

Desarrollo de Aplicaciones Web Empresariales

Java JEE - Spring & AWS Amazon



Java JEE Spring MVC & AWS Amazon Cloud

Tabla de contenido.

<u>Introducción.</u>	2
<u>Instalar Java versión 8.</u>	4
<u>Instalar el Servidor Web Tomcat versión 7.0.</u>	9
<u>Instalar Eclipse versión Mars1 JEE Web Developer.</u>	13
<u>Importar y configurar el proyecto SEEFACTURE en el IDE Eclipse Mars.</u>	14
<u>Descripción del framework y la arquitectura MVC del proyecto SEEFACTURE.</u>	23
<u>Creación de cuenta de usuario en AWS Amazon de capa gratuita.</u>	27
<u>Creación de Instancia de Base de Datos SQL Server en AWS Amazon.</u>	32
<u>Creación de Instancia de Java Web Server Tomcat en AWS Amazon.</u>	38
<u>Despliegue del proyecto SEEFACTURE en la instancia AWS Amazon.</u>	42
<u>Cliente FTP WinSCP para la Instancia Web Server Tomcat en AWS Amazon.</u>	49
<u>Instalación del certificado SSL en AWS Amazon y configuración para HTTPS.</u>	57
<u>Descargar fuentes JAVA-Spring del proyecto SEEFACTURE, scrips SQL, videos, instaladores y demás recursos.</u>	71

Desarrollo de Aplicaciones web empresariales en Java

JEE para AWS Amazon

1. Introducción.

Durante este tutorial desarrollaremos una aplicación web empresarial de ventas y facturación electrónica, útil para cualquier negocio comercial; farmacia, restaurant, librería, lavandería, etc. negocios que tienen la necesidad de emitir comprobantes de pago electrónicos y declarar estos comprobantes a la administración tributaria central del gobierno.

La aplicación web está desarrollada con tecnología Java JEE, utilizando JSON y el framework de desarrollo Spring dentro de una adecuada arquitectura de software basada en el patrón de diseño MVC (Modelo Vista Controlador), además utilizaremos los patrones de diseño de software Value Object (Vo), Business Object (Bo), Data Access Object (Dao) y Singleton con el propósito de hacer una aplicación eficiente, robusta pero a la vez elegante y fácil de mantener.

Se utiliza JQuery y HTML5 para el desarrollo responsive del FrontEnd, es decir la interfaz de usuario será adaptable al dispositivo que lo presente como un smartphone, una tablet, una laptop o pc permitiendo un sencilla y elegante interfaz de interacción con la aplicación web.

La base de datos a utilizar será el SQL Server 15.0 (2019), SQL Server nos permite contar con una plataforma inicial gratuita (SQL Server Express) que es más que suficiente para la mayoría de aplicaciones empresariales, además SQL Server

Express nos asegura una escalabilidad a plataformas SQL Server de mayor rendimiento. Oracle también es una muy buena opción robusta y escalable cuyos costes suelen ser mucho mayores de los de SQL Server.

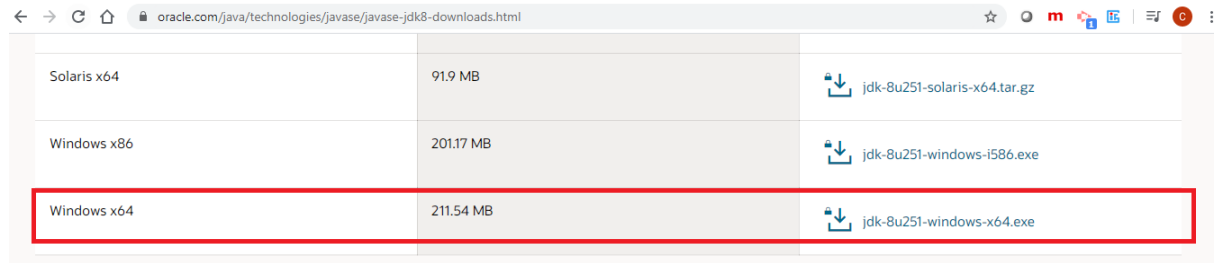
La aplicación SEEFATURE que se utiliza durante el tutorial es una aplicación comercial real que lo utilizan empresas de tamaño mediano (PYMES). Se vio por conveniente usar durante el tutorial una aplicación comercial real debido a que facilitara al estudiante la comprensión integral del desarrollo de aplicaciones web empresariales.

La aplicación Web se desplegará íntegramente en Amazon Web Services (AWS) plataforma plataforma en la nube más adoptada y completa en el mundo, que ofrece servicios integrales de Cloud Computing bastante robustos y profesionales usado por miles de empresas emergentes, compañías grandes y organismos gubernamentales. el uso de AWS nos permite reducir los costos, aumentar su agilidad e innovar de forma más rápida en nuestras aplicaciones empresariales. AWS posee una capa gratuita que se puede utilizar perfectamente para el desarrollo de aplicaciones empresariales sofisticadas, durante este tutorial utilizaremos esta capa gratuita de AWS.

2. Instalar Java versión 8.

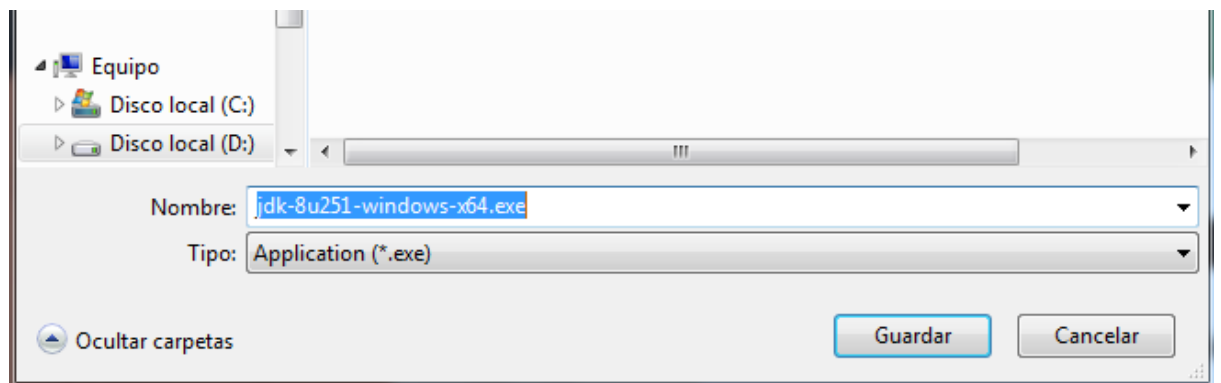
Descargar la versión 8 de Java del siguiente enlace:

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>



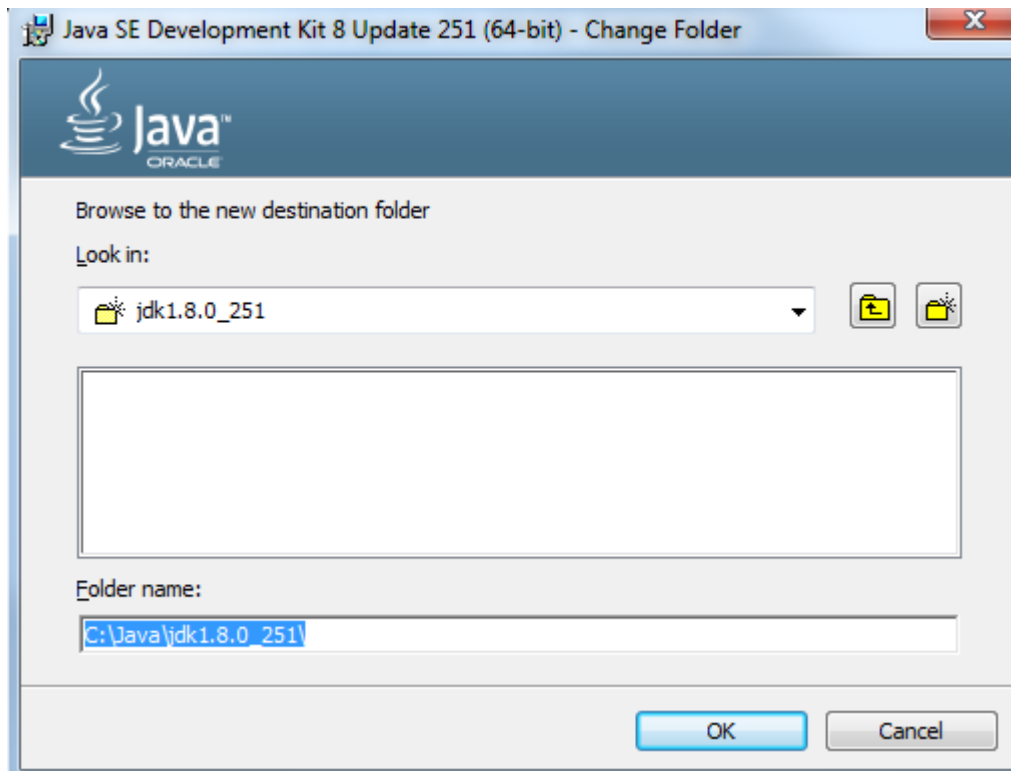
Solaris x64	91.9 MB	jdk-8u251-solaris-x64.tar.gz
Windows x86	201.17 MB	jdk-8u251-windows-i586.exe
Windows x64	211.54 MB	jdk-8u251-windows-x64.exe

Durante la descarga se le pedirá crear una cuenta oracle, la creación de esta cuenta es sencilla y no requiere pago.

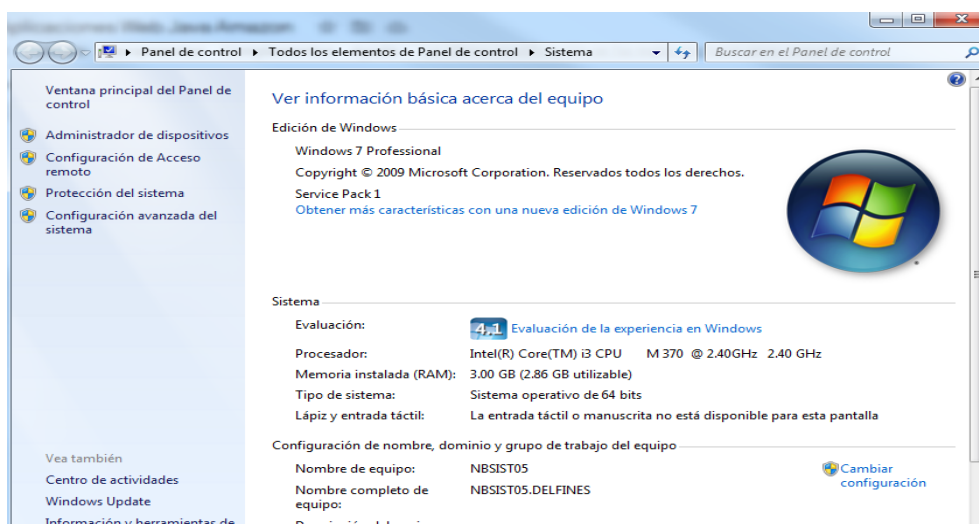


Se descarga el instalador del jdk 8.x para windows 64.

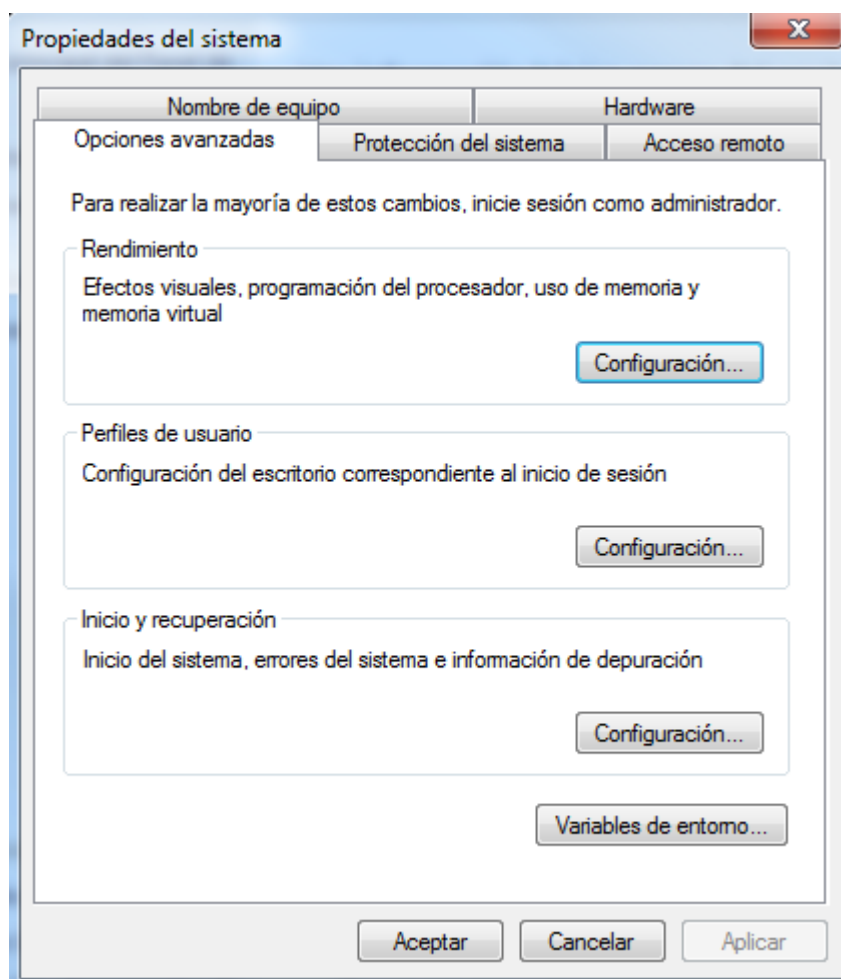
Instalar el jdk en la ruta `c:\Java\`. Por buena práctica se recomienda la instalación dentro de una carpeta sin espacios en blanco.



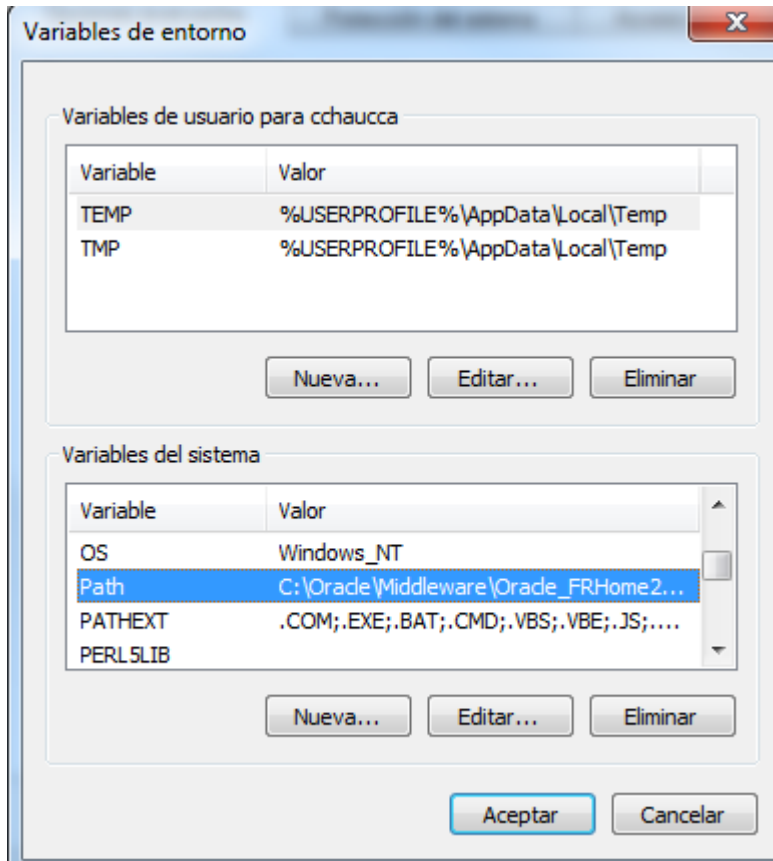
Luego de completada la instalación se debe asignar la ruta de instalación de java en la variable PATH del sistema operativo. En el panel de control del sistema ingrese a la opción configuración avanzada del sistema.



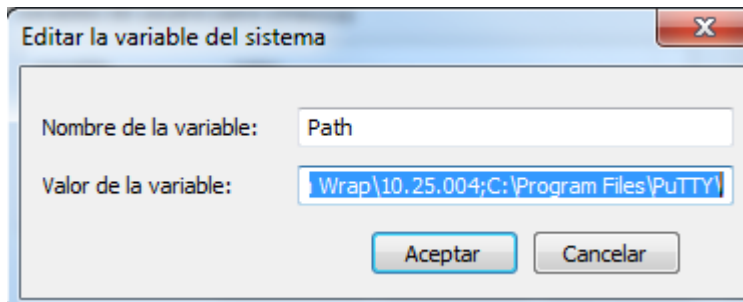
Se mostrará la pantalla "Propiedades del sistema"



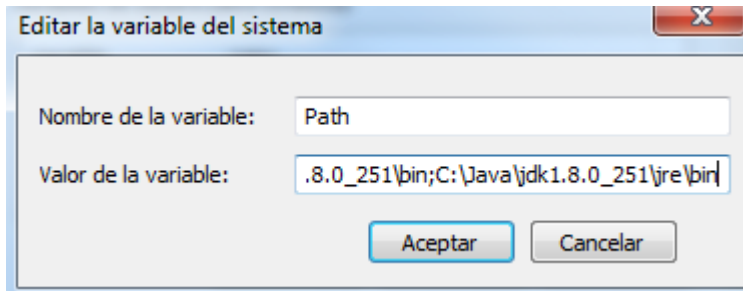
Click en el botón “Variables de entorno” se mostrará la pantalla de asignación de variables del sistema operativo.



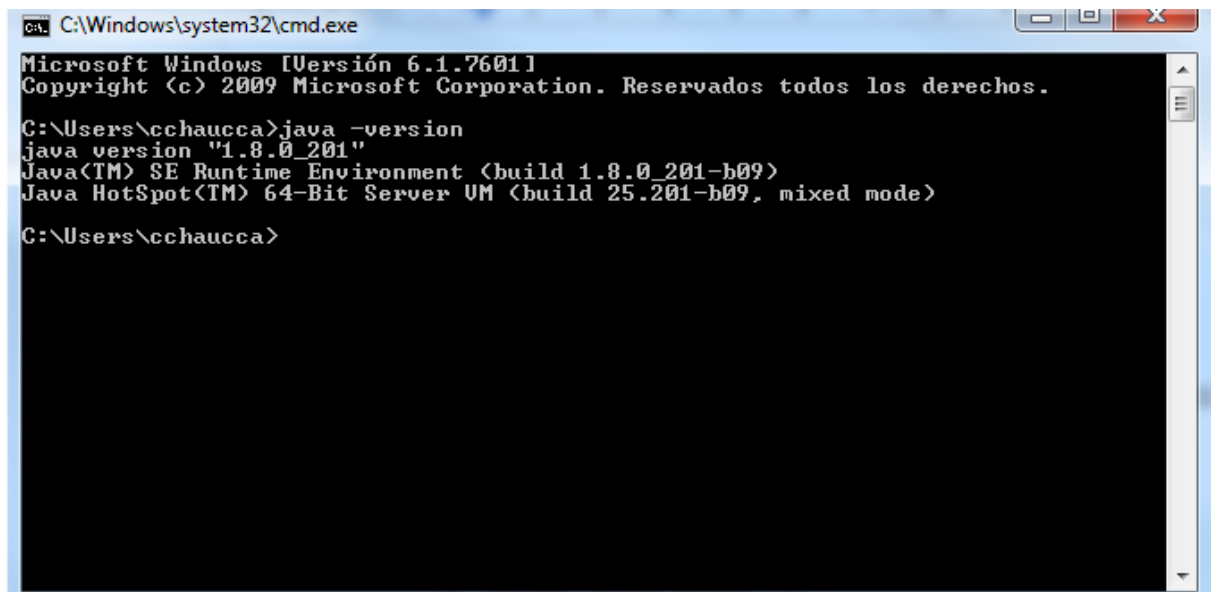
En la sección "Variables del sistema" seleccione la variable "Path" y haga click en el botón "Editar" para mostrar el formulario de edición de la variable "Path"



Agregarle la siguiente ruta al valor de la variable path.
;C:\Java\jdk1.8.0_251\bin;C:\Java\jdk1.8.0_251\jre\bin



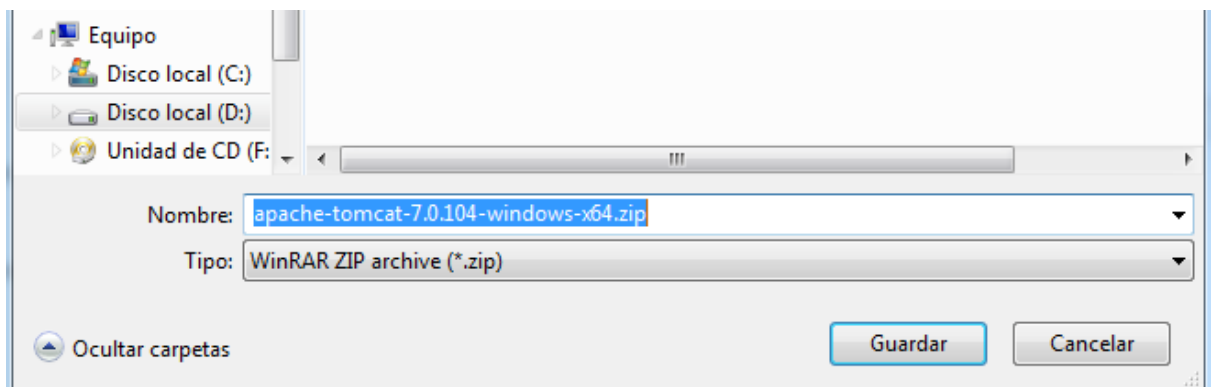
Para verificar la adecuada instalación de java en la versión 8.x ejecute la sentencia `java -versión` en la consola de Windows se debe mostrar la versión 1.8.0 como se muestra en la figura.



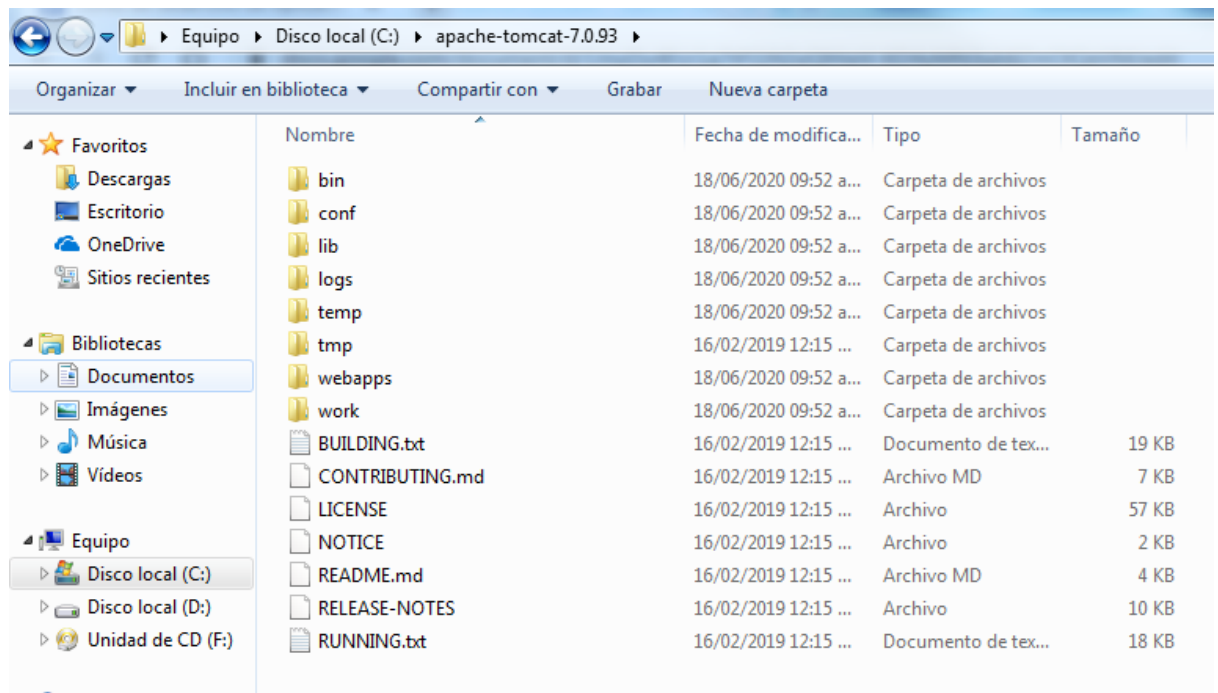
3. Instalar el Servidor Web Tomcat versión 7.0.

Descargar el Servidor Web Apache Tomcat 7.0 del siguiente enlace:

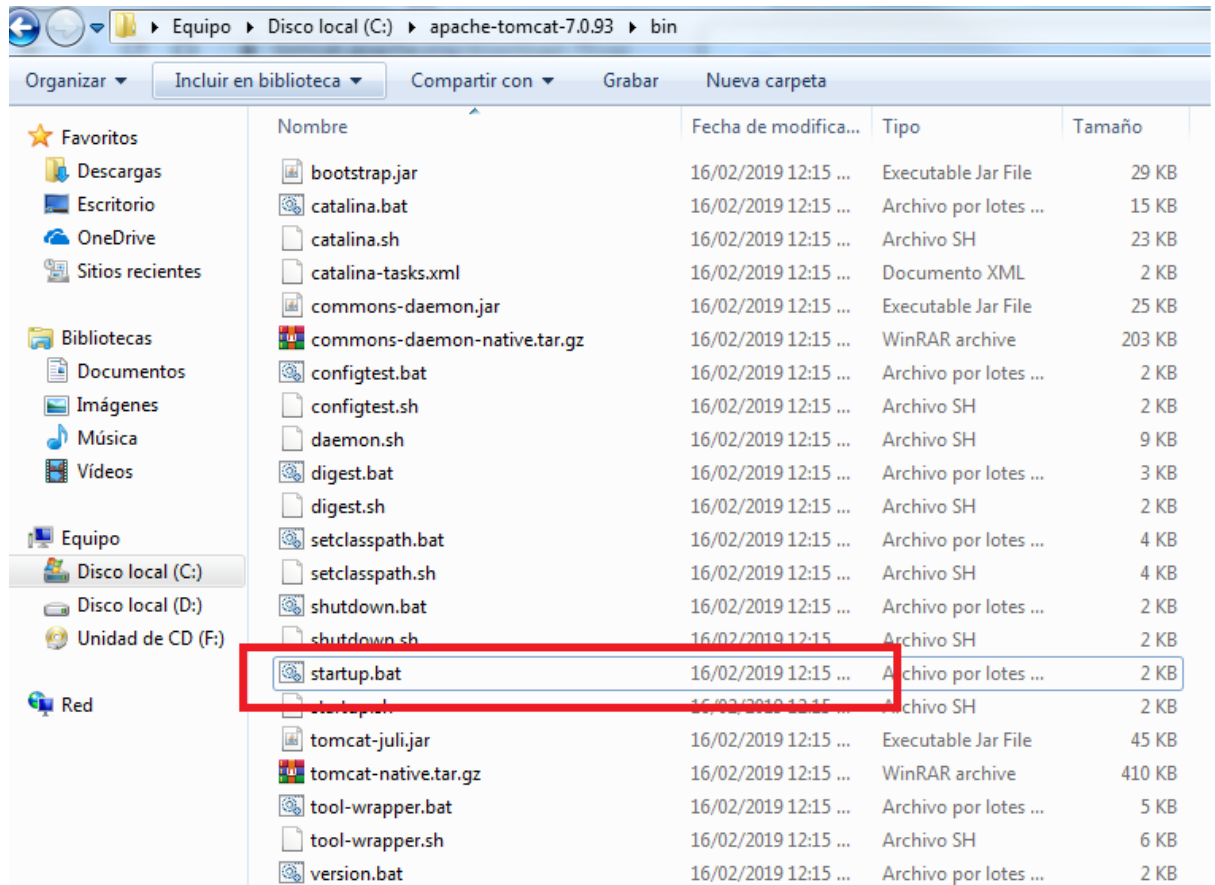
<https://tomcat.apache.org/download-70.cgi>



Al descargar la versión empaquetada .zip del software este no requiere ser instalado de la manera tradicional, este simplemente se descomprime en una unidad.



