Build a Low-Cost, High-Performance Analytic Platform with Kubernetes and Open Source*

Robert Hodges
Altinity

*And ClickHouse, too!
A brief message from our sponsor...

**Robert Hodges**
Database geek with 30+ years on DBMS. Kubernaut since 2018. Day job: Altinity CEO

**Altinity Engineering**
Database geeks with centuries of experience in DBMS and applications

ClickHouse support and services: [Altinity.Cloud](#) and [Altinity Stable Builds](#)
Authors of [Altinity Kubernetes Operator for ClickHouse](#)
How do cloud analytic databases work?

And what are some of the tradeoffs?
Snowflake database architecture

Cloud Services
- Authorization & Access Control
- Metadata Management (FoundationDB)
- Infra Mgmt
- Optimizer
- Xact Mgr
- Security

Query Processing
- Cache

Storage
- Data in object storage

Multi-tenant cloud services enabling secure, efficient ops
Virtual warehouse with dedicated compute
Data in object storage
What’s great about Snowflake?

✔ General purpose
✔ Serverless operation
✔ Handles large numbers of tenants with completely different applications
✔ Standards-compliant SQL
  ○ Complete implementation with ACID transactions
  ○ Sophisticated query optimizer
  ○ Efficient columnar storage with self-tuning partitioning and compression
  ○ Big table joins
✔ UI with built-in SQL editing and management
What Snowflake does not do

✘ Keep data in customer cloud account
✘ Minimize costs, especially for 24x7 analytics
✘ Deliver stable real-time response
✘ Handle SaaS user-facing analytics
✘ No vendor lock-in
Let’s design an analytic service with open source
Focus on a specific problem

Deliver a GDPR-compliant replacement for Google Analytics

Analytic Platform

- Data visualization and consumption
- Data pipelines and service integration
- Analytic data storage and query processing

Source Event Data

Visualization APIs
Extracts
Alerting
First, scope the requirements

**Snowflake strengths**

- ✗ General purpose
- ✔ Serverless operation
- ✗ Handle wide range of applications
- ✗ Standards-compliant SQL
- ✔ UI with SQL editing & management

**Snowflake weaknesses**

- ✔ Keep data in your own cloud account
- ✔ Minimize costs for 24x7 systems
- ✔ Deliver stable real-time response
- ✔ Handle SaaS user-facing analytics
- ✔ No vendor lock-in

© 2023 Altinity, Inc.
Second: pick an open source analytic database

- **OpenSearch**
  - Apache 2.0
  - Full-text search, log analytics

- **ClickHouse**
  - Apache 2.0
  - Web analytics, network management, real-time bidding, financial asset valuation, security event & incident management, …

- **Presto**
  - Apache 2.0
  - Federated query on data lakes and DBMS
  - Enterprise analytics on large volumes of data across disparate sources
ClickHouse database architecture

- Vectorized, parallel query engine
- Columnar data in block storage
- Columnar data and data lake files
- S3-Compatible Object Storage
- Replication, sharding, distributed query
- ZooKeeper
- Cluster consensus
Third: lay out the analytic platform logical design

- ClickHouse
  - Analytic DB
- CloudBeaver
  - SQL Editing
- Grafana
  - Dashboards
- ZooKeeper
  - Cluster Consensus
- Prometheus
  - Operational Metrics
Implementing on Kubernetes with ArgoCD
Kubernetes orchestrates container-based applications

Logical Design
- ClickHouse Server
- Block Storage

Kubernetes Resource Manifests
- Stateful Set
- Pod
- Persistent Volume Claim
- Persistent Volume

Mapping
- Physical Infrastructure
- Process running on host
- AWS EBS Storage

Mapping
Map the logical design to Kubernetes resources

- **ClickHouse**: Analytic DB
  - Install using Altinity ClickHouse operator

- **CloudBeaver**: SQL Editing
  - Install using manifest

- **Grafana**: Dashboards
  - Install using manifest

- **ZooKeeper**: Cluster Consensus
  - Install using manifest

- **Altinity Operator for ClickHouse**: Operational Metrics
  - Install using community Helm chart

- **Prometheus**: Operational Metrics
  - Install using community Helm chart

© 2023 Altinity, Inc.
How can we deploy the stack in a cloud native way?

