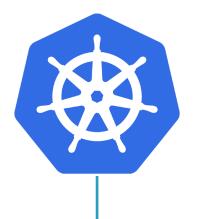
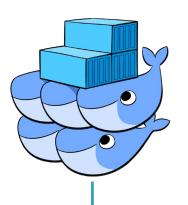
Using Cloud Container Services



Elton Stoneman CONSULTANT & TRAINER

@EltonStoneman | blog.sixeyed.com

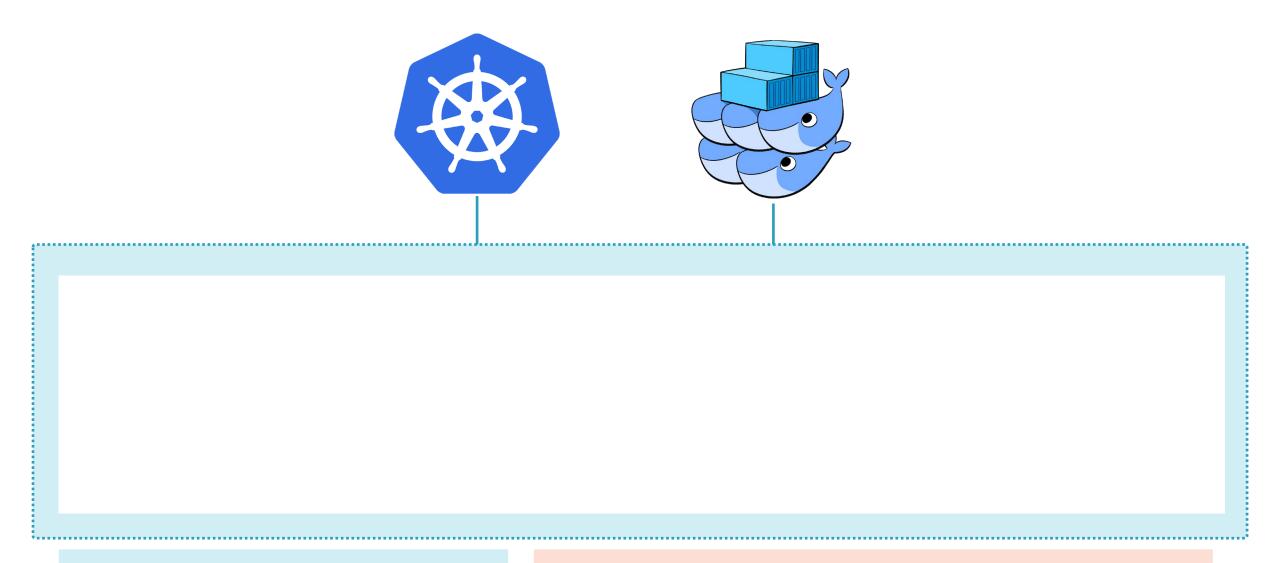






Control Plane

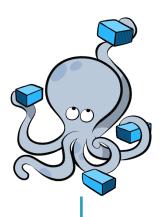
Workers



Control Plane

Workers





- Fully managed
- Scale on-demand
- Service integration

AKS ACI **ECS** 4-----4-----

kubectl apply # AKS

docker compose up # ACI and ECS

Managed Container Platforms

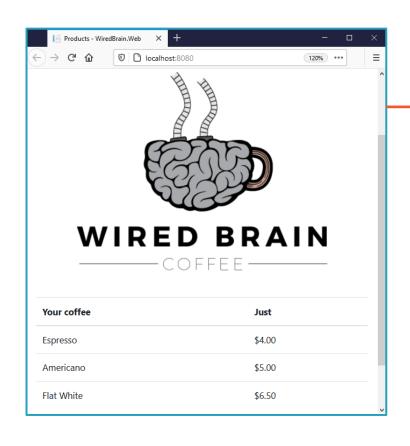
Standard modelling - Compose or Kubernetes

