating another Public Subnet

Snapshot 6.8 shows the multiple public subnets. The Public subnet can set up security and routing so that the web servers can communicate with the database servers. The instances in the public subnet can send outbound traffic directly to the internet. A public subnet is a subnet that's associated with a route table that has a route to an internet gateway. Include a routing table entry to an Internet gateway to support inbound/outbound access to the public Internet.

6.1.9 Creating the Private Subnet with unique IP address



Subnet 4 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
vedha_private
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.
Asia Pacific (Mumbai) / ap-south-1b

IPv4 CIDR block Info
172.80.4.0/24

Tags - *optional*

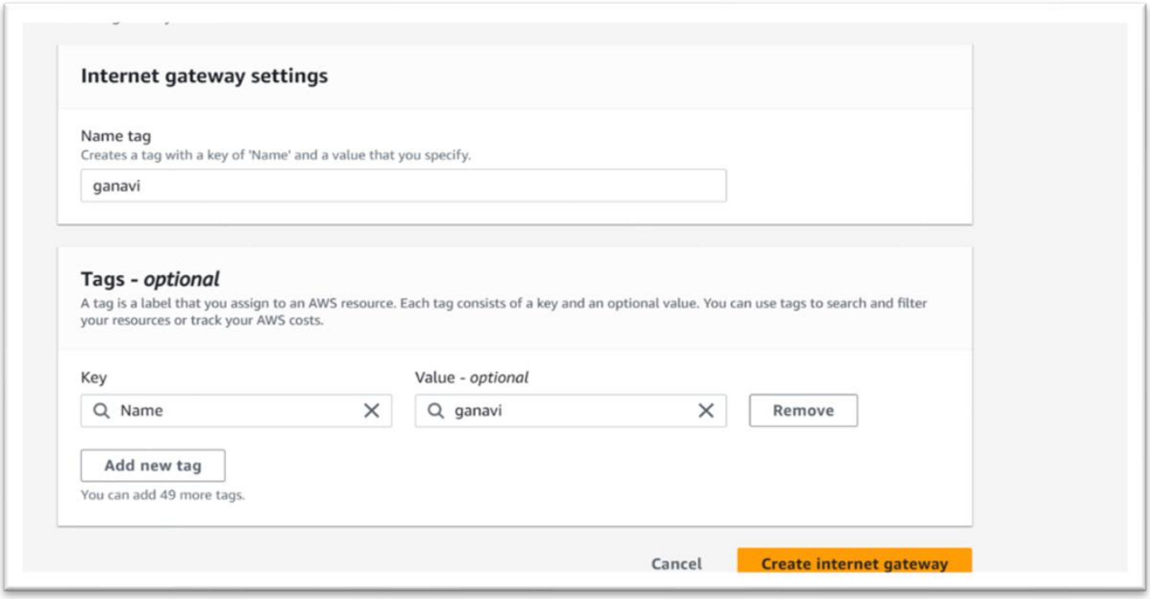
Key	Value - *optional*
Name	vedha_private

You can add 49 more tags.

Snapshot 6.9: Creating another Private subnet

Snapshot 6.9 shows about the creation of multiple available of private subnets. Private Subnet is a range of IP address which cannot connect to the internet. A private subnet is a subnet that is associated with a route table that doesn't have a route to an internet gateway. The instances in the private subnet can access the internet by using a network addresstranslation (NAT) gateway that resides in the public subnet.

6.1.10 Creation of IGW



Internet gateway settings

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

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	ganavi

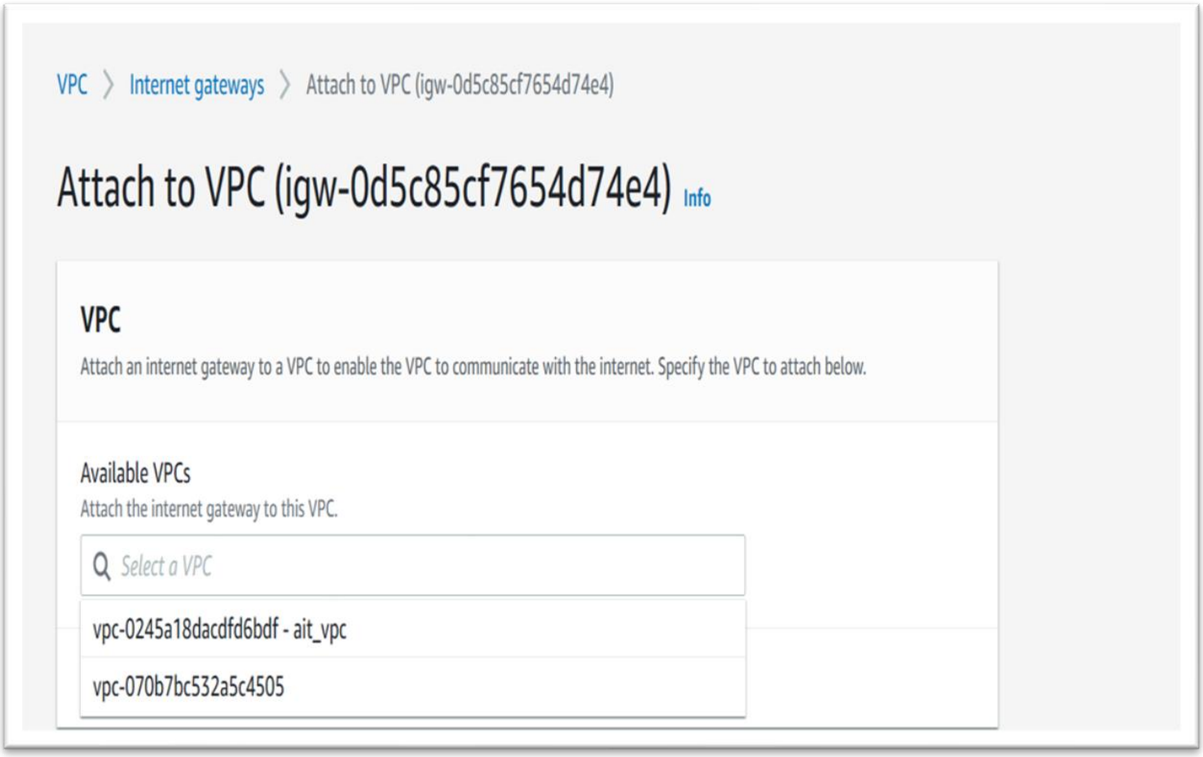
You can add 49 more tags.

Snapshot 6.10 Creation of Internet Gateway

Snapshot 6.10 shows the creation of an Internet Gateway (IGW) that allows communication between VPC and the Internet and provides a target to VPC route tables for Internet-routable traffic. It does not cause availability risks or bandwidth constraints on your network traffic. An Internet Gateway is a network component in AWS that allows traffic to flow in and out of the Virtual Private Cloud (VPC) from the public Internet. It is a gateway that is connected to the VPC to route traffic to and from the internet.

When you create a VPC in Amazon Web Services (AWS), by default, instances in that VPC are not accessible from the internet. If you want your VPC instances to communicate with the internet or allow internet users to connect to your instances, you need to attach an Internet Gateway to your VPC.

6.1.11 Attaching IGW to VPC



VPC > Internet gateways > Attach to VPC (igw-0d5c85cf7654d74e4)

Attach to VPC (igw-0d5c85cf7654d74e4)

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.

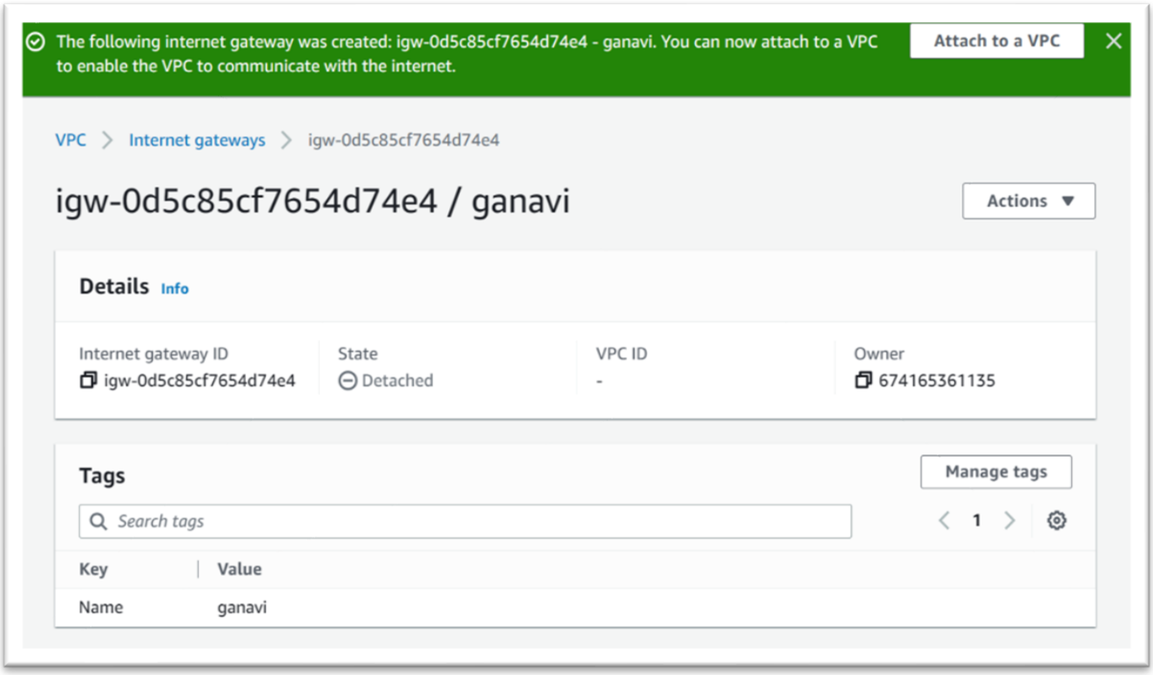
vpc-0245a18dacdfd6bdf - ait_vpc

vpc-070b7bc532a5c4505

Snapshot 6.11 Attaching IGW to VPC

Snapshot 6.11 shows attaching an internet gateway to a VPC, enabling connectivity between the internet and the VPC. It performs network address translation (NAT) for instances that have NOT been assigned public IP addresses. Attaching an Internet Gateway (IGW) to a Virtual Private Cloud (VPC) in Amazon Web Services (AWS) is essential for enabling communication between resources in the VPC and the Internet. The IGW acts as a gateway for traffic to and from the internet and the VPC.

