

BUILDING A CONNECTED CLOUD NETWORK IN AWS

A report submitted in partial fulfilment of the requirements for the Award of a Degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

By

PENUGONDA SRI RAM KARTHIK

21B91A54E4

Under Supervision of Mr. GURU SANTHOSH (Trainer name)

Blackbuck Engineers Pvt. Ltd, Road No 36, Jubilee Hills, Hyderabad

(Duration: 5th July 2023 to 5th September 2023)



DEPARTMENT OF INFORMATION TECHNOLOGY

S.R.K.R. ENGINEERING COLLEGE

(Autonomous)

SRKR MARG, CHINNA AMIRAM, BHIMAVARAM-534204, A.P

(Recognized by A.I.C.T.E New Delhi) (Accredited by NBA & NAAC)

(Affiliated to JNTU, KAKINADA)

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE
(Autonomous)

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the Summer Internship Report titled “BUILDING A CONNECTED CLOUD NETWORK IN AWS” is the bonafide work done by Mr. Penugonda Sri Ram Karthik bearing Register Number 21B91A54E4 at the end of second year second semester at Blackbuck Engineers Pvt. Ltd, Road No 36, Jubilee Hills, Hyderabad

from 5th July 2023 to 5th September 2023 in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology.

Department Internship Coordinator

Dean -T & P Cell

Head of the Department



**BLACKBUCK
ENGINEERS**



Internship Certificate

Certificate ID: BB23INT00659 Issued Date: 9th September 2023

To Whom It May Concern

This is to certify that PENUGONDA SRI RAM KARTHIK bearing Reg No: 21B91A54E4 has successfully completed an internship at Blackbuck Engineers Pvt Ltd Hyderabad from 5th July 2023 to 5th September 2023.

He She has worked on a project titled **BUILDING A CONNECTED CLOUD NETWORK IN AWS** by learning and incorporating Amazon Web Services concepts under the supervision of our project mentor.

We found that he/she is sincere, hardworking, technically sound and result oriented. He/She worked well as part of a team during his tenure.

We wish him/her all the best for his/her future endeavors.

Best regards,

Kathyayani. R
Project Head

Mounika Bezwada
HR Manager



www.theblackbucks.com



Jubilee Hills, Hyderabad



+91 9392900172



contact@blackbucks.me



Verify Certificate at
verify.blackbucks.me

BLACKBUCK INTERNSHIP WORK

Title:

Building a connected cloud network in AWS.

Abstract:

The "Connected Cloud Network" project aims to create a robust and scalable cloud network infrastructure within the Amazon Web Services (AWS) ecosystem. This network will facilitate seamless communication and data exchange between various resources and services hosted on AWS, providing a secure and efficient environment for applications and systems to interact with each other. The successful completion of this project will result in a fully operational and secure connected cloud network, empowering organizations to leverage AWS's capabilities effectively for their applications and services.

TABLE OF CONTENTS

<u>Services used</u>	3
<u>Rough architecture</u>	3
<u>Final architecture</u>	4
<u>Cloud computing</u>	4
<u>Cloud computing services</u>	5
IaaS	5
PaaS	6
SaaS	6
<u>Cloud service providers</u>	6
<u>Amazon web services</u>	7
<u>Why AWS?</u>	8
<u>List of AWS Services</u>	9
<u>Amazon EC2</u>	10
<u>Amazon RDS</u>	11
Multiple AZ Deployment...	11
<u>Read replicas</u>	11
<u>Performance metrics and monitoring</u>	12
<u>Amazon VPC</u>	12
<u>Amazon S3</u>	13
<u>Amazon IAM</u>	14
<u>AWS Lambda</u>	16
<u>AWS Cloud9</u>	17
Environment and computing resources.....	18
<u>AWS Elastic BeanStalk</u>	18
<u>AWS CodeCommit</u>	19
<u>Amazon CloudWatch</u>	21
Amazon EBS	22
<u>Features of Amazon EBS</u>	22
<u>Amazon Aurora</u>	24
<u>AWS Autoscaling</u>	25
<u>Autoscaling benefits</u>	26
<u>Implementation</u>	27
Screenshots.....	27 – 58

Services used:

- EC2 (Elastic Compute Cloud)
- VPC (Virtual Private Cloud)
- IAM (Identity and Access Management)

Description:

The "Connected Cloud Network" project aims to create a robust and scalable cloud network infrastructure within the Amazon Web Services (AWS) ecosystem. This network will facilitate seamless communication and data exchange between various resources and services hosted on AWS, providing a secure and efficient environment for applications and systems to interact with each other. The successful completion of this project will result in a fully operational and secure connected cloud network, empowering organizations to leverage AWS's capabilities effectively for their applications and service.

Cloud computing

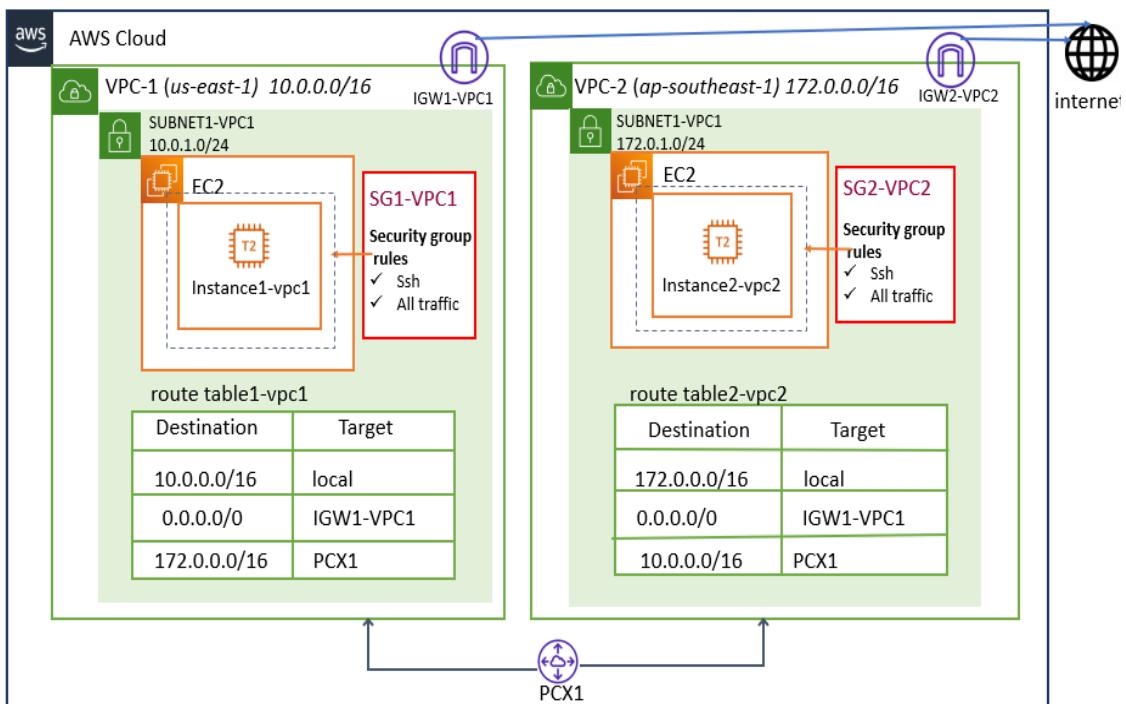
Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage.

