#argocon

March 19th, 2024

Argo CD
Plugins as services

Presented by **Alan Clucas - Pipekit**

Who

pipekit

Alan Clucas

Senior Software Engineer
@ **Pipekit.io**

- Argo Workflows maintainer
- Co-author argocd-lovely-plugin
- Professional services

Config Management Plugins?

- Take the contents of your git repository
- Morph this into Kubernetes YAML
  - Helm
  - Hera
- Accessed:
  - By name
  - Discovery: *“Yes, I can handle that”*
- Configured using `plugin.yaml`

Plugin installation is non-trivial

As a plugin author, I'd like people to be able to **play with my plugin easily**, and **uninstall it easily**.

Just install this:

Helm chart **Kustomize**

Improves understandability

Register the plugin sidecar

To install a plugin, patch argocd-repo-server to run the plugin container as a sidecar, with argocd-cmp-server as its entrypoint. You can use either off-the-shelf or custom-built plugin image as sidecar image. For example:

```
containers:
- name: my-plugin
  command: [/var/run/argocd/argocd-cmp-server] # Entrypoint should be Argo CD light
  image: busybox # This can be off-the-shelf or custom-built image
  securityContext:
    runAsNonRoot: true
    runAsUser: 999
  volumeMounts:
    - mountPath: /var/run/argocd
      name: var-files
    - mountPath: /home/argocd/cmp-server/plugins
      name: plugins
    # Remove this volumeMount if you've chosen to bake the config file into the sid
    - mountPath: /home/argocd/cmp-server/config/plugin.yaml
      subPath: plugin.yaml
      name: my-plugin-config
    # Starting with v2.4, do NOT mount the same tmp volume as the repo-server conta
    # mitigate path traversal attacks.
    - mountPath: /tmp
      name: cmp-tmp
volumes:
- configMap:
    name: my-plugin-config
  name: my-plugin-config
- emptyDir: {}
  name: cmp-tmp
```

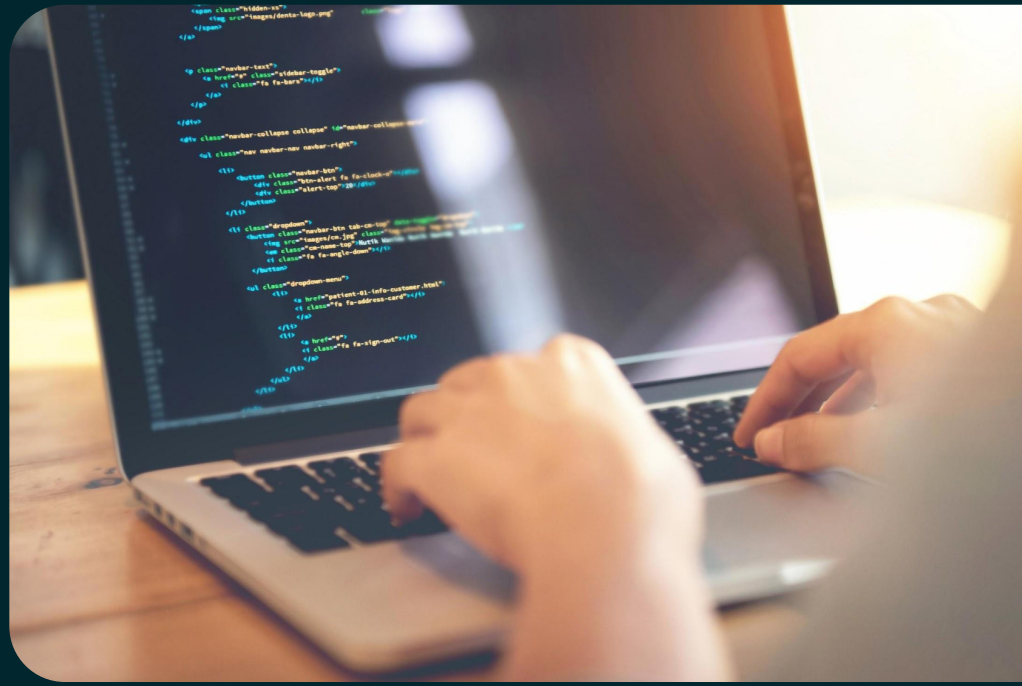
Plugin development

Kubernetes and **Argo** ecosystem is building blocks

Drive adoption of plugins

Hard to develop without admin

Iteration is slow



So...

