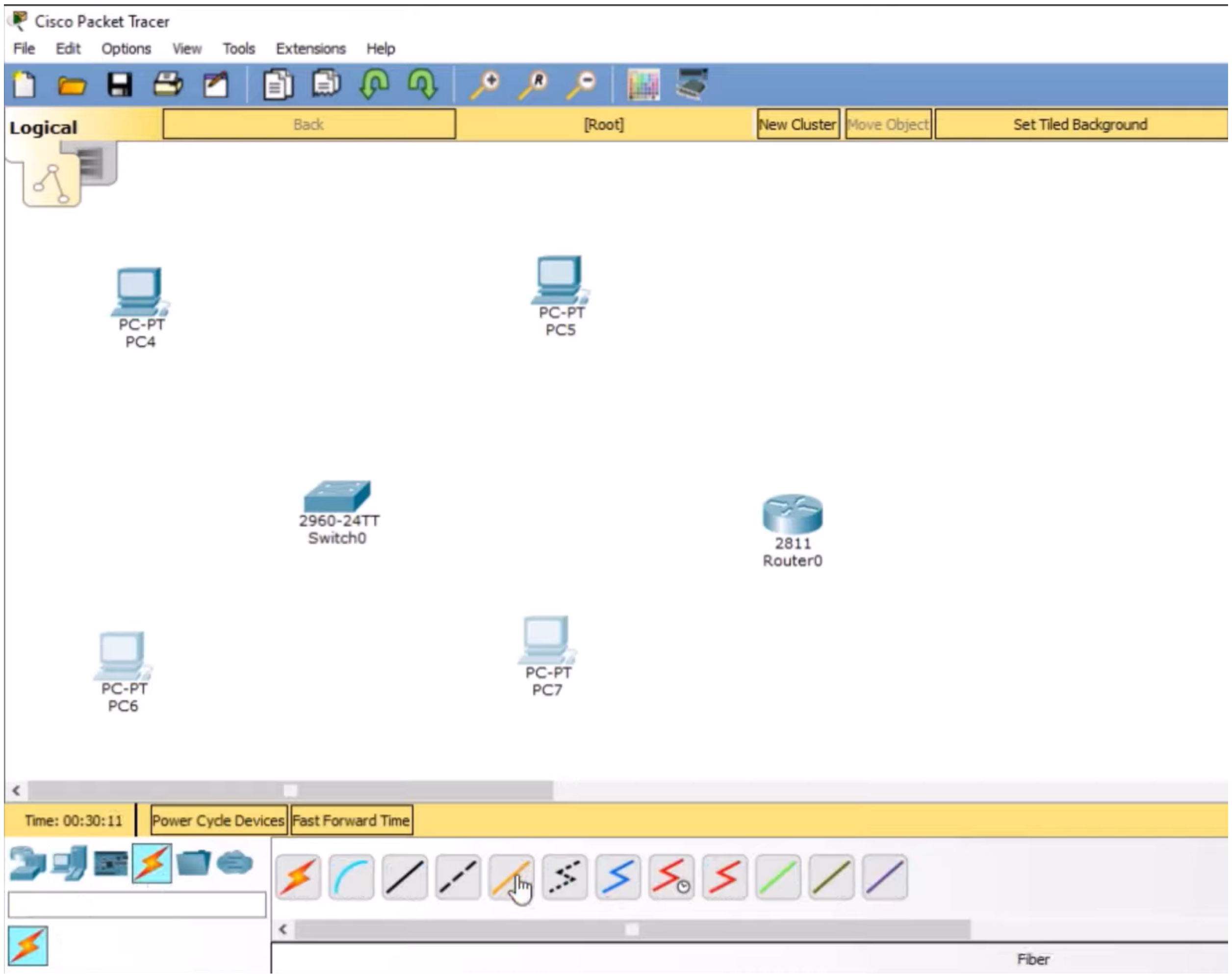
- **Cloud computing**

[1] Cisco simulator - VLAN design, Routing, Subnetting, Gateway configuration

ans.

1. Add Devices:
    - Drag and drop a switch (e.g., 2960) and several PCs onto the workspace.



2. Create VLANs:

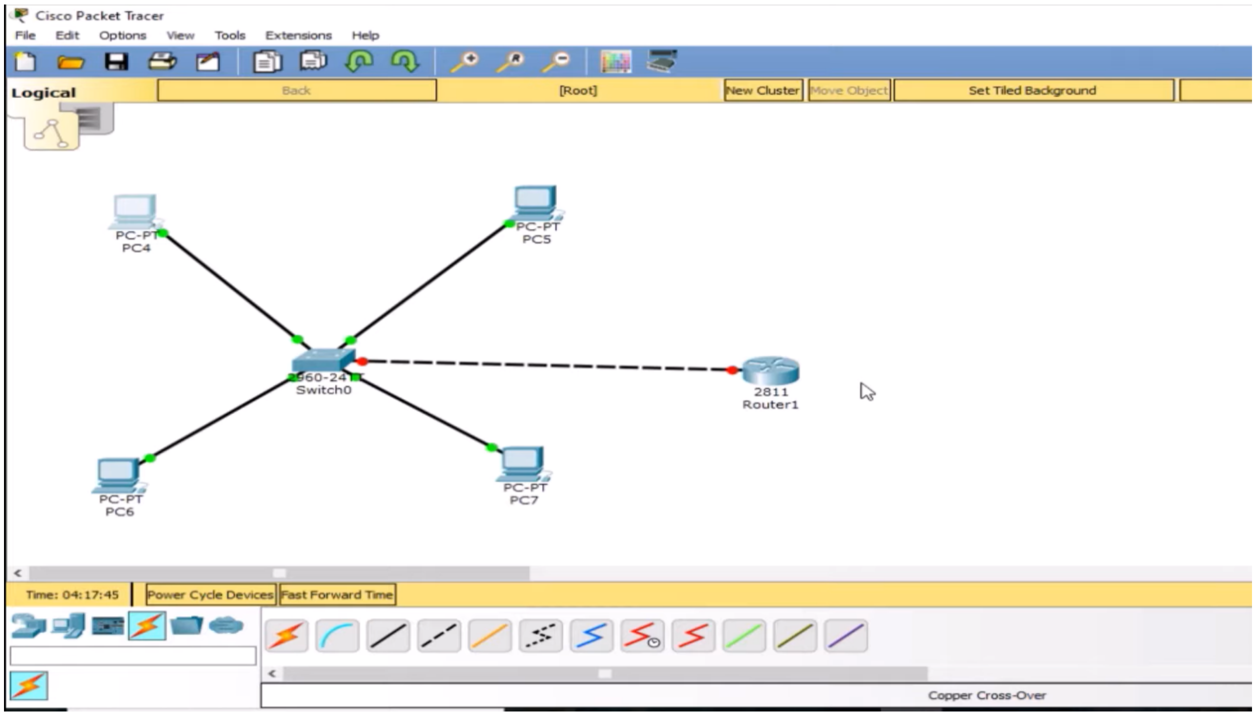
- Enter the following commands to create VLANs:

```
#enable
configure terminal
vlan 10
name VLAN10
exit
vlan 20
name VLAN20
exit#
```

3. Assign Ports to VLANs:

- Assign specific ports to each VLAN.

```
#interface range fa0/1 - 2
switchport mode access
switchport access vlan 10
exit

interface range fa0/3 - 4
switchport mode access
switchport access vlan 20
exit#
```



