



An **easier** way to deploy website

Meet your new WordPress assistant

User Manual

Table of Contents

1. Introuction	4
1.1 What is XMOPS Accelerate?	4
1.2. About the Documentation	4
1.2.1. Purpose	4
1.2.2. What's Included	4
1.2.3. Target Audience	5
1.2.4. Conventions Used	5
1.2.5. Feedback and Contributions	6
2. Prerequisites	7
2.1. Directory Structure	7
3. Essential Tools	8
3.1. Visual Studio Code (VSCode)	8
3.1.1. What It Is	8
3.1.2. Why You Need It	8
3.1.3. How to Install	8
3.2. Windows Subsystem for Linux (WSL)	8
3.2.1. What It Is	8
3.2.2. Why You Need It	8
3.2.3. How to Install	9
4. Development & Deployment Tools	10
4.1. Git	10
4.1.1. Installation	10
4.1.2. Configuration	10
4.1.3. Official Website	10
4.2. Node.js	10
4.2.1. Installation	10
4.2.2. Official Website	11
4.3. AWS CLI	11
4.3.1. Step 1: Downloading the AWS CLI	11
4.3.2. Step 2: Unzip the Installer	12
4.3.3. Step 3: Running the Installation	12
4.3.4. Step 4: Verification	12
4.3.5. Step 5: Initial Configuration	13

4.4. Terraform	13
4.4.1. Installation	13
4.4.2. Official Website	14
5. Setting up the Project	15
5.1. Cloning the XMOPS Accelerate Project	15
5.2. Setting up Development Environment	16
5.2.1. Initializing React in the Project	16
5.2.1.1. Verify Node.js and npm Installation	16
5.2.1.2. Navigate to Your Project Directory	16
5.2.1.3. Install React Dependencies	16
5.2.2. Integrating AWS Amplify	17
5.2.2.1. Install the AWS Amplify CLI	17
5.2.2.2. Configure AWS Amplify	17
5.2.2.3. Initialize Amplify in Your Project	17
5.2.2.4. Add Authentication with Amazon Cognito	18
5.2.2.5. Integrate Authentication into Your Application	18
5.3. Running the Project	19
6. Final Words	20
7. Contributors	21

1. Introduction

1.1 What is XMOPS Accelerate?

XMOPS Accelerate is a Software as a Service (SaaS) platform designed to streamline and simplify the deployment process of web applications (specifically WordPress). It caters to a variety of deployment architectures, including Monolith, Highly Available, and LightSail, providing a user-friendly interface that facilitates quick and efficient application deployment.

1.2 About the Documentation

Welcome to the comprehensive guide for setting up and deploying projects with XMOPS Accelerate. This documentation is meticulously crafted to assist developers at all levels, from beginners to experienced professionals, in navigating through the setup, development, and deployment phases of their web applications using XMOPS Accelerate. Whether you are looking to streamline your development process, integrate cloud services seamlessly, or ensure your applications are deployed efficiently, this guide provides all the necessary instructions, tips, and best practices.

1.2.1 Purpose

The purpose of this documentation is to:

Empower Users: Provide clear, step-by-step instructions to help users set up their development environment, clone and initiate the XMOPS Accelerate project, and leverage essential tools and services.

Simplify Complexity: Break down complex processes into easy-to-follow steps, ensuring users can successfully implement the XMOPS Accelerate platform without prior expertise in its constituent technologies.

Promote Best Practices: Share industry-standard practices for software development, deployment, and cloud integration, enhancing the quality and security of user projects.

Encourage Self-Sufficiency: Equip users with the knowledge to troubleshoot common issues, make informed decisions about their deployment strategies, and customize their setup as per their project requirements.

1.2.2 What's included

This documentation covers:

Essential Tools: Guidance on installing foundational tools such as Git, Visual Studio Code (VSCode), and the Windows Subsystem for Linux (WSL), setting the stage for a productive development environment.

Development and Deployment Tools: Detailed instructions for installing and configuring key development and deployment tools, including Docker, Node.js, the AWS CLI, and Terraform, ensuring users can build, containerize, and deploy applications smoothly.

Project Setup: Step-by-step process to clone the XMOPS Accelerate project from GitHub, navigate its structure, and prepare it for further development and customization.

Cloud Integration: A walkthrough on creating and configuring AWS Cognito User and Identity Pools, installing AWS Amplify, and integrating these services with the project, enabling authentication and authorization features along with cloud resource management.

