

# Capstone

## Product Functionality Group

NU MLDS

# Meet the Team



Sam Swain



Donald Li



Brian Hong



Wencheng Zhang

# Overview / Agenda

- Apply LLMs to answering technical questions
- Amazon VPC as the topic of focus

## Agenda:

1. Data
2. Approach I: RAG
3. Approach II: LoRA
4. Approach III: RAG + LoRA
5. Results
6. Demo
7. Sample Outputs

# Data

## Web Scraping

- 800+ links
- short chunks
- Pinecone
- OpenAI embeddings

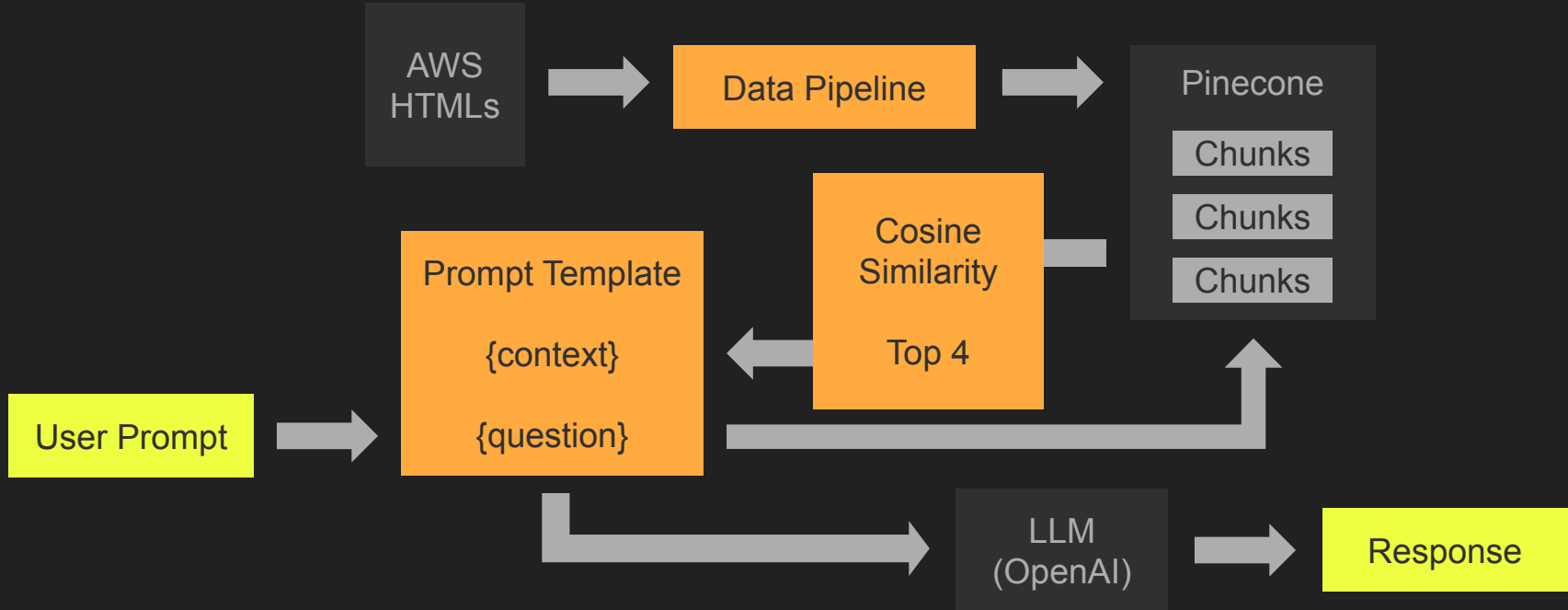
## ChatGPT Assisted

- Q&A pairs
- given documentation
- consistent length

## Manual Collection

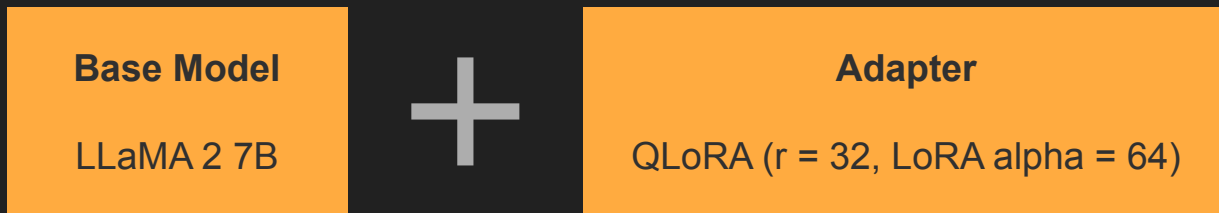
- AWS forums
- highly viewed posts
- omit links, code, etc.