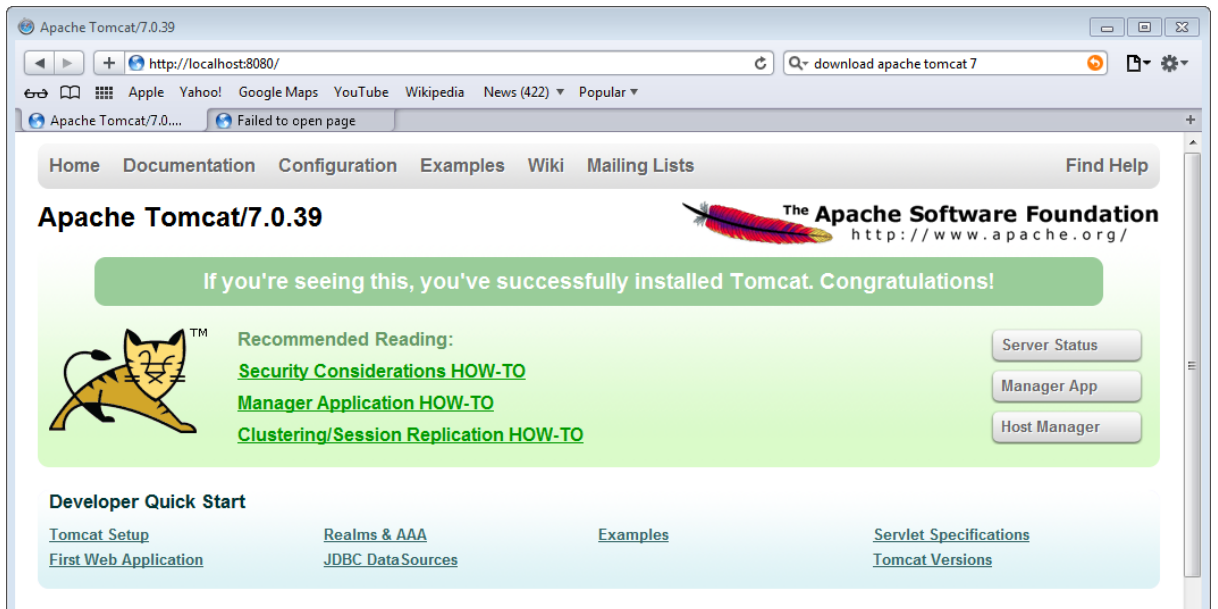
Para verificar su adecuado funcionamiento ingrese a la carpeta /bin de la carpeta que contiene Apache Tomcat. dentro de la carpeta /bin ejecute el programa "startup.bat"



Se debe mostrar la consola de ejecución del web server Apache Tomcat.

```
Tomcat
INFORMACI[3]N: Initializing ProtocolHandler ["http-bio-8080"]
jun 18, 2020 10:00:55 AM org.apache.coyote.AbstractProtocol init
INFORMACI[3]N: Initializing ProtocolHandler ["ajp-bio-8009"]
jun 18, 2020 10:00:55 AM org.apache.catalina.startup.Catalina load
INFORMACI[3]N: Initialization processed in 1455 ms
jun 18, 2020 10:00:55 AM org.apache.catalina.core.StandardService startInternal
INFORMACI[3]N: Arrancando servicio Catalina
jun 18, 2020 10:00:55 AM org.apache.catalina.core.StandardEngine startInternal
INFORMACI[3]N: Starting Servlet Engine: Apache Tomcat/7.0.93
jun 18, 2020 10:00:55 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Despliegue del directorio C:\apache-tomcat-7.0.93\webapps\docs de la aplicaci[3]n web
jun 18, 2020 10:00:56 AM org.apache.catalina.util.SessionIdGeneratorBase createSecureRandom
ADVERTENCIA: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [247] milliseconds.
jun 18, 2020 10:00:56 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: One or more Filters failed to start. Full details will be found in the appropriate container log file
jun 18, 2020 10:00:56 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: Fall[3] en arranque del Contexto [/docs] debido a errores previos
jun 18, 2020 10:00:56 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Deployment of web application directory C:\apache-tomcat-7.0.93\webapps\docs has finished in 1.004 ms
jun 18, 2020 10:00:56 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Despliegue del directorio C:\apache-tomcat-7.0.93\webapps\examples de la aplicaci[3]n web
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: One or more Filters failed to start. Full details will be found in the appropriate container log file
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: Fall[3] en arranque del Contexto [/examples] debido a errores previos
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Deployment of web application directory C:\apache-tomcat-7.0.93\webapps\examples has finished in 742 ms
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Despliegue del directorio C:\apache-tomcat-7.0.93\webapps\host-manager de la aplicaci[3]n web
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: One or more Filters failed to start. Full details will be found in the appropriate container log file
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: Fall[3] en arranque del Contexto [/host-manager] debido a errores previos
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Deployment of web application directory C:\apache-tomcat-7.0.93\webapps\host-manager has finished in 195 ms
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Despliegue del directorio C:\apache-tomcat-7.0.93\webapps\manager de la aplicaci[3]n web
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: One or more Filters failed to start. Full details will be found in the appropriate container log file
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: Fall[3] en arranque del Contexto [/manager] debido a errores previos
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Deployment of web application directory C:\apache-tomcat-7.0.93\webapps\manager has finished in 156 ms
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Despliegue del directorio C:\apache-tomcat-7.0.93\webapps\ROOT de la aplicaci[3]n web
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: One or more Filters failed to start. Full details will be found in the appropriate container log file
jun 18, 2020 10:00:57 AM org.apache.catalina.core.StandardContext startInternal
GRAVE: Fall[3] en arranque del Contexto [/] debido a errores previos
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.HostConfig deployDirectory
INFORMACI[3]N: Deployment of web application directory C:\apache-tomcat-7.0.93\webapps\ROOT has finished in 115 ms
jun 18, 2020 10:00:57 AM org.apache.coyote.AbstractProtocol start
INFORMACI[3]N: Starting ProtocolHandler ["http-bio-8080"]
jun 18, 2020 10:00:57 AM org.apache.coyote.AbstractProtocol start
INFORMACI[3]N: Starting ProtocolHandler ["ajp-bio-8009"]
jun 18, 2020 10:00:57 AM org.apache.catalina.startup.Catalina start
INFORMACI[3]N: Server startup in 2347 ms
```

Ingrese a un browser y coloque en la url : <http://localhost:8080/>

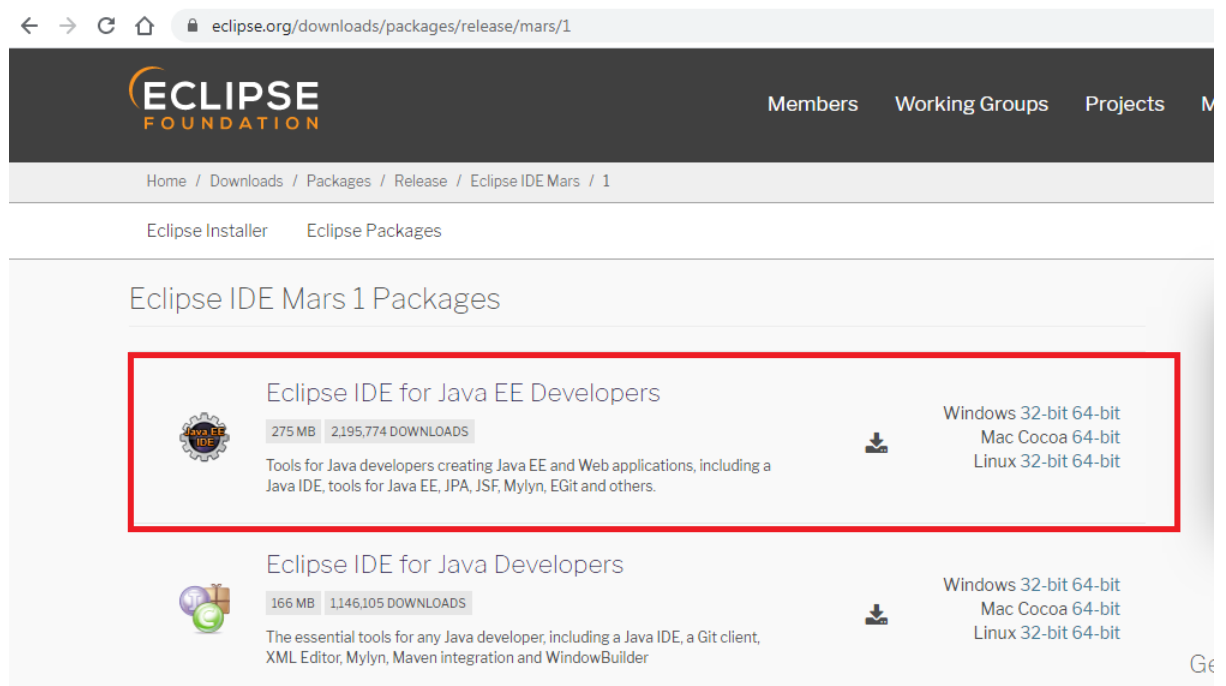


4. Instalar Eclipse versión Mars1 JEE Web Developer.

Descargar la versión Mars 1 de Eclipse del siguiente enlace

<https://www.eclipse.org/downloads/packages/release/mars/1>

Asegurate que sea la versión Java EE Developers, esta versión le permitirá desarrollar aplicaciones Web.



The screenshot shows the Eclipse website's download page for Eclipse IDE Mars 1 Packages. The page features a navigation bar with the Eclipse Foundation logo and links for Members, Working Groups, and Projects. Below the navigation bar, there is a breadcrumb trail: Home / Downloads / Packages / Release / Eclipse IDE Mars / 1. The main content area is titled 'Eclipse IDE Mars 1 Packages' and lists two options:

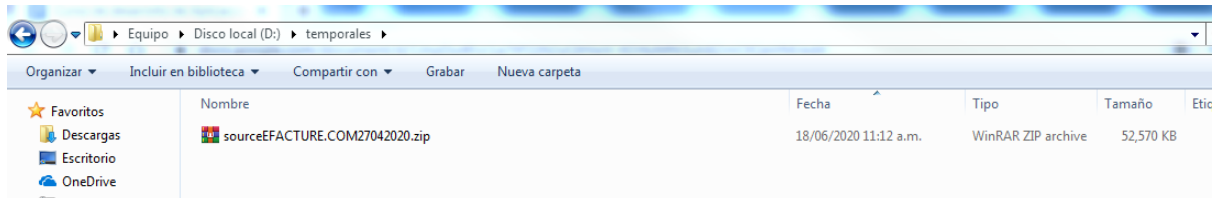
- Eclipse IDE for Java EE Developers**: 275 MB, 2,195,774 DOWNLOADS. Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others. Supported operating systems: Windows 32-bit 64-bit, Mac Cocoa 64-bit, Linux 32-bit 64-bit.
- Eclipse IDE for Java Developers**: 166 MB, 1,146,105 DOWNLOADS. The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder. Supported operating systems: Windows 32-bit 64-bit, Mac Cocoa 64-bit, Linux 32-bit 64-bit.

The first option, 'Eclipse IDE for Java EE Developers', is highlighted with a red box.

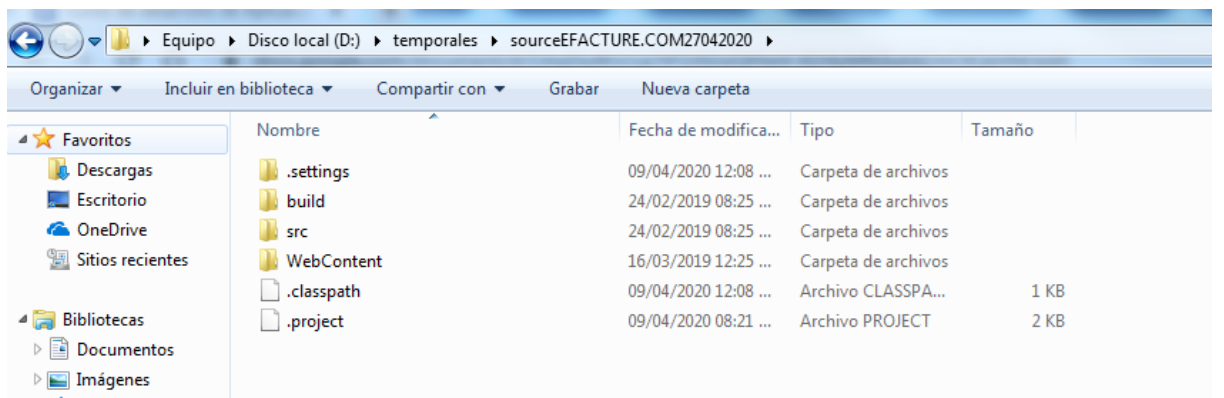


5. Importar y configurar el proyecto SEEFACURE en el IDE Eclipse Mars.

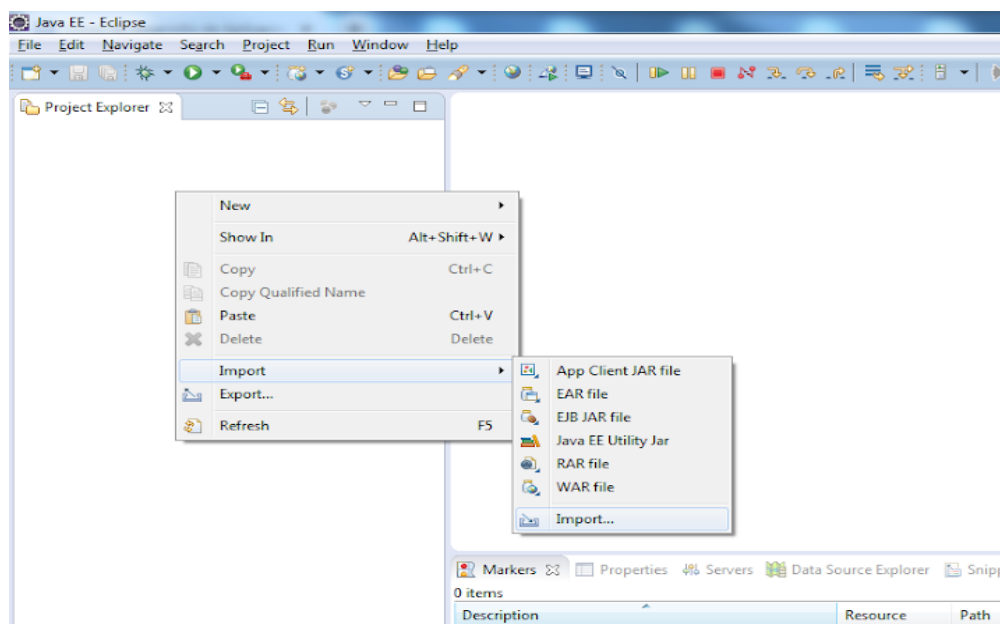
(Descargar las fuentes desde el repositorio indicado en el Anexo A)



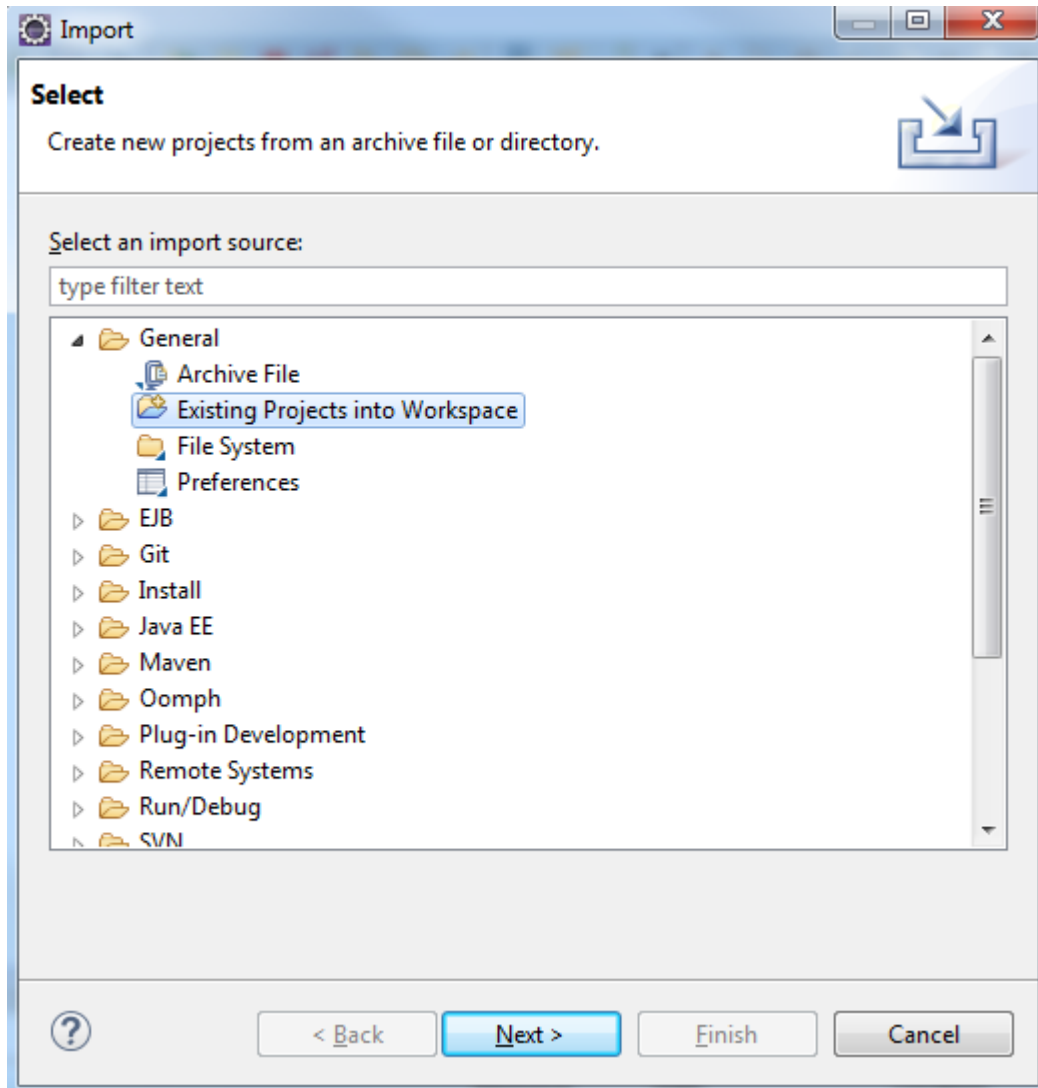
Descomprimir el proyecto descargado, debemos tener una estructura de carpetas similares a la siguiente imagen:



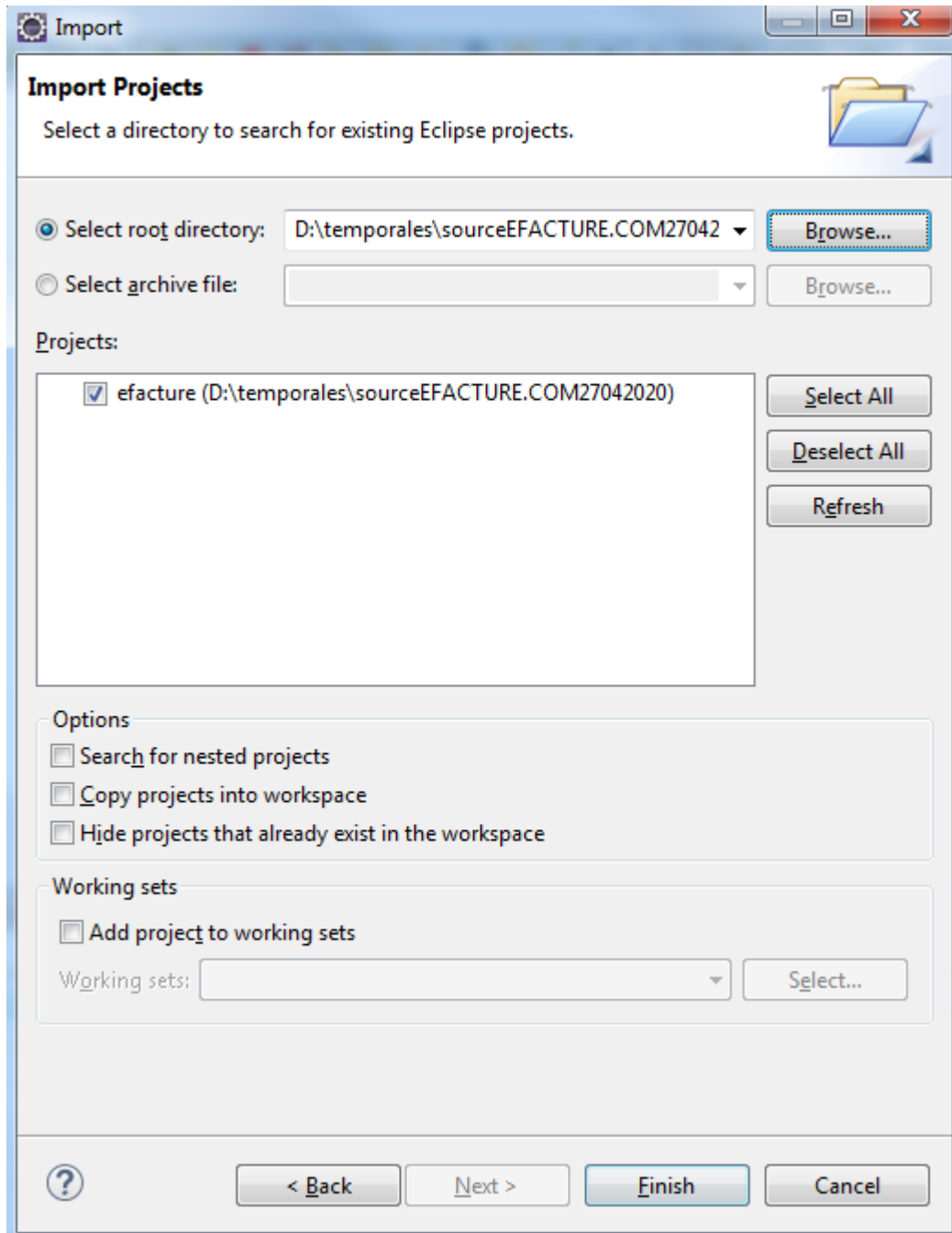
importe el proyecto en el IDE de eclipse



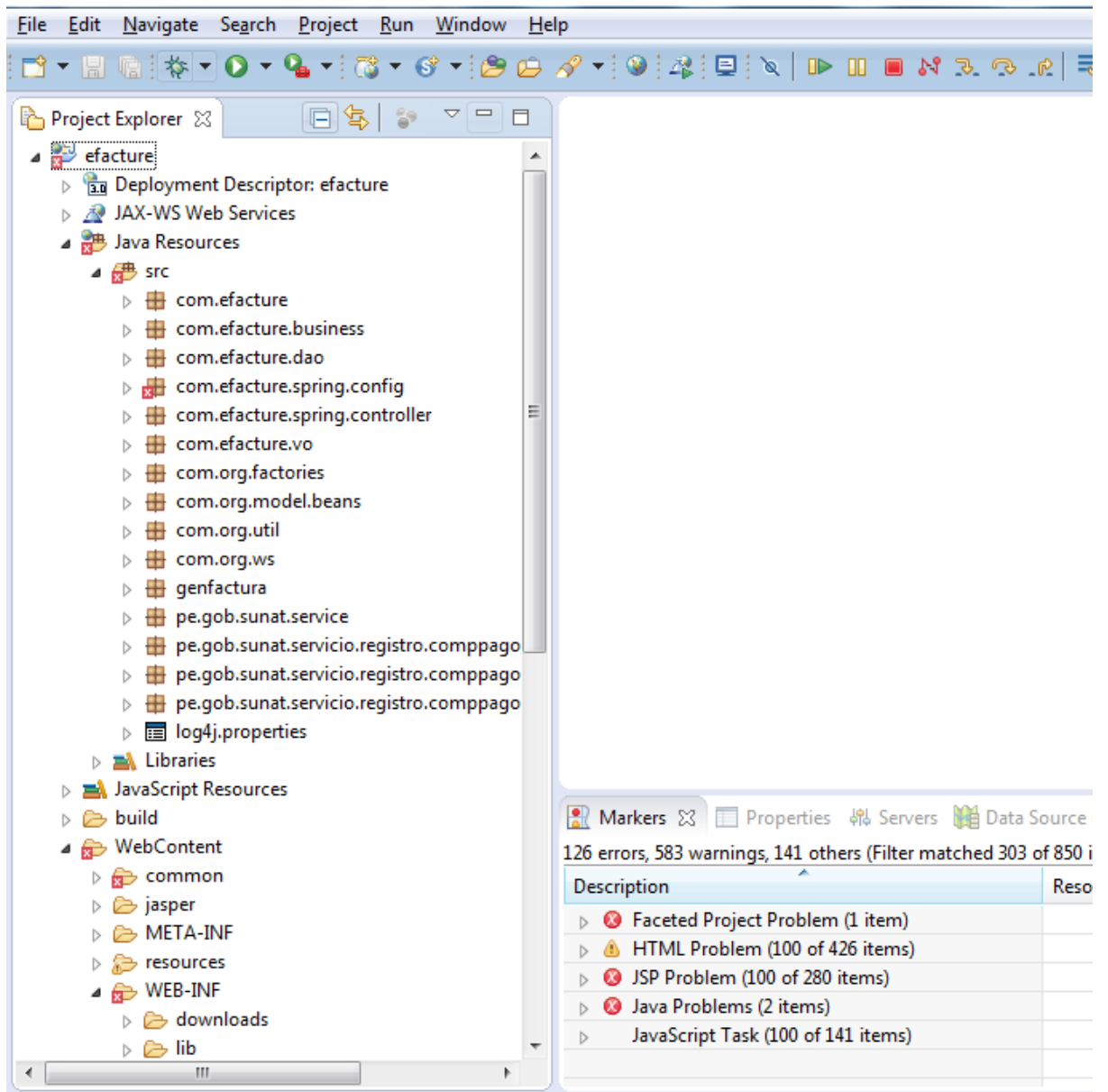
Seleccione la opción “Existing Project into Workspace”



