A Very High-Level Overview of Cloud Architecture - Deploying A Three Tier Application on the AWS Console

AWS is useful if you have an application in which you have the features in hand and you would like to deploy it on AWS console management.

For instance, if you click the Netflix.Com URL, you are taken straight to the website.

You may not know where the servers are located. (Can be in different regions and availability zones)

A three-tier application is a very high-level overview of the cloud.

This is because part of the network is accessible to the public but the servers which are the most important components of the architecture are not accessible through the internet.

- Part of network-accessible through internet is – Public Subnet
- Part of the network not accessible to the internet - Private Subnet

As a cloud architecture, your biggest role is to ensure the services deployed in the network environment (VPC) are secure from malicious activities.

You can increase the security of the network by not giving access to the server through the internet.

All servers are going to be in the private subnets. (Web Server, App Server, and Data-Tier)

Therefore, AWS is very secure because we are not going to give access to the Web, Application, and Data Tiers servers through the internet.

This is very similar to not giving access to your computer/laptop to anyone which eliminates any chances of being hacked.

The private subnet is going to access the internet through the NAT (Network Access Translation), which will convert the private IP Addresses to Public

Addresses. This is an increased security measure to the network environment.

It is important to know that the entire Netflix infrastructure is built using AWS services.

Even though the Netflix servers are in the private subnets,

- Webserver Tier
- Application Tier
- Data Tier

the content must be delivered to the public.

This is done through **Amazon ELB** and **CDN**.

CDN stands for Content Delivery Network, and the most famous service on AWS is the cloud front, which is in the CDN Regions.

Amazon ELB – It is a load balancer that is in a network that can be accessed through the internet. Remember there are many different ELB

- Application Load Balancer
- Gateway Load Balancer
- Network Load balancer
- Classic Load Balancer

Netflix has 1000 Kinesis.

Kinesis is a streaming data analysis tool that shows how users are streaming and interacting with Netflix three tiers application. When you go to the Netflix website and select a movie, they can follow everything you are doing and analyze all your activities.

Netflix has **100,000 EC2** servers plus more **T2 unlimited Servers** for bursting purposes especially when a new movie is released.

Netflix utilizes edge locations to stream famous movies at a faster rate all over the world.

Netflix does utilize CDN regions all over the world, especially, the **CloudFront services**.

Netflix utilizes both RDS and DynamoDB data engines and both can scale as needed.

ElastiCache services are highly utilized by Netflix to reduce latency and increase throughput traffic. This means the data are stored temporarily in a high-speed data layer where they can be processed quickly without accessing the primary data engines in the data tier.

Amazon Rekognition – Utilized by Netflix, especially when delivering children-oriented content to filter naked and nudity images.

S3 buckets – Have unlimited space and are located at the regional levels. Netflix uses S3 Buckets to store static content to decrease latency and increase throughput traffic.

S3 buckets are also used to store streaming data originating from Kinesis before they are processed by EMR and transferred to Redshift where they are analyzed through Athena and Quicksight services for more insight.

Glue – used to integrate the data in the format that can be processed by EMR and then stored in the Redshift for future analysis with Athena or Quicksight.

For every user, approximately 20 MB of RAM is required (Random Access Memory).

Scaling Metrics on AWS are usually CPU-oriented.

CPU - Central Processing Unit is the brain of the server.

Netflix paid Amazon **$ 84 Million per month, approximately $ 1 Billion a year to use the AWS services.**

AWS Free tier account allows new users free usage of up to 1 GB RAM and 1 vCPU or 1 core CPU for every server launched.

As a company, the recommendation is that each instance of **EC2** instance should have **8 vCPU cores** and **64GB RAM** (This is attached to the EBS – Elastic Block Storage).

What To Look for With EC2 Monitoring

You are always going to configure your instance in accordance with your requirements.

Regardless of the configurations, it is good to monitor basic system-level metrics to keep an eye on the health of your core infrastructure.

The ability to monitor will help you to understand whether the EC2 resource capacity is able to match the demand of the users.

Available EC2 metrics generally fall into three types:

- Disk I/O - Input / Output
- Network
- CPU

- In addition to these resource metrics from EC2, you also have access to binary status checks, which report the health of your instances and the AWS systems they are hosted on.

- You can track scheduled events that might affect your instances' status or availability.

RAM (Random Access Memory) and Hard Disk Drive (HDD) are both types of computer memory.

RAM is used to store computer programs and data that the CPU needs **in real-time**.

RAM data is volatile and is erased once the computer is switched off.

HDD, the hard disk drive has permanent storage, and it is used to store user-specific **data and operating system files**.

Amazon's CloudWatch monitoring system is the easiest way to see most resource metrics for your EC2 instances and other AWS services, with a few things to keep in mind.

