AWS EC2 Monitoring

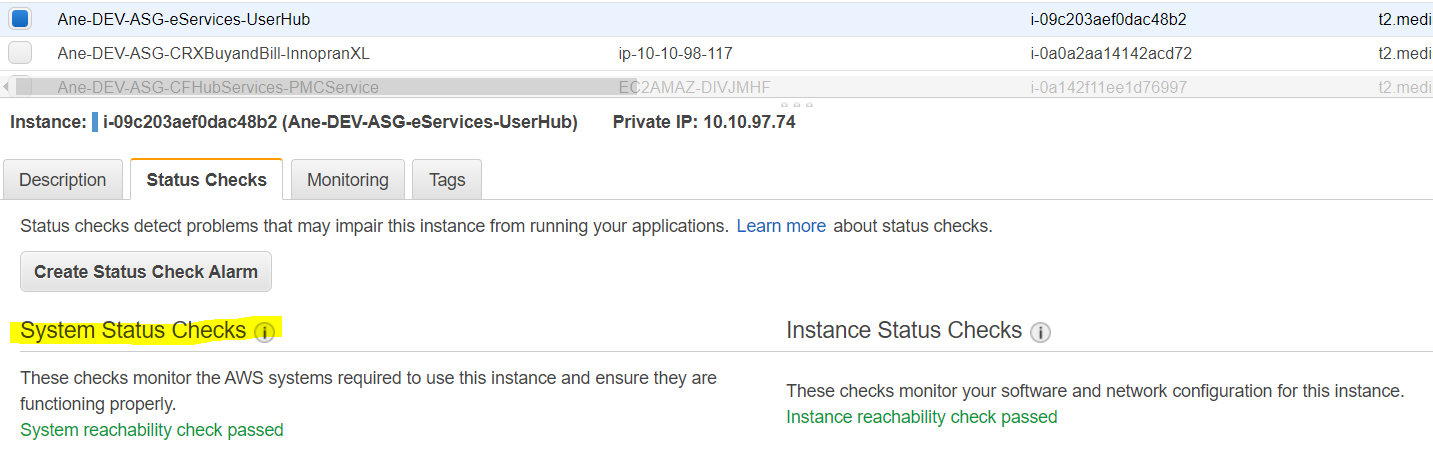
# EC2 Monitoring

## Status Checks

* Status monitoring help quickly determine whether EC2 has detected any problems that might prevent instances from running applications.
* EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.
* Status checks are performed every minute and each returns a pass or a fail status.
* If all checks pass, the overall status of the instance is **OK**.
* If one or more checks fail, the overall status is **Impaired**.
* **Status checks are built into EC2, so they cannot be disabled or deleted**.
* Status checks data augments the information that EC2 already provides about the intended state of each instance (such as pending, running, stopping) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).
* **Alarms can be created, or deleted, that are triggered based on the result of the status checks. for e.g., an alarm can be created to warn if status checks fail on a specific instance.**

