{CUSTOMER\_NAME}

Cloud Ready Assessment

PREPARED

1st Quarter 2024

PREPARED FOR

{CUSTOMER\_NAME } {CITY NAME }

BY {SA-NAME} {SA\_EMAIL}

PREPARED BY

{RAX\_TEAM}

[Document Version 2](#_Toc1241420885)

[Copyright 3](#_Toc996937646)

[EXECUTIVE SUMMARY 3](#_Toc955277169)

[BUSINESS CHALLENGE 4](#_Toc932116248)

[PROJECT OVERVIEW 5](#_Toc732243799)

[Scope 5](#_Toc458863487)

[Methodology 5](#_Toc505704813)

[Data Collection and Assessment Activities 6](#_Toc1808284786)

[Private Cloud Options 6](#_Toc1575706032)

[Private Cloud Preference Summary 6](#_Toc2017114784)

[GOVERNANCE 10](#_Toc503574384)

[Risk 10](#_Toc729476333)

[Financial/Resource Availability 10](#_Toc782873618)

[Security 11](#_Toc49877966)

[Stability 11](#_Toc1065570515)

[Availability 11](#_Toc765619111)

[Recommendations 11](#_Toc1911267665)

[Success Factors 12](#_Toc1233380892)

[Guiding Principles 12](#_Toc880830360)

[Biases 12](#_Toc526509066)

[Staffing 13](#_Toc903850959)

[Governance Committee 13](#_Toc371729097)

[Program/Project Management 14](#_Toc1127836169)

[Migration Architects/Engineers 14](#_Toc1414571807)

[Pre/Post Migration Support Teams 15](#_Toc1942806828)

[Vendor Support 15](#_Toc526608127)

[Project Oversight 15](#_Toc439891288)

[Staff Training/Documentation 15](#_Toc835524720)

[CLOUD PLATFORM CONSIDERATIONS 15](#_Toc1890208767)

[Automation and Infrastructure as Code (IaC) 16](#_Toc346680974)

[Platform as a Service (PaaS) 18](#_Toc700051845)

[Vendor Supported Architecture 18](#_Toc874794951)

[Ephemeral Environments and Blue-Green Deployment 19](#_Toc1774210767)

[Best Practice Security Model 19](#_Toc177961)

[Key Management 20](#_Toc544850107)

[Network 20](#_Toc70563144)

[BEST-FIT CLOUD DECISION MATRIX 21](#_Toc2013162917)

[Target Platform Recommandation 25](#_Toc2071906245)

[MIGRATION TIMELINE AND MOVE GROUPING 26](#_Toc1134126380)

[Disposition 26](#_Toc501694327)

[Rehost, Retain, Refactor Recommendations 29](#_Toc1736878079)

[Move Groups 31](#_Toc144751046)

[Migration Strategies and Roadmap 32](#_Toc71987568)

[DISASTER RECOVERY 32](#_Toc208127251)

[SLA Objectives 33](#_Toc170176752)

[Recommendations 35](#_Toc2128229819)

[SUMMARY OF RECOMMENDATIONS (In Scope and Out of Scope) 36](#_Toc1535343575)

[Recommendations Introduction 37](#_Toc1723684788)

[Cloud Center of Excellence 37](#_Toc985046928)

[User Management, Governance, and Access Control 38](#_Toc2101009530)

[Local Area Networking (LAN) and Wide Area Network (WAN) 39](#_Toc1321948732)

[Backups, Security, Logging, Monitoring, and RISK Management 39](#_Toc2136790358)

[Applications and Operating Systems 40](#_Toc519481749)

[Databases, Artificial Intelligence, and Machine Learning 41](#_Toc1179111953)

[Future of Success – Next steps and performance optimization 42](#_Toc63539201)

[Appendix A – Medium Level Design 44](#_Toc33608747)

[Introduction 45](#_Toc1794231813)

[Background 45](#_Toc10758927)

[Assumptions 45](#_Toc1439745202)

[Architecture 46](#_Toc555706106)

[Software Versions 49](#_Toc578361782)

[Supported Hardware 49](#_Toc722417898)

[Networking 49](#_Toc1983771312)

[Logical Network Configuration 50](#_Toc1760271897)

[Physical IP ranges 50](#_Toc340235954)

[Additional Network considerations 51](#_Toc127187597)

[Ironic Node Flavor Configuration 54](#_Toc1153463488)

[SSL Certificates 55](#_Toc1114436338)

[OpenStack components 55](#_Toc590378310)

[Project/User Configuration and Naming 56](#_Toc485409671)

[Deployment Configurations 57](#_Toc1468725840)

[Controller Node and HA Proxy Nodes 57](#_Toc390114890)

[Compute Nodes (Nova) 58](#_Toc1932945645)

[Appendix B – Supplemental materials for Operations controls 58](#_Toc1062094605)

[Service Quality Management and Systems Management 61](#_Toc329228026)

[Capacity Management 62](#_Toc437131799)

[Standard Operating Environment 62](#_Toc330436857)

[Availability Management 62](#_Toc1747165616)

[Data Management 62](#_Toc1105527335)

[Continuity Management 62](#_Toc1400802716)

[Information Security Management 63](#_Toc1753528184)

[Operational Readiness 63](#_Toc972807063)

[Software Requirements 63](#_Toc1311797082)

[Appendix C – Total cost of ownership 63](#_Toc132828139)

## Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Owner | Date Updated | Notes |
| 1.0 – Initial Draft | Matt Richins | February 6, 2024 | First Draft based on workshop input |
| 1.1 | Matt Richins | March 14 | Updates made based on additional documens and information provided by {PARTNER\_NAME } |
| 1.2 | Matt Richins | March 30th | Grammar cleanup and additional details added pending automation review. |
| 1.3 | Matt Richins | April 23 | Updated based on requirements added by {PARTNER\_NAME } feedback and clarification on tooling output |

## Copyright

This document contains proprietary information, which is for exclusive use of Rackspace, {PARTNER\_NAME} and {CUSTOMER\_NAME} . This document is not to be shared with personnel other than Rackspace, {PARTNER\_NAME} or {CUSTOMER\_NAME} . This document, and any portion thereof, may not be copied, reproduced, photocopied, stored electronically on a retrieval system, or transmitted without the express written consent of the Rackspace, {PARTNER\_NAME} or {CUSTOMER\_NAME} Owners.

Rackspace does not warrant this document to be free of errors or omissions. Rackspace reserves the right to make corrections, updates, revisions, or changes to the information contained herein. Rackspace does not warrant the material described herein to be free of patent infringement.

Unless provided otherwise in writing BY RACKSPACE, the information and programs described herein are provided “as is” without warranty of any kind, including but not limited to the implied warranties of merchant-ability and fitness for a particular purpose. In no event will RACKSPACE, its officers, directors, or employees or affiliates of RACKSPACE, their respective officers, directors, or employees be liable to any entity for any special, collateral, incidental, or consequential damages, including without any limitation, for any lost profits or lost savings, related or arising in any way from or out of the use or inability to use the information or programs set forth herein, even if it has been notified of the possibility of such damage by the purchaser or any third party.

# EXECUTIVE SUMMARY

{EXECSUM}

{presentation of Company}

{company Business and Customers}

{CUSTOMER\_NAME} is seeking an analysis of it core applications and servers to establish a technology best-fit cloud for its application estate. The goal is to enable {CUSTOMER\_NAME} to provide continuity of services, reduce cost, maintain or increase reliability, scalability, and elasticity of their application sets.

As a continuance of this growth and improvement culture, Rackspace Professional Services has recently conducted a Cloud Readiness assessment of the {CUSTOMER\_NAME } infrastructure to provide application readiness, target platform mapping, design recommendations relative to the target landing zone, and considerations for digital transformation. The organization is driven to adopt a cloud strategy, adaptable to rapid change, while achieving a predictable and competitive cost structure. The major objectives and goals affiliated with this Cloud Readiness Assessment (CRA) are promoting efficiency, cost, and stability coupled with the ability to scale IT operations and development.

Overall, our assessment has discovered that {CUSTOMER\_NAME} has adopted best in suite technologies such as Oracle for Databases, VMware for virtualization, and Cisco for network controls. {CUSTOMER\_NAME} has mature processes around operating their technology stacks. Additionally, {CUSTOMER\_NAME} has a robust planning framework and internal change controls. Where needed {CUSTOMER\_NAME} has deviated from best practices to meet the demands of business logic and operational debt which is appropriate for IT operations. Moreover, {CUSTOMER\_NAME} has documented the exception processes when deviating from best practices which allows for a re-architecture at a later date when the non-functional requirement or business logic has changed.

Our assessment focused on the following elements:

* Determine and recommend best-fit cloud for the {CUSTOMER\_NAME} application portfolio.
* Evaluate all considerations outlined in our discussions as it applies to the scaffolding design of a private cloud architecture.
* Determine he best private cloud to sustain HA, increase scalability, and maintain consistent performance.
* Build efficiencies in both infrastructure and operations.
* Develop a predictable cost model streamlining financial management.
* Establish a set of cloud best practices for governance, security, and application workload design.

The specific recommendations from Rackspace Professional Services are located in the latter portion of this document in the section titles “SUMMARY OF RECOMMENDATIONS”.

# 

# BUSINESS CHALLENGE

**{BIZ CHALLENGE}**

Rackspace Professional Services (“Rackspace”) is assisting {CUSTOMER\_NAME} in providing recommendations to assist {CUSTOMER\_NAME} on a platform that will enable {CUSTOMER\_NAME} to modernize its operations into a more cloud like manner and to make recommendations on operational practices based on current Rackspace recommended best practices. The migration to a cloud operations model is commonly referred to as a digital transformation journey. To make recommendations on the best practices and the ideal target cloud platform a series of workshops/interviews have been conducted to help assess cloud readiness of {CUSTOMER\_NAME} ’s core applications sets to determine best-fit cloud and readiness and cloud needs. Additionally, {CUSTOMER\_NAME } provided operational documentation for review and a infrastructure and TCO assessment was performed.

## PROJECT OVERVIEW

**{PROJECT OVERVIEW}**

Rackspace Professional Services has been asked to complete a Cloud Readiness assessment for {CUSTOMER\_NAME} to determine the feasibility of their application workloads migrating to a best-fit cloud infrastructure. As outlined within the deliverables collection, Rackspace will provide a high-level reference architecture of the target cloud platform, financial analysis and run state cost modeling, target platform comparison and recommendation, application workload disposition, and process governance standards. The information analyzed and reported are the results of our discussions and the {Audit tools } tooling discovery and collection process.

This document provides a best-fit cloud platform recommendation, and best practice to assist {CUSTOMER\_NAME} to determine a roadmap in resolving the aforementioned business challenges. We understand the need for {CUSTOMER\_NAME} to act within a steady pace in its deliberation on the proposed solution as it encompasses multiple platforms and workload environments and the need to work with the other {CUSTOMER\_NAME } organizations on strategy for operational changes. Given the complexities of operational transformation several sections have been added to provide details for transformation with a summary of next steps at the end of this report.

Rackspace has prepared this assessment based on discussion with the {CUSTOMER\_NAME} operational teams and {CUSTOMER\_NAME } providers such as {PARTNER\_NAME } . The document contains multiple sections pertaining to the corresponding activity. In the first section, we describe the request from {CUSTOMER\_NAME} and our project methodology in performing the assessment and subsequent efforts that may be outputs from this engagement. This is followed by our findings about the current infrastructure, application workloads, and network posture of {CUSTOMER\_NAME} . Finally, we provide a section of infrastructure and workload recommendations, financial model, and our recommendations.

## Scope

**{Scope}**

The defined scope of this engagement is to present an assessment report of recommendations for {CUSTOMER\_NAME} based on Rackspace best practices with a proposed infrastructure platform that will assist {CUSTOMER\_NAME } in initiating a digital transformation Journey. A comparison of current costs and future costs will be provided in a separate report by {PARTNER\_NAME } based on the outputs of the design identified in this report.

## Methodology

Contrary to Rackspace’s standard approach in delivering an estimate of the anticipated number of hours required to complete an individual phase, the Statement of Work (SOW) for this project was developed on a fixed fee basis. The scope of the engagement clearly defined the required inputs and desired outcomes of the engagement. To reiterate our scope of services for this engagement:

1. Identify business objectives and pain points and understand the baseline for the current state of application architecture and delivery practices within the organization.
2. Inventory the application environments in scope using software tooling for data gathering from Flexera.
3. Determine operational gaps and identify the areas for improvement at a program and team level.
4. Create appropriate and tailored infrastructure and application workload recommendations.
5. Deliver a high-level design target infrastructure and migration roadmap incorporating findings from the onsite workshops and Cloudscape tooling and onsite workshop interviews.

Following the information gathering and stakeholder sessions, Rackspace Professional Services (RPS) task is to provide an assessment report of Rackspace’s recommendations regarding the target cloud platform and recommendations for improvements based on Rackspace best practices, provide a design for a private cloud platform that enables {CUSTOMER\_NAME} ’s digital transformation, and a Cost analysis that compares current costs with future costs of the proposed architecture based on the workloads captured during the assessment phase of the project.

At the completion of this phase once the assessment document has been delivered and reviewed, the deliverables can function as inputs for the next phase(s) of the digital transformation journey. {CUSTOMER\_NAME} may choose to leverage RPS Team and {PARTNER\_NAME } for the deployment of the target platform or may otherwise reassign the accountability as they deem appropriate.

## Data Collection and Assessment Activities

Rackspace and {PARTNER\_NAME } engaged with {CUSTOMER\_NAME} in a series of workshops during the month of January 2024 to gain insight on the organization’s infrastructure, cloud maturity, and application readiness to determine a best-fit cloud and a proper private cloud platform. The applications in scope were GIS, Doffice, HRMS, KPI, PMIS, ERP, CMIS, Dashboards, Telemetry MDIS and HES, Digital Certs, BIM, and Portal. The individual activities undertaken will be described below.

## Private Cloud Options

Many technologies exist to help build a private cloud infrastructure. The first step is to identify a mature Hypervisor that will support the virtual machine layer. In determining the best hypervsior fit for {CUSTOMER\_NAME} ; VMware ESX, Hyper-V, XCP-ng, Xen, and KVM were considered. The next step is to consider the best tooling of orchestration tools for operations surrounding the virtual machine environment. Factoring in tools that function well at small scale and large scale that also provide a means to adhere to operational and regulatory compliance requirements is critical to ensuring the success of a well-built private cloud. The orchestration tools that were considered for meeting {CUSTOMER\_NAME} ’s operational requirements are VMware, Proxmox, OpenStack, and Nutanix. Proxmox was subsequently filtered out of the selection process due to its lack of enterprise controls and Nutanix was filtered out of the selection process due to the requirement to have a full HCI based deployment on specific hardware profiles that do not fit in withbest practices for operating a private cloud.

## Private Cloud Preference Summary

As part of the engagement {CUSTOMER\_NAME} has asked Rackspace to recommend a best-fit cloud platform. Although many technologies exist for hosting a private cloud platform, very few meet the enterprise operational requirements that {CUSTOMER\_NAME} has. Of all the possible orchestration tools available as private-retail and open-source options, VMware and OpenStack are the best fits to meet {CUSTOMER\_NAME} ’s needs. Rackspace recommends OpenStack as the preferred orchestration platform with KVM as the underlying hypervisor. Although both options are mature, each with its own advantages and disadvantages, OpenStack is recommended for financial reasons and due to it being a better fit in interoperability with modern tooling that is being leveraged by software developers and private cloud operators. Overall OpenStack is the best fit for the future state of {CUSTOMER\_NAME} ’s digital transformation. The details that influence the recommendation are further outlined below.

1. The ability to manage geographic distribution of systems and data for redundancy and disaster recovery.
2. Compatibility with industry standard tools for automation, such as Terraform, Ansible, Spinnaker, Octopus Deploy, Git, Azure Devops (formerly TFS), Jenkins, Bamboo, artifactory, Git, Rally, Chocolatey, Microsoft Endpoint Configuration Manager, Microsoft DSC, and many others.
3. Individual resource management technologies
4. Automation to manage scaling up or down the number of compute instances in a cluster via operations API, CLI, or console.
5. A library for managing compute instance images for rapid deployment of new architectures.
6. Network access control at the device or group level.
7. Ease of Integration with CISCO technology via OVS and OVN.
8. Ease of application deployment and lower cost of operations for K8S technologies that are rapidly being adopted for new application development.
9. Ease of integration with 3rd party commercial off the shelf products allowing for the most flexibility between using opensource and commercially available software products and hardware.

