# Configuring OpenStack with GCS Backup Driver

## Configuration Guide

Version 1.4

24 Dec 2015

BIARCA

## PREFACE

This Configuration Guide walks the Administration team through the procedures to be followed for configuring Google cinder backup driver in the OpenStack Environment. It details the recommended guidelines to be followed for deploying and using the Google cinder backup driver.

## COPYRIGHT



# Table of Contents

## Introduction

Google Cinder Backup Driver is designed to support backup and restore functionalities. Some of its main operations are detailed below:

1. Cinder offers OpenStack tenants self-service backup and restore operations for their Cinder volumes.
2. These operations are performed on individual volumes.
3. A Cinder backup operation creates a point-in-time, read-only set of data and metadata that can be used to restore the contents of a single Cinder volume either to a new Cinder volume (the default) or to an existing Cinder volume.
4. Backups are stored in a dedicated repository(Google Cloud Storage repository), independent of the storage pool containing the original volume or the storage backend providing its block storage.
5. Cinder backup repositories must be implemented using the Google Cloud Storage.

# Configuring Google Cinder Backup Driver

To configure the Google Cinder Backup Driver, perform the following procedures in sequence:

- Check pre-requisites
- Install Google Cloud Storage (GCS) packages
- Setup Google service account
- Copy Google backup driver file to cinder backup drivers directory
- Edit cinder.conf file
- Updating Cinder exception file
- Restart cinder backup service

## Check Pre-requisites

Before you start configuring the Google Cinder Backup Driver, ensure to meet the following pre-requisites:

1. Deploy Openstack cloud with; cinder volume and cinder backup service installed on the same server.
2. Google Cloud Storage (GCS) backup driver is compatible with any cinder volume backend such as LVM or any third party cinder volume vendors.
3. Install Openstack Liberty on Ubuntu 14.04.03 server.

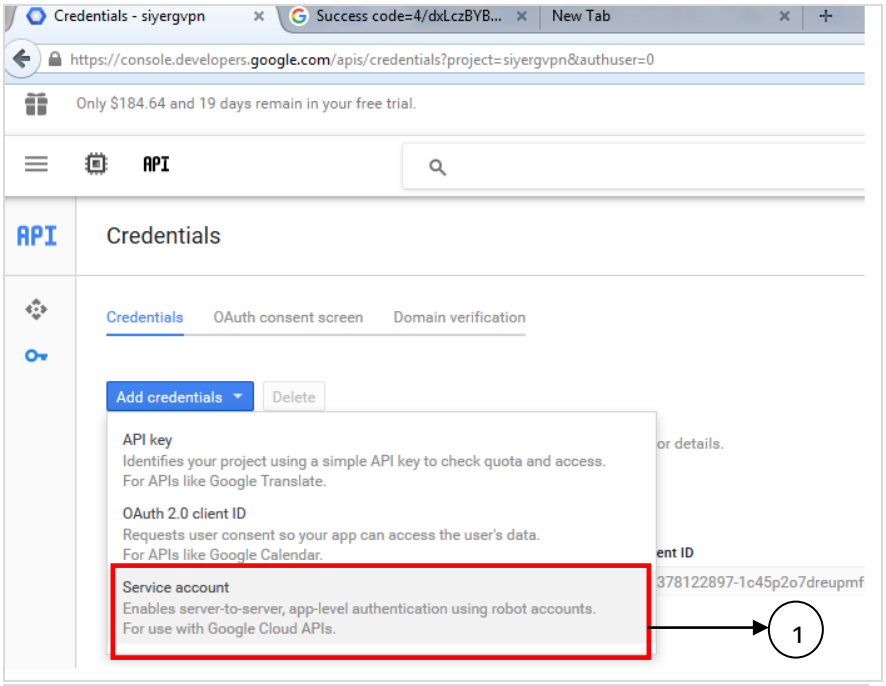
## Install Google Cloud Storage (GCS) Packages

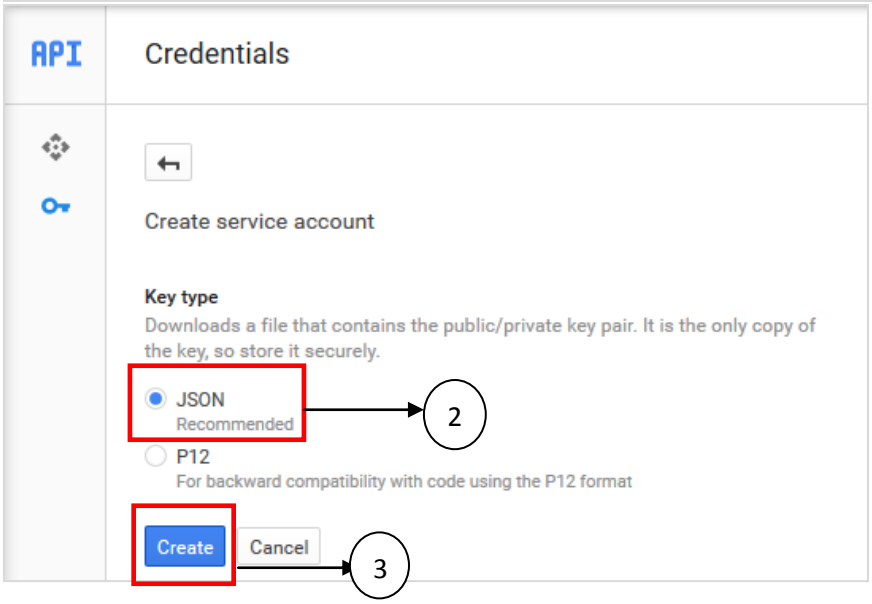
To install Google Cloud Storage python client libraries on the the Cinder Backup node, run the following command on that node:

```
sudo pip install --upgrade google-api-python-client
```

