**Comparison & Deployment Recommendation: Azure vs AWS for Full-Stack Projects (Laravel + Python + React)**

### 🔍 Overview

This document compares **Microsoft Azure** and **Amazon Web Services (AWS)** for deploying a full-stack application that includes:

* Laravel (PHP backend)
* Python (FastAPI + AI/LLM APIs)
* React (Frontend)
* MySQL (Database)

The goal is to recommend the best infrastructure option for performance, scalability, ease of use, and future flexibility.

### ⚖️ Azure vs AWS: Feature-by-Feature Comparison

| Feature | Azure | AWS |
| --- | --- | --- |
| Ease of Use | ✅ Easier UI, VS Code Integration | ⚫ Complex UI, CLI-focused |
| Laravel (PHP) Support | ✅ App Service / VM | ✅ EC2 / Lightsail |
| Python (FastAPI, AI APIs) | ✅ App Service / VM / Container Apps | ✅ EC2 / Lambda / SageMaker |
| React & Node.js | ✅ Static Web Apps / Blob + CDN | ✅ S3 + CloudFront |
| MySQL | ✅ Azure MySQL | ✅ Amazon RDS (MySQL) |
| File Upload & Storage | ✅ Azure Blob Storage | ✅ Amazon S3 |
| SSL + Custom Domain | ✅ Free SSL + Easy DNS | ✅ Free SSL (CloudFront or ACM) |
| Root Access (for VMs) | ✅ Full root via Azure VM | ✅ Full root via EC2 |
| LLM/AI APIs | ✅ Azure OpenAI, Form Recognizer | ✅ AWS Bedrock, SageMaker |
| Performance & Scaling | ✅ Good | ✅ Excellent (more data centers) |
| Cost Efficiency (small scale) | ✅ More affordable base plans | ⚫ Slightly costlier |
| CI/CD Integration | ✅ GitHub, Azure DevOps | ✅ GitHub, CodePipeline |
| Best For | Teams with MS stack / AI services | DevOps-heavy, Linux-heavy, container use |

### 🚀 Recommended Infrastructure: **Azure VM (Ubuntu)**

**Why:**

* Full root access like your current aaPanel VPS
* Flexibility to run Laravel, FastAPI, and React in one place
* Easy to scale later with App Gateway or Load Balancer
* Better integration with Azure Form Recognizer or OpenAI if needed

**Minimum VM Specs:**

* Ubuntu 22.04 LTS
* 2 vCPU / 4 GB RAM (for dev/testing)
* 8 GB RAM recommended for LLM APIs
* Disk: 100 GB SSD

### 📂 Suggested Folder Structure on Azure VM

/var/www/  
├── aicew-laravel # Laravel project  
├── aicew-fastapi # Python FastAPI APIs  
├── aicew-react # React frontend build

### 🔌 Nginx Reverse Proxy Setup

# Laravel (PHP-FPM)  
server {  
 server\_name aicew.example.com;  
 root /var/www/aicew-laravel/public;  
 index index.php index.html;  
 ...  
}  
  
# FastAPI APIs  
location /api/ {  
 proxy\_pass http://127.0.0.1:8000/;  
 proxy\_set\_header Host $host;  
 ...  
}  
  
# React App  
server {  
 server\_name react.aicew.example.com;  
 root /var/www/aicew-react;  
 index index.html;  
 try\_files $uri /index.html;  
 ...  
}

### 🎯 Summary: Which Should You Choose?

| If you… | Use… |
| --- | --- |
| Already have Azure credits or subscriptions | ✅ Azure VM |
| Need full terminal/root access like aaPanel | ✅ Azure VM |
| Need serverless scaling and microservices | AWS (ECS/Lambda) |
| Prefer Windows/Microsoft ecosystem | Azure |
| Heavy Docker and multi-container use | AWS (ECS) |
| Want lowest pricing at scale | AWS (Reserved EC2) |

### 🔧 Next Steps

* Provision Azure VM with Ubuntu 22.04 LTS
* Install PHP, MySQL, Node.js, Python, Composer, Nginx
* Deploy each project folder
* Set up domain DNS and SSL with Certbot
* Configure firewall (UFW or Azure NSG)
* Optional: Install aaPanel or CapRover if GUI needed