Network Guardrails for AWS
Network Security Zones

Overview

The Networking Model consists of six core Network Security Zones (NZ). Each NZ is defined by one or more CIDR ranges. Listed below is a definition and assumed default range for each NZ; however, default ranges can be (are expected to be) modified based on the owning organization’s own network and policies. Example:

**Public**: Configured; CIDR ranges specifying the Internet or portions of the Internet that are considered public from the application’s context. Default:
- 0.0.0.0/0
- ::/0

**Intranet**: Configured; CIDR ranges specifying the boundaries of the organization’s intranet. Default:
- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

**Trusted**: Dynamically calculated; CIDR ranges consisting of the union between the *Interior* network (see below) and *Interior* network of any trusted VPCs
**Interior**: Dynamically calculated; CIDRs of all individual subnets in the current VPC. External subnets are explicitly excluded.

**Bastion**: Configured; CIDR ranges specifying the boundaries of networks with trusted bastion hosts. Please note that although the associated diagram shows a typical bastion configuration inside the organization’s intranet, bastion NZs could be configured to exist outside the address range of the Intranet NZ or even the Public NZ. Default: None.

**Pair**: Dynamically managed by AWS; Multiple /32 CIDR ranges (Individual IP addresses) of all resources sharing the same Security Group.

**Policies – General**

Network Zones in Turbot have sensible defaults, but are configurable to match enterprise network configurations through Policies:

- Turbot > Network > CIDR Ranges – Internet
- Turbot > Network > CIDR Ranges – Intranet
- Turbot > Network > CIDR Ranges – Bastion
- Turbot > Network > CIDR Ranges – Docker Bridge

**Policies – Advanced VPC**

VPC Policies reference the General NZ Policies by default, but can also be customized for specific VPC resource configuration:

- AWS > VPC > EGW CIDR Ranges
- AWS > VPC > IGW CIDR Ranges
- AWS > VPC > NAT Gateway CIDR Ranges
- AWS > VPC > VGW Internet CIDR Ranges
- AWS > VPC > VGW Intranet CIDR Ranges

Changes to these advanced Policies are rare and should be limited to rare, highly custom requirements.

**Policies – Advanced Security Groups**

Security Group Policies reference the General NZ Policies by default, but can also be customized for specific Security Group resource configuration:

- AWS > VPC > Security Group Bastion CIDR Ranges
- AWS > VPC > Security Group Internet CIDR Ranges
- AWS > VPC > Security Group Intranet CIDR Ranges

Changes to these advanced Policies are rare and should be limited to rare, highly custom requirements.
**Subnet Type & Routing**

While VPC is the high-level building block in AWS networking, Subnets control the fundamental aspects of physical location and connectivity across Network Zones.

**Regions, Availability Zones, Disaster Recovery and High Availability**

Amazon VPCs are available in multiple Regions worldwide (e.g. us-east-1, eu-west-1), each including multiple Availability Zones (e.g. us-east-1a, us-east-1b). Regions are geographically dispersed with a latency in the 10’s of milliseconds. Availability Zones are locally dispersed, with a latency of <1 millisecond.

Each Subnet is defined in a single Availability Zone. Best practice is to provide multiple Subnets, spread across multiple Availability Zones. Applications designed for High Availability will leverage these Subnets in a single Region as a group with real-time traffic and data synchronization across Availability Zones. Critical applications with the highest availability requirements may be specifically designed with a cross-Region architecture which is more tolerant to latency considerations.

**Subnet Types**

Servers, databases and other resources in the Network are specifically limited to the design and connectivity of the Subnet where they reside. Load Balancers might be publicly accessible, while databases are limited to internal access.

In Turbot, Subnet routing is calculated and continually enforced by guardrails based on the following core Subnet Types:

**Public**: Publicly accessible resources with no connectivity to internal resources. Typically used for standalone, static or serverless websites.

**DMZ**: Publicly accessible resources with limited connectivity to internal resources. Typically used to host Load Balancers for a public-facing web application in another Subnet.

