Moving servers to Google Cloud Platform

Eelis Orvas

21.11.2018
This thesis goes into detail about migrating servers towards Google Cloud Platform. It was commissioned by Gapps Oy, where the writer works.

Google Cloud Platform is a cloud computing platform that anyone can start using and take advantage of the many services that Google has developed for themselves originally but has now shared with the public. They offer Google Compute Engine as an Infrastructure as a Service solution, which lets consumers or organizations use Google’s data center capacity for hosting servers.

The thesis goes through the free tools that Google offers for migrating existing servers to Google Cloud Platform. CloudEndure and Velostrata have different requirements for functioning and the way they work is opened.

There is almost always a need to know what type of things you need to pay attention to when migrating servers, so those are explained. These include networks, firewalls, licensing and other dependencies. The explanation also serves as a learning method for a reader to know what kind of a setup there needs to exist on Google Cloud Platform.

The results of this thesis are task lists for CloudEndure and Velostrata migrations that someone with previous knowledge of Google Cloud Platform can use as a guide when performing server migrations. The task list for CloudEndure was used to migrate a live server from UpCloud to Google Cloud Platform successfully. This served as a Proof of Concept method to verify that it can be used for migrating servers.

Further improvements need to be made to the Velostrata task list as it was not tested in this thesis due to the heavy configuration requirements.
Terms

Amazon Web Services = Amazon’s cloud platform that is similar to Google Cloud Platform or Microsoft Azure.

Google App Engine = Platform for deploying applications that can be set to scale automatically depending on the amount of traffic. Platform as a Service.

Google Cloud Storage = Google’s online data storage that can host very large amounts of data.

Google Compute Engine = Google’s virtual machine service. In essence an Infrastructure as a Service offering.

Google Cloud Platform = Google’s own public cloud that offers customers their tools in several different areas including Machine Learning, Artificial Intelligence, containers, databases and analytics.

G Suite = Google’s set of production tools that are the cloud equivalent to Office 365 by Microsoft.

Image = A snapshot of a device’s current state, which can be imported to other devices.

On-premise = As the name suggests, something that is located in a physical location.

Microsoft Azure = Microsoft’s cloud platform, similar to AWS and GCP.

Virtual Machine = Shortened as VM. A machine that is not physical, like a laptop for example.

Virtual Private Cloud = Virtual network built in the cloud for example, that can be connected to On-premise networks. This creates the standard model of Hybrid IT (On-premise and cloud used in tandem)
1 Introduction
   1.1 Objectives and structuring of this thesis
   1.2 Gapps Oy
   1.3 Scope of the thesis
2 Server migrations
   2.1 Moving to the cloud
   2.2 Costs
3 Google Cloud Platform
   3.1 Projects and folders
   3.2 Organization and IAM
   3.3 Compute Engine
4 Migrating your servers
   4.1 CloudEndure
   4.2 Velostrata
   4.3 Manual methods
5 What to pay attention to when migrating?
   5.1 Network
   5.2 Firewalls
   5.3 Dependencies to the old environment
6 Mapping the server and task lists for migrations
   6.1 Task list for CloudEndure migrations
   6.2 Task list for Velostrata migrations from AWS to GCP
7 Proof of Concept
   7.1 Migrating with CloudEndure
8 Conclusion
9 References
1 Introduction

This chapter will explain the objectives of this thesis, give a brief introduction to the commissioning company and tell the reader what the scope of the thesis is. The writer of this thesis is currently employed by the commissioning company and has extensive knowledge and experience on Google’s cloud services, including Google Cloud Platform.

The process for creating this thesis was done by first figuring out the necessary things to look at when considering migrations to another platform. These would be the requirements for the server’s functionality in the target cloud. After that it was needed to take a look at how these could be fulfilled in Google Cloud Platform and when that became clear, the migration tools themselves became the focus. It was necessary to look quite deep on how they function, and what their requirements are, to be able to create task lists that were as accurate as possible.

1.1 Objectives and structuring of this thesis

This thesis consists of eight chapters. Starting from chapter 2 this thesis aims to give the reader an understanding of why these migrations are very much a current topic. Chapter 3 is about Google Cloud Platform (referred as GCP) and will explain the reader, on the sections that are inside the scope of this thesis, how it works and what it is used for. Chapter 4 explains the methods available for individuals or companies to migrate their existing servers to GCP. These are either automatic or manual methods and they will be briefly explained. In chapter 5, I will be going through the common issues one might meet in these server migrations. Chapter 6 shows tables for mapping the server being migrated and task lists, to help the commissioning company’s employees do these migrations for themselves or customers. Chapter 7 is a Proof-of-Concept on a server migration to GCP and explains my process in making it successful. Chapter 8 is deliberation about the whole thesis.

The objective of this thesis is to package these migrations to GCP as a service that can be sold by Gapps Oy to customers and at the same time teach the current and future employees on what to take into account, when doing them. Some previous knowledge of GCP or other similar platforms like Microsoft Azure or Amazon Web Services (referred as AWS) is preferred and probably even needed before jumping ahead and going forward with the migrations.
1.2 Gapps Oy

Gapps Oy is a Finnish IT and change-management company that specializes in enabling a new way of working for its customers. Gapps is achieving this by utilizing Google Cloud technologies and developing completely new digital services, for or with, it’s customers by solving business problems. Gapps is a leading Google Cloud partner in Finland. (Gapps 2018.)

Gapps Oy is a part of Gapps Group together with its sister company Happeo (previously Universe). Gapps Group was formed in the beginning of 2018 after Happeo and Gapps were demergerg and Happeo was spun off as its own company in pursuit of scaling up its Software-as-a-Service (SaaS) business. While Happeo received 8 million dollars of funding for scaling up their business, Gapps remains completely (owned) by its original three founders.

