Building a compelling Modern Apps and Managed Application Cloud Provider Business

A Natural Partnership for Cloud and Service Providers

Gerrit Lehr
Principal Cloud Strategist
VMware Multi-Cloud Architecture and Strategy
April 2021
# Table of Contents

- Executive Summary ........................................................................................................... 3
- Example Partner Profile .................................................................................................... 3
- IaaS as the fundamental platform ....................................................................................... 4
  - Containers-as-a-Service .................................................................................................. 4
    - VMware Tanzu Kubernetes Grid vs. native Kubernetes ............................................ 5
    - Competitive CaaS Cost Model ..................................................................................... 6
    - CaaS Pricing and Packaging ....................................................................................... 6
    - CaaS Margin Optimization ......................................................................................... 7
    - CaaS Competitive Pricing vs. Hyperscale Public Cloud ............................................. 8
- Managed Application Business .......................................................................................... 10
  - Containers vs. VMs for managed Applications .............................................................. 11
  - Managed Services Capabilities and Tools ..................................................................... 13
  - Managed Application Service TCO Model .................................................................. 14
  - Managed Application Pricing Comparison .................................................................. 16
- Outlook and Expansion to full Tanzu Portfolio ................................................................. 17
- Conclusion ......................................................................................................................... 18
- List of Figures .................................................................................................................... 20
- Appendix 1: TCO Model Parameters ............................................................................. 21
- Appendix 2: Managed Service Platform Implementation Estimate ................................ 23
- Appendix 3: CaaS Price Comparison Estimates ............................................................... 24
- Appendix 4: Managed MySQL Database Instance Price Comparison Estimates .......... 24
Cloud Provider Profile

- 7,000 VMs
- 8.7 Million USD Revenue
- 30-40% Gross Margin
- Price Competitiveness
- Value-Add Differentiation
- Full Automation and Scalability

Executive Summary

While VMware Cloud partners are well positioned and established as trusted Infrastructure-as-a-Service (IaaS) providers, new application patterns and technology trends are providing large opportunities for future growth and customer relationships. The ability to succeed in this developing market often relies on the solid foundation of cost-efficient and reliable IaaS platforms. This white paper builds on the Cloud Provider Total Cost of Ownership (TCO) model for IaaS offerings, which is detailed in a recently published paper. IaaS provides an essential element to generate revenue from modern application and platform services and can have a critical impact on a provider’s profit.

In this white paper, we will focus on modern applications and managed application platform services, ranging from Containers-as-a-Service (CaaS) to databases, web services, middleware and more, all of which can be enabled through the VMware Partner Connect Cloud Provider program. The approach we describe is intended as a guide for providers to offer new services on a proven platform that can be both innovative and competitively priced against the growing portfolios of hyperscale public cloud providers. We will explore in detail a solution stack based on a TCO model for managed applications and modern application platforms that we believe can allow providers to achieve 30-40 percent margin while still being price and service quality competitive against large hyperscale public clouds and including superior service quality. Please note, however, that each provider’s circumstances are unique and that the guidance contained in this white paper does not guarantee a successful outcome or any particular outcome.

The information delivered in this paper comes directly from the VMware Multi-Cloud Architecture and Strategy team, a VMware organization that supports and enables cloud providers around the globe with strategy and architecture consulting services. They annually deliver hundreds of workshops and engagements, including detailed business case analysis and TCO models for partners.

Example Partner Profile

Like the first white paper of this series, the information is based on numbers and insights from real-world providers and the VMware Multi-Cloud Architecture and Strategy team’s experience from working with all kinds and sizes of cloud providers around the world.

In particular, this white paper looks at a hypothetical VMware Cloud Service Provider partner coming from an infrastructure hosting business limited to one region. Throughout the paper, we’ll refer to this partner as “Sample Partner”. It wants to significantly increase revenue from 3.3 to around 8.7 million USD annually by following two strategic guideposts: first, offer the most competitive platform for commodity and price sensitive workloads; and second, add standardized and scalable cloud migration, global data center expansion, disaster recovery and application modernization services.

Achieving this goal requires a range of improvements in the partner’s practice and operations, for example an expansion away from a single datacenter region to an eventual global presence of locations to fulfill the offered services, as well as the associated migration, workload mobility and application modernization capabilities.

Overall, Sample Partner anticipates running more than 7,000 VMs and expects to double the number of customers over the business term of three years. A critical component to financing the growth strategy is to achieve the target gross margin profile of at least 30 percent, which can only be achieved through higher value-added services and the right combination of skilled people, innovative partners, efficient processes as well as standardized and automated tools.
IaaS as the fundamental platform

Cloud provider businesses rely on IaaS as the foundation for successful value-added and higher abstraction services. Even if a service is labeled and marketed as “serverless”, there are of course infrastructure servers running somewhere. The same is true for Platform-as-a-Service (PaaS) or Software-as-a-Service (SaaS). These terms simply describe the fact that a provider is taking care of these servers. Their service levels and everything that comes with them are the provider’s responsibility and the customer does not need to worry about them. Similarly, the unit of sales for these higher abstraction services are typically not servers or virtual machines, but higher abstraction units like seats or users in SaaS offerings and service instances, transactions, storage or compute-time for serverless and PaaS offerings.

For our Sample Partner we utilize a comprehensive TCO model that outlines the positive effect on total costs and margin as well as service quality and capabilities, that VMware Cloud Solution Providers can enjoy though the VMware Cloud Solution Provider platform and by leveraging VMware Cloud Director (VCD), vSphere, NSX and more. The partner defined an average VM of 4 vCPU, 16 GB vRAM and 200 GB of VSAN-based Storage. The TCO for this average VM was calculated at 84.62 USD and allowed for a 40 percent margin and a price of 118.46 USD, which is still competitive with low-priced hyperscale public cloud alternatives. This white paper will adopt the above-mentioned IaaS TCO model.

Containers-as-a-Service

Just like IaaS is the foundation for any higher abstraction cloud service, Containers-as-a-Service builds the foundation for any modern application and platform business. Containers and their de-facto standard for orchestration, Kubernetes, are top of mind for application developers, DevOps and platform operations teams. Kubernetes has often been labeled “a platform for building platforms”³, and was chosen as the basis for Sample Partner’s modern application services portfolio for two main reasons, both of which can enable important synergies for cloud providers:

- **Positioning**: Customers expect an enterprise-ready and managed container offering from any provider they would trust to run their modern applications. The largest hyperscale public cloud providers and many regional providers offer this type of service. It is in high demand, especially from those customers that develop modern applications and embrace increasingly common microservices as their application architecture pattern.

- **Cost optimization**: For higher abstraction platform services, for which a customer does not need to know or care about the underlying infrastructure, containers provide an attractive way to optimize infrastructure utilization, remove unnecessary overhead and reduce overall TCO. This allows VMware Cloud Solution Providers to compete successfully in the market while maintaining the desired margin.

Let’s start with the business model and TCO of building a CaaS offering. At its core, there are two main components to any CaaS offering: The control plane that provides core services to operate the container platform and the worker nodes that run the actual container pods. In a cloud provider setting, there are additional requirements, especially regarding the control plane. Since a CaaS offering typically incorporates the automated lifecycle management of the environment, cloud providers require automation and self-service capabilities for creating, scaling and deleting control plane and worker node instances. These providers also supply and support pre-defined stable Kubernetes versions and integrations into adjacent components for data protection, monitoring, networking and more for their customers to consume. All these services, features and functionalities need to be consumable and billable in a self-service way from multiple tenants and customers.

---

1. [https://www.vmware.com/asaro/585331_REG.html](https://www.vmware.com/asaro/585331_REG.html)
3. [https://twitter.com/kelseyhightower](https://twitter.com/kelseyhightower)
4. [https://www.vmware.com/asaro/585331_REG.html](https://www.vmware.com/asaro/585331_REG.html)

---

“Kubernetes is a platform for building platforms. It’s a better place to start; not the endgame.”

Kelsey Hightower
@kelseyhightower
On Twitter³
27. Nov. 2017
The tool of choice for VMware Cloud Solution Providers to achieve this functionality, is by using the Kubernetes UI and Container Service Extension (CSE) of VMware Cloud Director, which is a free of charge add-on that comes with a user interface, command line interface and an optional backend service, depending on the implementation. CSE and the VCD Kubernetes UI enable the partner to expose self-service capabilities for creating, accessing, resizing, upgrading and deleting Kubernetes clusters, which consist of control plane and worker nodes.

**VMware Tanzu Kubernetes Grid vs. native Kubernetes**

The VCD Kubernetes UI can be used to deploy native upstream Kubernetes clusters, which are free of charge, or VMware Tanzu Kubernetes Grid (TKG) clusters, which is VMware’s enterprise-ready Kubernetes distribution. Before building the financial model around both alternatives, it’s important to understand the differences and consider their financial impact. To make reasonable forecasts for this new technology where real-world experience is limited, numbers were taken from a recent Enterprise Strategy Group study about the economic benefits of operationalizing Kubernetes with VMware Tanzu. The study observed an 87 percent saving for planning and architecture, 70 percent for setting up the platform as well as 94 percent for day 2 operations.

