



Stable Diffusion Deployed with Apptainer on Rocky Linux in
the Oracle Cloud

Ctrl IQ, Inc.

ciq.co



The Leader in
Enterprise HPC & Enterprise Linux

**Secure, Stable, Trusted
Enterprise Linux**



ROCKY LINUX

**Application Containers for High
Performance Computing**



APPTAINER

**Cluster Management
and Provisioning**



WAREWULF

**Cloud-Native Federated
Computing Platform**



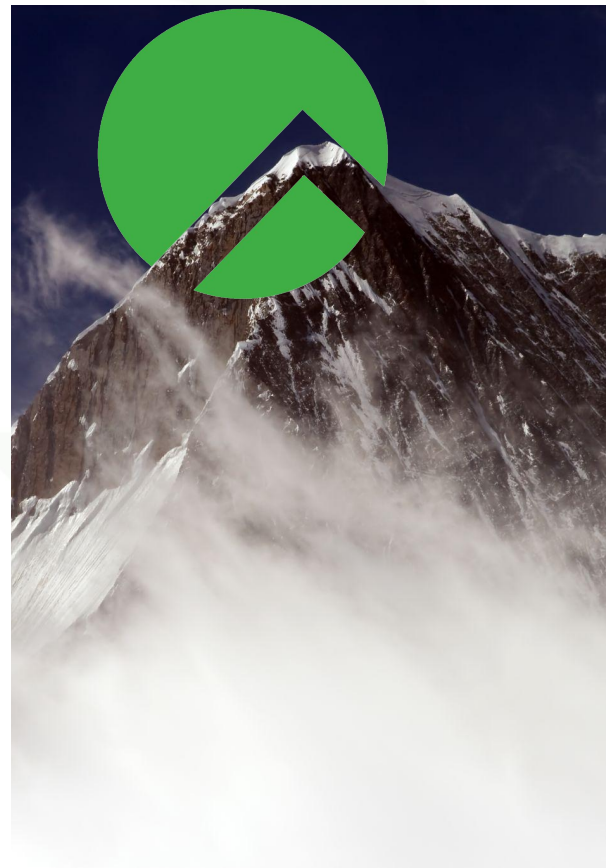
FUZZBALL

Rocky Linux

Community Enterprise Linux

As CentOS comes to its end of life, Rocky Linux is there as the trusted successor. By design it is a freely available, community driven, bug-for-bug compatible version of Enterprise Linux.

Rocky Linux is quickly becoming the dominant operating system for the enterprise & HPC.



Rocky Linux

Features

- **Community Enterprise Linux:** The Rocky Enterprise Software Foundation is designed to be a community driven, controlled, and architected OS that will stand the test of time.
- **Compatibility:** Rocky Linux is 100% compatible with Enterprise Linux which makes it a simple, drop-in solution.
- **Stability:** A community of vested individuals, organizations and enterprises of every shape and size provides long-term stability.
- **Every Cloud, Every OEM, Every ISV:** Our vision from the beginning is to have every cloud, every OEM, every component and every ISV vested, compatible and partnering for the good of everyone.
- **The Ethos of CIQ & The Name of Rocky:** CIQ is committed to empowering everyone to do what they do great. Rocky McGaugh, CentOS co-founder and namesake of Rocky Linux, was great at Linux and it is important that everyone know their contributions make a difference in this world.

Compatible

Rocky Linux is designed to be 100% compatible with the Enterprise Linux family of distributions

Open Source

The Rocky Enterprise Software Foundation (RESF) is the organization behind Rocky Linux, community led, supported, and maintained.

Security

Rocky Linux is all about security and compliance with accreditations like CIS, Nessus, RESF Secure Boot, and CIQ sponsoring FIPS certification.



Oracle Cloud support

- Images for Rocky Linux 8 and 9
- ARM support coming very soon
- *Big thanks for having me here to speak!*
- Great time working on OCI

The Oracle Cloud logo is positioned in the bottom right corner of the slide. It features the word "ORACLE" in a bold, red, sans-serif font, with the word "Cloud" in a smaller, black, sans-serif font directly below it. The background of the slide includes a large, faint, light green circular graphic that partially overlaps the logo area.