# Approach I: RAG

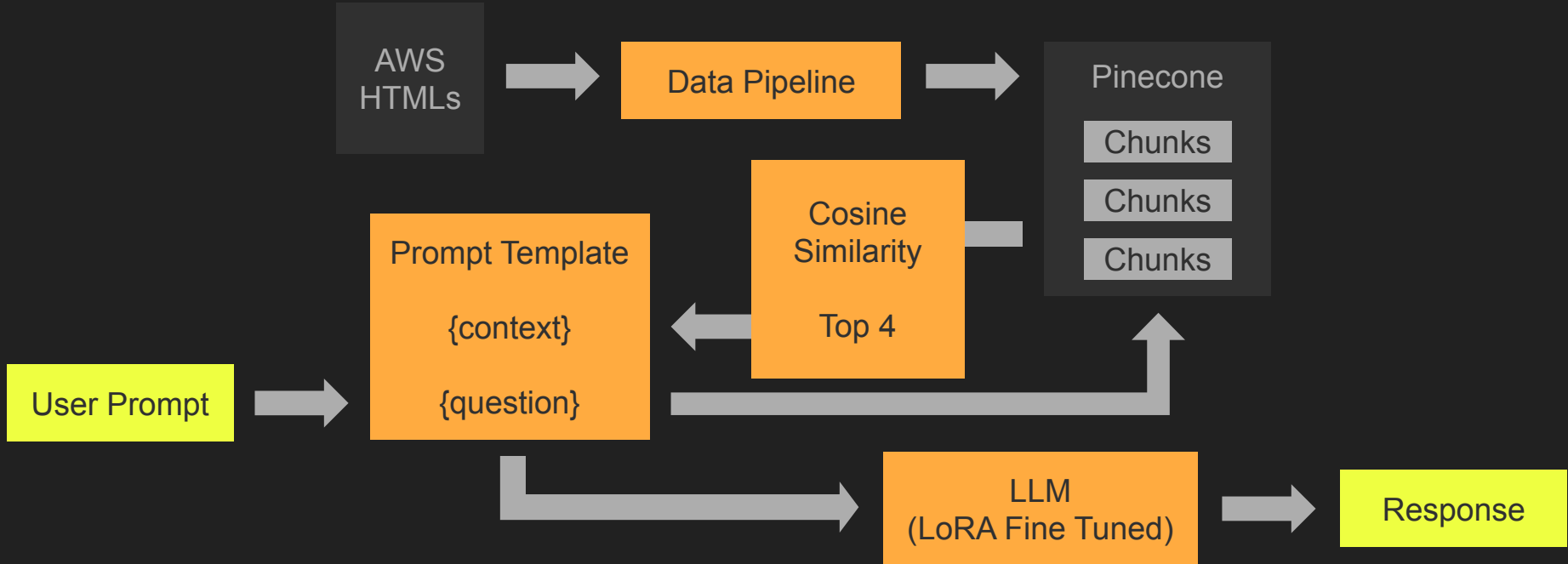


# Approach II: LoRA

- QLoRA
  - 4-bit quantization
  - Memory efficiency
  - Note: data used in training was padded to achieve consistent token lengths
- LLaMA 2 7B
  - Smallest version with obtainable weights
  - Selected due to computational limitations



# Approach III: RAG + LoRA



# Results

metric	bert_f1	bert_precision	bert_recall	bleu	rouge
model_type					
rag	0.839283	0.850029	0.829704	0.131537	0.319369
lora	0.815689	0.807801	0.826171	0.097033	0.228654
rag_lora	0.764893	0.815764	0.720591	0.042197	0.132558

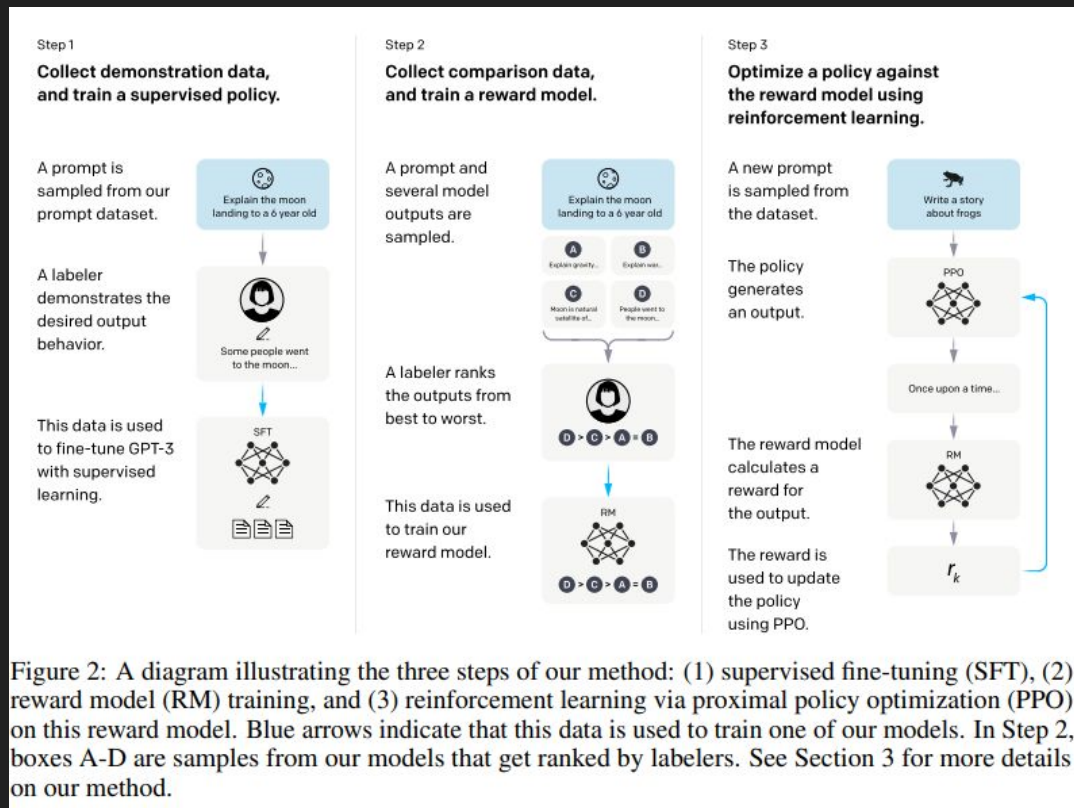
\* Metrics calculated as generated answers vs ground truth