4. Configure Router Interfaces:

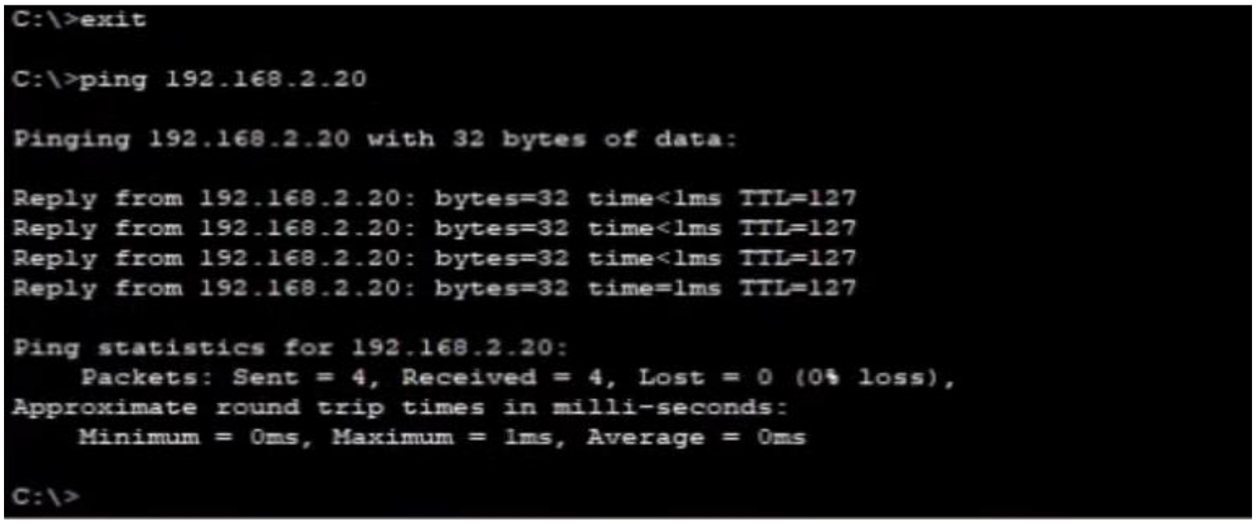
- Access the router's CLI and configure sub-interfaces for each VLAN:

```
enable
configure terminal
interface gig0/0.10
encapsulation dot1Q 10
ip address 192.168.10.1 255.255.255.0
exit

interface gig0/0.20
encapsulation dot1Q 20
ip address 192.168.20.1 255.255.255.0
exit

interface gig0/0
no shutdown
```

1. Assign IP Addresses:
   - For PCs in VLAN 10 (e.g., PC1 and PC2):PC1: IP Address: 192.168.10.2
     Subnet Mask: 255.255.255.0
   - Default Gateway: 192.168.10.1

