Understanding and Managing Costs in the AWS Cloud

NASA's Earth Observing System Data and Information System (EOSDIS) is evolving to implement a commercial cloud-based solution to ingest, archive, process, distribute, and manage the anticipated large volumes of new mission data. Placing the EOSDIS archive collectively in the cloud will, for the first time, place NASA Earth Observing data "close to compute" and improve management and accessibility of these data while also expediting science discovery for data users.

Having EOSDIS data in the cloud will not change the existing user experience of interacting with these data, but it will offer new methods of access not otherwise possible with on-premises infrastructure. See the Why Use the Cloud - A Getting Started Guide for more background on NASA's evolution and efforts to move to a cloud-based Big Data solution.

Under NASA's full and open data policy, all NASA mission data (along with the algorithms, metadata, and documentation associated with these data) must be **freely available and provided to the public as soon as possible** following a checkout period to ensure data accuracy and validity. As such, **all NASA data will continue to be free to access and download, regardless of whether the user is working in the cloud or not**.

NASA's Office of the Chief Information Officer has chosen Amazon Web Services (AWS) as the source of general-purpose cloud services for NASA, including ingest, archive, processing, distribution, and management of data. In addition, NASA EOSDIS is working with Google Earth Engine (GEE) to make NASA data accessible in the GEE cloud-based analysis platform.

The AWS cloud platform uses Elastic Compute Cloud (EC2) instances, which are virtual computers that a user creates to perform processing operations in place of using a desktop or laptop computer or other on-premises computing hardware. They are customizable based on user computing needs and consist of the user's choice of operating system, type of central processing unit (CPU), amount of memory and storage; and the security settings that determine how the instance can be accessed. Some of these customizations (e.g., CPU, memory, storage) can impact the user's cost of using the cloud.

This introductory tutorial for understanding and estimating cost in the cloud:

1. provides an overview of the cost model for NASA EOSDIS - cloud data access and use
2. highlights several relevant user scenarios using the AWS Pricing Calculator to provide guiding examples and consideration for Earth Observing System (EOS) data users in the cloud, and
3. summarizes AWS resources for budgeting cost in the cloud.

Cost in the cloud

Cost model

Figure 2.1 shows a general high-level user workflow diagram, giving a sense for the data management and cost model of discovering, accessing and using NASA EOSDIS data in the cloud. **All NASA data is free to access and download, regardless of whether the user is working in the cloud.**

Cloud technology is evolving so fast that it is likely that some details in the primer may no longer match reality when you are trying to use it. If you find mismatches (e.g. broken third-party links), please send them to support@earthdata.gov so that we can feed them into the next release of the primer.

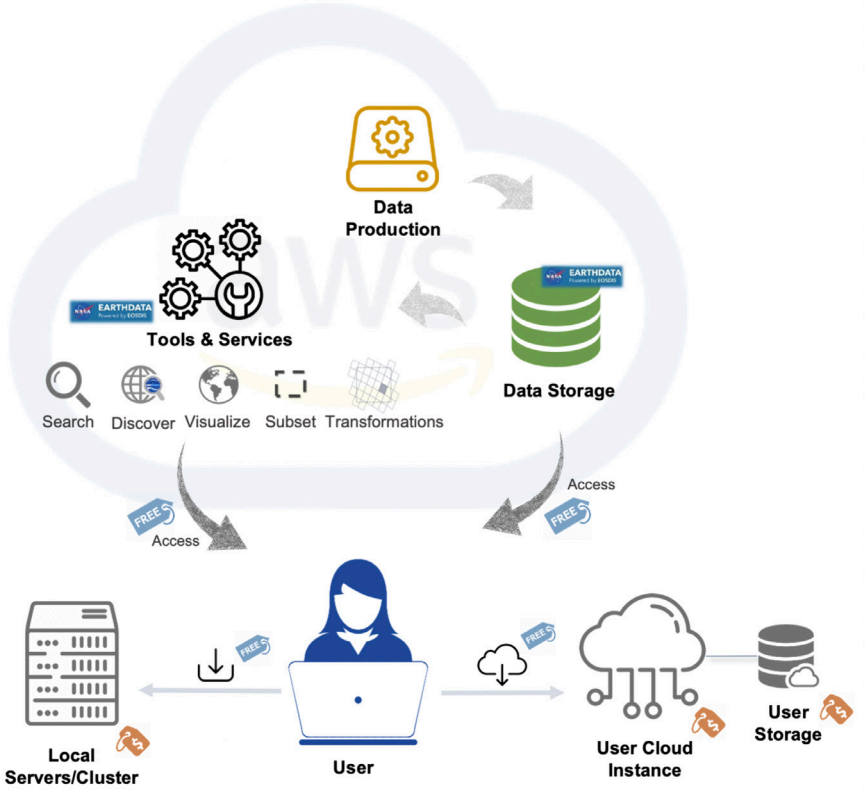


Figure 2.1: Cloud paradigm workflow and cost diagram (as of July 2020).

A user will be able to employ these cloud-based EOSDIS-provided services to discover, search, access, and download data, at no cost to the user. The user has the option to:

1. download the data to their own local machine, server, or HPC account - free of charge
2. connect to the Earthdata data from their (AWS) cloud instance - free of charge, or
3. move the data out of the AWS-hosted cloud (free of charge) to another cloud platform of choice.

If a user wishes to store data in their own (AWS) cloud instance or cloud storage, the user is responsible for that storage cost.

User Scenarios

The following scenarios provide select examples of representative but varying Earth science user workflows. It can be a quick reference guide for the user to get a sense of how storage amount, compute power, memory, frequency of use, pricing strategy, etc. might affect total cost.

User Scenarios descriptions:

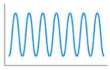
1. **Climate Researcher** (large data volume): This user has large data volume (e.g., over 20 years of global ocean data) and needs to complete an analysis in 3 years, due to the length of the proposal award. Due to the data volume, storage cached memory will not suffice and cloud storage will be needed. Since the researcher has 3 years to complete the analysis, "spot market" may be a way to reduce cost as the analysis will be queued and processed when the processing cost is lower (note: the table does not reflect spot market). **Note:** the AWS spot market gives users the opportunity to purchase unused EC2 instances at a rate less than the on-demand price.
2. **Operational User** (quick results): This user runs a forecasting model; latency needs to be short, for warnings and natural disaster mitigation to be provided in a timely manner. The fastest way to accomplish this is to run the

model in the same region the data are located, but that might not be possible if the necessary data products reside in multiple locations (NASA vs. NOAA vs. USGS). The user will use on-demand processing, as there is no wait to process the data, but it is more expensive.

