



Orchestrating GPU workloads with CloudStack

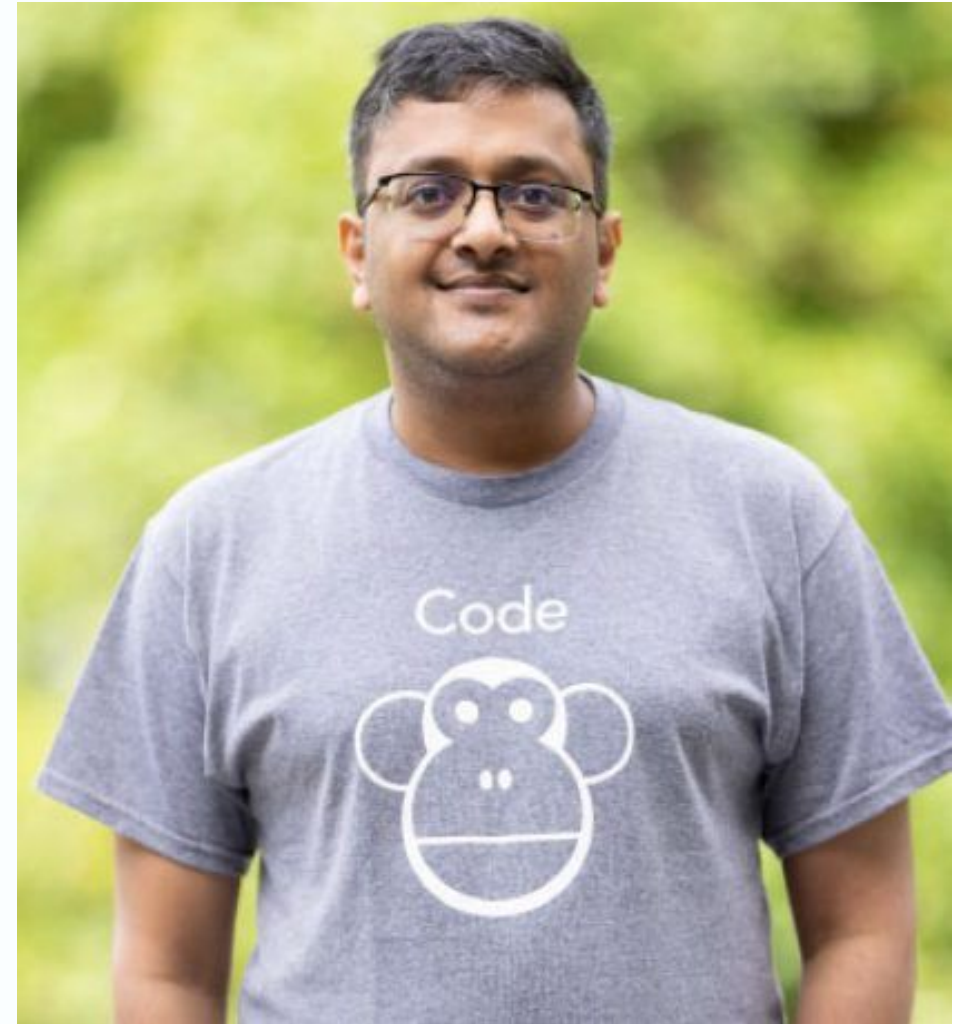
Vishesh Jindal





About me

- Contributing to ACS since 2023
- Committer @ Apache CloudStack since 2024
- PMC @ Apache CloudStack since last 2 weeks
- Software Engineer @ ShapeBlue
- Worked as a Backend & Infrastructure engineer in the past
- Experience managing and administering kubernetes, AWS, PostgreSQL in production
- Passionate about cloud infrastructure, security, and AI





What's in it for you?



Remember Last Year?



CCC24 Madrid

<https://www.youtube.com/watch?v=UyJTcy69ncg>



Who's behind this effort?



Thanks to Community Members





Let's see it in action



Select a GPU enabled Service Offering



3

Image

Type

Template

ISO

OS image that can be used to boot Instances.

