Felix Hochleitner
Sr. DevOps Engineer
Gepardec

# GitOps powered CICD

## Gluing it together with Argo Events, Workflows & CD

KCD AUSTRIA September 26th-27th 2023

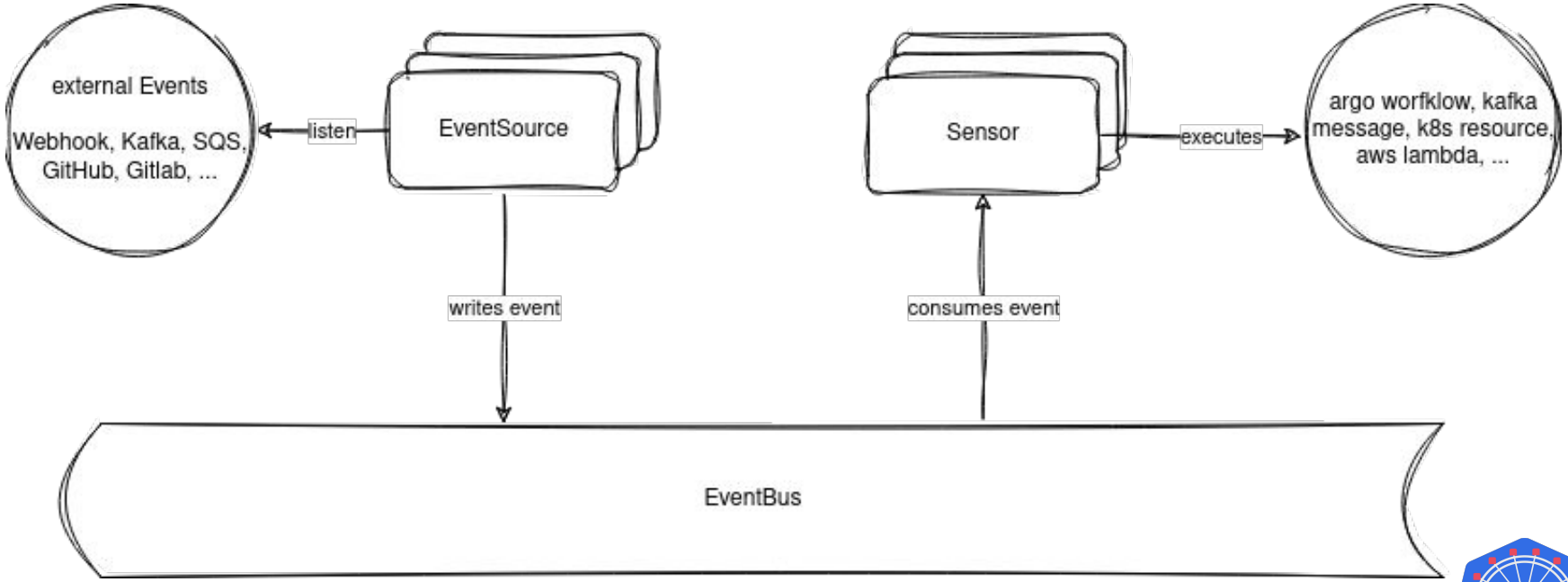
# objectives

- want everything managed by ArgoCD
- want every code change to be built & tested
- want ephemeral environments for feature-branches/MR
- want workflows to be reusable

# Argo Events

triggering our build and deployment processes

# Argo Events


external Events
Webhook, Kafka, SQS, GitHub, Gitlab, ...
listen
EventSource
Sensor
executes
argo worfklow, kafka message, k8s resource, aws lambda, ...
writes event
consumes event
EventBus

# EventSource

```yaml
apiVersion: argoproj.io/v1alpha1
kind: EventSource
metadata:
  name: demo-microservice-configurator
  namespace: argo-events-eventbus
spec:
  service:
    ports:
      - port: 12000
        targetPort: 12000
  github:
    demo-microservice-event:
      owner: "gepaplexx-demos"
      repository: "demo-microservice"
      webhook:
        endpoint: "/event"
        port: "12000"
        method: "POST"
      events:
        - "*"
      active: true
```

