Cloud Native Chaos Engineering

Chaos Engineering for Kubernetes

@uma_mukkara
23 January 2020

JANUARY 23 2020

CONF42

CHAOS ENGINEERING

LONDON, UK

About me

Uma Mukkara
Co-Founder & COO
@uma_mukkara

MayaData

Open source projects

OpenEBS

Litmus

KUBE MOVE

Chaos Eng is ...

- A need
- A culture
- A practice
- I got all that.. But how do I start?
    - Yes, I am on Kubernetes

Agenda

- What is Cloud-Native Chaos Engineering ?
- Principles of Cloud-Native Chaos Engineering
- Introduction to Litmus
- Chaos-Hub
- Examples
- What can you do?
- Q&A

Cloud Native Chaos Engineering

(Introduction)

Chaos engineering for cloud native enviro

Chaos engineering done cloud-native way

Chaos engineering done Kubernetes native way

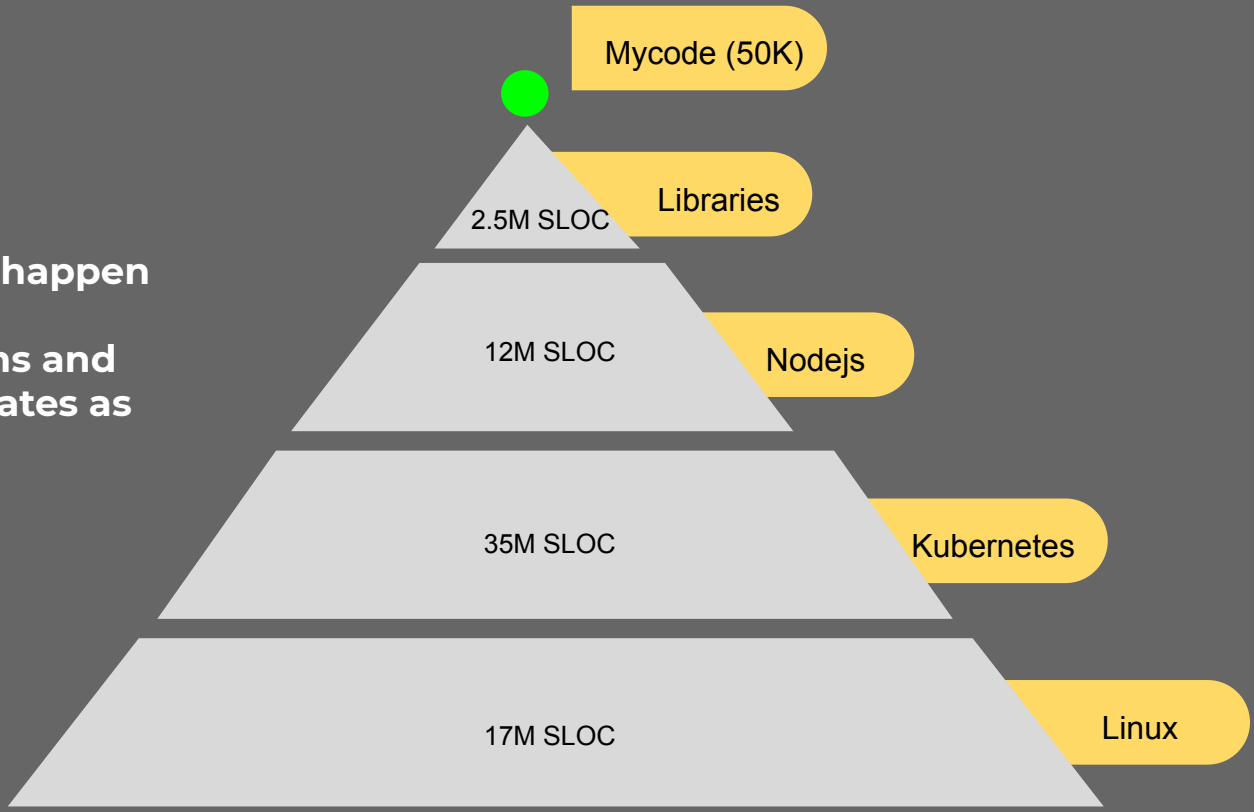
Cloud-Native environments

- *Credits: Takens from GitLab commit conference slides; Author is Dan Kohn, CNCF*