1.2.3 Target Audience

This documentation is designed for:

- Developers looking to deploy web applications using XMOPS Accelerate.
- Teams adopting cloud-native development and deployment practices.
- Organizations aiming to streamline their development pipeline with modern tools and services.

1.2.4 Conventions Used

Throughout this documentation, we use the following conventions to ensure clarity and consistency:

Commands: All terminal commands are presented in a code block, indicating exact input required in your terminal or command prompt.

Notes: Additional information, tips, or best practices are highlighted to draw attention to important details.

Warnings: Potential issues, common pitfalls, and crucial considerations are clearly marked to help you avoid common mistakes.

1.2.5 Feedback and Contributions

Your feedback is invaluable to us. If you encounter any issues, have suggestions for improvements, or wish to contribute to this documentation, please reach out through our project repository or support channels. We are committed to continuously improving our documentation to meet your needs and enhance your experience with XMOPS Accelerate.

2. Prerequisites

Before diving into XMOPS Accelerate, ensure your workstation is equipped with the following tools and services. These prerequisites are essential for a seamless experience with XMOPS Accelerate on a Windows-based system, enabling efficient development, deployment, and management of web applications.

2.1 Directory Structure

```
xmops-accelerate/  
├─ backend/  
│   └─ (Node.js app files here)  
├─ frontend/  
│   └─ (React app files here)  
└─ terraform/  
    ├─ highlyavailable/  
    │   └─ (Terraform files for highly available architecture)  
    ├─ lightsail/  
    │   └─ (Terraform files for Lightsail configuration)  
    └─ monolith/  
        └─ (Terraform files for monolithic architecture)
```

3. Essential Tools

3.1 Visual Studio Code (VSCode)

3.1.1 What it is

VSCode is a powerful, open-source code editor developed by Microsoft for Windows, Linux, and macOS. It features built-in support for JavaScript, TypeScript, and Node.js, with a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes.

3.1.2 Why you need it

For editing code, managing Git repositories, and integrating with various development tools directly within the editor.

3.1.3 How to install

- Visit VSCode's official download page and download the installer for Windows.
- Run the installer and follow the on-screen instructions to complete the installation.

3.2 Windows Subsystem for Linux (WSL)

3.2.1 What it is

WSL allows Windows users to run a GNU/Linux environment directly on Windows, unmodified, without the overhead of a traditional virtual machine or dual-boot setup.

3.2.2 Why you need it

For running Linux-based applications and toolchains, including Terraform and AWS CLI, directly within Windows.

3.2.3 How to install

- Open Visual Studio Code (VSCode) as Administrator and run:

```
wsl --install
```

- This command installs WSL with the default Linux distribution. To install a specific distribution, first list available distributions with:

```
wsl --list --online
```

- Then, install your preferred distribution:

```
wsl --install <DistributionName>
```

4. Development & Deployment Tools

4.1 Git

4.1.1 Installation

Git is usually available in your Linux distribution's package manager. Install it by running:

```
sudo apt-get update  
sudo apt-get install git
```

This updates your package lists and installs Git.

4.1.2 Configuration

Configure your Git installation with your username and email:

```
git config --global user.name "Your Name"  
git config --global user.email "youremail@example.com"
```

4.1.3 Official Website

For more details or alternative installation methods, visit Git's official site.
www.git-scm.com

4.2 Node .js

4.2.1 Installation

```
curl -o-  
https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh  
| bash
```

This command fetches the install script from nvm's GitHub page and executes it. The script clones the nvm repository to ~/.nvm and adds the script to your shell profile so nvm is available when you open a new terminal. After installation, close and reopen your terminal, then install Node.js by running:

```
nvm install node
```

This command installs the latest version of Node.js. To install a specific version, replace node with the version number, e.g., nvm install 14.

4.2.2 Official Website

For alternative installation methods, visit Node.js's official website. www.nodejs.org

4.3 AWS CLI

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With it, you can control multiple AWS services from the command line and automate them through scripts. This guide will take you through the process of installing the AWS CLI on Linux and Windows (via WSL), including initial configuration steps.

4.3.1 Downloading the AWS CLI

- Open your Terminal.
- Use curl to download the AWS CLI version 2 installation script.

```
curl  
"https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip"
```

- This command downloads the latest version of the AWS CLI version 2 for Linux into a file named awscliv2.zip.