The study was based on the Tanzu Standard Edition which includes very advanced multi-cluster and multi-cloud management capabilities with Tanzu Mission Control. To make these numbers valid for cloud providers using Tanzu Basic Edition, it must be noted that in regard to deployment and lifecycle management of Kubernetes, VCD provides a similar set of capabilities for deployments in a regional partner cloud. Further, cost improvements from the study were reduced by about 10 percent to allow some margin of error from these assumptions:

- **Support**: While TKG comes with enterprise-grade VMware support, native upstream Kubernetes as a free open-source project comes with no commercial support by default. This makes it either very expensive or difficult to build a CaaS practice that can be supported by realistically achievable and financially backed SLAs. The TCO model assumes 85 percent reduction in Kubernetes support FTE costs through Tanzu compared to native upstream Kubernetes.

- **Networking & Security**: Native upstream Kubernetes can be build using numerous networking and security solutions and complex architectures, which need to be integrated, managed and supported. TKG supports NSX-T, which already builds the network foundation for the Sample Partner’s IaaS cloud. Through the already integrated Container Networking Interface (CNI), network resources such as load balancers, container ingress or isolated network namespaces are automatically created in NSX-T as needed by containers. Also, Tanzu-based Kubernetes distributions come with pre-implemented security best practices like hardened base OS images for control plane and worker nodes, admission controllers, role-based access control and more. Building and supporting this stack from open-source tools would be assumed to reduce the number of supported container pods per networking and security FTE by 85 percent and require over three times more engineering and architecture effort to design and setup the base stack.

- **Consumption and Lifecycle Management**: TKG clusters use the open-source Cluster API project for lifecycle management, which in turn uses the vSphere VM Operator to manage the VMs that make up the cluster. This allows providers’ vSphere admins to leverage their existing skills for lifecycle management of Kubernetes clusters, provide developers with the Kubernetes resources they need and lets everyone focus on their core responsibility by making use of existing know-how. This will be attributed to 85 percent saving in day 2 operations costs.

All of these Tanzu features contribute to a TCO reduction compared to a Do-It-Yourself implementation of native upstream Kubernetes but do come with a cost themselves. For Tanzu Basic Edition, there is a VMware Cloud Service Provider program charge per GB of chargeable vRAM. It is an add-on to the VMware Cloud Service Provider program flex core bundle, which allows usage of VMware Cloud Director, vSphere and more. Another requirement is the use of NSX-T Advanced Edition to provide load balancing services for containers, as well as additional shared resources for the Tanzu Supervisor Cluster.

---

**Competitive CaaS Cost Model**

The Tanzu Basic setup requires an estimated 120 hours for a Solution Developer to implement the solution on top of the existing VCD deployment, two additional employees to train, as well as salaries for Kubernetes admins to operate the environment. All of these costs are added to the existing IaaS platform TCO and adjusted accordingly to compare between Tanzu and native Kubernetes implementations. The example Kubernetes clusters for the TCO comparison are running mixed pods ranging from microservices with 0.1 vCPU and 0.2 GB RAM to backend services with 2 vCPUs and 4 GB RAM on worker nodes with 17 vCPUs and 52 GB of vRAM that accommodate the default Kubernetes maximum of 100 pods each.\(^7\)

---

**Figure 1: Average Cost Savings for building CaaS on VCD with Tanzu Kubernetes Grid**

The last important factor to consider is the scale of each cluster as well as the overall platform. Therefore, several calculations ranging from one hundred small clusters with 100 pods to ten thousand large clusters with 100,000 pods each were calculated. The results in figure 1 show that the cost advantage is mainly dependent on the size of Kubernetes clusters and varies between 21 and 37 percent for Tanzu Kubernetes Grid compared to native upstream Kubernetes implementations.

---

**CaaS Pricing and Packaging**

To build a viable business model for CaaS, providers first need to understand the market price and service they are trying to compete with. The first pricing component is the Kubernetes control plane, which is equally priced at 0.10 USD per hour per cluster across Amazon Web Services Elastic Kubernetes Service, Microsoft Azure Kubernetes Service and Google Kubernetes Engine.\(^8\) This equals 73 USD per 31-day-month to provide Kubernetes control plane services to customers. VCD gives customers and partners the option to deploy a single or multiple control plane nodes, which has an impact on the service SLA.

---

\(^7\) https://kubernetes.io/docs/setup/best-practices/cluster-large/
\(^8\) Reviewed in March of 2021:
https://aws.amazon.com/eks/pricing/
https://cloud.google.com/kubernetes-engine/pricing?hl=en
Public cloud CaaS offering SLAs for the control plane range from 99.5 to 99.95 percent, which equals between 3 hours, 39 minutes, 8 seconds and 21 minutes, 54 seconds of acceptable downtime per month. While the initial reaction would usually be to deploy critical components like the Kubernetes control plane in a redundant way using three nodes, meeting even the higher-end SLA of 99.95 percent does not require this. Since control plane nodes in TKG are simple vSphere VMs, providers can make use of vSphere HA to achieve the required uptime even with a single node. The temporary loss of the control plane still would not impact the availability of existing container pods.

Matching the 73 USD target market price that our Sample Partner has defined to be market competitive, while providing similar public cloud scalability and achieving the partner’s target margin of at least 30 percent, is still a challenge. Adding the Tanzu and NSX-T Advanced Edition license costs to the original IaaS TCO model at 30 percent margin for a single instance control plane allows for a size of up to 2 vCPU and 8 GB vRAM. This would mean the control plane can handle up to 10 worker nodes or 1,000 container pods in accordance with Kubernetes best practices. The calculation shows how the partner can meet the market price for the Kubernetes control plane and their margin expectations while maintaining a similar SLA, but not similar scale as a public cloud provider that typically supports up to thousands of worker nodes per cluster. The reasons for that are multifold: Hyperscale public cloud providers with a broad portfolio of services can recover lower or even negative margin from one service with other higher margin services, such as the IaaS, CaaS worker nodes or networking services. Beyond that, they scale across a larger set of customers and workloads, can invest heavily in custom implementations and lack other underlying product features that a VMware Cloud Service Provider delivers, for example from vSphere and NSX.

In either way, 10 worker nodes and 1,000 pods provide enough scale for the majority of anticipated tenant profiles of Sample Partner. It was therefore decided to limit the contractually defined scale per standard Kubernetes cluster at service launch and create a higher SLA and scalability cluster control plane offering for more critical workloads. This gives customers the choice to deploy multiple control plane nodes of different sizes if required and pay on a per node basis.

For the second pricing component of CaaS offerings, the worker nodes, partners can basically use the same numbers as for the standard IaaS VM model. An important difference is that container networking requires the Advanced Edition of NSX-T. Together with Tanzu Basic, this adds additional points to the flex core bundle. For an average 4 vCPU and 16 GB vRAM VM, this would increase overall TCO to 103.81 USD from 84.62. Hyperscale public clouds follow a model where Kubernetes worker node VMs have the same price as standard IaaS VMs. Following this model would bring Sample Partner’s margin down to 14.1 percent for worker nodes. However, VMware Cloud Service Provider partners have two mechanisms that can be used to push their margin back towards the desired 30-40 percent.

CaaS Margin Optimization

One mechanism is to make use of the vRAM cap that VMware grants their partners. Through that, larger worker node VMs that cross the limit of chargeable vRAM will not be charged for the additional vRAM licensing, which improves margins for larger VMs. Even though this licensing mechanism has a positive impact on TCO, it is limited due to the default maximum number of 100 pods per Kubernetes worker node. Another effective way to improve the margin is charging for adjacent add-on services used by the Kubernetes workloads, which is again in line with the hyperscale public cloud model of pricing and packaging. Microsegmentation, load-balancing, domain allow lists and other advanced networking features are already included in the NSX-T Advanced Edition license which is calculated into the CaaS TCO. This allows providers to generate additional revenues and higher margins from these value-add services in the CaaS environment without incremental costs. Also, adding and monetizing capabilities like vRealize Operations Advanced Edition for Monitoring, vRealize Log Insight for Logging, Container Registry Services and more can be used to pursue the desired margin across the CaaS stack.

