

Aprovisiona una instancia en AWS

Jefferson A. Peña Torres
jefferson.amado.pena@correounivalle.edu.co

Consumir servicios expuestos con Node+Express

EC2

1.¿Qué es EC2?

2. Ambiente de desarrollo

3.Cloud 9

4.Aprovisiona MERN

Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo.

Aprovisiona una instancia en AWS

¿Qué es EC2?

En EC2 encuentras Entornos informáticos virtuales, conocidos como instancias. Plantillas preconfiguradas que tienen en cuenta sistemas operativos y stacks de desarrollo y que disponen de diversas configuraciones de CPU, memoria, almacenamiento.

Para acceder se necesita un identificador de cuenta, un nombre de usuario y una contraseña.



The screenshot shows the AWS IAM login interface. At the top is the AWS logo. Below it is the heading 'Iniciar sesión como usuario de IAM'. There are three input fields: 'ID de cuenta (12 dígitos) o alias de cuenta' containing '896886022616', 'Nombre de usuario:' containing 'jefferson.amado.pena', and 'Contraseña:' with masked characters. Below the password field is a checkbox labeled 'Recordar esta cuenta'. A blue 'Iniciar sesión' button is at the bottom. At the very bottom, there are links for 'Iniciar sesión con el email del usuario raíz' and '¿Olvidó la contraseña?'.

Aprovisiona una instancia en AWS

Ambiente de desarrollo

En EC2 encuentras Entornos informáticos virtuales, conocidos como instancias. Plantillas preconfiguradas que tienen en cuenta sistemas operativos y stacks de desarrollo y que disponen de diversas configuraciones de CPU, memoria, almacenamiento.

Para acceder se necesita un identificador de cuenta, un nombre de usuario y una contraseña.

Desde la consola de administración puedes revisar todas las opciones y servicios que tiene AWS

AWS Management Console

AWS services

► Recently visited services

► All services

Aprovisiona una instancia en AWS

Ambiente de desarrollo

1

En EC2 encuentras Entornos informáticos virtuales, conocidos como instancias. Plantillas preconfiguradas que tienen en cuenta sistemas operativos y stacks de desarrollo y que disponen de diversas configuraciones de CPU, memoria, almacenamiento.

2

3

4

Para acceder se necesita un identificador de cuenta, un nombre de usuario y una contraseña.

Desde la consola de administración puedes revisar todas las opciones y servicios que tiene AWS, en especial aquellos que han sido creados para los desarrolladores



Developer Tools

CodeStar

CodeCommit

CodeArtifact

CodeBuild

CodeDeploy

CodePipeline

Cloud9

CloudShell

X-Ray

AWS FIS

Aprovisiona una instancia en AWS

AWS Cloud9

Es un entorno de desarrollo integrado (IDE) que ofrece una completa experiencia de edición de código, con soporte para varios lenguajes de programación y depuradores de tiempo de ejecución, además de un terminal integrado. Contiene una colección de herramientas que se utilizan para codificar, compilar, ejecutar, probar y depurar software, y le ayuda a lanzar software en la nube.

Developer Tools

CodeStar

CodeCommit

CodeArtifact

CodeBuild

CodeDeploy

CodePipeline

Cloud9

CloudShell

X-Ray

AWS FIS

Aprovisiona una instancia en AWS

AWS Cloud9

Es un entorno de desarrollo integrado (IDE) que ofrece una completa experiencia de edición de código, con soporte para varios lenguajes de programación y depuradores de tiempo de ejecución, además de un terminal integrado. Contiene una colección de herramientas que se utilizan para codificar, compilar, ejecutar, probar y depurar software, y le ayuda a lanzar software en la nube.

Es posible acceder al IDE de AWS Cloud9 a través de un navegador web.

