ICT171 Assignment 2

Name: Ryan Lie

Student number: 35381408

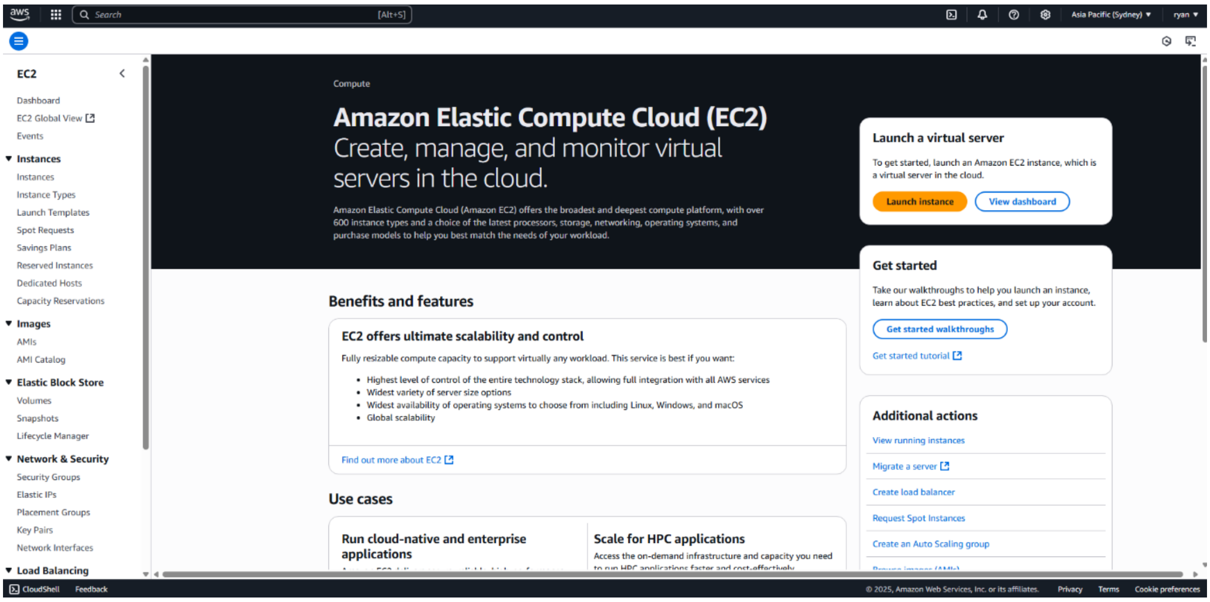
IPv4 address: 16.176.191.217

Website: ryanbooks.com http://16.176.191.217/

Video: https://youtu.be/0Mzw70M0OdQ

Part 1 – EC2 Setup and Environment Configuration

1. Launching an EC2 Instance



Compute

Amazon Elastic Compute Cloud (EC2)
Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

Launch a virtual server

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Get started

Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.

Benefits and features

EC2 offers ultimate scalability and control

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

Additional actions

Use cases

Run cloud-native and enterprise applications

Scale for HPC applications

Access the on-demand infrastructure and capacity you need

© 2025, Amazon Web Services, Inc. or its affiliates.

a. Sign into the AWS Console

Go to the AWS website and log into your account.

Navigate to the EC2 service by typing “EC2” in the search bar.

b. Start a new instance

Click on “Launch Instance”.

Choose an Amazon Machine Image (AMI)

For instance type, select t3.micro (or t2.micro). *t3.micro* is helpful if SSH fails.

c. Configure instance details

Use default configuration settings.

Set storage to 8GB unless your project requires more.

Generate a new key pair (use ED25519 for better security).

Save the key pair file (.pem) in a safe location like ~/Downloads.

d. Set up Security Group

Create a new security group (e.g., project-sec-group).

Add the following rules:

SSH (TCP port 22)

HTTP (TCP port 80)

