Chaos Mesh and Predator Integration

Requirements

1. Predator job should have the ability to work with/without chaos mesh .
2. Open source users can choose whether or not to apply chaos mesh experiment as part of the job.
3. Predator platform will allow to create a chaos experiment and execute it as part of the running job.
4. Predator platform will allow to select chaos experiments from created list, to be executed it as part of the running job.
5. When experiment is triggered successfully - experiment resource should be cleaned (to keep uniqueness of resource).

Prerequisites:

1. Chaos mesh learning - Chaos Mesh Overview | Chaos Mesh
2. Chaos mesh integration - https://git.zooz.co/payu-clan-sre/chaos/chaos-mesh
3. Chaos mesh expose https://ingress-internal.eks-mars-apps.zooz.co/chaos-mesh/

API Description

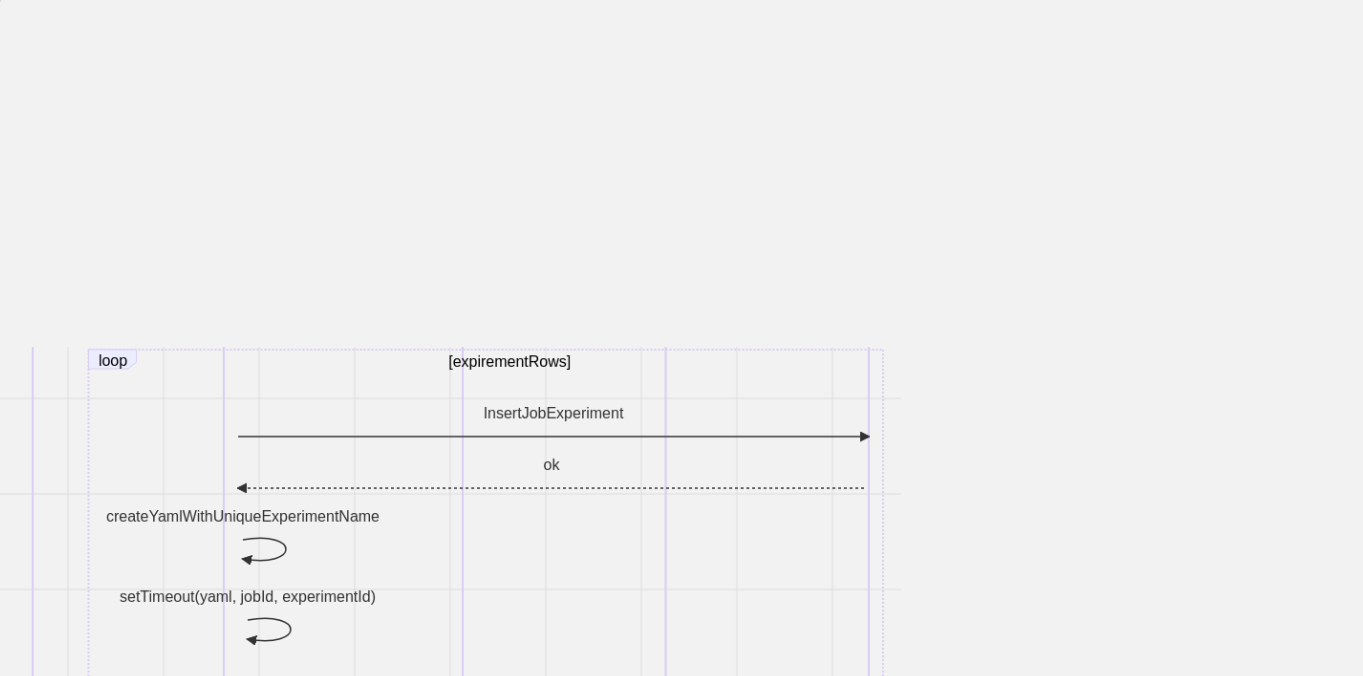
Swagger of CRUD for experimients

Visualize OpenAPI (Swagger) documentation app

Export to PDF of the OpenAPI specification is not supported. See interactive documentation online.

