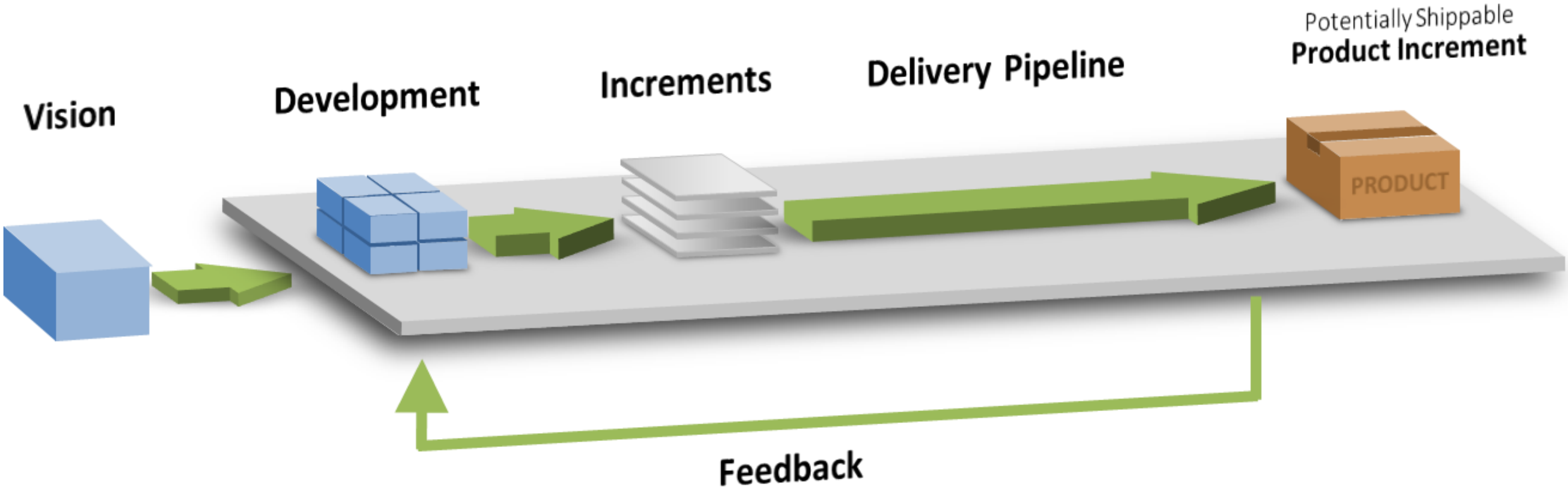
Leveraging Test Automation to Verify Infrastructure