2. For PCs in VLAN 20 (e.g., PC3 and PC4):PC3: IP Address: 192.168.10.3

   - Subnet Mask: 255.255.255.0
   - Default Gateway: 192.168.10.1

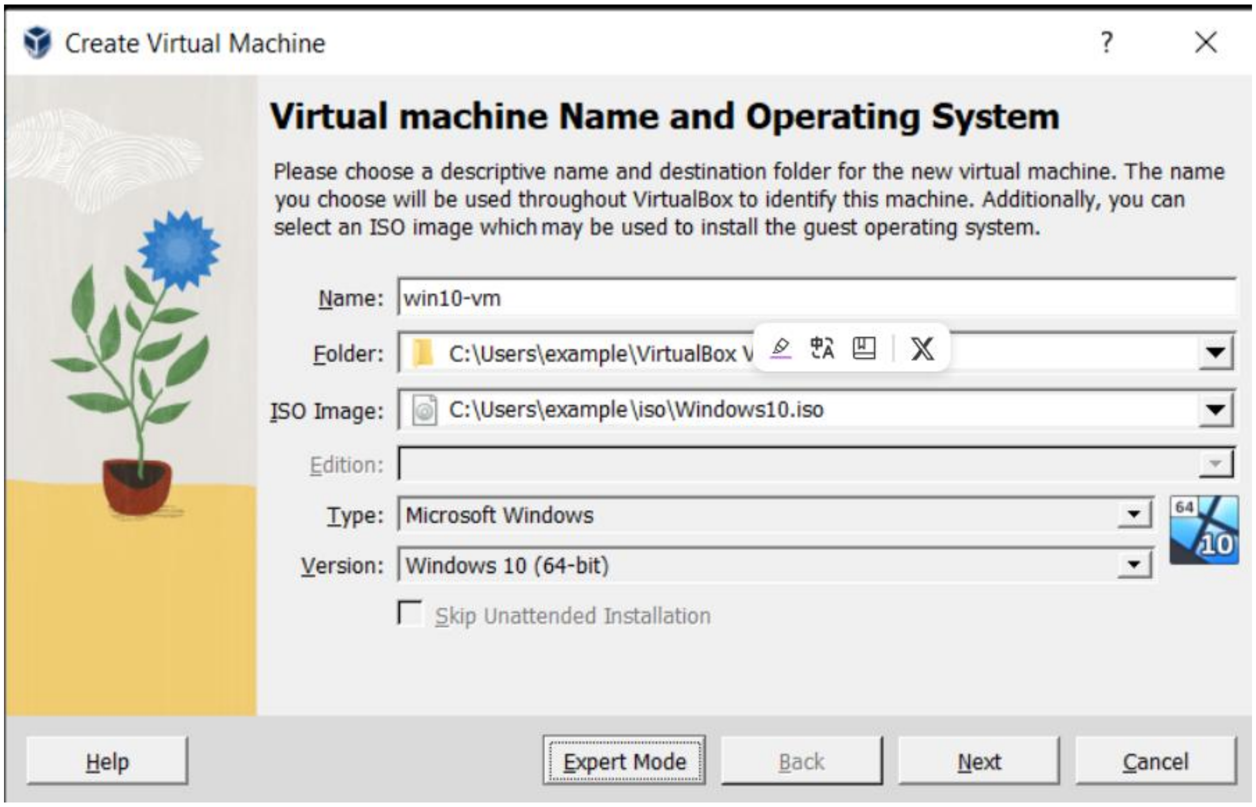


```
C:\>exit

C:\>ping 192.168.2.20

Pinging 192.168.2.20 with 32 bytes of data:

Reply from 192.168.2.20: bytes=32 time<1ms TTL=127
Reply from 192.168.2.20: bytes=32 time<1ms TTL=127
Reply from 192.168.2.20: bytes=32 time<1ms TTL=127
Reply from 192.168.2.20: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

2. Virtual box-based Web Server creation, Images/Snapshots access webpage from 2nd VM on another subnet work

Ans.

Create a New Virtual Machine



Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: win10-vm

Folder: C:\Users\example\VirtualBox V

ISO Image: C:\Users\example\iso\Windows10.iso

Edition:

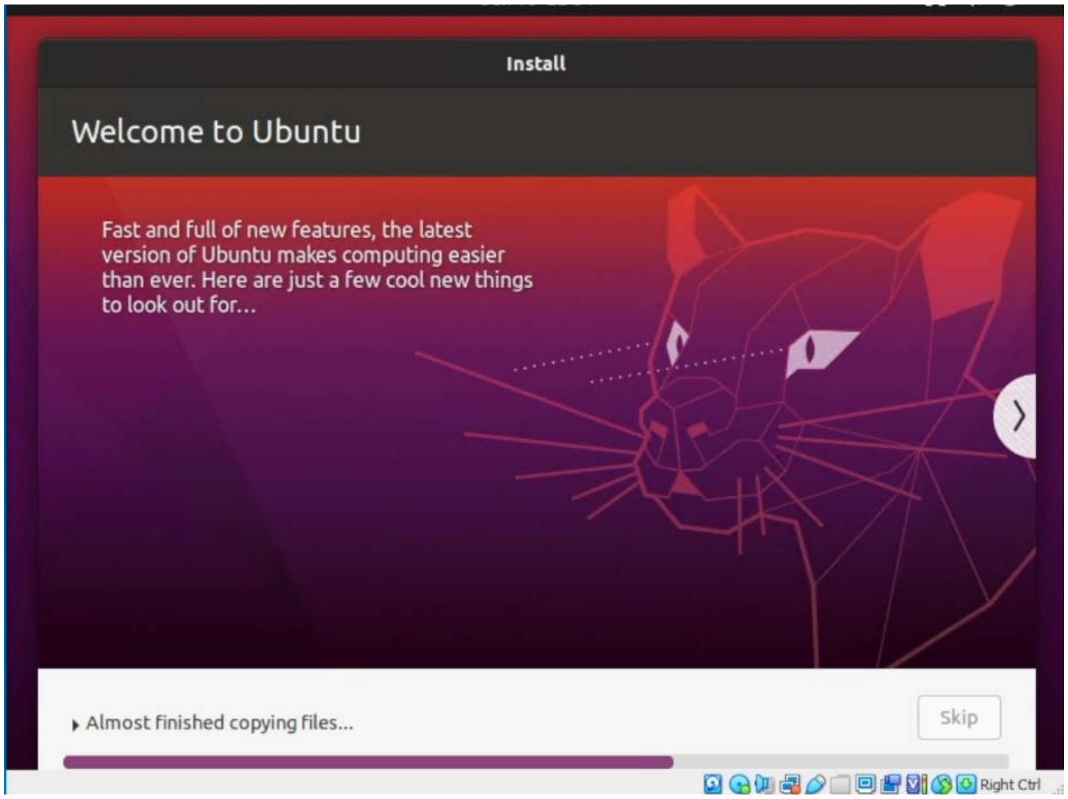
Type: Microsoft Windows

Version: Windows 10 (64-bit)

Skip Unattended Installation

1. Select your new VM, then click on Settings.
2. Go to the Storage tab:
   - Click on the empty CD/DVD icon under Storage Devices.
   - On the right side, click on the CD icon and select "Choose a disk file".
   - Browse to your downloaded Ubuntu Server ISO and select it.

Install Ubuntu Server



Welcome to Ubuntu

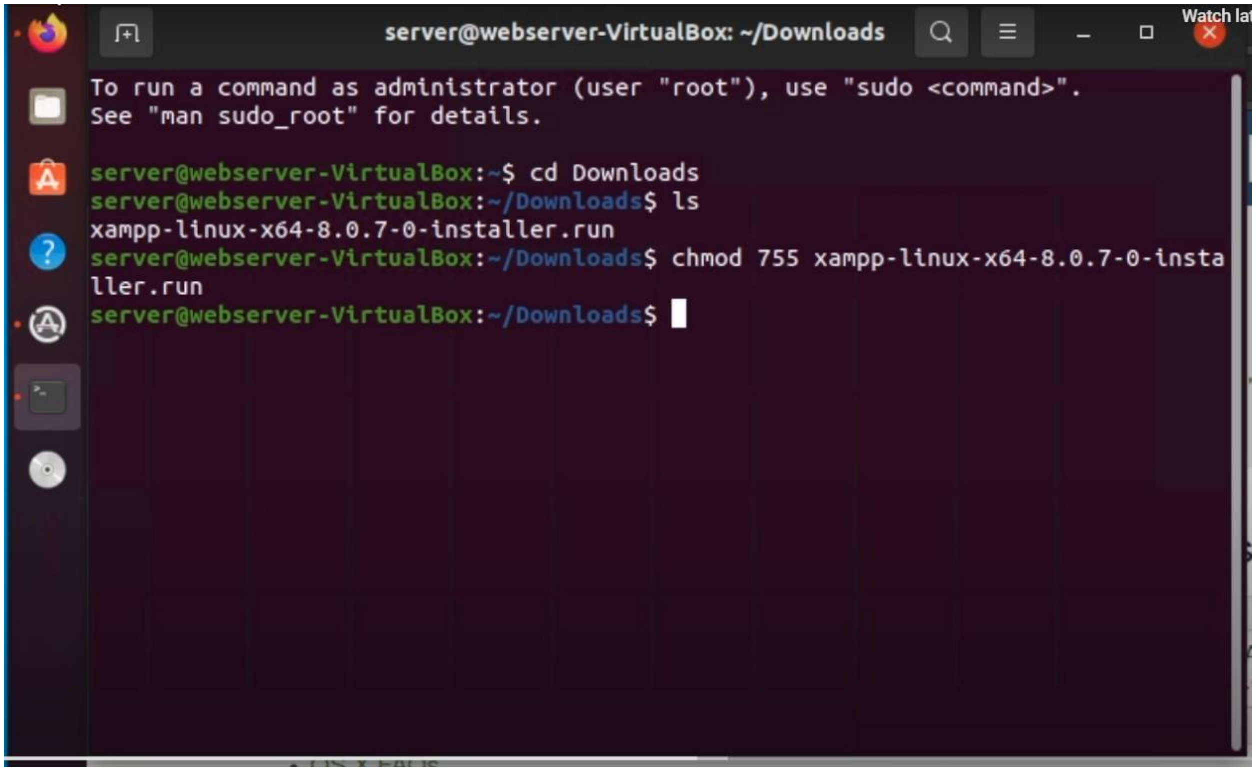
Fast and full of new features, the latest version of Ubuntu makes computing easier than ever. Here are just a few cool new things to look out for...

Almost finished copying files...

Access Your Web Server

1. After installation, reboot your VM.
2. Log in using your credentials.
3. Update your package list:

#sudo apt update



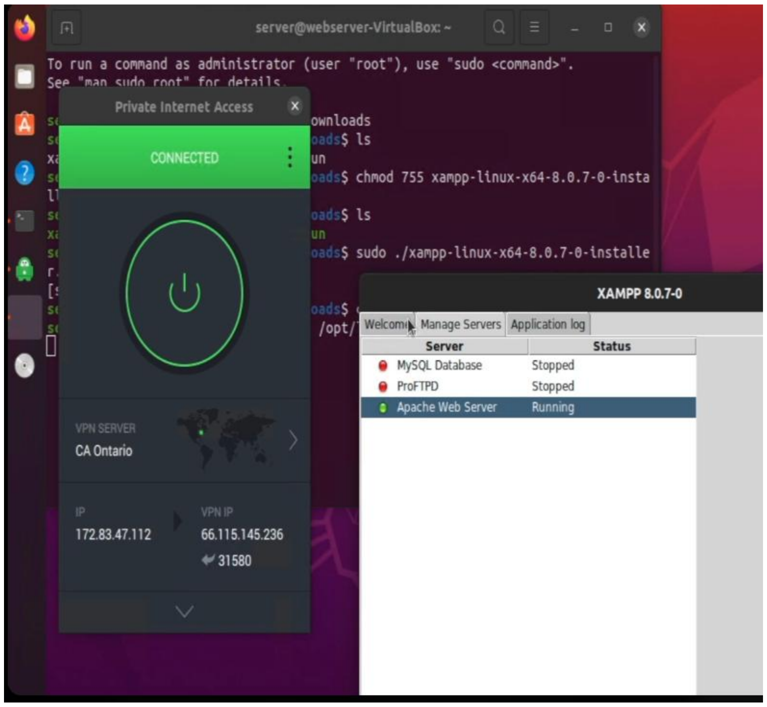
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

server@webserver-VirtualBox:~$ cd Downloads
server@webserver-VirtualBox:~/Downloads$ ls
xampp-linux-x64-8.0.7-0-installer.run
server@webserver-VirtualBox:~/Downloads$ chmod 755 xampp-linux-x64-8.0.7-0-insta
ller.run
server@webserver-VirtualBox:~/Downloads$ 
```

Install a web server package (e.g., Apache):

#sudo apt install apache2

#sudo systemctl start apache2

#sudo systemctl enable apache2



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
                                   ownloads
                                   oads$ ls
                                   un
                                   oads$ chmod 755 xampp-linux-x64-8.0.7-0-insta

                                   oads$ ls
                                   un
                                   oads$ sudo ./xampp-linux-x64-8.0.7-0-installe

