USING CHAOS TO BUILD RESILIENT SYSTEMS

@tammybütow, Gremlin

What’s the scale of your infra?

How many services do you
have running in production?

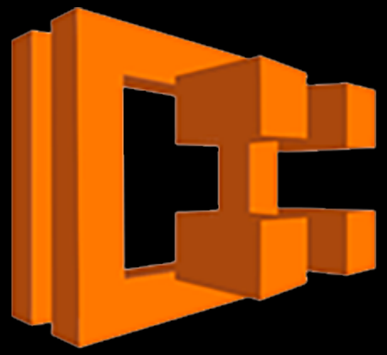
How many engineers do you
have at your company?

A Common Chaos Engineering Journey

TOP 5 MOST POPULAR WAYS TO USE CHAOS ENGINEERING IN 2018

kubernetes

kafka

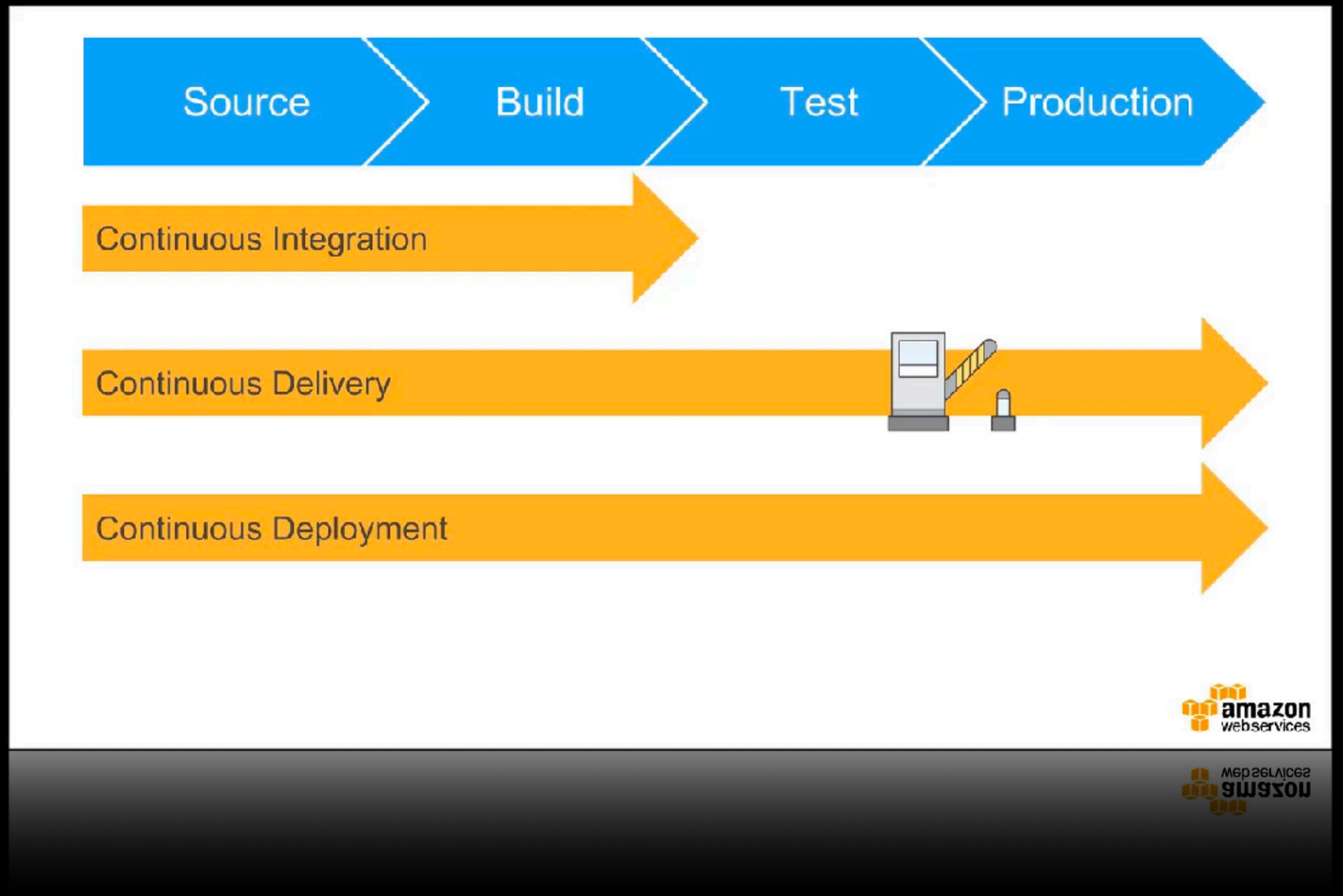


AWS ECS

cassandra

elasticsearch

ADVANCED USES OF CHAOS ENGINEERING



What happened this week: June 2018 Slack Outage

▲ londons_explore 2 days ago [-]

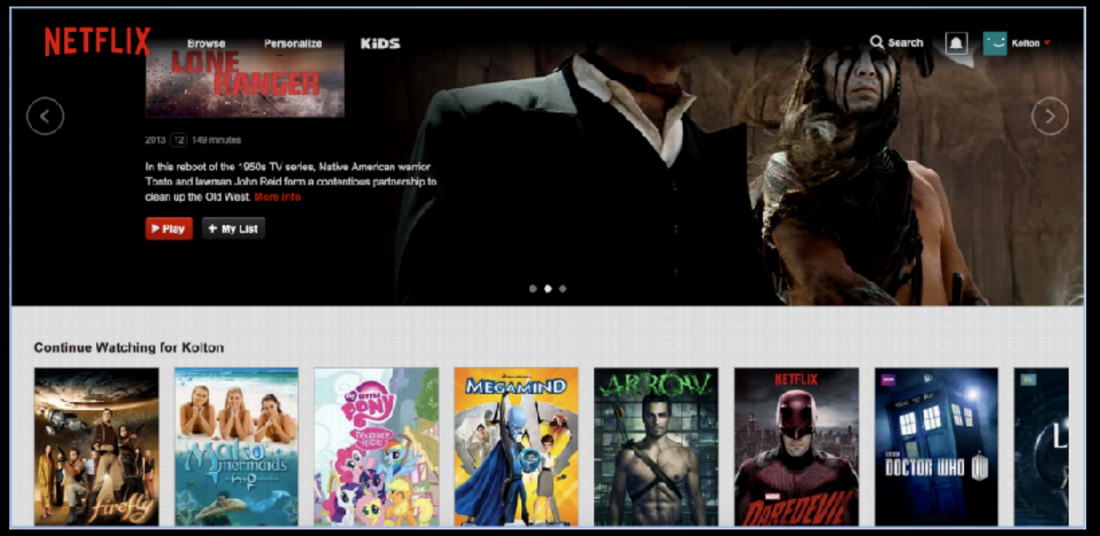
Google is expert at designing services which *you won't notice when there is downtime*.

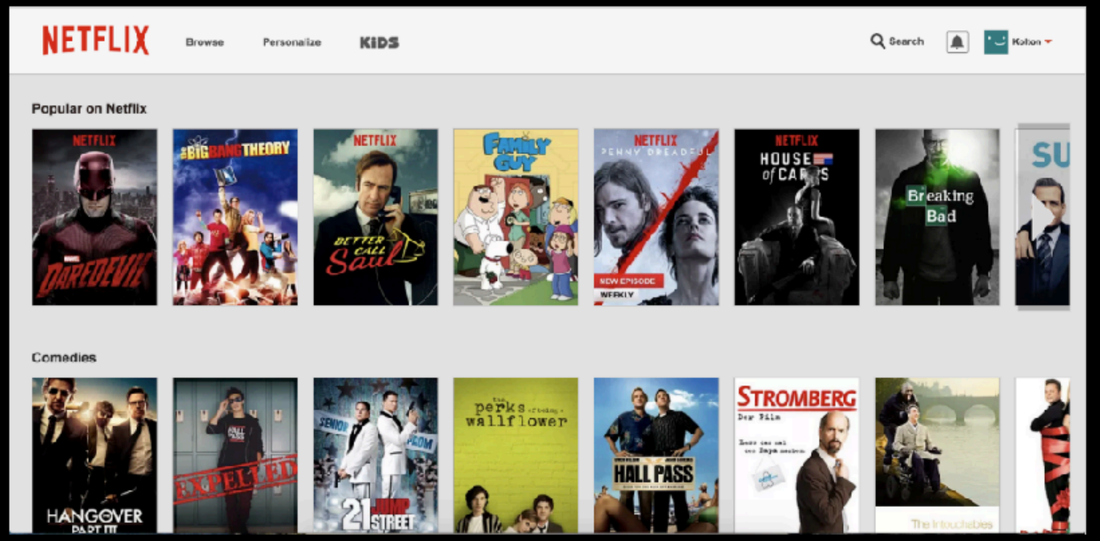
Take Google Search for example. When there is downtime, results might be slightly less accurate, or the parcel tracking box might not appear, or the page won't say the "last visited" time beside search results.

