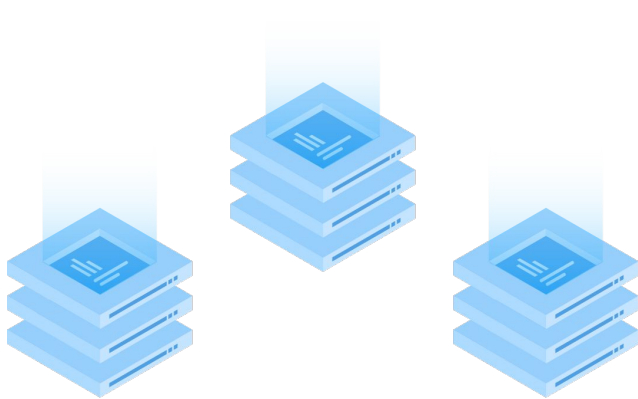


Chaos Engineering at PingCAP

Presented by Max Liu @ngaut1



Hello TiDB



About TiDB

22K GitHub stars | 1000+ production users

- Features
 - Scale out MySQL, no more manual sharding
 - Hybrid transactional / analytical processing (HTAP)
 - Distributed transactions with strong consistency
 - High availability and auto-failover
 - Cloud neutral alternative for Amazon Aurora and Google Spanner

PingCAP: true open-source believer

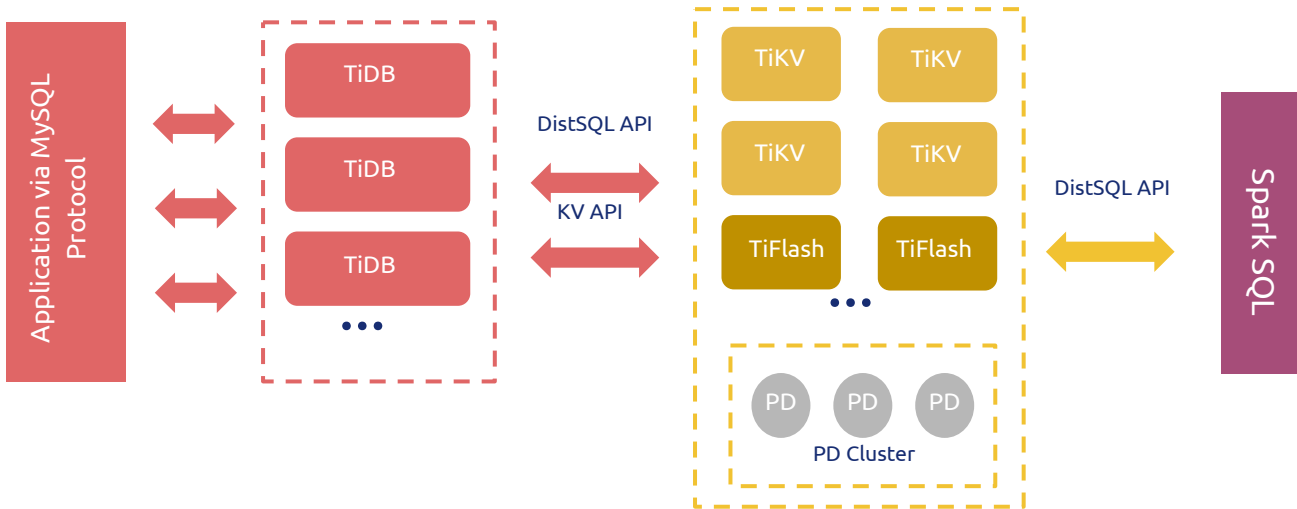
CNCF contributions statistics, ranking by companies. PingCAP is the only startup

<https://all.devstats.cncf.io/d/5/companies-table?orgId=1>

- #1 Google
- #2 Red Hat
- #3 VMWare
- #4 Independent
- #5 Microsoft
- #6 PingCAP**
- #7 Huawei
- #8 IBM

Rank ▲	Company	Number
	All	3376415
1	Google	879850
2	Red Hat	491437
3	VMware	118552
4	Independent	87235
5	Microsoft	79446
6	PingCAP	51493
7	Huawei	49316
8	IBM	48010
9	Lyft	40700