---

9 Reviewed in March of 2021:
https://aws.amazon.com/iaas-sla/
https://docs.microsoft.com/azure/aks/uptime
https://cloud.google.com/kubernetes-engine/sla
Building a compelling Modern Apps and Managed Application Cloud Provider Business

<table>
<thead>
<tr>
<th>Service Offering</th>
<th>VMware Cloud Provider Cost</th>
<th>Common Public Cloud Charging Model</th>
<th>Provider Margin Optimization Strategy</th>
</tr>
</thead>
</table>
| Kubernetes Control Plane | per Node vRAM                    | per Cluster                       | - Offer with limited scalability  
- Charge per control plane node  
- Offer full control over control plane nodes as an additional option |
| Kubernetes Worker Nodes | per Node based on vRAM           | per Node based on Type            | - Make use of vRAM cap for larger worker nodes                                                      |
| Load Balancer          | NSX-T Advanced included in CaaS TCO | per Load Balancer Instance / Load Balancing Unit | - Charge per load balancer instance  
- Charge per load balancing unit via artificial metric  
- No / low incremental provider cost for service |
| Service Firewall       | NSX-T Advanced included in CaaS TCO | per Firewall Unit                 | - Charge per container host  
- Make use of vRAM cap for larger worker nodes  
- Charge per container protected by a distributed firewall |
| Monitoring             | vROps per Node vRAM / OSI        | per Metric, Dashboard, Alert, Event, Protocol etc. | - Chose per vRAM or OSI to optimize license costs for nodes  
- Charge per metric, dashboard, alert, event, protocol etc.  
- No incremental provider cost for metrics, dashboards, alerts etc. |
| Logging                | vRNI included in Flex Core       | per Event, Alert etc.             | - Charge for logging per instance or event, alert etc.  
- No / low incremental provider cost for events, alerts etc. |
| Container Registry     | per Node vRAM                    | per GB of Storage and Feature set | - Deploy in container and charge per GB of Storage  
- Share between tenants and add managed services |

Figure 2: Provider Margin Optimization Strategies for CaaS and adjacent Services

CaaS Competitive Pricing vs. Hyperscale Public Cloud

With these numbers and optimization strategies defined, it is possible to estimate a comparison between Sample Partner’s offering and hyperscale public cloud alternatives. This will also show that the Sample Partner’s margin expectation can be sustained. For this example, the comparison looks at a Kubernetes cluster running 1,000 mixed pods as previously defined. Because each provider’s cloud service has slightly different characteristics and pricing varies over time and by region, this comparison can only serve as a rough and illustrative example. The exact service components and prices used for this comparison are included in the appendix of this paper.

![Figure 3: Example Price Estimation Comparison for running Kubernetes in the Cloud](image)

**Container as a Service – Kubernetes Cluster Market Price Estimate Comparison**

Cluster running 1,000 pods on 10 hosts with 16 vCPU, 54 GB of RAM and 800 GB of Storage each

Adjacent Services (excludes traffic charges): 500 GB Container Registry, 50 Load Balancer Instances, Firewalling

![Figure 3](image)
Modern Application Platform

To build modern and distributed applications, customers require more than a CaaS offering to run Kubernetes workloads. As stated earlier in this white paper, Kubernetes is a platform for building platforms. The Sample VMware Cloud Service Provider partner therefore focused on providing a range of platform services for consumption by its customers. The portfolio was designed to be launched in a phased approach, which allowed the managed services practice of the partner to grow alongside the self-service portfolio. In Phase 1, which we will focus on in this white paper, the described platform provides all services that developers need to build and run multi-tier or microservices-based web services and web applications, like databases, web- and application servers. Phase 2 of the roadmap focuses on the developer experience and DevOps tools, while the final launch, Phase 3, includes innovation tools and frameworks for BigData and machine learning.

<table>
<thead>
<tr>
<th>Phase 1: Web Services Platform</th>
<th>Phase 2: DevOps Platform</th>
<th>Phase 3: Innovation Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Server: NGINX, Apache HTTPD</td>
<td>CI/CD: Jenkins, GitLab</td>
<td>BigData: Hadoop</td>
</tr>
<tr>
<td>Application Server: Apache Tomcat</td>
<td>Source Code Management: GitHub, GitLab</td>
<td>Machine Learning: Tensorflow, MXNet, PyTorch</td>
</tr>
<tr>
<td>Middleware: RabbitMQ, ActiveMQ</td>
<td>Artifact Management: JFrog Artifactory</td>
<td></td>
</tr>
<tr>
<td>Database: PostgreSQL, MySQL</td>
<td>Infrastructure as Code: Terraform, Ansible</td>
<td></td>
</tr>
<tr>
<td>NoSQL: MongoDB, Cassandra</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Platform Services Phased Launch Plan

Offering these services supports Sample Partner’s strategy in two major ways. It broadens the portfolio way beyond what is available from typical IaaS providers. This portfolio breadth allows the partner to capture many use-cases, that likely would have previously led customers towards hyperscale public clouds. Leveraging the cloud in the context of modern applications is about way more than just VMs and IaaS. Customers need instant availability of tools, services, and platforms to innovate in a fast and agile way. Through this modern application portfolio, the Sample Partner can offer these tools on the regional and trusted platform on which its customers already rely. The platform can counter the risk of losing customers to hyperscale public cloud providers, generates additional revenue from capturing new customer workloads and positions the partner as a true alternative to public clouds. Offering this rich range of services also supports Sample Partner’s strategy because each of these applications offers an opportunity to create incremental revenue from managed services for the application. Examples of applications with this potential include Database-as-a-Service, managed web services, managed Hadoop clusters and more. The TCO detail and business model behind these managed services are described in the next section of this white paper.

The VMware solution of choice to build this modern application service portfolio is the VMware Cloud Marketplace with application packages from Bitnami. Bitnami is the leader in application packaging and provides the largest catalog of click to deploy applications and development stacks. It enables partners and customers to launch their favorites quickly and easily on their own servers or deploy to every major cloud environment. They can choose from local installers, single VMs, multi-tier VMs, container images or Kubernetes Helm charts. Bitnami, which is part of VMware Tanzu, has spent years perfecting the business of packaging applications. During this time, they have built the expertise and internal tooling required to automate the process of publishing and maintaining a catalog of applications and development environments. Bitnami continuously monitors and updates every one of the 200+ applications in its catalog including their components and dependencies to keep these applications and development packages up-to-date and secure – a task that would otherwise be the responsibility of the cloud provider. Applications and stacks are pre-configured and ready-to-deploy immediately on almost any platform. For Sample Partner’s provider setting, the packaged applications can be deployed to either VCD-based virtual machines or Kubernetes-based containers.
Regarding the TCO and profitability of this self-service portfolio, the most crucial factor is that Bitnami does not impact the TCO since it is free of charge to VMware Cloud Service Provider partners. The entire range of 200+ services is available from the VMware Cloud Marketplace to be deployed into the VMware Cloud Director-based tenant clouds. There are multiple ways that partners can publish this catalog or make applications available to their customers’ tenants:

- **Download from Marketplace**: Application packages can be downloaded from the VMware Cloud Marketplace and published to customers via vApps in the VCD catalog. This is the traditional way to publish VM-based apps in VCD.

- **Kubeapps**: Kubeapps is an open-source and community supported web-based UI from the VMware Bitnami team for deploying and managing applications in Kubernetes clusters. Kubeapps can be deployed in one cluster but configured to manage one or more additional clusters, for example on top of the TKG-based CaaS offering.

- **Kubernetes Operators**: Many of the open-source packages available from Bitnami also have Kubernetes Custom Resource Definitions (CRD) and controllers available. These are extensions to a Kubernetes Cluster or a partner’s CaaS offering, which allow customer developers to deploy resources natively from Kubernetes YAML definition files.

- **App Launchpad**: App Launchpad is a free plug-in for VMware Cloud Director that provides a user interface to easily access and launch applications from VMware Cloud Director content catalogs. Using App Launchpad, developers and DevOps engineers can launch applications to VMware Cloud Director in seconds. In addition to VM applications, App Launchpad 2.0 can display and deploy container applications from Helm Charts in the App Launchpad catalog.

While none of these options add additional license costs for the provider, App Launchpad (ALP) is the recommended way forward since it is actively developed and supported by VMware and integrated in VCD.

![Figure 5: Developer Resource Consumption in VCD](image)

**Managed Application Business**

As described in the previous chapter, providing a self-service catalog of modern application components and platforms is mainly a strategy to broaden the portfolio and thereby increase revenues from IaaS and CaaS utilization. The partner still sells VMs and Container resources but does not capture additional sources of revenue. Given that the underlying components Bitnami, VMware Cloud Marketplace, Kubeapps and App Launchpad don’t cause additional costs, it is still a valid way of growing the existing business towards the achievement of revenue goals.
Yet beyond that, a modern application platform portfolio offers an additional opportunity to benefit from increased customer spending and differentiate strongly from competitors: Managed applications. So far, the Sample Partner has given customers the ability to self-service deploy the applications covered by the Phase 1 web services portfolio. But the responsibility to update, upgrade, monitor and manage these applications remains with the customer. Likewise, the service that the customer pays for is the underlying VM or container and their respective SLAs don’t necessarily reflect the requirements of the application. To free customers from these responsibilities, our Sample Partner decided to grow the existing managed infrastructure business to offer managed application services in line with the self-service portfolio. Since this does of course require investments in automation, tools and staffing, building a comprehensive TCO model becomes an important exercise to foster profitability and competitiveness.

