All In One Project Management Workspace

Local Setup + AWS Deployment | Windows & Mac | Beginner friendly

Part 1 (local): 30–50 min | Part 2 (deploy): 1–3 hours

Local Development Setup

Run the application on your computer (~30–50 min)

Complete Part 1 first. Your app must run locally at http://localhost:3000 before deploying to AWS.

This guide helps you run the product on your own computer only. When finished, you will use the app at http://localhost:3000, complete the setup wizard once, and sign in as administrator.

Item	Value
App address	http://localhost:3000
Setup wizard (first time)	http://localhost:3000/install
Sign-in page	http://localhost:3000/sign-in
Skills needed	Copy and paste only — no programming
Internet	Required for packages and Supabase database

Tip: Localhost means this computer only. The public cannot access your site until you deploy to a server later.

1. Prerequisites

Install the required tools before opening the project.

Tool	Required	Purpose
Chrome or Microsoft Edge	Yes	Browser for app and setup wizard
Node.js 20 or newer	Yes	Runs the application
pnpm 10 or newer	Yes	Installs project packages
Terminal / PowerShell	Yes	Run setup commands

Visual Studio Code	Recommended	Open project and built-in terminal
Application ZIP + LICENSE.txt	Yes	Source code and license keys
Supabase account (free)	Yes	PostgreSQL database in the cloud
AWS S3	Optional	File uploads — can skip in wizard
Email (Postmark or SMTP)	Optional	Can skip for local testing

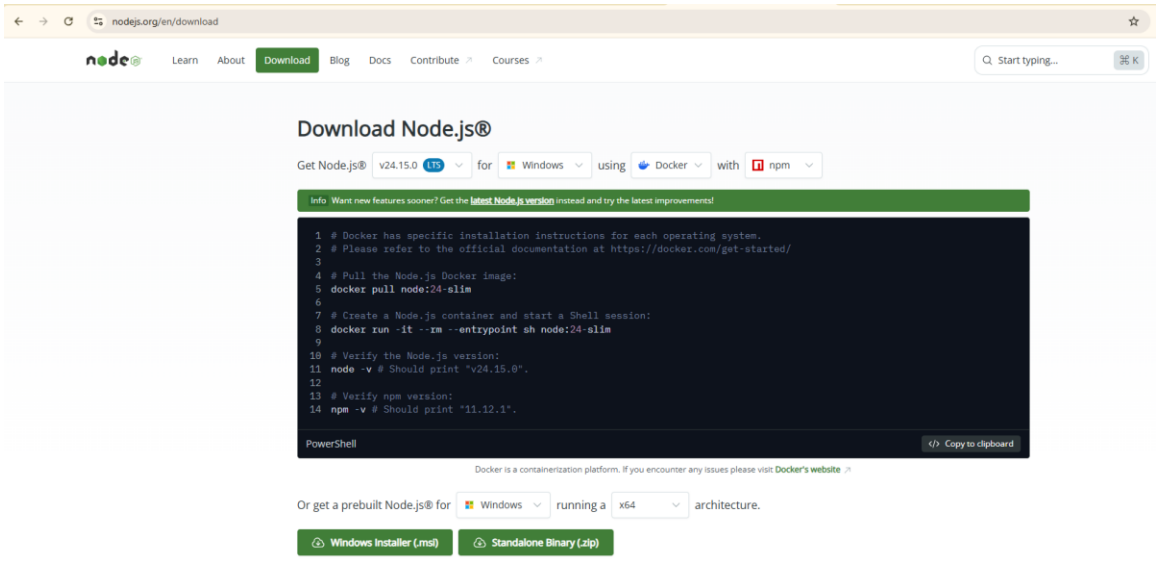
2. Install Required Software

2.1 Web browser

- Install Google Chrome from https://www.google.com/chrome/ or use Microsoft Edge (included on Windows 11).
- Use a normal browser window during setup (avoid strict private/incognito mode).

2.2 Node.js (version 20+)

1. Open https://nodejs.org
2. Download the LTS version (20.x or newer).
3. Run the installer and accept default options.
4. Close and reopen PowerShell or VS Code terminal.



Verify installation:

```
node -v
```

Expected result: v20.x.x or higher (example: v20.18.0)

2.3 pnpm

Install pnpm after Node.js:

```
npm install -g pnpm
```

Verify:

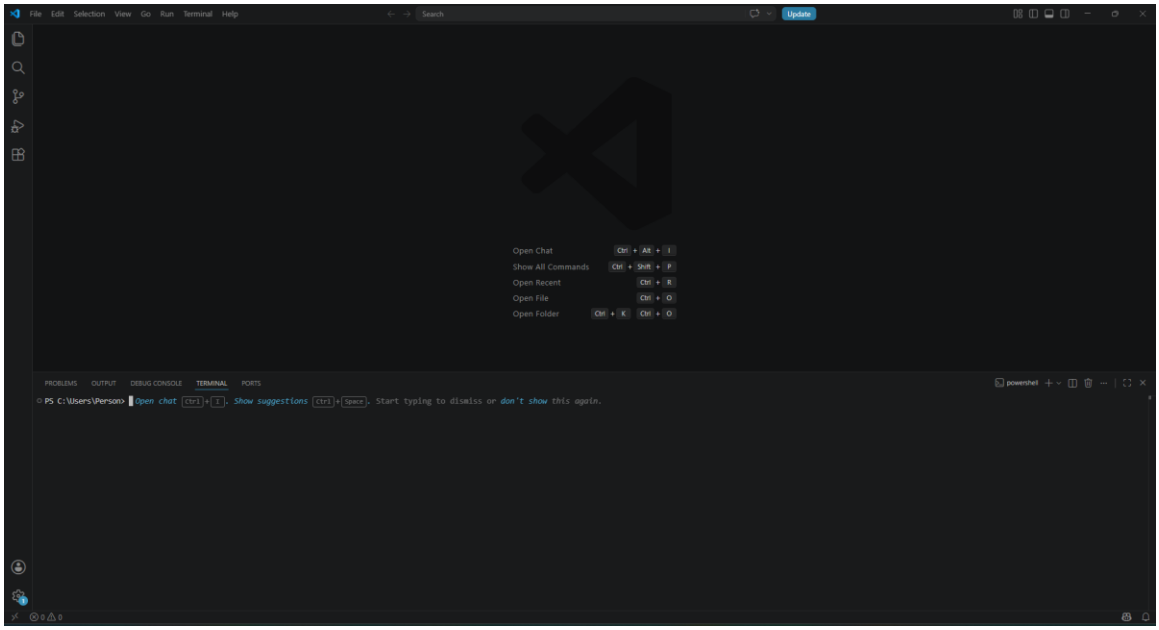
```
pnpm -v
```

Expected result: Version 10 or higher (example: 10.7.1)

Windows tip: If the command fails, open PowerShell as Administrator, run the install again, then restart VS Code.

