

Introduction

In this whitepaper, we will try and dispel some of the art involved in VDI and turn it back into a science. It's all about the Template - the very heart of your VDI deployment.

Whether you are focused on Healthcare, Manufacturing, Education, or any other industry, the IT used to power your business is often a "behind the scenes" resource that users do not directly interact with. For example, Active Directory just runs and there are numerous ways to make it redundant. Should a single server fail, the impact to the end-user is unnoticeable. VDI is the exception to this. If you get the Template wrong in VDI, the effect is immediately felt by the end-user. Choose the wrong amount of memory, and applications will perform poorly. Over optimize the desktop and it will be fast, but users won't like to use the platform you are delivering – it could be as simple as they don't have their desktop wallpaper. IT is now scrutinized by the most critical of judges, the end-user and the perception of the end-user community will make or break your VDI role out.

There are loads of papers out there that tell you what to do. This paper will focus on the WHY. We have worked on countless VDI deployments and have seen templates done well and templates that can bring users screaming to your door. The WHY behind optimization will help you to create a Template that works for you organization. Following a how to guide off the net will not help if you don't know their motivation for each step. If there is one thing we have learned at HiveIO, there are no two companies that look alike.

Let's start at the beginning, skip ahead if this is too basic.

The Template is your golden image, often called the image. The combination of an operating system and just the right number of applications installed, all optimized for VDI. The Template has been referred to as the golden image, the replica, or various other names - all of which mean the base "C:\\" of the computer. It is a re-usable package that is easy to update and maintain. The Holy Grail: manage a handful of templates rather than hundreds of thousands of virtual desktops.

Too often, virtualization vendors stop at stating, "make sure to optimize the template before rolling this into production." This leaves you, the customer, to research the how and why and, more importantly, what the implications are of creating and optimizing the Template. But why? They are the experts after all! Leaving a customer to fend for themselves is not the HiveIO way. While it is true, no two companies are alike. Optimizing the Template is specific to each company. It's the hard part, you shouldn't be left guessing how to do it! At HiveIO, we get it. It is one of the most critical pieces to a successful VDI deployment. Being focused on a customer's success is exactly why we are writing this whitepaper. While we can't write every template for every customer, we can write a whitepaper that gives everyone the guide they need to develop the best possible template for their organization. Optimizations come in many forms: choosing the right Windows OS, removing unnecessary software, disabling services, sometimes making Windows Registry changes, or changing the startup behavior, and even modifying the default user settings. (As a side note, the simplest would be to create an optimized template ready for your applications. Unfortunately, that would mean distributing Microsoft Windows and Microsoft takes issue with that!)

The Template is crucial to the success of your VDI deployment!

There is a different approach taken in VDI deployments, depending on whether the computer is Persistent or Stateless.

Persistent Pools of persistent desktops allow users to make changes to the applications and even the OS itself and have these changes persist across a reboot of the VM.

Persistent VDI is a typical approach for virtual desktops used by developers and IT administrators. However, most users do not require machines that save all changes across reboots or ones where they need administrative rights - despite what they may tell you.

Stateless (also referred to as non-persistent) VDI pools do not save changes made within the virtual machines, a VM rebuilds every time a user logs off their session, or the VM reboots. This architecture ensures that a clean build of the Windows OS is provided each time a user logs in – they get the same fast end-user experience every time they login.

With profile management technologies, the end-user now has the ability to *persist their settings*, such as their desktop wallpaper while in a Stateless VDI. This level of persistence gives a seamless experience to the end-user and the impression that the VM has not changed between logins. A great balance between end-user experience and simple IT management.

