PRJ-SEC-007: Zero Trust Network on AWS

AWS Console Step-by-Step Deployment Guide

Author: Mo Suleiman | **Project:** PRJ-SEC-007 | **Date:** 2026-03-06

Overview

This guide walks you through building a **Zero Trust network architecture** on AWS entirely through the AWS Management Console. The architecture enforces the principle that no traffic is trusted by default — all communication between VPCs and the internet is routed through a centralised **AWS Network Firewall** for inspection before being permitted.

GRC Mapping

Framework	Control	Description
NIST CSF	PR.AC-5	Network integrity is protected by enforcing network segmentation and inspection.
ISO 27001	A.13.1.3	Networks are segregated to protect sensitive information systems.
PCI-DSS	1.3	Prohibit direct public access between the internet and any component in the cardholder data environment.
Business Value	—	Eliminates implicit trust between internal VPCs, contains lateral movement after a breach, and provides a centralised audit point for all network traffic.

Architecture Components

The deployment consists of four VPCs connected through a Transit Gateway, with all traffic forced through a Network Firewall in the central Inspection VPC before reaching its destination.

Component	Name	Purpose
Inspection VPC	`prj-sec-007-inspection-vpc`	Houses the Network Firewall. All traffic transits through here.
Web VPC	`prj-sec-007-web-vpc`	Public-facing web tier (`10.1.0.0/16`)
App VPC	`prj-sec-007-app-vpc`	Application logic tier (`10.2.0.0/16`)
DB VPC	`prj-sec-007-db-vpc`	Database tier (`10.3.0.0/16`)
Transit Gateway	`prj-sec-007-tgw`	Cloud router connecting all four VPCs
Network Firewall	`prj-sec-007-firewall`	Stateless + stateful inspection engine

Prerequisites

Before starting, confirm the following:

- You are logged into the AWS Console with an IAM user or role that has `AdministratorAccess` or equivalent permissions for VPC, EC2, Network Firewall, and Transit Gateway.
- Your region is set to **us-east-1** (visible in the top-right corner of the console).
- You have a plain text editor open to store resource IDs as you create them. You will need to copy and paste IDs frequently throughout this guide.

Part 1: Create All Four VPCs

Navigate to the **VPC** service using the search bar at the top of the console.

1.1 — Inspection VPC

Go to **Your VPCs** → **Create VPC**. Select **VPC only** (not VPC and more). Fill in the fields exactly as follows and click **Create VPC**.

Field	Value
Name tag	`prj-sec-007-inspection-vpc`
IPv4 CIDR block	`10.0.0.0/16`
IPv6 CIDR block	No IPv6 CIDR block
Tenancy	Default

After creation, copy the **VPC ID** (e.g., `vpc-0abc...`) to your text editor and label it `INSPECTION_VPC_ID`.

1.2 — Web, App, and DB VPCs

Repeat the same process three more times with the following values:

VPC Name	IPv4 CIDR	Label to Save
`prj-sec-007-web-vpc`	`10.1.0.0/16`	`WEB_VPC_ID`
`prj-sec-007-app-vpc`	`10.2.0.0/16`	`APP_VPC_ID`
`prj-sec-007-db-vpc`	`10.3.0.0/16`	`DB_VPC_ID`

Part 2: Create Subnets

Go to **Subnets** → **Create subnet**. Select the correct VPC for each group of subnets below. You can add multiple subnets in a single creation flow by clicking **Add new subnet**.

2.1 — Inspection VPC Subnets

Select **VPC ID:** `INSPECTION_VPC_ID`. Create four subnets:

Subnet Name	Availability Zone	IPv4 CIDR	Purpose
`prj-sec-007-inspection-fw-a`	us-east-1a	`10.0.1.0/24`	Firewall endpoint AZ-A
`prj-sec-007-inspection-fw-b`	us-east-1b	`10.0.2.0/24`	Firewall endpoint AZ-B
`prj-sec-007-inspection-tgw-a`	us-east-1a	`10.0.3.0/24`	TGW attachment AZ-A
`prj-sec-007-inspection-tgw-b`	us-east-1b	`10.0.4.0/24`	TGW attachment AZ-B

After creation, copy all four **Subnet IDs** to your text editor with their labels.