2.4 Terminal (Command Prompt / Terminal)

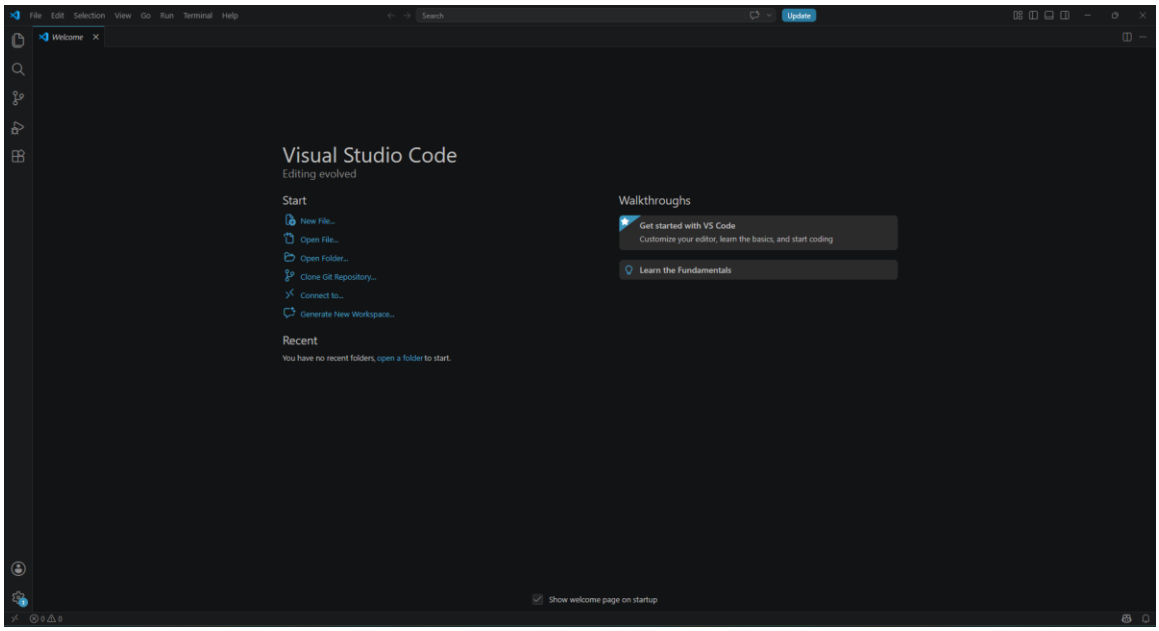
System	How to open
Windows	Press Win + X → Terminal or PowerShell
Mac	Cmd + Space → type Terminal → Enter
VS Code	Menu: Terminal → New Terminal (recommended)



2.5 Visual Studio Code

1. Download from https://code.visualstudio.com
2. Install with default settings.

3. Open VS Code → File → Open Folder → select your project folder.



Optional extensions

Extension	Purpose
Prisma	View database schema files
DotENV	Easier reading of .env file

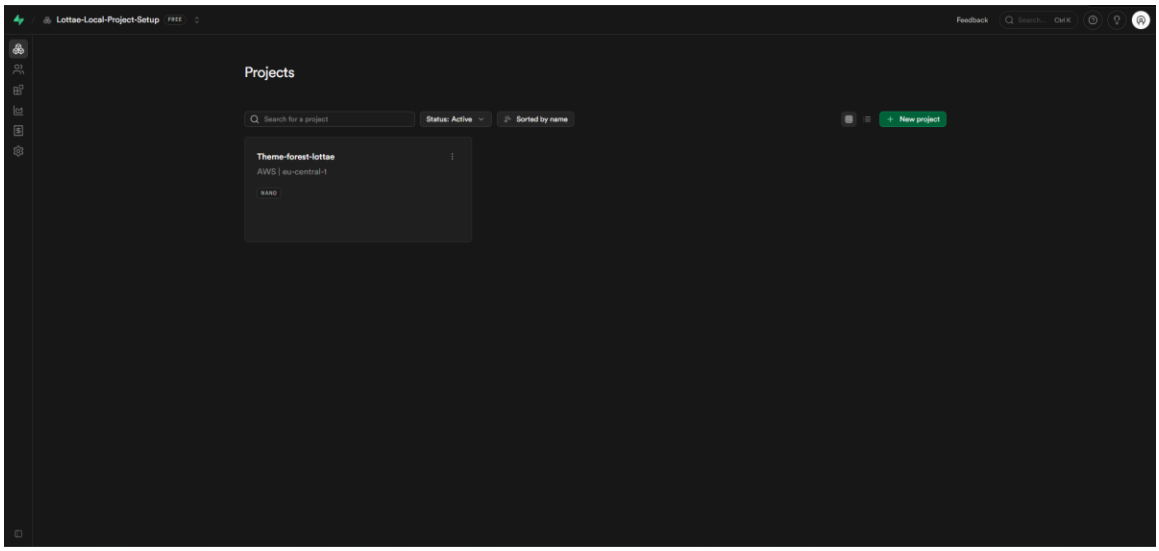
Note: Extensions are optional. The app runs without them.

3. Configure Supabase Database

Your computer runs the app. Data is stored in PostgreSQL on Supabase (free tier).

3.1 Create a supabase project.

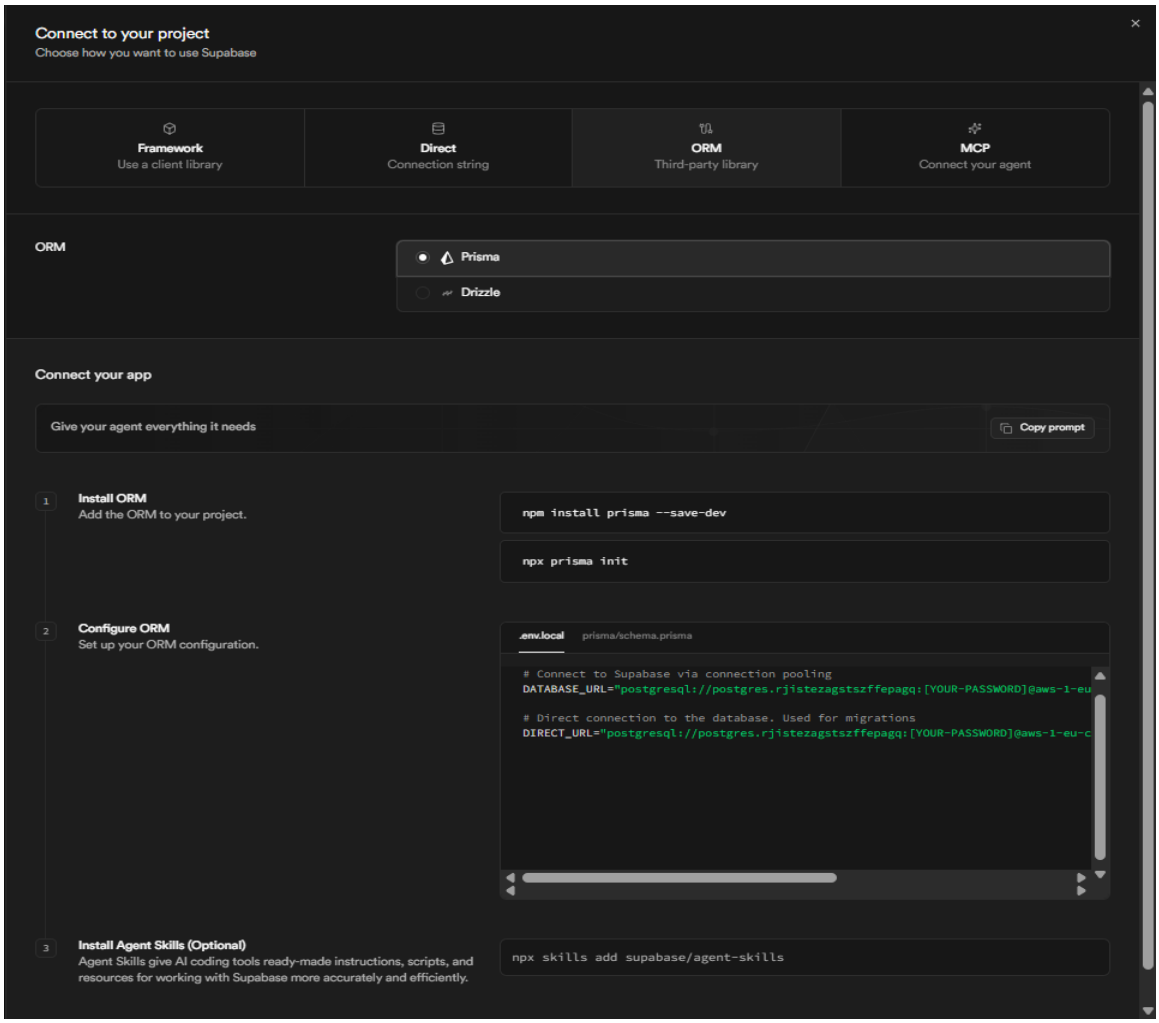
1. Go to https://supabase.com and sign up.
2. Click New project.
3. Set project name (example: project-local).
4. Set a strong database password and save it securely.
5. Choose the region closest to you.
6. Click Create new project and wait until status is Active.