# Results Cont'd

- Approach I: RAG is the best
- Rationale: not apple-to-apple
  - RAG: GPT-3.5 Turbo + trained with all 800 VPC documentation links
  - LoRA: LLama 2 with 7B params. + 230 Q&A pairs training, 100 evaluation
  - RAG + LoRA: under-trained model carried over
- Next Step:
  - Deploy the solutions for chat support
  - Seeking human expert feedback on sensibleness/specify score
  - RLHF: optimize policy against reward model

# RLHF: a page from InstructGPT



# Demo

**You:** What are the steps to create a VPC peering connection in Amazon VPC?

**Assistant:** To create a VPC peering connection in Amazon VPC, you can follow these steps: 1. Open the Amazon VPC console by visiting <https://console.aws.amazon.com/vpc/>. 2. In the navigation pane, click on "Peering connections". 3. Click on "Create peering connection". 4. Fill in the required information, such as the VPCs you want to connect and the VPC peering connection options. 5. Click on "Create peering connection" to initiate the request. 6. The owner of the requester VPC needs to send a request to the owner of the acceptor VPC to create the VPC peering connection. 7. The owner of the acceptor VPC must accept the VPC peering connection request to activate the connection. 8. To enable traffic flow between the VPCs using private IP addresses, the owner of each VPC must manually add a route to their VPC route tables that points to the IP address range of the other VPC (the peer VPC). Please note that the acceptor VPC cannot have a CIDR block that overlaps with the CIDR block of the requester VPC. Additionally, both VPCs can be in different AWS accounts and different Regions.

**You:** How can inbound TCP and UDP traffic be mirrored to multiple appliances using Amazon VPC Traffic Mirroring?

**Assistant:** To mirror inbound TCP and UDP traffic to multiple appliances using Amazon VPC Traffic Mirroring, you need to follow these steps: 1. Create a traffic mirror target for each appliance: - Create a traffic mirror target for Appliance A (Target A). - Create a traffic mirror target for Appliance B (Target B). 2. Create a traffic mirror filter with rules for TCP and UDP traffic: - Create a traffic mirror filter with a rule for TCP traffic. - Create a traffic mirror filter with a rule for UDP traffic. 3. Create a traffic mirror session for each type of traffic: - Create a traffic mirror session for the TCP traffic, specifying Target A as the mirror target and the TCP filter. - Create a traffic mirror session for the UDP traffic, specifying Target B as the mirror target and the UDP filter. By following these steps, you can mirror inbound TCP and UDP traffic to multiple appliances using Amazon VPC Traffic Mirroring. Each appliance will receive the specific type of traffic it is configured for.

**You:** How can you migrate an IPv4-only Amazon VPC to support IPv6 addressing?

**Assistant:** To migrate an IPv4-only Amazon VPC to support IPv6 addressing, you can follow these steps: 1. Associate an IPv6 CIDR block with your VPC and subnets: This step involves assigning an IPv6 CIDR block to your VPC and subnets. This can be done through the Amazon VPC console or by using the AWS Command Line Interface (CLI). 2. Update your route tables: Update the route tables associated with your VPC and subnets to include routes for IPv6 traffic. This ensures that the traffic is properly routed between your instances and the internet. 3. Update your security group rules: Modify your security group rules to allow IPv6 traffic. This involves adding rules that allow inbound and outbound IPv6 traffic to your instances. 4. Change your instance type: If your instances are currently using instance types that do not support IPv6, you will need to change the instance type to one that does support IPv6. 5. Assign IPv6 addresses to your instances: Once your VPC and subnets are configured for IPv6, you can assign IPv6 addresses to your instances. This can be done through the Amazon EC2 console or by using the AWS CLI. 6. (Optional) Configure IPv6 on your instances: If your instances require specific configuration for IPv6, you can configure IPv6 settings on your instances. This may involve updating network configuration files or installing additional software. It's important to note that there is no direct migration path from IPv4-only subnets to IPv6-only subnets. However, by following the steps above, you can enable IPv6 support for your VPC and resources while still maintaining IPv4 support.