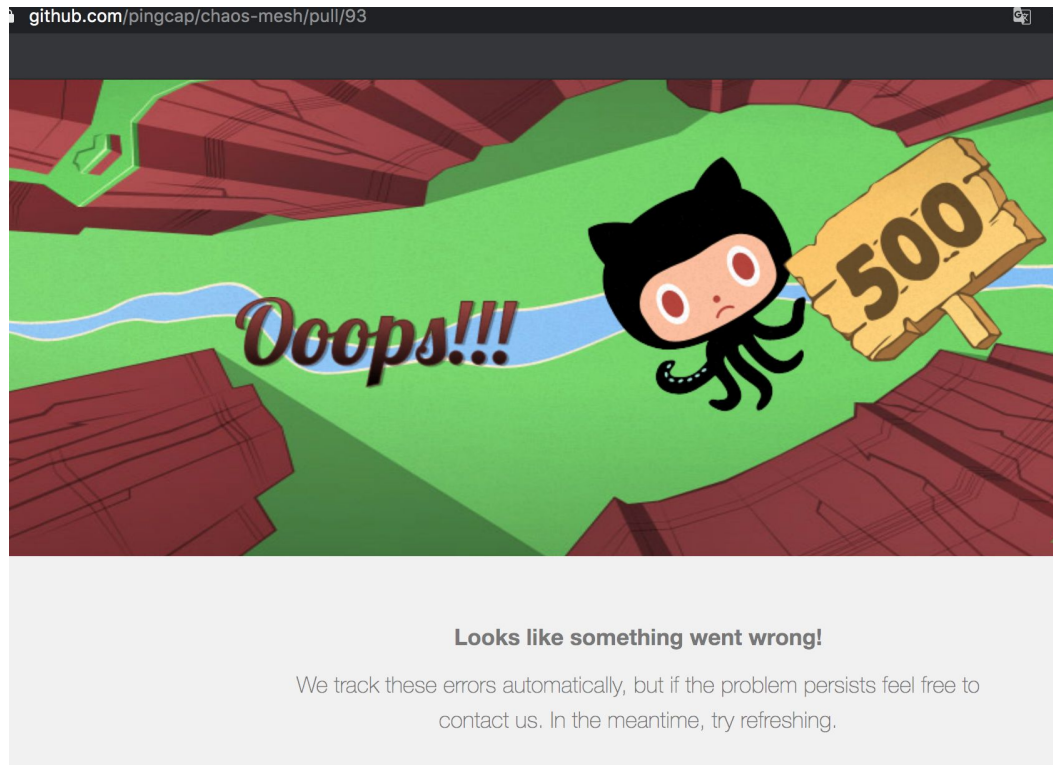
Architecture of TiDB



Quality Assurance



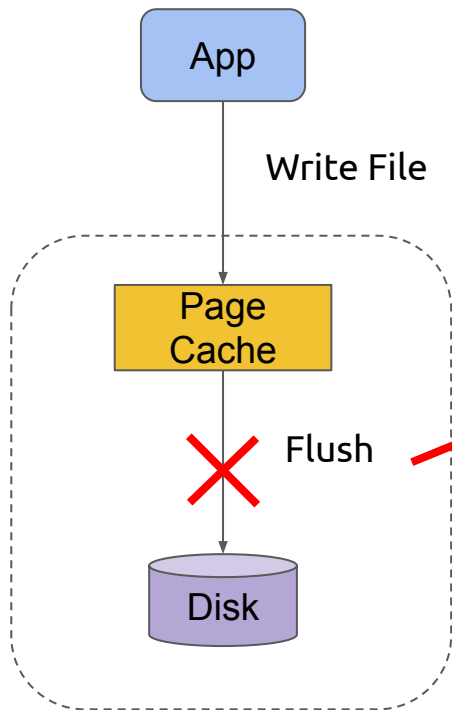
Error happens, any type, anytime, anywhere, any device.



Black Swan



Linux Kernel bug



```
[17988717.953807] 0 pages in swap cache
[17988717.953808] Swap cache stats: add 0, delete 0, find 0/0
[17988717.953809] Free swap = 0kB
[17988717.953810] Total swap = 0kB
[17988717.953811] SLUB: Unable to allocate memory on node -1 (gfp=0x20)
[17988717.953813] cache: kmalloc-8192, object size: 8192, buffer size: 8192,
[17988717.953815] node 0: slabs: 78, objs: 312, free: 21
[17988717.953816] node 1: slabs: 37, objs: 148, free: 0
```

Linux **kernel** bugs happen in the real world!!!

<https://pingcap.com/blog/try-to-fix-two-linux-kernel-bugs-while-testing-tidb-operator-in-k8s/>

American fuzzy lop

Time to first bug

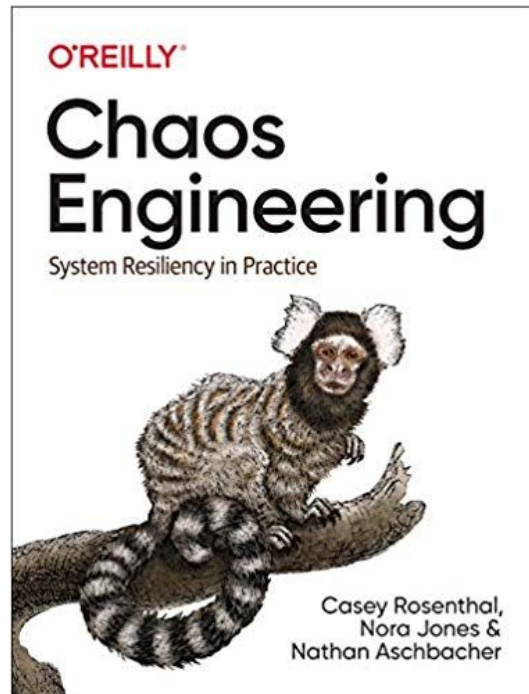
Filesystem	Time	Type
Btrfs	5s	BUG()
Ext4	2h	BUG()
F2fs	10s	BUG()
Gfs2	8m	Double free
Hfs	30s	Page Fault
Hfsplus	25s	Page Fault
Nilfs2	1m	Page Fault
Ntfs	4m	Soft lockup
Ocfs2	15s	BUG()
Reiserfs	25s	BUG()
Xfs	1h45m	Soft lockup

Chaos Engineering



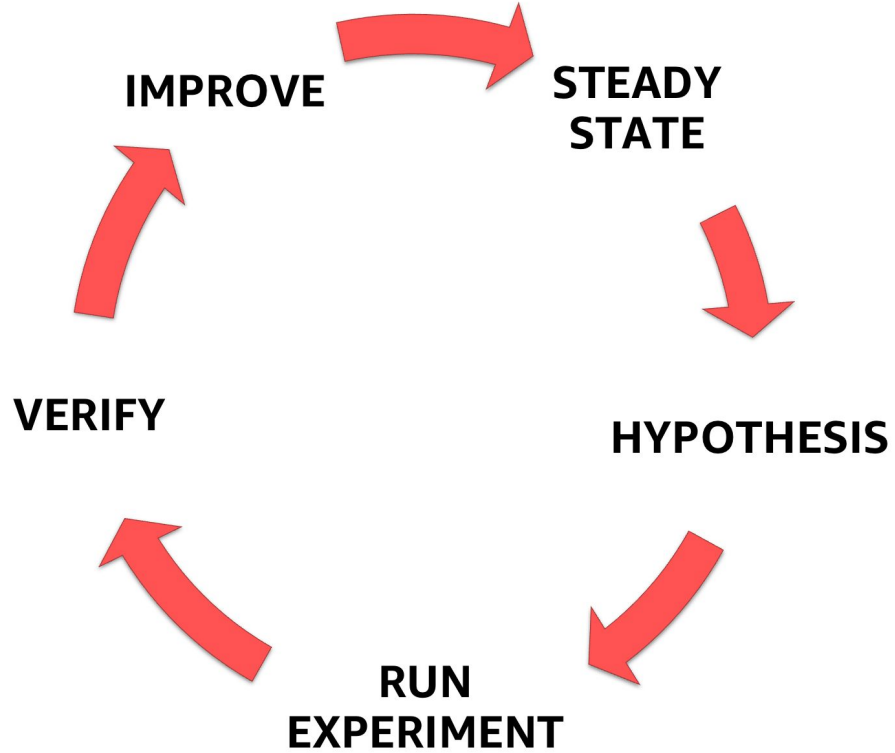
Chaos engineering in PingCAP

- [From Chaos to Order -- Tools and Techniques for Testing TiDB, A Distributed NewSQL Database](#)
- [Safety First! Common Safety Pitfalls in Distributed Databases Found by Jepsen Tests](#)
- [Chaos Engineering - System Resiliency in Practice](#)

