

**OpenRadioss™** +  **ParaView**

**Fast-track to Exploring OpenRadioss with  
Visualization Using ParaView on Oracle Cloud**



[ciq.com](http://ciq.com)

# Fast-track to Exploring OpenRadioss with Visualization Using ParaView on Oracle Cloud

This guide provides you with a fast-track path to run OpenRadioss simulations and use ParaView to visualize the results. CIQ created builds of OpenRadioss and ParaView and packaged them into a single Rocky Linux 9 image [published in the Oracle Cloud marketplace](#). In this guide, we'll provide some of the details of the effort and, most importantly, the instructions to try out OpenRadioss simulations quickly. By following the instructions in this guide, you can try a sample workload or give your own workload a go!

**OpenRadioss is an industry-proven open source Finite Element Analysis solver recently provided by Altair. More information, community support of the code, and additional workloads may be found on the [OpenRadioss community website](#).**

**[ParaView](#) is an award-winning open source visualization application.**

**[Rocky Linux](#) is a popular Linux distribution across all domains, including scientific applications like OpenRadioss. CIQ is a founding partner in Rocky Linux, the Rocky Linux Enterprise Software Foundation, and also provides professional support for Rocky Linux.**

As a prerequisite to executing this workflow, you will need to have an account on Oracle Cloud. This guide does not cover the instructions for setting up an account.

## Minimizing the cost of running in the cloud

The instructions for executing OpenRadioss and ParaView in Oracle Cloud take into account that running in the cloud is not free. With that in mind, this section provides some background on minimizing the cost of running in the cloud.

The primary cost-efficient consideration comes down to selecting the right compute instance shape for each part of the workflow. OpenRadioss executes its solver on CPUs, while ParaView needs GPU power for the visualization. The cost of a GPU instance shape is much higher, typically, than a shape with just CPUs. If we choose one instance shape that satisfies both requirements, then the overall cost of execution will be greater. Instead, this guide uses a two-step workflow. The first uses a CPU only shape to execute the OpenRadioss simulation. The second uses a shape with an NVIDIA GPU to visualize the results. While there are additional steps to perform, this is overall a more cost effective path. Otherwise, while OpenRadioss is solving your workload, you're paying for a GPU that is sitting idle.

## Workflow step 1: OpenRadioss simulation

In this example, we will run an OpenRadioss crash simulation using a 2019 generic mid-size ADS vehicle model from [CCSA](#). For the first step in the workflow, we will use an AMD CPU shape with 64 CPU cores to run the OpenRadioss workload.

George Mason University | College of Science

Center for Collision Safety and Analysis **CCSA**

About Research Focus Academics People Research Capability News

Center for Collision Safety and Analysis / Finite Element Models / 2019 Generic ADS Vehicles

Small-size model Mid-size model Large-size model Tractor model

### 2019 Generic Mid-size ADS Vehicle Model

Version 1, released November 2019 [Download Mid-size ADS zip](#) 10.7 MB

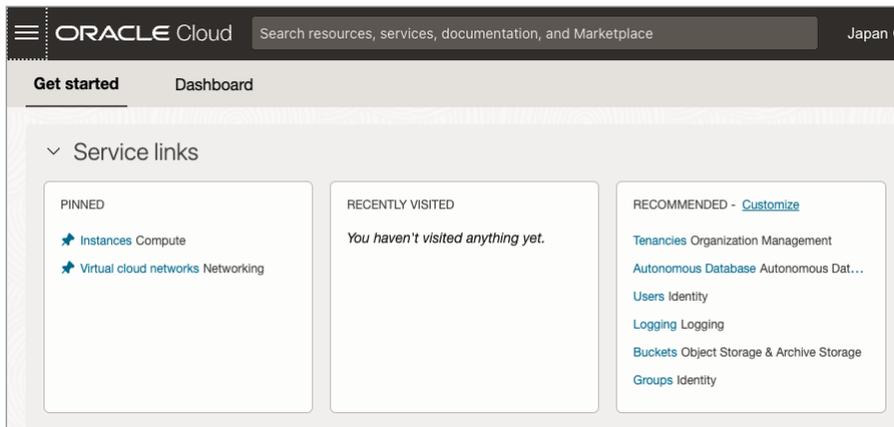
**Version 1 Model Details**

Elements	439,303
Nodes	456,453
Parts	371
Features	Structural components details, Interior component details, Suspension system details, Uniform mesh throughout (multi mode impacts)

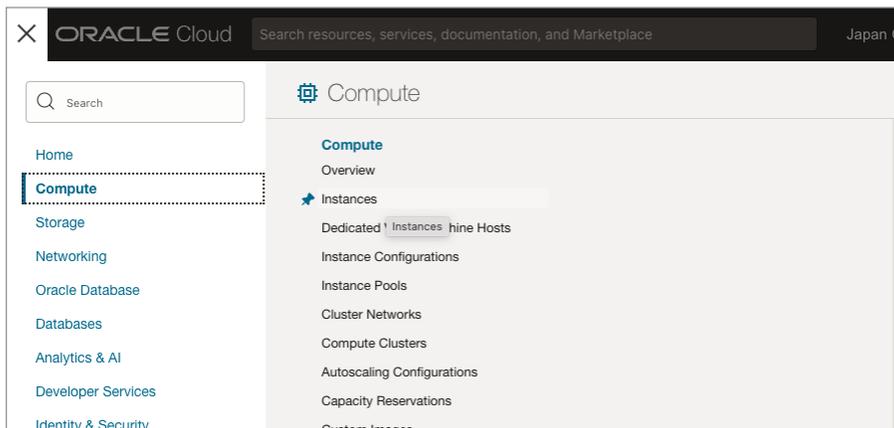
## Create the instance on Oracle Cloud

Create an instance that has 64 cores and 64 GB of memory for the OpenRadioss workload.

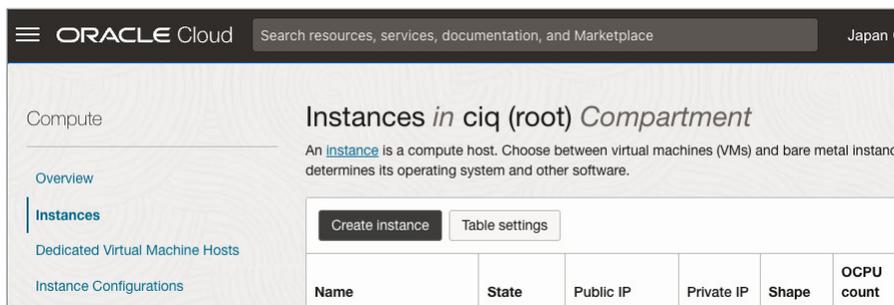
This screen appears right after logging in to Oracle Cloud. Click the top left corner to open the menu.



Click Compute > Instances



Click "Create Instance"



It shows many options, but first we are going to choose the Rocky Linux image and then the shape of the instance.

ORACLE Cloud Search resources, services, documentation, and Marketplace Japan Central (Osaka) ▾

## Create compute instance

Shielded instance: Disabled

### Image and shape [Collapse](#)

A [shape](#) is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The image is the operating system that runs on top of the shape.

Image


**Oracle Linux 8**  
 Image build: 2023.04.25-0 [Change image](#)

Shape


**VM.Standard.E4.Flex**  
 Virtual machine, 1 core OCPU, 16 GB memory, 1 Gbps network bandwidth [Change shape](#)

Click “Change image,” select “Rocky Linux,” and then select the OpenRadioss image from the list.

mentation, and Marketplace Japan Central (Osaka) ⌵ ⌵ 🔔 ? 🌐 👤

## Select an image

Oracle Linux

Ubuntu

Red Hat

CentOS

Windows

SUSE

AlmaLinux

Rocky Linux ✓

Marketplace

My images  
Custom images & boot volumes

App name	Publisher	Price
<input checked="" type="checkbox"/> OpenRadioss and ParaView on Rocky Linux 9.1	Ctrl IQ, Inc.	Free

CIQ brings together everything you need for simulating and visualizing complex, non-linear problems. It comes bundled with Rocky Linux 9, OpenRadioss, and ParaView, creating a comprehensive software stack for high-performance computation of shapes. This all-encompassing solution is readily available on the Oracle Cloud.

Rocky Linux, which is supported by CIQ and recognized as the successor to CentOS, stands at the forefront of operating systems for simulation tasks. CIQ instills resilience and security into the supported distribution, assuring smooth operation on Oracle's computing shapes.

In 2022, Altair made the significant move of open-sourcing the Radioss source code, leading to the creation of OpenRadioss. For over three decades, Radioss has been a game-changer in digital design processes and product development. Now, OpenRadioss offers a community-supported codebase, providing a direct route to professionally backed code from Altair itself.

ParaView is a widely-used visualization tool for analyzing simulation data. It boasts the ability to produce both static images and dynamic animations from simulations, enabling designers to visually interpret the simulation outcomes with essential result indicators.

Agreement for partner image **OpenRadioss and ParaView on Rocky Linux 9.1**

I have reviewed and accept the [Oracle Terms of Use](#), [Partner terms and conditions](#), and the [Oracle General Privacy Policy](#).

Select image
Cancel

Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

Click “Change shape, maximize “Number of OCPUs” to 64 cores, and specify “Amount of memory (GB)” to 64 GB.

## Browse all shapes

A **shape** is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance.

### Instance type

#### Virtual machine

A virtual machine is an independent computing environment that runs on top of physical bare metal hardware. ✓

#### Bare metal machine

A bare metal compute instance gives you dedicated physical server access for highest performance and strong isolation.

### Shape series

#### AMD



Flexible OCPU count. Current generation AMD processors. ✓

#### Intel



Flexible OCPU count. Current generation Intel processors.

#### Ampere



Arm-based processor.

#### Specialty and previous generation

Always Free, Dense I/O, GPU, HPC, Generic, and earlier generation AMD and Intel standard shapes.

Image: Custom Custom

Shape name	OCPU <sup>i</sup>	Memory (GB)	Security
<input checked="" type="checkbox"/> VM.Standard.E4.Flex	64	64	

**Network bandwidth (Gbps):** 40

**Maximum VNICs:** 24 <sup>i</sup>

You can customize the number of OCPUs and the amount of memory allocated to a flexible shape. The other resources scale proportionately. [Learn more about flexible shapes.](#)

Number of OCPUs

1 22 43 64

Amount of memory (GB) <sup>i</sup>

1 64 342 683 1024

**Burstable**

[Burstable instances](#) are virtual machine (VM) instances that provide a baseline level of CPU performance with the ability to burst to a higher level to support occasional increases in usage.

1 selected Showing 1 item

Don't see the shape you want? [View your service limits and request an increase.](#) If you're looking for an older shape, check the **Specialty and previous generation** section.



This is after you choose the image and shape.

ORACLE Cloud Search resources, services, documentation, and Marketplace Japan

## Create compute instance

### Image and shape [Collapse](#)

A [shape](#) is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The image is the operating system that runs on top of the shape.

