Diamanti D-Series

v2.3.0 Appliance Capabilities

Diamanti D-Series v2.3.0 Technical Note: Exploring Appliance Capabilities, First Edition, January 2020

Contents

Exploring D-Series Appliance Capabilities

Demo Script and Pod Definition Template

1

Exploring D-Series Appliance Capabilities

This technical note describes how to explore the functional and performance capabilities of Diamanti D-Series appliances using a Diamanti-supplied script to quickly and easily create an representative working environment.

The note shows how to run the script on a Diamanti cluster, and describes how to run and interpret output from the Diamanti command line interface (CLI). The note then describes how to further explore and monitor performance using the Diamanti User Interface (UI).

Note: For information about installing and configuring the Diamanti D-Series appliance, see the *Diamanti Installation Guide*. Refer to the *Diamanti Command Line Interface Guide* for details about the command line interface. For more information about the Diamanti User Interface (UI), see the *Diamanti User Interface Guide*.

Introduction

Diamanti offers tools that can be used to effortlessly deploy a series of pods to create a representative environment that exercises basic I/O (storage and network) on a Diamanti cluster. This allows users to quickly and easily experience Diamanti performing typical, real-world operations.

Note: The tools described in this technical note complement the resources available in the following location: `/usr/share/diamanti/manifests/examples/demo`.

Getting Started

The Diamanti software pools resources across all nodes in a cluster, enabling Kubernetes to efficiently schedule containers within the cluster.

Before running the demo to explore Diamanti D-Series capabilities, a cluster consisting of three nodes needs to be available. For the purposes of this demo, use the following command to create the demo cluster:

```
$ dctl cluster create autotb11 appserv76,appserv77,appserv78
--vip 172.16.20.251 --poddns cluster.local --svlan 500 -p Test1234!
```

Note: Clustering D-Series appliances requires basic preparation before running the `dctl cluster create` command. For detailed information, refer to the *Diamanti Quick Start Guide*.

After creating the cluster, check the status using the following command:

```
$ dctl cluster status
Name            : autotb11
UUID            : b0465c1d-7628-11e9-86a6-a4bf01147a45
State           : Created
Version         : 2.3.0 (108)
Master          : appserv76
Etcd State      : Healthy
Virtual IP      : 172.16.19.61
Storage VLAN    : 500
Pod DNS Domain  : test.eng.datawisesystems.com
NAME                     NODE-STATUS   K8S-STATUS  MILLICORES   MEMORY          STORAGE
IOPS       VNICS     BANDWIDTH   SCTRLS          LABELS

LOCAL, REMOTE
appserv76/(master, etcd)   Good        Good        0/40000     1GiB/128GiB      90.24GB/3.05TB
75K/500K   9/63      1.88G/40G   9/64, 0/64      beta.kubernetes.io/arch=amd64,beta.
kubernetes.io/os=linux,kubernetes.io/hostname=appserv76,node=node0
appserv77/(etcd)          Good        Good        250/40000    1.06GiB/128GiB   90.24GB/3.05TB
75K/500K   10/63     1.88G/40G   9/64, 0/64      beta.kubernetes.io/arch=amd64,beta.
kubernetes.io/os=linux,kubernetes.io/hostname=appserv77,node=node1
appserv78/(etcd)          Good         Good        0/40000     1GiB/128GiB      90.24GB/3.05TB
75K/500K   10/63     1.88G/40G   9/64, 0/64      beta.kubernetes.io/arch=amd64,beta.
kubernetes.io/os=linux,kubernetes.io/hostname=appserv78,node=node2
```

In this instance, the `dctl cluster status` command tells us the following:

- Each node in the cluster has 40 hyper-threaded cores with 128GB of RAM
- Each node has a usable storage capacity of 3.05TB and 40Gbps network bandwidth
- Each node has IOPS set at 500K, which means that up to 450K can be provisioned on each node of the cluster (see note below)

Each D-Series node is capable of driving up to 1 million IOPS. However, only 500K IOPS is assumed as the node capability to provide a worst-case guarantee and meet application SLAs (service level agreements).