A mid-level design will be included in the appendix of this report that will outline deployment for the recommended OpenStack environment. Rackspace and {PARTNER\_NAME} will need to work with {CUSTOMER\_NAME} to refine some of the details around networking and hardware availability in a future engagement. Overall the design calls for OpenStack to be deployed in (2) of {CUSTOMER\_NAME} ’s datacenters. Where possible the reuse of existing hardware will be preferred. The high-level design for the OpenStack deployment is depicted as follows:

**{ Arch\_diagram}**

A computer screen shot of a diagram

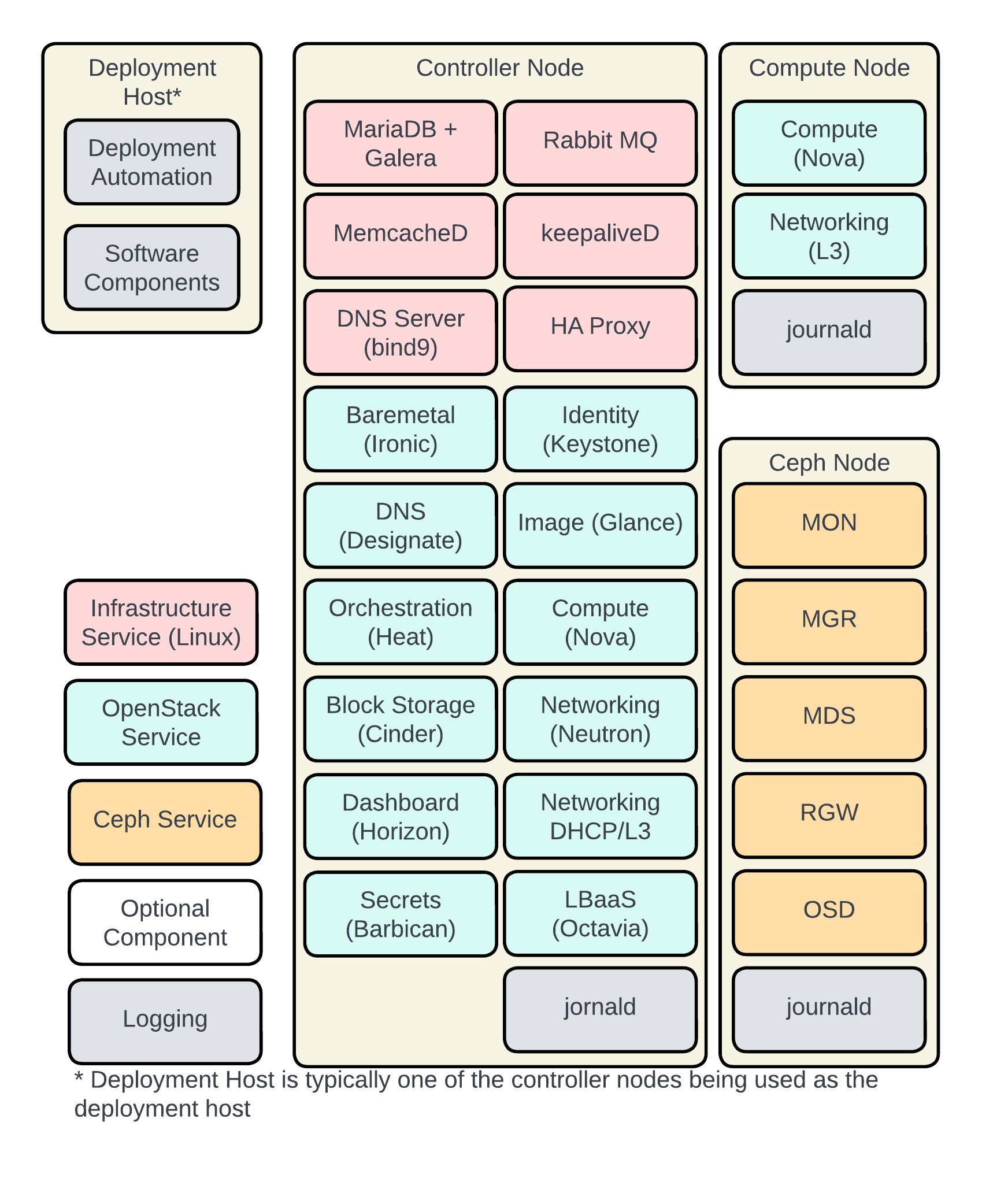
Description automatically generated

As {CUSTOMER\_NAME} continues to evolve its operations capabilities and moves into the containerization space, OpenStack also opens itself up into a natural fit for deploying Kubernetes workloads:

A diagram of a server

Description automatically generated

Deployment services that are being proposed on the controller nodes as part of the design are as follows. Deploying CEPH is still to be determined based on further analysis of needs for object based storage. It is anticipated that CEPH or an object storage solution will be configured at a later date and not part of an initial scope.



## GOVERNANCE

Gartner defines IT governance as “the processes that ensure the effective and efficient use of IT in enabling an organization to achieve its goals.”

During the course of discovery, it was identified that {CUSTOMER\_NAME} is primarily operating on a “Check Point” strategy for ensuring that people, policies and technology combine deliver business value. For example, procurement processes provide a check point control of capital expenditures, change management imposes a check point control over changes to productions systems, etc. In addition to check points encountered in day-to-day operations, physical facilities, secure network infrastructure and robust hardware platforms are pre-deployed according to an agreed upon standard.

The typical process that an engineer or business user must follow to provision new services is managed at every critical point by a review process and constrained by the environment into which it is deployed. In most enterprises, no individual can deploy resources without the assistance of several others. This creates an environment where many errors need to be made by several individuals to significantly compromise production systems. This also creates a system where artificial time constraints are injected into the process resulting in longer deployment times due to the human factor of illness, workload burden, and vacation schedules. Adopting a private cloud that provides tenant-controlled resources via automation tooling will help to remove human error and lag time associated with human checkpoints.

While many of the same policies and tools that have been employed in traditional IT Governance can be adapted to the cloud, the dynamic nature of cloud resources requires special attention to address risk. As {CUSTOMER\_NAME} looks to migrate workloads to the best-fit target cloud, the documented {CUSTOMER\_NAME} control framework policies need to also extend across and apply the same governance posture. Careful attention needs to be focused on automated workloads to ensure that the proper security and management controls remain consistent as instances scale based on demand.

### Risk

Due to the simplicity of deploying services in the cloud, the following risks need to be considered when developing a Cloud IT based Governance program. A single individual with a single keystroke can create global change across the entirety of an enterprise’s cloud footprint. This capability enables IT and engineering teams to rapidly respond to issues and deploy new features quickly and cost effectively. That same flexibility can lead to disastrous results if not properly governed.

### Financial/Resource Availability

In cloud environments, resources can be allocated with the click of a button or automatically by a line of code. Once any resource is allocated, incremental billing begins in the hyperscaler clouds which can add up quickly to cost over-runs. In private clouds, financial costs are already assumed as part of the deployment, however capacity is quickly consumed which reduces the resource availability for other users. Due to the relative ease of deployment, groups creating new instances for new projects will dramatically increase which will help to increase the pace of technological innovation. However, to help contain costs and usage, policies for tagging, quotas, and auditing longevity of resources need to be implemented to help prevent orphaned and abandoned instances so that resource availability can be freed up in lieu of acquiring more hosts. Many corporate IT cloud budgets have been destroyed by large collections of unused, but unterminated cloud resources. Governance and tagging strategies will need to be implemented.

### Security

The ease with which services can be deployed within cloud environments, the flexibility with which they can be configured and the global way they can be connected creates an opportunity for vulnerabilities to be introduced unknowingly into an otherwise extremely secure environment. Cloud infrastructure in and of itself is very secure and with proper governance and security controls can be equally or more secure than traditional datacenter facilities.

### Stability

The cloud, when taken as a whole, is “available” and “durable” in the extreme. Individual cloud resources are traditionally provided without service-level agreements (SLA) to the end user. To be reliable, available, and durable, applications deployed in the cloud must make use of properly architected infrastructure. This is especially true for containerized applications. Attention to details that may be overlooked in a traditional environment must be enforced through governance to ensure that business applications are available to their consumers. It is human nature for an individual when faced with two ways to accomplish a task, to seek the simpler solution. Proper governance ensures that deployed solutions follow reference architecture patterns and provide the level of stability that application users require.

### Availability

To this point, the ability to instantly provision cloud resources has been the focus of the risks faced by cloud-enabled enterprises. In the cloud, resources can be destroyed as quickly and easily as they can be created. Once a resource is destroyed, it is completely unrecoverable, whether it is a storage volume, database, or carefully crafted compute environment. There is no option to restore; the resource must be re-built. A single button click, or keystroke can destroy an entire environment, including DR systems and backups, as easily as a single system can be destroyed. This is another area where governance provides protection from unintended user operations.

## Recommendations

The Rackspace team in partnership with {PARTNER\_NAME } brings decades of collective experience engineering distributed systems and network architectures to assist {CUSTOMER\_NAME} in this cloud assessment project. Our experience teaches us that there is value in following certain best practices when designing cloud architecture. We list here a set of strategies that we adopt as default solutions because of their repeated proof of value.

When considering a cloud migration or infrastructure extension, Rackspace has found that projects have a higher probability of success when the following foundational elements are agreed upon prior to other work being performed.

1. Clearly stated, Success Factors by which success or failure of the effort will be measured.
2. High level Guiding Principles that dictate strategic objectives and drive decisions.
3. A list of strategic Biases that influence, rather than drive, decisions.

Once Governing Principles are determined, there are several other governance topics which should also be addressed, including but not limited to:

* Staffing
* Vendor Support
* Project Oversight
* Staff Training/Documentation

## Success Factors

Following are Success Factors for {CUSTOMER\_NAME} to successfully adopt a modern operations strategy:

1. Modernize and optimize {CUSTOMER\_NAME} IT footprint with a private cloud (OpenStack based) cloud.
2. Establish a DR solution at each application level that is tested at a minimum of once per year.
3. Maintain the capex controls of acquiring and managing its own infrastructure.
4. Increase availability, scalability, and stability across the infrastructure by implementing newer technologies associated with the OpenStack platform and Kubernetes projects.
5. Generating an internal ability to launch new workloads as the business requires without the need for lengthy preparation and provisioning.
6. Establish cost governance to maintain predictable cost models on new and existing workload instances.
7. Establishing an environment where newly launched workloads maintain the identical security and monitoring controls dictated by the platform and internal controls for access, reporting, backups, and anomaly detection.
8. Ability to maintain or increase the level of security posture using network isolation, and micro segmentation.
9. Infrastructure as Code (IaC) workload deploy and destroy timelines should be within reasonable allotments and automated as much as possible.

As the project progresses and once it is complete, success or failure of the project will be measured on how the work performed met the Success Factors above. This should be true of any migration effort whether it be cloud targeted or on premise.

## Guiding Principles

Based on the Success Factors above, Rackspace recommends the following Guiding Principles:

* Continual monitoring of cloud consumption using alerting triggers.
* All egress traffic should route through the planned hub network design with proper perimeter scanning controls.
* The principle is not required; however, EAST/WEST network traffic scanning controls are highly recommended for when data needs to talk across tenants.
  1. Separation of networks for critical low-latency data replication networks should be an exemption to this design pattern.
* All design architectures should be cognizant of target platform soft/hard limitations.
* Enforce existing Naming Convention to preserve consistency.
* Enforcing tagging strategies that identify creation, project, owner, business unit/department, environment, application identifier, data classification, workload name, operations team, cost center, end of date, start date of project, and service class (i.e. gold, silver, bronze).

Guiding principles should be adhered to at all costs. If a case is found that violated the guiding principles, they should be amended rather than an exception case being implemented.

## Biases

To meet the objectives of the Success Factors and adhere to the guiding principles, Rackspace recommends that {CUSTOMER\_NAME} to adopt the following Biases:

* Adopt API’s for integration such as ticketing with Manager Service Provider or with partner services.
  1. Ensure that API authentication is based on tokens, Oauth, API Keys, or some other means of authorization control.
  2. Adopt API gateways to help with security controls.
  3. Implement an API monitoring and logging solution to detect unusual API request behaviors.
  4. Enforce Rate Limiting
  5. Enforce encrypted transmissions via HTTPS.
* Comparable SaaS solutions are preferred over IaaS or PaaS solutions.
* Where possible and reasonable, PaaS solutions are preferred over IaaS solutions.
* Solutions should be deployed according to Vendor Supported Architectures
* Solution deployments should be automated.
  1. Use tools such as Ansible, Terraform, Chef, Morpheus, or others for configuration management and server deployment.
  2. Adopt a source code management tool such as enterprise github.
  3. Use automated build tools for CI/CD such as Jenkins, Spinnaker, Git Actions, ArgoCD, etc.
  4. Deploy automated security tools (Harbor, sysdig, mend, quails, or others)
  5. Centralize on a common repository for all artifacts such as artifactory
  6. Test for malicious code and opensource licenses and implement an automated code testing process using tools such as Rally/Tempest, Selenium, and/or sonar.
* Blue-Green rather than Dev/Test-Stg-Prod should be utilized for QA and deployment when applicable

Biases should be followed when appropriate. There will likely be application deployments which are architected and built contrary to one or more of the biases above. While this may be acceptable, {CUSTOMER\_NAME} IT must ensure that project teams who choose to build contrary to a Bias should adhere to the governance standards and controls of the {CUSTOMER\_NAME} policy framework.

## Staffing

Staffing can be broken down into four primary categories:

* Governance Committee (Typically referred to as as a CCOE)
  1. Note: {CUSTOMER\_NAME } has this in place through the corporate structure of the higher {CUSTOMER\_NAME } group. That group will need to be educated on Cloud standards to properly set the operation governance of the recommended environment.
* Program/Project Management
* Migration Architects/Engineers
* Pre/Post Migration Support Teams
* Operations/DevOps Teams

### Governance Committee

{CUSTOMER\_NAME} should form a Migration & Operational Governance Committee. This committee would include representation from the following groups:

* {CUSTOMER\_NAME} IT Management
* Project Management

The purpose of this committee is to periodically review and approve the documented Success Factors, Guiding Principles and Biases. As the project moves forward and more is leaned about the effort required and expected outcomes, this group must compare governance program with the strategic requirements of the enterprise to ensure that they are aligned. Additionally, where Biases are concerned, any biases that are regularly ignored in favor of alternate solutions should be considered for removal.

This committee is also responsible for determining the ongoing and ultimate overall success or failure of the migration project and operations.

It is important to note that this group does not make tactical decisions, they are responsible for strategic vision. Due to the technical nature of many members of this group, the desire to debate the merits of one technology over another will be present and leadership needs to encourage not having technology adoption discussions. Debating technology or solution decisions in this group will be counterproductive and result in a higher probability of overall project failure.

Due to the strategic nature of this committee, it should only meet monthly, or as needed to address the governing program, for example if/when a solution case is identified that contradicts a guiding principle.

### Program/Project Management

Any project with the complexity of this type of migration will require at least one Program Manager and one or more Project Managers. It could be possible that a single resource could fill both roles, however, depending on the rate of migration, a single resource may quickly find themselves overwhelmed and unable to respond in a timely fashion. As the primary role of a PM is to enhance communication, this would introduce significant risk into the project.

The Program Manager will oversee the overall migration effort, determining schedules, communication plans, and resource assignments. S/He will also be responsible for identifying project work that may impact decisions made by the Governance Committee and bring these issues to their attention.

Project Manager(s) will in turn manage individual application migration efforts. They may manage a single application or multiple applications at any time, depending on the complexity of the various projects. Project managers report to the Program Manager, and build schedules, communication plans and make resource assignments according to the overall program as designed by the Program Manager.

Rackspace recommends an individual who has the most familiarity with the {CUSTOMER\_NAME} IT estate should be retained on a full-time basis as the overall Program Manager/Project Manager for migration. Individual contributors within the organization may also be sourced as required.

During discovery workshop sessions it was uncovered that the {CUSTOMER\_NAME} employees have deep knowledge of their applications and {CUSTOMER\_NAME} is very well positioned to have representation of the application owners who understand the intricacies of the applications. A non-comprehensive list of experts that provided deep knowledge on {CUSTOMER\_NAME } ’s core in application workshop sessions are as follows:

|  |  |
| --- | --- |
| Application/Topic | {CUSTOMER\_NAME} Representative |
| Infrastructure and Operations |  |
| Application and Database |  |
| Application specialty Management and Operations |  |

### Migration Architects/Engineers

Each project will require a collection of Architect and Engineering resources who are familiar with the application or infrastructure that needs to be migrated to the new environment. The composition of this team will depend on the work breakdown of the project to which they are assigned. As it is typical that a single resource may be assigned to multiple projects based on his/her skillset, it is important that the Program Manager is aware of potential conflicts and can adequately direct Team Leads for resource scheduling.

The greatest opportunity for success exists when a project team includes members who are familiar with the existing IT estate, migration technologies to be employed and end state facilities/technologies. Training can be provided where necessary, but each team should include senior individuals who have intimate knowledge and experience.

### Pre/Post Migration Support Teams

In complex migrations, there are two separate support requirements.

1. Application stacks need to be supported in their legacy state until they are migrated, user acceptance testing (UAT) has been completed and business users are fully migrated to the new system.
2. Application stacks need to be supported through migration, UAT and beyond.

It is possible that the same teams can solve both requirements. In this case, it is likely that existing teams will require training on future state support practices and technologies.

## Vendor Support

For the success of any migration effort to hybrid-cloud target, contracts with the existing providers or suitable surrogates must be maintained until a migration is complete.

## Project Oversight

Few IT projects can have as sweeping an impact on the enterprise as a cloud migration. As a result, it is key that the proper team be organized to oversee a project of this magnitude. A cloud migration should not be viewed as strictly an IT project. Migrations require significant input and direction from business users and leadership as well as IT leadership. This will ensure that as decisions are made, they will reflect not only IT objectives, but more importantly the requirements of the business.

To ensure that business requirements as well as IT objectives are met, a project director who has a good understanding of the enterprise should be identified and placed in the role of Project Director. This individual would interface with all the various business units to ensure that the project outcomes meet each of their requirements. This person would direct all migration efforts and lead the Governance Committee.

## Staff Training/Documentation

To manage an automated Rackspace Private Cloud (RPC) based environment, there aren’t many new skills and expertise that will be required. Rackspace and {PARTNER\_NAME } will be the driving force partnering with the {CUSTOMER\_NAME} team in supporting the proposed OpenStack environment. Rackspace OpenStack engagements are designed where the operations IT team resources learn by watching and working with the Rackspace engineers. It is expected that although Rackspace is heavily engaged in operating the OpenStack platform at the initial deployment, overtime {CUSTOMER\_NAME} personnel will take over the majority of the cloud operations.

# CLOUD PLATFORM CONSIDERATIONS

## Automation and Infrastructure as Code (IaC)

Resources in the cloud are available and can be fully configured through application programming interfaces (APIs) provided by the OpenStack Platform. Command Line options are also available for integration with tools such as Ansible that will configure machine states through ansible scripting. This means that a cloud infrastructure can be precisely specified by code that uses APIs to create and configure the cloud infrastructure. An Infrastructure as Code (IaC) philosophy provides great advantages over the lifecycle of cloud infrastructure. By incurring the cost of developing the automation up front, many times the effort are saved later in a project.