Image



**OpenRadioss and ParaView on Rocky Linux 9.1**  
OpenRadioss and ParaView on Rocky Linux 9.1

[Change image](#)

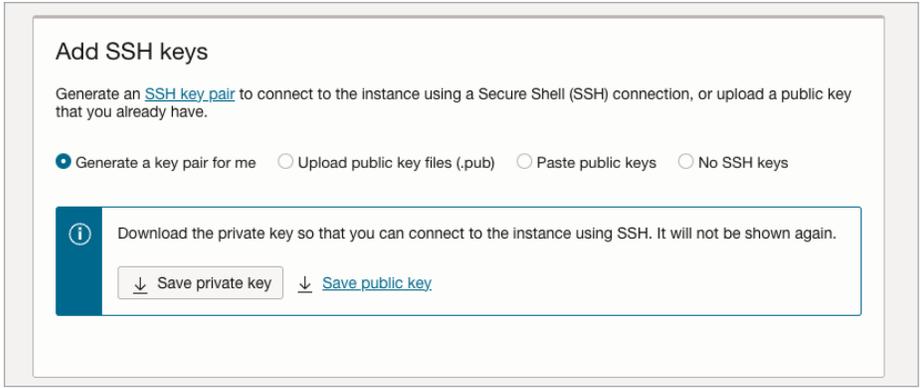
Shape



**VM.Standard.E4.Flex**  
Virtual machine, 64 core OCPU, 64 GB memory, 40 Gbps network bandwidth

[Change shape](#)

Click “Save private key” and “Save public key” to download keys for SSH. We use the private key to SSH log-in to the instance we created here, and we use the public key when we create other GPU-equipped instances for visualization.



Now you have two keys under the Downloads directory (depending on your environment and how you saved those keys); one is like “ssh-key-yyyy-mm-dd.key”. This is the private key that we use for SSH log-in, and the other is like “ssh-key-yyyy-mm-dd.key.pub”. This is the public key that we use when we create other instances.

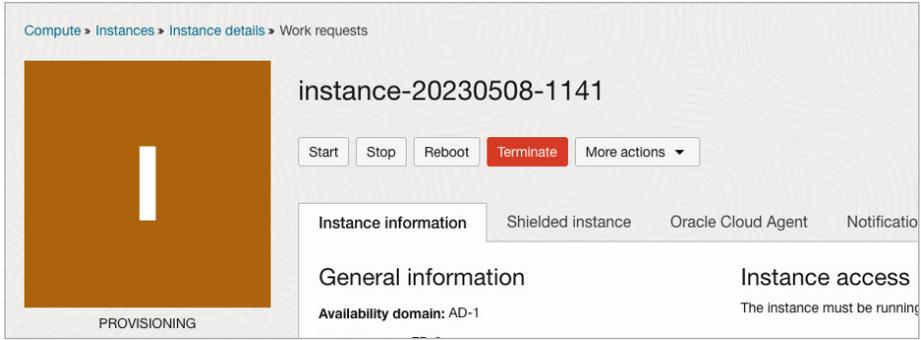
Open your terminal and then type the following to attach read write permission only to the current user; otherwise, SSH log-in fails. This procedure is only required for MacOS and Linux users.

```
$ chmod 600 ~/Downloads/ssh-key-2023-05-08.key
```

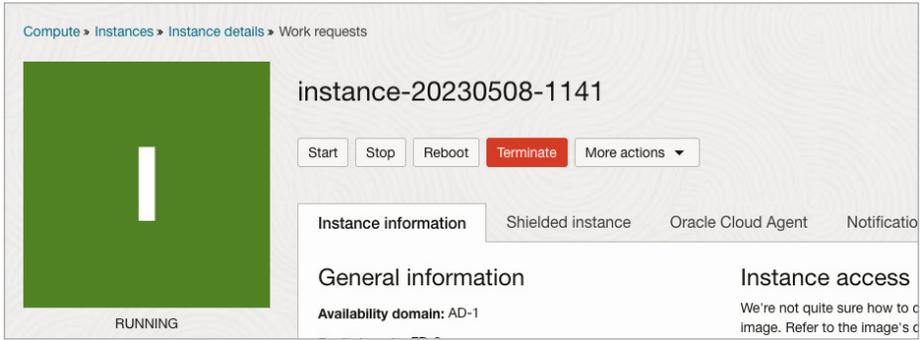
Click “Create”



The instance is in a “provisioning” state for usually less than a minute.

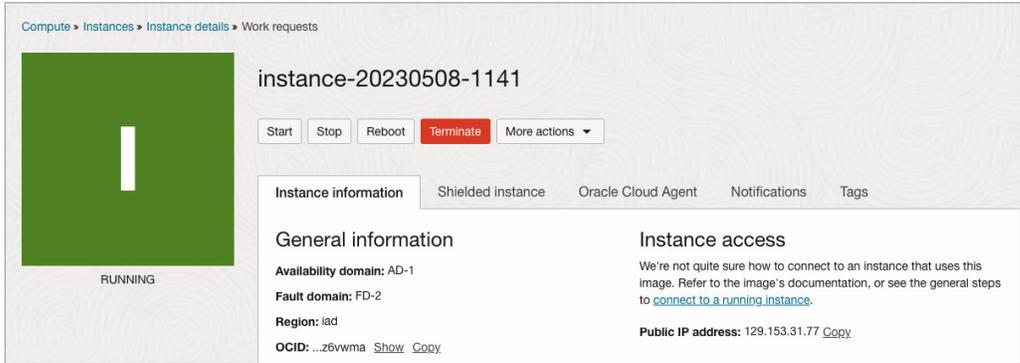


When your instance becomes “running” state, you can SSH log-in to this instance. Sometimes you need to wait a bit after an instance becomes a “running” state, due to the cloud-init script still running, etc.



## SSH

Let’s SSH log-in to the instance just created. For SSH log-in, we need the IP address of the instance, username, and SSH key. The IP address of the instance is shown on the “Instance access” section of the Instance details page. The username is “rocky” and the SSH key is the private key that we downloaded when we created the instance.



Open your terminal (when you use Windows, it's "Windows Terminal"; when you use MacOS, it's "Terminal"), and then type the following to SSH log-in to the instance.w

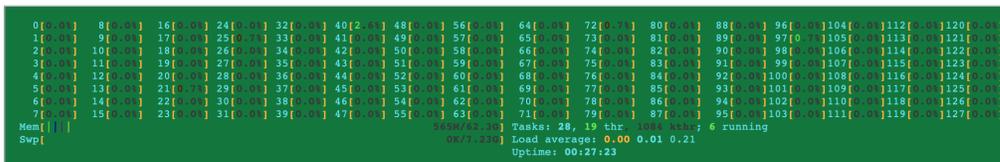
```
$ ssh -i ~/Downloads/ssh-key-2023-05-08.key rocky@129.153.31.77
```



This is what you see with a successful SSH log-in to the instance.it's "Terminal"), and then type the following to SSH log-in to the instance.w



Open your terminal (when you use Windows, it's "Windows Terminal"; when you use MacOS, it's "Terminal"), and then type the following to SSH log-in to the instance.w



This shows the instance we just created has 128 cores. This is because AMD Simultaneous Multithreading (SMT) is enabled. We use 64 cores for our simulation, since AMD SMT and Intel HT don't contribute to the performance of OpenRadioss and HPC applications in general.

type “q” to quit “htop”

## Run the simulation

### Preparing Input Files

- Move to “Simulation” directory
- Unarchive “2019-generic-mid-size-ads-vehicle-v1.zip”
- Copy “ADSOpenRadioss.key”
- Move to “2019-generic-mid-size-ads-vehicle-v1” directory

```
$ cd Simulation
$ unzip 2019-generic-mid-size-ads-vehicle-v1.zip
Archive:  2019-generic-mid-size-ads-vehicle-v1.zip
2019 Generic Mid-size ADS Vehicle Model
Center for Collision Safety and Analysis
George Mason University
  inflating: 2019-generic-mid-size-ads-vehicle-v1/README.txt
  inflating: 2019-generic-mid-size-ads-vehicle-v1/combine.key
  inflating: 2019-generic-mid-size-ads-vehicle-v1/gmu-ccsa-generic-mid-size-ads-vehicle-v1.key
  inflating: 2019-generic-mid-size-ads-vehicle-v1/wall.key
$ cp ADSOpenRadioss.key 2019-generic-mid-size-ads-vehicle-v1
$ cd 2019-generic-mid-size-ads-vehicle-v1
```

### Decompose Simulation Input Data

Use “tmux”, terminal multiplexer, here. This is convenient when you want to exit your terminal while the simulation is still running or when you lose connection while the simulation is still running. “tmux” creates a session, and you can detach from that session without stopping your simulation.

```
$ tmux
```

Decompose simulation input data for 64 processes MPI parallel simulation.

```
$ openradioss.sif
Apptainer> starter_linux64_gf -i ADSOpenRadioss.key -np 64
```

### Run Simulation

Run the crash simulation. It takes about 40 minutes.

```
Apptainer> mpirun -np 64 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad
```

This is optional, but let's check out CPU usage with “htop” for fun.

Press keys “Ctrl + b” and then press key “d”. This makes you detach from the current session, such as the OpenRadioss simulation terminal session.

Then, type the following and press Enter.

```
$ htop
```

```

0[0]  ] 8[0]  ] 16[2] ] 24[0] ] 32[100] 40[100] 48[0]  ] 56[100] 64[100] 72[0]  ] 80[0]  ] 88[0]  ] 96[0]  ] 104[0]  ] 112[3.0] 120[0]  ]
1[100] 9[100] 17[71.] 25[100] 33[0]  ] 41[0.0] 49[100] 57[0]  ] 65[0]  ] 73[100] 81[100] 89[100] 97[100] 105[100] 113[100] 121[100]
2[100] 10[0]  ] 18[100] 26[100] 34[0]  ] 42[100] 50[100] 58[100] 66[100] 74[100] 82[100] 90[100] 98[0]  ] 106[100] 114[100] 122[100]
3[0]  ] 11[100] 19[0]  ] 27[0]  ] 35[100] 43[0]  ] 51[0]  ] 59[0]  ] 67[0]  ] 75[0]  ] 83[0]  ] 91[0]  ] 99[100] 107[0]  ] 115[0]  ] 123[0]  ]
4[100] 12[0]  ] 20[100] 28[0]  ] 36[100] 44[100] 52[0]  ] 60[100] 68[0]  ] 76[100] 84[100] 92[0]  ] 100[100] 108[100] 116[100] 124[0]  ]
5[0]  ] 13[100] 21[0]  ] 29[100] 37[0]  ] 45[0]  ] 53[100] 61[0]  ] 69[100] 77[0]  ] 85[0]  ] 93[100] 101[0]  ] 109[0]  ] 117[0]  ] 125[100]
6[100] 14[100] 22[100] 30[100] 38[0]  ] 46[100] 54[100] 62[0]  ] 70[0]  ] 78[100] 86[100] 94[100] 102[0]  ] 110[0]  ] 118[100] 126[100]
7[0]  ] 15[0]  ] 23[0]  ] 31[0]  ] 39[100] 47[0.0] 55[0]  ] 63[100] 71[100] 79[0]  ] 87[0]  ] 95[0]  ] 103[100] 111[100] 119[0.0] 127[0]  ]
Mem[|||||]
Swp[
          9.750/82.30 Tasks: 97, 169 thr. 1000 sdb; 65 running
          98/7.23% Load average: 34.06 9.12 3.16
          Dptime: 00:41:45

Main 170
PID USER  PRI  NI  VIRT  RES  SHR  S  CPU%MEM%  TIME+  Command
4723 rocky  20   0  663M  220M 126M R 100.8  0.3  0:40.70 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad
4726 rocky  20   0  649M  201M 121M R 100.8  0.3  0:40.73 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad
4728 rocky  20   0  651M  203M 121M R 100.8  0.3  0:40.74 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad
4729 rocky  20   0  650M  199M 119M R 100.8  0.3  0:40.73 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad
4731 rocky  20   0  649M  195M 116M R 100.8  0.3  0:40.72 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad
4733 rocky  20   0  649M  202M 123M R 100.8  0.3  0:40.73 engine_linux64_gf_ompi -i ADSOpenRadioss_0001.rad

```

This shows the simulation utilizes 64 cores on your instance.

