**Automation Script**

1. Copy your code from part C here :

driver:

      name: vagrant

      network:

        - ["public\_network"]

      customize:

        memory: 512

    provisioner:

      name: chef\_zero

      product\_name: chef

      product\_version: 14.12.9

    verifier:

      name: inspec

    platforms:

      - name: centos-7

    suites:

      - name: cgsp-2-core

        driver:

          vm\_hostname: core.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg2-web-front

        driver:

          vm\_hostname: front.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg2-web-back

        driver:

          vm\_hostname: back.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg-db

        driver:

          vm\_hostname: db.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg-2-micropayment

        driver:

          vm\_hostname: micropayment.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg-2-CoopPlayCore

        driver:

          vm\_hostname: coopCore.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg-2-CoopPlay-gw

        driver:

          vm\_hostname: Coop-gw.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

      - name: cg-2-CoopPlay-env

        driver:

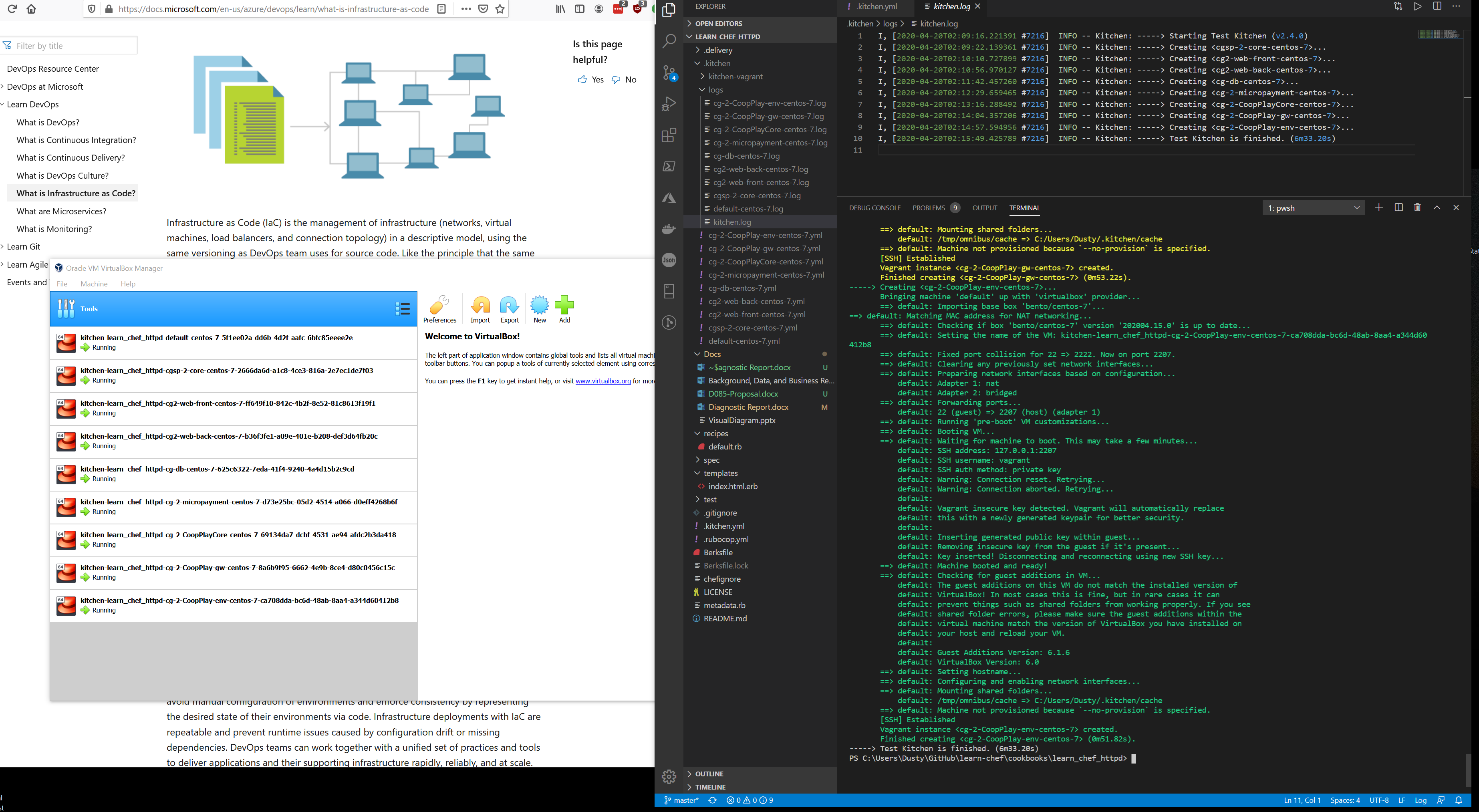
          vm\_hostname: Coop-env.SparkIT-Games.com

