

Tanzu Community Workshop: "Spring on Kubernetes"

<https://tanzu.vmware.com/developer/workshops/>

☐ Scaffold

- Reference: Section 17 in the workshop
- iterate on your application source code locally then deploy to local or remote Kubernetes clusters

☐ - Use Config CRD with kubectl in the deploy property

- ``scaffold dev --port-forward``
- References:
 - <https://github.com/GoogleContainerTools/skaffold>

☐ Kustomize CLI vs Helm package install with values file

- Reference: Section 24 in the workshop
- eg. ``mkdir -p kustomize/base`` : contains base definitions for all environments
- eg. ``mkdir -p kustomize/qa`` : contains ``patchesStrategicMerge`` and files defining overrides.
- In both directory above, there is a ``kustomization.yaml`` file defining resources (Kustomization CRD)
- eg. ``kustomize build kustomize/qa | kubectl apply -f -``
- By default kustomize generates a random name suffix for the ConfigMap. Kustomize will take care of reconciling this when the ConfigMap is referenced in other places (ie in volumes). It does this to force a change to the Deployment and in turn force the app to be restarted by Kubernetes. This isn't always what you want, for instance if the ConfigMap and the Deployment are not in the same Kustomization. If you want to omit the random suffix, you can set `behavior=merge` (or `replace`) in the `configMapGenerator`.

☐ - References:

- <https://kustomize.io/>
- Kustomize + Kubectl : <https://kubectl.docs.kubernetes.io/guides/introduction/>

☐ Combine the Kustomize/Kubectl with Skaffold

- Update Skaffold Config CRD definition to use kustomize in the deploy property, and added profiles property
- Run ``skaffold dev --port-forward`` to deploy without overrides

- Run `skaffold dev --port-forward -p qa` to deploy with `profile`.
- Inject environment variables into Spring App by creating configmap using Kustomize
 - Create `kustomize/base/application.yaml` : similar to application.properties file in Spring resources
 - Update `kustomize/base/kustomization.yaml`