EVERTRUST

Cloud

2025-11-05

Table of Contents

Chapter 1. Overview

EVERTRUST Cloud is a fully-managed platform designed to remove the burden of deploying and operating EVERTRUST products, by offering them to customers as a service.

Concepts

EVERTRUST Cloud orchestration relies on the EVERTRUST Control Plane, which holds logic regarding organizations, workspaces and instances:

- an **Organization** uniquely identifies you as a customer, manages some resources such as backup encryption keys and multiple *Workspaces*
- a **Workspace** is a group of *Instances* that is linked to an *Environment*. A typical deployment would be comprised of a staging and a production workspace, but there's not limit on how many workspaces can be created for an organization.
- an **Instance** is a product deployed in a *Workspace*. It has a logical role such as a PKI or a CLM and can communicate with other instances in the same *Workspace*.
- an **Environment** is a group of cloud resources, such as a cloud provider and a region, where instances can be deployed. An environment can be shared among multiple customers, or for performance/compliance reasons, be dedicated to a single customer.

Technical overview

EVERTRUST Cloud is built upon battle-tested technologies to deliver outstanding reliability and performance. A component-based approach allows us to scale and improve the service continuously while limiting responsibility of each component to its minimum.

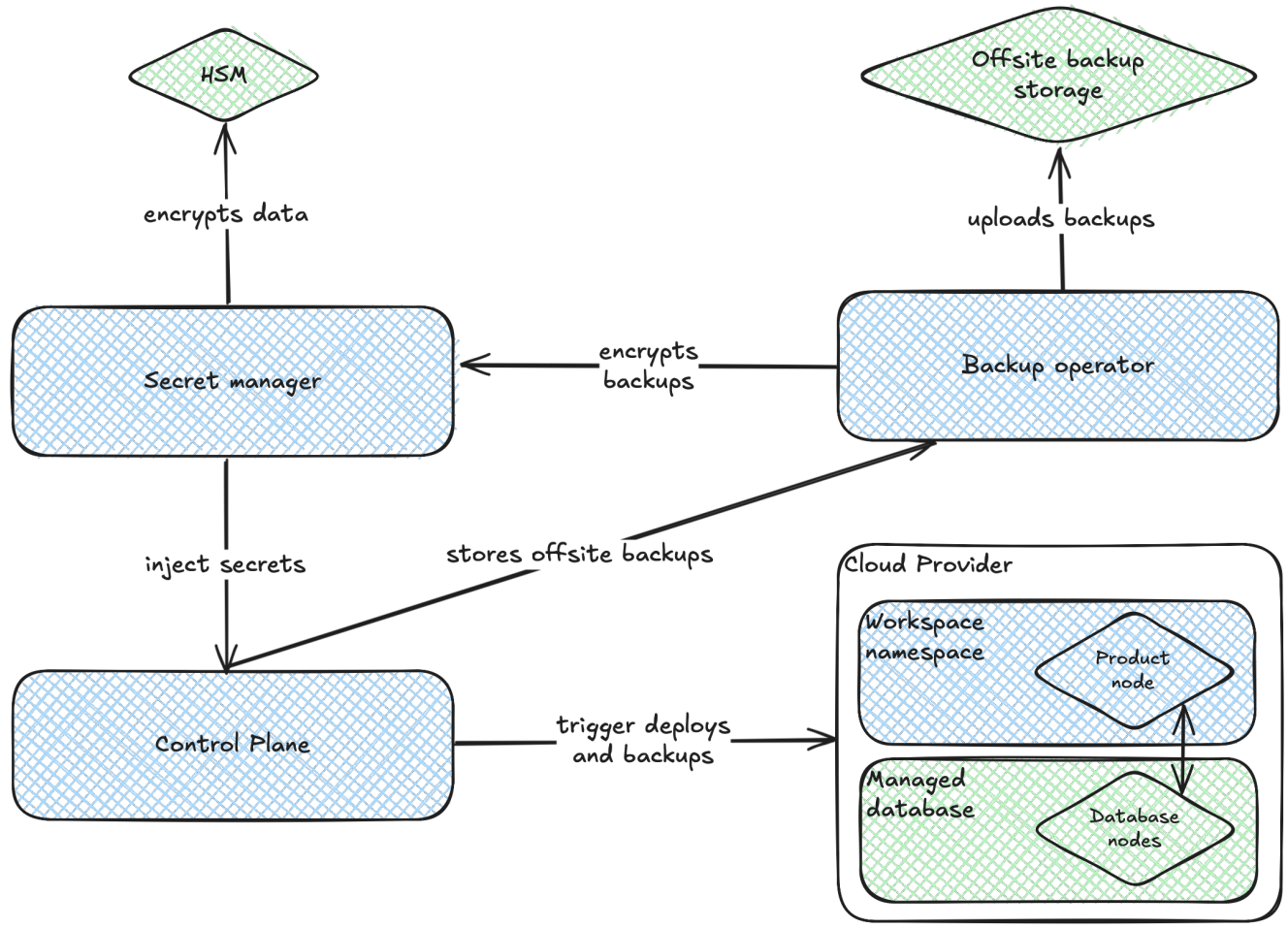


Figure 1. EVERTRUST Cloud components

Control plane

The control plane orchestrates operations and enables cloud-agnostic operations, as it centralizes deployment and configuration on every supported cloud provider. However, as mentioned in Platform outages, in case of control plane unavailability, instances can still operate independently, reducing the probability of a large-scale outage.

