XENON eResearch NZ 2026 – Modern Multi-tenancy
SLIDE NOTES

Slide 1

Hi, I'm Ron with XENON Systems. Today I would like to talk about some of our recent experiences working with GPU cloud platform providers to deploy multi-tenancy solutions, and the decisions we've helped clients make about the options available in this space.

Slide 2

During this talk, we'll review what multi-tenancy means and take a look at some typical solutions.

We'll review the range of options available for hard to soft multi-tenancy, and the trade-offs involved as you move along the scale.

I'll run through some interesting large cloud provider examples, and how we've seen storage integrated into these environments.

And finally, we'll cover some of the challenges and lessons learned along the way.

Slide 3

What exactly is multi-tenancy? In general, it's when you have multiple groups of users sharing the same system with a degree of logical isolation from each other.

Sometimes it's multiple users within the same team, collaboratively running workloads that may need to communicate with each other. This is typically what you would see with classic HPC batch systems, which you will find at many universities and supercomputing centres.

Or, you can have multiple teams within the same organisation who operate independently, where it's preferable to have some level of isolation between the teams.

On the other hand, when you have multiple customers sharing the same system, you will want each customer to be strongly isolated from one another, which is the usual case for public facing platforms such as cloud providers.

So, what are our options for implementing these multi-tenancy architectures?

Slide 4

You could simply have dedicated hardware for each tenant, so each tenant is physically isolated from one another. We see this with clients who are seeking large amounts of dedicated infrastructure. They'll work with a partner who can implement the infrastructure as a single tenant, anything from several racks or perhaps a whole data centre all to themselves.

On the other hand, you may have multiple clients in the same environment, with dedicated compute and storage, and a shared network. In that case the systems can be logically isolated on the network level.

Then there's virtualisation, which enables compute to be logically isolated while still sharing the same physical hardware. Typical virtualisation platforms include those based on Linux KVM like OpenStack, Open Nebula, and ProxMox. There's also XEN based options with XenServer and XCP-NG, and there's also VMWare and Hyper-V.

In a similar way, containers can also be used to logically isolate workloads on the same hardware using container engines like Docker or Podman, scaling up to large multi node Kubernetes clusters.

So how do we decide which options are the most suitable and what are the trade offs for each approach?

Slide 5

There's the question of hardware management and utilisation. We could be looking to manage several types of compute node, different compute to RAM ratios, along with different types of GPU. How are these systems divided between tenants? How do we go about, for example, allocating a single GPU out of an eight GPU node to a tenant? Or perhaps a fraction of one GPU? There are solutions available to cater to any of these scenarios, but each comes with its own set of trade offs.

When we look to divide GPUs into fractions, we can increase utilisation by efficiently slotting jobs into available resource gaps. Though, this level of granularity tends to come with a high management overhead. We can consider allocating individual GPUs to a tenant, but like with fractional GPU allocations, we're sharing the same server hardware between multiple tenants.

This means we need to consider how the system is managed, how we can ensure fair quality of service, and system security.

Who's responsible for managing the operating system and security updates? If the provider is aiming to allocate entire nodes to a client, then perhaps the responsibility for node security falls to the client, which would reduce the

platform provider's management overheads. It all depends on the level of abstraction and involvement the platform provider is aiming for.

There's the performance impact of running Virtual Machines on top of bare metal, which is quite low thanks to hardware acceleration, but it's still worth considering.

And for application management, perhaps the platform provider is aiming to support a full-service application deployment stack? If that's so, then how are software dependencies and asset management handled? Who's responsible for securing which components?

Slide 6

When it comes to planning the type of multi-tenancy you want to implement, it can help to think of it as a spectrum, from hard to soft multi-tenancy.

The 'Hard' side of the spectrum is more focused on hardware level controls. Moving from separate physical networks and compute, to shared network hardware with network virtualization and separate compute, through to shared hardware with separate compute virtualization.

Beyond that there's containerization including Kubernetes which can use namespaces and virtual networks to isolate tenants from each other.

Then there's application-level controls, software managed process, workflow allocations, and Software as a Service multi-tenancy platforms where tenant access is managed using identity and role-based access controls.

In general, it comes down to balancing the required features against the management and configuration overheads.

Slide 7

Let's take a closer look at implementing hard multi-tenancy at the network level.

We recommend this approach for larger cloud providers who are looking to onboard multiple clients. In this architecture, there are several different network types to consider:

There's the North-South network which is designed to handle traffic between the compute nodes and the outside world, or any other supporting systems like data storage. This network is typically high speed ethernet and supports RDMA over converged ethernet, known as RoCE, to enable very low latency data transfers to and from locally connected systems.