Let's get back to the OpenRadioss simulation session. Type "q" to quit "htop" and then type the following to attach to the session:

```
$ tmux attach -t 0
```

When the simulation is finished, it will show you something like this:

```

** MEMORY USAGE STATISTICS **

TOTAL MEMORY USED .....: 14671 MB
MAXIMUM MEMORY PER PROCESSOR.....: 259 MB
MINIMUM MEMORY PER PROCESSOR.....: 225 MB
AVERAGE MEMORY PER PROCESSOR.....: 229 MB

** DISK USAGE STATISTICS **

TOTAL DISK SPACE USED .....: 5386 MB
ANIMATION/H3D/TH/OUTP SIZE .....: 4272 MB
RESTART FILE SIZE .....: 1113 MB

ELAPSED TIME = 2341.64 s
              0:39:01
ESTIMATED SPEEDUP= 63.62

NORMAL TERMINATION
TOTAL NUMBER OF CYCLES : 152280
Aptainer>
[0] 0:bash*

```

64 cores with "ESTIMATED SPEEDUP" of 63.62 looks like pretty good performance. If you wish to accelerate more, you will need multiple nodes set up, and an HPC cluster. That is indeed an interesting topic, but it is not covered in this article.

## Prepare Output Data for ParaView Visualization

OpenRadioss produces visualization data in ANIM format, but ParaView does not have a plugin for reading ANIM format. That means we need to convert ANIM format data to VTK format, which is the primary supported data format by ParaView.

Let's convert the ANIM format data to VTK format data. It takes about 5 minutes to finish.

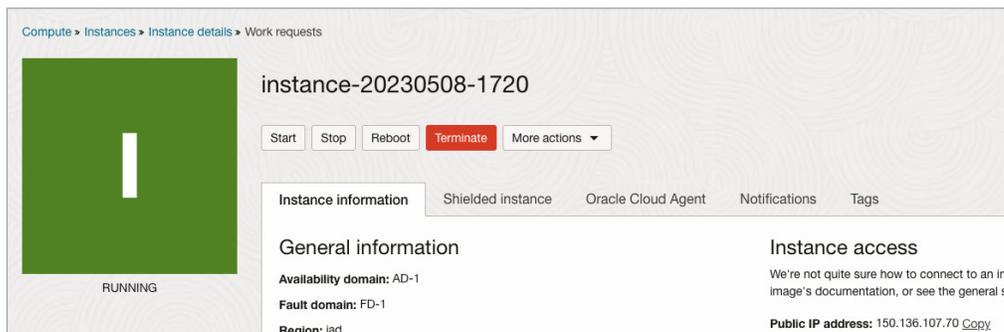
```
$ seq -f ADSOpenRadiossA%03g 076 | xargs -I{} sh -c 'openradioss.sif anim_to_vtk_linux64_gf "$1" > "$1.vtk"' -- {}
```

If you didn't finish your simulation during the previous step, don't worry. The custom image we use here has sample VTK files. Please use VTK files under `~/visualization` if you don't have your own data.

## Terminate the CPU Instance

We have done CPU-intensive workloads such as the OpenRadioss simulation, so let's terminate the instance. Please keep in mind that we will reuse the disk (boot volume) that we just used for the OpenRadioss simulation.

Click "Terminate"



Please leave the following checkbox unchecked; we would like to use this boot volume that contains data for visualization. We can create a GPU-equipped instance from this boot volume in later steps.

Click "Terminate instance"

## Terminate instance [Help](#)

Do you want to permanently delete instance **instance-20230508-1141**?

Permanently delete the attached boot volume

Terminate instance
Cancel

It takes less than a few minutes to be terminated.

Compute > Instances > Instance details

TERMINATING

### instance-20230508-1141

Start
Stop
Reboot
Terminate
More actions ▾

**Instance information** Shielded instance Oracle Cloud

**General information**

**Availability domain:** AD-1

**Fault domain:** FD-2

**Region:** IAD

After the instance has terminated, please move on to the next step.

## Workflow step 2: visualizing with ParaView

Create an instance for the visualization workload from the same boot drive that is used for the OpenRadioss simulations.

Click “Create instance”

ORACLE Cloud

Japan C

Compute

Overview

Instances

Dedicated Virtual Machine Hosts

Instance Configurations

### Instances *in ciq (root) Compartment*

An [instance](#) is a compute host. Choose between virtual machines (VMs) and bare metal instance determines its operating system and other software.

Create instance
Table settings

Name	State	Public IP	Private IP	Shape	OCPU count

Click “Change image”

**ORACLE Cloud** Search resources, services, documentation, and Marketplace Japan Central (Osaka) ▾

### Create compute instance

Shielded instance: Disabled

**Image and shape** [Collapse](#)

A [shape](#) is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The image is the operating system that runs on top of the shape.

**Image**

Oracle Linux 8  
Image build: 2023.04.25-0 [Change image](#)

**Shape**

VM.Standard.E4.Flex  
Virtual machine, 1 core OCPU, 16 GB memory, 1 Gbps network bandwidth [Change shape](#)

Click “My images”

### Select an image

Oracle Linux Ubuntu Red Hat CentOS

Windows SUSE AlmaLinux Rocky Linux

Marketplace **My images**  
Custom images & boot volumes ✓

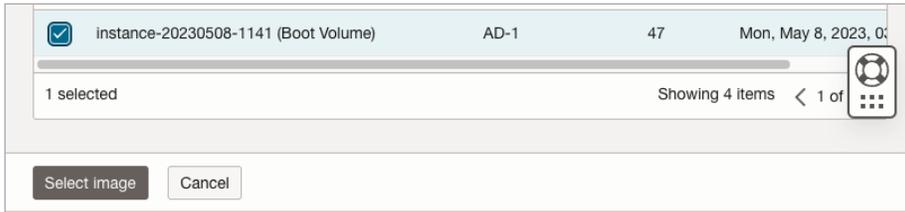
Custom images  Boot volumes  Image OCID

[Boot volumes](#) contain the image used to create a new instance.

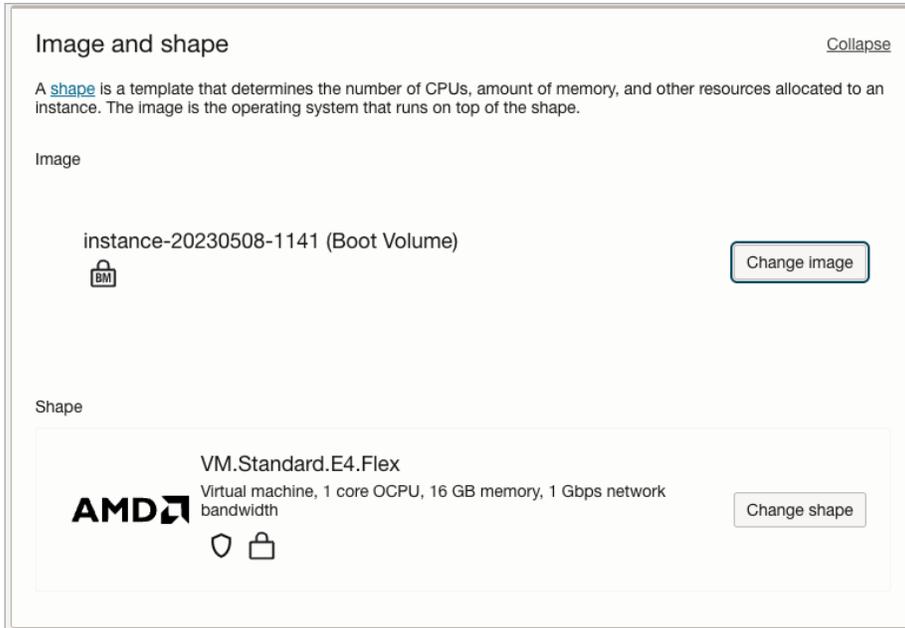
Compartment

and then click “Boot volumes”

Choose Boot volume and click “Select image”



Next, click “Change shape”



This time, we use the NVIDIA GPU equipped shape. Click “Specialty and previous generation”.

## Browse all shapes

A **shape** is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance.

Instance type

**Virtual machine**

A virtual machine is an independent computing environment that runs on top of physical bare metal hardware. ✓

**Bare metal machine**

A bare metal compute instance gives you dedicated physical server access for highest performance and strong isolation.

Shape series

**AMD**

Flexible OCPU count. Current generation AMD processors.

**Intel**

Flexible OCPU count. Current generation Intel processors.

**Ampere**

Arm-based processor.

**Specialty and previous generation**

Always Free, Dense I/O, GPU, HPC, Generic, and earlier generation AMD and Intel standard shapes. ✓

Click “VM.GPU.A10.1” shape that has a NVIDIA A10 GPU and 15 cores Intel Xeon processor, and then click “Select shape”

VM.GPU.A10.1 15    240    ^

---

**Network bandwidth (Gbps):** 24

**Maximum VNICs:** 15 ⓘ

**Local disk:** Block storage only

**Processor:** 2.6 GHz Intel® Xeon® Platinum 8358 (Ice Lake)

I have reviewed and accept the following documents: [Oracle and Nvidia Terms of Use](#)

Now it looks like this:

### Image and shape Collapse

A **shape** is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The image is the operating system that runs on top of the shape.