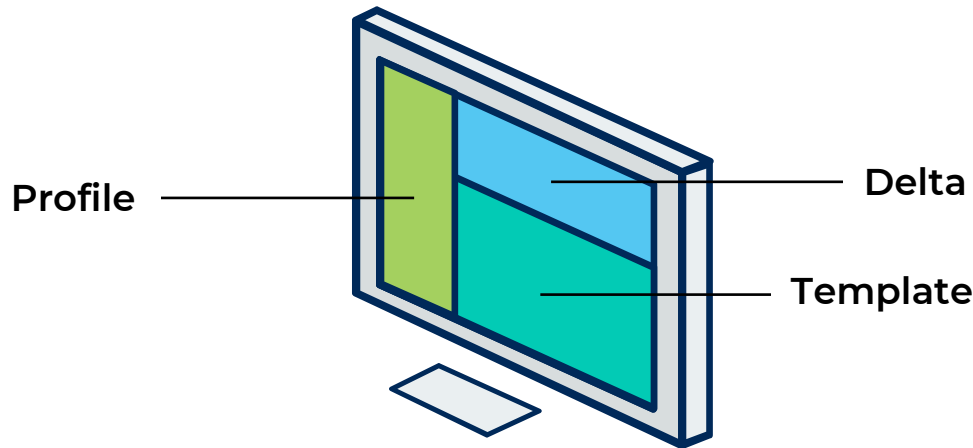
There has long been a battle between whether to deploy Stateless or Persistent desktops. As you are quickly learning with VDI - there isn't a single answer for every organization. The right approach in the majority of organizations we work with is to force a default to Stateless desktops and then bring Persistent Desktops back in for the use cases that need them. It may seem simpler to migrate all of the desktop management techniques that have been used in the past to your new VDI environment, it won't pay off in the long term.

A Stateless desktop shouldn't restrict what the end-user can do and will allow you to be much more agile as an IT organization. There may be some push back in the beginning and it may take time to get the process right, but once you do, it will pay dividends.

Why? With a stateless desktop you can leverage the benefits of leading-edge acceleration technologies, like In-Memory storage – providing the very best end-user experience. You also don't waste infrastructure deploying persistent VMs that are hardly used or are there 'just in case.'

Persistent desktops in general are harder to manage, as they tie the user to a specific desktop creating additional maintenance work for IT and reducing your ability to leverage your infrastructure. Once assigned, the Persistent desktop is not available for pooling resources even when it is not in use, thereby reducing your efficiency. Persistent desktops do have their place. For example, some applications are written in a way that demand a persistent desktop, some peripherals install drivers that need persistence, and there are legacy licensing models for applications that mandate their use. Based on industry trends with regard to licensing and drivers, we believe the need for Persistent desktops will continue to reduce over time.

From Template to Virtual machine



In most VDI platforms, a desktop or virtual machine has three parts. The **Template**, your optimized golden image that is read-only when deployed. A **Delta** disk is created automatically by the system when starting a desktop and is used to provide the read-write layer on top of the Template. Any changes to the VM that occur while it is running reside in this disk. Finally, the **Profile** is how we personalize the desktop. Today we tackle the Template.



*A Template is an install of Windows, locked at a specific point in time.
It includes your organization's applications.*

There is very little difference between a persistent and stateless virtual machine from a technical perspective. In most VDI deployments, the difference is centered around the Delta disk. When an end user logs out or a computer reboots – do we keep it or throw it away.

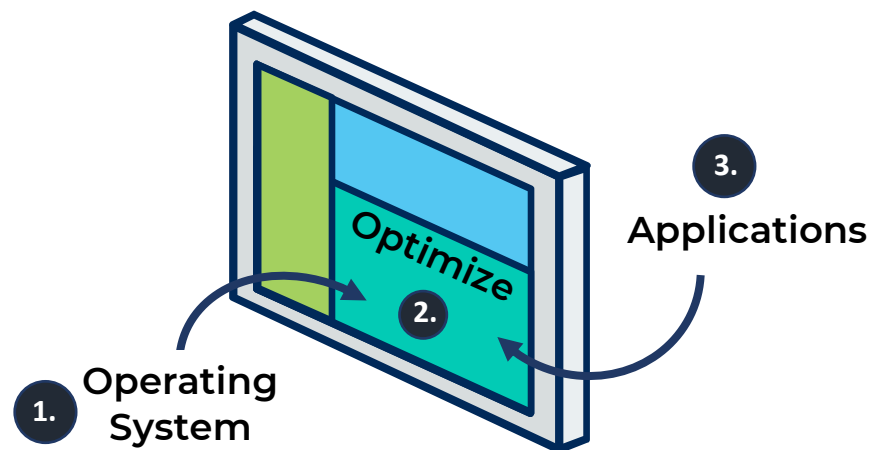
Keep it: **Persistent**
Throw it away: **Stateless**

This is an important and powerful concept as it allows us to keep a single Template for both stateless and persistent pools of desktops.

How should I Build and Manage a Template?

Think about all of the changes that occur to your local computer on any given day, from internet cache and downloads to additional applications you add. These can cause significant bloat to the delta disk, which in turn can consume capacity, and in some cases, performance problems on the underlying storage. Getting the Template right is the key to avoiding these issues.

