D085 Performance Assessment

Table of Contents

A. Introduction of Solution

Purpose

This solution was developed to provide Satellite Ads Inc. (SAd_INC) a flexible and scalable platform to deploy additional servers and services as they increase customer count, advertisements, and satellite clusters.

Goals and Objectives

Satellite Ads Inc. required a scalable solution that would allow them to monetize space advertisement using their second generation of microsatellites. In order to do so they need a cloud-based automation system that will allow them to handle the expected loads during advertising window throughout the year as more customers are onboarded.

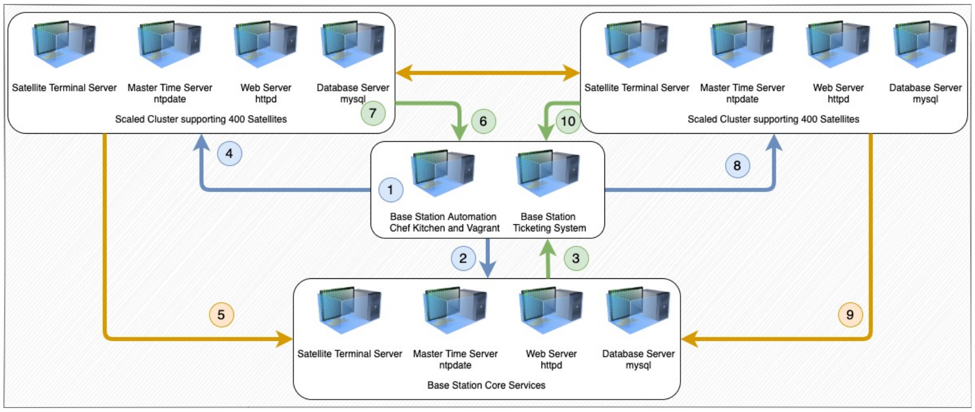
Scope

The current customer retention is projected to increase from 250 to 14,173 from March of this current year to March of the following year. With the increase of ads projected to hit 141,735 a month by March the scalable automated system will need to scale up to an additional 59 clusters that communicates with the cloud-based base station master cluster. Each cluster is projected to handle a max of 400 satellites for a total of 23,600 satellites in orbit and providing ad services by March. As the satellites will be geographically located to a region, when those regions enter daytime hours the cluster will scale down and minimum services will be provided to ensure communication is still present without over utilizing cloud resources and increasing expenses.

Functionality

Before the launch of the first set of second-generation satellites prepare to launch the initial cloud-based base station cluster will come online. Chef, an infrastructure automation tool, will deploy the scaled clusters to both on-premises and cloud providers to ensure an outage from a service provider does not cause the servers to go down as time is an important variable to the success in the communication of each cluster and satellite. With a threshold of 300 satellites another scaled cluster will come online and mark itself as ready. The time server will sync to the closest time server and the other servers in the cluster will communicate with it to ensure latency is kept to a minimal amount. As each satellite comes online it will sync with the nearest time server and receive updates and advertisement workflows from the closest web server. As clusters are scaled up and down depending on workloads a message is sent via the queuing service to the helpdesk ticketing system where an email is sent to helpdesk@satelliteads.com. If there is an error or a time becomes out of sync on any server or satellite a message is also sent.

B. Visual Representation



1 Base Station Automation system running on-premises Chef and Vagrant will deploy the clusters automatically as demand increases. The Base Station Ticketing system will also be created to ensure queuing services are routed properly.

The kitchen.yml file will be populated with all required services and names. It will deploy the first server cluster, Base Station Core Services, using the following commands:

2
- kitchen create # This will create the specified VMs on the kitchen.yml file.
- kitchen converge # This will install the required services on each VM according to kitchen.yml.

The following servers will be created:

- Satellite Terminal Server
- Master Time Server
- Web Server
- Database Server

3 A message is added to the queuing servers to let the Base Station Automation server and the ticketing system know it is online and ready.

The Base Station Automation Server will then deploy the first cluster in charge of the initial cluster of satellites.

4 The following servers will be created:

- Scaled Satellite Terminal Server
- Scaled Master Time Server
- Scaled Web Server
- Scaled Database Server

5 NTP is synced with the Base station and data is replicated to each server to ensure low latency to satellites.

6 A message is added to the queuing servers to let the Base Station Automation server and the ticketing system know it is online and ready.

7 When 300 Satellites come online a message is added to the queuing service to inform the Base Station Automation server to spin up another cluster.

The Base Station Automation Server will then deploy the another cluster in charge of the next cluster of satellites.

8 The following servers will be created:

- Scaled Satellite Terminal Server
- Scaled Master Time Server
- Scaled Web Server
- Scaled Database Server

9 NTP is synced with the Base station or closest NTP server and data is replicated to each server to ensure low latency to satellites.