The SRE's are running around fixing whatever subsystem is down or broken, but you the user probably don't notice.

kolton 12:04 PM

And testing those graceful degradations ... is where chaos eng comes in. :)


NETFLIX
Browse
Personalize
KIDS
Search
Kolton
LONE RANGER
2013 12 149 minutes
In this reboot of the 1950s TV series, Native American warrior Tonto and lawman John Reid form a contentious partnership to clean up the Old West. More info
Play
My List
Continue Watching for Kolton
Firefly
Mako Mermaids
MEGAMIND
ARROW
DAREDEVIL
DOCTOR WHO


NETFLIX
Browse
Personalize
KIDS
Search
Kolton
Popular on Netflix
DAREDEVIL
THE BIG BANG THEORY
BETTER CALL SAUL
FAMILY GUY
HOUSE of CARDS
Breaking Bad
Comedies
HANGOVER PART III
EXPELLED
21 JUMP STREET
perks of being a wallflower
HALL PASS
STROMBERG

❤️

🙈

TAMMY BÜTOW

Principal SRE, Gremlin
Causing chaos in prod since 2009.

Previously SRE Manager @ Dropbox
leading Databases, Block Storage and
Code Workflows for 500 million users
and 800 engineers.

@tammybütow

GREMLIN

- We are practitioners of Chaos Engineering

- We build software that helps engineers build resilient systems in a safe, secure and simple way.

- We offer 11 ways to inject chaos for your Chaos Engineering experiments (e.g. host/container packet loss and shutdown)

PART 1: LAYING THE FOUNDATION

Let’s Define A Resilient System:

- A resilient system is a highly available and durable system.
- A resilient system can maintain an acceptable level of service in the face of failure.
- A resilient system can weather the storm (a misconfiguration, a large scale natural disaster or controlled chaos engineering).

It would be silly to give an
Olympic pole-vaulter a broom
and ban them from practicing!

“Thoughtful planned experiments designed to reveal the weaknesses in our systems”

- Kolton Andrus, Gremlin CEO

Think of it like a vaccination:

Inject something harmful in order to build an immunity.

Eventually systems will break
in many undesired ways.

Break them first on purpose with
controlled chaos! 💥

DOGFOODING

- Using your own product.
- For us that means using Gremlin for our Chaos Engineering experiments.
- Failure Fridays

Failure Fridays are dedicated time for teams to collaboratively focus on using Chaos Engineering practices to reveal weaknesses in your services.

WHY DO DISTRIBUTED SYSTEMS NEED CHAOS?

- Unusual hard to debug failures are common

- Systems & companies scale rapidly and Chaos Engineering helps you learn along the way

FULL-STACK CHAOS ENGINEERING

- You can inject chaos at any layer.
- API, App, Cache, Database, OS, Host, Network, Power & more.

WHY RUN CHAOS ENGINEERING EXPERIMENTS?

Are you confident that your metrics and alerting are as good as they should be?

#pagerpain 

Are you confident your customers are
getting as good an experience
as they should be?
#customerpain 😞

Are you losing money due to downtime and broken features?

#businesspain 

HOW DO YOU RUN CHAOS ENGINEERING EXPERIMENTS?

HOW TO RUN A CHAOS ENGINEERING EXPERIMENT

- Form a hypothesis
- Consider blast radius
- Run experiment
- Measure results
- Find & fix issues or scale

Don’t run before you can walk

The 3 Prerequisites for Chaos Engineering

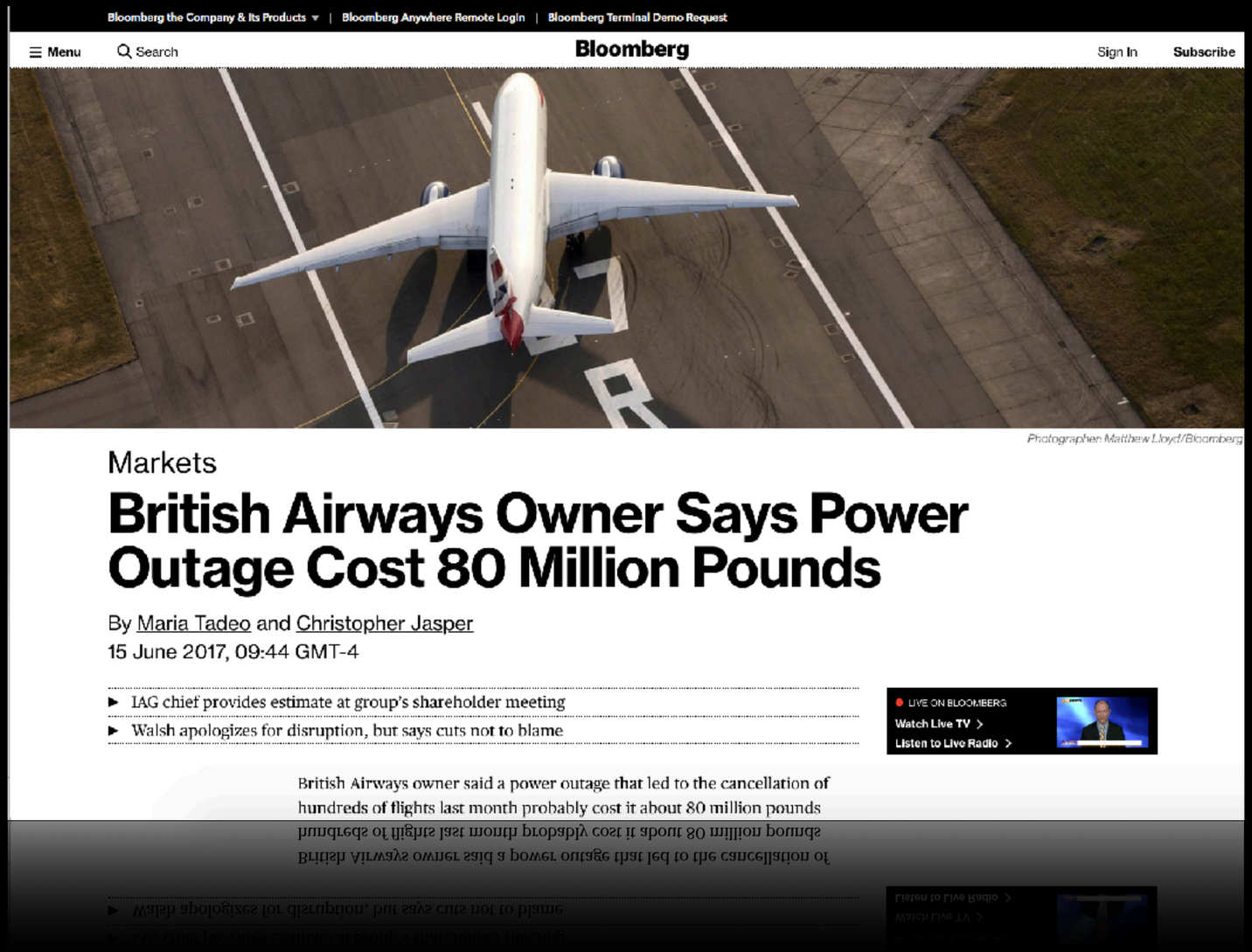
1. Monitoring & Observability
2. On-Call & Incident Management
3. Know Your Cost of Downtime Per Hour

What Do I Use For Monitoring & Observability?

