Google Cloud Platform

How to Lift-and-Shift a Line of Business Application onto Google Cloud Platform

by Andy Wu, Solutions Architect, Magenic

Table of Contents

Scenario Description **3**

Background Information 3

Project Approach 3

Current On-Premises System Architecture 4

Final Targeted On-Cloud System Architecture 4

The How-Tos **5**

Phase One Implementation 5

Creating a GCP network 5

Creating the Site-to-Site VPN 5

Creating the VMs to Support the On-Cloud System Architecture 6

Code Deployment 9

Encryption of Sensitive Data as part of Deployment Process 10

Phase Two **10**

Creating the SQL Server AlwaysOn Availability Groups 10

Phase Three **11**

Setting up the AD Replication for Redundancy in the Cloud 11

On the on-premises DC 12

On the GCP AD DC 16

Final Thoughts and Conclusion **21**

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

GitHub Source Url: https://github.com/Magenic/GCELiftAndShift

Scenario Description

Magenicons, a fictional comic book publishing company, has decided that its IT infrastructure needs to go through a modernization effort to increase system reliability while providing cost savings. Its IT staff believes leveraging cloud computing will help the company gain better agility for its IT infrastructure and applications. As part of an evaluation process, the company has selected an existing intranet-based expense reporting application as the proof-of-concept for its cloud migration strategy.

Background Information

The expense reporting application is a standard two-tier web-based application that currently relies on an on-premises Microsoft Internet Information Server (IIS) server with data storage on a separate on-premises Microsoft SQL Server. A second on-premises IIS server also provides auditing services. Access to the application is secured by authenticating against an on-premises Active Directory (AD) instance while data access is secured by using SQL Server Authentication with an application service account.

Project Approach

In order to minimize risk while gradually ramping up its teams' cloud knowledge and experience, Magenicons wants to execute the project in phases, with each phase having a defined objective to achieve. These objectives will be used at the end of the project for evaluation of the long-term viability of cloud computing for the company. Google Cloud Platform (GCP) was selected as the cloud provider due to the robust capabilities of the platform and Google's excellent technical reputation.

- Phase one – Migrate the application to the cloud using Google's Infrastructure-as-a-Service (IaaS) offering, Google Compute Engine.
    - Objective: ***Lift-and-shift*** the expense reporting application by leveraging Compute Engine with minimal cost. In particular, the company would like to execute the move with minimal to no code change to the existing application.
    - Prerequisite: Environment setup, such as establishing a network connection between Magenicons' local network and GCP, will be required in this phase to support the lift-and-shift of the application.

- Phase two – Leverage the cloud for high availability (HA).
    - Objective: Once the application is properly operating in the cloud, Magenicons would like to reduce the risk of potential downtimes by adding high availability to SQL Server used by the application. AlwaysOn Availability Groups is SQL Server's recommended solution, allowing users to configure replicas for automatic failover in case of failure. GCP supports Windows Server Failover Clustering (WSFC) and SQL Server AlwaysOn Availability Groups.

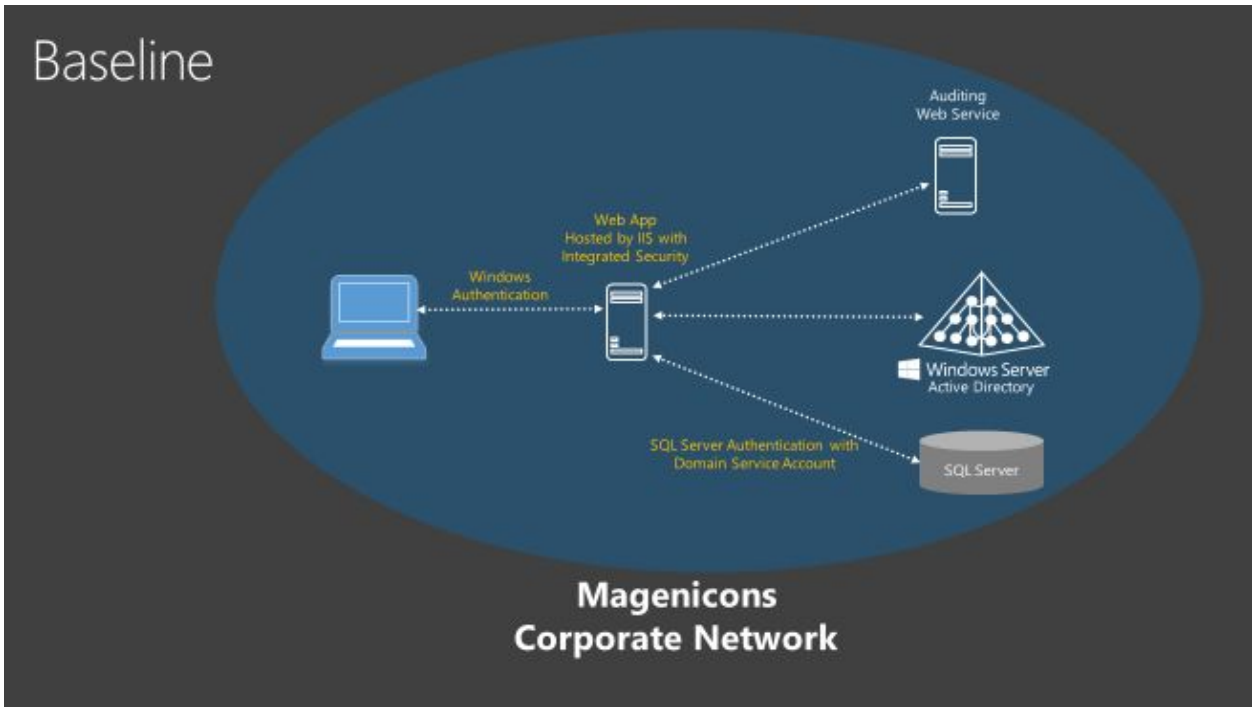
- Phase three – Leverage the cloud for disaster recovery (DR)
    - Objective: Magenicons would like to then further enhance application availability and

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

improve its DR plan by extending their on-premises AD into the cloud. This provides a cost-effective option for protecting AD in DR scenarios. In the event of a physical disaster or outage at company's data center, a virtual machine (VM) running as an Active Directory Domain Controller (AD DC) in GCP can provide uninterrupted access to AD for cloud-based applications and any on-premises AD-integrated applications unaffected by the outage. As an added benefit, having an AD hosted in the cloud alongside the application will generally shorten the network latency and thus improve system response time.