Operating System

Ubuntu

CentOS

Other

User

Filter by

Search

Ubuntu 24.04

Featured

Public

HVM

Override root disk size

Total 1 items < 1 > 10 / page

4

Compute offering

Show GPU offerings

Search

Compute offering	CPU	Memory	GPU Cards
☐ Tiny Instance	1 CPU x 0.26 Ghz	256 MB	1 x Apache CloudStack Simulator Simulated Graphics Card VF (VF-P2)
☒ Medium [GeForce RTX 3050 Ti Mobile]	1 CPU x 1.00 Ghz	2048 MB	1 x NVIDIA Corporation GA107M [GeForce RTX 3050 Ti Mobile]

Override root disk offering

Total 2 items < 1 > 10 / page

5

Data disk

Search

Your Instance

KVM

OS type

Ubuntu 24.04 LTS

CPU

1 CPU x 1.00 GHz

Memory

2048 MB memory

GPU Cards

1 x NVIDIA Corporation GA107M [GeForce RTX 3050 Ti Mobile] (passthrough)

Disk size (in GB)

1 GB (Root)

Networks

test (Default)

Template

Ubuntu 24.04

Compute offering

Medium [GeForce RTX 3050 Ti Mobile]

Zone

DC



Instance's details page

 / Instances / QA-5cbf18ab-2338-4bc5-8555-3f53484ad8e3 

 Refresh





QA-5cbf18ab-2338-4bc5-8555-3f53484ad8e3



Simulator

Status

 Running

ID

 5cbf18ab-2338-4bc5-8555-3f53484ad8e3

OS type

 QA-5cbf18ab-2338-4bc5-8555-3f53484ad8e3

IP address

 10.1.1.104

CPU

 1 CPU x 1.00 GHz 

Memory

 1000 MB memory

GPU Cards

 1 x ACS Simulated Graphics Card Pro (sim-2q)

Network



Details

Metrics

Volumes

GPU Cards

NICs

Instance Snapshots

Backups

Schedules

Settings

Events

Comments

Name	Total
ACS Simulated Graphics Card Pro	1



Host's details page



Details

GPU

Events

Comments

Discover GPU devices

Add GPU Device

Summary

GPU Devices

Manage GPU Device

Unmanage GPU Device

Delete GPU Device

☐	Address	GPU Card	Profile	Managed state	State	Instance name	Actions
☐	00:01.0	Simulator Graphics Card Basic	passthrough	Managed	Allocated	i-4-5-VM	
☐	00:02.0	Simulator Graphics Card Basic	passthrough	Managed	Allocated	i-4-5-VM	
☐	00:03.0	Simulator Graphics Card Basic	passthrough	Managed	Allocated	i-2-7-VM	
☐	00:04.0	Simulator Graphics Card Basic	passthrough	Managed	Allocated	i-2-7-VM	
☐	00:05.0	Simulator Graphics Card Pro					
☐	0b1ecef4-94f6-49f4-9d77-fa405ffc5a4d	Simulator Graphics Card Pro	sim-2q	Managed	Free		
☐	15ca1a7e-d5af-42e0-a19e-5839dcf2f4ea	Simulator Graphics Card Pro	sim-2q	Managed	Free		
☐	be2a11f1-5501-4aea-9911-c35e9bbe833	Simulator Graphics Card Pro	sim-8q	Managed	Free		
☐	fd6162cd-84e6-4140-9a08-b10acdb3445c	Simulator Graphics Card Pro	sim-4q	Managed	Free		
☐	00:06.0	Simulator Graphics Card Pro					
☐	27cc4289-1803-4c10-beb7-ee96967112db	Simulator Graphics Card Pro	sim-4q	Managed	Free		
☐	4554ea70-22d0-4741-b973-54f589b68556	Simulator Graphics Card Pro	sim-2q	Managed	Free		
☐	a397d82b-7bcf-4d4e-9b2c-ceb7d6b2aba	Simulator Graphics Card Pro	sim-2q	Managed	Free		
☐	df4f2753-666a-475a-8f22-ce7e8fb70c96	Simulator Graphics Card Pro	sim-8q	Managed	Free		
☐	00:07.0	Simulator Graphics Card Pro					
☐	8ed940a3-e37f-4882-a670-	Simulator Graphics Card Pro	sim-8q	Managed	Free		

Configure Service Offerings

- Requires GPU Card & vGPU Profiles in CloudStack
- Created automatically when the `gpudiscovery.sh` script discovers GPUs on a host

Add Compute Offering ?

* Name ⓘ

Name

Description ⓘ

Description

Compute Offering type

Fixed Offering

Custom constrained

Custom unconstrained

* CPU cores ⓘ

the CPU number of the service...

* CPU (in MHz) ⓘ

For VMware and Xen based hy...

* Memory (in MB) ⓘ

the total memory of the servic...

