

# Zero To Cloud Native on IBM Cloud

## Introduction

Kevin Collins  
Technical Sales Leader  
IBM Cloud Americas – Enterprise Containers  
[kevincollins@us.ibm.com](mailto:kevincollins@us.ibm.com)

Kunal Malhotra  
Cloud Platform Engineer  
IBM Cloud MEA  
[Kunal.malhotra3.ibm.com](mailto:Kunal.malhotra3.ibm.com)

# GitHub Repositories

- ❖ **API Frontend** – Python flask based microservice that serves as the frontend for calling the three APIs.  
<https://github.com/kmcolli/api-frontend-02cn>
- ❖ **Web Frontend** – Web based frontend for invoking the APIs.  
<https://github.com/kmcolli/web-frontend-02cn>
- ❖ **Utility** – a utility microservice that performs common tasks that many of the other microservices use.  
<https://github.com/kmcolli/utility-02cn>
- ❖ **Enable SSH** – enables SSH on IBM Cloud Managed OpenShift Worker Nodes  
<https://github.com/kmcolli/enable-node-SSH-02cn>
- ❖ **OCP Realtime** – ‘realtime’ APIs that will return an OpenShift login token and a list of currently supported OpenShift versions.  
<https://github.com/kmcolli/ocp-realtime-02cn>
- ❖ **Load OCP Versions** – cronjob implementation that will retrieve a list of supported versions of IBM Cloud Managed OpenShift versions and store the results in Redis for quick access.  
<https://github.com/kmcolli/load-OCP-versions-02cn>
- ❖ **Secrets** – secrets template yaml file. The zero-to-cloud-native application will require that a number of Kubernetes secrets are set. This template file will need to be updated with parameters for your environment.  
<https://github.com/kmcolli/secrets-02cn>

# Code Layout

Each repository has the same folder layout. You will see the following folder structure:

- ❖ **app** – contains the application source code.
- ❖ **deployments** – contains Kubernetes deployment files
- ❖ **pipeline** – contains ‘Classic’ pipeline configuration scripts – used for our classic CI/CD toolchain
- ❖ **.tekton** – contains ‘Tekton’ pipeline configuration scripts – used for our Tekton CI/CD toolchains.
- ❖ **Dockerfile** – Dockerfile used to build the image for the microservice configuration – Python requirements file
- ❖ **.dockerignore** – file that tells docker which files to ignore while building an image