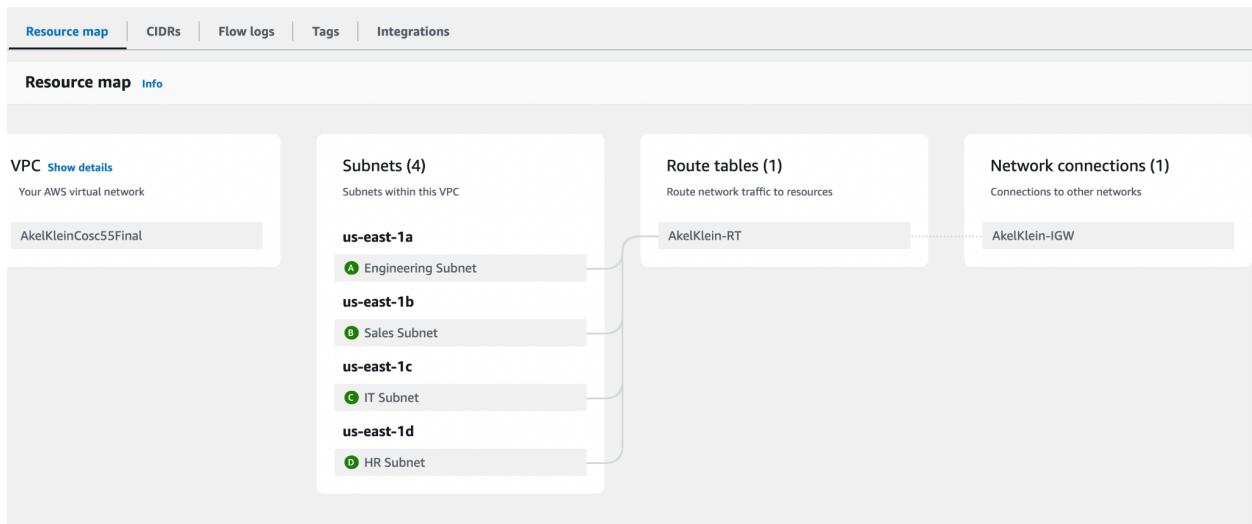


Task 1: Detailed Documentation of Cloud Security Solution

Process, Setup, and Configuration

1. Create VPC
2. Create 4 different subnets within VPC (each being from a separate availability zone to allow for use of ALB)
 - a. Engineering Subnet
 - b. Sales Subnet
 - c. IT Subnet
 - d. HR Subnet
3. Create IGW to allow for access to internet from VPC
4. Create Routing table to connect subnets



5. Initiate 4 instances → each connected to respective subnets

Instances (4) Info								
		Last updated less than a minute ago		Actions		Launch instances		
		Running						
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
Sales Instance	i-02ce11c1977b7a397	Running	t2.small	2/2 checks passed	View alarms +	us-east-1b	-	
IT Instance	i-0c0216858dfce4aa3	Running	t2.small	2/2 checks passed	View alarms +	us-east-1c	-	
Engineering In...	i-00a8e5f8a01b06881	Running	t2.small	2/2 checks passed	View alarms +	us-east-1a	-	
HR Instance	i-0b9092f8f03db82ac	Running	t2.small	2/2 checks passed	View alarms +	us-east-1d	-	

- a. Engineering Instance (Ubuntu t2.small)
- b. Sales Instance (Ubuntu t2.small)
- c. IT Instance (Ubuntu t2.small)
- d. HR Instance (Ubuntu t2.small)

6. Register a GoDaddy Domain (to be used as central login console)

The screenshot shows a list of domains under the heading "Domains". There is one item listed: "warrenkleinowenakelcosc55.com". Below the domain name, it says "Protection Plan: None" and has a link "Upgrade Protection". To the right of the domain name are two buttons: "DNS" and "Manage". At the top right of the list area is a "Manage All" button with a right-pointing arrow.

