

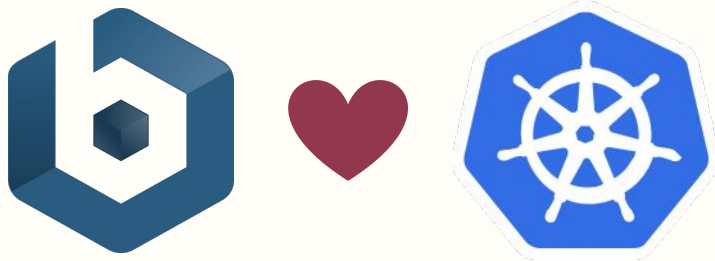
# TELEPRESENCE

SEAMLESS DEVELOPMENT ENVIRONMENTS IN KUBERNETES



bitnami

Ara Pulido (@arapulido)



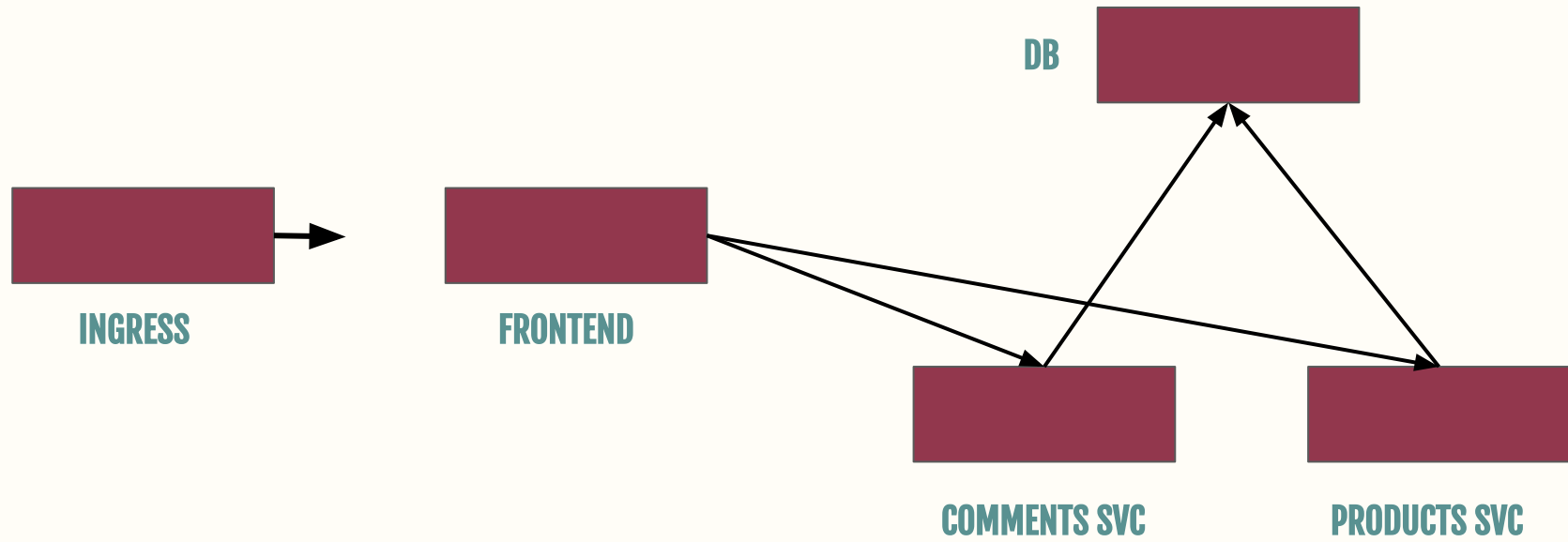
- **LEADER IN APPLICATION PACKAGING**
- **KUBERNETES CERTIFIED SERVICE PROVIDER**
- **CO-DEVELOP WITH MSFT, DEIS, HEPTIO**
- **MAINTAINING MANY OFFICIAL HELM CHARTS**



**KUBERNETES IS A GREAT  
PRODUCTION ENVIRONMENT,**

**but what about dev?**

# EXAMPLE APPLICATION



# DEVELOPING WITH DOCKER COMPOSE

```
services:
  ui:
    tty: true
    build: ./docker
    ports:
      - 4200:4200
    volumes:
      - ./app
    command: "ng serve"
  api:
    tty: true
    image: bitnami/monocular-api:latest
    volumes:
      - ./config.yaml:/config/monocular.yaml
    environment:
      - ENVIRONMENT=development
      - MONOCULAR_AUTH_SIGNING_KEY=secret
```