6.1.12 Overview of IGW Creation



The following internet gateway was created: igw-0d5c85cf7654d74e4 - ganavi. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-0d5c85cf7654d74e4 / ganavi

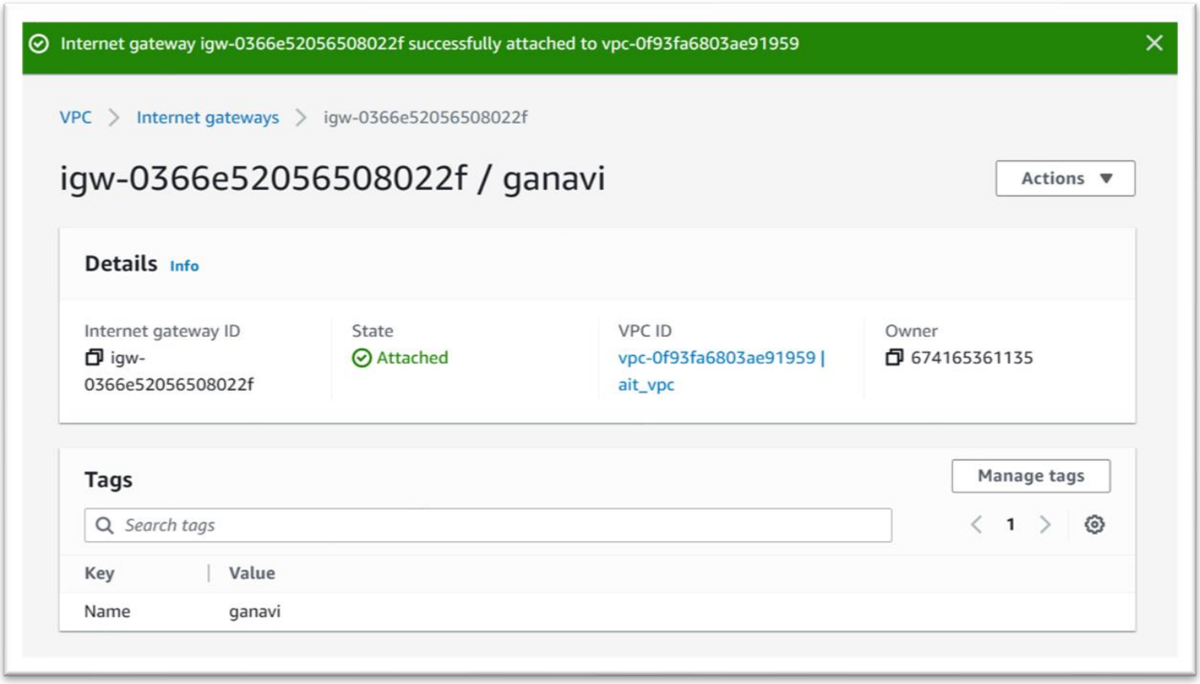
Internet gateway ID	State	VPC ID	Owner
igw-0d5c85cf7654d74e4	Detached	-	674165361135

Key	Value
Name	ganavi

Snapshot 6.12 Overview of IGW

Snapshot 6.12 shows the Overview of IGW creation. To access the Internet from within a VPC, the VPC must be connected to the Internet Gateway, which requires the attachment of a VPC Internet Gateway to the VPC. Once the VPC is attached to the Internet Gateway, instances within the VPC can have public IP addresses assigned to them and can communicate with the Internet.

6.1.13 Overview of Attaching IGW to VPC



Internet gateway igw-0366e52056508022f successfully attached to vpc-0f93fa6803ae91959

igw-0366e52056508022f / ganavi

Internet gateway ID	State	VPC ID	Owner
igw-0366e52056508022f	Attached	vpc-0f93fa6803ae91959 | ait_vpc	674165361135

Key	Value
Name	ganavi

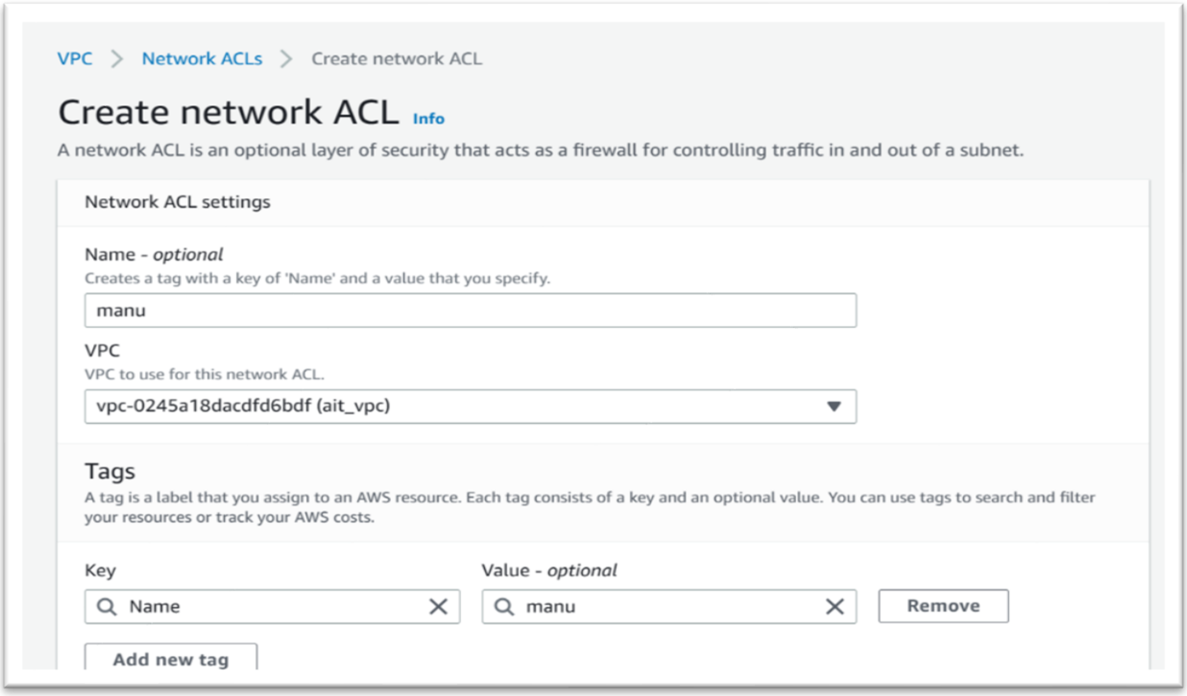
Snapshot 6.13 Overview of Attaching IGW to VPC

Snapshot 6.13 shows attaching an Internet gateway to VPC that allows communication between VPC and the Internet provides a target to VPC route tables for internet-routable traffic.

When you attach an IGW (Internet Gateway) to a VPC (Virtual Private Cloud) in AWS (Amazon Web Services), you enable your VPC to communicate with the public internet. An IGW is a horizontally scaled, redundant, and highly available gateway that enables communication between instances in your VPC and the internet. Once attached, your VPC can access the internet and resources hosted on the internet, and resources hosted on the internet can access resources within your VPC.

To attach an IGW to a VPC in AWS, you must first create an IGW and then attach it to your VPC using the VPC console or the AWS CLI (Command Line Interface). Once attached, you can configure the routing tables of your subnets to route traffic through the IGW. This allows instances within your VPC to communicate with resources outside of your VPC, such as web servers, databases, and other cloud resources.

