Cloud Computing Homework 2: Assessing Hosting Options

Detailed Report -1

Time Tracking Information:

Approximately 8 hours taken to accomplish the task which included:

- Around 2 hours of thorough analysis of the task required.
- Around 2 hours of research on the data and trends of many companies along with the hardware and software, i.e. cloud requirements and costs.
- Around 1 hour in understanding the working and assumptions of the TCO Calculator.
- Around 3 hours in comprehensively linking all the information and preparing the report.

Task Specifications:

The company, Aggie Startup Consultants, has a new prospective client: a startup in College Station that wants help in implementing a new mobile app and its backend services. The company is in stealth mode, and it has been very successful in raising venture capital. They want to avoid information leaks about their new idea and its business model, so they are somewhat vague at this stage.

They have asked to provide estimates on how much money they need to spend to host their application, even though they are revealing as little as possible about what kind of application they want to run. A report answering the prospective client's questions with much precision is needed with all assumptions clearly specified.

Application Requirements:

1) The app is a multimedia-heavy application and thus will require large storage to occupy videos/images uploaded form the user's servers.

2) There are 80% of active users that upload videos/images on a daily basis along with daily reports of listened songs out of which 10% are heavy content generators while rest 20% are low content generators. The remaining 20% of the users don't generate media but only consume them.

3) The application uses an image/video processing algorithm which composes thematic videos and suggests them to the user based on the user's library of photos/videos and his/her musical preferences. This is a highly compute-intensive task which involves image processing and machine learning to classify the media into different themes. This must be done in a highly efficient distributed system to reduce the processing time.

4) The founders of the application expect to attract a few celebrities to their platform, so that their videos go to the top of trending lists, generating more usage of their app. This means that some of their videos will be extremely popular. Other generated videos will never be seen by anyone. To improve user experience, the access to the most popular videos should have as little latency as possible, while the access to unpopular videos can be less efficient. Popular videos will require

a fast data retrieval system so as to make them quickly available on demand. For reducing latency, the application would need to store such videos with high availability and replication across all servers.

5) The application both read and write heavy. It asks users to provide access to their Google Photos, iCloud and similar repositories and uploads the media therein to its backend. So, continuous media uploads are expected. Also, since users can make their videos public, all the users can view the published media along with the composed thematic videos that will be generated automatically by the backend.

6) The founders are assuming growth of user base similar to Pinterest. Pinterest was launched in March, 2010. According to Wikipedia, by December 2010, it had 10,000 users. After the launch of mobile/tablet apps in 2011 the user base expanded to about 1.5 million. In January 2012, as reported by comScore, the site had 11.7 million unique visitors in the US itself. In Feb 2013, Pinterest had 48.7 million users globally. Over a period of just 3 years, the company amassed about 50 million users worldwide, which is the figure that founders are expecting for their application after a period of 3 years.

Assumptions/Estimations -

1) Data Estimations:

- Consider the User Base to be X. The average image size as 2MB, average video size as 80 MB/minute, and average video lengths as 2 minutes is assumed by me.
- 10% of X are heavy content generators, i.e. 100 photos and 10 videos per day. That accounts for a total of 1800MB of data per user per day, i.e., 180X MB of data per day.
- 70% of X are regular users uploading 5 photos every day and 0.3 videos per day. This makes up 58MB of data per user per day, i.e., 40.6X MB of data per month.
- The remaining 20% of the application users do not contribute to content creation at all.
- The total data generated equals 220.6X MB/day.
- The current student counts for UT Austin and Texas A&M are 51,331 and 68,625 respectively, a total of 119956. The first month after launch of the application, it is expected to garner 5% of this population as its users. That makes about 6000, i.e. X, users in the first month.
- The total amount of data that the application's backend needs to handle in the first month alone is around 1.26 terabytes per day.
- At around 48 million users in 3 years of its launch, Pinterest was able to scale its business because it was built on Amazon Web Services (AWS). Pinterest uses AWS to manage a high-performance social application that stores more than 8 billion objects and 400 terabytes of data in the AWS Cloud using Amazon Simple Storage Service (Amazon S3), and 225,000 instance hours a month with Amazon Elastic Compute Cloud (Amazon EC2). As we are expecting the same growth rate as of 50-60 million users after a period of 3 years, I am assuming a 500TB Storage Capacity.

2) Machine Estimations:

- The image/video processing algorithms have been optimized to run on 2 configurations.
- The assumptions for the "Powerful Machines" configuration and the "Smaller Machines" configuration is as follows:

- Configuration 1 (Powerful Machines):
    - 4 machines, 16 cores, GPU, NAS Storage, and 32 GB RAM
- Configuration 2 (Smaller Machines):
    - 16 computers, 4 cores, Regular SAN Block Storage, and 8 GB RAM

3) Cost Estimations:

- The TAMU-based Private Cloud offers the following for the first three years: Allowing the app to host as many computer racks as the app needs at 5% of the average rates for San Francisco or New York.
- The average rates for San Francisco and New York are computed as follows: According to the CIA World Factbook, United States' per capita income is approximately 59,000$. According to Open Data Network, Austin's per capita income is about 61,000$, which is very close to the national average. Now Austin's Cost of Living is then compared with San Francisco's and New York's using Bankrate's Cost of Living Calculator for buying cost comparison and it can be deduced that these two cities are on average 80% more expensive than Austin.
- By using this approach and the average US rack costs, it can be deduced that rack costs will be around 80% more in San Francisco and New York than the national average.
- Here, the Rack costs are being considered for San Francisco and New York only and not the national average. However, for all the other costs, i.e. Virtualization Software Costs, Maintenance and Operation Costs, etc. the national average has been considered as it is very close to Austin's Per Capita Income.
- IT Labour Cost and Network Traffic Cost is not charged.
- Disaster Recovery is also available free of cost.

Cost Analysis and Comparison Methodology

The cost analysis and comparison contained in this report are based on the following assumptions.

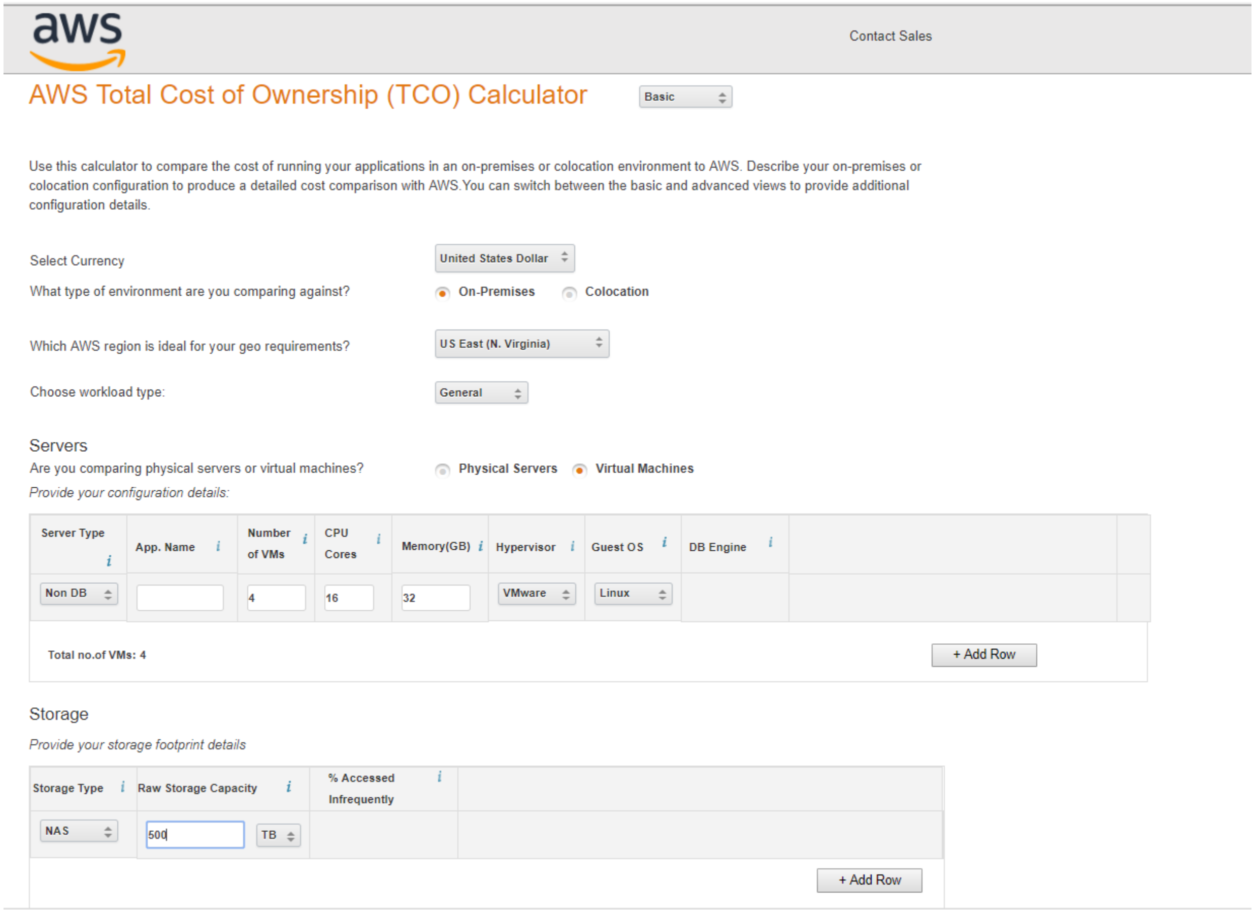
Amazon Web Services (AWS), the most widely used Public Cloud is chosen (US-West Northern California Region) as the cloud service to compare against the TAMU-based Private Cloud. AWS offers a Total Cost of Ownership Calculator through which entities can compute estimated costs for utilizing AWS for three years and comparing it with the average expenses to using a Private Cloud. This tool was leveraged to compute the costs based on the assumptions above and the costs were modified as per our requirements.

The Cost Analysis (Public Cloud vs TAMU-based Private Cloud) of both the Configuration-1 and Configuration-2 for the period of three years is listed below. Here, 'OnPremises' refers to the TAMU-based Private Cloud. The 'On-premises' server prices are based on Dell PowerEdge Rack servers and HP ProLiant Rack servers.