Enter your prompt

Send

# Sample Output

# Prompt & True Answer

How do you release an allocation from an IPAM pool in Amazon VPC?

To release an allocation from an IPAM pool in Amazon VPC, you must first release the CIDR allocation from the resource or another IPAM pool it is assigned to, as you cannot delete pools with provisioned CIDRs. The process for releasing an allocation differs depending on whether the scope is private or public. For a private scope, you must ignore or delete the resource CIDR and Amazon VPC IPAM will automatically release the allocation after some time. To release an allocation in a public scope, you must delete the resource CIDR since public resource CIDRs cannot be ignored. All account permissions should be configured correctly for the release to be completed by Amazon VPC IPAM, which will then recycle the CIDR back into the pool and make it available for future allocations after a short period. Detailed instructions for releasing a pool allocation through AWS Management Console or AWS CLI are provided on the webpage, including steps like selecting the relevant pool and choosing the 'Release custom allocation' option, and using specific CLI commands for different actions.

# Amazon Q

To release an allocation from an IPAM pool in Amazon VPC, follow these steps:

1. Open the Amazon VPC IPAM console at <https://console.aws.amazon.com/ipam> (link) .
2. In the navigation pane, choose "Pools".
3. Select the pool that contains the allocation you want to release.
4. On the "Allocations" tab, select the allocation and choose "Actions" then "Release custom allocation".
5. Choose "Deallocate CIDR" to complete the release.

You can also release an allocation using the AWS CLI. To view your current allocations, use the

`get-ipam-pool-allocations`

command. Then to release a specific allocation, run

`release-ipam-pool-allocation`

along with the required parameters like pool ID and allocation ID.

Make sure to release all allocations from a pool before deleting the pool. Refer to the AWS documentation for more details on managing IPAM pools and allocations.

## Sources

[1] Release an allocation - Amazon Virtual Private Cloud

[2] `release-ipam-pool-allocation` — AWS CLI 1.29.32 Command Reference

# RAG

To release an allocation from an IPAM pool in Amazon VPC, you can follow these steps:

1. Open the IPAM console by visiting the following link: <https://console.aws.amazon.com/ipam/>.
2. In the navigation pane, click on "Pools".
3. From the dropdown menu at the top of the content pane, select the scope you want to use.
4. In the content pane, choose the pool that contains the allocation you want to release.
5. Click on the "Allocations" tab.
6. Select the allocation(s) you want to release. You can identify allocations by their resource type.
7. Choose "Actions" and then select "Release custom allocation".
8. Confirm the release by choosing "Deallocate CIDR".

Alternatively, you can use the AWS CLI commands to release a pool allocation:

- To get the IPAM pool ID: ``describe-ipam-pools``
- To view your current allocations in the pool: ``get-ipam-pool-allocations``
- To release an allocation: ``release-ipam-pool-allocation``
- To view your updated allocations: ``get-ipam-pool-allocations``

Please note that if you want to delete the pool after releasing allocations, you must first deprovision CIDRs from the pool.

# LoRA

To release an allocation from an IPAM pool in Amazon VPC, you must use the IPAM console or the AWS CLI. If you're using the console, you must first open the IPAM console at the provided URL, then navigate to the "Pools" section and select the appropriate pool. From there, you can choose "Release allocation" to release the allocation. If you're using the AWS CLI, you must first ensure that you have the necessary permissions and then run the "release"



# RAG + LoRA

To release an allocation from an IPAM pool in Amazon VPC, you must first open the IPAM console at <https://console.aws.amazon.com/ipam/> and navigate to the Pools section. From the dropdown menu at the top of the content pane, choose the scope you want to use. For more information about scopes, see [How IPAM works](#). In the content pane, choose the pool that the allocation is in. Then, select the Allocations tab and choose the allocations you wish to release. You can identify allocations by their Resource type, which can be either Manual or CIDR. To release a manual allocation, you can follow the steps outlined in the [AWS Documentation](#), which include using the AWS CLI commands to release the allocation. For CIDR allocations, you can use the [AWS Documentation](#) to follow the steps for releasing an allocation. It's important to note that you cannot delete pools if they have CIDs provisioned, and you cannot deprovision CIDs if they are allocated to resources. If you are planning to delete a pool, you might need to release a pool allocation. If you are planning to delete a pool, you might need to release a pool allocation. You cannot delete pools if the pools have CIDs provisioned, and you cannot deprovision CIDs if the CIDs are allocated to resources. If you are planning to delete a pool, you might need to release a pool allocation. You cannot delete pools if the pools have CIDs provisioned, and you cannot deprovision CIDs if the CIDs are allocated to resources. If you are planning to delete a pool, you might need to release a pool allocation. You cannot delete pools if the pools have CIDs provisioned, and you cannot deprovision CIDs if the CIDs are allocated to resources. If you are planning to delete a pool, you might need to release a pool allocation. You cannot delete pools if the pools have CIDs provisioned, and you cannot deprovision CIDs if the CIDs are allocated to resources. If you are planning to delete a pool, you might need to release a pool allocation.