Host tags ⓘ

the host tag for this service offering.

Network rate (Mb/s) ⓘ

data transfer rate in megabits per second allowed...

Offer HA ⓘ

☐

Dynamic scaling enabled ⓘ

☒

CPU cap ⓘ

☐

Volatile ⓘ

☐

Deployment planner ⓘ

GPU Card

Simulator Graphics Card Basic

GPU Profile

passthrough (8192 MB, 4096x2160)

GPU Count ⓘ

2

GPU Display ⓘ

☐

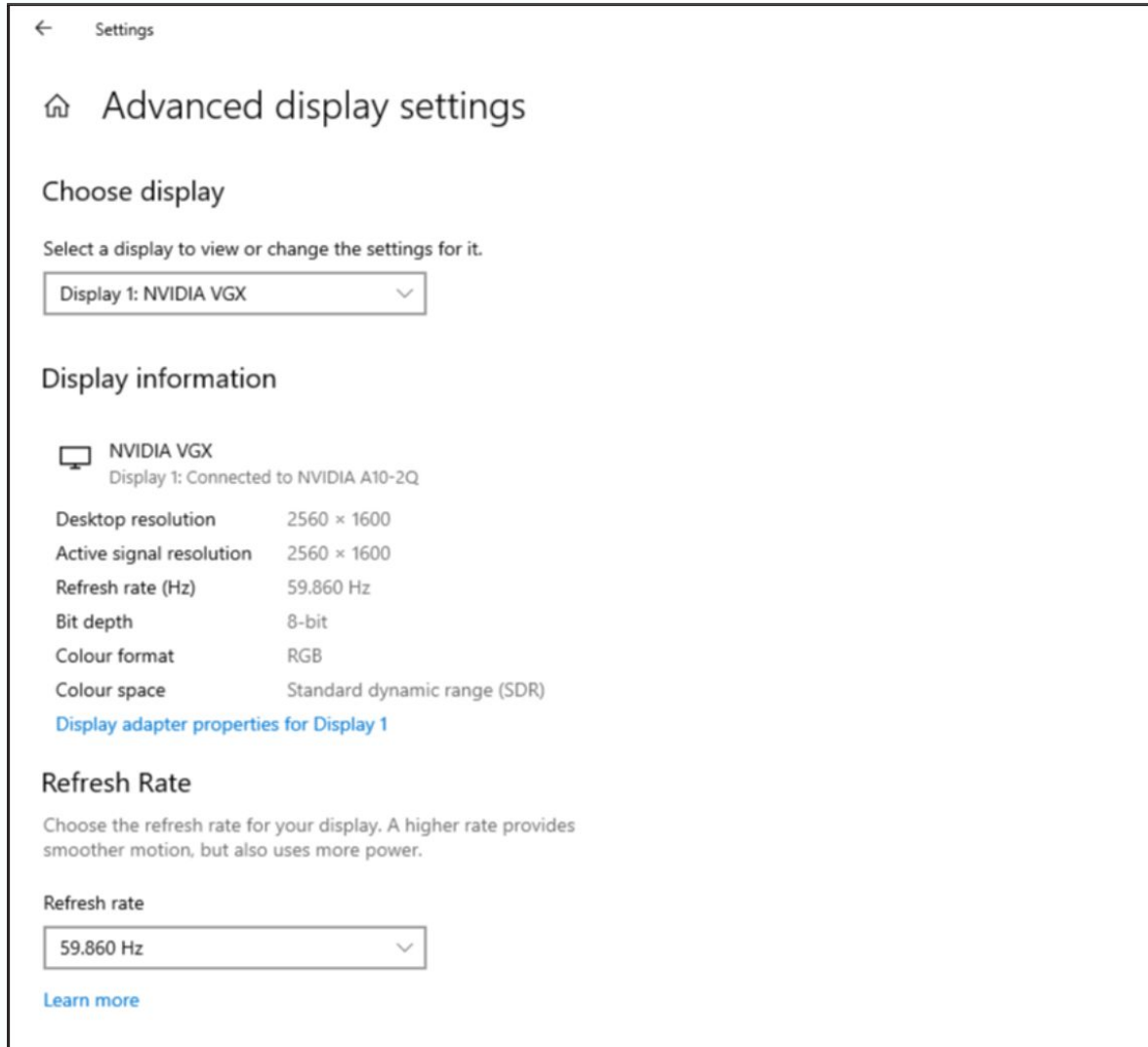
Public



vGPU for Display

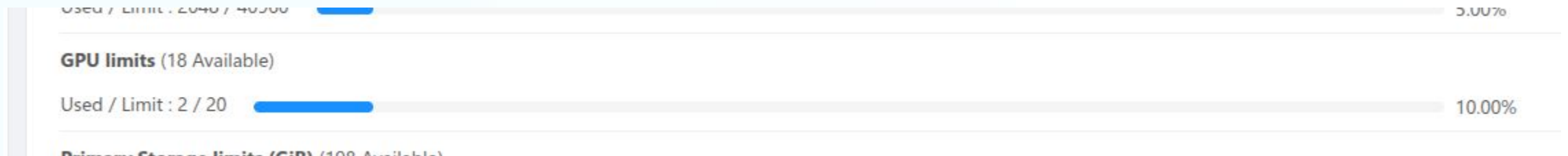


video.hardware - none



Configure Limits

- `max.account.gpus`
- `max.domain.gpus`
- `max.project.gpus`





Usage/Billing



Usage Type 1 - RUNNING_VM

Usage Type 2 - ALLOCATED_VM

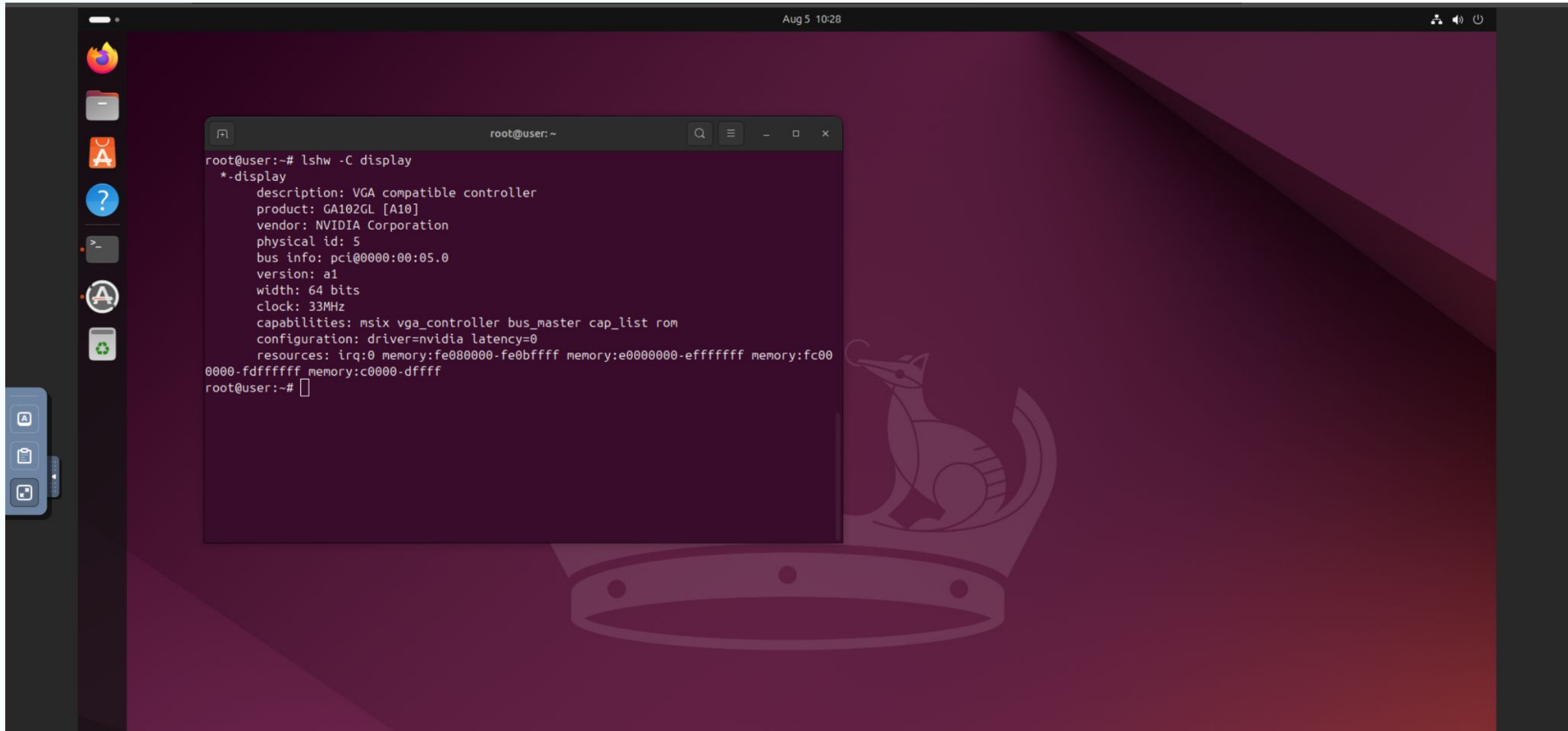
Detach GPU for stopped instances

`gpu.detach.on.stop`

Domain Account Allocated Vm Usage ▾ Resource ID 2025-11-12 → 2025-11-14 📅 Show usage records Download CSV Clear ☐ Fetch usage records for child domains										
Account	Domain	Usage type	Resource ID	Start date and time	End date and time	Raw usage (in hours)	Description	Offering ID		View
user	ROOT	Allocated Vm Usage	92769e2a-1314-4c4e-8f4d-b7d8dcfc6a81	12 Nov 2025 00:00:00	12 Nov 2025 23:59:59	13.37	Allocated VM usage for VM-92769e2a-1314-4c4e-8f4d-b7d8dcfc6a81 (i-4-5-VM) (92769e2a-1314-4c4e-8f4d-b7d8dcfc6a81) using service offering Large GPU Instance (de7a266d-2c6a-4c90-aeb2-66d2da0a774e) and template CentOS 5.6 (64-bit) no GUI (Simulator) (ebe45758-be56-11f0-8a4e-1e003400042e)	de7a266d-2c6a-4c90-aeb2-66d2da0a774e		🔍
user	ROOT	Allocated Vm Usage	92769e2a-1314-4c4e-8f4d-b7d8dcfc6a81	13 Nov 2025 00:00:00	13 Nov 2025 23:59:59	24	Allocated VM usage for VM-92769e2a-1314-4c4e-8f4d-b7d8dcfc6a81 (i-4-5-VM) (92769e2a-1314-4c4e-8f4d-b7d8dcfc6a81) using service offering Large GPU Instance (de7a266d-2c6a-4c90-aeb2-66d2da0a774e) and	de7a266d-2c6a-4c90-aeb2-66d2da0a774e		🔍



vGPU for Display





With Kubernetes

```
~ kubectl get no -o json | jq ".items[].metadata.labels" | grep nvidia
```

```
"nvidia.com/cuda.driver-version.full": "570.172.08",
"nvidia.com/cuda.driver-version.major": "570",
"nvidia.com/cuda.driver-version.minor": "172",
"nvidia.com/cuda.driver-version.revision": "08",
"nvidia.com/cuda.driver.major": "570",
"nvidia.com/cuda.driver.minor": "172",
"nvidia.com/cuda.driver.rev": "08",
"nvidia.com/cuda.runtime-version.full": "12.8",
"nvidia.com/cuda.runtime-version.major": "12",
"nvidia.com/cuda.runtime-version.minor": "8",
"nvidia.com/cuda.runtime.major": "12",
