

AEROSPIKE
NEXTGEN
NOW
SUMMIT '20

Operations @ Scale with Aerospike Management API(s)

PayPal

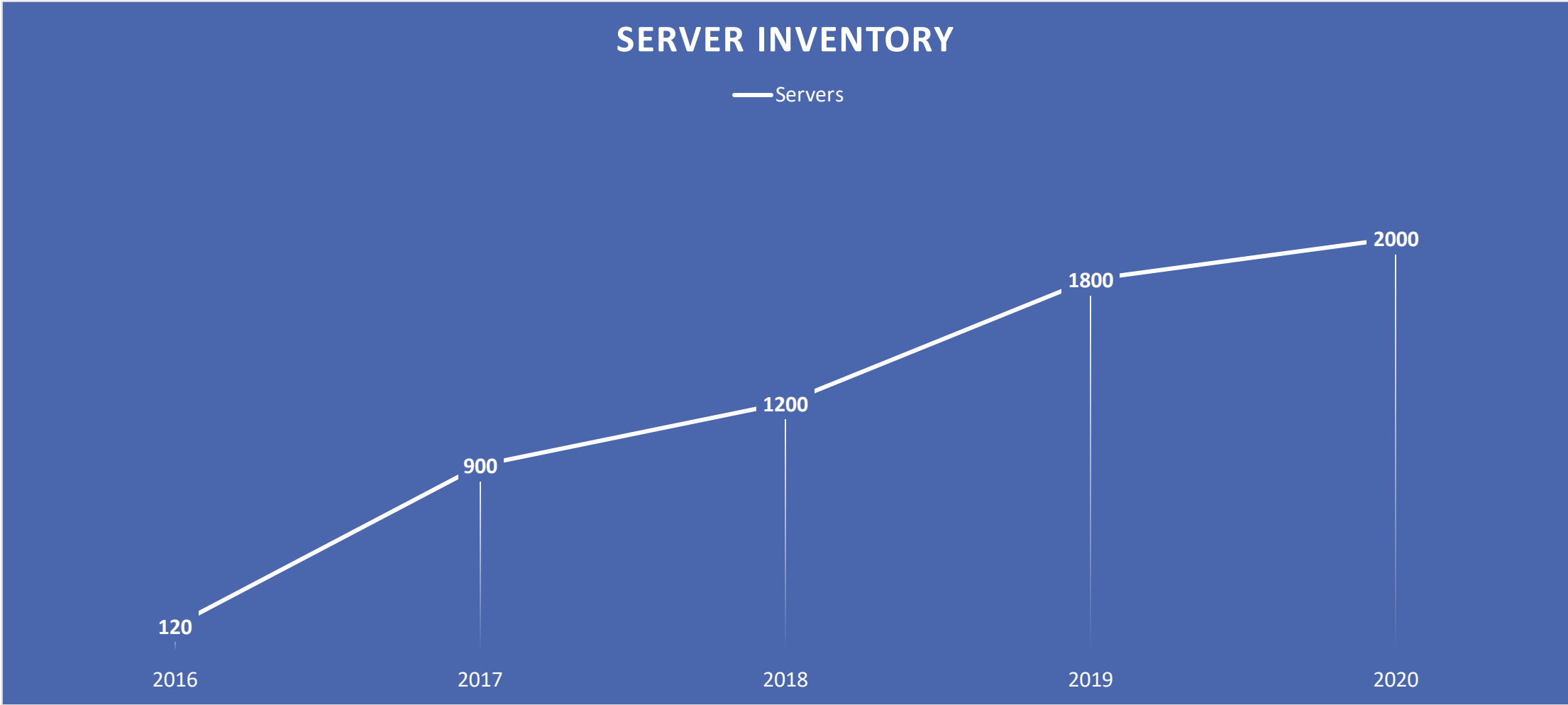
Agenda

- Growth Journey
- Automation Framework
- Monitoring @ Scale

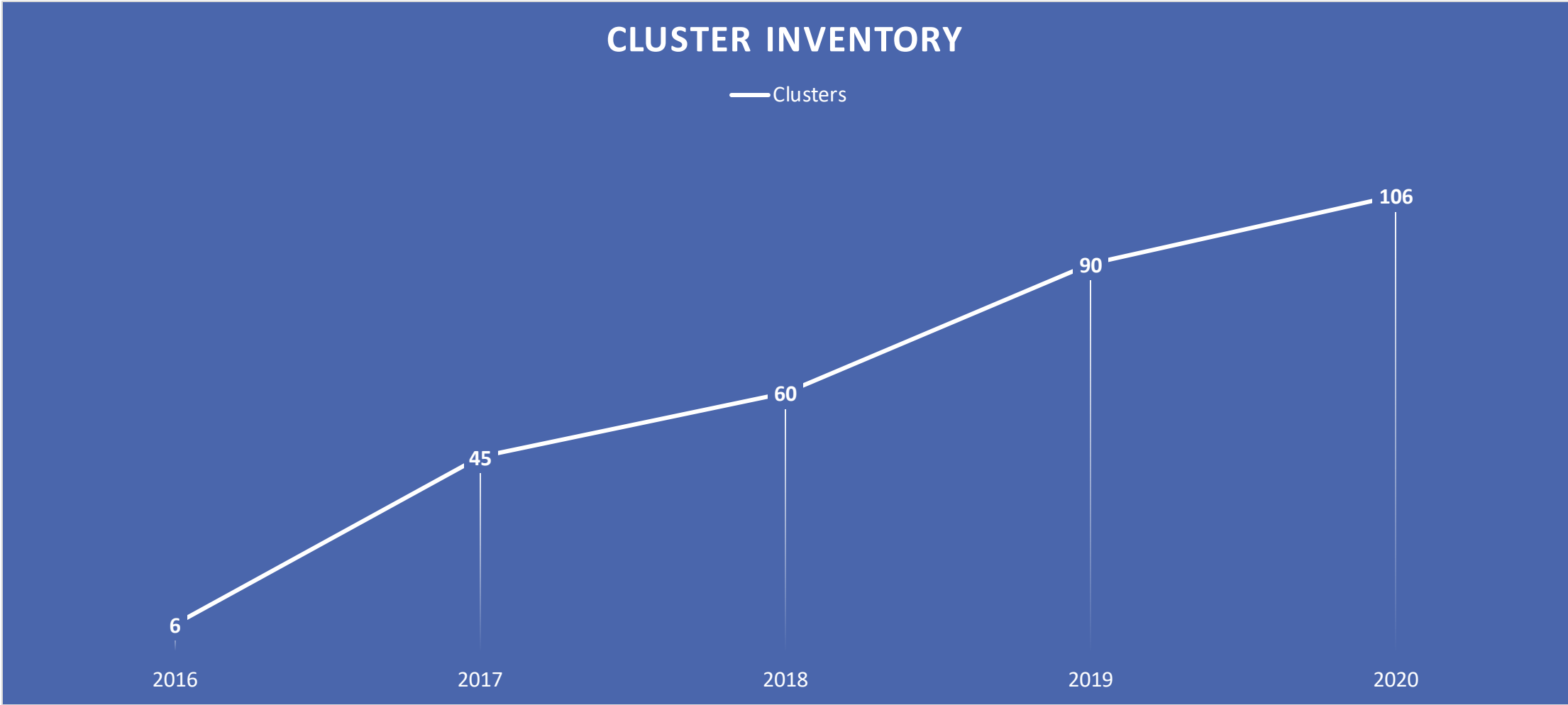


Growth Journey

Aerospike - 5 Years Journey

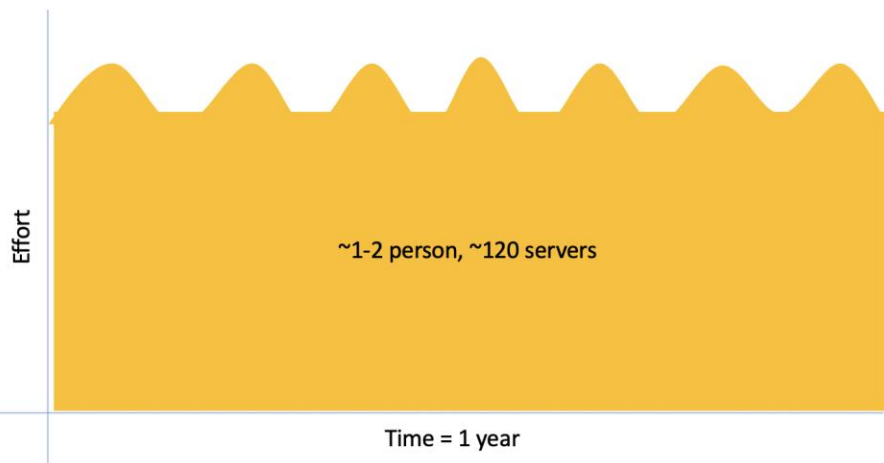


Aerospike - 5 Years Journey