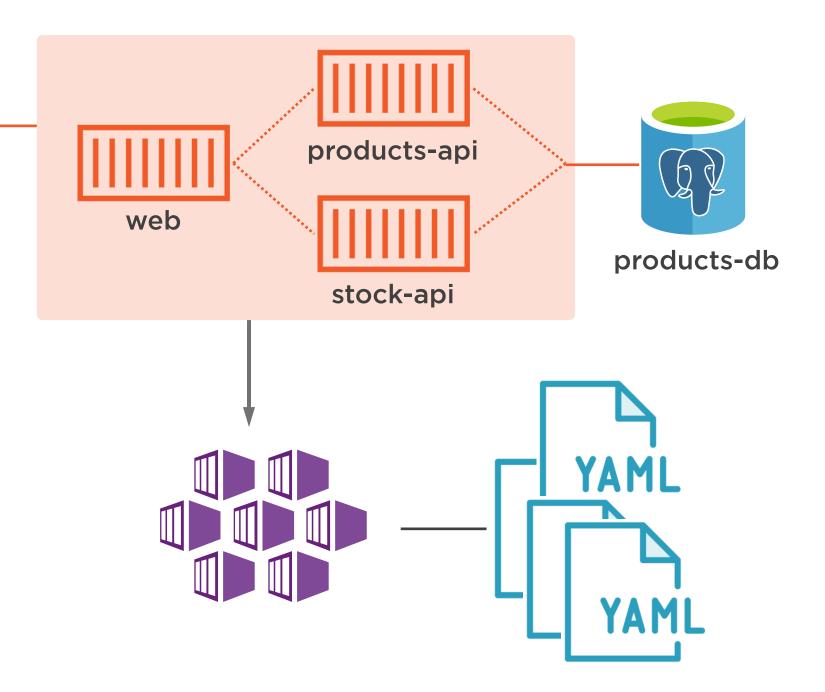
Demo



Deploying Kubernetes apps to AKS

- Creating the cluster
- Using managed databases
- Deploying to AKS





```
kubectl create secret # local sources
kubectl apply # config & app
kubectl get svc # print external IP
```