3. **Synoptic-scale Researcher** (fluctuations): The synoptic researcher studies regional phenomenon from satellites and in situ data sources. The data volume fluctuates as data from in situ comes in batches. The user needs flexible storage and processing. A good choice for this user is "spot market", as mentioned above in the climate researcher example.
4. **Field Experimenter** (no local resources): When out conducting a field experiment, this user needs to know where the next location should be based on weather and/or ocean conditions. When out in the field, either on a research vessel or in a remote area, the only internet access is either from cell towers or communications. Downloading data is not an option due to the low bandwidth. The best option is to run scripts that can process and analyze satellite data in the cloud, and provide access to the resulting visualizations and data tables. The processing needs to be done on demand so results are timely, but there does not need to be a lot of storage; most of this can be kept in the cache.

Using the AWS pricing model (AWS published rates at a given time), the resources used and the cost to each user is summarized in Table 2.1

Table 2.1 User scenarios* and associated cost** in using EC2 Instances (e.g., for processing data), and S3 and local (instance-attached) Elastic Block Storage (EBS) storage. Note that AWS storage and storage types are further discussed in the following section.

User:	Researcher, climatology		Operational User		Synoptic-scale Researcher		Field Experiment		Notes
Compute									
Compute Power	2 vCPU		2 vCPU		1 vCPU		2 vCPU		AWS correlates higher vCPU with more memory
Memory	64 GiB		32 GiB		32 GiB		8 GiB		
Use frequency[1]	 Monthly spike traffic	25%	 Constant traffic	100%	 Weekly spike traffic	25%	 Daily spike traffic	25%	Workload | Duty cycle (%)
Project Length	3 years		On going		3 years		3 weeks		
AWS Pricing model, instance type[2]	On-demand , m5.4xlarge (0.77/hr)	$141	Std RI, m5.2xlarge, 1-yr, $0 up front	$180	On-demand, m5.2xlarge (0.39/hr)	$71	On-demand, m5.large (0.1/hr)	$185	Can be reduced if processing can fit within Lambda model
Local Storage[3] (EBS: general purpose SSD)	2 TB	**$205**	500 GB	**$50**	1 TB	**$102**	150 GB	**$15**	Can be reduced if contents moved to S3 during downtime
S3 Storage[4]	2 TB	**$47**	1 TB	**$24**	1 TB	**$24**	1 TB	**$24**	If S3 is only storage used, total cost/mo = S3 cost
Estimated Cost[5] per month (@2019-06-04)		**$412**		**$267**		**$207**		**$60**	=sum(local,S3,use freq)*1.05 (to build incidentals cushion)

NOTE: Table 2.1 should be used as a general guide only. Many factors, including workflow and user-specific needs as well as AWS costing, can influence and vary the ultimate cost to a user.

*Notes

1. Use Frequency (duty cycle) refers to processing time requirements (how often you use the instance in a given month) and is used to scale the pricing in the Local Storage (EBS) and S3 Storage rows.
2. Instance type: on-demand and reserved instance (RI) - if the processing is going to operate most of the day, every day, for a long number of months, then RI might be a more cost-efficient solution. If you have any slack in daily (or even weekly) processing, then on-demand may prove much more cost efficient if the user is willing to architect their application accordingly. See Section 3.2 for further discussion on EC2 instance types.
3. Local EBS storage is further discussed in the section to follow.
4. S3 storage is further discussed in the section to follow.
5. Estimated Cost per month in the table is listed based on AWS published rates for the date mentioned in the table (plus 5% cushion), and is likely to change over time, based on values set by the cloud platform provider.

The pricing model previously described does not consider:

- Enterprise-institutional agreements.
- Optimization (e.g., use of Dask).
- Specialized/esoteric resources, e.g., algorithm processing requiring GPU, specialized database architectures.

The user should be aware that their monthly AWS bill will likely have additional fees such as interacting with Storage Services, which is to be expected. These are further discussed in the following section.

We suggest the user first take their best guess at what their workflow may be in the cloud to (using perhaps a small test workflow/use case), and then use something similar to the AWS Cost Manager & AWS Cost Explorer tool (also briefly summarized in the Cost Resource section below) to assess their use and cost and adjust accordingly.

AWS storage

Many factors can impact how users leverage the Cloud, which will then impact costs. For users who associate with the Earth science community, the primary factors to consider are storage cost, compute power, and frequency of use. Below we summarize storage types in AWS.

1. Elastic Block Storage (EBS): Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 Instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use block storage. For pricing of EBS, see Amazon resources.

NOTE: This type of storage does not persist (data is deleted) once the instance is **terminated**. You can prevent this by changing the **DeleteOnTermination and DeletionOnTermination** settings. If the instance is stopped, and if it has an Amazon EBS volume as its root device, **the data persists and you will incur charges for its storage**.

- Suggested Use: for short-term storage needs, such as when running an analysis and need temporary storage for analysis output.

2. Simple Storage Service (S3): This is object storage designed to store and access any type of data over the Internet. S3 is used for backup and recovery; tiered archive; user-driven content; data lakes for Big Data analytics and data warehouse platforms; or as a foundation for serverless computing design.
3. S3 Glacier: extremely low cost and highly durable object storage service for long-term backup and archive of any type of data.

Based on current AWS published rates (June 2019), **1 TB** will roughly be **US$100/month in EBS, US$23/month in S3**, and **US$4/month in Glacier** storage. Let's break down the AWS bill a bit more (Figure 2.2).

Below is a sample of an AWS monthly bill:

Amazon Simple Storage Service USW2-Requests-Tier1 (4)		$27.15
$0.005 per 1,000 PUT, COPY, POST, or LIST requests	5,430,690 Requests	$27.15
Amazon Simple Storage Service USW2-Requests-Tier2 (4)		$0.93
$0.004 per 10,000 GET and all other requests	2,335,089 Requests	$0.93
Amazon Simple Storage Service USW2-TimedStorage-ByteHrs (1) (2)		$979.58
$0.022 per GB - next 450 TB / month of storage used	39,095.442 GB-Mo	$860.10
$0.023 per GB - first 50 TB / month of storage used	5,194.795 GB-Mo	$119.48
Amazon Simple Storage Service USW2-TimedStorage-GlacierByteHrs (3)		$0.01
$0.004 per GB / month of storage used - Amazon Glacier	2.027 GB-Mo	$0.01

Figure 2.2: Sample AWS monthly bill (as of June 2019)

If we look at the item with the largest cost (US$979.58), it is the storage in S3 (1) for roughly 40 TB of storage. (2) Below that is the $0.01 charge for storing 2 GB in Glacier. (3) The Glacier costs use similar Gigabyte-Month (GB-Mo) units, but the cost per unit is much lower (about 1/6 the cost of S3). On the bill, we can see some additional charges that are assessed based upon the usage: GET, PUT, COPY, POST, LIST, etc. (4) These are interaction fees — fees added for interacting with the storage service. Every time you add something, delete something, list something, change something, these costs get added to your bill with micro-penny charges.

