Table of Contents

Prerequisites........................................................................................................................................2
Azure Configuration..............................................................................................................................3
  Setup Storage Accounts..................................................................................................................3
Provisioning Google..........................................................................................................................4
  Configuring the Agent....................................................................................................................5
Prerequisites

In order to make use of this cmdlet the firewall on the Azure SQL Server will need to be configured to "Allow Azure services and resources to access this server". If this is not configured then GatewayTimeout errors will be experienced.
Azure Configuration

Setup Storage Accounts

Choose LRS for lower costs:

After creation, using the menu on the left, create a container, call it staging or similar.
Also create a storage container called “restored” or similar
Provisioning Google

If you don't already have a Google cloud account for your organization begin at https://console.cloud.google.com/

After entering your billing information etc:

Create a storage bucket in cloud storage:

Be sure to pick a region away from your current storage location. For example if your main database is on the west coast, choose the east coast for your offsite backup location.

Choose coldline storage for lower costs while maintaining availability:
Choose a default storage class for your data

A storage class sets costs for storage, retrieval, and operations. Pick a default storage class based on how long you plan to store your data and how often it will be accessed. Learn more

- **Standard**
  - Best for short-term storage and frequently accessed data

- **Nearline**
  - Best for backups and data accessed less than once a month

- **Coldline**
  - Best for disaster recovery and data accessed less than once a quarter

- **Archive**
  - Best for long-term digital preservation of data accessed less than once a year

Configuring the Agent

Create a VM instance using Debian, again choose a location away from your databases in Azure:

**Name**
- **offsite-vm-instance**

**Labels**
- **ADD LABELS**

**Region**
- **us-east1 (South Carolina)**
  - Region is permanent

**Zone**
- **us-east1-b**
  - Zone is permanent

Machine configuration

**Machine family**

- **GENERAL-PURPOSE**
- **COMPUTE-OPTIMIZED**
- **MEMORY-OPTIMIZED**
- **GPU**

**Series**

- **E2**

**Machine type**

- **e2-medium (2 vCPU, 4 GB memory)**

  - **vCPU**
    - 1 shared core
  - **Memory**
    - 4 GB
Disable the public IP, then start your new VM instance and SSH into it via the SSH button to the right on the instance:

Install Docker:  [https://docs.docker.com/engine/install/debian/](https://docs.docker.com/engine/install/debian/)

Add your runtime user to the docker group:
sudo usermod -a -G docker $USER
su -l $USER

Install SecuritasMachina docker image:
docker pull securitasmachina2022/securitasmachinaoffsiteagent:latest

Create a prod.env text file, fill it in with your values:
customerAgentAuthKey
azureBlobEndpoint
azureSourceBlobContainerName
azureBlobRestoreContainerName
googleStorageBucketName
encryptionPassPhrase
retentionDays
maxThreads
GOOGLE_APPLICATION_CREDENTIALS

Add your ~/agent1Credentials.json to the Docker users home directory by first creating a service account with a private key in json format:
Assign the service account Creator and Viewer:

Run the container passing in your home directory (or directory containing json file):

docker run -it -d --env-file=./prod.env -v ~:/mnt/localhome
securitasmachina2022/securitasmachinaoffsiteagent

Note we suggest running interactively the first time to validate your configuration:

docker run -it --env-file=./prod.env -v ~:/mnt/localhome
securitasmachina2022/securitasmachinaoffsiteagent