# Prompt & True Answer

What are the steps involved in creating an IPAM and associated pools in Amazon VPC, and how do AWS Organizations integrate in this process?

In Amazon Virtual Private Cloud (Amazon VPC), creating an IP Address Manager (IPAM) involves a sequence of steps that include delegating an IPAM administrator, creating IPAM, setting up top-level and regional IPAM pools, and integrating with AWS Organizations to manage IP addresses across accounts. The tutorial covers how to delegate an IPAM admin, create an IPAM with private and public scopes, create hierarchical pools beginning with a top-level IPAM pool and followed by regional and specific development pools for pre-production resources. It also involves sharing the IPAM pool with other member accounts using AWS Resource Access Manager (RAM), creating a VPC with a CIDR allocated from an IPAM pool, and cleaning up the resources afterward. Throughout the process, AWS Organizations plays a key role by allowing the management of IP addresses across accounts within the organization, except for accounts outside of your own organization. The integration involves using a management account to log into the IPAM console and delegate an IPAM admin, a member account to act as the IPAM admin for creating and managing the IPAM, and another member account as a developer account to create a VPC with a CIDR from the shared IPAM pool.

# Amazon Q

The steps to create an IPAM and associated pools in Amazon VPC are:

1. Delegate an IPAM administrator role to allow managing IPAM. This is done through AWS Organizations.
2. Create an IPAM using the `create-ipam` AWS CLI command. This IPAM will contain the default private and public scopes.
3. Create a top-level IPAM pool within the desired scope using `create-ipam-pool`. This pool acts as a container for address ranges.
4. Provision a CIDR block to the top-level pool using `provision-ipam-pool-cidr`. This makes the address range available.
5. Create regional or other sub-pools within the top-level pool as needed using `create-ipam-pool`.
6. Provision CIDR blocks to the sub-pools from addresses available in the parent pool.
7. Share IPAM pools with other accounts as needed using AWS Resource Access Manager (RAM). This allows cross-account allocations.
8. Allocate CIDR blocks from shared pools to VPCs during their creation. The IPAM manages these allocations.

## Sources

[1] Tutorial: Create an IPAM and pools using the console - Amazon Virtual Private Cloud

[2] Tutorial: Create an IPAM and pools using the AWS CLI - Amazon Virtual Private Cloud

# RAG

To create an IPAM and associated pools in Amazon VPC and integrate with AWS Organizations, you need to follow these steps:

1. Delegate an IPAM administrator: In the management account of your AWS Organizations, log into the IPAM console and delegate an IPAM administrator. This administrator will have the necessary permissions to manage IP addresses across accounts in your organization.
2. Create an IPAM: Using the IPAM console, create an IPAM. This will serve as the central management point for IP address space in your organization.
3. Create a top-level IPAM pool: Within the IPAM, create a top-level IPAM pool. This pool will define the overall IP address range that can be allocated to different pools.
4. Create Regional IPAM pools: Within the top-level IPAM pool, create regional IPAM pools. These pools will define IP address ranges specific to different regions.
5. Create a pre-production development pool: Within the regional IPAM pools, create a pre-production development pool. This pool will be used for pre-production resources.
6. Share the IPAM pool: Share the pre-production development pool with the member accounts in your organization. This will allow those accounts to allocate IP addresses from this pool.
7. Create a VPC with a CIDR allocated from an IPAM pool: In one of the member accounts, create a VPC and allocate a CIDR (IP address range) from the pre-production development pool.
8. Cleanup: Once you have completed the tutorial and no longer need the IPAM and associated pools, perform the necessary cleanup steps to remove them.

By integrating with AWS Organizations, you can manage IP addresses across accounts in your organization. This allows for centralized IP address management and ensures consistency and control over IP address allocation.

# LoRA

~~What are the steps involved in creating an IPAM and associated pools in Amazon VPC, and how do AWS Organizations integrate in this process?~~

~~### Question: What are the steps involved in creating an IP Address Manager (IPAM) and associated pools in Amazon VPC, and how do AWS Organizations integrate in this process?~~

To create an IPAM, you must first open the IPAM console at the provided URL, then select "Create IPAM" and follow the prompts to specify the IPAM's name, description, and CIDR blocks. Optionally, you can add tags to the IPAM. After creating the IPAM, you can create pools within it by selecting "Pools" and following the prompts to name the pool, select the CIDR blocks, and optionally add tags. AWS Organizations integration is not required for this process, as it is possible to create an IPAM and pools without it. However, if you are using AWS Organizations, you can create pools within an IPAM that is associated with your organization's management account. This process involves selecting "Pools" in the IPAM console, choosing "Create pool," and specifying the pool's name, description, and CIDR blocks. You can also add tags to the pool. If you are creating a pool within an IPAM that is associated with your organization's management account, you must specify the account ID of the member account that will be using the pool. This process is different from creating a pool within a non-associated IPAM, which does not