10 A message is added to the queuing servers to let the Base Station Automation server and the ticketing system know it is online and ready.

C. Automation Script

Kitchen.yml

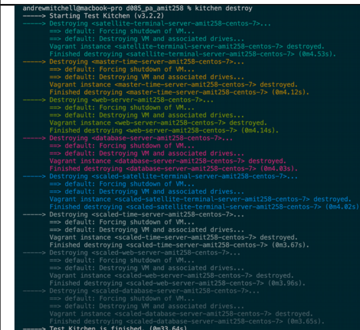
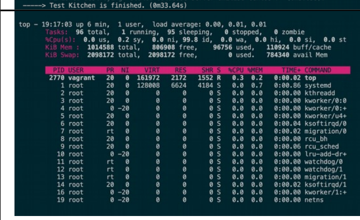
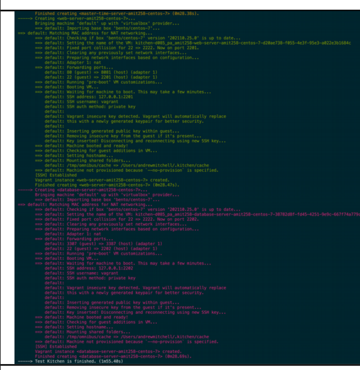
```
# Created by.
driver:
  name: vagrant # I used vagrant to deploy locally for testing.
  network:
  #  NAT enabled, If external ports are needed they are set at individual VMs since I am not using AWS/Azure.
  customize:
    memory:  1024  # sets the memory for default VMs to 1024 MB
    cpus: 2   # set the cpu for default VMs to 2 vCPUs
```

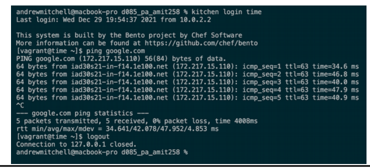
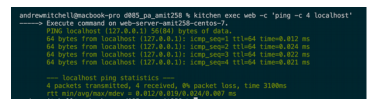
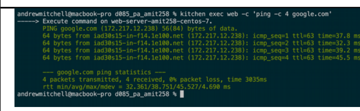
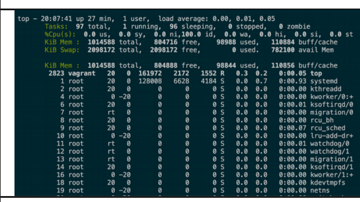
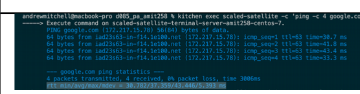
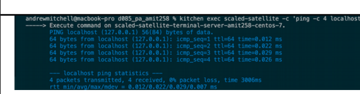
```
provisioner:
  name: chef_zero
  product_name: chef
  product_version: 14.12.9
# Runs automated tests.
verifier:
  name: inspec
# CentOS 7 has been selected.
platforms:
  - name: centos-7
suites:
  - name: satellite_terminal_server_amit258
    driver:
      vm_hostname: terminal.satelliteads.com
    run_list:
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: master_time_server_amit258
    driver:
      vm_hostname: time.satelliteads.com
      network:
      - ["forwarded_port", {guest: 123, host: 8123}]
    run_list:
      - recipe[d085_pa_amit258::ntp_service]
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: web_server_amit258
    driver:
      vm_hostname: web-server.satelliteads.com
      network:
      - ["forwarded_port", {guest: 80, host: 8081}]
    run_list:
      - recipe[d085_pa_amit258::web_front]
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: database_server_amit258
    driver:
      vm_hostname: db.satelliteads.com
      network:
      - ["forwarded_port", {guest: 3307, host: 3387}]
    run_list:
      - recipe[d085_pa_amit258::mysql_service]
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: scaled_satellite_terminal_server_amit258
    driver:
      vm_hostname: terminal.scaled.satelliteads.com
      customize:
```

```
        memory: 512
        cpus: 1
    run_list:
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: scaled_time_server_amit258
    driver:
      vm_hostname: time.scaled.satelliteads.com
      customize:
        memory: 512
        cpus: 1
      network:
      - ["forwarded_port", {guest: 123, host: 8124}]
    run_list:
      - recipe[d085_pa_amit258::ntp_service]
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: scaled_web_server_amit258
    driver:
      vm_hostname: web-server.scaled.satelliteads.com
      customize:
        memory: 512
        cpus: 1
      network:
      - ["forwarded_port", {guest: 80, host: 8082}]
    run_list:
      - recipe[d085_pa_amit258::web_front]
      - recipe[d085_pa_amit258::default]
    attributes:
  - name: scaled_database_server_amit258
    driver:
      vm_hostname: db.scaled.satelliteads.com
      customize:
        memory: 512
        cpus: 1
      network:
      - ["forwarded_port", {guest: 3307, host: 3388}]
    run_list:
      - recipe[d085_pa_amit258::mysql_service]
      - recipe[d085_pa_amit258::default]
    attributes:
```

