VOTING-Services

Data Protection (en)

Author	Abraxas Informatik AG
Classification	public
Version	0.9
Date	19. Aug 2022

For digital Switzerland. For Sure.

abraxas

Table of contents

1. Overview

This chapter covers how data protection is covered across our VOTING services:

- VOTING Basis
- VOTING Ausmittlung

The following diagram illustrates communication ways and where data is stored and backed up:

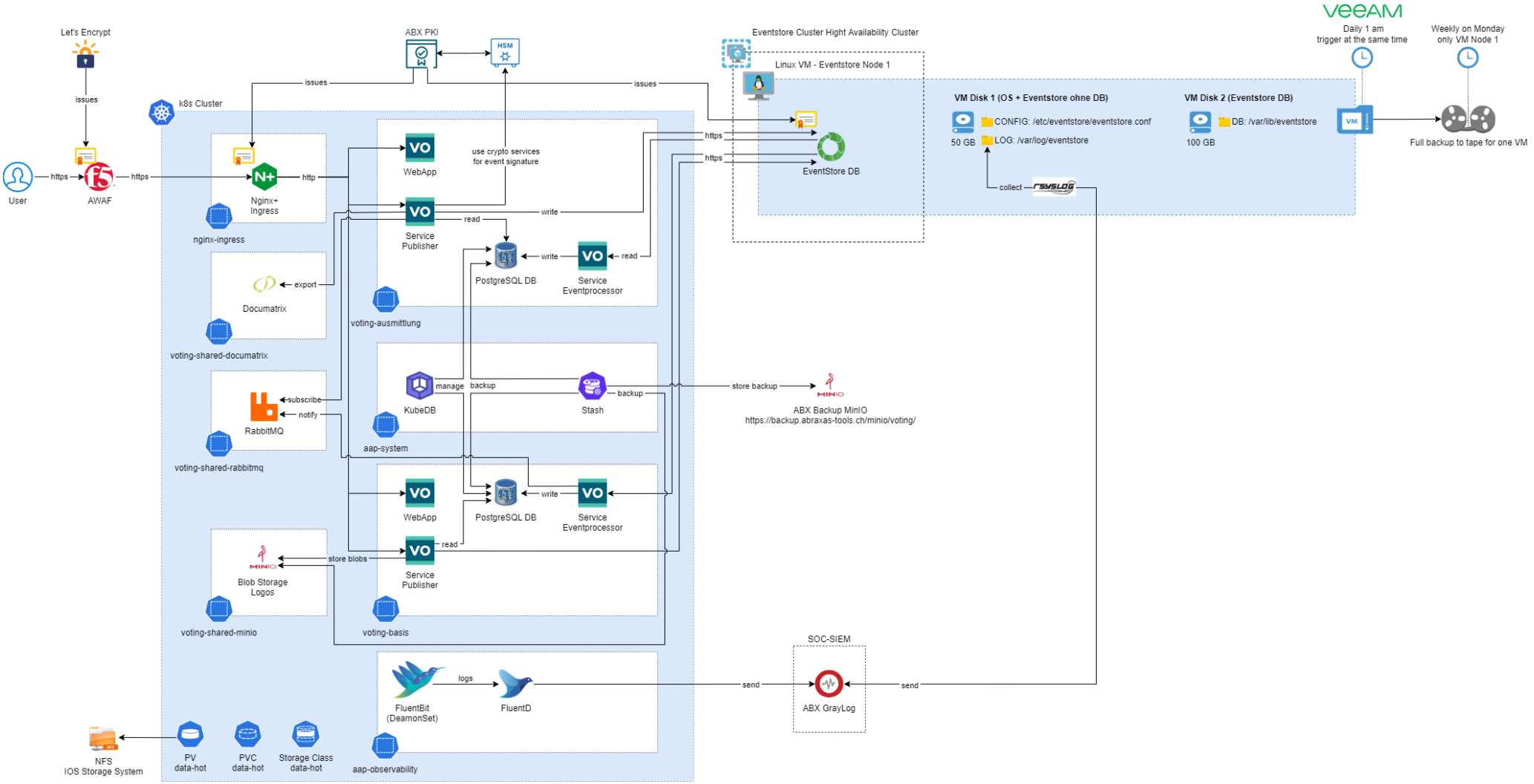

Figure 1 Communication ways

2. Reliability

To ensure data reliability the following measures are taken.

2.1 Event Signature

Within the event sourcing architecture there is an additional security layer implemented which signs all events with cryptographic methods to ensure data integrity. For more information see the VOTING Ausmittlung – Event Signature (en), which will be available as soon as it is added to the scope.

2.2 Data Encryption

The services use industry-accepted encryption products to protect data and communications during transmission between a client's network and the services, i.e. HSTS-Enforcement, Certificate-Pinning etc.