2.2 — Spoke VPC Subnets

Repeat for each spoke VPC:

VPC	Subnet Name	AZ	IPv4 CIDR
Web VPC	`prj-sec-007-web-a`	us-east-1a	`10.1.1.0/24`
Web VPC	`prj-sec-007-web-b`	us-east-1b	`10.1.2.0/24`
App VPC	`prj-sec-007-app-a`	us-east-1a	`10.2.1.0/24`
App VPC	`prj-sec-007-app-b`	us-east-1b	`10.2.2.0/24`
DB VPC	`prj-sec-007-db-a`	us-east-1a	`10.3.1.0/24`
DB VPC	`prj-sec-007-db-b`	us-east-1b	`10.3.2.0/24`

Part 3: Create Internet Gateway

Go to **Internet Gateways** → **Create internet gateway**.

Field	Value
Name tag	`prj-sec-007-igw`

Click **Create internet gateway**. On the next screen, click **Actions** → **Attach to VPC** and select the `prj-sec-007-inspection-vpc`. Copy the **IGW ID** and label it `IGW_ID`.

Part 4: Create Transit Gateway

Navigate to **Transit Gateways** using the left sidebar (under **Transit Gateways**) → **Create transit gateway**.

Field	Value
Name tag	`prj-sec-007-tgw`
Description	Zero Trust hub router
Amazon side ASN	`64512` (default)
DNS support	Enabled
VPN ECMP support	Enabled
Default route table association	**Disabled**
Default route table propagation	**Disabled**
Auto accept shared attachments	Disabled

> ***Important:*** *Disabling default route table association and propagation is critical. It prevents any VPC from automatically being able to reach any other VPC. You will manually control all routing.*

Click **Create transit gateway**. Wait approximately 2–3 minutes for the state to change from `pending` to `available`. Copy the **Transit Gateway ID** and label it `TGW_ID`.

4.1 — Create TGW Attachments

Go to **Transit Gateway Attachments** → **Create transit gateway attachment**. Create one attachment per VPC.

Inspection VPC Attachment:

Field	Value
Name tag	`prj-sec-007-tgw-attach-inspection`
Transit gateway ID	`TGW_ID`
Attachment type	VPC
VPC ID	`INSPECTION_VPC_ID`
Subnet IDs	Select `prj-sec-007-inspection-tgw-a` AND `prj-sec-007-inspection-tgw-b`

Copy the **Attachment ID** and label it `ATTACH_INSPECTION`.

Spoke VPC Attachments: Repeat for each spoke VPC:

Attachment Name	VPC	Subnets	Label
`prj-sec-007-tgw-attach-web`	Web VPC	`web-a` and `web-b`	`ATTACH_WEB`
`prj-sec-007-tgw-attach-app`	App VPC	`app-a` and `app-b`	`ATTACH_APP`
`prj-sec-007-tgw-attach-db`	DB VPC	`db-a` and `db-b`	`ATTACH_DB`

Wait for all four attachments to show state `available` before proceeding.

4.2 — Create TGW Route Tables

Go to **Transit Gateway Route Tables** → **Create transit gateway route table**.

Route Table 1 — Spokes to Inspection:

Field	Value
Name tag	`prj-sec-007-tgw-rt-spokes`
Transit gateway ID	`TGW_ID`

After creation, select this route table and perform the following actions:

- **Associations tab → Create association**: Associate `ATTACH_WEB`, `ATTACH_APP`, and `ATTACH_DB` (one at a time).
- **Routes tab → Create static route**: CIDR `0.0.0.0/0`, Attachment: `ATTACH_INSPECTION`. This forces all spoke traffic to the Inspection VPC.

Route Table 2 — Inspection to Spokes:

Field	Value
Name tag	`prj-sec-007-tgw-rt-inspection`
Transit gateway ID	`TGW_ID`

After creation, select this route table and perform the following actions:

- **Associations tab → Create association**: Associate `ATTACH_INSPECTION`.
- **Propagations tab → Create propagation**: Add propagations for `ATTACH_WEB`, `ATTACH_APP`, and `ATTACH_DB`. This automatically populates the route table with the spoke VPC CIDRs.

Part 5: Create Network Firewall

Navigate to **Network Firewall** using the search bar.

5.1 — Create Stateless Rule Group

Go to **Network Firewall rule groups → Create network firewall rule group**.