        run\_list:

          - recipe[learn\_chef\_httpd::default]

        attributes:

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



**Diagnostic Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Description** | **Optimal Range** | **Data and Results** | **Script Used to Extract Data** | **Screenshot of Result of Script** |
| Time to scale from 1 cluster to 200 clusters  (8 million users expected at peak after PAX West) based on 40K users per cluster (subject to change based on load testing) | 15–30 minutes for each cluster | 1 cluster = 9min11sec  200 clusters = 1836 Min or 30.6 hours | 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 | 1m17s | Kitchen destroy |  |
| Peak load averages per system at 20K, 30K, and 40K users per game cluster | 60% of CPU triggers new cluster launch; if reaching core load at 20K users, launch new cluster on 60% CPU loads | Load average:  0.04, 0.03, 0.05 | kitchen exec cgsp-2-core -c 'top' |  |
| Write times to the diagnostic data drive | <30 milliseconds | 1073741824 bytes (1.1 GB) copied, 6.45524 s, 166 MB/s | kitchen exec cgsp-2-core -c 'dd if=/dev/zero of=testwritespeed.txt bs=1G count=1' |  |
| Pull time from the game instances (1 core, 1 web front end, 1 web back end, 1 database, and 1 micropayment server) and initialization time | Part of cluster launch 15–30 minutes | 1m30.74s | Kitchen create |  |
| Average messaging service (queue) time | <1 minute in queue | N/A | N/A | N/A |
| Average latency for the micropayment server | <30 milliseconds | rtt min/avg/max/mdev = 21.875/26.027/31.081/3.294 ms | kitchen exec cg-2-micropayment -c 'ping www.google.com' |  |
| Average latency of each cluster | <30 milliseconds | rtt min/avg/max/mdev = 0.016/0.026/0.030/0.005 ms | kitchen exec cg-2-micropayment -c 'ping localhost' |  |
| Network data in and out for each cluster | <1 second | rtt min/avg/max/mdev = 19.950/25.331/34.682/5.841 ms | kitchen exec cg-db -c 'ping -c 4 www.google.com' |  |
| Overall CPU utilization of the environment for each cluster | Not >60% | Load average:  0.00, 0.01, 0.05 | kitchen exec cg2-web-front -c 'top' |  |
| Diagnostic data able to be written by the automation to the correct cloud bucket storage space | Show read/write times <1 second | 1073741824 bytes (1.1 GB) copied, 6.73196 s, 159 MB/s | kitchen exec cg-2-CoopPlayCore -c 'dd if=/dev/zero of=testwritespeed.txt bs=1G count=1' |  |
| Cooperative play cluster latency | <30 milliseconds | rtt min/avg/max/mdev = 0.016/0.028/0.034/0.007 ms | kitchen exec cg-2-CoopPlay-gw -c 'ping -c 4 localhost' |  |
| Cooperative play latency between gateway/matching and core | <30 milliseconds | rtt min/avg/max/mdev = 17.479/19.634/21.559/1.543 ms | kitchen exec cg-2-CoopPlayCore -c 'ping -c 4 www.wgu.edu' |  |
| Cooperative play latency between gateway/matching and environment | <30 milliseconds | rtt min/avg/max/mdev = 17.359/20.833/22.619/2.114 ms | kitchen exec cg-2-CoopPlay-env -c 'ping -c 4 www.wgu.edu' |  |
| Pull time from the cooperative play instances and initialization time | 15–30 minutes for each cluster | 1m30.74s | Kitchen create |  |