In the case above, the usage fees are very small (less than 3% of the larger storage service bill). However, you can see that if your application were to churn the data in S3 at very high rates, these charges might become more appreciable. It's likely you would have to work very hard to make these charges exorbitant, but this points to the relationship between your process sophistication and the resulting charges.

Some additional salient points:

1. This billing reflects a lot of interaction with S3 (ingesting and archiving of data). That is demonstrated by having a relatively high number of "interaction requests" (4) something you should consider when architecting a processing pipeline. Users should not be too anxious about this unless they unleash a lot of automated processing without first auditing those interactions. Still, the charges are small, so a mistake should not break the bank. Again, an AWS tool such as Cost Explorer could be useful in monitoring and evaluating user data pipeline and workflow.
2. Glacier storage is considered "part of S3" and one would use S3 to transition to, and restore from, it. A user should not really expect to use "Glacier only" storage and do real work.
3. Once you transition something from S3 to Glacier, you can no longer interact with that "object" directly. You can see if it exists and gather some metadata about it, but you cannot interact with it until you restore the object back to S3. The restoration can take anywhere from a few moments to a few hours, depending on how much you're willing to pay. There are three restoration options that will restore data at faster rates with a higher price.
4. When you restore an object from Glacier, you will then have two copies: the original Glacier copy and a new S3 copy that you can interact with, but it will be short-lived. At restoration time, the user can configure a length of time after which the restored S3 version will be automatically deleted (you can copy it somewhere else to permanently keep it before the restored copy is automatically deleted).

> **NOTE**: During the time you have the two copies you are charged for both Glacier and S3 rates.

Even with the additional storage interaction charges, S3 cost could still be significantly less than EBS storage (associated with EC2 instance) as shown in Figure 2.3:

Amazon Elastic Compute Cloud running Linux/UNIX		$1,956.61
$0.0116 per On Demand Linux t2.micro Instance Hour	2,264.668 Hrs	$26.27
$0.0928 per On Demand Linux t2.large Instance Hour (6)	744 Hrs (6)	$69.04
$0.1856 per On Demand Linux t2.xlarge Instance Hour	(7) 6,639.795 Hrs	$1,232.35
$0.3328 per On Demand Linux t3.2xlarge Instance Hour	201.156 Hrs	$66.94
$0.3712 per On Demand Linux t2.2xlarge Instance Hour	34.222 Hrs	$12.70
$0.768 per On Demand Linux c5d.4xlarge Instance Hour	715.240 Hrs	$549.30
EBS		$49.74
$0.05 per GB-month of Magnetic provisioned storage - US West (Oregon)	(5) 256.000 GB-Mo	$12.80
$0.10 per GB-month of General Purpose SSD (gp2) provisioned storage - US West	369.379 GB-Mo	$36.94

Figure 2.3: Example cost table (as of June 2019).

Charges in Figure 2.3 reflect the "per instance type" costs (as opposed to "per instance") for the month. They also include the EBS (local direct storage) that the instances consume. EBS charges can be substantial,(5) and one should recall they are attached to one instance, whereas S3-stored data can be accessed from a multitude of instances.

What happens when you terminate an instance that has EBS storage attached to it? You can have the data be deleted and removed from EBS, or you can keep the data on EBS so you can attach that storage to a new instance later. However, you will pay EBS storage rates for all of that time, whether or not you are using it with an operating instance.

Some interesting notes on the EBS storage bill:

1. The month we're looking at in this particular case is May, which has 31 days — or, more appropriately here, 744 hours. From this you can tell that we had the equivalent of exactly one t2.large instance running for the entire month.(6) Also, almost 9 full-time t2.xlarge instances were used,(7) but this may be due to a lot of processing activity. The thing to note is all the various operational times are summed up and you don't see a cost associated with any particular instance. With on-demand, instances come and go and are not special.

2. You'll see that EBS charges are 2-4 times higher than those of S3. So, if a user can architect the processing to utilize S3 directly (which is not always easy to do), the savings can be substantial.

3. The type of EBS selected (magnetic spinning disks vs. Solid-state Drives (SSD)) makes quite a bit of difference in cost. If the user processing requires local EBS storage, but can operate adequately with the performance of magnetic storage, the cost will be half that of SSDs.

4. Duty cycles for processing come into play here as well since, unlike reserved instances, you can stop/start processing and only pay for the hours you actually use.

Table 2.2 below, available on the AWS website, presents other considerations for storage type, such as the way you would use storage - what are *your* needs?

If You Need:	Consider Using:
Persistent local storage for Amazon EC2, for relational and NoSQL databases, data warehousing, enterprise applications, Big Data processing, or backup and recovery	Amazon Elastic Block Store (Amazon EBS)
A simple, scalable, elastic file system for Linux-based workloads for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, so your applications have the storage they need – when they need it.	Amazon Elastic File System (Amazon EFS)

If You Need:	Consider Using:
A fully managed file system that is optimized for compute-intensive workloads, such as high-performance computing, machine learning, and media data processing workflows, and is seamlessly integrated with Amazon S3	Amazon FSx for Lustre
A fully managed native Microsoft Windows file system built on Windows Server so you can easily move your Windows-based applications that require file storage to AWS, including full support for the SMB protocol and Windows NTFS, Active Directory (AD) integration, and Distributed File System (DFS).	Amazon FSx for Windows File Server
A scalable, durable platform to make data accessible from any Internet location, for user-generated content, active archive, serverless computing, Big Data storage or backup and recovery	Amazon Simple Storage Service (Amazon S3)
Highly affordable long-term storage that can replace tape for archive and regulatory compliance	Amazon Glacier
A hybrid storage cloud augmenting your on-premises environment with Amazon cloud storage, for bursting, tiering or migration	AWS Storage Gateway
A portfolio of services to help simplify and accelerate moving data of all types and sizes into and out of the AWS cloud	Cloud Data Migration Services
A fully managed backup service that makes it easy to centralize and automate the backup of data across AWS services in the cloud as well as on premises using the AWS Storage Gateway.	AWS Backup

Elastic Compute Cloud (EC2) Instance Considerations

Aside from considering storage use, needs, and cost, AWS Pricing Model for Instance type should also be considered. Below are some thoughts on this topic.