ORACLE
Cloud

More Rocky info at CloudWorld!

- *Colleague Jonathon Anderson will be giving a talk about Rocky with many more exciting updates and a demo of a migration path to Rocky Thursday, 10:15 AM - 11:00 AM PDT!*

Apptainer / Singularity

Containers for HPC

Kurtzer founded Singularity in 2015, and since then, it has risen to being one of the most utilized tools in HPC.

It has now been moved into the Linux Foundation and renamed to Apptainer.

CIQ, is the official commercial support arm of Apptainer.



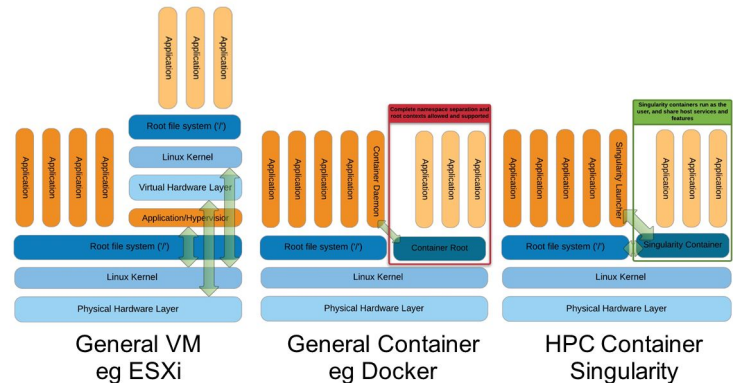
As Docker brought containers into enterprise, Singularity brought containers into HPC with a containerization strategy that just works for HPC architectures, security models, and use-cases.

Now Singularity is part of the Linux Foundation and re-released under a new name for the open source project, Apptainer.



Apptainer - why?

- HPC-focused containerization platform...
 - Used to be Singularity!
 - General history
 - Move into Linux Foundation
 - How did I get involved?
- Technical distinctions
 - Security model allows users to build their own untrusted containers in a trusted manner
 - “Integration over isolation” with host OS to provide access to accelerators, MPI, and other HPC stacks
 - Makes reproducibility, onboarding, container encryption, and container signing easy for academia, private industry
 - For users...
 - ...and for sysadmins?
- Mature technology and part of The Linux Foundation



Apptainer - why?

- ...Making model deployment simple
 - Easy to package and deploy Stable Diffusion (and other models) with Apptainer on cloud architectures like the Oracle Cloud Infrastructure for inference
 - Easy for HPC cluster users to pull once and use as an executable
- Rootless!
 - *New development!*
- Compatible with CI/CD systems
- CRI with Kubernetes
- Future developments:
 - Checkpointing for containerized MPI applications with DMTCP
 -

Apptainer - who?

- NIH
- SDSC
- ALCF
- ORNL
- TACC
- ...greater than 25,000 different institutions
- Many in private enterprise and elsewhere

The Apptainer community

- Apptainer.org
- Community meeting 8 AM PDT first Tuesday of the month
- Mailing lists
- Slack
- Github
 - <https://github.com/apptainer/apptainer>

Stable Diffusion - what?

- AI model trained across millions of image/description pairs that can reliably generate (mostly) arbitrary images
 - LAION
- Like DALL·E 2 or Midjourney in terms of results - but...
- ...open source?!?!?
 - Anyone can deploy with *just* **6 GB** VRAM???
 - ...Needs more for most detailed and fastest predictions
 - Make it at least 10-12 GB, 16 GB preferred!
- Diffusion-based model
 - Adds noise, learns to remove noise
 - Will leave the heavy explanation to your nearest AI/ML engineer

Stable Diffusion - *what?!?*

- Weights controversy
 - Destruction of artist communities?
 - Copyright issues?
- Statement of ethics
 - **Don't be evil!**
- Githubs:
 - <https://github.com/CompVis/stable-diffusion>
 - <https://github.com/jquesnelle/txt2imghd>
 - <https://github.com/jquesnelle/txt2imghd>

The tech!

