ShardingSphere

ShardingSphere 的云上之旅
快速部署 + 开箱即用

苗立尧 20220903

苗立尧

Apache ShardingSphere Contributor
SphereEx 云研发负责人

- 负责 Apache ShardingSphere-on-Cloud 子项目
- 目前专注于为 ShardingSphere 构建云上解决方案

目录

ShardingSphere

ShardingSphere
云化的机遇和挑战

数据库碎片化与云原生浪潮

微服务带来了快速响应市场的能力，增加了部署、上线、更新、测试等管理维护的复杂度。

以 Kubernetes 为代表的云原生浪潮帮助解放了运维的生产力，使得微服务上线发布和治理提升了一个新台阶。

随着业务场景越来越多元，数据应用的方案呈现烟囱状，对数据的管控有孤岛化的趋势，摆在技术人员面前的是“选型难、成本高、管控复杂”等问题。

Uber Microservice
https://twitter.com/msuriar/status/1110244877424578560/photo/1

ShardingSphere

ShardingSphere 架构设计

ShardingSphere 的架构在演变的过程中与云原生十二因子的约定不谋而合

The Twelve Factors

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

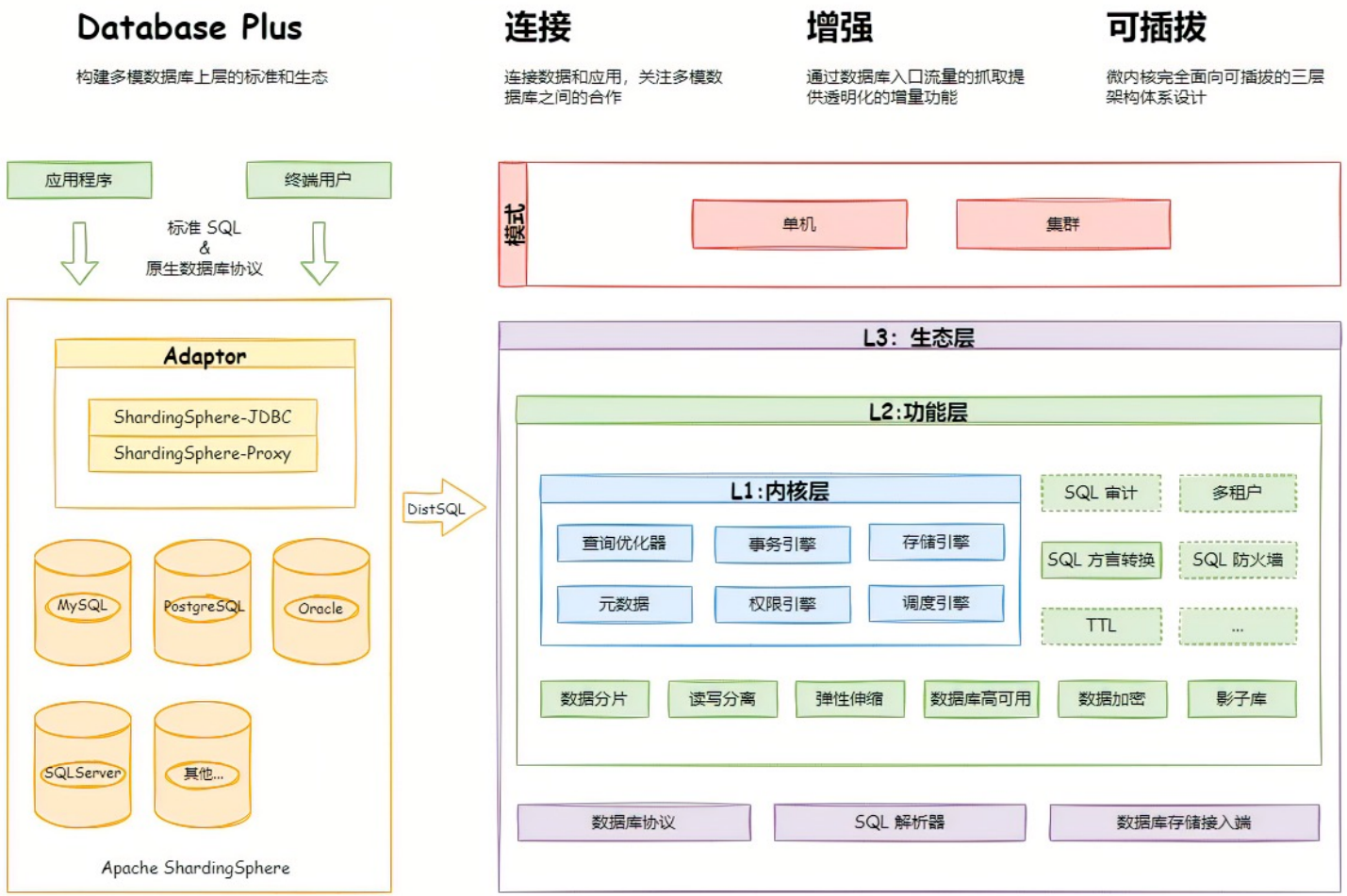
IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

https://12factor.net/



ShardingSphere 云化的机遇和挑战

如何把一只大象塞到冰箱里？

趋势：Gartner 曾预测到今年全球会有 75% 的数据库上云

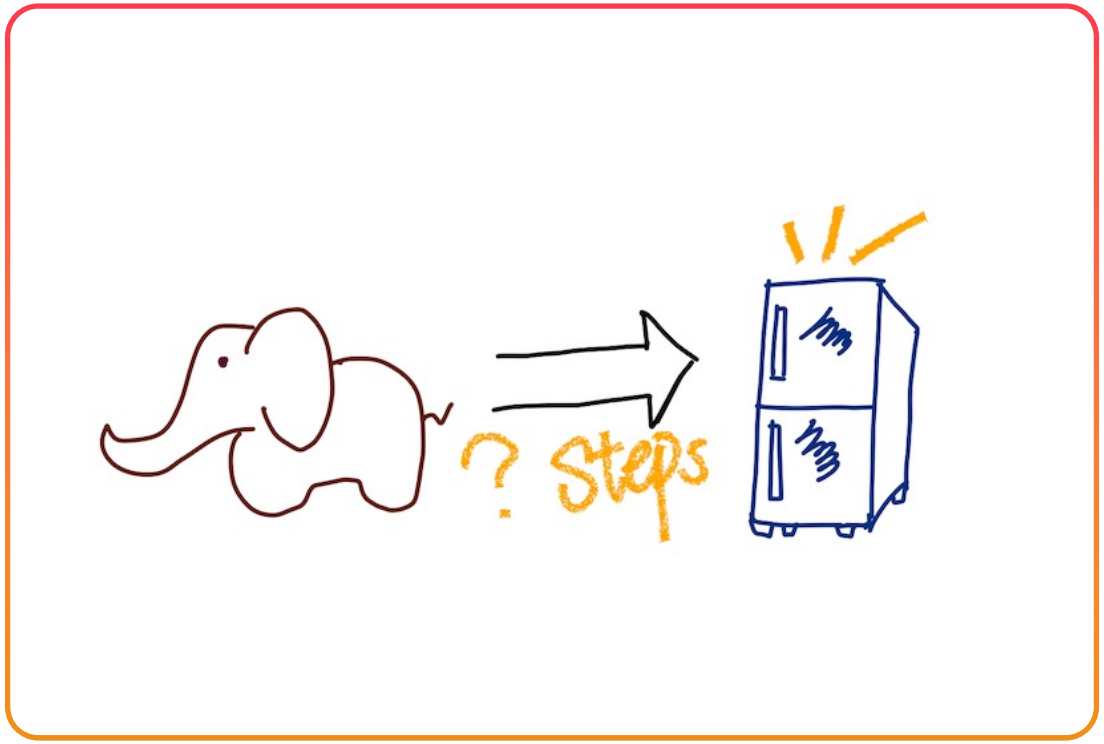
机遇：

- AWS DynamoDB, RDS, Aurora
- Google CloudSQL
- Kubernetes
- ...

挑战：

- 快速部署和迁移
- 高可用和灾备
- 安全和合规
- ...