Image

instance-20230508-1141 (Boot Volume)

Shape

**VM.GPU.A10.1**

Virtual machine, 15 core OCPU, 240 GB memory, 24 Gbps network bandwidth

Now, let's look at SSH key settings. This time we upload the public key that we downloaded when we created an instance for the OpenRadioss simulation.

Click "Upload public key files (.pub), and then click "Browse"

### Add SSH keys

Generate an [SSH key pair](#) to connect to the instance using a Secure Shell (SSH) connection, or upload a public key that you already have.

Generate a key pair for me
  Upload public key files (.pub)
  Paste public keys
  No SSH keys

SSH public keys

Drop .pub files here. [Browse](#)

Select your public key and then click "open". Now it looks like this:

### Add SSH keys

Generate an [SSH key pair](#) to connect to the instance using a Secure Shell (SSH) connection, or upload a public key that you already have.

Generate a key pair for me
  Upload public key files (.pub)
  Paste public keys
  No SSH keys

SSH public keys

Drop .pub files here. [Browse](#)

ssh-key-2023-05-08.key.pub x

We are ready to go! Click "Create"

[Show advanced options](#)

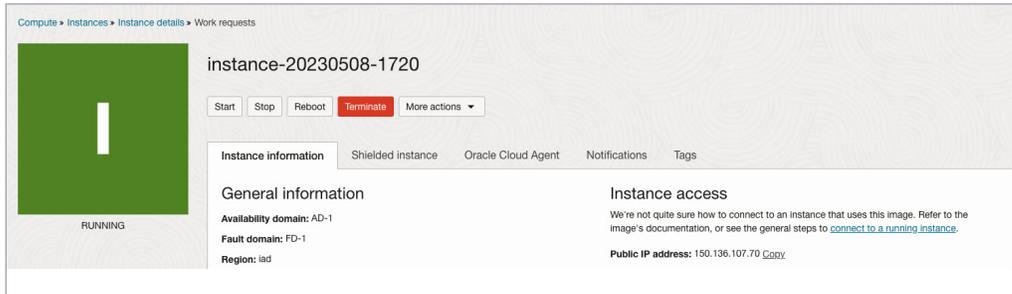
Create
Save as stack
Cancel

[Terms of Use and Privacy](#)
[Cookie Preferences](#)

## SSH

After the newly created instance becomes the “Running” state, SSH log-in to the instance.

Username is the same as “rocky”, and the ssh key is also the same “ssh-key-yyyy-mm-dd.key”; only the IP address is changed here. Please check the new IP address that is assigned to the instance.



## Optional: Verify the GPU driver is loaded

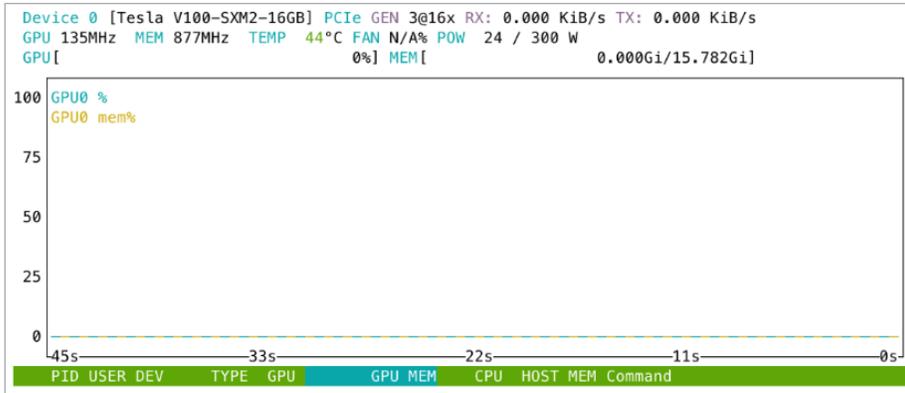
If you follow these instructions verbatim, you'll be following a path that has been verified, meaning everything should be working. It's never a bad idea, though, to do a quick verification that the GPU is up and running. You can use the following command to verify that the NVIDIA driver is running.

```
$ nvidia-smi
```

```
Mon May 8 08:52:39 2023
+-----+
| NVIDIA-SMI 530.30.02                  Driver Version: 530.30.02   CUDA Version: 12.1   |
+-----+-----+-----+-----+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A   | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap|      /          |      GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+
|  0  NVIDIA A10      Off          | 00000000:00:04.0 Off  |      18%      Default  |
|    0%    35C    P0              52W / 150W| 0MiB / 23028MiB |      N/A      N/A    |
+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                              |
| GPU  GI    CI           PID  Type   Process name          | GPU Memory |
| ID   ID   ID             |              |           | Usage         |
+-----+-----+-----+-----+-----+
| No running processes found              |
+-----+
```

You can also monitor GPU activity using “nvidia-smi”:

```
$ nvidia-smi
```



Type “q” to quit “nvidia-smi”

## Start ParaView Server

```
$ pvserver
```

```
INFO: underlay of /usr/bin/nvidia-smi required more than 50 (341) bind mounts
Waiting for client...
Connection URL: cs://instance-20221219-1118:11111
Accepting connection(s): instance-20221219-1118:11111
```

## Client Side

### Install the ParaView client

Download the exact same version (v5.11) of the ParaView client from the official download page <https://www.paraview.org/download/>. You can use Windows, Linux, and MacOS for client side OS.

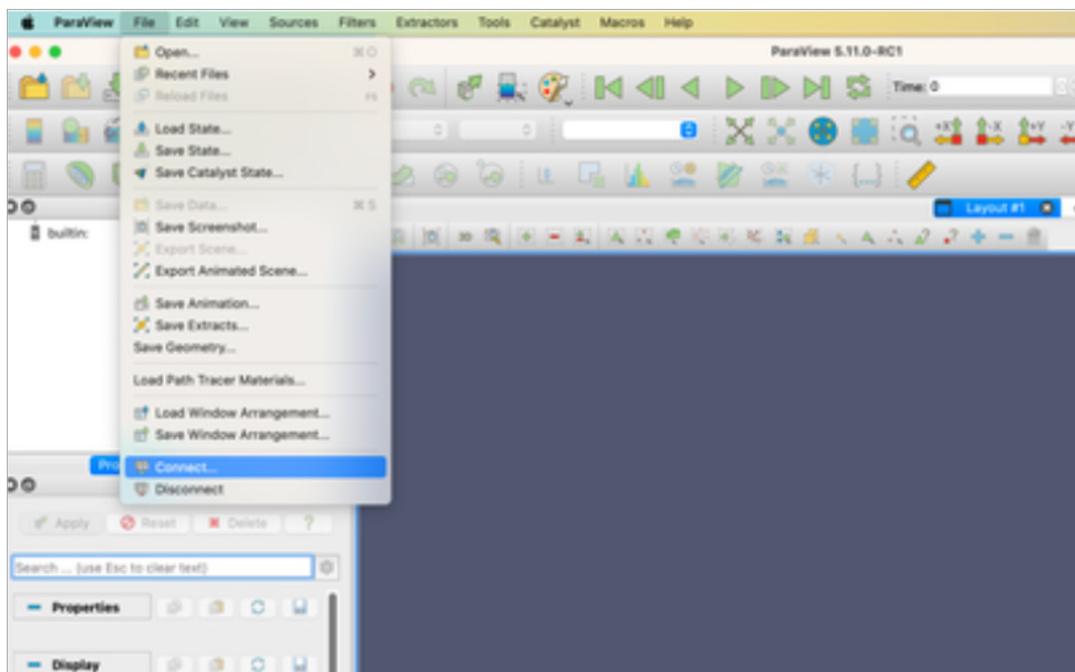
### SSH Port Forward

Before you connect to the ParaView server on OCI, forward port 11111 on the GPU instance to port 11111 on the local PC using SSH. This way, we don't need to open port 11111 on the OCI side.

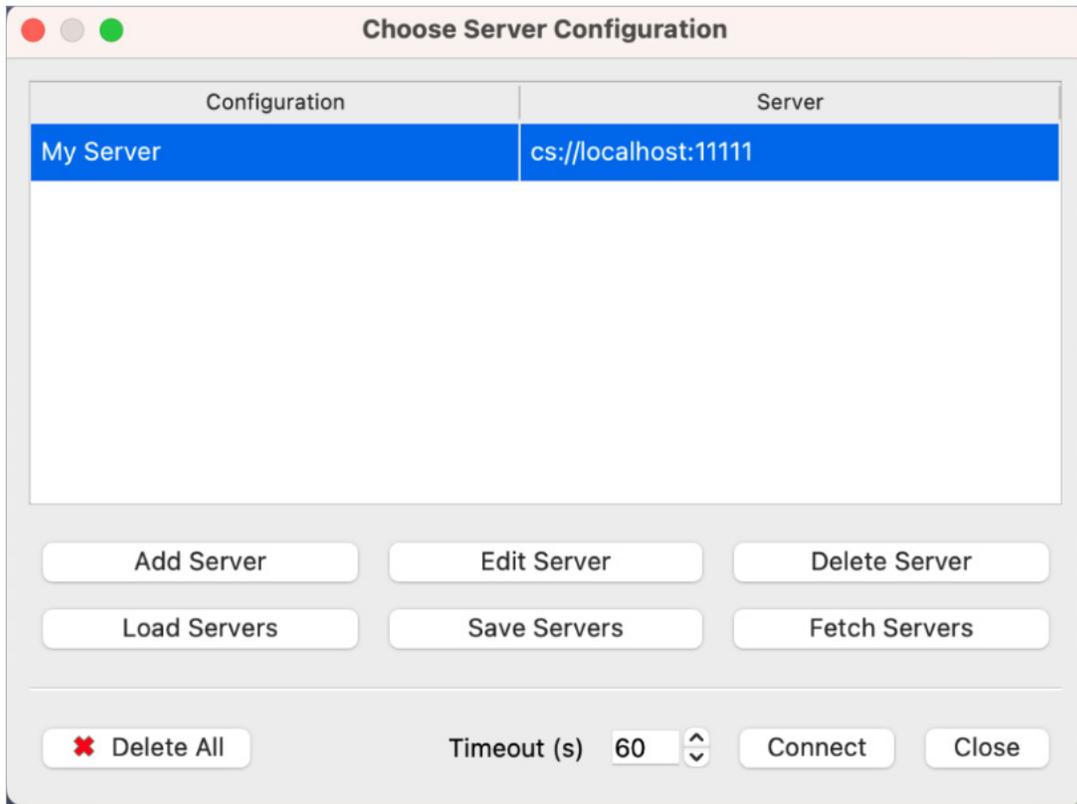
```
$ ssh -i ~/Downloads/ssh-key-2023-05-08.key -L 11111:localhost:11111 rocky@<INSTANCE IP HERE>
```

