RUBRIK CLOUD DATA MANAGEMENT

SECURITY TARGET

VERSION 1.2

Rubric, Inc. – www.rubrik.com

1001 Page Mill Road, Building 2, Palo Alto CA, 94304 USA

□

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviations

Abbreviation	Description
AD	Active Directory
API	Application Programming Interface
CDM	Cloud Data Management
CHAP	Challenge-Handshake Authentication Protocol
CIFS	Common Internet File System
IPMI	Intelligent Platform Management Interface
iSCSI	Internet Small Computer Systems Interface
LDAP	Lightweight Directory Access Protocol
MSRPC	Microsoft RPC
NBD	Network Block Device
NFS	Network File System
ORM	Object-relational mapping
REST	Representational State Transfer
RBAC	Role Based Access Control
RPC	Remote Procedure Call
SMB	Server Message Block
SOAP	Simple Object Access Protocol
SPN	Service Principal Names
SSH	Secure Shell
SDFS	Dedup File-System
TLS	Transport Layer Security
VADP	VMware vStorage API for Data Protection
VIM	Virtual Infrastructure Methodology
VMDK	Virtual Machine Disk

Definitions

Definition	Description
AD	Microsoft Windows directory service that facilitates working with interconnected, complex and different network resources in a unified manner
Avahi	Free zero-configuration networking (zeroconf) implementation, including a system for multicast DNS/DNS-SD service discovery
API	Set of routines, protocols, and tools for building software and applications
Cloud Data Management	A system that distributes data, metadata, and task management across the cluster in order to deliver predictive scalability and eliminate performance bottlenecks.
CHAP	Authenticates a user or network host to an authenticating entity, for example an Internet service provider
SMB	A protocol that defines a standard for remote file access using millions of computers at a time. With SMB, users with different platforms and computers can share files without having to install new software. This was previously referred to as CIFS however Microsoft has deprecated

	usage of the term CIFS in favor of SMB.
Data At Rest Encryption	Encryption protecting of data that is not moving through networks
Data deduplication	Specialized data compression technique for eliminating duplicate copies of repeating data
Edge	A software-only version of the Rubrik CDM product that runs on a virtual machine.
Hypervisor	Hypervisor (or VM monitor) is a piece of computer software, firmware or hardware that creates and runs VMs
IPMI	A remote hardware health monitoring and management system that defines interfaces for use in monitoring the physical health of servers, such as temperature, voltage, fans, power supplies and chassis
iSCSI	IP based storage networking standard for linking data storage facilities
MSRPC	Microsoft Remote Procedure Call applies to Windows Server, for creating distributed client/server programs
LDAP	An open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network
NBD	A device node whose content is provided by a remote machine, used to access a storage device that does not physically reside in the local machine but on a remote one
NFS	Distributed file system protocol allowing a user on a client computer to access files over a computer network much like local storage is accessed
ORM	A programming technique for converting data between incompatible type systems in object-oriented programming languages, which creates, in effect, a "virtual object database" that can be used from within the programming language
Quiescing	To put a computer, a program, a thread, or some other computer resource into a temporarily inactive or inhibited state
REST	The underlying architectural principle of the web (The software architectural style of the WWW)
REST API	An application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data
RBAC	Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file.
RPC	Client/Server system in which a computer program causes a subroutine or procedure to execute in another address space without the programmer explicitly coding the details for this remote interaction
SDFS	Open source software for deduplication of data
Snapshot	Backup copy of a virtual machine. Each snapshot is a file. The first snapshot is a full copy of the virtual machine.

	Each subsequent snapshot is an incremental delta from the previous file. Every snapshot is a fully functional, point-in-time copy of the source VM
SOAP	Protocol specification for exchanging structured information in the implementation of web services in computer networks
SPN	A unique identifier of a service instance, used by Kerberos authentication to associate a service instance with a service logon account. This allows a client application to request that the service authenticate an account even if the client does not have the account name
SSH	A program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels
Thrift	Interface definition language and binary communication protocol that is used to define and create services for numerous languages. It is used as a RPC framework that combines a software stack with a code generation engine to build services that work efficiently to a varying degree and seamlessly between numerous languages
TLS	A protocol that guarantees privacy and data integrity between client/server applications communicating over the Internet
VADP	Backs up and restores vSphere VMs. (VADP was introduced in vSphere 4 and replaces the VMware Consolidated Backup framework)
VIM	A four-phased methodology developed and employed by VMware to consistently deliver comprehensive solutions to assess, plan, build and manage VMware virtual infrastructure
VM	Virtual Machine
VMDK	File format that describes containers for virtual hard disk drives to be used in virtual machines like VMware Workstation or VirtualBox (hypervisor)
VMware	Virtualizes computing, from the data center to the cloud to mobile devices, to help Rubrik’s customers be more agile, responsive, and profitable
VMware ESXi (formerly ESX)	Enterprise-class hypervisor developed by VMware for deploying and serving virtual computers
VMware VADP	VMware vStorage API that backs up and restores vSphere virtual machines
VMware VIM	Four-phased methodology designed to create a range of virtual infrastructure solutions
VMware vSphere	Server virtualization suite which consists of several technologies that provide live migration, disaster recovery protection, power management and automatic resource balancing for data centers

1. ST Introduction (ASE_INT)

1.1. ST and TOE References

The following table identifies the Security Target (ST).

Item	Identification
ST title	Rubrik Cloud Data Management Security Target
ST version	1.2
ST author	NTT Security (Norway) AS / System Sikkerhet AS

The following table identifies the Target of Evaluation (TOE).

Item	Identification
TOE name	Rubrik Cloud Data Management
TOE version	Rubrik Version 4.1.2

The following table identifies common references for the ST and the TOE.

Item	Identification
CC Version	3.1 Revision 5
Assurance level	EAL2 augmented with ALC_FLR.1
Protection Profile	None

1.2. TOE Overview

Rubrik Cloud Data Management is a software platform that distributes data, metadata, and task management across the cluster in order to deliver predictive scalability and eliminate performance bottlenecks.

- **"The Core"** is the foundation of Rubrik and is comprised of the file system, metadata service, cluster management, and task framework.
- **"The Logic"** functions as the brains of Rubrik by organizing, removing redundancy, and making data available for search.
- **"The Interface"** provides a RESTful API-driven interface that interacts with users and supports virtualization, applications, and public cloud technologies.

The Core

Rubrik Cloud-Scale File System

Rubrik Cloud-Scale File System is a distributed file system built from the ground up to store and manage versioned data. We have designed this file system to be:

- **Fault Tolerant:** The system is resilient to multiple node and disk failures. We employ an intelligent replication scheme to distribute multiple copies of data throughout the cluster.

- **Flash-Optimized:** The system is built for a hybrid flash/disk architecture to maximize I/O throughput.

- **Storage Efficient:** The system utilizes zero-space clones to make multiple copies of data from one "golden image."

- **Scale-out NAS server:** The system exposes itself as a scale-out NAS server to any host when a snapshot is mounted.

Rubrik Distributed Metadata System
Rubrik Distributed Metadata System operates alongside our Cloud-Scale File System, providing an index that can be accessed at high speeds. It delivers continuous availability, linear scalability, and operational simplicity with no single point of failure in the cluster. Our system is built to handle large amounts of data, distribute replicas of data across nodes (access to metadata is maintained even in the case of node failure), and provide low latency operations.

