**IWC Fall 2024 EPICS Installation Guides and Application Information**

**Table of Contents**

1. Installing Npm and Node.js
2. Installing Git
3. AWS
4. EC2
5. RDS
6. Postgresql
7. React
8. S3

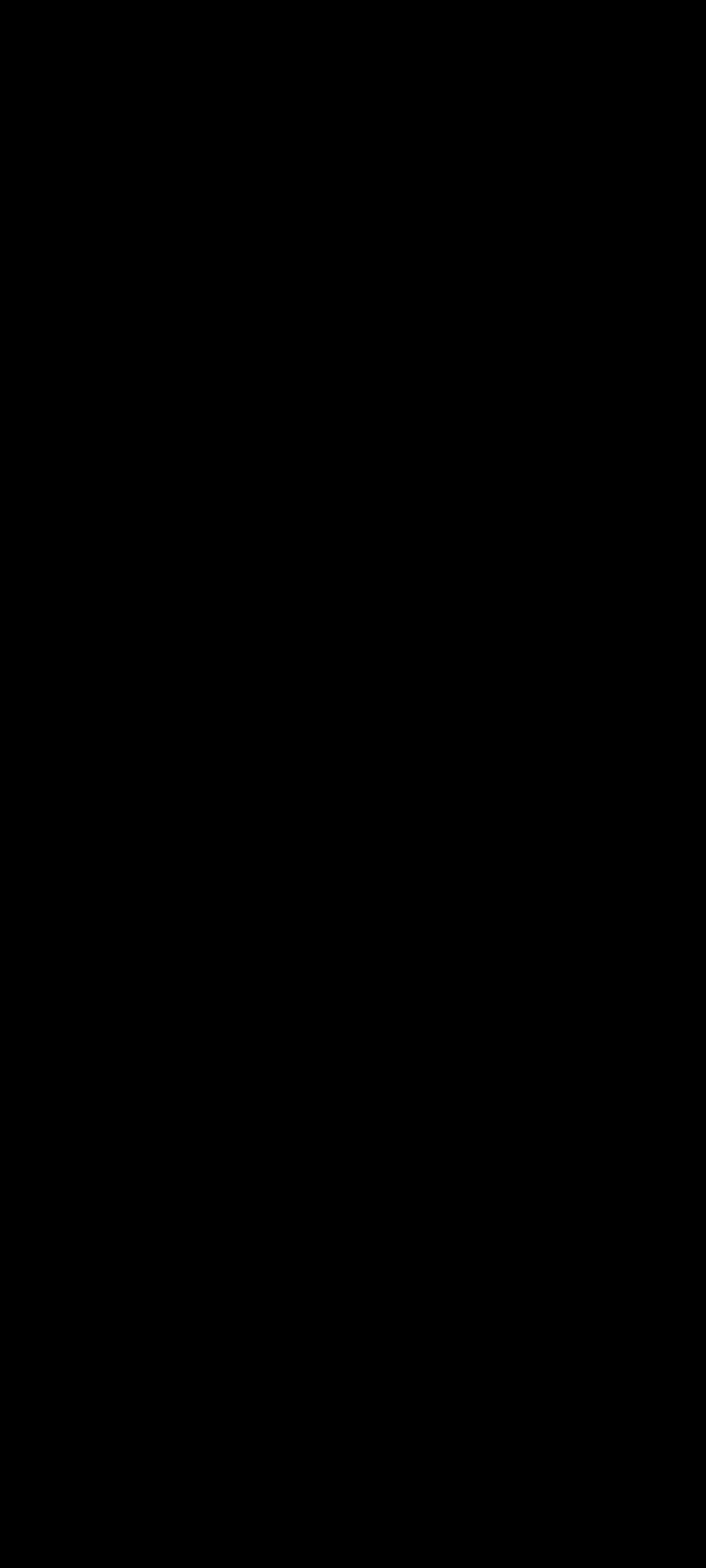
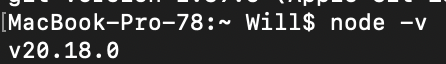
**Node.js and NPM**

**What are node.js and npm?**

* Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser
  + It is required for building applications
  + [More Information](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)
* NPM, (AKA Node Package Manager) is a package manager used for javascript applications
  + [More Information](https://www.npmjs.com/)

**Installing node.js and npm On a Mac Device**

**WARNING**: For some parts of the installation process, you may need to put the word “sudo” before the command you are trying to run. This will prompt you to enter your computer’s username and password, which is needed to run commands as an administrator.

