

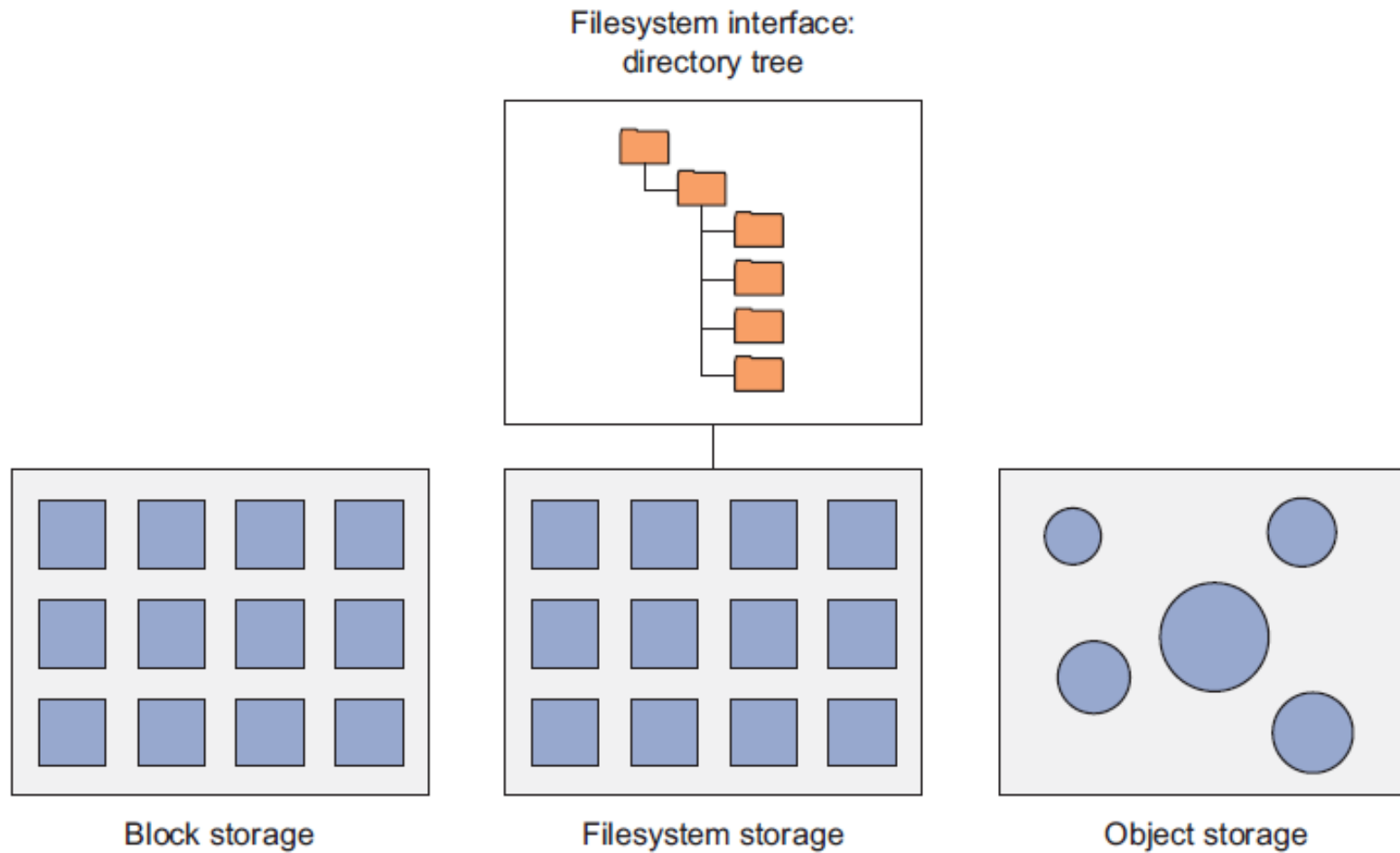
**2.2**

## **Capa de almacenamiento**

## 2.2 Capa de almacenamiento

- En la primera práctica se desplego una web de wordpress en una maquina donde el sistema de ficheros se albergaba en un volumen de disco EBS de 8Gb, relativamente pequeño, pero suficiente para albergar las necesidades iniciales de un sitio Wordpress ejemplo
- En el mundo real donde las aplicaciones generan bases de datos que crecen rápidamente , contienen directorios con una gran cantidad de videos y elementos multimedia además gigas de logs incluso 80 GB son pocos
- ¿Entonces porque no se provisiona al sistema con un disco duro mayor?
- **Un disco duro de estado solido de 100GB = 10\$/mes**
- Por ejemplo el servicio de almacenamiento de objetos es más barato , pero su objetivo es que sea de lectura
  - Amazon AWS S3  $1GB = 0.025\$/mes$   $100 GB = 2.5\$/mes$
- ¿Qué contratarías si el sitio Wordpress esta destinado a albergar ficheros multimedia de gran tamaño?

## 2.2 Capa de almacenamiento



## 2.2 Capa de almacenamiento

- Almacenamiento de nivel de objetos
  - AWS S3 y Glacier
  - Todo elemento tiene asociado una URL para su acceso y único



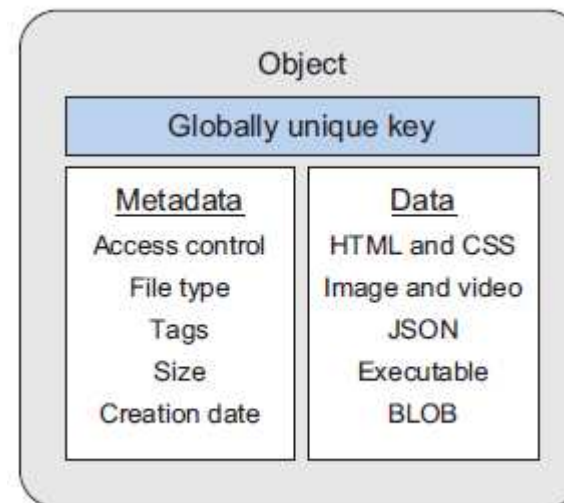
Object-level storage



Designed for 99.99999999 durability



Event triggers



[https://s3-ap-northeast-1.amazonaws.com/\[bucket name\]/](https://s3-ap-northeast-1.amazonaws.com/[bucket name]/)



