



# The Journey of Testing with Stubs and Proxies in AWS

Lucy Chang

Senior Software Engineer in Quality, Intuit

<https://www.linkedin.com/pub/lucy-chang/11/312/a83>



## About Us



Consumers

Small Businesses

Accounting Professionals

intuit.  
**TurboTax** ✓

qb intuit  
QuickBooks

qb intuit  
QuickBooks.  
Online Accountant

intuit  
**mint**

intuit.  
Payroll

intuit. Lacerte

intuit.  
Tax Online

## Challenges

- Integration tests fails due to unreliable dependency servers
- Hard to test different test data permutations for dependency servers
- Need to do resiliency testing



## Our Solution

# Wiremock

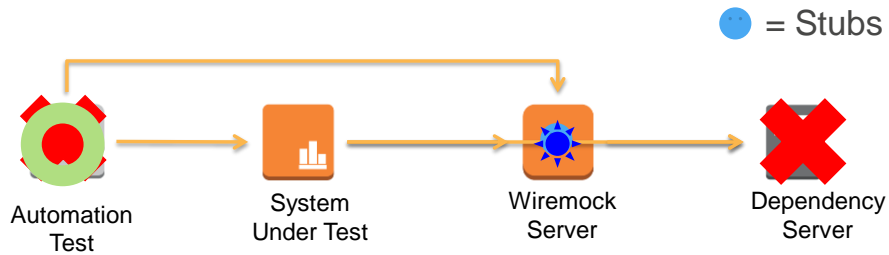


## Overview of Wiremock

Wiremock is a library for stubbing and proxying web services

- Stubbing
- Fault Injection
- Easy Set up
- Easy onboarding

## How does Wiremock work?



- Configure the Wiremock server to be man-in-the-middle
- Increased integration test pass rate
- Increased code coverage
- Does not interrupt other team's calling the SUT

## Overview of AWS

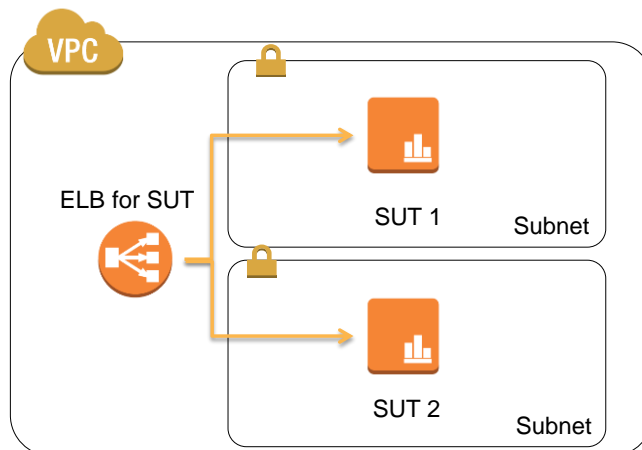
## Overview of AWS

Amazon Web Services (AWS) is a cloud computing services

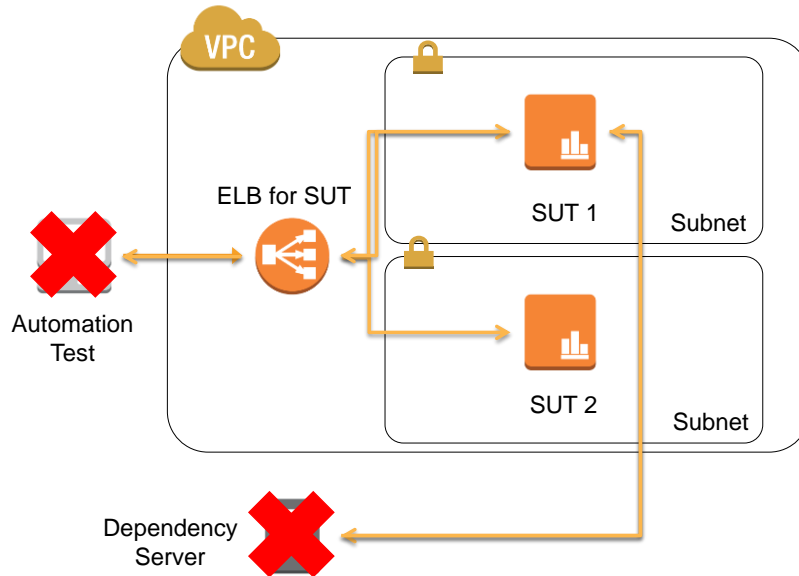
- On-demand delivery of IT resources and applications
- Pay-as-you-go pricing