It is possible to configure networks and compute instances in the OpenStack cloud manually, as if the cloud architecture were the same as physical machines. Planning Tenant and Network placements can be dynamic; however, it is recommended to design the network layout in advance. Assigning specific subnet ranges will allow for better integration with {CUSTOMER\_NAME } -HCM’s planned Cisco ACI deployment. Some things for {CUSTOMER\_NAME} to consider in network design is to allow tenants to still configure their own networks for special projects but also to have a set of mandated production, management, and other networks that are automatically attached to new VM instances as they are created. This will help to minimize confusion with application communication. Moreover, although it is possible to deploy networks manually, it is recommended that an automation approach be taken where the instances are documented in a git repo and the network ID’s are attached as part of the server creation process.

During the workshop discussions, we briefly discussed monitoring services and IaC scripting capabilities in the target cloud platform. However, to avoid any cloud lock-in and to maintain some portability, it is recommended that {CUSTOMER\_NAME} choose Terraform as its configuration management tool. Terraform is a cross-platform tool that interacts with other platforms such as GCP, VMware, AWS, OpenStack, and Azure. Terraform also interacts with several other services that are totally unrelated to a specific cloud provider. In cloud operations, rebuilding is not only frequent, it is advantageous. {CUSTOMER\_NAME} must plan for the eventuality that any piece of their architecture will be rebuilt at some point in the future. Beyond using automation to recover from errors at the service provider level, automation allows a company to make use of the cloud’s scalability and flexibility as a hosting environment as well as force the security and management controls to remain consistent on future builds. Automation when done with the correct tooling helps to remove human error and can be a great mechanism for enforcing operational controls. When provisioning an application today, {CUSTOMER\_NAME} will likely need to orchestrate the following as part of its deployment:

* Networks
* Load balancers
* Scaling groups
* Database layers (e.g., NoSQL or SQL databases)
* Persistent storage

Managing change is also easier in the target cloud platform. New servers can be created with a new configuration profile specific to updated requirements. The new configuration can be tested, and the servers can be destroyed. This cycle can repeat dozens of times until the desired quality is attained. Without automation, this process would be prohibitively costly. Automation is critical to take advantage of the flexibility hosting in the cloud provides.

Examples of a custom CI/CD pipeline to be followed for image management are as follows:

A diagram of a software process

Description automatically generated

Rackspace is also recommending that operations be setup at {CUSTOMER\_NAME} in a way that the infrastructure and application deployments are managed through automation tooling. Today updates are performed via manual login to the server and uploading of additional .dll or other files. The proposed operations model would follow a process where the code is checked into a code repository which would then star automated build pipelines. The build would trigger a deployment into dev or test for review. Upon UAT acceptance the code can then be pushed into production. An example of that process is as follows:

A computer screen shot of a computer

Description automatically generated

## Platform as a Service (PaaS)

OpenStack has many projects that help enable IT operations. OpenStack offers PaaS like services for automated workload provisioning. Services such as Trove for Database as a Service, Magnum for Container Orchestration, and Sahara for automated deployment of Big Data Services should be evaluated for integration into the OpenStack design. For example the time it takes to have a database created and configured could take days to weeks that consumes multiple DBA and other operations resources. OpenStack Trove is a Database as s Service for OpenStack that is designed to run on OpenStack with the goal of allowing users to quickly and easily utilize features of relational databases without having to be DBA’s. Trove provides easy operations for database creation, scaling, patching, backups, restores, and monitoring. Making Trove DBaaS available to users will help save countless hours in human operations time.

Rackspace will advise the use of PaaS solutions where they exist and where they are a good fit for {CUSTOMER\_NAME} . For complicated, scaling solutions the marginal ongoing cost is well worth the reduced risk and implementation cost of developing similar solutions within the company.

## Vendor Supported Architecture

A shift to a cloud operations model promises efficiency and the ability to scale IT operations. These efforts can be hamstrung by selecting systems and components that require significant support resources. Systems which are designed to work together require less management.

A bias toward vendor-supported architecture means specific preferences are expressed in cloud architecture designs:

1. Platform and software services supported by a cloud vendor are more desirable than solutions which require the creation of clusters or complex systems to be managed by Rackspace, {PARTNER\_NAME } or {CUSTOMER\_NAME} operations.
2. Reference architectures and standard configurations for systems are preferred when implementing complex systems.
3. Balancing open-source software with commercial retail applications requires a detailed analysis. Retail commercial off the shelf applications where vendor support contracts are available for support of the software is preferable to helping ensure application availability and uptime. Similarly adopting open-source software that is mature, widely adopted and popular should be considered as preferable if a company exists (or inhouse expertise exists) that can help provide guidance on configuring the software in such a way that availability and uptime is maximized. When adopting a company that supports open-source software, the preference should be towards a company that has not made customizations of the opensource where they offer their own version of the open-source software as this introduces a vendor lock in problem. In summary, {CUSTOMER\_NAME} should ensure that when adopting open source software that they can continue to receive updates from the opensource repos and not from the open source support vendor.

## Ephemeral Environments and Blue-Green Deployment

The ephemeral nature of cloud resources, and the ability to re-create them with IaC, can revolutionize change management. Where a production environment may consume three servers, an additional staging and development environment may collectively consume an additional 100-200% of those resources. These duplicates exist to test changes in one or multiple places before moving to production. While a tiered deployment model is a sound change management process, on the contrary, within a cloud environment, it is not necessary to devote so many resources to change management on an ongoing basis. Additional environments can be provisioned as needed.

Rackspace will adopt a posture to advise provisioning as few environments as possible to support change management. In an ideal scenario, development/test or staging environments will exist only while quality assurance tests are operating. Workloads will be deployed as needed and spun down when the operation is complete. Although the concept of Blue-Green deployments is understood and may be under consideration with {CUSTOMER\_NAME} , the anticipated use cases subject to deployment in the near term do not warrant the use of Ephemeral environments.

It should be noted that Blue-Green should not be the immediate posture for {CUSTOMER\_NAME} to adopt as the current application updates are done via installation and upgrades of dynamic link library (DLL) files for a majority of its applications. Blue-Green could be an adopted deployment methodology for production failover, however during discovery it was called out that maintenance windows are available for DLL updates and server reboots. As {CUSTOMER\_NAME} moves towards application sets that need to be designed for 24/7/365 availability then blue-green should become the standard.

## Best Practice Security Model

Cloud service providers enable and encourage a “least privilege” security model. In a “least privilege” environment, users and systems are segregated according to functional requirements. Different application workloads mandate different security controls. It is best practice to segment application workloads into similar roles/functions that impose the same security controls. Generally, network teams interpret segmentation as part of infrastructure design and apply network access control (VLANS, Subnets) and firewall rules to manage access. However, security zoning and network segmentation are two separate concepts. A network segment simply isolates one network from another. A security zone creates a group of system components that have similar security controls and requirements.

In a traditional environment, network segmentation is required. OpenStack networking at the host operating systems and network infrastructure can still be configured and deployed to support this type of deployment. However, this level of segmentation is augmented using Security Groups in OpenStack tenant environments. Security groups allow for granular control of security zones while simplifying overall design and management as well as allow the enforcement of both inbound and outbound traffic to instances in the group, regardless of the subnet or availability zone where those instances are located. Multiple security groups may be applied to a single instance and a single network interface. For automated deployments, security groups can be further defined in code which allows for SOC controls to be easily maintained by the SOC and easily implemented on new deployments. This allows you to apply both common and application specific controls to your instances, thereby, simplifying management while still providing granular control. Finally, each tenant’s networks are isolated to their specific tenants. Routing outside of the tenant is possible and the OpenStack controls will restrict tenants from breaking out of their environments without intervention from a designated architecture/network team to help assist with cloud routers. This type of tenant level control helps to isolate security and minimize the blast radius due to compromised systems.

Every security group should only provide the required inbound and outbound port access needed for that specific application/role. A set of global and workload common security groups should be applied which will provide the basis for the security group framework. Additional application specific security groups can then be applied to the instances to allow traffic for application specific ports. Security groups not only allow enforcement of inbound/outbound traffic based on source/destination IPs, but also secures instance traffic based on source/destination security groups.

{CUSTOMER\_NAME} should plan for the necessary monitor alerts to be triggered when specific log audit events occur. For example, if there is a Security Group defined in code and the ingress/egress rules change, the log detection rule will automatically send notifications alerting the security team as rules should be managed via automation and not manual changes. These rulesets should be configured to ensure the security group framework continues to maintain compliance by sending notifications when an IAM user opens critical ports that may compromise platform security.

Rackspace advises proper and thorough analysis of the business needs for {CUSTOMER\_NAME} ’s applications to allow the creation of rules that follow a “least access privilege” model. The rules applied, and the classification of staff and services should be subject to regular scheduled review to ensure the network design continues to match business needs and continues to observe a “least access privilege” model.

## Key Management

OpenStack offers Barbican, a native PaaS service for Key Management that works as a secure secrets store. Barbican allows the creation of multiple secure containers called vaults. Barbican by itself offers management of secred data such as passwords, encryption keys, x.509 certificates and raw binary data. It is recommended that an HSM be used in conjunction with barbican. Barbican should only be considered if {CUSTOMER\_NAME} intends to manage all deployments on OpenStack. If {CUSTOMER\_NAME} is planning to run applications on other platforms such as VMware, or a public hyperscaler then Rackspace recommends looking at other tooling that will better support cross platform integration. Other tooling to consider include Hashii Vault as a commercial offering. CNCF spiffe or CNFF Spire should be considered as opensource offerings.

## Network

{PARTNER\_NAME } in conjunction with {CUSTOMER\_NAME} are in the process of implementing and configuring Cisco ACI for network controls. OpenStack will plugin to the deployed tooling based on {CUSTOMER\_NAME } ’s network design. OpenStack deployments will be a series of bonded NIC’s from the host systems setup in VLAN’s. An example deployment of the network vlans is as follows:

A diagram of a computer network

Description automatically generated

{NETWORK\_DIAGRAM}

# **BEST-FIT CLOUD DECISION MATRIX**

Rackspace evaluated several options for private cloud in determining the best fit for {CUSTOMER\_NAME} . Of all possible options VMware (a commercial retail offering) and OpenStack (an OpenSource) offering were evaluated. The decision matrix takes into consideration the benefits of each offering. All other private cloud platforms were eliminated as they were deemed not aligned based on the criteria below. Hyperscaler offerings were eliminated from the decision matrix due to requirements for keeping data in Vietnam and each hyperscaler not having an adequate offering located within Vietnam. The goal is to provide a platform that alows existing applications to easily be migrated into it while providing a future cloud operations model. OpenStack was ultimately chosen due to its lower cost and ease of operations.

**{DECISION\_MATRIX\_TABLE}**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Legend | **Best** | **Good** | **Acceptable** | **Equal** |

| **Criteria** | **OpenStack Private Cloud** | | **VMware VCF with VRO and VRA** | |
| --- | --- | --- | --- | --- |
| **Performance** |  | Highly performant infrastructure |  | Highly performant infrastructure |
| **Reliability** |  | Equally reliable |  | Equally reliable |
| **Resiliency** |  | Equally resilient |  | Equally resilient |
| **Scalability** |  | Provision on-demand basis (requires hosts to exist) |  | Provision on-demand basis (requires hosts to exist) |
| **Security** |  | Equally secure |  | Equally secure |
| **Compute** |  | Any commercially available hardware that supports Ubuntu distributions.   * IaaS: Virtual Machines * Managed Hypervisor KVM for VM migrations from unhealthy to healthy nodes * Additional OpenStack PaaS Services available * Native and ease of integration with Kubernetes services, no additional license fees |  | Any commercially available hardware that supports VMware’s ESX hypervisor.   * IaaS: Virtual Machine * ESX hypervisor for vMotion of migrations of VM’s from unhealthy to healthy nodes * Offers a robust Kubernetes solution however it is a VMware modified offering and not a native CNCF deployment. Additional license fees apply. |
| **Networking** |  | * Compatibility with any hardware that supports OVN/OVS. Support for complex |  | * Compatibility with any hardware that supports NSX-T |
| **Database Services** |  | * Flexible options to host {CUSTOMER\_NAME} ’s DB’s providing support and control. This also helps in maintain a predictable cost. * Offers DBaaS tooling |  | * Limitations on vendors that will support databases running on VMware’s hypervisor |
|  |
|  |
|  |
|  |
|  |
| **Storage Services** |  | * Support for any hardware that has an OpenStack Cinder API plugin. |  | * Support for any hardware that has a compatible VMware plugin. |  |
| **Financial** |  | * Predictable and flexible financial model. * Costs per node are the cost of additional hardware nodes. * No per-core or per-socker licensing allowing for dense host deployments. * If consumed via a managed service provider such as Rackspace the cost per node managed is known and predictable. * No feeds for orchestration tooling |  | * Licensing costs for VMware are subscription based. * Licensing is based per # of sockets which can lead to a significant license costs spike when deploying dense CPU nodes or blade servers. * Additional services for orchestration such as vRA, vRO, PowerCLI, Tanzu Kubernetes Grid, VCF, Cloud Automation Services, and AutoDeploy require additional licensing with some being per-VM based leading into increased expenses to implement automation. * Unknown future of predictability for costs or platform releases due to Broadcom acquisition |  |
| **Latency** |  | Based on {CUSTOMER\_NAME} network topology, OpenStack deployments should not inherently introduce additional latency factors |  | Based on {CUSTOMER\_NAME} network topology, VMware deployments should not inherently introduce additional latency factors |  |
|  |
|  |
| **Manageability** |  | * Requires a team of IT operations personnel to keep up with maintenance and operations tasks. * OpenStack skillset is a harder skill set to find compared to VMware operators. * Operators with backgrounds in recommended tooling such as Terraform, Ansible, and other opensource tooling are readily abundant. * No limitations on customization and extensibility when compared to VMware service limits. |  | * Requires a team of IT operations personnel to keep up with maintenance and operations tasks. * VMware vRealize, vRops, VCF and other skillsets are harder skillsets to find. |  |
| **Vendor Lock-in** |  | * OpenStack is a community developed Open Source project that is free and public to any person or corporation. * Rackspace deploys standard OpenStack community master branch which will allow {CUSTOMER\_NAME} to replace Rackspace and operate the environment with their own staff or any chosen provider that supports open source OpenStack |  | * VMware might not be ideal in the long run due to vendor lock-in, which can limit {CUSTOMER\_NAME} choices and potentially increase costs over time. * Limited Choices: Because VMware is a commercial product, {CUSTOMER\_NAME} will be tied to their ecosystem. This will restrict the ability to explore or switch to alternative virtualization solutions that might better suited to {CUSTOMER\_NAME} needs. i.e. K8S on VMware requires Tanzu licenses versus deploying a native CNCF based K8S environment. * Potential Cost Inflation: Vendor lock-in can give VMware more leverage when it comes to pricing. Without the ability to easily switch to competitors, {CUSTOMER\_NAME} might be less likely to negotiate favorable pricing or be forced to accept price increases with limited options. |  |
| **Migration** |  | * Tooling exists that can be leveraged to easily migrate VM’s from VMware to OpenStack. * Time and cost will need to be considered for changing the way that {CUSTOMER\_NAME } will operate its applications on top of OpenStack as {CUSTOMER\_NAME} moves to a cloud like operating model |  | * No migration required as {CUSTOMER\_NAME} has a current VMware implementation. * Significant cost and time will need to be spent into moving from the current operating model into a cloud like operating model. |  |
| **DR Solution** |  | * Control plane services are designed for HA and can be stretched across regions. * OpenStack can be stretched across clusters/regions however failover is not as fast as VMware’s failover scenarios. * Replication tooling is managed via storage susbystems, Ceph, or agent-based replication are required. There is not a portal based native replication tool integrated with OpenStack like VMware has. * Third party solutions that integrated with the OpenStack services can be leveraged to help with large complex failover scenarios. |  | * Native orchestration tools exist to make DR easy. SRM, vSphere Replication, and Stretched clusters make for an easy integrated approach to DR services via a portal offering. * {CUSTOMER\_NAME} utilizes stretched clusters today which allows for very low RTO and RPO failover scenarios. * Additional costs apply to setup DR services. * Third party solutions such as Xerto and Veeam are available that integrate with vSphere replication are available to help with large complex failover scenarios. |  |
|  |
| **Licensing** |  | * OpenSource – no licensing fees. |  | * Subscription based. Additional services required additional licensing fees. |  |
|  |
| **Technology Alignment** |  | * Better aligned for cloud operation technology integration for companies looking to adopt CI/CD and devops operational practices. |  | * Moderate – Refactor/Re-Architecture may be required to fit into VMware’s VCF, VRA, and VRO tooling. |  |
| **Hybrid Platform** |  | * Flexibility to extend and connect with any Hyperscaler and other technologies. |  | * Flexibility to extend and connect with any Hyperscaler and other technologies. |  |
| **Investment** |  | * This will require training of {CUSTOMER\_NAME} staff on OpenStack tooling. * OpenStack is designed to be operated as a fully automated open-source platform with integration to other popular opensource software. * {CUSTOMER\_NAME} will not become locked into a retail ecosystem. |  | * Continue to leverage VMware * This will require training {CUSTOMER\_NAME} staff on VMware VCF, VRA, VRO and other tooling to move into a cloud operating model. * Become locked into the VMware retail ecosystem. |  |
| **Conclusion** |  | * OpenStack is a compelling alternative to VMware. OpenStack is an open-source cloud computing platform that provides infrastructure-as-a-service (IaaS) solutions and PaaS solutions for building and managing private and public clouds. Implementing disaster recovery (DR) is not as simple compared to VMware, however OpenStack. However, OpenStack is a compelling alternative to VMware due to its ability to operate at enterprise scale, be cost efficient, offers no vendor lock-in, and helps organizations to drive innovation by offering an agile by adopting a cloud first operating model while helping to reduce and optimize costs associated with a retail platform such as VMware. * OpenStack is proven to be Enterprise ready as it is being used by large companies such as CERN, Comcast, USA NSA, Best Buy, Yahoo, Blizzard Gaming, Sovereign Cloud for the Government of Saudi Arabia, Walmart, AT&T, PayPal, China Mobile, Asia Commercial Bank, and many other large global organizations. * OpenStack and cloud operations will be a new skill set for {CUSTOMER\_NAME} operations personnel to learn and will require training and operational changes to implement. |  | * VMware has been a trusted solution for virtualization for 20+ years. VMware has a reputation for providing a platform with proven reliability and stability. VMware offers advanced disaster recovery solutions and VMware’s stretched cluster offering currently implemented by {CUSTOMER\_NAME} is the best deployment standard for operational resiliency that can implemented today. * VMware is already deployed at {CUSTOMER\_NAME} which will mitigate the costs and time of having to do a full environment migration. However, applications will need to be redeployed as additional cloud management tooling is adopted. * VMware VCF, VRO, VRA, and cloud operations will be a new skill set for {CUSTOMER\_NAME} operations personnel to learn and will require training and operational changes to implement. There will be additional costs to adopt the VMware automation tooling. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Legend | **Best** | **Good** | **Acceptable** | **Equal** |

