Software Defined Operations: Application Centric Networking
Overview

The intent of this whitepaper is to outline the architecture, define the terminology, and communicate the concepts built into Turbot’s Application Centric Networking model. The design goals of this model are to:

- Provide clear, consistent and common definitions for the layers of the model in a way that enables all stakeholders (e.g. developers, architects, operations, security and networking) to make informed decisions on policies and configuration.
- Meet the needs of most common enterprise cloud networking patterns through default configuration, while allowing for customization and deeper levels of specificity, complexity for edge cases.
- Create simple building blocks that can effectively communicate context and intent to Turbot’s automation engine, so that policy adherence can easily be calculated and acted upon in a consistent and logical way.

Application Centric Networking

Historically, organizations have designed, built and secured networks from the perspective of organization and physical network boundaries. Infrastructure as a Service (IaaS) and Software Defined Networks (SDN) capabilities now allow organizations to layer security at every level of the network and require a new way of thinking about and architecting network security for cloud workloads.

Application isolation is a core principal of network design and security in the cloud. Isolating applications at the account level allow organizations to use the scale of the cloud provider to leverage the same level of protection between an organization’s own applications as exists between the customers of the cloud service provider. This level of isolation creates clear boundaries for administrative users, access rights, cost control, infrastructure and networking. Automated controls across accounts give the organization the ability to set strong guardrails to enforce policies and implement best practices at scale.

With a strong application isolation, the networking model shifts from data-center centric to application centric; networks exist around and in context of the applications they serve.

This concept of application context is key to any Software Defined Operations Model (SDOps). Understanding the application context allows development, operations and security teams to automate the configuration and management of these environments,
including management of key controls (detective, preventative & corrective) and automated response to events and unintended/unauthorized change.

**Network Security Zones**

The SDOps 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:

<table>
<thead>
<tr>
<th>Zone</th>
<th>CIDR ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Public</strong></td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td><strong>Intranet</strong></td>
<td>10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16</td>
</tr>
<tr>
<td><strong>Trusted</strong></td>
<td>10.1.1.0/24, 10.1.2.0/24</td>
</tr>
<tr>
<td>VPC aaa Interior</td>
<td>10.1.1.0/24</td>
</tr>
<tr>
<td>Subnet 1</td>
<td>10.1.1.0/25</td>
</tr>
<tr>
<td>VM 1</td>
<td>10.1.1.10</td>
</tr>
<tr>
<td>Pair: sg-abcd1234</td>
<td></td>
</tr>
<tr>
<td>VM 2</td>
<td>10.1.1.138</td>
</tr>
<tr>
<td>Subnet 2</td>
<td>10.1.1.128/25</td>
</tr>
<tr>
<td>VPC aab Interior</td>
<td>10.1.2.0/24</td>
</tr>
<tr>
<td>Bastion</td>
<td>192.168.1.0/24</td>
</tr>
</tbody>
</table>

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

**Intranet**: Configured; CIDR ranges specifying the boundaries of the organization’s intranet. Turbot 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

© 2017 Turbot HQ, Inc | https://turbot.com | info@turbot.com
**Interior:** Dynamically calculated; CIDRs of all individual subnets in the current VPC. External subnets are explicitly excluded.

**Bastion:** 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. No Turbot default.

**Pair:** Dynamically calculated; Multiple /32 CIDR ranges (Individual IP addresses) of all resources sharing the same security group.

**Subnets**

Subnets are a key building block of the SDOps Networking Security Model. Subnet routing is calculated and continually enforced by Turbot guardrails based on the following core Subnet Types:

- **Public:** Public IPs; Inbound access from Internet and no connectivity to private networks; outbound internet through cloud service provider.
- **DMZ:** Public & Private IPs; Inbound access from Internet and very limited connectivity to private networks; outbound internet through cloud service provider.
- **Independent:** Private IPs; Limited inbound access from specific known IPs and no access to private networks; outbound internet through cloud service provider.
- **Direct:** Private IPs; No inbound access from public networks; controlled connectivity to private networks; outbound internet through cloud service provider.
- **Private:** Private IPs; No inbound access from public networks; controlled connectivity to private network; outbound internet hair-pinned through private networks.
- **Restricted:** Private IPs; No inbound access from public networks; controlled connectivity to private network; no outbound internet.
- **Isolated:** Private IPs; Limited access from specific known IPs and no access to private networks; no outbound internet.

