Cost Optimization with AWS

Architecture, Tools, and Best Practices

February 2016

Notices

Contents

Abstract 3

Introduction 4

Cost Optimization 4

Design Principles 5

Matching Supply and Demand 6

- Compute 6
- Storage 10
- Application Services 12

Cost-Effective Resources 17

- A Granular Approach 17

Expenditure Awareness 18

- AWS Price List API 19
- Detailed Billing Reports 19

Optimizing Over Time 20

Conclusion 20

Contributors 20

Appendix: Cost-Effective Storage Example 21

Further Reading 22

Abstract

The focus of this paper is the Cost Optimization pillar of the Well-Architected Framework. It provides guidance to help customers apply best practices in the design, delivery, and maintenance of AWS environments.

Introduction

At AWS, we understand the value of educating our customers about architectural best practices for designing reliable, secure, efficient, and cost-effective systems in the cloud. As part of this effort, we developed the AWS Well-Architected Framework, which helps you understand the pros and cons of decisions you make while building systems on AWS. We believe well-architected systems greatly increase the likelihood of business success.

The framework is based on four pillars:

- Security
- Reliability
- Performance Efficiency
- Cost Optimization

The Cost Optimization pillar helps you avoid or eliminate unnecessary costs or suboptimal resources and use the savings on differentiated benefits for your business. A cost-optimized system allows you to pay the best price possible while still achieving your business objectives and meeting or exceeding requirements for the other pillars in the framework. This paper will provide in-depth, best-practice guidance for selecting the appropriate architecture and maximizing your AWS resources as efficiently as possible.

Cost Optimization

Business problems can be operational in nature (having support teams on call), wasteful (overprovisioning hardware resources), vulnerable to failure (tightly coupled deployments), and expensive (running a data center, employing security staff, and so on). AWS offers solutions to these problems, but it’s important to understand and implement services with cost optimization in mind.

At AWS, we think about four areas of cost optimization:

- Matched supply and demand
- Cost-effective resources

- Expenditure awareness
- Optimizing over time

Along with improved security or speed to market, cost and return on investment (ROI) are often determining factors in a purchasing decision. AWS has several tools, such as AWS Trusted Advisor, detailed billing reports, and the Total Cost of Ownership Calculator, to help you understand your bill, calculate your cost of ownership, and identify areas for improvement. There are also AWS case studies and references that can serve as quantitative support to aid your decision-making process.

Design Principles

Keep these design principles in mind as we discuss best practices in the four areas of cost optimization:

- **Move from a forecast-procurement model to a consumption model:** Instead of investing heavily in data centers and servers based on historical data or pro-forma forecasts, pay only for the computing resources you consume and increase or decrease usage depending on need, not elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can stop these resources when they are not in use for a potential cost savings of 75% (40 hours versus 168 hours).
- **Benefit from economies of scale:** Hundreds of thousands of customers are aggregated in the AWS cloud, which translates into lower pay-as-you-go prices.
- **Stop spending money on data center operations:** Cloud computing vendors do the heavy lifting of racking, stacking, and powering servers, so you can focus on your customers and core business rather than on IT infrastructure.
- **Transparently attribute expenditure:** The cloud makes it easier to identify the cost of a system and attribute IT costs to individual business owners. This helps measure ROI and gives those owners an incentive to optimize their resources and reduce costs.
- **Use managed services to reduce cost of ownership:** In the cloud, managed services remove the operational burden of maintaining servers for tasks like sending email or managing databases. And because managed

services operate at cloud scale, they can offer a lower cost per transaction or service.

- **Continuously reevaluate design choices**: Unlike traditional IT infrastructure approaches where you are required to make large capital investments in hardware and software, AWS offers pay-as-you-go pricing for most of its services. This means you are not bound by decisions made at a design level at the beginning of a project's lifecycle. This reduces the risks of overprovisioning or not being able to meet unexpected demand. You can continually reevaluate your design decisions. You can also explore the use of new AWS products to see if they lead to even greater efficiencies.