"nvidia.com/cuda.runtime.minor": "8",
"nvidia.com/gfd.timestamp": "1754387750",
"nvidia.com/gpu-driver-upgrade-state": "upgrade-done",
"nvidia.com/gpu.compute.major": "8",
"nvidia.com/gpu.compute.minor": "6",
"nvidia.com/gpu.count": "2",
"nvidia.com/gpu.deploy.container-toolkit": "true",
"nvidia.com/gpu.deploy.dcgmm": "true",
"nvidia.com/gpu.deploy.dcgmm-exporter": "true",
"nvidia.com/gpu.deploy.device-plugin": "true",
"nvidia.com/gpu.deploy.driver": "pre-installed",
"nvidia.com/gpu.deploy.gpu-feature-discovery": "true",
"nvidia.com/gpu.deploy.node-status-exporter": "true",
"nvidia.com/gpu.deploy.nvsm": "",
"nvidia.com/gpu.deploy.operator-validator": "true",
"nvidia.com/gpu.family": "ampere",
"nvidia.com/gpu.machine": "CloudStack-KVM-Hypervisor",
"nvidia.com/gpu.memory": "2048",
"nvidia.com/gpu.mode": "graphics",
"nvidia.com/gpu.present": "true",
"nvidia.com/gpu.product": "NVIDIA-A10-2Q",
"nvidia.com/gpu.replicas": "1",
"nvidia.com/gpu.sharing-strategy": "none",
"nvidia.com/mig.capable": "false",
"nvidia.com/mig.strategy": "single",
"nvidia.com/mps.capable": "false",
"nvidia.com/vgpu.host-driver-branch": "r573_48",
"nvidia.com/vgpu.host-driver-version": "570.172.07",
"nvidia.com/vgpu.present": "true"
```