7. Create a wildcard SSL/TLS certificate in order to allow for encryption of subdomains
(Source:

<https://medium.com/@samuelnnanna71/a-guide-to-obtaining-a-public-ssl-certificate-for-your-godaddy-domain-b6869c50f625>

The screenshot shows the "Certificate status" section of the ACM console. It displays a table with columns: Identifier, ARN, Type, Status, and Renewal status. The identifier is "42a75b8e-5ae3-4205-8390-240460958960", the ARN is "arn:aws:acm:us-east-1:695402586984:certificate/42a75b8e-5ae3-4205-8390-240460958960", the type is "Amazon Issued", and the status is "Issued".

Below this, there is a table titled "Domains (1)". It has columns: Domain, Status, Renewal status, Type, and CNAME name. The domain listed is ".warrenkleinowenakelcosc55.com" with a status of "Success", renewal status of "-", type of "CNAME", and a CNAME value of "_776d307b55cdde2ab0fc582e6ea3663.warrenklein".

At the bottom, there is a note about CNAME records: "CNAME records are a type of subdomain, or alias, that points to another domain name." There is also a form to edit a CNAME record, with fields for Type (set to CNAME), Name (set to "_776d307b55cdde2ab0fc582e6ea3663.warrenklein"), Value (set to "_e73968cdb77f6e4148812d47bbef6388.djqtssxkq.ai"), and TTL (set to "1 Hour"). Buttons for "Save" and "Close" are at the bottom right.

8. Create Target Groups to be used for Application Load Balancer (Use HTTP protocol as cannot connect directly to port 443)

Target groups (4) Info						
	Name	ARN	Port	Protocol	Target type	Load balancer
<input type="checkbox"/>	engineering-tg	arn:aws:elasticloadbalanci...	80	HTTP	Instance	AkelKleinCosc55-AL
<input type="checkbox"/>	hr-tg	arn:aws:elasticloadbalanci...	80	HTTP	Instance	AkelKleinCosc55-AL
<input type="checkbox"/>	it-tg	arn:aws:elasticloadbalanci...	80	HTTP	Instance	AkelKleinCosc55-AL
<input type="checkbox"/>	sales-tg	arn:aws:elasticloadbalanci...	80	HTTP	Instance	AkelKleinCosc55-AL

9. Link each Target Group to an EC2 instance

Targets	Monitoring	Health checks	Attributes	Tags
Registered targets (1) Info				
			Anomaly mitigation: Not applicable	C Deregister Register targets
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.				
Instance ID	Name	Port	Zone	Health status
i-00a8e5f8a01b06881	Engineering In...	80	us-east-1a	Healthy - August 24...

10. Create Application Load Balancer (Purpose: to distribute incoming HTTP and HTTPS traffic across multiple targets, providing high availability, scalability, and advanced routing capabilities for web applications deployed on aws) (Source: [easydeploy.io](#)) → Ensure that ALB is using the correct VPC and availability zones of all subnets are selected →

see below

AkelKleinCosc55-ALB			
▼ Details			
Load balancer type Application	Status Active	VPC vpc-0bb5564ce24f523c6	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-0029c400bd6356806 us-east-1c (use1-az1) subnet-0327750a3f3fc46a4 us-east-1b (use1-az6) subnet-067da017371922dd4 us-east-1d (use1-az2) subnet-01cabab8dd7a2c0187 us-east-1a (use1-az4)	Date created August 24, 2024, 18:07 (UTC-04:00)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:695402586984:loadbalancer/app/AkelKleinCosc55-ALB/e4ea399ce5eed8b5	DNS name Info AkelKleinCosc55-ALB-629858546.us-east-1.elb.amazonaws.com (A Record)		

11. Using GoDaddy.com create subdomains such that subdomains of root can be redirected to EC2 instances through the ALB (CNAME: name = desired subdomain, value = ALB)
→ see example below of engineering subdomain creation

(source:

<https://montanawong.medium.com/how-to-point-your-custom-domain-to-an-aws-load-balancer-51dc2eb6d84c>)

CNAME records are a type of subdomain, or alias, that points to another domain name.