Developer Tools

CodeStar

CodeCommit

CodeArtifact

CodeBuild

CodeDeploy

CodePipeline

Cloud9

CloudShell

X-Ray

AWS FIS



El futuro digital
es de todos

MinTIC

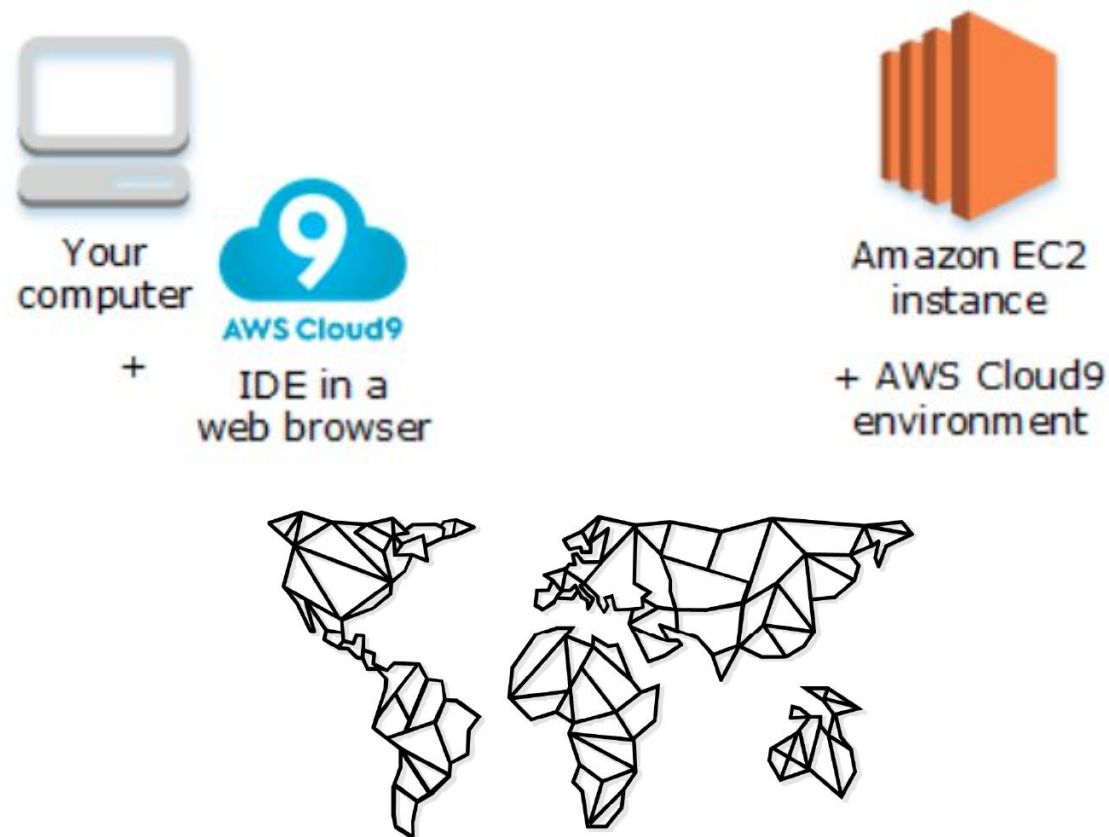


Aprovisiona una instancia en AWS

AWS Cloud9

Es un entorno de desarrollo integrado (IDE) que ofrece una completa experiencia de edición de código, con soporte para varios lenguajes de programación y depuradores de tiempo de ejecución, además de un terminal integrado. Contiene una colección de herramientas que se utilizan para codificar, compilar, ejecutar, probar y depurar software, y le ayuda a lanzar software en la nube.

Es posible acceder al IDE de AWS Cloud9 a través de un navegador web.

