
Data Center Automation



Trusted partner for your Digital Journey

Atos



| | |
|----|---|
| 04 | Executive Summary |
| 05 | What is IT Automation |
| 05 | Scripting is an art not a chore |
| 05 | PowerShell |
| 06 | Python |
| 06 | JavaScript |
| 06 | Bash |
| 06 | Infrastructure as code |
| 07 | New-fangled Automation |
| 08 | API should be your first question |
| 09 | Automation in Datacenter Transformation |
| 09 | Automating ITSM |
| 09 | CMP |
| 09 | Automating Operations |
| 09 | Automating Continuity |
| 10 | Overcoming Cultural Diversity |
| 11 | What's in it for you? |
| 12 | Traditional IT Management |
| 12 | Automated IT Management |
| 13 | Mitigating Risk |
| 13 | Technology |
| 13 | Keep an eye on beginner activities |
| 14 | Steps to Success |
| 15 | About the Author |

Executive Summary

This paper discusses the idea that nearly everything in your datacentre will be automated in some fashion in the coming years. We'll discuss some of the different facets of IT automation and how they apply to your day-to-day datacentre or cloud management. As an IT manager or CIO, by the end of this paper you should be convinced that automation is the way forward, especially if you are in a large enterprise. You may however, encounter some resistance from within your business. The said people in this context are those people opposed to change. We present some techniques or concepts that may help you overcome the legacy mindset that humans should be in control of everything.

Without automation, your IT can be slow in responding to growth and market demand. Examples we have seen in the field include provisioning new workloads, and you may have to go through painful manual on-boarding of services. Tasks may include the manual inclusion of workloads into Active Directory, application deployments, Security configuration, Backup setup and CMDB updating. In some extreme cases, some clients wait weeks for new IT services to be enabled. This paper should kick-start your journey to tackle some of these pain points.

As you'll see, before we reach nirvana, there may be some bumps in the road, but let's start with a quote from Bruce Lee to get your mind wondering. He was quoted to say "Empty your mind. Be formless, shapeless like water. Now you put water into a cup, it becomes the cup. You put water into a bottle, it becomes the bottle. You put water in a teapot, it becomes the teapot. Now water can flow or it can crash. Be water my friend."

Bruce Lee? He was a Kung Fu expert who rose to fame in the 60s for being the best in his field. In fact, he was the best in all facets of Martial Arts. He broke the mould by being able to adapt his style to take on any other style of Martial Art. His actions were frowned upon by the traditionalists, but the facts are the facts. He was the best of the best. He was unbeatable.

What does this have to do with technology and automation?
Well, draw some parallels between your technology journey and his philosophy.

Take this simple message with you and learn to adapt to modern methods and technologies or fall behind the crowd. Be Agile! Adapt your business through small iterations. That is the message Bruce brings.

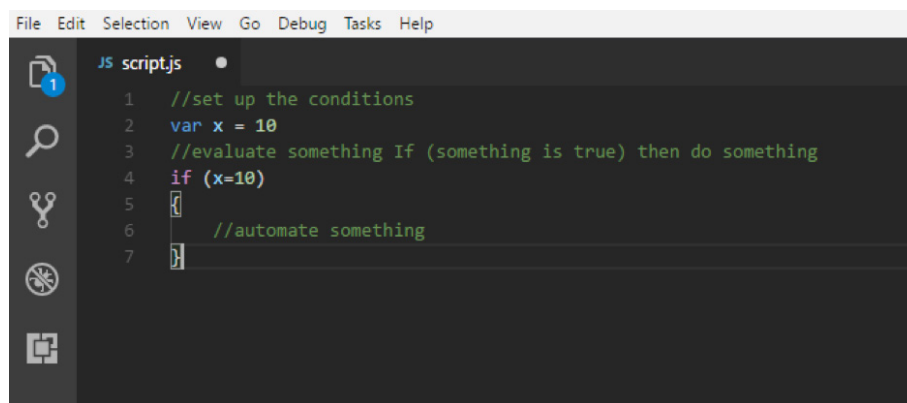
What is IT Automation

Today you may not realise it, but some things are automated in your datacentre or cloud. Whether it's the movement of virtual machines or the automatic backup of your data something is being automated in your datacentre. How can you enhance or add to the transparent automation that is going on behind the scene?