3.2 Copy connection strings

1. In Supabase: Project Settings (gear) → Database.
2. Under Connection string, select URI.
3. Copy two URLs (replace [YOUR-PASSWORD] with your real password).

Wizard field	**Supabase source**	**Port**
Database URL	Connection pooling → Transaction mode	6543 (pooler)
Direct Database URL	Direct connection or Session pooler	5432

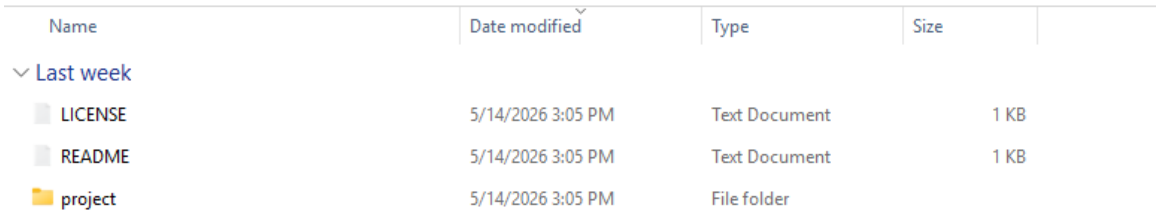


Important: Do not use port 6543 for Direct Database URL. If your password contains @ or #, encode them in the URL (@ becomes %40).

4. Download and Extract Project Files

4.1 Download from ThemeForest

1. Log in to ThemeForest → Downloads.
2. Download the main ZIP file for this item.
3. Download bundle files that include LICENSE.txt.



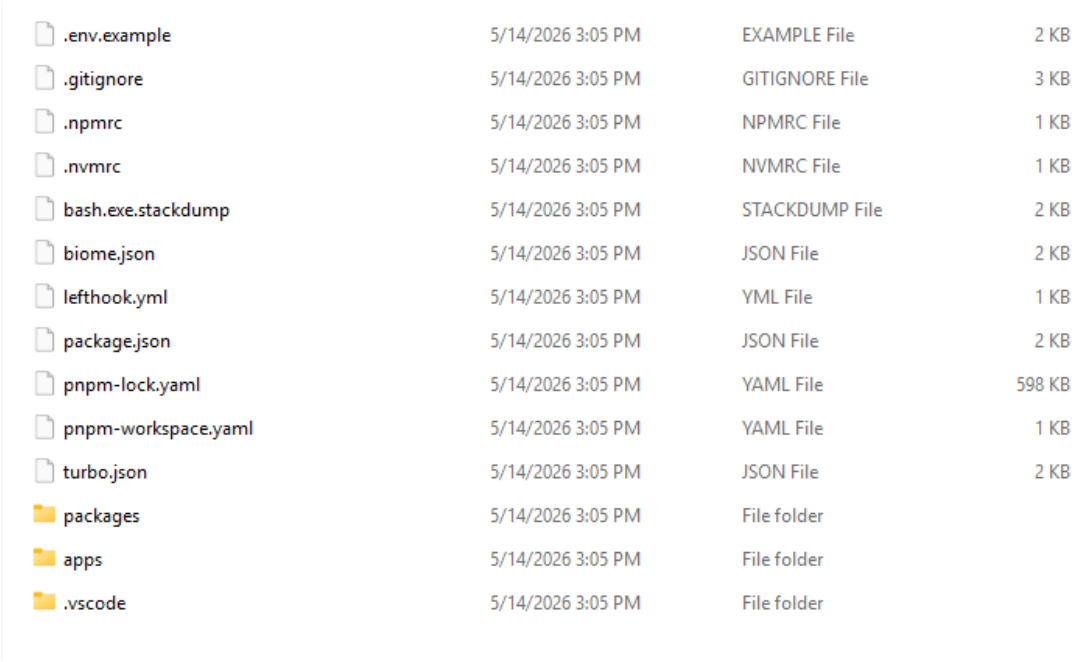
Name	Date modified	Type	Size
Last week			
LICENSE	5/14/2026 3:05 PM	Text Document	1 KB
README	5/14/2026 3:05 PM	Text Document	1 KB
project	5/14/2026 3:05 PM	File folder	

4.2 Extract ZIP

- Windows: Right-click ZIP → Extract All → choose Documents\Project-Management
- Mac: Double-click ZIP → move folder to Documents

Correct folder must contain at the top level:

- package.json
- pnpm-workspace.yaml
- apps folder
- packages folder



.env.example	5/14/2026 3:05 PM	EXAMPLE File	2 KB
.gitignore	5/14/2026 3:05 PM	GITIGNORE File	3 KB
.npmrc	5/14/2026 3:05 PM	NPMRC File	1 KB
.nvmrc	5/14/2026 3:05 PM	NVMRC File	1 KB
bash.exe.stackdump	5/14/2026 3:05 PM	STACKDUMP File	2 KB
biome.json	5/14/2026 3:05 PM	JSON File	2 KB
lefthook.yml	5/14/2026 3:05 PM	YML File	1 KB
package.json	5/14/2026 3:05 PM	JSON File	2 KB
pnpm-lock.yaml	5/14/2026 3:05 PM	YAML File	598 KB
pnpm-workspace.yaml	5/14/2026 3:05 PM	YAML File	1 KB
turbo.json	5/14/2026 3:05 PM	JSON File	2 KB
packages	5/14/2026 3:05 PM	File folder	
apps	5/14/2026 3:05 PM	File folder	
.vscode	5/14/2026 3:05 PM	File folder	

Warning: If package.json is inside a nested subfolder, open the inner folder that contains apps and packages.

4.3 LICENSE.txt

- Open LICENSE.txt from your bundle.
- Copy the Purchase code (UUID format).
- Copy the License key (64 characters).