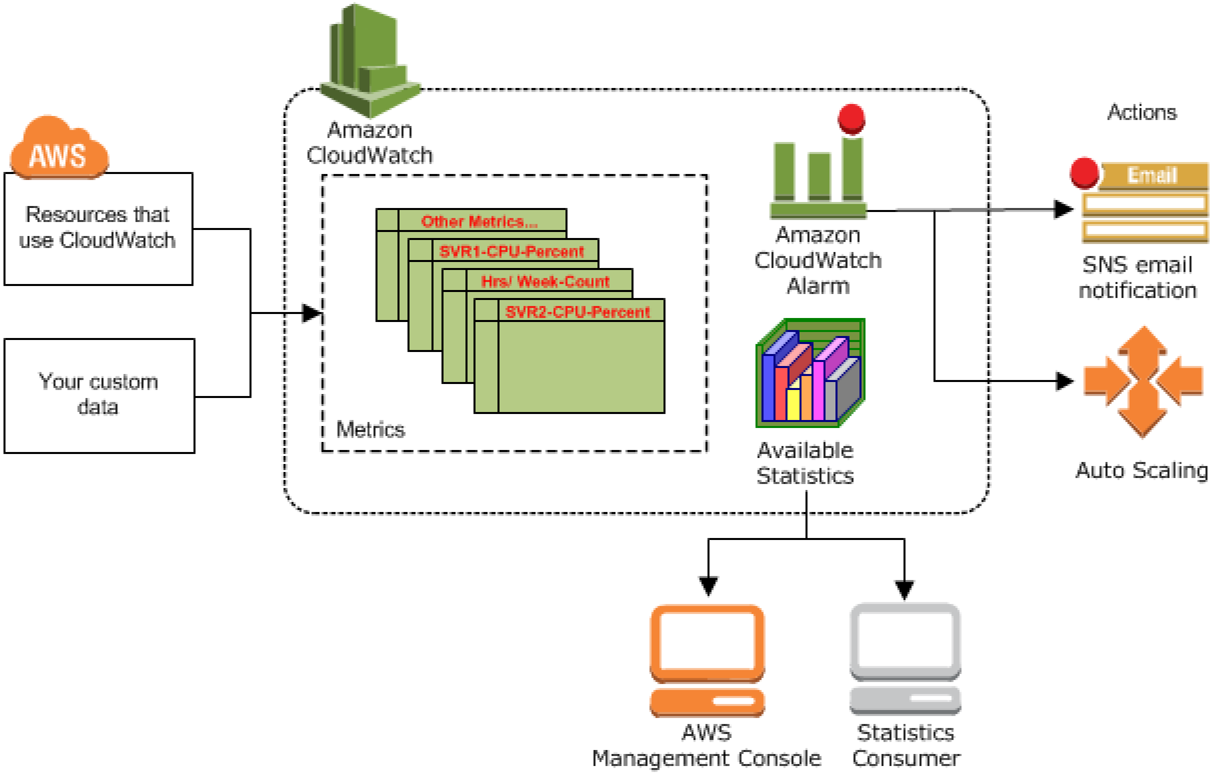
First, by default CloudWatch uses basic monitoring, which only publishes metrics at five-minute intervals.

You can enable detailed monitoring when available to increase that resolution to one minute, at additional cost.

Third, AWS separates most resources by region, so you can generally only view CloudWatch metrics for instances within a single region at a time.

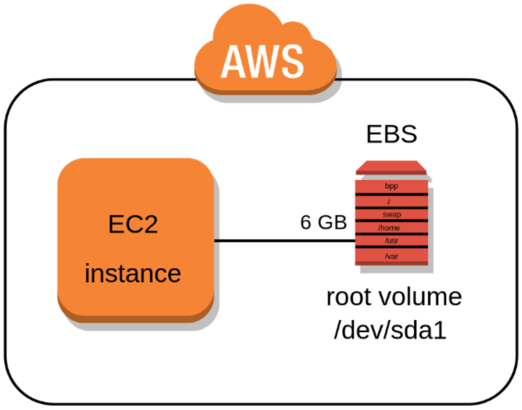
Very Important:

CloudWatch does not expose metrics related to instance memory. (RAM and HDD of your EC2) are not monitored by cloud watch.