                                   oads$
                                    /opt/
```

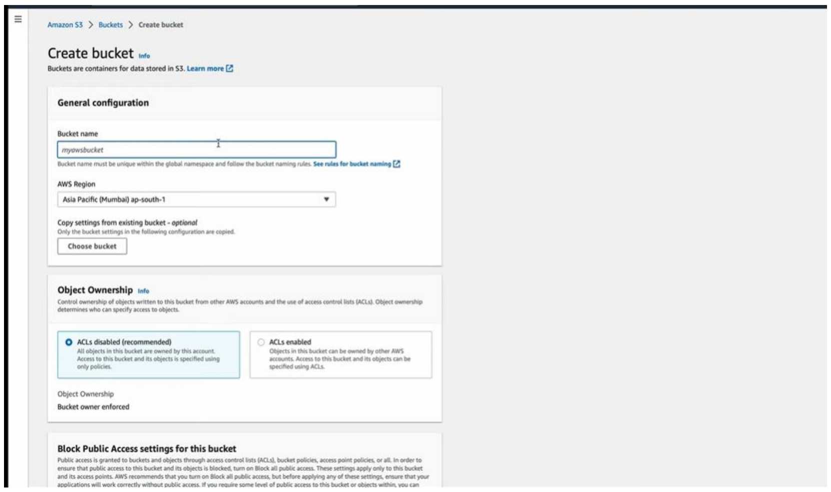
XAMPP 8.0.7-0

Server	Status
MySQL Database	Stopped
ProFTPD	Stopped
Apache Web Server	Running

3.EC2 AWS-S3 bucket based static web pages

Create a Bucket:

- Click on Create bucket.
- Enter a unique bucket name
- Choose a region close to your target audience to minimize latency.
- Keep the default settings for Block Public Access unless you need to allow public access for your website.
- Click Create bucket to finalize.



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: myawsbucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

AWS Region: Asia Pacific (Mumbai) ap-south-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

[✓ ACLs disabled (recommended)] All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled: Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership: Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can

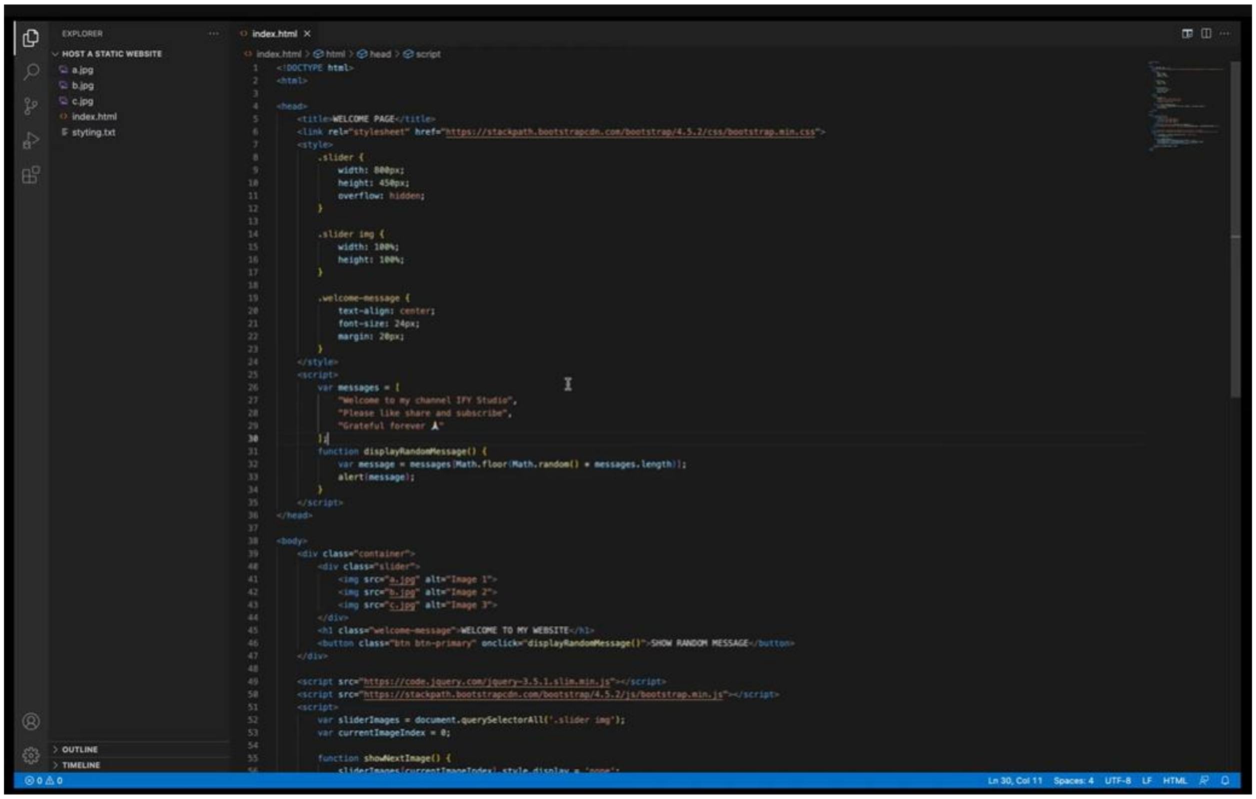
Configure Bucket Policy

Set Permissions:

- Go to the Permissions tab.
- Click on Bucket Policy and add a policy that allows public access:

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::example.com/*"
  }
```

Upload Files:



```html
<!DOCTYPE html>
<html>

<head>
    <title>WELCOME PAGE</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        .slider {
            width: 800px;
            height: 450px;
            overflow: hidden;
        }

        .slider img {
            width: 100%;
            height: 100%;
        }

        .welcome-message {
            text-align: center;
            font-size: 24px;
            margin: 20px;
        }
    </style>
    <script>
        var messages = [
            "Welcome to my channel IFY Studio",
            "Please like share and subscribe",
            "Grateful forever 🙏"
        ];
        function displayRandomMessage() {
            var message = messages[Math.floor(Math.random() * messages.length)];
            alert(message);
        }
    </script>
</head>

<body>
    <div class="container">
        <div class="slider">
            <img src="a.jpg" alt="Image 1">
            <img src="b.jpg" alt="Image 2">
            <img src="c.jpg" alt="Image 3">
        </div>
        <h1 class="welcome-message">WELCOME TO MY WEBSITE</h1>
        <button class="btn btn-primary" onclick="displayRandomMessage()">SHOW RANDOM MESSAGE</button>
    </div>

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    <script>
        var sliderImages = document.querySelectorAll('.slider img');
        var currentImageIndex = 0;

        function showNextImage() {
            sliderImages[currentImageIndex].style.display = 'none';
```

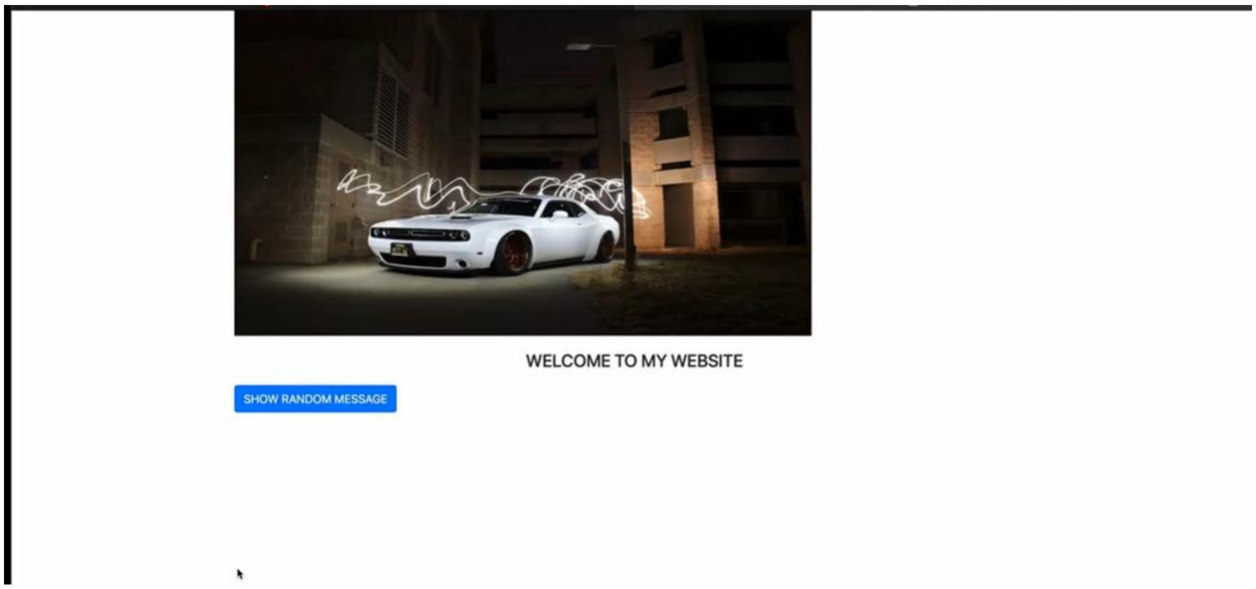
- After uploading, your static website can be accessed using the endpoint provided in the static website hosting section (http://example.com.s3-website-us-east-1.amazonaws.com).

