Monitoring the AWS EC2 Cloud

eG Enterprise V6

Restricted Rights Legend

The information contained in this document is confidential and subject to change without notice. No part of this document may be reproduced or disclosed to others without the prior permission of eG Innovations Inc. eG Innovations Inc. makes no warranty of any kind with regard to the software and documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Trademarks

Microsoft Windows, Windows NT, Windows 2003, and Windows 2000 are either registered trademarks or trademarks of Microsoft Corporation in United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Copyright

Table of Contents

Table of Figures

1

Introduction

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizeable computing capacity—literally, server instances in Amazon's data centers—that you use to build and host your software systems. You can get access to the infrastructure resources that EC2 provides by using APIs, or web tools and utilities.

With EC2, you use and pay for only the capacity that you need. This eliminates the need to make large and expensive hardware purchases, reduces the need to forecast traffic, and enables you to automatically scale your IT resources to deal with changes in requirements or spikes in popularity related to your application or service.

With many mission-critical applications now being delivered via the cloud, end-users have come to expect from the cloud the same quality of service that local service deployments are known to deliver. This means that even the slightest dip in performance levels will not be tolerated!

A sudden non-availability of the cloud, no matter how brief, or a slowdown/failure of any of its regions/availability zones/instances, can make it impossible for cloud providers to build and launch mission-critical services on the cloud and for consumers to access these services for prolonged periods. If you are a (public or private) cloud service provider therefore, your primary concerns would be - can people access my service? Is the self service portal up? Can users see their VMs? Can users connect to their VMs? If not, you need to be able to determine why the problem is happening – is it the web front-end? is it due to the virtualization platform? is it due to the SAN? etc. The action you take depends on what you diagnose as being the root-cause of the problem. Besides problem diagnosis, you are also interested in understanding how you can get more out of your current cloud investments. You want to be able to see how to balance load across your servers to serve a maximum number of users and how you can optimize the capacity of the infrastructure without sacrificing on performance. You need performance management "FOR" the cloud.

eG Enterprise is a unique solution that can provide you performance management **FROM** the cloud, **OF** the cloud and **FOR** the cloud!

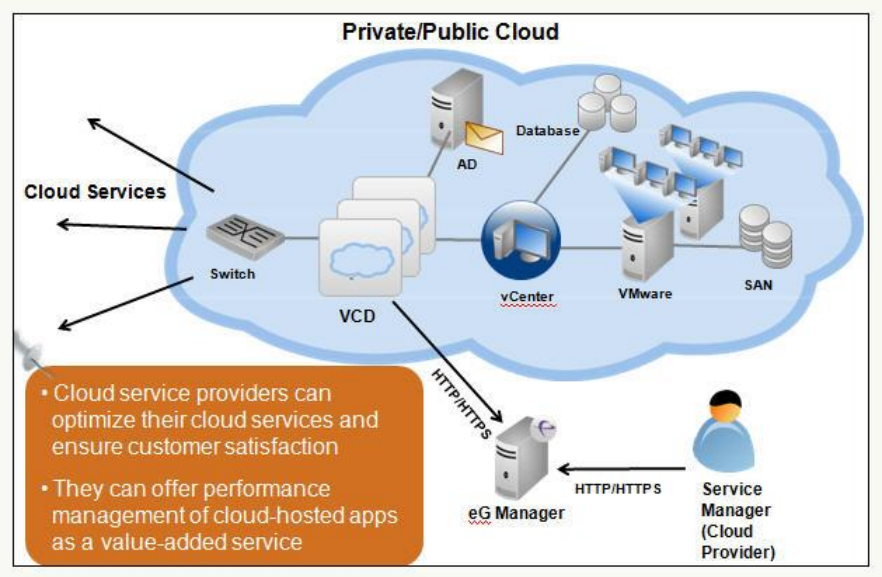


Figure 1.1: How eG monitors the cloud?

To deliver performance management **FOR** the AWS EC2 cloud in particular, the solution offers two specialized monitoring models - the *AWS EC2 Cloud* model and the *AWS EC2 Region* model. The *AWS EC2 Cloud* monitoring model provides you with proactive updates on the overall health and status of the cloud and points you to unavailable regions/availability zones and resource-hungry instances in the cloud. To zoom into the health of specific regions and the instances operating within those regions, use the *AWS EC2 Region* model.

This document engages in detailed discussions on both the models.

2

Monitoring the AWS EC2 Cloud

Figure 2.1 depicts the *AWS EC2 Cloud* monitoring model that eG Enterprise offers out-of-the-box for monitoring the Amazon EC2 cloud.

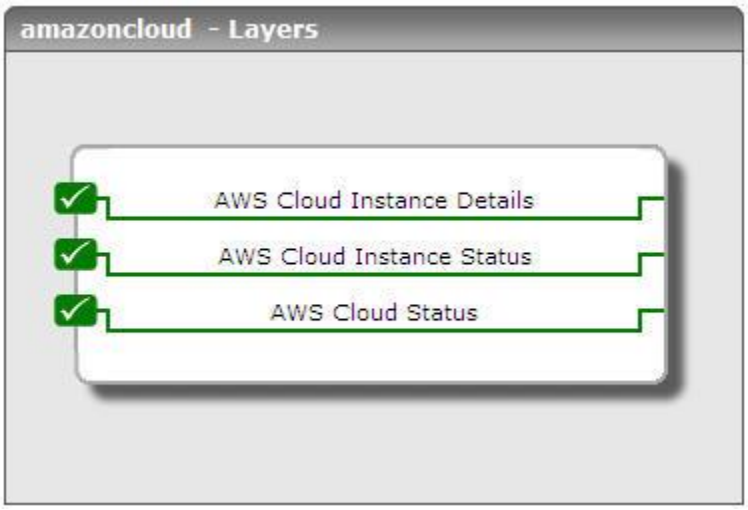


Figure 2.1: Layer model of the AWS EC2 Cloud

Each layer of this model is mapped to tests that reveal the availability of the cloud and whether the regions/availability zones/instances on the cloud are accessible. Using these statistics, cloud administrators can find quick and accurate answers for the following critical performance queries:

a. Is web-based (HTTP/HTTPS) access to the cloud available?

b. Does it take an unreasonably long time to establish contact with the cloud?

c. How many regions does the cloud support? What are they?

d. Is any region unavailable?

e. Were any connectivity issues experienced while attempting to connect to a region? If so, which region is this?

f. How many availability zones exist in each region? What are they?

g. Is any availability zone currently unavailable? If so, which one is it?

h. Is the default region on the cloud accessible? If so, is it taking too long to connect to the default region?

i. Are all instances on the cloud accessible over the network?

j. Are any instances powered off currently?

k. Were any instances launched/removed recently? If so, which ones are these?

l. What type of instances are resource-intensive?

m. Is any particular instance consuming too much CPU?

n. Is the network traffic to/zofrom any instance unusually high?

o. Is the disk I/O of instances optimal?

p. Was any instance rebooted recently? If so, which one is it?

To enable the eG agent to collect these useful metrics, the following pre-requisites need to be fulfilled:

- The eG agent should be deployed on a remote Windows host in the environment.
- The **eGurkhaAgent** service of the remote agent should run using 'domain administrator' privileges. To know how to set this up, refer to the *eG User Manual*.
- Each test executed by the remote agent uses the AWS API to collect the required metrics. To enable the tests to access the AWS API, you need to configure the tests with the access key and password of a user with a valid AWS user account. To obtain this access key, do the following:
    - Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page.
    - Provide the details of the user for whom you wish to create the AWS account.
    - Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2.
    - Once the payment is made, the user will be automatically signed in to the AWS account.
    - From the newly created AWS account, you can request for an **access key**. You will be provided with an **access key** and a corresponding **secret key**.

Note:

The eG agent reports metrics for only those regions, availability zones, and instances on the cloud that the configured AWS user account is allowed to access.

- Some tests require the **AWS CloudWatch** service to be enabled. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. For enabling this service, you need to pay CloudWatch fees. Refer to the AWS web site for the fee details.

The sections that will follow discuss each of the layers of Figure 2.1 in great detail.

2.1 The AWS Cloud Status Layer

Using the tests mapped to this layer, you can promptly detect the non-availability of the cloud, inaccessibility of regions and availability zones on the cloud, and connection bottlenecks experienced while connecting to the cloud or its components.

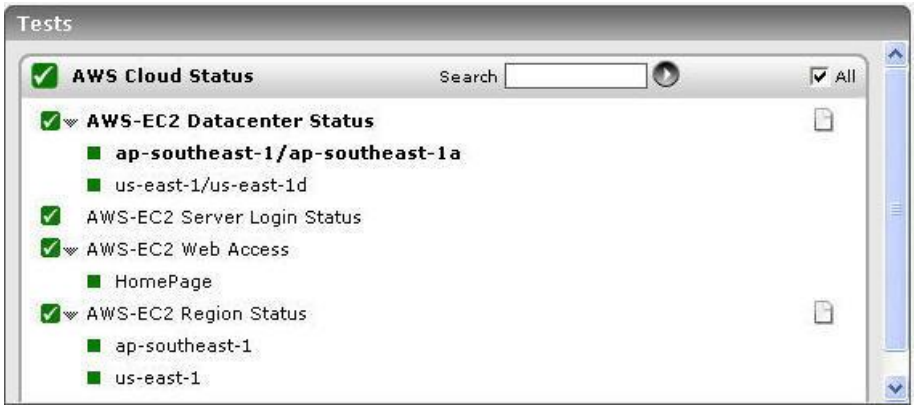


Figure 2.2: The test associated with the Hardware layer

2.2 AWS-EC2 Availability Zones Test

Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.). Correspondingly, EC2 is available to use in different *Regions*. Each Region contains multiple distinct locations called *Availability Zones* (illustrated in the following diagram). Each Availability Zone is engineered to be isolated from failures in other Availability zones and to provide inexpensive, low-latency network connectivity to other zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

If users complaint that their server instances are inaccessible, you may want to know whether it is because of the non-availability of the availability zone within which the instances have been launched. This test auto-discovers the

regions and availability zones on the Amazon EC2 Cloud, and reports the availability of each zone.

