Shared AWS VPC FAQs

Frequently Asked Questions

FAQs about the **Multitenant Subnets** and the **Exclusive Use Subnets** option within the Shared VPC offering.

Basics

Is the VPC really shared?

No, the VPC itself isn't shared, but we use the "shared VPC" term as a shortcut and generalization to make this easier to discuss.

- In the **Multitenant Subnets** option, all the multitenant subnets are shared with all opted-in Cornell AWS customers.
- In the **Exclusive Use Subnets** option, each set of exclusive use subnets is shared with a customer-defined set of AWS accounts.

How are the VPC subnets shared?

The subnets in the VPC are shared to your Cornell AWS account using the AWS Resources Access Manager.

Getting Started

Do I need an AWS account to use the Shared VPC offering?

Yes, you still need a Cornell AWS account to use either Shared VPC option. When you opt-in to use the Shared VPC, you get visibility of and permission to deploy resources to its subnets using your Cornell AWS account.

Can I still manage and use other VPCs if I opt into using the Shared VPC?

Yes, you can continue to create and manage custom VPCs in your Cornell AWS account even after you opt in to use either of the Shared VPC options. However, note that you will not be able to peer your custom VPCs to the Shared VPC.

I have a Cornell AWS account without Cornell networking. How do I opt-in to use the Shared VPC offering?

Contact Cloud Support.

I already have a Cornell Standard VPC in my AWS account. Can I opt-in to use the Shared VPC?

We encourage Cornell AWS customers to transition to using the Shared VPC offerings if they are looking for simplicity and cost-effectiveness. Such customers can opt-in to use the Shared VPC and vacate their Cornell Standard VPC, which would be decommissioned. Contact Cloud Support if you are in this position.

Security and Access

How do I access EC2 instances running on Shared VPC subnets?

There are three options to connect to EC2 instances deployed to a Shared VPC subnet:

- Connect to the the Cornell VPN and then use SSH or Windows Remote Desktop to access your instance using its private IPv4 address. The Shared VPC Network ACL allows all traffic from clients connected to the Cornell VPN, but you will need to ensure that security groups attached to your instance allows this network traffic.
- Use the AWS Systems Manager Session Manager to connect. This method requires that your instance be configured to support the Session Manager, and requires you to have specific IAM privileges to use Systems Manager actions, but it bypasses all network-based security controls.
- You can create an EC2 Instance Connect Endpoint and use EC2 Instance Connect access your instance. Please contact the Cloud Team prior to taking this pathway because they may be able to offer centralized EC2 Instance Connect Endpoints, alleviating the burden of managing these Endpoints yourself.

Where do resources deployed to the Shared VPC subnets reside?

From a management and financial standpoint, the resources you deploy to the Shared VPC subnets reside in your AWS account. You have full access to manage the resources via the AWS console or APIs, and you have full responsibility to pay for those resources via the standard Cornell AWS billing process.

From a networking perspective, those resources reside in the Shared VPC even though the VPC is owned by another Cornell AWS account.

Can other Cornell AWS accounts access the resources I deploy to the Shared VPC?

No. The resources deployed to the Shared VPC are visible and manageable only from the AWS account from which they were created.

> ⚠ From a network perspective, your resources are as accessible to other resources on the Cornell network (including other resources deployed to the Shared VPC) as you allow (via settings in the Security Groups you apply to your resources).

Who is responsible for security in the Shared VPC?

While the CIT Cloud team manages the Network ACL associated with the Shared VPC, you are completely responsible for managing the overall network access to the resources you deploy to the Shared VPC (e.g., by using Security Groups and host firewalls) and to managing the resources themselves (e.g, EC2 instances) according to best practices and Cornell policy.

Do my resources deployed to the Shared VPC automatically have access to a target resource which is also on the Cornell network?

From an access (reachability) perspective, a resource deployed to the Shared VPC is no different from any other AWS resource deployed to a VPC connected to the Cornell network. You may still have to work with the team that manages the target resource to allow your resource to access the target.

Costs and Pricing

Am I charged for using the Shared VPC?

You are charged for the resources you deploy to the Shared VPC just like you would be charged for such resources deployed to a VPC that you owned. You are also charged for the network traffic (bandwidth) attributable to those resources, again as if you deployed them to a VPC you owned.

However, the overhead costs of the VPC (e.g. NAT Gateway costs, VPC Flow Log costs) are not charged to or cost-shared by Cornell AWS accounts.

Will I save money by using the Shared VPC offering?

Probably. If you switch to a Shared VPC offering, you can probably retire NAT Gateways that you currently run to support private subnets in your own VPC (s). Similarly, you will save the costs of VPC Flow Logs if you can stop using your own VPC(s).

Limitations and Quotes

Are there quotas or limits associated with the Shared VPC?

No. While we don't want customers to make deployments to the Shared VPC that will gobble up IP addresses, as of 08 Dec 2023 we don't have specific quotas about how many addresses each customer can use.

We centrally monitor IP address utilization in both **Multitenant Subnets** and **Exclusive Use Subnets** in the Shared VPC offering. **We** will reach out to customers if their usage seems excessive or unprecedented.

What happens if my needs outgrow the Shared VPC offering?

If you are using one or both of the Shared VPC options and that no longer meets your needs contact Cloud Support to request consultation about deploying a Cornell Standard VPC, which provides more flexibility than using the Shared VPC. There may also be alternatives where you could continue to use the Shared VPC offerings and meet networking/VPC needs in other ways.

Here are some indications that you are outgrowing the Shared VPC offerings:

- You have a new need to use lots of IP addresses. E.g., you might be migrating to use Kubernetes.
- You have a new need for custom Network ACL rules. The Shared VPC offerings use Baseline AWS Network ACL. If your needs are at odds with that baseline, you may need your own VPC so that your can control and customize the NACL.
- You need to peer to a vendor VPC or a custom VPC that you manage. The Shared VPC offerings don't offer peering with arbitrary VPCs.

What types of resources or AWS services can I deploy to the Shared VPC?

Any resource type or AWS service can be used in the Shared VPC, as long as it does not consume large numbers of IP addresses. For example, Amazon Elastic Kubernetes Service (EKS) should not be used in the Shared VPC since it uses large number of IP addresses, even for modest deployments.

We are not aware of any AWS services that require VPCs and that won't work with the Shared VPC offerings, as long as the service supports deployment to private subnets.

Miscellaneous

Can I use Elastic IPs in the Shared VPC?

There is nothing stopping you from assigning Elastic IPs to your resources in the Shared VPC. However, since the initial release of the Shared VPC offerings supports only private subnets, assigning an EIP to a resource deployed to the Shared VPC won't allow that resource to offer services to the public internet. I.e., there is no point to assigning an EIP to resources in the Shared VPC.

References

- Cornell
  - Shared AWS VPC for Cornell AWS Accounts
  - Using the AWS Shared VPC Offerings
  - Baseline AWS Network ACL
  - The Cornell Standard VPC
  - Cornell VPN
- AWS
  - AWS Resource Access Manager
  - AWS Systems Manager Parameter Store
  - AWS Systems Manager Session Manager
  - EC2 Instance Connect Endpoint