Year 1 - Operations

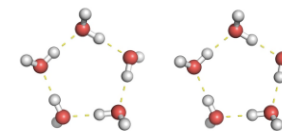
2016



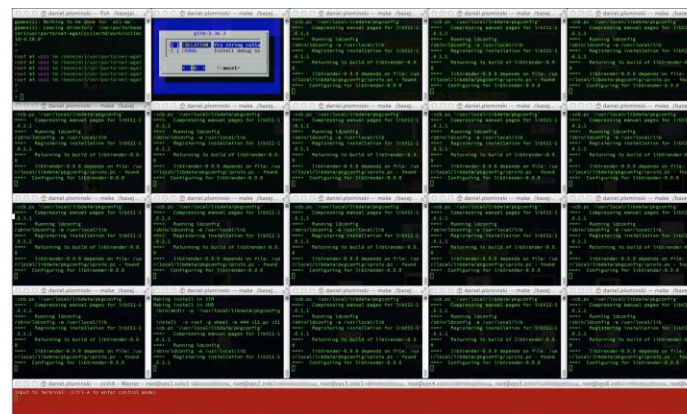
Servers = 120



Clusters = 2



Shell Scripts = 20

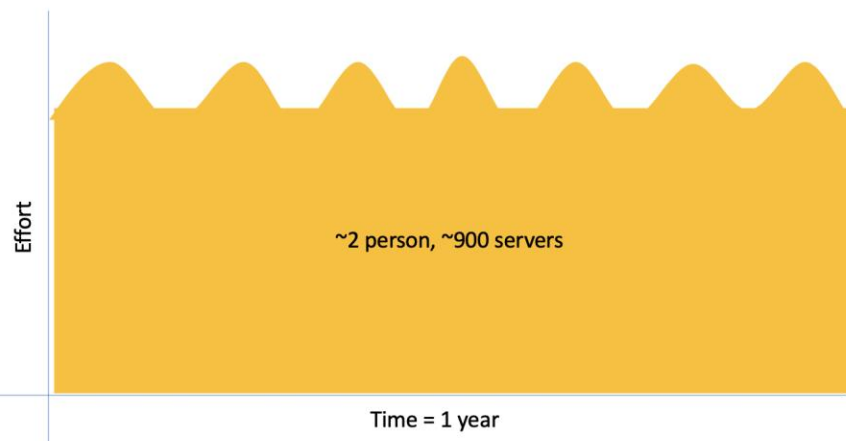


Team Size = 2



csshx

Year 2 - Operations

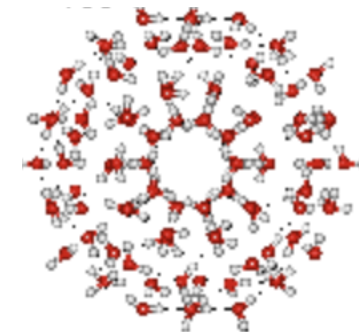


Servers = 900

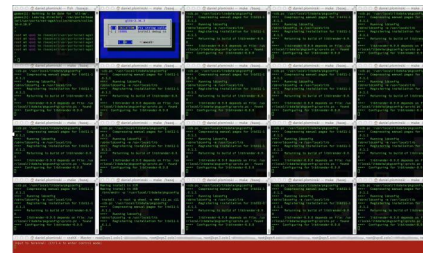


2017

Clusters = 45



