**Sandbox Environment Setup & Deployment Documentation**

This document provides the step-by-step process to set up and deploy services into the **khoros-playvox-sandbox** project on GCP. It includes database migration, application build and containerization, deployment to Cloud Run, IAM configuration, and automation setup.

1. **Cloud SQL Setup**

A screen shot of a computer

AI-generated content may be incorrect.

**Instance Information:**

*NAME: khoros-playvox-sandbox*

*DATABASE\_VERSION: SQLSERVER\_2019\_STANDARD*

*LOCATION: us-central1-a*

*TIER: db-custom-2-8192*

*PRIMARY\_ADDRESS: 104.154.98.22*

*STATUS: RUNNABLE*

**Steps:**

1. Connect to SQL instance:
2. sqlcmd -S 104.154.98.22,1433 -U sqlserver -P <PASSWORD>
3. Create a Sandbox bucket:
4. gcloud storage buckets create gs://khoros-playvox-prod-backup \

--project=khoros-playvox-sandbox \

--location=us-central1

1. Export Prod database BAK file to storage bucket:

gcloud sql export bak khoros-playvox-prod \ gs://khoros-playvox-sandbox-backup/khoros-prod-backup.bak \ --database=khoros \ --project=sgs-playvox

1. Copy production backup to sandbox bucket:

gcloud storage cp gs://khoros-playvox-sandbox-backup/khoros-prod-backup.bak \

gs://khoros-playvox-prod-backup/

1. Get IAM access to Cloud SQL service account:

gcloud sql instances describe khoros-playvox-sandbox \

--format="value(serviceAccountEmailAddress)"

1. Grant IAM access to Cloud SQL service account:

gcloud storage buckets add-iam-policy-binding gs://khoros-playvox-prod-backup \

--member="serviceAccount:<SERVICE\_ACCOUNT>" \

--role="roles/storage.objectViewer"

1. Import BAK file:

gcloud sql import bak khoros-playvox-sandbox \

gs://khoros-playvox-prod-backup/khoros-prod-backup.bak \

--database=khoros\_sandbox \

--project=khoros-playvox-sandbox

1. Verify database:

gcloud sql databases list --instance=khoros-playvox-sandbox

sqlcmd -S <INSTANCE\_IP> -U <USERNAME> -P <PASSWORD> -d khoros\_sandbox

1. **Application Build & Containerization**

**Environment Setup:**

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

export PATH=$JAVA\_HOME/bin:$PATH

java -version

**Build & Containerize:**

1. Update build.gradle:
   * Upgrade **Spring Boot** from 2.7.11 → 2.7.18
   * Ensure **Tomcat** version compatibility
   * Create application-sandbox.properties

A screenshot of a computer screen

AI-generated content may be incorrect.

1. Build and create image:

./gradlew clean build

./gradlew bootBuildImage \

--imageName=us-central1-docker.pkg.dev/khoros-playvox-sandbox/sandbox-repo/<service-name>:<tag>

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

3.Push image to Artifact Registry:

docker push us-central1-docker.pkg.dev/khoros-playvox-sandbox/sandbox-repo/<service-name>:<tag>

A screen shot of a computer

AI-generated content may be incorrect.

1. **Cloud Run Deployment**

**Deploy service:**

gcloud run deploy <service-name>-sandbox \

--image=us-central1-docker.pkg.dev/khoros-playvox-sandbox/sandbox-repo/<service-name>:<tag> \

--platform=managed \

--region=us-central1 \

--allow-unauthenticated \

--set-env-vars SPRING\_PROFILES\_ACTIVE=sandbox \

--memory=1Gi \

--port=<service-port>

**Check logs:**

gcloud run services logs read <service-name>-sandbox \

--project=khoros-playvox-sandbox \

--region=us-central1 \

--limit=100

**Verify service account:**

gcloud run services describe <service-name>-sandbox \

--project=khoros-playvox-sandbox \

--region=us-central1 \

--format="value(spec.template.spec.serviceAccountName)"