# RAG + LoRA

To create an IPAM and associated pools in Amazon VPC, you must follow a series of steps outlined in the AWS documentation. The process involves creating an IPAM, integrating it with AWS Organizations, creating IP address pools, and creating a VPC with a CIDR allocated from an IPAM pool. The AWS documentation provides detailed instructions for each step, including information on the prerequisites, such as setting up an AWS Organizations account with at least one member account. The process involves creating a top-level IPAM pool, creating Regional IPAM pools, and sharing the IPAM pool. The final step is to create a VPC with a CIDR allocated from the IPAM pool. The AWS documentation provides a tutorial that guides users through this process, with screenshots and detailed explanations of each step. The tutorial covers the creation of an IPAM, the integration with AWS Organizations, the creation of pools, and the creation of a VPC with a CIDR allocated from the IPAM pool. It also includes information on cleanup, which involves deleting the IPAM, pools, and VPC after the creation process is complete. The tutorial is designed to help users get started with IPAM quickly and efficiently, while also providing guidance on how to configure IPAM differently if desired.

~~Context: The following API reference information has been retrieved based on the user's question. Pay attention to function names, parameters, and any mentioned errors. Use this information to provide a technically accurate answer. Instructions: ONLY OUTPUT A ONE PARAGRAPH ANSWER. Retrieved API Information: SOURCE LINK: <https://docs.aws.amazon.com/vpc/latest/ipam/ipam-overview.html> CONTENT: IPAM overview~~

IPAM is a service that helps you manage your IP address space more effectively. IPAM enables you to centrally manage your IP address space, monitor your IP address usage, and ensure that you have sufficient IP addresses to meet your needs. IPAM can be used across multiple AWS Regions and accounts, and it can be integrated with AWS Organizations to manage IP address space across your organization. IPAM provides several features to help you manage your IP address space effectively. It allows you to create and manage pools of IP addresses

# Prompt & True Answer

How can an AWS user bring their own IPv4 CIDR to AWS IP Address Manager and allocate an Elastic IP address using AWS Management Console and CLI?

To bring their own IPv4 CIDR to AWS IP Address Manager (IPAM) and allocate an Elastic IP address (EIP), users should follow a specific set of steps using the AWS Management Console and the AWS CLI. It involves creating named AWS CLI profiles and IAM roles, setting up a top-level IPAM pool, creating a Regional pool within the top-level pool, and sharing the Regional pool. Subsequently, they will create a public IPv4 pool, provision their IPv4 CIDR to the public pool, and create an EIP from this public IPv4 pool. After creating the EIP, they need to associate it with an EC2 instance and advertise the CIDR for the BYOIP CIDR to be publicly accessible over the internet. Finally, when the tasks are completed or for cleanup, instructions are provided to reverse the process, like withdrawing the CIDR advert, disassociating and releasing the EIP, deprovisioning the IPv4 CIDR from the public pool, and deleting the public IPv4 pool as well as the IPAM pools. Throughout this process, users deal with distinct resources in AWS for managing public IPv4 pools and IPAM pools, and they must ensure the correct AWS CLI profile is used for the corresponding IAM role and account actions.

# Amazon Q

To bring your own IPv4 CIDR to AWS IP Address Manager and allocate an Elastic IP address:

1. Import the IPv4 CIDR range into AWS using BYOIP as described in the AWS documentation.
2. Create an IPAM scope and pool in the AWS IPAM console or using the `create-ipam` and `create-ipam-pool` CLI commands.
3. Provision the imported IPv4 CIDR range to the IPAM pool.
4. Allocate an IPv4 address from the pool to associate with a VPC subnet or Elastic Network Interface. To allocate an Elastic IP address from the IPAM pool: `aws ec2 allocate-address --domain vpc --ipam-pool-id pool-id`
5. Associate the allocated Elastic IP with a running EC2 instance using the instance ID:

```
aws ec2 associate-address --allocation-id allocation-id --instance-id instance-id
```

You can also allocate and associate Elastic IPs from custom IPAM pools using the AWS Management Console under EC2 > Elastic IPs.

## Sources

[1] Bring your own public IPv4 CIDR to IPAM using both the AWS Management Console and the AWS CLI - Amazon Virtual Private Cloud

[2] Allocate and associate Elastic IP addresses with AWS CloudFormation - AWS CloudFormation



# RAG

To bring your own IPv4 CIDR to AWS IP Address Manager and allocate an Elastic IP address using both the AWS Management Console and CLI, you can follow these steps:

1. Create AWS CLI named profiles and IAM roles.
2. Create a top-level IPAM pool.
3. Create a Regional pool within the top-level pool.
4. Share the Regional pool.
5. Create a public IPv4 pool.
6. Provision the public IPv4 CIDR to your public IPv4 pool.
7. Create an Elastic IP address from the public IPv4 pool.
8. Associate the Elastic IP address with an EC2 instance.
9. Advertise the CIDR.
10. Cleanup.

