Table of Contents

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

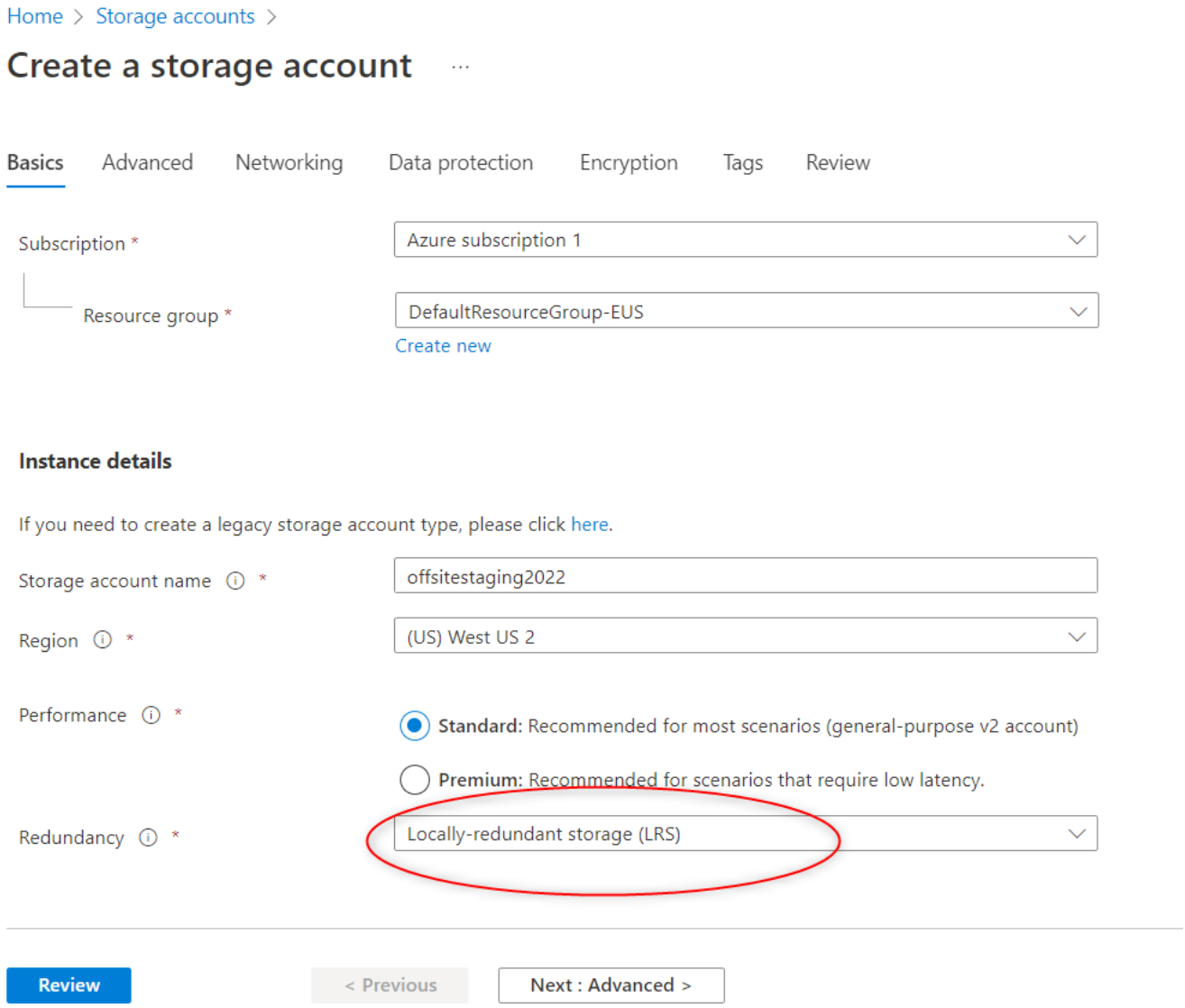
Prerequisites

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.