Cloud Computing Services:

- IaaS (Infrastructure-as-a-Service)
- PaaS (Platform-as-a-Service)
- SaaS (Software-as-a-service)

are the three most common models of cloud services, and it's not uncommon for an organization to use all three.

ARCHITECTURE:



IaaS (Infrastructure-as-a-Service)

IaaS provides on-demand access to fundamental computing resources—physical and virtual servers, networking, and storage—over the internet on a pay-as-you-go basis. IaaS enables end users to scale and shrink resources on an as-needed basis, reducing the need for high, up-front capital expenditures or unnecessary on-premises or ‘owned’ infrastructure and for overbuying resources to accommodate periodic spikes in usage.

In contrast to SaaS and PaaS (and even newer PaaS computing models such as containers and serverless), IaaS provides the users with the lowest-level control of computing resources in the cloud.

IaaS was the most popular cloud computing model when it emerged in the early 2010s. While it remains the cloud model for many types of workloads, use of SaaS and PaaS is growing at a much faster rate.

PaaS (Platform-as-a-service)

PaaS provides software developers with on-demand platform—hardware, complete software stack, infrastructure, and even development tools—for running, developing, and managing applications without the cost, complexity, and inflexibility of maintaining that platform on-premises.

With PaaS, the cloud provider hosts everything—servers, networks, storage, operating system software, middleware, databases—at their data center. Developers simply pick from a menu to ‘spin up’ servers and environments they need to run, build, test, deploy, maintain, update, and scale applications.

Today, PaaS is often built around containers, a virtualized compute model one step removed from virtual servers. Containers virtualize the operating system, enabling developers to package the application with only the operating system services it needs to run on any platform, without modification and without need for middleware.

SaaS (Software-as-a-Service)

SaaS—also known as cloud-based software or cloud applications—is application software that’s hosted in the cloud, and that user’s access via a web browser, a dedicated desktop client, or an API that integrates with a desktop or mobile operating system. In most cases, SaaS users pay a monthly or annual subscription fee; some may offer ‘pay-as-you-go’ pricing based on your actual usage.

In addition to the cost savings, time-to-value, and scalability benefits of cloud, SaaS offers the following:

- **Automatic upgrades:** With SaaS, users take advantage of new features as soon as the provider adds them, without having to orchestrate an on-premises upgrade.
- **Protection from data loss:** Because SaaS stores application data in the cloud with the application, users don’t lose data if their device crashes or breaks.

SaaS is the primary delivery model for most commercial software today—there are hundreds of thousands of SaaS solutions available, from the most focused industry and departmental applications to powerful enterprise software database and AI (artificial intelligence) software.

Cloud Service Providers:

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Oracle
- IBM cloud
- Salesforce

Amazon Web Services:

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered, pay-as-you-go basis. Oftentimes, clients will use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, computing, storage, middleware, IoT and other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware, and operating systems.

One of the foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard disk /SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).

AWS services are delivered to customers via a network of AWS server farms located throughout the world. Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either.

Amazon provides select portions of security for subscribers (e.g., physical security of the data centers) while other aspects of security are the responsibility of the subscriber (e.g., account management, vulnerability scanning, patching). AWS operates for many global geographical regions including seven in North America.

Amazon markets AWS to subscribers as a way of obtaining large-scale computing capacity more quickly and cheaply than building an actual physical server farm. All services are billed based on usage, but each service measures usage in varying ways. As of 2021 Q4, AWS has 33% market share for cloud infrastructure while the next two competitors Microsoft Azure and Google Cloud have 21%, and 10% respectively, according to Synergy Grou

List of AWS Services:

Amazon, the preeminent cloud vendor, broke new ground by establishing the first cloud computing service, Amazon EC2, in 2008. AWS offers more solutions and features than any other provider and has free tiers with access to the AWS Console, where users can centrally control their ministrations.

Designed around ease-of-use for various skill sets, AWS is tailored for those unaccustomed to software development utilities. Web applications can be deployed in minutes with AWS facilities, without provisioning servers or writing additional code.

- Amazon EC2 (Elastic Compute Cloud)
- Amazon RDS (Relational Database Services)
- Amazon S3 (Simple Storage Service)
- Amazon Lambda
- Amazon Cognito
- Amazon Glacier
- Amazon SNS (Simple Notification Service)
- Amazon VPC (Virtual Private Cloud)
- Amazon Lightsail
- Amazon CloudWatch
- Amazon Cloud9
- Amazon Elastic Beanstalk
- Amazon CodeCommit
- Amazon IAM (Identity and Access Management)
- Amazon Inspector
- Amazon Kinesis
- Amazon Dynamo DB
- Amazon Codecatalyst
- Amazon Kinesis
- AWS Athena
- AWS Amplify
- AWS Quicksight
- AWS Cloudformation

Amazon EC2:

Amazon Elastic Compute Cloud (EC2) is a part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy. In November 2010, Amazon switched its own retail website platform to EC2 and AWS.

Amazon RDS:

Amazon Relational Database Service (or Amazon RDS) is a distributed relational database service by Amazon Web Services (AWS). It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications. Administration processes like patching the database software, backing up databases and enabling point-in-time recovery are managed automatically. Scaling storage and compute resources can be performed by a single API call to the AWS control plane on-demand. AWS does not offer an SSH connection to the underlying virtual machine as part of the managed service.

Amazon VPC:

Amazon Virtual Private Cloud (VPC) is a commercial cloud computing service that provides a virtual private cloud, by provisioning a logically isolated section of Amazon Web Services (AWS) Cloud. Enterprise customers are able to access the Amazon Elastic Compute Cloud (EC2) over an IPsec based virtual private network. Unlike traditional EC2 instances which are allocated internal and external IP numbers by Amazon, the customer can assign IP numbers of their choosing from one or more subnets.

Amazon S3:

Amazon S3 manages data with an object storage architecture which aims to provide scalability, high availability, and low latency with high durability. The basic storage units of Amazon S3 are objects which are organized into buckets. Each object is identified by a unique, user-assigned key. Buckets can be managed using the console provided by Amazon S3, programmatically with the AWS SDK, or the REST application programming interface.

Amazon IAM:

IAM provides the infrastructure necessary to control authentication and authorization for your AWS account. The IAM infrastructure is illustrated by the following diagram.

IMPLEMENTATION

Creating an IAM user with EC2 full access

Step-1 : Select **users** section in IAM management console and then select **add users**.

User name	Groups	Last activity	MFA	Password a...
iam-userx	None	9 days ago	Virtual	9 days ago
iam-usery	srkr	9 days ago	None	9 days ago
iamkarthik	None	1 minute ago	None	13 days ago
karthik-penugonda	None	10 days ago	None	13 days ago

Step-2: Give the username, select the checkbox and select *I want to create an IAM user.*

Specify user details

User details

User name
assignment

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#).

Are you providing console access to a person?

User type

Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Step-3: Give a password to your IAM account and select **Next**.

Console password

Autogenerated password
You can view the password after you create the user.

Custom password
Enter a custom password for the user.

• Must be at least 8 characters long
• Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | '

Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Step-4: Select ***Attach policies directly*** in Permissions options.

The screenshot shows the 'Set permissions' step of the IAM user creation wizard. The 'Permissions options' section contains three radio button choices:

- Add user to group: Adds user to an existing group or creates a new group. Recommended for managing user permissions by job function.
- Copy permissions: Copies all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly: Attaches a managed policy directly to a user. Recommended for attaching policies to a group instead, then adding the user to the group.

