



VPC Sharing (Subnets)

In AWS, VPC (Virtual Private Cloud) sharing is a feature that allows you to share your VPC resources with other AWS accounts within the same AWS organization. This feature simplifies collaboration between different teams or departments within an organization by allowing them to use shared resources while maintaining the isolation of other resources.

With VPC sharing, you can share the following VPC resources:

1. Subnets: You can share one or more subnets within your VPC with other AWS accounts.
2. Route tables: Shared subnets can use route tables from the owner account or from a shared account.
3. Security groups: You can share security groups with other AWS accounts, allowing them to apply the same security policies to their resources.
4. Network access control lists (ACLs): Similar to security groups, you can share network ACLs with other accounts for controlling inbound and outbound traffic.

VPC sharing simplifies network management and resource sharing within complex AWS environments while maintaining security and isolation between different accounts. It's particularly useful in scenarios where multiple teams or departments need to collaborate on AWS resources while having separate AWS accounts.



Resource Access Manager

AWS Resource Access Manager (RAM) is a service provided by Amazon Web Services (AWS) that allows you to securely share your AWS resources with other AWS accounts. This can be particularly useful in scenarios where different teams, departments, or even different organizations need to collaborate and use the same resources without duplicating them.

Key Features of AWS RAM:

1. **Resource Sharing:** AWS RAM lets you share AWS resources across AWS accounts. Resources can be shared with specific AWS accounts, organizational units, or even entire AWS Organizations.
2. **Supported Resources:** You can share various types of resources, including but not limited to Amazon VPC subnets, AWS Transit Gateways, AWS License Manager configurations, and Amazon Route 53 Resolver rules.
3. **Granular Permissions:** You can control which specific resources are shared and with whom, providing fine-grained access control to your AWS resources.
4. **Consistency and Cost Efficiency:** By sharing resources, you can maintain consistency across environments and avoid duplicating resources, potentially reducing costs.
5. **Integration with AWS Organizations:** AWS RAM integrates with AWS Organizations, making it easier to manage and share resources across multiple accounts within an organization.

Use Cases

- **Multi-Account Architectures:** Organizations often use multiple AWS accounts for different departments or teams. AWS RAM allows these accounts to share resources like VPCs, eliminating the need to replicate resources in each account.
- **Centralized Management:** Centralized teams can manage resources like licensing, directory services, or network infrastructure and share these resources with other teams, ensuring compliance and uniformity.
- **Cost Optimization:** By sharing resources rather than duplicating them, organizations can optimize costs, reducing the need for additional resources in separate accounts.

How It Works

1. **Create a Resource Share:** Define a resource share by selecting the resources you want to share and the accounts with which you want to share them.
2. **Add Principals:** Specify the principals (AWS accounts, organizational units, or entire organizations) that will have access to the shared resources.
3. **Accept Invitations:** The recipient accounts must accept the invitation to access the shared resources. Once accepted, they can use the resources according to the permissions granted.

AWS RAM simplifies resource management and sharing in multi-account environments, helping organizations streamline operations, maintain security, and optimize costs.

What are we doing in this Lab?

In this exercise, you're setting up and sharing a Virtual Private Cloud (VPC) between two AWS accounts using AWS Resource Access Manager (RAM). The main goal is to create a VPC in one account (the "main" account) and share its subnet with another account (the "member" account) within the same AWS Organization. By doing this, you can demonstrate how to use shared resources in AWS for cross-account collaboration.

Key Steps:

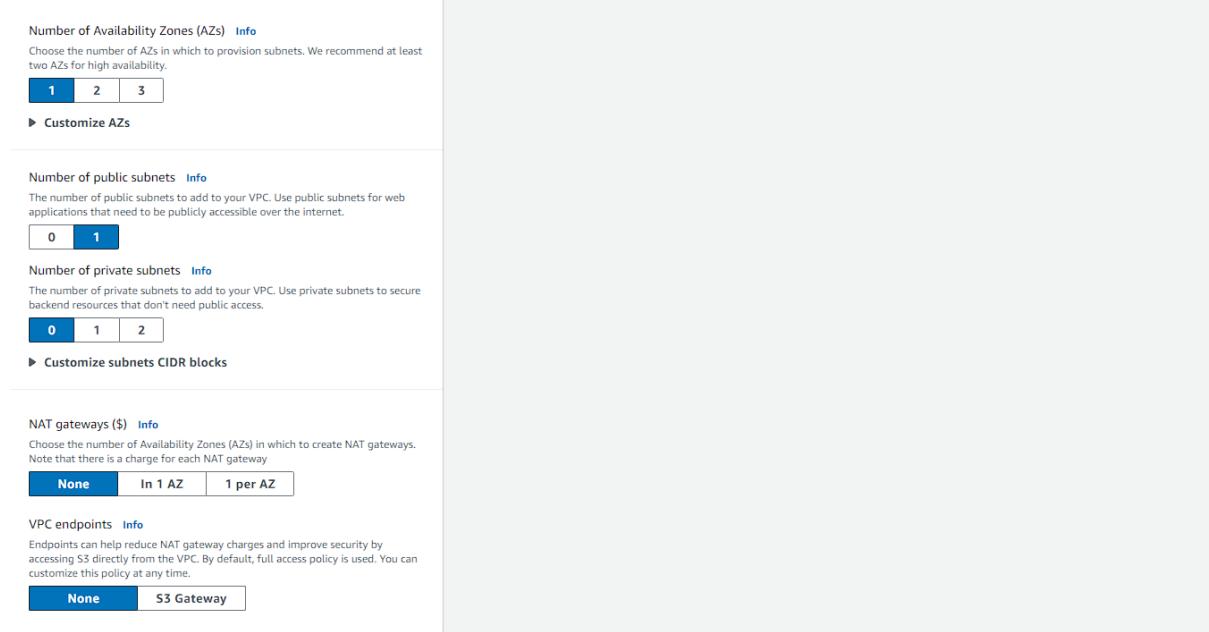
1. **Create and Share a VPC Subnet:** You create a VPC and its subnet in the main account, then share the subnet with the member account via AWS RAM.
2. **Launch EC2 Instances:** You launch an EC2 instance in each account using the shared VPC.
3. **Test Connectivity:** You establish network connectivity between the instances by configuring security group rules, allowing them to communicate.

End Goal:

The purpose is to understand how to share AWS resources securely across multiple accounts and how to set up network configurations to allow communication between resources in different accounts. This setup is useful for scenarios where different teams or departments need to access shared infrastructure while maintaining separate accounts.

To begin with the Lab:

1. There are some prerequisites for this lab. You should have two AWS Accounts.
2. Login to both of them in the AWS Console.
3. They should be connected via AWS Organizations.
4. Now you have to navigate to VPC in your main account which ever it may be.
5. Then you are going to create a new VPC.
6. While creating the VPC use the VPC and more option. Then choose the options shown below and create your VPC.



The screenshot shows the configuration options for creating a new VPC:

- Number of Availability Zones (AZs)**: Info. Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability. Options: 1 (selected), 2, 3. Action: Customize AZs.
- Number of public subnets**: Info. The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet. Options: 0 (selected), 1, 2. Action: Customize subnets CIDR blocks.
- Number of private subnets**: Info. The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access. Options: 0 (selected), 1, 2. Action: Customize subnets CIDR blocks.
- NAT gateways (\$)**: Info. Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway. Options: None (selected), In 1 AZ, 1 per AZ.
- VPC endpoints**: Info. Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time. Options: None (selected), S3 Gateway.

7. You will see that your VPC is created in no time.

Create VPC workflow

Success

Details

- Create VPC: [vpc-0aed2931859836b4f](#)
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: [vpc-0aed2931859836b4f](#)
- Create subnet: [subnet-0772f0d6ed7bb92bc](#)
- Create internet gateway: [igw-0231d2a0ff08c50da](#)
- Attach internet gateway to the VPC
- Create route table: [rtb-05b3fd65789d43e2f](#)
- Create route
- Associate route table
- Verifying route table creation

[View VPC](#)

8. Now open Resource Access Manager in both of the accounts.

Resource Access Manager

Shared by me

Resource shares

Shared resources

Principals

Shared with me

Resource shares

Shared resources

Principals

Managed permissions library

Settings

AWS Resource Access Manager

Share AWS resources with other AWS accounts.

Start sharing your AWS resources with other accounts

Create a resource share

Pricing

AWS RAM is offered at no additional charge. There are no setup fees or upfront commitments.

More resources

What is AWS Resource Access Manager?

Getting started

Documentation

Your AZ ID

AZ IDs provides a consistent way of identifying the location of a resource across all your accounts. This makes it easier for you

How it works

Use cases

9. Now in here click on settings and enable the sharing with AWS Organizations.

Resource Access Manager

Shared by me

Resource shares

Shared resources

Principals

Shared with me

Resource shares

Shared resources

Principals

Managed permissions library

Settings

Enable sharing with AWS Organizations

If you enable sharing with the accounts of your organization, you can share resources without using invitations. You can enable sharing in the organization's management account. The organization must support all features.

Save settings

10. Once this is done go to Resource share and click on create resource share.

Resource Access Manager > Shared by me: Resource shares

Shared by me: Resource shares

Resource shares owned by your account.

