Rehosting your applications in a multi-account architecture on AWS by using VPC interface endpoints

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Rehosting your applications in a multi-account architecture on AWS by using VPC interface endpoints

Table of Contents

Rehosting your applications in a multi-account architecture on AWS by using VPC interface endpoints

Dipin Jain and Saurabh Shankar, Amazon Web Services

Many organizations rehost (lift and shift) their applications to AWS from isolated or semi-isolated network environments, including on-premises data centers and other cloud or hybrid infrastructures, by using AWS Transform MGN. These isolated networks typically do not allow any egress traffic to the public endpoints that are described in the network requirements for MGN. You can address this limitation by using virtual private cloud (VPC) interface endpoints, as discussed in the AWS Prescriptive Guidance pattern Connect to MGN data and control planes over a private network.

Many organizations also use a multi-account landing zone (MALZ) architecture for isolation, security, and per-account billing benefits. To enable lift-and-shift migration by using MGN over a secured network to multiple target AWS accounts, you must create VPC interface endpoints in every target AWS account. This adds to the cost and complexity of managing those resources.

This guide discusses options for centralizing VPC interface endpoints for rehosting your applications in a multi-account architecture on AWS. These options simplify the architecture and reduce costs for such use cases.

Targeted business outcomes

This guide provides solutions for three rehosting use cases. It focuses on reducing costs and operational overhead, and improving security and resource utilization.

Use cases:

- Your applications are mapped to different business units, and you want to migrate them to different AWS target accounts in the same AWS Region to simplify billing and isolation.

  The solution to this scenario involves creating VPC endpoints in a central AWS networking account and using AWS Transit Gateway to connect to target application accounts.
- You want to migrate your applications to different AWS target accounts in multiple AWS Regions to keep your applications close to your users or to facilitate disaster recovery. This is an extension of the first use case.

The solution to this scenario involves creating VPC endpoints for each AWS Region in an AWS central networking account, and enabling cross-account access by using a peered transit gateway and Amazon Route 53.

- Your applications are mapped to different business units, and you want to migrate them to different AWS target accounts in the same AWS Region for billing and isolation purposes. You also want to use fewer VPCs for MGN staging, and you want to centrally manage routing for the staging VPCs to reduce costs and administrative overhead.

  The solution to this scenario involves sharing VPC endpoints by using a shared staging area subnet, with AWS Organizations and AWS Resource Access Manager (AWS RAM).

Intended audience

This guide is for migration consultants, application architects, infrastructure architects, and application owners who are looking for solutions for these use cases.

Solution 1: Creating VPC endpoints in a central networking account for a single Region

Use case

Your applications are mapped to different business units and you want to migrate them to different AWS target accounts in the same AWS Region for billing and isolation purposes.