本次分享是关于 the missing puzzle of cloud



ShardingSphere

ShardingSphere 的云上实践

数据库上云用户最关心的九个问题

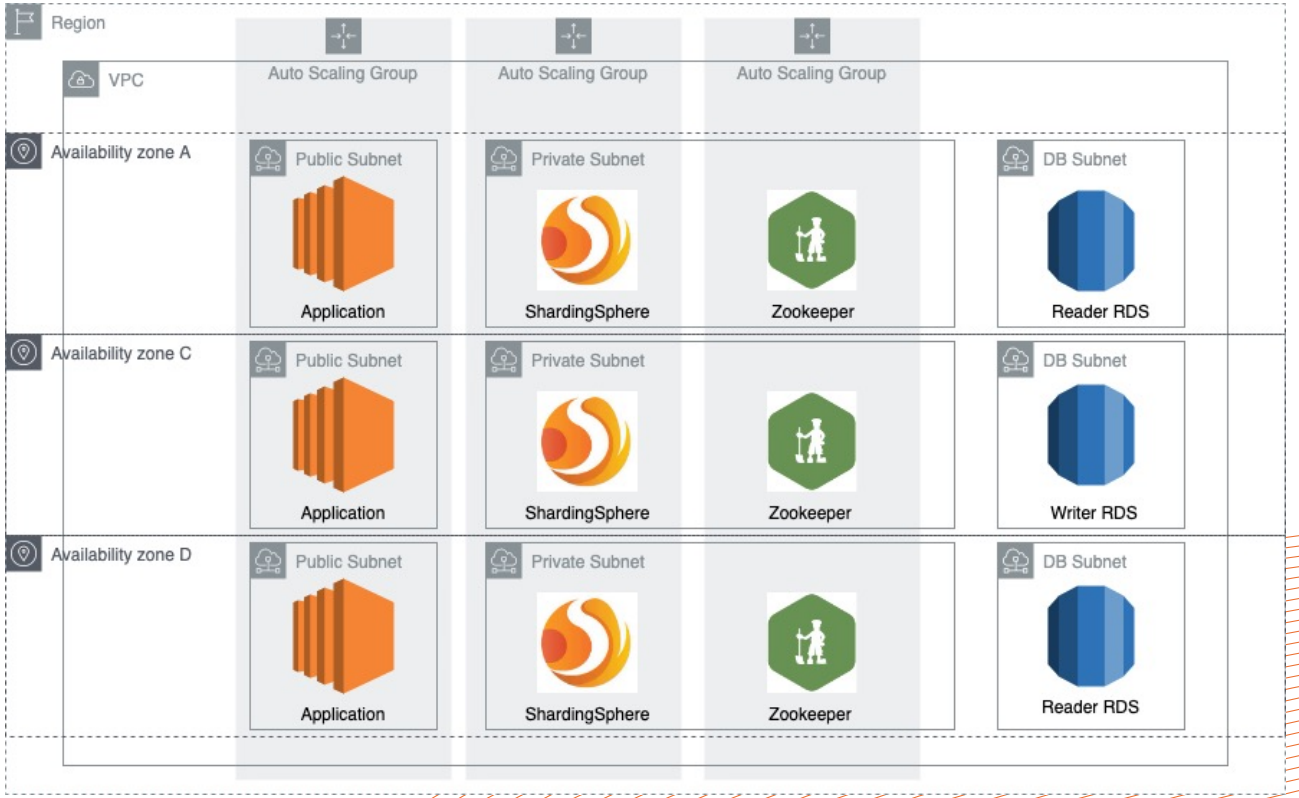
- 快速部署
- 迁移
- 高可用
- 灾备
- 备份
- 安全合规
- 故障排查
- 配置
- 监控报警

云上快速部署

云上的部署效率极高，通常都会有多种工具链帮助加速部署，比如云上版本的 Puppet 和 Ansible。而近几年火热的 Infrastructure as Code 大大简化了版本化的部署过程，比如阿里云的 ROS，华为云的 AOS。

AWS 提供了丰富的应用部署工具集，包括 OpsWork、SystemManager、CloudFormation 等，可以统一配置 VPC、子网、安全组等基础网络环境，同时可以部署 EC2、节点组、自动扩容组等计算资源，实例运行的时候自动启动指定应用。

ShardingSphere Proxy 得益于其简洁的部署架构，将治理节点、计算节点和存储节点分离开，并根据需要组合不同的实例，进行快速的部署。

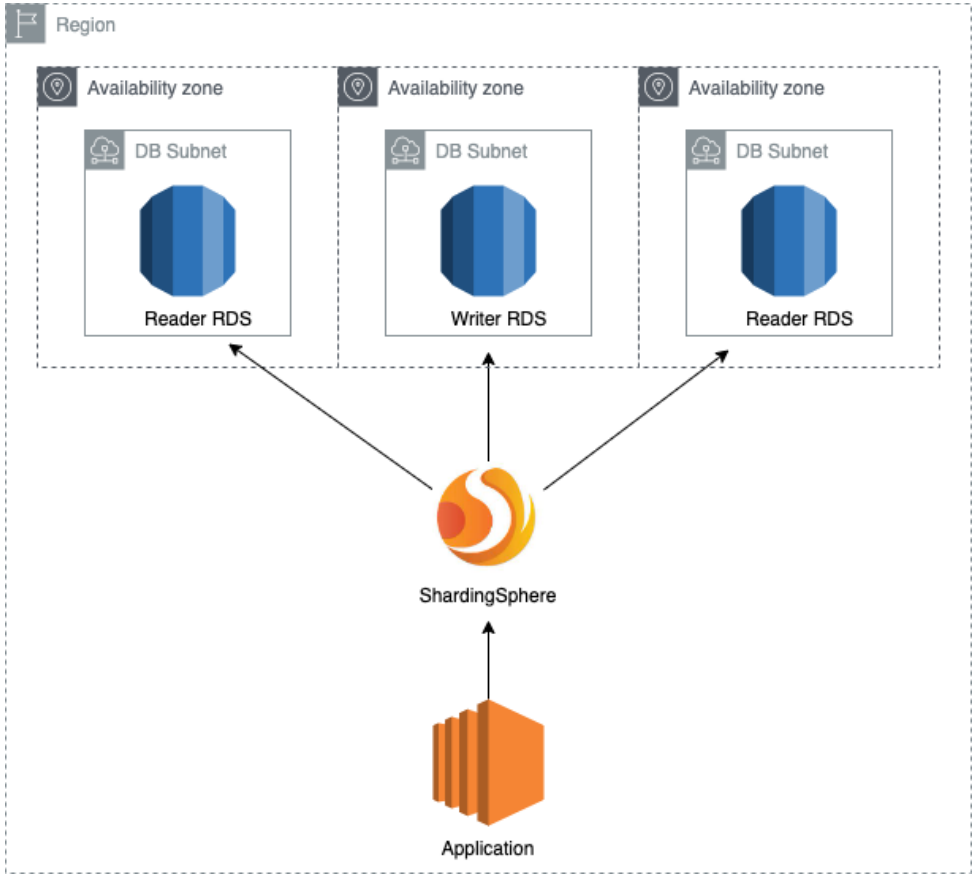


云上高可用实践

在云上，通常会引入 Region、Availability Zone 的概念用来描述部署位置，并基于此构建不同的高可用方案。

比如阿里云、华为云和 AWS 的 RDS 都有单 AZ 和多 AZ 的部署方式，并有多种灾备切换方式。

ShardingSphere 自身的去状态化保证其良好的弹性能力，而它还提供了 Database Discovery 能力，实现用户应用无感知的数据库维护升级和灾备切换。



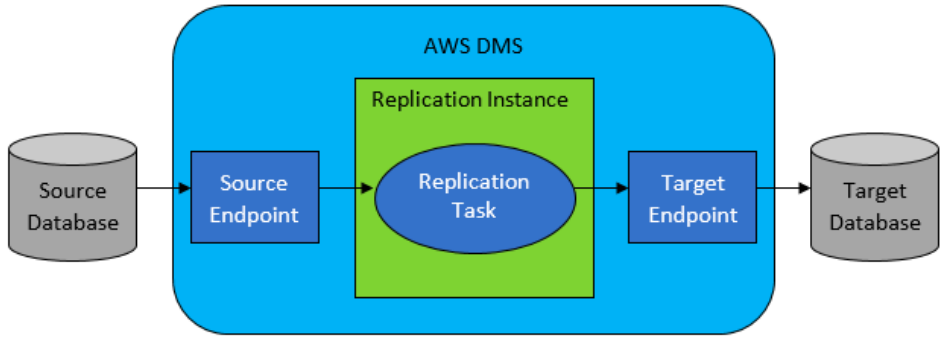
https://github.com/apache/shardingsphere-on-cloud/issues