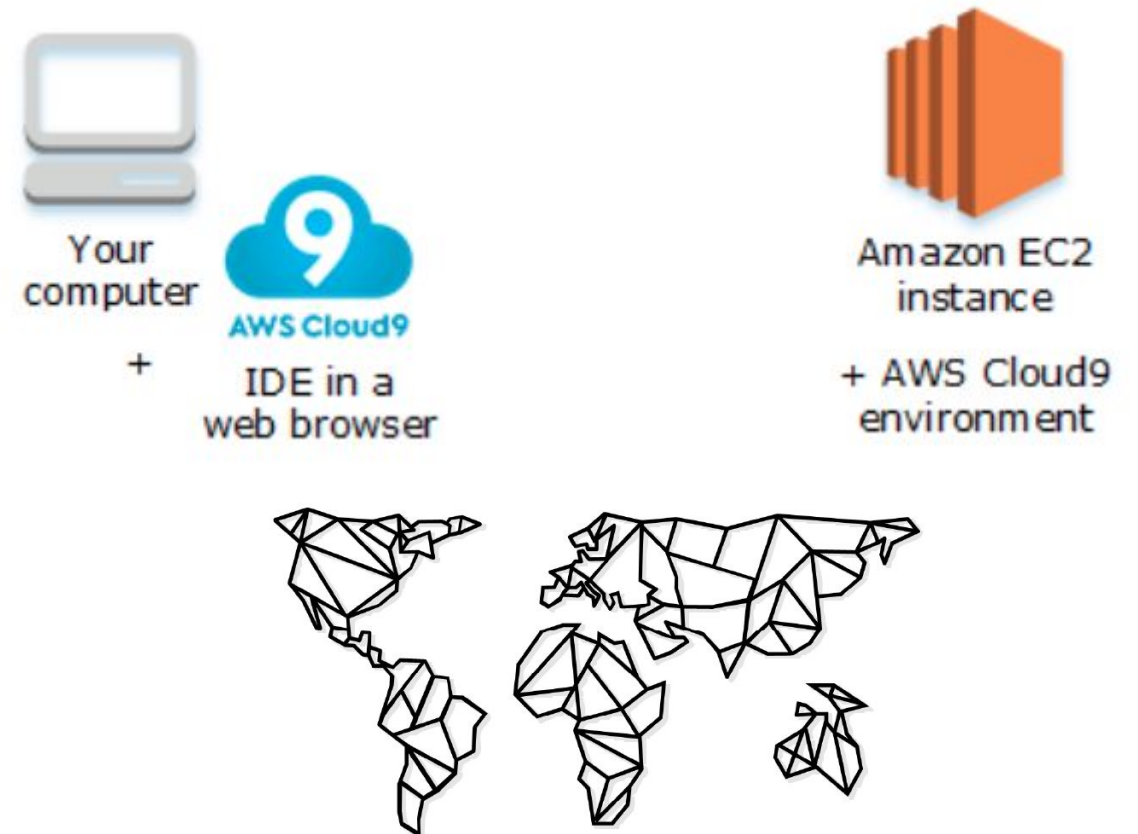


Aprovisiona una instancia en AWS

AWS Cloud9

Con AWS Cloud9, puede codificar, compilar, ejecutar, probar, depurar y publicar software en muchas situaciones y variaciones interesantes.

Trabajar con código en varios lenguajes de programación, utilizar contenedores de Docker, descargar repositorios de código online, colaborar con otros usuarios en tiempo real, entre otros



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

1

2

3

4

The screenshot displays the AWS Cloud9 IDE interface. The top menu bar includes 'AWS Cloud9', 'File', 'Edit', 'Find', 'View', 'Go', 'Run', 'Tools', 'Window', and 'Support'. The main workspace is divided into three sections: 'Environment' (left sidebar), 'Source Control' (middle sidebar), and 'AWS' (bottom sidebar). The 'Environment' sidebar shows a file tree with folders 'databases', 'legacy', 'node_modules', 'src', and 'tmp', and files 'formulario.html', 'index.js', and 'package.json'. The 'Source Control' sidebar is empty. The 'AWS' sidebar shows the 'AWS' logo. The main workspace is labeled 'Área de edición'. Below the workspace is a terminal window titled 'bash - "ip-10-0-0-0"' showing the following commands and output:

```
environment package-lock.json
jefferson.amado.pena:~ $ rm -rfv package-lock.json
removed 'package-lock.json'
jefferson.amado.pena:~ $ 
jefferson.amado.pena:~ $ ls
environment
jefferson.amado.pena:~ $
```

Terminal



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

1

En la parte inferior del IDE encontrarás la terminal, para obtener permisos de instalación y/o configuración de la instancia es necesario acceder al usuario raíz (root) del sistema operativo.