Challenges

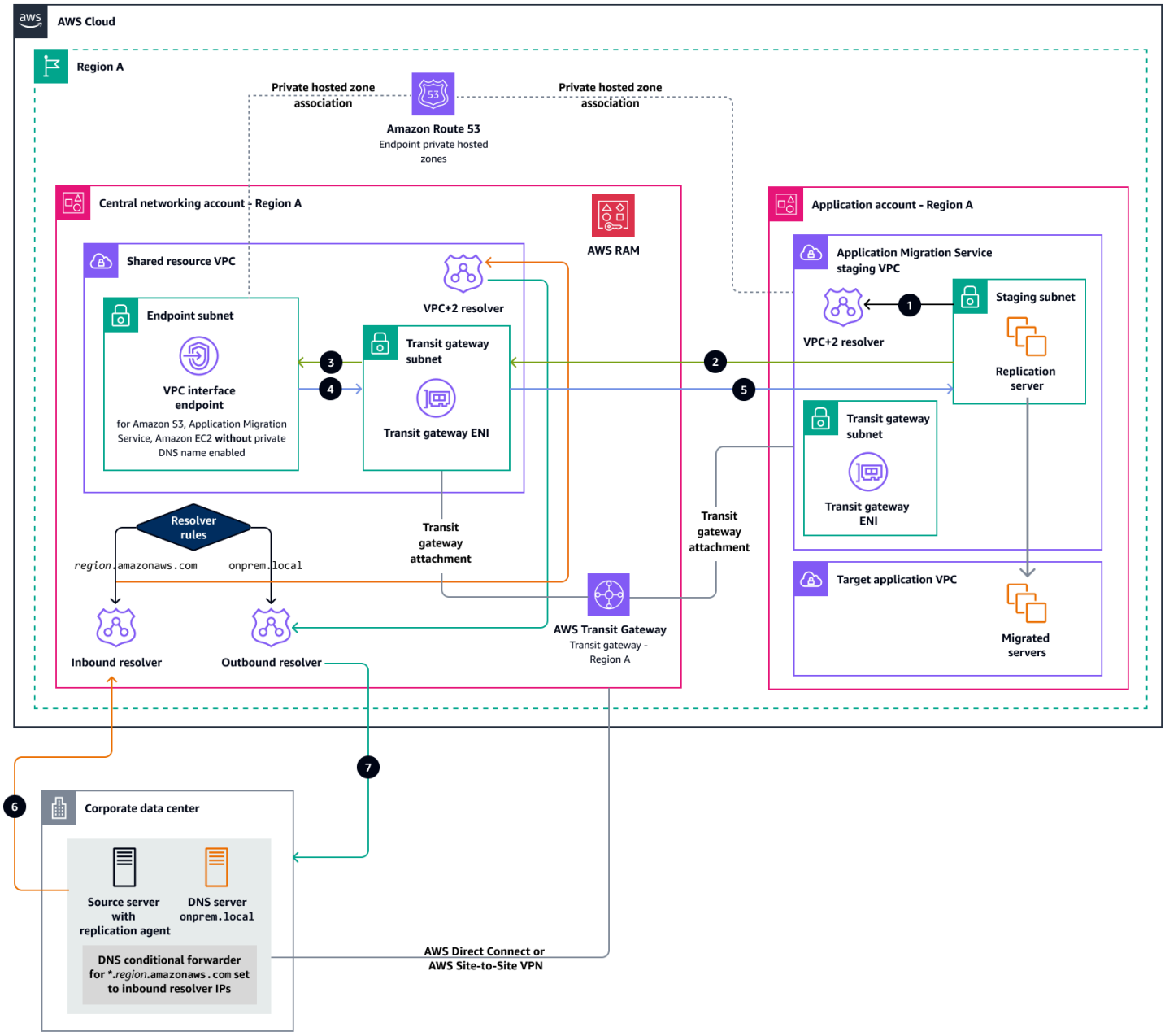
To achieve this over a private network, you would have to create multiple VPC interface endpoints in every target account. This adds to administrative overhead and costs for maintaining endpoints. (See AWS PrivateLink pricing.)

Solution

Create VPC endpoints in a central AWS networking account and use Transit Gateway to connect to target application accounts.

Architecture

The following diagram illustrates the architecture for this solution.



In the diagram, the numbers represent the following traffic flow:

1. The Amazon Elastic Compute Cloud (Amazon EC2) instance or MGN replication server that needs to connect to Amazon Simple Storage Service (Amazon S3), MGN, or Amazon EC2 through an interface endpoint that's located in the central networking account VPC first needs to resolve the domain name by querying the VPC+2 resolver. The endpoint private hosted zone is associated with the MGN staging VPC in the same Region to complete the domain resolution.

> **Note**
>
> For each private hosted zone that you associate with a VPC, the resolver creates a rule and associates it with the VPC. If you associate the private hosted zone with multiple VPCs, the resolver associates the rule with all the VPCs.

2. When the instance knows the private IP to connect to, it sends the traffic to the transit gateway ENI. The traffic is sent to the transit gateway and forwarded to the shared resource VPC, based on the transit gateway route table.
3. The transit gateway ENI in the shared resource VPC forwards the traffic to the corresponding interface endpoint.
4. The VPC endpoint sends the response back to the transit gateway ENI.
5. The traffic is forwarded to the transit gateway. As specified in the transit gateway route table, the traffic is sent to the spoke MGN staging VPC. The response is sent by the transit gateway ENI to the destination, that is, to the EC2 instance or MGN replication server.
6. A client application that's located in the corporate data center resolves a domain name in the form `<aws_service>.<aws_region>.amazonaws.com` (for example, `mgn.us-east-1.amazonaws.com`). It sends the query to its preconfigured DNS resolver. The DNS resolver in the corporate data center has a forwarding rule that points any DNS query for `amazonaws.com` domains to the Route 53 Resolver inbound endpoint. Transit Gateway forwards the query to the shared resource VPC, which forwards the DNS query at the Route 53 Resolver inbound endpoint.