Disclaimer: The costs mentioned in report on are crude estimates and approximations and may wary considerably from actual numbers.
Also, please consider all the assumptions of the TCO Calculator from the following link:
https://aws.amazon.com/tco-calculator/

Cost Analysis and Comparison (Configuration 1: Powerful Machines) Vs. (AWS)

Input Parameters:



AWS Total Cost of Ownership (TCO) Calculator   Basic

Use this calculator to compare the cost of running your applications in an on-premises or colocation environment to AWS. Describe your on-premises or colocation configuration to produce a detailed cost comparison with AWS.You can switch between the basic and advanced views to provide additional configuration details.

Select Currency: United States Dollar

What type of environment are you comparing against? ● On-Premises ○ Colocation

Which AWS region is ideal for your geo requirements? US East (N. Virginia)

Choose workload type: General

Servers

Are you comparing physical servers or virtual machines? ○ Physical Servers ● Virtual Machines

Provide your configuration details:

Server Type	App. Name	Number of VMs	CPU Cores	Memory(GB)	Hypervisor	Guest OS	DB Engine
Non DB		4	16	32	VMware	Linux	

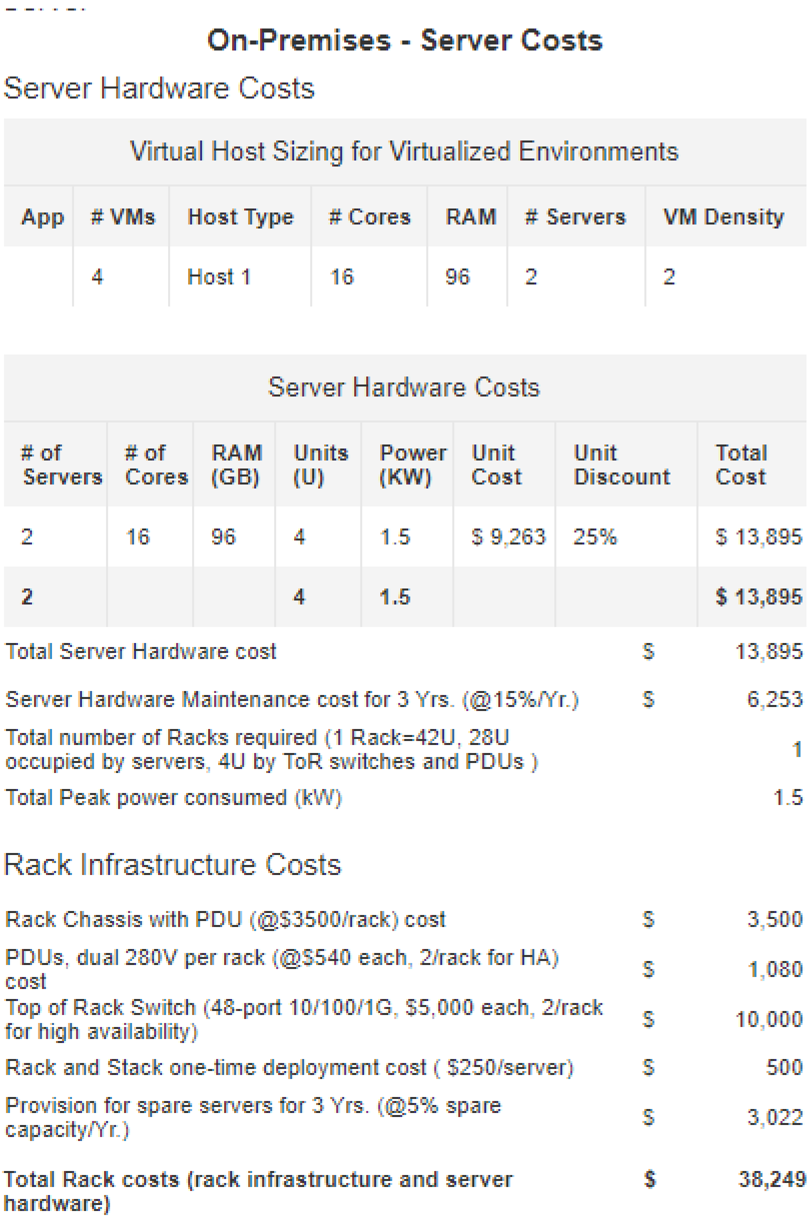
Total no.of VMs: 4

Storage

Provide your storage footprint details

Storage Type	Raw Storage Capacity	% Accessed Infrequently
NAS	500 TB	

Private Cloud – Configuration 1 Costs:



On-Premises - Server Costs

Server Hardware Costs

Virtual Host Sizing for Virtualized Environments						
App	# VMs	Host Type	# Cores	RAM	# Servers	VM Density
	4	Host 1	16	96	2	2

Server Hardware Costs							
# of Servers	# of Cores	RAM (GB)	Units (U)	Power (KW)	Unit Cost	Unit Discount	Total Cost
2	16	96	4	1.5	$ 9,263	25%	$ 13,895
2			**4**	**1.5**			**$ 13,895**

Total Server Hardware cost	$	13,895
Server Hardware Maintenance cost for 3 Yrs. (@15%/Yr.)	$	6,253
Total number of Racks required (1 Rack=42U, 28U occupied by servers, 4U by ToR switches and PDUs)		1
Total Peak power consumed (kW)		1.5

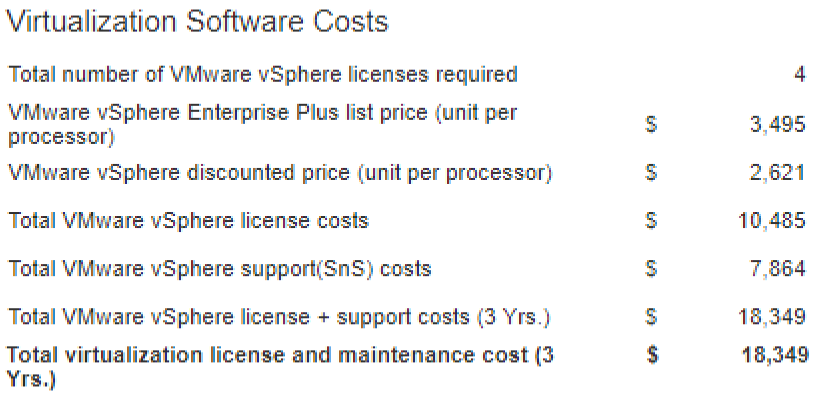
Rack Infrastructure Costs

Rack Chassis with PDU (@$3500/rack) cost	$	3,500
PDUs, dual 280V per rack (@$540 each, 2/rack for HA) cost	$	1,080
Top of Rack Switch (48-port 10/100/1G, $5,000 each, 2/rack for high availability)	$	10,000
Rack and Stack one-time deployment cost ($250/server)	$	500
Provision for spare servers for 3 Yrs. (@5% spare capacity/Yr.)	$	3,022
Total Rack costs (rack infrastructure and server hardware)	**$**	**38,249**

Total Rack Costs Adjusted for San Francisco or New York, i.e. 80% More = $ 68848.2

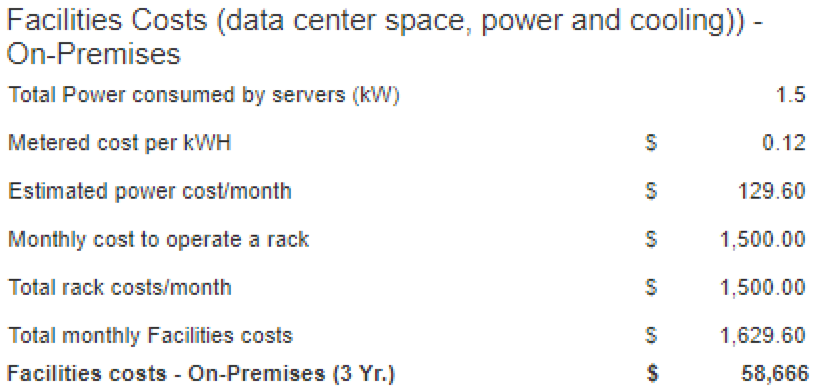
Total Rack Costs After 95% Discount = $ 3442.41

Also Networking Costs are not taken into consideration.



