The Total Cost of (Non) Ownership of Web Applications in the Cloud

Jinesh Varia

August 2012

(Please consult http://aws.amazon.com/whitepapers/ for the latest version of this paper)

Abstract

Weighing the financial considerations of owning and operating a data center facility versus employing a cloud infrastructure requires detailed and careful analysis. In practice, it is not as simple as just measuring potential hardware expense alongside utility pricing for compute and storage resources. The Total Cost of Ownership (TCO) is often the financial metric used to estimate and compare direct and indirect costs of a product or a service. Given the large differences between the two models, it is challenging to perform accurate apples-to-apples cost comparisons between on-premises data centers and cloud infrastructure that is offered as a service. In this whitepaper, we explain the economic benefits of deploying a web application in the Amazon Web Services (AWS) cloud over deploying an equivalent web application hosted in an on-premises data center.

The goal of this whitepaper is to help you understand the different cost factors involved when you deploy and manage a scalable web application hosted on-premises versus when you deploy an equivalent web application in the cloud. We walk through three example scenarios: a corporate website (a steady-state web application), a sports event website (a spiky web application), and a social web application (an uncertain, unpredictable web application). Our comparison highlights the total costs over a 3-year period. We compare the total costs of running these web applications on-premises with the total cost of running these in the AWS cloud, reviewing a variety of different AWS purchasing options. In each scenario, we will highlight the purchasing option with the highest cost savings.

Our analysis shows that AWS offers significant cost savings (up to 80%) over the equivalent on-premises option in each scenario. More importantly, you will see that AWS not only helps you lower your costs and maximize your savings but also encourages innovation in your company by lowering the cost of experimentation. We state our assumptions in each option and recommend that you adjust these assumptions based on your own research or quotes from your hardware vendors.

AWS Pricing Philosophy

While the number and types of services offered by AWS has increased dramatically, our philosophy on pricing has not changed: you pay only for the resources that you use. The key tenets of the AWS pricing philosophy are:

- **Pay as you go**. No minimum commitments or long-term contracts required. You replace your upfront capital expense with low variable cost and pay only for what you use. There is no need to pay upfront for excess capacity or get penalized for under-planning. For compute resources, you pay on an hourly basis from the time you launch a resource until the time you terminate it. For data storage and transfer, you pay on a per gigabyte basis. We charge based on the underlying infrastructure and services that you consume. You can turn off your cloud resources and stop paying for them when you don't need them.

- **Pay less when you reserve**. For certain products, you can invest in reserved capacity. In that case, you pay a low upfront fee and get a significantly discounted hourly rate, which results in overall savings between 42% and 71% (depending on the type of instance you reserve) over equivalent on-demand capacity.

- **Pay even less per unit by using more.** You save more as you grow bigger. For storage and data transfer, pricing is tiered. The more you use, the less you pay per gigabyte. For compute, you get volume discounts up to 20% when you reserve more.

- **Pay even less as AWS grows**. Most importantly, we are constantly focused on reducing our data center hardware costs, improving our operational efficiencies, lowering our power consumption, and generally lowering the cost of doing business. These optimizations and AWS's substantial and growing economies of scale result in passing savings back to you in the form of lower pricing. In the past six years, AWS has lowered pricing on 20 different occasions.

- **Custom pricing**. What if none of our pricing models work for your project? Custom pricing is available for high volume projects with unique requirements. For assistance, contact us to speak with a sales representative.

Leveraging Reserved Pricing in TCO Comparisons

Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) provide different ways to purchase an instance (virtual server) in the cloud. The On-Demand Instance pricing option lets you purchase an instance by the hour with no long-term commitments—you turn capacity on and off instantly. The Reserved Instance (RI) pricing option lets you make a low, one-time payment for each instance you want to reserve, and in turn, you receive a significant discount on the hourly usage charge for that instance, and are guaranteed capacity. The Spot Instance pricing option (available only for Amazon EC2) allows you to bid for unused compute capacity. Instances are charged at the Spot Price, which fluctuates periodically depending on the supply and demand for Spot Instance capacity. Functionally, Reserved Instances, On-Demand Instances, and Spot Instances are the same.

When you are comparing TCO, we highly recommend that you use the Reserved Instance (RI) pricing option in your calculations. They will provide the best apples-to-apples TCO comparison between on-premises and cloud infrastructure. Reserved Instances are similar to on-premises servers because in both cases, there is a one-time upfront cost. However, unlike on-premises servers, Reserved Instances can be "purchased" and provisioned within minutes—and you have the flexibility to turn them off when you don't need them and stop paying the hourly rate.

If you know how much you plan to utilize your Reserved Instances, you can save even more. AWS offers Light, Medium, and Heavy Utilization Reserved Instances. The Light Utilization model is a great option if you have periodic workloads that run only a couple of hours a day or a few days a week. Medium Utilization Reserved Instances are the same Reserved Instances that Amazon EC2 has offered for last several years. They are a great option if you don't plan to run your instances all the time, and if you want the option to shut down your instances when you're not using them. If you need a consistent baseline of capacity or if you run steady-state workloads, the Heavy Utilization model is the best option. Table 1 shows how much you can potentially save compared to running On-Demand Instances.

Reserved Instance Offering Types	Savings Over On-Demand Instances[1]	
Light Utilization Reserved Instances	up to 42% (1-year)	up to 56% (3-year)
Medium Utilization Reserved Instances	up to 49% (1-year)	up to 66% (3-year)
Heavy Utilization Reserved Instances	up to 54% (1-year)	up to 71% (3-year)

Table 1: Savings of Reserved Instance Types over On-Demand Instances

Web Application Usage Patterns

Usage traffic can dramatically affect the TCO of a web application. When determining TCO, you should consider the nature of the application and historical statistical data. This information can help you determine the usage pattern of the application that you plan to deploy. In this paper, we compare costs for three different usage patterns:

1. **Steady State**. The load remains at a fairly constant level over time and you can accurately forecast the likely compute load for these applications.
2. **Spiky but Predictable**. You can accurately forecast the likely compute load for these applications, even though usage varies by time of day, time of month, or time of year.
3. **Uncertain and Unpredictable**. It is difficult to forecast the compute needs for these applications because there is no historical statistical data available.

Scenarios

Amazon Web Services is designed to allow you to save money in each of the usage patterns described above. The AWS cloud provides you with a range of options to reduce costs while retaining the flexibility and scalability benefits of the cloud. In this whitepaper, we use three web application scenarios, map each scenario to a usage pattern, and compare the costs of running these web applications in an on-premises data center vs. the equivalent cloud environment on AWS.

Usage Pattern	Scenario
Steady State	A Corporate Website
Spiky but Predictable	A Sports Event Website
Uncertain and Unpredictable	A Social Coupon Sharing Application

Table 2: Web Application Scenarios

[1] assuming 100% utilization ("Always-On")

Compute and database resources account for the majority of the costs when deploying a web application. While our customers also find AWS to be less expensive for other resources (such as load balancers, content delivery network, storage, and data transfer), we have not included these costs in the calculations to keep the model relatively simple.

Summary of TCO Analysis of Scenarios

With AWS, you can match compute and database capacity to the usage pattern, which both saves money and allows you to scale to meet your performance objectives. With on-premises infrastructure, you really only have one option for all three usage patterns—you have to pay upfront for the infrastructure that you think you'll need, and then hope that you haven't over-invested (paying for unused capacity) or under-invested (risking performance or availability issues). The graph in Figure 1 shows the summary of the TCO cost analysis for the three scenarios that we discuss in the next section in detail. AWS offers significant savings in each scenario over an equivalent solution deployed on-premises.