Type *	Name *	Value *	TTL
CNAME	engineering	akelkleincosc55-alb-629858546.us-east-1.elb.amazonaws.com	1 Hour

Save **Close**

12. Go back to the ALB and create two listeners, 1 for port 80 (HTTP) and 1 for port 443 (HTTPS)

Listeners and rules (2) Info						
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.						
<input type="text"/> Filter listeners						
Protocol:Port	Default action	Rules	ARN	Security policy	Def.	
<input type="checkbox"/> HTTP:80	Forward to target group <ul style="list-style-type: none">engineering-tg: 1 (100%)Target group stickiness: Off	3 rules	<input type="checkbox"/> ARN	Not applicable		
<input type="checkbox"/> HTTPS:443	Forward to target group <ul style="list-style-type: none">engineering-tg: 1 (100%)Target group stickiness: Off	5 rules	<input type="checkbox"/> ARN	ELBSecurityPolicy-TLS13-1-2-... *.well-known		

13. Rules for Port 80: redirect all traffic to HTTPS → allow for secure connection (for condition select **Path**: enter /* to allow for all, and the action select Redirect to HTTPS, make this rule the highest priority)

HTTP:80 [Info](#)

▼ Details
A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol:Port HTTP:80	Load balancer AkelKleinCosc55-ALB	Default actions Forward to target group <ul style="list-style-type: none">engineering-tg: 1 (100%)Target group stickiness: Off
Listener ARN arn:aws:elasticloadbalancing:us-east-1:695402586984:listener/app/AkelKleinCosc55-ALB/e4ea399ce5eed8b5/00aba1d93a15e014		

Rules [Tags](#)

Listener rules (3) [Info](#)

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

<input type="checkbox"/>	Name tag	Priority	Conditions (If)	Actions (Then)	ARN
<input type="checkbox"/>	redirectRule	2	Path Pattern is /*	Redirect to HTTPS://#[host]:443/#[path]?#[query] • Status code: HTTP_301	ARN

14. Rules for Port 443, for each of the target groups, set conditions for hostname to redirect to the various subdomains: ex sales.warrenkleinowenakelcosc.55 → aligns with subdomain created on GoDaddy, connect the rule to the target group created above → as a result the traffic from HTTP will be redirected to HTTPS as per the rule above which will in turn create a secure connection for each of the subdomains

Listener rules (5) [Info](#)

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

<input type="checkbox"/>	Name tag	Priority	Conditions (If)	Actions (Then)	ARN
<input type="checkbox"/>	engineRule	1	HTTP Host Header is engineering.warrenkleinowenakelcosc55.com	Forward to target group <ul style="list-style-type: none">engineering-tg: 1 (100%)Target group stickiness: Off	ARN
<input type="checkbox"/>	salesRule	2	HTTP Host Header is sales.warrenkleinowenakelcosc55.com	Forward to target group <ul style="list-style-type: none">sales-tg: 1 (100%)Target group stickiness: Off	ARN
<input type="checkbox"/>	itRule	3	HTTP Host Header is it.warrenkleinowenakelcosc55.com	Forward to target group <ul style="list-style-type: none">it-tg: 1 (100%)Target group stickiness: Off	ARN
<input type="checkbox"/>	hrRule	4	HTTP Host Header is hr.warrenkleinowenakelcosc55.com	Forward to target group <ul style="list-style-type: none">hr-tg: 1 (100%)Target group stickiness: Off	ARN

