Configuration Manual

MSc Research Project
Master of Science in Cloud Computing

Sandra Jacintha Mohanraj
Student ID: 23302241

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sandra Jacintha Mohanraj
Student ID:	23302241
Programme:	Master of Science in Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	11/08/2025
Project Title:	Designing and Enhancing Cloud Security through the Implementation of Zero Trust Architecture in Cloud Environments
Word Count:	1706
Page Count:	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sandra
Date:	11th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	YES
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	YES
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	YES

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sandra Jacintha Mohanraj
23302241

1 Introduction

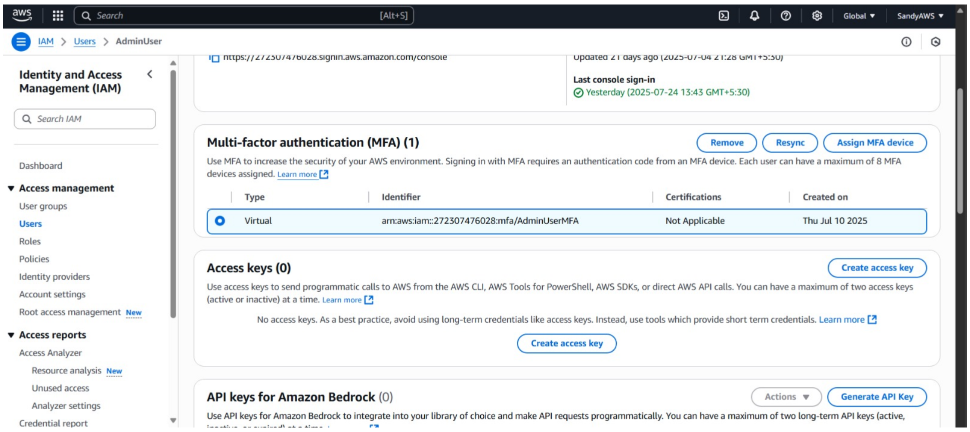
This manual provides a detailed, step-by-step guide for the configuration and deployment of a Zero Trust Architecture (ZTA) within the Amazon Web Services (AWS) cloud platform. The objective is to construct a secure, multi-tiered environment based on the core ZTA principles of assuming breach, verifying explicitly, and enforcing least privilege access. By following these instructions, the user will provision a complete infrastructure, including networking, compute resources, identity controls, and a comprehensive security monitoring fabric. The manual concludes with instructions for executing a custom-developed, automated testing framework to empirically validate the security posture and quantify the performance characteristics of the implemented architecture. Upon completion, the user will have a fully functional and validated ZTA prototype.

2 Initial IAM User and MFA Configuration

This foundational section details the creation of the necessary user accounts and enforces the core Zero Trust principle of mandatory Multi-Factor Authentication (MFA). Identity is the primary control plane in this architecture.

1. Sign in to the AWS Management Console as the root user and navigate to the IAM (Identity and Access Management) console.
2. **Create an Administrative User:** The first step is to create a dedicated user for administrative tasks, adhering to the best practice of not using the root account for routine configuration.
   - Create a new IAM user with the name `AdminUser`.
   - Attach the `AdministratorAccess` AWS managed policy. This provides the necessary permissions for the initial setup.
   - Create a strong, unique console password for this user.
3. **Assign and Enforce MFA:** To ensure strong authentication, MFA is made mandatory for the administrative user.
   - Navigate to the security credentials tab of the newly created `AdminUser`.
   - Assign a new MFA device. Use a virtual MFA application (e.g., Google Authenticator on a mobile device) to scan the QR code and complete the setup. The resulting MFA device will have a unique Amazon Resource Name (ARN) similar to `arn:aws:iam::272307476028:mfa/AdminUserMFA1`.