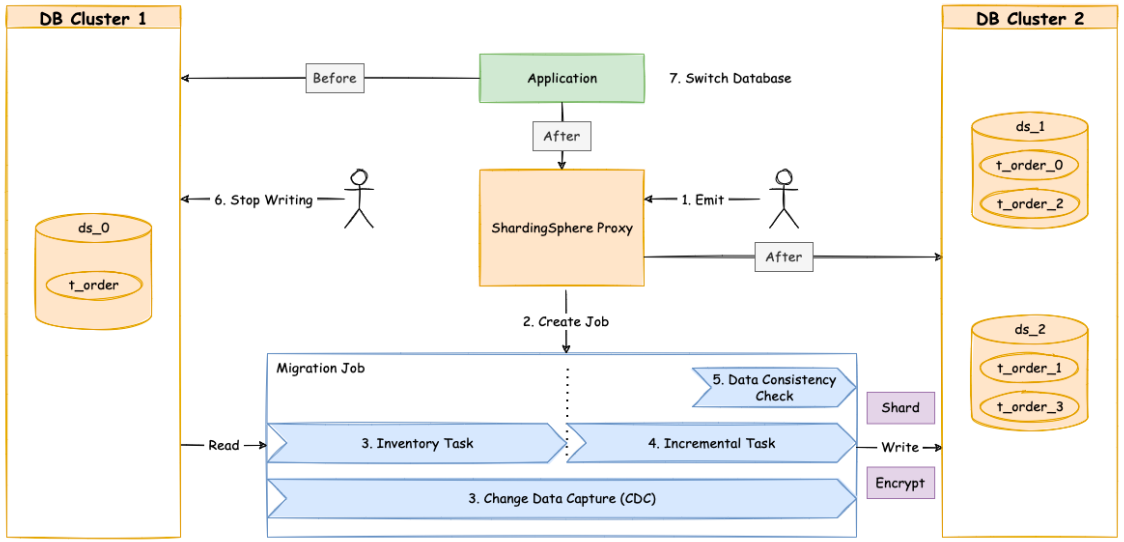
云上数据迁移实践

为了将存量数据或遗留数据库系统迁移至云上，一般都会通过 Data Migration Service 的方式进行。常见的有阿里云的 DTS、华为云的 DRS，以及 AWS 的 DMS 等。

DMS 提供的迁移能力，可以以并行任务的方式将数据从本地或原有的云服务上迁移至 AWS RDS，配合 Schema Conversion Tool 还可以实现 Schema 元数据变更。

ShardingSphere 在面对存量数据分片、分片键变更等场景里同样支持数据迁移，类似 DMS，配置简单，最小化数据不可用的时间窗口以及保证数据的正确性。





https://github.com/apache/shardingsphere-on-cloud/issues

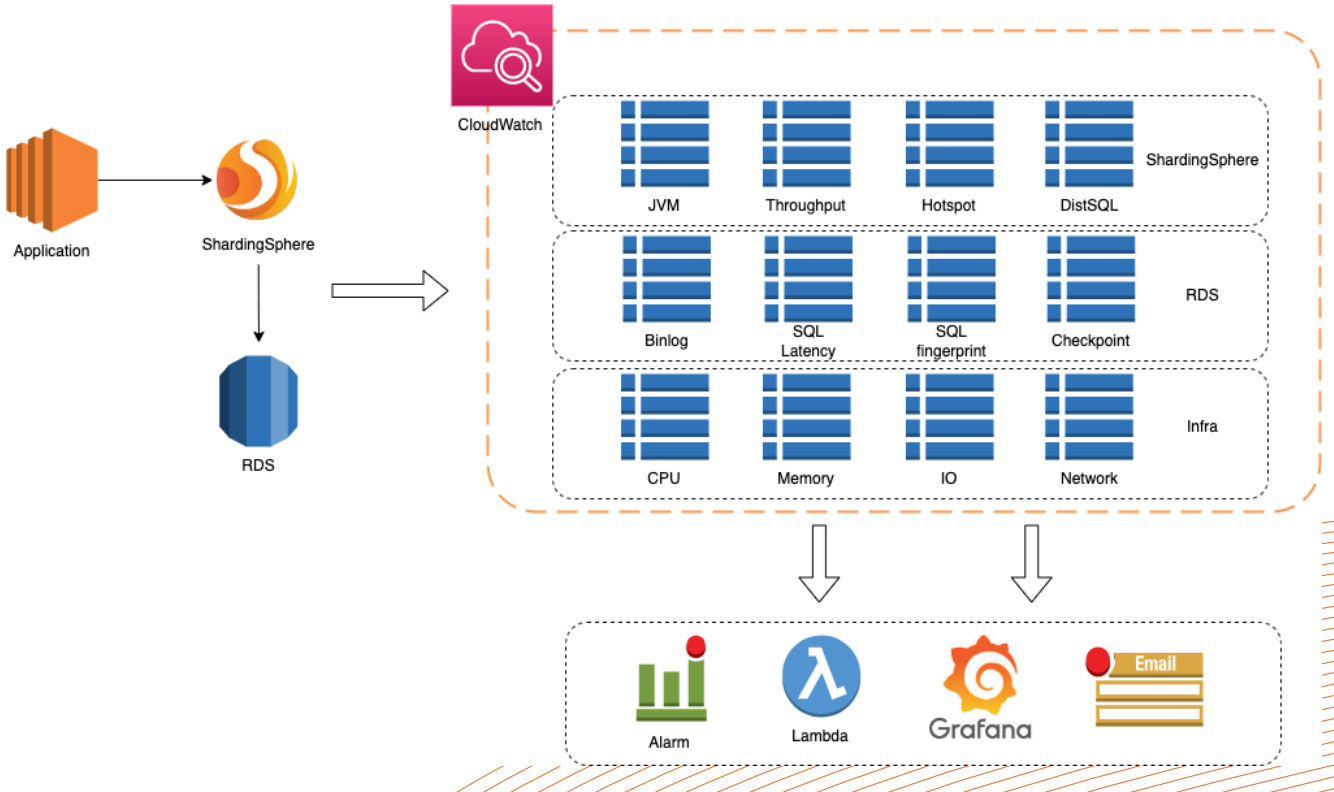
云上可观测性实践

对于可观测性建设的一个要点是尽量减少信息孤岛的影响。云上通常已经具备了从软件到硬件整套分层的监控和洞察能力，还可以配合通知服务、Serverless 服务等，构建多种指标消费渠道。

AWS RDS 就提供了 CloudWatch 和 Performance Insight 两种丰富的监控和透视指标，并支持报警和事件处理。

ShardingSphere 利用 Agent 采集数据，弥补了从原有监控的不足，帮助用户更清楚地发现热点分片、慢 SQL 瓶颈、DistSQL 统计信息等，形成完整的监控洞察链路。

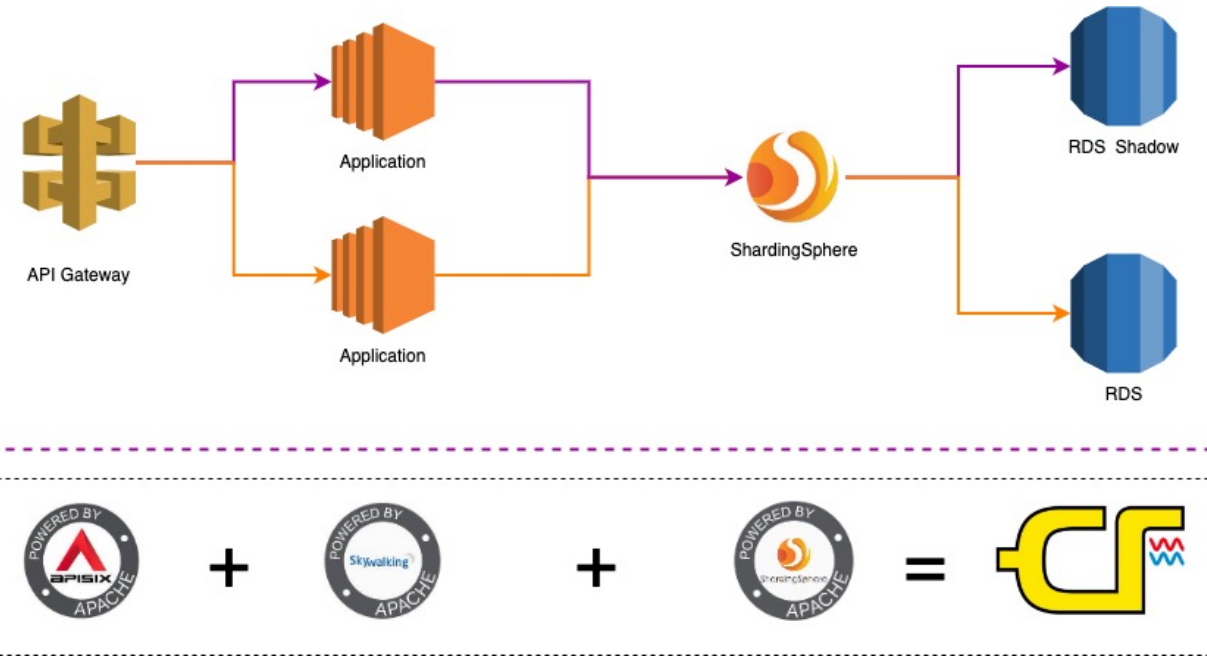
CloudWatch 指标还可以被展示到 Grafana，构成完整的端到端监控大屏。



