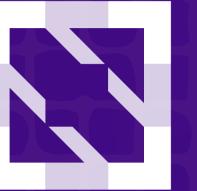




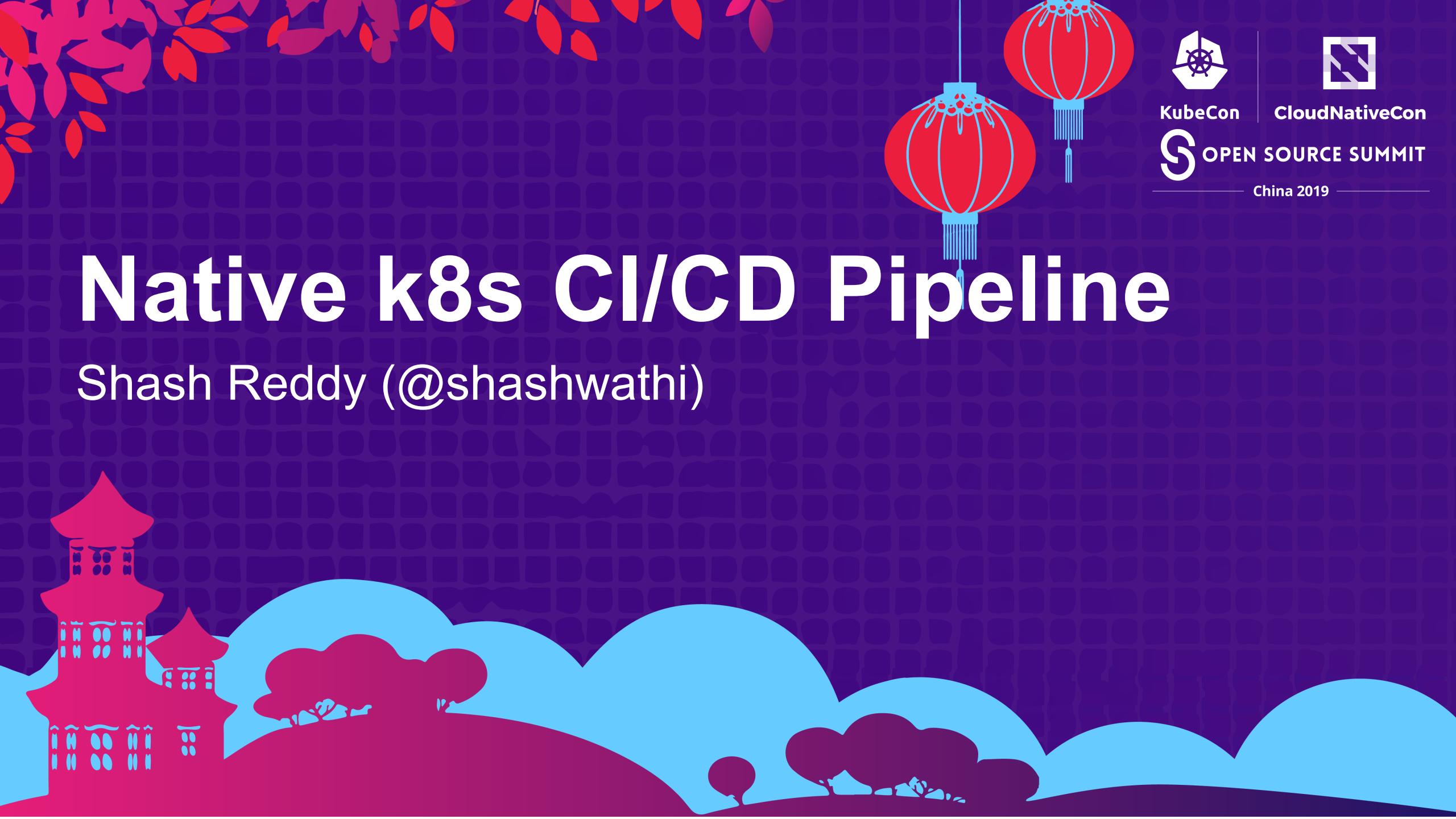
KubeCon



CloudNativeCon

 OPEN SOURCE SUMMIT

China 2019



# Native k8s CI/CD Pipeline

Shash Reddy (@shashwathi)



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2019



# Agenda

 Introduction

 Why?

 How?

