Continuous documentation

Table of Contents

1. About this repo

Use this repo to generate HTML-with-PDF documentation for arbitrary collections of Asciidoc and Markdown files.

Conversion is handled by the separate image gerald1248/asciidoctor, which derives from the upstream image asciidoctor/asciidoctor, adding `pandoc` and charting plugins.

To document your own projects, consider including your own Git repositories as git modules in subfolders or simply copy the contents of this folder to the root folder of your repository, taking care not to overwrite files you care about.

The generated HTML page consolidates all images in a single `images` folder and ends with a download link to the PDF.

1.1. Getting started

Start by adjusting the file `values.json`. It contains the `title` of your project as it will appear at the top of the HTML output and on the title page of the PDF version. Another key is `filename`, which determines the name of the PDF outputl. (The web page always takes `index.html`.)

Configuration

```
{
  "title": "Continuous documentation",
  "filename": "continuous-documentation",
  "substitutions": { ①
    "Fibonacci": "Iccanobif"
  }
}
```

① This lookup table helps with cross-references and resource paths that would otherwise not be linked correctly. Keys and values represent `find` and `replace` expressions.

1.2. Build

```
$ make [build]
```

1.3. TEST-CONTENT

1.4. k8s-analysis

Analysis pod for debugging: deploy within your project and run `ab`, `dig`, `netstat`, `nslookup`, `telnet`, `traceroute`, `wget`, and so on as well as database clients for MySQL and PostgreSQL.

1.4.1. Install

```
$ make install
helm install --name=k8s-analysis .
NAME:   k8s-analysis
LAST DEPLOYED: Mon Oct  1 00:12:28 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Deployment
NAME          AGE
k8s-analysis  0s

==> v1/Pod(related)

NAME                           READY  STATUS             RESTARTS  AGE
k8s-analysis-6d76cfddb5-7s42l  0/1    ContainerCreating  0         0s
$ kubectl get po
NAME                                  READY     STATUS      RESTARTS   AGE
k8s-analysis-6d76cfddb5-7s42l         1/1       Running     0          1m
$ kubectl exec -it k8s-analysis-6d76cfddb5-7s42l -- /bin/sh
/app $
```