## Target Platform Recommandation

During the analysis we have found that the applications assessed in the workshops can be migrated into the OpenStack platform and operated in a similar fashion to how they are operated today. This will allow {CUSTOMER\_NAME} to continue to operate their VM’s in a similar operational model while moving away from the high costs associated with VMware licensing. However, it is recommended that when possible, the applications be refactored in how the server instances are deployed and managed. Tools such as Terraform should be configured to help define how windows and linux VM’s are configured. Additional tooling such as TFS or Chocolatey should be considered for code packaging and deployments in an automated fashion. This one of the strong reasons why Rackspace is proposing the OpenStack Private Cloud platform. The following table summarizes the benefits and constraints between OpenStack and VMware.

| **Target Platform Benefits & Constraints** | | |
| --- | --- | --- |
| **Target Platform** | **Benefits** | **Constraints** |
| All workloads in OpenStack Cloud | * Ability to host all workloads within one environment. * Path to modernize legacy application operations. * Lower operational cost | * {CUSTOMER\_NAME} stretched cluster is the ideal standard for Disaster Recovery and the method for Disaster Recovery will need to be modified. |
| All workloads in VMware | * Existing deployment already exists * Path to modernize legacy application operations. | * Additional License costs to enable automation tooling. * DR solution may require additional work. * Higher license costs associated with using VMware VCF, VRO, Tanzu, and other services offerings that are natively part of OpenStack. |

Rackspace recommends OpenStack as a target cloud platform for manageability, risk, reduction of costs, flexibility, portability, reduction of vendor locking, and its ability to modernize {CUSTOMER\_NAME} operations to a cloud operating model

# MIGRATION TIMELINE AND MOVE GROUPING

Detailed application migration planning is not in scope for this writeup however a migration overview is being included as {CUSTOMER\_NAME} decision makers should be aware of the process for moving to a cloud-based operations environment. The planned migration for {CUSTOMER\_NAME} should be based on the number of server instances that met the Cloud Readiness criteria. Each of the licensed servers is assigned a Landing Zone, a Disposition value, and a Move Group number. Will provide report if tooling is run, pending discovery tooling.

## Disposition

A disposition value corresponds to one of the six most common application migration strategies: Rehost, Replatform, Repurchase, Refactor, Retire, and Retain as illustrated in the following diagram. The disposition values for the applications interviewed was provided in a previous section. A diagram of a diagram

Description automatically generated

The focus of this assessment is on Rehost, Retire, and Retain. Refactoring involves redesign and rewriting applications, while Repurchase involves changing the current licensing model.

* Rehost (Lift and Shift): This involves simply moving your application "as-is" from its current environment (often on-premises) to a new environment, typically a cloud platform cloud. It's a quick and cost-effective approach, however, it does not leverage the full benefits of modernization. Most Rehosting can be automated with tools.
  + Note: DevOps would further determine the best path to migrate with transformation or without transformation.
* Replatform: Migrate the application logic to a new platform while potentially keeping the same underlying codebase. This could involve moving to a different operating system, containerizing the application, or using a cloud-native platform. Re-platforming can offer some improvements in scalability and manageability.
* Refactor/Rearchitect: This is a more involved approach where the application code base is modified to improve its architecture and operational resilience. Typically, monolithic applications are broken down to microservices or other cloud technologies are leveraged to provide better flexibility, scalability, and maintainability of the application.
* Repurchase: Sometimes, the most efficient solution is to abandon the current application and purchase a commercially available Software-as-a-Service (SaaS) solution that performs a similar function.
* Retire: If an application is no longer critical or has low usage, it might be best to simply retire it. This frees up resources and reduces maintenance overhead. Care should be taken to ensure that the service is decommissioned in a fashion that is in line with the current procedure of retiring a platform.
* Retain: If an application functions well or not a good fit for a different platform there may be no immediate need to modernize it. It is still important to consider these types of applications for future re-platforming and rearchitecting activities.
  + Based on the interview workshops the only servers out of scope for migration to the OpenStack environment are those running on non x86 platforms such as the Sparc systems.

The definitions below provides guidance on optimize benefits against each category moving to the cloud.

| Category | Migration Timeline | Hosting / Migration Complexity | Resource Demand | Cloud features leverage and cost advantages |
| --- | --- | --- | --- | --- |
| Re-host   * Lift-And-Shift * No code modification | Fastest | Low | Low | Minimal |
| Re-Platform   * Migration with minor changes (e.g. upgrading server versions) * Positions to leverage base cloud functionality and cost advantages | Medium | Medium | Medium | Basic |
| Re-factor:   * Re-architect/re-code/sw upgrade | Long | High | High | Minimal  Sophisticated |
| Re-purchase:   * New product licensing * Replace – drop and Shop | Long | High | High |  |
| Retire | N/A | N/A | N/A | N/A |

## Rehost, Retain, Refactor Recommendations

Considering {CUSTOMER\_NAME} ’s application portfolio footprint, Rackspace recommends following “Lift, Tinker & Shift” model followed by updates to servers and applications. In this model, the core applications will be migrated as is which allows {CUSTOMER\_NAME} personnel to become familiar with the OpenStack platform. This will significantly reduce risks with migration and lets {CUSTOMER\_NAME}  move off of VMware and into OpenStack quickly. Once the workloads are deployed in OpenStack, Rackspace recommends refactoring automation into the management of the applications for application updates.

**{APP\_MIG\_PLAN}**

Example

|  |  |  |
| --- | --- | --- |
| Infrastructure Cloud Readiness | | |
| Application | Recommendation | Comments |
| GIS | Rehost first, Refactor after migration | Oracle is the primary backend database and should be left in its current operating environment. The application servers for GIS are in redundant data centers and can be remapped into an OpenStack environment setup in each datacenter (Region). There is a file server associated with this application, suspected to be a DFS-R deployment. DFS-R will deploy into OpenStack without issues however DFS-R has file locking limitations that would be an inhibitor to a future active/active deployment and that portion of the application should be refactored to leverage an objectstorage solution. Not objectstore based solutions such as FreeFileSync(OpenSource) or Resilio(commercial package) can also be explored as alternatives. |
| Doffice | Rehost first, Refactor after migration | This application is already close to being a cloud based application as it uses RabbitMQ, API tiers for requests, and web frontends behind a load balancer. Once migrated the mechanism for storing documentation that is submitted needs to be factored similar to the recommendation for GIS. |
| HRMS | Rehost first, Refactor after migration | Standard 3 tier application that will run without issues on OpenStack. Currently an active/passive deployment in (2) DC’s for the database backend. Getting to active/active would require enabling always-on and also devising a mechanism for user session management. There will need to be some work to refactor this application for user session state. |
| KPI | Rehost first, Refactor after migration | Standard 3 tier application deployment that will run without issues on OpenStack. The database is an always-on WSQL deployment which fits into an active/active design pattern. The application code is managed in github however updates are manual code installs. Automation for code deployment should be utilized once migrated into OpenStack. |
| CMIS | Rehost first, Refactor after migration | Extremely critical application. Oracle is the primary backend database and should be left in its current operating environment. There are components of this application that ready for cloud operations such as the API and weblogic tier. There are sFTP components that should be updated for capturing uploads in a more secure and modern fashion. Additionally, the applications are patched with manual .dll file updates that can be automated into a CI/CD process. |
| PMIS | Rehost first, Refactor after migration | 2 tier application of web and database. The application will rehost without issues. The applications are patched with manual .dll file updates that can be automated into a CI/CD process. The database tier is a wSQL deployment using always-on replication which lends itself to an active/active design pattern. |
| LiveStream | Rehost | Special network design will need to be considered for this application due to the amount of potential video streams and bandwidth that will be consumed. |
| ERP | Retain | Currently this is an oracle-based application with Oracle as the backend database and the EBS modules running on sparc instances. The application can be refactored to run on x86 but would not be a good first candidate mover. |
| Dashboard | Rehost | Data sources are ingested from other systems. It may be possible to containerize this application quickly based on the reporting features that it presents. |
| HES-MDIS | Rehost | Standard 3 Tier application however the app tier is an API layer that would lend itself into a good cloud design pattern. Data comes from multiple sources and is aggregated into an MQ system. All servers are stateless which lends this into being a good cloud design application. The backend database servers are oracle based and should be retained on the oracle platform. Due to the design of this application it fits well into a cloud design pattern for horizontal scaling and future automation should be quickly adopted to monitor for scale up and scale down events. |
| Portal | Rehost | Two tier application of web and database. The database tier was undefined. There is a documentation capture process that happens and it is unclear if the documents are stored as objects in the database or in a file store. In either case the application functionality for document capture should be considered to be changed into an object store location similar to what was recommended for GIS |
| Digital Cert | Rehost | Oracle is the primary backend database and should be left in its current operating environment. Information on how the application operate was not provided. Based on the underlying technology there are no blockers found for this to run in an OpenStack environment. Application specific recommendations will need to be provided in a future engagement at {CUSTOMER\_NAME} ’s request. |
| BIM | Rehost | Web based application with a front end server and backend server that fit into a standard cloud design pattern. The applications are patched with manual .dll file updates that can be automated into a CI/CD process. The backend database is SQL and should be retained in the SQL deployment. |

## Move Groups

A “move group” or “migration wave” is a group of servers or applications that need to be migrated together to reduce overall business impact based on interdependencies or other criteria. Recommended strategy would be to move or provision infrastructure components, databases and all others sequentially.

A detailed project plan would be developed outlining timelines based on recommendation for migration and transformation strategy.

## {Migration Strategies and Roadmap}

The following table summarizes the migration strategy and the associated risk profile.

{mig\_table}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Migration Type | Timeline | Strategy | Risk | Approach |
| Lift & Shift | Immediate | * Move all workloads (AS-IS) without platform upgrade * Perform in-place upgrades to O/S and transform applications. * Achieve 100% supportable state | Low | Tool based |
| Modernize & Migrate | Depends on effort involved in modernizing / Re-platforming Apps. | * Perform upgrades to servers and applications * Migrate to target platform * Achieve 100% supportable state | High | Tool based |

Through the discovery process, the table below shows the findings and suitability of applications for Cloud Readiness. For the first phase of modernization a rehost approach best aligns with the modernization objectives and offers the lowest risk and shortest time migration path. Once the applications are rehosted into the OpenStack deployment it is recommended that automation be a fast follow-on phase to deploy application updates and for server configuration.

# {DISASTER RECOVERY}

A complete Disaster Recovery solution was not in scope for this statement of work, however Disaster Recovery is a vital component of operational resiliency and guidance is being reported in this section of the report as it will be an important factor for {CUSTOMER\_NAME} to consider when deciding on a final solution for their private cloud needs.

{CUSTOMER\_NAME} has {DC\_NB} of data centers within a metropolitan region that are being used for production and Disaster Recovery (DR). The current solution uses VMware stretched clusters with replicated storage which allows for an active/active deployment and the automatic failover of instances across data centers within the metropolitan area. These types of DR deployments are extremely costly to build and maintain and without the proper standardization for each region in place, operational parity is extremely difficult to achieve, the VMware solution allows for the simplicity of near synchronous replication and a fast failover in the event of an unplanned outage.

There are many possibilities that need to be explored with {CUSTOMER\_NAME} for a final Disaster Recovery (DR) strategy. Disaster recovery options range from using native OpenStack tooling to commercial off the shelf products along with a mix of native projects and commercial software. This section will describe the OpenStack alternatives; however Rackspace recommends that {CUSTOMER\_NAME} continues with the tools that it is already using for backups to help minimize operational changes and complexity of deployments. The only viable reason for using an **OpenStack freezer service for backup would be to save on licensing costs associated with current backup software.**

**{DR\_METHOD}**

Example

The default project for backup and disaster recovery in OpenStack is the OpenStack Freezer project and OpenStack Senlin with Masakari are the default projects for HA cluster management. Freezer is designed to integrate specifically with OpenStack on KVM environments. By default, freezer provides an automated policy-based backup, fill and incremental backups, and full restores. The target for freezer is any storage device compatible with cinder or remote storage that can be connected to via ssh. In a freezer design the backups for the production environment will be captured at the primary source datacenter and replicated to the secondary target datacenter. In the event of an incident Freezer can be leveraged to rehydrate the backed-up data. It should be noted that OpenStack Freezer does not have deduplication capabilities or as robust of a backup administrator interface compared to current commercial software offerings.

Several commercial off the shelf (Retail) options exist for backup services including Commvault, Acronis, Rubrik, Veritas, Veeam and many more. Rackspace recommends that {CUSTOMER\_NAME} continues with its current provider.

A final disaster recovery option will need to be confirmed with {CUSTOMER\_NAME} once the full design and migration phase starts. There are two viable Disaster recovery scenarios that should be considered.

Disaster Recovery scenario one will be an all OpenStack solution where the PureStorage that {CUSTOMER\_NAME} uses will be configured into a Metro Cluster deployment and OpenStack control plane services will be stretched across both datacenters with a (5) node control plane services deployment. MQ, Galera, and other components will be replicated across and the volumes in the SAN will be known to the database replicated across regions. KVM will handle any VM migrations due to host failure and in the unlikely event of a catastrophic disaster where the primary datacenter is lost the instances will be restored in the target Disaster Recovery site. This type of setup is similar to the stretched VMware cluster solution and will require a layer 2 VLAN across both locations.

The second disaster recovery scenario will be a deployment where OpenStack is deployed in each datacenter as its own stand-alone cluster with a (3) node control plain in each datacenter. Agent based tooling for virtual machine level replication will be utilized to synchronize server state across datacenters. Currently {CUSTOMER\_NAME} has Veeam which is compatible with OpenStack, and it would be anticipated that {CUSTOMER\_NAME} continues leveraging Veeam for the data replication and failover. Other commercial options that can be considered that are designed to work specifically with OpenStack deployments include Hystax, Storware, Carbonite(Doubletake), Vinchin, and Trilio.

### SLA Objectives

This will pave the way to gain cost efficiencies in the new DR solution. This can be accomplished with two key metrics, recovery time objective (RTO) and recovery point objective (RPO).  RTO is the duration of time within which an application or business process must be restored after a disaster, while RPO describes the interval of time during a disruption before the quantity of data loss becomes intolerable.

The following table is an example list all known RTO and RPO numbers for major applications depicts per application DR plan based on information gathered during the onsite workshops. It should be noted that the RTO’s and RPO’s are not officially mandated by {CUSTOMER\_NAME} . In the course of discovery, a formal statement on requirements was not available and interviewees were asked to help provide an estimate based on their knowledge of the business and the applications.

{SLA\_APPS}

|  |  |  |  |
| --- | --- | --- | --- |
| Application | RTO | RPO | Notes |
| GIS | 24 Hours | No Loss | Not listed as a critical application however GIS information does feed into BMIS and PMIS. Operates on Solaris and Oracle – resiliency is built into the Oracle replication. Oracle failover is not automatic and is set as a manual failover. |
| Doffice | 24 Hours | 48 hours | Public Intrantet application with front end web server and ocelot API server. Essentially a queue, data is stored in the Oracle Database server. |
| HRMS | < 24 Hours | < 24 hours | Multiple Modules for personnel, recruitment, contracts, salary, offboarding, etc. Microsoft SQL backend database. |
| KPI | 24 Hours | 1 week | Work performance for all employees. The impact to this system would be dependent on the time of year. |
| CMIS | < 2 Hours | < 15 minutes | CMIS is multiple modules integrated into portal and other customer service roles. |
| PMIS | < 2 Hours | < 2 Hours | Power and Grid technical management. Very critical for operations. MSSQL backend, heavily utilized server, replication solution should be full 1:1 production to DR for hardware. |
| LiveStream | 24 Hours | N/A | Livestream data required for employee occupational safety for field workers. Workers can technically function without it however it does open up concerns for employe risk exposure. |
| ERP | < 2 Hours | < 30 Minutes | Highly critical application for {CUSTOMER\_NAME } operations. 17 Modules deployed to support billing and financial services. Modules include Purchasing (PO), warehouse and material management (INV), Payables (AP), Receivables (AR), Money Management (CM), Taxes (TA), Fixes Assets Management (FA), general ledgers and accounting (GL), Cost aggregation and calculation (CO), Financial planning (BU), Financial Supervisory(FS), Payroll (PR), Loan Management (LE), Financial Investment management (EM), Financial Analysis (FI), Consolidated financial statements for all of {CUSTOMER\_NAME } (HN\_ {CUSTOMER\_NAME } ). Currently deploying e-payment services that will connect to ERP.  Backend Database is Oracle 11G and should be updated. |
| Dashboard | 24 Hours | N/A | 2 modules that produce dashboards for the Board of Directors and the Member’s Council of {CITY NAME } Power Corporation. Current deployment is a single point of failure however it can be recreated in a reasonable amount of time. Not deemed critical but treated as critical due to information that it shares and the level of people that consume the information. RPO is not applicable as the data it presents is loaded from the CMIS and PMIS systems. |
| HES-MDIS | < 24 Hours | 0 | Internal telemetry system that collects telemetry from multiple datapoints over a 3G and other networks. Considered a critical application as it feeds into customer billing and other systems. Everything is over private network and the network endpoints are established in both DC’s. RPO should be as close to no data loss as possible. |
| Portal | < 8 Hours | < 24 Hours | RPO was debated because of source feeds, generally less than 24 hours of data is consistent with the data required for decision making |
| Digital Cert | 24 Hours | 24 Hours | Critical application for functionality of systems, the full impact would only be felt in the event of a disaster just as a certificate was being renewed |
| BIM | 48 Horus | 72 Hours | BIM is business process management. The BIM team is focused on digitizing the processes, and they are connecting into several different systems based on the task assigned. For example, if there is a purchase of information then that would generate the office report. Not deemed a critical application and currently only a single server in the primary DC. SQL backend database that can be replicated but is not currently replicated. |

Based on the RTO/RPO numbers above and the various requirements in different regions, {CUSTOMER\_NAME} can incorporate a hybrid model for its DR needs; however, a full replication and metro cluster is readily available, and that solution should be preferred. A Hybrid model should be adopted if {CUSTOMER\_NAME} moves its disaster recovery DC to a farther location greater than 20km as anything past 20km would not allow for a performance stretched cluster.

