

CHAMPLAIN COLLEGE LENNOXVILLE  
INTEGRATIVE PROJECT - (420-610-LE)

VALCOURT 2030 DEPLOYMENT DOCUMENTATION

FROM:

VLADIMIR ESTEVEZ (LE 0679993). - MUNIR KHALIQYAR (LE2176022)  
JEREMY BLACKBURN (LE 2152908) AND GUILLAUME BLAIS (LE2027247)

420-350-LE USING DATABASES  
COMPUTER SCIENCE TECHNOLOGY

PRESENTED TO GABRIEL GASTUDILLO  
DUE DATE: 17th DECEMBER 2024

Documentation for Deployment of the Valcourt 2030 Application .....	3
Prerequisites .....	3
Setting up webhook in WordPress .....	3
Configuring tags in WordPress.....	7
Deploying a React Application to AWS Elastic Beanstalk .....	9
Deployment Process: .....	9
Create a zipped version of the build of the Valcour2030 application. ....	9
Initialization and environment creation: .....	11
Creating the MongoDB Database .....	15
Accessing the Application: .....	19
Creating a new Event on the WordPress Website.....	20

# Documentation for Deployment of the Valcourt 2030 Application

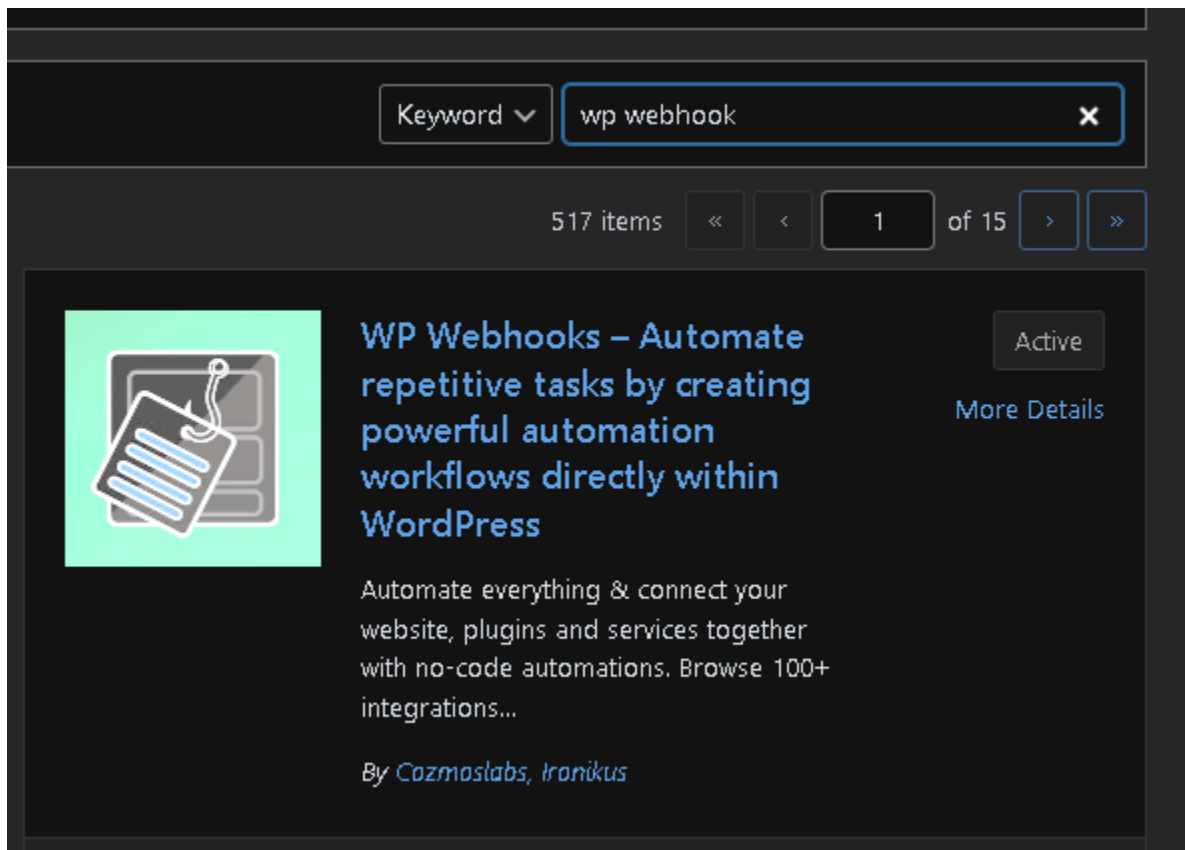
## Prerequisites

### Setting up webhook in WordPress

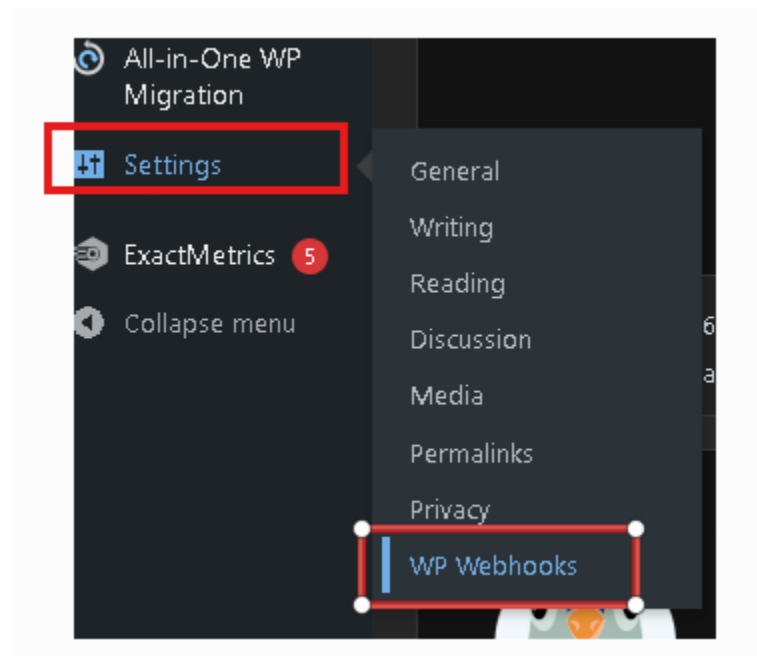
To be able set up the Webhooks part of the app you will have to install the WebHooks Plugin:

First, we will need to install the Webhook plugin:

- 1) Go in the PPlugin Section of your WordPress search for the following plugin and install it:



2) Go to your WP settings and click on WP WebHooks:



3) Click on Send Data > Post created > Add Webhook URL

The screenshot shows the 'Available Webhook Triggers' page. The top navigation bar includes 'Home', 'Send Data', 'Receive Data', 'Flows', 'Authentication', 'Data Mapping', 'Logs', 'Settings', and 'Pro'. The 'Send Data' menu item is highlighted with a red box. Below the navigation bar, the page title 'Available Webhook Triggers' is displayed. A paragraph explains that users will find a list of available WP Webhooks triggers and need to specify a URL to be triggered. Below this, there is a search bar and a list of triggers. The 'WP Webhooks' category is expanded, showing 'Action fired', 'Custom button clicked', and 'Custom link clicked'. The 'WordPress' category is also expanded, showing 'Comment created', 'Comment deleted', 'Comment trashed', 'Comment updated', 'Custom trigger called', and 'Post created'. The 'Post created' trigger is highlighted with a red box. To the right of the trigger list, the details for the 'Post created' trigger are shown. It includes the WordPress logo, the trigger name 'Post created', and a description: 'This webhook fires after a new post was created.' The 'Add Webhook URL' button is highlighted with a red box. Below the description, there is a table with columns 'Webhook Name', 'Webhook URL', and 'Action'. The table contains two rows: 'Outgoing data' and 'Description', both with an 'Expand' button.