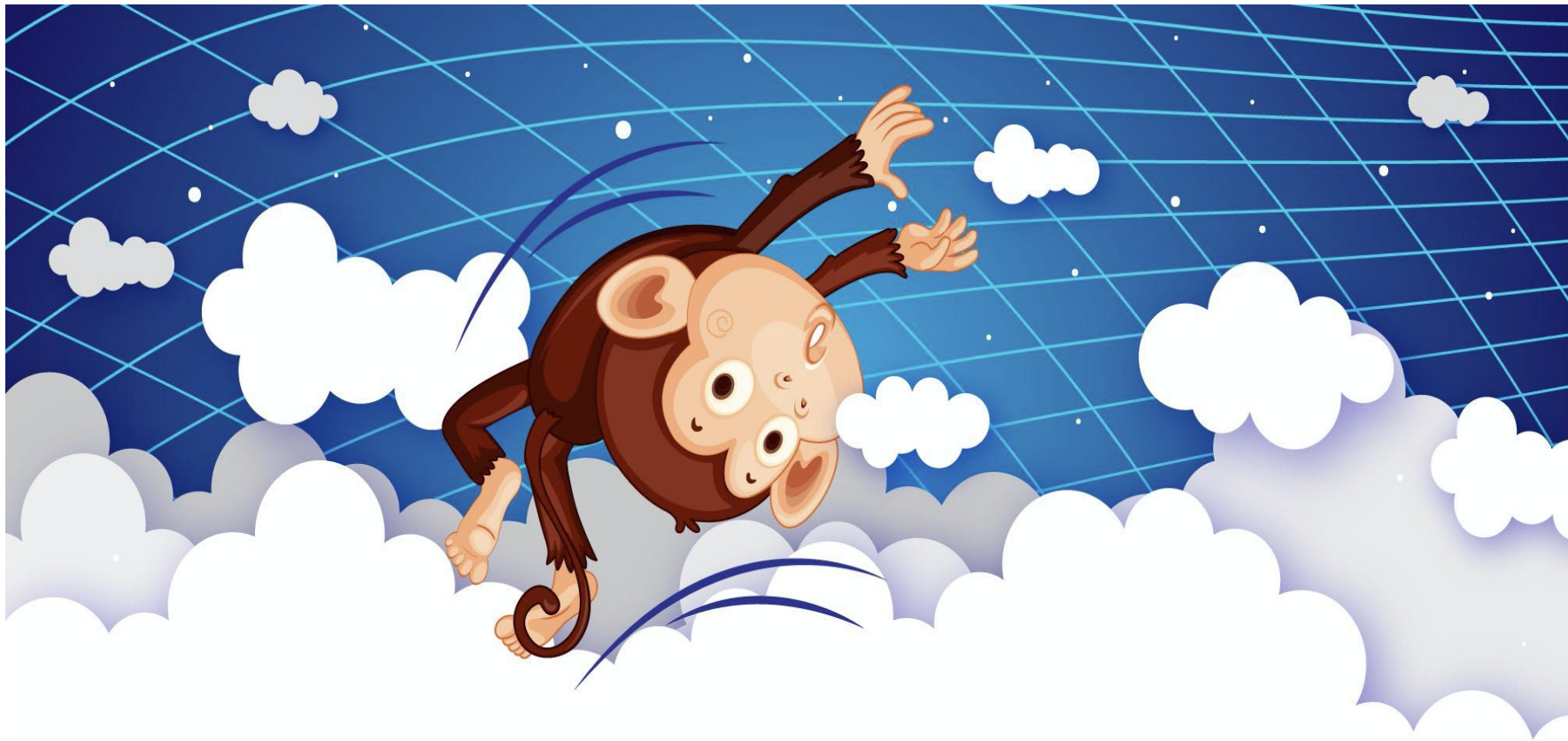


A Brief History of Chaos Engineering

- **2019**
 - [Chaos Mesh](#) is open sourced by PingCAP
- **2018**
 - “[Principle of Chaos Engineering](#)” became online
- **2014**
 - Netflix decided they would create a new role: the Chaos Engineer
- **2010**
 - Netflix Eng Tools team created Chaos Monkey



The Phases of Chaos Engineering



Chaos Mesh

Trending


See what the GitHub community is most excited about today.

Repositories Developers

Spoken Language: Any Language: **Go** Date range: Today

pingcap / chaos-mesh

A Chaos Engineering Platform for Kubernetes


Go ★ 564 🍴 26 Built by 

★ 286 stars today

★ Unstar

quii / learn-go-with-tests

Learn Go with test-driven development


Go ★ 9,578 🍴 1,051 Built by 

★ 135 stars today

★ Star

TesraSupernet / Tesra

tesra


Go ★ 129 🍴 16 Built by 

★ 58 stars today

★ Star

open-policy-agent / opa

An open source, general-purpose policy engine.


Go ★ 2,905 🍴 306 Built by 

★ 11 stars today

★ Star

OWASP / Amass

In-depth Attack Surface Mapping and Asset Discovery


Go ★ 2,466 🍴 410 Built by 

★ 8 stars today

★ Star

rancher / k3s

Lightweight Kubernetes. 5 less than k8s.

Go ★ 11,031 🍴 768 Built by 

★ 15 stars today

★ Star

- JavaScript (github.com)
453 points by atum47 16 hours ago | flag | hide | 76 comments
6. ▲ Measuring mutexes, spinlocks and how bad the Linux scheduler is (probablydance.com)
204 points by bazzargh 15 hours ago | flag | hide | 53 comments
7. ▲ VidGear: Complete Video Processing library for all Python lovers (github.com)
27 points by abhiTronix 6 hours ago | flag | hide | discuss
8. ▲ Doing a database join with CSV files (johndcook.com)
291 points by chmaynard 16 hours ago | flag | hide | 85 comments
9. ▲ Calculate the day of the week for any date in your head (rudyc.ca)
109 points by emrehan 10 hours ago | flag | hide | 10 comments
10. ★ A Chaos Engineering Platform for Kubernetes (github.com)
19 points by ngaut 3 hours ago | hide | discuss
11. ▲ A City Is Not a Tree (1965) (patternlanguage.com)
48 points by dedalus 7 hours ago | flag | hide | 7 comments
12. ▲ Libcamera – A complex camera support library for Linux, Android, and ChromeOS (libcamera.org)
145 points by executesorder66 15 hours ago | flag | hide | 43 comments
13. ▲ Crafting Interpreters: Classes and Instances (craftinginterpreters.com)
193 points by azhenley 16 hours ago | flag | hide | 10 comments
14. ▲ 'Man on the Moon' moment – the year's big breakthroughs in medicine (bbc.com)
64 points by sdumi 10 hours ago | flag | hide | 15 comments

	chaos-mesh	chaosmonkey	chaosblade	chaoskube	litmus
Platform supported	K8s	VMs/ Container	JVM/ Container/ K8s	K8s	K8s
CPU burn	✗	✗	✓	✗	✗
Mem burn	✗	✗	✓	✗	✗
container kill	✗	✓	✓	✗	✓
pod failure	✓	✗	✗	✗	✗
pod kill	✓	✗	✓	✓	✓
network partition	✓	✗	✗	✗	✗
network duplication	✓	✗	✗	✗	✗
network corrupt	✓	✗	✓	✗	✗
network loss	✓	✗	✓	✗	✓
network delay	✓	✗	✓	✗	✓
I/O delay	✓	✗	✗	✗	✗
I/O errno	✓	✗	✗	✗	✗

Talk is cheap,
Show me the code!
But first, show me the demo!!!