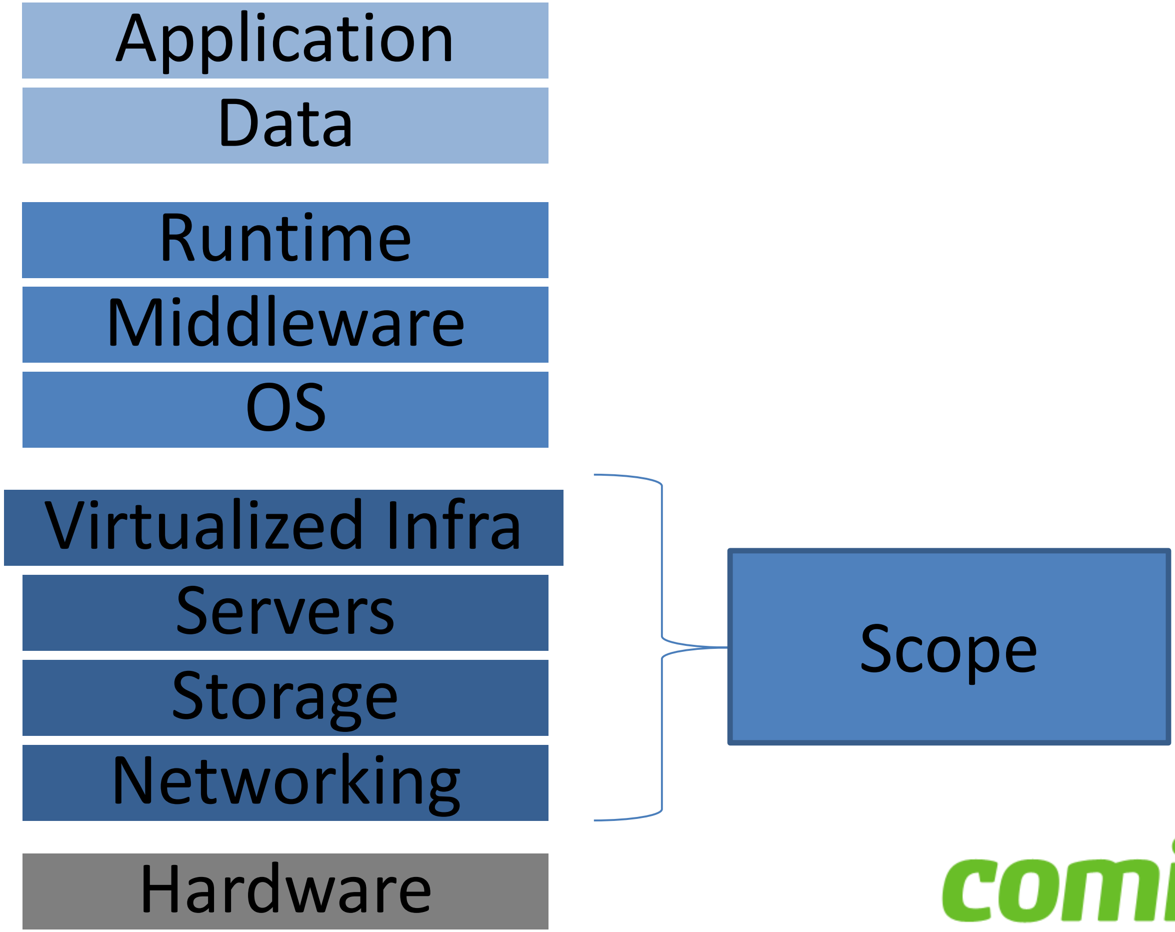


comiq

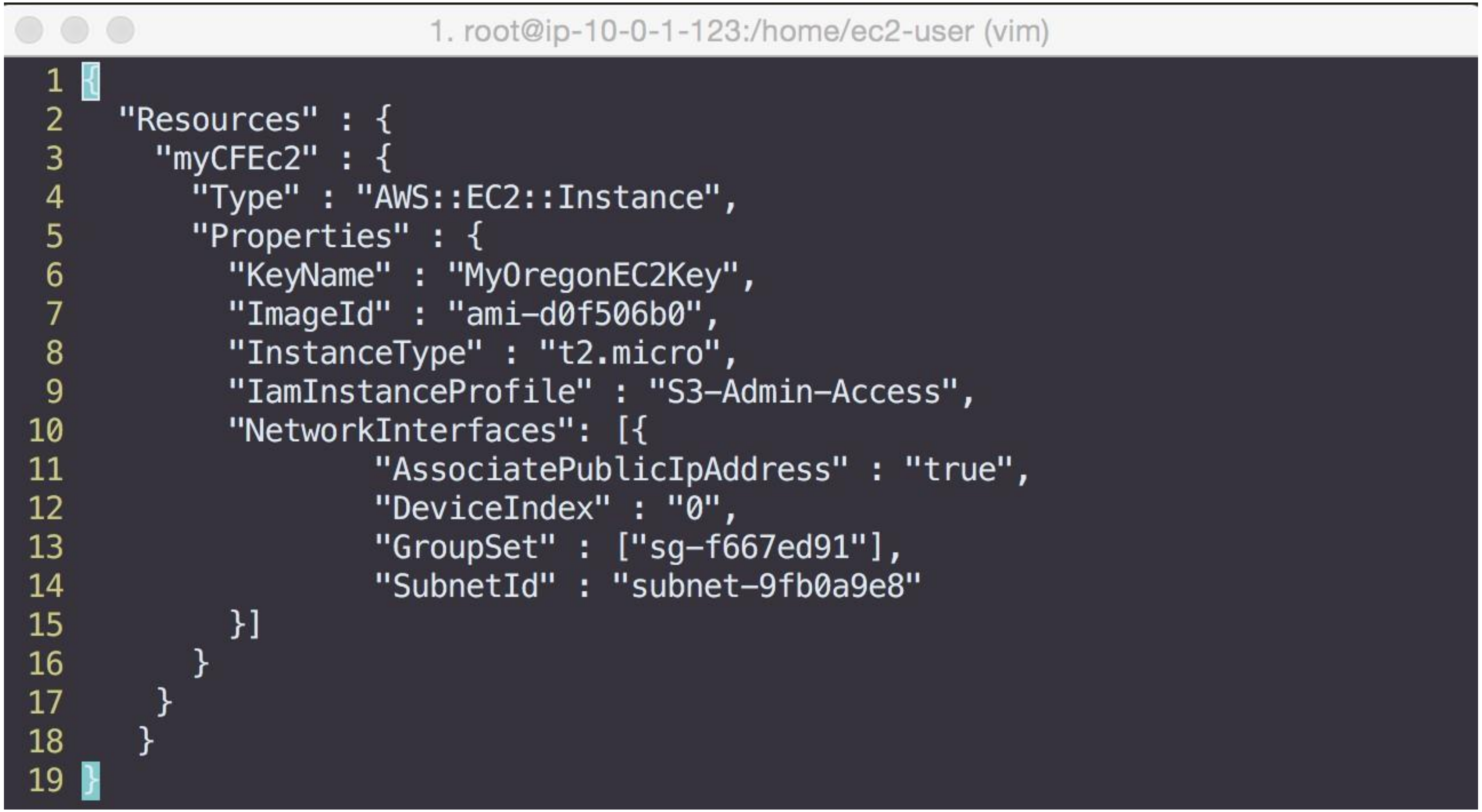
What Are We Testing?

comiq

Product Under Test



Infrastructure as Code (IaC)



```
{
  "Resources" : {
    "myCFEc2" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : "MyOregonEC2Key",
        "ImageId" : "ami-d0f506b0",
        "InstanceType" : "t2.micro",
        "IamInstanceProfile" : "S3-Admin-Access",
        "NetworkInterfaces": [{
                "AssociatePublicIpAddress" : "true",
                "DeviceIndex" : "0",
                "GroupSet" : ["sg-f667ed91"],
                "SubnetId" : "subnet-9fb0a9e8"
        }]
      }
    }
   }
}
```