**Grant Cloud SQL access:**

gcloud projects add-iam-policy-binding khoros-playvox-sandbox \

--member="serviceAccount:493845609335-compute@developer.gserviceaccount.com" \

--role="roles/cloudsql.client"

1. **Services Deployed**

| **Service** | **Port** | **Image Tag** | **Status** |
| --- | --- | --- | --- |
| author-service | 8080 | 0.0.1 | ✅ Deployed |
| actionlog-service | 8084 | 0.0.1 | ✅ Deployed |
| batchprocessing | 8086 | 0.0.1 | ✅ Deployed |
| email-service | 8090 | 0.0.1 | ✅ Deployed |
| monitor-service | 8092 | 0.0.1 | ✅ Deployed |
| playvoxbatch | 8088 | 0.0.1 | ✅ Deployed |
| table-cleanup | 8094 | 0.0.1 | ✅ Deployed |

1. **Source Repositories**

Sandbox repositories created under **khoros-playvox-sandbox**:

* author-sandbox
* actionlog-sandbox
* batchProcessing-sandbox
* email-sandbox
* monitor-sandbox
* playvoxBatch-sandbox
* tableDeletion-sandbox

Sample repo push:

* *git init*
* *git remote add sandbox https://source.developers.google.com/p/khoros-playvox-sandbox/r/author-sandbox*
* *git add .*
* *git commit -m "Initial commit for sandbox repo"*
* *git push sandbox main*

1. **Cloud Build & IAM Setup**

Enable required services:

*gcloud services enable cloudbuild.googleapis.com \*

*run.googleapis.com \*

*artifactregistry.googleapis.com \*

*sourcerepo.googleapis.com*

Create Cloud Build Services:

*gcloud builds submit \*

*--project=khoros-playvox-sandbox \*

*--config=cloudbuild.yaml*

A screen shot of a computer

AI-generated content may be incorrect.

1. **Cloud Build CI/CD Pipeline**

Create a cloudbuild.yaml:

*steps:*

*- name: 'gcr.io/cloud-builders/docker'*

*args: ['build', '-t', 'us-central1-docker.pkg.dev/sandbox-proj/cloud-run-services/author-service:$SHORT\_SHA', '.']*

*- name: 'gcr.io/cloud-builders/docker'*

*args: ['push', 'us-central1-docker.pkg.dev/sandbox-proj/cloud-run-services/author-service:$SHORT\_SHA']*

*- name: 'gcr.io/cloud-builders/gcloud'*

*args: [*

*'run', 'deploy', 'author-service',*

*'--image', 'us-central1-docker.pkg.dev/sandbox-proj/cloud-run-services/author-service:$SHORT\_SHA',*

*'--region', 'us-central1',*

*'--platform', 'managed'*

*]*

Trigger CI/CD on every push:

*gcloud beta builds triggers create cloud-source-repositories \*

*--repo="author-service-repo" \*

*--branch-pattern=".\*" \*

*--build-config="cloudbuild.yaml"*

A screen shot of a computer

AI-generated content may be incorrect.

1. **Cloud Scheduler Setup**

**Job: author-job**

gcloud scheduler jobs create http author-job \

--schedule="\*/5 \* \* \* \*" \

--uri="https://author-service-sandbox-65maxlhe6a-uc.a.run.app/khoros/author/" \

--http-method=GET \

--oidc-service-account-email="scheduler-sa@khoros-playvox-sandbox.iam.gserviceaccount.com" \

--oidc-token-audience="https://author-service-sandbox-65maxlhe6a-uc.a.run.app/khoros/author/" \

--location=us-central1

A screenshot of a computer

AI-generated content may be incorrect.

**Job Config:**

* Schedule: Every 5 mins
* Timezone: UTC
* Auth: OIDC with service account

1. **Monitoring & Logs**

* Use **Cloud Run logs** for service-level monitoring.
* Use **Cloud SQL Insights** for DB monitoring.
* Use **Cloud Scheduler logs** for job execution tracking.

Sandbox environment is fully configured with SQL, Cloud Run services, Source Repos, IAM, and Scheduler.