### Recommendations

1. Rackspace recommends establishing SLA objectives per application based on their business value and criticality to the business. The value should be based on the cost of downtime and criticality Tiers should be established to help provide guidance on DR solutioning.
   1. Example:
   2. Tier – 1: Any application with a less than 4 hour RTO and less than 60 Minute RPO
   3. Tier – 2: Any application with a less than 8 hour RTO and less than 4 Hour RPO
   4. Tier – 3: Any application with a less than 24 Hour RTO and Less than 12 Hour RPO.
   5. Tier – 4: Any application with a less than 48 Hour RTO and less than 24 Hour RPO.
   6. Tier – 5: Any application greater than 48 Hour RTO and greater than 24 Hour RPO.
2. Expand the duties of the emergency response group to conducting table top exercises for applications and possible emergency response scenarios only a monthly basis.
3. Conduct a failover event tests for applications at least once per year for all critical applications in Tier’s 1 and 2. Tier’s 3 and 4 are also recommended to test on a yearly basis.
4. Further document failover procedures for all critical applications
5. Define possible disaster scenarios for the region to help define impact scenarios and create plans based on the possible impact of each scenario.
   1. Examples: Train carrying explosive chemicals, power plant explosion, power outage, Earthquake, flooding due to tropical storms and typhoons, storm surges, heatwaves, pollution.
   2. Although {CITY NAME } is not an earthquake prone city it is subject to heatwaves and urban heat island effects. Excessive heat has been a significant cause of data center outages due to chiller and air conditioning failures forcing a shut down of servers. Misters for top of roof equipment and portable AC units to have on standby are highly recommended. Similarly, the DC locations should be assessed for likelihood of flooding.
6. Consider moving disaster recovery to a datacenter farther than 10km if the scenario planning identifies a catastrophic event that would impact both data centers simultaneously.
7. Identify application interdependencies to help define application criticality.
   1. The profiling software used to asses the environment for move readiness can provide this information.

# SUMMARY OF RECOMMENDATIONS (In Scope and Out of Scope)

Upon conducting a comprehensive assessment as outlined in the Statement of Work (SOW), Rackspace and {PARTNER\_NAME } have identified certain aspects that were not explicitly within the predefined scope of our engagement. However, in the interest of thoroughness and ensuring a comprehensive understanding of the situation, we have taken the liberty of including these findings in the assessment report. We believe that providing these insights and findings into additional areas will contribute to helping improve {CUSTOMER\_NAME} ’s operations and help to assist in the decision-making moving forward.

The overall observation of {CUSTOMER\_NAME} ’s IT operations is that it is a mature organization that adheres to controlled operational standards. Over the course of the assessments several factors were assessed as follows:

* Access Controls: {CUSTOMER\_NAME} has defined policies for granting access to IT systems including user authentication and privilege Management. A bastion environment was found to exist to help with user authentication and controls to servers.
* Change Management Processes: The change management process is documented and includes steps for planning, testing, approval, and documentation of changes to the environment. Feedback was captured from various groups so that at the end of a project any issues are captured, and a formal report is generated. Monthly and quarterly meetings capture these issues and recommendations are put forth for fixes if needed.
* Incident Response Plan: {CUSTOMER\_NAME} has an emergency response group that assembles for incidents, and it has tooling in place for detecting to and responding to incidents, however additional focus (outlined in DR section) should be applied to scenario-based planning and tabletop exercises.
* Backup and Recovery: The backup software used is mature, backed up to storage and replicated to a secondary site. Additional steps are outlined below for mitigation against ransomware.
* Patch Management: {CUSTOMER\_NAME} follows a regular patch management cadence on the x86 platform. Several database components are out of date on the Oracle instances.
* Configuration Management: Configuration management is older based on VMware technologies. The current system is adequate for current needs and should be modernized by implementing more automation controls. The controls are outlined in the OpenStack recommendations of this report. If OpenStack is not adopted then licensing and implementing the additional Aria and VCF VMware suite is highly recommended.
* Physical Security Controls: Access to the business buildings is controlled by perimeter security with internal security to check on additional entry. Workspaces are locked down to non-employee entry. DC entry was not assessed.
* Disaster Recovery and Business Continuity Planning: Disaster Recovery has been considered in the stretch cluster design of the current infrastructure. The optimal design is a stretched cluster design for ensuring maximum availability and uptime of the environment, however there is an inherent risk in the DR facility potentially being impacted by the same event that impacts the primary facility. Recommendations have been made in the Disaster Recovery section above and should be considered.
* Compliance and Regulator Controls: {CUSTOMER\_NAME } -General has a group that evaluates relevant laws, regulations, and industry standards governing IT systems and their use within the electrical distribution and power creation areas. The controls are well documented, and they mandate specifics for deployment and operations of the IT environment. {CUSTOMER\_NAME} follows the mandated controls without deviation.
* Operations Governance: {CUSTOMER\_NAME} is ISO-27001 certified as the core operational standard for compliance and Tier-3 is the minimum requirement for the Data Center standards.

## Recommendations Introduction

Detailed recommendations based on the workshop interviews and {Audit tools } tooling that are outlined in this report can be summarized as follows:

## Cloud Center of Excellence

Create a cloud center of excellence (CCoE). A CCoE is a team or group within an organization that focuses on leading and governing the adoption of cloud computing in a strategic way. The primary objective of the CCoE is to ensure that the organization gets the most value out of the cloud technologies implemented while minimizing risk and defining operational controls. A CCoE is typically a cross-functional team with members from various departments such as central IT, security, architecture, and where appropriate some business units. The CCoE will have responsibility for defining cloud adoption strategies and best practices, establishing governance policies and security measured, providing training and support for cloud users, evaluating and selecting cloud service components and providers, and optimizing cloud resource utilization and managing costs.

Overall, a CCoE acts as a central hub for guiding and overseeing the organization's journey towards successful cloud adoption. The CCoE can also act as the migration of the current technologies into the recommended OpenStack cloud deployment. As part of the migration the CCoE would operate as follows:

1. Organize Project Governance and adopt Guiding Principles and Strategic Biases
2. Extend existing support contract through the migration.
3. Setup a governance board that maps key {CUSTOMER\_NAME} personnel to the core migration sponsors. For example:

A diagram of company's company's company

Description automatically generated

{Gov\_Diagram}

## User Management, Governance, and Access Control

1. Continue use of current LDAP and AD controls and integrate them into Keystone for the proposed OpenStack solution.
2. Servers are still highly manually controlled and minimal automation exists, which violates the principles of least privilege controls. Service accounts should be created for logging into systems and configuration management software tooling such as Terraform or SCOM should be used for server controls. Current centralized management is managed in Mr. Foo and Mr. Hau’s groups where a skillset already exists to take on this type of system of control.
3. As recommended in the appendix a cloud governance (CCOE) needs to be set up for defining policies and operation controls as part of the adoption of the OpenStack platform.
4. User IT Self-Service is limited. Any requests for new VM’s and other services need to go through the proper channels with resources being assigned to assist with network, storage, VM’s, etc. It is recommended that {CUSTOMER\_NAME} look at providing an ITSM model where users are exposed to a catalog of services that they can request and those services can be deployed within minutes to hours. OpenStack is an enabler of IT self-service models. {CUSTOMER\_NAME} is also utilizing VMware for it’s underlying hypervisor, VMware SDDC services could also enable this type of model. Enabling user-self service will help to free up overburdened resources to focus more on IT operations and future enhancements. It is important to note that when enabling self-service for users that automation, zero trust, and security tooling are important to embed into the platform services.
5. Consider moving to an ITSM model to enable users self-service deployments as outlined in this report.
6. Move workloads to a Private Cloud for supportability, risk, cost, and management flexibility. Configure the access system so that each group has its own tenant with private networking and security groups. Leverage the CCoE to create the governance policies for logging, backups, metrics collection, and automation controls.
7. Create a digital lifecycle program (DLP) with policies based on the identification of data types. A structure of data classification needs to be created that identifies data categories from public through internal only. The Lifecycle and deletion policies of that data as well as data migration and exfiltration controls should be configured based on the program policies.
8. Cloud will require configuration management, automation, CI/CD, API, and other tools to help with service management and automation. {CUSTOMER\_NAME } should work with its vendors to explore tools such as terraform, tfs, git, Jenkins, OpenStack, Veeam, okta, ServiceNow, rubrik, and others for cloud type operation controls.
9. Plan app. portfolio transformation (Refactor/Rearchitect/Replace legacy applications ) initiatives for OpenStack readiness.
10. Plan a phased migration by order, criticality, and priority to limit risks of business interruption.
11. Talent availability for any cloud foundation platform is still considered a scarce commodity. It is recommended that {CUSTOMER\_NAME} work with providers that can provide on the job training for operating the platform while bringing their own best practices and operational guidance for {CUSTOMER\_NAME} to mimic.

## Local Area Networking (LAN) and Wide Area Network (WAN)

1. Continue with {PARTNER\_NAME } on the Cisco ACI implementation.
2. {CUSTOMER\_NAME} operates a large and complex metropolitan network with thousands of potential perimeter vulnerabilities due to the nature of the 3G devices spread across the electrical services within the area. It is recommended that {CUSTOMER\_NAME} implement a SASE solution to help with detection of anomalous intrusions into the {CUSTOMER\_NAME} network. Zscaler, Palo Alto Prisma, and Cisco Secure Access are commercially available providers that {CUSTOMER\_NAME} should consider.
3. Encryption policies for network traffic data in transit were not found. This could be solved via the implementation of the OpenStack platform.
4. {CUSTOMER\_NAME} uses several tools for network scanning; Cyberpark, Palo Alto SOAR, Citrix, F5, and others. The tool management is spread across different groups and departments. These groups need to be included when designing the Network schema for an OpenStack deployment.

## Backups, Security, Logging, Monitoring, and RISK Management

1. Review the recommendations for a Disaster Recovery Program in the Disaster Recovery Section.
   1. The General {CUSTOMER\_NAME } group operates an emergency response team and it is recommended that they provide more autonomy to {CUSTOMER\_NAME} in focusing on local specific threat incidents or take a more active role in focusing on local threats specific to the {CUSTOMER\_NAME} geography.
2. Expand log capture into a central repository for root-cause analysis of systems/apps
3. {CUSTOMER\_NAME} backup system is vulnerable to ransomware. Although the systems are backed up, the backups are to a storage subsystem that is replicated to another location here both systems and the control servers are on the same network. Should any of those systems be compromised it would require minimal effort for a bad actor to also lock those systems as part of the ransomware attack. It is recommended that {CUSTOMER\_NAME} implement an air gapped storage target that receives only backups at a pre-defined interval or that tape based or optical technologies be implemented to help provide another copied set of data not susceptible to the same vulnerabilities as the primary systems.
4. Policies for data in transit and at rest were not articulated or found during the workshops. An OpenStack implementation could help solve data in transit. Data at rest could be solved by the current pure storage subsystem if self-encrypting drives or in frame encryption exists however a vulnerability still exists where once a system is compromised the unencrypted data would be presented to the compromised system. It is recommended that {CUSTOMER\_NAME} look at tools such as Thales which provides data visibility based on the unique user profile and privileges. Additional tools to consider are Digital Guardian, and Guardium as retail packages or VeraCrypt for an opensource tool.
5. API controls must be updated. During discovery workshops it was highlighted that several API’s are open with no authentication controls. At a minimum token based authentication should be immediately implemented where possible. API gateways such as Apigee should be considered and additional tooling for security and user controls such as MuleSoft Anypoint, IBM API Connect, Kong Gateway, or Akamai Kona should also be considered.
6. Several older operating systems are deployed. Older operating systems are more vulnerable to exploitation and updates should be made.
7. Tools outside of {CUSTOMER\_NAME} security controls for auditing are leveraged. Viber, Zalo, and other informal communication tools are used for cross employee communication. This could open up {CUSTOMER\_NAME} to possible security breaches due to the types of information that might be shared. {CUSTOMER\_NAME} needs to adopt a formal communication tool with enterprise controls such as Slack or Microsoft Teams.
8. Advanced persistent threats, malware/ransom attacks, and other threats cannot be predicted. It is recommended that {CUSTOMER\_NAME} find a company where they can put agreements into place before an attack on how that vendor will help them when an event happens. The company will need to be able to help with immediate response, forensic analysis, triage and containment, communication, data recovery, system isolation, identification of the attack type, incident response, blast radius impact and scope, engagement with law enforcement, and post incident analysis. Find a copy to do all of this work after the event has occurred will take too long and agreements along with actionable controls need to be in place before the incident occurs.
9. Backups are to a storage platform and replicated within DC’s that are close to each other. This strategy needs to be evaluated after the scenario risk planning is completed. Additionally, for purposes of malware threats the data should be dumped on a daily or weekly cadence to tape copies or some other mechanism isolated from the disk subsystems. This will help provide a retrievable set of data should the data subsystems become compromised in a ransomware attack.
10. Update and the DR strategy, bring additional functions to local {CITY NAME } - {CUSTOMER\_NAME } , create a list of potential risk scenarios and update datacenter locations and operations according to the scenarios.
11. Profile application criticality and cost of application downtime per application to help create the appropriate RTO and RPO.
12. Find and implement a SASE provider such as CATO or Sentinel.
13. DR/Emergency response teams must execute tabletop exercises on a 6 month cadence.
14. Consider other tools such as crowdstrike for low level server process analysis, CyberArk for VM controls, and Octopus or Terraform for configuration management.
15. Focus on security designs that implement zero trust and least privilege to help reduce security incidents.
16. Implement policies “as code” in the new platform and use as much automation as possible
17. Implement a CI/CD deployment pipeline and integrate vulnerability scanning in the into the pipeline, including scanning of code as it is checked into the repo, compiled, and deployed.
18. Implement logging tools that capture as much telemetry as possible for anomaly analysis and detection. Telemetry capture will also become highly important as containerized solutions are adopted it will help provide developers with information that they require on application performance bottlenecks and poor coding deployments.

## Applications and Operating Systems

1. Modern configuration management tooling is not being leveraged; it is recommended that {CUSTOMER\_NAME} look at tooling outlined in the report. Examples are Octopus Deploy, Terraform, and ansible that should be considered.
2. The majority of applications are deployed manually, and updates are deployed through a process where users login to the systems and push DLLs. This method opens significant possibilities for user-error, and it violates least privileged access controls. The recommended solution is that automation be setup where the application updates are pushed into a code repository and then the updated files are pushed to the severs via automation tooling where the tooling has a service account that grants it access to specifically update the software directories.
3. Several applications were discovered to be built on web-based technologies with backend databases. These applications can be containerized or continue to run on VM’s. If containerized, the advantages will be that the applications become more portable and will fit into a developer pipeline that allows for a more rapid development of the application components as well as help fit into a blue-green deployment that will help to provide a higher level of application uptime. If the applications continue to reside on virtual machines, then an image-based build and deployment methodology should be adopted where each new update is built into an image, the image is deployed behind a load balancer and the old image sessions are drained from the load balancer pool following a blue-green deployment methodology. Either methodology will allow the application to better fit horizonal scaling requirements which can help with spikes in usage and allow the servers to run on less CPUs until additional instances are needed which will help free up capacity and potentially have an impact on improving cost savings.
4. There is an abundance of Windows 2016 and 2019 instances running in the environment. 2016 was EOL at the time of the interviews and 2019 reached EOL in January of 2024. It is recommended that {CUSTOMER\_NAME} migrate these instances to a newer windows platform or consider migrating to .net core on a Linux platform.
5. Ensure build automation for applications according to concepts written above.
6. Apply microservices design approach in building or re-factoring applications selectively to gain benefits.
7. Application development is done using TFS/Azure Devops, it is recommended that {CUSTOMER\_NAME} look at enabling co-pilot features to assist with code development and automated testing.
8. {Audit tools } reports showed that most machines were overprovisioned, when migrating to the OpenStack platform the provisioning of VM’s should not match what is currently in production. It is possible to match the VMware deployment sizes but not recommended as it would require tenants to have higher quotas which could quickly result in false reports on needed host hardware to be added.
9. The majority of the current IT processes are manually implemented and can be automated using existing tooling and services that exist in {CUSTOMER\_NAME} . Migrating to an OpenStack platform will provide API’s and other functionality that will simplify automation operations. For example, the update of most applications are performed by a user logging into the system and manually loading newer .dll’s. The .DLL’s could be checked into a repo such at github where the check-in triggers an automated job to copy the .dll’s into the servers. This would require service accounts to be created for server access but would help to improve the security profile of the
10. Implement an APM tool for capturing application flows and performance that assists with application tuning and troubleshooting.
11. servers by minimizing the needs for humans to login and it would help to tighten security controls as logs and activities can be focused on the single automation infrastructure.
12. Design micro-segmentation for additional security between apps. and databases.

