

100Kubernetes 系列（九十三）CloudNative-PG：在 K8s 中运行 PostgreSQL

介绍

过去 6 个月我一直在从事一个非常酷的项目，该项目需要使用 AxonOps 在多个 Apache Cassandra 集群上运行大量数据。一些数据库在 Kubernetes 中运行，另一些数据库在 Kubernetes 之外运行，而所有客户端都在 K8s 中。

Apache Cassandra 等分布式数据库有一件事做得不好：计数器。想象一下点击数、投票数、浏览量等。Apache Cassandra 确实支持 counters，但它们并非没有问题。我的建议是，如果可以的话，避免使用它们。

因此，我的任务是在 Kubernetes 中运行 PostgreSQL 数据库。

但在 Kubernetes 中运行数据应用程序并不总是那么容易。您通常需要一个优秀的 Operator 通过执行故障转移和备份等关键操作来帮助管理数据库。

我已经使用过 CrunchyData 和 StackGres Operator。它们都很好，但并不完美。我的一位同事建议看看 CloudNative-PG，所以我对其进行了测试。

CloudNativePG 是一个开源 Operator，旨在管理在私有、公共、混合或多云环境中运行的任何受支持的 Kubernetes 集群上的 PostgreSQL 工作负载。CloudNativePG 遵循 DevOps 原则和概念，例如声明性配置和不可变基础设施。

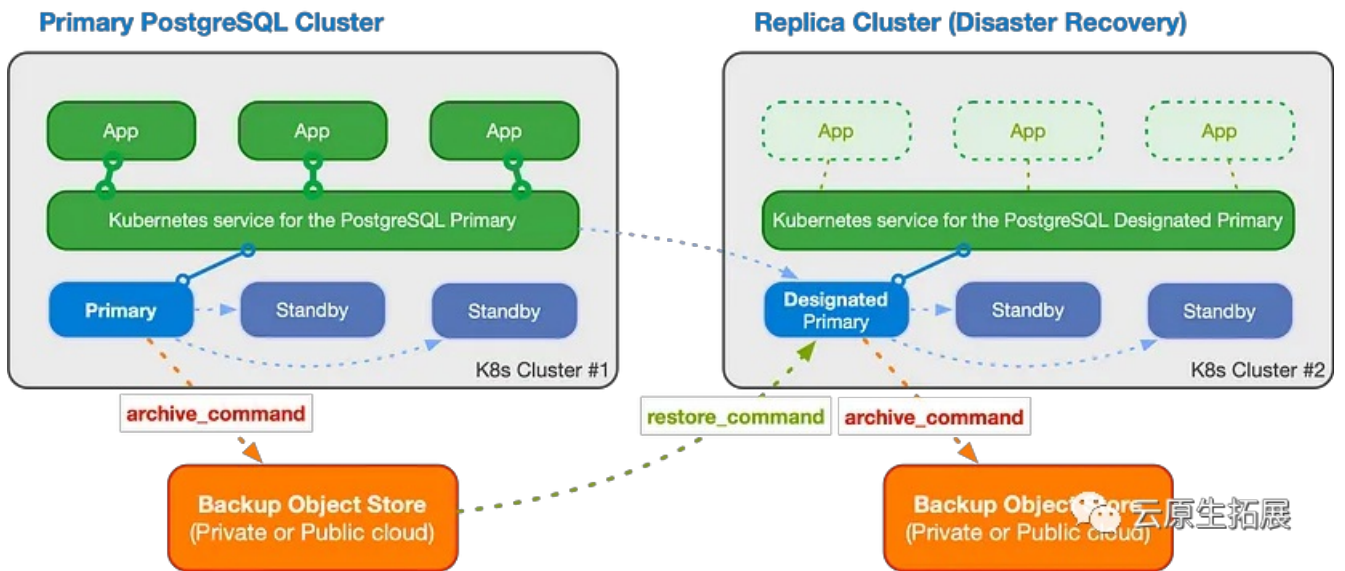
它的好处

我注意到的第一件事是他们使用不可变的容器。我真的很喜欢这个功能，因为在处理数据时，安全性必须放在第一位。此外，TLS 默认情况下与自行部署的证书一起使用，您可以使用 cert-manager 自带证书。

它使用 barman 进行备份和恢复，并支持 PITR（时间点恢复）。配置非常简单，只需要对象存储来存储备份（即：S3、GCP、Azure、MinIO 等）。

```
backup:
  barmanObjectStore:
    destinationPath: s3://cloudnative-pg-clusters/pg001
    s3Credentials:
      accessKeyId:
        name: backups-s3
        key: AWS_ACCESS_KEY_ID
      secretAccessKey:
        name: backups-s3
        key: AWS_SECRET_ACCESS_KEY
```

您可以使用几个用于灾难恢复的配置行创建副本集群。



在谈论数据时，您始终需要确保将数据复制到至少一个位置。如果你在云中，你不需要的是某个区域消失。（事实上，别人在伦敦大火期间亲身经历过这一点，当时我们不得不将故障转移到欧洲）



CloudNativePG Operator 允许您创建同步复制约束，以确保它们分布在不同的可用区。

```
spec:
  instances: 3
  postgresql:
    syncReplicaElectionConstraint:
      enabled: true
      nodeLabelsAntiAffinity:
        - topology.kubernetes.io/zone
```

默认情况下使用异步复制，但如果需要，您也可以启用同步复制。

```
maxSyncReplicas    <integer>
    The target value for the synchronous replication quorum, that can be
    decreased if the number of ready standbys is lower than this. Undefined or
    0 disable synchronous replication.

minSyncReplicas    <integer>
    Minimum number of instances required in synchronous replication with the
    primary. Undefined or 0 allow writes to complete when no standby is
    available.
```

云原生拓展

我真的很喜欢导入数据库的功能。当您拥有一次性环境（例如开发）并且您希望使用所需的所有数据快速创建新环境时，这非常有用。

kubectl 的 `cnpg` 插件将所有这些强大的功能结合在一起，允许您执行备份、隔离和滚动重启等操作。

```
~$ kubectl cnpg backup -n pg001 pg001
~$ kubectl get backup/pg001-20230511084800 -n pg001
NAME                                AGE    CLUSTER  PHASE    ERROR
pg001-20230511084800              1m     pg001    completed
```

集群运行后，出现问题时需要收到警报。这意味着建立某种监控。我一直使用 `postgres_exporter` 来实现此目的，但令人高兴的是，CloudNativePG 附带了所有唱歌和跳舞配置，您不需要任何额外的配置。

该项目提供了一个做得很好的 Grafana 仪表板和一些示例 Prometheus 规则，这是您入门所需的一切。

缺少什么

我只发现缺少一项我想要的功能，尽管这不是一个大问题。Operator 不具备创建入口端点以允许从外部 K8s 访问数据库的功能。

例如，如果您有两个 Kubernetes 集群，一个运行主数据库，另一个运行副本，则这是必需的。

该文档(https://cloudnative-pg.io/documentation/1.18/expose_pg_services/)提供了一些关于如何使用 Nginx Ingress 实现此目的的线索。我已经使用 Traefik 和 SNI 来完成此操作，类似于我使用 Kafka 所做的事情，并且效果很好。

如果这是像 Strimzi 那样属于 Operator 的一部分那就太好了。

总结

我在完成部署之前进行了基本的 OAT（操作验收测试），类似于我们为客户所做的测试，并且它通过了所有测试。

总的来说，它看起来是一个功能强大但易于使用的 Operator。我印象深刻。