**Independent**: Isolated resources with outbound Internet access. Typically used for large-scale HPC workloads that require IP space beyond internal allocations.
**Direct**: Internal resources with internal connectivity, direct outbound access to the Internet and AWS. Typically used for internal applications with high bandwidth or latency sensitive Internet access requirements.

**Limited**: Internal resources with internal connectivity, outbound Internet access via internal network, and direct access to AWS. Typically used for internal applications with high bandwidth or latency sensitive AWS access requirements.

**Private**: Internal resources with internal connectivity, access to the Internet and AWS via internal network. Typically used for internal enterprise applications with limited Internet access.

**Restricted**: Internal resources with internal connectivity, no access to the Internet or AWS. Typically used for sensitive internal enterprise applications where data leakage must be carefully controlled.

**Isolated**: Isolated resources with no connectivity outside their Subnet. Typically used for very sensitive, local workloads with very limited access from specific Bastion hosts.

**Routing**

As is likely clear from their definition, each Subnet Type is largely defined by its connectivity and routing to internal, external and AWS resources. Specifically:

<table>
<thead>
<tr>
<th>Subnet Type</th>
<th>Intranet Route</th>
<th>Internet Route</th>
<th>AWS Route</th>
<th>Route Table</th>
<th>IP Address Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>None</td>
<td>IGW</td>
<td>IGW</td>
<td>Per VPC</td>
<td>Public</td>
</tr>
<tr>
<td>DMZ</td>
<td>VGW</td>
<td>IGW</td>
<td>IGW</td>
<td>Per VPC</td>
<td>Public</td>
</tr>
<tr>
<td>Independent</td>
<td>None</td>
<td>NAT / EIGW</td>
<td>VPCe</td>
<td>Per AZ</td>
<td>Private</td>
</tr>
<tr>
<td>Direct</td>
<td>VGW</td>
<td>NAT / EIGW</td>
<td>VPCe</td>
<td>Per AZ</td>
<td>Private</td>
</tr>
<tr>
<td>Limited</td>
<td>VGW</td>
<td>VGW</td>
<td>VPCe</td>
<td>Per VPC</td>
<td>Private</td>
</tr>
<tr>
<td>Private</td>
<td>VGW</td>
<td>VGW</td>
<td>VGW</td>
<td>Per VPC</td>
<td>Private</td>
</tr>
<tr>
<td>Restricted</td>
<td>VGW</td>
<td>None</td>
<td>None</td>
<td>Per VPC</td>
<td>Private</td>
</tr>
<tr>
<td>Isolated</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Per VPC</td>
<td>Private</td>
</tr>
</tbody>
</table>

**Policies – Subnet Type**

Subnets have their Subnet Type set through Turbot Policy. This is the most critical Policy in the Network Guardrails and controls the entire design, configuration and
management of all resources surrounding the Subnet. Changing the Subnet Type may change routes, create resources such as IGWs and significantly alter then network configuration.

- VPC > Subnet Type

Policies – Availability

Turbot can be configured to require the use of multiple Availability Zones for Subnets in a VPC.

- VPC > VPC Availability Zones - External
- VPC > VPC Availability Zones - Internal

Subnet Purpose & Ports

Overview

While the Subnet Type defines connectivity of Subnets, the Subnet Purpose is used to define what types of workloads the Subnet will be hosting. For example, the Subnet Type determines the Subnet is a DMZ, while the Subnet Purpose determines the Subnet is suitable for Web workloads (not Databases). Subnet Type defines routing, Subnet Purpose defines ports.