Gapps' offering can be roughly split into three high-level categories as follows:

G Suite and related offering – Comprehensive deployments for new customers, which include change management, project management, technical implementation, and ongoing support services after the deployment of G Suite. Consulting for customers that are already using G Suite to fix a problem or problems they have noticed or even walking in with no prior knowledge of what they can improve and finding them in collaboration.

Google Cloud Platform (GCP) and related offering – Infrastructure modeling for companies that are either already using GCP or want to do so in the future. In a way, Gapps gives the customer the means to safely use GCP without worrying that its infrastructure is fundamentally built insecurely, incurring hidden IT costs or is just unmanaged by them. Gapps also has developers capable of building completely new solutions or software on top of GCP depending on the customer’s needs.

Happeo - While Gapps is no longer developing Happeo itself, it still remains one of the main partners of Happeo and continues offering the platform for Finnish customers.

Naturally, as an agile company, Gapps is open to brainstorm possible solutions outside of these fields, if the possible solution can be integrated into Google’s services somehow.
1.3 Scope of the thesis

The thesis does not tell the reader completely on how to build a GCP environment or infrastructure to suit the servers that are being migrated. It is also important to note that the GCP in this thesis is restricted to organizations. Either a G Suite environment or a Cloud Identity environment is needed to get access to an organization-specific GCP. GCP can be accessed with a consumer Google account, but it does not have all the infrastructure related methods/solutions available. Depending on the server that is being migrated, there might be certain network or firewall requirements that need to be met in GCP before the server functions as it did in the source location. This will be explained and there will be a section about the possibility of extending the organizations internal network to GCP, so that servers that require an IP address from an internal network can be configured to work as they did before being migrated. This work ignores the migration of databases or workstations and is confined purely to server migrations.

In addition, this thesis will focus on Google’s offering of external free server migration services in GCP and not on migrating your organizations complete server infrastructure to GCP. There are solutions available for planning full On-premise to cloud transfers, like Cloudamize, CloudPhysics and ATADATA. (Google 2018a.)

Velostrata can be used for single server migrations and very large server migrations. It will not be used in the Proof of Concept section of this thesis as it requires extensive network configuration before it functions. The task list for Velostrata is meant for AWS to GCP server migrations. On-premise migrations have a different flow and they require a vCenter On-premise server management solution which I did not have access to.
2 Server migrations

The purpose of this chapter is to briefly give a view on the cloud adoption rates of companies and give the reader an understanding on why these migrations are important. This thesis does not have the option of migrating On-premise servers towards cloud in the Proof of Concept part, but there is no real difference between migrating from local data centers to a public cloud like GCP or doing it from for example AWS.

2.1 Moving to the cloud

As public cloud is becoming more of a standard (partly due to cheaper costs) for enterprises, having a clear package to offer regarding migrations is surely going to be valuable. According to an IDG Communications survey, 38% of companies are facing pressure from either customers or management to migrate all of their applications to the cloud (IDG Communications 2018, 3). The same survey lists being vendor locked as the top challenge for companies in their cloud adoption (IDG Communications 2018, 7). A great part of the current public cloud platforms currently is that they do not vendor lock you, which is surely one helping fact in the increasing adoption rates. This coincides well with the topic of this thesis. Moving servers from either On-premises, or another Cloud Service Provider, like AWS, to GCP is a very current topic.

Rightscale’s State of the Cloud report (2018, 36) notes that the growing trend of containerization and serverless databases in the public cloud can take quite a bit of the old server infrastructure away. Even with those growing, the amount of enterprises that have over 50 VMs in the cloud has also risen from 2017 (Rightscale 2018, 41).

2.2 Costs

Migrations that use the methods mentioned in this thesis do not cost a fee on their own.

A survey conducted by Survata for OpsRamp, found that 94% of IT decision makers were of the opinion that moving more of their workloads to the public cloud would lower their IT costs. (OpsRamp 2017, Question 9.) With that, we can see that cost efficiency is something that IT-professionals think can be achieved by moving their workloads to the public cloud, like GCP.
GCP pricing is very transparent towards customers and it is fairly easy to get estimates about what one would pay for a certain workload running in GCP. These costs are lower when, for instance, you run a server permanently, instead of spinning one up periodically. There is a calculator available to get cost estimates for just about all the services available on GCP at https://cloud.google.com/products/calculator/.

This calculator can be used to pre-emptively calculate the costs of a new server created within a migration project. It can be used to count the price for a month, or a year.
3 Google Cloud Platform

Google Cloud Platform is a Cloud computing platform that offers the general public a collection of Google’s own computing resources (Meier 10.2.2017.). These resources are bundled to the general public as services, for example Google Compute Engine (referred as GCE), Google App Engine (referred as GAE) and Google Cloud Storage (referred as GCS).

These services are offered to customers from Google’s own data centers that are distributed globally. Even Finland has their own Hamina data center, that had its official launch in September 2018. GCP has a large offering of these services that range from categories like computing, databases, networking, storage, management tools, Internet of Things (IoT), analytics, security, Artificial Intelligence (AI) and machine learning. In total there are over 90 different services available for the general public inside these different categories. Many of these services can be set to automatically scale up based on traffic, which minimizes costs and maximizes efficiency.

These services can be used as stand-alone services or used with other services as shown in picture 1, where Cloud Load Balancing is used to split the traffic between regions to GCE instance groups that automatically scale up or down depending on the amount of traffic the load balancer sends them.