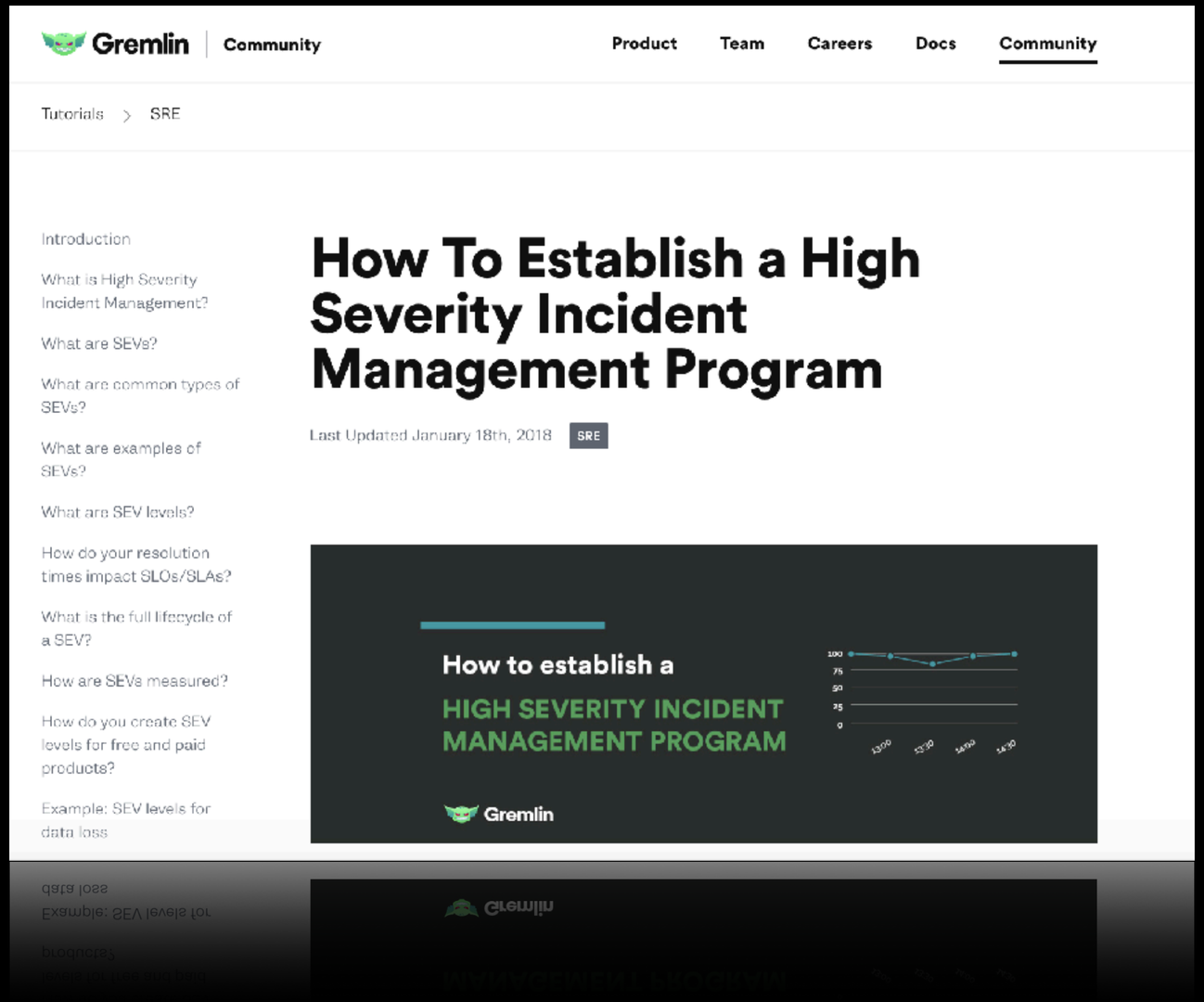
SENTRY

WIRESHARK

We All Need To Know The Cost Of Downtime



We All Need Incident Management



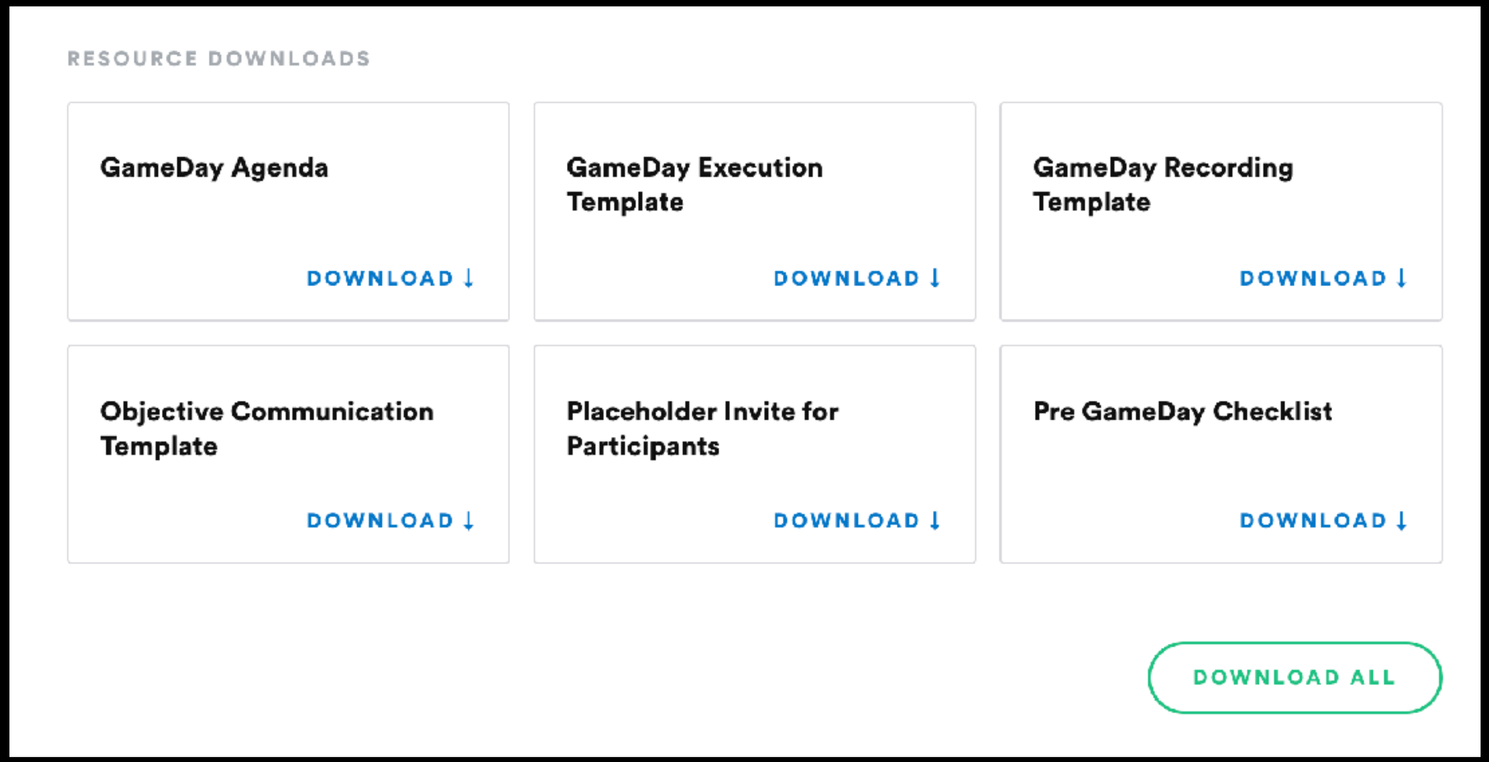
HOW TO CHOOSE A CHAOS EXPERIMENT

- Identify top 5 critical systems
- Choose 1 system
- Whiteboard the system
- Select attack: resource/ state/network
- Determine scope

WHAT SHOULD WE MEASURE?