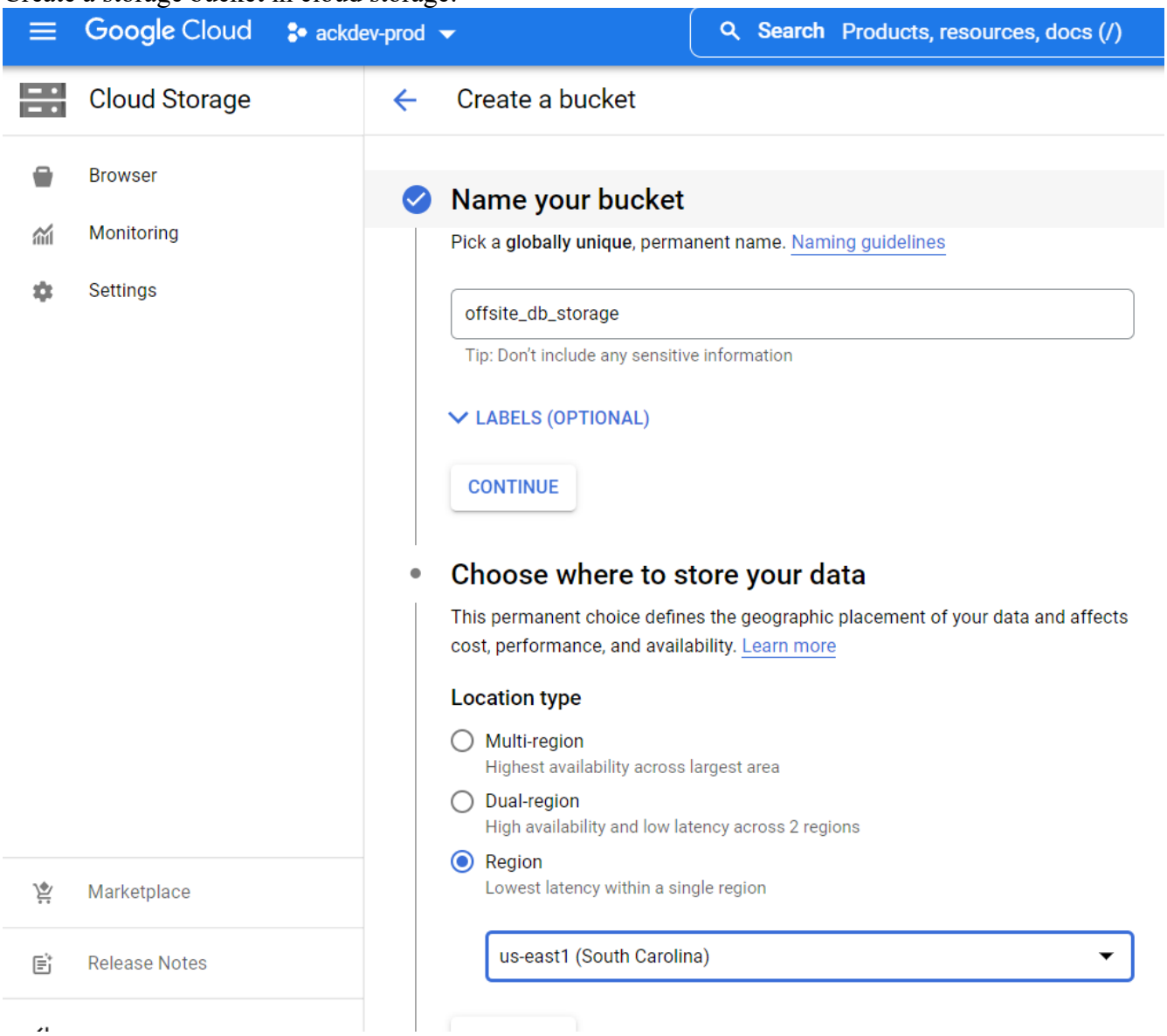
Also create a storage container called “restored” or similar

