[Satellite Ads Inc.]

Cloud-Based Automation System

[D085 - Automation and Scaling Tools]

[Monica Fyfe]

[ ]-14-2022

[Version 2.0]

WESTERN GOVERNORS UNIVERSITY

CONTENTS

A. INTRODUCTION OF SOLUTION

Purpose

The purpose of this solution is to provide a scalable and highly-available platform for Satellite Ads, Inc.'s (Sad_INC) second wave of advertising satellites by implementing a cloud-based automation system.

Goals and Objectives

The goals and objectives of this solution are to meet the scaling needs presented by SAd_INC to facilitate the deployment of new second generation satellites to broadcast SAd_INC's advertisements. The automation system should be elastic to ensure that the cloud-based back end can handle the advertising loads and be continually monitored. . SAd_INC requires the ability to scale to critical events based on the lessons learned from the initial launch and the projections for updated. To meet these goals the following solution will leverage AWS OpsWorks to create a Chef Automate server, create an AWS Auto Scaling group, and then create a Lamada function for the termination of EC2 instances once an Amazon EventBridge rule is triggered.

Scope

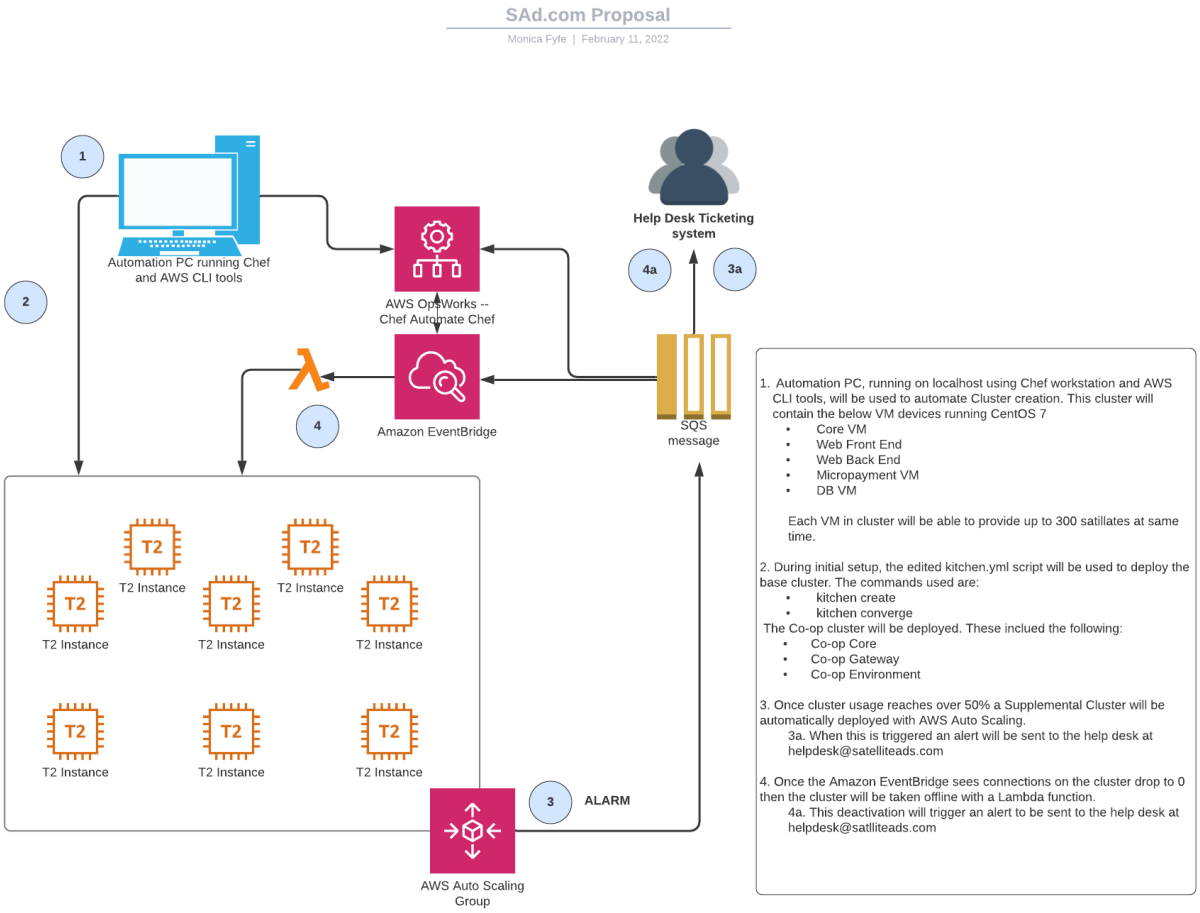
After the initial deployment of the baseline cluster Amazon EventBridge will track the ClusterScale rule and if the satellite usage reaches 300 satellites, then the AWS Auto Scaling group will trigger the deployment of an additional cluster. When this is triggered, it will also generate an AWS SQS message to the help desk ticketing system at helpdesk@satelliteads.com.

