Networking & Content Delivery

How to solve Private IP exhaustion with Private NAT Solution

by SaiJeevan Devireddy and Chandini Penmetsa | on 01 SEP 2021 | in Networking & Content Delivery | Permalink |

Share

Introduction:

As our computing needs evolve, one of the most common questions we hear from customers is, “how do I manage my private IP space? I’m almost out of it.”

It’s difficult to assign separate Private IP ranges (RFC 1918) to different business units in an organization because the available IPv4 address range is restricted. With the growing popularity of micro-service based architecture, where each task needs it’s own IP address, organizational need for IP addresses is ever increasing. Moreover, when an organization grows, they will eventually run out of Private IP ranges and are forced to use overlapping Private IP ranges across business units. It becomes even more challenging to establish connectivity between business units with overlapping CIDR ranges.

To overcome overlapping IP address limitations, customers can use solutions like AWS PrivateLink, IPv6 or use self managed NAT’ing appliances to translate IPv4 addresses and enable communication between networks with overlapping CIDR ranges. In the last approach, managing NAT rules and the IP address assignment will induce operational overhead.

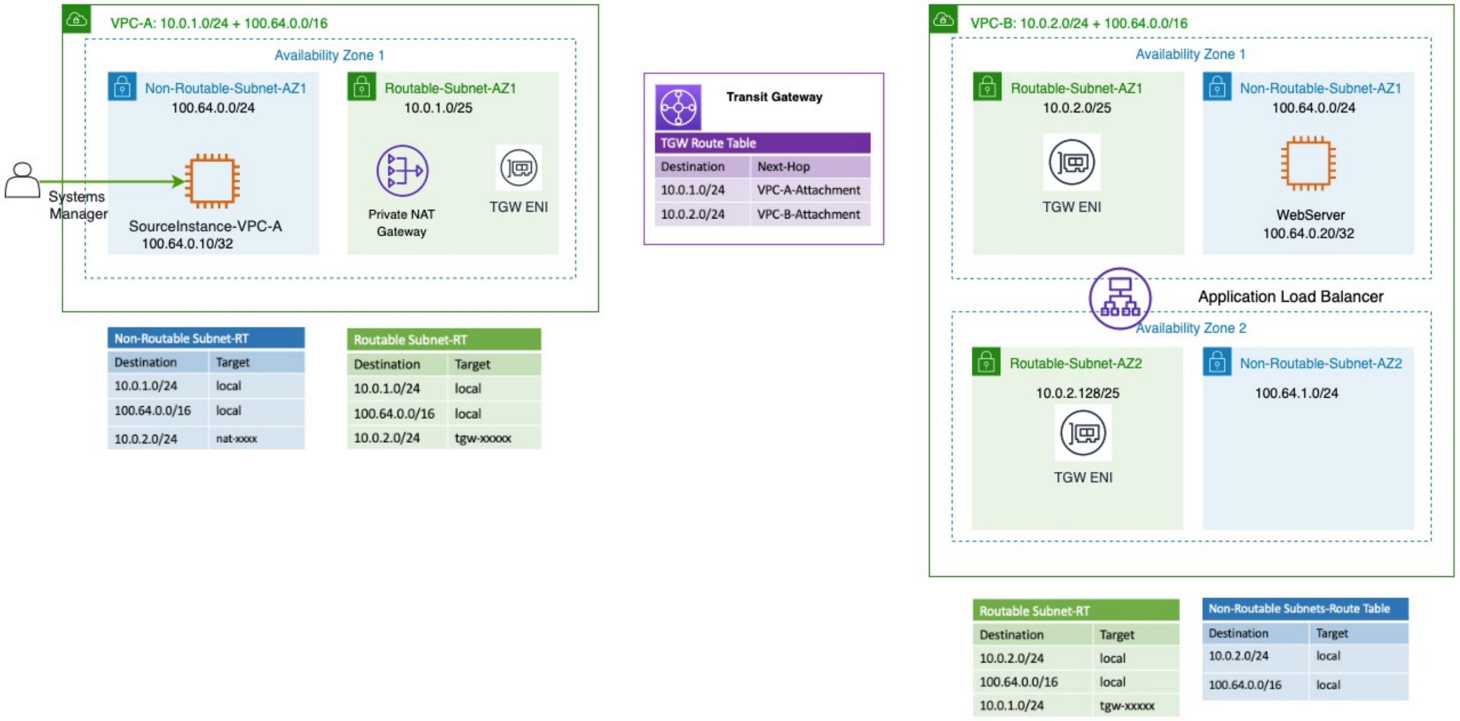
We now have a Cloud Native Solution to provide IPv4 address translation functionality between private environments, thanks to the launch of the new Private NAT Gateway. In this blog post, we’ll illustrate how to use a managed service like Private NAT Gateway to maximize private IPv4 consumption and enable communication across networks with minimal operational overhead.