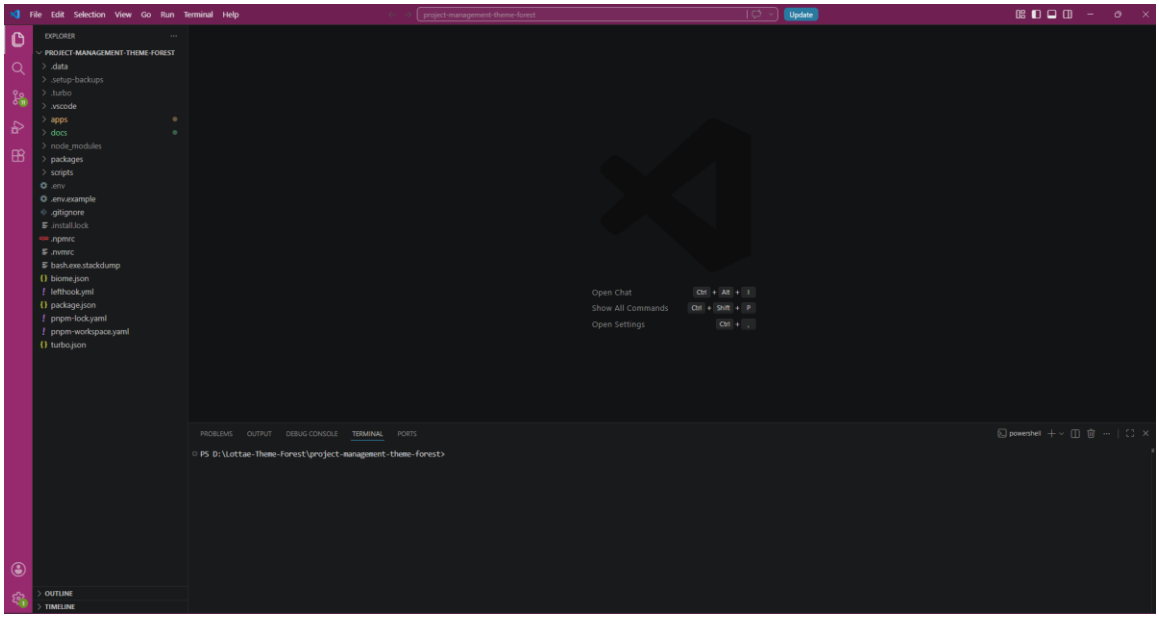
- Keep both ready for the setup wizard.

File Edit View

```
Project Management — issued project bundle credentials
==================================================

Purchase code: a8ba48b8-axo3-44f2-8505-4e55b7f1faf3
License key:   be327f68498fc3775cb9ab546261b25778bcf2b2b1339a566be0dcee39b8052d

Issued at (UTC): 2026-05-14T22:03:11.524Z
Issued to:       admin@masterportal.com

Keep these values secure. Do not commit this file to public repositories.
```

5. Open the Project in VS Code

1. Open VS Code.
2. File → Open Folder.
3. Select the folder that contains package.json.
4. If prompted to trust authors, click Yes, I trust the authors.
5. Terminal → New Terminal.
6. Confirm the terminal path shows your project folder.



6. Run the Application Locally

6.1 Install packages (first run only)

In the VS Code terminal, run:

```
pnpm install
```

What it does	Downloads all required packages (3–10 minutes)
Success	Finishes without red ERROR lines

6.2 Start the development server

```
pnpm dev
```

What it does	Starts the app on your computer
Success	Terminal shows Ready and localhost:3000
Important	Keep this terminal open while using the app

1. Open Chrome or Edge.
2. Go to http://localhost:3000
3. You should be redirected to http://localhost:3000/install

Stop server: Press Ctrl + C in the terminal when finished.

7. Complete the Installation Wizard

Complete all steps in one browser tab. Do not clear browser data until finished.

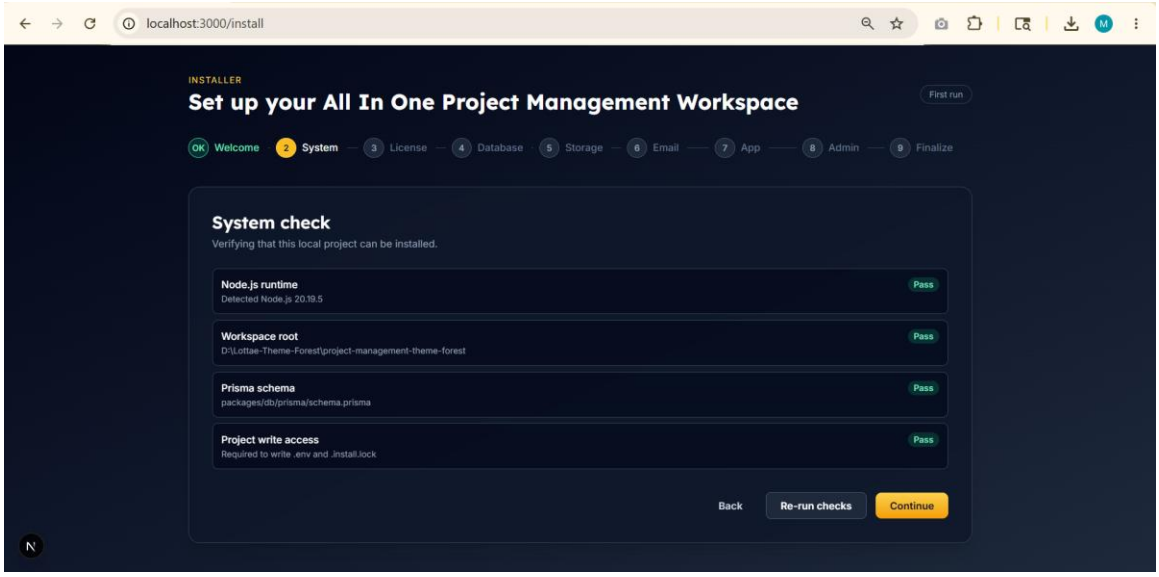
Step 1 — Welcome

- Click Start installation.
- Expected: moves to System check.

Step 2 — System check

All items must show Pass:

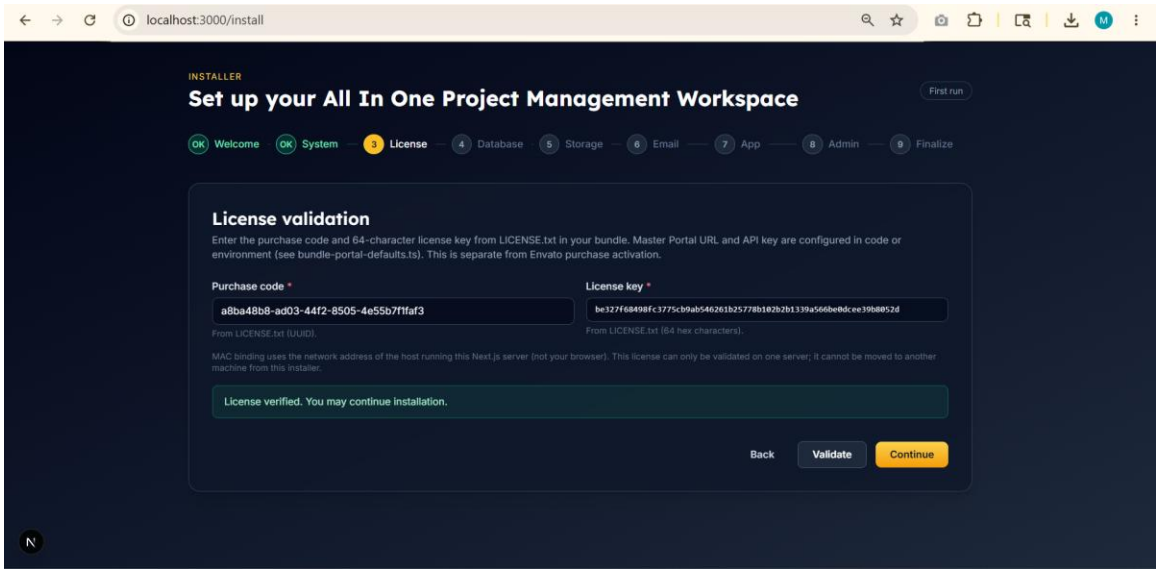
- Node.js runtime (20+)
- Workspace root
- Prisma schema
- Project write access