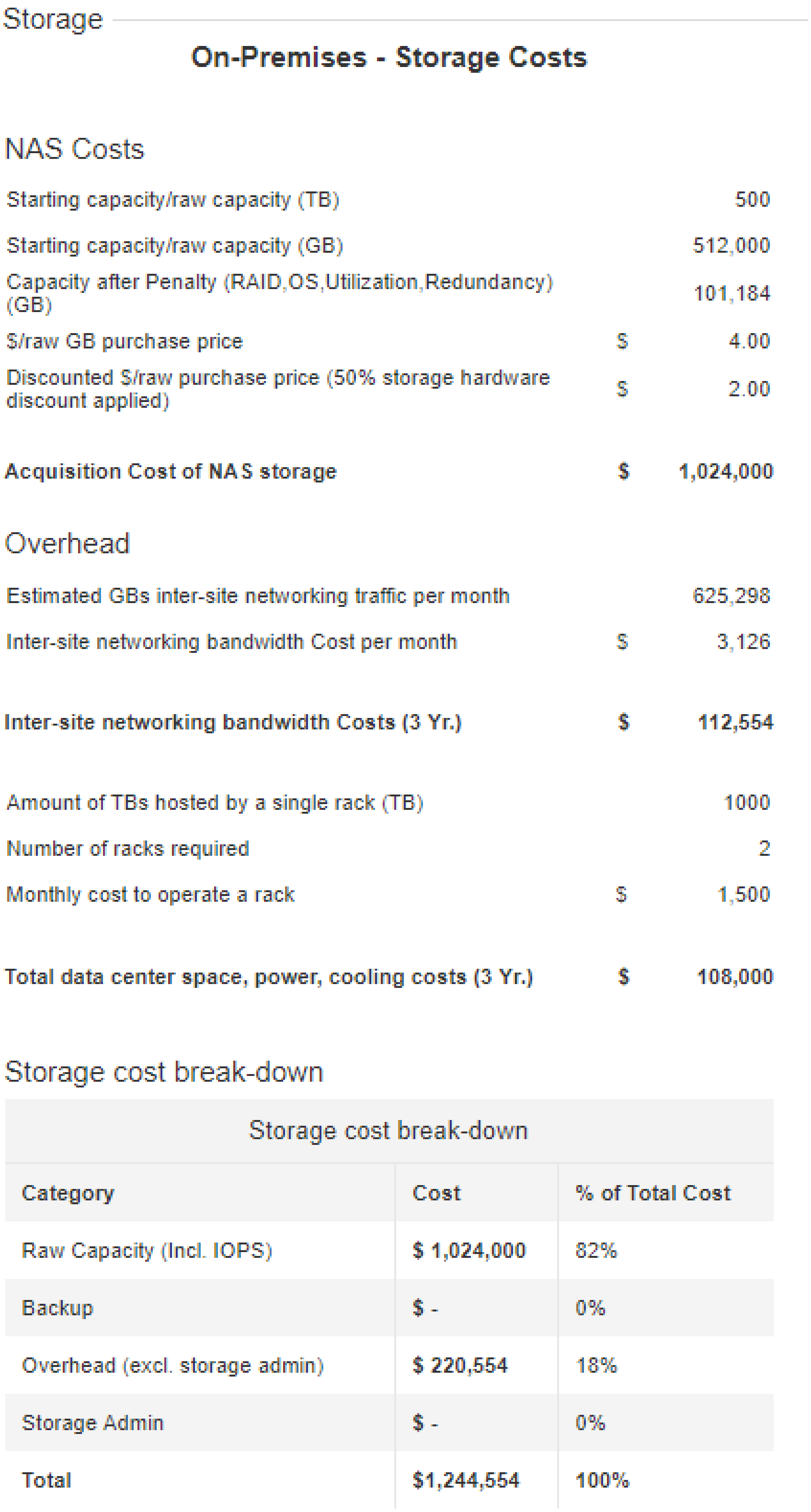
Virtualization Software Costs

Total number of VMware vSphere licenses required		4
VMware vSphere Enterprise Plus list price (unit per processor)	$	3,495
VMware vSphere discounted price (unit per processor)	$	2,621
Total VMware vSphere license costs	$	10,485
Total VMware vSphere support(SnS) costs	$	7,864
Total VMware vSphere license + support costs (3 Yrs.)	$	18,349
Total virtualization license and maintenance cost (3 Yrs.)	**$**	**18,349**



Facilities Costs (data center space, power and cooling)) - On-Premises

Total Power consumed by servers (kW)		1.5
Metered cost per kWH	$	0.12
Estimated power cost/month	$	129.60
Monthly cost to operate a rack	$	1,500.00
Total rack costs/month	$	1,500.00
Total monthly Facilities costs	$	1,629.60
Facilities costs - On-Premises (3 Yr.)	**$**	**58,666**



Storage

On-Premises - Storage Costs

NAS Costs

Starting capacity/raw capacity (TB)		500
Starting capacity/raw capacity (GB)		512,000
Capacity after Penalty (RAID,OS,Utilization,Redundancy) (GB)		101,184
$/raw GB purchase price	$	4.00
Discounted $/raw purchase price (50% storage hardware discount applied)	$	2.00
Acquisition Cost of NAS storage	**$**	**1,024,000**

Overhead

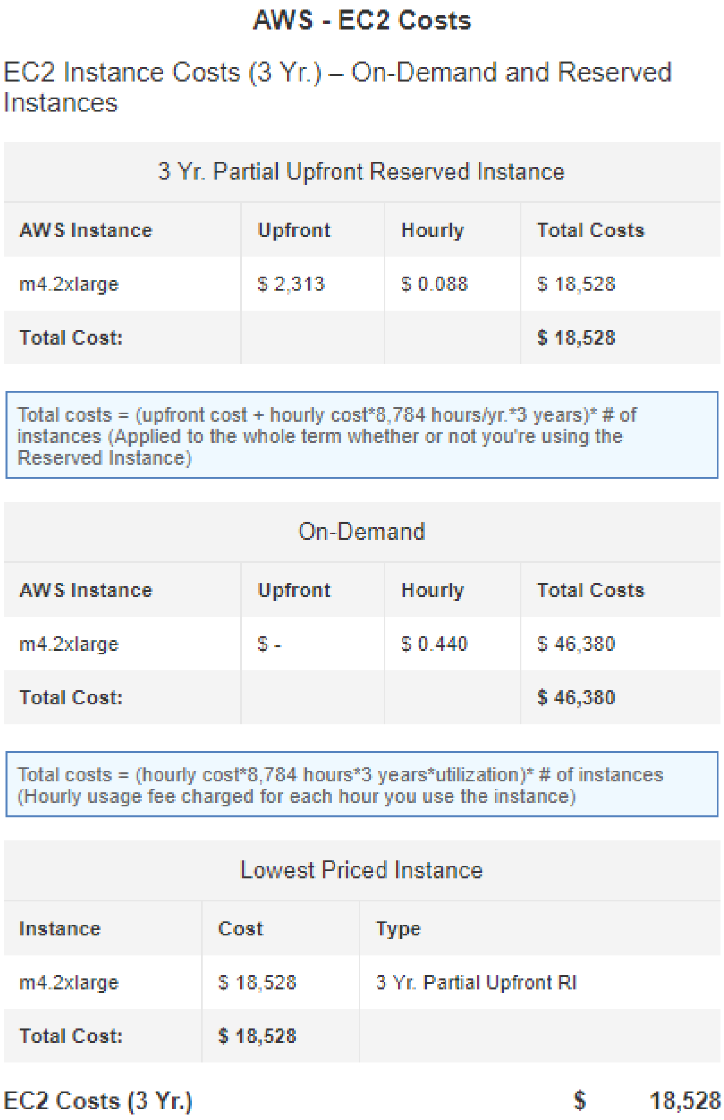
Estimated GBs inter-site networking traffic per month		625,298
Inter-site networking bandwidth Cost per month	$	3,126
Inter-site networking bandwidth Costs (3 Yr.)	**$**	**112,554**
Amount of TBs hosted by a single rack (TB)		1000
Number of racks required		2
Monthly cost to operate a rack	$	1,500
Total data center space, power, cooling costs (3 Yr.)	**$**	**108,000**

Storage cost break-down

Storage cost break-down

Category	Cost	% of Total Cost
Raw Capacity (Incl. IOPS)	**$ 1,024,000**	82%
Backup	**$ -**	0%
Overhead (excl. storage admin)	**$ 220,554**	18%
Storage Admin	**$ -**	0%
Total	**$1,244,554**	**100%**

Public Cloud - AWS Server Costs:



AWS - EC2 Costs

EC2 Instance Costs (3 Yr.) – On-Demand and Reserved Instances

3 Yr. Partial Upfront Reserved Instance			
AWS Instance	**Upfront**	**Hourly**	**Total Costs**
m4.2xlarge	$ 2,313	$ 0.088	$ 18,528
Total Cost:			**$ 18,528**

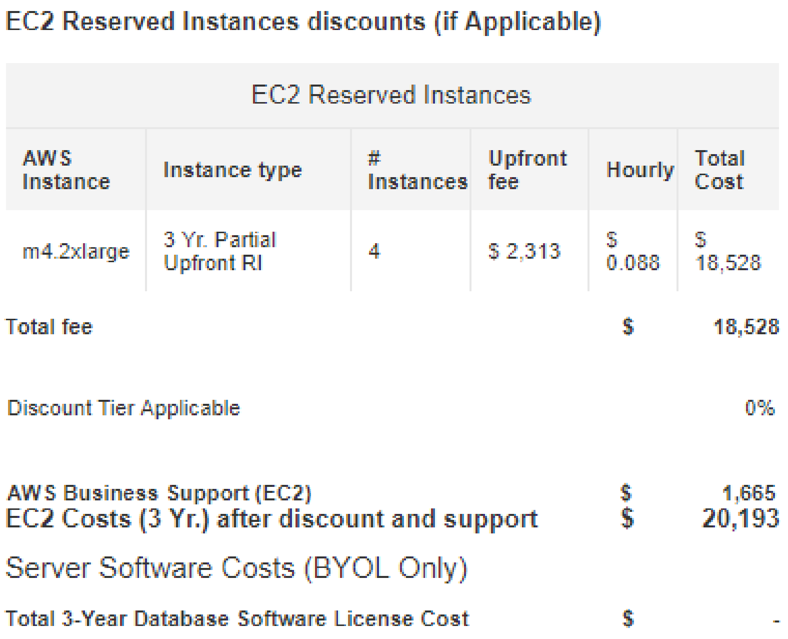
Total costs = (upfront cost + hourly cost*8,784 hours/yr.*3 years)* # of instances (Applied to the whole term whether or not you're using the Reserved Instance)

On-Demand			
AWS Instance	**Upfront**	**Hourly**	**Total Costs**
m4.2xlarge	$ -	$ 0.440	$ 46,380
Total Cost:			**$ 46,380**

Total costs = (hourly cost*8,784 hours*3 years*utilization)* # of instances (Hourly usage fee charged for each hour you use the instance)

Lowest Priced Instance		
Instance	**Cost**	**Type**
m4.2xlarge	$ 18,528	3 Yr. Partial Upfront RI
Total Cost:	**$ 18,528**	