Solution Overview:

This solution illustrates a mechanism to conserve and manage the RFC1918 IP address space allocation using the concept of routable & non-routable IP ranges. Furthermore, the solution describes how to connect overlapping CIDR ranges using a Private NAT Gateway that is located in the non-overlapping (or routable) IP space.

If a business unit in an organization wishes to deploy a workload that demands the use of thousands of IP addresses, the workload will be deployed on the non-routable IP address range. The non-routable IP space is used by many other business units and the overlapping nature of this space makes it non-routable. The workload will be assigned a small routable IP address range by the centralized IP Address Management (IPAM) team. The assigned routable IP range can be used by the individual business units to connect to the consolidated network. When identifying routable and non-routable IP spaces, use compatible CIDR ranges, as the CIDR blocks that can be attached as the secondary CIDR to a VPC are restricted based on the VPC’s primary CIDR block.

The following diagram (Figure 1) shows how to use AWS Transit Gateway and Private NAT to solve IP exhaustion problem and enable communication between two Amazon Virtual Private Clouds (VPCs) with overlapping CIDR ranges.



TGW Route Table

Destination	Next-Hop
10.0.1.0/24	VPC-A-Attachment
10.0.2.0/24	VPC-B-Attachment

Non-Routable Subnet-RT

Destination	Target
10.0.1.0/24	local
100.64.0.0/16	local
10.0.2.0/24	nat-xxxx

Routable Subnet-RT

Destination	Target
10.0.1.0/24	local
100.64.0.0/16	local
10.0.2.0/24	tgw-xxxxx

Routable Subnet-RT

Destination	Target
10.0.2.0/24	local
100.64.0.0/16	local
10.0.1.0/24	tgw-xxxxx

Non-Routable Subnets-Route Table

Destination	Target
10.0.2.0/24	local
100.64.0.0/16	local

Figure 1: Architecture Diagram

Scenario Walkthrough:

In this section, we will use routable and non-routable IP address ranges to establish connectivity from VPC-A to VPC-B with overlapping CIDRs using Private NAT Gateway. This walkthrough is divided into three sections as follows:

1. Identify the routable IP address space
2. Assign the IP address space
3. Configure the routing to enable connectivity

1. Identify the routable IP address space:

For non-routable address ranges, any IPv4 address range, including RFC 1918 or publicly routable IP ranges can be used as the secondary CIDR block of the VPC. Multiple teams can use the same secondary address range and thus, it should be treated as non-routable. Please keep in mind that there are some constraints when it comes to allocating IP addresses in a VPC.