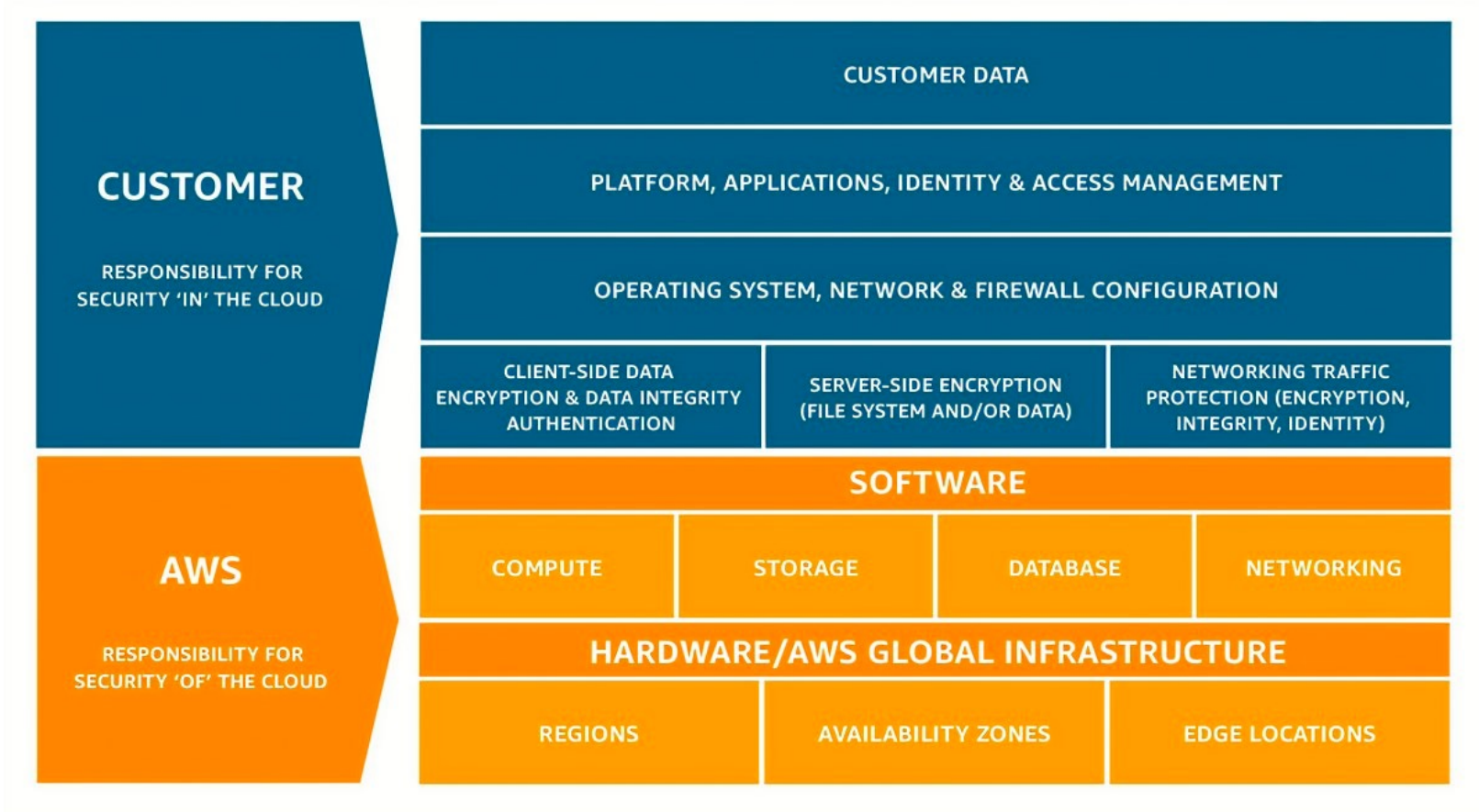
云上影子库实践

2022年初的时候，Apache ShardingSphere 联合 Apache APISIX 和 Apache Skywalking 共同推出了 Cyborgflow 项目，实现从网关到数据的全链路、可观测的压测方案。

在 AWS 上可以利用多套 RDS 或者 Aurora Clone 快速部署一套用于压测的数据库实例，和 ShardingSphere 的影子库能力二者相互配合，通过 Hint 或基于列的影子算法，将数据请求路由到影子库，实现全链路的压测能力。



AWS 责任共担模式



ShardingSphere 可以帮助用户更好地实现责任共担模式

云上加密实践

数据安全是重中之重，不同的行业有着对应的审计要求，比如细粒度的认证授权、操作日志需要脱敏保留、数据传输加密、机密数据加密存储等。

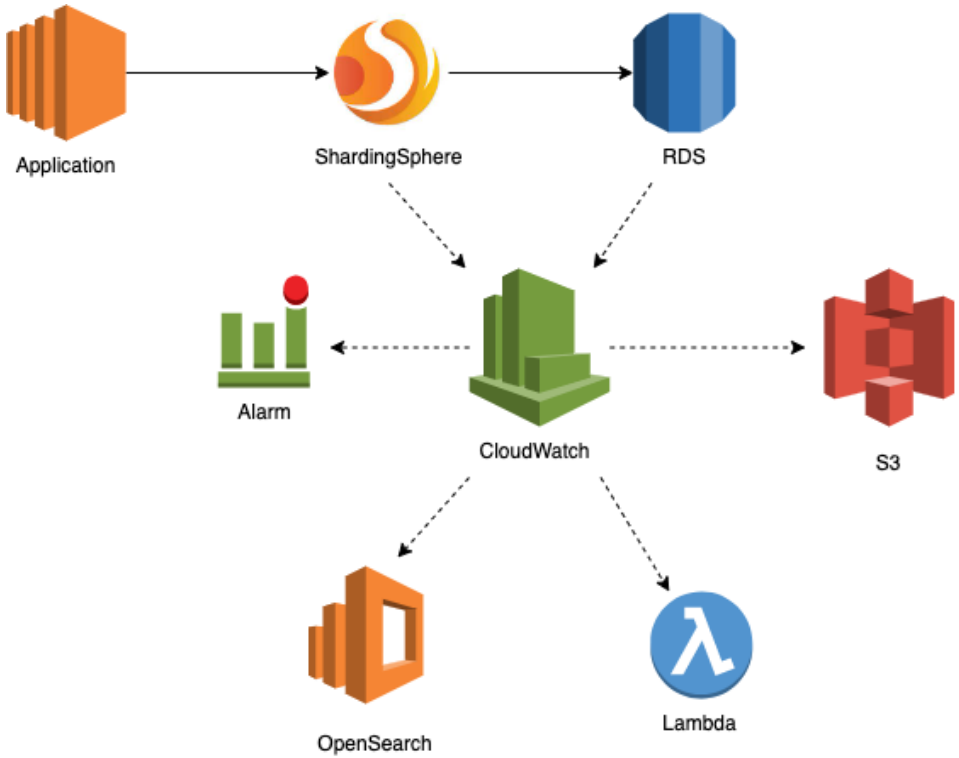
在 阿里云、华为云和 AWS 上提供了多种安全和合规工具，比如 KMS，CloudHSM 以及 CMK 帮助解决加密相关的问题。

ShardingSphere 提供了强悍的数据加密能力，通过逻辑列、密文列、明文列和查询辅助列的组合，实现对业务影响最小、用户体验友好的数据加密体验。

云上日志审计实践

而对于数据库行为审计，除了可以通过在 RDS 配置中进行实现外，还可以通过 ShardingSphere 进行表级别的权限控制，以及敏感操作事件推送。这些事件记录被存储在 CloudWatch LogGroups 对应的 S3 中，以满足审计检查需要，同时这些事件还可以导入 OpenSearch 进行分析，或者发送至 Lambda 进行更具体的计算等。