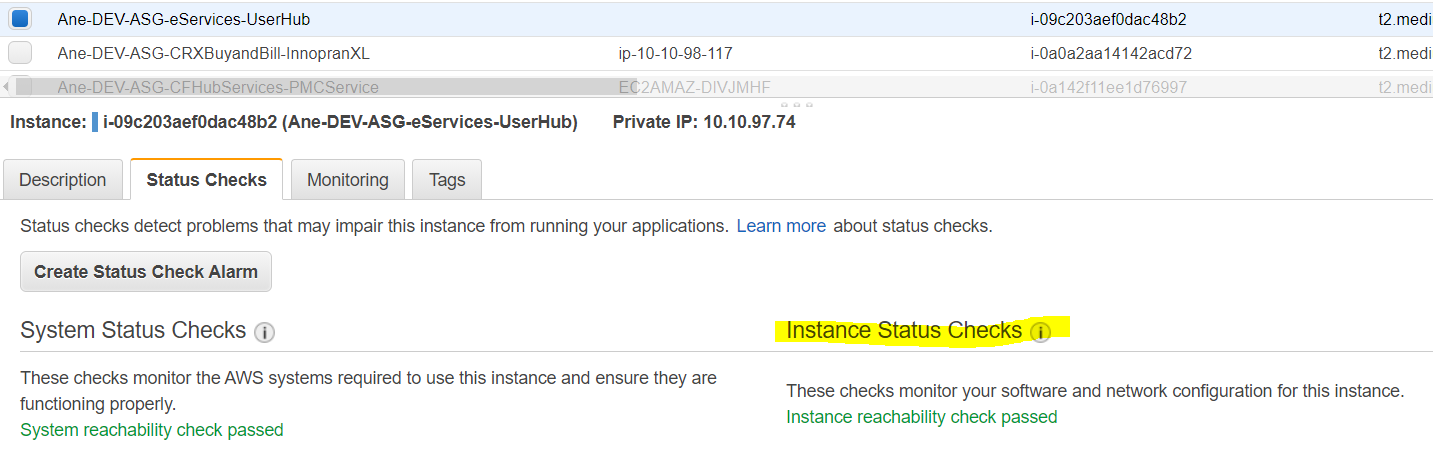
### System Status Checks

* This check verifies that your instance is reachable. We test that we are able to get network packets to your instance.
* When a system status check fails, one can either
  + wait for AWS to fix the issue
  + or resolve it by by stopping and restarting or terminating and replacing an instance
* System status checks failure might be cause of
  + Loss of network connectivity
  + Loss of system power
  + Software issues on the physical host
  + Hardware issues on the physical host



### Instance Status Checks

* monitor the software and network configuration of the individual instance
* When an instance status check fails, it can be resolved by either rebooting the instance or by making modifications in the operating system
* Instance status checks failure might be cause of
  + Failed system status checks
  + Misconfigured networking or startup configuration
  + Exhausted memory
  + Corrupted file system
  + Incompatible kernel



## CloudWatch Monitoring

* CloudWatch, helps monitor EC2 instances, which collects and processes raw data from EC2 into readable, near real-time metrics.
* Statistics are recorded for a period of two weeks, so that historical information can be accessed and used to gain a better perspective on how  
  the application or service is performing.
* By default Basic monitoring is enabled and EC2 metric data is sent to CloudWatch in 5-minute periods automatically
* Detailed monitoring can be enabled on EC2 instance, which sends data to CloudWatch in 1-minute periods.
* **Aggregating Statistics Across Instances/ASG/AMI ID**
  + Aggregate statistics are available for the instances that have detailed monitoring (at an additional charge) enable, which provides data in 1-minute periods
  + Instances that use basic monitoring are not included in the aggregates.
  + CloudWatch does not aggregate data across Regions. Therefore, metrics are completely separate between Regions.
  + CloudWatch returns statistics for all dimensions in the AWS/EC2 namespace, if no dimension is specified
  + The technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces published to CloudWatch.
  + Statistics include Sum, Average, Minimum, Maximum, Data Samples
  + With custom namespaces, the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point must be specified
* **CloudWatch alarms**
  + can be created to monitor any one of the EC2 instance’s metrics.
  + can be configured to automatically send you a notification when the metric reaches a specified threshold.
  + can automatically stop, terminate, reboot, or recover EC2 instances
  + can automatically recover an EC2 instance when the instance becomes impaired due to an underlying hardware failure a problem that requires AWS involvement to repair
  + **can automatically stop or terminate the instances to save costs (EC2 instances that use an EBS volume as the root device can be stopped  
    or terminated, whereas instances that use the instance store as the root device can only be terminated)**
  + can use **EC2ActionsAccess** IAM role, which enables AWS to perform stop, terminate, or reboot actions on EC2 instances
  + **If you have read/write permissions for CloudWatch but not for EC2, alarms can still be created but the stop or terminate actions won’t be performed on the EC2 instance**

