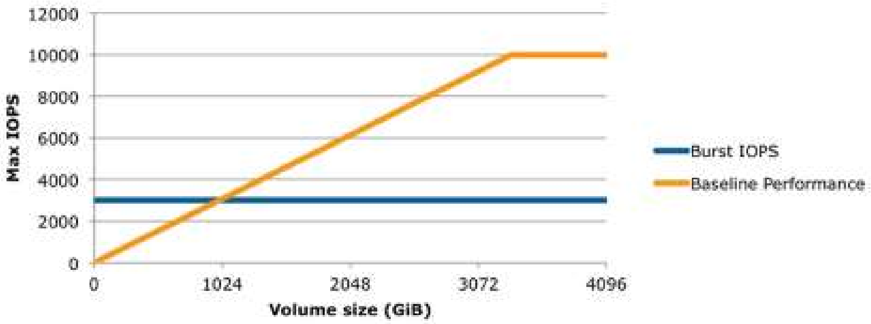
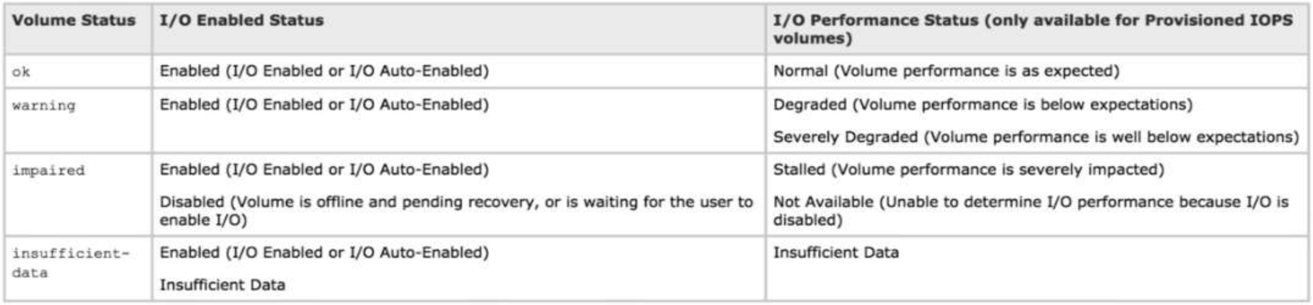
| | |
|---|---|
| **CloudWatch** | It is a monitoring service to monitor AWS resources, as well as applications that you run on AWS.<br>CloudWatch can monitor things like:<br>1. Compute<br>Autoscaling Groups<br>Elastic Load Balancers<br>Route53 Health Checks<br>2. Storage & Content Delivery<br>EBS Volume<br>Storage Gateways<br>CloudFront<br>3. Databases & Analytics<br>Dynamo DB<br>Elasticache Nodes<br>RDS Instances<br>Elastic MapReduce Job Flows<br>Redshift<br>4. Other<br>SNS Topics<br>SQS Queues<br>Opsworks<br>CloudWatch Logs<br>Estimated Charges on AWS Bill<br>5. EC2 Instance (Host Level Default Metrics)<br>CPU<br>Network<br>Disk<br>Status Check<br>**Exam Tips**: RAM utilization and Storage of disks couldn't monitor by using host level metrics. |
| **CloudWatch Alarms** | We can create an alarm to monitor any Amazon CloudWatch metric in our account. This can include EC2 CPU Utilization, Elastic Load Balancer Latency and Charges on AWS bill. We can set the appropriate thresholds in which to trigger the alarms and also set what actions should be taken if an alarm state is reached. |

<table>
<tr>
<td><b>CloudWatch Exam Tips</b></td>
<td><b>Custom Metrics</b> - Minimum granularity is 1 minute.<br>RAM and Disk utilizations are custom metric.<br><b>Terminated Instances</b> - We can retrieve data from any terminated EC2 or ELB instance after its termination. CloudWatch logs by default are stored indefinitely.<br><b>Metric Granularity</b> -<br>1 minute for detailed monitoring.<br>5 minute for standard monitoring.<br><b>CloudWatch can be used on premise</b> - It’s not restricted to just AWS resources. Can be on premise too. Just need to download and install the SSM agent and CloudWatch agent.</td>
</tr>
<tr>
<td><b>Monitoring EBS</b></td>
<td>4 Different Volume Types:<br>General Purpose (SSD) - gp2, Provisioned IOPS (SSD) - io1, through put Optimized (HDD) - st1 and Cold (HDD) – sc1<br><b>IOPS &amp; Volume Examples</b>