[https://s3-ap-northeast-1.amazonaws.com/\[bucket name\]/Preview2.mp4](https://s3-ap-northeast-1.amazonaws.com/[bucket name]/Preview2.mp4)

## 2.2 Capa de almacenamiento

- S3 escenarios de uso
  - Escribir una vez , leer muchas
  - Data sets crecientes
  - Backup de datos
  - Ficheros estáticos de una web vía CDN
  - Alojamiento de una web estática, por ejemplo un SPA
  - Almacenamiento \$0.03GB/mes
  - Transferencia de datos al exterior descargado \$0.09GB



### Pay only for what you use, including:

GBs per month  
Transfer OUT to other regions or the internet  
PUT, COPY, POST, LIST, and GET requests

### You do NOT have to pay for:

Transfer IN to Amazon S3  
  
Transfer OUT to Amazon EC2 in the same region, or to CloudFront



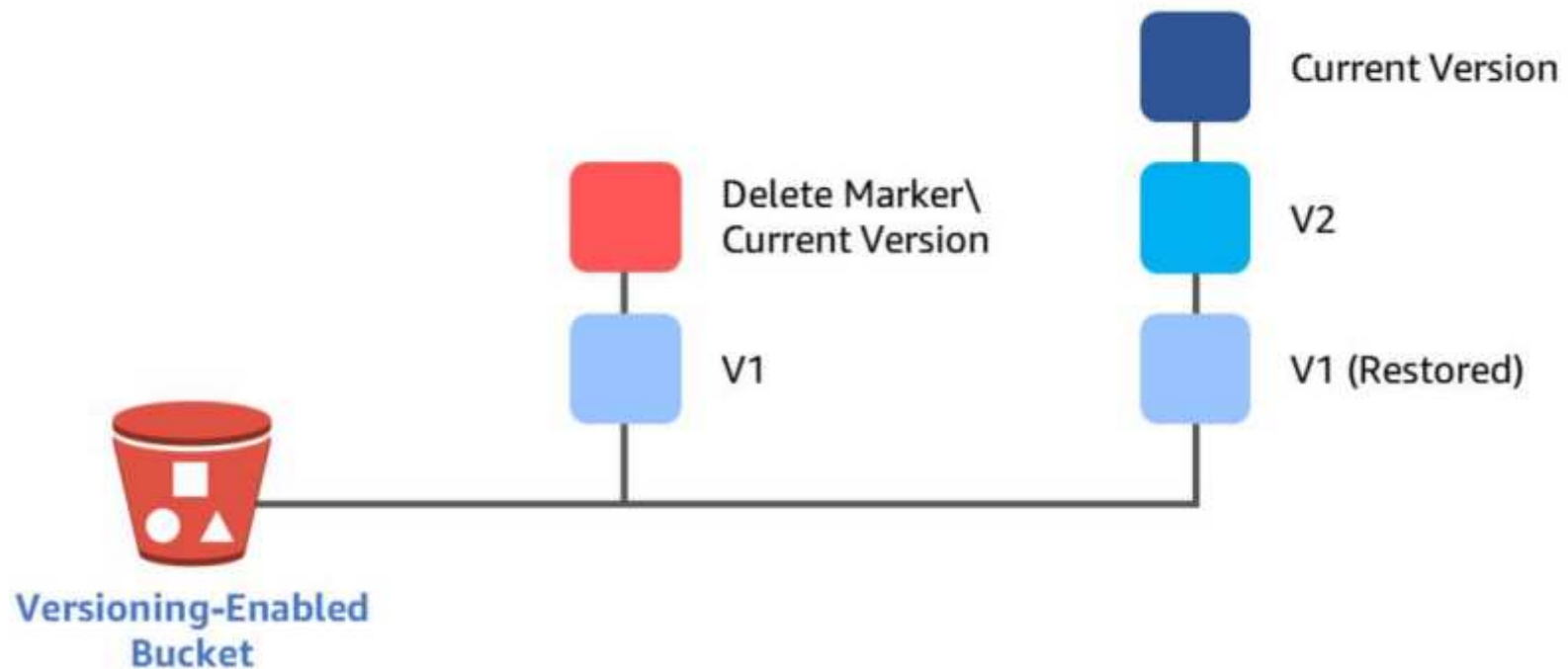
## 2.2 Capa de almacenamiento

- S3 : Niveles de disponibilidad

Value	Standard	Standard-IA	Reduced Redundancy
Durability	99.999999999%	99.999999999%	99.99%
Availability	99.99%	99.9%	99.99%