Working Kubernetes Cluster

ClickHouse Resources
ZooKeeper Resources
Prometheus Resources

Definitions in Git

ClickHouse
Prometheus
Cluster Consensus
Analytic DB
Operational Metrics
ArgoCD maps deployments from GitHub to K8s

- Kubernetes Manifest
- Kustomize + Manifests
- Helm Chart

ArgoCD

Kubernetes
- App Resources
- App Resources
- App Resources

© 2023 Altinity, Inc.
Basic GitOps using GitHub, ArgoCD, and Kubernetes

- **Git Push**
  - https://github.com/yourco/modernstack.git
  - ClickHouse application
  - ZooKeeper application
  - Prometheus application

- **ArgoCD**
  - argocd namespace
  - argocd app create
  - argocd app sync
  - argocd app delete

- **Local k8s Cluster**
  - API
  - Repository Service
  - Application Controller
  - ch namespace

- **Prod k8s Cluster**

© 2023 Altinity, Inc.
Life cycle for ArgoCD applications

Map GitHub state to Kubernetes cluster

```
argocd app create
```

Apply state GitHub to Kubernetes

```
argocd app sync
```

Remove resources and delete mapping

```
argocd app remove
```
Managing Kubernetes applications with ArgoCD

DEMO TIME!
Wiring and dependencies in the stack

Applications

Data Platform

CloudBeaver → ZooKeeper

ClickHouse

ClickHouse Operator

Grafana

Prometheus

Forwarding, peering, or VPN

© 2023 Altinity, Inc.
ArgoCD Assessment

**Strengths**
- Enables infrastructure as code - your configuration lives in Git
- Can map configuration to multiple environments
- Very adaptable - you can usually get things to install
- Exchange components to evolve the stack

**Weaknesses**
- Have to understand Kubernetes to understand ArgoCD
- Not all features are mature
- Full GitOps automation is complex
- Does not handle deployment outside of Kubernetes
Getting to a production analytic stack
Buy vs. build, aka “Pick your battles”

- **Application**
  - CloudBeaver
  - Grafana
  - Prometheus
  - ClickHouse

- **EKS & GKE (Managed Kubernetes)**
  - Usually outsourced

- **Application**
  - Grafana
  - CloudBeaver
  - Altinity. Cloud
  - Prometheus

- **EKS & GKE (Managed Kubernetes)**
  - Rarely outsourced

**Build or Buy**
Kubernetes enables more flexible managed services

Altinity.Cloud Manager ("ACM")

Tenant A Environment

Tenant B Environment

Anywhere Environment

Altinity Connector

Google GKE

Amazon EKS

User VPC

User Kubernetes

© 2023 Altinity, Inc.
Security in analytic platforms requires work

- **Client App**
- **kubernetes**
- **DBMS Pod**
- **Service**
- **ConfigMap**
- **Secret**
- **Operator**

**TLS encrypted connection**

- **Source IP whitelisting**
- **Service configures private cloud endpoint**

- **Insecure ports locked down**

- **X509 / private key inserted and TLS configured**

- **Credentials and private keys passed via Secrets**
Look for operators and hardening guides for components

```yaml
apiVersion: "clickhouse.altinity.com/v1"
kind: "ClickHouseInstallation"
metadata:
  name: "prod"
spec:
  templates:
    serviceTemplates:
      - generateName: clickhouse-{chi}
        metadata:
          annotations:
            service.beta.kubernetes.io/aws-load-balancer-internal: "true"
        name: default-service-template
        spec:
          ports:
            - name: https
              port: 8443
            - name: secureclient
              port: 9440
          type: LoadBalancer
```

Vendor specific config for internal load balancer without public IP address

Only permit secure protocols
More tasks to deploy the analytic stack

- What other services do you need?
  - Airflow, Flink, Spark, …
- Adding hooks to synchronize Git fully with ArgoCD
- Building a dev/staging/prod pipeline
  - Or blue/green deployments
- Capacity planning and performance scaling
- Backup
- Monitoring

And of course, building your applications.
Final notes and more to come
Tips for building your own analytics platform

1. Open source stacks beat proprietary services for specific problems
2. Keep the problem small
3. Kubernetes offers state-of-the-art platform for constructing the stack
4. ArgoCD maps Git state flexibly to Kubernetes resources
   a. Papers over installation differences
   b. Enables infrastructure as code for the entire stack
5. Production systems require expertise and careful design
6. Outside Kubernetes you need other options: Terraform or Ansible
How to get started with the example application

`git clone https://github.com/Altinity/argocd-examples-clickhouse`
Projects that went into the stack

- Altinity Projects
  - ArgoCD Examples
  - Altinity Kubernetes Operator for ClickHouse
  - Altinity Stable Builds for ClickHouse
- The rest of the stack
  - ClickHouse: [https://github.com/ClickHouse/ClickHouse](https://github.com/ClickHouse/ClickHouse)
  - Prometheus: [https://github.com/prometheus-community/helm-charts](https://github.com/prometheus-community/helm-charts)
  - Grafana: [https://github.com/grafana/grafana](https://github.com/grafana/grafana)
  - CloudBeaver: [https://github.com/dbeaver/cloudbeaver](https://github.com/dbeaver/cloudbeaver)
Thank you and good luck!

Any Questions?

Robert Hodges
https://altinity.com

Altinity.Cloud
Altinity Stable Builds for ClickHouse
Altinity Kubernetes Operator for ClickHouse