**Availability Zones**

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones (AZ). Subnets may not cross availability zones, so for any Turbot managed VPC that is configured to expect a specific
subnet type, Turbot ensures that a subnet of that type exists in each available AZ (number of AZs available can be configured as well). An Example:

Subnet Routing: Each subnet type is differentiated by the following attributes:

- **Facing**: A category that describes the intended default facing of the subnet.
  - Inward: Default inbound traffic is from closed networks (e.g. intranet).
  - Outward: Default inbound traffic is from the Internet.
  - None: No default inbound traffic.
- **IP Address Type**: Does the subnet use publicly addressable IP addresses or are they in the privately addressable RFC 1918 ranges
- **Internet Route**: How does traffic destined for ranges in the Public network route out of the subnet.
  - Through an Internet gateway (IGW)
  - Through a network address translation (NAT) instance, or via an egress only Internet gateway (EIGW)
  - Through a virtual gateway (VGW) back to the organization’s private outbound Internet access.
  - No routing allowed to public Internet address ranges.
- **Intranet Route**: How does traffic destined for IP addresses in the Intranet network route out of the subnet.
- **NAT Gateway**: Is a NAT gateway routed from the subnet.
• **Route Table Scope**: Are route tables in the subnet specific to the availability zone or applicable across the entire VPC.

<table>
<thead>
<tr>
<th>Subnet Type</th>
<th>Facing Type</th>
<th>IP Addr. Type</th>
<th>Internet Route</th>
<th>Intranet Route</th>
<th>NAT Gateway</th>
<th>Route Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>outward</td>
<td>Public</td>
<td>IGW</td>
<td>None</td>
<td>Disabled</td>
<td>Per VPC</td>
</tr>
<tr>
<td>DMZ</td>
<td>outward</td>
<td>Public</td>
<td>IGW</td>
<td>None</td>
<td>Disabled</td>
<td>Per VPC</td>
</tr>
<tr>
<td>Independent</td>
<td>none</td>
<td>Private</td>
<td>NAT / EIGW</td>
<td>VGW</td>
<td>Disabled</td>
<td>Per VPC</td>
</tr>
<tr>
<td>Direct</td>
<td>inward</td>
<td>Private</td>
<td>NAT / EIGW</td>
<td>VGW</td>
<td>Enabled</td>
<td>Per-AZ</td>
</tr>
<tr>
<td>Private</td>
<td>inward</td>
<td>Private</td>
<td>VGW</td>
<td>VGW</td>
<td>Disabled</td>
<td>Per VPC</td>
</tr>
<tr>
<td>Restricted</td>
<td>inward</td>
<td>Private</td>
<td>None</td>
<td>VGW</td>
<td>Disabled</td>
<td>Per VPC</td>
</tr>
<tr>
<td>Isolated</td>
<td>none</td>
<td>Private</td>
<td>None</td>
<td>None</td>
<td>Disabled</td>
<td>Per VPC</td>
</tr>
</tbody>
</table>

**Security Groups**

Security groups act as virtual firewalls for instances launched within an AWS VPC. Instances can be associated with up to 5 security groups and because security groups act at the instance level and not the subnet level, different sets of security groups can be applied to instances inside the same subnet – allowing creation of resource zones.

**Security groups Characteristics:**

• By default, a security group has no inbound rules. Therefore, no inbound traffic is allowed until inbound rules are added to the security group.

• If a security group has no outbound rules, no outbound traffic is allowed.

• Security groups specify allow rules, but not deny rules.

• Security groups specify separate rules for inbound and outbound traffic.

• Security groups are stateful — when a request is sent from an instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to approved inbound traffic are allowed to flow out, regardless of outbound rules.

• Instances associated with a security group can’t talk to each other unless rules are added allowing it

• Security groups are associated with network interfaces. Changing the security groups associated with the instance changes the security groups associated with the primary network interface (eth0). Instances can also be associated with additional network interfaces with separate sets of security groups.

**Turbot Security Group Design Goals:**

Common, simple security groups are intended to provide a secure, easily understood starting point for boundary protection and dynamic isolation of application components.
Common security groups do not aim to solve every scenario, applications are expected to create specific, custom security groups according to their specific application requirements & security needs.

Turbot’s security group model should make it simple to implement core security design goals:
- Full application isolation, even inside the intranet.
- Within-application security across outward and inward facing subnets.
- Bastion level access by administrators.
- Trusted sub-networks.

**Default Security Groups**

Turbot enables default security group definitions that are applied across all VPCs in a Network. Specifically:
- **default** – Alias of default_inward.
- **default_inward** – Required for EC2 instances in inward facing subnet types (Direct, Private & Restricted).
- **default_outward** – Required for EC2 instances in outward facing subnet types (Public & DMZ). This group is only created for VPCs configured with outward facing subnets.
- **default_independent** – Required for EC2 instances in “Independent” subnet types. This group is only created for VPCs configured with “Independent” subnets.
- **default_isolated** – Required for EC2 instances in “Isolated” subnet types. This group is only created for VPCs configured with “Isolated” subnets.

Default Security Groups are critical to provide:
- Common egress rules (e.g. Active Directory, web).
- Security based ingress rules (e.g. security scanners, SSH, RDP).