Rubrik Cluster Management
Rubrik Cluster Management manages the Rubrik system setup and ongoing system health. We use a zero-configuration multicast DNS protocol to automate appliance discovery – the cluster expands with minimal manual intervention with new nodes auto-discovering each other. Post system setup, it maintains the status of each node by performing health checks on individual nodes.

Rubrik Distributed Task Framework
Rubrik Distributed Task Framework is the engine that globally assigns and executes tasks across the cluster in a fault tolerant and efficient manner. As a result, tasks are load balanced across the entire cluster, and tasks are distributed to the nodes that house the impacted data. This engine runs on all nodes and incorporates a masterless architecture where all nodes cooperatively schedule and run tasks.

THE LOGIC

Rubrik Data Management and Global Search

Rubrik Data Management serves as the "brains" of the system, enabling cradle-to-grave data lifecycle management from data ingest to archive and retirement. Fast, efficient data delivery is made possible by the ability to:

- store versions of data (we use a combination of full snapshot with forward incremental and reverse incremental copies)
- ensure data integrity (we build multiple checks within the file system and data management layers)
- apply content-aware global deduplication and compression (we intelligently apply data reduction at a global level while enabling fast data reconstruction)

THE INTERFACE

Programmatic Interface & Ecosystem Support

Our user interface is built on a RESTful API-driven framework with a HTML5 web user interface. Our UI is designed to provide ease-of-use and drive intuitive actions while reducing information overload for the user.

Rubrik is a vendor-agnostic platform with the ability to support any third-party ecosystem technology by building additional modules to the integration layer. This layer exposes the API set for building custom integration points into applications, hypervisors, containers, and protocols.

How It Works

Rack-and-Go System Setup

Once racked, Rubrik system setup is easily and quickly completed in 10-15 minutes. We invoke multicast DNS protocols to automatically discover and self-configure each of the nodes within the cluster. The user assigns IP addresses to each of the nodes (e.g., a r340 Appliance has four nodes) and login credentials for the virtualized primary environment to be managed by Rubrik. Various physical OSes and databases are also supported. To expand cluster size, the user simply assigns new IP addresses through the management dashboard. To reduce cluster size, the user selects the nodes to remove. Thereafter, the cluster automatically self-adjusts and re-balances to deliver fault tolerance against node and disk failures.

Automated Data Discovery

Once the user enters the credentials for its virtualized environment (e.g., vCenter username/password for VMware vSphere environments), Rubrik auto-discovers details of the entire virtualized environment, such as hosts and applications. Auto-discovery happens a variety of ways, depending on the user environment. Rubrik utilizes VMware APIs (vStorage APIs for Data Protection) to discover VMware environments. Support for additional virtualization hypervisors, containers, and applications will be rolled out in future releases.

Dynamic Policy Engine

From the list of discovered virtual machines (VMs), the user selects which VMs to protect and what SLA policies to apply for recovery. An SLA policy is made of 4 components that can be configured within minutes.

1. Frequency of backup
2. Retention of backups
3. Archival policy (when data is archived for cost-effective long term retention)
4. Replication to another Rubrik cluster for Disaster Recovery restore purposes).

Once a policy is configured, there is no need to configure individual jobs or tasks for scheduling or data movement between archival or replication targets. To illustrate this ease of management, we have pre-configured SLA policies based on industry standards.

As stated, the user has the flexibility to create new SLA domain policies by specifying the desired snapshot capture frequency and data retention policy. Users can select where data is stored, whether on-premise in the Rubrik Appliance or in a public cloud service (e.g., Amazon S3). The user simply slides the bar to the time at which data should be stored in the public cloud (e.g., 30 days). Rubrik provides a cost-effective alternative to tape for long-term data retention.

Rubrik allows users to intelligently and safely leverage the cloud. Only deduplicated data is transferred to the cloud. Data inflight and at-rest in the cloud utilize military-grade AES 256-bit encryption.

Flash-Speed Data Ingestion

We have designed Rubrik as a high-speed data ingestion engine that can easily handle large volumes of data. Rubrik pioneers the usage of flash in backup and recovery, resulting in extremely fast data extraction and minimizing performance impact to the production environment. In addition, we have built an intelligent distributed workflow management system to maximize the number of parallel data streams processed. Since Rubrik is architected to be a web-scale system, performance for every dimension (such as network and disk throughput) increases predictably at a linear pace as more nodes are added to the cluster.

For VMware environments, we utilize VMware's Changed Block Tracking to identify and copy only the changed blocks from the previous operation. We apply intelligent global deduplication and compression before the data is stored in our cloud-scale file system. All metadata is stored in the flash tier for rapid access in a search pulldown. Data is distributed across multiple nodes to deliver a fault tolerant file system.

Easy, Fast Global File Search

Rubrik eliminates the file search complexity inherent in legacy backup and recovery solutions by introducing consumer-grade file search that delivers query results instantly. As the user types the query, Rubrik expedites the query by displaying suggested search results with auto-complete functionality. The user can instantly locate specific versions of files across all VMs.

Instant_Recovery

By converging backup software and globally deduplicated backup storage into a single software fabric, Rubrik radically simplifies the recovery process. With just a click, users can instantly recover the VM by booting the virtual machine disk file (VDMK) directly on the Rubrik system. Rubrik serves as a storage endpoint for users to recover as many VMs as needed, eliminating the complexity and time wasted in transferring data back into the production system for recovery, thus providing a near zero RTO. Post-recovery, users can either choose to Storage vMotion the VMDK to the primary storage environment or continue using Rubrik as a storage endpoint. Rubrik's flash usage delivers fast IO performance. Writes and reads are gathered on the flash tier to deliver performance required by the recovered application.

Live Storage for DevOps

Rubrik pioneers the concept of Live Storage in which any copy and many copies of data can be mounted directly on Rubrik as a storage endpoint. As a result, Rubrik can be used to accelerate application development by providing multiple copies to developers from just one "golden image". Our Cloud-Scale File System has built-in native cloning capabilities to allow any number of mounts to be created without requiring additional storage capacity. Users can provision as many copies to developers as needed without impacting storage capacity and within a sandbox environment to prevent any network conflicts. As developers alter the provisioned data set, Rubrik stores the deltas by forking to a new branch. Our journaled Cloud-Scale File System provides an extremely efficient mechanism for accelerating and provisioning the latest data for application development. For medium-sized workloads, users receive all-flash performance comparable to a primary system of similar capacity. Rubrik intelligently allocates the flash tier for all writes and hot reads when utilizing Live Storage.

Rubrik redefines how data can be simply managed across data protection, disaster recovery, archival for compliance and long-term retention, application development, and data analytics.

We deliver the industry's first Cloud Data Management Platform by combining backup software and globally deduplicated storage into a single, scale-out fabric. Rubrik horizontally scales to thousands of nodes in a single system. We package Rubrik with industry standard hardware and avoid usage of proprietary hardware.