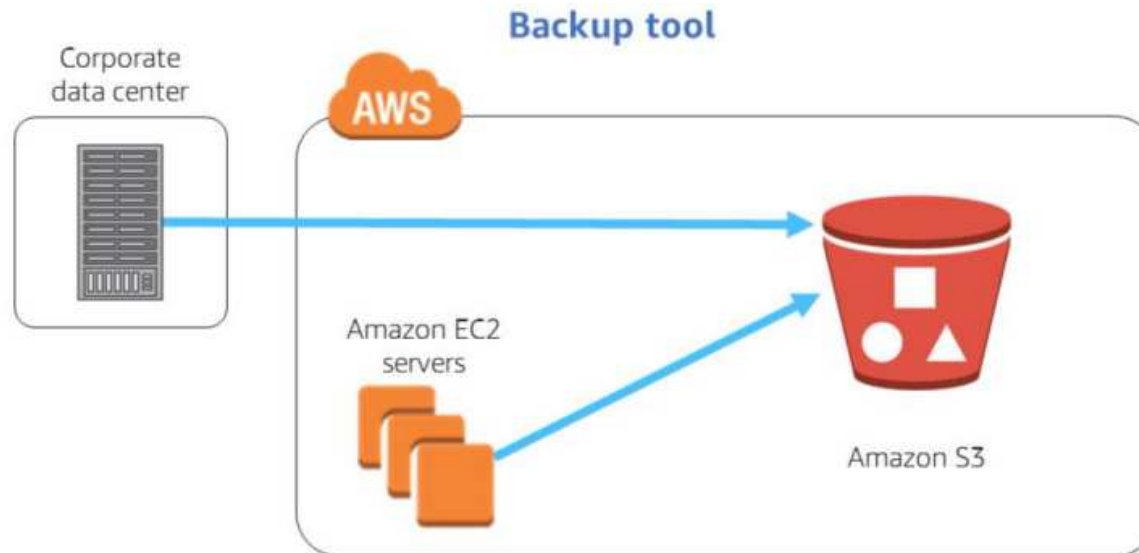
## 2.2 Capa de almacenamiento

- S3 : Versioneado



## 2.2 Capa de almacenamiento

- S3 : Como servicio de backup
  - Por ejemplo , para los AMI

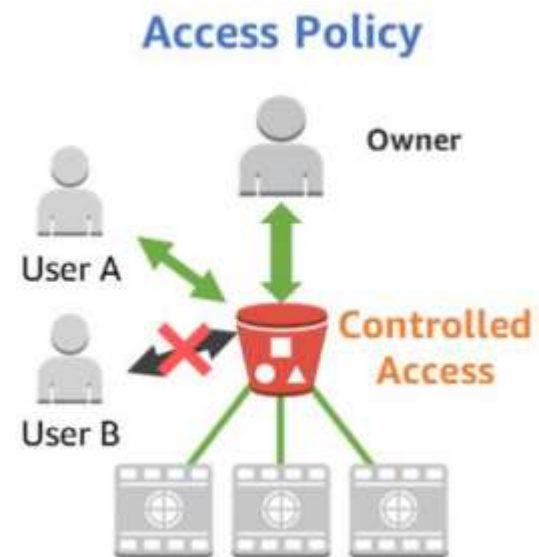


- Para datos de larga duración y poca modificación mejor el servicio AWS GLACIER



## 2.2 Capa de almacenamiento

- S3 : Control de acceso

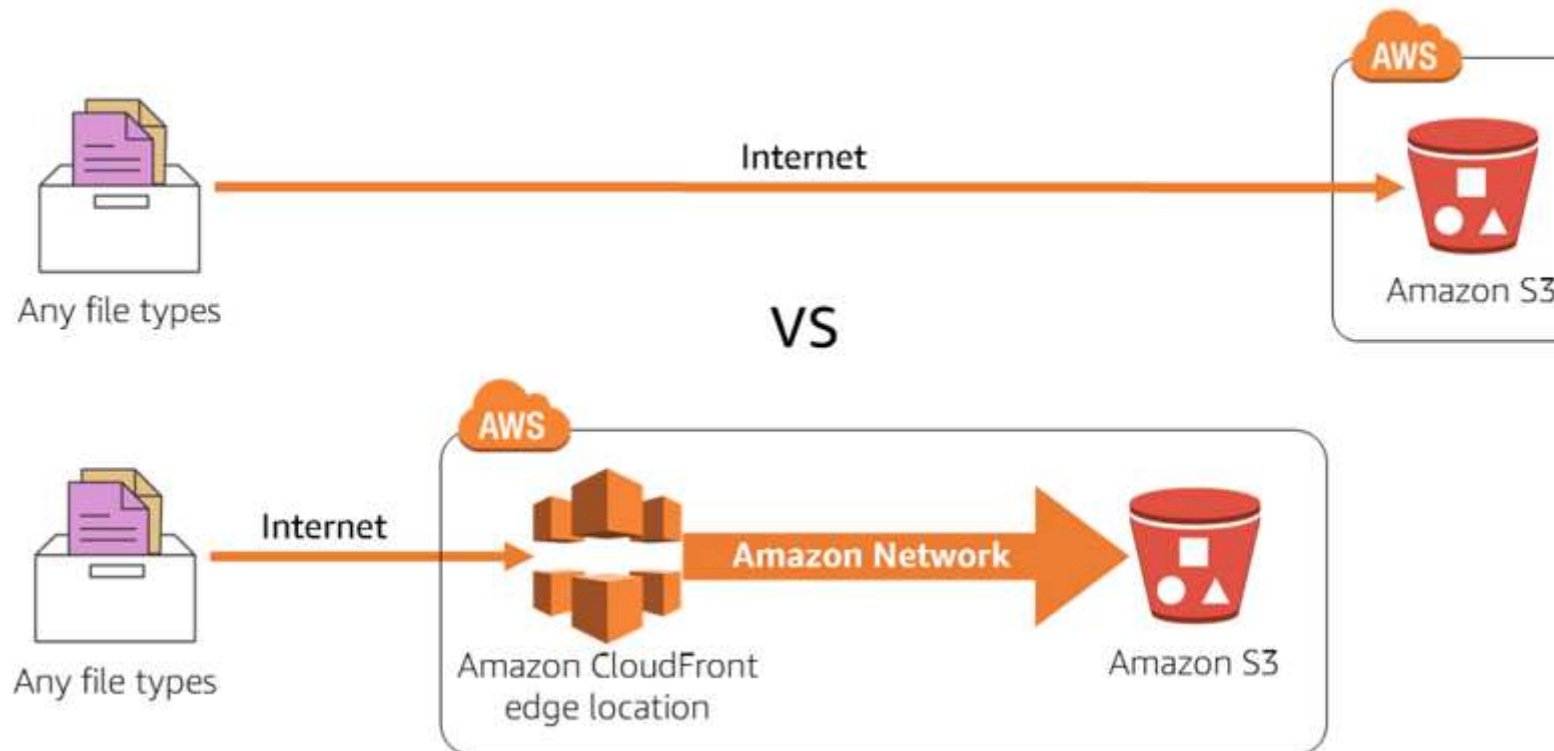


## 2.2 Capa de almacenamiento

- S3 : Sincronización upload
  - Multipart (5TB)
  - SFTP
  - Datasync (NFS)