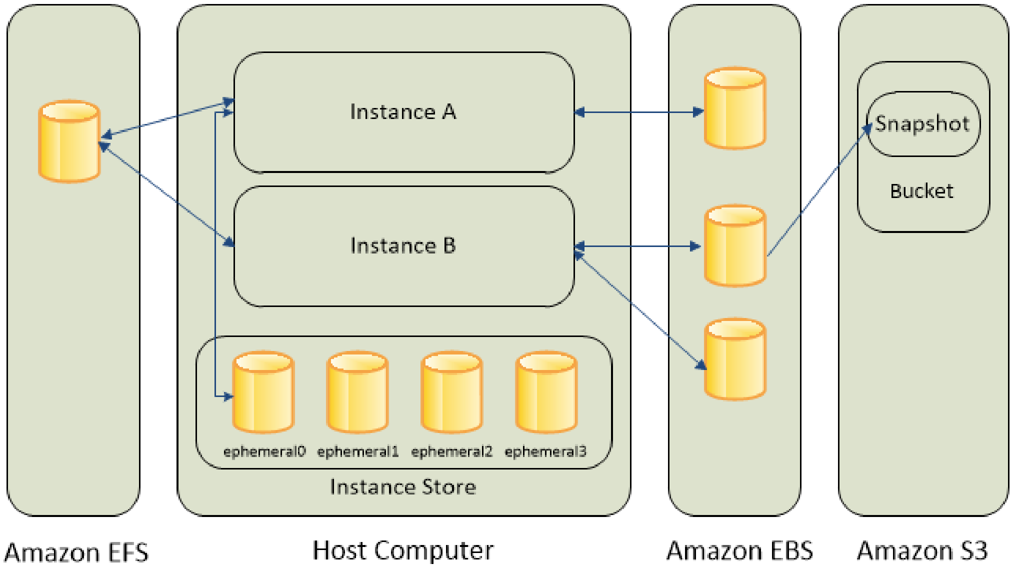


Disk I/O metrics – Disk Input and Output operations

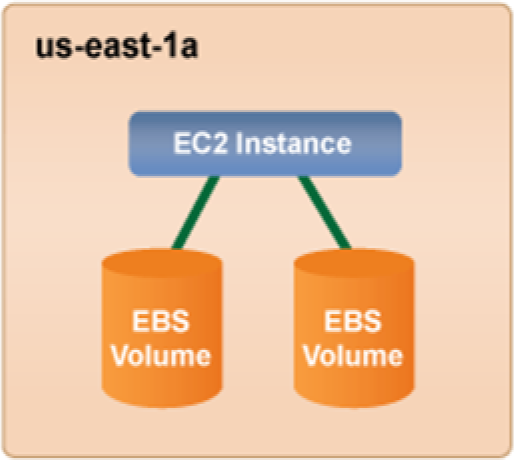
- Disk I/O includes read or write or input/output operations (defined in KB/s) involving a physical disk (EBS).
- In simple words, it is the speed with which the data transfer takes place between the hard disk drive and RAM, or basically it measures active disk I/O time.
- There are two primary kinds of block-level storage volumes attached to EC2 instances:
- EBS volumes and instance store (ephemeral) volumes.



Instance store volumes are physically attached to the host computer the instance runs on.



- This means that their performance levels are more predictable than EBS volumes, which might be splitting hardware resources among multiple tenants.
- All data on instance store volumes is lost when the instance is stopped or if the disk fails (hence “ephemeral”). Ephemeral means – lasting for a short period
- EBS volumes, on the other hand, provide persistent storage. Note that many EC2 instance types do not support instance store volumes. They support Elastic Block Storage.



Attach Multiple Volumes to the same EC2 Instance.

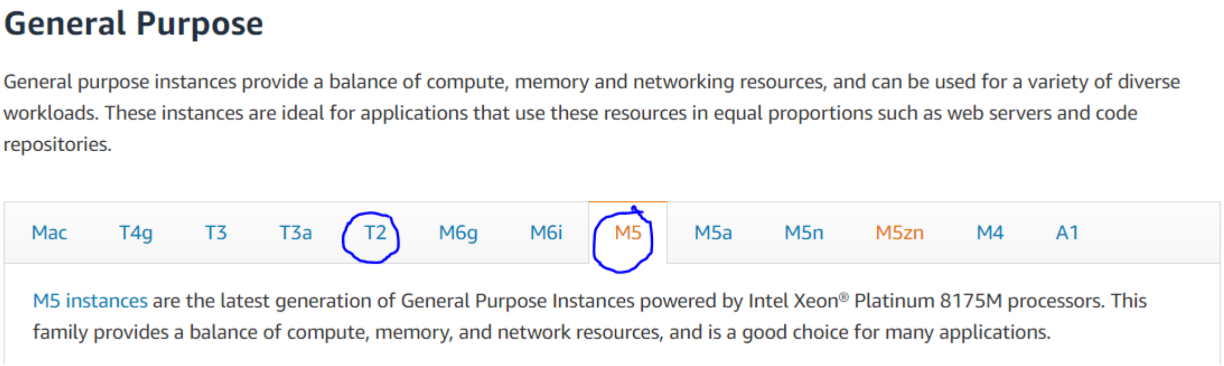
EBS Volumes

- 1TB -16TB
- $0.10/GB per month
- Attach an EBS Volume(s) to any EC2 instance in the same Availability Zone
- Create an EBS Snapshot of an EBS Volume at any point in time
- Create an EBS Volume(s) from any EBS Snapshot

- Both EBS and instance store volumes can be in solid-state drive (SSD) or hard disk drive (HDD) format.
- A solid State Drive is much faster than a hard drive disk because SSD does not have moving parts.
- The number, capacity, and performance of these disks differ based on the instance type and volume configuration.
- Monitoring EC2 disk I/O can help you ensure that your chosen instance type's disk IOPS and throughput match your application's needs.
- CloudWatch's main EC2 disk I/O metrics only collect data from instance store volumes.
- For all other instance types, disk I/O for EBS volumes must be monitored via CloudWatch's EBS metrics, but these are only available for C5 and M5 instance types.