https://medium.com/@samx18/the-future-of-infrastructure-as-code-373206a9dc96

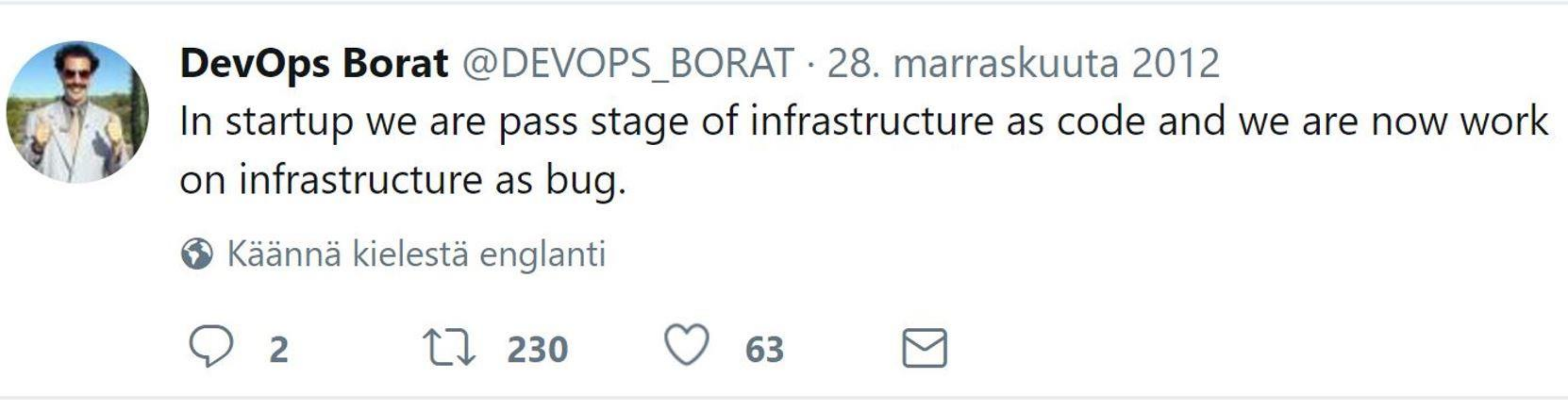
IaC for Testers?

- (Test) environment availability
- Independence from other teams
- Isolated test runs

Why Should We Test IaC?

- Similar level of quality for infrastructure as for your code
- Don’t assume
- Software dev & test good practices for IaC scripts

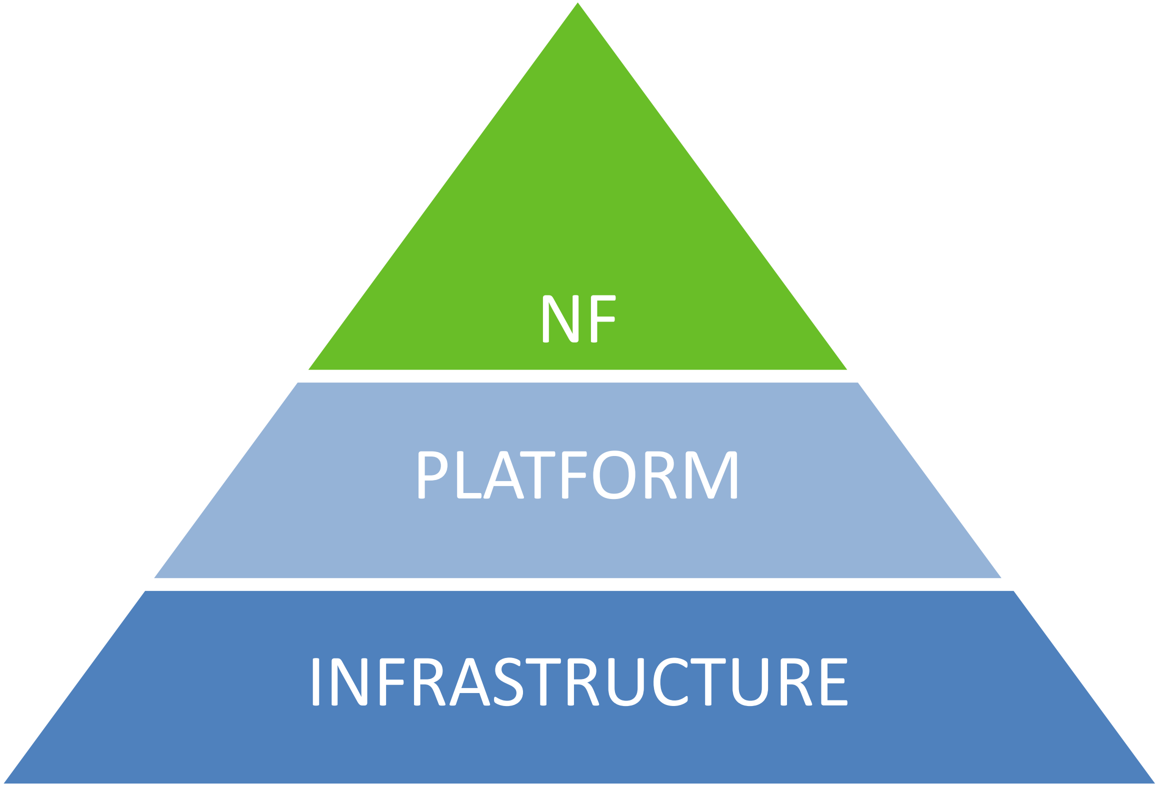
Why Should We Test IaC?



DevOps Borat @DEVOPS_BORAT · 28. marraskuuta 2012

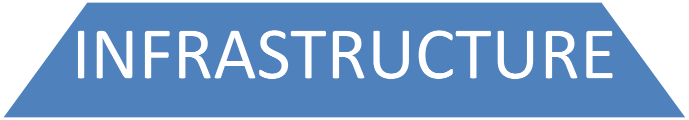
In startup we are pass stage of infrastructure as code and we are now work on infrastructure as bug.