1.4.2. Cleanup

```
$ make delete
helm delete --purge k8s-analysis
release "k8s-analysis" deleted
```

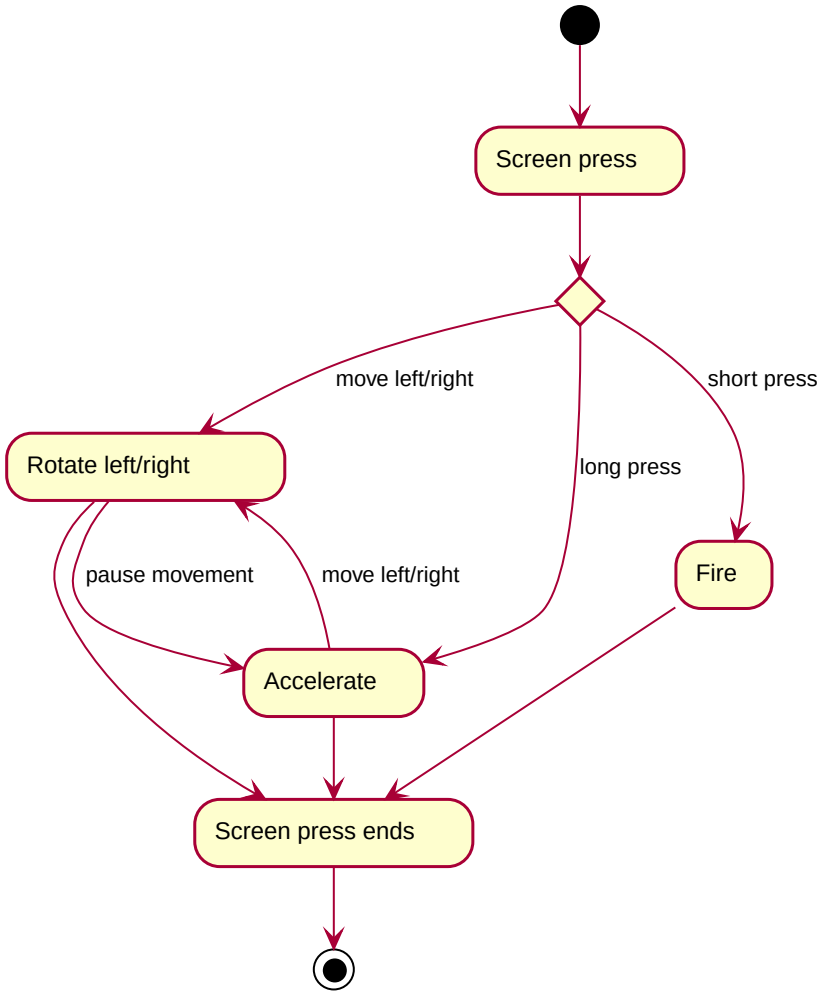


Figure 1. State machine

1.4.3. Creating and assigning a cluster role

To create a cluster role for our service account, apply the following cluster role definition:

Cluster role definition

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: cluster-reader
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["get", "watch", "list"]
- apiGroups: [""]
  resources: ["pods/exec"]
  verbs: ["create"]
- apiGroups: ["rbac.authorization.k8s.io"] ①
  resources:
  - rolebindings
  - clusterrolebindings
  verbs:
  - get
  - watch
  - list
```

① Requests for RBAC-related data will fail if this group is not set.

1.5. The Great American Novel

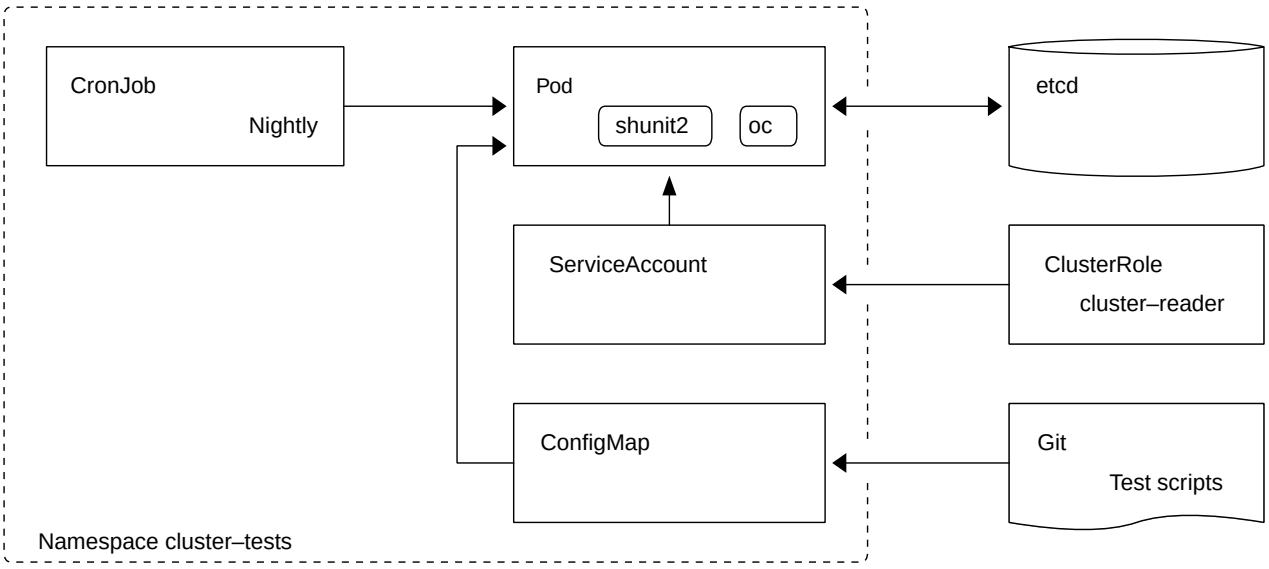


Figure 2. ASCII-based artwork

This looks important, so be sure not to forget it.

1.5.1. MARKDOWN

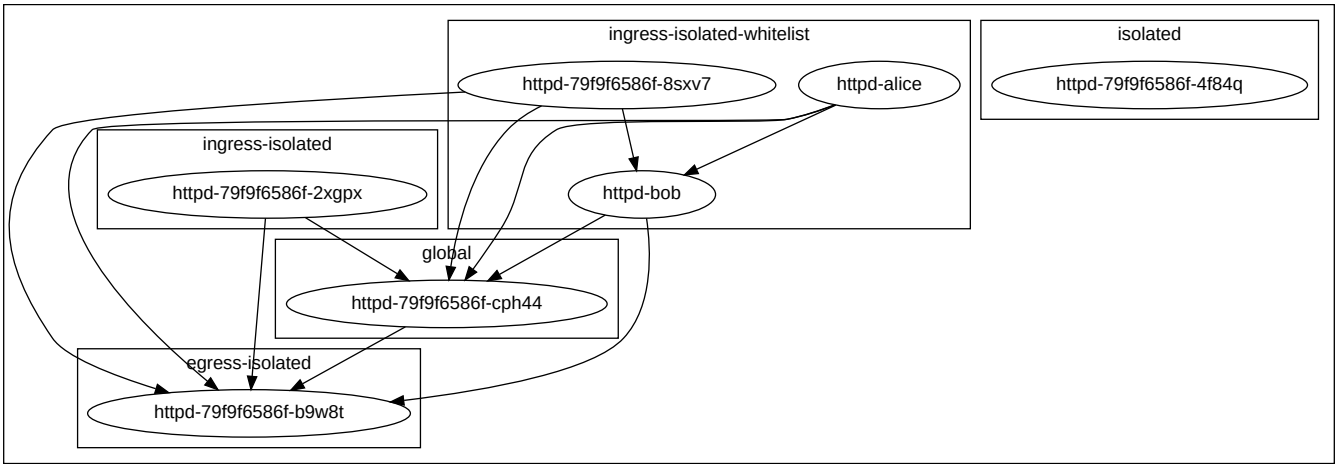
Sample Markdown document

In HTML the following should appear as an emoji: .

1.5.2. SAMPLE-REPO

k8s-network-policy-viewer

The network policy viewer visualizes the pod network. It is far from complete, but basic isolation rules can be represented in JSON, YAML or dot (Graphviz).



In this example, the names of the namespaces match their respective network policies, the exception being the `global` namespace (which has none) and `ingress-isolated-whitelist` (which has two).

The policies `isolated`, `egress-isolated`, `ingress-isolated` each apply to the namespace as a whole.

`ingress-isolated-whitelist` whitelists `httpd-bob`, which is why `httpd-bob` can be reached from `httpd-alice` and the generic httpd pod in the namespace.

Deployment

Install the helm chart defined in the folder chart:

```
$ make -C chart install
```

Point your browser to the URL given in `values.yaml` (e.g. `http://minikube.info/`).

The available endpoints are:

Endpoint	Description
/	Show graph
/health	Health endpoint
/api/v1/metrics	Metrics endpoint

Build

The build steps are the following:

```
$ go mod download
$ go get
$ go vet
$ go test -v
$ go build -o k8s-network-policy-viewer .
```

`make build` will run these steps in a two-stage docker build process.

Alternatively, you can use the default image k8s-network-policy-viewer. This is also the image referenced in the helm chart.

Testdata

To build the sample data, run:

```
$ make -C testdata init
$ make -C testdata create
```

2. Get the PDF

Download continuous-documentation.pdf