Understanding Performance Tiers

Diamanti uses performance tiers to enforce minimum network throughput and storage IOPS for containers, offering deterministic high performance with high workload density. While the configuration is completed globally across the cluster, each node in the cluster enforces the specified performance tier for all workloads running on the node.

Use the following command to display the built-in performance tiers in a Diamanti cluster:

```
$ dctl perf-tier list
NAME         STORAGE IOPS   NETWORK BANDWIDTH  MAX STORAGE IOPS   MAX NETWORK BANDWIDTH   LABELS
best-effort   0              0                    -                  -
diamanti.com/template=true
high          20k            500M                 -                  -
diamanti.com/template=true
medium        5k             125M                 -                  -
diamanti.com/template=true
```

Administrators can create new performance tiers or delete existing tiers, as needed. When creating a new performance tier, administrators can specify the requested storage IOPS, maximum storage IOPS, requested network bandwidth, maximum network bandwidth, and any labels, as appropriate. By default, performance tiers do not have maximum storage IOPS and network bandwidth limits.

Note: Diamanti allows pods to access 90% of the total IOPS. For example, 500K IOPS provides pods with access to 450K IOPS (500K * 0.9 = 450K).

The built-in `high` performance tier offers 20K IOPS. This means that a maximum of 22 pods assigned to the `high` performance tier can be deployed on each node (500K/20K = 22.5). A similar calculation reveals the maximum number of pods that can be assigned to the `medium` performance tier, or any other user-defined tier.

Note that the built-in `best-effort` performance tier has special characteristics. A Diamanti cluster allows pods assigned the `best-effort` tier to run on any node, while providing a guarantee that these pods will *not* interfere with pods assigned to other performance tiers. However, when pods assigned to other performance tiers do *not* use their IOPS/bandwidth, `best-effort` pods are provided the excess system resources. This makes `best-effort` pods particularly suited for specific types of tasks, such as running batch jobs.

Understanding Virtual Network Controllers (VNICs)

Diamanti D-Series appliances provide 64 VNICs per node. Diamanti reserves one instance for secure communication between the SR-IOV card and the system. This means that 63 VNICs are available per node for deploying pods.

Exploring Virtual Storage Controllers (Local and Remote I/O)

Diamanti D-Series appliances provide 64 local and 64 remote NVMe storage controllers per node. This means that a node can simultaneously serve 64 local and 64 remote volumes.

Note: Volumes are bound to NVMe storage controllers at pod deployment time, and the association is removed at when the pod is deleted. However, volumes and the contents of the volumes remain.

For example, a pod deployed using a local NVMe storage controller to access a volume on the `appserv76` node uses one local NVMe storage controller resource. Users can then deploy the same pod on the `appserv77` node using the volume on `appserv76`. In this case, the pod uses one local NVMe storage controller on `appserv77` and one remote NVMe storage controller on the `appserv76` node.

Deploying the Demo Pods

Diamanti offers a script to deploy a series of pods to exercise basic I/O (storage and network) on a Diamanti cluster. After these pods are running on the cluster, users can access the Diamanti command line interface (CLI) and Diamanti User Interface (UI) to monitor performance and explore D-Series appliance capabilities.

The script, `create-iperf-fio.sh`, creates three pods per performance tier for each node, and runs FIO and iperf traffic across these pods. For FIO, the script varies the IO pattern across the three nodes (100% reads, 70/30 read/write mix, and 100% writes). Among the nine pods per node, three are `high` performance tier pods; three are `medium` performance, and three are `best-effort` pods.

Diamanti also supplies a pod definition template file, `iperf-fio.json`, which the `create-iperf-fio.sh` script customizes to deploy the pods.

Note: The `create-iperf-fio.sh` script and the `iperf-fio.json` pod definition template are listed in Appendix A.