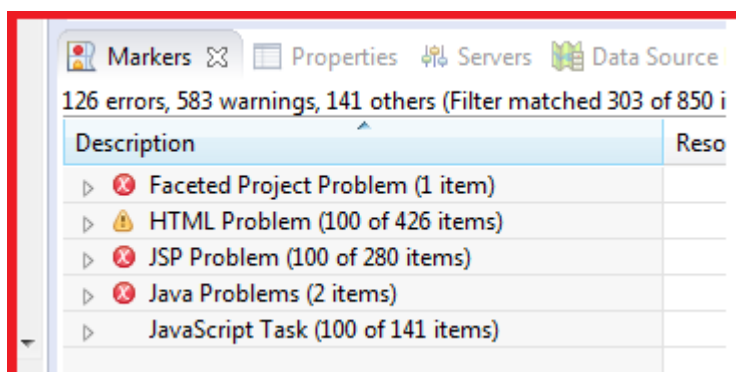
En la pantalla Import seleccione la carpeta donde se encuentra el proyecto efacture, para luego dar Click en el botón "Finish"



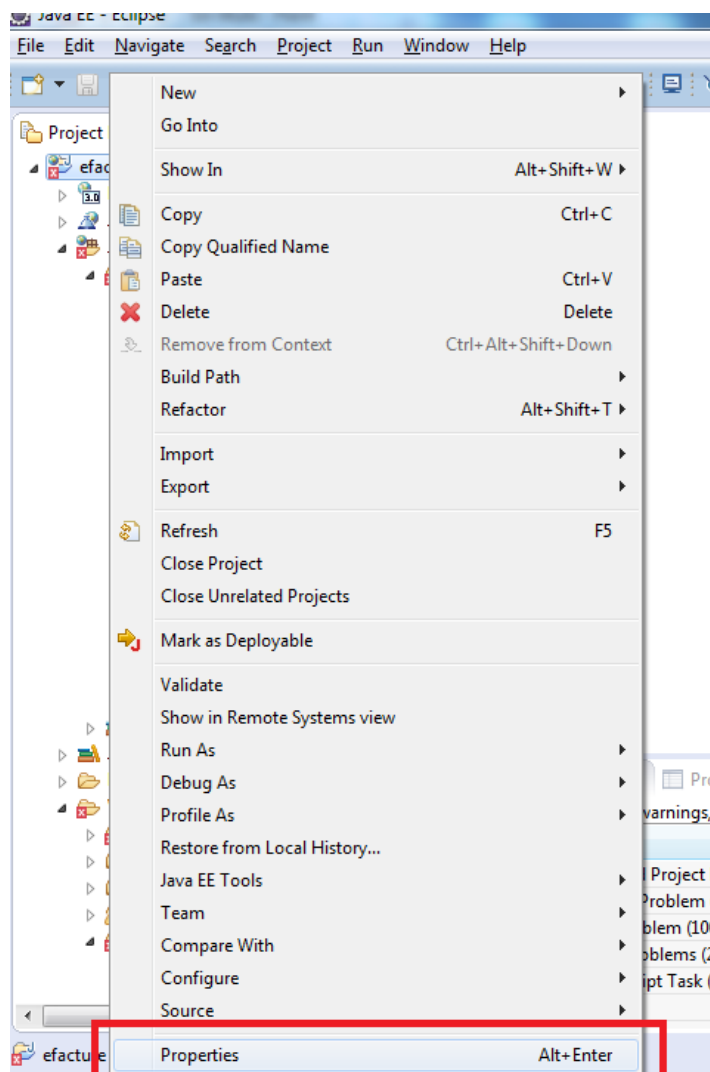
Se debe tener la siguiente estructura de proyecto



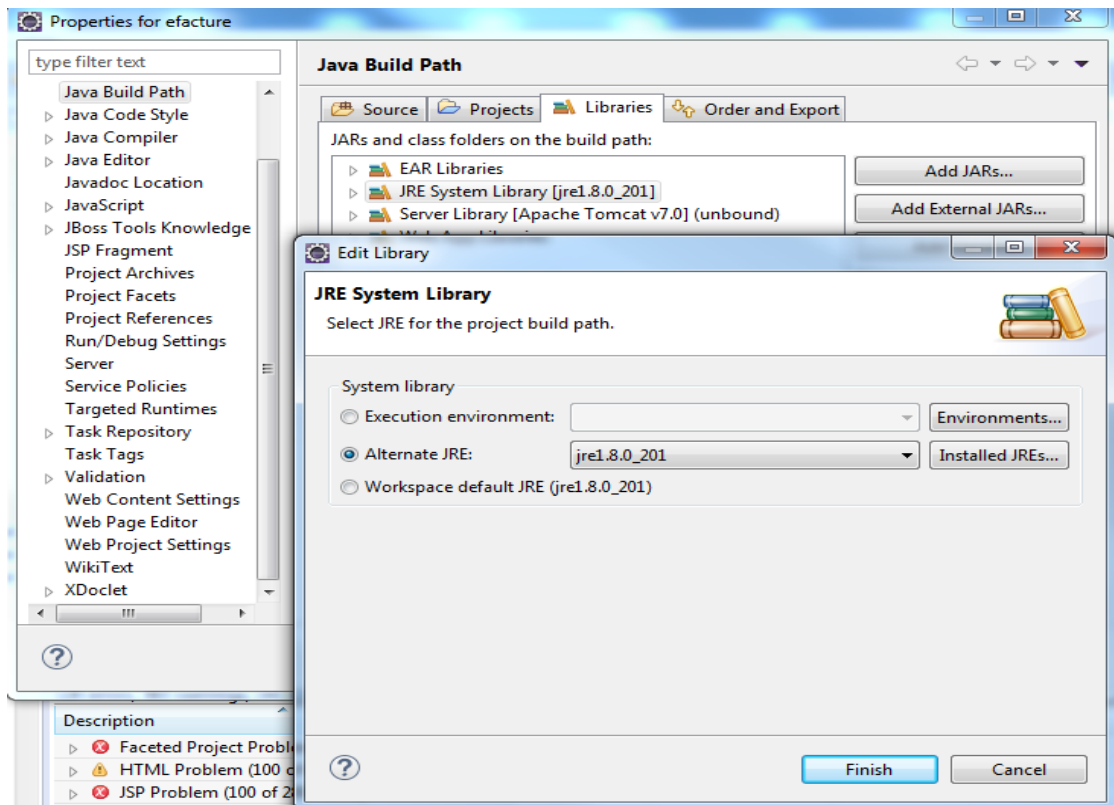
Se debe realizar configuraciones adicionales para evitar los errores que se visualizan en la sección Markets



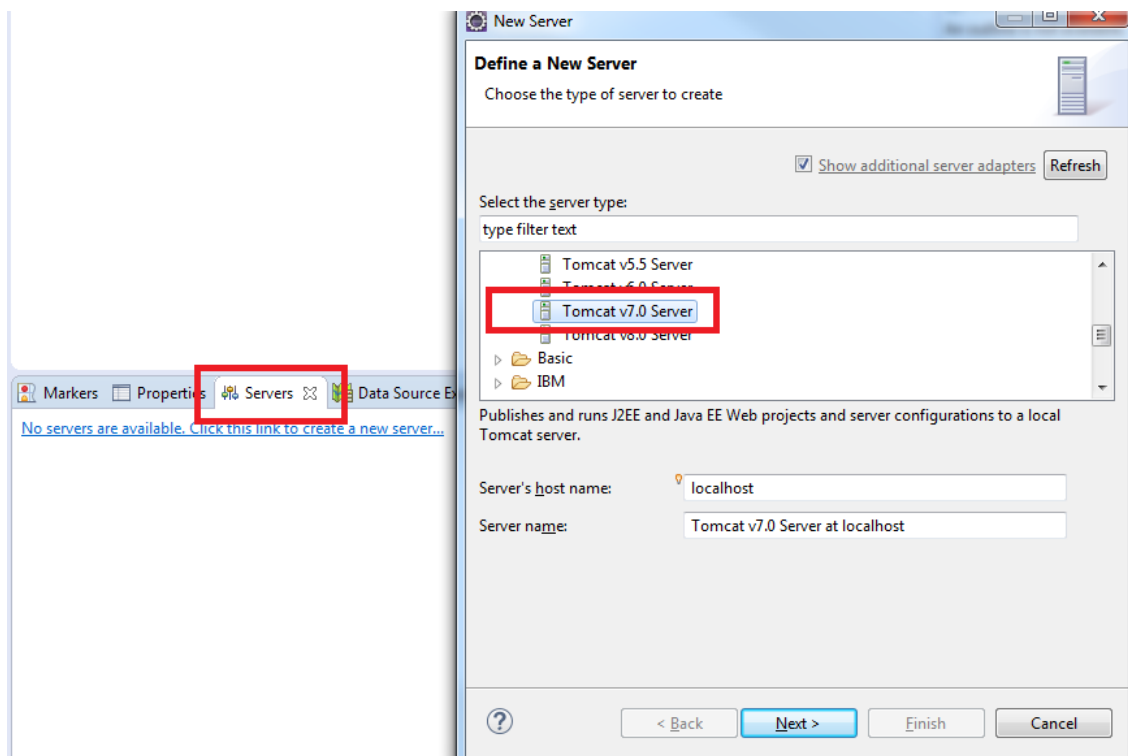
realice click derecho en el proyecto y seleccione la opción "Properties"



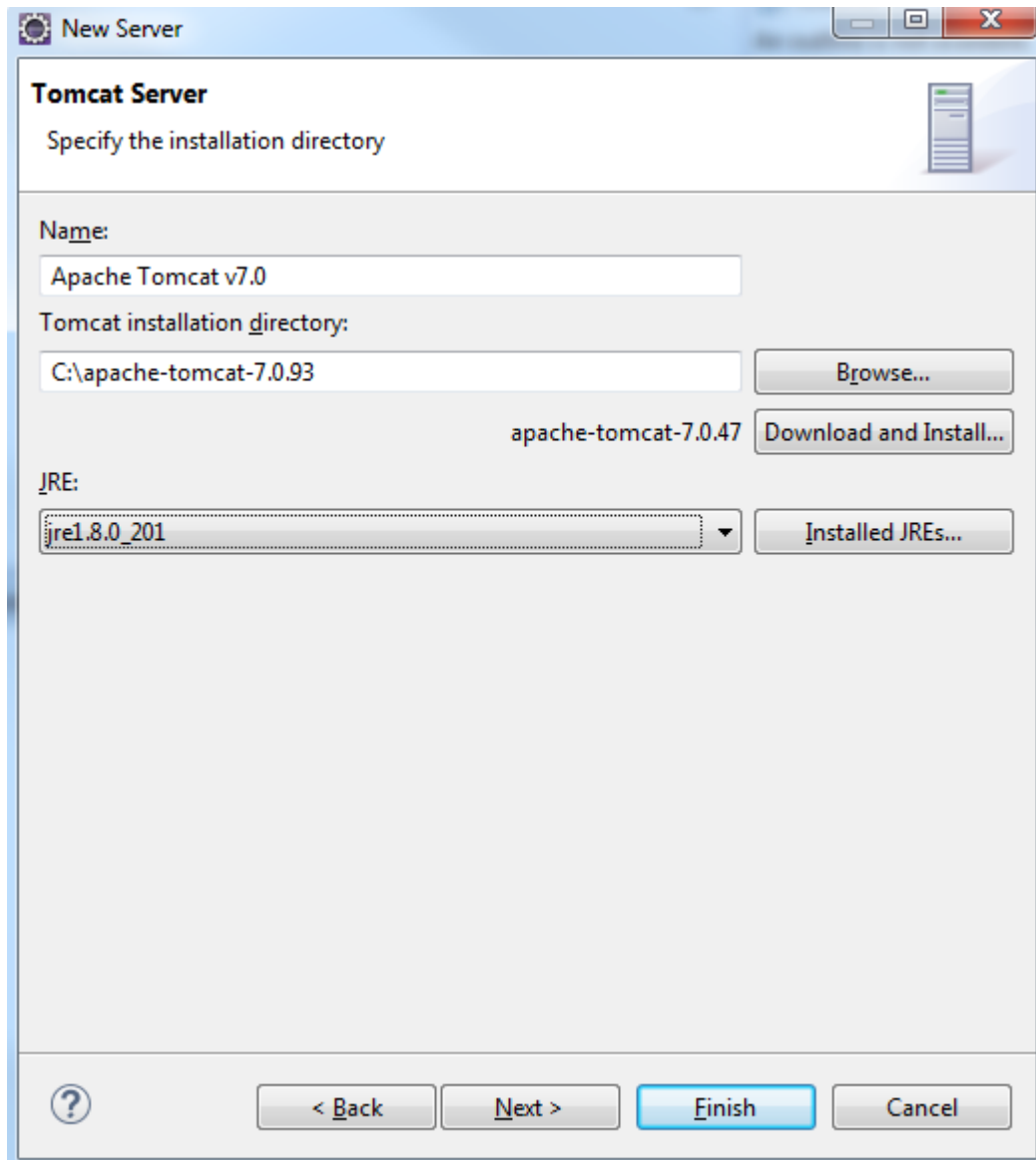
Asegúrese que el JRE sea el instalado cuya versión es 8.5



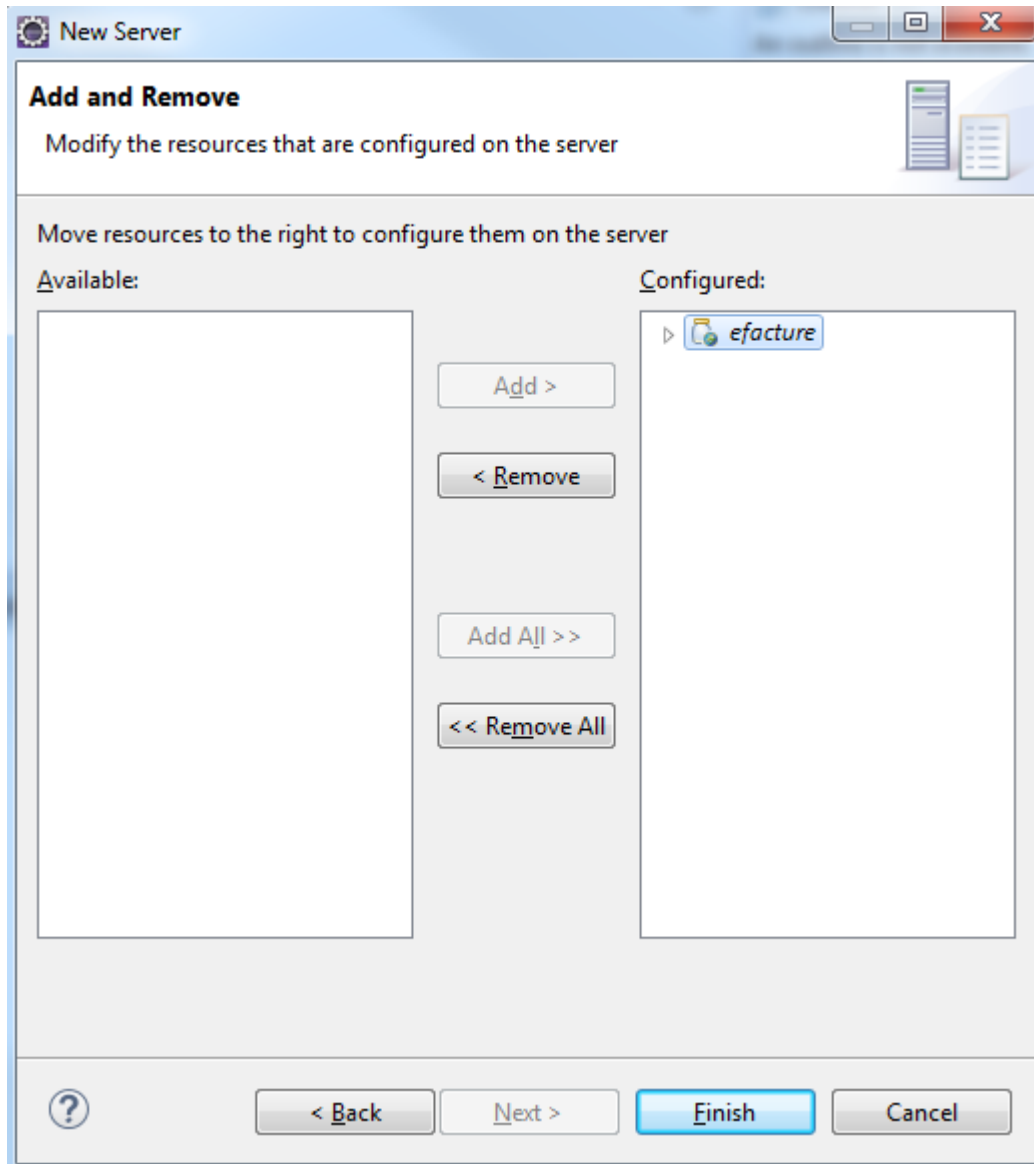
Asigne el Servidor Web Apache Tomcat a eclipse, en la sección "Servers"



Seleccione la carpeta donde se encuentra ubicado Apache Tomcat y el JRE 1.8 para luego dar click en el botón “Finish”.

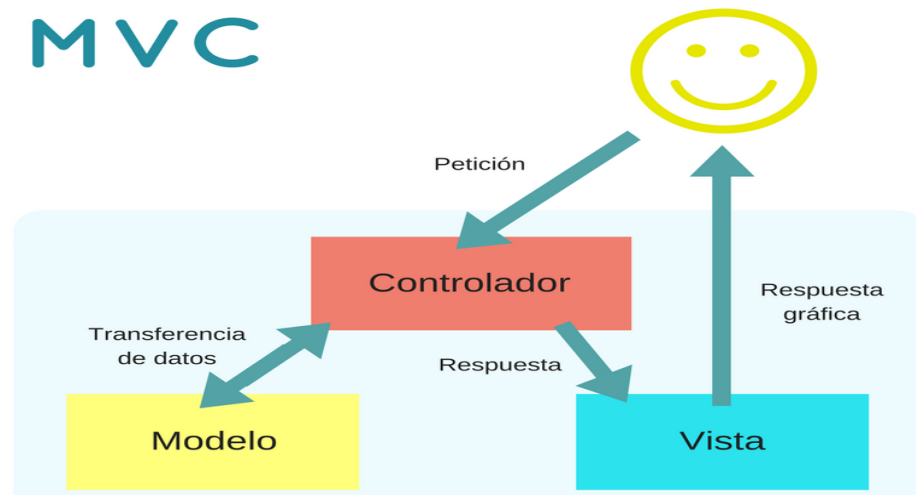


En la siguiente pantalla seleccione agregue el proyecto efacture y luego click en el botón “Finish”.



6. Descripción del framework y la arquitectura MVC del proyecto SEEFATURE.

La arquitectura del software se basa en el patrón de diseño de software MVC (Modelo Vista Controlador).



Este patrón de diseño de software es uno de los más usados, en la actualidad se puede encontrar en pequeños como en grandes sistemas de información web, en el mundo laboral es muy indispensable.

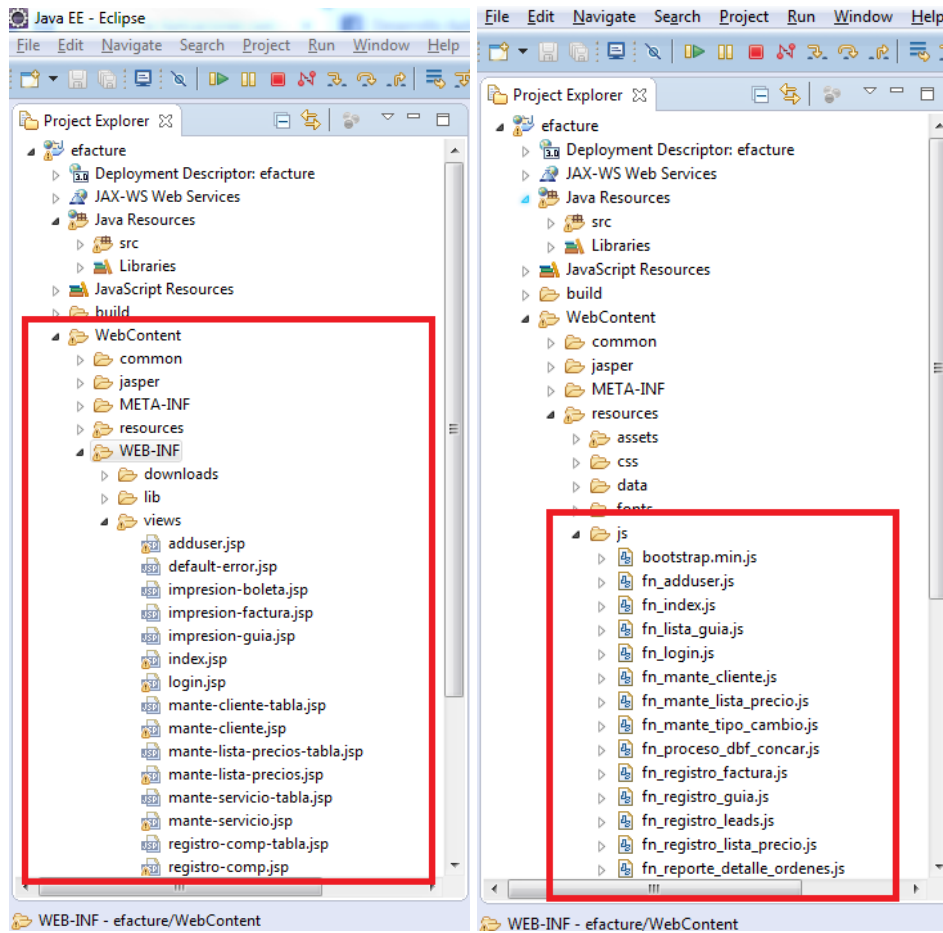
Vista: Este componente se encarga de mostrarle al usuario final las pantallas, ventanas, páginas y formularios. Desde la perspectiva del programador este componente es el que se encarga del FrontEnd la “Lógica de la Presentación”. Los artefactos de esta capa son: CSS, HTML, HTML5 y Javascript y las librerías asociadas a esta capa con JQuery, JSON y otras librerías que enriquecen la presentación de las aplicaciones web.

Controlador: Este componente se encarga de gestionar las peticiones (Request) que se reciben, atenderlas y procesarlas, por medio del Controlador se comunican el Modelo y la Vista. Los artefactos de esta capa son: Java y las librerías que permiten enriquecer la presentación de la aplicación como la generación de Reportes por JasperReport, generación de reportes de Business Intelligence, Lectura de Web Services, Excels y demás, que mejoren la presentación de los datos en la capa Vista.

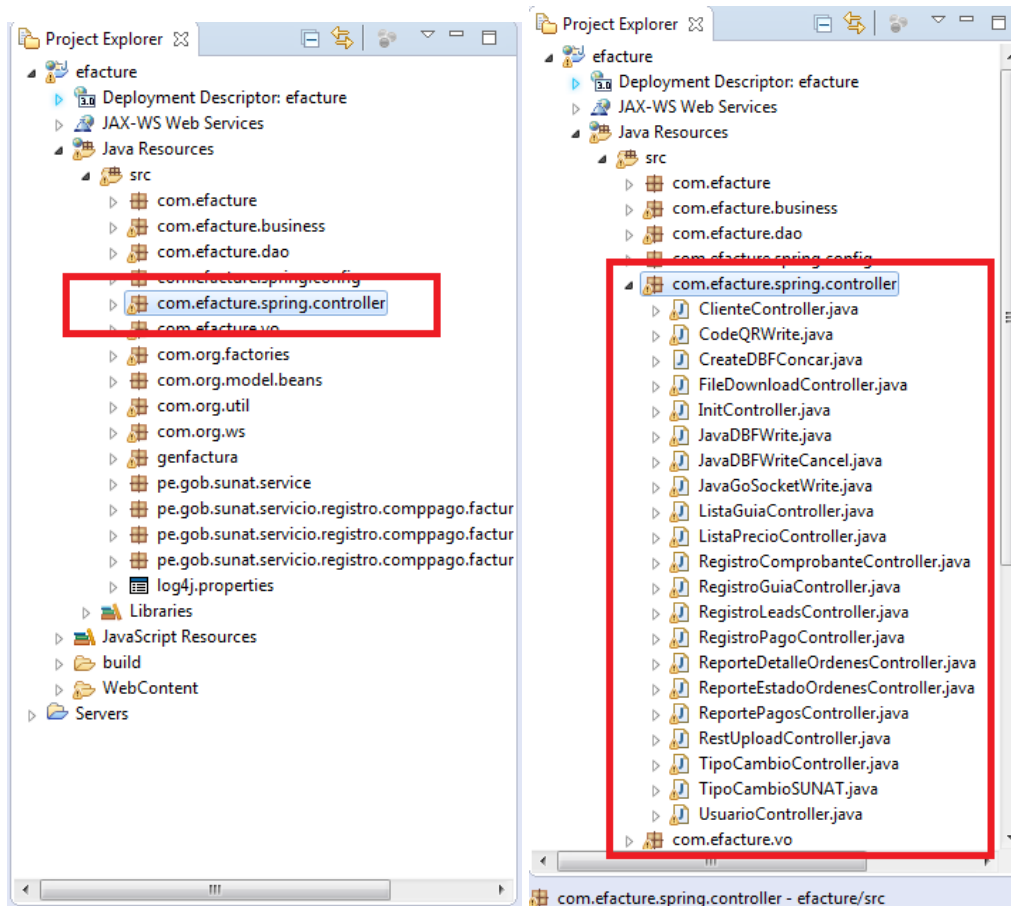
Modelo: Este componente se encarga de manipular, gestionar y actualizar los datos en la Base de Datos llamado también el BackEnd de la aplicación. Si se utiliza una base de datos aquí es donde se realizan las consultas, búsquedas, filtros y actualizaciones. Los artefactos de esta capa son: Java y las librerías que permiten accesos a Bases de Datos, Web Services, Sistemas Legados y demás que pueden almacenar los datos de la aplicación.