Set Up an EC2 Instance

If you need server-side processing or other functionalities not supported by S3 alone:

1. Launch an EC2 Instance:
    - Navigate to EC2 in the AWS Management Console.
    - Click on Launch Instance, choose an Amazon Machine Image (AMI), select instance type, configure security groups, and launch
2. Install a Web Server
   If you need dynamic content or server capabilities:

```
sudo apt update

sudo apt install apache2
```

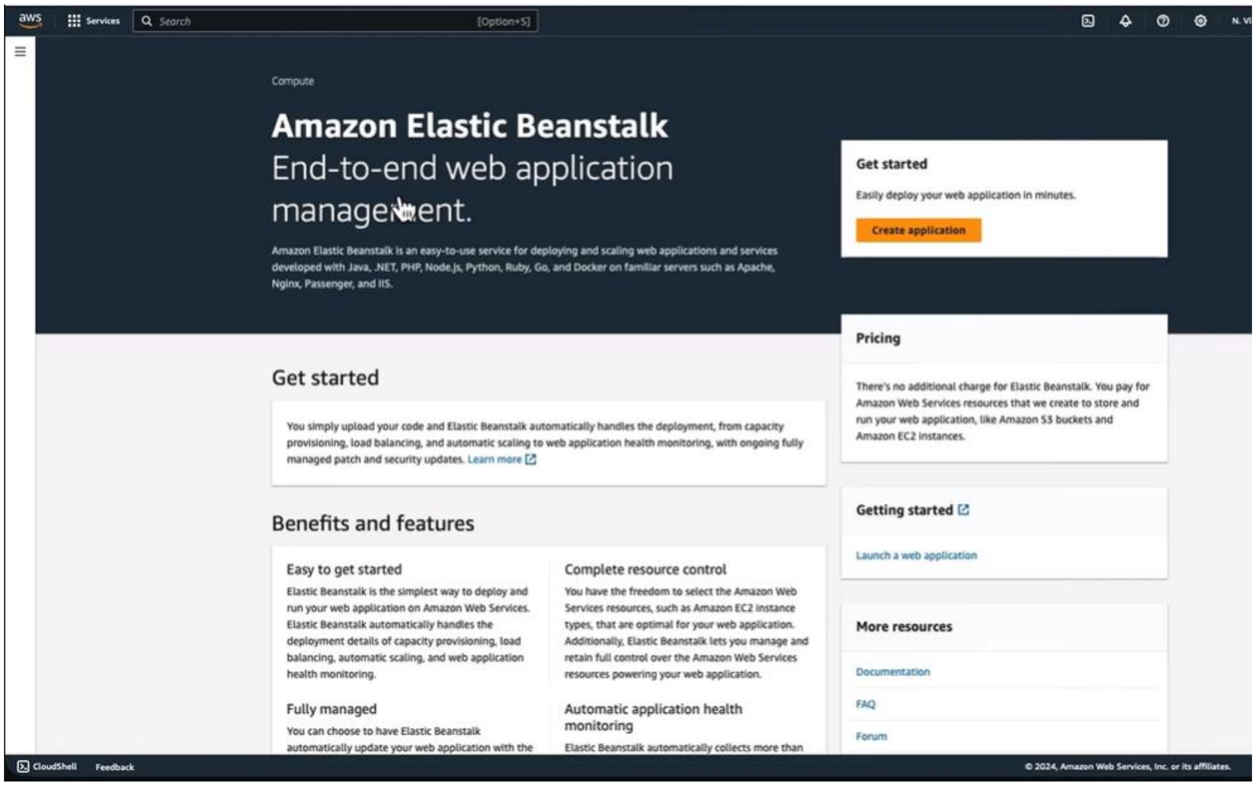


4. EC2 AWS - Web application using Beanstalk

Configure Environment

1. Choose Environment Type:
    - Select Web server environment as the environment tier.
2.
3. Select Platform:
    - Choose the platform for your application (e.g., Node.js, Python, Java, etc.).
4.

5. Configure Service Role:
    - Choose an existing service role or create a new one if necessary.
    - If creating a new role, navigate to the IAM console to set permissions like `AWSElasticBeanstalkWebTier`, `AWSElasticBeanstalkWorkerTier`, and others as needed 12.


Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create application

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. Learn more

Getting started

Launch a web application

Benefits and features

Easy to get started

Elastic Beanstalk is the simplest way to deploy and run your web application on Amazon Web Services. Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, automatic scaling, and web application health monitoring.

Complete resource control

You have the freedom to select the Amazon Web Services resources, such as Amazon EC2 instance types, that are optimal for your web application. Additionally, Elastic Beanstalk lets you manage and retain full control over the Amazon Web Services resources powering your web application.

More resources

Documentation

FAQ

Forum

Fully managed

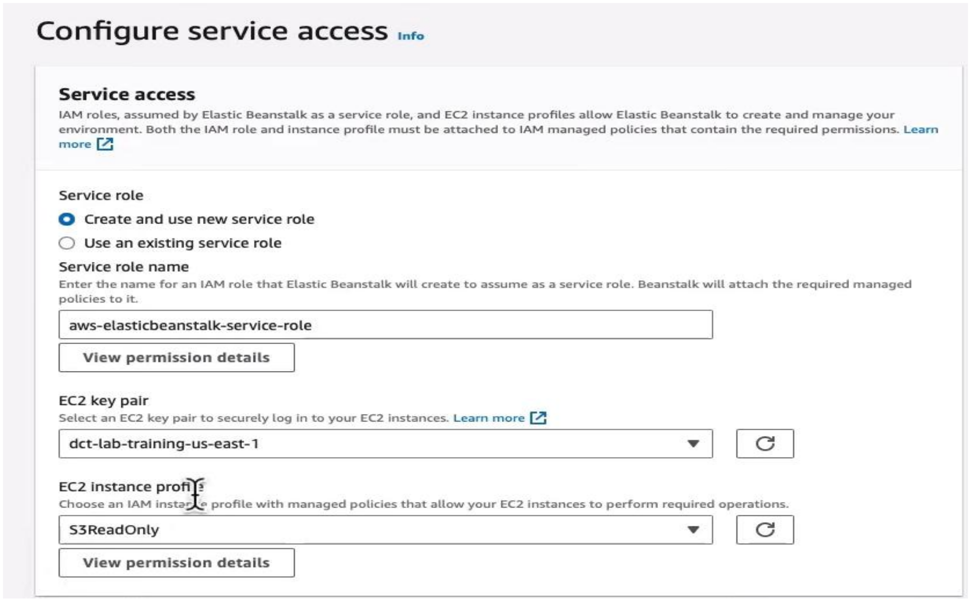
You can choose to have Elastic Beanstalk automatically update your web application with the

Automatic application health monitoring

Elastic Beanstalk automatically collects more than

© 2024, Amazon Web Services, Inc. or its affiliates.

