



SASjs

DevOps for SAS



Challenges with Traditional SAS Projects

- Scattered Artefacts > *Where is that job?*
- Shared Environment > *Who changed that macro?!*
- Single Server > *How will we get this into production?*



The absence of a ***standardised framework*** makes for ***inconsistent project delivery*** and ***on-boarding struggles*** for new developers



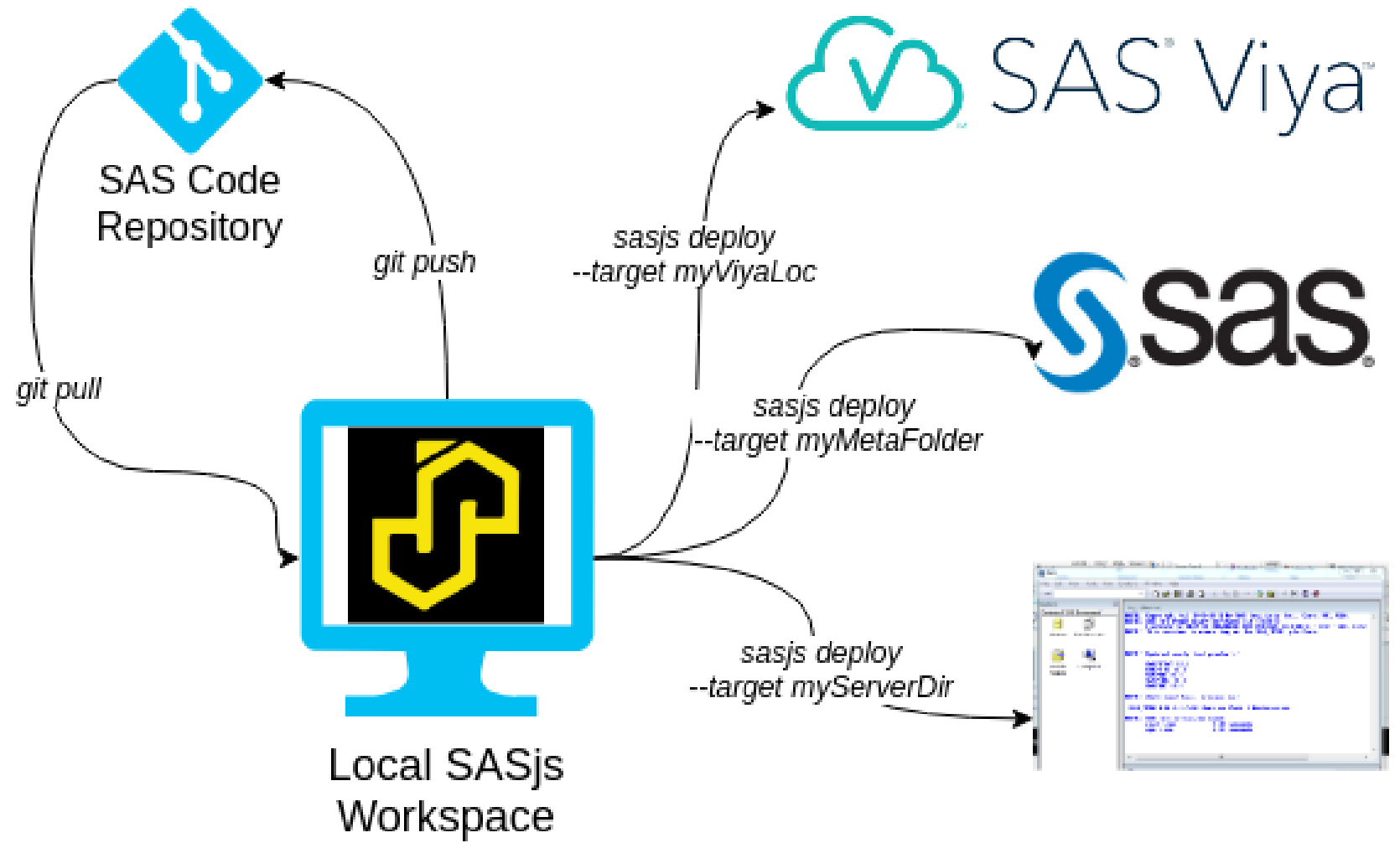
Why use SASjs?