Cloud-Native environment

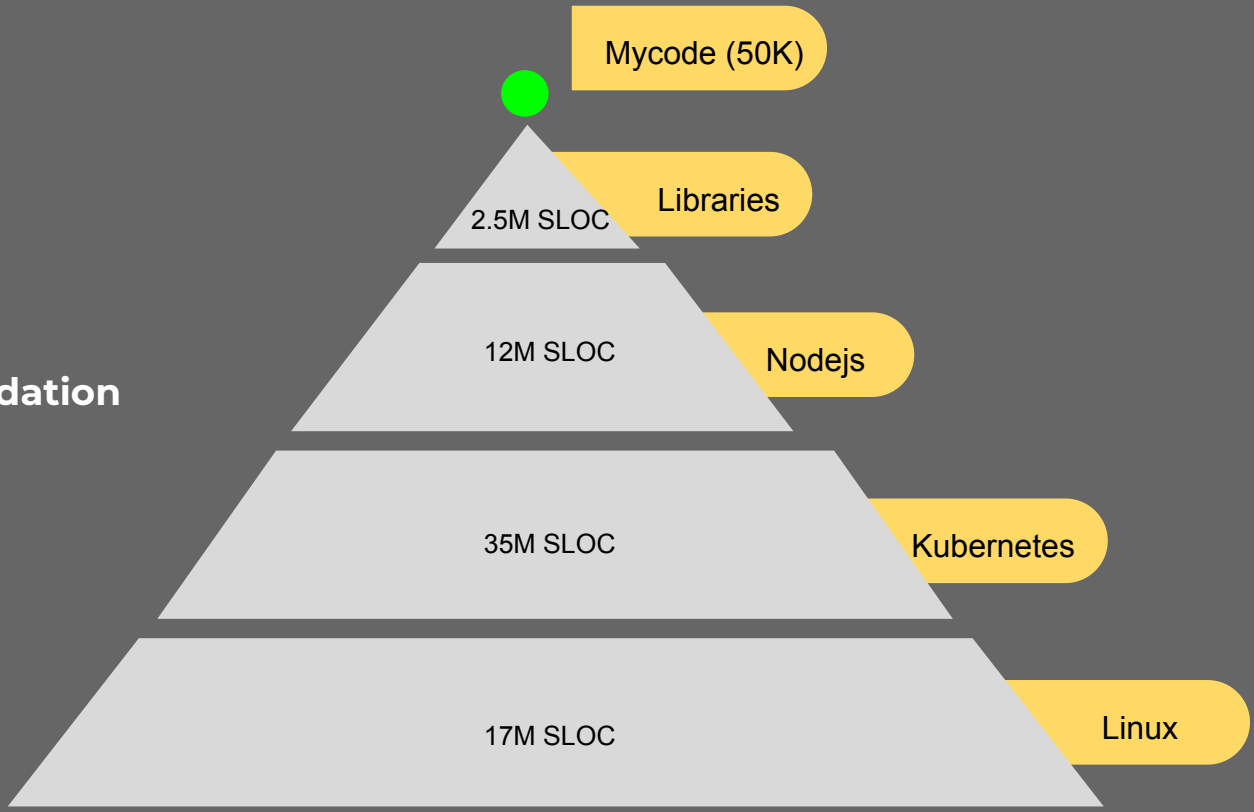
Your code vs entire stack

- **Kubernetes upgrades happen frequently**
- **Associated applications and libraries will have updates as often as your code**



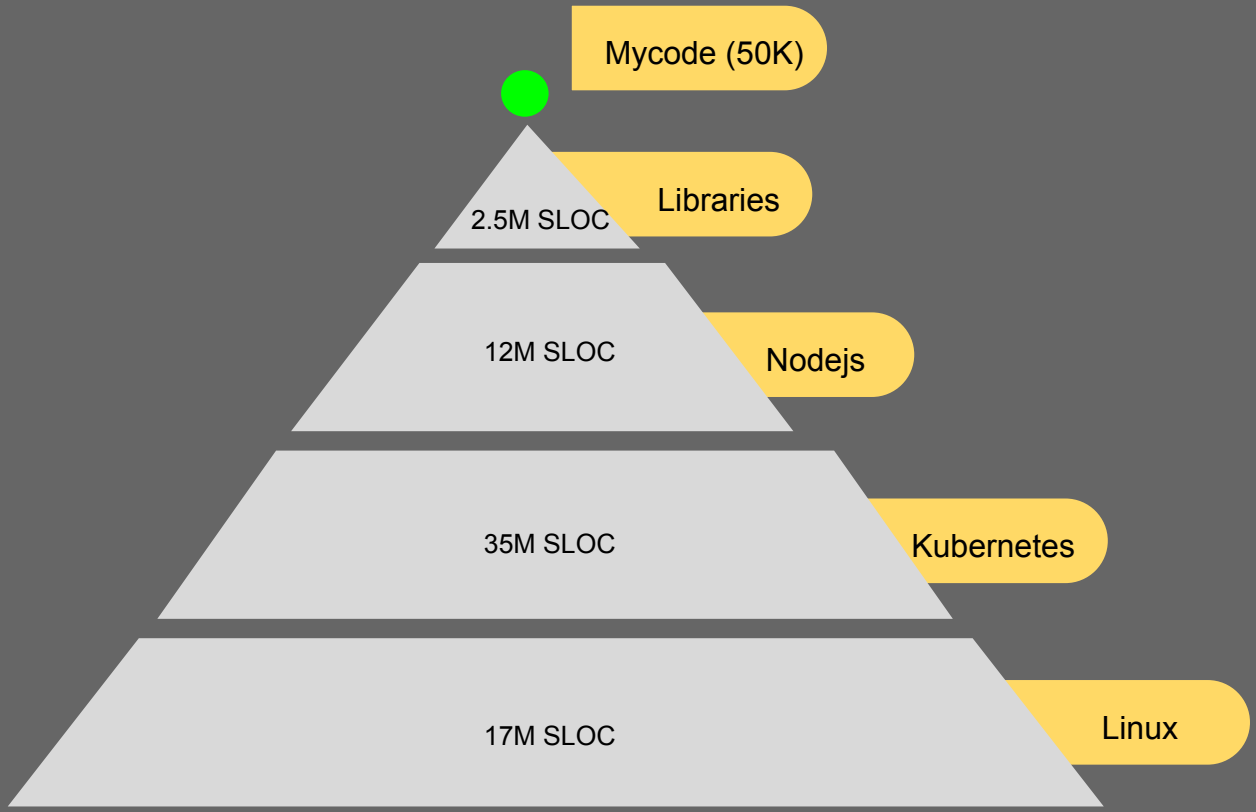
Cloud-Native environment

Your environment
- **Is very Dynamic**
- **Needs continuous validation**



Cloud-Native environment

You need
- **Chaos Engineering**



Cloud-Native environment

You need
- **Cloud-Native Chaos Engineering**

The other big differences in Cloud-Native environment are:
- **YAML manifests for intent (kubectl apply)**
- **GitOps**