Turbot has predefined the following Subnet Purposes:

**All**: Any workload type.

**Web**: Web applications or load balancers.

**DB**: Databases.

**App**: Application servers (e.g. Tomcat).

**Undefined**: Subnet has no defined purpose, sensible defaults for the Type will be used.

Policies – Subnet Purpose

In most cases, the Subnet Purpose does not need to be defined. By default, Turbot will use sensible defaults for the Subnet Purpose based on the Subnet Type – e.g. a DMZ Subnet has a Purpose of Web only (not DB or App).
But, for organizations that prefer to segregate workload types by Subnet, the Subnet Purpose can be explicitly set. For example, two Private Subnets in a VPC may be set with a Purpose of DB and App respectively – isolating the workload types into different Subnet tiers.

The Policy to control the list of Purposes for a Subnet is:

- VPC > Subnet Purpose

Policies – Advanced Purpose Definitions

If required, the default Subnet Purposes for each Subnet Type can be through Policies:

- VPC > DMZ Subnet Default Purposes
- VPC > Direct Subnet Default Purposes
- VPC > Independent Subnet Default Purposes
- VPC > Isolated Subnet Default Purposes
- VPC > Limited Subnet Default Purposes
- VPC > Private Subnet Default Purposes
- VPC > Public Subnet Default Purposes
- VPC > Restricted Subnet Default Purposes

Routing & Gateway Management

Overview

At the heart of every network is the Route Table. It has ultimate responsibility for connectivity and traffic flow between Subnets. For Turbot’s Network Guardrails, setup of the various Gateways, configuration of the Route Table and maintenance of Routes is the intersection of many higher level settings.

Turbot will automatically reuse, update and create IGWs, EGWs, VGWs, Endpoints and NAT Gateways to create the desired Subnet Type configuration. For example, if a Subnet is changed from Private to Public then Turbot will automatically create an IGW, attach it to the VPC and update Routes in the Route Table.

Policies – Route Configuration

Route manipulation is determined by multiple Policies, each of which manage certain parts of the Route Table based on the Policies for Subnet Type, Approved, etc. Each type of route can be skip, checked or enforced with:

- VPC > EGW Configuration
- VPC > Endpoint Configuration
• VPC > IGW Configuration
• VPC > NAT Gateway Configuration
• VPC > VGW Configuration

Instead of using Turbot to manage routes, route propagation via the VGW can be enabled. In this case, Turbot will only manage routes to the external Gateways above and will skip all route configuration to the VGW:
• VPC > VGW Propagation

Policies – High Availability Routing

NAT Gateways run in a specific Availability Zone, but are used to provide general Internet access. A single NAT Gateway can be used by Subnets in different Availability Zones, but with increased risk of failed access should the Availability Zone experience a failure. Best practice is to run a separate NAT Gateway in each Availability Zone – which Turbot will automatically create and enforce. But, in advanced cases, Turbot allows this to be customized (e.g. to reduce costs):
• VPC > NAT Gateway High Availability
• VPC > Internet Route High Availability

Policies – IGW

Unattached IGWs can be monitored and cleaned up with:
• VPC > IGW Attached

Policies – VGW

Unattached VGWs can be monitored and cleaned up with:
• VPC > VGW Attached

Policies – Gateway Endpoints

Direct access to AWS services may be available through a VPC Gateway Endpoint. As of Jan 2018, AWS uses this model for:
• S3
• DynamoDB

Turbot will manage Endpoints for a Subnet based on both the Subnet Type and type of Endpoint.

Advanced customization of the Subnet Types to be configured with Gateway Endpoints is available in:
• VPC > Route Table Endpoints Enabled
Advanced customization of the types of AWS services that will be configured for Gateway Endpoints by Turbot is available in:
- VPC > Endpoint Types Allowed

**Policies – NAT Gateway**

NAT Gateways have many possible configurations, particularly as they work across both Subnet Types (e.g. setup in Public, routed from Direct) and Availability Zones (e.g. single NAT Gateway used from multiple zones).