Resource shares (0)				
<input type="button" value="C"/>	<input type="button" value="Modify"/>	<input type="button" value="Delete"/>	<input type="button" value="Create resource share"/>	
<input type="text"/> Filter by text and property value				

No resource shares found.
Start sharing resources by creating a resource share.

Create resource share

11. In there you have to give it a name then in the resources search for subnets and select your newly created subnet. Then move to next page.

Resource share name

Name
Provide a descriptive name for the resource share.
subnet-sharing

Resources - optional
Choose the resources to add to the resource share

<input checked="" type="checkbox"/>	ID	Name	VPC ID	Availability zone	Availability zone ID	IPv4 CIDR
<input checked="" type="checkbox"/>	subnet-0772f0d6ed7bb92bc	demo-subnet-public1-ap-south-1a	vpc-0aed2931859836b4f	ap-south-1a	aps1-az1	10.0.0.0/2

12. One the page or say step 2 leave as it is and move to next.

13. Now here you need to click on display organizational structure. Then select your member account on which you are going to share your subnet.

Principals - optional

<input type="radio"/> Allow sharing with anyone You can share resources with any AWS accounts, roles, and users. If you are in an organization, you can also share with the entire organization or organizational units in that organization.	<input checked="" type="radio"/> Allow sharing only within your organization You can share resources with the entire organization, organizational units, or AWS accounts, roles, and users in that organization.
--	---

Principals

You can add multiple principals of different types. To display and select principals from a hierarchical view of your organization, turn on [Display organizational structure](#).

Display organizational structure

Name	ID	Contact
<input type="checkbox"/> Organization	o-5d8jobeveu	-
<input checked="" type="checkbox"/> DemoAccount	533267094905	[REDACTED]

14. Then just move to review page and create your subnet.

15. Below you can see that the status is associating. So, now you have to wait until it gets to associated.

Shared resources (1)	Disassociate		
<input type="text"/> Filter by text			
<input type="checkbox"/> Resource ID	Resource type	Status	
<input type="checkbox"/> subnet-0772f0d6ed7bb92bc	-		
Shared resources (1)			
<input type="text"/> Filter by text			
Managed permission name	Version	Resource type	Status
arn:aws:ram:aws:permission/AWSRAMDefaultPermissionSubnet	1 (default)	ec2:Subnet	
Shared principals (1)			Disassociate
<input type="text"/> Filter by text			
<input type="checkbox"/> Principal ID	Principal type	Status	
<input type="checkbox"/> 533267094905	Account		

16. After sometime just refresh the page and you will see that the status is green now. Always check that the status should be green or say Associated.

Shared resources (1)	Disassociate		
<input type="text"/> Filter by text			
<input type="checkbox"/> Resource ID	Resource type	Status	
<input type="checkbox"/> subnet-0772f0d6ed7bb92bc	ec2:Subnet		
Shared resources (1)			
<input type="text"/> Filter by text			
Managed permission name	Version	Resource type	Status
arn:aws:ram:aws:permission/AWSRAMDefaultPermissionSubnet	1 (default)	ec2:Subnet	
Shared principals (1)			Disassociate
<input type="text"/> Filter by text			
<input type="checkbox"/> Principal ID	Principal type	Status	
<input type="checkbox"/> 533267094905	Account		

17. Now move to the other account and navigate to Resource Access Manager. You will see those shared resources in this account.

Resource Access Manager

Shared with me: Resource shares

Resource shares my account has access to.

Resource shares (1)

Name	ID	Owner	Status
subnet-sharing	536f7ca1-268c-4947-b846-feb13a66b707	878893308172	Active

Managed permissions library
Settings

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

18. Now for a quick review go to VPC in this account and check for the VPC. Now in the VPC you will see that there are two VPCs.

VPCs

Asia Pacific 2

See all regions ▾

19. There are two ways two ways to recognize your shared VPC. One is to see the CIDR and other one is to select it and click on action you won't be able to perform any action on this VPC.

Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
-	vpc-0aed2931859836b4f	Available	10.0.0.0/16	-	dopt-C
-	vpc-0de8e4739f775bef	Available	172.31.0.0/16	-	dopt-C

Your VPCs (1/2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> -	vpc-0aed2931859836b4f	Available	10.0.0.0/16	-
<input type="checkbox"/> -	vpc-0de8e4739f775bef	Available	172.31.0.0/16	-

Actions ▲ Create VPC

- Create default VPC
- Create flow log
- Edit VPC settings
- Edit CIDs
- Manage middlebox routes
- Manage tags
- Delete VPC

20. Now in your member account or say account 2 navigate to EC2 and launch an instance using the shared VPC.

Instances (1/1) **Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 address
<input checked="" type="checkbox"/> account-2-instance	i-0106683885a610c26	Running	t2.micro	Initializing	View alarms +	ap-south-1a	10.0.7.240

Instance: i-0106683885a610c26 (account-2-instance)

Details | Status and alarms **New** | Monitoring | Security | Networking | Storage | Tags

Instance summary **Info**

Instance ID	i-0106683885a610c26 (account-2-instance)	Public IPv4 address	Private IPv4 addresses
IPv6 address	-	Running	10.0.7.240
			Public IPv4 DNS
			-

- Now you need to do the same thing in your main account, create an EC2 instance with the same VPC and remember to enable public IP address.

Instances (1/1) **Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
<input checked="" type="checkbox"/> account-1-ec2	i-02ced0cc15dacd6db	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-233-1

Instance: i-02ced0cc15dacd6db (account-1-ec2)

Details | Status and alarms **New** | Monitoring | Security | Networking | Storage | Tags

Instance summary **Info**

Instance ID	i-02ced0cc15dacd6db (account-1-ec2)	Public IPv4 address	Private IPv4 addresses
IPv6 address	-	13.233.10.61 [open address]	10.0.5.22
		Running	Public IPv4 DNS
			ec2-13-233-10-61.ap-south-1.compute.amazonaws.com [open address]

- Now SSH into your main account instance and try to ping the other account instance.
- Currently, you will see that is not able to connect to the other instance. For that you are going to add rule in the security group of the account 2.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-5-22 ~]$ ping 10.0.7.240
PING 10.0.7.240 (10.0.7.240) 56(84) bytes of data.

```

- Now from EC2 quickly open security click on edit inbound rules. And then add all traffic from VPC CIDR block. Then save the rules.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range	Source Info	Description - optional Info
sgr-00b12214a5734262d	SSH	TCP	22	Custom Info	<input type="text" value="Q"/> 0.0.0.0/0 X
-	All traffic	All	All	Custom Info	<input type="text" value="Q"/> 10.0.0.0/16 X

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

25. Now if you will go back, you'll be able to see the connection working.

```
#  
~\_ ##### Amazon Linux 2023  
~~ \#####\  
~~ \|##|  
~~ \|/  
~~ V~'`->  
~~ /  
~~-/-/  
~/m/  
[ec2-user@ip-10-0-5-22 ~]$ ping 10.0.7.240  
PING 10.0.7.240 (10.0.7.240) 56(84) bytes of data.  
64 bytes from 10.0.7.240: icmp_seq=137 ttl=127 time=0.576 ms  
64 bytes from 10.0.7.240: icmp_seq=138 ttl=127 time=0.483 ms  
64 bytes from 10.0.7.240: icmp_seq=139 ttl=127 time=0.455 ms  
64 bytes from 10.0.7.240: icmp_seq=140 ttl=127 time=0.933 ms  
64 bytes from 10.0.7.240: icmp_seq=141 ttl=127 time=0.535 ms  
64 bytes from 10.0.7.240: icmp_seq=142 ttl=127 time=0.489 ms  
64 bytes from 10.0.7.240: icmp_seq=143 ttl=127 time=0.483 ms  
64 bytes from 10.0.7.240: icmp_seq=144 ttl=127 time=0.465 ms  
64 bytes from 10.0.7.240: icmp_seq=145 ttl=127 time=0.414 ms  
64 bytes from 10.0.7.240: icmp_seq=146 ttl=127 time=0.529 ms  
64 bytes from 10.0.7.240: icmp_seq=147 ttl=127 time=0.494 ms  
64 bytes from 10.0.7.240: icmp_seq=148 ttl=127 time=0.458 ms  
64 bytes from 10.0.7.240: icmp_seq=149 ttl=127 time=0.482 ms  
64 bytes from 10.0.7.240: icmp_seq=150 ttl=127 time=0.478 ms  
64 bytes from 10.0.7.240: icmp_seq=151 ttl=127 time=0.524 ms  
64 bytes from 10.0.7.240: icmp_seq=152 ttl=127 time=0.483 ms  
64 bytes from 10.0.7.240: icmp_seq=153 ttl=127 time=0.421 ms  
64 bytes from 10.0.7.240: icmp_seq=154 ttl=127 time=0.455 ms  
64 bytes from 10.0.7.240: icmp_seq=155 ttl=127 time=0.439 ms  
64 bytes from 10.0.7.240: icmp_seq=156 ttl=127 time=0.496 ms  
64 bytes from 10.0.7.240: icmp_seq=157 ttl=127 time=0.521 ms
```

26. Do not forget to terminate the instance in both of your accounts.