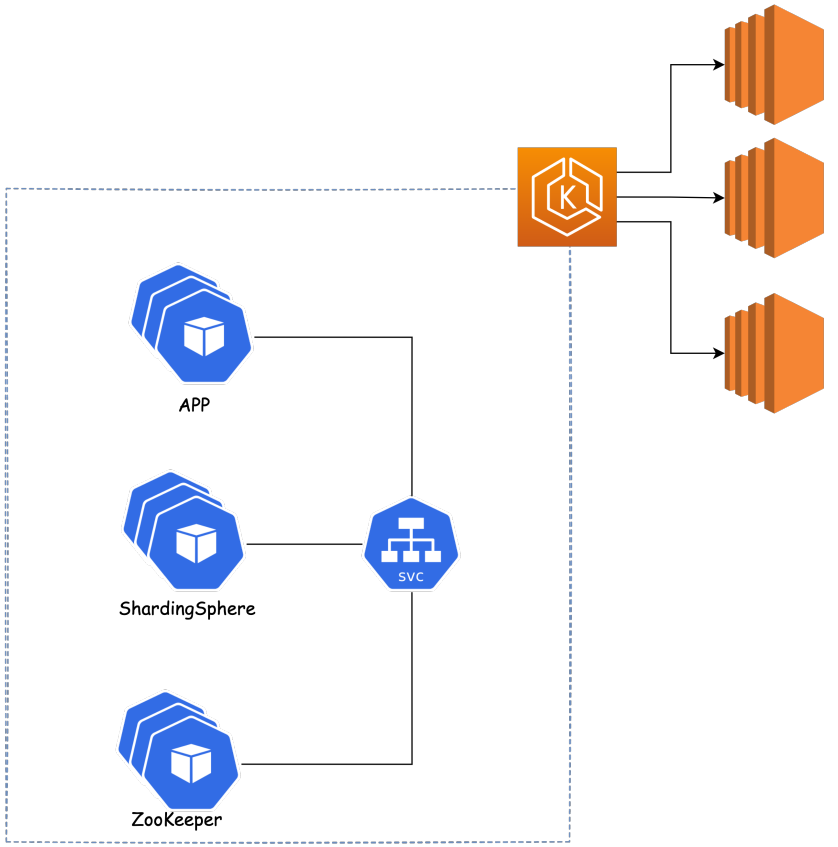
ShardingSphere 是数据保护的第一道屏障。



你好！Kubernetes

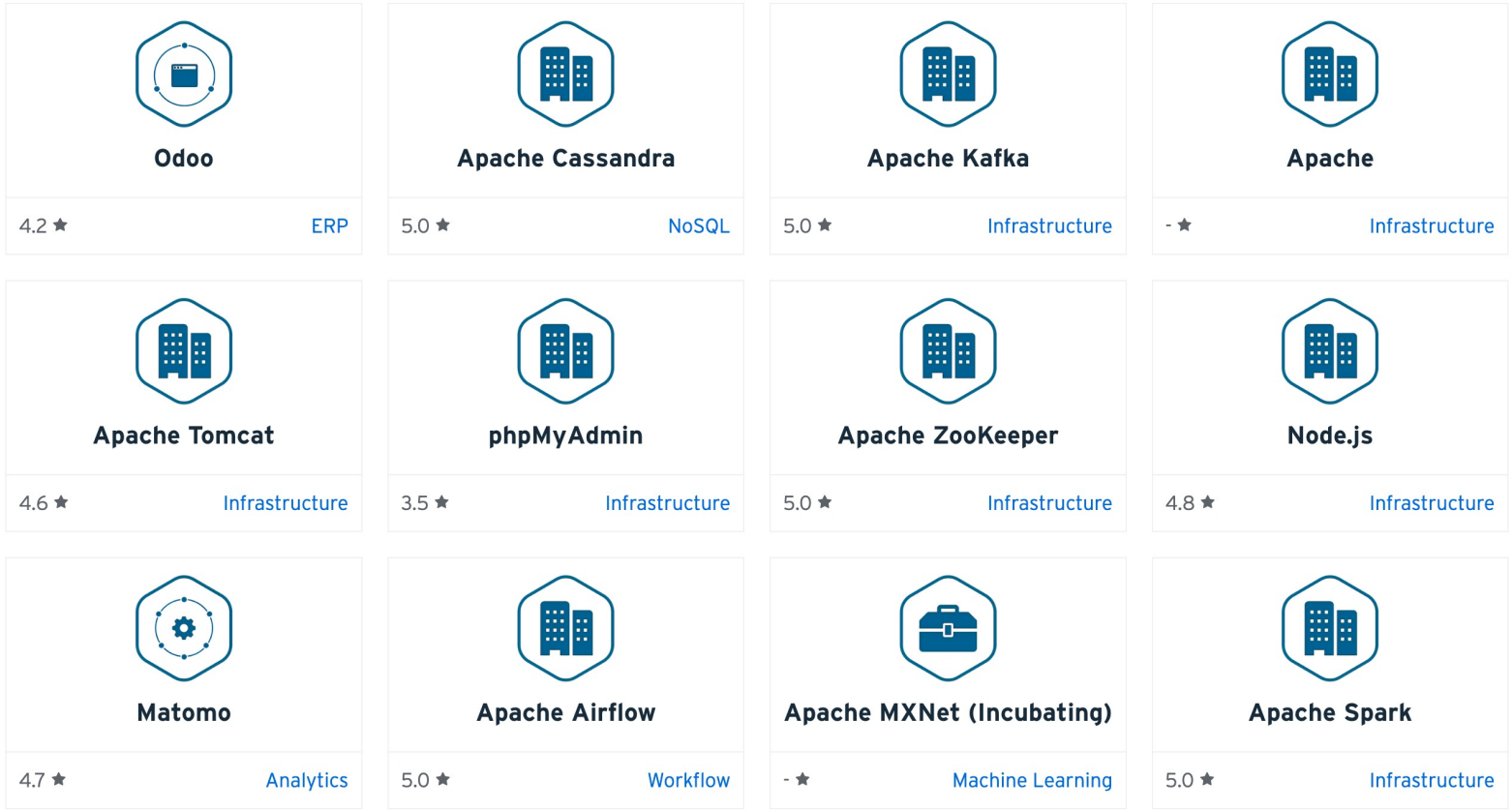
作为容器编排的事实标准，Kubernetes 为大规模应用的部署提供了新的解决方案，也掀起了云原生的浪潮。

越来越多的应用开始选择在 Kubernetes 进行部署和交付，ShardingSphere 社区先后贡献了 Helm Chart 和 Operator 两种云原生方式，针对性解决云原生的快速部署和自动化运维的问题。



云原生交付：Helm Charts

Helm Charts 是 Kubernetes 社区推荐的应用交付和发布模式。



Helm Charts 示例

通过 install、upgrade、rollback、uninstall 子命令实现应用版本的生命周期管理

```
→ shardingsphere-charts (log) ✗ helm install shardingsphere-proxy shardingsphere-proxy-1.1.0.tgz -n sstest
NAME: shardingsphere-proxy
LAST DEPLOYED: Wed Jun  8 21:23:18 2022
NAMESPACE: sstest
STATUS: deployed
REVISION: 1
TEST SUITE: None
→ shardingsphere-charts (master) ✗ helm upgrade shardingsphere-proxy shardingsphere-proxy -n sstest
Release "shardingsphere-proxy" has been upgraded. Happy Helming!
NAME: shardingsphere-proxy
LAST DEPLOYED: Fri Jun 10 10:54:47 2022
NAMESPACE: sstest
STATUS: deployed
REVISION: 2
TEST SUITE: None
→ shardingsphere-charts (master) ✗ helm rollback shardingsphere-proxy -n sstest
Rollback was a success! Happy Helming!
→ shardingsphere-charts (master) ✗ helm list -n sstest
NAME                    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                           APP VERSION
shardingsphere-proxy    sstest          1               2022-06-08 21:29:12.227283 +0800 CST    deployed        shardingsphere-proxy-1.1.0      5.1.2
→ shardingsphere-charts (log) ✗ kubectl get pod -n sstest
NAME                                    READY   STATUS     RESTARTS   AGE
shardingsphere-proxy-5dd945dc7d-p66bm   0/1     Init:0/1   0          31s
shardingsphere-proxy-5dd945dc7d-t4z46   0/1     Init:0/1   0          31s
shardingsphere-proxy-5dd945dc7d-vhbxv   0/1     Init:0/1   0          31s
shardingsphere-proxy-zookeeper-0        1/1     Running    0          26s
```