The principal of IT automation is to use scripts, workflows, orchestration and any other tool necessary to automate the day-to-day operations normally conducted manually by a human. The next chapter discusses what kind of operations can be automated, what tools and how to automate them. For now, let's continue to discuss the basic methods of IT automation.

Scripting is an art not a chore

Automation Scripting is a process of utilising a programming language that was designed to allow you to create a step by step set of instructions to do something in the form of what we call a script. Scripts are stored in a readable text files. The intention is not to build a full-blown application but a recipe to perform numerous steps that lead to a single action. The intention of this paper is not to go too technical and describe how scripting works, but to give you a taste of why you need it (if you don't already script for automation in some fashion).

A screenshot of a code editor window titled 'JS script.js'. The editor shows a JavaScript script with the following code:

```
1 //set up the conditions
2 var x = 10
3 //evaluate something If (something is true) then do something
4 if (x=10)
5 {
6     //automate something
7 }
```

The editor has a dark theme and a sidebar on the left with icons for file operations, search, and other functions.

In the diagram above we demonstrate the most basic of scripts. The reality is so much more extensible. With a script, you apply logic to what you are trying to achieve. Without the logic the possibilities of what can trigger automation are limited. In a script, you need to be able to test a condition which helps your script evaluate whether to automate something or not. The condition could be a trigger, like a specific time or a problem found, or a complex query in the form of a regular expression. The action could be a whole sequence of actions to automate something like a fix of a problem or change or deployment. The options are endless. It's a piece of art to scripters.

Often initiated by a higher level scheduler, the scripts are executed based on some condition like a specific time or an environmental parameter outside of the script. Scripting often provides the most flexible way to automate things because you can develop the logic to automate an action the way you want it to flow without constraints of what the features of, let us say, a workflow engine or orchestrator. The more common scripting languages include PowerShell, Python, JavaScript, and Bash to name a few. Let's look at these in a bit more detail.

PowerShell

PowerShell was developed by Microsoft and saw daylight around 2006 and in 2016 was made open-source. The premise of PowerShell is tap into the .NET Framework which forms the basis of every Windows operating system to automate oodles of potential tasks and applications. .NET framework is core collection of libraries that enable a way for developers to write applications to work with the Windows OS. The .NET framework abstracts away the need to have to develop your own code to handle low-level operations like handling the file-system or user interface. PowerShell simplifies the Windows automation by exposing the programmable libraries. PowerShell also provides core fundamental programming constructs like object-oriented programming which also makes it a very powerful scripting language.

This idea is not limited to Windows. Because it's built on top of .NET, anyone developing using this framework can build modules for PowerShell. The means you can automate Infrastructure, Applications, Cloud DR you name it. This idea of adaptable modules is not unique to PowerShell. In fact, the same principle applies to other scripting languages we will discuss in this paper.

What is IT Automation

Python

Python is an open-source scripting language that works in a similar approach to PowerShell, but doesn't feed into or from the .NET framework. First appearing in the mid-90s, is still very powerful and has features like object-oriented programming. The key difference is its cross-platform approach, and a lot of Linux Admins prefer Python because it will run on just about any device and operating system. Python has a large community of contributors, and as such there are hundreds of frameworks that have emerged to simplify the use of Python. If you can think of a problem, there is a chance there is a framework for that. So if you have many different environments and devices that you need to automate, Python might be the one for you.

NOTE: A Framework in this context is a modification of the core product to enhance the way the scripting language can work. If something complex is repeatable, then modification of the core language allows communities to be abstract the complex repeatable code so that others can easily use it.