2.2.1 Data at rest

2.2.1.1 AAP - Kubernetes Cluster

The full AAP storage concept is defined in the internal document "Storage Concept".
Services that store data and run within the k8s cluster (i.e. PostgreSQL DB) are able to use one of the following storage classes to provision their persistent volume claim:

- data-hot
- logs

These classes are bound to a NFS provisioner which stores data on the Network File Share provided by the IOS Storage System.
At the moment there is no encryption in place for data at rest on the NFS Share, but it is on the roadmap.

2.2.1.2 VMs - EventStore

Currently, there is no data at rest encryption for VMs/disks in use, but it is on the roadmap. VMware offers support for data at rest encryption, but therefore a key management server is required, which is not in place for the VOTING VMware infrastructure right now.

3. Backup

3.1 EventStore DB Backup

Since the VOTING EventStore DB infrastructure is virtualized on Linux rhel 8 VM's, the database is backed up with disk snapshots using Veeam as recommended by EventStore backup and restore procedures.
The chosen backup service for the Veeam backup is "Premium - mirrored", which means it is geo-replicated in two Abraxas data centers (Glattbrugg und Lupfig).
Additionally every week one Veeam backup is getting outsourced to the Abraxas Tape Storage.

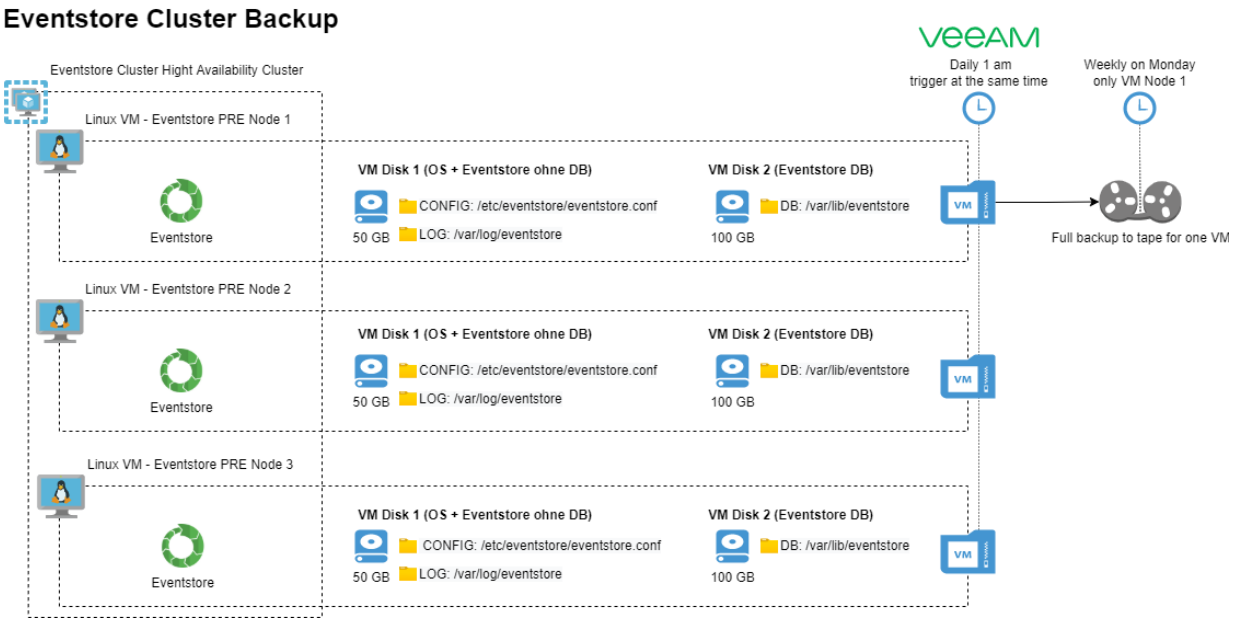

Figure 2 Eventstore Cluster Backup

3.1.1 Backup Scheduling and Retention

Since every appended event is persisted in EventStore DB every historical state can be replayed. Therefore the main purpose of the EventStore DB backup is for disaster recovery if the DB gets corrupted for some reason. That's why retention policies are set short as follows:

- Daily VEEAM Backup at 1 am per VM including all disks, max age: 30 days
- Weekly tape backup on Monday

3.2 PostgreSQL Database Backup

Any PostgreSQL Database within the VOTING context is getting backed up using KubeDB in combination with Stash.
Stash utilizes restic to encrypt and securely backup the databases. The database backup is stored within the Abraxas Backup S3 Storage which is based on MinIO.
The Storage Secret is provisioned by our custom ArgoCD secrets-plugin …

```
data:
  admin.enabled: "false"
  accounts.gopasssync: apiKey
  accounts.gopasssync.enabled: "true"
  configManagementPlugins: |
    - name: secrets-plugin
      generate:
        command: [bash, -c]
        args: ['gopass sync &> /dev/null && kustomize build && echo "---" && bash
./secrets.yaml.sh']
```