Current On-Premises System Architecture

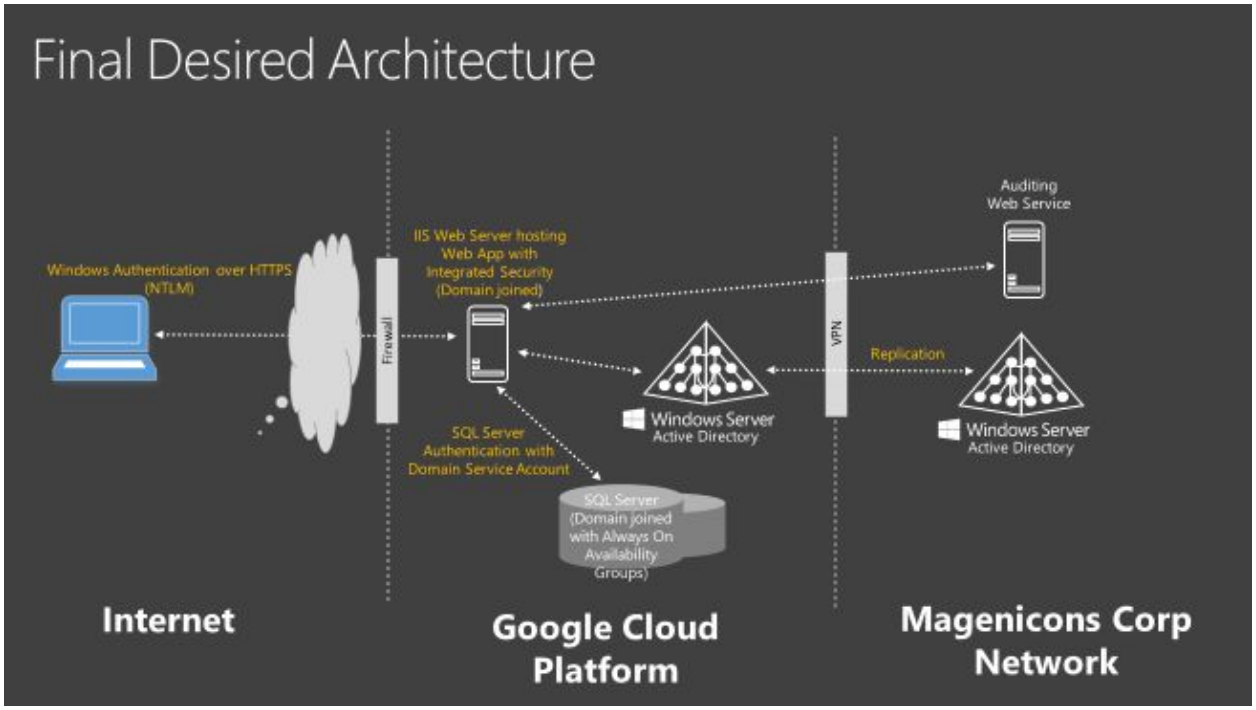
The expense report system uses a standard ASP.NET MVC application architecture for an intranet environment. The application is deployed onto an IIS webserver hosted in Windows Server and joined to the AD domain. The system is secured by leveraging Windows Integrated Security for all access to the application. Connection to SQL Server is also quite standard by using SQL Server Authentication with a domain service account user id and password.



Final Targeted On-Cloud System Architecture

The final targeted system architecture should look similar to the original on-premises system architecture, as it is treating GCP as an extension to the on-premises data center via a virtual private network (VPN) with additional features for SQL Server HA and DR for AD.

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.



The How-Tos

Phase One Implementation

For the phase-one objective of lifting-and-shifting the expense reporting application to the cloud, three major tasks were identified as requirements:

1. Create a GCP network suitable for the project
2. Create a VPN from the Magenicons corporate network to GCP
3. Create the VM instances that are necessary to support the application
4. Make any necessary configuration and or code changes to support the lift-and-shift

Creating a GCP network

GCP networks connect VM instances to each other and to the Internet, allowing users to segment their networks, create firewall rules for access control as well as create static routes to forward traffic to specific destinations. All of these capabilities will be needed as the project moves along its various phases. A tutorial on the particulars of GCP networking can be found here.

Important Note: Any type of supported subnet network mode (auto or custom) can be used to achieve phase oneôs objectives. However, as detailed in phase two below, in order to install SQL Server AlwaysOn Availability Groups a custom subnet must be used. Therefore, if one has the desire to eventually install this feature, it is highly recommended that a custom subnet be created for the project from the beginning to avoid any unnecessary rework down the road.

Creating the Site-to-Site VPN

Creating the VPN was a straightforward exercise and the project team did not run into any issues of

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

note. They simply followed the Google documentation and the VPN was up and running within a day.

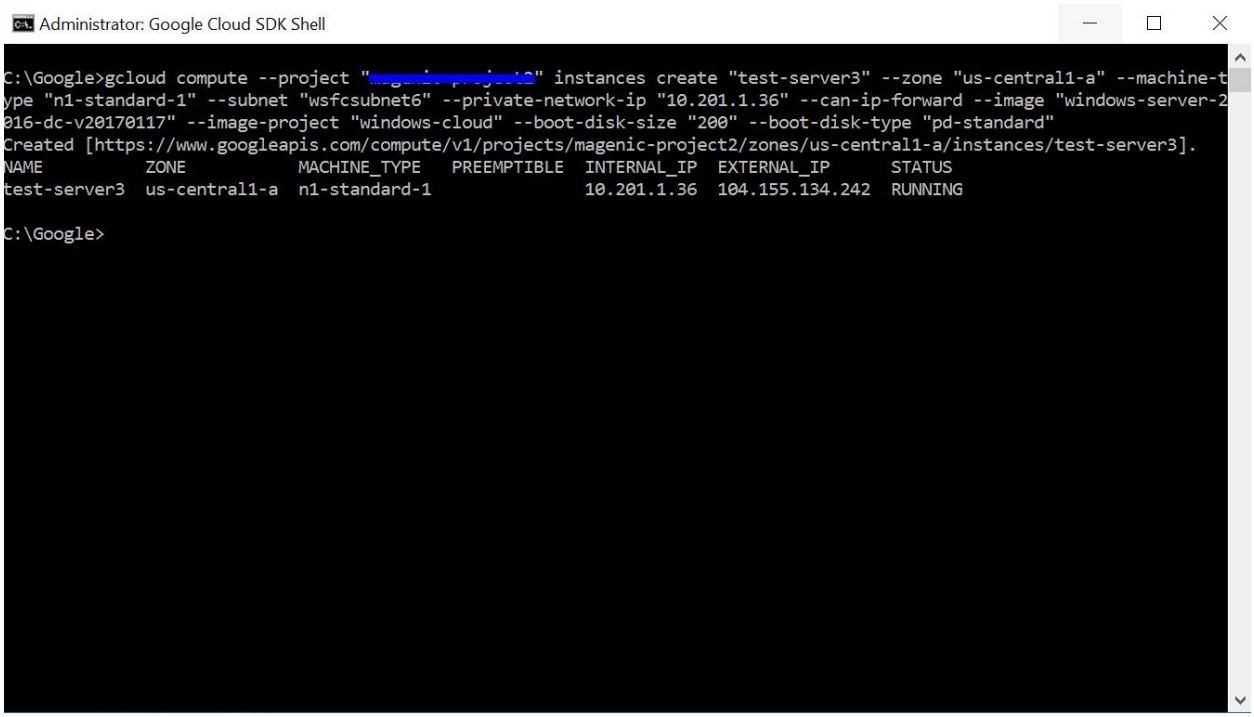
Creating the VMs to Support the On-Cloud System Architecture

Creating VM instances in Compute Engine can be done in three ways:

1. The point-and-click interface: Google Cloud Console in the subscriber's portal
2. REST API
3. Command Line Interface (CLI), which in Google's case is called the ***gcloud*** command line interface

Since the team understands that automation is a key ingredient to long term sustainability for cloud computing, it decided that a code-based approach using the gcloud command line interface would be the preferred choice.