Picture 1. Globally autoscaling a web service on Compute Engine (Google 2018e)
3.1 Projects and folders

The services used in GCP are called resources and they have to reside in projects as they cannot be created without being set inside a certain project. These projects can be placed inside folders, which are used as Identity Access Management (IAM) control points in an organizations GCP architecture. Folders are not mandatory, but they provide more flexibility to IAM control and make it easier to determine which project belongs to what business unit of a company for example, from GCP’s cloud console.

3.2 Organization and IAM

Organization is the root node of a GCP environment and it is the hierarchical super node of projects. (Google 2018b.) In layman’s terms, it is the top-level IAM access management point that can be used to trickle down permissions for groups or users to all folders and projects in the environment. These permissions can vary from view-access to all projects and resources for a group or a user, to an access for deploying an application via GAE, but nothing else. Picture 2 shows a simplified standard architecture in GCP that has the Organization, Folders, projects and the resources at the bottom. It is possible to give permissions for users at any level of the architecture, so the IAM controls are very granular.

Picture 2. Resource Hierarchy (Google 2018f)
Below is Google explaining resource permissions in their best practices for IAM.

In larger companies the management of network and security resources such as firewalls are typically managed by a dedicated team, which is different from the development team. The development teams may want the flexibility to launch instances and carry out other actions related to instances in their projects. Granting bob@example.com the Compute Network Admin role at the organization level and alice@example.com the Compute Instance Admin role on her project project_2 allows Alice to carry out any actions on instances while preventing her from making any changes to the network resources associated with her project. Only Bob can make changes to the network resources in the organization and to any projects under that organization. (Google 2018c.)

3.3 Compute Engine

As Virtual machines (referred as VMs) run in the service called Google Compute Engine and this thesis focuses on migrating servers to it, it is necessary to expand on how it works.

GCE is a service that can be used inside projects to create resources that run inside the project. These can either be single VMs running Windows/Linux or automatically scalable instance groups. It is also possible to use custom Operating System (referred as OS) images that can be deployed to the VMs you are running, provided they are created with a certain type of container-optimized OS. These can be containers like Docker images. (Google 2018d.)

Depending on what the need is, GCE VMs can be used to serve customer facing applications or used to replace traditional On-premise servers like Print servers.
4 Migrating your servers

It is important to note that there are two different ways on how to move your servers to GCP - Automatically with tools built for this, or manually by going through the process of building the server anew on the target platform. These migrations can be performed by the organization striving for it themselves as a self-service with existing free tools, or with the help of a Google partner. (Google 2018a.) We will be going through the self-service ways of doing this.

Google offers two free external services for importing existing servers into GCE. These are called CloudEndure and Velostrata. They can be used in GCP, when the user goes to GCE and chooses to Import instead of creating a new VM.

4.1 CloudEndure

CloudEndure functions in a way where you install the CloudEndure Agent on all the machines (physical, virtual or cloud-based) that you want to migrate to GCP, choose the destination and the process begins. CloudEndure continuously replicates the source machine towards the destination and you are able to test that everything is in order before you cut off the replication and swap to using your new cloud machines. (CloudEndure 2018.) CloudEndure configuration is the same for all server migrations, no matter where you decide to bring a server from to GCP. On-premise and cloud servers will all work. All that is needed on GCP side before starting, is a GCP project that has billing enabled. (Google 2017a.) There is an option to create a Virtual Private Cloud (VPC) to GCP before migrating the servers, but it is not mandatory like with Velostrata. VPC can be used as a staging network to host the CloudEndure replication servers. (Google 2018g.) To get the VPC and the possible On-premise network working together, a Virtual Private Network (referred as VPN) needs to be configured between the intranet and GCP.

CloudEndure Agent is installed to the On-premise servers that you would want to migrate. The Agents then send information towards the CloudEndure Management Server that there is a device pending migration and at the same time starts creating a Replication Server in GCP and continuously replicates data towards it. When the Replication Server gets a command from the Management Server to create a server, a request is sent to the Agent for a snapshot of the latest data and a Replica instance is created in the Replica Network. (picture 3.)
4.2 Velostrata

Velostrata is a solution, and a company acquired by Google, that is primarily meant for migrating On-premise servers or servers from Amazon Web Services to GCP. It is wider than CloudEndure in the sense, that it is used to bring larger numbers of servers to GCP in one go.

Migrating servers from a corporate On-premise data center requires some configuration. On top of the project and billing requirements mentioned with CloudEndure, it is also required to set up a VPN or a Cloud Interconnect connection between the On-premise data center and GCP as well as a Virtual Private Cloud (VPC) in GCP. Inside the VPC, subnets need to be created as the Velostrata Cloud Edge components are deployed there. (Velostrata 2018b.)

There is also a need to install Velostrata Management Servers to either the source data center or to the destination (GCP) as well. (Velostrata 2018a.)

After you have a working VPN or Cloud Interconnect connection between On-premise and GCP, have installed the Velostrata Manager to an On-premise network and have added the necessary firewall rules in GCP as well as installed the Cloud Extension towards GCP, the process can begin. Velostrata Manager scans the On-premise network and sends the information to GCP through APIs and HTTP/TCP to the Cloud.
Edge servers that are created by the Cloud Extension. Those in turn send image information towards Cloud Storage and on request that Storage is used to create the Migrated VMs. (picture 4.)

![Diagram of GCP Deployment Overview](image)

**Picture 4. GCP Deployment Overview (Velostrata 2018a.)**

AWS to GCP migrations follow a similar setup and the only real difference is that the Management server is deployed in GCP. You still need VPC’s on both clouds, a VPN between them and the firewall rules.

### 4.3 Manual methods

There are, generally speaking, two ways to do manual server migrations. Those are:

- **Build the server and its functions from the ground up**, to a new server that resides in GCP and then connect it to whichever service the server is taking part in at its original place. In essence this is not a migration, but a rebuilding operation.

