mongoDB®

Power Up With Tekton Pipelines

Joel Lord - August 2021

CodePaLOUsa - #CPL21

Agenda for today

Intro · · · · · · · · · · · · **Set Up** · · · · · · · · · · · · **Hands-On** · · · · · · · · · · · · **Full Example**

Intro

Intro and General Information (about 10 minutes)

Set Up

Setting up the environment. (git, docker, minikube, kubectl, tkn) Installing Tekton (about 20 minutes)

Hands-On

Hands-on examples of Tekton components.

- Tasks
- Pipelines
- Pipeline Resources
- Workspaces
- When Expressions

Full Example

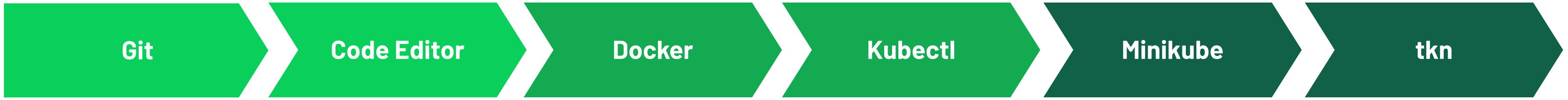
If time permits, a full example of a Tekton pipeline

Requirements

You should already be familiar with containers
You should have basic knowledge of Kubernetes

Environment Setup

Setting up your environment



Setting up your environment

Git

Installation: https://git-scm.com/

Setting up your environment

Code Editor

Installation: https://code.visualstudio.com/
Tekton extension:
https://github.com/redhat-developer/vscode-tekton

Setting up your environment

Docker

Installation: https://www.docker.com/get-started
Alternative: https://podman.io

Setting up your environment

Kubectl

Installation:
https://kubernetes.io/docs/tasks/tools/install-kubectl/

```
$ kubectl --version
```

Setting up your environment

Minikube

You can use your own cluster if you have one available
Ask your instructor for credits on various platforms

Installation: https://minikube.sigs.k8s.io/docs/start/

```
$ minikube start
```

Setting up your environment

tkn

Installation: https://github.com/tektoncd/cli/releases

```
$ tkn version
```

Setting up your environment

Tekton CRD's

Installation: https://tekton.dev/docs/getting-started/

Aboot me, eh?

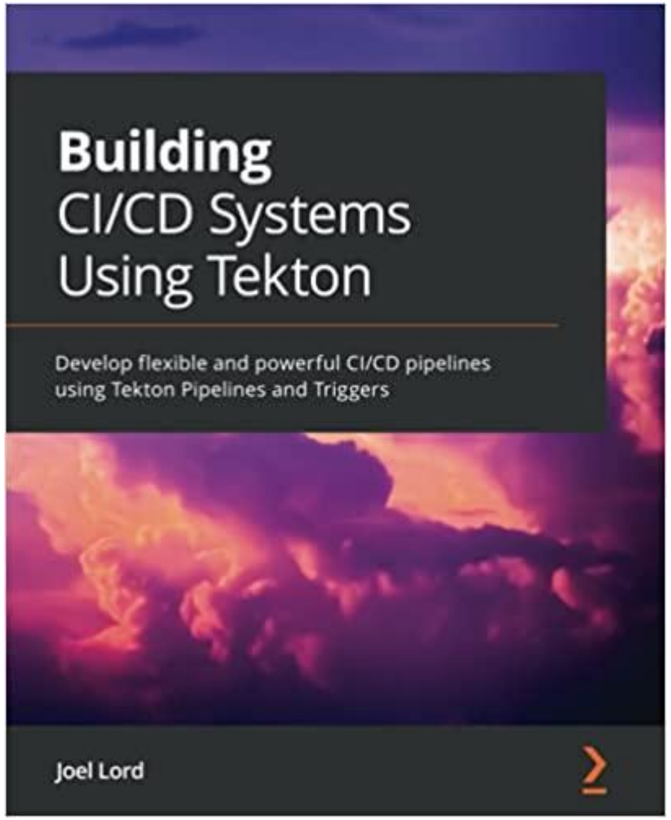
Hi! I’m Joel!

Developer Advocate at MongoDB

Based in Canada

💙 Twitter: @joel__lord

Aboot me, eh?