- Open Source (MIT) & Actively Developed
- Faster iterations / time to value
- Work locally with your preferred tools

What is SASjs?

1. An Opinionated Approach for SAS Solution Delivery
2. A Collection of Tools
 - [@sasjs/cli](#) - CI/CD and Automated Deployment
 - [@sasjs/core](#) - Macro library
 - [@sasjs/vscode-extension](#) - Battery Pack for VS Code
 - [@sasjs/server](#) - Build Apps on Base SAS

SASjs Workflow



SASjs Projects

- Push to Any or Many > *Server Agnostic*
- Centralised Artefacts > *Source Controlled*
- Isolated Developer Environments > *Move Fast and Break Things*

✓ Highly suitable where SAS Programming is involved

✗ Needs additional work for graphical artefacts (VA, Flows etc)

Why Do SAS Admins ❤️ It?

- Nothing to install or provision on the SAS server
- No SSH nor special permissions needed anywhere
- Nothing to configure, except:
 - Optional: Access to the target folder
 - Optional: Client / Secret (Viya)
 - Optional: SAS 9 setting for encoded passwords

Why Do Project Leads ❤️ It?

- Faster to on-board new developers
- Ability to work on multiple features in parallel
- Complete visibility of what is being developed

SASjs Config File

Every SASjs Project has a `sasjs/sasjsconfig.json` file, for configuration of:

- Location of Artefacts (+ Dependencies & Init/Term programs)
- Macro Variables
- Target attributes / connection settings

SASjs Target

A location on a SAS server

Core attributes:

- `name` > Alias for SASjs commands, eg: `sasjs deploy -t dev3`
- `serverUrl` > Protocol + Host + Port
- `serverType` > either `SASVIYA` , `SAS9` or `SASJS`
- `appLoc` > Root deployment folder in SAS Drive or Metadata

Running SAS

- `sasjs run` > *Execute arbitrary SAS code*
- `sasjs job execute` > *Run a Job*
- `sasjs flow execute` > *Run a collection of Jobs*

But - how to run a remote job with local macros???

Compilation - Scaffolding Consistency

```
@param access_token_var= The glo
@param grant_type= valid values
| The default is "detect" (which

@version VIYA V.03.04
@author Allan Bowe, source: http

<h4> SAS Macros </h4>
@li mp_abort.sas
@li mf_getplatform.sas
@li mf_getuniquefileref.sas
@li mf_getuniquelibref.sas
@li mf_isblank.sas

**/
```



```
},
"jobConfig": {
  "jobFolders": [
    "../jobs/land",
    "../jobs/publish",
    "../jobs/stage",
    "../jobs/wrapped"
  ],
  "initProgram": "../jobs/jobin
  "termProgram": "../jobs/jobte
},
"programFolders": [
  "../db/stg",
  "../db/pub",
  "../db/r\nd",
  "../db/rpub"
],
"serviceConfig": {
```