6.1.14 Creation of NACL



Create network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Network ACL settings

Name - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
manu

VPC
VPC to use for this network ACL.
vpc-0245a18dacdfd6bdf (ait_vpc)

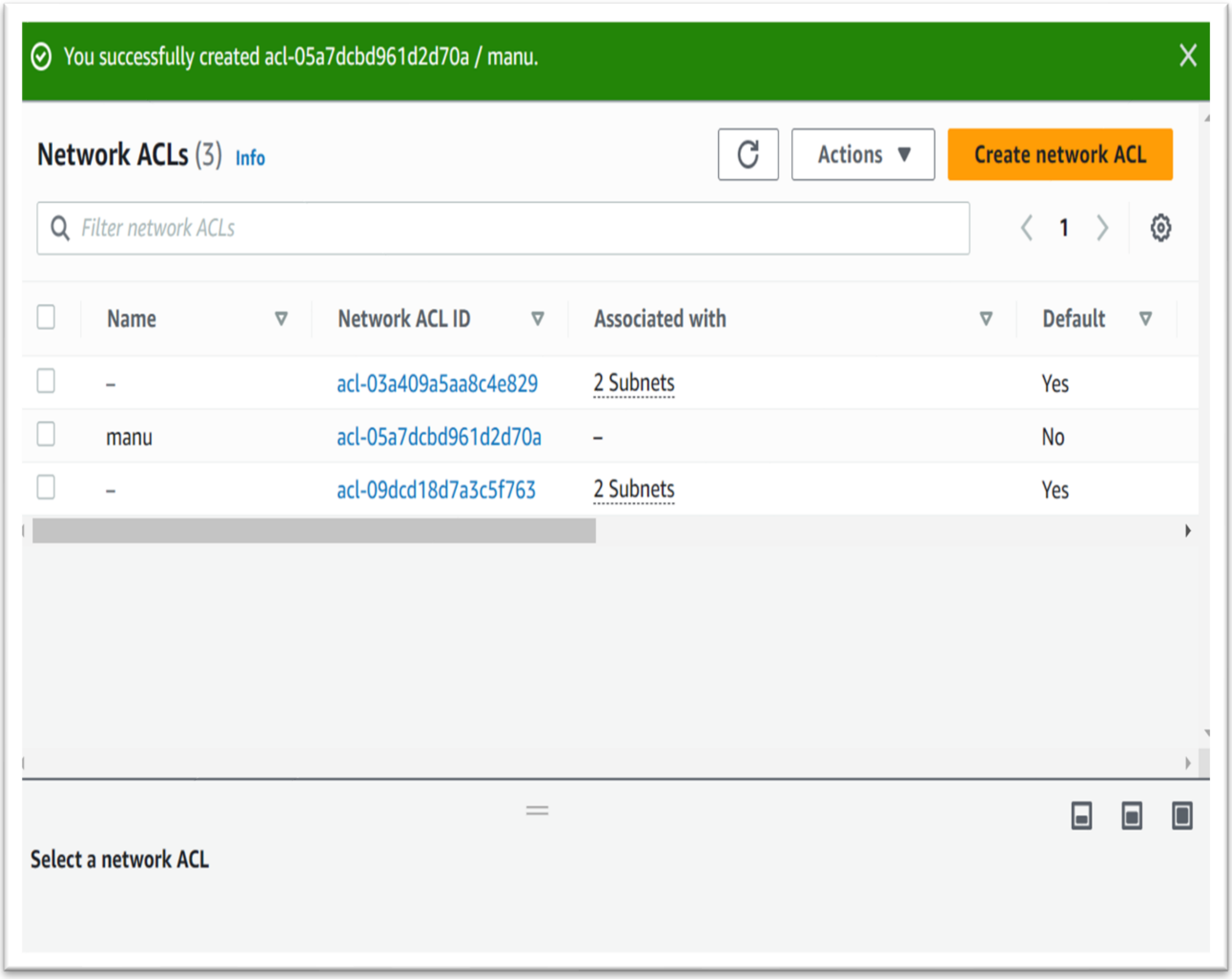
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	manu

Snapshot 6.14 Creation of NACL

Snapshot 6.14 shows the creation of NACL acts as an extra layer of protection applied at the subnet level and it can remove the traffic burden from the subnet, improving security and optimizing performance. NACLs are used to control access to network resources. They sit on subnets and evaluate traffic based on set rules, then determine whether or not that traffic should be allowed to continue.

6.1.15 Overview of Creation of NACL



You successfully created acl-05a7dcbd961d2d70a / manu.

Network ACLs (3) Info

□	Name	Network ACL ID	Associated with	Default
□	-	acl-03a409a5aa8c4e829	2 Subnets	Yes
□	manu	acl-05a7dcbd961d2d70a	-	No
□	-	acl-09dcd18d7a3c5f763	2 Subnets	Yes

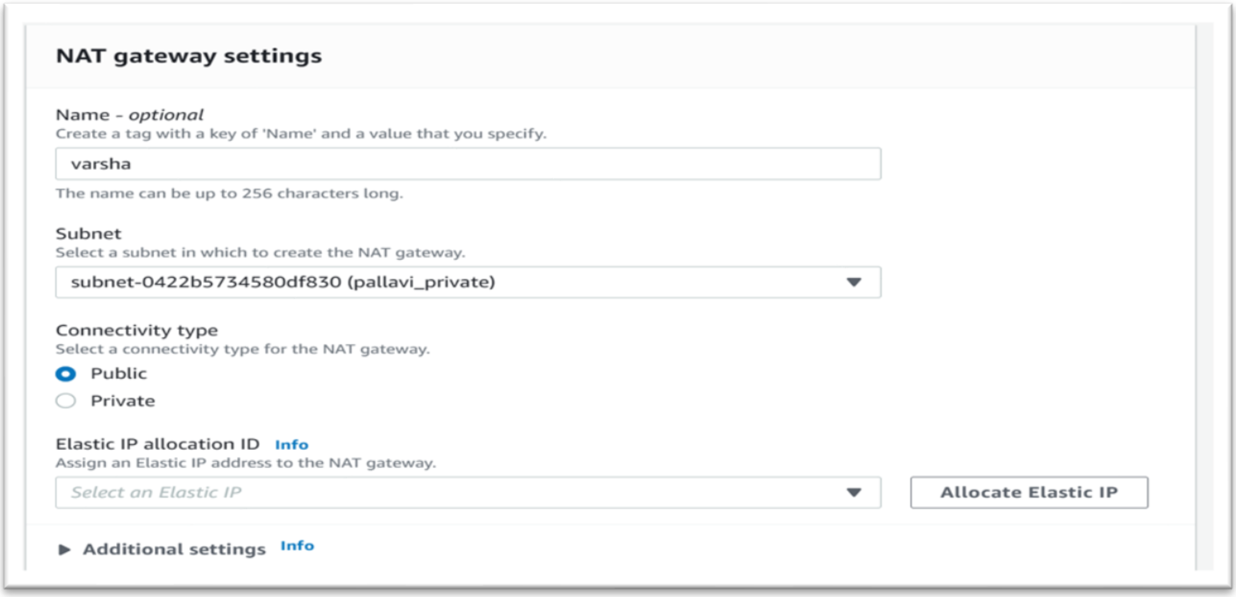
Select a network ACL

Snapshot 6.15 Overview of creation of NACL

Snapshot 6.15 shows Overview of creation of NACL. Network ACL is associated with both inbound and outbound rules that can either deny or allow the rules. A Network ACL contains numbered lists of rules that are evaluated in order, starting from the lowest numbered rule, to determine whether the traffic goes in or out of the subnet associated with the Network ACL.

Each subnet is associated with a Network Access Control List, which contains a set of rules that govern inbound and outbound traffic to and from the subnet. NACLs are stateless, meaning that they evaluate each packet in isolation, rather than considering the context of previous packets in a session.

6.1.16 Creation of NAT Gateway



NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
varsha
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-0422b5734580df830 (pallavi_private)

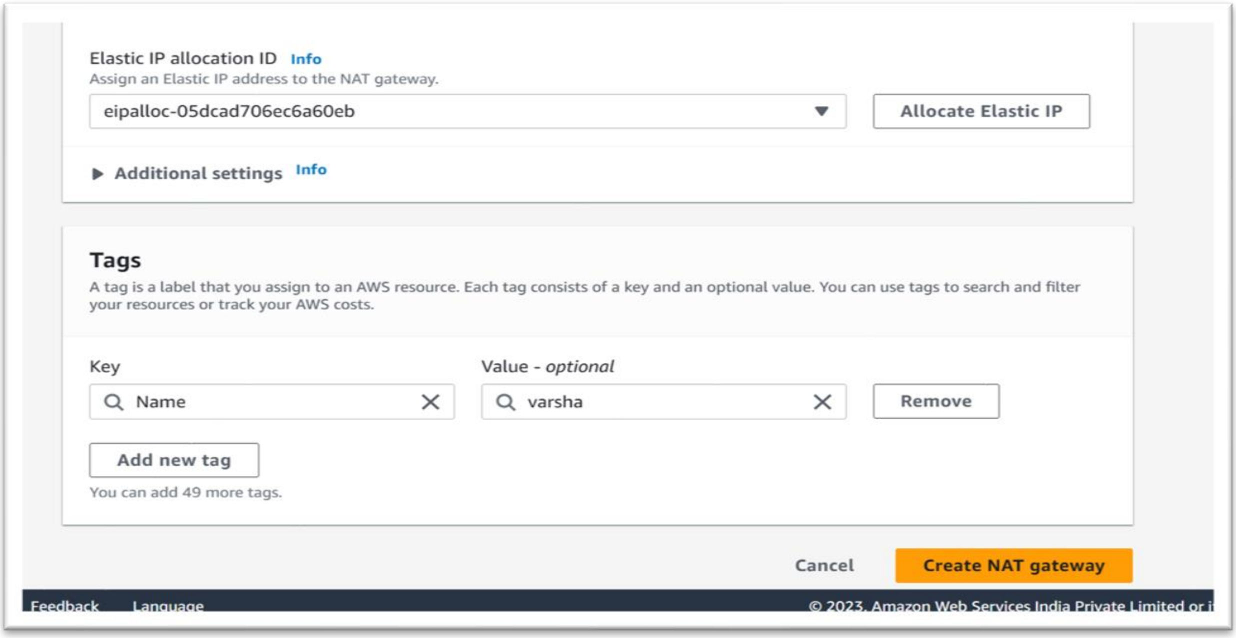
Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
Select an Elastic IP

Snapshot 6.16 Creating NAT Gateway by selecting the private subnet

Snapshot 6.16 shows Creating NAT Gateway When a resource in a private subnet within a VPC needs to access the internet, it requires a public IP address. However, AWS does not provide a public IP address to resources in a private subnet. NAT gateway that enables instances in a private subnet to connect to the internet or other AWS services. A managed service that provides a highly available, scalable, and secure way to provide outbound internet access to resources in a private subnet.