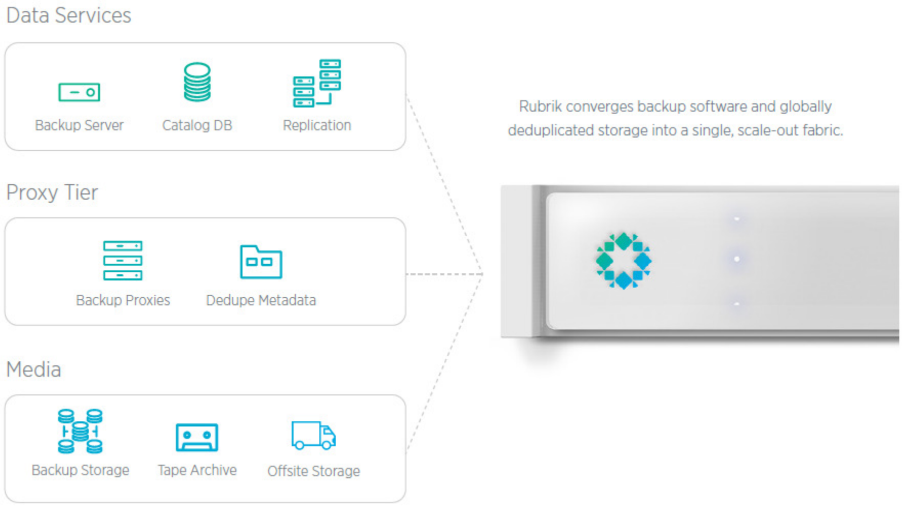

Figure 1: Single Converged, Scale-out Fabric

The Rubrik cluster accesses virtual machine data through a connection with the VMware vCenter Server that manages the hypervisor that is running the virtual machine. To successfully connect with a vCenter Server, the Rubrik cluster requires connection information for that vCenter Server.

1.2.1. Usage and major security features of a TOE

TOE has been designed in such a way that it can be combined with commodity hardware.

TOE makes it possible to provide the following important capabilities:

1. End-to-end data protection.
2. Storage requirements.

These capabilities shall be satisfied for an enterprise with simplicity, scalability and ease of management.

1.2.2. TOE TYPE

The Rubrik Cloud Data Management (TOE) is categorized as a Cloud Data management platform.

1.2.3. REQUIRED NON-TOE HARDWARE AND SOFTWARE

TOE requires the following non-TOE hardware:

- A VMware ESXi server used for running Rubrik Edge, which is a commodity computer (that is centrally managed by the Rubrik cluster).

TOE requires the following non-TOE software:

- Rubrik Edge, which is a virtualized software-only version of the Rubrik Cloud Data management product

TOE requires the following non-TOE software:

- VMware ESXi, which is the host environment for a Rubrik Edge virtual machine.

TOE requires the following non-TOE software:

- Rubrik Backup Connector, which is installed on each backup source, is required to do a backup of the physical infrastructure.

1.3. TOE DESCRIPTION

Each CDM Platform contains the same set of software components, see figure 2 below.

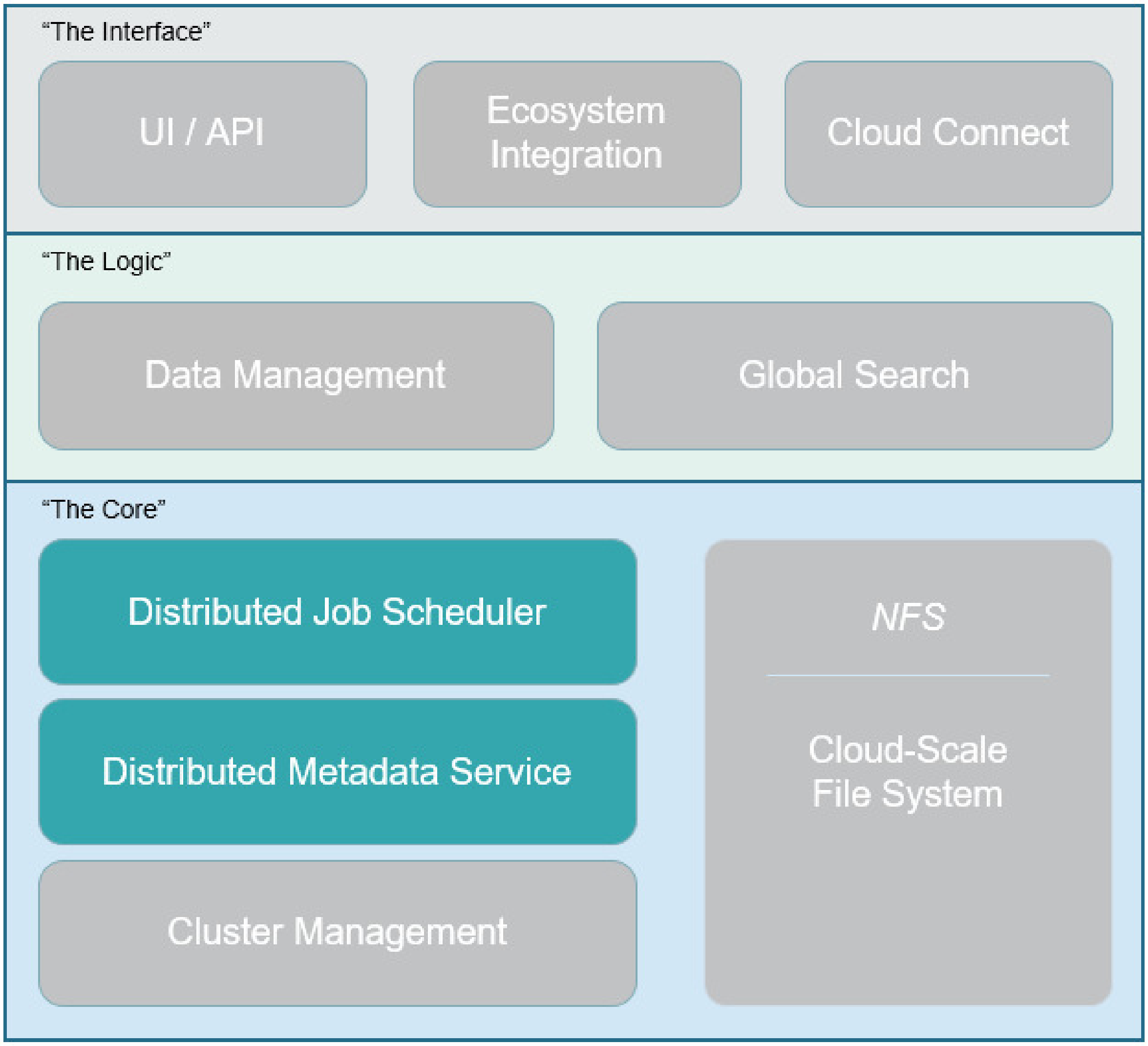

Figure 2: Rubrik Technology Stack

The internal components Cluster Management, Cloud-Scale File System, Data Management Layer and Distributed Job Scheduler, talk to other instances of the same component on other nodes using a Thrift RPC protocol. This protocol is implemented on top of an encrypted, TLS based transport layer. To prevent man-in-the-middle attacks, all internode TLS connections are verified using a private key and a public key certificate shared by all nodes in the cluster (data in flight encryption), using both client and server certificate. The key (2048-bit RSA) and the certificate (self-signed including the clusters' UUID in the name) are generated when the customer first installs the Rubrik cluster ("bootstrapping"), and are known only by nodes within that cluster. In addition, the private key and the certificate are distributed to all nodes during bootstrap via a Thrift RPC. Prior to and during bootstrap, this Thrift RPC uses a fixed TLS certificate and private key shipped with all nodes. After bootstrap, the new per-cluster certificate and private key are used.

To keep TOE data secure, data at rest encryption shall protect sensitive data contained in the backups, both locally and archived across all media supported by the TOE.