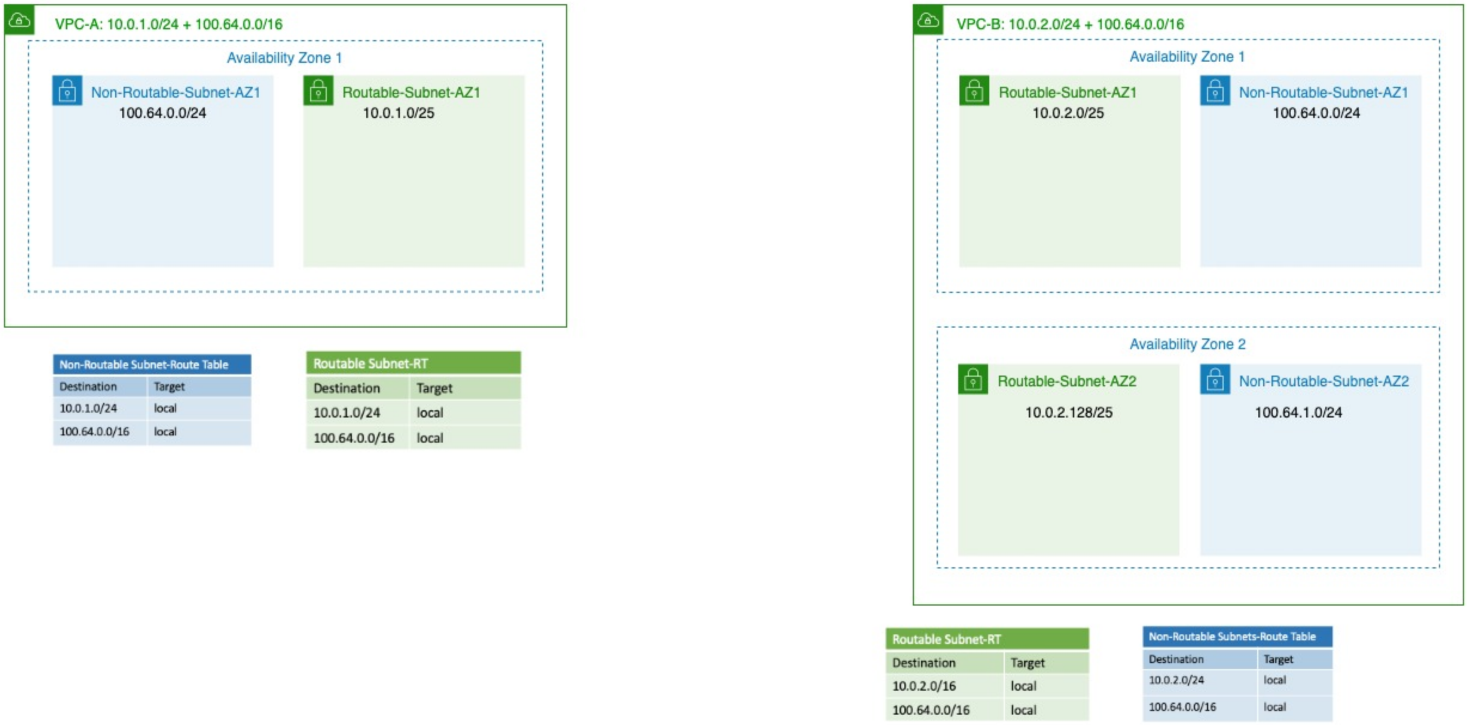
The routable space is carefully allocated by the IP management team from the central routable IP pool. Only the routable IP space is unique and advertised to the organization's consolidated network through Transit Gateway or Virtual Private Gateway.

2. Assign the IP address space:

In this blog post, the project team provisions the VPCs with primary CIDR from the routable range and uses the non-routable CIDR range as the secondary CIDR. In this walk through, non-routable and routable IP addresses are allocated as follows:

For non-routable address range we have assigned 100.64.0.0/16 IP range from the Shared Address Space(RFC 6598: i.e., 100.64.0.0/10) as the secondary CIDR to both VPC-A and VPC-B. For routable address range, we have assigned 10.0.1.0/24 to VPC-A and 10.0.2.0/24 to VPC-B as primary address ranges from RFC1918. These CIDR ranges were selected after ensuring that they are compatible with the limitations for adding a secondary CIDR block to VPC, as described in the "Solution Overview" section.

The IP address allocation for both VPC-A and VPC-B is depicted in the following diagram (Figure 2).