AWS Reserved Instance (RI) vs On-demand usage and pricing:

1. AWS-advertised savings for using RI may not be readily realized: https://www.cloudhealthtech.com/blog/aws-reserved-instances-vs-on-demand.
2. The average achievable discount of RI over on-demand is actually 40%; optimistically 50%.
3. Some institutions are reluctant to agree to RI contracts, so that option might be unavailable to a particular user.
4. If your processing can live with a 50% duty cycle, you can readily use on-demand and turn off processing when not in use.
5. Any RIs that are sitting idle, are wasting money. Though this is also true for on-demand, you can turn these off and stop paying (recall any storage (EBS) associated with the on-demand instances will be lost once the instance is stopped unless it is a root volume).
6. During the terms of the contract, RIs don't typically get a price reduction when AWS lowers the pricing on the on-demand instances. You are stuck for the duration of the contract.
7. Architecting your processing so that it can be easily turned on and off to better use on-demand instances, thereby paying only for what you use, is a win. If you do this, then you are now one step closer to being able to effectively use "spot instance pricing". This is where real cost savings can come into play.

8. Unless your processing is going to keep an instance busy for > 70% of the time, consider specifying an on-demand instance.
9. Even if your processing operates at a higher duty cycle, thereby allowing you to effectively use RI, you may be better off with 1-year contracts. You can then take advantage of new instance types and pricing as they become available.

Resources

AWS offer tools to help the user understand, estimate, and track their usage and cost:

Amazon Web Services (AWS) Pricing Calculator

It is important to keep in mind these prices can change with time depending on the cloud provider, so it might be best to revisit the tool and create an estimate as close as possible to when you'd be needing the estimate. Pricing calculator options, or considerations for the user to take into account that might impact cost in using the cloud include (but are not limited to):

- Number of instances **[see definition in Cloud Primer (CP) glossary]**
- Operating system (e.g., Windows, Linux)
- Compute power (vCPUs)
- Memory (GiB, RAM)
- Storage amount
- Frequency of use
- Region **[see more on AWS regions in note below]**
- Pricing model
- Reservation term
- Payment options

Enterprise-level cloud computing (agreements for an institution) could also be considered, where appropriate, which could impact (reduce) the cost of working in the Cloud.

AWS Cost Manager

The AWS Cost Manager offers several tools to understand and control your AWS cost:

1. **AWS Cost Explorer** - analyze your AWS cost and usage
   - It provides a set of default reports that can help the user get started on looking at their usage and cost. Filters and groupings then allow the user to look deeper into their cost, by project, AWS region, AWS instance type, etc., as well as forecast future usage and cost. See AWS Cost Explorers for more details.
   - This could be a useful tool that provides insight into the cloud usage and cost to help the user manage their projects, adjust their usage to optimize work (research, applications, data analysis, storage) and cost, and plan for future usage. Insight from this tool could be used in the Advanced option of the AWS Pricing Calculator.
2. **AWS Budgets** - set custom cost and usage budgets
3. **AWS Cost and Usage Report** - access comprehensive cost and usage information
4. **Reserved Instance Reporting** - dive deeper into your reserved instances (RIs)
5. **Alarms** - **see *Create Billing Alarm to Avoid Unexpected AWS Charges*** section.

AWS Free Tier and Training Programs

If you are interested in an introduction to AWS, which could also give you an idea if cloud computing is the appropriate solution for your scientific or applications endeavors, and to better understand the AWS cost model through hands-on experimenting, there are a few options:

1. **Free tier**: The AWS Free Tier allows a user to gain free, hands-on experience on the AWS cloud platform, and includes 750 hours of Linux and Windows t2.micro instances each month for one year. To stay within the Free Tier, use only EC2 Micro instances.

   - The AWS Free Tier includes 30GB of Storage, 2 million I/O operations, and 1GB of snapshot storage with Amazon Elastic Block Store (EBS).

2. **AWS Programs for Research and Education:** This program provides short-term access to the AWS platform, allowing researchers, application communities, or educators the opportunity to move their workflow or system to the cloud to test, usually with free credits up to 12 months.

A note on AWS regions

Amazon cloud computing resources are hosted in multiple locations world-wide. These locations are composed of **AWS regions**, Availability Zones, and Local Zones. Each AWS region is a separate geographic area, and is completely independent of another. Any Amazon activity the user initiates runs only in the user's current default AWS region (which can be changed in the AWS console). When viewing resources, a user will see only the resources that are tied to the AWS region that is specified. This is because AWS regions are isolated from each other, and resources are not automatically replicated across AWS regions (AWS reference).

NASA EOSDIS data will be archived in AWS region US-West-2 (Oregon). Typically, when data is downloaded from its "home" region, there is an egress cost associated with that data pull. However, a user can search and download NASA EOSDIS data regardless of which region they may be working in, at no cost to the user. EOSDIS will pay for any egress of data outside the archiving region, but note that means it is also subject to throttling. In other words, the download of data from outside its home storage region may be slower (compared to accessing data from the same region). In-region users can achieve a much higher degree of bandwidth and parallelism than users going through https. Users out of region (or on prem – those working off the cloud, or HPC users for example) will access data through the HTTPS/ Cloudfront URL, just like most files they would access today. Also recall all NASA data will continue to be free to access and download, regardless of whether the user is working in the cloud or not.

Create a Billing Alarm to Avoid Unexpected AWS Charges

A Billing Alarm will notify you by email when your AWS account charges for the month have exceeded the monetary amount you have set. This is an extremely useful tool to stay on-budget. It can also be a way to let you know if you have forgotten to stop or terminate an AWS service – such as an EC2 instance – and help avoid unexpected charges.

Enable Billing Alerts for your account

Billing Alerts must be enabled on your account before you can create a Billing Alarm.

1. SIGN IN to your AWS account.

2. Click on your **account name** in the upper right corner to expose the dropdown menu.
3. Select My Billing Dashboard to open the *Billing & Cost Management Dashboard* (Figure 2.1).