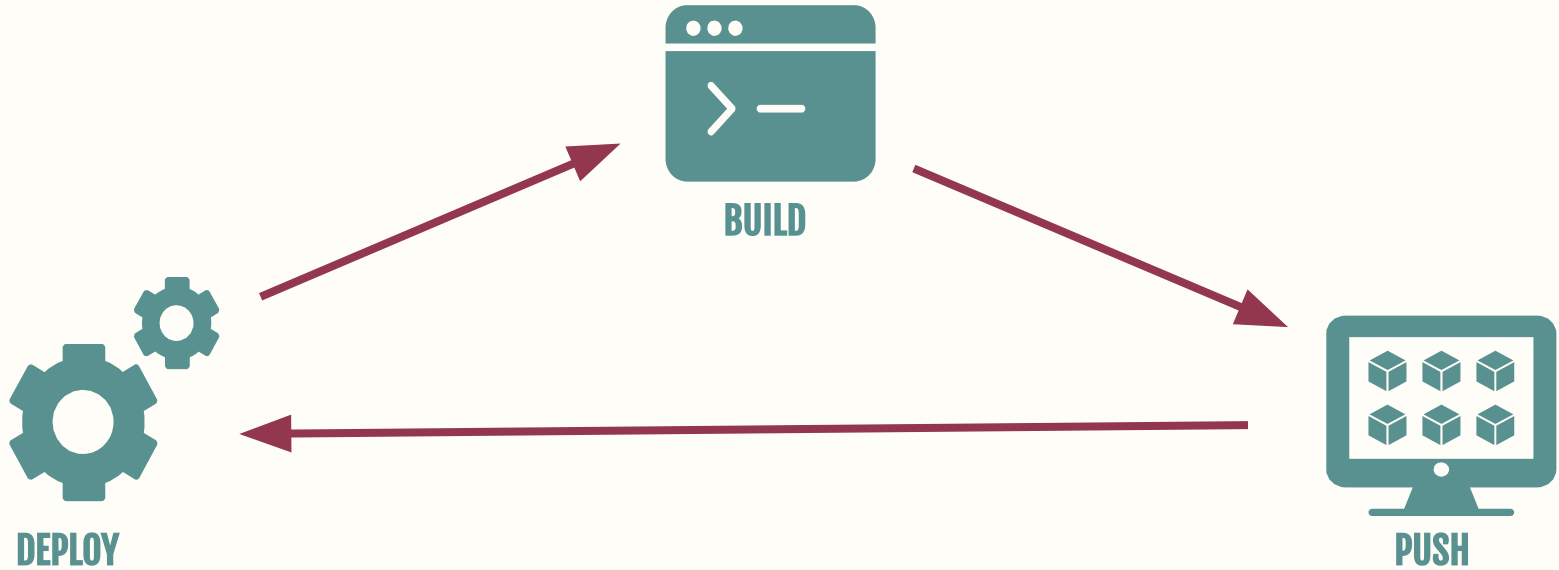
```
ratesvc:
  image: kubeapps/ratesvc:latest
  environment:
    - JWT_KEY=secret
  command:
    - /ratesvc
    - --mongo-url=mongodb://root@mongodb
    - --mongo-database=ratesvc
  mongodb:
    image: bitnami/mongodb:3
    environment:
      - MONGODB_ROOT_PASSWORD=password123
  auth:
    image: kubeapps/oauth2-bitnami:latest
    ...
  volumes:
    monocular-data:
```