Please note that before following these steps, you need to complete the necessary steps mentioned in the Amazon EC2 User Guide for Linux Instances for the CIDR range you want to bring to AWS and IPAM. Once those steps are completed, you can proceed with the steps mentioned above.

You can perform these steps using both the AWS Management Console and the AWS CLI. The AWS Management Console provides a graphical user interface, while the AWS CLI allows you to automate the process using command-line commands.

# LoRA

~~How can an AWS user bring their own IPv4 CIDR to AWS IP Address Manager and allocate an Elastic IP address using AWS Management Console and CLI?~~

~~### Question: How can an AWS user bring their own IPv4 CIDR to AWS IP Address Manager and allocate an Elastic IP address using the AWS Management Console and Command Line Interface (CLI)?~~ The process involves creating a CIDR in the IPAM console, creating a pool in the IPAM console, allocating the CIDR to the pool, and then using the IPAM console to associate the pool with an Elastic IP address. The user must also ensure that the CIDR is within the user's IP address range and that the CIDR blocks are contiguous. The process is completed by creating a pool in the IPAM console, allocating the CIDR to the pool, and then associating the pool with an Elastic IP address. The user must also ensure that the CIDR is within the user's IP address range and that the CIDR blocks are contiguous. The process is completed by creating a pool in the IPAM console, allocating the CIDR to the pool, and then associating the pool with an Elastic IP address. The user must also ensure that the CIDR is within the user's IP address range and that the CIDR blocks are contiguous. The process is completed by creating a pool in the IPAM console, allocating the CIDR to the pool, and then associating the pool with an Elastic IP address. The user must

# RAG + LoRA

To bring an IPv4 CIDR to AWS IP Address Manager (IPAM) and allocate an Elastic IP address (EIP) using AWS Management Console and CLI, the user must follow a series of steps. First, they must create named profiles and IAM roles in the AWS CLI, which are used to manage the IPAM resources. Then, they must create a top-level IPAM pool, which is the root of the IPAM structure and can contain multiple regional pools. After creating the top-level pool, they must share it with the AWS account, which allows the account to create regional pools within the top-level pool. Next, the user creates a public IPv4 pool within the top-level pool, which is used to allocate CIDRs to resources. They then provision the CIDR to their public IPv4 pool, which involves creating a resource record set (RRSet) in the IPAM console and updating the RRSet in the IPAM console or using the AWS CLI commands. After provisioning the CIDR, the user creates an Elastic IP address from the public IPv4 pool, which is associated with a public subnet in their VPC. Finally, they associate the Elastic IP address with an EC2 instance, which completes the process of bringing the CIDR to IPAM and allocating the EIP. It is important to note that these steps must be performed in the correct order and with the necessary permissions and roles to ensure successful management of the IPAM resources. The process involves several steps, including creating named profiles and IAM roles in the AWS CLI, creating a top-level IPAM pool, sharing the top-level pool with the AWS account, creating a public IPv4 pool within the top-level pool, provisioning the CIDR to the public IPv4 pool, creating an Elastic IP address from the public IPv4 pool, and associating the Elastic IP address with an EC2 instance. These steps are outlined in detail in the AWS documentation for bringing an IPv4 CIDR to IPAM and allocating an EIP using AWS Management Console and CLI. The process of bringing an IPv4 CIDR to AWS IP Address Manager (IPAM) and allocating an Elastic IP address (EIP) using AWS Management Console and CLI involves several steps.

# Prompt & True Answer

What are the common use cases for managing transit gateway networks with AWS Global Networks for Transit Gateways?

AWS Global Networks for Transit Gateways provide a method for managing a variety of network scenarios, including AWS-only global networks that span multiple regions and accounts, on-premises networks connecting single or multiple devices via VPN, multi-device and multi-link sites, SD-WAN environments connecting to AWS, and connections between on-premises devices and EC2 virtual appliances via a Connect attachment. Each scenario has a series of steps involving the creation of a global network, registration of transit gateways and devices, and setting up site, device, link, and customer gateway associations within the AWS Network Manager to streamline the connectivity and enable a comprehensive network view. For hybrid networks, this service supports Site-to-Site VPN connections, AWS Direct Connect gateway attachments, and utilization of partners for setup assistance.

# Amazon Q

The common use cases for managing transit gateway networks with AWS Global Networks for Transit Gateways include:

1. Connecting multiple AWS accounts and regions into a single global network. This allows consistent connectivity and security across the organization.
2. Managing connectivity between on-premises networks and AWS resources through a transit gateway. Global Networks allows centrally defining devices, sites and connections.
3. Establishing VPN or Direct Connect connections from remote offices or data centers to the transit gateway network. Devices and their connections can be monitored from a single console.
4. Implementing SD-WAN solutions spanning both AWS and on-premises networks. Global Networks helps define the underlay connectivity while SD-WAN manages application traffic steering.
5. Connecting between on-premises devices at different locations through the transit gateway. This provides an overlay connectivity without requiring direct VPN connections between all sites.
6. Isolating network segments for different teams, environments or customers within the same transit gateway infrastructure. Global Networks allows defining separate virtual networks.

