



# Zero Trust Security for AWS Workloads with Zscaler Cloud Connector

Reference Architecture

# Contents

<b>About Zscaler Reference Architecture Guides</b>	<b>4</b>
Who Is This Guide For?	4
A Note for Federal Cloud Customers	4
Conventions Used in This Guide	4
Finding Out More	4
Terms and Acronyms Used in This Guide	5
Icons Used in This Guide	6
<b>Introduction</b>	<b>7</b>
Key Features and Benefits	10
New to Zscaler Cloud Connector?	10
<b>Cloud Infrastructure Protection Using Cloud Connector</b>	<b>11</b>
<b>Leveraging Workload Discovery Service for Policy Action</b>	<b>12</b>
Managing Overlapping IP Addresses with Namespace Tags	14
<b>Choosing a Cloud Connector Deployment Model</b>	<b>16</b>
<b>Deploying the Zero Trust Gateway Service</b>	<b>18</b>
<b>Custom Cloud Connector Deployments</b>	<b>20</b>
Pre-Deployment Considerations	20
High Availability Deployment Design	24
Leveraging Auto Scaling Groups for Redundancy	27
Cloud Connector Logging and Service Dashboards	30
Deploying Cloud Connector via Scripts	31
Upgrading Your Cloud Connectors	33
Directing Traffic to Cloud Connector	34
Forwarding Options	36
Choosing the Correct Design Model	37

Use Case: Direct to Internet Using Zscaler Internet Access	38
Use Case: Integrating with AWS Transit Gateway	40
Use Case: Distributed Gateway Load-Balancing Endpoints	42
Use Case: Integrating Zscaler Private Access	45
Use Case: Securing Traffic Between Clouds	47
<b>Summary</b>	<b>48</b>
<b>About Zscaler</b>	<b>49</b>

## About Zscaler Reference Architecture Guides

The Zscaler™ Reference Architecture series delivers best practices based on real-world deployments. The recommendations in this series were developed by Zscaler's transformation experts from across the company.

Each guide steers you through the architecture process and provides technical deep dives into specific platform functionality and integrations.

The Zscaler Reference Architecture series is designed to be modular. Each guide shows you how to configure a different aspect of the platform. You can use only the guides that you need to meet your specific policy goals.

### Who Is This Guide For?

The Overview portion of this guide is suitable for all audiences. It provides a brief refresher on the platform features and integrations being covered. A summary of the design follows, along with a consolidated summary of recommendations.

The rest of the document is written with a technical reader in mind, covering detailed information on the recommendations and the architecture process. For configuration steps, we provide links to the appropriate Zscaler Help site articles or configuration steps on integration partner sites.

### A Note for Federal Cloud Customers

This series assumes you are a Zscaler public cloud customer. If you are a Federal Cloud user, please check with your Zscaler Account team on feature availability and configuration requirements.

### Conventions Used in This Guide

The product name ZIA Service Edge is used as a reference to the following Zscaler products: ZIA Public Service Edge, ZIA Private Service Edge, and ZIA Virtual Service Edge. Any reference to ZIA Service Edge means that the features and functions being discussed are applicable to all three products. Similarly, ZPA Service Edge is used to represent ZPA Public Service Edge and ZPA Private Service Edge where the discussion applies to both products.



Notes call out important information that you need to complete your design and implementation.



Warnings indicate that a configuration could be risky. Read the warnings carefully and exercise caution before making your configuration changes.

### Finding Out More

You can find our guides on the Zscaler website at [Reference Architectures](https://www.zscaler.com/resources?type=reference-architectures) (<https://www.zscaler.com/resources?type=reference-architectures>).

















You can join our user and partner community and get answers to your questions in the [Zenith Community](https://community.zscaler.com/) (<https://community.zscaler.com/>).

## Terms and Acronyms Used in This Guide

Acronym	Definition
C2	Command & Control
DC	Data Center
DNS	Domain Name System
DoH	DNS over HTTPS
FQDN	Fully Qualified Domain Name
GWLB	Gateway Load Balancer
GWLB <sub>e</sub>	Gateway Load Balancer Endpoint
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IP	Internet Protocol
NAT	Network Address Translation
SSL	Secure Socket Layer (superseded by TLS)
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Universal Resource Locator
ZDX	Zscaler Digital Experience™
ZIA	Zscaler Internet Access™
ZPA	Zscaler Private Access™
ZTE	Zero Trust Exchange™

## Icons Used in This Guide

The following icons are used in the diagrams contained in this guide.

						
Amazon Web Services (AWS)	AWS Cloud	Amazon Elastic Compute Cloud (Amazon EC2)	Auto Scaling Group	AWS Lambda	Instance	AWS Auto Scaling
						
AWS CloudFormation	Public Subnet	Private Subnet	Region	Virtual Private Cloud (VPC)	AWS Transit Gateway	Endpoints
						
Gateway Load Balancer	Internet Gateway	NAT Gateway	Microsoft Azure	Azure Enterprise Applications	Advanced User	Zscaler App Connector
						
Bad Actor	Branch Office	Data Center	Factory	Generic Application or Workload	Headquarter Office	Invalid
						
Public Internet	Threat Actor	Zscaler Cloud Connector	Zscaler Internet Access (ZIA)	Zscaler Private Access (ZPA)	Zscaler Zero Trust Exchange (ZTE)	

## Introduction

The shift to cloud services has rebuilt the enterprise data center off-premises and outside of traditional security boundaries. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) enable organizations to quickly build out and scale their platforms and services. Securing services across multiple clouds, vendors, and support features requires a different approach than that of the traditional data center.

Securing this communication through the layering of legacy Access Control Lists (ACL), on-premises firewalls, and service-chaining has always been both complicated to build and difficult to maintain. Private applications were accessed via virtual private networks (VPNs) that extended the network to locations in an any-to-any access model. This large, flat network gave users a single location to connect to for access to private applications.

Leveraging the cloud breaks these models. You now have multiple vendors across different clouds, products, and services. Your policy must be interpreted at each cloud and application to determine how best to implement it with the tools available. This risk goes up given a mistake, potentially exposing your organization to a host of network-born attack vectors.

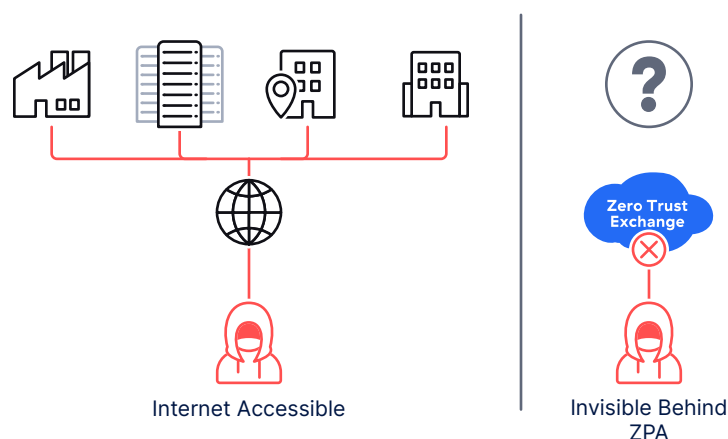


Figure 1: ZPA makes your applications invisible outside of your organization

Ideally, an organization's security policy should be at the foundation of its network design. Connectivity to and from devices happens as a product of the security policy and not the other way around. This is the heart of the Zscaler Zero Trust Exchange (ZTE) model. Users must be authorized before they can connect to that service. Even knowing the application's hostname and the services it provides won't give the attacker any information, as that service won't resolve until the user authenticates. Your applications are effectively hidden from the internet and each other until you define policy to allow access.

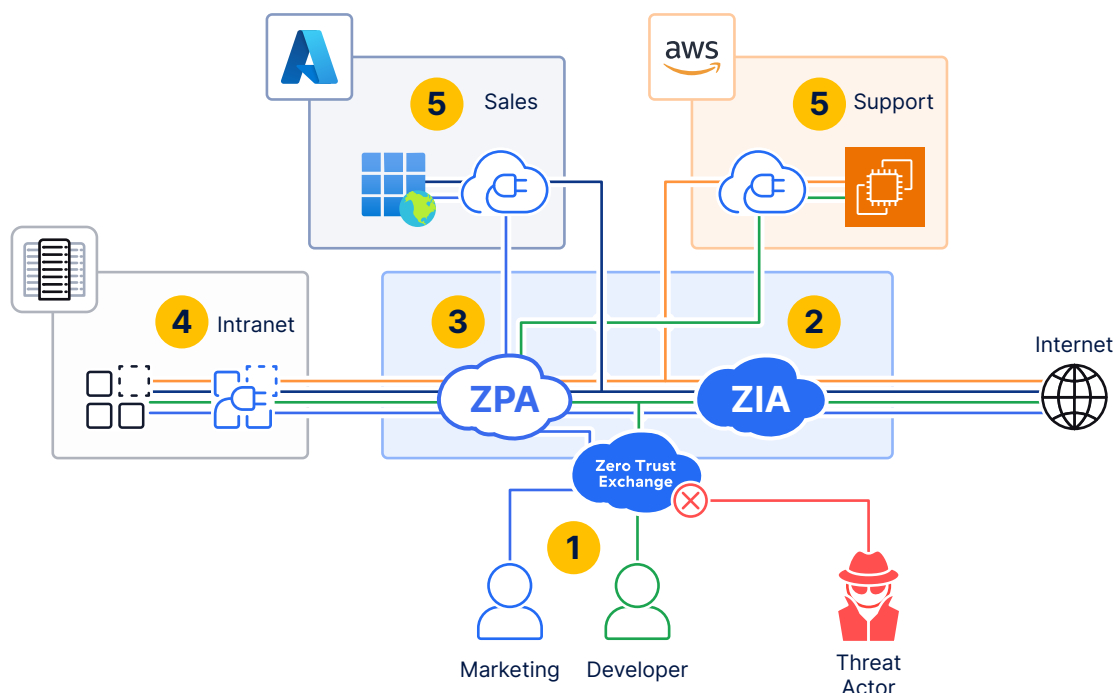


Figure 2: Zero Trust principles applied to users and workloads

1. **Authentication** – All users must first authenticate to Zscaler. Based on multiple criteria such as user group membership, device posture, and location, the user is assigned a set of policies. These include the ability to see internal applications.
2. **ZIA Service Edge** – When traffic from users or workloads needs to be routed to the internet, a ZIA Service Edge inspects the traffic. If your policy allows the traffic out, the return traffic is also scanned for malicious content on its way back to the user.
3. **ZPA Service Edge** – Traffic from users or workloads bound for other internal applications is handled by ZPA. Based on the user's authentication and assigned policy, only approved resources are resolved. All other resources are hidden from unauthorized users as if the services do not exist.
4. **App Connector** – Sitting in front of internal applications, App Connector allows ZPA connections to applications for authorized users.
5. **Cloud Connector** – Deployed in front of your internal applications, Cloud Connector creates a set of outbound tunnels to ZIA and ZPA. They decide where the tunnel connects based on policy.

In the previous model, your users in marketing have workloads running in Microsoft Azure, and your developers have workloads in Amazon Web Services (AWS). All users have access to internal applications in the data center, as well as general internet access. In this case, each user and workload is limited to which applications they can resolve and access, based on the policy applied to them.

Your marketing user (blue) can access their workloads in Azure, the data center, and internet based on policy. Your developer (green) can access their workloads in AWS, the data center, and internet. Finally, your workloads in Azure, AWS, or your data center can all reach one another and the internet via the ZTE, without the need to set up additional VPN links. All these connections are subject to the policy you set.

Cloud Connector ensures that cloud workloads adhere to organizational security policy when accessing both public and private endpoints. This is achieved by intelligently forwarding traffic to the Zscaler Internet Access (ZIA) and Zscaler Private Access (ZPA) platforms. Cloud Connector also enables multi-cloud connectivity and enforces a security policy for cloud-to-cloud traffic.



Cloud Connector deployments in AWS are typically done in a transit or security VPC. Much like the security gateways in your DMZ, these gateway VPCs provide an inspection point for data transiting in or out of your workload VPCs. The Cloud Connector appliances sit between your Gateway Load Balancer (GWLB) and your internet gateway to enforce policy.

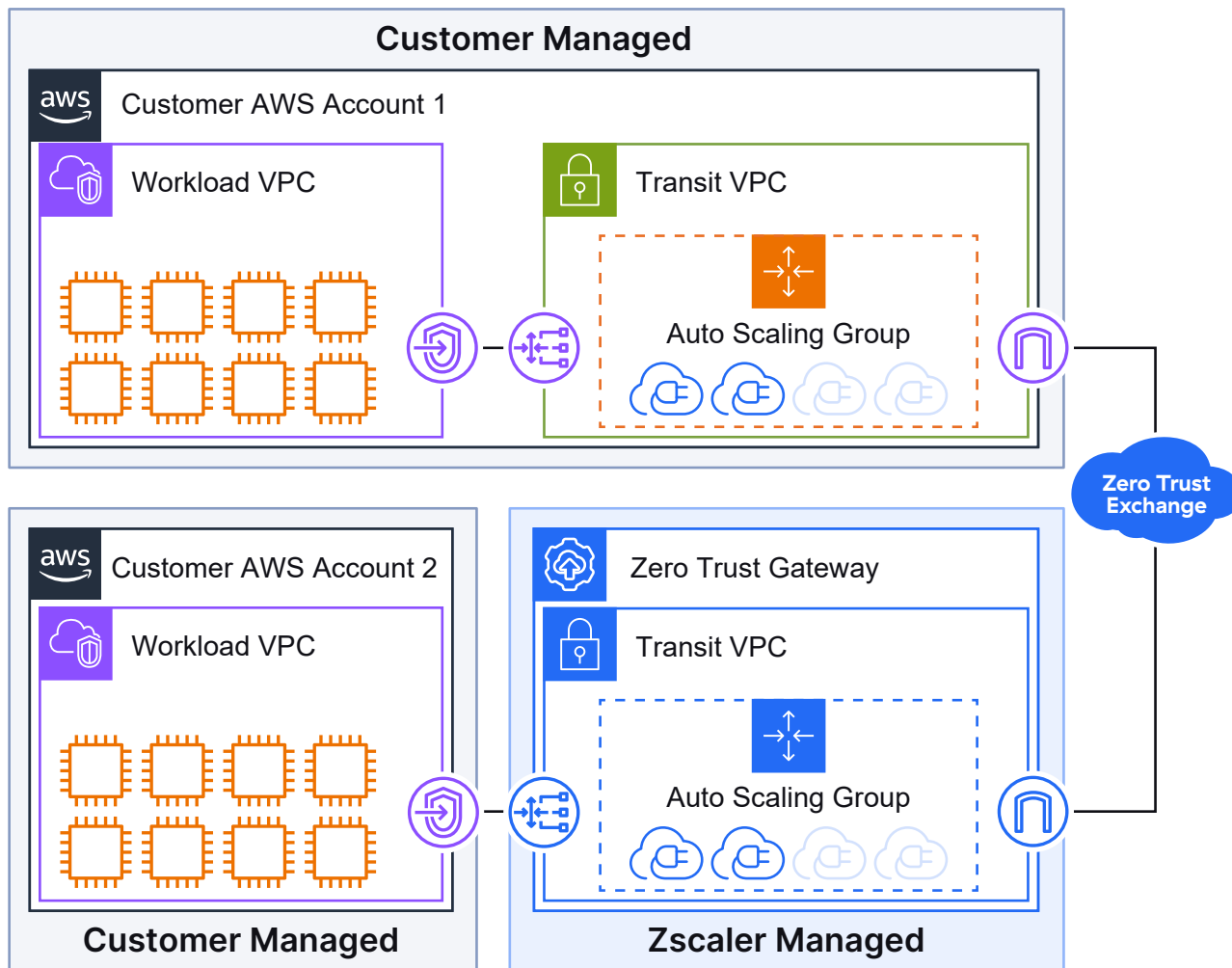


Figure 3: Deploying Cloud Connector can be done by your team or provided as a service by Zscaler

The Zscaler Zero Trust Gateway service simplifies cloud workload security. This subscription service provides a complete security gateway managed by Zscaler in your AWS tenant. Zscaler's Zero Trust Exchange deploys and manages a gateway VPC complete with Cloud Connectors and load balancers. Your teams simply point your traffic to the transit gateway. Zscaler handles traffic forwarding, auto-scaling Cloud Connector instances, traffic management, and software maintenance. Zscaler also handles the transit costs for your data as a part of your subscription.