Bugs found by chaos-mesh

- TiDB QPS recovered too slowly after long-term failure
 - <https://github.com/pingcap/tidb/pull/11391>
- tidb-binlog table not sync
 - <https://github.com/pingcap/tidb-binlog/issues/851>
- tidb-binlog time sync error
 - <https://github.com/pingcap/tidb-binlog/issues/839>
- CDC table not sync
 - <https://github.com/pingcap/ticdc/issues/172>
- CDC log issue
 - <https://github.com/pingcap/ticdc/issues/175>
- More...

Demo: pod kill

```
[root@172.16.4.4 chaos-demo]# kubectl -n chaos-demo-1 get pods
NAME                                READY   STATUS    RESTARTS   AGE
chaos-demo-1-tidb-discovery-6597d55bd8-fsdc5   1/1     Running   0          14h
chaos-demo-1-tidb-monitor-694bb4dcd9-2bpql    2/2     Running   0          14h
chaos-demo-1-tidb-pd-0                        1/1     Running   0          14h
chaos-demo-1-tidb-pd-1                        1/1     Running   0          14h
chaos-demo-1-tidb-pd-2                        1/1     Running   0          12h
chaos-demo-1-tidb-tidb-0                      2/2     Running   0          14h
chaos-demo-1-tidb-tidb-1                      2/2     Running   0          14h
chaos-demo-1-tidb-tikv-0                      1/1     Running   0          17m
chaos-demo-1-tidb-tikv-1                      1/1     Running   0          21m
chaos-demo-1-tidb-tikv-2                      1/1     Running   0          16m
[root@172.16.4.4 chaos-demo]#
```

```
~/bench/tidb-bench/sysbench(master*) » ./run.sh point_select run 256
sysbench 1.0.11 (using system LuaJIT 2.1.0-beta3)
```

Running the test with following options:

Number of threads: 256

Report intermediate results every 3 second(s)

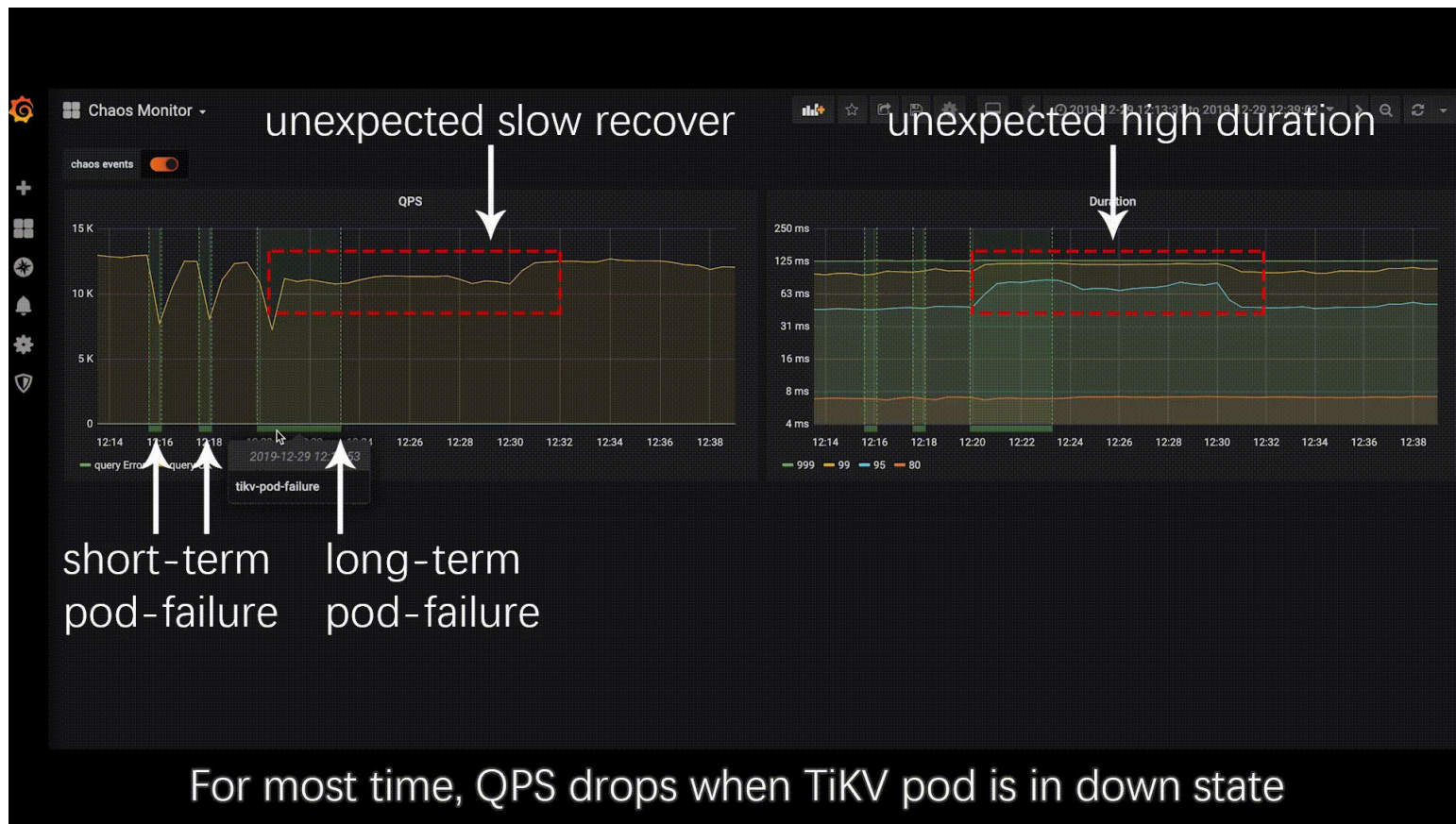
Initializing random number generator from current time

Initializing worker threads...