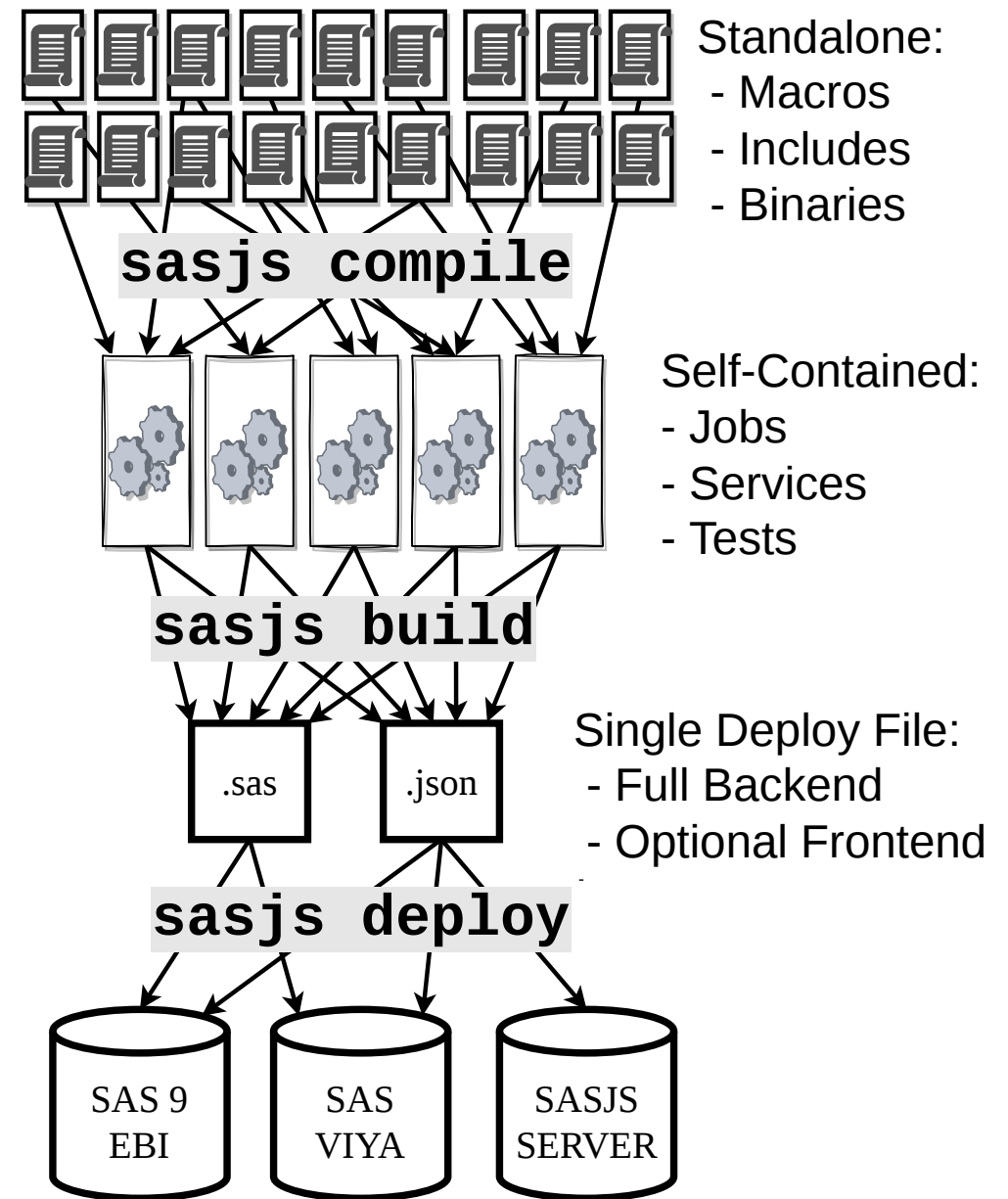
Compilation Components

Targets are compiled by reference to the dependencies listed in the program header.

- Artefacts: **Jobs, Services, Tests**
- Dependencies: **Macros, Includes, Binary Files**
- Optional Frontend - convert a web project into a [streaming app](#).

sasjs cbd

- **sasjs compile** > *Self-contained Jobs, Services & Tests*
- **sasjs build** > *Deployment pack (JSON / SAS Program)*
- **sasjs deploy** > *Send to target location*






Being a Responsible Developer

- `sasjs test` > *Trigger one or more Tests*
- `sasjs lint` > *Check Code Quality / Best Practices*
- `sasjs doc` > *Generate docs (+ lineage) from program headers*

More commands at <https://cli.sasjs.io>

CLI Pre-Requisites

Required	Recommended	Recommended
		

To install the CLI: `npm i -g @sasjs/cli`



Demo

```
sasjs lint
```

```
sasjs doc
```

```
sasjs auth
```

```
sasjs cbd
```

```
sasjs test
```

```
sasjs run
```

```
sasjs job execute
```



Resources

- <https://sasjs.io/resources>
- <https://datacontroller.io>
- <https://github.com/sasjs>
- <https://cli.sasjs.io>
- <https://sasapps.io>