JavaScript

JavaScript has slightly different beginnings than the previous scripting languages we mentioned. As with Python, JavaScript first appeared in the mid-90s, but was designed to enhance HTML and CSS to make more interactive Web pages on the internet. So why are we discussing it here? On your travels to automate your world, you may come across different orchestration engines that use JavaScript to provide logic for automating IT that makes said orchestration engine more extensible. Also as with Python, there are a plethora of frameworks that can make JavaScript easy and powerful to use.

Bash

Bash is a Unix/Linux command line language first released in 1989 that can be used to execute commands, be used in regular expressions, build a pipeline for logic and numerous other scripting features you'd expect. While originally designed as the command-line environment you can, of course, save the commands execute into a file which transforms it into a script. Popular as a scripting language by Unix and Linux admins because its seen as the stock native command-line in those operating systems.

Infrastructure as code

Infrastructure as code has been building momentum for the last decade and is very popular as the core automation of datacentres. But what is Infrastructure as code (IaC)? IaC is a mechanism for automating just about anything in your datacentre by producing a recipe of desired states. Let's look at this word recipe because it's a good analogy. When cooking food, the ingredients are something you take for granted. You don't need to make the flour yourself. You don't even need to know how the flour is made, it just comes a bag ready to use. Well, that idea is used by the mainstream IaC tool vendors, like Redhat (Ansible), Puppet and Chef.

When it comes to cooking, the less you need to do yourself the simpler things are. That's the same with automation. In traditional scripting, if you wanted to automatically install a piece of software on a virtual machine, you may have to write in code every step to produce a script that resulted in a fully installed application. With Infrastructure as code you take a "make it so" approach, and the rest is handled by the IaC.

We call this the "Magic Source" because the complexity is abstracted.

This abstraction is extended by the tooling allowing for 3rd-party plugins or modules. These components are usually developed by the open-source community which means there are all manner of convenient modules. If you want to automate SQL we have a module for that. If you want to automate operating system deployment on physical servers (bare-metal) we have a module for that. If you want to automate the network we have a module for that. You get the idea. Because Infrastructure as code tools tend to be extensible by 3rd-party modules the options and features you can automate are endless.

It doesn't stop there though. Another main use case for IaC tools is "state configuration". It's great that we change the state of things like software or devices using this automation, but how do we ensure they stay in that state? Things have a habit of shifting. People change things or environmental parameters can cause change. Maybe you want to make

sure that when you automatically changed a network port that it stays like that and nothing should affect that change. Because IaC uses this magic source, this "make it so" approach which ultimately changes the state of something through automation. As a result you can expect the same tool to monitor if that state change has drifted. If a drift does occur and you had to provide the tool with a recipe, then the tool can quite easily apply that change again to make sure whatever you automated in the first place is not out of drift.

Switching use cases slightly, at Atos we started to embrace IaC to speed up migrations of datacentres as well. Below is a table that demonstrates the person-hours required to migrate a client with or without Infrastructure as code. In this scenario, the traditional lift & shift process was not sufficient, and re-platforming of the client's applications and data was required. This means redeploying applications on fresh operating systems and moving the data across opposed to doing a bit-by-bit copy of everything.

What is IT Automation

Manual Execution Scenario

| | Pre-Migration | Post-Migration | Person Hours | |
|-------------|---------------|----------------|--------------|--|
| Time (mins) | 60 | 60 | | |
| Server | 2000 | 2000 | | |
| Total Hours | 2000 | 2000 | 4000 | |

Without automating using IaC migration would have taken 4000 person-hours. Whereas with IaC, it would take, at most 402 person-hours. A 90% reduction in person-hours to manage the whole migration.

We see Infrastructure as code as one the main automation technologies most datacentres will have in production. Organisations may invest in a multi-vendor approach. You may find that one tool does something better than the next and vice versa.

