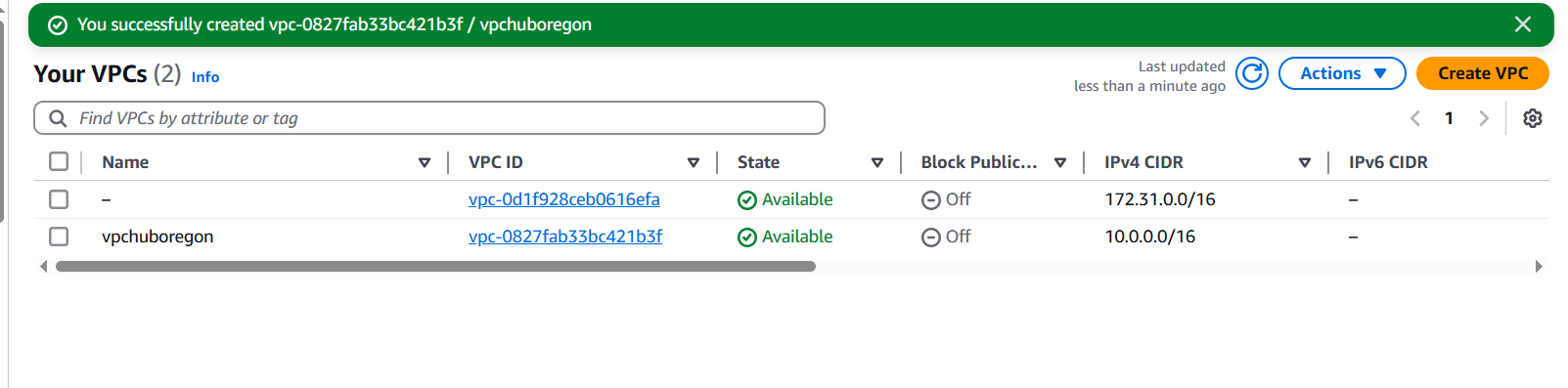
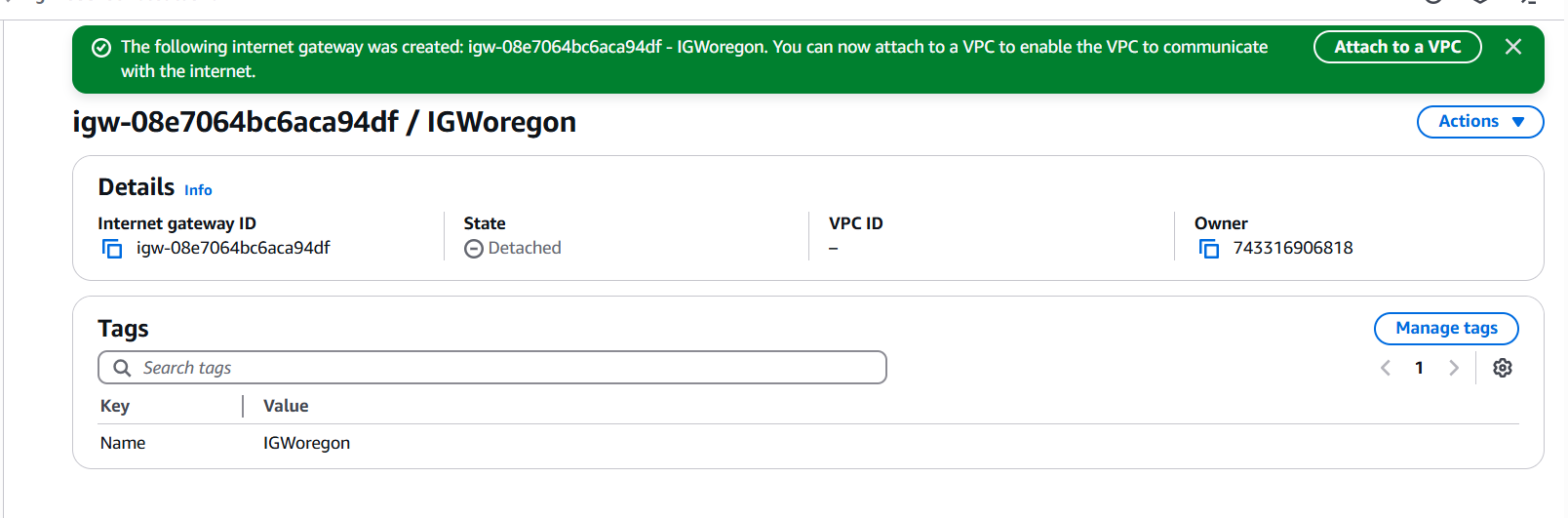
**Setting up Transit Gateway and VPC Endpoints for a Multi-VPC   
Architecture**