Containers vs. VMs for managed Applications

The first question that the TCO model must answer is whether to run managed application instances in VMs or containers. Existing managed service offerings of the Sample Partner were entirely VM-based, which makes sense given that CaaS and the ability to run containers at all had just been added to the portfolio. The natural tendency of the partner is therefore to build the expanded managed application portfolio in a familiar fashion based on VMs. Yet using containers instead of VMs promised a positive impact on the overall TCO for the future managed services, since containers offer better resource utilization, less compute overhead and reduced storage footprint. To validate these assumptions, two TCO models for managed application clusters based on VMs and on containers were calculated.

Assumptions need to be made for the sales forecast and overall workload of the managed application cluster. The assumption was that the Sample Partner would be able to sell 2,500 managed application instances over the business term of 36 months. Beyond that, the required resources and workload profiles of these applications needed to be defined. This is where the difference between VMs and containers has the first impact. While each VM would run its own operating system instance, containers share the host operating system and only add the application-specific portion. The overhead for VM deployments was assumed with the minimal requirements for an Ubuntu Linux 20.04 Server Edition of an additional 1 vCPU, 1 GB of vRAM and 2.5 GB of disk storage. The workload-profile mix for the services planned for Phase 1 were as follows:

- 50% Stateless Micro-services (Web / Application): 0.1 vCPUs, 0.5 GB of vRAM, 5 GB of storage at 50 IOPS
- 25% Stateless Front-end Services (Web / Application): 0.3 vCPUs, 1 GB of vRAM, 15 GB of storage at 150 IOPS
- 15% Stateless Business Services (Application / Middleware) 0.5 vCPU, 2 GB vRAM, 25 GB of storage at 250 IOPS
- 10% Stateful Backend Services (Middleware / Database): 2 vCPU, 4 GB vRAM, 100 GB of storage at 1,000 IOPS

Obviously, these are only rough estimates and overly simplified assumptions based on a snapshot of current customer workloads. The actual workload sizes will vary significantly. Yet since this is true for VM and container pods, it does not have a relevant impact on the overall basis for deciding between both platform alternatives from a TCO perspective.

Beyond the reduction in overhead for running containers compared to VMs, the vRAM cap for VMware Cloud Service Provider partners can bring a cost advantage when running application instances in containers. Since many small application instance VMs by themselves would rarely cross the chargeable limit, they would be fully charged based on their actual vRAM reservation. Kubernetes worker nodes, however, may cross that threshold when running many application instances in containers. This can allow partners to drive down costs and is conceptually illustrated in figure 6.
An additional factor that will have an impact is the pod uptime. Kubernetes and Tanzu Kubernetes Grid come with a so-called Horizontal Pod Autoscaler that automatically scales the number of instances for an application based on the observed CPU utilization or even other application-provided metrics. Auto-scaling in general is possible using VMs, too. Yet in practice, it is rarely implemented thoroughly for several reasons. First, VMs are slower to boot meaning start boot-up times for containers are on average more than 77 times shorter compared to VMs. Beyond that, monitoring the scaling metrics, automation of scaling, stateless container images, scale-out deployment practices and rolling update mechanisms that are required for a reliable auto-scaling platform are already there in the Kubernetes world. Application images are typically stateless, instances get deployed via declarative definition files which are stored and versioned in a trackable repository, deployed via pipelines that integrate seamlessly into the Kubernetes ecosystem and the formerly mentioned Horizontal Pod Autoscaler of Kubernetes handles all the scaling out of the box. Building a similar set of tools and practices for VMs would require a large degree of customization and investment from the Sample Partner. All this means is that it can safely be assumed that managed applications on a VM platform would be running 75 percent of the time since scaling up and down would be a semi-automated task and take longer. Running in containers is assumed to reduce this to 50 percent of uptime across the application portfolio for the Sample Partner, especially due to highly fluctuating front-end and micro-services.

Additional parameters were selected similarly for both deployments. NSX-T Datacenter for Service Providers was assumed to be the Advanced Edition since the application cluster must maintain high security between tenants and workloads through microsegmentation based on the distributed firewall. Physical storage performance and capacity was assumed to be provisioned at a ratio of 60 percent to the allocated resources and at a cost of 0.17 USD per GB per month, memory reservation was calculated at 100%. All other parameters were chosen from to the original IaaS cloud TCO model and can be found in the appendix of this paper. To address the larger failure domain that Kubernetes container hosts create compared to having one VM per application instance, the Kubernetes cluster was designed with 25 percent worker node failover capacity and three control plane nodes for premium managed services with high SLAs.

The resulting TCO comparison between both alternatives resulted in 7.22 USD for running container-based applications versus 37.06 USD for running VM-based applications, which constitutes cost savings of up to 81 percent.

---

1) https://www.researchgate.net/publication/331655088_Collectors_vs_Virtual_Machines_for_Auto-scaling_Multitier_Applications_Under_Dynamically_Increasing_Workloads
These numbers are the costs for an average application instance based on the previously defined application mix and workload profiles. The average workload profile for the containerized instances built from these assumptions are 0.4 vCPU, 1.20 GB of vRAM and 20 GB of storage at 200 IOPS. The costs can easily be calculated for the actual anticipated workload profile per application or service. However, since the sizing and distribution of these workloads are assumptions by themselves and will vary significantly on the real platform, moving forward with the cost for this average workload profile is adequate with regards to estimation the financial numbers.

As shown, running managed application instances in Containers can result in up to 81 percent improvement for the modelled TCO compared to running in VMs. These cost advantages originate from the lower operating system overhead, decreased instance uptime as well as economies of scale when running many instances in fewer larger worker node VMs.

Managed Services Capabilities and Tools

There is more to building a managed application service for modern workloads than running a cost-efficient platform that hosts the application instance and auto-scaling it. Application metrics such as utilization, performance, latency, service status and more need to be collected, presented to customers and actions must be taken accordingly to ensure service delivery in line with the agreed SLA. Tasks like backup, restore, updates and other lifecycle management tasks need to be available to the customers and must be executed in the most efficient and automated way by the platform. All this requires additional tooling and processes inside the managed services practice of cloud providers. Let us map these requirements to tools and solutions available to VMware Cloud Service Provider partners:

- **Lifecycle management of service.** App Launchpad or Kubeapps support secure authentication with role-based access control, deployment into the desired Kubernetes cluster and different catalogs to isolate tenants. From there, customers can request, delete, and even manually scale and upgrade application instances. Likewise, the cloud provider can start a rolling upgrade process to move the managed application to the latest supported version.

- **Deployment customization for service.** The cloud provider and the customer both have requirements to customize the base Bitnami application images:
  - The cloud provider needs to prepare the respective image with service accounts, configuration that allows logging and monitoring as well as additional provider specific settings. These customized images can be stored in a private repository that ALP or Kubeapps uses to deploy application instances from.
  - The customer needs to prepare each application deployment with environment variables such as credentials, network parameters and additional customizations and configurations. ALP and Kubeapps allow the customer to do this out of the box.

## AVG Required Infrastructure

<table>
<thead>
<tr>
<th></th>
<th>App VM</th>
<th>App Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructre VMs</td>
<td>2,500</td>
<td>35</td>
</tr>
<tr>
<td>Total vCPU [#]</td>
<td>616</td>
<td>652</td>
</tr>
<tr>
<td>Total vRAM [GB]</td>
<td>4,331</td>
<td>1,965</td>
</tr>
<tr>
<td>Total Physical Storage Capacity [GB]</td>
<td>33,750</td>
<td>30,000</td>
</tr>
<tr>
<td>Total Physical Storage Performance [IOPS]</td>
<td>337,500</td>
<td>300,000</td>
</tr>
<tr>
<td>Total CNIs in Pool [#]</td>
<td>3,500</td>
<td>3,500</td>
</tr>
</tbody>
</table>

## Typical Workloads Profile

<table>
<thead>
<tr>
<th></th>
<th>App VM</th>
<th>App Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU [vCPU]</td>
<td>1.60</td>
<td>0.40</td>
</tr>
<tr>
<td>Memory [GB]</td>
<td>2.20</td>
<td>1.20</td>
</tr>
<tr>
<td>Storage Capacity Allocated [GB]</td>
<td>23</td>
<td>20</td>
</tr>
<tr>
<td>Storage Performance Allocated [IOPS]</td>
<td>225</td>
<td>200</td>
</tr>
<tr>
<td>NIC/CNI Count [#]</td>
<td>1.40</td>
<td>1.40</td>
</tr>
<tr>
<td>Uptime [%]</td>
<td>75%</td>
<td>50%</td>
</tr>
</tbody>
</table>

## Direct Costs

<table>
<thead>
<tr>
<th></th>
<th>App VM</th>
<th>App Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>23.36</td>
<td>4.30</td>
</tr>
</tbody>
</table>