Automated Execution Scenario with Ansible

| | Pre-Migration | Post-Migration | Person Hours | |
|-------------|---------------|----------------|--------------|---|
| Time (mins) | 60 | 60 | | |
| Server | 2000 | 2000 | | |
| Total Hours | 1 | 1 | 2 | If all executions are OK |
| | | | 400 | Time to Fix (Considering there would be some failure job) |

New-fangled Automation

This paper intends to kick-start your journey in automating your datacentre. The focus is not to overwhelm you with a million and one technologies for automation. Still, we believe it's appropriate to discuss emerging automation tech loosely. The three trends in automation you might want to look out for later down the line include:

- **Robotic Process automation**
- **Cognitive Automation**
- **Bots**

In fact that you may even come across a tool vendor that tells you their tool does all of these things. But what are they? For now, know this:

Robotic Process automation aka RPA is technology that allows you to automate things that don't usually have an interface for automation. If you can imagine an application that cannot be scripted or doesn't have an API, but has in the middle of the user interface a big button labelled "GO". How can you programmatically push that button? Well, that is the magic source of RPA. It's software that allows you to automate button press or click on the screen.

Cognitive Automation often connected with RPA or uses RPA to automate stuff, but the differentiating principle behind it is that it uses analytics to help automate actions based on trend for example. Not quite AI but in the realms of AI. Some vendors claim to use machine learning to help spot trends, and from this trend detection, you can take action. For example, if you could monitor the efficiency of your datacentre and predict failure or capacity overflow. This data would help you plan better.

Bots are code that uses a set of algorithms to produce repetitive tasks. Think of bots as a one-trick pony, doesn't do everything but does one thing well. You could get a bot to pull down data from a website without manually having to do it, automatically create tickets on your service tools. Bots tend to be used in the e-commerce world but have an obvious place in the datacentre too.

Section Summary: IT Automation is not just one thing to consider. There are many facets and many technologies that could help you enhance your IT operations, make them run smoother, and faster. The easiest to realise is scripted automation. Low cost, low impact there will always be something that can be automated. Spend time assessing what could be automated and how you should invest in technology to automate more things.

— API should be your first question

Another key take away from this paper is you should always ask this question first when buying technology for your datacentre: Does it have an API? Whether it's hardware or software, does it have an API? API stands for Application Programming Interface and allows you to use your code or scripts to control, manage or automate said technology. Without it then you have to be creative if you need to automate aspects of the technology. For example, you may have to create a convoluted process in an RPA product to automate the un-automatable. Let's use an analogy. You purchase a car without a key. If you hotwire the car, you can still drive it, but it would be better if you had the key.

APIs will provide you with a way of executing features in the technology from an external source. This mechanism allows for easy integration between other technologies too.

Even your automation tool should have an API. You may need to automate the automation. For example, you have an automation workflow that is executed by a high-level scheduler.

Automation in Datacenter Transformation

Let's focus on what kind of things can be automated in your datacentre. This is from the perspective that you have invested in a private cloud in your datacentre. Not limited to this idea as you could apply these concepts to traditional datacentres too.

Automating ITSM

You may be an IT service management driven organisation. The ITSM layer in your datacentre is a layer of technology dedicated to help you organise, structure, plan, design and operate your IT services. You'll often find components like a CMDB, Ticketing and a Portal in the ITSM Layer. Imagine this: instead of manually logging change in the CMDB or manually producing a service ticket you could have your automation layer do this for you. So the automation tooling that is doing the automation should be able to feedback into the ITSM layer.

CMP

CMP aka Cloud Management Platform often will sit inside your ITSM layer. CMP products often have two major features:

1. They provide a portal (marketplace), so you can choose service requests from a catalogue like:
 - Deploy a virtual machine
 - Request a change on the network
 - Request more storage
 - The possibilities are endless
2. An orchestration component that can automate components itself or instruct an automation engine to perform automation.

From a datacentre consumption perspective, this is your shop front, your menu, your way of requesting automation without having to touch the automation components.