# OTHER THINGS WE NEED

- **INGRESS**
- **JOB / CRONJOB**
- **INIT CONTAINERS**
- **MOUNT KUBECONFIG  
FOR EXTENSIONS**

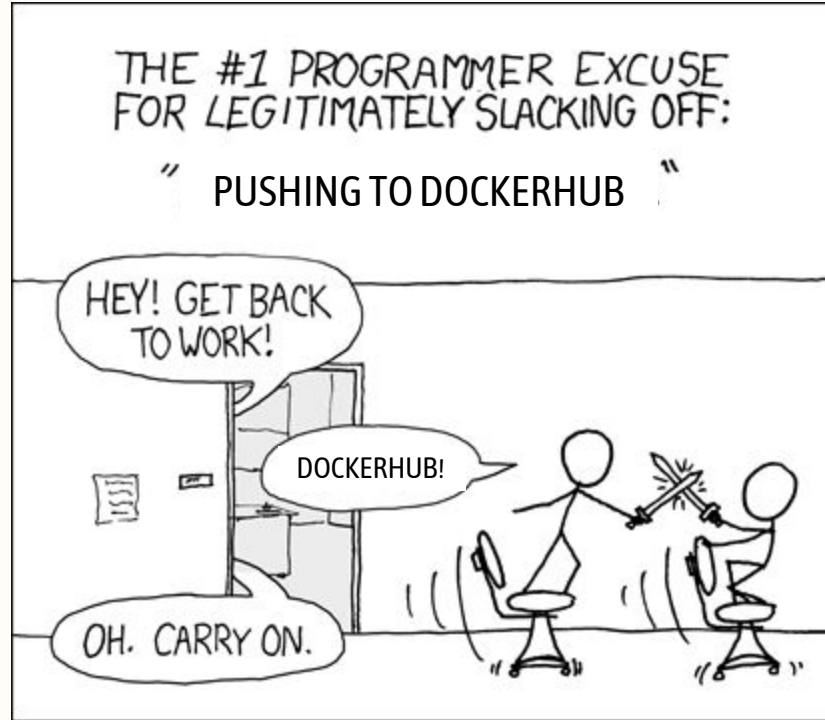
# DEVELOPMENT CYCLE

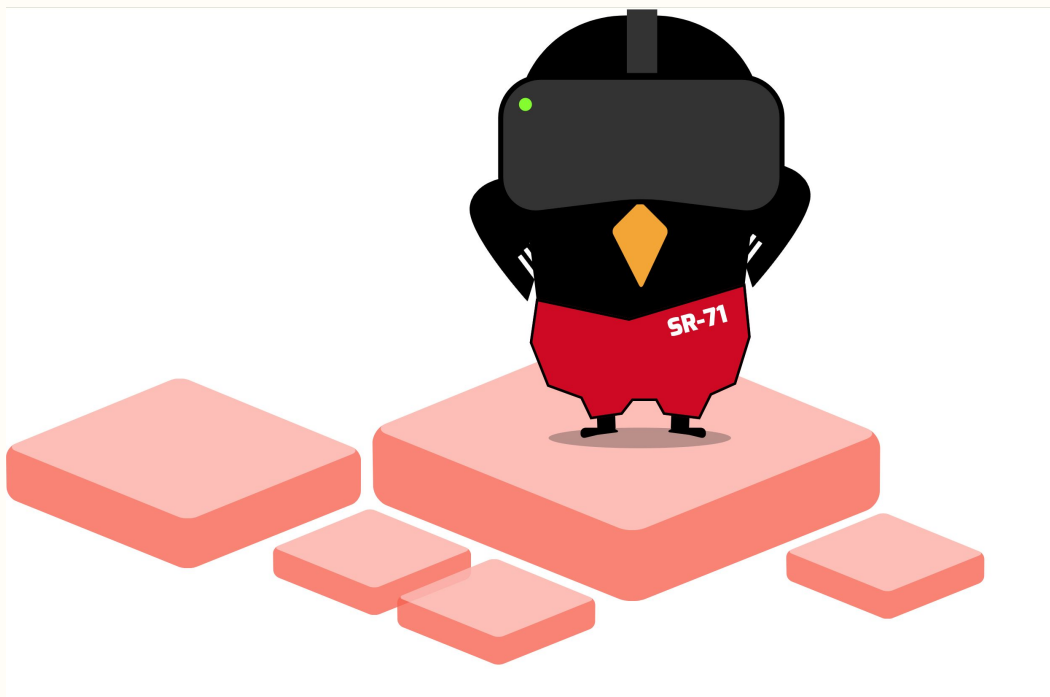




THE #1 PROGRAMMER EXCUSE  
FOR LEGITIMATELY SLACKING OFF:

" PUSHING TO DOCKERHUB "