Asociaremos estas capas del modelo MCV con las fuentes del proyecto efacture:

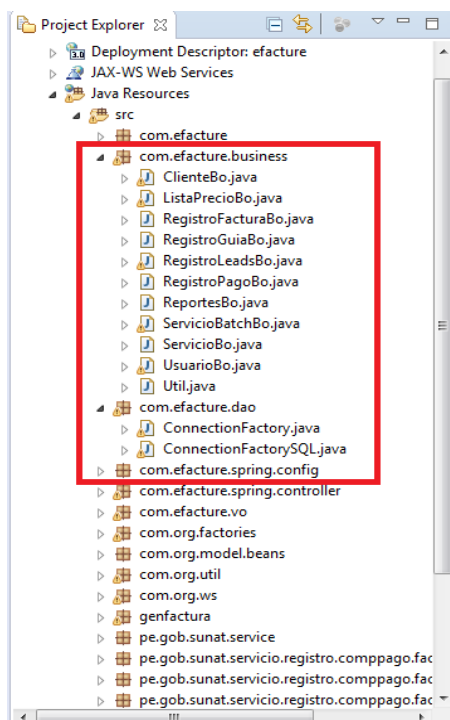
Capa Vista:



Capa Controlador:

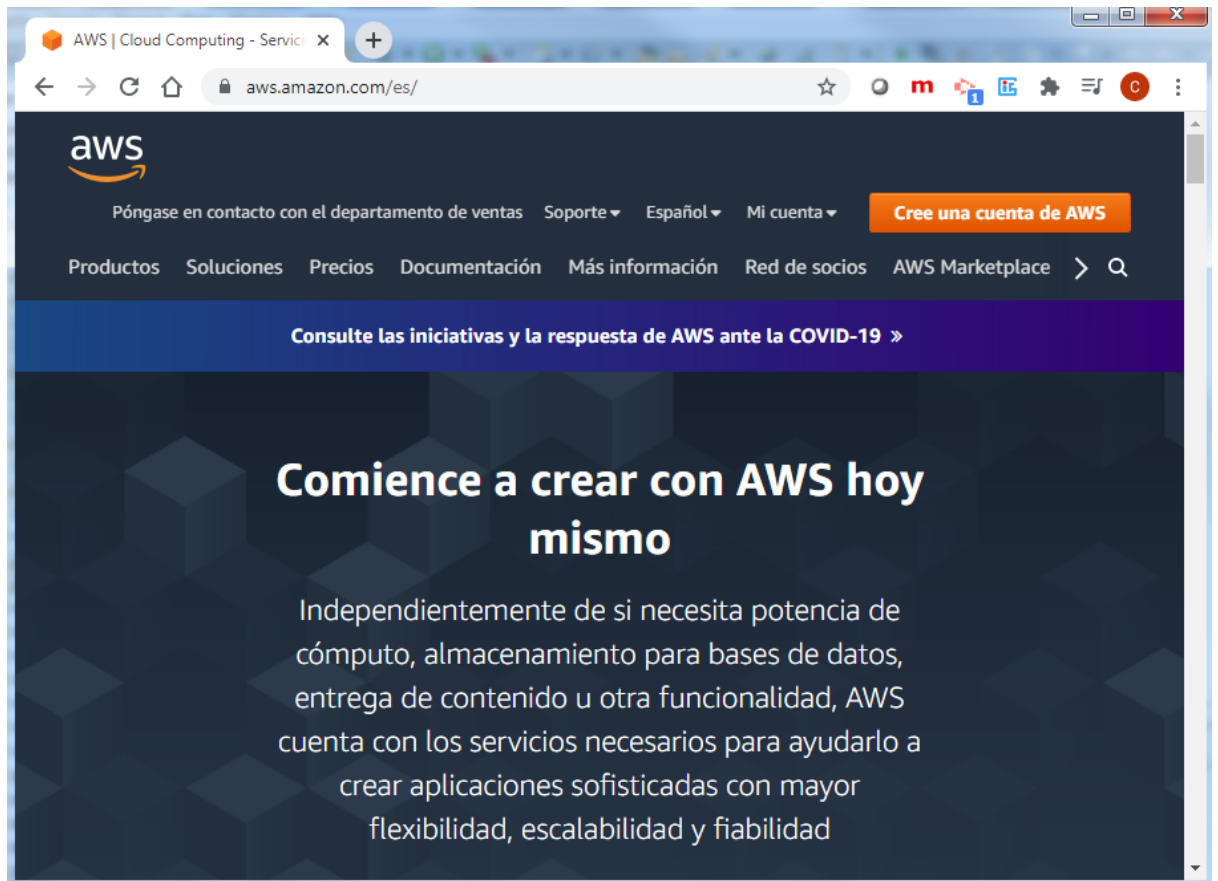


Capa Modelo:

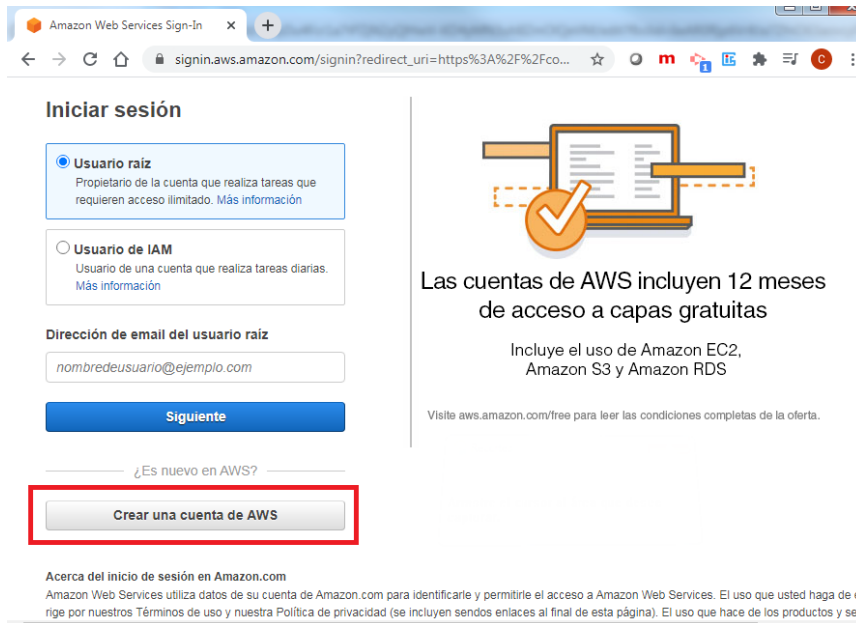


7. Creación de cuenta de usuario en AWS Amazon de capa gratuita.

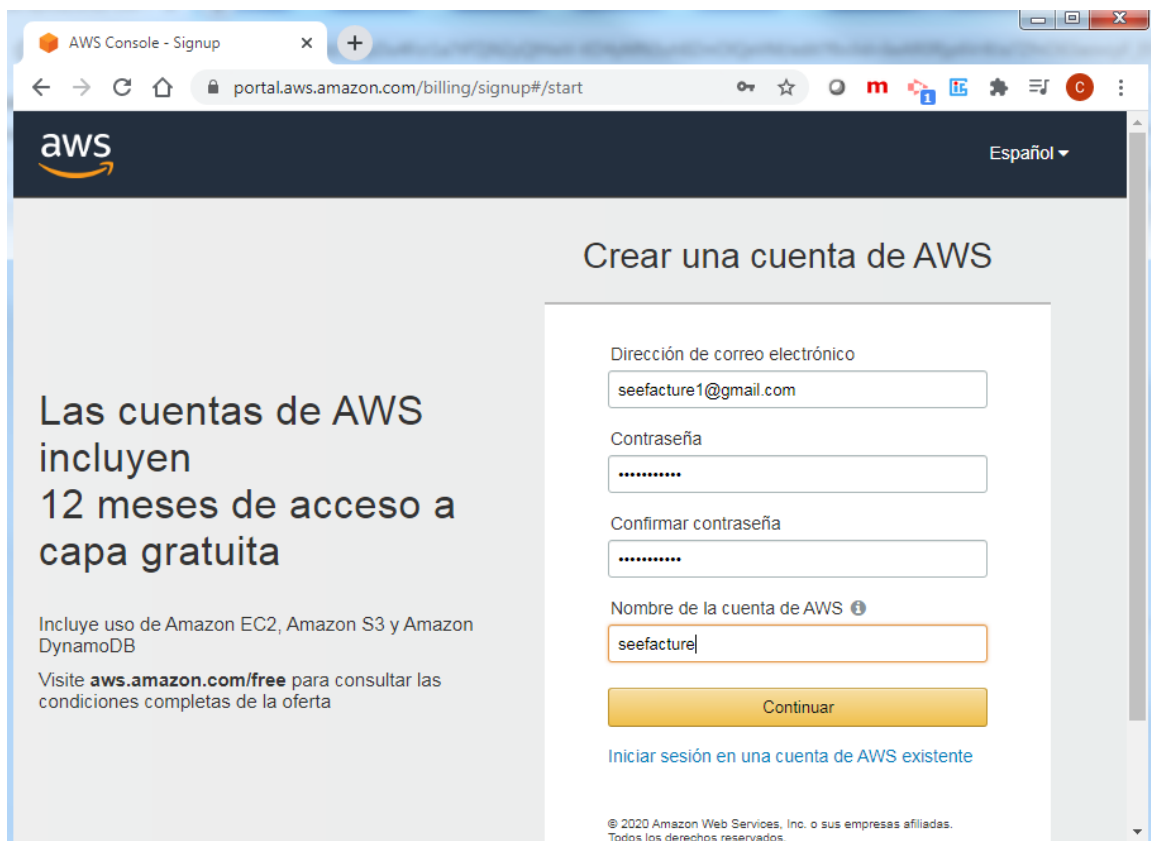
Ingresa a la pagina <http://aws.amazon.com/>



ingresa a “Mi cuenta”, submenu “Consola de administración de AWS”,



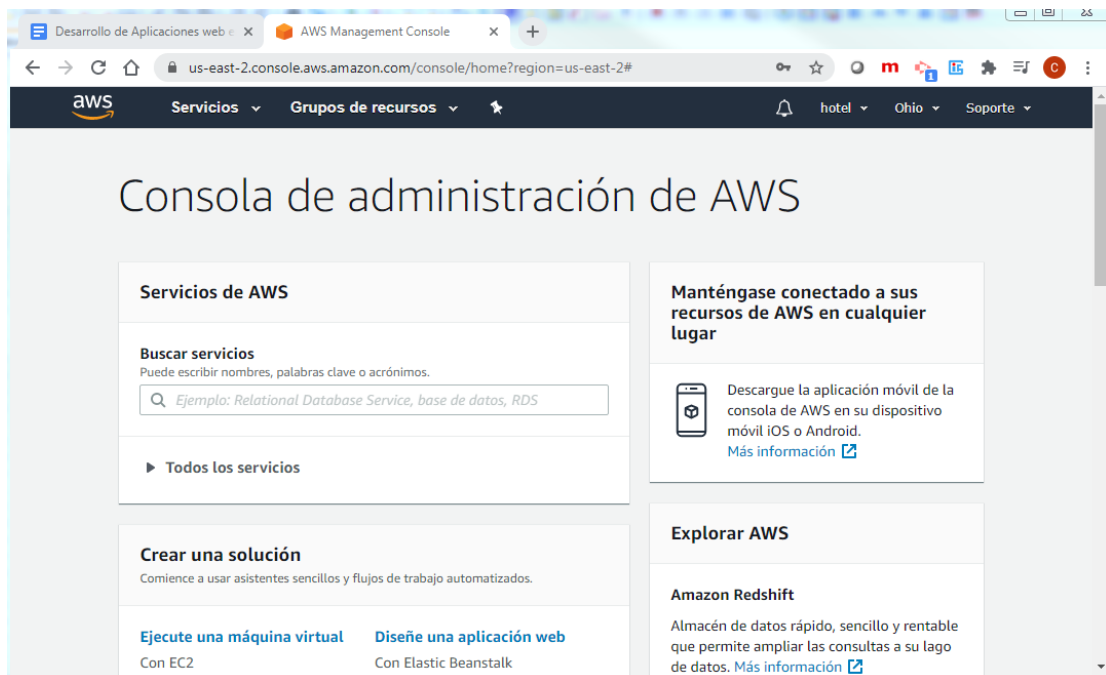
haga click en el botón “Crear una cuenta de AWS”,

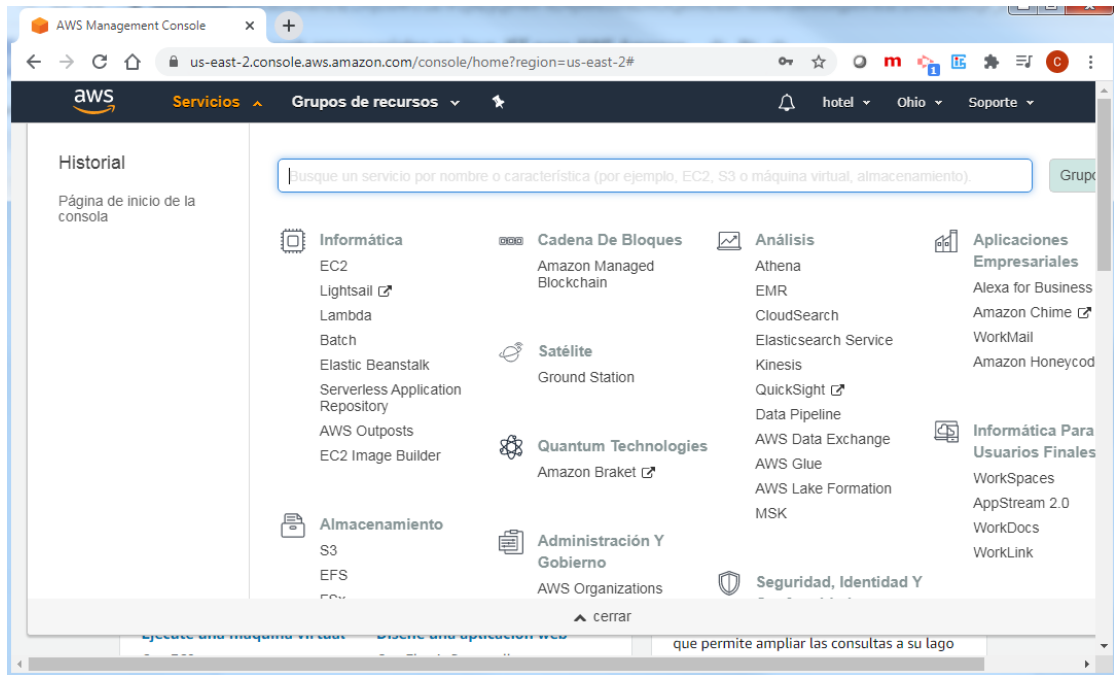


click en el botón “Continuar”, se te solicitará los datos de dirección, telefono y demás datos personales además de asignar una tarjeta de crédito.

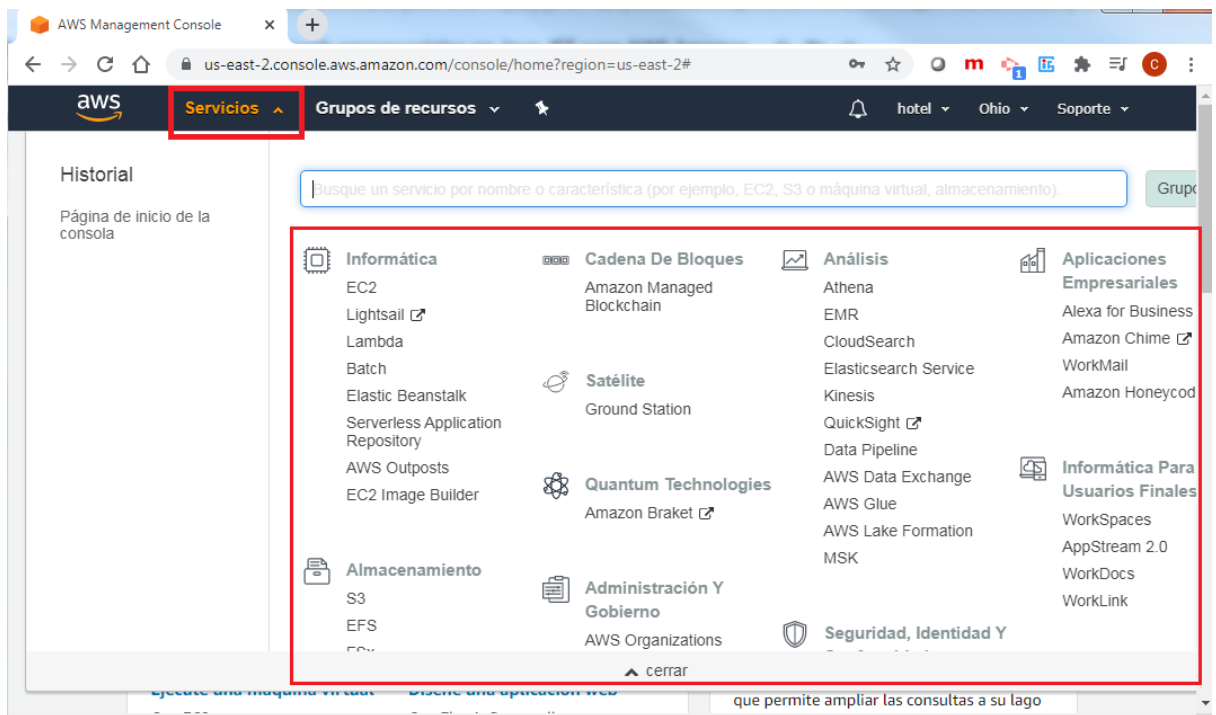
La asignación de la tarjeta de crédito es solo para comprobar que es una cuenta válida, Amazon brinda una capa gratuita que te permitirá acceder a la mayoría de los servicios de AWS por un año. La capa gratuita y el tiempo el suficiente para echar a andar una plataforma lo suficientemente profesional.

Luego de creado la cuenta de usuario de AWS Amazon, se debe tener acceso a la “Consola de administración de AWS”,





En el menú “Servicios” se observa los más de 170 servicios de AWS a los que se tiene acceso.

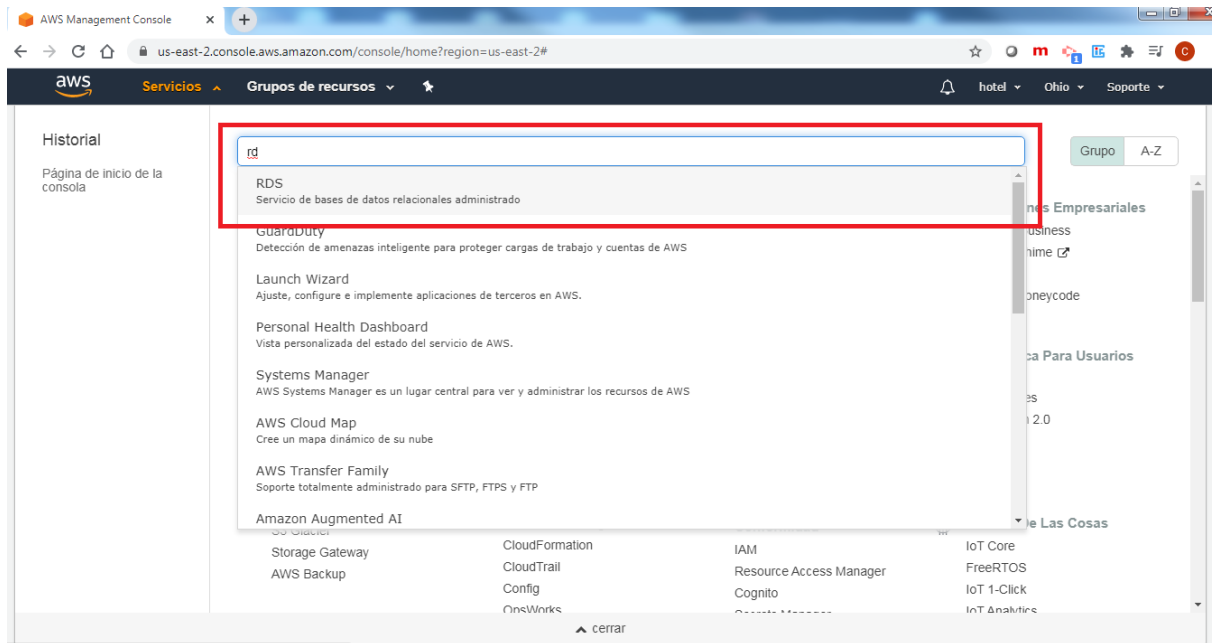


Los servicios de AWS contemplan entre otras Computación en la Nube (Cloud Computing), Almacenamiento, Bases de Datos, Servicios Multimedia, Seguridad, Analítica, Inteligencia Artificial, Internet de las Cosas (IoT), BlockChain, etc.

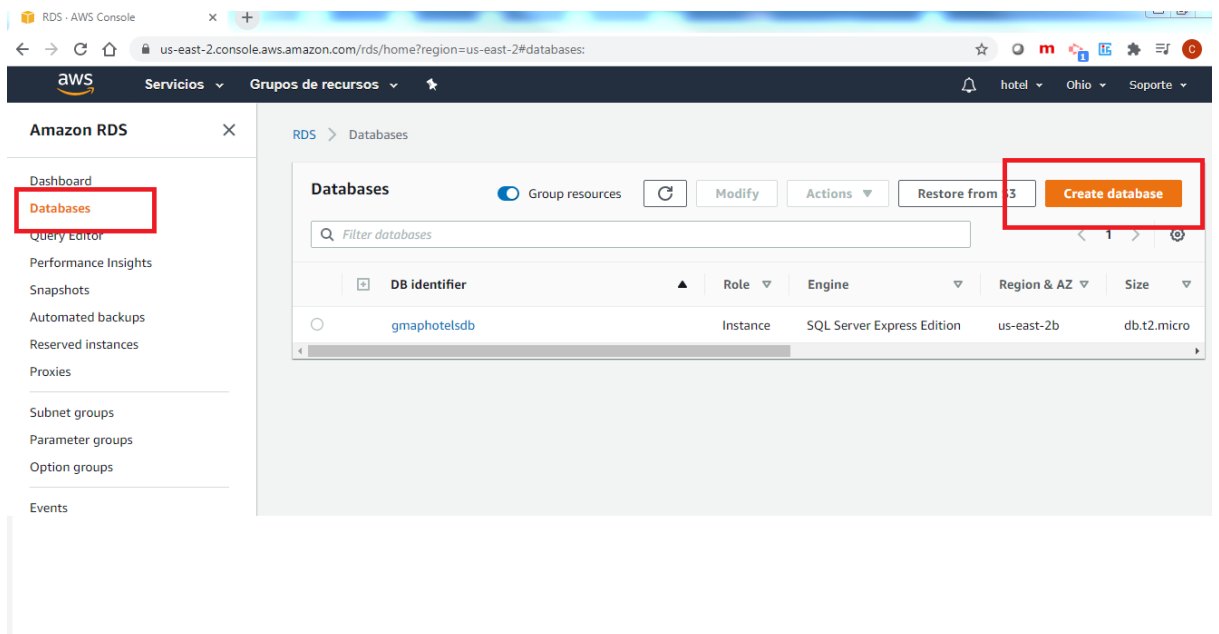
En este tutorial utilizaremos servicios asociados a Computación en la Nube (Cloud Computing) y Bases de Datos correspondientes a su capa gratuita.

8. Creación de Instancia de Base de Datos SQL Server en AWS Amazon.

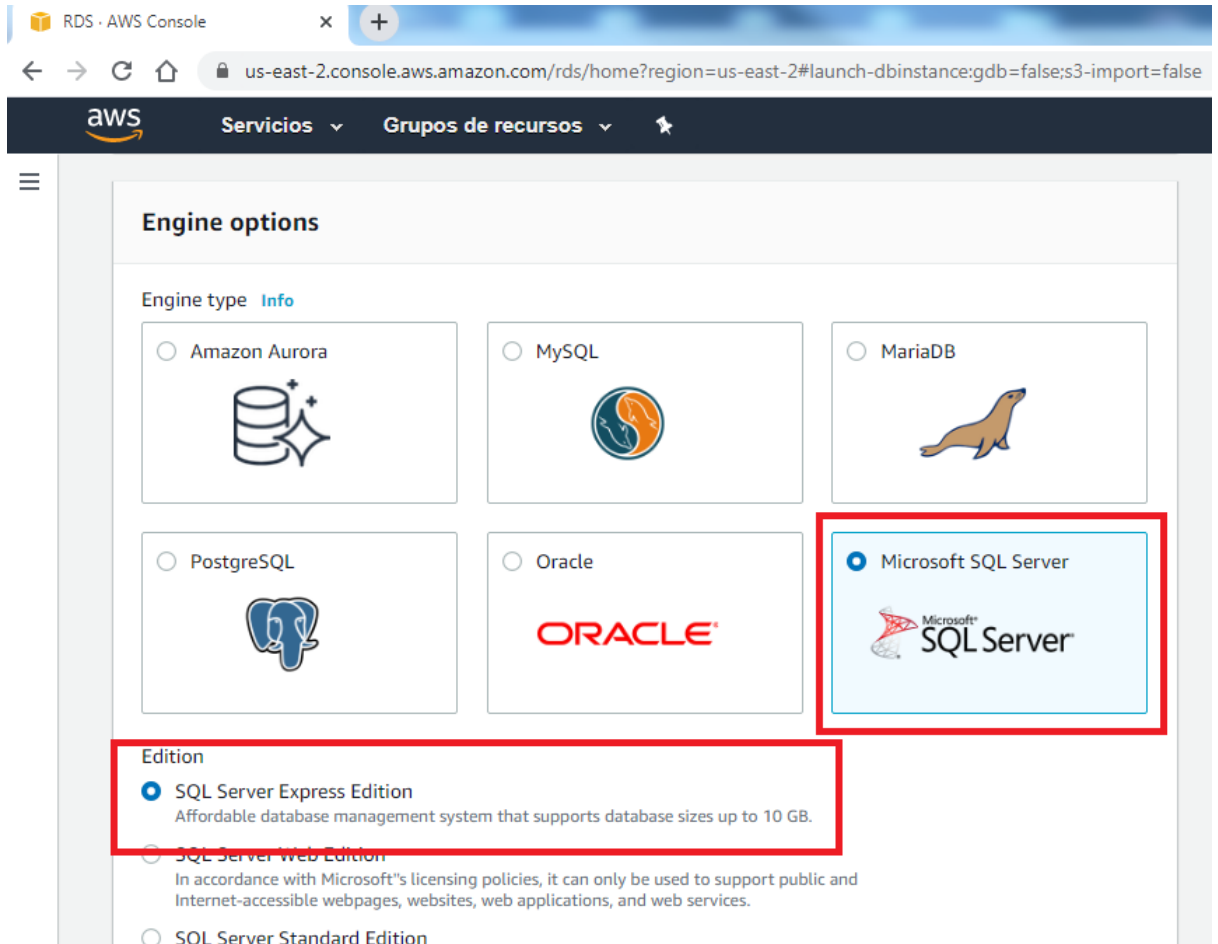
En el panel de Servicios de AWS busque y seleccione el servicio RDS