Provisioning Google

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

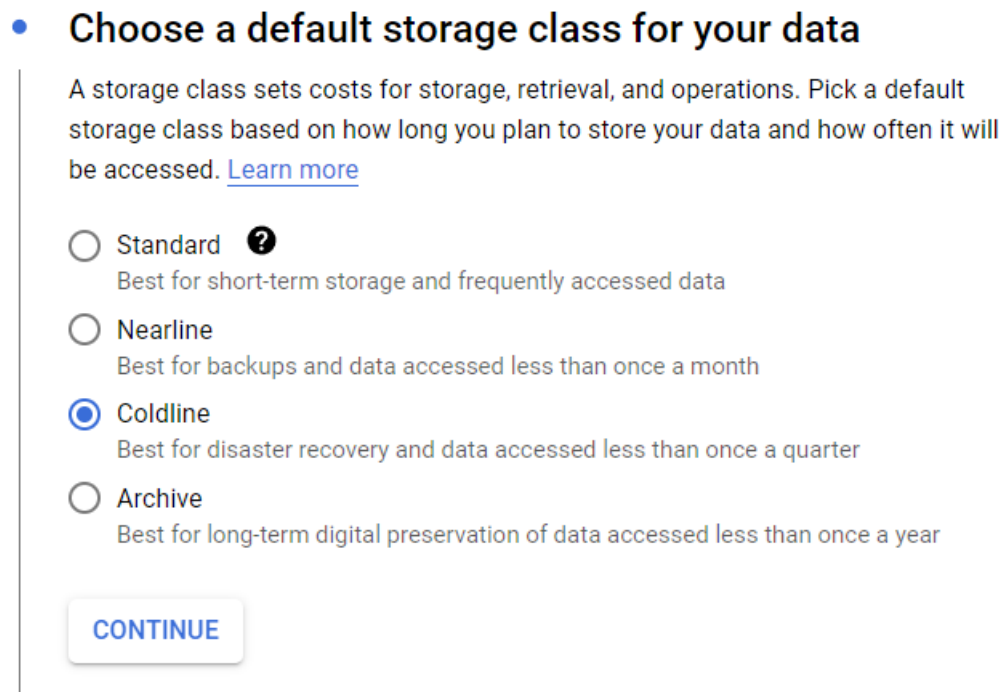
After entering your billing information etc:

Create a storage bucket in cloud storage:



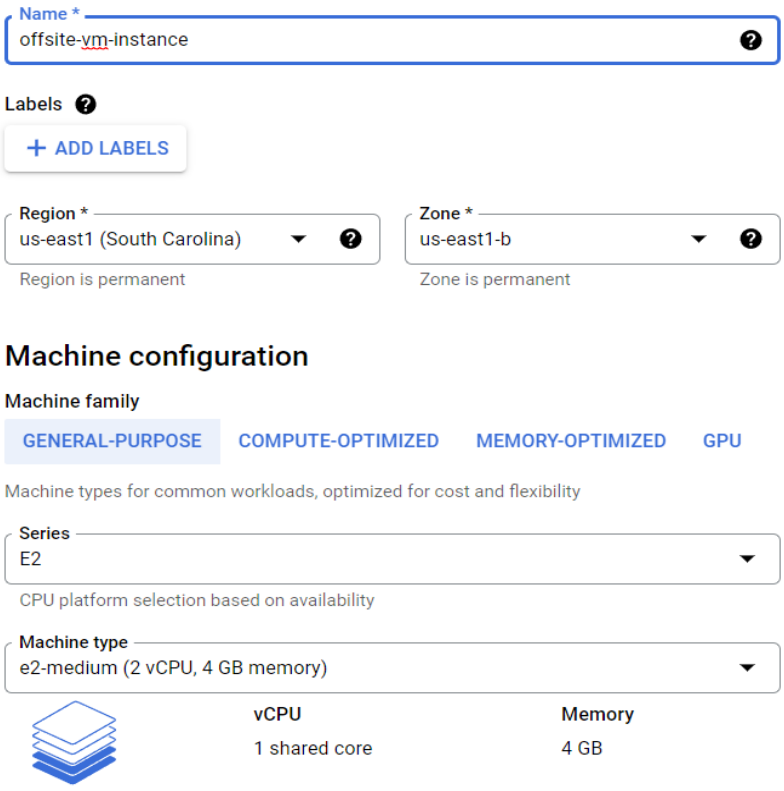
Be sure to pick a region away from your current storage location. For example if your main database is on the west coast, choose the east coast for your offsite backup location.

Choose coldline storage for lower costs while maintaining availability:

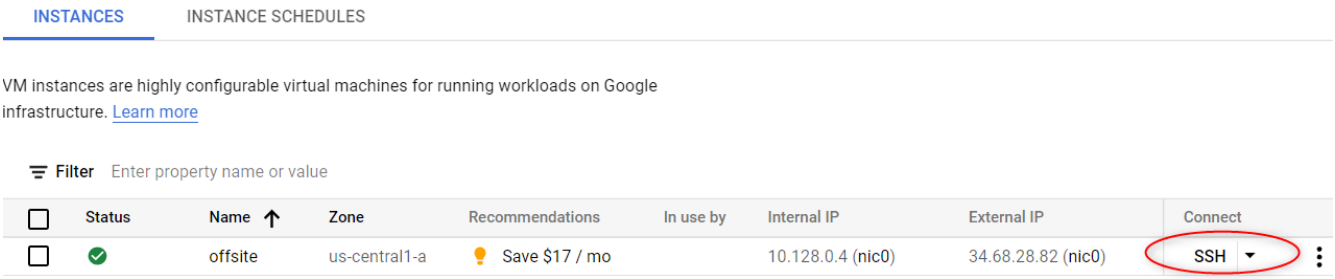


Configuring the Agent

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



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/

Add your runtime user to the docker group:

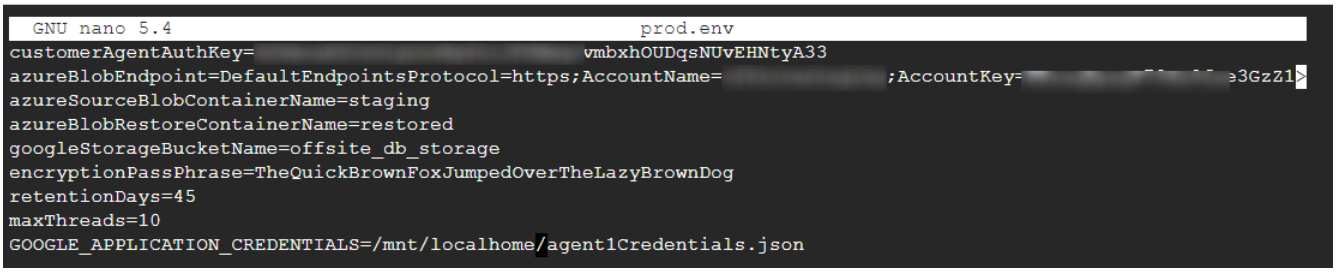
```
sudo usermod -a -G docker $USER
su -l $USER
```

Install SecuritasMachina docker image:

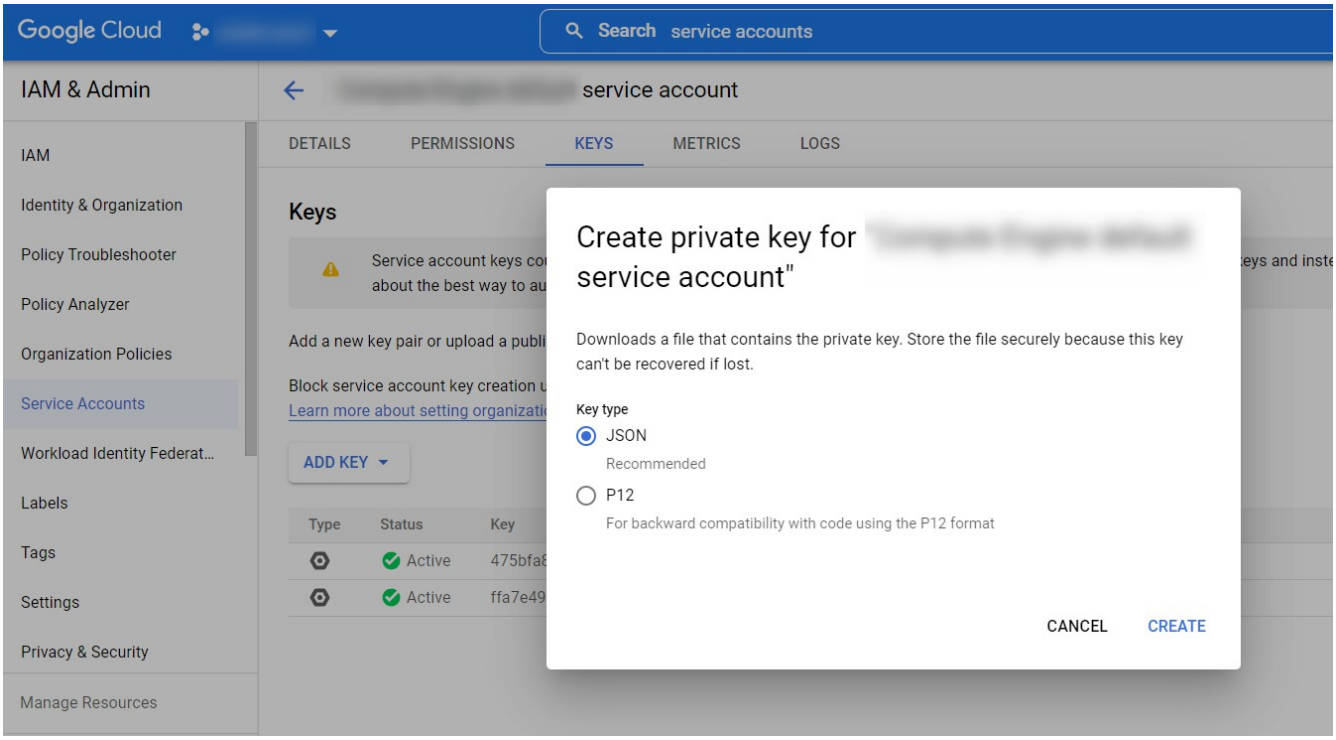
```
docker pull securitasmachina2022/securitasmachinaoffsiteagent:latest
```