Chaos Engineering = **Cloud Native Chaos Engineering**

Cloud-Native Chaos Engineering

Cloud-native Application

Cloud Native APIs

For Development

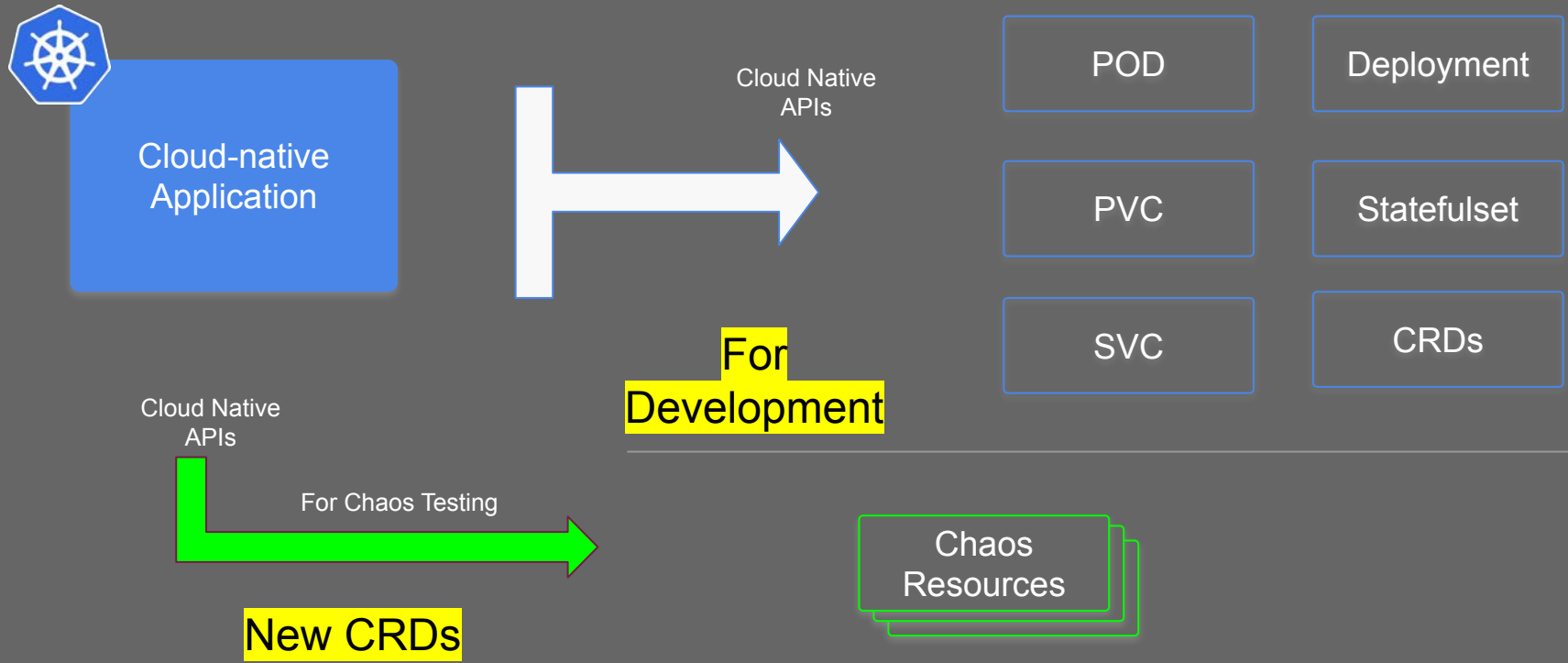
- POD
- Deployment
- PVC
- Statefulset
- SVC
- CRDs

Cloud Native APIs

For Chaos Testing

?

Cloud-Native Chaos Engineering



Cloud-Native Chaos Engineering

Chaos CRDs

Chaos Operator

Chaos Metrics

Cloud Native Chaos Engineering

Principles

Principles of Cloud-Native Chaos Engineering

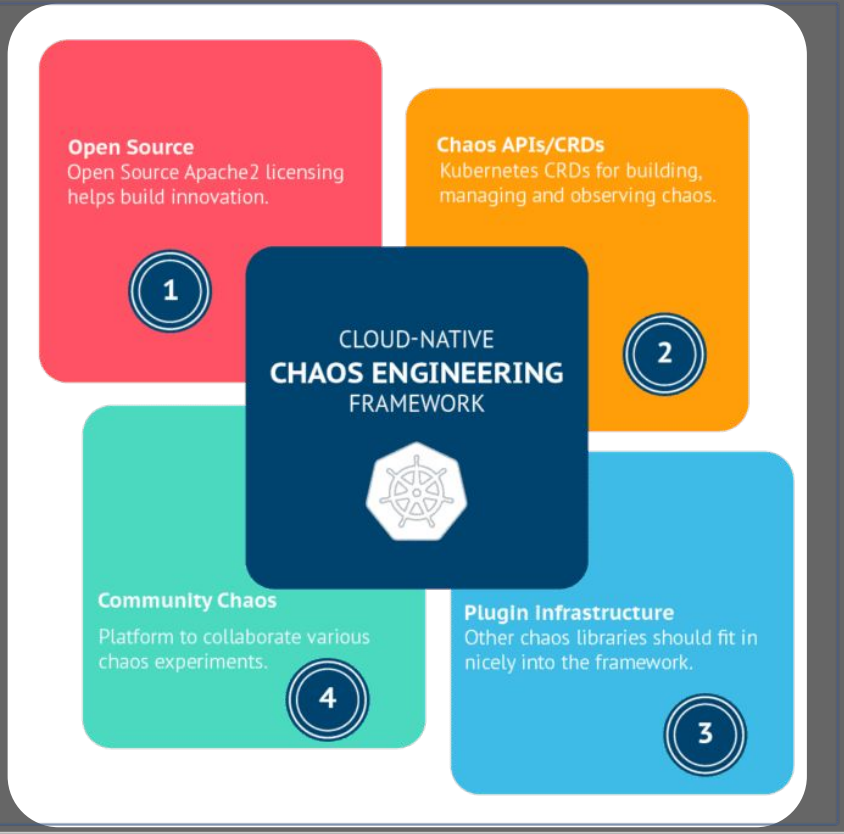
Open Source

Chaos API/CRDs

Pluggable chaos

Community driven

Principles of Cloud-Native Chaos Engineering



CNCF blog: http://bit.ly/cncf-chaos

Cloud Native Chaos Engineering

Litmus project

Litmus project - Introduction

- Leading open source project for Chaos Engineering on Kubernetes
- Apache2 License
- https://github.com/litmuschaos
- 50+ contributors
- 600+ stars
- CNCF Landscape - https://landscape.cncf.io/selected=litmus
- Chaos Hub https://hub.litmuschaos.io/
- CNCF Blog https://bit.ly/cncf-chaos