   The Route 53 Resolver inbound endpoint uses the VPC+2 resolver. The endpoint private hosted zone that's associated with the shared resource VPC holds the DNS records for `amazonaws.com`, so Route 53 Resolver can resolve the query.
7. The Route 53 Resolver outbound endpoint returns the DNS query response to the on-premises client application.

Implementation steps

To set up the architecture shown in the previous diagram, follow these steps:

1. Connect your corporate data center to the central networking account on AWS through AWS Direct Connect or AWS Site-to-Site VPN.

2. In the networking account, use Transit Gateway to provide connectivity between VPCs. The transit gateway is shared between AWS accounts by using AWS RAM, and further connected to the target application account staging subnet through a VPC attachment.

> **Note**
>
> Target AWS accounts do not have to be part of the same AWS organization. For more information, see Shareable resources in the AWS RAM documentation.

3. Create VPC interface endpoints for Amazon EC2, MGN, and Amazon S3 in the central network account **without** enabling a private DNS name.

> **Note**
>
> When you create a VPC endpoint to an AWS service, you can enable private DNS. When enabled, the setting creates a managed Route 53 private hosted zone for you. This managed zone resolves the DNS name within a VPC. However, it doesn't work outside the VPC. This is the reason for using private hosted zone sharing and Route 53 Resolver to help get unified name resolution for shared VPC endpoints.

4. Create a private hosted zone for each endpoint (MGN, Amazon EC2, Amazon S3). For example, for MGN in the `us-east-1` Region:
   - Create a private hosted zone with the domain name `mgn.us-east-1.amazonaws.com`.
   - Create a host record of type A that points the domain name to endpoint IPs.
5. Create Route 53 Resolver inbound and outbound rules to facilitate hybrid DNS resolution, and share the rules with the required AWS account in the same Region.

Solution 2: Creating VPC endpoints in a central networking account for multiple Regions

Use case

You want to migrate your applications or servers to different AWS target accounts in multiple AWS Regions, to keep them close to your users or to enable business continuity in disaster recovery scenarios. This is an extension of the first use case.

Challenge

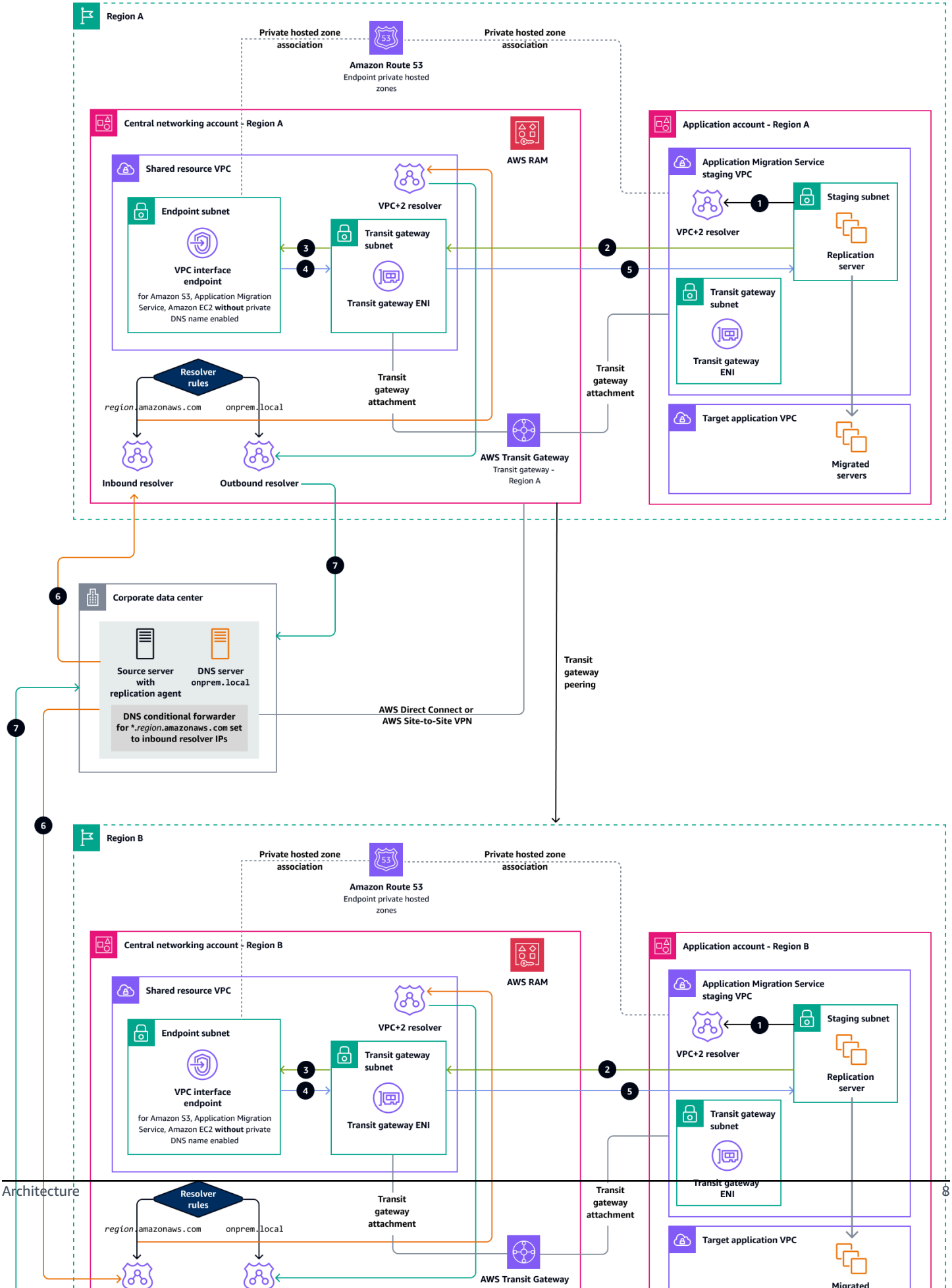
To achieve this over a private network, you would have to create multiple VPC interface endpoints in every target account. This gets even more complex in a multi-Region scenario and adds to administrative overhead and costs for maintaining multiple endpoints. (See AWS PrivateLink pricing.)