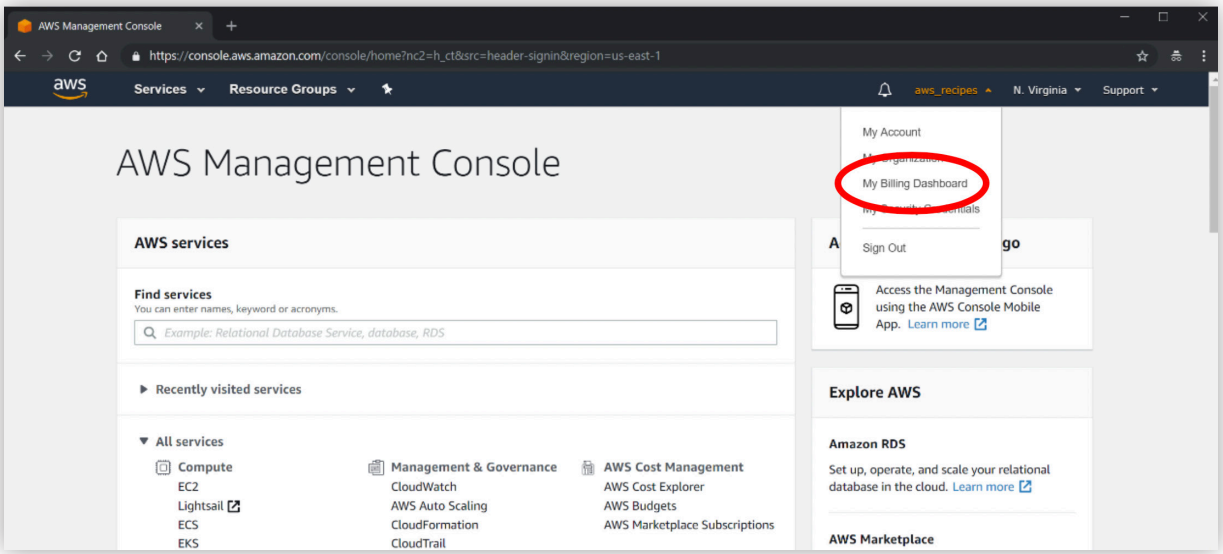


Figure 2.1

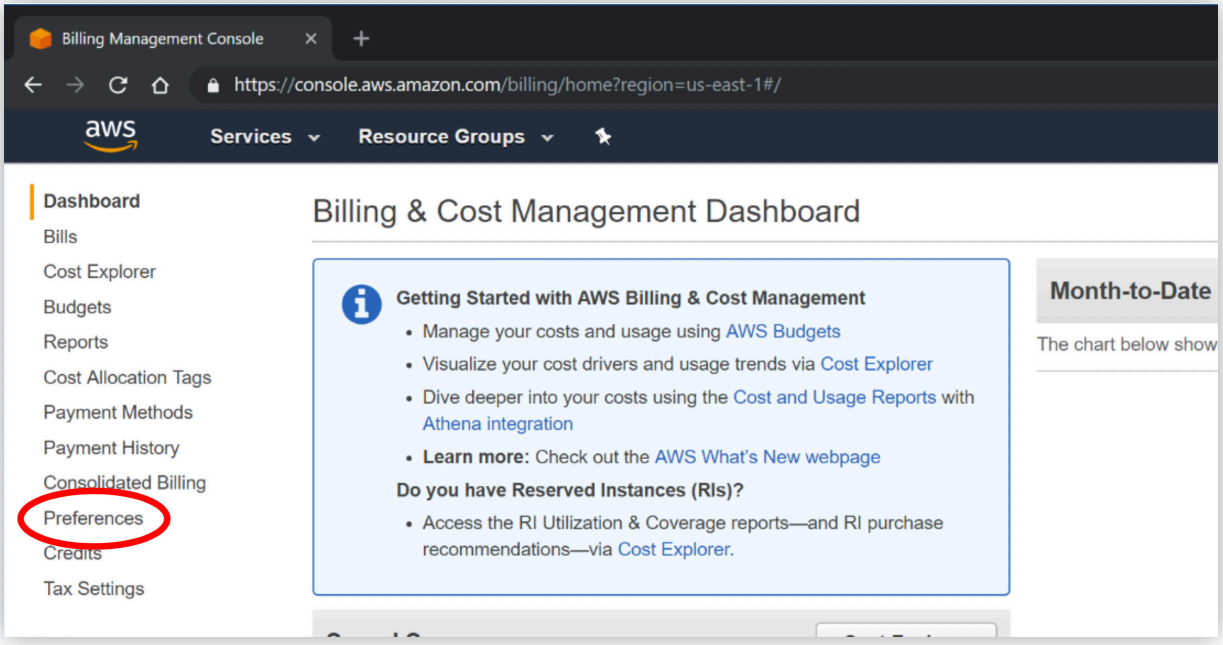


Figure 2.2

4. Click on **Preferences** in the left Dashboard menu (Figure 2.2)

5. Check the box in front of **Receive Billing Alerts** and then click **Save Preferences** (Figure 2.3)

- If you are eligible, make sure the **Receive Free Tier Usage Alerts** box is checked as well.

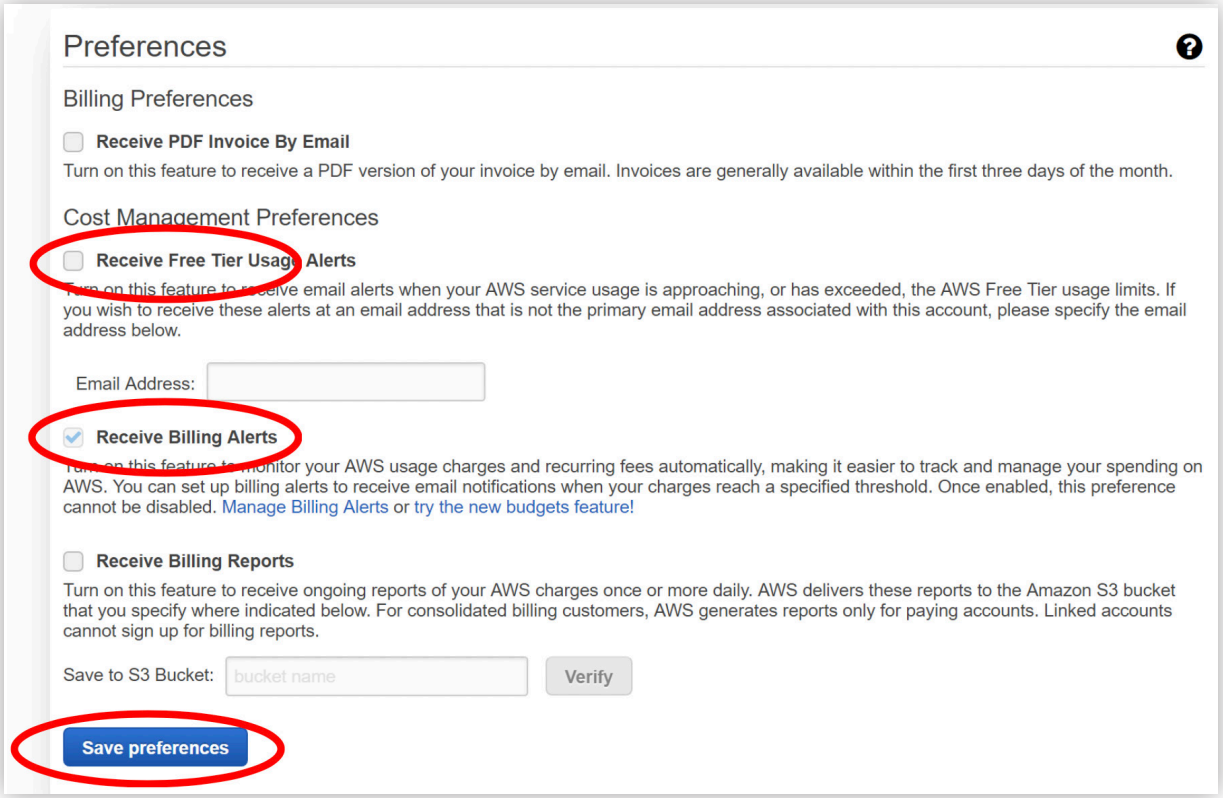


Figure 2.3

Create a Billing Alarm using the CloudWatch console

1. Click on aws in the upper left of the *Billing & Cost Management Dashboard* screen to open the *AWS Management Console*.
2. Type cloudwatch in the *AWS Management Console* search box under AWS Services and then click on CloudWatch (Figure 2.4).

