

# Carbonite Server Backup API – Monitoring v1.5

Installation and Usage Guide

© 2019 Carbonite, Inc. All rights reserved.

Carbonite makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Carbonite reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Carbonite to notify any person of such revision of changes. All companies, names and data used in examples herein are fictitious unless otherwise noted.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval System or translated into any language including computer language, in any form or by any means electronic, mechanic, magnetic, optical, chemical or otherwise without prior written permission of:

Carbonite, Inc. Two Avenue de Lafayette Boston, MA 02111 www.carbonite.com

Carbonite and the Carbonite logo are registered trademarks, and Carbonite Server Backup, Carbonite Server Backup SaaS, and Carbonite Server Backup DeltaPro, are trademarks, of Carbonite, Inc. All other products or company names mentioned in this document are trademarks or registered trademarks of their respective owners.

Acknowledgements: Two encryption methods, DES and TripleDES, include cryptographic software written by Eric Young. The Windows versions of these algorithms also include software written by Tim Hudson. Bruce Schneier designed Blowfish encryption.

"Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are Copyright 2001-2006 Robert A. van Engelen, Genivia Inc. All Rights Reserved. THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

The Carbonite Server Backup Agent, Carbonite Server Backup CentralControl, and Carbonite Server Backup Director applications have the encryption option of AES (Advanced Encryption Standard). Advanced Encryption Standard algorithm (named Rijndael, pronounced "Rain Doll") was developed by cryptographers Dr. Joan Daemen and Dr. Vincent Rijmen. This algorithm was chosen by the National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce to be the new Federal Information Processing Standard (FIPS).

The Carbonite Server Backup Agents and Carbonite Server Backup Director applications also have the added security feature of an over the wire encryption method.

Version	Date	Description
1	November 2019	Initial installation and usage guide provided for API – Monitoring 1.5x.

### **Document History**

# Contents

1	Intro	duction to Carbonite Server Backup API – Monitoring	4
	1.1	Monitoring overview	4
	1.2	Data deletion overview	5
	1.3	API – Monitoring components	6
2	Prepa	are for an API – Monitoring installation	8
	2.1	Determine where to install API – Monitoring	8
	2.2	Check system requirements	8
	2.3	Determine how to secure API – Monitoring	10
	2.4	Set up a public domain name or IP address	10
	2.5	Plan ongoing SQL Server database maintenance	10
3	Insta	ll or upgrade API – Monitoring	11
	3.1	Install API – Monitoring	11
	3.2	Upgrade API – Monitoring	15
	3.3	Troubleshoot an API – Monitoring installation	16
	3.4	Verify an API – Monitoring installation	17
	3.5	Uninstall API – Monitoring	17
	3.6	Reinstall API – Monitoring	19
4	Confi	gure and maintain API – Monitoring	20
	4.1	Configure the refresh interval for job and safeset data	20
	4.2	Replace the API – Monitoring certificate	20
5	Regis	ter Director Reporting services to the API	22
	5.1	Obtain a token for registering Director Reporting services to the API	22
6	View	documentation for API – Monitoring calls	23
7	Moni	tor protected data	26
	7.1	Create a Keycloak admin user	26
	7.2	Create API – Monitoring clients	27
	7.3	Obtain the ID and secret for a client	30
	7.4	Test API – Monitoring calls	32
8	Set u	p data deletion	37

# **1** Introduction to Carbonite Server Backup API – Monitoring

Carbonite Server Backup API – Monitoring is a public OData Web API that enables service providers to:

- Remotely monitor their protected data. As described in <u>Monitoring overview</u>, client applications can make read-only calls to API – Monitoring to obtain information about the following entities in Carbonite Server Backup Portal and in Carbonite Server Backup Director vaults:
  - Agents
  - Companies (sites)
  - Company users
  - Jobs, job schedules and retentions
  - Safesets
  - Data deletion requests
  - Vaults

Detailed information about available calls is available in the Swagger UI that is installed with API – Monitoring. See <u>View documentation for API – Monitoring calls</u>.

Delete data from Director vaults in response to requests from Portal. See <u>Data deletion overview</u>.

To install and set up API – Monitoring, see <u>Prepare for an API – Monitoring installation</u>, <u>Install or upgrade API –</u> <u>Monitoring and Configure and maintain API – Monitoring</u>.

To monitor protected data using API – Monitoring, see the required steps in Monitor protected data.

To delete data from Director vaults in response to request from Portal, see the required steps in <u>Set up data</u> <u>deletion</u>.

*Note:* API – Monitoring is sometimes referred to as "the API" in this guide.

Because API – Monitoring uses structured logging, you can use a tool such as SEQ or Splunk to analyze API log files. By default, log files are saved in C:\Logs\Carbonite Server Backup API\Monitoring.

### 1.1 Monitoring overview

To monitor agents, jobs, safesets and other entities in Carbonite Server Backup, client applications can make read-only calls to API – Monitoring. In response, the API provides requested data from Portal databases and from a Monitoring database that is installed with the API.

As shown in the <u>Monitoring overview diagram</u>, a Reporting service on each registered vault sends data to API – Monitoring. API components then process and store the data in the Monitoring database. A Reporting service should be installed on each vault where your data is stored.

To authenticate and authorize API calls, the API uses an authorization server that is installed with API – Monitoring. Client applications obtain access tokens from the authorization server and include them in calls to the API. If a token is valid, the API returns the requested data from the Portal and Monitoring databases.

Tokens from the authorization server are also used to authorize communications between Reporting services and the API.

To view available API calls, see <u>View documentation for API – Monitoring calls</u>.

To start monitoring protected data using API – Monitoring calls, see required steps in Monitor protected data.

### Monitoring overview diagram



### **1.2 Data deletion overview**

As shown in the <u>Data deletion overview diagram</u>, when Portal and Director vaults are registered to the same API – Monitoring instance, Director can delete job and computer data in response to requests from Portal.

When deleting a job or computer in Portal, a Portal Admin user can request that backup data for the job or computer be deleted from all vaults. The data deletion is scheduled for 72 hours after the request is made.

If an Admin user does not cancel a scheduled data deletion during the 72-hour waiting period, the deletion request is sent to Director vaults that are registered to API – Monitoring. In response to the request, Director deletes the data from any standalone, Base or Active vault where the data is stored. Replication processes then delete the data from any associated Satellite or Passive vault.

*Note:* Although it is not required for data deletion, the Reporting service must be installed with Satellite and Passive vaults and registered to API – Monitoring to provide data through API – Monitoring calls.

If a deletion request fails, an email notification is sent to a vault administrator whose email address is specified in Portal. The vault administrator can then manually delete the data.

For required steps for setting up data deletion, see Set up data deletion.

### Data deletion overview diagram



# 1.3 API – Monitoring components

When you install API – Monitoring, the following system components and their prerequisites are installed:

Component	Description
API – Monitoring	Validates access tokens in API calls. If access tokens are valid and have correct authorization, returns requested data from the Portal and Monitoring databases.
Monitoring database	Stores data received from Director vaults. Installed on the Portal database SQL Server instance.

Component	Description
Keycloak	Third-party authorization server that is used to authenticate and authorize clients
	and API system components.
Keycloak database	Database that includes client authorization information for Keycloak.
Message Bus	Validates tokens from vault reporting services and authorizes communications
Authentication with the message bus.	
RabbitMQ Server	Third-party message bus enabling asynchronous communication among
	Carbonite Server Backup solution components.
Collector	Processes data coming from Carbonite Server Backup solution components and
	makes them available for querying via the API – Monitoring.
Registration Service	Securely registers Carbonite Server Backup solution components with the API –
	Monitoring system. Enables secure communication between components and
	common authorization mechanism.