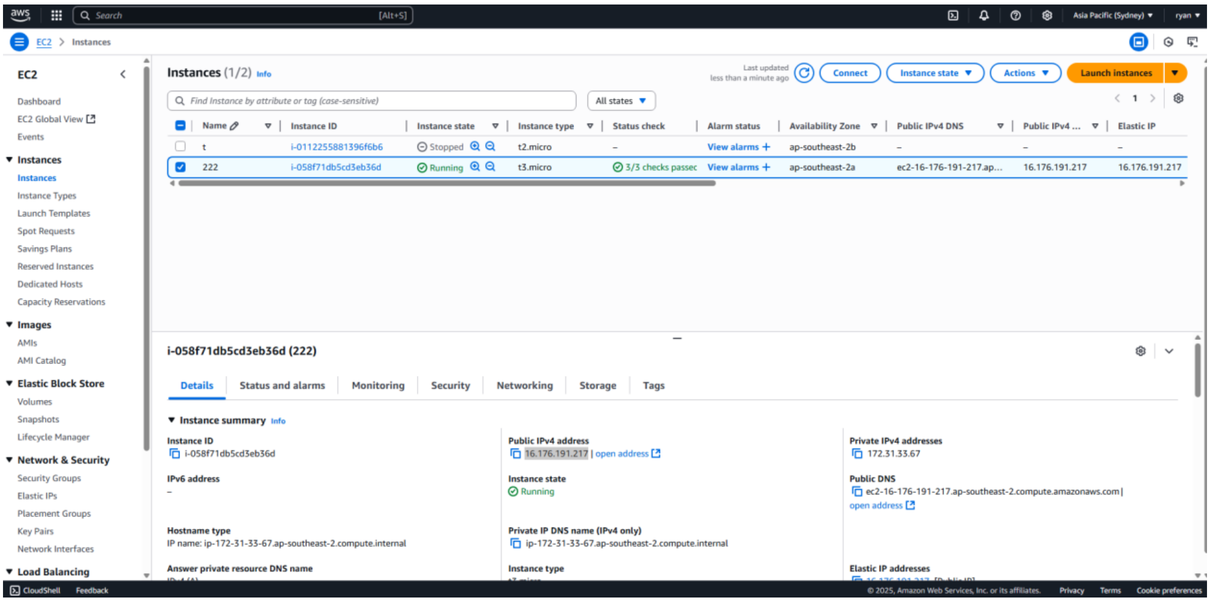
HTTPS (TCP port 443)

e. Allocate and link an Elastic IP

Go to Elastic IPs under the “Network & Security” menu.

Allocate an IP from Amazon's IPv4 pool.

Associate the Elastic IP with your new instance.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	t	i-0112255881396f6b6	Stopped	t2.micro	–	View alarms +	ap-southeast-2b	–	–	–
☑	222	i-058f71db5cd3eb36d	Running	t3.micro	3/3 checks passec	View alarms +	ap-southeast-2a	ec2-16-176-191-217.ap...	16.176.191.217	16.176.191.217

i-058f71db5cd3eb36d (222)

Instance summary

Instance ID
i-058f71db5cd3eb36d

Public IPv4 address
16.176.191.217 | open address

Private IPv4 addresses
172.31.33.67

IPv6 address
–

Instance state
Running

Public DNS
ec2-16-176-191-217.ap-southeast-2.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-33-67.ap-southeast-2.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-33-67.ap-southeast-2.compute.internal

Answer private resource DNS name

Instance type

Elastic IP addresses

Choose connect to get into EC2 Terminal top right

To change script

Nano /home/ubuntu/testscript

3. Script (Github script)

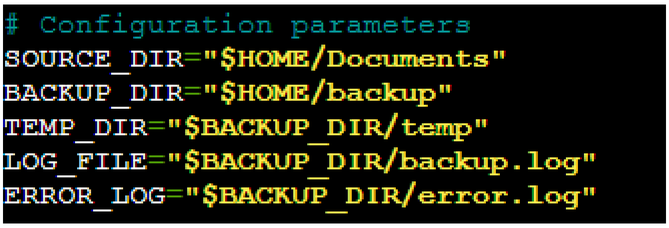
1. This Bash script automatically backs up the contents of a specific local folder (~/Documents) by:

- Copying files into a temporary directory
- Compressing them into a .tar.gz archive
- Logging each step and any errors

- Optionally, the final archive can be uploaded to a remote EC2 server (Elastic IP: 16.176.191.217)

It ensures folder creation, permission checks, and graceful failure handling.