2

3

- Utilice el siguiente comando

4



```
bash - "root@ip- × (+)  
jefferson.amado.pena:~/environment $ sudo -s
```



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Las instancias tiene Amazon Linux una distribución de amazon para las instancias, se debe agregar el repositorio donde el sistema va obtener el instalador de MongoDB en este caso la versión 3.4. Se debe crear un archivo (.repo)

- Utilice el siguiente comando

bash - "root@ip- ×



```
jefferson.amado.pena:~/environment $ sudo -s
```

```
[root@ip-10-0-0-45 environment]# nano /etc/yum.repos.d/mongodb-org-3.4.repo
```

[Aprovisiona una instancia en AWS](#)

Aprovisiona M.E.R.N en AWS

nano es un editor de texto por consola que creará el archivo y permitirá agregar contenido.



El futuro digital
es de todos

MinTIC



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

^G Get Help
^X Exit

^O Write Out
^R Read File

^W Where Is
^_ Replace

^K Cut Text
^U Uncut Text

[Wrote 6 lines]
^J Justify
^T To Spell

^C Cur Pos
^_ Go To Line

^Y Prev Page
^V Next Page



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

Utiliza el control (^) y la letra O para
guardar

^G Get Help
^X Exit

^O Write Out
^R Read File

^W Where Is
^_ Replace

^K Cut Text
^U Uncut Text

[Wrote 6 lines]
^J Justify
^T To Spell

^C Cur Pos
^_ Go To Line

^Y Prev Page
^V Next Page



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

Verifica que sea el nombre correcto y
oprime Enter

File Name to Write: /etc/yum.repos.d/mongodb-org-3.4.repo

^G Get Help

M-D DOS Format

M-A Append

M-B Backup File

^C Cancel

M-M Mac Format

M-P Prepend

^T To Files



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

[mongodb-org-3.4]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/3.4/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc

Utiliza el control (^) y la letra X para salir de
nano

^G Get Help
^X Exit

^O Write Out
^R Read File

^W Where Is
^_ Replace

^K Cut Text
^U Uncut Text

[Wrote 6 lines]
^J Justify
^T To Spell

^C Cur Pos
^_ Go To Line

^Y Prev Page
^V Next Page

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Una vez se ha incluido el repositorio, se debe utilizar el instalador del sistema operativo para que obtenga las rutas donde se encuentra mongoDB. Para la instancia será yum.

- Utiliza el siguiente comando

```
bash - "root@ip-10-0-0-45"
[root@ip-10-0-0-45 environment]# nano /etc/yum.repos.d/mongodb-org-3.4.repo
[root@ip-10-0-0-45 environment]# sudo yum update
```

Actualiza los repositorios para que el SO pueda obtener las rutas donde se encuentra MongoDB

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Una vez se ha incluido el repositorio, se debe utilizar el instalador del sistema operativo para que obtenga las rutas donde se encuentra mongoDB. Para la instancia será yum.

- Utiliza el siguiente comando

```
bash - "root@ip-10-0-0-45"
[root@ip-10-0-0-45 environment]# nano /etc/yum.repos.d/mongodb-org-3.4.repo
[root@ip-10-0-0-45 environment]# sudo yum update
```

Actualiza los repositorios para que el SO pueda obtener las rutas donde se encuentra MongoDB

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Una vez se han actualizado los repositorios, procedemos con la instalación de mongo.

- Utilice el siguiente comando

```
bash - "root@ip-10-0-0-45 environment" # sudo yum install -y mongodb-org
```