Building CI/CD Systems Using Tekton

Develop flexible and powerful CI/CD pipelines using Tekton Pipelines and Triggers

Joel Lord

https://www.amazon.com/Building-Systems-Using-Tekton-pipelines-dp-1801078211/dp/1801078211

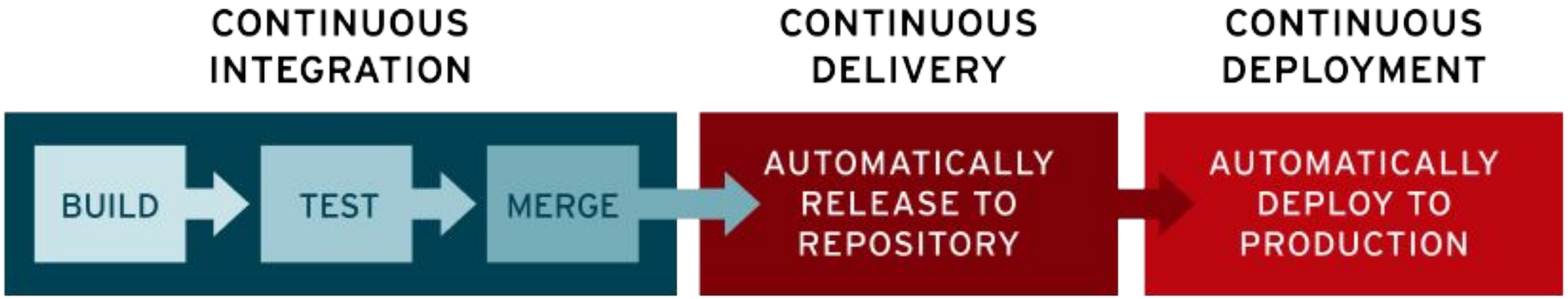
What is CI/CD

What is CI/CD

CI/CD introduces ongoing automation and continuous monitoring throughout the lifecycle of apps, from integration and testing phases to delivery and deployment.

https://www.redhat.com/en/topics/devops/what-is-ci-cd

What is CI/CD



https://www.redhat.com/en/topics/devops/what-is-ci-cd

What about Cloud-Native CI/CD?

What about Cloud-Native CI/CD?

Containers

Built for container apps and runs on Kubernetes

Serverless

Runs serverless with no CI/CD engine to manage and maintain

DevOps

Designed with microservices and distributed teams in mind

What about Cloud-Native CI/CD?