Our definition file

- Weights included... and then not included for T&C respect
- Utilizes standard conda environment provided by Stable Diffusion authors - note “source...”
- Single run of each pipeline done inside of container to package downloaded components with container so they aren’t reached for at model runtime
- Runscript allows for transparent use of container as application
- Other useful AI art tools in the Stable Diffusion family also packaged here
- Overall most difficult part was finding a solution to conda environments in a container
 - Notice no activation of environment necessary

```

Bootstrap: docker
From: rockylinux:9

%environment
source /miniconda3/etc/profile.d/conda.sh
conda activate ldm
export PATH=/usr/local/cuda-11.8/bin:${PATH}

%files
/data/sd-v1-4-full-ema.ckpt /data/sd-v1-4-full-ema.ckpt
/data/test.jpg /data/test.jpg

%post

dnf -y update
dnf -y groupinstall "Development Tools"
dnf -y install git wget epel-release strace unzip iptables iptables-devel

dnf -y config-manager --add-repo http://developer.download.nvidia.com/compute/cuda/repos/rhel9/x86_64/cuda-rhel9.repo \
  && dnf -y clean all \
  && dnf -y module install nvidia-driver:latest-dkms \
  && dnf -y install cuda \
  && export PATH=/usr/local/cuda-11.8/bin:${PATH}

cd /
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash ./Miniconda3-latest-Linux-x86_64.sh -b -p /miniconda3
export PATH=/miniconda3/bin:$PATH
conda init bash
source /miniconda3/etc/profile.d/conda.sh

cd /
git clone https://github.com/CompVis/stable-diffusion.git
git clone https://github.com/jquesnelle/txt2imghd.git
mv /txt2imghd/txt2imghd.py /stable-diffusion/scripts
rm -rf /txt2imghd
cd stable-diffusion

wget https://github.com/xinntao/Real-ESRGAN/releases/download/v0.2.5.0/realesrgan-ncnn-vulkan-20220424-ubuntu.zip \
  && unzip ./realesrgan-ncnn-vulkan-20220424-ubuntu.zip \
  && rm -rf *.mp4 *.jpg ./models/*anime* \
  && chmod +x ./realesrgan-ncnn-vulkan

conda update -n base -c defaults conda
conda env create -f environment.yaml
mkdir -p models/ldm/stable-diffusion-v1/
ln -s /data/sd-v1-4-full-ema.ckpt models/ldm/stable-diffusion-v1/model1.ckpt
conda activate ldm

python3 scripts/txt2img.py --plms --prompt 'test'
python3 scripts/img2img.py --prompt 'sunset' --init-img /data/test.jpg
python3 scripts/txt2imghd.py --realesrgan /stable-diffusion/realesrgan-ncnn-vulkan --prompt 'A massive city in a cavern deep
underground, many lights from houses in the city, cobblestone roads, castle at one side of the city, dark, steampunk, grungy, dim,
oil painting'
rm -rf outputs/*
rm /data/sd-v1-4-full-ema.ckpt
rm /data/test.jpg

%runscript
cd /stable-diffusion && python3 scripts/txt2img.py --plms --prompt

```


Testing the instance/def file...

- We boot a standard Oracle Cloud Infrastructure VM.GPU3.1 shape with the Rocky Linux 8.6 image
- Expand root FS with `oci-utils oci-growfs-bash`
- Install NVIDIA drivers and CUDA toolkit via the instructions NVIDIA has in their installation guides
- Test for GPU functionality - “`nvidia-smi`”
- Build container from `.def`
- Test *inside container* for GPU functionality
 - Working! Cool!
- Let’s generate some images!


