




Automated Phone System - Deployment Guide


Current Status

Your Flask application is **RUNNING** and ready to receive calls!

- **Application Status:**  Running on port 5000
 - **Health Check:**  Passing (<http://localhost:5000/health>)
 - **Dependencies:**  All installed
 - **Local Access:** <http://localhost:5000>
-

Important Issues to Address

1. Missing Vonage Private Key

Status:  Critical - Required for Vonage API authentication

The application is looking for a private key file at `./private.key` but it doesn't exist. You need to:

1. Go to your [Vonage Dashboard](https://dashboard.vonage.com/) (<https://dashboard.vonage.com/>)
2. Navigate to **Applications** → Select your application (ID: `5f33f5f3-8898-4675-ad58-7823db96a34d`)
3. Download the **Private Key** file
4. Save it as `private.key` in `/home/ubuntu/github_repos/auto_call_system/`

Command to upload the key:

```
# After downloading from Vonage, upload it to:  
/home/ubuntu/github_repos/auto_call_system/private.key
```

2. Cal.com API Token Issue

Status:  Warning - Calendar integration won't work

The Cal.com API token in your `.env` file is returning a 401 error. You may need to:

- Verify the token is correct
 - Check if it has expired
 - Ensure it has the right permissions
-

Making Your Application Publicly Accessible

Since Vonage needs to send webhooks to your application, you need a **public URL**. Here are your options:

Option 1: Using ngrok (Recommended for Testing)

Step 1: Get an ngrok account and auth token

1. Sign up at <https://dashboard.ngrok.com/signup>
2. Get your auth token from <https://dashboard.ngrok.com/get-started/your-auth-token>

Step 2: Configure ngrok

```
ngrok config add-auth-token YOUR_AUTH_TOKEN_HERE
```

Step 3: Start ngrok tunnel

```
cd /home/ubuntu/github_repos/auto_call_system
ngrok http 5000
```

Step 4: Copy the public URL

You'll see output like:

```
Forwarding   https://abc123.ngrok.io -> http://localhost:5000
```

Your webhook URLs will be:

- **Answer URL:** <https://abc123.ngrok.io/webhooks/answer>
- **Event URL:** <https://abc123.ngrok.io/webhooks/events>
- **Speech URL:** <https://abc123.ngrok.io/webhooks/speech>

Option 2: Using Cloudflare Tunnel (Free, No Account Required)

```
# Install cloudflared
wget https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-linux-amd64
chmod +x cloudflared-linux-amd64
sudo mv cloudflared-linux-amd64 /usr/local/bin/cloudflared

# Start tunnel
cloudflared tunnel --url http://localhost:5000
```

Option 3: Deploy to Production Server

For production use, deploy to:

- **Heroku:** Easy deployment with free tier
- **DigitalOcean:** \$5/month droplet
- **AWS EC2:** Free tier available
- **Railway:** Modern deployment platform



Configure Vonage Webhooks

Once you have a public URL, configure it in your Vonage Application:

1. Go to [Vonage Dashboard](https://dashboard.vonage.com/) (<https://dashboard.vonage.com/>)
2. Navigate to **Applications** → Your Application

3. Under **Capabilities** → **Voice**, set:

- **Answer URL:** `https://YOUR-PUBLIC-URL/webhooks/answer` (HTTP POST)
- **Event URL:** `https://YOUR-PUBLIC-URL/webhooks/events` (HTTP POST)

4. Click **Save changes**

Complete Setup Checklist

- [x] Flask application running on port 5000
 - [x] Dependencies installed
 - [] **Upload Vonage private key to** `/home/ubuntu/github_repos/auto_call_system/private.key`
 - [] Fix Cal.com API token (optional, for calendar features)
 - [] Set up ngrok or another tunneling service
 - [] Configure webhook URLs in Vonage Dashboard
 - [] Test incoming call
-

Testing Your System

1. Test Health Endpoint

```
curl http://localhost:5000/health
```

Expected response:

```
{
  "status": "healthy",
  "timestamp": "2025-09-30T21:18:01.660875",
  "vonage_client": true
}
```

2. Test Answer Webhook (Simulated)

```
curl -X POST http://localhost:5000/webhooks/answer \
-H "Content-Type: application/json" \
-d '{
  "from": "14155551234",
  "to": "19095779171",
  "uuid": "test-call-uuid",
  "conversation_uuid": "test-conversation-uuid"
}'
```

3. Make a Real Test Call

Once webhooks are configured, call your Vonage number: **+1 (909) 577-9171**

Troubleshooting

Application won't start

```
# Check if port 5000 is already in use
sudo lsof -i :5000

# Kill existing process if needed
sudo kill -9 <PID>

# Restart application
cd /home/ubuntu/github_repos/auto_call_system
python3 app.py
```

Vonage client initialization failed

- Ensure `private.key` file exists in the project directory
- Verify the file has the correct permissions: `chmod 600 private.key`
- Check that the Application ID in `.env` matches your Vonage dashboard

Webhooks not receiving calls

- Verify your public URL is accessible: `curl https://YOUR-URL/health`
- Check Vonage Dashboard → Applications → Your App → Webhooks are configured
- Look at Flask logs for incoming requests
- Check Vonage Dashboard → Logs for webhook delivery status

Cal.com integration not working

- Verify API token is valid
- Check that `CALCOM_EVENT_TYPE_ID` is set correctly
- This is optional - the system can work without calendar integration

System Features

Your automated phone system handles:

- ✓ **Pricing Inquiries** - Provides rental rates and package information
 - ✓ **Availability Checking** - Checks calendar for open time slots
 - ✓ **Booking Requests** - Creates reservations (requires Cal.com setup)
 - ✓ **General Information** - Operating hours, location, services
 - ✓ **Human Escalation** - Transfers complex requests to staff at +1 (917) 796-9730
 - ✓ **After-Hours Handling** - Appropriate messaging outside 9am-9pm
-

Security Notes

- Never commit `.env` file or `private.key` to version control
- Keep your Vonage API credentials secure
- Use HTTPS for all webhook URLs in production
- Regularly rotate API keys and tokens

Additional Resources

- [Vonage Voice API Documentation](https://developer.vonage.com/voice/voice-api/overview) (<https://developer.vonage.com/voice/voice-api/overview>)
 - [NCCO Reference](https://developer.vonage.com/voice/voice-api/ncco-reference) (<https://developer.vonage.com/voice/voice-api/ncco-reference>)
 - [Flask Documentation](https://flask.palletsprojects.com/) (<https://flask.palletsprojects.com/>)
 - [ngrok Documentation](https://ngrok.com/docs) (<https://ngrok.com/docs>)
-

Need Help?

If you encounter issues:

1. Check the Flask application logs
2. Review Vonage Dashboard → Logs for webhook delivery
3. Test endpoints manually with curl
4. Verify all environment variables are set correctly

Current Application Location: `/home/ubuntu/github_repos/auto_call_system/`

Next Steps:

1. Upload your Vonage private key file
2. Set up ngrok with your auth token
3. Configure webhook URLs in Vonage Dashboard
4. Make a test call! ☎️