Home **Send Data** Receive Data ▼ Flows Authentication Data Mapping Logs Settings ▼ Pro

## Available Webhook Triggers

Below you will find a list of all available WP Webhooks triggers. To use one, you need to specify a URL that should be triggered to send the available data. For more information on that, you can check out each webhook trigger description or our product documentation by clicking [here](#).

Search triggers

WP Webhooks

Action fired

Custom button clicked Pro

Custom link clicked Pro

WordPress

Comment created

Comment deleted

Comment trashed

Comment updated

Custom trigger called

Post created

WordPress

Post created

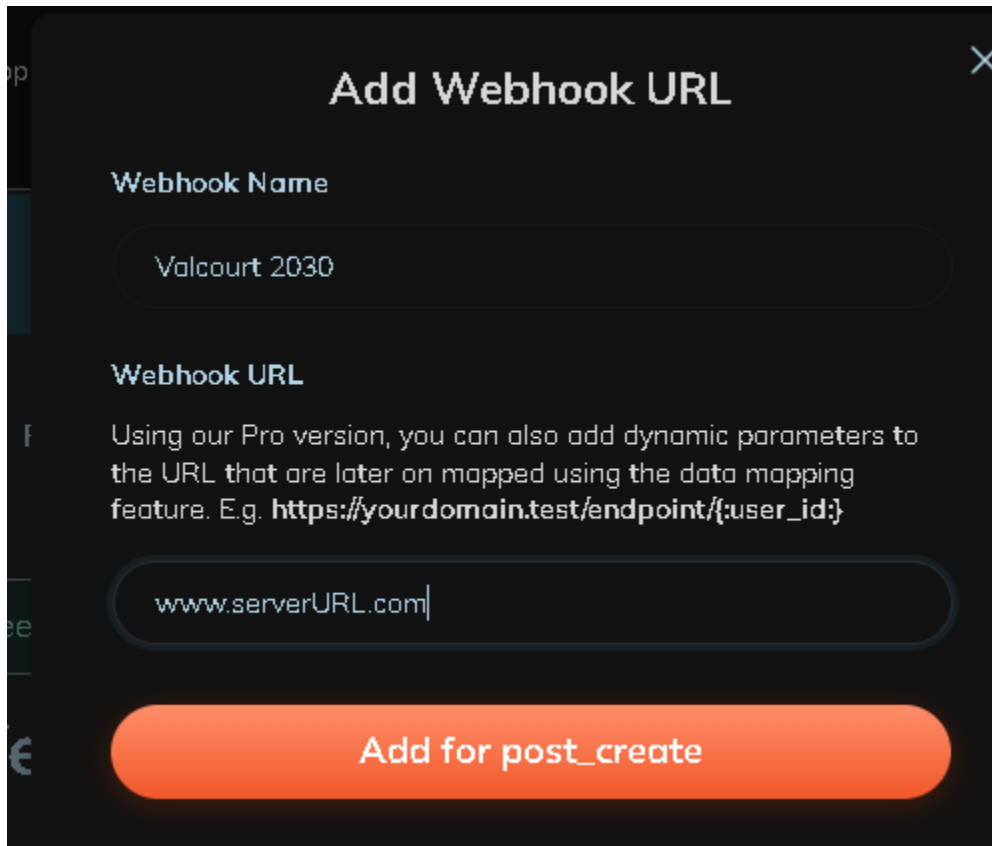
post\_create

This webhook fires after a new post was created.

Add Webhook URL

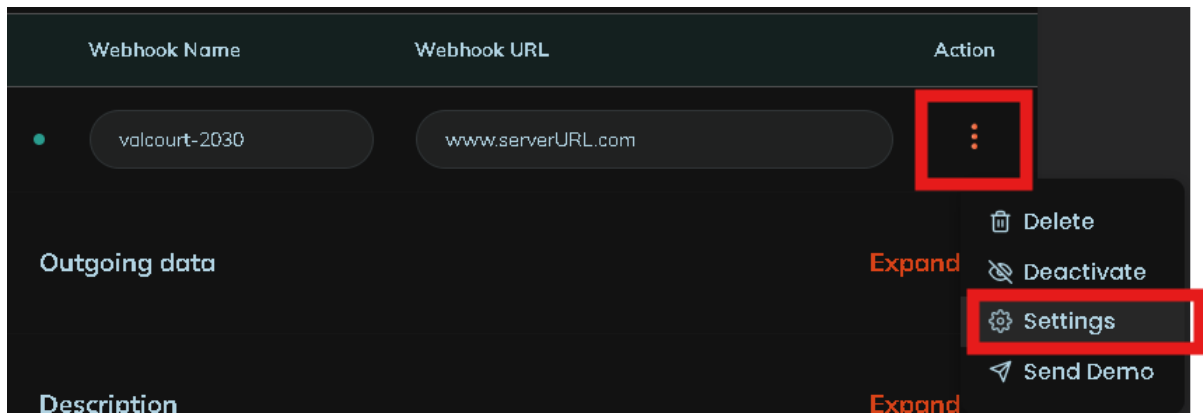
Webhook Name	Webhook URL	Action
Outgoing data		Expand ▼
Description		Expand ▼

- 4) Enter the webhook name and the URL of your server and click on the Add for post\_create button.



The image shows a dark-themed dialog box titled "Add Webhook URL" with a close button (X) in the top right corner. It contains two input fields: "Webhook Name" with the text "Valcourt 2030" and "Webhook URL" with the text "www.serverURL.com". Below the URL field is a large orange button labeled "Add for post\_create". A text block above the URL field explains that dynamic parameters can be added to the URL, with an example: "https://yourdomain.test/endpoint/{:user\_id:}".