Shell Scripts = 20



csshx

Team Size = 3



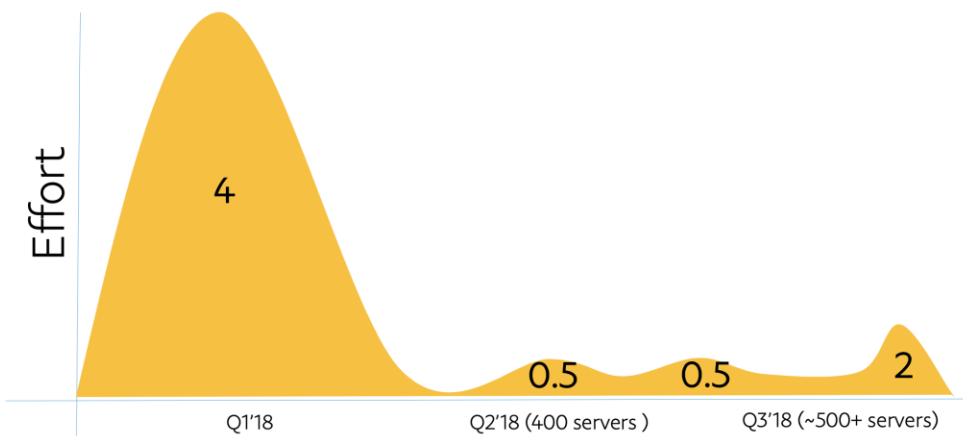
Production POC

Create cluster

Upgrade cluster



Year 3 - Operations

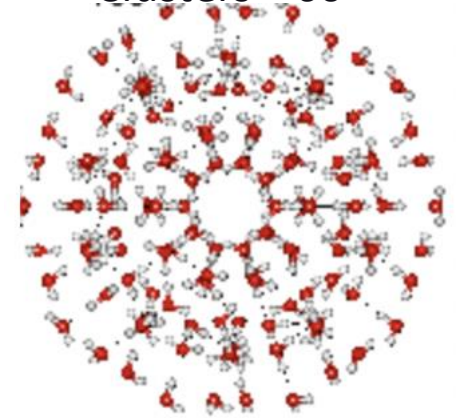


Servers = 1200

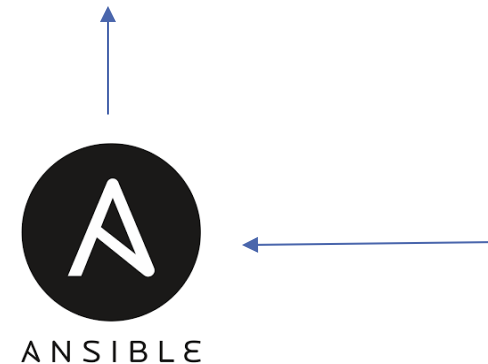


2018

Clusters = 60



Team Size = 4

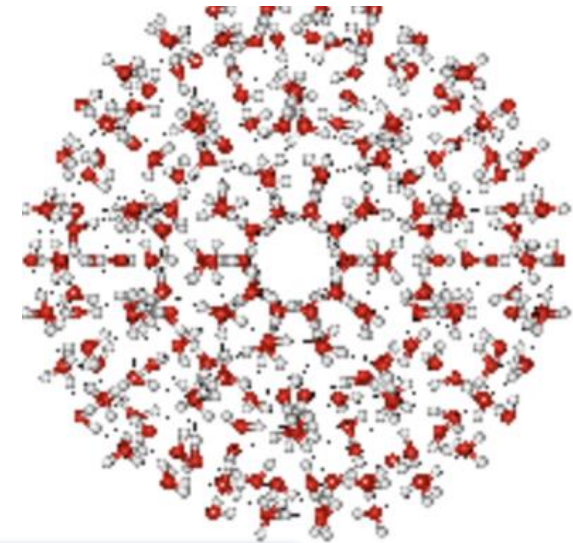
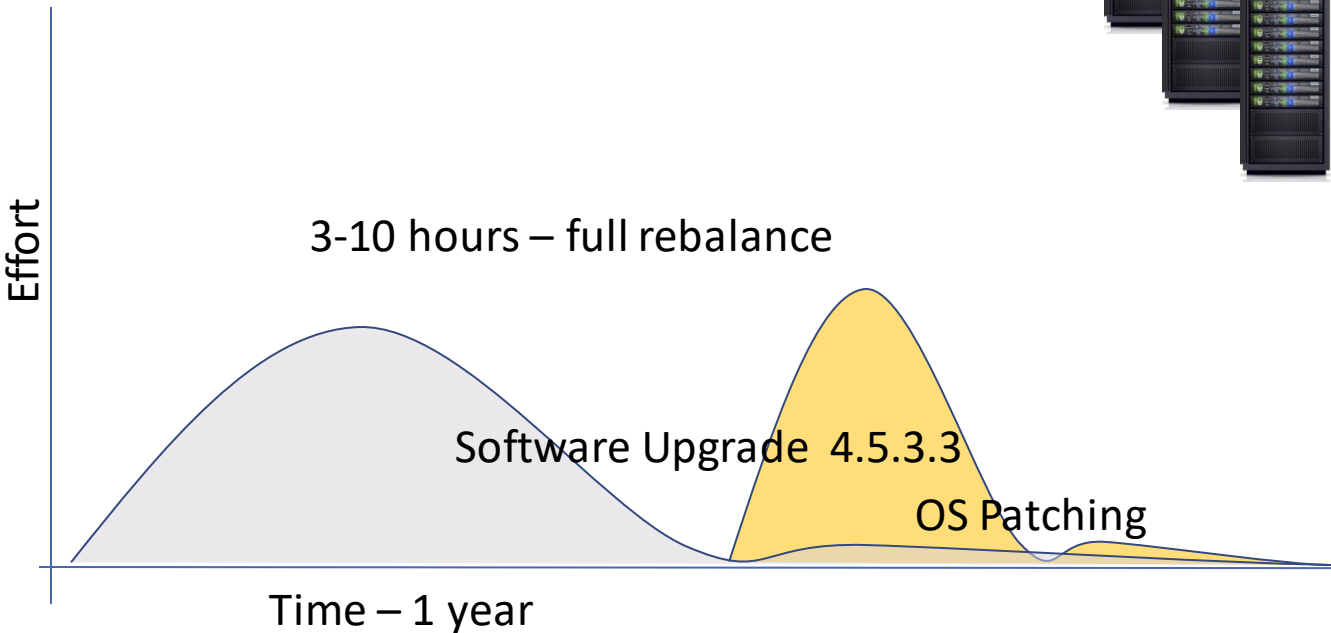


