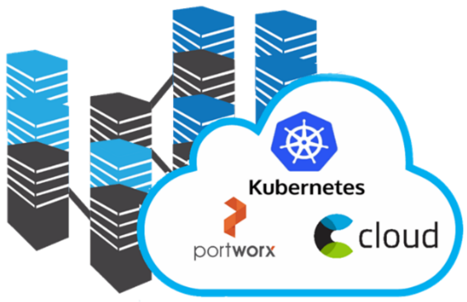
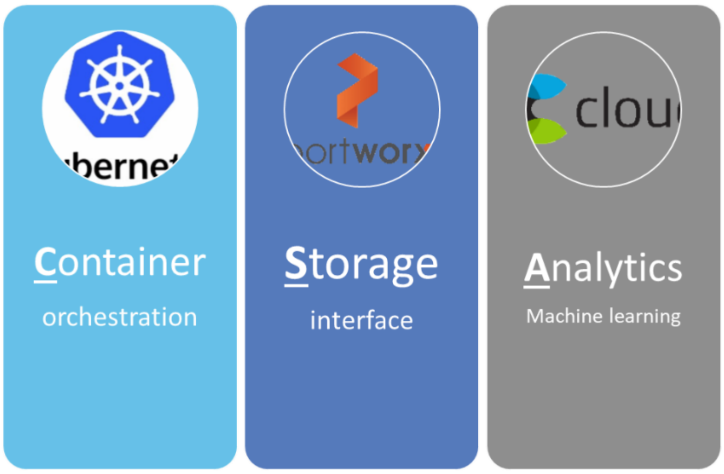
CSA - Container Storage Analytics Appliance

CSA Appliance Whitepaper
PoC conducted in Lab 25th June 2019 by Hyperscalers





HYPER SCALERS is an Australian registered company.
ABN - 83 600 687 223
ACN - 600 687 223

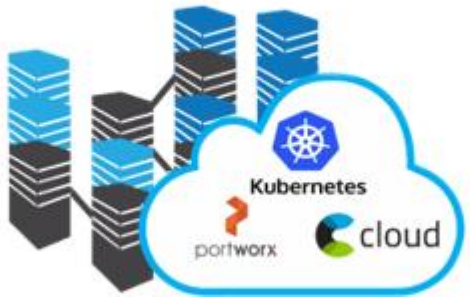
Table of Contents

1. Executive Summary 3
2. Introduction 4
3. CSA Appliance 5
4. Appliance architecture 6
5. Kubernetes Dashboard 7
6. Portworx CSI Solution 7
7. Elastic database analytics 8
8. Appliance accessibility .... **Error! Bookmark not defined.**
9. Conclusion 11

1. Executive Summary

Hyperscalers has developed a **Digital-IP-Appliance Design Process** along with a utility being the **Appliance Optimiser Utility** which together assists in productizing appliance(s) that enable providers with everything they need (inc. hardware, software, configuration & support) to hyperscale their services quickly, reliably and at a fraction of historical costs.

The problem: Many enterprises who have tried to use Kubernetes for stateful applications at scale have stumbled when using the platform for services like databases, queues, key-value stores, automation and machine learning.



The objective of this project was to pre-engineer, optimise, test and productize an appliance called **"CSA"** that could solve these enterprise challenges.

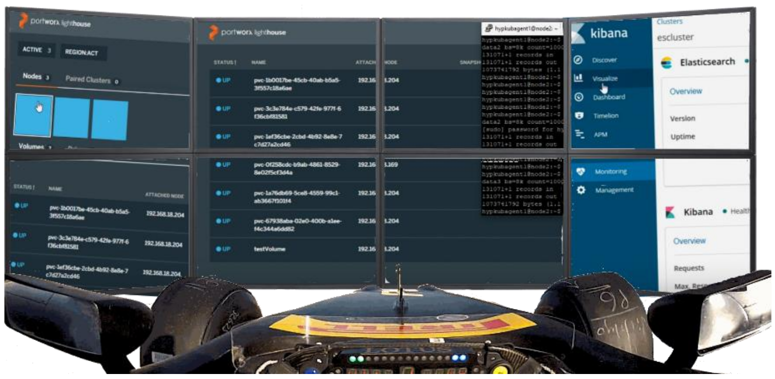
The solution: Service providers using the CSA appliance could deploy and scale in 2RU 4-node building blocks gaining the following capabilities out of the box:

