



CUSTOMER CASE STUDY

# From on-premises to cloud — fast

Leading energy provider uses HashiCorp Terraform to provision faster, enhance security, and fast-track its move to the cloud.

# About AGL Energy

At AGL, we believe energy makes life better and are passionate about powering the way Australians live. Proudly Australian for 185 years, AGL supplies around 4.3 million<sup>1</sup> energy and telecommunications customer services. AGL is committed to providing our customers simple, fair and accessible essential services as they decarbonise and electrify the way they live, move and work. AGL operates Australia's largest private electricity generation portfolio within the National Electricity Market, comprising coal and gas-fired generation, renewable energy sources such as wind, hydro and solar, batteries and other firming technology, and gas production and storage assets. We are building on our history as one of Australia's leading private investors in renewable energy to now lead the business of transition to a lower emissions, affordable and smart energy future in line with the goals of our Climate Transition Action Plan. We'll continue to innovate in energy and other essential services to enhance the way Australians live, and to help preserve the world around us for future generations.



Deployed new virtual machines in less than 10 minutes



Built more than 1,500 virtual machines



Migrated 210+ apps to Azure



Exited two large datacenters in 24 months



Moved from Terraform Enterprise to Terraform Cloud for advanced collaboration and high availability



Created module library for faster deployment collaboration and high availability

## Reducing business risk required a large-scale, multi-year transformation

As a large organization in business for many years, AGL Energy has seen its technology estate grow and evolve. A few applications were migrated to the cloud by 2017. About two-thirds, however, still ran on company-owned infrastructure in two large third-party datacenters.

According to Dan Hermans, Cloud Engineering Manager at AGL Energy, the company had a problem: one of its third-party datacenters was closing, and the other facility — along with the AGL-owned infrastructure components within it — was aging. To avoid significant business risk, the company needed to either refresh its infrastructure or move everything to the cloud.

After careful analysis, company executives decided to vacate both datacenters and migrate all applications to Microsoft Azure. Given the size and complexity of AGL's technology estate they knew it would be a multi-year effort and formed more than 50 teams made up of employees, contractors, and third-party partners dedicated to the project.

The teams faced many technical challenges. Applications built on legacy 32-bit operating systems had to be transformed before moving to the cloud. There were many old commercial off-the-shelf (COTS) software applications that had to be migrated or replaced — many of them were missing vendor documentation or installation materials; some were from companies that no longer existed.

In addition to application challenges, the team also had to overcome infrastructure issues. Using their standard processes for infrastructure builds, it could take up to a week to deploy a new virtual machine (VM). For such a large-scale project, this would not work. The process was simply too cumbersome and took too long to complete.

AGL knew executing a major migration like this one required automated infrastructure provisioning. To meet the project timeline, the company needed a tool that simplified collaboration, kept cloud environments secure and consistent, and helped teams move quickly.

“ Terraform eliminates many of the governance challenges we had before when we were trying to manage our work through different processes and tools.”

## **Infrastructure as code modules for fast, secure, consistent builds**

A few years prior to the migration, AGL Energy had undertaken an automation initiative. The goal was to automate as many processes as possible, including provisioning infrastructure. It was then that the company first started using HashiCorp Terraform Enterprise.

“The ‘automate everywhere’ initiative was set up to automate landing zone management and build a module library,” said Dan Hermans. “To do that, AGL purchased Terraform Enterprise. But we were not using it at scale.”

### **Infrastructure as code**

As an infrastructure as code tool, Terraform Enterprise helped AGL create configuration files to define infrastructure. This let the company set standards for infrastructure builds, make sure every environment adhered to policies, and automate provisioning.

Knowing what Terraform was capable of doing, the decision was obvious to make it a central tool for the migration teams.

“Cloud information templates can be thousands of lines long,” said Dan Hermans. “If you get something wrong, like forgetting to include a bracket, the whole thing stops working. With Terraform, you can do the same thing in a hundred commented lines that anyone can read. The equivalent Terraform template can be quite succinct and much more readable, much more documentable.”

Hermans added that AGL chose Terraform Enterprise over the community edition because of its enterprise-specific features. “Active Directory linked single sign-on with automatic onboarding and offboarding, the team concept for collaboration, the Sentinel policy system, the module registry, private worker containers, and VCS-linked workspaces were all critical features,” he said.

“ Terraform makes it easy for our developers to work independently and self-serve the infrastructure their projects need to be successful.”