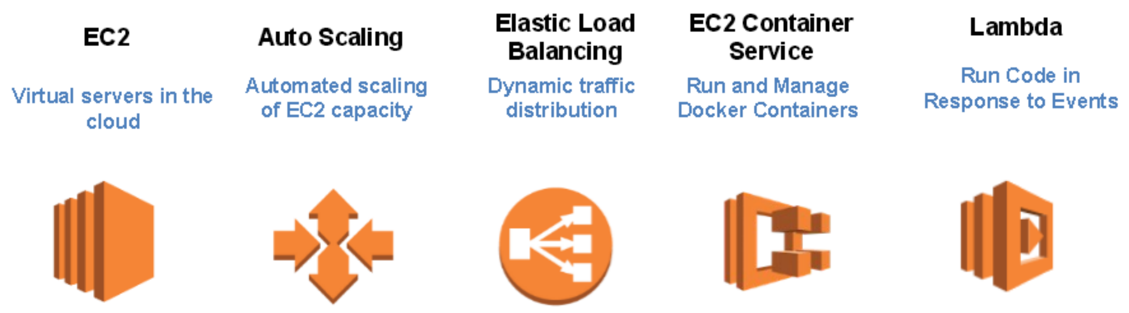
Matching Supply and Demand

Unlike traditional IT infrastructure models, AWS is, by its very nature, elastic and on-demand. AWS provides mechanisms to programmatically scale up and down, as needed, or to implement lifecycle rules that archive or expire storage objects automatically. Consider these features and services to help you achieve a cost-optimized architecture.

Compute

In the AWS ecosystem, compute begins with Amazon Elastic Compute Cloud (Amazon EC2). At its most basic level, EC2 is virtual machines in the cloud.

Amazon EC2 has evolved since 2006 and, as shown in the following diagram, led to other services, such as Auto Scaling and Amazon EC2 Container Service (for increased efficiency), Elastic Load Balancing (for resilience), and AWS Lambda (for server-less computing).


EC2
Virtual servers in the cloud
Auto Scaling
Automated scaling of EC2 capacity
Elastic Load Balancing
Dynamic traffic distribution
EC2 Container Service
Run and Manage Docker Containers
Lambda
Run Code in Response to Events

Amazon EC2

Cost optimization can only be successful if you are able to match the requirements of your application or workload with the most appropriate instance type. The Amazon EC2 instance types and sizes[1] offer varying combinations of CPU, memory, storage, and network capacity to give you the flexibility to choose the right mix of resources for your application. For example, C4 instances are suited to workloads that require heavy computational resources. M4 instances are a multi-role instance type. G2 instances are aimed at workloads that require dedicated graphic processors. Choosing the right instance type will help you optimize performance in a cost-effective way.

After you understand your workload requirements, consider taking advantage of Reserved Instances and Spot Instances, which can result in a 30% or up to 90% reduction, respectively, in your EC2 spending.

Benchmarking

AWS recommends you select an instance type from the instance family that most closely matches your workload, and then start a period of benchmarking. Because you can change the instance type, it's possible to perform benchmarking against each type. You can also do this when AWS releases new instance types.

Reserved Instances

After you have settled on an instance type, you have the option of purchasing a Reserved Instance. This is an upfront commitment to purchase capacity in a particular AWS region, which will dramatically reduce your running costs.

A Reserved Instance is a billing construct; it ensures you have capacity available in the Availability Zones[2] (AZ) you have selected and purchased for that instance type, and will significantly reduce your hourly rate. Besides treating a Reserved Instance as a 24x7 resource, it is also possible for you to combine a Reserved Instance if your workload is time-dependent.

For example, let's assume you have a Reserved Instance for a multi-purpose instance type like an m4.large. You only need to run this instance during office

[1] Each instance type has a different running cost (per hour). Typically, the larger the instance type, the more expensive.

[2] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-regions-availability-zones

hours for a total of nine hours (8:00 A.M.-5:00 P.M.). However, you have another workload in the same Availability Zone that can use the same instance type and be run after office hours (5:00 P.M.-8A.M.). You could select the same instance type (m4.large) and start the evening workload on that instance after the daytime instance has shut down.