Cloud Connector can also be deployed in your organization like any other security and inspection service, either in the cloud or in your data centers. For organizations with robust AWS deployments and skill sets, it might make sense for the cloud team to manage the Cloud Connector services themselves. This includes setting up the necessary load balancing services and horizontal auto-scaling of Cloud Connector devices. Your team would also handle software updates, routing, and other routine integration tasks.

## Key Features and Benefits

- Reduce complexity by connecting directly to the internet and eliminate the need for complex routing configurations through source NAT, transit gateways, and transit hubs.
- Total visibility, control, and awareness for workload communications. Centralized logging and real-time streaming can also be used with third-party monitoring solutions.
- Flexible scalability with elastic, horizontal scaling made possible through the Zero Trust Exchange architecture, which operates in over 150 global data centers.
- High availability with built-in automatic failover with N+2 redundancy is provided for forwarding and security. Additional redundancy can be built into the AWS deployment via Auto Scaling groups and warm pools.
- Lower operational costs by removing expenses associated with complex network configurations, network service replication, and hidden costs for cloud connectivity.

## New to Zscaler Cloud Connector?

If this is your first time reading about Zscaler Cloud Connector, we encourage you to explore the following resources:

- If you are new to AWS, Amazon offers a range of courses for different roles at [AWS Training and Certification](https://aws.amazon.com/training/) (<https://aws.amazon.com/training/>).
- To watch a demonstration of Zscaler Cloud Connector, see [Zero Trust Your Cloud Workloads](https://www.youtube.com/watch?v=S-g_qmuxnqU&t=1845s) ([https://www.youtube.com/watch?v=S-g\\_qmuxnqU&t=1845s](https://www.youtube.com/watch?v=S-g_qmuxnqU&t=1845s)).
- Zscaler Internet Access (ZIA) provides outbound internet protection for users. Learn more at [Zscaler Internet Access](https://www.zscaler.com/products/zscaler-internet-access) (<https://www.zscaler.com/products/zscaler-internet-access>).
- Zscaler Private Access (ZPA) provides private access to applications, not networks. Learn more at [Zscaler Private Access](https://www.zscaler.com/products/zscaler-private-access) (<https://www.zscaler.com/products/zscaler-private-access>).
- To learn more about the Zero Trust architecture concept, we recommend the National Institute of Standards and Technology (NIST) paper [Zero Trust Architecture](https://www.nist.gov/publications/zero-trust-architecture) (<https://www.nist.gov/publications/zero-trust-architecture>).

## Cloud Infrastructure Protection Using Cloud Connector

As organizations began moving workloads to the cloud, securing those resources became a challenge. Securing applications against attack both from outside actors and malicious content on legitimate sites led some organizations to provide access to applications over VPN. Attempting to leverage legacy security products simply adds latency and frustration for your users. As cloud usage expanded, you now have data moving between the cloud and the user, between systems in the cloud vendor, and between applications across cloud vendors and your legacy data center.

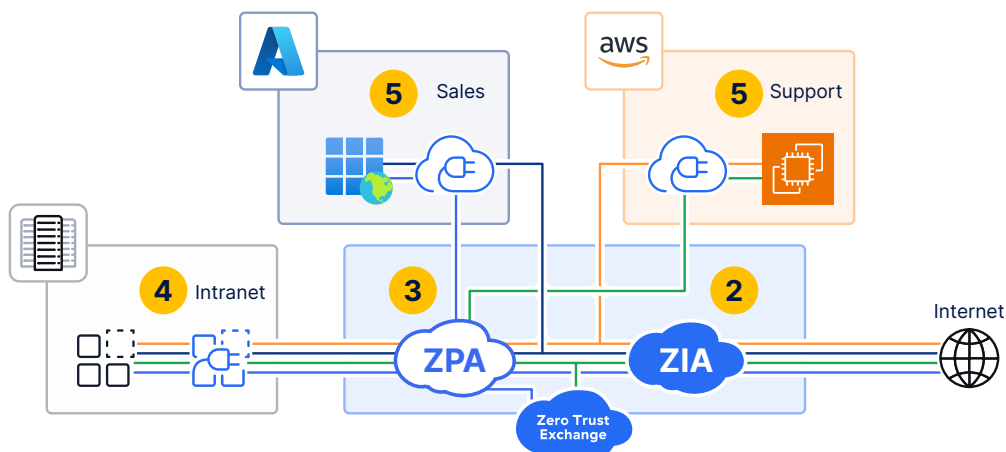


Figure 4: Workload communication between private and public applications

The communication can be from private workloads (IaaS or physical DC) to public workloads (SaaS internet application), or between private workloads (IaaS to IaaS, or physical DC to IaaS). Securing these communications channels with physical or virtual appliances is cumbersome and can lead to inconsistent configuration.

In the previous example, our application `sales.azure.internal.safemarch.com` sits behind a Cloud Connector with access to both ZPA and ZIA platforms. In this model, the workload can reach out to the support workloads in AWS, allowing the sales team to file support and product requests without logging into the support portal. The sales portal is accessed by our intranet workload in our data center to pull deals and rankings for the company dashboard. Finally, our sales workload can reach the internet to update our cloud CRM, which in turn only accepts connections from Zscaler IP addresses for our tenant.

Zscaler Cloud Connector virtual machines extend the security of ZIA and ZPA to cloud native workloads. ZIA protects your workload traffic communicating with a public application. ZPA protects your communications between private workloads. This allows organizations to secure all workload communications over any network. The Zscaler Zero Trust Exchange allows workloads to communicate with each other with a granular security policy applied.

- Applications-to-Internet Communication for applications that might need to access any internet or SaaS destination, such as third-party APIs, software updates, etc. A scalable, reliable security solution that inspects all transactions and applies advanced threat prevention and data loss protection controls.
- Application-to-Application Communication to other public clouds and corporate data centers for multi/hybrid cloud connectivity. Delivered with better security and a dramatically simplified operational model, as compared with traditional solutions like proxies, virtual firewalls, and IDS/IPS.
- Application-to-Application Communication within a Virtual Private Cloud by securing process-to-process communication. This achieves microsegmentation of traffic with no changes to the application or the network.

## Leveraging Workload Discovery Service for Policy Action

Zscaler Client Connector provides an additional feature called Workload Discovery. Workload Discovery can discover instances of running workloads in your AWS tenant. As a part of the discovery process, Workload Discovery learns about metadata tags associated with your VMs and VPCs. This includes AWS-generated tags and user-defined metadata tags. AWS is polled every 5 minutes by default to discover any new workloads or tags associated with them. You can reduce this polling interval to speed up the discovery of tag key-value pairs. The available AWS predefined tags include the following items:

- **GroupId:** The ID of the security group assigned to the attached Elastic Network Interface (ENI). This can identify the AWS Lambda function workload.
- **GroupName:** The name of the security group assigned to the attached ENI.
- **ImageId:** The ID of the image used to launch the instance.
- **PlatformDetails:** The details of the OS running on the instance. Zscaler also supports AWS Lambda and other services when not used in the context of Amazon Elastic Compute Cloud (EC2).
- **Vpc-id:** The ID of the virtual private cloud (VPC) that the ENI runs in.
- **IamInstanceProfile-Arn:** The Amazon Resource Name (ARN) of the Identify Access Management (IAM) instance profile.

The AWS Tag Editor allows you to define key-value pairs that become part of the accessible metadata tags for the workload. When you start a workload in AWS, you supply your own tag keys and values that are later used to identify and group your workloads together. Typically, organizations attach tags to deployed resources to add context for management activities including cost allocation, organizational ownership, or operational needs. When that workload is discovered by Zscaler, those tags become available for you to use in policy creation via workload groups.

In the ZIA and ZPA Admin Portals, a workload group is a new attribute that can be used as a source in different policies. Workload groups allow you to select tags using 'AND' or 'OR' expressions to match resources. This definition can include types such as VM and VPC as tag types, and then specify the key-value pairs to complete the definition. You define a workload group that matches the tags and values you supplied to your workloads. The workload group can then be used in ZIA and ZPA policy creation. In ZIA, workload groups are available in URL filtering, firewall policy, TLS/SSL inspection, and DLP policies. In ZPA, you can leverage workload groups in access policies.

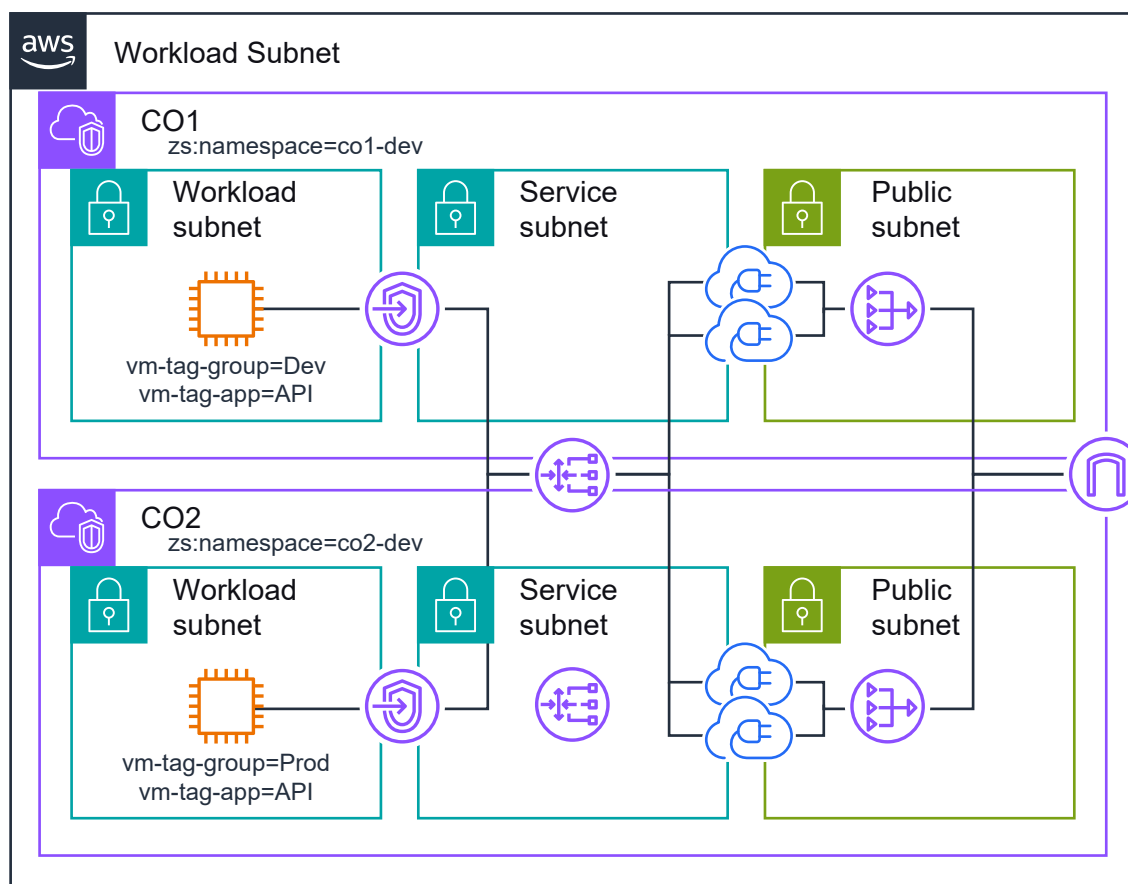


Figure 5: Your instances can be grouped using key-value pair tags you define, as well as AWS-defined tags

As an example, you deploy both production and development instances of your API application. Using the previous model, a single workload group is created by selecting the following tags:

```
vm-tag-group=Dev OR vm-tag-group=Prod
```

```
AND
```

```
vm-tag-app=API
```

```
AND
```

```
VPC-1 OR VPC-2
```

This selection policy says VMs are tagged with Dev or Prod, tagged API, and reside in VPC-1 or VPC-2. Give the workload group a name such as "API instances," and you can then use this single "API Instances" object to apply policy to all of your API application workloads. You can further build groups that only encapsulate the production or development servers, or only applications in certain VPCs.

- Watch a demonstration of this feature at [Implementing Policy with AWS Tagging for Workload Communications](https://www.youtube.com/watch?v=unVEkU-Uo9E) (<https://www.youtube.com/watch?v=unVEkU-Uo9E>).
- Learn more at [Configuring Workload Discovery for Workloads in Amazon Web Services](https://help.zscaler.com/cloud-branch-connector/configuring-workload-discovery-workloads-amazon-web-services) (<https://help.zscaler.com/cloud-branch-connector/configuring-workload-discovery-workloads-amazon-web-services>).
- Learn more about [AWS tags](https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html) (<https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html>).

## Managing Overlapping IP Addresses with Namespace Tags

There are instances where you might find you have overlapping IP address space in your deployments. This is often due to merger and acquisition activity between organizations, or previously isolated groups within an organization being brought together. In these cases, Zscaler policy engines need to uniquely identify the correct device to apply the appropriate policy which can't be done by leveraging IP addresses alone.

To give machines unique identifiers, you can apply namespace tags at the VPC level. When these tags are set, the workloads and GWLB endpoints have the namespace label applied and can be used as a part of the policy definition in addition to IP addresses.

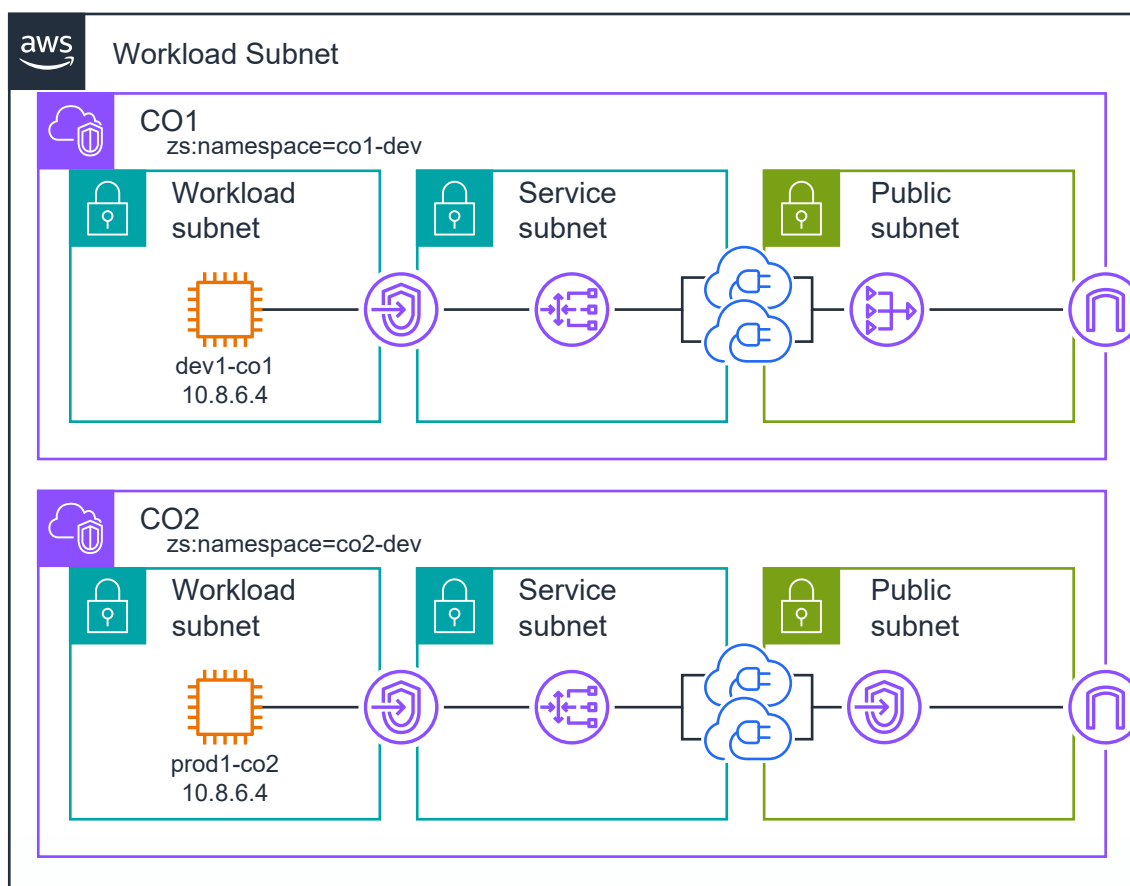


Figure 6: Namespace tags allow the same private IP addresses to be reused while having a unique identifier for each workload

Consider the previous example where two organizations, CO1 and CO2, are merging. Both organizations use an overlapping subnet space in their deployments. When Cloud Connector is deployed, it discovers both instances of duplicate IP address use. By applying namespace tags to the VPCs, you add another identifier besides the IP address to uniquely identify each device. The discovered instance data is summarized in the following table.