- **Backup a server to an image file and just import it to GCE**. After it has been imported, it is possible to spin up a new server and choose this said image as its disk volume.

  One easy way to do this is to use a free software that creates the image, like Clonezilla. Not all programs work on Linux and Windows, but Clonezilla does. (Burns 29.8.2018) Clonezilla should not be used on live
production servers, as it does not run the backup process in the background and effectively shuts the server down for the duration of the process.

The latter does not differentiate from the CloudEndure migration methodology in a major way apart from the image being taken from a certain moment and the process being run in the background. The principle is the same, but the automation handles the movement of the image to GCP and the server’s continuous replication/synchronization until the user commands CloudEndure to spin up a new server on GCP.
5 What to pay attention to when migrating?

There are several things that need to be taken into account when migrating a production server to a new destination platform. These things can be either obstacles or just hindrances, depending on what the server is used to operate. The most important thing before migrating the server is to understand what it is used for and what configurations effect its functionality (that do not come from the server itself and cannot be copied to an image).

5.1 Network

It is important to have full understanding on what limitations the network imposes on the server you are going to migrate. Companies often do not want their servers to have an external IP, which means that the devices are safely located in their internal network (intranet) and can only be accessed from inside it. Often their internal network is limited to be accessed only from their physical locations and via an encrypted VPN connection. In picture 5, you can see a very simplified diagram about accessing a company intranet from outside their physical locations.

Picture 5. Company intranet off-site access

If the server we want to migrate is and will continue to be located safely in the company intranet without external access, it is necessary to create a VPN connection between the company intranet and GCP as well as create Virtual Private Cloud between the intranet and GCP. This VPC can be turned to a Shared VPC. The Shared VPC can be used to link the different projects in GCP to the same network configuration. (Google 2018h.) Without using the Shared VPC, your networks are different on all projects and cannot be used in a wider sense. When you have a VPN connection in place between your On-premise network and GCP, you can use Google Cloud Router and its Border Gateway
Protocol (BGP) to establish a session with your On-premise router. This BGP session basically tells the On-premise router what kind of network settings and resources are in your Shared VPC (or VPC) and vice versa. (Google 2018i.) This BGP session is not required for the company intranet and GCP VPC to work together, but it acts as an information relay between them.

Picture 6, shows how a VPC network that has a VPN configured between an On-premise network and GCP, uses the BGP Session and Cloud Router to exchange information with the On-Premise network. This allows both of the networks to know what is inside each other.

![Diagram of VPC Network with BGP Session and Cloud Router](image)

As you can see, it does take quite a bit of planning to build an “extension” of your intranet to GCP, so you can operate servers meant for internal use there. For servers that can be used outside from your intranet, a large network setup is not really required, and you would only need to look at the firewall rules.

### 5.2 Firewalls

Some servers are completely walled off from the internet and others might have certain ports open, so they can do select operations that require access to the
internet. An example of this would be access to servers that control the licensing of a certain product. It is possible to stop all internet traffic from a server, except to license and update a service providers server with information about a licensed product. Naturally it is also possible to add a rule that would allow the server to accept connections from a certain address. One example that comes to mind is an application connecting to the server to fetch information from a database that is hosted on it.

The way that firewall rules work on GCP is that they are configured with either egress or ingress. These define the direction of the request. Egress describes traffic from a GCP resource (often a VM) toward a destination, whatever it may be. Ingress describes traffic from a destination, whatever it may be, towards a GCP resource. So, to say it simply: Egress rules are outbound and Ingress rules are inbound. These rules are either deny or allow. These can either be configured to allow or deny traffic to or from certain IP’s or all IP’s. It is also possible to add port or protocol requirements to the rule. For larger environments you can also create tags, which follow a certain rule, that you just attach to instances. This allows you to skip configuring the same rule to several instances or allow it to all instances. (Google 2018j.)

5.3 Dependencies to the old environment

For dependencies I would list at least the Operating System of the server and the licensing of said OS as well as licensing of applications running on the server. What does this mean? When you clone a server in the migration, is it possible to move the licensed product to another server or is the license tied to the original server? There are licensing models that are tied to a device for its whole life and cannot be moved. Licensing problems can often be negotiated and solved with some work, but it is necessary to recognize that they exist.

Straight dependencies are also services or devices that need to connect to the IP of the server being migrated. The server’s IP will almost always change in the migration and if these other services or devices connect to the IP and not a possible DNS name, the changes will have to be accounted for.
6 Mapping the server and task lists for migrations

Below, you can find a table to map the server and task lists made for migrating servers to the cloud with CloudEndure and Velostrata. The idea for the mapping is to list and comment on the configurations of a server about to be migrated to be clear that it is possible and mapped our properly. The task list will then go through the necessary preparations when moving the server. The task lists in particular are made in a way that following them requires previous experience with GCP. It would be very hard and arduous to create it in a way where every step that needs to be taken, is explained in detail. For the task lists, I have referenced materials from CloudEndure and Velostrata documentations available as well as my own experience. The CloudEndure task list will work for On-premise or cloud migrations to GCP and the Velostrata task list for AWS to GCP migrations.