Web Application

The user interface is accessible over HTTP/HTTPS, where the HTTP requests are redirected to HTTPS. The HTTPS interface services both static file requests (e.g. JavaScript, CSS and HTML) and REST API requests. The static file requests are unauthenticated and convey no private information. The REST API requests require authentication via a username and password. Initially, the user logs in using the login REST endpoint, which provides a session token that is included in all subsequent requests. This token is invalidated after the user explicitly logs out, or after a configurable inactivity timeout expires.

Backup Integrations

The Snappable component is used for communication with backup sources. The component supports VMware backups by use of VMware's VIM and VADP APIs, and in addition it supports physical backups. The interfaces of SMB, MSRPC and SSH are for management use.

VMware Backup

Snappable supports backups from VMware vSphere infrastructure, by use of VMware's VIM and VADP APIs. VIM is vSphere's SOAP/RPC API for management operations, and VADP transfers data with ESX hosts using VMware's NBD which is authenticated with the same credentials used for the VIM API. Optionally, VADP may transfer data directly from iSCSI data stores if they are in use, where this connection is authenticated with separate credentials in bidirectional CHAP.

Snappable will store credentials with access to vSphere, VM guest OS and optionally iSCSI data stores:

- vSphere credentials are used in the VIM API to manage VMs and trigger snapshots, then used again with VADP to authenticate the connection for VMDK data transfer, (one feature of the VIM API is to transfer files or run commands within the virtual guest OS)
- VM guest OS credentials are used to deploy and manage its application quiescing agents to the VM guest
- iSCSI CHAP credentials are used with VADP to transfer VMDK data directly with iSCSI data stores

Physical Backups

Snappable supports backups or filesets of physical machines (Windows, Linux, Microsoft SQL Server, or NAS) using a Rubrik Backup Connector that is installed on each backup source. The Rubrik Backup Connector runs as a service with system level (root) permissions on the backup source. The Rubrik cluster and each connector communicate via a TLS encrypted Thrift RPC protocol. A fileset defines a set of files and folders on a Linux or Windows host. The Rubrik cluster uses the filesets that are paired with a host to determine the data to manage and protect. Rubrik interprets a fileset based on the information provided in the Include, Exclude, and Exempt fields, and on a set of fileset rules.

CLOUD ARCHIVAL

The Cloud Connect component is used for Cloud Data archival, allowing backups to be archived to an NFS server or an S3 compatible object stores.

A user provides its own Archival export on a server under user control, and configures the TOE to archive to this export using either the Web UI or REST API. The user provides the IP/hostname and export directory for the NFS archival target. Cloud Connect allows the user to specify two possible authentication mechanisms for NFS, UNIX (the default for the NFS protocol) and Kerberos:

- For UNIX, the user allows the TOE to access the user's NFS archival target using an IP address based whitelist (configured on the user's NFS server)
  - For Kerberos, the user must first add the Rubrik cluster to its MS AD domain. As part of joining the domain, the Rubrik cluster creates a machine account with appropriate SPN to enable Kerberized NFS access. When archiving to an NFS server using Kerberos, the TOE first requests a Kerberos ticket, and then authenticates against the NFS server using this ticket. The TOE will support this Kerberos configuration provided by the NFS protocol: krb5p (authentication and privacy).

FILE SYSTEM

The Cloud-Scale File System component is the TOE's distributed file system, communicating with other Rubrik nodes using a Thrift RPC protocol. Additionally, it exposes a NFS export to enable remote ESXi hypervisors to mount virtual machines using snapshots stored within the TOE ("live mounts"). To launch a live mount, the user issues a request via the TOE's Web UI or REST API to mount a specific point-in-time snapshot of a particular VM. The Cloud-Scale File System then materializes the set of files needed by the hypervisor to mount this snapshot. These files are exposed via the NFS export. The NFS export uses an IP addressed based whitelist to enable only specific ESXi hosts to access the exported snapshots.

CLUSTER MANAGEMENT

The Cluster Management component is responsible for managing and monitoring the Rubrik cluster. It is also responsible for the initial customer install ("bootstrap") process where a coherent cluster is first formed from a collection of new Rubrik nodes. A new ("non-bootstrapped") Rubrik node can be added to any Rubrik cluster without authentication. It is the bootstrap process that associates a node with a particular cluster and provides it with a basis for further authentication (e.g., for the internode Thrift RPC calls made by the TOE's services).

Bootstrap process

To bootstrap a cluster, the user accesses the Web UI on any non-bootstrapped node. New Rubrik nodes are shipped without IPv4 enabled and have only a link-local IPv6 address. To facilitate auto-discovery, pre-bootstrapped nodes broadcast their IPv6 addresses using Multicast DNS (Avahi). The user can access a node's UI in their browser by entering the node-name broadcasted by Avahi, which is derived from the node's serial number.

Once the user has successfully accessed a node's UI using IPv6, they are prompted to set a password for the local admin user of the cluster. This user has the highest-level access on the Rubrik cluster. After configuring the local admin, the user is shown a list of non-bootstrapped Rubrik nodes that have been discovered via Multicast DNS. For a standard four-node cluster or a virtual appliance that runs on a customer's hypervisor, this list will include the node whose UI the user is accessing, along with the other three nodes in the same physical appliance. If other non-bootstrapped Rubrik nodes are present on the network, they will be shown as well. The user selects the set of nodes that will form the cluster and sets IPv4 addresses for each of the nodes. After this screen, the cluster begins its bootstrap process.

Internally, the bootstrap process configures the Distributed Metadata instance on each node to form a coherent metadata cluster. It also enables IPv4 using the user provided addresses. The bootstrap process then installs the metadata schema and configures services on each node. Finally, it generates a 2048-bit RSA local-cluster key and copies it to each node. This key is used to allow SSH access between the nodes, which is used by Ansible as part of the software upgrade process. After bootstrap, the user may access the Web UI using the IPv4 address for any node in the cluster. Because the cluster has been bootstrapped, this access will require authenticating as the local admin or a user authorized by the local admin.

Adding nodes

In addition to bootstrapping a new cluster, non-bootstrapped nodes may be added to an existing cluster. To add a node, the user logs into the Web UI for any node in the cluster. They then visit an "add node" screen that displays all nodes discovered via Multicast DNS. The user selects the desired nodes to add and configures the IPv4 addresses for the new nodes. The Cluster Management component then adds these nodes to the existing cluster and copies the local-cluster key to the new nodes. After this process completes, the new nodes are in the same state as nodes added during the initial cluster bootstrap process.

Distributed metadata Service

The Distributed Metadata Service component is used for distributed storing of metadata. Each chunk of metadata is replicated onto three Rubrik nodes for fault tolerance and durability. The distributed metadata instance on one node communicates with the instance on other nodes using a TLS encrypted gossip protocol. For authentication, this protocol uses both client and server verification, using a private key known only to nodes belonging to the cluster (like the local-cluster key).

Data Management Layer, Search Component and Distributed Job Scheduler

The Data Management Layer, Search component, and the Distributed Job Scheduler components are internal to each node and do not directly interact with outside interfaces:

- The Data Management Layer manages backup policies, including snapshot expiration
- The Search component indexes each snapshot to facilitate individual file recovery through the Web UI. The index it produces is stored as a file in the Cloud-Scale File System
- The Distributed Job Scheduler coordinates jobs across the cluster by interfacing with the local Distributed Metadata Service instance on each node

1.3.1. Physical scope

TOE is purely software and can be installed on several physical devices if the non-TOE requirements in section 1.2.3 are valid.