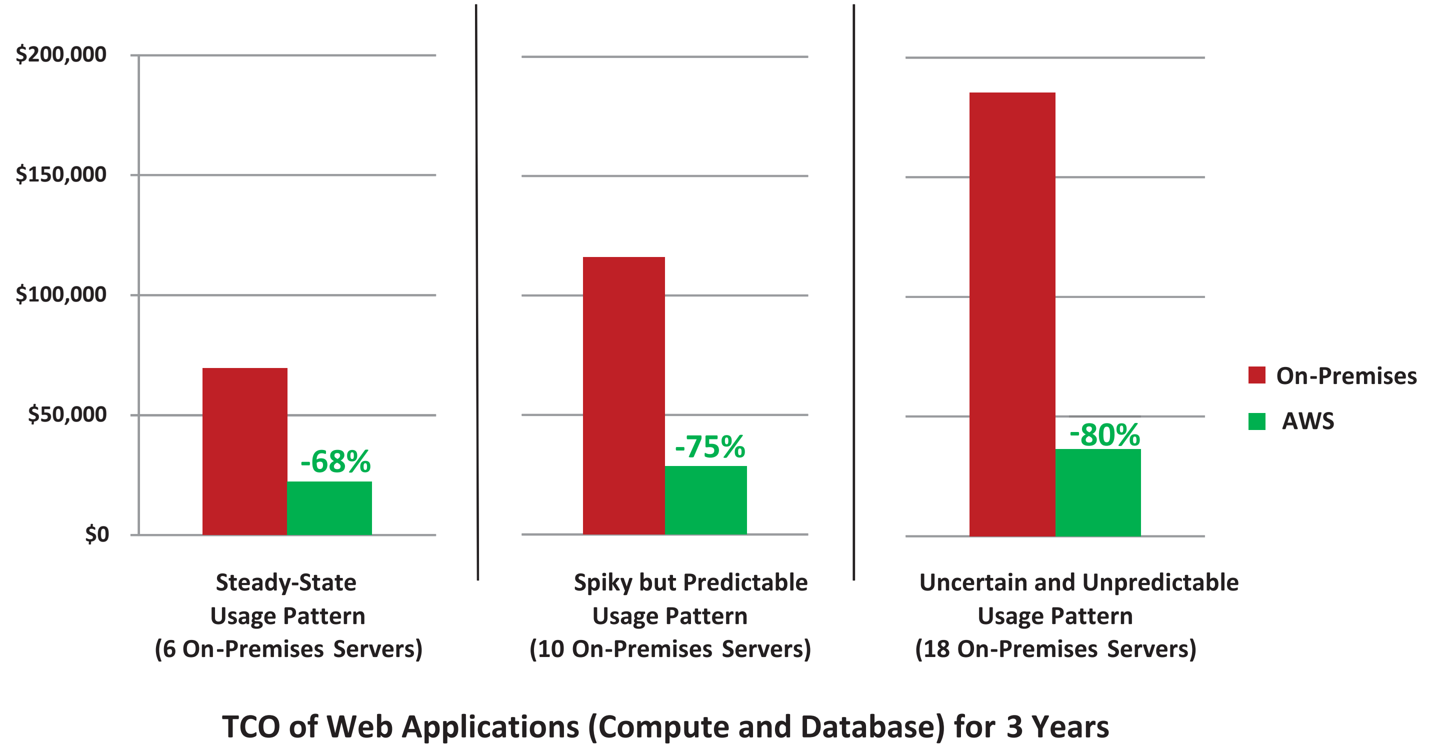

Figure 1: Summary of TCO Analysis of Web Application Scenarios

Although there are significant one-time costs when provisioning hardware, in this paper, we have amortized the costs monthly over a 3-year period for fair comparison across Reserved Instances, On-Demand Instances, and on-premises servers. Hence as the number of servers or traffic load increase, the corresponding savings also increase in an essentially linear relationship.

Scenario 1 – Steady-State Web Application

For this scenario, we assume that your company wants to deploy its corporate website—the official public-facing site that it uses to interact with prospects, customers, and partners. The website showcases all of the various brands of your company and its subsidiaries, provides a listing of all the products and their specifications in an online catalog, lists all of the key stakeholders and the board of directors, and offers investor and public relations services.

The website attracts hundreds of thousands of visitors every month and is regularly accessed by thousands of customers outside the United States. For the most part, the traffic flow is fairly steady state with small occasional blips every few months.

The website is a three-tier web application that leverages open source content management and publishing software, stores and serves a large amount of static media content (videos and PDFs) through a content delivery network, and uses a relational database to drive dynamic content that provides a personalized and interactive user experience.

To support this website, let's assume the following compute resources:

- Two Linux servers for the web servers
- Two Linux servers for the application servers
- Two Linux servers for the MySQL database servers

Usage Graph

The usage graph in Figure 2 shows an example traffic pattern for a steady-state web application. In order to meet this demand in the on-premises environment, you would order, pay for, install, and configure 6 physical servers. With AWS, you would have multiple options from which to choose.

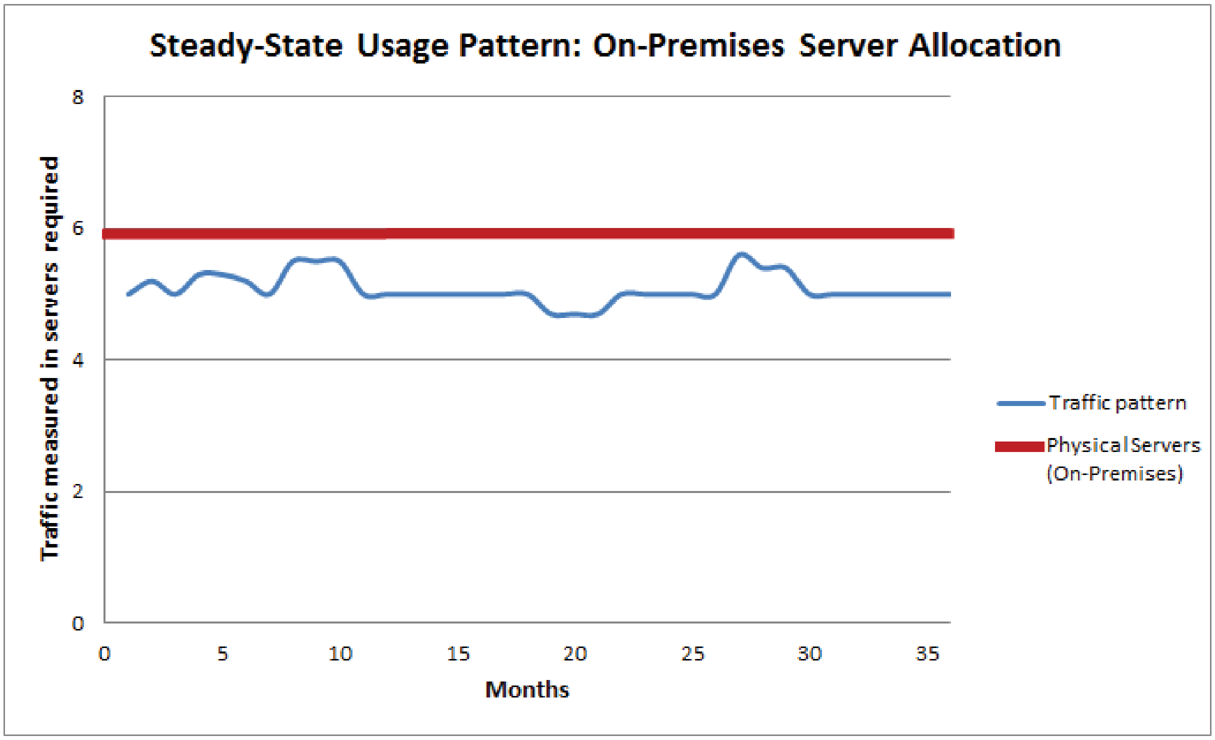

Figure 2: On-Premises Server Allocation for Steady-State Usage Pattern

Different Options Considered

Table 3 shows different deployment options (on-premises and AWS) you can consider for steady-state web application workloads:

	On-Premises Option	AWS Option 1 All Reserved (3-Year Heavy)	AWS Option 2 Mix of On-Demand and Reserved	AWS Option 3 All On-Demand
Web servers	2 Servers	2 Reserved Heavy Utilization (3-Year Term)	Baseline: 1 Reserved Heavy Utilization (3-Year Term) Additional: 1 On-Demand Instance	2 On-Demand Instances
App servers	2 Servers	2 Reserved Heavy Utilization (3-Year Term)	Baseline: 1 Reserved Heavy Utilization (3-Year Term) Additional: 1 On-Demand Instance	2 On-Demand Instances
Database servers	2 Servers	2 Reserved Heavy Utilization (3-Year Term)	2 Reserved Heavy Utilization (3-Year Term)	2 On-Demand Instances

Table 3: Different Options Considered for Steady-State Web Application Scenario

TCO Comparison Across Options Considered

Table 4 compares the TCO of various AWS options vs. the on-premises alternative:

TCO	**Web Application – Steady-State Usage Pattern**			
Amortized Monthly Costs Over 3 Years	**On-Premises Option**	**AWS Option 1** All Reserved (3-Year Heavy)	**AWS Option 2** Mix of On-Demand and Reserved	**AWS Option 3** All On-Demand
Compute/Server Costs				
Server Hardware	$306	$0	$0	$0
Network Hardware	$62	$0	$0	$0
Hardware Maintenance	$47	$0	$0	$0
Power and Cooling	$172	$0	$0	$0
Data Center Space	$144	$0	$0	$0
Personnel	$1,200	$0	$0	$0
AWS Instances	$0	$618	$1,079	$2,138
Total – Per Month	**$1,932**	**$618**	**$1,079**	**$2,138**
Total – 3 Years	**$69,552**	**$22,260**	**$38,859**	**$76,982**
Savings over On-Premises Option		**68%**	**44%**	**–11%**

Recommended option (most cost-effective)

Table 4: TCO Comparison – Steady-State Usage Pattern

Cost Assumptions

On-Premises Option

System costs: $1,932 per month ($322 per server).

This is the monthly cost of running 6 physical servers with a High-Memory system configuration amortized over 3 years. This includes the cost of server hardware, network hardware, power and cooling, and data center real estate and personnel costs. Detailed cost breakdown and assumptions are highlighted in Appendix A.

Personnel costs ($1,200 per month to manage 6 physical servers) include the cost of the sizable IT infrastructure teams that are needed to handle the "heavy lifting" of managing physical infrastructure:

- Hardware procurement teams are needed. These teams have to spend a lot of time evaluating hardware, negotiating contracts, holding hardware vendor meetings, managing delivery and installation, etc. It's expensive to have a staff with sufficient knowledge to do this well.

- Data center design and build teams are needed to create and maintain reliable and cost-effective facilities. These teams need to stay up-to-date on data center design and be experts in managing heterogeneous hardware and the related supply chain, managing legacy software, moving facilities, scaling and managing physical growth—all the tasks that an enterprise needs to do well if it wants to achieve low incremental costs.