Purpose	Auto-discovers the regions and availability zones on the Amazon EC2 Cloud, and reports the availability of each zone
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees**. **For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** In the **EXCLUDE INSTANCE** text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*. 8. **REPORT INSTANCE DATACENTER** - By default, this test reports the availability of only those availability zones that contain one/more instances. Accordingly, this flag is set to **true** by default. If you want the test to report metrics for all availability zones, regardless of whether/not they host instances, set this flag to **false**.

<table>
<tr><td></td><td colspan="3">9. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
10. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
11. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
12. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option against DETAILED DIAGNOSIS. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability.
• Both the bad and normal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each availability zone in each region of the AWS EC2 Cloud being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**Availability:** Indicates whether/not this availability zone in this region is currently available.	Number	The value *0* indicates that the availability zone is *Not Available* and the value *100* indicates that it is *Available*. If an availability zone fails, then all server instances operating within that zone will also be rendered unavailable. If you host all your Amazon EC2 instances in a single location that is affected by such a failure, your instances will be unavailable, thereby bringing your entire application to a halt. On the other hand, if you have instances distributed across many Availability Zones and one of the instances fails, you can design your application so the instances in the remaining Availability Zones handle any requests.

2.2.1 AWS-EC2 Server Logins Test

This test attempts to connect to the default region in the cloud; in the process, the test reports whether the configured AWS user account is able to access the cloud-based infrastructure or not, and if so, how quickly the connection with the infrastructure was established.

If a user is denied access to a server instance on a clod, or if a user experiences a significant delay in connecting to his/her instances, you can use this test to validate the user credentials and to figure out whether any connectivity issues exist.

Purpose	Attempts to connect to the default region in the cloud; in the process, the test reports whether the configured AWS user account is able to access the cloud-based infrastructure or not, and if so, how quickly the connection with the infrastructure was established
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** In the EXCLUDE INSTANCE text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
11. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option against DETAILED DIAGNOSIS. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability.
• Both the bad and normal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the AWS EC2 Cloud being monitored</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td></td><td>Default region availability:
Indicates whether or not the the test is able to access the default region on the cloud using the configured AWS user account .</td><td>Percent</td><td>The value 0 indicates that the region is not accessible, and the value 100 indicates that it is accessible. If the default region is inaccessible, it could be owing to any one of the following reasons:
q. The cloud is unavailable;
r. The configured AWS account does not have the access rights to the default region;
s. The test has been configured with incorrect login credentials.</td></tr>
</table>

	Response time: Indicates the time taken by the test to establish a connection with the default region on the cloud.	Secs	A low value is desired for this measure. A high value or a consistent increase in this value could indicate connection bottlenecks.

2.2.2 AWS-EC2 Web Access Test

This test emulates a user accessing a web page on the cloud via HTTP(S), and reports whether that page is accessible or not. In the process, the test indicates the availability of the cloud over the web, and the time it took for the agent to access the cloud over the web. This way, issues in web-based access to the cloud come to light.

Purpose	Emulates a user accessing a web page (by default, the login page) on the cloud via HTTP(S), and reports whether that page is accessible or not. In the process, the test indicates the availability of the cloud over the web, and the time it took for the agent to access the cloud over the web.
Target	An AWS-EC2 cloud
Agent deploying this test	A remote agent
Configurable parameters for this test	1. **TEST PERIOD** – How often should the test be executed 2. **URL** – The web page being accessed. While multiple URLs (separated by commas) can be provided, each URL should be of the format **URL name:URL value**. **URL name** is a unique name assigned to the URL, and the **URL value** is the value of the URL. By default, the url parameter is set to *HomePage:http://aws.amazon.com/ec2/*, where *HomePage* is the **URL name**, and *http://aws.amazon.com/ec2* is the **URL** value. You can modify this default setting to configure any URL of your choice - eg., the URL of the login page to your cloud-based infrastructure. 3. **HOST** - The host for which the test is to be configured. 4. **PORT** - The port to which the specified **HOST** listens 5. **COOKIEFILE** – Whether any cookies being returned by the web server need to be saved locally and returned with subsequent requests 6. **PROXYHOST** – The host on which a web proxy server is running (in case a proxy server is to be used) 7. **PROXYPORT** – The port number on which the web proxy server is listening 8. **PROXYUSERNAME** – The user name of the proxy server 9. **PROXYPASSWORD** – The password of the proxy server 10. **CONFIRM PASSWORD** – Confirm the password by retyping it here. 11. **CONTENT** – Is a set of instruction:value pairs that are used to validate the content being returned by the test. If the CONTENT value is *none:none*, no validation is performed. The number of pairs specified in this text box, must be equal to the number of URLs being monitored. The instruction should be one of *Inc* or *Exc*. *Inc* tells the test that for the content returned by the test to be valid, the content must include the specified value (a

simple string search is done in this case). An instruction of *Exc* instructs the test that the test's output is valid if it does not contain the specified value. In both cases, the content specification can include wild card patterns. For example, an Inc instruction can be *Inc:*Home page**. An Inc and an Exc instruction can be provided in quick succession in the following format: *Inc:*Home Page*,Exc:*home*.

12. **CREDENTIALS** – The HttpTest supports HTTP authentication. The **CREDENTIALS** parameter is to be set if a specific user name / password has to be specified to login to a page. Against this parameter, the **URLname** of every configured url will be displayed; corresponding to each listed **URLname**, a **Username** text box and a **Password** text box will be made available. These parameters will take either of the following values:

    a. a valid **Username** and **Password** for every configured **URLname**

    b. *none* in both the **Username** and **Password** text boxes of all configured **URLnames** (the default setting), if no user authorization is required

    Where NTLM (Integrated Windows) authentication is supported, valid **CREDENTIALS** are mandatory. In other words, a *none* specification will not be supported in such cases. Therefore, in this case, against each configured **URLname**, you will have to provide a valid **Username** in the format: *domainname*|*username,* followed by a valid **Password**.

    Please be sure to check if your web site requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many sites use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the username and password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the this test.

13. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default.

14. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default.

15. **PROXY DOMAIN** and **PROXY WORKSTATION** - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.

16. **TIMEOUT** - Here, specify the maximum duration (in seconds) for which the test will wait for a response from the server. The default TIMEOUT period is 30 seconds.

Outputs of the test	One set of outputs for every URL being monitored		
Measurements	**Measurement**	**Measurement Unit**	**Interpretation**

of the test	**Availability:** This measurement indicates whether the test was able to access the configured URL or not.	Percent	Availability failures could be caused by several factors such as the web server process(es) (hosting the configured web page) being down, the web server being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web server is overloaded. Availability is determined based on the response code returned by the test. A response code between 200 to 300 indicates that the configured web page is available.
	Total response time: This measurement indicates the time taken by the test to access this URL.	Secs	Response time being high denotes a problem. Poor response times may be due to an overload. If the URL accessed involves the generation of dynamic content, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.
	Tcp connection availability: This measure indicates whether the test managed to establish a TCP connection to this URL.	Percent	Failure to establish a TCP connection may imply that either the web server process hosting the web page is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the web page may start functioning properly again.
	Tcp connect time: This measure quantifies the time for establishing a TCP connection to the configured URL.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds).
	Server response time: This measure indicates the time period between when the connection was established and when the test sent back a HTTP response header to the client.	Secs	While the total response time may depend on several factors, the server response time is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
	Response code: The response code returned by the test for the simulated request	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
	Content length: The size of the content returned by the test	Kbytes	Typically the content length returned by the test for a specific URL should be the same across time. Any change in this metric may indicate the need for further investigation.

	Content validity: This measure validates whether the test was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, "Invalid Login" is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string "Invalid Login," in the above scenario content validity would have a value 0.

2.2.3 AWS-EC2 Regions Test

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Each Region is completely independent.

By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements.

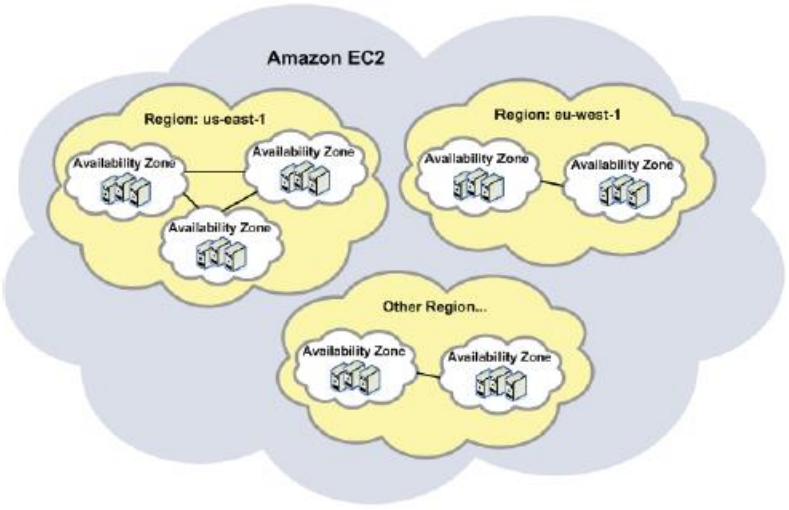


Figure 2.3: Regions and Availability zones

If a region is unavailable, then users to that region will not be able to access the server instances launched in that region. This may, in turn, adversely impact the user experience with the cloud. To avoid such an unpleasant outcome, it is best to periodically monitor the availability of each region, so that unavailable regions can be quickly and accurately identified, and the reasons for their non-availability remedied.

This test performs periodic availability checks on each region on the cloud, and reports the status of the individual regions. In addition, the test also indicates the time taken for connecting to a region so that, regions with connectivity issues can be isolated.

