**UNIVERSITY OF INFORMATION TECHNOLOGY**

**FACULTY OF COMPUTER NETWORKS AND COMMUNICATIONS**



**NETWORKS SECURITY**

**PROJECT FINAL REPORT**

***GROUP 16:***

# **ZERO TRUST NETWORK ACCESS**

# 

## Lecturer: PhD. Nguyễn Ngọc Tự

## Students: Nguyễn Đại Nghĩa - 21521182

## Phạm Hoàng Phúc - 21521295

## Hoàng Gia Bảo - 21521848

Table of contents

**I. INTRODUCTION ............................................................................. 3**

**II. SCENARIO ....................................................................................... 3**

**III. SOLUTION ..................................................................................... 4**

**IV. SECURITY GOALS ......................................................................... 9**

**V. REFERENCES .................................................................................... 9**

# **Introduction**

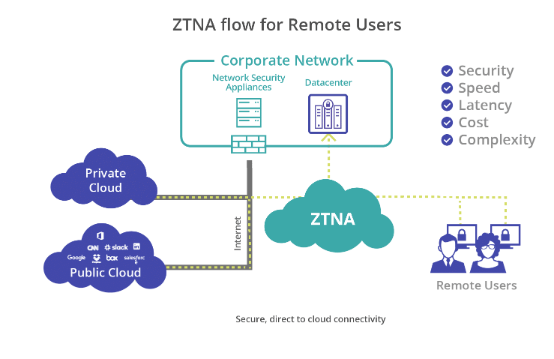
NIST defines both Zero Trust and a Zero Trust Architecture as: "Zero trust (ZT) provides a collection of concepts and ideas designed to minimize uncertainty in enforcing accurate, least privilege per-request access decisions in information systems and services in the face of a network viewed as compromised. Zero Trust Architecture (ZTA) is an enterprise’s cybersecurity plan that uses zero trust concepts and encompasses component relationships, workflow planning, and access policies. Therefore, a zero trust enterprise is the network infrastructure (physical and virtual) and operational policies that are in place for an enterprise as a product of a ZTA plan." Zero Trust Network Access (ZTNA) is a cybersecurity solution that embodies the principle of "never trust, always verify." It's part of the broader Zero Trust security model, which assumes that threats 2 can exist both outside and inside the network. ZTNA specifically focuses on controlling access to network resources based on strict identity verification and context-aware policies.