6.1.17 Creation of NAT Gateway by allocating Elastic IP



Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-05dcad706ec6a60eb

Allocate Elastic IP
▶ Additional settings Info

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: Name
Value - *optional*: varsha

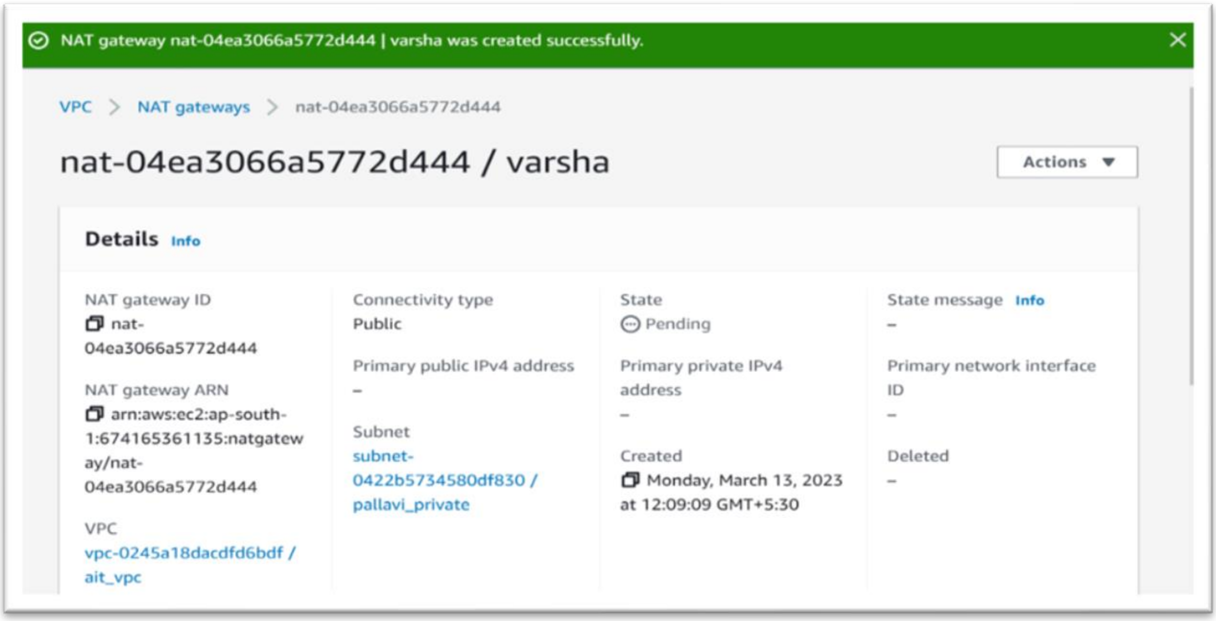
You can add 49 more tags.

Cancel
Create NAT gateway
Feedback
Language
© 2023, Amazon Web Services India Private Limited or i

Snapshot 6.17 Creation of NAT gateway

Snapshot 6.17 shows the Creation of a NAT gateway by allocating an Elastic IP address which is a static IPv4 address designed for dynamic cloud computing. By using an Elastic IP address with a NAT gateway, you can avoid the need to update your DNS settings whenever the public IP address of the NAT gateway changes. This is because Elastic IP addresses can be associated with the NAT gateway and remain constant, even if the underlying infrastructure changes.

6.1.18 Overview of NAT Gateway



NAT gateway nat-04ea3066a5772d444 | varsha was created successfully.

VPC > NAT gateways > nat-04ea3066a5772d444

nat-04ea3066a5772d444 / varsha

Details Info

NAT gateway ID
nat-04ea3066a5772d444

NAT gateway ARN
arn:aws:ec2:ap-south-1:674165361135:natgateway/nat-04ea3066a5772d444

VPC
vpc-0245a18dacdfd6bdf / ait_vpc

Connectivity type
Public

Primary public IPv4 address
–

Subnet
subnet-0422b5734580df830 / pallavi_private

State
Pending

Primary private IPv4 address
–

Created
Monday, March 13, 2023 at 12:09:09 GMT+5:30

State message Info
–

Primary network interface ID
–

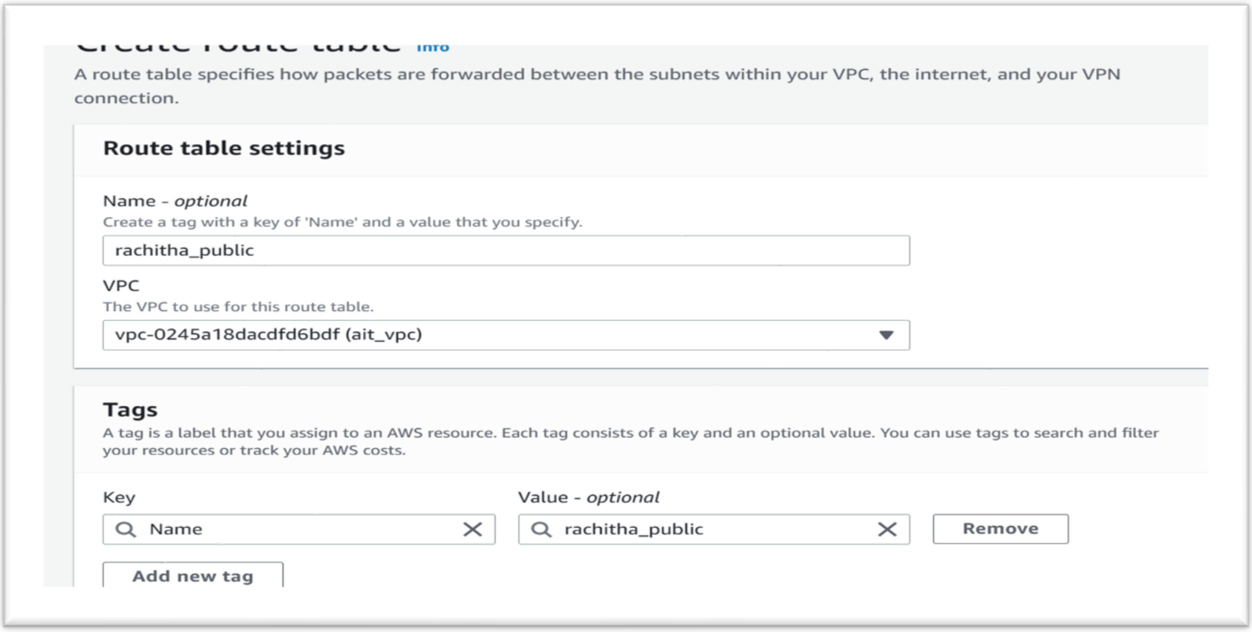
Deleted
–

Snapshot 6.18 Successful creation of NAT Gateway

Snapshot 6.18 shows Creating a NAT gateway that enables instances in a private subnet to connect to the internet or other AWS services. Creation of NAT gateway by allocating Elastic IP address which is a static IPv4 address designed for dynamic cloud computing. NAT gateway that enables instances in a private subnet to connect to the internet or other AWS services. A managed service that provides a highly available, scalable, and secure way to provide outbound internet access to resources in a private subnet.

You can create a NAT gateway in a public subnet and configure the private subnet to route internet-bound traffic through the NAT gateway. By using an Elastic IP address with a NAT gateway, you can avoid the need to update your DNS settings whenever the public IP address of the NAT gateway changes. This is because Elastic IP addresses can be associated with the NAT gateway and remain constant, even if the underlying infrastructure changes.