VM ID	Namespace ID	VM IP Address
dev1-co1	co1-dev	10.8.6.4
prod1-co2	co2-prod	10.8.6.4

Namespace tags use the following format where [VALUE] is replaced with the name for the namespace:

```
Tag zs:namespace = [VALUE]
```

Zscaler recommends setting the tag at the VPC level. This allows you to have common deployment scripts for workloads. These devices then inherit the name space from the VPC level tag when the workload is deployed.



Zscaler recommends setting namespace tags at the VPC level for consistency in deployment and resource assignment.

To learn more, see [Namespace and Duplicate IP Addresses](https://help.zscaler.com/cloud-branch-connector/configuring-workload-discovery-workloads-amazon-web-services#namespace-duplicate-ips) (<https://help.zscaler.com/cloud-branch-connector/configuring-workload-discovery-workloads-amazon-web-services#namespace-duplicate-ips>).

## Choosing a Cloud Connector Deployment Model

Cloud Connectors are virtual machines deployed most often in a transit or security VPC. The deployment can be handled by Zscaler via the Zero Trust Gateway service, or deployed by your teams as part of your AWS deployment. The difference in the deployments is who is responsible for deploying and maintaining the Cloud Connector instances and associated services. All product features and capabilities are available in both deployment models.

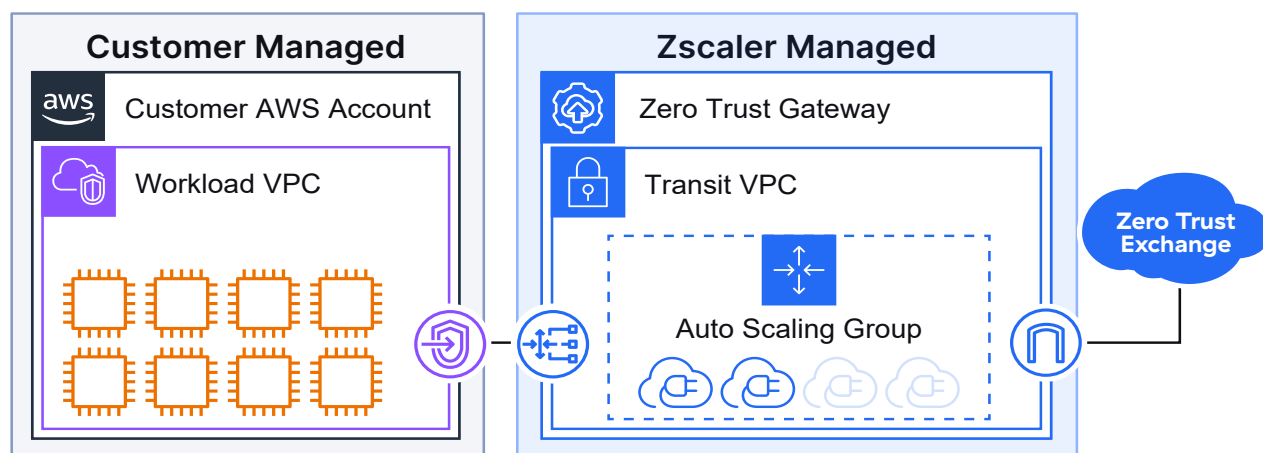


Figure 7: Zscaler Zero Trust Gateway provides the same services, but with all management tasks handled by Zscaler

The Zscaler Zero Trust Gateway service provides all the transit VPC features for you including your transit VPC. This includes your internet gateway, Cloud Connectors, routing, load balancing, scaling, and maintenance. Zscaler takes care of the complexity, setup, and transit billing for your cloud deployment.



Your team will deploy an AWS Gateway Load Balancer Endpoint (GWLBe) in each of your workload VPCs or in existing transit VPCs. This endpoint is responsible for taking the traffic from your workloads and sending it to the Zero Trust Gateway. The GWLBe is configured with a URL provided to you in the Cloud Connector Admin Portal pointing to your Zscaler-managed transit VPC. Your workloads then use the GWLBe to forward traffic to Zscaler.

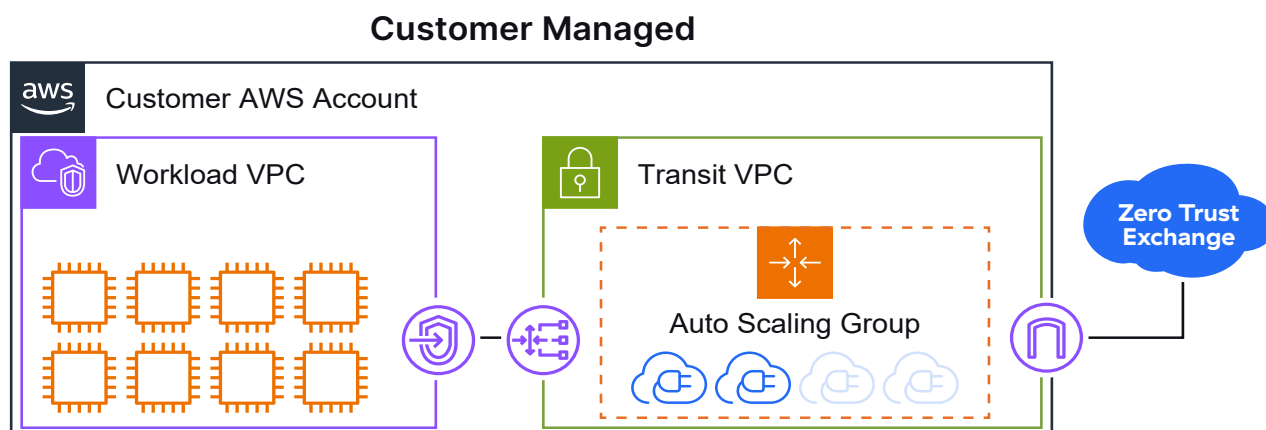


Figure 8: Zscaler Cloud Connectors can be deployed as a service by your AWS team

When your team deploys Cloud Connector instances, they are responsible for the management and maintenance of those instances. This includes updating images, ensuring routing is handled correctly, and managing horizontal scale-out and scale-in events as traffic patterns change. Routing and other tasks also remain within your team's control.

To learn more about Zero Trust Gateway subscriptions and deployments, contact your Zscaler Account team.

## Deploying the Zero Trust Gateway Service

Subscribing to the Zero Trust Gateway service eliminates the need for your teams to deploy and manage Cloud Connector appliances. Instead, Zscaler assumes responsibility for deploying, managing, and maintaining the cloud infrastructure on your behalf. Your operations team can focus on managing your cloud infrastructure and applications, and Zscaler manages Cloud Connectors and their supporting infrastructure.

To get started deploying a Zero Trust Gateway, the following information is required from your AWS account:

- **One or more AWS account numbers** – You can associate up to 128 AWS accounts with a single Zero Trust Gateway.
- **AWS deployment region** – This region in the AWS cloud is where the Zero Trust Gateway will be deployed. The selected region should be the same as your workload location.
- **Name of the instance** – An identifier for your Zero Trust Gateway.

After configuration, you are supplied an Endpoint Service Address that corresponds to a Gateway Load Balancer (GWLB) running in the Zero Trust Gateway. You will deploy and configure an AWS Gateway Load Balancer Endpoint (GWLBe) in your VPC using the provided Endpoint Service Address.

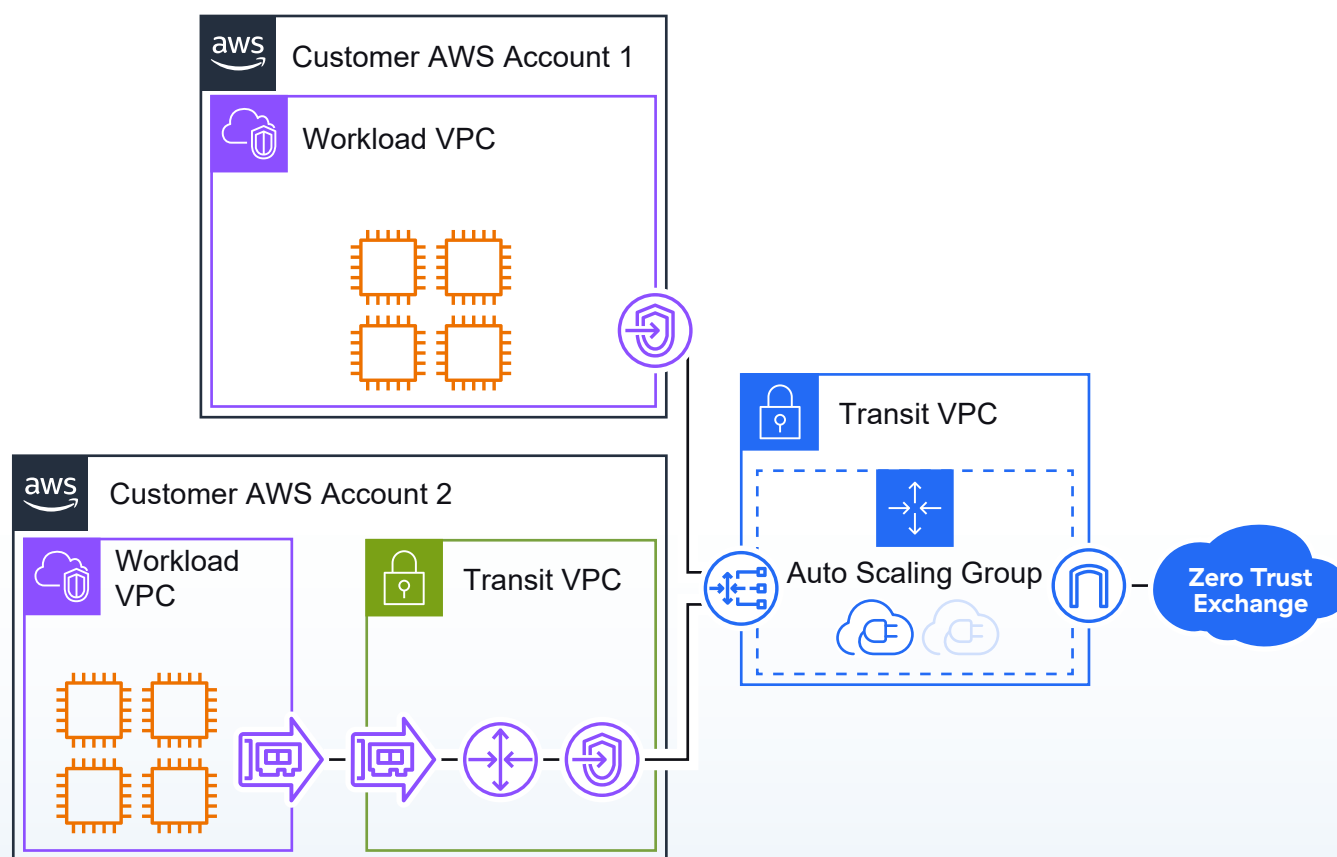


Figure 9: The Zero Trust Gateway can be leveraged from multiple deployment styles and AWS accounts

Placement of the GWLBe can happen at multiple locations based on your existing AWS architecture. In a distributed model, the GWLBe sits at the workload VPC level. It is the default gateway for the workloads and sends traffic directly to the Zero Trust Gateway. If you have a centralized transit gateway in place, the GWLBe can be placed within that VPC. This simplifies the number of network changes that need to be made. It is also possible to run a hybrid of the two models.

Using this information, the Zscaler Zero Trust Exchange deploys your Zero Trust Gateway. It generally takes less than 10 minutes to complete the setup. When complete, you are provided the URL of the GWLB for your Zero Trust Gateway deployment. In your workload VPCs, your teams will deploy Gateway Load Balancer Endpoints (GWLBe). When configuring the endpoints, you use the supplied URL to point them to the Zero Trust Gateway. Your workloads must be updated to use the endpoints as their default route out of the VPC.

Zscaler then begins to monitor your Zero Trust Gateway. Logs become available in the Cloud Connector Admin Portal with your other logs. Zscaler handles maintenance and automatically scales Cloud Connector instances based on demand. Your team is only responsible for policy creation for your subscribed services. All features are available for configuration, including discovery services and workspace tags.

To learn more about Zero Trust Gateway subscriptions and deployments, contact your Zscaler Account team.

## Custom Cloud Connector Deployments

The following section outlines your options when deploying Cloud Connector yourself, either in AWS or in your own data centers. You can design your network using the tools that best match your cloud deployment. We recommend that you review each use case to familiarize yourself with the various options, which can be combined to meet your organization's deployment needs. For example, in many production environments Cloud Connector would be deployed in a transit, security, or egress VPC providing outbound internet and application access.

### Pre-Deployment Considerations

The following sections provide some general design recommendations common to all deployment types.

#### NAT Gateway vs. Internet Gateway

Zscaler recommends that the Cloud Connector appliances leverage the NAT gateway functionality of AWS for outbound internet access, as opposed to a simple internet gateway, for the following reasons:

- **Scalability** – NAT gateways can provide outbound internet access for multiple hosts (Cloud Connectors) using a single elastic IP address.
- **Conserve public IP space** – NAT gateways use a single elastic IP address, reducing your costs compared to internet gateways that require each host to have their own unique elastic IP addresses.
- **Security** – NAT gateways are stateful in nature, allowing traffic initiated from the “inside” zone towards the “outside” (public internet) as well as the corresponding return traffic. However, the NAT drops traffic initiated to the “inside” from the “outside” by default. This prevents attackers from directly targeting the Cloud Connector appliance from the outside. Internet gateways, by contrast, allow bidirectional communication and expose hosts with elastic IP allocations directly to the public internet from the outside.

Learn more about [AWS NAT gateways](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html) (<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>).

In cases where internet gateways are the only option for your organization, Zscaler recommends that AWS Security Groups and network Access Control Lists (ACLs) be used to limit access to the Cloud Connector interfaces and where the Cloud Connector can direct traffic. Outbound traffic from the Cloud Connectors should be limited to ZIA Public Service Edge and ZPA Public Service Edge IP addresses. You can find a listing of Zscaler public IP ranges, ports, and protocols in use by each Zscaler cloud at [Cloud Enforcement Node Ranges](https://config.zscaler.com/zscaler.net/cenr) (<https://config.zscaler.com/zscaler.net/cenr>).