- Operations staff is needed 24/7/365 in each facility.

- Database administration teams are needed to manage the MySQL Databases. This staff is responsible for installing, patching, upgrades, migration, backups, snapshots and recovery of databases, ensuring availability, troubleshooting, and performance enhancements.

- Networking teams are needed for running a highly available network. Expertise is needed to design, debug, scale, and operate the network and deal with the external relationships necessary to have cost-effective Internet transit.

- Security personnel are needed at all phases of the design, build, and operations process.

While the actual personnel costs to support production web application projects typically involve many different people, we use a simple ratio of servers to people in our cost models for the sake of simplicity. We are using a total annual cost of $120,000 per person, which is intended to represent a fully loaded cost (both salary and benefits), and we're assuming a 50:1 server-to-people ratio. The actual server-to-people ratio can vary a lot because it depends on a number of factors such as sophistication of automation and tools and preference for virtualized vs. non-virtualized environments. Based on our discussions with customers, we have found that a 50:1 ratio represents a good middle point of the range of what we see. We recommend that you adjust these assumptions based on your own research and experience and include the personnel costs of all the people involved in building and managing a physical data center, not just the people who rack and stack servers (that's why we are calling the ratio "server-to-people" instead of "server-to-admin").

The total cost of running the steady-state web application (compute and database) on-premises for 3 years = $69,552.

AWS Option 1: All Amazon EC2 Reserved Instances (3-Year Heavy Utilization)

In this option, we assume that you will purchase Reserved Instances for a 3-year term. Since this is a steady-state workload and you are planning to run these instances 24 hours per day, Heavy Utilization Reserved Instances is an attractive, cost-effective option.

Total monthly cost of 6 Reserved Instances amortized over a 3-year period:

> 2 web servers and 2 application servers: The instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon EC2 Instance running in the US East Region at a rate of $0.07 per hour with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$374**.
>
> 2 database servers: The DB instance type used is a High-Memory Extra Large, 3-Year Reserved Amazon RDS DB Instance running in the US East Region with Master-Slave (Multi-AZ) configuration at a rate of $0.011 per hour with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$244**.

The total cost of running the steady-state web application (compute and database) on Reserved Instances for 3 years = $22,260 ($618 per month).

Summary

This is the most cost-effective option. You save 68% over the on-premises alternative. By purchasing 3-Year Heavy Utilization Reserved Instances, you get the maximum savings and lowest rates for your Amazon EC2 Instances and Amazon RDS DB Instances.

AWS Option 2: Mix of Amazon EC2 Reserved Instances (3-Year Heavy Utilization) and On-Demand Instances

In this option, we assume you will purchase 3-year Heavy Utilization Reserved Instances for the minimum number of servers you need to run your application (i.e. your baseline), thereby reducing your total upfront commitment. For additional servers, we assume you will leverage On-Demand Instances.

Please note that you can purchase Reserved Instances anytime. Unlike the on-premises option with Reserved Instances, you don’t have to plan ahead for capacity or allocate the time it takes to build out physical data center capacity. When you purchase Reserved Instances, your billing will automatically switch from the On-Demand Instance hourly rate to the Reserved Instance discounted hourly rate.

Baseline (minimum servers needed to run a three-tier web application)

Total monthly cost of 4 Reserved Instances amortized over a 3-year period:

> 1 web server and 1 application server: The instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon EC2 Instance running in the East Region at a rate of $0.07 per hour with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$187**.
>
> 2 database servers: The DB instance type used is a High-Memory, Extra Large 3-Year Heavy Utilization Reserved Amazon RDS DB Instance running in the US East Region with Master-Slave (Multi-AZ) configuration at a rate of $0.011 per hour, with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$244**.

Peak (additional servers needed)

Total monthly cost of 2 On-Demand Instances amortized over a 3-year period:

1 web server and 1 application server: The instance type used is a High-Memory Extra Large, On-Demand Amazon EC2 Instance running in the US East Region at a rate of $0.45 per hour for 24 hours/day (Always-on). The amortized monthly cost for these servers is **$648.**

The total cost of running the steady-state web application (compute and database) on Reserved Instances for 3 years = $38,859 ($1,079 per month).

Summary

This option offers 44% savings over the on-premises alternative. It's a lower upfront commitment ($6,200) than either AWS Option 1 ($9,300) or the on-premises option ($14,952). If you're not confident about your peak capacity needs or if you want to have a little more flexibility while still saving costs, you might choose this option. However, since this is a steady-state workload where demand is largely predictable, we still recommend the AWS Option 1 over this more flexible AWS Option 2.

AWS Option 3: All Amazon EC2 On-Demand Instances

In this option, we assume that you will choose On-Demand Instances to run your steady-state web application. Unlike the on-premises option, with On-Demand Instances, you don't have to plan ahead for capacity or purchase any resources ahead of time. You simply start and stop your Amazon EC2 Instances and Amazon RDS DB Instances for the hours you want to use. You are billed every month based on your usage. In this case, since this is a steady-state workload, we assume you will keep the instances running for 24 hours per day.

Total monthly cost of 6 On-Demand Instances:

4 web and application servers: The instance type used is a High-Memory Extra Large, On-Demand Amazon EC2 Instance running in the US East Region at a rate of $0.45 per hour for 24 hours/day (Always-on).

2 database servers: The DB instance type used is a High-Memory Extra Large, On-Demand Amazon RDS DB Instance running in the US East Region at a rate of $0.585 per hour for 24 hours/day (Always-on).

The total cost of running the steady-state web application (compute and database) on On-Demand Instances for 3 years = $76,982 ($2,138 per month).

Summary

With AWS, you have an option to choose **zero upfront commitment** and leverage On-Demand Instances for your steady-state workloads. Some AWS customers prefer this option because it allows them to start small with no upfront commitment, and provides maximum flexibility while reducing risk to close to zero. For only a 11% cost premium over on-premises infrastructure—which requires 100% upfront purchase and very little flexibility—they have an environment that can be started up or completely shut down to zero at a moment's notice. And, of course, you can always optimize your cost later by replacing On Demand instances with Reserved Instances.

Recommended Option for Steady-State Web Application: 3-Year Heavy Utilization Reserved Instances

As you can see from the above calculations, if you have a web application with uniform steady-state traffic, the most cost-effective option is to use 3-Year Heavy Utilization Reserved Instances (AWS Option 1). This option offers 68% savings over the on-premises alternative.

Scenario 2 – Spiky But Predictable Web Application

For this scenario, we assume that your company has a usage pattern that is similar to that of a sports association that manages a website to connect with its members and fans. The website provides scores in real-time, live updates from annual tournaments, and detailed historical data and player profiles from previous matches and tournaments.

The website is a three-tier web application that leverages open source content management and publishing software, stores and serves a large amount of static media content (videos and PDFs) through a content delivery network, and uses a relational database to drive a personalized and interactive user experience.

The website attracts hundreds of thousands of visitors every month and is regularly accessed by fans and members outside the United States. Once a year, during the annual tournament, the website experiences a surge in traffic that is three times higher than its steady-state traffic. Since the tournament occurs during a specific time of the year, the company has plenty of time to plan ahead. Also, since it has data from previous years, it has a pretty good idea of how much infrastructure it will need to meet the demand.

Since the annual tournament is the company's most important event of the year, the company cannot afford to deliver a poor user experience during this event. Hence, it always provisions for peak capacity for the tournament.

To support this website, let's assume the following compute resources:

- Baseline servers (minimum servers needed)
  - One Linux server for the web server
  - One Linux server for the application server
  - Two Linux servers for the MySQL database servers
- Peak servers (additional servers needed)
  - Three Linux servers for the web servers
  - Three Linux servers for the application servers

Usage Graph

The usage graph in Figure 3 shows an example traffic pattern for a spiky web application where the spikes represent the company's annual big event. To meet this demand in the on-premises environment, you would provision for peak capacity (10 servers). To meet this demand in the AWS cloud environment, you have several options, which are detailed below.