Jenkins

JENKINS X

Spinnaker

http://cd.foundation

Introducing Tekton

http://tekton.dev

Introducing Tekton

Composable

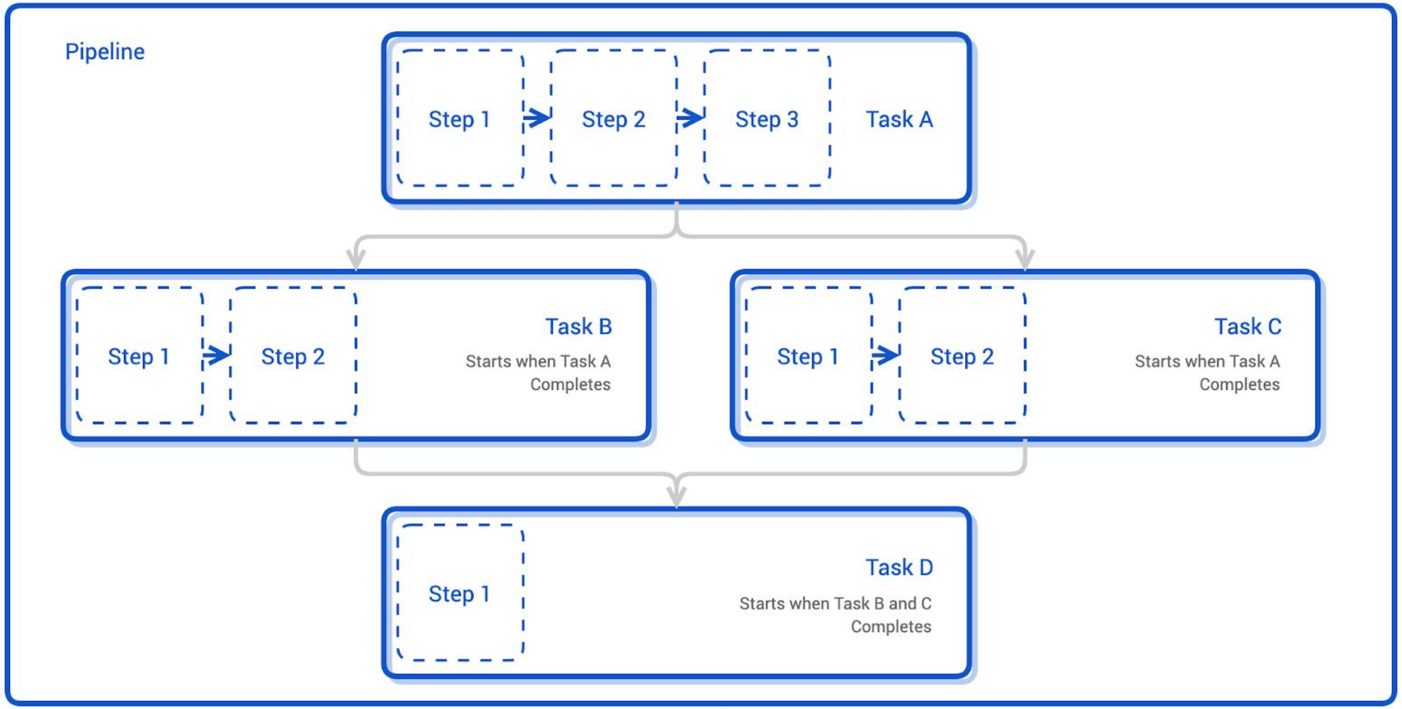
Declarative

Reproducible

Cloud Native

Tekton Building Blocks

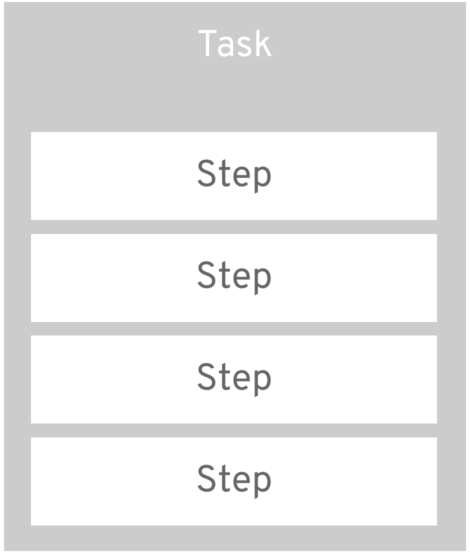
Tekton Building Blocks


Pipeline
Step 1
Step 2
Step 3
Task A
Step 1
Step 2
Task B
Starts when Task A Completes
Step 1
Step 2
Task C
Starts when Task A Completes
Step 1
Task D
Starts when Task B and C Completes

Tasks

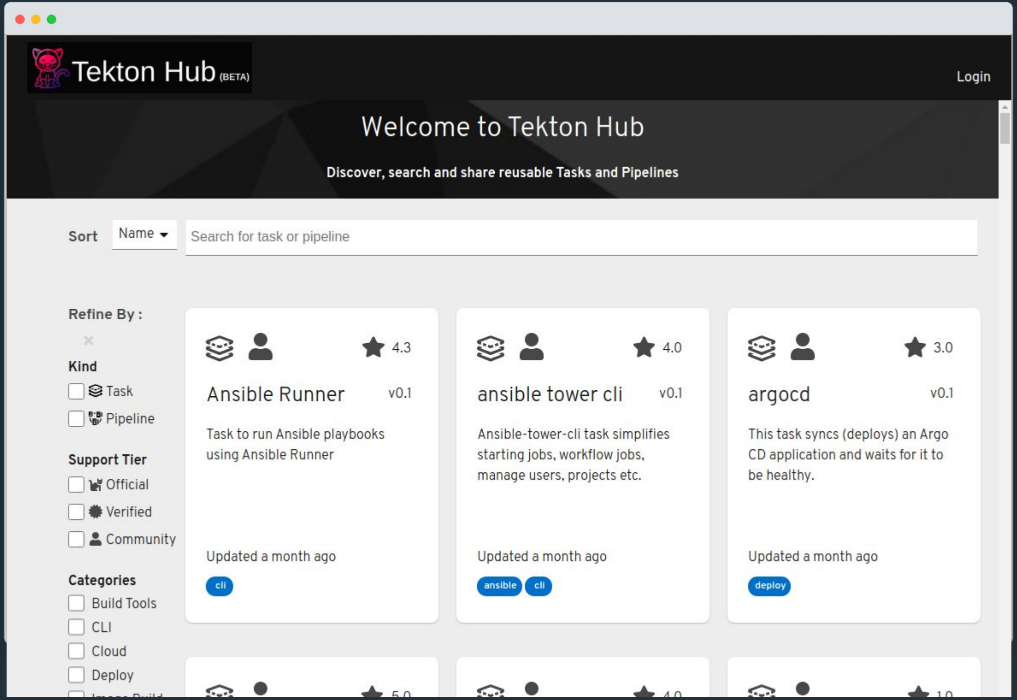
Tasks

- Defines a unit of work to be executed
- A list of steps to run sequentially
- Step containers run in the task pod
- Has inputs, outputs and parameters
- Workspaces and results for sharing data
- Can run independent of pipelines