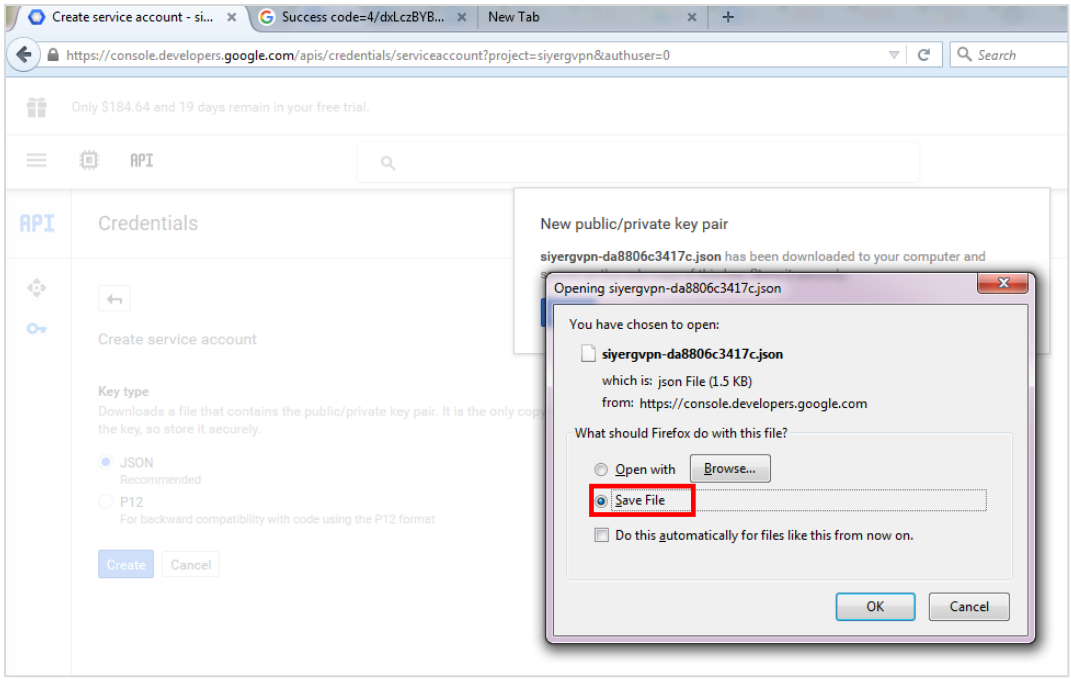
## Setup Google Service Account

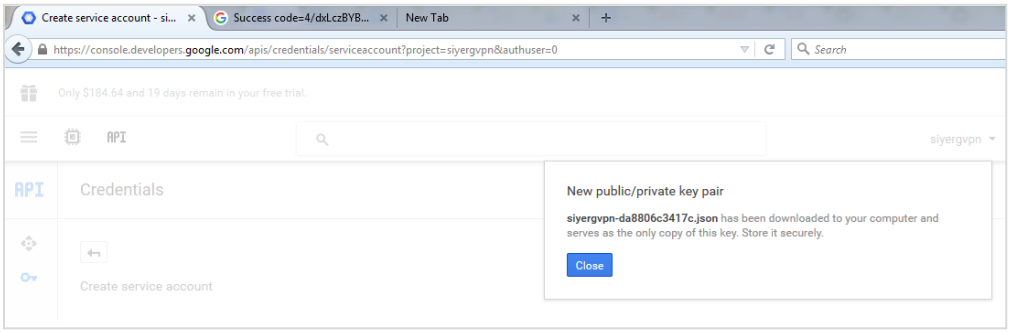
1. Create a service account through the Google developer console by following the steps highlighted below:




2. Save the `json` file on the Cinder Backup Server node:




## Copy Google backup driver file to Cinder Backup Drivers directory

Copy the `google.py` file from Git-hub repo of `biarca/google` and place it to the following location on the cinder backup server node:

```
cinder/backup/drivers
```