Year 4 - Operations

Servers = 1800

2019

Clusters = 90



ANSIBLE

Team Size = 4



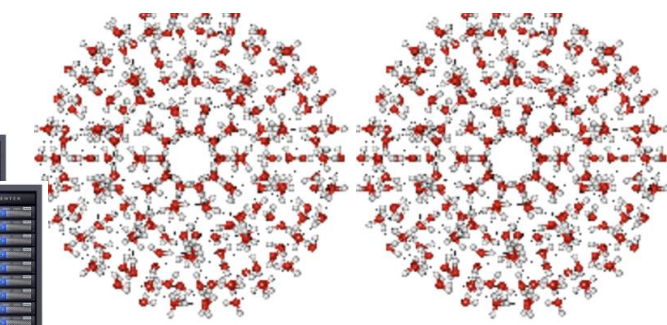
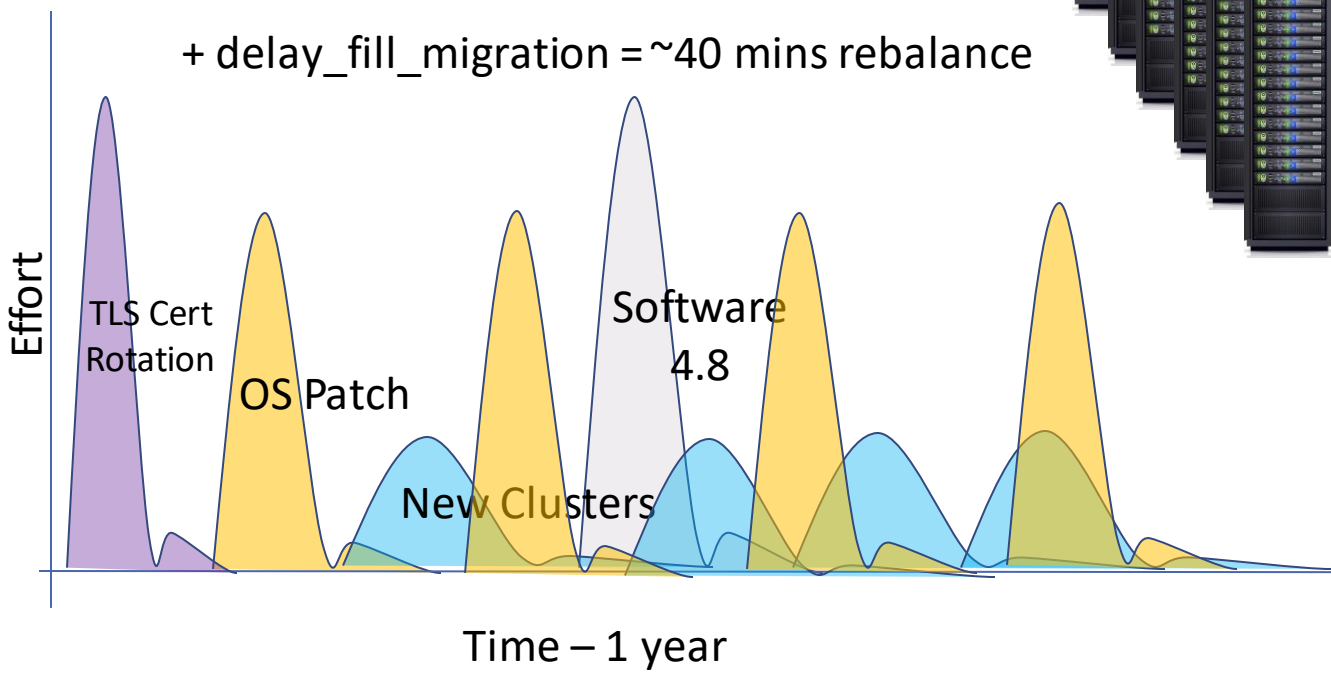
Year 5 - Operations

Servers = 2400

2020

Clusters = 150

+ delay_fill_migration = ~40 mins rebalance



ANSIBLE

Team Size = 4

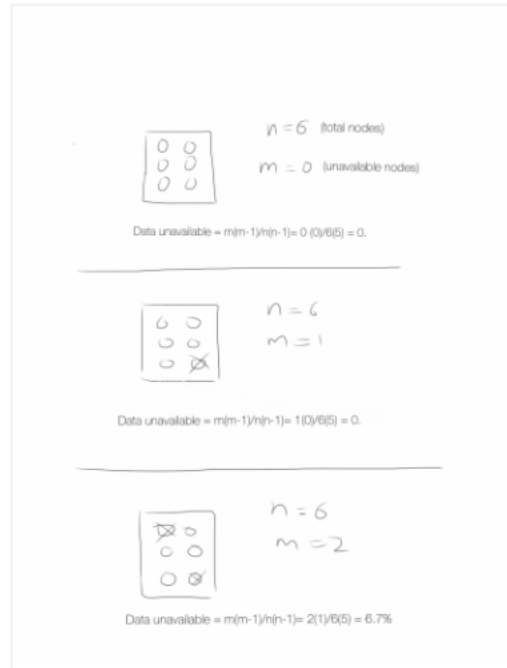


Deployment Architecture

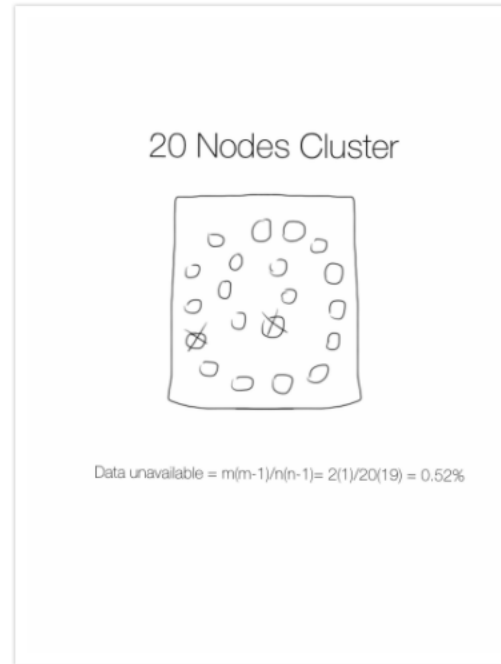
- 3-cluster group
- Primary, LDR, DR
- RF = 2
- 20 nodes per cluster

Smaller Manageable Clusters

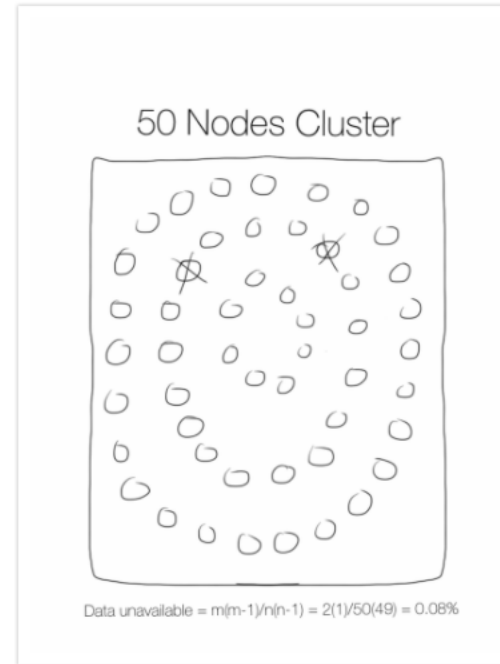
Example: Small Cluster (2 node failure)