Cnfigure Application Settings


Configure service access Info

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more

Service role

(●) Create and use new service role

( ) Use an existing service role

Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

View permission details

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

dct-lab-training-us-east-1

EC2 instance profile

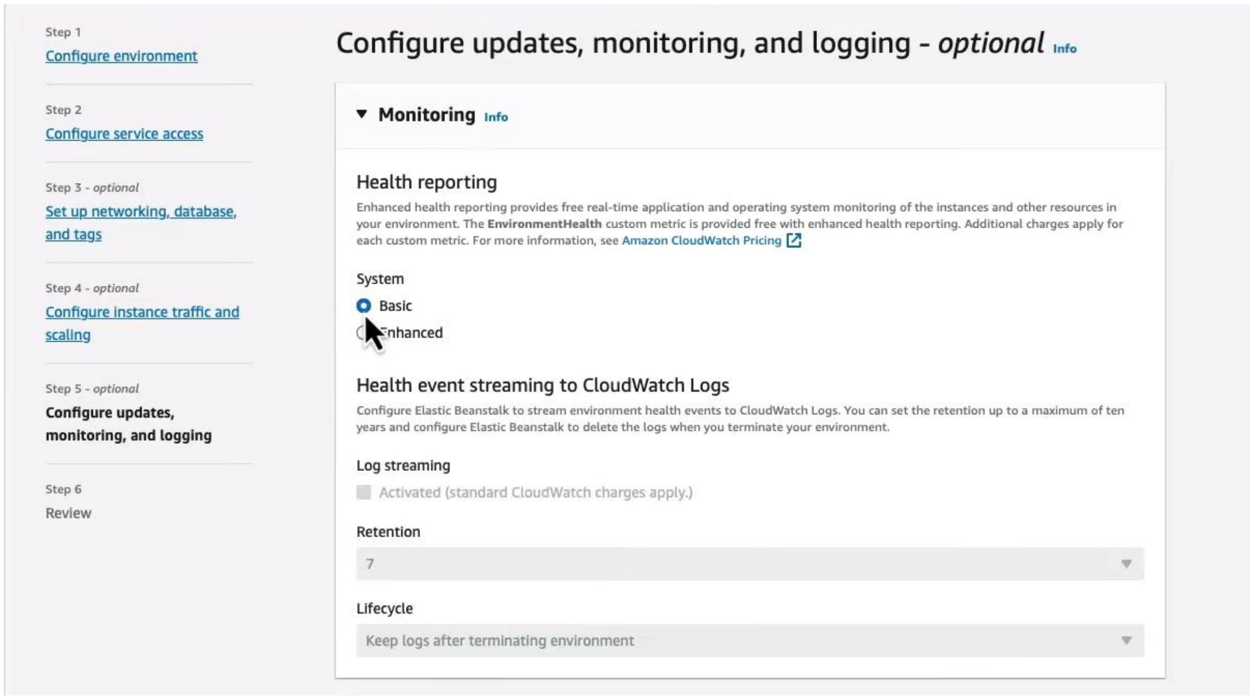
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

S3ReadOnly

View permission details

Create Environment:

- Click on Create Environment.
- AWS will begin provisioning resources, including EC2 instances, security groups, and S3 buckets for storage



Step 1
Configure environment

Step 2
Configure service access

Step 3 - *optional*
Set up networking, database, and tags

Step 4 - *optional*
Configure instance traffic and scaling

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Configure updates, monitoring, and logging - *optional* Info

Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see Amazon CloudWatch Pricing

System
[✓ Basic] Enhanced

Health event streaming to CloudWatch Logs

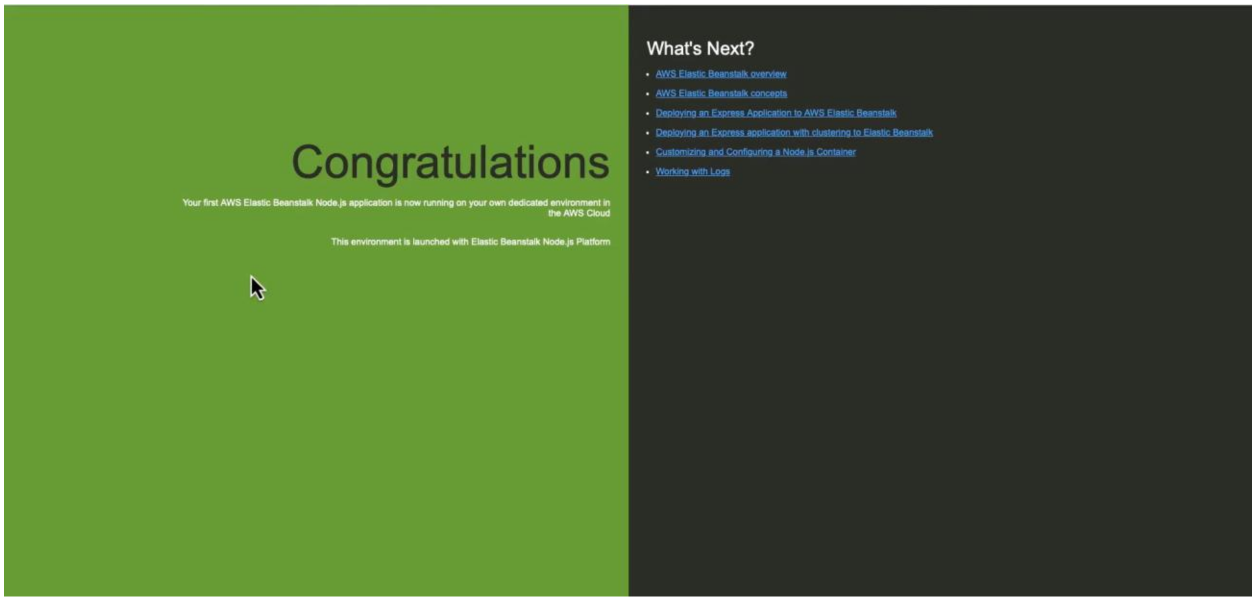
Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming
Activated (standard CloudWatch charges apply.)

Retention
7

Lifecycle
Keep logs after terminating environment

Deploy Your Application



Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

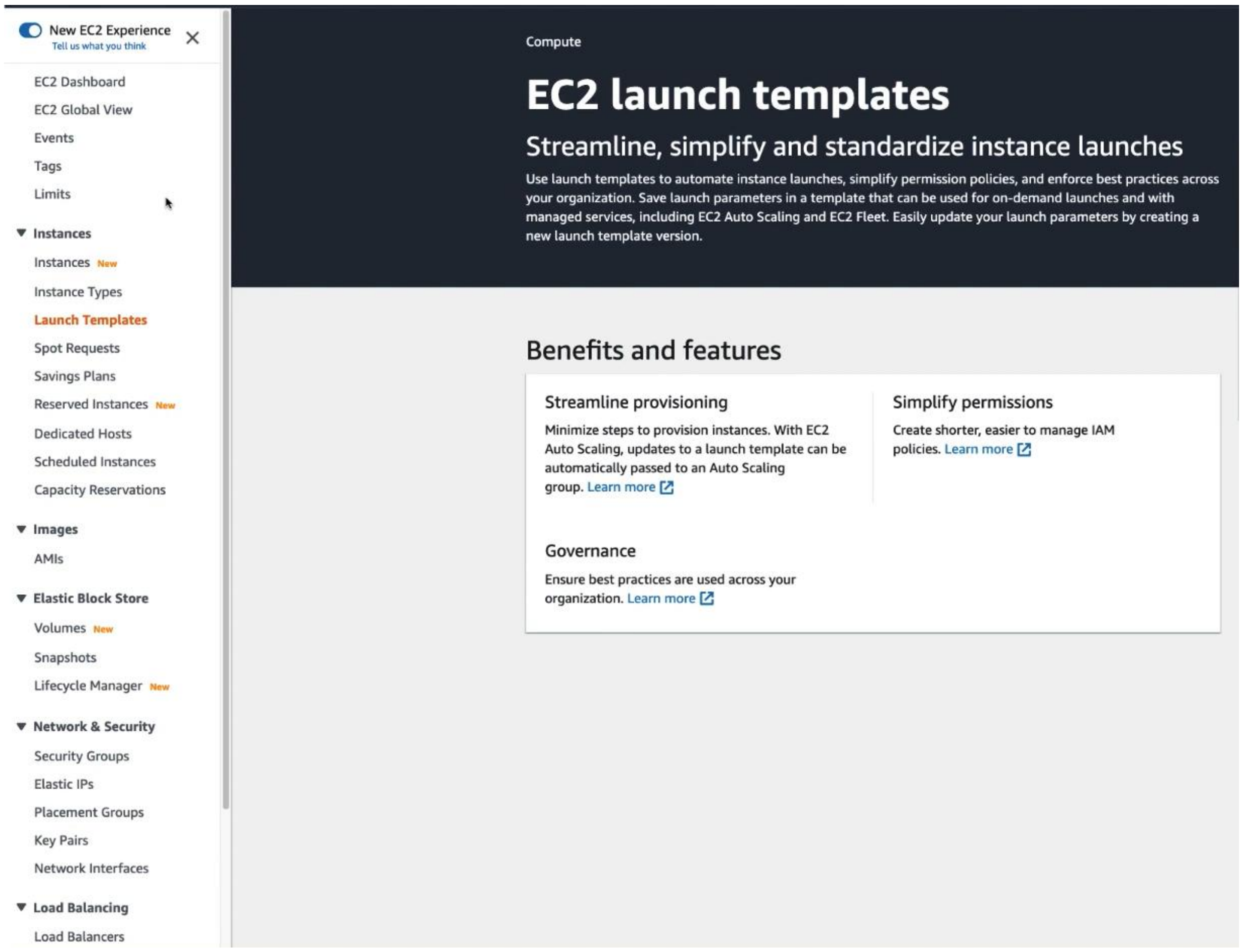
What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploying an Express Application to AWS Elastic Beanstalk
- Deploying an Express application with clustering to Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