In VDI deployments, regardless of the stateful nature of the desktop, each of the pooled desktops boots from the same Template. This includes the OS, core applications, settings, styling, and restrictions for the Windows computer itself. There are many methods for deploying applications once the OS is in place. For VDI, add as many of these core components to the base template as possible. This will keep the growth of your overlay disk to a minimum.



1. The operating system. The Operating System (OS) is the foundation of your Template, and its installation is the first step in building a template. You can install the OS from an ISO or fully automate the install using [various tools](#) such as the [Windows ADK](#) or [Windows MDT](#). Document your install procedure so that you can build new templates from scratch in the future. The decision on which technology to use to build the template will likely depend on the experience and knowledge of the administrator. For a small number of templates simply using the ISO to create a base 'Gold Image' that every other template is built from might be the simplest approach. For environments with a larger number of templates or rapidly changing templates, automating the install may be advantageous and can be achieved with the tools mentioned above.
2. Optimize the OS. This is easier said than done. There is an entire section down below covering this in detail.
3. Install your applications. The applications you install will depend on a few things: how many templates you wish to manage, any application licensing considerations you may have, and your breakout of user groups. The ideal situation is to have a single template, but in large organizations or groups of users with diverse application sets, this is often just not possible.

How many Templates do I need?

Before we get to optimizing the Template, let's talk template strategy. The number of templates you need to maintain will vary depending on your organization structure and application diversity. For example, there are educational establishments that operate their entire school from a single template for staff and students. Many of the healthcare organizations we deploy have a core set of applications that they bake into a single master Template and then branch this for specific departments who have unique application requirements. There may be specific instances that warrant multiple Templates as well. For example, some of the more legacy applications are licensed on a 'per machine' basis, so to avoid being charged a license fee for every desktop, deploying this into a separate template and delivering this template to a certain Active Directory User Group makes far more sense.

A good template strategy is essential for the sanity of the administrator running the VDI environment.

We have mentioned that you could potentially run Stateless and Persistent desktops off the same Template. However, there are subtle differences to consider.

When deploying Persistent desktops keeping the Template generic enough to support your users and allowing them to install the one-off or unique applications should be the approach. When implementing Persistent desktops, the goal is to reduce the overall management of the desktop – make it simple to deploy a large number of desktops automatically. Once deployed, the management of persistent desktops is akin to traditional physical desktops requiring updates and desktop support.

When deploying Stateless desktops, it is essential to cover as many of the applications and settings within a single Template as possible. This keeps the management overhead to a minimum for the stateless pools, only requiring a Template update when patches and applications need to be added.

Optimizing a Windows 10 Desktop

Now that you have a thorough understanding of why we create templates, how to build them, and the various aspects to think about when managing them, we will now switch to the optimization of the Template and resulting desktops deployed from it. Don't underestimate how important it is to get this right – The success of your VDI deployment depends on it. Getting this wrong can mean wasted resources, specifically CPU, end-users getting desktop sessions that are slow to re-draw screens and worse – running out of memory when deploying in-memory desktops causing all desktops to stop on that server. The rest of this paper will walk you through how a number of our customers have recently deployed Windows 10 for their Hive Fabric Stateless VDI deployments. Irrespective of the chosen optimization tool, it's crucial to understand what the tool does to your base install and how this may affect your applications.

There isn't such a thing as a generic optimization tool. Windows has changed dramatically in the last few years, so when you look for ways to optimize your Template, make sure it's compatible with your version of Windows, and has been tested with your version of Windows. A customer was recently bitten by this when they optimized their Windows 10 1903 Template with a tool designed for Windows 10 1803. They quickly found this out when the taskbar didn't appear properly and explorer crashed.

There are also additional considerations for specific types of applications, and HiveIO has started to maintain a best practice guide in our knowledge base and will update the KB and this document regularly to include the very latest in how to get the most out of your Template.

At HiveIO, we recommend using the [PowerShell optimization tool](#) from the VDI Guys, written by Robert Smith and Tim Muessig, two Windows experts working for Microsoft Premier Services. The VDI Guys were one of the first teams to take a lengthy document from Microsoft and turn it into a set of scripts that can run on top of a Windows 10 Install. They have the very latest information on best practice for optimizing Windows and have taken the time to create this tool and share it with the industry. We love to see this kind of information sharing in our industry. It's simple to run and has versions to support most of the production deployments of Windows 10. We recommend using this as the starting point for optimizing your VDI template.

