KubeDB

2020.10.24

AppsCode Inc.
https://appscode.com
Email: sales@appscode.com

Table Of Contents

At a Glance

KubeDB by AppsCode is a production grade cloud-native database management solution for Kubernetes. KubeDB simplifies and automates routine database tasks such as provisioning, patching, backup, recovery, failure detection, and repair for various popular databases on private and public clouds. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

KubeDB makes it easy to use replication to enhance availability and reliability for production workloads. Using the Multi-AZ deployment option, you can run mission-critical workloads with high availability and built-in automated fail-over from your primary database to a replicated secondary database. Using Read Replicas, you can scale out beyond the capacity of a single database deployment for read-heavy database workloads.

KubeDB provides you with seven familiar database engines to choose from, including PostgreSQL, MySQL, MongoDB, Elasticsearch, Redis, Memcached, and Percona XtraDB. This means that the code, applications, and tools you already use today with your existing databases can be used with KubeDB.

Problem

Kubernetes (K8s) has emerged as the de-facto orchestrator for automating deployment, scaling, and management of modern containerized applications. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community. Enterprises both small and large are adopting Kubernetes to institute a single control plane for their IT governance across public and private clouds.

With Kubernetes, it is relatively easy to manage and scale stateless applications like web apps, microservices, and mobile backends. The basic Kubernetes APIs, like Deployments, can scale and recover from failures without additional knowledge. But the dynamic compute environment of Kubernetes makes it challenging to manage stateful applications, like databases and caches. These systems require application domain knowledge to correctly provision, scale, upgrade, and reconfigure while protecting against data loss or unavailability.

The conventional wisdom is to apply one of the following solutions to this problem: (1) use cloud provider managed database services, (2) use third party managed database-as-a-service (DBaaS) solutions usually provided by database vendors themselves, or (3) run databases on legacy infrastructure. According to our user survey from Feb 2019, 31% users were using cloud providers managed offering, 43% was self-hosting, 45% was using Helm to manage databases on Kubernetes. All of these options suffer from one or more problems as described below:

- According to Gartner research only 3% of IT spend happens on public cloud. For the vast majority of IT infrastructure, managed database-as-a-service solutions are not an option.

- All of these options mean DevOps teams have to maintain one stack for stateless workloads and a separate stack for stateful workloads. This significantly increases the operational complexity. This results in loss of visibility across the stack and the agility gained by adopting Kubernetes is drastically reduced.

- Not all cloud providers have the choice of database engines in their managed service offerings that are desired by modern app developers.

- Since these options run databases outside a Kubernetes cluster, they introduce additional latency on database connections.

- Third party DBaaS solutions usually support a few cloud providers and even then are not available on all regions of those cloud providers.

- DBaaS providers charge a pretty penny for their managed DBaaS solutions. This can get expensive if each developer on the team uses their own instance for development and testing.

- To deliver a managed service experience, some cloud providers (eg: Amazon RDS) don't provide shell access to DB instances, and restrict access to certain system procedures and tables that require advanced privileges.

Solution

Kubernetes API server has a unique feature called WATCH (similar to push notifications in smartphones). This enables a new type of application called “Operators” that can mimic the actions of a human operator in response to user requests or system events. An operator can encode application-specific operational knowledge into software leveraging this powerful Kubernetes abstraction. An operator extends the Kubernetes API through custom resources, enabling users to provision, scale and manage applications.

KubeDB is a collection of Kubernetes operators with the operational knowledge of a DBA programmed into it. Each operator builds upon the basic Kubernetes custom resource and controller concepts. We have baked into each operator the knowledge and intelligence of a human DBA to manage common database administration tasks. Leveraging operator pattern, KubeDB delivers a “managed” database service on Kubernetes for hybrid and multi-cloud environments that is on par with the Industry leading database services for free. All of this is accomplished using standard Kubernetes tools, CLI and API.

Key Benefits

- KubeDB simplifies many of the difficult or tedious management tasks of running production grade databases on private and public clouds. Maintain one stack for all your stateless and stateful applications and simplify the operational complexity.

- Standard Kubernetes is all you need. If you can run Kubernetes, you can provision and manage databases using KubeDB. Use standard Kubernetes CLI and API to provision and manage databases.

- KubeDB shortens time-to-value. Create a single YAML file representing your desired database specification and you are in business. With KubeDB deploy any popular database engines in minutes, so your team can focus on what matters most — velocity.

- KubeDB utilizes the Kubernetes constructs for compute, memory and storage so that you can scale them independently on private and public cloud. If you need more CPU, less IOPS, or more storage, you can easily allocate them.

- KubeDB manages backups, automatic failure detection, and recovery.

- You can have automated backups performed when you need them, or manually create your own backup snapshot. Backup snapshots are deduplicated, encrypted and stored in various cloud stores, eg, S3, GCS, Azure Blobstore etc. You can use these backups to restore a database. The KubeDB restore process works reliably and efficiently.