Automating Operations

You also have the option to automate aspects of your operations. Remember that we can automate the ITSM layer, and operation functions sit there too. So outside of ticketing and so on what we mean by automating operations is we can affect change on the datacentre to fix problems or enhance performance automatically. If you align yourself with a feature-rich tool, you may find that these kinds of operational automation can be done out-of-the-box, meaning its hands-free. You sit back drink your favourite beverage and let the tool do the automation for you.

However, in an Enterprise datacentre change should be logged in the CMDB right? If an operational tool is making a change to optimise your datacentre automatically, then that should be recorded. Well, no matter because now you know you can monitor and automate that change request.

Automating Continuity

When purchasing tech for business continuity, it should have an API. This principle should apply to all your continuity products like disaster recovery or backup. The advantage of being able to automate such components means you can allow a non-admin user control their usage of those technologies without giving them administrative access. Take for example:

- You would like to give your users the ability to request, control and manage backup schedules for their respective virtual machines without giving them access to the backup software
- It's the same principle for Disaster Recovery, give the user control for fail-over of mission-critical apps without giving them the keys to the Ferrari.

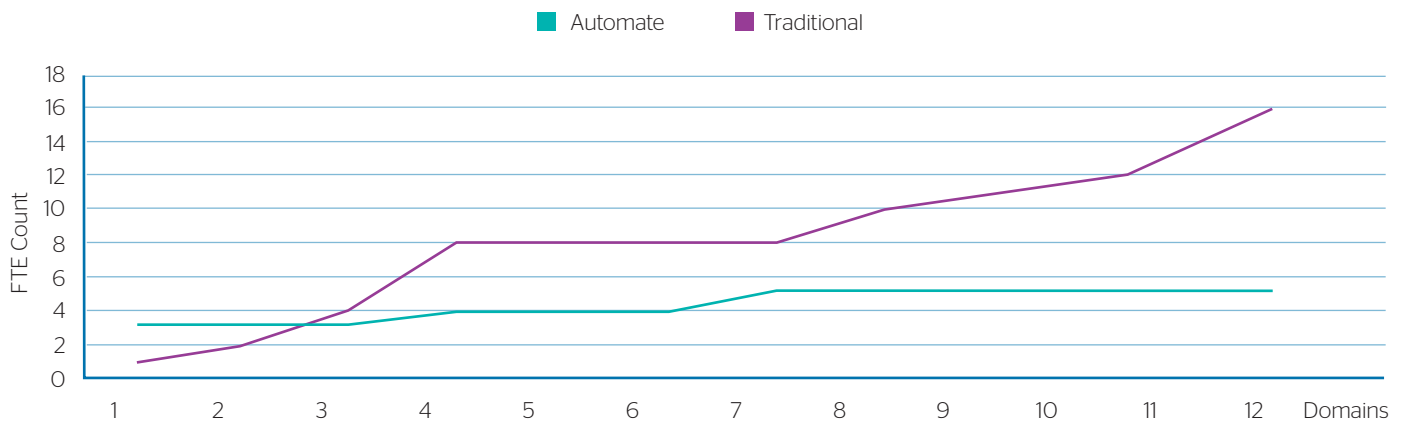
Section Summary: In your Datacentre try to automate everything you can. But start small. The obvious suspects that could be automated and enhance your day-2-day IT operations, could include how your services management framework (ITSM) is automated so that that service requests and change control or handled by automation. This is a must for any large organisation. Also, integration into functions like business continuity is also a good starting point. Consider adding a layer of automation to remove some of the stress of managing things like backup processes.

What's in it for you?

From an operations perspective, the number of resources or FTEs (Full-Time Equivalents / someone doing tasks) required to manage your IT grows with the amounts of domains (areas of IT) that need to be managed.

There is obviously some truth in the objection that more skills are needed to automate services in the first place, and these skills are often NOT found with the existing IT management. However, we found (demonstrated in the example diagram below) that the FTE count to start with could be greater when automating, and as IT services grow the need for more operational FTEs scales better than the traditional operations.