# **2** Prepare for an API – Monitoring installation

Before installing API – Monitoring, you must do the following:

- <u>Determine where to install API Monitoring</u>
- Ensure that the server where you want to install the API meets all system requirements and that all prerequisites are installed. See <u>Check system requirements</u>.
- Determine how to secure API Monitoring
- <u>Set up a public domain name or IP address</u>
- Plan ongoing SQL Server database maintenance

For compatible Portal and Director versions, see the API – Monitoring release notes.

## **2.1** Determine where to install API – Monitoring

You can install API – Monitoring on:

- A server where no Portal components are installed. Carbonite recommends this installation location.
- A server where Portal components are installed. Although these installation locations are not recommended:
  - If all Portal components are installed on one server, you can install the API on the Portal server.
  - If Portal is installed as a distributed system, you can install the API on the same server as front-end Portal components. In this case, Portal and API Monitoring must have different IP addresses or host names (e.g., portal.domain.com and api.domain.com). Do not use a subdirectory of the Portal URL for the API (e.g., portal.domain.com/api).

When installing API – Monitoring on a separate server from the Portal SQL Server instance, you can choose SQL Server authentication or Windows authentication to connect to SQL Server. Carbonite recommends using SQL Server authentication in this case. To use Windows authentication, the servers must be in the same domain and you must run the API installer using a domain administrative account that has sysadmin rights to the SQL Server instance.

# 2.2 Check system requirements

The server where you install API – Monitoring must meet the requirements described in the following sections:

- Hardware requirements
- <u>Software requirements</u>
- <u>Required ports</u>

### 2.2.1 Hardware requirements

API – Monitoring requires a minimum of:

- 8 CPUs
- 8 GB of RAM
- 50 GB of free disk space

*Note:* If API – Monitoring is installed on the same server as the Portal database, these hardware requirements are in addition to Portal requirements.

### 2.2.2 Software requirements

Before you can install the API, Portal and its prerequisites must be installed, either on the server where you are installing the API or on a separate server. See <u>Determine where to install API – Monitoring</u>.

Portal databases must be running on SQL Server or SQL Server Express 2017, 2016, 2014, 2012 SP1 or 2008 R2 (64-bit). The latest SQL Server service packs and updates are recommended.

TCP/IP must be enabled for the SQL Server instance and the SQL Server Browser service must be running.

The following prerequisites must be installed on the server where you install the API:

- .NET Framework 4.7
- Powershell version 5.0 or later

We recommend using TLS 1.2 for secure communications. API – Monitoring components cannot communicate with each other using TLS 1.0 or SSL 3.0.

### 2.2.3 Required ports

For the API – Monitoring installation to succeed, port 888 must be reserved for internal communications.

In addition, the external ports shown in following table must be free of other services and open for API – Monitoring components.

Component	Port	Protocol	
Registration service	8080	HTTP or HTTPS	
Keycloak endpoint	8081	HTTP or HTTPS	
RabbitMQ endpoint	5672	AMQP (Advanced Message Queuing Protocol)	
	5672 and 5671	AMQPS (secured Advanced Message Queuing	
		Protocol)	
API – Monitoring	80	НТТР	
enapoint	443	HTTPS	

# 2.3 Determine how to secure API – Monitoring

To secure the API, you can select the HTTPS communication protocol when you install the API and select a certificate in .pfx format. A .pfx file contains the certificate and private key for the API server, and is password-protected. The API – Monitoring installer will prompt you for the .pfx file location and password.

For a production environment, Carbonite recommends obtaining a certificate from a Certificate Authority. For a test environment, you can create a self-signed certificate. The certificate's identity must match the API – Monitoring hostname provided during the installation, and must be trusted by the machine where it is installed.

Alternatively, you can use a third-party TLS termination proxy to handle secure connections, and select the HTTP communication protocol when you install the API. Because Vault Reporting services can only communicate with API – Monitoring with a secured channel using TLS, a TLS termination proxy is required if the API is installed with the HTTP communication protocol.

# 2.4 Set up a public domain name or IP address

When you install API – Monitoring, you must enter a public domain name or IP address for the API. API – Monitoring must have a different host name or IP address than Portal (e.g., api.domain.com and portal.domain.com). Do not use a subdirectory of the Portal URL for the API (e.g., portal.domain.com/api).

If you enter a domain name, the name must be properly registered with your domain server or it will not be available outside the system. To verify that clients can access API – Monitoring, use the ping command to check that the domain name resolves on the client system that is trying to access the API.

# 2.5 Plan ongoing SQL Server database maintenance

For best API performance, ongoing SQL Server database maintenance is required.

We recommend rebuilding table indexes for the Monitoring database nightly, or at least once per week. To achieve this, you can create a SQL scheduled job. A fill factor setting of 75-80% is recommended. For more information, see Microsoft SQL Server documentation: <u>https://docs.microsoft.com/en-us/sql/relational-databases/indexes/reorganize-and-rebuild-indexes?view=sql-server-2014</u>. To rebuild indexes using a script, the following page provides a starting point: <u>https://gallery.technet.microsoft.com/scriptcenter/Script-for-rebuilding-all-8d079754</u>

# **3** Install or upgrade API – Monitoring

After planning your API – Monitoring installation as described in <u>Prepare for an API – Monitoring installation</u>, you can install the API and verify that API – Monitoring is working. See <u>Install API – Monitoring</u> and <u>Verify an API – Monitoring installation</u>. If a previous version of the API is installed, you can upgrade API – Monitoring. See <u>Upgrade API – Monitoring</u>.

After API – Monitoring 1.5 is installed, you can configure and maintain the installation. See <u>Configure and</u> <u>maintain API – Monitoring</u>.

You can also uninstall API - Monitoring or reinstall API - Monitoring.

# 3.1 Install API – Monitoring

To install API — Monitoring:

1. Sign in as an administrator to the server where you want to install API – Monitoring.

*Note:* Carbonite recommends installing API – Monitoring on its own server where no Portal components are installed. For more information, see <u>Determine where to install API – Monitoring</u>.

- 2. Double-click the API Monitoring installation kit.
- 3. On the Welcome page, click Next.

🔵 Carbonite Server I	Backup API - InstallShield Wizard	×
	Welcome to the InstallShield Wizard for Carbonite Server Backup API	
	The InstallShield(R) Wizard will install Carbonite Server Backup API on your computer. To continue, click Next. WARNING: This program is protected by copyright law and international treaties.	
InstallShield	<u>N</u> ext > Cance	el

- 4. On the License Agreement page, read the license agreement. Select I accept the terms in the license agreement, and then click Next.
- 5. On the Destination Folder page, note the API installation location (C:\Program Files\Carbonite Server Backup API), and then click **Next**.

*Note:* The Collector, Monitoring, Registration and Message Bus Authentication services will be installed in subfolders in C:\Program Files\Carbonite Server Backup API. The Message Bus Authentication service subfolder name will be "RMQ Authentication Server". The remaining components will be installed in their default locations:

- Adopt OpenJDK: C:\Program Files\AdoptOpenJDK
- Keycloak: C:\keycloak-3.2.1.Final
- Erlang : C:\Program Files\erl9.2
- RabbitMQ: C:\Program Files\RabbitMQ Server

Installation logs for all API components are saved in C:\Logs.

- 6. On the API Configuration page, do the following:
  - a. Select the communication protocol: https or http

If you select the https protocol, a certificate in .pfx format is required. If you select the http protocol, a third-party TLS termination proxy is required. See <u>Determine how to secure API – Monitoring</u>.

b. Type the public domain name or IP address for the API (e.g., api.carbonite.com or 192.168.1.20).
 Do *not* include http:// or https:// with the domain name or IP address.