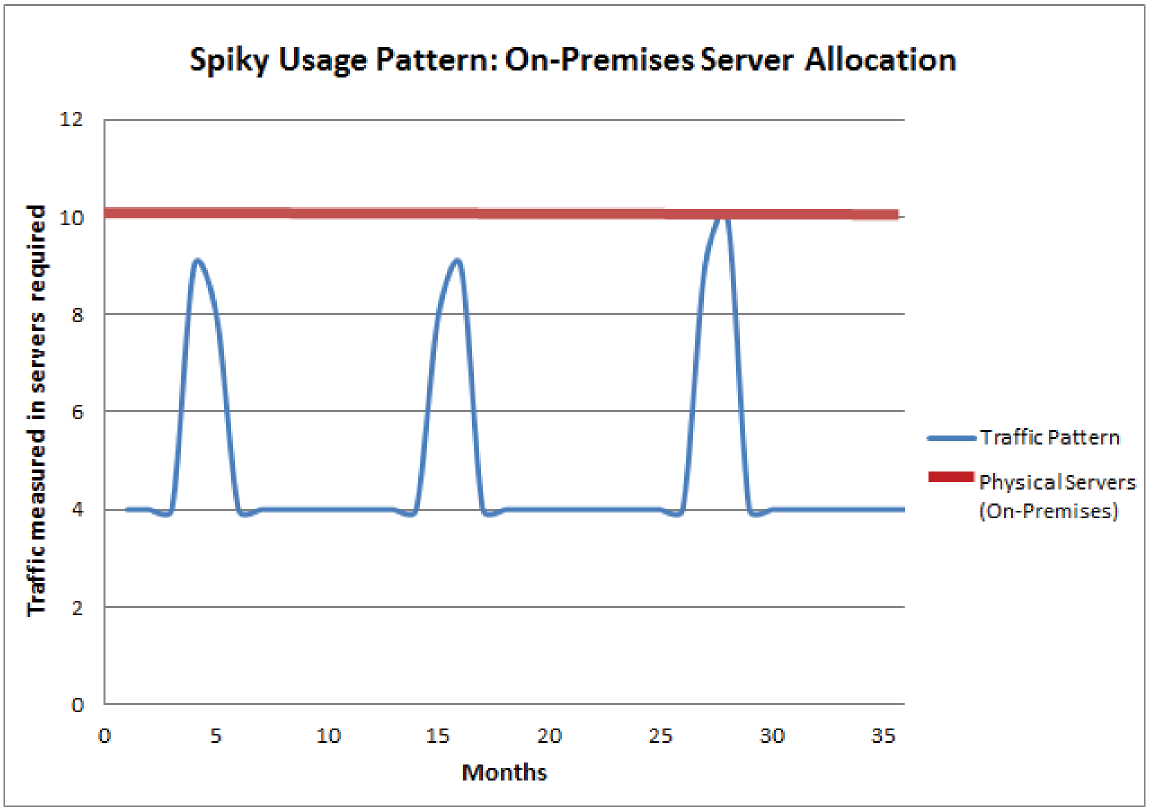

Figure 3: On-Premises Server Allocation for Spiky Usage Pattern

Different Options Considered

Table 5 shows different options (on-premises and AWS) to consider for spiky web application workloads:

	On-Premises Option	AWS Option 1 All Reserved	AWS Option 2 Mix of On-Demand and Reserved	AWS Option 3 All On-Demand
Web servers	4 Servers	4 Reserved Heavy Utilization	Baseline: 1 Reserved Heavy Utilization Peak: 3 On-Demand Instances	4 On-Demand Instances
App servers	4 Servers	4 Reserved Heavy Utilization	Baseline: 1 Reserved Heavy Utilization Peak: 3 On-Demand Instances	4 On-Demand Instances
Database servers	2 Servers	2 Reserved Heavy Utilization	Baseline: 2 Reserved Heavy Utilization	2 On-Demand Instances

Table 5: Different Options Considered for Spiky Web Application Scenario

TCO Comparison Across Options Considered

Table 6 compares the TCO of various AWS options vs. the on-premises alternative:

TCO	Web Application – Spiky Usage Pattern			
Amortized Monthly Costs Over 3 Years	**On-Premises Option**	**AWS Option 1** All Reserved	**AWS Option 2** Mix of On-Demand and Reserved	**AWS Option 3** All On-Demand
Compute/Server Costs				
Server Hardware	$511	$0	$0	$0
Network Hardware	$103	$0	$0	$0
Hardware Maintenance	$79	$0	$0	$0
Power and Cooling	$287	$0	$0	$0
Data Center Space	$241	$0	$0	$0
Personnel	$2,000	$0	$0	$0
AWS Instances	$0	$992	$791	$1,850
Total – Per Month	**$3,220**	**$992**	**$791**	**$1,850**
Total – 3 Years	**$115,920**	**$35,718**	**$28,491**	**$66,614**
Savings over On-Premises Option		**69%**	**75%**	**43%**

⬆

Recommended option (most cost-effective)

Table 6: TCO Comparison – Spiky Usage Pattern

Cost Assumptions

On-Premises Option

System costs: $3,220 ($322 per server per month).

This is the monthly cost of running 10 physical servers with High-Memory system configuration amortized over a 3-year period. This includes the cost of server hardware, network hardware, power and cooling, and data center real estate. Detailed cost breakdown and assumptions are highlighted in Appendix A.

Personnel costs ($2,000 per month to manage 10 physical servers) are calculated using the same assumptions as the previous scenario.

The total cost of running the spiky web application (compute and database) on-premises for 3 years = $115,920.

AWS Option 1: All Amazon EC2 Reserved Instances

In this option, we assume you will purchase Reserved Instances for a 3-year term. Since you are planning to run these instances 24 hours per day, we recommend using Heavy Utilization Reserved Instances.

Total monthly cost of 10 Reserved Instances amortized over a 3-year period:

> 4 web servers: The instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon EC2 Instance running in the US East Region at a rate of $0.07 per hour, with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$374**.
>
> 4 application servers: The instance type used is a High-Memory, Extra Large, 3-Year Heavy Utilization Reserved Amazon EC2 Instance running in the US East Region at a rate of $0.07 per hour, with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$374**.
>
> 2 database servers: The DB Instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon RDS DB Instance running in the US East Region with Master-Slave (Multi-AZ) configuration at a rate of $0.011 per hour with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$244**.

The total cost of running the spiky web application (compute and database) on all Reserved Instances for 3 years = $35,718 ($992 per month).

Summary

This option offers 69% savings over the on-premises alternative. By purchasing 3-Year Heavy Utilization Reserved Instances (to match the capacity in the on-premises option), you get the lowest hourly rate for your Amazon EC2 and Amazon RDS DB Instances.

AWS Option 2: Mix of Amazon EC2 On-Demand Instances and Reserved Instances

In this option, we assume that you will choose 3-Year Heavy Utilization Reserved Instances for your baseline steady-state traffic and On-Demand Instances for the annual peak of the sports tournament, and that you will stop running these On-Demand Instances after the traffic peak subsides so you are only paying for the extra capacity when you need it during that peak.

Baseline Servers (minimum needed for non-peak user traffic)

Total monthly cost of 4 Reserved Instances amortized over a 3-year period:

> 1 web server and 1 application server: The instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon EC2 Instance running in the US East Region at a rate of $0.07 per hour with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$187**.

2 database servers: The DB instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon RDS DB Instance running in the US East Region with Master-Slave (Multi-AZ) configuration at a rate of $0.011 per hour, with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$244**.

Peak Servers (maximum needed to handle the spikes)

Since there is one spike every year that lasts for 3 months, you will add additional On-Demand servers to handle the additional traffic. On-Demand Instances can be turned off anytime; you stop paying for them as soon as they are shut down. To keep things simple, we assume the instances are running 24/7 the entire month. The number of additional servers (web and application servers) needed to handle that spike as shown in Table 7.

Additional capacity needed to handle the spikes every year for 3 years (including the buffer capacity) is 28,800 instance hours. The instance type used is a High-Memory Extra Large On-Demand Amazon EC2 Instance running in the US East Region at a rate of $0.45 per hour.

The total monthly cost of the elastic On-Demand Instances, amortized over a 3-year period is **$360**.

Month #	Additional servers needed for peak load	Instance Hours Consumed
1–2	0	8,640 (12 instances X 24 hours X 30 days)
3	5	
4	6	
5	1	
6–12	0	
13–14	0	8,640 (12 instances X 24 hours X 30 days)
15	6	
16	5	
17	1	
18–24	0	
25–26	0	11,520 (16 instances X 24 hours X 30 days)
27	4	
28	6	
29	6	
30–36	0	
Total	**40**	**28,800**

Table 7: On-Demand Instance Assumptions

The total cost of running the spiky web application (compute and database) on a combination of Reserved Instances and On-Demand Instances for 3 years = $28,491 ($791 per month).

Summary

This is the most cost-effective option and also the most flexible option. By purchasing 3-Year Heavy Utilization Reserved Instances to handle your baseline traffic and leveraging On-Demand Instances for your peaks, you save 75% over the on-premises option. These significant savings are driven by highly efficient resource utilization—you use your resources only when you need them and shut them down after your peak traffic subsides. You never pay for capacity when you don't need it. You also have lower total upfront cost ($6,200) than AWS option 1 ($15,500) and the on-premises option ($24,920).

AWS Option 3: All Amazon EC2 On-Demand Instances

In this option, we assume you will choose all On-Demand Instances to run your spiky web application. With On-Demand Instances, you don't have to plan ahead for capacity or purchase any resources ahead of time. You simply start and stop your Amazon EC2 Instances and Amazon RDS DB Instances for whatever hours you see fit, and are billed every month based on your usage.

Total monthly cost of all On-Demand Instances for a 3-year period:

1 web server and 1 application server: The instance type used is a High-Memory Extra Large On-Demand Amazon EC2 Instance running in the US East Region at a rate of $0.45 per hour. The monthly cost is **$648**.

2 database servers (running "Always On"): The DB instance type used is a High-Memory Extra Large On-Demand Amazon RDS DB Instance running in the US East Region, with a Master-Slave (Multi-AZ) configuration, at a rate of $0.585 per hour. The monthly cost is **$842**.

Additional capacity (web servers and application servers—running "On/Off" as demand dictates) needed to handle the spikes (including the buffer capacity) is 28,800 Instance hours (same as shown in AWS Option 2 above). The instance type used is a High-Memory Extra Large On-Demand Amazon EC2 Instance running in the US East Region at a rate of $0.45 per hour. The total monthly cost of On-Demand Instances amortized over a 3-year period is **$360.**

The total cost of running the spiky web application (compute and database) on all On-Demand Instances for 3 years = $66,614 ($1,850 per month).

