Kubernetes and OpenShift made simple with Kedge

Simplified app definition and workflow for developers

kedgeproject.org
github.com/kedgeproject/kedge

Tomas Kral <tkral@redhat.com>
Kubernetes definition for simple application

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
labels:
  app: nginx
spec:
  replicas: 3
selector:
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
spec:
  containers:  
    - name: nginx
      image: nginx:1.7.9

apiVersion: v1
kind: Service
metadata:
  name: nginx
labels:
  app: nginx
spec:
  selector:
    app: nginx
  ports:
    - port: 80

apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx
labels:
  app: nginx
spec:
  rules:
    - host: web.example.com
      http:
        paths:
          - backend: 
            serviceName: nginx
            servicePort: 80
            path: /
```
What is the problem?

- Kubernetes YAML definitions are not user/developer friendly
- Complicated YAML structure
- Some redundant values and items
- Write everything manually
- Kubernetes resources are designed for API and they are computer friendly not human friendly
Kedge equivalent

name: nginx

deployments:
  - replicas: 3

containers:
  - image: nginx:1.7.9

services:
  - ports:
    - port: 80
  endpoint: web.example.com
Where Kedge shines

- Less complicated and straightforward structure
- Smart defaults and best practices
- Use environment variables for parameterization
- LSP (Language Server Protocol) support
- Define only stuff that you need to define
- You don’t have to repeat yourself
Adding one common label to my whole application?

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
labels:
  app: nginx
  version: 0.1.3
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
template:
  metadata:
    labels:
      app: nginx
spec:
  containers:
    - name: nginx
      image: nginx:1.7.9
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
labels:
  app: nginx
spec:
  selector:
    app: nginx
  ports:
    - port: 80
```

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx
labels:
  app: nginx
spec:
  rules:
    - host: web.example.com
      http:
        paths:
          - path: /
            backend:
              serviceName: nginx
              servicePort: 80
            path: /
```
Easy with Kedge

name: nginx
labels:
  version: 0.1.3

deployments:
- replicas: 3
containers:
  - image: nginx:1.7.9

services:
- ports:
  - port: 80
    endpoint: web.example.com
Demo time

PLEASE,

LET THIS WORK...
Kedge has more features than this

- Full OpenShift support
  - S2I builds
  - BuildConfigs
  - DeploymentConfigs
  - Routes
  - ImageStreams
- Shortcuts for Ingress, Ports and Health Checks
- Builds (s2i in OpenShift cluster or local docker builds)
- Bootstrapping new kedge files using `kedge init`
Roadmap and plans for future

- More shortcuts
  - Triggers for OpenShift DeploymentConfigs
  - Environment variables (valueFrom)
- Smart builds (autodetect as much as possible)
- Automatic file bootstrapping (`kedge init`)
Thank you

github.com/kedgeproject/kedge

kedgeproject.org

PRs, Issues or any suggestions are welcome and appreciated