

WeldVision X5 Deployment - Visual Quick Reference

Print this page or save as image for quick lookup!

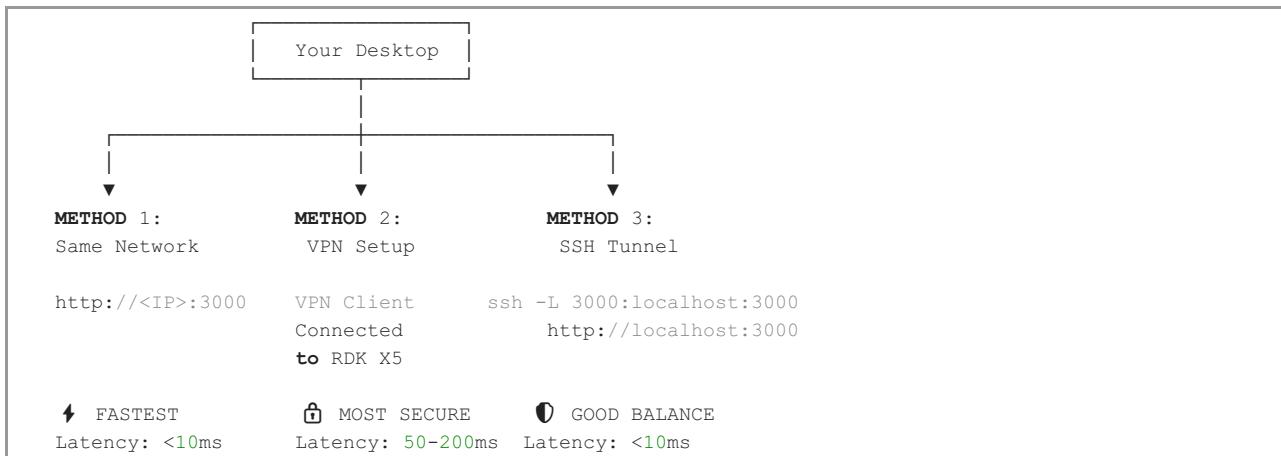
⌚ The 3-Step Deployment Process

```
STEP 1: PREPARE
└── Hardware
    └── RDK X5 (powered on)
    └── RDK Stereo Camera (connected to MIPI)
    └── Ethernet cable (or WiFi)
└── Network
    └── Find RDK X5 IP address
    └── SSH access verified

STEP 2: DEPLOY
└── SSH to RDK X5
└── Copy code
└── Install dependencies
    └── Python (backend)
    └── Node.js (frontend)
└── Build frontend
└── Start services

STEP 3: ACCESS
└── Open browser
└── Navigate to http://<RDK_X5_IP>:3000
└── Add student
└── Test Live Scanner
└── Done! 
```

🌐 Network Access Methods



📊 Deployment Methods Comparison

QUICK START			
Setup: 5-10 min	Production: Not yet	Multiple: X	
Complexity: —	Learning: —	Docker: X	

COMPLETE SYSTEMD GUIDE			
Setup: 30-45min	Production: ✓ YES	Multiple: X	
Complexity: —	Learning: —	Docker: X	

DOCKER DEPLOYMENT			
Setup: 15-20min	Production: ✓ YES	Multiple: ✓	
Complexity: —	Learning: —	Docker: ✓	

💡 Key Ports Reference

FRONTEND			
<table border="1"> <tr> <td>Port 3000</td> </tr> <tr> <td>React/Vite App</td> </tr> <tr> <td>http://<IP>:3000</td> </tr> </table>	Port 3000	React/Vite App	http://<IP>:3000
Port 3000			
React/Vite App			
http://<IP>:3000			
BACKEND API			
<table border="1"> <tr> <td>Port 5000</td> </tr> <tr> <td>Flask Server</td> </tr> <tr> <td>http://<IP>:5000</td> </tr> </table>	Port 5000	Flask Server	http://<IP>:5000
Port 5000			
Flask Server			
http://<IP>:5000			
SSH ACCESS			
<table border="1"> <tr> <td>Port 22</td> </tr> <tr> <td>SSH Terminal</td> </tr> <tr> <td>ssh root@<IP></td> </tr> </table>	Port 22	SSH Terminal	ssh root@<IP>
Port 22			
SSH Terminal			
ssh root@<IP>			

