

Lab

Exit handlers

An exit handler is a template that *always* executes, irrespective of success or failure, at the end of the workflow.

Some common use cases of exit handlers are:

- cleaning up after a workflow runs
- sending notifications of workflow status (e.g., e-mail/Slack)
- posting the pass/fail status to a web-hook result (e.g. GitHub build result)
- resubmitting or submitting another workflow

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: exit-handlers-
  namespace: argo
spec:
  entrypoint: intentional-fail
  serviceAccountName: training
  onExit: exit-handler          # invoke exit-handler template at end
                                # of the workflow
  templates:
    # primary workflow template
    - name: intentional-fail
      container:
        image: alpine:latest
        command: [sh, -c]
        args: ["echo intentional failure; exit 1"]

    # Exit handler templates
    # After the completion of the entrypoint template, the status of the
    # workflow is made available in the global variable {{workflow.status}}.
    # {{workflow.status}} will be one of: Succeeded, Failed, Error
    - name: exit-handler
      steps:
        - - name: notify
            template: send-email
          - name: celebrate
            template: celebrate
            when: "{{workflow.status}} == Succeeded"
        - name: cry
            template: cry
            when: "{{workflow.status}} != Succeeded"
```

```

- name: send-email
  container:
    image: alpine:latest
    command: [sh, -c]
    args: ["echo send e-mail: {{workflow.name}} {{workflow.status}}
{{workflow.duration}}"]
- name: celebrate
  container:
    image: alpine:latest
    command: [sh, -c]
    args: ["echo hooray!"]
- name: cry
  container:
    image: alpine:latest
    command: [sh, -c]
    args: ["echo boohoo!"]

```

```

# This example demonstrates the use of retry back offs
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: retry-backoff-
  namespace: argo
spec:
  entrypoint: retry-backoff
  serviceAccountName: training
  templates:
    - name: retry-backoff
      retryStrategy:
        limit: 10
        retryPolicy: "Always"
        backoff:
          duration: "1"      # Must be a string. Default unit is seconds. Could
also be a Duration, e.g.: "2m", "6h", "1d"
          factor: 2
          maxDuration: "1m" # Must be a string. Default unit is seconds. Could
also be a Duration, e.g.: "2m", "6h", "1d"
          affinity:
            nodeAntiAffinity: {}
      container:
        image: python:alpine3.6
        command: ["python", -c]
        # fail with a 66% probability
        args: ["import random; import sys; exit_code = random.choice([0, 1, 1]);
sys.exit(exit_code)"]

```

