Agenda

- Intros
  - Name, Company, interest in cloud computing / security, how did you find out about this MeetUp
- Netflix Chaos Monkey
- Forming a CSA Chapter
- Next Meeting

Intro to Netflix Chaos Monkey

What is Chaos Monkey

- Tool that deliberately terminates instances in Auto Scaling Groups
- Developed by Netflix and released as open source on 30 July
- Terminated 65,000 instances in production and test for Netflix, most of the time they never noticed
- Source: http://techblog.netflix.com/2012/07/chaos-monkey-released-into-wild.html

Why use Chaos Monkey

- Failure happens
- Software is complex
- Examples
  - Do load balancers route correctly when instances go offline
  - Can you reliably rebuild your instances
  - Patch deployment issues

Setting Up Chaos Monkey

- Uses Amazon Auto Scaling Groups
- Java based command line execution
- Optional Use of Asgard
- Amazon SimpleDB for storing events

What is Auto Scaling

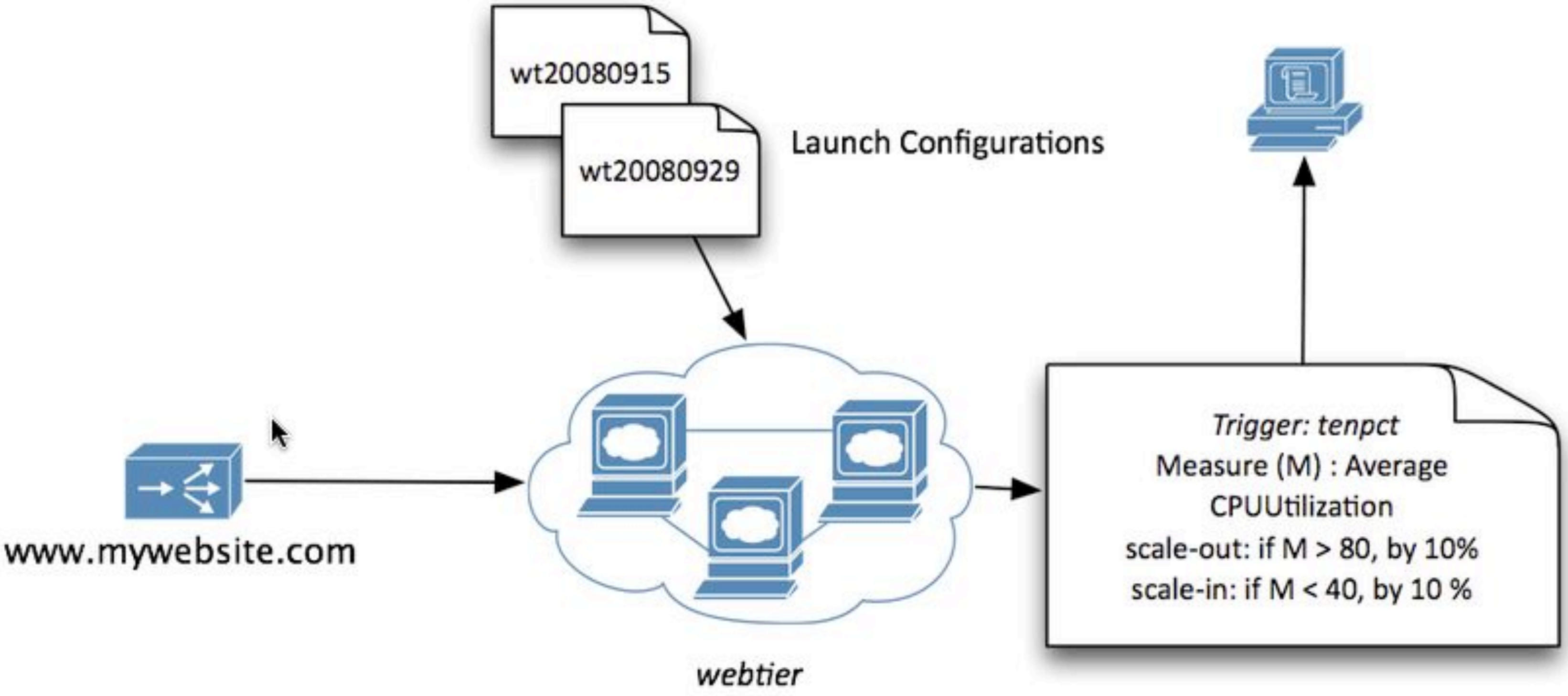
"Auto Scaling is a web service designed to launch or terminate EC2 instances automatically based on user-defined policies, schedules, and health checks."

