

XENON CLUSTER STACK

Re-Writing the HPC Play Book



Agile, Flexible, Easy ... words not normally associated with high performance computing clusters. Until now! XENON Cluster Stack transforms HPC clusters into modern, flexible, infrastructure-as-code environments that deliver high performance results.

XENON has been building high performance solutions for over 25 years, and the XENON Cluster Stack (XCS) reflects decades of cumulative experience across our team. Customers have been looking for new ways to manage their clusters, to address common issues such as:

- Up-dating software – operating systems, management tools, applications – and ensuring it all works with the latest versions.
- Expanding the cluster with different hardware vendors.
- Implementing new tools or applications – testing and roll-out process is tedious and long.
- Maintenance, and monitoring cluster performance are complex tasks.
- Integrating the cluster into Active Directory, LDAP and other common protocols.

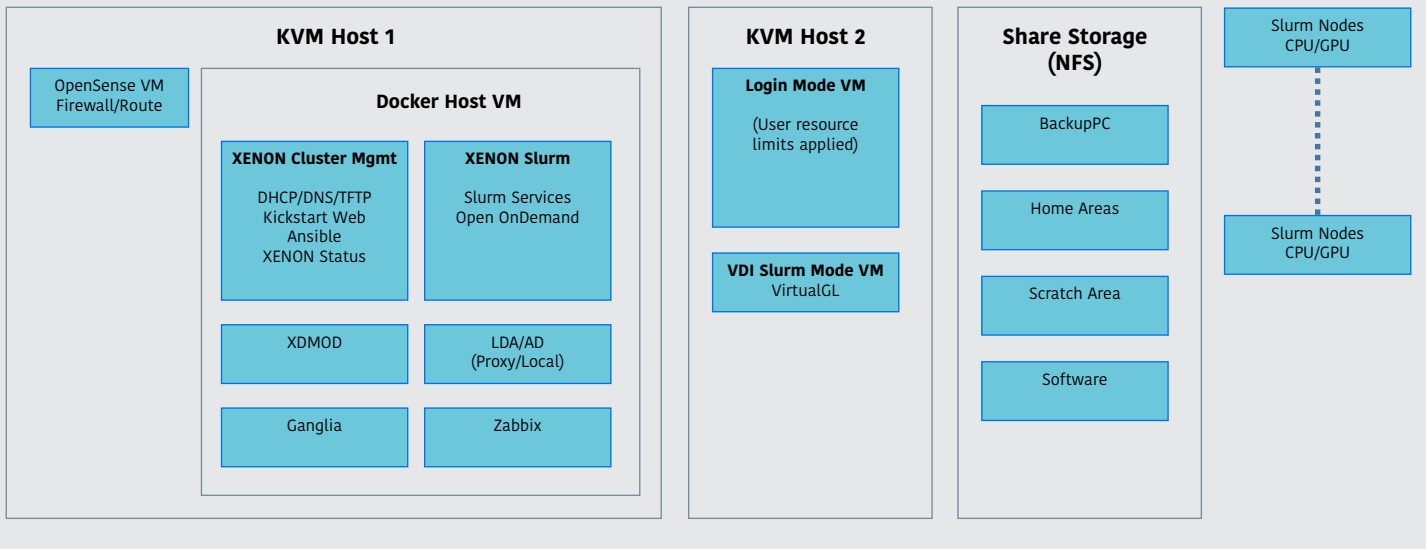
XENON created the XCS to address these issues and to bring HPC cluster management into a modern, containerized framework that is easy to implement. The results are an HPC cluster that is modular, flexible and easy to monitor and manage.

THE XENON CLUSTER STACK

Starting with a blank slate, the XENON team's goal was to create an HPC middleware software stack that has clear advantages in six areas:

- **Cost:** must be cost effective while being commercially robust.
- **Easy:** Maintenance, up-dates, monitoring, performance tuning.
- **Flexible:** Ability to replace individual components or the whole stack; use variety of common job management solutions; run variety of applications and even web collaboration tools.
- **Modular:** Easy to expand; hardware agnostic; cloud and on-prem agnostic.
- **Protected:** Ability to implement standard backups, snapshots, and ability to test various restore points and restore time objectives.
- **Secure:** Components and the environment; integrate with common authentication protocols like LDAP, AD.