## Indirect Costs

<table>
<thead>
<tr>
<th></th>
<th>App VM</th>
<th>App Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>13.70</td>
<td>2.92</td>
</tr>
</tbody>
</table>

## Cost per typical App

<table>
<thead>
<tr>
<th></th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>App VM</td>
<td>37.06</td>
</tr>
<tr>
<td>App Container</td>
<td>7.22</td>
</tr>
</tbody>
</table>

## Cost Advantage

|                         | 81%    |

*Figure 7: Cost for Application Instances in VMs vs. on CaaS*
• **Monitoring and dashboards:** Managed application cloud providers need high-end monitoring capabilities for their managed applications. One of the solutions that provides these high-end capabilities is vRealize Operations (vROps), super-charged for application management through Telegraf agents and the VMware vRealize True Visibility Suite (TVS). Again, there is a provider and a customer view on this:
  o The cloud provider needs to collect all metrics relevant to monitoring and ensuring the application complies with the agreed SLAs. Beyond that, comprehensive monitoring capabilities allows the identification of general platform issues or potential to improve the service or costs. This functionality is provided by TVS Enterprise Edition management packs and vROps Enterprise Edition, which support a whole range of different applications like NGINX, Apache HTTPD and Tomcat, Rabbit MQ, MongoDB, MySQL, PostgreSQL, Hadoop and more.
  o The customer needs a way to get an overview, alerts and reports of the deployed applications and key metrics about performance and utilization, preferable in the form of easy-to-use dashboards. For this purpose, vROps can create custom dashboards and reports. The vROps Tenant App for VMware Cloud Director can further expose these custom reports for the customer tenant, send reports and alert via e-mail.

• **Logging:** VMware Cloud Service Providers can make use of vRealize Log Insight (vRLI), which is included in the flex core bundle and therefore does not cause additional license costs. Through pre-defined configuration of the application images and application-specific content packs, the provider can collect all logging information into one repository for easy troubleshooting and reporting.

• **Automation and integration:** Additional task that need to be fulfilled as part of a managed application service will require either additional automation or manual intervention from the managed services team. Both depends on integration into additional systems. The Sample Partner leveraged two main tools for this: vRealize Orchestrator and ServiceNow.
  o vRealize Orchestrator is used for automating basic tasks from within VCD, such as requesting and restoring basic, script-based application backups or creating service-related tickets.
  o For tasks that require manual intervention or advanced workflows as well as customer ticket management, ServiceNow as the established ITSM and workflow management platform of the Sample Partner was integrated with VMware Cloud Director.

• **Billing and chargeback:** To generate cost overviews and bills for the managed applications, partners can use vROps Chargeback as part of the flex core bundle. Besides billing of core VCD resources for tenants, it also allows metering CaaS based on resources from CSE, managed application instances from AppLaunchpad as well as custom billing logics based on vSphere or VCD tags. Charges can be applied for base rates, fixed costs, consumption metrics and conditional rate factors. Though these flexible pricing policies and integrations via API, vROps Chargeback can be used as a standalone billing engine or customized to meter and report metrics for rating and billing to a third-party billing system, like in the Sample Partner’s case.

Managed Application Service TCO Model

Now that the Sample Partner has an understanding of the required tooling and processes as well as the systems and integrations for their Phase 1 managed service portfolio, it can start gathering the associated cost items required to build the end-to-end TCO model.

As already outlined, most VMware software components are already included in the previous calculations, either as part of the core flex bundle or within the TCO model for IaaS and CaaS. The only additions to the managed application TCO are for management through vROps and TVS. TVS only allows for licensing per Operating System Instance (OSI), which makes it easy. vROps Enterprise is a requirement for TVS Enterprise Edition and also adds VMware Cloud Service Provider program consumption points per OSI. For the Sample Partner it was decided to build vROps Enterprise with application monitoring capabilities from Telegraf agents into the standard managed service and include TVS on top in a premium managed service offering. This premium offering will come with advanced monitoring capabilities from TVS as well as superior SLA.
Integrating and development costs are estimated based on annual salary for a solution developer including an overall loading factor to accommodate unproductive times for vacation, sick leave and especially training for these new technologies. The loading factor was chosen higher compared to initial IaaS calculations since for newer technologies, it must be assumed that learning curve effects are yet to develop over time. An additional training effort for 10 employees was calculated to run the standard managed service practice. From here, the costs can be estimated based on the hours needed for each integration. Since the Sample Partner has no prior experience with these particular integrations, the TCO model used estimations from VMware’s Professional Services organization which have supported a range of similar integrations in the past:

- App Launchpad / Kubeapps and private repository setup: 40 hours
- Custom vROps dashboard and report development for application services in VCD: 80 hours
- VCD integration with ServiceNOW for managed services ticket management: 80 hours
- Custom vRO-based orchestration to trigger application backup and restore scripts: 140 hours
- Application image customization to include service accounts, billing tags, logging and monitoring: 80 hours
- Project management overhead: 20%
- Buffer for issues, troubleshooting etc.: 25%

Since these estimated hours originate from VMware Professional Services, partners can outsource implementations to reduce time-to-market and free up internal resources. The details of these engagements and estimates are contained in the appendix.

The monthly recurring managed services and lifecycle labor costs were estimated based on application package release cycles and experience from existing managed application services. At one new release per application per month and 4 hours for image preparation, catalog refresh and supervision of rolling upgrades, the required time would be 36 hours per month. The same amount is factored in for troubleshooting and fixing problems with failed upgrades or other issues.

Adding this up results in total costs of 20.35 USD for a managed application instance including:

- Best-practice compliant and security hardened application service
- Automated deployment with customization options from self-service portal
- Monitoring and logging dashboards, reporting and alerting for the application instance
- Self-service backup capabilities for the application data
- Regularly updated and refreshed versions of the application
- Manual or managed rolling upgrade of the application instances
- Pro-active and SLA-based monitoring and management of the application service, infrastructure and integration
- Secure micro-segmented networking between application instances

### Table: VMware Software Hosting Costs

<table>
<thead>
<tr>
<th>VMware Software Hosting Costs</th>
<th>Integration / Development Labor Costs (Non-Recurring)</th>
<th>Managed Service and Lifecycle Labor Costs (Monthly Recurring)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• VCD, vRLI, vROps Chargeback:</td>
<td>• ALP or Kubeapps and private repository setup</td>
<td>• Updating application base image, testing and catalog refresh</td>
</tr>
<tr>
<td>Included in flex core bundle</td>
<td>• Custom vROps dashboard and report development</td>
<td>• Supervision of rolling upgrades for existing customer deployments</td>
</tr>
<tr>
<td>• ALP / Kubeapps: No additional charge</td>
<td>• VCD integration with ServiceNOW</td>
<td>• Troubleshooting of failed upgrades and other problems</td>
</tr>
<tr>
<td>• Tanzu / CSE: Included in CaaS TCO</td>
<td>• Custom vRO-based orchestration to trigger application backup and restore scripts</td>
<td>• General monitoring and management of managed application services through vROps, vRLI and TVS</td>
</tr>
<tr>
<td>• NSX-T Advanced: Included in App and</td>
<td>• Application image customization to include service accounts, billing tags, logging and monitoring</td>
<td></td>
</tr>
<tr>
<td>CaaS TCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• vROps Enterprise: per OSI Pts.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• vRealize TVS Enterprise: per OSI Pts.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continued from page 15)
Building a compelling Modern Apps and Managed Application Cloud Provider Business

Managed Application Pricing Comparison

To conclude this white paper, let’s try to compare the cost and pricing models for containerized or VM-based managed applications on the VMware Cloud Service Provider platform against a couple of hyperscale public cloud offering price estimates. Given the complexity and differences in public cloud pricing and metrics, this can only serve as an example for comparing the managed application instance against a roughly equivalent service. Since the average application instance plus OS overhead handled by the Kubernetes worker node has an odd size that’s hard to compare against standard hyperscale public cloud offerings, the premium price for a containerized managed app instance of equal size was calculated for the comparison. The results of such illustrative comparisons will typically vary between different instance types, sizes, charging metrics, regions and competitors. Also, some application and platform services are not priced on a per instance but on a per throughput metric, which requires estimating the throughput per instance size to arrive at a more granular metric. The details of this estimated comparison are included in the appendix.

For the premium option of managed application instances, the calculation must add costs for vROps TVS enterprise with additional consumption points per application, increased Kubernetes failover capacity as well as a higher SLA achieved through double the number of managed services staffs that is tasked with monitoring, updating and troubleshooting the environment.