## 2.2 Capa de almacenamiento

- S3 : Aceleración de descarga : CDN : Cloudfront



## 2.2 Capa de almacenamiento

- S3 : Curisiodad: Moviendo datos : AWS Snowball

### AWS Snowball

Petabyte-scale data transport




### AWS Snowmobile

Exabyte-scale data transport




## 2.2 Capa de almacenamiento


- S3 vs Glacier




Amazon Glacier



Long-term data storage

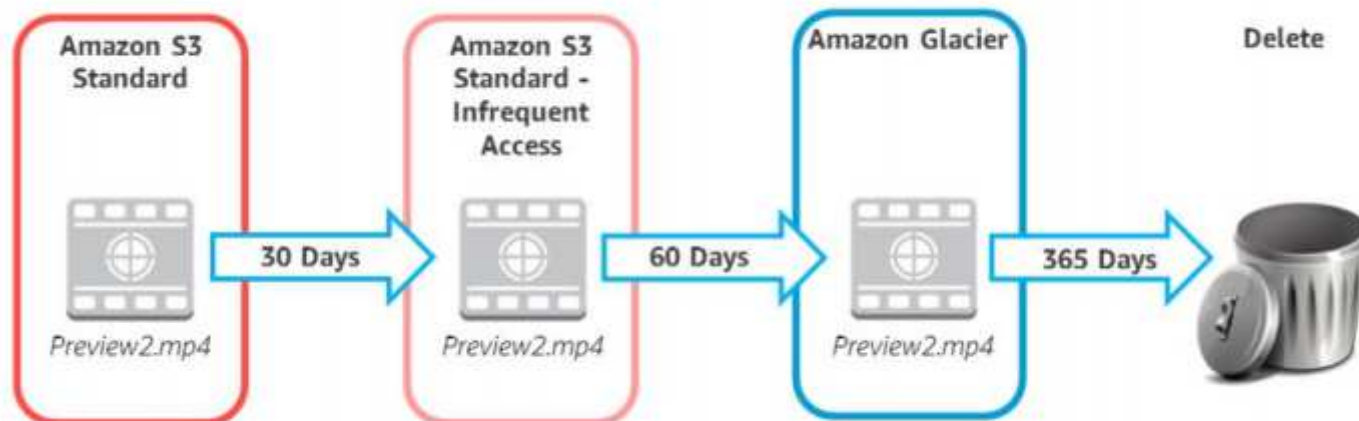


Archival or backup



Very low-cost storage

	S3	Glacier
Cost per GB	\$0.03 USD	\$0.01 USD
Accessibility	Immediate upon request	3–5 hours after request
Durability	Designed annual durability of 99.999999999%	Designed annual durability of 99.999999999%



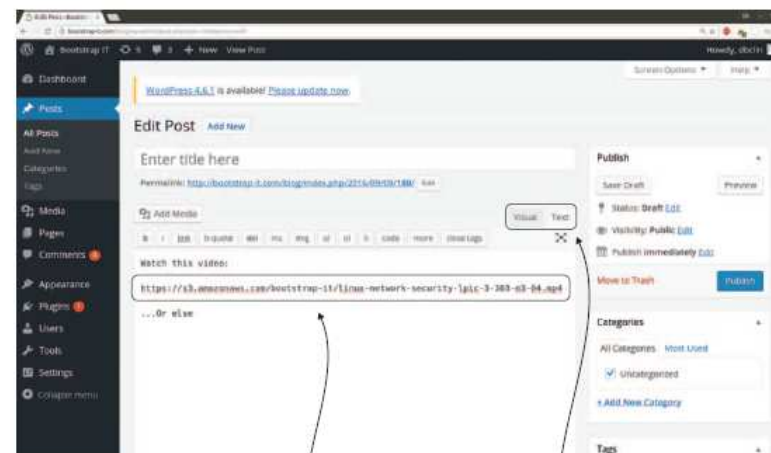
## 2.2 Capa de almacenamiento

- Casos de uso S3
- Ejercicios
  - UtilizarS3 para albergar los ficheros multimedia de una página web
  - Back up de datos y snapshots
  - Archivando ficheros , reglas de ciclo de vida
  - Integrando S3 en las aplicaciones
  - Web hosting
- Práctica
  - UtilizarS3 para albergar los ficheros multimedia de una web basada en Wordpress y ofrecer ese contenido mediante un servicio CDN

## 2.2 Capa de almacenamiento

- Ejercicios 1: Utilizar S3 para albergar los ficheros multimedia de una página web
- Utilizando un servidor Ec2 con wordpress simplemente en un post utilizar la url del objeto S3 para insertar elementos multimedia

```
<div align="center">  
<br><br>  
<hr>  
<b>  
<video width="640" height="480" controls>  
  <source src="https://s3.amazonaws.com/bootstrap-it/linux-network-  
    security-  
    lpic-3-303-m3-04.mp4" type="video/mp4">  
</video>  
<hr>
```



The endpoint is already  
inserted into the page.

Toggle between Visual  
and Text views.

## 2.2 Capa de almacenamiento

- Ejercicios 2: Web hosting de páginas estáticas
  - Crear un Budget y configurar sus propiedades para que sea para alojamiento web

Amazon S3 > practica-macc-web-estatica

Información general	Propiedades	Permisos
Administración		

### Control de versiones

Conservar varias versiones de un objeto en el mismo bucket.

[Más información](#)

☐ Deshabilitada

### Registro de acceso del servidor

Configure las entradas de registro de acceso que proporcionen detalles sobre las solicitudes de acceso.

[Más información](#)

☐ Deshabilitada

### Alojamiento de sitios web estáticos

Alojar un sitio web estático, que no necesita tecnologías de servidor.

[Más información](#)

☐ Deshabilitada

### Registro de nivel de objeto

Registre la actividad de la API de nivel de objeto utilizando la característica de eventos de datos de CloudTrail (con costo adicional).