Infrastructure Tests Levels



Smoke & Sanity Tests

- Quick, no transaction data
- Save time, better reliability
- Verify infra during deployment
- Example: cURL-call



INFRASTRUCTURE

Platform & Service Tests

- Infrastructure’s functional testing
- Examples: database available, load balancers working, platform scalability

PLATFORM

Non-Functional Tests

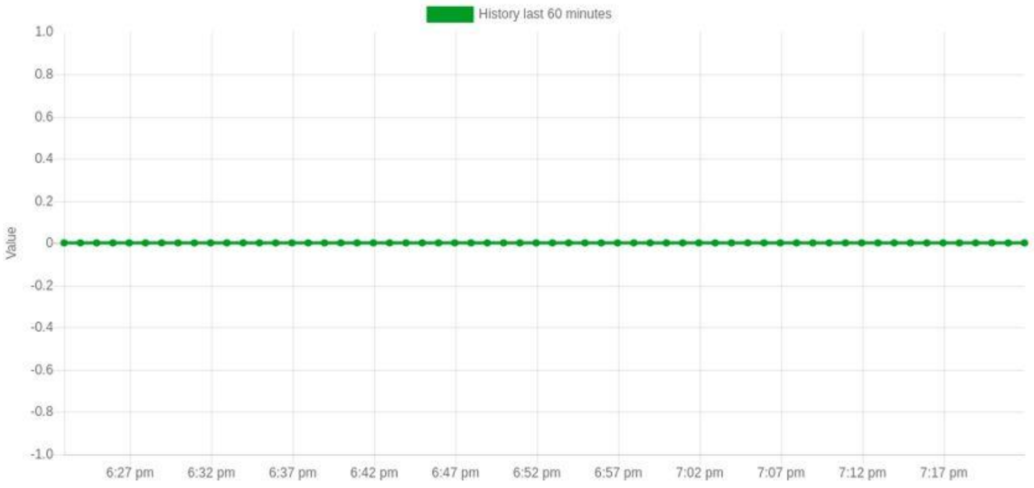
NF

- Verify environment behaviour
- Examples:
  - deployment, rollback, recovery, autohealing
  - Upscaling (demo) + downscaling
  - Chaos injection

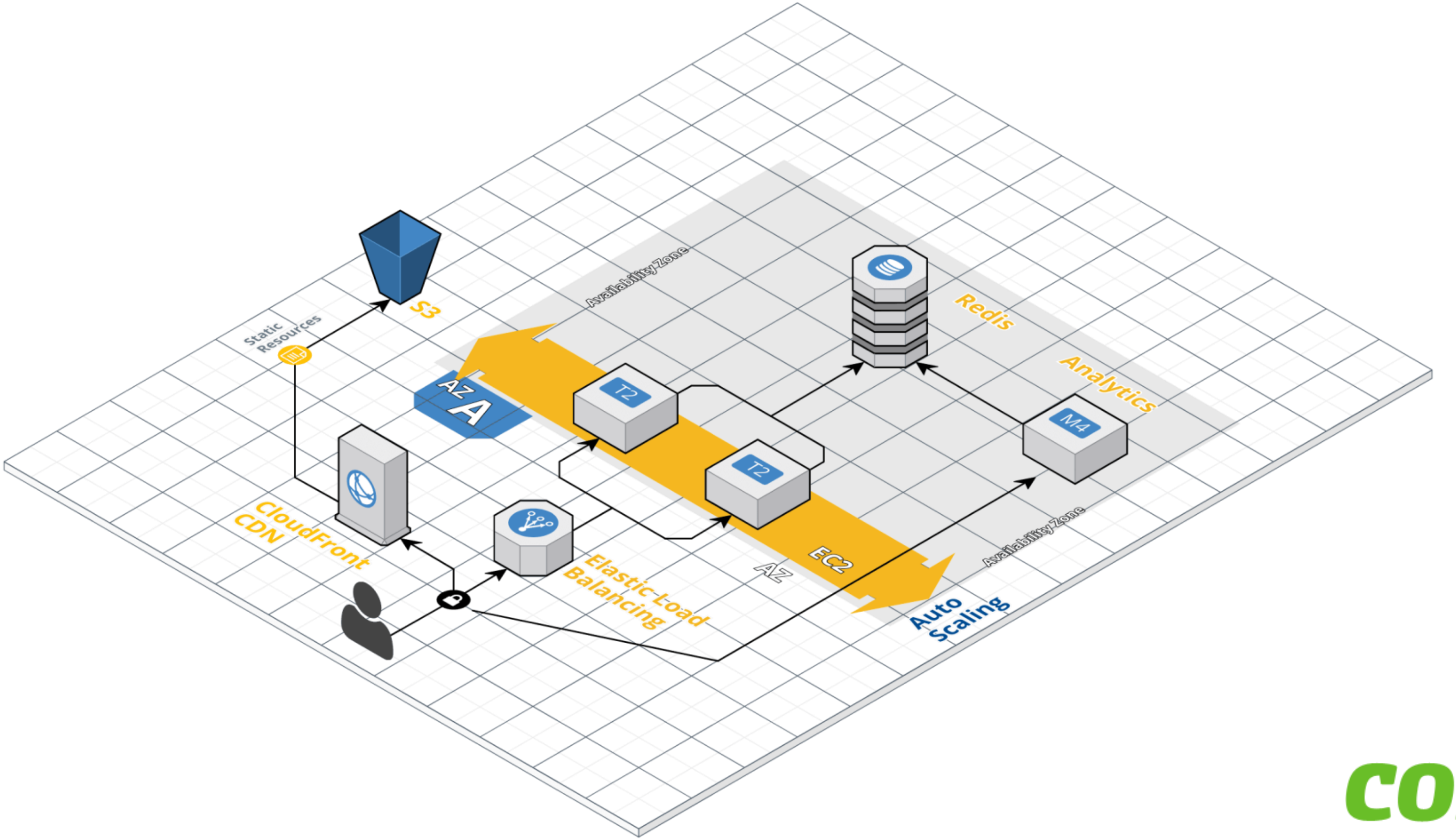
Example:
Testing infrastructure in AWS setup

Express your Valued opinion about this presentation!

