## Strapi Setup & Installation

**Create strapi Project (CLI)**

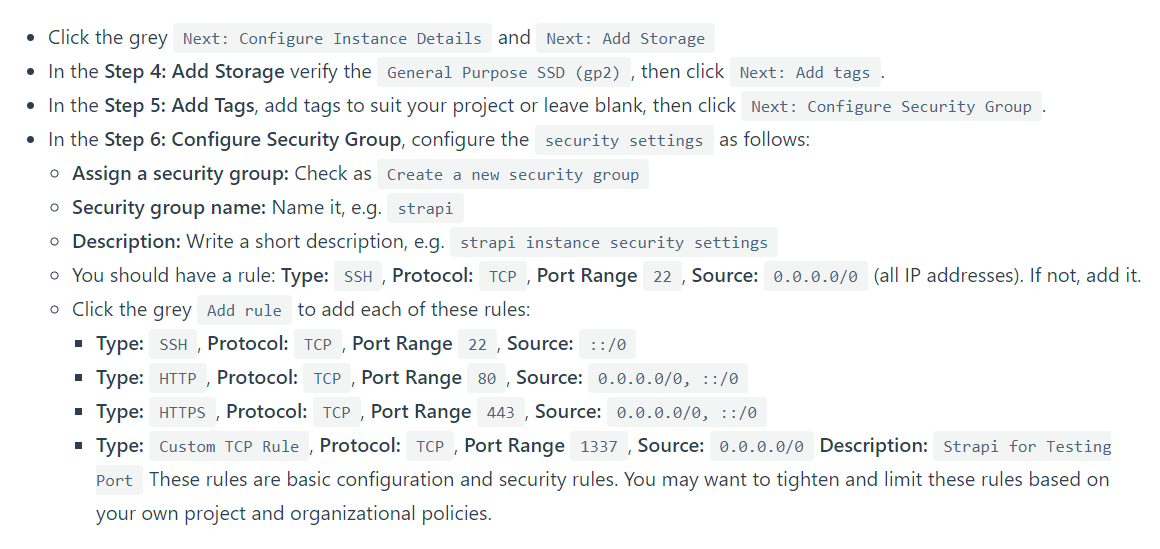
1. Make sure node is installed on local machine (node -v)
2. Create strapi app command -> **npx create-strapi-app@latest my-project --quickstart** (optional flag)
3. For running strapi - **yarn install -> sudo yarn develop**

**Deployment Amazon AWS**

1. Make sure to not use root account user & to use instead IAM service
2. Next, create a regular user for the creation and management of your Strapi project.
   1. In the **Set user details** screen:
   2. Provide a **User name**.
   3. **Access Type**: Check both Programmatic access and AWS Management Console access.
   4. Autogenerate a password or click Custom password and provide one.
   5. In the **Set Permissions** screen, do the following:
   6. Click Create group, name it, e.g. Developers, and then choose appropriate policies under **Policy Name**:
   7. search for ec2 and check AmazonEC2FullAccess
   8. search for RDS and check AmazonRDSFullAccess
   9. search for s3 and check AmazonS3FullAccess
   10. Click Create group
   11. Click to Add user to group and check the Developers group, to add the new user.

3. **Success** - These are very **IMPORTANT CREDENTIALS** *If you do not do these steps you will have to reset your Access key ID and Secret access key later.*

4. Create an **ec2** instance with an appropriate region & click Launch Instance & follow below details -



1. Click Review & Launch & select an existing key pair or create new key pair & download the private key file (.pem file). The file is needed, so note where it was downloaded. .

**Configure EC2 as a Node.js server**

1. Note the **IPv4 Public OP** of the current Running Instance.
2. .pem file needs to be included in each attempt to SSH into your EC2 server. Move your .pem file to ~/.ssh/. On your local machine, navigate to folder containing .pem file & move it to .ssh & set file permissions.

**mv ec2-strapi-key-pair.pem ~/.ssh/**

**chmod 400 ~/.ssh/ec2-strapi-key-pair.pem**

Or below can be used temporarily, for instance –

“chmod 400 test-key-pair.pem”

“ssh -i test-key-pair.pem [ec2-user@”AWSIPv4”](mailto:ec2-user@52.72.128.48)”

1. Install Node.js with npm

**cd ~**

**curl -sL https://deb.nodesource.com/setup\_14.x | sudo -E bash -**

**...**

**sudo apt-get install nodejs**

**...**

**node -v && npm –v**

#### Create and change npm's default directory.



**Install pg(PostgresSQL) in strapi project**

1. On your development machine, navigate to your Strapi project root directory **pm install pg**
2. Copy/paste the following to **./my-project/config/database.js** –

**module.exports = ({ env }) => ({**

**connection: {**

**client: "postgres",**

**connection: {**

**host: env("DATABASE\_HOST", "127.0.0.1"),**

**port: env.int("DATABASE\_PORT", 5432),**

**database: env("DATABASE\_NAME", "strapi"),**

**user: env("DATABASE\_USERNAME", ""),**

**password: env("DATABASE\_PASSWORD", ""),**

**},**

**useNullAsDefault: true,**

**},**

**});**

**Configure S3 for image hosting**

1. Create a bucket & select most appropriate region.
2. Under **Block public access**:
   1. Uncheck Block all public access and set the permissions as follows:
      1. Uncheck Block new public ACLs and uploading public objects (Recommended)
      2. Uncheck Block public access to buckets and objects granted through any access control lists (ACLs)
      3. Check Block public access to buckets and objects granted through new public bucket policies
      4. Check Block public and cross-account access to buckets and objects through any public bucket policies
   2. Select Do not grant Amazon S3 Log Delivery group write access to this bucket.
3. Click Next.
4. **Review** and click Create bucket
5. Install the Strapi AWS S3 Upload Provider at ./my-project/**: npm install @strapi/provider-upload-aws-s3**
6. To enable and configure the provider, create or edit the file at ./config/plugins.js (file name should be plural)

module.exports = ({ env }) => ({

upload: {

config: {

provider: 'aws-s3',

providerOptions: {

accessKeyId: env('AWS\_ACCESS\_KEY\_ID'),

secretAccessKey: env('AWS\_ACCESS\_SECRET'),

region: env('AWS\_REGION'),

params: {

Bucket: env('AWS\_BUCKET\_NAME'),

},

},

// These parameters could solve issues with ACL public-read access — see [this issue](https://github.com/strapi/strapi/issues/5868) for details

actionOptions: {

upload: {

ACL: null

},

uploadStream: {

ACL: null

},

}

},

}

});

Points to note –

Go to Bucket > Permissions Tab > acls enabled & Bucket owner preferred checked

Try uploading media into media library & copy link & check . The link should be of S3 & the media should get uploaded to the designated bucket.

**Make required changes in .env file –**

DATABASE\_HOST=your-unique-url.rds.amazonaws.com  
DATABASE\_PORT=5432  
DATABASE\_NAME=strapi  
DATABASE\_USERNAME=postgres  
DATABASE\_PASSWORD=Password  
AWS\_ACCESS\_KEY\_ID=aws-access-key-id  
AWS\_ACCESS\_SECRET=aws-access-secret  
AWS\_REGION=aws-region  
AWS\_BUCKET\_NAME=my-project-bucket-name

**Commands to build, run & develop strapi project on npm**

* npm run build
* npm run start
* npm run develop
* nohup npm run develop