Non-Routable Subnet-Route Table

Destination	Target
10.0.1.0/24	local
100.64.0.0/16	local

Routable Subnet-RT

Destination	Target
10.0.1.0/24	local
100.64.0.0/16	local

Routable Subnet-RT

Destination	Target
10.0.2.0/16	local
100.64.0.0/16	local

Non-Routable Subnets-Route Table

Destination	Target
10.0.2.0/24	local
100.64.0.0/16	local

Figure 2: IP Address allocation

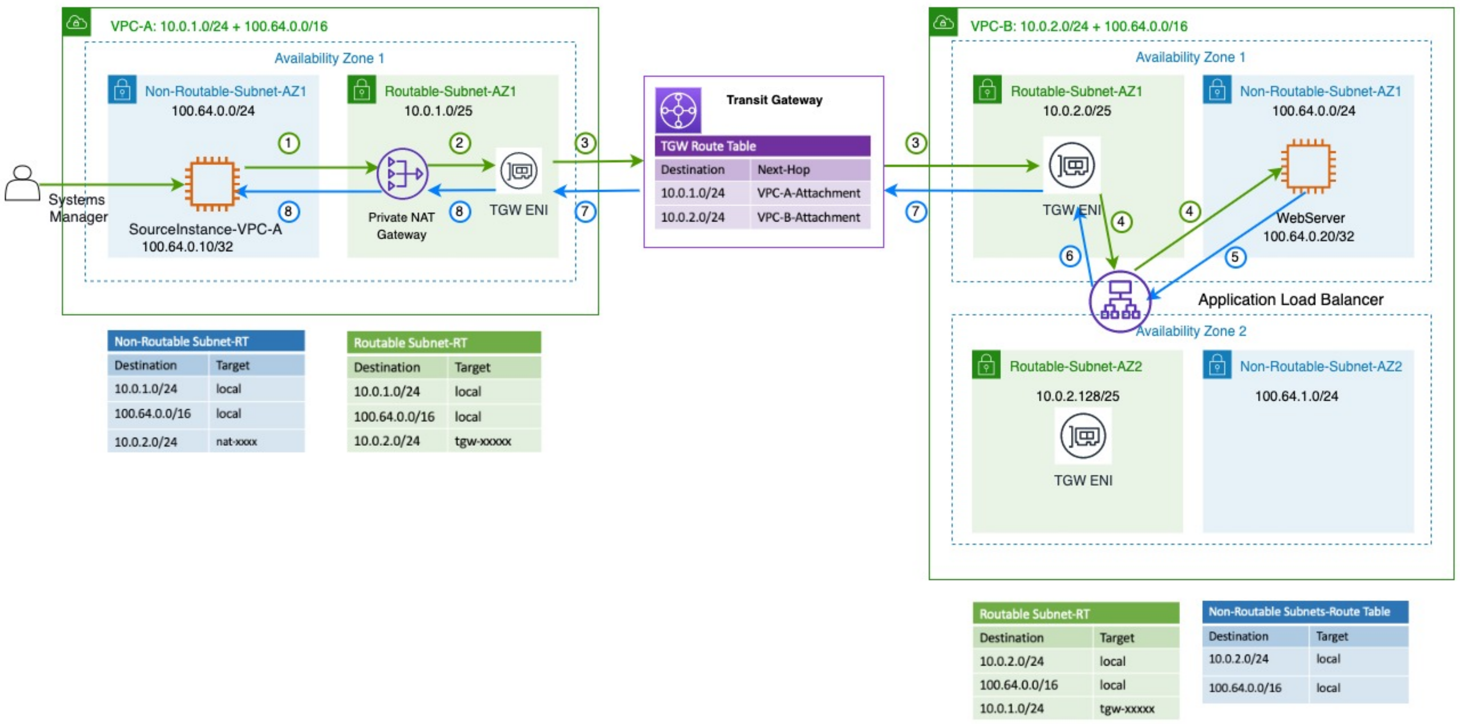
3. Configure the routing to enable connectivity:

Now that we have assigned the primary and secondary IP ranges for both VPC-A and VPC-B, let's configure the routing to enable the connectivity between VPC-A and VPC-B. We are using an internal Application Load Balancer (ALB) to expose the resources in the non-routable subnet inside VPC-B. A Private NAT Gateway is deployed in VPC-A to Source NAT the non-routable IP to a routable IP in VPC-A.