15. Look up website and verify that connection is secure

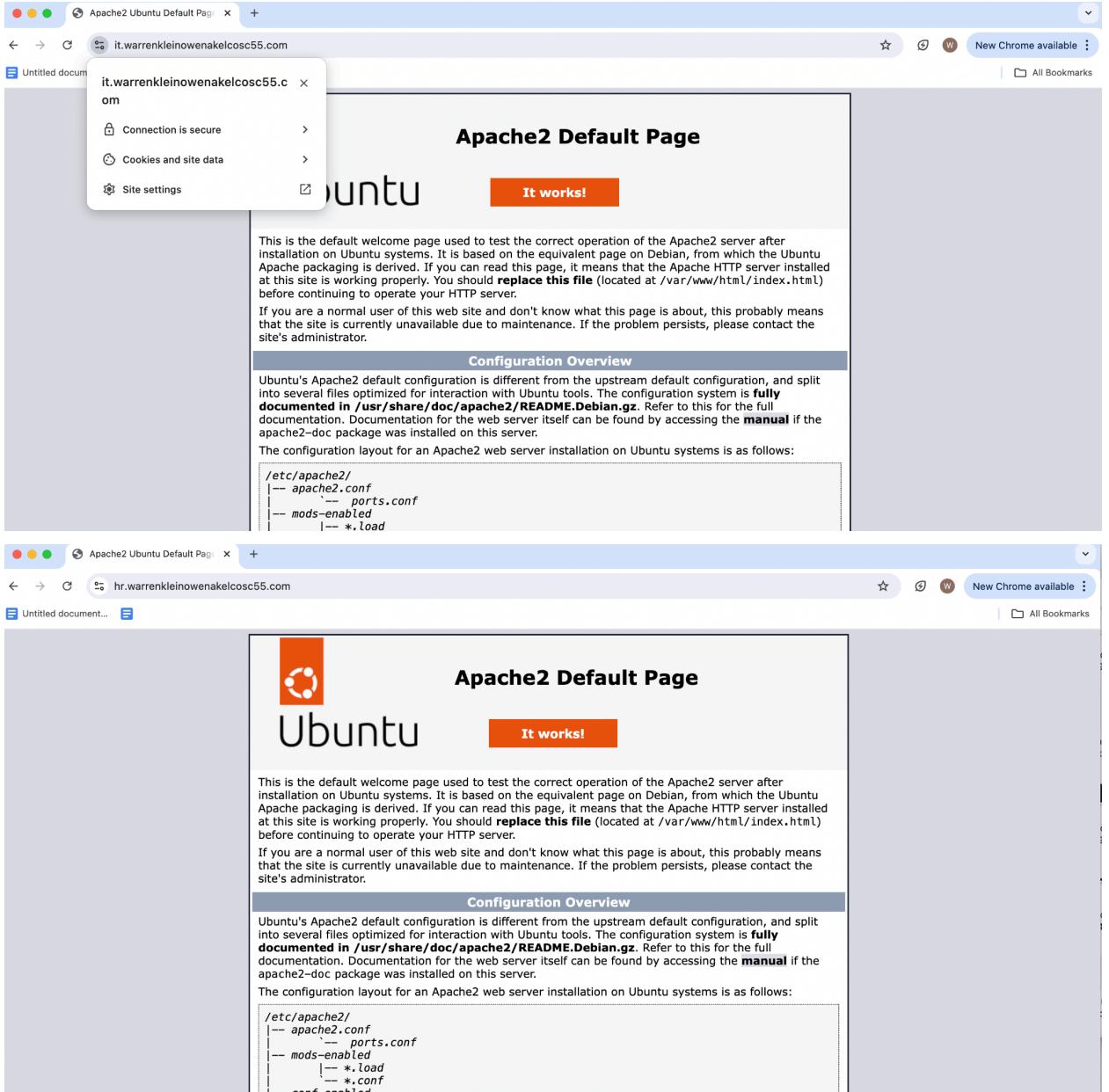
The image contains two screenshots of a web browser window, likely Google Chrome, displaying the Apache2 Default Page for two different websites: `sales.warrenkleinonenakelcosc55.com` and `engineering.warrenkleinonenakelcosc55.com`.

Screenshot 1: sales.warrenkleinonenakelcosc55.com

The browser title bar shows "Apache2 Ubuntu Default Page". The address bar shows "sales.warrenkleinonenakelcosc55.com". A context menu is open over the page content, showing options: "Connection is secure", "Cookies and site data", and "Site settings". The main content area displays the "Apache2 Default Page" with the text "It works!". Below this, there is a "Configuration Overview" section with detailed configuration information.