Network ACLs act as stateless firewalls for traffic entering the VPC and allow you to permit or deny traffic to or from subnets. By default, a Network ACL is created with every subnet and allows all traffic. You can adjust the ACL to drop all inbound traffic. Learn more about the [Access Control List \(ACL\)](https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html) (<https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html>).

Security groups operate at the network interface layer Elastic Network Interface (ENI) of the instance. This service controls inbound and outbound traffic from the instance. Learn more about [controlling traffic](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html) ([https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html)).

Learn more about [AWS internet gateways](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html) ([https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_Internet\\_Gateway.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html)).

## Availability Zones

Zscaler recommends that Cloud Connector appliances be installed in pairs for high availability. When building high-availability pairs of Cloud Connector appliances, Zscaler recommends that each appliance be deployed in different availability zones within the same region. This ensures that individual Cloud Connector appliances exist on physically separate pieces of underlying hardware from one another and provide failover access.

Learn more about [AWS availability zones](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/using-regions-availability-zones.html#concepts-availability-zones) (<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/using-regions-availability-zones.html#concepts-availability-zones>).

## Network Connectivity

The Cloud Connector appliance has two interfaces: a Service Interface, where workload traffic is brought in and DTLS tunnels are terminated towards the Zscaler cloud, and a Management Interface. Both interfaces are associated with a Service and Management Subnet, respectively. If needed, they can share a single subnet. When created, subnets are associated with an availability zone. When creating subnets for a high-availability deployment of Cloud Connector, Zscaler recommends that you create them in pairs as well, with each subnet associating to a different AZ.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

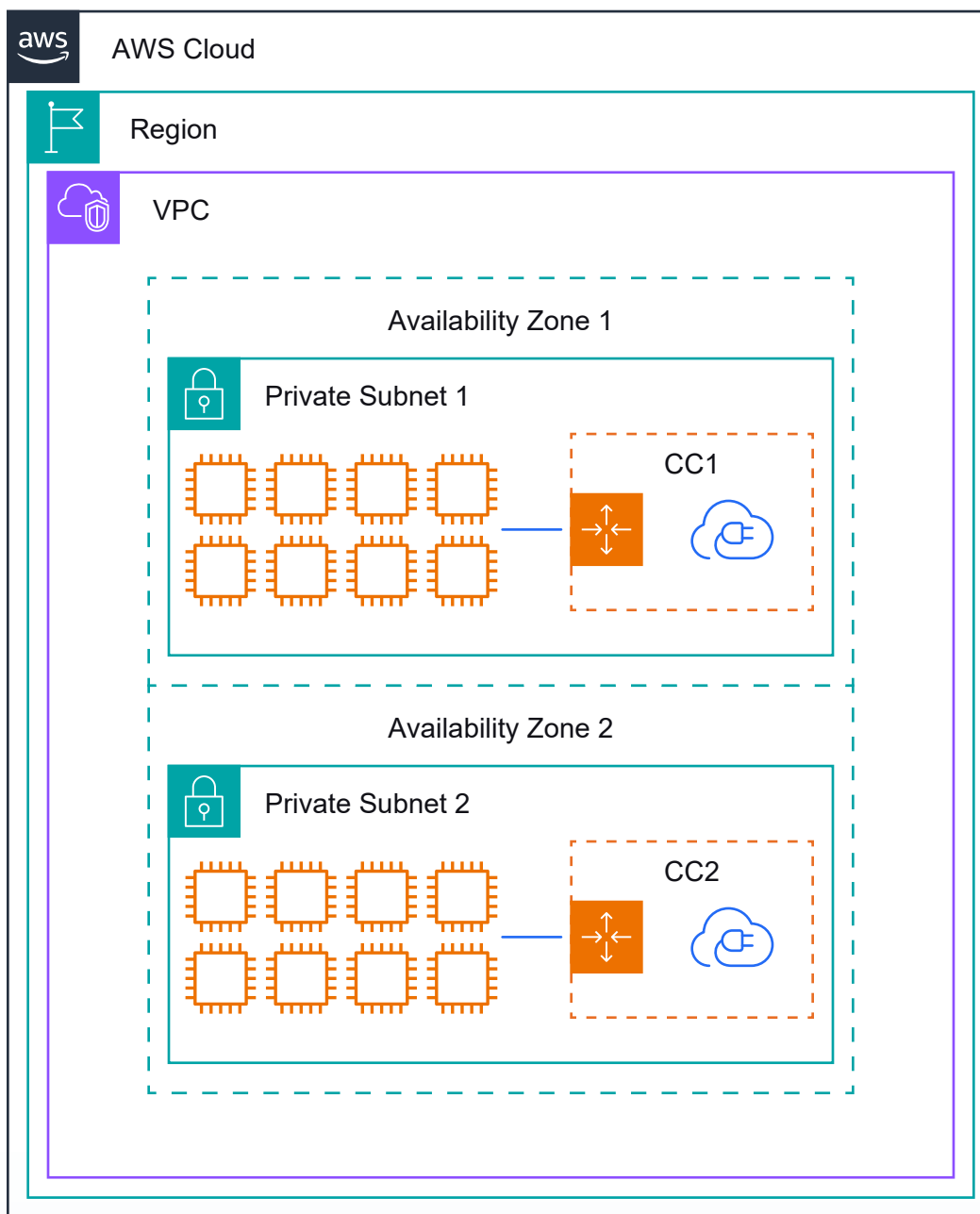


Figure 10: Service Interface/ENI interfaces spread across two subnets and two availability zones

For example, the Service Interface/ENI of Cloud Connector 1 connects to Subnet 1 in Availability Zone 1, while the Service Interface/ENI of Cloud Connector 2 connects to Subnet 2 in Availability Zone 2. This ensures the appliances maintain physical separation from each other at the network level.

Additionally, when employing NAT gateways as recommended, consider the fact that a NAT gateway also has an availability zone associated with it, and that NAT gateways can be shared across availability zones. However if a failure of the NAT gateway occurs, this can lead to internet connectivity issues across all availability zones that share the NAT gateway. For this reason, Zscaler recommends that NAT gateways also be deployed as a pair, with one gateway in each of the Cloud Connector availability zones.

## Virtual Compute

Just as subnets ensure physical separation at the network level, EC2 instances can also be deployed within availability zones. This ensures that individual Cloud Connector appliances exist on physically separate pieces of underlying hardware from one another. When building high-availability pairs of Cloud Connector appliances, Zscaler recommends that each appliance be deployed within different availability zones.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

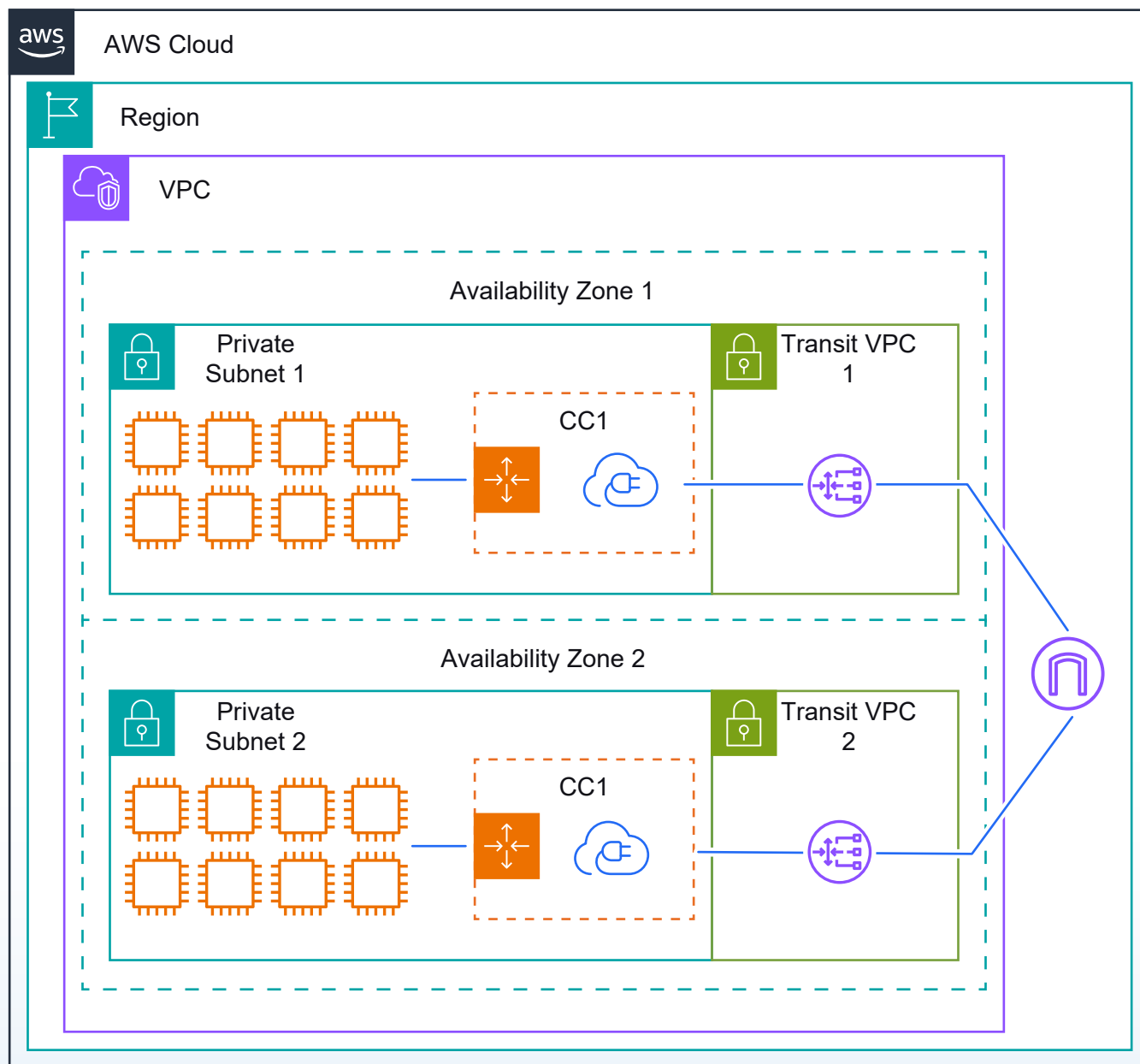


Figure 11: Cloud Connector appliances in two different AWS availability zones

Cloud Connector is delivered in several form factors. It is available as a virtual appliance in Amazon Web Services, Microsoft Azure, and Google Cloud Platform, as well as for VMs for on-premises deployment.

If you are deploying on AWS:

- Zscaler recommends the *c5* or *m5* instance size to support Cloud Connector as it offers the best throughput performance.
- The appliance is available on the [Amazon marketplace](https://aws.amazon.com/marketplace/pp/prodview-cvzx4oiv7oljm?sr=0-2&ref_=beagle&applicationId=AWSMPContessa) ([https://aws.amazon.com/marketplace/pp/prodview-cvzx4oiv7oljm?sr=0-2&ref\\_=beagle&applicationId=AWSMPContessa](https://aws.amazon.com/marketplace/pp/prodview-cvzx4oiv7oljm?sr=0-2&ref_=beagle&applicationId=AWSMPContessa)).

For on-premises deployments, the image requires:

- VMware ESXi and CentOS/Linux (KVM) images
- 2 virtual CPUs
- 4 GB of RAM

## High Availability Deployment Design

Cloud Connector leverages AWS Gateway Load Balancer (GWLB) functionality to achieve high availability and horizontal scalability. The GWLB is made up of two components: the Gateway Load Balancer, and the Gateway Load Balancer Endpoint (GWLBe). The GWLBe is a VM that is deployed in the same VPC as your workloads. Outbound traffic from workload VPCs route their traffic to the GWLBe.

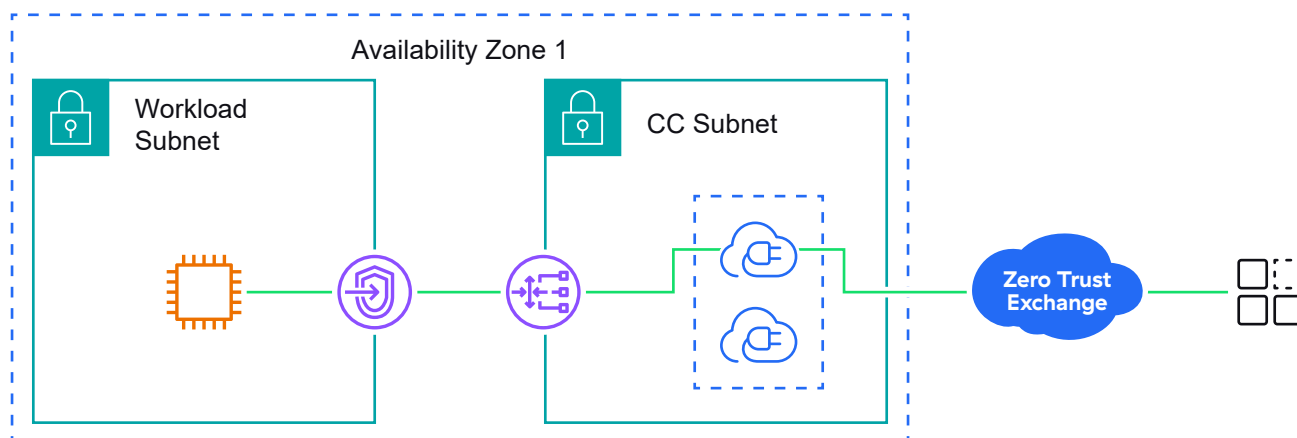


Figure 12: AWS Gateway Load Balancer distributes requests from workloads across available Cloud Connector instances

Between the GWLBe and the GWLB, traffic is tunneled using the Geneve protocol. Geneve encapsulation, as with any overlay tunneling protocol, allows a service provider to transparently route traffic towards value-add services while keeping the original packet intact. In the case of Cloud Connector, this means that Route Tables remain simple, while still allowing horizontal scalability. The transparent tunneling to the Cloud Connectors means that inspection can be inserted nearly anywhere within the cloud environment, and traffic can be tunneled between VPCs and tenants.

Deployment can occur within the same VPC as the workloads or attached to a transit VPC. Zscaler recommends a minimum of two Cloud Connector appliances, each in a different availability zone (AZ).

- Learn more about [Gateway Load Balancer](https://aws.amazon.com/blogs/networking-and-content-delivery/best-practices-for-deploying-gateway-load-balancer/) (<https://aws.amazon.com/blogs/networking-and-content-delivery/best-practices-for-deploying-gateway-load-balancer/>).
- Learn more about [Geneve: Generic Network Virtualization Encapsulation](https://www.rfc-editor.org/rfc/rfc8926.html) (<https://www.rfc-editor.org/rfc/rfc8926.html>).



## Gateway Load Balancer Health Checks

The GWLB needs to keep track of which machines are up and running in the network. To do this, the GWLB performs periodic health checks on each virtual machine. By default, this happens every 10 seconds, sending an HTTP request to the virtual machine to verify its readiness to accept traffic. If three consecutive checks fail, the instance is removed from the available pool for a total of 30 seconds.



These settings can be adjusted down to 5 seconds and 2 missed heartbeats for a total of 10 seconds before the resource is removed. While this can speed discovery of an unresponsive device, care should be taken that devices are not removed from service due to temporary congestion.

## Gateway Load Balancer Traffic Distribution

When traffic reaches the GWLB, it distributes the traffic among available Cloud Connector resources. The GWLB distributes traffic among available instances using a 5-tuple hash: source and destination IP address, source and destination port, and protocol in use hash towards the Cloud Connector appliances. Using a 5-tuple hash ensures that inbound flows from EC2 instances are more efficiently balanced across all available appliances, even when the source or destination IP address is the same.