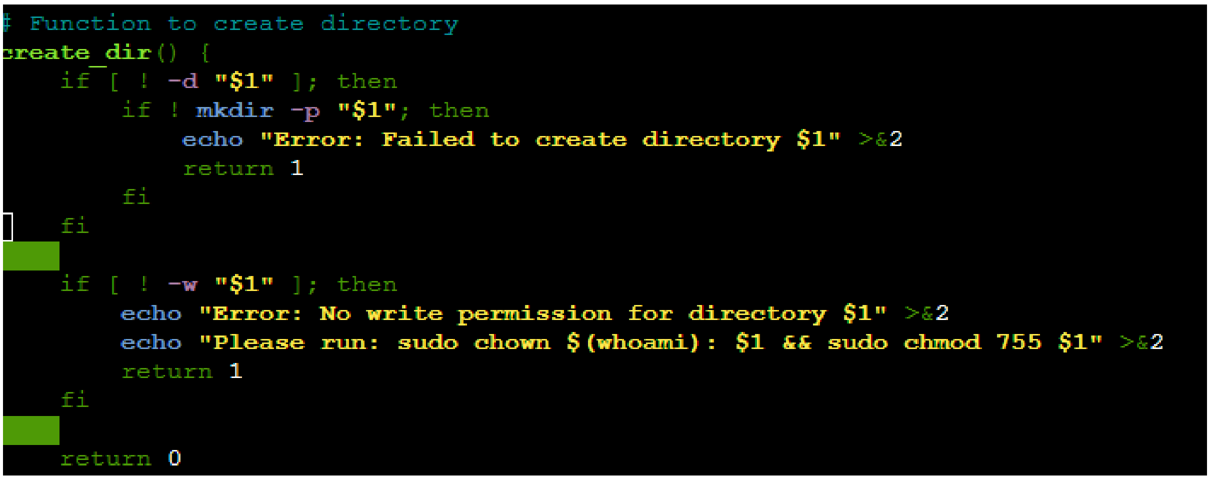
2. Define Key Paths



```
# Configuration parameters
SOURCE_DIR="$HOME/Documents"
BACKUP_DIR="$HOME/backup"
TEMP_DIR="$BACKUP_DIR/temp"
LOG_FILE="$BACKUP_DIR/backup.log"
ERROR_LOG="$BACKUP_DIR/error.log"
```

- SOURCE_DIR: Folder to be backed up
- BACKUP_DIR: Where final backups will be stored
- TEMP_DIR: Used for staging before compression
- LOG_FILE / ERROR_LOG: Stores logs and error messages

3. Create Required Directories Function

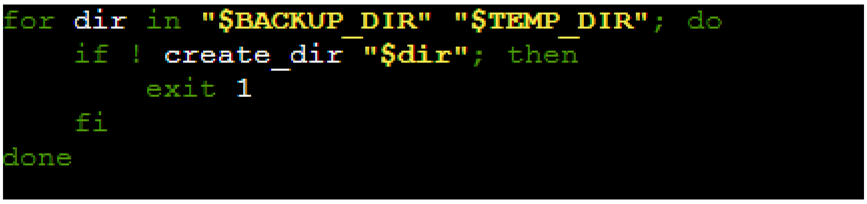


```
# Function to create directory
create_dir() {
    if [ ! -d "$1" ]; then
        if ! mkdir -p "$1"; then
            echo "Error: Failed to create directory $1" >&2
            return 1
        fi
    fi

    if [ ! -w "$1" ]; then
        echo "Error: No write permission for directory $1" >&2
        echo "Please run: sudo chown $(whoami): $1 && sudo chmod 755 $1" >&2
        return 1
    fi

    return 0
```

This function:

- Checks if a directory exists — creates it if not
- Verifies write permissions
- Gives clear instructions if permission is denied

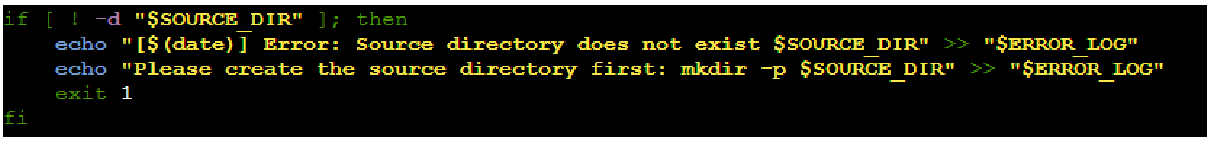
4. Setup Directories and Initialize Logs



```
for dir in "$BACKUP_DIR" "$TEMP_DIR"; do
    if ! create_dir "$dir"; then
        exit 1
    fi
done
```

- Calls the create_dir function for each required folder
- Exits if any folder can't be made or accessed
- Creates log files for process tracking