📋 Quick Checklist

Pre-Deployment

- RDK X5 powered **on**
- Camera connected **to** MIPI
- Ethernet cable connected
- Router/**switch** accessible
- Desktop **on same network**
- SSH client installed (Windows/Mac/Linux)
- Code repository cloned **locally**

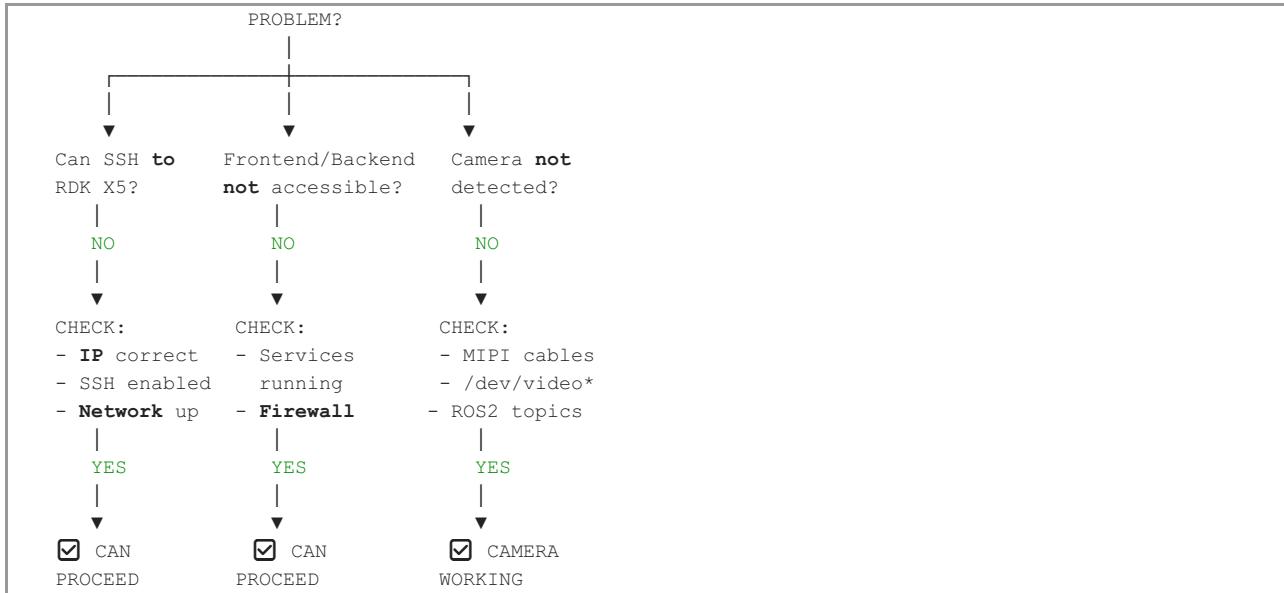
Deployment

- Found RDK X5 IP address
- SSH access verified
- Code copied to RDK X5
- Dependencies installed (Python + **Node**)
- Frontend** built
- Services **started**
- Services running **in** background

Post-Deployment

- Frontend loads (`http://<IP>:3000`)
- Backend responds (`http://<IP>:5000/api/health`)
- Can add students
- Live Scanner works
- Results displayed
- Data persists **in** history
- Remote access configured
- Security hardened
- Backups scheduled

🔧 Troubleshooting Flow Chart



⚡ 5-Minute Deployment Commands

```

# 1. Find IP
ssh root@192.168.1.100 # or check router

# 2. Deploy
ssh root@<IP> "mkdir -p /opt/weldvision"
scp -r ./code root@<IP>:/opt/weldvision/

# 3. Setup (inside RDK X5)
cd /opt/weldvision/backend
python3 -m venv venv && source venv/bin/activate
pip install -r requirements.txt

cd ..
apt-get install -y nodejs npm
npm install && npm run build

# 4. Start
cd backend && source venv/bin/activate && python3 app.py &
cd .. && npm run preview -- --host 0.0.0.0 &

# 5. Access
# Open: http://<RDK_X5_IP>:3000