Threads started!

```
[]
```

Demo: pod failure



Usage

Step1. Checkout the repo

```
$ git clone https://github.com/pingcap/chaos-mesh.git  
$ cd chaos-mesh/
```

Step2. Create CRD (Custom Resource Definitions) objects

```
$ kubectl apply -f manifests/  
$ kubectl get crd podchaos.pingcap.com
```

Step3. Install Chaos Mesh

```
$ helm install helm/chaos-mesh --name=chaos-mesh --namespace=chaos-testing  
$ kubectl get pods --namespace chaos-testing -l app.kubernetes.io/instance=chaos-mesh
```

Example: Random killing Pods

Step1. edit pod-kill-example.yaml

```
kind: PodChaos
spec:
  action: pod-kill
  mode: one
  selector:
    namespaces:
      - tidb-cluster-demo
    labelSelectors:
      "app.kubernetes.io/component": "tikv"
  scheduler:
    cron: "@every 2m"
```

action: pod-kill / pod-failure

mode: one / all / fixed / fixed-percent /
random-max-percent

selector: blast radius

scheduler: scheduling rules

Example: Random killing Pods

Step2. Create a chaos experiment

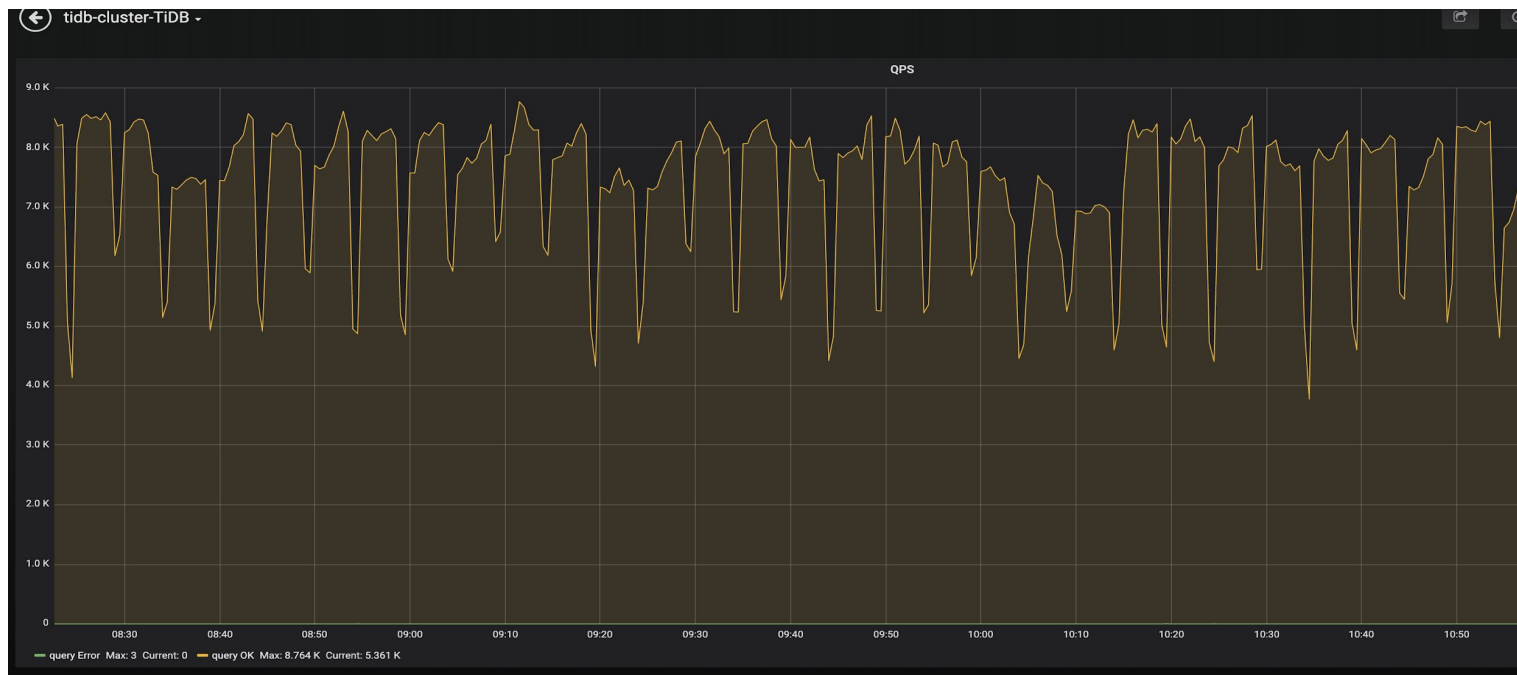
```
$ kubectl apply -f pod-kill-example.yaml  
$ kubectl get podchaos --namespace=chaos-testing
```

Step3. Terminate a chaos experiment

```
$ kubectl delete -f pod-kill-example.yaml
```

Effects: pod-kill

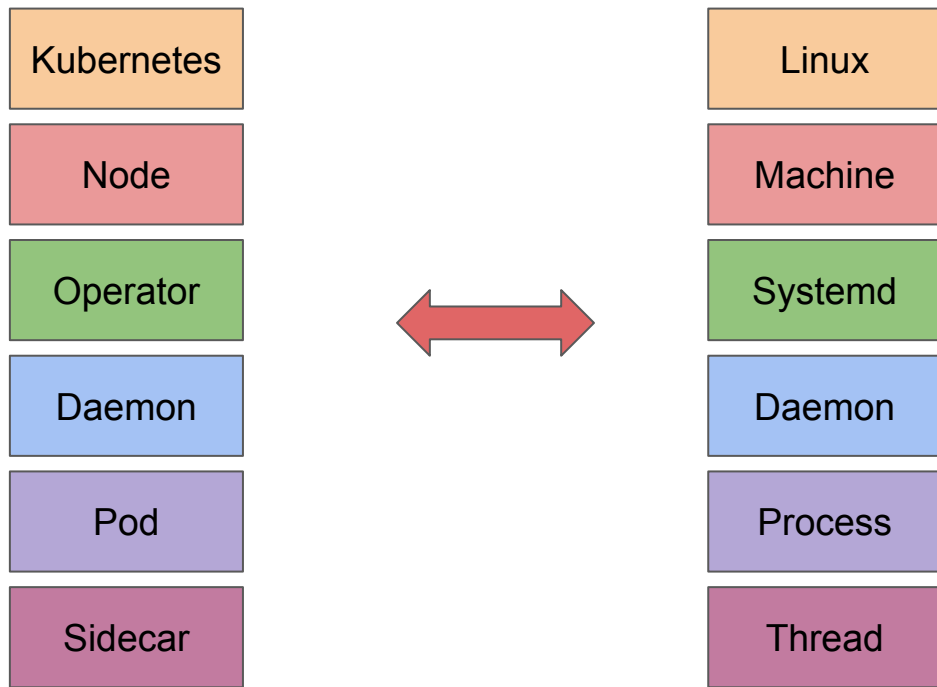
Kill a random pod every 5 minutes (sysbench - Point Select)