6.1.19 Creating the Public Routetable



Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*

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

rachitha_public

VPC

The VPC to use for this route table.

vpc-0245a18dacdfd6bdf (ait_vpc)

Tags

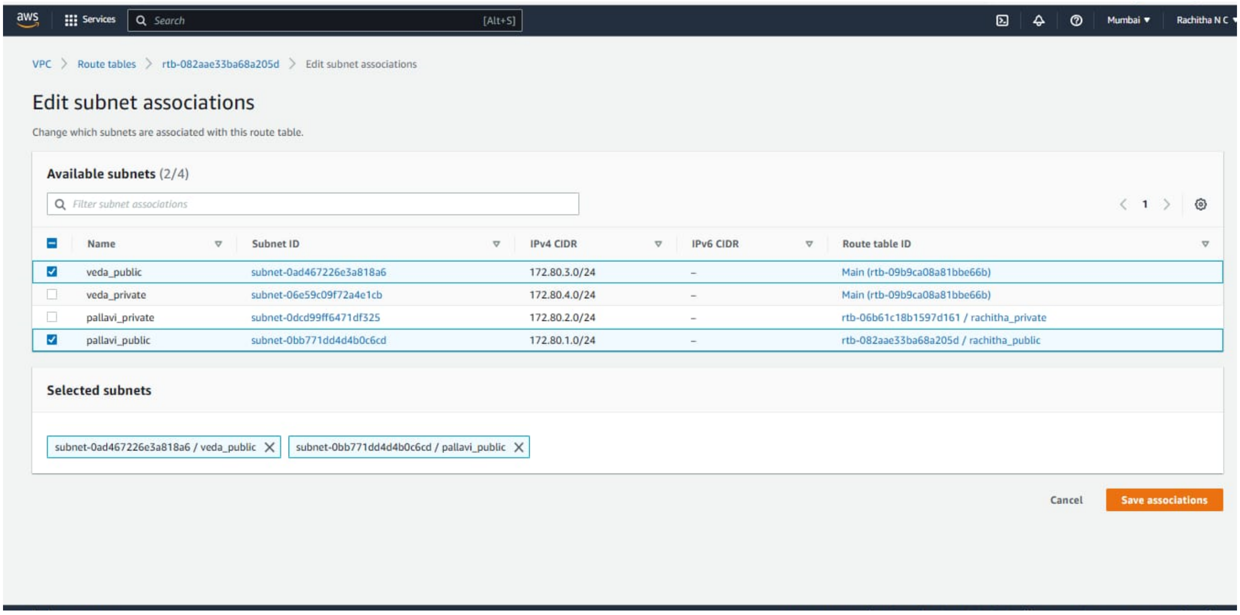
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	rachitha_public

Snapshot 6.19: Creating the public routetable

Snapshot 6.19 shows the Creation of a public route table. In the navigation pane, choose Route Tables. For Name, enter a name for your route table. For VPC, choose your VPC. To add a tag, choose Add new tag and enter the tag key and tag value. Choose to Create route table. In the navigation pane, choose Route tables.

6.1.20 Editing the Subnet Associations



VPC > Route tables > rtb-082aae33ba68a205d > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

Filter subnet associations

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	veda_public	subnet-0ad467226e3a818a6	172.80.3.0/24	–	Main (rtb-09b9ca08a81bbe66b)
□	veda_private	subnet-06e59c09f72a4e1cb	172.80.4.0/24	–	Main (rtb-09b9ca08a81bbe66b)
□	pallavi_private	subnet-0dcd99ff6471df325	172.80.2.0/24	–	rtb-06b61c18b1597d161 / rachitha_private
☑	pallavi_public	subnet-0bb771dd4d4b0c6cd	172.80.1.0/24	–	rtb-082aae33ba68a205d / rachitha_public

Selected subnets

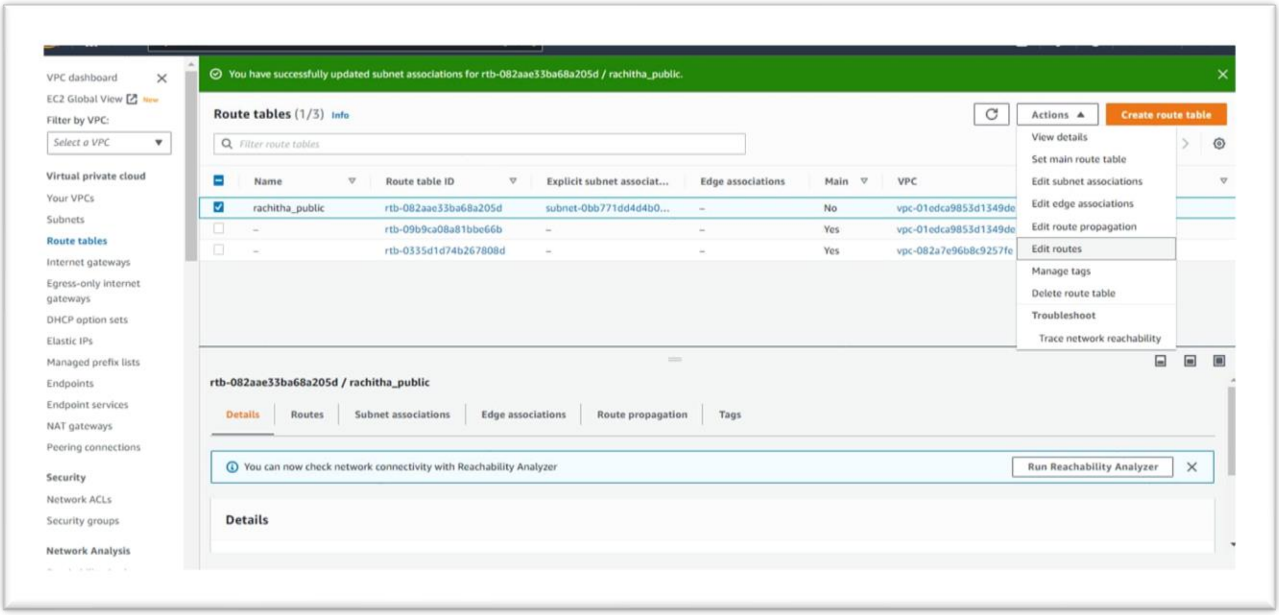
subnet-0ad467226e3a818a6 / veda_public

subnet-0bb771dd4d4b0c6cd / pallavi_public

Snapshot 6.20: Editing the subnet associations

Snapshot 6.20 shows editing the subnet associations. Check the Explicit subnet association column to determine the explicitly associated subnets and the Main column to determine whether this is the main route table. Select the route table and choose the Subnet associations tab.

6.1.21 Editing Routes to Public Routetable



You have successfully updated subnet associations for rtb-082aae33ba68a205d / rachitha_public.

Route tables (1/3)

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
☑	rachitha_public	rtb-082aae33ba68a205d	subnet-0bb771dd4d4b0...	–	No	vpc-01edca9853d1349de
□	–	rtb-09b9ca08a81bbe66b	–	–	Yes	vpc-01edca9853d1349de
□	–	rtb-0335d1d74b267808d	–	–	Yes	vpc-082a7e96b8c9257fe

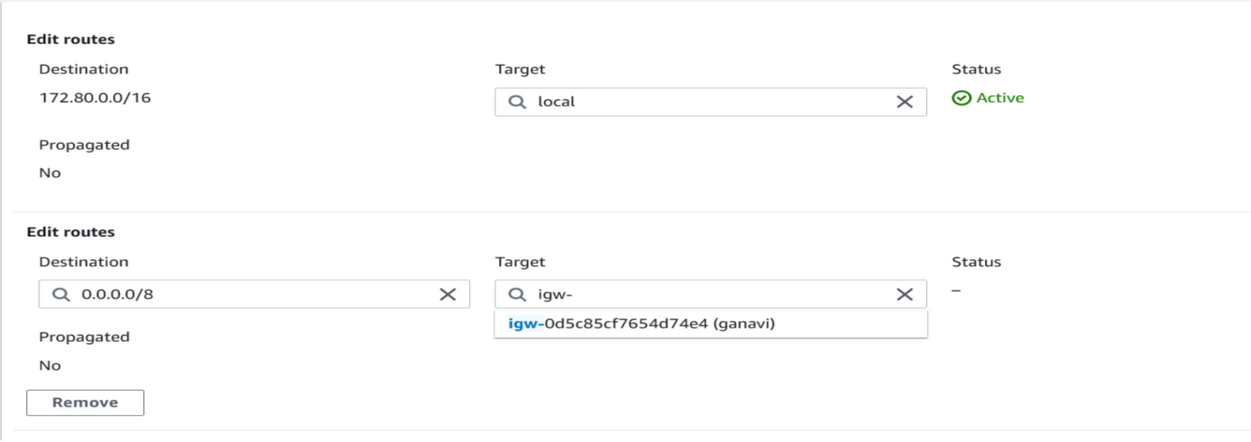
rtb-082aae33ba68a205d / rachitha_public

You can now check network connectivity with Reachability Analyzer

Snapshot 6.21: Editing route to the public routetable

The subnets under Explicit subnet associations are explicitly associated with the route table. The subnets under Subnets without explicit associations belong to the same VPC as the route table, but are not associated with any route table, so they are implicitly associated with the main route table for the VPC.