- **C**ontainer orchestration using Kubernetes;
- **S**torage Interface using Port Worx for running stateful applications tested at 950MB /s; and
- **A**nalytics with Machine Learning using Elastic Search

S5S Building Block – 2RU 4Nodes

see video https://www.youtube.com/watch?v=ndk9zuwKWUE

Test drives of CSA are free to partners via Hyperscalers' Lab as a Service **LaaS**.



HYPER SCALERS is an Australian registered company.
ABN - 83 600 687 223
ACN - 600 687 223

2. Introduction

About Kubernetes is an open source container orchestration platform that simplifies the development and management of applications bringing greater ease and agility for developers, operators, and data scientists alike. Kubernetes offers:

Distributed System	As a distributed system, Kubernetes includes a group of agent nodes that are coordinated by a group of master nodes. Like other distributed systems, several of the components running on the master nodes perform leader election with their peers.
Cluster Manager	As a cluster manager, Kubernetes manages both resources and tasks running on the agent nodes. The agent nodes provide resources to the cluster.
Container Platform	As a container platform, Kubernetes includes two built-in task schedulers (Marathon and Metronome) and two container runtimes (Docker and Mesos). Combined, this functionality is commonly referred to as container orchestration.
Operating System	As an operating system, Kubernetes abstracts the cluster hardware and software resources and provides common services to applications.

About Portworx software provides cloud native storage and data management for Kubernetes. Containers makes it easy to build and run modern distributed applications in production at scale, by pooling resources across an entire datacenter or cloud. While the container platform works great for stateless applications, many enterprises who have tried to use Kubernetes for stateful applications at scale have stumbled when it comes to using the platform for services like databases, queues and key-value stores. Portworx, which scales up to 1000 nodes per cluster and is used in production by Kubernetes users, solves the operational and data management problems enterprises encounter when running stateful applications on Kubernetes.

About Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases. As the heart of the elastic, it centrally stores the databases, which can discover the data patterns and run analytics on them. Elasticsearch lets you perform and combine many types of searches — structured, unstructured, geo, metric etc.