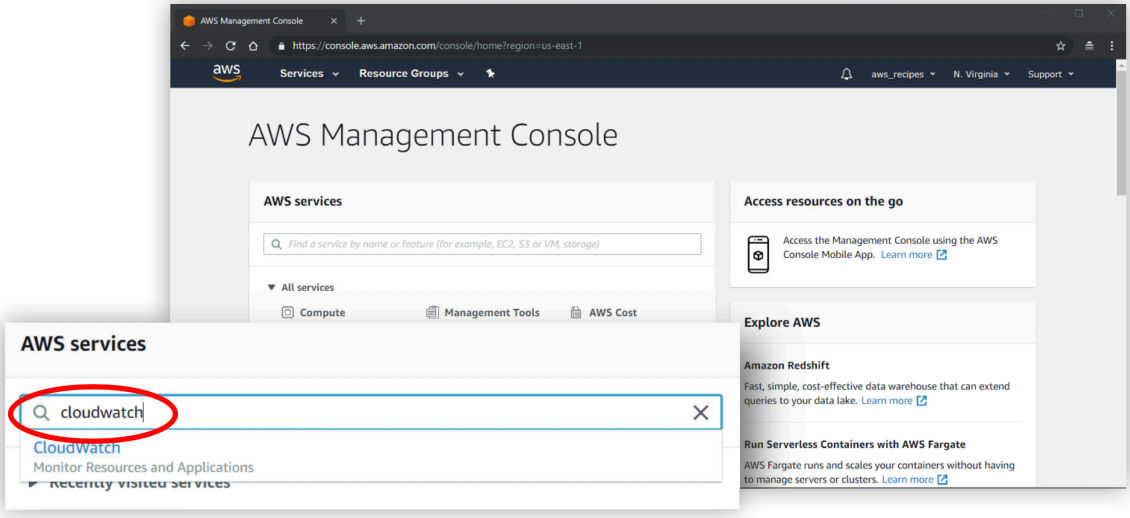


Figure 2.4

3. Click on **Billing** in the menu on the left under **Alarms** in the *CloudWatch* console, then click on **Create Alarm** (Figure 2.5).