Deploying the Pods (3 High, 3 Medium, 3 Best-Effort)

Use the following command to deploy the demo pods:

```
$ ./create-iperf-fio.sh appserv76 appserv77 appserv78 blue
```

The script takes the names of the nodes as arguments (`appserv76`, `appserv77`, and `appserv78`) and the name of the network (`blue`, in this instance) to use to allocate IP addresses. The script then deploys the appropriate number of pods for each performance tier per node, selecting the I/O load type to generate on the various nodes.

After running the script, use the following command to display the cluster status:

```
$ dctl cluster status
Name            : autotb11
UUID            : b0465c1d-7628-11e9-86a6-a4bf01147a45
State           : Created
Version         : 2.3.0 (108)
Master          : appserv76
Etcd State      : Healthy
Virtual IP      : 172.16.19.61
Storage VLAN    : 500
Pod DNS Domain  : test.eng.datawisesystems.com
NAME                     NODE-STATUS  K8S-STATUS   MILLICORES   MEMORY          STORAGE
IOPS       VNICS     BANDWIDTH   SCTRLS          LABELS

LOCAL, REMOTE
appserv76/(master, etcd)  Good         Good        0/40000     1GiB/128GiB     90.24GB/3.05TB
75K/500K   9/63      1.88G/40G   9/64, 0/64      beta.kubernetes.io/arch=amd64,beta.
kubernetes.io/os=linux,kubernetes.io/hostname=appserv76,node=node0
appserv77/(etcd)         Good         Good        250/40000    1.06GiB/128GiB   90.24GB/3.05TB
75K/500K   10/63     1.88G/40G   9/64, 0/64      beta.kubernetes.io/arch=amd64,beta.
kubernetes.io/os=linux,kubernetes.io/hostname=appserv77,node=node1
appserv78/(etcd)         Good         Good        0/40000     1GiB/128GiB     90.24GB/3.05TB
75K/500K   10/63     1.88G/40G   9/64, 0/64      beta.kubernetes.io/arch=amd64,beta.
kubernetes.io/os=linux,kubernetes.io/hostname=appserv78,node=node2
```

The `dctl cluster status` command output shows that 75K IOPS have been provisioned per node in the cluster. This is calculated as follows:

> Three `high` at 20K = 60K
>
> Three `medium` at 5K = 15K
>
> Three `best-effort` at 0K = 0K (note that `best-effort` pods do *not* count as provisioned resources)
>
> TOTAL = 60K + 15K + 0K = 75K of provisioned IOPS

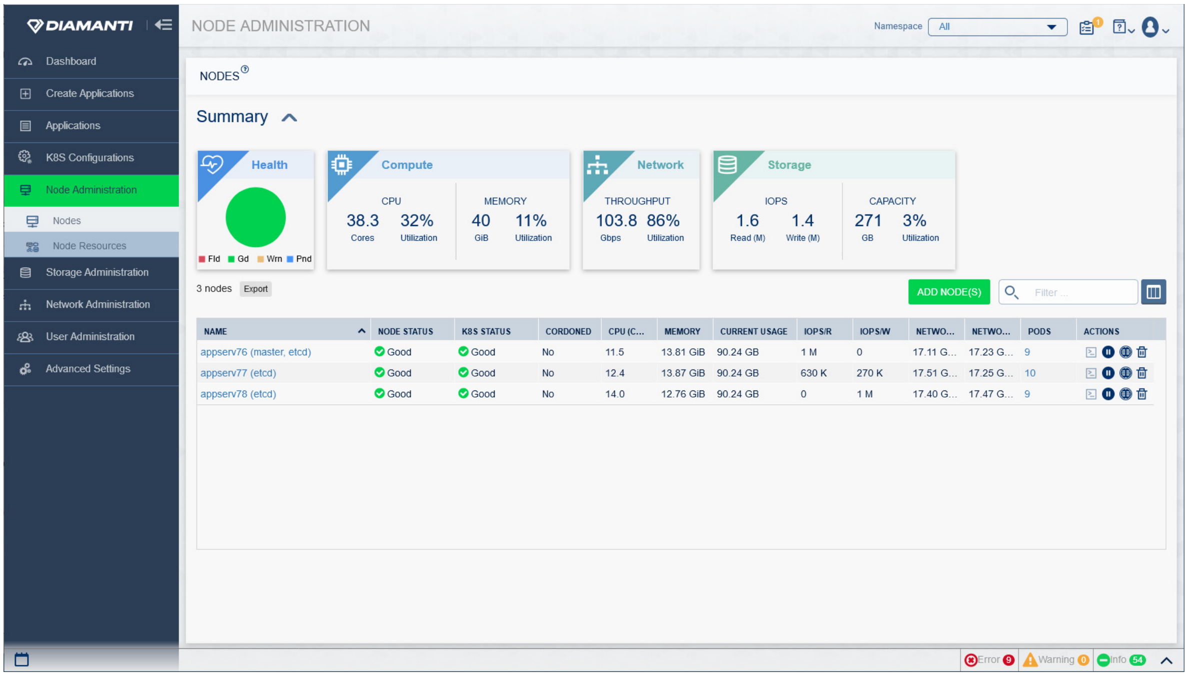
Similar calculations show that the provisioned network bandwidth is 1.875G.