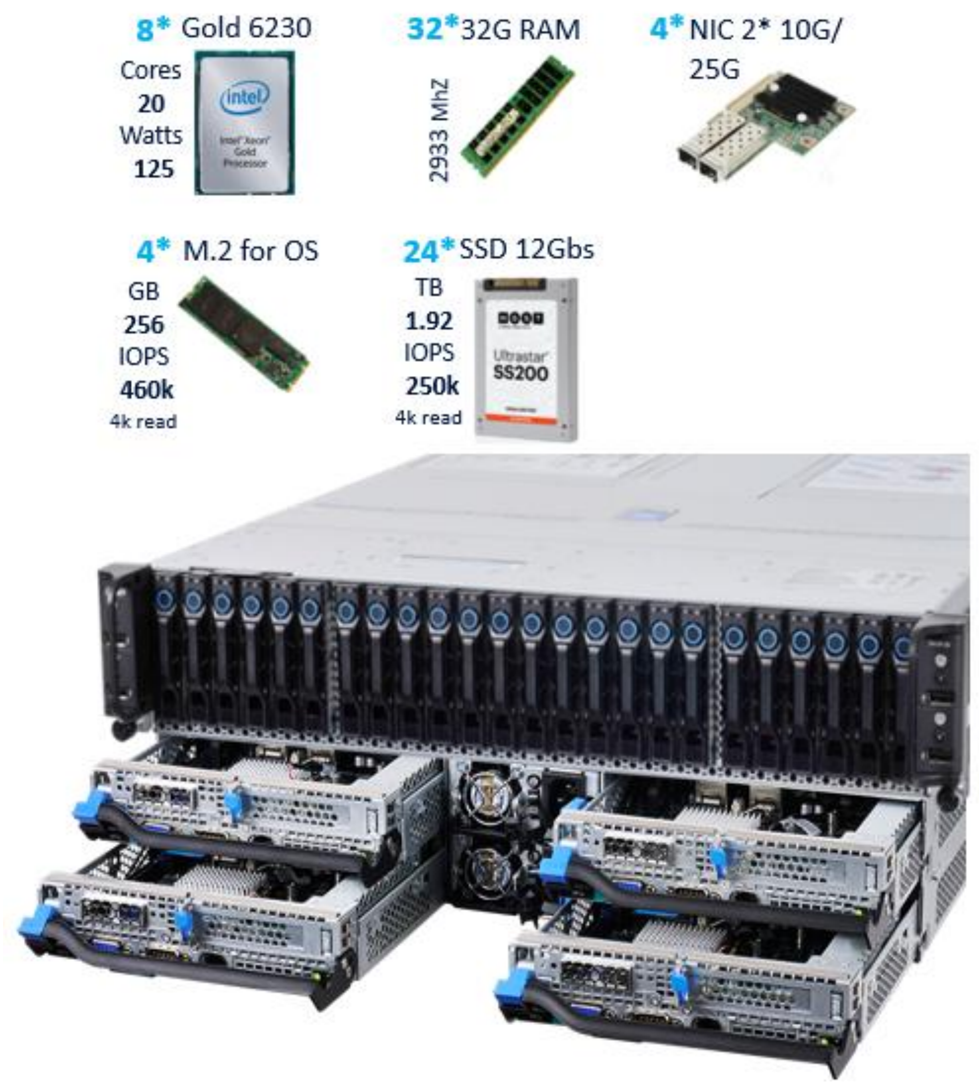
3. CSA Appliance

The CSA appliance consists of following hardware and software components:

Hardware **S5S Tier 1 Converged Server (4 nodes) (1 Master + 3 Agents)**

Download S5S data sheet

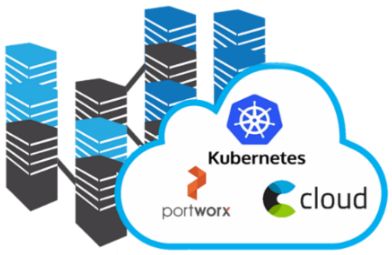
- 2x Intel Xeon Gold 6230 @ 3.9 GHz
- 256GB RAM: 8 x 32GB @ 2933 MHz
- OS SSD - M.2 PCIe SSD 256GB Samsung
- 6x1.9 TB HGST ULTRASTAR SS220 SAS SSD
- Storage Card - Mezz 12Gbps S5B QS 3216 16i SAS Mezz
- NIC - OCP Mezz 25Gb SFP28 2 port ConnectX-4

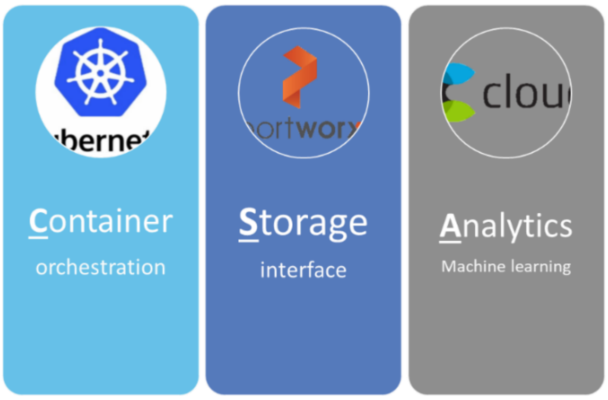


Switches

- IX2 – 25G leaf switch with 100G uplinks
- IX1 – 100G spine switch with 100G uplinks

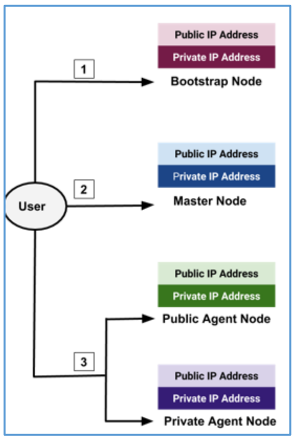
Software	• Centos 7.5 host OS (minimal) • Kubernetes 1.15.3 • Cumulus Linux ONOS v3.5.3 • Docker Version 1.13.1 • Portworx pxctl version 2.0.3.0 • Portworx-Elastic 2.5.0-6.3.2





4. Appliance architecture

The Kubernetes installation process requires a bootstrap node, master node, public agent node and a private agent node. In the PoC there are one master node, two agent nodes and one bootstrap node; both master and agent nodes are in a S5S and bootstrap node is a VM running on VmWare.



