AWS SysOps Associate Exam Notes

Description

Notes and information that were collected while studying and prepping for the AWS SysOps Associate Exam.

Topic	Answer
Exam Time:	80 Minutes
No. Questions:	60 Questions
Question Types:	Scenario and Multiple Choice
Passing Score:	~ 70%
Validity Period:	2 years
Renewal Exam:	1/2 price off

Monitoring:

Monitoring is accomplished through the usage of CloudWatch, which is a service to monitor your AWS resources as well as the applications that you run on AWS.

CloudWatch Monitoring:

- Can monitor EC2, DynamoDB, RDS, custom metrics generated by your applications and services, and any log files that your applications generate, etc..
- EC2 will by default monitor your instances @5 minute intervals
- EC2 instances can monitor your instances @1 minute intervals if the 'detailed monitoring' option is set on the instance
- By default CloudWatch will monitor CPU, Network, Disk, and Status Checks
- 2 types of Status Checks:
  - System Status Checks:
    - Checks the underlying physical host
    - Checks for loss of network connectivity
    - Checks for loss of system power
    - Checks for software issues on the physical host
    - Checks for hardware issues on the physical host
    - Best way to resolve issues is to stop the instance and start it again (will switch physical hosts)
  - Instance Status Checks
    - Checks the VM itself
    - Checks for failed system status checks
    - Checks for mis-configured networking or startup configs
    - Checks for exhausted memory
    - Checks for corrupted file systems
    - Checks for an incompatible kernel
    - Best way to troubleshoot is rebooting the instance or modifying the instance OS
- RAM utilization is a custom metric
- By default CloudWatch metrics are stored for 2 weeks
- Can retrieve data that is longer than 2 weeks using the GetMetricStatistics API endpoint, or by using third party tools
- Can retrieve data from any terminated EC2 or ELB instance for up to 2 weeks after its termination
- Many default metrics for many default services are 1 min, but it can be 3-5 minutes depending on the service
- Custom metrics have a minimum 1 minute granularity
- Alarms can be created to monitor any CloudWatch metric in your account
- Alarms can be set on charges on your AWS bill
- Within the alarm, actions can be set, triggering things like lambda functions, or SNS notifications if the alarm threshold is reached

Configuring custom metrics:

- RAM utilization for example must be set up as a custom metric
- yum install -y perl-Switch perl-Sys-Syslog perl-LWP-Protocol-https
- wget http://ec2-downloads.s3.amazonaws.com/cloudwarch-samples/CloudWatchMonitoringScripts-v1.1.0.zip
- unzip CloudWatchMonitoringScripts-v1.1.0.zip
- rm -fr CloudWatchMonitoringScripts-v1.1.0.zip
- cd aws-scripts-mon
- ./mon-put-instance-data.pl --mem-util --verify --verbose (dry run no data will be sent to CloudWatch)
- ./mon-put-instance-data.pl --mem-util --mem-used --mem-avail (set this up on 1/5 minute cron job)

Monitoring EBS:

- GP2 volumes have a base of 3 IOPS per GB of volume size
- Maximum volume size is 16 GB
- Maximum IOPS size of 10K IOPS total (after which you need to move to provisioned IOPS storage tier)
- Can burst performance on the volume up to 3K IOPS
- Bursting uses I/O credits
- Each volume receives an initial I/O credit balance of 5.4 million I/O credits
- This is enough to sustain the max burst performance of 3K IOPS for 30 minutes
- When not going over provisioned IO level (bursting) you earn credits back
- Don't need to know the calculation to replenish the credit balance
- When creating new or restoring a volume from snapshots, the first time you access the storage block, you can see a 5 to 50 % loss of IOPS due the volume either needing to be wiped clean or instantiated from a snapshot
- Performance is restored after the data is accessed once
- To avoid the performance hit, volumes can be pre-warmed
- For a new volume, you should write to all blocks before using the volume
- For a volume that has been restored from a snapshot, you should read all blocks that have data before using the volume
- Instructions for pre-warming Linux volumes can be found here
- EBS CloudWatch Metrics:
    - VolumeReadBytes
    - VolumeWriteBytes
        - Provides info on the I/O operations in a specified period of time
        - The SUM statistic reports the total number of bytes transferred during the period
        - The AVG statistic reports the average size of each I/O operation during the period