# 

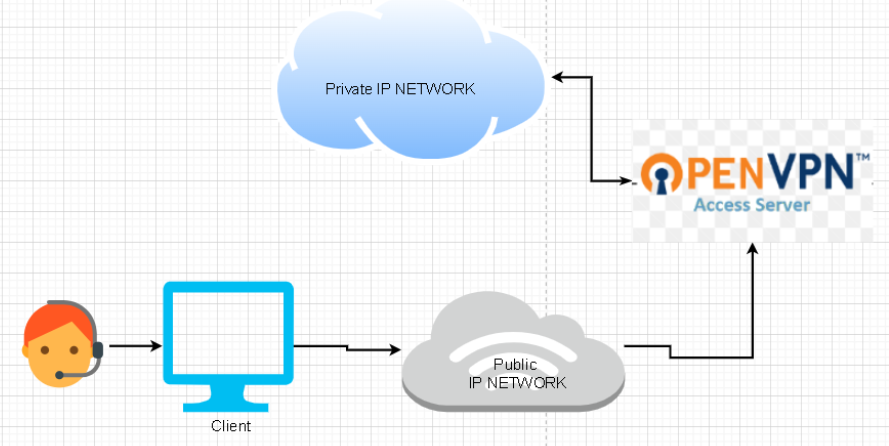
# **Scenario**

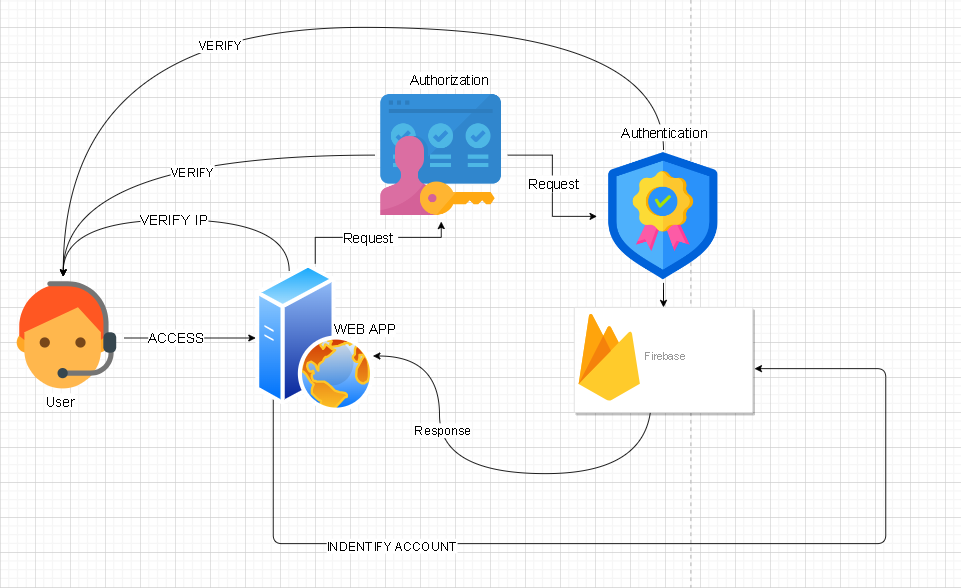
Most organizations already have some elements of zero trust in their enterprise infrastructure or are on their way through implementation of information security and resiliency policies and best practices. Here is one example deployment scenarios that lend themselves readily to a zero trust architecture:

An enterprise with a single headquarters and one or more geographically dispersed locations that are not joined by an enterprise-owned physical network connection. Employees at the remote location may not have a full enterprise owned local network but still need to access enterprise resources to perform their tasks. In this case, Enterprises may wish to grant access to some resources but deny access or restrict actions to more sensitive resources (e.g., HR database).



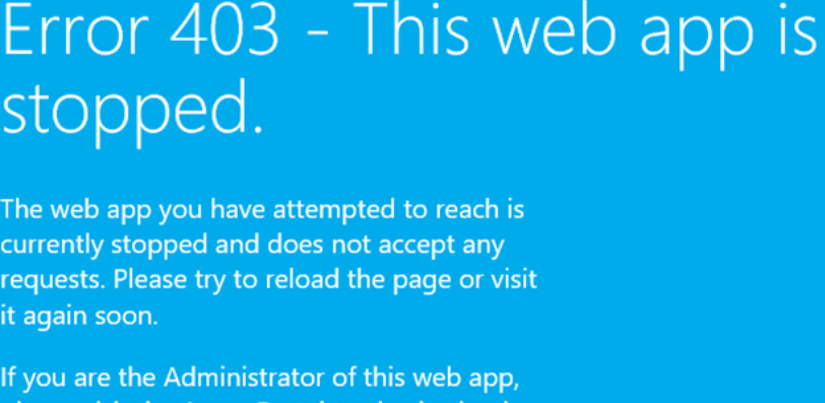
1. **Solution**
2. Architect





**Firstly**, for users to access the data, they need to connect to a private network through OpenVPN.Only then can users access the website and request resources from the server.

If not it will be like this :



**Following that**, if users want to access our resources such as images and videos, they still need to undergo authentication and authorization steps to verify permissions and identity. But at first we need a account to access.