### Connect the ParaView client to the ParaView server on OCI

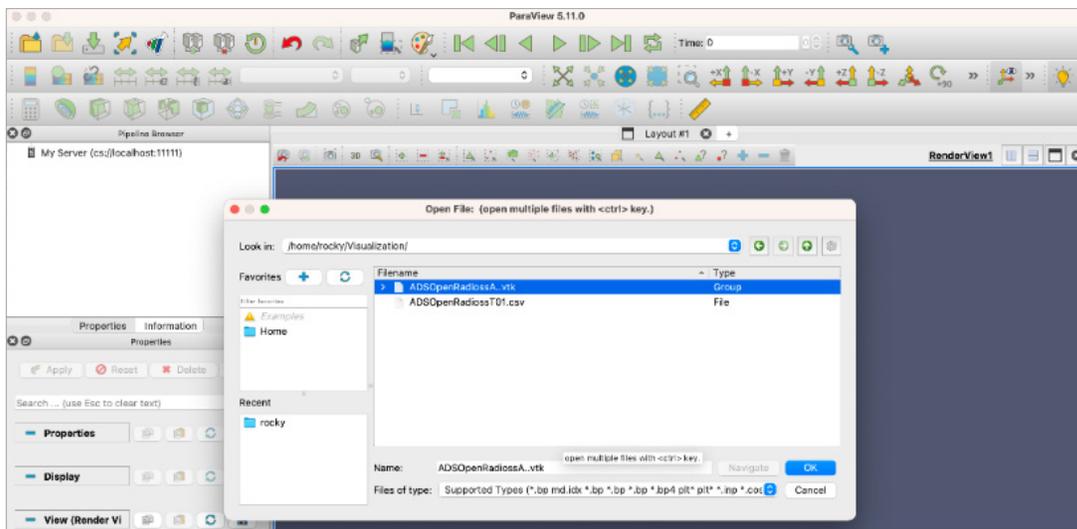
Launch ParaView and click **File > Connect**.



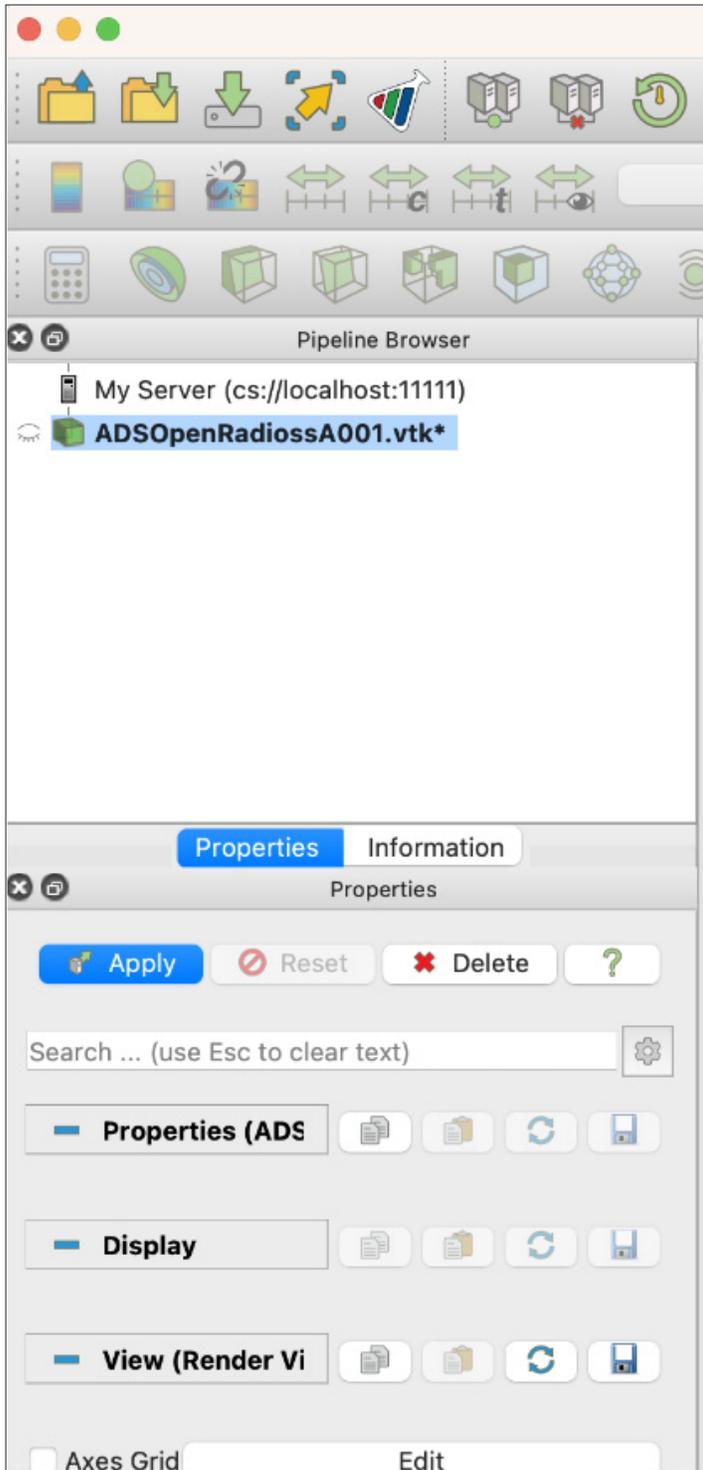
Connect to “localhost:1111”



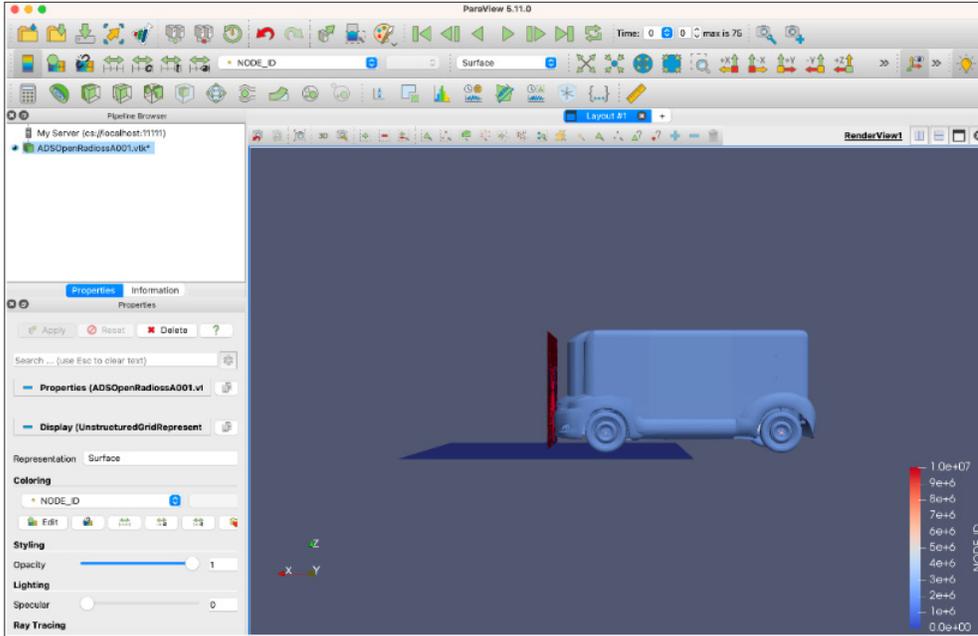
- Click **“Open”**
- Double Click **“Visualization”**
- Select **“ADSOpenRadiossA..vtk”** and Click **“OK”**



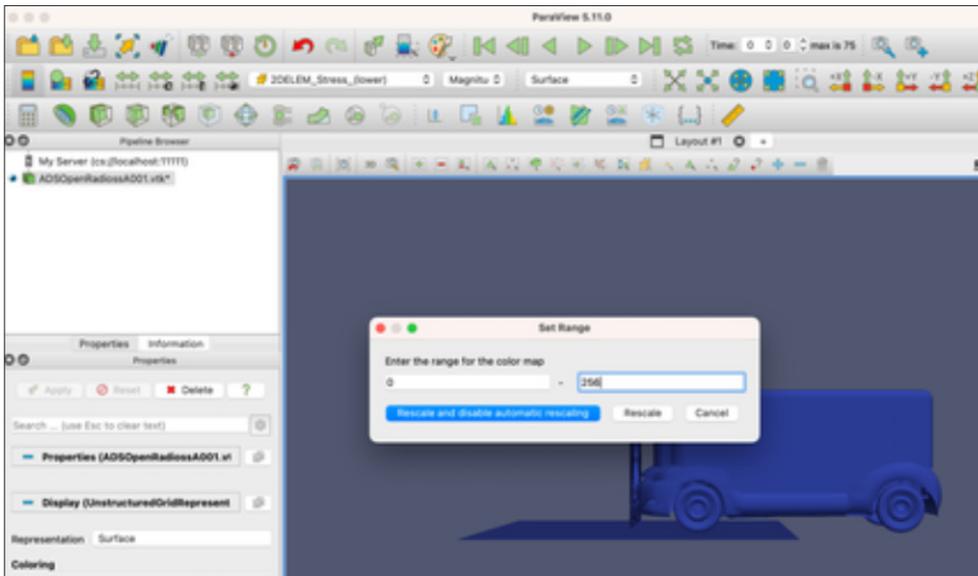
Click **Apply**



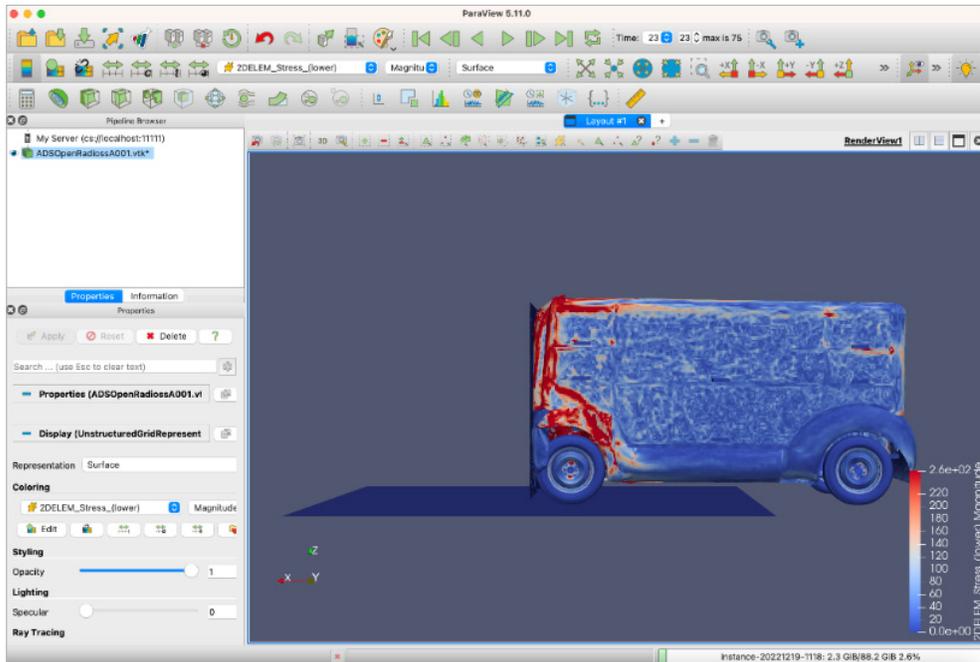
Now you can see the rendered VTK file.