Monitoring using the Diamanti UI

The Diamanti UI is available to further explore and monitor D-Series appliance capabilities.

Note: For complete details about using the Diamanti UI, see the *Diamanti User Interface Guide*.

Launch the Diamanti UI and navigate to the Nodes page.



Health: 🟥 Fld 🟩 Gd 🟧 Wrn 🟦 Pnd

Compute: CPU 38.3 Cores, 32% Utilization; MEMORY 40 GiB, 11% Utilization

Network: THROUGHPUT 103.8 Gbps, 86% Utilization

Storage: IOPS 1.6 Read (M), 1.4 Write (M); CAPACITY 271 GB, 3% Utilization

3 nodes

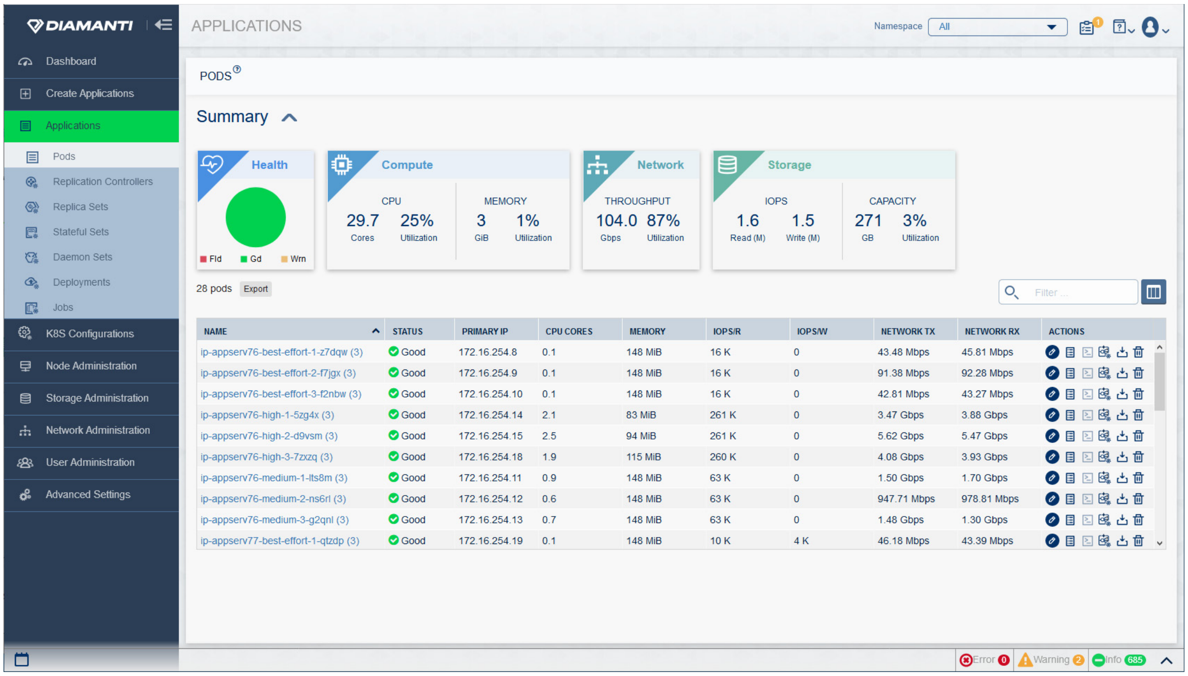
NAME	NODE STATUS	K8S STATUS	CORDONED	CPU (C...	MEMORY	CURRENT USAGE	IOPS/R	IOPS/W	NETWO...	NETWO...	PODS	ACTIONS
appserv76 (master, etcd)	✓ Good	✓ Good	No	11.5	13.81 GiB	90.24 GB	1 M	0	17.11 G...	17.23 G...	9	
appserv77 (etcd)	✓ Good	✓ Good	No	12.4	13.87 GiB	90.24 GB	630 K	270 K	17.51 G...	17.25 G...	10	
appserv78 (etcd)	✓ Good	✓ Good	No	14.0	12.76 GiB	90.24 GB	0	1 M	17.40 G...	17.47 G...	9	