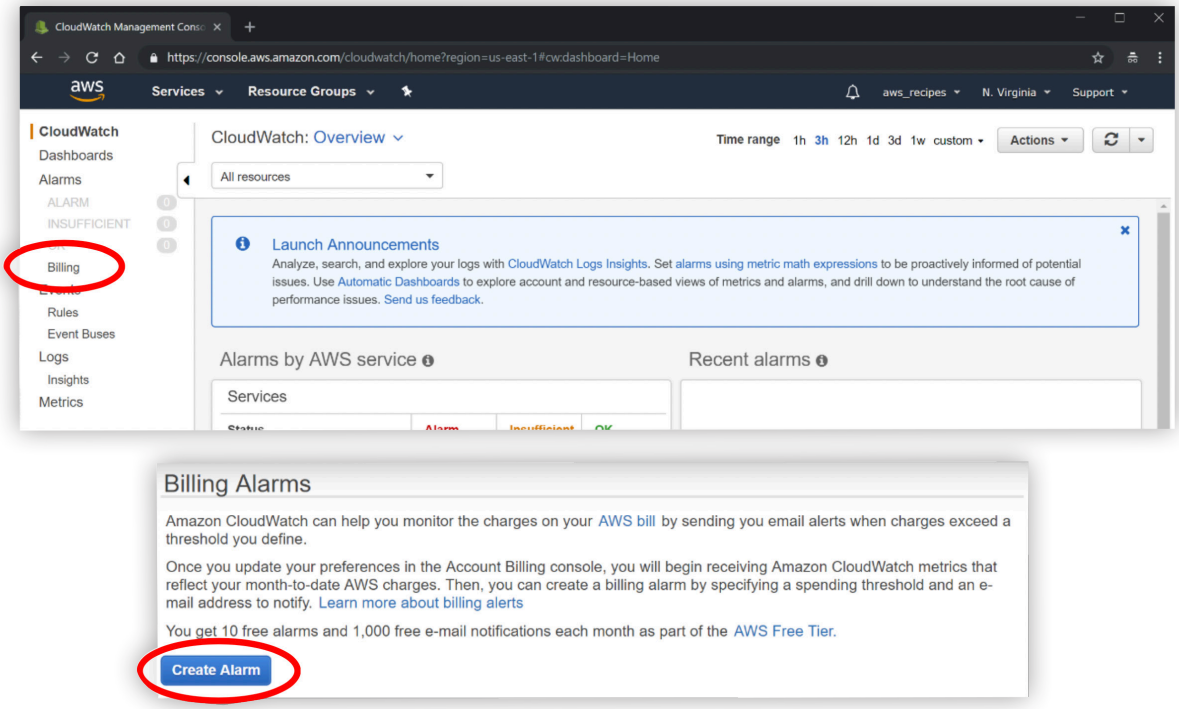


Figure 2.5

4. Scroll to the bottom of the Create *new alarm* page and click on **show advanced** (Fig. 2.6).

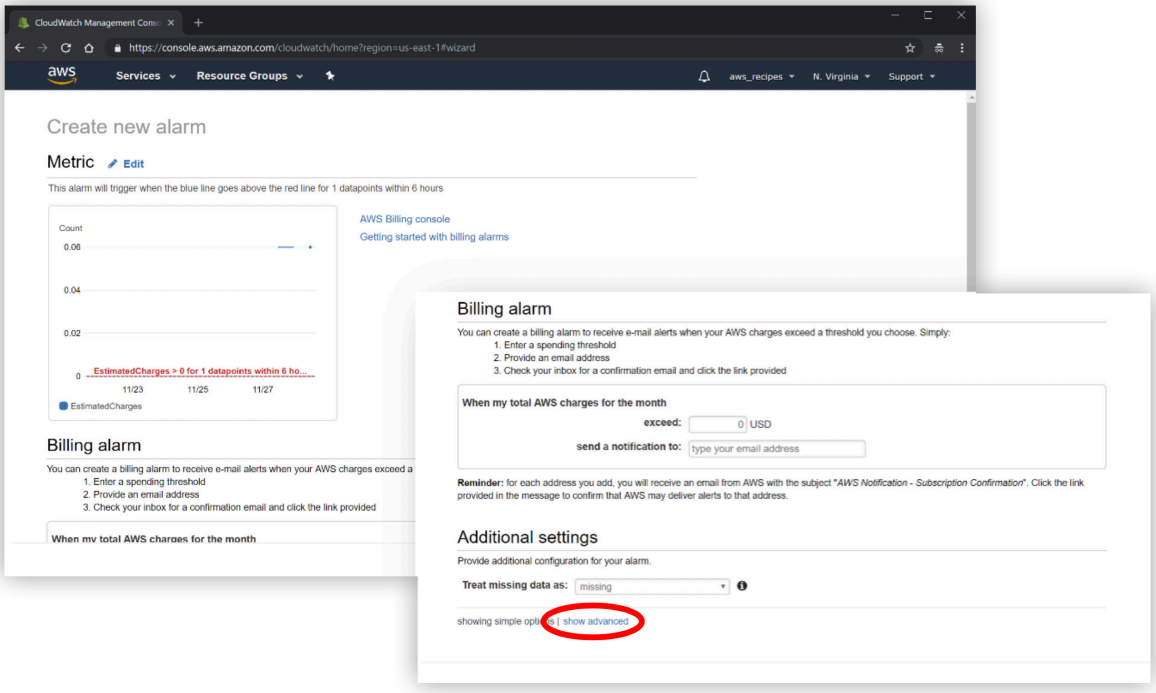


Figure 2.6

You will then see the Alarm Details screen (Figure 2.7).

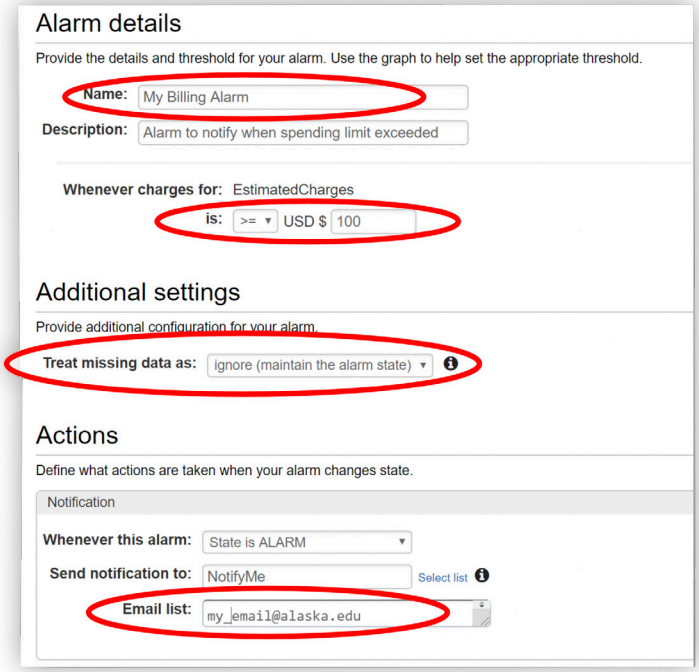


Figure 2.7

5. Add a **Name** for your new alarm and a **Description** (optional).
6. Select a condition and monetary amount under **Whenever charges for**: Set **is**: to >= and **USD $** to the amount in dollars you want to set as a limit.
   - You will receive email notification when your account charges have reached or exceeded this amount.
7. Choose **ignore (*maintain the alarm state*)** under **Treat missing data as:**
8. Enter your **email address** under **Email list**:.
9. Click on the **Create Alarm** button at the bottom of the Create *new alarm* page (Figure 2.8).

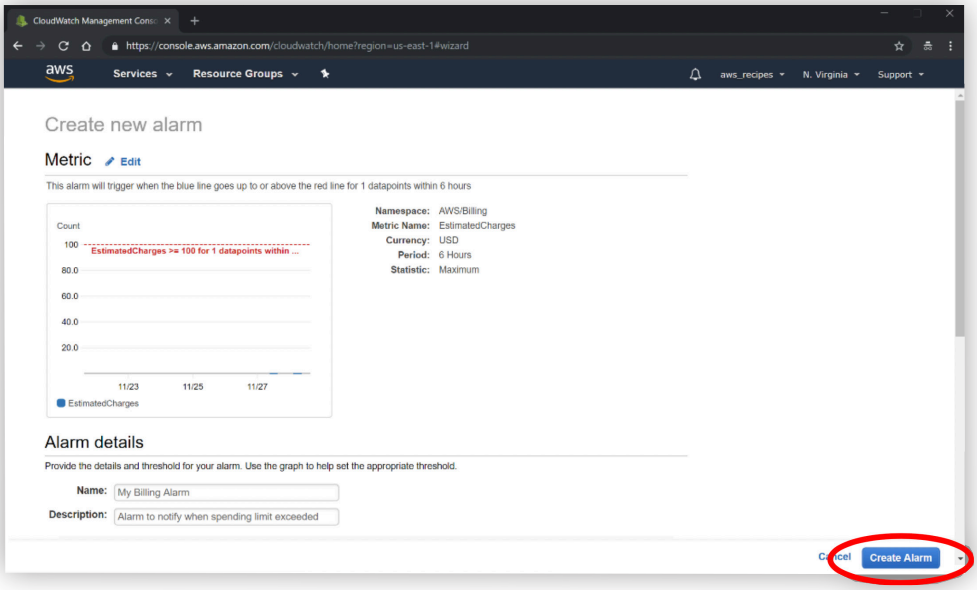


Figure 2.8

Important: You will be sent an email to confirm the email address set in your alarm. Open the email and click on the **"Confirm subscription"** link to enable notifications.

When you create a new alarm, an alarm status window will open. The State will initially show INSUFFICIENT_DATA but will update to OK in a minute or so (Figure 2.9). If the State shows PENDING_CONFIRMATION, open the confirmation email that was mailed to you and click on **Confirm subscription**.