The 5-tuple hashing can break certain traffic flows which use multiple source or destination ports as part of a single session (such as FTP traffic), resulting in unpredictable behavior. A workaround is addressed in the following section.

## Avoiding Traffic Asymmetry with Load Balancers

As mentioned, the default AWS GWLB distribution hash is 5-tuple which provides highly granular and efficient traffic distribution. However, it has been known to create problems for cloud environments and applications where multiple traffic streams exist between the stations and traffic symmetry is important. Using multiple paths prevents cloud security services like ZIA from seeing the entire traffic stream. Applications receiving traffic from multiple source IP addresses from the same user typically cause the application to error and fail.

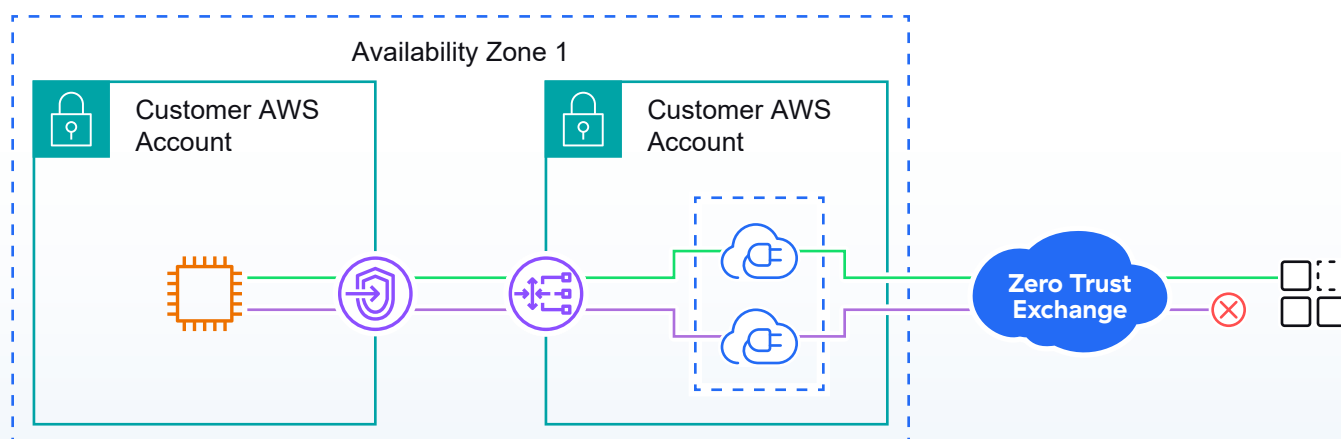


Figure 13: The 5-tuple hash can result in applications taking different routes to the same resource when the port changes

Applications that operate on multiple ports can be broken due to different streams being routed through different Cloud Connectors. This can lead to traffic flows with multiple source IP addresses, causing the application to break. Security devices such as cloud proxies and cloud firewalls must see all packets within a given flow to maintain accurate session state and provide predictable application behavior.

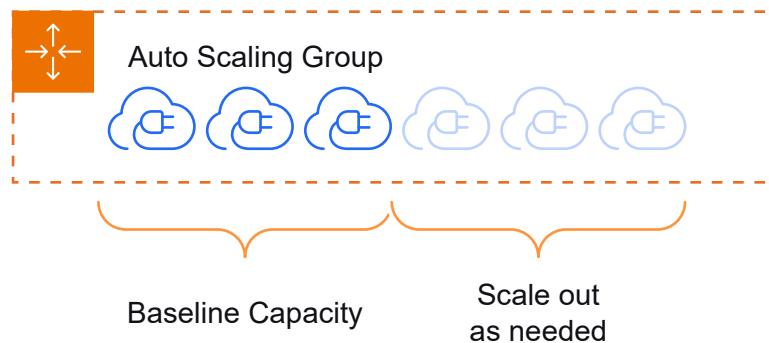
For these situations, Zscaler recommends adjusting the Subnet route table to avoid the GWLB and directing traffic at the Cloud Connector Service Interface instead. This can be accomplished by installing a more specific route to the destination within the Subnet route table that targets the Cloud Connector Service ENI. For fault tolerance, AWS Lambda can be invoked using CloudFormation scripts (found in the Deployment Templates section of the Administration menu) to ensure this route is updated if the primary appliance is unavailable. If a Cloud Connector appliance fails, Lambda functionality automatically updates route tables to redirect traffic to the active appliance in the adjacent AZ.

Learn more about [AWS Lambda](https://aws.amazon.com/lambda/) (<https://aws.amazon.com/lambda/>).

## Leveraging Auto Scaling Groups for Redundancy

The AWS service uses the concept of Auto Scaling groups, which allows you to set a minimum, desired, and maximum number of instances of your application. These groups allow you to configure the number of instances you need for your average connection load, automatically bursting when needed. This allows you to purchase long-term reserved instances for your baseline capacity, lowering cost. You can also be ready to expand automatically should something unexpected occur.

 Auto Scaling groups are optional, and are not required for a ZPA deployment.



*Figure 14: Scale Cloud Connector instances out to meet surprise spikes in demand*

Auto Scaling groups are a form of redundancy and additional capacity. For example, if you know your users working from your headquarters always hit the same Cloud Connector instances, you want to scale those to meet your average daily load. Then, you can build an Auto Scaling group to account for any large temporary influxes of users, such as company meetings or customer events you host.

Another reason for building Auto Scaling groups for redundancy is to handle outages. Should access to a set of Cloud Connectors become unavailable due to an outage, users will fail over to the next nearest data center. If those Cloud Connectors become saturated, your ability to automatically launch a new instance to scale up automatically can help mitigate load issues.

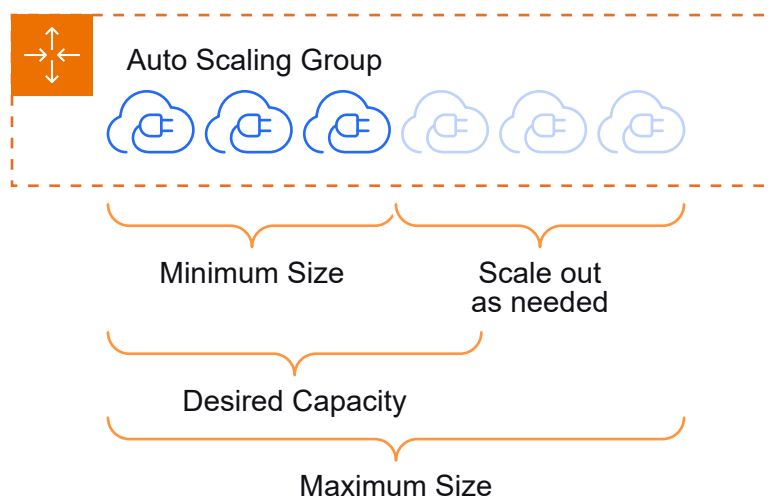


Figure 15: Auto Scaling group of Cloud Connectors

You should set the minimum scale size to match your expected daily load. These would be your reserved instances. You should set the maximum large enough to account for any expected burst load. When you no longer need the extra instances, the Auto Scaling group can terminate the instance. User sessions will be moved to the remaining instances prior to stopping the Cloud Connector instances.

There are 4 components to leveraging Auto Scaling Groups:

### Warm Pool

When planning your Auto Scaling groups, you should keep in mind that Auto Scaling is not instantaneous. Triggering a new Cloud Connector instance to come online still requires that instance to spin up, connect to the Zscaler cloud, perform any updates, and finally start serving clients. To avoid this delay, the concept of warm pools exists using previously initialized VMs. These instances are brought up preemptively as load increases, allowing them time to initialize and register with the cloud. The instances can then be brought up or down as needed with minimal delay.

Zscaler recommends deploying Auto Scaling Group with Warm Pool to reduce the amount of time it takes to increase capacity with a scale-out event. Instances in the warm pool will show as inactive in the Cloud Connector Portal when they are in the stopped state. Instances in the warm pool can be pulled back in service from a stopped state due to the instance reuse policy.

## Launch Templates

Launch templates provide instance configuration information to AWS, allowing the Auto Scaling group to dynamically add capacity to your deployment. Launch templates can contain AWS Marketplace AMI ID, Security Groups, SSH keypairs, and which version of Cloud Connector instances should be brought up. When setting your values of your launch templates, Zscaler recommends setting your Cloud Connector group to a minimum and desired value of at least two instances for redundancy purposes.



Be sure to configure your Cloud Connector provisioning key to allow for the same level of maximum instances that are in the Auto Scaling group. If the provisioning key maximum is lower than the Auto Scaling group maximum, new Cloud Connectors will fail to launch successfully.

## Lambda Monitoring

The AWS Lambda service is used to provide monitoring of CloudWatch logs. A script is deployed as part of the installation process. When your deployment has unused or unhealthy instances, they can be removed from service. This typically happens when scaling down instances as demand decreases.

## Auto Scale Policy

When configuring thresholds to scale out or scale in the number of instances, the Auto Scale Policy is where you configure your thresholds. In the case of Cloud Connector, the Auto Scale Policy leverages custom metrics inside the CloudWatch Zscaler namespace. This provides access to utilization data such as CPU, memory, bytes in/out, and the health of the instance.

## Auto Scaling Group Considerations

Zscaler recommends leveraging Auto Scaling groups for your deployment to help automatically provide capacity and containing costs. If you have previously deployed Cloud Connector, there are some considerations to keep in mind:

- If using CloudFormation, the Lambda script should be housed within an S3 bucket that you control. This script should be uploaded to that bucket prior to initializing Auto Scaling group.
- There is no migration of existing Cloud Connectors to Auto Scaling group instances. During launch template creation, the Marketplace AMI is used to deploy new appliances per the Min/Max configuration of the Auto Scaling group.
- Auto Scale Cloud Connectors are incompatible with any currently deployed GWLB configurations. We recommend deploying a new cluster, then adjusting route tables to utilize the new cluster after the cluster status is healthy.
- Decommissioned Cloud Connector appliances that have been deleted or are otherwise unhealthy will automatically be removed from the UI using AWS Lifecycle Hooks. In the event they are not removed, they can be deleted via the Cloud Connector API or directly in the portal.
- Scale-out events are triggered based on custom metrics exported to CloudWatch or EventBridge. The data-plane process utilization rate is the most used metric. Currently, this threshold is set at 80% or greater for groups composed of small instances. Zscaler recommends starting at this value for a more conservative and quicker scale out. This value can be lowered as needed. Zscaler recommends a lower value of 40% which is very aggressive if faster scale out response is required.

Learn more about [AWS Auto Scaling](https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html) (<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>).

To view instructions on provisioning launch templates and Auto Scaling groups, including provisioning key scripts, see [Deploying Cloud Connector with Amazon Web Services](https://help.zscaler.com/cloud-branch-connector/deploying-cloud-connector-amazon-web-services) (<https://help.zscaler.com/cloud-branch-connector/deploying-cloud-connector-amazon-web-services>).

## Cloud Connector Logging and Service Dashboards

Cloud Connector can use built-in logging functionality through the Insights page of the portal. Zscaler streams all logs to centralized log locations, allowing you to view logs from across your organization. The dashboard has views for Session Insights, DNS Insights, and ZIA Tunnel Insights. All three facilities allow you to review traffic that passes through the Cloud Connector appliance from a different perspective.

- Learn more at [Analyzing Traffic Using Insights](https://help.zscaler.com/cloud-connector/analyzing-traffic-using-insights) (<https://help.zscaler.com/cloud-connector/analyzing-traffic-using-insights>).
- Learn more about [ZIA Dashboards](https://help.zscaler.com/zia/about-dashboards) (<https://help.zscaler.com/zia/about-dashboards>).
- Learn more about [ZPA Dashboards and Diagnostics](https://help.zscaler.com/zpa/dashboard-diagnostics) (<https://help.zscaler.com/zpa/dashboard-diagnostics>).

Cloud Connector supports both the Nanolog Streaming Service (NSS) for ZIA use cases and Log Streaming Service (LSS) for ZPA use cases. NSS uses a virtual machine (VM) to stream traffic logs in real time to your security information and event management (SIEM) system, such as Splunk or ArcSight. LSS operates in a similar way, with the deployment of a ZPA App Connector VM that receives the log stream and then forwards it to the log receiver.

Both services enable real-time alerting and correlation of logs with your other devices. NSS and LSS can be configured from the Cloud Connector Portal.



NSS and LSS require separate subscriptions for each virtual machine.

- Learn more at [Understanding Nanolog Streaming Service](https://help.zscaler.com/zia/about-nanolog-streaming-service) (<https://help.zscaler.com/zia/about-nanolog-streaming-service>).
- Learn more at [About the Log Streaming Service](https://help.zscaler.com/zpa/about-log-streaming-service) (<https://help.zscaler.com/zpa/about-log-streaming-service>).

## Deploying Cloud Connector via Scripts

Cloud Connector can be deployed in AWS, leveraging scripts to simplify deployments and ensure consistency. Cloud Connector can be deployed directly from the AWS Marketplace, through CloudFormation scripts or via Terraform. While supported, deploying Cloud Connector manually is not best practice outside of lab environments. Zscaler recommends using scripting to provide consistent deployments.

CloudFormation scripts require the user to preconfigure several items prior to deployment, but are quite user-friendly and work well with brownfield deployments. CloudFormation scripts in AWS are more “native” to their respective platforms, however the preferred method for deploying the appliances is via Terraform.

Terraform is the most flexible option. Its goal is to be as “hands-off” as possible by automatically configuring items without user intervention. However, Terraform is more complex in its initial setup. Both options allow you to automate your deployment and achieve the same results.

More information is available on AWS deployment scripts at:

- AWS CloudFormation (<https://aws.amazon.com/cloudformation/>)
- Terraform scripts (<https://www.terraform.io/intro>)

## Deploying Cloud Connector via Terraform

Zscaler Terraform scripts provide complete end-to-end automation to not only deploy the Cloud Connector appliances, but all the secondary and tertiary components as well (in a best practice configuration).

Terraform scripts can be downloaded from the Cloud Connector Portal in four versions:

- Starter Deployment Template – Deploy Cloud Connector, VPCs, route tables, subnets, NAT gateway, internet gateway, etc. for use cases where only ZIA is required. In addition, Terraform also creates a t2.micro EC2 instance for use as a Management/Bastion host in the VPC that Cloud Connector is deployed in. This host is not a requirement long term, but is recommended for easier troubleshooting and testing.
- Starter Deployment Template with ZPA – Deploy Cloud Connector, VPCs, route tables, subnets, NAT gateway, internet gateway, etc. for use cases where ZIA and ZPA is the requirement. This script also includes Route 53 resources for outbound resolution and redirection to the ZPA service. In addition, Terraform also creates a t2.micro EC2 instance for use as a Management/Bastion host in the VPC that Cloud Connector is deployed in. This host is not a requirement long term, but is recommended for easier troubleshooting and testing.



For inbound access, App Connector must be installed as part of a separate workflow.

- Starter Deployment Template with High Availability – Deploy Cloud Connector in high availability mode (currently using Lambda function), along with required AWS constructs mentioned previously for use cases where ZIA and ZPA is the requirement. In addition, Terraform also creates a t2.micro EC2 instance for use as a Management/Bastion host in the VPC that Cloud Connector is deployed in. This host is not a requirement long term, but is recommended for easier troubleshooting and testing.



For inbound access, App Connector must be installed as part of a separate workflow.

- Starter Deployment Template with ZPA and High Availability – Deploy Cloud Connector in high availability mode (currently using Lambda function), along with required AWS constructs mentioned previously for use cases where ZIA and ZPA is the requirement. This script also includes Route 53 resources for outbound resolution and redirection to the ZPA service. In addition, Terraform also creates a t2.micro EC2 instance for use as a Management/Bastion host in the VPC that Cloud Connector is deployed in. This host is not a requirement long term, but is recommended for easier troubleshooting and testing.