Source: http://docs.amazonwebservices.com/AutoScaling/latest/DeveloperGuide/WhatIsAutoScaling.html

Autoscaling Features

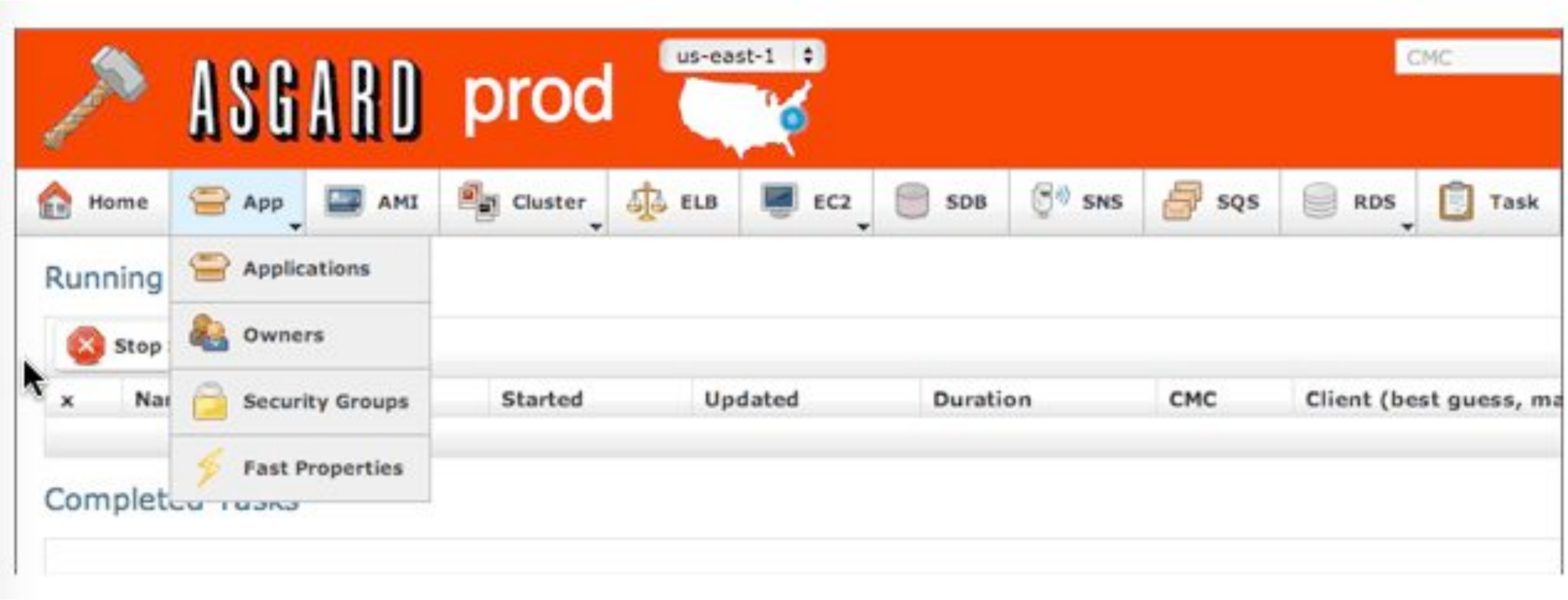
- Add and remove capacity based on use
- Mange instances across several Availability Zones via command line ore web service API
- Replace unhealthy instances based on predefined alarms and thresholds
- Schedule scaling on specific dates based on anticipated needs