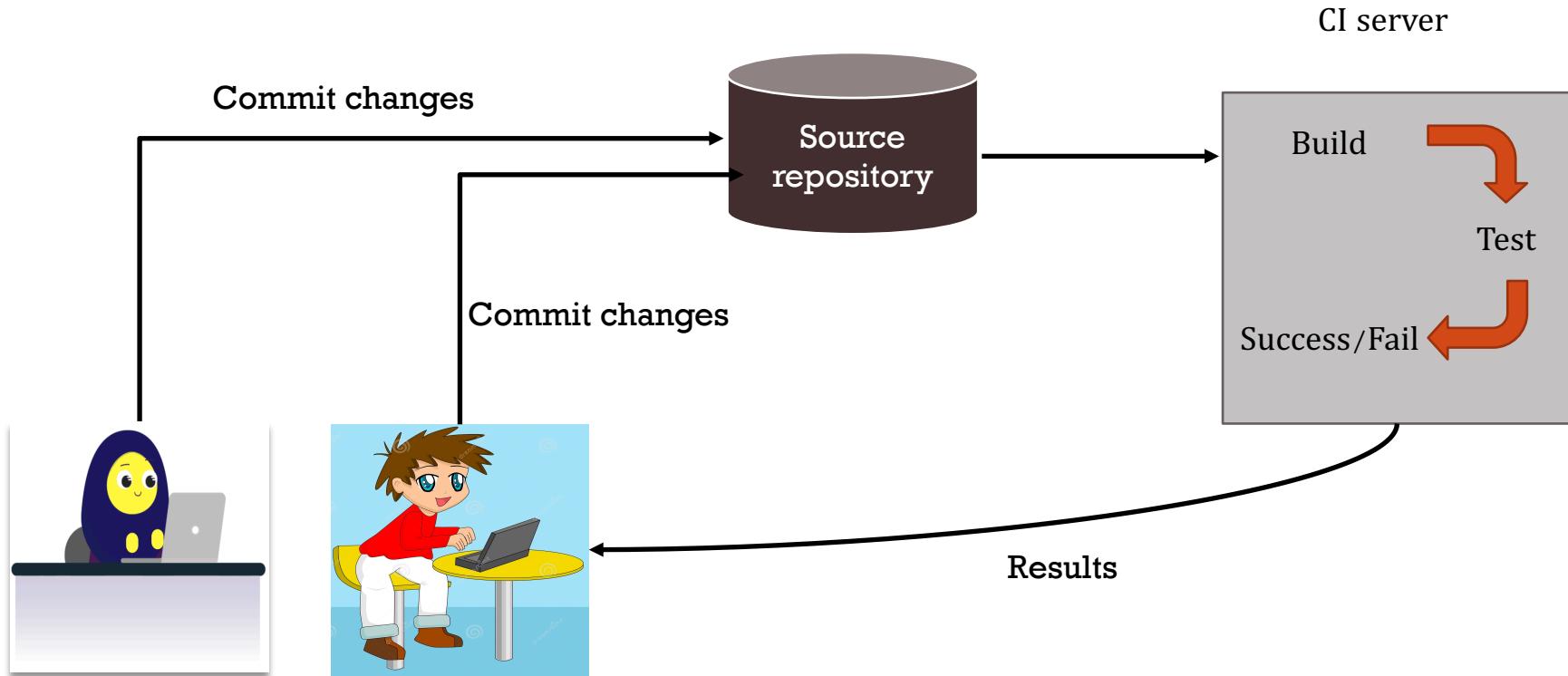
 Whaatt?

 Demo

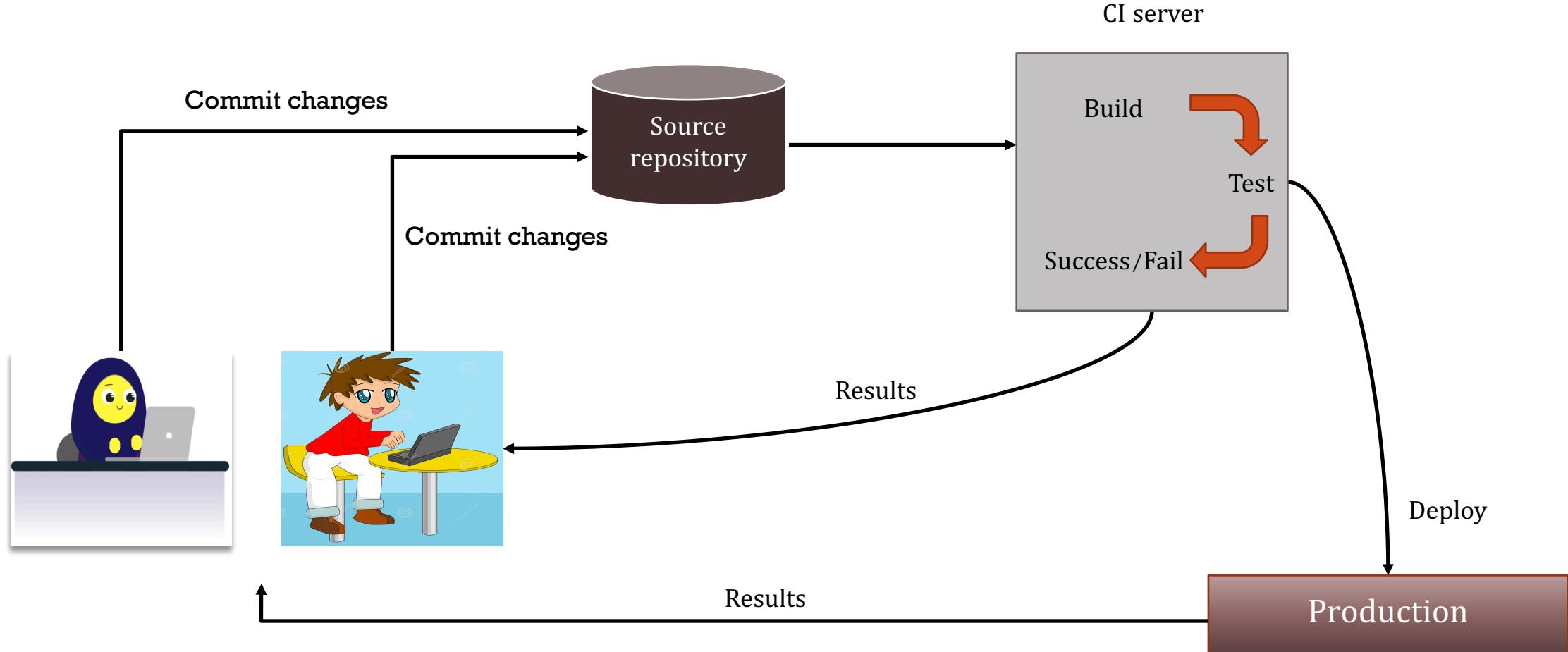
 Q&A



# What is CI?



# What is CD?





WHY?