INSTALAR MONGODB

```
sudo yum install -y mongodb-org
```




Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

La instalación de dependencias y paquetes puede tomar unos minutos

bash - "root@ip- ×



```
[root@ip-10-0-0-45 environment]# sudo yum install -y mongodb-org
Loaded plugins: priorities, update-motd, upgrade-helper
Repository mongodb-org-3.4 is listed more than once in the configuration
1114 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
----> Package mongodb-org.x86_64 0:3.4.24-1.amzn1 will be installed
--> Processing Dependency: mongodb-org-tools = 3.4.24 for package: mongodb-org-3.4.24-1.amzn1.x86_64
--> Processing Dependency: mongodb-org-shell = 3.4.24 for package: mongodb-org-3.4.24-1.amzn1.x86_64
--> Processing Dependency: mongodb-org-server = 3.4.24 for package: mongodb-org-3.4.24-1.amzn1.x86_64
--> Processing Dependency: mongodb-org-mongos = 3.4.24 for package: mongodb-org-3.4.24-1.amzn1.x86_64
--> Running transaction check
----> Package mongodb-org-mongos.x86_64 0:3.4.24-1.amzn1 will be installed
----> Package mongodb-org-server.x86_64 0:3.4.24-1.amzn1 will be installed
----> Package mongodb-org-shell.x86_64 0:3.4.24-1.amzn1 will be installed
----> Package mongodb-org-tools.x86_64 0:3.4.24-1.amzn1 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Pero al terminar se verá en la terminal el mensaje
completed

bash - "root@ip-10-0-0-45" x

```
Installing : mongodb-org-tools-3.4.24-1.amzn1.x86_64
Installing : mongodb-org-mongos-3.4.24-1.amzn1.x86_64
Installing : mongodb-org-shell-3.4.24-1.amzn1.x86_64
Installing : mongodb-org-3.4.24-1.amzn1.x86_64
Verifying  : mongodb-org-shell-3.4.24-1.amzn1.x86_64
Verifying  : mongodb-org-mongos-3.4.24-1.amzn1.x86_64
Verifying  : mongodb-org-tools-3.4.24-1.amzn1.x86_64
Verifying  : mongodb-org-server-3.4.24-1.amzn1.x86_64
Verifying  : mongodb-org-3.4.24-1.amzn1.x86_64
```

Installed:

mongodb-org.x86_64 0:3.4.24-1.amzn1

Dependency Installed:

mongodb-org-mongos.x86_64 0:3.4.24-1.amzn1

mongodb-org-tools.x86_64 0:3.4.24-1.amzn1

mongodb-org-server.x86_64 0:3.4.24-1.amzn1

mongodb-org

Complete!

[root@ip-10-0-0-45 environment]#



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

MongoDB sigue la estructura de cualquier aplicación web, tiene una capa de datos, una de servidor y una cliente. La capa del servidor se llama mongod (mongo daemon) que levanta la aplicación servidor en el puerto por defecto

- Utiliza el siguiente comando

```
bash - "root@ip-10-0-0-45 environment" # mongod
```

mongod es el comando para levantar el **servicio** que ofrece los recursos de mongo.



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

MongoDB sigue la estructura de cualquier aplicación web, tiene una capa de datos, una de servidor y una cliente. La capa del servidor se llama mongod (mongo daemon) que levanta la aplicación servidor en el puerto por defecto

Puerto por defecto

Máquina que ejecuta la
aplicación servidor

Nos falta crear la
capa de datos

```
bash - "root@ip-10-0-0-45" x
[root@ip-10-0-0-45 environment]# mongod
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] MongoDB starting : pid=8315 port=27017 dbpath=/data/db 64-bit host=ip-10-0-0-45
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] db version v3.4.24
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] git version: 865b4f6a96d0f5425e39a18337105f33e8db504d
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.0-fips 29 Mar 2010
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] allocator: tcmalloc
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] modules: none
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] build environment:
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten]     distmod: amazon
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten]     distarch: x86_64
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten]     target_arch: x86_64
2021-11-05T15:18:42.732+0000 I CONTROL [initandlisten] options: {}
2021-11-05T15:18:42.774+0000 I STORAGE [initandlisten] exception in initAndListen: 29 Data directory /data/db not found., terminating
2021-11-05T15:18:42.774+0000 I NETWORK [initandlisten] shutdown: going to close listening sockets...
2021-11-05T15:18:42.774+0000 I NETWORK [initandlisten] shutdown: going to flush diaglog...
2021-11-05T15:18:42.774+0000 I CONTROL [initandlisten] now exiting
2021-11-05T15:18:42.774+0000 I CONTROL [initandlisten] shutting down with code:100
```



Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

MongoDB sigue la estructura de cualquier aplicación web, tiene una capa de datos, una de servidor y una cliente. En ocasiones es necesario indicar donde será alojado la capa de datos.

- Utiliza el siguiente comando

bash - "root@ip-"