The following cloud providers are currently supported :

- AWS
- Google Cloud
- Scaleway

All supported cloud providers support high availability.

Secret manager

The Secret manager handles all secret generation, rotation and injection operations. Its secure storage is backed by an HSM, to ensure secret confidentiality.

It manages:

- backup encryption keys
- instance encryption keys

The reliability of the Secret manager is key to the overall platform stability and disaster recovery is described here.

Database

For performance and reliability reasons, EVERTRUST Cloud offloads database management to a third-party provider. The provider depends on the cloud provider where the instance is deployed to:

- for instances deployed on AWS or Google Cloud: the database is managed by MongoDB Atlas;
- for instances deployed on Scaleway: the database is managed by Scaleway Managed MongoDB.

The database is a critical component as it stores all stateful customer data and configuration. Database backups are managed by the Backup operator.

Backup operator

The backup operator handles database backups from managed databases. It introduces encryption and distribution across multiple cloud providers, and performs backups according to the backup policy.

A unique encryption key is generated by the secret manager for any given organization, to ensure encrypted backups can only be read by entitled organization members or service accounts. It will be used when backing up and restoring data for each organization.

Additionally, backup data will be replicated amongst multiple cloud providers to ensure maximum durability.

Chapter 2. Product versions

As an EVERTRUST Cloud customer, you benefit from automatic upgrades to the latest versions of EVERTRUST software. An instance is enrolled into a release channel, which determines when and how versions are applied. Overrides and maintenance windows are supported to give you more control over how and when upgrades are applied.

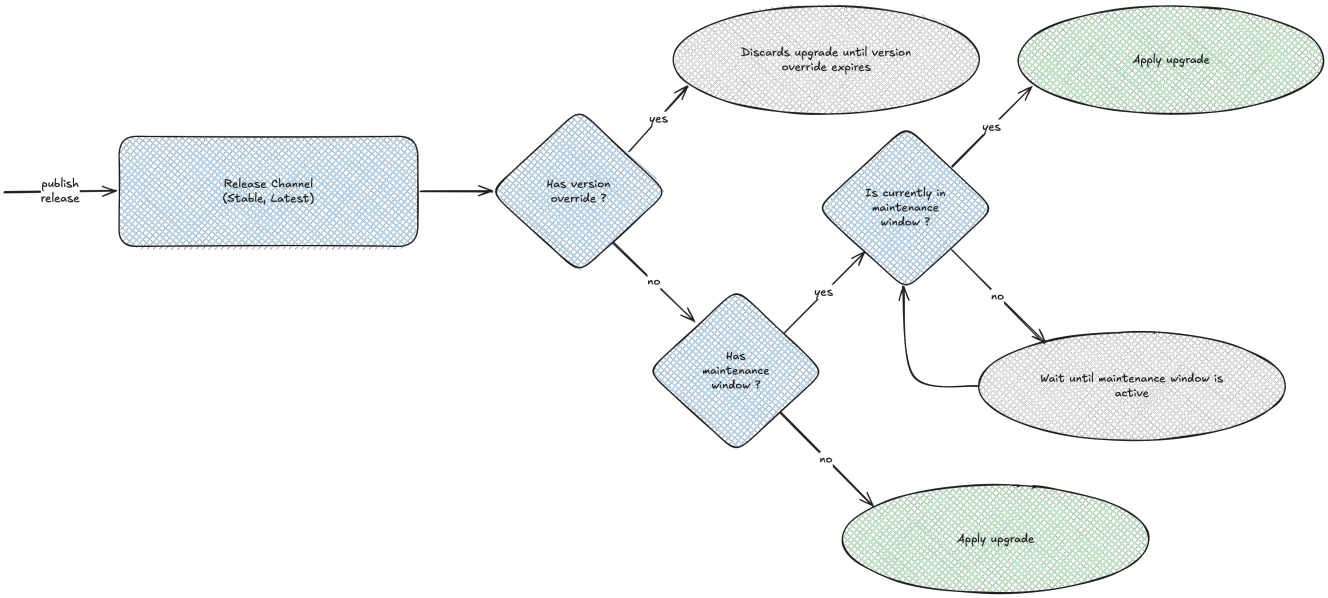


Figure 2. EVERTRUST Cloud upgrade flow

Versioning definitions

EVERTRUST Cloud products follow semantic versioning principles, which are reflected in version numbers structured as `MAJOR.MINOR.PATCH`.