Automation vs Traditional



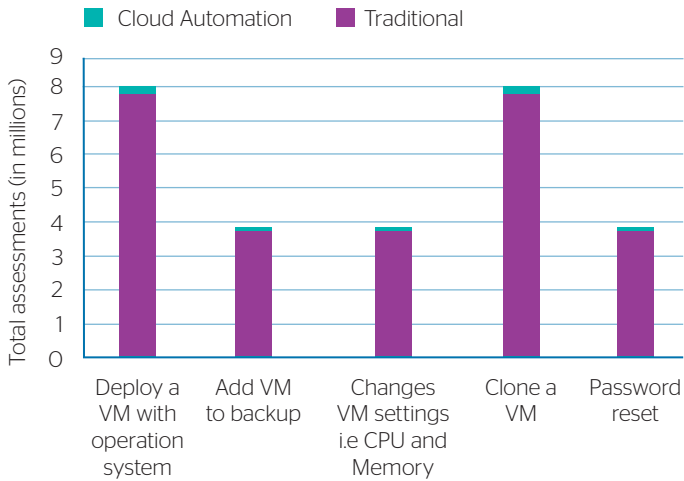
So yes there is a big bang approach, but it's a case of CAPEX vs Long-term Investment. Why the big bang? This is often due to the lack of skill level to automate IT in the first place. It could be a big investment to start with, but you'll reap the rewards later. This is especially true if you take a DevOps approach to maintaining the automation.

Let's look a bit deep into why automation should be attractive and demonstrate some ROI (Return on Investment) If we look at some examples of how the effect of automation works in reality, we can eventually see there is high Return on Investment.

If you take some typical day-to-day tasks, you might expect from your IT team or service

provider like deploying a virtual machine or requesting a password reset we can see from a traditional approach typical SLAs (Service Level Agreements). So requesting a new virtual machine can take up to 8 hours (1 business day) whereas if the task was automated the process can be achieved in 10 minutes on average.

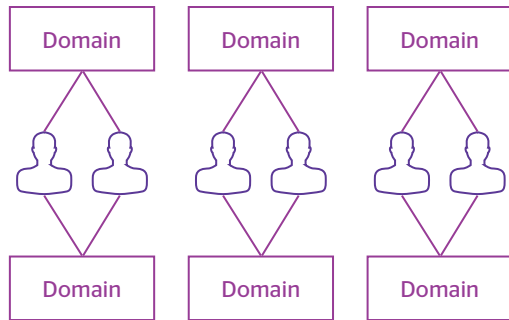
What's in it for you?



| Service | Traditional | Cloud Automation |
|--|-------------|------------------|
| Deploy a VM with operating system | 8 Hours | 10 mins |
| Add VM to backup | 4 Hours | Few seconds |
| Changes VM settings i.e CPU and Memory | 4 Hours | Few seconds |
| Clone a VM | 8 Hours | 10 mins |
| Password reset | 4 Hours | Few seconds |

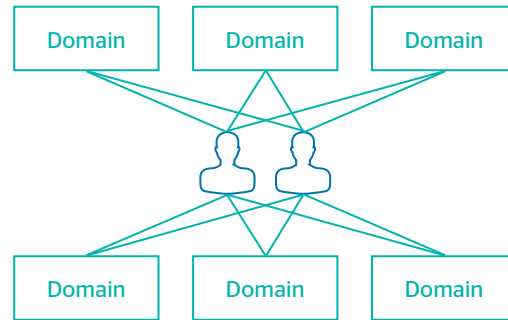
Just from this simple graph, you can see that the ROI is obvious to appreciate. Let's drum this point in. Traditional vs Automated Approach means quicker time to market and fewer FTEs

Traditional IT Management



- Operations FTE's can be shared between domains (where skills are transferable), but scaling is limited
- The amount of FTEs needed to grow as the number of domains grow
- IT costs will stay fixed and will grow no matter how many domains you have

Automated IT Management



- Time invested in automation means less FTE's are needed to operate in the long run
- The amount of FTEs needed stays fixed (minimal increase)
- The more domains you have automated drives down the cost of IT management over time.