For inbound access, App Connector must be installed as part of a separate workflow.

It is important to note that Terraform does not modify brownfield deployments. When executing Terraform scripts, new VPCs, route tables, subnets, and EC2 instances are spawned to support the current workflow. It is your responsibility to integrate the new deployment into your existing environment. This might mean that the new Cloud Connector VPC is peered with an existing Transit Gateway, or that new workloads are installed within the Cloud Connector VPC. Bear this in mind when considering whether Terraform is the correct option to use when integrating with a brownfield environment.

For detailed deployment instructions and to find the templates listed above, see [About Cloud Automation Scripts](https://help.zscaler.com/cloud-connector/about-cloud-automation-scripts) (<https://help.zscaler.com/cloud-connector/about-cloud-automation-scripts>).



## Deploying Cloud Connector via CloudFormation

If you are seeking a more native automation option for deploying Cloud Connector, Zscaler offers CloudFormation scripts. Though CloudFormation scripts can be used in greenfield situations, their value shines when you are deploying in a brownfield deployment, since many of the prerequisites are already satisfied if you have an existing AWS buildout. These YAML scripts can be downloaded from the Cloud Connector Portal in three versions:

- **Starter Deployment Template** – Deploy Cloud Connector and appropriate ENIs, and associate the appliance to the VPC, subnet, and route table specified in the workflow. This script is a requirement to run any of the other CloudFormation scripts.
- **Add-on Template with ZPA** – Deploy Route 53 resources for outbound resolution and redirection to the ZPA service for use cases where ZPA is the requirement.



For inbound access, App Connector must be installed as part of a separate workflow.



You must run the Starter Deployment Template prior to this script.

- **Add-on Template with High Availability** – Deploy a Lambda functionality for high availability. This script assumes that a pair of Cloud Connector instances already exists (with associated subnets, route tables, and availability zones). This script package also contains a Python script used to provide the HA probing and intelligence functionality to Lambda. This script must be hosted in an AWS-accessible repository (such as an S3 bucket).



You must run the Starter Deployment Template twice prior to this script.

It should be noted that although CloudFormation scripts work well with brownfield deployments, it is still your responsibility to integrate them into the environment. For detailed deployment instructions and to find the templates listed above, see [About Cloud Automation Scripts](https://help.zscaler.com/cloud-connector/about-cloud-automation-scripts) (<https://help.zscaler.com/cloud-connector/about-cloud-automation-scripts>).

## Upgrading Your Cloud Connectors

Cloud Connector runs the Zscaler OS in the virtual machine. Software updates and OS updates are provided by Zscaler via automatic upgrades. When a Cloud Connector is deployed, the software is automatically upgraded to the latest version. Cloud Connector instances check for new software daily. If a new version is available, the Cloud Connector upgrades itself automatically at midnight local time, based on the deployed cloud region.

This automatic check and update means that it is critical that your Cloud Connector locations are accurate. An inaccurate location can lead to upgrades occurring in the middle of the day. Always specify exactly where the Cloud Connector is located when deploying the virtual machine.

As a matter of redundancy during upgrades, Cloud Connector is installed in pairs within an availability zone. Multiple pairs of Cloud Connectors should be instantiated within different availability zones, thereby minimizing the impact of service upgrades or infrastructure failures.

Cloud Connector is based on the Zscaler OS, and therefore the software updates and OS updates are provided and automatically applied by Zscaler. When a Cloud Connector is deployed, the software is automatically updated to the latest version. A Cloud Connector then checks for new software daily and upgrades itself automatically at midnight local time, based on the deployed cloud region.

You can configure this upgrade window from the Cloud Connector Admin Portal. As mentioned throughout this document, Zscaler recommends that Cloud Connector appliances be deployed as redundant, high-availability instances. Specifically, we recommend deploying two appliances per availability zone with a minimum regional cluster size of four (two in AZ1 and two in AZ2). The Zscaler software upgrade process upgrades one instance of a pair at a time, providing availability for the AZ from the remaining instance.

Specific to software upgrades performed by Zscaler, this ensures that you incur no downtime. After an appliance is rebooted to accept a new update, Azure Load Balancer automatically moves traffic over to the redundant, active appliance.

Although cloud IaaS providers such as AWS are responsible for ensuring the security and availability of their infrastructure, organizations are ultimately still responsible for the security of their workloads, applications, and data. Learn more about the [Shared Responsibility Model](https://aws.amazon.com/compliance/shared-responsibility-model) (<https://aws.amazon.com/compliance/shared-responsibility-model>).

### Directing Traffic to Cloud Connector

Cloud Connector acts as a gateway to cloud workloads. Directing traffic through Cloud Connector is as simple as modifying the default gateway route of the transit gateway or workload VPC route table to point to the appliance. In most circumstances, this ensures that both internet-bound traffic destined for ZIA, and DNS traffic that requires modification for [ZPA use cases](#) where redirection to an App Connector is necessary, are appropriately handled.

For example, with a single instance of Cloud Connector, the workload route table can be updated with a default route using the Elastic Network Interface (ENI) of the service interface of the Cloud Connector appliance as the target. The Cloud Connector appliance uses the service subnet and route table created during the deployment process. The default route for this route table should point towards the NAT gateway, also created in the deployment process. A public subnet and route table should have also been created during the deployment process and reference the corresponding internet gateway with its default route.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

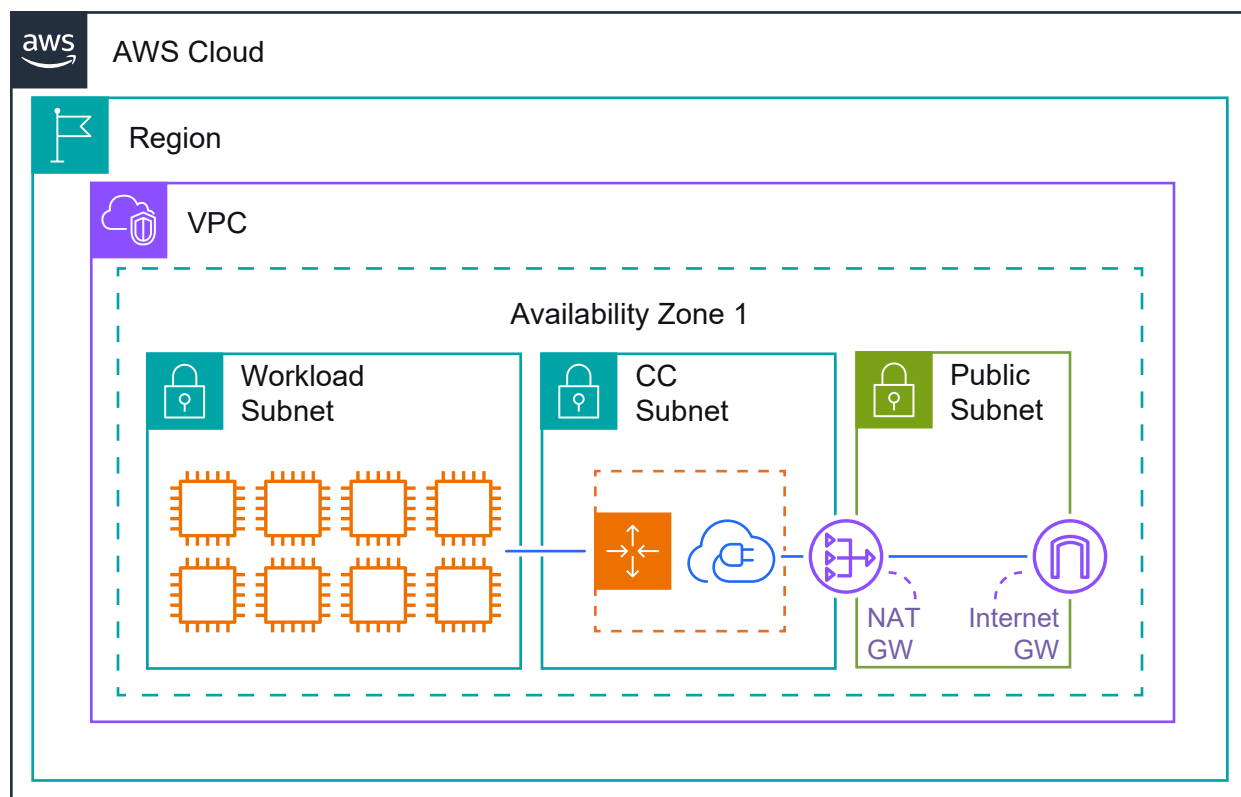


Figure 16: Route table updates from workload to internet

Network	Next Hop
Workload subnet	Cloud Connector interface
CC subnet	NAT gateway
Public subnet	Internet gateway

In the case of Transit Gateway (TGW), also known as a Transit or Egress VPC, you need to modify both the Transit Gateway route table and the route table for the VPC that houses the Cloud Connectors. This assumes that workload VPCs already use the TGW as their default gateway for unknown destinations.

In this case, the TGW route table should be peered and associated with all workload VPCs, as well as the Transit/Egress VPC. The TGW should then have a default route in its route table directing traffic towards the Transit/Egress VPC on its private subnet. The Transit VPC then follows a similar configuration to the previous example: a default route inside the TGW attachment subnet directs traffic towards the ENI of the Cloud Connector appliance.

Next, a default route in the service subnet of the Cloud Connector appliance directs traffic towards the NAT gateway. The public subnet servicing the NAT gateway then receives a default route directing traffic towards the internet gateway. As with all network traffic, ensure you have routing set up as well so that returning traffic from the internet is correctly directed back towards the Transit Gateway.

## Forwarding Options

When traffic has reached the Cloud Connector, there are four Traffic Forwarding options available to direct traffic out of the AWS cloud:

- **Direct** – Traffic matching the criteria defined bypasses the Cloud Connector and is routed out of the service interface, where it follows AWS route tables towards the destination.
- **Zscaler Internet Access (ZIA)** – Traffic matching the criteria defined is forwarded to the ZIA cloud for inspection.
- **Zscaler Private Access (ZPA)** – Traffic matching the criteria defined is forwarded to the ZPA cloud for inspection.
- **Drop** – Traffic matching the criteria is dropped by the Cloud Connector.

Each of the four options permits the administrator to define a range of match criteria. In general, macro forwarding logic can be defined within the Cloud Connector Portal, whereas ZIA or ZPA can perform more granular inspection.

Traffic Forwarding policy is in the Policy Management section of the Cloud Connector Admin Portal. Rule creation and assessment model ZIA and ZPA workflows. More specific rules should be ordered near the top, while more broad rules ordered towards the bottom. Match criteria is as follows:

### General

- **Location** – Locations identify the various VPCs from which your workloads send traffic. As Cloud Connector appliances are brought online, the VPC they are installed within automatically populates this menu. It should be noted, however, that in a Transit/Egress VPC scenario, downstream VPCs do not automatically populate. In such a case, you must use Source or Destination FQDN (recommended) or IP as match criteria. In ZIA, if the traffic is from a known location, the service processes the traffic based on the location settings. For example, the service checks whether the location has authentication enabled and proceeds accordingly. It also applies any location policies that you configure and logs internet activity by location.
- **Location Group** – If necessary, location groups can be created to organize various cloud VPCs, such as a “Dev VPCs” location group, “Prod VPCs” location group, etc. If there are many locations and associated sub-locations within your organization, consider using location groups.
- **Branch and Cloud Connector Groups** – Branch and Cloud Connector groups allow you to match traffic transiting specific Cloud Connector appliances.

### Source

- **Source IP Groups** – When multiple source IP addresses must be matched across multiple policy rules, it is operationally more efficient to create source IP groups. These groups allow you to organize IP addresses for easier rule creation and visualization.
- **Source IP Addresses** – This match criteria allows you to specify the source IP address of the workload.

### Destination

- **Destination FQDN/IP** – For individual FQDN (recommended) or IP address matching, enter the value you want to be matched in this field.

- **Destination FQDN/IP Group** – You can group together destination FQDNs (recommended) and the IP address that you want to control in a Forwarding Policy rule by specifying FQDN, IP addresses, countries where servers are located, and URL categories.



Wildcard domain identifiers ("\*") are not currently supported.

- **Destination Country** – This match criteria allows you to specify the destination country of the remote machine.



Destination criteria is not supported when ZPA is selected as the Forwarding Method.

After configuring a Forwarding Method and match criteria, you must choose an action. By default for ZIA use cases, the Cloud Connector appliance uses geolocation to locate a ZIA Public Service Edge in geographic proximity to the appliance. Alternatively, you can manually specify which Public Service Edge to use by configuring a gateway under the Forwarding Methods section of the Administration menu. Zscaler recommends using geolocation where possible.



Gateway selection criteria is not supported when ZPA is selected as the Forwarding Method. Cloud Connector automatically selects a broker.

Lastly, specifically for ZPA use cases, Cloud Connector also allows for the filtering of DNS requests and responses. In the Administration menu within DNS Control, administrators can add additional rules to permit or deny specific DNS requests from workload segments. More importantly, this functionality can be used to determine which traffic gets consumed by ZPA, and therefore which synthetic IP pool is used to address traffic within Microtunnels.

To view configuration instructions, see [Configuring Traffic Forwarding Rules \(https://help.zscaler.com/cloud-branch-connector/configuring-traffic-forwarding-rule\)](https://help.zscaler.com/cloud-branch-connector/configuring-traffic-forwarding-rule).

## Choosing the Correct Design Model

Cloud Connector is extremely flexible in how it can be deployed: directly adjacent to the workloads it services, or in a dedicated island by itself wherein traffic can be directed through it via AWS networking constructs like Transit Gateway. There is no single design model that fits every environment. Many organizations pull elements from all design models to suit their goals. There are three main questions to ask when determining how best to get started:

### Is ZPA a requirement?

ZPA requires workload DNS queries to transit the Cloud Connector so a synthetic IP address can be assigned to the connection. Consider how DNS is employed within the cloud. If using cloud-hosted DNS servers, it is possible that DNS resolution requests are never directed across the Cloud Connector which would break ZPA. For this reason, automation scripts implement AWS Route 53 to intercept and redirect DNS traffic. Route 53 is not required, however consider how DNS resolution requests inherently transit Cloud Connector, such as if a public DNS server outside of the cloud is used. Additionally if this cloud implementation also services inbound requests from remote clouds, consider pointing App Connectors towards real DNS servers in this scenario.

### **Is high availability a requirement?**

Zscaler recommends that high availability be employed in all use cases. However when deploying directly into the workload VPC, compute costs can quickly spiral out of control. For this reason, you might consider using dedicated Transit/Egress VPCs peered with a Transit Gateway. This allows you to maintain high availability, without a large compute footprint. This model likely requires that functions like AWS Transit Gateway are implemented.

### **Will Cloud Connector be deployed within the workload VPC, or in a dedicated VPC?**

For small environments with only a handful of VPCs, Cloud Connector instances can be deployed directly within the workload VPC. However, the number of VPCs and EC2 instances tend to increase as an organization grows larger and invests further in the cloud. As new VPCs are added, they will require new appliances. As you consider where the Cloud Connector appliances will be installed, ensure you plan for adequate growth in the number of workloads and VPCs that Cloud Connector will protect. If the future state of the environment becomes operationally cumbersome, or if the environment already contains several VPCs, it might be best to consider a Transit Gateway approach with a dedicated Transit/Egress VPC for Cloud Connector.

### **Use Case: Direct to Internet Using Zscaler Internet Access**

Implementing Cloud Connector to provide outbound internet access through ZIA is one of the first steps to cloud workload protection. The following deployment model represents a recommended option that can be leveraged to satisfy this business requirement and offer a foundation to build on when looking to implement services like ZPA.