- **MAJOR version**: Incremented when incompatible API changes are made. May require planning and preparation for upgrades.
- **MINOR version**: Incremented when new functionality is added in a backward-compatible manner. These updates enhance product capabilities while maintaining existing functionality.
- **PATCH version**: Incremented for backward-compatible bug fixes and minor improvements. These updates focus on stability and security.

In addition, certain versions may include suffixes like `-rc` (Release Candidate) to indicate pre-release status.

Version numbers provide clear signals about the nature and impact of changes between releases, helping you understand the potential implications of upgrades to your environment.

Release channels

Release channels are the main way upgrades are distributed by EVERTRUST. A channel defines the current version of the software that is installed on an instance. If no override is set, the instance will be automatically upgraded to the latest version available in the channel.

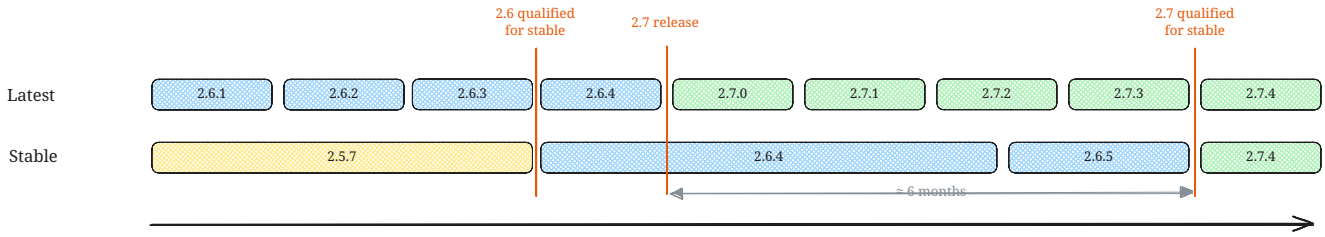


Figure 3. Example product release timeline with fictional releases

All events in orange will be announced on the status page. It gives instances in the Latest channel plenty of time to prepare for the upgrade on other instances in the Stable channel.

Stable channel

The Stable channel is best suited for production workloads. It prioritizes stability over new features. On instances using the **Stable** release channel:

- patches versions will be applied as soon as they are released;
- minor versions will be applied around six month after their initial release, after being qualified for stability. A migration notice will be published to status.evertrust.io **at least 30 days** before the upgrade;
- major version upgrades are always handled specifically and will be communicated in advance to customers.

Latest channel

The Latest channel is best suited for development and testing workloads. It prioritizes a fast delivery of new features over stability. On instances using the **Latest** release channel:

- patches and minor versions will be applied as soon as they are released;
- major version upgrades are always handled specifically and will be communicated in advance to customers.

Version overrides

Version overrides can be used to delay or expedite upgrades to a specific version. This can be useful to ensure compatibility with other systems or to test new features before they are rolled out to all instances.

The following override types can be requested :

- **Upgrade delay** can prevent an automated planned update to take place, and defer it to a later date;
- **Expedite upgrade** can upgrade an instance sooner than its planned automatic upgrade takes place.

Overrides should be requested through the EVERTRUST support portal, at least 48 hours before the

planned upgrade.

Maintenance window

A maintenance window can be set to define a time frame during which upgrades can be applied. This can be useful to ensure that upgrades are applied during a time that minimizes business impact, or to define exclusion windows. A maintenance window is defined at the service subscription and can be later amended by submitting a request on the EVERTRUST support portal.

If a maintenance window is supplied, it should allow for at least 4 continuous hours to apply patches and upgrades each week.

Chapter 3. Product usage

3.1. CRL distribution

Certificate Revocation Lists (CRLs) are files containing the list of revoked certificates that shouldn't be trusted anymore. As it's usually used by clients (with OCSP) to validate certificates, it's a critical part of the infrastructure requiring the highest availability.

As CRLs are flat files, there are multiple ways to distribute them to clients depending on the level of control required by the customer but an object storage service is usually a good solution. Two buckets are required when deploying a Cloud PKI:

- a bucket holding CRLs, that will be automatically updated by Stream
- a bucket holding AIAs, that will need to be updated manually each time a CA is signed.

