**Real Time Data Processing and Storage**

**And Storage**

*A project report submitted*

*to*

**MANIPAL ACADEMY OF HIGHER EDUCATION**

*For Partial Fulfillment of the Requirement for the*

*Award of the Degree*

*of*

**Bachelor of Technology**

*in*

**Information Technology**

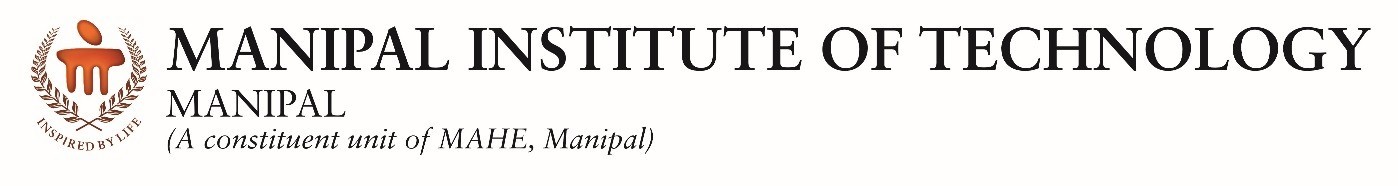
*by*

**Kaushal Sinha**

**Reg. No. 140911382**

*Under the guidance of*

|  |  |
| --- | --- |
| Mr. Chethan Sharma | Mrs. Malini Venkataraman |
| Professor | Lead IOT Architect |
| Department of I & CT | Professional Services |
| Manipal Institute of Technology | Honeywell International Inc. |
| Manipal, India | Bangalore, India |



**July 2018**

I dedicate my thesis to my friends and family who have always been a constant source of support and encouragement during the challenges of my whole college life. This project work is also dedicated to my mentors, who have been my source of inspiration and continually provided guidance both

morally and technically.

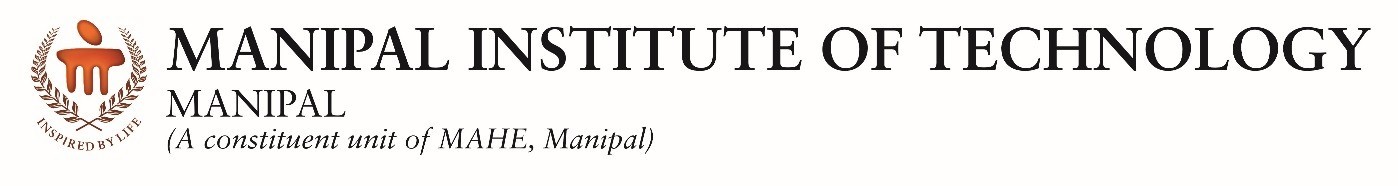
**DECLARATION**

I hereby declare that this project work entitled **Real Time Data Processing and Storage** is original and has been carried out by me in Honeywell Technology Solutions Pvt. Ltd., Bangalore and Department of Information and Communication Technology of Manipal Institute of Technology, Manipal, under the guidance of **Malini Venkataraman, Lead IOT Architect**, Honeywell Technology Solutions, Bangalore and **Chethan Sharma**, **Professor**, Department of Information and Communication Technology, M. I. T., Manipal. No part of this work has been submitted for the award of a degree or diploma either to this University or to any other Universities.

Place: Bangalore

Date :20-07-18

Kaushal Sinha



**CERTIFICATE**

This is to certify that this project entitled **A Proactive Health Monitoring Tool for Software Defined Datacenters** is a bonafide project work done by Mr. Nikhil Gupta at Manipal Institute of Technology, Manipal, independently under my guidance and supervision for the award of the Degree of

Bachelor of Technology in Information Technology.

|  |  |
| --- | --- |
| Dr. Raghvendra Achar | Dr.Balachandra |
| Associate Professor | Professor & Head |
| Department of I & CT | Department of I & CT |
| Manipal Institute of Technology | Manipal Institute of Technology |
| Manipal, India | Manipal, India |

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my mentor Keerthi Kumar and my manager Priya Pandian here at VMware as well as Raghvendra Achar back at college for providing their invaluable guidance, comments and suggestions throughout the course of the project. I would also like to thank Dr. Balachandra, HOD, I&CT Department, Manipal Institute of Technology, Manipal for providing me the opportunity to undergo this project. This project would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

# ABSTRACT

The world has more data than it knows what to do with. Industrial Internet of

Things (IIoT) sensor technologies are collecting and sending unparalleled

amounts of data. But the challenge is turning this big data into action. Time-

series database performs data analysis and anomaly detection among other

functions. These databases can scale easily, integrate with multiple data

sources and software, and process more data, quicker than ever.

At the core of the latest offerings is the Honeywell Sentience IoT platform that

delivers robust and secure big data capabilities for all of Honeywell’s

connected solutions, demonstrating Honeywell’s vision for its overarching

software platform and ecosystem.

A real-time data processing system takes input of rapidly changing data and

to provide near-instantaneous output to ensure everything is up to date in

such a system. Real-time data is generated from factory devices and are

configured to work with Honeywell sentience. Each device needs to be

registered first so that it can communicate with sentience.

Combining advanced automation and data analytics, Outcome Based Service

enables Honeywell service engineers to assess and scrutinize building assets

around the clock, promoting the identification of anomalies and

misconfigurations earlier than traditional maintenance and helping save time.

Without close monitoring, energy consumption can drift by up to 7 percent

annually. The service taps building connectivity and sensors with the goal of

ensuring that building equipment is performing at its best. Results from early

pilot deployments have shown it can help organizations cut energy spend

while cutting reactive service calls.

**Contents**

[**Acknowledgements** **iv**](#_Toc30742)

[**Abstract** **v**](#_Toc30743)

[**List of Figures** **x**](#_Toc30744)

[**Abbreviations** **x**](#_Toc30745)

[**Notations** **xi**](#_Toc30746)

[**1 Introduction** **1**](#_Toc30747)

[1.1 Present Day Situation 1](#_Toc30748)

[1.2 Problem Definition and Motivation 2](#_Toc30749)

[1.3 Areas of Work 2](#_Toc30750)

[1.3.1 Cloud Computing 2](#_Toc30751)

[1.3.2 Software Engineering 3](#_Toc30752)

[1.3.3 Data Science 3](#_Toc30753)

[1.4 Project Objectives 3](#_Toc30754)

[1.5 Target Specification 4](#_Toc30755)

[**2 Literature Survey** **5**](#_Toc30756)

[2.1 Virtualization Technology 5](#_Toc30757)

[2.2 VMware Products 6](#_Toc30758)

[2.2.1 VMware ESXi 6](#_Toc30759)

[2.2.2 vCenter Server 7](#_Toc30760)

[2.2.3 vSAN 8](#_Toc30761)

[2.2.4 vMotion 9](#_Toc30762)

[2.3 Statistical Analysis and Machine Learning 9](#_Toc30763)

[2.3.1 Control Charts 10](#_Toc30764)

[2.3.2 K-Means 12](#_Toc30765)

[2.3.3 Kernel Density Estimation 12](#_Toc30766)

[2.4 Introduction to the Project Title 15](#_Toc30767)

[2.4.1 Reactive Monitoring 15](#_Toc30768)

[2.4.2 Proactive Monitoring 15](#_Toc30769)

[**3 Methodology** **17**](#_Toc30770)

[3.1 Software Architecture 17](#_Toc30771)

[3.1.1 Data Collector 19](#_Toc30772)

[3.1.2 The Database 20](#_Toc30773)

[3.1.3 POP Reverse Proxy 20](#_Toc30774)

[3.1.4 Processing Appliance 21](#_Toc30775)

[3.2 Data Analysis and Error Prediction 22](#_Toc30776)

[3.2.1 Clustering 22](#_Toc30777)

[3.2.2 Kernel Density Estimation 24](#_Toc30778)

[3.3 Remediation and Troubleshooting Service 26](#_Toc30779)

[3.3.1 RTS Design 27](#_Toc30780)

[3.3.2 Remediation procedure through RTS 28](#_Toc30781)

[3.3.2.1 Example scenario 31](#_Toc30782)

[**4 Results** **34**](#_Toc30783)

[**5 Conclusion and Future Work** **36**](#_Toc30784)

[5.1 Conclusion 36](#_Toc30785)

[5.2 Future Work 36](#_Toc30786)

[**References** **38**](#_Toc30787)