# Why K8s?

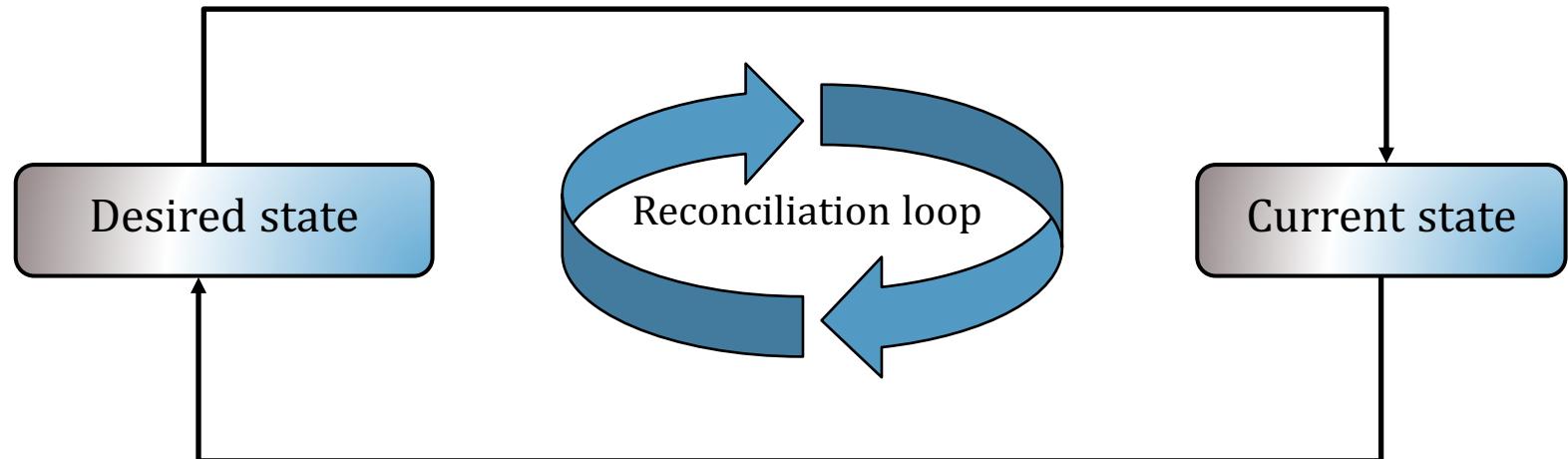
- Containerization
- Orchestration
- Observability
- Self healing system
- Cloud platform agnostic
- Identity management



HOW?

# Custom Resource Definition

- Custom Resources
- Custom Controllers



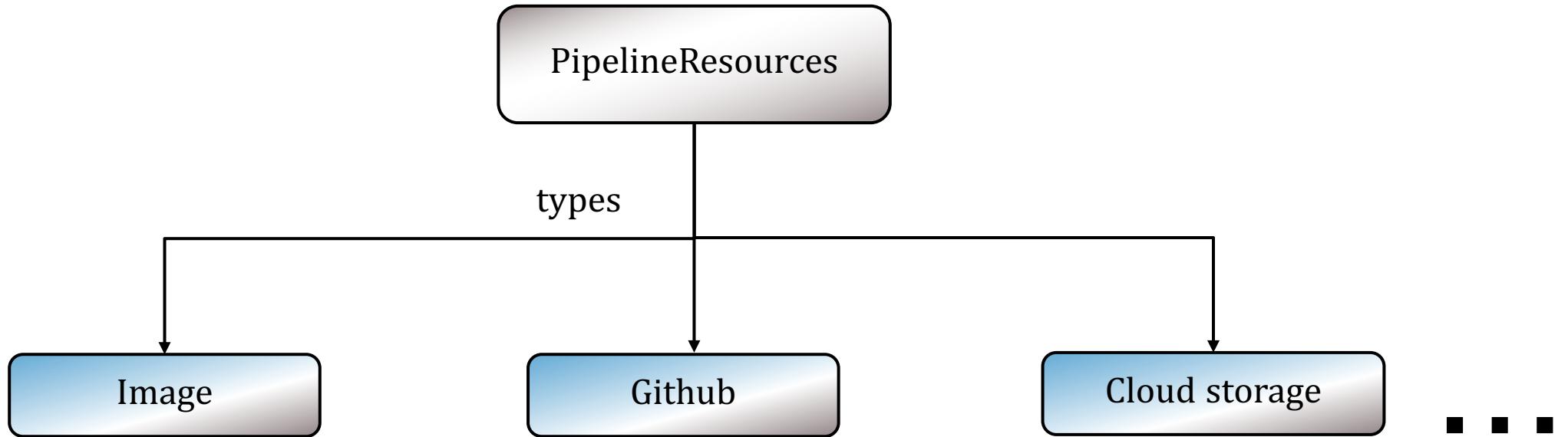


WHAATT?