- Instances for general purpose. The most famous one is T2 and M5.
- T2 is very appropriate for bursting situations.
- They are all used in an application that has a balance in computing, memory, and networking. The application using these instances

Disk read/write operations



General Purpose

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

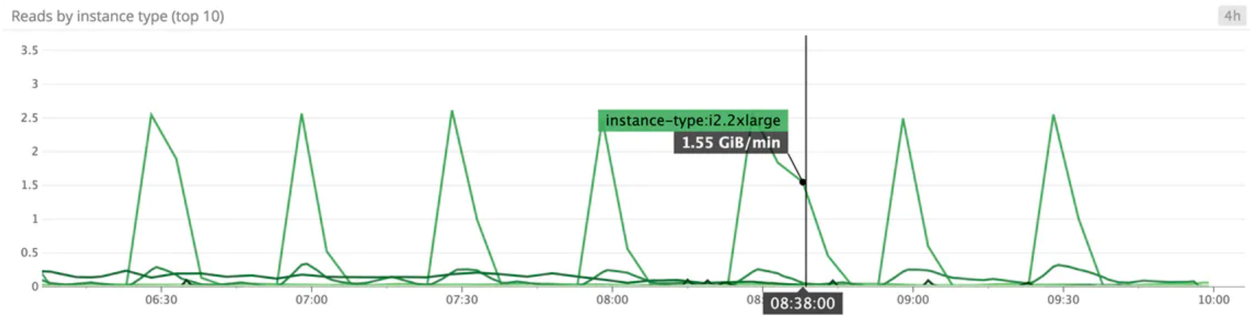
M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175M processors. This family provides a balance of compute, memory, and network resources, and is a good choice for many applications.

- Because any data stored on instance store volumes is lost if the instance stops or fails, these types of volumes are best suited for I/O-intensive uses such as buffers, caches, and other cases where data is stored temporarily and changes frequently.

- This metric pair can help determine if degraded performance is the result of consistently high IOPS (input/output operations per second), causing bottlenecks as disk requests become queued/lined up.
- If your instance volumes are HDD, you can consider a move to faster SSDs.
- Or you can upgrade the VM to an instance with more volumes attached to it.

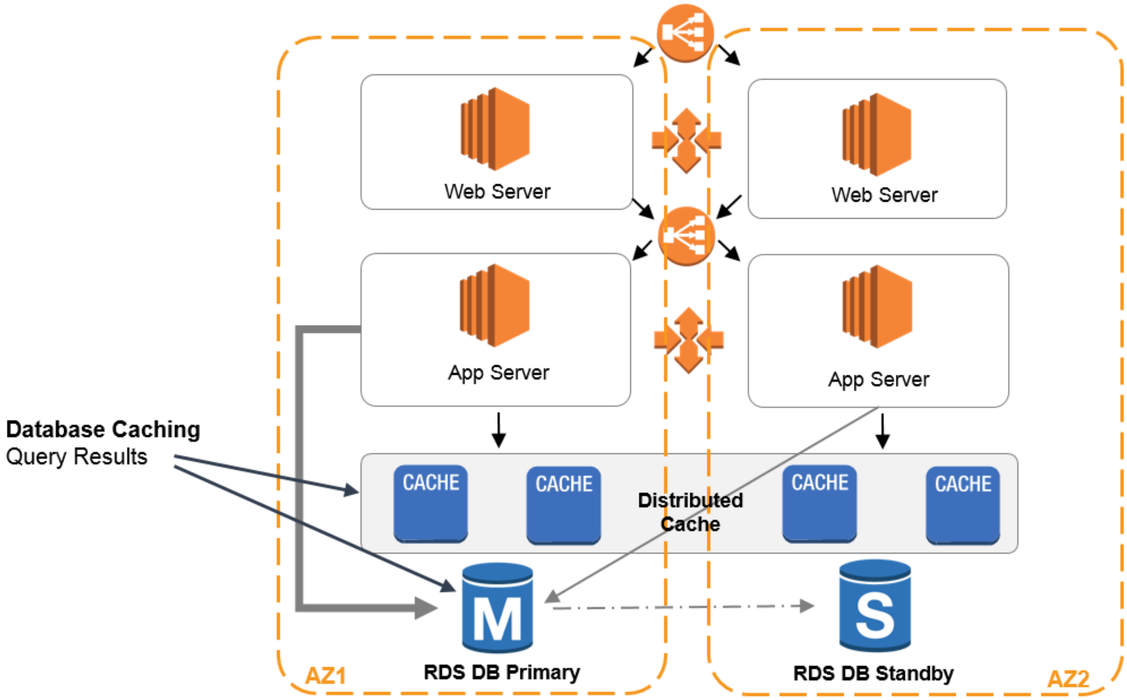
Disk read/write bytes

- Monitoring the amount of data being written to and read from disk can help reveal application-level problems.
- Too much data being read from a disk might indicate that your application would benefit from adding a caching layer.
- Prolonged higher-than-anticipated disk read or write levels could also mean request queuing and slowdowns if your disk speed is not fast enough to match your use case.