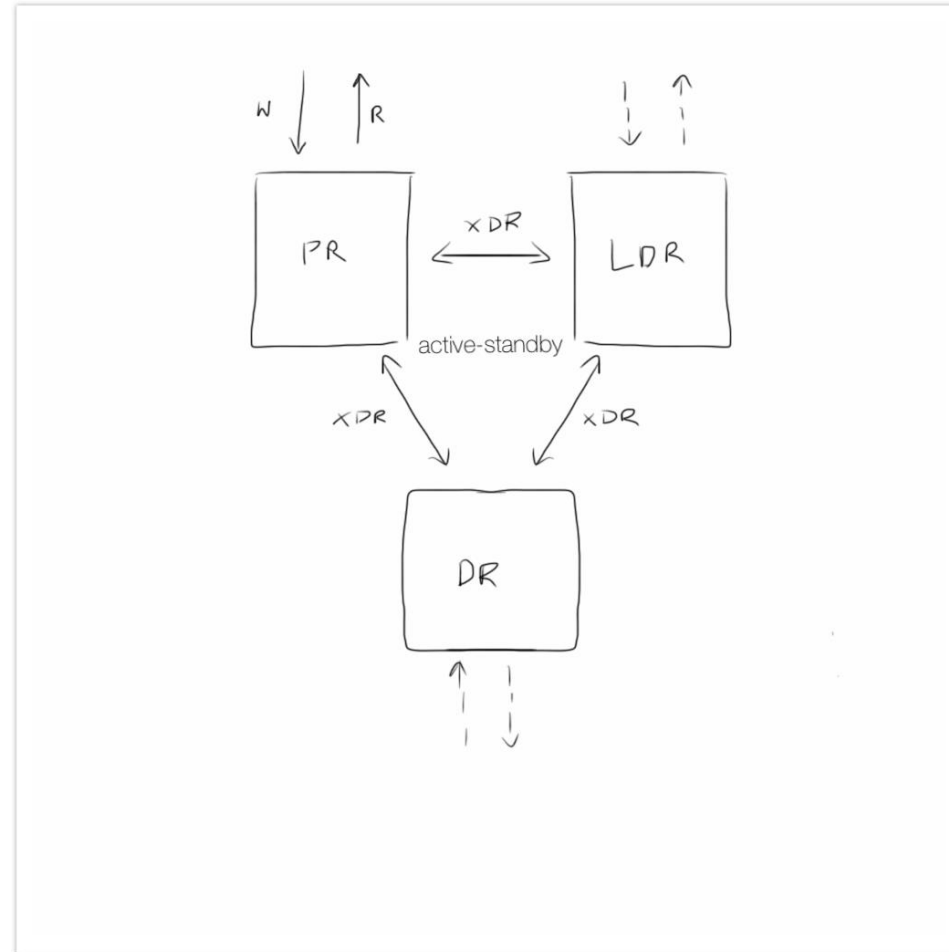


Example: Medium Cluster (2 node failure)



Example: Large Cluster (2 node failure)





Operational Decision Factors

- How to reload data
 - Offline (have original source)
 - Independent data load to each cluster
 - Online
 - Use XDR
 - Warm up new online data
- Having LDR facilitate data movement
 - Need to serve production traffic 24x7





ANSIBLE

Aerospike Database Management API(s)



Automation using Ansible Engine

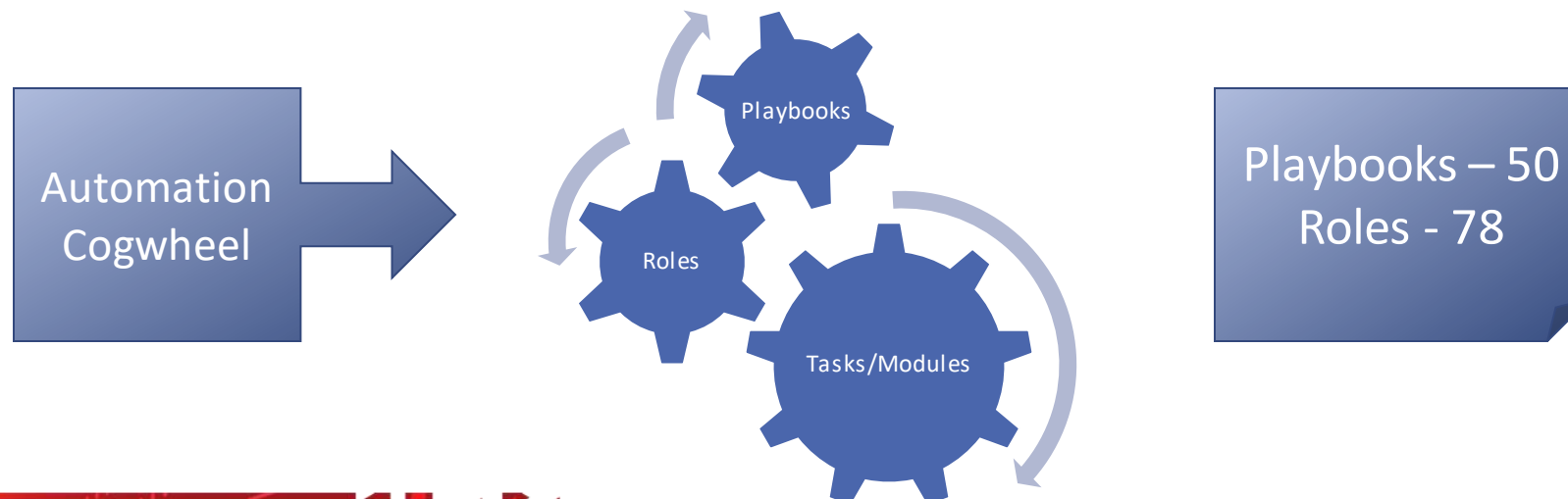
- ❑ Automate OS Patching and DB Upgrades for entire Aerospike footprint
- ❑ Playbook – readable YAML code to perform set of tasks (Role)
- ❑ Tasks - calls modules(in built-libraries)

Playbooks :

1. Upgrade databases in rolling fashion
2. Apply OS patches in rolling fashion
3. Create new clusters
4. Add new node to cluster

Roles:

1. `aero_stop` : *Stop Aerospike services*
2. `execute_os_patch` : *Apply OS patch scripts*
3. `reboot_os` : *Reboot the server*
4. `aero_start` : *Start Aerospike services post patching*



Database API(s)

Playbooks

add_node.yaml
backup.yaml
change_paxos_protocol.yaml
new_cluster.yaml
turn_off_clear_port.yaml
remove_node.yaml
reset_pwd.yaml
prepare_new_node.yaml →
software_upgrade.yaml
speed_up_migrations.yaml
switch_paxos_protocol.yaml
apply_os_patch_single_node.yaml
validate_cluster.yaml
rotate_certificate.yaml
wipe_out_server.yaml

Roles

aero_stop
prepare_os
install_aero
configure_log_rotate
configure_astools
configure_systemd
generate_config_file →
change_ownerships
install_tls
aero_start_port_verify
change_password
grant_sys_admin_privilege
create_xdr_user
create_app_user

Tasks

```
1 - name: back up aerospike.conf
2   shell: "cd /etc/aerospike;filename=aerospike.conf;
3         filestamp=$(date +%Y-%m-%d);
4         newfilename=$filename.$filestamp;
5         cp aerospike.conf $newfilename"
6
7   run_once: true
8   register: backup_result
9
10 - name: prepare aerospike.conf for file system
11   template:
12     src: ./templates/aerospike.conf.j2
13     dest: /etc/aerospike/aerospike.conf
14     owner: aerospike
15     group: aerospike
16     mode: u=rw,g=r,o=r
17     backup: yes
18   tags:
19     - configuration
```

Dynamic Generation of Configuration Files via Jinja2 Templates

Host Config file

#XDR related configurations

```
#-----  
enable_xdr=true
```

#Remote XDR 1

```
#-----  
enable_xdr_1_remote=true  
xdr_data_center_name=DC_1  
aero_dc_xdr_hosts=xdr.b.c.1,x  
dr.b.c.2,xdr.b.c.3
```