Figure 1. Nodes Screen

Note the following:

- appserv76 generates approximately 1M IOPS/R across the three pods (100% read pattern)
- appserv77 generates approximately 630K IOPS/R and 270K IOPS/W (70:30 read/write pattern)
- appserv78 generates approximately 1M IOPS/W (100% write pattern)

Navigate to the Applications > Pods page and scroll through the list to display the distribution of the three performance tiers of `high`, `medium`, and `best-effort` pods across the nodes.



PODS

Summary

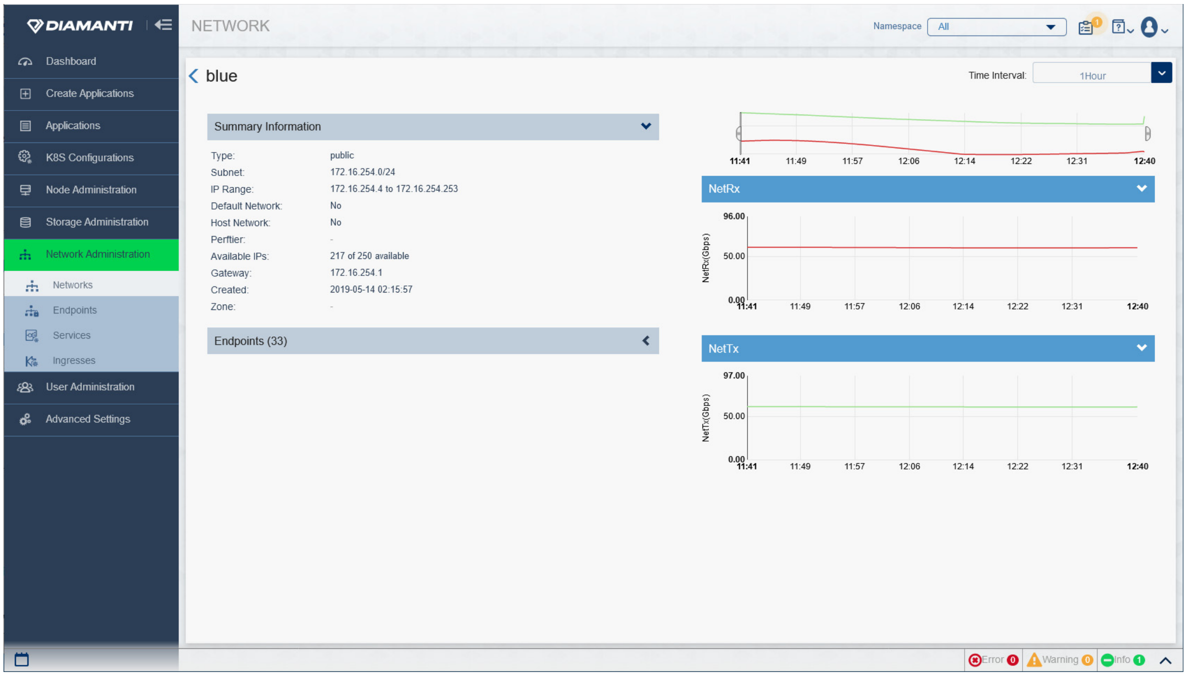
Health	Compute		Network	Storage	
	CPU	MEMORY	THROUGHPUT	IOPS	CAPACITY
	29.7 Cores 25% Utilization	3 GiB 1% Utilization	104.0 Gbps 87% Utilization	1.6 Read (M) 1.5 Write (M)	271 GB 3% Utilization