Helm Charts 的挑战

劣势

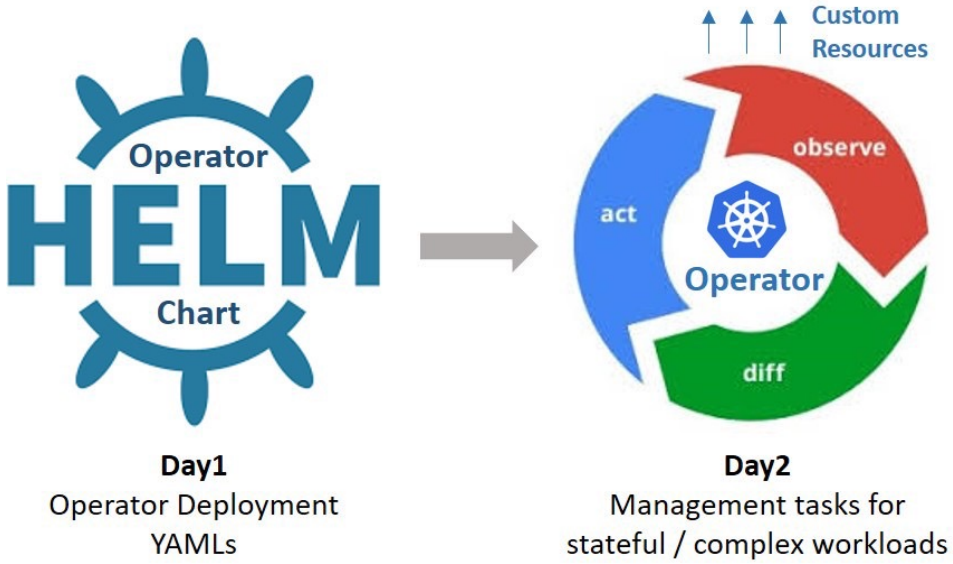
- 只能使用 Kubernetes 原生资源
- 只能管理部署流程，后续运维流程无法管理
- 对于复杂的部署模式，编写 Charts 复杂度也会增加
- ...

优势

- 简化了对于描述文件的配置过程
- 版本化管理部署流程
- 快速的回滚、升级、清理
- ...