#Remote XDR 2

```
#-----  
enable_xdr_1_remote=false  
xdr_data_center_name=DC_2  
aero_dc_xdr_hosts=
```

- Passed variable enable_xdr=true
- Jinja2 template reads the value and generates corresponding xdr configuration

Jinja2 template

```
{% if enable_xdr == 'true' %}  
  
xdr {  
    enable-xdr {{enable_xdr}}                # Globally enable/disable XDR on local node.  
    xdr-digestlog-path /x/aerospike/xdr/digestlog {{xdr_digestlog_size}} # Track digests to be shipped.  
    xdr-max-ship-bandwidth {{max_xdr_bandwidth_bytes}} # Limit the n/w bandwidth to 62MB/sec=0.5Gbps  
    xdr-ship-bins true                        # Ships only modified bins  
  
    {% if enable_xdr_1_remote == 'true' and enable_xdr == 'true' %}  
        datacenter {{xdr_data_center_name}} {  
            {% set aero_dc_xdr_hosts_lst = aero_dc_xdr_hosts.split(',') %}  
            {% for dc_xdr_host in aero_dc_xdr_hosts_lst %}  
                tls-node {{ dc_xdr_host }} {{ tls_name }} 4333  
            {% endfor %}  
            tls-name {{tls_name}}  
            dc-security-config-file /x/aerospike/security/xdr-credentials  
        }  
    {% endif %}  
}
```

Snippet of aerospike.conf generated via Jinja2 template

```
xdr {  
    enable-xdr true                # Globally enable/disable XDR on local node.  
    xdr-digestlog-path /x/aerospike/xdr/digestlog 20G # Track digests to be shipped.  
    xdr-max-ship-bandwidth 62000000 # Limit the n/w bandwidth to 62MB/sec=0.5Gbps  
    xdr-ship-bins true            # Ships only modified bins  
    datacenter DC_1 {  
        tls-node xdr.b.c.1 4333  
        tls-node xdr.b.c.2 4333  
        tls-node xdr.b.c.3 4333  
        tls-name aerospike.server.xxxx.xxx.paypal.com  
        dc-security-config-file /x/aerospike/security/xdr-credentials  
    }  
}
```

Variable Grouping via Ansible vars

```
[all:vars]
tls_profile=devqa
flavor=devqa_standard_3
region=gcp
build=4.8.0.5
config=gcp
user_profile=deeptsharma
#####
```

4.5.3.3
4.8.0.5

- Passed variable build=4.8.0.5
- It in turn calls file named 4.8.0.5
- 4.8.0.5 contains all the path to Aerospike binaries for this version.

Only 16 variables
creates a cluster

aero_rpms_folder: aerospike-server-enterprise-4.8.0.5-el7

aero_tools: /tmp/aerospike-server-enterprise-4.8.0.5-el7/aerospike-tools-3.23.0-1.el7.x86_64.rpm

aero_server: /tmp/aerospike-server-enterprise-4.8.0.5-el7/aerospike-server-enterprise-4.8.0.5-1.el7.x86_64.rpm

	Variables	Description
1	tls_profile	TLS definition
2	flavor	Picks the storage pattern: -- file system vs. block device
3	region	Specific to Data Centers
4	build	Aerospike binaries
5	config	Generic configuration for host files: default/cloud
6	user_profile	User specific information
7	aerospike-cluster	Server details
8	cluster_size	Number of nodes in cluster
9	paxos_single_replica_limit	Min number of nodes in the cluster below which replica records will not be stored.
10	cluster_group_name	e.g. my-cluster
11	zone	us-central1-f
12	cluster_name	Combination of cluster_group_name-zone
13	metadata_namespace_name_1	Name for metadata namespace
14	storage_namespace_name_1	Name for storage namespace
15	aero_mesh_seed_addresses	Comma separated node details
16	enable_xdr	True/false

Supported features and operating limits

Features/Invocation1	Min	Max	Default
Cluster	1	1	1
Namespaces	1	4	1
XDR Remotes	0	3	0
Replication factor	1	3	1
Rack awareness	1	3	None
Encryption at rest	False	True	False
TLS – intra cluster	NA	NA	Enabled
TLS - xdr	NA	NA	Enabled
Namespace storage type	NA	NA	file or block
Index type	NA	NA	shmem or pmem

Simplistic Interface

Aerospike-Automation

Database Lifecycle Test

```
ansible-playbook database_lifecycle_test -i ./hosts/  aero23_ccg01
```

Validating Cluster:

```
ansible-playbook validate_cluster.yaml -i ./hosts/  aero23_ccg01
```

Rolling Software Upgrade:

```
ansible-playbook database_software_upgrade.yaml -i ./hosts/  aero23_ccg01
```

🔗 Rolling OS Patching:

```
ansible-playbook os_patching_database_rolling.yaml -i ./hosts/  aero23_ccg01
```

Create New Cluster:

```
ansible-playbook create_cluster.yaml -i ./hosts/  aero23_ccg01
```

Wipe out a server:

```
ansible-playbook wipe_out_server.yaml -i ./hosts/  aero23_ccg01
```



Automation For Aerospike : RESULTS ACHIEVED !!

Automation, 3 months

- OS patching completed for ~2000 servers in just < 3 months
- Zero Human intervention for checking vs manual constant monitoring.
- OS patching via single click vs server by server with manual effort.
- Continuous 24x7 patching by scheduling Ansible playbook overnight.
- Increased team efficiency and productivity to focus on other tasks at hand

Manual, 6-9 months, may be... 🤔

Planning and execution may take >6 months to complete patching.

Bigger team size



More work



Monitoring @ Scale



Monitoring using Manticore

Why do we need monitoring

- 2000 and counting
- 106+ clusters
- Multiple teams using Aerospike asking for same data
- Automating management of such a huge fleet



What Manticore does

- Reports all critical statistics
- Manages custom fail over strategies
- Informs about the availability
- Memory, disk, client latencies, server latencies, migrations, connections etc



How Manticore works

Manticore Agents

- Agents per cluster
- Initializes XDR reader and writer
- Checks for availability and fails over if primary
- When Dashboard requests data –
 - Connects to all the nodes in a cluster using ASInfo queries
 - Fetches node stats, namespace stats, latency stats, XDR stats, storage stats



Manticore Dashboard

- Creates a HTML dashboard using data from the agents



Manticore Reporters

- Sends the HTML data in email format to specified DLs periodically



Reporting @ Scale with Manticore

Metrics	Cluster 1
Time	5-03-2020 10:17:06 PM PDT
SystemReadyCount	6
Nodes[(build)=count;]	6 [(4.5.3.3)=6;]
UsedKeys	193,698,699
UsableKeys(Estimated)	4,872,103,526
MemoryType	shmem
UsedMemory	23.09GB
UsableMemory	580.8GB
FreeMemory%	96%
UsedDisk	0.17TB
UsableDisk	11.66TB
FreeDisk%	98%
ReadTPS	480
WriteTPS	221
XdrTPS	10
XdrLag	0
Connections	6,330