28 pods Export

NAME	STATUS	PRIMARY IP	CPU CORES	MEMORY	IOPS/R	IOPS/W	NETWORK TX	NETWORK RX	ACTIONS
ip-appserv76-best-effort-1-z7dqw (3)	✓ Good	172.16.254.8	0.1	148 MiB	16 K	0	43.48 Mbps	45.81 Mbps	
ip-appserv76-best-effort-2-f7jgx (3)	✓ Good	172.16.254.9	0.1	148 MiB	16 K	0	91.38 Mbps	92.28 Mbps	
ip-appserv76-best-effort-3-f2nbw (3)	✓ Good	172.16.254.10	0.1	148 MiB	16 K	0	42.81 Mbps	43.27 Mbps	
ip-appserv76-high-1-5zg4x (3)	✓ Good	172.16.254.14	2.1	83 MiB	261 K	0	3.47 Gbps	3.88 Gbps	
ip-appserv76-high-2-d9vsm (3)	✓ Good	172.16.254.15	2.5	94 MiB	261 K	0	5.62 Gbps	5.47 Gbps	
ip-appserv76-high-3-7zxzq (3)	✓ Good	172.16.254.18	1.9	115 MiB	260 K	0	4.08 Gbps	3.93 Gbps	
ip-appserv76-medium-1-lts8m (3)	✓ Good	172.16.254.11	0.9	148 MiB	63 K	0	1.50 Gbps	1.70 Gbps	
ip-appserv76-medium-2-ns6rl (3)	✓ Good	172.16.254.12	0.6	148 MiB	63 K	0	947.71 Mbps	978.81 Mbps	
ip-appserv76-medium-3-g2qnl (3)	✓ Good	172.16.254.13	0.7	148 MiB	63 K	0	1.48 Gbps	1.30 Gbps	
ip-appserv77-best-effort-1-qtzdp (3)	✓ Good	172.16.254.19	0.1	148 MiB	10 K	4 K	46.18 Mbps	43.39 Mbps	

Figure 2. Pods Screen

Navigate to the Network page and select the `blue` network to display the network view across *all* nodes in the cluster.



blue

Summary Information

Type: public
Subnet: 172.16.254.0/24
IP Range: 172.16.254.4 to 172.16.254.253
Default Network: No
Host Network: No
Perftier: -
Available IPs: 217 of 250 available
Gateway: 172.16.254.1
Created: 2019-05-14 02:15:57
Zone: -

Endpoints (33)

Figure 3. Network Details Page

Across the three nodes in this example, there is approximately 60Gbps of network receive traffic and 60Gbps transmit traffic for all containers in the cluster.

Finally, navigate to the Dashboard to display the QoS across both network and storage.