AKS Deployment

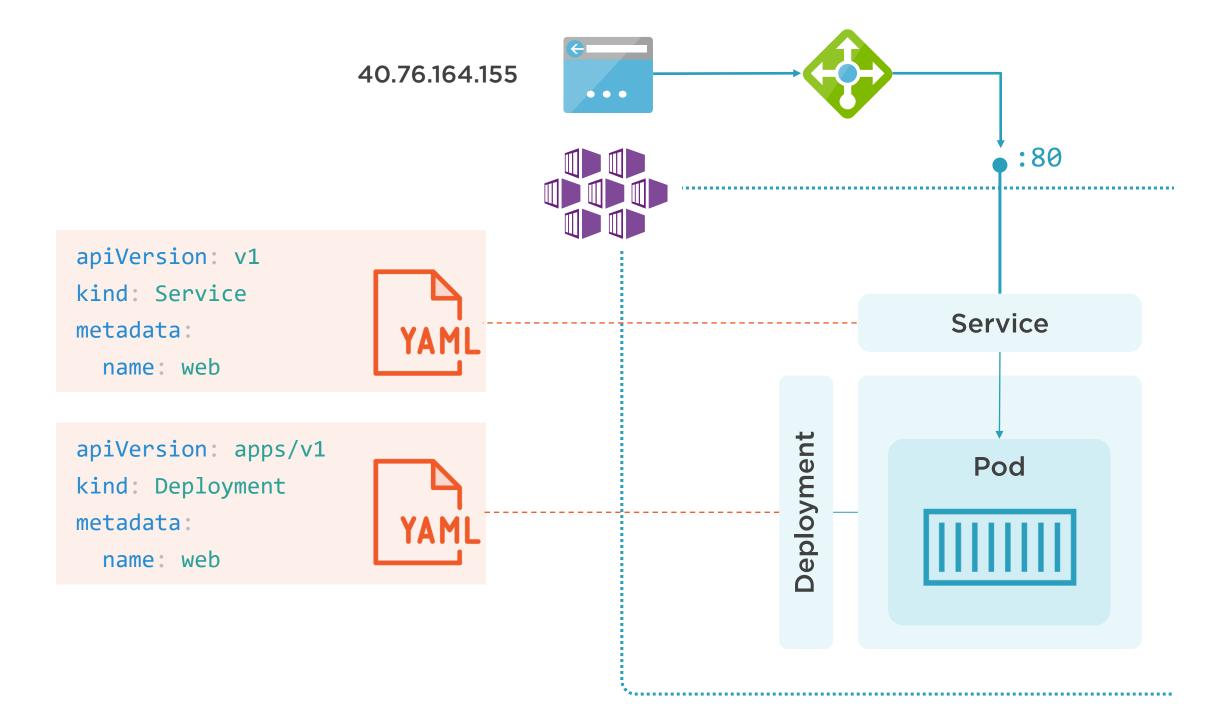
Standard Kubernetes model & tools

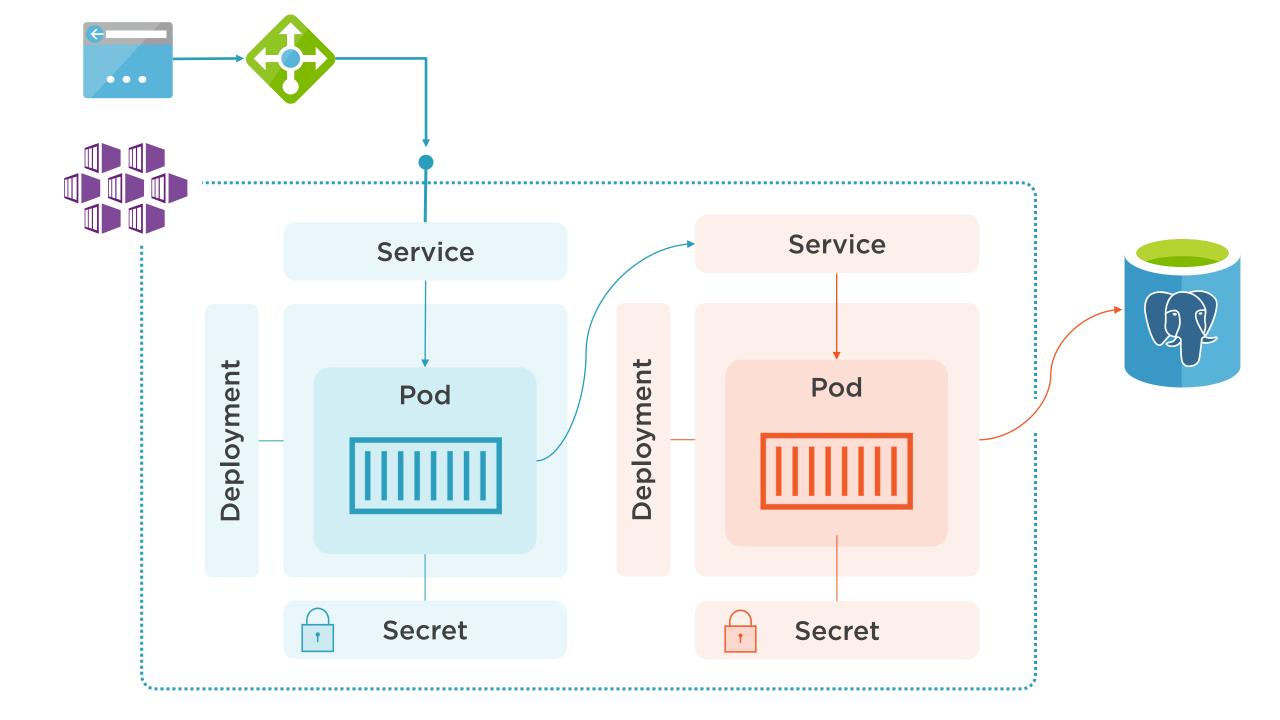
web.yaml

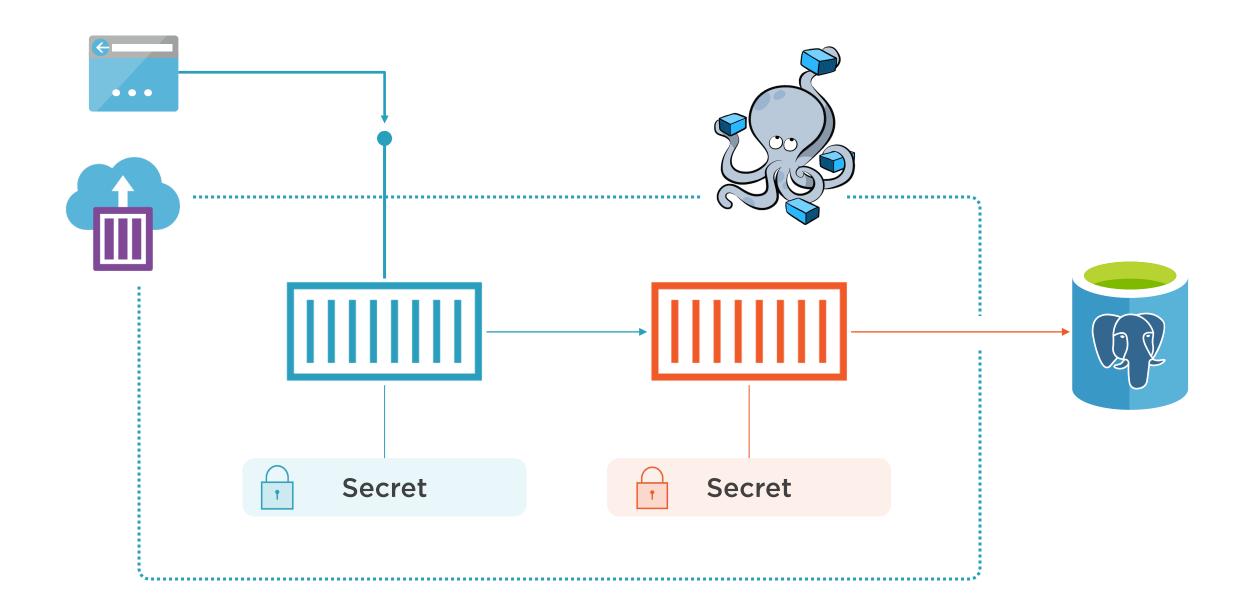
```
# in the Pod template
    containers:
      - name: api
        image: psdockerrun/web
        volumeMounts:
          # mount secrets
        resources:
          requests:
            cpu: 500m
            memory: 250Mi
          limits:
            cpu: 1000m
            memory: 500Mi
```

web.yaml (continued)

```
apiVersion: v1
kind: Service
metadata:
  name: web
  namespace: wb-prod
spec:
  ports:
    - port: 80
      targetPort: 80
  selector:
    app: web
  type: LoadBalancer
```