Unused NAT Gateways can be monitored and cleaned up with:
- VPC > NAT Gateway Used

Misconfiguration can result in multiple NAT Gateways being used from a single route table. Turbot will automatically repair this, but advanced cases may require an exception:
- VPC > Route Table Single NAT Gateway

**Policies – Route Table**

Route Tables should never have Blackhole Routes, which is managed by Turbot with:
- VPC > Route Table Blackhole Routes

The main Route Table in is used as a default for Subnets with no specific Route Table assigned. If configured for a specific Subnet Type it would add risk in the case of being a fallback Route Table to other Subnets. Turbot recommends the Main Route Table is not used, but allows advanced customization if required:
- VPC > Main Route Table Not Used

Similarly, by default each Route Table should only be associated with a single Subnet Type:
- VPC > VPC Route Table Types

Unassociated Route Tables can be monitored and cleaned up with:
- VPC > Route Table Associated

**Peering Management**

Peering between VPCs provides significant performance benefits, but is complex to define and very manual to manage – particularly with regards to approved links and
accurate routing. Turbot provides full automation of peering connections, including request, accept and route configuration.

Peering is based on trust between VPCs, which is managed through:

- VPC > Peering Connection Trusted Peers

Using this policy, each VPC specifies a scope of trust for peering. For example, a core management account with shared services may trust all other VPCs for peering; while a specific application may only trust the core account and one other closely related application. VPC > Peering Connection Trusted Peers is resolved by Turbot to provide a list of account / region combinations. For example:

```
# Any VPC in any Region of Turbot Account aaa
- aaa

# Any VPC in us-east-1 of Turbot Account aab
- { id: aab, region: us-east-1 }

# Any VPC in AWS Account 776697123538.
# May be managed by Turbot, or an external account.
- 776697123538

# Any VPC in any Region of any Account in Turbot Cluster c1
- clusterId: c1
```

This list is used both for a definition of approved connections (e.g. from external accounts) as well as those which may be automatically created by Turbot (Turbot managed both sides).

Management of Peering Connections is based on the settings for Policy:

- VPC > Peering Connections

A peering connection is only approved and configured if it’s trusted by both VPCs, so be careful to set this Policy appropriately at both ends.

Management of Routes through Peering Connections (making them actually work), is based on:

- VPC > Peering Connection Routes

Typically, these Policies are set identically, but their separation allows administrators to choose the level of automated configuration. For example, automation of Routes to Peering Connections may be considered lower risk than automated management of actual Peering Connections.
VPC Configuration

While much of the core Network architecture is defined at the Subnet level, the VPC determines shared attributes and configuration settings.

Policies – Security

AWS creates a default VPC in each region. This is convenient for new customers, but undesirable for organizations with tighter control over their network. Turbot is typically configured to automatically delete the default VPC:

- VPC > Default VPCs

Enabling VPC Flog Logs provides a deep audit trail of network activity for inspection and reporting. Turbot provides complete automation of flow logs configuration:

- VPC > VPC Flow Logs Configuration

Since Turbot can automatically configure the Network to various definitions based on the Subnet Type, it’s critical that controls are available for the Subnet Types that may be used in a VPC.

- VPC > Subnet Types Allowed – A list of Subnet Types that are approved for configuration within the VPC. If Networks are managed by a central Network Team this can be wide open, but if Networks are configured by Account teams it is typically restricted to internal Subnet Types (e.g. Private, Restricted, Isolated).
- VPC > Subnet Types – Should Turbot check the Subnet Types Allowed for the VPC.

Policies – General

General Policies for configuration of the VPC service:

- VPC > Regions – AWS regions where the VPC service should be managed by Turbot. Typically, all regions.

Subnet Configuration

Policies – Public IP Management

Resources are allocated private and/or public IPs based on their Subnet location. Public facing Subnets are often designed to automatically allocate public IPs to resources.

In Turbot, Public IP Address allocation is based on the Subnet Type:

- VPC > Subnet Auto Assign Public IP Types
The actual assignment flag on the Subnet is set based on the Subnet Type matching both the Policy above and the setting in:

- VPC > Subnet Auto Assign Public IP

### DNS & DHCP Configuration

Turbot supports configuration and management of settings like DNS and DHCP across many VPCs and Accounts.