To achieve this desired state of connectivity:

1. ALB is placed in the routable subnets in VPC-B and the back-end instances in the non-routable subnets in VPC-B.
2. Private NAT Gateway is created in the routable subnets in VPC-A and the "SourceInstance- VPC-A" is created in the non-routable subnets in VPC-A to test the connectivity.
3. A Transit Gateway is created with two attachments, in routable subnets of both VPC-A and VPC-B.

Now that we understand the architecture, let's walk through a life of a packet when "SourceInstance-VPC-A" in VPC-A communicates with the ALB in VPC-B.



TGW Route Table

Destination	Next-Hop
10.0.1.0/24	VPC-A-Attachment
10.0.2.0/24	VPC-B-Attachment

Non-Routable Subnet-RT

Destination	Target
10.0.1.0/24	local
100.64.0.0/16	local
10.0.2.0/24	nat-xxxx

Routable Subnet-RT

Destination	Target
10.0.1.0/24	local
100.64.0.0/16	local
10.0.2.0/24	tgw-xxxxx

Routable Subnet-RT

Destination	Target
10.0.2.0/24	local
100.64.0.0/16	local
10.0.1.0/24	tgw-xxxxx

Non-Routable Subnets-Route Table

Destination	Target
10.0.2.0/24	local
100.64.0.0/16	local

Figure 3: Packet Flow

Packet flow from SourceInstance-VPC-A to the WebServer:

Step 1: “SourceInstance-VPC-A” in VPC-A wishes to communicate with the WebServer in VPC-B, which is located behind the ALB in VPC-B. As a result, the packet originates from “SourceInstance-VPC-A” (Source IP: 100.64.0.10/32) and is headed to the ALB. The “SourceInstance-VPC-A” will do a DNS lookup on the ALB DNS name to get the destination IP (Destination IP: 10.0.2.x/32). The “Non-Routable Subnet-RT“ in VPC-A is configured to route the traffic destined for 10.0.2.0/24 to the Private NAT Gateway.

Step 2: The Private NAT Gateway translates Source IP from non-routable IP address to a routable IP address and then sends the traffic to the Transit Gateway Attachment in VPC-A as it is associated with the “Routable Subnet-RT” route table in VPC-A.

Step 3: Transit Gateway uses the 10.0.2.0/24 route and sends the traffic to VPC-B Transit Gateway Attachment.

Step 4: TGW ENI in VPC-B uses the VPC-B local route to forward the traffic to the ALB which then forwards the traffic to the WebServer behind the ALB.

Steps to return traffic from WebServer to the SourceInstance-VPC-A:

Step 5: WebServer behind the ALB receives the request sent by the “SourceInstance-VPC-A” instance and returns a response to the ALB using the VPC-B local route.

Step 6: The ALB forwards the response traffic to the Transit Gateway Attachment in VPC-B as it is associated with the “Routable Subnet-RT” route table in VPC-B. This response traffic from the ALB has source IP address of the ALB and destination IP address of the Private NAT Gateway.

Step 7: Transit Gateway uses the 10.0.1.0/24 route and sends the traffic to VPC-A Transit Gateway Attachment.