The supporting guidance documents are:

1. Rubrik User Guide, Version 4.1
2. Rubrik CLI Reference Guide, Version 4.1
3. Rubrik Guidance Documentation, v. 1.3
4. Rubrik REST API for Rubrik 4.1, Version 1.0

1.3.2. Logical Scope

The TOE is comprised of several security features:

1. Security Audit
2. Identification and Authentication
3. Security Management
4. Cryptographic Support
5. Protection of the TSF

Each of the security features identified consists of several security functionalities, as identified below.

1.3.2.1. Security Audit

The Rubrik Cloud Data Management platform provides extensive auditing capabilities. The TOE generates a comprehensive set of audit logs that identify specific TOE operations. For each event, the TOE records the date and time of each event, the type of event, and the outcome of the event.

1.3.2.2. Identification and Authentication

The TOE provides authentication services for local and AD administrative users wishing to connect to the TOEs Secure Web UI or REST API administrator interface.

The TOE requires authorized administrators to authenticate prior to being granted access to any of the management functionality. After successful authentication, the TOE determines the permitted level of access for a user based on the local authorization setting for that user and provides role-based access.

When a Rubrik node joins a new AD domain, it temporarily obtains domain admin credentials to create a TOE service account in AD. The domain admin credentials are never stored on disk (but temporarily stored in memory) on the TOE.

1.3.2.3. Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure SSH or TLS/HTTPS session, or via a local console connection. The TOE supports an Administrator role.

There are two primary use cases that shape the default TOE RBAC roles: Administrator, and End User (who does not have administrative privileges).

1.3.2.4. Cryptographic Support

The TOE provides cryptography in support of remote administrative management via SSH and TLS/HTTPS.

TOE components of the Rubrik Nodes communicate with each other using TLS and SSH.

Sensitive data contained in the backups, both locally and archived across all mediums supported by the TOE, is encrypted.

1.3.2.5. Protection of the TSF

The TOE protects against interference and tampering by untrusted subjects by implementing identification, authentication, and access controls to limit configuration to only authorized administrators.

1.4. Notations and Formatting

The notations and formatting used in this ST are consistent with version 3.1 Revision 5 of the Common Criteria (CC).

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**. Deleted words are denoted by ~~strike-through text~~.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *italicized* text in square brackets, [*Selection value*].

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignment is indicated by showing the value with bold face in square brackets, [**Assignment_value**].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

Assets: Assets to be protected by the TOE are given names beginning with "AS." – e.g. AS.CLASSIFIED_INFO.

Assumptions: TOE security environment assumptions are given names beginning with "A."- e.g., A.Security_Procedures.

Threats: Threat agents are given names beginning with "TA." – e.g., TA.User. Threats to the TOE are given names beginning with "TT." – e.g., TT.Filter_Fails. TOE security environment threats are given names beginning with "TE."-- e.g., TE.Crypto_Fails.

Policies: TOE security environment policies are given names beginning with "P."—e.g., P.Information_AC.

Objectives: Security objectives for the TOE and the TOE environment are given names beginning with "O." and "OE.", respectively, - e.g., O.Filter-msg and OE.Clearance.

2. CC Conformance Claim (ASE_CCL)

This TOE and ST are conformant with the following specifications.

Item	Identification
CC Part 2	Security functional components, April 2017, Version 3.1, Revision 5, extended
CC Part 3	Security assurance components, April 2017, Version 3.1, Revision 5, conformant, EAL2 augmented with ALC_FLR.1
Assurance level	EAL2 augmented with ALC_FLR.1
Protection Profile	None
Package conformance	None
Extended SFRs	FAU_GEN_EXT.1

3. Security Problem Definition (ASE_SPD)

3.1. Threats to security

3.1.1. Assets

Assets	Description
AS.DATA	Sensitive or security functional data contained in TOE backups, both locally and archived across all mediums supported by the TOE.
AS.KEY	Cryptographic keys contained in the TOE, for encryption of ‘data in flight’.

3.1.2. Threat Agents

Threat Agents	Description
TA.ATTACKER	A person/company or process with skills and resources to mislead the system in any way necessary to reveal/divulge/misuse data and prevent the system from intended operations.
TA.ADMIN	Authorized person/process that performs installation and configuration/setup of the TOE to ensure that the TOE operates according to the needs of the enterprise/organization.

3.1.3. Identification of Threats

3.1.3.1. Threats to the TOE

Threats to the TOE	**Description**
TT.ADMIN_ERROR	The TOE may be incorrectly configured that may result in the TOE’s acquisition of ineffective security mechanisms.
Threat agent:	TA.ADMIN
Assets:	AS.DATA
Attack method:	During operation, the administrator unintentionally configures the TOE incorrectly, making the TOE inoperable or resulting in ineffective security mechanisms.
TT.ADMIN_EXPLOIT	A person/company may gain access to an administrator account.
Threat agent:	TA.ATTACKER
Assets:	AS.DATA
Attack method:	A person/company uses hacking methods to exploit missing, weak or incorrectly implemented access control in the TOE.
TT.CRYPTO_ COMPROMISE	An attacker may compromise cryptographic keys and the data protected by the cryptographic mechanisms.
Threat agent:	TA.ATTACKER
Assets:	AS.DATA and AS.KEY

Attack method:	An attacker cause key or data associated with the cryptographic functionality to be inappropriately accessed (viewed/modified/deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
TT.HACK_ACCESS	A person/company gets undetected system access to the TOE due to missing, weak and/or incorrectly implemented access control, causing potential violations of integrity, confidentiality or availability.
Threat agent:	TA.ATTACKER
Assets:	AS.DATA
Attack method:	A person/company uses hacking methods to exploit missing, weak or incorrectly implemented access control in the TOE.
TT.MALFUNCTION	The TOE may malfunction which may compromise information and data processing.
Threat agent:	TA.ATTACKER
Assets:	AS.DATA and AS.KEY
Attack method:	A malfunction in the TOE implies unauthorized access to TOE resources.

3.1.3.2. Threats to the TOE environment

Threats to the TOE environment	Description
TE.EAVESDROPPING	Eavesdropping of the communication between Rubrik nodes. This includes man-in-the-middle, side-channel, or other redirection attacks.
Threat agent:	TA.ATTACKER
Assets:	AS.DATA
Attack method:	An unauthorized person with no physical access to TOE is eavesdropping on the communication between Rubrik nodes to intercept information.

3.2. Organizational security Policies

Organizational security Policies	Description
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHIC	The TOE shall provide cryptographic functions for its own use, including encryption/decryption operations.

3.3. Assumptions

Assumptions	Description
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

A.TRUSTED_ADMIN	The administrators of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and shall follow the administrator guidelines.

4. Security objectives (ASE_OBJ)

This chapter defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats, comply with defined organizational security policies, and address applicable assumptions.

4.1. TOE Security Objectives

This section defines the security objectives that are to be addressed by the TOE.

Security Objectives	Description
O.ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate.
O.AUDIT	The TOE shall record and maintain security-related events to enable tracing of responsibilities for security-related acts and shall provide means to review the recorded data.
O.CRYPTOGRAPHY	The TOE shall provide cryptographic functions to maintain the confidentiality of 'data in flight'.
O.MANAGE	The TOE shall provide means for the administrators of the TOE to efficiently manage the TOE in a secure manner, and restrict these means from unauthorized use.
O.PROTECTION	The TOE must protect itself and its resources from unauthorized modifications and access to its functions and data.