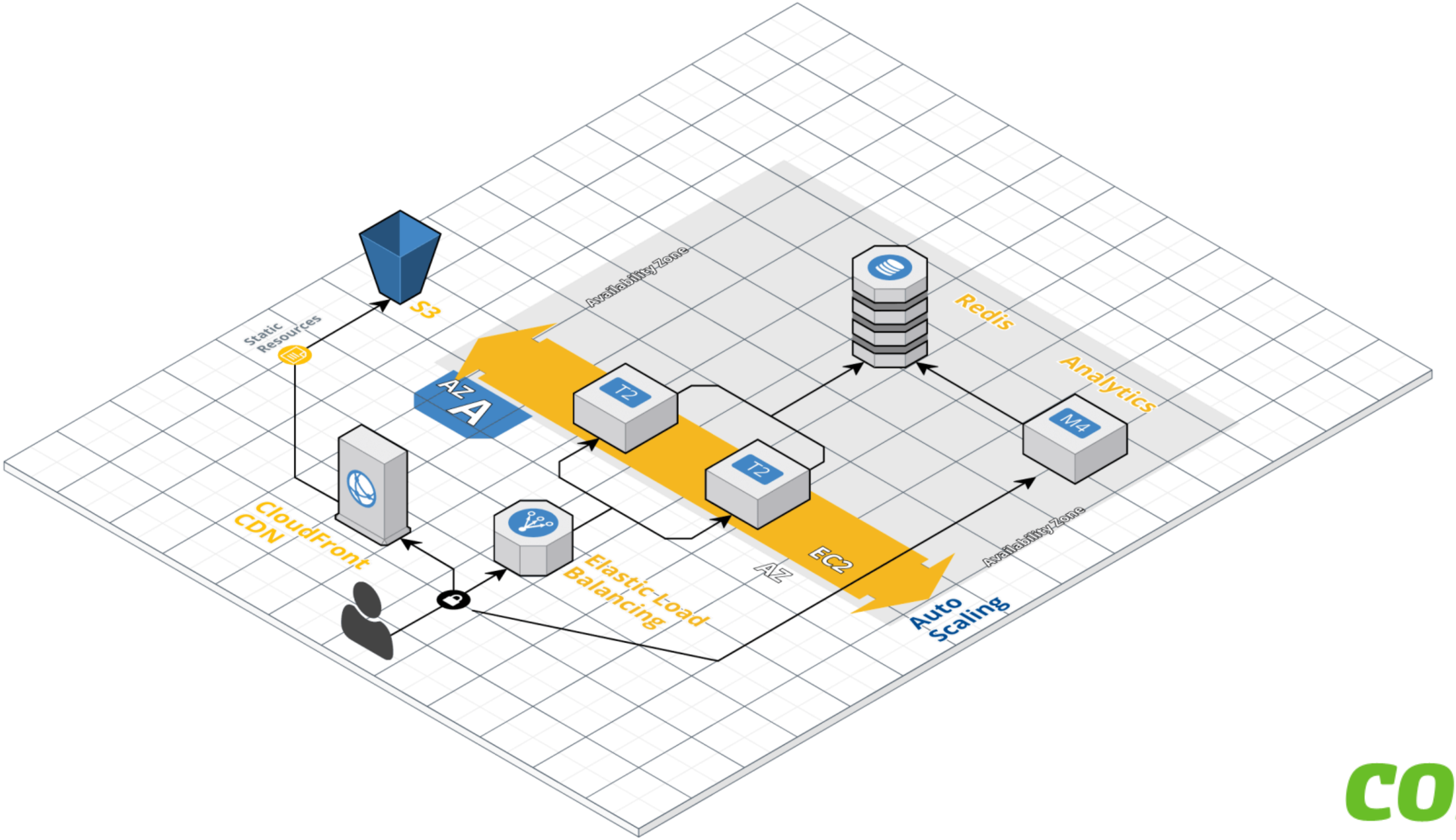
Change during last minute: +0



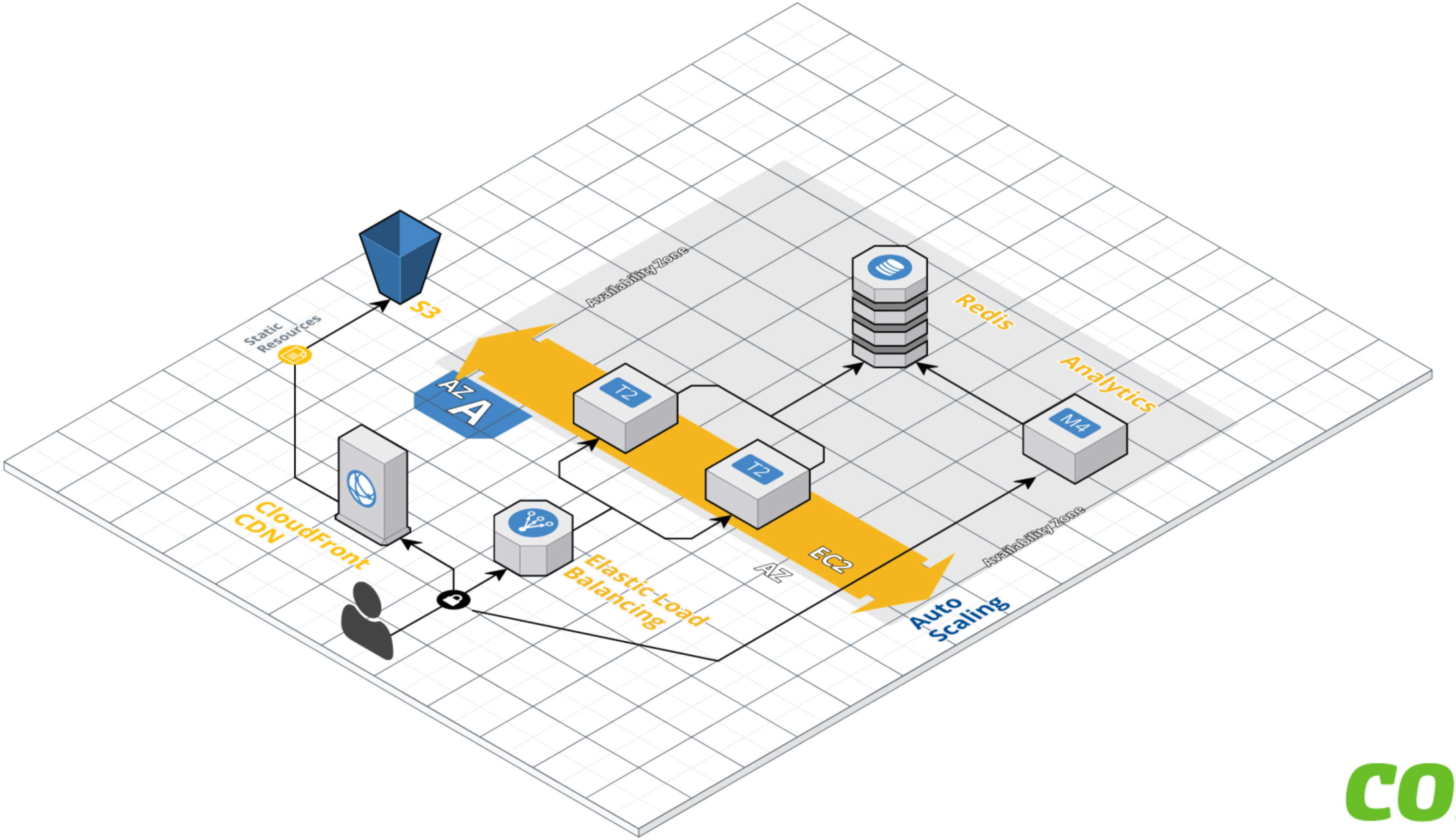
Initial Setup



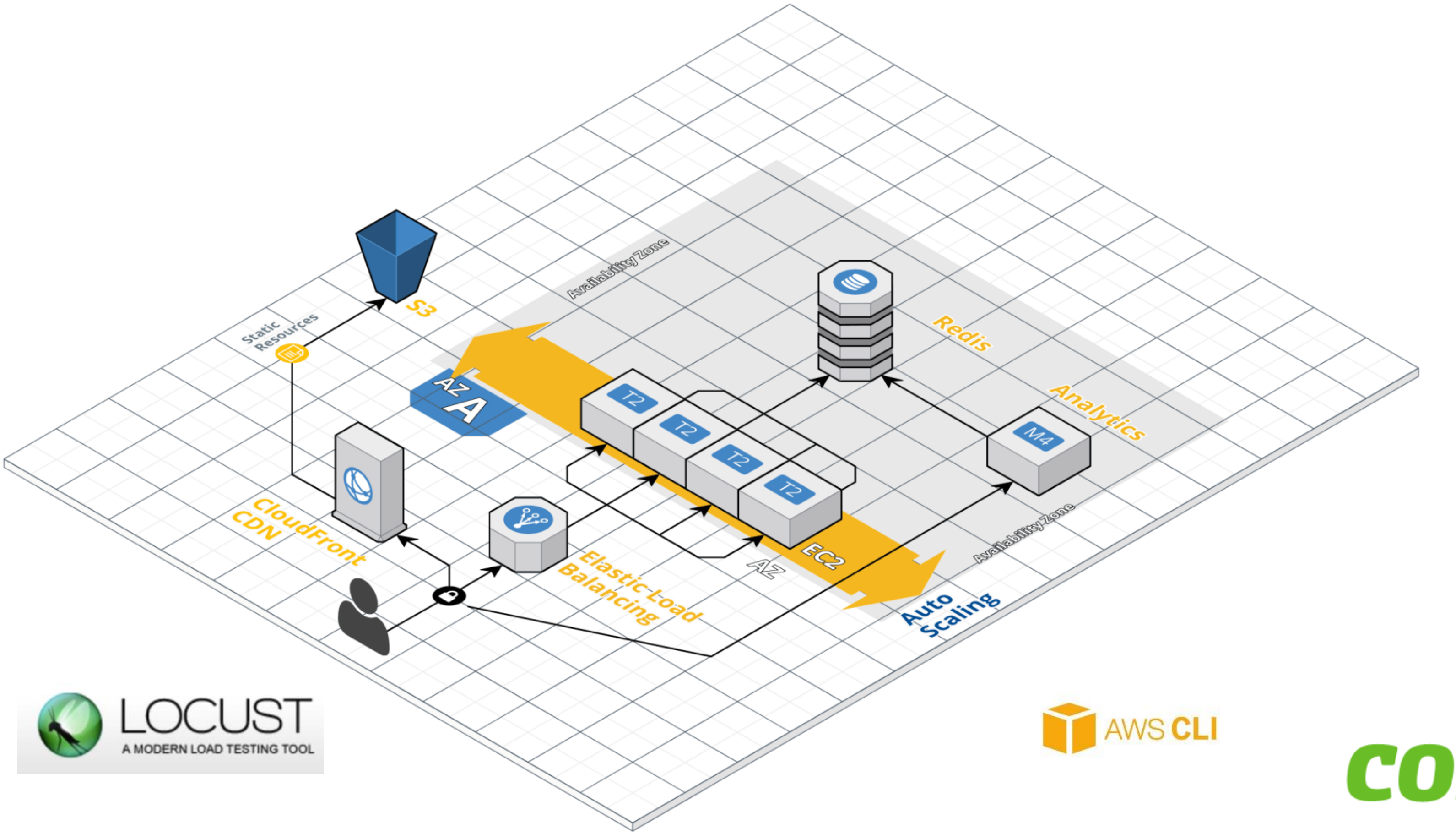
Infrastucture Tests



Platform Tests



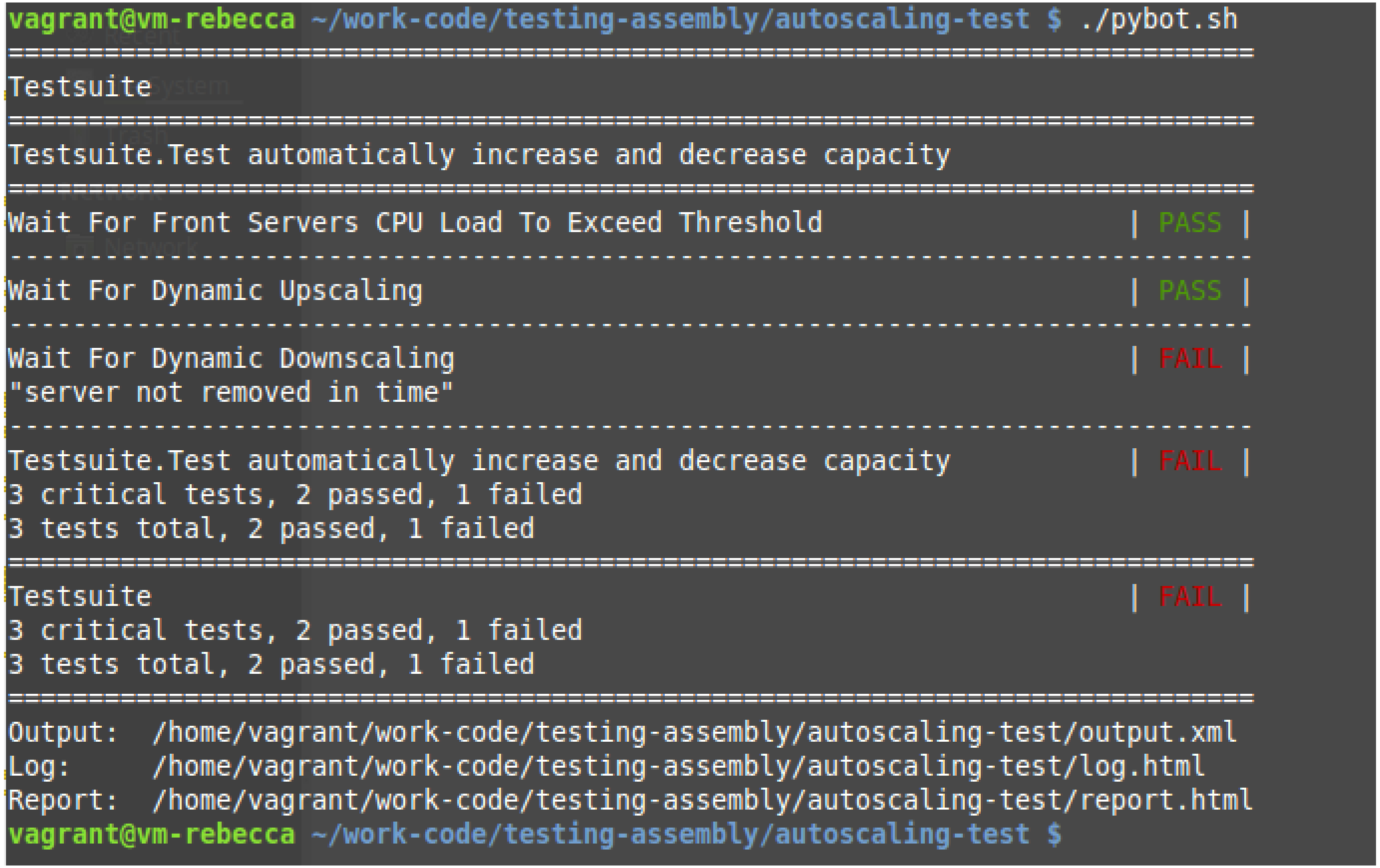
Autoscaling test



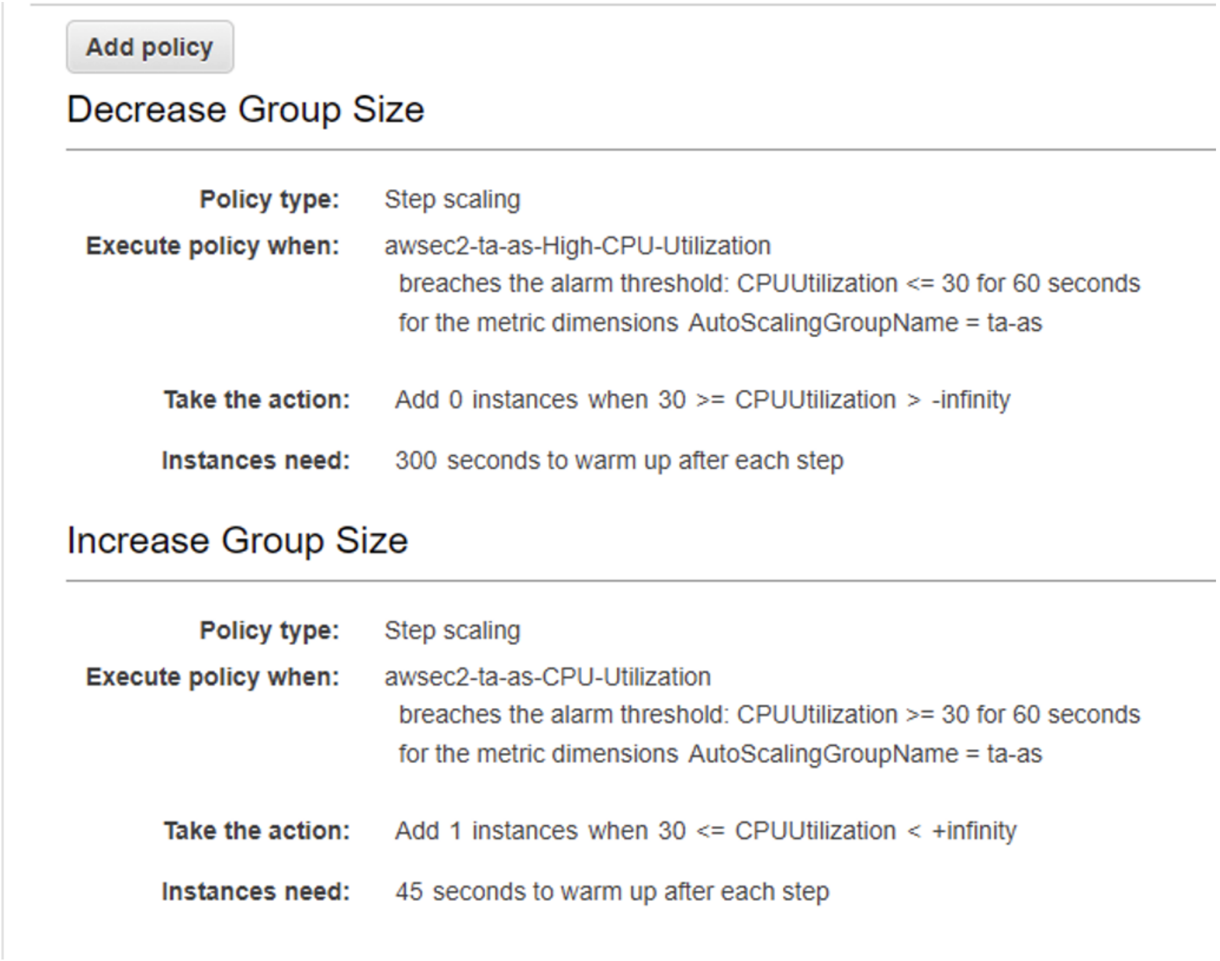
LOCUST
A MODERN LOAD TESTING TOOL

AWS CLI

Test run



```
vagrant@vm-rebecca ~/work-code/testing-assembly/autoscaling-test $ ./pybot.sh
==============================================================================
Testsuite
==============================================================================
Testsuite.Test automatically increase and decrease capacity
==============================================================================