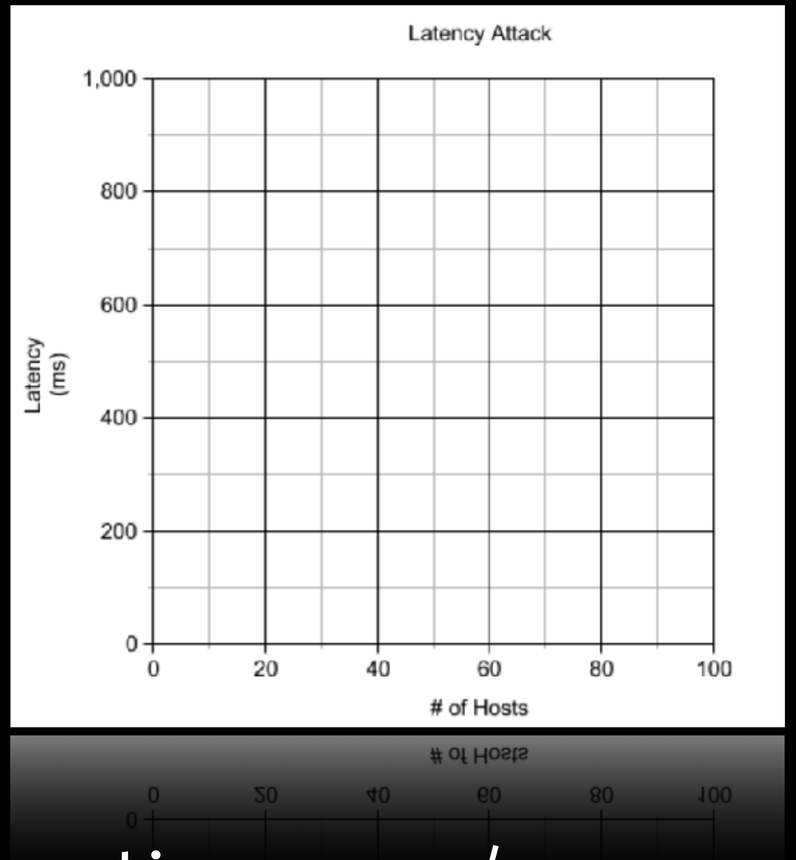
- Availability — 500s
- Service specific KPIs
- System metrics: CPU, IO, Disk
- Customer complaints

HOW TO RUN YOUR OWN GAMEDAY!



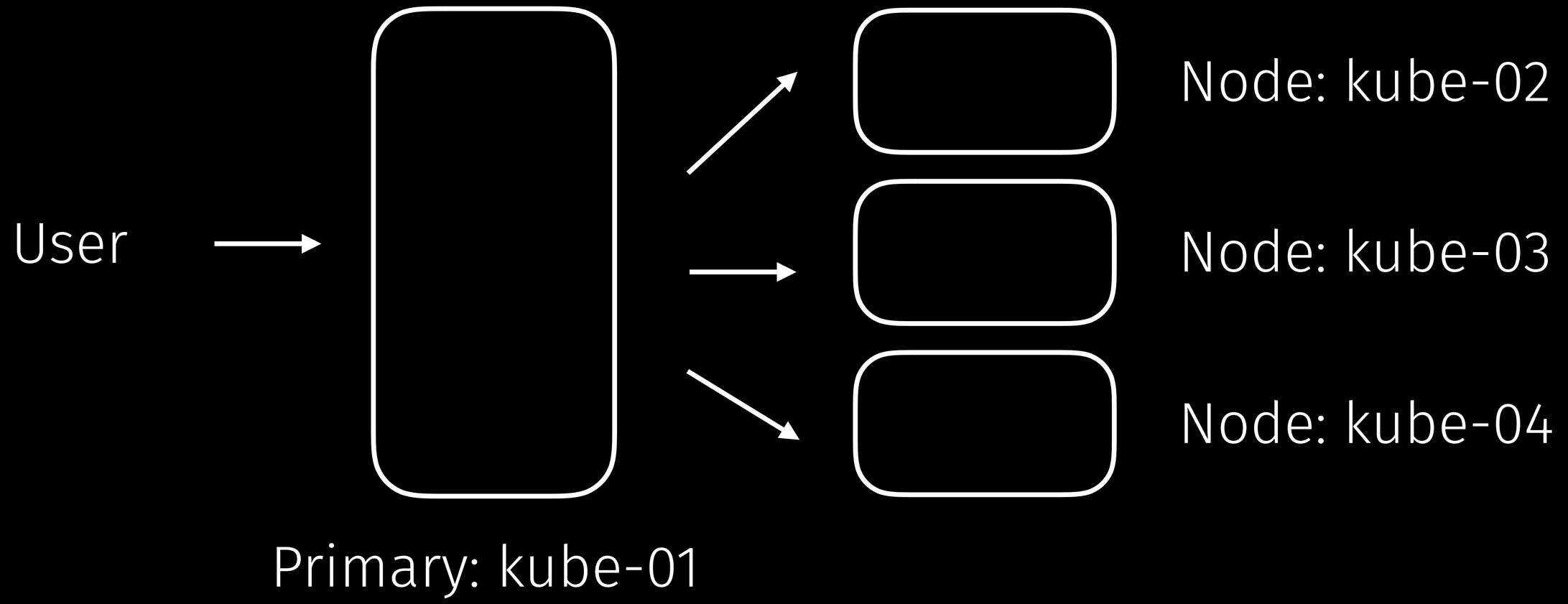
gremlin.com/gameday

HOW TO RUN YOUR OWN GAMEDAY!



gremlin.com/gameday

EXAMPLE SYSTEM: KUBERNETES RETAIL STORE



PART 2: RESOURCE CHAOS ENGINEERING

RESOURCE CHAOS

We can increase CPU, Disk, IO & Memory consumption to ensure monitoring is setup to catch problems.

Important to catch issues before they turn into high severity incidents (unable to purchase new product!) and downtime for customers.