Summary

In this option, there is **no upfront commitment** and you still get significant savings (43%) over the on-premises alternative. By leveraging On-Demand Instances, you pay only for what you use. This option is best if you want maximum flexibility and zero upfront cost (e.g. many early-stage start-ups fit this profile). The savings are less than the AWS options with Reserved Instances, but you still get significant savings and flexibility over on-premises alternative with this option.

Recommended Option for Spiky Web Applications: Mix of Reserved Instances and On-Demand Instances

As you can see from the above calculations, when you expect your web application to have spiky usage patterns and you can accurately predict the timing and approximate size of the peaks, the most cost-effective option is to use Reserved Instances for baseline servers and On-Demand Instances to handle spikes in traffic. This option offers 72% savings over the on-premises alternative.

Scenario 3 – Uncertain and Unpredictable Usage Pattern

For this scenario, we assume that your company launches a social web application as a new business initiative. This application integrates with Facebook and allows people to share coupons for discounts on your products with their friends.

The website is a three-tier web application that leverages open-source content management and publishing software, stores and serves a large amount of static media content (videos and PDFs) through a content delivery network, and uses a relational database to deliver a personalized user experience to its visitors.

The company has no historical data or experience in launching such an application. Although you believe that this "experiment" has the potential to bring in a lot of advertising revenue, you have no idea whether it will be successful. You would like to maximize your savings if your application is successful, and lower your risk and cost if it is not. The company decides to purchase the infrastructure with a "best guess estimate" of the total number of servers needed for a 3-year period, which in this case are 16. Choosing the right number of servers in a scenario with highly uncertain usage patterns is an exercise of balancing cost and risk. In this case, the web application is public facing and relatively high profile as it is deployed as a Facebook application. The negative impact to your business of under-provisioning and being caught off-guard by an unexpected spike in traffic is very high. Thus, we are going to assume that you are being conservative with your initial specification for server requirements.

To support this website, let's assume the following compute resources:

- Seven Linux servers for the web server
- Seven Linux servers for the application server
- Two Linux servers for the MySQL database servers

At first, usage of the application grows steadily, but after the first year, customer usage starts to drop off. After around 15 months, the traffic drops to a very low level and never recovers. This new business initiative is deemed to be a failure.

Usage Graph

The usage graph in Figure 4 shows an example of an uncertain and unpredictable traffic pattern.

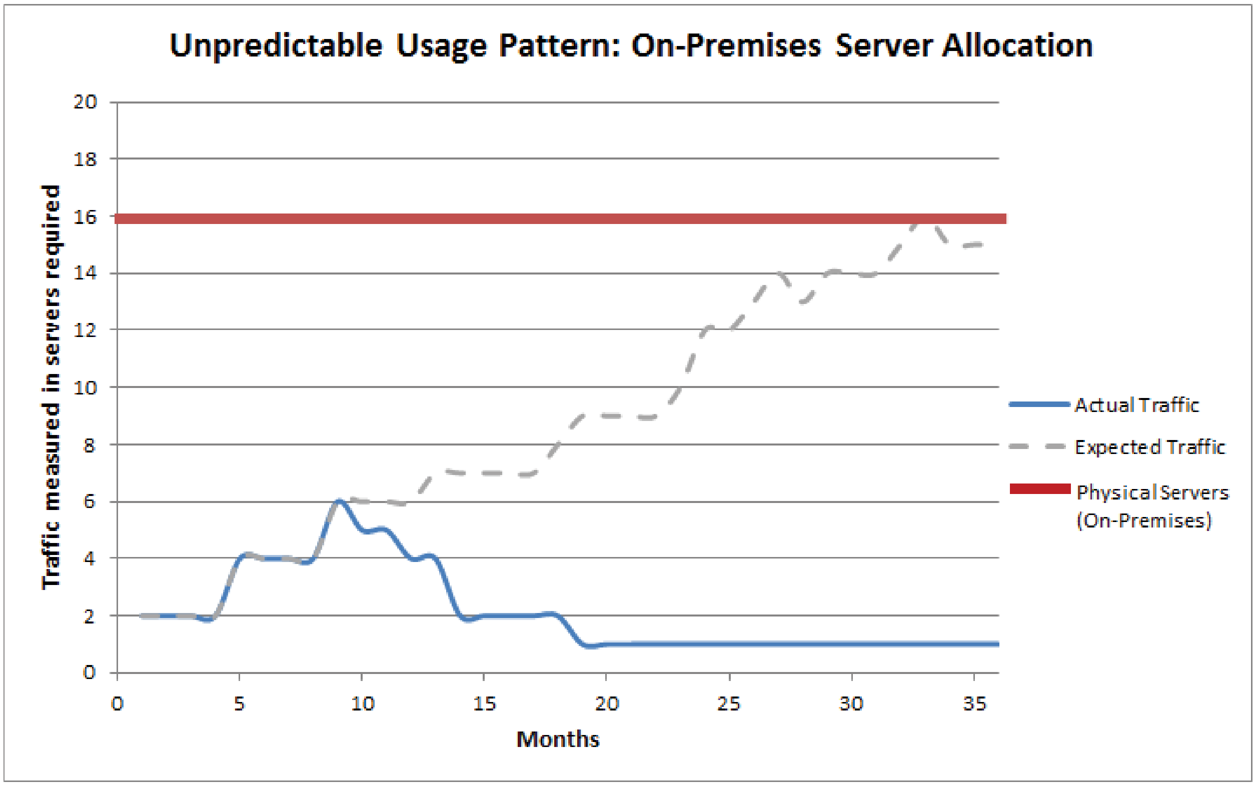

Figure 4: On-Premises Server Allocation for Uncertain and Unpredictable Usage Pattern

Different Options Considered

Table 8 shows different options (on-premises and AWS) to consider for uncertain workloads:

	On-Premises Option	AWS Option 1 All Reserved	AWS Option 2 Mix of On-Demand and Reserved	AWS Option 3 All On-Demand
Web servers	7 Servers for 3 years	7 Reserved Heavy Utilization (3-Year Term)	7 Reserved Heavy Utilization (1-Year Term) for 1 year On-Demand Instances after 1 Year	On-Demand Instances
App servers	7 Servers for 3 years	7 Reserved Heavy Utilization (3-Year Term)	7 Reserved Heavy Utilization (1-Year Term) for 1 year On-Demand Instances after 1 Year	On-Demand Instances
Database servers	2 Servers for 3 years	2 Reserved Heavy Utilization (3-Year Term)	2 Reserved Heavy Utilization (1-Year Term) for 1 year On-Demand Instances after 1 Year	On-Demand Instances

Table 8: Different Options Considered for Uncertain Unpredictable Web Application Scenario

TCO Comparison Across Options Considered

Table 9 compares the TCO of various AWS options vs. the on-premises alternative:

TCO	Web Application – Unpredictable Usage Pattern			
Amortized Monthly Costs Over 3 Years	**On-Premises Option**	**AWS Option 1** All Reserved	**AWS Option 2** Mix of On-Demand and Reserved	**AWS Option 3** All On-Demand
Compute/Server Costs				
Server Hardware	$817	$0	$0	$0
Network Hardware	$165	$0	$0	$0
Hardware Maintenance	$126	$0	$0	$0
Power and Cooling	$459	$0	$0	$0
Data Center Space	$385	$0	$0	$0
Personnel	$3,200	$0	$0	$0
AWS Instances	$0	$1,553	$1,394	$1,051
Total – Per Month	**$5,152**	**$1,553**	**$1,394**	**$1,051**
Total – 3 Years	**$185,472**	**$55,904**	**$50,193**	**$37,843**
Savings over On-Premises Option		**70%**	**73%**	**80%**

Recommended option (most cost-effective)

Table 9: TCO Comparison – Unpredictable Usage Pattern

Cost Assumptions

On-Premises Option

System costs: $5,152 ($322 per server per month).

> This is the monthly cost of running 16 physical servers with a High-Memory system configuration amortized over a 3-year period. This includes the cost of server hardware, network hardware, power and cooling and data center real estate. Detailed cost breakdown and assumptions are highlighted in Appendix A.
>
> Personnel costs ($3,200 per month to manage 16 physical servers) are calculated using the same assumptions as the previous scenarios.

The total cost of running a web application (compute and database) on-premises for 3 years = $185,472.

AWS Option 1: All Amazon EC2 Reserved Instances (3-Year Heavy Utilization)

In this option, we assume you will purchase 3-Year Heavy Utilization Reserved Instances ahead of time so your expected traffic can be handled.

Total monthly cost of 16 Reserved Instances amortized over a 3-year period:

> 7 web servers and 7 application servers: The instance type used is a High-Memory Extra Large, 3-Year Heavy Utilization Reserved Amazon EC2 Instance running in the US East Region at a rate of $0.07 per hour, with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$1,309**.
>
> 2 database servers: The DB instance type used is a High-Memory Extra Large, 3-Year Reserved Amazon RDS DB Instance running in the US East Region with Master-Slave (Multi-AZ) configuration at a rate of $0.011 per hour, with a one-time upfront cost of $1,550. The amortized monthly cost for these servers is **$244**.