[Más información](#)

☐ Deshabilitada

### Alojamiento de sitios web estáticos

Punto de enlace : <http://practica-macc-web-estatica.s3-website.eu-central-1.amazonaws.com>

☒ Usar este bucket para alojar un sitio web [Más información](#)

Documento de índice [i](#)

Documento de error [i](#)

Reglas de redireccionamiento (opcionales) [i](#)

☐ Solicitudes de redireccionamiento [Más información](#)

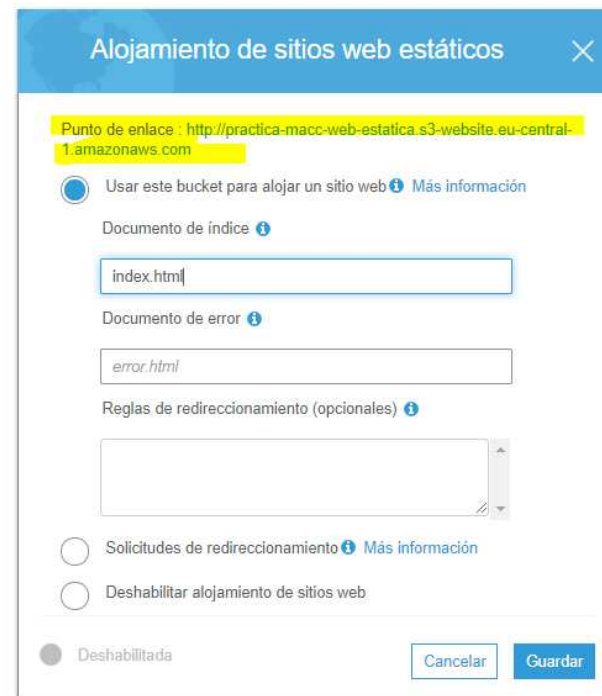
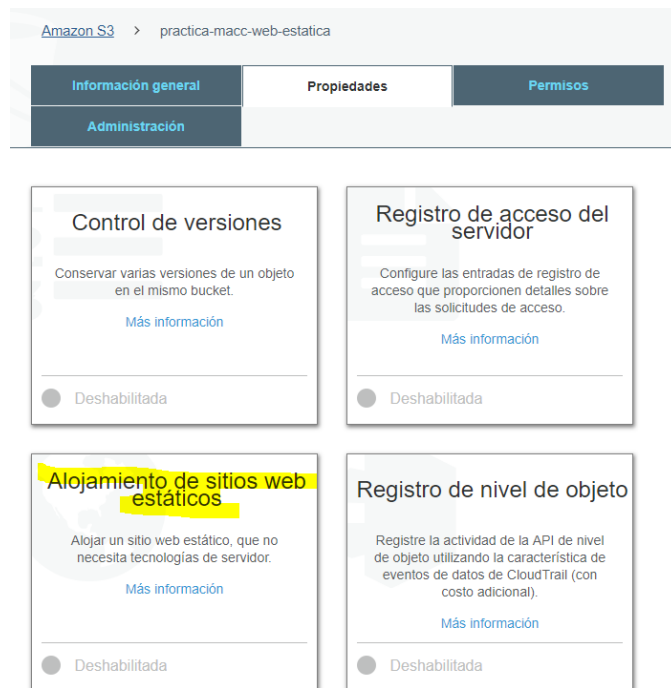
☐ Deshabilitar alojamiento de sitios web

☐ Deshabilitada



## 2.2 Capa de almacenamiento

- Ejercicios 2: Web hosting de páginas estáticas
  - Crear un Budget
    - configurar sus propiedades para que sea para alojamiento web
      - Os indica la URL endpoint a utilizar por la página WEB
    - Configurar acceso publico



## 2.2 Capa de almacenamiento

- Ejercicios 2: Web hosting de páginas estáticas
  - Modificar , crear imagenes



## 2.2 Capa de almacenamiento

- Ejercicios 2: Web hosting de páginas estáticas
  - Copiar carpeta de imágenes, modificar permisos y actualizar la página

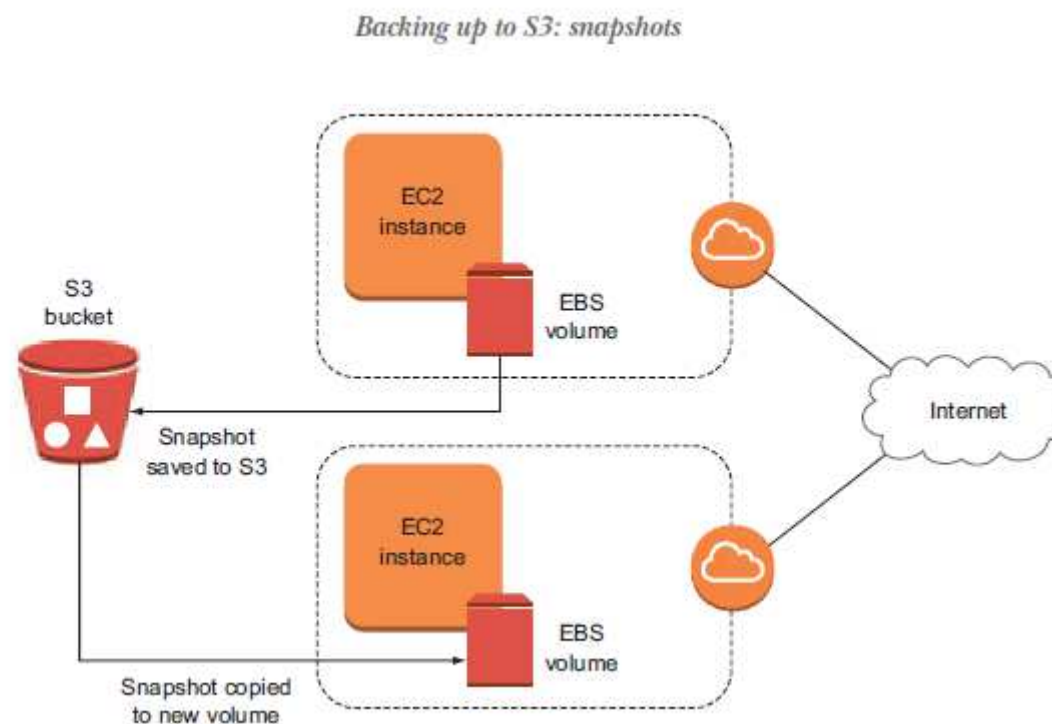


## 2.2 Capa de almacenamiento

- Ejercicios 3: Backup de datos
  - Backup del sistema de ficheros de una instancia EC2
    1. Crear snapshot
      1. Crear una nueva imagen
    2. Crear AMI
  - Backup de ficheros del SO
    - Mover ficheros grandes dentro de AWS no es tan lento!!!