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


Rocky Linux 8.6
Rocky Linux
Change image

Shape


VM.GPU3.1
Virtual machine, 6 core OCPU, 90 GB memory, 4 Gbps network bandwidth
Change shape

[Show advanced options](#)

```
[rocky@instance-20221011-1307 libexec]$ sudo ./oci-growfs-bash
```

3.3. RHEL 8 / Rocky 8

3.3.1. Prepare RHEL 8 / Rocky 8

```
[rocky@instance-20221011-1307 ~]$ apptainer build --fakeroot --nv stable-diffusion.sif stable-diffusion.def
```

```
[rocky@instance-20221011-1307 ~]$ nvidia-smi
Sun Oct 16 19:09:59 2022

+-----+
| NVIDIA-SMI 520.61.05   Driver Version: 520.61.05   CUDA Version: 11.8   |
+-----+
| 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   Tesla V100-SXM2...  Off      | 00000000:00:04:0  Off |   0%        Default |
| N/A   42C    P0     37W / 300W    | 0MiB / 16384MiB |           | N/A   |
+-----+-----+

Processes:
+-----+
| GPU   GI   CI        PID   Type   Process name                      GPU Memory |
| ID   ID   ID             |              | Usage     |
+-----+-----+
| No running processes found |
+-----+
```

```
[rocky@instance-20221011-1307 ~]$ apptainer exec --nv stable-diffusion.sif nvidia-smi
Sun Oct 16 19:11:42 2022

+-----+
| NVIDIA-SMI 520.61.05   Driver Version: 520.61.05   CUDA Version: 11.8   |
+-----+
| 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   Tesla V100-SXM2...  Off      | 00000000:00:04:0  Off |   0%        Default |
| N/A   42C    P0     37W / 300W    | 0MiB / 16384MiB |           | N/A   |
+-----+-----+

Processes:
+-----+
| GPU   GI   CI        PID   Type   Process name                      GPU Memory |
| ID   ID   ID             |              | Usage     |
+-----+-----+
| No running processes found |
+-----+
```


txt2img pipeline

- Prompt in, output images out
- 512x512 by default as Stable Diffusion trained on this size of images - scaling requires much more VRAM than available in many places!
- Increasingly fidelity increases compute time but doesn't increase memory usage
- Results *incredible!!!*
 - Near-professional quality art in a couple minutes from a machine for a couple bucks worth of compute?
 - Landscapes, people, cities - *all generated!*
- **Command lines:** (after `apptainer shell --nv --bind /data:/data stable-diffusion.sif`)
 - `python3 scripts/txt2img.py --plms --prompt "A steampunk airship, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, oil on canvas, colorful, cinematic" --scale 7.5 --ddim_steps 250 --n_iter 2 --seed $RANDOM --outdir /data/txt2img`
 - `python3 scripts/txt2img.py --plms --prompt "A cityscape, cyberpunk style, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --scale 7.5 --ddim_steps 250 --n_iter 2 --seed $RANDOM --outdir /data/txt2img`
 - `python3 scripts/txt2img.py --plms --prompt "Massive city in a giant cavern with a castle at one side of it, many lights from houses, steampunk style, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --scale 7.5 --ddim_steps 250 --n_iter 2 --seed $RANDOM --outdir /data/txt2img`
 - `python3 scripts/txt2img.py --plms --prompt "A massive utopian city with many geodesic domes, green spaces, parks, glass buildings, and monorails, futuristic, solarpunk, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --scale 7.5 --ddim_steps 250 --n_iter 2 --seed $RANDOM --outdir /data/txt2img`









img2img pipeline

- Image and modification prompt in, modified original image out
- Have to downscale input images to somewhere around 512x512 - again, increasing pixel count = increasing problem size = “exponentially” increasing VRAM usage
- As before increasing fidelity increases compute time but doesn't increase memory usage
- Command lines:
 - `python3 scripts/img2img.py --scale 7.5 --n_iter 2 --seed 1927 --outdir /data/img2img --init-img /data/txt2img/samples/00155.png --prompt "A massive utopian city with many geodesic domes, green spaces, parks, glass buildings, monorails, and airships, futuristic, solarpunk, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --ddim_steps 250`
 - `python3 scripts/img2img.py --scale 7.5 --n_iter 2 --seed 1927 --outdir /data/img2img --init-img /data/txt2img/samples/00155.png --prompt "A massive utopian city with many geodesic domes, green spaces, parks, glass buildings, monorails, and airships, futuristic, solarpunk, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --ddim_steps 250 --strength 0.3`
 - `python3 scripts/img2img.py --scale 7.5 --n_iter 2 --seed 1927 --outdir /data/img2img --init-img /data/txt2img/samples/00155.png --prompt "airships" --strength 0.3`