Screenshot 2: engineering.warrenkleinonenakelcosc55.com

The browser title bar shows "Apache2 Ubuntu Default Page". The address bar shows "engineering.warrenkleinonenakelcosc55.com". A context menu is open over the page content, showing options: "Connection is secure", "Cookies and site data", and "Site settings". The main content area displays the "Apache2 Default Page" with the text "It works!". Below this, there is a "Configuration Overview" section with detailed configuration information.



16. Create a new subnet labeled root / home to be used as the homepage to redirect traffic based off login credentials → will forward to secured pages above (IT, HR, Engineering,

Sales)

us-east-1e

E Root Domain Subnet

17. Follow same steps as before with connecting to GoDaddy (apologies for misnamed subnet), EC2 instance, target groups, routing table, target group, and link to ALB with listener rules (makes secure)
18. Develop HTML script to be used as login screen for home domain (see login credentials and desired department)

```
GNU nano 7.2 index.html tenance for Application
<!DOCTYPE html> GRANT USAGE ON *.* TO 'warrenowen'@localhost
<html lang="en"> GRANT ALL PRIVILEGES ON 'company_info'.* TO 'warrenowen'@localhost
<head> +-----+-----+-----+-----+-----+-----+
<meta charset="UTF-8"> 3 rows in set (0.00 sec)
<meta name="viewport" content="width=device-width, initial-scale=1.0"> next to
<title>Login Page</title>
</head> Broadcast message from root@ip-10-0-3-100 (Tue 2024-08-27 20:08:42 UTC):
<body> The system will power off now!
<h2>Login</h2> Downloads -- ubuntu@ip-10-0-3-100:~$ cd /var/www/html --> nano...
<form action="login.php" method="post"> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    <label for="username">Username:</label><br> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    <input type="text" id="username" name="username" required><br><br> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    <label for="password">Password:</label><br> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    <input type="password" id="password" name="password" required><br><br> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    <label for="department">Select Department:</label><br> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    <select id="department" name="department" required> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
        <option value="">--Please choose an option--</option> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
        <option value="engineering">Engineering</option> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
        <option value="it">Information Technology</option> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
        <option value="sales">Sales</option> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
        <option value="hr">Human Resources</option> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    </select><br><br> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
    Open an SSH connection to ip-10-0-3-100:~$ cd /var/www/html --> nano...
    <button type="Submit">Login</button> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
</form> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
</body> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
</html> nano: /var/www/html/index.html:1: syntax error at unexpected token: login.php
3. Run this command to ensure you can log in: chmod 755 index.html
The system will power off now!
```

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL `home.warrenkleinonenakelcosc55.com`.
- Title Bar:** Displays "Login Page".
- Content Area:** A form titled "Login". It includes:
 - A "Username:" label with an associated input field.
 - A "Password:" label with an associated input field.
 - A "Select Department:" label with a dropdown menu showing the option "--Please choose an option--".
 - A "Login" button at the bottom.

19. Connect PHP to MySQL to be used to authenticate login credentials and grant all privileges to newly created database

```

mysql> CREATE DATABASE company_info;
Query OK, 1 row affected (0.01 sec)

mysql> USE company_info;
Database changed
mysql> CREATE TABLE employees (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     username VARCHAR(50) NOT NULL,
    ->     password VARCHAR(255) NOT NULL,
    ->     department VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO employees (username, password, department) VALUES
    -> ('Warren', 'dog', 'engineering'),
    -> ('Owen', 'cat', 'hr'),
    -> ('Bennett', 'bug', 'it'),
    -> ('William', 'monkey', 'sales');
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM employees;
+----+-----+-----+-----+
| id | username | password | department |
+----+-----+-----+-----+
| 1  | Warren   | dog      | engineering |
| 2  | Owen     | cat      | hr          |
| 3  | Bennett  | bug      | it          |
| 4  | William  | monkey   | sales       |
+----+-----+-----+-----+

mysql> GRANT ALL PRIVILEGES ON company_info.* TO 'warrenowen'@'localhost' IDENTIFIED BY 'owenwarren';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to
use near 'IDENTIFIED BY 'owenwarren'' at line 1
mysql> GRANT ALL PRIVILEGES ON company_info.* TO 'warrenowen'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'warrenowen'@'localhost';
+-----+
| Grants for warrenowen@localhost |
+-----+
| GRANT USAGE ON *.* TO 'warrenowen'@'localhost' |
| GRANT ALL PRIVILEGES ON `company_info`.* TO 'warrenowen'@'localhost' |
+-----+
2 rows in set (0.00 sec)