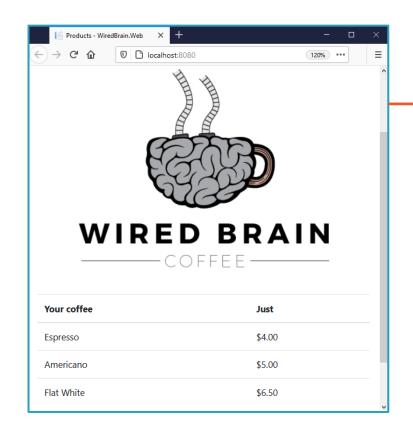


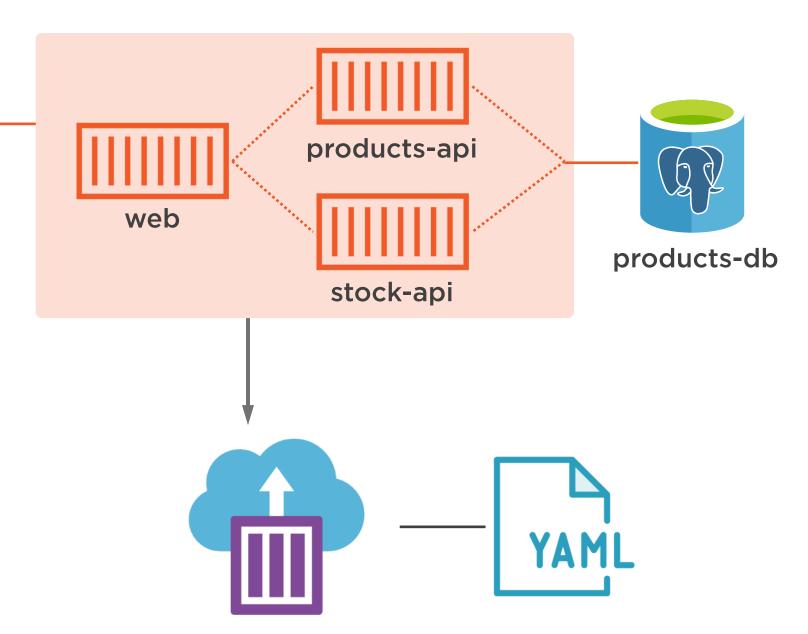
Demo



Deploying apps to ACI

- Configuring a Docker Context
- Modelling secrets
- Deploying with the Docker CLI





```
docker login azure# choose authdocker context create aci# includes RGdocker context use# or set DOCKER_CONTEXT
```

Connecting Docker to ACI Creating and using an ACI context

```
docker compose up # not docker-compose!
docker ps
docker logs
docker exec
```

Deployment and Operations

Standard Docker and Compose commands

docker-compose.yaml

```
services:
  products-api:
    image: psdockerrun/products-api
    secrets:
      - source: products-api-config
        target: /app/config/...
  stock-api:
    image: psdockerrun/stock-api
    env file:
      - ./secrets/stock-api.env
```

docker-compose.yaml (cont.)

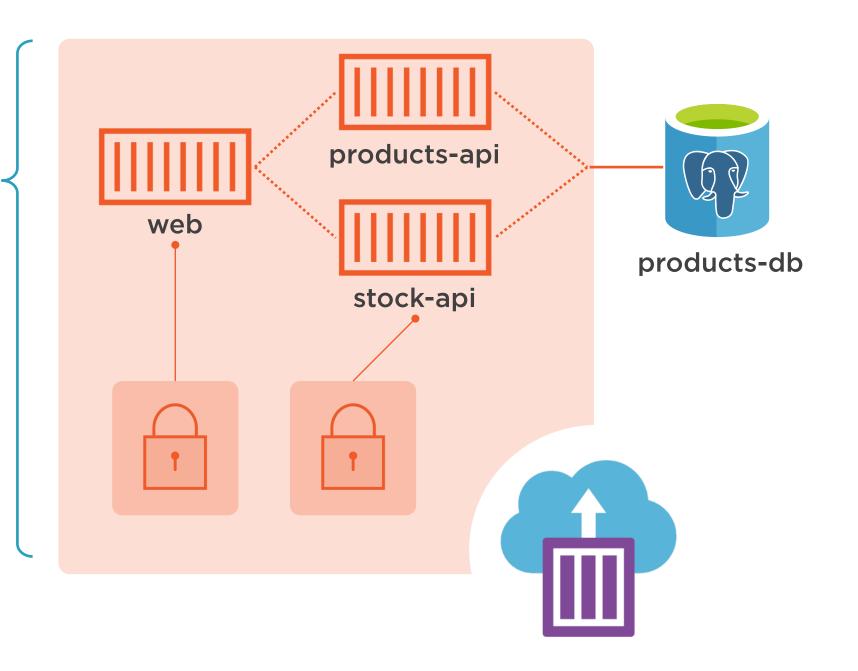
```
web:
    image: psdockerrun/web
    ports:
      - "80:80"
secrets:
 products-api-config:
   file: ./secrets/app.properties
 web-api:
   file: ./secrets/api.json
```