API – Monitoring must have a different IP address or host name than Portal (e.g., api.domain.com and portal.domain.com). Do not use a subdirectory of the Portal URL for the API (e.g., portal.domain.com/api).

As described in <u>Set up a public domain name or IP address</u>, a domain name must be properly registered with your domain server so that clients can access the system.

#### Click Next.

😑 Carbonite Server Backup API - InstallShield Wizard			×
API Configuration			•
Please select the communication protocol to use for API M	onitoring		
https			
() http			
Carbonite recommends the use of https. Customers should and benefits.	d only rely on http at	ter careful review of	the risks
Please enter the public domain name or IP Address for AP	1 - Monitoring (e.g.,	api.carbonite.com)	
InstallShield	< Back	Next >	Cancel

If you select the https communication protocol, the Certificate Selection page appears. On the Certificate Selection page, do the following:

a. Click **Browse**. In the Select Certificate File dialog box, navigate to the certificate file in .pfx format and click **Open**.

The certificate's identity must match the API – Monitoring hostname provided during the installation.

For a production environment, Carbonite recommends obtaining a certificate from a Certificate Authority. For a test environment, you can create a self-signed certificate. If the certificate is self-signed, it must be copied into the Trusted Root Certification Authorities Certificate store for the server.

- b. In the password box, enter the password for the certificate file in .pfx format.
- c. Click Next.

🔵 Carbonite Server Backup API - InstallShield Wizard	
Certificate Selection	C
Please select the Certificate file you wish to use to setup	KeyCloak:
Select Certificate File	Browse
Please enter the password for this certificate	e:
InstallShield	< Back Next > Cancel

7. On the Portal SQL Server page, type the name of the database server that hosts the Carbonite Server Backup Portal database (e.g., WINDOWSSERVER). If the default SQL Server instance is not used, include the instance name (e.g., WINDOWSSERVER\SQLSERVER).

*Note:* If you are installing API – Monitoring on the same server as the Portal SQL Server instance, you can type <code>localhost</code> as the database server name. Do not type (<code>local</code>) as the database server name.

Two databases will be installed in the Portal SQL Server instance: a Monitoring database, which will store data from Vault Reporting services, and a KeyCloak database.

Sarbonite Server Backup API - InstallShield Wizard	×
Portal SQL Server	
Please enter the name or IP address of the database server that hosts the Carbonite Server Backup Portal datab If the default SQL Server instance is not used, include the instance name (e.g., WINDOWSSERVER\SQLSERVI Two databases will be installed in the Portal SQL Server instance.	ise. ER).
Connect Using:	
Windows Authentication	
Suc Server authentication using the Login ID and password below	
Password:	
InstellShield Cance	

- 8. Do one of the following:
  - To connect to SQL Server using the current Windows credentials, select Windows Authentication.

For Windows authentication, you must be signed in as an administrator that has sysadmin rights to the SQL Server instance where Portal databases are running.

- To connect to SQL Server using SQL Server credentials, select SQL Server authentication using the Login ID and password below. In the Login ID and Password boxes, enter SQL Server credentials.
- 9. Click Install.

The installer begins installing and configuring API components and prerequisites. The Installing Carbonite Server Backup API screen shows the component currently being installed, and the installation progress.

Note: The installation of API components can take a long time (e.g., 15 minutes).

🔵 Carbonite Se	rver Backup API - InstallShield Wizard	×
Installing	g Carbonite Server Backup API	•
5	The program features you selected are being installed. Installing package KeyCloak-3.2.1.Final KeyCloak-3.2.1.Final	
InstallShield	P	Cancel

When the installation is finished, the InstallShield Wizard shows information that is required for registering Portal and Vault Reporting services to the API.

**IMPORTANT:** Copy the Client ID, Client Secret and Registration URL values from the InstallShield Wizard page. These values are required for registering Portal and Vault Reporting services to the API.

**IMPORTANT:** To ensure the security of your data, keep the Client ID and Client Secret values private and secure.

🔵 Carbonite Server E	Backup API - InstallSh	ield Wizard	×
	The InstallShield V API. Click Finish to	Vizard has successfully installed Carbonite Server Backup exit the wizard.	
	Record the follo	by the second seco	
	Client Secret:	nrKbCpJ0lHqaKIY74AMOVz1tWhsS37K2WyyKXLy+OhtP	
	Registration URL:	https://api.carbonite.com:8080	
		Finish	_
InstellShield		Finish	

10. Click Finish.

Once API – Monitoring is installed, you can view detailed information about available API – Monitoring calls. See <u>View documentation for API – Monitoring calls</u>.

# 3.2 Upgrade API – Monitoring

You can upgrade API – Monitoring. After an upgrade, the Client ID, Client Secret and Registration URL values used to register Vault Reporting services remain the same as before the upgrade. You can also register Portal to the API using these values.

*Note:* You cannot change from Windows authentication to SQL Server authentication during an upgrade.

To upgrade API – Monitoring:

- 1. Sign in as an administrator on the server where you want to upgrade API Monitoring.
- 2. Double-click the API Monitoring installation kit.

Note: There could be a significant delay before the upgrade begins.

3. On the Welcome page, click Next.

The installer begins upgrading API components. The Installing Carbonite Server Backup API screen shows the component currently being upgraded and the upgrade progress.

Note: The upgrade can take a long time (e.g., five minutes).

🔵 Carbonite Se	rver Backup API - InstallShield Wizard	×
Installing	g Carbonite Server Backup API	0
5	The program features you selected are being installed. Installing package Erlang Erlang	
		Cancel

4. When the upgrade is finished, click **Finish**.

Once API – Monitoring is installed or upgraded, you can view detailed information about available API – Monitoring calls. See <u>View documentation for API – Monitoring calls</u>.

### 3.3 Troubleshoot an API – Monitoring installation

If a problem occurs during an API – Monitoring installation, you can run the installer in debug mode to collect more detailed logs.

To troubleshoot an API – Monitoring installation:

- 1. Create a directory for saving the detailed installation logs (e.g., C:\Logs).
- 2. At a command prompt, run the following command:

```
installKitName / debuglog"location\fileName"
```

Where:

- installKitName is the name of the API Monitoring installation kit.
- location is the directory for saving the logs (created in Step 1).
- fileName is the file name of the log file.

For example, the command could be:

```
CarboniteServerBackupAPI.Monitoring.1.5.x.xxxx
/debuglog"C:\Logs\Install.log"
```

# 3.4 Verify an API – Monitoring installation

To verify the API – Monitoring installation:

- 1. Check that the following services are running:
  - Carbonite Registration Service
  - Carbonite Server Backup API Collector
  - Carbonite Server Backup API Message Bus Authentication
  - Carbonite Server Backup API Monitoring
  - KeyCloak
  - RabbitMQ
- 2. In a web browser, go to the Swagger UI for API Monitoring:

https://APIdomainNameOrIPaddress/monitoring/swaggerui/index

Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of <u>Install</u> <u>API – Monitoring</u>.

For example: https://api.carbonite.com/monitoring/swaggerui/index For more information about the Swagger UI, see <u>View documentation for API – Monitoring calls</u> and <u>Test API – Monitoring calls</u>.

# 3.5 Uninstall API – Monitoring

To uninstall API – Monitoring:

- 1. In the Control Panel, click **Uninstall a program**.
- 2. On the Uninstall or change a program page, click an API Monitoring component (e.g., Carbonite Registration Service) and then click **Uninstall.**
- 3. In the confirmation dialog box, click **Yes**.
- 4. Double-click the API Monitoring installation kit.
- 5. On the Welcome page, click **Next**.