Allocated resources:

(Total limits may be over 100 percent, i.e.,

Resource	Requests	Limits
-----	-----	-----
cpu	275m (6%)	50m (1%)
memory	252Mi (6%)	1656Mi (43%)
ephemeral-storage	0 (0%)	0 (0%)
hugepages-1Gi	0 (0%)	0 (0%)
hugepages-2Mi	0 (0%)	0 (0%)
nvidia.com/gpu	0	0



For Inferencing

```
Every 1.0s: nvidia-smi
VM-8a337393-192f-4e73-aca6-8a461dbda077: Fri Aug 8 06:50:02 2025
Fri Aug 8 06:50:02 2025
+-----+
| NVIDIA-SMI 570.172.08                Driver Version: 570.172.08      CUDA Version: 12.8   |
+-----+-----+-----+-----+-----+-----+
| GPU  Name      Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|                                           | MIG M.         |
+-----+-----+-----+-----+-----+-----+
|  0   NVIDIA A10-4Q          On   | 00000000:00:06.0 Off |           0         |
| N/A   N/A    P0              N/A /  N/A | 1602MiB /  4096MiB |      0%      Default |
|                                           | N/A              |
+-----+-----+-----+-----+-----+-----+
|  1   NVIDIA A10-4Q          On   | 00000000:00:07.0 Off |           0         |
| N/A   N/A    P0              N/A /  N/A |  1MiB /  4096MiB |      0%      Default |
|                                           | N/A              |
+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                         |
| GPU   GI    CI          PID    Type   Process name                      GPU Memory |
| ID    ID    ID                      | Usage      |
+-----+-----+-----+-----+-----+-----+
|  0   N/A  N/A             1181     C   /usr/local/bin/ollama              1601MiB |
+-----+-----+-----+-----+-----+-----+

root@VM-8a337393-192f-4e73-aca6-8a461dbda077:~# ollama run deepseek-r1:1.5b
>>> Hi. Tell me a joke.
Why don_t skeletons fight each other?
Because they don_t have the *guts*! __

>>> Send a message (/? for help)
```



How to enable GPU Support for my KVM Host?



Checklist for enabling GPU support

- Enable IOMMU
- Disable the nouveau kernel module
- Install pciutils & xlmstarlet for auto-discovery script
- Install GPU driver and create vGPUs (Optional)
- Discover GPU devices on the host



Enable IOMMU

Ubuntu

- Add `intel_iommu=on` for Intel and `amd_iommu=on` for AMD to GRUB_CMDLINE_LINUX_DEFAULT
- `sudo grub-mkconfig -o /boot/grub/grub.cfg`
- `sudo reboot`

Enterprise Linux

- `grubby --args="intel_iommu=on iommu=pt" --update-kernel DEFAULT`
- `sudo reboot`

Verify IOMMU is enabled

```
sudo dmesg | grep IOMMU
```

```
[    0.057175] DMAR: IOMMU enabled
```

```
...
```



Disable the nouveau kernel module

- Update modprobe configuration to disable nouveau

```
echo "blacklist nouveau" | sudo tee /etc/modprobe.d/disable-nouveau.conf
echo "options nouveau modeset=0" | sudo tee -a
/etc/modprobe.d/disable-nouveau.conf
```
- Regenerate the initial ramdisk
 - Ubuntu

```
update-initramfs -u
```
 - Enterprise Linux

```
dracut --force
```
- Reboot

```
sudo reboot
```
- Verify driver in use by the pci device

```
lspci -v | grep -A 10 " 3D "
```



Install dependencies for auto discovery script



Ubuntu

```
apt install pciutils xmlstarlet
```

Enterprise Linux

```
dnf install pciutils xmlstarlet
```




How do I enable vGPUs?

For nvidia GPUs only

- Install nvidia drivers from ui.licensing.nvidia.com
- Follow the documentation to create vGPUs on the host
- Execute `gpudiscovery.sh` script to confirm the vGPUs are getting discovered.