Litmus - Cloud Native

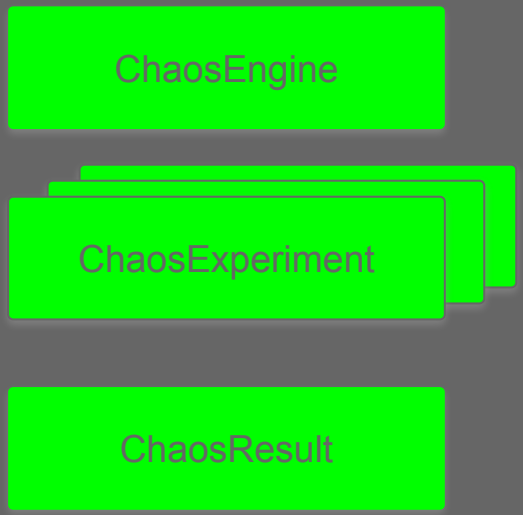
Open Source

Chaos API/CRDs

Pluggable chaos

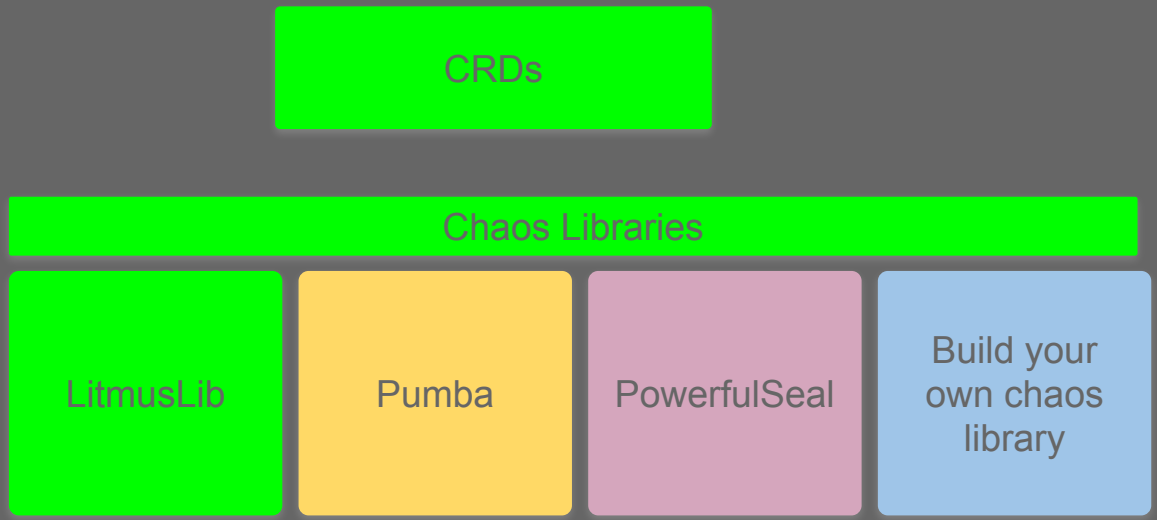
Community driven

Litmus - Cloud Native



Litmus - Cloud Native

Pluggable chaos



Example of plugging new chaos library

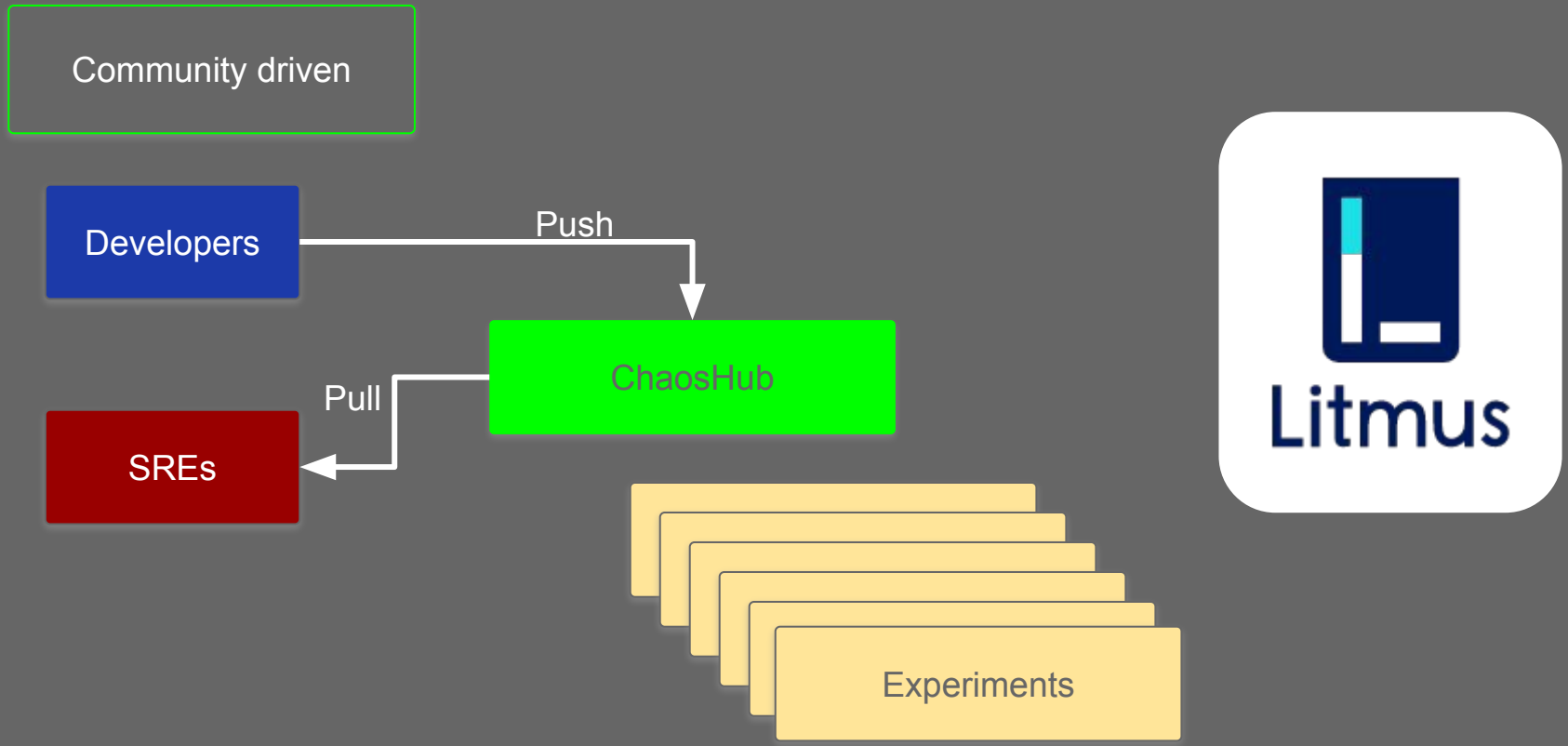
Pluggable chaos

PowerfulSeal Chaos logic → Build a Docker image → Create PowerfulSeal Deploy

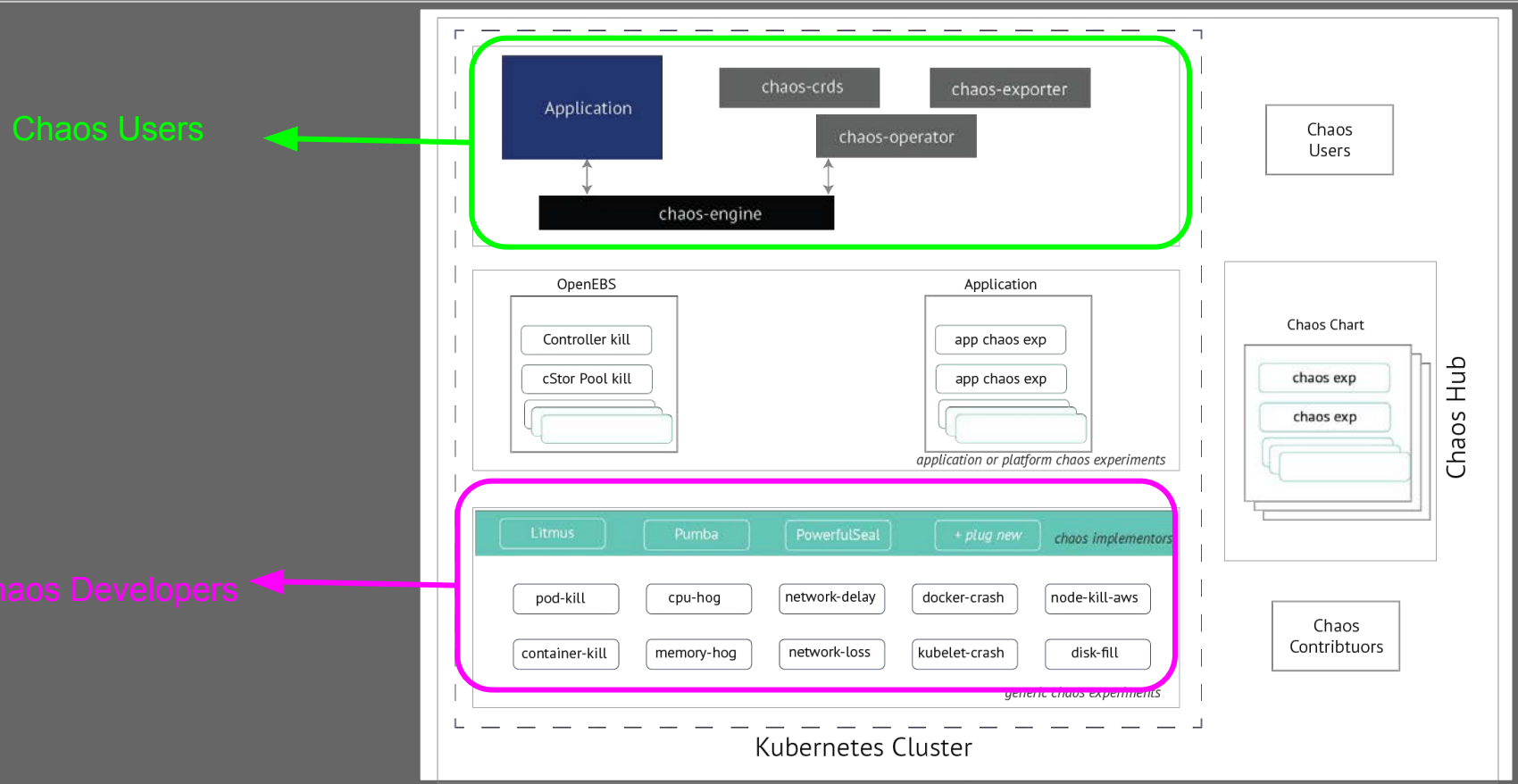
Create a new experiment → Create a new CR → Set powerfulseal as the ChaosLib