# Sensor - filtering and transformation

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Sensor
metadata:
  name: demo-microservice
  namespace: argo-events-eventbus
spec:
  template:
    serviceAccountName: operate-workflow-sa
  dependencies:
    - name: demo-microservice-event
      eventSourceName: demo-microservice-configurator
      eventName: demo-microservice-event
      filters:
        data:
          - path: "body.X-GitHub-Event"
            type: string
            value:
              - "push"
              - "create"
              - "delete"
```

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Sensor
metadata:
  name: demo-microservice-gitlab
  namespace: gp-cicd-eventbus
spec:
  template:
    serviceAccountName: operate-workflow-sa
  dependencies:
    - name: demo-microservice-gitlab-delete
      eventSourceName: demo-microservice-gitlab-configurator
      eventName: demo-microservice-gitlab-event
      filters:
        data:
          - path: body.after
            type: "string"
            operator: "="
            value:
              - "0000000000000000000000000000000000000000"
      transform:
        jq: .body.event_name = "delete"
```

# Triggering a workflow

```yaml
triggers:
  - template:
      name: demo-microservice-event-trigger
      conditions: "demo-microservice-event"
      argoWorkflow:
        parameters:
          - src:
              dependencyName: demo-microservice-event
              dataTemplate: '{{ toJson .Input.body }}'
            dest: spec.arguments.parameters.0.value
        operation: submit
        source:
          resource:
            apiVersion: argoproj.io/v1alpha1
            kind: Workflow
            metadata:
              generateName: demo-microservice-configurator-
              namespace: demo-microservice-cicd
            spec:
              arguments:
                parameters:
                  - name: event
                    value: "will-be-set-from-payload"
              entrypoint: configure
              workflowTemplateRef:
                name: workflow-configurator-github
                clusterScope: true
```

# Argo Workflows

## building our application

# Main Features

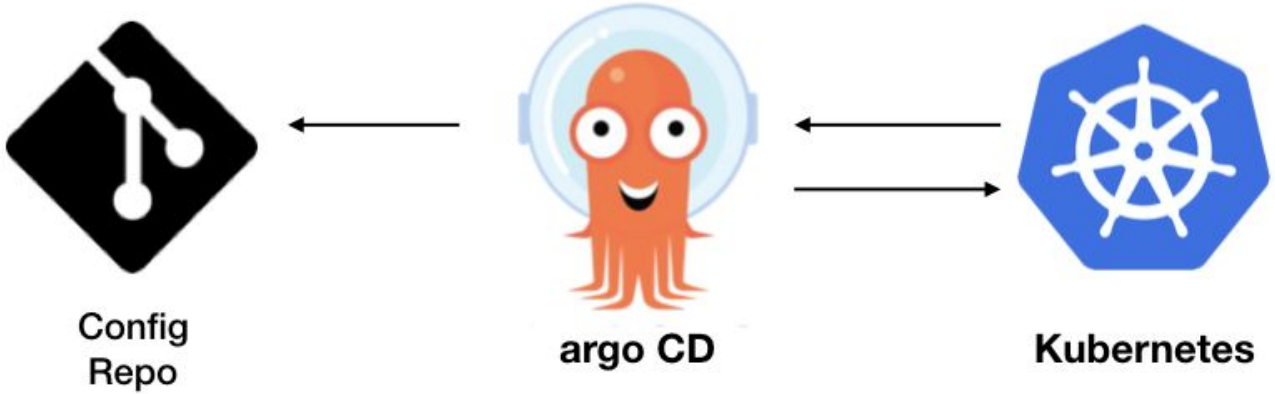
- every step is executed in a container
- extensive templating support
- Workflow & (Cluster-)WorkflowTemplate
- templates can invoke other templates