- 5) Click on the settings option

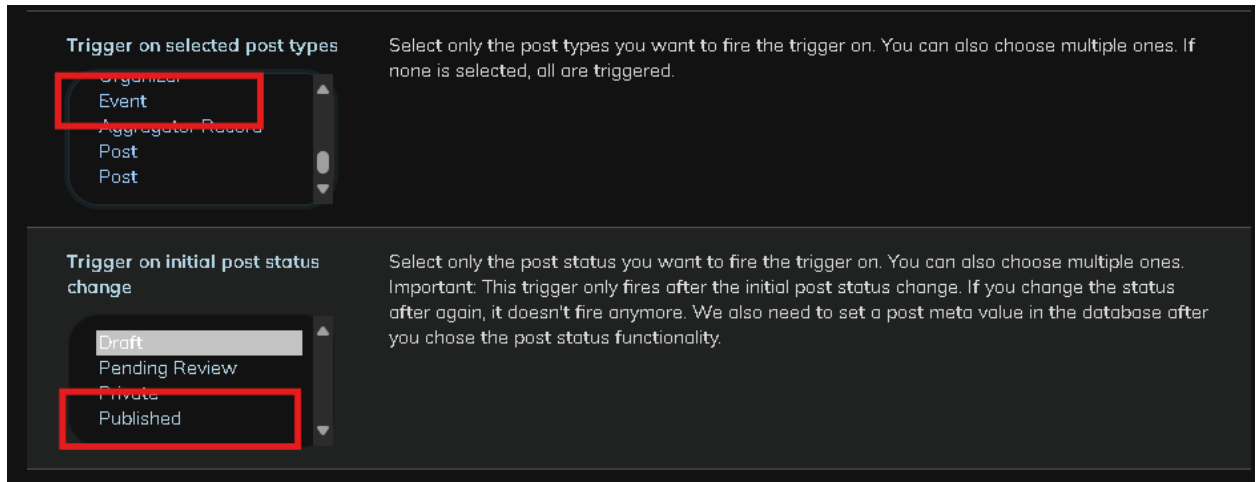


The image shows a table with three columns: "Webhook Name", "Webhook URL", and "Action". The first row contains "valcourt-2030" and "www.serverURL.com". A red box highlights the three-dot menu icon in the "Action" column. A dropdown menu is open, showing options: "Delete", "Deactivate", "Settings" (highlighted with a red box), and "Send Demo". Below the table, there are sections for "Outgoing data" and "Description", each with an "Expand" link.

Webhook Name	Webhook URL	Action
valcourt-2030	www.serverURL.com	⋮

- Delete
- Deactivate
- Settings
- Send Demo

- 6) Change the following options. On Trigger on selected post types select EVENT and on Trigger on initial post status change select PUBLISHED, scroll down and click on SAVE and your Webhook should be good to go.

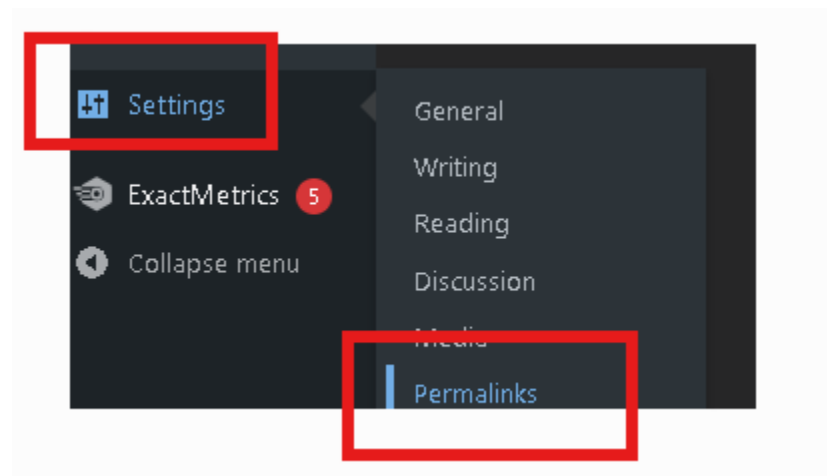


Now every new Event will be redirected to the Database and will be displayed on the Website.

## Configuring tags in WordPress

To make sure that the tags work on the WordPress website when clicking them on the events. (To see all the events that share the similar tag) You will need to twitch some WordPReSS settings so they are displayed correctly.

To do so you will need to go to your Wordpress Permalink option on your settings and set them up as Plain:



## Common Settings

Select the permalink structure for your website including the `%postname%` tag makes links easy to understand, and can help

### Permalink structure

☒ Plain

`http://54.161.93.11/?p=123`

☐ Day and name

`http://54.161.93.11/2024/05/15/sample-post/`

☐ Month and name

`http://54.161.93.11/2024/05/sample-post/`

Afterwards whenever you will click on the tags on a post it will show all the post with those tags.

## DETAILS

Date:

June 22

Time:

8:00 am – 7:00 pm

Event Tags:

[Arts](#), [Concertation et partenariats](#), [Environnement](#), [Festival](#), [Interculturel](#), [Rencontre sociale](#), [Social Inclusion](#)



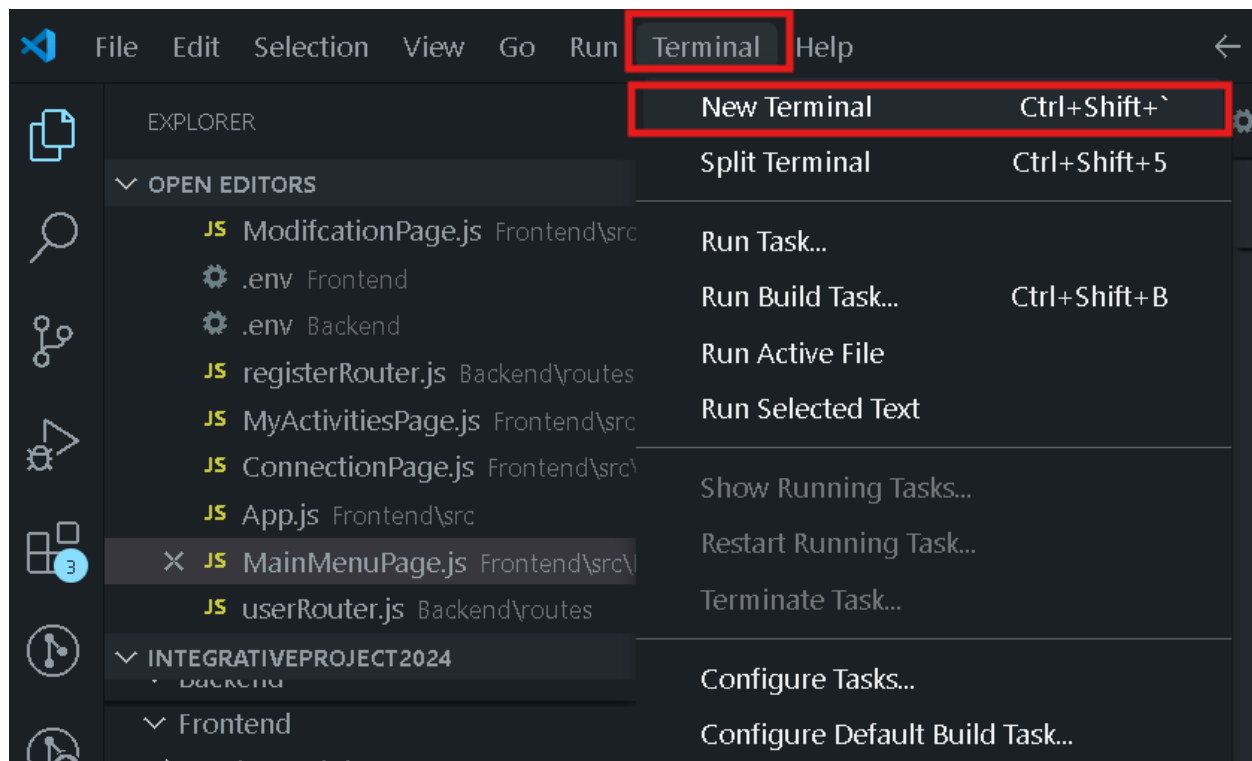
## Deploying a React Application to AWS Elastic Beanstalk

This documentation outlines the steps to deploy a React application to AWS Elastic Beanstalk using the Elastic Beanstalk Command-Line Interface (CLI).