Using the CLI is quite simple—just download the gcloud command line interface from Google and follow the documentation. Below is a screenshot of the experience when creating a Windows Server Instance:



Below are the VM instances and their roles needed for the various phases of the project:

- Phase 1
  - IIS Web Server
  - SQL Server
- Phase 2
  - Additional SQL Server instance used as part of SQL Server Availability Group replica
- Phase 3
  - An AD Domain Controller instance running in GCP with full replication to the

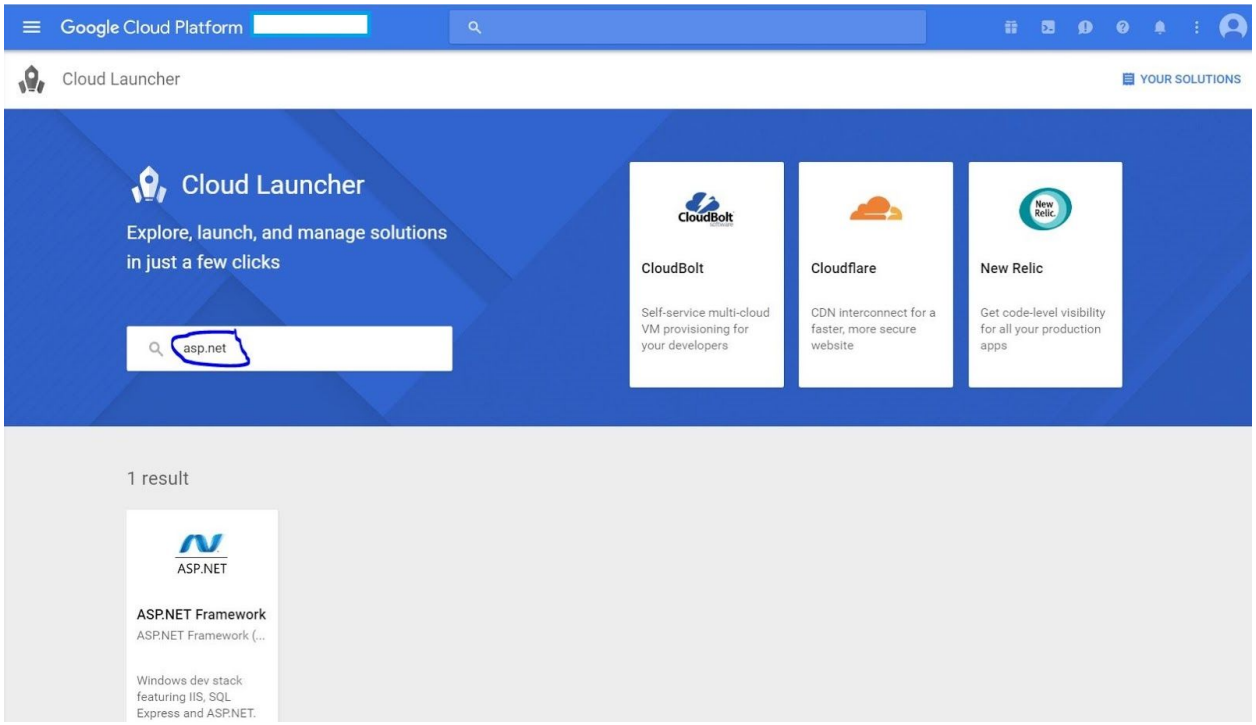
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

on-premises instance of AD Domain Controller

In addition to the above mentioned VM creation methods, GCP offers another great time-saving alternative for creating VM instances: Google Cloud Launcher. Google Cloud Launcher is a marketplace for third party ready-to-go development stacks, solutions and services. If the workload type fits what's available, then one can create the needed VM with simply a few clicks.

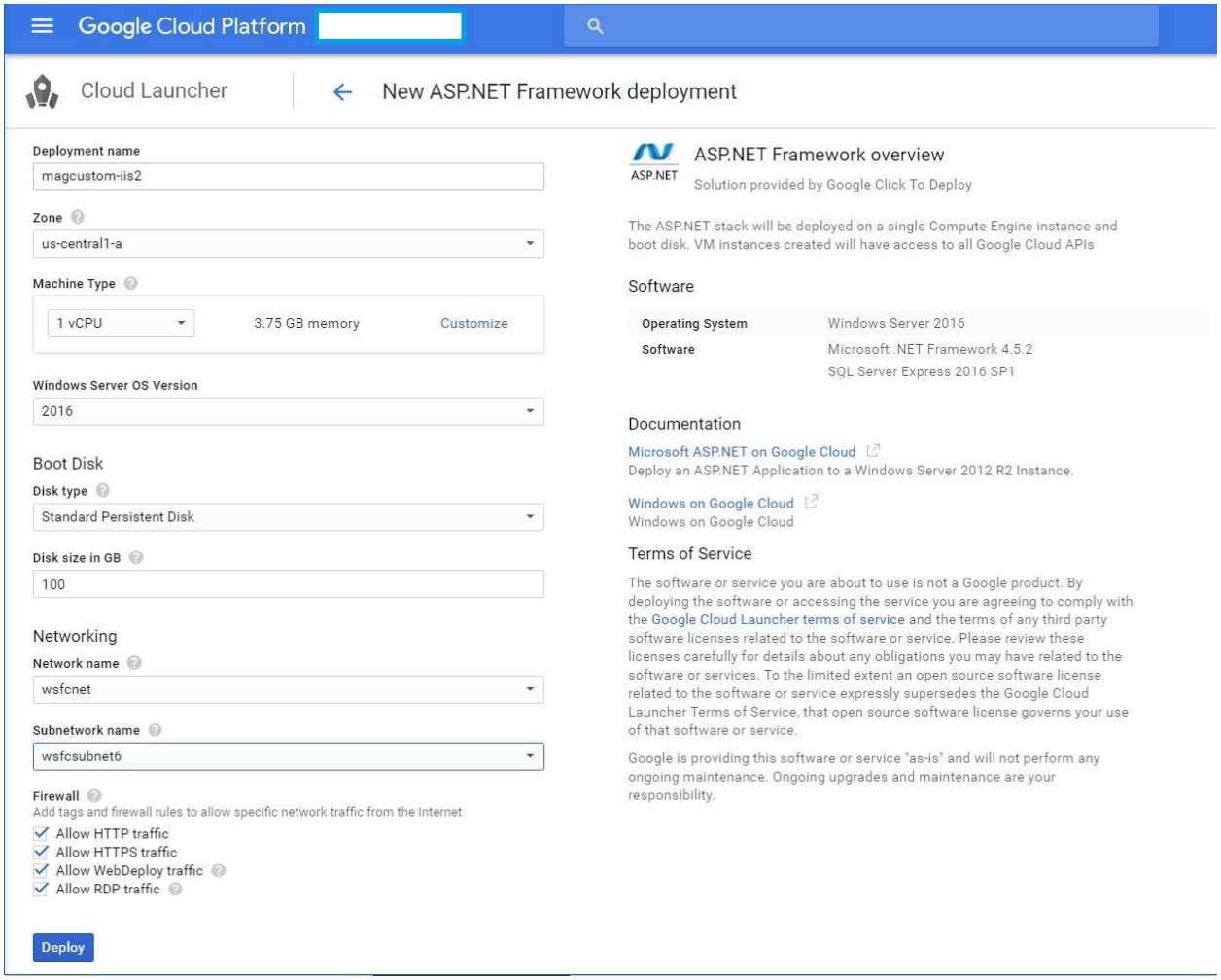
For Phase 1 development, the staff needed an IIS Server with ASP.NET 4.6 and its supporting .NET Framework installed. With the usual VM creation method, one would have to create the OS VM, and manually install all the various .NET Framework & ASP.NET component. With Cloud Launcher for ASP.NET, however, the process is dramatically simplified:

1. Go to the following url: https://console.cloud.google.com/launcher
2. Type in asp.net in the search input, and click on search result to initiate the Cloud Launcher process for ASP.NET Framework.



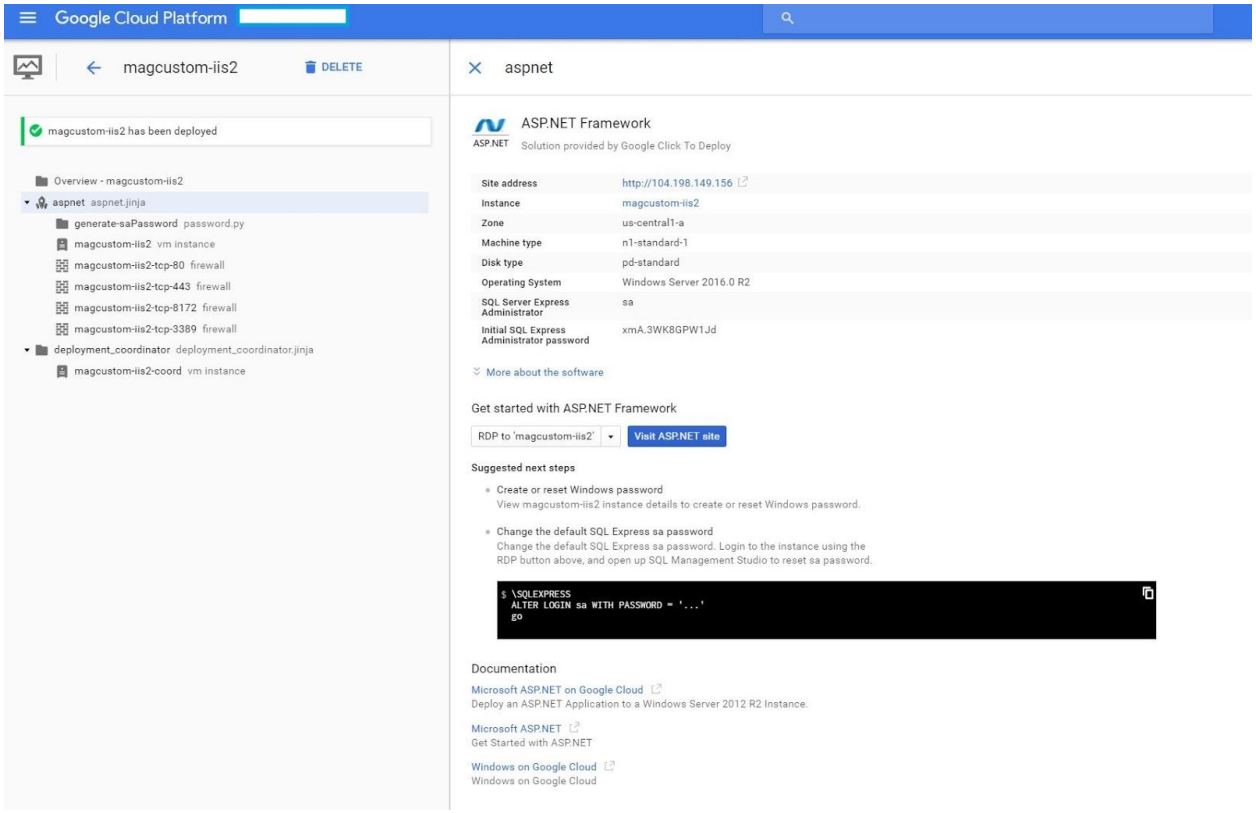
3. Fill in the deployment specifics such as machine name, disk, and cpu size:

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.



4. Within minutes, the desired VM, along with all the needed components, will be created:

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.



For the creation of the SQL Server instance, the team uses the following gcloud command line interface to create the image:

The gcloud command line interface (CLI) compute instances create "magcustom-sql1" --machine-type "n1-standard-4" --zone "us-central1-a" --subnet "wsfcsubnet1" --image-project windows-sql-cloud --image-family sql-ent-2016-win-2016--boot-disk-size "200" --boot-disk-type "pd-ssd” --private-network-ip=10.201.1.3 --can-ip-forward

Note: During phase 2 of the project, when the team is going to set up SQL Server HA, specific network routes will need to be created (see section on phase 2 for details). Therefore, this VM instanceôs internal IP address will need to conform to the network design described in phase 2. This is the reason the ó private-network-ipô parameter was used to specify a preferred internal IP address at instance creation time. If this parameter is used to specify a specific IP address, then one cannot change it to different static IP address afterwards without the risk of losing access to the instance. Use of the CLI to deploy Cloud Launcher solutions is subject to the Cloud Launcher Terms of Service and related fees.

Code Deployment

Once the IIS VM instance is provisioned and properly set up, the next task for the team is to deploy the code onto the instance. This is accomplished by leveraging one of Microsoft’s many offerings in this space: Web Deploy.

Web Deploy is a mature, extensible client-server tool for publishing website content between a

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

developer's or SysOps' workspace onto an IIS instance. The actual mechanism is well documented in the ASP.NET community and is out-of-scope for this document, but an overview of this technology can be found here.

Security best practices call for always encrypting sections of a configuration file which contain sensitive information, e.g., credentials or other secrets. This improves security by making it difficult for unauthorized access even if an attacker gains access to your configuration file. The same principle applies to this application.