<b>Volume Status Check</b>


<table>
<tr><th>Volume Status</th><th>I/O Enabled Status</th><th>I/O Performance Status (only available for Provisioned IOPS volumes)</th></tr>
<tr><td>ok</td><td>Enabled (I/O Enabled or I/O Auto-Enabled)</td><td>Normal (Volume performance is as expected)</td></tr>
<tr><td>warning</td><td>Enabled (I/O Enabled or I/O Auto-Enabled)</td><td>Degraded (Volume performance is below expectations)<br>Severely Degraded (Volume performance is well below expectations)</td></tr>
<tr><td>impaired</td><td>Enabled (I/O Enabled or I/O Auto-Enabled)<br>Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)</td><td>Stalled (Volume performance is severely impacted)<br>Not Available (Unable to determine I/O performance because I/O is disabled)</td></tr>
<tr><td>insufficient-data</td><td>Enabled (I/O Enabled or I/O Auto-Enabled)<br>Insufficient Data</td><td>Insufficient Data</td></tr>
</table>
</td>
</tr>
</table>

| | |
|---|---|
| **ELB - Elastic Load Balancer** | 3 Types of ELBs<br>1. Application Load Balancer<br>2. Network Load Balancer<br>3. Classic Load Balancer<br>ELB - Monitoring Types<br>4 Different ways to monitor load balancer<br>1. CloudWatch metrics<br>2. Access logs<br>3. Request tracing<br>4. CloudTrail logs<br>**CloudWatch Vs CloudTrail**<br>✓ CloudWatch monitors performance.<br>✓ CloudTrail monitors API calls in the AWS platform |

| | |
|---|---|
| **Monitoring ELB - Load Balancer Types** | **3 Different Types of Elastic Load balancers**<br>1. Application Load Balancer<br>2. Network Load Balancer<br>3. Classic Load Balancer<br><br>**4 Different ways to monitor load balancers**<br>1. CloudWatch metrics<br>ELB publishes data points to Amazon CloudWatch. CloudWatch enables us to retrieve statistics about those points as an ordered set of time series data, known as metrics. Think of a metric as a variable to monitor, and the data points as the values of that variable over time.<br>For example, we can monitor the total number of healthy targets for a load balancer over a specified time period. Each data point has an associated time stamp and an optional unit of measurement.<br>2. Access logs<br>ELB provides access logs that capture detailed information about requests sent to load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. We can use this access logs to analyze traffic patterns and troubleshoot issues.<br>It is an optional feature of ELB that is disabled by default. The captured logs are stored in the S3 bucket.<br>**"Access logs can store data where the EC2 instances has been deleted." For some reason our application has a load of 5xx errors which is only reported by customers a couple of days after the event. If we are not storing the web server logs anywhere persistent, it is still possible to trace these 5xx error using Access logs which would be stored on S3.**<br>3. Request tracing (*Available for Application Load Balancers Only*)<br>We can use request tracing to track HTTP requests from clients to targets or other services. When the load balancer receives a request from a client, it adds or updates the Trace id header before sending the request to the target. Any services or applications between the load balancer and the target can also add or update this header.<br>4. CloudTrail logs<br>We can use AWS CloudTrail to capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in S3. We can use these logs to determine which calls were made, the source IP address where the call came from, who made the call, when the call was made, and so on.<br><br>**CloudWatch Vs CloudTrail**<br>1. **CloudWatch** monitors performance.<br>Monitor EC2 resources like CPU utilizations, Memory utilizations etc.<br>2. **CloudTrail** monitors API calls in the AWS platform.<br>Monitor EC2 instances, RDS instances, Users, S3 buckets etc. |