Tasks Catalog

https://hub.tekton.dev


Tekton Hub (BETA)

Login

Welcome to Tekton Hub

Discover, search and share reusable Tasks and Pipelines

Sort: Name

Search for task or pipeline

Refine By :

Kind
- Task
- Pipeline

Support Tier
- Official
- Verified
- Community

Categories
- Build Tools
- CLI
- Cloud
- Deploy

Ansible Runner v0.1 — 4.3
Task to run Ansible playbooks using Ansible Runner
Updated a month ago
cli

ansible tower cli v0.1 — 4.0
Ansible-tower-cli task simplifies starting jobs, workflow jobs, manage users, projects etc.
Updated a month ago
ansible cli

argocd v0.1 — 3.0
This task syncs (deploys) an Argo CD application and waits for it to be healthy.
Updated a month ago
deploy

Tasks - Steps

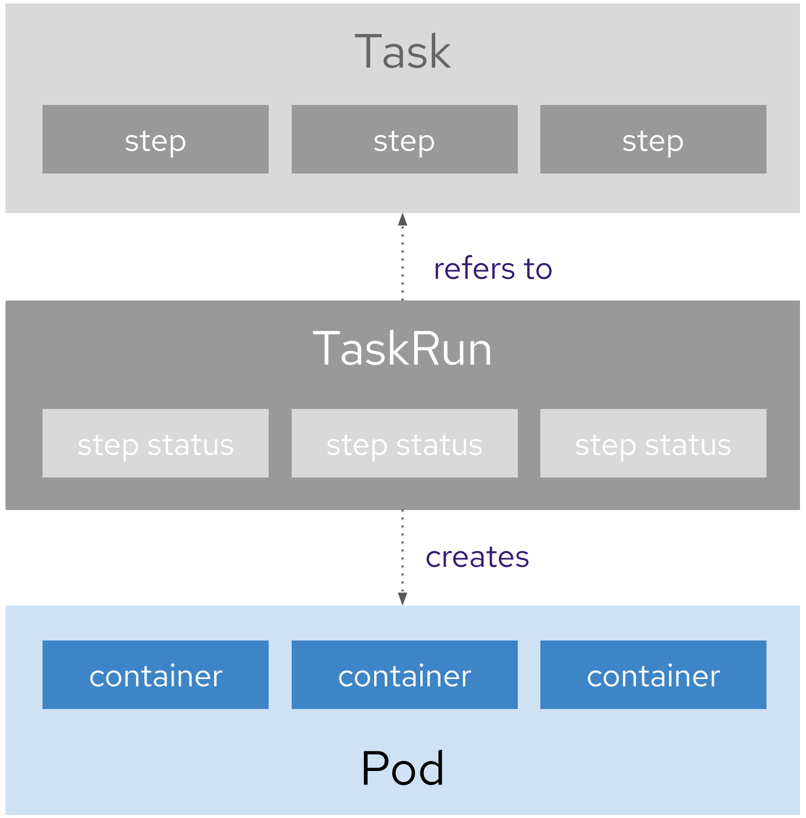
- Run command or script in a container
- Kubernetes container spec
  - Env vars
  - Volumes
  - Config maps
  - Secrets

```yaml
  - name: greeting
    image: registry.access.redhat.com/ubi8/ubi
    command:
      - "/bin/bash"
    args:
      - "-c"
      - "echo Welcome to the second task"
```

```yaml
  - name: read
    image: registry.access.redhat.com/ubi8/ubi
    script: |
      #!/usr/bin/env bash
      echo "Reading from
$(workspaces.files.path)/$(params.filename)"
      cd $(workspaces.files.path)
      cat $(params.filename)
```

TaskRuns

- Runs a Task to completion in a pod
- References or embeds a Task spec
- Provides input to Tasks
- Parameters
- Service account
- Workspaces
- Contains execution status and metadata