# TELEPRESENCE

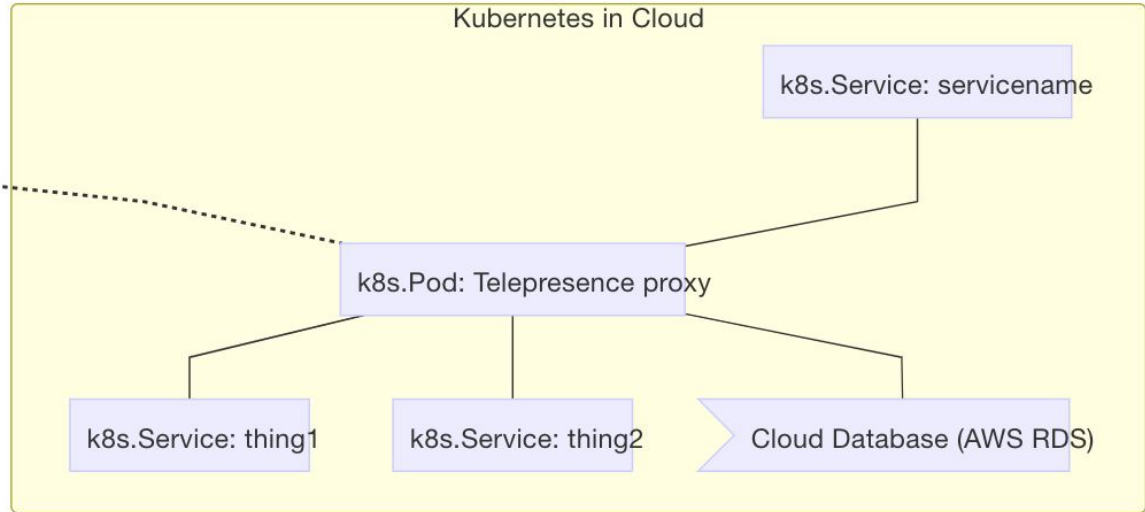
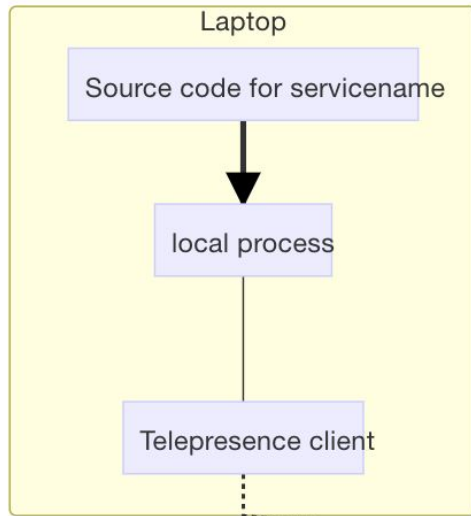
(telepresence.io)

BY



DATAWIRE

# HOW IT WORKS



**YOU GET SERVICE DISCOVERY, VOLUMES  
AND THE SAME ENVIRONMENT...**

**... FROM YOUR LOCAL MACHINE**

# PROXYING METHODS

## VPN

- ❑ WORKS BEST WITH GO
- ❑ DOESN'T WORK WITH OTHER VPNS

## INJECT-TCP

- ❑ INJECTS A SHARED LIBRARY INTO THE PROCESS
- ❑ DOESN'T WORK WITH STATICALLY LINKED

## DOCKER

- ❑ ALLOWS TO DO DOCKER-RUN
- ❑ USES DOCKER NETWORKING
- ❑ IDEAL FOR A CONTAINER-NATIVE DEVELOPMENT

# DEMO

```
    2170 function(element, attr, ngSwitchController) {
    2171     var selected = attr.ngSwitch || attr.on,
    2172         selectedTranscludes = [],
    2173         selectedElements = [],
    2174         previousElements = [],
    2175         selectedScopes = [];

    2176     scope.$watch/watchExpr, function ngSwitchWatchAction(value) {
    2177         var i, ii;
    2178         for (ii = 0, ii = previousElements.length; i < ii; ++i) {
    2179             previousElements[i].remove();
    2180         }
    2181         previousElements.length = 0;
    2182         for (ii = 0, ii = selectedScopes.length; i < ii; ++i) {
    2183             var selected = selectedElements[i];
    2184             selectedScopes[i].destroy();
    2185             previousElements[i] = selected;
    2186             animate.leave(selected, function() {
    2187                 previousElements.splice(i, 1);
    2188             });
    2189         }
    2190     };

    2191     selectedElements.length = 0;
    2192     selectedScopes.length = 0;
```

# ROADMAP

## ROBUSTNESS & SPEED

- ❑ RECONNECT
- ❑ BETTER CLEANUP ON EXIT
- ❑ STARTUP SPEED
- ❑ RBAC
- ❑ REMOVE INJECT-TCP

## FUTURE

- ❑ TRAFFIC SHADOWING
- ❑ SIMPLER UX
- ❑ IDE INTEGRATIONS
- ❑ ISTIO SUPPORT

```
1140 function(scope, element, attr, ngSwitchController) {
1141   var selected = attr.ngSwitch || attr.on,
1142       selectedTranscludes = [],
1143       selectedElements = [],
1144       previousElements = [],
1145       selectedScopes = [];
1146
1147   function switchAction(value) {
1148     // Find the new selected element
1149     var selectedElements = [];
1150     for (i = 0, ii = selectedScopes.length; i < ii; ++i) {
1151       var selected = selectedElements[i];
1152       selectedScopes[i].destroy();
1153       previousElements[i] = selected;
1154       animate.leave(selected, function() {
1155         previousElements.splice(i, 1);
1156       });
1157     }
1158     selectedElements.length = 0;
1159     selectedScopes.length = 0;
1160   }

```

# THANK YOU



bitnami

bitnami.com  
kubernetes.com