Create a prod.env text file, fill it in with your values:

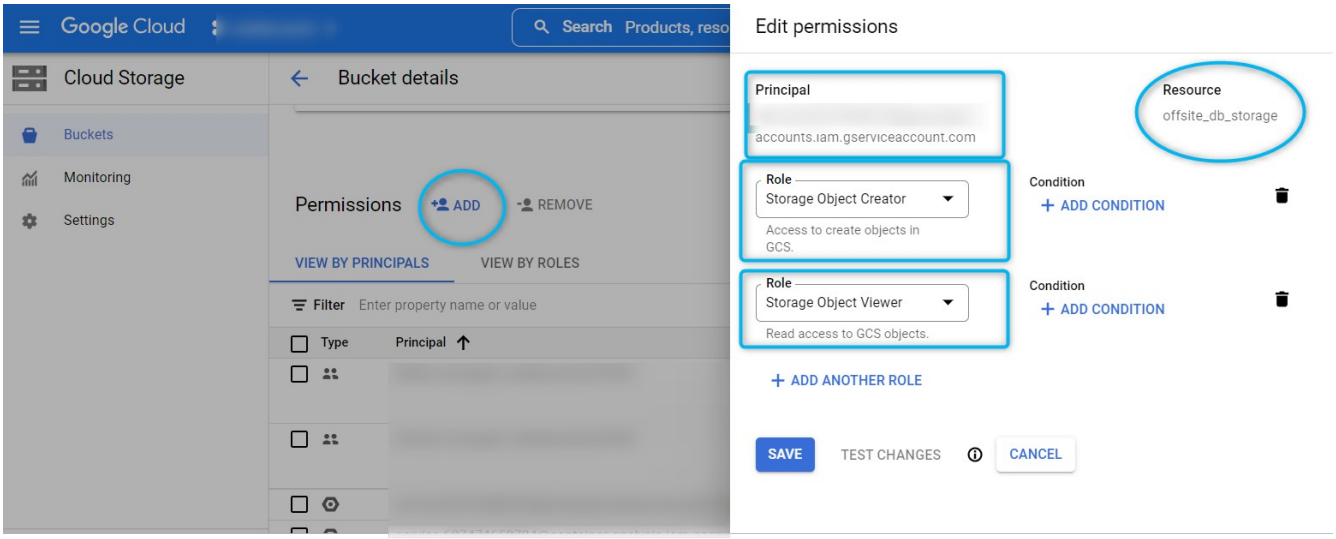
```
customerAgentAuthKey
azureBlobEndpoint
azureSourceBlobContainerName
azureBlobRestoreContainerName
googleStorageBucketName
encryptionPassPhrase
retentionDays
maxThreads
GOOGLE_APPLICATION_CREDENTIALS
```



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


Google Cloud
Search service accounts
IAM & Admin
service account
IAM
Identity & Organization
Policy Troubleshooter
Policy Analyzer
Organization Policies
Service Accounts
Workload Identity Federat...
Labels
Tags
Settings
Privacy & Security
Manage Resources
DETAILS
PERMISSIONS
KEYS
METRICS
LOGS
Keys
Service account keys co
about the best way to au
Add a new key pair or upload a publi
Block service account key creation u
Learn more about setting organizati
ADD KEY
Type
Status
Key
Active
Active
Create private key for "Compute Engine default service account"
Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.
Key type
JSON
Recommended
P12
For backward compatibility with code using the P12 format
CANCEL
CREATE

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
```