EC2 Costs (3 Yr.) $ 18,528



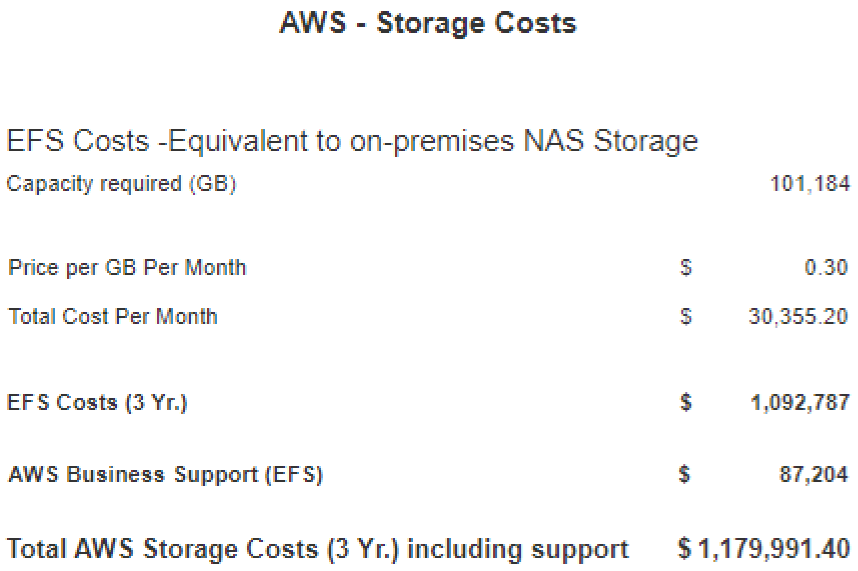
EC2 Reserved Instances discounts (if Applicable)

EC2 Reserved Instances					
AWS Instance	Instance type	# Instances	Upfront fee	Hourly	Total Cost
m4.2xlarge	3 Yr. Partial Upfront RI	4	$ 2,313	$ 0.088	$ 18,528

Total fee	**$**	**18,528**
Discount Tier Applicable		0%
AWS Business Support (EC2)	**$**	**1,665**
EC2 Costs (3 Yr.) after discount and support	**$**	**20,193**

Server Software Costs (BYOL Only)

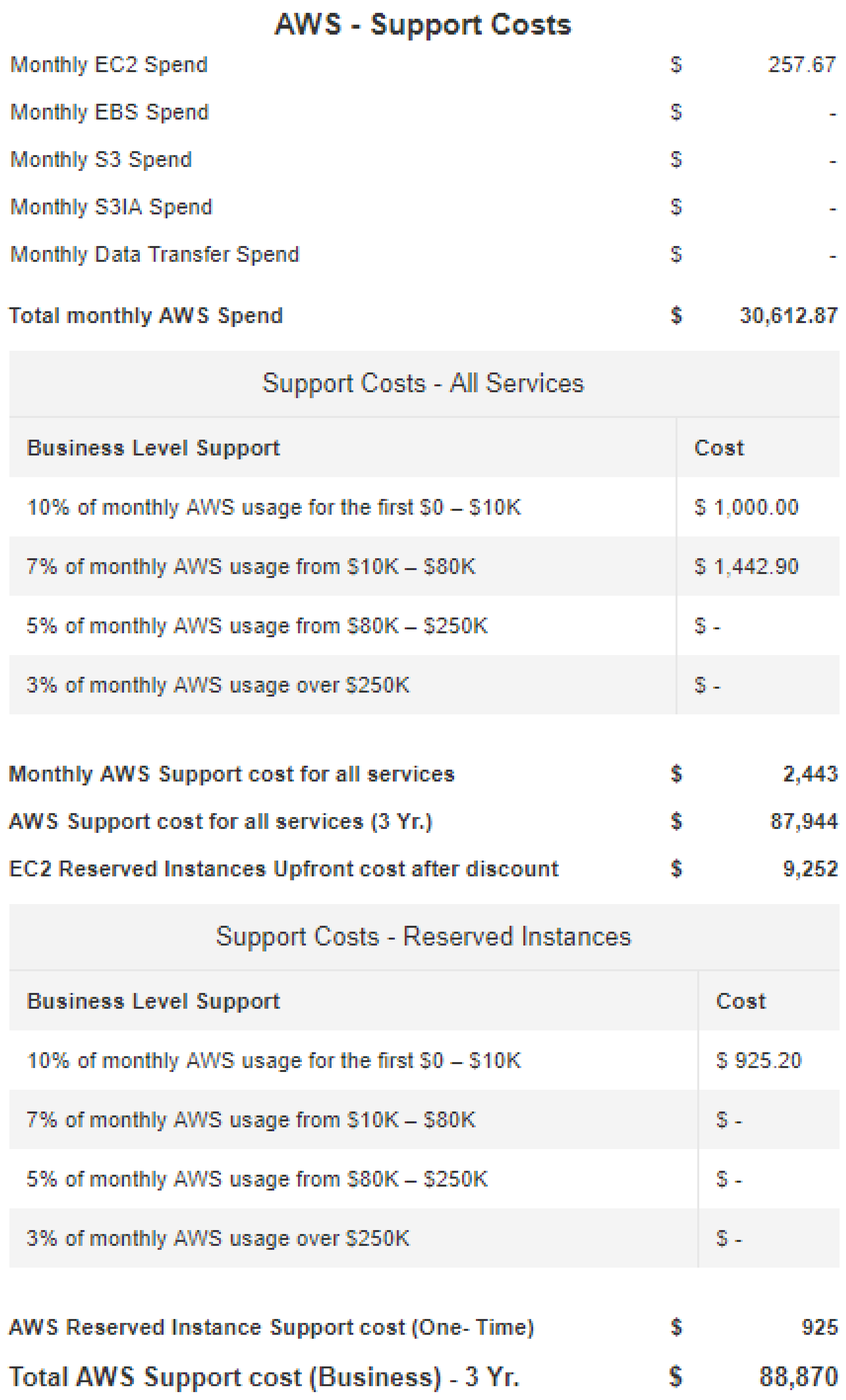
Total 3-Year Database Software License Cost	**$**	-



AWS - Storage Costs

EFS Costs -Equivalent to on-premises NAS Storage

Capacity required (GB)		101,184
Price per GB Per Month	$	0.30
Total Cost Per Month	$	30,355.20
EFS Costs (3 Yr.)	**$**	**1,092,787**
AWS Business Support (EFS)	**$**	**87,204**
Total AWS Storage Costs (3 Yr.) including support	**$**	**1,179,991.40**



AWS - Support Costs

Monthly EC2 Spend	$	257.67
Monthly EBS Spend	$	-
Monthly S3 Spend	$	-
Monthly S3IA Spend	$	-
Monthly Data Transfer Spend	$	-
Total monthly AWS Spend	**$**	**30,612.87**

Support Costs - All Services

Business Level Support	Cost
10% of monthly AWS usage for the first $0 – $10K	$ 1,000.00
7% of monthly AWS usage from $10K – $80K	$ 1,442.90
5% of monthly AWS usage from $80K – $250K	$ -
3% of monthly AWS usage over $250K	$ -

Monthly AWS Support cost for all services	**$**	**2,443**
AWS Support cost for all services (3 Yr.)	**$**	**87,944**
EC2 Reserved Instances Upfront cost after discount	**$**	**9,252**

Support Costs - Reserved Instances

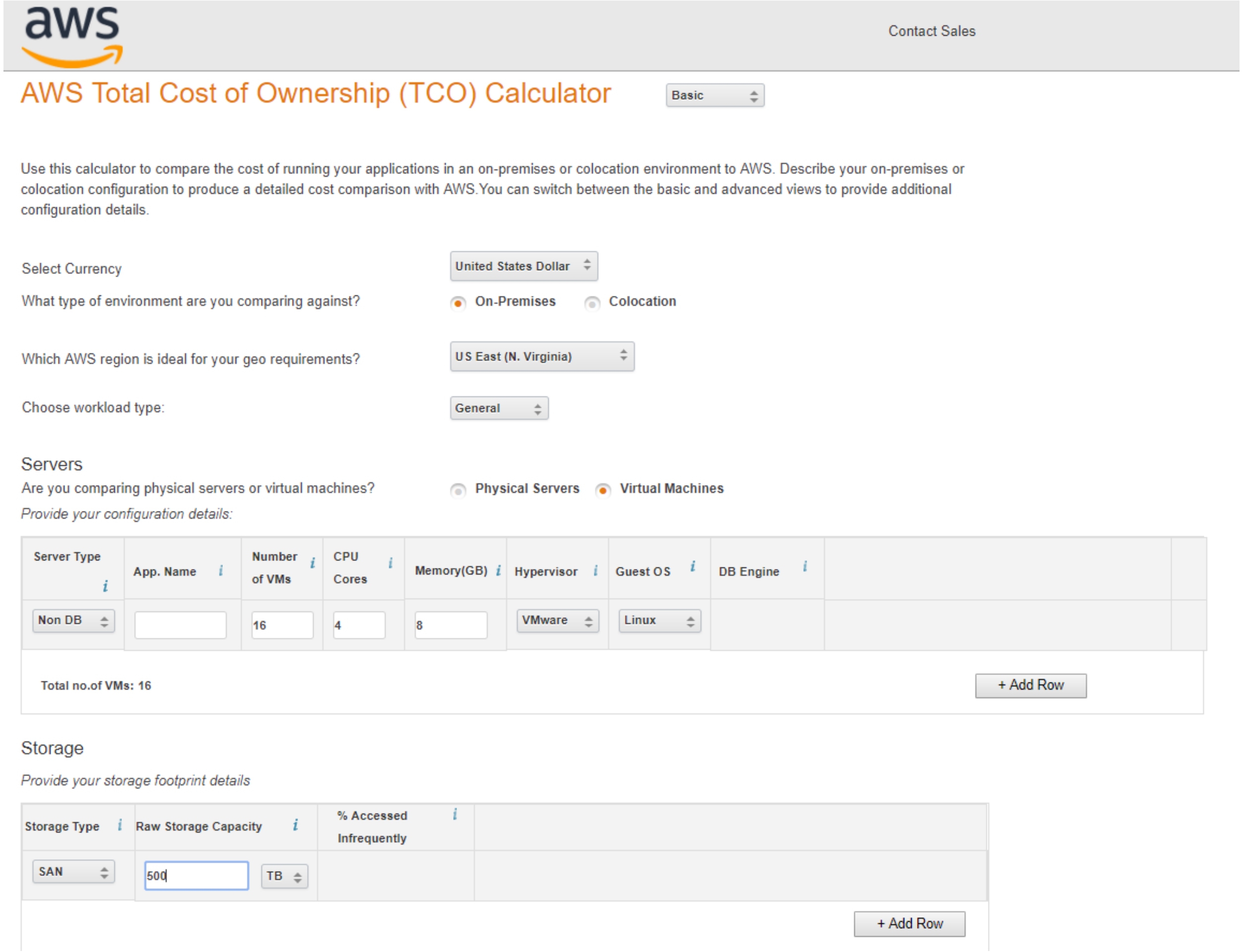
Business Level Support	Cost
10% of monthly AWS usage for the first $0 – $10K	$ 925.20
7% of monthly AWS usage from $10K – $80K	$ -
5% of monthly AWS usage from $80K – $250K	$ -
3% of monthly AWS usage over $250K	$ -

AWS Reserved Instance Support cost (One- Time)	**$**	**925**
Total AWS Support cost (Business) - 3 Yr.	$	88,870

Total Cost after 3 Years for the Private Cloud - Configuration 1 = $ 1,325,011.41
Total Cost after 3 Years for the Public Cloud – AWS = $ 1,200,185
Cost Savings = $ 124,826.41

Cost Analysis and Comparison (Configuration 2: Smaller Machines) Vs. (AWS)

Input Parameters:



AWS Total Cost of Ownership (TCO) Calculator

Basic

Use this calculator to compare the cost of running your applications in an on-premises or colocation environment to AWS. Describe your on-premises or colocation configuration to produce a detailed cost comparison with AWS.You can switch between the basic and advanced views to provide additional configuration details.