- You can get high availability with a primary instance and one or more secondary instances that you can fail over to when problems occur.

- KubeDB supports a wide variety of database engines: PostgreSQL, MySQL, MongoDB, Elasticsearch, Redis, Memcached, and Percona XtraDB server. Unlike managed DBaaS, you are not restricted to what your cloud provider supports.

- KubeDB runs your database instance in the private Kubernetes pod network by default. This protects your database from accidental access from outside and allows low latency access from applications running inside the cluster.

- KubeDB applies pod security policies to secure your database instances. Additionally, you can help control who can access your database instances by using Kubernetes Role Based Access Control (RBAC) to define users and permissions. KubeDB optionally integrates with HashiCorp Vault so that you can provision restricted database user accounts and rotate their credentials periodically. You can also help protect your databases by applying network policies that restrict unwanted access from pods running in the cluster..

- KubeDB has native support for monitoring via Prometheus. You can use the builtin Prometheus scrapper or Prometheus Operator to monitor KubeDB managed databases.

- Configurable termination policy to protect against accidental deletion of production databases by CLI or API.

- Keeps track of deleted databases, wipeouts properly backup snapshots on deletion with a single command to avoid billing surprises at the end of the month.

- KubeDB is free to use and 100% open source. You only pay for the resources used to your infrastructure provider.

Features

Lower administrative burden

Unleash developer velocity

With KubeDB, you can launch a database instance via a simple CLI, management console or API and start developing in minutes. KubeDB eliminates lengthy deployment and management processes with on-demand provisioning, scaling, patching and updating database instances and shortens time-to-value. KubeDB gives each developer on your team their own development instance, so your team can focus on what matters most — velocity and not have to worry about stepping on each other's toes.

Freedom of choice of Database engines

Build and deploy on the cloud or on-premises faster because KubeDB offers seven familiar database engines to choose from, including PostgreSQL, MySQL, MongoDB, Elasticsearch, Redis, Memcached, and Percona XtraDB. This means that the code, applications, and tools you already use today with your existing databases can be used with KubeDB. KubeDB provides users the option to build their custom database Docker images from official KubeDB images and bundle any plugin they prefer.

Native Kubernetes Support

Infrastructure as code

KubeDB defines each supported database engine as a custom resource (CRD) for Kubernetes. You can use Kubernetes CLI or API to provision, manage, and interact with KubeDB database instances. This native integration with Kubernetes means your database instances are ready for any higher level deployment tooling like GitOps, Open Policy Agent (OPA), etc.

Hybrid and Multi-cloud

To run KubeDB, all you need is a Kubernetes cluster. So, you can run a production-grade database deployment in your desktop via minikube to hybrid and multi-cloud environments against a single consistent interface. KubeDB follows the official deprecation policy for Kubernetes so that your database instances always keep pace with the latest security and maintenance updates.

Single Stack for Stateless and Stateful Workloads

KubeDB enables enterprises to maintain one stack for their stateless and stateful workloads. This means your developer and operations team don't have to learn, maintain and manage separate stack for deployments, monitoring, logging and alert management. KubeDB allows you to use the same workflow across public and private clouds and avoid vendor lock-in. KubeDB makes your engineering teams agile so that they can focus on delivering the best value for your customers and users.

Performance

Cloud agnostic Storage

Kubernetes provides a powerful cloud agnostic storage abstraction that enables Kubernetes workloads to use a wide variety of block and file storage to persist data. KubeDB uses Persistent Volume Claims (PVC) to dynamically provision disks for database instances. Using appropriately defined StorageClasses, KubeDB provisioned database instances are designed to scale from small development workloads up to performance-intensive workloads on private and public cloud environments.

Provisioned IOPS (SSD) Storage

Cloud providers like AWS provide provisioned IOPS Storage which is an SSD-backed storage option designed to deliver fast, predictable, and consistent I/O performance. You specify an IOPS rate when creating a StorageClass, and KubeDB provisions disks at that IOPS rate for the lifetime of the database instance. This storage type is optimized for I/O-intensive transactional (OLTP) database workloads.

Cloud-Native Storage

There is a growing ecosystem of cloud-native storage solutions like Rook (Ceph), OpenEBS, Portworx that provide network attached storage for containerized workloads. These systems can provide block storage via CSI drivers in cloud or bare metal clusters. KubeDB can provision database instances that utilize these storage systems using standard Kubernetes abstractions.

Local Persistent Volume

Most Kubernetes Storage systems enable remote storage which persists data independent of the Kubernetes node where the data originated. Remote storage usually can not offer the consistent high performance guarantees of local directly-attached storage. Using the Local Persistent Volume plugin, KubeDB can consume high performance local storage seamlessly. However, local volumes are still subject to the availability of the underlying node. HA database instances using local volumes are able to tolerate this reduced availability, as well as automatically recover from potential data loss, depending on the durability characteristics of the underlying disk.

