Kubernetes Native DevOps Practice

— 王磊 @TenxCloud
Agenda

• Our DevOps Expectations
• Kubernetes Capabilities/Advantages to Build DevOps Solution

  • Architecture and Features
    • CRD and operator design
    • Pipeline / Stage / Task / Task Template / Version Control
    • Logging, monitoring, autoscaling, high availability
    • Extensibility / Integration
    • CI/CD examples

• Future plan
Our DevOps Expectations

- Build a platform and easy to integrate with other DevOps/third-party tools
- Easy to be customized as user requirements are diverse
- Easy to setup, maintain, extend and scale
- Reduce the learning curve for customers and ourselves
- Get consistent user experience and data, leverage with PaaS capability
- Facilitate our PaaS and micro-service product
Kubernetes Capabilities/Advantages to Build DevOps Solution

• k8s itself is NOT a PaaS or DevOps platform, but …
• k8s resources that can be used to build DevOps solution

Pod
Job
CronJob
Volumes

ConfigMap
Secret
ResourceQuota / LimitRanges

• Scheduler / Affinity
• And more …
Pod Spec

- **InitContainers**: Initialize the build environment
- **Affinity / SchedulerName**: Configure scheduler policy
- **Volumes**: Share files between containers, or cache build files
- **ActiveDeadlineSeconds**: Timeout of build task
- **Containers**: Build tasks and the dependent environments (sidecar)

**Container**

- **Image**: Image of build / dependent environment
- **Command**: Command to execute
- **Args**: Argument
- **Env**: Environment variable
- **VolumeMounts**: Files to be shared or persisted
- **Resources**: Resource requirement
- **Lifecycle**: Actions defined for postStart/preStop
Kubernetes Capabilities/Advantages to Build DevOps Solution

<table>
<thead>
<tr>
<th>Kubernetes Capabilities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>parallelism</strong></td>
<td>Expected maximum number of parallel build tasks</td>
</tr>
<tr>
<td><strong>completions</strong></td>
<td>Expected number of completed build tasks</td>
</tr>
<tr>
<td><strong>Job Spec</strong></td>
<td></td>
</tr>
<tr>
<td>activeDeadlineSeconds</td>
<td>The timeout of the running build tasks, default: 12 hours</td>
</tr>
<tr>
<td>backoffLimit</td>
<td>Maximum retry count before mark the build task as failed, default: 6</td>
</tr>
<tr>
<td><strong>v1.12 alpha</strong></td>
<td></td>
</tr>
<tr>
<td>ttlSecondsAfterFinished</td>
<td>Time to clean up finished build tasks after if finishes</td>
</tr>
<tr>
<td><strong>Pod Template</strong></td>
<td></td>
</tr>
<tr>
<td>schedule</td>
<td>Cron style scheduler configuration</td>
</tr>
<tr>
<td>concurrencyPolicy</td>
<td>Concurrency policy of CronJob</td>
</tr>
<tr>
<td><strong>CronJob Spec</strong></td>
<td></td>
</tr>
<tr>
<td>suspend</td>
<td>Whether suspend latter jobs if the previous job is still running</td>
</tr>
<tr>
<td>successfulJobsHistoryLimit</td>
<td>Number of successful/failed history jobs to keep</td>
</tr>
<tr>
<td>failedJobsHistoryLimit</td>
<td></td>
</tr>
</tbody>
</table>

Job Template

CronJob Spec
Agenda

• Our DevOps Expectations
• Kubernetes Capabilities and Advantages to Build DevOps Solution

• **Architecture and Features**
  - CRD and operator design
  - Pipeline/Stage/Task/Task Template/Version Control/UI generation/Volume...
  - Logging, monitoring, autoscaling, high availability
  - Extensibility/Integration
  - CI/CD examples

• Future plan
Overall Architecture

- Kubernetes Cluster
  - Node
  - Pod
  - CronJob
- Kubernetes Cluster
  - Job
  - Pod
- Scheduling customization
  - kubectl can do image/container GC

- Logging Service
  - Monitor/Alert Service
  - Unified logging, monitoring, alert with PaaS
  - Consistent data

- Node group of build nodes
- Node group of user applications

DevOps Operator

- DevOps Manager
- docker registry

MySQL

- Pipeline configuration and history in MySQL
- Logging in central logging service - ElasticSearch
- Metric data in monitoring system - prometheus
- Alertmanager to invoke various alert and related actions
CRD and Operator Design

Kubernetes Cluster

DevOps Manager

MySQL

DevOps Operator

Create job

BuildJob

Update jobs status to buildjob

Submit buildjob

List/Watch buildjob

Pipeline / Stage / Task

Task Template

Pipeline / Stage / Task build logs

Version Control

Build task configuration - map to k8s Job, can also be a raw k8s job

Job / Pod / Node info

Job / BuildJob / Job status

sync / watch

clean history jobs
Basic Concepts (partial)