Litmus chaos runner automatically takes care of calling powerfulseal kill experiment, observes the results and updates the ChaosResult CR.

Litmus - Cloud Native



Cloud Native Architecture



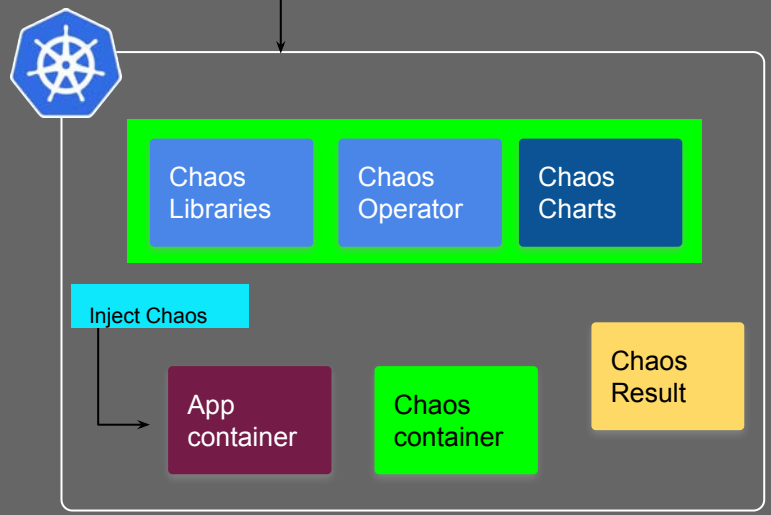
Cloud Native Chaos Engineering

Getting Started

How Litmus works

Install Litmus

```
kubectl apply -f <litmus.yaml>
```



Install Charts

hub.litmuschaos.io

Cloud-Native Chaos Engineering - Example

Cloud Native Developer

Create POD

```
apiVersion: v1
kind: Pod
metadata:
 name: percona-pod
 labels:
  app: percona
spec:
 containers:
 - name: percona
   image: percona:2.4
```