Purpose	Performs periodic availability checks on each region on the cloud, and reports the status of the individual regions. In addition, the test also indicates the time taken for connecting to a region so that, regions with connectivity issues can be isolated
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each region of the AWS EC2 Cloud being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Availability:
Indicates whether/not this this region is currently available.</td><td>Number</td><td>The value 0 indicates that the region is Not Available and the value 100 indicates that it is Available.</td></tr>
<tr><td>Response time:
Indicates the time taken to connect to this region.</td><td>Secs</td><td>A low value is typically desired for this measure. A high value or a consistent increase in this value could be indicative of connection bottlenecks.
Compare the value of this measure across regions to know which region takes the longest to connect to.</td></tr>
</table>

2.3 The AWS Cloud Instance Status Layer

The tests mapped to this layer take stock of the total number of instances (that are available for the configured AWS user account) on the cloud, and points you to the following:

- The powered-off instances
- The newly launched/removed instances
- Instances that are unavailable over the network

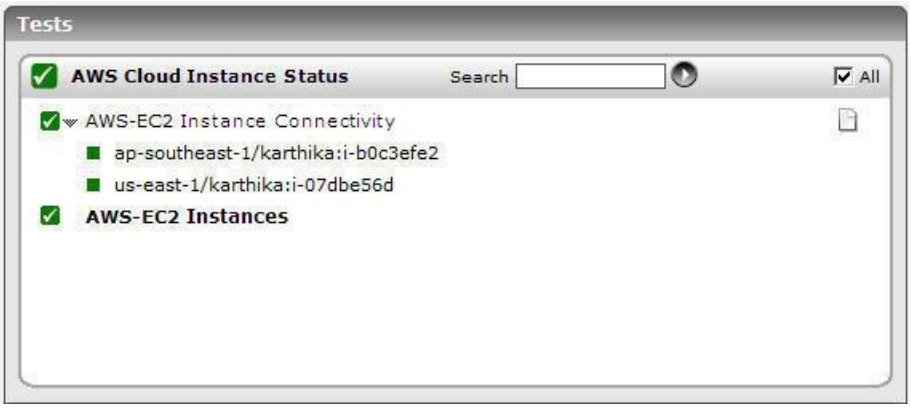


Figure 2.4: The tests mapped tothe AWS Cloud VM Status layer

2.3.1 AWS-EC2 Instance Connectivity Test

Sometimes, an instance could be in a powered-on state, but the failure of the operating system or any fatal error in internal operations of the instance could have rendered the instance inaccessible to users. In order to enable you to promptly detect such 'hidden' anomalies, this test periodically runs a connectivity check on each instance available for the configured AWS user account, and reports whether the instances are accessible over the network or not.

Purpose	Runs a connectivity check on each instance available for the configured AWS user account, and reports whether the instances are accessible over the network or not
Target of the test	An AWS-EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees**. **For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

	8. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default. 9. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default. 10. **PROXY DOMAIN** and **PROXY WORKSTATION** - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.		
Outputs of the test	One set of results for each instance available for the configured AWS user account		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Avg network delay: Indicates the average delay between transmission of packets to this instance and receipt of the response to the packet at the source.	Secs	An increase in network latency could result from misconfiguration of the router(s) along the path, network congestion, retransmissions at the network, etc.
	Min network delay: The minimum time between transmission of a packet and receipt of the response back.	Secs	A significant increase in the minimum round-trip time is often a sure sign of network congestion.
	Packet loss: Indicates the percentage of packets lost during transmission from source to target and back.	Percent	Packet loss is often caused by network buffer overflows at a network router or by packet corruptions over the network. The detailed diagnosis for this measure provides a listing of routers that are on the path from the external agent to target server, and the delays on each hop. This information can be used to diagnose the hop(s) that could be causing excessive packet loss/delays.
	Network availability of Instance: Indicates whether the network connection to this instance is available or not.	Percent	A value of 100 indicates that the instance is accessible over the network. The value 0 indicates that the instance is inaccessible. Typically, the value 100 corresponds to a *Packet loss* of 0.

2.3.2 AWS-EC2 Instances Test

An Amazon Machine Image (AMI) contains all information necessary to boot instances of your software. For example, an AMI might contain all the software to act as a web server (e.g., Linux, Apache, and your web site) or it might contain all the software to act as a Hadoop node (e.g., Linux, Hadoop, and a custom application). After an AMI is launched, the resulting running system is called an **instance**. All instances based on the same AMI start out identical and any information on them is lost when the instances are terminated or fail.

Users with valid AWS user accounts can sign into the EC2 cloud to view and use available instances, or purchase and launch new ones. With the help of this test, you can determine the total number of instances that are currently available for the configured AWS user account, the number of instances that were newly purchased/terminated, and the count of powered-off instances.

Purpose	Determine the total number of instances that are currently available for the configured AWS user account, the number of instances that were newly purchased/terminated, and the count of powered-off instances
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
11. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option against DETAILED DIAGNOSIS. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability.
• Both the bad and normal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the AWS EC2 Cloud being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Total instances:
Indicates the total number of instances currently available for the configured AWS user account.</td><td>Number</td><td>The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances available for use for the configured AWS account, regardless of the current state of the instances.</td></tr>
<tr><td>Instances powered on:
Indicates the total number of instances that are currently powered-on.</td><td>Number</td><td>The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-on instances available for use for the configured AWS account.</td></tr>
</table>

	Instances powered off: Indicates the total number of instances that are currently powered-off.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-off instances available for the configured AWS account.
	Added instances: Indicates the total number of instances that were newly purchased by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly purchased and launched by the configured AWS user account.
	Removed instances: Indicates the total number of instances that were newly terminated by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly terminated/removed by the configured AWS user account.

2.4 The AWS Cloud Instance Details Layer

The tests mapped to this layer auto-discover the server instances that are available for the configured AWS user account on the cloud, and reports the uptime and the resource usage of the individual instances.

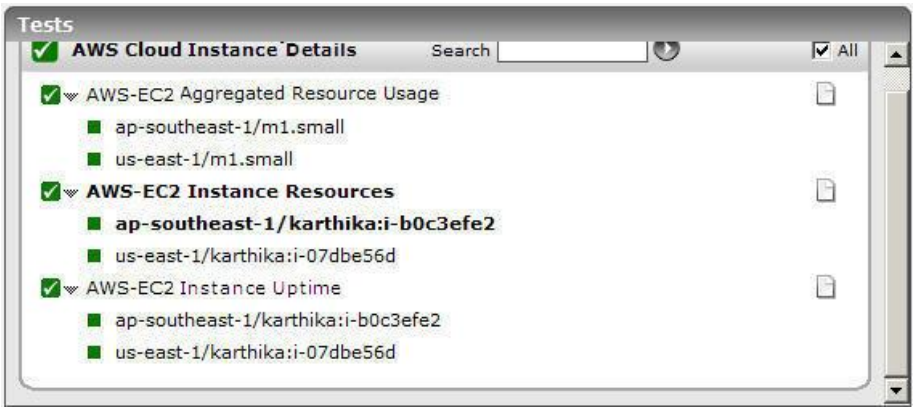


Figure 2.5: The tests mapped to the AWS Cloud VM Details layer

2.4.1 AWS-EC2 Aggregated Resource Usage Test

When users launch an instance using the AWS management console, they need to specify the instance type. An instance type is a specification that defines the memory, CPU, storage capacity, and hourly cost for an instance.

Some instance types are designed for standard applications, whereas others are designed for CPU-intensive applications, or memory-intensive applications, etc. The different instance types offered by the AWS EC2 cloud are as follows:

Type	CPU	Memory	Local Storage	Platform	I/O	Name
Small	1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)	1.7 GB	160 GB instance storage (150 GB plus 10 GB root partition)	32-bit	Moderate	m1.small
Large	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	7.5 GB	850 GB instance storage (2 x 420 GB plus 10 GB root partition)	64-bit	High	m1.large
Extra Large	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)	15 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	m1.xlarge
Micro	Up to 2 EC2 Compute Units (for short periodic bursts)	613 MB	None (use Amazon EBS volumes for storage)	32-bit or 64-bit	Low	t1.micro
High-CPU Medium	5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)	1.7 GB	350 GB instance storage (340 GB plus 10 GB root partition)	32-bit	Moderate	c1.medium
High-CPU Extra Large	20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)	7 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	c1.xlarge
High-Memory Extra Large	6.5 EC2 Compute Units (2 virtual cores with 3.25 EC2 Compute Units each)	17.1 GB	420 GB instance storage (1 x 420 GB)	64-bit	Moderate	m2.xlarge
High-Memory Double Extra Large	13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)	34.2 GB	850 GB instance storage (1 x 840 GB plus 10 GB root partition)	64-bit	High	m2.2xlarge
High-Memory Quadruple Extra Large	26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)	68.4 GB	1690 GB instance storage (2 x 840 GB plus 10 GB root partition)	64-bit	High	m2.4xlarge
Cluster Compute	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)	23 GB	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cc1.4xlarge
Cluster GPU	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture), plus 2 NVIDIA Tesla M2050 "Fermi" GPUs	22 GB (see note after this table)	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cg1.4xlarge

By closely monitoring the CPU usage and the network and disk I/O of each instance type, and comparing these metrics across instance types, you can quickly isolate resource-intensive types. Once again, the test will report metrics for only those types of instances that were launched by the AWS user account configured for the test.

