Wek8s Deployment: Introduction

This guide is designed to walk you through the basic steps of deploying an app for the first time using Wek8s. This guide does not describe using databases and object storage, running monitoring and alerts, or troubleshooting. Those topics will be addressed in other guides. The goal of this guide is to help you:

- Confirm that you have all of the necessary tools, configured correctly, in order to deploy an app to a Wek8s cluster.
- Deploy your app and verify success.

How do I deploy my app with Wek8s?

Start with this Overview of what Wek8s is and who will be using it.

You can also look at this page of Additional Resources and definitions, which directs you to other sources of information about Wek8s and defines terms you'll see frequently in this guide. (This will be especially helpful if you're new to using containers, Docker, and Kubernetes.)

Deployment prerequisites

Complete these prerequisites before you deploy your app.

1. Confirm your access to developer tools. Verify that you have the tools you need in order to deploy your app with Wek8s. Without access to the necessary tools, properly configured, you won't be able to access Wek8s clusters. If you went through WeWork Engineering's onboarding process, you should have received access to the tools described in this section—but your first step should be to confirm that each app is working properly for you.
2. Configure your SSH authentication in Github.
3. Create (or choose) a namespace for your deployment.
4. Install Docker and create a Dockerfile for your application.
5. Connect Docker to Quay and push your Docker image to the Quay repository.
6. Test your access to the Wek8s cluster. Currently, WeWork maintains three Wek8s clusters: Hydra and Kraken (production), and Phoenix (development).
7. Configure your CircleCI projects.

Steps for deploying your application

After you have set up your environment, follow these steps to deploy.

1. Configure your Helmfile.
2. Configure your CircleCI config.yml.
3. Secure your app using the Bifrost gateway or another authentication/authorization method.
4. Refer to CircleCI to see details of the deployment of your app. After you commit changes in your repo, CircleCI automagically begins deploying your app to the Wek8s cluster. You can go to CircleCI to watch your app's progress and confirm that it has been pushed live.

Overview: What is Wek8s?

WeK8s is a platform for running containerized applications at WeWork. It enables engineers to easily deploy applications on secure, compliant, multi-cloud or on-premise Kubernetes clusters.

Kubernetes is the foundation of Wek8s. You can use additional tools when working with Wek8s for increased control over applications. However, this guide explains only how to deploy your app. Other topics—such as using Vault secrets, doing monitoring and logging on your apps, and setting up databases and object storage—will be addressed in other guides. This guide is your "first-time deployment walk-through."

How is my app deployed to Wek8s?

The simplified answer is that:

1. You "containerize" your application in a Dockerfile to prepare it for deployment.
2. You push the resulting Docker file "image" to WeWork's container repository (we use Quay instead of Docker Hub).
3. You set variables (in a Helm file that you add to your application's Github repo) so that your application can run properly on a Wek8s cluster.
4. You tell CircleCI (by configuring a circleci.yml file) how to grab and build your image every time you push a change to your application's Github repo.

As a result of the steps above, when you push code to your app's Github code repo, Github sends a signal to CircleCI, telling it to:

- read the config.yml file
- build your Docker image (that is, your containerized app)
- push it to Quay (the repository we use for Dockerized apps)
- deploy it to the WeK8s cluster that you identified in your Helm files

With your code, Dockerfile, Helm file, and CircleCI config.yml file all in place and configured correctly, a push to your application's Github repo will cause your app to build and deploy.

Who is this guide for?

This guide is for WeWork application developers, WeWork product managers, and anyone else who wants to understand or use Wek8s for application deployment.

Additional resources

These links point to more information about the apps and concepts surrounding Wek8s. You do not need to read all of this information in order to follow the basic steps in this guide; however, if you want some context for what these tools are and how they work together, these articles will help.

- Here's a good introduction to containers.
- This is Docker's intro to Docker.
- Here's an intro to Helm. Helm is a package manager for Kubernetes. It allows you to deploy to the Kubernetes infrastructure with the help of .yml files instead of executing several **`kubectl`** commands to achieve the deployment. For details, see this very helpful learning resource on Helm.
- WeWork uses Quay as its registry for Docker containers. Here is an article that explains a little more about what a container registry is and why you might use one.

Glossary of terms

When we talk about Wek8s, several components are mentioned frequently. It helps to have a clear definition of each of them.

- **AWS Account**: We have AWS accounts for Development and Production. Each AWS account holds one or more Wek8s clusters inside of it.
- **AWS Role**: An AWS role specifies the permissions you have inside of AWS. Each role is mapped to a namespace in Wek8s. You can choose the role you want to use - a normal role or an administrative role.
- **Docker container**: A Docker container is a standalone, executable package of software that includes everything needed to run an application (the application code and its dependencies).
- **Dockerfile**: A text document that contains all of the instructions that Docker needs in order to assemble an image.
- **Docker image**: A file, comprised of multiple layers, used to execute code (run your application) inside a Docker container. The Docker image is built using instructions contained in the Dockerfile.
- **Namespace**: A namespace maps to an AWS role. A namespace is a logical entity used to represent cluster resources that will be used by a particular group of users. "Namespace" is a concept that comes directly from Kubernetes. You can read about how Kubernetes defines a namespace for a broader definition.
- **Wek8s Cluster**: A cluster consists of at least one main controlling unit, called a cluster master, and multiple machines running applications and workflows, called nodes. Wek8s clusters live inside of one AWS Account. If you're trying to reach a dev cluster, use a dev account; to reach a prod cluster, use a prod account.