# workflows in action

# Argo CD

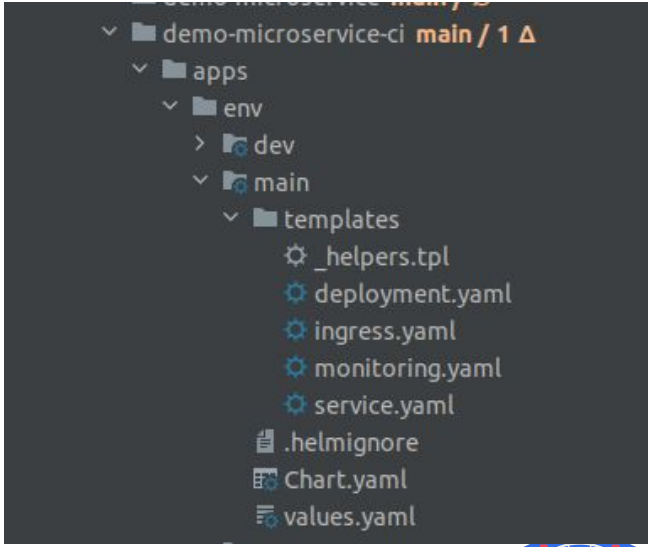
deploying our application

# high-level overview of Argo CD

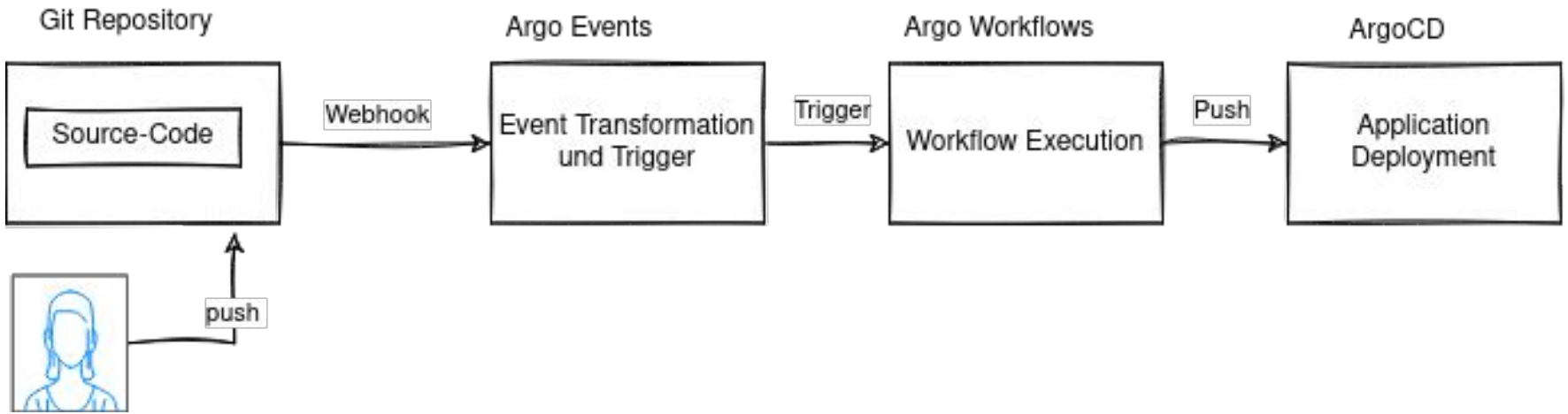


# ArgoCD Application

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
  name: demo-microservice-main
  namespace: gp-cicd-tools
spec:
  destination:
    namespace: demo-microservice-main
    server: 'https://kubernetes.default.svc'
  project: demo-microservice
  source:
    path: apps/env/main
    repoURL: 'git@github.com:gepaplexx-demos/demo-microservice-ci.git'
    targetRevision: main
  syncPolicy:
    automated:
      prune: true
      selfHeal: true
    syncOptions:
      - CreateNamespace=true
```



# Putting everything together



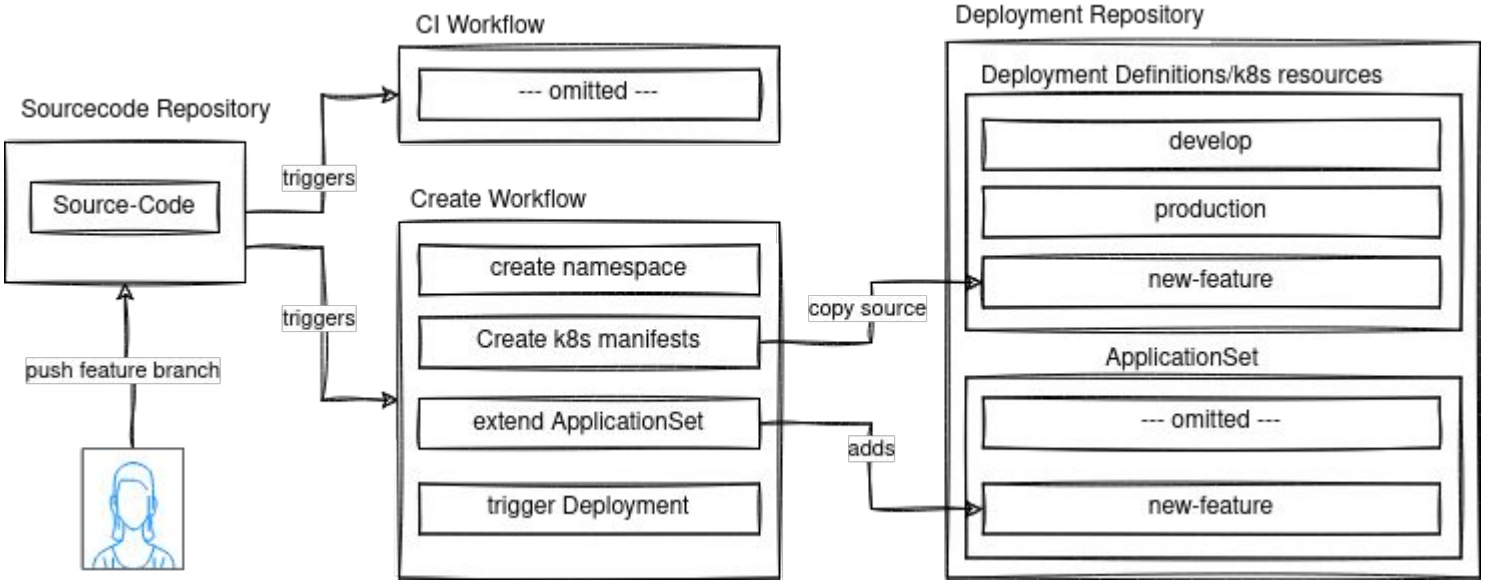
# Ephemeral Environments

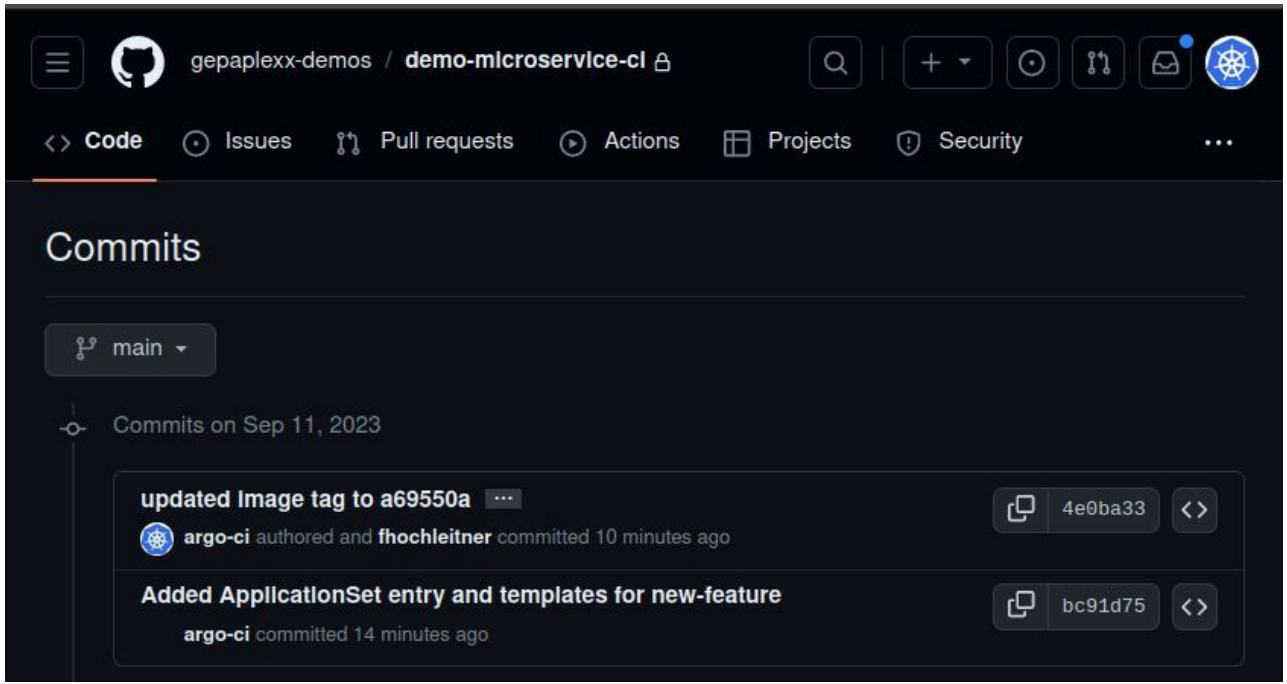
on the fly test environments

# ApplicationSets & the power of GitOps