**ProjectDetail 38**

# List of Figures

|  |  |
| --- | --- |
| 2.1 VMware ESXi . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| 2.2 vCenter Server Appliance hierarchy . . . . . . . . . . . . . . . | 7 |
| 2.3 Hyperconverged vSAN system . . . . . . . . . . . . . . . . . . | 8 |
| 2.4 vMotion in action . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
| 2.5 Types of machine learning . . . . . . . . . . . . . . . . . . . . | 10 |
| 2.6 Control chart . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
| 2.7 Normal distribution in data . . . . . . . . . . . . . . . . . . . | 11 |
| 2.8 K-means in action . . . . . . . . . . . . . . . . . . . . . . . . . | 13 |
| 2.9 Kernel Density Estimation . . . . . . . . . . . . . . . . . . . . | 14 |
| 2.10 Bandwidth in KDE . . . . . . . . . . . . . . . . . . . . . . . . | 14 |
| 3.1 Software Architecture . . . . . . . . . . . . . . . . . . . . . . . | 18 |
| 3.2 Layers evaluated from high impact to low impact components | 18 |
| 3.3 Software Workflow . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
| 3.4 Reverse Proxy . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
| 3.5 Repetitive K-Means algorithm . . . . . . . . . . . . . . . . . . | 23 |
| 3.6 Kernel Density Estimation Flow . . . . . . . . . . . . . . . . . | 25 |
| 3.7 RTS Architecture . . . . . . . . . . . . . . . . . . . . . . . . . | 29 |
| 3.8 Log Collection Interface . . . . . . . . . . . . . . . . . . . . . | 30 |
| 3.9 vSan Scripts . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 31 |
| 3.10 Sample execution output . . . . . . . . . . . . . . . . . . . . . | 32 |
| 3.11 vSan Runbooks . . . . . . . . . . . . . . . . . . . . . . . . . . | 32 |
| 3.12 A runbook to check the overall physical disk health . . . . . . | 33 |

# ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| API | : | Application Programming Interface |
| AWS | : | Amazon Web Services |
| CPU | : | Central Processing Unit |
| EC2 | : | Elastic Compute Cloud |
| ESX | : | Elastic Sky X |
| JSON | : | JavaScript Object Notation |
| KDE | : | Kernel Density Estimation |
| LCL | : | Lower Control Limit |
| PoP | : | Point of Presence |
| PSC | : | Platform Service Controller |
| REST | : | Representational State Transfer |
| RTS | : | Remediation and Troubleshooting Service |
| SAN | : | Storage Area Network |
| SDDC | : | Software Defined Datacenter |
| SCP | : | Secure Copy |
| SRE | : | Site Reliability Engineering |
| SSH | : | Secure Shell |
| UCL | : | Upper Control Limit |
| UI | : | User Interface |
| VC | : | vCenter |
| VCSA | : | VMware vCenter Server Virtual Appliance |
| VMC | : | VMware Cloud |
| VSI | : | VMware Sys Info |

# NOTATIONS

*µ* : Mean

*σ* : Standard deviation

**Chapter 1**

# Introduction

This chapter briefly describes the project by expanding on a few basic but important points the use case of the project from the company’s perspective, the motivation to carry the project out and the objective it focuses on achieving, the consequences that will follow and a timeline of how the project proceeded.

## 1.1 Present Day Situation

## Honeywell Building Solutions is a part of Honeywell Home and Building

## Technologies (HBT), a global business with more than 44,000 employees

## worldwide. HBT is a leader in the Internet of Things (IoT) and creates

## products, software and technologies found in more than 150 million homes

## and 10 million buildings worldwide. We help homeowners stay connected and

## in control of their comfort, security and energy use. Commercial building

## owners and occupants use our technologies to ensure their facilities are safe,

## energy efficient, sustainable and productive. Our advanced metering

## hardware and software solutions help electricity, gas and water providers

## supply customers and communities more efficiently.

## 1.2 Problem Definition and Motivation

Since log analysis to find errors is a time-consuming process, manual resolution of erroneous incidents poses a threat of crossing the time limits specified in Service Level Agreements, there is a definite need of a software that can detect system problems beforehand to some degree, by proactively analyzing some critical performance metrics of the systems in an SDDC and reporting any anomalous behavior in the analyzed system components.

Internet of Things (IoT) makes our world as possible as connected together. Nowadays we almost have internet infrastructure wherever and we can use it whenever. Embedded computing devices would be exposed to internet influence. Common instances for embedded computing devices are MP3 players, MRI, traffic lights, microwave ovens, washing machines and dishwashers, GPS even heart monitoring implants or biochip and etc.

IoT tries to establish advanced connectivity (with the aid of internet) among these mentioned device or systems or services in order to little by little makes automation in all areas. Image that all thing are connected to gather and all information would be interacted to each other over standard and different protocol domain and applications.

Recent researches shows by 2020 we have over 20 billion devices which uses IoT. IoT does that because of controlling on device and lower expense on radio. But these huge fields make challenges such as lacking IP address, developing compatible and useful protocol and environment.

Automation is hence increasingly important over the lifetime of the product since it ensures scalability into a bigger market. It is an effective way to continue services without having to acquire extra human resources.

## 1.3 Areas of Work

The project spans across a multitude of computer science subjects and uses several state-of-the-art technological tools and concepts. Some of the major concepts that are extensively involved are

### 1.3.1 Cloud Computing

The main theme of not only the project, but the company’s business orientation is cloud computing. At a time when all companies are shifting towards cloudbased data centers to support their business operations in order to reduce costs, Honeywell’s latest projects are naturally aligned to the trend. This project is built for data centers deployed on the cloud exclusively, and an understanding of how the cloud environment works was a must to get started.

### 1.3.2 Software Engineering

To build a robust software, it is necessary to follow some cardinal guidelines set forth by software engineering principles. Without a proper structure, it becomes increasingly difficult to maintain code as the size of the codebase starts to grow. It was therefore important to make sure that the foundation of the software was flexible enough to accommodate changes and allow for incremental additions to the code.

### 1.3.3 Data Science

Data is the backbone of any monitoring system; Without the right metrics and logs, it becomes difficult to evaluate the performance of data centers. Acquisition of data from multiple sources and filtering it out for consumable information was a challenge. Once the data was acquired, it was followed by analytical studies

## 1.4 Project Objectives

To get real time data from factory devices and sensors. We try to establish advanced connectivity (with the aid of internet) among these mentioned device or systems or services to make automation in all areas.

* First, instruments and control devices, including operator stations are cloud enabled or connected to the cloud via gateways.
* Finding a way to collect data without causing a lot of stress on the hypervisors memory and network bandwidth.
* Organizing received metrics in a standardized structure that can then be read with ease from the database and analyzed effectively.
* Using tsdb (time series data base) for storing the real-time data. We use influxDb for this purpose.
* Have another history database which stores previous data points. We use azure cosmos database to store history data.
* Provide API’s to query from both influxDb and cosmos. The Api for influxDb is capable of getting min, max and average values from the real time database.
* The analytics team is given endpoints to access the influxDb for real time analytics.
* With this working efficiently, any faulty device or sensor in a factory can be easily detected. On the event of a fault detection, a command can be given from the cloud to initiate the process of repair/removal.

## 1.5 Target Specification

Using diagnostic tools and Honeywell’s network of localized technicians, the service focuses on targeted interventions to help facilities more easily identify building problems and improvement opportunities that can drive energy savings along with operational and comfort improvements.

**Chapter 2**

# Literature Survey

## 2.1 Industrial IoT Technology

The IIoT is part of a larger concept known as the Internet of Things (IoT). The IoT is a network of intelligent computers, devices, and objects that collect and share huge amounts of data. The collected data is sent to a central Cloud-based service where it is aggregated with other data and then shared with end users in a helpful way. The IoT will increase automation in homes, schools, stores, and in many industries.

The application of the IoT to the manufacturing industry is called the IIoT (or Industrial Internet or Industry 4.0). The IIoT will revolutionize manufacturing by enabling the acquisition and accessibility of far greater amounts of data, at far greater speeds, and far more efficiently than before.

## 2.2 Honeywell Products

This section covers a brief introduction of the various products [1] being used during the project.

### 2.2.1 Sentience Edge

ControlEdgePLC dramatically reduces configuration, integration, and support costs while decreasing risk with embedded cyber security. Combined with Experion PKS, it minimizes downtime through unified support, and lowers total cost of ownership through extended system lifecycle.  
  
The latest release provides the ultimate engineering flexibility with more I/O options for system design, connectivity to more devices, the most protocols for native connectivity, and support for lean project execution. EtherNet/IP, a serial interface module, and cloud enabled engineering enhance the first PLC to have earned ISASecure Level 2 certification.integrates vital OS components, such as a kernel.



Figure 2.1: Honeywell Edge

Built on Android, Honeywell's Mobility Edge platform is an integrated, scalable approach based on common software and hardware tools that drive an integrated, repeatable, scalable approach for accelerated and secure development, deployment, performance management and lifecycle management.