If failed	Fix
Node.js	Install Node 20+, restart terminal, Re-run checks
Write access	Move project to Documents folder, not Program Files

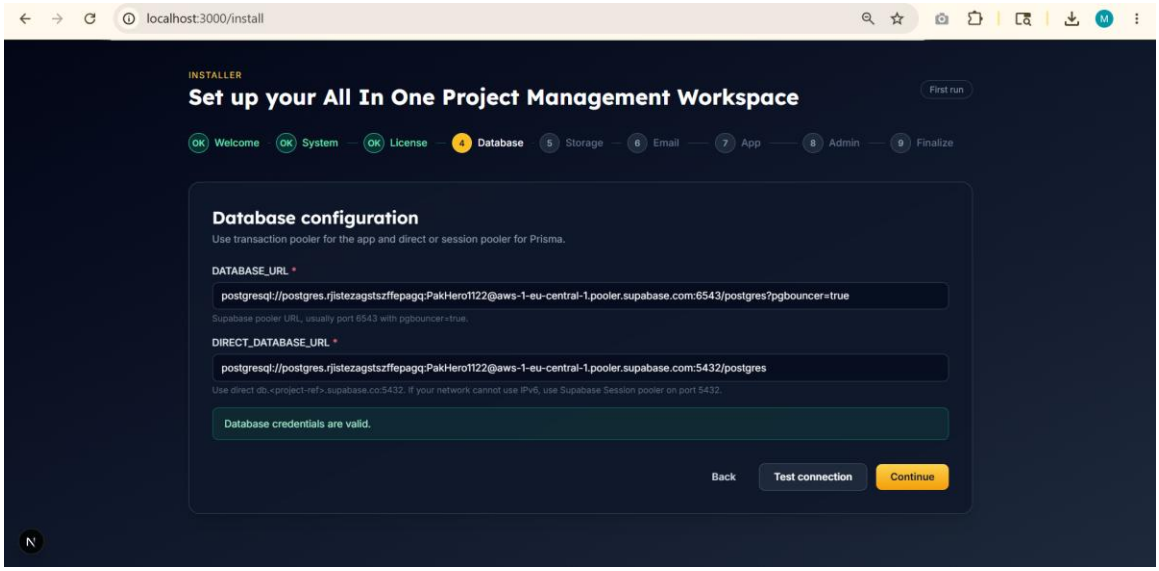
Step 3 — License

- Paste Purchase code from LICENSE.txt.
- Paste License key (64 characters).
- Click Validate.
- Continue when validation succeeds.



Step 4 — Database

- Paste Database URL (Supabase pooler, port 6543).
- Paste Direct Database URL (port 5432).
- Click Test connection.
- Continue when test passes.



Step 5 — Storage (optional)

- Skip for local testing, or enter AWS S3 credentials if you have them.
- Click Continue.

Step 6 — Email (optional)

- Skip for local testing, or configure Postmark/SMTP to test emails.
- Click Continue.

Step 7 — App settings

Field	Local value
Application URL	http://localhost:3000
Public App URL	http://localhost:3000
Base URL	http://localhost:3000

Optional: Enable Import demo data for sample projects and tasks.

Step 8 — Admin account

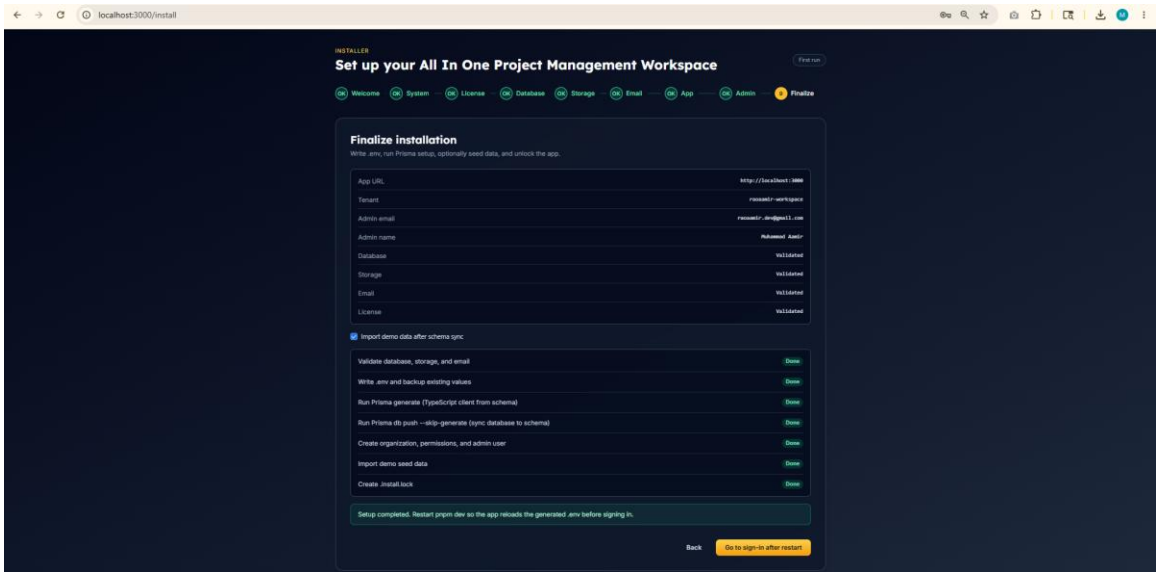
There is no default password. Create your own administrator:

Field	Requirement
Admin email	Valid email address
Display name	At least 2 characters
Password	At least 8 characters
Confirm password	Must match password

Warning: Save your email and password. You need them to sign in.

Step 9 — Finalize

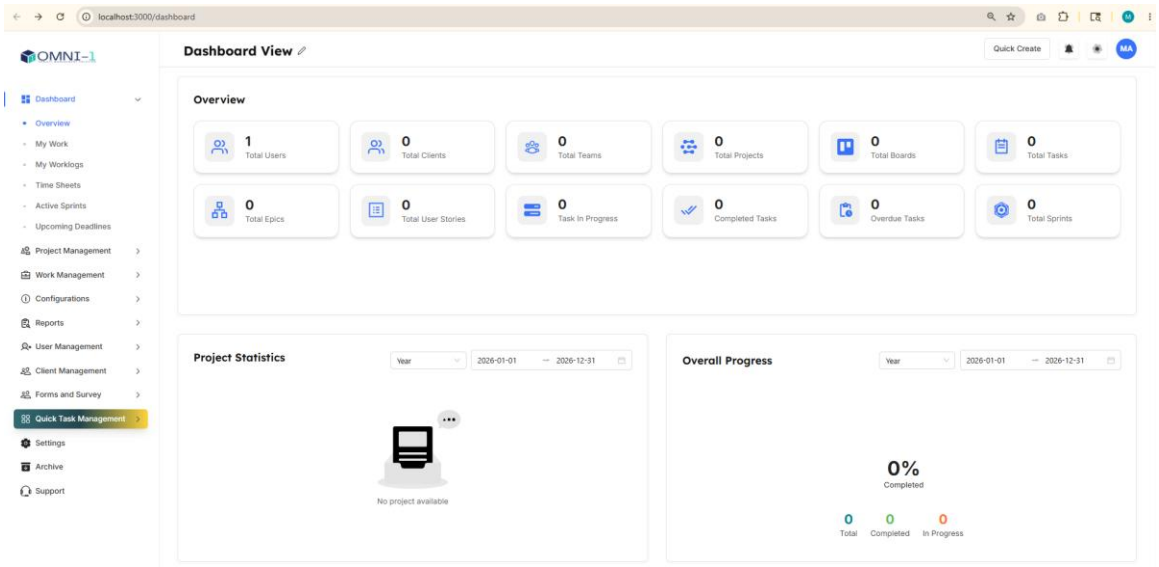
- Review the summary.
- Click Install or Complete setup.
- Wait until every stage shows Done.
- Validate services
- Write .env file
- Prisma generate and database sync
- Create organization and admin user
- Create .install.lock file



Verify in project root: .env and .install.lock files exist.