Codeblock 1 ArgoCD Secrets Plugin

... which inserts the secrets from the service specific gopass secret repository and applies it as k8s Secret CRD

```
AWS_ACCESS_KEY_ID=$(gopass show voting-shared/shared/stash/aws-access-key-id
2>/dev/null)
AWS_SECRET_ACCESS_KEY=$(gopass show voting-shared/shared/stash/aws-secret-access-
key 2>/dev/null)
RESTIC_PASSWORD=$(gopass show voting-shared/shared/stash/restic-password
2>/dev/null)

cat <<EOL
apiVersion: v1
kind: Secret
metadata:
  labels:
    part-of: voting
    app: voting-servicename
    app/component: database
    env: $env
  name: voting-servicename-database-backup
  namespace: $env-voting-servicename
type: Opaque
stringData:
  AWS_ACCESS_KEY_ID: "$AWS_ACCESS_KEY_ID"
  AWS_SECRET_ACCESS_KEY: "$AWS_SECRET_ACCESS_KEY"
  RESTIC_PASSWORD: "$RESTIC_PASSWORD"
EOL
```

Codeblock 2 secrets.yaml.sh

How Stash backups work, see

- https://stash.run/docs/v2022.05.18/addons/postgres/overview/
- https://kubedb.com/docs/v2022.05.24/guides/postgres/backup/overview/

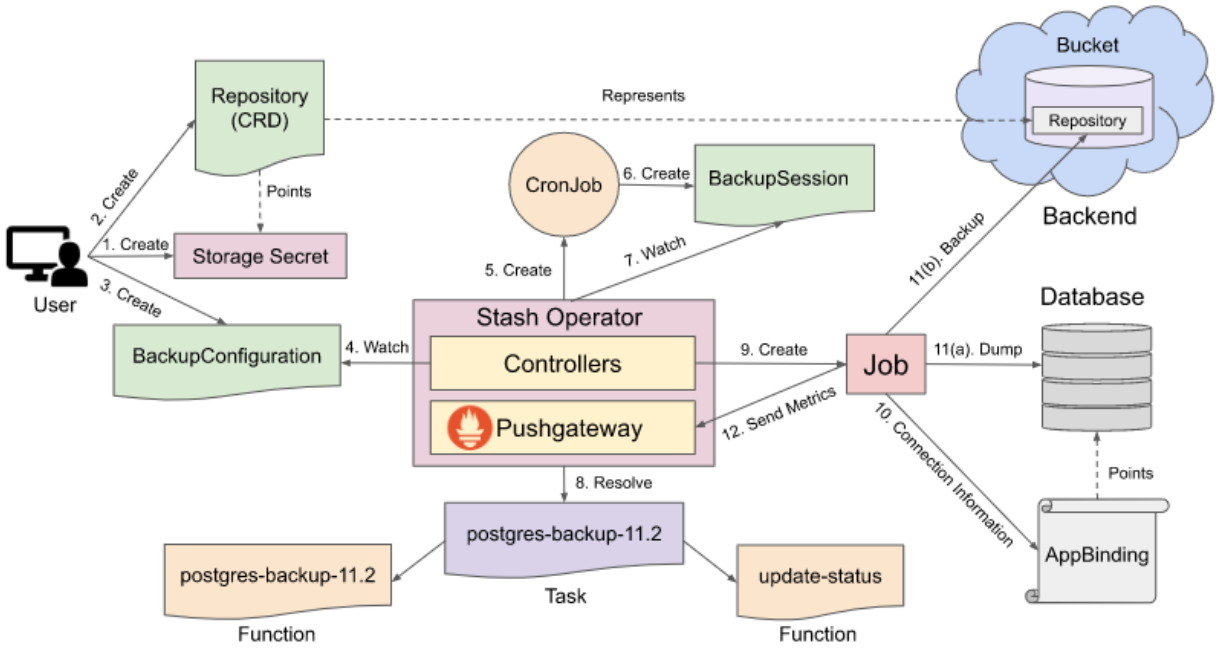

Figure 3 PostgreSQL Backup Overview

3.2.1 Backup Scheduling and Retention

PostgreSQL DB's on Production are backed up every full hour according to the Stash BackupConfiguration which is defined as follows:

```
apiVersion: stash.appscode.com/v1beta1
kind: BackupConfiguration
metadata:
  name: voting-product-database-backup
spec:
  schedule: "0 * * * *"
  task:
    name: postgres-backup-13.1
  repository:
    name: voting-product-database-backup-repository
  target:
    ref:
      apiVersion: appcatalog.appscode.com/v1alpha1
      kind: AppBinding
      name: voting-product-database
  retentionPolicy:
    name: 'abx-default'
    keepDaily: 6
    keepWeekly: 3
    keepMonthly: 11
    keepYearly: 1
    prune: true
```

Codeblock 3 Backup Configuratoin

4. Logs

All applications log information to their respective system log facility or a centralised GrayLog server in order to enable security reviews and analysis.