Once resources are no longer needed then the AWS EC2 Autoscaling will trigger another alert to run a Lambda function to terminate the unneeded clusters. This will also generate a AWS SQS message to the help desk ticketing system at helpdesk@satelliteads.com.

Functionality

There are few tools being used to scale this solution. The Chef cookbooks will be initially triggered from the Automation PC. Then once the initial cluster is in production the EC2 instances will be enabled as a part of the AWS Auto Scaling Group along with the CloudWatch rules. Alerts will be sent to the help desk for any errors and scaling events that occur for tracking and informational reasons.

B. VISUAL REPRESENTATION



C. AUTOMATION SCRIPT

1. Copy your automation script from part C here (text only):

```
#Monica Fyfe Student ID: #000467874
#default driver used to create cookbook testing instance; will be using AWS
driver:
  name: ec2
  network:
    - ["public_network"]
  customize:
    memory:  512
```

```
#how will Chef run, chef_zero simulates a local server
provisioner:
  name: chef_zero
  product_name: chef
  product_version: 14.12.9
# application used when running automated tests
verifier:
  name: inspec
# OS of VMs
platforms:
  - name: centos-7
#collection of the virtual test environment
suites:
#core services
  - name: sattest1_core
    driver:
      vm_hostname: sattest1.SAdcore.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#front-end web images
  - name: sattest2_webfront
    driver:
      vm_hostname: sattest2.SAdwebfront.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#back-end web images
  - name: sattest3_webback
    driver:
      vm_hostname: sattest3.SAdwebback.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#data services
  - name: sattest4_db
    driver:
```

```
      vm_hostname: sattest4.SAddb.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#micropayment transaction services
  - name: sattest5_mirco
    driver:
      vm_hostname: sattest5.SAdmicropayment.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#CoopPlayCore
  - name: sattest6_playcore
    driver:
      vm_hostname: sattest6.SAdplaycore.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#CoopPlay-gw
  - name: sattest7_playgw
    driver:
      vm_hostname: sattest7.SAdplaygw.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
#CoopPlay-env
  - name: sattest8_playenv
    driver:
      vm_hostname: sattest8.SAdplayenv.com
    run_list:
      - recipe[learn_chef_httpd::default]
    attributes:
```

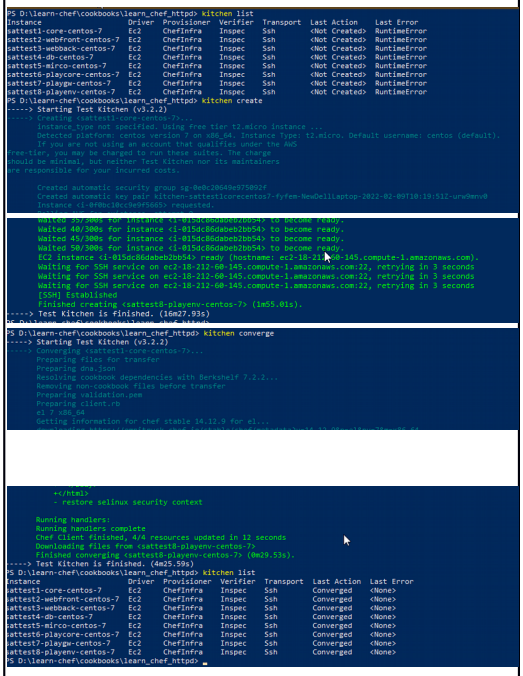
2. Screenshot showing that the automation script executes without errors (from part D):