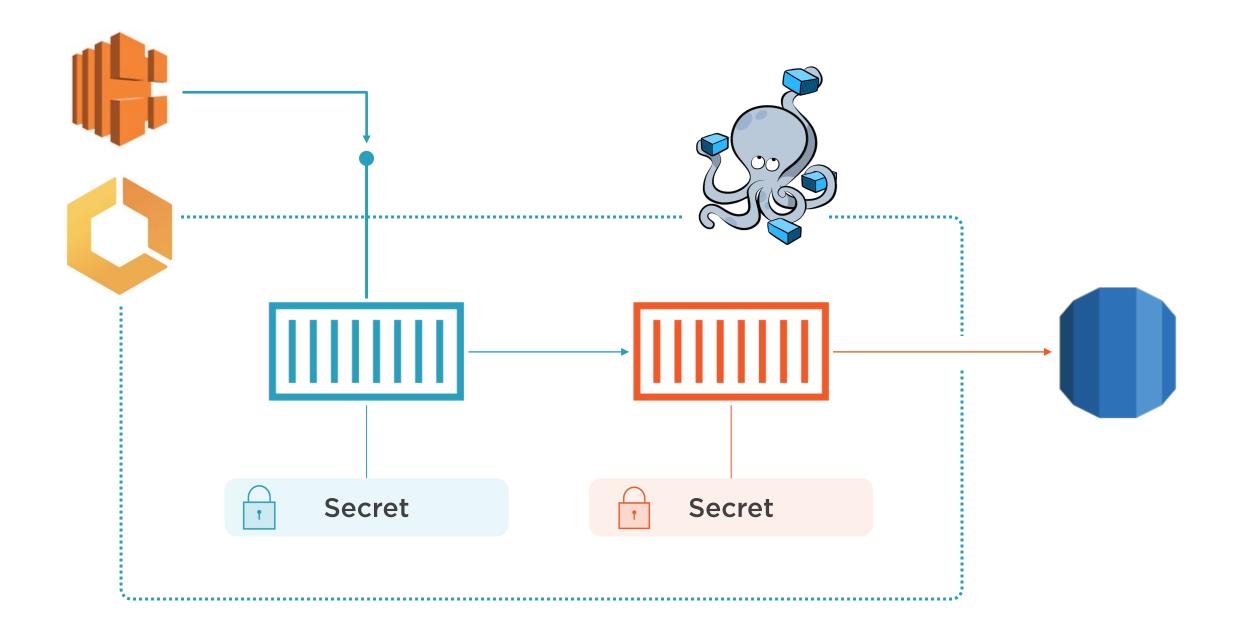
Ops / DevOps / SRE





Config Management



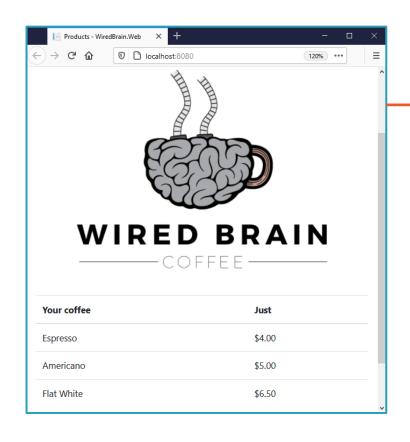


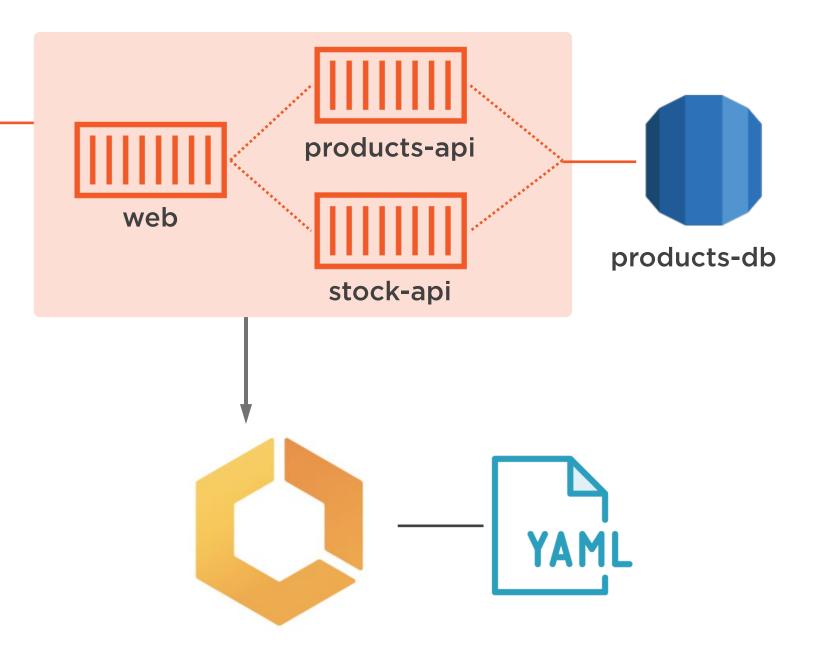
Demo



Deploying apps to ECS

- Configuring a Docker Context
- Modelling secrets
- Deploying with the Docker CLI





```
docker context create ecs # select region & auth
docker context use # or set DOCKER_CONTEXT
```

Connecting Docker to ECS

Creating and using an ECS context

```
docker compose up  # not docker-compose!
docker compose ps
docker compose logs
```

Deployment and Operations

Standard Compose commands

docker secret create docker secret ls