Caching layer

In computing, a cache is **a high-speed data storage layer that stores a subset of data, typically temporary in nature**, so that future requests for that data are served up faster than is possible by accessing the data's primary storage location.



Network metrics

Network metrics are particularly important for cloud-based services like EC2 that rely on consistently strong network connections, and that might be dispersed across various availability zones.

This is especially true if you have attached **EBS volumes to your instances**, as **they are networked drives**.

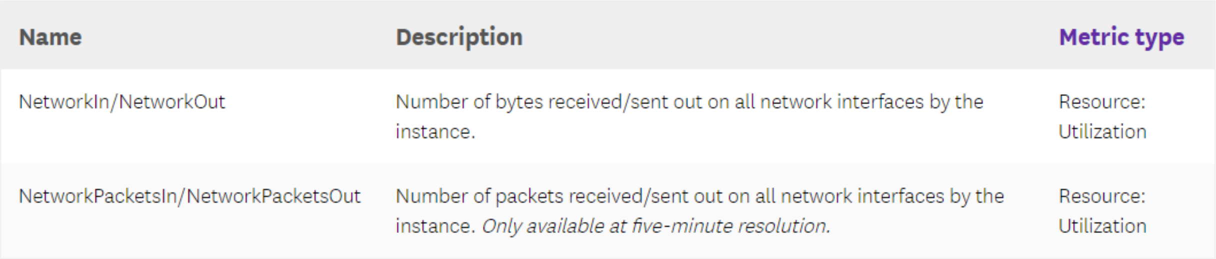
Network MTU is a standard 1,500 bytes for most instance types, but many allow jumbo frames of as much as 9,001 bytes, increasing efficiency and reducing overhead for applications that transmit large amounts of data.

Selecting the right type and availability zone for your instances can improve network performance, as can configuration options such as placement groups and enhanced networking.

Therefore, if you are in **Africa, it will be advisable to select the Cape Town region** for improved network performance especially because of **the placement groups and enhanced networking**.

In addition to measuring network throughput in bytes, CloudWatch provides metrics for packets sent and received.

Note, though, that packet metrics are only available in basic monitoring, at five-minute resolution.



Name	Description	Metric type
NetworkIn/NetworkOut	Number of bytes received/sent out on all network interfaces by the instance.	Resource: Utilization
NetworkPacketsIn/NetworkPacketsOut	Number of packets received/sent out on all network interfaces by the instance. *Only available at five-minute resolution.*	Resource: Utilization

Network in/out

- These metrics report the network throughput, **in bytes,** of your EC2.
- Drops or fluctuations **in the network in and network out in bytes** can be correlated with other application-level metrics to pinpoint possible issues.
- It is unlikely that your instances **will approach their network throughput limits unless** they are severely mismatched with your application’s needs.
- It is important to monitor the network so that one can determine if certain instances are receiving **considerably more network traffic than others**, which in such a case, you may wish to use a load balancer to distribute traffic more evenly.

CPU metrics

- EC2 instance types have a wide range of vCPU configurations, **so tracking CPU usage can help ensure that your instances are appropriately sized for your workload.**
- **CPU** – Central processing unit is the brain for your EC2 instance.
- Note that CloudWatch measures the percent utilization of the compute unit.
- Compute unit is the processing capacity or capability of the EC2 instance.

- It does not report the CPU **usage of the underlying hardware that the instance is being hosted on. (The host CPU utilization).**
- Remember EC2 is the server and has its own hardware where it is hosted on. On that hardware which we can look at it as a hypervisor 2, there is an OS system and CPU. The hardware also has an instant store volume which is ephemeral.
- There is another type of instance called T2 instance that is capable of bursting during the high traffic timelines to provide enough processing power or capacity for the demand.
- It is recommended that your EC2 should have 8 vCPU or core and 64 GB of RAM.
- This is ideal for applications that are not generally CPU intensive but may benefit from higher CPU capacity for brief intervals. See the T2 instance documentation for details on this instance type.
- A general estimation is that 1 vCPU = 1 Physical CPU Core. However, 1 vCPU is much stronger than the actual 1 Physical CPU Core.