### 2.2.2 Cloud Connector

Honeywell Sentience Edge C++ Cloud Connector enables a developer to securely exchange data with the Honeywell Sentience IoT platform. The C++ Cloud Connector offers C++ APIs for various services that includes data transport to the Honeywell Sentience IoT platform, firmware download, bulk data upload, authentication, provisioning, and management.

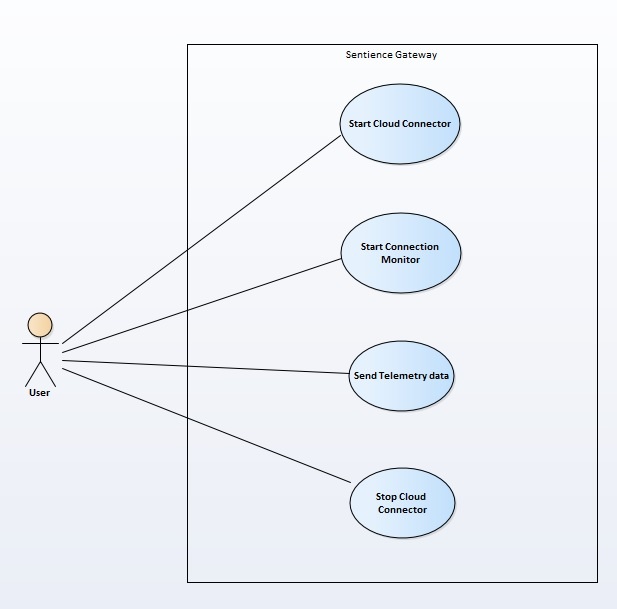


Figure 2.2: use case realization

The C++ SDKprovides a library which can be used to develop a C++ based sample device. This SDK consists of core services which can open and maintain the Honeywell Sentience Platform connection.

### 2.2.3 Cloud Platform //justify

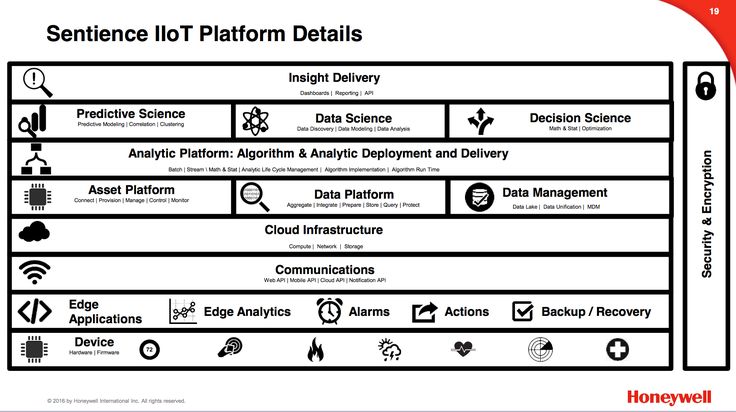
The Sentience Cloud provides a Platform for app developers wanting to build mobile, web and cloud based analytic offerings on top of IoT devices. Sentience creates common environments, services, and libraries for our IoT software, which connects the products our customers use daily to the Internet – allowing them to send, receive & analyze data. The goal of the Sentience Platform is to provide a productive development experience for app developers wanting to build mobile, web and cloud based analytic offerings on top of IoT devices. Sentience uses open messaging protocols combined with micro-services that present REST APIs or Java Client libraries based around the key IoT concepts of equipment, spaces and people. These APIs provide read and write of device values, history collection and query of received device data, alarm and event storage and forwarding, device management including firmware updates and a graph based store for storing rich data models. In addition, and Analytics platform is provided where the developer is able to use the IoT data and context data to drive business analytics.

Figure 2.3: Sentience Platform

Sentience analytics provides a set of key capabilities to analyze large volumes of information, derive insights, and enable applications to take required actions. Making sense of IoT data requires fast and efficient analytics, including but not limited to handling big data. Analytics capabilities provided by the Sentience IoT platform are Stream Processing, Batch Processing and Machine Learning.

## 2.3 Statistical Analysis and Machine Learning

If you use an online service to send email, edit documents, watch movies or TV, listen to music, play games or store pictures and other files, it is likely that cloud computing is making it all possible behind the scenes. The first cloud computing services are barely a decade old, but already a variety of organizations—from tiny startups to global corporations, government agencies to non-profits—are embracing the technology for all sorts of reasons. Cloud computing is a big shift from the traditional way businesses think about IT resources. Most cloud computing services fall into three broad categories: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). These are sometimes called the cloud computing stack, because they build on top of one another. Knowing what they are and how they are different makes it easier to accomplish your business goals.

### 2.3.1 Infrastructure-as-a-service (IaaS)

The most basic category of cloud computing services. With IaaS, you rent IT infrastructure—servers and virtual machines (VMs), storage, networks, operating systems—from a cloud provider on a pay-as-you-go basis. The quality range is typically calculated using standard deviations and averages of historical data.

### 2.3.2 Platform as a service (PaaS)

K-means clustering is a type of unsupervised learning, which is used when you Platform-as-a-service (PaaS) refers to cloud computing services that supply an on-demand environment for developing, testing, delivering and managing software applications. PaaS is designed to make it easier for developers to quickly create web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development.

### 2.3.3 Software as a service (SaaS)

Software-as-a-service (SaaS) is a method for delivering software applications over the Internet, on demand and typically on a subscription basis. With SaaS, cloud providers host and manage the software application and underlying infrastructure and handle any maintenance, like software upgrades and security patching. Users connect to the application over the Internet, usually with a web browser on their phone, tablet or PC.

## 2.4 Bamboo

CI is a software development methodology in which a build, unit tests and integration tests are performed, or triggered, whenever code is committed to the repository, to ensure that new changes integrate well into the existing code base. Integration builds provide early 'fail fast' feedback on the quality of new changes. Release management describes the steps that are typically performed to release a software application, including building and functional testing, tagging releases, assigning versions, and deploying and activating the new version in production.

### 2.4.1 Build

Focus on coding and count on Bamboo as your CI and build server! Create multi-stage build plans, set up triggers to start builds upon commits, and assign agents to your critical builds and deployments.

### 2.4.2 Test

Testing is a key part of continuous integration. Run automated tests in Bamboo to regress your products thoroughly with each change. Parallel automated tests unleash the power of Agile Development and make catching bugs easier and faster.

### 2.4.3 Deploy

Bamboo offers first-class support for the "delivery" aspect of continuous delivery. Deployment projects automate the tedium right out of releasing into each environment, while letting you control the flow with per-environment permissions.

**Chapter 3**

# Methodology

The first part of this chapter talks about the architecture of the software. This is subsequently followed by detailed implementation level information of every single component present in the architecture.

## 3.1 Software Architecture

Any given software is a collection of smaller modules that serve their own purpose. This project is no different. The software can be broken down into smaller functional units that when put together achieve the desired objectives. An overall structure of the project is given below in Figure 3.1.

The software is implemented on a layered basis with the lowest layers corresponding to the physical devices and kernel functions and the upper layers corresponding to services and products that depend on these lower layers. Identifying these separate layers are subject to experimentation and following the kernel code.

As an example of layers, consider the edge platform. Edge Platform reduces the complexity to build, deploy, manage, and extend the lifecycle of mobile solutions—and it does it all better, faster, with less risk, and with lower costs than how you were able to manage enterprise mobility in the past.