Sidecar plugins

gRPC (network) protocol already

Service instead

Downsides

Transfer repository to the service will take longer

- Large repos aren't good for ArgoCD anyway
- "Discovery" will be slower for the same reason

Additional RBAC for repo-server

- Needs `get`, `list`, `watch` on `Services`

Upsides

Scalable independently

- HPA can be used
- Multiple plugins

Run plugins on spot nodes

Separated lifecycle

Upsides

More of a normal model

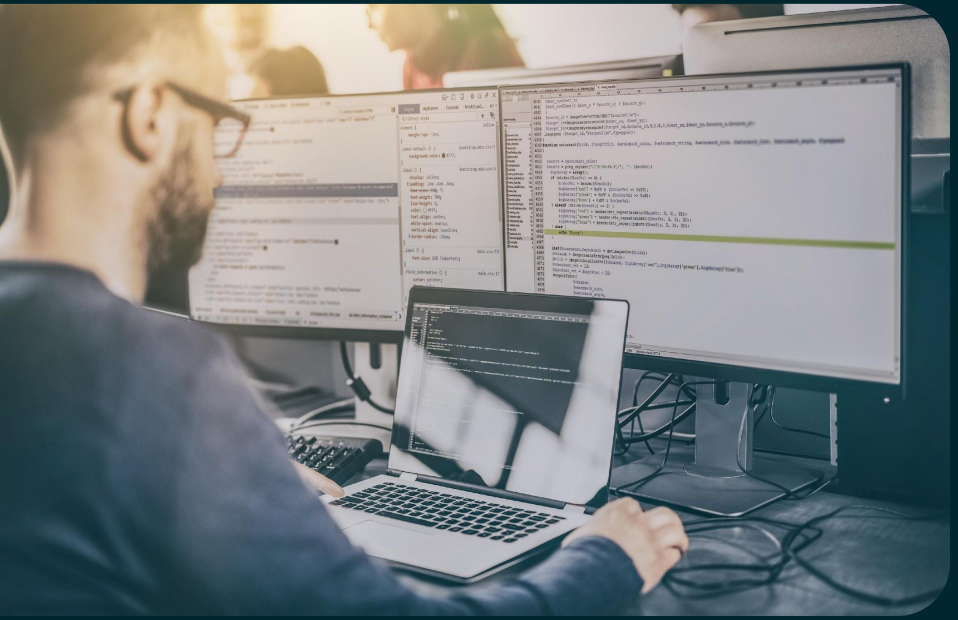
- Could have it's own monitoring

Failure to pull doesn't bring down the repo server

Plugin development is much easier

- Because of separation of concerns
- ServiceAccount
- Stateful plugins

argocd-lovely-plugin



Originally conceived to
"**do what you'd hope**"

Helm + Kustomize

Grown into a bit like a unix pipe
joining together transformations

plugin.yaml

```yaml
apiversion: argoproj.io/v1alpha1
kind: ConfigManagementPlugin
metadata:
  name: lovely
spec:
  version: ""
  generate:
     command:
     - argocd-lovely-plugin
  parameters:
    ...
  listenAddress: 0.0.0.0:8080
```

Demo

Convert a sidecar?

No code changes

Dockerfile

```
FROM lucj/argocd-plugin-helmfile:latest
USER 0
RUN echo "  listenAddress: 0.0.0.0:8080" >>
/home/argocd/cmp-server/config/plugin.yaml
USER 999
COPY --from=docker.io/joibel/argocd:latest /usr/local/bin/argocd-cmp-server
/var/run/argocd/argocd-cmp-server
```

Where next?

Pull request #16852

Implement "native" as separate plugins?

Helm **Kustomize** **Jsonnet**

Golang pkgs for building integrated plugins?

About Pipekit

- **Argo experts & maintainers**
- Save engineering time and **up to 60%** on compute costs
- Add a team of **3 Argo maintainers** and **5 Argo contributors** to your Slack
- Serving **startups & enterprises** since 2021:
  - **Enterprise Support for Argo**
    → Ideal for Platform Eng teams scaling with Argo
  - **Control Plane for Argo Workflows**
    → Ideal for Data teams who don't speak K8s OR multi-cluster setups

Pipekit

ARGO/INFRASTRUCTURE ASSISTANCE & ADVICE

- Booth E34
- Regular Office Hours (pipekit.io/office-hours)