Table 1. Server mapping table

<table>
<thead>
<tr>
<th>Server Operating System</th>
<th>Windows Server 2012 R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the specs of the server or the preferred specs for the new server.</td>
<td></td>
</tr>
<tr>
<td>What is it used for?</td>
<td>Database, Print Server, Domain Controller, etc.</td>
</tr>
<tr>
<td>Is it located in an On-premise network or a VPC?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Should it stay that way?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Is the On-premise network firewalled behind a VPN for external connections?</td>
<td>Yes (should just about always be)</td>
</tr>
<tr>
<td>Are there services that connect to its current IP and not a DNS name?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does it have to receive traffic from the internet?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does it have to send traffic through the internet?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>What licensed products are on it and can the licensing be moved to another server?</td>
<td>Yes/No/No idea</td>
</tr>
</tbody>
</table>
Opening the Server Mapping table, a bit:

- List the specs of the server or the preferred specs for the new server.
  - Here you would list either the current specs of the server or higher specs has been having some performance issues lately.
- What is it used for?
  - Pretty self-explanatory, the server should have some kind of a function if you want to migrate it.
- Is it located in an On-premise network or a VPC?
  - This relates to the need for building a VPN/VPC extension from the current source location to GCP. If it’s not located in either, there should be no need for any network juggling.
- Should it stay that way?
  - If it doesn’t need to be in an internal network, it makes the migration configuration easier.
- Is the On-premise network firewalled behind a VPN for external connections?
  - In rare cases an internal network does not have a VPN configured at all. This might indicate that getting the VPN to work towards GCP is a bit harder than normally.
- Are there services that connect to its current IP and not a DNS name?
  - These are generally internal servers and such, very important to know because all of those services need to be reconfigured towards the new IP that is generated in GCP.
- Does it have to receive traffic from the internet?
  - Translated as: Do we need to allow certain firewall rules on GCP side?
- Does it have to send traffic through the internet?
  - On default, this is allowed on Google side VPC’s, but I have noticed some companies walling off outside connections as well.
- What licensed products are on it and can the licensing be moved to another server?
  - This might kill the whole migration. If the licensing is very rigid and tied to the device at hand, you would have to contact the license provider and ask how this can be done.

6.1 Task list for CloudEndure migrations

From Google’s own guide to using CloudEndure, I created the following task list, that I will be using in the Proof of Concept chapter. (Google 2018g.)
1. Obtain access to the GCP environment where the server is being migrated to.
   a. If you don't have it yet, ask the customer to either create the target project and give access in it to your Google account, or give you access to a certain folder or the whole organization from IAM. The level of access needed depends on networking requirements.

2. Create a project in the target environment, where you will migrate the source server.
   a. Take note of the project ID, you will need it later.

3. Create a Service Account in the project (can be created in another project but needs access in IAM to the target project afterwards).
   a. Save the .json file, you will need it later.

4. Sign up to CloudEndure VM Migration Console.
   a. Done from GCP -> Compute Engine -> Import -> CloudEndure

5. (Optional) Create a VPC network or a Shared VPC and a VPN connection in GCP. This is needed only in cases where you are migrating servers that you want to connect to your company intranet.

6. Provide your Project ID and the .json file you get when creating a Service Account in the CloudEndure portal you activated in step 4.

7. Choose the region where you want to create your new servers. For Gapps Oy this would often be europe-north1 (Hamina data center).

8. Choose the replication server network, if you didn't create a VPC network just choose default.

9. Install the CloudEndure Agent program on source server.
   a. Get an installation token from the Migration Console that you input to the Agent on the source server.

10. Check that connection is working from source server (Port 443 towards console.cloudendure.com open) to the CloudEndure VM Migration Console.
    a. You can check this from the Migration Console. If the connection is not working, there should be a red stop sign icon in the status column.

11. Configure the target server in the Migration Console’s Blueprint section (this includes machine type, name, network, internal IP and disk).
    a. This is where the understanding of the server you are migrating comes to play. If it needs to get an internal IP and be seen by the On-premise network’s router, you would need to have a VPC network that has a VPN configured towards the On-premise
network. You could also add a Cloud Router and a BGP Session as well.

12. Check from the Migration Console that the initial synchronization is completed.
   a. On the Status column it should read “Ready for Testing” if it is completed.

13. Test the VM creation towards GCP from the Migration Console.

14. Test that connections are working to the newly created GCP server and applications on it are working as they should.

15. Disable access to source server and Cut the migration from CloudEndure console. Test everything still works on the new server.
   a. This causes downtime for the service until the DNS is updated.

16. Change DNS to point users towards the new server.

17. (Optional) If needed, install the guest environment package on the new server.
   a. This helps if you have issues with users logging in to the source server

18. (Optional, depending on if you are just terminating the old server) Remove the installed Agent from the source machines.
   a. Check all the boxes on the Migration Console machines and press “remove machines from this console”

19. You are done. Congratulations!

### 6.2 Task list for Velostrata migrations from AWS to GCP

Disclaimer: I highly recommend that you have already configured VPC’s on both the AWS and GCP side that are connected via a VPN before starting this. Velostrata offers scripts that are supposed to perform the configuration of networks on both sides, but from my experience they do not work properly.

For On-premise source migrations, the GCP configuration is the same. On-premise and GCP need have to have the same VPN connectivity available and you need to have vCenter in use, to which you install the Velostrata appliance. For On-premise migrations, the Velostrata Deployment Manager is installed On-premise and not in the GCP.

Velostrata migrations are very complex as the solution can be used to move at least 250 servers to GCP in one go. They have an extensive, but a bit inaccurate documentation regarding the solution at their official documentation page, which is used to create this whole task list. (Velostrata 2018c.) Be prepared to use it as there is no way to create an all-in-one task list for Velostrata.
Pre-requisites:

- **GCP**:
  - A project that has a billing account attached to it.
  - VPC with subnets
- **AWS**:
  - VPC with subnets
  - VMs that you want to migrate
  - Account with access to create networks, instances, security groups and users.