The VDI Guys Windows Optimization.

The following focuses on Windows 10 Version 1909, however, the concepts will broadly apply to previous versions.

https://github.com/TheVDIGuys/Windows_10_VDI_Optimize

The script optimizes the following areas of the Windows 10 OS:

Remove Default Applications and Packages

- Windows Media Player, and any packages associated with Windows Media Player are uninstalled. This is handled by a separate section in the main script to keep Windows Media Player comment outlines 46 – 51
- The OneDrive setup is also removed, to keep this comment out lines 98 and 101 - 112 in Win10_1909_VDI_Optimize.ps1
- Applications and packages listed in [Win10_1909_AppxPackages.txt](#) are removed from the system and for all users.
- We tend to leave the Microsoft Calculator application for end-users as this is often useful. To do this, remove Microsoft.WindowsCalculator from Win10_1909_AppxPackages.txt.
- By default, the Microsoft Store application is not removed, it is possible to uninstall, but almost impossible to add back and removing it invalidates Microsoft support. The simplest way to disable the Store is by using Group Policy: open the Local Group Policy Editor and go to:

Computer Configuration -> Administrative Templates -> Windows Components -> Store

In the Settings pane on the right, double click **Turn off Store application**, select Enabled in the properties page for the policy, and click **OK**.

Turns of scheduled tasks

- We advise leaving this list as per the default in [Win10_1909_SchTaskList.txt](#)

Set Default User Settings

- The script loads the default user registry hive from C:\Users\Default\NTUSER.DAT and applies the settings listed in [Win10_1909_DefaultUserSettings.txt](#)

Disable Auto Logging

- An autologger is a trace session that records events from user-mode and kernel-mode trace providers during the boot process. This feature lets you trace the actions of a trace provider while Windows is booting
- We advise leaving this list as per the default in [Win10_1909_ServicesAutologgersDisable.txt](#)

Disable Windows Services

- All of the services in the list [Win10_1909_ServicesDisable.txt](#) are set to 'Disabled'.
- **Note:** If you plan to leverage Data Protection or User Volumes from HiveIO, then you need to remove the Volume Shadow Copy Service, or the 'VSS' entry in the services file above from the services list. Ensure the VSS service is Enabled and set to Automatic in your Template.

Disk Cleanup, Final Optimizations, and Cleanup

- This part of the script cleans the installation of any temporary files, installer packages, and other remnants on the hard drive leftover that bloats the Template.
- cleanmgr.exe cleans the hard drive, and the actions it carries out are determined by the registry settings in [Win10_1909_DiskCleanRegSettings.txt](#)
- In addition, a number of optimizations are made for Windows File Sharing.
- Note: One of the final optimizations in the script is a specific network optimization for Hyper-V. If you are optimizing your Template for Hive Fabric then you should comment this out as it has no effect. Update line 212 to the following:

```
#Set-NetAdapterAdvancedProperty -DisplayName "Send Buffer Size" -DisplayValue 4MB
```

- Finally, the script finishes by removing all of the following types of files: *.tmp, *.etl, *.evtx and anything in windir\Temp* and TEMP*
- Once complete, the script will ask you to reboot the machine; this must be carried out.

Testing your Template

Once you have created your Template, it's critical to understand how it behaves when you deploy it. Testing should be carried out in an environment that mimics your production environment if you have no option other than to roll it into production then ensure you keep an eye on its behavior. Be prepared to roll it back out in the event of any issues.

A few key things to look out for:

Delta disk growth – The growth of your delta disk is especially significant to monitor when using RAM for storage, as sudden unexpected growth can cause delta disks to fill the storage and cause you to run out of storage. Our target for a delta disk is ~500MB for an idle VM, once the user logs in and starts to use the VM the target is <2GB. If the delta disk grows much above this then something has been missed in the template. Tracking down what is causing this bloat can be challenging. Still, the typical candidates are Windows update, Application updates, or a misconfigured user profile – One of our customers recently triggered this by leaving Windows Update enabled in the VM so each time a stateless desktop rebuilds the Windows Update re-occurs causing increased bandwidth utilization from the download and increased growth of the Delta disk from the differences over the Template.