# TEKTONCD PROJECT

# PipelineResource CRD



# EXAMPLE(YAML)

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineResource
metadata:
  name: myapp-source
```

**spec:**

**type:** git



Resource type

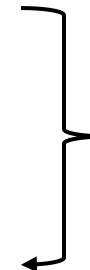
```
params:
- name: revision
  value: master
- name: url
  value: https://github.com/shashwathi/helloworld-go.git
```



# EXAMPLE(YAML)

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineResource
metadata:
  name: myapp-source
spec:
  type: git

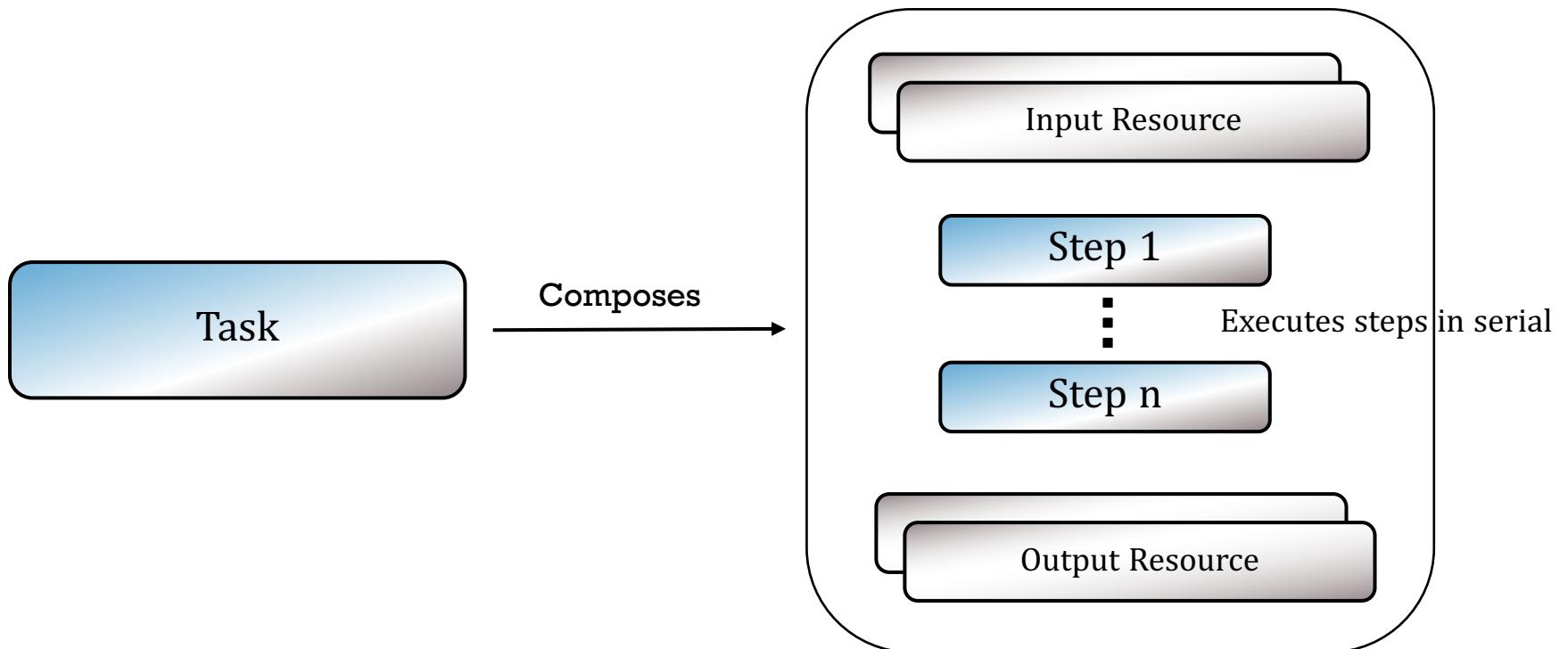
params:
  - name: revision
    value: master
  - name: url
    value: https://github.com/shashwathi/helloworld-go.git
```



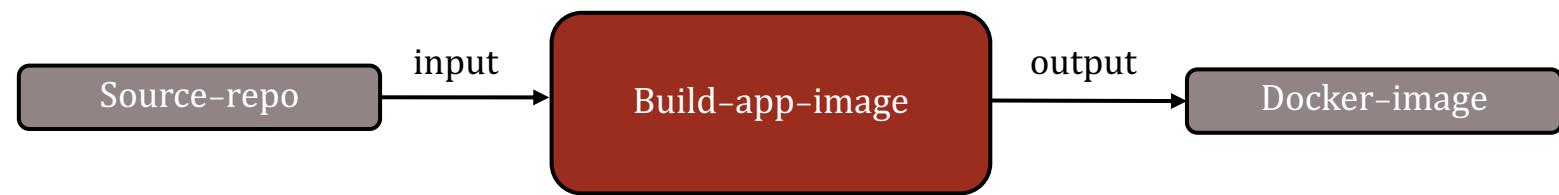
Git resource parameters



# Task CRD



# EXAMPLE



# EXAMPLE(YAML)

**spec:**

**inputs:**

**params:**

- name: DOCKERFILE

description: Path to the Dockerfile.

default: ./Dockerfile

**resources:**

- name: source

type: git

**outputs:**

**resources:**

- name: appImage

type: image

**steps:**

- name: build-and-push

workingdir: \${inputs.resources.source.path}

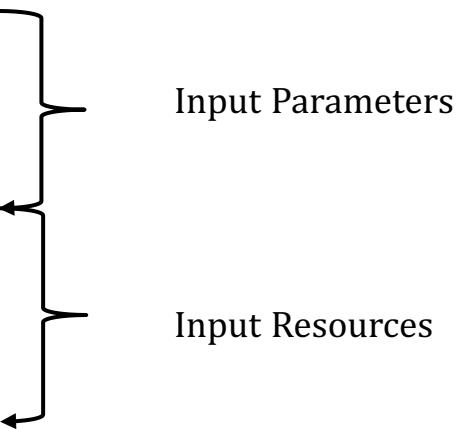
image: gcr.io/kaniko-project/executor:v0.9.0

command:

- /kaniko/executor

- --dockerfile=\${inputs.params.DOCKERFILE}

- --destination-path=\${outputs.resources.appImage.url}

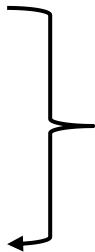


# EXAMPLE(YAML)

```
spec:  
  inputs:  
    params:  
      - name: DOCKERFILE  
        description: Path to the Dockerfile.  
        default: ./Dockerfile  
    resources:  
      - name: source  
        type: git
```

## outputs:

```
  resources:  
    - name: appImage  
      type: image
```



Output resources

```
resources:  
  - name: appImage  
    type: image
```

```
steps:  
  - name: build-and-push  
    workingdir: ${inputs.resources.source.path}  
    image: gcr.io/kaniko-project/executor:v0.9.0  
    command:  
      - /kaniko/executor  
      - --dockerfile=${inputs.params.DOCKERFILE}  
      - --destination-path=${outputs.resources.appImage.url}
```