6. On the Ready to Remove API – Monitoring Components page, click Begin Removal.



7. When a message states that the removal of pre-existing components is complete, click **Cancel**.

) Carbonite Server Backup API - Ins	tallShield Wizard	×
Ready to Remove AP	I - Monitoring Components	
Before we can reinsta components that cor	all API - Monitoring, we must remove any API - Monitoring t are installed on this machine. To begin removing API mponents, please click the button below.	3
Rem	ioval of pre-existing components complete!	
InstallShield	< Back Next >	Cancel

- 8. In the confirmation message box, click Yes.
- 9. On the wizard interrupted page, click Finish.
- 10. If desired, manually uninstall Adopt OpenJDK, which was installed with API Monitoring 1.5.

If you uninstall API – Monitoring version 1.4.2 or earlier, you can manually uninstall the following software that was installed with earlier API versions: Python and Oracle Java Runtime Environment.

### 3.6 Reinstall API – Monitoring

*Important:* Vault Reporting services do not remain registered to API – Monitoring after you reinstall the API. You must re-register Reporting services to API – Monitoring if you reinstall the API. See <u>Register Director</u> <u>Reporting services to the API</u>.

To reinstall API – Monitoring:

- 1. In the Control Panel, click **Uninstall a program**.
- 2. On the Uninstall or change a program page, click an API Monitoring component (e.g., Carbonite Registration Service) and then click **Uninstall.**
- 3. Double-click the API Monitoring installation kit.
- 4. On the Welcome page, click **Next**.
- 5. On the Ready to Remove API Monitoring Components page, click Begin Removal.

Ready to Remove API - Moni	toring Compone	nts	0
Before we can reinstall API - Monit components that are installed components, ple	oring, we must remove any on this machine. To begin ase click the button below.	API - Monitoring removing API	
Beg	in Removal		

6. When a message states that the API components have been removed, click **Next**. Complete the installation as described in Install API – Monitoring.

Ready to Rem	ove API - Monitoring Components	
,		
Before w	e can reinstall API - Monitoring, we must remove any API - Monitori	ing
com	components, please click the button below.	
	Removal of pre-existing components complete!	

# 4 Configure and maintain API – Monitoring

# 4.1 Configure the refresh interval for job and safeset data

To improve the performance of job and safeset calls, API – Monitoring 1.5 caches some required data from Portal databases in the Monitoring database.

Data is refreshed in the Monitoring database every 5 minutes by default, but you can change the refresh interval to a value from 1 minute to 60 minutes. A lower interval (e.g., 1 minute) ensures that data returned by API calls is more up-to-date but requires more resources. A higher interval (e.g., 60 minutes) results in less traffic but the data returned by API calls may be out-of-date.

To configure the refresh interval for job and safeset data:

- 1. In a text editor, open the PlatformAPI.Monitoring.API.exe.config file from the C:\Program Files\Carbonite Server Backup API\Monitoring folder.
- 2. Find the following line in the config file:

```
<add key="PlatFormAPI.Monitoring.API.EtlExecutionIntervalMins" value=""></add>
```

When no value is specified in this line, the default interval of 5 minutes is used.

3. Enter a value from 1 minute to 60 minutes in the line.

For example, if you want the data to be refreshed every 2 minutes, enter the following line:

<add key="PlatFormAPI.Monitoring.API.EtlExecutionIntervalMins" value="2"></add>

4. Save the file.

### 4.2 Replace the API – Monitoring certificate

If API – Monitoring was installed using the HTTPS communication protocol and secured using a certificate, you can replace the certificate if it is approaching its expiry date.

As described in <u>Determine how to secure API – Monitoring</u>, Carbonite recommends obtaining a certificate from a Certificate Authority for a production environment. The certificate must be in .pfx format.

To replace the API – Monitoring certificate:

- Check whether the Microsoft Visual C++ 2015 (or later) Redistributable is installed on the API Monitoring server. If it is not installed, download the vc\_redist.x64.exe file from Microsoft (<u>https://support.microsoft.com/en-ca/help/2977003/the-latest-supported-visual-c-downloads</u>) and install it on the API – Monitoring server.
- 2. In a Powershell window, navigate to C:\Program Files\Carbonite Server Backup API\Tools.
- 3. Run the following command:
  - .\Update-pAPICert.ps1 logPathAndFileName certificate certificatePassword

Where:

- *logPathAndFileName* is the path and filename of a log file for the certificate replacement process.
- *certificate* is the path and filename of the new certificate file in .pfx format.
- *certificatePassword* is the password for the certificate file in .pfx format.

For example, you could run the following command:

```
.\Update-pAPICert.ps1 C:\CertLog.txt C:\Certificate.pfx password
```

4. To check that the API – Monitoring certificate was replaced, open the certificate manager on the machine, and view the certificates in the Local Computer Personal certificate store.



# **5** Register Director Reporting services to the API

To provide vault data through API calls or to delete data in response to requests from Portal, Director Reporting services must be registered to API – Monitoring.

A vault administrator can register a Director Reporting service to the API when the Reporting service is being installed or upgraded, or using a command after the Reporting service is installed. See "Install and register the Reporting service" in the *Director Installation Guide*.

We recommend registering Reporting services to API – Monitoring using a registration token and registration URL. To obtain a registration token, see <u>Obtain a token for registering Director Reporting services to the API</u>.

Director Reporting services can also be registered to the API using Client ID and Client Secret values from the last page of the API – Monitoring installation wizard, but this is not recommended. To protect your data, access to these values should be limited. Keep the Client ID and Client Secret private and secure.

# 5.1 Obtain a token for registering Director Reporting services to the API

Using a script that is provided with API – Monitoring, you can obtain a token for registering Director Reporting services to the API. This ObtainRegistrationToken.ps1 script requires you to enter the Registration URL, Client ID and Client Secret values from the last page of the API – Monitoring installation wizard, and returns a registration token. You can then give the registration token to the person who is registering one or more Director Reporting services to the API.

A registration token is valid for 48 hours after it is created. If someone tries to register a Reporting service to the API using an expired token, an error appears in the API Registration service log file.

To obtain a registration token for registering a Director Reporting service to API - Monitoring:

- 1. Save the ObtainRegistrationToken.ps1 script on a machine that can communicate with the API Monitoring server. Powershell version 5.0 or later must be installed on the machine.
- 2. In a Powershell window, navigate to the directory where the script was saved in Step 1 of this procedure.
- 3. Run the following command:
  - .\ObtainRegistrationToken.ps1
- 4. At the following prompts, enter the required information from the last page of the API Monitoring installation wizard (see Install API Monitoring):
  - Registration Service Address Enter the Registration URL (e.g., https://api.carbonite.com:8080).
  - Client ID Enter the Client ID (Carbonite-Registration-Client).
  - Client Secret Enter the Client Secret (e.g., nrKbCpJOlHqaKlY74AMOVz1tWhsS37K2WyyKXLy+0htP).

The registration token is returned in the Powershell window. You can copy the token and provide it to the person who is registering Director Reporting services to the API.

# **6** View documentation for API – Monitoring calls

Client applications can make read-only calls to API – Monitoring to obtain information about entities such as agents, jobs and safesets in Portal and in Director vaults.

For detailed information about available API – Monitoring calls, view documentation in the Swagger UI that is installed with the API. For information about Swagger, see <u>https://swagger.io</u>.

To view documentation for API – Monitoring calls:

1. In a web browser, go to the Swagger UI for API – Monitoring:

https://APIdomainNameOrIPaddress/monitoring/swaggerui/index

Where *APIdomainNameOrIPaddress* is the public domain name or IP address entered in Step 6 of Install API – Monitoring.