The information about it will be encrypt and storage in local storage

In this case :

After all of that, it not enough to deploy “ zero trust “

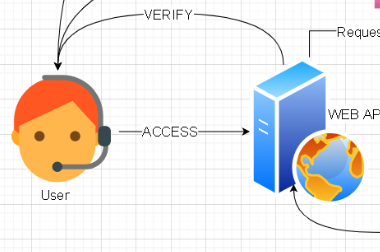
After the aforementioned steps, we perform additional verification to ensure the security of both users and resources. We utilize session tokens with expiration periods; these tokens are continuously encrypted and change every time a user clicks. If a user clicks on the login token, it is altered to prevent session hijacking from third parties or middleware. Additionally, we use tokens to record user actions on the website. **BUT** we also keep a record of each user's activities and interactions with the system on our web platform



However, there are still some aspects that have not been fully completed in the demo below.

We have only draw architecture :

* For session tokens :

When a user accesses the system, if the server verifies that the account exists on Firebase, it will create a login session and record it in the system to allow user access. The content of the session before encryption will include the user's name and permissions.

This session will have a 5-minute expiration and will automatically be deleted if exceeded. The time will follow the real-time network to prevent attacks by adjusting the local clock.

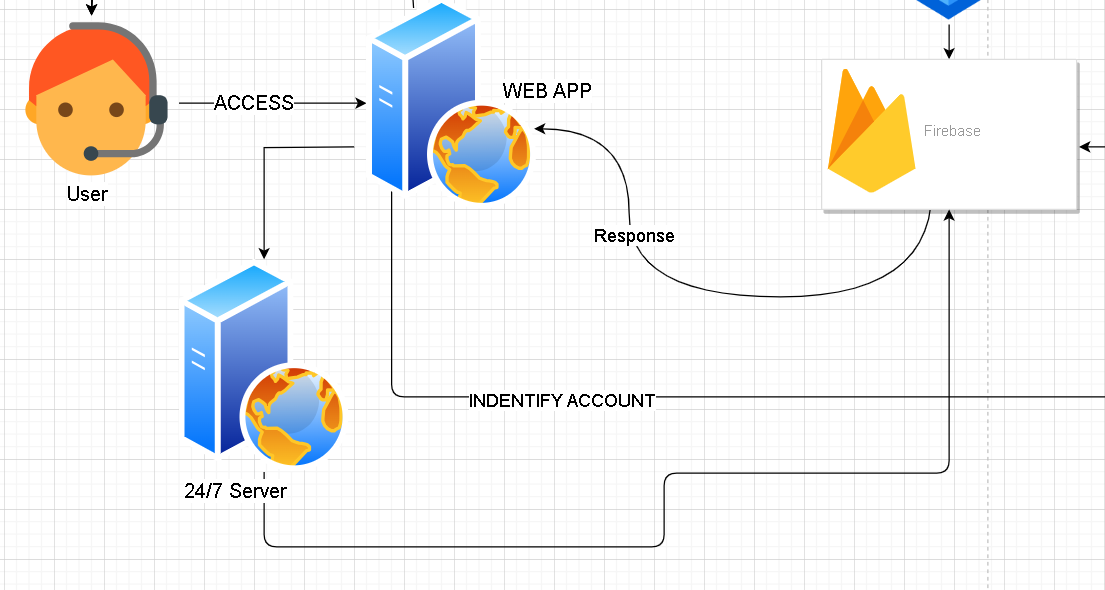
During this 5-minute period, each time the user continuously double-clicks the mouse, the login session will be altered, ensuring safety for the user.

**Simultaneously,** we will use RSA256 encryption for this session. The private key will be generated using the timestamp when the user logs in, and the public key will be the username plus their permissions.

**However**, there is no need to encrypt the data on the local storage. All necessary security operations can be performed on our backend. On local storage or session storage, we only store a randomly generated numeric value, which serves as a verification link between the user and the database. This value must be securely written to Firebase.