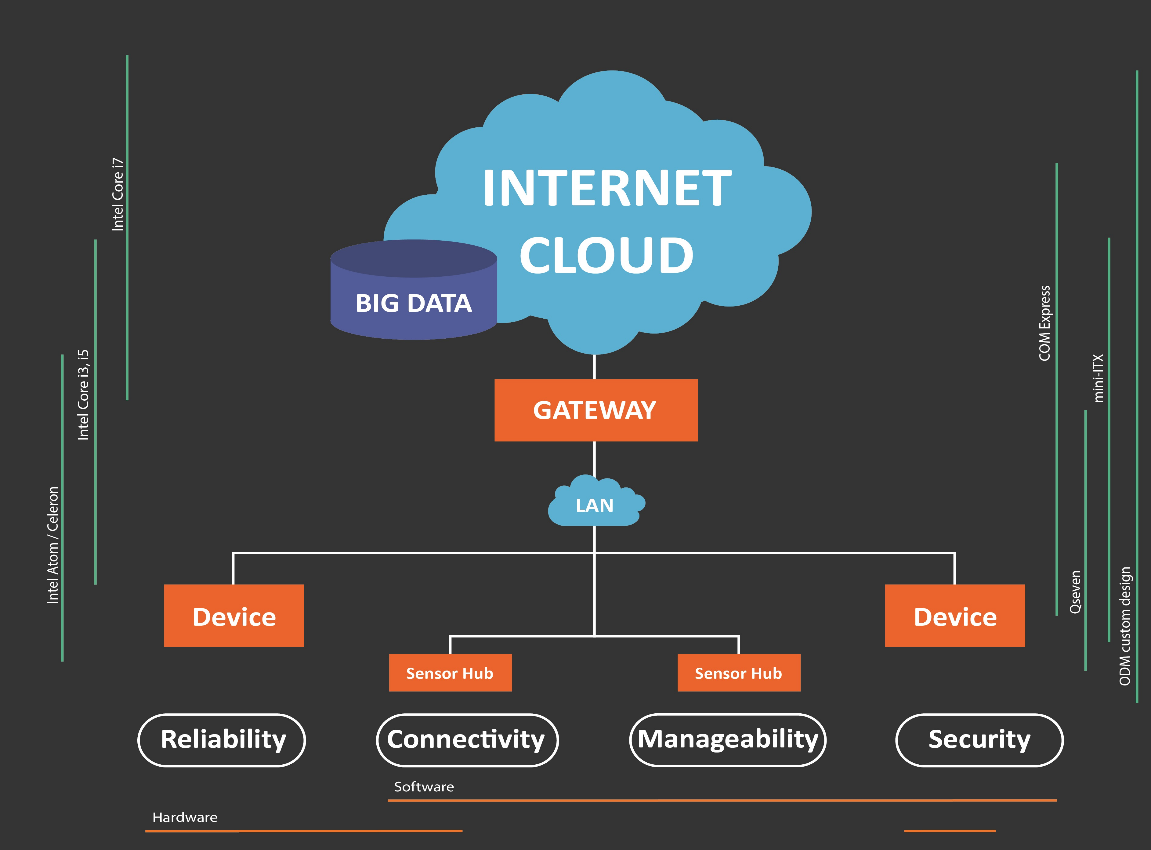


Figure 3.1: Architecture

As the figure shows, the data from different sources such as sensors, security devices, industrial devices pass through the LA, then reaches the edge gateway which is configured to talk to a particular server.

### 3.1.1 Data Collector

Each SDDC contains various software defined hosts. The health of the SDDC itself As the figure shows, the data from different sources such as sensors, security devices, industrial devices passes through the LA, then reaches the edge system for their metrics [7]. Since the hosts run ESXi hypervisors on them, it is impossible to run any new software on them. The developers want the ESXi to have as less bloat as possible, since the hypervisor runs on RAM and not on permanent storage.

Data collection can hence only be carried out using the tools that are already built into the hypervisor. Some examples of these tools are

* **VMKernel Sys Info Shell (Vsish)** This is an extremely important tool that keeps track of kernel level logs and can be easily read by a python library called vsi. Vsish is built into all ESXi versions and gives us direct information related to hardware devices and services. This information is stored into nodes and is updated at a very high frequency.
* **Tcpdump-uw** A barebones implementation of the tcpdump command normally found on Linux distributions. For any network related information, a tcpdump is taken for a given set duration and its characteristics are analyzed. Some features like the number of duplicate packets and the round-trip time give us accurate information about the health of the network.

Along with normal data, collection of erroneous data is also required. The VSI shell has a few nodes that can have their values manipulated to emulate hardware faults. For example, there exists a node that enables us to set a delay at the storage adapter level, for each IO that takes place between the datastore and any connected virtual machine.

### 3.1.2 The Database

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. The fact that its a NoSQL database gives us the flexibility to readily modify the fields of data we need to record, unlike SQL databases which must define a schema before storing any data. Even though MongoDB was utilized for the project, the code is flexible enough to accommodate any change in the database.

### 3.1.3 POP Reverse Proxy

In computer networks, a reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client as if they originated from the Web server itself. Figure 3.4 shows us the position and the importance of a reverse proxy.

Since all companies value their privacy, VMware has made sure that any given host on the SDDC or the vCenter Server Appliance is not directly accessi-

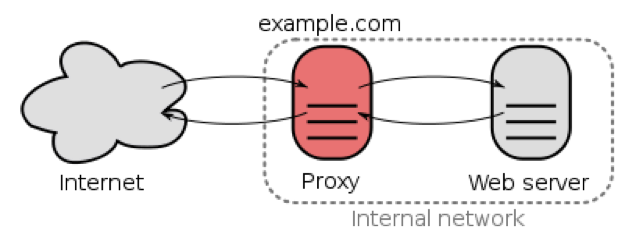


Figure 3.4: Reverse Proxy

ble from the internet. Instead, all requests for connection must be mandatorily routed through a reverse proxy (internally known as POP). Since POP is the only single point of entry to an SDDCs internal network, our processing appliance can only read data through one of the many agents running on it. All the data collection code must also be run on the POP since the processing appliance cannot directly contact the ESXi hosts.

### 3.1.4 Processing Appliance

The data processing component runs every x seconds (configurable frequency). It reads the last 2000 records written to the database and processes it to find any anomalies detected in the system. This program contacts the POP Reverse Proxy for all the SDDCs to be monitored and fetches the data to be processed. Starting from the lowest layer, the software moves to the higher layers and reports any anomaly it finds in the process.

The process of anomaly detection happens through a novelty detection algorithm that is explained in detail in the implementation details. Once an anomaly has been detected, the application takes the necessary steps by either raising an alert on a dashboard or by taking an action through the Remediation and Troubleshooting System.

## 3.2 Data Analysis and Error Prediction

Perhaps the biggest challenge for any monitoring project is picking the right amount of data and the right metrics. Both too little data and too much of noisy data can harm analysis and prediction. At first, a control chart technique (Explained in 2.3.1) was used to determine normal upper and lower bounds for data. However, due to reasons stated above, the control chart technique was not always accurate.

In fact, not only the control chart technique, but most of the error detection techniques were extremely susceptible to erroneous values. So, a new novelty detection algorithm was researched upon which was not sensitive to erroneous values. This new error detection algorithm also gave us a better upper thresholds for values in the case where the standard deviation of the values was close to, if not zero. Data that comprised of a lot of similar values also posed a problem to the control chart technique since it would result in a very low standard deviation which in turn would result in an acceptable range that was not very tolerant of incoming values.

The algorithm that was finally used comprised mainly of two parts Clustering and Probability Estimation to find an upper threshold.

### 3.2.1 Clustering

Since all the novelty detection algorithms were heavily affected by erroneous values, there was a need to get rid of values that were significantly far away from the normal trend followed by the data points. To deal with this, a new approach of K-Means, Repetitive K-Means was used as a pre-processing step. Figure 3.5 is shows the flow of this pre-processing algorithm, and is followed by an explanation.

The algorithm goes through multiple steps before returning with the filtered data, as explained below