For example: https://api.carbonite.com/monitoring/swaggerui/index

*Note:* You might not be able to view the Swagger UI in Internet Explorer 11. If you encounter problems, use a different web browser to access the Swagger UI.

CARBONITE	https://api/monitoring/swagger/v1	~

#### Carbonite Server Backup API - Monitoring v1

Carbonite Server Backup API - Monitoring is a public OData Web API that enables you to monitor companies, users, agents and jobs in Portal. For more information about OData, see the OData web site: <u>http://www.odata.org/</u>.

agent	Show/Hide   List Operations   Expand Operations
company	Show/Hide List Operations Expand Operations
datadeletiondetails	Show/Hide List Operations Expand Operations
job	Show/Hide   List Operations   Expand Operations
safeset	Show/Hide   List Operations   Expand Operations
vault	Show/Hide   List Operations   Expand Operations
version	Show/Hide   List Operations   Expand Operations

2. To view available calls for an entity (e.g., agent), click the entity name or click **List Operations** for the entity.

agent	Show/Hide List Operations Expand Operation
company	Show/Hide   List Operations   Expand Operation
Available calls appear below the entity name.	
agent	Show/Hide List Operations Expand Operation

age	ent		Show/Hide	List Operations	Expand Operations
	GET	/monitoring/agents			
	GET	/monitoring/agents({agentId})			

3. To view information about a particular call, click the operation or click **Expand Operations**.

age	ent		Show/Hide	List Operations	Expand Operations
	GET	<u>/monitoring/agents</u>			
	GET	/monitoring/agents({agentId})			

Information about the call appears below the call name.

4. To view detailed information about the call, click **Model**.



Information about the call, including descriptions of each value returned, appears.

agent	Show/Hide	List Operations	Expand Operations
GET /monitoring/agents			
<b>RESPONSE CLASS (STATUS 200)</b> OK			
MODEL EXAMPLE VALUE			
ODataListResponse[Agent] {			
@odata.nextLink (string, optional),			
@odata.count (integer, optional): [format: int32],			
@odata.context (string, optional),			
value (Array[Agent], optional)			
}Agent {			
id (string, optional): [format: uuid] Unique identifier (GUID)	for an agent i	n Portal. This GUID	is
automatically generated when an agent registers to Po	ortal,		
companyid (string, optional): [format: uuid] Unique identif	ier (GUID) for	the agent's compa	ny,
name (string, optional): Agent name,			
description (string, optional): Agent description,			
version (string, optional): Agent version,			
operatingSystem (string, optional): Operating system of the	ne computer w	here the agent is i	nstalled,
<b>hostName</b> (string, <i>optional</i> ): Name of the computer where	the agent is i	nstalled,	
<ul> <li>@odata.nextLink (string, optional, @odata.count (integer, optional,: [format: int32], @odata.context (string, optional, value (Array[Agent], optional, }</li> <li>JAgent {         id (string, optional,: [format: uuid] Unique identifier (GUID) automatically generated when an agent registers to Po companyId (string, optional,: [format: uuid] Unique identifien name (string, optional,: Agent name, description (string, optional,: Agent description, version (string, optional,: Agent version, operatingSystem (string, optional,: Operating system of th hostName (string, optional,: Name of the computer where</li> </ul>	for an agent i ortal, ier (GUID) for ne computer w the agent is in	n Portal. This GUID the agent's compa there the agent is i nstalled,	) is ny, nstalled,

# 7 Monitor protected data

As described in <u>Monitoring overview</u>, client applications can make read-only calls to API – Monitoring to obtain information about agents, jobs and other entities in Portal and registered vaults. Client applications are authenticated and authorized using Keycloak: the third-party authorization server that is installed with API – Monitoring.

After API – Monitoring is installed and Director Reporting services are registered to the API (see <u>Install API –</u> <u>Monitoring</u> and <u>Register Director Reporting services to the API</u>), you must do the following so that Keycloak can authenticate and authorize client applications:

- 1. <u>Create a Keycloak admin user</u>. This admin user is required before you can create API Monitoring clients in Keycloak.
- 2. <u>Create API Monitoring clients</u>. API Monitoring clients must be created in Keycloak so they can be authenticated and authorized when they make calls to the API.
- 3. <u>Obtain the ID and secret for a client</u>. A Client ID and secret is generated for each client created in Keycloak.

*Note:* These Client ID and secret values are client-specific. These values are not the same as the values shown on the last page of the API – Monitoring InstallShield wizard.

Client applications use these Client ID and secret values to obtain access tokens and make API – Monitoring calls. For more information about access tokens, see <u>Obtain an access token</u>. For examples of client applications that retrieve data using API calls, see the API Client sample code that is available from Carbonite.

You can also check that the system is working by testing API – Monitoring calls using the Swagger UI that is installed with API – Monitoring. See <u>Test API – Monitoring calls</u>.

# 7.1 Create a Keycloak admin user

You must create a Keycloak admin user before you can create clients that obtain data from API – Monitoring. Keycloak is the third-party authorization server that is installed with the API and is used to authenticate and authorize client applications.

To create a Keycloak admin user:

- 1. On the server where API Monitoring is installed, open a Powershell window.
- 2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts
- 3. Run the following command:

.\KeyCloak-AddUser.ps1 -u KeycloakAdminUser -p KeycloakAdminPassword

For example, to create an admin user for Keycloak named "keycloakadmin", you could run the following command:

.\KeyCloak-AddUser.ps1 -u keycloakadmin -p strongpassword

# 7.2 Create API – Monitoring clients

Before a client application can obtain data from API – Monitoring, you must create the client in Keycloak using a script that is provided with the API. Each client can have one of the following three access types:

- Admin access type. Clients with the Admin access type have unrestricted access to all data in a Portal instance and all vault information. Typically, this client type is used by the service provider that hosts the Portal/Platform API instance. See <u>Create an API client with the Admin access type</u>.
- Partner access type. Clients with the Partner access type have access to all data in a Portal instance but do not have access to vault information. See <u>Create an API client with the Partner access type</u>.
- Reseller access type. Clients with the Reseller access type have access to data for one or more specific companies in a Portal instance, and safesets that belong to agents in these companies. This type of client is equivalent to a Portal Admin user in a parent company. If the client has access to a parent company, it also has access to data for the child companies. See <u>Create an API client with the Reseller access type</u>.

### 7.2.1 Create an API client with the Admin access type

A client with the Admin access type has unrestricted access to all data in a Portal instance and all vault information. Typically, this client type is used by the service provider that hosts the Portal/Platform API instance.

To create an API client with the Admin access type:

- 1. On the server where API Monitoring is installed, open a Powershell window.
- 2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts
- 3. Run the following command:

.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final" "https://APIdomainNameOrIPaddress:8081/auth" KeycloakAdminUser KeycloakAdminPassword Carbonite-Monitoring APIclientName urn:carb:sb:api:monitoring admin

Where:

- APIdomainNameOrIPaddress is the API domain name or IP address entered in Step 6 of Install API Monitoring.
- *KeycloakAdminUser* and *KeycloakAdminPassword* are the Keycloak admin user name and password created in <u>Create a Keycloak admin user for creating API clients</u>.
- APIclientName is the name of the API client that you want to create.

