# Configure Your Cloud for Auto-Recovery

Walter Bentley
Cloud Solutions Architect – Private Cloud

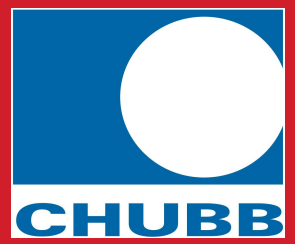
Chris Woodard
OpenStack Architect

# Walter Bentley

Cloud Solutions Architect, RPC

Twitter: @djstayflypro
LinkedIn: http://goo.gl/r2p21i
GitHub: wbentley15
Blog: hitchnyc.com

- Over 17 years of IT experience
- New Yorker (soon to be a Texan)
- Cloud Advocate (hybrid is my favorite)
- Knowledge sharer/author
- OpenStack believer
- Motorcyclist and DJ (literally…no lie)
- Always about living life now!








## Chris Woodard
OpenStack Architect, RPC

- Over 10 years at Rackspace
- 4 Years OpenStack
- Proves that anyone can do this job, even someone with a Louisiana public school education

# Ground Rules

- **Not going to ask you to turn off mobile phones but, if it rings its mine :D**
- **Ask questions *(requirement)***
- **Take any side conversations outside *(mainly because I like hearing myself talk only)***
- This workshop is hands on, please group yourself into groups no larger than 2-3. Please take turns doing stuff!
- Materials for the workshop are available here (case matters):

https://goo.gl/z0Nhdk

# Before we get started…

- **Please form teams of 3 or more**
- **Each group will be given a StudentID and instructions to connect to the OpenStack cloud**
- **Each team will be given an OpenStack Liberty release cloud and** access to the StackStorm server
- All teams will actively use the Nagios server to determine environment health

SECTION 1

# Services Auto-Recovery Approach

Using Nagios and StackStorm

# OpenStack Services – the facts

- ✓ Services will fail (trust me its normal :D)
- ✓ There is an extreme amount of dependency between the core OpenStack services
- ✓ One services failure can and will cause other services to fail
- ✓ Recovering a service can be as simple as restarting the service(s)
- ✓ There are many ways to manage and monitor the individual services
  a. Lots of prayers
  b. Monitoring platform
  c. Let Rackspace do it…

# Automated remediation –

**is the ability to** identify and verify system failures with the focus of taking actions in response to events in an automated fashion

# Lab Overview

- The teams will be tasked with first manually troubleshooting an OpenStack service failure
- Next the teams will then step thru the process of automating the recovery of OpenStack service(s) with StackStorm

