

Xalott - manual deploy for frontend apps

 Link:: [Xalott vidneo](#)

 Tags:: [#output/techdocs](#)

Xalott - manual deploy for frontend apps

Xalott frontend apps are hosted as SPA in [AWS S3](#) buckets. Each project and environment have a separate bucket for project files.

Configure AWS CLI

To interact with AWS services, `aws cli` tool is required and need to be configured with user credentials that have permissions configured in [AWS IAM](#).

This process need to be done once per device you want to use to interact with [AWS](#) services.

Use separate profiles

Good practice is to not configure AWS CLI with global profile. Use separate, per-project profiles instead.

It will require `--profile {PROFILE_NAME}` parameter in each command you use, so no mistakes when you have multiple account and projects.

To configure CLI, just run:

```
aws configure --profile xala
```

It will ask about:

- AWS Access Key ID
- AWS Secret Access Key
- Default region name – set it to `None`

Access Keys can be generated in AWS Management Console in logged-in user profile.

Deployment checklist

- ☐ [Build application](#)
- ☐ [Sync build folder with AWS S3 bucket](#)
- ☐ [Invalidate S3 CloudFront cache for AWS S3 bucket](#)

Build application

Each frontend app uses `.env` files with configuration per environment. Each `.env` file have configured `S3_BUCKET_NAME` that have unique name of [AWS S3](#) bucket used to deploy application.

Env name	Env file name	Build command
Develop	<code>.env.dev</code>	<code>yarn build:dev</code>
Staging	<code>.env.staging</code>	<code>yarn build:staging</code>
Test	<code>.env.test</code>	<code>yarn build:test</code>
Prod	<code>.env.prod</code>	<code>yarn build:prod</code>

Go to base monorepo directory and load node in same version as used in project (`.nvmrc`). If you use `nvm` utility, then it should be loaded automatically when you enter project directory.

Project	Base monorepo directory	App directory
Web	<code>vidneo-xala-app</code>	<code>packages/web</code>
Studio	<code>vidneo-xala-studio</code>	<code>packages/scmtool</code>

Go to app directory and load deploying env file to your environment using command:

```
export $(cat {ENV_FILE_NAME} | xargs)
```

Check if correct `S3_BUCKET_NAME` is loaded:

```
echo $S3_BUCKET_NAME
```

It should print the same name as is set in the environment file. This env variable will be used in future deploy commands.

Run build command:

```
yarn build:{ENV_NAME}
```

Sync build folder with AWS S3 bucket

When the application is built, it's time to upload build files to AWS S3 bucket. Go to build directory in app directory:

```
cd build
```

Run sync command:

```
aws s3 sync . s3://$S3_BUCKET_NAME --profile xala
```

⚠ Be in right place

Make sure you are in correct directory before running sync command. This command will upload all files from current directory to [AWS S3](#) bucket.

Invalidate S3 CloudFront cache for AWS S3 bucket

Application is delivered by [CDN](#) using [AWS CloudFront](#) service. When a new version is deployed, [CDN](#) need to be informed that there are changes.

Run cache invalidation command:

```
aws cloudfront create-invalidation --  
distribution-id {DISTRIBUTION-ID} --paths "/*" --  
profile xala
```

Web app:

Env	Domain	Distribution ID
dev	dev-xalott.xal-vidneo.cat	E343APVPRYMO2G
stage	web-xalott-staging.xal-vidneo.cat	EQW2NJPEGXBZV
test	test-xalott.xal-vidneo.cat	E1VY5A2PNP6MH2
prod	laxarxames.cat	E1B78ZXAFUFGHE

Studio:

Env	Domain	Distribution ID
dev	dev-studio.xal-vidneo.cat	E3T810DQLBYD2I
stage	media-xalott-staging.xal-vidneo.cat	E2B2BS1EXEKF31
test	test-studio.xal-vidneo.cat	E4MASGW0KXXW8
prod	studio.laxarxames.cat	E39HYIEENJU9Z9