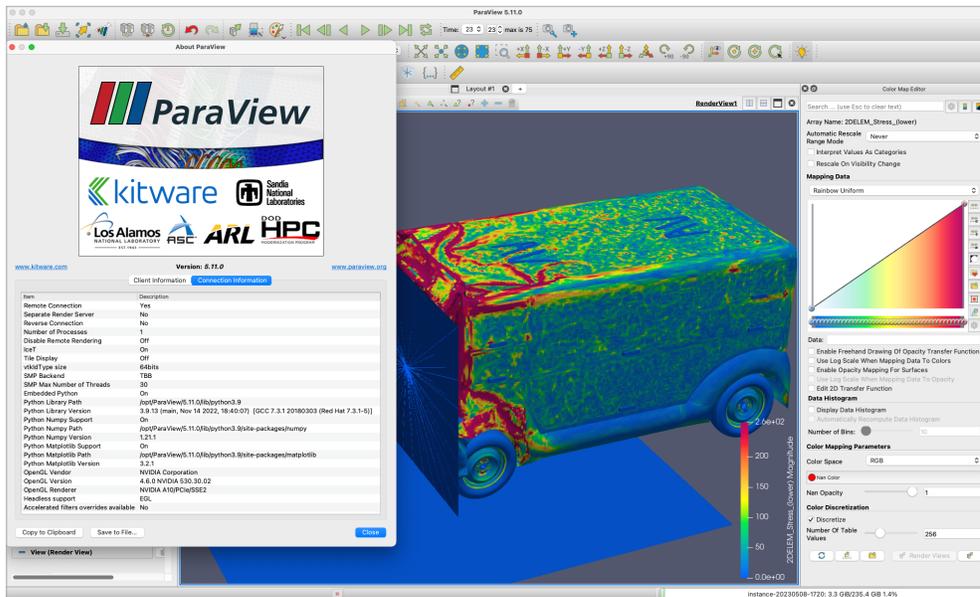
- Set view direction to “-Y”
- Select “2DLEM\_Stress\_(lower)”
- Set Range “0-256”



- Change **Time** to 23

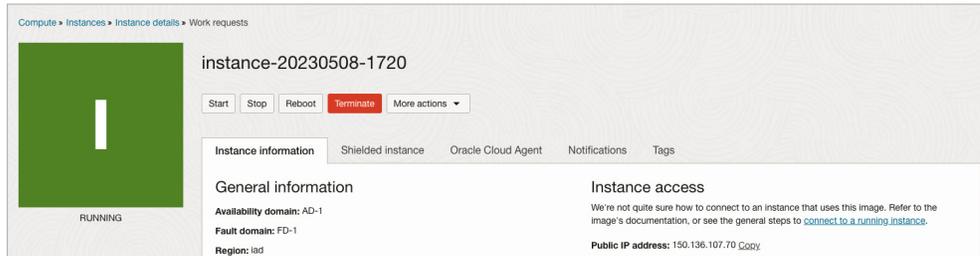


Lastly, click “ParaView” and then click “About ParaView” when you use MacOS; that shows the GPU device and GPU driver version that the ParaView server running on Oracle cloud resource is using for this session.



## Terminate the GPU Instance

After finishing the visualization, let's terminate the GPU instance. This time, we can delete the boot drive.



Check “Permanently delete the attached boot volume” and then click “Terminate instance”.



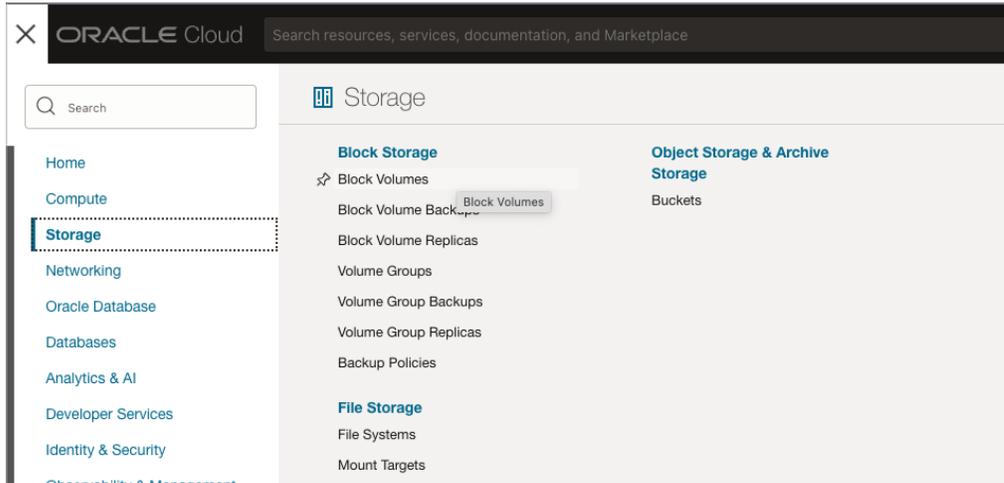
## Summary

We walked through the car crash simulation to its visualization using the newly published OpenRadioss / ParaView Rocky Linux 9-based image on OCI and appropriate shape for each task.

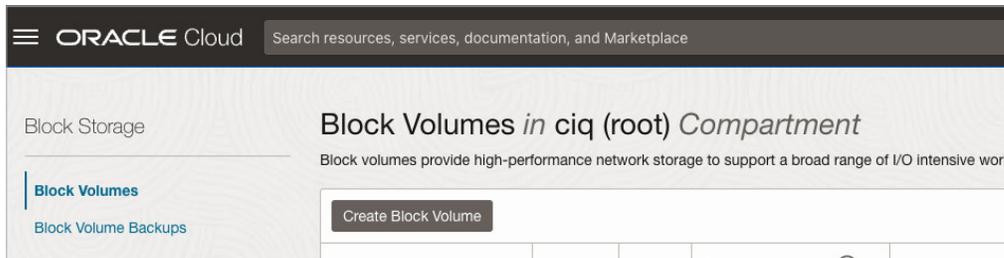
## Appendix A: Attaching extra drive

If you'd like to play with all of the examples in this image, understand that the default boot drive doesn't have enough space. In this section, we provide instructions on how to create extra drives and attach them to your instance.

Click the top left corner to open the menu, and then click "Storage", "Block Volumes".



Click "Create Block Volume"



Type a name for the block volume and then click "Create Block Volume". The default size of the block volume is 1024GB.

### Create block volume [Help](#)

Name:

Create In Compartment:

Availability Domain:

**Volume Size and Performance**

Default  Custom

**Volume Size:** 1024 GB

**Volume Performance:** Balanced

**IOPS:** 25,000 IOPS (60 IOPS/GB)

**Throughput:** 480 MB/s (480 KB/s/GB)

[Cancel](#)

Provisioning will take less than a few minutes.

[Block Storage](#) » [Block Volumes](#) » Block Volume Details



PROVISIONING...

## extra

**Block Volume Information** Tags

**Availability Domain:** zNpx:US-ASHBURN-AD-1

**Compartment:** ciq (root)

**OCID:** ...c2bltq [Show](#) [Copy](#)

**Created:** Tue, May 9, 2023, 07:54:35 UTC

**Size:** 1024 GB (i)

**Auto-tune Performance Policies**

When block volume becomes available state, we are good to go

Block Storage > Block Volumes > Block Volume Details



AVAILABLE

## extra

Edit
Move Resource
Add tags
Terminate

Block Volume Information
Tags

**Availability Domain:** zNpx:US-ASHBURN-AD-1

**Compartment:** ciq (root)

**OCID:** ...c2bltq [Show](#) [Copy](#)

**Created:** Tue, May 9, 2023, 07:54:35 UTC

**Size:** 1024 GB ⓘ

**Auto-tune Performance Policies**

Go back to the instance page and click "Attached block volumes" > "Attach block volume"

**Shape:** VM.Standard.E4.Flex

**OCPU count:** 64

**Network bandwidth (Gbps):** 40

**Memory (GB):** 64

**Local disk:** Block storage only

In-

Se

Me

Tru

Co

Resources

- Metrics
- Quick actions
- Attached block volumes
- Attached VNICS
- Boot volume
- Console connection

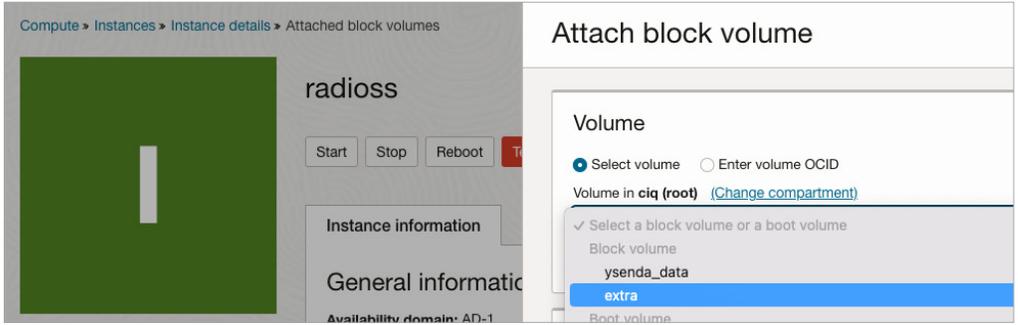
### Attached block volumes

[Block volumes](#) provide high-performance network storage to support a broad range of I/O intensiv

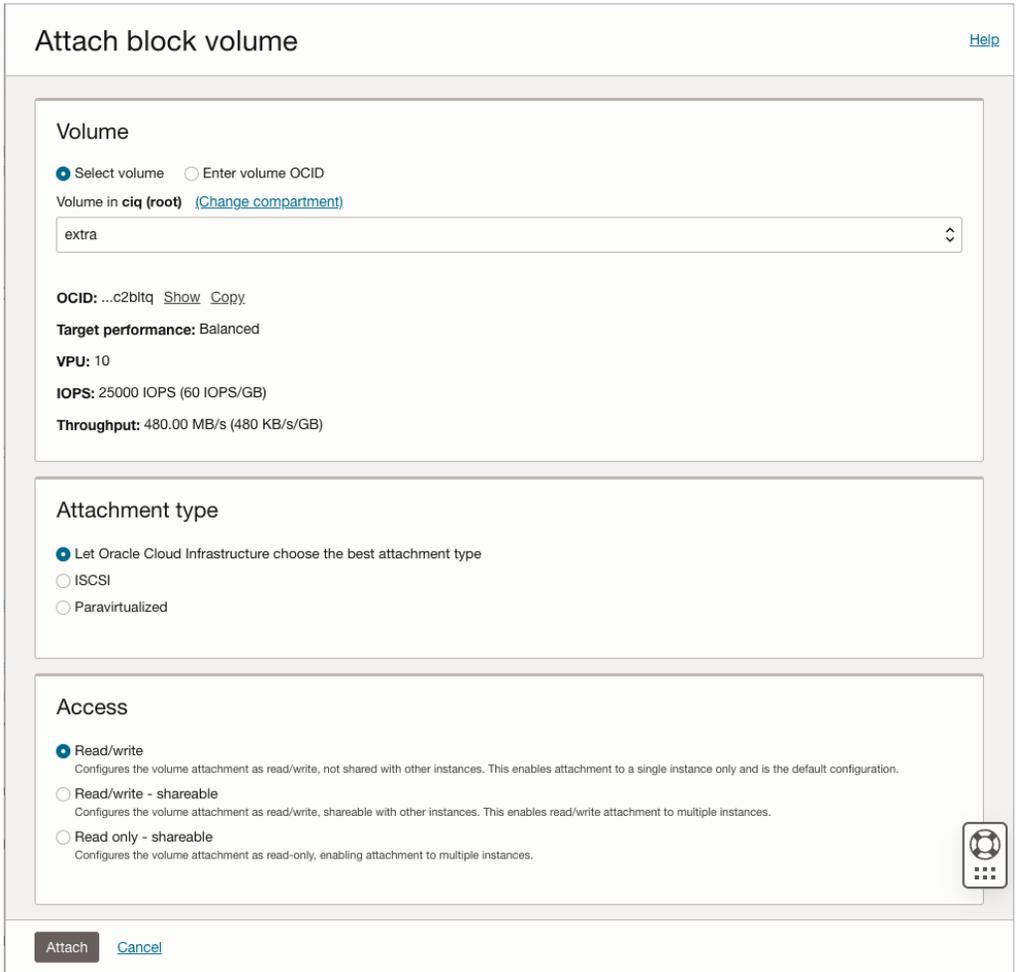
Attach block volume

Name	State	Volume type	Device path	Type
There are no block volumes at				

Choose the drive you created



Click "Attach"



Attaching the block volumes takes about a minute.

**Attached block volumes**  
[Block volumes](#) provide high-performance network storage to support a broad range of I/O intensive workloads.

Attach block volume

Name	State	Volume type	Device path	Type	Access	Size	VPU	Multipath	Attached
<a href="#">extra</a>	● Attaching	Block volume	-	iscsi	Read/write	1 TB	10	No	Tue, May 9, 2023, 08:00:37 UTC

Showing 1 item < 1 of 1 >

It's ready!

**Attached block volumes**  
[Block volumes](#) provide high-performance network storage to support a broad range of I/O intensive workloads.

Attach block volume

Name	State	Volume type	Device path	Type	Access	Size	VPU	Multipath	Attached
<a href="#">extra</a>	● Attached	Block volume	-	iscsi	Read/write	1 TB	10	No	Tue, May 9, 2023, 08:00:37 UTC

Showing 1 item < 1 of 1 >

Click menu on the right, and click "iSCSI commands & information".

**Attached block volumes**  
[Block volumes](#) provide high-performance network storage to support a broad range of I/O intensive workloads.

Attach block volume

Name	State	Volume type	Device path	Type	Access	Size	VPU	Multipath	Attached
<a href="#">extra</a>	● Attached	Block volume	-	iscsi	Read/write	1 TB	10	No	Tue, M

- View block volume details
- ISCSI commands & information
- Copy attachment OCID
- Copy resource OCID
- Detach

Copy command to "connect".

## iSCSI commands & information [Help](#)

Linux

- You must unmount the drive and then run the disconnect commands before detaching or instance reboot will fail.
- If you add this volume to the instance's etc/fstab file to automatically mount on boot, you must include the `_netdev` and `nofail` options.

Connect

```
sudo iscsiadm -m node -o new -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-
sudo iscsiadm -m node -o update -T iqn.2015-12.com.oracleiaas:c534eee8-b0
sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-ak
```

Copy.

Disconnect

```
sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-ak
sudo iscsiadm -m node -o delete -T iqn.2015-12.com.oracleiaas:c534eee8-b0
```

Copy.

**IP address and port:** 169.254.2.2:3260 [Copy](#)

**Volume IQN:** iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f [Copy](#)

> Windows

Close

Go back to your terminal and paste the command:

```
[rocky@radioss ~]$ sudo iscsiadm -m node -o new -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
sudo iscsiadm -m node -o update -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
iscsiadm: using tgt 1 from existing record
New iSCSI node [tcp:[hw=,ip=,net_if=,iscsi_if=default] 169.254.2.2,3260,]
[rocky@radioss ~]$
```

Check that the disk attached correctly using “lsblk”

```
[rocky@radioss ~]$ lsblk
NAME                MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINTS
sda                  8:0    0 46.6G  0  disk
├─sda1                8:1    0   99M  0  part /boot/efi
├─sda2                8:2    0 1000M  0  part /boot
├─sda3                8:3    0    4M   0  part
├─sda4                8:4    0    1M   0  part
├─sda5                8:5    0 45.5G  0  part
└─┬─rocky-root        253:0  0 45.5G  0  lvm  /
   └─
```

Create partition using “parted”

```
$ sudo parted /dev/sdb
(parted) mklabel gpt
(parted) mkpart primary 0% 100%
(parted) quit
```

Create filesystem using “mkfs.ext4”

```
$ sudo mkfs.ext4 /dev/sdb1
```

Create mount point “/data”

```
$ sudo mkdir -p /data
```

Mount extra drive to “/data”

```
$ sudo mount /dev/sdb1 /data
```

Change permission to 777

```
$ sudo chmod 777 /data
```

Now that you have enough space on “/data” to try all examples under “~/Simulations”, copy examples from “~/Simulations” to “/data” and then try the examples.

## Appendix B: Detaching extra drive

Unmount “/data”

```
$ sudo umount /data
```

Open the instance page, and click “Attached block volumes”. Open the menu and click “iSCSI commands & information”.

### Attached block volumes

[Block volumes](#) provide high-performance network storage to support a broad range of I/O intensive workloads.

[Attach block volume](#)

Name	State	Volume type	Device path	Type	Access	Size	VPU	Multipath	Attached
<a href="#">extra</a>	Attaching	Block volume	-	iscsi	Read/write	1 TB	10	No	Tue, May 9, 2023, 08:00:37 UTC

Showing 1 item < 1 of 1 >

Copy the “Disconnect” command

## iSCSI commands & information [Help](#)

Linux

**!**

- You must unmount the drive and then run the disconnect commands before detaching or instance reboot will fail.
- If you add this volume to the instance's etc/fstab file to automatically mount on boot, you must include the `_netdev` and `nofail` options.

**Connect**

```
sudo iscsiadm -m node -o new -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
sudo iscsiadm -m node -o update -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
```

[Copy](#)

**Disconnect**

```
sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
sudo iscsiadm -m node -o delete -T iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f
```

[Copy](#)

**IP address and port:** 169.254.2.2:3260 [Copy](#)

**Volume IQN:** iqn.2015-12.com.oracleiaas:c534eee8-b0dc-4713-abe5-3396a5657e6f [Copy](#)

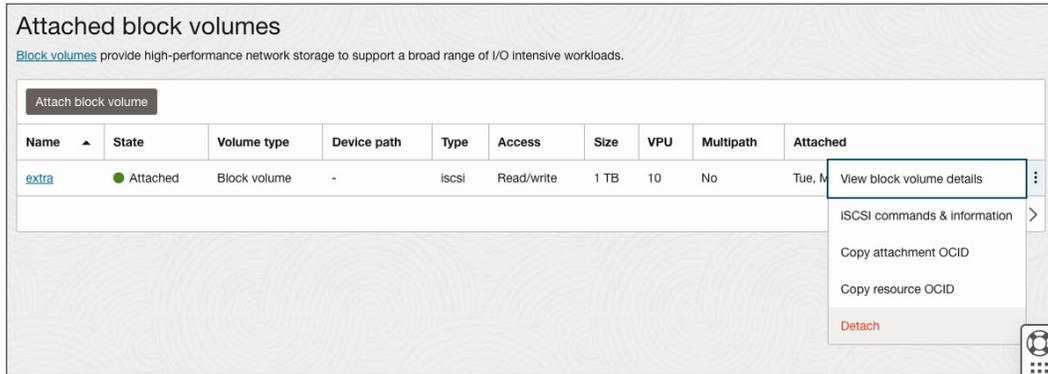
> Windows

[Close](#)

Paste “Disconnect” command to terminal

```
[rocky@radioss ~]$ sudo iscsiadm -m node -T ign.2015-12.com.oracle
sudo iscsiadm -m node -o delete -T ign.2015-12.com.oracleiaas:c53
```

Click “Detach”



## Appendix C: Using openradioss.sif with /data

Copy example from “~/Simulations” to “/data”

```
$ cp ~/Simulations/Neon1m11_2017.zip /data
```

Unarchive zip file

```
$ cd /data
$ unzip Neon1m11_2017.zip
$ cd Neon1m11_2017
```

Apptainer shell command with bind mount option to bind mount “/data” to openradioss container

```
$ apptainer shell -B /data /usr/local/bin/openradioss.sif
Apptainer> pwd
/data/Neon1m11_2017
Apptainer>
```

Running a Neon 1M benchmark example takes about an hour to finish.

```
Apptainer> starter_linux64_gf -i NEON1M11_0000.rad -np 64
Apptainer> mpirun -np 64 engine_linux64_gf_ompi -i NEON1M11_0001.rad
```

When the simulation is finished, you will see an output similar to this:

```

** CUMULATIVE CPU TIME SUMMARY **

CONTACT SORTING.....: .1063E+05      8.07 %
CONTACT FORCES.....: .1297E+04      0.98 %
```



**\*\* CUMULATIVE CPU TIME SUMMARY \*\***

```
CONTACT SORTING.....: .2968E+05    14.60 %
CONTACT FORCES.....: .2441E+05    12.01 %
ELEMENT FORCES.....: .9330E+05    45.91 %
KINEMATIC COND.....: .2055E+04     1.01 %
INTEGRATION.....: .1684E+05     8.28 %
ASSEMBLING.....: .1068E+05     5.25 %
OTHERS (including I/O).....: .2628E+05    12.93 %
TOTAL.....: .2032E+06   100.00 %
```

**\*\* MEMORY USAGE STATISTICS \*\***

```
TOTAL MEMORY USED .....:      59767 MB
MAXIMUM MEMORY PER PROCESSOR.....:      4070 MB
MINIMUM MEMORY PER PROCESSOR.....:       831 MB
AVERAGE MEMORY PER PROCESSOR.....:       933 MB
```