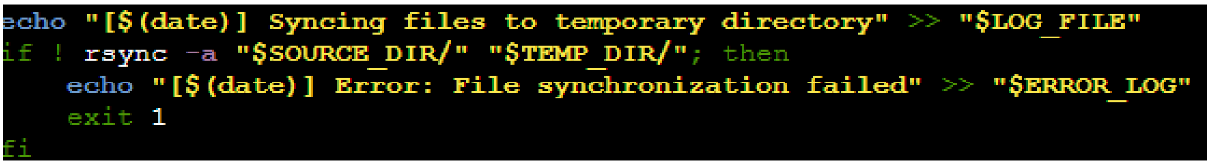
5. Check Source Folder



```
if [ ! -d "$SOURCE_DIR" ]; then
    echo "[$(date)] Error: Source directory does not exist $SOURCE_DIR" >> "$ERROR_LOG"
    echo "Please create the source directory first: mkdir -p $SOURCE_DIR" >> "$ERROR_LOG"
    exit 1
fi
```

- Prevents the backup from running if the source folder is missing
- Logs the error and gives user instructions

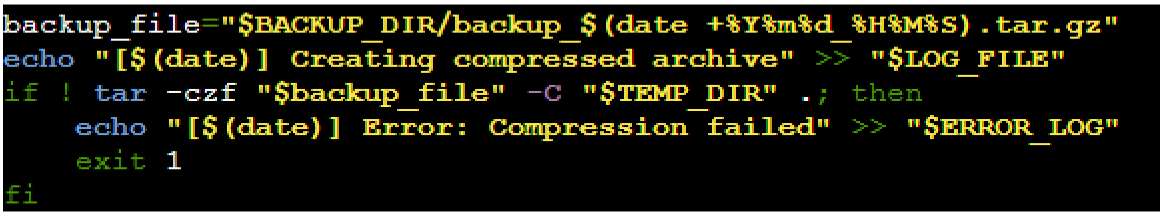
6. File Synchronization with rsync



```
echo "[$(date)] Syncing files to temporary directory" >> "$LOG_FILE"
if ! rsync -a "$SOURCE_DIR/" "$TEMP_DIR/"; then
    echo "[$(date)] Error: File synchronization failed" >> "$ERROR_LOG"
    exit 1
fi
```

- Efficiently copies files from Documents to the temp directory
- Handles symbolic links, file timestamps, and permissions

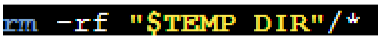
7. Create Timestamped Archive



```
backup_file="$BACKUP_DIR/backup_$(date +%Y%m%d_%H%M%S).tar.gz"
echo "[$(date)] Creating compressed archive" >> "$LOG_FILE"
if ! tar -czf "$backup_file" -C "$TEMP_DIR" .; then
    echo "[$(date)] Error: Compression failed" >> "$ERROR_LOG"
    exit 1
fi
```

- Creates a .tar.gz file named with the date and time
- Ensures backups don’t overwrite each other

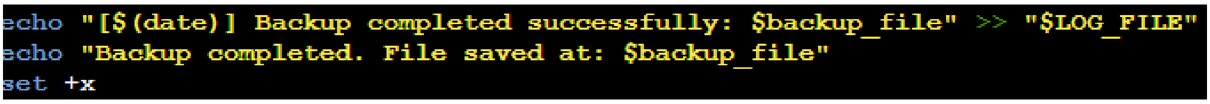
8. Clean Up Temporary Files



```
rm -rf "$TEMP_DIR"/*
```

- Clears out the temporary folder after compression
- Keeps things clean for the next backup

9. Completion Logging



```
echo "[$(date)] Backup completed successfully: $backup_file" >> "$LOG_FILE"
echo "Backup completed. File saved at: $backup_file"
set +x
```