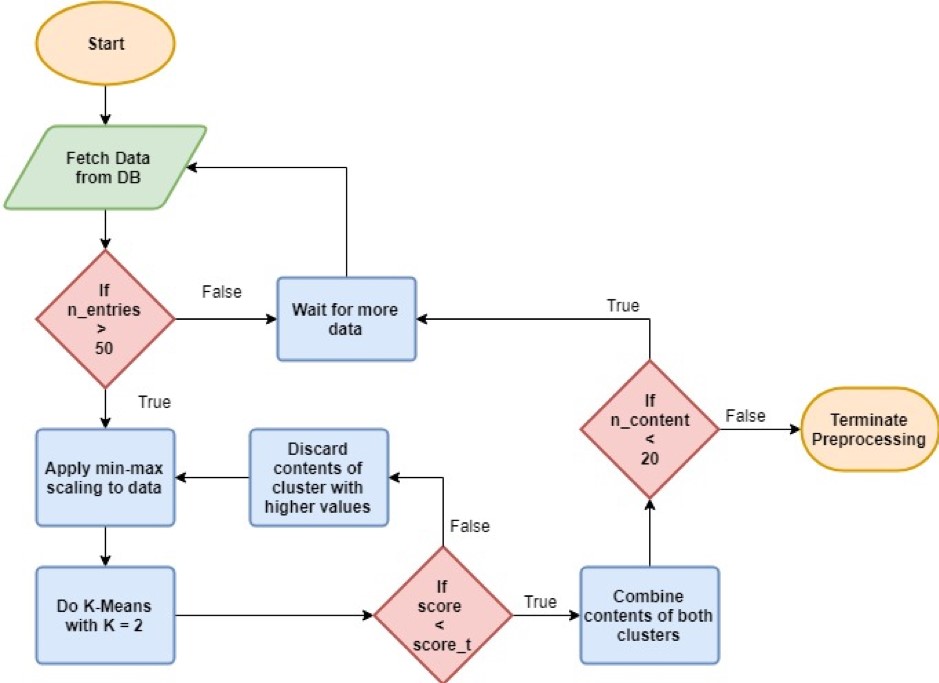


Figure 3.5: Repetitive K-Means algorithm

* Time Series data is read from the data source (POP Reverse Proxy in our case) on a per-host, per-layer basis. If sufficient non-zero records exist on the database, the data processing starts, otherwise we wait for more data to be written into the database from the data collector.
* Once we have enough data, we apply a min-max scaling over the values over the range 0 to 1, i.e., the highest value present will now be scaled down to 1, and the lowest value will be scaled down to 0.
* Then we apply K-Means and form 2 clusters out of the given data.
* The distance between the two centroids so formed will be in the range 0 to 1. This distance is denoted by score. score t is an adjustable hyperparameter which gives us a minimum score for termination, i.e., if the value of score is greater than the threshold score, score t, it implies that the 2 clusters formed are still farther from each other than we would like.
* If the clusters are far enough, we pick the cluster with smaller values (since we would always ideally want latencies to be low), and repeat from the min-max scaling step until the score is lesser than the threshold set.
* Once this condition is satisfied, we check if we are left with enough clean data points. If not, we wait for more data to come in and then try again.
* When we have enough data points (hardcoded to 20 points in the project), we can feed these valid data points in to the next Kernel Density Estimation step.

### 3.2.2 Kernel Density Estimation

Once we have collected enough clean data, we move on to forming an estimation model which can then be used to set upper thresholds on latencies. Unlike the pre-processing clustering step, Kernel Density Estimation must happen much more often to make sure that the thresholds remain updated depending on the most recent values and never get stale.

For the project, we have taken the a Gaussian (normal distribution) kernel and a bandwidth of 1. The thresholds are set by finding the point at which the area under the probability density function (PDF) sums to 0.995. A flow of the second part of the data analysis is show in Figure 3.6, followed by an

explanation.

The steps involved in the algorithm are as follows –

* Start by pre-processing the data using the repetitive K-Means clustering explained in the previous topic.
* Form a KDE model using this clean data to ensure that the probability density function obtained is uniform.
* Determine the upper threshold by finding the point at which the area under the PDF reaches 0.995 (configurable value). This can be done by

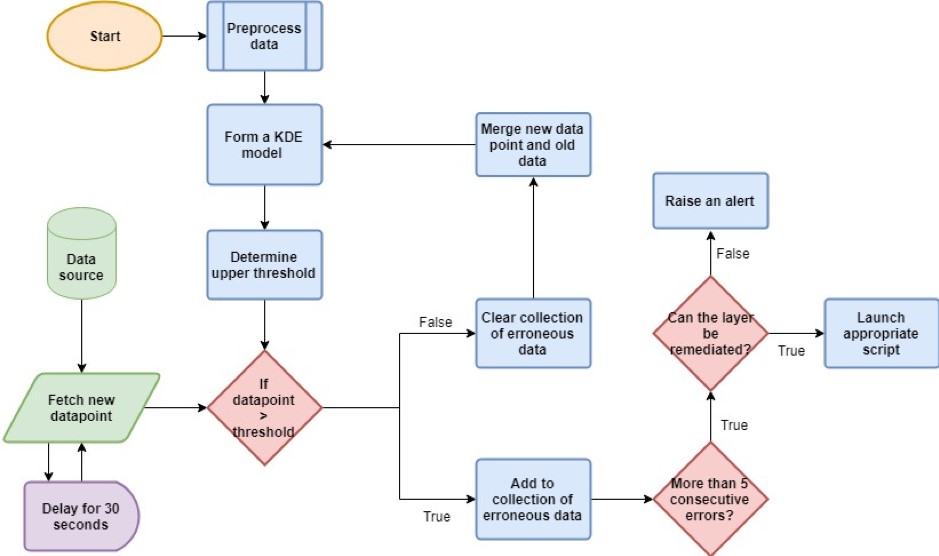


Figure 3.6: Kernel Density Estimation Flow

iteratively checking for an integral point where the cumulative density function just crosses the required value.

* This upper threshold can then be used to compare new data points. If the incoming data points are larger than this threshold value, we can keep track of them in a separate collection.
* If the last few values have been consecutively above the normal threshold set by KDE, there is a certain problem that needs to be addressed. If a rule exists to fix the said problem, apply it. Else, raise an alert so that someone could look into it.
* If a normal value comes in, we queue it so that it can be included in the KDE when the next threshold is to be calculated. This ensures that the threshold is ever-changing and accounts for slight changes in trends.
* The KDE model is updated every x iterations (x = 10 for this project), where this x could depend on how frequently the user wants to change the thresholds. A very high frequency can lead to slowdowns since computing KDE is computationally costly.

## 3.3 Remediation and Troubleshooting Service

The product is composed of the following three components -

* *RTS UI* – It renders an interface developed using Angular 2. It is a single central server that has access to all SDDC instances. It gives the ability to view or execute runbooks. It also gives the ability to execute individual scripts and lets user see the progress of the script execution. The UI will show task execution progress, results, errors, and logs.
* *RTS Server* – It provides the infrastructure to connect to the software defined datacenters and perform operations on them. The features of RTS server are listed below -
  + Library of Runbooks. A runbook is made up of several automation scripts (Modularity is implemented in the automation scripts that is they can be added or removed without effecting others) **–** Provides REST APIs for each script.
  + These scripts can call RTS Agent and the other SDDC components (VC, PSC, NSX, ESX)
  + It suupports asynchronous and synchronous script execution mechanisms.
  + Script execution progress and state can be monitored.
  + Keeps the list of previous script executions together with invocation parameters, performance metrics, execution results, errors, logs.
  + Provides direct management capabilities for services running within the ESXi hosts, VCenter appliance and NSX cluster nodes. The capabilities include starting, stopping, restarting and checking the status of these services.
  + It can also generate log bundles for troubleshooting the components.
* *RTS Agent* – The Remediation and Troubleshooting Agent is an intermediary web service inside the SDDC PoP. It provides the framework to execute commands on SDDC components like VC, ESX, NSX, and

VSAN in a controlled and audit-able manner. It reduces the need for SSH access to a SDDC components and protects the stability of these components. The commands are exposed to the users as REST APIs. Command line interface to access the APIs is also available. The agent has a minimal footprint and is stateless. The features of RTS Agent are listed below -

* + Modular APIs to promote ease of use and maintainability. API additions or removals do not impact the functionality of other APIs within RTS Agent.
  + Ability to access any SDDC component (VC, PSC, NSX, ESX)
  + It uses SSH, SCP, etc. to access SDDC components
  + Access to APIs need to be protected by an authentication mechanism.
  + Generates Logs for RTS Agent troubleshooting, and command au-

dits

### 3.3.1 RTS Design

The architecture is shown in Figure 3.7. The workflow is outlined below –

1. Monitoring components collects monitoring data.
2. Monitoring data is pushed into Operational Insight (OI).
3. Rules and Alerts engine (Nagios) listens to OI and triggers an alert. The alert description contains http link to a Runbook together with context (sddc id, esx id, ...).
4. User clicks the Runbook http link. Context is passed as parameter.
   1. RTS Service Runbook manager shows Runbook document in browser

UI.

* 1. Runbook document contains Execute buttons for automated steps.

1. User clicks the Execute button. Context is passed as parameter together with id of script to execute.
   1. RTS Service Script manager starts an async task in SS Base to execute the script. Script makes one or more calls to SDDC components and RTS agent.
   2. UI tracks task execution and provide updates (spinner or percent-

age).

* 1. Task completes. UI shows task status as success or failure. Failed tasks provide a link to failure details (error message or exception

text).

### 3.3.2 Remediation procedure through RTS

The software developed for anomaly detection identifies the layer where the problem occured. To remediate the issue, commands can be executed on the SDDC through RTS.

The general tools available within RTS that can be used for solving these problems are-

* **Log bundle collection** - All the products including vCenter, NSX, ESX etc save the logs in the file-system of their hosts at a specific location.