Scenario: A large organization is migrating its on-premises infrastructure to   
the AWS cloud. The organization's architecture involves multiple VPCs for   
different departments and applications, each requiring secure   
communication with centralized services and external resources. The IT team  
needs to design and implement a scalable and efficient network architecture   
to accommodate the organization's growth and ensure robust connectivity   
between VPCs and external services.  
Objectives: Design and deploy a scalable network architecture using AWS   
Transit Gateway to simplify network connectivity between multiple VPCs.   
Configure VPC endpoints to securely access AWS services without internet   
gateways or NAT gateways, ensuring data privacy and minimizing exposure   
to external threats.  
Design and deploy a scalable network architecture using AWS Transit   
Gateway to simplify network connectivity between multiple VPCs  
4 regions  
Oregon us-west-2  
Tokyo ap-northeast-1  
Frankfurt eu-central-1  
Paris eu-west-3  
Oregon us-west-2  
To set uo Tansit Gateway and VPC endpoints for a multi VPC Architecture  
 I have selected 4 regions in AWS  
1. Oregon : us-west-2  
2. Tokyo : ap-northeast-1

3. Frankfurt: eu-central-1

4. Paris: eu-west-3  
   
Considering oregon as hub and spoke region for remaining regions  
   
Consider Oregon region  
Create VPC  
Go to Amazon AWS console in that select VPC  
select create VPC  
Name: vpchuboregon  
CIDR: 10.0.0.0/16  
   
Create internet gateway and attach vpc created  
   
Create Subnets:  
Public subnet :10.0.1.0/24  
Private subnet:10.0.2.0/24  
   
Create Transit Gateway  
Configure transit gateway with default ASN number 64512  
Now go to Transit Gateway attachments and create 1 attachment with   
created vpc-hub-oregon



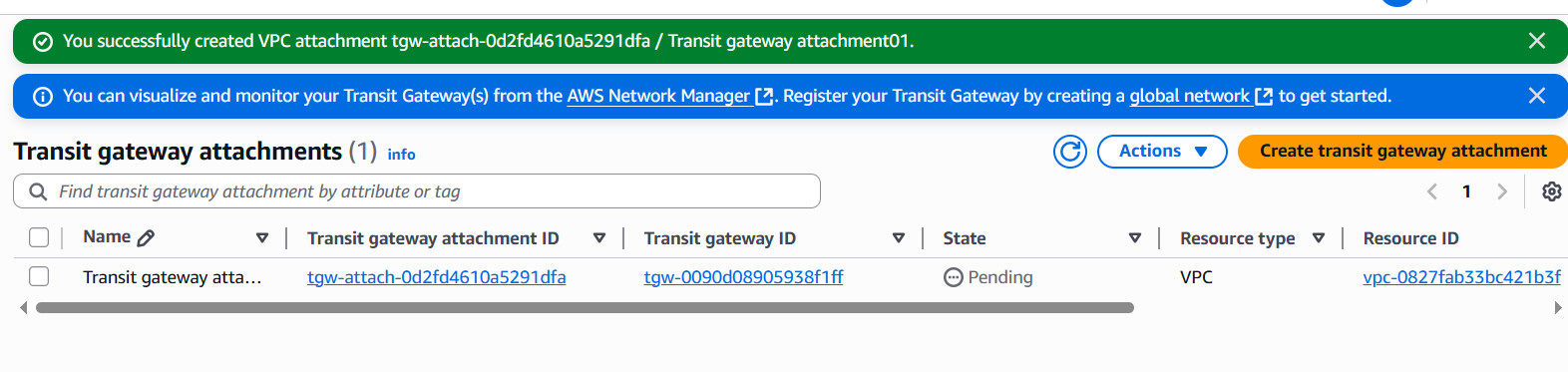


A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



**Tokyo ap-northeast-1**

Create VPC  
Go to Amazon AWS console in that select VPC  
select create VPC  
Name: vpc-tokyo  
CIDR: 11.0.0.0/16  
   
Create internet gateway and attach vpc created  
   
Create Subnets:  
Public subnet :11.0.1.0/24  
Private subnet:11.0.2.0/24  
   
Create Transit Gateway  
Configure transit gateway with default ASN number 64513  
Now go to Transit Gateway attachments and create 1 attachment with   
created vpc-tokyo

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Frankfurt eu-central-1**

Create VPC  
Go to Amazon AWS console in that select VPC  
select create VPC  
Name: vpc-frankfurt  
CIDR: 12.0.0.0/16  
   
Create internet gateway and attach vpc created  
   
Create Subnets:  
Public subnet :12.0.1.0/24  
Private subnet:12.0.2.0/24  
   
Create Transit Gateway  
Configure transit gateway with default ASN number 64514  
Now go to Transit Gateway attachments and create 1 attachment with  
created vpc-frankfurt

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.