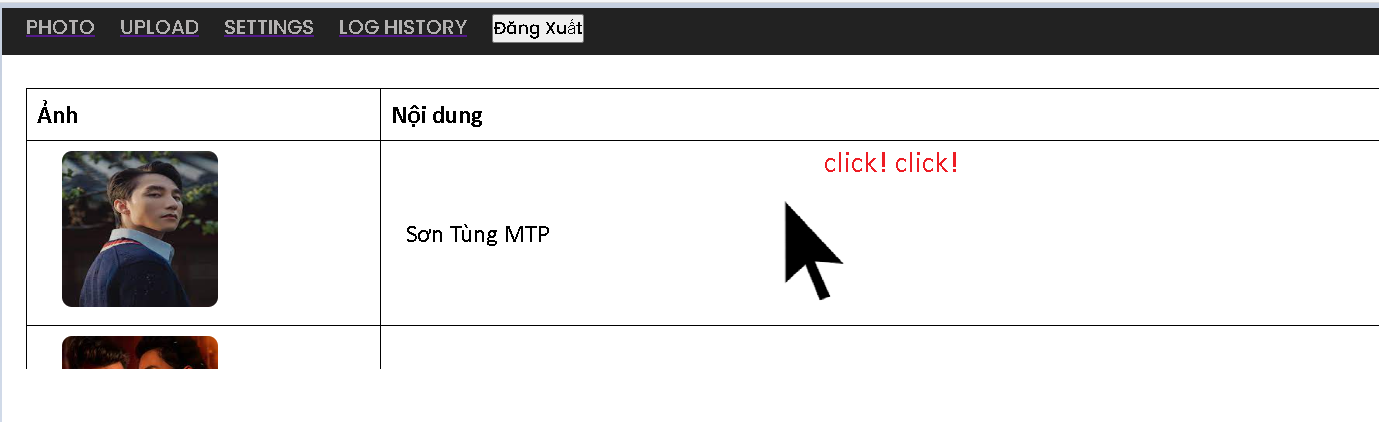
Certainly! When users log out before the expiration time, the login session will also be deleted