Field	Value
Name	`zero-trust-stateless-block`
Type	Stateless
Capacity	`100`

Under **Stateless rules**, add the following three rules to block high-risk ports:

Priority	Protocol	Destination Port	Action
10	TCP (6)	445–445	Drop
20	TCP (6)	3389–3389	Drop
30	TCP (6)	137–139	Drop
100	All traffic	Any	Forward to stateful rule groups

Click **Create rule group**.

5.2 — Create Stateful Domain Allow-List Rule Group

Go to **Create network firewall rule group** again.

Field	Value
Name	`zero-trust-stateful-domain-allowlist`
Type	Stateful
Capacity	`100`
Rule order	Strict order

Under **Rules**, select **Domain list**. Set the following:

Field	Value
Rule type	Allow-list
Protocols	HTTP, HTTPS
Domains	`.amazonaws.com`, `.ubuntu.com`, `.debian.org`

Click **Create rule group**.

5.3 — Create Stateful East-West Rule Group

Go to **Create network firewall rule group** again.

Field	Value
Name	`zero-trust-stateful-east-west`
Type	Stateful
Capacity	`200`
Rule order	Strict order

Under **Rules**, select **Suricata compatible rule string** and paste the following:

```
pass tcp 10.1.0.0/16 any -> 10.2.0.0/16 8080 (msg:"Allow Web to App";
sid:1001; rev:1;)
pass tcp 10.2.0.0/16 any -> 10.3.0.0/16 5432 (msg:"Allow App to DB";
sid:1002; rev:1;)
drop tcp any any -> any any (msg:"Block all other East-West TCP"; sid:9999;
rev:1;)
```

Click **Create rule group**.

5.4 — Create Firewall Policy

Go to **Firewall policies** → **Create firewall policy**.

Field	Value
Name	`prj-sec-007-fw-policy`

Under **Stateless default actions**, set both **Fragmented packets** and **Full packets** to **Forward to stateful rule groups**.

Under **Stateless rule groups**, add `zero-trust-stateless-block` with priority `10`.

Under **Stateful rule evaluation order**, select **Strict order**.

Under **Stateful default actions**, select **Drop all**.

Under **Stateful rule groups**, add both `zero-trust-stateful-domain-allowlist` (priority 10) and `zero-trust-stateful-east-west` (priority 20).

Click **Create firewall policy**.

5.5 — Create the Firewall

Go to **Firewalls** → **Create firewall**.

Field	Value
Name	`prj-sec-007-firewall`
VPC	`prj-sec-007-inspection-vpc`

Under **Availability Zones**, add both AZs:

Availability Zone	Subnet
us-east-1a	`prj-sec-007-inspection-fw-a`
us-east-1b	`prj-sec-007-inspection-fw-b`

Under **Firewall policy**, select `prj-sec-007-fw-policy`. Click **Create firewall**.

> ***Wait 5–10 minutes*** *for the firewall to reach the* `Ready` *state. This is the longest provisioning step.*

Once ready, click on the firewall name → **Firewall details** tab. Under **Endpoints**, copy the **Endpoint ID** for each AZ. They look like `vpce-0abc...`. Label them `FW_ENDPOINT_AZ_A` and `FW_ENDPOINT_AZ_B`.

Part 6: Configure VPC Route Tables

This is the most critical part of the deployment. Incorrect routing will either break connectivity or allow traffic to bypass the firewall. You must create separate route tables for each subnet type.

6.1 — Inspection VPC: TGW Subnet Route Tables

Go to **Route Tables** → **Create route table**.

TGW Subnet Route Table (AZ-A):

Field	Value
Name	`prj-sec-007-inspection-tgw-rt-a`
VPC	`prj-sec-007-inspection-vpc`

After creation, go to the **Subnet associations** tab → **Edit subnet associations** → associate `prj-sec-007-inspection-tgw-a`.

Go to the **Routes** tab → **Edit routes** → **Add route**:

Destination	Target
`0.0.0.0/0`	`FW_ENDPOINT_AZ_A` (select VPC Endpoint from the dropdown)

TGW Subnet Route Table (AZ-B): Repeat the above for AZ-B, associating `prj-sec-007-inspection-tgw-b` and pointing the default route to `FW_ENDPOINT_AZ_B`.