```
[root@ip-10-0-0-45 environment]# mkdir /data  
[root@ip-10-0-0-45 environment]# mkdir /data/db
```

Crear la carpeta
data

Crear la carpeta **db**

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

MongoDB sigue la estructura de cualquier aplicación web, tiene una capa de datos, una de servidor y una cliente. La capa del servidor se llama mongod (mongo daemon) que levanta la aplicación servidor en el puerto por defecto

Ejecuta nuevamente
mongod

```
mongod - "root@x"
2021-11-05T15:24:47.512+0000 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=256M,session_max=20000,eviction=(thread:
4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=10000
60,log_size=2GB),statistics_log=(wait=0),verbose=(recovery_progress),
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten]
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommend
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten]
2021-11-05T15:24:47.813+0000 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory '/data/db/diagnosti
2021-11-05T15:24:47.837+0000 I INDEX [initandlisten] build index on: admin.system.version properties: { v: 2, key: { version: 1 }, nam
h_version_32", ns: "admin.system.version" }
2021-11-05T15:24:47.837+0000 I INDEX [initandlisten] building index using bulk method; build may temporarily use up to 500 me
2021-11-05T15:24:47.838+0000 I INDEX [initandlisten] build index done. scanned 0 total records. 0 secs
2021-11-05T15:24:47.838+0000 I COMMAND [initandlisten] setting featureCompatibilityVersion to 3.4
2021-11-05T15:24:47.839+0000 I NETWORK [thread1] waiting for connections on port 27017
```

Ahora sí!!
mongo en ejecución

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

MongoDB sigue la estructura de cualquier aplicación web, tiene una capa de datos, una de servidor y una cliente. La capa del cliente se llama utilizando el comando mongo

```
mongo - "ip-10-0 x" mongod - "root@ x"
jefferson.amado.pena:~/environment $ mongo
MongoDB server version: 4.0.1:27017
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2021-11-05T15:24:47.512+0000 I STORAGE [initandlisten]
2021-11-05T15:24:47.512+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger :
2021-11-05T15:24:47.512+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten]
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-11-05T15:24:47.761+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
```

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Node está instalado en las instancias, con este entorno de ejecución de javascript y su administrador de paquetes (npm), instalaremos el resto del stack de desarrollo.

- Verifica que está instalado con este comando

```
bash - "ip-10-0-0-  
jefferson.amado.pena:~ $ node -v  
v10.24.1  
jefferson.amado.pena:~ $
```

En la instancia se encuentra node

[Aprovisiona una instancia en AWS](#)

Aprovisiona M.E.R.N en AWS

1 Crea y nombra de manera coherente las carpetas que
2 alojarán las capas de tu aplicación web, no se debe
3 considerar datos en la instalación ya que mongo nos
4 provee esa capa.

```
bash - "ip-10-0-0- × (+)
jefferson.amado.pena:~ $ mkdir backend
jefferson.amado.pena:~ $ mkdir frontend
jefferson.amado.pena:~ $ █
```

Crea los directorios donde se alojarán las aplicaciones de cada capa

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Para iniciar la instalación de los elementos del back es necesario inicializar

1

2

3

4

```
npm - "ip-10-0-0- x
jefferson.amado.pena:~ $ cd backend/
jefferson.amado.pena:~/backend $ npm -y init
Wrote to /home/ec2-user/backend/package.json:

{
  "name": "backend",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Ingresa al directorio