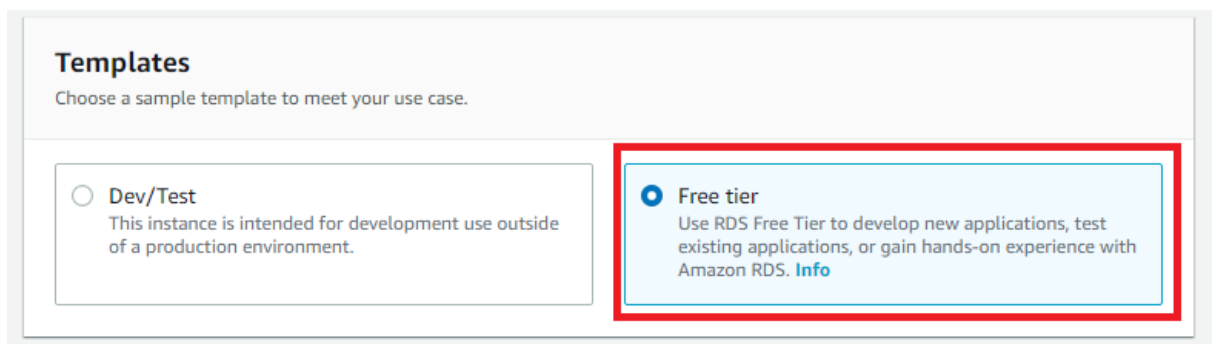
Sobre la consola de administración RDS de Amazon, seleccione el menú “Databases” y luego haga click en el botón “Create database”



En la pantalla “Create Database” seleccione el Engine Type “Microsoft SQL Server” y Edition “SQL Server Express Edition”



En la selección “Template” seleccione la opción “Free tier”



Asigna un password de acceso a la base de datos bastante sencillo, fácil de recordar.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign)

Confirm password [Info](#)

Es importante mantener todas las opciones siguientes por defecto,

► **Additional configuration**
Database options, backup enabled, backtrack disabled, Performance Insights enabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#) [↗](#)

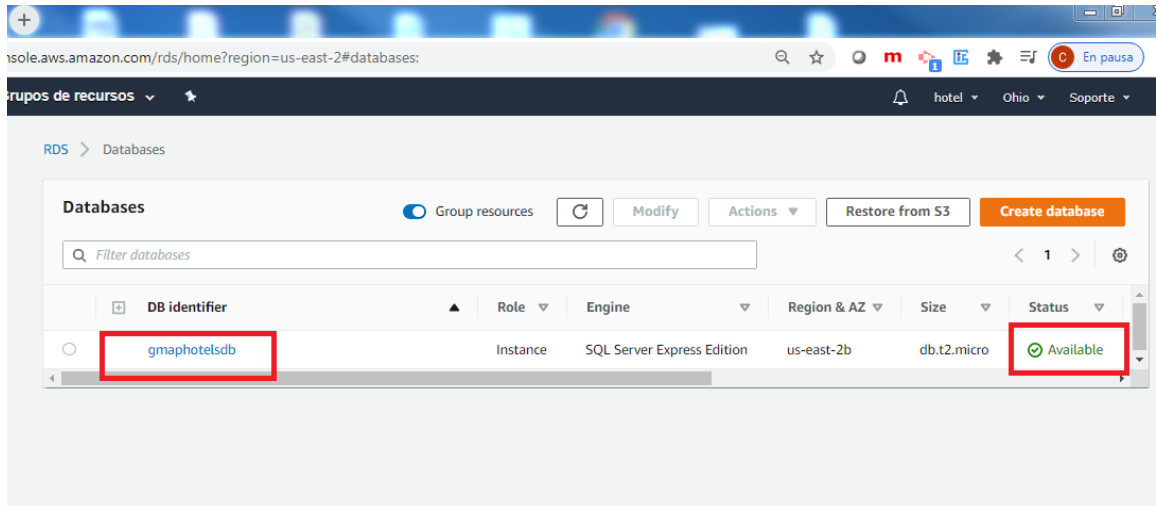
When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page.](#) [↗](#)

i You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

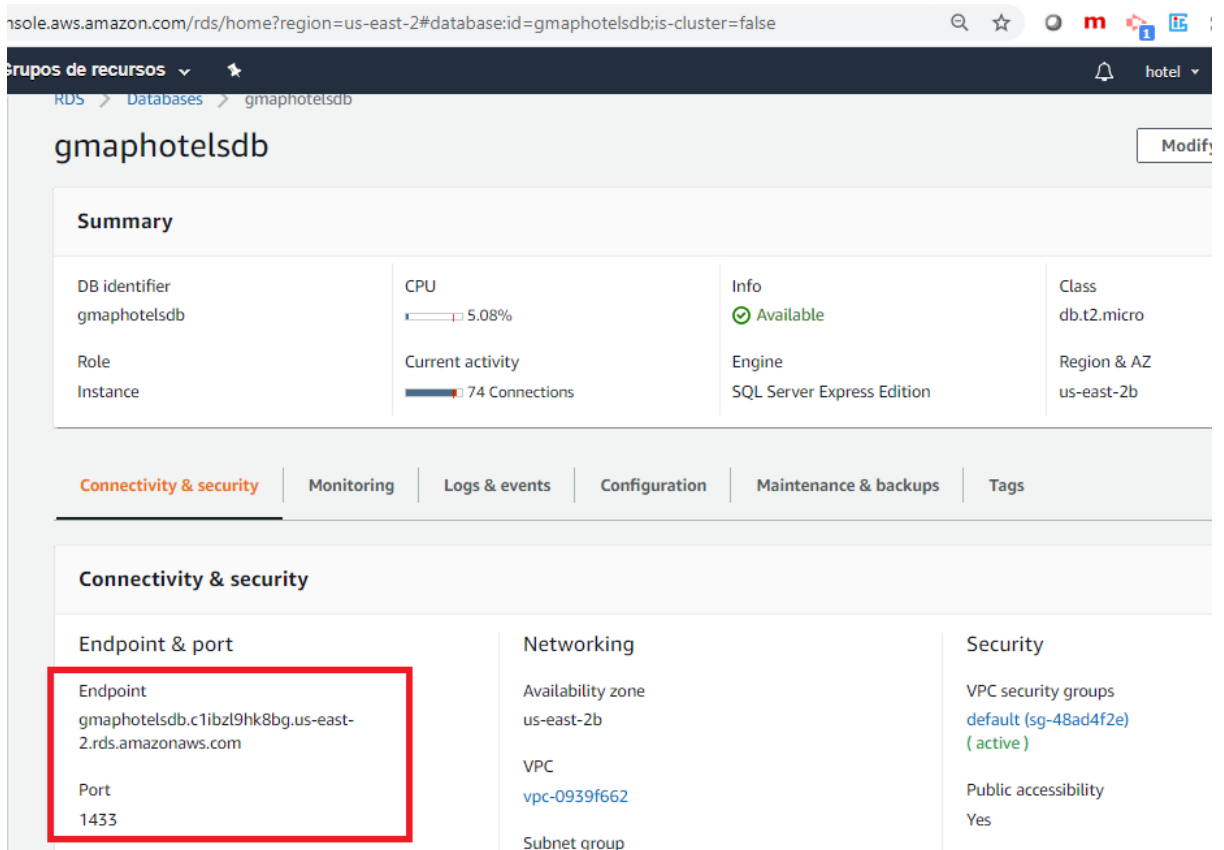
Cancel **Create database**

Complete el proceso haciendo click en el botón “Create database”

Luego de crear la base de datos en el RDS de AWS Amazon debes observar la instancia "Available"

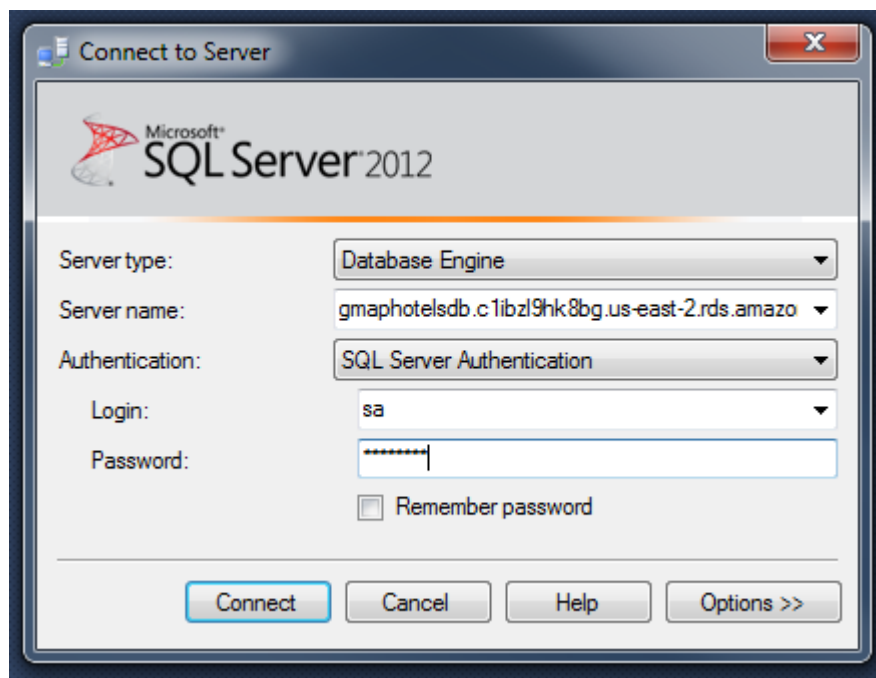


seleccione la instancia de base de datos para mostrar el detalle de la instancia de base de datos:



Es importante considerar el Endpoint y el Port para acceder a la base de datos desde el “SQL Server Management Studio”

es importante acceder a la base de datos desde el “SQL Server Management Studio” para poder crear las tablas, procedimientos almacenados, funciones, etc



The screenshot shows Microsoft SQL Server Management Studio with a query window open. The query is a SELECT statement with TOP 1000 rows, listing various client details. The results pane shows a table with columns: nro_documento, denominacion, telefono, correo, and dir. The status bar at the bottom indicates the query was executed successfully, returning 95 rows.

```

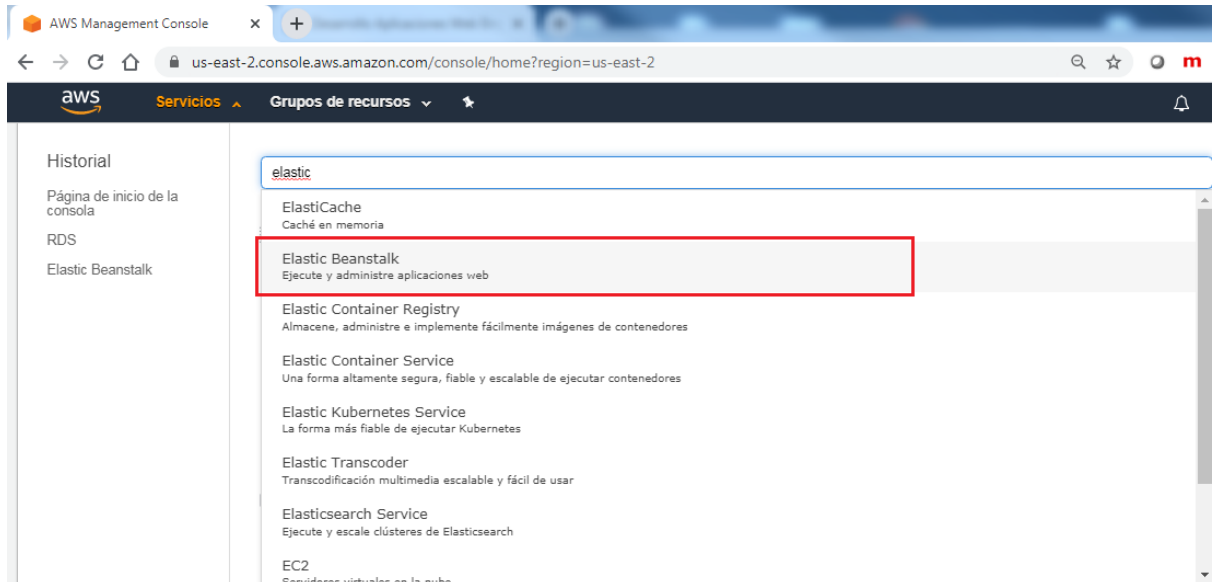
/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [id_cliente]
, [id_tipo_cliente]
, [id_tipo_documento]
, [nro_documento]
, [denominacion]
, [telefono]
, [correo]
, [direccion]
, [estado]
, [audit_usuarioid]
, [audit_fecha hora]
, [audit_terminal]
, [audit_estado]
, [id_empresa]
    
```

	nro_documento	denominacion	telefono	correo	dir
1	20333976235	BASADRE SUITE	44223076153939	NULL	1
2	10004612488	LINARES BRIN CARLOS	2217639-972731816	NULL	
3	20431549159	G.C.SER.MED. DERMATOLOGICO	2644394	NULL	
4	20333684021	CONQUISTADORES SWISS MANAGEMENT	4217790-4215326	NULL	S
5	20292512661	PERU STAR INSERINSA	4427376	NULL	
6	20386858561	EMBAJADA CHINA	4429458-ANEXO 110	NULL	9
7	20178344952	COELVISAC S.A.	4405660-998351956	NULL	R
8	20341137935	SILVER STANDAR PERU	7009000	NULL	
9	20517052111	CASA DELIA D & D C A C	4217354	NULL	

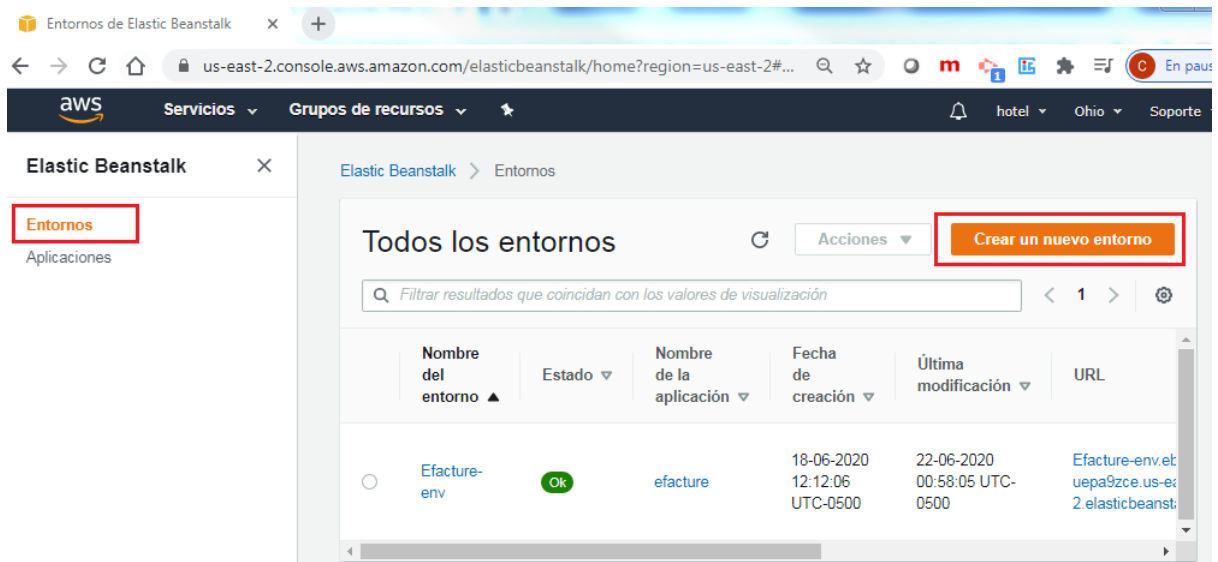
Query executed succe... | gmaphotelsdb.c1ibzI9hk8bg.u... | sa (125) | master | 00:00:00 | 95 rows

9. Creación de Instancia de Java Web Server Tomcat en AWS Amazon.

En la consola de administración seleccione el servicio “Elastic Beanstalk”



En la sección “Entornos” haga clic en el botón “Crear un nuevo entorno”



Seleccione la opción “Entorno del servidor web”, haga clic en el botón “Seleccionar”,

Elastic Beanstalk > Crear un entorno

Seleccionar una capa de entorno

AWS Elastic Beanstalk tiene dos tipos de capas de entorno que admiten distintos tipos de aplicaciones web. Los servidores web son aplicaciones estándar que escuchan y luego procesan solicitudes HTTP, por lo general, en el puerto 80. Los procesos de trabajo son aplicaciones especializadas que tienen una tarea de procesamiento de fondo que escucha los mensajes de una cola de Amazon SQS. Las aplicaciones de trabajo publican esos mensajes en su aplicación por medio de HTTP.

- Entorno del servidor web**
Ejecute un sitio web, una aplicación web o una API web que atienda las solicitudes HTTP.
[Más información](#)
- Entorno de trabajo**
Ejecute una aplicación de trabajo que procese cargas de larga duración a petición o que realice tareas conforme a una programación.
[Más información](#)

Cancelar **Seleccionar**

Coloque el nombre y la descripción de la aplicación

Elastic Beanstalk > Crear un entorno

Crear un entorno de servidor web

Publique un entorno con una aplicación de ejemplo o su propio código. Al crear un entorno, permite que AWS Elastic Beanstalk administre los recursos de AWS y los permisos en su nombre. [Más información](#)

Información de la aplicación

Nombre de la aplicación

Hasta 100 caracteres Unicode, sin incluir la barra (/).

► **Etiquetas de la aplicación (opcional)**

Información del entorno

Elija el nombre, el subcampo y la descripción para su entorno. Estas no se pueden cambiar más adelante.

Nombre del entorno

Para poder crear una instancia de Java con la plataforma Tomcat selecciones las opciones indicadas en la siguiente figura, es importante indicar plataforma “Tomcat With Java” para luego hacer click en el botón “Crear Entorno”.

Plataforma

Plataforma administrada
Plataformas publicadas y mantenidas por AWS Elastic Beanstalk. [Obtenga más información](#)

Plataforma personalizada
Plataformas creadas por usted y que le pertenecen.

Plataforma
Tomcat

Ramificación de la plataforma
Tomcat 8.5 with Java 8 running on 64bit Amazon Linux

Versión de la plataforma
3.3.7 (Recommended)

Código de la aplicación

Aplicación de muestra
Comenzar de inmediato con un código de muestra.

Versión existente
Versiones de la aplicación que ha cargado para **Mi Aplicación**.

Se creará el entorno de la nueva aplicación en Java Tomcat .

Elastic Beanstalk > Entornos > MiAplicacion-env

Creando MiAplicacion-env
Este proceso tardará unos minutos. ...

```
2:17am Using elasticbeanstalk-us-east-2-821312097137 as Amazon S3 storage bucket for environment data.  
2:17am createEnvironment is starting.
```



Cuando completa la creación del entorno de la aplicación se debe mostrar el siguiente entorno:

Elastic Beanstalk > Entornos > MiAplicacion-env

MiAplicacion-env Actualizar Acciones ▾

[MiAplicacion-env.eba-pubrg7as.us-east-2.elasticbeanstalk.com](https://miaplicacion-env.eba-pubrg7as.us-east-2.elasticbeanstalk.com) (e-
eadihcmby)

Nombre de la aplicación: **Mi Aplicacion**

Estado	Versión en ejecución	Plataforma
 Aceptar Causas	Sample Application Cargar e implementar	 Tomcat 8.5 with Java 8 running on 64bit Amazon Linux/3.3.7 Cambiar

Eventos recientes Mostrar todo

< 1 >

Hora	Tipo	Detalles
------	------	----------

al realizar click en el enlace se muestra la página web de prueba.

No es seguro | miaplicacion-env.eba-pubrg7as.us-east-2.elasticbeanstalk.com

En pausa

Congratulations

Your first AWS Elastic Beanstalk Application is now running on your own dedicated environment in the AWS Cloud

What's Next?

- [Learn how to build, deploy and manage your own applications using AWS Elastic Beanstalk](#)
- [AWS Elastic Beanstalk concepts](#)
- [Learn how to create new application versions](#)
- [Learn how to manage your application environments](#)

Download the AWS Reference Application

- [Explore a fully-featured reference application using the AWS SDK for Java](#)

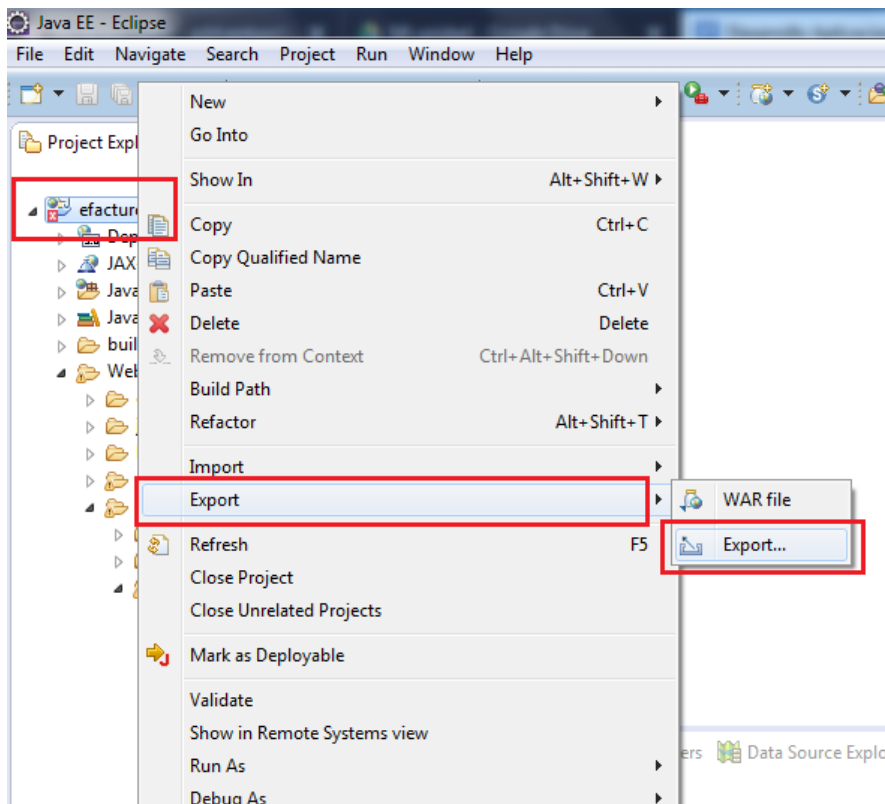
AWS Toolkit for Eclipse

- [Developers may build and deploy AWS Elastic Beanstalk applications directly from Eclipse](#)
- [Get started with Eclipse and AWS Elastic Beanstalk by watching this video](#)
- [View all AWS Elastic Beanstalk documentation](#)

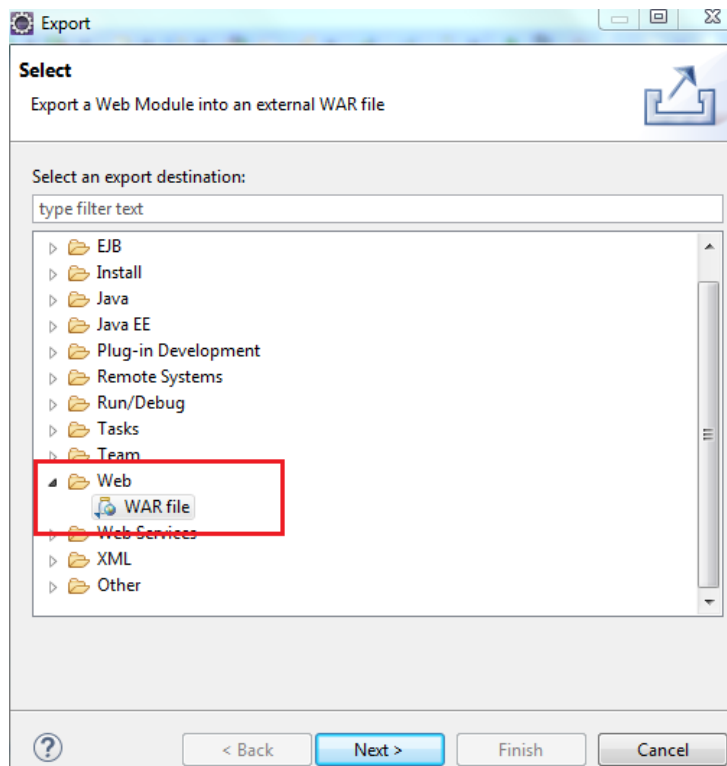
10. Despliegue del proyecto SEEFACTURE en la instancia AWS Amazon.

Para hacer el despliegue del aplicativo web se debe generar el empaquetado WAR (Web Application Archive), es un archivo utilizado para distribuir una colección de (JSP) JavaServer Pages, servlets, clases Java, archivos XML, bibliotecas de tags y páginas web estáticas (HTML y archivos relacionados) que juntos constituyen una aplicación web y son gestionados por los servidores web.

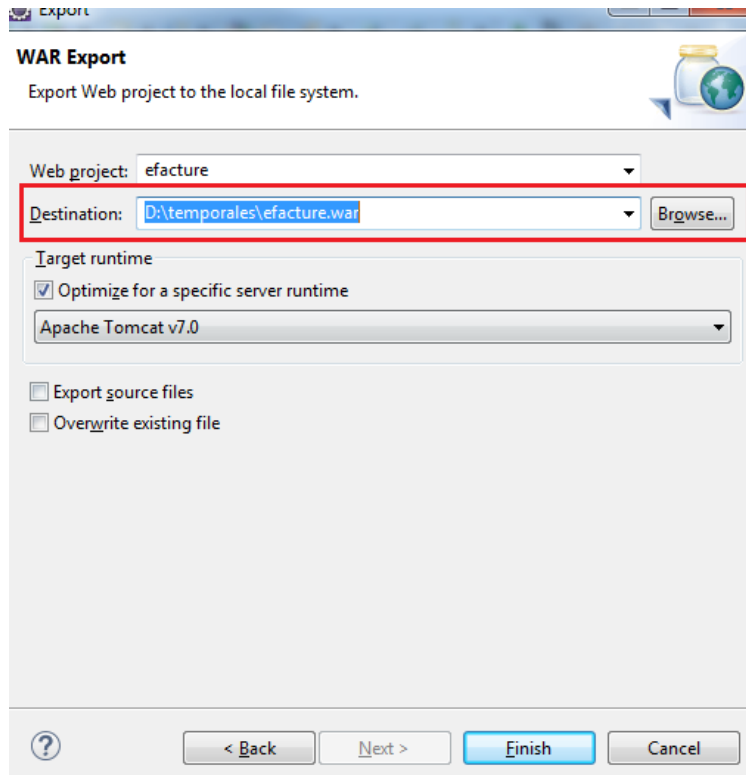
Realice click derecho en el proyecto “efacture”, seleccione el menú “Export”, submenú “Export...”.