CPU utilization

- CPU usage/utilization is one of the prime host-level metrics to monitor.
- Depending on the application, consistently high utilization levels may be normal because of the increased demand from the traffic.
- However, if performance is **degraded**, and the application is not constrained by disk I/O, memory, or network resources, then maxed-out CPU may indicate a resource bottleneck or **application performance problems.**
- **CPU** usually processes the activities in the server and therefore, they are mostly related to the performances.
- Bursting in AWS is a common term that means the ability of the T2 instance to come in and increase the processing capability of the server during the high traffic demand. It is important to know that this comes at a cost because extra CPU utilization than the standard level usually costs more money to the company.

- With Google Compute Engine preferring to go the sustained use discount route, Amazon EC2 is following a similar path The T2 instance types are ideal for running dev/test, blogs, and dynamic websites that do not demand **consistent CPU performance.**
- It will not be a surprise if AWS takes the concept of CPU credits beyond T2 instances to offer discounts for all **underutilized EC2 instances.**

CPU Credit Balance – Available CPU Limited

- For standard T2 instances with bursting, a burst can continue only if there are available CPU credits.
- This is why it is important to monitor your instance's balance.
- Credits are earned any time the instance is running below its baseline CPU performance level.
- This is the credit that is utilized by the T2 instances to burst during the high-demand throughput traffic.

CPU Credit Usage

CPU Usage Baseline

- One CPU credit is equivalent to one minute of 100 percent CPU utilization (or two minutes at 50 percent, etc.).
- Whenever an instance requires CPU performance above that instance type's baseline, of 100 percent utilization in 1 minute, the T2 instances will burst to cater for increased traffic at the expense of the CPU credit balance.
- You need to set your cloud watch to monitor the CPU usage so that you can know the CPU credit balance in case of bursting.
- If you have constant bursting, it will be appropriate to consider using instances that are optimized to handle high workloads.
- This is in the situation where you have limited T2 Bursting - The CPU Credit Balance

CPU Surplus Credit Balance

- In the case of **T2 Unlimited instances**, if the CPU credit balance is exhausted but burst performance is still required, the instance will consume additional credits to maintain greater CPU usage.
- This metric tracks the accumulated balance which will be added cost for the company. I believe the balance will CPU.

Surplus Credits Charged

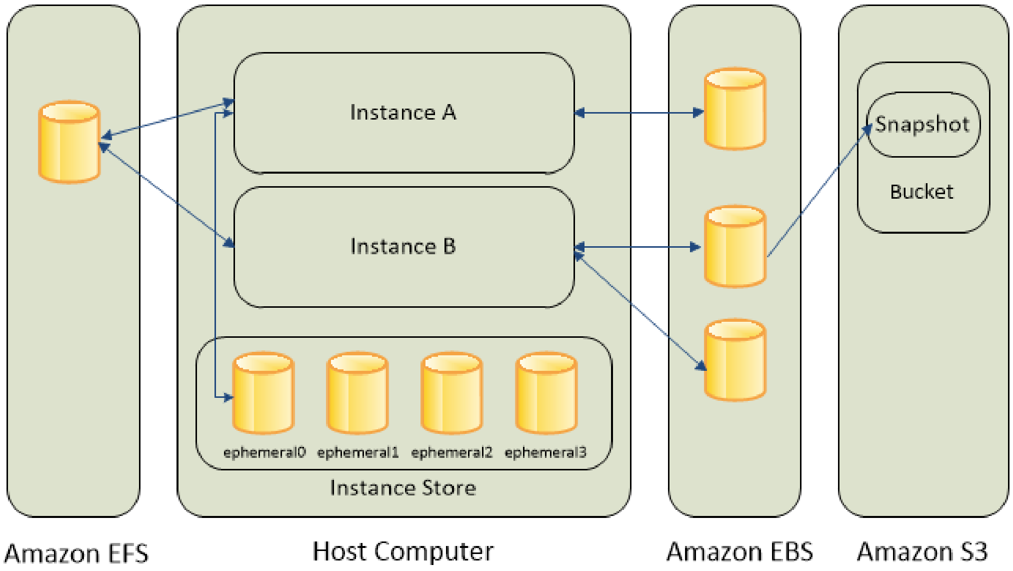
- This metric tracks the difference between the number of credits accumulated and the current credit balance that can be used to **pay down the surplus balance.**
- In other words, it is **a measure of extra credits that will result in additional charges.**

Surplus Credits Charged = CPU Surplus Credit balance (Overused beyond the normal baseline standard) - CPU Credit balance (Underused below the normal baseline standard).

Status checks

- EC2 status checks are, simply, checks on the status of an individual instance and of the AWS systems hosting it.
- Status checks are available at one-minute intervals.
- They provide a clear, high-level indication of an instance's health and whether there is a problem with either the larger AWS infrastructure or with the software or network configuration of the instance itself.

Metric to watch: Status check failed system



- This status check reports whether there are problems detected with the system hosting the instance.
- Generally, these are problems with the Amazon-administered computer on which your instance is hosted and are outside of your control for example, power loss.
- Possible resolutions include stopping and restarting an instance to switch it to a new host computer.
- (Keep in mind that **instance store–backed volumes** will be lost if the instance is stopped.) This check returns False (0) if an instance passes the system status check, and True (1) if it fails.