- Logs success to both the terminal and the backup log

10. ./testscript shows its working

Enabling firewall and check status

SSH: sudo ufw allow 22/tcp

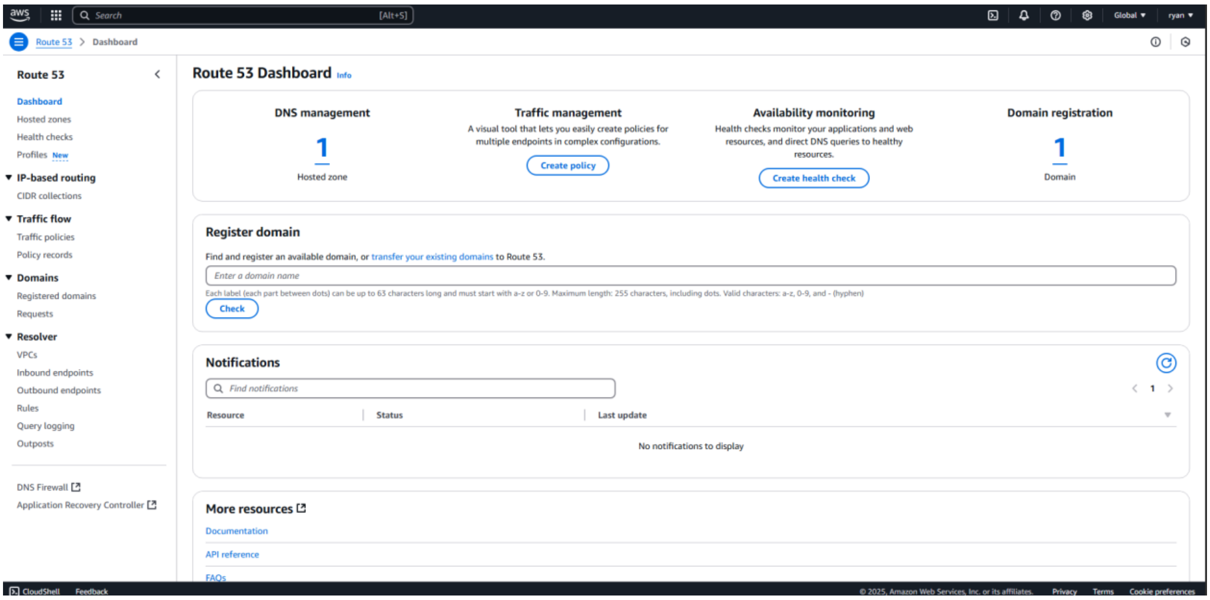
HTTP: sudo ufw allow 80/tcp

HTTPS: sudo ufw allow 443/tcp

sudo ufw enable

sudo ufw status

Part 2 – Domain and HTTPS Setup



Route 53 Dashboard Info

DNS management: 1 Hosted zone

Traffic management: A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

Availability monitoring: Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

Domain registration: 1 Domain

Register domain

Find and register an available domain, or transfer your existing domains to Route 53.

Enter a domain name

Each label (each part between dots) can be up to 63 characters long and must start with a-z or 0-9. Maximum length: 255 characters, including dots. Valid characters: a-z, 0-9, and - (hyphen)

Notifications

Resource	Status	Last update

No notifications to display

More resources

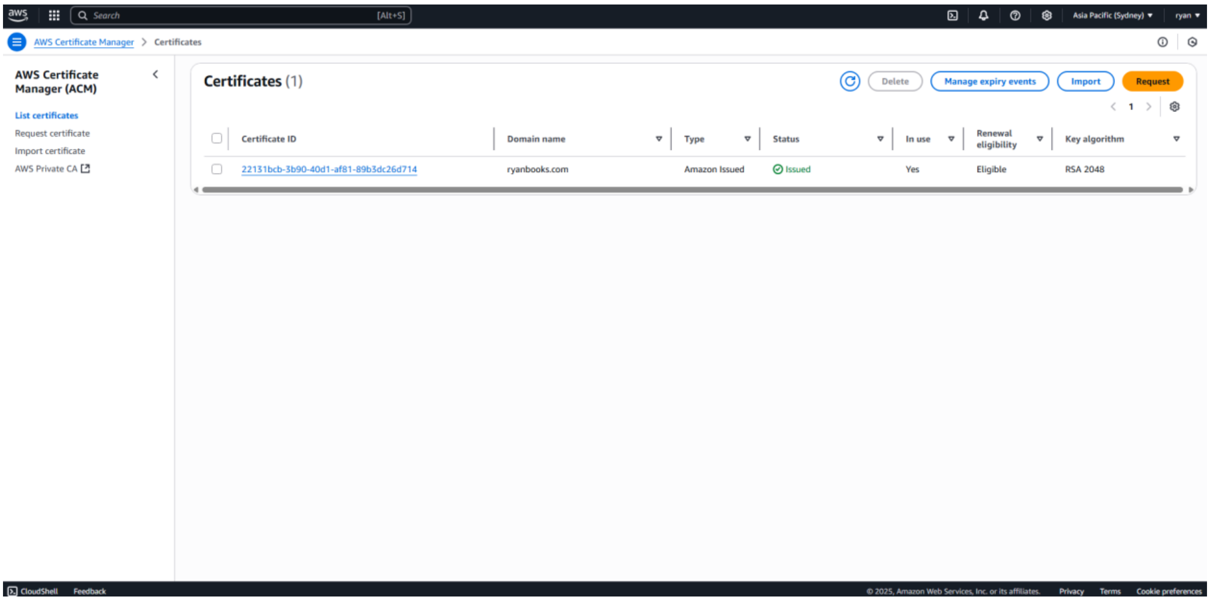
© 2025, Amazon Web Services, Inc. or its affiliates.