### Deployment Process:

*Create a zipped version of the build of the Valcour2030 application.*

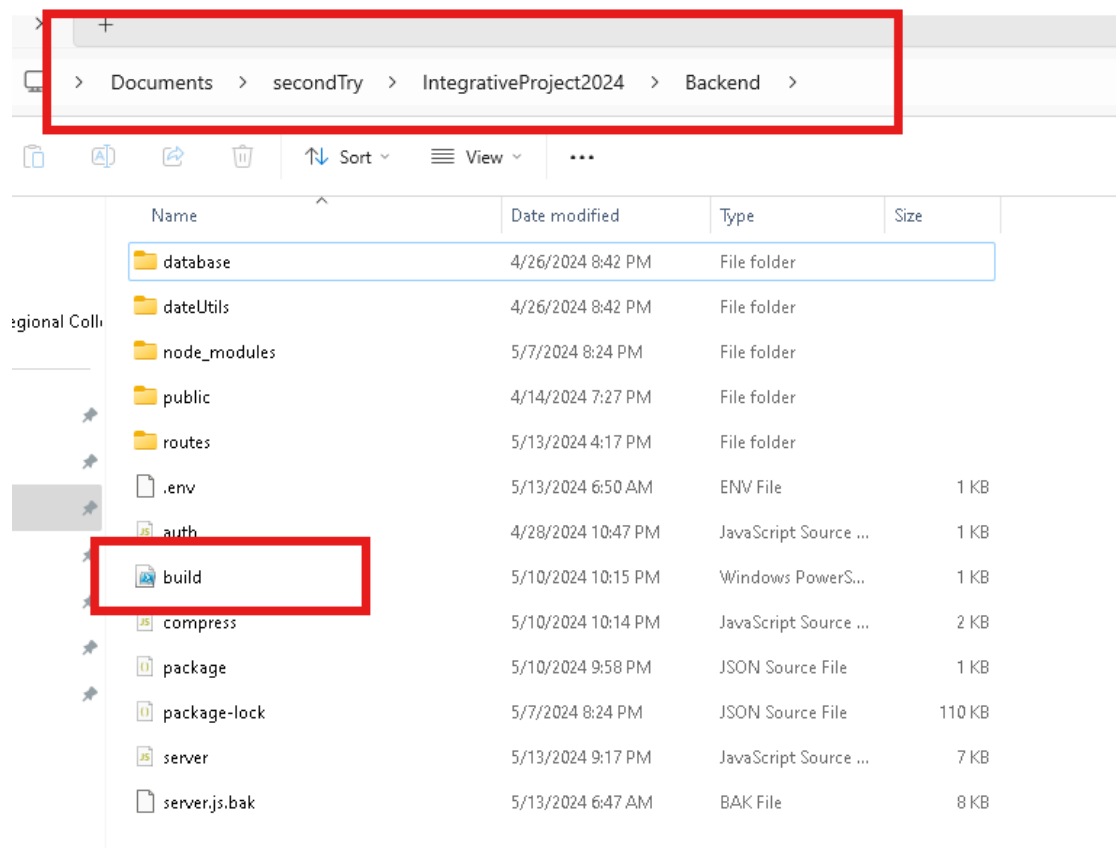
- To do so the person in charge of deploying would have to pull the last version of the application from the GitRepo:
  - <https://github.com/VladimirEstevez/IntegrativeProject2024.git>
- After they will need to install Node.js
  - [Node.js — Run JavaScript Everywhere \(nodejs.org\)](https://nodejs.org/)
- They will need to install Visual Studio Code and open the root folder of the project by doing Open Folder and selecting the folder that was cloned from the main repo.
- Afterwards, they will need to open a Terminal window in the project's root folder by clicking on Terminal, New Terminal.

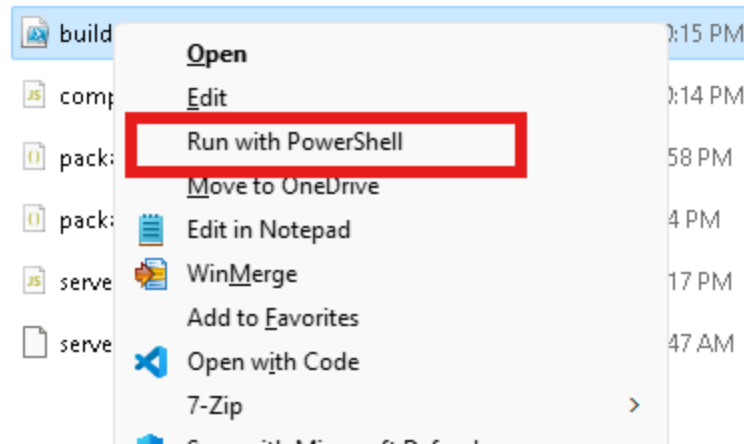


- Next, click on the Split Terminal option pointed by the red arrow

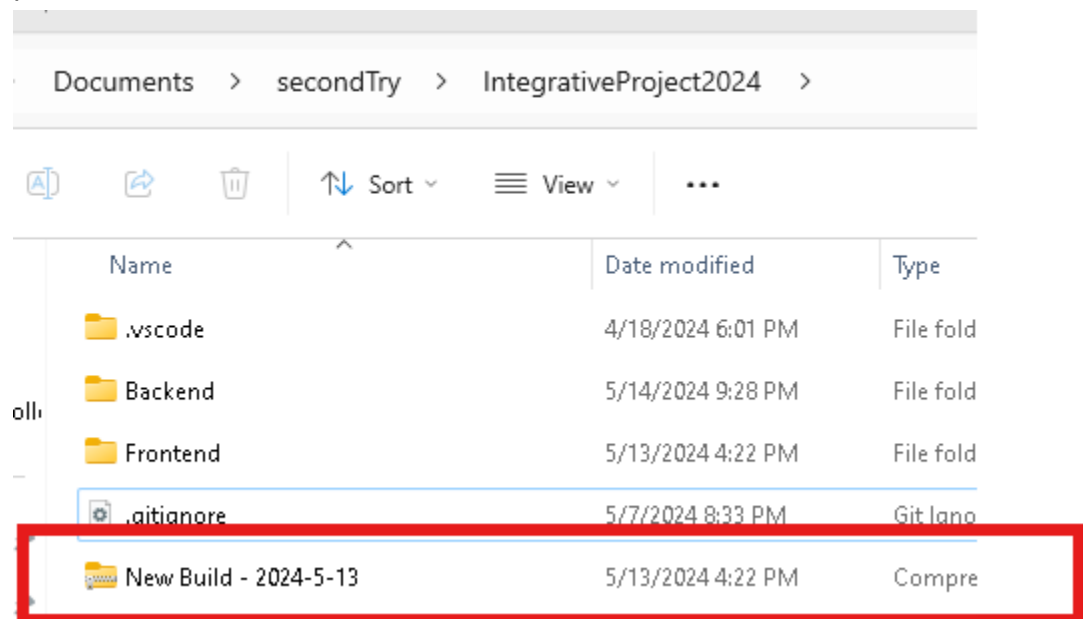


- Then tap the following:
  - On one of the tabs:
    - ***cd Frontend ; npm i***
  - And on the other one
    - ***cd Backend ; npm i***
- These commands should install all the dependencies needed to the application to be able to be build.
- Finally you will need to run the build to do so you could run the ***npm run bundle*** command on the Backend terminal or:
  - Go to the root folder of the project and press on the Shift + right click on the build script and select Run with PowerShell:





- This command will perform the same as the npm run bundle. It will generate a Zip file next to the Backend and Frontend folder with the date the build was performed.