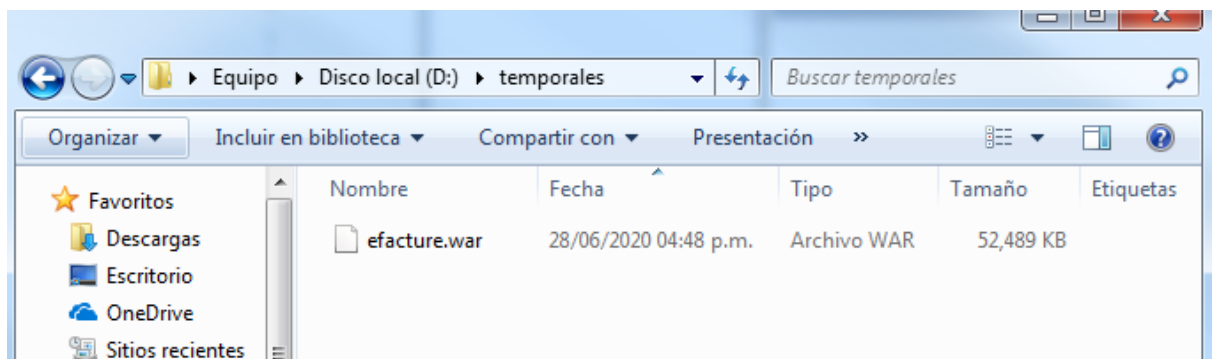
En la pantalla “Select” seleccione “WAR file” de la sección “Web”, luego hacer click en el botón “Next”



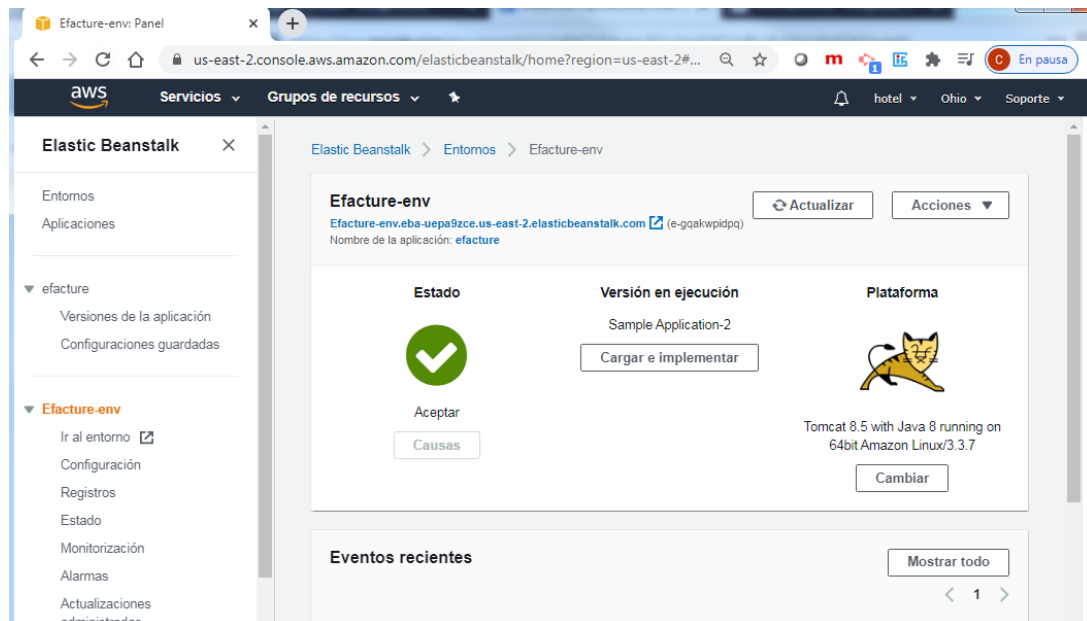
En la pantalla “War Export”, seleccione la carpeta donde se generará el archivo WAR, además de indicar el nombre del archivo, en este ejemplo efacture.war, para luego hacer click en el botón “Finish”.



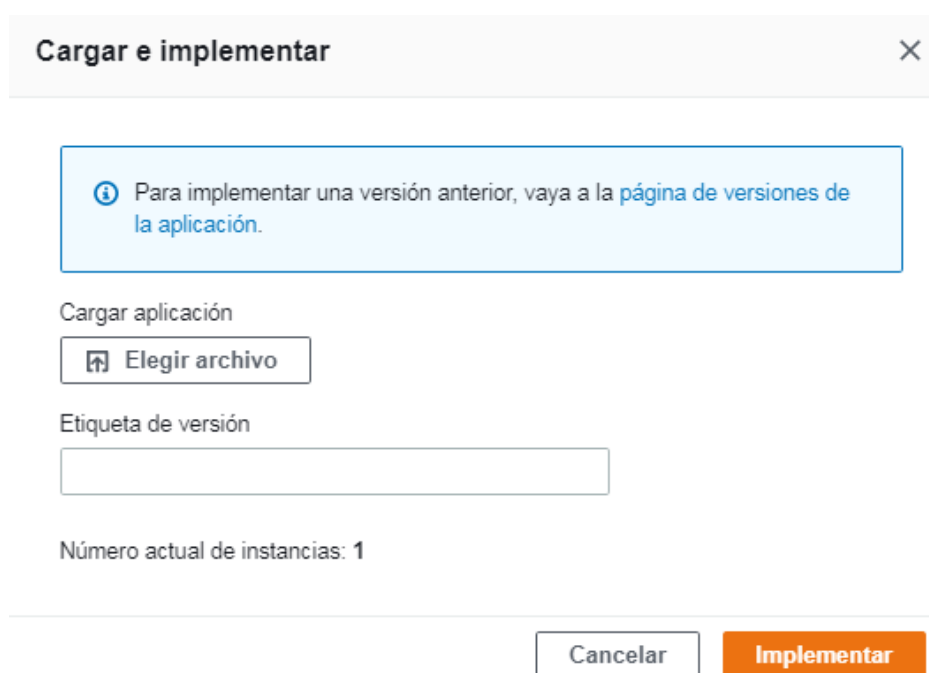
En la carpeta indicada se debe generar el archivo WAR.



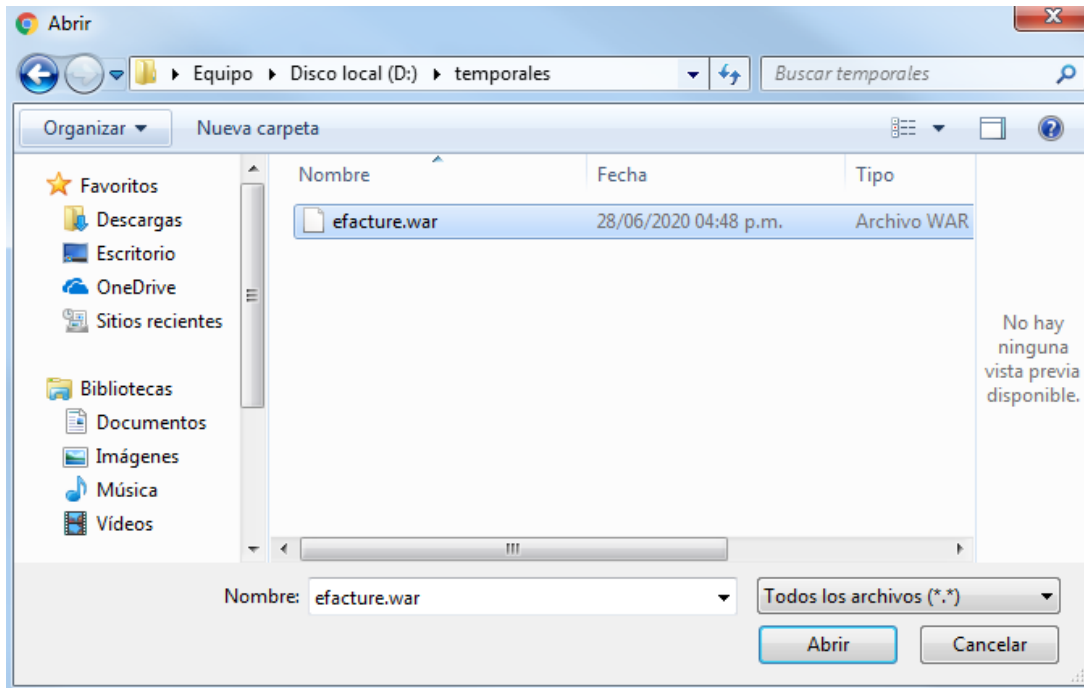
Ingrese a la cuenta amazon, en la consola de administración de “Elastic Beanstalk” seleccione el entorno de su aplicación, en este ejemplo “Efacture-Env”.



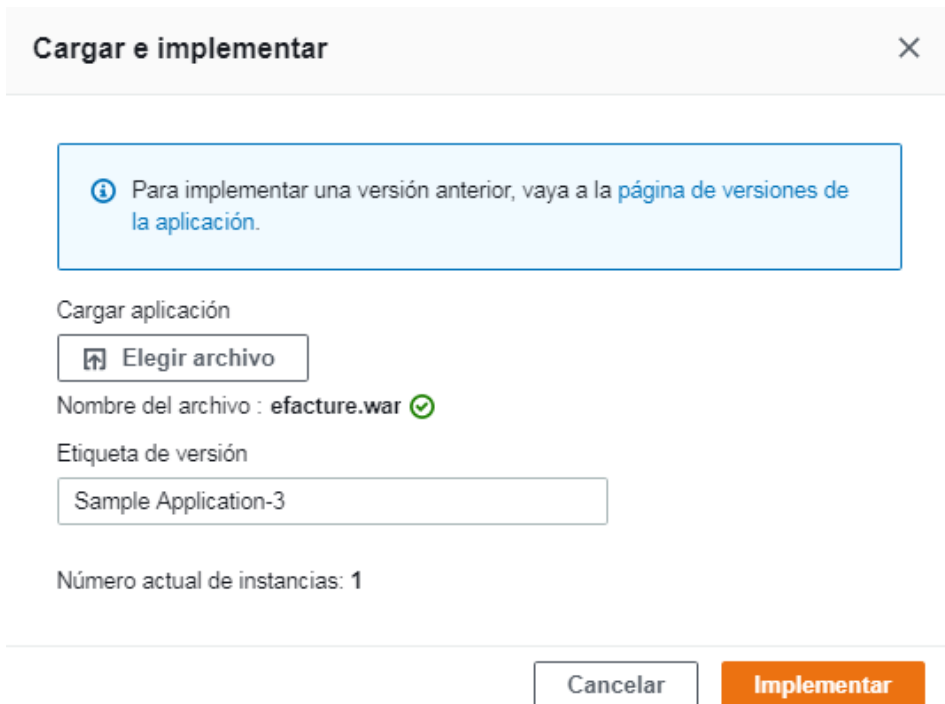
dentro del entorno de la aplicación realice click en el botón “Cargar e Implementar”



Dentro del formulario de “Cargar e Implementar” realice click en el botón “Elegir archivo”, se mostrará el formulario de búsqueda de archivos mediante el cual debe ubicar y seleccionar el archivo WAR generado, para luego hacer click en el botón “Abrir”.



Una vez cargado realice click en el botón “Implementar”, esta acción enviará el WAR al contenedor del servidor web Tomcat en Amazon.



al completar el proceso de implementación, realice click en el enlace del entorno.



Elastic Beanstalk > Entornos > MiAplicacion-env

MiAplicacion-env

[MiAplicacion-env.eba-pubrg7as.us-east-2.elasticbeanstalk.com](https://miaplicacion-env.eba-pubrg7as.us-east-2.elasticbeanstalk.com) (e-zadhschmby)

Nombre de la aplicación: **Mi Aplicacion**

[Actualizar](#) [Acciones ▼](#)

Estado	Versión en ejecución	Plataforma
 Aceptar Causas	Sample Application Cargar e implementar	 Tomcat 8.5 with Java 8 running on 64bit Amazon Linux/3.3.7 Cambiar

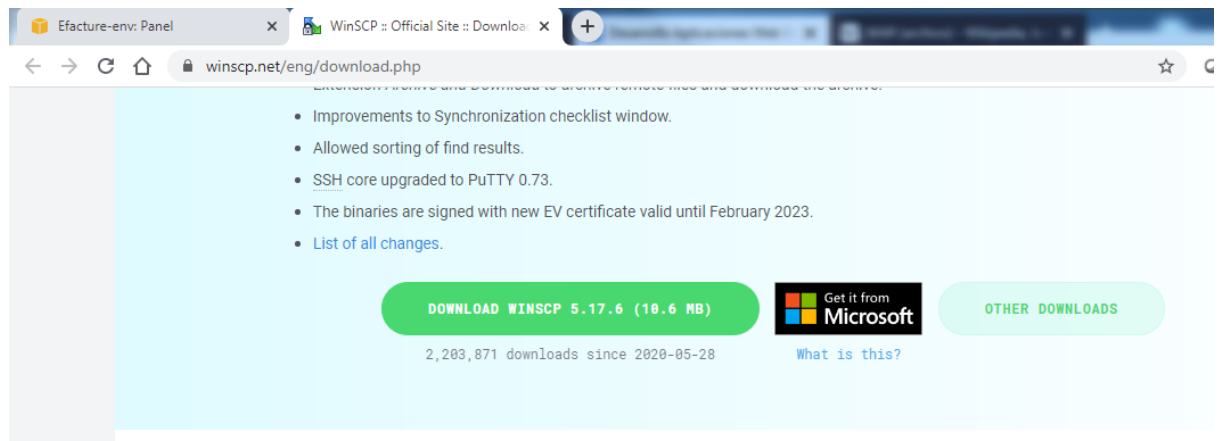
Se debe visualizar el login del aplicativo efacture.



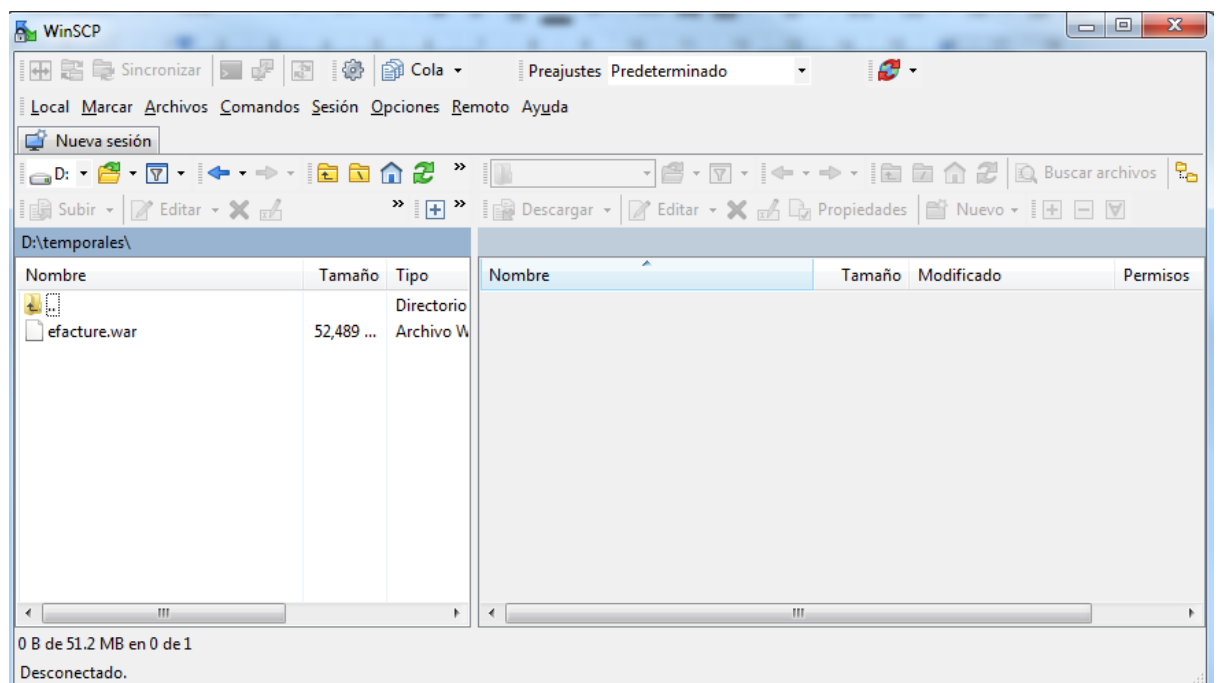
11. Cliente FTP WinSCP para la Instancia Web Server Tomcat en AWS Amazon.

Descargar el cliente FTP WinSCP del siguiente link:

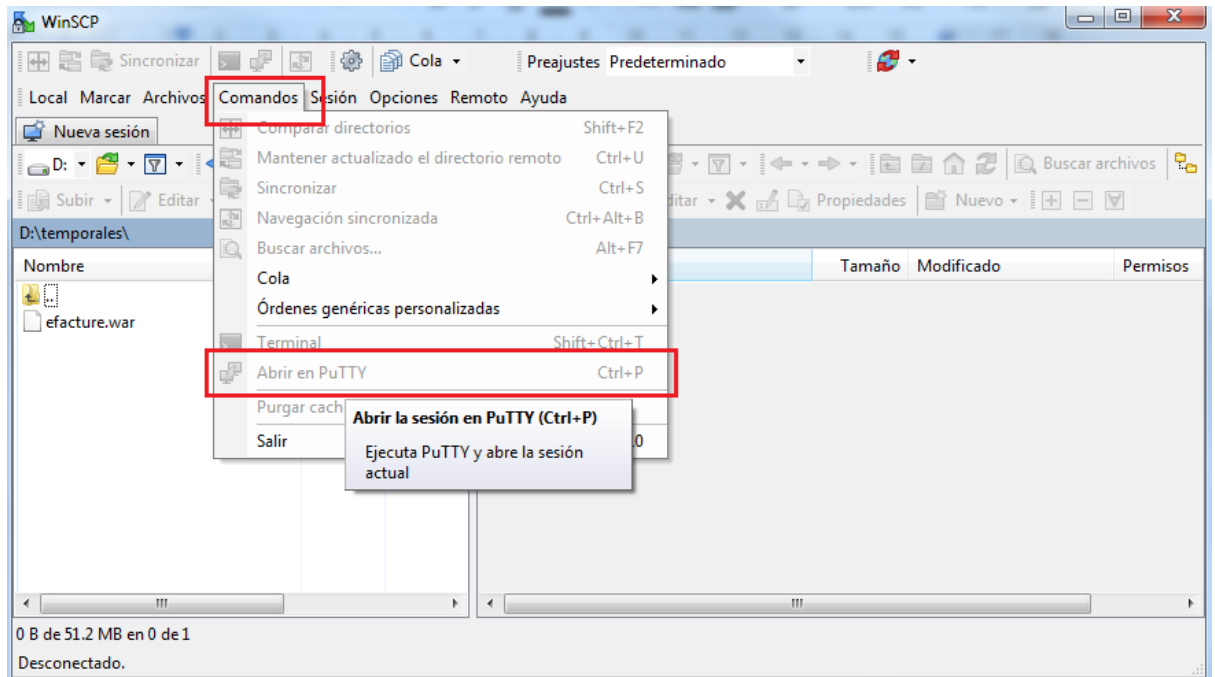
<https://winscp.net/eng/download.php>



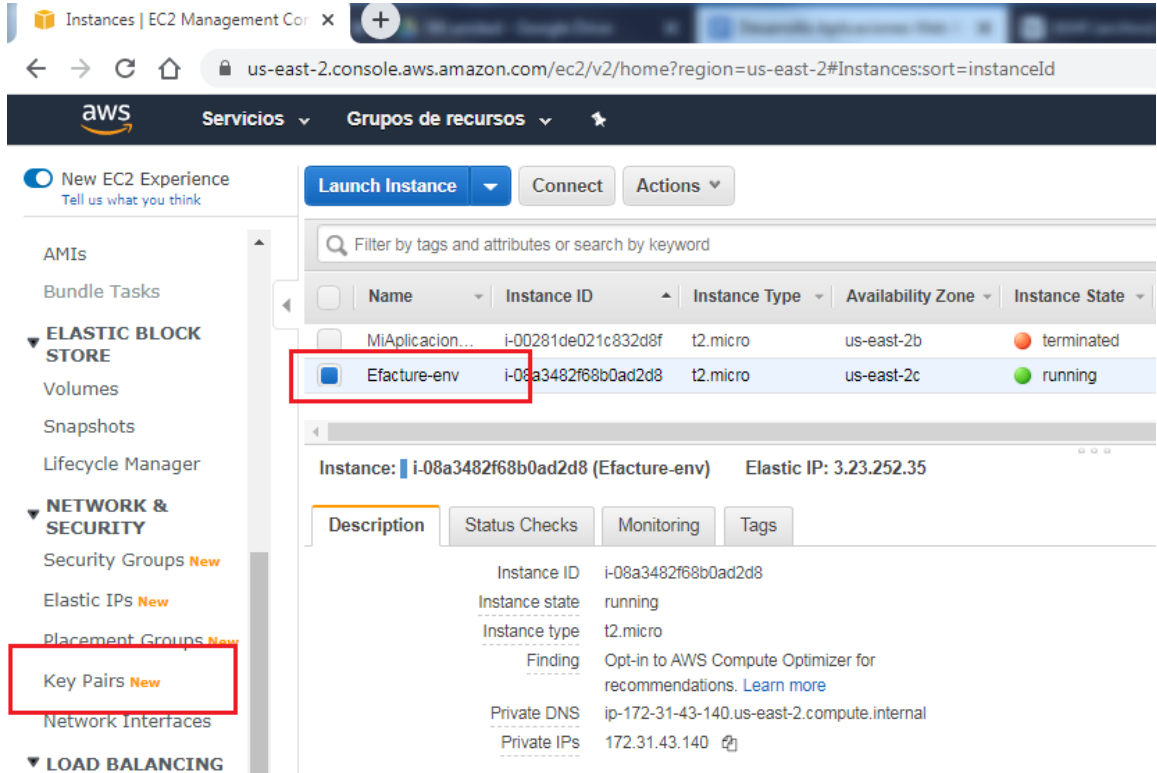
Realice la instalación del WinSCP,



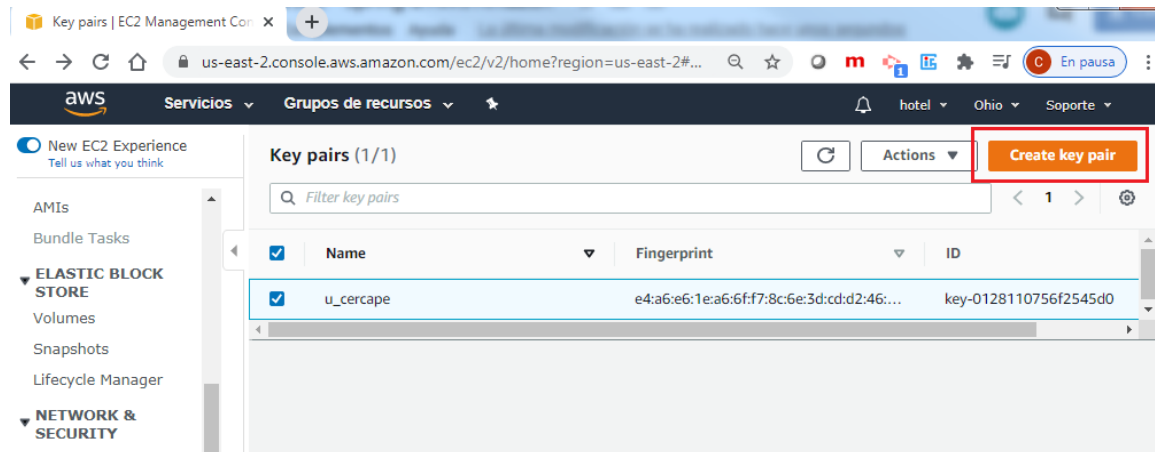
es importante verificar que se tenga el cliente Putty como parte del WinSCP.



En la consola de administración EC2 de Amazon, seleccione la instancia de su proyecto, en este ejemplo "Efacture-Env", para luego seleccionar la opción "Key Pairs" de la sección "NETWORK & SECURITY"

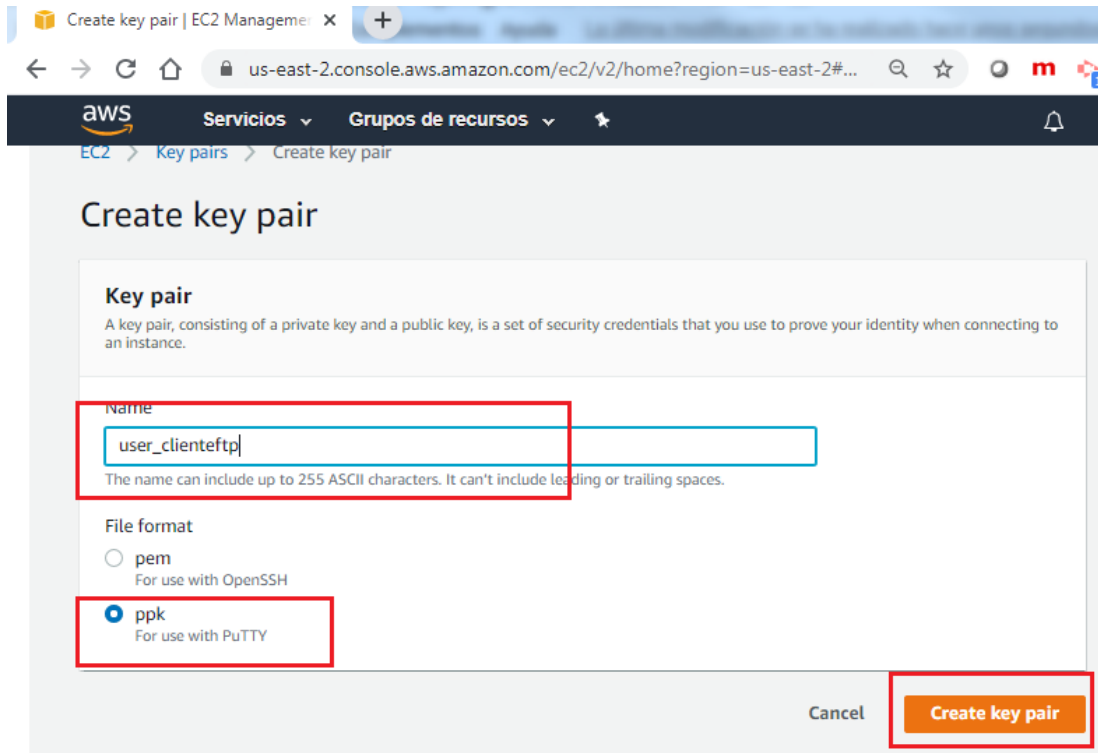


Se mostrará la consola de administración “Key Pairs” de AWS Amzon, esta consola permite la gestión de claves de acceso por aplicaciones externas contemplando esquemas de seguridad de cifrado asimétrico Clave Pública y Clave privada (par de claves).