## 2.2 Capa de almacenamiento

- Ejercicios 3: Backup de datos
  - Crear snapshot de una instancia Ec2 y crear una imagen con dicho snapshot



## 2.2 Capa de almacenamiento

- Ejercicios 3: Backup de datos
  - Backup de ficheros del SO
    - ¿Si un servidor se avería que ficheros nos gustaría utilizar en una nueva versión? ¿De que haríamos el backup?
    - Suponer un servidor donde los desarrolladores tienen acceso
      - /home/ (datos)
      - /usr (aplicaciones)
      - /var (aplicación web)
      - /etc (ficheros de configuración)
    - Acceder via SSH al servidor y comprimir las carpetas seleccionadas para el backup

```
$ cd ~
```

```
$ tar czf mybackup.tar.gz /etc /var /home
```

```
$ tar ztf mybackup.tar.gz|
```

## 2.2 Capa de almacenamiento

- Ejercicios 3: Backup de datos
  - Backup de ficheros del SO
    - Para poder interactuar con los servicios AWS de forma programática instalar el AWS CLI (en la instancias Amazon Linux el CLI esta preinstalado)
    - Es necesario tener Python instalado

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o
```

```
$ unzip awscli-bundle.zip
```

```
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

- Configurar el CLI para utilizar con la cuenta correspondiente
  - Para utilizar el CLI se necesitan las credenciales de AWS , las claves de acceso remoto de CLI

```
$ aws configure
```

- Finalmente realizamos el backup a S3

## 2.2 Capa de almacenamiento

- Ejercicios 3: Backup de datos
  - Finalmente realizamos el backup a S3
  - CLI
    - \$aws s3 help
  - Crear el bucket
    - \$aws s3 mb s3://nombre\_bucket
  - Crear copia del fichero
    - \$aws s3 cp mybackup.tar.gz s3://nombre\_bucket
  - Sincronizar carpeta , solo subir los cambios
    - \$aws s3 sync ./ s3://nombre\_bucket
  - Descargar
    - \$aws s3 s3://nombre\_bucket cp mybackup.tar.gz
  - Borrar bucket
    - \$aws s3 rb s3://nombre\_bucket

Cargar + Crear carpeta Descargar Acciones
 EE.UU. Este (Norte de Virg

mostrando desde 1 de 16

<input type="checkbox"/>	Nombre	Última modificación	Tamaño	Clase de almacen
<input type="checkbox"/>	aws	--	--	--
<input type="checkbox"/>	cache	--	--	--
<input type="checkbox"/>	ssh	--	--	--
<input type="checkbox"/>	awscli-bundle	--	--	--
<input type="checkbox"/>	.bash_history	sept. 16, 2019 12:47:59 p. m. GMT+0200	279.0 B	Estándar
<input type="checkbox"/>	.bash_logout	sept. 16, 2019 12:47:59 p. m. GMT+0200	220.0 B	Estándar
<input type="checkbox"/>	.bashrc	sept. 16, 2019 12:47:59 p. m. GMT+0200	3.7 KB	Estándar
<input type="checkbox"/>	.mysql_history	sept. 16, 2019 12:47:59 p. m. GMT+0200	160.0 B	Estándar
<input type="checkbox"/>	.profile	sept. 16, 2019 12:47:59 p. m. GMT+0200	807.0 B	Estándar
<input type="checkbox"/>	.sudo_as_admin_successful	sept. 16, 2019 12:47:59 p. m. GMT+0200	0 B	Estándar
<input type="checkbox"/>	awscli-bundle.zip	sept. 16, 2019 12:47:59 p. m. GMT+0200	12.0 MB	Estándar
<input type="checkbox"/>	mybackup.tar.gz	sept. 16, 2019 12:45:50 p. m. GMT+0200	21.9 MB	Estándar
<input type="checkbox"/>	wordpress_domain.sh	sept. 16, 2019 12:47:59 p. m. GMT+0200	568.0 B	Estándar
<input type="checkbox"/>	wordpress_domain.sql	sept. 16, 2019 12:47:59 p. m. GMT+0200	262.0 B	Estándar









## 2.2 Capa de almacenamiento

- Ejercicios 3: Backup de datos
  - \$aws s3api help
  - Configurar bucket
    - \$aws s3api put-bucket-versioning --bucket bucket\_name --versioning-configuration Status=Enabled
    - \$aws s3api list-object-versions --bucket name
    - \$aws s3api get-object --bucket froga-s3-bucket-version --key froga.txt ./froga\_bi.txt --version-id gA4Bwuj.e.LlcoxpO28qVWwdumlb4W8O

```
ubuntu@ip-172-31-89-109:~$ aws s3api list-object-versions --bucket froga-s3-bucket-version
{
  "Versions": [
    {
      "LastModified": "2019-09-16T13:07:44.000Z",
      "VersionId": "dVMwF7Yd_YgFNJlgVt9Pa0rvoA0axX_N",
      "ETag": "\"a5549928c8c7c4eb0aaaf84600311f0c\"",
      "StorageClass": "STANDARD",
      "Key": "froga.txt",
      "Owner": {
        "DisplayName": "jaagibas",
        "ID": "a5f5fd55466d95921d0c9e74e8b0bcb85efeflaa2f0c69a1383682c79e02d2ef"
      },
      "IsLatest": true,
      "Size": 10
    },
    {
      "LastModified": "2019-09-16T13:05:25.000Z",
      "VersionId": "gA4Bwuj.e.LlcoxpO28qVWwdumlb4W8O",
      "ETag": "\"dd2c5ab3e577a0be935d22f4474c545f\"",
      "StorageClass": "STANDARD",
      "Key": "froga.txt",
      "Owner": {
        "DisplayName": "jaagibas",
        "ID": "a5f5fd55466d95921d0c9e74e8b0bcb85efeflaa2f0c69a1383682c79e02d2ef"
      },
      "IsLatest": false,
      "Size": 12
    }
  ]
}
```