## AWS Architecture

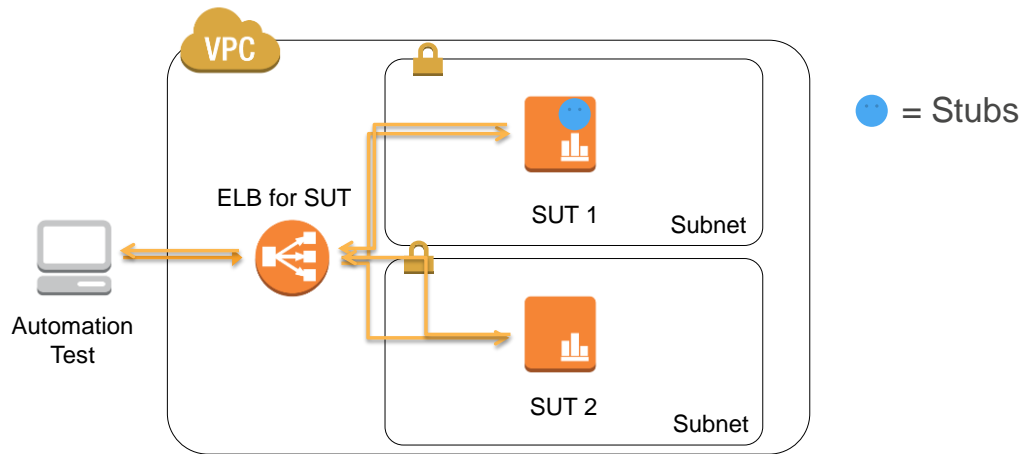


## Before Wiremock



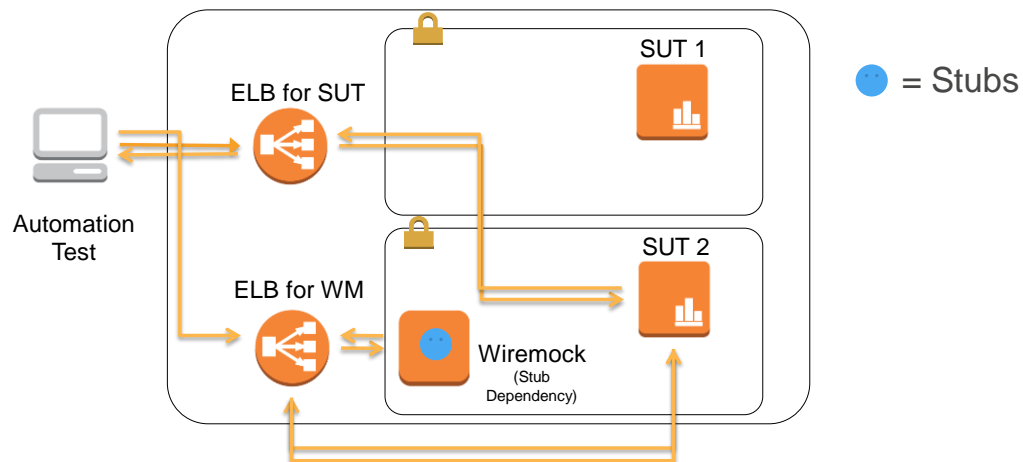
## Wiremock in AWS

## We tried this



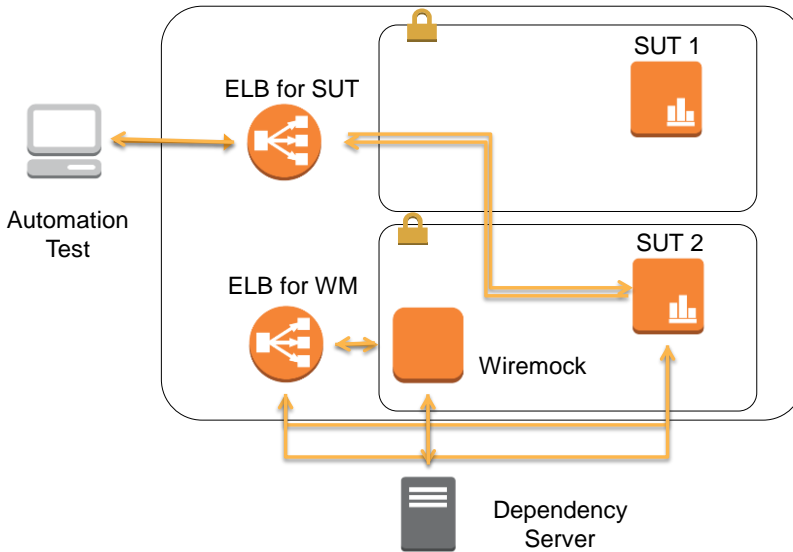
- Deploy Wiremock on SUT EC2 instance
- No consistent stub response!