Managing Secrets

Stored in AWS Secrets Manager

docker-compose.yaml

```
services:
  products-api:
    image: psdockerrun/products-api
    secrets:
      - source: products-api-config
        target: app.properties
    networks:
      - wb-net
  stock-api:
    image: psdockerrun/stock-api
    environment:
      - POSTGRES CONNECTION STRING
```

docker-compose.yaml (cont.)

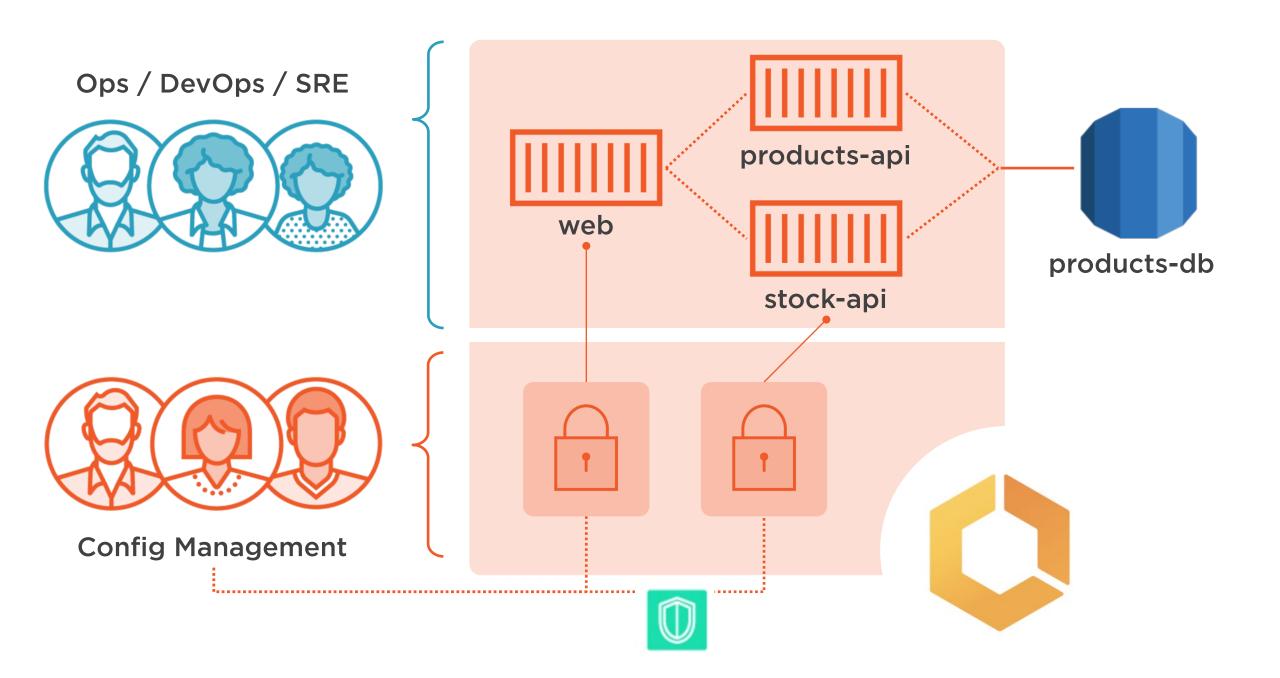
```
web:
    image: psdockerrun/web
    ports:
      - "80:80"
secrets:
  products-api-config:
    external: true
    name: arn:aws:secrets...FhC0p7
  web-api:
    external: true
    name: arn:aws:secrets... fdXo0p
```