**\*\* DISK USAGE STATISTICS \*\***

```
TOTAL DISK SPACE USED .....:      17730 MB
ANIMATION/H3D/TH/OUTP SIZE .....:       1202 MB
RESTART FILE SIZE .....:      16528 MB
```

```
ELAPSED TIME      =      3226.62 s
                   0:53:46
ESTIMATED SPEEDUP=      62.99
```

NORMAL TERMINATION

TOTAL NUMBER OF CYCLES : 10103

Running a Toyota Camry model takes about 2 hours to finish.

```
Apptainer> cp ~/Simulations/2012-toyota-camry-detailed-v5a.zip /data
Apptainer> cp ~/Simulations/CamryOpenRadioss.key
Apptainer> cd /data
Apptainer> unzip 2012-toyota-camry-detailed-v5a.zip
Apptainer> cp CamryOpenRadioss.key 2012-toyota-camry-detailed-v5a
Apptainer> cd 2012-toyota-camry-detailed-v5a
Apptainer> starter_linux64_gf -i CamryOpenRadioss.key -np 64
Apptainer> mpirun -np 64 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
```

```
01[00.] 8[100.] 16[100.] 24[100.] 32[100.] 40[00.] 48[100.] 56[00.] 64[00.] 72[100.] 80[100.] 88[100.] 96[00.] 104[00.] 112[00.] 120[00.]
11[00.] 9[00.] 17[00.] 25[00.] 33[00.] 41[100.] 49[00.] 57[100.] 65[100.] 73[00.] 81[00.] 89[00.] 97[100.] 105[100.] 113[100.] 121[100.]
21[00.] 10[100.] 18[00.] 26[100.] 34[00.] 42[00.] 50[00.] 58[100.] 66[00.] 74[100.] 82[100.] 90[100.] 98[100.] 106[00.] 114[00.] 122[100.]
31[00.] 11[00.] 19[100.] 27[00.] 35[100.] 43[100.] 51[100.] 59[00.] 67[100.] 75[00.] 83[00.] 91[00.] 99[00.] 107[100.] 115[100.] 123[00.]
41[00.] 12[00.] 20[100.] 28[00.] 36[00.] 44[00.] 52[100.] 60[100.] 68[00.] 76[00.] 84[00.] 92[100.] 100[100.] 108[100.] 116[100.] 124[100.]
51[00.] 13[100.] 21[00.] 29[100.] 37[100.] 45[100.] 53[00.] 61[00.] 69[100.] 77[100.] 85[100.] 93[00.] 101[00.] 109[00.] 117[00.] 125[00.]
61[00.] 14[100.] 22[100.] 30[100.] 38[00.] 46[00.] 54[00.] 62[00.] 70[00.] 78[00.] 86[00.] 94[100.] 102[100.] 110[100.] 118[00.] 126[00.]
71[00.] 15[00.] 23[00.] 31[00.] 39[100.] 47[100.] 55[100.] 63[100.] 71[100.] 79[100.] 87[100.] 95[00.] 103[00.] 111[00.] 119[100.] 127[100.]
Mem|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Swap|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Main|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
PID USER PRI NI VIRT RES SHR S CPU%MEM% TIME+ Command
24112 rocky 20 0 825M 295M 47372 R 100.6 0.5 0:17.32 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
24114 rocky 20 0 819M 296M 48448 R 99.9 0.5 0:17.32 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
24115 rocky 20 0 825M 290M 43672 R 99.9 0.5 0:17.32 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
24116 rocky 20 0 842M 319M 47624 R 99.9 0.5 0:17.31 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
24117 rocky 20 0 809M 285M 47592 R 99.9 0.4 0:17.32 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
24129 rocky 20 0 812M 287M 47088 R 99.9 0.5 0:17.33 engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
```

When the simulation is finished, you will see an output similar to this:

**\*\* CUMULATIVE CPU TIME SUMMARY \*\***

```

CONTACT SORTING.....: .5613E+05    10.56 %
CONTACT FORCES.....: .1404E+06    26.40 %
..INCLUDING CONTACT NORMALS: .6792E+05    12.77 %
ELEMENT FORCES.....: .1793E+06    33.71 %
KINEMATIC COND.....: .1355E+05     2.55 %
INTEGRATION.....: .2393E+05     4.50 %
ASSEMBLING.....: .3340E+05     6.28 %
OTHERS (including I/O).....: .8511E+05    16.01 %
TOTAL.....: .5318E+06   100.00 %
  
```

**\*\* MEMORY USAGE STATISTICS \*\***

```

TOTAL MEMORY USED .....:      30993 MB
MAXIMUM MEMORY PER PROCESSOR.....:      563 MB
MINIMUM MEMORY PER PROCESSOR.....:      453 MB
AVERAGE MEMORY PER PROCESSOR.....:      484 MB
  
```

**\*\* DISK USAGE STATISTICS \*\***

```

TOTAL DISK SPACE USED .....:      4958 MB
ANIMATION/H3D/TH/OUTP SIZE .....:      4958 MB
RESTART FILE SIZE .....:           0 MB
  
```

```

ELAPSED TIME      =      8324.31 s
                   2:18:44
ESTIMATED SPEEDUP=      63.88
  
```

```

NORMAL TERMINATION
USER BREAK
TOTAL NUMBER OF CYCLES :    84061
  
```

Running a Toyota Yaris model takes about 3 hours to finish.

```

Apptainer> cp ~/Simulations/2010-toyota-yaris-detailed-v2j.zip /data
Apptainer> cp ~/Simulations/YarisOpenRadioss.key
Apptainer> cd /data
Apptainer> unzip 2010-toyota-yaris-detailed-v2j.zip
Apptainer> cp YarisOpenRadioss.key 2010-toyota-yaris-detailed-v2j
Apptainer> cd 2010-toyota-yaris-detailed-v2j
Apptainer> starter_linux64_gf -i YarisOpenRadioss.key -np 64
Apptainer> mpirun -np 64 engine_linux64_gf_ompi -i YarisOpenRadioss_0001.rad
  
```

```

0[0.0] 8[100] 16[0.0] 24[1.0] 32[0.0] 40[76.1] 48[0.0] 56[100] 64[100.0] 72[0.0] 80[0.0] 88[0.0] 96[0.0] 104[0.0] 112[0.0] 120[0.0]
1[100] 9[1.0] 17[100] 25[100] 33[100] 41[2.0] 49[100] 57[0.0] 65[0.0] 73[100.0] 81[100.0] 89[100.0] 97[100.0] 105[100.0] 113[100.0] 121[100.0]
2[0.0] 10[100] 18[0.0] 26[0.0] 34[100] 42[0.0] 50[0.0] 58[0.0] 66[100.0] 74[100.0] 82[100.0] 90[0.0] 98[0.0] 106[0.0] 114[100.0] 122[0.0]
3[100] 11[0.0] 19[100] 27[100] 35[0.0] 43[100] 51[100] 59[100] 67[0.0] 75[0.0] 83[0.0] 91[100.0] 99[100.0] 107[100.0] 115[0.0] 123[100.0]
4[100] 12[100] 20[100] 28[100] 36[100] 44[100] 52[0.0] 60[0.0] 68[5.0] 76[100.0] 84[100.0] 92[100.0] 100[0.0] 108[0.0] 116[0.0] 124[100.0]
5[0.0] 13[0.0] 21[0.0] 29[0.0] 37[0.0] 45[0.0] 53[100] 61[100] 69[100.0] 77[0.0] 85[0.0] 93[0.0] 101[100.0] 109[100.0] 117[100.0] 125[0.0]
6[100] 14[100] 22[100] 30[0.0] 38[0.0] 46[0.0] 54[100] 62[100] 70[0.0] 78[100.0] 86[100.0] 94[100.0] 102[0.0] 110[100.0] 118[100.0] 126[100.0]
7[0.0] 15[0.0] 23[0.0] 31[100] 39[100] 47[100] 55[0.0] 63[0.0] 71[100.0] 79[0.0] 87[0.0] 95[0.0] 103[100.0] 111[0.0] 119[0.0] 127[0.0]
Mem[|||||] 12.80/83.20 Tasks: 98, 170 thr 1113 kba; 65 running
Swp[|] 28.0M/7.2M Load average: 53.89 19.90 8.31
Uptime: 13:35:55

Main [|||||]
PID Usrsk PRI NI VIRT RES SHR S CPU%MEM% TIME+ Command
29949 rocky 20 0 753M 331M 146M R 100.3 0.5 1:46.29 engine_linux64 gf omp -i YarisOpenRadioss_0001.rad
30083 rocky 20 0 777M 351M 146M R 100.3 0.6 1:46.32 engine_linux64 gf omp -i YarisOpenRadioss_0001.rad
29946 rocky 20 0 757M 335M 149M R 99.7 0.5 1:46.28 engine_linux64 gf omp -i YarisOpenRadioss_0001.rad
29947 rocky 20 0 756M 329M 147M R 99.7 0.5 1:46.27 engine_linux64 gf omp -i YarisOpenRadioss_0001.rad
29948 rocky 20 0 784M 363M 150M R 99.7 0.6 1:46.26 engine_linux64 gf omp -i YarisOpenRadioss_0001.rad
29950 rocky 20 0 754M 331M 148M R 99.7 0.5 1:46.28 engine_linux64 gf omp -i YarisOpenRadioss_0001.rad

```

When the simulation is finished, you will see an output similar to this:

```

** CUMULATIVE CPU TIME SUMMARY **

CONTACT SORTING.....: .8854E+05      11.71 %
CONTACT FORCES.....: .1441E+06      19.06 %
..INCLUDING CONTACT NORMALS: .8145E+05      10.77 %
ELEMENT FORCES.....: .2944E+06      38.94 %
KINEMATIC COND.....: .1924E+05       2.54 %
INTEGRATION.....: .2373E+05       3.14 %
ASSEMBLING.....: .5593E+05       7.40 %
OTHERS (including I/O).....: .1301E+06      17.20 %
TOTAL.....: .7561E+06     100.00 %

** MEMORY USAGE STATISTICS **

TOTAL MEMORY USED .....:      25495 MB
MAXIMUM MEMORY PER PROCESSOR.....:      450 MB
MINIMUM MEMORY PER PROCESSOR.....:      375 MB
AVERAGE MEMORY PER PROCESSOR.....:      398 MB

** DISK USAGE STATISTICS **

TOTAL DISK SPACE USED .....:      9194 MB
ANIMATION/H3D/TH/OUTP SIZE .....:      5386 MB
RESTART FILE SIZE .....:      3808 MB

ELAPSED TIME      =      11840.29 s
                    3:17:20
ESTIMATED SPEEDUP=      63.86

NORMAL TERMINATION
TOTAL NUMBER OF CYCLES : 200848

```



560 Mill Street, Suite 302, Reno, NV 89502

Phone: 800 220 5243 Email: [info@ciq.com](mailto:info@ciq.com) [www.ciq.com](http://www.ciq.com)