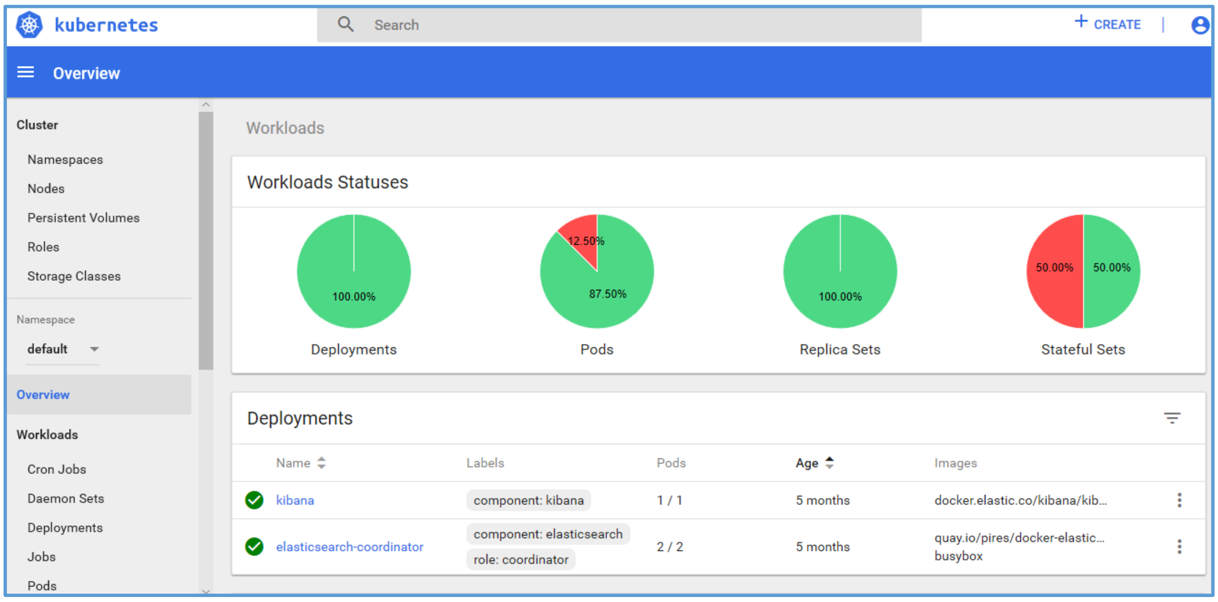
Master Nodes	Master node works with other master nodes to manage the rest of the cluster. Master nodes contain the bulk of the Kubernetes components, including it’s master process.
Agent Nodes	Agent node is a node on which user tasks are run. Agent nodes contain a few Kubernetes components, including an agent process. Agent nodes can be public or private, depending on agent and network configuration.
Bootstrap Node	Bootstrap node contains initial setup modules like ISO images and configuration script. This node provides boot environments for master and agent nodes for their installations and configuration.

Post installation; the bootstrap node VM can be de-commissioned as its needed for initial appliance installations only. All nodes and VM are initially installed with CentOS 7.5 as host operating system and they are configured in same vlan; so that they are accessible to each other via host IP addresses.

Kubernetes is installed in Hyperscalers environment by using a dynamically generated configuration file. This file is generated by using specific parameters that are set during configuration. This installation file contains a Bash install script and a Docker container that is loaded with everything needed to deploy a customized Kubernetes build. The Docker container contains all of the elements for Kubernetes so it can be used for offline installation. The Kubernetes installation process requires a cluster of nodes to install Kubernetes onto and a single node to run the Kubernetes installation from. For fully functional clusters on any infrastructure including on-premise, public or private clouds, follow the On-Prem installation instructions.