   - Log out and immediately log back in as `AdminUser` to confirm that the console now requires both a password and an MFA code for access. This verifies the successful enforcement of strong authentication.



https://272307476028.signin.aws.amazon.com/console

Updated 21 days ago (2025-07-04 21:28 GMT+5:30)

Last console sign-in

✓ Yesterday (2025-07-24 13:43 GMT+5:30)

Multi-factor authentication (MFA) (1)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

	Type	Identifier	Certifications	Created on
●	Virtual	arn:aws:iam::272307476028:mfa/AdminUserMFA	Not Applicable	Thu Jul 10 2025

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more

API keys for Amazon Bedrock (0)

Use API keys for Amazon Bedrock to integrate into your library of choice and make API requests programmatically. You can have a maximum of two long-term API keys (active,

Figure 1: IAM Console showing the assigned MFA device for the AdminUser.

4. **Create a Least-Privilege Test User:** To demonstrate the principle of least privilege, a non-administrative user is created with strictly limited permissions.
   - While logged in as the `AdminUser`, create another IAM user named `AppUser`.
   - Attach the following AWS managed policies: `AmazonS3ReadOnlyAccess` and `AmazonEC2ReadOnlyAccess`. These policies grant permissions to view resources but deny any modification or creation actions.
   - Create a console password for this user.
5. **Create the Mandatory MFA Policy:** A custom policy is created to programmatically enforce MFA for a set of test users.
   - In the IAM console, navigate to Policies and create a new policy named `mfacompliancepolicy`.
   - Using the JSON editor, input the policy that explicitly denies all actions if the user's session is not authenticated with MFA. This policy is the cornerstone of the identity-centric approach, programmatically enforcing strong authentication as a prerequisite for any action.

3 Building the Network Foundation (VPC)

This section covers the creation of the logically isolated network environment. This VPC acts as the virtual data center, providing the foundational boundary for the architecture and enabling micro-segmentation.

1. Navigate to the VPC console in the AWS Management Console.
2. **Create the Main VPC:** This VPC will contain all network resources.
   - Initiate the "Create VPC" process.

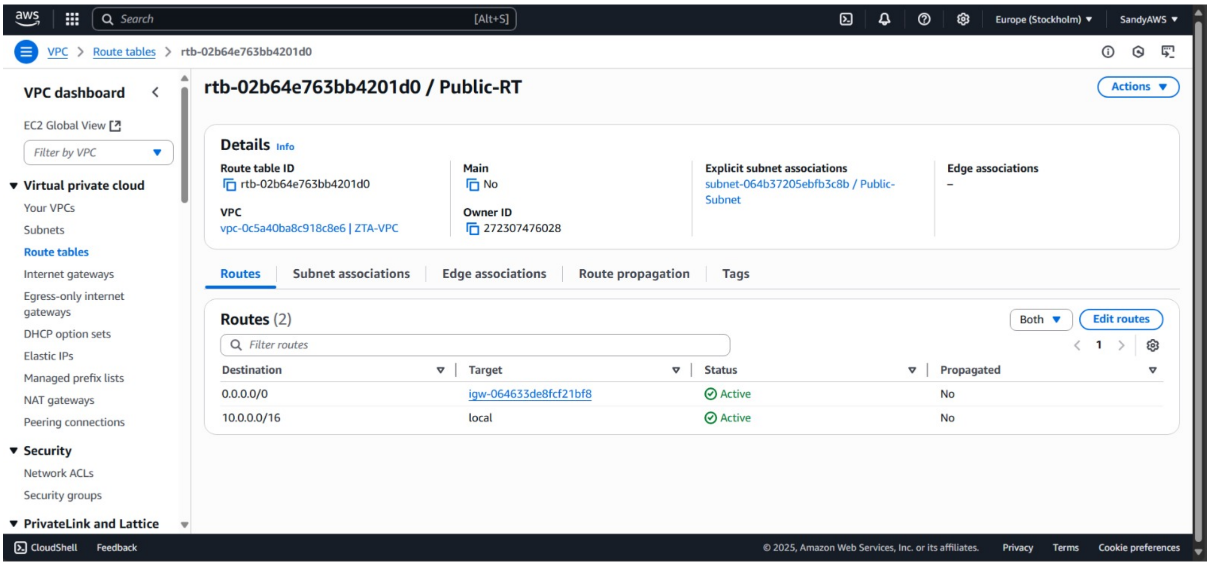
    - **Name:** `ZTA-VPC`
    - **IPv4 CIDR block:** `10.0.0.0/16`. This provides a large private address space.