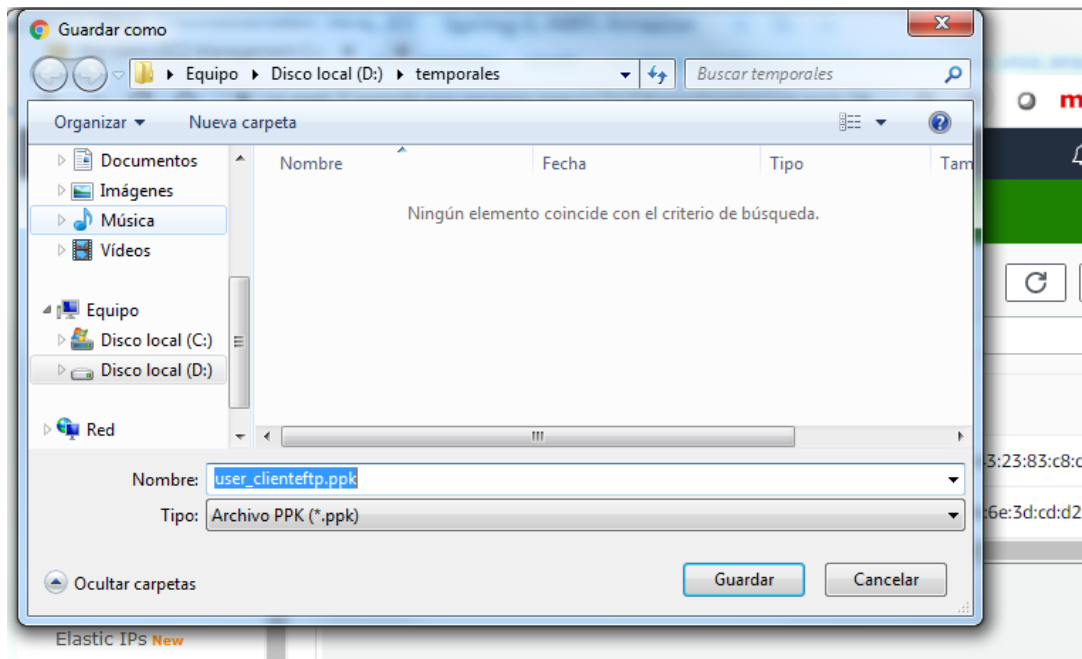


Realizamos click en el botón “Create key pair”, con el propósito de crear una credencial de acceso basado en cifrado asimétrico.

En la pantalla “Create Key Pair” indique el nombre del archivo de par de claves, y seleccione la opción “PPK”, este formato es muy importante debido a que los clientes FTP y SFTP utilizan archivos .ppk para almacenar claves cifradas. realice click en el botón “create key pair”.



la consola de administración de Key Pair descargará el archivo .ppk creado.



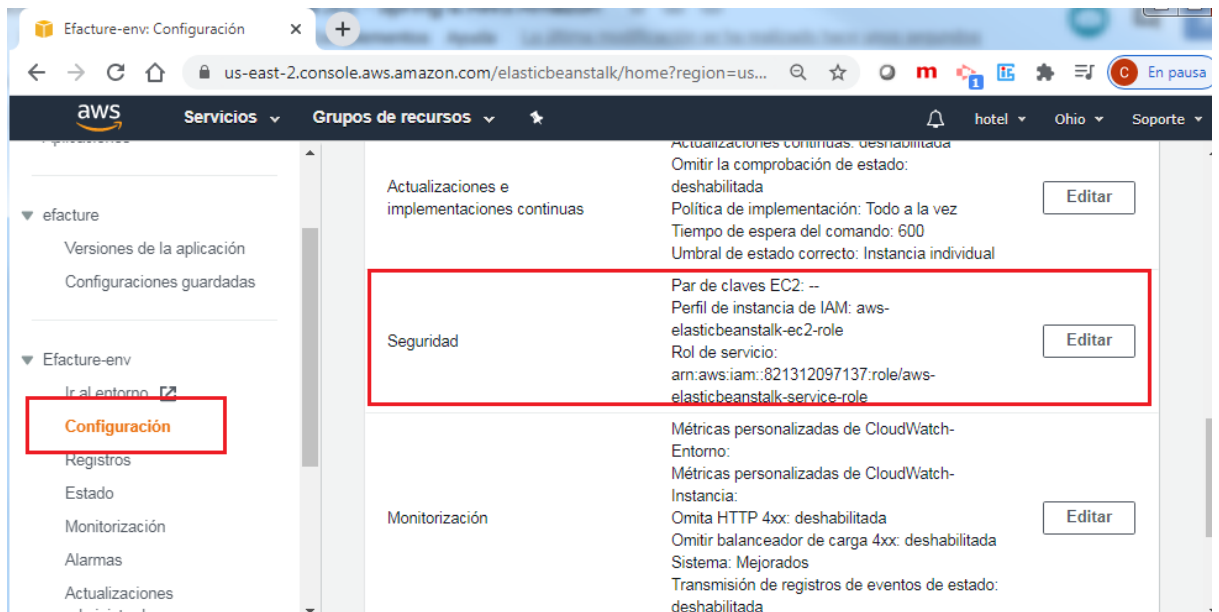
Luego de crear el Key Pair debemos asociar este al entorno de nuestro proyecto en Elastic Beanstalk.

Ingrese a la consola de entornos de su proyecto y seleccione la opción de

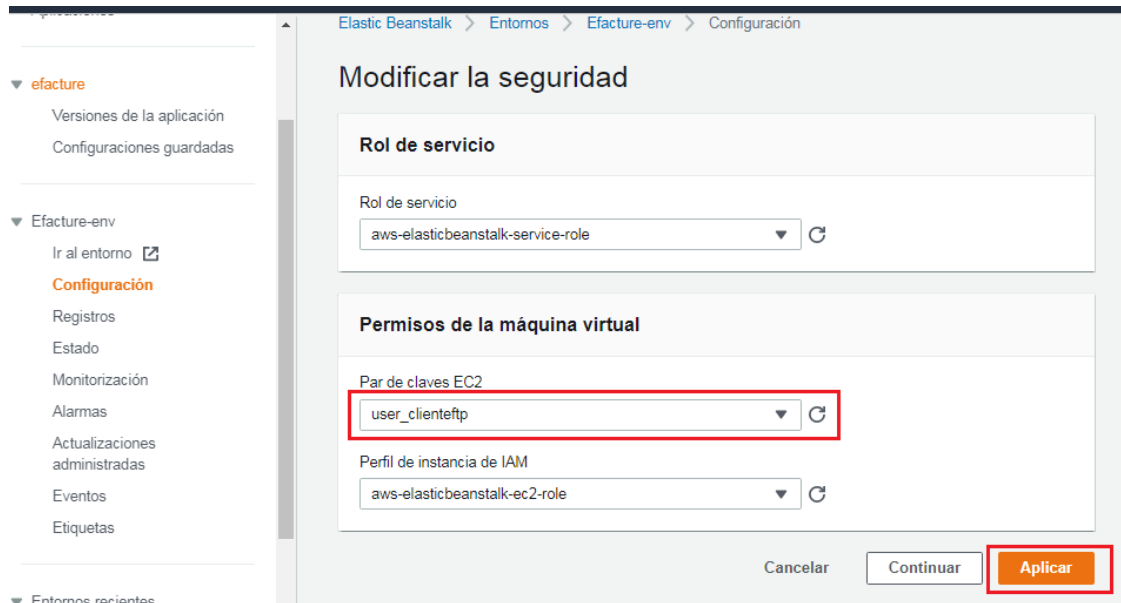
“Configuración”.



Se mostrará la consola de configuración del entorno de su proyecto, en esta consola de configuración ubique la sección “Seguridad”, para luego hacer click en el botón “Editar”.

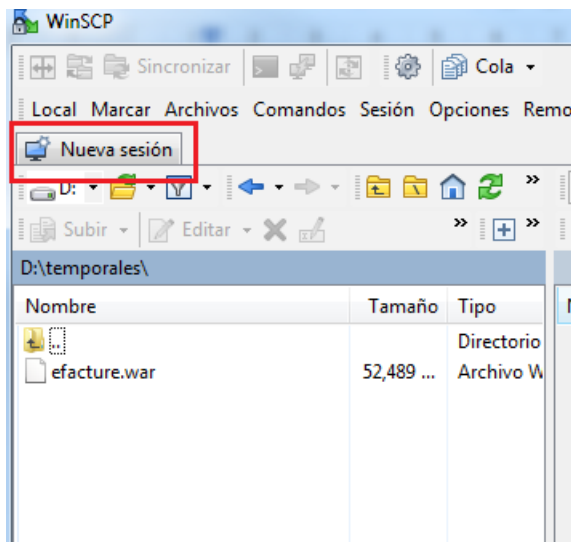


En el panel de seguridad seleccione el Key Pair creado, mantener las demás opciones por defecto y luego hacer click en el botón “Aplicar”.

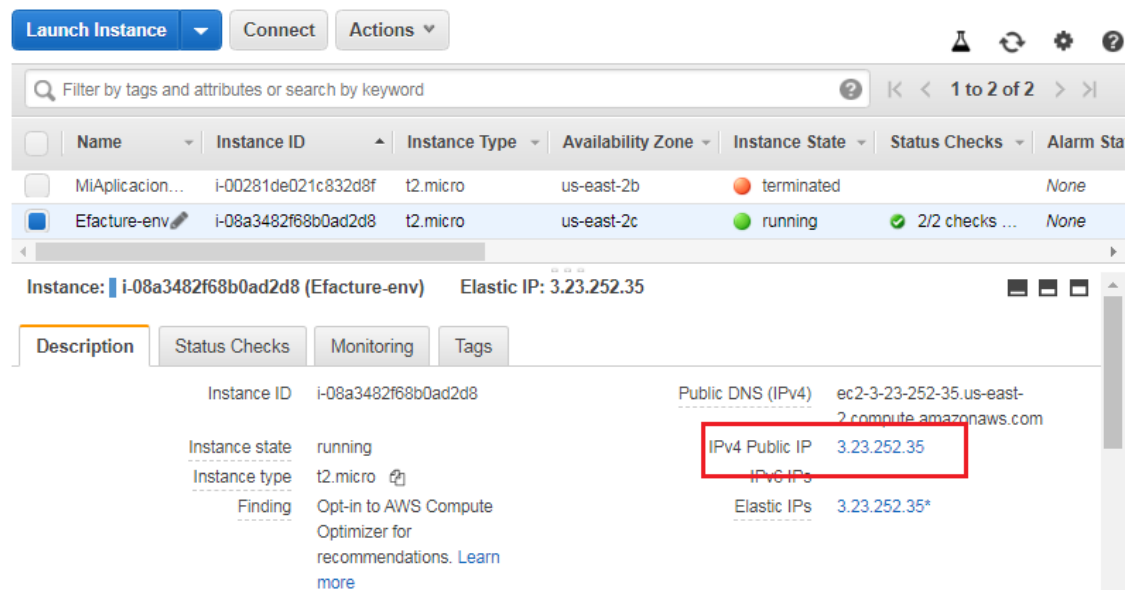
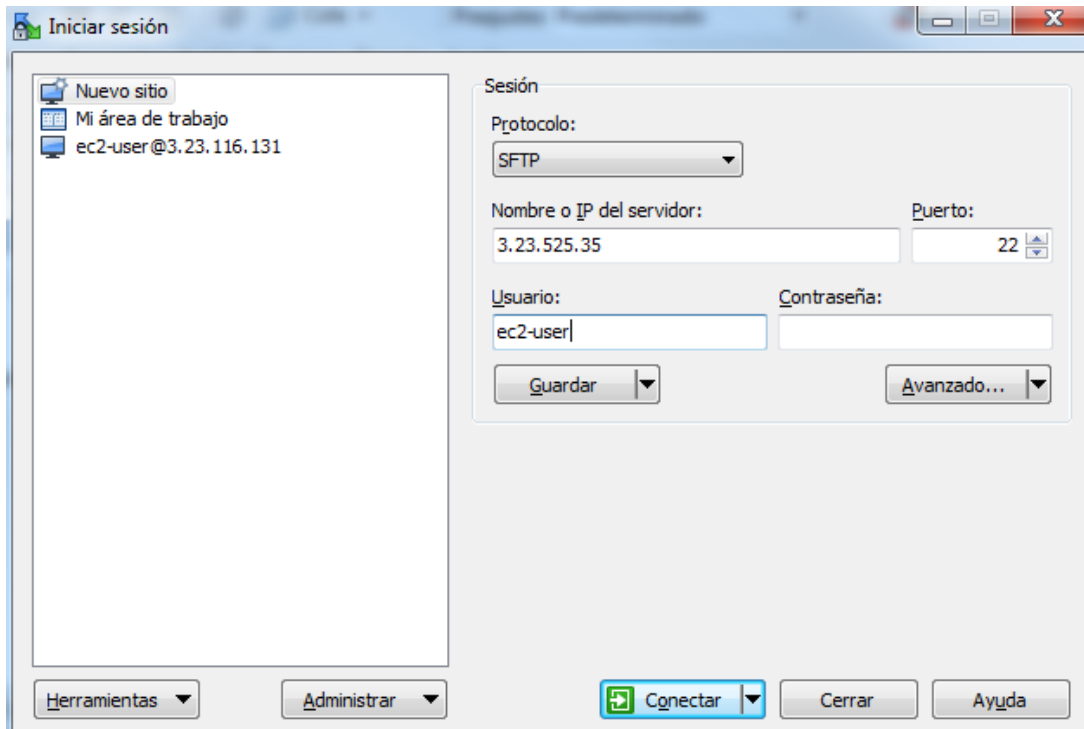


Con ello asociamos el Key Pair y por consecuencia el archivo cifrado PPK al entorno de nuestro proyecto en Elastic Beanstalk.

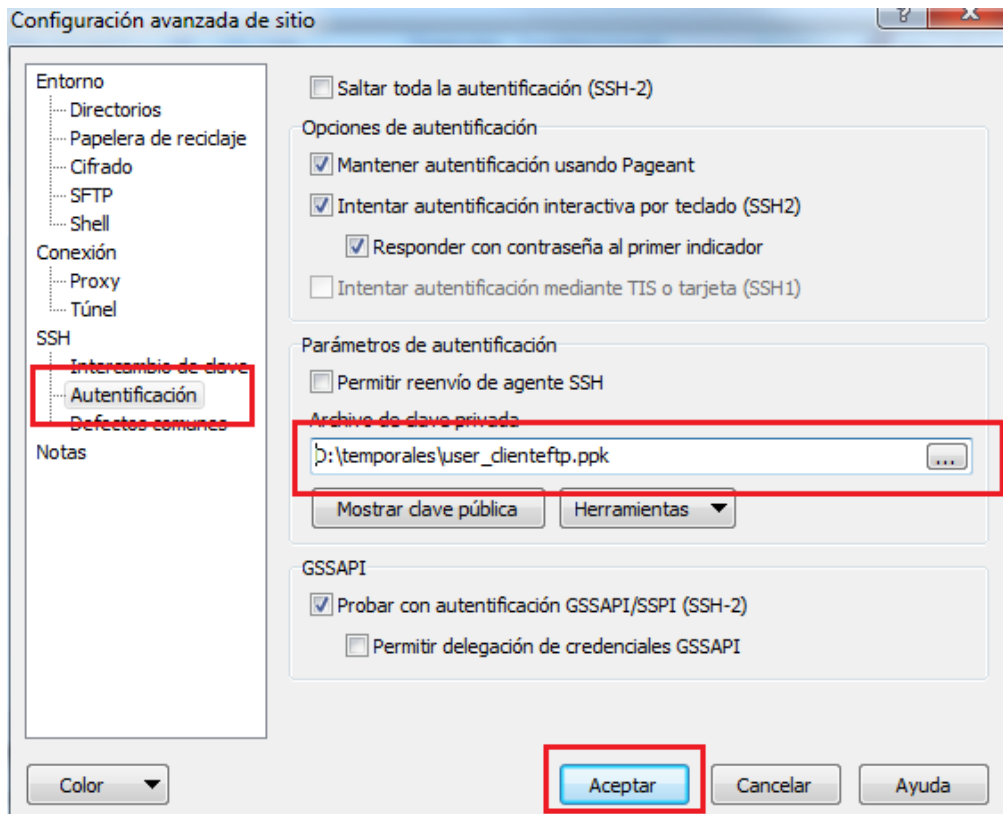
Abrir el cliente SFTP WinSCP, haga click en el botón “Nueva Sesión”, se mostrará la pantalla de Conexión de Sesión.



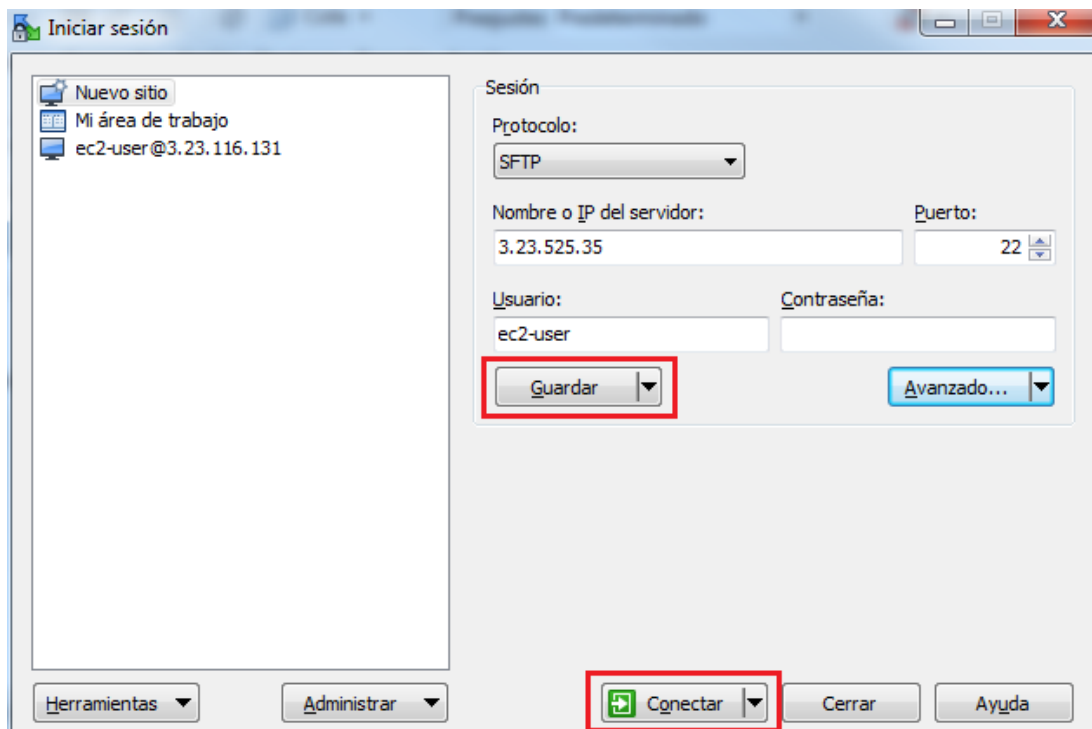
En la pantalla “Iniciar Sesión” seleccione la opción de “Nuevo Sitio”, especifique el protocolo como SFTP, el número de IP de su instancia en EC2, puerto 22, usuario ec2-user y deje el campo Contraseña en blanco como se observa en la imagen.



realice click en el botón “Avanzado” de la pantalla “Iniciar Sesión”. en la pantalla de “configuración avanzada de sitio”, seleccione la sección “Autenticación”, en esta sección debe hacer referencia al archivo de clave privada “PPK”, debe dejar todas las demas opciones por defecto. realice click en el botón “Aceptar”



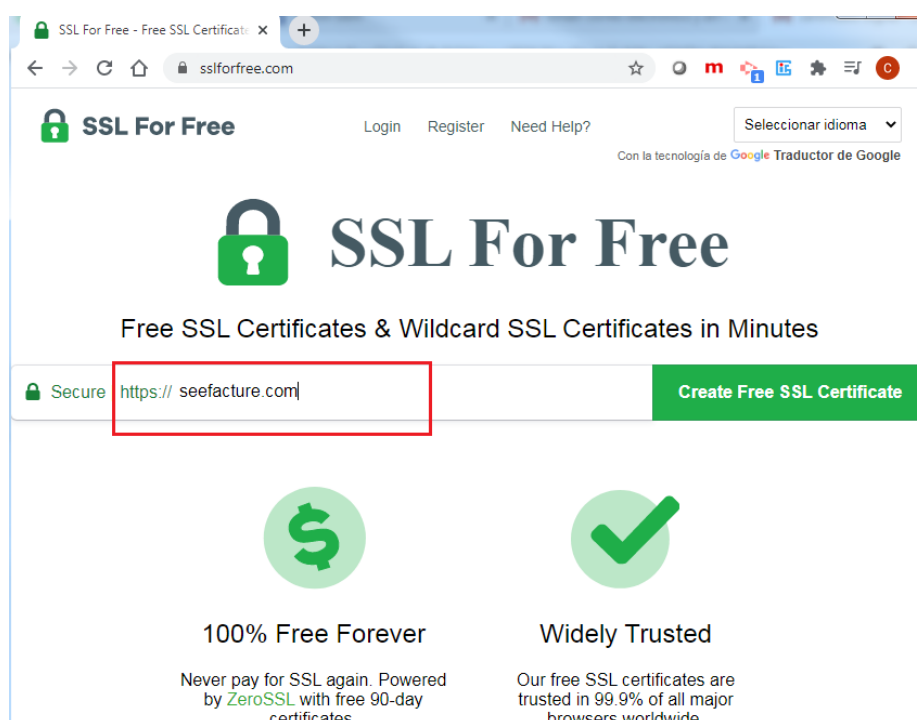
realice click en el botón “Guardar” y luego click en el botón “Conectar”



12. Instalación del certificado SSL en AWS Amazon y configuración para HTTPS.

Asignaremos al proyecto un certificado de seguridad SSL con el propósito de permitir el acceso a nuestra página web mediante https.

Ingresa a la página <https://www.sslforfree.com/> e indica el dominio de tu página web, para luego hacer click en el botón “Create Free SSL Certificate”,



Se te solicitará crear una cuenta de usuario, realizar click en el botón “Register”

Sign Up

Sign up for a free account to create and manage SSL certificates.

Domain: **seefacture.com**

Email Address
seefacture1@gmail.com

Password
.....

Register

I agree to receive important service updates.

Login



Al registrar el usuario indicar el dominio y hacer click en el botón “Next Step”

Domains

I need a wildcard certificate **PRO**

Please enter at least one domain to secure. For single-domain certificates the WWW-version of your domain will always be included at no extra charge.

Enter Domains

seefacture.com www.seefacture.com

+ Add Domain **PRO**

Next Step →

Indicar que la validez del certificado será de 90 días, la opción anual PRO posee un costo. realice click en el botón “Next Step”.

Validity

You can now choose between generating 90-day or one-year certificate validity. To keep manual work at a minimum, we recommend 1-year certificates.

90-Day Certificate

1-Year Certificate **PRO**

Next Step →

Indicar que se autogenera el archivo CSR. realice click en el botón “Next Step”.

CSR & Contact

Before validation, we will auto-generate contact information and a CSR for your certificate. To enter your information manually or paste an existing CSR, please uncheck the box below.

Auto-Generate CSR **?**

Next Step →

Mantenga seleccionado la opción Free, esta opción nos permitirá el uso del certificado por 3 meses, tiempo suficiente para proyectos académicos y de pruebas. realice click en el botón “Next Step”.

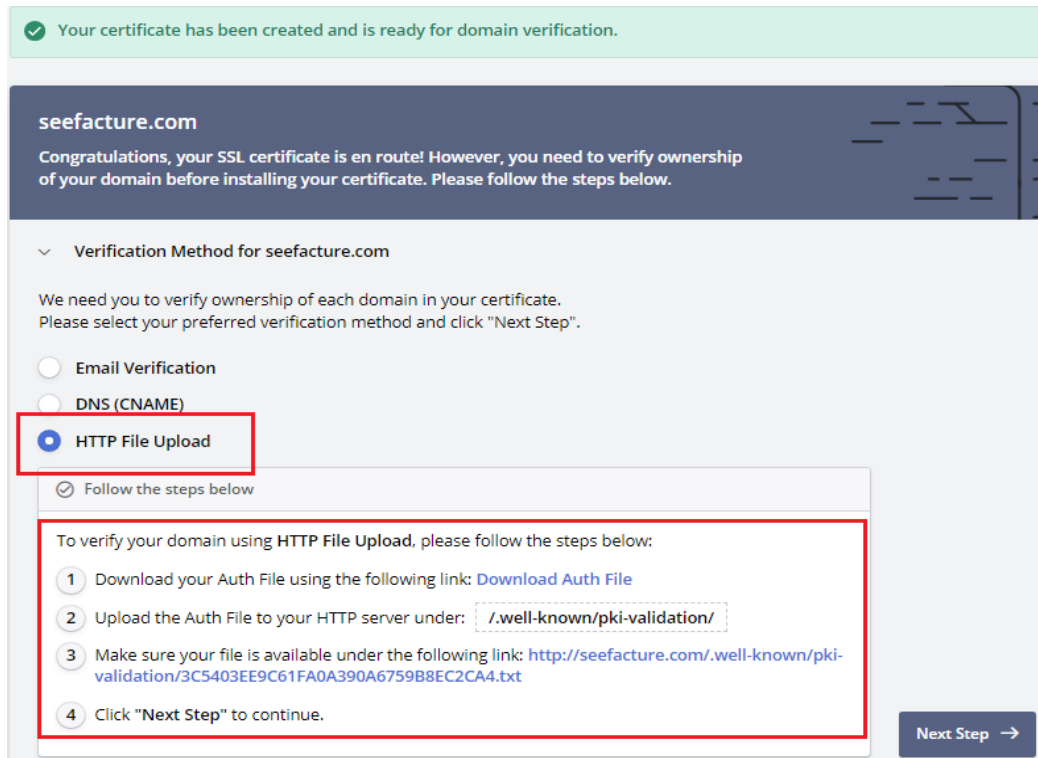
Finalize Your Order

Based on your selection of a 90-Day SSL Certificate you are fine staying on the Free Plan. To create and validate your SSL Certificate, please click "Next Step" below.