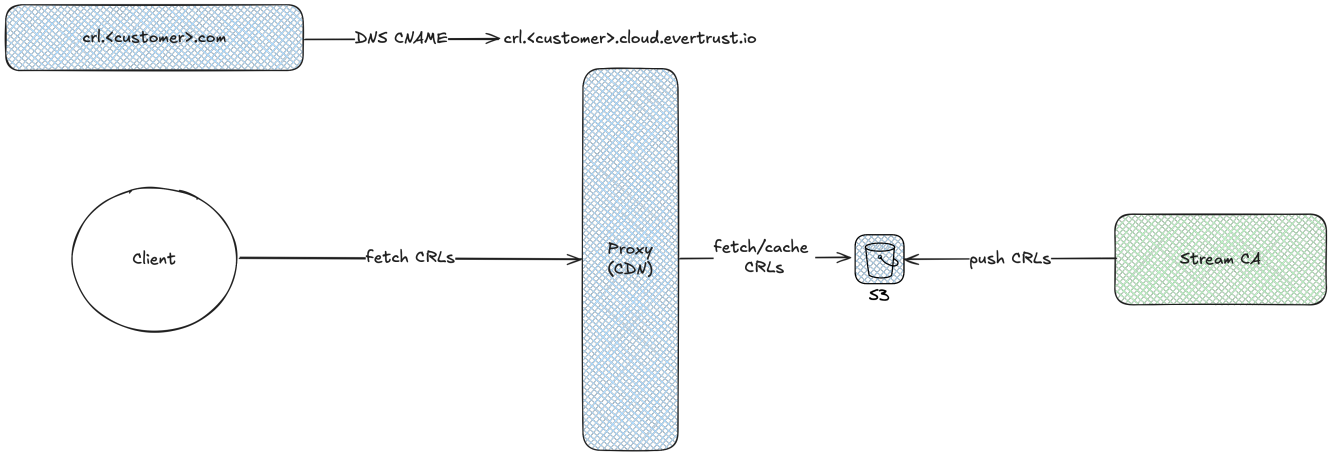
Customer-managed S3 buckets (Recommended)

The ideal storage solution is an S3 bucket hosted inside the customer's own cloud provider, as it allows the customer to be in control of the distribution point. Stream supports any S3-compatible storage provider and will upload CRLs as soon as they're generated (see the related documentation on how to set it up in Stream).

Should the customer need to integrate existing and legacy distribution URLs, or switch to another solution than Stream in the future, this option is the most flexible. However, this implies some responsibilities on the customer side:

- the customer will need to set up and manage the infrastructure to distribute CRLs: appropriate DNS records, usually some kind of cache proxy (CDN)…
- SLAs will need to be negotiated directly with the cloud provider, and won't be covered by EVERTRUST Cloud standard SLAs as the infrastructure is out of the EVERTRUST control

This set up can be summarized by the following diagram, where blue-colored elements are customer-managed and green-colored elements are EVERTRUST-managed:



EVERTRUST-managed S3 buckets

If the customer is not able to host the infrastructure themselves, EVERTRUST will provide S3 buckets hosted in the same environment as the Stream CA Instance. Credentials with write access to both buckets will be provided to the customer, so an External CRL Storage can be configured in Stream.

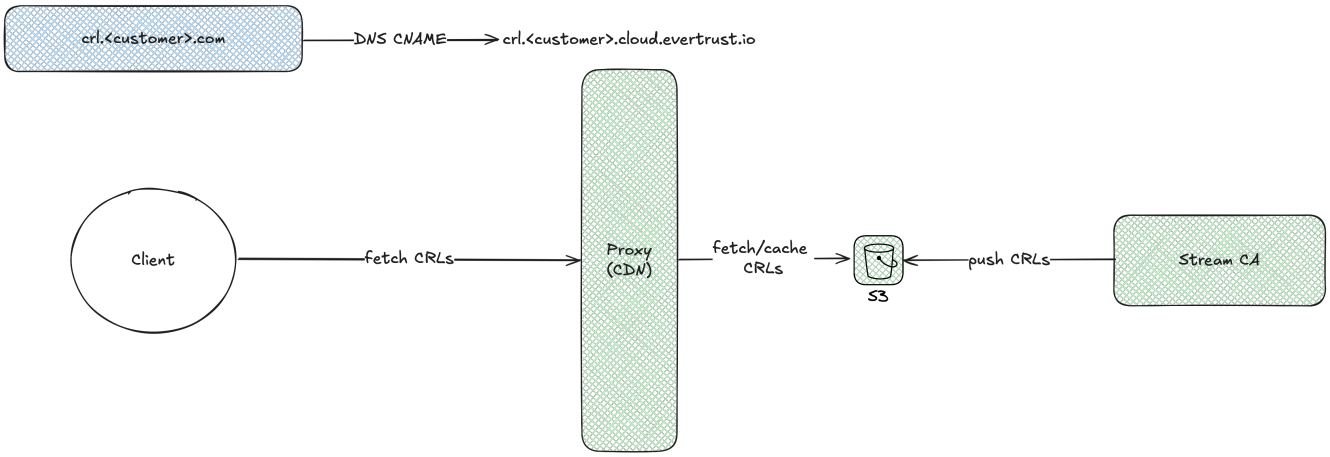
This type of setup introduces a dependency between EVERTRUST and the customer when client secrets need to be rotated. When doing so, the customer will be notified at least a week in advance and will need to update their External CRL Storage configuration accordingly to the EVERTRUST requirements, as EVERTRUST doesn't have direct access to the platform.

EVERTRUST will also provide a caching proxy that will distribute the CRLs to clients, which will allow access from a customer-defined DNS endpoint aliased to the EVERTRUST provided domain name :

Record type	Record value
CNAME	`crl.<customer>.cloud.evertrust.io`

For reversibility reasons, CRL distribution from an endpoint containing `evertrust.io` **is not supported**, and the customer-managed alias should always be used in CRLDPs.