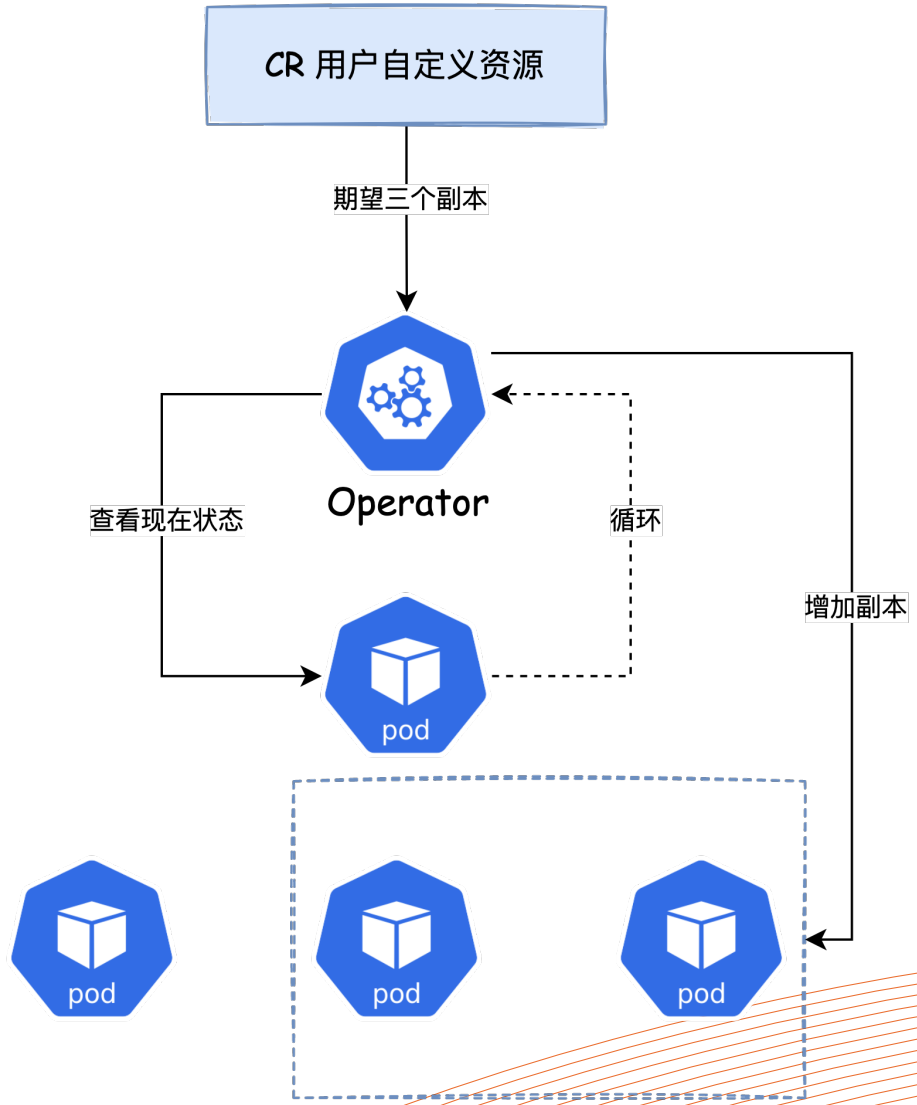
基于 Operator 模式的自动化

Operator 是什么



Day1
Operator Deployment
YAMLs

Day2
Management tasks for
stateful / complex workloads

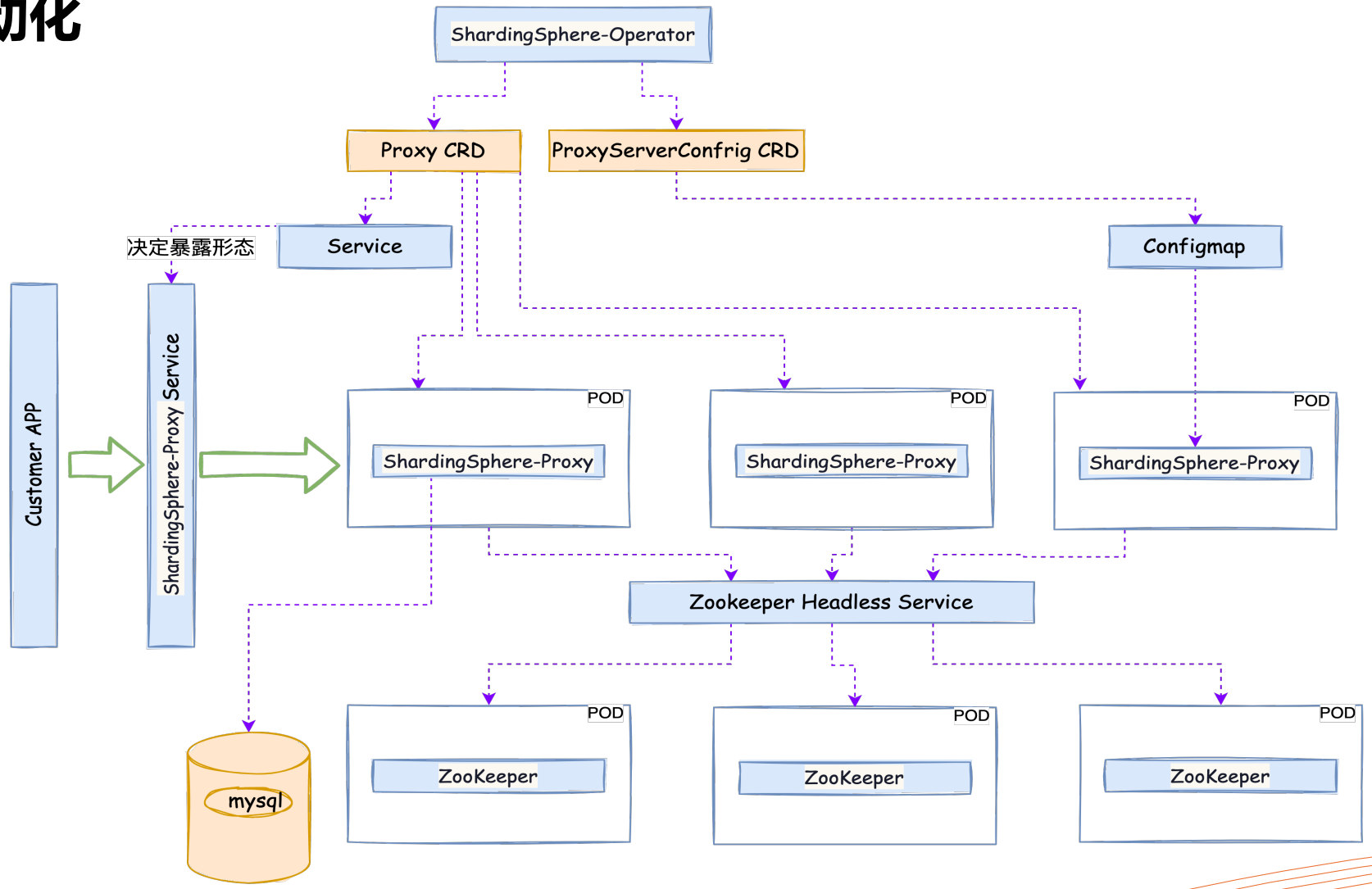


基于 Operator 模式的自动化

Operator 的架构设计

挑战：

- 配置复杂
- 治理节点依赖
- 虚拟机和云上部署行为不同
- 弹性和水平扩容

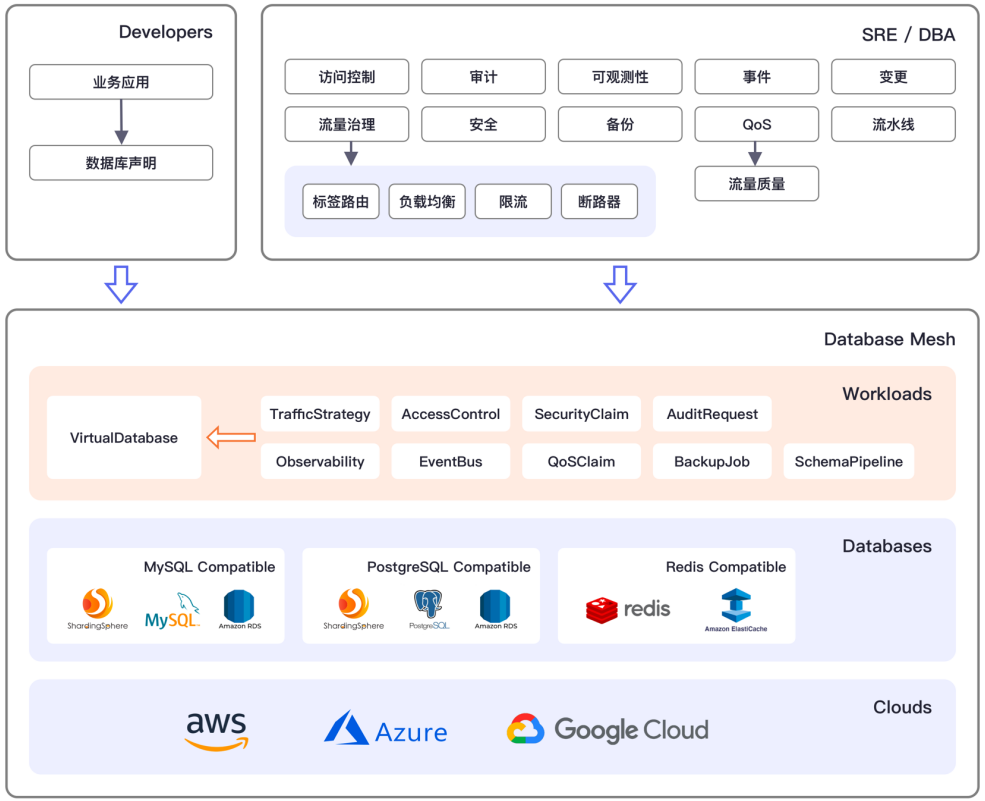


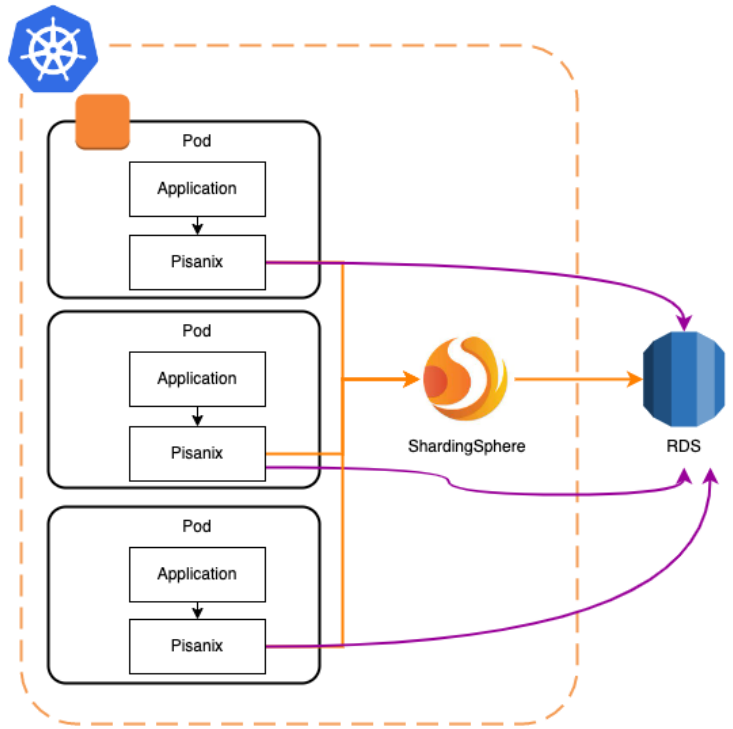
Operator 示例

```
→ ~ helm repo add shardingspherecloud https://sphereex.github.io/shardingsphere-on-cloud/
"shardingspherecloud" has been added to your repositories
→ ~ helm search repo shardingspherecloud
NAME                                            CHART VERSION   APP VERSION     DESCRIPTION

shardingspherecloud/shardingsphere-cluster      0.1.0           5.1.2           A Helm chart for ShardingSphere-Cluster

shardingspherecloud/shardingsphere-operator     0.1.0           0.1.0           A Helm chart for ShardingSphere-Operato
r
→ ~ kubectl create ns  shardingsphere-operator
namespace/shardingsphere-operator created
→ ~ helm install shardingsphere-operator shardingspherecloud/shardingsphere-operator -n shardingsphere-operator
NAME: shardingsphere-operator
LAST DEPLOYED: Thu Jul 21 14:43:45 2022
NAMESPACE: shardingsphere-operator
STATUS: deployed
REVISION: 1
TEST SUITE: None
→ ~ kubectl create ns  shardingsphere
namespace/shardingsphere created
→ ~ helm install  shardingsphere-cluster shardingspherecloud/shardingsphere-cluster -n shardingsphere
NAME: shardingsphere-cluster
LAST DEPLOYED: Thu Jul 21 14:44:23 2022
NAMESPACE: shardingsphere
STATUS: deployed
REVISION: 1
TEST SUITE: None
→ ~
→ ~
→ ~ kubectl get pod -n shardingsphere
NAME                                      READY   STATUS     RESTARTS   AGE
shardingsphere-cluster-6455899b85-2lxhg   0/1     Init:0/1   0          10s
shardingsphere-cluster-6455899b85-558hf   0/1     Init:0/1   0          10s
shardingsphere-cluster-6455899b85-znpx4   0/1     Init:0/1   0          10s
shardingsphere-cluster-zookeeper-0        0/1     Running    0          10s
```