```

20. Develop PHP script that accesses database, authenticates credentials and redirects to web pages

```
GNU nano 7.2
#!/usr/bin/php
//Error reporting
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

//Database login
$Username='warrenkenlein';
$password='owenwarren';
$database='company_info';
$host='localhost';
$table = 'employees';

//Retrieve user login information for Application
$user_username=$_POST['username'];
$user_password=$_POST['password'];
$user_department=$_POST['department'];

//Create MySQL connection
$connection = new mysqli($host, $Username, $password, $database);

//Prepare login
$login_attempt = $connection->prepare("SELECT * FROM $table WHERE username = ? AND password = ? AND department = ?");

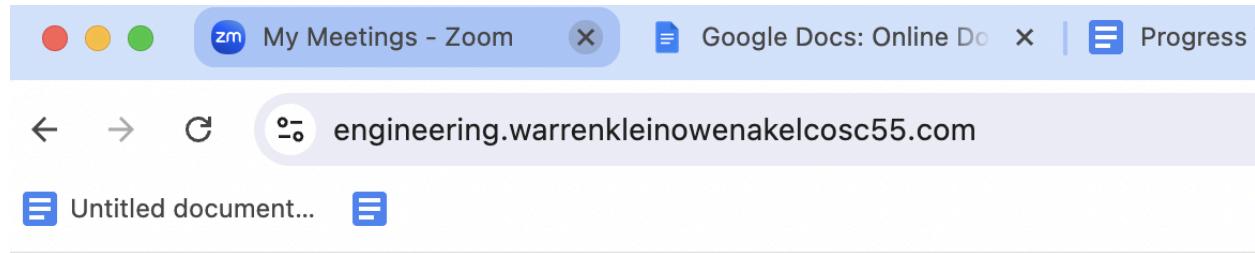
//Bind parameters
$login_attempt->bind_param('sss', $user_username, $user_password, $user_department);

//Execute login
$login_attempt->execute();
$result = $login_attempt->get_result();

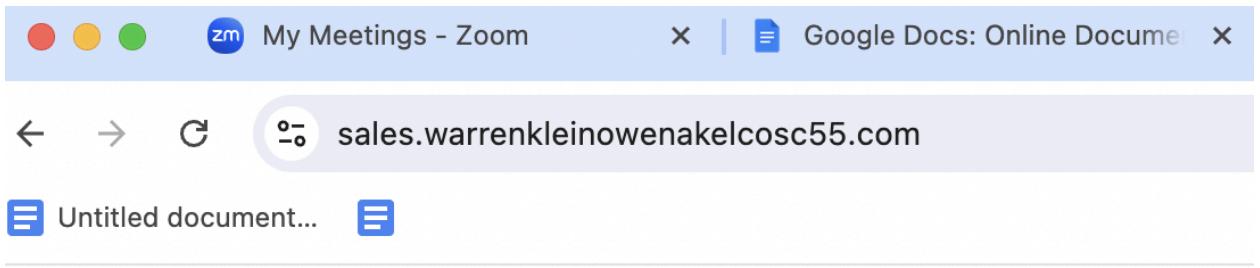
//Validate login attempt
if ($result->num_rows > 0) {
    //Login successful
} else {
    //Login failed
}

//Close connection
$connection->close();
```