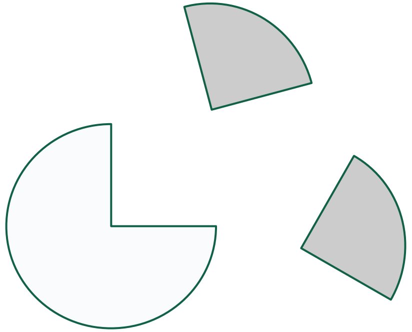
Tasks

Hands-On Exercise

Parameters

Parameters are supplied to the Task at execution time

You can access them with variable substitution `$(params.<name>)`



Tasks - Parameters

Hands-On Exercise

Shared Volumes

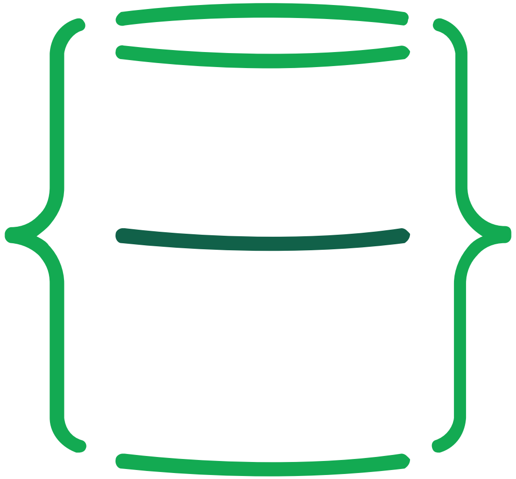
```
cd ~

echo $(pwd)

// /tekton



/tekton/results can be used
for $(results)
```



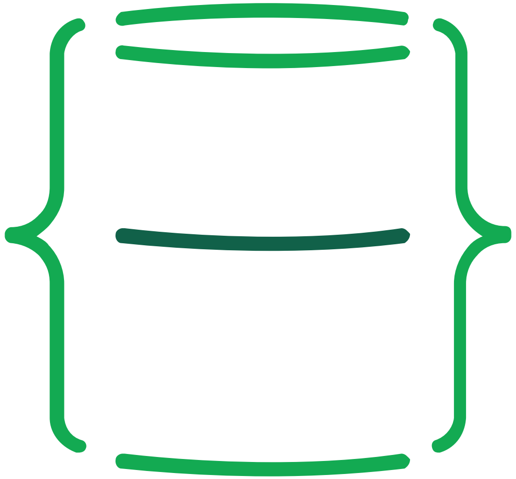
Shared Volumes

```
cd ~
```

DEPRECATED

```
// /tekton
```

```
/tekton/results can be used
for $(results)
```