D. Diagnostic Report

Data Description	Optimal Range	Data and Results	Automation Script Used to	Screenshot of Result of Script

			Extract Data (text only)	
Time to scale from 1 cluster to 200 clusters (60,000 advertisements expected at peak global usage) based on 300 satellites per cluster (subject to change based on load testing)	15–30 minutes for each cluster	1 x 8 node cluster takes **4m11sec**. **200** x 4m11sec = 836.67m = **13hr56mins**	kitchen create	See line 280 of Part D 
Time to register a cluster and then quench connections to the load balancer, taking the cluster off-line (start-up, operation, shutdown)	1 minute per connection quench, start of cluster launch, and part of time to scale cluster, can be tracked separately as a quench	1 x 8 node cluster takes **33secs.**	Kitchen destroy	
Peak load averages per system at 200, and 300, satellites per cluster	60% of CPU triggers new cluster launch; if reaching core load at 200 satellites, launch new cluster on 60% CPU loads	Load average: 0.00, 0.01, 0.01	Kitchen exec database -c top	
Write times to the diagnostic data drive	<30 milliseconds	1073741824 bytes (1.1 GB) copied, 6.56613 s, 164 MB/s	kitchen exec web -c 'dd if=/dev/zero of=testWriteSpeed.txt bs=1G count=1'	
Pull time from the game instances (1 Satellite Terminal Server, 1 Web Server, 1 Database, and 1 time server) and initialization time	Part of cluster launch 15–30 minutes	1x 4 node cluster takes **1m55sec**.	Kitchen create	
*Average messaging service (queue) time	<1 minute in queue	{ "userId": 48593, "title": "messaging service test", "body": "Testing	Kitchen login satellite time curl -X POST -H "Content-Type: application/json	

		messaging service queue time.", "id": 101 } real 0m0.995s user 0m0.058s sys 0m0.056s	" -d '{"userId": 48593, "title": "messaging service test", "body": "Testing messaging service queue time."}' https://jsonplaceholder.typicode.com/posts	
Average latency for the Time server	<30 milliseconds	rtt min/avg/max/mdev = 34.641/42.078/47.952/4.853 ms	Kitchen login time Ping google.com	
Average latency of each cluster	<30 milliseconds	rtt min/avg/max/mdev = 0.012/0.019/0.024/0.007 ms	Kitchen exec web -c ‘ping -c 4 localhost’	
Network data in and out for each cluster	<1 second	rtt min/avg/max/mdev = 32.361/38.751/45.527/4.690 ms	Kitchen exec web -c ‘ping -c 4 google.com’	
Overall CPU utilization of the environment for each cluster	Not >60%	load average: 0.00, 0.01, 0.05	kitchen exec web -c 'top'	
*Diagnostic data able to be written by the automation to the correct cloud bucket storage space	Show read/write times <1 second	n/a	n/a	n/a
Scaled Satellite Cluster latency	<30 milliseconds	rtt min/avg/max/mdev = 30.782/37.359/43.446/5.393 ms	kitchen exec scaled-satellite -c 'ping -c 4 google.com'	
Scaled Satellite Cluster latency between gateway/scaled clusters and core	<30 milliseconds	rtt min/avg/max/mdev = 0.012/0.022/0.029/0.007 ms	kitchen exec scaled-satellite -c 'ping -c 4 localhost'	

Scaled Satellite Cluster latency between scaled clusters and environment	<30 milliseconds	rtt min/avg/max/mdev = 0.015/0.021/0.026/0.004 ms	kitchen exec scaled-web -c 'ping -c 4 localhost'	
Pull time from the scaled clusters and initialization time	15–30 minutes for each cluster	Test Kitchen is finished. (1m55.28s)	Kitchen create scaled	

E. Web Sources

Web sources used:

- Learn.chef.io - https://learn.chef.io/courses/course-v1:chef+Infra101+perpetual/courseware/
- Docs.chef.io - https://docs.chef.io/workstation/ctl_kitchen/#kitchen-exec
- Docs.chef.io - https://docs.chef.io/workstation/config_yml_kitchen/
- Httpd Recipe - https://supermarket.chef.io/cookbooks/httpd
- Mysql Recipe - https://supermarket.chef.io/cookbooks/mysql
- NTP Recipe - https://supermarket.chef.io/cookbooks/ntp
- Dd command - https://man7.org/linux/man-pages/man1/dd.1.html
- Ping command - https://man7.org/linux/man-pages/man8/ping.8.html
- Top command - https://man7.org/linux/man-pages/man1/top.1.html

F. Sources

No other sources were used outside of web sources provided in section E.