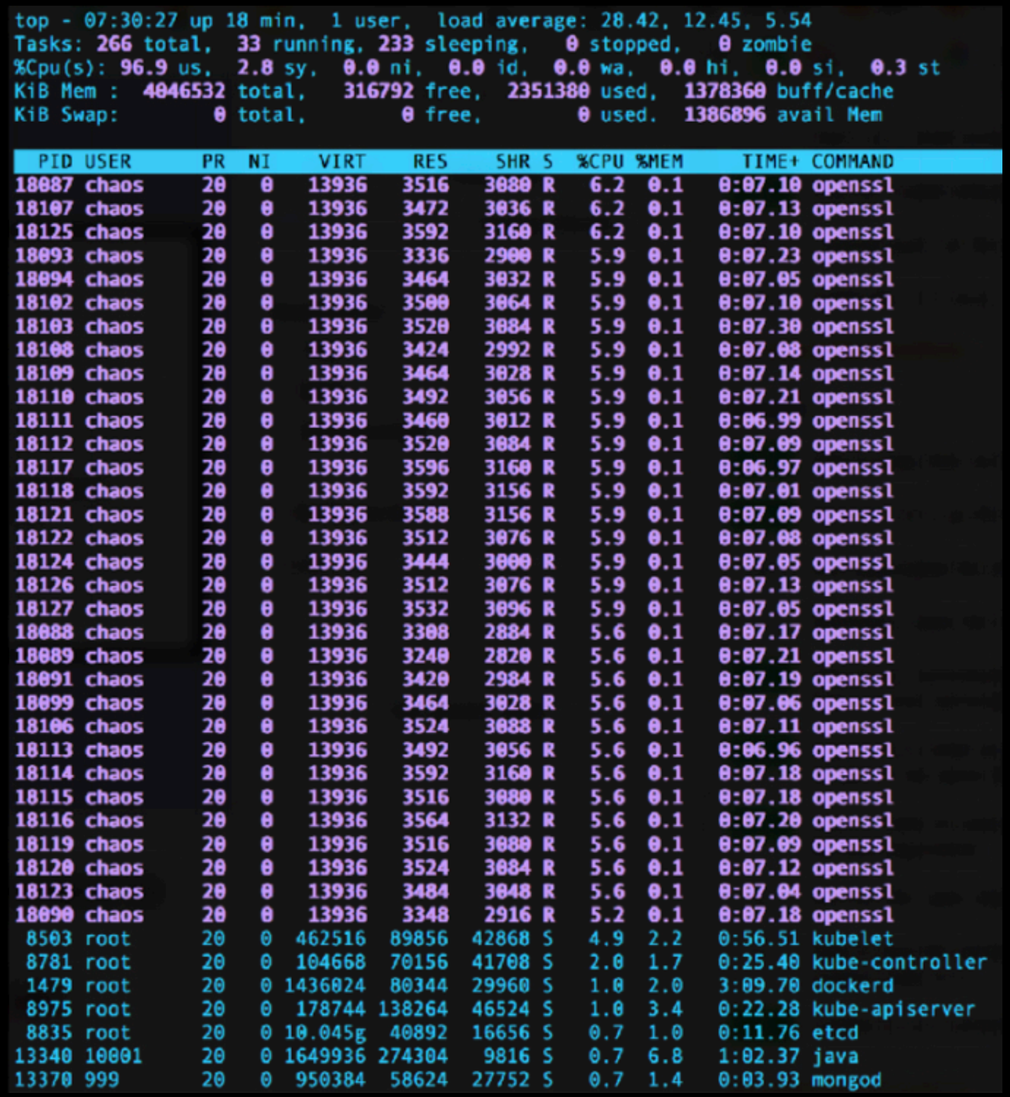
CPU CHAOS

LET'S CREATE A "KNOWN-KNOWN" EXPERIMENT

```
su - chaos
cd scripts
ls
cat burncpu.sh
./burncpu.sh
```