1. Check your GCP Billing account ID and store it somewhere, it’s needed later.
2. Go to your GCP project and enable the following APIs for it
   a. Cloud Resource Manager API
   b. Identity and Access Management (IAM) API
   c. Compute Engine API
   d. Google Cloud Storage API
   e. Google Cloud Deployment Manager API
3. Check from GCP IAM that you have the following roles
   a. Owner
   b. Compute Admin
   c. Organization Administrator
4. Configure your GCP VPC to talk to your AWS VPC through a VPN, if you have not done so already. There are fairly many guides available for doing this.
   a. After setting it up, check on GCP side that the VPN tunnels show the connections as established and AWS side shows the tunnels as "UP".
   b. Test the connection
5. Add the firewall rules in GCP that are laid out in picture 7.
6. Create the following Service Accounts in GCP (you can also just create a single Service Account and have it do everything that is needed, but the recommended way is three different ones with least privilege required for operations):
   a. velos-gcp-mgmt-sa
   b. velos-gcp-ce-sa
   c. velos-gcp-worker-sa

7. (In GCP) Create the required roles and assign them to the Service Accounts
   a. Required commands and script files for creating the roles and assigning them to the service accounts can be found from Velostrata’s GCP role creation documentation (Velostrata 2018e).
   b. These .yaml files should be uploaded to the GCP Cloud Shell and executed from there.
   c. There might be a problem with the command to assign a role to a Service Account listed in the link, but just change the following
part: --role "velos_xx_role" to --role projectid(whichever it
is)/roles/"velos_xx_role"
d. It is possible to create these custom roles manually, but the
largest one has around 80 different permissions, so it's not
practical.

8. (in GCP) Create a key for velos-gcp-mgmt-sa Service Account and
download it as a .json file. You will need it later.

9. Download this template, created by Velostrata, for AWS Role, Policy and
Security Group creation to be used in CloudFormation. Set the template
to the VPC you have up and running in AWS.

10. Create a user with programmatic access in AWS and assign it to the
Security Group you just created with the template. Create an access key
and secret key for the user and save them.

11. Deploy the Velostrata Management Server to your GCP with the following
Cloud Shell command:

```
gcloud beta compute --project=YOURPROJECTID instances create mgmt-gcp --zone=us-east1-b --machine-type=n1-standard-2 --subnet=velos-deployment-subnet-private --
network-tier=PREMIUM --metadata=subscriptionId=YOURBILLINGID --maintenance-policy=MIGRATE --service-account=YOURCOMPUTESERVICEACCOUNTID-
compute@developer.gserviceaccount.com --
only,https://www.googleapis.com/auth/trace.append --tags=fw-velostrata --
image=velostrata-mgmt-3-5-0-20487-18086-os --image-project=velospublic --boot-disk-size=60GB --boot-disk-type=pd-standard --boot-disk-device-name=mgmt-gcp
```
a. Add your project ID to the corresponding part.
b. Change the zone depending where you want to deploy the server.
c. Change the subnet name to what you have available in the zone.
d. Replace YOURBILLINGID with the Billing ID of your Billing
account.

e. The Compute Service Account can be found from your projects
IAM page. This is automatically created by GCP.

12. Wait for the server to spin up and then connect to it from your browser
using this URL https://your.server.external.ip (example https://
34.55.123.17)

```
a. Go past the malicious site warning if that pops up.
b. For credentials, user: localsupport and password: your billing
account ID
c. Decide if you want to send data towards velostrata, what region
the data is stored at and accept the terms.
13. Go to Target Cloud, use the following credentials to get in, user: apiuser
password: your billing account ID, move to the Cloud Credentials tab and
add the velos-gcp-mgmt-sa Service Account and upload the .json file you
got when creating the key.

   a. Edge network tag should be “fw-velostatrata”
   b. Default network tag should be “fw-workload”
   c. Service Account for Cloud Edge should be “velos-gcp-worker-sa”
   d. Availability zones can be the same, but also different in case
      networks go down.
   e. Pick your Node subnets, these can be same or different as well.
   f. Labels can be used to mark devices that are being created on
      GCP side.
   g. Velostrata has a full listing, where they open each Parameter In
      bigger detail (Velostrata 2018f).
   h. You create the Cloud Extension by clicking “OK”

15. Go to Home, and head over to Source Cloud. In the Cloud Credentials
    tab, add the AWS user that you created earlier and it’s Access key and
    Secret key.

16. Go to the Cloud Extension tab and click create.
   a. the only thing you need to add yourself is the name. Other fields
      can be filled from the drop-down menu and should be easy to
      figure out.
   b. Create the Cloud Extension by clicking “OK”

17. Wait for the Cloud Extensions to be deployed and then head over to
    Home -> Automation Runbook.
   a. Make sure your AWS instances have metadata tags added to
      them. This can be done easily by choosing several of them at the
      same time and going to Actions -> Instance Settings -> Add/Edit
      tags. Add tags if there are none.
   b. Use those tags in the automation runbook in the Filter by Source
      tags section.
   c. Tap the checkbox in the Target Network setting and hit Create.

18. The Runbook is downloaded to your device as a .csv file.
   a. change the RunGroup from -1 (skipped) to for example 100, 200,
      300. These mean the order that the machines are being migrated
      in, 100 being the lowest(first).
   b. Populate the TargetEdgeNode and the TargetInstanceType if they
      are not populated.
c. all values need to be in lowercase or they might fail.
d. Save the file

19. Go to the Automation Runbook window again and click Start New Job, name the job, upload your .csv back and choose the operation.
   a. “Full Migration” Creates the VMs to the target cloud and starts waiting for you to Create a new job with the same .csv and “Prepare to detach” operation. Once you run that, the next step is to run a “Detach” operation once more with the same .csv. When that is done, you can run a new job with the “Cleanup” operation.

This Operation cycle is good for making sure that everything works as expected on the target cloud. At any point before the “Cleanup” operation you can run a “Move Back” operation to reverse the migration.

b. For On-premise as a source, there is an option to run a “Test Clone” operation.

20. When you have performed the cycle with checks in between that everything is working, you are done.
7 Proof of Concept

In this chapter we will use the task list created in the previous chapter to move an actual server to GCP using CloudEndure.

CloudEndure will be used to move a server from another Cloud Service Provider (referred as CSP), UpCloud to GCP. The server is basically a fresh one and was created for this migration.

Velostatra will not be used in this Proof of Concept as the configuration required for it to work is so extensive, that it falls out of the scope of this thesis.

7.1 Migrating with CloudEndure

Tasks 1-3:
I started the CloudEndure migration by creating a project in GCP called “servermigtest” and by creating a Service account to it in the IAM -> Service Accounts page in Google Cloud Console. I also added a project owner role to the Service Account when prompted during the creation. As per the task list I laid out before, I saved the .json file locally just before creating the Service Account. In picture 8, you can see the created Service Account and that it has a key configured. The other Service Account is created automatically when CloudEndure spins up a replication server to the project.

![Service Account created](image)

Picture 8. Service Account created.

Task 4:
Next, I headed to GCE -> Instances -> Import VM, which gave me a pop up to migrate between CloudEndure and Velostatra. For this migration I chose CloudEndure and clicking the icon opened the Google VM Migration service portal login screen (picture 9).
Tasks 6-8:

After signing in with my Google account, I had to input my project ID (servermigtest) and the .json key that I downloaded when I created the Service Account. Straight after setting those I chose the network region and zone for the replication server that CloudEndure creates. This replication server is used as a go-between for the source server and the target server. In essence, it keeps requesting new data from the source server and is used to create the new server when the time comes. I chose Google EU North (Finland) and europe-north1-a as the zone. On the same step it is possible to choose the network you want to use on GCP side for the replication server and for me the default network works out fine. Picture 10 shows the network, zone and default network choices.

Picture 9. Signing up for CloudEndure.

Picture 10. Replication Settings.
Tasks 9-10:
With that, the initial configuration just needed the CloudEndure agent to be installed on the server that I wanted to migrate. In the Other settings tab visible in picture 10, you can find the installation token needed for the agent. By pressing the “How to add machines” button you get a link for the agent and the full command needed to run in Windows command prompt (token edited for security reasons):