Here comes the hardcore detail

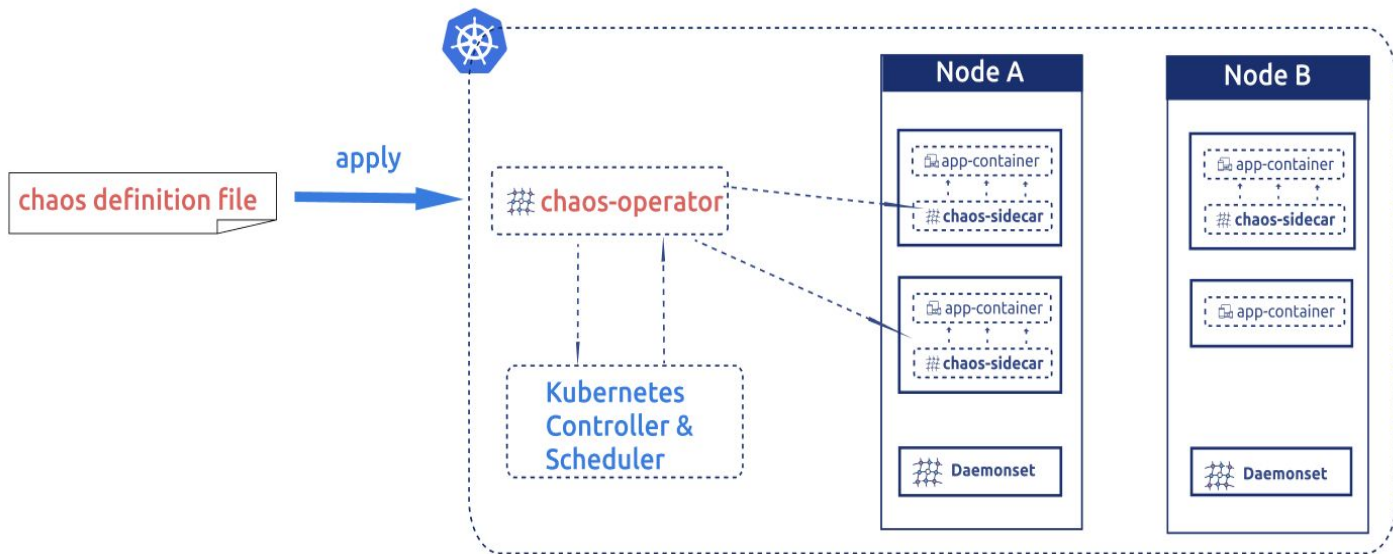


If you are a K8s newbie

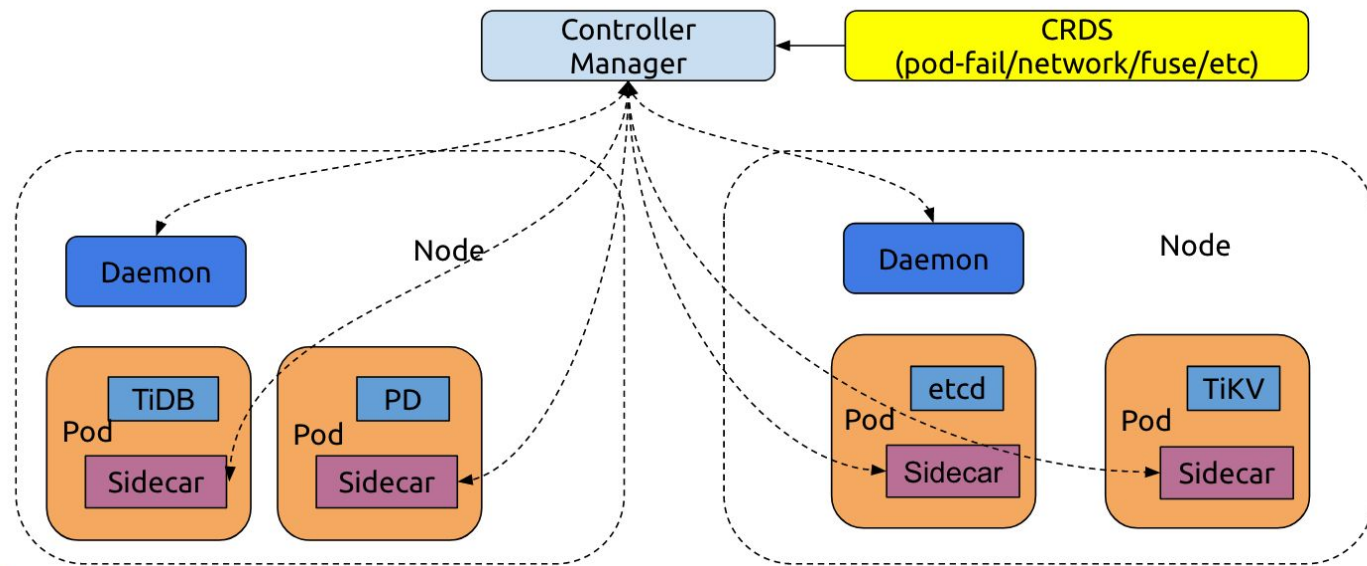


Design

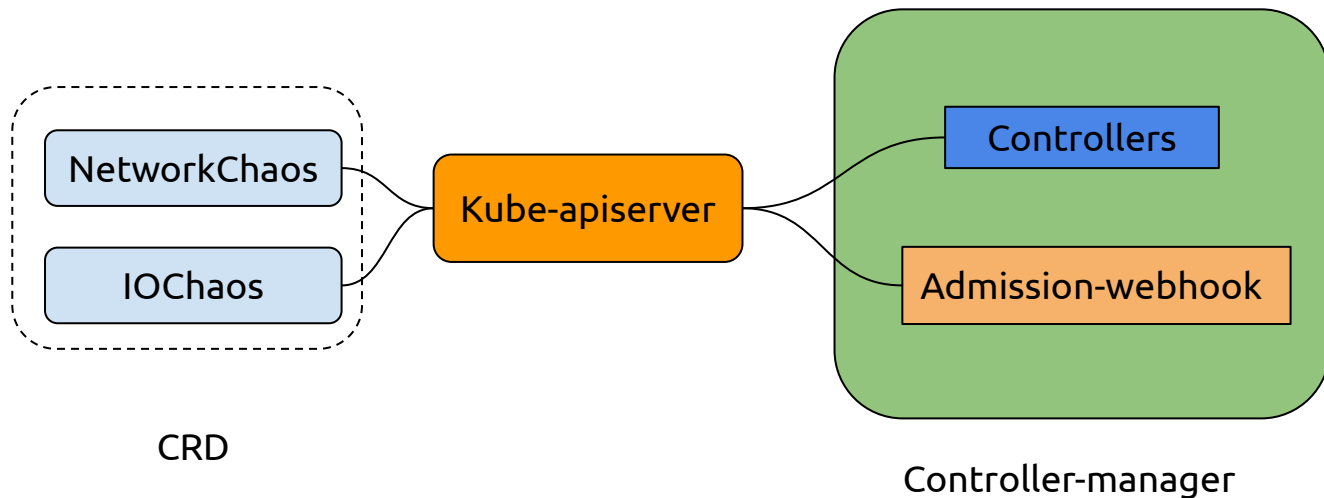
- Based on Kubernetes, can be directly deployed on minikube
- Use CRD to define chaos
- Transparent to applications
- Use YAML to define chaos experiment
- Easy to integrate with other testing platforms
- Convenient to extend new chaos kinds