Purpose	Closely monitors the CPU usage and the network and disk I/O of each instance type, and enables usage comparison across instance types, so as to quickly isolate resource-intensive types
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each type of instance launched by the configured AWS user account</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>CPU utilization:
Indicates the percentage of allocated CPU consumed by all instances of this type.</td><td>Percent</td><td>A high value for this measure indicates that one/more instances of a type are utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on the instances.
Compare the value of this measure across types to identify the types of instances that are CPU-intensive.</td></tr>
<tr><td>Incoming network traffic:
Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to all the instances of this instance type.</td><td>KB/Sec</td><td>Compare the values of these measures across instance types to quickly identify the types of instances that are utilizing the network bandwidth excessively.</td></tr>
</table>

	Outgoing network traffic: Indicates the volume of outgoing network traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to all the instances of a particular instance type.	KB/Sec	
	Disk reads: Indicates the rate at which data is read from the disks of all instances of this type.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
	Disk writes: Indicates the rate at which data is written to the disks of all instances of this type.	KB/Sec	
	Disk read operations: Indicates the rate at which disk read operations were performed on the disks of all instances of this type.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
	Disk write operations: Indicates the rate at which disk write operations were performed performed on the disks of all instances of this type.	Operations/Sec	

2.4.2 AWS-EC2 Instance Resources Test

Tracking the CPU usage, disk and network I/O of every instance launched by a configured AWS user account will provide administrators with valuable insights into how well the instances are utilizing the allocated resources. The **AWS-ECS VM Resource Usage** test does just that. This test auto-discovers the instances available for the configured AWS user account, and reports the resource usage of each instance so that, administrators can quickly compare the usage metrics across instances and pinpoint which instance is resource-hungry.

Purpose	Auto-discovers the instances available for the configured AWS user account, and reports the resource usage of each instance so that, administrators can quickly compare the usage metrics across instances and pinpoint which instance is resource-hungry
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each instance launched by the configured AWS user account</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>CPU utilization:
Indicates the percentage of allocated CPU consumed by this instance.</td><td>Percent</td><td>A high value for this measure indicates that an instance is utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on that instance.
Compare the value of this measure across instances to identify the CPU-intensive instances.</td></tr>
<tr><td>Incoming network traffic:
Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to this instance.</td><td>KB/Sec</td><td rowspan="2">Compare the values of these measures across instances to quickly identify the instance that is utilizing the network bandwidth excessively.</td></tr>
<tr><td></td><td>Outgoing network traffic:
Indicates the volume of outgoing network traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to this instance.</td><td>KB/Sec</td></tr>
</table>

	Disk reads: Indicates the rate at which data is read from the disks of this instance.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance. By comparing the values of these measures across instances, you can accurately determine which instance is performing I/O-intensive operations.
	Disk writes: Indicates the rate at which data is written to the disks of this instance.	KB/Sec	
	Disk read operations: Indicates the rate at which disk read operations are performed on this instance.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance. By comparing the values of these measures across instances, you can accurately determine which instance is performing I/O-intensive operations.
	Disk write operations: Indicates the rate at which disk write operations were performed on this instance.	Operations/Sec	

2.4.3 AWS-EC2 Instance Uptime Test

In cloud-based environments, it is essential to monitor the uptime of server instances launched on the cloud. By tracking the uptime of each of the instances, administrators can determine what percentage of time an instance has been up. Comparing this value with service level targets, administrators can determine the most trouble-prone areas of the infrastructure hosted on the cloud.

In some environments, administrators may schedule periodic reboots of their instances. By knowing that a specific instance has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on an instance.

This test monitors the uptime of each instance available to the configured AWS user account.

Purpose	Monitors the uptime of each instance available to the configured AWS user account
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the AWS-EC2 VM Resource Usage and AWS-EC2 VM Aggregate Resource usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">
8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
11. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability
• Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each instance launched by the configured AWS user account</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Has the instance been rebooted?:
Indicates whether this instance has been rebooted during the last measurement period or not.</td><td>Boolean</td><td>If this measure shows 1, it means that the instance was rebooted during the last measurement period. By checking the time periods when this metric changes from 0 to 1, an administrator can determine the times when this instance was rebooted.</td></tr>
</table>

	Uptime of the instance during the last measure period: Indicates the time period that the instance has been up since the last time this test ran.	Secs	If the instance has not been rebooted during the last measurement period and the agent has been running continuously, this value will be equal to the measurement period. If the instance was rebooted during the last measurement period, this value will be less than the measurement period of the test. For example, if the measurement period is 300 secs, and if the instance was rebooted 120 secs back, this metric will report a value of 120 seconds. The accuracy of this metric is dependent on the measurement period - the smaller the measurement period, greater the accuracy.
	Total uptime of the instance: Indicates the total time that this intance has been up since its last reboot.	Mins	Administrators may wish to be alerted if an instance has been running without a reboot for a very long period. Setting a threshold for this metric allows administrators to determine such conditions.

3

Monitoring the AWS EC2 Region

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Availability Zones are distinct locations within a Region that are engineered to be isolated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region. By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements.

The *AWS EC2 Region* model offered by eG Enterprise monitors a specific region on the cloud and reports the availability and responsiveness of that region.

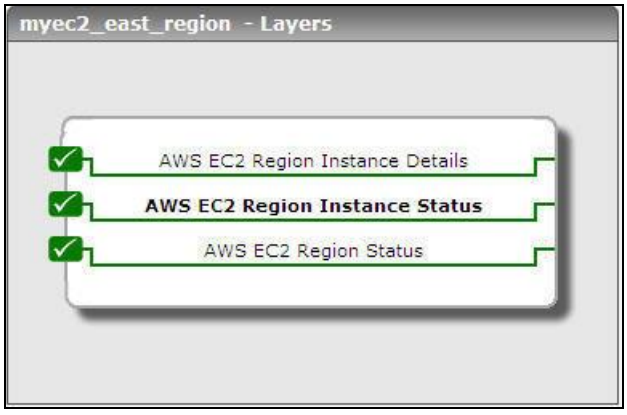


Figure 3.1: The layer model of the AWS EC2 Region

In addition, using a single eG agent installed on a remote Windows host in the environment, the model auto-discovers the IP address and the operating system of the instances launched on the cloud, periodically checks the powered-on status of each of the instances, continously assesses how each instance is utilizing the allocated resources, and thus promptly alerts you to unavailable and resource-hungry instances. As the solution also automatically determines what applications have been deployed on the instances, whenever one of these applications experience slowdowns, administrators can use the eG solution to instantly and accurately diagnose the root-cause of the slowdown - is it owing to the corresponding instance being unavailable or the application being resource-hungry?

Using the metrics so reported, administrators can ascertain the following:

- Is web-based (HTTP/HTTPS) access to the region available?
- Does it take an unreasonably long time to establish contact with the region?
- How many availability zones exist in the monitored region? What are they?
- Is any availability zone currently unavailable? If so, which one is it?

- Are all instances launched in the region accessible over the network?
- Are any instances powered off currently?
- Were any instances launched/removed recently? If so, which ones are these?
- What type of instances are resource-intensive?
- Is any particular instance consuming too much CPU?
- Is the network traffic to/from any instance unusually high?
- Is the disk I/O of instances optimal?
- Was any instance rebooted recently? If so, which one is it?

To enable the eG agent to collect these useful metrics, the following pre-requisites need to be fulfilled:

- The eG agent should be deployed on a remote Windows host in the environment.
- The **eGurkhaAgent** service of the remote agent should run using 'domain administrator' privileges. To know how to set this up, refer to the *eG User Manual*.
- Each test executed by the remote agent uses the AWS API to collect the required metrics. To enable the tests to access the AWS API, you need to configure the tests with the access key and password of a user with a valid AWS user account. To obtain this access key, do the following:
    - Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page.
    - Provide the details of the user for whom you wish to create the AWS account.
    - Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2.
    - Once the payment is made, the user will be automatically signed in to the AWS account.

Note:

The eG agent reports metrics for only availability zones and instances in a region that the configured AWS user account is allowed to access.

- Some tests require the **AWS CloudWatch** service to be enabled. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. For enabling this service, you need to pay CloudWatch fees. Refer to the AWS web site for the fee details.

The sections that follow will discuss each layer of Figure 3.1 elaborately.

3.1 The AWS EC2 Region Status Layer

Using the tests mapped to this layer, you can promptly detect the non-availability of a target region and the availability zones in that region, and connection bottlenecks experienced while connecting to the cloud or its components.

Figure 3.2: The tests mapped to the AWS EC2 Region Status layer

3.1.1 EC2 - Availability Zones Test

Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.). Correspondingly, EC2 is available to use in different *Regions*. Each Region contains multiple distinct locations called *Availability Zones* (illustrated in the following diagram). Each Availability Zone is engineered to be isolated from failures in other Availability zones and to provide inexpensive, low-latency network connectivity to other zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

If users complaint that their server instances are inaccessible, you may want to know whether it is because of the non-availability of the availability zone within which the instances have been launched. This test auto-discovers the availability zones configured within the monitored EC2 region, and reports the availability of each zone.

Purpose	Auto-discovers the availability zones configured within the monitored EC2 region, and reports the availability of each zone
Target of the test	Amazon EC2 Region
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the monitored region. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** In the EXCLUDE INSTANCE text box, provide a comma-separated list of instance names or instance name patterns that you do not wish to monitor. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*. 8. **REPORT INSTANCE DATACENTER** - By default, this test reports the availability of only those availability zones that contain one/more instances. Accordingly, this flag is set to **true** by default. If you want the test to report metrics for all availability zones, regardless of whether/not they host instances, set this flag to **false**.

<table>
<tr><td></td><td colspan="3">
9. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
10. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
11. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
12. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option against DETAILED DIAGNOSIS. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability.
• Both the bad and normal frequencies configured for the detailed diagnosis measures should not be 0.
</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each availability zone in the AWS EC2 Region being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Availability:
Indicates whether/not this availability zone is currently available.</td><td>Number</td><td>The value 0 indicates that the availability zone is Not Available and the value 100 indicates that it is Available.
If an availability zone fails, then all server instances operating within that zone will also be rendered unavailable. If you host all your Amazon EC2 instances in a single location that is affected by such a failure, your instances will be unavailable, thereby bringing your entire application to a halt.
On the other hand, if you have instances distributed across many Availability Zones and one of the instances fails, you can design your application so the instances in the remaining Availability Zones handle any requests.</td></tr>
</table>

3.1.2 EC2 - Regions Test

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Each Region is completely independent.