Tasks - Shared Volumes

Hands-On Exercise

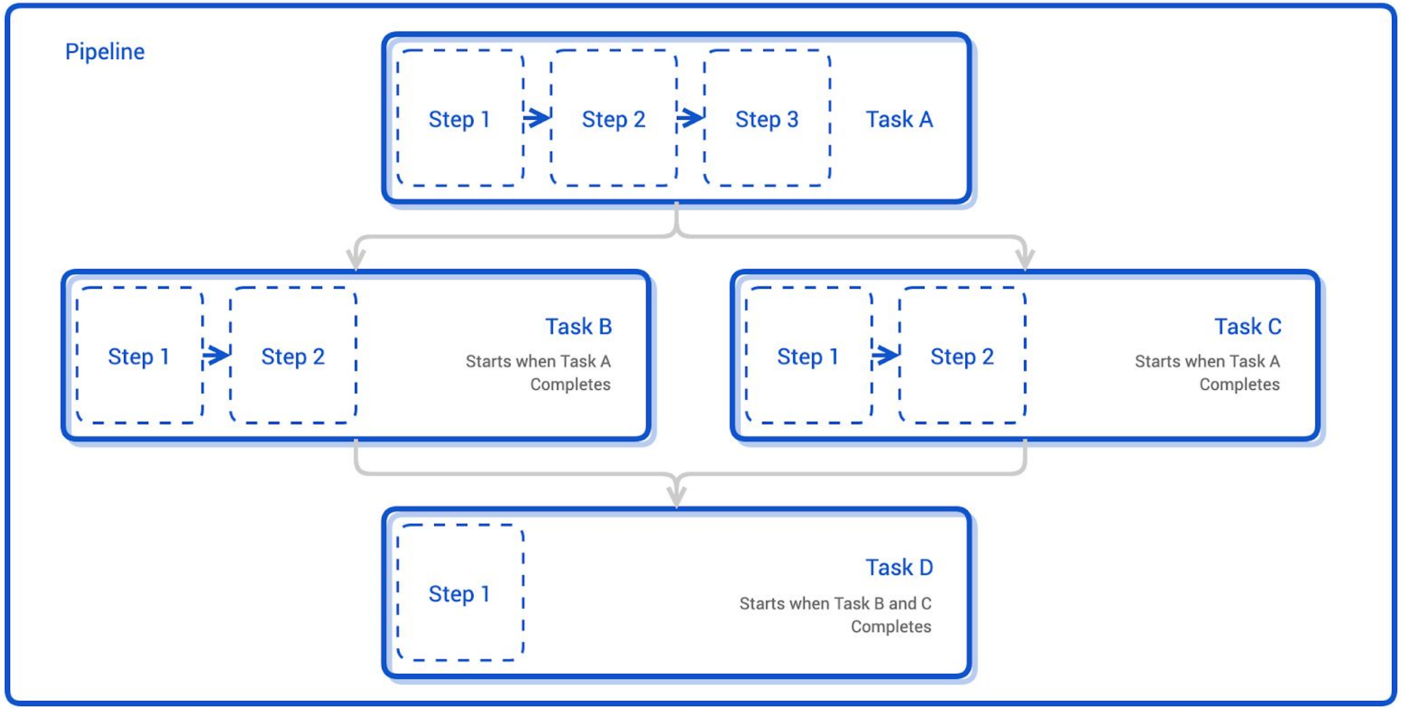
Exercise

Write your own Task

- It should output a greeting message to a user whose name is passed as a parameter
- That greeting should only output after a configurable sleep step
  - The sleep step would be your first step
  - It will require a sleep duration parameter
  - It will use the `sleep` bash command

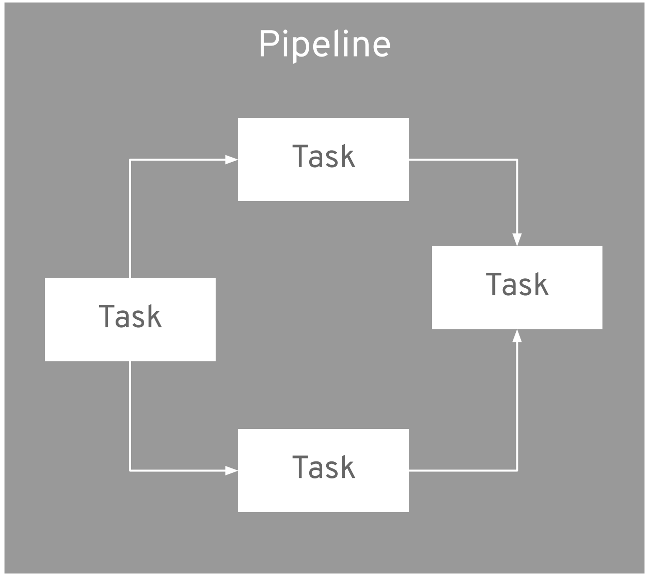
Pipelines

Tekton Building Blocks


Pipeline
Step 1
Step 2
Step 3
Task A
Step 1
Step 2
Task B
Starts when Task A Completes
Step 1
Step 2
Task C
Starts when Task A Completes
Step 1
Task D
Starts when Task B and C Completes

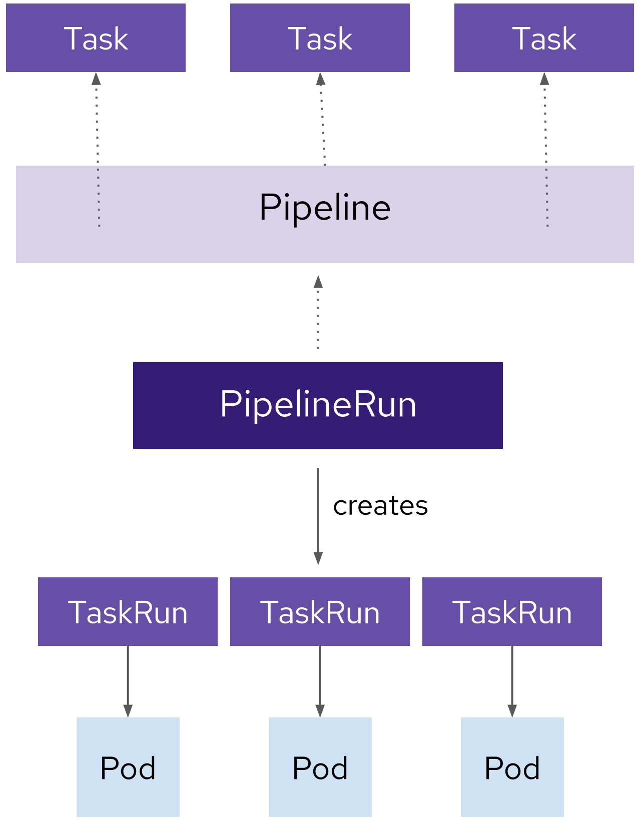
Pipelines

- Define Tasks execution order (graph)
- Inputs and parameters
- Retries tasks
- Conditional task execution
- Workspaces for sharing data between tasks
- Reusable across projects



PipelineRuns

- Runs a pipeline to completion
- References or embeds a Pipeline spec
- Creates TaskRuns to execute Tasks in the Pipeline
- TaskRun pods may get scheduled on different node
- Provides inputs and params to pipeline
- Provides volumes for declared pipeline workspaces