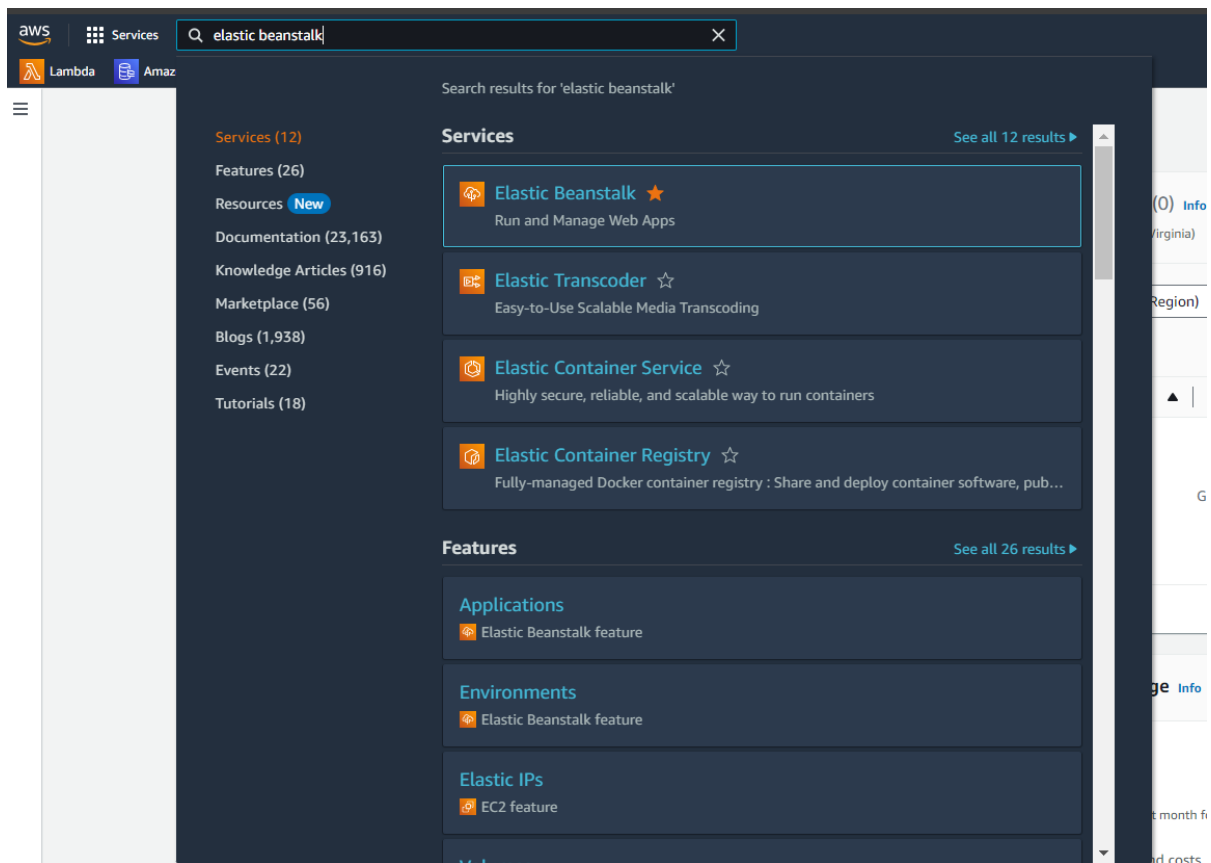


- This will be the Zip file we will be later uploading to the Elastic Beanstalk instance.

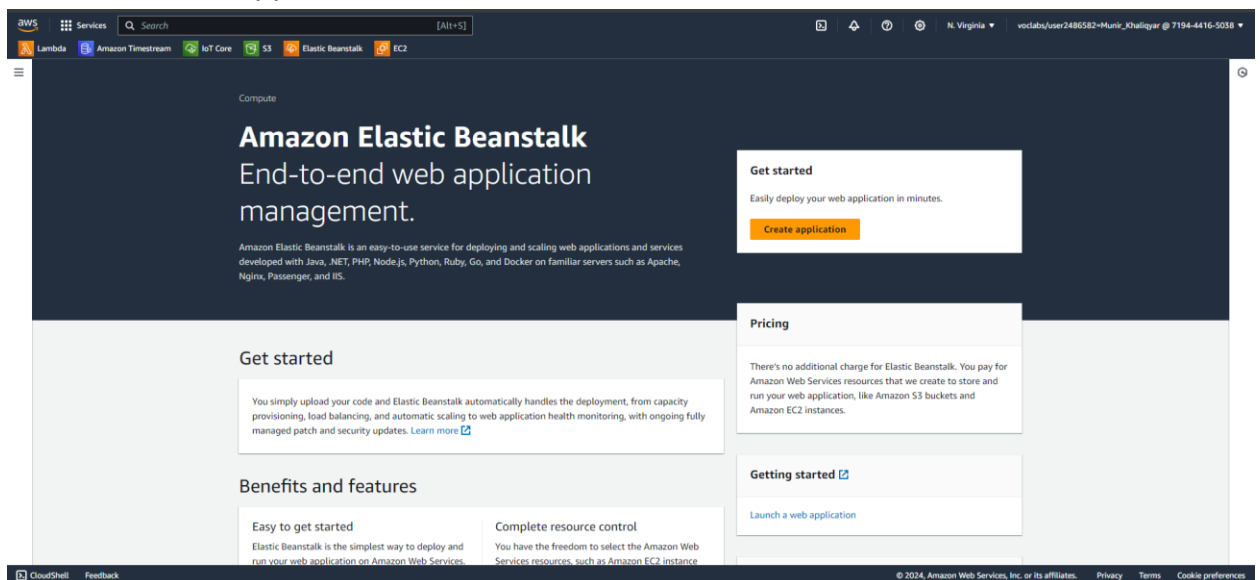
#### *Initialization and environment creation:*

You need an application environment for your application to be deployed and hosted in AWS and you can use the “Elastic Beanstalk” as your server.