This set up can be summarized by the following diagram, where blue-colored elements are customer-managed and green-colored elements are EVERTRUST-managed:



3.2. Mail notifications

EVERTRUST software is able to deliver e-mail notifications to users when provided with an SMTP server correctly configured to deliver mails.

Default SMTP server

EVERTRUST Cloud uses Amazon SES as its SMTP provider. The simplest (and the default) way that

comes configured with every instance is an EVERTRUST-managed email address, `noreply@cloud.evertrust.io`. This address will be enforced and won't be customizable.

When using the default SMTP server, EVERTRUST has to ensure compliance with anti-spam regulations. To do so, when an sent email bounces (because the recipient address is invalid), EVERTRUST Cloud will automatically disable email notifications to that specific address in the future. That may prevent a newly created address to receive notifications.

Custom sending domain

The sending domain can be customized by configuring a DKIM DNS record using a customer-managed domain containing a key that will be generated by EVERTRUST to sign emails. The "From" field will then be unlocked on the deployed Instances to allow customizing the sender identity, but the mail server will still be managed by EVERTRUST, reducing integration complexity.

It's recommended to dedicate a subdomain of the customer's main domain in order to identify notifications coming from third-party vendors such as EVERTRUST. If you don't, EVERTRUST Cloud will be able to send notifications as any of your existing email accounts.

Customer-managed SMTP server

Should you need more flexibility and happen to have an internet-exposed SMTP server, Instances can be configured to use it. This solution does not require any DNS modification but requires credentials to be managed by the customer. The following information is then needed to correctly configure the Instance:

- SMTP server host & port
- security type (SSL, TLS…)
- username & password

IP whitelisting is not supported as an SMTP authentication method, username/password should always be used.

3.3. Platform access

EVERTRUST Cloud instances are accessed through the Internet. Each deployed Instance is assigned a unique domain name with the following naming format: `<product>.<customer>.<environment>.evertrust.io`, where:

- `<product>` is either `ra`, `ca` or `va`;
- `<customer>` is either the customer identifier or an anonymous generated name, upon customer demand;
- `<environment>` is either `staging` for staging environments and `cloud` for production.

Custom domain

A custom domain can be configured for an instance by setting up a CNAME record pointing to the EVERTRUST-provided endpoint. It will have to be whitelisted by submitting a request to EVERTRUST's support, and will afterward be available alongside the default EVERTRUST endpoint. It is not possible to disable the EVERTRUST-provided endpoint.

To configure `clm.customer.com` as an alias for an instance, the following record should be created :

```
clm.customer.com. IN CNAME 3600 clm.customer.cloud.evertrust.io.
```

IP whitelisting

Outbound traffic (customer)

EVERTRUST Cloud instances are exposed on Internet behind load balancers. These load balancers use IP addresses that may change over time, in case of maintenance or scaling operations. To ensure proper connectivity from your infrastructure to EVERTRUST Cloud instances, if you use outbound IP whitelisting, EVERTRUST publishes the list of IP addresses used by its load balancers in a public text file, available at the following URL:

https://raw.githubusercontent.com/evertrust/ip-addresses/main/ips.txt

The list provided at this URL is updated automatically when changes occur. It is recommended to use this URL as a source for your outbound IP whitelisting configuration, if applicable, as IPs may be added or removed with a 48-hour notice.

Inbound traffic

It is recommended to configure inbound IP whitelisting to restrict which addresses or ranges that are authorized to connect to your cloud instance from the Internet.

Two types of ranges can be whitelisted:

- static CIDRs
- third-party services

Third party services are IP ranges from providers that are maintained by EVERTRUST. They can be used if you rely on such a third party, such as Microsoft Entra or Okta for SCIM provisioning. The following third-party services are supported:

- Microsoft Entra
- Okta
- Jamf

If a connection from a non-whitelisted address reaches the firewall, it will be dropped before reaching the application server.

Ingress configuration

Trust anchors

Multiple Root CAs are used for redundancy purposes. Public certificates used by the load balancer are issued by one of the following Root CAs:

- USERTrust RSA Certification Authority (zerossl.com)
- ISRG Root X1 (letsencrypt.org)
- GTS Root R1, R2, R3 (pki.goog)

Make sure your clients trust these Root CAs to ensure operational continuity.

As of January 2025, custom certificates are no longer supported for TLS termination of public endpoints. Private endpoints are not affected by this change, and you're still responsible for managing the certificates used for private endpoints.

TLS termination

The following ciphers are accepted for TLS termination :

- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-CHACHA20-POLY1305
- ECDHE-RSA-CHACHA20-POLY1305
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES256-GCM-SHA384

3.4. Key management

EVERTRUST Stream supports integrating with third-party KMSs and HSMs to secure signing private keys.