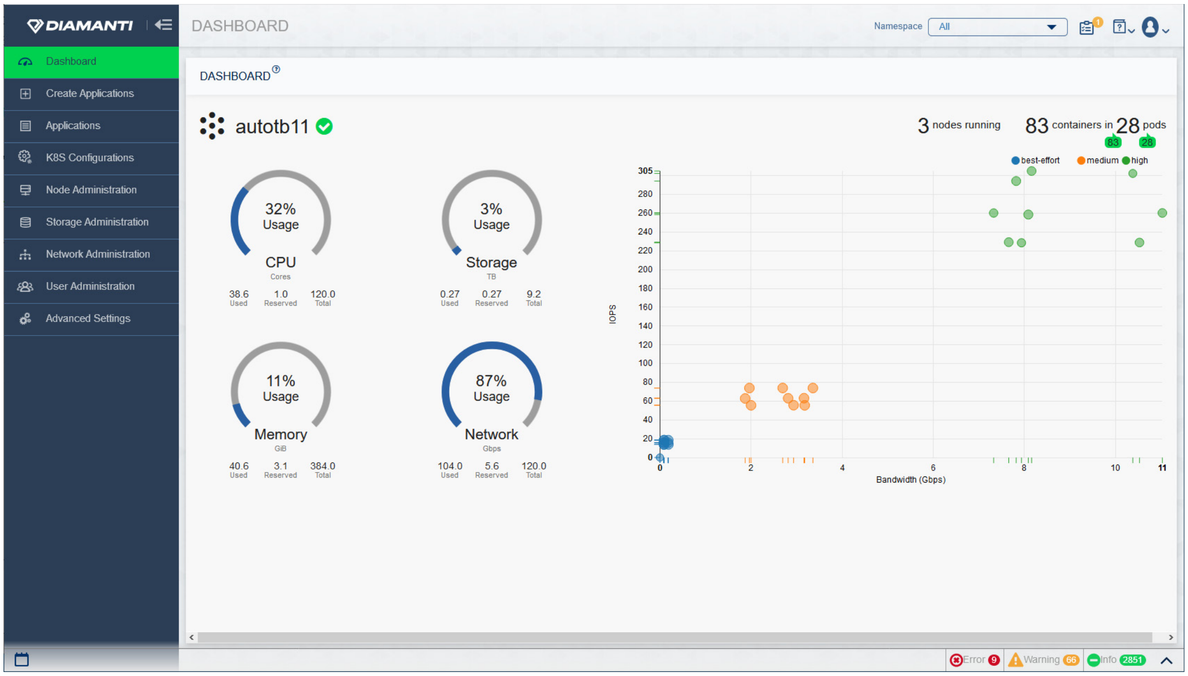


Figure 4. Diamanti UI Dashboard

A

Demo Script and Pod Definition Template

This appendix provides listings of the `create-iperf-fio.sh` demo script and the `iperf-fio.json` pod definition template file.

Demo Script Listing

The `create-iperf-fio.sh` script creates 9 pods per node, and runs FIO and iperf traffic across these pods. The following shows a listing of the `create-iperf-fio.sh` script:

```
#!/bin/bash
create_pods() {
        node=$1
        dest=$2
        net=$3
        qos=$4
        op=$5
        per_type=$6
        num=0
        j=0
        case $qos in
            high) num=3 ;;
            medium) num=3 ;;
            best-effort) num=3;;
        esac
        for i in `seq 1 $num`
        do
             dctl volume create vol-$node-$qos-$i -s 10G --sel kubernetes.io/
hostname=$node
              sed -e 's/NET/'$net'/g' -e 's/NODE/'$node'/g' -e 's/QOS/'$qos'/
g' -e 's/INDEX/'$i'/g' -e 's/DEST/'$dest'/g' -e 's/OP_TYPE/'$op'/g' -e 's/
PER_TYPE/'$per_type'/g' iperf-fio.
json | kubectl create -f -
        done
}
create_pods_with_qos() {
    for qos in best-effort medium high
    do
       create_pods $1 $2 $3 $qos $4 $5
    done
}
create_pods_with_qos $1 $2 $4 randrw 100
create_pods_with_qos $2 $3 $4 randrw 70
create_pods_with_qos $3 $1 $4 randrw 0
```