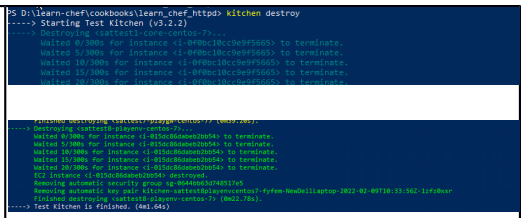
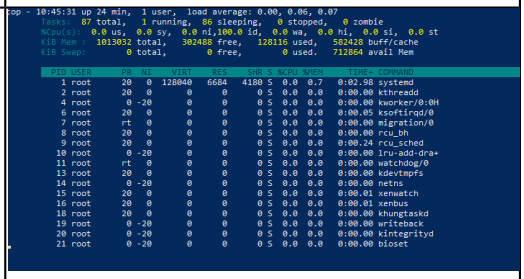
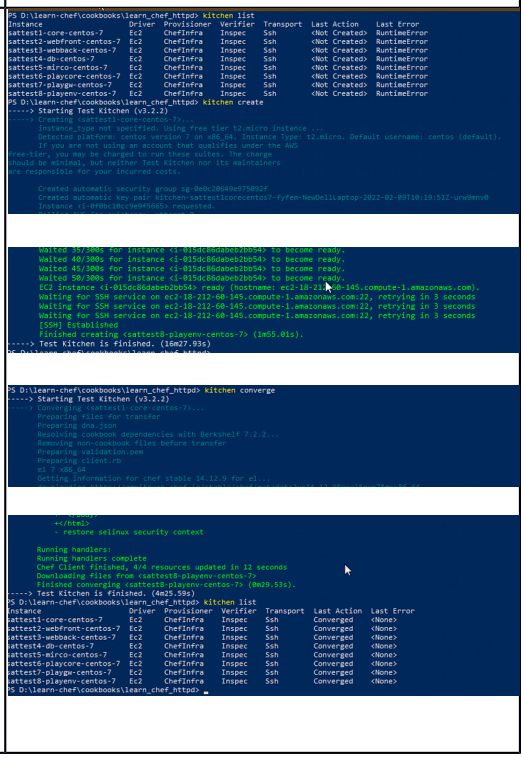
```
PS D:\learn-chef\cookbooks\learn_chef_httpd> kitchen list
Instance                  Driver  Provisioner  Verifier  Transport  Last Action    Last Error
sattest1-core-centos-7    Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest2-webfront-centos-7  Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest3-webback-centos-7   Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest4-db-centos-7        Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest5-mirco-centos-7     Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest6-playcore-centos-7  Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest7-playgw-centos-7    Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
sattest8-playenv-centos-7   Ec2     ChefInfra    Inspec    Ssh        <Not Created>  RuntimeError
PS D:\learn-chef\cookbooks\learn_chef_httpd> kitchen create
-----> Starting Test Kitchen (v3.2.2)
-----> Creating <sattest1-core-centos-7>...
       instance_type not specified. Using free tier t2.micro instance ...
       Detected platform: centos version 7 on x86_64. Instance Type: t2.micro. Default username: centos (default).
       If you are not using an account that qualifies under the AWS
free-tier, you may be charged to run these suites. The charge
should be minimal, but neither Test Kitchen nor its maintainers
are responsible for your incurred costs.

       Created automatic security group sg-0e0c20649e975092f
       Created automatic key pair kitchen-sattest1corecentos7-fyfem-NewDellLaptop-2022-02-09T10:19:51Z-urw9mnv0
       Instance <i-0f0bc10cc9e9f5665> requested.
```

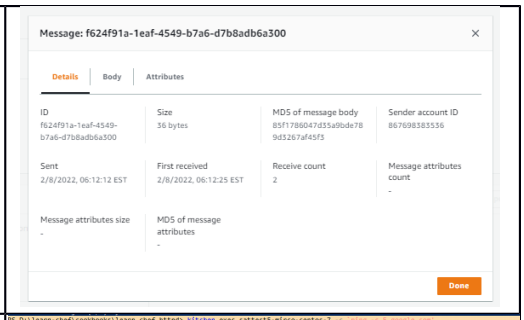
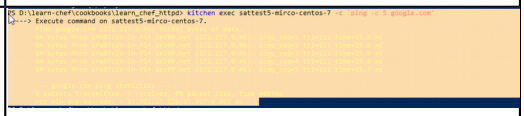
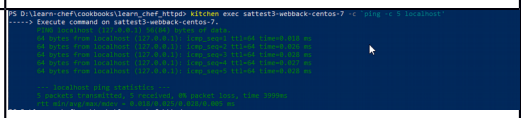
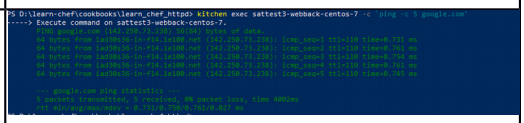
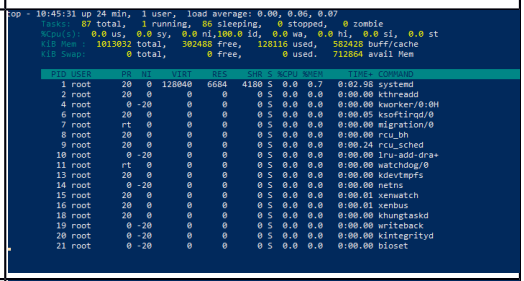
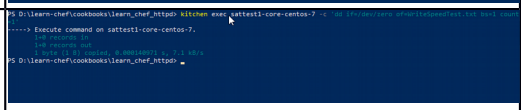
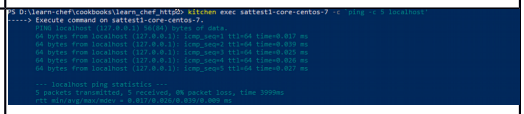
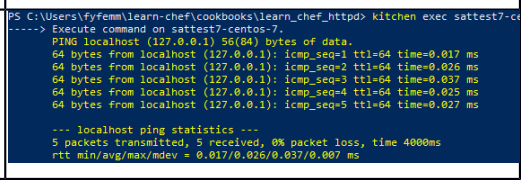
```
       Waited 35/300s for instance <i-015dc86dabeb2bb54> to become ready.
       Waited 40/300s for instance <i-015dc86dabeb2bb54> to become ready.
       Waited 45/300s for instance <i-015dc86dabeb2bb54> to become ready.
       Waited 50/300s for instance <i-015dc86dabeb2bb54> to become ready.
       EC2 instance <i-015dc86dabeb2bb54> ready (hostname: ec2-18-212-60-145.compute-1.amazonaws.com).
       Waiting for SSH service on ec2-18-212-60-145.compute-1.amazonaws.com:22, retrying in 3 seconds
       Waiting for SSH service on ec2-18-212-60-145.compute-1.amazonaws.com:22, retrying in 3 seconds
       Waiting for SSH service on ec2-18-212-60-145.compute-1.amazonaws.com:22, retrying in 3 seconds
       [SSH] Established
       Finished creating <sattest8-playenv-centos-7> (1m55.01s).
-----> Test Kitchen is finished. (16m27.93s)
```