The total cost of running a web application (compute and database) on Reserved Instances for 3 years = $55,904 ($1,523 per month).

Summary

If this application had turned out to be as successful as expected, and your prediction of 16 servers was the correct (or too low an) assumption for the Amazon EC2 capacity that you would need, you would have saved 70% vs. using on-premises infrastructure. The application in this scenario was not successful, however, and thus you have ended up over-buying capacity vs. the other AWS options below.

AWS Option 2: Mix of Amazon EC2 Reserved Instances (1-Year Heavy Utilization) and On-Demand Instances

In this option, we assume you will buy Reserved Instances but will choose a 1-year term instead of a 3-year term, since you are uncertain about the usage pattern and you want the option to renew your Reserved Instances after one year only if your traffic grows as expected.

Since the traffic dropped after the first year, you did not renew your Reserved Instances and ran only On-Demand instances to satisfy the subsequent demand in years 2 and 3. As a result, you saved significant cost in years 2 and 3 vs. having bought 3-year Reserved Instances for all of your projected capacity needs over the next 3 years. This is not possible to do in the on-premises scenario because you have already paid for and provisioned the servers in your data center.

Year 1: 1-Year Reserved Instances

Total monthly cost of 16 Reserved Instances amortized over a 3-year period ($825 per month):

> 7 web servers and 7 application servers: The instance type used is a High-Memory Extra Large, 1-Year Heavy Utilization Reserved Amazon EC2 Instance for 1 year running in the US East Region at a rate of $0.088 per hour, with a one-time upfront cost of $1,030. The amortized monthly cost for these servers is **$696**.
>
> 2 database servers: The DB instance type used is a High-Memory Extra Large, 1-Year Heavy Utilization Reserved Amazon RDS DB Instance for 1 year, running in the US East Region, with Master-Slave (Multi-AZ) configuration at a rate of $0.15 per hour, with a one-time upfront cost of $1,030. The amortized monthly cost for these servers is **$129**.
>
> Note that during the first year, you do not need additional On-Demand Instances as the Reserved Instances are sufficient to handle the traffic.

Years 2 and 3: Switch from Reserved Instances to On-Demand Instances

> After the first year, since the traffic was less than expected, you decided not to renew your Reserved Instance subscription. As a result, you automatically get billed at the On-Demand hourly rate for your Amazon EC2 instances ($0.45 per hour) and Amazon RDS DB Instances ($0.585 per hour). The total number of On-Demand Instance hours consumed during the next 24 months is 40,320 (Table 10 shows the actual usage for each month). The total monthly cost of these On-Demand Instances amortized over a 3-year period is **$569**.

Month #	On-Demand Instances	Instance Hours Consumed
13	5	40,320 (56 instances X 24 hours X 30 days)
14-18	3	
19–36	2	

Table 10: On-Demand Instance Assumptions for Years 2 and 3

The total cost of running a web application (compute and database) on Reserved Instances for 3 years = $50,193 ($1,394 per month).

Summary

This option offers 73% savings over the on-premises option. By purchasing 1-Year Heavy Utilization Reserved Instances, you pay less upfront ($16,480) than AWS option 1 ($24,800) and the on-premises option ($39,872) and are committed only to a 1-year period. You have the flexibility to turn off unneeded Reserved Instances after a year and avoid any further upfront costs or commitments.

AWS Option 3: All Amazon EC2 On-Demand Instances

In this option, we assume you will choose all On-Demand Instances to run your web application. With On-Demand Instances, you don't have to plan ahead for capacity or purchase any resources ahead of time. You can simply start and stop your Amazon EC2 Instances and Amazon RDS DB Instances for whatever hours you see fit, and are billed every month based on your usage. In this case, you consistently lower the amount of On-Demand capacity that you consume to match the falling traffic that your application is experiencing.

Table 11 illustrates the assumptions we have used for the number of On-Demand Instances that are used in each month of our scenario. The total number of On-Demand Instance hours consumed is 76,320.

Total monthly cost of running on On-Demand Instances:

> Web and application servers: The instance type used is a High-Memory Extra Large On-Demand Amazon EC2 Instance running in the US East Region at a rate of $0.45 per hour. The total monthly amortized cost of running Amazon EC2 On-Demand Instances (50,400 Instance hours) is **$630**.
>
> 1 database server: The DB instance type used is a High-Memory Extra Large, On-Demand Amazon RDS DB Instance running in the US East Region at a rate of $0.585 per hour for 36 months (25,920 instance hours). The total amortized monthly cost of running Amazon RDS DB Instances is **$421**.

The total cost of running a web application (compute and database) on On-Demand Instances for 3 years = $37,843 ($1,051 per month).

Month #	On-Demand Instances	Instance Hours Consumed
1	2	76,320 (106 instances X 24 hours X 30 days)
2	2	
3	2	
4	2	
5	4	
6	5	
7	5	
8	5	
9	6	
10	6	
11	6	
12	5	
13	5	
14-18	3	
19-36	2	

Table 11: On-Demand Instance Assumptions

Summary

This is the most-cost effective option and will give you the maximum savings (80%, in this case) over the on-premises alternative. Since you are not sure whether your application will be successful, the On-Demand Instances option is the most attractive option because it requires zero upfront commitment and provides the lowest cost of failure compared to the other AWS options and the on-premise option. And, if at any point, you determine that your application is going to be successful and you want to minimize your predictable costs, you can purchase Reserved Instances at a fraction of the on-premises costs (as shown in these previous scenarios).

Recommended Option for an Uncertain Unpredictable Web Application: On-Demand Instances

As you can see from the preceding calculations, if you are dealing with a new web application and are unsure about its traffic pattern or likelihood for success, the most prudent approach is to use On-Demand Instances (AWS Option 3) because it limits your downside risk, eliminates any upfront or long-term commitment, and provides lower cost and much higher flexibility than the on-premises option. With AWS, customers can start-out with minimal risks and no upfront commitment using on-demand pricing. If their projects are successful, customers can easily (and usually do) shift to some combination of Reserved and On-Demand Instances to achieve additional price savings as their usage patterns become more predictable.

Making predictions about web traffic is a difficult endeavor. The odds of guessing wrong are high, as are the costs. This is also a good illustration of one of the really exciting benefits of cloud computing—lowering the cost of failure. When you lower the cost of failing with new web application projects, you have an opportunity to change the dynamics of decision making and encourage your company to lean-forward into innovation. With cloud computing, you can experiment often, fail quickly at a very low cost, and end up with more innovation as more of your company's ideas are tested in the market.

Scenario Summary

Table 12 compares costs for each option and shows the total savings over the on-premises alternative for the 3-year term:

TCO Summary of Web Application Scenarios						
Usage Patterns	**Steady-State**		**Spiky but Predictable**		**Uncertain and Unpredictable**	
Options	**On-Premises**	**AWS**[1]	**On-Premises**	**AWS**[1]	**On-Premises**	**AWS**[1]
Total – Per Month	**$1,932**	**$618**	**$3,220**	**$791**	**$5,152**	**$1,051**
Total – 3 Years	**$69,552**	**$22,260**	**$115,920**	**$28,491**	**$185,472**	**$37,843**
Savings over On-Premises Option		**68%**		**75%**		**80%**

[1] AWS costs based on recommended configuration for each scenario.

Table 12: TCO Summary of Web Application Scenarios

For each of the three web application scenarios, AWS offers significant savings over hosting the same application on-premises. AWS provides you with the flexibility to choose numerous combinations of On Demand and Reserved Instances that match your usage projections. We offer a wide range of Reserved Instance types that allow you to save more money as you become more certain of individual instance utilization or more certain that you will continue to use your instances for longer periods of time.

The most important thing to remember is that you can start out risk-free and commitment-free with On-Demand Instances for days or weeks or months or years, until you can more clearly evaluate the chances of success for your application. If your application is successful, you can switch to some combination of Reserved Instances and On-Demand Instances to lower your costs for baseline usage, and then use On-Demand Instances to handle spiky or unpredictable traffic. The cost of these choices is significantly less than the on-premises option. If your application is not successful, you walk away having spent a fraction of what you would pay to buy your own technology infrastructure. You not only get lower prices, lower risk, and higher flexibility, but you also get more done, try more new ideas, gain substantial business agility. You get to focus scarce engineering resources on initiatives that differentiate your business rather than on the undifferentiated heavy lifting of infrastructure. When you multiply this model (and the resulting savings) and apply it to the hundreds of applications that your company manages, it becomes clear how powerful the AWS purchasing model is for your web applications, and the overall economics of your business or organization.

Conclusion

While the number and types of services offered by AWS have increased dramatically, our philosophy on pricing has not changed. You pay as you go, pay for what you use, pay less as you use more and grow bigger, and pay even less when you reserve capacity. These are the important points to consider when calculating Total Cost of Ownership (TCO) of running web applications.

Cost-Savings Customer Success Stories

Airport Nuremberg

Airport Nuremberg is the 2nd largest airport in Bavaria, and one of the 10 biggest airports in Germany. The airport handles approximately 4 million passengers and 100,000 tons of cargo every year. Airports are faced with unpredictable access rates on their websites. Extreme weather conditions, strikes or ash clouds can lead to very high access rates. Airport Nuremberg's main goal was to increase the scalability and reliability of their eCommerce and information services while also reducing costs. Led by the marketing department, Airport Nuremburg moved all of their web applications to Infopark's Cloud Express (ICE) service, delivering a highly scalable CMS/CRM-backed personalized dynamic web experience solution running on AWS.