Pod Definition Template Listing

The `iperf-fio.json` file is a pod definition template that is customized by the `create-iperf-fio.sh` script to deploy the demo pods. The following shows a listing of the `iperf-fio.json` file:

```
{
    "apiVersion": "v1",
    "items": [
        {
            "apiVersion": "v1",
            "kind": "ReplicationController",
            "metadata": {
                "name": "ip-NODE-QOS-INDEX"
            },
            "spec": {
                "template": {
                    "metadata": {
                        "annotations": {
                            "diamanti.com/endpoint0":
"{\"network\":\"NET\",\"perfTier\":\"QOS\"}"
                        },
                        "labels": {
                            "diamanti.com/app": "ip-NODE-QOS-INDEX"
                        }
                    },
                    "spec": {
                        "nodeSelector": {
                                "kubernetes.io/hostname" : "NODE"
                        },
                        "containers": [
                            {
                                "env": [
                                    {
                                        "name": "RT",
                                        "value": "60000"
                                    },
                                    {
                                        "name": "RW",
                                        "value": "OP_TYPE"
                                    },
                                    {
                                        "name":"RWRPCT",
                                        "value": "PER_TYPE"
                                   },
                                    {
                                        "name": "QD",
                                        "value": "16"
                                    },
                                    {
                                        "name": "BLOCK_SIZE",
                                        "value": "4k"
                                    }
                                ],
                                "image": "diamanti/fio-rw:0.1",
                                "imagePullPolicy": "IfNotPresent",
                                "name": "fio",
                                "volumeMounts": [
                                    {
                                        "mountPath": "/data",
                                        "name": "vol-NODE-QOS-INDEX"
                                    }
                                ]
                            },
                            {
                                "command": [
                                    "/usr/bin/iperf3",
                                    "-s"
                                ],
                                "image": "diamanti/iperf:v0.1",
                                "imagePullPolicy": "IfNotPresent",
                                "name": "server"
                            },
                            {
                                "command": ["/bin/sh", "-c"],
                              "args": [ "sleep 60 && /usr/bin/iperf3 -M 1454
-c ip-DEST-QOS-INDEX -P 8 -t 60000"],
                                "image": "diamanti/iperf:v0.1",
                                "imagePullPolicy": "IfNotPresent",
                                "name": "client"
                            }
                        ],
                        "dnsPolicy": "ClusterFirst",
                        "restartPolicy": "Always",
                        "terminationGracePeriodSeconds": 30,
                        "volumes": [
                            {
                                "name": "vol-NODE-QOS-INDEX",
                                "flexVolume": {
                                    "driver": "diamanti.com/volume",
                                    "fsType": "xfs",
                                    "options": {
                                        "name": "vol-NODE-QOS-INDEX",
                                        "perfTier": "QOS",
                                        "type": "Simple"
                                    }
                                }
                            }
                        ]
                    }
                }
            }
        },
        {
            "apiVersion": "v1",
            "kind": "Service",
            "metadata": {
                "name": "ip-NODE-QOS-INDEX"
            },
            "spec": {
                "clusterIP": "None",
                "ports": [
                    {
                        "name": "tcp",
                        "port": 5201,
                        "protocol": "TCP",
                        "targetPort": 5201
                    }
                ],
                "selector": {
                    "diamanti.com/app": "ip-NODE-QOS-INDEX"
                }
            }
        }
    ],
    "kind": "List",
    "metadata": {}
}
```

Legal Notices

Publication Date: This document was published on January 7, 2020.

Publication Number: DM-TNcapabilities-20200107-01

Copyright

Diamanti believes the information it furnishes to be accurate and reliable. However, Diamanti assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of Diamanti except as specifically described by applicable user licenses. Diamanti reserves the right to change specifications at any time without notice.

Trademarks