6.1.22 Connecting the IGW Component



Edit routes

Destination: 172.80.0.0/16
Target: local
Status: Active
Propagated: No

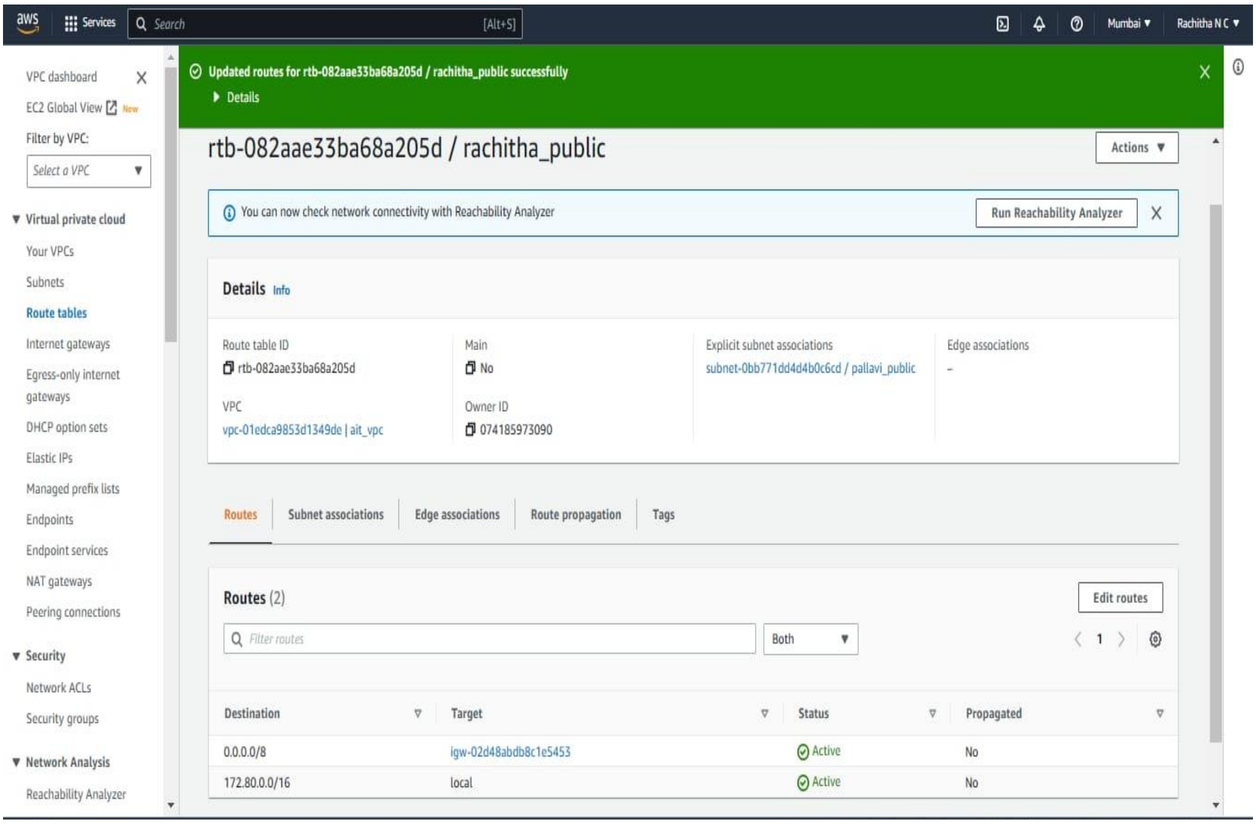
Edit routes

Destination: 0.0.0.0/8
Target: igw-
igw-0d5c85cf7654d74e4 (ganavi)
Status: –
Propagated: No

Snapshot 6.22: Connecting the IGW component

Snapshot 6.22 shows the connecting the igw components. Connecting the selected subnets the public subnet to the IGW gateway for the data transformation. Connecting an IGW to your VPC in AWS provides you with a powerful way to connect your cloud resources to the public internet and other cloud resources. However, it is important to configure your VPC components carefully to ensure that your resources are secure and only accessible to authorized users.

6.1.23 Updated Public Routetable



Updated routes for rtb-082aae33ba68a205d / rachitha_public successfully

rtb-082aae33ba68a205d / rachitha_public

You can now check network connectivity with Reachability Analyzer

Details

Route table ID: rtb-082aae33ba68a205d
Main: No
Explicit subnet associations: subnet-0bb771dd4d4b0c6cd / pallavi_public
Edge associations: –
VPC: vpc-01edca9853d1349de | ait_vpc
Owner ID: 074185973090

Routes (2)

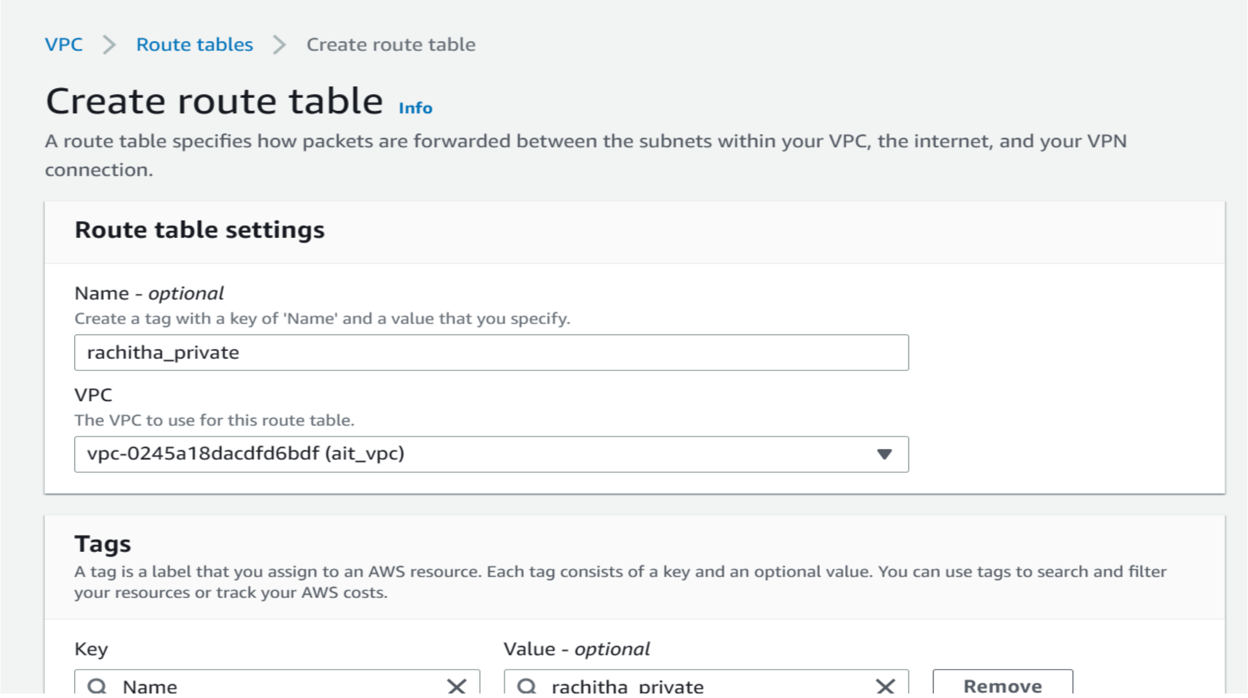
Destination	Target	Status	Propagated
0.0.0.0/8	igw-02d48abdb8c1e5453	Active	No
172.80.0.0/16	local	Active	No

Snapshot 6.23: Updated public routetable

Snapshot 6.23 tells the updating the public route table. The final updation of created routable can be seen as per the above image. Routing table is created, maintained, and updated by a routing protocol running on the router. Intelligent routing protocols are capable of dynamically choosing a different (or better) path when there is a change to the routing infrastructure. When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. The metric value for the path is increased by 1, and the sender is indicated as the next hop.

When you establish a VPC peering connection between two VPCs, you must update the route tables for each VPC to route traffic to the other VPC. This is achieved by adding routes for the remote VPC's CIDR block to the local VPC's route table and vice versa.

6.1.24 Creation of private Routetable



VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
rachitha_private

VPC
The VPC to use for this route table.
vpc-0245a18dacdfd6bdf (ait_vpc)

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	rachitha_private	Remove

Snapshot 6.24: Creation of private routetable

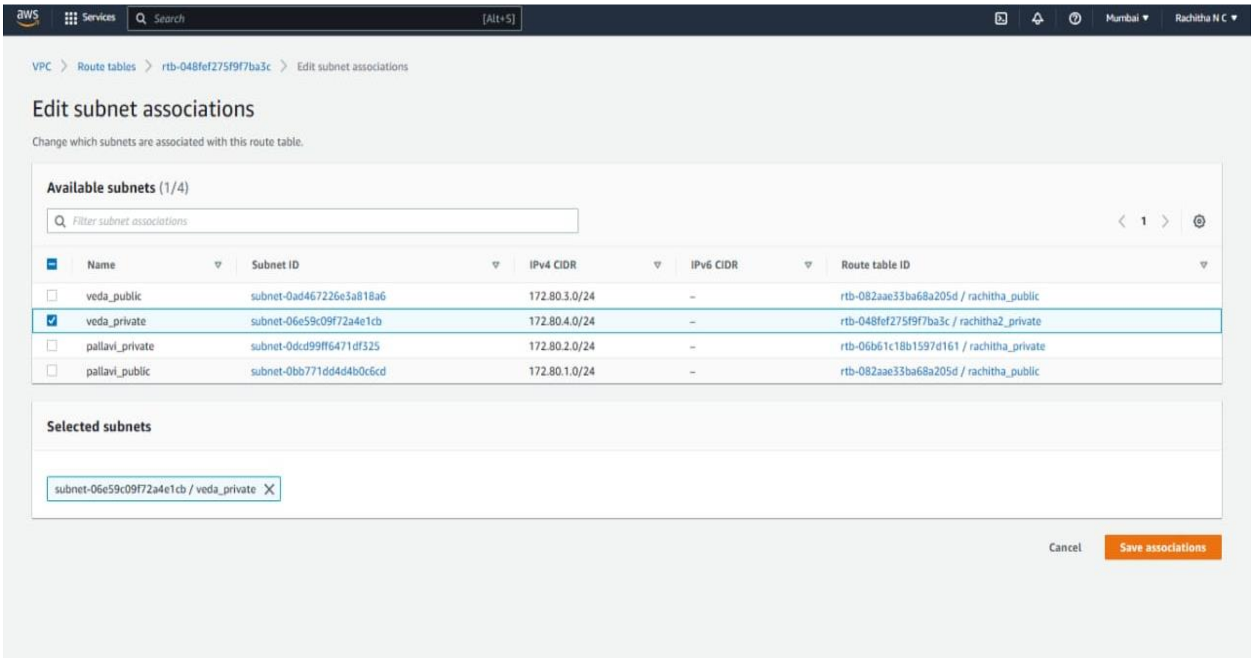
Snapshot 6.24 talks about creating the private routetable. When you create a VPC (Virtual Private Cloud) in AWS (Amazon Web Services), a default route table is automatically created for the VPC. However, you can create additional route tables to customize the routing of traffic within your VPC.