1. Sign in to your AWS account and look for “Elastic Beanstalk” in the console:



2. Click on “Create application”:



3. Put a name for your application, for example “IntegrativeProjectEnvironment” and choose environment name that AWS suggests:

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar lists the steps: Step 1: Configure environment (selected), Step 2: Configure service access, Step 3 - optional: Set up networking, database, and tags, Step 4 - optional: Configure instance traffic and scaling, Step 5 - optional: Configure updates, monitoring, and logging, Step 6: Review. The main content area is titled 'Configure environment' and includes an 'Info' link. It contains three sections: 'Environment tier' with radio buttons for 'Web server environment' (selected) and 'Worker environment'; 'Application information' with a text field for 'Application name' containing 'IntegrativeProjectEnvironment' and a note about the maximum length of 100 characters; and 'Environment information' with a text field for 'Environment name' containing 'IntegrativeProjectEnvironment-env' and a note about the name's length and uniqueness.

4. In the Platform section choose “Node.js” as Platform, choose “Node.js 20 running on 64bit Amazon Linux 2023” as Platform branch and choose “6.1.3 (Recommended)” as Platform version. Choose “Sample application” for Application code.

The screenshot shows the 'Platform' section of the AWS Elastic Beanstalk console. It includes a 'Platform type' section with radio buttons for 'Managed platform' (selected) and 'Custom platform'. Below this are three dropdown menus: 'Platform' set to 'Node.js', 'Platform branch' set to 'Node.js 20 running on 64bit Amazon Linux 2023', and 'Platform version' set to '6.1.3 (Recommended)'. The 'Application code' section has radio buttons for 'Sample application' (selected), 'Existing version', and 'Upload your code'. The 'Presets' section is partially visible at the bottom.

5. For the “Service access” choose “LabRole” for the Existing service roles, choose “vockey” for EC2 key pair, and “LabInstanceProfile” for EC2 instance profile then

click on “Skip to review” and then click on “Submit” to create the application environment.

The screenshot shows the 'Configure service access' step in the AWS Elastic Beanstalk console. The left sidebar lists steps from 1 to 6, with 'Configure service access' (Step 2) selected. The main panel is titled 'Configure service access' and contains the following sections:

- Service access:** A note explaining that IAM roles are assumed by Elastic Beanstalk. Below this, the 'Service role' is set to 'Use an existing service role' with a dropdown menu showing 'LabRole'.
- EC2 key pair:** A dropdown menu showing 'vockey'.
- EC2 instance profile:** A dropdown menu showing 'LabInstanceProfile'.

At the bottom of the panel, there are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

- When the environment is created, you need to add some environment variables. Go to Configuration, scroll down and find “Updates, monitoring, and logging” then click on “Edit”.

The screenshot shows the 'Updates, monitoring, and logging' configuration page in the AWS Elastic Beanstalk console. The left sidebar shows the 'Configuration' tab selected. The main panel is titled 'Updates, monitoring, and logging' and contains the following sections:

- Monitoring:** A table with columns for 'System', 'Cloudwatch custom metrics - instance', and 'Cloudwatch custom metrics - environment'. The 'System' column is set to 'enhanced', 'Log streaming' is 'Deactivated', and 'Managed updates' is 'Activated'.
- Updates:** A table with columns for 'Managed updates', 'Command timeout', and 'Instance replacement'. The 'Managed updates' column is set to 'Activated', 'Command timeout' is '600', and 'Instance replacement' is 'Ignore health check'.

At the bottom right of the panel, there is an 'Edit' button.

- Add these variables:

DATABASE\_URL=

SECRET\_TOKEN=

EMAIL\_PASSWORD=

PORT=8080

RECIPIENT\_EMAIL=”Valcourt2030@outlook.com”

REACT\_APP\_SERVER\_URL=

EMAIL\_PROVIDER="outlook"

SERVER\_URL=

The "SECRET\_TOKEN" is random numbers and letters. I recommend making it at least 10 characters including letters and numbers.

To be able to retrieve the EMAIL\_PASSWORD Application password for [valcourt2030@outlook.com](mailto:valcourt2030@outlook.com) follow these instructions:

[Microsoft two-step verification - Microsoft Support](#)

- a. Go to the <https://account.microsoft.com/security> basics page and sign in to your Microsoft account.
- b. Select more security options.
- c. Under Two-step verification, choose Set up two-step verification to turn it on. If you see Turn off two-step verification, it's already turned on.
- d. Once you've verified two-step verification is turned on, select Create a new app password under App passwords.
- e. A new app password is generated and appears on your screen. Copy this password (without the spaces) and enter this password when Outlook prompts you for a password.

### *Creating the MongoDB Database*

The DATABASE\_URL variable will be provided to you when you create a MongoDB database. Here are the instructions on how to create a MongoDB database:

## **Prerequisites**

- A MongoDB Atlas account. If you don't have one, you can sign up at [mongodb.com/cloud/atlas](https://mongodb.com/cloud/atlas).
- Basic understanding of database concepts.
- MongoDB Compass installed on your computer. You can download it from the [MongoDB Compass download page](#).

## **Step-by-Step Guide**

### **Step 1: Log In to MongoDB Atlas**

- Visit [MongoDB Atlas](#).

- Click on the **Sign In** button and enter your credentials to log in to your account.

## Step 2: Create a New Project

- After logging in, you will be directed to the Atlas dashboard.
- Click on **New Project**.
- Enter a name for your project and click **Next**.
- Optionally, add members to your project by entering their email addresses and selecting their roles. Click **Create Project**.

## Step 3: Build a Cluster

- In your new project, click on **Build a Cluster**.
- At the top, select the **Shared** option to access the shared clusters.
- Choose the **M0 Sandbox** (Free Tier) to get started without any cost. Note that the M0 Sandbox has a storage limit of up to 512 MB.
- Choose **AWS** as your cloud provider.
- Select **N. Virginia (us-east-1)** as the region. This region is geographically close to many users in the United States and offers good performance.
- Click **Create Cluster**. The cluster creation process will take a few minutes.

## Step 4: Configure Cluster Security

- Once your cluster is created, you need to configure the security settings.
- Click on **Database Access** in the left-hand menu.
- Click **Add New Database User**.
- Enter a username and password for your database user. **Take note of the username and password** because you will not be able to retrieve the password after it is created. Make sure to save these credentials securely as you will need them to connect to your cluster.
- Assign the user appropriate roles based on your needs (e.g., read Write).
- Click **Add User**.

Next, configure the IP Whitelist to allow connections to your cluster.

- Click on **Network Access** in the left-hand menu.
- Click **Add IP Address**.
- You can allow access from anywhere by adding 0.0.0.0/0, or specify your IP address or CIDR range for more restricted access.
- Click **Confirm**.



## Step 5: Connect to Your Cluster

- Go back to the **Clusters** view.
- Click **Connect** next to your cluster.
- Choose your connection method. You can connect using MongoDB's native drivers, MongoDB Compass, or via the command line.
- To connect using MongoDB Compass:
  - If you haven't installed MongoDB Compass, download and install it from the [MongoDB Compass download page](#).
  - Open MongoDB Compass.
  - In the Atlas UI, click on **Connect using MongoDB Compass**.
  - Select your version of MongoDB Compass and copy the provided connection string.
  - Open MongoDB Compass and paste the connection string in the **New Connection** dialog.
  - Replace <username> and <password> in the connection string with the database user credentials you created earlier.
  - Click **Connect** to connect to your cluster.

