



# Google Cloud Deployment Status Report

**Date:** November 12, 2025

**Project:** amazon-ppc-474902

**Repository:** natureswaysoil/video

**Status:** ⚠️ BLOCKED - Insufficient Permissions



## Executive Summary

The deployment process has been **prepared and configured** but cannot proceed due to insufficient IAM permissions on the service account. All deployment scripts are ready and tested, authentication is successful, but the service account `video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com` lacks the necessary roles to:

- Enable required Google Cloud APIs
- Create Cloud Run jobs and services
- Deploy container images
- Set up Cloud Scheduler jobs
- Manage secrets



## Completed Steps

### 1. Service Account Key Configuration

- JSON key file created at `/home/ubuntu/service-account-key.json`
- File permissions set to 600 (secure)
- Service account authenticated successfully

### 2. GCP Authentication

- Active account: `video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com`
- Project configured: `amazon-ppc-474902`
- Region configured: `us-central1`
- Timezone: `America/New_York`

### 3. Deployment Scripts

All deployment scripts are ready and executable:

- `scripts/deploy-gcp.sh` (7.0K) - Main infrastructure deployment
- `scripts/deploy-blog-automation.sh` (3.7K) - Blog automation setup
- `scripts/verify-deployment.sh` (11K) - Deployment verification

### 4. Environment Configuration

Environment variables configured for deployment:

```
PROJECT_ID=amazon-ppc-474902
REGION=us-central1
TIME_ZONE=America/New_York
REPO_NAME=video-repo
IMAGE_NAME=video-app
JOB_NAME=video-processing-job
SCHED_NAME=video-scheduler-2x
SA_NAME=video-job-sa
SCHED_SA_NAME=scheduler-invoker
```

## Blocking Issues

### Permission Errors Encountered

The service account lacks permissions to:

#### 1. Enable Google Cloud APIs

```
Permission denied to enable service [artifactregistry.googleapis.com]
Permission denied to enable service [cloudbuild.googleapis.com]
Permission denied to enable service [cloudscheduler.googleapis.com]
Permission denied to enable service [run.googleapis.com]
Permission denied to enable service [secretmanager.googleapis.com]
```

#### 2. Access Project Resources

```
Permission 'run.jobs.list' denied on resource 'namespaces/amazon-ppc-474902/jobs'
Permission 'cloudscheduler.jobs.list' denied
The caller does not have permission to access projects
```

## Required IAM Roles

The service account **video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com** needs the following roles:

<b>Role</b>	<b>Purpose</b>	<b>Priority</b>
<code>roles/serviceusage.serviceUsageAdmin</code>	Enable required APIs	<b>Critical</b>
<code>roles/run.admin</code>	Create/manage Cloud Run jobs & services	<b>Critical</b>
<code>roles/cloudscheduler.admin</code>	Create/manage Cloud Scheduler jobs	<b>Critical</b>
<code>roles/artifactregistry.admin</code>	Create repositories & push images	<b>Critical</b>
<code>roles/cloud-build.builds.editor</code>	Build container images	<b>Critical</b>
<code>roles/secretmanager.admin</code>	Create/manage secrets	<b>Critical</b>
<code>roles/iam.serviceAccountUser</code>	Deploy with service accounts	<b>Critical</b>
<code>roles/resourcemanager.projectIamAdmin</code>	Grant permissions (optional)	Optional

## How to Fix - Grant Permissions

### Option 1: Google Cloud Console (Recommended for non-technical users)

**1. Navigate to IAM page:**

- Go to: <https://console.cloud.google.com/iam-admin/iam?project=amazon-ppc-474902>

**2. Find the service account:**

- Search for: `video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com`

**3. Edit permissions:**

- Click the pencil icon (Edit) next to the service account
- Click “ADD ANOTHER ROLE”
- Add each of the roles listed above
- Click “SAVE”

### Option 2: Using gcloud CLI (Requires Owner/Admin access)

Run these commands with an account that has **Owner** or **Project IAM Admin** role:

```
# Authenticate with an admin account first
gcloud auth login

# Set the project
gcloud config set project amazon-ppc-474902

# Grant all required roles
gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/serviceusage.serviceUsageAdmin"

gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/run.admin"

gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/cloudscheduler.admin"

gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/artifactregistry.admin"

gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/cloudbuild.builds.editor"

gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/secretmanager.admin"

gcloud projects add-iam-policy-binding amazon-ppc-474902 \
--member="serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com" \
--role="roles/iam.serviceAccountUser"
```

### Option 3: Terraform/IaC (For automated deployments)

```
resource "google_project_iam_member" "video_job_sa_roles" {
  for_each = toset([
    "roles/serviceusage.serviceUsageAdmin",
    "roles/run.admin",
    "roles/cloudscheduler.admin",
    "roles/artifactregistry.admin",
    "roles/cloudbuild.builds.editor",
    "roles/secretmanager.admin",
    "roles/iam.serviceAccountUser",
  ])
  project = "amazon-ppc-474902"
  role    = each.value
  member  = "serviceAccount:video-job-sa@amazon-ppc-474902.iam.gserviceaccount.com"
}
```