```yaml
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: demo-microservice
  namespace: gp-cicd-tools
spec:
  syncPolicy:
    preserveResourcesOnDeletion: false
  generators:
    - list:
        elements:
          - cluster: dev
            url: https://kubernetes.default.svc
            branch: dev
          - cluster: prod
            url: https://api.production-cluster.mycompany.com
            branch: prod
          - cluster: new-feature
            url: https://kubernetes.default.svc
            branch: main
  template:
    metadata:
      name: "demo-microservice-{{cluster}}"
    spec:
      project: demo-microservice
      -- ommited for brevity --
      source:
        repoURL: git@github.com:gepaplexx-demos/demo-microservice-ci.git
        targetRevision: "{{ branch }}"
        path: apps/env/{{ cluster }}
      destination:
        server: "{{url}}"
        namespace: "demo-microservice-{{cluster}}"
```

# ApplicationSets & the power of GitOps




gepaplexx-demos / demo-microservice-ci
Code
Issues
Pull requests
Actions
Projects
Security
Commits
main
Commits on Sep 11, 2023
updated Image tag to a69550a
argo-ci authored and fhochleitner committed 10 minutes ago
4e0ba33
Added ApplicationSet entry and templates for new-feature
argo-ci committed 14 minutes ago
bc91d75

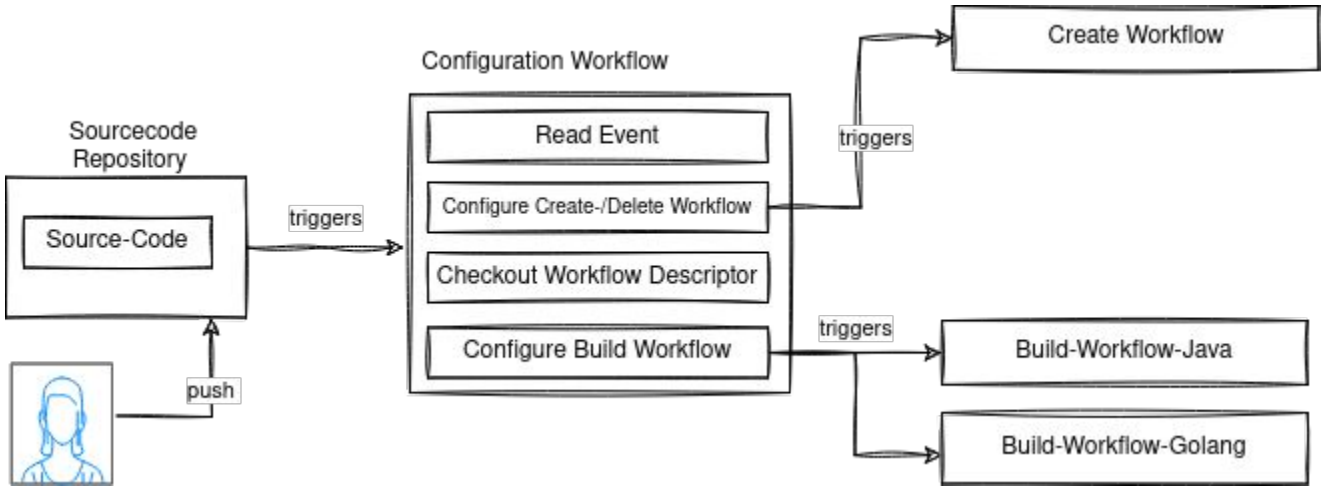
# multi-branch workflows

## branch-specific configuration & reusability

# build your own jenkinsfile