Below the options is a 'Permissions policies' list with a search bar and a 'Create policy' button. The list shows 1106 policies. A filter bar at the bottom allows filtering by 'Policy name' (set to 'ec2fullaccess'), 'Type' (set to 'AWS managed'), and 'Attached entities' (set to 0).

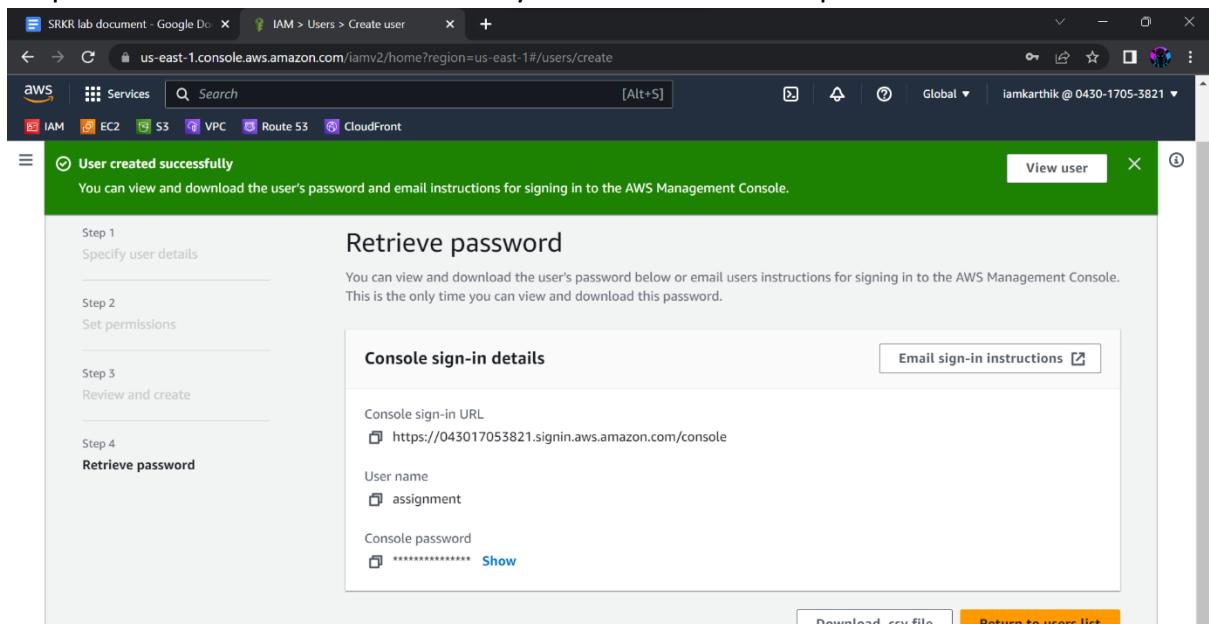
In Permission policies, select ***AmazonEC2FullAccess*** and select ***Next***.

The screenshot shows the 'Permissions policies' list after filtering by 'ec2fullaccess'. The 'AmazonEC2FullAccess' policy is selected and highlighted with a blue border. The 'Next' button is visible at the bottom right of the screen.

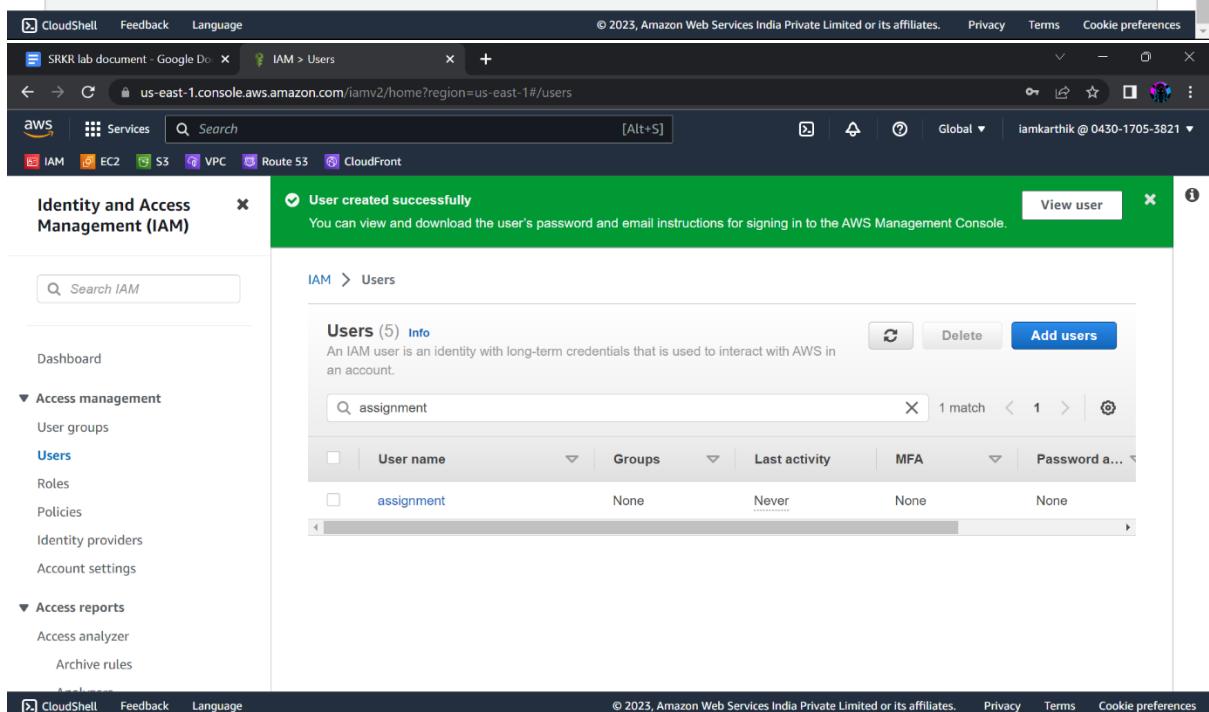
Step-5: In the next step Review and Create, click on **Create user**.

The screenshot shows the 'Review and create' step of the AWS IAM 'Create user' wizard. The user name is set to 'assignment'. The console password type is 'Custom password'. The 'Require password reset' option is set to 'No'. The 'Permissions summary' section shows a single policy named 'AmazonEC2FullAccess' which is an 'AWS managed' policy used as a 'Permissions policy'. The 'Tags - optional' section indicates no tags are associated with the resource. At the bottom, there are 'Cancel', 'Previous', and 'Create user' buttons, with 'Create user' being highlighted.

Step-6: IAM user created successfully with EC2 full access permissions.



The screenshot shows the 'Retrieve password' step of the IAM user creation process. It displays the 'Console sign-in details' section, which includes the sign-in URL (<https://043017053821.signin.aws.amazon.com/console>), user name ('assignment'), and console password ('*****'). Buttons for 'Download .csv file' and 'Return to users list' are at the bottom.



The screenshot shows the 'Users' list page in the IAM service. It lists five users, including the newly created one named 'assignment'. The user details table includes columns for User name, Groups, Last activity, MFA, and Password last used. The user 'assignment' is listed with 'None' in all these categories.

Creating a VPC with VPC only wizard, subnet, internet gateway and route table

Step-1: Select **Create VPC** in VPC Management Console.