Select Currency: United States Dollar

What type of environment are you comparing against? ● On-Premises ○ Colocation

Which AWS region is ideal for your geo requirements? US East (N. Virginia)

Choose workload type: General

Servers

Are you comparing physical servers or virtual machines? ○ Physical Servers ● Virtual Machines

Provide your configuration details:

Server Type	App. Name	Number of VMs	CPU Cores	Memory(GB)	Hypervisor	Guest OS	DB Engine
Non DB		16	4	8	VMware	Linux	

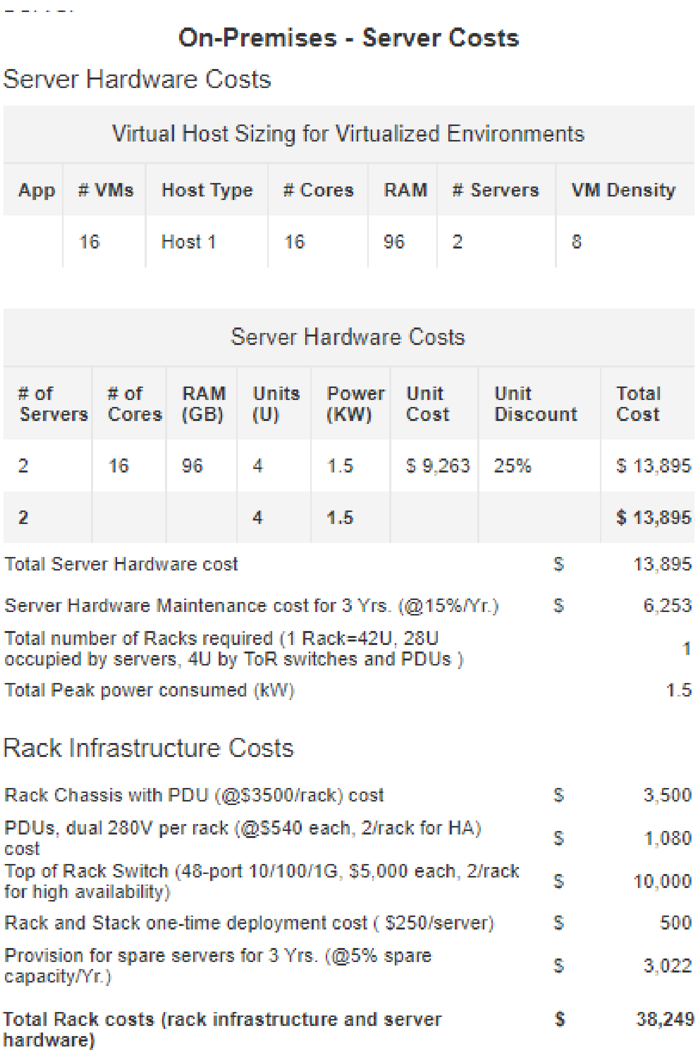
Total no.of VMs: 16

Storage

Provide your storage footprint details

Storage Type	Raw Storage Capacity		% Accessed Infrequently
SAN	500	TB	

Private Cloud – Configuration 2 Costs:



On-Premises - Server Costs

Server Hardware Costs

Virtual Host Sizing for Virtualized Environments

App	# VMs	Host Type	# Cores	RAM	# Servers	VM Density
	16	Host 1	16	96	2	8

Server Hardware Costs

# of Servers	# of Cores	RAM (GB)	Units (U)	Power (KW)	Unit Cost	Unit Discount	Total Cost
2	16	96	4	1.5	$ 9,263	25%	$ 13,895
2			**4**	**1.5**			**$ 13,895**

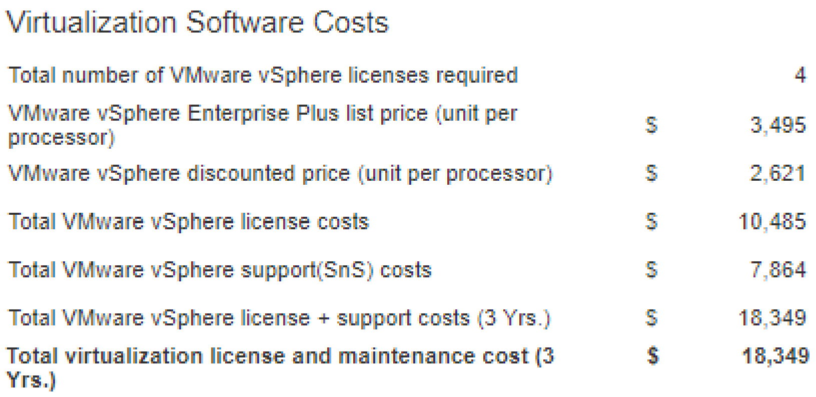
Total Server Hardware cost	$	13,895
Server Hardware Maintenance cost for 3 Yrs. (@15%/Yr.)	$	6,253
Total number of Racks required (1 Rack=42U, 28U occupied by servers, 4U by ToR switches and PDUs)		1
Total Peak power consumed (kW)		1.5

Rack Infrastructure Costs

Rack Chassis with PDU (@$3500/rack) cost	$	3,500
PDUs, dual 280V per rack (@$540 each, 2/rack for HA) cost	$	1,080
Top of Rack Switch (48-port 10/100/1G, $5,000 each, 2/rack for high availability)	$	10,000
Rack and Stack one-time deployment cost ($250/server)	$	500
Provision for spare servers for 3 Yrs. (@5% spare capacity/Yr.)	$	3,022
Total Rack costs (rack infrastructure and server hardware)	**$**	**38,249**

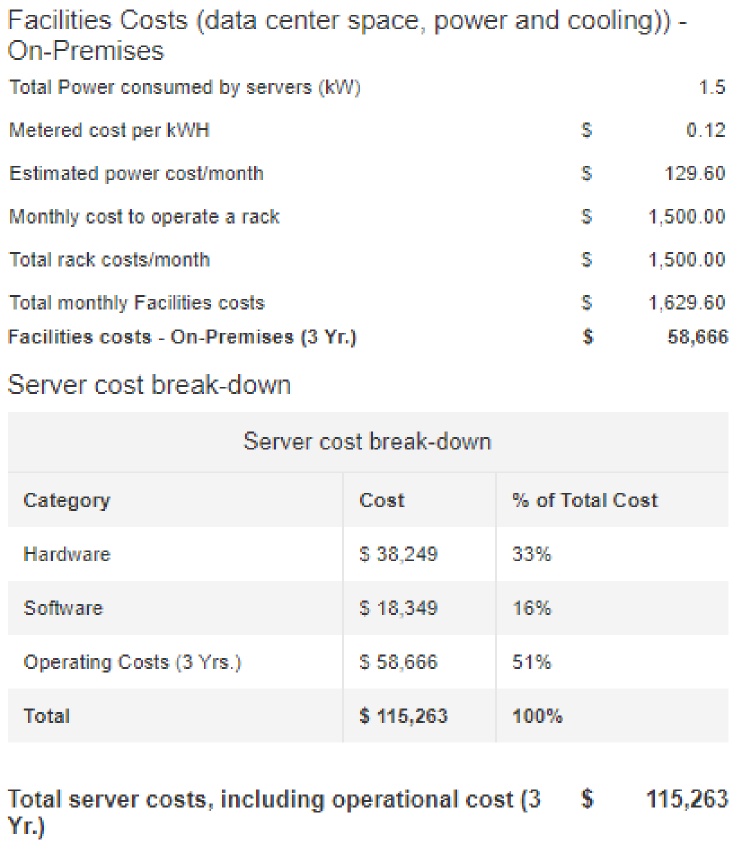
Total Rack Costs Adjusted for San Francisco or New York, i.e. 80% More = $ 68848.2
Total Rack Costs After 95% Discount = $ 3442.41

Also Networking Costs are not taken into consideration.