## EC2 Metrics

* **CPUCreditUsage**
  + **(Only valid for T2 instances) The number of CPU credits consumed**during the specified period.
  + This metric identifies the amount of time during which physical CPUs  
    were used for processing instructions by virtual CPUs allocated to  
    the instance.
  + CPU Credit metrics are available at a 5 minute frequency.
* **CPUCreditBalance**
  + **(Only valid for T2 instances) The number of CPU credits that an instance has accumulated.**
  + This metric is used to determine how long an instance can burst beyond its baseline performance level at a given rate.
  + CPU Credit metrics are available at a 5 minute frequency.
* **CPUUtilization**
  + % of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance.
* **DiskReadOps**
  + Completed read operations from all instance store volumes available to the instance in a specified period of time.
* **DiskWriteOps**
  + Completed write operations to all instance store volumes available to the instance in a specified period of time.
* **DiskReadBytes**
  + Bytes read from all instance store volumes available to the instance.
  + This metric is used to determine the volume of the data the application reads from the hard disk of the instance.
  + This can be used to determine the speed of the application.
* **DiskWriteBytes**
  + Bytes written to all instance store volumes available to the instance.
  + This metric is used to determine the volume of the data the application writes onto the hard disk of the instance.
  + This can be used to determine the speed of the application.
* **NetworkIn**
  + The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to an application on a single instance.
* **NetworkOut**
  + The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic to an application on a single instance.
* **NetworkPacketsIn**
  + The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance.
  + This metric is available for basic monitoring only
* **NetworkPacketsOut**
  + The number of packets sent out on all network interfaces by the instance. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance.
  + This metric is available for basic monitoring only.
* **StatusCheckFailed**
  + Reports if either of the status checks, StatusCheckFailed\_Instance and StatusCheckFailed\_System, that has failed.
  + Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.
  + Status check metrics are available at 1 minute frequency
* **StatusCheckFailed\_Instance**
  + Reports whether the instance has passed the Amazon EC2 instance status check in the last minute.
  + Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.
  + Status check metrics are available at 1 minute frequency
* **StatusCheckFailed\_System**
  + Reports whether the instance has passed the EC2 system status check in the last minute.
  + Values for this metric are either 0 (zero) or 1 (one.) A zero indicates that the status check passed. A one indicates a status check failure.
  + Status check metrics are available at a 1 minute frequency

## AWS Certification Exam Practice Questions

1. In the basic monitoring package for EC2, Amazon CloudWatch provides the following metrics:
   1. Web server visible metrics such as number failed transaction requests
   2. Operating system visible metrics such as memory utilization
   3. Database visible metrics such as number of connections
   4. **Hypervisor visible metrics such as CPU utilization**
2. Which of the following requires a custom CloudWatch metric to monitor?
   1. **Memory Utilization of an EC2 instance**
   2. CPU Utilization of an EC2 instance
   3. Disk usage activity of an EC2 instance
   4. Data transfer of an EC2 instance
3. A user has configured CloudWatch monitoring on an EBS backed EC2 instance. If the user has not attached any additional device, which of the below mentioned metrics will always show a 0 value?
   1. **DiskReadBytes (**For EC2 when the user is monitoring the  
      EC2 instances, it will capture the 7 Instance level and 3 system check parameters for the EC2 instance. Since this is an EBS backed instance, it will not have ephermal storage attached to it. Out of the 7 EC2 metrics, the 4 metrics DiskReadOps, DiskWriteOps, DiskReadBytes and DiskWriteBytes are disk related data and available only when there is ephermal storage attached to an instance. For an EBS backed instance without any additional device, this data will be 0.**)**
   2. NetworkIn
   3. NetworkOut
   4. CPUUtilization
4. A user is running a batch process on EBS backed EC2 instances. The batch process starts a few instances to process Hadoop Map reduce jobs, which can run between 50 – 600 minutes or sometimes for more time. The user wants to configure that the instance gets terminated only when the process is completed. How can the user configure this with CloudWatch?
   1. **Setup the CloudWatch action to terminate the instance when the CPU utilization is less than 5%**
   2. Setup the CloudWatch with Auto Scaling to terminate all the instances
   3. Setup a job which terminates all instances after 600 minutes
   4. It is not possible to terminate instances automatically
5. An AWS account owner has setup multiple IAM users. One IAM user only has CloudWatch access. He has setup the alarm action, which stops the EC2 instances when the CPU utilization is below the threshold limit. What will happen in this case?
   1. It is not possible to stop the instance using the CloudWatch alarm
   2. CloudWatch will stop the instance when the action is executed
   3. The user cannot set an alarm on EC2 since he does not have the permission
   4. **The user can setup the action but it will not be executed if the user does not have EC2 rights**
6. A user has launched 10 instances from the same AMI ID using Auto Scaling. The user is trying to see the average CPU utilization across all instances of the last 2 weeks under the CloudWatch console. How can the user achieve this?
   1. **View the Auto Scaling CPU metrics**(Refer [AS Instance Monitoring](https://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-monitoring.html))
   2. Aggregate the data over the instance AMI ID (Works but needs detailed monitoring enabled)
   3. The user has to use the CloudWatchanalyser to find the average data across instances
   4. It is not possible to see the average CPU utilization of the same AMI ID since the instance ID is different