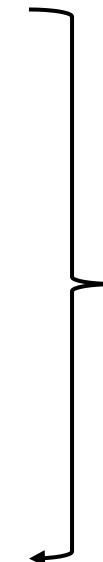


# EXAMPLE(YAML)

```
spec:  
  inputs:  
    params:  
      - name: DOCKERFILE  
        description: Path to the Dockerfile.  
        default: ./Dockerfile  
    resources:  
      - name: source  
        type: git  
  outputs:  
    resources:  
      - name: appImage  
        type: image
```

## steps:

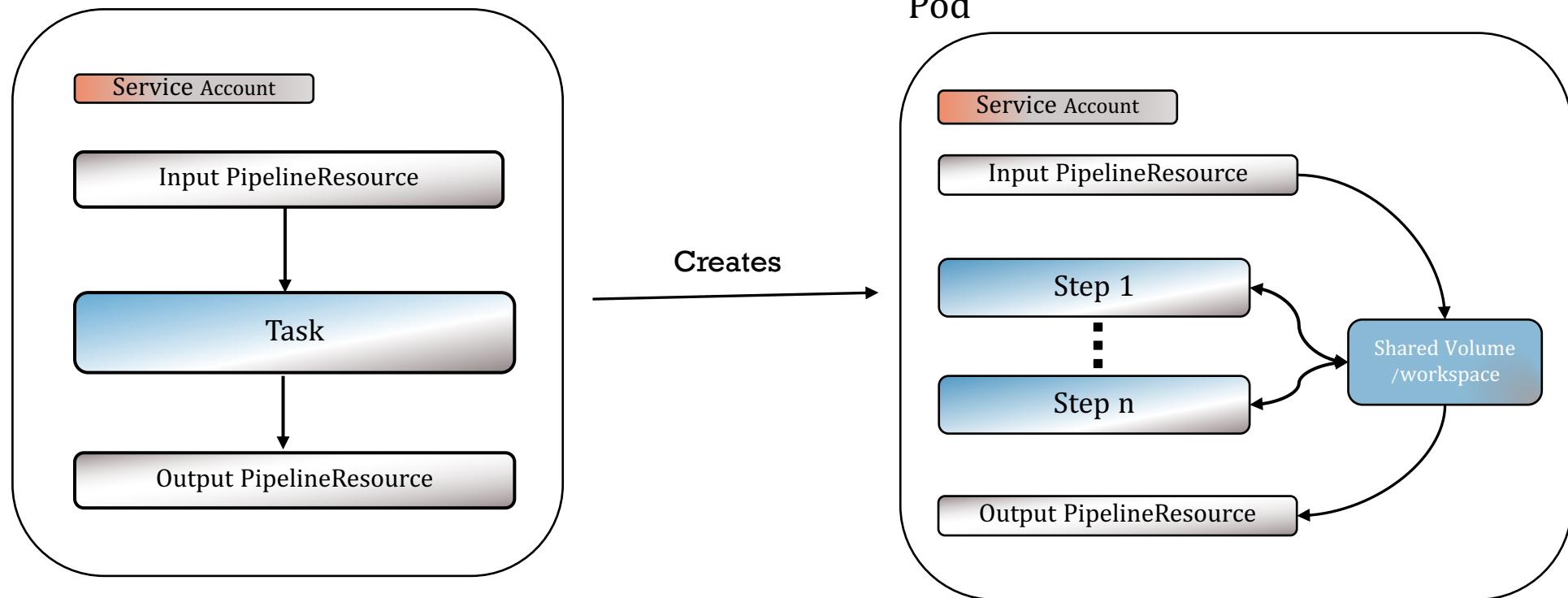
```
- name: build-and-push  
  workingdir: ${inputs.resources.source.path}  
  image: gcr.io/kaniko-project/executor:v0.9.0  
  command:  
    - /kaniko/executor  
    - --dockerfile=${inputs.params.DOCKERFILE}  
    - --destination-path=${outputs.resources.appImage.url}
```



Container definition



# Taskrun CRD



# EXAMPLE(YAML)

```
spec:  
  taskRef:  
    name: kaniko  ——————> Task reference  
  
    inputs:  
  
    resources:  
      - name: source  
  
      resourceRef:  
        name: myapp-source  
  
    outputs:  
  
    resources:  
      - name: app-image  
  
    resourceRef:  
      name: myapp-image
```



# EXAMPLE(YAML)

```
spec:  
taskRef:  
name: kaniko
```

```
inputs:  
resources:  
- name: source  
resourceRef:  
name: myapp-source
```



Input resource reference

```
outputs:  
resources:  
- name: app-image  
resourceRef:  
name: myapp-image
```

**spec:**

```
inputs:  
params: ...  
resources:  
- name: source  
type: git
```

```
outputs:  
resources:  
- name: app-image  
type: image  
steps:  
name: build-and-push  
...
```