By using ICE and AWS, Airport Nuremburg can scale their website up and down seamlessly, based on traffic demands, and ensure that the website, which is an important source of passenger information, always remains available. **By using the cloud solution, the airport estimates that it is saving 60% – 70% on costs compared to their previous hosting solution.**

foursquare

foursquare Labs, Inc. is a location-based social network in which its more than 10 million users check in via a smartphone app or SMS to exchange travel tips and to share their location with friends. By checking in frequently, users earn points and virtual badges. To perform analytics across more than 5 million daily check-ins, foursquare uses Amazon Elastic MapReduce, Amazon EC2 Spot Instances, Amazon S3, and open-source technologies MongoDB and Apache Flume.

"***By expanding our clusters with Reserved Instances and On-Demand Instances, plus the Amazon EC2 price reductions, we have reduced our analytics costs by over 50% when compared to hosting it ourselves."*** – Matthew Rathbone, Software Engineer, foursquare

Global Blue

Global Blue provides value added tax (VAT)/goods and services tax (GST) refunds in 38 countries, processing millions of transactions a year. In addition to using Amazon S3, Amazon EC2, and European Availability Zones to host its BI Factory Reporting tool, Global Blue uses AWS for hosting their corporate website, customer-facing vertical websites, and development environments.

"By moving their business intelligence (BI) application to AWS, Global Blue was able to save nearly $1M."
– Waleed Hanafi, Senior Vice President - Chief Technology Officer for Global Blue

Hitachi

Hitachi Systems & Services, a member of the Hitachi Group, has turned to Amazon Simple Storage Service (Amazon S3) to address their growing storage demands for their new, first of its kind in Japan, mobile service "Mobile Broadcast Solution." Unlike the traditional storage procurement process, Amazon S3 offered a simple, cost-effective, and fast storage solution that attracted Hitachi Systems to further pursue.

"Associated with the costs of procurement, placement, and operation within a data center, we estimate cost savings of $70,000." – Hiroshi Saijo, General Manager of the Platform Solution Division at Hitachi

Junta de Andalucía

Junta de Andalucía is the Health Department of Andalucía, Spain and provides Andalucía's citizens with improved access to their health system. Using AWS, Junta de Andalucía developed a public web portal for citizens to gain quick access to Health Department information.

"We estimate that the department's technology infrastructure cost is now 1/30th the cost of other technology infrastructure options." – Mr. Carlos González Florido, Chief Technology Officer of Iavante Foundation, the organization that is fully owned by Junta de Andalucía and provides technological services to the Health Department.

NASDAQ

NASDAQ hosts their Market Replay system on Amazon Web Services. This system allows customers on the trade support desk to validate client questions. Compliance officers use it to validate execution requirements and rate National Market System (NMS) compliance. Traders and brokers use it to look at certain points in time to view missed opportunities or, potentially, unforeseen events. The team saw that Amazon S3 would enable them to deliver hundreds of thousands of small files per day to AWS, and then back to the customer—in seconds—at a low cost.

"When considering AWS, we were able to go immediately to senior executives and sell the idea of a low-cost solution by giving them evidence. The solution cost $50 the first month, and that resonated very much with senior management. Thus, we were able to accelerate the product launch." – Jeff Kimsey, Associate Vice President of Product Management for NASDAQ OMX Global Data Products

NASA/JPL

NASA's Jet Propulsion Laboratory (JPL) has developed the All-Terrain Hex-Limbed Extra-Terrestrial Explorer (ATHLETE) robot. As part of the Desert Research and Training Studies (D-RATS), JPL performs annual field tests on the ATHLETE robot in conjunction with robots from other NASA centers. While driving the robots, operators depend on high-resolution satellite images for guidance, positioning, and situational awareness. To streamline the processing of the satellite images, JPL engineers developed an application that takes advantage of the parallel nature of the workflow. JPL relies on Amazon Web Services (AWS) for this effort.

"***The application allowed us to process nearly 200,000 Cassini images within a few hours under $200 on AWS." Due to the lack of elasticity available internally before switching to AWS, NASA explains that "We were only able to use a single machine locally and spent more than 15 days on the same task."*** – Khawaja Shams, Senior Solution Architect, NASA JPL

Newsweek

In 2009, *Newsweek* began looking for cost-cutting opportunities. The publication realized that migrating its online presence from its previous co-location facility to a cloud services provider would significantly reduce operating expenses. After exploring various options, *Newsweek* chose AWS due to its comprehensive stack of services that could meet the demands of the widely read publication. *Newsweek* expanded its AWS platform to include the Domain Name System (DNS) web service, Amazon Route 53.

"We were able to reduce our DNS costs by 93%, which in tandem allowed us to shorten our time-to-live (TTLs) for easier, timelier management of DNS records. In the cloud, IP addresses are largely ephemeral, so we needed a service that would allow us to increase the amount of DNS requests due to a shorter TTL without increasing our spend. The AWS-based infrastructure has decreased the publication's overall monthly operating costs by 75%. The publication has

also been able to streamline its system administration personnel by approximately 50%." – Nathan Butler of The Newsweek/Daily Beast Company

Pfizer

Pfizer's high performance computing (HPC) software and systems for worldwide research and development (WRD) support large-scale data analysis, research projects, clinical analytics, and modeling. Pfizer's HPC services are used across the spectrum of WRD efforts, from the deep biological understanding of disease, to the design of safe, efficacious therapeutic agents. Pfizer has set up an instance of Amazon Virtual Private Cloud (Amazon VPC) to provide a secure environment with which to carry out computations for WRD. Amazon VPC has enabled Pfizer to respond to these challenges by providing the means to compute beyond the capacity of their dedicated HPC systems, which enables them to provide answers in a timely manner

"Pfizer did not have to invest in additional hardware and software, which is only used during peak loads; that savings allowed for investments in other WRD activities. AWS enables Pfizer's WRD to explore specific difficult or deep scientific questions in a timely, scalable manner and helps Pfizer make better decisions more quickly." – Dr. Michael Miller, Head of HPC for R&D at Pfizer

Razorfish

Amazon Elastic MapReduce lets Razorfish focus on application development without having to worry about time-consuming setup, management, or tuning of Hadoop clusters or the compute capacity upon which they sit.

"With AWS, there was no upfront investment in hardware, no hardware procurement delay, and no need to hire additional operations staff." – Mark Taylor, Program Director, Razorfish

Samsung

Samsung uses Amazon EC2, Amazon RDS, Amazon S3, Amazon CloudFront, and Amazon Virtual Private Cloud to run its Smart Hub application, which supports Samsung's devices such as TVs, Blu-Ray players, tablets, and phones. The Smart Hub application authenticates devices, delivers applications and content, pushes notifications across multiple devices, and takes other actions that support the specific device. **The Smart Hub application has saved the company $34 million and reduced costs by 85%.**

"If we were to use the traditional on-premises data center, we would have spent an additional $34 million dollars in hardware and maintenance expenses during the first two years. With the AWS cloud, we met our reliability and performance objectives at a fraction of the cost." – Mr. Chun Kang, Principal Engineer, Samsung

SEGA

The SEGA Online Operations team builds and maintains Internet platforms for the company's western divisions and subsidiary studios. The team has successfully migrated its public websites to AWS, making use of Amazon EC2, Amazon S3, Amazon CloudFront, and Amazon RDS.

"SEGA reduced server costs by over 50% with On-Demand Instances when unplanned load spikes hit after game launches" – Stuart Wright, IT & Network Director for the Online Operations team

Spiegel TV

Spiegel TV is the online television news service that offers German viewers the latest, high quality programming live and on demand, 24 hours a day from anywhere in the world. Spiegel.tv is using Amazon S3 for storage, Amazon EC2 for video transcoding and Amazon CloudFront for streaming static and video files. By the end of 2011 Spiegel.tv served over 1 billion static objects over Amazon CloudFront.

"Take, for example, video transcoding. Next month, we are going to transcode more than 20,000 videos into seven high quality formats. This job is going to use approximately 40,000 high CPU hours, and we are going to transcode everything in under two days. We couldn't even pay the electricity bill for all the servers that would be required to perform this operation in our own data center." – *Nikolai* Longolius, CEO of Schnee Von Morgen, the company that managed the project

Unilever

With the help of Eagle Genomics, Unilever Research and Development created a digital data program to advance biology and informatics innovation. The program's architecture combines Amazon EC2, Amazon RDS, and Amazon S3 with the eHive open-source workflow system. Since the program started, Unilever has been able to maintain its operational costs while processing genetic sequences 20 times faster and substantially increasing simultaneous workflows.

