

# Integration with AWS


Prior to enabling the Wazuh rules for Amazon Web Services, follow the steps below to enable the AWS API to generate log messages and store them as JSON data files in an Amazon S3 bucket. A detailed description of each of the steps can be found below.

1. [Turn on CloudTrail](#).
2. [Create a user with permission to access S3](#).
3. [Install Python Boto on your Wazuh manager](#).
4. [Configure user credentials with Python Boto](#).
5. [Run the python script to download the JSON data](#).
6. [Collect AWS log data](#).

## Turn on CloudTrail


Create a trail for your AWS account. Trails can be created using the AWS CloudTrail console or the AWS Command Line Interface (AWS CLI). Both methods follow the same steps. In this case we will be focusing on the first one:

- Turn on CloudTrail. Note that, by default, when creating a trail in one region in the CloudTrail console, it will actually apply to all regions.

 **Warning**

Please do not enable *Enable log file validation* parameter; it's not supported by the provided python script.

- Create a new Amazon S3 bucket or specify an existing bucket to store all your log files. By default, log files from all AWS regions in your account will be stored in the bucket selected.


 **Note**

When naming a new bucket, if you get this error `Bucket already exists. Select a different bucket name.`, then try a different name, since the one you have selected is already in use by another Amazon AWS user.

From now on, all the events in your Amazon AWS account will be logged. You can search log messages manually inside `CloudTrail/API activity history`. Note that every 7 minutes, a JSON file containing new log messages will be stored in your bucket.


## Create a user with permission to access S3

Sign in to the `AWS Management Console` and open the [IAM console](#). In the navigation panel, choose `Users` and then choose `Create New Users`. Type the username for the user you would like to create.

 **Note**

Username can only use a combination of alphanumeric characters and these characters: plus (+), equal (=), comma (,), period (.), at (@), and hyphen (-). Names must be unique within an account.

The user will need access to the API, which requires an access key. To generate an access key for a new user, select `Generate an access key` and choose `Create`.

 **Warning**

This is your only opportunity to view or download the secret access key, and you must provide this information to your user before they can use the AWS Console. If you don't download and save it now, you will need to create new access keys for the user later. You will not have access to the secret access key again after this step.

Give the user access to this specific S3 bucket (based on [Writing IAM Policies: How to Grant Access to an Amazon S3 Bucket](#))

Under the IAM console, select `Users` and go to the `Permissions` tab, in the `Inline Policies` section, push the `Create User Policy` button. Click the `Custom Policy` option and push the `Select` button.

On the next page enter a `Policy Name` e.g. ossec-cloudtrail-s3-access, and for `Policy Document` use the example provided below:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3:::YOURBUCKETNAME"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::YOURBUCKETNAME/*"]
    }
  ]
}
```

## Install Python Boto on your Wazuh manager

To download and process the Amazon AWS logs that already are archived in S3 Bucket we need to install Python Boto on the Wazuh manager and configure it to connect to AWS S3.

Prerequisites for Python Boto installation using Pip

- Windows, Linux, OS X, or Unix
- Python 2 version 2.7+ or Python 3 version 3.3+
- Pip

Check if Python is already installed:

```
$ python --version
```

If Python 2.7 or later is not installed, then install it with your distribution's package manager as shown below:

- On Debian derivatives such as Ubuntu, use APT:

```
$ sudo apt-get install python2.7
```

- On Red Hat and derivatives, use yum:

```
$ sudo yum install python27
```

Open a command prompt or shell and run the following command to verify that Python has been installed correctly:

```
$ python --version
Python 2.7.9
```

To install Pip on Linux:

- Download the installation script from [pypi.io](https://pypi.io):

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

- Run the script with Python:

```
$ sudo python get-pip.py
```

Now that Python and pip are installed, use pip to install boto:

```
$ sudo pip install boto
```

## Configure user credentials with Python Boto

It is necessary to configure the AWS CLI on your Wazuh manager to use the credentials of the newly created user. Create a file called `/etc/boto.cfg` like this:

```
[Credentials]
aws_access_key_id = <your_access_key_here>
aws_secret_access_key = <your_secret_key_here>
```

## Run the python script to download the JSON data

We use a python script to download JSON files from the S3 bucket and convert them into flat files that can be used with Wazuh. This script was written by Xavier Martens *@xme* <<https://blog.rootshell.be>> and contains minor modifications done by Wazuh. It is located in our [repository](#) at `wazuh/wazuh-ruleset/tools/amazon/getawslog.py`.

Run the following command to use this script:

```
$ ./getawslog.py -b s3bucketname -d -j -D -l /path-with-write-permission/amazon.log
```

Where `s3bucketname` is the name of the bucket created when CloudTrail was activated (see the first step in this section: “Turn on CloudTrail”) and `/path-with-write-permission/amazon.log` is the path where the flat log file is stored once has been converted by the script.

### Note

If you don't want to use an existing folder, create a new one manually before running the script.

### Warning

The above script will delete all logs on the Amazon S3 bucket after download.

if you want to maintain the logs files in the bucket, you need to use the script without `-D` parameter like the following example:

```
$ ./getawslog.py -b s3bucketname -d -j -l /path-with-write-permission/amazon.log -s /path-with-write-permission/awslogstat.db
```

Using `-s /path-with-write-permission/awslogstat.db` will track downloaded log files avoiding processing them again, without it the script will download previously processed log files adding its content again to `/path-with-write-permission/amazon.log`. Also you need to install `sqlite` module for python:

```
$ sudo pip install pysqlite
```

CloudTrail delivers log files to your S3 bucket approximately every 7 minutes. Create a cron job to periodically run the script. Note that running it more frequently than once every 7 minutes would be useless. CloudTrail does not deliver log files if no API calls are made on your account.

Run `crontab -e` and, at the end of the file, add the following line

```
*/5 * * * * /usr/bin/flock -n /tmp/cron.lock -c python path_to_script/getawslog.py -b s3bucketname -d -j -  
D -l /path-with-write-permission/amazon.log
```

### Note

This script downloads and deletes the files from your S3 Bucket. However, you can always review the log messages generated during the last 7 days within the CloudTrail console.

## Collect AWS log data

Now the Wazuh manager needs to be configured to be able to collect the log messages generated by AWS. In other words, the file `/path-with-write-permission/amazon.log` generated by the script mentioned above needs to be added to the configuration file `/var/ossec/etc/ossec.conf` using the `<ossec_config>` tag as shown below.

```
<ossec_config>  
  <localfile>  
    <log_format>syslog</log_format>  
    <location>/path-with-write-permission/amazon.log</location>  
  </localfile>  
</ossec_config>
```

### Note

The file `/path-with-write-permission/amazon.log` must be the same one you setup in the above step: [Run the python script to download the JSON data](#).

Finally, restart the Wazuh manager to apply changes:

a. For Systemd:

```
systemctl restart wazuh-manager
```

b. For SysV Init:

```
service wazuh-manager restart
```