Step 8: The TGW ENI in VPC-A uses the local route to forward the traffic to the Private NAT Gateway, which then translates the destination IP to that of the "SourceInstance-VPC-A" instance. This packet is routed to "SourceInstance-VPC-A" using the local route

Prerequisites:

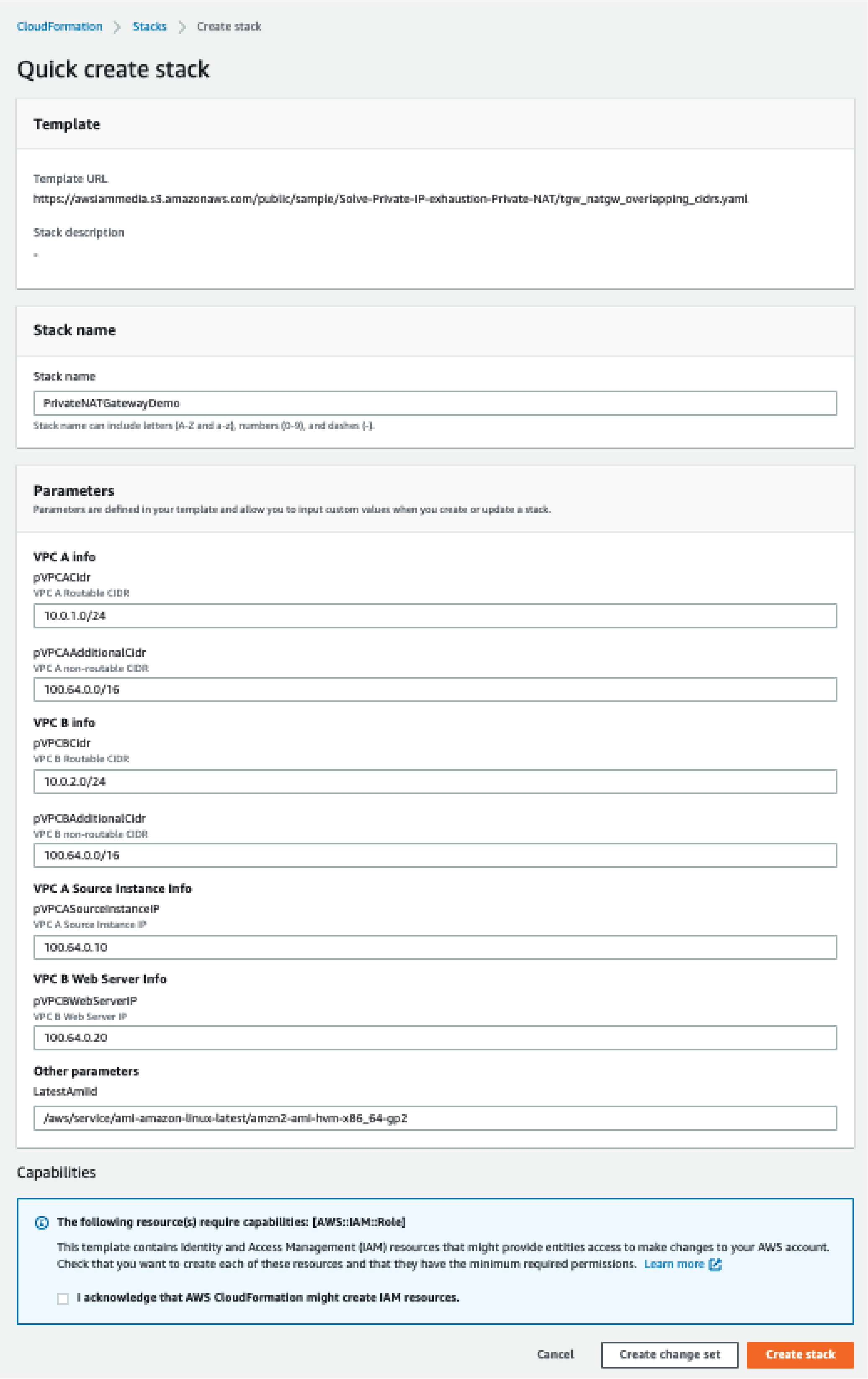
To complete this walk-through, you need:

1. An AWS account
2. An IAM user with access to AWS resources, including Systems Manager, AWS Transit Gateway, VPC and Amazon EC2.