5. AWS Local balancing and auto scaling

Create a Launch Template

- Click on Launch Templates in the left-hand menu.
- Click on Create launch template.
- Fill in the required details:
    - Name: Provide a name for your template (e.g., `my-launch-template`).
    - Amazon Machine Image (AMI): Select an appropriate AMI (e.g., Amazon Linux 2).
    - Instance Type: Choose an instance type (e.g., `t2.micro` for free tier).
    - Key Pair: Select or create a key pair for SSH access.
    - Security Group: Define security group settings to control inbound/outbound traffic.



Compute

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

Benefits and features

Streamline provisioning

Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group. Learn more

Simplify permissions

Create shorter, easier to manage IAM policies. Learn more

Governance

Ensure best practices are used across your organization. Learn more

Create an Auto Scaling Group

Navigate to Auto Scaling Groups:

- In the EC2 Dashboard, click on Auto Scaling Groups in the left-hand menu.

Create Auto Scaling Group:

- Click on the Create Auto Scaling group button.
- Enter a name for your Auto Scaling group (e.g., `my-auto-scaling-group`).
- Select the launch template you created earlier.

Configure Network Settings:

- Choose the VPC and subnets where instances will be launched. Select multiple subnets for high availability.
- Click on Next to proceed.

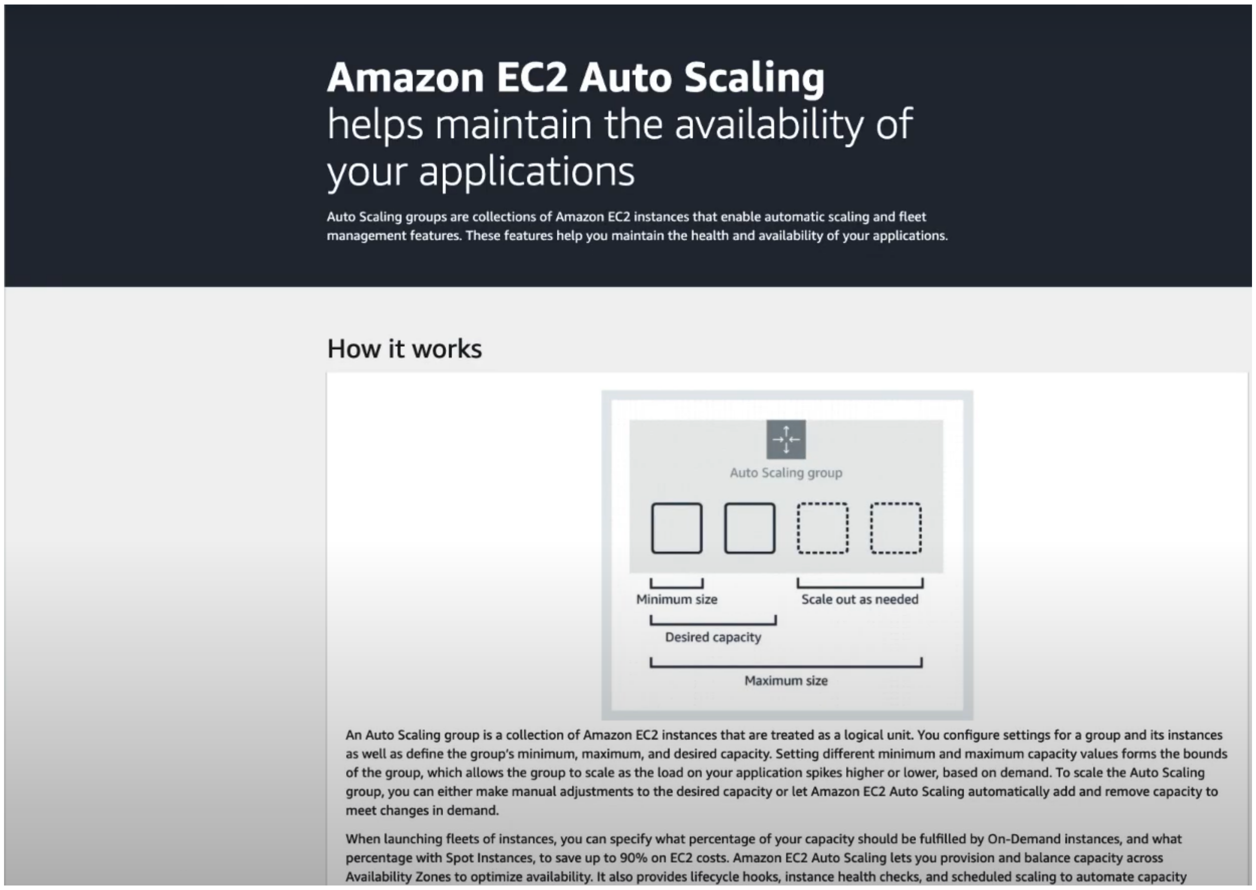
Set Scaling Policies

Define Capacity Settings:

- Set minimum, maximum, and desired capacity for your instances.
- For example:Minimum: 1

Configure Scaling Policies:

- You can choose between different scaling policies:Target Tracking Scaling
- Set up notifications if desired to alert you of scaling events.



Amazon EC2 Auto Scaling helps maintain the availability of your applications

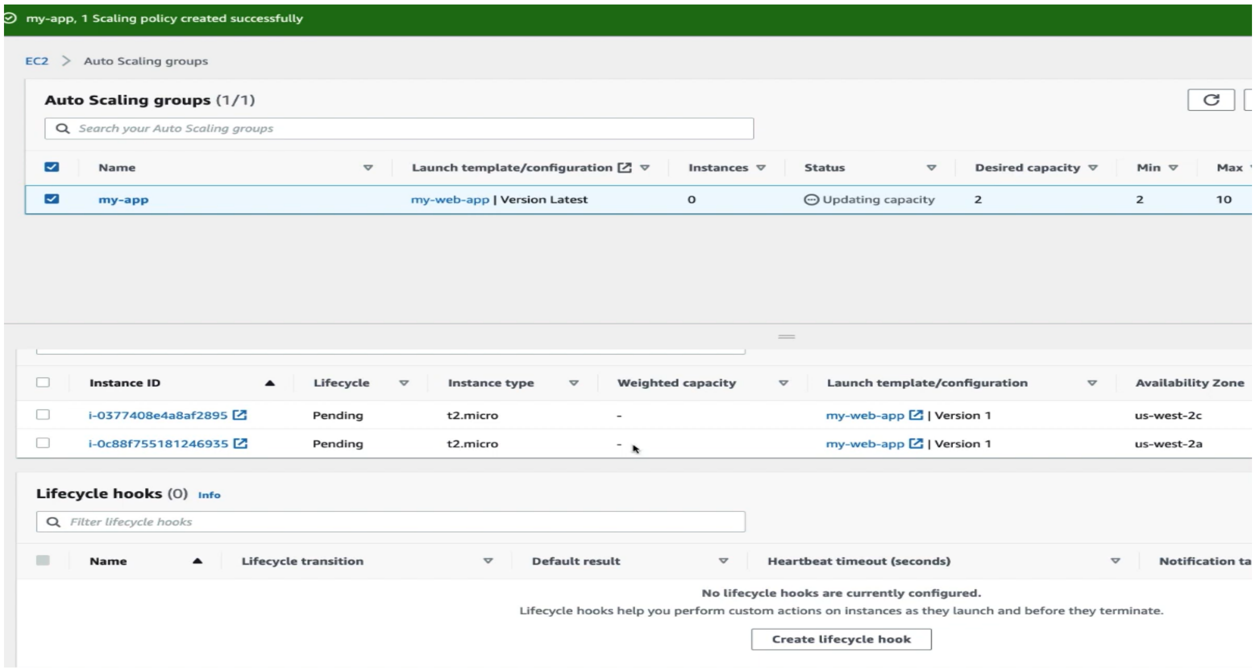
Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