After the first instance is shut down, the Reserved Instance hourly rate will apply to the after-hours instance, thus maximizing your overall cost efficiency.

It's important to continually reevaluate your instance selection, because workloads and instance types will change over time. Reserved Instances are currently offered as one- or three-year commitments[3], and your requirements may change before the Reserved Instance commitment expires. If this should happen, you could use EC2 Container Service (Amazon ECS) to increase instance usage, or you could sell your Reserved Instances in the AWS Reserved Instance Marketplace.

Auto Scaling

Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. For example, traditional IT deployments require you to determine your hardware requirements based on peak usage. If your compute resources are determined by the highest predictable period of load and do not allow you to scale down, you are wasting compute resources when the load is at its lowest. Auto Scaling optimizes efficiency and costs by increasing or reducing the number of instances as demand for resources change.

If your workload runs during business hours (9:00 A.M. - 5:00 P.M.), you can configure Auto Scaling to launch instances appropriate for the predicated load. After business hours, Auto Scaling can then reduce the number of instances, minimizing expenditure on the per-hour running costs.

Spot Instances

Spot Instances allow our customers to bid on spare Amazon EC2 computing capacity. Because Spot Instances are often available at a discount compared to

[3] https://aws.amazon.com/ec2/purchasing-options/reserved-instances/

On-Demand pricing, you can significantly reduce the cost of running your applications and grow your application's compute capacity and throughput.

Leveraging the Spot market[4] can help reduce operating costs for workloads that do not require a persistent state. Workloads that lend themselves to this type of model include Amazon Elastic MapReduce (Amazon EMR) clusters. You can use Spot Instances to augment an on-demand cluster with additional core nodes and reduce the time required to complete Amazon EMR Hadoop jobs.

If the bid price exceeds your maximum bid, then you are notified the core nodes that are running as Spot Instances are scheduled to be terminated. When this happens, Hadoop will restart the job running on the remaining available instances in the cluster automatically.

Another example of a non-persistent use case for Spot Instances is a batch-processing task. These jobs can often continue processing with interruption, if such logic is included. One nightly processing job can begin when Spot Instances with the right bid price are available. Should those instances be terminated, the task can resume when Spot Instances are available again.

Amazon ECS

Amazon EC2 Container Service (Amazon ECS) provides a platform for EC2 instances to host Docker containers. This increases the level of granularity in your workloads and the efficiency of your EC2 instances. This is particularly useful when you have large instance types that are capable of hosting several containers instead of running applications or infrastructure components on multiple EC2 instances.

From a cost-optimization perspective, Amazon ECS can be used with containers on underutilized Reserved Instances as well.

Let's assume you use an m4.xlarge Reserved Instance, but are only using 5% of your CPU resources. This is a good use case for Amazon ECS because you could add several containers to the instance and increase the utilization rate to a more efficient level.

[4] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instances-work.html

Amazon ECS can even be used to place containers on instances according to a schedule. To expand on the Reserved Instance example, rather than power down the original instance, you could schedule containers to be placed on the instance from 6:00 P.M. until 8:00 A.M. and use it at the Reserved Instance hourly rate.

AWS Lambda

Running always-on instances can be wasteful, especially when a service is waiting on something to do. For example, if you created a photo processing application, traditionally you would have a listener or polling service look for incoming jobs, and then start your processing code when something was uploaded. You would still need to run a server (instance) and determine and then maintain a number of instances to provide the compute platform for the application code. With Lambda, you just write the code and set up the trigger event that will start it.

Lambda is server-less computing; the Lambda service will provide the compute platform for your code and trigger an event based on parameters you provide. This eliminates the need for your own running instance and the administrative overheard of the operating system.

Lambda is an appropriate choice for workloads that respond to events, such as uploads or even request to a service that can be decoupled loosely. Lambda is also granular, because it is metered per 100 milliseconds.