Repository

Managed Project

Task created from template

Pipeline / Stage / Task

Tasks in same stage can run sequentially or in parallel

Task Template

Dockerfile / Scripts

Data Volume

Common Configuration

ConfigMap/Secret

PVC
• Build task can also expose custom metric data
• Ephemeral build task can push metric to gateway if needed
• Cluster autoscaler will add/remove node from build group for scaling
• HA is guaranteed by cluster HA, k8s Job controller and cluster autoscaler, can also use container probe if needed
Extensibility / Integration

- Easy to extend task template, just docker images
- Use can define raw k8s job, more features can be enabled
- Leverage all k8s resources and extensibility to fulfill various requirements, prompt innovation
- New features of k8s may help enhance or improve DevOps

  - Help integration test - use sidecar container as dependent environment
  - Encapsulate API / SDK of other tools using image for better integration/collaboration
  - Leverage k8s integration capabilities, such as external service registration, service catalog, etc…
CI/CD Examples - Build Docker Image

- **dockerfile using ConfigMap**

**DevOps Operator**
Manage the Job

**Job - pod template**
- **init task**
  - prepare code repository
  - volumes

- **user build task**
  - build the docker images
  - volumes
  - resources

**Environment variables**

**Image information**

**Memory / CPU / GPU**

**Storage Service**
- rbd / nfs / glusterfs

**registry credential using secret**

**pvc**
CI/CD Examples - Artifact Management

**Job - pod template**
- init task
  - prepare code repository
- user build task
  - build the application package
- sidecar build task
  - upload files to storage service once user build completes

**DevOps Operator**
- Manage the Job

**Storage APIs**
- storage config using secret

**Query artifact data**
CI/CD Examples - Human/Manual Task

User Action

DevOps Manager

Authorization check

Approve/Deny to make job successful or failed

Job / Pod

- send out email if needed
- wait for user actions

- approver list
- activeDeadlineSeconds
- volumes

- system email config using secret

- environment variable

- approval timeout

- task name: approval task
- approver: admin
- admin email: admin@tenxcloud.com
- approval timeout: 5 days
- environment: DevTest
CI/CD Examples - Continuous Deployment

**Upgrade an existing service**
- Use helm template
- Deploy using kubectl or k8s API
- Deploy using helm client
- Call Spinnaker webhook

**Deploy using helm template**
- Use helm template
- Deploy using Spinnaker

**Deploy using Spinnaker**
- Rolling Upgrade / Recreate
- Deploy / Upgrade using helm, rollback if unready pods
- Check tracing data, rollback if get unexpected failure rate
- Call Spinnaker API to start deployment pipeline

Security policy for the build task
Events Trigger with payload

- Gitlab webhook
- On schedule
- Registry notification

Encapsulate API / SDK of third party tools to docker image
Pass events from other system to build task, user can do what they want based on the payload
CI/CD Examples - Gitlab/Harbor/Jira Integration

Event payload can be passed to build task if needed

- Call harbor API to add/remove label of image, can also lock the test passed image

- Get the commits between two builds
- Invoke Jira API to mark the status and add the build info

git log --pretty=oneline c5eff7ea..3211901e
3211901e9b877c92ab059a6f25180469dcbf1629
5d6cb9bd9a8e8a34cad8a27df036c07f3b4c291
b10192764026df3e3ac57d69019032eeb722ceb
7797990da64929112a90e9f6839bd342864c99b
7797990da64929112a90e9f6839bd342864c99b
96e85fdec50aafafca2eae6a2a1fe4b1ae72d

Merge branch 'dev-branch' into 'dev-branch'

• Get the commits between two builds
• Invoke Jira API to mark the status and add the build info

git commit should have naming convention
• Our DevOps Expectations
• Kubernetes Capabilities and Advantages to Build DevOps Solution
• Architecture and Features
  • CRD and operator design
  • Pipeline/Stage/Task/Task Template/Version Control/UI generation/Volume...
  • Logging, monitoring, autoscaling, high availability
  • Extensibility/Integration
  • CI/CD examples

• Future plan
Our Future Plan

• More task templates to be added, integrate more CI/CD and project management tools
• Optimize UI generation methodology
• Improve development experience, such as CLI, plugin for IDE, dev on Cloud
• Move forward to better DevOps under micro-service architecture
• Consolidate the consistent events and data, and create k8s native ChatOps tool

- Automation
- Integration
- Collaboration
- Quantification

• Contribute to open source community
Thank you!