Solution

Create VPC endpoints for each Region in a central networking account and enable cross-account access by using a peered transit gateway and Route 53.

Architecture

The following diagram illustrates the architecture for this solution.


Region A
Private hosted zone association
Amazon Route 53
Endpoint private hosted zones
Private hosted zone association
Central networking account - Region A
AWS RAM
Application account - Region A
Shared resource VPC
VPC+2 resolver
Application Migration Service staging VPC
Endpoint subnet
VPC interface endpoint
for Amazon S3, Application Migration Service, Amazon EC2 without private DNS name enabled
Transit gateway subnet
Transit gateway ENI
Staging subnet
Replication server
Transit gateway attachment
Resolver rules
region.amazonaws.com
onprem.local
Inbound resolver
Outbound resolver
AWS Transit Gateway
Transit gateway - Region A
Target application VPC
Migrated servers
Corporate data center
Source server with replication agent
DNS server onprem.local
DNS conditional forwarder for *.region.amazonaws.com set to inbound resolver IPs
AWS Direct Connect or AWS Site-to-Site VPN
Transit gateway peering
Region B
Private hosted zone association
Amazon Route 53
Endpoint private hosted zones
Private hosted zone association
Central networking account - Region B
AWS RAM
Application account - Region B
Shared resource VPC
VPC+2 resolver
Application Migration Service staging VPC
Endpoint subnet
VPC interface endpoint
for Amazon S3, Application Migration Service, Amazon EC2 without private DNS name enabled
Transit gateway subnet
Transit gateway ENI
Staging subnet
Replication server
Transit gateway attachment
Resolver rules
region.amazonaws.com
onprem.local
AWS Transit Gateway
Target application VPC
Migrated

The traffic flow is the same as in solution 1, except that the accounts in the two Regions are connected by transit gateway peering.

Implementation steps

1. In the central networking account, create a VPC interface endpoint for each target AWS Region.
2. In the central networking account, create a private hosted zone for each endpoint in each Region, and associate the zone with the target application VPCs in the same Region.
3. In the central networking account, create a transit gateway for each target Region, and share the gateway with target accounts in same Region by using AWS RAM.
4. Connect transit gateways across Regions by using transit gateway peering, and update the transit gateway route tables as required.
5. In the central networking account, create resolver rules for each target Region, and share these rules with target accounts in the same Region by using AWS RAM.