4.2. Operational Environment Security Objectives

This section defines the security objectives that are to be addressed by the operational environment of the TOE.

Security Objectives	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	The administrator of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and shall follow the administrator guidelines as indicated by Rubrik and any additional trusted parties in which an agreement has been entered.

4.3. Security Objectives Rationale

The following tracing shows which security objectives address which threats, OSPs and assumptions.

Threats/ Policies/ Assumptions **Objectives**	TT.ADMIN_ERROR	TT.ADMIN_EXPLOIT	TT.CRYPTO_COMPROMISE	TT.HACK_ACCESS	TT.MALFUNCTION	TE.EAVESDROPPING	P.ACCOUNTABILITY	P.CRYPTOGRAPHIC	A.PHYSICAL	A.TRUSTED_ADMIN
TOE Security Objectives										
O.ACCESS		X		X	X		X			
O.AUDIT				X	X		X			
O.CRYPTOGRAPHY					X	X		X		
O.MANAGE	X	X		X		X				
O.PROTECTION			X	X	X					
Operational Environment Security Objectives										
OE.PHYSICAL									X	
OE.TRUSTED_ADMIN	X	X		X			X			X

Table 1: Mapping of Objectives to Threats, Policies and Assumptions

The following table is a set of justifications that shows that all threats, OSPs, and assumptions are effectively addressed by the security objectives.

Threat/Policy/Assumption	Security Objective Rationale
TT.ADMIN_ERROR	O.MANAGE provides administrators the capability to view and manage configuration settings. OE.TRUSTED_ADMIN ensures that the administrators are non-hostile and are trained to appropriately manage and administer the TOE.
TT.ADMIN_EXPLOIT	O.ACCESS includes mechanisms to authenticate TOE administrators and place controls on administrator sessions. O.MANAGE restricts access to administrative functions and management of TSF data to the administrator. OE.TRUSTED_ADMIN ensures that the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner.

TT.CRYPTO_ COMPROMISE	O.PROTECTION ensures that the TOE will have adequate protection from external sources and that all TOE Security Policy functions are invoked.
TT.HACK_ACCESS	O.AUDIT provides the TOE the capability to detect and create records of security-relevant events associated with users. O.ACCESS includes mechanisms to authenticate TOE administrators and place controls on administrator sessions. O.MANAGE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the administrator. These objectives ensure that no other user can modify the information flow policy to bypass the intended TOE security policy. O.PROTECTION ensures that the TOE will have adequate protection from external sources and that all TOE Security Policy functions are invoked. OE.TRUSTED_ADMIN ensures that the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner.
TT.MALFUNCTION	O.AUDIT provides the TOE the capability to detect and create records of security-relevant events associated with users. O.ACCESS includes mechanisms to authenticate TOE administrators and place controls on administrator sessions. O.CRYPTOGRAPHY requires the TOE to implement cryptographic services to provide confidentiality protection of data in flight. O.PROTECTION ensures that the TOE will have adequate protection from external sources and that all TOE Security Policy functions are invoked.
TE.EAVESDROPPING	O.CRYPTOGRAPHY requires the TOE to implement cryptographic services to provide confidentiality protection of data in flight. O.MANAGE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the administrator.
P.ACCOUNTABILITY	O.ACCESS requires the TOE to identify and authenticate users prior to allowing any TOE access or any TOE mediated access on behalf of those users O.AUDIT provides the administrator with the capability of recording the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's user identifier is recorded when any security relevant change is made to the TOE (e.g. modifying TSF data). OE.TRUSTED_ADMIN ensures that the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner.
P.CRYPTOGRAPHIC	O.CRYPTOGRAPHY requires the TOE to implement cryptographic

	services to provide confidentiality protection of the TOE.
A.PHYSICAL	OE.PHYSICAL ensures that the environment provides physical security, commensurate with the value of the TOE and the data it contains.
A.TRUSTED_ADMIN	OE.TRUSTED_ADMIN ensures that the administrators of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and shall follow the administrator guidelines.

Table 2: Rationale between Objectives and SPD

5. Extended Components Definition (ASE_ECD)

The following extended component has been included in this Security Target because the Common Criteria components were found to be insufficient as stated.

5.1. Extended Component

Explicit Component	Identifier	Rationale
FAU_GEN_EXT.1	Audit data generation	This extended component is necessary to describe that the TOE enables the audit functions during cluster initialization, and that the audit functions cannot be turned on or off during the TOE operation.

Table 3: Rationale for Extended Component

6. Security requirements (ASE_REQ)

6.1. Security Functional Requirements (SFRs)

Functional Class	Functional Component	
FAU: Security audit	FAU_GEN_EXT.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_STG.1	Protected audit trail storage
	FAU_SAR.1	Audit Review
FDP: User Data Protection	FDP_ACC.1	Subset Access Control
	FDP_ACF.1	Security attribute based access control
FCS: Cryptographic support	FCS_CKM.1	Cryptographic key generation
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1	Cryptographic operation
FIA: Identification and authentication	FIA_ATD.1	User attribute definition
	FIA_UAU.1	Timing of authentication
	FIA_UID.1	Timing of identification
FMT: Security management	FMT_MTD.1	Management of TSF data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Restrictions on security roles
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
FPT: Protection of the TSF	FPT_STM.1	Reliable time stamps

Table 4: Security Functional Requirements

6.1.1. Security Audit (FAU)

6.1.1.1. FAU_GEN_EXT.1 Audit data generation

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN_EXT.1.1 The TSF shall be able to generate an audit record of the following auditable events:
a) All auditable events for the [*not specified*] level of audit; and
b) [**All administrative actions and the following security events:**

- **Resuming all protection activity,**
- **Pausing all protection activity**].

FAU_GEN_EXT.1.2 The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**None**].

6.1.1.2. FAU_GEN.2 User identity association

Dependencies: FAU_GEN_EXT.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3. FAU_STG.1 Protected audit trail storage

Dependencies: FAU_GEN_EXT.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to [*prevent*] unauthorised modifications to the stored audit records in the audit trail.

6.1.1.4. FAU_SAR.1 Audit review

Dependencies: FAU_GEN_EXT.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide [**authorized administrator**] with the capability to read [**all information**] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

6.1.2. User Data Protection (FDP)

6.1.2.1. FDP_ACC.1 Subset Access Control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [**Administrator Access Control SFP**] on [subjects: **authorized administrator**, objects: **commands**, operations: **execute**].

6.1.2.2. FDP_ACF.1 Security attribute based access control

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the [**Administrator Access Control SFP**] to objects based on the following: [subjects: **authorized administrator**; subject attributes: **user name, password**; object: **commands**; object attributes: **none**].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**no additional rules**].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**no additional rules**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**no additional rules**].

6.1.3. Cryptographic Support (FCS)

6.1.3.1. FCS_CKM.1 Cryptographic key generation

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**TLS and SSH key generation**] and specified cryptographic key sizes [**as specified in Table 5**] that meet the following: [**NIST SP 800-135**].

6.1.3.2. FCS_CKM.4 Cryptographic key destruction

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**zeroization of keys**] that meets the following: [**FIPS PUB 140-2**].