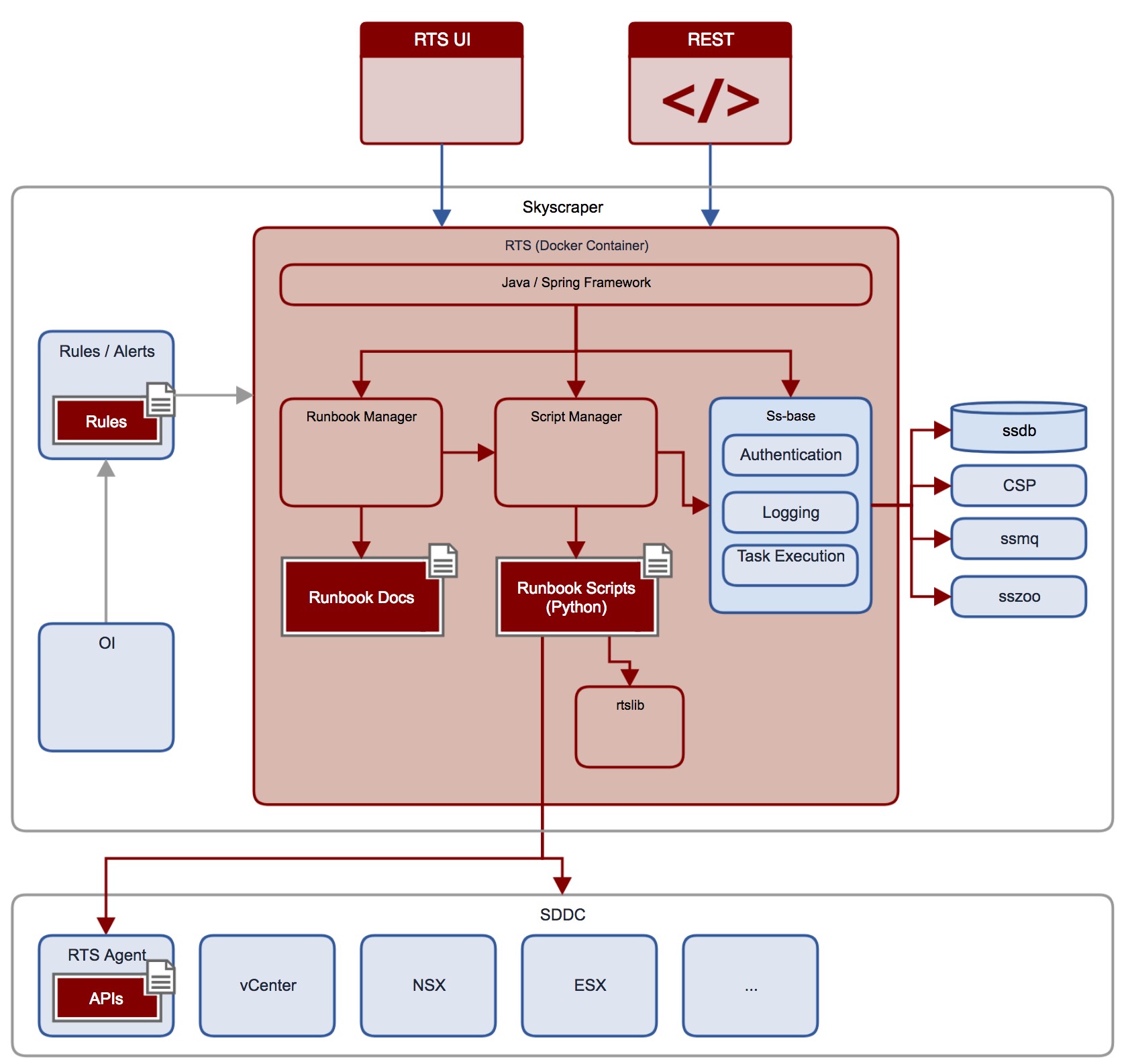


Figure 3.7: RTS Architecture These bundles can be downloaded to the RTS server and then onto our local machines using certain scripts and then can be used for debugging purposes. The Log collection UI is shown in Figure 3.8.

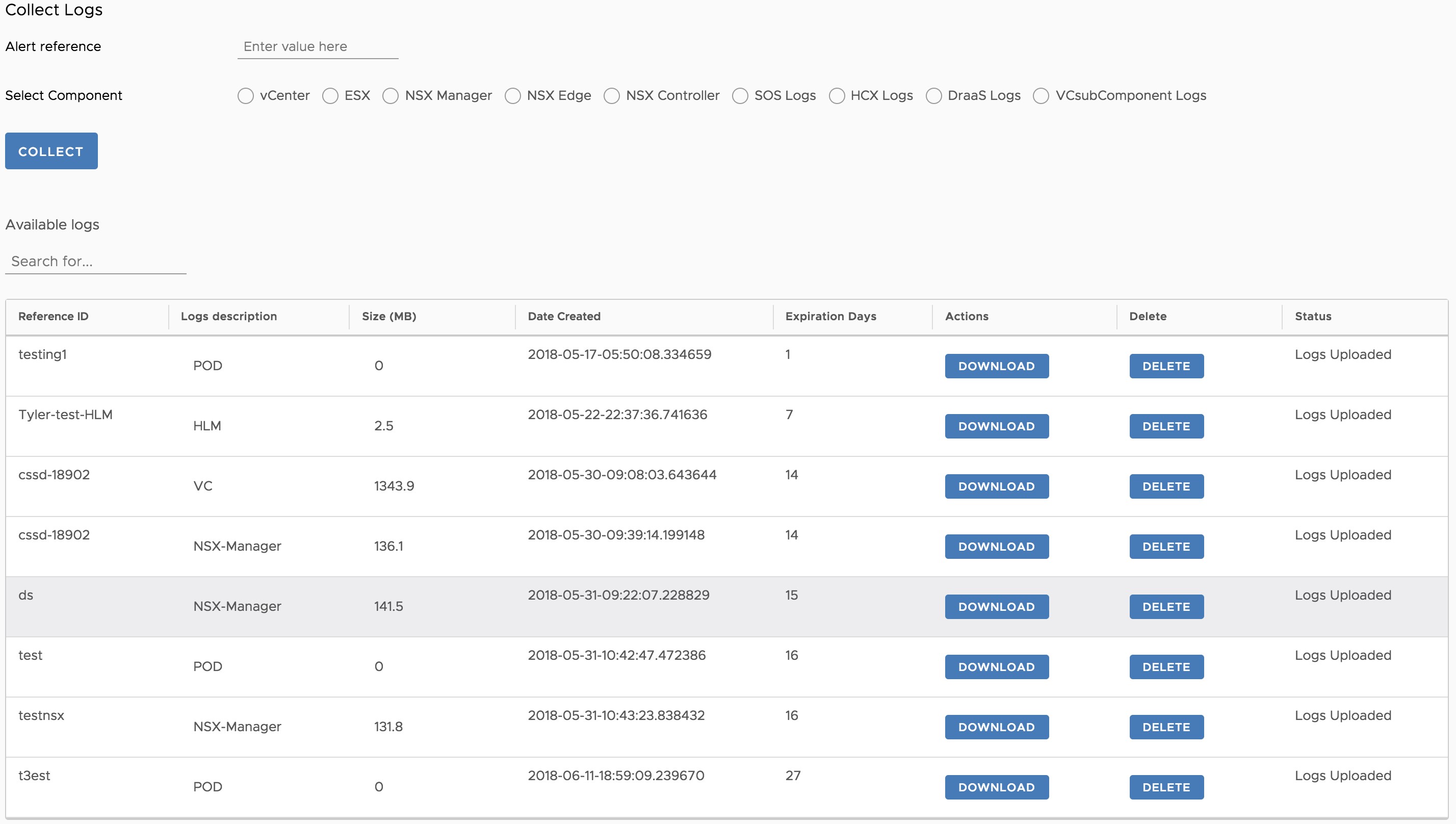


Figure 3.8: Log Collection Interface

* **Managing the services** - The system level processes/ services (usually, *Systemd* services) running in the appliances which make up the components work can be started, stopped or restarted using RTS after checking their present status.
* **Runbooks** - Runbooks constitute a pre-defined series of scripts which are ran together in a sequential manner to solve a specific problem. They are created when a problem is solved using a series of scripts. That sequence is then recorded and stored so that it can be used for solving a same or similar problem in the future.
* **Alarms** - Most products are configured to raise alarms when a system metric is out of place (for example, the network latency is high or the storage limit is about to be exceeded). Alarms from all these products are aggregated and listed in RTS and hence, can be used for troubleshooting.

#### 3.3.2.1 Example scenario

An example of the layer where the problem was detected and the corresponding scripts/runbooks that can be used for remediation is shown below-

* **Problem detected in layer:** *vsan*
* **Scripts that can be used:** The scripts in Figure 3.9 correspond to the automation scripts related to vSAN. They range from checking the status of the services, monitoring the performance object’s space usage to seeing the results of the last health tests. A sample execution ouput for the script which checks the liveliness of the clomd service is shown in Figure 3.10.