txt2imgHD pipeline

- 512x512 quite limiting for a lot of purposes
- Need intelligent, AI-powered upscaling
 - *"But Photoshop, GIMP - upscaling through those?"*
 - Nearest-neighbor interpolation and similar algorithmic solutions not capable of scaling ex. 512x512 to HD or beyond - too little original detail
 - AI has some intelligence about what *could* need to be part of an image, so can upscale more effectively
- Another open source project, based on Stable Diffusion
 - Utilizes another model called Real-ESRGAN
- Generate HD from the start or make smaller images first?
 - *Generally faster iteration is possible if you generate txt2img first and then upscale!*
 - Upscaling takes a lot of time and having a selection of worthwhile base images you've identified for upscaling is useful
- Command lines:
 - `python3 scripts/txt2imghd.py --scale 7.5 --steps 200 --detail_steps 200 --n_iter 2 --seed 1927 --outdir /data/txt2imghd --img /data/txt2img/samples/00155.png --prompt "A massive utopian city with many geodesic domes, green spaces, parks, glass buildings, and monorails, futuristic, solarpunk, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --realesrgan /stable-diffusion/realesrgan-ncnn-vulkan --passes 3 --strength 0.4`
 - `python3 scripts/txt2imghd.py --scale 7.5 --steps 200 --detail_steps 200 --n_iter 2 --seed $RANDOM --outdir /data/txt2imghd --prompt "A group of of researchers making a breakthrough discover in a brightly lit lab, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --realesrgan /stable-diffusion/realesrgan-ncnn-vulkan --passes 3 --strength 0.3`
 - `python3 scripts/txt2imghd.py --scale 10 --steps 200 --detail_steps 200 --n_iter 2 --seed $RANDOM --outdir /data/txt2imghd --prompt "A group of of researchers making a breakthrough discover in a brightly lit lab, sharp focus, wide shot, trending on artstation, masterpiece, by greg rutkowski, octane, soft render, digital art, colorful, cinematic" --realesrgan /stable-diffusion/realesrgan-ncnn-vulkan --passes 3 --strength 0.3`







Retraining/Fine-tuning

- Stable Diffusion originally trained on LAION dataset with 32 x 8 A100 nodes at a cost of over \$600,000
- Later versions trained on LAION image subsets considered to be particularly aesthetically pleasing
- *Can we fine-tune this training to increase fidelity for specific objects not well covered in LAION?*
- Sort of!
- No time to demo here, but guides are out there

Outpainting/Inpainting

- Outpainting is taking a base image and running it through Stable Diffusion such that the content of the image is increased logically relative to the base image
- Inpainting is taking a base image and running it through Stable Diffusion such that the base content of the image is modified in some way
 - Similar to img2img but very targeted
- *Meaning???*
 - Take a famous painting... and see what else is in the scene!
 - Fix up Stable Diffusion results that come out incongruent with reality using other tools
 - Generally expanding images to create new results
- Again - no time to demo here, but guides are out there

Tips and Tricks

- Seed should be changed to switch up results - or be kept the same for comparing effects of different prompts
 - Can be used to replicate other's results as well!
- As said - VRAM size required goes up exponentially with input image size or requested output image size
- <https://lexica.art/>
- “greg rutkowski”, “masterpiece”, “trending on artstation” and other queries
 - *Ethics?*
- LAION browsers (*watch out! NSFW!*)
- Change sampler
- Txt2imghd strength tends to “deep dream” images when much higher than 0.45
- Scale is roughly equivalent to how “literally accurate” the image should be to the prompt
 - ~15 produces true to life images for realism...
 - ...but will force AI to use whatever meager results there are literally for that prompt in the case of something more imaginative
 - ~7.5 recommended for more whimsical requests
- Not enough VRAM?
 - Untested by me but might look into lowering precision as this can help at potential cost of result fidelity
 - *Maybe check for former miners selling cheap GPUs to get something with the VRAM needed?*

Wait - you promised... VMs, bare metal, running on a GPU?