```
installer_win.exe -t A7BA-5559-FB49-8S2D-4822-B149-620B-6048-E1C3-8415-EX0C-9876-8F35-4F73-A66F-56F5 --no-prompt
```

This prompted the install of the agent and nothing else needed to be done on source server. When I headed over to see my machines after the installation, I could see that the replication process had begun, as seen in picture 11.

Picture 11. Data replication process.

This replication process can take up to 24 hours depending on the amount of data on the server and the only step to do before it completes is to edit the blueprint ready for launch, it is opened by clicking the device in the machines tab.

Task 11:
For my blueprint, I chose the default network instead of creating a new one and set my new servers name as “migration1” (picture 12).

Picture 12. Blueprint of the server being migrated.
Tasks 12-14:
I continued the migration when the data replication process was completed, and the synchronization had turned to continuous. I thought it would be a good example to install Google Chrome on the server and seeing if the continuous replication would bring the program through to the new server. After installing Chrome on the server, I clicked the machine in the migration console and chose “Test Mode” (picture 13).

![Image](image13.png)

Picture 13. Launching Test Mode.

When Test Mode is launched, CloudEndure sends a command to the agent installed on the source server and asks for the latest snapshot of the device to make sure it gets all the latest changes. The Job progress tab seen in picture 13 is very convenient and shows what is happening with the test server creation. Creating the test server took six minutes in total. I logged in to the new server and could verify that Google Chrome was indeed installed there as well.

Task 15:
My next step was to add another program to the source server and see if it still gets replicated to my new server that was already up in GCP. I installed PuTTY to the source server and pinned it to the Windows taskbar. I went to the migration
console and chose “Cutover Mode” for the server being migrated. CloudEndure then deleted my Test server and fetched a new updated snapshot from the source server. Using that snapshot, it created a new server in GCP with the latest data available. This cutover launch took just under seven minutes and I had a live server in GCP. When I logged in to check the server, I could see that PuTTY was installed and properly pinned in the taskbar, just like I did it was on the source server.

Task 16:
If this was a production server, I would have had to change the dependencies toward the server now, which usually is the DNS name that points to the server’s IP address. In cases where the source server IP is what dependent services use to connect to, the IP for the new server needs to be changed to all of them separately.

Task 18:
Next, I removed the source server from my migration console, which deletes the migration agent on the source server as well (picture 14).

Remove Machine From Migration Console

You are about to uninstall the CloudEndure Agent from 1 Source machine.
This will cause Data Replication to stop and the machine will no longer appear in the CloudEndure Console.
Note that the 1 Target machine that you have launched will also no longer be visible here, however it will not be deleted and can still be accessed in your Google Cloud Platform console.

Picture 14. Removing the source machine.

Task 19:
With that, the CloudEndure migration is completed and the source server can be seen in GCP with its new IP (picture 15).
Picture 15. Migrated server up and running.
8 Conclusion

After going through a pretty deep dive on CloudEndure and Velostrata I’ve realized that they do not go for the same market in my eyes. Before starting this, I thought that these two solutions are competitors, but the truth is that even though they tackle the same issue, one is much wider than the other in the services that it provides.

CloudEndure can be used as a sort of “quick and dirty” method of moving Servers from one place to another and for that it works very well. Of course, there is the option to use it in a similar way as Velostrata with building the VPC and VPN constructions to GCP and the source platform, but it is not needed and after getting to know Velostrata properly it felt very refreshing. Overall, I felt like CloudEndure would be my pick for migration projects that consist of tens of servers, instead of hundreds. You need to do a bit more of the configurations for each server that you migrate, but I also felt that it gives you a bit of security that everything will work out. Maybe that’s just me though, as I’m more accustomed to working hands-on in an agile manner and not in large companies.

I didn’t get to try Velostrata in the Proof of Concept phase, but from what I can tell, it is clearly made for enterprise grade migrations where you want to bring hundreds of servers to the cloud. This can be seen from the quote below by their CEO.