## Databases, Artificial Intelligence, and Machine Learning

1. Several instances of the applications have Oracle based back end databases that are no longer supported by Oracle. Oracle 19c is being run for several other applications and it recommended that {CUSTOMER\_NAME} move the older databases to the more modern 19c implementation that {CUSTOMER\_NAME} is operating today.
2. Rearchitect & consolidate separating DB's from applications for HA and replication.
3. Data Life cycle policies were not found. It is recommended that {CUSTOMER\_NAME} evaluate standards such as the DGI lifecycle management framework or NIST 800-61 and adopt the standards that fit into its business requirements.
4. AI has the possibility to significantly improve {CUSTOMER\_NAME} ’s business operations. It is recommended that {CUSTOMER\_NAME} work with Rackspace or other ML providers to help with AI and data analytics. Rackspace has an office in District 9 with over 60 data specialists that could assist.
   1. Example #1: Several Large Language Models exist that would be capable of analyzing {CUSTOMER\_NAME } ’s field and technical manuals. These manuals should be fed into the LLM and trained into an application that would be made available to field technicians. The field technicians will then feed the observed issues into a prompt that will take the prompt and search for known solutions within the documentation. Rackspace has implemented several of these types of solutions for its customer base and seen field response and resolution times improve by as high as 200% due to the LLM providing an answer immediately across 10000+ pages of scanned documentation.
   2. Example #2: An online customer service chatbot can be implemented to help users with common questions and issues around services and billing.
      1. Use of the LLM and Chatbot should be notified to the end user with a disclaimer that the bot is there to help with guidance and all official responses are in the online documentation. The bot should provide reference links to its sources.
5. The OpenStack platform supports the ability to quickly deploy and scale bare metal servers. This is the ideal use case for Hadoop and other big data platforms that provide processing for large data sets. The implementation of a BareMetal Hadoop (BigData) solution would enable {CUSTOMER\_NAME} to do complex analytics of data collections. Some examples are as follows:
   1. Smart Grid Data Analysis: The rise of smart grids generates massive amounts of data from sensors monitoring energy usage, grid health, and consumer behavior. Hadoop can handle this data volume and help analyze it to identify patterns in energy consumption, predict outages, and optimize grid management.
   2. Customer Demand Forecasting: {CUSTOMER\_NAME} can leverage Hadoop to analyze historical customer data, weather patterns, and other relevant factors. This allows for more accurate demand forecasting, which helps optimize power generation and distribution to meet fluctuating needs and avoid disruptions.
   3. Predictive Maintenance: Hadoop can be used to analyze data from sensors placed on power distribution equipment. By identifying trends and anomalies, it can predict potential equipment failures and enable proactive maintenance, preventing outages and improving grid reliability.
   4. Renewable Energy Integration: The integration of renewable energy sources like wind and solar power introduces variability into the grid. Hadoop can analyze data from renewable sources and conventional power plants, helping optimize energy dispatch and maintain grid stability.
   5. Fraud Detection and Security: Hadoop can be used to analyze large volumes of data related to power usage patterns. This can help identify potential anomalies and suspicious activity, enabling the detection of electricity theft or other security issues within the power distribution network.

# Future of Success – Next steps and performance optimization

Changing operational behaviors and organizational policies is a large undertaking. As {CUSTOMER\_NAME} starts on its digital transformation journey there is a plethora of activities that need to be completed to prepare for the journey, execution of the migration to a new platform, and operational governance upon completion of the transformation. Digital transformation should follow a very specific outline. The high level roadmap for the path of a successful digital transformation and migration should follow this formula:

1. Pre- Engagement - scoping of services and applications for migration are defined, a migration readiness checklist is created, and validation of funding and resource allocation is confirmed.
2. Assess – Requirements and current state are reviewed, user stories are collected, acceptance criteria is defined, and a baseline of the current environment is performed. Applications in scope should be assessed for migration fit and validated by the business owners as migration ready. Typically this is done via the use of discovery software that profiles entire environments and maps application dependencies as well as usage patterns.
3. Design – the details of the future state architecture are created based on the requirements gathered in the assess phase and the CCoE adds in requirements for operational controls and other requirements for security and automation.
4. Planning – Migration run books are created, test scenarios are created, migration planning on application migration waves, and migration schedules and timelines are created. Due to the uniqueness of each application, specific plans at an individual application level must be created to help ensure success for migration to the new platform. Migration software with automated failover should be used as much as possible.
5. Build and Implement – Service delivery teams are created along with an implementation team that will create operations run books, configure service management integrations, and then build the target platform and test for acceptance based on CCoE defined criteria.
6. Migrate – Migration of the applications is executed. Users test application performance and functionality and then accept or reject the migration.
7. Training and transformation – Monitoring, security, operations, and other groups confirm that their components are implemented and working within the new environment.

The summary of the next steps for {CUSTOMER\_NAME} are outlined as follows.

1. Perform an in-depth application analysis of application interdependencies using tooling that will help define the requirements in the Assess phase of the digital transformation roadmap outlined above.
2. Establish a CCoE. The CCoE is outside of the migration roadmap but will play a critical role in defining Archiecture, Operational, and Security Governance controls. A good CCoE will start with business alignment workshops that lead into an operational minimum viable program. As the CCoE progresses the team, charter, vision, and operating model will develop. The CCoE will develop governance artifacts, create playbooks, and implementation patterns. The CCoE will help to define the following:
   1. DevSecOps for product development and operations
   2. Infrastructure as code tooling and operations adoption
   3. Application modernization pilot controls for first application Replatform activities
   4. Application modernization for remaining applications.
3. Identify a cloud provider to implement and manage the new platform for OpenStack. The provider should have experience with low level design, follows a public source deployment, and offer services that allow for customization of the platform to {CUSTOMER\_NAME} ’s unique specific needs. Rackspace and {PARTNER\_NAME } together provide capabilities that will help to do the design as well as the application migration and application refactoring.
4. AI and Machine Learning – as mentioned in the recommendations section of this report, LLM and ML models can be built that will help operations, service technicians, and others with the ability to ask questions on complex issues and needs where the LLM and ML can provide direct answers and reference to documentation that describes the solutions and answers to the posed questions. Additional analytics can be built on the OpenStack. This should start with ideation sessions where user requirements and resources are identified to help determine the best method for building and training a model specific to {CUSTOMER\_NAME} ’s needs. Once the standard models are created, the next step will be to setup automated model learning and deployment, setup model predictions for tracking model versions and hyperparameters to help with model refining, and then model governance for explaining parameters that were followed to auditors.
5. Data Analytics – Analytics plays a vital role in modernizing the power generation and distribution industry. Proper analytics can help {CUSTOMER\_NAME} in the following ways:
   1. Grid Optimization - Real-time data analysis: By analyzing data from smart meters, sensors, and SCADA systems, utilities can optimize grid operations in real-time. This includes Identifying areas of congestion and potential outages as well as balancing supply and demand by dynamically adjusting power generation and distribution.
   2. Demand response and load management - Understanding customer energy consumption patterns allows utilities to offer dynamic pricing plans that incentivize off-peak energy usage. Implementing a targeted demand response programs based on real customer usage data allows {CUSTOMER\_NAME} to offer motivational incentives where customers temporarily reduce consumption during peak hours which will help to improve grid stability by managing overall electricity demand.
   3. Customer Behavior Analysis and Asset Management - Analytics can predict equipment failures by analyzing sensor data and historical maintenance records. This will enable {CUSTOMER\_NAME} to perform proactive maintenance on equipment before it fails, which will help to prevent unexpected outages and prevent costly repairs.
   4. Energy theft detection - Advanced analytics can identify unusual energy consumption patterns that might indicate meter tampering or electricity theft. This will allow {CUSTOMER\_NAME} to detect and address theft incidents quickly, reducing revenue loss.
6. Other activities outlined in this report such as establishing a disaster recovery program, implementing security governance and security controls, and defining and implementing a DLP program should be performed in conjunction with the digital transformation initiatives.

# Appendix A – Medium Level Design

## Introduction

As part of the consulting engagement between Rackspace and {CUSTOMER\_NAME} a request was made for the recommendation of an infrastructure platform that will enable {CUSTOMER\_NAME} to modernize its operations into a more cloud like operations set. Although only a recommendation is required and a detailed design is out of scope for the statement of work, the following information is being provided as a mid tier design to assist {CUSTOMER\_NAME} in understanding what Rackspace’s recommendation of an OpenStack deployment will look like.

This document describes the architecture, components, and configurations required to meet the needs of {CUSTOMER\_NAME} . Rackspace understands the need for {CUSTOMER\_NAME} to be able to continue to offer cloud resources to various internal groups. The goal of this deployment is to create an OpenStack environment where {CUSTOMER\_NAME} has the tools to be successful in delivering the following value to its internal stakeholders:

* Infrastructure as a Service
* Strong security & operational controls over the environment
* On-going support and management
* Secure platform and networking with governance of data
* Scalable infrastructure with the ability to grow over time

Many specifics on details of configuration and automation being run are already documented in the OpenStack Ansible repo and the Rackspace private cloud repo. This section will be brief to describe components for networking and hardware specifics to {CUSTOMER\_NAME} for deploying a development environment. If {CUSTOMER\_NAME} follows the guidance in this section, then the environment will be prepared to sufficiently run the OpenStack-Ansible deployment scripts. The larger documentation repository should be consulted for the specifics of services being configured and enabled.

* The documentation for OpenStack ansible is located here: [OpenStack Ansible Latest](https://docs.openstack.org/project-deploy-guide/openstack-ansible/latest/)

## Background

Rackspace performed a broad assessment of {CUSTOMER\_NAME} ’s architecture to determine needs for continued ongoing operations. This is to remain compliant with General {CUSTOMER\_NAME } standards on minimizing downtime and security controls. This section has been created to help provide a design for a development environment. It is important to note that due to the complexities of upgrading OpenStack services it is always possible that during upgrades of an environment that live troubleshooting be a requirement due to things such as missing packages. For this reason, an upgrade maintenance should always be scheduled when appropriate personnel with expertise in OVS, OpenStack engineering, and Linux operations be available during the upgrade process.

## Assumptions

The following assumptions are made within this section:

* {CUSTOMER\_NAME} will provide hardware that meets the minimums outlined in this document.
* {CUSTOMER\_NAME} will take advantage of recommendations on creating an IaaS cluster. If not, then the sections that outline network for those services can be ignored.
* {CUSTOMER\_NAME} will consult with Rackspace prior to execution on a design for validation
* The current PureStorage subsystem will continue be maintained as an internal {CUSTOMER\_NAME} tooling for storage and will be upgraded to a maintained driver that is compatible to Cinder.
* CEPH is under consideration for Object Storage requirements.
* The same OU and defined domains used in production for LDAP authentication can be used for the development group and that configuration will be duplicated into the development environment.

## Architecture

The recommended layout for installing OpenStack would be as follows:

* Two infrastructure hosts for running HAProxy and Keepalived load balancer services.
* Three control plane hosts for running the OpenStack API services in each data center.
* 8 compute hosts for compute services in each data center.

Additional hosts like five Ceph storage hosts or file based shared storage systems like NetApp is required to offload the consumer and guest data from the compute node.

All hosts require at least two 10 Gbps network interfaces and one 1Gbps network interface. Storage hosts like Ceph require additional two 10 Gbps network interfaces to separate the storage replication traffic from the compute node traffic.

To use the optional iSCSI Block Storage (cinder) service, additional hosts are required. Cinder via iSCSI is

no longer part of the reference architecture as Ceph or PureStorage is required as block storage backend for

Cinder. Cinder iSCSI block storage hosts require an LVM volume group named cinder-volumes.

The hosts are called target hosts because Ansible deploys the Rackspace OpenStack (RPC-O) environment within these hosts.

The RPCO environment also requires a deployment host from which Ansible orchestrates the deployment process. Usually, the first controller host also serves as the deployment host.

The controller target hosts contain the following services:

* OpenStack foundation:
  + Maria DB with Galera
  + RabbitMQ
  + Memcached
* OpenStack:
  + Identity (keystone)
  + Image service (glance)
  + Compute management (nova)
  + Networking (neutron)
  + Orchestration (heat)
  + Dashboard (horizon)
  + Load Balancing (octavia)
  + Bare metal as service (ironic)
  + DNS as service (designate)

The infrastructure target host contains the following services:

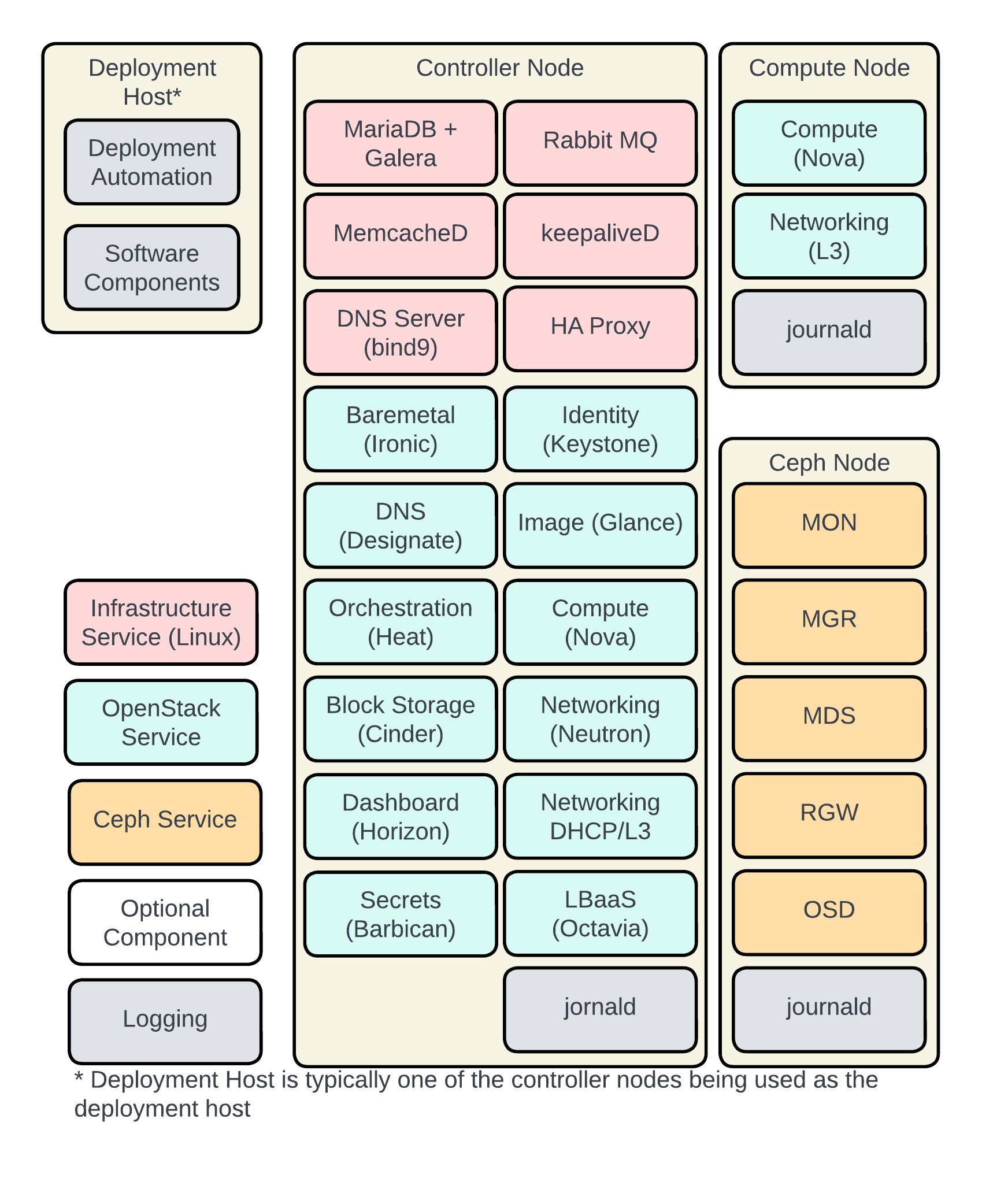
* HAProxy load balancer
* Keepalived IP failover

Discussion on the monitoring environment at {CUSTOMER\_NAME} was not had, the services can be integrated into {CUSTOMER\_NAME} current tooling, monitored by Rackspace as part of the management service, or should {CUSTOMER\_NAME} wish to setup a separate monitoring service them optional services can be provided with a separate log\_hosts target:

* Logstash
* Elasticsearch with Kibana

The Compute target host contains the following services:

* Compute virtualization
* Networking agents



The following graphic is an example of the interface links that will be deployed.

A diagram of a company

Description automatically generated

## Software Versions

To adhere to a version equal to the current production environment, it is recommended that {CUSTOMER\_NAME} deploy the Rackspace RPC-O deployment onto servers that are certified by Ubuntu's Hardware Certified List for the supported Ubuntu Linux LTS release to be deployed. For {CUSTOMER\_NAME} , the Ubuntu 22.04 release will be used as the operating system to match what is already deployed in production.

The Rackspace nomenclature of releases follows the OpenStack and OpenStack Ansible release guides. This indicated that Rackspace RPC-O versions correspond directly to OpenStack releases. Once an OpenStack release is made available upstream, Rackspace begins the testing process to validate it. Some of the validation tests include:

* Testing against standard Rackspace RPC server hardware and the supported Ubuntu LTS operating system version(s)
* Testing the OpenStack-Ansible release process with the standard set of OpenStack services that are a part of RPC-O
* Testing the upgrade process from the prior RPC-O version to the current version
* Testing against the storage backend

OpenStack releases occur approximately twice per year. However, not every OpenStack release is turned into a supported RPC-O release. The version of OpenStack being installed for {CUSTOMER\_NAME} to start should at a minimum be Antelope.

## Supported Hardware

Rackspace’s recommendation is that {CUSTOMER\_NAME} deploy on hardware certified by Ubuntu’s Hardware Compatibility list ([Ubuntu HCL](https://ubuntu.com/certified/servers)) for the current or n-1 supported Ubuntu Linux LTS release.

## Networking

Several Networks are required to deploy OpenStack. Each server host used must have a minimum of (2) interfaces and be at a minimum of 10 GbE.

The current network outline below leverages VLAN ID’s identified by Rackspace PS, specific VLAN design will need to be worked out with {CUSTOMER\_NAME} and its network provider {PARTNER\_NAME } . The vlans are all non-routed.

## Logical Network Configuration

The following table includes the listing of networks that need to be accessed by servers in the environment.