- VMs or bare metal no problem for Apptainer - performant in either!
 - No easy exact apples to apples here, but well-established use case
 - As always, you know your workloads best so feel free to benchmark
- Other operating systems?
 - Just need to have Apptainer runtime installed!
- Running on a GPU - easy!
 - *Every image in this presentation generated via Volta-architecture GPU!*
- Running on multiple GPUs?
 - *Also possible!*
 - *Bit difficult to demo, but...*

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



Rocky Linux 8.6

Rocky Linux

[Change image](#)

Shape



BM.GPU4.8

Bare metal machine, 64 core OCPU, 2048 GB memory, 50 Gbps network bandwidth

[Change shape](#)

Are there any requests?

Icing on the cake - DALL·E mini

- The original - DALL·E mini
 - First AI image generation model generally available
 - Kicked off this whole firestorm as was available while DALL·E 2 was still in closed beta
 -
- Not as advanced, but still containerized and fun to check out here
- <https://ciq.co/using-apptainer-to-run-dalle-mini/>

```
Bootstrap: docker
From: rockylinux:8.5

%files
    ./dalle-mini.py /scripts/dalle-mini.py

%post

    dnf -y update && dnf -y upgrade
    dnf -y groupinstall "Development Tools"
    dnf -y install cmake make autoconf python39 python39-devel
    epel-release

    dnf -y config-manager --add-repo
    http://developer.download.nvidia.com/compute/cuda/repos/rhel8/x86_64/cuda-rhel8.repo \
    && dnf -y clean all \
    && dnf -y module install nvidia-driver:latest-dkms \
    && dnf -y install cuda libcuDNN8 libcuDNN8-devel

    python3 -m pip install --upgrade pip
    python3 -m pip install "jax[cuda]" -f
    https://storage.googleapis.com/jax-releases/jax_cuda_releases.html
    python3 -m pip install -q
    git+https://github.com/borisdayma/dalle-mini.git
    git+https://github.com/patil-suraj/vqgan-jax.git
    python3 -m pip install --upgrade notebook ipywidgets

    chmod u+w /root
```

Final thought... future of human art?

- AI art very good... but lacks very fine human fidelity
 - Incredible opportunity for the rest of us to explore creativity
 - But look at enough of it closely... and cracks in it start to show, especially around certain types of prompts
- Ultimately models will get better
 - *All these arbitrary brains running around?*

| CIQ SLA | Standard | | Advanced | |
|-------------------------------------|--|------------------|---|-------------------|
| Hours of Coverage | Standard Business Day 6am to 6pm PT | | 24x7 | |
| Support Channels | Self Help, Phone, Tickets, Email & Chat | | Self Help, Phone, Tickets, Email & Chat | |
| Number of Cases | Unlimited | | Unlimited | |
| Number of Simultaneous Active Cases | 1 per person, cumulative for entire organization | | 2 per person, cumulative for entire organization | |
| Escalation Efforts | User, System Administration, and Security Escalation | | Dedicated Point of Contact for User, System Administration, Security, Engineering, and Development Escalation | |
| When By the Person | Unlimited Nodes, Containers, Virtual Machines. | | Unlimited Nodes, Containers, Virtual Machines. | |
| When By the Node | 50 Nodes = 1 Person | | 50 Nodes = 1 Person | |
| Response Times | Initial Response | Ongoing Response | Initial Response | Ongoing Response |
| Severity 1 | 1 Business Hour | 1 Business Hour | 15 Minutes | 1 Hour |
| Severity 2 | 6 Business Hours | 6 Business Hours | 2 Hours | 2 Hours or Agreed |
| Severity 3 | 6 Business Hours | 1 Business Day | 2 Hours | 1 Day or Agreed |
| Severity 4 | 6 Business Hours | 1 Business Day | 2 Hours | 1 Day or Agreed |

The image features a dark green background with abstract geometric shapes in lighter shades of green. On the left, there are curved lines and a vertical bar. On the right, there are concentric circles and a large, stylized 'Q' shape. The text 'CIQ' is prominently displayed in white on the left side.

CIQ