Rockset Security Design

Abstract

Rockset is a serverless search and analytics engine. This document describes Rockset's security design.

June 2020

contact@rockset.com

[ROCKSET]

Rockset Security Design

Security is important to Rockset and taken very seriously. This document describes the current state of security of Rockset. This information can change rapidly, as we are constantly improving our security as we continue to build our product. Rockset is not yet compliant with HIPAA, SOX or PCI.

Use of Cloud Infrastructure

Rockset is a data processor. All data loaded into Rockset is owned by the customer. Rockset indexes and stores all the data to allow SQL queries, but the data is always owned by the customer.

Rockset uses cloud native best practices and exploits the underlying security policies of the public cloud it is hosted on. Architecturally, Rockset is designed to be cloud agnostic and may be run on AWS, Google Cloud, Azure or other public clouds in the future. However, currently, all of Rockset's services are run and hosted in Amazon Web Services (AWS), hence our security policies follow AWS best practices at this time. Anytime this document mentions Rockset servers, it means EC2 instances in AWS. Rockset does not operate any physical hosting facilities or physical computer hardware of its own. You can view the AWS security processes document here: https://d1.awsstatic.com/whitepapers/Security/AWS_Security_Whitepaper.pdf

Security Features

Data masking

For sensitive data like PII, Rockset supports data masking at the time of ingest utilizing field mappings. A field mapping allows you to specify transformations to be applied on all documents inserted into a collection. This can be used for type coercion, anonymization, tokenization, etc. When a particular field is masked using a hashing function like SHA256, only the hashed information is stored in Rockset. You can see how to do field mappings in Rockset here: https://docs.rockset.com/field-mappings/

Role based access control

Role-based access control (RBAC) helps you manage user privileges in your account. Specify roles and privileges to ensure that you limit access to your data to the individuals that need it.

Advanced encryption with user controlled keys

Rockset uses AWS Key Management Service to make it easy for you to create and manage keys and control the use of encryption. With advanced encryption, you can bring your own key which allows you to control the encryption and delete the key if needed.

Security Architecture

Data in flight

Data in flight from customers to Rockset and from Rockset back to customers is encrypted via SSL/TLS certificates, which are created and managed by AWS Certificate Manager. An AWS application load balancer terminates SSL connections to our API endpoint. We currently use the ELBSecurityPolicy-TLS-1-1-2017-01 security policy for our HTTPS load balancers: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html#describe-ssl-policies

AWS CloudFront distributions terminate SSL connections to our website and Console.

Within Rockset's Virtual Private Cloud (VPC), data is transmitted unencrypted between Rockset's internal services. Unencrypted data will never be sent outside of Rockset's VPC.

The only ways to generate traffic inside the VPC are through the Rockset API and secure VPN. Access to these is described in the following sections.

Data at rest

Data is persisted in three places within Rockset:

1. In a log buffer service on encrypted AWS EBS volumes. Rockset uses this log buffer as transient storage to independently scale data indexing (writes) and data serving (reads).
    1. http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html
2. On our servers, which have local solid state drives which are encrypted via dm-crypt.
    1. the configuration is based on this article: https://aws.amazon.com/blogs/security/how-to-protect-data-at-rest-with-amazon-ec2-instance-store-encryption/
3. In AWS S3, where all stored objects are encrypted
    1. http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingKMSEncryption.html

In all cases, the encryption keys are managed by AWS Key Management Service (KMS). The master keys are never exposed to anyone (not even Rockset), as they never leave the KMS hardware. For evaluation accounts, the master keys used are created in Rockset's AWS account. For commercial installations, Rockset will provide a way for customers to provide their own KMS master key. All customer API keys and integration credentials are stored in a secure admin database that is encrypted at rest.

Access controls

The only ways to access any stored data, servers, or services running inside Rockset's VPC are through the VPN server and Rockset API.

Access to the VPN server requires a TLS auth key, a CA certificate, and the user's individual password. Access to the VPN server also requires two-factor authentication and per user certificates. There are no shared passwords for accessing the VPN server. Only employees that require development and administrative access will receive credentials to access the VPN server. Access to resources in the VPN is controlled on a per user basis. For example only required individuals have the ability to send SSH traffic. VPN access for each individual employee can be quickly revoked if necessary.

Access to Rockset servers are controlled by a configuration management system that distributes employee public SSH keys to servers they are allowed to access, configures which UNIX group each employee belongs to, and configures /etc/sudoers to grant access to run commands to each employee. SSH private keys only live on the employee's laptop. SSH keys can be quickly removed and rotated on all servers in the event that the laptop is lost or the key is compromised. Only SSH key pairs are used for SSH access, not passwords. SSHing as root is also disabled. All laptops have Hexnode MDM enabled which enforces alphanumeric password of at least 12 characters, auto screen lock after 10 minutes, network firewall, FileVault disk encryption enabled, ability to remotely lock or wipe the computer and periodic scans to ensure the above are in compliance.

Within the VPC, we define specific network security groups for instances depending on the services scheduled on them. By default, none of them are publicly reachable via the Internet. Only the VPN server is reachable directly via the Internet, and security groups are locked down so that only the ports necessary to access the given service are allowed, and only from sources that require this access. Network access per security group is given only on an as needed basis (whitelist only the needed ports/protocols/sources as opposed to blacklist). There are Internet-facing load balancers (AWS ALBs) that service the API. These load balancers only accept inbound traffic over the HTTPS protocol and port 443, and automatically redirect HTTP traffic to HTTPS. There are also AWS CloudFront distributions that service traffic to our website and Console, and terminate SSL connections for those services. These endpoints also redirect HTTP traffic to HTTPS.

Access to Rockset's Console is based on having a valid username/password to an Auth0 user that has been granted access to the Console. Rockset relies on Auth0 for user authentication instead of implementing our own for now: https://auth0.com/docs/overview

Access to Rockset's API is based on API keys, which can only be created in the Console. Each API key can only access the resources granted to the user that created the key. All customer API keys and integration credentials are stored in a secure admin database that is encrypted at rest.

Access to AWS resources is provisioned through AWS Identity and Access Management (IAM) users and roles. Our EC2 instances are granted some access required for processes running on them to function (such as access to some S3 buckets) through instance roles. Developers and administrators who require AWS access are each given an IAM user account and a key for AWS API access. Access to the production AWS account also requires two-factor authentication.

Access controls can be granted on a per user and key level, so we can control fine grained access per employee.

Access to our cluster manager API is done through a role based access control mechanism.

Confidentiality

Confidentiality is maintained by only allowing those users, non user accounts, and employees access to systems, services, and data when they need it. Secrets that are needed by Rockset's services during run time are stored in AWS Secret Parameter Store, with access to those secrets provisioned via AWS instance roles. Secrets needed by employees to access various third party services are stored in 1Password. Rockset usernames and passwords are stored in Auth0.

Audit Logging

Rockset uses AWS Cloudtrail to log all actions taken on our cloud infrastructure. For each AWS account Rockset uses (including production), all logs are shipped to an S3 bucket in a completely separate security AWS account. This bucket only grants write (not read) access to the AWS Cloudtrail service. Cloudtrail log file validation is turned on so that any modification to the log files can be detected:
https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-intro.html?icmpid=docs_cloudtrail_console