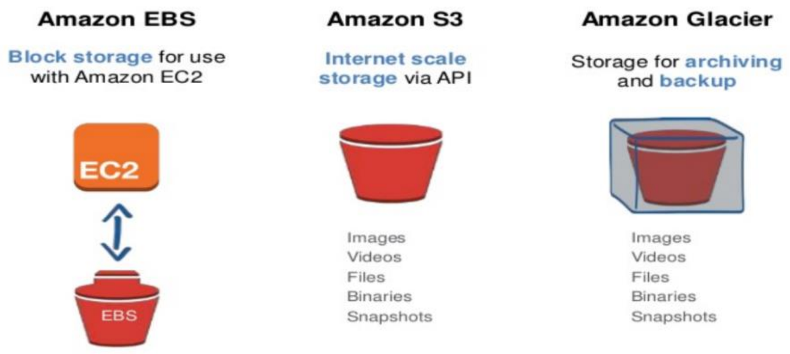
You can achieve cost optimization through architectural planning and continuous revision and assessment. The use of AWS services, from virtualized instances to server-less computing, means no design decision needs to be permanent. In addition, you can assess new products and services as they're released to continuously optimize your deployment.

Storage

When considering a storage solution, these elements are especially important:

- Durability
- Availability
- Regulatory and governance requirements
- Security

- Functional requirements



AWS storage solutions can meet different technical and price requirements. From a cost-optimization perspective, the lowest compute-related cost storage option is locally attached, ephemeral storage[5]. This is storage included in the run rate for EC2 instances. The local processing nature of ephemeral storage isn't for long-term, persistent data storage. For more persistent storage, Amazon Elastic Block Store[6] (Amazon EBS) enables EC2 instances to write to block-level devices that can be migrated to different instances. It also provides snapshot functionality to back up and migrate these volumes.

There are two storage types available for EC2 instances: non-persistent or ephemeral storage, which is appropriate for log files and non-critical data, and persistent storage for applications and workloads that require persistency.

Each option has pros and cons in terms of performance and costs. Ephemeral storage is included in the EC2 instance hourly rate. Amazon EBS comes at additional cost. You should consider these pros and cons per workload during your initial assessment, and then again after the workload has been running for a time. That way, you can see if you're using the resources you've allocated and

[5] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html

[6] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html

make adjustments, if necessary. For example, among the Amazon EBS volume types[7], you could switch from Provisioned IOPS to GP2 instances if you find that your workload is more efficiently run on a "bursting" level of IOPS. AWS makes volume information available to customers through Trusted Advisor, which will highlight unattached Amazon EBS volumes and other important deployment information. In this case, with a proper archive strategy, deleting unattached Amazon EBS volumes will minimize storage costs.

Amazon S3

Accessible from anywhere in the world with an internet connection and providing 99.999999999% durability, Amazon S3 has no limits on the amount of storage that can be used and provides mechanisms for customers to delete or archive (to Amazon Glacier) data no longer needed in quick-access storage. Using Amazon S3 lifecycle policies will enable you to migrate data from Amazon S3 to Amazon Glacier or delete it entirely. Amazon S3 also provides customers with storage classes[8] with different levels of durability, availability, and pricing. These other classes can be useful when your storage mechanism does not require 99.999999999% durability.

Amazon Glacier

You can use Amazon Glacier to address your archiving requirements and improve your cost optimization. Amazon Glacier is, by design, cold storage. It should be used for data you need to retain, but do not need to frequently access. Although S3-to-Amazon Glacier migrations can be seamless, many small, frequent retrievals from Amazon Glacier will cause you to incur higher costs. For this reason, you should carefully consider a retrieval process.

Application Services

Application services are products that remove a lot of the management and undifferentiated administration tasks associated with workload deployments to enable our customers to focus on their core business. For example, Amazon Relational Database Service (Amazon RDS) and Amazon DynamoDB are managed database services. The DynamoDB service will provision and maintain the database instance required to provide your NoSQL solution.