Iniciar con npm

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

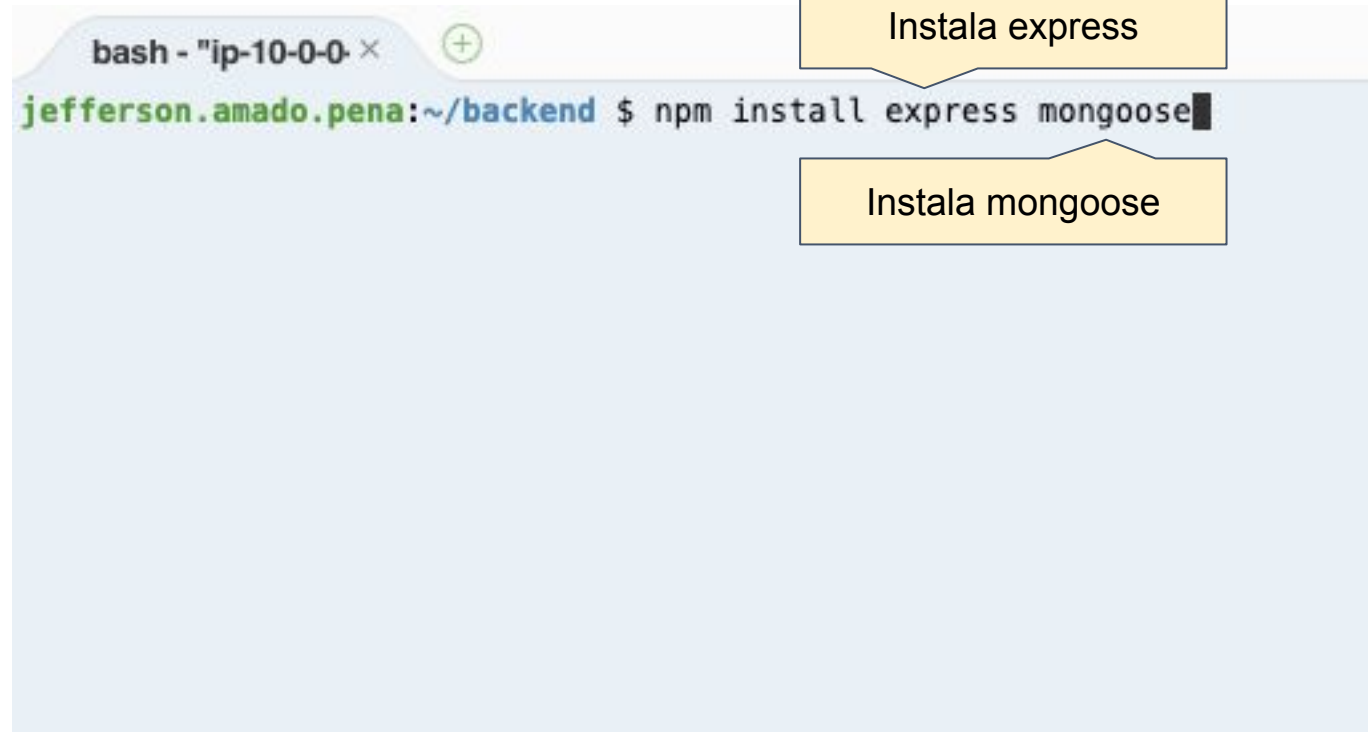
Después de inicializar el proyecto instala las dependencias express y mongoose.

1

2

3

4



```
bash - "ip-10-0-0-... x
jefferson.amado.pena:~/backend $ npm install express mongoose
```

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Luego en la carpeta donde residirá la aplicación que se ejecuta en el cliente

1

2

3

4

```
bash - "ip-10-0-0" ×
jefferson.amado.pena:~/backend $ npm install express mongoose^C
jefferson.amado.pena:~/backend $ cd ..
jefferson.amado.pena:~ $ cd frontend/
jefferson.amado.pena:~/frontend $
```

Cambia de directorio al frontend

Aprovisiona una instancia en AWS

Aprovisiona M.E.R.N en AWS

Luego en la carpeta donde residirá la aplicación que se ejecuta en el cliente instala el framework para SPA. En este caso ReactJS utilizando create-react-app

```
bash - "ip-10-0-0" ×  
jefferson.amado.pena:~/frontend $ npx create-react-app .
```

Creador de aplicaciones
SPA con React



www.upb.edu.co/es/mision-tic
#MisiónTICSomosTodos