Wait For Front Servers CPU Load To Exceed Threshold                   | PASS |
------------------------------------------------------------------------------
Wait For Dynamic Upscaling                                            | PASS |
------------------------------------------------------------------------------
Wait For Dynamic Downscaling                                          | FAIL |
"server not removed in time"
------------------------------------------------------------------------------
Testsuite.Test automatically increase and decrease capacity           | FAIL |
3 critical tests, 2 passed, 1 failed
3 tests total, 2 passed, 1 failed
==============================================================================
Testsuite                                                             | FAIL |
3 critical tests, 2 passed, 1 failed
3 tests total, 2 passed, 1 failed
==============================================================================
Output:  /home/vagrant/work-code/testing-assembly/autoscaling-test/output.xml
Log:     /home/vagrant/work-code/testing-assembly/autoscaling-test/log.html
Report:  /home/vagrant/work-code/testing-assembly/autoscaling-test/report.html
vagrant@vm-rebecca ~/work-code/testing-assembly/autoscaling-test $
```



Decrease Group Size

Policy type: Step scaling

Execute policy when: awsec2-ta-as-High-CPU-Utilization
breaches the alarm threshold: CPUUtilization <= 30 for 60 seconds
for the metric dimensions AutoScalingGroupName = ta-as

Take the action: Add 0 instances when 30 >= CPUUtilization > -infinity

Instances need: 300 seconds to warm up after each step

Increase Group Size

Policy type: Step scaling

Execute policy when: awsec2-ta-as-CPU-Utilization
breaches the alarm threshold: CPUUtilization >= 30 for 60 seconds
for the metric dimensions AutoScalingGroupName = ta-as

Take the action: Add 1 instances when 30 <= CPUUtilization < +infinity

Instances need: 45 seconds to warm up after each step

Summary

- ✓ Treat infrastructure as code
- ✓ Include automated infrastructure tests to your delivery pipeline
- ✓ Gradular test automation to build confidence and execute fast

comiq

Thank You!

comiq