        - The SampleCount statistic reports the total number of I/O operations during the period
        - The Minimum and Maximum statistics are not relevant for this metric
        - Data is only reported to CloudWatch when the volume is active
        - If the volume is idle, no data is reported to CloudWatch
    - VolumeReadOps
    - VOlumeWriteOps
        - The total number of I/O operations in a specified period of time
        - To calculate the AVG IOPS for the period, divide the total operations in the period by the number of seconds in that period
    - VolumeTotalReadTime
    - VolumeTotalWriteTime
        - The total number of seconds spent by all operations that completed in a specified period of time
        - If multiple requests are submitted a the same time, the total could be greater than the length of the period
    - VolumeIdleTime
        - The total number of seconds in a specified period of time when no read or write operations were submitted
    - VolumeQueueLength
        - Then number of read and write operation requests waiting to be completed in a specified period of time
    - VolumeThroughputPercentage
        - Used with Provisioned IOPS (SSD) volumes only
        - The percentage of IOPS delivered of the total IOPS provisioned for an EBS volume
        - Provisioned IOPS SSD volumes deliver within 10% of the provisioned IPS performance 99.9% of the time over a given year
        - During a write, if there are no other pending I/O requests in a minute, the metric value will be 100%
        - A volume's I/O performance may become degraded temporarily due to an action that was taken (such as creating a snapshot of a volume during peak usage, or running the volume on a non-EBS-optimized instance, or accessing data on the volume for the first time, if the volume wasn't pre-warmed)
    - VolumeConsumedReadWriteOps
        - Used with Provisioned IOPS (SSD) volumes only
        - The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time
        - I/O operations that are smaller than 256K each count as 1 consumed IOPS
        - I/O operations that are larger than 256K are counted in 256K capacity units
- VolumeQueueLength can come up frequently, know what it is
- Volume Status Checks:
    - OK:
        - I/O Enabled status:

      - Enabled (I/O Enabled or I/O Auto-Enabled)
    - I/O Performance Status:
      - Only available for Provisioned IOPS (IO1) volumes
      - Normal (Volume performance is as expected)
  - Warning:
    - I/O Enabled status:
      - Enabled (I/O Enabled or I/O Auto-Enabled)
    - I/O Performance Status:
      - Only available for Provisioned IOPS (IO1) volumes
      - Degraded (Volume performance is below expectations)
  - Impaired:
    - I/O Enabled status:
      - Enabled (I/O Enabled or I/O Auto-Enabled)
      - Disabled (volume is off-line and pending recovery, or is waiting for the user to enable I/O)
    - I/O Performance Status:
      - Only available for Provisioned IOPS (IO1) volumes
      - Stalled (Volume performance is severely impacted)
  - Insufficient Data:
    - I/O Enabled status:
      - Enabled (I/O Enabled or I/O Auto-Enabled)
      - Insufficient Data
    - I/O Performance Status:
      - Only available for Provisioned IOPS (IO1) volumes
      - Insufficient Data
- Degraded, Severely Degraded = Warning
- Stalled or Not Available = Impaired
-

Monitoring RDS:

- 2 types of monitoring:
  - Monitor by metrics (CloudWatch monitoring):
    - Per-Database Metrics
    - By Database Class
    - By Database Engine
    - Across All Databases
  - Monitor by events (RDS monitoring):
    - Located in Events tab
    - Events of everything that has happened with your instance
    - Can set event subscriptions which work like SNS topics
    - Events like fail-overs can be a notifying event using subscriptions

- Available RDS Metrics:
  - BinLogDiskUsage
  - CPUUtilization
  - DatabaseConnections
  - DiskQueueDepth
  - FreeableMemory
  - FreeStorageSpace
  - ReplicaLag (Seconds)
  - SwapUsage
  - ReadIOPS
  - WriteIOPS
  - ReadLatency
  - WriteLatency
  - ReadThroughput
  - WriteThroughput
  - NetworkReceiveThroughput
  - NetworkTransmitThroughput
- Have general idea of what each of the RDS metrics do
- DatabaseConnections, DiskQueueDepth, FreeStorageSpace, ReplicaLag (Seconds), ReadIOPS, WriteIOPS, ReadLatency, WriteLatency are all important ones to know