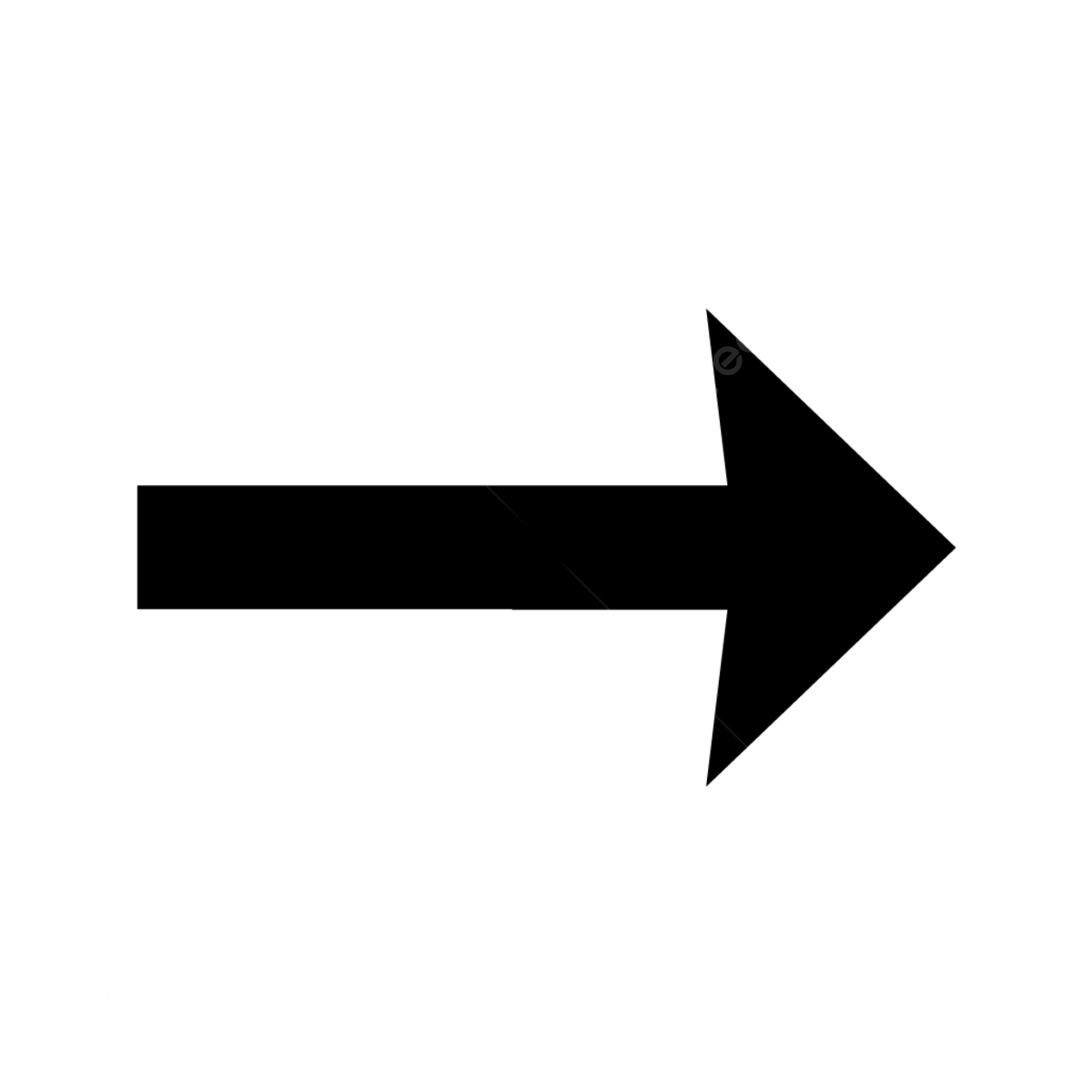
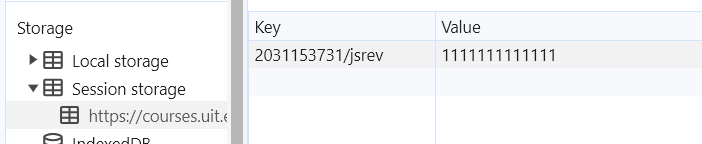
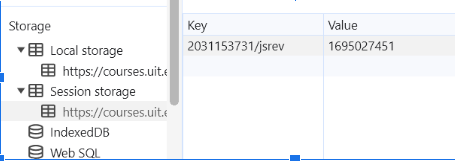
**In this case I propose use second server support for that**



At times, we may face security threats from third parties. In the event that the web app used by users fails to adequately secure our resources, having a dedicated 24/7 server solely responsible for managing tokens and sessions can significantly enhance the effectiveness of securing both our resources and user accounts

Scenario :