System Report

NameSpace	SetInUse	Keys	UsedDisk(Approx.)	ObjectSize(KB)/Objects%
cluster1-nm1	set1	1,047	0.04GB	8(KB)=1.0%;4(KB)=6.0%;1(KB)=21.0%;2(KB)=50.0%;3(KB)=9.0%;5(KB)=4.0%;6(KB)=2.0%;7(KB)=2.0%;9(KB)=2.0%;10(KB)=1.0%;13(KB)=1.0%;
cluster1-nm1	set2	103	0GB	1(KB)=100.0%;
cluster1-nm1	set3	218	0.05GB	1(KB)=61.0%;2(KB)=12.0%;3(KB)=2.0%;63(KB)=20.0%;
cluster1-nm1	set4	7	0.04GB	1(KB)=14.0%;252(KB)=14.0%;501(KB)=71.0%;
cluster1-nm1	set5	862	0.04GB	8(KB)=3.0%;4(KB)=7.0%;2(KB)=58.0%;3(KB)=14.0%;5(KB)=4.0%;6(KB)=3.0%;7(KB)=2.0%;9(KB)=1.0%;10(KB)=1.0%;11(KB)=1.0%;12(KB)=1.0%;13(KB)=1.0%;252(KB)=14.0%;15(KB)=1.0%;
cluster1-nm1	set6	241	0.05GB	1(KB)=50.0%;5(KB)=25.0%;37(KB)=1.0%;47(KB)=24.0%;
cluster1-nm1	set7	1	0GB	8(KB)=100.0%;

Set Statistics

Unavailable Nodes at 5-04-2020 11:52:37 AM PDT		
ClusterName	NodeIP	NodeName
cluster2	1.1.1.1	abc.ppxyz
Total	1	

Database_Nodes_Report													
Version	DbType	GroupName	ClusterName	NodeName	Connections	FreeRAM	FreeDisk	ReadTPS	WriteTPS	UdTPS	XdrTPS	XdrLag	UsedMemory
4.5.3.3	aerospike	MainCluster	cluster1	abc1.ppxyz	23946	61	61	161	458	0	0	0	93.55GB
4.5.3.3	aerospike	MainCluster	cluster3	abc2.ppxyz	26047	62	62	214	399	0	0	0	88.91GB

Aerospike Capacity Report													
DbType	GroupName	ClusterName	Nodes	FreeMemory	FreeDisk	UsedKeys	UsableKeys	RemainingKeys	UsedMemory	UsableMemory	UsedDisk	UsableDisk	Time
aerospike	MainCluster1	cluster1	20	30%	76%	15,268,240	21,924,465	6,656,217	820.41GB	613.6GB	12.44TB	52.49TB	5-04-2020 10:02:41 AM PDT
aerospike	MainCluster1	cluster2	19	27%	74%	14,959,297	20,706,439	5,747,142	783.57GB	468.4GB	12.61TB	49.57TB	5-04-2020 09:59:10 AM PDT
aerospike	MainCluster1	cluster3	20	18%	73%	17,760,837	21,924,465	4,163,628	117.4GB	613.6GB	13.83TB	52.49TB	5-04-2020 10:03:23 AM PDT
aerospike	MainCluster2	cluster1	20	66%	6%	7,369,106	21,924,465	14,555,359	78.47GB	613.6GB	48.91TB	52.49TB	5-04-2020 09:58:41 AM PDT
aerospike	MainCluster2	cluster2	20	40%	3%	12,970,088	21,924,465	8,954,377	546.16GB	613.6GB	50.81TB	52.49TB	5-04-2020 09:58:28 AM PDT
aerospike	MainCluster2	cluster3	20	65%	8%	7,643,936	21,924,465	14,280,529	11.23GB	613.6GB	49.76TB	52.49TB	5-04-2020 09:58:34 AM PDT

Capacity Report

Inventory Report

Manticore

Manticore Agents

Graphs
And
Alerting

System Report - Overall Stats

- System ready count – If zero, cluster is no more primary or taking live traffic
- Total Nodes and build
- Cluster level stats
 - Total keys (excluding replica)
 - Total used memory, usable memory, free memory %
 - Total used disk, usable disk, free disk %
 - Server side latencies
 - Max client connections

Metrics	Cluster 1
Time	5-03-2020 10:17:06 PM PDT
SystemReadyCount	6
Nodes[(build)=count;]	6 [(4.5.3.3)=6;]
UsedKeys	193,698,699
UsableKeys(Estimated)	4,872,103,526
MemoryType	shmem
UsedMemory	23.09GB
UsableMemory	580.8GB
FreeMemory%	96%
UsedDisk	0.17TB
UsableDisk	11.66TB
FreeDisk%	98%
ReadTPS	480
WriteTPS	221
XdrTPS	10
XdrLag	0
Connections	6,330

System Report - XDR Stats

- Total 50 records of 1-2KB length are
 - written to peer clusters
 - read from peer clusters
 - Latencies are calculated by storing the timestamp of writing a record and read timestamp
 - Min, max, avg, 95th and 99th percentile
- Total 100 records are added to cluster1 by its peers and similar metrics are captured

Percentile	Latency @
Min	327
Average	10,911
95th	26,455
99th	34,188
Max	34,188

Percentile	Latency @
Min	601
Average	10,867
95th	25,379
99th	78,756
Max	78,756

System Report – Set Statistics (~ approximate)

- When disk is full, we don't know which set is eating up storage the most
- How to proceed to truncating which set
- For this, set storage stats have been very helpful
- We use histogram from only one node and multiply it with total nodes count

NameSpace	SetInUse	Keys	UsedDisk(Approx.)	ObjectSize(KB)/Objects%
cluster1-nm1	set1	1,047	0.04GB	8(KB)=1.0%;4(KB)=6.0%;1(KB)=21.0%;2(KB)=50.0%;3(KB)=9.0%;5(KB)=4.0%;6(KB)=2.0%;7(KB)=2.0%;9(KB)=2.0%;10(KB)=1.0%;13(KB)=1.0%;
cluster1-nm1	set2	103	0GB	1(KB)=100.0%;
cluster1-nm1	set3	218	0.05GB	1(KB)=61.0%;2(KB)=12.0%;3(KB)=2.0%;63(KB)=20.0%;
cluster1-nm1	set4	7	0.04GB	1(KB)=14.0%;252(KB)=14.0%;501(KB)=71.0%;
cluster1-nm1	set5	862	0.04GB	8(KB)=3.0%;4(KB)=7.0%;2(KB)=58.0%;3(KB)=14.0%;5(KB)=4.0%;6(KB)=3.0%;7(KB)=2.0%;9(KB)=1.0%;10(KB)=1.0%;11(KB)=1.0%;12(KB)=1.0%;13(KB)=1.0%;15(KB)=1.0%;
cluster1-nm1	set6	241	0.05GB	1(KB)=50.0%;5(KB)=25.0%;37(KB)=1.0%;47(KB)=24.0%;
cluster1-nm1	set7	1	0GB	8(KB)=100.0%;