Integrating with a KMS

The recommended way to set up a KMS-protected key in Stream is to use the native Cloud KMS integration.

It’s also recommended that customers uses their cloud provider tenant to provision KMS keys, as this allows for reversibility and credentials management that is compliant with their own internal policies. In case the customer is unable to configure a KMS key in their tenant, EVERTRUST can provide a ready-to-use key. However, this has the same drawbacks as an EVERTRUST-managed customer bucket.

Integrating with an HSM

EVERTRUST builds container images that integrate multiple HSMs middlewares. The following HSM vendors are currently supported in EVERTRUST Cloud :

- Luna Cloud HSM (Thales DPoD)
- Entrust nSaaS

Chapter 4. Platform architecture

4.1. Security

Tenancy

EVERTRUST Cloud provides a single-tenant architecture, meaning each instance is in-use by exactly one customer. However, some resources are shared in the cloud platform among multiple customers:

- the database cluster holds multiple databases. Each instance authenticates to the database with a dedicated user which holds limited read and write rights to its own database. No instance can access other instances data.
- the workloads Kubernetes clusters hosts multiple workspaces. Each workspace has a dedicated namespace, which has strict firewall policies preventing communication with other namespaces. All instances within one namespace will however be able to communicate with each other.

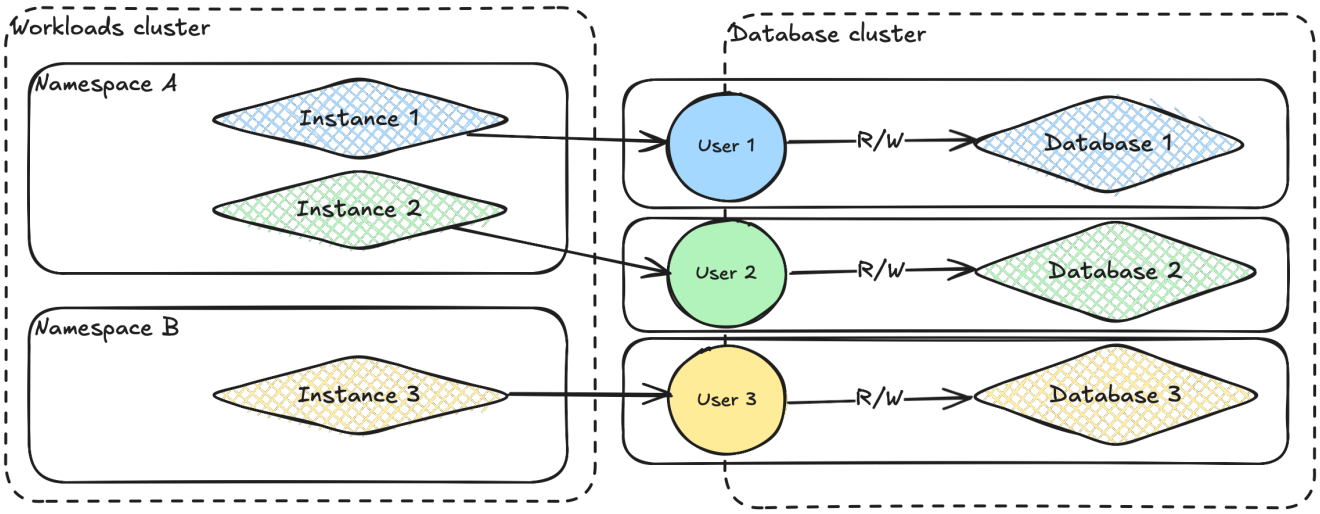


Figure 4. Tenancy diagram

In the diagram above, dotted line represent shared resources while non-dotted lines represent boundaries between customers.

Encryption

Encryption is at the center of operating PKI software. Multiple layers of encryption occur in EVERTRUST Cloud :

At-rest encryption

- The MongoDB database is configured to use MongoDB encryption at rest using an EVERTRUST-managed key, it encrypts data in the database engine on disk.
- Backups are encrypted on disk using a key derived from the secret manager component.

In-memory encryption

Provided by EVERTRUST products, it ensures that sensitive data is never written in clear onto disk or in database. It is backed by a symmetric key, unique to each instance, provided by the secret manager

In-transit encryption

All data in transit is secured by TLS encryption. Details about the TLS encryption parameters can be found in ingress configuration.