3. **Create and Configure Subnets for Micro-segmentation:**

    - Create a **Public Subnet** named `Public-Subnet` with a CIDR block of `10.0.1.0/24`. In its settings, it is critical to enable ”Auto-assign public IPv4 address” so that resources launched within it can be reached from the internet.
    - Create a **Private Subnet** named `Private-Subnet` with a CIDR block of `10.0.2.0/24`. This subnet will host protected resources and must not have direct internet access.

4. **Configure Internet Gateway and Routing:** This step establishes the path for internet traffic to and from the public subnet only.

    - Create an Internet Gateway named `ZTA-IGW` and attach it to the `ZTA-VPC`.
    - Create a new Route Table named `Public-RT`.
    - Edit the routes for `Public-RT` and add a default route (Destination `0.0.0.0/0`) with the Target set to the newly created `ZTA-IGW`.
    - Associate this `Public-RT` with the `Public-Subnet`. The `Private-Subnet` should remain associated with the main (default) route table, which contains no route to the internet, thus ensuring its isolation.



rtb-02b64e763bb4201d0 / Public-RT

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-02b64e763bb4201d0	No	subnet-064b37205ebfb3c8b / Public-Subnet	–

VPC	Owner ID
vpc-0c5a40ba8c918c8e6 | ZTA-VPC	272307476028

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-064633de8fcf21bf8	✓ Active	No
10.0.0.0/16	local	✓ Active	No

Figure 2: VPC Route Table `Public-RT` showing the default route to the Internet Gateway, enabling internet access only for its associated subnet.

4 Configuring Network Access Controls

Security Groups function as stateful, instance-level firewalls and are the primary mechanism for enforcing the micro-segmentation policy defined in the previous section. Their configuration is detailed in Table 1.

1. Navigate to "Security Groups" under the EC2 service in the console.
2. **Create the Web Server Security Group (Web-SG):** This group defines the allowed inbound traffic for the public-facing server.
   - Set the Name to Web-SG and associate it with the ZTA-VPC.
   - Add the specified inbound rules for public web traffic (HTTP, HTTPS) and administrative access (SSH).
3. **Create the Database Security Group (DB-SG):** This group enforces strict controls to protect the database.
   - Set the Name to DB-SG and associate it with the ZTA-VPC.
   - Add inbound rules for the database port and SSH. Crucially, the source for these rules must be set to the Group ID of the Web-SG. This is a ZTA best practice that ensures the database can only be reached from the application tier, effectively preventing lateral movement from any other source.

Table 1: Security Group Inbound Rule Configuration for Micro-segmentation

Group Name	Protocol	Port Range	Source
Web-SG	TCP	80	0.0.0.0/0 (HTTP)
	TCP	443	0.0.0.0/0 (HTTPS)
	TCP	22	0.0.0.0/0 (SSH)
DB-SG	TCP	3306	Group ID of Web-SG (MySQL/Aurora)
	TCP	22	Group ID of Web-SG (SSH for bastion access)

5 Deploying Compute Infrastructure

This step deploys the virtual servers for the web and database tiers into their respective secure subnets. The configuration for each instance is summarized in Table 2.