For example:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keyCloak-3.2.1.Final"
"https://api.carbonite.com:8081/auth" KeyCloakAdmin strongpassword
Carbonite-Monitoring APIadmin urn:carb:sb:api:monitoring admin
```

If you selected the HTTP communication protocol when you installed the API, you can run the command using HTTP. For example:

.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
"http://api.carbonite.com:8081/auth" KeycloakAdmin strongpassword CarboniteMonitoring APIadmin urn:carb:sb:api:monitoring admi

### 7.2.2 Create an API client with the Partner access type

A client with the Partner access type has access to all data in a Portal instance but does not have access to vault information.

To create an API client with the Partner access type:

- 1. On the server where API Monitoring is installed, open a Powershell window.
- 2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts
- 3. Run the following command:

.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
"https://APIdomainNameOrIPaddress:8081/auth" KeycloakAdminUser KeycloakAdminPassword
Carbonite-Monitoring APIclientName urn:carb:sb:api:monitoring partner

Where:

- APIdomainNameOrIPaddress is the API domain name or IP address entered in Step 6 of <u>Install API</u> <u>— Monitoring</u>.
- KeycloakAdminUser and KeycloakAdminPassword are the Keycloak admin user name and password created in <u>Create a Keycloak admin user</u>.
- APIclientName is the name of the API client that you want to create.

#### For example:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keyCloak-3.2.1.Final"
"https://api.carbonite.com:8081/auth" KeyCloakAdmin strongpassword
Carbonite-Monitoring Partner1 urn:carb:sb:api:monitoring partner
```

If you selected the HTTP communication protocol when you installed the API, you can run the command using HTTP. For example:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keyCloak-3.2.1.Final"
"http://api.carbonite.com:8081/auth" KeyCloakAdmin strongpassword Carbonite-
Monitoring Partner1 urn:carb:sb:api:monitoring partner
```

### 7.2.3 Create an API client with the Reseller access type

A client with the Reseller access type has access to data for one or more specific companies in a Portal instance but does not have access to vault information. This type of client is equivalent to a Portal Admin user in a parent company. If the client has access to a parent company, it also has access to data for the child companies.

To create an API client with the Reseller access type:

- 1. On the server where API Monitoring is installed, open a Powershell window.
- 2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts

#### 3. Run the following command:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
"https://APIdomainNameOrIPaddress:8081/auth" KeycloakAdminUser KeycloakAdminPassword
Carbonite-Monitoring APIclientName urn:carb:sb:api:monitoring reseller
companyGUID
```

Where:

- APIdomainNameOrIPaddress is the API domain name or IP address entered in Step 6 of <u>Install API</u> — <u>Monitoring</u>.
- KeycloakAdminUser and KeycloakAdminPassword are the Keycloak admin user name and password created in <u>Create a Keycloak admin user</u>.
- APIclientName is the name of the API client that you want to create.
- companyGUID is the globally unique identifier (GUID) of the company for which the API client can access data. To obtain the company GUID for a client, see <u>Obtain the reseller's Company ID from</u> <u>Portal</u>.

For example:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keyCloak-3.2.1.Final"
"https://api.carbonite.com:8081/auth" keyCloakadmin strongpassword Carbonite-
Monitoring Site1Clienturn:carb:sb:api:monitoring reseller 7a877d8c-b20b-
4f70-984d3a990719
```

If you selected the HTTP communication protocol when you installed the API, you can run the command using HTTP. For example:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keyCloak-3.2.1.Final"
"http://api.carbonite.com:8081/auth" keyCloakadmin strongpassword Carbonite-
Monitoring Site1Clienturn:carb:sb:api:monitoring reseller 7a877d8c-b20b-
4f70-984d3a990719
```

#### 7.2.3.1 Obtain the reseller's Company ID from Portal

To obtain the reseller's Company ID from Portal:

- 1. In a web browser, go to the Portal website. Sign in to Portal as a super user.
- 2. Click **Sites** on the navigation bar.

Note: A "site" in Portal is the same as a "company" in API – Monitoring.

3. Find the reseller's site. To find the site, you can enter criteria in the filter row under the column headings.

O Create New Site O Assign	Child Sites	1 2 3 4 - 34	Show 2	🔁 👻 Records per page
Site Name .	Contact	Phone	Sta	Filter row
				•

4. Right-click the reseller's site, and choose **Inspect** (in Google Chrome) or **Inspect Element** (in Firefox or Internet Explorer) from the menu.

The source code for the site appears. The companyld appears in the line before the company (site) name.

5. Record the companyId value (a GUID).

### 7.3 Obtain the ID and secret for a client

Before a client can access data using the API, the authorization server administrator must provide the customer's client ID and secret values. The client can then obtain a JWT access token from the authorization server. See <u>Obtain an access token</u>.

*Note:* The client ID and secret provided through this process are not the same as the client ID and secret values shown on the last page of the API – Monitoring InstallShield wizard. The values in this procedure are client-specific.

To obtain the ID and secret for a client:

1. In a web browser, go to the KeyCloak administration console:

https://APIdomainNameOrIPaddress:8081/auth/admin

Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of <u>Install</u> <u>API – Monitoring</u>.

For example: https://api.carbonite.com:8081/auth/admin

- 2. Log in to the KeyCloak administration console with the Keycloak admin user that you created in <u>Create</u> a Keycloak admin user.
- 3. Point to Select realm in the top left corner, and select the Carbonite-Monitoring realm from the list.



4. Select Clients from the menu on the left.



5. Click the client's Client ID link.

Carbonite-Monitoring ~	Clients 😨	
Configure	Search Q	
া Realm Settings	Client ID	
🕥 Clients	account	
	admin-cli	
🗞 Client Templates	broker	
🧱 Roles	Carbonite-Monitoring-Admin	
⇒ Identity Providers	realm-management	
Ser Federation	ResellerUser ResellerUser2	

6. Go to the Settings tab. Copy the Client ID value and provide it to the customer.

Carbonite-Monitoring ~	Clients » ResellerUser	
Configure	ResellerUser 👕	
·····gare	Settings Credentials Roles	Mappers 🚱 Scope 🚱
🍞 Clients	Client ID 😡	ResellerUser
🚷 Client Templates		·
Roles	Name 🖗	
≓ Identity Providers	Description 🚱	ResellerUser

7. Go to the Credentials tab. Copy the Secret value and provide it to the customer.

*Important:* If the Credentials tab is hidden, enter a description for the client in the **Description** field, and then click **Save** at the bottom of the page. The credentials tab will then appear.

Carbonite-Monitoring ~	Clients » ResellerUser					
Configure	Reseller	User 👕				
🚻 Realm Settings	Settings	Credentials	Roles	Mappers 🚱	Scope 🔞	Authorization
😚 Clients		Client Authenti	ator 🔞	Client Id and S	ecret	$\mathbf{\vee}$
💫 Client Templates 📰 Roles			Secret	3a2f87d2-c917	450e-94da-828	c834106a3

### 7.4 Test API – Monitoring calls

This section describes how to test API – Monitoring calls using the Swagger UI that is installed with the API.

For examples of client applications that retrieve data using API calls, see the API Client sample code that is available from Carbonite.

An authorization parameter, or access token, is required for each API – Monitoring call. This access token is a JSON Web Token (JWT) that specifies the data that the client can access. See <u>Obtain an access token</u>.

Each API – Monitoring call can return a maximum of 500 records. To obtain more than 500 records, and for best performance, use query options in API calls. See <u>Best practice: Use query options in API – Monitoring calls</u>. For more information about OData, see the OData website: <u>http://www.odata.org</u>

To test API – Monitoring calls:

1. In a web browser, go to the Swagger UI for API – Monitoring:

https://APIdomainNameOrIPaddress/monitoring/swaggerui/index

Where *APIdomainNameOrIPaddress* is the public domain name or IP address entered in Step 6 of Install API – Monitoring.

For example: https://api.carbonite.com/monitoring/swaggerui/index

*Note:* You may not be able to view the Swagger UI in Internet Explorer 11. If you encounter problems, use a different web browser to access the Swagger UI.