Metric to watch: Status check failed instance

Reasons why an instance can fail

Corrupted file system

The network configuration that is not balanced

- This check reports whether there are any problems detected with the instance itself and returns False (0) if an instance passes the status check and True (1) if it fails.
- Problems that might cause this check to fail include software or network configuration issues, a corrupted file system, etc.

- Amazon's troubleshooting tips offer causes and possible solutions for common errors that result in a failed status check.

Events

Events are scheduled changes in an instance's lifecycle. AWS may initiate events if problems are detected or if standard maintenance is required on an instance's host computer.

Events include:

- Stopping an instance. This is only applicable to EBS-backed instances, which retain their data and can be restarted. If restarted, the instance will be hosted on a new computer.
- Retiring an instance. This will terminate the instance and delete any attached volumes.
- Rebooting either the instance (again, only applicable to EBS-backed instances) or the host computer.
- System maintenance, possibly affecting the instance's performance or availability.
- AWS will inform users if an event has been scheduled for their instances.
- But you can also use **CloudWatch's events stream to track events and monitor upcoming changes to** your EC2 infrastructure that might degrade performance or affect data availability.
- This is particularly important for any **instance store volumes**—even if they are connected to an EBS-backed instance—as all data stored on those volumes is lost.
- Keeping an eye on your EC2 events will help you determine if you need to migrate data to a new instance before the current one is terminated or stopped.

Memory Metrics

- For many use cases, such as large, high-performance databases and in-memory applications, memory metrics are particularly vital to keeping an eye on your infrastructure and identifying problems and performance bottlenecks.
- **However, Amazon's CloudWatch does not report system-level memory metrics for instances.**
- Simply put, the cloud watch does not provide memory-level metrics for the instances.

Flow Logs Basics

You can create a flow log for a **VPC**, **a subnet**, or **a network interface**.

A network interface is the point of interconnection between a server (Instance) and a Private or Public network.

If you create a flow log for a subnet or VPC, each network interface in that subnet or VPC is monitored.

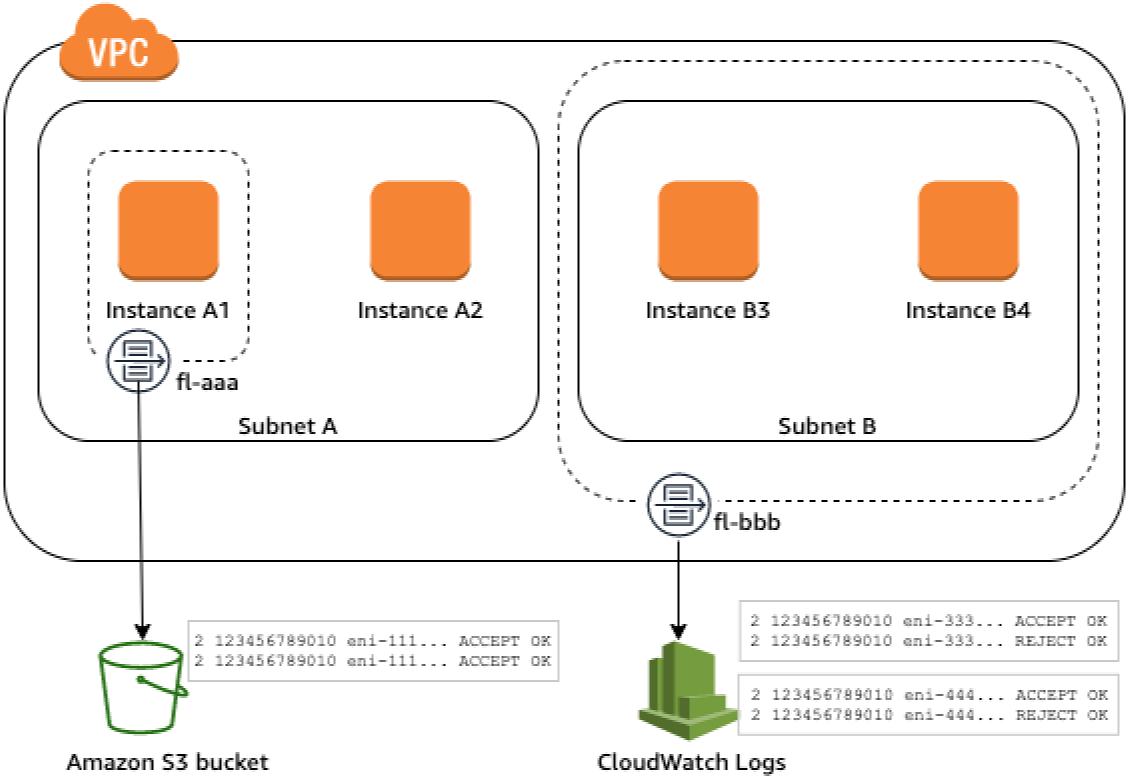
Flow log data for a monitored network interface is recorded as **flow log records**, which are log events consisting of fields that describe the traffic flow.

