

# MERNlite (React + Express + JSON) Setup on Ubuntu AMIs

## ■ Part A — Common VPC + Security Setup

VPC with public subnet + private subnet

IGW for public, NAT for private

Security Groups:

- demo-web-sg on public EC2: inbound 22 (your IP), 80 (0.0.0.0/0)
- demo-app-sg on private EC2: inbound TCP 4000 from demo-web-sg

## ■ Part B — Private EC2 (Backend on Ubuntu)

Launch Ubuntu Server 22.04/24.04 in private subnet with SG = demo-app-sg

```
# On private EC2 (Ubuntu)
sudo apt update && sudo apt upgrade -y
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt install -y nodejs git
```

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

```
mkdir -p ~/backend && cd ~/backend
npm init -y
npm install express cors
```

```
cat > weather.json <<'JSON'
[
  { "city": "Hyderabad", "date": "2025-08-20", "tempC": 28, "conditions": "Cloudy", "humidity": 78 },
  { "city": "Mumbai", "date": "2025-08-20", "tempC": 26, "conditions": "Rain", "humidity": 88 },
  { "city": "Bengaluru", "date": "2025-08-20", "tempC": 24, "conditions": "Partly Cloudy", "humidity": 70 }
]
JSON
```

```
cat > server.js <<'JS'
const express = require('express');
const cors = require('cors');
const fs = require('fs');

const PORT = process.env.PORT || 4000;
const app = express();
app.use(cors());
app.use(express.json());

let records = [];
function loadData() {
  try {
    records = JSON.parse(fs.readFileSync('./weather.json', 'utf8'));
    console.log(`Loaded ${records.length} weather records`);
  } catch (e) {
    console.error('Failed to load weather.json', e.message);
    records = [];
  }
}
loadData();

function isoToday() { return new Date().toISOString().slice(0,10); }
function findTodayOrLatest(city) {
  const c = city.trim().toLowerCase();
  const cityRows = records.filter(r => r.city.toLowerCase() === c);
```

```

    if (cityRows.length === 0) return null;
    const today = isoToday();
    const recToday = cityRows.find(r => r.date === today);
    return recToday || cityRows.sort((a,b)=>b.date.localeCompare(a.date))[0];
  }

app.get('/api/weather', (req,res)=>{
  const city = (req.query.city||'').trim();
  if (!city) return res.status(400).json({error:'city is required'});
  const rec = findTodayOrLatest(city);
  if (!rec) return res.status(404).json({error:`No weather for ${city}`});
  res.json(rec);
});

app.listen(PORT, '0.0.0.0', ()=>console.log(`Backend listening on ${PORT}`));
JS

```

```

cat | sudo tee /etc/systemd/system/demo-backend.service > /dev/null <<'UNIT'
[Unit]
Description=Demo Backend (Express + JSON)
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/backend
Environment=NODE_ENV=production
ExecStart=/usr/bin/node /home/ubuntu/backend/server.js
Restart=always

[Install]
WantedBy=multi-user.target
UNIT

sudo systemctl daemon-reload
sudo systemctl enable --now demo-backend
sudo systemctl status demo-backend --no-pager

```

## ■ Part C — Public EC2 (Frontend + Nginx on Ubuntu)

Launch Ubuntu Server AMI in public subnet with SG = demo-web-sg

```

sudo apt update && sudo apt upgrade -y
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt install -y nodejs nginx git

```

```

mkdir -p ~/frontend && cd ~/frontend
npm create vite@latest . -- --template react
npm install

cat > src/App.jsx <<'REACT'
import { useState } from 'react';

export default function App() {
  const [city, setCity] = useState('');
  const [result, setResult] = useState(null);
  const [err, setErr] = useState(null);

  async function fetchWeather() {
    setErr(null); setResult(null);
    try {
      const res = await fetch(`~api/weather?city=${encodeURIComponent(city)}`);

```

```

    const data = await res.json();
    if (!res.ok) throw new Error(data.error || 'Request failed');
    setResult(data);
  } catch (e) {
    setErr(e.message);
  }
}

return (
  <div>
    City Weather (Today)
    <input
      placeholder="Enter city (Hyderabad)"
      style={{width:"100%",padding:12,fontSize:16}}/>

    <button>Get Weather</button>

    {err && Error: {err}}
    {result && {JSON.stringify(result,null,2)}}

  </div>
);
}
REACT

npm run build

```

```

sudo mkdir -p /var/www/frontend
sudo rsync -a ~/frontend/dist/ /var/www/frontend/

sudo tee /etc/nginx/sites-available/frontend <<'NGINX'
server {
    listen 80 default_server;
    server_name _;

    root /var/www/frontend;
    index index.html;

    location / {
        try_files $uri /index.html;
    }

    location /api/ {
        proxy_pass http://10.0.2.123:4000;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
NGINX

sudo ln -s /etc/nginx/sites-available/frontend /etc/nginx/sites-enabled/frontend
sudo nginx -t
sudo systemctl enable --now nginx

```

## ■ Part D — Test

```

# In browser: http://
# Enter Hyderabad / Mumbai / Bengaluru → should show JSON weather

# CLI test:

```

```
curl "http:///api/weather?city=Hyderabad"
```