5. Kubernetes Dashboard

Kubernetes is composed of many open source microservice components meticulously tuned and configured to work together. It includes most of the open source components but also includes several additional components, modules, and plugins.

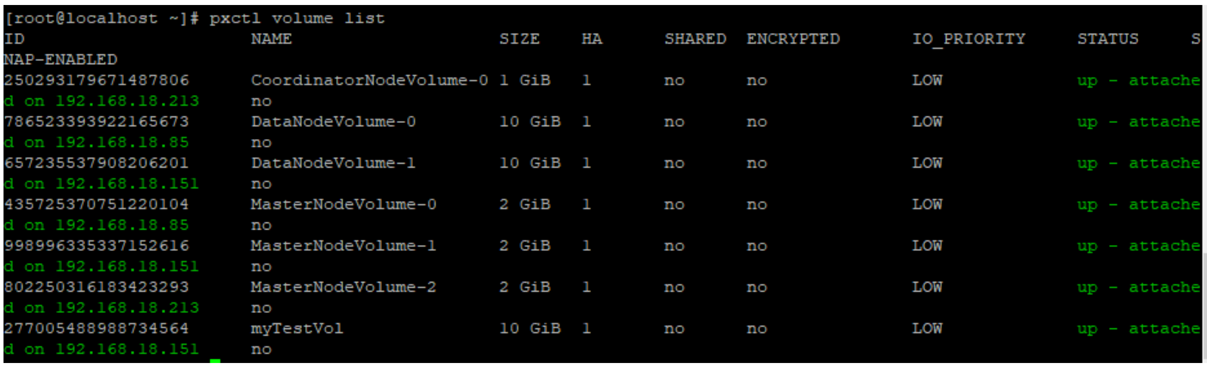


Deployments

Name	Labels	Pods	Age	Images
kibana	component: kibana	1 / 1	5 months	docker.elastic.co/kibana/kib...
elasticsearch-coordinator	component: elasticsearch role: coordinator	2 / 2	5 months	quay.io/pires/docker-elastic... busybox

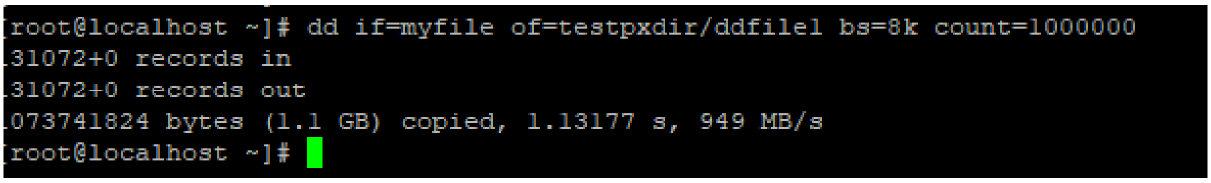
6. Portworx Storage Performance

Portworx is available as installable in the Catalog section of its product page. It requires atleast three agent nodes for installations and a dedicated block storage disk on all three agent nodes. In the PoC; there are 1.9TB dedicated SAS ssd in each agent nodes. After installations; the agent nodes would create a distributed pool of storage with CSI (Container Storage Interface) to be used by the containers on kubernetes.



```
[root@localhost ~]# pxctl volume list
ID                      NAME                    SIZE    HA      SHARED  ENCRYPTED       IO_PRIORITY     STATUS          S
NAP-ENABLED
250293179671487806      CoordinatorNodeVolume-0 1 GiB   1       no      no              LOW             up - attache
d on 192.168.18.213     no
786523393922165673      DataNodeVolume-0        10 GiB  1       no      no              LOW             up - attache
d on 192.168.18.85      no
657235537908206201      DataNodeVolume-1        10 GiB  1       no      no              LOW             up - attache
d on 192.168.18.151     no
435725370751220104      MasterNodeVolume-0      2 GiB   1       no      no              LOW             up - attache
d on 192.168.18.85      no
998996335337152616      MasterNodeVolume-1      2 GiB   1       no      no              LOW             up - attache
d on 192.168.18.151     no
802250316183423293      MasterNodeVolume-2      2 GiB   1       no      no              LOW             up - attache
d on 192.168.18.213     no
277005488988734564      myTestVol               10 GiB  1       no      no              LOW             up - attache
d on 192.168.18.151     no
```

The screenshot shows "pxtrl" portwork CLI listing the volumes created by different containers. In the PoC; we created a sample volume "myTestVol" of 10GB size to execute the storage performance scripts. The volume is mounted on a Linux root file systems and DD command is executed.