By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements.

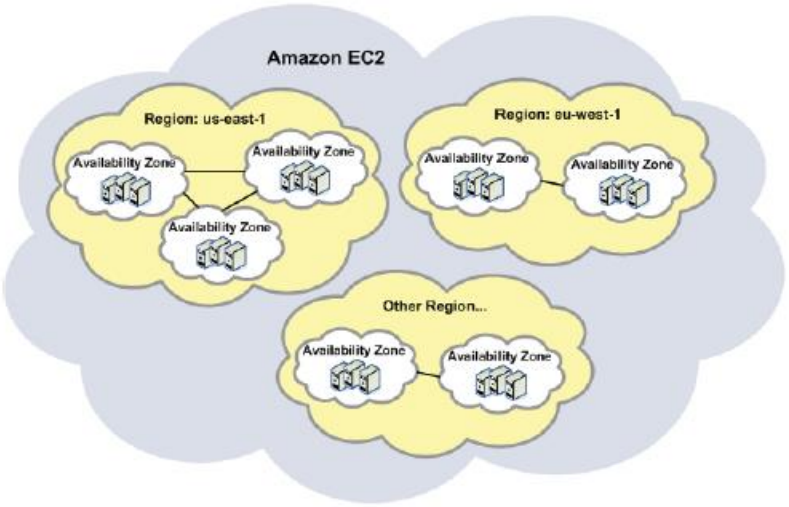


Figure 3.3: Regions and Availability zones

If a region is unavailable, then users to that region will not be able to access the server instances launched in that region. This may, in turn, adversely impact the user experience with the cloud. To avoid such an unpleasant outcome, it is best to periodically monitor the availability of each region, so that unavailable regions can be quickly and accurately identified, and the reasons for their non-availability remedied.

This test performs periodic availability checks on the monitored region, and reports the status of that region. In addition, the test also indicates the time taken for connecting to the region so that, connectivity issues can be isolated.

Purpose	Performs periodic availability checks on the monitored region, and reports the status of that region. In addition, the test also indicates the time taken for connecting to the region so that, connectivity issues can be isolated.
Target of the test	Amazon EC2 Cloud
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

	8. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default. 9. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default. 10. **PROXY DOMAIN** and **PROXY WORKSTATION** - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.		
Outputs of the test	One set of results for the AWS EC2 region being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Availability: Indicates whether/not the region is currently available.	Number	The value *0* indicates that the region is *Not Available* and the value *100* indicates that it is *Available*.
	Response time: Indicates the time taken to connect to the region.	Secs	A low value is typically desired for this measure. A high value or a consistent increase in this value could be indicative of connection bottlenecks.

3.1.3 AWS-EC2 Web Access Test

This test emulates a user accessing a web page on the cloud via HTTP(S), and reports whether that page is accessible or not. In the process, the test indicates the availability of the cloud over the web, and the time it took for the agent to access the cloud over the web. This way, issues in web-based access to the cloud come to light.

Purpose	Emulates a user accessing a web page (by default, the login page) on the cloud via HTTP(S), and reports whether that page is accessible or not. In the process, the test indicates the availability of the cloud over the web, and the time it took for the agent to access the cloud over the web.
Target	An AWS-EC2 cloud
Agent deploying this test	A remote agent
Configurable parameters for	1. **TEST PERIOD** – How often should the test be executed 2. **URL** – The web page being accessed. While multiple URLs (separated by commas) can be

this test	provided, each URL should be of the format **URL name:URL value**. **URL name** is a unique name assigned to the URL, and the **URL value** is the value of the URL. By default, the url parameter is set to *HomePage:http://aws.amazon.com/ec2/*, where *HomePage* is the **URL name**, and *http://aws.amazon.com/ec2* is the **URL** value. You can modify this default setting to configure any URL of your choice - eg., the URL of the login page to your cloud-based infrastructure. 3. **HOST** - The host for which the test is to be configured. 4. **PORT** - The port to which the specified **HOST** listens 5. **COOKIEFILE** – Whether any cookies being returned by the web server need to be saved locally and returned with subsequent requests 6. **PROXYHOST** – The host on which a web proxy server is running (in case a proxy server is to be used) 7. **PROXYPORT** – The port number on which the web proxy server is listening 8. **PROXYUSERNAME** – The user name of the proxy server 9. **PROXYPASSWORD** – The password of the proxy server 10. **CONFIRM PASSWORD** – Confirm the password by retyping it here. 11. **CONTENT** – Is a set of instruction:value pairs that are used to validate the content being returned by the test. If the CONTENT value is *none:none*, no validation is performed. The number of pairs specified in this text box, must be equal to the number of URLs being monitored. The instruction should be one of *Inc* or *Exc*. *Inc* tells the test that for the content returned by the test to be valid, the content must include the specified value (a simple string search is done in this case). An instruction of *Exc* instructs the test that the test's output is valid if it does not contain the specified value. In both cases, the content specification can include wild card patterns. For example, an Inc instruction can be *Inc:*Home page**. An Inc and an Exc instruction can be provided in quick succession in the following format: *Inc:*Home Page*,Exc:*home*. 12. **CREDENTIALS** – The HttpTest supports HTTP authentication. The **CREDENTIALS** parameter is to be set if a specific user name / password has to be specified to login to a page. Against this parameter, the **URLname** of every configured url will be displayed; corresponding to each listed **URLname**, a **Username** text box and a **Password** text box will be made available. These parameters will take either of the following values: a. a valid **Username** and **Password** for every configured **URLname** b. *none* in both the **Username** and **Password** text boxes of all configured **URLnames** (the default setting), if no user authorization is required Where NTLM (Integrated Windows) authentication is supported, valid **CREDENTIALS** are mandatory. In other words, a *none* specification will not be supported in such cases. Therefore, in this case, against each configured **URLname**, you will have to provide a valid **Username** in the format: *domainname|username,* followed by a valid **Password**. Please be sure to check if your web site requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many sites use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the username and password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the this test. 13. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments,

<table>
<tr><td></td><td colspan="3">you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
14. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
15. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
16. TIMEOUT - Here, specify the maximum duration (in seconds) for which the test will wait for a response from the server. The default TIMEOUT period is 30 seconds.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of outputs for every URL being monitored</td></tr>
<tr><td rowspan="4">Measurements of the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Availability:
This measurement indicates whether the test was able to access the configured URL or not.</td><td>Percent</td><td>Availability failures could be caused by several factors such as the web server process(es) (hosting the configured web page) being down, the web server being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web server is overloaded. Availability is determined based on the response code returned by the test. A response code between 200 to 300 indicates that the configured web page is available.</td></tr>
<tr><td>Total response time:
This measurement indicates the time taken by the test to access this URL.</td><td>Secs</td><td>Response time being high denotes a problem. Poor response times may be due to an overload. If the URL accessed involves the generation of dynamic content, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.</td></tr>
<tr><td>Tcp connection availability:
This measure indicates whether the test managed to establish a TCP connection to this URL.</td><td>Percent</td><td>Failure to establish a TCP connection may imply that either the web server process hosting the web page is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the web page may start functioning properly again.</td></tr>
</table>

	Tcp connect time: This measure quantifies the time for establishing a TCP connection to the configured URL.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds).
	Server response time: This measure indicates the time period between when the connection was established and when the test sent back a HTTP response header to the client.	Secs	While the total response time may depend on several factors, the server response time is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
	Response code: The response code returned by the test for the simulated request	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
	Content length: The size of the content returned by the test	Kbytes	Typically the content length returned by the test for a specific URL should be the same across time. Any change in this metric may indicate the need for further investigation.
	Content validity: This measure validates whether the test was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, "Invalid Login" is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string "Invalid Login," in the above scenario content validity would have a value 0.

3.2 The AWS EC2 Region Instance Status Layer

To determine issues in accessibility server instances launched in a region, and to detect the current state of each instance, use the tests mapped to this layer.

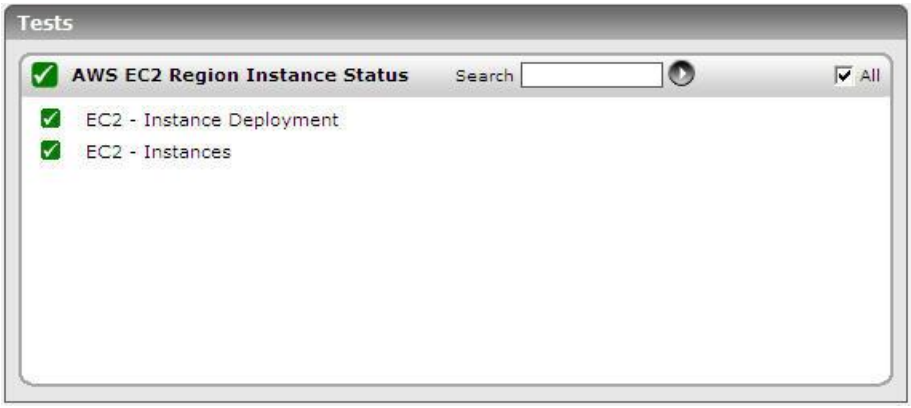


Figure 3.4: The tests mapped to the AWS EC2 Region Instance Status layer

3.2.1 EC2 - Instance Deployment Test

This test powers a specified VM on and off at configured intervals. In the process, the test verifies the success/failure of the corresponding operation (i.e., power on / off), and also reports the time taken by that instance to power on and off. Failed attempts to power-on and significant delays in powering on are thus brought to light.

This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, pick the desired **Component type**, set *Performance* as the **Test type**, choose the test from the **DISABLED TESTS** list, and click on the **>>** button to move the test to the **ENABLED TESTS** list. Finally, click the **Update** button.