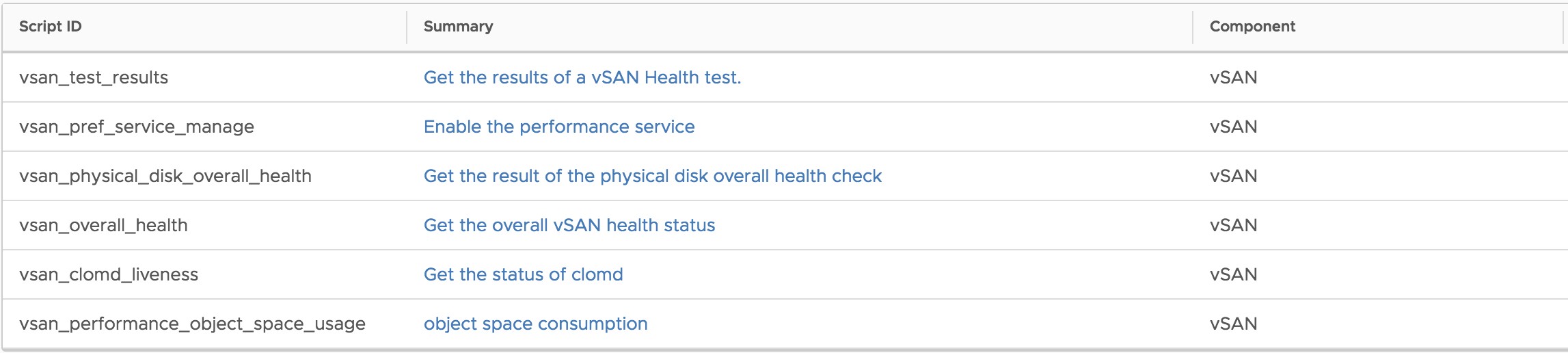


Figure 3.9: vSan Scripts

* **Runbooks that can be used:** The runbooks shown in Figure 3.11 correspond to the runbooks related to vSAN. They include runbooks related to checking the status of the *Stats database*, checking the overall status of the physical hard disks etc.
* An expanded runbook which is used to check the overall health of the physical disk is shown in Figure 3.12.