1. Navigate to the EC2 console.
2. **Create a Key Pair for Secure Access:**
   - Create a new key pair named ZTA-KeyPair.
   - Select **RSA** for the key pair type and **.pem** for the private key file format.
   - Download and store this private key file in a secure location; it is required for SSH access.
3. **Launch Instances:**
   - Launch two separate EC2 instances (WebServer and DBServer) according to the detailed specifications outlined in Table 2.

- For each instance, utilize the ”Advanced Details” section to provide a user data script. This script will run automatically on the first boot to install and configure the necessary software (Apache for the `WebServer` and MariaDB for the `DBServer`).

Table 2: EC2 Instance Configuration Specifications

Parameter	**WebServer Configuration**	**DBServer Configuration**
Name	`WebServer`	`DBServer`
AMI	Amazon Linux 2023	Amazon Linux 2023
Instance Type	`t3.micro`	`t3.micro`
Key Pair	`ZTA-KeyPair`	`ZTA-KeyPair`
VPC	`ZTA-VPC`	`ZTA-VPC`
Subnet	`Public-Subnet`	`Private-Subnet`
Security Group	`Web-SG`	`DB-SG`
Auto-assign Public IP	**Enabled**	**Disabled**

6 Enabling Security Services and Data Protection

This section details the configuration of the continuous monitoring fabric and data encryption controls, which are essential for threat detection and data security in a ZTA.

1. **Enable Data-at-Rest Encryption:**

    - In the KMS console, create a customer-managed, symmetric key with the alias `ztakmskey`. Define the `AdminUser` as a key administrator.
    - In the S3 console, create a bucket named `zta-s3-bucket`.
    - In the bucket’s properties, edit the ”Default encryption” settings to use Server-Side Encryption with KMS (SSE-KMS), selecting the custom `ztakmskey`. This ensures all data is encrypted by default.

2. **Enable Monitoring, Logging, and Threat Detection:**

    - **CloudTrail:** Create a trail named `ZTA-CloudTrail`. Configure it to be multi-region, log all management and data events, and deliver the logs to a new, secure S3 bucket.
    - **GuardDuty:** Navigate to the GuardDuty console and enable the service with a single click. GuardDuty will immediately begin analyzing logs for threats.
    - **VPC Flow Logs:** In the VPC console, select the `ZTA-VPC` and create a new flow log. Configure it to capture ”All” traffic (Accepted and Rejected) and to send the logs to a new CloudWatch Logs group for analysis.

7 Executing the Automated Test Framework

The final step is to run the comprehensive, automated test suite from a local machine to validate the architecture's security controls and measure its performance characteristics.

1. **Local Environment Prerequisites:**
   - The AWS Command Line Interface (CLI) must be installed and configured on the local machine. It should be authenticated with credentials that have administrative access (e.g., using the `AdminUser` profile created in Step 1).
   - Python 3 and the AWS SDK for Python ('boto3') must be installed.
2. **Prepare the Test Suite:**
   - Place all provided test scripts (e.g., `create_test_users.bat`, `run_zta_tests.bat`, `performance_monitor.py`, etc.) into a single, dedicated directory on the local machine.
3. **Execute the Master Test Script:**
   - Open a command prompt or terminal and navigate to the directory containing the scripts.
   - Run the master script, which orchestrates the entire validation process, by executing the following command:

     ```
     execute_all_tests.bat
     ```

   - The script will proceed through all automated stages: creating test user personas, applying security policies, running a series of functional security checks, measuring performance latency, and simulating a user revocation event. The entire process is logged to the console and a local file for subsequent review and analysis.

8 Conclusion

By following the procedures outlined in this manual, a user can successfully deploy a complete Zero Trust Architecture within the AWS cloud. The resulting environment is not only built upon foundational ZTA principles like micro-segmentation and least privilege but is also equipped with a full suite of monitoring and data protection services. Furthermore, the provided automated testing framework allows for the immediate and repeatable validation of the architecture's security effectiveness and performance, confirming that the system is both secure and efficient. This manual provides a practical, end-to-end guide for building and verifying a modern cloud security posture.