The .NET Framework includes two protected built-in configuration providers that can be used to encrypt sections of a configuration file. The RsaProtectedConfigurationProvider class uses the RSACryptoServiceProvider to encrypt configuration sections. The DpapiProtectedConfigurationProvider class uses the Windows Data Protection API (DPAPI) to encrypt configuration sections. However, given the expense reporting application's required usage of integrated security and impersonation, RSACryptoServiceProvider is not a suitable choice as it would require granting access to the RSA Key Container used for encryption to a large group of users.

Encryption of Sensitive Data as part of Deployment Process

One of the major downsides of using the DpapiProtectedConfigurationProvider is the fact that it's not the default Configuration Provider used by Web Deploy, and therefore it is not able to automatically encrypt sensitive data as part of the deployment process. After a bit of research, the Magenicons development staff comes up with a solution that will be able to build, deploy and encrypt sensitive data (on the deployed server) with one single call to MSBuild. The solution calls for leveraging PowerShell's Invoke-command cmdlet, which has the ability to run commands on local or remote computers. Combining this capability with MSBuild's extensible feature of embedding scripts for various build and deployment events (in this case, after 'MSDeployPublish'), the team is able to optimize the build/deployment process while enhancing the security of the application.

Phase Two

Creating the SQL Server AlwaysOn Availability Groups

Enterprise SQL Server workloads require support for HA and DR. AlwaysOn Availability Groups is SQL Server's flagship HA/DR solution. This technology provides hot-standby for the servers and duplicate data for the database. AlwaysOn can also provide read-only access to one or more secondary replicas, alleviating load from the primary database in reporting and other read-only scenario.

For these reasons, Magenicons' IT staff selects this technology to achieve the project's HA requirement. Coincidentally, Google recently added support for SQL Server AlwaysOn Availability Group on Compute Engine.

In planning for the installation for AlwaysOn Availability Groups, there are several requirements one needs to pay special attention to.

1. At the current time, AlwaysOn Availability Groups can only be installed and supported in a GCP subnet network type. It **can not** be installed in a legacy network. Moreover, the subnet network must be in custom mode and not the default auto mode (details on the difference in

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

these network types and subnet modes can be found here).

2. Each node in the AlwaysOn Availability Group must reside on a different subnetwork, therefore one would need a minimum of two subnetworks for the setup.
3. Each database replica is hosted by an instance of SQL Server on a different node of the Windows Server Failover Cluster (WSFC) cluster.
4. To implement a two-node failover cluster, four IP addresses must be provisioned for the cluster itself as well as the Availability Group Listener. It's important to note that these designated IP addresses must fall outside of the actual subnetwork IP address range of the cluster nodes, but still be addressable with an appropriate subnet mask.

   Let's walk through a quick example:

   If a subnetwork is defined as 10.0.1.0/24, the VM's static IP and subnet mask are set up as 10.0.1.4 and 255.255.0.0 (/16). From the VM's perspective, the addressable subnet is 10.0.0.0/16. Therefore, one should pick an IP address such as 10.0.2.4 for the listener, which is outside the 10.0.1.0/24 subnetwork the VM resides in, but still addressable from the guest OS's perspective due to its wider subnet mask. One needs to apply this requirement for all the IP Addresses needed for WSFC and Availability Group Listener purposes.

   See table below for a sample network address scheme needed for entire setup. A step-by-step tutorial is also available here.

   Example network address scheme for the AlwaysOn Installation:

subnetworks	IP addresses ranges
wsfcsubnet1	10.201.1.32/29
wsfcsubnet2	10.202.1.32/29

Node Instances		subnetworks	WSFC		Availability Group Listener	
magcustom-sql 1	10.201.1.34	wsfcsubnet1	magcustom-wsfc	10.201.1.50	magcustom-as	10.201.1.51
magcustom-sql 2	10.202.1.34	wsfcsubnet2		10.202.1.51		10.202.1.51

5. Lastly, as noted in the step-by-step tutorial, network routes for the cluster and the availability group listener are needed in order for the listener and cluster to be able to reach the node instances. To create the routes, simply follow the example commands provided in the tutorial (modify them to fit one's own networking scheme as needed).

Phase Three

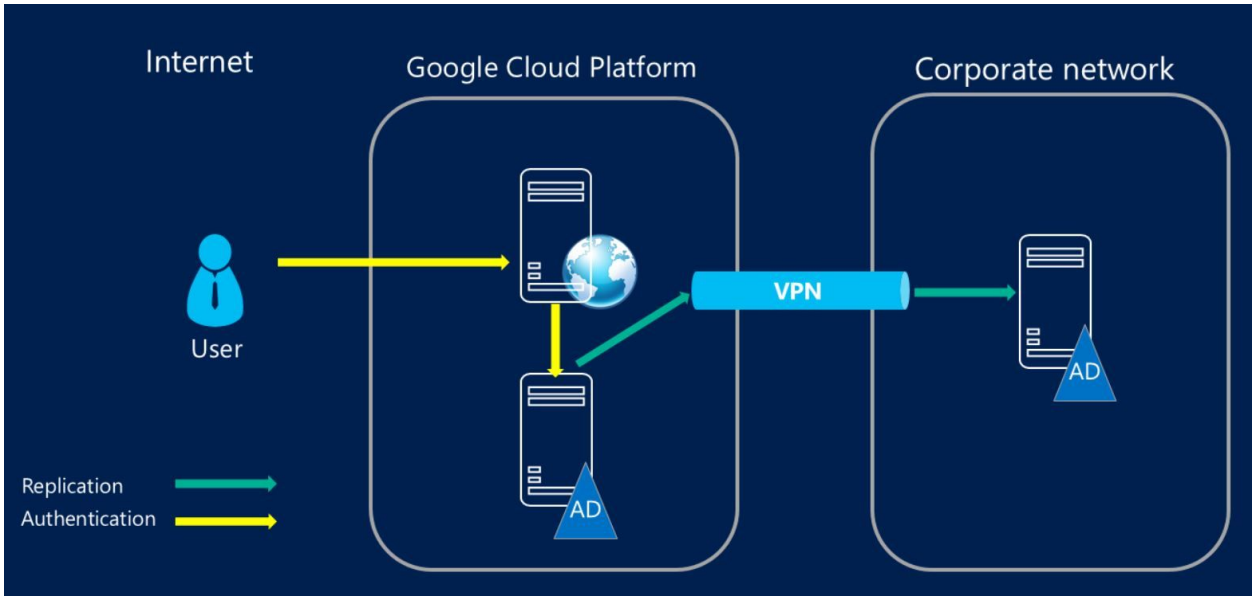
Setting up the AD Replication for Redundancy in the Cloud

As an underlying goal for the entire project, the network in GCP should be treated as an extension of

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

the Magenicons on-premises network so that applications can move from one site to the other seamlessly. Once the VPN connectivity between the two sites is set up, one can create an AD DC and Domain Name Server (DNS) in GCP as if it's just another branch office. Once this is set up and running, application(s) running in GCP will not have to traverse the internet for authentication and lookup purposes thus improving system performance.