[7] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

[8] http://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html

Although the savings aren't reflected in your monthly AWS bill, these managed services free up your DBAs and administrators to focus on your core business and development, like backup and recovery.

Of course, it's possible to run these applications or databases directly on EC2 instances. However, if you consider that building a database or Hadoop framework on EC2 will require your team to configure, administer, and maintain your deployments, you'll see how an application service like Amazon EMR or DynamoDB saves time and money.

Database and analytics products like Amazon RDS and Amazon EMR and other services like Amazon Elastic Transcoder and Amazon Workspaces can help you minimize undifferentiated administrative activities, such as automating database backup, initialization and ongoing management of the environment, and automatic host replacement. The time and efficiency gained from the use of these services can deliver intangible financial benefits, such as speed to market.

Data Transfer

Data transfer in and out refers to transfer into and out of the AWS regions. AWS offers tiered pricing for data transfer from AWS to the Internet (the more you use, the lower the rate you pay). To help reduce data transfer costs, you can architect parts of your infrastructure to use AWS services like AWS CloudFront[9] and AWS Direct Connect[10]. For example, if your deployment is a hybrid model that contains on-premises data centers and AWS, you can leverage AWS Direct Connect to establish a private fiber link, instead of the Internet, between your AWS resources and your data center. Data traffic that is destined to your on-premises data centers is considered data-transfer-out and it results in a charge to your monthly bill. Customers can eliminate these charges by using AWS Direct Connect, because costs associated with data traffic through this private connection are included in the service. Weighing your monthly data-transfer-out related charges against the cost of AWS Direct Connect will help you make the right financial choice.

[9] https://aws.amazon.com/cloudfront/details/

[10] https://aws.amazon.com/directconnect/details/

Likewise, if you use Amazon S3 for content delivery, using Amazon CloudFront will reduce the latency between your end users and your S3 bucket. For example, a transfer from an S3 bucket in the US West (Oregon) region to the Internet would cost $0.090 per GB, up to the first 10 TB per month. If you deployed an Amazon CloudFront solution, the transfer rate would drop to $0.085 per GB while reducing overall latency.

By continuously assessing your AWS architecture and understanding where your data transfer costs are incurred, you can adapt your deployment to reduce overall running costs while improving the end-user experience.

Operational Techniques

Automation tools can help minimize some of the management and administrative tasks associated with an IT deployment. Like the benefits from application services, an automated or DevOps approach to your AWS infrastructure will provide scalability and elasticity with minimal manual intervention. This also provides a level of control over your AWS environment and the associated spending. For example, when engineers or developers are allowed to provision AWS resources only through an established process and tools (for example, a provisioning portal such as AWS Service Catalog[11]) that can be managed and audited you can avoid the expense and waste that results from simply turning on (and most often leaving on) standalone resources.

AWS is constantly developing tools that can be used to manage AWS resources. Services like AWS CloudFormation, AWS Elastic Beanstalk, AWS CodeCommit, AWS CodeDeploy, and AWS CodePipeline can help reduce administration overhead and thus development time. Your teams aren't provisioning racks, managing cables, and raising purchase orders, but are instead focusing on product development. Combining these services with Auto Scaling, Spot Instances, and lifecycle policies will improve efficiency and consistency.

Tagging

A tag is a label you assign to an AWS resource. Each tag consists of a key and value, both of which you define. AWS uses tags as a mechanism to organize your resource costs on your cost-allocation reports. For example, tagging your EC2

[11] https://aws.amazon.com/servicecatalog/

instances helps you identify resources that belong to different projects, departments, or owners. Tags can also contain other information that's helpful to any reconciliation, analysis, or internal chargeback model.

Amazon CloudWatch

With Amazon CloudWatch, you can monitor metrics for CPU utilization, data transfer, and disk usage activity from your EC2 instances. You can also use Amazon CloudWatch to monitor metrics on Amazon DynamoDB tables, Amazon EBS volumes, Amazon RDS database instances, Amazon EMR job flows, Elastic Load Balancing load balancers, Amazon SQS queues, and Amazon Simple Notification Service (Amazon SNS) topics. Amazon CloudWatch is included when you use these services.