# EXAMPLE(YAML)

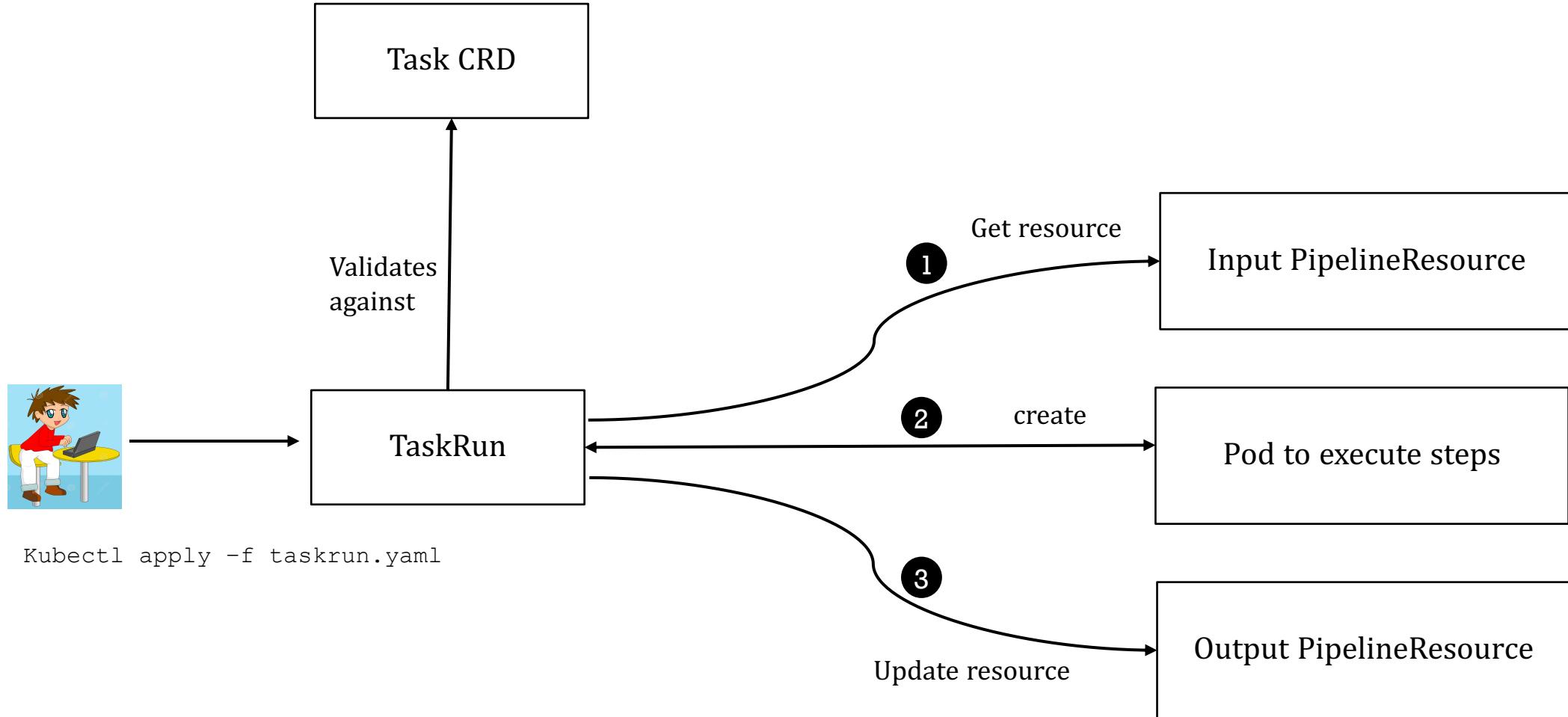
```
spec:  
  taskRef:  
    name: kaniko  
  
  inputs:  
  resources:  
    - name: source  
      resourceRef:  
        name: myapp-source  
  
outputs:  
resources:  
  - name: app-image  
    resourceRef:  
      name: myapp-image
```



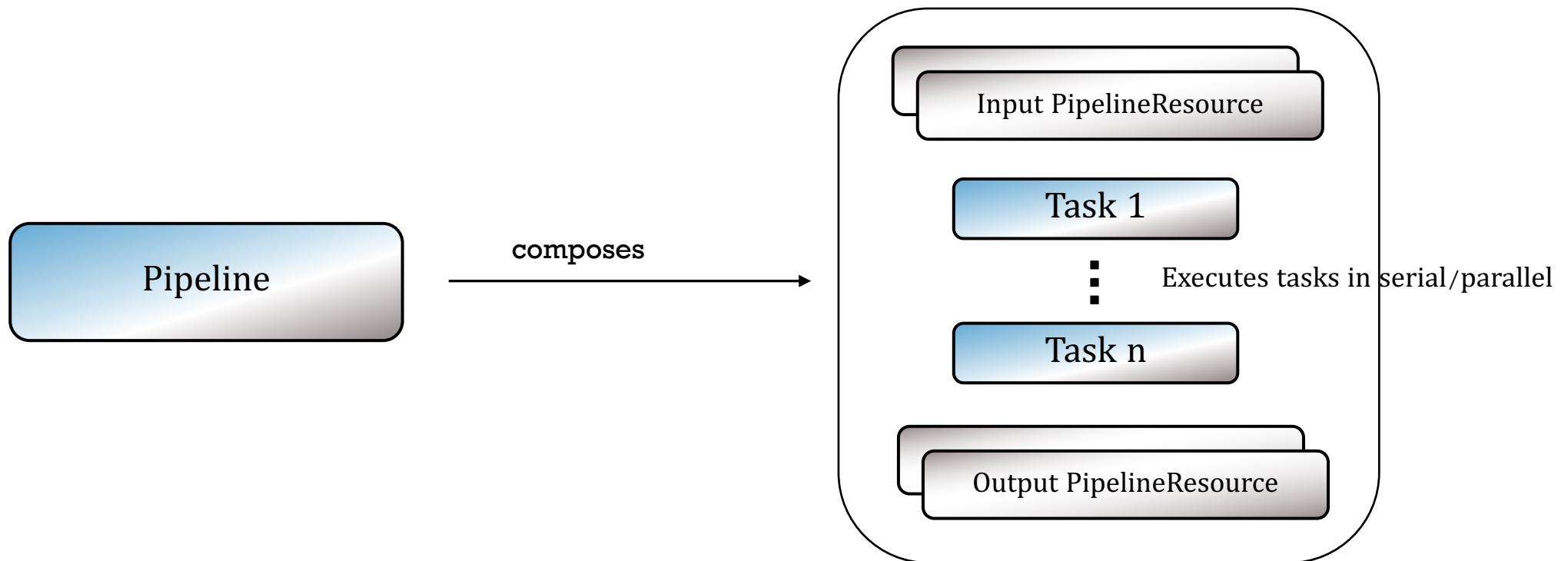
```
spec:  
  steps:  
  ...  
  inputs:  
  params: ...  
  resources:  
    - name: source  
      type: git  
  
outputs:  
resources:  
  - name: app-image  
    type: image
```