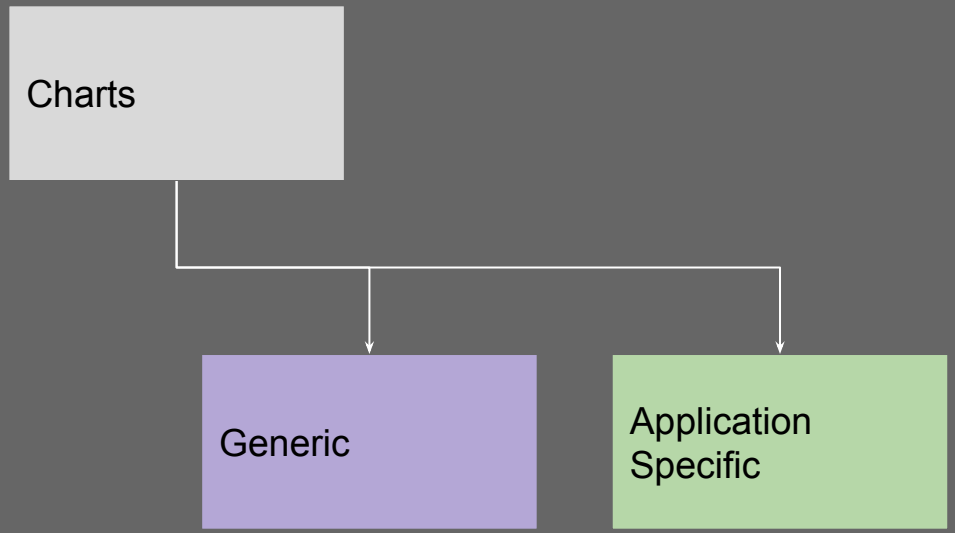
Create PV

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  Name: demo-vol1-claim
spec:
  storageClassName:
openebs-jiva-default
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

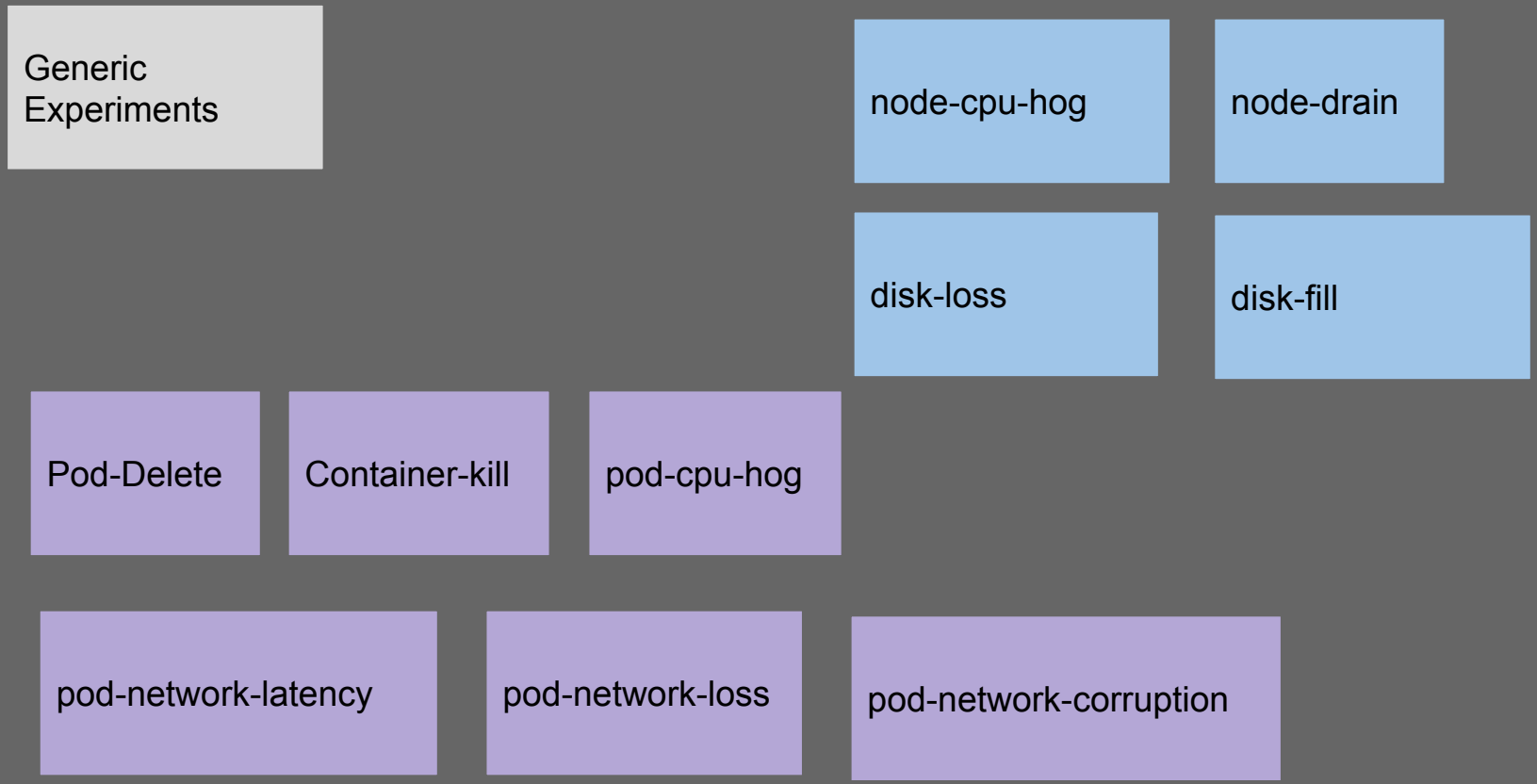
Inject Chaos

```
apiVersion: litmuschaos.io/v1alpha1
kind: ChaosEngine
metadata:
  name: engine-percona
spec:
  appinfo:
    appns: default
    applabel: "app=percona"
  experiments:
    - name: replica-kill
      spec:
        components:
    - name: target-kill
      spec:
        components:
```