The screenshot shows the AWS VPC Management Console interface. On the left, there's a sidebar with options like 'Virtual private cloud', 'Your VPCs', 'Subnets', 'Route tables', and various gateway types. The main area displays 'Resources by Region' with counts for VPCs, Subnets, Route Tables, Internet Gateways, Egress-only Internet Gateways, NAT Gateways, VPC Peering Connections, Network ACLs, Security Groups, and Customer Gateways. On the right, there are sections for 'Service Health', 'Settings', 'Additional Information', and 'AWS Network Manager'.

Step-2: Select **VPC only wizard** and give IPV4 CIDR then select **Create VPC**.

The screenshot shows the 'Create VPC' wizard. The first step, 'Create VPC Info', is completed. The second step, 'VPC settings', is shown. Under 'Resources to create', the 'VPC only' option is selected. There's a field for 'Name tag - optional' containing 'assignment-vpc'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the value '10.199.2.0/24' is entered. The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, Language, and cookie preferences.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)
 Default

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

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="assignment-vpc"/>

Add tag
 You can add 49 more tags

Cancel Create VPC

You successfully created **vpc-04e0f94301abcaa77 / assignment-vpc**

VPC > Your VPCs > vpc-04e0f94301abcaa77

vpc-04e0f94301abcaa77 / assignment-vpc

Details [Info](#)

VPC ID <input type="text" value="vpc-04e0f94301abcaa77"/>	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set <input type="text" value="dopt-0b8365933d7172353"/>	Main route table <input type="text" value="rtb-0050fa3acc34fb9f6"/>	Main network ACL <input type="text" value="acl-042be2bbef734011f"/>
Default VPC No	IPv4 CIDR 10.199.2.0/24	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID <input type="text" value="043017053821"/>	-

[Resource map](#) [New](#) [CIDRs](#) [Flow logs](#) [Tags](#)

Step-3: Select your VPC in ***Filter by VPC.***

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
assignment-vpc	vpc-04e0f94301abcaa77	Available	10.199.2.0/24	-
-	vpc-0046264f562b278ce	Available	172.31.0.0/16	-

Step-4: Select Subnets and then click on ***Create Subnet.***

Name	Subnet ID	State	VPC
No matching resource found			

Step-5: Select your VPC in VPC ID and select *Create subnet*.

VPC ID: vpc-04e0f94301abcaa77 (assignment-vpc)

Associated VPC CIDRs: 10.199.2.0/24

Name	Subnet ID	State	VPC
my-subnet	subnet-0b3b4281829f9d5b0	Available	vpc-04e0f94301abcaa77 assig...

Step-6: In internet gateways, select **Create internet gateway**.

VPC dashboard X

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
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services

Internet gateways Info

Actions Create internet gateway

Filter internet gateways

search: vpc-04e0f94301abcaa77 Clear filters

Name	Internet gateway ID	State	VPC ID
No matching resource found			

Select an internet gateway above

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Give a name to your internet gateway and click **Create internet gateway**.

SRKR lab document - Google Doc + Create internet gateway | VPC M ...

Services Search [Alt+S] N. Virginia iamkarthik @ 0430-1705-3821

for the gateway below.

Internet gateway settings

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

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
Q Name	Q my-ig Remove

Add new tag
You can add 49 more tags.

Cancel Create internet gateway

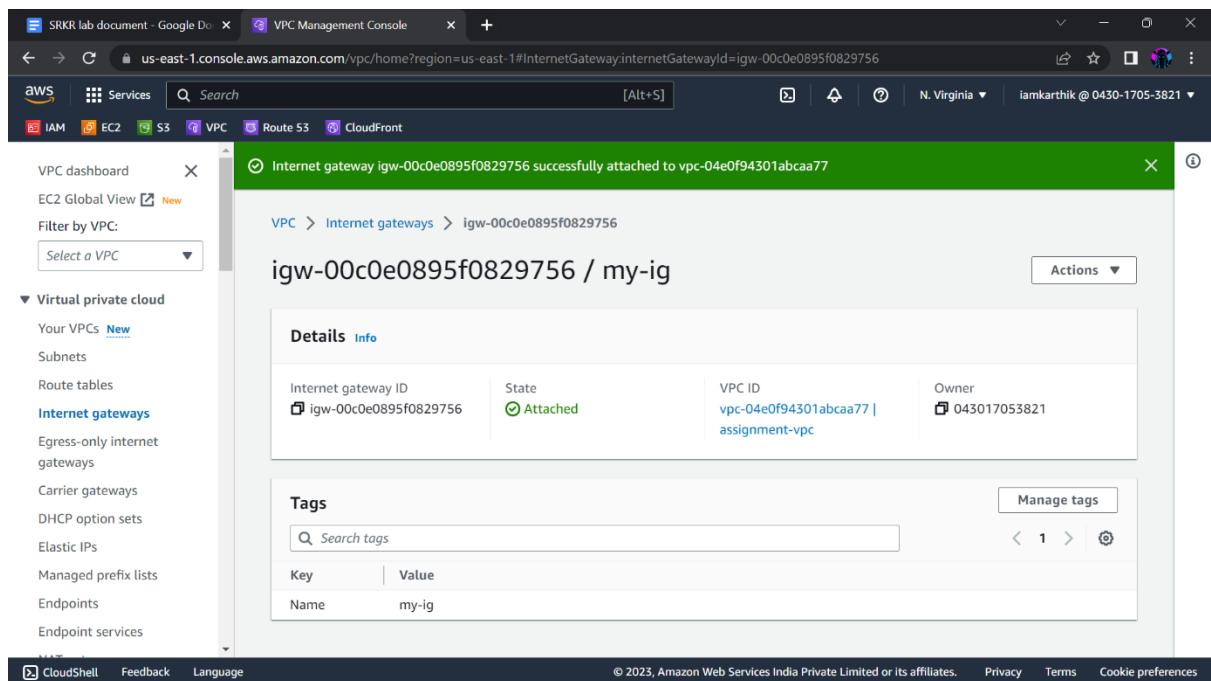
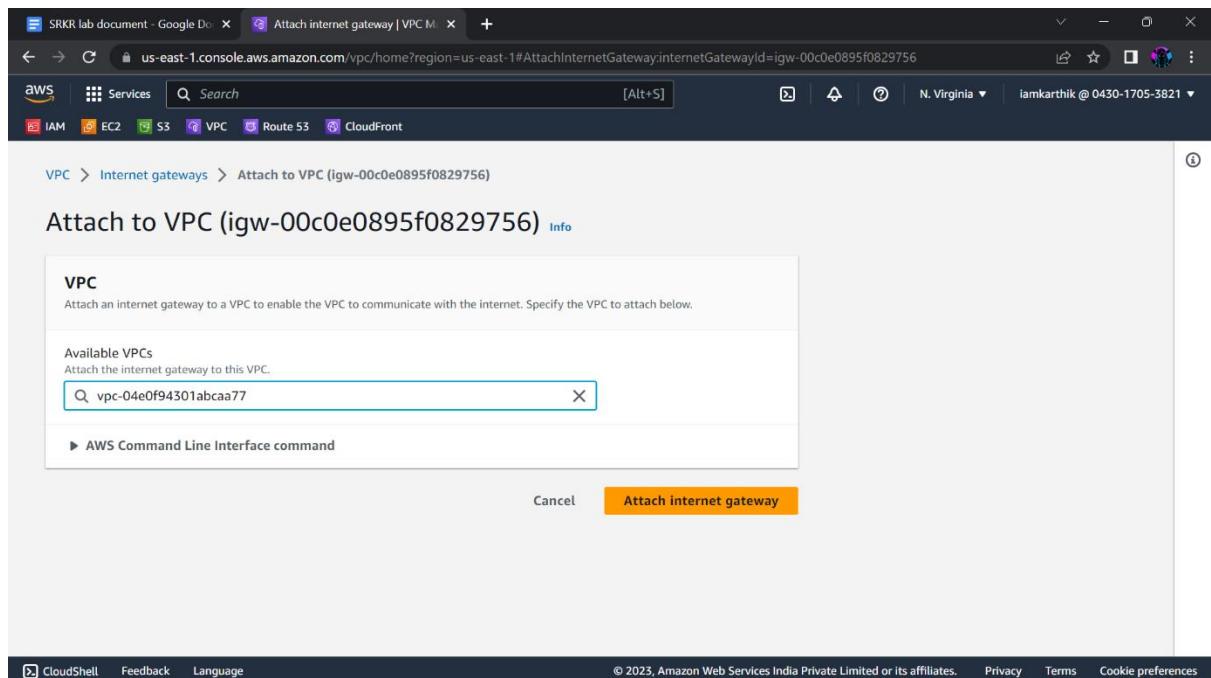
CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS VPC Management Console. A green banner at the top indicates that an internet gateway has been created and can now be attached to a VPC. The main page displays the details of the new internet gateway, including its ID (igw-00c0e0895f0829756), state (Detached), and owner (iamkarthik @ 0430-1705-3821). The 'Actions' dropdown menu is open, showing options like 'Attach to a VPC'.