和 Database Mesh 的方案

Database Mesh 可以为 ShardingSphere 提供更加灵活的流量治理能力



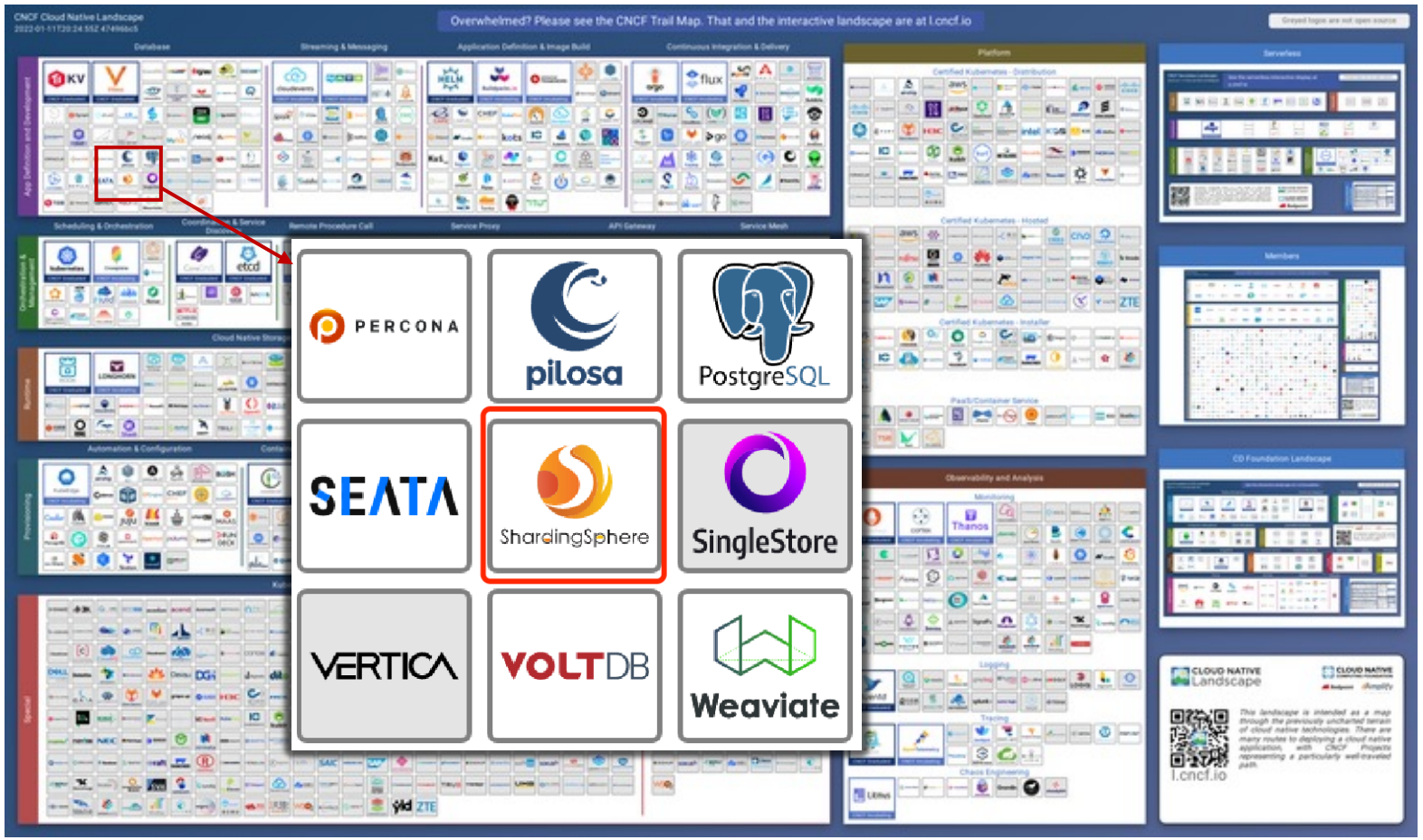


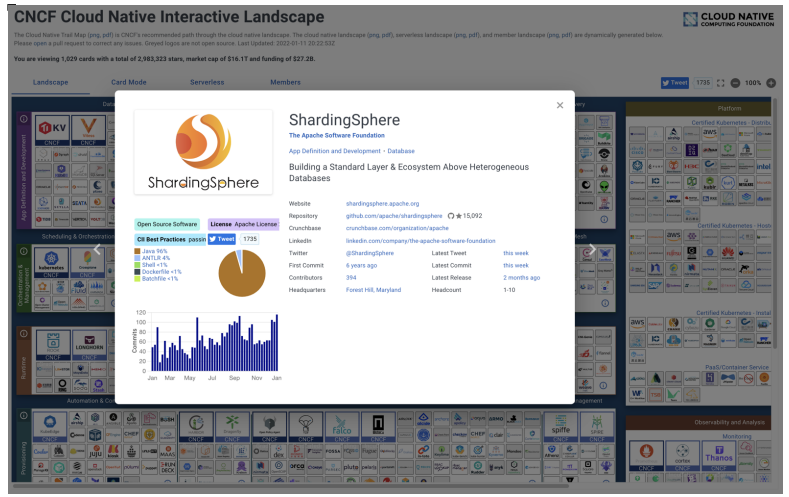
ShardingSphere

社区与未来规划

ShardingSphere 与 CNCF

ShardingSphere 已入选了 CNCF 全景图，成为云原生数据库生态的重要组成部分





社区

目前 SphereEx 已经将 ShardingSphere-on-Cloud 项目作为 子项目捐赠给了 Apache ShardingSphere 社区，并同时社区已经发表了多篇和云计算 、Kubernetes 的相关技术文章：

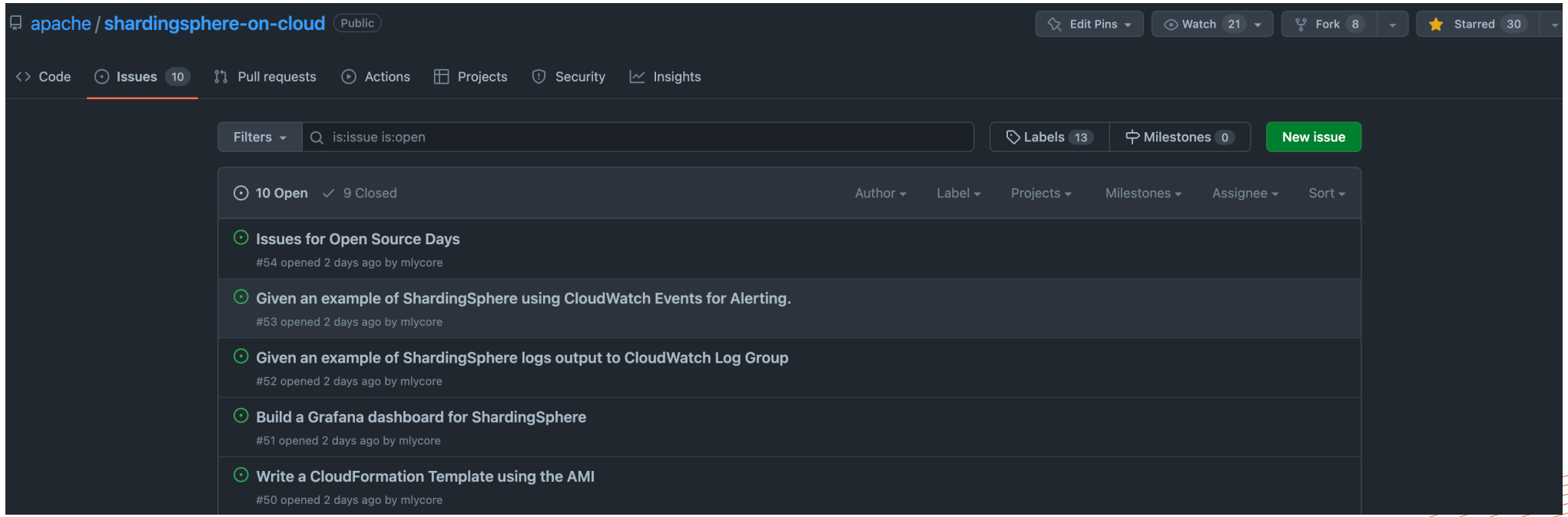
- Amazon Aurora 读写分离扩展之 ShardingSphere-JDBC 篇
- Amazon Aurora 的读写能力扩展之 ShardingSphere-Proxy 篇
- Creating a Secure Distributed Database Cluster Leveraging Your Existing Database Management System
- ShardingSphere 云上实践：开箱即用的 ShardingSphere-Proxy 集群
- Database Plus 的云上之旅：SphereEx 正式开源 ShardingSphere on Cloud 解决方案
- 开源二三事 | ShardingSphere 与 Database Mesh 之间不得不说的那些事

此外在 ShardingSphere Slack 已经开设了相关的 channel #shardingsphere-on-cloud 方便交流

未来规划

- 进一步丰富数据库计算增强能力在公有云上的实践，覆盖快速部署、高可用、迁移、安全合规、可观测性等。
- 提高 Helm Charts 快速部署能力
- Operator 优化 ShardingSphere Proxy 集群部署结构、自动扩容、混沌工程等

在 github.com/apache/shardingsphere-on-cloud 仓库也创建了一些容易上手的 issue 帮助大家快速参与：



ShardingSphere

Cloud is future
and now it comes !

ShardingSphere

谢谢观看