Increased CPU usage – This can affect your overall VDI density per host. If you start to see host-level CPU alerts after rolling out a new template, then it's probably a change in the Template that has caused this increased CPU utilization. Log into a few VMs and understand what is using the CPU inside the VM. Either update the Template or adjust the CPU assigned to the desktop pool to allow for the increased CPU utilization in the guest. On occasion you have to accept that certain applications need more resources.

Application specific optimizations

Often there are certain considerations for specific applications when deploying them into VDI. This section will be updated as we come across some of the more challenging applications or scenarios that customers have experienced

High CPU Utilization in Chrome – After updating the Template, all VMs started to show high CPU utilization after users had logged in and were using them for a couple of days. The CPU utilization was due to the customer changing their default home page in Chrome to their corporate website, which had a large video playing. Users opened tabs in Chrome but didn't use them, leaving the video playing across the environment.

Solution: Replace the home page in Chrome with a static page i.e. no videos.

Sophos Endpoint Protection reports duplicate VMs. When installing the Sophos Endpoint Protection agent to a Template, it goes through its normal process and registers an ID with the Sophos management server. When the Template is used to create a pool of VDI desktops, they have this agent built-in so each desktop registers with the same ID causing issues with duplicate entries in the management tool all referencing the same VM. This makes management of the protection platform very difficult. Also, note it is important to make sure that the Tamper Protection password is correct in the script you run, otherwise, the script in the Template will not run correctly.

Solution: <https://community.sophos.com/kb/en-us/120560#Manual>

Server-Client Based Applications: As in the Sophos example above, it is not uncommon for any server-client based application where the client auto-registers with a central server to require specific procedures in order to work within a VDI environment.

Solution: Most application vendors have specific steps on their website detailing how to create the Template for VDI.

Microsoft Windows Updates

Sometimes, one of the hardest challenges is the balance between security and delta disk growth – while allowing Windows and the installed applications to update automatically is simple, it will result in substantial delta disks that may cause you to run out of storage. Windows updates tend to be significant and have more of an impact than individual application updates.

As a result of this growth, it is important to consider how you are going to manage the process of updating Windows itself. Once you have optimized the Template, the service will be disabled by default. It is important to remember to regularly boot the Template up and turn the service back on, let Windows update and then turn the service off again and shut down the Template. This approach is good practice to consider for any application that has a centralized update service (on-premise or cloud-based.) Automatic updates will increase the size of the delta disk and cause unnecessary resource utilization. This will consume capacity and, more importantly, consider that every time a Stateless Virtual Desktop is recreated, the VDI goes through the update process, resulting in wasted compute cycles repeating the update process in every VM, each time that VM is rebuilt.

The key here is to balance how often you update the Template with the services that remain enabled to allow the auto-update of applications. Our recommendation is to manage Windows update via Group Policy. This gives you control of the settings inside the VDI environment without having to bake them into your template. Setting GPOs allows you to change settings 'on the fly' without having to edit your template every time. Ultimately this shortens the time it takes to roll out a new setting or control. If you need to do this inside your Template, then you can turn the Windows Update Service on and off using the following PowerShell commands:

To disable Windows update:

```
sc.exe config wuauserv start=disabled
sc.exe stop wuauserv
sc.exe query wuauserv
REG.exe QUERY HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wuauserv /v Start
```

To enable Windows update:

```
sc.exe config wuauserv start=auto
sc.exe start wuauserv
sc.exe query wuauserv
REG.exe QUERY HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wuauserv /v Start
```

Cleaning up after your template update:

Following a Windows update of your template, make certain that you run Disk Cleanup prior to rolling the template into production. The application update process may leave behind remnant files and roll-back features that increase the capacity usage of the template. When running the Disk Cleanup, make certain you choose the option to Clean up system files. This will remove the update files left behind from the Windows update process.

Additional Optimizations to Consider

In addition to running the optimization tool inside your Template there are a few other optimizations to consider.

Hibernation

Disable Hibernation – It does not make sense for VMs to have a hibernation mode in VDI. Hibernation is typically used to conserve power usage, however, with VDI the virtualization platform deals with the power management of the VM. In Hive Fabric, if Hibernation is enabled then you won't be able to successfully validate the template. Validation is the process that happens automatically when you have finished authoring the template to ensure the template is in a usable state. A message is displayed in the UI to warn you of this. To disable hibernation run the following command, from an elevated command prompt, in your Template:

```
powercfg.exe /h off
```

Windows Page File

The Page File is a hotly debated topic in VDI, should you disable it altogether, make it very small, copy what you do on a physical desktop or let Windows manage it. There is at least consensus on the last part – Having Windows manage it doesn't make sense. There is also lots of information dating back more than ten years, which is now outdated. Understanding what the Page File does is key to helping decide how to configure it.