Step-7: Select your internet gateway and in Actions select *Attach to VPC*.

The screenshot shows the AWS VPC Management Console displaying a list of internet gateways. One gateway, 'my-ig' (ID: igw-00c0e0895f0829756), is listed with a green checkmark indicating it is attached to a VPC. The 'Actions' dropdown menu for this gateway includes the option 'Attach to VPC'. Below the list, a detailed view of the selected gateway shows its configuration.

Select your VPC and click on **Attach internet gateway**.



Step-8: In Route tables section, select **Create route table**.

The screenshot shows the AWS VPC Management Console with the URL us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#RouteTables:vpcId=vpc-04e0f94301abcaa77. The left sidebar is expanded to show the 'Route tables' section under 'Virtual private cloud'. At the top right, there is a prominent orange 'Create route table' button.

Give a name to your route table and select your VPC then click on **Create route table**.

The screenshot shows the 'Create route table' configuration page. It includes fields for 'Name - optional' (set to 'my-routetable-01'), 'VPC' (set to 'assignment-vpc'), and 'Tags' (one tag named 'Name' with value 'my-routetable-01'). At the bottom right is a large orange 'Create route table' button.

The screenshot shows the AWS VPC Management Console with a success message: "Route table rtb-063dde2a421c737ae | my-routetable-01 was created successfully." Below this, the "Details" tab is selected, showing the route table ID (rtb-063dde2a421c737ae), which is Main and has no explicit subnet associations or edge associations. It also lists the VPC (vpc-04e0f94301abcaa77) and its owner ID (043017053821). At the bottom, there are tabs for "Routes", "Subnet associations", "Edge associations", "Route propagation", and "Tags".

Step-9: In the routes, select *Edit Routes*.

This screenshot is identical to the one above, showing the successful creation of the route table. However, the "Routes" tab is now selected, revealing a table titled "Routes (1)". The table includes columns for Destination (10.199.2.0/24), Target (local), Status (Active), and Propagated (No). A prominent "Edit routes" button is located at the top right of the routes table.

Add internet **0.0.0.0/0** in destination and your **internet gateway** in target then click on **Save changes**.

The screenshot shows the AWS VPC Management Console with the URL <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#EditRoutes:RouteTableId=rtb-063dde2a421c737ae>. The page is titled "Edit routes". A table lists routes with columns: Destination, Target, Status, and Propagated. One route is listed with a destination of 10.199.2.0/24 and a target of "local". Another route is being added with a destination of 0.0.0.0/0 and a target of "igw-00c0e0895f0829756". The status is "Active" and propagated is "No". Buttons at the bottom include "Cancel", "Preview", and a highlighted "Save changes".

The screenshot shows the AWS VPC Management Console with the URL <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-063dde2a421c737ae>. The left sidebar shows "Virtual private cloud" with "Route tables" selected. The main area shows a success message: "Updated routes for rtb-063dde2a421c737ae / my-routetable-01 successfully". The "Subnet associations" tab is active. It shows a table with one entry: "VPC" (vpc-04e0f94301abcaa77) and "Owner ID" (043017053821). Below this, the "Explicit subnet associations (0)" section shows a table with a single row for "Find subnet association". The "Subnets without explicit associations (1)" section shows one subnet: "subnet-00000000000000000000". The footer includes "CloudShell", "Feedback", "Language", "© 2023, Amazon Web Services India Private Limited or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

Step-10: In Subnet Associations, select **Edit subnet associations**.

VPC > Route tables > rtb-063dde2a421c737ae > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)					
<input type="text"/> Filter subnet associations					
<input checked="" type="checkbox"/> Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
<input checked="" type="checkbox"/> my-subnet	subnet-0b3b4281829f9d5b0	10.199.2.0/24	-	Main (rtb-0050fa3acc34fb9f6)	

Selected subnets	
subnet-0b3b4281829f9d5b0 / my-subnet <input type="button" value="X"/>	

Cancel

Select your subnet in available subnets and click on **Save changes**.

You have successfully updated subnet associations for rtb-063dde2a421c737ae / my-routetable-01.

rtb-063dde2a421c737ae / my-routetable-01

You can now check network connectivity with Reachability Analyzer

Details		Info
Route table ID	Main	Explicit subnet associations
<input type="checkbox"/> rtb-063dde2a421c737ae	<input type="checkbox"/> No	subnet-0b3b4281829f9d5b0 / my-subnet
VPC	Owner ID	Edge associations
vpc-04e0f94301abcaa77 assignment-vpc	<input type="checkbox"/> 043017053821	-

Routes Subnet associations Edge associations Route propagation Tags

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Creating an instance inside the above created VPC

Step-1: Search for EC2 in the search bar.

The screenshot shows the AWS VPC Management Console interface. The search bar at the top contains the query 'ec2'. The main pane displays search results under 'Services' and 'Features'. The 'EC2' service is listed under 'Services' with a star icon and the description 'Virtual Servers in the Cloud'. Other services like EC2 Image Builder, Recycle Bin, and Amazon Inspector are also listed. The sidebar on the left shows navigation options for VPCs, Virtual private cloud, and other AWS services like IAM and S3. The bottom right corner includes standard AWS footer links for CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

Step-2: Click on *Instances*

The screenshot shows the AWS EC2 Management Console interface. The sidebar on the left has 'Instances' selected, which is highlighted in blue. The main dashboard shows a summary of resources: 0 running instances, 0 auto scaling groups, 0 dedicated hosts, 0 elastic IPs, 0 instances, 1 key pair, 1 load balancer, 0 placement groups, 4 security groups, 0 snapshots, and 0 volumes. A callout box provides information about using the AWS Launch Wizard for Microsoft SQL Server Always On availability groups. The right side of the screen shows account attributes, supported platforms, and settings. The bottom right corner shows system status and connectivity icons.

Step-3: Select *Launch instance*.

The screenshot shows the AWS EC2 Instances page. The left sidebar is expanded, showing categories like EC2 Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), and Images (with sub-option AMIs). The main content area has a heading 'Instances Info' with a search bar and filters (Instance state = running). It displays a message: 'No matching instances found'. Below this is a 'Select an instance' section.