Pipelines

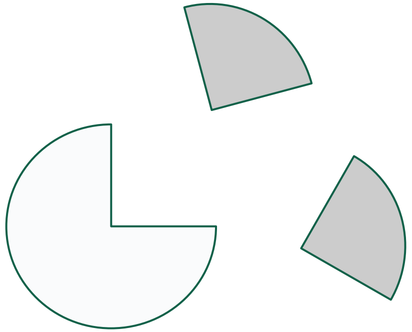
Hands-On Exercise

Pipeline Parameters

Pipelines can also have parameters

They can be passed to Task Parameters

With the CLI, you can use the `-p` option or `--use-param-defaults`

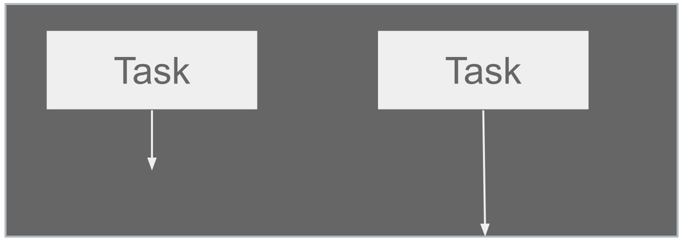


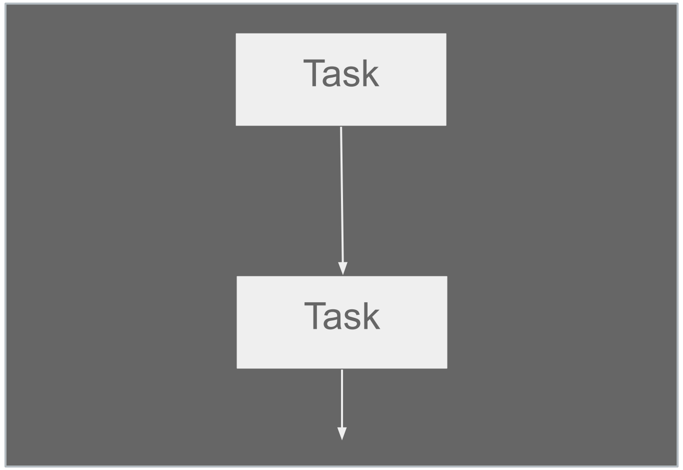
Pipelines - Parameters

Hands-On Exercise

Task Reordering

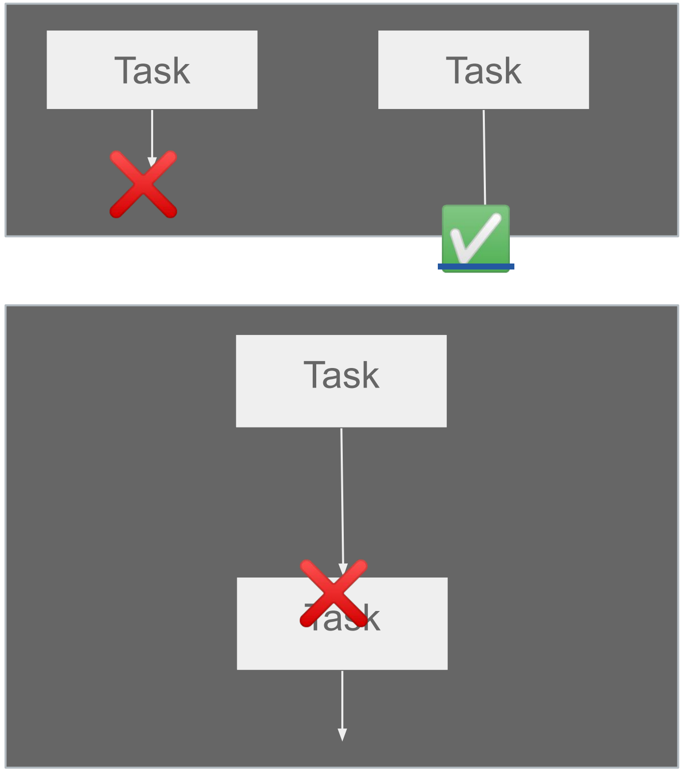
- Racing conditions
- You can adjust the order
- runAfter





