# **MongoDB Backup Architecture - Design Summary**

## **📌 Overview**

This document outlines the design decisions and architecture used to implement a scheduled MongoDB backup system in a Kubernetes environment, leveraging Azure infrastructure and GitOps practices.

## **📂 Components**

### **1. MongoDB Instance**

* **Host**: Ubuntu 18.04 VM
* **Location**: Azure VM provisioned via Terraform
* **Version**: MongoDB 4.4
* **Access**: Secured using SSH with a private RSA key

### **2. Backup Application**

* **Container Image**: cvp01/mongo-backup:latest
* **Language**: Python with paramiko for SSH and Azure SDK for Blob Storage
* **Functionality**:
  + Connects to MongoDB via SSH
  + Dumps the database
  + Compresses and uploads the dump to Azure Blob Storage

### **3. Kubernetes Cluster**

* **Platform**: Azure Kubernetes Service (AKS)
* **Deployment**: Managed via Terraform
* **Namespace**: backup

### **4. CronJob in Kubernetes**

* **Resource**: CronJob
* **Frequency**: Every hour (0 \* \* \* \*)
* **RestartPolicy**: OnFailure
* **Trigger**: Also supports manual Job trigger for testing

### **5. Secrets Management**

* **Kubernetes Secret**: backup-secrets
* **Injected into Pod**:
  + blob-conn: Azure Blob Storage connection string
  + id\_rsa: Base64-encoded private RSA key used by paramiko

### **6. Terraform Infrastructure**

* Manages:
  + Azure Resource Group
  + AKS Cluster
  + Azure VM
  + Azure Blob Storage
  + K8s Secret and Namespace

## **🔧 Design Decisions & Tradeoffs**

| **Choice** | **Justification** | **Alternative** | **Tradeoff** |
| --- | --- | --- | --- |
| **Use of CronJob** | Built-in periodic execution in K8s, supports logging, retry, manual triggering | GitHub Actions / external scheduler | CronJobs are internal to K8s, no external trigger unless exposed |
| **Secrets via K8s Secret** | Secured and scoped within the backup namespace | ConfigMap (not secure), external vault | Simpler than full Vault setup, but less auditable |
| **Terraform** | Idempotent, declarative setup of entire stack | Manual setup or shell scripts | Requires Terraform knowledge and CI discipline |
| **Private Key Mount** | Container mounts key at runtime from secret | Bake key into container (bad) | Separation of secrets from container image improves security |
| **Python + Paramiko** | Simple SSH connection with file transfer | rsync, scp, or agent forwarding | Python gives more control but adds image size & dependency mgmt |

## **🚀 Deployment Process**

1. Run terraform apply to deploy full environment.
2. Get MongoDB VM IP and update the mongo-backup-cronjob.yaml.
3. Apply the CronJob: kubectl apply -f mongo-backup-cronjob.yaml
4. Trigger test: kubectl create job --from=cronjob/mongo-backup mongo-backup-manual -n backup

## **🛡 Security Considerations**

* RSA key is stored in a K8s Secret with read-only mount
* Blob Storage credentials stored securely in same Secret
* Pod has no elevated permissions or host mounts

## **📈 Observability & Maintenance**

* Logs are accessible via kubectl logs
* Failures trigger CrashLoopBackOff and show meaningful error traces
* Rotation and pruning of old backups can be handled via Azure Lifecycle Management or cron in Blob Storage

## **🧭 Future Improvements**

* Replace Secret with HashiCorp Vault or Azure Key Vault
* Add backup integrity verification step
* Notify via email/Slack on backup success/failure
* Schedule daily backup in addition to hourly delta
* Auto-discover MongoDB IP via Terraform output or DNS