To create a new route table in AWS, you can use the VPC console or the AWS CLI (Command Line Interface). When creating a route table, you must specify the VPC that it will be associated with, as well as any specific routes that you want to add. You can also specify any subnets that you want to associate with the route table.

Once a new route table is created, you can add or remove routes to control how traffic is routed within your VPC. You can also associate subnets with the route table to control the flow of traffic to and from those subnets.

When creating a route table in AWS, it is important to consider the network topology of your VPC and to carefully plan your routing rules to ensure that traffic is flowing correctly. You should also consider security implications when configuring route tables, as improper configuration can leave your VPC open to security threats.

6.1.25 Editing of The Subnet Associations



VPC > Route tables > rtb-048fef275f9f7ba3c > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/4)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
□	veda_public	subnet-0ad467226e3a818a6	172.80.3.0/24	–	rtb-082aae33ba68a205d / rachitha_public
☑	veda_private	subnet-06e59c09f72a4e1cb	172.80.4.0/24	–	rtb-048fef275f9f7ba3c / rachitha2_private
□	pallavi_private	subnet-0dcd99ff6471df325	172.80.2.0/24	–	rtb-06b61c18b1597d161 / rachitha_private
□	pallavi_public	subnet-0bb771dd4d4b0c6cd	172.80.1.0/24	–	rtb-082aae33ba68a205d / rachitha_public

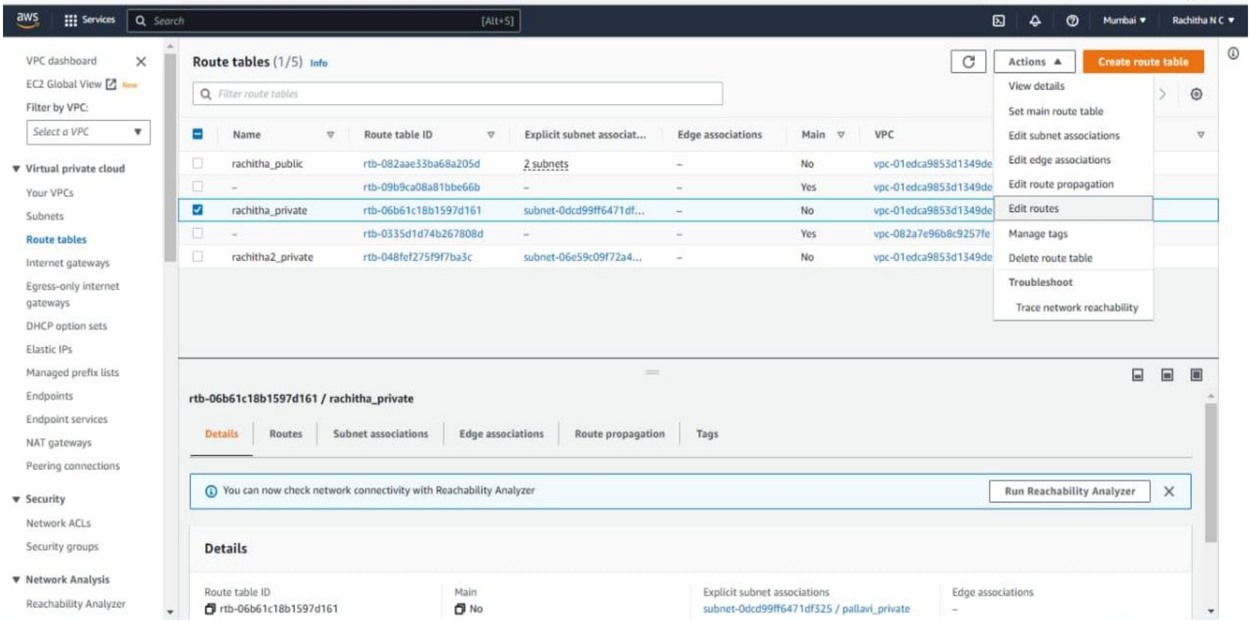
Selected subnets

subnet-06e59c09f72a4e1cb / veda_private

Snapshot 6.25: Editing of the subnet association

Snapshot 6.25 Check the Explicit subnet association column to determine the explicitly associated subnets and the Main column to determine whether this is the main route table. Select the route table and choose the Subnet associations tab.

6.1.26 Edit routes of the Private routetable



Route tables (1/5) Info

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
□	rachitha_public	rtb-082aae33ba68a205d	2 subnets	–	No	vpc-01edca9853d1349de
□	–	rtb-09b9ca08a81bbe66b	–	–	Yes	vpc-01edca9853d1349de
☑	rachitha_private	rtb-06b61c18b1597d161	subnet-0dcd99ff6471df...	–	No	vpc-01edca9853d1349de
□	–	rtb-0335d1d74b267808d	–	–	Yes	vpc-082a7e96b8c9257fe
□	rachitha2_private	rtb-048fef275f9f7ba3c	subnet-06e59c09f72a4...	–	No	vpc-01edca9853d1349de

rtb-06b61c18b1597d161 / rachitha_private

You can now check network connectivity with Reachability Analyzer

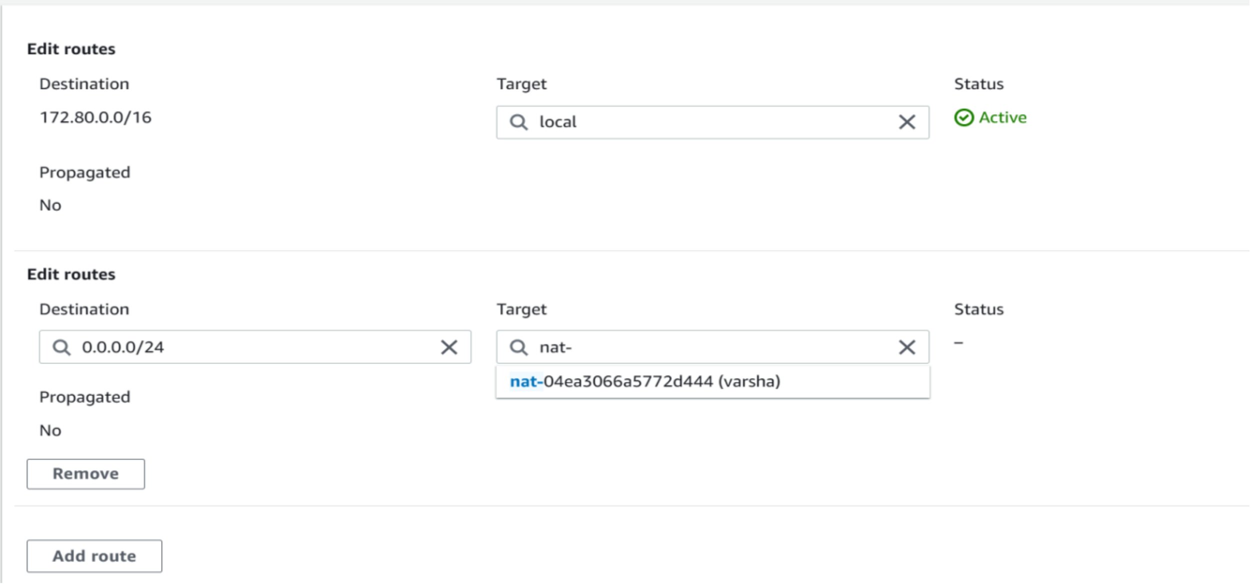
Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-06b61c18b1597d161	No	subnet-0dcd99ff6471df325 / pallavi_private	–

Snapshot 6.26: Editing routes of the private routetables

Snapshot 6.26 talks about the editing routes. Subnets under Explicit subnet associations are explicitly associated with the route table. The subnets under Subnets without explicit associations belong to the same VPC as the route table, but are not associated with any route table, so they are implicitly associated with the main route table for the VPC.