Task Reordering

- Racing conditions
- You can adjust the order
- runAfter

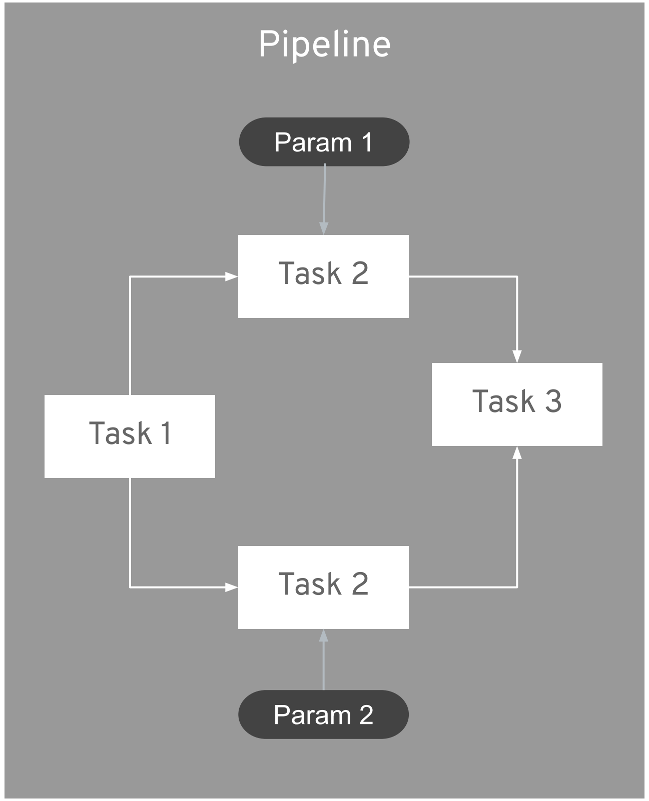


Pipelines - Task Reordering

Hands-On Exercise

Task Reusability

You can reuse tasks, even within a single Pipeline



Pipelines - Task Reusability

Hands-On Exercise

Exercise

Write your own Pipeline

- Given the `gather` and `spread` tasks, build a pipeline to make a PB&J sandwich
- Reuse tasks when you can and ensure correct ordering
- Add an additional parameter to the Pipeline to echo who is eating the sandwich

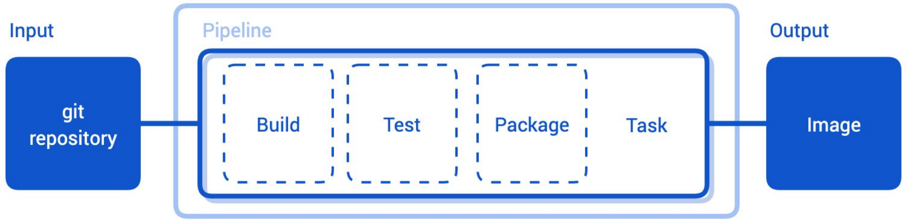
PipelineResources

PipelineResources

Still in *alpha*

Serves as an input or output to the Pipeline

Reusable and configurable



Workspaces

Workspaces

- Share a volume between Tasks (for Pipelines) or across Steps (for Tasks)
- Can have many types
  - volumeClaimTemplate (for PipelineRuns)
  - persistentVolumeClaim
  - emptyDir (for Tasks)
  - configMap
  - secret

Workspaces

- Variable substitutions
  - `$(workspaces.<name>.path)`
  - `$(workspaces.<name>.bound)`
  - `$(workspaces.<name>.claim)`
  - `$(workspaces.<name>.volume)`
- Remember, you are responsible for the Task sequence

Persistent Volume Claims

- You must create a PV and a PVC prior to running the Pipeline
- Instructions may differ on various cloud providers

Workspaces - PVC

Hands-On Exercise

Claim Templates

- You may create a PipelineRun directly
- Lets you use templates in the workspaces
- You must use
  - `kubectl create`

Workspaces - ClaimTemplates

Hands-On Exercise

Exercise

Write a PipelineRun that uses a Workspace

- Using the git-clone task from the Catalog, clone the repository https://github.com/joellord/tekton-lab-sample
- Output the content of sample.txt in a subsequent Task
- Use a PipelineRun and a workspace template

WhenExpressions

WhenExpressions

- Add conditional statements to your Pipelines
- Blocks the execution of Tasks based on conditions
- Replacement candidate for Conditionals

WhenExpressions

Hands-On Exercise

Exercise

Tweak your last Pipeline with a WhenExpression

- Output a warning (using a new Task) when the branch that is cloned is \`development\`

Triggers

Task Reusability

Create WebHooks that can be triggered based on Events

Trigger a Pipeline when a commit is performed on the master branch of your git repository

Uses Ingresses or Routes

Putting it all together

Exercise

Real World Example

- Clone the repository
  https://github.com/joellord/tekton-lab-app
- Share the source code in a shared Workspace
- Run npm install, npm test, npm lint
- Use podman to build the Docker image and push to registry