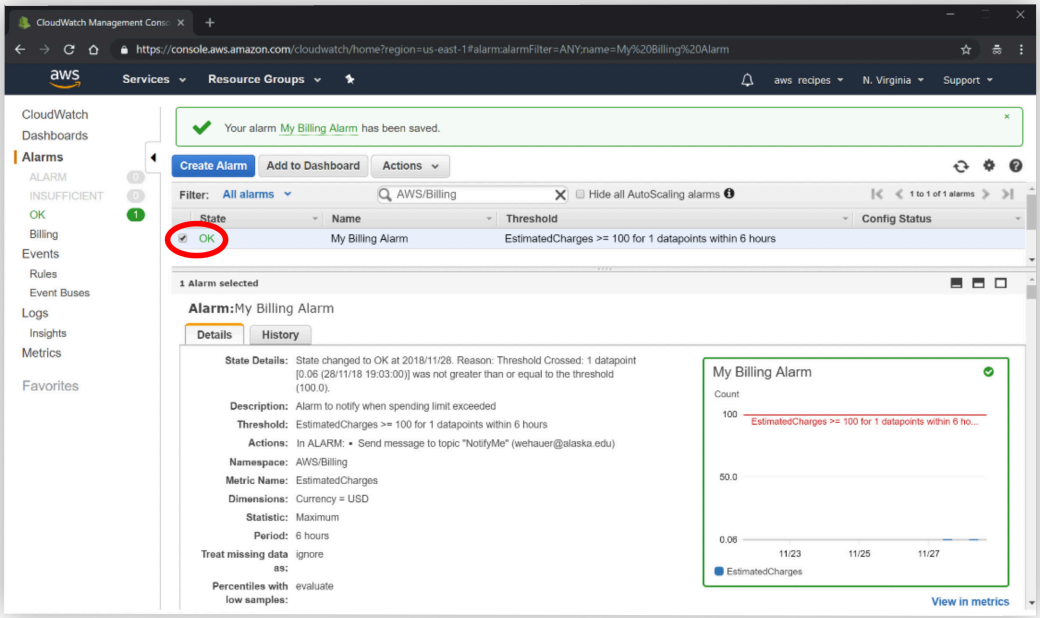


Figure 2.9

Monitor a Billing Alarm status

1. To see details about your alarm, check the box in front of the alarm description (Figure 2.9).
2. When the monetary limit set for your alarm has been exceeded, the State will change to **ALARM**, the graph border will be red, and an email will be sent to you notifying you of this (Figure 2.10).

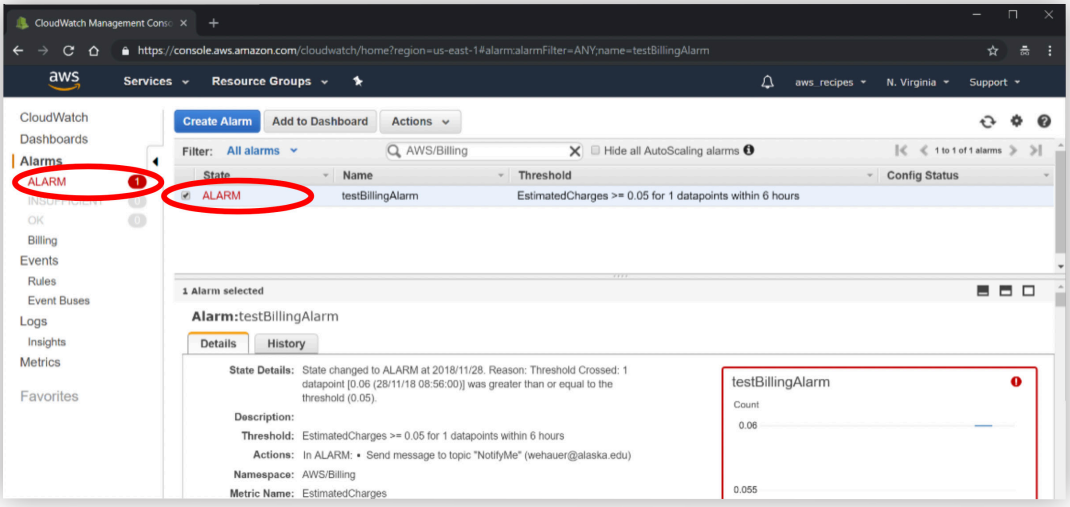


Figure 2.10

You can check the status of an alarm at any time by opening the *CloudWatch* console.

3. Click **Billing** In the left menu, under **Alarms**,
4. Check the box in front of an alarm in the list to see its details.

Delete a Billing Alarm

1. Open the *CloudWatch* console
2. Click **Billing** in the left menu, under **Alarms**.
3. Check the box in front of an alarm in the list to see its details (Figure 2.11).

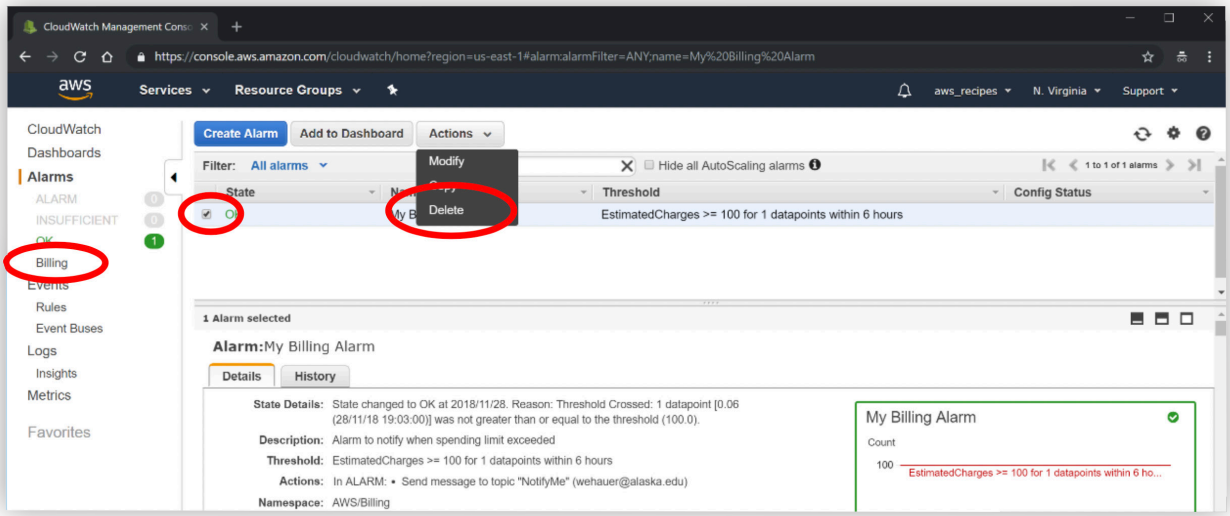


Figure 2.11

4. Click on the **Action** button to expose the dropdown menu.
5. Select **Delete**.
6. Confirm Yes, Delete when prompted