https://github.com/tammybutow/chaosengineeringbootcamp

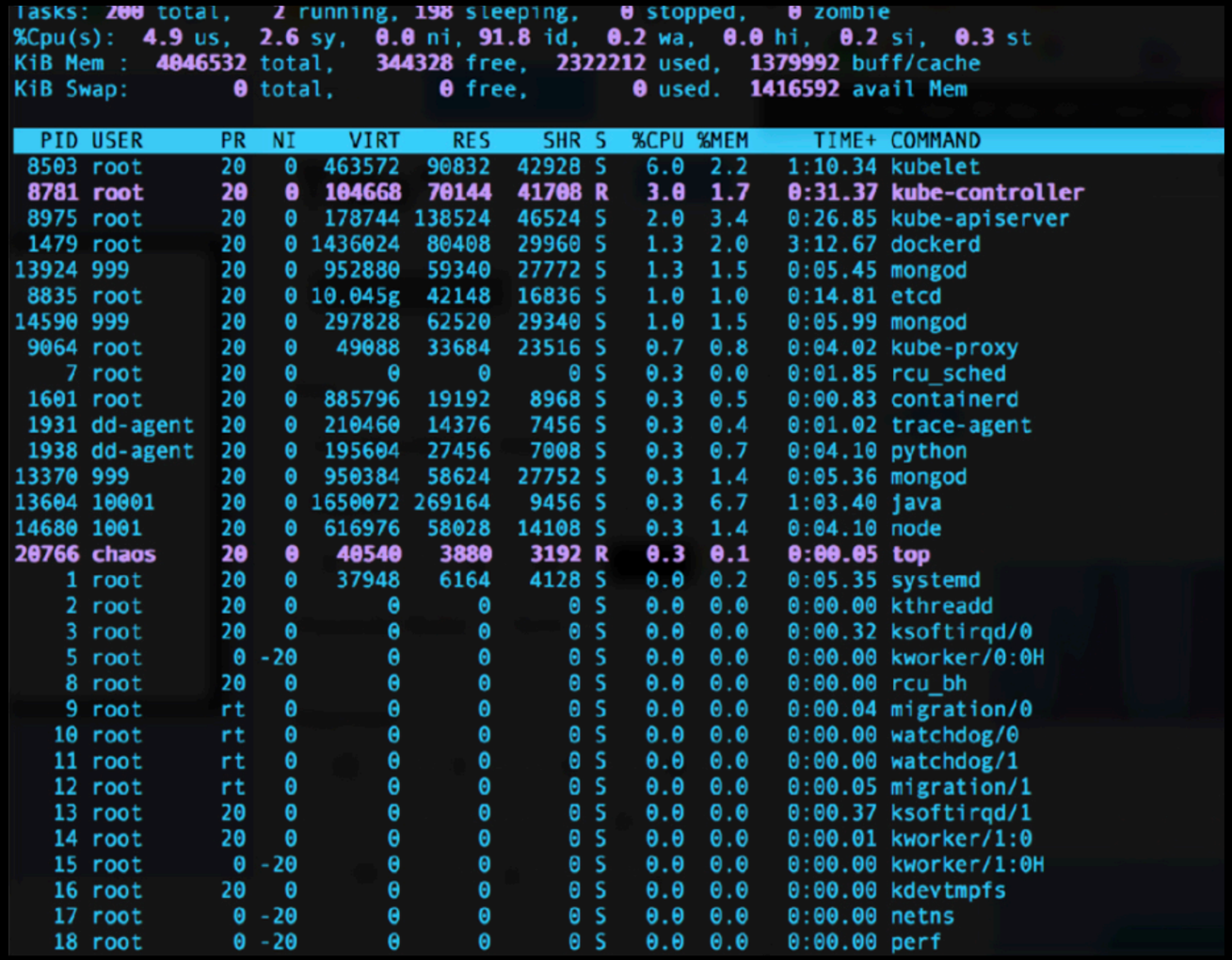
CHAOS IN TOP


top - 07:30:27 up 18 min, 1 user, load average: 28.42, 12.45, 5.54
Tasks: 266 total, 33 running, 233 sleeping, 0 stopped, 0 zombie
%Cpu(s): 96.9 us, 2.8 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.3 st
KiB Mem : 4046532 total, 316792 free, 2351380 used, 1378360 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 1386896 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
18087 chaos 20 0 13936 3516 3080 R 6.2 0.1 0:07.10 openssl
18107 chaos 20 0 13936 3472 3036 R 6.2 0.1 0:07.13 openssl
18125 chaos 20 0 13936 3592 3160 R 6.2 0.1 0:07.10 openssl
18093 chaos 20 0 13936 3336 2900 R 5.9 0.1 0:07.23 openssl
18094 chaos 20 0 13936 3464 3032 R 5.9 0.1 0:07.05 openssl
18102 chaos 20 0 13936 3500 3064 R 5.9 0.1 0:07.10 openssl
18103 chaos 20 0 13936 3520 3084 R 5.9 0.1 0:07.30 openssl
18108 chaos 20 0 13936 3424 2992 R 5.9 0.1 0:07.08 openssl
18109 chaos 20 0 13936 3464 3028 R 5.9 0.1 0:07.14 openssl
18110 chaos 20 0 13936 3492 3056 R 5.9 0.1 0:07.21 openssl
18111 chaos 20 0 13936 3460 3012 R 5.9 0.1 0:06.99 openssl
18112 chaos 20 0 13936 3520 3084 R 5.9 0.1 0:07.09 openssl
18117 chaos 20 0 13936 3596 3160 R 5.9 0.1 0:06.97 openssl
18118 chaos 20 0 13936 3592 3156 R 5.9 0.1 0:07.01 openssl
18121 chaos 20 0 13936 3588 3156 R 5.9 0.1 0:07.09 openssl
18122 chaos 20 0 13936 3512 3076 R 5.9 0.1 0:07.08 openssl
18124 chaos 20 0 13936 3444 3000 R 5.9 0.1 0:07.05 openssl
18126 chaos 20 0 13936 3512 3076 R 5.9 0.1 0:07.13 openssl
18127 chaos 20 0 13936 3532 3096 R 5.9 0.1 0:07.05 openssl
18088 chaos 20 0 13936 3308 2884 R 5.6 0.1 0:07.17 openssl
18089 chaos 20 0 13936 3240 2820 R 5.6 0.1 0:07.21 openssl
18091 chaos 20 0 13936 3420 2984 R 5.6 0.1 0:07.19 openssl
18099 chaos 20 0 13936 3464 3028 R 5.6 0.1 0:07.06 openssl
18106 chaos 20 0 13936 3524 3088 R 5.6 0.1 0:07.11 openssl
18113 chaos 20 0 13936 3492 3056 R 5.6 0.1 0:06.96 openssl
18114 chaos 20 0 13936 3592 3160 R 5.6 0.1 0:07.18 openssl
18115 chaos 20 0 13936 3516 3080 R 5.6 0.1 0:07.18 openssl
18116 chaos 20 0 13936 3564 3132 R 5.6 0.1 0:07.20 openssl
18119 chaos 20 0 13936 3516 3080 R 5.6 0.1 0:07.09 openssl
18120 chaos 20 0 13936 3524 3084 R 5.6 0.1 0:07.12 openssl
18123 chaos 20 0 13936 3484 3048 R 5.6 0.1 0:07.04 openssl
18090 chaos 20 0 13936 3348 2916 R 5.2 0.1 0:07.18 openssl
8503 root 20 0 462516 89856 42868 S 4.9 2.2 0:56.51 kubelet
8781 root 20 0 104668 70156 41708 S 2.0 1.7 0:25.40 kube-controller
1479 root 20 0 1436824 80344 29960 S 1.0 2.0 3:09.70 dockerd
8975 root 20 0 178744 138264 46524 S 1.0 3.4 0:22.28 kube-apiserver
8835 root 20 0 10.045g 40892 16656 S 0.7 1.0 0:11.76 etcd
13340 10001 20 0 1649936 274304 9816 S 0.7 6.8 1:02.37 java
13370 999 20 0 950384 58624 27752 S 0.7 1.4 0:03.93 mongod

LET’S KILL THE CHAOS NOW

```
pkill -u chaos
```

NO MORE CHAOS IN TOP



DISK CHAOS

DISK CHAOS

```
root@ubuntu-s-1vcpu-1gb-sfo2-01:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            490M     0  490M   0% /dev
tmpfs           100M  3.1M   97M   4% /run
/dev/vda1        25G  1.1G   23G   5% /
tmpfs           497M     0  497M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           497M     0  497M   0% /sys/fs/cgroup
/dev/vda15      105M  3.4M  101M   4% /boot/efi
tmpfs           100M     0  100M   0% /run/user/0
root@ubuntu-s-1vcpu-1gb-sfo2-01:~#
```

MEMORY CHAOS

MEMORY CHAOS

FREE(1) User Commands FREE(1)

NAME

free - Display amount of free and used memory in the system

SYNOPSIS

free [options]

DESCRIPTION

free displays the total amount of free and used physical and swap memory in the system, as well as the buffers and caches used by the kernel. The information is gathered by parsing /proc/meminfo. The displayed columns are:

total Total installed memory (MemTotal and SwapTotal in /proc/meminfo)

used Used memory (calculated as **total** - **free** - **buffers** - **cache**)

free Unused memory (MemFree and SwapFree in /proc/meminfo)

shared Memory used (mostly) by tmpfs (Shmem in /proc/meminfo, available on kernels 2.6.32, displayed as zero if not available)

buffers

free -m

PART 3: STATE
CHAOS ENGINEERING

PROCESS CHAOS

PROCESS CHAOS

Ways to create process chaos on purpose:

- Kill one process
- Loop kill a process
- Spawn new processes
- Fork bomb

PROCESS CHAOS

PGREP(1) User Commands PGREP(1)

NAME

pgrep, pkill - look up or signal processes based on name and other attributes

SYNOPSIS

pgrep [options] pattern
pkill [options] pattern

DESCRIPTION

pgrep looks through the currently running processes and lists the process IDs which match the selection criteria to stdout. All the criteria have to match. For example,

```
$ pgrep -u root sshd
```

will only list the processes called **sshd** AND owned by **root**. On the other hand,

```
$ pgrep -u root,daemon
```

will list the processes owned by **root** OR **daemon**.

pkill -u chaos

SHUTDOWN CHAOS

SHUTDOWN CHAOS

```
SHUTDOWN(8)                         shutdown                         SHUTDOWN(8)

NAME
       shutdown - Halt, power-off or reboot the machine

SYNOPSIS
       shutdown [OPTIONS...] [TIME] [WALL...]

DESCRIPTION
       shutdown may be used to halt, power-off or reboot the machine.

       The first argument may be a time string (which is usually "now").
       Optionally, this may be followed by a wall message to be sent to all
       logged-in users before going down.

       The time string may either be in the format "hh:mm" for hour/minutes
       specifying the time to execute the shutdown at, specified in 24h clock
       format. Alternatively it may be in the syntax "+m" referring to the
       specified number of minutes m from now.  "now" is an alias for "+0", i.e.
       for triggering an immediate shutdown. If no time argument is specified,
       "+1" is implied.

       Note that to specify a wall message you must specify a time argument, too.
```

shutdown -h

WHAT ARE OTHER WAYS YOU CAN
TURN OFF A SERVER?

WHAT IF YOU WANT TO
TURN OFF EVERY SERVER
WHEN IT'S ONE WEEK OLD?

HALT, REBOOT & POWEROFF CHAOS

```
HALT(8)                          halt                          HALT(8)

NAME
       halt, poweroff, reboot - Halt, power-off or reboot the machine

SYNOPSIS
       halt [OPTIONS...]

       poweroff [OPTIONS...]

       reboot [OPTIONS...]

DESCRIPTION
       halt, poweroff, reboot may be used to halt, power-off or reboot the
       machine.

OPTIONS
       The following options are understood:

       --help
           Print a short help text and exit.

       --halt
           Halt the machine, regardless of which one of the three commands is
```

halt

WHAT ABOUT SHUTTING DOWN CONTAINERS AND K8'S PODS?

THE MANY WAYS TO KILL CONTAINERS

- Kill self
- Kill a container from the host
- Use one container to kill another
- Use one container to kills several containers
- Use several containers to kill several

The average lifespan of a container is 2.5 days
And they fail in many unexpected ways.

TIME TRAVEL CHAOS

TIME TRAVEL CHAOS AKA CLOCK SKEW

serverfault

Questions Tags

Clock skew on my VPS server

1

I am having a problem with the time on my VPS hosted at Rackspace. Its running Red Hat Enterprise Linux 5.4. Date commands gives a time which is 11 minutes ahead of the actual time.

I am running `ntpd` and my `/etc/ntp.conf` is as per the 2nd option given here

I also modified `/etc/sysconfig/ntpd` and changed `SYNC_HWCLOCK=yes` and rebooted the server.

None of this has helped so far. I contacted Rackspace and their answer was the time on the main host is OK.

Output of `ntpq -p` is:

```
root@webserver [/etc/sysconfig]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*ln1.dijudi.co   128.138.140.44   2 u    7   64  377   24.645  -673871  19.246
+ns1.your-site.c 72.8.140.240     3 u   24   64  377   20.666  -673874  24.355
+linode.aput.net 10.0.0.250       3 u   10   64  377   54.058  -673838  23.812
 LOCAL(0)        .LOCL.          10 l   40   64  377    0.000    0.000   0.001
root@webserver [/etc/sysconfig]#
```

I would appreciate any assistance on this.

ntpq

PART 4: NETWORK CHAOS ENGINEERING

BLACKHOLE CHAOS

BLACKHOLE CHAOS

ROUTE(8) Linux Programmer's Manual ROUTE(8)

NAME

route - show / manipulate the IP routing table

SYNOPSIS

route [-CFvnee]

route [-v] [-A family] **add** [-net|-host] target [**netmask** Nm] [**gw** Gw] [**met-ric** N] [**mss** M] [**window** W] [**irtt** I] [**reject**] [**mod**] [**dyn**] [**reinstate**] [[**dev**] If]

route [-v] [-A family] **del** [-net|-host] target [**gw** Gw] [**netmask** Nm] [**met-ric** N] [[**dev**] If]

route [-V] [--version] [-h] [--help]

DESCRIPTION

Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the **ifconfig**(8) program.