<table>
<tr>
<td><b>Monitoring ElastiCache</b></td>
<td>
<b>ElastiCache consists of two engines</b><br>
1. MemCached<br>
2. Redis<br>
ElastiCache provides metrics that enable us to monitor our clusters. We can access these metrics through CloudWatch.<br>
ElastiCache provides both host-level metrics (for example, CPU usage) and metrics that are specific to the cache engine software (for example, cache gets and cache misses). These metrics are measured and published for each Cache node in 60-second intervals.<br>
We should consider setting CloudWatch alarms on certain key metrics, so that we will be notified if our cache cluster's performance starts to degrade.<br>
<br>
<b>ElastiCache monitor our caching engines there are 4 important things to look at:</b><br>
1. CPU Utilization
<table>
<tr><th>MemCached</th><th>Redis</th></tr>
<tr>
<td>• Multi-threaded.<br>• Can handle loads of up to 90%. If it exceeds 90% add more nodes to the cluster</td>
<td>• Not multi-threaded. To determine the point in which to scale, take 90 and divided by the number of cores.<br>For example, suppose we are using a cache.m1.node, which has four cores. In this case, the threshold for CPU utilization would be (90/4), or 22.5%</td>
</tr>
</table>
2. Swap Usage<br>
3. Evictions
<table>
<tr><th>MemCached</th><th>Redis</th></tr>
<tr>
<td>• There is no recommended settings. Choose a threshold based off our application.<br>• Either Scale Up (i.e. increase the memory of existing nodes) or Scale Out (add more nodes).</td>
<td>• There is no recommended settings. Choose a threshold based off our application.<br>• Only Scale Out (add read replicas)</td>
</tr>
</table>
4. Concurrent Connections<br>
MemCached &amp; Redis<br>
• There is no recommended settings. Choose a threshold based off our application.<br>
• If there is a large and sustained spike in the number of concurrent connections this can either mean a large traffic spike or our application is not releasing connections as it should be.
</td>
</tr>
</table>

| | |
|---|---|
| **CloudWatch Dashboard** | Dashboards are multi-region and can display any widget to any region. To add the widget, change to the region that you need and then add the widget to the dashboard. |
| **AWS Organizations** | **AWS Organizations**<br>• Centrally managed policies across multiple AWS accounts.<br>• Control access to AWS services.<br>• Automate AWS account creation and management.<br>• Consolidate billing across multiple AWS accounts. |
| **AWS Recourse Group & Tagging** | **Tags**<br>• It's a key value pairs attached to AWS resources.<br>• Metadata<br>• Tags can sometimes be inherited<br>- Autoscaling, CloudFormation, and Elastic Beanstalk can create other resources.<br>**Resource Groups**<br>Resource groups make it easy to group your resources using the tags that are assigned to them. You can group resources that share one or more tags.Resource group contain information such as;<br>• Region<br>• Name<br>• Health Checks<br>Specific information<br>• For EC2 - Public & Private IP Address<br>• For ELB - Port Configurations<br>• For RDS – Database Engine etc.<br>**Two Types of Resource Groups**<br>1. Classic Resource Groups<br>2. AWS Systems Manager |
| **EC2 Pricing** | **On Demand** - Allow you to pay a fixed rate by the hour with no commitment.<br>**Reserved** - Provide you with a capacity reservation, and offer a significant discount on the hourly charge for an instance. 1 - 3 year terms.<br>**Spot** - Enable you to bid whatever price you want for instance capacity, providing for even greater savings if your application have flexible start and end times.<br>**Dedicated Hosts** - Physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software license. |

| | |
|---|---|
| **AWS Config Rules** | **Permission needed for Config:**<br>▪ **AWS Config requires an IAM Role with**<br>1. Read only permission to the recorded resources.<br>2. Write access to S3 logging bucket<br>**3.** Publish access to SNS<br>**Restricted Access:**<br>1. Users need to be authenticated with AWS and have the appropriate permissions set via IAM policies to gain access.<br>2. Only Admins needing to set up and manage Config require full access.<br>3. Provide read only permissions for Config day-to-day use.<br>**Monitoring Config:**<br>1. Use CloudTrail with Config to provide deeper insight into resources.<br>2. Use CloudTrail to monitor access to Config, such as someone stopping the Config Recorder.<br>FAQ – https://aws.amazon.com/config/faq/ |
| **CloudWatch, CloudTrail and Config** | 1. **CloudWatch** monitors performance.<br>Monitor EC2 resources like CPU utilizations, Memory utilizations etc.<br>2. **CloudTrail** monitors API calls in the AWS platform.<br>Monitor EC2 instances, RDS instances, Users, S3 buckets etc.<br>**3. AWS Config** records the state of AWS environment and can notify you of changes. |
| **Health Dashboards** | **Service Health Dashboard** - Shows the health of each AWS Service as a whole per region.<br>**Personal Health Dashboard -** It provides alerts when AWS is experiencing events that may impact you. |