Solution 3: Sharing VPC interface endpoints

Use case

Your applications are mapped to different business units, and you want to migrate them to different AWS target accounts in the same Region for billing and isolation purposes.

Challenge

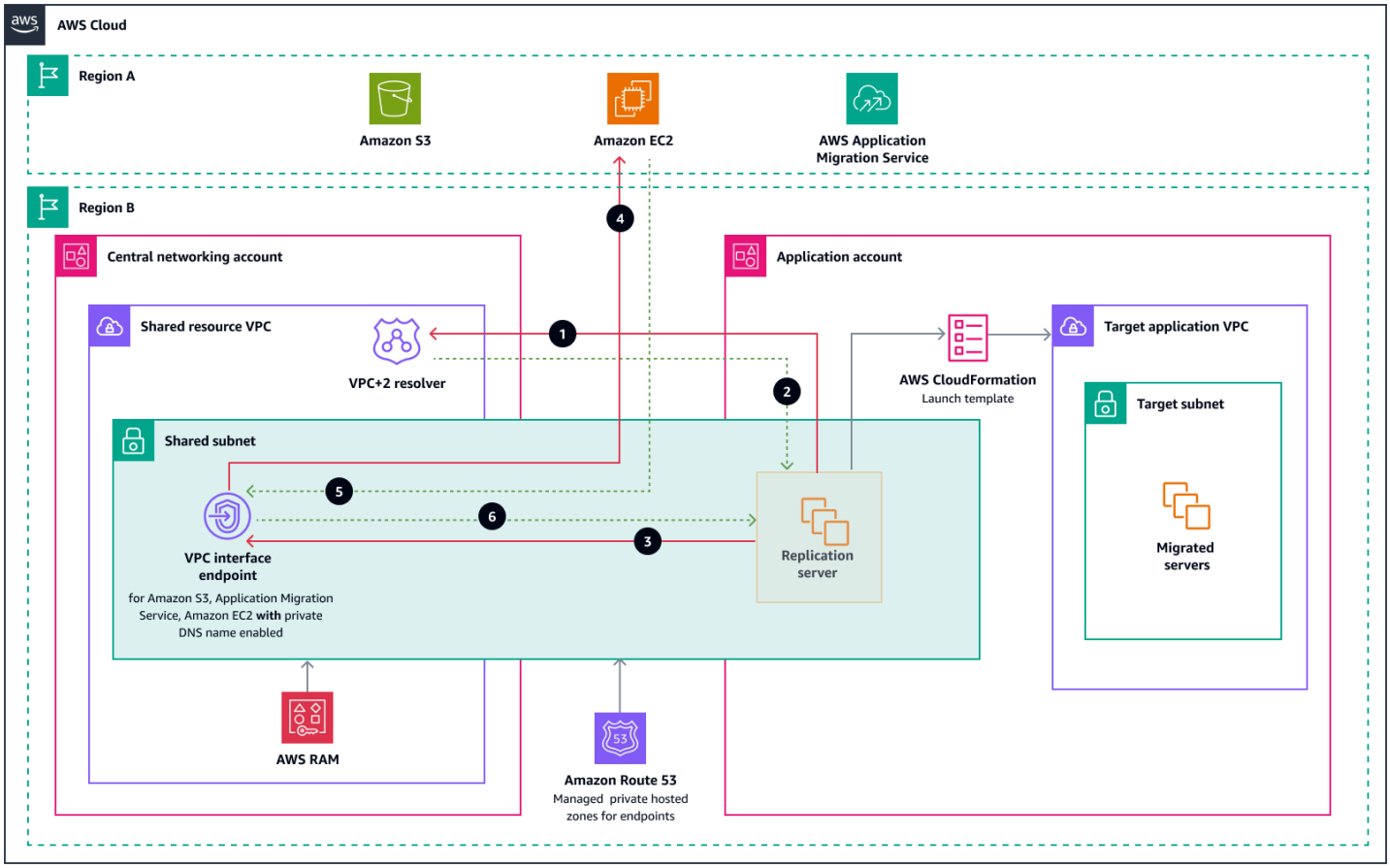
Multiple workload accounts increase both the administrative overhead and the cost of individual VPC interface endpoints in each account. Therefore, you might want to have fewer staging VPCs for MGN and centrally manage routing for the staging VPCs to reduce costs and administrative overhead.