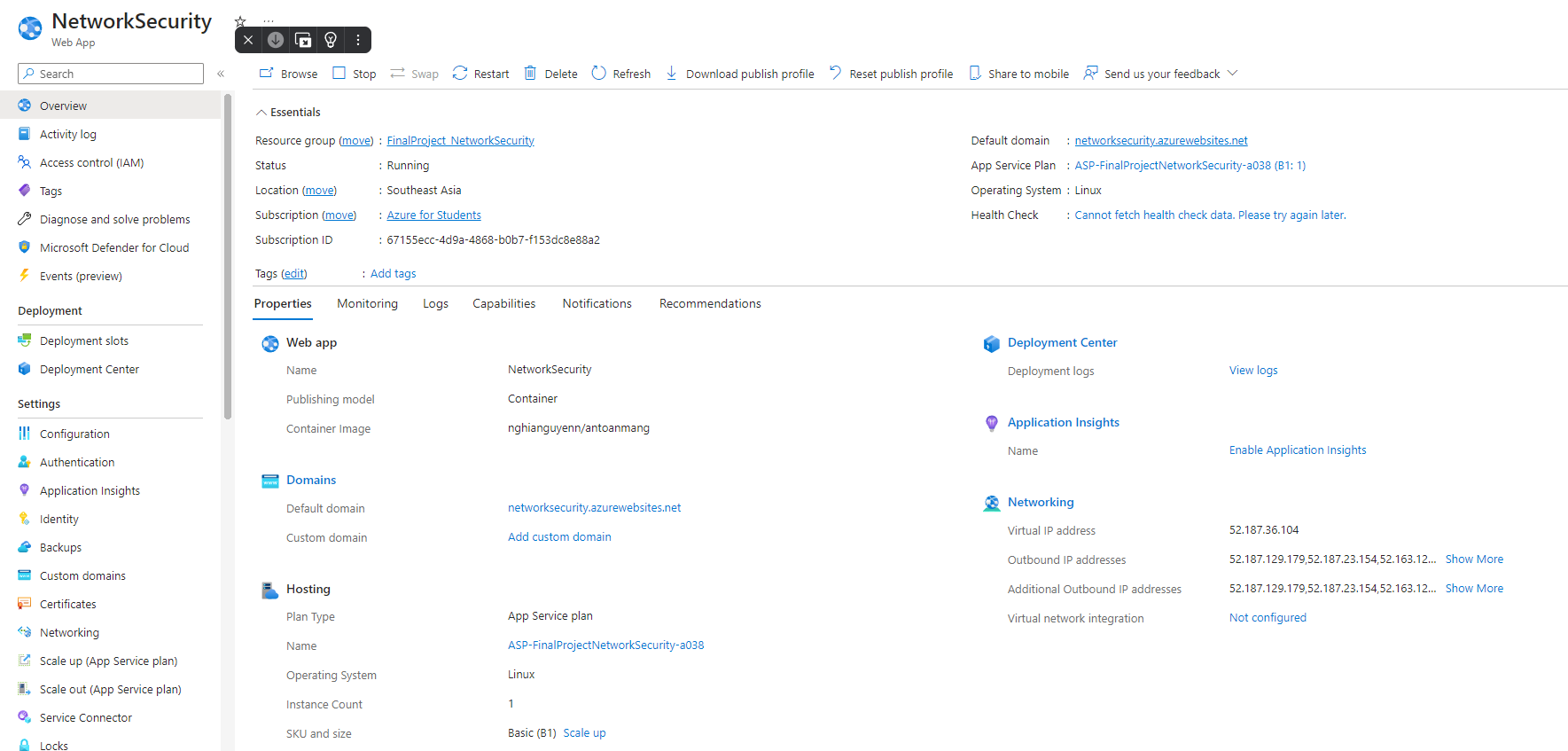




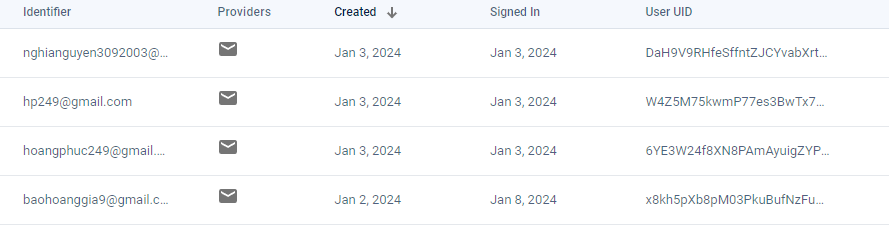
# VIDEO DEMO : [DEMO ZTNA](https://drive.google.com/drive/folders/1429XxKBKykrq_JAF8XUcaaC772Qh6HRW?usp=sharing)

In this case we use :

* Service of azure to host Web, and a public VPN



* Database : Firebase ( firebase support any service about protect our account like two step factor authentication,store resources)



1. **SECURITY GOALS**

* User Authentication
* Role-Based Access Control
* Reauthentication Before Certain Actions
* Secure Remote Access
* Zero Trust

1. **References**

NIST Special Publication 800-207 Zero Trust Architecture