Request lifecycle

Each request in EVERTRUST Cloud goes through multiple steps before reaching the instance application. TLS termination occurs before in the WAF, so it can provide in-depth request analysis, such as :

- decrease the attack surface by restricting allowed file types, headers, endpoints...
- match and disrupt attack signatures
- provide a DDoS protection

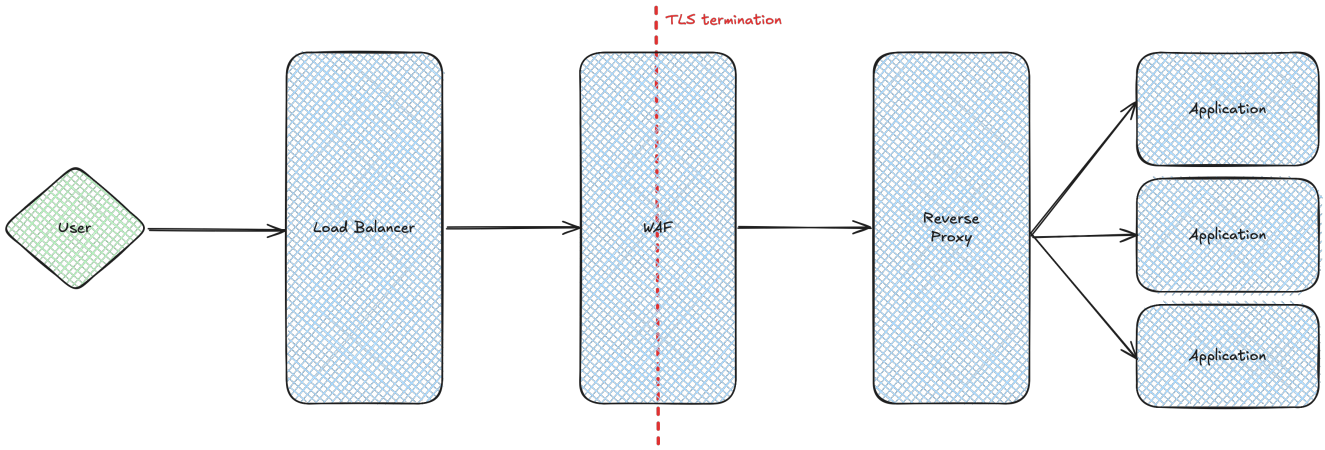


Figure 5. Request lifecycle in EVERTRUST Cloud

4.2. Monitoring

Log management

Two kind of logs are collected in EVERTRUST Cloud:

- technical logs
- application logs

Application logs (events) are stored in-database and signed using each instance seal secret. These logs are available to customers in the products UIs.

Technical logs include details such as WAF and reverse proxy activity, and application errors. These

logs are managed and accessible only by EVERTRUST. They are used for debugging purposes by the support and also trigger alerts, managed by on-call engineers as needed.

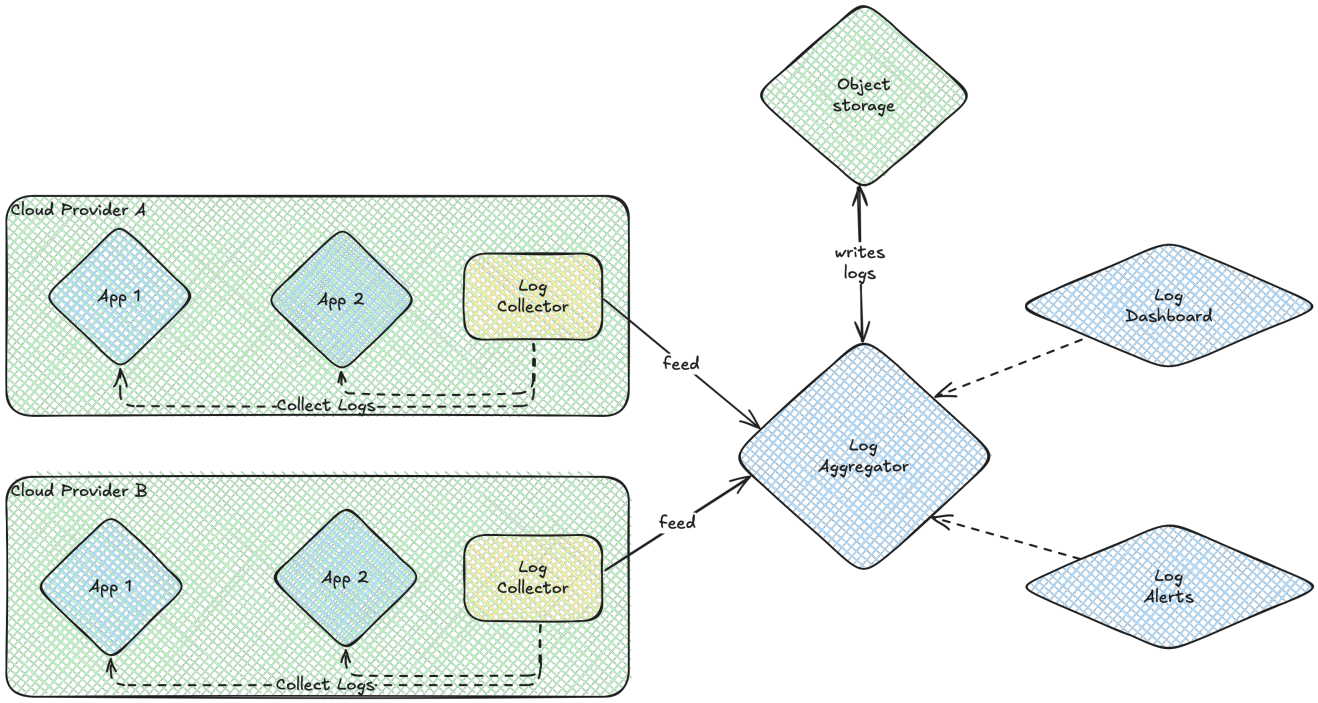


Figure 6. Log management infrastructure

Technical logs are retained for 90 days in EVERTRUST infrastructure.