On the other hand, you can create your own custom metrics, which are generated by your applications through a simple API request and then monitored by Amazon CloudWatch. These metrics can be useful for understanding how your workloads on AWS are performing so you can adjust resource allocation, if necessary.

You can also integrate log files of your existing systems and applications. These log files can be sent to Amazon CloudWatch Logs for you to monitor in almost real time.

Amazon CloudWatch Alarms

It is helpful to use Amazon CloudWatch metrics to alarm and notify you of under- or overutilization. When you set alarms on any of your metrics, you can configure the service to send you notifications or take other actions (for example, use Auto Scaling to add or remove instances).

It is also possible to use Amazon CloudWatch to directly monitor your estimated monthly AWS charges. Amazon CloudWatch receives estimated charges several times a day for each AWS service you're using. You set a threshold and when you meet it, an alert can be sent to you in email. For example, if you set a threshold of $1,000 per month for EC2 usage, and your usage exceeds that amount, you will receive an email alert through Amazon SNS.

Each AWS account receives 10 Amazon CloudWatch alarms and up to 1,000 Amazon SNS email notifications a month, making this a cost-effective way to proactively monitor your AWS usage.

Another feature that can help you in your cost-optimization efforts is setting usage budgets to help plan your costs (or "spend data"). Budgets are updated every 24 hours. They track your unblended costs and subscriptions, but not refunds.

Budgets can be created for different types of costs (for example, to see how much you are spending on a service or how often you call a particular API operation[12]). You can also use Amazon CloudWatch alarms on your budgets to trigger alerts from Amazon SNS.

AWS Service Catalog

As your AWS deployments mature, you can take advantage of AWS Service Catalog, a personalized self-service portal that can be used by the end users in your organization to browse and launch services. Your IT administrators can work with development teams to provide Amazon CloudFormation templates that can be used to launch approved AWS configurations for each service.

The cost optimization benefit is that the environment can use parameters (such as the instance type, storage type, and so on) specified by your policy. This controlled approach allows you to define budgets and forecast costs based on the design choices and options available to your end users. By collaborating with your technical users to define the right templates to include in AWS Service Catalog, you can prevent unnecessary resources from being launched (and often left on).

AWS Config

Keeping track of your AWS deployment and infrastructure is critical to managing costs and ensuring your environment is well-architected and cost-optimized. Although AWS services can be integrated with number of third-party configuration management tools, there is also a fully managed service, AWS

[12] http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/budgets-managing-costs.html

Config, that provides you with an AWS resource inventory, configuration history, and configuration change notifications.

An AWS Config rule represents your desired configuration settings for specific AWS resources or an entire AWS account. You can implement AWS Config rules to meet internal compliance policies and regulatory standards. You can also assess the impact on other resources in your infrastructure and whether configuration changes are in compliance with rules you defined.

Cost-Effective Resources

Selecting the right solution for your workload is what will make your deployment cost-efficient. For example, after you have completed your initial benchmarking, decide on an instance that offers you solid performance and pricing. At this point, you might consider purchasing Reserved Instances. A one-year or three-year commitment will significantly lower your overall running costs for those instances. A tool like Trusted Advisor will help you quickly identify the potential monthly savings from the purchase of a Reserved Instance. Trusted Advisor examines common metrics, such as EC2 instance utilization, idle load balancers and Amazon RDS database instances, underutilized Amazon EBS volumes, and unassociated elastic IP addresses. Trusted Advisor has sent over 2.6 million usage notifications to customers. To date, it is estimated that Trusted Advisor has helped customers save over $350 million.

A Granular Approach

AWS pricing makes it possible to produce a breakdown of resources per workload. You are no longer constrained by a one-off purchase at the start of a project based on how many workload components can be hosted on a physical server. You are now able to decouple the differing elements of each application and drill into how many of the resources provisioned for your workloads are being used by the components. A truly cost-effective deployment is one in which you continue to examine design decisions and assess resource usage.