NVIDIA vGPU Type	VFIO Framework	Supported in CloudStack
Legacy: SR-IOV not supported	mdev	Yes
SR-IOV supported	mdev	Yes
SR-IOV supported	Vendor specific	Yes
Multi Instance GPU		No



Discover the GPU cards on the host

On Host

```
sudo systemctl restart cloudstack-agent
```

Or

Using cmk

```
cmk discover gpudevices id=<host id>
```



Checklist for enabling GPU support

- Enable IOMMU
- Disable the nouveau kernel module
- Install pciutils & xlmstarlet for auto-discovery script
- Install GPU driver and create vGPUs (Optional)
- Discover GPU devices on the host



Is that it?





Post-host setup configuration

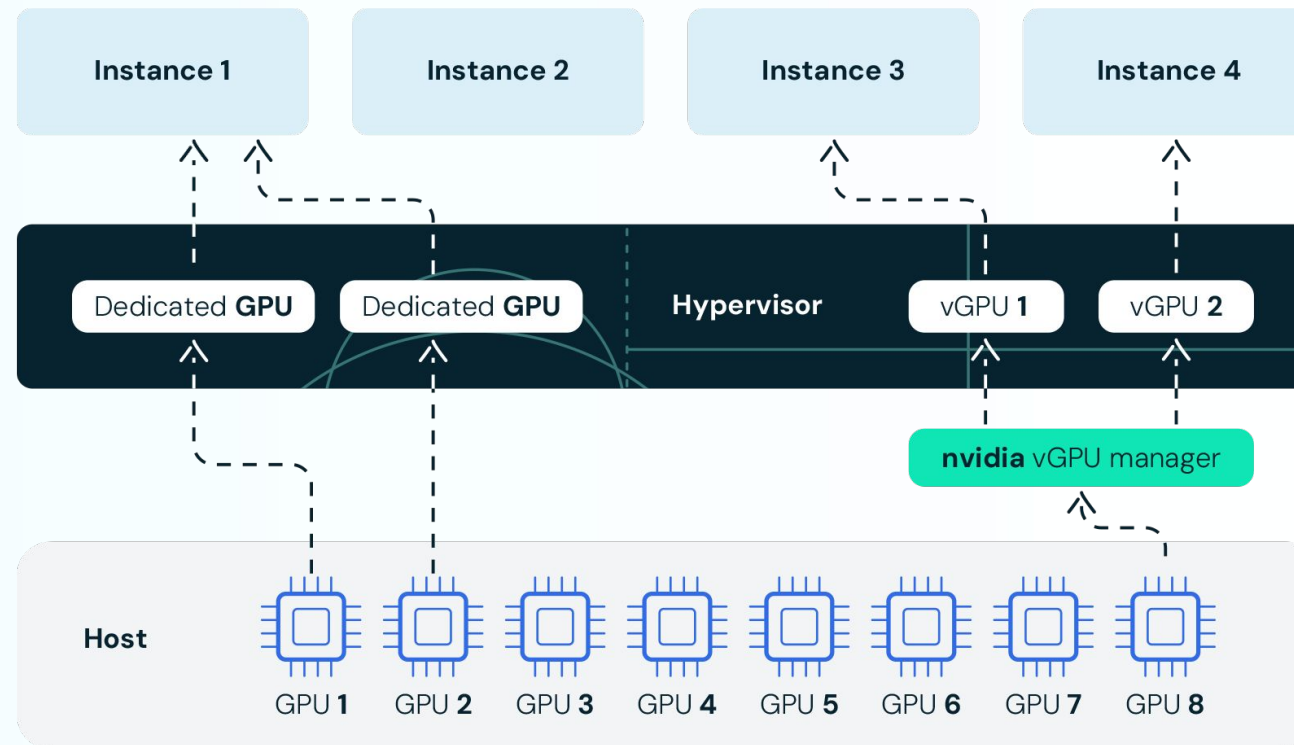
- Create GPU enabled Service Offerings
- Create Templates
 - ◆ Install GPU/vGPU drivers
 - ◆ Set `video.hardware` to `none` in Settings for VDI use cases
- Configure global settings and limits (per account, domain or project) as per requirements
 - ◆ `max.account.gpus` (Default - 20)
 - ◆ `max.domain.gpus` (Default - 20)
 - ◆ `max.project.gpus` (Default - 20)
 - ◆ `gpu.detach.on.stop` (Default - false)



How does it work with CloudStack?