* 1. Installing node.js and npm
     1. To install node.js use the following link: [node\_js\_install](https://nodejs.org/en/download/prebuilt-installer)
     2. The Mac user in our group used the prebuilt installer, which also included npm
     3. To make sure you have npm and node.js installed, run the commands “node -v” and “npm-v” to make sure you have the correct versions
     4. 

**Installing Node.js and npm on Windows**

1. Install Node.js and npm
   1. <https://nodejs.org/en/download/prebuilt-installer>
2. Follow the same Mac instructions to make sure everything is installed (not using the sudo command)

**Installing Git**

**What is Git?**

* Git is used for source code management. Git is important because it is able to track changes in the code, making it vital for projects that multiple people are working on.

**Installing Git**

* On Mac:
  + Just by running ‘git’ in your terminal command line, it should prompt you to install git.
  + If not, [visit this website](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) for the installer
  + Type ‘git’ into your command line to make sure it’s installed
* On Windows
  + For Windows, it’s recommended to install Git via the Chocolatey package
  + Visit the “installing on windows” section of [this website](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) to find the Chocolatey installer



**AWS**

**What is AWS?**

* <https://aws.amazon.com/about-aws/?nc2=h_header>
* AWS, better known as Amazon Web services
* We used AWS because it is a cloud hosting service that we can use for our application.
* For AWS, we used AWS Aurora
* Gave us free access to EC2 and RDS, S3 for 12 months

**Aurora**

* AWS Aurora is a database hosting service. Aurora is compatible with MYSQL and Postgresql, and is faster than both. In the future, the database will hopefully be hosted on AWS Aurora



**EC2**

**What is EC2?**

* Also known as Elastic Compute Cloud, EC2 is a service provided by AWS that allows us to create a “virtual computer” in the cloud. You can use EC2 to be a sort of “server” for your applications
* For more information: <https://aws.amazon.com/ec2/>

**Why did we use EC2?**

* We need a virtual machine in the cloud to host pretty much everything you make. Since EC2 was provided to us, we used it for our application

**How did we use EC2?**

* EC2 helps host our database and application code



**RDS**

**What is RDS?**

* RDS stands for relational database service, and it supports many different SQL flavors
* RDS can help with database management and offers various services for your database.
* For more information: <https://aws.amazon.com/rds/>

**Why did we use RDS?**

* Whenever you use a relational database, you’ll need a database service to help with it, and since RDS was also provided to us by AWS, we used it for our own database.



**Postgresql**

**What is Postgresql?**

* Postgresql is like MySQL but with a slightly different syntax. It’s also open source, so by extension, it’s compatible with the greatest amount of other tools and libraries (even ones that aren’t open source)
* Allows the project to run on localhost
* <https://www.postgresql.org/about/>

**Installing Postgresql (For Running Locally)**

* Download PostgreSQL from website
* Run wizard
* Use credentials from IWC/iwc-backend/.env
* Use port 5432
* Do not check the stack installer box
* Complete installation

**How did we use it?**

* We could make our database with Postgresql on our computers as opposed to using Aurora, which is only available with an RDS. We didn’t want to use Aurora right away, so we wanted the basics of the database built before we moved to RDS.



**React**

**What is React?**

* React is a javascript library, used for building user interfaces. Allows for easy building of front end for websites
* <https://legacy.reactjs.org/>

**How we used it**

* React was used to build our actual front-end website and communicate with the Postgresql database through an API
  + API stands for application programming interface. It’s a set of rules that allow for different software applications to communicate. Think of it as a filter that translates back-end input to front-end input and vice versa.
  + <https://www.mulesoft.com/api/what-is-an-api>



**S3**

**What is S3?**

* S3 is a simple storage for files. AWS provides it
* <https://aws.amazon.com/s3/>

**How we used it**

* We used S3 to store images and language audio. It’s better than just an entire file system, as we can simply store individual files