Amazon S3 > froga-s3-bucket-version > froga.txt

froga.txt Última versión ▾

sept. 16, 2019 3:07:44 p. m. GMT+0200 (Última versión)	Estándar	 
sept. 16, 2019 3:05:25 p. m. GMT+0200	Estándar	 
sept. 16, 2019 3:05:12 p. m. GMT+0200	Estándar	 
<a href="#">Más información</a>		<a href="#">Más información</a>
<input checked="" type="radio"/> Estándar		<input type="radio"/> Ninguno

## 2.2 Capa de almacenamiento

- Ejercicios 4: Archivando ficheros, reglas de ciclo de vida
  - Crear una regla en un bucket de S3 para pasar los datos de S3 a Glacier tras pasar X días

	S3	Glacier
Cost per GB	\$0.03 USD	\$0.01 USD
Accessibility	Immediate upon request	3–5 hours after request
Durability	Designed annual durability of 99.999999999%	Designed annual durability of 99.999999999%

## 2.2 Capa de almacenamiento

- Ejercicios 4: Archivando ficheros, reglas de ciclo de vida
  - Configurando ciclo de vida S3-> Glacier

The screenshot shows the AWS S3 console interface. On the left, a list of buckets is displayed with checkboxes. The bucket 'practica-macc-web-estatica' is selected. On the right, a configuration panel for the selected bucket is shown. The 'Administración' (Administration) tab is active, displaying various settings. Below the configuration panel, a navigation bar includes tabs for 'Información general', 'Propiedades', 'Permisos', and 'Administ'. Under the 'Permisos' tab, there are buttons for 'Ciclo de vida', 'Replicación', 'Análisis', 'Métricas', and 'Inventari'. At the bottom, there is a section for managing lifecycle rules, including a '+ Añadir regla de ciclo de vida' button, an 'Editar' button, an 'Eliminar' button, and an 'Acciones' dropdown menu.

Política de bucket	No
Lista de control de acceso	1 Beneficiario(s)
Configuración de CORS	No

Administración	Ciclo de vida	Deshabilitada
	Replicación	Deshabilitada
	Análisis	Deshabilitada
	Inventario	Deshabilitada
	Métricas	Deshabilitada

Información general | Propiedades | Permisos | Administ

Ciclo de vida | Replicación | Análisis | Métricas | Inventari

+ Añadir regla de ciclo de vida | Editar | Eliminar | Acciones

## 2.2 Capa de almacenamiento

- Ejercicios 4: Archivando ficheros, reglas de ciclo de vida
  - Configurando ciclo de vida S3-> Glacier

The screenshot displays the AWS S3 Lifecycle Rule configuration interface, showing the 'Transiciones' (Transitions) and 'Vencimiento' (Expiration) steps.

**Step 1: Nombre y ámbito (Name and scope)**

- Nombre de regla: `archivado_glacier`
- Filtro: `prefijo *.html`

**Step 2: Transiciones (Transitions)**

- Transición de clases de almacenamiento: `Transición a Glacier después de 1 día`
- Selección: ☒ Versión actual, ☒ Versiones anteriores

**Step 3: Vencimiento (Expiration)**

- Configurar el vencimiento: ☒ Versión actual, ☒ Versiones anteriores
- Hacer que venza la versión actual del objeto: ☒ Después de 10 días desde la creación del objeto
- Eliminar definitivamente las versiones anteriores: ☒ Después de 10 días después de convertirse en una vers. anterior
- Limpiar los marcadores de eliminación de objetos que han vencido y las cargas multiparte incompletas: ☐ Limpiar marcadores de eliminación de objetos que han vencido, ☐ Limpiar cargas multiparte incompletas

**Warning:** La transición de objetos pequeños a Glacier Deep Archive aumentará los costos.

## 2.2 Capa de almacenamiento

- Ejercicios 4: Archivando ficheros, reglas de ciclo de vida
  - Recuperar datos de Glacier y reponerlos en S3
    - Abrir el bucket y elegir la acción restore

Q Escriba un prefijo y pulse Intro para buscar. Pulse ESC para borrar.

Cargar Crear carpeta Descargar Acciones ▾

UE (Fráncfort)

Mostrando desde 1 hasta 2

<input type="checkbox"/> Nombre ▾	Tamaño ▾	Clase de almacenamiento ▾
<input type="checkbox"/> img	--	--
<input type="checkbox"/> index.html	970.0 B	Estándar

Mostrando desde 1 hasta 2

Abrir  
Descargar como  
Obtener tamaño total  
Cambiar clase de almacenamiento  
**Restaurar**  
Cambiar cifrado  
Cambiar metadatos  
Añadir etiquetas  
Hacer público  
Cambiar nombre  
Eliminar

## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones
  - Despliegue de una Galería de fotos WEB donde las fotos se alojan en S3
  - Crear el bucket que utilizara la aplicación
    - `aws s3 mb s3://gallery-joseba`
  - Se utilizará el AWS SDK de Node.js
    - Instalar node
      - `$apt install nodejs`
      - `$apt install npm`
    - `$git clone https://github.com/AWSinAction/code.git`
    - `$cd code/chapter7/gallery`
    - `$npm install`
    - `$node server.js gallery-Joseba`
      - Lanza el servidor express en el puerto 8080
      - Modificar acl de acceso a servidor y security group para abrir el puerto 8080

## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones
  - Modificar el security group para abrir el puerto 8080
  - Ir a Ec2 y ver el id del security group del servidor
  - Ir al security group y modificarlo

Launch Instance ▾ Connect Actions ▾

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-0ebc2bb3233b0b1cc	t2.micro	us-east-1c	running	2/2 checks ...	None