IMPORTANT: The IP addresses in the following table are placeholder examples for reference only of IP allocation. IP space is managed by {CUSTOMER\_NAME} ’s network group and the {CUSTOMER\_NAME} operations group will need to work with {CUSTOMER\_NAME} ’s network management group to update IP addresses. The host names are also recommendations and can be changed.

{NET\_ALLOCATION\_TABLE }

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RPC Network Allocation | | | | |
| VLAN Name | VLAN ID | CIDR | Gateway | Addr. Assignment |
| Public VIP | 201 | 10.50.85.0/27 | 10.50.85.1 | Static |
| Host | 202 | 10.50.85.64/26 | 10.0.85.65 | Static |
| Ironic Provisioning | 203 | 172.16.0.0/20 | non-routed | Ansible, Neutron |
| IPMI | 204 | 172.16.16.0/21 | non-routed | Ansible, DHCP, Static |
| CP A – CEPH Storage | 205 | 172.16.24.0/24 | non-routed | Ansible, Static |
| CP A – Container | 206 | 172.16.25.0/24 | non-routed | Ansible, Static |
| CP A – CEPH Replication | 207 | 172.16.26.0/24 | non-routed | Ansible, Static |
| CP B – CEPH Storage | 208 | 172.16.27.0/24 | non-routed | Ansible, Static |
| CP B - Container | 209 | 172.16.28.0/24 | non-routed | Ansible, Static |
| CP B – CEPH Replication | 210 | 172.16.29.0/24 | non-routed | Ansible, Static |
| CP C – CEPH Storage | 211 | 172.16.30.0/24 | non-routed | Ansible, Static |
| CP C - Container | 212 | 172.16.31.0/24 | non-routed | Ansible, Static |
| CP C – CEPH Replication | 213 | 172.16.32.0/24 | non-routed | Ansible, Static |

## Physical IP ranges

IMPORTANT: The IP addresses in the following table are placeholder examples for reference only of IP allocation. IP space is managed by {CUSTOMER\_NAME} ’s network group and the {CUSTOMER\_NAME} Pod will need to work with {CUSTOMER\_NAME} ’s network management group to update IP addresses. The host names are also recommendations and can be changed.

|  |  |  |
| --- | --- | --- |
| Host Name | IP Address | Allocation |
| ironic.na.devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.4/32 | public VIP |
| haproxy01.dc1.deveopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.67 | HAproxy Hypervisor |
| haproxy02.dc1.deveopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.68 | HAproxy Hypervisor |
| haproxy-01-a.dc1. devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.70/26 | haproxy 1 (Virtual) |
| haproxy-02-a.dc1. devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.73/26 | haproxy 2 (Virtual) |
| infra01-a.dc1.devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.82/26 | control node 1 |
| infra02-a.dc1.devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.83/26 | control node 2 |
| infra02-a.dc1.devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.84/26 | control node 3 |

Note: -a was appended to the HAproxy and control nodes under the assumption that {CUSTOMER\_NAME} will deploy more than one development type control plane. -a can be upgraded to -b, -c, etc. to match the current control planes nomenclature (i.e. Norwest Ussuri is currently cluster -b) in production or use it as a means to identify additional control plane services.

Note #2: for purposes of networking the 10.50.x.x space is being carved up as an example of how the network space should be utilized, {CUSTOMER\_NAME} ’s network group may define the space however they would like, this is meant to be a reference guide only.

IP Address allocation

Each node consumes a different amount of IP addresses from various networks. The exact network connections and IP address requirements depend on the RPC node type. Guidance for IP consumption is as follows:

Controller Node

* In the HOST network
* (21) in the Container network
* in the Storage network (currently not used by {CUSTOMER\_NAME} today)
* in the Tunnel Network

Compute Nodes

* In the HOST network
* in the Container network
* in the Storage network (currently not used by {CUSTOMER\_NAME} today)
* in the Tunnel Network

Ironic Nodes

* In the HOST network
* in the Storage network (currently not used by {CUSTOMER\_NAME} today)
* in the Tunnel Network

## {Additional\_Network\_Options}

Example

The RPC control planes are deployed leveraging Ansible automation, which requires several baseline inputs or initial configuration values such as IP ranges and API VIPs. The HAproxy and utility nodes will primarily be provisioned and configured manually, with some components being configured via control plane automation. {CUSTOMER\_NAME} uses a utility node for integration of specific services in their CI/CD pipeline, the initial network values for the utility node are captured in this table along with additional network configurations on the HA Proxy nodes that will be required for network routing

|  |  |
| --- | --- |
| Initial Configuration – Utility Node | |
| Attribute | Value |
| util.nor.prod. {CUSTOMER\_NAME} host IP (primary) | 10.50.85.75/26 |
| Util.nor.prod. {CUSTOMER\_NAME} IPMI IP | 172.16.16.5/21 |
| cobbler.nor.prod. {CUSTOMER\_NAME} host IP (Cobbler LXC) | 10.84.71.77/26 |
| jenkins.nor.prod. {CUSTOMER\_NAME} host IP (Jenkins LXC) | 10.84.71.76/26 |

Note: The cobbler and Jenkins nodes are not currently deployed. . If a different set of servers exist for deployment pipelines then the IP values above will need to be updated to those values.

|  |  |
| --- | --- |
| Initial Configuration – HAproxy Nodes | |
| Attribute | Value |
| haproxy01.dc1.devopesnstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.67/26 |
| haproxy01 OOB static IP | 172.16.16.3/21 |
| haproxy01 IPMI DHCP server IP | 172.16.16.6/21 |
| haproxy01- dc1.devopenstack.app.in. {CUSTOMER\_NAME} VM | 10.50.85.69/26 |
| haproxy01-b.dc1.devopenstack.app.in. {CUSTOMER\_NAME} VM | 10.50.85.70/26 |
| haproxy01-b.dc1.devopenstack.app.in. {CUSTOMER\_NAME} VM | 10.50.85.71/26 |
| haproxy01 CEPH storage IP – CP A | 172.16.24.3/24 |
| haproxy01 CEPH storage IP – CP B | 172.16.27.3/24 |
| haproxy01 CEPH storage IP – CP C | 172.16.30.3/24 |
| haproxy01 Container IP – CP A | 172.16.24.3/24 |
| haproxy01 Container IP – CP B | 172.16.28.3/24 |
| haproxy01 Container IP – CP C | 172.16.31.3/24 |
| dhcp01.dc1.devopenstack.app.in. {CUSTOMER\_NAME} | 10.50.85.90/26 |
|  |  |
| haproxy02. dc1.devopesnstack.app.in. {CUSTOMER\_NAME} | 10.50.85.68/26 |
| haproxy02 OOB static IP | 172.16.16.4/21 |
| haproxy02 IPMI DHCP server IP | 172.16.16.7/21 |
| haproxy02-dc1.devopenstack.app.in. {CUSTOMER\_NAME} VM | 10.50.85.72/26 |
| haproxy02-dc1.devopenstack.app.in. {CUSTOMER\_NAME} VM | 10.50.85.73/26 |
| haproxy02-dc1.devopenstack.app.in. {CUSTOMER\_NAME} VM | 10.50.85.74/26 |
| haproxy02 CEPH storage IP – CP A | 172.16.24.4/24 |
| haproxy02 CEPH storage IP – CP B | 172.16.27.4/24 |
| haproxy02 CEPH storage IP – CP C | 172.16.30.4/24 |
| haproxy02 Container IP – CP A | 172.16.24.4/24 |
| haproxy02 Container IP – CP B | 172.16.28.4/24 |
| haproxy02 Container IP – CP C | 172.16.31.4/24 |
| dhcp02.nor.prod. {CUSTOMER\_NAME} | 10.50.85.91/26 |
|  |  |
| IPMI DHCP allocation range | 172.16.16.50/21 – 172.16.23.254/21 |
|  |  |
| Public VIP FQDN – CP A | cloud1a.dc1.devopenstack.app.in. {CUSTOMER\_NAME} |
| Public VIP – CP A | 10.50.85.3/27 |
| FQDN need to be validated and defined by {CUSTOMER\_NAME} |  |
| Public VIP FQDN – CP B | cloud1b.dc1.devopenstack.app.in. {CUSTOMER\_NAME} |
| Public VIP – CP B | 10.50.85.4/27 |
| FQDN need to be validated and defined by {CUSTOMER\_NAME} |  |
| Public VIP FQDN – CP C | CEPH.dc1.devopenstack.app.in. {CUSTOMER\_NAME} |
| Public VIP – CP C | 10.50.85.5/27 |
| FQDN need to be validated and defined by {CUSTOMER\_NAME} | FQDN for Turon, Terry, and Chery CEPH deployments and IP address should exist for current CEPH Deployments and should be substituted in the lines above depending on the environment to be used |

The following is a list of IP ranges for the (3) control plane services nodes

|  |  |
| --- | --- |
| Initial Configuration – Control Plane A | |
| Attribute | Value |
| infra01-dc1.devopenstack.app.in. {CUSTOMER\_NAME} HOST IP | 10.50.85.78/26 |
| infra02-dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.79/26 |
| infra03-dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.80/26 |
| Provisioning range for physical hosts | 172.16.0.5/20 – 172.16.0.19/20 |
| Provisioning range for containers | 172.16.0.20/20 – 172.16.0.49/20 |
| Provisioning range for Ironic (via DHCP) | 172.16.0.50/20 – 172.16.7.255/20 |
| Container range for physical hosts | 172.16.25.5/24 – 172.16.25.19/24 |
| Container range for containers | 172.16.25.20/24 – 172.16.25.254/24 |
| CEPH storage range for physical hosts | 172.16.24.5/24 – 172.16.24.19/24 |
| CEPH storage range for containers | 172.16.24.20/24 – 172.16.24.254/24 |
| CEPH replication range for physical hosts | 172.16.26.5/24 – 172.16.26.254/24 |

|  |  |
| --- | --- |
| Initial Configuration – Control Plane B | |
| Attribute | Value |
| infra01-b.dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.82/26 |
| infra02-b.dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.83/26 |
| infra03-b.dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.84/26 |
| Provisioning range for physical hosts | 172.16.8.5/20 – 172.16.8.19/20 |
| Provisioning range for containers | 172.16.8.20/20 – 172.16.8.49/20 |
| Provisioning range for Ironic (via DHCP) | 172.16.8.50/20 – 172.16.15.254/20 |
| Container range for physical hosts | 172.16.28.5/24 – 172.16.28.19/24 |
| Container range for containers | 172.16.28.20/24 – 172.16.28.254/24 |
| CEPH storage range for physical hosts | 172.16.27.5/24 – 172.16.27.19/24 |
| CEPH storage range for containers | 172.16.27.20/24 – 172.16.27.254/24 |
| CEPH replication range for physical hosts | 172.16.29.5/24 – 172.16.29.254/24 |

|  |  |
| --- | --- |
| Initial Configuration – Control Plane C | |
| Attribute | Value |
| infra01-c.dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.86/26 |
| infra02-c.dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.87/26 |
| infra03-c.dc1.devopenstack.app.in. {CUSTOMER\_NAME} host IP | 10.50.85.88/26 |
| Container range for physical hosts | 172.16.31.5/24 – 172.16.31.19/24 |
| Container range for containers | 172.16.31.20/24 – 172.16.31.254/24 |
| CEPH storage range for physical hosts | 172.16.30.5/24 – 172.16.30.128/24 |
| CEPH storage range for containers | 172.16.30.129/24 – 172.16.30.254/24 |
| CEPH replication range for physical hosts | 172.16.32.5/24 – 172.16.32.254/24 |

## Ironic Node Flavor Configuration

The following is an example of who the ironic node flavor configuration should look. The Memory and disk size were pulled from a current {CUSTOMER\_NAME} inventory report and should be changed to match and ironic nodes deployed in the development environment as it is assumed that smaller node configurations will be used for development.

|  |  |  |
| --- | --- | --- |
| Nova Flavors | | |
|  | Value | Description | |
| Name | Ironic\_standard | A descriptive name. | |
| Memory\_MB | 256000 | Ironic machine memory in megabytes. | |
| Disk | 600 | Root disk size in gigabytes. | |
| vCPUs | 20 | Number of CPUs presented to the instance. | |
| Is\_Public | true | Boolean value, whether flavor is available to all users or private to the tenant it was created in. Defaults to True. | |
| Flavor\_key | cpu\_arch=x86\_64 | Key and value pairs that define on which compute nodes a flavor can run. These pairs must match corresponding pairs on the ironic nodes. | |

## {SSL Certificates}

Traffic between services and client->server is secured using SSL/TLS 1.2 technologies. The use of a valid (signed) SSL certificate is important to avoid warnings from various API clients and to avoid deeper issues related to unsigned certificates. Often these issues can be resolved by using a private certificate authority (CA), but the distribution of related intermediate and root certificates is problematic. Using a reputable public certificate authority is recommended.

Because various services within OpenStack require the use of certificates and may have independent endpoints, a wildcard certificate is highly preferred in the following format (preferred):

SSL: \*.dc1.devopenstack.app.in. {CUSTOMER\_NAME}

FQDN: portal.dc1.devopenstack.app.in. {CUSTOMER\_NAME} (NOTE Actual FQDN needs to be provided, this is an estimated ID)

These are examples only and can be set to whatever {CUSTOMER\_NAME} prefers.

## OpenStack components

OpenStack consists of a set of interdependent services that work together to provide a cloud. Rackspace supports a subset of the OpenStack services included in each release. This is based on priority, scope, enterprise readiness for our customers, and ability to support.

The following OpenStack projects recommended for a development deployment:

|  |  |  |
| --- | --- | --- |
| Service | Description | Notes |
| Keystone | Identity | Keystone will be synchronized with LDAP. Users are mapped back to projects and groups |
| Nova | Compute | Nova services are not required for deployment however they are recommended for future use by {CUSTOMER\_NAME} and are included here. A 16:1 CPU and 15:1 overcommit is recommended |
| Neutron | Networking | {CUSTOMER\_NAME} currently uses VLAN for most networking and L3 VXLAN should be implemented and used for tenant segregation once the Nova Services become a standard offering |
| Glance | Image Storage | Currently an optional service being proposed to support the Nova Service |
| Horizon | Web Interface | Web interface for console view of the OpenStack environment for admins and also for each tenant that would get created with the Nova services. |
| Designate | DNS as a service | Currently an optional service being proposed to support the Nova Service |
| Ocatvia | Load Balancing | Currently an optional service being proposed to support the Nova Service |
| Manila | NFS File Sharing | Currently an optional service being proposed to support the Nova Service. This service will need to be paired with a NetApp or other compatible storage device. |
| Cinder | Block Storage | Currently an optional service being proposed to support the Nova Service. Ceph is the recommended block storage component however Cinder should be considered if {CUSTOMER\_NAME} has a compatible block storage subsystem compatible to the cinder API. |

## {Custom\_OpenStack\_Components}

## {Project/User Configuration and Naming }

The Rackspace Private Cloud environment requires an administrative Tenant/Project name and corresponding user. Recommended names are as follows to match current production standards:

|  |  |  |
| --- | --- | --- |
| Keystone Project/User Configuration | | |
| Project Name: | Usernames | Description |
| admin | acoe\_ {CUSTOMER\_NAME} \_admin\_cpX | Rackspace administrative user |
| admin | admin | Ephyra service user |
| drone | drone | {CUSTOMER\_NAME} drone user |
| service | acoe\_ {CUSTOMER\_NAME} \_glance\_cpX | Glance service user |
| service | acoe\_ {CUSTOMER\_NAME} \_swift\_cpX | Swift service user |
| service | acoe\_ {CUSTOMER\_NAME} \_nova\_cpX | Nova service user |
| service | acoe\_ {CUSTOMER\_NAME} \_heat\_cpX | Heat service user |
| service | acoe\_ {CUSTOMER\_NAME} \_ks\_cpX | Keystone service user |
| service | acoe\_ {CUSTOMER\_NAME} \_neutron\_cpX | Neutron service user |
| service | acoe\_ {CUSTOMER\_NAME} \_ironic\_cpX | Ironic service user |
| service | acoe\_ {CUSTOMER\_NAME} \_dsp\_cpX | Swift dispersion report user |
| service | acoe\_ {CUSTOMER\_NAME} \_dom\_adm\_cpX | Heat stack domain service user |

## {Deployment Configurations}

### Controller Node and HA Proxy Nodes

Server profile: HP DL380 G9 or newer, Dell 7XX series or newer. The hardware should be found on the Ubuntu HCL as mentioned previously.

* Operating System: Ubuntu 22.04 LTS or higher
* Processsors: Dual Intel Silver 4514 processors or higher
* Memory: 512GB RAM
* Storage: (2) 480GB – 6GB STA Read Intensive SSD in Raid 1 and (8) 480GB SATA Multi Use drives SSD
* Networking: (2) 10GbE for Bond 1 and (2) 10GbE for Bond 0 (1) GbE or higher nic for IPMI for ironic out of band support

There should be (3) Controller nodes and (2) HA Proxy Nodes. HA Proxy nodes do not require the additional (8) 480GB SATA drives.

The following partition layout should be used for the controller Nodes:

|  |  |  |  |
| --- | --- | --- | --- |
| Mount Point | Size | Filesystem | LVM VG |
| /boot | 1 GB | Ext4/XFS | - |
| /boot/EFI | 256 MB | FAT32 | - |
| /home | 10 GB | Ext4/XFS | vglocal00 |
| /openstack | 100 GB | Ext4/XFS | vglocal01 |
| /opt | 10 GB | Ext4/XFS | vglocal00 |
| /root | 20 GB | Ext4/XFS | vglocal00 |
| /tmp | 10 GB | Ext4/XFS | vglocal00 |
| /var | 50 GB | Ext4/XFS | vglocal00 |
| /var/log | 50 GB | Ext4/XFS | vglocal00 |

HA Proxy setup should follow a standard HA Proxy and can be installed on the raid 1 array.