Flow logs records are important in showing the traffic flow in the VPC, Subnets, and Instances.

To create a flow log, you specify:

- The resource for which to create the flow log (Instances, and Subnets).
- The type of traffic to capture (**accepted traffic, rejected traffic, or all traffic**)

- The destinations to which you want to publish the flow log data (S3 buckets or Cloud Watch logs).



```
2 123456789010 eni-111... ACCEPT OK
2 123456789010 eni-111... ACCEPT OK
```

```
2 123456789010 eni-333... ACCEPT OK
2 123456789010 eni-333... REJECT OK
```

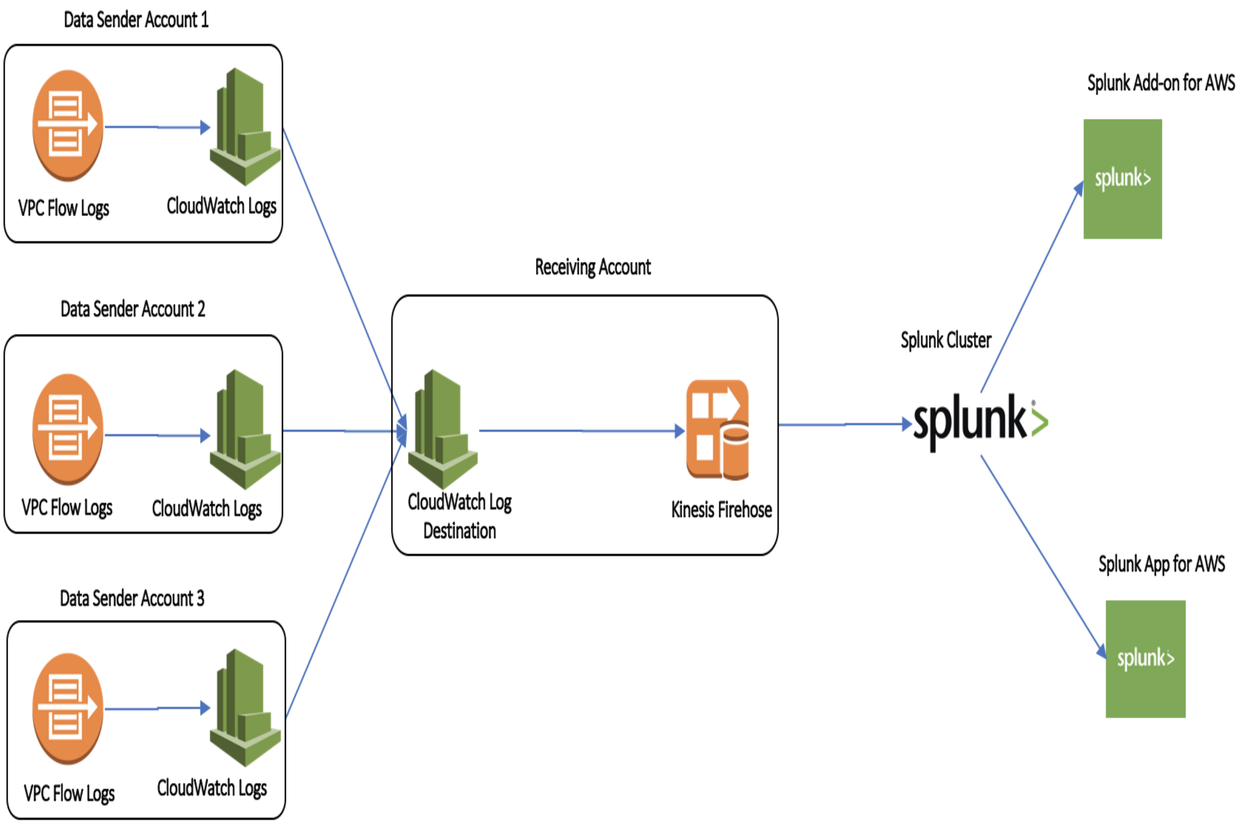
```
2 123456789010 eni-444... ACCEPT OK
2 123456789010 eni-444... REJECT OK
```

Regardless of the type of network interface, you must use the Amazon EC2 console or the **Amazon EC2 API** to create a flow log for a network interface.

In the following example, you create a flow log (fl-aaa) that captures **accepted traffic for the network interface for instance A1,** and publishes the **flow log records to an Amazon S3 bucket.**

One can create **a second flow log that captures all traffic for subnet B** and **publishes the flow log records to Amazon CloudWatch Logs**.

The flow log (fl-bbb) captures traffic for all network interfaces in subnet B. **There are no flow logs that capture traffic for instance A2's network interface.**



Prepared by Josh Wahome

302 235 9992

josh.kidfileapp@gmail.com