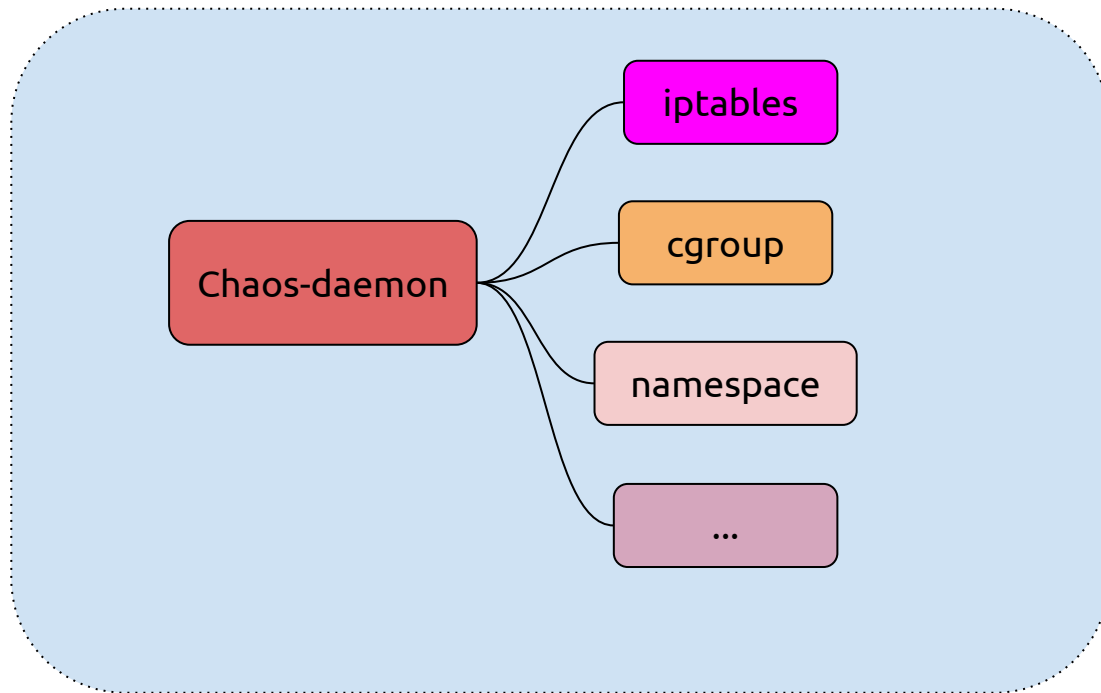
Chaos Operator Architecture



Controller manager

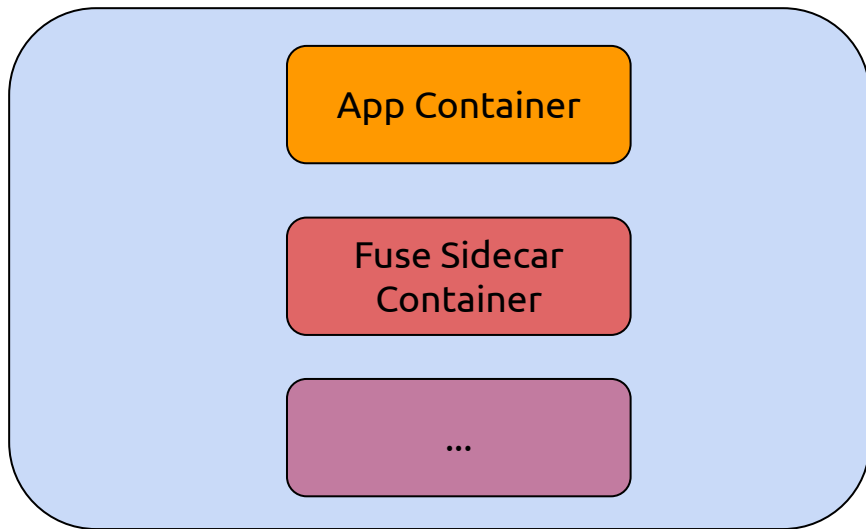


Chaos Daemon



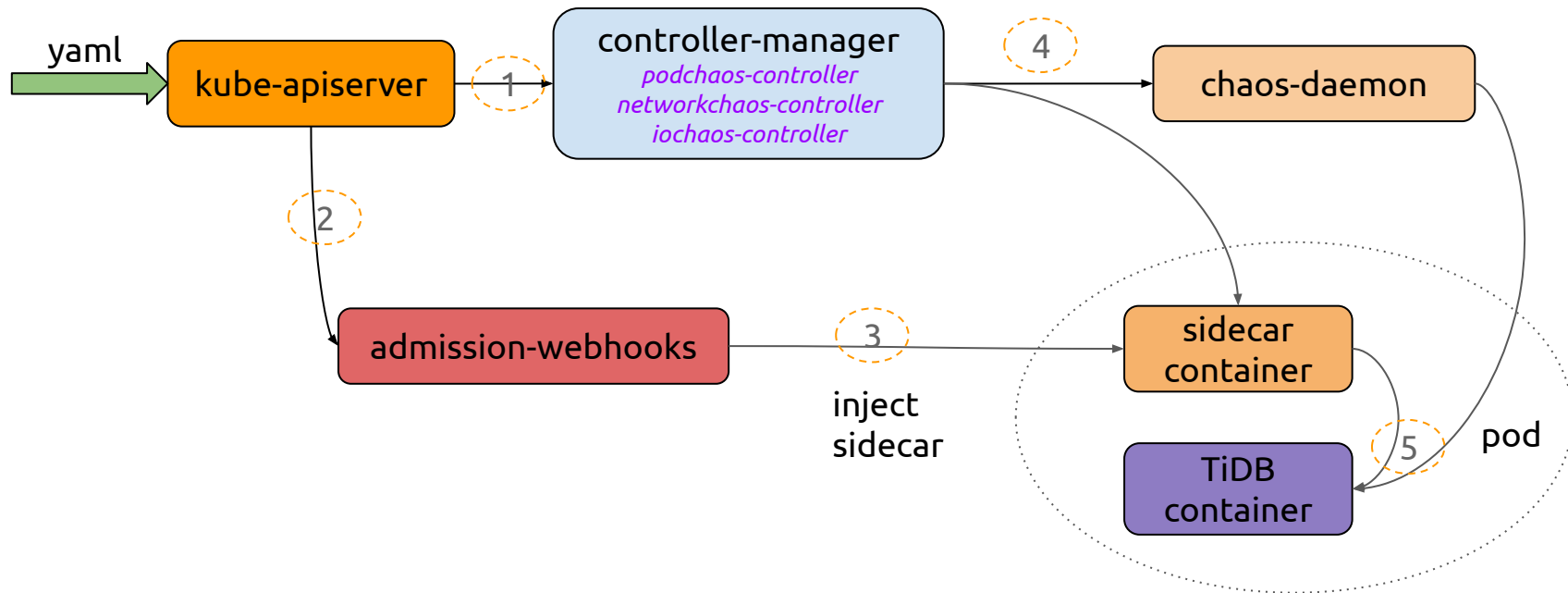
Node

Sidcar



Pod

Workflow (TiDB as an example)



Objects to Attack

Pod

- kill
- process fail

Network

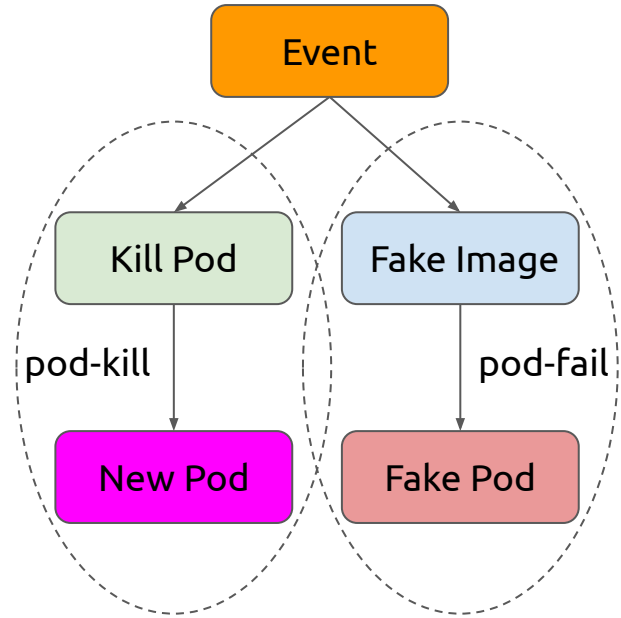
- simulate partition
- introduce latency
- lose / dup packets

I/O

- delay
- errno

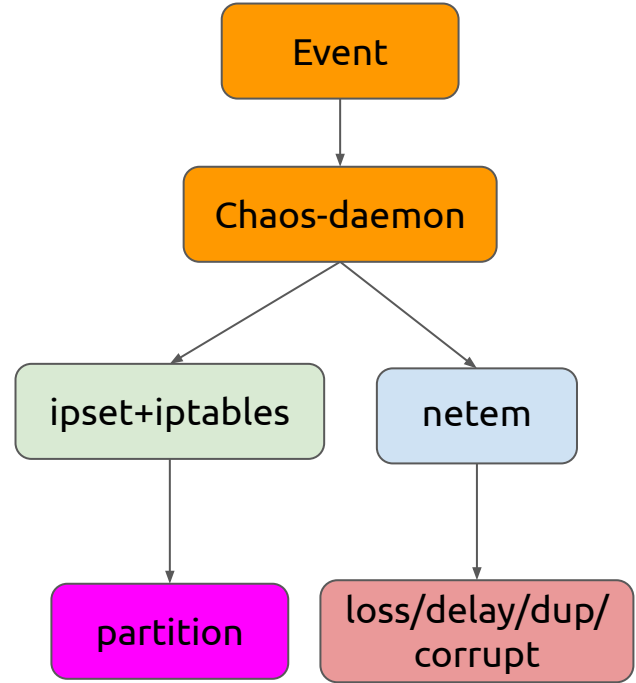
How it works : Pod Chaos

- Receive events from podchaos-controller
- Kill Pod so that this pod will be recreated
- Update Pod with fake image hence a fake pod will be created and will not be recreated by k8s



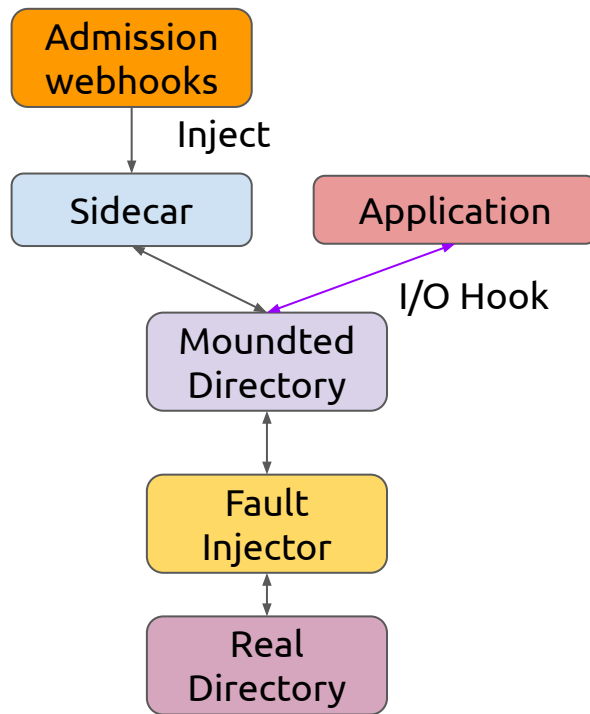
How it works : Network Chaos

- Receive events from networkchaos-controller
- Send requirements to Chaos-daemon
- Enter Pod network namespace to set ipset and iptables
- Enter Pod network namespace to set netem qdisc for pod network interface



How it works: I/O Chaos

- Inject a Sidecar Container
- Use fusemount to mount app's data directory in Sidecar Container
- Start app containers
- Inject I/O fault by fuse-daemon



Future plans

- Chaos Verifier
 - SLA measurement
 - Measure tolerance for different faults
 - Linearizability check for databases
- Chaos Engine
 - Record Chaos action time series and support different kinds of API such as uploading data from test cases
- Chaos Admin
 - Support easy-to-use chaos orchestration interface
- Chaos Cloud
 - Enable Chaos engineering on different cloud vendors such as AWS, GCP

Chaos Observability



Contact

- Github: [pingcap/chaos-mesh](https://github.com/pingcap/chaos-mesh)
- Slack: #sig-chaos-mesh channel in the [TiDB Community](#)
- Twitter: [chaos_mesh](#)

Thank You !