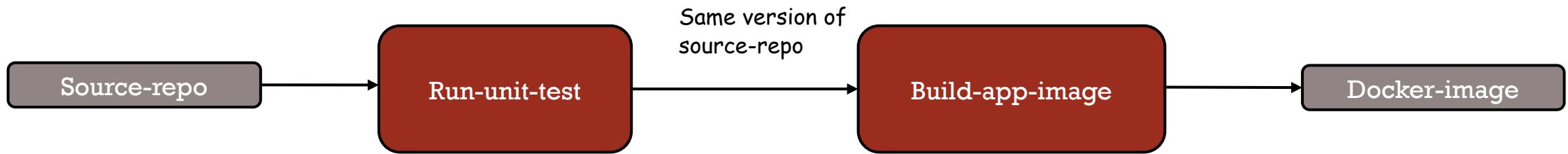
# Summary



# Pipeline CRD

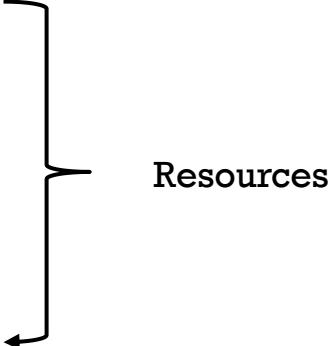


# EXAMPLE



# EXAMPLE(YAML)

```
spec:  
  resources:  
    - name: source-repo  
      type: git  
    - name: app-image  
      type: image  
  tasks:  
  ...
```



Resources



# EXAMPLE(YAML)

```
spec:  
  resources:  
    - name: source-repo  
      type: git  
    - name: app-image  
      type: image  
tasks:  
  - name: run-unit-tests  
    taskRef:  
      name: unit-tests  
resources:  
  inputs:  
    - name: workspace  
      resource: source-repo  
  - name: build-app-image  
    runAfter: [run-unit-tests]  
    taskRef:  
      name: kaniko  
  ...
```



Task for unit tests

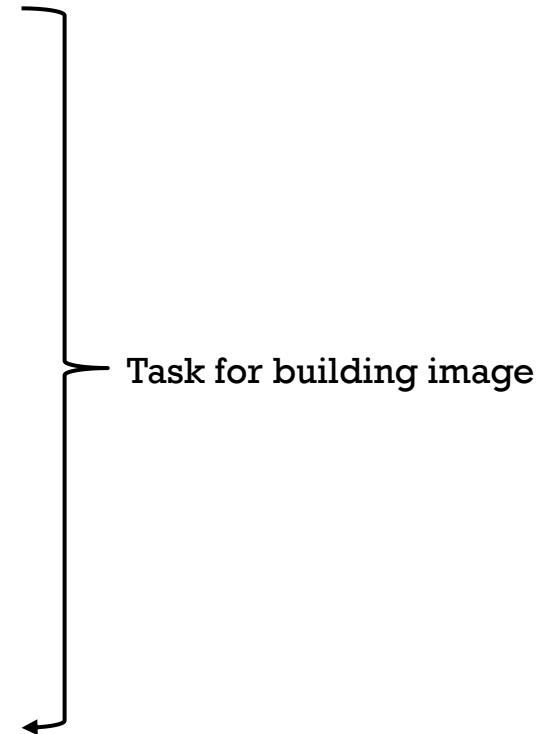


# EXAMPLE(YAML)

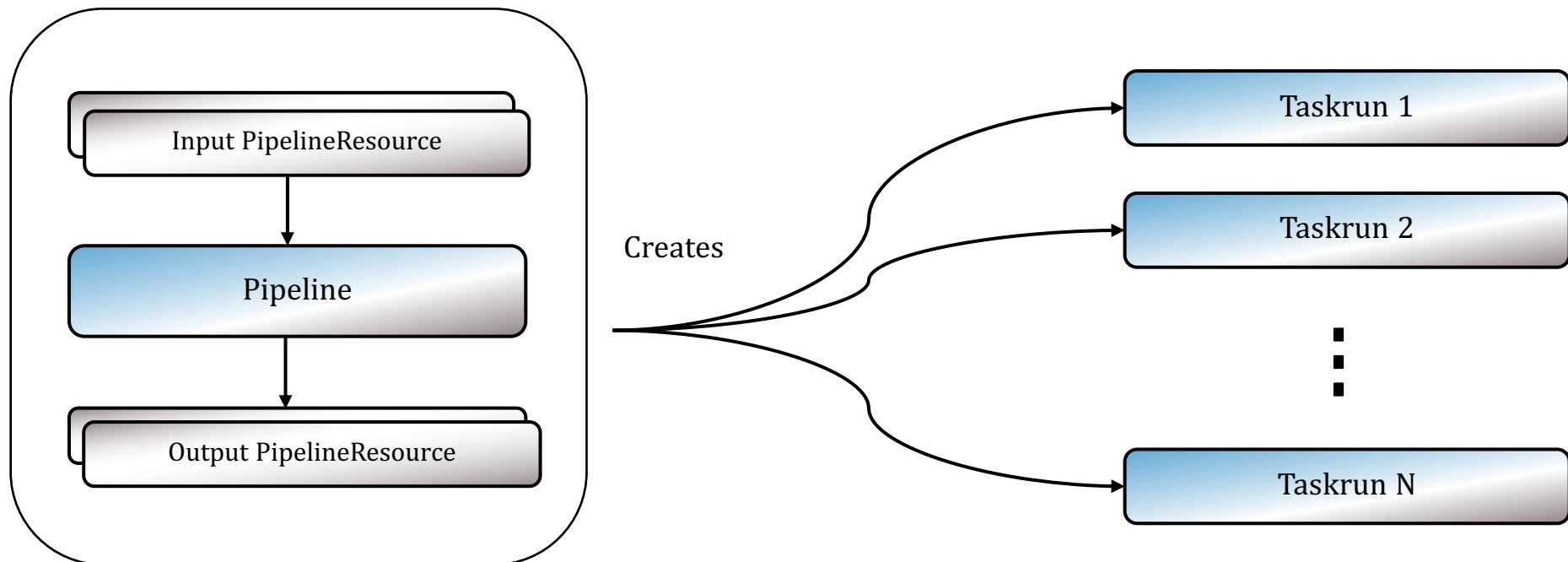
```
- name: run-unit-tests
  taskRef:
    name: unit-tests
  resources:
    inputs:
      - name: workspace
        resource: source-repo
- name: build-app-image
  runAfter: [run-unit-tests]
  taskRef:
    name: kaniko
  params:
    ...
  resources:
    inputs:
      ...
    outputs:
      ...
```