What about those "job protection syndrome" people? The answer is to get them involved in the lifecycle process which has the effect of freeing up their time to do other important tasks. We have seen a 40% reduction of FTEs needed to manage an environment due to automation, but that doesn't mean you have to reduce 40% of your IT workforce. It just means they will be able to respond 40% more efficiently.

Section Summary: New skills may be needed to kick-start your automation journey, and a big bang approach could be daunting. You should expect to see a return on investment down the line though. The usual bottle-neck for IT is that lack of resources. If you automate your IT better, this, in the long run, frees up those resources to manage the IT that cannot/should not be automated.

So what is stopping you? You may understand there are examples of performing multiple manual tasks that can be automated but are nervous due to the cost of achieving automation of IT services. IT teams may lack the skills to automate services and reskilling could be costly. Also, building the automation is often much more time consuming initially than performing the manual tasks.

Experience has shown that the approach to building automation by piecing together several contributions from experts with a deep but narrow scope into one bigger workflow is time-consuming and error-prone. A lot of gaps can be found in those workflows and so automation may fail. The best approach is where one automation expert takes responsibility end-to-end and thinks through the whole process. This automation expert gathers most of the needed knowledge oneself, consulting with the deep experts. This will have the effect that will change the self-image of those "job protection syndrome" people. They are no longer OS, database or application experts, but more generalists with a broad, but not so deep skill over many technologies again. After a decade where deep specialisation was the key for a career in IT, HR will have to open career paths with corresponding training curricula that reflect this new skill-demand. You will have to find those people who are willing to work on such a broader scope of technologies again. Now we know the long-term investment should outweigh the initial cost for the cultural change, what other things should you look out for?

Technology

Make sure as mentioned earlier that you align with technology that allows for automated processes. Think about how you can automate your business processes as well as your operational processes and what technology will enable that. From a Datacentre Tooling perspective you should be looking out for keywords like "Software Defined" and "API". If you intend to run IT "cloud-like" on premises, you should invest in a private cloud that gives automated service features. Nearly all aspects of your Datacentre could be automated. Technologies like Hyperconverged Infrastructure, Software Defined Networks and Software Defined Storage all fronted by a good orchestrator will be a very good starting point.

If you are not doing Infrastructure as code, you should be. Given the convenience of IaC (mentioned earlier), this should be the key take away for you. Why reinvent the wheel when someone else has done it for you. Would you build a car from raw materials? Then why should automation be any different?

If you require that non-admin folk have access to automation, then provide them with a portal. Always make the automation look simple when in reality it's probably the most complex thing to manage in your datacentre. Invest in a good CMP as mentioned earlier. After all, you don't want inexperienced people meddling with your automation scripts. They could break the internet. Joking aside, depending what you are automating don't open yourself up to something destructive. If a human can break something so can automation.

DevOps

You're probably asking why is there a section about DevOps in the mitigating risk section? Seems strange right? DevOps is founded on principals like Agile Methodology. Some of the techniques that you may use to develop and release software are transferable to automation scripts and workflows. After all, these are code too. For example, you could host your automation workflows/scripts on an SCM (System Configuration Management) platform like GitHub. This way you can collaborate with colleagues on automation and control versions. Also, you could use a technique like CI/CD to test if your automation for release is ready. Automated testing of automation! Release often and don't fall into the trap of old waterfall techniques of developing and releasing code. This is why DevOps is in this section. After all, automation could be the cornerstone of your time to market. Inhibit that in any way, and you could cause your company lost revenue... "Be water my friend."

Keep an eye on beginner activities

The first activities that are automated are normally the 'simple/easy tasks'. These are often the first tasks you should let beginners do as part of the learning-by-doing approach often used in IT. With these tasks disappearing from daily operations the bar over which new employees must jump when they want to work as an engineer in the delivery, will rise quickly. Therefore, you must find ways to compensate for these 'lost first steps' in their Development. At the same time, you must keep the existing engineers fluent in these basic tasks as the automation might suddenly fail, so that manual execution would be necessary again. Invest in 'simulators/labs' to train these skills, like what pilots are doing to stay fluent in all tasks and keep their license. Actually, that is a good analogy. Pilots will often go back to the simulator to make sure they are doing things correctly or updating for the latest model of aeroplane. Keep your experts keen!