**Paris eu-west-3**  
Create VPC  
Go to Amazon AWS console in that select VPC  
select create VPC  
Name: vpc-Paris  
CIDR: 13.0.0.0/16  
   
Create internet gateway and attach vpc created

Create Subnets:  
Public subnet :13.0.1.0/24  
Private subnet:13.0.2.0/24  
   
Create Transit Gateway  
Configure transit gateway with default ASN number 64515  
Now go to Transit Gateway attachments and create 1 attachment with  
created vpc-Paris

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

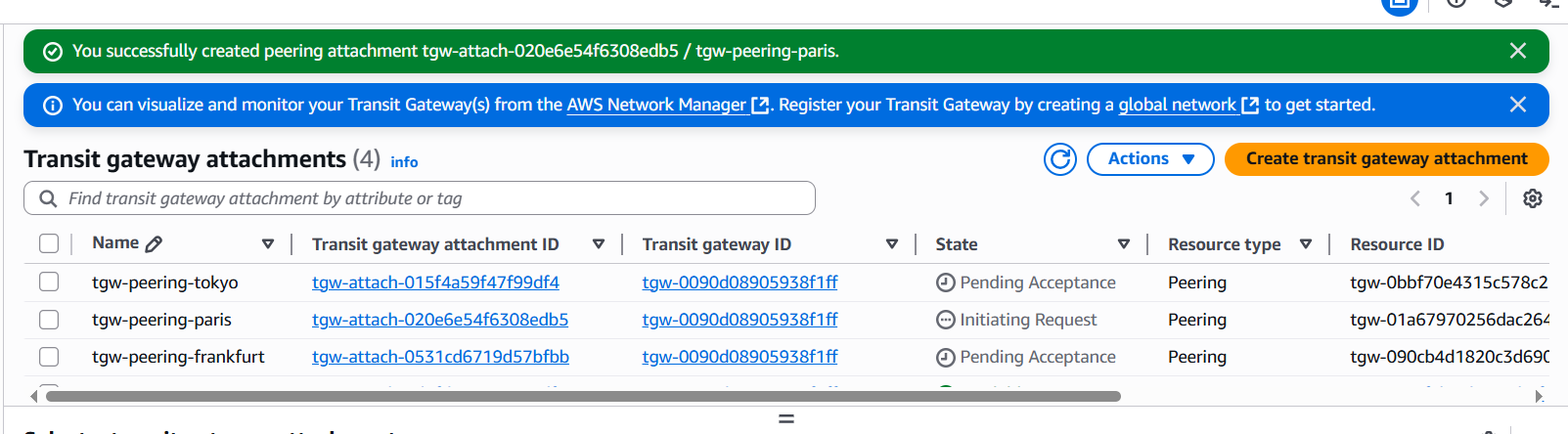
A screenshot of a computer

AI-generated content may be incorrect.

Multi-VPC-Peering-TGW  
Create Transit Gateway  
Configure transit gateway with default ASN number 64515

Now go to Transit Gateway attachments and create 1 attachment with   
created vpc-Paris  
   
Now again select Oregon region  
Go to Transit Gateway attachments  
Create transit gateway attachment there select attachment type as peering   
select appropriate id and subnet ids  
create vpc peering attachment  
same repeat for other regions create transit gateway peering attachments   
with respective configurations  
send the request to accept to peering for transit gateway and accept in   
respective regions  
   
Now come to Transit gateway Route table  
check for propagation is enabled for hub region  
Now create the static Routes with resource as peering  
11.0.0.0/16 transit gateway  
12.0.0.0/16 transit gateway  
13.0.0.0/16 transit gateway

Same repeat for other regions and create appropriate static routes



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Now its time to launch the EC2 instances in AWS console  
Paris-Ec2-instance  
Now create the private ec2- instance in aws by selecting appropriate vpc   
created and subnet as private subnet and launch the instance

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Now access CLI/Gitbash  
Connect your Oregon (hub) public instance using ssh –i oregon.pem ec2-  
user@public ip  
Once the connection is established check for remote connectivity of other   
region instances  
Ping private ip of tokyo instance  
Ping private ip of Frankfurt instance  
Ping private ip of Paris instance  
That means we have successfully Design and deploy a scalable network   
architecture using AWS Transit Gateway to simplify network connectivity   
between multiple VPCs

A screenshot of a computer screen

AI-generated content may be incorrect.