The following diagram depicts the traffic flow:



The process for setting up an AD DC with Compute Engine is similar to other VM roles. The first task is to provision a VM instance by running a gcloud command::

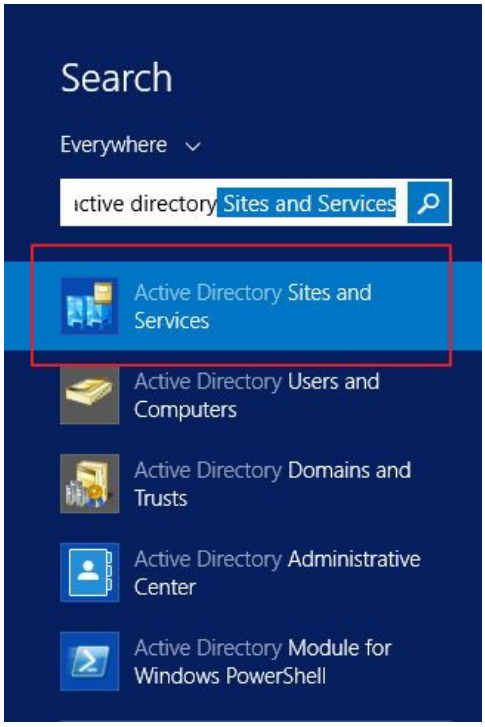
***gcloud compute instances create your-dc-machine-name --machine-type n1-standard-1 ***
***--boot-disk-type pd-ssd --image-project windows-cloud ***
***--image-family windows-2016 --boot-disk-size 200GB ***
--zone us-central1-a --subnet wsfcsubnet3 --private-network-ip=10.2.0.100

Once the VM is provisioned, the following tasks are performed in order to set up the site to site or inter-Site Domain replication:

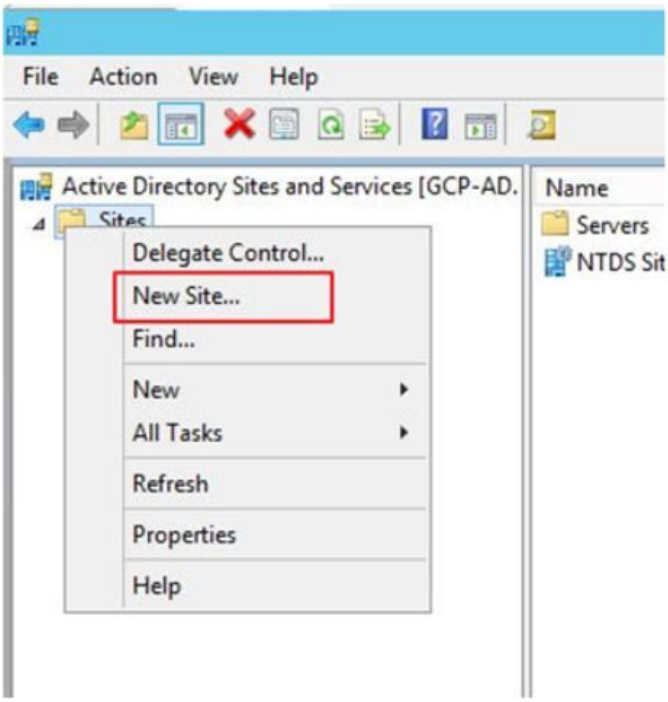
On the on-premises DC

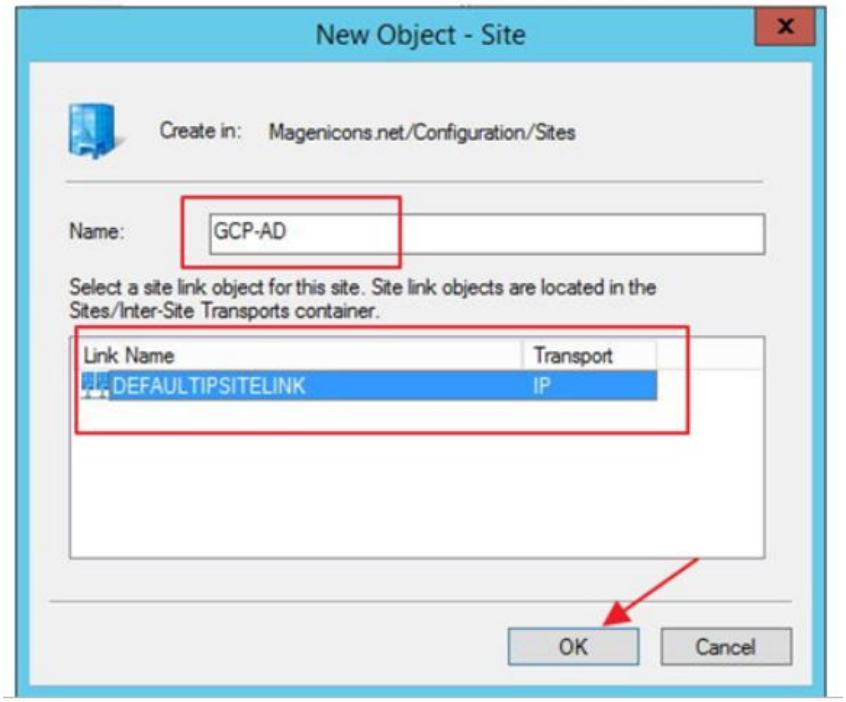
- Create a new site for GCP:

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.



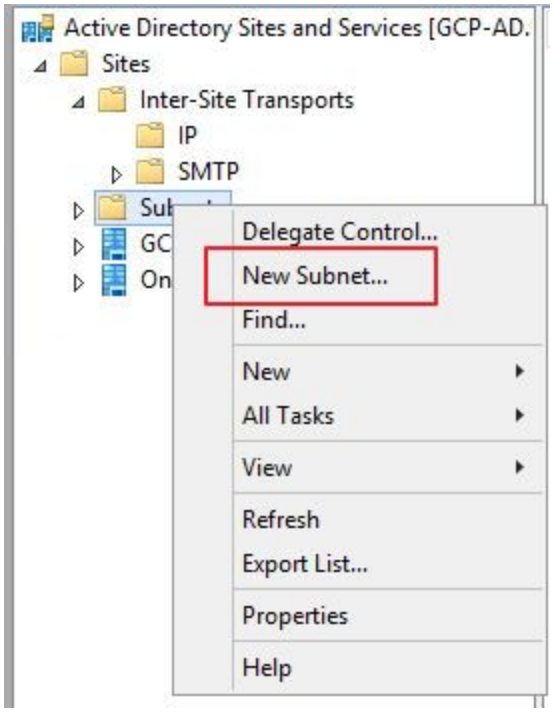
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