1. Register a Domain with Route 53

Open Route 53, go to “Dashboard”, and search for a domain under “Register domain”.

Purchase and it will appear under “Hosted Zones”.

2. Request SSL Certificate



Certificates (1)

□	Certificate ID	Domain name	Type	Status	In use	Renewal eligibility	Key algorithm
□	22131bcb-3b90-40d1-af81-89b3dc26d714	ryanbooks.com	Amazon Issued	✓ Issued	Yes	Eligible	RSA 2048

Open AWS Certificate Manager → Click “Request Certificate”.

Enter your registered domain.

After approval, it will show under “List Certificates”.

Check if CNAME is auto added in Route 53.

3. Configure HTTPS with Load Balancer

Go to EC2 → “Load Balancers” → Create Application Load Balancer

Requirements:

Add instance’s subnet

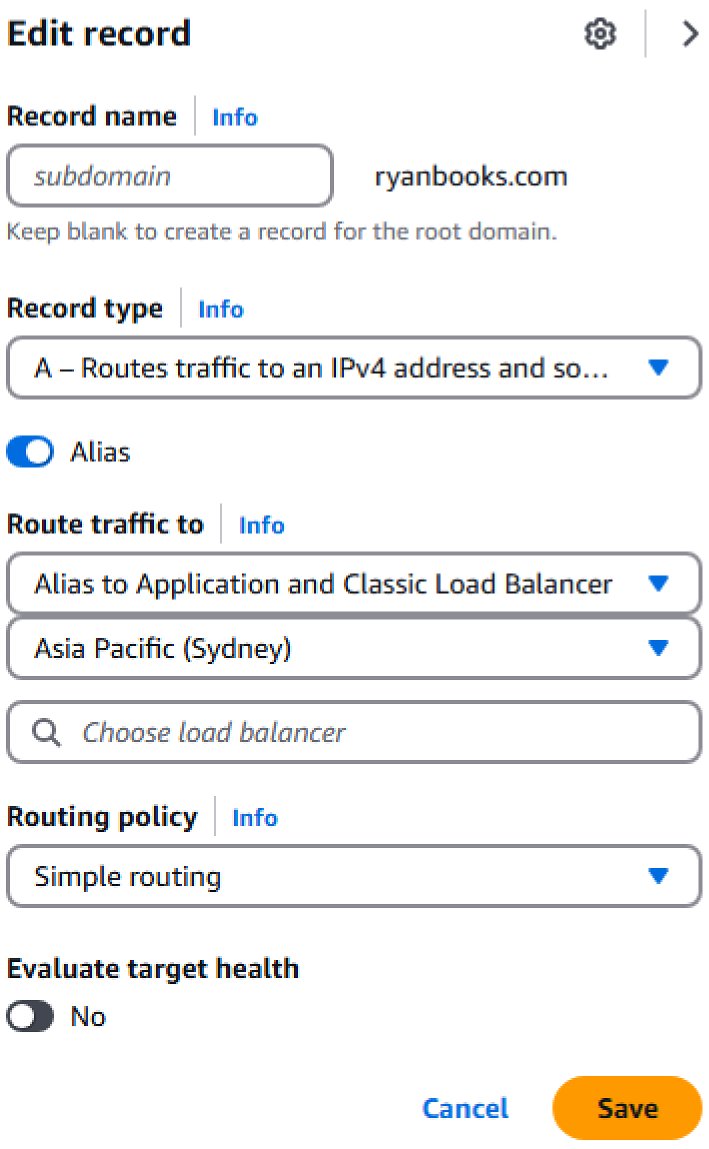
Security group must allow ports 80 and 443

Listener protocol: HTTPS → Target group: HTTP

Use previously approved SSL certificate

Launch it and ensure the status is Active

4. DNS Record Setup



Edit record

Record name: subdomain ryanbooks.com

Keep blank to create a record for the root domain.

Record type: A – Routes traffic to an IPv4 address and so...

Alias

Route traffic to: Alias to Application and Classic Load Balancer

Asia Pacific (Sydney)

Choose load balancer

Routing policy: Simple routing

Evaluate target health: No

Go to Route 53 → “Hosted Zones” → Select your domain

Create a new A Record

Enable “Alias”

Point it to the Load Balancer

Once everything is complete, your site will be accessible via HTTPS.