Advanced monitoring

In addition to logs, EVERTRUST Cloud also collects metrics about your instance, such as:

- license usage
- application specific metrics, such as:
  - for Horizon: PKI connector statuses, credentials expiration
  - for Stream: keystore statuses

These metrics can be used by the support team to provide you with better contextual responses, as well as by other relevant teams (such as sales when your license is about to expire).

4.3. Reliability

High-availability

Every instance is deployed in a cluster where, if an availability zone fails, it can instantly be scheduled on another node in an available AZ.

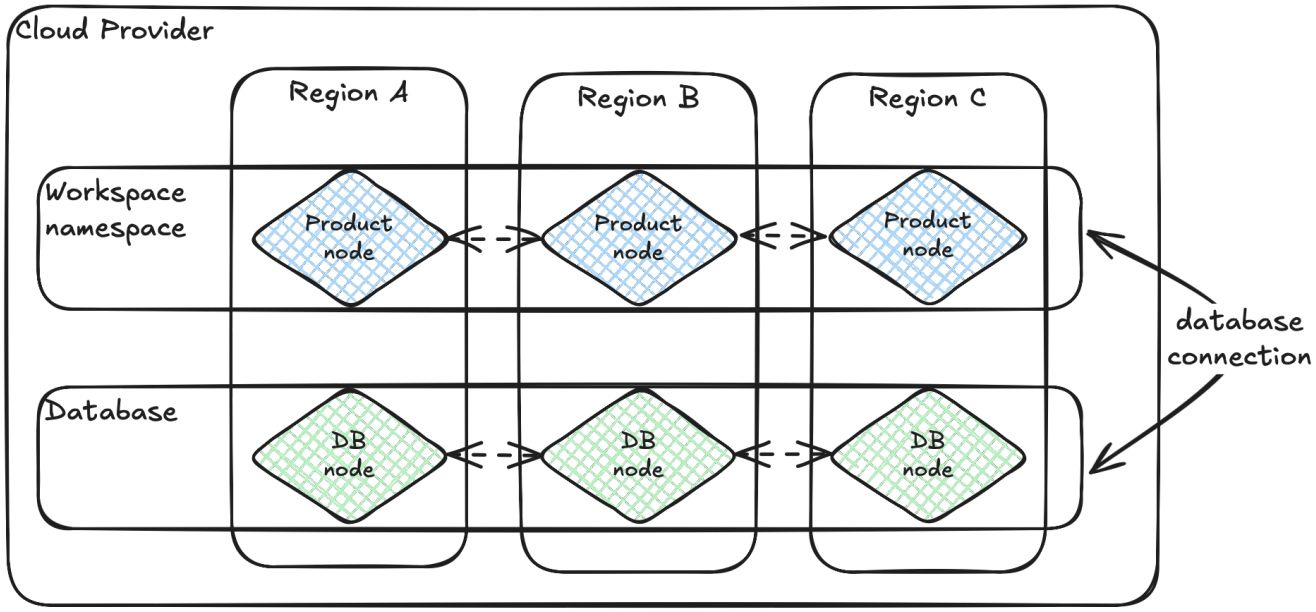


Figure 7. Deployment diagram across multiple regions

In case of a unavailability in multiple or all AZs in a cluster region, the EVERTRUST control plane will automatically migrate workloads to a cluster in the same cloud provider but in a different region.

Instances that include VPN connectivity or network peering cannot be automatically migrated to another region. Manual action will be required on the customer side to implement changes.

Disaster recovery

Backups are taken regularly on every instance database, triggered by the following policy:

- full instance backup: every 6 hour
- cluster snapshot: every hour

Full instance backups

Full instance backups are specific to a deployed instance, and can be restored upon customer request. They are replicated across multiple regions and cloud providers, and retention is 1 year.

Cluster snapshots

Cluster snapshots are used by EVERTRUST Cloud to ensure business continuity in case of an outage or data loss on MongoDB Atlas side. They are replicated in two regions in the same cloud provider, and retention is the following:

- Hourly snapshots: 2 days
- Daily snapshots: 7 days
- Weekly snapshots: 4 weeks

- Monthly snapshots: 12 months

Platform outages

The cloud platform service status is available at all times at https://status.evertrust.io. Incidents and maintenances will be reported through the status page above.

Additionally, the EVERTRUST Cloud platform has been designed to be able to operate even when some components are in a degraded state. If the control plane suffers from unavailability, some operations will be unavailable (upgrade, restores) but existing instances will continue to operate as normal.