Step-4: Give a name to the instance, select the AMI and select your key pair.

The screenshot shows the 'Launch an instance' wizard. The top navigation bar includes CloudShell, Feedback, and Language. The main steps are: EC2 > Instances > Launch an instance. The 'Launch an instance' step is active. It contains fields for 'Name and tags' (Name: project-instance) and 'Application and OS Images (Amazon Machine Image)' (Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...). On the right, a summary panel shows: Number of instances (Info: 1), Software Image (AMI) (Amazon Linux 2 Kernel 5.10 AMI...), Virtual server type (instance type) (t2.micro), Firewall (security group) (New security group), Storage (volumes) (1 volume(s) - 8 GiB), and a 'Launch instance' button.

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - **required**

Select NVirginia-srkrec

Proceed without a key pair (Not recommended) Default value

NVirginia-srkrec Type: rsa

Virtual server type (instance type)

t2.micro

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more ami-04823729c75214919

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Summary

Number of instances: 1

Launch instance

Step-5: *Edit* the network settings.

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - **required**

NVirginia-srkrec

Network settings

Network: vpc-0046264f562b278ce

Subnet: No preference (Default subnet in any availability zone)

Auto-assign public IP: Info

Enable

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Summary

Number of instances: 1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...read more ami-05548f9cccf47b442

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Launch instance

Step-6: Select your VPC and subnet.

Network settings

VPC - required: vpc-0046264f562b278ce (default)

Subnet: subnet-091c202e89a5bc175 project-subnet

Virtual server type (instance type): t2.micro

Storage (volumes): 1 volume(s) - 8 GiB

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

Network settings

VPC - required: vpc-04e0f94301abcaa77 (assignment-vpc)

Subnet: subnet-091c202e89a5bc175 project-subnet

Virtual server type (instance type): t2.micro

Storage (volumes): 1 volume(s) - 8 GiB

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

Step-7: In the Inbound Security Group rules, remove ssh and select *All traffic* then *Launch instance*.

Description - required Info
launch-wizard-2 created 2023-07-22T15:51:45Z

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type Info Protocol Info Port range Info

ssh TCP 22

Custom Protocol

All TCP

All UDP

All ICMP - IPv4

All ICMP - IPv6

All traffic

ssh

Add security group rule

Source Info Description - optional Info e.g. SSH for admin desktop

0.0.0.0 / 0 ::/0

We recommend to allow all IP addresses to access your instance. We recommend to allow access from known IP addresses only.

Summary

Number of instances Info 1

Software Image (AMI) Amazon Linux 2023 AMI 2023.1.2... read more ami-05548f9cecf47b442

Virtual server type (instance type) t2.micro

Firewall (security group) New security group

Storage (volumes) 1 volume(s) - 8 GiB

Cancel Launch instance Review commands

New EC2 Experience Tell us what you think

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
project-instance	i-00b6cf7779835e433	Running	t2.micro	Initializing	No alarms	us-e

Instance: i-00b6cf7779835e433 (project-instance)

Answer private resource DNS name

Instance type

t2.micro

Auto-assigned IP address

VPC ID

vpc-04e0f94301bcaa77 (assignment-vpc)

Subnet ID

subnet-091c202e89a5bc175 (project-

IAM Role

Elastic IP addresses

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations.

Learn more

Auto Scaling Group name

CloudShell Feedback Language

Establish a peering connection between the two VPC's

Step-1: Create a VPC in N. Virginia region.

The screenshot shows the AWS VPC Management Console interface. The top navigation bar includes the AWS logo, services like IAM, EC2, S3, VPC, Route 53, CloudFront, Lambda, and DynamoDB, and the region selector "N. Virginia". The main content area displays a table titled "Your VPCs (1/2)" with columns for Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. One VPC, "assignment-vpc" (VPC ID: vpc-04e0f94301abcaa77), is listed as Available with IPv4 CIDR 10.199.2.0/24. Below the table, a detailed view for "vpc-04e0f94301abcaa77 / assignment-vpc" is shown with tabs for Details, Resource map, CIDRs, Flow logs, and Tags. The "Details" tab is selected, showing information such as VPC ID, State (Available), DNS hostnames (Disabled), and DNS resolution (Enabled).

Step-2: Change to another region (**ap-southeast-1**) and go to VPC and select **Create VPC**.

The screenshot shows the AWS VPC Management Console interface with the region changed to "ap-southeast-1". The top navigation bar and sidebar are identical to the previous screenshot. The main content area features a "Create VPC" button and a "Launch EC2 Instances" button. A note below states "Note: Your Instances will launch in the US East region." To the right, a dropdown menu lists various AWS regions and their corresponding codes. The "Asia Pacific (Singapore)" region is highlighted in orange, indicating it is the selected region for the current session.

Step-3: Select **VPC only wizard** and give IPV4 CIDR then select **Create VPC**.

VPC settings

Resources to create: **VPC only** (selected)

Name tag - optional: project-vpc

IPv4 CIDR block: **IPv4 CIDR manual input** (selected)

IPv4 CIDR: 10.199.1.0/24

IPv4 CIDR block: No IPv6 CIDR block (selected)

IPv4 CIDR: 10.199.1.0/24

Tenancy: Default

Tags: project-vpc

Your VPCs | VPC Management Console

ap-southeast-1.console.aws.amazon.com/vpc/home?region=ap-southeast-1#CreateVpc.createMode=vpcOnly

Services Search [Alt+S]

IAM EC2 S3 VPC Route 53 CloudFront Lambda DynamoDB

10.199.1.0/24

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)
 Default

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

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="project-vpc"/>
Add tag	
You can add 49 more tags	

Cancel [Create VPC](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Your VPCs | VPC Management Console

ap-southeast-1.console.aws.amazon.com/vpc/home?region=ap-southeast-1#VpcDetails:VpcId=vpc-0d6f555d214f53a4a

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

You successfully created vpc-0d6f555d214f53a4a / project-vpc

VPC > Your VPCs > vpc-0d6f555d214f53a4a

vpc-0d6f555d214f53a4a / project-vpc

[Actions](#)

Details Info	
VPC ID	vpc-0d6f555d214f53a4a
State	Available
DNS hostnames	Disabled
DNS resolution	Enabled
Tenancy	DHCP option set
Default	dopt-0c75a61f2880acf3
Main route table	rtb-08840eb9b235a5f85
Main network ACL	acl-0c71159fa5663d9df
Default VPC	IPv4 CIDR
No	10.199.1.0/24
IPv6 pool	-
IPv6 CIDR (Network border group)	-
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups
Disabled	Owner ID 043017053821

Resource map [New](#) CIDRs Flow logs Tags

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Step-4: Click on **Peering connections**.

The screenshot shows the AWS VPC Management Console. The URL is <https://ap-southeast-1.console.aws.amazon.com/vpc/home?region=ap-southeast-1#VpcDetails>. The left sidebar has a tree view with 'Your VPCs' selected. Under 'Peering connections', there is a link labeled 'Peering connections'. The main content area displays VPC details for 'vpc-0d6f555d214f53a4a'. Below this, there is a 'Resource map' section with tabs for 'VPC', 'Subnets', and 'Route tables'. A modal window titled 'Introducing the VPC resource' is open over the 'VPC' tab.

Step-5: Select **Create peering connection**.