Instance: i-0ebc2bb3233b0b1cc Public DNS: ec2-11

### Edit inbound rules

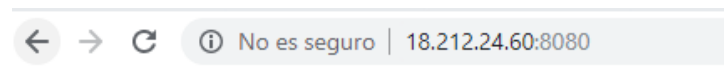
Description	Status Checks	Monitoring	Tags	Type	Protocol	Port Range	Source	Description
Instance ID: i-0ebc2bb3233b0b1cc	Instance state: running	Instance type: t2.micro	Elastic IPs	HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Availability zone: us-east-1c				HTTP	TCP	80	Custom ::/0	e.g. SSH for Admin Desktop
Security groups: launch-wizard-12				SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Scheduled events: No scheduled events				HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
AMI ID: wordpress_macc				HTTPS	TCP	443	Custom ::/0	e.g. SSH for Admin Desktop
Platform: -				Custom TCP	TCP	8080	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
IAM role: -								
Key pair name: practicas_master								
Owner: 420693608596								
Launch time: September 12, 2019 at 4:50:05 PM UTC-2 (94 hours)								

### Security Groups associated with i-0ebc2bb3233b0b1cc

Ports	Protocol	Source	launch-wizard-12
80	tcp	0.0.0.0/0, ::/0	✓
22	tcp	0.0.0.0/0	✓
443	tcp	0.0.0.0/0, ::/0	✓

## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones
  - Probar aplicación



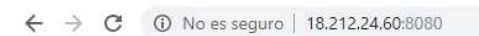
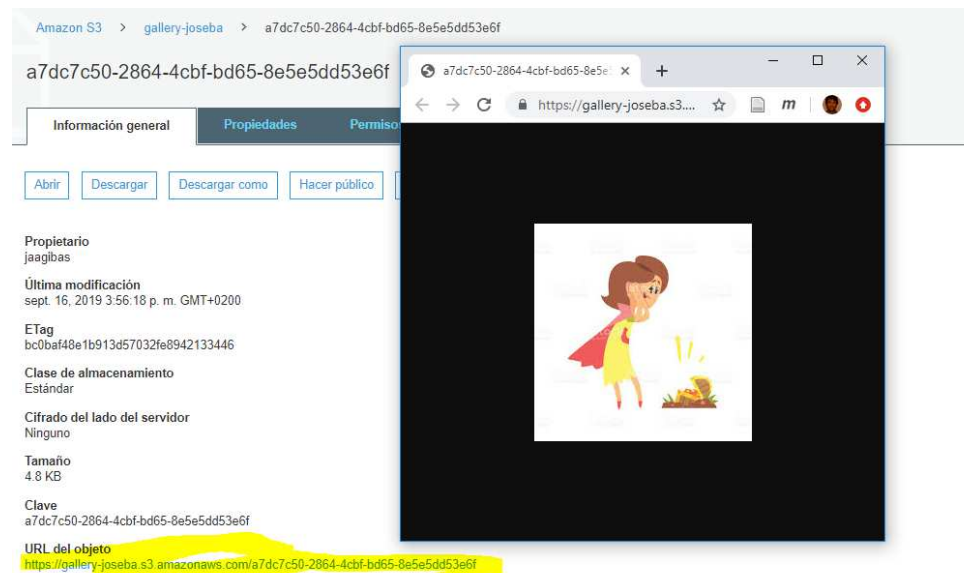
### Simple S3 Gallery

#### Upload

Seleccionar archivo Ningún archivo seleccionado

Upload

#### Images



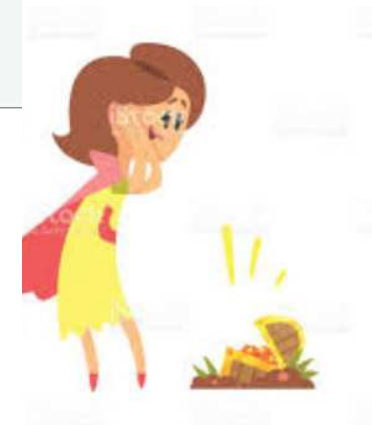
### Simple S3 Gallery

#### Upload

Seleccionar archivo Ningún archivo seleccionado

Upload

#### Images





## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones

- Código

```
var express = require("express");
var AWS = require("aws-sdk");
var mu = require("mu2-updated");
var uuid = require("uuid");
var multipart = require("multipart");

var app = express();
var s3 = new AWS.S3({
  "region": "us-east-1"
});

var bucket = process.argv[2];
if (!bucket || bucket.length < 1) {
  console.error("Missing S3 bucket. Start with node server.js BUCKETNAME instead.");
  process.exit(1);
}

function listImages(response) {

}

function uploadImage(image, response) {

}

app.get('/', function (request, response) {
  listImages(response);
});

app.post('/upload', function (request, response) {
  var form = new multipart.Form();
  form.on("part", function(part) {
    uploadImage(part, response);
  });
  form.parse(request);
});

app.listen(8080);

console.log("Server started. Open http://localhost:8080 with browser.");
```

## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones
  - Código

```
<html>
  <head>
    <title>Simple S3 Gallery</title>
  </head>
  <body>
    <h1>Simple S3 Gallery</h1>
    <h2>Upload</h2>
    <form action="/upload" method="post" enctype="multipart/form-data">
      <p><input type="file" name="file"></p>
      <p><input type="submit" value="Upload"></p>
    </form>
    <h2>Images</h2>
    {{#Objects}}
      <p></p>
    {{/Objects}}
  </body>
</html>
```

## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones
  - Código

```
function listImages(response) {  
  var params = {  
    Bucket: bucket  
  };  
  s3.listObjects(params, function(err, data) {  
    if (err) {  
      console.error(err);  
      response.status(500);  
      response.send("Internal server error.");  
    } else {  
      var stream = mu.compileAndRender(  
        "index.html",  
        {  
          Objects: data.Contents,  
          Bucket: bucket  
        }  
      );  
      stream.pipe(response);  
    }  
  });  
}
```

## 2.2 Capa de almacenamiento

- Ejercicios 4: Integrando S3 en las aplicaciones
  - Código

```
function uploadImage(image, response) {  
    var params = {  
        Body: image,  
        Bucket: bucket,  
        Key: uuid.v4(),  
        ACL: "public-read",  
        ContentLength: image.byteCount,  
        ContentType: image.headers["content-type"]  
    };  
    s3.putObject(params, function(err, data) {  
        if (err) {  
            console.error(err);  
            response.status(500);  
            response.send("Internal server error.");  
        } else {  
            response.redirect("/");  
        }  
    });  
}
```