### Managed MySQL Database Instance Price Estimate Comparison

<table>
<thead>
<tr>
<th>VMware Cloud Provider Platform</th>
<th>Amazon Web Services</th>
<th>Microsoft Azure</th>
<th>Google Cloud Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Containers</td>
<td>App VMs</td>
<td>Region: US-east</td>
<td>Region: US-east</td>
</tr>
<tr>
<td>Container Instance Sizing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>including reduced OS overhead</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and Storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 vCPU, 4 GB vRAM, 65 GB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced Storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS RDS for MySQL db.13.medium</td>
<td>Azure DB for MySQL</td>
<td>GCP Cloud SQL for MySQL 2 vCPU, 4 GB RAM</td>
<td></td>
</tr>
<tr>
<td>AWS EBS 65 GB io1 storage</td>
<td>Basic Gen5 2 vCPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>240 provisioned IOPS (60%)</td>
<td>2 vCPU, 4 GB RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Premium managed service incl.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vROps TVS Enterprise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and 99.99% SLA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS CloudWatch monitoring</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Metrics, 5 Dashboards, 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alarms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unlimited dashboards, metrics,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alerts, logging, monitoring</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full NSX Advanced Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>capabilities like micro</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>segmentation, firewall etc.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not included: Load-balancer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>instance charges, Backup</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monthly Listprice</td>
<td>Monthly Listprice</td>
<td>On-Demand monthly</td>
<td>On-Demand monthly</td>
</tr>
<tr>
<td>40% Margin: $47.37</td>
<td>40% Margin: $81.37</td>
<td>Price: $100.77</td>
<td>Price: $69.64</td>
</tr>
<tr>
<td>Monthly Price at 10% Discount</td>
<td>Monthly Price at 10% Discount: $73.23</td>
<td>1 Year RI monthly Price: $86.02</td>
<td>1 Year RI monthly Price: $69.64</td>
</tr>
<tr>
<td>Premium managed service incl.</td>
<td>Basic Support included</td>
<td>Basic Support included</td>
<td></td>
</tr>
<tr>
<td>vROps TVS Enterprise</td>
<td>Developer: From $29</td>
<td>Developer: From $29</td>
<td></td>
</tr>
<tr>
<td>and 99.99% SLA</td>
<td>Business: From $100</td>
<td>Standard: From $100</td>
<td></td>
</tr>
<tr>
<td>AWS CloudWatch monitoring</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Metrics, 5 Dashboards, 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alarms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full NSX Advanced Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>capabilities like micro</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>segmentation, firewall etc.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not included: Load-balancer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>instance charges, Backup</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monthly Listprice</td>
<td>Monthly Listprice</td>
<td>On-Demand monthly</td>
<td>On-Demand monthly</td>
</tr>
<tr>
<td>40% Margin: $47.37</td>
<td>40% Margin: $81.37</td>
<td>Price: $100.77</td>
<td>Price: $69.64</td>
</tr>
<tr>
<td>Monthly Price at 10% Discount</td>
<td>Monthly Price at 10% Discount: $73.23</td>
<td>1 Year RI monthly Price: $86.02</td>
<td>1 Year RI monthly Price: $69.64</td>
</tr>
<tr>
<td>Premium managed service incl.</td>
<td>Basic Support included</td>
<td>Basic Support included</td>
<td></td>
</tr>
<tr>
<td>vROps TVS Enterprise</td>
<td>Developer: From $29</td>
<td>Developer: From $29</td>
<td></td>
</tr>
<tr>
<td>and 99.99% SLA</td>
<td>Business: From $100</td>
<td>Standard: From $100</td>
<td></td>
</tr>
<tr>
<td>AWS CloudWatch monitoring</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Metrics, 5 Dashboards, 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alarms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full NSX Advanced Edition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>capabilities like micro</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>segmentation, firewall etc.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not included: Load-balancer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>instance charges, Backup</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monthly Listprice</td>
<td>Monthly Listprice</td>
<td>On-Demand monthly</td>
<td>On-Demand monthly</td>
</tr>
<tr>
<td>40% Margin: $47.37</td>
<td>40% Margin: $81.37</td>
<td>Price: $100.77</td>
<td>Price: $69.64</td>
</tr>
<tr>
<td>Monthly Price at 10% Discount</td>
<td>Monthly Price at 10% Discount: $73.23</td>
<td>1 Year RI monthly Price: $86.02</td>
<td>1 Year RI monthly Price: $69.64</td>
</tr>
<tr>
<td>Premium managed service incl.</td>
<td>Basic Support included</td>
<td>Basic Support included</td>
<td></td>
</tr>
<tr>
<td>vROps TVS Enterprise</td>
<td>Developer: From $29</td>
<td>Developer: From $29</td>
<td></td>
</tr>
<tr>
<td>and 99.99% SLA</td>
<td>Business: From $100</td>
<td>Standard: From $100</td>
<td></td>
</tr>
<tr>
<td>AWS CloudWatch monitoring</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Metrics, 5 Dashboards, 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alarms</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Reserved Instance pricing was not available for the selected service at the time of writing*
As shown, VMware Cloud Service Provider partners can compete and, in many cases, underbid hyperscale public cloud offerings. The numbers in figure 10 can of course only serve as an estimation and a snapshot at the time of comparison since all services listed are subject to different service characteristics, price changes and regional price differences. Yet it shows that even at a 40 percent margin, the Sample Partner used throughout this white paper remains competitive through use of the VMware Cloud Provider platform and an architecture that optimizes the total costs through automation, software-defined capabilities and containers in the backend. It can even provide superior support and additional services through features like distributed firewall and unlimited monitoring or logging capabilities which would usually incur additional charges at other platforms.

Outlook and Expansion to full Tanzu Portfolio

Besides adding Phase 2 and Phase 3 managed application services following the same model and architecture as detailed for Phase 1, the Sample Partner plans to expand their developer focused portfolio over time. Key components planned on the roadmap are coming from the emerging and growing VMware Tanzu portfolio. The partner’s roadmap already details a plan of future services, value propositions, as well as value-added services and an overview of competitive offerings. Such a roadmap helps to establish the bigger picture internally and externally. It is an important tool to also plan ahead and consider future requirements for the technical architecture, the commercial customer engagement model and the overall business plan. The current roadmap spans across the entire Tanzu portfolio and adds capabilities to build and manage modern applications, besides the now established portfolio for running applications.

The build portfolio contains additional and higher abstraction or automated services that can enhance or run on top the CaaS offering of the Sample Partner. It ranges from automatically assembling compliant container images with Tanzu Build Services (TBS), packaging open-source applications into a curated catalog with enterprise governance from Tanzu Application Catalog (TAC), easily applying caching, messaging and databases from Tanzu Data Services (TDS) all the way to delivering software to a secured and automated platform with Tanzu Application Service (TAS).

<table>
<thead>
<tr>
<th>Solution</th>
<th>Use-Case</th>
<th>Value Proposition</th>
<th>Value-Add options</th>
<th>Compare to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tanzu Application</td>
<td>DELIVER software to a secured</td>
<td>&quot;TAS increases developer productivity and reliable day-2 operations through an</td>
<td>- Platform Operations</td>
<td>AWS Elasticbeanstalk</td>
</tr>
<tr>
<td>Service</td>
<td>and automated platform</td>
<td>integrated platform that combines application runtime, simplified buildpack</td>
<td>- Provide underlying IaaS</td>
<td>Azure App Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>deployment, monitoring, extensibility, routing, storage, scaling</td>
<td>- Design modern Apps</td>
<td>CloudFoundry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and CICD integration, which would otherwise need to be integrated from separate</td>
<td>- Replatform / Refactor Apps</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>solutions.&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tanzu Build Service</td>
<td>ASSEMBLE apps from internally</td>
<td>&quot;TBS helps DevOps teams automate container creation, lifecycle management and</td>
<td>- Customize / Secure Build OS</td>
<td>AWS CodeBuild</td>
</tr>
<tr>
<td></td>
<td>written source</td>
<td>governance at enterprise scale to improve security and automate the</td>
<td>- Provide Repository / Registry</td>
<td>Azure DevOps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source-to-container workflow across all development frameworks, unlike traditional</td>
<td>- Manage Stack Updates</td>
<td>GCP Anthos CloudBuild</td>
</tr>
<tr>
<td></td>
<td></td>
<td>build models which are manual and often lack governance and scalability.&quot;</td>
<td></td>
<td>GitLab AutoBuild</td>
</tr>
<tr>
<td></td>
<td>CURATE data services and</td>
<td>&quot;TAC helps operators maintain a production-ready, curated catalog of open-source</td>
<td>- Customize / Secure Build OS</td>
<td>AWS/GCP Marketplace</td>
</tr>
<tr>
<td></td>
<td>projects from OSS</td>
<td>container images for developers which is easy to govern and audit, unlike open-</td>
<td>- Provide Repository / Registry</td>
<td>Bitnami Catalog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source packages which lack enterprise governance and best practices.&quot;</td>
<td>- Manage Policies</td>
<td>Docker Hub</td>
</tr>
<tr>
<td></td>
<td>APPLY caching, messaging and</td>
<td>&quot;TDS helps developers break down data-monoliths and spawn on-demand caching,</td>
<td>- Add App manages Services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>databases for apps</td>
<td>messaging and database software with high availability and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>autoscaling, unlike DIY solutions which are complex</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>to setup and manage.&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11: Tanzu Build Portfolio and Value Proposition

To manage modern applications and infrastructure end to end, the Tanzu portfolio offers Tanzu Observability (TO), as well as Tanzu Service Mesh (TSM) as an enterprise service mesh to unify microservices across clusters and clouds. For governance and policy management of multi-cloud Kubernetes, partners can add Tanzu Mission Control (TMC) on top of TKG and any other conformant Kubernetes cluster on any cloud.
Building a compelling Modern Apps and Managed Application Cloud Provider Business

For the Sample Partner, TMC and TAC are planned as the first additions to the portfolio. This is tightly aligned with the services detailed in this white paper since it adds additional management capabilities on top of the TKG CaaS offering, as well as additional governance, control and automation for building application packages based on the Bitnami catalog.