Deployment Instructions:

In this section, we demonstrate how to deploy the architecture from figure 1, including Transit Gateway, VPCs, Subnets, Private NAT Gateway, Transit Gateway Route Tables, VPC Route Tables, Application Load Balancer, EC2 instances, etc., using the provided CloudFormation template.

1. Click on Launch Stack ▶ to deploy the CloudFormation template.
2. By default, the link takes you to the **Create stack** page within the CloudFormation console in the N. Virginia Region (us-east-1) and the solution's CloudFormation template is automatically populated. You can change the region at the top right corner of the console if needed.



Quick create stack

Template

Template URL
https://awsiammedia.s3.amazonaws.com/public/sample/Solve-Private-IP-exhaustion-Private-NAT/tgw_natgw_overlapping_cidrs.yaml

Stack description
-

Stack name

Stack name
PrivateNATGatewayDemo
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

VPC A info
pVPCACidr
VPC A Routable CIDR
10.0.1.0/24

pVPCAAdditionalCidr
VPC A non-routable CIDR
100.64.0.0/16

VPC B info
pVPCBCidr
VPC B Routable CIDR
10.0.2.0/24

pVPCBAdditionalCidr
VPC B non-routable CIDR
100.64.0.0/16

VPC A Source Instance Info
pVPCASourceInstanceIP
VPC A Source Instance IP
100.64.0.10

VPC B Web Server Info
pVPCBWebServerIP
VPC B Web Server IP
100.64.0.20

Other parameters
LatestAmiId
/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more

☐ I acknowledge that AWS CloudFormation might create IAM resources.