Consider an application that requires an Amazon EMR job to complete. In benchmark testing, you see the job can be complete on a smaller instance type in five hours. On a larger instance type, the job can be complete in just one hour.

Although the larger instance is twice the price per hour, it will complete the job in less time, and therefore lower overall running costs.

In this example, you might even consider using a Spot Instance to further lower your running costs. If you bid at a price lower than the current per-hour charge, you can realize even more savings.

Expenditure Awareness

The use of tools such as AWS Config, Amazon CloudWatch, AWS OpsWorks, Auto Scaling, and AWS CloudFormation will help ensure your resources are being used efficiently, but cost optimization is achieved through a continuous cycle of assessment, benchmarking, and integration with Operations.

1. **Initial assessment**
   Which projects can be migrated to AWS? Which instance types and storage options will meet the project requirements and budget?
2. **Benchmarking**
   Is the system performing well? If not, make changes to the architecture (for example, adjust instance sizes, choose SSD devices, and so on).
3. **Integration with Operations**
   Augment or replace existing operational processes with AWS solutions:
   a. Use AWS OpsWorks, AWS CloudFormation, and Auto Scaling to automate and enable resource elasticity.
   b. Determine whether Spot Instances or Reserved Instances are appropriate at this stage.
   c. Set up Amazon CloudWatch alarms and notifications to give you immediate insight into your expenditure and, where appropriate, define actions to be taken automatically.

As your deployment grows and you migrate or create projects and services on AWS, you can increase your level of awareness through the use of tools like AWS Config. You can also use the AWS Simple Monthly Calculator to get working estimates for AWS expenditures.

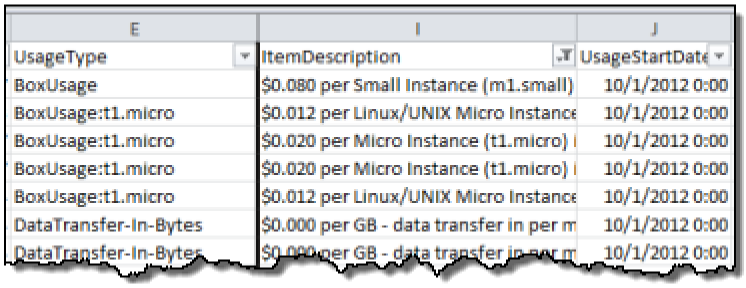
AWS Price List API

The AWS Price List API enables you to query for the prices of AWS services. You can also subscribe to Amazon SNS notifications to be notified when prices for the services change. AWS prices change periodically (for example, when new instance types are launched or when new services are introduced).

Detailed Billing Reports

The detailed billing reports include hourly line items, which make it possible for you to predict, for example, if you have enough Reserved Instances to cover your requirements. Used in conjunction with the Consolidated Billing feature, these reports simplify the tracking and management of costs associated with Reserved Instances.

As Jeff Barr pointed out in his blog post, detailed reports "provide an additional allocation model for linked accounts, with two key features — RI Affinity and Unblended Rates." For more information, see the blog post.



E	I	J
UsageType	ItemDescription	UsageStartDate
BoxUsage	$0.080 per Small Instance (m1.small)	10/1/2012 0:00
BoxUsage:t1.micro	$0.012 per Linux/UNIX Micro Instance	10/1/2012 0:00
BoxUsage:t1.micro	$0.020 per Micro Instance (t1.micro)	10/1/2012 0:00
BoxUsage:t1.micro	$0.020 per Micro Instance (t1.micro)	10/1/2012 0:00
BoxUsage:t1.micro	$0.012 per Linux/UNIX Micro Instance	10/1/2012 0:00
DataTransfer-In-Bytes	$0.000 per GB - data transfer in per m	10/1/2012 0:00
DataTransfer-In-Bytes	$0.000 per GB - data transfer in per m	10/1/2012 0:00

Detailed billing reports can also be customized to show your estimated costs by:

- hour, day, or month
- each account in your organization
- product or product resource
- tags you define

You can also create tags for your AWS resources to add your own labels to nearly every line item in your reports. You can use the billing reports to do any of the following:

- Bring your billing data into an application that can read the data.
- Build an application that uses your billing data.
- Monitor your month-to-date charges.
- Forecast your monthly charges.
- Share data with a partner.
- Import your billing data into your accounting system.
- Retrieve your bill for multiple accounts.