Virtualization Software Costs

Total number of VMware vSphere licenses required		4
VMware vSphere Enterprise Plus list price (unit per processor)	$	3,495
VMware vSphere discounted price (unit per processor)	$	2,621
Total VMware vSphere license costs	$	10,485
Total VMware vSphere support(SnS) costs	$	7,864
Total VMware vSphere license + support costs (3 Yrs.)	$	18,349
Total virtualization license and maintenance cost (3 Yrs.)	**$**	**18,349**



Facilities Costs (data center space, power and cooling)) - On-Premises

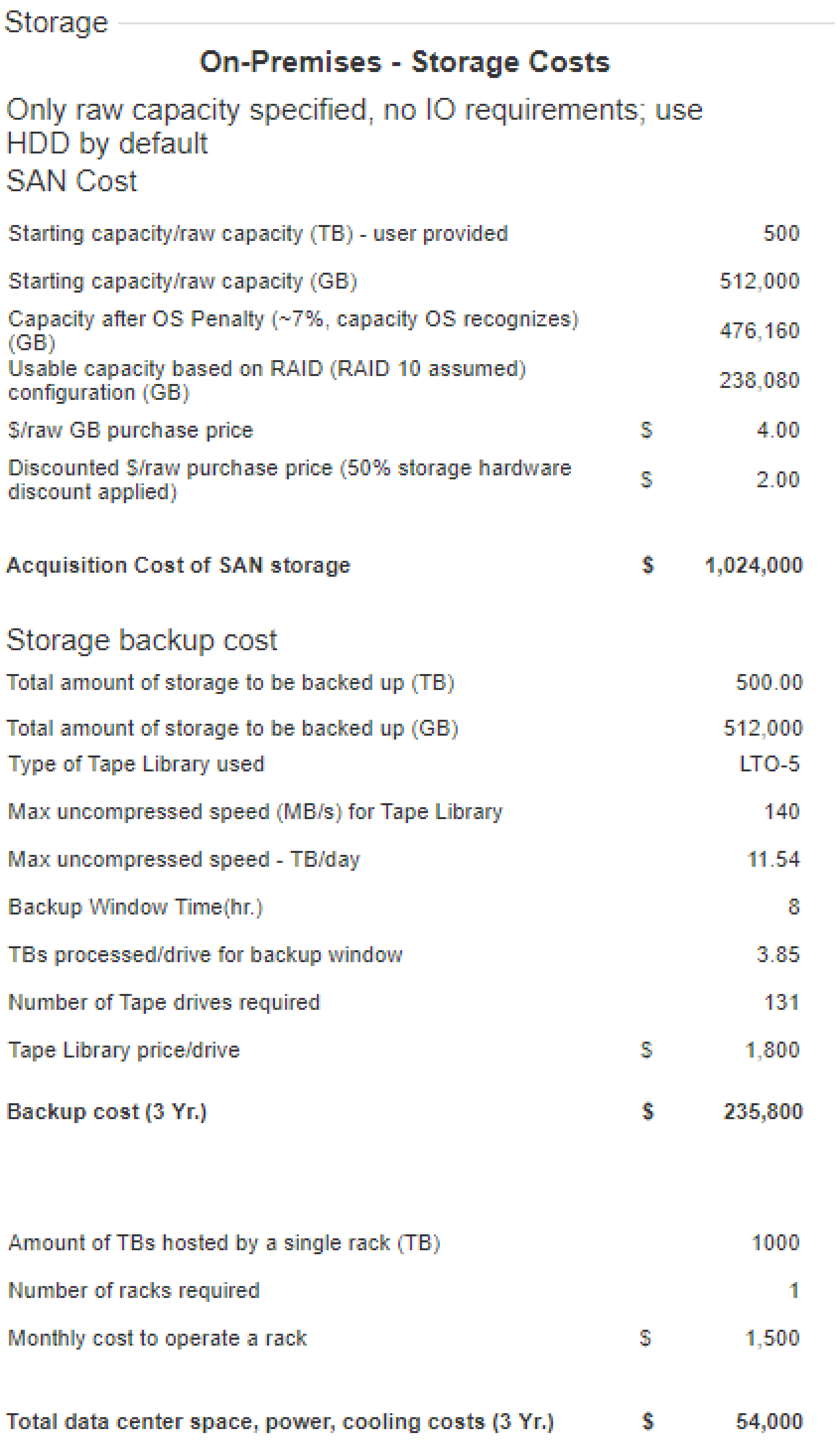
Total Power consumed by servers (kW)		1.5
Metered cost per kWH	$	0.12
Estimated power cost/month	$	129.60
Monthly cost to operate a rack	$	1,500.00
Total rack costs/month	$	1,500.00
Total monthly Facilities costs	$	1,629.60
Facilities costs - On-Premises (3 Yr.)	**$**	**58,666**

Server cost break-down

Server cost break-down

Category	Cost	% of Total Cost
Hardware	$ 38,249	33%
Software	$ 18,349	16%
Operating Costs (3 Yrs.)	$ 58,666	51%
Total	**$ 115,263**	**100%**

Total server costs, including operational cost (3 Yr.)	**$**	**115,263**



Storage

On-Premises - Storage Costs

Only raw capacity specified, no IO requirements; use HDD by default

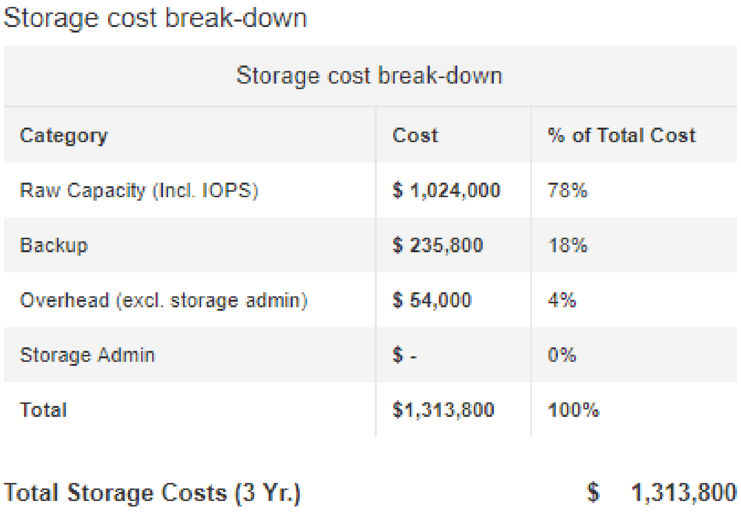
SAN Cost

Starting capacity/raw capacity (TB) - user provided		500
Starting capacity/raw capacity (GB)		512,000
Capacity after OS Penalty (~7%, capacity OS recognizes) (GB)		476,160
Usable capacity based on RAID (RAID 10 assumed) configuration (GB)		238,080
$/raw GB purchase price	$	4.00
Discounted $/raw purchase price (50% storage hardware discount applied)	$	2.00
Acquisition Cost of SAN storage	**$**	**1,024,000**

Storage backup cost

Total amount of storage to be backed up (TB)		500.00
Total amount of storage to be backed up (GB)		512,000
Type of Tape Library used		LTO-5
Max uncompressed speed (MB/s) for Tape Library		140
Max uncompressed speed - TB/day		11.54
Backup Window Time(hr.)		8
TBs processed/drive for backup window		3.85
Number of Tape drives required		131
Tape Library price/drive	$	1,800
Backup cost (3 Yr.)	**$**	**235,800**

Amount of TBs hosted by a single rack (TB)		1000
Number of racks required		1
Monthly cost to operate a rack	$	1,500
Total data center space, power, cooling costs (3 Yr.)	**$**	**54,000**

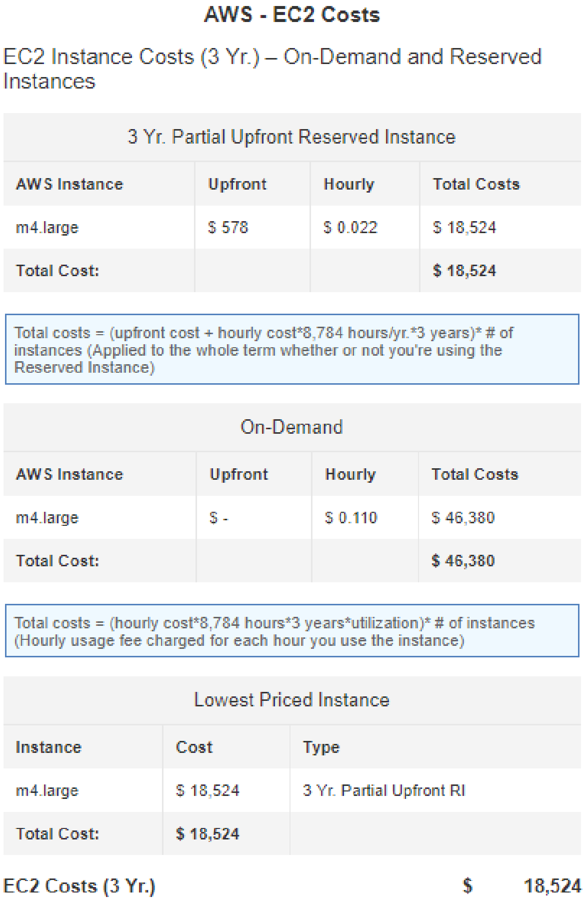


Storage cost break-down

Storage cost break-down		
Category	Cost	% of Total Cost
Raw Capacity (Incl. IOPS)	$ 1,024,000	78%
Backup	$ 235,800	18%
Overhead (excl. storage admin)	$ 54,000	4%
Storage Admin	$ -	0%
Total	$1,313,800	100%

Total Storage Costs (3 Yr.) $ 1,313,800

Public Cloud - AWS Server Costs:



AWS - EC2 Costs

EC2 Instance Costs (3 Yr.) – On-Demand and Reserved Instances

3 Yr. Partial Upfront Reserved Instance			
AWS Instance	Upfront	Hourly	Total Costs
m4.large	$ 578	$ 0.022	$ 18,524
Total Cost:			$ 18,524

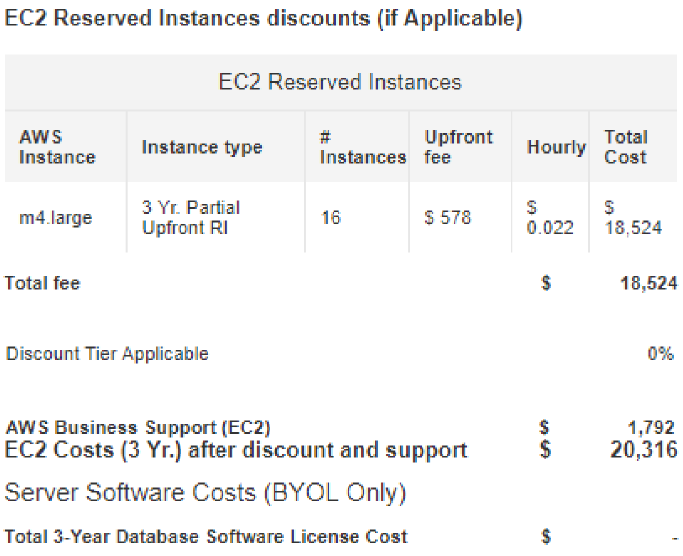
Total costs = (upfront cost + hourly cost*8,784 hours/yr.*3 years)* # of instances (Applied to the whole term whether or not you're using the Reserved Instance)