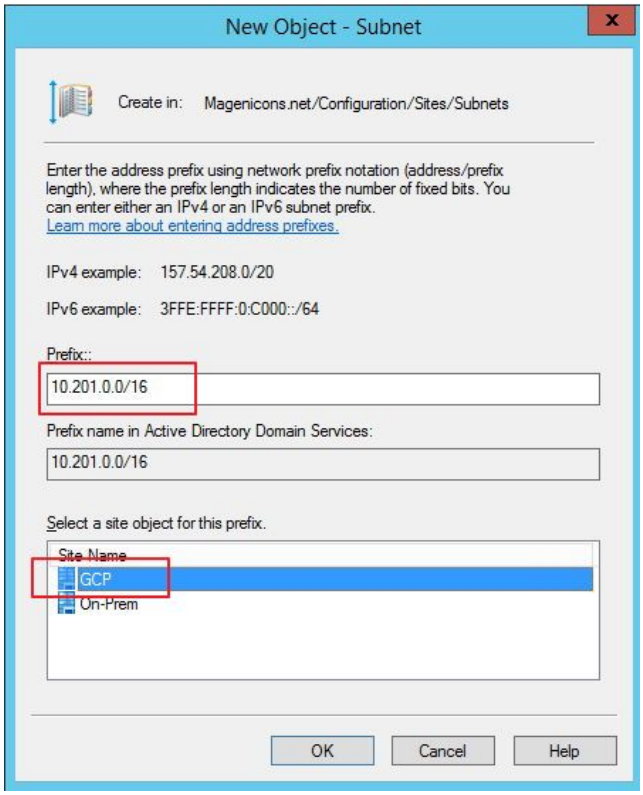




- Add a GCP subnet in Active Directory Sites and Services:

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

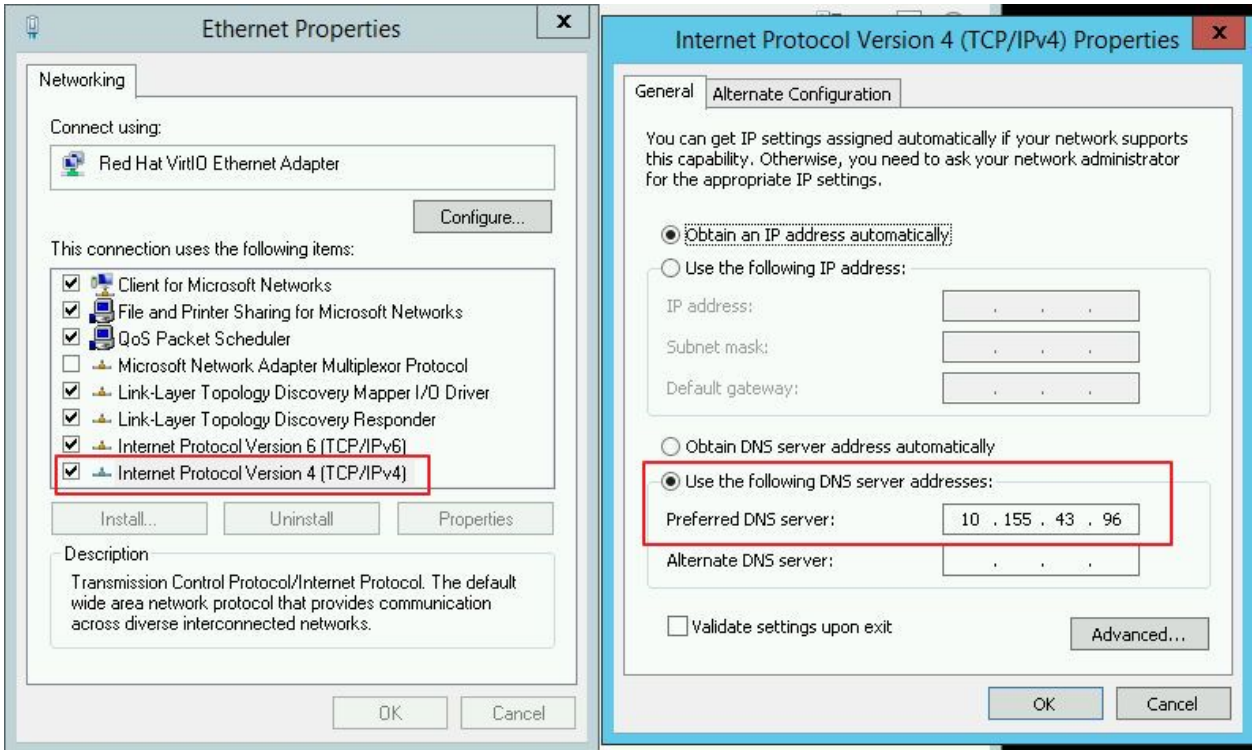




© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

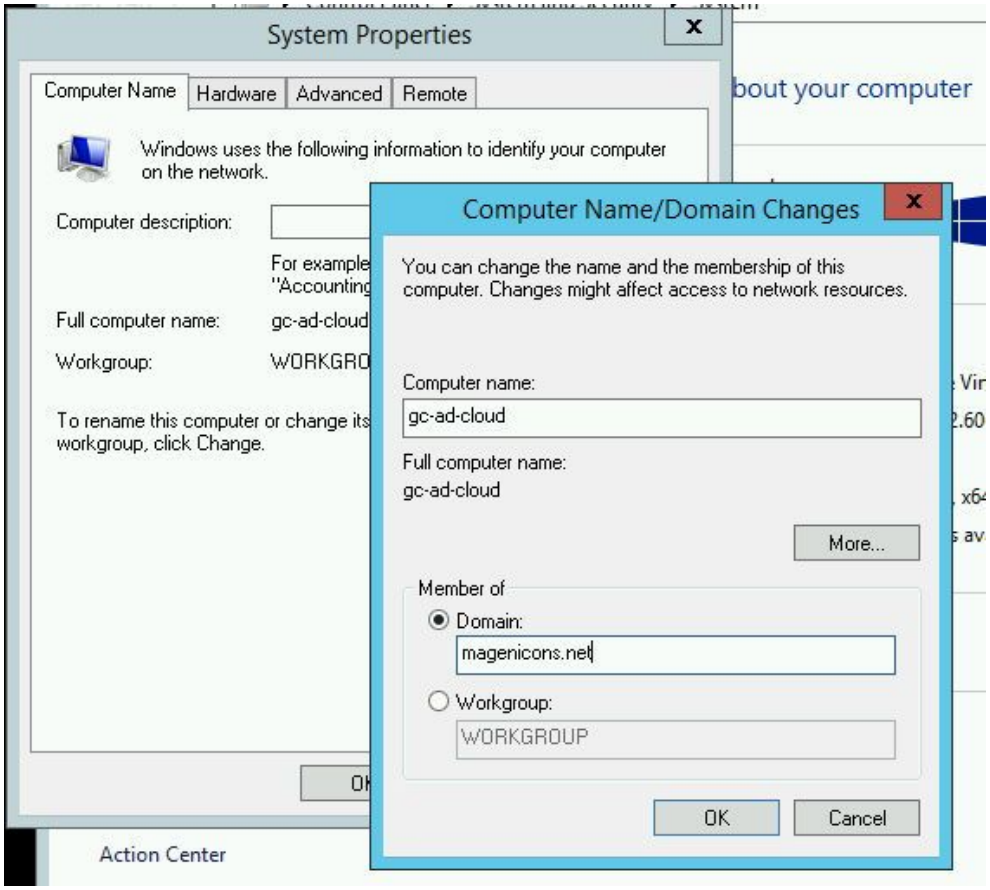
On the GCP AD DC

- Change the default DNS IP to point to the existing DC on-prem:

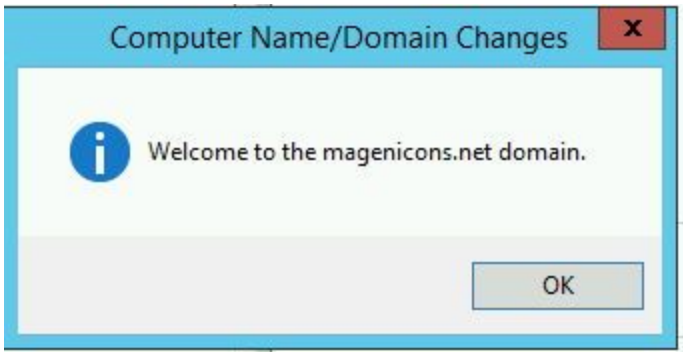


- Join the VM to the on-premises domain:

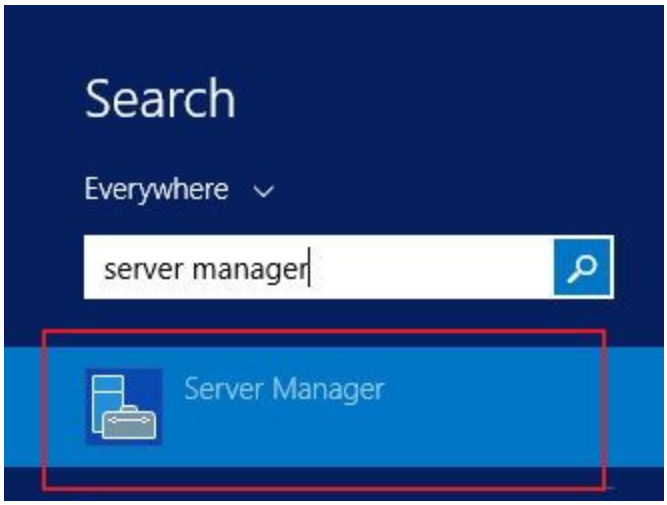
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

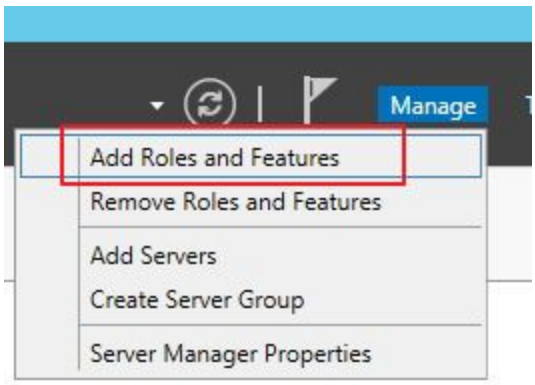


© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

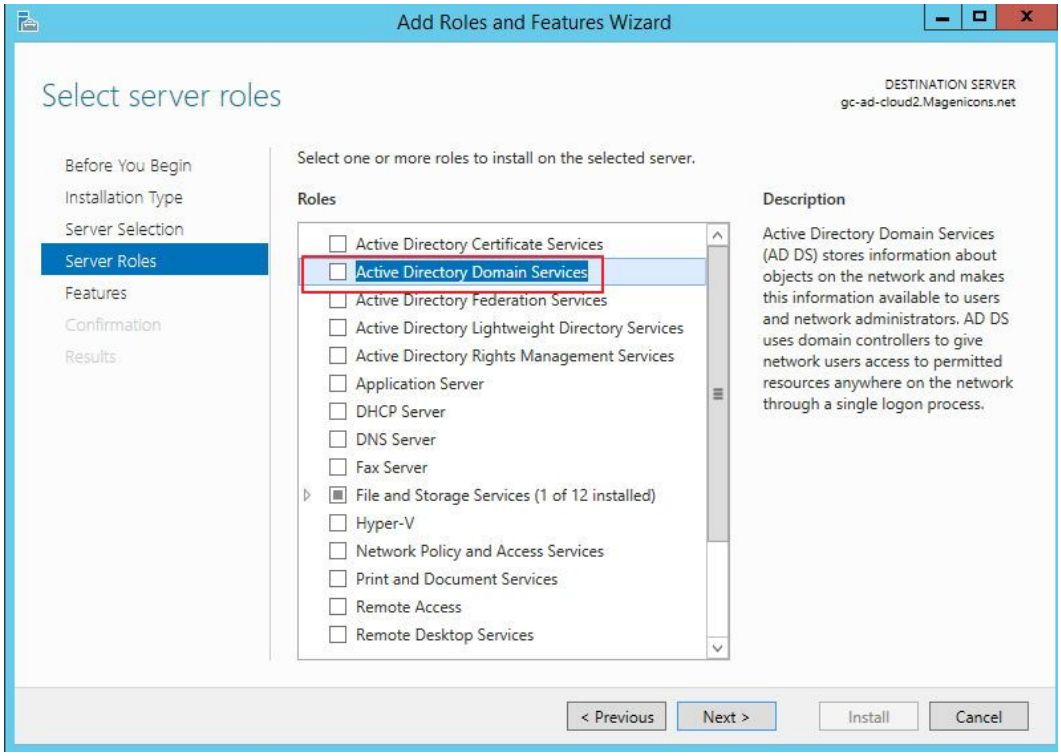


- Install Active Directory

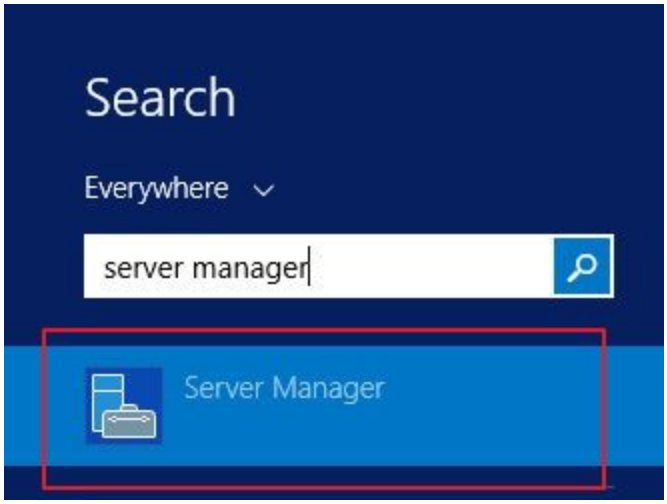




© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.



- Promote the VM to a domain controller



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.