6.1.3.3. FCS_COP.1 Cryptographic operation

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [**cryptographic operations listed in Table 5: Cryptographic Operations**] in accordance with a specified cryptographic algorithm [**listed in Table 5: Cryptographic Operations**] and cryptographic key sizes [**listed in**

Table 5: Cryptographic Operations] that meet the following: [**standards listed in Table 5: Cryptographic Operations**].

Cryptographic operations	Cryptographic algorithm	Key sizes (bits)	Standards
Encryption/decryption	AES	256	FIPS PUB 197
Encryption/decryption	RSA	2048	NA
TLS Session Keys Generation	TLS KDF	All TLS Session Key Sizes	NIST SP 800-135
SSH Session Key Generation	SSH KDF	All SSH Session Key Sizes	NIST SP 800-135
Hashing	SHA-1 SHA-256 SHA-384 SHA-512		FIPS PUB 180-4
HMAC	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	160, 256, 384, 512	FIPS PUB 198-1
Key Agreement	EC Diffie-Hellman	256, 384, 521	NA

Table 5: Cryptographic Operations

6.1.4. Identification and Authentication (FIA)

6.1.4.1. FIA_ATD.1 User attribute definition

Dependences: None.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) [User identity: **user name;**
b) Local authentication data: **password;**
c) Authorizations: **access rights;** and
d) **Email address**].

6.1.4.2. FIA_UAU.1 Timing of authentication

Dependences: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow [**entry of username and corresponding password**] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.4.3. FIA_UID.1 Timing of identification

Dependences: None.

FIA_UID.1.1 The TSF shall allow [**entry of username and corresponding password**] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

6.1.5. Security management (FMT)

6.1.5.1. FMT_MTD.1 Management of TSF data

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to [***manage***] the [**TSF data**] to [**authorized administrator**].

6.1.5.2. FMT_SMF.1 Specification of Management Functions

Dependencies: None.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [**administer the TOE locally and remotely**].

6.1.5.3. FMT_SMR.2 Restrictions on security roles

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles: [**Administrator, End User**].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions [**only the authorized administrator shall administer the TOE locally and remotely**] are satisfied.

6.1.5.4. FMT_MSA.1 Management of security attributes

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the [**Administrator Access Control SFP**] to restrict the ability to [*modify*] the security attributes [**in Administrator Access Control SFP**] to [**authorized administrator**].

6.1.5.5. FMT_MSA.3 Static attribute initialisation

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [**Administrator Access Control SFP**] to provide [*permissive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized administrator**] to specify alternative initial values to override the default values when an object or information is created.

6.1.6. Protection of the TSF (FPT)

6.1.6.1. FPT_STM.1 Reliable time stamps

Dependencies: None.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

6.2. Security assurance requirements (SARs)

The assurance level of the TOE is EAL2 augmented with ALC_FLR.1.

Assurance Class	Assurance Components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_FLR.1 Basic Flaw Remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

Table 6: Assurance requirements

6.3. Security Requirements Rationale

6.3.1. Relation between SFRs and security objectives

The following tracing shows which SFRs address which security objectives for the TOE.

Requirements Objectives	FAU_GEN_EXT.1	FAU_GEN.2	FAU_STG.1	FAU_SAR.1	FDP_ACC.1	FDP_ACF.1	FCS_CKM.1	FCS_CKM.4	FCS_COP.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MTD.1	FMT_SMF.1	FMT_SMR.2	FMT_MSA.1	FMT_MSA.3	FPT_STM.1
O.ACCESS					X	X				X	X	X				X	X	
O.AUDIT	X	X		X														X
O.CRYPTOGRAPHY							X	X	X									
O.MANAGE				X									X	X	X	X	X	
O.PROTECTION			X		X	X							X	X	X	X		

Table 7: Tracing of functional requirements to Objectives

The following set of justifications shows that all security objectives for the TOE are effectively addressed by the SFRs.

Security Objectives	Security Functional Requirement Rationale
O.ACCESS	FIA_ATD.1 defines the attributes of users, including a user identifier that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE, and ensures that untrusted users cannot be associated with a role and reduces the possibility of a user obtaining administrative privileges. FIA_UAU.1 ensures that users are authenticated before they are provided access to the TOE or its services. In order to control logical access to the TOE an authentication mechanism is required. The local user authentication mechanism is necessary to ensure that an administrator has the ability to login to the TOE regardless of network connectivity (e.g., it would be unacceptable if an administrator could not login to the TOE because the authentication server was down, or that the network path to the authentication server was unavailable). FIA_UID.1 ensures that every user is identified before the TOE performs any mediated functions. FMT_MSA.3 defines static attribute initialization for the Administrator Access Control SFP. FMT_MSA.1 specifies which roles can access security attributes. FDP_ACC.1 requires the TOE to enforce Access Control SFP. FDP_ACF.1 specifies the attributes used to enforce Access Control SFP.
O.AUDIT	FAU_GEN_EXT.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security relevant event that takes place in the TOE.

	FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the user ID. FAU_SAR.1 provides administrators the capability to read the audit records. FPT_STM.1 supports the audit functionality by ensuring that the TOE is capable of obtaining a time stamp for use in recording audit events.
O.CRYPTOGRAPHY	FCS_CKM.1 ensures that the TOE is capable of generating cryptographic keys. FCS_CKM.4 provides the functionality for ensuring that keys and key material is zeroized. FCS_COP.1 requires that for data decryption and encryption an approved algorithm is used, and that the algorithm meets the standard.
O.MANAGE	The FMT requirements are used to satisfy this management objective, as well as other objectives that specify the control of functionality. The requirement's rationale for this objective focuses on the administrator's capability to perform management functions in order to control the behavior of security functions. FMT_MTD.1, FMT_SMF.1 and FMT_SMR.2 ensure that only the Administrator role can manage the entire TOE, and that the TOE supports both local administration and remote administration. FAU_SAR.1 provides the administrators the capability to read all information from the audit records. FMT_MSA.3 defines static attribute initialization for the Administrator Access Control SFP. FMT_MSA.1 specifies which roles can access security attributes.
O.PROTECTION	FAU_STG.1 requires the TOE to protect the audit data from deletion and modification. FMT_MTD.1, FMT_SMF.1 and FMT_SMR.2 ensure that only authorized administrators of the TOE may manage the TOE and TSF data. FMT_MSA.1specifies which roles can access security attributes. FDP_ACC.1 requires the TOE to enforce Access Control SFP. FDP_ACF.1 specifies the attributes used to enforce Access Control SFP.

Table 8: Rationale between Objectives and SFRs

6.3.2. SFR Dependencies

The table below shows the dependencies of the security functional requirements of the TOE and gives a rationale for each of them if they are included or not.