Purpose	Powers a specified VM on and off at configured intervals, and in the process verifies the success/failure of the corresponding operation (i.e., powering on / off), and also reports the time taken by that instance to power on and off. Failed attempts to power-on and significant delays in powering on are thus brought to light
Target of the test	Amazon EC2 Region
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed; by default, this is set to *24 hrs*. 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **INSTANCE ID** - Specify the ID of the instance that is to be powered on/off by this test. By default, this test is not designed to report any metrics for any instance; this is why, the instance id is set to *none* by default. **Note that this test will not run until a valid** instance id **and** instance type **are provided**. 7. **INSTANCE TYPE** - Specify the type of the configured instance. Since this test is not designed to report any metrics for any instance by default, this parameter is set to *none* by default. **Note that this test will report metrics only if a valid** instance id **and** instance type **are provided**. 8. **FREQUENCY** - Indicate how frequently (in seconds) the test needs to power an instance on and off. By default, the test period and frequency of this test will be the same; both will be set to *24 hrs (*i.e., *86400 seconds*) by default. However, some administrators may not want their critical instances to be powered on and off at the same frequency at which the test runs. In such a case, you can define a separate power on/off frequency for the test using the frequency parameter. 9. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default. 10. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default.

<table>
<tr><td></td><td colspan="3">11. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the AWS EC2 region being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Was instance powered on sucessful?:
Indicates whether/not the configured instance was powered on successfully.</td><td></td><td>If the instance was powered-on successfully, then the value of this measure will be Yes. If not, then the value of this measure will be No.
The numeric values that correspond to the measure values mentioned above are as follows:<table><tr><th>Measure value</th><th>Numeric value</th></tr><tr><td>Yes</td><td>100</td></tr><tr><td>No</td><td>0</td></tr></table>Note:
By default, this measure reports one of the Measure values listed in the table above. The graph of this measure however, represents the success/failure of a power-on operation using the numeric equivalents - ‘0’ and '100' - only.</td></tr>
<tr><td></td><td>Was instance powered off sucessful?:
Indicates whether/not the configured instance was powered off successfully.</td><td></td><td>If the instance was powered-off successfully, then the value of this measure will be Yes. If not, then the value of this measure will be No.
The numeric values that correspond to the measure values mentioned above are as follows:<table><tr><th>Measure value</th><th>Numeric value</th></tr><tr><td>Yes</td><td>100</td></tr><tr><td>No</td><td>0</td></tr></table>Note:
By default, this measure reports one of the Measure values listed in the table above. The graph of this measure however, represents the success/failure of a power-off operation using the numeric equivalents - ‘0’ and '100' - only.</td></tr>
</table>

	Time taken for instance to be powered on: Indicates the time taken for the configured instance to be powered on.	Secs	Ideally, the value of these measures should be low. A sudden increase in the value could indicate a problem situation that requires further investigation. **Note that these measures will report values only if the corresponding operation succeeds. For instance, the 'Time taken for instance to be powered on' measure will report valid metrics only if the 'Was instance powered on successful?' measure reports the value 'Yes'.**
	Time taken for instance to be powered off: Indicates the time taken for the configured instance to be powered off.	Secs	

3.2.2 EC2 - Instance Connectivity Test

Sometimes, an instance could be in a powered-on state, but the failure of the operating system or any fatal error in internal operations of the instance could have rendered the instance inaccessible to users. In order to enable you to promptly detect such 'hidden' anomalies, this test periodically runs a connectivity check on each instance available for the configured AWS user account in the monitored region, and reports whether the instances are accessible over the network or not.

Purpose	Runs a connectivity check on each instance available for the configured AWS user account in the monitored region, and reports whether the instances are accessible over the network or not
Target of the test	An AWS-EC2 Region
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**.. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

	8. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default. 9. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default. 10. **PROXY DOMAIN** and **PROXY WORKSTATION** - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.
Outputs of the test	One set of results for each instance available for the configured AWS user account in the monitored region

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Avg network delay: Indicates the average delay between transmission of packets to this instance and receipt of the response to the packet at the source.	Secs	An increase in network latency could result from misconfiguration of the router(s) along the path, network congestion, retransmissions at the network, etc.
	Min network delay: The minimum time between transmission of a packet and receipt of the response back.	Secs	A significant increase in the minimum round-trip time is often a sure sign of network congestion.
	Packet loss: Indicates the percentage of packets lost during transmission from source to target and back.	Percent	Packet loss is often caused by network buffer overflows at a network router or by packet corruptions over the network. The detailed diagnosis for this measure provides a listing of routers that are on the path from the external agent to target server, and the delays on each hop. This information can be used to diagnose the hop(s) that could be causing excessive packet loss/delays.
	Network availability of instance: Indicates whether the network connection to this instance is available or not.	Percent	A value of 100 indicates that the instance is accessible over the network. The value 0 indicates that the instance is inaccessible. Typically, the value 100 corresponds to a *Packet loss* of 0.

3.2.3 EC2 - Instances Test

An Amazon Machine Image (AMI) contains all information necessary to boot instances of your software. For example, an AMI might contain all the software to act as a web server (e.g., Linux, Apache, and your web site) or it might contain all the software to act as a Hadoop node (e.g., Linux, Hadoop, and a custom application). After an AMI is launched, the resulting running system is called an **instance**. All instances based on the same AMI start out identical and any information on them is lost when the instances are terminated or fail.

Users with valid AWS user accounts can sign into an EC2 region to view and use available instances, or purchase and launch new ones. With the help of this test, you can determine the total number of instances that are currently available for the configured AWS user account in the monitored region, the number of instances that were newly purchased/terminated, and the count of powered-off instances.

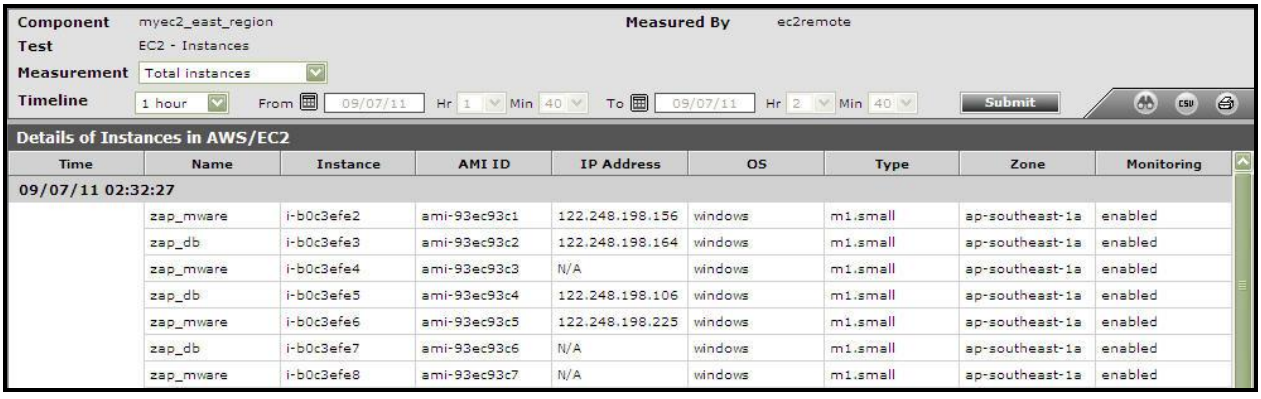
Purpose	Helps determine the total number of instances that are currently available for the configured AWS user account in the monitored region, the number of instances that were newly purchased/terminated, and the count of powered-off instances

Target of the test	Amazon EC2 Region
Agent deploying the test	A remote agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
11. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option against DETAILED DIAGNOSIS. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability.
• Both the bad and normal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the AWS EC2 Region being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Total instances:
Indicates the total number of instances currently available for the configured AWS user account in the monitored region.</td><td>Number</td><td>The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances available for use for the configured AWS account, regardless of the current state of the instances.</td></tr>
</table>

	Instances powered on: Indicates the total number of instances that are currently powered-on in the monitored region.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-on instances available for use for the configured AWS account.
	Instances powered off: Indicates the total number of instances that are currently powered-off in the monitored region.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the powered-off instances available for the configured AWS account.
	Added instances: Indicates the total number of instances that were newly purchased by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly purchased and launched by the configured AWS user account.
	Removed instances: Indicates the total number of instances that were newly terminated by the configured AWS user account during the last measurement period.	Number	The detailed diagnosis capability of this measure, if enabled, shows the details of all the instances that were newly terminated/removed by the configured AWS user account.

The detailed diagnosis capability of the *Total instances* measure, if enabled, shows the details of all the instances available for use for the configured AWS account in the monitored region, regardless of the current state of the instances.



Time	Name	Instance	AMI ID	IP Address	OS	Type	Zone	Monitoring
09/07/11 02:32:27								
	zap_mware	i-b0c3efe2	ami-93ec93c1	122.248.198.156	windows	m1.small	ap-southeast-1a	enabled
	zap_db	i-b0c3efe3	ami-93ec93c2	122.248.198.164	windows	m1.small	ap-southeast-1a	enabled
	zap_mware	i-b0c3efe4	ami-93ec93c3	N/A	windows	m1.small	ap-southeast-1a	enabled
	zap_db	i-b0c3efe5	ami-93ec93c4	122.248.198.106	windows	m1.small	ap-southeast-1a	enabled
	zap_mware	i-b0c3efe6	ami-93ec93c5	122.248.198.225	windows	m1.small	ap-southeast-1a	enabled
	zap_db	i-b0c3efe7	ami-93ec93c6	N/A	windows	m1.small	ap-southeast-1a	enabled
	zap_mware	i-b0c3efe8	ami-93ec93c7	N/A	windows	m1.small	ap-southeast-1a	enabled

Figure 3.5: The detailed diagnosis of the Total instances measure

The detailed diagnosis capability of the *Instances powered on* measure, if enabled, shows the details of all the powered-on instances available for use for the configured AWS account in the monitored region.