Inventory Report

Unavailable Nodes at 5-04-2020 11:52:37 AM PDT

ClusterName	NodeIP	NodeName
cluster2	1.1.1.1	abc.pqr.xyz
Total	1	

Database_Nodes_Report

Version	DbType	GroupName	ClusterName	NodeName	Connections	FreeRAM	FreeDisk	ReadTPS	WriteTPS	UdfTPS	XdrTPS	XdrLag	UsedMemory	UsedDisk	LastRestarted
4.5.3.3	aerospike	MainCluster	cluster1	abc1.pqr.xyz	23946	61	61	161	458	0	0	0	93.55GB	655.89GB	Mon Apr 06 17:25:48 PDT 2020
4.5.3.3	aerospike	MainCluster	cluster3	abc2.pqr.xyz	26047	62	62	214	399	0	0	0	88.91GB	622.55GB	Mon Apr 06 17:30:43 PDT 2020

- It shows which all nodes are down in all the cluster groups
- AS version, connections, memory, disk, latency stats per node based, last restarted time
- Shows data for 2000 hosts / 106+ clusters in one report

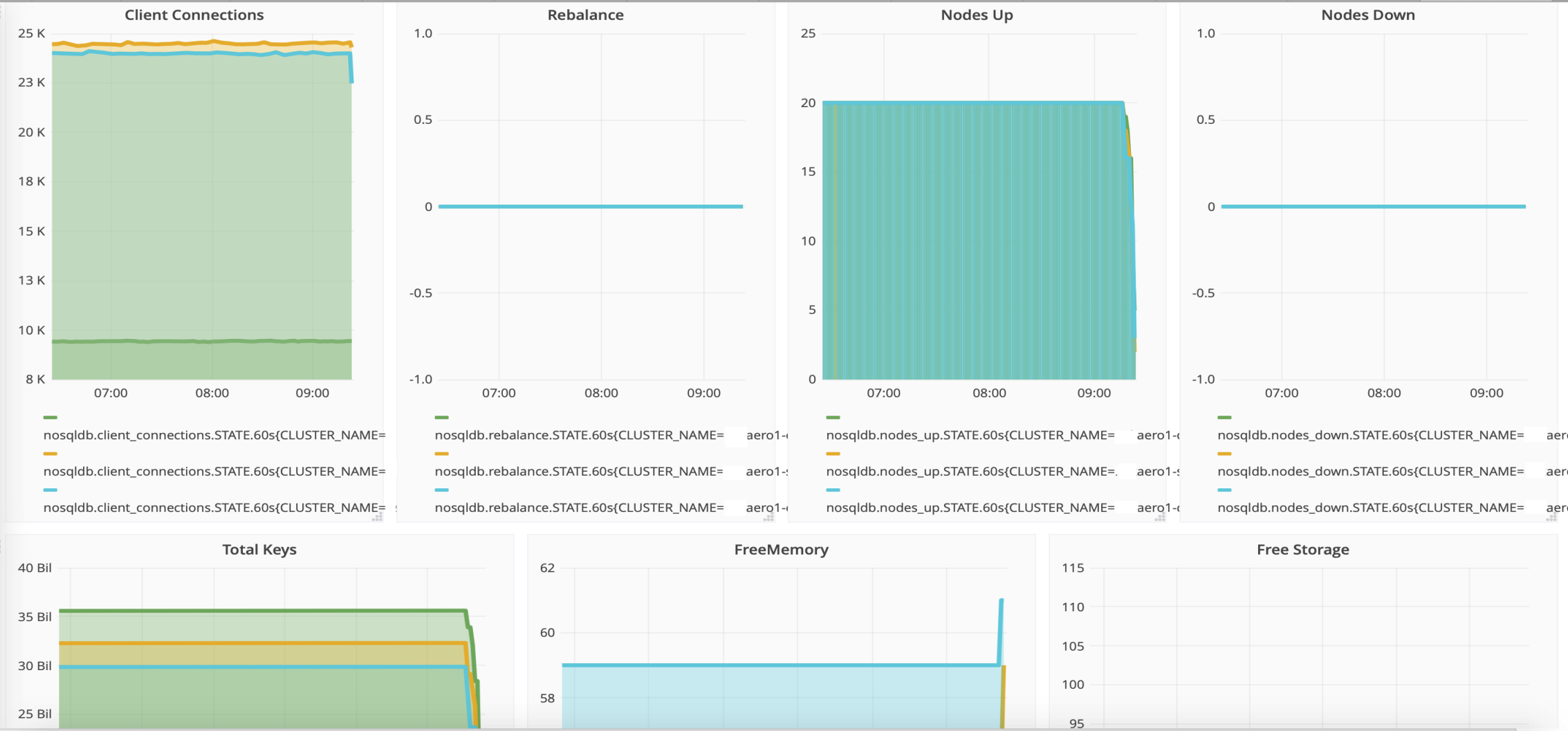
Capacity Report

Aerospike Capacity Report

DBType	GroupName	ClusterName	Nodes	FreeMemory	FreeDisk	UsedKeys	UsableKeys	RemainingKeys	UsedMemory	UsableMemory	UsedDisk	UsableDisk	Time
aerospike	MainCluster1	cluster1	20	30%	76%	15,268,248	21,924,465	6,656,217	820.41GB	613.6GB	12.44TB	52.49TB	5-04-2020 10:02:41 AM PDT
aerospike	MainCluster1	cluster2	19	27%	74%	14,959,297	20,706,439	5,747,142	783.57GB	468.4GB	12.61TB	49.57TB	5-04-2020 09:59:10 AM PDT
aerospike	MainCluster1	cluster3	20	18%	73%	17,760,837	21,924,465	4,163,628	117.4GB	613.6GB	13.83TB	52.49TB	5-04-2020 10:03:23 AM PDT
aerospike	MainCluster2	cluster4	20	66%	6%	7,369,106	21,924,465	14,555,359	78.47GB	613.6GB	48.91TB	52.49TB	5-04-2020 09:58:41 AM PDT
aerospike	MainCluster2	cluster5	20	40%	3%	12,970,088	21,924,465	8,954,377	546.16GB	613.6GB	50.81TB	52.49TB	5-04-2020 09:58:28 AM PDT
aerospike	MainCluster2	cluster6	20	65%	5%	7,643,936	21,924,465	14,280,529	11.23GB	613.6GB	49.76TB	52.49TB	5-04-2020 09:58:34 AM PDT

- Color coded memory and disk utilization
- Easy for detection and prompts immediate actions
- Shows capacity stats for all 2000 hosts / 106+ clusters in one report

Graphs Dashboard





Q & A