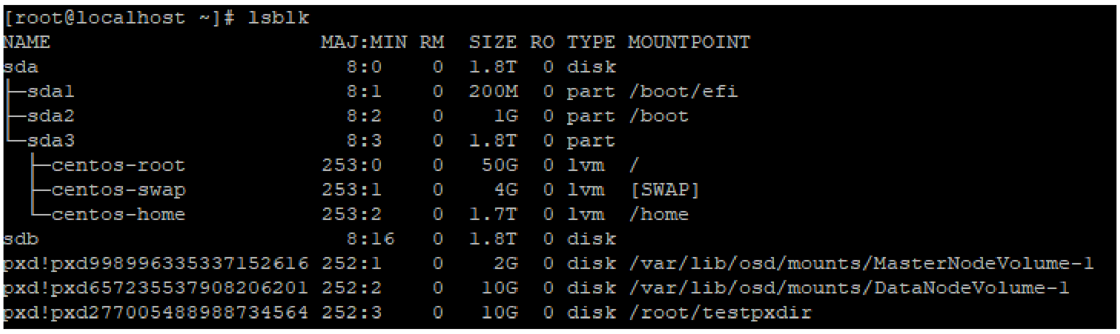
```
[root@localhost ~]# dd if=myfile of=testpxdir/ddfile1 bs=8k count=1000000
.31072+0 records in
.31072+0 records out
.073741824 bytes (1.1 GB) copied, 1.13177 s, 949 MB/s
[root@localhost ~]#
```

The input file "myfile" is a 1GB file with random data and dd command shows storage performance of around 950GB/s. The performance number is better than some of other container storage solutions; as discussed in following paper.

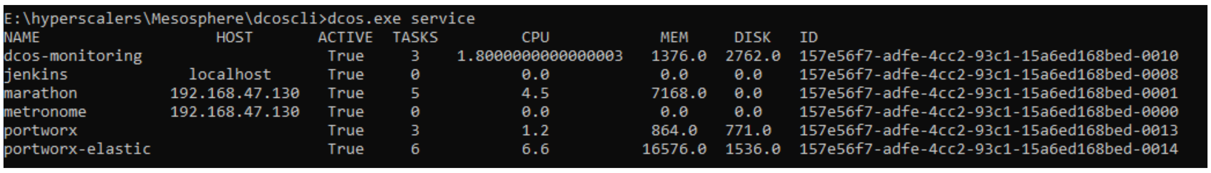
https://medium.com/vescloud/kubernetes-storage-performance-comparison-9e993cb27271