Turbot provides guardrails to ensure that the default security groups are always used appropriately, for example:
- EC2 instances in inward facing subnets MUST have the default security group and MUST NOT use the default_outward security group.
- EC2 instances in outward facing subnets MUST use the default_outward security group and MUST NOT use the default/default_inward security group.
**Function based Port Sets**

Custom security groups can be highly specific, but standard security groups should be simple and flexible enough to understand. Otherwise users reduce security through arbitrary application of rules.

Turbot encourages the use of standard ports by default – this reduces application complexity and the learning curve for developers. If obfuscation is required, custom security groups should be adopted for those specific application cases.

Turbot breaks security groups into common port sets, striking a balance between simplicity of groups and common security patterns. Here is our recommendation which covers a majority of use cases:

**Web:** By default, http/https are authorized; however, many organizations are moving to only allow https (even for intranet only workloads), and this posture can easily be configured into policy.

**App:** These are ports common in use across Tomcat/Weblogic and other app servers.

**DB:** Common default ports for most typical enterprise databases.

<table>
<thead>
<tr>
<th>Function</th>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>TCP</td>
<td>80</td>
<td>http</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>443</td>
<td>https</td>
</tr>
<tr>
<td>App</td>
<td>TCP</td>
<td>8080-8083</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>8443-8446</td>
<td></td>
</tr>
<tr>
<td>DB</td>
<td>TCP</td>
<td>1433</td>
<td>SQL Server</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>1521</td>
<td>Oracle</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>3306</td>
<td>MySQL</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>5432</td>
<td>Postgres</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>5439</td>
<td>Redshift</td>
</tr>
</tbody>
</table>

**Source Based Access Controls**

Ports map out functionality, but to be effective security groups need to carefully manage the source of access. For standard security groups, Turbot uses the previously articulated Network model to define the source of network traffic:

**Public:** Configured; CIDR ranges specifying the Internet. Turbot Default:
- 0.0.0.0/0
- ::/0
Intranet: Configured; CIDR ranges specifying the boundaries of the organization’s intranet. Turbot 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: CIDR ranges specifying the boundaries of networks with trusted bastion servers (jump boxes). No default.

Pair: Dynamically calculated; Multiple /32 CIDR ranges (Individual IP addresses) of all resources sharing the same security group.

Standard Security Groups

By combining source Network Zone and function together, a standard set of security groups is easily understood and applied. For Example:

intranet_web: Access to web ports (80/443) from within the intranet.

public_web: Access to web ports (80/443) from the Internet.

interior_db: Access to the database from within the VPC.

Application stacks can then use combinations of these to create secure configurations. For example, a common Intranet application:

- Intranet facing load balancer: intranet_web
- Intranet facing Web server: intranet_web
- Application server only accessed within a VPC: interior_app
- Database server only accessed within a VPC: interior_db

Pair Security Groups

In some cases we wish to pair servers for access, making them more specifically connected than the functional groups provide. For example, a DMZ based Elastic Load Balancer can be paired with an Intranet based Web server for specific access.
Turbot provides pair security groups for this purpose:

- **pair_web**: All instances with the “pair_web” security groups allow inbound access to ports 80/443 from other instances that share that security group.
- **pair_db**: e.g. All Oracle databases sharing “pair_db” can communicate with each other on port 1521.
- **pair_app**: e.g. Would allow an application load balancer to talk to all app server instances sharing the “pair_app” security group.

The pair security group is defined with ingress & egress rules to itself for the specific function. For example, pair_web is defined as:

- Allow ingress on TCP-80 from the pair security group.
- Allow ingress on TCP-443 from the pair security group.
- Allow egress on TCP-80 to the pair security group.
- Allow egress on TCP-443 to the pair security group.

An outward facing web site can make use of pairing to allow communication across the DMZ. Note that without the pair_web connection this outward facing load balancer has no access to any resources in the VPC.

- Public facing load balancer in outward facing subnets: public_web & pair_web.
- Web server in inward facing subnets: pair_web.
- App server in inward facing subnets: private_app.
- DB server in inward facing subnets: private_db.

Pairs can be used throughout the stack to prevent any unwanted access. For example, to prevent the Web server having any possible access to the DB server use this model:

- Public facing load balancer in outward facing subnets: public_web & pair_web.
- Web server in inward facing subnets: pair_web & pair_app.
- App server in inward facing subnets: pair_app & pair_db.
- DB server in inward facing subnets: pair_db.

Pairs are simple, yet powerful. This functionality is enabled by the Turbot best practice of separating each application into their own VPCs.

**Simple Open Access Inside a VPC**

Some environments, including development, may desire simpler open access within the VPC. The VPC itself is separated from the rest of the Intranet, so this provides low risk but with high flexibility for the application team.

Turbot optionally provides a simple combination to suit this scenario called:

- **interior_all**: This group provides access on any protocol to any port from all subnets within the 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