Monitoring ELB:

- Monitored every 60 seconds provided there is traffic
- Only reports when requests are flowing through the LB
- If there are no requests or data for a given metric, the metric will not be reported to CloudWatch
- If there are requests flowing through the LB, ELB will measure and send metrics for that LB in 60 second intervals
- Available Metrics:
  - HealthyHostCount:
    - The count of the number of healthy instances in each AZ
    - Hosts are declared healthy if they meet the threshold for the number or consecutive health checks that are successful
    - Hosts that have failed more health checks then the value of the unhealthy threshold are considered unhealthy
    - If cross-zone is enabled, the count of the number of healthy instances is calculated for all AZs
    - Preferred Statistic: Average
  - UnHealthyHostCount:
    - The count of the number of unhealthy instances in each AZ

        - Hosts that have failed more health cheeks than the value of the unhealthy threshold are considered unhealthy
        - If cross-zone is enabled, the count of the number of unhealthy instances is calculated for all AZs
        - Instances may become unhealthy due to connectivity issues, health checks returning non-200 responses (in the case of HTTP or HTTPS health checks), or timeouts when performing the health check
        - Preferred Statistic: Average
    - RequestCount:
        - The count of the number of completed requests that were received and routed to the back end instances
        - Preferred Statistic: Sum
    - Latency:
        - Measures the time elapsed in seconds after the request leaves the load balancer until the response is received
        - Preferred Statistic: Average
    - HTTPCode_ELB_4XX
        - The count of the number of HTTP 4XX client error codes generated by the load balancer when the listener is configured to use HTTP or HTTPS protocols. Client errors are generated when a request is malformed or is incomplete
        - Preferred Statistic: Sum
    - HTTPCode_ELB_5XX
        - The count of the number or HTTP 5XX server error codes generated by the load balancer when the listener is configured to use HTTP or HTTPS protocols
        - This metric does not include any responses generated by back end instances
        - The metric is reported if there are no back-end instances that are healthy or registered to the load balancer, or if the request rate exceeds the capacity of the instances or the load balancers
        - Preferred Statistic: Sum
    - HTTPCode_Backend_2XX:
    - HTTPCode_Backend_3XX:
    - HTTPCode_Backend_4XX:
    - HTTPCode_Backend_5XX:
        - The count of the number of HTTP response codes generated by back-end instances
        - Metric does not include any response codes generated by the load balancer
        - The 2XX class status codes represent successful actions
        - The 3XX class status codes indicate that the user agent requires action
        - The 4XX class status code represents client errors
        - The 5XX class status code represents back-end server errors
        - Preferred Statistic: Sum
    - BackendConnectionErrors:

    - The count of the number of connections that were not successfully established between the LB and the registered instances
    - The LB will retry when there are connection errors, so the count can exceed the request rate
    - Preferred Statistic: Sum
  - SurgeQueueLength:
    - A count of the total number of requests that are pending submission to a registered instance
    - Preferred Statistic: Max
  - SpilloverCount:
    - A count of the total number of requests that were rejected due to the queue being full
    - Preferred Statistic: Sum
- Have an idea of what each metric does
- Important metrics to note are SurgeQueueLength & SpilloverCount

Monitoring Elasticache:

- Consists of 2 different engines:
  - Memcached
  - Redis
- When it comes to monitoring cache engines, there are 4 monitoring points:
  - CPU Utilization
    - Memcached:
      - Multi-threaded
      - Handles loads of up to 90%
      - If > 90% add more nodes to the cluster
    - Redis:
      - Single-threaded
      - Take 90 and / No. cores to determine scale point
      - Will not have to calculate Redis CPU utilization in exam
  - Swap Usage
  - Evictions
  - Concurrent Connections

White Paper Review:

Documentation built with MkDocs.