Low Latency

KubeDB runs your database instances inside the same pod network as the workloads that use these database instances. Unlike database instances managed by DBaaS vendors, there is no cross VPC latency between your database instances and your applications.

Scalability

Easy storage scaling

As your storage requirements grow, you can also provision additional storage. KubeDB takes advantage of the Kubernetes CSI drivers to automatically grow the size of your database volume as your database storage needs grow, up to the maximum allowed by your infrastructure provider. Storage scaling is on-the-fly with zero downtime.

Read Replicas

Read Replicas make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can create one or more replicas of a given source DB instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput. Read replicas are available in KubeDB managed MySQL, PostgreSQL, and MongoDB.

Connection pooling

KubeDB can provision out-of-process connection pools like PgBouncer and ProxySQL for your database instances. Opening a database connection is expensive; you have to open up network sessions, authenticate, have authorization checked, and so on. Pooling keeps the connections active so that, when a connection is later requested, one of the active ones is used in preference to having to create another one. This can be particularly useful for serverless workloads.

Availability and durability

Automated backups

KubeDB will backup your database and transaction logs at a user defined frequency and store both for a user-specified retention period in a cloud object store (S3, GCS, etc.) or local filesystem (like NFS, etc.). You can also initiate one-off backups. KubeDB uses Stash by AppsCode to take deduplicated and encrypted backups so that you only incur the cost of incremental storage use. You can create a new instance from database snapshots whenever you desire.

Automated Failover and self-healing

KubeDB will automatically replace the pods powering your database instances in the event of an involuntary disruption eg, hardware failure). In case of a voluntary disruption (eg. draining a node for repair or upgrade, draining a node from a cluster to scale the cluster down, etc.), PodDisruptionBudget for clustered database instances is used to make maintenance of underlying infrastructure transparent. KubeDB configures liveness and readiness probes for database instances for continuous health-checking and automatically fails over if an instance is not healthy.

Multi-AZ deployments

KubeDB Multi-AZ deployments provide enhanced availability and durability for database instances, making them a natural fit for production database workloads. When you provision a Multi-AZ database instance, KubeDB replicates your data to standby instances across multiple Kubernetes failure-domains. KubeDB enables enterprises to target database instances into a placement zone to meet particular data affinity, governance and performance requirements.

Security

Encryption at rest and in transit

KubeDB allows you to use encrypted storage for your databases using keys you manage through your cloud provider's key management service. A database instance running on AWS can keep its data stored at rest in the underlying storage encrypted, as are its automated backups, read replicas, and snapshots.

Network isolation

KubeDB provisioned database instances by default runs inside the private Kubernetes pod network. To further secure your database instances, you can configure network policies so that only the necessary pods can communicate with database pods. It creates firewalls between pods running on a Kubernetes cluster.

Resource-level permissions

KubeDB is integrated with Kubernetes Role Based Access Control (RBAC) and provides you the ability to control the actions that your Kubernetes users and groups can take on specific KubeDB database instances. For example, you can configure your RBAC roles to ensure developers are able to modify "Development" database instances, but only Database Administrators can make changes to "Production" database instances.

Private registry & air-gapped cluster

KubeDB operator can be configured to work with an in-cluster private registry. This makes KubeDB viable to run a air-gappeed Kubernetes cluster. Cluster administrators can use private registries to ensure that only pre-approved docker images are able to run inside a cluster.

Database User management

KubeDB by default creates the root user account for databases that support it like PostgreSQL, MySQL, etc. Optionally, you can provision additional database users with custom permissions and rotate their credentials using HashiCorp Vault for PostgreSQL, MySQL and MongoDB. Database secret engines in Vault generate database credentials dynamically based on configured roles. Using KubeVault, you can configure a secret engine, create roles and issue credentials from Vault. You can request credentials and after it's been approved by the database administrator, the Vault operator will create a Kubernetes Secret containing the credential and also creates RBAC Role and RoleBinding so that the user can access the Secret.

Manageability

Monitoring and metrics

KubeDB comes with native support for monitoring via Prometheus. You can use the builtin Prometheus scrapper or Prometheus Operator to monitor KubeDB supported databases as well as KubeDB operator itself. You can use the Grafana to view key operational metrics, including compute/memory/storage capacity utilization, I/O activity, and instance connections. You can also use any metrics solutions like Datadog with KubeDB.

Event notifications

Prometheus Alertmanager can send alerts based on the Prometheus metric exposed by KubeDB managed database instances. It takes care of deduplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie. It also takes care of silencing and inhibition of alerts.

Inexpensive / Cost-effectiveness

Reserved instances

Cloud providers like AWS and Azure have the option to reserve virtual machines in advance for 1-3 year terms and provide a significant discount (up to 75% - 80%) compared to on-demand VM pricing. KubeDB operators can provision database instances on reserved instances and significantly reduce your TOC for the DB instances.

Halt and Resume

KubeDB allows you to easily halt and resume your database instances. This makes it easy and affordable to use databases for development and test purposes, where the database is not required to be running all of the time.