6.1.27 Connecting the NAT Component



Edit routes

Destination	Target	Status
172.80.0.0/16	local	Active

Propagated

No

Edit routes

Destination	Target	Status
0.0.0.0/24	nat-	–

nat-04ea3066a5772d444 (varsha)

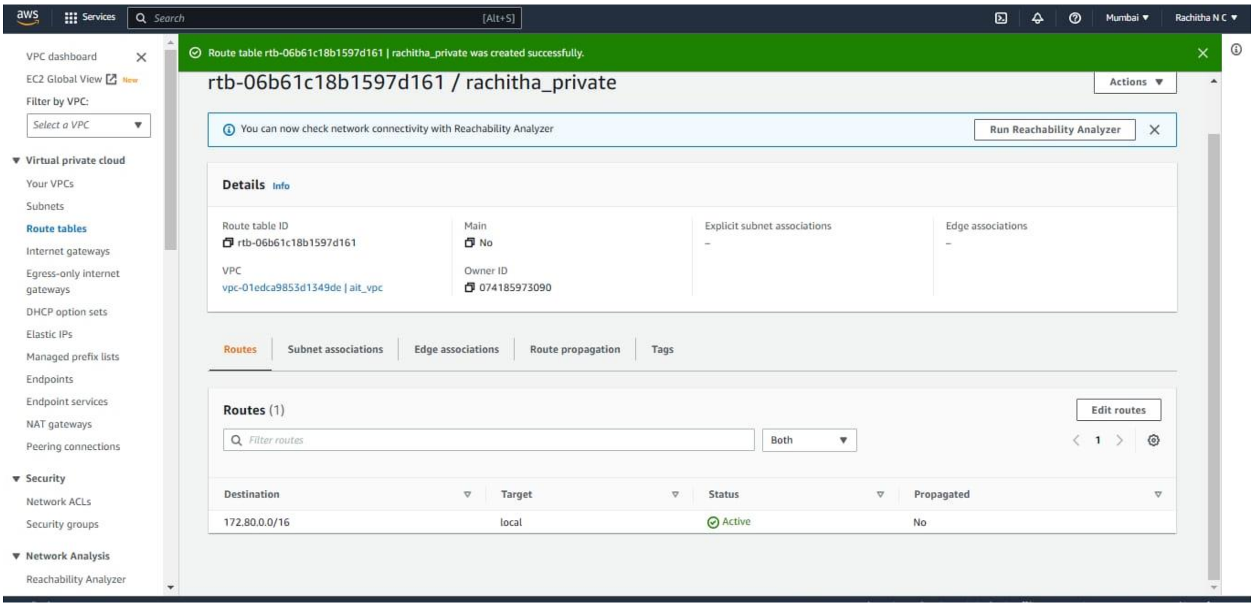
Propagated

No

Snapshot 6.27: Connecting the NAT components

Snapshot 6.27 shows connecting the NAT components. Connecting the selected subnet i.e., the private subnet to the NAT gateway for the data transformation.

6.1.28 Updated Private Routetable



Route table rtb-06b61c18b1597d161 | rachitha_private was created successfully.

rtb-06b61c18b1597d161 / rachitha_private

You can now check network connectivity with Reachability Analyzer

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-06b61c18b1597d161	No	–	–
VPC	Owner ID		
vpc-01edca9853d1349de | ait_vpc	074185973090		

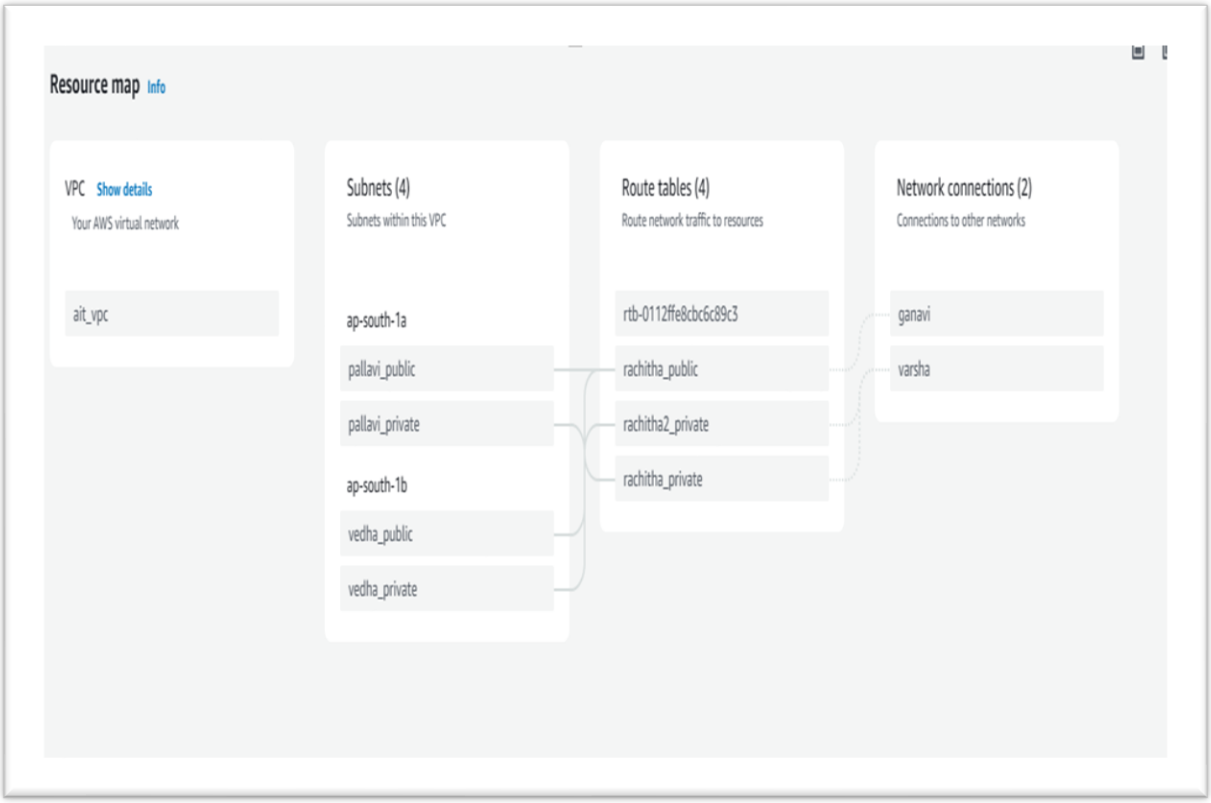
Routes (1)

Destination	Target	Status	Propagated
172.80.0.0/16	local	Active	No

Snapshot 6.28: Updated private routetable

Snapshot 6.28 The final updation of created routable can be seen as per the above image. When you establish a VPC peering connection between two VPCs, you must update the route tables for each VPC to route traffic to the other VPC. This is achieved by adding routes for the remote VPC's CIDR block to the local VPC's route table and vice versa.

6.1.29 Final Result of VPC:



Snapshot 6.29: Final result of the VPC

Snapshot 6.29 shows the result of the VPC connection in the AWS. Brief structure of VPC connection makes user possible to understand the connection and any disturbance if occurred while making the connections in the VPC.

To successfully create a VPC in AWS, you must first choose the appropriate IP address range for your VPC. This IP address range will be used to define the IP addresses that can be assigned to the resources within your VPC. You should also consider the number of subnets that you will need within your VPC, and the number of resources that will be deployed within each subnet.

Once you have defined your IP address range and subnet structure, you can create your VPC using the VPC console or the AWS CLI (Command Line Interface). When creating your VPC, you will need to specify the IP address range and the number of subnets that you want to create. After your VPC is created, you can begin deploying resources within your subnets. These resources can include EC2 instances, RDS databases, and other cloud resources that are deployed within your VPC.

To ensure the success of your VPC deployment, it is important to carefully plan and configure your VPC components. This includes configuring your routing tables, security groups, and network access control lists (NACLs) to control access to your resources and limit exposure to security threats. Overall, a successfully created VPC in AWS provides you with a powerful way to deploy cloud resources in a virtual network environment. By carefully planning and configuring your VPC, you can ensure that your resources are secure and operating at optimal performance.

6.2 Summary

This chapter 6 gives the overall view of the Results and Discussions. Section 6.1 shows VPC connections offer several benefits, including improved security, increased flexibility, and cost-effectiveness. By connecting to on-premises data centers or other VPCs, users can create hybrid cloud environments that enable them to leverage the benefits of both cloud and on-premises infrastructure.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

Virtual Private Cloud (VPC) is a powerful networking service provided by Amazon Web Services (AWS) that allows users to create a logically isolated virtual network within the AWS cloud infrastructure.VPC allows users to create and manage virtual networks, subnets, and route tables, as well as control inbound and outbound traffic using security groups and network access control lists (ACLs). It also provides the ability to connect VPCs to on-premises data centers or other VPCs using various networking options such as AWS Direct Connect, VPN, and VPC Peering.

VPC (Virtual Private Cloud) in AWS (Amazon Web Services) is a powerful way to deploy and manage cloud resources in a virtual network environment. A VPC provides you with complete control over your cloud infrastructure, allowing you to define your own network topology, IP address range, and subnet structure. By creating a VPC in AWS, you can deploy cloud resources in a secure and isolated environment that can be customized to meet your specific needs. You can also connect your VPC to the internet using an Internet Gateway (IGW), allowing resources within your VPC to communicate with resources on the public internet.

When deploying resources within your VPC, it is important to carefully plan and configure your VPC components, including your routing tables, security groups, and network access control lists (NACLs). By doing so, you can ensure that your resources are secure and only accessible to authorized users. Overall, a VPC in AWS provides you with a flexible and scalable way to deploy and manage cloud resources. By taking advantage of the powerful features and tools provided by AWS, you can create a secure and reliable cloud infrastructure that meets your specific needs and requirements.

7.2 Future Enhancement

VPCs are already quite secure, but AWS could potentially enhance their security features even further. For example, the company could add more granular controls for network traffic, or introduce new features for detecting and responding to potential security threats. Greater automation and ease of use AWS could work to make it easier for customers to create and manage

VPC connections. This could involve introducing new automation tools, improving the user interface, or offering more detailed documentation and support resources.

More flexible connectivity options: Finally, AWS could consider offering more flexible connectivity options for VPCs. For example, the company could introduce new options for connecting VPCs across different regions or availability zones, or offer more granular control over how traffic is routed between different VPCs"