#### Policies – DHCP Configuration

DHCP Policies provide common settings across the environment
- VPC > DHCP Options Domain Name
- VPC > DHCP Options Domain Name Servers
- VPC > DHCP Options NTP Servers
- VPC > DHCP Options NetBIOS Name Servers
- VPC > DHCP Options NetBIOS Node Type

DNS and NTP servers often vary by region, providing access to local resources. Turbot Resource Groups can be a powerful way to create shared settings in these scenarios.

#### Policies – DHCP Management

To enforce the DHCP Configuration defined above, use:
- VPC > DHCP Option Settings

DHCP Options can be checked for best practice attachment, and cleaned up if unattached using:
- VPC > DHCP Options Attached
- VPC > DHCP Options Per VPC

#### Policies – DNS Configuration & Management

DNS Search Suffixes are not advertised by DHCP, but required by servers and other resources in the environment to complete their local DNS configuration. Turbot uses this policy in Ansible playbooks etc.
- VPC > DNS Search Suffixes

AWS supports DNS features within the VPC, such as DNS resolution through AmazonProvidedDNS. VPCs can be configured for these features using:
• VPC > VPC DNS Hostnames
• VPC > VPC DNS Resolution

DNS is also configurable for peering connections:
• VPC > Peering Connection DNS Resolution

**Subnet Group Management**

Subnet Groups are used in AWS to specify the Subnets that highly available resources (e.g. RDS Instance, ElastiCache Cluster) should use when creating instances or failing over.

Turbot can automate the setup and management of Subnet Groups based on the Subnet Type and Subnet Purpose. Turbot Subnet Groups combine Subnets that have the same Type - e.g. all Private Subnets; OR all DMZ Subnets. Turbot Subnet Groups are only created if the Subnets have the relevant purpose – e.g. Only Subnets with the DB Purpose will be added to an RDS Subnet Group.

**Policies – Subnet Group Types**

Policies available:
• VPC > ElastiCache Subnet Groups (Subnets with App Purpose)
• VPC > RDS Subnet Groups (Subnets with DB Purpose)
• VPC > Redshift Subnet Groups (Subnets with DB Purpose)

**Policies – Subnet Group Format**

To prevent conflicts, all Subnet Groups managed by Turbot can have an optional prefix:
• VPC > Subnet Group Turbot Name Prefix

**Tag management & resource names**

**Policies - Tags**

As usual, Turbot provides a standard model for automatic tagging of VPC resources. Each resource has two Policies:
• VPC > {resource} Tags Template: Define the tags to be set on the resource.
• VPC > {resource} Tags: Skip, Check or Enforce the Template.
For example:
- VPC > DHCP Options Tags Template
- VPC > DHCP Options Tags
- VPC > Subnet Tags Template
- VPC > Subnet Tags

Policies – Naming Scheme

Clear naming of VPC resources is critical to make navigating the network easier – particularly for key structural resources like VPCs, Subnets and Peering Connections.

The Tags Template can be used to set the Name of resources, but Turbot also provides specific Policies to assist with naming:
- VPC > Route Table Naming Scheme
- VPC > Route Table Naming Scheme Template
- VPC > Subnet Naming Scheme
- VPC > Subnet Naming Scheme Template
- VPC > VPC Naming Scheme
- VPC > VPC Naming Scheme Template

WARNING – Naming Scheme Policies can “fight” with Tags Policies. Please be careful to set them identically, or to choose a single approach. Naming Scheme Policies provide a single advantage over Tags Policies in that they can check custom names meet policy rather than enforcing automated naming as Tags Policies do.

Managing VPC resources at Account level with VPC/* Rights

All Policies above relate to Turbot’s automatic network management capabilities. They operate independently of who created the network or how it was created. All network resources are subject to the automated Guardrails – providing both protection against bad settings, but also acceleration of configuration steps.