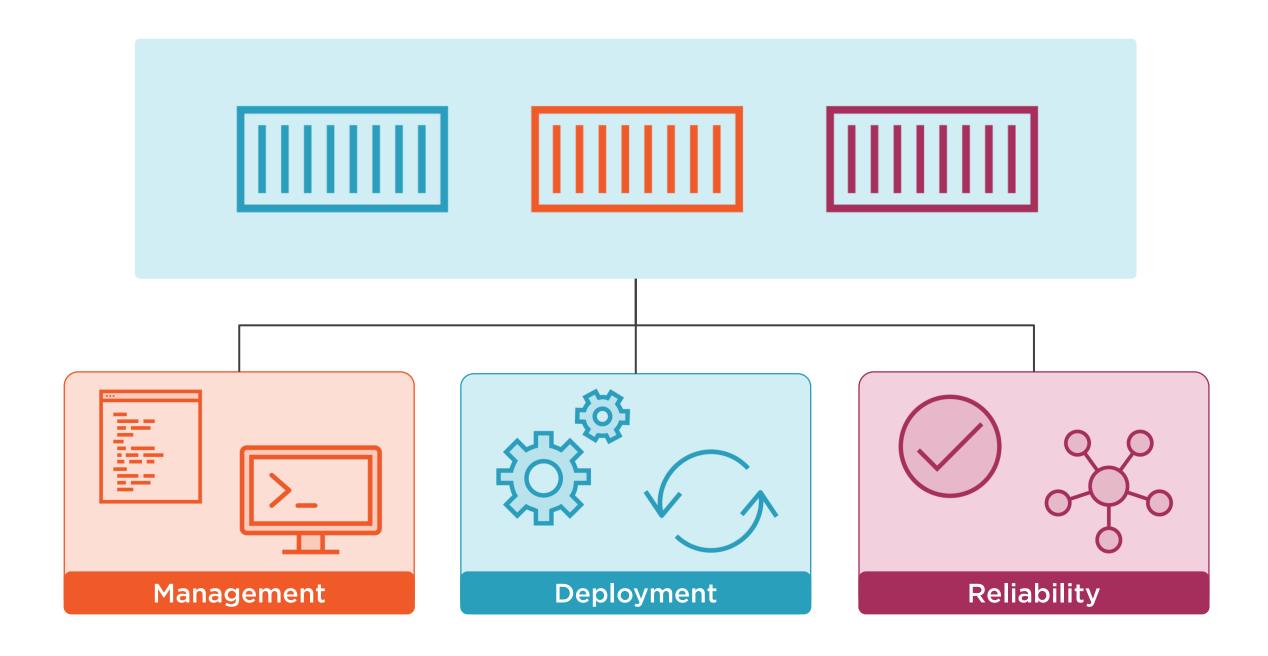
docker-compose.yaml

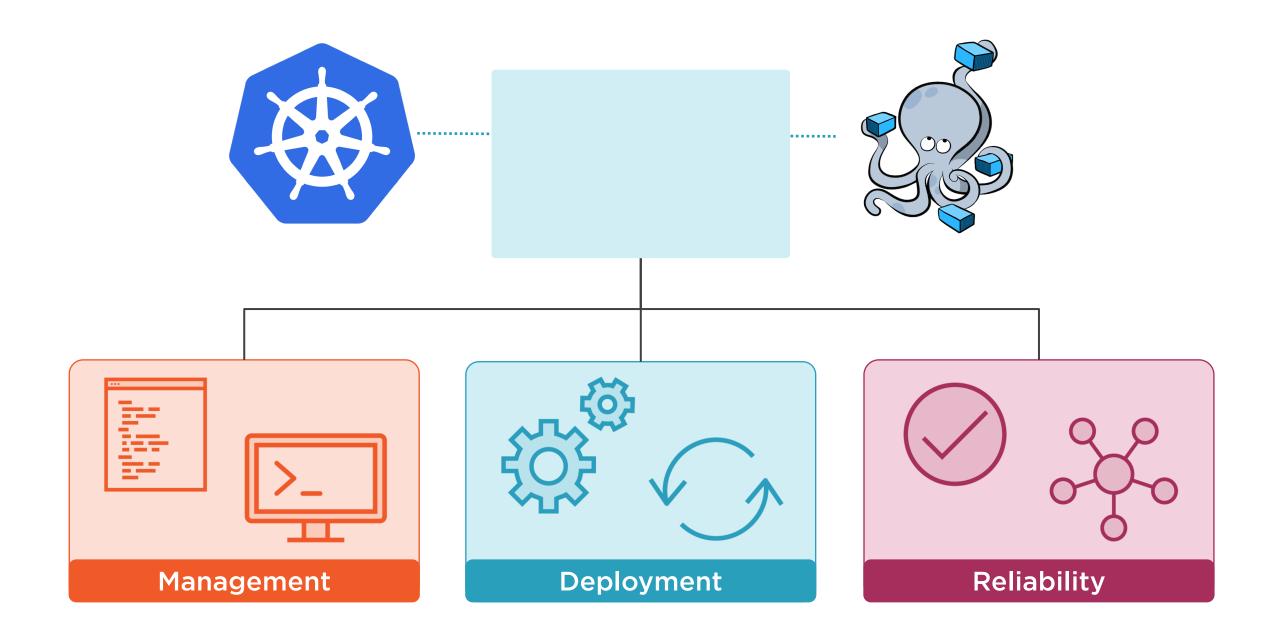
```
services:
  products-api:
    image: psdockerrun/products-api
    secrets:
      - source: products-api-config
        target: app.properties
    networks:
      - wb-net
  stock-api:
    image: psdockerrun/stock-api
    environment:
      - POSTGRES CONNECTION STRING
```