There's the East-West network, which is designed to provided high speed, very low latency, non-blocking communication between the compute nodes. This network can be high speed ethernet with RoCE support, or perhaps Infiniband, to enable distributed parallel compute jobs that require a large amount of inter-node communication, such as when training a large language model.

And finally, there's the out of band network, which is designed so that cloud providers can connect to all switches, storage and compute nodes to enable easy configuration and maintenance. This network is typically gigabit ethernet, and usually only the platform maintainer has access.

Across the network, traffic is controlled using VLANs, route leaking, ACLs and firewalls, and in the case of Infiniband, the network can be segmented using partition keys.

These networks can be provisioned by manually configuring the switches, semi automatically using config as code systems like ansible, and fully automatically using software defined networking and cloud management tools such as Netris, Aarna or Rafay just to name a few.

The software defined networking solutions are great for keeping up with the configuration churn expected when onboarding tenants, as the network configuration can become quite complex, involving EVPN VXLAN, BGP and a Leaf Spine architecture, with configurations of at least 11 network links per node, 8 for the East West network, 2 to 4 for the North South network, and 1 to 2 for the out of band network.

Configuring the network manually is not recommended, but if you must, then we recommend leaning on tools like Ansible to assist with automating and testing as much as possible. Automation tools like Ansible are key to building a consistently redeployable network, which enables you to use services such as Nvidia Air to build a digital twin of your network so you can test and verify configuration changes before they are deployed live.

If you are using a software defined networking solution, be prepared to build the network configuration from scratch as importing an existing network is usually not recommended or supported.

Slide 8

Let's take a closer at Netris, which is a cloud provider style network automation tool designed to work with Nvidia's Cumulus Spectrum switches.

In the diagram we can see the north-south network above the nodes, which is also known as the TAN or Tenant Access Network

And the east west network below the nodes.

Netris has an API to support integration with other control and monitoring systems that you would see in a typical Nvidia AI Factory deployment such as BCM and NetQ. It also supports Nvidia Air for simulating and testing digital twin environments.

Slide 9

Using Netris really helps manage the large number of links involved, especially in the east-west network which is designed to be non-blocking, therefore requires many more links between the leaves and spines.

Building a non-blocking network with today's link speeds typically means each node has 8 links of 400 gigabits connected to the leaves, which requires 4 links of 800 gigabits from the leaves to the spines.

As you can see with this 32 node example architecture, the number of links adds up quickly.

Slide 10

Moving up the stack, we have Rafay. Rafay provides a service for orchestrating Kubernetes clusters. This service works well when combined with Netris for network level tenant isolation. On top of that you can use Rafay to configure soft tenant isolation at the Kubernetes level.

Users can use the web portal for self service deployment of machine learning pipelines and other applications like Jupyter notebooks. The portal also keeps track of finance and billing information, along with other features that you would expect to see in a GPU cloud platform.

Slide 11

On the left there's a high level system architecture diagram showing how everything can be connected. We can see some nodes are running VMs in Kubernetes. Then there's bare metal deployments, and the Base Command Manager nodes to handle bare metal provisioning.

On the right we can see how Rafay organises tenancies and instances, with instances grouped by workspace, and workspaces grouped by organisation, offering plenty of flexibility.

Slide 12

Then there's the question of data storage which can be an interesting challenge too.

In most cases, you want a large powerful storage system with isolated namespaces per tenant. Network access to those name spaces is enabled using isolated virtual IP address pools to manage traffic at the network level.

There's additional security controls that can be implemented at the protocol level, and integrations with identity providers for each tenant. Other features typically include encryption both in flight and at rest, quotas and quality of service.

There are several storage solutions available that support multi-tenancy. Plenty of details to consider.

Slide 13

So to summarise the challenges, multi-tenancy can be difficult due to the large management overhead and system complexity of sharing network and compute infrastructure.

Manual network configuration becomes unsustainable quickly. Automation can help with this, especially when onboarding tenants and validating configuration changes.

Switching to a software defined network solution will likely require a clean slate.

There's a variety of approaches to implementing multi-tenancy, and many of the approaches can be combined.

Carefully consider your requirements and plan for the future, changing direction can be challenging to manage.

And be sure to pick the right solution, and partner, for the implementation.

Slide 14

I hope this talk has been helpful, thank you for your time.

<<Talk Ends>>