Typically, organizations will limit network management to a single, central team to ensure consistent design, implementation and a wider view of the inter-related settings. Bad network configuration in a single location can leave the organization vulnerable to attack.

But, with cloud, we can create a set of automated guardrails and isolation to allow limited network management by individual application teams. The Policies in this section relate to granting permission to Account level teams to manage VPC resources.
Policies – Permissions for VPC Management

When a service is Enabled in Turbot, the permissions for that service will be Whitelisted in the AWS Account and available for granting using Turbot Rights like VPC/Admin, VPC/Metadata, etc.

- VPC > App Enabled – General flag to enable VPC management, referenced by multiple policies.
- VPC > Rights – Specific flag to enable VPC rights management by Turbot.

DO NOT enable VPC > App Enabled unless you require Account level users to manage network resources. Typically, VPC > App Enabled will be Disabled, limiting VPC management to the core cloud or network team.

Policies – Advanced Permission Restrictions

If enabling VPC management for Account teams, it is often important to restrict their access to the exact VPC resources they are permitted to manage. For example, allow VPC and Subnet management but block the use of IGWs (which may open outside traffic).

By default, Turbot allows complete VPC management if VPC > App Enabled is Enabled. But, the following advanced options are provided to allow restriction of the actual resources that can be managed:

- VPC > CGW Management
- VPC > IGW Management
- VPC > DHCP Options Management
- VPC > EGW Management
- VPC > Endpoint Management
- VPC > Flow Logs Management
- VPC > IGW Management
- VPC > NAT Gateway Management
- VPC > Network ACL Management
- VPC > Peering Connection Management
- VPC > Route Table Management
- VPC > Security Group Management
- VPC > Subnet Management
- VPC > VGW Management
- VPC > VPC Management
- VPC > VPN Connection Management

NOTE: Allowing a resource type to be Managed (e.g. user can edit VGW configuration) is different from whether the resource type is Approved (e.g. VGW can exist in the VPC).
Approved VPC resources

Even though Turbot can perform complete management of VPC resources and preventative controls for management, per-resource detective guardrails are also provided. This is a defense in depth measure for automation, but more specifically useful if users are granted permission to manage network resources.

These Policies configure detective controls to complete the preventative Management controls.

Policies – Approved

Approved Policies determine only if the resource is allowed to exist. They do not relate to specific settings that can be modified to meet Policy. For example, a Subnet can exist as defined with its CIDR range, or not – it’s not possible to modify without deleting and recreating the Subnet.

For VPC resources specifically related to connectivity with other networks (e.g. IGW, Peering, etc), Turbot provides two Policies:

- VPC > {resource} Approved – Action to take if resource is not approved.
- VPC > {resource} Approved Usage – Define if resource type is permitted for use.

NOTE – Some resources are still defined in a deprecated “Allowed” format. These will be progressively migrated to the “Approved” format:

- VPC > {resource} Enabled – General flag as to whether this resource type is enabled for the VPC.
- VPC > {resource} Allowed – Specific flag defining what to do if the resource is found.

WARNING: These Policies can “fight” with Turbot network management Policies. If using Turbot for network management then these Policies should all be Enabled (best), Skipped (good) or specifically Enabled for required resources (complex, but good).

Notes on Deprecated Policies

- VPC > Custom VPC Allowed: In earlier Turbot versions networking was assumed to be managed by Turbot through CloudFormation with some small exceptions (Custom VPCs). With Network Guardrails there is no longer any concept of a Custom VPC.
About Turbot

Our mission at Turbot is to unlock the value of public cloud for enterprises through innovation, insight and speed. Turbot has built a strong brand and reputation by hiring the right people, continually innovating and enhancing the product, and providing unparalleled customer service.

Contact Us

Email: info@turbot.com
Twitter: @turbothq
LinkedIn: https://www.linkedin.com/company/turbot
Phone: +1 973 922 0297
Address: https://turbot.com

Turbot HQ, Inc
1732 1st Ave #20232
New York, NY 10128 USA