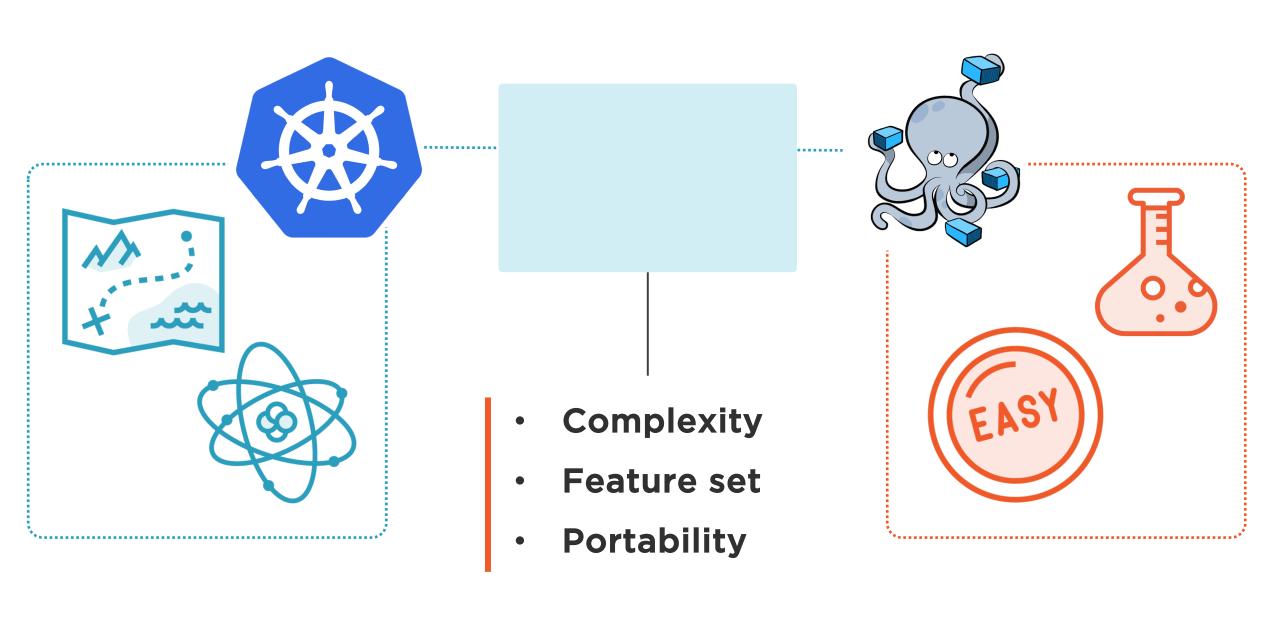
docker-compose.yaml (cont.)

```
web:
    image: psdockerrun/web
    ports:
      - "80:80"
secrets:
  products-api-config:
    external: true
    name: arn:aws:secrets...FhC0p7
  web-api:
    external: true
    name: arn:aws:secrets... fdXo0p
```









Summary



Cloud Container Services

- Managed Kubernetes AKS
- Azure Container Instances ACI
- Elastic Container Service ECS

Application Modelling

- Compose vs. Kubernetes
- Secret storage and management

Choosing a Container Platform

- Features and integration
- Modelling complexity
- Portability apps and skills

We're Done!



So...

- Please leave a rating
- Follow @EltonStoneman on Twitter
- Check out blog.sixeyed.com
- Watch my other courses ©