Free	Basic	Premium	Business
\$0 / month Free Forever	\$10 / month or \$8 if billed yearly	\$50 / month or \$40 if billed yearly	\$100 / month or \$80 if billed yearly
<input checked="" type="checkbox"/> Selected	<input type="checkbox"/> Select	<input type="checkbox"/> Select	<input type="checkbox"/> Select
<input checked="" type="checkbox"/> 3 90-Day Certificates	<input checked="" type="checkbox"/> ∞ 90-Day Certificates	<input checked="" type="checkbox"/> ∞ 90-Day Certificates	<input checked="" type="checkbox"/> ∞ 90-Day Certificates
<input type="checkbox"/> X 1-Year Certificates	<input checked="" type="checkbox"/> 3 1-Year Certificates	<input checked="" type="checkbox"/> 10 1-Year Certificates	<input checked="" type="checkbox"/> 25 1-Year Certificates
<input type="checkbox"/> X Multi-Domain Certs	<input checked="" type="checkbox"/> ✓ Multi-Domain Certs	<input checked="" type="checkbox"/> ✓ Multi-Domain Certs	<input checked="" type="checkbox"/> ✓ Multi-Domain Certs
<input type="checkbox"/> X 90-Day Wildcards	<input type="checkbox"/> X 90-Day Wildcards	<input checked="" type="checkbox"/> ∞ 90-Day Wildcards	<input checked="" type="checkbox"/> ∞ 90-Day Wildcards
<input type="checkbox"/> X 1-Year Wildcards	<input type="checkbox"/> X 1-Year Wildcards	<input checked="" type="checkbox"/> 1 1-Year Wildcards	<input checked="" type="checkbox"/> 3 1-Year Wildcards
<input type="checkbox"/> X REST API Access	<input checked="" type="checkbox"/> ✓ REST API Access	<input checked="" type="checkbox"/> ✓ REST API Access	<input checked="" type="checkbox"/> ✓ REST API Access
<input type="checkbox"/> X Technical Support	<input checked="" type="checkbox"/> ✓ Technical Support	<input checked="" type="checkbox"/> ✓ Technical Support	<input checked="" type="checkbox"/> ✓ Technical Support

Next Step →

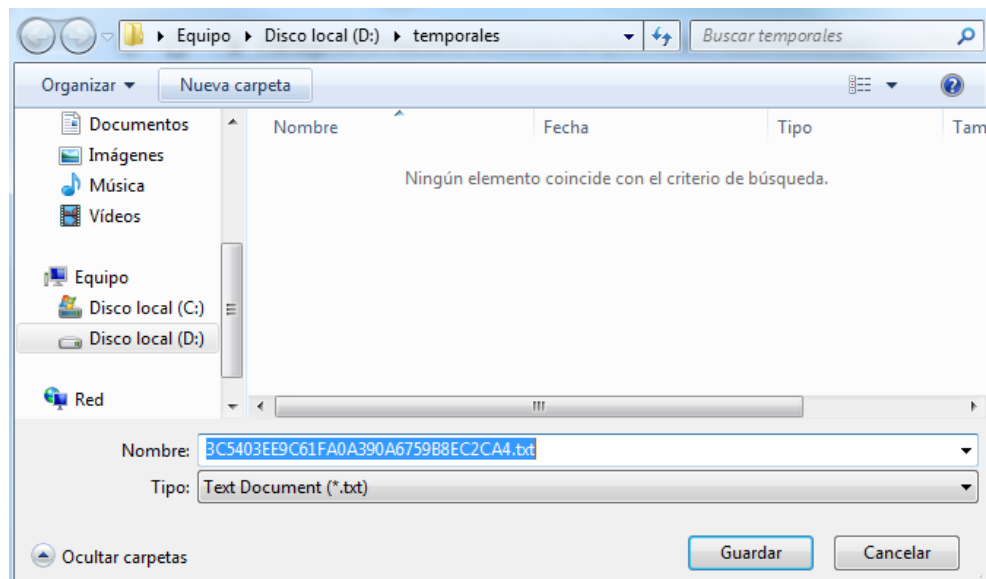
En la pantalla de verificación de la propiedad del dominio, debe especificar que la verificación se realizará mediante HTTP File Upload



Debe seguir los pasos indicados para la verificación de la propiedad del dominio mediante HTTP File Upload.

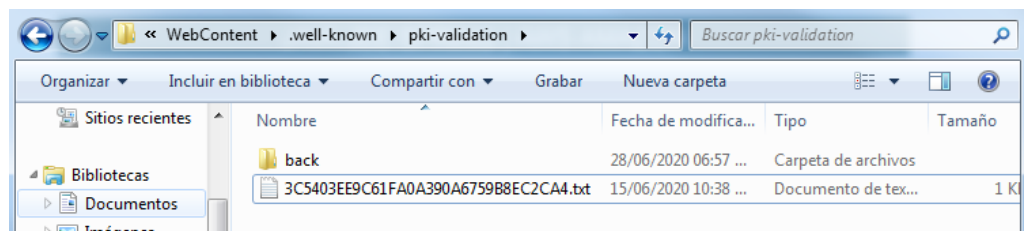
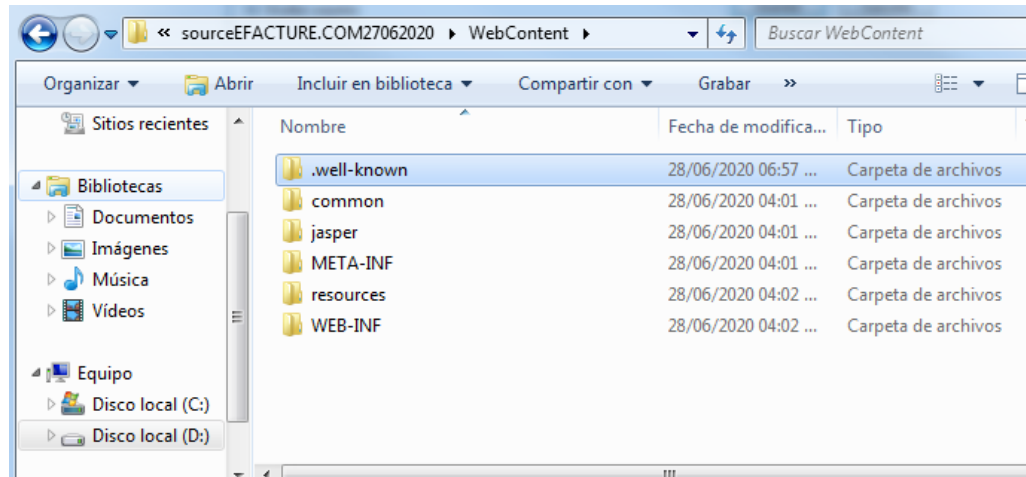
1. *Download your Auth File using the following link: Download Auth File*

Descargar el archivo TXT de verificación de la url proporcionada "Download Auth File"



2. Upload the Auth File to your HTTP server under: `/.well-known/pki-validation/`

Crear las carpetas “.well-known/pki-validation” dentro WebContent del proyecto y copiar en esta carpeta el .TXT de verificación que se ha descargado en el paso anterior.



Utilice el cliente SFTP para subir el archivo de verificación TXT al entorno de AWS, por lo regular estos archivos se suben a la carpeta “/home/ec2-user/” dentro del AWS.

mediante Putty ejecute el siguiente comando para transferir el archivo de verificación .TXT a la carpeta `/.well-known/pki-validation/`

```
sudo cp -fR 551625181947837EF2524EFFC37801FE.txt  
/var/lib/tomcat8/webapps/ROOT/.well-known/pki-validation/
```

3. *Make sure your file is available under the following link:*

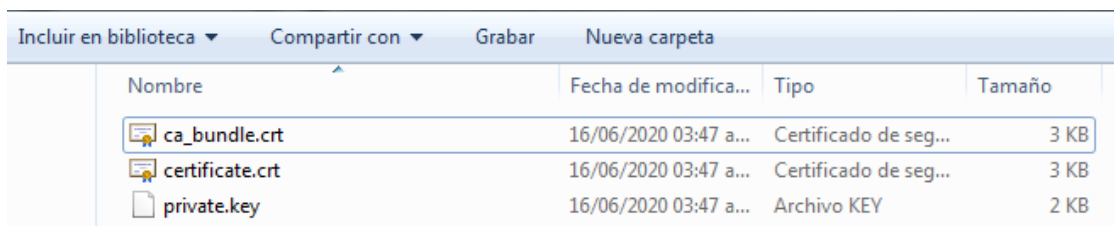
<http://seefacture.com/.well-known/pki-validation/3C5403EE9C61FA0A390A6759B8EC2CA4.txt>

es importante que el enlace a la URL sea valida:

<http://seefacture.com/.well-known/pki-validation/3C5403EE9C61FA0A390A6759B8EC2CA4.txt>

4. *Click "Next Step" to continue.*

Luego realizar la verificación exitosa de la propiedad del dominio, se descargara los archivos correspondientes al certificado de seguridad SSL “CRT y KEY” para el dominio.



Nombre	Fecha de modifica...	Tipo	Tamaño
ca_bundle.crt	16/06/2020 03:47 a...	Certificado de seg...	3 KB
certificate.crt	16/06/2020 03:47 a...	Certificado de seg...	3 KB
private.key	16/06/2020 03:47 a...	Archivo KEY	2 KB

Subir los archivos CRT y KEY mediante el cliente SFTP a la carpeta

“/home/ec2-user/ssl” en el AWS, previamente se debió crear la carpeta ssl.

ejecutar los comandos mediante putty

```
cd /home/ec2-user/
```

```
sudo mkdir ssl
```

transferir mediante SFTP los archivos ca_bundle.crt, certificate.crt y private.key a

```
/home/ec2-user/ssl
```

```
sudo cp -fR ssl/ /etc/tomcat8/ssl
```

esto copiará los archivos ca_bundle.crt, certificate.crt y private.key a la carpeta ssl del home del tomcat en el AWS “/etc/tomcat8/ssl”
ejecute los siguientes comandos mediante putty

```
cd /etc/tomcat8/ssl
```

```
sudo openssl pkcs12 -export -in certificate.crt -inkey private.key -name  
efacture -out efacture-free-PKCS-12.p12
```


coloque como password “efacture”

```
[ec2-user@ip-172-31-39-2 ssl]$ sudo openssl pkcs12 -export -in certificate.crt -inkey private.key -name efacture -out efacture-free-PKCS-12.p12
Enter Export Password:
Verifying - Enter Export Password:
[ec2-user@ip-172-31-39-2 ssl]$ █
```

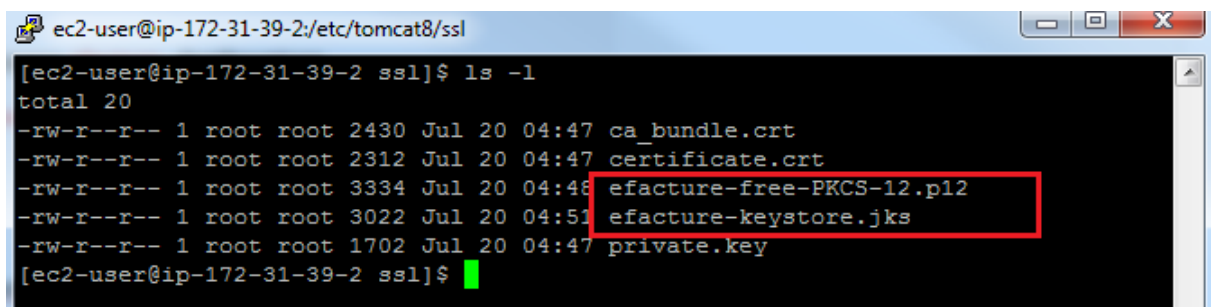
`sudo keytool -importkeystore -deststorepass efacture -destkeystore`

`efacture-keystore.jks -srckeystore efacture-free-PKCS-12.p12 -srcstoretype`

`PKCS12`

A la solicitud de password coloque “efacture”

verificar que los archivos `efacture-keystore.jks` y `efacture-free-PKCS-12.p12` se hayan generado.



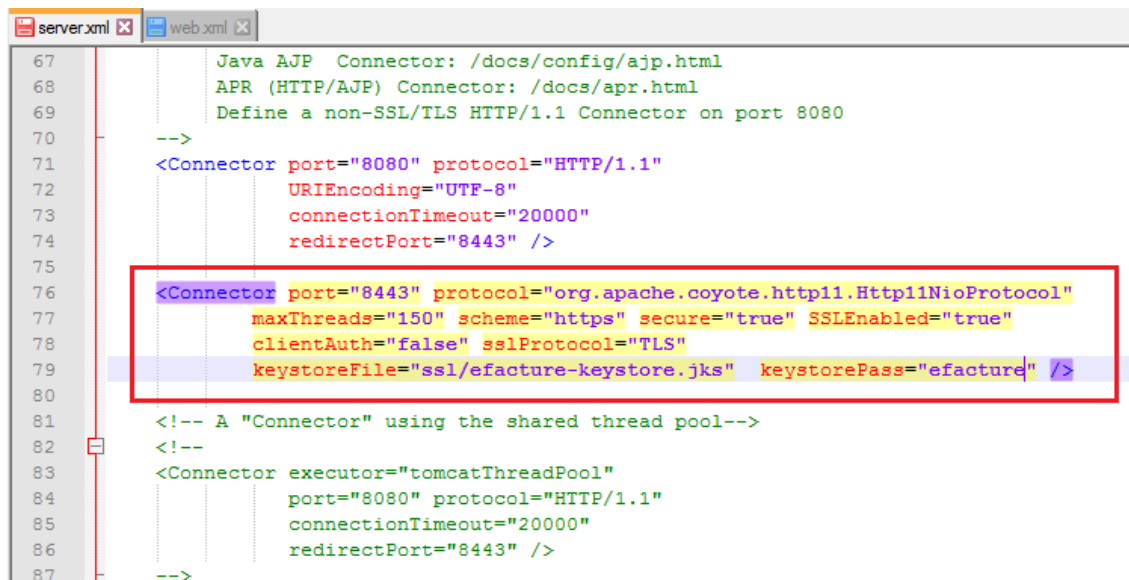
```
ec2-user@ip-172-31-39-2:/etc/tomcat8/ssl
[ec2-user@ip-172-31-39-2 ssl]$ ls -l
total 20
-rw-r--r-- 1 root root 2430 Jul 20 04:47 ca_bundle.crt
-rw-r--r-- 1 root root 2312 Jul 20 04:47 certificate.crt
-rw-r--r-- 1 root root 3334 Jul 20 04:48 efacture-free-PKCS-12.p12
-rw-r--r-- 1 root root 3022 Jul 20 04:51 efacture-keystore.jks
-rw-r--r-- 1 root root 1702 Jul 20 04:47 private.key
[ec2-user@ip-172-31-39-2 ssl]$ █
```

Luego de la creación de los archivos .p12 y jks se debe copiar toda carpeta ssl a la carpeta `/usr/share/tomcat8`

`sudo cp -fR /etc/tomcat8/ssl/ /usr/share/tomcat8/`

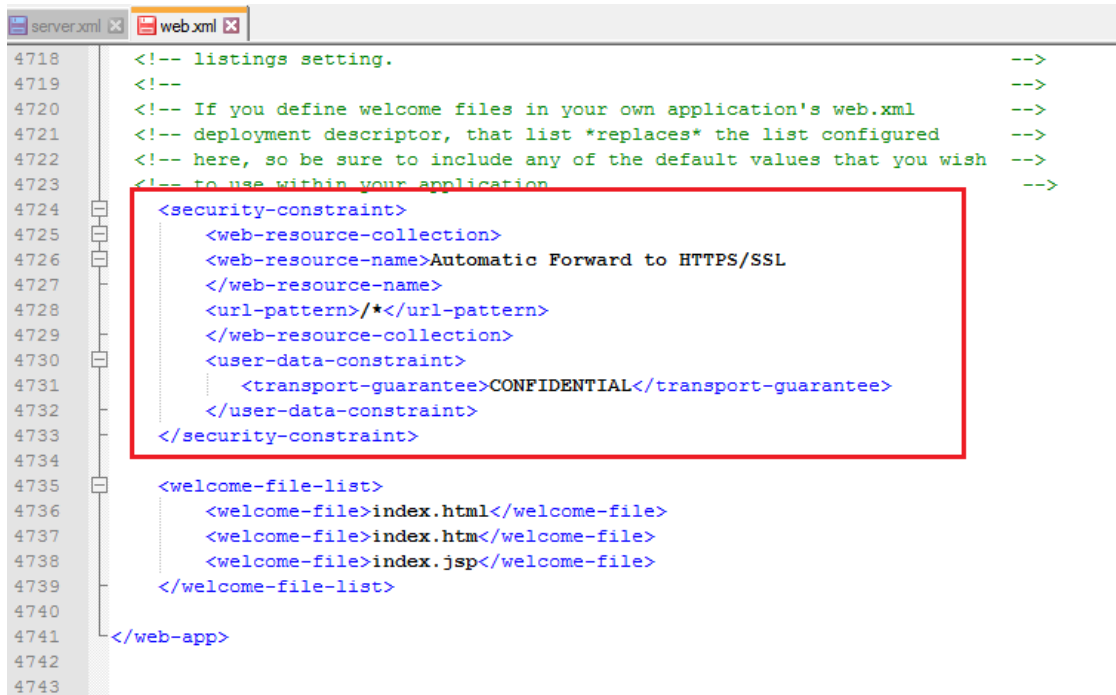
Luego debe descargar los archivos mediante SFTP los archivos de configuración del servidor Web “server.xml” y “web.xml” como se indica en las imágenes. esto archivos se encuentran de configuración del servidor web se encuentran en /etc/tomcat8/

Una vez descargado estos archivos de configuración a la unidad local, modificarlos agregando las secciones como se indica en las siguientes figuras:



```
67      Java AJP Connector: /docs/config/ajp.html
68      APR (HTTP/AJP) Connector: /docs/apr.html
69      Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
70      -->
71      <Connector port="8080" protocol="HTTP/1.1"
72              URIEncoding="UTF-8"
73              connectionTimeout="20000"
74              redirectPort="8443" />
75
76      <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
77              maxThreads="150" scheme="https" secure="true" SSLEnabled="true"
78              clientAuth="false" sslProtocol="TLS"
79              keystoreFile="ssl/efacture-keystore.jks" keystorePass="efacture" />
80
81      <!-- A "Connector" using the shared thread pool-->
82      <!--
83      <Connector executor="tomcatThreadPool"
84              port="8080" protocol="HTTP/1.1"
85              connectionTimeout="20000"
86              redirectPort="8443" />
87      -->
```

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" scheme="https" secure="true" SSLEnabled="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="ssl/efacture-keystore.jks" keystorePass="efacture"/>
```



```
server.xml x web.xml x
4718 <!-- listings setting. -->
4719 <!-- -->
4720 <!-- If you define welcome files in your own application's web.xml -->
4721 <!-- deployment descriptor, that list *replaces* the list configured -->
4722 <!-- here, so be sure to include any of the default values that you wish -->
4723 <!-- to use within your application -->
4724 <security-constraint>
4725   <web-resource-collection>
4726     <web-resource-name>Automatic Forward to HTTPS/SSL
4727   </web-resource-name>
4728   <url-pattern>/*</url-pattern>
4729 </web-resource-collection>
4730   <user-data-constraint>
4731     <transport-guarantee>CONFIDENTIAL</transport-guarantee>
4732   </user-data-constraint>
4733 </security-constraint>
4734
4735 <welcome-file-list>
4736   <welcome-file>index.html</welcome-file>
4737   <welcome-file>index.htm</welcome-file>
4738   <welcome-file>index.jsp</welcome-file>
4739 </welcome-file-list>
4740
4741 </web-app>
4742
4743
```

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Viewpoint Secure URLs</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Subir los archivos modificados mediante SFTP a la carpeta home del usuario

ec2-user "/home/ec2-user"

Luego ejecute los siguientes comandos en el cliente Putty

```
sudo cp -fR server.xml /etc/tomcat8/
```

```
sudo cp -fR web.xml /etc/tomcat8/
```

por último debemos configurar las reglas del IPTABLES

Iptables es un firewall incluido en el kernel de los servidores dedicados Linux. Su funcionamiento se basa en aplicar reglas que el mismo firewall ejecute permitiendo accesos y redirección de puertos.

Ejecute los siguientes comandos desde el cliente putty a fin de redirigir las invocaciones del puerto 80 al 8080.

El puerto 8080 es el puerto HTTP de escucha por defecto del servidor web Tomcat en AWS.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

```
sudo iptables -t nat -I OUTPUT -p tcp -d 127.0.0.1 --dport 80 -j REDIRECT --to-ports 8080
```

```
sudo iptables -t nat --line-numbers -n -L
```

Ejecute los siguientes comandos desde el cliente putty a fin de redirigir las invocaciones del puerto 443 al 8443.

El puerto 8443 es el puerto HTTPS de escucha por defecto del servidor web Tomcat en AWS.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 8443
```

```
sudo iptables -t nat -I OUTPUT -p tcp -d 127.0.0.1 --dport 443 -j REDIRECT  
--to-ports 8443
```

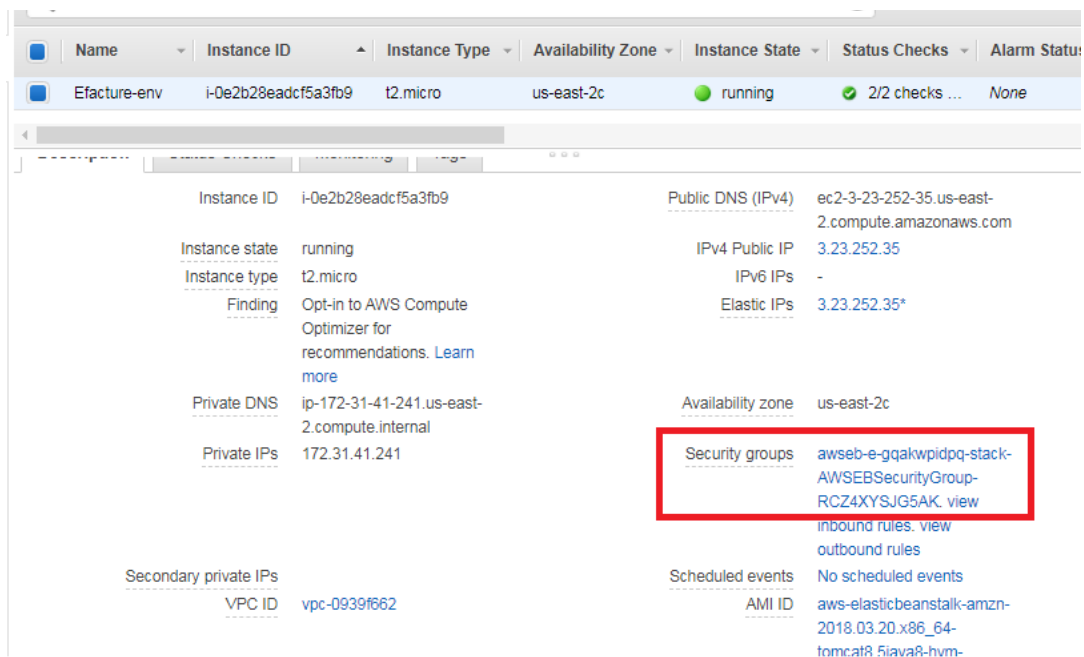
```
sudo iptables -t nat --line-numbers -n -L
```

Luego ejecutar el siguiente comando en el cliente putty para reiniciar el servicio del Tomcat del AWS para refrescar todos los cambios realizados.

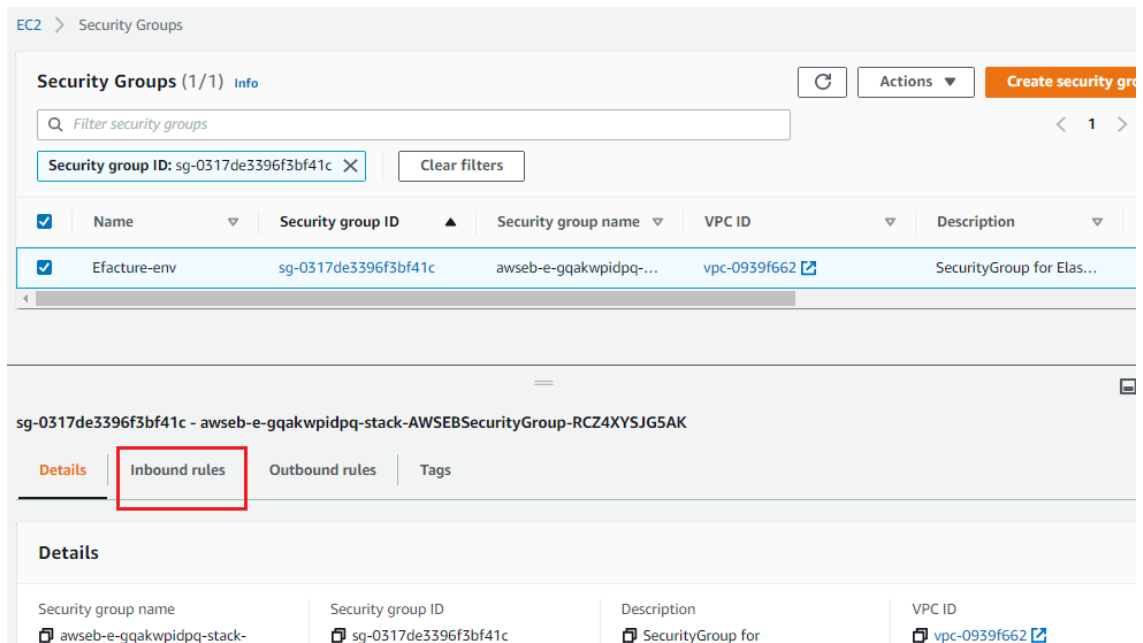
```
sudo service tomcat8 restart
```

Además de agregar las reglas de acceso y redirección de los puertos 8080 y 8443 mediante el IPTABLES, es importante habilitar los accesos a los puertos desde el AWS Amazon.

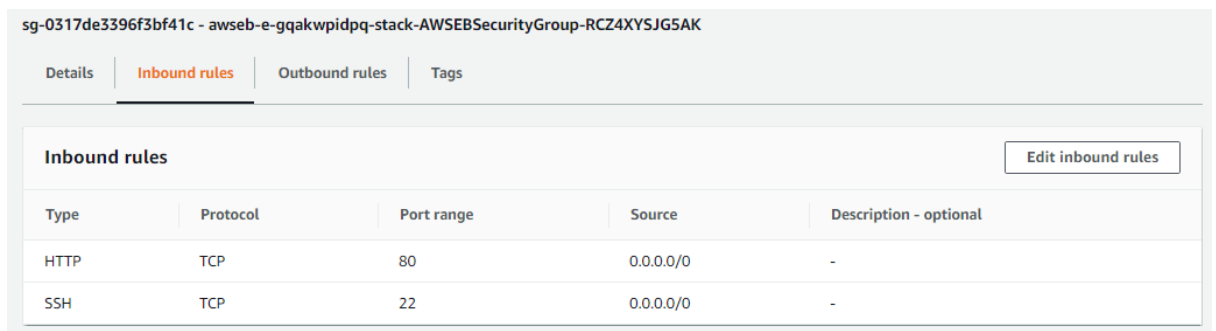
En la consola de administración EC2 en el AWS seleccione la instancia del proyecto, para luego hacer click en el enlace a la configuración de grupos de seguridad.



Se mostrará la consola de configuración de “Security Group”, en esta consola se debe configurar reglas en las secciones “Inbound rules”.



en la sección “Inbound rules”, realice click en el botón “Edit inbound rules”



En la pantalla “Edit inbound rules”, debe agregar las reglas HTTPS de puerto 443 y la regla Custom TCP para el puerto 8443, como se muestra en la siguiente figura:

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info
HTTP	TCP	80	Custom 0.0.0.0/0
SSH	TCP	22	Custom 0.0.0.0/0
HTTPS	TCP	443	Anywh... 0.0.0.0/0 ::/0
Custom TCP	TCP	8443	Anywh... 0.0.0.0/0 ::/0

13. Descargar fuentes JAVA-Spring del proyecto SEEFACTURE, Scripts SQL, videos, instaladores y demás recursos.

El costo de la asesoría personalizada y el acceso a los recursos es de 30.00 USD mediante paypal <https://paypal.me/javaaws>

Para soporte y asesoría comuníquese al WhatsApp <https://wa.link/7r5obp>

- Instalador de Eclipse Mars JEE
- Instalador de SQL Server 2012
- Video Demo de proyecto SEEFACTURE
- Video de Desarrollo sobre el proyecto
- Video de Asignación de HTTPS
- Arquitectura JEE Spring - AWS Amazon Cloud
- Framework desarrollo JEE Spring - AWS Amazon Cloud
- Pool de Conecciones JEE Spring - AWS Amazon Cloud
- Fuentes JAVA JEE - Spring del proyecto SEEFACTURE
- Script SQL de creación de Base de Datos del proyecto SEEFACTURE