### **Faster provisioning, better security**

Prior to implementing Terraform at scale, deploying infrastructure was a manual process at AGL. Teams would submit a form and wait while the request made its way through the management ranks, getting approvals and cost codes, and eventually assigned to a resource who would build it.

With Terraform, AGL has eliminated the manual process. Configuration files make sure infrastructure is built to tested, approved standards, helping teams provision infrastructure much faster.

“Before Terraform, it would literally take a week to get a VM up and running,” Hermans said. “But now, we can deploy a new VM in less than 10 minutes.”

The automated process has also improved security. According to Hermans, AGL’s old manual process required that security agents were manually installed to protect against vulnerabilities and connect VMs to patching servers. Sometimes this step was forgotten, and machines were at risk until the oversight was noticed.

Terraform has helped AGL reduce this risk. Each new VM build includes the installation of security agents, and policies are enforced using Sentinel. As a result, as soon as a new resource is deployed, it’s compliant from a security perspective.

According to Hermans, these benefits were clearly visible during the datacenter exit project and helped the teams overcome infrastructure challenges. “By publishing tested modules to the registry, third-party providers and partners could deploy quality virtual infrastructure faster,” he said. “The modules came pre-loaded with all required security settings, so our security teams could give fast approvals. In addition, migration teams could review our Git code library, see a vast assortment of infrastructure code for applications that have already been migrated, and use these as a source reference for quicker deployment.”





Being able to provision VMs fast using Terraform helped AGL boost team productivity and complete the datacenter exit program in only two years. “Speed, no question, was a big benefit,” Hermans said. “Without Terraform, there’s no way we would’ve gotten out of the datacenters as quickly and as efficiently as we did.”

## Conclusion

Terraform Enterprise helped AGL Energy complete its datacenter migration, eliminate the need for on-premises infrastructure, and move all its applications to the cloud.

Since the migration, AGL's cloud team is now focused on the next phase of the company's cloud journey. This involves redeploying the "original" cloud deployments (those environments where approximately one-third of its applications were running before the datacenter exit program) using Terraform Cloud and the techniques learned during the migration to secure and improve those applications. In addition, the team is working to enhance their cloud footprint, migrating all their cloud environments again, using Terraform Cloud, to clean up any outliers, optimize cloud resources to reduce costs, and build in more mature access control features.

## Outcomes

-  **Created a module library for faster deployment**
-  **Deployed new VMs in less than 10 minutes**
-  **Moved from Terraform Enterprise to Terraform Cloud for advanced collaboration and high availability**
-  **Migrated 210+ apps to Azure**
-  **Exited two large datacenters in 24 month**

## Solution

Today, AGL Energy runs all of its applications in Microsoft Azure, with the exception of a small footprint in AWS. As part of the company's datacenter exit program, teams built 1,500 VMs and migrated more than 210 applications in approximately two years, eliminating the need for on-premises infrastructure.

## AGL Partner



**Dan Hermans**  
AGL Energy's Cloud  
Engineering Manager

Dan Hermans is currently Cloud Engineering Manager at AGL Energy. A highly experienced Cloud and DevOps Engineer and Leader with a proven track record of delivering large-scale, complex projects in major banks and energy retailers both in Australia and overseas. Skilled in architecting and implementing complex solutions that leverage the latest technologies and best practices to drive efficiency, scalability and security.

## Technology Stack

- **Infrastructure:** Microsoft Azure, AWS, and some onsite hardware at remote sites
- **Workload type:** approx. 1/2 VM-based (due to the nature of the industry) with large AppService environments
- **Container runtime:** Small amount of Kubernetes and container apps
- **CI/CD:** Azure DevOps
- **Storage:** Cloud native with some NetApp and Commvault
- **Version Control:** Git
- **Networking:** Cloud native
- **Provisioning:** Terraform
- **Security management:** Azure Policy





## About HashiCorp

HashiCorp is a leader in multi-cloud infrastructure automation software. The HashiCorp software suite enables organizations to adopt consistent workflows and create a system of record for automating the cloud: infrastructure provisioning, security, networking, and application deployment. HashiCorp's portfolio of products includes Vagrant™, Packer™, Terraform®, Vault™, Consul®, Nomad™, Boundary™, and Waypoint™. HashiCorp offers free community source-available products, enterprise products, and managed cloud services. The company is headquartered in San Francisco, though most HashiCorp employees work remotely, strategically distributed around the globe.

For more information visit [hashicorp.com](https://www.hashicorp.com).