A new table will be added to MySQL on the project, named chaos_expirements that describes the experiments included in a job.oolean

Architecture


loop
[expirementRows]
InsertJobExperiment
ok
createYamlWithUniqueExperimentName
setTimeout(yaml, jobId, experimentId)

Flows

Installation

1. create helm option `serviceAccount.chaosMesh` to allow role binding to chaos mesh resources
   `kubectl get rolebinding fancy-frog-predator -n predator -o yaml`

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  annotations:
    meta.helm.sh/release-name: fancy-frog
    meta.helm.sh/release-namespace: predator
  creationTimestamp: "2021-03-24T07:46:54Z"
  labels:
    app: fancy-frog-predator
    app.kubernetes.io/managed-by: Helm
    chart: predator-1.6.1
    heritage: Helm
    release: fancy-frog
  name: fancy-frog-predator
  namespace: predator
  resourceVersion: "223566370"
  uid: c3c5e393-1c31-4b2d-9f89-2c9d1ea20fdf
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: fancy-frog-predator
subjects:
- kind: ServiceAccount
  name: fancy-frog-predator
  namespace: predator
```

create new role for chaos mesh - `predator-cluster-role.yaml`
and apply (`kubectl apply -f predator-cluster-role.yaml` `)

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: role-cluster-manager-nexnk
rules:
- apiGroups: [""]
  resources: ["pods", "namespaces"]
  verbs: ["get", "watch", "list"]
- apiGroups: ["chaos-mesh.org"]
  resources: ["*"]
  verbs: ["get", "list", "watch", "create", "delete", "patch", "update"]
```

edit rolebinding using `kubectl edit rolebinding fancy-frog-predator -n predator` `

```
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: role-cluster-manager-nexnk
```

chaos experiments crud

Crud will allow us to add the experiments as part of the job executed by predator

1. create chaos mesh to be saved in storage (set as template)
2. get template of experiment to be part of the job

chaos experiments as part of job

If chaos mesh experiment is added to the job, the job will include the chaos mesh as part of the job execution

Visualize OpenAPI (Swagger) documentation app

Export to PDF of the OpenAPI specification is not supported. See interactive documentation online.

A new table will be added to MySQL on the project, named job_expirements that describes the experiments included in a job.oolean

implantation steps

On the existing `createJob` function, the flow will be as following:

1. existing create job flow - including triggering the job.
2. get all experiments from db, by id.
3. For each experiment of the job:
   a. insert row to `job_expirements` table.
   b. create new yaml text with unique name.
   c. Call setTimeout with remaining time till planned execution, with function `applyChaosExperiment(chaosYaml: string, jobId: string, experimentId: string)`

The function `applyChaosExperiment ` shall:

1. Create FILE with the given text (locally on the pod) called `${experiment-name}.yaml`
2. execute `kubectl apply -f ${experiment-name}.yaml`

3. update `job_expirements` table row with `is_triggered = true`
4. delete the file.

Feature Control

Feature will be merged into a feature branch GitHub - Zooz/predator at fb-chaos-mesh-support
and will relate to issue/feature request - chaos mesh integration with predator · Issue #634 · Zooz/predator

Resilience

- need to make sure when chaos mesh experiment is executed - to create a unique name - as it is a Kubernetes resource
- Perform deletion of chaos mesh resource after execution

Security & Data Privacy

Security Considerations

see Chaos Mesh and Predator Integration | Installation
When chaos mesh is used it is important to create a clusterRole rule and bind to service account. this should be a configuration in helm and done as a prerequisite

Data Consumption

chaos experiments

column	type	description
id	text (uud)	primary key
name	text (uud)	
kubeObject	text (uud)	
created_at	timestamp	
updated_at	timestamp	
context_id	boolean	

job_chaos experiments

column	type	description
id	text (uud)	primary key
job_id	text (uud)	foreign key
experiment_id	text (uud)	foreign key
start_time	timestamp	
end_time	timestamp	
is_triggered	boolean	

Infrastructure Considerations

Implementation Steps

Phase	domain	Items	Business impact
1	chaos-experiments crud	chaos experiment data model in sequalize	2MD

	chaos-experiments crud	chaos experiment first template - insertion (pod fault)	1MD
	chaos-experiments crud	GET chaos experiment by id	0.5MD
	chaos-experiments crud	GET chaos experiments	0.5MD
	chaos-experiments crud	PATCH set chaos experiment as template	0.5MD
	chaos-experiments crud	POST chaos experiment (no yaml validation)	1MD
	jobs crud	add chaos experiment id to jobs crud	3MD
	jobs execution	execute job with chaos mesh	6MD
	chaos mesh resources cleanup		2MD
	UI -create chaos mesh		
	UI -create chaos mesh		
	UI -create chaos mesh		

Open Issues