AWS Auto Scaling


wt20080915
wt20080929
Launch Configurations
www.mywebsite.com
webtier
Trigger: tenpct
Measure (M) : Average CPUUtilization
scale-out: if M > 80, by 10%
scale-in: if M < 40, by 10 %

Source: http://docs.amazonwebservices.com/AutoScaling/latest/DeveloperGuide/AS_Concepts.html

Asgard



- Cloud deployment and management tool tool developed and released as open source by Netflix

Setting Up Chaos Monkey

- Download Auto Scaling Tools
- Create Auto Scaling Image
- Create Auto Scaling Group
- Start Auto Scaling Group
- Set Up SimpleDB Table
- Build the Monkeys with Gradle

Download Auto Scaling Tools

```
$ wget http://ec2-downloads.s3.amazonaws.com/AutoScaling-2011-01-01.zip

$ unzip AutoScaling-2011-01-01.zip

$ cd AutoScaling-1.0.61.0/

$ export AWS_AUTO_SCALING_HOME=`pwd
```

Setup Environment

```
$ export
AWS_AUTO_SCALING_URL=http://
autoscaling.us-west-2.amazonaws.com

$ export
ACCOUNT_KEY=your_account_key

$ export SECRET_KEY=your_secret_key
```

Create Launch Config

```
$ $AWS_AUTO_SCALING_HOME/bin/as-create-launch-config lc1 --instance-type t1.micro -I $ACCOUNT_KEY -S $SECRET_KEY --image-id ami-fcf27fcc

OK-Created launch config
```

Create Auto Scaling Group

```
$ $AWS_AUTO_SCALING_HOME/bin/as-
create-auto-scaling-group monkey-target -I
$ACCOUNT_KEY -S $SECRET_KEY --
launch-configuration lc1 --availability-zones
us-west-2a --min-size 1 --max-size 10
```

Start Auto Scale Group

```
$ $AWS_AUTO_SCALING_HOME/bin/as-describe-auto-scaling-groups -I $ACCOUNT_KEY -S $SECRET_KEY

AUTO-SCALING-GROUP  monkey-target  lc1  us-west-2a  1  1  1

INSTANCE  i-8b55fbb8  us-west-2a  InService  Healthy  lc1
```

Set Up SimpleDB

- See website

Build Simian Army

- Check out code from github
- Build with Gradle

Unleash the Monkey

```
set simianarmy.chaos.leashed=false
```

Configuration

- Opt in or Opt out model
- Tunable probability
- Probability of 1 will terminate 1 instance per ASG per day if set to day

Probability

“The probability is the run probability. If Chaos is running hourly between 9am and 3pm with an overall configured probability of "1.0" then the probability provided to this routine would be 1.0/6 (6 hours in 9am-3pm). So the typical probability here would be .1666. For Chaos to select an instance it will pick a random number between 0and 1. If that random number is less than the .1666 it will proceed to select an instance and return it,otherwise it will return null. Over 6 runs it is likely that the random number be less than .1666, but it is not certain.”

Source Code

Configuration

- Can set timeframe that it runs
- Netflix runs during normal working hours

Simian Army

Discussed by Netflix but not released as open source yet

Latency Monkey

- Adds delays to RESTFUL client server communications

Conformity Monkey

- Terminates instances that don’t follow “best practices”

Doctor Monkey

- Does health checks for things like high CPU, removed from service

Janitor Monkey

- Looks for unused resources and removes them

Security Monkey

- Terminates instances with vulnerabilities or security violations
- Also checks for outdated DRM and SSL certs

10-18 Monkey

- Looks for problems with instances in different langauges or using different character sets

Chaos Gorilla

- Simulates an outage in an entire region

Where to learn more

- References are available here:

  http://brightmoonsecurity.com/blog/

- Contact me with any questions:

  chris@brightmoonsecurity.com

  @BrightMoonSec