When the **add** or **del** options are used, **route** modifies the routing tables. Without these options, **route** displays the current contents of the routing

ip route show

DNS CHAOS

DNS CHAOS

2016 Dyn cyberattack

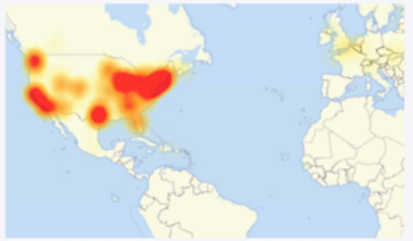
From Wikipedia, the free encyclopedia

The **2016 Dyn cyberattack** took place on October 21, 2016, and involved multiple distributed denial-of-service attacks (DDoS attacks) targeting systems operated by Domain Name System (DNS) provider Dyn, which caused major Internet platforms and services to be unavailable to large swathes of users in Europe and North America.[2][3] The groups Anonymous and New World Hackers claimed responsibility for the attack, but scant evidence was provided.[4]

As a DNS provider, Dyn provides to end-users the service of mapping an Internet domain name—when, for instance, entered into a web browser—to its corresponding IP address. The distributed denial-of-service (DDoS) attack was accomplished through a large number of DNS lookup requests from tens of millions of IP addresses.[5] The activities are believed to have been executed through a botnet consisting of a large number of Internet-connected devices—such as printers, IP cameras, residential gateways and baby monitors—that had been infected with the Mirai malware.

Contents [hide]

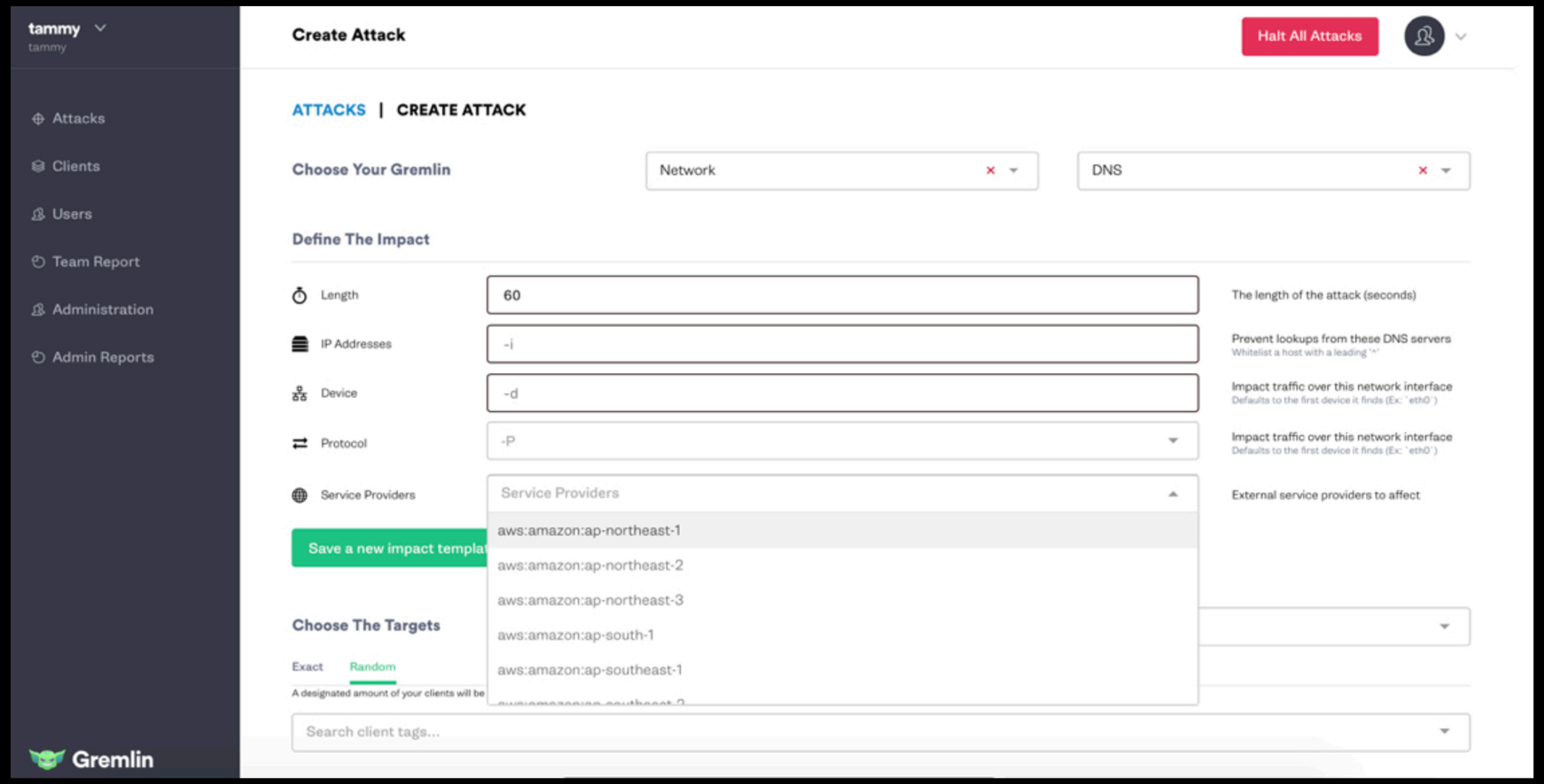
Dyn cyberattack



Map of areas most affected by attack, 16:45 UTC, 21 October 2016.[1]

Date	October 21, 2016
Time	12:10 – 14:20 UTC 16:50 – 18:11 UTC 21:00 – 23:11 UTC [*citation needed*][*needs update*]
Location	Europe and North America, especially the Eastern United States
Type	Distributed denial-of-service
Participants	Unknown

DNS CHAOS


tammy
tammy
Attacks
Clients
Users
Team Report
Administration
Admin Reports
Gremlin
Create Attack
Halt All Attacks
ATTACKS | CREATE ATTACK
Choose Your Gremlin
Network
DNS
Define The Impact
Length
60
The length of the attack (seconds)
IP Addresses
-i
Prevent lookups from these DNS servers
Device
-d
Impact traffic over this network interface
Protocol
-P
Impact traffic over this network interface
Service Providers
Service Providers
External service providers to affect
aws:amazon:ap-northeast-1
aws:amazon:ap-northeast-2
aws:amazon:ap-northeast-3
aws:amazon:ap-south-1
aws:amazon:ap-southeast-1
Choose The Targets
Exact
Random
Search client tags...