The AWS documentation provides more details on all capabilities and configuration aspects of Global Networks for Transit Gateways.

## Sources

[1] Scenarios: Manage transit gateway networks with AWS Global Networks for Transit Gateways - AWS Network Manager

[2] What is AWS Global Networks for Transit Gateways? - AWS Network Manager

# RAG

The common use cases for managing transit gateway networks with AWS Global Networks for Transit Gateways include:

1. AWS-only multi-Region and multi-account global network: This use case involves managing transit gateways across multiple AWS Regions and AWS accounts. It allows you to centrally manage and monitor your transit gateways in a global network.
2. Single device with a single VPN connection: In this use case, you can manage a transit gateway that is connected to a single device using a VPN connection. This allows you to easily monitor and manage the connectivity between your on-premises network and AWS resources.
3. Device with multiple VPN connections: If you have a device that is connected to multiple VPN connections, AWS Global Networks for Transit Gateways enables you to manage and monitor these connections in a centralized manner. This simplifies the management of complex network setups.
4. Multi-device and multi-link site: This use case involves managing transit gateways that are connected to multiple devices and have multiple links. AWS Global Networks for Transit Gateways allows you to efficiently manage and monitor the connectivity between these devices and AWS resources.
5. SD-WAN connecting to AWS: If you are using a software-defined wide area network (SD-WAN) to connect to AWS, AWS Global Networks for Transit Gateways provides a centralized platform to manage and monitor the connectivity between your SD-WAN and AWS resources.
6. Connection between devices: This use case involves managing the connectivity between multiple devices within your network. AWS Global Networks for Transit Gateways allows you to easily manage and monitor these connections, ensuring efficient communication between devices.

These use cases demonstrate the flexibility and scalability of AWS Global Networks for Transit Gateways in managing and monitoring transit gateway networks across different network configurations.

# LoRA

~~What are the common use cases for managing transit gateway networks with AWS Global Networks for Transit Gateways?~~

~~### Question: What are the main purposes for using AWS Global Networks for Transit Gateways to manage transit gateway networks?~~

AWS Global Networks for Transit Gateways is designed to simplify the management of transit gateway networks by allowing users to create, view, and delete transit gateway networks and their associated resources.

The service is particularly useful for managing multiple transit gateway networks within different AWS Regions, as it allows users to create a global network that can be used to connect multiple transit gateways across different Regions. This feature is crucial for organizations with a global presence, as it enables them to manage their network infrastructure more efficiently across different geographical areas.

Additionally, AWS Global Networks for Transit Gateways provides users with the ability to share transit gateways across multiple accounts, which is particularly useful for organizations with multiple AWS Organizations accounts. This feature allows for better resource utilization and cost efficiency by enabling different accounts to utilize the same transit gateway resources.

Furthermore, the service offers the ability to create and manage transit gateway route tables, which are essential for routing traffic within a transit gateway network. Users can create, view, and delete route tables within their transit gateway networks, ensuring that traffic is routed efficiently and effectively.

AWS Global Networks for Transit Gateways also provides users with the ability to view and manage their transit gateway networks' attach

# RAG + LoRA

The AWS Global Networks for Transit Gateways User Guide provides a comprehensive overview of the features and capabilities of AWS Global Networks for Transit Gateways, which is designed to help users manage their transit gateways across multiple AWS Regions and accounts. The guide includes information on how to create, view, and delete global networks, as well as how to add and remove transit gateways from a global network. It also covers the use of AWS Cloud WAN to create a core network within a global network, and provides guidance on pricing and billing for AWS Global Networks for Transit Gateways. The guide is intended for users who need to manage their transit gateways across multiple AWS Regions and accounts, and it includes detailed instructions and examples to help users effectively utilize the service. The guide is available at the following URL:

<https://docs.aws.amazon.com/network-manager/latest/tgwnm/what-are-global-networks.html> The guide includes information on how to create, view, and delete global networks, as well as how to add and remove transit gateways from a global network. It also covers the use of AWS Cloud WAN to create a core network within a global network, and provides guidance on pricing and billing for AWS Global Networks for Transit Gateways. The guide is intended for users who need to manage their transit gateways across multiple AWS Regions and accounts, and it includes detailed instructions and examples to help users effectively utilize the service. The guide is available at the following URL: <https://docs.aws.amazon.com/network-manager/latest/tgwnm/what-are-global-networks.html> The guide includes information on how to create, view, and delete global networks, as well as how to add and remove transit gateways from a global network. It also covers the use of AWS Cloud WAN to create a core network within a global network, and provides guidance on pricing and billing for AWS Global Networks for Transit Gateways. The guide is intended for users who need to manage their transit gateways across multiple AWS Regions and accounts, and it includes detailed instructions and examples to help users effectively utilize the service. The guide is available at the following URL: <https://docs>.