"Unilever's digital data program now processes genetic sequences twenty times faster—without incurring higher compute costs. In addition, its robust architecture supports ten times as many scientists, all working simultaneously."
– Pete Keeley, Unilever Research's eScience IT Lead for Cloud Solutions

References

1. AWS Economics Center – http://aws.amazon.com/economics
2. Amazon EC2 Cost Comparison Calculator – http://media.amazonwebservices.com/Amazon_EC2_Cost_Comparison_Calculator_042810.xls
3. AWS Simple Monthly Calculator – http://aws.amazon.com/calculator
4. AWS Architecture Center – http://aws.amazon.com/architecture
5. AWS Free Usage Tier – http://aws.amazon.com/free
6. AWS Documentation – http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/concepts-on-demand-reserved-instances.html

Further Reading

- Web Applications Solutions Web Page – http://aws.amazon.com/web-applications/
- Digital Media Solutions Web Page – http://aws.amazon.com/digital-media/
- Whitepaper: "How AWS Pricing Works" – http://media.amazonwebservices.com/AWS_Pricing_Overview.pdf
- Whitepaper: "The Total Cost of (Non) Ownership of a NoSQL Database Service" – http://media.amazonwebservices.com/AWS_TCO_DynamoDB.pdf

Appendix A

On-Premises Cost Break Down and Assumptions

On-Premises Costs for 1 Server Amortized over a 3-year period	
Server Hardware	$51
Network Hardware	$10
Hardware Maintenance	$8
Power and Cooling	$29
Data Center Space	$24
Personnel	$200
Total Per Month	**$322**

Although there are significant one-time costs ($2,492 per server) when provisioning hardware, in this paper, we have amortized the costs monthly over a 3-year period for fair comparison across Reserved Instances, On-Demand Instances, and on-premises servers. Also, we have assumed that servers are not virtualized and have not included virtualization software licensing and management costs in order to keep the calculations relatively simple.

Our Assumptions:

1. Server Hardware
   We assumed Dell PowerEdge R310 configuration, equivalent of High-Memory Extra Large (M2.XL) Amazon EC2 Instance (see configuration and cost below). This includes on-site installation and warranty.

2. Network Hardware
   We assumed Dell PowerEdge Rack Chassis Dell PowerConnect Switches and a management switch (see configuration and cost below). This includes on-site installation and warranty. The Rack has a server density of 24 physical servers.

3. Hardware Maintenance
   We assumed 3-year Dell ProSupport and NBD On-site Service – $216 for server hardware maintenance and 3-year Dell ProSupport and NBD On-site Service – $799 for network hardware maintenance.

4. Power and Cooling
   We assumed power/cooling for 1 server, with a data center PUE of 2.5 and electricity price of $0.09 per kW hour (see Amazon EC2 Cost Comparison Calculator).

5. Data Center Space
   We assumed $23,000 per kW of redundant IT power and $300 per square foot of space divided by useful life of 15 years (see Amazon EC2 Cost Comparison Calculator).

6. Personnel
   Personnel costs include the cost of the sizable IT infrastructure teams that are needed to handle the "heavy lifting" of managing physical infrastructure:

- Hardware procurement teams are needed. These teams have to spend a lot of time evaluating hardware, negotiating contracts, holding hardware vendor meetings, managing delivery and installation, etc. It's expensive to have a staff with sufficient knowledge to do this well.

- Data center design and build teams are needed to create and maintain reliable and cost-effective facilities. These teams need to stay up-to-date on data center design and be experts in managing heterogeneous hardware and the related supply chain, dealing with legacy software, moving facilities, scaling and managing physical growth —all the things that an enterprise needs to do well if it wants to achieve low infrastructure costs.

- Operations staff is needed 24/7/365 in each facility.

- Database administration teams are needed to manage MySQL Databases. This staff is responsible for installing, patching, upgrades, migration, backups, snapshots and recovery of databases, ensuring availability, troubleshooting, and performance enhancements.

- Networking teams are needed for running a highly available network. Expertise is needed to design, debug, scale, and operate the network and deal with the external relationships necessary to have cost-effective Internet transit.

- Security personnel are needed at all phases of the design, build, and operations process.

Even though the personnel costs to support production web application projects typically involve many different people, we will use a simple ratio of servers to people in our cost models for the purpose of simplicity. We are using a total annual cost of $120,000 per person which is intended to represent a fully loaded cost (both salary and benefits), and we're assuming a 50:1 server-to-people ratio. The actual server-to-people ratio can vary a lot as it depends on a number of factors such as sophistication of automation and tools, and preference for virtualized vs. non-virtualized environment). Based on our discussions with customers, we have found that a 50:1 ratio represents a good middle point of the range of what we see. We recommend that you adjust these assumptions based on your own research and experience and include the personnel costs of all the people involved in building and managing a physical data center and not just the people who rack and stack servers (that's why we are calling the ratio "server-to-people" instead of "server-to-admin").

Server and Network Hardware Configuration (Equivalent to High Memory Extra Large Amazon EC2 Instance – m2.xlarge)

Dell PowerEdge R310

Starting Price	**$2,162**
Instant Savings	**–$324**
Subtotal	**$1,838**

Date	4/25/2012 5:56:12 PM Central Standard Time
Catalog Number	4 Retail 04

Catalog Number / Description	Product Code	Qty	SKU	Id
PowerEdge R310: PowerEdge R310 Chassis, Up to 4 Cabled Hard Drives and Quad Pack LED diagnostics	R310C	1	[224-8311]	1
Ship Group: Shipping for PowerEdge R310	SHIPGRP	1	[330-8208]	2
Processor: Intel® Xeon® L3406 2.26 GHz, 4M Cache, Turbo, HT	L3406	1	[317-4054][330-8207]	6
Memory: 16GB Memory (4x4 GB), 1333MHz, Dual Ranked UDIMM	164U3D	1	[317-2022][317-2409]	3
Operating System: No Operating System	NOOS	1	[420-6320]	11
Hard Drive Configuration: No RAID - Onboard SATA, 1–4 Hard Drives connected to onboard SATA Controller	OBS14HD	1	[330-8157]	27
Hard Drives (Multi-Select): 500 GB 7.2K RPM SATA 3.5in Cabled Hard Drive	500S35C	1	[341-9209]	1209
Internal Controller: No Controller	NCTRLR	1	[341-3933]	9
Power Supply: Power Supply, Non-Redundant, 350W	NRPS	1	[330-8210]	36
Power Cords: Power Cord, NEMA 5-15P to C13, wall plug, 10 feet	WP10F	1	[330-5113]	38
Embedded Management: Baseboard Management Controller	BMC	1	[313-7919]	14
Network Adapter: On-Board Dual Gigabit Network Adapter	OBNIC	1	[430-2008]	13
Rails: 2/4 -Static Post Static Rails	STATIC	1	[330-4138]	28
Bezel: No Bezel	NOBEZEL	1	[313-0869]	17
Internal Optical Drive: DVD-ROM Drive, SATA	DVD	1	[313-9126][330-8866]	16
System Documentation: Electronic System Documentation and OpenManage DVD Kit	EDOCS	1	[330-8869]	21
Primary Hard Drive: HD Multi-Select	HDMULTI	1	[341-4158]	8
Asset Tag on System Chassis (CFI): Basic Support Label (Bill To Company Name)	BASWC	1	[365-0529]	352

Hardware Support Services: 3 Yr. Basic HW Warranty Repair with SATA Ext: 5x10 HW-Only, 5x10 NBD Onsite	Q3OSSX	1	[909-4347][909-4488][923-8249][923-8952][927-3190][993-9412][994-4500]	29
Installation Services: ONSITE INSTALLATION: PowerEdge Hardware Install only	UMOUNT	1	[985-0937]	32
Keep Your Hard Drive: Keep Your Hard Drive, 3 Years	KYHD3Y	1	[983-6402]	159
Proactive Maintenance: MAINTENANCE DECLINED	NOMAINT	1	[926-2979]	33

Dell PowerConnect 6224

Subtotal: $2991

Date	4/25/2012 6:01:44 PM Central Standard Time

Catalog Number / Description	Product Code	Qty	SKU	Id
PowerConnect 6224: PowerConnect 6224, 24 GbE Ports, Managed Switch, 10GbE and Stacking Capable	PC6224	1	[222-6710]	1
Modular Upgrade Bay 1: Modules: PowerConnect 6xxx SFP+ Module supports up to two SFPs (no SFPs included)	PC2SFP	1	[330-2467]	182
Modular Upgrade Bay 1: Optics: Two POWERCONNECT 6xxx Short Range, Multi-Mode SFP+ Optics	28024SS	1	[330-2405][330-2405]	187
Hardware Support Services: Lifetime Limited Hardware Warranty with Basic Hardware Service Next Business Day Parts Only	PD	1	[934-7080][981-0890][985-5977]	29
Installation Services: ONSITE INSTALLATION: Power Connect Hardware Installation only	QMOUNT	1	[989-6188]	32

Dell PowerEdge Rack 4220

TOTAL: $2,252

Date	4/25/2012 6:05:08 PM Central Standard Time

Catalog Number / Description	Product Code	Qty	SKU	Id
PowerEdge Rack 4220: Dell 4220 42U Rack with Doors and Side Panels, Ground Ship, NOT for AK / HI	42GFDS	1	[224-4934]	1
Dell PDU and Accessories: PDU,30A,208V,(21)C13,(6)C19,Vertical, with L6-30P 3m attached cord	P30A208	1	[331-0017]	1232
Hardware Support Services: 3 Yr. Basic Hardware Warranty Repair: 5x10 HW-Only, 5x10 NBD Parts	3PD	1	[992-1802][992-5080][993-4108][993-4117]	29
Installation Services: Rack Installation, QLX	QINSTL	1	[980-7677]	32