6.2 — Inspection VPC: Firewall Subnet Route Tables

Firewall Subnet Route Table (AZ-A):

Field	Value
Name	`prj-sec-007-inspection-fw-rt-a`
VPC	`prj-sec-007-inspection-vpc`

Associate with `prj-sec-007-inspection-fw-a`. Add the following routes:

Destination	Target
`10.1.0.0/16`	Transit Gateway (`TGW_ID`)
`10.2.0.0/16`	Transit Gateway (`TGW_ID`)
`10.3.0.0/16`	Transit Gateway (`TGW_ID`)
`0.0.0.0/0`	Internet Gateway (`IGW_ID`)

Firewall Subnet Route Table (AZ-B): Repeat for AZ-B, associating `prj-sec-007-inspection-fw-b` with the same routes.

6.3 — Spoke VPC Route Tables

For each spoke VPC, create a single route table and associate it with both subnets in that VPC. Each spoke route table has only one route: a default route pointing to the Transit Gateway.

Route Table Name	VPC	Subnets to Associate	Route: Destination	Route: Target
`prj-sec-007-web-rt`	Web VPC	`web-a`, `web-b`	`0.0.0.0/0`	Transit Gateway
`prj-sec-007-app-rt`	App VPC	`app-a`, `app-b`	`0.0.0.0/0`	Transit Gateway
`prj-sec-007-db-rt`	DB VPC	`db-a`, `db-b`	`0.0.0.0/0`	Transit Gateway

Part 7: Create Security Groups

Navigate to **VPC** → **Security Groups** → **Create security group**. Create one security group per tier.

Web Tier Security Group:

Field	Value
Name	`prj-sec-007-web-sg`
VPC	Web VPC

Inbound rules:

Type	Protocol	Port	Source	Description
HTTP	TCP	80	`0.0.0.0/0`	Public web traffic
HTTPS	TCP	443	`0.0.0.0/0`	Public web traffic

App Tier Security Group:

Field	Value
Name	`prj-sec-007-app-sg`
VPC	App VPC

Inbound rules:

Type	Protocol	Port	Source	Description
Custom TCP	TCP	8080	`10.1.0.0/16`	Web VPC only

DB Tier Security Group:

Field	Value
Name	`prj-sec-007-db-sg`
VPC	DB VPC

Inbound rules:

Type	Protocol	Port	Source	Description
PostgreSQL	TCP	5432	`10.2.0.0/16`	App VPC only

Part 8: Enable Firewall Logging

Go to **Network Firewall** → **Firewalls** → select `prj-sec-007-firewall` → **Logging** tab → **Edit**.

First, create two CloudWatch Log Groups by navigating to **CloudWatch** → **Log groups** → **Create log group**:

Log Group Name	Retention
`/aws/network-firewall/prj-sec-007/flow`	30 days
`/aws/network-firewall/prj-sec-007/alert`	30 days

Back in the firewall logging settings, add two log destinations:

Log type	Destination type	Log group
Flow	CloudWatch Logs	`/aws/network-firewall/prj-sec-007/flow`
Alert	CloudWatch Logs	`/aws/network-firewall/prj-sec-007/alert`

Click **Save**.

Part 9: Validation

To validate the architecture is working correctly, launch a test EC2 instance in the Web VPC using the AWS Systems Manager Session Manager (no SSH key required). From the instance terminal, test the following:

```
# This should SUCCEED (allowed by domain allowlist)
curl -I https://aws.amazon.com

# This should FAIL (blocked by East-West rules — App VPC not reachable from Web without going through firewall)
curl -m 5 http://10.2.1.47:8080

# This should FAIL (RDP port blocked by stateless rule group)
nc -zv 10.2.1.47 3389
```

Check the **Alert** CloudWatch Log Group to see the firewall drop events for the blocked traffic.

Part 10: Cleanup

Delete resources in the **reverse order** of creation to avoid dependency errors:

1. Terminate EC2 test instances.
2. Delete Security Groups.
3. Delete the Network Firewall (wait for it to fully delete before proceeding).
4. Delete the Firewall Policy.
5. Delete all three Rule Groups.
6. Delete TGW Attachments (wait for all to reach `deleted` state).
7. Delete TGW Route Tables.
8. Delete the Transit Gateway.
9. Delete all Route Tables (custom ones only — the main route table cannot be deleted).
10. Delete the Internet Gateway (detach from VPC first).

11. Delete all Subnets.
12. Delete all four VPCs.
13. Delete CloudWatch Log Groups.

> ***Estimated cost for a 1-hour deployment:*** *The Transit Gateway (~* $0.05/hr), NetworkFirewall($ *0.40/hr), and data processing charges are the primary costs. Always clean up after testing to avoid ongoing charges.*