Steps to Success

From a technology perspective, you can see, there is not one size fits all. Our advice is to choose a combination of tools to automate everything you can.

When purchasing systems your first question should be “Can it be automated and does it have an API?” Be prepared to invest in reskilling. You’ll then see a return on investment in the long run. Atos have spent millions of dollars reskilling our staff over the last five years to help our clients provide self-service hybrid, private and public cloud platforms that include day two operational automation.

Do not try to reinvent the wheel. Use existing automation workflows that can be repurposed. So social media and the internet to see if someone already created what you are trying achieve.

Summarising what you should look out for and need:

- **Invest in scripting skills**
- **Invest in Infrastructure as code to speed up automation**
- **Invest in software and hardware that can be automated like SDDC (Software Defined Datacentre)**

Keep an eye on beginner activities

1. Education and Knowledge. Education and Knowledge transfer are at the forefront of everything we do. Atos Management are all tasked to make sure up to date skills are part of employee MBOs. In addition Atos have put a great deal of focus on key trends in Automation to make sure we are at the forefront of latest innovation too. This knowledge is passed down to help build automation specific to our clients needs so clients realise:

- **Benefit: reduces time-to-market**
- **Benefit: increases innovation and competitive edge**

2. Repository. We have blueprints to automate everything from Network configuration, Resource Requests, Application deployment, CMDB updates, backup scheduling and anything we don’t have out of the box we will develop for you. This reuse of knowledge and work helps:

- **Benefit: reduces cost**
- **Benefit: reduces risk**

3. Efficiency. Making the right choice about your future IT can be daunting. Knowing when to spend on new IT or repurpose legacy technology can be advantages to those looking to maxims their investment. Atos provides first-class consultancy to help you understand your readiness for new technologies and innovation and when to continue using your existing IT and its longevity. This puts you a place that again:

- **Benefit: reduces cost**
- **Benefit: reduces risk**

4. Unlimited Technology Choice. Atos is not limited by technology choices. We propose what is best for our clients when it comes to technology, and will consult using a vendor agnostic approach. This mitigates vendor lock-in, and means that clients will never be inhibited by future decisions.

- **Benefit: increased choice and innovation**

Contact your Atos salesperson:

- **Request a demo**
- **Request a Proof of Technology**
- **Request a quote**
- **Get automating your IT today!**

About the Author

Ricky El-Qasem



A Virtualisation and Cloud Veteran with 25 years under his belt. Ricky designed the first high-level architecture of the first Atos Private Cloud Solution, currently known as DPC. While designing its predecessor he had automation in mind from day one. Ricky has built simple automation engines for VMware environments and now runs several communities inside Atos including the Automation Experts Domain.

About Atos

Atos is a global leader in digital transformation with approximately 100,000 employees in 73 countries and annual revenue of around € 13 billion. European number one in Big Data, Cybersecurity, High Performance Computing and Digital Workplace, the Group provides Cloud services, Infrastructure & Data Management, Business & Platform solutions, as well as transactional services through Worldline, the European leader in the payment industry. With its cutting-edge technologies, digital expertise and industry knowledge, Atos supports the digital transformation of its clients across various business sectors: Defense, Financial Services, Health, Manufacturing, Media, Energy & Utilities, Public sector, Retail, Telecommunications and Transportation. The Group is the Worldwide Information Technology Partner for the Olympic & Paralympic Games and operates under the brands Atos, Atos Consulting, Atos Worldgrid, Bull, Canopy, Unify and Worldline. Atos SE (Societas Europaea) is listed on the CAC40 Paris stock index.

Find out more about us
atos.net

Let's start a discussion together