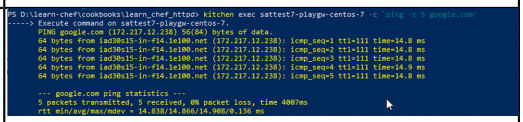
D. DIAGNOSIS REPORT

Data Description	Optimal Range	Data and Results	Automation Script Used to Extract Data (text only)	Screenshot of Result of Script

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	One cluster = 16m.28s
		
		200 clusters = 16m.28s * 200 = 3256 minutes (54 hours)
			`kitchen list`
			`kitchen create`
			`kitchen converge`
				

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	4m2s	`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, 0.06, 0.07	`kitchen exec sattest1-core-centos-7 -c 'top'`	
Write times to the diagnostic data drive	<30 milliseconds	1 byte (1 B) copied, 0.000140971 s, 7.1 kB/s	`kitchen exec sattest1-core-centos-7 -c 'dd if=/dev/zero of=WriteSpeedTest.txt bs=1 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	16 m.28s	`kitchen create` `kitchen converge`	

*Average messaging service (queue) time	<1 minute in queue	12ms	AWS SQS	
Average latency for the Time server	<30 milliseconds	rtt min/avg/max/mdev = 15.781/15.826/15.917/0.092 ms	`kitchen exec sattest5-mirco-centos-7 -c 'ping -c 5 google.com'`	
Average latency of each cluster	<30 milliseconds	rtt min/avg/max/mdev = 0.018/0.025/0.028/0.005 ms	`kitchen exec sattest3-webback-centos-7 -c `ping -c 5 localhost'`	
Network data in and out for each cluster	<1 second	rtt min/avg/max/mdev = 0.731/0.750/0.761/0.027 ms	`kitchen exec sattest3-webback-centos-7 -c `ping -c 5 google.com'`	
Overall CPU utilization of the environment for each cluster	Not >60%	Load average: 0.00, 0.06, 0.07	`kitchen exec sattest1-core-centos-7 -c 'top'`	
*Diagnostic data able to be written by the automation to the correct cloud bucket storage space	Show read/write times <1 second	1 byte (1 B) copied, 0.000140971 s, 7.1 kB/s	`kitchen exec sattest1-core-centos-7 -c 'dd if=/dev/zero of=WriteSpeedTest.txt bs=1 count=1'`	
Scaled Satellite Cluster latency	<30 milliseconds	rtt min/avg/max/mdev = 0.017/0.026/0.039/0.009 ms	`kitchen exec sattest1-core-centos-7 -c `ping -c 5 localhost'`	
Scaled Satellite Cluster latency between gateway/scaled clusters and core	<30 milliseconds	rtt min/avg/max/mdev = 0.017/0.026/0.037/0.007 ms	`kitchen exec sattest7-centos-7 -c 'ping -c 5 localhost'`	

Scaled Satellite Cluster latency between scaled clusters and environment	<30 milliseconds	rtt min/avg/max/mdev = 14.838/14.866/14.908/0.136 ms	`kitchen exec sattest7-playgw-centos-7 -c 'ping -c 5 google.com'`	
Pull time from the scaled clusters and initialization time	15–30 minutes for each cluster	16m.28s	`kitchen create` `kitchen converge`