In this model, Cloud Connectors can be installed directly into the workload VPC adjacent to the individual workloads they service. As with all deployment models, Zscaler highly recommends deploying Cloud Connector in high availability.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

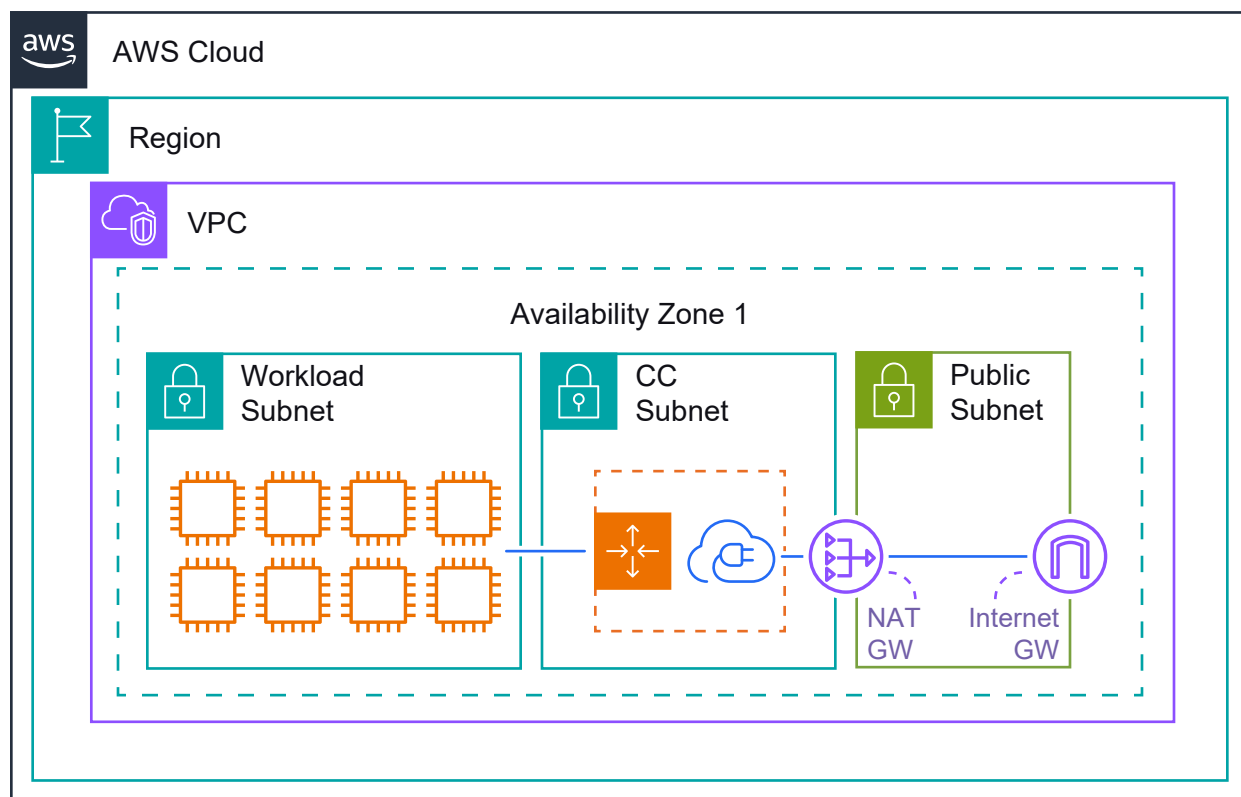


Figure 17: Redundant workload model

The primary benefit to this design option is its simplicity and time to implement. Since each Cloud Connector instance is spawned within the workload VPC that it services, routing is made simple. Likewise, whether via Terraform or CloudFormation, Zscaler automation can implement this model in a matter of minutes. From a cost perspective, you are only paying for egressing data fees one time (as the workload traffic leaves the Cloud Connector).

If you have many workload VPCs, however, this design option can be cumbersome. Any cost savings associated with egress fees can be eliminated by the increased compute footprint, since separate Cloud Connector EC2 instances are required per workload VPC. Additionally, this option requires the modification of many route tables to direct traffic accordingly, which is further complicated when high availability enters the picture.

When deploying this option via AWS CloudFormation, Cloud Connector can be implemented within existing brownfield VPCs. You must create new subnets (ideally associated with separate availability zones) and route tables to apply to the EC2 instances being inserted into the VPC. Conversely, when deploying this option via Terraform, the Cloud Connector instance is placed into a programmatically created VPC with new subnets and route tables.



For brownfield implementations, CloudFormation can provide more seamless integration. For greenfield implementations, consider using Terraform.

### Use Case: Integrating with AWS Transit Gateway

Cloud Connector can also be placed in a dedicated VPC where outbound workload traffic is first directed through a centralized hub, such as Transit Gateway. Transit Gateway is a design option that is growing in adoption as organizations seek to address scalability concerns and operational deficiencies imposed by legacy inter-VPC networking.

This model closely resembles a traditional hub-and-spoke network since the Transit/Egress VPC, where Cloud Connector operates, receives traffic from many workloads spoke VPCs through a hub Transit Gateway. As with all deployment models, Zscaler highly recommends deploying Cloud Connector in high availability. As workload traffic enters the Transit Gateway from one availability zone, TGW attempts to direct that traffic to a Cloud Connector appliance that exists in the same availability zone. This lowers cost and latency while also providing a mechanism for leveraging all available Cloud Connector appliances in an Active/Active fashion. Cloud Connector appliances provide high availability for one another, while simultaneously servicing traffic from their own availability zone.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.



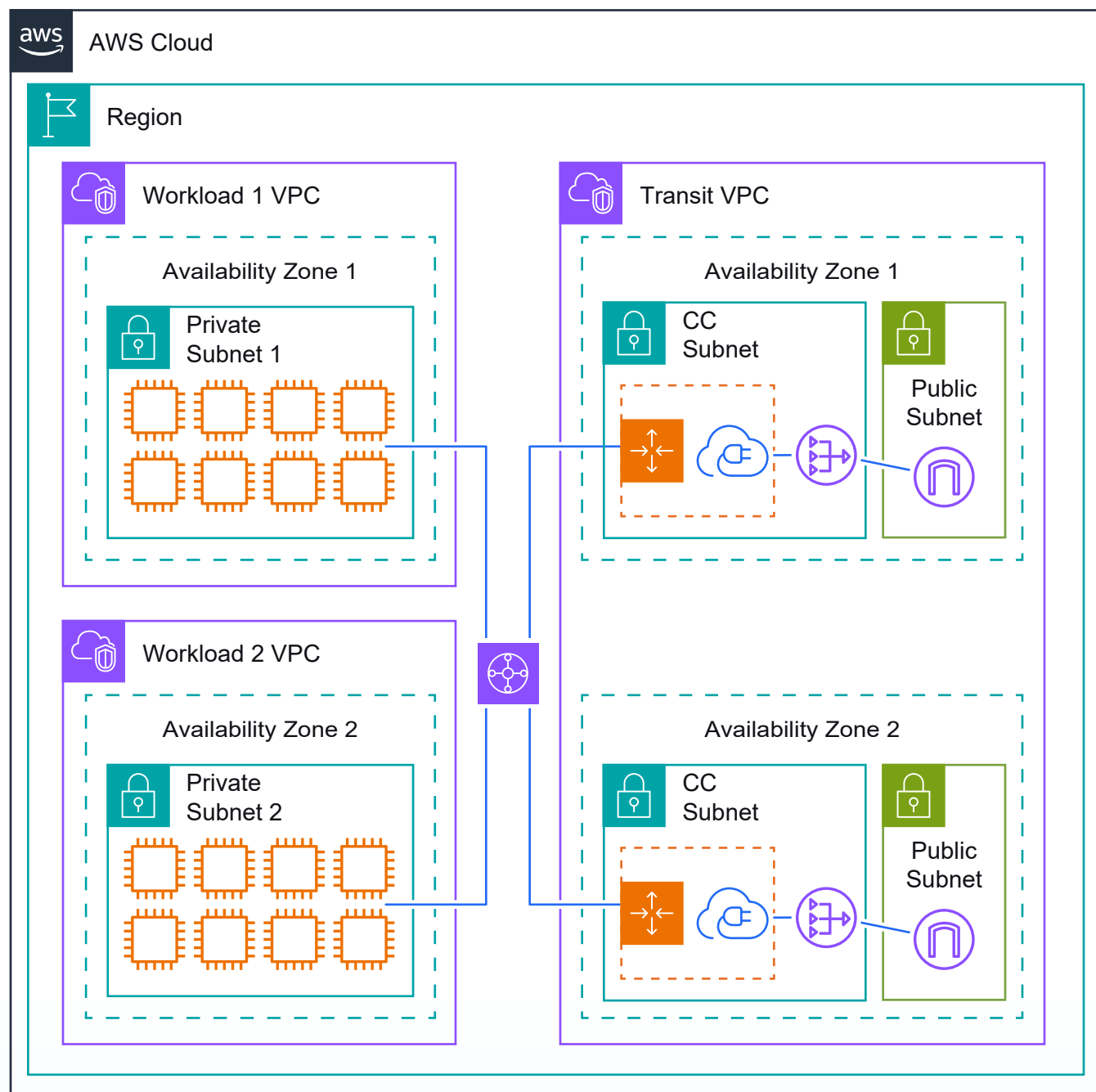


Figure 18: Deploying Cloud Connector using a Transit/Egress VPC with Transit Gateway

Deploying Cloud Connector using a Transit/Egress VPC with Transit Gateway allows the organization to simplify cloud routing and reduce the compute footprint required when deploying directly to the workload VPCs. In this option, only a single pair of Cloud Connector appliances is necessary for a Transit/Egress VPC. Spoke VPC workloads requiring internet or private access are directed towards the Transit Gateway using a simple default route, where they can then be directed towards the Cloud Connector appliances for outbound routing.



By default, Terraform installs Cloud Connector using a Transit/Egress VPC model, though it can be customized to suit an organization's deployment needs.

AWS recommends using a Transit Gateway in conjunction with a Transit/Egress VPC for ZIA and ZPA use cases. For ZPA-only use cases, VPC Peering without a Transit Gateway is not recommended. Depending on the size of the environment, Transit Gateway can incur additional costs that could eliminate the cost savings from a reduced compute footprint.

When deploying this option via AWS CloudFormation, you must create the Transit/Egress VPC, subnets, and route tables. When deploying this option via Terraform, the Cloud Connector instance is placed into a programmatically created VPC with new subnets and route tables. For this option, consider using Terraform as the path of automation whether you are integrating with a greenfield or brownfield environment.

### Use Case: Distributed Gateway Load-Balancing Endpoints

Another model for centralized deployment of Cloud Connector instances leverages the AWS Gateway Load Balancer (GWLB). Like the Transit Gateway model, this model closely resembles a traditional hub-and-spoke network.

The GWLB is deployed centrally with your Zscaler Cloud Connectors instances. Traffic received from workloads is load balanced across available Cloud Connector instances. The GWLB works in conjunction with instances of the AWS Gateway Load Balancer Endpoints (GWLBs), which are deployed in the workload subnets.

The model works by having your workload routing tables updated to point to the GWLBs as their default gateway. The GWLBs in turn forwards traffic to the central GWLB for distribution to the Cloud Connector instances. Distributed GWLB is a design option that, like Transit Gateway, is growing in adoption as organizations seek to employ GWLB's benefits in a scalable architecture without adding additional complexity.

GWLB, by default, attempts to maintain traffic affinity across availability zones. As workload traffic enters the GWLB from one availability zone, the GWLB attempts to direct that traffic to a Cloud Connector that exists in the same availability zone.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

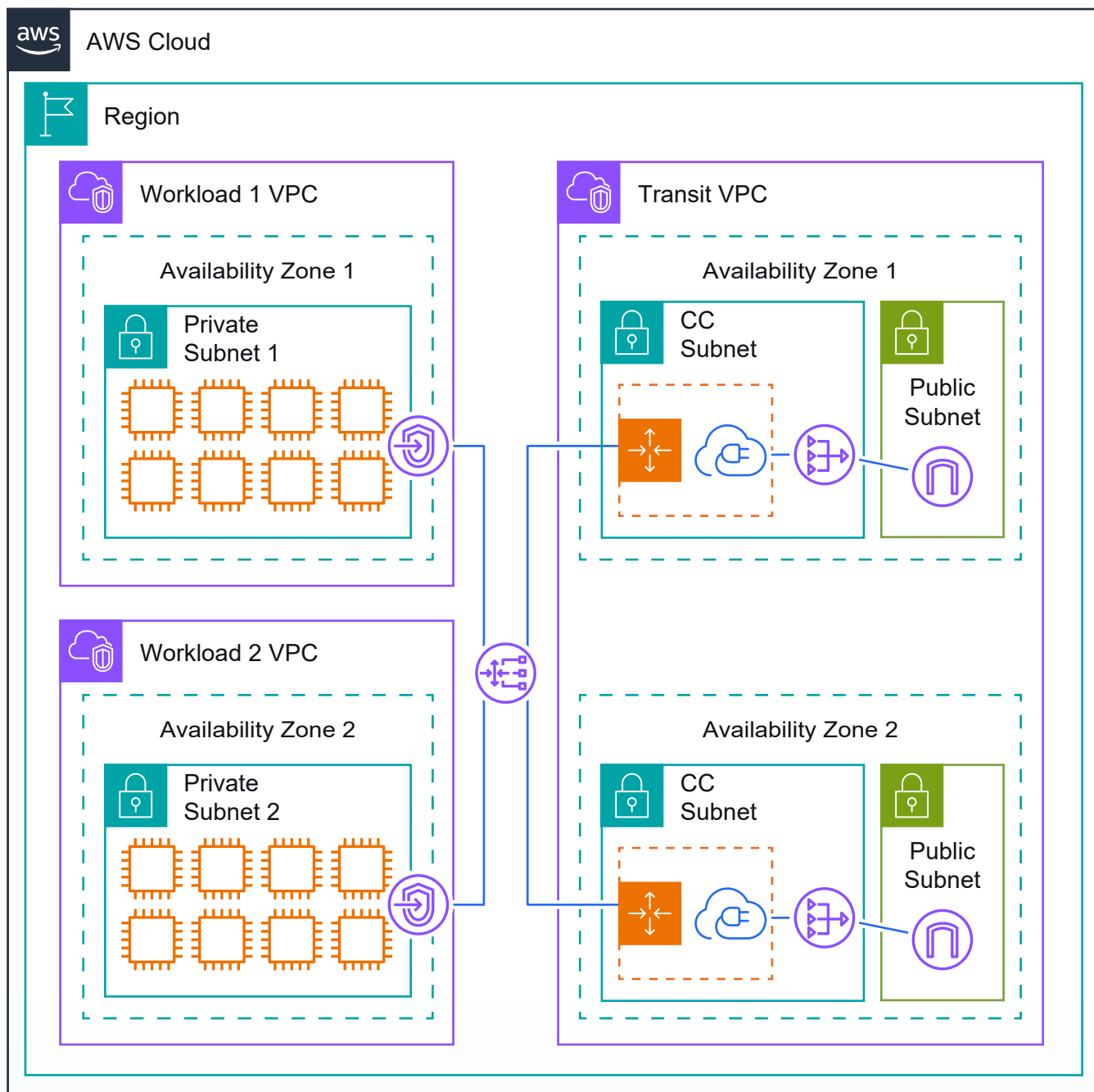


Figure 19: Workload traffic is balanced across Cloud Connector instances by the AWS Gateway Load Balancer

Deploying Cloud Connector using a Transit/Egress VPC with distributed GWLB instances provides two benefits to your organization. First, routing is greatly simplified. Second, it reduces the compute footprint versus deploying Cloud Connector directly in the workload VPCs. In this option, only a single pair of Cloud Connector appliances is necessary for a Transit/Egress VPC.