Adds dependency  
on previous task



# Pipelinerun CRD



# EXAMPLE(YAML)

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineRun
metadata:
  name: demo-pipeline-run
spec:
  pipelineRef:
    name: demo-pipeline ← Pipeline reference for flow
    serviceAccount: default
    resources:
    - name: source-repo
      resourceRef:
        name: myapp-git-source
    - name: app-image
      resourceRef:
        name: myapp-docker-image
```



# EXAMPLE(YAML)

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineRun
metadata:
  name: demo-pipeline-run
spec:
  pipelineRef:
    name: demo-pipeline
```

**serviceAccount:** default ← Credentials

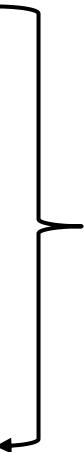
```
resources:
- name: source-repo
  resourceRef:
    name: myapp-git-source
- name: app-image
  resourceRef:
    name: myapp-docker-image
```



# EXAMPLE(YAML)

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineRun
metadata:
  name: demo-pipeline-run
spec:
  pipelineRef:
    name: demo-pipeline
  serviceAccount: default

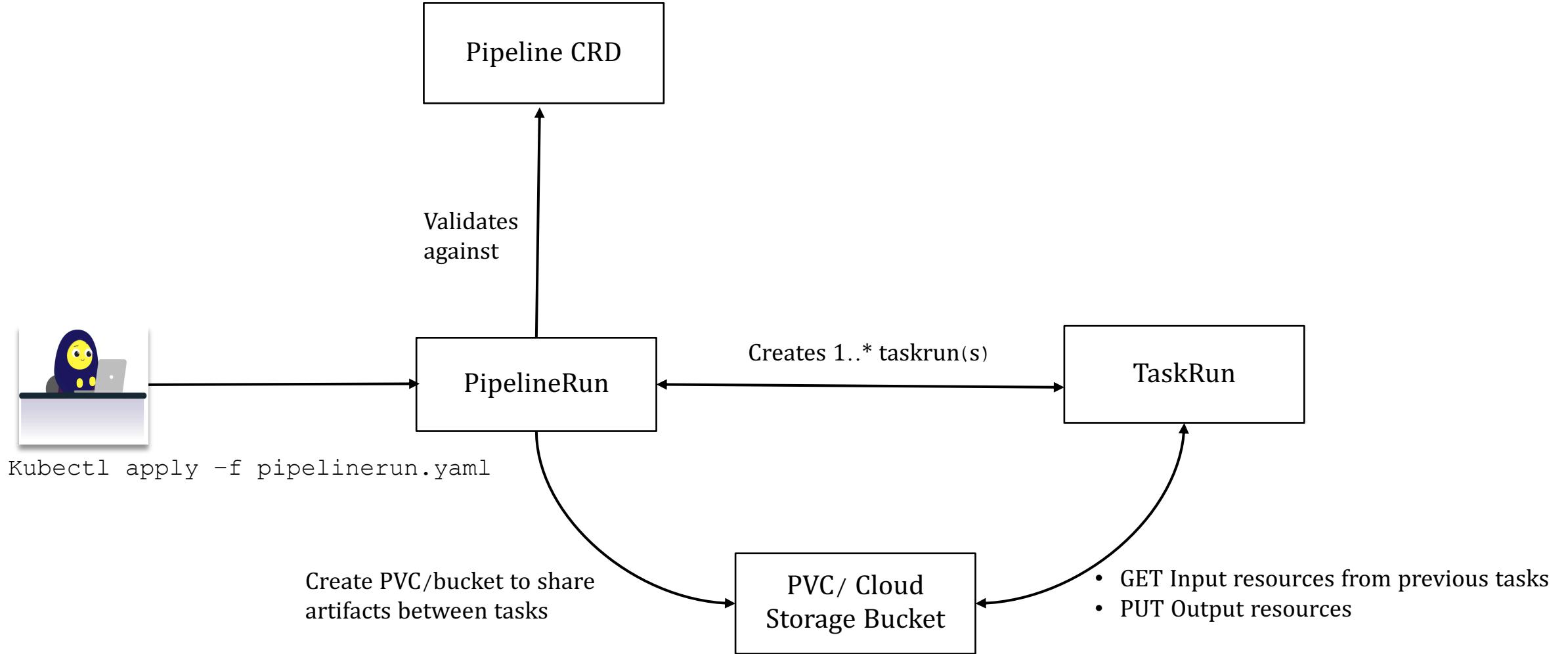
resources:
  - name: source-repo
    resourceRef:
      name: myapp-git-source
  - name: app-image
    resourceRef:
      name: myapp-docker-image
```



Binding Pipeline resources to runtime object



# Summary





**DEMO**



**Q&A**

# References

- [Tektoncd Pipeline](#) project
- [Documentation](#)
- [Catalog of tasks](#)
- [Roadmap](#) document
- [Contributors guide](#) to get started