Tanzu Observability is also considered as an optional future add-on service that can work in conjunction with vROps but addresses a set of additional use-cases for customer development, SRE and DevOps teams.

Outside of the Tanzu Portfolio, the partner is also evaluating the addition of NSX Advanced Load Balancer to enable additional networking capabilities across clouds in future iterations of the CaaS offering.

Conclusion

The purpose of this white paper is to empower VMware partners to build a profitable and future-ready portfolio of modern application services. The Sample Partner’s journey detailed herein is very common across the cloud provider landscape: Starting with a profitable CaaS offering built on Tanzu Kubernetes Grid as the foundation for running modern applications, adding modern application packages in an easy to consume and managed way with Bitnami, vRealize Operations and True Visibility Suite, as well as having a long-term portfolio strategy that incorporates the whole range of developer and DevOps use-cases in alignment with the evolving VMware Tanzu portfolio for cloud provider partners.
As shown, it is imperative to conduct a thorough TCO analysis and calculate the optimal model for maximized profitability along the way and for every service offering. Through that process, partners can achieve competitive pricing while maintaining a margin of 30 percent and more. This is possible through advanced and software-defined capabilities from the VMware Cloud Service Provider platform. Based on the example numbers and models described in this white paper, it is possible to reduce total cost per service by up to 80 percent for application services and above 37 percent on average for container services.

To learn more and get started with building a compelling modern apps and managed application cloud provider business, reach out to your partner manager or visit https://cloud.vmware.com/providers/.
List of Figures

Figure 1: Average Cost Savings for building CaaS on VCD with Tanzu Kubernetes Grid .......................................................... 6
Figure 2: Provider Margin Optimization Strategies for CaaS and adjacent Services .............................................................. 8
Figure 3: Example Price Estimation Comparison for running Kubernetes in the Cloud ................................................................. 8
Figure 4: Platform Services Phased Launch Plan ................................................. 9
Figure 5: Developer Resource Consumption in VCD ........................................ 10
Figure 6: Conceptual Example of Cost Advantage for running managed Apps in Containers vs. VM ................................................ 12
Figure 7: Cost for Application Instances in VMs vs. on CaaS ........................................................ 13
Figure 8: Additional Cost Components for Managed Application Cost Model .......................................................... 15
Figure 9: Standard vs. Premium Managed Services Offerings .......................................................... 16
Figure 10: Example Hyperscale Public Cloud Price Estimation and Feature Comparison ......................................................... 16
Figure 13: Tanzu Build Portfolio and Value Proposition ................................................. 17
Figure 14: Tanzu Manage Portfolio and Value Proposition ................................................. 18
Figure 15: vRealize Operations and Tanzu Observability positioning ................................................. 18
Figure 16: Offering Structure and Evolution to full Tanzu Portfolio ................................................. 19
Appendix 1: TCO Model Parameters

The following cost items were reflected in the TCO models:

- **Term for business modelling**: 36 months
- **Depreciation of hardware**: 60 months
- **Network and security costs per physical network port**: 1,500 USD
- **Rack costs p.a.**: 25,000 USD
- **Rack units occupied by servers**: 30
- **FTE salaries per year**:
  - Server Admin: 89,960 USD
  - VMware Admin: 101,631 USD
  - Kubernetes Admin: 115,917 USD
  - Storage Admin: 129,414 USD
  - Network and Security Admin: 105,923 USD
  - Software Engineer: 111,983 USD
  - Consultant: 245,000 USD
  - Product Manager: 150,000 USD
- **Loading factor for FTEs**: 1.43
- **Labor Productivity**:
  - Servers per FTE: 100
  - VMs per FTE: 800
  - Kubernetes Pods per FTE: 80,000
  - Networking ports per FTE: 2,500
  - CNIs per FTE: 50,000
  - Storage GBs per FTE: 250,000
- **Annual Working hours**: 1,950
- **Physical Server Profile**:
  - CPU per Server: 2
  - Core per CPU: 28
  - GHz per Core: 2.7
  - Hyper-Threading: Enabled
  - Ram per Server: 1,536 GB
  - Network Port per Server: 4
  - Cost per Server: 30,000 USD
  - Rack Units per Server: 1
  - Other monthly Costs: 18.50 USD for ITSM management software
- **Resource Management VMs**:
  - Oversubscription CPU Target: 4 vCPU per 1 pCPU
  - Oversubscription RAM Target: 1:1
  - Virtualization Overhead: 15%
  - Physical Storage Capacity Provisioned: 83%
  - Physical Storage Performance Provisioned: 60%
  - VSAN Failures to tolerate: 2
  - Fault Tolerance Method: Raid 6
- **Resource Management Kubernetes Clusters**:
  - Capacity for Failover: 25%
  - Pods per Worker Node: 100
  - Memory Reservation: 100%
  - Physical Storage Capacity Provisioned: 60%
  - Physical Storage Performance Provisioned: 60%
  - Hyper-Threading: No
- **Average VM**:
  - No. of vCPU: 4
  - vCPU Speed: 2.7 GHz
  - Memory Allocated: 16 GB
  - Storage Capacity per VM: 200 GB
  - Storage Performance per VM: 580 IOPS
  - Storage Workload Profile:
    - Access Pattern: Random
    - IO Size: 4 KB
    - IO Ratio: 70/30
Building a compelling Modern Apps and Managed Application Cloud Provider Business

- Dedupe: 1.42
  - Avg. vNIC count per VM: 2
  - Uptime per VM: 100%
  - Additional Recurring Costs per VM: 2.5 USD for server management
- Shared Costs:
  - DC Sites with Management Pods: 2
  - VMs per Pod: 30
  - Typical management VM:
    - Memory: 32 GB
    - Disk Size: 550 GB
    - VM Memory Reservation: 50%
    - VM Uptime: 100%
  - Physical Storage Capacity Provisioned: 60%
  - Servers per Management Pod: 3
  - CPU Sockets per Server: 2
  - Cores per CPU: 16
  - RAM per Server: 256 GB
  - Network Ports per Server: 2
  - Cost per Server: 12,010 USD
  - Rack Units per Server: 1
  - Other monthly Costs: 18.50 USD for ITSM management software
- Solution Engineering Costs:
  - No. Of Engineering FTEs: 3
  - Engineering Duration: 4 Months
  - Engineering Time Allocated to Project: 75%
- Development Costs:
  - No. Of Development FTEs: 3
  - Engineering Duration: 4 Months
  - Engineering Time Allocated to Project: 75%
- Additional CaaS Implementation Costs:
  - 120 hours internal solution engineer for Tanzu and CSE: 9,854.46 USD
  - 363.63 hours internal solution engineer for native open-source Kubernetes: 25,640.55 USD
- Training Costs:
  - No. of FTEs to train for IaaS: 10
  - No. of additional FTEs to train for TKG CaaS: 2
  - No. of additional FTEs to train for TKG CaaS: 24
  - Training Cost per FTE: 2,865 USD
- Workload Profiles:
  - 50% Stateless Micro-services (Web / Application): 50 IOPS, 1 Network Interface
    - Container: 0.1 vCPUs, 0.5 GB of vRAM, 5 GB Storage, 40% Uptime per Pod
    - VM: 1.1 vCPUs, 1.5 GB of vRAM, 7.5 GB Storage, 70% Uptime per VM
  - 25% Stateless FrontEnd Services (Web / Application): 150 IOPS, 2 Network Interfaces
    - Container: 0.3 vCPUs, 1 GB of vRAM, 15 GB Storage, 40% Uptime per Pod
    - VM: 1.3 vCPUs, 2 GB of vRAM, 17.5 GB Storage, 70% Uptime per VM
  - 15% Stateless Business Services (App / Middleware): 250 IOPS, 2 Network Interfaces
    - Container: 0.5 vCPUs, 2 GB of vRAM, 25 GB Storage, 70% Uptime per Pod
    - VM: 1.5 vCPUs, 3 GB of vRAM, 27.5 GB Storage, 85% Uptime per VM
  - 10% Stateful Backend Services (Middleware / Database): 1000 IOPS, 1 Network Interface
    - Container: 2 vCPUs, 4 GB of vRAM, 100 GB Storage, 95% Uptime per Pod
    - VM: 3 vCPUs, 5 GB of vRAM, 102.5 GB Storage, 95% Uptime per VM
## Appendix 2: Managed Service Platform Implementation Estimate