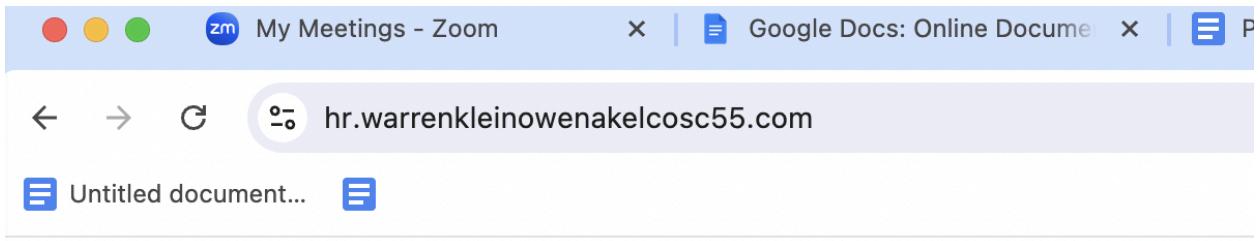
21. Create sample webpages for each of the different departments that the login page redirects to



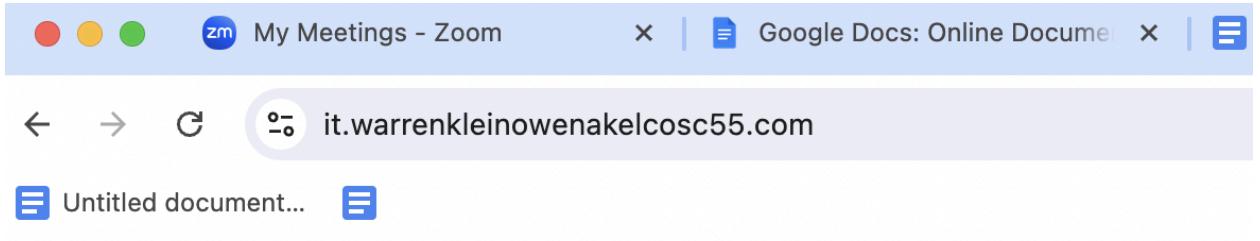
Welcome to the engineering page



Welcome to the sales page



Welcome to the HR page



Welcome to the IT page

22. Video Demo:

https://drive.google.com/file/d/1rqTRogNbPhHJdHp7OnDiwT76MguqLSmK/view?usp=drive_link

Setbacks and Challenges with initial setup:

- **Domain Registration and DNS Configuration:**
 - **Initial Setup Issues:** Faced difficulties with registering a domain through AWS Route 53, leading to the decision to use GoDaddy for domain registration.
 - **DNS Propagation Delay:** Experienced delays in DNS propagation when configuring subdomains with GoDaddy, which affected the accessibility of your subdomains.
 - **CNAME vs. A Record Confusion:** Encountered uncertainty about whether to use CNAME records or A records, particularly for the root domain and its integration with the AWS ALB.
- **ALB Listener and Rule Configuration:**
 - **Host Header Rule Configuration:** Initially struggled with configuring host header rules to route traffic from subdomains to the correct target groups through the ALB.
 - **Redirection Rule Setup:** Faced challenges in setting up HTTP to HTTPS redirection, including determining the appropriate condition and ensuring the redirection rule was prioritized correctly.
- **Health Check Failures:**
 - **Unhealthy Target Groups:** Encountered issues with target groups showing as unhealthy, likely due to incorrect health check paths, protocols, or ports, which needed troubleshooting and adjustment.
- **HTTPS and SSL/TLS Setup:**
 - **Wildcard SSL/TLS Certificate Application:** Successfully applied a wildcard SSL/TLS certificate but faced difficulties in ensuring that all HTTP traffic was correctly redirected to HTTPS, securing all subdomains.

Reasons for using ALB and horizontal integration of the departments:

Isolation of Services:

- Each department (e.g., Engineering, HR, IT) operates independently on its own EC2 instance, reducing the risk of one department's issues affecting others.

Customized Resource Allocation:

- Different departments can have EC2 instances tailored to their specific needs, optimizing performance and cost efficiency.

Simplified Management and Scaling:

- By isolating departments on separate instances, you can scale each department's resources independently based on demand.

Enhanced Security and Compliance:

- Keeping departments on separate instances allows for more granular security policies and easier compliance with data protection regulations.

Simplified Troubleshooting and Maintenance:

- Issues within one department's application can be isolated and addressed without impacting the entire organization, leading to more efficient maintenance.

Brainstorming/Troubleshooting:

