QUMULO PRODUCT DEMO

Genomics Sequencing Pipeline in AWS

QUMULO OVERVIEW

Qumulo powers demanding healthcare and life sciences workloads, from medical imaging captured by CT and MRI equipment, to cytogenetic microscopy, to groundbreaking genomics research.

In fact, some of the world's fastest genomic sequencing pipelines are backed by Qumulo all-NVMe storage. Researchers can easily capture raw data from instruments on-premises and burst to the cloud for analysis, all while keeping the workload and data safely and securely stored in Qumulo.

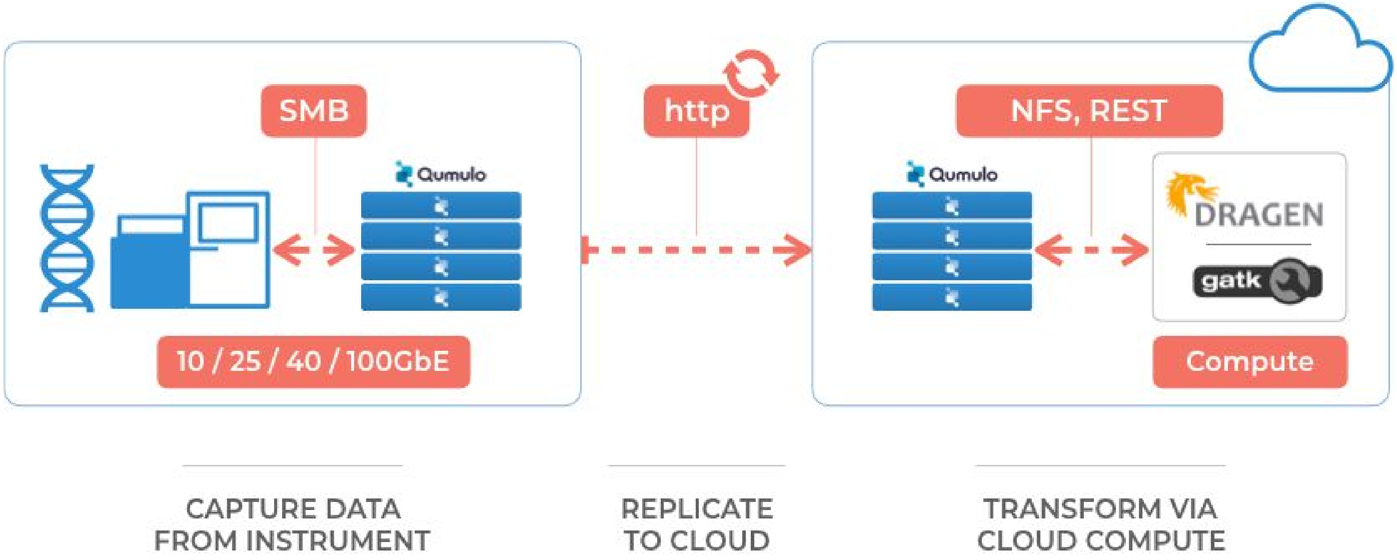
Qumulo's distributed file system meets the performance and capacity demands of HPC environments, in the data center and in the cloud yet Qumulo provides high-performance file services via standard protocols over Ethernet.

Qumulo's modular architecture of storage nodes runs on Qumulo hardware, or options from HPE and Dell. Managing growth is simple. Simply add nodes to the existing infrastructure with no disruption or downtime.

Only Qumulo offers true multi-protocol collaboration for high-volume workflows in a single, extremely large namespace. This means users across your institution can collaborate natively via NFS, SMB, FTP, or integrate with REST, all while leveraging full POSIX and SMB ACL permissions models without worry of permissions conflicts or broken inheritance policies.

Because Qumulo is resilient, self-healing, and simple to manage, it is the perfect complement to the technology stack of research organizations of all sizes—from national laboratories to smaller institutions without full-time IT staff.

DEMO OVERVIEW



Learn more at www.qumulo.com

DEMO HIGHLIGHTS

Our SC'19 demo shows off Qumulo performance, flexibility, and ease-of-management. Here are the high-level takeaways:

PERFORMANCE	● Extreme low latency ● High throughput and IOPS under load ● Hundreds of thousands of small file READS and WRITES with each pipeline run ● Two simultaneous pipelines running with minimal system impact ● Network protocols (SMB, NFS, FTP, REST) are fully distributed and offer consistent, high performance
FLEXIBILITY	● Burst to Qumulo in the cloud for compute workloads—same file architecture, familiar interface and all services ● Keep your workload as native file (no conversion to object storage) while working in cloud ● Connect to the cluster via standard protocols—SMB, NFS, FTP, REST ● Keep data in sync with Qumulo in the Cloud using Replication
EASE OF MANAGEMENT	● Dashboard shows activity in real-time allowing admins to pinpoint usage and performance events. ● For long-running scripts (such as those in genomics which might run 24 hours or more), it's important to be able to spot a rogue compute activity and address issues ● Scaling Qumulo is as simple as adding additional nodes—all services run on all nodes—no sizing headaches in balancing new service nodes with new metadata nodes. Just add a Qumulo node and you're done

CLOUD DEMO TALK TRACK

The demo walks through the early steps of a typical life science research workflow — capturing data from an instrument and getting it into a format that researchers can analyze and study. The on-prem data capture and replicate to cloud steps (1 and 2, below) are not shown but you can talk to them using the details below.

1. DATA CAPTURE

WHAT'S HAPPENING	WHAT TO HIGHLIGHT
We are speaking to this step—not showing it in the demo. 1. Imagine researcher is generating data on an instrument in the lab – for example a genomics sequencer like **Illumina NovaSeq** or **Illumina X10** running a Windows-based administration console which controls the instrument 2. Instrument is connected via SMB to Qumulo (on-prem), writing 100,000s of small files (kb size) 3. The sequencing process can take many hours	● **SMB is important** — many scientific instruments run Windows-only capture utilities and admin tool. Qumulo has enterprise- grade, high-performance, distributed SMB ● **Cross-Protocol Perms (XPP)** — cross-protocol workflows need robust permissions management. Qumulo offers best-in-class, fully automatic cross-protocol permissions handling ● **Small file writes** — SMB read and write leasing functionality boosts performance for small byte reads and writes

2. DATA REPLICATION

WHAT'S HAPPENING	WHAT TO HIGHLIGHT
We are speaking to this step—not showing it in the demo. 4. Assume a replication relationship was set up between the on-prem cluster capturing data, and the Qumulo instance running in AWS 5. Continuous replication ensures data is copied to Qumulo in AWS moments after it is captured from the instrument on-premises. Once the capture job is complete, the files are already in Qumulo on AWS ready for the next step	● **Put your data where you need it** — replicate any part of your data set (by directory) to another Qumulo whether remote site or in the cloud. Use for cloud bursting, DR, or backup ● **Extensible via REST API** — scripts could be written to have Qumulo move data from the replication target directory to an 'input' directory based on a notification from the instrument, feeding it to the next step in the pipeline

3. DATA TRANSFORMATION — BCL to Fastq

WHAT'S HAPPENING	WHAT TO HIGHLIGHT
6. Once the instrument data is on Qumulo in AWS, file conversion takes place (this is where our demo begins) 7. BCL files (raw instrument data format) are converted to Fastq (interchange format) so that they can be analyzed in the next step	• **System Performance** — during file conversion, tens of thousands of BCL files will be converted into ~30 larger Fastq files. This process will spawn a large volume of small file read activity

4. DATA TRANSFORMATION — Analysis Pipeline

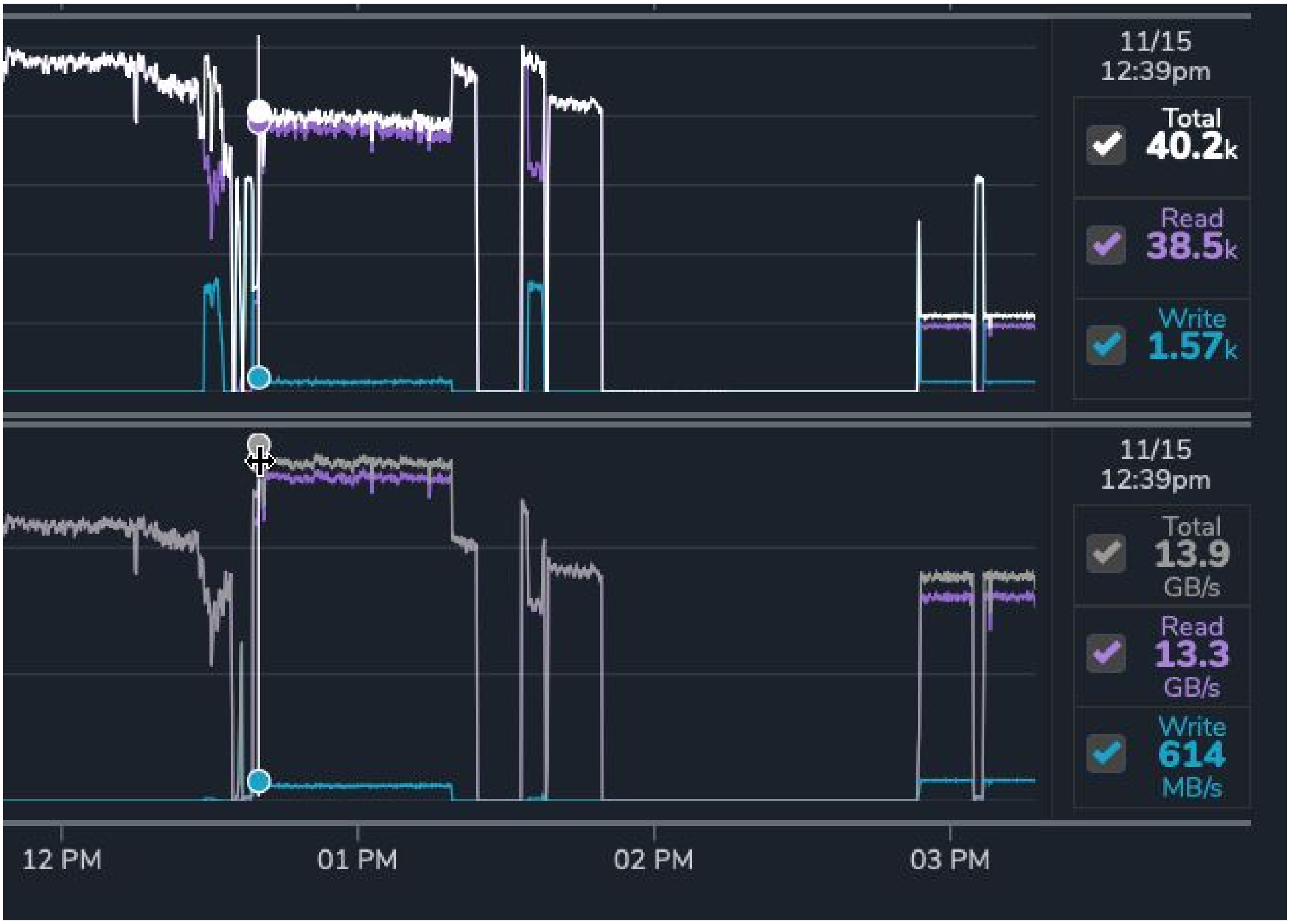
WHAT'S HAPPENING	WHAT TO HIGHLIGHT
8. Once the data is converted to Fastq, the genomics pipeline kicks off. We are showing two parallel paths: GATK pipeline (open source), and Illumina DRAGEN pipeline (commercial). 9. GATK and DRAGEN both transform the data by aligning and deduplicating data, and performing some initial analysis 10. **DONE!** The result is a VCF file, which is the input researchers use to study the data in applications such as Genome Browser	• **What's a Pipeline?** — pipelines are a series of apps and scripts linked as a toolkit to transform data. Conversion from one format to another is commonly required in genomics workflows. There are two major pipelines in this area: GATK (open source) and Illumina DRAGEN (commercial solution) • **Ready for Anything** — GATK was built by researchers, not by an engineering team. It can hit compute and storage in odd, non-optimized ways. By contrast, DRAGEN is optimized for HPC and is more than 50x faster than GATK. Either way, Qumulo works great! • **Cross-Protocol Perms (XPP)** — Cross-protocol permissions support is important here again. Once the pipeline is complete, researchers will work with the resulting VCF over SMB and NFS. Admins will want to ensure permissions are set and enforced correctly

ON-PREM ALL-NVME FLASH DEMO TALK TRACK

We have a separate demo to show pure performance. This demo is not related to the Genomics in the Cloud story, but this shouldn't be a huge issue because SC'19 is about computing performance, not specifically about life sciences IT.

- The demo is run on 4 clients hitting a 4-node Qumulo P-23 All-NVMe cluster connected via 100GbE
- The demo simulates a video transcoding workflow (converting MP4 to 720p, 1080i, 4k, and so on)
- The demo simulates transcoding 8 separate movies to different formats
- During transcode, all movie frames are read (millions of small files)
- Files are piped into an encoding queue, which performs transcode
- Output is a single large movie file for each flavor (1080i, 4k, etc.)

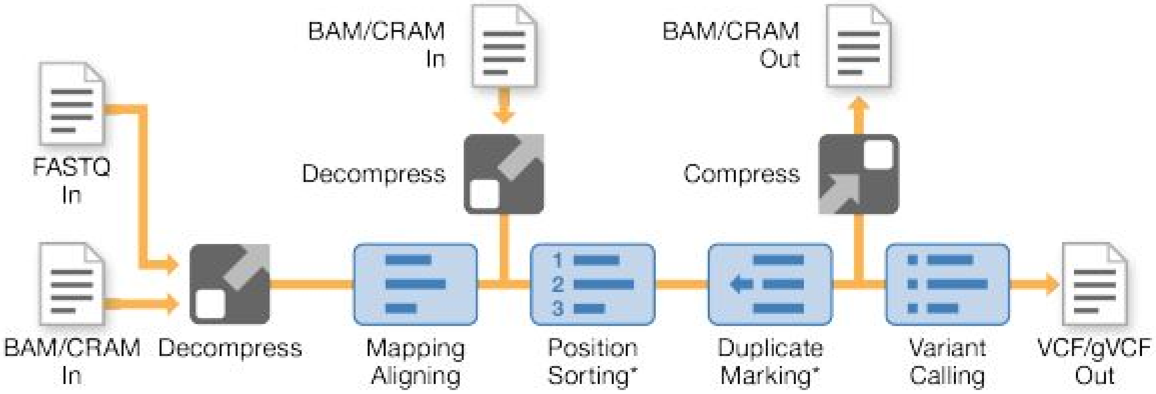
Performance is awesome, delivering greater than 13GB/s reads!



MORE INFORMATION ON GENOMIC COMPUTING

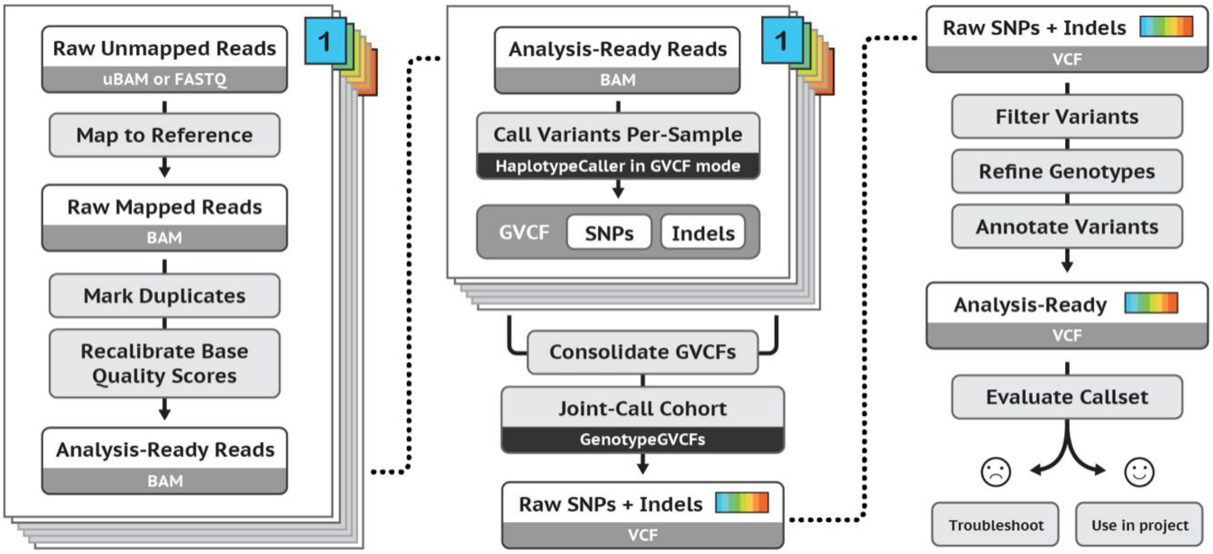
DRAGEN

Read more about the DRAGEN genomics sequencing pipeline in this whitepaper.



GATK

Read more about GATK pipelines in this document.



Q&A

Use these talking points as a guide to steer the conversation back to Qumulo benefits and unique capabilities for healthcare / life sciences workloads.

Q: Why would I buy NAS like Qumulo instead of Lustre or GPFS?

A: Lustre and GPFS are specialized solutions which can deliver high performance, but getting the best out of them can be a full-time job. Even then, you might not get the results you expect if your workload is not optimized for Lustre or GPFS. This is common with open source pipelines such as GATK which are not highly parallelized or built for HPC.

Qumulo delivers high-performance, and unlike Lustre and GPFS, it does so over standard Ethernet and standard protocols. Qumulo does not require proprietary protocols or client-side drivers and applications.

Q: Why would I buy NAS like Qumulo instead of Lustre or GPFS? (Another angle)

A: HPC (high-performance computing) is more accessible than ever, as even commodity server hardware can now begin to deliver HPC-level computing is becoming more mainstream, many customers don’t need the ultra high performance of parallel file systems. The reliable, fast performance of Qumulo All-NVMe or Hybrid platforms can meet their requirements, without all the complexities.

Qumulo is simple to set up and easy to add performance and capacity to over time. Parallel file systems require two networks to offer NAS, have 5+ interfaces to manage, and performance/capacity upgrades are done separately to different parts of the system.

Q: Can Qumulo scale?

A: Yes, Qumulo scales easily to billions of files—small and large—in a single namespace. Because Qumulo has a highly efficient architecture, even billions of small files (byte or kilobyte size) can be stored with excellent economics.

Increasing the size of a Qumulo cluster on premises is as simple as adding one or more additional nodes. Unlike other solutions such as Lustre, all services are distributed across all nodes—there is no sizing or balancing between metadata service and target nodes (MDS, MDT), object storage target nodes (OST) and object storage server nodes (OSS). Each Qumulo node runs all services in a distributed fashion across the cluster.

Q: What’s the advantage of running Qumulo in the cloud?

A: Qumulo lets you take your data where you need it, whether on-premises, in the cloud, or both. You can easily replicate data between Qumulo clusters and ensure data is protected and retains full fidelity (permissions, metadata, and so on) in the Qumulo ecosystem. Best of all, your file data can remain as file data wherever it resides in Qumulo — even in the cloud. This lets you easily burst to the cloud to take advantage of cloud compute as we’ve shown in the demo.

Q: Is Qumulo fast?

A: Yes. Very fast. For example, the all-NVMe flash-based system can deliver 14GB/s write throughput, and 19GB/s read throughput. That’s fast.

Q: Why does multiprotocol matter? Our institution is strictly Linux and NFS.

A: Qumulo supports true multiprotocol collaboration. All nodes speak all protocols so the entire cluster offers full protocol support in a distributed way. Even if your institution uses only NFS, some of your scientific instruments may require SMB. Because Qumulo offers native multiprotocol support (note our SMB is *enterprise distributed SMB*, not open source Samba), you can connect your scientific instruments directly to Qumulo over SMB, NFS, REST, or FTP. This means no intermediary server is required between your instruments and your main storage, further reducing cost and complexity.

Q: What you’re showing isn’t the whole genomic analysis pipeline!

A: There are many steps to a typical genomics analysis pipeline—this demo doesn’t try to represent the entire flow. We focused on the most system-intensive pieces of the pipeline. Illumina DRAGEN is one of the fastest pipelines in the world—50x faster than GATK on average and we’re showing how Qumulo running in AWS easily accommodates this level of load.

DEMO DAILY RUN INSTRUCTIONS

Follow these steps to start the demo each morning.

1. LOG INTO THE DEMO STATION

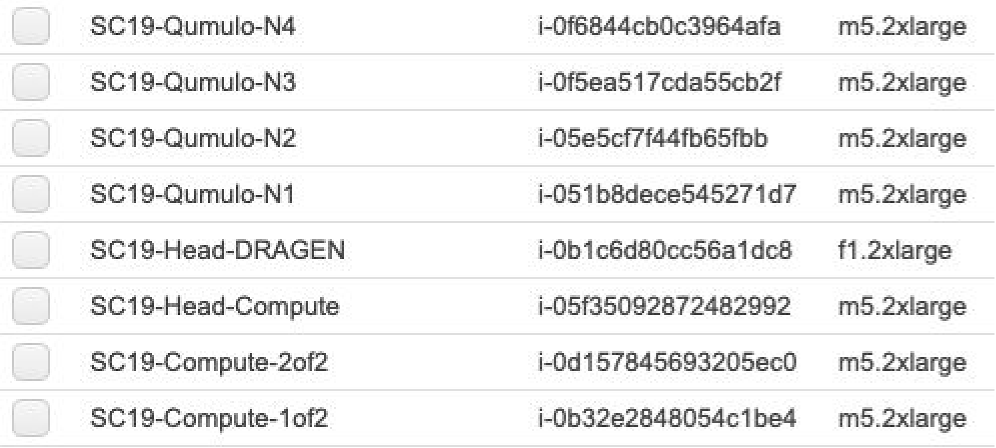
User: SC’19 Demo

Password: kumquats

2. START THE AWS INSTANCE

NOTE: This step will be done for you by the AWS admin for the show since not everyone has AWS credentials.

1. Open a browser and go to https://qumulo.onelogin.com
2. Open AWS Console — **qumulo-techmarketing (957420008347)** account
3. Start the following instances:



□	SC19-Qumulo-N4	i-0f6844cb0c3964afa	m5.2xlarge
□	SC19-Qumulo-N3	i-0f5ea517cda55cb2f	m5.2xlarge
□	SC19-Qumulo-N2	i-05e5cf7f44fb65fbb	m5.2xlarge
□	SC19-Qumulo-N1	i-051b8dece545271d7	m5.2xlarge
□	SC19-Head-DRAGEN	i-0b1c6d80cc56a1dc8	f1.2xlarge
□	SC19-Head-Compute	i-05f35092872482992	m5.2xlarge
□	SC19-Compute-2of2	i-0d157845693205ec0	m5.2xlarge
□	SC19-Compute-1of2	i-0b32e2848054c1be4	m5.2xlarge

4. Copy IP Address
   - Copy the public IPv4 address for **SC19-Head-Compute node**
   - You will paste it in the next step

3. LAUNCH SEQUENCING PIPELINES

Once you start them, pipeline jobs will automatically queue and loop throughout the day. However, they will not restart once the cloud instance is stopped/resumed, so you’ll need to repeat this procedure each morning.

1. Open Terminal
2. Change path to the ~/.ssh directory

```
cd ~/.ssh
```

3. Connect to the Head Compute node using the IP address you copied earlier, replacing the highlighted area below. *(For SC'19, we are using an Elastic IP, so you will be able to connect to this same IP address each day of the show)*

   ```
   ssh -i "sniderman.pem" centos@xxx.us-west-2.compute.amazonaws.com
   ```

4. You are now logged into the cloud compute node. Change user to "scdemo". The password is **cluster**

   ```
   su - scdemo
   ```

5. Kick off the pipeline jobs

   ```
   ./b2f.sh

   ./gatk.sh

   ./dragen.sh
   ```

6. You will see a confirmation after submitting each job

4. OPEN THE QUMULO DASHBOARD (CLOUD)

1. Open Safari (Chrome won't allow you to connect to connect to sites without proper certificates)
2. Safari should open https://34.209.29.145/ by default
3. A warning may appear. If so, click "Show Details", then click "Visit this Website"
4. Log into Qumulo Dashboard
   - User: admin; Password: admin

5. OPEN THE QUMULO DASHBOARD (ON-PREM)

1. Open a new tab or window and connect to https://af2-100g.eng.qumulo.com
2. You must be on a VPN connection to connect. You can also connect to ...af6, af10, or af12
   - User: admin; Password: admin

6. START THE JOB TAIL

Tailing log files will show progress as the demo runs. You can tail logs for each of the three scripts (BCL, GATK, and DRAGEN). **NOTE** if tailing stops, this means the job has finished. New jobs are continuously kicking off, so you'll need to find the names of the new log files to tail. You will need to do this throughout the day.

The names of the log files change with each run, so you'll need to check the queue first to get the file names:

```
qstat -f
```

Below shows example `qstat` output. You'll notice there is a job number associated with each running job type (BCL, DRAGEN, and GATK). We will tail each of these jobs in three separate Terminal windows which will highlight activity most clearly.

```
queuename         qtype resv/used/tot. load_avg arch        states
------------------------------------------------------------------
bcl@head          BIP   0/2/2          9.57     lx-amd64
903 0.55500 bcl   scdemo       r     11/18/2019 18:39:02     1
904 0.55500 bcl2  scdemo       r     11/18/2019 18:39:02     1
------------------------------------------------------------------
dragen@ip         BIP   0/0/1          0.01     lx-amd64      u
------------------------------------------------------------------
gatk@node01       BIP   0/1/1          0.80     lx-amd64
902 0.55500 GATK  scdemo       r     11/18/2019 18:07:47     1
------------------------------------------------------------------
gatk@node02       BIP   0/1/1          0.83     lx-amd64
900 0.55500 GATK  scdemo       r     11/18/2019 18:07:17     1
```

- Tail both BCL jobs in one window. **Note** you can append the names of additional log files to tail both in the same window. If more than one BCL job appears, tail both as follows:

```
tail -f /home/scdemo/bcl.o903 bcl2.o904
```

- Tail both GATK jobs in a new Terminal window

```
tail -f /home/scdemo/GATK.o902 GATK.o900
```

- Tail the DRAGEN job in a third Terminal window

DEMO TROUBLESHOOTING

RESETTING JOB QUEUE

If the pipelines are not appearing to process data or not functioning properly, you can reset them.

1. Verify the queue is working properly

```
qstat -f
```

2. When the queue is empty and ready to accept jobs, it will appear like this -- no slots used in the RESV/USED/TOT column and no errors listed in the STATES column:

```
queuename        qtype resv/used/tot. load_avg arch         states
-----------------------------------------------------------------
bcl@head         BIP   0/0/2          0.84     lx-amd64
-----------------------------------------------------------------
dragen@dragen    BIP   0/0/1          0.82     lx-amd64
-----------------------------------------------------------------
gatk@nod         BIP   0/0/1          0.82     lx-amd64
```

3. If you see errors in the states column, such as the errors listed below, **delete jobs** or **reset the queue**.

```
qdel -j #JOB_NUM

qmod -c #QUEUE_NAME
```

a – Load threshold alarm
o – Orphaned
A – Suspend threshold alarm
c – Configuration ambiguous
d – Disabled
s – Suspended
u – Unknown
E – Error

4. Submit new jobs as detailed above in LAUNCH SEQUENCING PIPELINES

RESET DOCKER PIPELINE

If resetting the queue does not help, you can reset the Docker container running the pipelines.