In short, Paging is a memory-management technique to store and retrieve frequently accessed data from secondary storage, i.e., not RAM. The Page File is a collection of pages that may be needed but can't currently fit into RAM. The pages that can be moved from RAM are identified and are pushed to disk. Your disk becomes an extension of the VMs memory. This process helps with reducing lag, increasing application performance, and avoiding system crashes that could occur if the desktop runs out of memory completely.

There are a lot of opinions out there on how to configure the Page File, but the one person who has more knowledge about the internals of Windows than most people I know is [Mark Russinovich](#). He has written about Windows Internals and was the creator of [SysInternals](#), which was acquired by Microsoft. Mark is now a technical fellow at Microsoft so we believe he is a trusted resource on this topic. Here is his advice:

“To optimally size your paging file you should start all the applications you run at the same time, load typical data sets, and then note the commit charge peak (or look at this value after a period of time where you know maximum load was attained). Set the paging file minimum to be that value minus the amount of RAM in your system (if the value is negative, pick a minimum size to permit the kind of crash dump you are configured for). If you want to have some breathing room for potentially large commit demands, set the maximum to double that number.”

As with many things in VDI, this may still not be the right answer. For Stateless desktops that often refresh or VDI machines that are automatically rebooted on a regular basis, running out of memory in the VM is unlikely. For this situation, it would be best to set a small page file ~200mb – ‘just in case’. However, if you are running the VMs in RAM based storage, then disable it entirely; otherwise, you are merely going to use more RAM.

For Persistent desktops, it makes sense to go through the process that is described above and size the Page File appropriately for your environment. Monitor memory utilization and continue to revisit this on a regular basis, as it can have a substantial effect on your application performance and, as a result, your end-user experience.

Windows Desktop Performance

Review the performance settings in Windows (Control Panel -> System -> Advanced System Settings -> Performance), many of these might have been updated from the optimization script. Setting the Desktop Performance to “best performance” will provide your end users with the best end user experience but may cause a change in behavior that your users are not expecting. You should always balance the end-user experience with setting ‘best performance’ but your end-users will generally see a snappier desktop. We recommend you set to best performance and then work with your pilot users to get the balance correct. These are easy settings to update via Group Policy Optimization (GPO) or in the Template directly.

OK, you have the Template set up just as you like...now what?

Investing time in getting the Template right is essential, although you can continue to evolve your processes as you roll out VDI and gain experience in running a production environment. Templates need to be regularly maintained – apply the latest Windows Updates, application updates etc. How often will come down to the number of updates, their size and balancing the impact this has on Delta disk growth vs the time invested to update.

The Template is important, but also how you track, monitor, and optimize your VDI estate is just as important, but that's a whitepaper for another day.

Conclusion

We will see “make things simple” become the mantra of this decade, and VDI will not be left out this time. Taking the complex architectures of VDI and driving out the multiple points of management and unnecessary bloat is vital. Hive Fabric is a much-needed departure from the traditional VDI solution, reducing the many layers of management and operational complexities into a single software layer that can be installed on any bare metal server. With Hive Fabric, the hardest part of VDI is no longer setting up the software itself. We have made that simple. Now the key to success is focusing on your Template and getting it right for your unique company.

The Template, and the strategy for managing it forms an integral part of your VDI deployment. Get it right, and your VDI administrators will have a much easier life, your security team will be happy as the Template is kept up to date regularly, and your end-users will have a seamless experience.

About the Authors:

This whitepaper provides many considerations for how to optimize a Template and hopefully also provides some much-needed insight into WHY we make certain decisions in building a Template. Have suggestions or additions? Feel free to [contact us](#).

Greg Dietrich – Greg is the Global Solution Engineering Manager at HiveIO and has been deeply entrenched in VDI architecture and performance tuning over the last 5 years. With a lengthy career as a systems administrator deeply rooted in Microsoft servers, desktops, and domain services, Greg has been working directly with customers for the last decade evaluating and optimizing their overall IT strategy.

Toby Coleridge – Toby runs the product and support organizations for HiveIO and has been involved in Virtualization and VDI for more than 15 years, he has broad experience deploying some of the largest VDI solutions globally and building industry leading cloud and virtualization products.