4.3.2 Unzip the Installer

- Before you can run the installer, you need to unzip the package you've just downloaded.
- Install the unzip utility if you don't have it:

```
sudo apt-get install unzip
```

- Unzip the downloaded file:

```
unzip awscliv2.zip
```

4.3.3 Running the Installation

- With the package unzipped, you're ready to install the AWS CLI.
- Execute the install script from the unzipped directory:

```
sudo ./aws/install
```

- This command will install the AWS CLI version 2 globally on your system, making it accessible from any terminal session.

4.3.4 Running the Installation

- After installation, you can verify the installation and check the installed version by running:

```
aws --version
```

- This command should return the version of the AWS CLI installed, along with the Python version it's running on.

4.3.5 Initial Configuration

Once installed, the AWS CLI needs to be configured with your AWS credentials and default region to interact with AWS services.

- Run the AWS configure command to set up your credentials:

```
aws configure
```

- You'll be prompted to enter:
 - AWS Access Key ID: Your access key.
 - AWS Secret Access Key: Your secret key.
 - Default region name: The AWS region (e.g., us-east-1).
 - Default output format: Preferred output format (e.g., json, yaml, text).
- Storing Credentials: The information provided is stored in plaintext in the `~/.aws/credentials` and `~/.aws/config` files. Ensure these files are properly secured.

4.4 Terraform

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With it, you can control multiple AWS services from the command line and automate them through scripts. This guide will take you through the process of installing the AWS CLI on Linux and Windows (via WSL), including initial configuration steps.

4.4.1 Installation

Visit Terraform's official website - <https://developer.hashicorp.com/terraform/install>

```
sudo apt-get update && sudo apt-get install -y gnupg
software-properties-common curl
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo
apt-key add -
sudo apt-add-repository "deb [arch=amd64]
https://apt.releases.hashicorp.com $(lsb_release -cs) main"
sudo apt-get update && sudo apt-get install terraform
```

These commands add the HashiCorp GPG key, add the HashiCorp Linux repository, and install Terraform.

4.4.2 Official Website

For more information, visit Terraform's official download page. -
<https://developer.hashicorp.com/terraform>

5. Setting up the Project

Assuming you have already installed Git, AWS CLI, Terraform, and Node.js as instructed previously, you're well-prepared to initialize and work on the XMOPS Accelerate project, specifically focusing on React setup and AWS Amplify integration. This guide will cover initializing React and configuring AWS Amplify in your existing project setup.

5.1 Cloning the XMOPS Project

- Launch Visual Studio Code on your machine
- Access the integrated terminal by using the shortcut **Ctrl + Shift + ~**. This shortcut brings up a command line interface within VSCode, allowing you to execute Git commands and more without leaving the editor.
- In the VSCode terminal, navigate to the directory where you want to clone the project using **cd path/to/directory**. Then, execute the following command to clone the XMOPS Accelerate repository:

```
git clone https://www.github.com/xmopsaccelerate/xmops.git
```

Replace path/to/directory with the actual path where you want the project to reside on your local machine.

- Once cloning is complete, open the project in VSCode by selecting **File > Open Folder** from the top menu and navigating to the cloned project directory. Alternatively, you can use the command **code xmops** if you're already in the directory where you cloned the repository.

5.2 Setting up Development Environment

- Since the XMOPS Accelerate project is already cloned to your machine, and you've installed all necessary dependencies, including Node.js, we'll focus on ensuring the React environment is correctly set up within this context.

5.2.1 Initializing React in the Project

5.2.1.1. Verify Node.js and npm Installation

Before proceeding, make sure Node.js and npm are correctly installed by running:

```
node -v  
npm -v
```

These commands should display the versions of Node.js and npm installed on your system, confirming their installation.

5.2.1.2. Navigate to Your Project Directory

Open your terminal or command line interface and change directory (**cd**) to your cloned XMOPS Accelerate project folder, where the React application code resides.

5.2.1.3. Install React Dependencies

Within your project directory, if you haven't already, run the following command to install all React dependencies listed in your **package.json** file:

```
npm install
```

This step ensures that all required React libraries and other dependencies are downloaded and ready for use in your development environment.

5.2.2 Integrating AWS Amplify

AWS Amplify will enable you to connect your React application to AWS services effortlessly.

Here's how to initialize AWS Amplify in your project:

5.2.2.1. Install the AWS Amplify CLI

If you haven't installed the AWS Amplify CLI globally, run:

```
npm install -g @aws-amplify/cli
```

This command installs the Amplify CLI globally on your system, allowing you to use it across all your projects.