On-Demand			
AWS Instance	Upfront	Hourly	Total Costs
m4.large	$ -	$ 0.110	$ 46,380
Total Cost:			$ 46,380

Total costs = (hourly cost*8,784 hours*3 years*utilization)* # of instances (Hourly usage fee charged for each hour you use the instance)

Lowest Priced Instance		
Instance	Cost	Type
m4.large	$ 18,524	3 Yr. Partial Upfront RI
Total Cost:	$ 18,524	

EC2 Costs (3 Yr.) $ 18,524



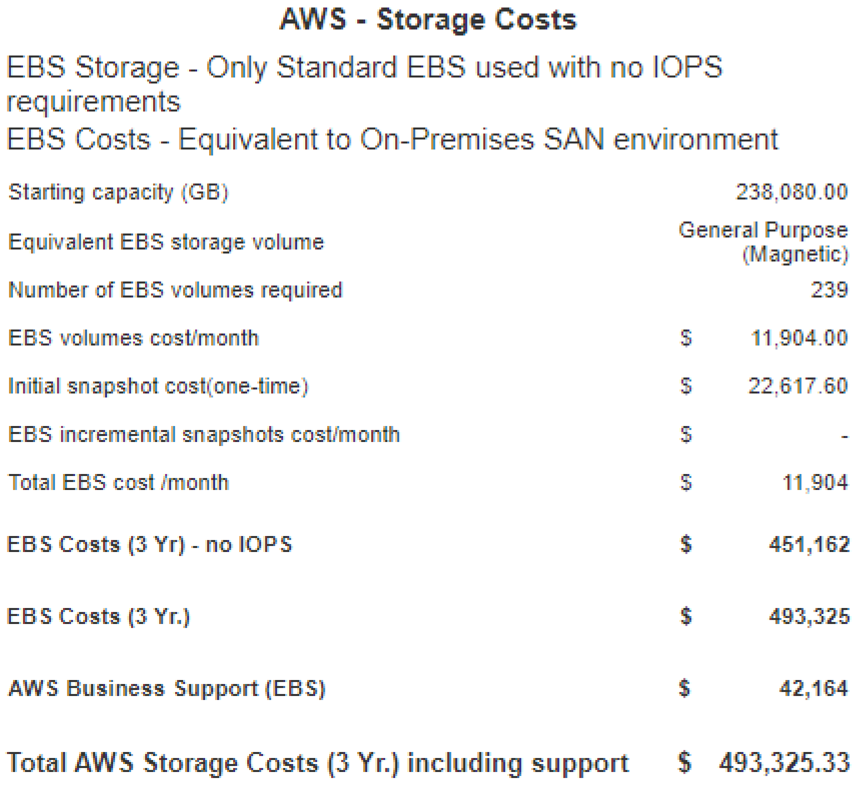
EC2 Reserved Instances discounts (if Applicable)

EC2 Reserved Instances					
AWS Instance	Instance type	# Instances	Upfront fee	Hourly	Total Cost
m4.large	3 Yr. Partial Upfront RI	16	$ 578	$ 0.022	$ 18,524

Total fee	**$**	**18,524**
Discount Tier Applicable		0%
AWS Business Support (EC2)	**$**	**1,792**
EC2 Costs (3 Yr.) after discount and support	**$**	**20,316**

Server Software Costs (BYOL Only)

Total 3-Year Database Software License Cost	**$**	-

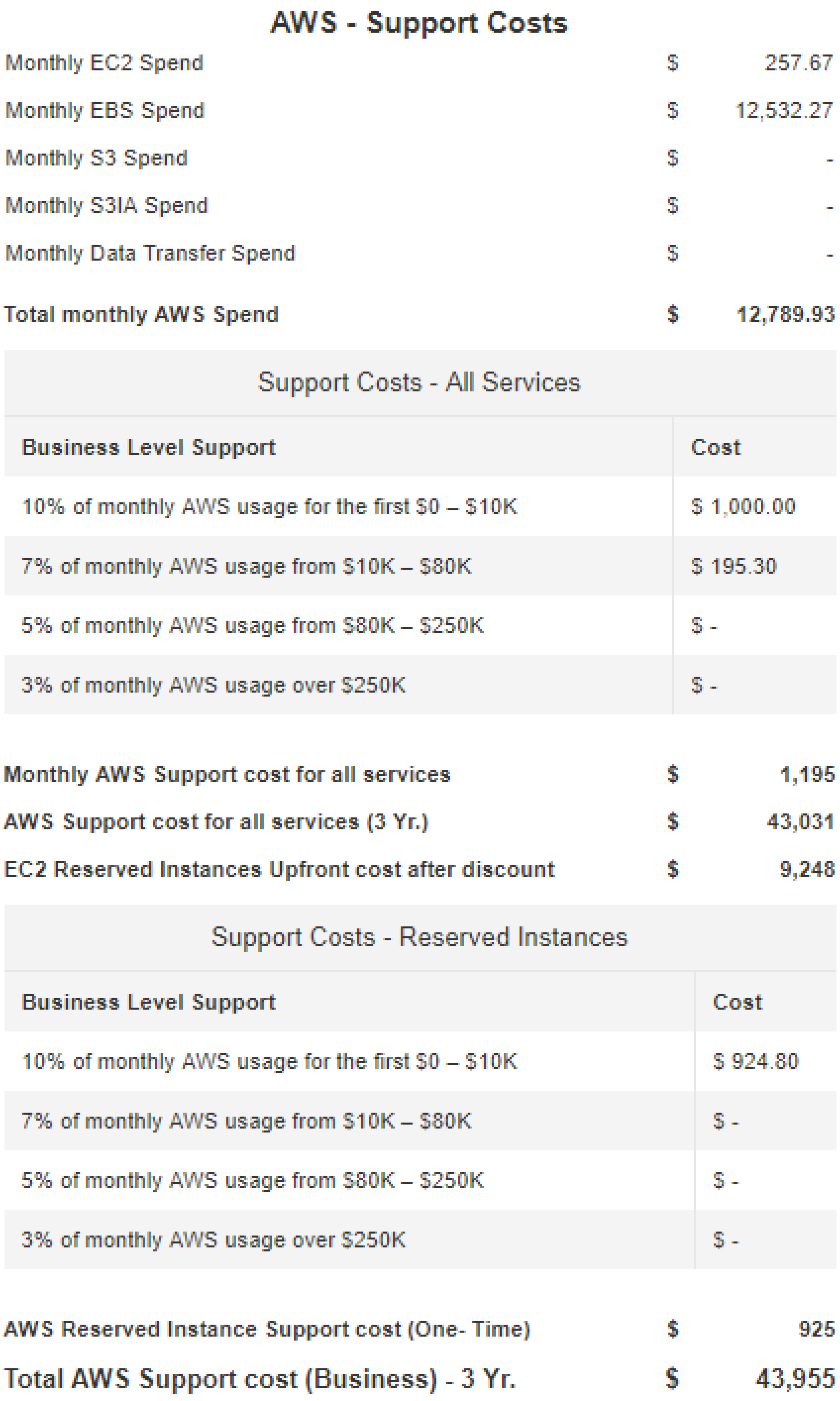


AWS - Storage Costs

EBS Storage - Only Standard EBS used with no IOPS requirements

EBS Costs - Equivalent to On-Premises SAN environment

Starting capacity (GB)		238,080.00
Equivalent EBS storage volume		General Purpose (Magnetic)
Number of EBS volumes required		239
EBS volumes cost/month	$	11,904.00
Initial snapshot cost(one-time)	$	22,617.60
EBS incremental snapshots cost/month	$	-
Total EBS cost /month	$	11,904
EBS Costs (3 Yr) - no IOPS	**$**	**451,162**
EBS Costs (3 Yr.)	**$**	**493,325**
AWS Business Support (EBS)	**$**	**42,164**
Total AWS Storage Costs (3 Yr.) including support	**$**	**493,325.33**



AWS - Support Costs

Monthly EC2 Spend	$	257.67
Monthly EBS Spend	$	12,532.27
Monthly S3 Spend	$	-
Monthly S3IA Spend	$	-
Monthly Data Transfer Spend	$	-
Total monthly AWS Spend	**$**	**12,789.93**

Support Costs - All Services

Business Level Support	Cost
10% of monthly AWS usage for the first $0 – $10K	$ 1,000.00
7% of monthly AWS usage from $10K – $80K	$ 195.30
5% of monthly AWS usage from $80K – $250K	$ -
3% of monthly AWS usage over $250K	$ -

Monthly AWS Support cost for all services	**$**	**1,195**
AWS Support cost for all services (3 Yr.)	**$**	**43,031**
EC2 Reserved Instances Upfront cost after discount	**$**	**9,248**

Support Costs - Reserved Instances

Business Level Support	Cost
10% of monthly AWS usage for the first $0 – $10K	$ 924.80
7% of monthly AWS usage from $10K – $80K	$ -
5% of monthly AWS usage from $80K – $250K	$ -
3% of monthly AWS usage over $250K	$ -

AWS Reserved Instance Support cost (One- Time)	**$**	**925**
Total AWS Support cost (Business) - 3 Yr.	**$**	**43,955**

Total Cost after 3 Years for the Private Cloud - Configuration 2 = $ 1,394,257.41
Total Cost after 3 Years for the Public Cloud – AWS = $ 513,641
Cost Savings = 880,616.41

Conclusion and Recommendation

From the Analysis and Cost Breakdown, it is clear that the Public Cloud AWS is a cheaper option than the TAMU-based Private Cloud to host the application.

- For Configuration 1, around 10% of cost (~$125,000) can be saved from using AWS for the three-year period.
- For Configuration 2, around 63% of cost (~$880000) can be saved from using AWS for the three-year period.

Clearly, a lot of cost can be saved by choosing to run the application and the video/image processing algorithms on AWS than on the TAMU-based Private Cloud.

Thus, my recommendation for the application is to use the Public Cloud (AWS).