## Conclusion

By following these steps, you have successfully created a MongoDB cluster on MongoDB Atlas and connected to it using MongoDB Compass or a connection string. You can now use this cluster to store, manage, and query your data efficiently. With the MongoDB Compass connection, you have direct access to view and manage your data within the database, providing a powerful interface for database operations. The connection string you used to connect to MongoDB Compass will be the one you will be using on the DATABASE\_URL= environment variable on the Elastic Beanstalk instance.

Lifecycle

Keep logs after terminating envir... ▼

### Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

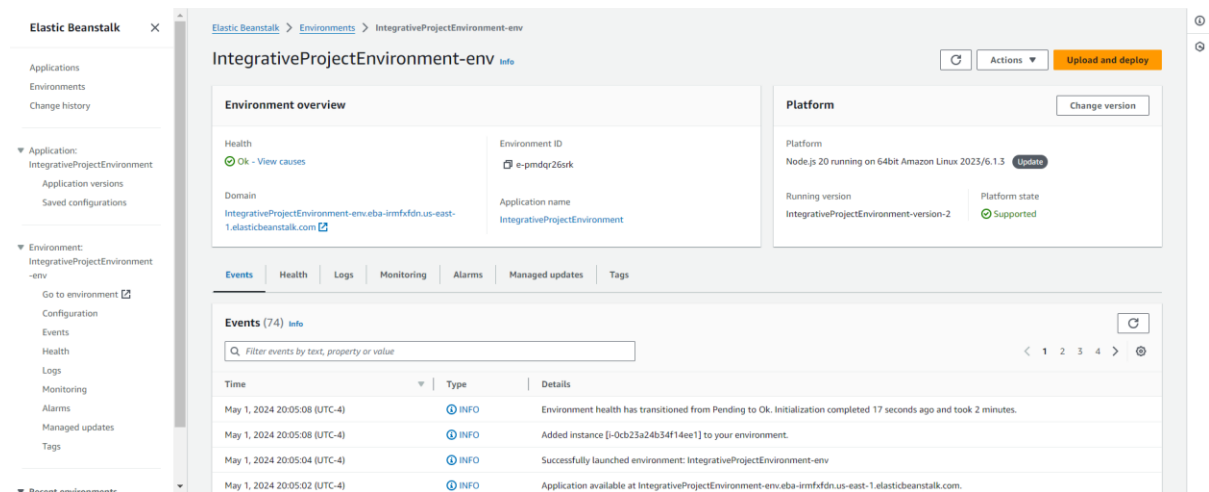
Name	Value	
<input type="text" value="DATABASE_URL"/>	<input type="text" value="mongodb+srv://host:mariadb@integrat"/>	<input type="button" value="Remove"/>
<input type="text" value="EMAIL_PASSWORD"/>	<input type="text" value="yqdw ikvw epgw nvwb"/>	<input type="button" value="Remove"/>
<input type="text" value="EMAIL_PROVIDER"/>	<input type="text" value="gmail"/>	<input type="button" value="Remove"/>
<input type="text" value="PORT"/>	<input type="text" value="8080"/>	<input type="button" value="Remove"/>
<input type="text" value="REACT_APP_SERVER_URL"/>	<input type="text" value="http://integrativeprojectenvironment-er"/>	<input type="button" value="Remove"/>
<input type="text" value="RECIPIENT_EMAIL"/>	<input type="text" value="integrativeprojectgroupthree@gmail.cor"/>	<input type="button" value="Remove"/>
<input type="text" value="SECRET_TOKEN"/>	<input type="text" value="123"/>	<input type="button" value="Remove"/>
<input type="button" value="Add environment property"/>		

Cancel

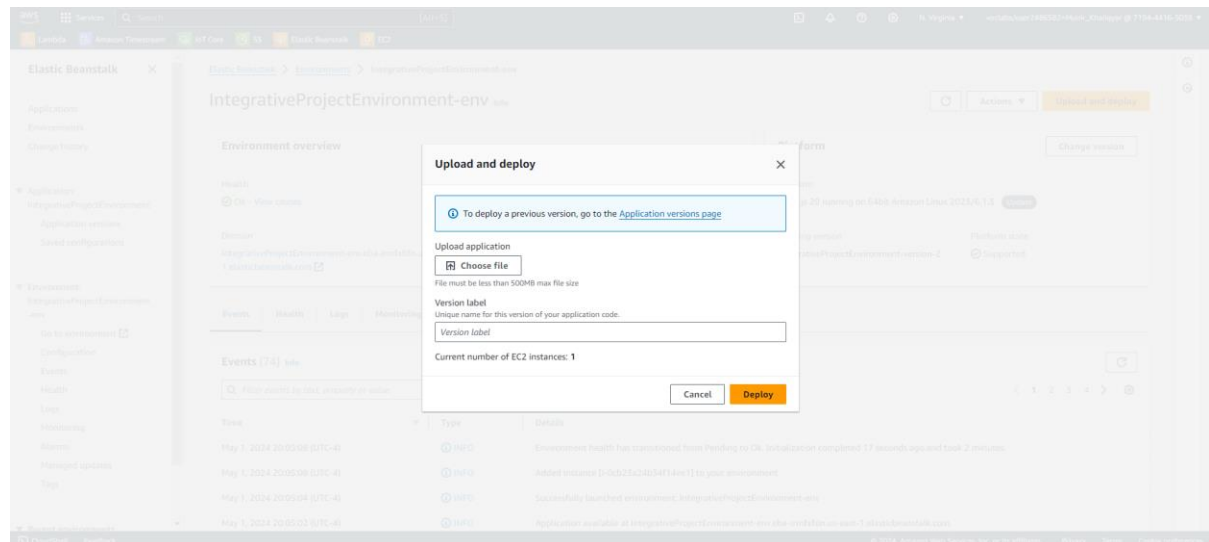
After filling in all the values of the different variables click on the Apply button to confirm the changes.

- When the changes are done you should see the status of the “Health” is “Ok” then you can upload the zipped folder of the application that we created before (see previous step Create a zipped version of the build of the Valcour2030 application )

by clicking on “Upload and deploy”.



9. Then you can choose the zipped folder from your computer and add a name to the version of the application in the “Version label”.



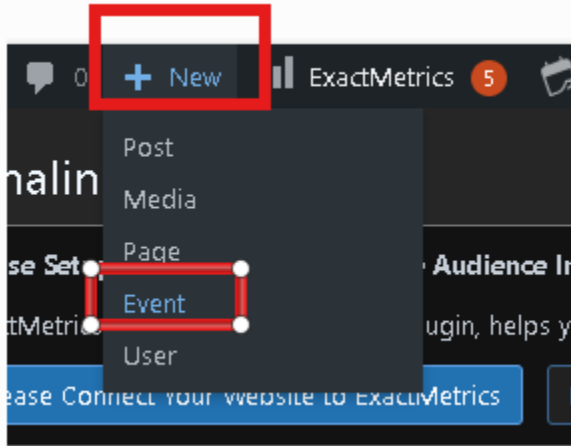
### Accessing the Application:

Upon successful deployment, access the deployed React application using the provided URL or DNS NAME.