5.2.2.2. Configure AWS Amplify

With AWS CLI already installed and configured with your credentials, proceed to configure Amplify by running:

```
amplify configure
```

This command opens a web browser asking you to sign in to the AWS Management Console and create an IAM user for Amplify. Follow the guided steps to complete the configuration.

5.2.2.3. Initialize Amplify in Your Project

Once AWS Amplify CLI is configured, initialize Amplify in your project by running:

```
amplify init
```

Answer the questions prompted in the terminal to specify the environment name, default editor, type of app you're building (choose javascript), framework (react), and other configuration options. This process sets up the necessary backend environment for your project in AWS.

5.2.2.4. Add Authentication with Amazon Cognito

To add a user authentication feature to your React app using Amazon Cognito, execute:

```
amplify add auth
```

Choose the default configuration or customize authentication settings as needed. After configuring, deploy the authentication service by running:

```
amplify push
```

Follow the prompts to confirm the deployment. This step will create the necessary resources in AWS and configure your project to use Amazon Cognito for authentication.

5.2.2.5. Integrate Authentication into Your Application

After deploying the authentication service, Amplify generates configuration files and resources that you can use to integrate authentication into your application.

- Install Amplify libraries in your project:

```
npm install aws-amplify @aws-amplify/ui-react
```

- Configure your application with Amplify by importing the Amplify library and loading the generated configuration in your application's entry point (e.g., *index.js* or *App.js*):

```
import Amplify from 'aws-amplify';  
import config from './aws-exports';  
Amplify.configure(config);
```

- Use Amplify UI Components to add authentication-related UI to your app. For example, to add a sign-in/sign-up form:

```
import { withAuthenticator } from
  '@aws-amplify/ui-react';

function App() {
  return (
    // Your app content
  );
}

export default withAuthenticator(App);
```

5.3. Running the Project

- With React set up and AWS Amplify integrated, you're now ready to run your project:

```
npm start
```

This command starts the React development server, typically opening your application in a web browser at **<http://localhost:3000>**.

- Change directory to backend and run the following command

```
node server.js
```

This command starts the node server, typically opening your application in a web browser at **<http://localhost:3001>**.

6. Final Words

Congratulations! You have successfully set up the XMOPS Accelerate project, integrated AWS Amplify for authentication, and learned how to run the project locally. You are now ready to dive into the development and customization of your web application.

As you progress through your development journey, remember to:

- **Regularly commit your changes:** Use Git to track your project's progress and maintain a clear history of your development. Commit your changes frequently with descriptive commit messages to make it easier to revert or collaborate with others.
- **Leverage AWS Amplify's features:** Explore the various features and services provided by AWS Amplify, such as API creation, database integration, file storage, and more. These features can significantly accelerate your development process and enhance your application's functionality.
- **Follow best practices:** Adhere to industry best practices and coding standards while developing your application. This includes writing clean, modular, and well-documented code, conducting thorough testing, and ensuring the security of your application and user data.
- **Stay updated:** Keep an eye out for updates to the tools and libraries used in the XMOPS Accelerate project, such as React, AWS Amplify, and Terraform. Regularly update your dependencies to leverage new features, performance improvements, and security patches.
- **Seek help when needed:** If you encounter any issues or have questions during your development process, don't hesitate to reach out to the XMOPS Accelerate community, consult the official documentation of the tools and libraries you are using, or seek assistance from experienced developers.

Remember, this user manual serves as a foundation for getting started with XMOPS Accelerate. As you become more familiar with the platform and its underlying technologies, you'll be able to customize and extend your application to meet your specific requirements.

We hope you have a productive and enjoyable experience using XMOPS Accelerate to build and deploy your web applications. Happy coding!



USER MANUAL

CONTRIBUTORS

- **Dollar Karan Preet Singh**

Full Stack Developer, UI/UX

Email - mrdollargill@gmail.com

Github - github.com/dollargill

Linkedin - [linkedin.com/in/dollargill](https://www.linkedin.com/in/dollargill)

- **Musrat Jahan Rimu**

UI/UX Designer, Developer

Email - musratjahanrimu@gmail.com

Github - github.com/Musrat-J-Rimu

Linkedin - www.linkedin.com/in/musratjahan

- **Ramandeep Singh Turna**

Full Stack Developer

Email - ramanturna337@gmail.com

Github - github.com/ramanturna

- **Amandeep Singh**

UI/UX Designer, Developer

Email - maanaman290@gmail.com

Github - github.com/Amandeep0025



An **easier** way to deploy website

Meet your new WordPress assistant