•

CARBONITE®	https://api/monitoring/swagger/v1

#### Carbonite Server Backup API - Monitoring v1

Carbonite Server Backup API - Monitoring is a public OData Web API that enables you to monitor companies, users, agents and jobs in Portal. For more information about OData, see the OData web site: <u>http://www.odata.org/</u>.

agent	Show/Hide   List Operations   Expand Operations
company	Show/Hide   List Operations   Expand Operations
datadeletiondetails	Show/Hide   List Operations   Expand Operations
job	Show/Hide   List Operations   Expand Operations
safeset	Show/Hide   List Operations   Expand Operations
vault	Show/Hide   List Operations   Expand Operations
version	Show/Hide List Operations Expand Operations

2. Click the entity (e.g., agent) for which you want to make a call, or click List Operations for the entity.

agent	Show/Hide	List Operations	Expand Operations
2			
company	Show/Hide	List Operations	Expand Operations

3. Click the call that you want to test or click **Expand Operations** for the entity.

agent	Show/Hide	List Operations	Expand Operations
GET /monitoring/agents RESPONSE CLASS (STATUS 200) OK			

4. To test the call, click Example Value. Enter parameters in the fields, and then click Try it out!

You must enter an authorization parameter (access token) for each API – Monitoring call. To obtain an access token, submit a post request to the authorization server using information from your authorization server administrator. See <u>Obtain an access token</u>.

In general, strings must be enclosed in single quotation marks in filters. GUID, Boolean, int and enum values do not need to be enclosed in quotation marks in filters. For more information, see <u>OData</u> <u>documentation</u>.

		Show/Hide	List Operatio	ons   Expand	operau
GET /monito	ring/agents				
RESPONSE CLA OK	SS (STATUS 200)				
MODEL EXAM	PLE VALUE				
{ "@o "@o "Va	data.nextLink": "string", data.count": 0, data.context": "string", lue": [				
{	"id": "string",				-
Response Conte	ant Type [application/json,api-version=1 ▼]				
PARAMETERS Parameter	Value	Description	Parameter	Data Type	
Authorization	(required)	The access token	header	string	
	u i i	(Format: Bearer [access_token]).			
\$select		(Format: Bearer [access_token]). Properties to include in the response (Example: id,name).	query	string	
Sselect Sfilter		(Format: Bearer [access_token]). Properties to include in the response (Example: id, name). Filter the response using an expression (Example: name eq 'Company Name).	query	string	
Sselect Sfilter SorderBy		(Format: Bearer [access_token]). Properties to include in the response (Example: id,name). Filter the response using an expression (Example: name eq 'Company Name'). Sort the response by one or more properties (Example: status,name).	query query	string string string	
Sselect Sfilter SorderBy Sskip		<ul> <li>(Format: Bearer [access_token]).</li> <li>Properties to include in the response (Example: id,name).</li> <li>Filter the response using an expression (Example: name eq 'Company Name').</li> <li>Sort the response by one or more properties (Example: status,name).</li> <li>Exclude the first n results from the response.</li> </ul>	query query query query	string string string integer	
Sselect Sfilter SorderBy Sskip Stop		<ul> <li>(Format: Bearer [access_token]).</li> <li>Properties to include in the response (Example: id,name).</li> <li>Filter the response using an expression (Example: name eq 'Company Name').</li> <li>Sort the response by one or more properties (Example: status,name).</li> <li>Exclude the first n results from the response.</li> <li>Include only the first n results in the response.</li> </ul>	query query query query query	string string string integer integer	

### 7.4.1 Obtain an access token

An authorization parameter, or access token, is required for each API – Monitoring call. This token is a JSON Web Token (JWT) that specifies the data that the client can access.

To obtain an access token:

- 1. Contact your authorization server administrator for the following required information:
  - URL of the authorization server endpoint
  - Client ID
  - Client Secret

The administrator obtains the Client ID and Secret values as described in <u>Obtain the ID and secret for a client</u>.

2. To obtain an access token, submit a post request using a REST client, as described in the following table:

Method:	POST
URL of the Keycloak	https://APIdomainNameOrIPaddress:8081/auth/realms/carbonite-
endpoint	monitoring/protocol/openid-connect/token
	Where <i>APIdomainNameOrIPaddress</i> is the API domain name or IP address entered in Step 6 of <u>Install API – Monitoring</u> . For example:
	https://api.carbonite.com:8081/auth/realms/carbonite-
	monitoring/protocol/openid-connect/token
Body type:	x-www-form-urlencoded
Form fields	
client_id	clientID
grant_type	client_credentials
client_secret	clientSecret