8. Sign In and Verify

1. Open http://localhost:3000/sign-in
2. Enter admin email and password from Step 8.
3. Click Sign in.



Verification checklist

- Website opens at http://localhost:3000
- Admin login works
- No database errors on screen
- License was validated in wizard
- Setup wizard is locked (install completed)

Daily workflow

1. Open VS Code → open project folder.
2. Run: pnpm dev
3. Open browser: http://localhost:3000
4. Stop server: Ctrl + C in terminal

9. Troubleshooting

node is not recognized

Cause: Node.js not installed or terminal not restarted.

Solution: Install Node 20+, close VS Code, reopen, run node -v.

pnpm is not recognized

Cause: pnpm not installed globally.

Solution: Run: npm install -g pnpm (Administrator PowerShell on Windows).

Port 3000 already in use

Cause: Previous dev server still running.

Solution: Close old terminal or end process on port 3000, then run pnpm dev again.

Database connection failed

Cause: Wrong URL, paused Supabase, or swapped pooler/direct URLs.

Solution: Use 6543 for Database URL and 5432 for Direct URL; confirm project is Active.

License validation failed

Cause: Wrong code/key or license used on another machine.

Solution: Re-copy from LICENSE.txt; contact support for license reset if needed.

Cannot write .env

Cause: Project in protected folder.

Solution: Move project to Documents and reopen in VS Code.

10. Quick Reference (Local)

Action	**Command or URL**
Open project	VS Code → File → Open Folder
Install packages	pnpm install
Start app	pnpm dev
Open in browser	http://localhost:3000
Sign in	http://localhost:3000/sign-in
Stop app	Ctrl + C in terminal

AWS Production Deployment

Publish the application on the internet (~1–3 hours)
After local setup works, deploy to AWS so your team can access the app on the internet. You need: a domain name (optional for testing), PostgreSQL (Supabase recommended), and AWS account.

Important: Update all localhost URLs in .env to your live HTTPS domain before production deploy.

11. Pre-Deployment Checklist

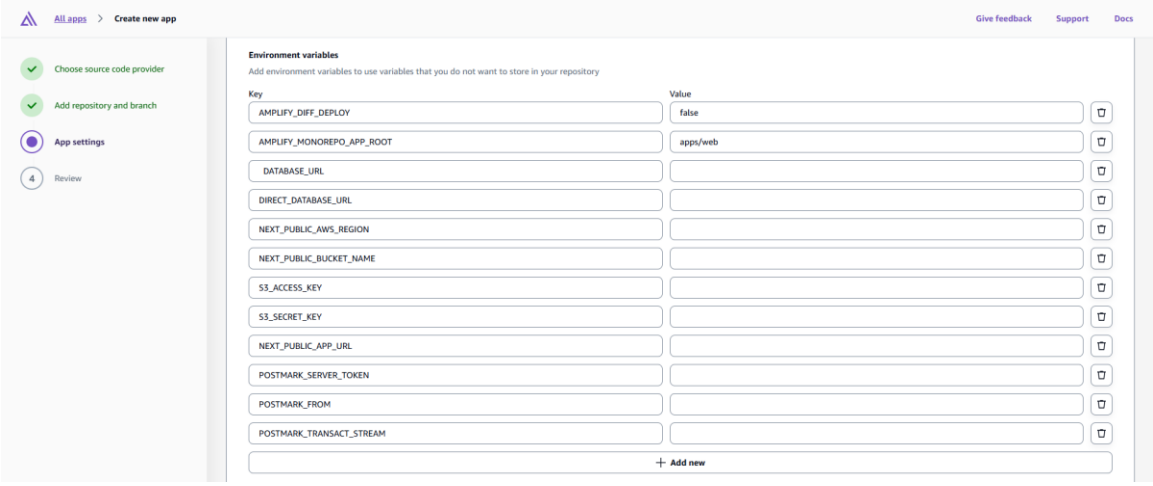
11.1 Deployment checklist

- Local setup completed (Part 1) — you can sign in at localhost.
- Production PostgreSQL database ready (Supabase or AWS RDS).
- Domain name pointed to your host (for HTTPS).
- LICENSE.txt purchase code validated on the production server.
- .env values prepared for production (see table below).

11.2 Production environment variables

Set these in AWS Amplify Console or in .env on EC2 (never commit .env to Git):

Variable	Production example
BETTER_AUTH_URL	https://app.yourdomain.com
NEXT_PUBLIC_APP_URL	https://app.yourdomain.com
NEXT_PUBLIC_BASE_URL	https://app.yourdomain.com
DATABASE_URL	Supabase pooler URL (port 6543)
DIRECT_DATABASE_URL	Supabase direct URL (port 5432)
BETTER_AUTH_SECRET	Long random secret (32+ characters)
AWS S3 / email keys	As configured in App Settings



11.3 Choose a hosting option

Option	Best for	Difficulty
AWS Amplify	Fastest deploy, managed CI/CD, auto SSL	Easier
AWS EC2	Full server control, custom nginx/PM2	Moderate

12. Deploy with AWS Amplify

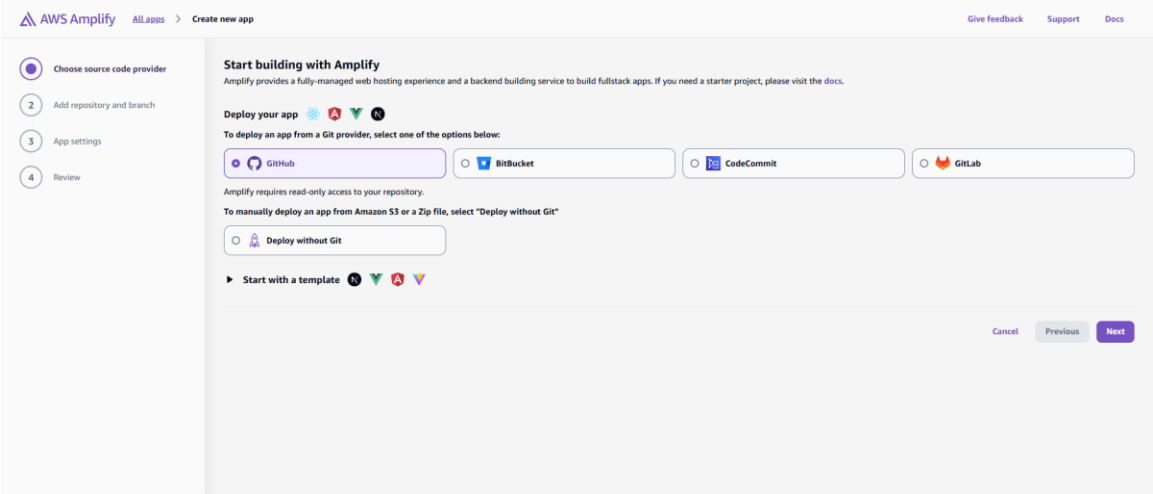
AWS Amplify hosts Next.js apps with automatic builds from GitHub, GitLab, or Bitbucket.