## Our Solution



- Deploy Wiremock on dedicated EC2 and ELB
- Consistent Stub responses!

## If no stubs...



WM will proxy the request to dependency server

## Wiremock Code Snippets



## Starting Up Wiremock Server

```
java -jar wiremock-1.53-standalone.jar --verbose --  
port 8080 --proxy-all=[Dependency Server DNS Name]
```

## Stubbing the response

```
//This calls wiremock API to stub the response
```

```
stubFor(get(urlEqualTo("/from/where"))  
.willReturn(aResponse().withStatus(200)  
.withHeader("Cache-Control", "no-cache")  
.withHeader("Content-Type", "text/plain")  
.withBody("Taiwan" ));
```

## Simulating Fault

```
//This calls wiremock API for fault injection  
  
stubFor(get(urlEqualTo("/some/thing"))  
  .willReturn(aResponse()  
    .withFault(Fault.EMPTY_RESPONSE)));
```

## Automate WM Deployment

## Automate Wiremock Deployment

Chef is an infrastructure automation code tool we use

- Code how you deploy and manage your infrastructure
- Allows version control
- Code can be reused



## Automate Wiremock Deployment

We wrote a Wiremock Recipe

- Download the Wiremock jar
- Start up the Wiremock server



## Automate Wiremock Deployment

Berkshelf is a dependency manager for chef

- Get the Java recipe to download Java
- Get the Wiremock recipe to deploy Wiremock server



## Chef Snippet

#This will start the wiremock server with the parameters passed in

```
function start { cd "${USER_DIRECTORY}" ;java -jar  
wiremock-${WIREMOCK_VERSION}-standalone.jar --port  
${PORT} --proxy-via ${PROXY_VIA} --proxy-all=  
${PROXY_ALL} --verbose > /var/log/wiremock.log 2>&1  
& }
```

## Automate Wiremock Stack Creation

Use AWS CloudFormation API

- Provision EC2 instances and ELB
- Create Auto Scaling Group
- Set up other AWS resources



**AWS** CloudFormation

## Use WireMock

Call Chef from CloudFormation



**AWS** CloudFormation



**CHEF**™

## How to Call Cookbook From CloudFormation

Write shell scripts In the InstanceLaunchConfig section

1. Download and install chef
2. Run Chef. In this case, we created a Wiremock role to execute the Java and Wiremock cookbooks.

## Call Chef from CloudFormation

```
"5_run_chef": {
    "command": { "Fn::Join": [ "", [
"/usr/bin/chef-solo -c /var/chef/config/solo.rb -o
'role[" , { "Ref": "Role" }, "]" -E '" , { "Ref":
"Environment" }, "" ] ]
    }
}
```

# Deep Dives

## Demo



# Recap

## Recap

Solution: Wiremock

- Tests don't fail due to unreliable dependency servers
- Less engineers' time spent on unnecessary debugging
- Resiliency issues are found before production
- Less production escapes and firefighting
- Happy Engineers





## What we learned

### What We Learned

- The initial investment is worth it
- Try to be flexible
- Set up DNS



## References

- [My talk for CI/CD with Mocking and Resiliency Testing in AWS re:invent Conference](#)
- <http://wiremock.org/>
- <https://www.chef.io/>
- <http://docs.aws.amazon.com/AWSCloudFormation/latest/APIReference/Welcome.html>



## Contact

Lucy Chang

Senior Software Engineer in Quality , Intuit

<https://www.linkedin.com/pub/lucy-chang/11/312/a83>

Twitter: @bingbongbong888



## Acknowledgement

### Paper Reviewers

Moss Drake

Dave Patterson



# Thank you!