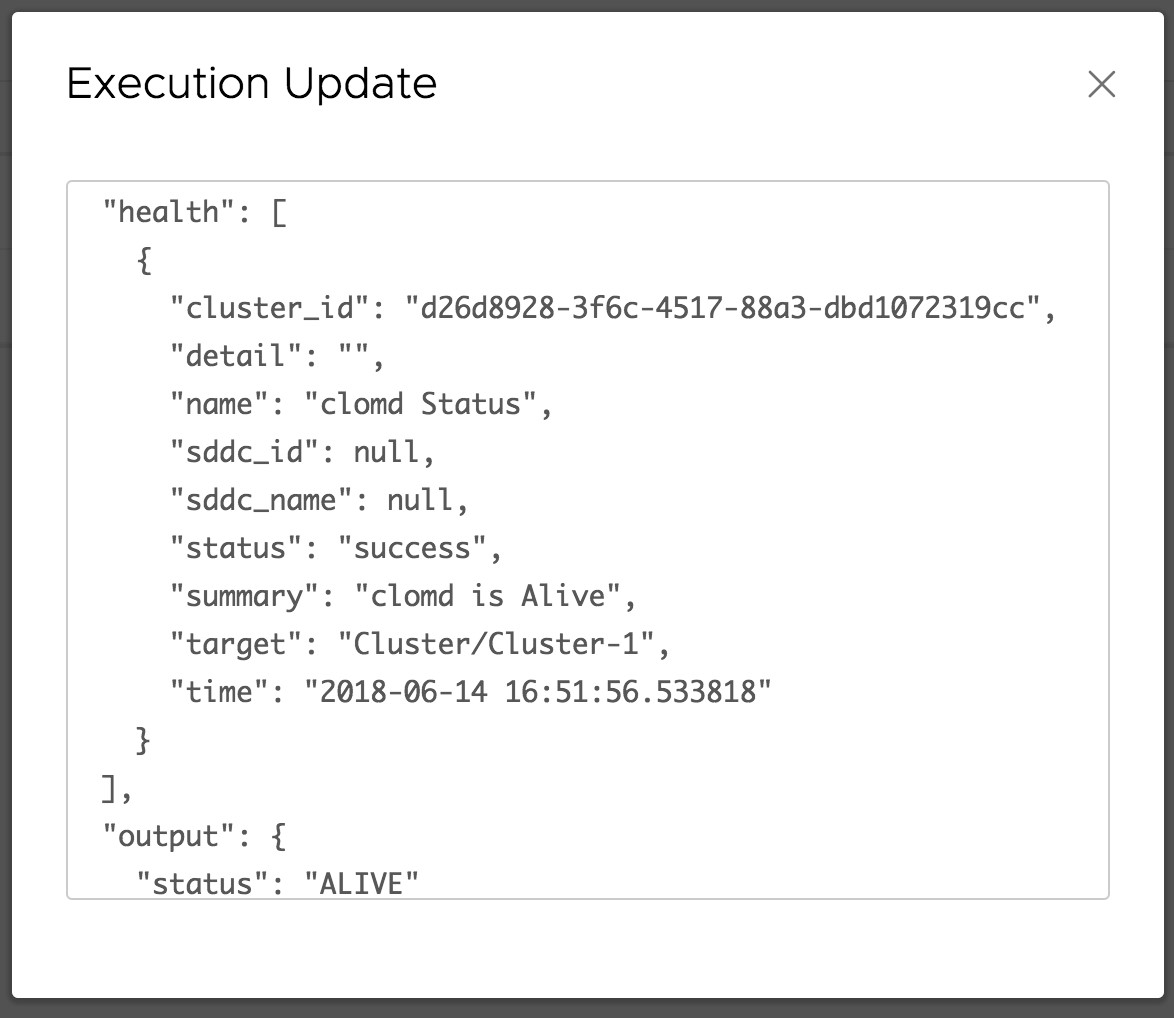


Figure 3.10: Sample execution output

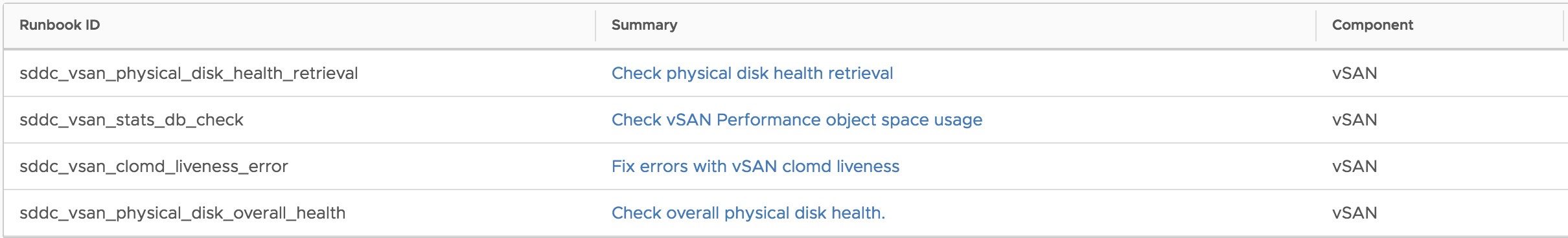


Figure 3.11: vSan Runbooks

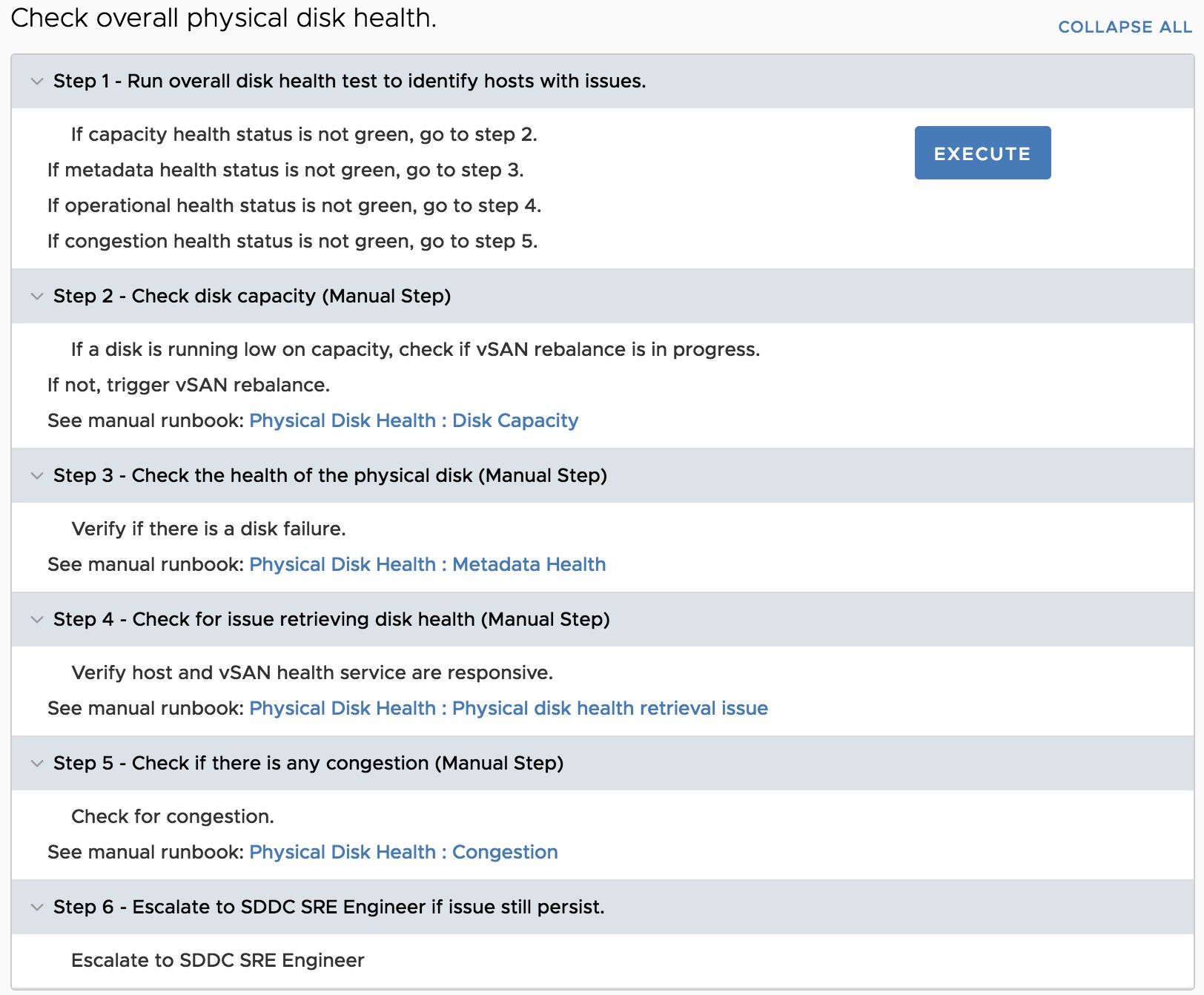


Figure 3.12: A runbook to check the overall physical disk health

**Chapter 4**

# Results

VMware Cloud on AWS is an integrated cloud offering jointly developed by AWS and VMware delivering a highly scalable, secure and innovative service that allows organizations to seamlessly migrate and extend their on-premises VMware vSphere-based environments including vSphere, vSAN, NSX, and vCenter Server to the AWS Cloud running on next-generation Amazon Elastic Compute Cloud (Amazon EC2) bare metal infrastructure. VMware Cloud on AWS brings the broad, diverse and rich innovations of AWS services natively to the enterprise applications running on VMware’s compute, storage and network virtualization platforms. The service automatically provisions infrastructure and provides full VM compatibility and workload portability between your on-premises environments and the AWS Cloud [8].

This collaborative venture has proven to be profitable to both the companies with the customer base rapidly increasing. Reliability and availability of the product thus poses an important challenge. Our software helps in solving the problem in the following ways –

* It provides a common interface for data collection from several largely unrelated sources which may be relevant to the problem. Right now, VMkernel Sys Info Shell, TCP dump, VProbes and some REST API

endpoints provide the required data but this list can be easily extended

as and when required.

* It provides a common algorithm using K-Means and Kernel density estimation which can be used throughout the stack of layers to recognize problems in the infrastructure by analyzing different metrics from each of the layers.
* The algorithms are self-adjusting and self-improving. They align with the general trend of the metrics during a particular period of time to decrease the detection of false positives.
* The software is flexible enough to introduce more layers and parameters to monitor pretty easily as and when the architecture of the SDDC components change.
* The layer in which the problem is detected is directly obtained from the software. Corresponding scripts and runbooks pre-existing in RTS can be used to remediate the issue. These automation scripts have to be created once and then can be used forever to solve the same or similar problems that may be encountered later removing the need for manual

interventions.

VMware’s SRE team was created when it’s VMC product was launched. Site Reliability Engineering (SRE) is a discipline that incorporates aspects of software engineering and applies that to IT operations problems. The main goals are to create ultra-scalable and highly reliable software systems. This software is a step towards achieving that goal.

**Chapter 5**

# Conclusion and Future Work

## 5.1 Conclusion

A complete product has been developed which will help in reducing the time taken by developers and engineers to root cause an issue and then resolving it in an industry level SDDC stack. It will also help in recording and comparing performance changes as well as removing bugs for the upcoming releases of the products without having to spend time on checking each product and each operation manually. The framework has been released internally and is being used by teams in Cloud Platform Business Unit. There are plans of making the product a fling so that people can provide feedback and suggestions based on which it can be launched as a standalone VMware product if the project receives positive responses during the fling period.

## 5.2 Future Work

There’s a scope for improvement of this software and some of the following suggestions can be worked upon in the future.

* The layered dependency of all the products and components of the SDDC stack haven’t been fully discovered yet. As these dependencies become more apparent, additional metric parameters and additional layers can be added to the software. More data sources can be added as well.
* Presently, the UI which shows the status of the layers in a graphical format is a part of the software package itself. Later, the backend and the user interface can be decoupled. REST endpoints can be created at the backend side to query the state of the SDDC by the UI. The UI service will then construct and display the dependency graph with the states in a browser. This decoupling of micro-services can prove to be beneficial when there is a single dedicated backend server and many consumer clients. It will help in easy debugging and show significant performance improvements.
* RTS is a go-to tool for the entire SRE team for troubleshooting the SDDC issues. If this software can be integrated with RTS and can be accessed through it’s UI itself, productivity of the engineers will be increased significantly as the switching time will be removed.

# References

1. Vmware products. [Online]. Available: [https://www.vmware.com/in/ products.html](https://www.vmware.com/in/products.html)
2. vsphere documentation. [Online]. Available: [https://pubs.vmware.com/ vsphere-51/](https://pubs.vmware.com/vsphere-51/)
3. Introduction to statistics. [Online]. Available: [https://en.wikipedia.org/ wiki/Statistics](https://en.wikipedia.org/wiki/Statistics)
4. Control chart. [Online]. Available: [http://asq.org/learn-about-quality/ data-collection-analysis-tools/overview/control-chart.html](http://asq.org/learn-about-quality/data-collection-analysis-tools/overview/control-chart.html)
5. 68–95–99.7 rule. [Online]. Available: [https://en.wikipedia.org/wiki/68% E2%80%9395%E2%80%9399.7](https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule) [rule](https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule)
6. Kernel density estimation. [Online]. Available: [https://en.wikipedia.org/ wiki/Kernel](https://en.wikipedia.org/wiki/Kernel_density_estimation) [density](https://en.wikipedia.org/wiki/Kernel_density_estimation) [estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation)
7. Introduction to vmware software-defined data center. [Online]. Available: <https://code.vmware.com/sddc-getting-started>
8. Vmware cloud on aws. [Online]. Available: [https://aws.amazon.com/ vmware/](https://aws.amazon.com/vmware/)

Project Details

*Student Details*

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** |  | Nikhil Gupta |  |
| Registration Number | 140911466 | Section/Roll No. | B/64 |
| Email Address | nanonikhil@gmail.com | Phone No.(M) | 7390932789 |

*Project Details*

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | A Proactive Health Monitoring Tool for Software Defined Datacenters | |  |
| Project Duration | 6 Months | Date of Reporting | 10-01-2018 |

*Organization Details*

|  |  |
| --- | --- |
| **Organization Name** | VMware Software India Pvt. Ltd. |
| Full Postal Address | Kalyani Vista, JP Nagar, Bengaluru(560076), India |
| Website Address | www.vmware.com |

*Supervisor Details*

|  |  |  |  |
| --- | --- | --- | --- |
| **Supervisor Name** | Priya Pandian | |  |
| Designation | Senior Manager | |  |
| Address | JP Nagar, Bengaluru(560076), India | |  |
| Email Address | kpandian@vmware.com | Phone No.(M) | 9886172287 |

*Internal Guide Details*

|  |  |
| --- | --- |
| **Faculty Name** | Raghvendra Achar |
| Address | Dept. of I&CT, MIT, Manipal(576104), India |
| Email Address | raghavendra.a@manipal.edu |