LATENCY CHAOS

LATENCY CHAOS

```
                        My traceroute  [v0.86]
ubuntu-s-1vcpu-1gb-sfo2-01 (0.0.0.0)                     Sat Jun  9 06:46:26 2018
Resolver: Received error response 2. (server failure)er of fields   quit
                                        Packets               Pings
 Host                                 Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 206.189.208.253                    0.0%    25    0.3   1.7   0.3  18.7   3.7
 2. 138.197.249.152                    0.0%    25    0.4   0.5   0.4   1.3   0.0
 3. 206.41.106.63                      0.0%    25    1.9   1.2   1.0   2.1   0.0
 4. 108.170.242.241                    0.0%    24    2.6   2.6   2.4   3.9   0.2
 5. 108.170.237.147                    0.0%    24    1.5   1.5   1.4   1.7   0.0
 6. nuq04s29-in-f14.1e100.net          0.0%    24    1.6   1.5   1.4   1.7   0.0
```

mtr google.com

PACKET LOSS CHAOS

PACKET LOSS CHAOS

Package: wireshark (2.4.5-1) [**universe**]

network traffic analyzer - meta-package

Other Packages Related to wireshark

● depends ◆ recommends ■ suggests • enhances

- wireshark-qt
  network traffic analyzer - Qt version
  or wireshark-gtk
  network traffic analyzer - GTK+ version

Download wireshark

Architecture	Package Size	Installed Size	Files
amd64	4.4 kB	54.0 kB	[list of files]
arm64	4.4 kB	54.0 kB	[list of files]
armhf	4.4 kB	54.0 kB	[list of files]
i386	4.4 kB	54.0 kB	[list of files]
ppc64el	4.4 kB	54.0 kB	[list of files]
s390x	4.4 kB	54.0 kB	[list of files]

Links for wireshark

Ubuntu Resources:

- Bug Reports
- Ubuntu Changelog
- Copyright File

Download Source Package wireshark:

- [wireshark_2.4.5-1.dsc]
- [wireshark_2.4.5.orig.tar.xz]
- [wireshark_2.4.5-1.debian.tar.xz]

Maintainer:

- Ubuntu MOTU Developers (Mail Archive)

PART 5: COMPLEX OUTAGES

We can combine different types of chaos engineering experiments to reproduce complicated outages.

Reproducing outages gives you confidence you can handle it if/when it happens again.

Let’s go back in time to look at some of the worst outage stories that kicked off the introduction of chaos engineering.

DROPBOX'S WORST OUTAGE EVER

Some master-replica pairs were impacted which resulted in the site going down.

https://blogs.dropbox.com/tech/2014/01/outage-post-mortem/

UBER'S DATABASE OUTAGE

1. Master log replication to S3 failed
2. Logs backed up on the primary
3. Alerts fired to engineer but they are ignored
4. Disk fills up on database primary
5. Engineer deletes unarchived WAL files
6. Error in config prevents promotion

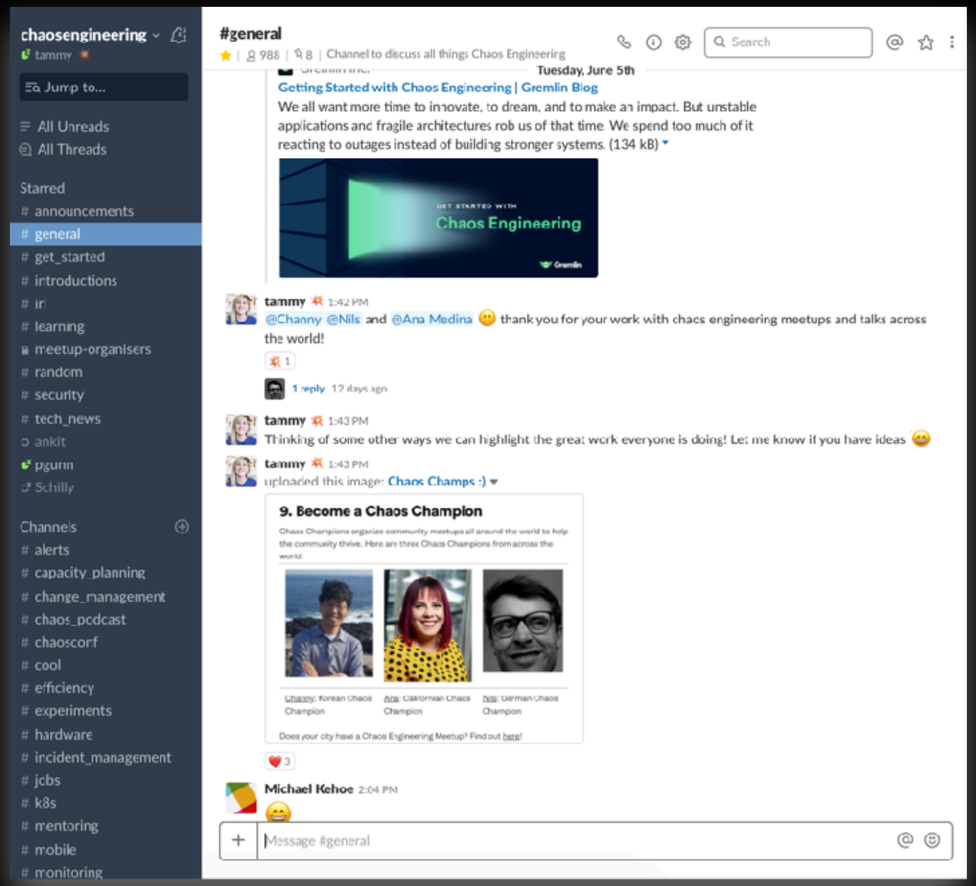
— Matt Ranney, Uber, 2015

OUTAGES HAPPEN.

THERE ARE MANY MORE OUTAGES
YOU CAN READ ABOUT HERE:
https://github.com/danluu/post-mortems

HOW CAN YOU CONTINUE YOUR CHAOS ENGINEERING JOURNEY?

JOIN THE CHAOS SLACK

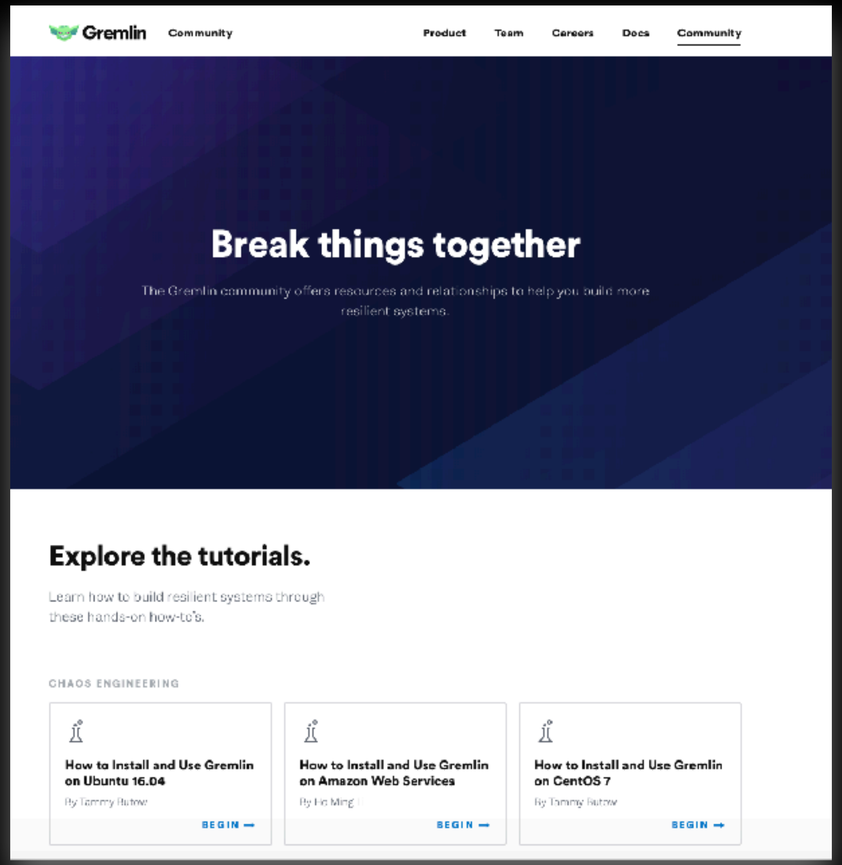


GREMLIN.COM/SLACK

@tammybütow

#QCONNYC

LEARN WITH THE GREMLIN COMMUNITY



GREMLIN.COM/COMMUNITY

@tammybütow

#QCONNYC

THE FIRST CHAOS ENGINEERING CONFERENCE!

CHAOSCONF.IO

@tammybütow

#QCONNYC

THANK YOU

QCON NYC

@tammybütow

#CHAOSENGINEERING