This customization allows you to generate reports that are meaningful to your organization. This data can be ingested into a data warehousing solution like Amazon Redshift for more analysis, reporting, and forecasting.

Optimizing Over Time

An initial assessment will never be 100% accurate, but the inherent flexibility of AWS services means you are not locked into initial design choices. Continuous reassessment and monitoring of your AWS resources is vital because requirements will change over time.

AWS will also release new features and services that can reduce costs and increase overall efficiency. For example, Amazon Redshift, a low-cost, scalable data warehousing solution, has reduced the cost of entry into data warehousing. Amazon Aurora is a fully managed, MySQL-compatible, relational database engine that combines the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases.

Conclusion

Cost optimization is an ongoing effort. From the first assessment of a pilot or test workload to mature AWS infrastructures, administrators should reassess and test their architectural approach. This effort is easier thanks to the programmatic functions and access and AWS features and services discussed in this paper.

AWS strives to help you maximize efficiency while you build highly resilient, responsive, and adaptive deployments. To make your deployment truly cost-optimized, you should use as many of these tools and techniques discussed in this paper.

Contributors

Bryan Tang and Callum Hughes

Appendix: Cost-Effective Storage Example

The critical question for compute-related storage is whether instance storage can satisfy the requirements of your workload, or if, for example, Amazon EBS might be required instead.

The following example uses an m3.xlarge EC2 instance in the US East (N. Virginia) region and assumes it is running 24x7 for a month:

1. EC2 only cost: $194.72/month
2. EC2 plus 500 GB EBS General Purpose (SSD) volume: $194.72 + $50 = $244.72/month
3. EC2 plus 500 GB EBS Provisioned IOPS (SSD) volume with 3000 IOPS: $194.72 + $257.50 = $452.22/month

In the first scenario, the storage requirements are met simply by leveraging instance storage. In the other two scenarios where the workload requires persistent data storage, Amazon EBS has been added.

We encourage AWS customers to consider if a consistently high number of IOPS is necessary. As shown in the third scenario, the performance advantage of PIOPS adds to the monthly cost. From a cost-optimization perspective, you should investigate if the PIOPS you set is being fully used. If not, consider lowering the PIOPS or move to the burstable model of EBS General Purpose (SSD).

AWS provides options so you can find the right balance between cost and performance, but it is important to confirm your choice through monitoring.

Given the technical benefits and low monthly rate of Amazon S3, some AWS customers keep all of their data in S3. For customers storing tens of millions of files on S3, an archival strategy will have a positive impact on storage costs. We believe it is always a good idea to implement an archival strategy, regardless of current usage, so that the process of archiving is embedded as your growth continues.

The following example assumes there is 10 TB of data for storage and archival in the US East (N. Virginia) region.

1. S3: $302.45/month
2. S3 Standard -Infrequent Access (Standard IA): $128.00/month
3. S3 Reduced Redundancy Storage (RRS): $242.08/month
4. Amazon Glacier: $71.68/month

Amazon S3 offers three storage classes with different durability and availability. However, for data that is rarely accessed, but required due to regulatory or other requirements, Amazon Glacier is an attractive option. We recommend you evaluate your data retention requirements and take advantage of Amazon S3 lifecycle management so that data is moved automatically to a less expensive storage class on a schedule you define. Based on these monthly cost estimates, you can see the difference between storing objects in Amazon S3 for 7 years versus storing them for 1 year in Amazon S3 and then archiving them for 6 years in Amazon Glacier.

Further Reading

For additional help, please consult the following sources:

- AWS Well-Architected Framework[1]