## Edit cinder.conf file

Modify the `cinder.conf [DEFAULT]` section, as shown below:

```
# edit /etc/cinder/cinder.conf
```

```
[DEFAULT]
os_privileged_user_tenant = service
os_privileged_user_password = Biarc8123
os_privileged_user_name = nova
glance_api_servers = http://192.168.2.157:9292
osapi_volume_workers = 2
logging_context_format_string = %(asctime)s.%(msecs)03d %(levelname)s %(
(message)s
volume_clear = none
rpc_backend = rabbit
backup_swift_url = http://192.168.2.157:8080/v1/AUTH_
default_volume_type = lvmdriver-1
enabled_backends = lvmdriver-1
os_region_name = RegionOne
enable_v1_api = true
periodic_interval = 10
state_path = /opt/stack/data/cinder
osapi_volume_listen = 0.0.0.0
osapi_volume_extension = cinder.api.contrib.standard_extensions
rootwrap_config = /etc/cinder/rootwrap.conf
api_paste_config = /etc/cinder/api-paste.ini
iscsi_helper = tgtadm
verbose = True
debug = True
auth_strategy = keystone
nova_catalog_admin_info = compute:nova:adminURL
nova_catalog_info = compute:nova:publicURL

backup_gcs_credential_file = "/home/biarca/gcscinder-0bea0f6844ab.json"
backup_gcs_bucket = "gcscinderbucket"

backup_driver = cinder.backup.drivers.google
backup_gcs_project_id = "gcscinder"
backup_gcs_user_agent = "biarca"
```

The following table details the main configurable GCS options in the `cinder.conf` file:

| Parameter | Purpose |
|---|---|
| `backup_gcs_credential_file` | Denotes the full path of the `json` file of Google service account (downloaded from the Google developer console). |
| `backup_gcs_bucket` | GCS bucket name to use for backup. Please refer "https://cloud.google.com/storage/docs/bucket-naming#requirements" for bucket naming guidelines. |
| `backup_driver` | Used for selecting Google backup driver |

The following table details the other configurable GCS options in the `cinder.conf` file:

| Parameter | Purpose |
|---|---|
| `backup_gcs_object_size` | The size in bytes of GCS backup objects.<br>default: 52428800 Bytes |
| `backup_gcs_block_size` | The size in bytes that changes are tracked for incremental backups.<br>`backup_gcs_object_size` has to be multiple of `backup_gcs_block_size`.<br>default: 32768 Bytes |
| `backup_gcs_project_id` | Denotes the project ID required for creating a GCS bucket. |
| `backup_gcs_user_agent` | Http user-agent string for gcs api. |
| `backup_gcs_reader_chunk_size` | GCS object will be downloaded in chunks of bytes.<br>default=2097152 bytes |
| `backup_gcs_writer_chunk_size` | GCS object will be uploaded in chunks of bytes.<br>Pass in a value of -1 if the file is to be uploaded as a single chunk.<br>default=2097152 bytes |
| `backup_gcs_num_retries` | Number of times to retry.<br>default=3 |
| `backup_gcs_bucket_location` | Location of GCS bucket.<br>default='US' |
| `backup_gcs_storage_class` | Storage class of GCS bucket.<br>default='NEARLINE' |

## Updating Cinder exception file

Update the cinder exception file with GCS cinder backup driver exceptions, as detailed below:

1. Go to the `cinder` directory.
2. Update the exception file, as shown below:

```
vi exception.py
```

```
# Google Cloud Storage(GCS) backup driver
class GCSConnectionFailure(BackupDriverException):
    message = _("Google Cloud Storage connection failure: %(reason)s")


class GCSApiFailure(BackupDriverException):
    message = _("Google Cloud Storage api failure: %(reason)s")


class GCSOAuth2Failure(BackupDriverException):
    message = _("Google Cloud Storage oauth2 failure: %(reason)s")
```

## Restart Cinder Backup Service

Once all the above steps are executed, restart the Cinder Backup Service to make the changes take effect.

## Create Backup

Create Backup of cinder volume data:

```
$ cinder list
+--------------------------------------+-----------+-----------------+---------------+------+-------------+----------+-------------+----------
--------------------------+
|                  ID                  |   Status  | Migration Status |      Name      | Size | Volume Type | Bootable | Multiattach |
  Attached to              |
+--------------------------------------+-----------+-----------------+---------------+------+-------------+----------+-------------+----------
--------------------------+

                           |
| cc95d405-1cdd-47b0-804b-bd0f38034bbd | available |        -        |    gcsdemo     |  2   | lvmdriver-1 |  false   |    False    |
+--------------------------------------+-----------+-----------------+---------------+------+-------------+----------+-------------+----------
--------------------------+
```

```
$ cinder backup-create cc95d405-1cdd-47b0-804b-bd0f38034bbd
```

## Restore Backup

Restore backed up data to a new volume or restore a volume to an old backup state:

```
$ cinder backup-restore --volume 1cf49ed6-26b0-4186-95f3-76922335f5d9 477673b6-cf3f-4edd-9746-6c7a6855eaef
```