Terraform, by default, installs Cloud Connector using a Transit/Egress VPC model, though it can be customized to suit an organization's deployment needs.

If your organization requires VPC-to-VPC communication within a region, you need to consider this traffic pattern. While VPC Peering and Transit Gateway are options to allow connectivity between VPCs, they are inherently less secure since traffic flowing over VPC Peering or Transit Gateways does not flow through the Cloud Connector and is not inspected by the Zero Trust Exchange. For this reason, consider ZPA as a mechanism to facilitate this communication between workloads. See [Use Case: Integrating Zscaler Private Access](#) in this guide for more details.



By default, Cloud Connector offloads ZPA traffic to the Zero Trust Exchange. This can incur additional latency and egress costs. If you have a large amount of traffic that will transit between workloads, consider a ZPA Private Service Edge for your deployment. You can learn more about the ZPA Private Service Edge in [Universal ZTNA with Zscaler Private Access Private Service Edge](https://www.zscaler.com/resources/reference-architectures/universal-ztna-zpa-private-service-edge.pdf) (<https://www.zscaler.com/resources/reference-architectures/universal-ztna-zpa-private-service-edge.pdf>).

When implementing this design option, the first step is to consider which automation technique to employ. When deploying this option via AWS CloudFormation, you are required to create the Transit/Egress VPC as well as subnets and route tables to deploy. When deploying this option via Terraform, the Cloud Connector instance is placed into a programmatically created VPC with new subnets and route tables. Consider using Terraform as the path of automation whether you are integrating with a greenfield or brownfield environment for this use case.

If using CloudFormation scripts, consider deploying a second Cloud Connector appliance within the Transit VPC in a separate availability zone. This can be done by simply re-running the CloudFormation script run initially. When complete, run the high availability script found in the Cloud Connector Admin Portal. This will instantiate GWLB and the appropriate constructs for liveness detection of the new appliances. Note that this is not required if using the Terraform high availability script, as it will automatically spawn two appliances and GWLB functionality as part of a single script execution.

If using CloudFormation scripts, it is recommended to deploy NAT gateway with internet gateway as discussed in the [CloudFormation section](#). Since NAT gateways operate within a single availability zone, Zscaler recommends creating a second NAT gateway in a different AZ so that an infrastructure failure of one AZ does not affect both NAT gateways. After the second NAT gateway is created, you need to revisit the public route table associated with the secondary Cloud Connector to point the default route towards the new NAT gateway. The route table associated with the new NAT gateway must also have a default route directed towards the internet gateway attached to the VPC.

CloudFormation Templates automatically create the Gateway Load Balancer and Gateway Load Balancer Endpoints for you. However, the GWLB endpoints created exist within the Transit/Egress VPC. You must manually install additional endpoints in your Spoke VPCs using the documentation provided by AWS. At present, neither CloudFormation nor Terraform scripts handle this task. Learn more about [creating a Gateway Load Balancer Endpoint](https://docs.aws.amazon.com/vpc/latest/privatelink/gateway-load-balancer-endpoints.html#create-gateway-load-balancer-endpoint) (<https://docs.aws.amazon.com/vpc/latest/privatelink/gateway-load-balancer-endpoints.html#create-gateway-load-balancer-endpoint>).

Learn more about [Gateway Load Balancers and Gateway Load Balancers Endpoints](https://docs.aws.amazon.com/vpc/latest/privatelink/vpce-gateway-load-balancer.html) (<https://docs.aws.amazon.com/vpc/latest/privatelink/vpce-gateway-load-balancer.html>).

## Use Case: Integrating Zscaler Private Access

Assuming that Cloud Connector has been deployed and traffic directed through it, we can now add support for ZPA. This use case is growing in popularity as organizations seek to depart from legacy VPN technologies to interconnect cloud and on-premises workloads. An important consideration with Cloud Connector is that it is designed to facilitate outbound workload traffic towards a remote destination. When the destination is in a location you control, we must consider how this traffic ingresses into the remote facility. We do this using the Zscaler App Connector appliance, where App Connector VMs sit adjacent to the workloads they provide access to.

This model builds on the foundation provided in the direct-to-internet and Transit Gateway use cases discussed previously. Cloud Connector provides outbound connectivity for cloud workloads to an on-premises data center, which uses App Connector VM appliances sitting in an application server segment to provide inbound connectivity. Both appliances build DTLS tunnels to the ZPA Broker and establish a Microtunnel between the source (cloud) workload and the destination data center workload. The traffic within the Microtunnel targets synthetic proxy IP addresses inside the Cloud Connector and App Connector, respectively.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

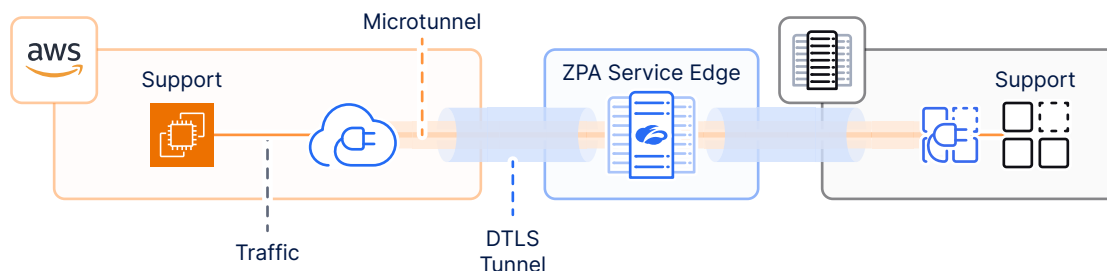


Figure 20: Cloud and on-premises workloads meet at the ZPA Service Edge

All communication between ZPA components travel within a client and server certificate-verified TLS connection. Within this TLS-encrypted Zscaler Tunnel, a microtunneling protocol exists. Select components of ZPA run through this encrypted Microtunnel end to end. Because the client and server use certificates issued by Zscaler, it is cryptographically impossible for ZPA to experience a Man-in-the-Middle (MITM) attack. The client certificates are verified against an organization's Certificate Authority (CA), and the server certificates are verified against Zscaler's CA which cannot be spoofed by any third-party compromised CA.

ZPA only accepts connections from the Zscaler Cloud Connector and the App Connector instances that present a client certificate signed by a CA associated with each tenant. Zscaler Cloud Connector and App Connector only connect to ZPA service components that present a certificate signed by the ZPA infrastructure PKI.

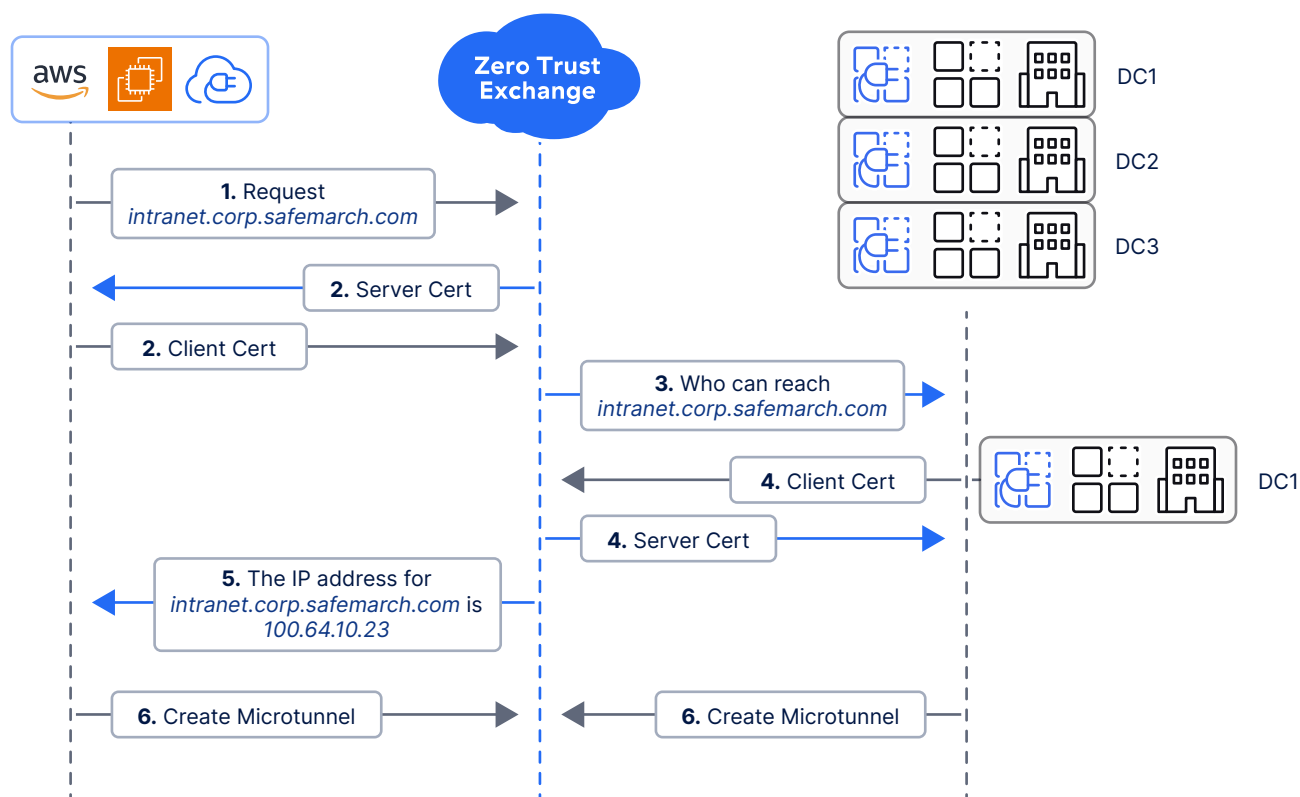


Figure 21: Authentication and tunnel setup between workloads and internal apps

1. A workload requests access to an application.
2. The ZPA Service Edge and Zscaler Cloud Connector authenticate via certificate exchange.
3. If the workload is authorized to access the requested application, the ZPA Service Edge determines which App Connector can service the request.
4. The ZPA Service Edge and Zscaler App Connector authenticate via certificate exchange.
5. The workload is presented with the synthetic IP address of the application.
6. A Microtunnel is established between the Zscaler Cloud Connector and Zscaler App Connector.

Zscaler Cloud Connector recognizes the internal applications that are available via ZPA. Access to these applications is defined by ZPA based on policies. Using information received from the ZPA Public Service Edge or ZPA Private Service Edge, Cloud Connector intercepts workload requests for applications and then forwards those requests to the ZPA cloud.

No network information is required to access available applications. To facilitate secure private connections that are abstracted from the physical network, Cloud Connector associates permitted internal applications with a set of synthetic IP addresses. When a workload sends out a DNS request, Zscaler Cloud Connector can recognize the domain as an internal application being protected by ZPA. Zscaler Cloud Connector then intercepts the DNS request and delivers a DNS response to the workload that uses the synthetic IP address associated with the internal application.

To intercept and modify DNS requests, Cloud Connector must “see” the initial request from the cloud workload. To facilitate this, Zscaler recommends adding AWS Route 53 support. By default, cloud workloads leverage AWS DNS. However, this traffic never crosses the Cloud Connector and can break ZPA. By running any of the following scripts, a new DNS resolver endpoint will be installed within the workload VPC:

- Starter Deployment Template with ZPA
- Starter Deployment Template with ZPA and High Availability
- Add-on Template with ZPA scripts

For every domain or DNS A Record configured, Route 53 ensures that the resolution request is first sent to the DNS resolver endpoint. The DNS Route 53 endpoint then redirects the resolution request to a public (or internal) DNS server, where the request crosses Cloud Connector.

- Note that wildcard domain identifiers are supported, but it is recommended that you use specific DNS A Records for applications.
- You must revisit AWS Route 53 configuration to add new or additional domains and DNS A Records.
- AWS Route 53 is only required when internal AWS DNS servers are used, where the DNS request does not traverse the Cloud Connector appliance. If using public DNS servers, Route 53 can safely be omitted.

### Use Case: Securing Traffic Between Clouds

Multi-cloud deployments, where workloads are spread across more than one cloud provider, are becoming more common as organizations look to provide hosting across more than one vendor. You can choose to host your cloud workloads in more than one cloud or, for redundancy or geoproximity, in multiple regions of the same cloud service provider. This use case focuses on how to solve for the challenges faced in this scenario and how we can secure this traffic using the ZPA model discussed [previously](#).

Using the ZPA model as a basis, this model builds on the fact that remote application destinations secured by ZPA might not be in an on-premises data center. Instead, these applications exist within a different cloud region or in a different cloud service provider altogether. As originating cloud workloads send requests to remote applications, Cloud Connector routes them to the appropriate App Connectors in the destination cloud.

The following image is simplified for clarity. Redundant instances of Zscaler Cloud Connector should be deployed in all instances.

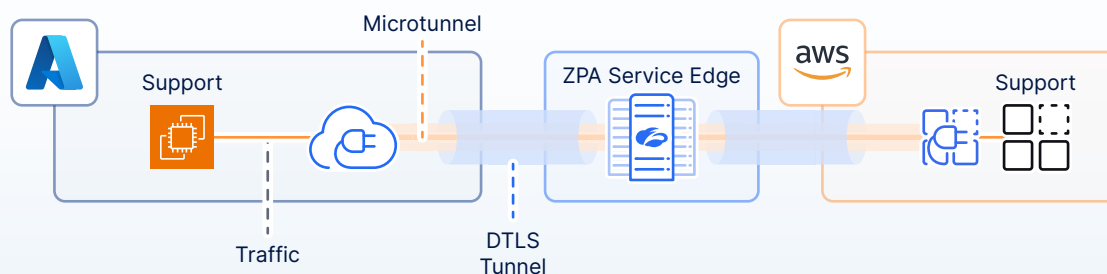


Figure 22: Workload-to-workload communication across cloud providers

Communication between two cloud workloads via ZPA mirrors the illustrations described in the [previous section](#), so they are not repeated here. However in the interest of discussing the underlay architecture, the previous figure depicts how App Connectors can be installed to facilitate this use case.

## Summary

Connecting workloads to the internet across different networks is difficult. What makes this harder is the traditional approach used by organizations to solve this challenge, such as using technologies like VPNs and firewalls. While the outcome of connecting these workloads is achieved, the cost to achieve these goals is significant:

- Risk of lateral threats and internet-based attacks by overextending the trusted network across the internet using VPN and WAN technologies.
- Complexity increases because of complicated route filtering, multiple network hops, and fragmented policy management.
- Poor visibility across application connectivity paths and increased network blind spots.
- Costs rise due to overprovisioning network services and the use of virtual appliances such as firewalls, IP addresses, routers, and other point products in cloud environments.
- Limited scale and performance from the increase in network and security services used in cloud environments.

As a result, there is a need for a better approach. Zscaler Cloud Connector is a cloud-native Zero Trust access service that provides fast and secure app-to-app, app-to-internet connectivity across multi-cloud environments. With integrated, automated connectivity and security, it reduces complexity and cost, and provides a faster, smarter, and more secure alternative to legacy network solutions.



## About Zscaler

Zscaler (NASDAQ: ZS) accelerates digital transformation so customers can be more agile, efficient, resilient, and secure. The Zscaler Zero Trust Exchange protects thousands of customers from cyberattacks and data loss by securely connecting users, devices, and applications in any location. Distributed across more than 150 data centers globally, the SASE-based Zero Trust Exchange is the world's largest inline cloud security platform.

©2025 Zscaler, Inc. All rights reserved. Zscaler, Zero Trust Exchange, Zscaler Private Access, ZPA, Zscaler Internet Access, ZIA, Zscaler Digital Experience, and ZDX are either (i) registered trademarks or service marks or (ii) trademarks or service marks of Zscaler, Inc. in the United States and/or other countries. Any other trademarks are the properties of their respective owners.