Steven				- 🗆 ×
Eait View Help     Toport Runner	🚦 My Works	space 🗸 🔹 🧿	SYNC OFF 😧 🎜 🌲	V Sign In
Q Filter       History     Collections	Obtain access token X + •••• • Obtain access token		No Environment	<ul> <li>✓ <ul> <li>✓ <ul> <li>✓ <ul> <li>✓ <ul></ul></li></ul></li></ul></li></ul></li></ul>
Clear all Today POST http://192.168.14.159:8081/auth/real ms/carbonite-monitoring/protocol/o penid-connect/token	POST     http://192.168.14.159:8081/aut       Authorization     Headers (4)     Body •     Pre       •     form-data     •     x-www-form-urlencoded     •     r	h/realms/carbonite-monitoring/protocol/ope. -request Script Tests aw binary	Params Send Y	Save 🔆
<ul> <li>GET http://192.168.14.159:8081/auth/real ms/carbonite-monitoring/protocol/o penid-connect/token</li> <li>GET http://192.168.14.159:8081/auth/real ms/carbonite-monitoring/protocol/o penid-connect/token</li> </ul>	Key       Image: Constraint of the second of th	Value ResellerUser client_credentials 3a2f87d2-c917-450e-94da-828c834106a3	Description	••• Bulk Edit
	New key Body Cookies (1) Headers (7) Test Re	Value	Description Status: 200 OK Time: 305 ms	Size: 3.12 KB
	Pretty         Raw         Preview         JSON ∨           1         k           2         "access_token": "eyJnbGciOiJSUZIIN           1         k           2         "access_token": "eyJnbGciOiJSUZIIN           0         ioXVIIEMTEINTENTENTCACL3pc8H0013NtccMV           0         iiiXVIIEMTEINTENTENTCACL3pc8H0013NtccMV           0         iiiXVIIEMTEINTENTCACL3pc8H0013NtccMV           0         iiiXVIIEMTEINTENTCACL3pc8H0013NtccMV           0         iiiXVIIEMTEINTENTCACL3pc8H0013NtccMV           0         iiiXVII2Vs2M8950502c16115ENI44MV           1         iiXVI2Vs2M8950502c1615ENI44MV           2         iiXVI2Vs2M8950502c1615ENI44MV           2         iiXVI2Vs2M8950502c1615ENI44MV           2         iiXVI2Vs2M8950502c1615ENI44MV           2         iiXVI2Vs2M8950502c1615ENI44MV           2         iiXVI2Vs2M8950502c1615ENI44MV           2         iiXVI2Vs2M8950502c16155VL14M704WT04           2         iiXVI2Vs2M8950502c1625M211M           3         "explires_in": 300,           3         "refresh_explires_in": 1800,           4         "refresh_explires_in": 1800,           5         "refresh_explires_in": 1800,           6         "refresh_explires_in": 1800, <td>ilsInR5cCIgOiAiSidUIiwia2lkIiA6ICJzNFpGY zE3viJuNjgxLTRiYjEtoTahNSiiZTUyHThiM8RMM Hkwoi&amp;WHKyjE20C4HKCAHTk6D0AH%95MAR0Li if6IjkwHWY42CZ1LWFJNjTtNGHWFS04NDEALTczz GltZSI6MCwic2Vzc21vbJ9zGF02SI6ImQ0HmJMY MudClawBrcyIsInz2XcHUJX2mS2J3KA08I Jw2KJ10nsicm9zZWH0JsiWIX38y03R1Y3Rpb Jw2CJWJCYzJ2NF505NQi0J3SZWH0SX13FX Jw2CJWJUYZYJ2NF505NQi0J3SZWH0SX12ST Jw2LTMANHG0J3ZXAJ2WH0JWFY291bnQtcmVzJ PCC2WUQHOPy_TA6XNG308T4FDJGHtczKSMUPYIY OLAX30H7YKv_VIJ3QXN7W52NS4SA4J61pubjcHkl ezV&amp;FjKr2M9qEsBuhRomBHBF1W02r7szrt1gtVJ NIISINR5cCIgOiAiSIdUIiwia21kLIA6ICJ2NFpGG</td> <td>RecXZhZXZiNXdTY0NkQ3dMZDFQY     QQiLCIeHAiOjEJNjQiNTU0NaQEI     JNWhcty9NXJib25QdUUTM9ua     JAwdjik0MzRhOCISInR5cCIGIKJIY     JIkLNUwHmQtNGFKZIINMTJMLVZIY     MiloisidUIXZFIdGhvcmJ6YXRpb     H2C2GINVy5JYX3IONIONE     WIZZXILCZWcmWZXJyZNR4dXNI2     MILC314dHUZSIGIAI2CySbGVI     MxAN0wXGQJkavzhAprQFXMc2Ro     BoGMP5jGTVL0JggLsiNr6QL1zV     caljinTEWtosgNV_SAY85DkwqOlu     VVBzcXZhZXZINXdTY0NkQ3dMZDFQ     VVBzcXZhZXZINXdTY0NkQ3dMZDFQ</td> <td>Save Response WIHVG92dEhjRWI m5i2i16MCwiaWF KNrcmlu2/IImF X11ci1sImF6cCI rfmHTU52DNHYII 44iXX8LTJ1C29 XH101i3UbFuYWa Tptb25pd69yaW5 m5hbW103122X3 m5hbW103122X3 m652ccHPcM39 MF52ccHPcM39 79k2dDxp2Q902c YWIHVG92dEhjRWI Tp532166via au</td>	ilsInR5cCIgOiAiSidUIiwia2lkIiA6ICJzNFpGY zE3viJuNjgxLTRiYjEtoTahNSiiZTUyHThiM8RMM Hkwoi&WHKyjE20C4HKCAHTk6D0AH%95MAR0Li if6IjkwHWY42CZ1LWFJNjTtNGHWFS04NDEALTczz GltZSI6MCwic2Vzc21vbJ9zGF02SI6ImQ0HmJMY MudClawBrcyIsInz2XcHUJX2mS2J3KA08I Jw2KJ10nsicm9zZWH0JsiWIX38y03R1Y3Rpb Jw2CJWJCYzJ2NF505NQi0J3SZWH0SX13FX Jw2CJWJUYZYJ2NF505NQi0J3SZWH0SX12ST Jw2LTMANHG0J3ZXAJ2WH0JWFY291bnQtcmVzJ PCC2WUQHOPy_TA6XNG308T4FDJGHtczKSMUPYIY OLAX30H7YKv_VIJ3QXN7W52NS4SA4J61pubjcHkl ezV&FjKr2M9qEsBuhRomBHBF1W02r7szrt1gtVJ NIISINR5cCIgOiAiSIdUIiwia21kLIA6ICJ2NFpGG	RecXZhZXZiNXdTY0NkQ3dMZDFQY     QQiLCIeHAiOjEJNjQiNTU0NaQEI     JNWhcty9NXJib25QdUUTM9ua     JAwdjik0MzRhOCISInR5cCIGIKJIY     JIkLNUwHmQtNGFKZIINMTJMLVZIY     MiloisidUIXZFIdGhvcmJ6YXRpb     H2C2GINVy5JYX3IONIONE     WIZZXILCZWcmWZXJyZNR4dXNI2     MILC314dHUZSIGIAI2CySbGVI     MxAN0wXGQJkavzhAprQFXMc2Ro     BoGMP5jGTVL0JggLsiNr6QL1zV     caljinTEWtosgNV_SAY85DkwqOlu     VVBzcXZhZXZINXdTY0NkQ3dMZDFQ     VVBzcXZhZXZINXdTY0NkQ3dMZDFQ	Save Response WIHVG92dEhjRWI m5i2i16MCwiaWF KNrcmlu2/IImF X11ci1sImF6cCI rfmHTU52DNHYII 44iXX8LTJ1C29 XH101i3UbFuYWa Tptb25pd69yaW5 m5hbW103122X3 m5hbW103122X3 m652ccHPcM39 MF52ccHPcM39 79k2dDxp2Q902c YWIHVG92dEhjRWI Tp532166via au

The following screenshot shows a post request in Postman (www.getpostman.com).

# 7.4.2 Best practice: Use query options in API – Monitoring calls

For best performance, use query options to reduce the number of records or properties returned in API calls. For example, if you only require specific information for some agents, use query options to specify the agents and properties instead of obtaining data for all agents.

You can use the following query options to reduce the number of records or properties in API calls:

- \$select Specifies a subset of properties to return.
- \$filter Specifies an expression that must be true for a record to be returned.
- \$top Specifies the maximum number of records to return.
- \$skip Specifies the number of records to skip before returning records.

For performance reasons, each API – Monitoring call can return a maximum of 500 records. To determine whether there are more than 500 results for a call, include the \$count query option. You can then use additional calls with the \$skip query option to return the remaining results. For example, if the \$count query option indicates that there are 1422 agents in your environment, you could use one call to obtain

the first 500 agent records, a second call with the \$skip query option to skip the first 500 agents, and a third call with the \$skip query option to skip the first 1000 agents.

For more information about OData query options, see the OData website: http://www.odata.org

#### Examples

To obtain status information for agents from a specific company, you could use the following call:

GET

APIdomainNameOrIPaddress/monitoring/agents?\$select=name,operatingSystem,stat us&\$filter=companyId eq 980abfd9-5ddd-4bf3-b9f2-cc8bbd10ce0e

Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of <u>Install API –</u> <u>Monitoring</u>.

To return only the top 50 agent records and determine how many results there are for the call, use the following call:

GET APIdomainNameOrlPaddress/monitoring/agents?\$top=50&\$count=true

To skip the first 50 agents and return the next 50 agent records, use the following call:

GET APIdomainNameOrlPaddress/monitoring/agents?\$top=50&\$skip=50

To skip the first 500 agents and return agents 501 to 550, use the following call:

GET APIdomainNameOrlPaddress/monitoring/agents?\$top=50&\$skip=500

# 8 Set up data deletion

As described in <u>Data deletion overview</u>, when Portal and Director vaults are registered to the same API – Monitoring instance, Director can delete job and computer data in response to requests from Portal.

After API – Monitoring is installed and Director Reporting services are registered to the API (see <u>Install API –</u> <u>Monitoring</u> and <u>Register Director Reporting services to the API</u>), the following steps must be completed so that Director can delete vault data in response to requests from Portal:

- Portal must be registered to the same API Monitoring instance as Director Reporting services, and the data deletion feature must be enabled in Portal. See "Enable data deletion" in the *Portal Administration Guide* or online help.
- 2. A machine key must be added in Portal config files. See "Add a machine key required for data deletion" in the *Portal Installation Guide*.
- 3. Data deletion email notifications must be set up in Portal. See "Set up automatic emails" in the *Portal Installation Guide*.
- 4. The Data Deletion feature must be enabled in Portal. See "Enable data deletion" in the *Portal Administration Guide* or online help.