```
{
  "build": {
    "language": {
      "name": "java",
      "version": "openjdk-11"
    },
    "type": {
      "name": "maven",
      "version": "3.8.6",
      "configuration": {
        "args": "clean install",
        "config": "pom.xml",
        "config-path": "",
        "options": ""
      }
    },
    "additional-configuration-options": {
      --- omitted ---
    },
    "static-code-analysis": {
      "active": "true",
      "type": {
        "name": "sonarqube"
      }
    }
  }
}
```

# meta-workflow



# Workflow Templating at Runtime

```yaml
apiVersion: argoproj.io/v1alpha1
kind: ClusterWorkflowTemplate
metadata:
  name: workflow-build-java
spec:
  entrypoint: pipeline
  arguments:
    parameters:
      --- omitted for brevity ---
  templates:
    - name: pipeline
      metadata:
        labels:
          template: pipeline
      dag:
        tasks:
          - name: checkout
            templateRef:
              name: git-operations
              template: checkout
              clusterScope: true
          - name: build
            depends: "checkout"
            templateRef:
              name: "{{ workflow.parameters.build-type }}-operations"
              template: build
              clusterScope: true
            arguments:
              parameters:
                - name: config
                  value: "{{ workflow.parameters['build-config'] }}"
```

```yaml
apiVersion: argoproj.io/v1alpha1
kind: ClusterWorkflowTemplate
metadata:
  name: maven-operations
spec:
  templates:
    - name: build
      inputs:
        parameters:
          --- omitted for brevity ---
      container:
        name: maven
        image: ghcr.io/gepaplexx/maven:{{ workflow.parameters.build-type-version }}-
{{ workflow.parameters.language-version }}
        command:
          - "/usr/bin/mvn-wrapper.sh"
        args:
          - --- omitted ---
```

# recap

- everything  is a Kubernetes resource and can be managed by GitOps engine
- very powerful templating possibilities
- prometheus metrics

- it’s not the easiest to get started with
- a lot of tinkering and building it yourself involved

Code Snippets & Slides

https://github.com/fhochleitner/kcd-2023

**Thank you**