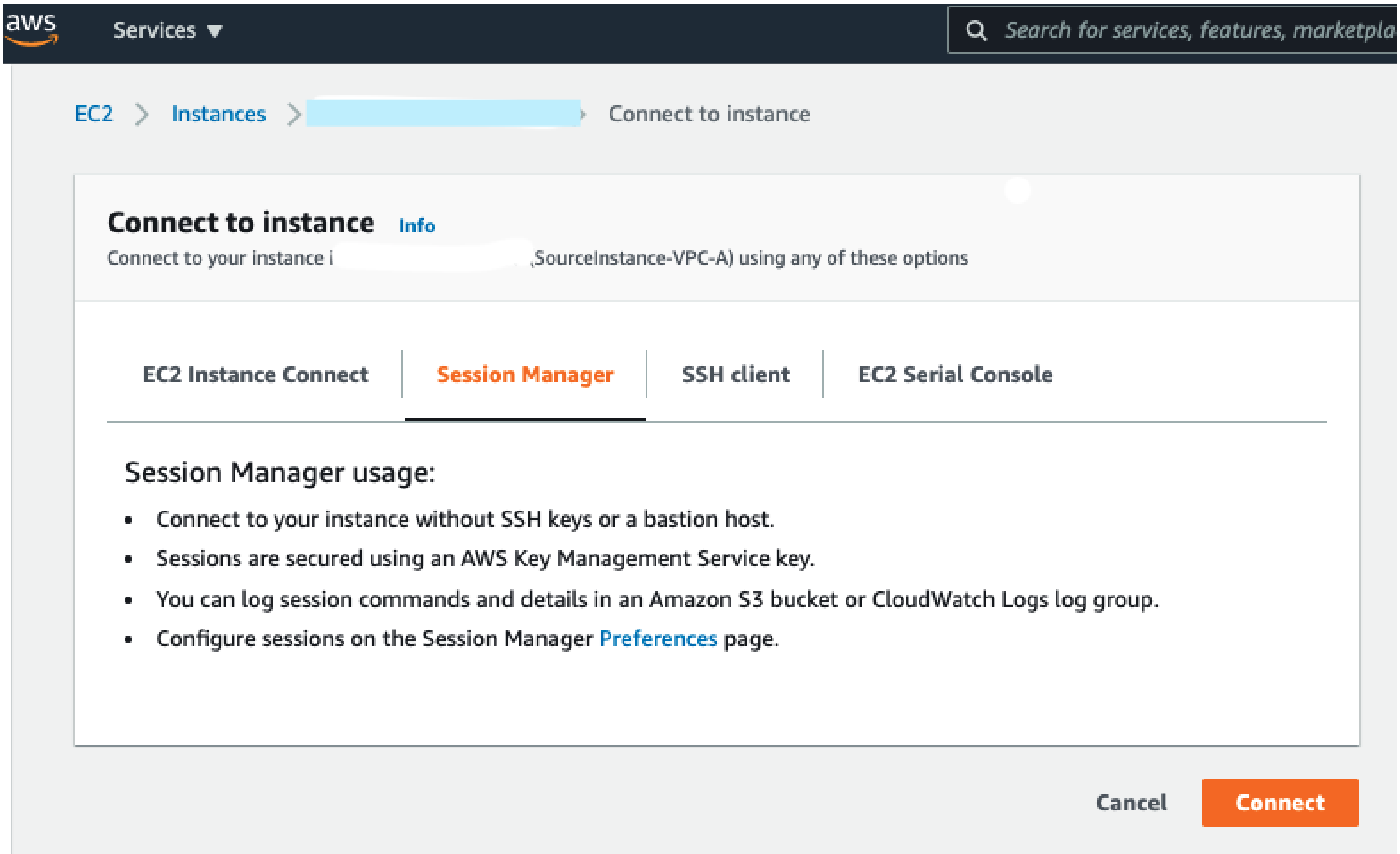
Figure 4: Create CloudFormation Stack

3. Enter a name for the stack in the **Stack name** and all the required parameters are populated with default values. In our demo environment, we have chosen to name our stack "**PrivateNATGatewayDemo**". Click "**Create Stack"** to continue.

Validation:

1. To connect to the **SourceInstance-VPC-A** instance securely using Session Manager, select the instance ID in the EC2 console and click on the **Connect** button.

2. There are four different ways to connect to the EC2 instance. Select the **Session Manager** tab and Click on **Connect** which opens a new browser-based shell session of your instance. Please keep in mind that it may take few minutes to be able to connect to the SourceInstance-VPC-A instance via the Session Manager.



Connect to your instance i[redacted],SourceInstance-VPC-A) using any of these options

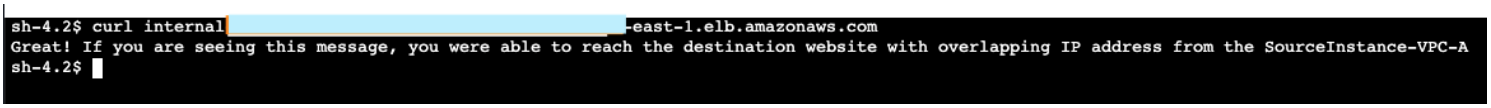
Session Manager usage:

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager Preferences page.

Figure 5: Connect to EC2

1. To verify that the "SourceInstance-VPC-A" can reach the web service which is running in VPC-B(100.64.0.0/24 non-routable subnet), use **curl** to connect the ALB DNS name(Refer **ALBHostName** value in the outputs section of the CloudFormation stack).

curl **<ALB DNS>**



```
sh-4.2$ curl internal[redacted]-east-1.elb.amazonaws.com
Great! If you are seeing this message, you were able to reach the destination website with overlapping IP address from the SourceInstance-VPC-A
sh-4.2$ █
```

Cleanup:

After testing the connectivity, please go ahead and delete the CloudFormation stack to avoid any costs associated with the resources launched by the CloudFormation template.

Conclusion:

In this post, you learned how to use routable and non-routable IP CIDR ranges along with AWS Transit Gateway and Private NAT Gateway to tackle Private IP exhaustion issue and enable communication between two Amazon VPCs with overlapping CIDR ranges. Please note that this illustration shows how to establish connectivity between two VPCs with over-lapping CIDR's. The same can be extended to a VPC and on-premise network or two on-premise networks with overlapping CIDRs.

TAGS: Amazon VPC, Application Load Balancer, AWS Transit Gateway, Networking