How it works

An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity. Setting different minimum and maximum capacity values forms the bounds of the group, which allows the group to scale as the load on your application spikes higher or lower, based on demand. To scale the Auto Scaling group, you can either make manual adjustments to the desired capacity or let Amazon EC2 Auto Scaling automatically add and remove capacity to meet changes in demand.

When launching fleets of instances, you can specify what percentage of your capacity should be fulfilled by On-Demand instances, and what percentage with Spot Instances, to save up to 90% on EC2 costs. Amazon EC2 Auto Scaling lets you provision and balance capacity across Availability Zones to optimize availability. It also provides lifecycle hooks, instance health checks, and scheduled scaling to automate capacity

Review



my-app, 1 Scaling policy created successfully

EC2 > Auto Scaling groups

Auto Scaling groups (1/1)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
my-app	my-web-app | Version Latest	0	Updating capacity	2	2	10

Instance ID	Lifecycle	Instance type	Weighted capacity	Launch template/configuration	Availability Zone
i-0377408e4a8af2895	Pending	t2.micro	-	my-web-app | Version 1	us-west-2c
i-0c88f755181246935	Pending	t2.micro	-	my-web-app | Version 1	us-west-2a

Lifecycle hooks (0) Info

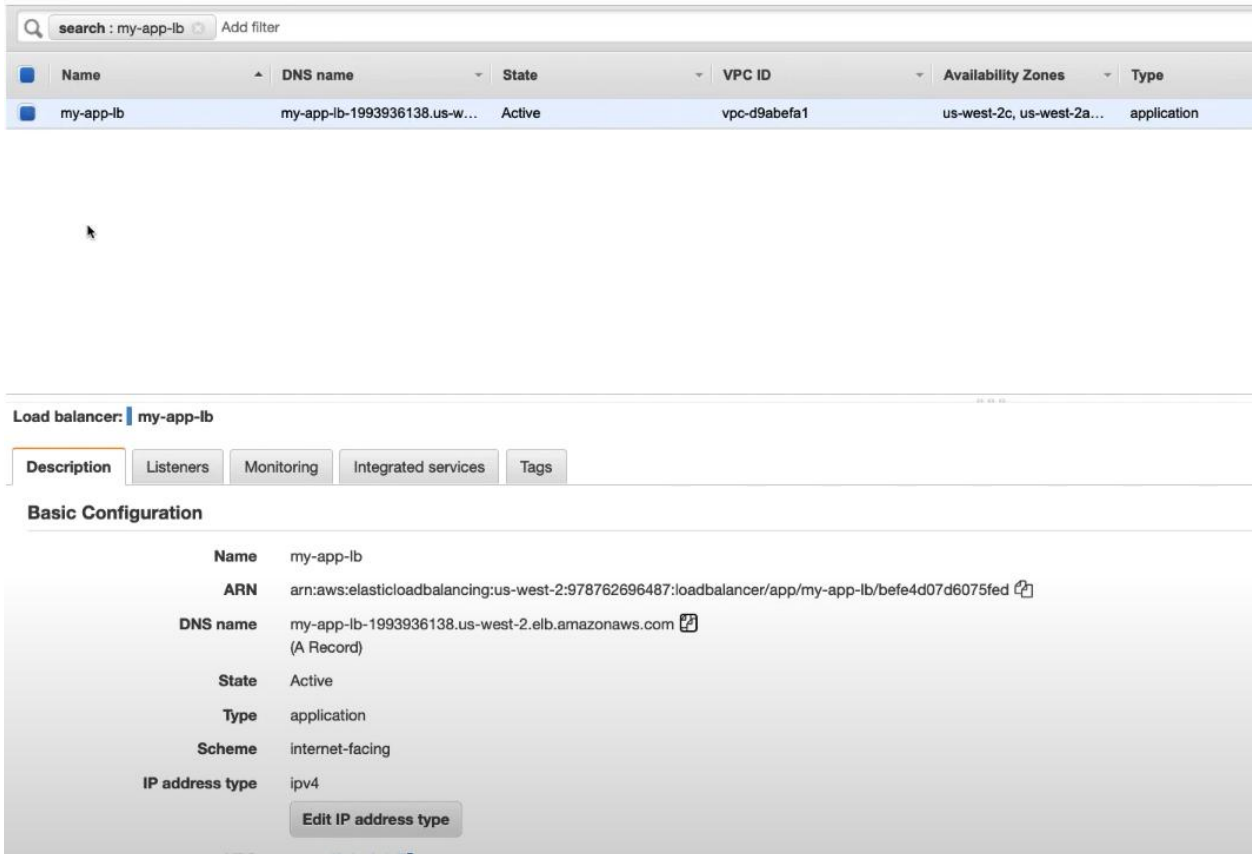
Name	Lifecycle transition	Default result	Heartbeat timeout (seconds)	Notification ta

No lifecycle hooks are currently configured.
Lifecycle hooks help you perform custom actions on instances as they launch and before they terminate.

Configure Load Balancer

Attach Load Balancer:

- During the Auto Scaling group setup, you can optionally create or attach an Elastic Load Balancer (ELB).
- If creating a new load balancer, follow the prompts to configure it, ensuring it distributes traffic across your EC2 instances effectively.


search : my-app-lb Add filter

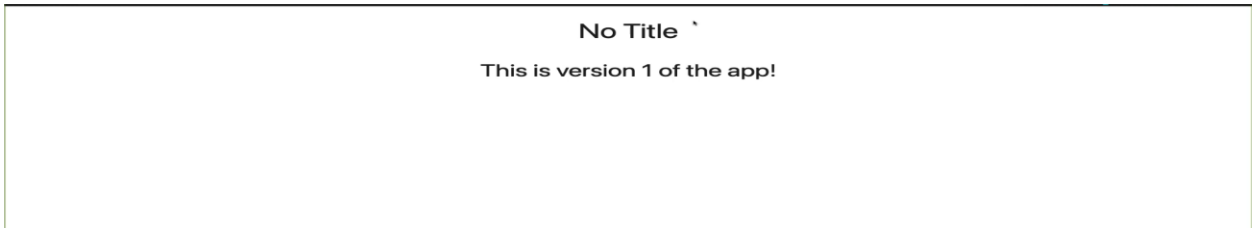
Name	DNS name	State	VPC ID	Availability Zones	Type
my-app-lb	my-app-lb-1993936138.us-w...	Active	vpc-d9abefa1	us-west-2c, us-west-2a...	application

Load balancer: my-app-lb

Basic Configuration

Name	my-app-lb
ARN	arn:aws:elasticloadbalancing:us-west-2:978762696487:loadbalancer/app/my-app-lb/befe4d07d6075fed
DNS name	my-app-lb-1993936138.us-west-2.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	internet-facing
IP address type	ipv4

Test Load Balancing (if configured)


No Title

This is version 1 of the app!