```

🔒 Security Quick Setup

```

1. Change Password
passwd root

2. Enable Firewall
ufw enable
ufw allow 22/tcp
ufw allow 3000/tcp
ufw allow 5000/tcp

3. SSH Keys (Optional)
ssh-keygen -t ed25519
ssh-copy-id -i ~/.ssh/id_ed25519.pub root@<IP>

4. Update System
apt-get update && apt-get upgrade -y

```

📊 Performance Expectations

```

FRONTEND LOAD TIME
Local      (SSH):    <200ms [██████]
LAN        (WiFi):   <500ms [██████]
LAN        (Ethernet): <100ms [██████]
VPN        (Remote): <1s [██████████]

BACKEND RESPONSE
Scan Request: <500ms
API Health:   <100ms
Database:     <50ms

NETWORK BANDWIDTH
Live Feed:    ~3-5 Mbps
API Traffic: <1 Mbps
Total Req'd:  10 Mbps (good margin)

```

📁 Directory Structure After Deploy

```

/opt/weldvision/
├── backend/
│   ├── venv/           # Python environment
│   ├── app.py          # Flask server
│   ├── weld_data.db    # Database
│   └── requirements.txt
├── frontend/
│   ├── src/            # React source
│   ├── dist/           # Built app
│   └── node_modules/
│       └── package.json
└── docker-compose.yml  # Docker config
└── nginx.conf         # Reverse proxy

```

⌚ Access After Deploy

WELDVISION X5 DEPLOYED	
🌐 Frontend	<code>http://192.168.1.100:3000</code> <code>http://weldvision-x5.local:3000</code>
⚙️ Backend API	<code>http://192.168.1.100:5000</code> <code>http://weldvision-x5.local:5000</code>
💻 SSH Access	<code>ssh root@192.168.1.100</code> <code>ssh root@weldvision-x5.local</code>
📊 Status	Services: Running <input checked="" type="checkbox"/> Camera: Detected <input checked="" type="checkbox"/> Database: Ready <input checked="" type="checkbox"/>

💡 Next Steps After Deployment

- IMMEDIATE (< 5 min)
 - Open frontend in browser
 - Add a test student
 - Run Live Scanner capture

- SHORT TERM (< 1 hour)
 - Configure remote access (VPN/SSH)
 - Set up firewall rules
 - Enable auto-backup
 - Test from another computer

- MEDIUM TERM (< 1 day)
 - Security hardening
 - System updates
 - Performance optimization
 - Documentation review
 - Team training

- LONG TERM (ongoing)
 - Regular backups
 - Security updates
 - Performance monitoring
 - Maintenance tasks
 - Issue resolution

🎓 Which Guide to Read?

```
START HERE
|
└ Do you want...?
  |
  └ Fast deployment?
    └ QUICK_START_DEPLOYMENT.md (5 min read)
  |
  └ Complete details?
    └ COMPLETE_DEPLOYMENT_AND_REMOTE_ACCESS_GUIDE.md (30 min read)
  |
  └ Containerized setup?
    └ DOCKER_DEPLOYMENT_GUIDE.md (20 min read)
  |
  └ Navigation help?
    └ DEPLOYMENT_DOCUMENTATION_INDEX.md (5 min read)
```

📞 Quick Help

Problem? Search for it in **Troubleshooting** section of chosen guide

Stuck? Run `sudo journalctl -u weldvision-backend -f` to see what's happening

Network issue? Run `ping <RDK_X5_IP>` to test connectivity

Camera issue? Run `ls /dev/video*` to check detection

Service issue? Run `sudo systemctl status weldvision-*.service`

☑ Success Indicator

You'll know it's working when you see:

- Browser shows WeldVision X5 dashboard
- Can navigate **to** "Live Scanner"
- Can **select** a student
- Can click "Capture"
- Get** analysis results back
- Results appear **in** history
- Can **access from** another computer **on** network

READY TO DEPLOY? → Pick a guide above and get started! 🔒

All guides are tested, production-ready, and include troubleshooting.