<table>
<thead>
<tr>
<th>Integration Steps</th>
<th>LoE (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consume VCD for Horizontal Auto Scaling of Applications workshop</td>
<td>8</td>
</tr>
<tr>
<td>Write Technical Specification</td>
<td>8</td>
</tr>
<tr>
<td>Review the Architectural Design</td>
<td>2</td>
</tr>
<tr>
<td>Implement solution installer and configurator</td>
<td>8</td>
</tr>
<tr>
<td>Deploy and configure an of Applications service instance</td>
<td>1</td>
</tr>
<tr>
<td>Verify the installation and configuration of an instance of the service</td>
<td>4</td>
</tr>
<tr>
<td>Update and Finalize Documentation</td>
<td>8</td>
</tr>
<tr>
<td>Workshop conducted to outline the solution design, configuration and how to use it</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>40</strong></td>
</tr>
</tbody>
</table>

### Consume VCD for ServiceNow Incidents workshop
- Write Technical Specification: 8 hrs
- Review the Architectural Design: 4 hrs
- Implement solution installer and configurator: 4 hrs
- Implement VCD user interface for creating new incidents: 16 hrs
- Implement VCD user interface for showing existing incidents: 8 hrs
- Implement VCD user interface for updating incidents: 24 hrs
- Deploy and configure a Consume VCD for ServiceNow service instance: 1 hr
- Verify the deployment and configuration of a Consume VCD for ServiceNow incidents service instance: 4 hrs
- Update and Finalize Documentation: 8 hrs
- Workshop conducted to outline the solution design, configuration and how to use it: 2 hrs

### Total
- 80 hrs

### Tailored use case requirements workshop
- Tailored use case review customer review: 4 hrs
- Review current state infrastructure: 0.5 hrs
- Configuration of vCenter adapter instances: 0.5 hrs
- Tailored configuration use case workshop - identify key customer use case aspects: 1 hr
- Tailored configuration use case workshop - foundational product principles: 1.5 hrs
- Custom group creation interactive workshop (VMS, apps, objects, etc.): 2 hrs
- Custom dashboard creation: 24 hrs
- Custom view creation: 4 hrs
- Custom report creation: 4 hrs
- Custom super metric creation: 4 hrs
- Custom metric interaction file creation: 4 hrs
- Custom symptoms creation: 4 hrs
- Custom alerts creation: 4 hrs
- Custom recommendations creation: 4 hrs
- Custom notifications creation: 4 hrs
- Custom policies creation: 2 hrs
- Custom security/access control creation: 2 hrs
- Update Solution Specification workbook document: 4 hrs
- Tailored configuration use case review workshop: 4 hrs
- Use Case Knowledge Transfer: Supplemental knowledge transfer for tailored use case: 1.5 hrs

### Total
- 80 hrs

### Container Apps Image customization workshop
- Write Technical Specification: 2 hrs
- Review the Architectural Design: 2 hrs
- Configure Identity Provider: 1 hr
- Verify service account prerequisites for vRealize Operations Manager and vRealize Log Insight instances: 0.5 hrs
- Install/Update single Management Pack for Log Insight (as needed): 0.5 hrs
- Validate health of adapter instance within vRealize Operations Manager: 0.5 hrs
- Validate health of vRealize Operations Manager configuration within vRealize Log Insight instance: 0.5 hrs
- Validate: "Launch in Context" feature with corresponding vRealize Operations Manager instance: 0.5 hrs
- Review solution design decisions (Service account, target vCenter instance, etc): 1 hr
- Perform validation of Metering Deployment: 1 hr
- Create and test basic backup script: 1.5 hrs
- Create and test basic restore script: 1.5 hrs
- Create and test metering, charging and billing script: 1.5 hrs

### Total per App
- 16 hrs

### Total for the Application types
- 80 hrs

### Customized VCD for Application Data Protection workshop
- Write Technical Specification: 8 hrs
- Review the Architectural Design: 8 hrs
- Implement VCD user interface for viewing state of App instance backups: 16 hrs
- Implement VCD user interface for protecting an App instance through backup scripts: 16 hrs
- Implement VCD user interface for removing an App instance from backup policies: 12 hrs
- Implement VCD user interface for initiating an on-demand recovery of App instance through restore script: 24 hrs
- Implement VCD user interface for performing on-demand restore of App instance through restore script: 24 hrs
- Implement VCD user interface for entitling tenants to the backup service: 8 hrs
- Install and configure an instance of the service: 2 hrs
- Verify the installation and configuration of an instance of the service: 8 hrs
- Update and Finalize Documentation: 8 hrs
- Workshop conducted to outline the solution design, configuration and how to use it: 4 hrs

### Total
- 160 hrs
Appendix 3: CaaS Price Comparison Estimates

Amazon Web Services estimation for region US-east 1 from https://calculator.aws/#/estimate (March 2021):
- Amazon EC2 (Operating system (Linux), Quantity (10), Storage amount (800 GB), Instance type (m5.4xlarge):
  - On-Demand pricing: 6,406.50 USD
  - 1 Year Reserved Instance pricing: 4,333.20 USD
  - 3 Year Reserved Instance pricing: 3,223.60 USD
- Amazon EKS (Number of EKS Clusters (1)): 73.00 USD
- Amazon Elastic Container Registry (Data transfer cost (0), Amount of data stored (500 GB per month)): 50.00 USD
- AWS Elastic Load Balancer (Number of Network Load Balancers (50)): 1,040.75 USD
- AWS Web Application Firewall (Number of Web Access Control Lists utilized (50 per month)): 250.00 USD

- Azure Kubernetes Service (10x D5v2: 56 GB RAM, 16 vCPU, 800 GB temp storage):
  - On-Demand pricing: 8,541.00 USD
  - 1 Year Reserved Instance pricing: 3,620.00 USD
  - 3 Year Reserved Instance pricing: 32,346.66 USD
- Azure Kubernetes Service (Cluster Uptime SLA): 73.00 USD
- Azure Container Registry Premium: 50.00 USD
- Azure Load Balancer Standard (50 rules): 346.75 USD
- Azure Firewall Standard: 912.50 USD

- GKE Cluster Management Fee (Regional Cluster): 73.00 USD
- GKE Node Pool (10x n1-standard-16 regular with 2x 375 GiB local SSD)
  - On-Demand pricing: 4,483.58 USD
  - 1 Year Commitment Term pricing: 3,877.60 USD
  - 3 Year Commitment Term pricing: 2,766.77 USD
- Google Cloud Storage for Container Registry: 10.00 USD
- Google Cloud Load Balancing (global, 50 rules): 346.76 USD
- Google Cloud Armor (50 policies): 250.00 USD

Appendix 4: Managed MySQL Database Instance Price Comparison Estimates

Amazon Web Services estimation for region US-east 1 from https://calculator.aws/#/estimate (March 2021):
- Amazon RDS for MySQL (Single-AZ, Instance type db.t3.medium):
  - On-Demand pricing: 6,406.50 USD
  - 1 Year Reserved Instance No Upfront pricing: 34.89 USD
  - 3 Year Reserved Instance Partial Upfront pricing: 11.90 USD monthly + 429 USD upfront / 12
- Amazon EBS provisioned IOPS SSD (io1) with 240 IOPS and 65 GB: 32.125 USD
- Amazon CloudWatch 5 Dashboards, 10 Alarms, 10 Metrics: 19.00 USD

- Azure Kubernetes Service (10x D5v2: 56 GB RAM, 16 vCPU, 800 GB temp storage):
  - On-Demand pricing: 49.64 USD
  - 1 Year Reserved Instance pricing: n/a
  - 3 Year Reserved Instance pricing: n/a
- Azure Database for MySQL Disk 65 GB: 65x 0.10 USD = 6.50 USD
- Azure Monitor 10 Metrics, 10 Metric Signals, 5 Protocol Signals, 1 ITSM Connector: 13.50 USD

- Google Cloud SQL for MySQL custom instance with 2 vCPU and 4 GB RAM
  - On-Demand pricing: 2x 32.26 USD + 4x 5.47 USD = 86.40 USD
  - 1 Year Commitment Term pricing: 2x 24.20 USD + 4x 4.11 USD = 64.84 USD
  - 3 Year Commitment Term pricing: 2x 15.49 USD + 4x 2.63 USD = 41.50 USD
- Google Cloud SQL Storage for 65 GB SSD: 65x 0.1819 USD = 11.82 USD
- Google Cloud operations Suite Metrics & Logging: free (limited)