YAML Required

- apiVersion : version of the k8s API used to create the object
- kind: kind of object you want to create
- metadata: data that helps uniquely identify the object (name, UID, namespace)
- spec: what state you desire for the object

YAML Example

```yaml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - name: nginx
      image: nginx:1.14.2
      ports:
      - containerPort: 80
```

YAML Maps and lists

- Look [here](#)
- Indentation
  - be consistent
  - I usually use 2 spaces, but the number doesn’t matter as long as you are consistent.
  - DON'T USE TABS

Next example

```yaml
...
apiVersion: v1 #deployments dont exist in v1
kind: Pod      #what we are creating, service,Deployment, etc...
metadata:
  name: rss-site
labels:
  app: web
spec:       #define actual objects within the pod
  containers:
  - name: front-end
    image: nginx
    ports:
      - containerPort: 80
  - name: rss-reader
    image: nickchase/rss-php-nginx:v1
    ports:
      - containerPort: 88
```
**How to spec a pod**

Maybe see the pod spec [here](#)

### Container spec

- `kubectl exec --stdin --tty rss-site -- /bin/bash`
- If a pod has multiple containers
  - `kubectl exec -i -t my-pod --container main-app -- /bin/bash`
- Pass other commands in:
  - `kubectl exec --stdin --tty rss-site -- env`

### Identidock Example

Let’s redo the identidock example that we did before, but in K8s