Security functional requirement	Dependency	Dependency Rationale
FAU_GEN_EXT.1 Audit data generation	FPT_STM.1 Reliable time stamps	Included
FAU_GEN.2 User identity association	FAU_GEN_EXT.1 Audit data generation FIA_UID.1 Timing of identification	Included
FAU_STG.1 Protected audit trail storage	FAU_GEN_EXT.1 Audit data generation	Included
FAU_SAR.1 Audit review	FAU_GEN_EXT.1 Audit data generation	Included
FDP_ACC.1 Subset Access Control	FDP_ACF.1 Security attribute based access control	Included
FDP_ACF.1 Security attribute based access control	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Included
FCS_CKM.1 Cryptographic key generation	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Included
FCS_CKM.4 Cryptographic key destruction	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	Included
FCS_COP.1 Cryptographic Operation	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Included
FIA_ATD.1 User attribute definition	None	
FIA_UAU.1 Timing of authentication	FIA_UID.1 Timing of identification	Included
FIA_UID.1 Timing of identification	None	

FMT_MTD.1 Management of TSF data	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Included[1]
FMT_SMF.1 Specification of Management Functions	None	
FMT_SMR.2 Restrictions on security roles	FIA_UID.1 Timing of identification	Included
FMT_MSA.1 Management of security attributes	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Included[2]
FMT_MSA.3 Static attribute initialisation	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Included[2]
FPT_STM.1 Reliable time stamps	None	

Table 9: SFR's dependencies and rationale

6.3.3. SAR RATIONALE

The SARs specified in this ST are according to EAL2, augmented with ALC_FLR.1.

[1] FMT_MTD.1 has a dependency to FMT_SMR.1 which is covered by FMT_SMR.2.

[2] FMT_MSA.1 and FMT_MSA.3 have a dependency to FMT_SMR.1 which is covered by FMT_SMR.2

7. TOE SUMMARY SPECIFICATION (ASE_TSS)

7.1. TOE SECURITY FUNCTIONS SPECIFICATION

This section describes the security functions provided by the TOE to meet the security functional requirements specified for the TOE in section 6.1 Security Functional Requirements (SFRs).

The table below shows the mapping between the SFRs and the implementing security functions, and a description is given in the following subsections.

Requirements Functions	FAU_GEN_EXT.1	FAU_GEN.2	FAU_STG.1	FAU_SAR.1	FDP_ACC.1	FDP_ACF.1	FCS_CKM.1	FCS_CKM.4	FCS_COP.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MTD.1	FMT_SMF.1	FMT_SMR.2	FMT_MSA.1	FMT_MSA.3	FPT_STM.1
SF.TOE_ACCESS_FUNCTIONS					X	X				X	X	X				X	X	
SF.SECURITY_AUDIT	X	X	X	X														X
SF.CRYPTOGRAPHIC_SUPPORT							X	X	X									
SF.SECURITY_MANAGEMENT				X									X	X	X	X	X	

Table 10: Mapping SFRs to security functions

7.1.1. SF. TOE_ACCESS_FUNCTIONS

(FIA_ATD.1, FMT_MSA.1, FMT_MSA.3)

User account information is stored in the TOE and contains the following attributes for local users:

- User name – The logon name of the user.
- Email address – The valid email address for the user.
- Password – The user must use a strong password.
- Retype password – The user must enter the password again to rule out typographical errors in the password.
- Access rights - By default, the TOE sets all new local user accounts to the Administrator authorization level.

The TOE provides permissive default values for security attributes. The TOE allows the authorized administrator to specify alternative initial values. The ability to modify the security attributes is restricted to the authorized administrator.

(FIA_UAU.1, FIA_UID.1, FDP_ACC.1, FDP_ACF.1)

Each individual must be successfully identified and authenticated with a username and password by the TSF before access is allowed to the TOE. User identification and authentication by the TSF uses the security attributes of the user account described above. When identification and authentication data is entered, the TOE attempts to identify the applicable user account from the provided identity and if a match is found, the password provided is compared against that stored with the user account information in the internal database or AD server. If a user account cannot be associated with the provided identity or the provided password does not match that stored with the user account information, identification and authentication will fail. No actions are allowed, other than entry of identification and authentication data, until successful identification and authentication. The TOE offers the following authorization levels:

- Administrator – Provides the user with full access to all functionality that is available through the Web UI.
- No Access – Acknowledges the authentication at the login screen but prohibits access to the Web UI.
- End User – Provides the user with access to assigned objects.

7.1.2. SF. SECURITY_AUDIT

(FAU_GEN_EXT.1)
The TOE generates a comprehensive set of audit logs that identify specific TOE operations whenever an auditable event occurs. Potentially time-sensitive notifications and completed replication tasks are recorded.

(FAU_GEN.2)
The TOE ensures that each auditable event is associated with the user that triggered the event. For an IT entity or device, the VM name of the endpoint is included in the audit record.

(FPT_STM.1)
The TOE provides a source of date and time information used in audit event timestamps, receiving clock updates from an NTP server.

(FAU_STG.1)
The TOE stores the audit records locally in a limited logging buffer, and protects the records from deletion and modification.

(FAU_SAR.1)
The administrators are allowed to read the audit records, but they have no other access privileges to the buffer containing the audit log.

7.1.3. SF.CRYPTOGRAPHIC_SUPPORT

(FCS_CKM.1)
In support of secure cryptographic protocols, the TOE supports the key generation schemes used by TLS and SSH as specified in NIST SP 800-135. The TOE is fully compliant to SP 800-135.

(FCS_CKM.4)
The TOE meets all requirements specified in FIPS 140-2 for destruction of keys. All keys within the TOE are zeroizable.

(FCS_COP.1)
The TOE provides encryption and decryption capabilities

- using 256 bits AES, described in FIPS PUB 197,
- using 2048 bits RSA,

The TOE provides key generation capabilities

- using TLS, described in NIST SP 800-135,
- using SSH, described in NIST SP 800-135,

The TOE provides Hashing capabilities

- using SHA-1, SHA-256, SHA-384, and SHA-512 described in FIPS PUB 180-4.

The TOE provides HMAC capabilities

- using HMAC-SHA-1, described in FIPS PUB 198-1,
- using HMAC-SHA-256, described in FIPS PUB 198-1,
- using HMAC-SHA-384, described in FIPS PUB 198-1,
- using HMAC-SHA-512, described in FIPS PUB 198-1.

The TOE provides Key Agreement capabilities

- using 256, 384 and 521 bits ECDH.

7.1.4. SF.SECURITY_MANAGEMENT

(FMT_MTD.1, FMT_MSA.1, FMT_MSA.3)

The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data and updates. Each of the predefined access right levels has a set of permissions that will grant them access to the TOE data. For the purposes of this evaluation, the ”Administrator” privileged level is equivalent to full administrative access to the local interface, SSH, Web UI or REST API. The term “authorized administrator” is used in this ST to refer to any user which has been assigned to an access right level that is permitted to perform the relevant action. The TOE provides permissive default values for security attributes. The TOE allows the authorized administrator to specify alternative initial values. The ability to modify the security attributes is restricted to the authorized administrator.

(FMT_SMF.1)

The TOE provides all the capabilities necessary to securely manage the TOE. The administrative user can connect to the TOE either locally, through the SSH, TOE Web UI or REST API to perform these functions. However, the specific configurable parameters available through the TOE locally are limited. All general administration is expected to take place through the Web UI or using the SSH protocol. The specific management capabilities available from the TOE include:

- Local and remote administration of the TOE services and security characteristics;
- Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE via the Web UI.

(FMT_SMR.2)

The term “authorized administrator” is used in this ST to refer to any user which has been assigned to a privilege level that is permitted to perform the relevant action and therefore has the appropriate privileges to perform the requested functions. The TOE authenticates all access to the administrative interfaces using a username and password. The TOE supports both local administration and remote administration, only by means of the authorized administrator.

The TOE supports the following RBAC roles:

- Administrator: manages users and shall have full TOE access.
- End User: can explore backups of VMs, find and recover files.

(FAU_SAR.1)

The administrators are allowed to read the audit records, but they have no other access privileges to the buffer containing the audit log.