EXAMPLE COMPONENTS



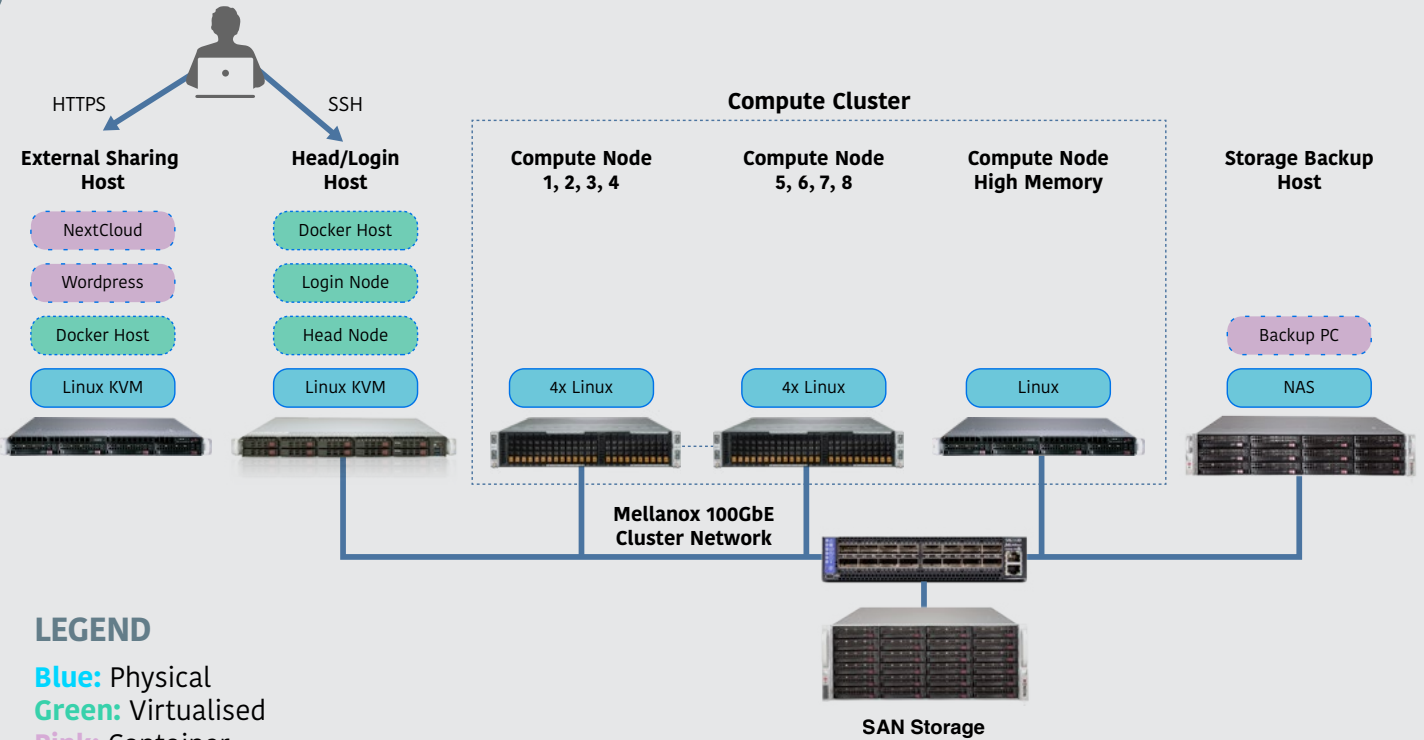
Building with open source tools and containerization, the team has developed the XENON Cluster Stack over the last two years. XCS is built on GNU/Linux OS and KVM/XEN virtualization. Open source projects are used for all core components, Docker is used to isolate various tools and services while Singularity supports containerized user applications. This approach delivers a very cost effective middleware cluster stack, while thoughtful deployment of virtualization and containers allows for a modular, flexible architecture.

FIELD TESTED

XENON has now deployed XCS in a variety of cluster environments, with a range of hardware, applications and workloads.

The example next page is a real-world deployment on a medium sized cluster used in academic environmental simulations and modelling.

CLUSTER DESIGN



PROVEN RESULTS

XCS is delivering tangible results for HPC users and administrators across Australia, New Zealand and even in South East Asia. The XCS approach of containerization in a modular architecture has proven to be flexible, easy to update and expand, and easy to backup and protect.

Deploying XCS across new and existing hardware has delivered a cost effective upgrade path for customers and has allowed for a hardware agnostic cluster. Free of vendor lock-in – at both hardware and middleware level – customers can make the most of existing hardware that has a longer life when abstracted from the software layer with virtualization and containers.

XCS is opening up interesting new opportunities to host external collaborators and web based services. Again, virtualization and containerization allows these services to be deployed in a secure and isolated manner while still accessing the resources needed to operate effectively and securely.

AVAILABLE NOW

XENON Cluster Stack is available from XENON with new HPC clusters, and is also available for existing clusters through XENON Managed Services for HPC Clusters.

[Talk to an HPC Solution Architect Today](#)