# Let’s get started…

**Please go to the URL below in your browser (case matters):**

**https://goo.gl/z0Nhdk**

**Next connect to the Lab environment,**
**connection details are on the handout**

# GO!!!

Part 1 of the Lab will take 30-40 minutes

SECTION 2

# What is StackStorm?

StackStorm Overview

# What is StackStorm?

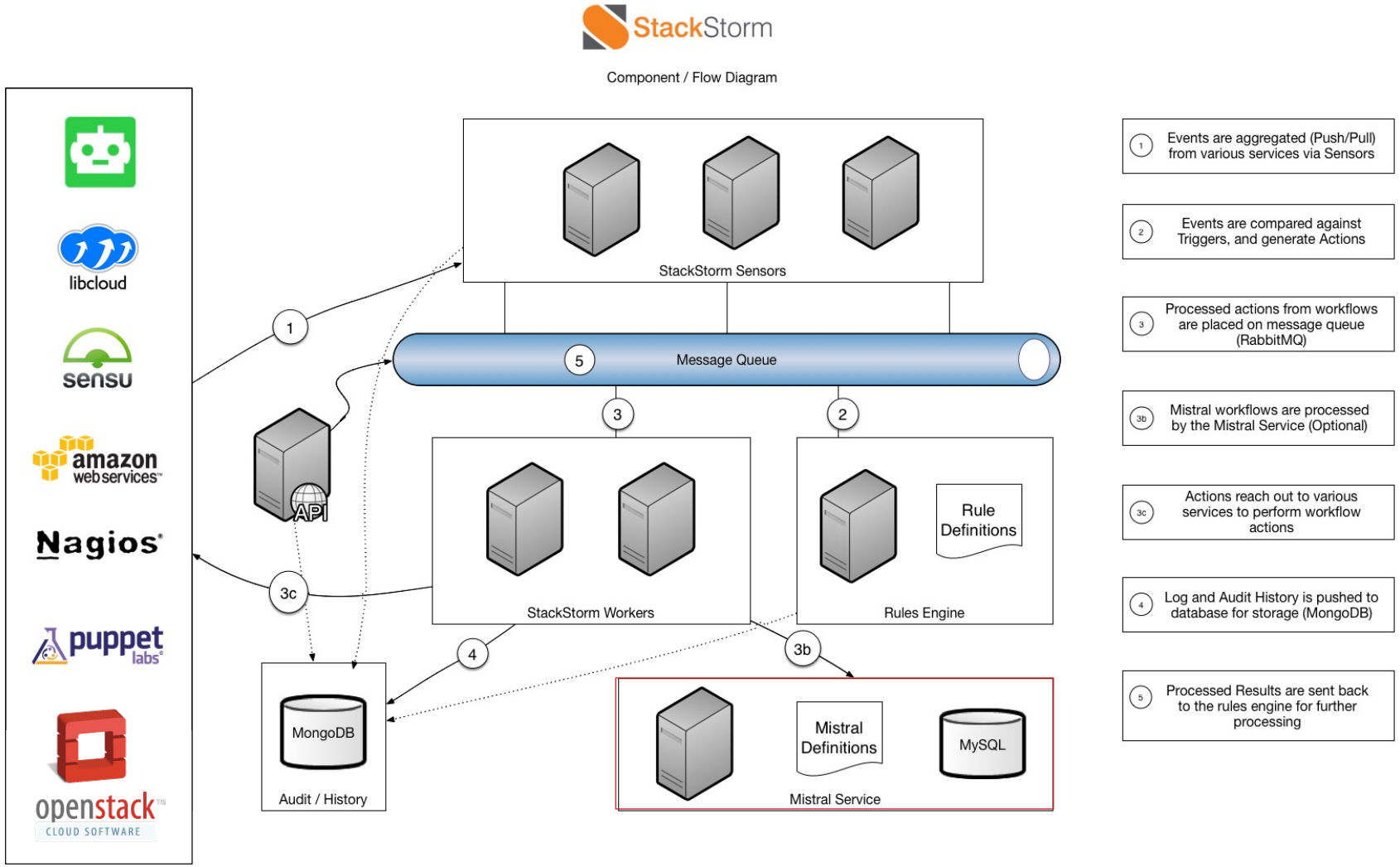
**StackStorm** *is a platform for integration and automation across services and tools. It ties together your existing infrastructure and application environment so you can more easily automate that environment – with a particular focus on taking actions in response to events.*

StackStorm helps automate common operational patterns.

- **Facilitated Troubleshooting**
- **Automated remediation**
- **Continuous deployment**

https://docs.stackstorm.com/latest/overview.html

# StackStorm –How it Works



# StackStorm – Nagios Integration Pack

**Repo Location:** https://github.com/StackStorm/st2contrib/tree/master/packs/nagios

Nagios Requirements:

- Must enable event handlers (global or service based)
- Be able to access the StackStorm API's
- Create custom Nagios command

Example:

```
define command {
        command_name    stackstorm
        command_line    /usr/lib/nagios/plugins/st2service_handler.py $SERVICEEVENTID$ "$SERVICEDESC$" $SERVICESTATE$ $SERVICESTATETYPE$
                        $SERVICEATTEMPT$ $HOSTNAME$
}
```

FREE OpenStack Books!


YES...FREE OpenStack Books!

#OPENSTACK @djstayflypr

COME TO THE

# RACKSPACE CANTINA

**RELAX | RECHARGE | REFRESH**

AT **2ND** AND **TRINITY**

(Right across the street from the Austin Convention Center)

See why it's a **NEW DAWN** for **OPENSTACK** with our **99.99% API UPTIME GUARANTEE.**


#OPENSTACK

rackspace