1. Restart Docker

```
systemctl restart docker
```

```
ps ax | grep docker
```

2. Clear BCL jobs remaining in the queue before restarting them:

```
qstat -f | grep bcl
```

```
qdel -f #BCL_JOB_NAME
```

3. Restart BCL jobs:

```
qsub b2f.sh
```

TROUBLESHOOTING ON-PREM PERFORMANCE DEMO

The on-prem demo is self-running and should not require any intervention. However, if one or more of the client nodes reboot, the workflow will need to be restarted.

Login to the root node

```
root@duc43-100g.eng.qumulo.com
```

Verify if the environment is running

```
docker stack ps supercompute
```

Restart the demo with:

```
docker stack deploy -c supercompute.yml supercompute
```

If that does not work, you can abort and restart the environment with:

```
docker stack rm supercompute
docker stack deploy -c supercompute.yml supercompute
```

Nodes:

P-23 Cluster Nodes	Client Nodes
af2-100g.eng.qumulo.com	duc43-100g.eng.qumulo.com
af6-100g.eng.qumulo.com	duc44-100g.eng.qumulo.com
af10-100g.eng.qumulo.com	duc45-100g.eng.qumulo.com
af12-100g.eng.qumulo.com	duc47-100g.eng.qumulo.com

RESOLVING AUTHENTICATION ISSUES

If you are having trouble connecting to the AWS environments, check the following:

- Check AWS console to make sure AWS instances are running
- Check that the public key is installed on your computer. Look for ‘sniderman.pem’ key

```
cd ~/.ssh

ls
```

- If the key is not on the system, copy it to the ~./ssh directory. Create the directory if it does not exist

```
cd ~

mkdir .ssh; chmod 700 ~/.ssh
```

- Make sure the permissions on the key are set correctly.

```
chmod 700 ~/.ssh sniderman.pem
```

DETAIL ON WHAT JOB SUBMISSION SCRIPTS DO

BCL

The “b2f.sh” script will start the Bcl2Fastq conversion process of the Genomics pipeline in the “BCL” SGE queue. Two runs will be started simultaneously, bcl.sh and bcl2.sh. These jobs will run against the same genomic sequence data but from different locations on the filesystem. This is to show more activity on the dashboard. The b2f.sh script will start two new jobs once the previous jobs have completed. This will provide Qumulo with a continuous loop of Bcl2Fastq conversion jobs throughout the entire show.

In the event of a queuing issue or general failure that is not able to be restarted via “./b2f.sh” or restarting the queuing system, “b2f.sh” can be run manually on the head node via the docker command that is listed below.

```
$ docker run --mount type=bind,source=/Data,target=/Data bf -R /Data/bcl_run/ -r 4 -p 4 -w 4
```

GATK

The “gatk.sh” script will start two GATK pipelines in the “gatk” SGE queue. This script kicks off two SGE jobs which in turn starts two GATK pipelines. One on Node01 and one on Node02. This script will also initiate a restart of the GATK pipeline that will continuously loop until terminated.

In the event of a queuing issue or general failure that is not able to be restarted via “/home/scdemo/gatk.sh,” or restarting the queuing system, the GATK pipeline can be initiated on Node01 and Node02 via the following command.

```
$ cd /Data/bt/$(whoami)/$1
```

```
$ /home/scdemo/bioteam/fastq2vcf.sh *
```

DRAGEN

The “dragen.sh” script will submit a single job to the “dragen” queue, running only on the Dragen compute node. This job will restart when finished to run a continuous loop of Dragen jobs. The most efficient way to troubleshoot issues with the Dragen FPGA is to first reboot the instance. Illumina or outside support may be required if resetting the Dragen does not correct the issue.