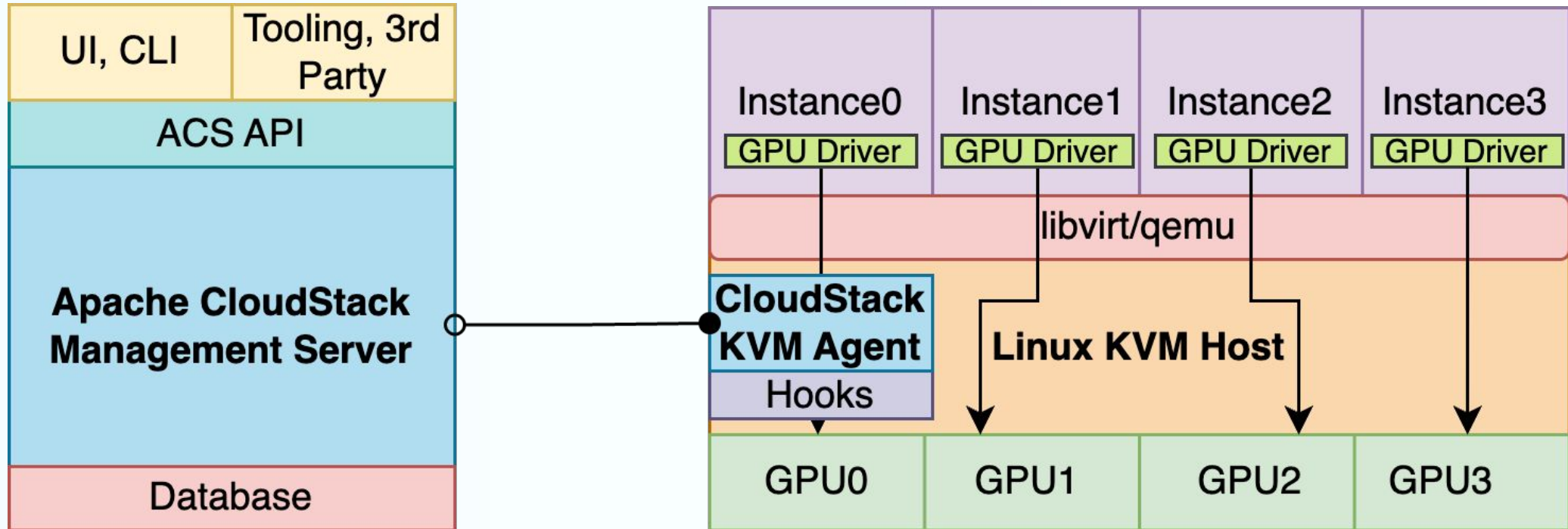
GPUs with Instances





Designing the Feature

How it works?



Finding GPUs Automatically

The Discovery Script

Operator customisable GPU auto-discovery script

[/usr/share/cloudstack-common/scripts/vm/hypervisor/kvm/gpudiscovery.sh](#)

```
{
  "gpus": [
    {
      "pci_address": "01:00.0",
      "vendor_id": "10de",
      "device_id": "25a0",
      "vendor": "NVIDIA Corporation",
      "device": "GA107M [GeForce RTX 3050 Ti Mobile]",
      "driver": "nvidia",
      "pci_class": "3D controller [0302]",
      "iommu_group": "16",
      "pci_root": "0000:01:00.0",
      "numa_node": -1,
      "sriov_totalvfs": 0,
      "sriov_numvfs": 0,
      "full_passthrough": {
        "enabled": 1,
        "libvirt_address": {
          "domain": "0x0000",
          "bus": "0x01",
          "slot": "0x00",
          "function": "0x0"
        }
      },
      "used_by_vm": null
    },
    {
      "vgpu_instances": [],
      "vf_instances": []
    }
  ]
}
```

Customisable Agent Hooks

Domain XML Transformer & States

In addition to the Groovy-based agent hooks, following are supported:

- Libvirt Domain XML Transformer **libvirt-vm-xml-transformer.sh**
- VM Start hook **libvirt-vm-state-change.sh**
- VM Stop hook **libvirt-vm-state-change.sh**

Location: **/etc/cloudstack/agent/hooks**



What's next?



Challenges & Future Work

- Accessibility of GPU-based hosts for feature development and QA
- Validate and improve built-in GPU auto-discovery and domain transformer scripts against different GPU cards/models
- GPU Metrics & Expand support for other hypervisors.



Questions?





Thank you!
#CSCollab25
@CloudStack



Fill in the CloudStack User Survey
Help us understand the CloudStack Ecosystem