7. Elastic database analytics

Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases. As the heart of the elastic, it centrally stores the databases; which can discover the data patterns and run analytics on them. Elasticsearch lets you perform and combine many types of searches — structured, unstructured, geo, metric etc. in multiple ways. Elasticsearch uses standard RESTful APIs and JSON and maintains clients in many languages such as Java, Python, .NET, SQL, and PHP. Kubernetes with Portworx provides container image to install the Elasticsearch service on Kubernetes cluster backed by PX volumes for persistent storage.



```
[root@localhost ~]# lsblk
NAME                       MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                          8:0    0   1.8T  0 disk
├─sda1                       8:1    0   200M  0 part /boot/efi
├─sda2                       8:2    0     1G  0 part /boot
└─sda3                       8:3    0   1.8T  0 part
  ├─centos-root            253:0    0    50G  0 lvm  /
  ├─centos-swap            253:1    0     4G  0 lvm  [SWAP]
  └─centos-home            253:2    0   1.7T  0 lvm  /home
sdb                          8:16   0   1.8T  0 disk
pxd!pxd998996335337152616 252:1    0     2G  0 disk /var/lib/osd/mounts/MasterNodeVolume-1
pxd!pxd657235537908206201 252:2    0    10G  0 disk /var/lib/osd/mounts/DataNodeVolume-1
pxd!pxd277005488988734564 252:3    0    10G  0 disk /root/testpxdir
```



```
E:\hyperscalers\Mesosphere\dcoscli>dcos.exe service
NAME                   HOST       ACTIVE  TASKS          CPU             MEM     DISK    ID
dcos-monitoring                    True     3     1.8000000000000003   1376.0   2762.0   157e56f7-adfe-4cc2-93c1-15a6ed168bed-0010
jenkins             localhost      True     0            0.0             0.0     0.0     157e56f7-adfe-4cc2-93c1-15a6ed168bed-0008
marathon         192.168.47.130    True     5            4.5           7168.0    0.0     157e56f7-adfe-4cc2-93c1-15a6ed168bed-0001
metronome        192.168.47.130    True     0            0.0             0.0     0.0     157e56f7-adfe-4cc2-93c1-15a6ed168bed-0000
portworx                           True     3            1.2            864.0   771.0    157e56f7-adfe-4cc2-93c1-15a6ed168bed-0013
portworx-elastic                   True     6            6.6          16576.0  1536.0    157e56f7-adfe-4cc2-93c1-15a6ed168bed-0014
```

The Elastic installation creates Data and Master node volumes and they can be used as storage for database applications. The “portworx-elastic” service is seen to be Active in the cli after complete installations.

The PoC executes examines ElasticSearch REST API and demonstrates basic operations using HTTP requests only. It would create a database with multiple entries and execute the search APIs to verify the features on cli; using “pxctl” commands.

```
[root@localhost ~]# curl -s -u elastic:changeme -XPUT 'coordinator.portworx-elastic.l4lb.thisdcos.directory:9200/text/article/4?pretty'   -H 'Content-Type: application/json' -d '
{
  "title": "Old education",
  "random_text": "Old education him departure any arranging one prevailed."
}'
```

We used JSON XPUT commands to create multiple entries as shown above in “/text/article” folder od elastic search volume. It’s possible to have multiple indexes, which in turn contain multiple types. These types hold multiple documents, and each document has multiple fields. We are going to store documents using the following scheme:

text: The index name.
article: The type name.
id: The ID of this particular example text-entry.

There are four such entries created in the database:

```
[root@localhost ~]# curl -s -u elastic:changeme -XGET 'coordinator.portworx-elastic.l4lb.thisdcos.directory:9200/text/article/_search?pretty' \
> -H 'Content-Type: application/json' -d '
> {
>   "query": {
>     "match_all": {}
>   }
> }'
{
  "took" : 49,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 4,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "text",
        "_type" : "article",
        "_id" : "2",
        "_score" : 1.0,
        "_source" : {
          "title" : "He oppose",
          "random_text" : "He oppose at thrown desire of no. Announcing impression unaffected day his are unreserved indulgence."
        }
      },
      {
        "_index" : "text",
        "_type" : "article",
        "_id" : "4",
        "_score" : 1.0,
        "_source" : {
          "title" : "Old education",
          "random_text" : "Old education him departure any arranging one prevailed."
        }
      },
      {
        "_index" : "text",
        "_type" : "article",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "title" : "He went",
          "random_text" : "He went such dare good fact. The small own seven saved man age."
        }
      },
      {
        "_index" : "text",
        "_type" : "article",
        "_id" : "3",
        "_score" : 1.0,
        "_source" : {
          "title" : "Repulsive questions",
          "random_text" : "Repulsive questions contented him few extensive supported."
        }
      }
    ]
  }
}
```

The search command is executed for specific pattern:

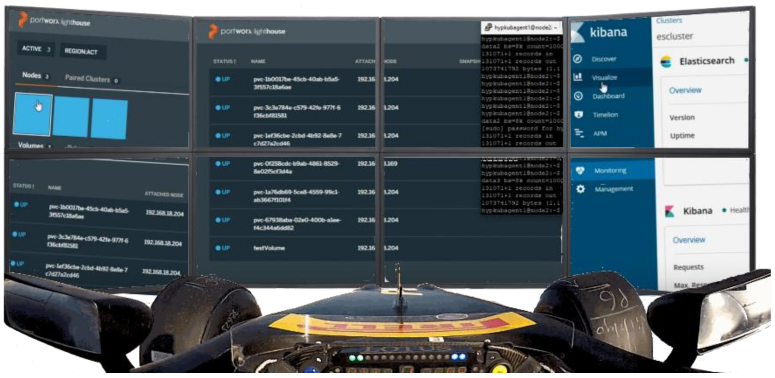
```
[root@localhost ~]# curl -s -u elastic:changeme -XGET 'coordinator.portworx-elastic.l4lb.thisdcos.directory:9200/text/article/_search?pretty' \
> -H 'Content-Type: application/json' -d '
> {
>   "query": {
>     "match": {
>       "random_text": "him departure"
>     }
>   }
> }'
{
  "took" : 280,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 2,
    "max_score" : 1.5834423,
    "hits" : [
      {
        "_index" : "text",
        "_type" : "article",
        "_id" : "4",
        "_score" : 1.5834423,
        "_source" : {
          "title" : "Old education",
          "random_text" : "Old education him departure any arranging one prevailed."
        }
      },
      {
        "_index" : "text",
        "_type" : "article",
        "_id" : "3",
        "_score" : 0.2876821,
        "_source" : {
          "title" : "Repulsive questions",
          "random_text" : "Repulsive questions contented him few extensive supported."
        }
      }
    ]
  }
}
```

The search is done for pattern “him departure” and we get two results with different scores. The first result is obvious because the text have the performed search inside of it and as we can see we have the score of 1.4513469. The second result is retrieved because the target document contains the word “him”. By default, Elasticsearch sorts matching results by their relevance score, that is, by how well each document matches the query. Note, that the score of the second result is small relative to the first hit, indicating lower relevance.

8. Test Drive the Appliance

The appliance can be accessible to the customers using WAP DDNS "http://hyperscalers.asuscomm.com/". Depending on the customer requirements; the administrator can open a port accessible via DDNS VPN.

Test drives of CSA are free to partners via Hyperscalers' Lab as a Service **LaaS**.



9. Conclusion

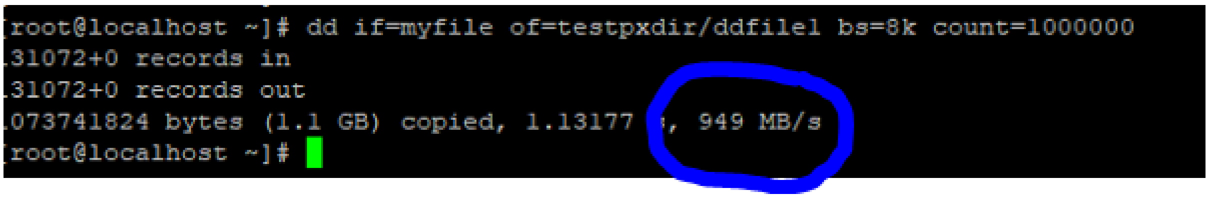
This paper proves CSA appliance works out of the box offering everything providers will need to turn around services such as:

- Container orchestration using Kubernetes;
- Storage Interface using Port Worx for running stateful applications tested at 950MB /s; and
- Analytics with Machine Learning using Elastic Search

Despite the CSA appliance setting new storage records: Read IOPS of 950MB/s using SAS SSD being twice that of the nearest container storage solution 500MB/s even greater performance gains could be achieved using better CPU and faster storage.

S5S Building Block – 2RU 4Nodes

see video https://www.youtube.com/watch?v=ndk9zuwKWUE



```
[root@localhost ~]# dd if=myfile of=testpxdir/ddfile1 bs=8k count=1000000
31072+0 records in
31072+0 records out
073741824 bytes (1.1 GB) copied, 1.13177 s, 949 MB/s
[root@localhost ~]#
```

HYPER SCALERS is an Australian registered company.
ABN - 83 600 687 223
ACN - 600 687 223