### Resume Deployment After Permissions Are Granted

Once the permissions are granted, run the following commands to complete the deployment:

```

# Navigate to the repository
cd /home/ubuntu/github_repos/video

# Ensure service account is active
gcloud auth activate-service-account --key-file=/home/ubuntu/service-account-key.json
gcloud config set project amazon-ppc-474902

# Run the main deployment
PROJECT_ID=amazon-ppc-474902 \
REGION=us-central1 \
TIME_ZONE=America/New_York \
bash scripts/deploy-gcp.sh

# Deploy blog automation
PROJECT_ID=amazon-ppc-474902 \
REGION=us-central1 \
bash scripts/deploy-blog-automation.sh

# Verify deployment
PROJECT_ID=amazon-ppc-474902 \
bash scripts/verify-deployment.sh

# Check deployment status
gcloud run jobs list --region=us-central1 --project=amazon-ppc-474902
gcloud scheduler jobs list --location=us-central1 --project=amazon-ppc-474902

```



## What Will Be Deployed

Once permissions are granted, the deployment will create:

### Cloud Run Jobs

- **video-processing-job**: Main video processing job
- Memory: 2Gi
- CPU: 1
- Timeout: 3600s (1 hour)
- Max retries: 3

### Cloud Run Services

- **blog-generator**: Blog automation service
- Automated blog post generation
- Integration with video processing

### Cloud Scheduler Jobs

- **video-scheduler-2x**: Scheduled video processing
- Schedule: `0 2 * * *` (2 AM daily)
- Timezone: America/New\_York
- **blog-automation-scheduler**: Scheduled blog generation
- Automated content creation

## Artifact Registry

- **video-repo:** Container image repository
- Location: us-central1
- Format: Docker

## Secret Manager

- API keys and credentials stored securely



## Expected Resources & Costs

### Estimated Monthly Costs (approximate)

Service	Usage	Estimated Cost
Cloud Run Jobs	2 executions/day @ 1hr	\$15-30/month
Cloud Scheduler	2 jobs	\$0.30/month
Artifact Registry	Storage	\$0.10/GB/month
Cloud Build	2 builds/day	Free tier
Secret Manager	10 secrets	\$0.06/month
<b>Total</b>		<b>~\$20-35/month</b>

Note: Costs vary based on actual usage and data transfer



## Verification Commands

After successful deployment, use these commands to verify:

```

# List all Cloud Run jobs
gcloud run jobs list --region=us-central1 --project=amazon-ppc-474902

# List all Cloud Run services
gcloud run services list --region=us-central1 --project=amazon-ppc-474902

# List Cloud Scheduler jobs
gcloud scheduler jobs list --location=us-central1 --project=amazon-ppc-474902

# Check Artifact Registry repositories
gcloud artifacts repositories list --location=us-central1 --project=amazon-ppc-474902

# List secrets
gcloud secrets list --project=amazon-ppc-474902

# View recent Cloud Run job executions
gcloud run jobs executions list --job=video-processing-job --region=us-central1 --project=amazon-ppc-474902

```



## Important Files & Locations

File/Directory	Location	Purpose
Service Account Key	/home/ubuntu/service-account-key.json	GCP authentication
Repository	/home/ubuntu/github_repos/video	Application code
Deployment Script	scripts/deploy-gcp.sh	Main deployment
Blog Automation	scripts/deploy-blog-automation.sh	Blog setup
Verification Script	scripts/verify-deployment.sh	Post-deployment checks
Deployment Logs	/tmp/deploy-gcp-final.log	Latest deployment attempt

## SOS Troubleshooting

### Issue: “Permission denied” errors

**Solution:** Grant the required IAM roles as described above

### Issue: “API not enabled”

**Solution:** The deployment script will enable APIs automatically once permissions are granted

## Issue: “Service account not found”

**Solution:** Verify the service account exists:

```
gcloud iam service-accounts describe video-job-sa@amazon-  
ppc-474902.iam.gserviceaccount.com
```

## Issue: “Region not available”

**Solution:** Check available regions:

```
gcloud run regions list
```

## 📞 Support & Resources

- **GCP IAM Documentation:** <https://cloud.google.com/iam/docs>
- **Cloud Run Documentation:** <https://cloud.google.com/run/docs>
- **Cloud Scheduler Documentation:** <https://cloud.google.com/scheduler/docs>
- **Service Account Best Practices:** <https://cloud.google.com/iam/docs/best-practices-service-accounts>

## ✓ Next Actions Required

1. **IMMEDIATE:** Grant IAM permissions to the service account (see “How to Fix” section above)
2. **AFTER PERMISSIONS:** Re-run deployment scripts
3. **VERIFY:** Check all resources are created successfully
4. **MONITOR:** Set up logging and monitoring for production use
5. **SECURE:** Review and audit service account permissions regularly

**Generated:** November 12, 2025

**Repository:** /home/ubuntu/github\_repos/video

**Status Document:** /home/ubuntu/github\_repos/video/DEPLOYMENT\_STATUS.md