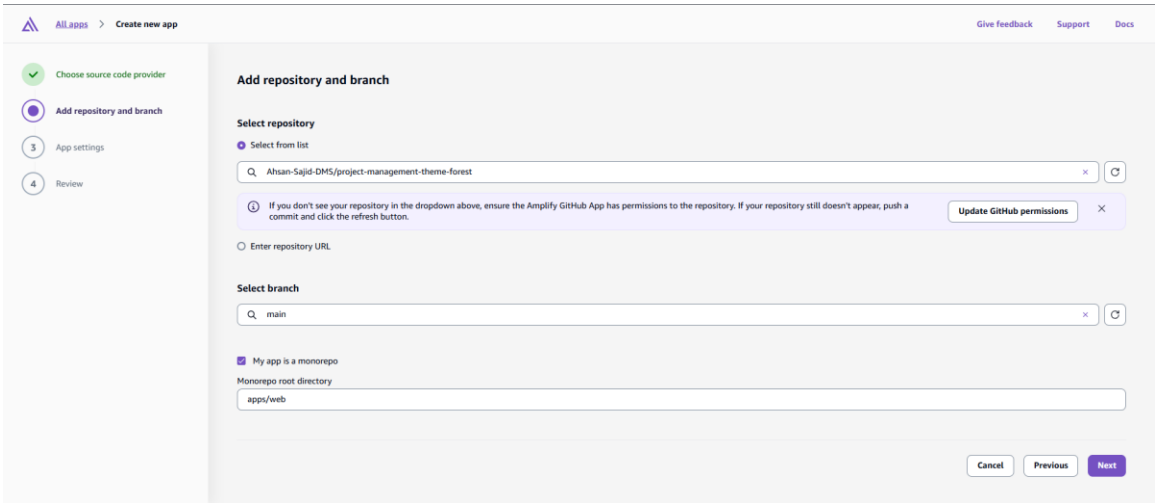
12.1 Prerequisites

- AWS account — https://aws.amazon.com
- Code in a Git repository (GitHub recommended).
- Node.js 20 supported (Amplify selects runtime automatically).

12.2 Connect repository

1. Open AWS Console → search Amplify → Hosting → Get started.
2. Choose Host web app → GitHub (or your provider) → Authorize AWS.
3. Select your repository and branch (e.g. main).
4. App name: e.g. project-management-workspace.





12.3 Configure build settings (monorepo)

Use these settings for this project (pnpm monorepo):

Setting	Value
App root	/ (repository root)
Build command	pnpm install && pnpm build
Start command / output	Amplify detects Next.js — use default SSR
Node.js version	20.x
Package manager	pnpm

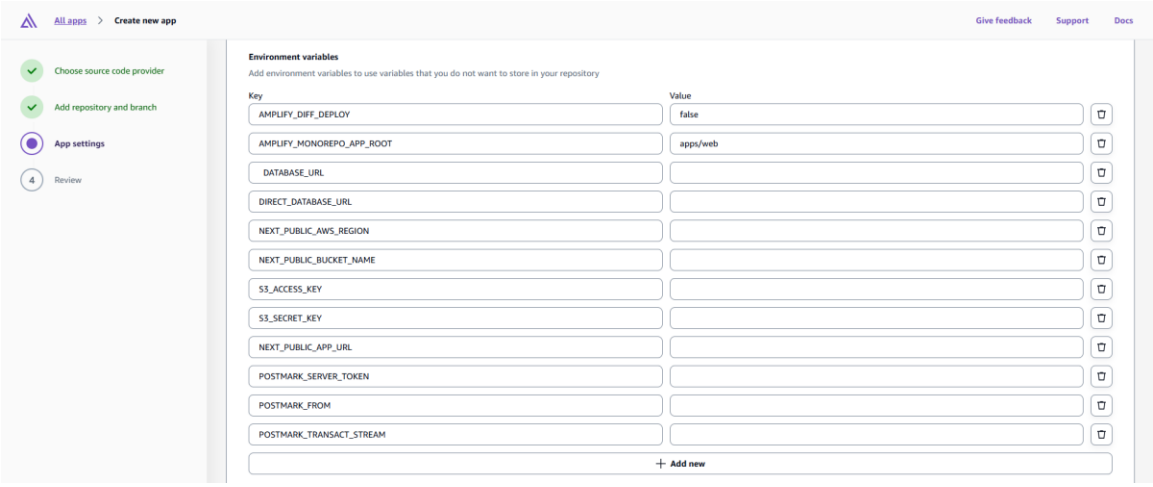
Optional: add amplify.yml in repository root:

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm install -g pnpm
            - pnpm install
        build:
          commands:
            - pnpm build
      artifacts:
        baseDirectory: apps/web/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
          - .pnpm-store/**/*
```

Tip: If build fails, check Amplify build logs. Ensure DATABASE_URL is set before build if env validation runs at build time.

12.4 Environment variables in Amplify

1. Amplify app → Environment variables → Manage variables.
2. Add every key from your production .env (Section 11.2).
3. Save and redeploy.



12.5 Custom domain and SSL

1. Amplify app → Hosting → Custom domains → Add domain.
2. Follow DNS instructions (CNAME records at your registrar).
3. Amplify provisions free SSL (HTTPS) automatically.
4. Update BETTER_AUTH_URL and NEXT_PUBLIC_* to https://yourdomain.com.

12.6 Run install wizard on production

1. Open https://yourdomain.com/install (first visit only).
2. Complete wizard: license, database, admin account.
3. Sign in at https://yourdomain.com/sign-in.

Verification: Site loads over HTTPS, admin login works, browser tab shows your favicon and site title.

12.7 Amplify troubleshooting

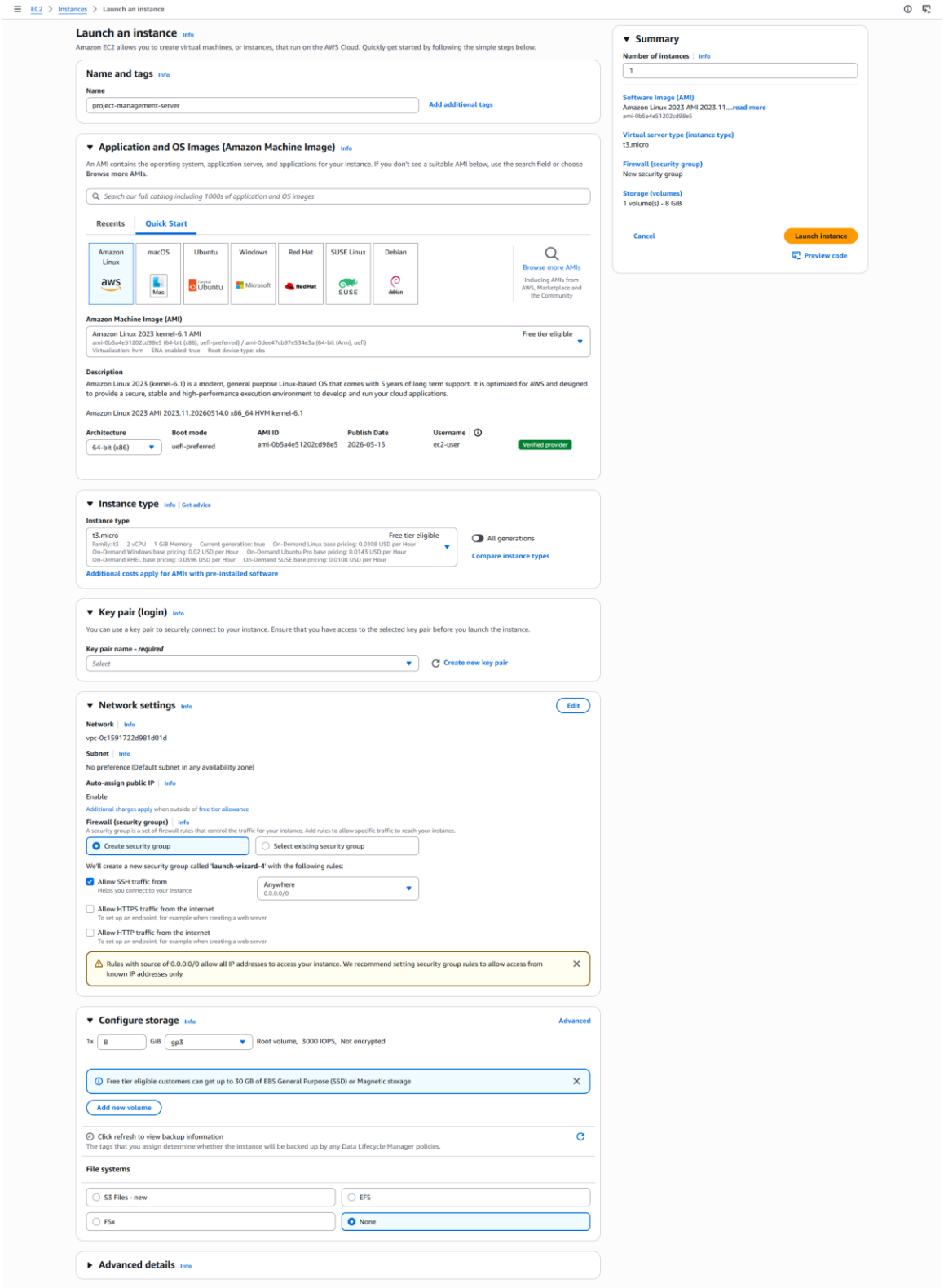
Problem	Solution
Build fails	Check Node 20, pnpm install logs; set SKIP_ENV_VALIDATION=1 only if needed for build
502 / app error	Verify env vars; check Amplify runtime logs
Database error	Allow Amplify outbound IP in Supabase; correct DATABASE_URL
Auth redirect loop	BETTER_AUTH_URL must match exact HTTPS domain

13. Deploy on AWS EC2

EC2 gives you a Linux server. You install Node.js, build the app, and run it with PM2 behind nginx.

13.1 Create EC2 instance

1. AWS Console → EC2 → Launch instance.
2. Name: project-management-server.
3. AMI: Ubuntu Server 22.04 LTS (64-bit).
4. Instance type: t3.small minimum (t3.medium recommended).
5. Create or select a key pair (.pem) for SSH.
6. Security group: allow SSH (22), HTTP (80), HTTPS (443).
7. Storage: 30 GB+.
8. Launch instance.


EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
project-management-server
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 kernel-6.1 AMI
Free tier eligible
Description
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
Amazon Linux 2023 AMI 2023.11.20260514.0 x86_64 HVM kernel-6.1
Architecture
Boot mode
AMI ID
Publish Date
Username
64-bit (x86)
uefi-preferred
ami-0b5a4e51202cd98e5
2026-05-15
ec2-user
Verified provider
Instance type Info | Get advice
Instance type
t3.micro
Free tier eligible
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Select
Create new key pair
Network settings Info
Edit
Network Info
vpc-0c1591722d981d01d
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-4' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Configure storage Info
Advanced
1x
8
GiB
gp3
Root volume, 3000 IOPS, Not encrypted
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage
Add new volume
Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.
File systems
S3 Files - new
EFS
FSx
None
Advanced details Info
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.11....read more
ami-0b5a4e51202cd98e5
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code

9.

13.2 Connect to server

From your computer (replace with your key and IP):

```
ssh -i "your-key.pem" ubuntu@YOUR_EC2_PUBLIC_IP
```

Windows: Use PowerShell or PuTTY. On first connect, type yes to trust the host.

13.3 Install Node.js and pnpm on EC2

1. Update packages: sudo apt update && sudo apt upgrade -y
2. Install Node 20: curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash - && sudo apt install -y nodejs
3. Verify: node -v (must show v20+)
4. Install pnpm: sudo npm install -g pnpm
5. Verify: pnpm -v
6. Install PM2: sudo npm install -g pm2
7. Install nginx: sudo apt install -y nginx

13.4 Upload project to EC2

Choose one method:

- Git clone: sudo apt install -y git → git clone YOUR_REPO_URL → cd project folder
- SFTP: Upload ZIP with FileZilla/WinSCP to /home/ubuntu/ and unzip
- Place project in e.g. /home/ubuntu/project-management-theme-forest
- Create .env in project root with production values (Section 11.2).

13.5 Build and database on EC2

1. cd /home/ubuntu/project-management-theme-forest
2. pnpm install
3. pnpm exec dotenv -e .env -- pnpm db:push (sync database schema)
4. pnpm build

Expected: Build completes without errors. If env validation fails, fix .env before build.

13.6 Run app with PM2

```
cd /home/ubuntu/project-management-theme-forest
pm2 start "pnpm start" --name lottae-app
pm2 save
pm2 startup
```

App runs on port 3000 internally. Verify: curl http://localhost:3000

13.7 Configure NGINX reverse proxy

Create nginx site config:

```
sudo nano /etc/nginx/sites-available/lottae

server {
```

```
        listen 80;
        server_name app.yourdomain.com;
        location / {
            proxy_pass http://127.0.0.1:3000;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection 'upgrade';
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_cache_bypass $http_upgrade;
        }
    }
```

1. Enable site: sudo ln -s /etc/nginx/sites-available/lottae /etc/nginx/sites-enabled/
2. Test: sudo nginx -t
3. Reload: sudo systemctl reload nginx

13.8 Enable HTTPS with Let's Encrypt

1. Point domain A record to EC2 public IP.
2. sudo apt install -y certbot python3-certbot-nginx
3. sudo certbot --nginx -d app.yourdomain.com
4. Follow prompts; certbot configures HTTPS automatically.

Warning: Update .env URLs to https://app.yourdomain.com and restart: pm2 restart lottae-app

13.9 Run the installation wizard in production

1. Open https://app.yourdomain.com/install
2. Complete license, database, and admin steps.
3. Sign in and test file uploads and email if configured.

13.10 EC2 troubleshooting

Problem	Solution
Cannot SSH	Check security group port 22; correct .pem permissions (chmod 400)
Site not loading	pm2 status; nginx -t; sudo ufw allow 80,443 if firewall enabled
502 Bad Gateway	Ensure pnpm start is running on port 3000
Out of memory on build	Use larger instance or add swap space

14. Post-Deployment Checklist

- HTTPS works (padlock in browser).

- https://yourdomain.com/sign-in — admin login works.
- Install wizard completed; .install.lock exists on server.
- Site title and favicon show in browser tab.
- Database backups scheduled (Supabase or RDS).
- PM2 or Amplify monitoring enabled.

15. Updating production

1. Amplify: push to Git — Amplify rebuilds automatically.
2. EC2: git pull (or upload new files) → pnpm install → pnpm build → pm2 restart lottae-app
3. Always backup database before major updates.

16. Quick Reference (Production)

Task	**Amplify**	**EC2**
Deploy	Git push → auto build	git pull + pnpm build + pm2 restart
Env vars	Amplify Console	.env file on server
Logs	Amplify build/runtime logs	pm2 logs lottae-app
SSL	Amplify custom domain	certbot --nginx

End of Setup Guide