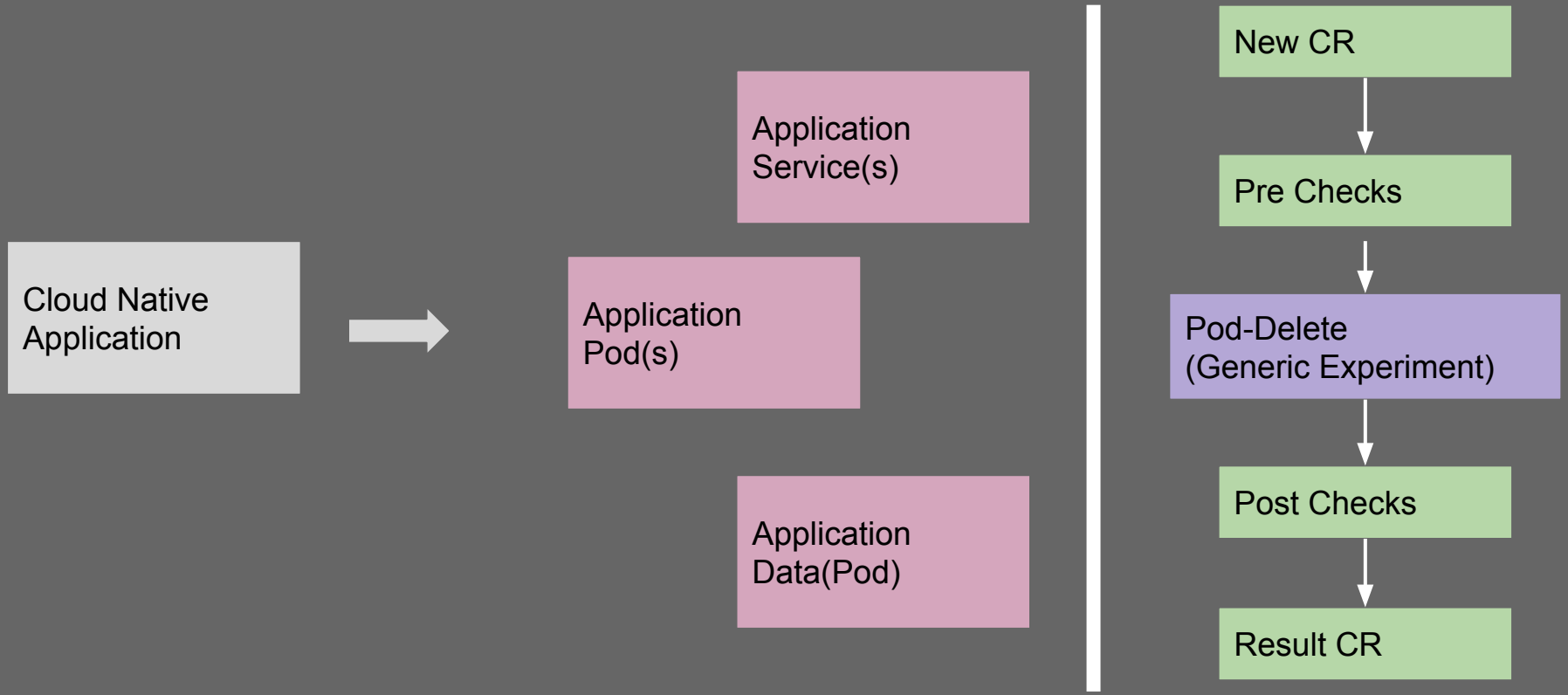
ChaosHub



ChaosHub - Generic experiments

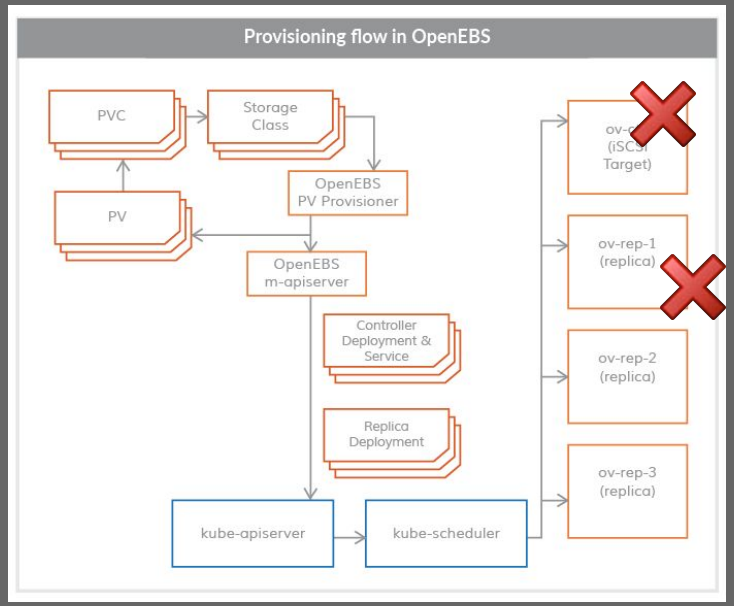


ChaosHub - Application specific experiments



ChaosHub - Application specific experiments

OpenEBS Experiments



openebs-pool-pod-failure

openebs-pool-container-failure

openebs-target-pod-failure

openebs-target-container-failure

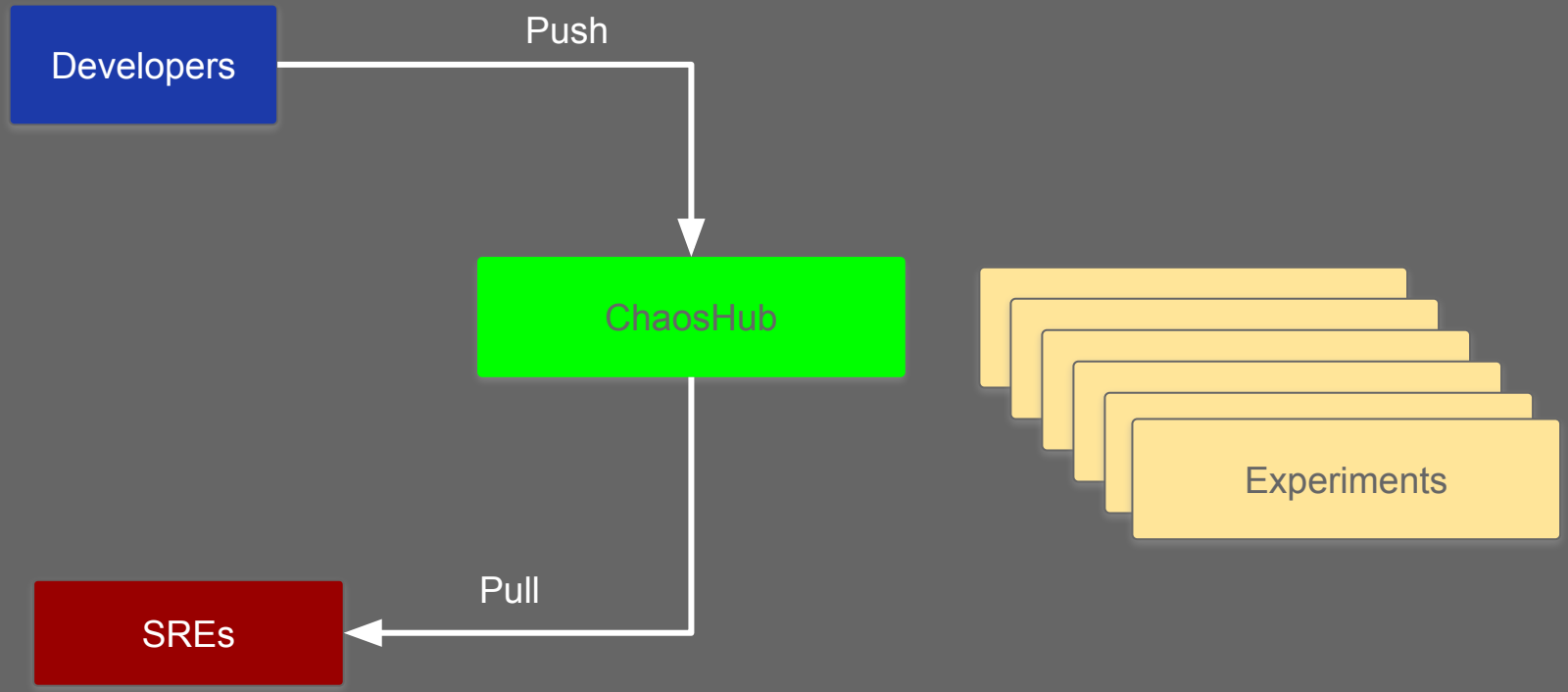
openebs-target-network-delay

openebs-target-network-loss

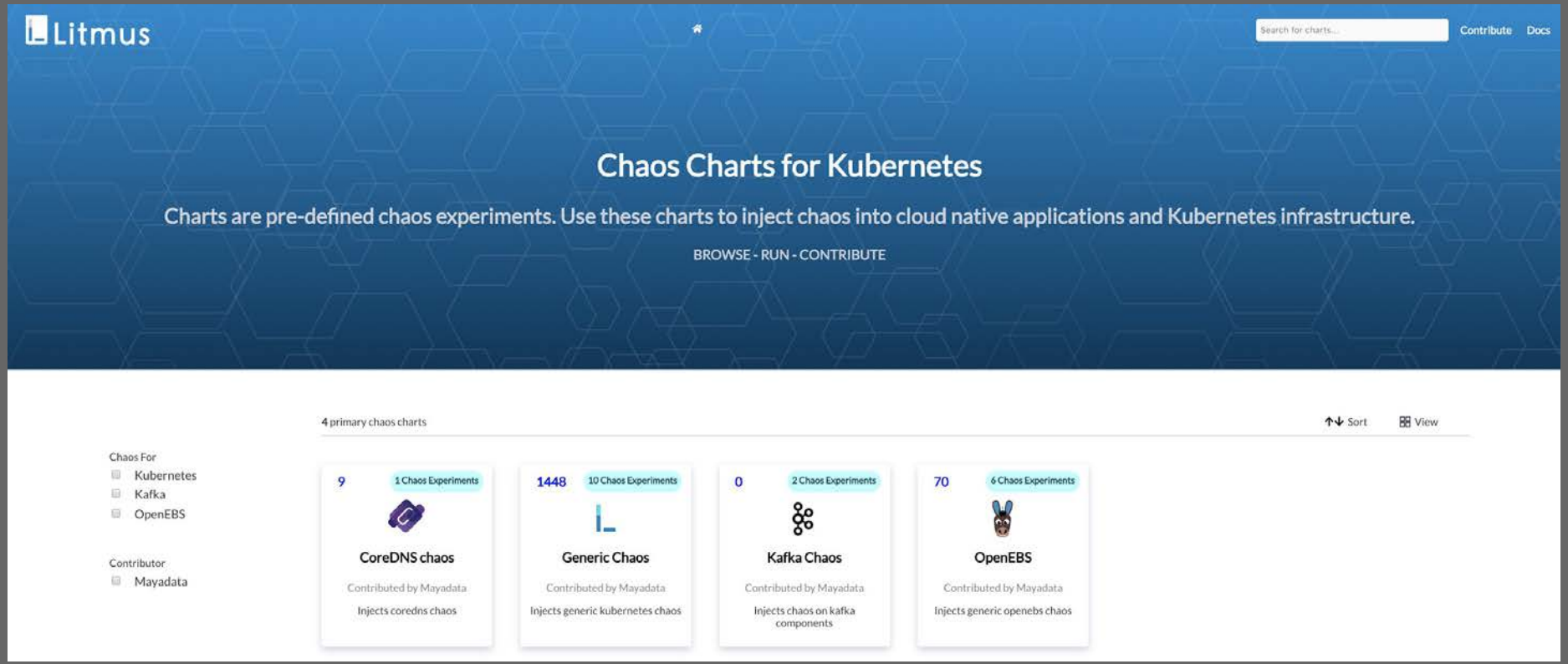
How can you contribute

- Join #litmus channel on Kubernetes Slack community
- Use Litmus and create new issues
- Push new experiments to ChaosHub

If you are practicing Chaos Engineering ..



ChaosHub


Litmus
Search for charts...
Contribute
Docs
Chaos Charts for Kubernetes
Charts are pre-defined chaos experiments. Use these charts to inject chaos into cloud native applications and Kubernetes infrastructure.
BROWSE - RUN - CONTRIBUTE
4 primary chaos charts
Sort
View
Chaos For
Kubernetes
Kafka
OpenEBS
Contributor
Mayadata
9
1 Chaos Experiments
CoreDNS chaos
Contributed by Mayadata
Injects coredns chaos
1448
10 Chaos Experiments
Generic Chaos
Contributed by Mayadata
Injects generic kubernetes chaos
0
2 Chaos Experiments
Kafka Chaos
Contributed by Mayadata
Injects chaos on kafka components
70
6 Chaos Experiments
OpenEBS
Contributed by Mayadata
Injects generic openebs chaos

Cloud Native Chaos Engineering

Thank you

Q&A