The screenshot shows the 'Peering connections' page in the AWS VPC Management Console. The URL is <https://ap-southeast-1.console.aws.amazon.com/vpc/home?region=ap-southeast-1#PeeringConnections>. The left sidebar shows 'Your VPCs' selected under 'Virtual private cloud', with 'Peering connections' also listed. The main area shows a table with columns: Name, Peering connection ID, Status, Requester VPC, and Acceptor VPC. A message 'No peering connection found' is displayed. At the top right, there is a prominent orange 'Create peering connection' button.

Step-6: Give a name to your peering connection and Select the requester VPC.

Peering connection settings

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

Select a local VPC to peer with
VPC ID (Requester)

Select another VPC to peer with

Account

- My account
- Another account

Peering connection settings

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

Select a local VPC to peer with
VPC ID (Requester)

Select another VPC to peer with

Account

- My account
- Another account

Step-7: Select **My account** as another accepter VPC. Select **Another region** and select the region of your accepter VPC.

VPC CIDR for your AdGEFEEA1714E572-1 (accepter vpc)

Select a region

VPC ID (Acceptor)

VPC ID

Tags

Step-8: Goto your Acceptor VPC and select your VPC. In the Details copy the VPC ID.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0046264f562b278ce	Available	172.31.0.0/16	-
assignment-vpc	vpc-04e0f94301abcaa77	Available	10.199.2.0/24	-

Details

VPC ID vpc-04e0f94301abcaa77	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0b8365933d7172353	Main route table rtb-0050fa3acc34fb9f6	Main network ACL acl-042be2bbeff734011f
Default VPC	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR (Network border)

Step-9: Paste the Acceptor VPC ID and click on **Create peering connection**.

Your VPCs | VPC Management Console > VPC Management Console

Region
This Region (ap-southeast-1)
Another Region
US East (N. Virginia) (us-east-1)

VPC ID (Acceptor)
vpc-04e0f94301abcaa77

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	assignment-project-peering

Add new tag
You can add 49 more tags.

Cancel **Create peering connection**

Your VPCs | VPC Management Console > VPC Management Console

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

A VPC peering connection pcx-0f4ed4a4099e21771 / assignment-project-peering has been requested.
Remember to change your region to us-east-1 to accept the peering connection.

VPC > Peering connections > pcx-0f4ed4a4099e21771

pcx-0f4ed4a4099e21771 / assignment-project-peering		
Actions ▾		
Details <small>Info</small>		
Requester owner ID 043017053821	Acceptor owner ID 043017053821	VPC Peering connection ARN arn:aws:ec2:ap-southeast-1:043017053821:vpc-peering-connection/pcx-0f4ed4a4099e21771
Peering connection ID pcx-0f4ed4a4099e21771	Requester VPC vpc-0d6f555d214f53a4a / project-vpc	Acceptor VPC vpc-04e0f94301abcaa77
Status Initiating Request to 043017053821	Requester CIDRs 10.199.1.0/24	Acceptor CIDRs -
Expiration time Saturday, July 29, 2023 at 10:10:41 GMT+5:30	Requester Region Singapore (ap-southeast-1)	Acceptor Region N. Virginia (us-east-1)

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

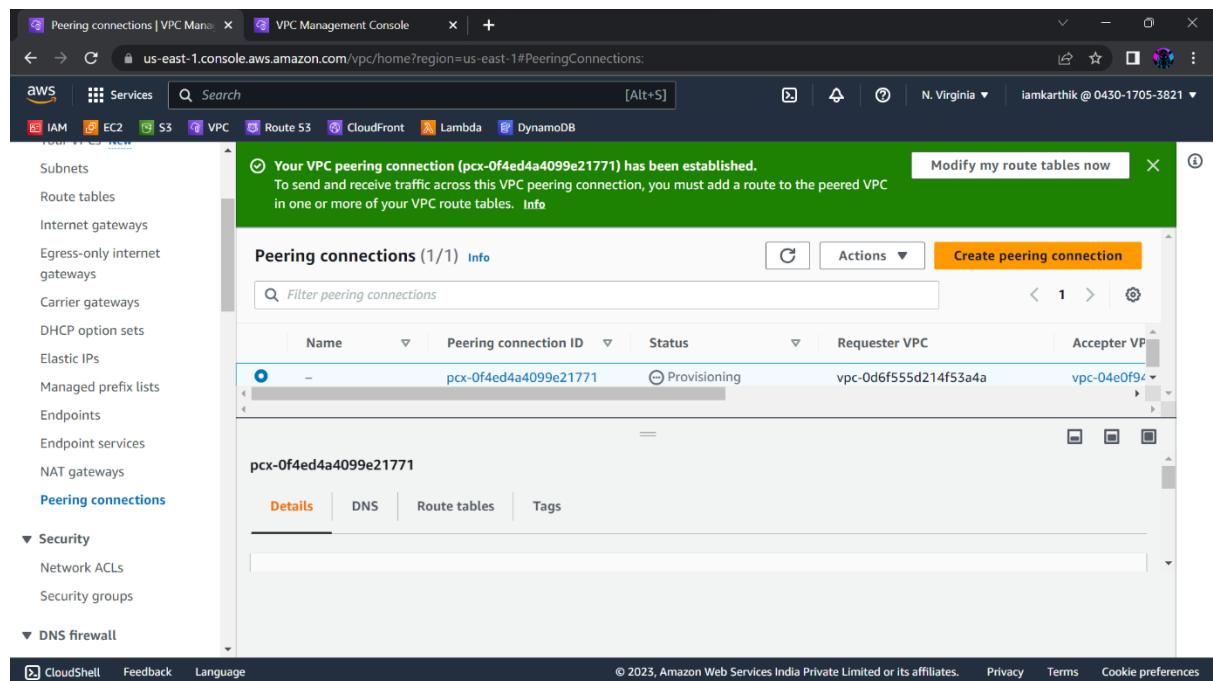
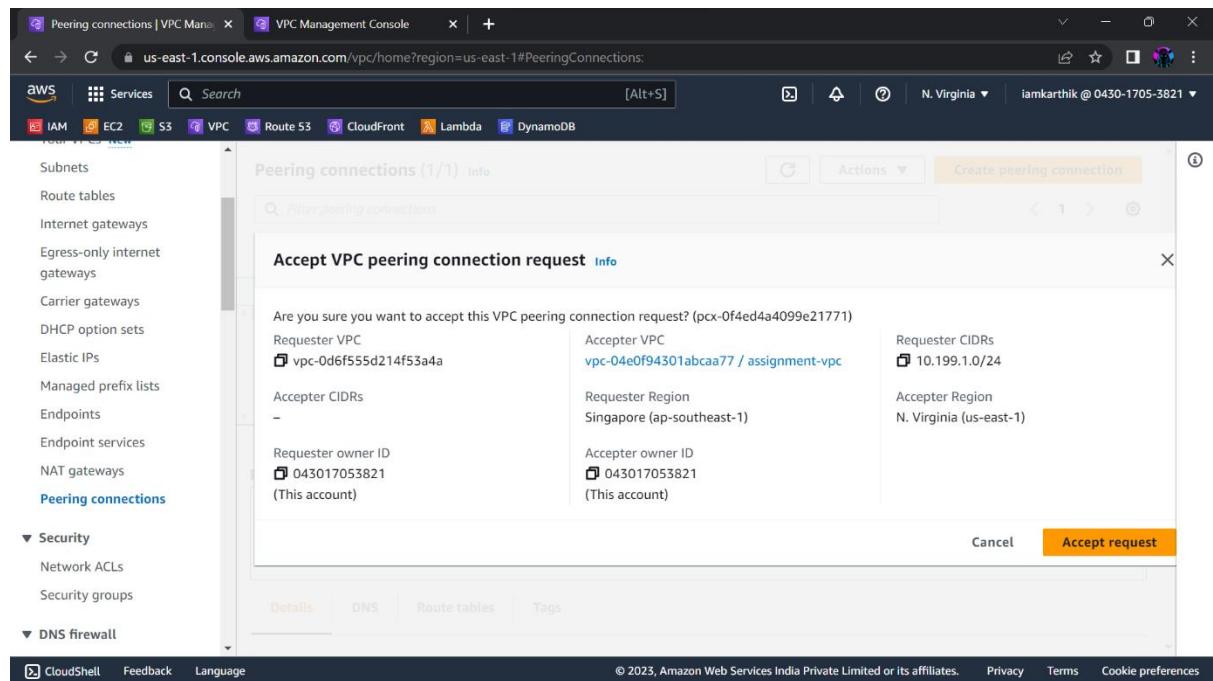
Step-10: Now go to peering connections in Acceptor VPC. There will be a peering request under acceptance.