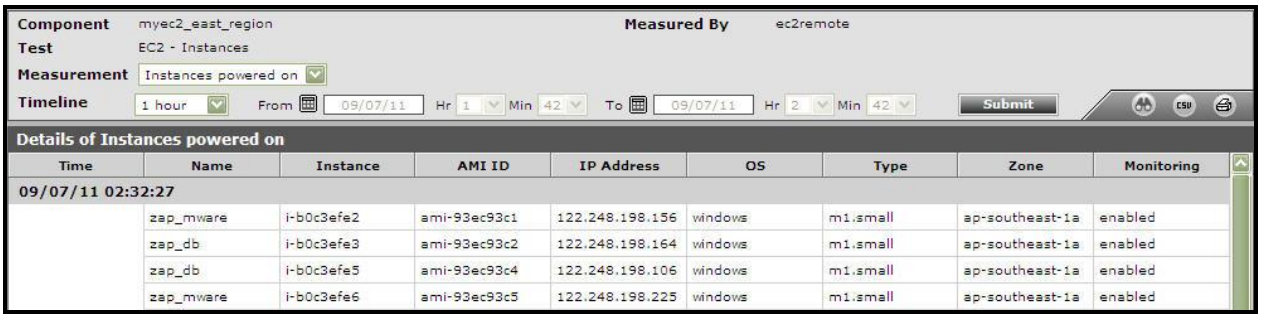


Figure 3.6: The detailed diagnosis of the Instances powered on measure

The detailed diagnosis capability of the *Instances powered off* measure, if enabled, shows the details of all the powered-off instances available for the configured AWS account.

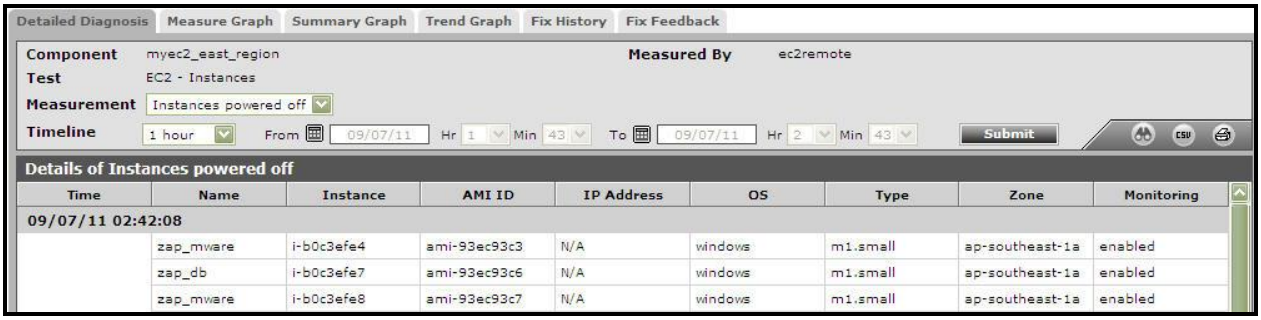


Figure 3.7: The detailed diagnosis of the Instances powered off measure

3.3 The AWS EC2 Region Instance Details Layer

The tests mapped to this layer auto-discover the server instances that are available (for the configured AWS user account) in a region, and reports the uptime and the resource usage of the individual instances. Resource-hungry instances and those that were recently rebooted can thus be isolated.

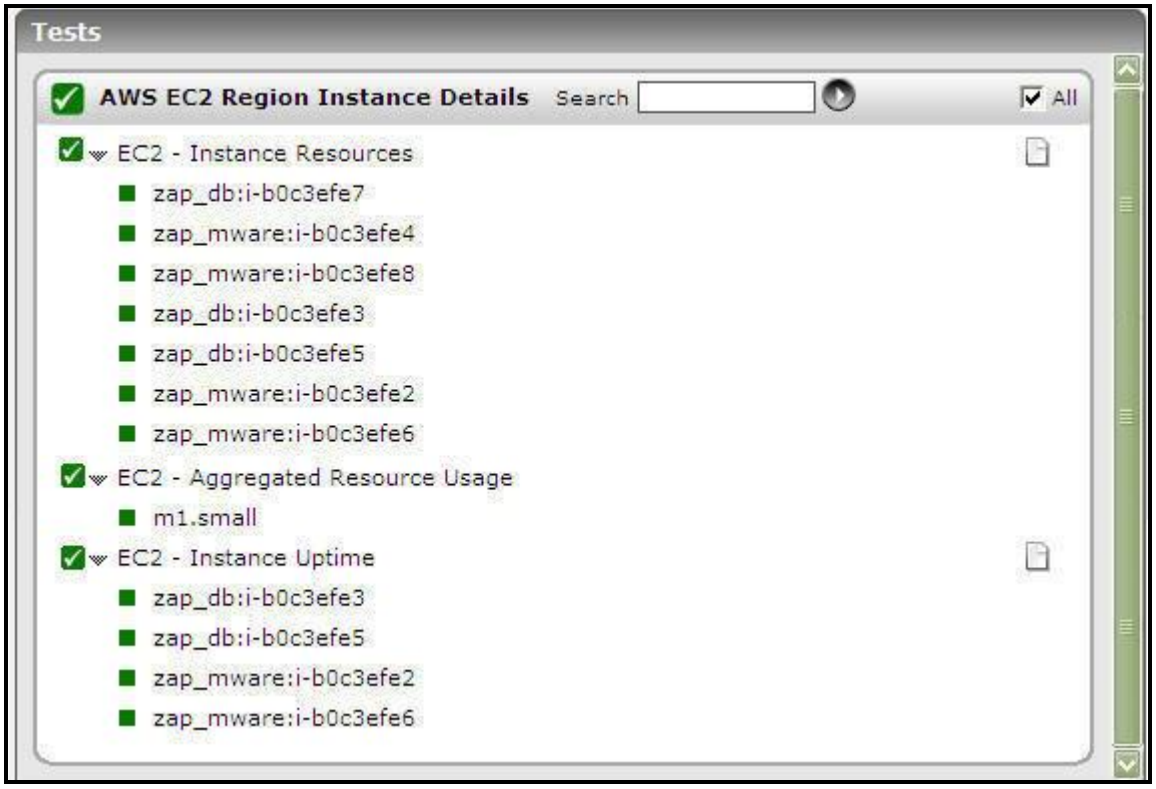


Figure 3.8: The tests mapped to the AWS EC2 Region Instance Details layer

3.3.1 EC2 - Aggregated Resource Usage Test

When users launch an instance using the AWS management console, they need to specify the instance type. An instance type is a specification that defines the memory, CPU, storage capacity, and hourly cost for an instance. Some instance types are designed for standard applications, whereas others are designed for CPU-intensive applications, or memory-intensive applications, etc. The different instance types offered by the AWS EC2 cloud are as follows:

Type	CPU	Memory	Local Storage	Platform	I/O	Name
Small	1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)	1.7 GB	160 GB instance storage (150 GB plus 10 GB root partition)	32-bit	Moderate	m1.small
Large	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	7.5 GB	850 GB instance storage (2 x 420 GB plus 10 GB root partition)	64-bit	High	m1.large
Extra Large	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)	15 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	m1.xlarge
Micro	Up to 2 EC2 Compute Units (for short periodic bursts)	613 MB	None (use Amazon EBS volumes for storage)	32-bit or 64-bit	Low	t1.micro
High-CPU Medium	5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)	1.7 GB	350 GB instance storage (340 GB plus 10 GB root partition)	32-bit	Moderate	c1.medium
High-CPU Extra Large	20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)	7 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)	64-bit	High	c1.xlarge
High-Memory Extra Large	6.5 EC2 Compute Units (2 virtual cores with 3.25 EC2 Compute Units each)	17.1 GB	420 GB instance storage (1 x 420 GB)	64-bit	Moderate	m2.xlarge
High-Memory Double Extra Large	13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)	34.2 GB	850 GB instance storage (1 x 840 GB plus 10 GB root partition)	64-bit	High	m2.2xlarge
High-Memory Quadruple Extra Large	26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)	68.4 GB	1690 GB instance storage (2 x 840 GB plus 10 GB root partition)	64-bit	High	m2.4xlarge
Cluster Compute	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)	23 GB	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cc1.4xlarge
Cluster GPU	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture), plus 2 NVIDIA Tesla M2050 "Fermi" GPUs	22 GB (see note after this table)	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)	64-bit	Very high (10 Gbps Ethernet)	cg1.4xlarge

By closely monitoring the CPU usage and the network and disk I/O of each instance type, and comparing these metrics across instance types, you can quickly isolate resource-intensive types. Once again, the test will report metrics for only those types of instances that were launched by the AWS user account configured for the test in the monitored region.

Purpose	Closely monitors the CPU usage and the network and disk I/O of each instance type, and enables usage comparison across instance types, so as to quickly isolate resource-intensive types
Target of the test	Amazon EC2 Region
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

	8. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default. 9. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default. 10. **PROXY DOMAIN** and **PROXY WORKSTATION** - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.		
Outputs of the test	One set of results for each type of instance launched by the configured AWS user account in the monitored region		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	CPU utilization: Indicates the percentage of allocated CPU consumed by all instances of this type.	Percent	A high value for this measure indicates that one/more instances of a type are utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on the instances. Compare the value of this measure across types to identify the types of instances that are CPU-intensive.
	Incoming network traffic: Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to all the instances of this instance type.	KB/Sec	Compare the values of these measures across instance types to quickly identify the types of instances that are utilizing the network bandwidth excessively.