However, before any decision all the pros and cons of using a Public Cloud over a Private Cloud must be considered.

Here is a List of the General Pros and Cons of using a Public Cloud Vs. Private Cloud

Public Cloud:
Pros:

- Pay-as-you go Model/Utility Computing.
- Cost Effectiveness
- Increased Reliability.

Cons:

- Network Overhead.
- Security Concerns.

Private Cloud:
Pros:

- Better Security.
- Greater Control over Servers and other Infrastructure.

Cons:

- Hard to Scale.
- At times costlier because of rack costs, operational and administration costs, etc.

Detailed Report - 2

Time Tracking Information:

Approximately 2 hours taken to accomplish the task which included:

- Around 1 hour going through the documentation, software installation & deployment on Heroku.
- Around 1 hour in comprehensively preparing the report.

Task Specifications:

The client wants an opinion on the ease-of-use of the Heroku cloud platform (heroku.com).

Applications can be deployed on Heroku for free. The client has requested to go through the "Getting Started on Heroku" material (devcenter.heroku.com) and to follow the instructions to deploy an app on Heroku's platform.

Heroku, being a PaaS (Platform-as-a-Service) provider offers its customers the ability to deploy applications using many runtime frameworks that they support i.e. Java, Python, Go, Node.js, Ruby & PHP. New versions of code can easily be deployed on Heroku's Cloud after developing and testing the application functionality in the local development machines.

The task is to follow the "Getting started on Heroku" documentation to get an application deployed on Heroku with any of the available runtime frameworks.

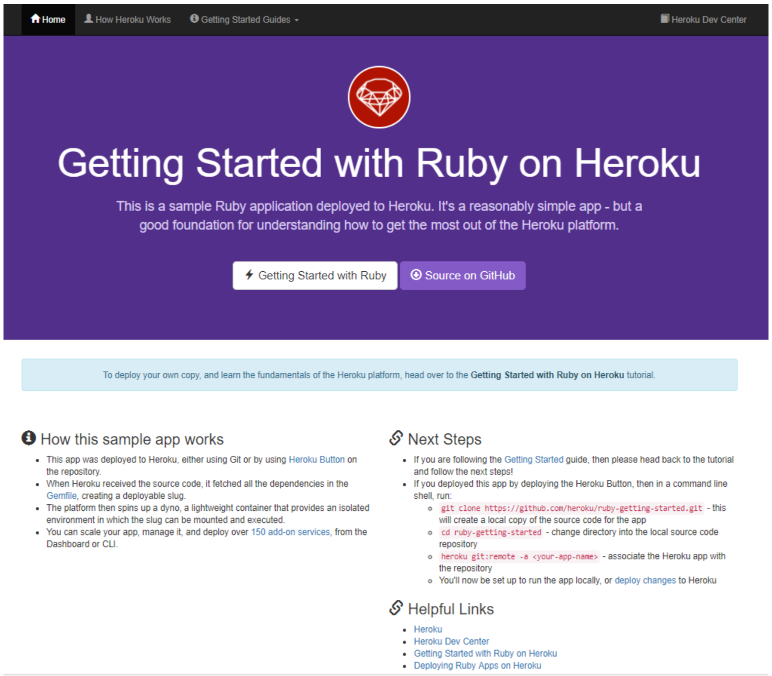
Sample Heroku Application:

I used the Ruby Platform to Deploy the Application.

Heroku Link: https://sheltered-inlet-44880.herokuapp.com/

Personal Github Link: https://github.tamu.edu/Sameer-TAMU/689-18-a

Screenshot of the Application:



Getting Started with Ruby on Heroku

This is a sample Ruby application deployed to Heroku. It's a reasonably simple app - but a good foundation for understanding how to get the most out of the Heroku platform.

To deploy your own copy, and learn the fundamentals of the Heroku platform, head over to the **Getting Started with Ruby on Heroku** tutorial.

How this sample app works

- This app was deployed to Heroku, either using Git or by using Heroku Button on the repository.
- When Heroku received the source code, it fetched all the dependencies in the Gemfile, creating a deployable slug.
- The platform then spins up a dyno, a lightweight container that provides an isolated environment in which the slug can be mounted and executed.
- You can scale your app, manage it, and deploy over 150 add-on services, from the Dashboard or CLI.

Next Steps

- If you are following the Getting Started guide, then please head back to the tutorial and follow the next steps!
- If you deployed this app by deploying the Heroku Button, then in a command line shell, run:
  - `git clone https://github.com/heroku/ruby-getting-started.git` - this will create a local copy of the source code for the app
  - `cd ruby-getting-started` - change directory into the local source code repository
  - `heroku git:remote -a <your-app-name>` - associate the Heroku app with the repository
  - You'll now be set up to run the app locally, or deploy changes to Heroku

Helpful Links

- Heroku
- Heroku Dev Center
- Getting Started with Ruby on Heroku
- Deploying Ruby Apps on Heroku

Problems Faced:

I did not encounter any issues in setting up Heroku or in deploying my application as I have already used Heroku many times to deploy some applications in the past.

Some of the Past Applications Deployed on Heroku:

https://arcane-mountain-66382.herokuapp.com/
https://mysterious-citadel-25783.herokuapp.com/

Usefulness of the Heroku Exercises:

- Heroku lets you deploy, run and manage applications written in Ruby, Node.js, Java, Python, Clojure, Scala, Go and PHP, etc. An application is a collection of source code written in one of these languages, perhaps a framework, and some dependency description that instructs a build system as to which additional dependencies are needed in order to build and run the application. Dependency mechanisms vary across languages: In Ruby you use a Gemfile.
- The source code for your application, together with the dependency file, should provide enough information for the Heroku platform to build your application, to produce something that can be executed.
- You don't need to make many changes to an application in order to run it on Heroku. One requirement is informing the platform as to which parts of your application are runnable. If you're using some established framework, Heroku can figure it out. For example, in Ruby on Rails, it's typically rails server.
- Git is a powerful, distributed version control system that many developers use to manage and version source code. The Heroku platform uses Git as the primary means for deploying applications. When you create an application on Heroku, it associates a new Git remote, typically named heroku, with the local Git repository for your application.
- The source code for your application, together with the fetched dependencies and output of the build phase such as generated assets or compiled code, as well as the language and framework, are assembled into a slug. Heroku executes applications by running a command you specified in the Procfile, on a dyno that's been preloaded with your prepared slug.
- An application's configuration is everything that is likely to vary between environments (staging, production, developer environments, etc.). This includes backing services such as databases, credentials, or environment variables that provide some specific information to your application. Heroku lets you run your application with a customizable configuration - the configuration sits outside of your application code and can be changed independently of it.
- All releases are automatically persisted in an append-only ledger, making managing your application, and different releases, a cinch. As Heroku contains a store of the previous releases of your application, it's very easy to rollback and deploy a previous release.
- A random selection algorithm is used for HTTP request load balancing across web dynos - and this routing handles both HTTP and HTTPS traffic. It also supports multiple simultaneous connections, as well as timeout handling.

My Impressions on Heroku's Pros and Cons:

Pros:

- Getting started on Heroku is ridiculously easy. You can have an app deployed there in just a minute or two, all from doing nothing more than adding a git remote and pushing to it.
- Heroku allows rapid prototyping and rapid deployment for apps where the user does not need to specify lower level configuration and system configuration through PaaS.
- Heroku's Cedar stack is quite good. You can run whatever server software you want on it.
- The flexibility Heroku provides allows you to run nearly anything in their managed environment, and then scale it however you find appropriate. With correct separation of concerns, this provides you an enormous amount of control over how your application is deployed and how you can respond to traffic influxes.
- Heroku's Postgres instances also do automatic backups and have always been available whenever I've needed them.
- I have never got paged from Heroku Server and never found an unresponsive page. Heroku has never, ever had a stability problem like that.
- It is easily extensible to add powerful features such as memcache.
- It has very little downtime and platform locking.
- Heroku is the cheapest option for a low traffic site.
- You don't have to add your credit card for payment at early stage.
- Database integration is pretty simple with PostgreSQL.

Cons:

- The biggest drawback is that it does not allow developers to choose system configuration and the underlying infrastructure. So, the underlying infrastructure of the hosted applications may not be the most effective and it does not give developers the flexibility to choose infrastructure.
- You have to pay very high once you decide to handle more traffic.
- You have to manually scale your application on high traffic.
- You can't login to your server via SSH.
- Heroku Addons such as RedisToGo and the Postgres database options are seriously expensive.
- At first instance, deploying to Heroku is easy and fast. But ongoing deployments to the servers take time as compiling a slug takes a while. Also, after slug compilation is complete, the dynos must restart which makes the application go completely offline.

References:

- https://devcenter.heroku.com/
- https://devcenter.heroku.com/articles/how-heroku-works
- https://aws.amazon.com/tco-calculator/
- https://www.digitalocean.com/community/tutorials/object-storage-vs-block-storage-services
- https://www.bankrate.com/calculators/index-of-checking-savings-calculators.aspx
- http://searchcloudcomputing.techtarget.com/feature/Networking-costs-swell-pay-per-use-public-cloud-bills
- https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/
- https://www.lifewire.com/san-vs-nas-818005
- https://aws.amazon.com/blogs/aws/cloud-computing-server-utilization-the-environment/
- https://techcrunch.com/2011/11/26/pinterest-viral/
- https://en.wikipedia.org/wiki/Pinterest
- https://www.comscore.com/
- http://www.ragingwire.com/data-center-tco-comparison-calculator
- https://www.referralcandy.com/blog/pinterest-marketing-strategy/
- http://growthdevil.com/pinterest-get-so-big/
- https://www.supermicro.com/products/nfo/gpu.cfm?gclid=EAIaIQobChMI6eqklvGB2gIVlrjACh1-OAakEAAYBCAAEgLmMvD_BwE
- https://www.cia.gov/library/publications/the-world-factbook/
- https://www.opendatanetwork.com/
- http://www.managedsolution.com/case-study-pinterest-uses-aws-manage-high-performance-social-application-stores-8-billion-objects-400-terabytes-data-aws-cloud/
- http://highscalability.com/blog/2012/5/21/pinterest-architecture-update-18-million-visitors-10x-growth.html