Solution

Share VPC endpoints by using a shared staging area subnet, AWS Organizations, and AWS RAM. For more information about VPC sharing, see the Amazon VPC documentation.

Architecture

The following diagram illustrates the architecture for this solution.



The diagram illustrates the following traffic flow:

1. The MGN replication server queries the VPC+2 DNS to resolve the API endpoint for MGN, Amazon EC2, or Amazon S3.

2. The VPC+2 DNS resolves the private IP of the endpoint with the help of AWS managed private hosted zones and responds to the MGN replication server.

3-6. The replication server uses that IP to connect to the AWS service API through the interface endpoint for the service.

Implementation steps

1. In the central networking account, create the staging VPC and subnet for MGN.
2. Create endpoints for MGN, Amazon EC2, or Amazon S3 with private DNS names enabled. This creates an AWS managed private hosted zone and associates it with the staging VPC.

3. Share the staging subnet with target application accounts in the same AWS organization and AWS Region.
4. For hybrid connectivity between AWS and your on-premises data center (not shown in the previous diagram) use Transit Gateway, Direct Connect or AWS Site-to-Site VPN with Route 53 Resolver endpoints, as shown in solution 1 and solution 2.

Limitations

- The AWS accounts for the participant VPCs and the owner VPC have to be part of the same organization in AWS Organizations.
- For a list of shareable resources, see the Amazon VPC documentation.
- For VPC sharing limitations, see the Amazon VPC documentation.

Design considerations

- To reduce your operational overhead, create a resource once, and then use AWS RAM to share that resource with other accounts. This eliminates the need to provision duplicate resources in every account and reduces operational overhead.
- Simplify security management for your shared resources by using a single set of policies and permissions. If you create duplicate resources in your separate accounts, you have to implement identical policies and permissions and keep them synchronized across all accounts. Instead, you can manage all users who share an AWS RAM resource through a single set of policies and permissions. AWS RAM offers a consistent experience for sharing different types of AWS resources.
- Provide visibility and auditability. View the usage details for your shared resources by integrating AWS RAM with Amazon CloudWatch and AWS CloudTrail. For more information, see Monitoring AWS RAM using EventBridge and Logging AWS RAM API calls with AWS CloudTrail in the AWS RAM documentation.

FAQ

Q: Who can I share resources with?

A: You can share resources with any AWS account. If you are part of an organization in AWS Organizations and sharing within your organization is enabled, you can also share resources with organizational units (OUs) or with your entire organization. For supported resource types, you can also share resources with AWS Identity and Access Management (IAM) roles and IAM users. If you share resources with accounts that are outside your organization, those accounts receive an invitation to join the resource share. After they accept the invitation, they can start using the shared resources.

Q: Will I incur any charges for sharing my resources with other AWS accounts?

A: No. You can share resources at no additional cost.

Q: Can I stop sharing a resource?

A: Yes. To stop sharing a resource, remove it from the resource share or delete the resource share.

Q: How can I ensure security in AWS RAM?

A: Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud. For more information, see Security in AWS RAM in the AWS RAM documentation.

Best practices

- Avoid single points of failure. For VPC endpoints and Route 53 Resolver endpoints, use two or more Availability Zones for high availability.
- Do not use Route 53 Resolver endpoints to forward DNS queries between VPCs. We strongly recommend that you use the Amazon DNS Server (.2 resolver) as the DNS resolver for all instances inside Amazon EC2. This resolver provides the highest level of availability and scalability with the lowest latency.
- For additional best practices, see Best practices for Resolver in the Route 53 documentation.

Resources

- Access an AWS service using an interface VPC endpoint (Amazon VPC documentation)
- Share your VPC subnets with other accounts (Amazon VPC documentation)
- Shareable Amazon VPC resources (AWS RAM documentation)
- Integrating AWS Transit Gateway with AWS PrivateLink and Amazon Route 53 Resolver (AWS blog post)
- Centralizing VPC endpoints with AWS Transit Gateway (AWS Reference Architecture)
- Connect to MGN data and control planes over a private network (AWS Prescriptive Guidance)

Document history

The following table describes significant changes to this guide.

Change	Description	Date
Initial publication	—	February 18, 2025