	Outgoing network traffic: Indicates the volume of outgoing network traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to all the instances of a particular instance type.	KB/Sec	
	Disk reads: Indicates the rate at which data is read from the disks of all instances of this type.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
	Disk writes: Indicates the rate at which data is written to the disks of all instances of this type.	KB/Sec	
	Disk read operations: Indicates the rate at which disk read operations were performed on the disks of all instances of this type.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance type. By comparing the values of these measures across types, you can accurately determine the type of instances that is performing I/O-intensive operations.
	Disk write operations: Indicates the rate at which disk write operations were performed performed on the disks of all instances of this type.	Operations/Sec	

3.3.2 EC2 - Instance Resources Test

Tracking the CPU usage, disk and network I/O of every instance launched by a configured AWS user account in a region will provide administrators with valuable insights into how well the instances are utilizing the allocated resources. The **EC2 - Instance Resources** test does just that. This test auto-discovers the instances available for the configured AWS user account in a region, and reports the resource usage of each instance so that, administrators can quickly compare the usage metrics across instances and pinpoint which instance is resource-hungry.

Purpose	Auto-discovers the instances available in the monitored region for the configured AWS user account, and reports the resource usage of each instance so that, administrators can quickly compare the usage metrics across instances and pinpoint which instance is resource-hungry
Target of the test	Amazon EC2 Region
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

	8. **PROXYHOST** and **PROXY PORT**– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the **PROXY HOST** and **PROXY PORT** parameters. By default, these parameters are set to *none* , indicating that the eG agent is not configured to communicate via a proxy, by default. 9. **PROXY USERNAME** and **PROXY PASSWORD** - If the proxy server requires authentication, then, specify a valid proxy user name and password in the **PROXY USER NAME** and **PROXY PASSWORD** parameters, respectively. By default, these parameters are set to *none* , indicating that the proxy sever does not require authentication by default. 10. **PROXY DOMAIN** and **PROXY WORKSTATION** - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the **PROXY DOMAIN** and **PROXY WORKSTATION** parameters. If the environment does not support a Windows NTLM proxy, set these parameters to *none*.		
Outputs of the test	One set of results for each instance launched by the configured AWS user account in the monitored region		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	CPU utilization: Indicates the percentage of allocated CPU consumed by this instance.	Percent	A high value for this measure indicates that an instance is utilizing CPU excessively - this could be because of one/more resource-intensive processes executing on that instance. Compare the value of this measure across instances to identify the CPU-intensive instances.
	Incoming network traffic: Indicates the rate of incoming network traffic i.e., the rate at which the bytes are received by all the network interfaces connected to this instance.	KB/Sec	Compare the values of these measures across instances to quickly identify the instance that is utilizing the network bandwidth excessively.
	Outgoing network traffic: Indicates the volume of outgoing network traffic i.e., the rate at which the bytes are transferred from all the network interfaces connected to this instance.	KB/Sec	

	Disk reads: Indicates the rate at which data is read from the disks of this instance.	KB/Sec	These measures are good indicators of the level of disk I/O activity on an instance. By comparing the values of these measures across instances, you can accurately determine which instance is performing I/O-intensive operations.
	Disk writes: Indicates the rate at which data is written to the disks of this instance.	KB/Sec	
	Disk read operations: Indicates the rate at which disk read operations are performed on this instance.	Operations/Sec	These measures are good indicators of the level of disk I/O activity on an instance. By comparing the values of these measures across instances, you can accurately determine which instance is performing I/O-intensive operations.
	Disk write operations: Indicates the rate at which disk write operations were performed on this instance.	Operations/Sec	

3.3.3 EC2 - Instance Uptime Test

In cloud-based environments, it is essential to monitor the uptime of server instances launched on the cloud. By tracking the uptime of each of the instances, administrators can determine what percentage of time an instance has been up. Comparing this value with service level targets, administrators can determine the most trouble-prone areas of the infrastructure hosted on the cloud.

In some environments, administrators may schedule periodic reboots of their instances. By knowing that a specific instance has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on an instance.

This test monitors the uptime of each instance available to the configured AWS user account.

Purpose	Monitors the uptime of each instance available to the configured AWS user account in the monitored region
Target of the test	Amazon EC2 Region
Agent deploying the test	A remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The host for which the test is being configured 3. **AWS ACCESS KEY** - To monitor an AWS EC2, the eG agent has to be configured with the "access key" of a user with a valid AWS account. To obtain the access key, follow the steps given below: • Sign up for a new AWS account from the http://aws.amazon.com/ec2/ page. • Provide the details of the user for whom you wish to create the AWS account. • Based on the AWS EC2 Regions, you will be requested to choose the pricing for an instance to be deployed in the AWS EC2. • Once the payment is made, the user will be automatically signed in to the AWS account. • From the newly created AWS account, you can request for an "access key". You will be provided with an "access key" and a corresponding "secret key". Provide the access key in the **AWS ACCESS KEY** text box; this will enable the eG agent to communicate with the AWS API and collect the required metrics. 4. **CONFIRM PASSWORD** - Confirm the password by retyping it here. 5. **AWS SECRET KEY** - Provide the secret key corresponding to the access key that you had obtained through your AWS account. 6. **CLOUDWATCH ENABLED** - **This flag applies to the EC2 - Instance Resources and EC2 - Aggregate Resource Usage tests only**. These tests report critical metrics pertaining to the resource usage of the server instances launched in the cloud. If you want these tests to report resource usage metrics very frequently - say, once every minute or lesser - you will have to configure the tests to use the **AWS CloudWatch** service. This is a **paid** web service that enables you to monitor, manage, and publish various metrics, as well as configure alarm actions based on data from metrics. To enable the above-mentioned tests to use this service, set the cloudwatch enabled flag to *true*. On the other hand, to report resource usage metrics less frequently - say, once in 5 minutes or more - these tests do not require the **AWS CloudWatch** service; in this case therefore, set the cloudwatch enabled flag to *false*. **Note that for enabling CloudWatch, you will have to pay CloudWatch fees. For the fee details, refer to the AWS EC2 web site.** 7. **EXCLUDE INSTANCE** - **This parameter applies only to EC2 - Instance Connectivity, EC2 - Instance Resources , EC2 - Instance Uptime, and EC2 - Instance Deployment tests.** Since these tests report metrics for each server instance launched on the cloud, you can optionally configure these tests to exclude one/more server instances from monitoring. For this, specify a comma-separated list of instance names or instance name patterns that need not be monitored in the EXCLUDE INSTANCE text box. For example: *i-b0c3e*,*7dbe56d*. By default, this parameter is set to *none*.

<table>
<tr><td></td><td colspan="3">
8. PROXYHOST and PROXY PORT– In some environments, all communication with the AWS EC2 cloud and its regions could be routed through a proxy server. In such environments, you should make sure that the eG agent connects to the cloud via the proxy server and collects metrics. To enable metrics collection via a proxy, specify the IP address of the proxy server and the port at which the server listens against the PROXY HOST and PROXY PORT parameters. By default, these parameters are set to none , indicating that the eG agent is not configured to communicate via a proxy, by default.
9. PROXY USERNAME and PROXY PASSWORD - If the proxy server requires authentication, then, specify a valid proxy user name and password in the PROXY USER NAME and PROXY PASSWORD parameters, respectively. By default, these parameters are set to none , indicating that the proxy sever does not require authentication by default.
10. PROXY DOMAIN and PROXY WORKSTATION - If a Windows NTLM proxy is to be configured for use, then additionally, you will have to configure the Windows domain name and the Windows workstation name required for the same against the PROXY DOMAIN and PROXY WORKSTATION parameters. If the environment does not support a Windows NTLM proxy, set these parameters to none.
11. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability
• Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each instance launched by the configured AWS user account in the monitored region</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td></td><td>Has the instance been rebooted?:
Indicates whether this instance has been rebooted during the last measurement period or not.</td><td>Boolean</td><td>If this measure shows 1, it means that the instance was rebooted during the last measurement period. By checking the time periods when this metric changes from 0 to 1, an administrator can determine the times when this instance was rebooted.</td></tr>
</table>

	Uptime of the instance during the last measure period: Indicates the time period that the instance has been up since the last time this test ran.	Secs	If the instance has not been rebooted during the last measurement period and the agent has been running continuously, this value will be equal to the measurement period. If the instance was rebooted during the last measurement period, this value will be less than the measurement period of the test. For example, if the measurement period is 300 secs, and if the instance was rebooted 120 secs back, this metric will report a value of 120 seconds. The accuracy of this metric is dependent on the measurement period - the smaller the measurement period, greater the accuracy.
	Total uptime of the instance: Indicates the total time that this intance has been up since its last reboot.	Mins	Administrators may wish to be alerted if an instance has been running without a reboot for a very long period. Setting a threshold for this metric allows administrators to determine such conditions.

The detailed diagnosis of the *Has VM been rebooted?* measure reveals when the instance was last shutdown, when it was rebooted, how long the shutdown lasted, and whether the instance was shutdown as part of a routine maintenance exercise.

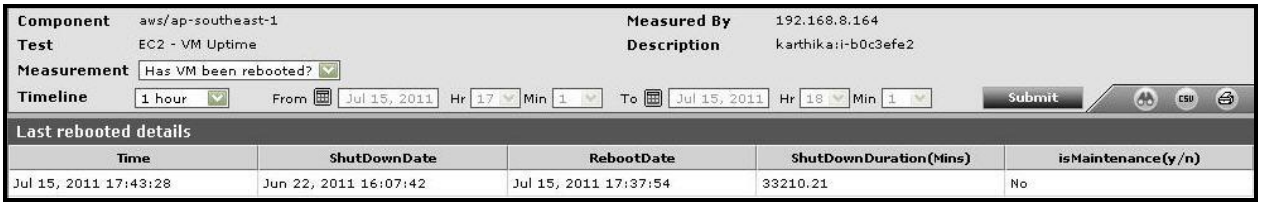


Figure 3.9: The detailed diagnosis of the Has VM been rebooted? measure

4

Conclusion

This document has clearly explained how eG Enterprise monitors the AWS EC2 cloud and region. For more information on eG Enterprise, please visit our web site at www.eginnovations.com or write to us at sales@eginnovations.com.