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
-	pcx-0f4ed4a4099e21771	Pending acceptance	vpc-0d6f555d214f53a4a	vpc-04e0f943c

Step-11: Select the peering request and select **Accept request** in actions.

i Pending acceptance
 You can accept or reject this peering connection request using the 'Actions' menu. You have until Saturday, July 29, 2023 at 10:10:41 GMT+5:30 to accept or reject the request, otherwise it expires.

Step-12: Click on *Accept request*.



The peering connection is now established.

The screenshot shows the AWS VPC Management Console with the URL us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#PeeringConnections. A green success message at the top states: "Your VPC peering connection (pcx-0f4ed4a4099e21771) has been established. To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables. Info". Below this, the "Peering connections (1/1)" section shows a single entry:

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
—	pcx-0f4ed4a4099e21771	Active	vpc-0d6f555d214f53a4a	vpc-04e0f94...

Below the table, there is a section titled "pcx-0f4ed4a4099e21771" with tabs for "Details", "DNS", "Route tables", and "Tags". The "Details" tab is selected. At the bottom of the page, there are links for CloudShell, Feedback, Language, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS VPC Management Console with the URL ap-southeast-1.console.aws.amazon.com/vpc/home?region=ap-southeast-1#PeeringConnections. A green message at the top says: "A VPC peering connection ppx-0f4ed4a4099e21771 / assignment-project-peering has been requested. Remember to change your region to us-east-1 to accept the peering connection." Below this, the "Peering connections (1)" section shows a single entry:

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
assignment-pr...	ppx-0f4ed4a4099e21771	Active	vpc-0d6f555d214f53a4a / pro...	vpc-04e0f94...

Below the table, there is a message: "Select a peering connection above". At the bottom of the page, there are links for CloudShell, Feedback, Language, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

Communicating through private IP's

STEP-1: Add the ROUTE as shown below to SINGAPORE REGION

The screenshot shows the 'Edit routes' page in the AWS VPC console. It lists three routes:

- Destination: 10.0.0.0/16, Target: local, Status: Active, Propagated: No.
- Destination: 0.0.0.0/0, Target: igw-0a7b394ba41abac4c, Status: Active, Propagated: No. A 'Remove' button is visible.
- Destination: 172.0.0.0/16, Target: pcx-03421b16da274ebcf, Status: -, Propagated: No. A 'Remove' button is visible.

Buttons at the bottom include 'Add route', 'Cancel', 'Preview', and a highlighted 'Save changes' button.

STEP-2: PEERING CONNECTION HAS BEEN ESTABLISHED SUCCESSFULLY

The screenshot shows the 'Route tables' page for route table 'rtb-032602e29c035166b'. A green header bar indicates: 'Updated routes for rtb-032602e29c035166b successfully' with a 'Details' link.

Below, the route table details are listed:

- Route table ID: rtb-032602e29c035166b
- Main: Yes
- Owner ID: 381251514697
- Explicit subnet associations: subnet-01423082261283c96 / guru-subnet
- Edge associations: -

Buttons include 'Run Reachability Analyzer' and 'Actions ▾'.

STEP-3: Here is the previously created instances with the VPC's in SINGAPORE REGION
click on connect

The screenshot shows the 'Instances (1/1)' page for the Singapore region. It lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
GURU-INSTANCE-SINGAPORE	i-0bead1424d669def2	Running	t2.micro	Initializing	No alarms	ap-southeast-1c

Buttons include 'Connect', 'Launch instances', and navigation controls.

STEP-4: Here is the previously created instances with the VPC's in VIRGINIA REGION
click on connect

The screenshot shows the 'Instances (1/1)' page for the Virginia region. It lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
GURU-INSTANCE-VIRGINIA	i-046f7b9aa22c6ee28	Running	t2.micro	Initializing	No alarms	us-east-1a

Buttons include 'Connect', 'Launch instances', and navigation controls.

STEP-5: Edit the username as **ROOT** and click **CONNECT**

Connect to your instance i-0bead1424d669def2 (GURU-INSTANCE-SINGAPORE) using any of these options

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
-----------------------------	------------------------	-------------------	---------------------------

Instance ID
i-0bead1424d669def2 (GURU-INSTANCE-SINGAPORE)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
18.143.197.144

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
root

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

STEP-6: Click **CONNECT** in this region

Connect to your instance i-046f7b9aa22c6ee28 (GURU-INSTANCE-VIRGINIA) using any of these options

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
-----------------------------	------------------------	-------------------	---------------------------

Instance ID
i-046f7b9aa22c6ee28 (GURU-INSTANCE-VIRGINIA)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
3.80.110.197

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

STEP-7: Copy the PRIVATE IP of N VIRGINIA REGION

```
  _|_ ( _|_ /   Amazon Linux 2 AMI
  __|_\_|__|_
https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-143 ~]$
```

i-046f7b9aa22c6ee28 (GURU-INSTANCE-VIRGINIA)
Public IPs: 3.80.110.197 Private IPs: 10.0.1.143

STEP-8: Enter the LINUX COMMAND and paste the PRIVATE IP of VIRGINIA REGION

```
  _|_ ( _|_ /   Amazon Linux 2 AMI
  __|_\_|__|_
https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[root@ip-172-0-1-202 ~]# ping 10.0.1.230█
```

STEP-9: Make sure that the PACKAGE LOSS IS 0%**NOW THE VPC's COMMUNICATION IS ESTABLISHED**

```
[root@ip-172-0-1-202 ~]# ping 10.0.1.230
PING 10.0.1.230 (10.0.1.230) 56(84) bytes of data.
64 bytes from 10.0.1.230: icmp_seq=1 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=2 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=3 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=4 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=5 ttl=255 time=21.2 ms
64 bytes from 10.0.1.230: icmp_seq=6 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=7 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=8 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=9 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=10 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=11 ttl=255 time=20.9 ms
64 bytes from 10.0.1.230: icmp_seq=12 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=13 ttl=255 time=20.9 ms
64 bytes from 10.0.1.230: icmp_seq=14 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=15 ttl=255 time=20.9 ms
64 bytes from 10.0.1.230: icmp_seq=16 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=17 ttl=255 time=21.0 ms
64 bytes from 10.0.1.230: icmp_seq=18 ttl=255 time=21.0 ms
^C
--- 10.0.1.230 ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time 17019ms
```