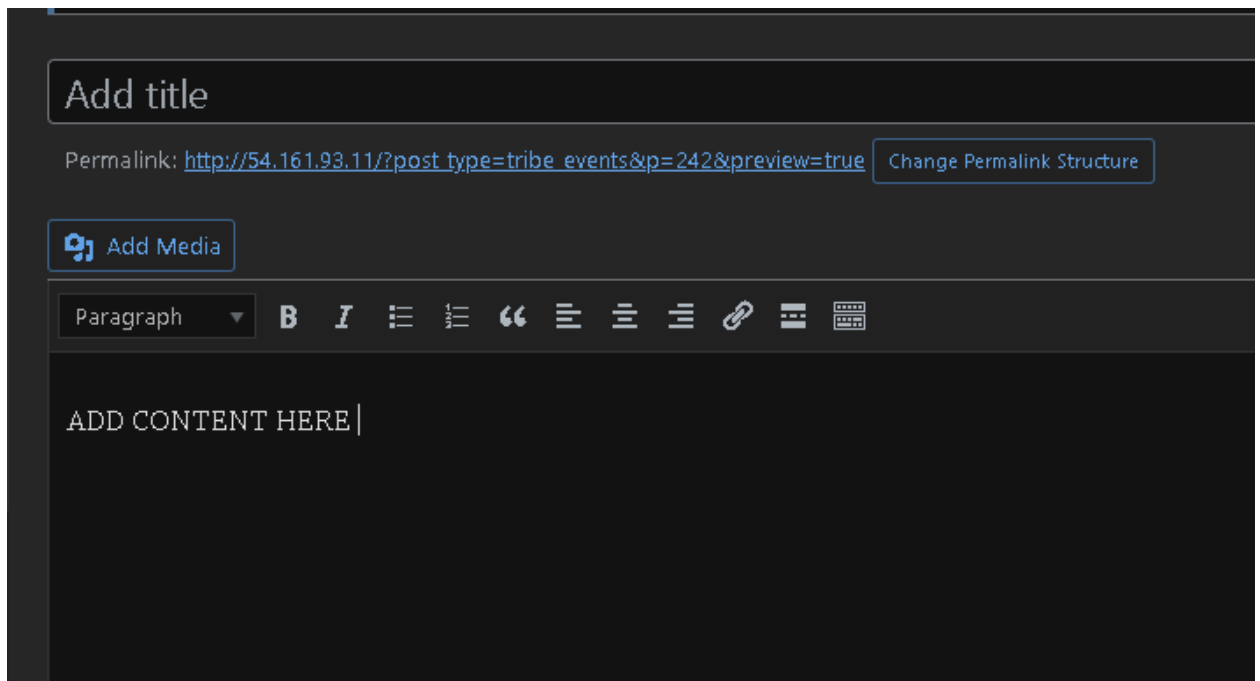
Navigate to the URL in a web browser to view the deployed application.

## Creating a new Event on the WordPress Website

- Click on New and Event

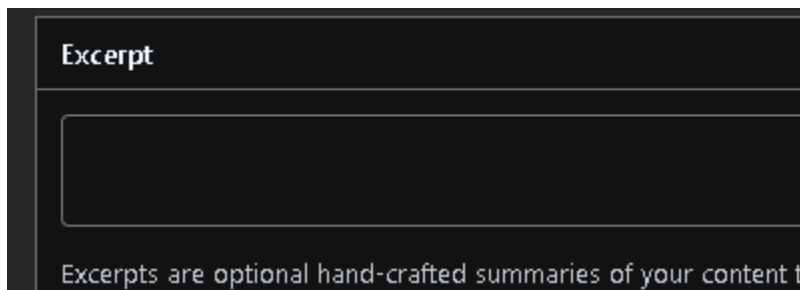


- You will add the title and content full description on the two following fields:



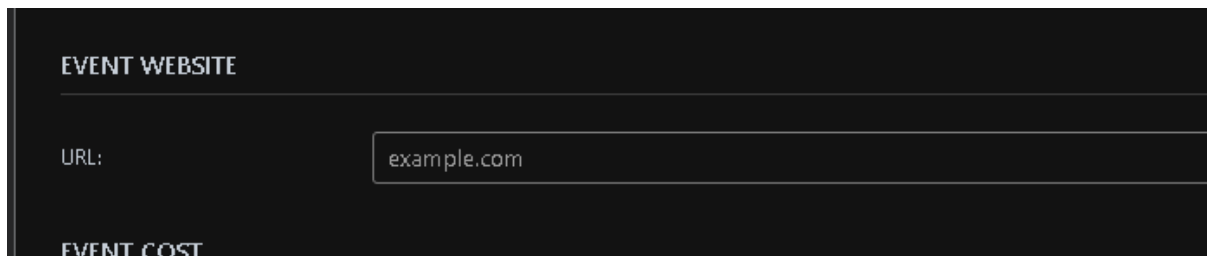
- You will add a short description of the event on the Excerpt field, this will be displayed next to the thumbnail where all the activities are shown but will not be

part of the Main Activity view.



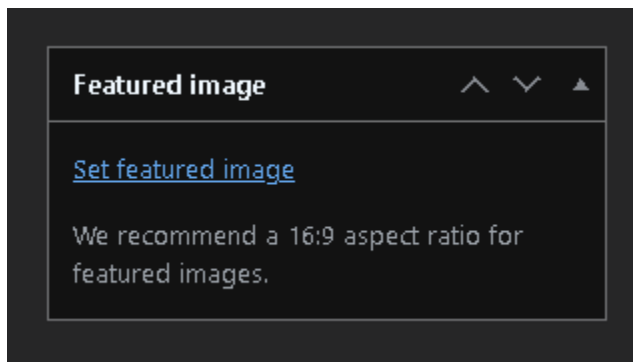
The screenshot shows a dark-themed form with a section titled "Excerpt". Below the title is a large, empty rectangular text input area. At the bottom of the section, there is a small line of text: "Excerpts are optional hand-crafted summaries of your content t".

- On this part you will add the form that the user will need to fill out to be able to register to the activity :



The screenshot shows a dark-themed form with two sections. The first section is titled "EVENT WEBSITE" and contains a label "URL:" followed by a text input field containing the value "example.com". The second section is titled "EVENT COST" and is partially visible at the bottom.

- You will add the image that will be shown as thumbnail and displayed in the following part of the site



The screenshot shows a dark-themed form with a section titled "Featured image". To the right of the title are three small icons: an upward arrow, a downward arrow, and a triangle. Below the title is a link that says "Set featured image". Underneath the link, there is a line of text: "We recommend a 16:9 aspect ratio for featured images."

- And on this section you will add the different tags that the activity will be having

Tags

Add

Separate tags with commas

Choose from the most used tags

- Arts cars
- Concertation et partenariats
- Cuisine Environnement
- Festival Friends FUN
- Guillaume IMAGINATION
- Interculturel Money Munir
- music Musique parade
- Rencontre sociale
- Social Inclusion
- Sports Vladimir
- Éducation

- After filling in the date and venue, you can publish the event, and it will be added to the database and website if the application has been correctly deployed.