### Compute Nodes (Nova)

Server profile: HP DL380 G9 or newer, Dell 7XX series or newer. The hardware should be found on the Ubuntu HCL as mentioned previously. Current estimates are (8) nodes per Datacenter

* Operating System: Ubuntu 22.04 LTS or higher
* Processors: Dual Gold 6430 processors or higher (32 core per CPU ore more)
* Memory: 1024 GB Minimum
* Storage: (2) 900GB – 6GB STA Read Intensive SSD in Raid 1
* Networking: (2) 10GbE for Bond 1 and (2) 10GbE for Bond 0

The following partition layout should be used for the controller Nodes:

|  |  |  |  |
| --- | --- | --- | --- |
| Mount Point | Size | Filesystem | LVM VG |
| /boot | 1 GB | Ext4/XFS | - |
| /boot/EFI | 256 MB | FAT32 | - |
| /home | 10 GB | Ext4/XFS | vglocal00 |
| /openstack | 100 GB | Ext4/XFS | vglocal01 |
| /opt | 10 GB | Ext4/XFS | vglocal00 |
| /root | 20 GB | Ext4/XFS | vglocal00 |
| /tmp | 10 GB | Ext4/XFS | vglocal00 |
| /var | 50 GB | Ext4/XFS | vglocal00 |
| /var/log | 50 GB | Ext4/XFS | vglocal00 |
| /var/lib/nova | 100GB + | Ext4/XFS | vglocal01 |

# Appendix B –{ Supplemental materials for Operations controls}

Operating an internal cloud platform requires that the operators manage the environment as if it was a commercial offering available to the internal users and business units of the organization. The operational objective should be to setup the platform in a way that users can consume it as a self-service platform managed via GUI, CLI, and API functions. Due to this being out of scope of the statement of work examples of how this can be achieved along with a platform management framework are provided below. A more exhaustive and comprehensive list can be developed via further engagements between {CUSTOMER\_NAME} with Rackspace and {PARTNER\_NAME } .

The first framework to consider is the Rackspace Nine Block framework and the second is the OneCloud operational teams framework. The Rackspace nine-block methodology is high-level and provides good strategic guidance for operating the platform as a product to be consumed by those within {CUSTOMER\_NAME} while the OneCloud framework/methodology is more tactical in nature and should be adopted as the methodology for day-to-day operations of the OpenStack environment.

The “Nine Block” Requirement Gathering Framework is a core component of the Rackspace solution design methodology. The Functional Requirements documented below are the end-result and output of this methodology.

|  |  |  |
| --- | --- | --- |
| **Service Quality Management** | **Capacity Management** | **Standard Operating Environment** |
| Service Quality Management is the appropriate level of performance, availability, and functionality of a solution that meets the customer's business and technical needs of an organization and maintains agreed upon service levels. | Capacity Management is the processes, policies, and technologies, which create and maintain suitable resource and service capacity to meet the demands of the customer's business in a cost-effective manner. | The Standard Operating Environment is the standard architectures, applications, operating environments, test frameworks, methodologies, procedures, and terminologies used to enhance consistency and manageability within an organization. |
| **Availability Management** | **Data Management** | **System Management** |
| Availability Management is the set of the processes, policies, and technologies that determine the availability of IT services as they relate to the business needs of the customer. | Data Management is the processes, policies, and technologies that control, protect, deliver, and enhance the value of data and information assets critical to the business needs of the customer. | System Management is the people, processes, policies, and technologies used to manage and maintain the business IT environment. |
| **Continuity Management** | **Information Security Management** | **Operational Readiness** |
| Continuity Management is the set of processes, policies, and technologies used to enable recovery or continuation of customer business processes after situations that impact business-critical IT services. | Information Security Management is the control of access to the customer's information and information systems and the protection of these systems from unauthorized access, use, and disruption. | Operational Readiness is a measure of an organization's existing knowledge and processes as they relate to platform and application knowledge, support, and escalation processes, and over all familiarity with the IT service. |

The OneCloud framework is designed to be a service experience for how users and consumers of the cloud configure their workloads to operate in the cloud environment. The goal of the framework is to limit risk and maintain cost controls while maximizing enablement via the use of tooling that provides accountability, visibility, and automation standardization. The proposed framework is centralized around a curator design where the “Curator” has ownership of the items shown in the following graphic:

A diagram of a company

Description automatically generated

The overarching structure of the design calls for a Cloud Governance Council that takes feedback from Architecture, Security, Engineering, Risk, Operations, DPG, Developers, and other groups to create policies, processes, and workload types that adhere to General- {CUSTOMER\_NAME } and {CUSTOMER\_NAME} governing operational standards and requirements. The polices are then incorporated into the {CUSTOMER\_NAME} cloud offerings in a way that they become automated and provided to the end consumer in a way that the end consumer can use the OpenStack services from a service catalog. For example, the OpenStack environment can be set up to offer Nova and Glance services which allows consumers to deploy pre-approved hardened images for use by the various {CUSTOMER\_NAME} business units. Moreover, this can be expanded upon where consumers can leverage tools such as GitHub for code repositories where a GitHub action could be used to trigger the deployment of a server image in the OpenStack platform, metadata within the image or metadata passed in via the create server call could specify other activities to take place such as the running of ansible playbooks to configure security policy, and application deployment. Furthermore, terraform and other configuration management tooling can be integrated into the platform for BU’s that wish to use those tools for configuration management and deployment. Finally, should {CUSTOMER\_NAME} wish to set policies on consumption limits and have operational controls regarding BU consumption and controls, each BU and sub-BU groups can be created as tenants in the OpenStack environment with defined limits on the amounts of CPU, RAM, Disk, and image types that are utilized.

By following this standard for providing automation integration of the platform, the {CUSTOMER\_NAME} operations team will be able to provide a method of operating that is already familiar to various {CUSTOMER\_NAME} business units. It is anticipated that the General {CUSTOMER\_NAME } group would provide guidance on the policies for operational controls and security standards. Examples of standards and operating requirements of the environment and how they should be grouped are as follows:

### Service Quality Management and Systems Management

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR1.0 | Monitoring of the underlying Services |
| FR1.0.1 | Implement comprehensive monitoring tools that can monitor the key OpenStack services. |
| FR1.0.2 | At a minimum monitor Nova, Cinder, Glance, Neutron, Keystone, and Ironic services. |
| FR2.0 | Monitor the underlying Host |
| FR2.0.1 | Monitor the CPU, Memory, and Disk |
| FR2.0.2 | Set CPU threshold alerts for 85%, 90%, and 95% |
| FR2.0.3 | Set Memory alerts for 85%, 90%, and 95% |
| FR2.0.4 | Set disk alerts for 80%, 90%, and 95% |
| FR3.0 | Monitor the Guest OS when possible |
| FR3.0.1 | Establish monitoring metrics called out in FR2.0 for each guest when possible |
| FR4.0 | Monitor Network Throughput |
| FR4.0.1 | Monitor Network links for thresholds >90% |
| FR5.0 | Monitor for Noisy Neighbor |
| FR5.0.1 | Set a policy that when a network threshold exceeds 90% |
| FR6.0 | Where possible baseline application requirements for performance. |

Table 4 - Service Quality Management and Systems Management

Regarding FR5.0; although “noisy neighbor” is no longer prevalent in OpenStack due to the advancement of workload isolation, QOS, and resource quotas, a noisy neighbor like complaint could be evidence of a compromised system being used for blockchain mining or some other APT attack that has compromised the guest. A noisy neighbor like event should signal an automatic notification to the application owner that their system needs to be looked at for compromised services.

It is recommended that in conjunction with FR 6.0; for FR 4.0 and 5.0 that the existing spelunk tooling, be used to look for anomalies. By baselining the expected CPU, Memory, disk, and network performance metrics on new applications as they are deployed an anomalous behavior alert can be set up in the spelunk tooling to identify a possibly compromised system.

### Capacity Management

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR2.0 | The system shall provide Capacity Management including: |
| FR2.0.1 | Sufficient instrumentation for {CUSTOMER\_NAME} group to monitor capacity of the RPC control plane. |
| FR2.0.2 | An Internal business process that identifies when additional Ironic nodes are required and triggers their addition to the cabinets. |

### Standard Operating Environment

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR3.0 | The system shall provide a Standard Operating Environment including: |
| FR3.0.1 | RPC OpenStack v24 or higher + Ironic running on Ubuntu 22.04 or higher |
| FR3.0.2 | Cobbler for Provisioning of Ubuntu 22.04 (RPC OpenStack v24 or higher a prerequisite) |
| FR3.0.3 | Jenkins for CI/CD of RPC OpenStack v24 + Ironic control planes |

### Availability Management

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR4.0 | The system shall provide Availability Management including: |
| FR4.0.1 | A 3-node (N+1) RPC OpenStack v24 control plane |

### Data Management

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR5.0 | The system shall provide Data Management including: |
| FR5.0.1 | Ironic control planes with an object store used only for ephemeral storage of glance images (i.e. no user data). |

### 

### Continuity Management

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR7.0 | The system shall provide Continuity Management including: |
| FR7.0.1 | The system shall allow for availability within a specific zone, or across multiple zones, which may require multiple geographies. |
| FR 7.0.3-2 | See recommendations in the Disaster Recovery section for application tiering. |

### Information Security Management

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| FR8.0 | The system shall provide Information Security Management including: |
| FR8.0.1 | The system shall utilize {CUSTOMER\_NAME} existing Active Directory environment for authenticating Keystone users. |
| FR8.0.2 | The system shall utilize internal role assignments for authorizing Keystone users. |
| FR8.0.3 | The system shall adhere to a jointly developed system security plan ad defined by General {CUSTOMER\_NAME } guidelines. |

### Operational Readiness

Operational Requirements define the parameters to keep the product or process functioning over a period of time.

|  |  |
| --- | --- |
| **Section/Requirement ID** | **Requirement Definition** |
| OR9.0 | The system shall provide Operational Readiness including: |
| OR1.0.1 | A documented process for enrolling new capacity into Ironic |
| OR1.0.2 | A documented process for decommissioning capacity from Ironic |
| OR1.0.3 | A documented process for monthly batching of physical hardware break/fix activities |

### Software Requirements

**{Soft\_req\_table}**

Example

|  |  |
| --- | --- |
| **Software Environment Details** | |
| Infrastructure Tier | Ubuntu 22.04 LTS, RPCv21 ussuri current + Ironic should consider Xena for certain and look into Zed release |
| Ironic Workload Images | To be determined based on data analytics requirements |
| Object Storage Tier | Currently not required, however CEPH is recommended |
| Block Storage Tier | PureStorage currently in use today |
| Frameworks | Docker, Cloudera Manager for Hadoop, Kubernetes |
| Monitoring | To be determined by {CUSTOMER\_NAME} monitoring group based on recommendations outlined in the requirements above. |
| Authentication | Active Directory |

# Appendix C – Total cost of ownership

**{TCO}**

THIS PART TO BE FILLED OUT BY {PARTNER\_NAME } – HERE ARE SOME THINGS TO CONSIDER

|  |
| --- |
| **Compute** |
| Server hardware costs - VxRail (Capital)) |
| Server hardware costs - VxRail (Opex) |
| Server hardware costs - Nice Remote 3D (Capital) |
| Server hardware costs - Nice Remote 3D (Opex) |
| Server hardware costs - HPC Reservoir GPU Cluster (Opex) |
| Server hardware costs - HPC Geoscience CPU Cluster (Capital) |
| Server hardware costs - HPC Geoscience CPU Cluster (Opex) |

### 🖥️ {Hardware Requirements}

Example

#### **🔹 HAProxy Servers (x2)**

* CPU: Intel Silver 4514 – 16 cores / 32 threads (each)
* RAM: 384 GB
* Storage: 2 × 960 GB SSD
* Network: 3 × Dual-port 10/25 GbE adapters

#### **🔹 Control Plane Servers (x6)**

* CPU: Intel Silver 4514 – 16 cores / 32 threads (each)
* RAM: 512 GB
* Storage:
  + 2 × 960 GB SSD (boot)
  + 8 × 960 GB SSD (data)
* Network: 3 × Dual-port 10/25 GbE adapters

#### **🔹 Compute Nodes (x12 – 6 per region)**

* CPU: Intel Gold 6430 – 32 cores / 64 threads (each)
* RAM: 24 × 64 GB = **1.5 TB**
* Storage:
  + 2 × 480 GB SSD (OS)
* Network: 2 × 10/25 GbE adapters
* Other: HBA adapters for external storage connectivity

#### **Storage**

* Expected to reuse existing **PureStorage** array (assumed SAN or NVMe-backed).

### {Network Requirements}

* Sufficient 10/25 GbE switch infrastructure to support all nodes and high availability
* VLANs or overlay network configuration for management, storage, and data traffic segregation

### {Software Requirements}

#### **⚙️ Core Software**

* Operating Systems (e.g., RHEL, Ubuntu Server, depending on solution stack)
* Backup Software (e.g., Veeam, Commvault)
* Security Tools (e.g., antivirus, EDR, firewall software)
* Monitoring Tools (e.g., Prometheus, Grafana, Zabbix)

### Cost Structure & Deployment Fee Overview

{DEP\_STRUCT\_FEES}

|  |  |  |
| --- | --- | --- |
| **Category** | **Description** | **Notes / Details** |
| **Deployment Fees** | Rackspace Standard MS Fees | Standard one-time or recurring fees for managed service setup |
| **EE Operational Fees** | Optional – Usually capital costs covered by customer's existing operational budget | May not apply if customer reuses current infrastructure or resources |
| **Partner PS Fees** | {PARTNER\_NAME} Professional Services Fees | Consulting or implementation costs from partner involved |
| **Headcount Fees** | Transfer of skillset from VMware to OpenStack | Generally similar to existing staff costs; reallocation of roles |

### Cost Breakdown Clarifications

* **EE Operational Fees (Optional):**  
   These typically represent capital expenditures and are often already accounted for in the customer’s existing operational budget. They may include hardware, software licenses, or infrastructure resources being reused or repurposed.
* **Partner Professional Services Fees ({PARTNER\_NAME}):**  
   Covers the implementation, configuration, and knowledge transfer services provided by the selected partner. These are typically one-time or project-based costs.
* **Headcount-Related Costs:**  
   Most staffing costs are expected to remain consistent, with existing VMware roles transitioning to equivalent OpenStack responsibilities. This involves a skillset shift rather than net new hiring in many cases.
* **Staffing Role Definition:**  
   Each staffing role should be clearly defined by:
  + Number of people required
  + Base salary per person
  + Fully burdened cost (salary + benefits such as healthcare, retirement, insurance, etc.)
  + Role status:
    - **Current**: Existing role retained as-is
    - **Displaced**: Existing role that becomes partially or fully redundant
    - **New**: Net new position required to support the platform

Example: A **Server Administrator** role may be displaced by up to 80% due to increased automation and self-service capabilities in the OpenStack platform.

* Server Administrator
* Server Engineer
* DBA
* Automation Engineer/CRE/SRE
* Network Engineer
* Storage Engineer
* Security Engineer
* Backup Administrator
* Application engineer/architect/operator

Example

### 📊 Total Operating Costs by Year (USD)

{TCO\_YEAR}

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Year** | **Server Maintenance** | **Software Support** | **Network Maintenance** | **Storage Support** | **Floor Space** | **Power & Cooling** | **IT Training** | **IT Turnover** | **Bandwidth** | **Labor Cost** | **Total** |
| **Year 1** | $110,453 | $619,648 | $11,374 | $23,216 | $213,375 | $91,128 | $374,000 | $148,500 | $133,200 | $5,139,940 | **$6,864,834** |
| **Year 2** | $121,498 | $681,613 | $12,511 | $25,537 | $241,754 | $100,241 | $385,220 | $157,544 | $146,520 | $5,452,962 | **$7,325,399** |
| **Year 3** | $133,648 | $749,774 | $13,763 | $28,091 | $268,733 | $110,265 | $396,777 | $167,138 | $161,172 | $5,785,048 | **$7,814,407** |
| **Year 4** | $147,012 | $824,751 | $15,139 | $30,900 | $298,784 | $121,291 | $408,680 | $177,317 | $177,289 | $6,137,357 | **$8,338,520** |
| **Year 5** | $161,714 | $907,226 | $16,653 | $33,990 | $332,262 | $133,420 | $420,940 | $188,115 | $195,018 | $6,511,122 | **$8,900,460** |
| **Year 6** | $177,885 | $997,949 | $18,318 | $37,389 | $369,567 | $146,762 | $433,569 | $199,572 | $214,520 | $6,907,650 | **$9,503,179** |
| **Year 7** | $195,673 | $1,097,744 | $20,150 | $41,128 | $411,144 | $161,438 | $446,576 | $211,725 | $235,972 | $7,328,325 | **$10,149,876** |

{TCO\_GRAPH}

A graph of a diagram

Description automatically generated with medium confidence

### VMware Licensing & Support Cost Summary

#### **{VMware Licensing Costs}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **CPU Cores** | **Unit Cost (Yearly)** | **Total Cost (Yearly)** | **ACV** | **TCV (3-Year)** |
| VMware MSRP Cost | 768 | $350.00 | $268,800.00 | $268,800.00 | $806,400.00 |
| **VMware Discounted Cost** | 768 | $240.00 | $184,320.00 | $184,320.00 | $552,960.00 |

#### **{Rackspace Support Services}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Service** | **Qty** | **Unit Cost (MRR)** | **Monthly Total** | **ACV** | **TCV (3-Year)** |
| Compute Host Support | 12 | $230.00 | $2,760.00 | $33,120.00 | $99,360.00 |
| Infrastructure Host Support | 6 | $1,150.00 | $6,900.00 | $82,800.00 | $248,400.00 |
| Virtualized Control Plane Support | 0 | $1,150.00 | — | — | — |
| **List Price Total** | — | — | **$9,660.00** | **$115,920.00** | **$347,760.00** |
| Support Spend Discount (7%) | — | — | — | — | — |
| **Discounted Total** | — | — | **$8,961.00** | **$107,532.00** | **$322,596.00** |

### Legend

* **MRR**: Monthly Recurring Revenue
* **ACV**: Annual Contract Value
* **TCV**: Total Contract Value (over 3 years)