> When we started Velostrata back in 2014, we laid out a clear vision – to simplify and accelerate enterprise transition to cloud (Ben-Shaul 9.5.2018).

Velostrata configuration is rigid and requires more, but the solution is also more advanced. If I were to guess, that would be one of the reasons for Google buying the company around 6 months ago. It also gives the option to bring workloads back to On-premise after you have migrated them to Google, which is not something that CloudEndure is capable of.

I feel like I managed to get together a very good task list for CloudEndure migrations, but for Velostrata time will tell how useful it actually is. We aim to use it to verify the solution later on, but for now it will serve as a starting point. It will most likely go through some modifications and there is a definite requirement to create the On-premise migration task list later on. That however, requires a vCenter solution that the commissioning company currently does not have.
I would say that I have learned a massive deal about Cloud networking and how it works in both AWS and GCP side during the time that I compiled the task list for Velostrata. To be specific, I learned how to create a VPN between AWS and GCP while doing my testing and actually got it to work, which was a first time for me. I haven’t really used AWS before and this was a pleasant surprise. As a process this thesis has been a big learning experience. For someone who has been solely focusing on Cloud technologies for over a year now, it opened my eyes on how much I still do not know, especially on the networking side. Generally, my work doesn’t involve much of the networking side, but this thesis has given me a slight boost in trying to get involved with it in the future.

Managing my time while doing this thesis wasn’t especially difficult. Neither was arranging the tasks for myself as I’m used to working projects and this can just be counted as one of them.
9 References

Ben-Shaul, I 9.5.2018. Velostrata has signed an agreement to join Google Cloud!
Readable: https://velostrata.com/blog/velostrata-has-signed-an-agreement-to-join-google-cloud/
Read: 19.11.2018

Burns, S. 29.8.2018. Create a backup image in 7 easy steps.
Readable: https://searchdatabackup.techtarget.com/tip/Create-a-backup-image-in-7-easy-steps
Read: 4.11.2018

Readable: https://www.cloudendure.com/live-migration/google-cloud/
Read: 27.10.2018


Read: 3.11.2018

Readable: https://cloud.google.com/vm-migrate/ Read: 20.10.2018

Readable: https://cloud.google.com/resource-manager/docs/creating-managing-organization Read: 27.10.2018

Readable: https://cloud.google.com/iam/docs/resource-hierarchy-access-control
Read: 27.10.2018

Google 2018d. Deploying Containers.
Readable: https://cloud.google.com/compute/docs/containers/deploying-containers
Read: 27.10.2018
Google 2018e. Globally autoscaling a web service on Compute Engine.
Readable: https://cloud.google.com/compute/docs/tutorials/globally-autoscaling-a-web-service-on-compute-engine
Read: 27.10.2018

Readable: https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy Read: 27.10.2018

Google 2018g. Migrating VMs to Compute Engine using CloudEndure
Readable: https://cloud.google.com/compute/docs/vm-migration/using-cloud-endure
Read 4.11.2018

Google 2018h. Shared VPC Overview.
Readable: https://cloud.google.com/vpc/docs/shared-vpc
Read: 10.11.2018

Google 2018i. Cloud Router Overview
Readable: https://cloud.google.com/router/docs/concepts/overview
Read: 10.11.2018

Google 2018j. Firewall Rules Overview
Readable: https://cloud.google.com/vpc/docs/firewalls#firewall_rule_components
Read 10.11.2018

Read: 18.11.2018

Meier, R. 10.2.2017. What is Google's Cloud Platform?
Readable: https://medium.com/@retomeier/what-is-googles-cloud-platform-d92a9c9e5e89 Read: 27.10.2018

OpsRamp 2017, Question 9. Readable: https://surveys.survata.com/public/3710cdfb-89ae-484a-a0fb-0b11e764a049/it-dms--500-emp
Read: 10.11.2018
Rightscale 2018. State of the Cloud report. Readable: 
Read: 18.11.2018

Readable: http://docs.velostrata.com/m/79700/l/927406-technologies-architecture-and-deployment Read: 3.11.2018

Velostrata 2018b. Requirements for GCP as Cloud Target (Accounts + VPC) / Overview. 
Readable: http://docs.velostrata.com/m/75846/l/819336-overview Read: 3.11.2018

Velostrata 2018c. Velostrata Deployment. 
Readable: http://docs.velostrata.com/m/75846 Read: 16.11.2018

Velostrata 2018d. Network tags configuration. Readable: 
http://docs.velostrata.com/m/75846/l/819338-network-tags-configuration Read 16.11.2018

Velostrata 2018e. Creating GCP Roles and Service Accounts manually 
Readable: http://docs.velostrata.com/m/75846/l/819339-appendix-creating-gcp-roles-and-service-accounts-manually Read 16.11.2018

Readable: http://docs.velostrata.com/m/75846/l/925057-add-a-cloud-extension#to-add-a-cloud-extension-to-gcp_1 Read 16.11.2018