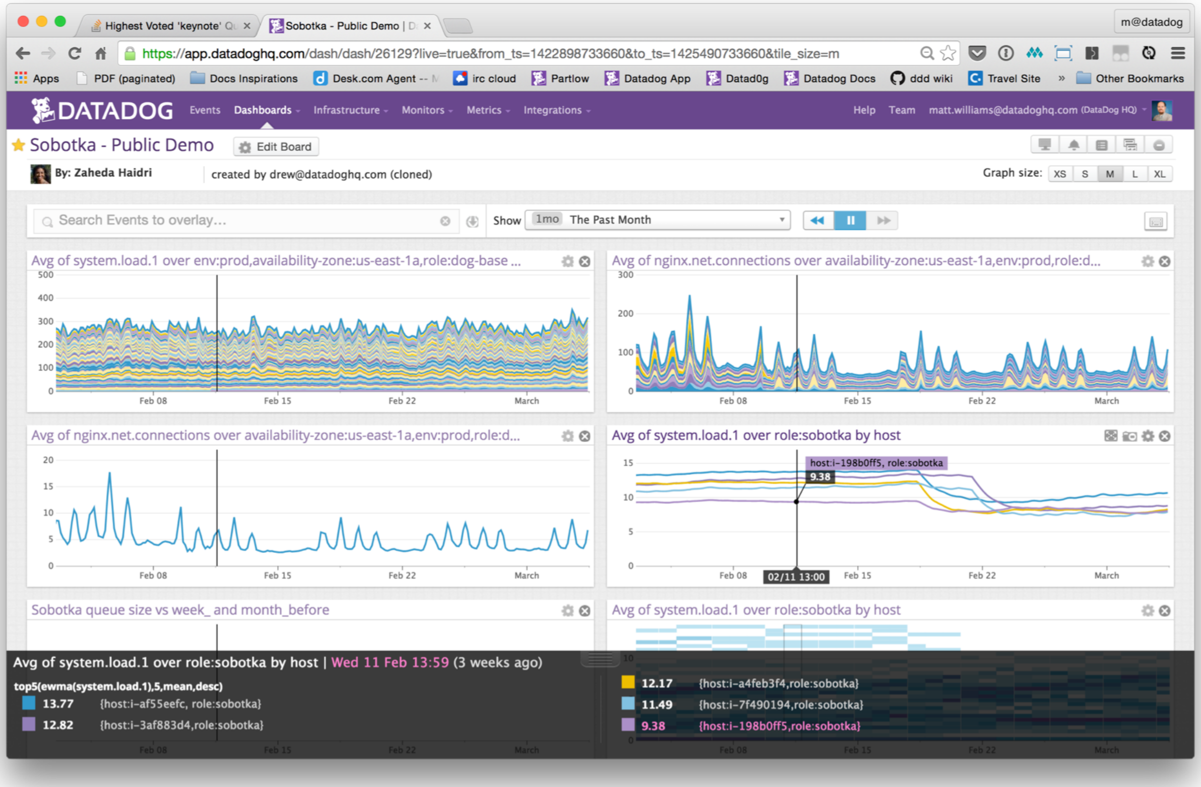
Using Luigi to build data pipelines…

…that won’t wake you at 3am

matt williams
evangelist @ datadog
@technovangelist
mattw@datadoghq.com

Who is Datadog



Sobotka - Public Demo

By: Zaheda Haidri

created by drew@datadoghq.com (cloned)

Avg of system.load.1 over env:prod,availability-zone:us-east-1a,role:dog-base ...

Avg of nginx.net.connections over availability-zone:us-east-1a,env:prod,role:d...

Avg of nginx.net.connections over availability-zone:us-east-1a,env:prod,role:d...

Avg of system.load.1 over role:sobotka by host

Sobotka queue size vs week_ and month_before

Avg of system.load.1 over role:sobotka by host

Avg of system.load.1 over role:sobotka by host | Wed 11 Feb 13:59 (3 weeks ago)

top5(ewma(system.load.1),5,mean,desc)

```
13.77   {host:i-af55eefc, role:sobotka}
12.82   {host:i-3af883d4,role:sobotka}
12.17   {host:i-a4feb3f4,role:sobotka}
11.49   {host:i-7f490194,role:sobotka}
9.38    {host:i-198b0ff5,role:sobotka}
```

DATADOG

How much data do we deal with?

- 200 BILLION datapoints per day
- 100’s TB of data
- 100’s of new trials each day

What is Luigi

- Character from a series of games from Nintendo
- Taller and thinner than his brother, Mario
- Is a Plumber by trade
- Nervous and timid but good natured

http://en.wikipedia.org/wiki/Luigi

What is Luigi?

- Python module to help build complex pipelines
  - dependency resolution
  - workflow management
  - visualization
  - hadoop support built in
- Created by Spotify
- Initial commit on github/spotify/luigi on Nov 17, 2011
  - committed by erikbern (no longer at spotify as of Feb 2015)
  - 2010 commits

What is Luigi?

The initial problems

1. select artist_id, count(1) from user_activities
   where play_seconds > 30 group by artist_id;

2. cron for lots of jobs?

What is Luigi?

- According to Erik Bernhardsson:

Doesn't help you with the code, that's what Scalding *(scala)*, Pig, or anything else is good at.

It helps you with the **plumbing** of connecting lots of tasks into complicated pipelines, especially if those tasks run on **Hadoop**.

http://erikbern.com/2014/12/17/luigi-presentation-nyc-data-science-dec-16-2014/

Luigi doesn't replace Hadoop, Scalding, Pig, Hive, Redshift. It orchestrates them.

What is Luigi?

The core beliefs:

1. should remove all boiler plate
2. be as general as possible
3. be easy to go from test to prod

Hello Luigi – The Concepts

- Tasks
  - Units of work that produce Outputs
  - Can depend on one or more other tasks
  - Is only run if all dependents are complete
  - Are idempotent

- Entirely code-based
  - Most other tools are gui-based or declarative and don't offer any abstraction
    - with code you can build anything you want

Luigi Task

```
class MyTask(luigi.Task):
  def output(self):
    pass

  def requires(self):
    pass

  def run(self)
    pass


luigi.run(main_task_cls=MyTask)
```

Luigi Task

```
class AggregateArtists(luigi.Task):
    date_interval = luigi.DateIntervalParameter()

    def output(self):
        return luigi.LocalTarget("data/artist_streams_%s.tsv" % self.date_interval)

    def requires(self):
        return [Streams(date) for date in self.date_interval]

    def run(self):
        artist_count = defaultdict(int)

        for input in self.input():
            with input.open('r') as in_file:
                for line in in_file:
                    timestamp, artist, track = line.strip().split()
                    artist_count[artist] += 1

        with self.output().open('w') as out_file:
            for artist, count in artist_count.iteritems():
                print >> out_file, artist, count
```

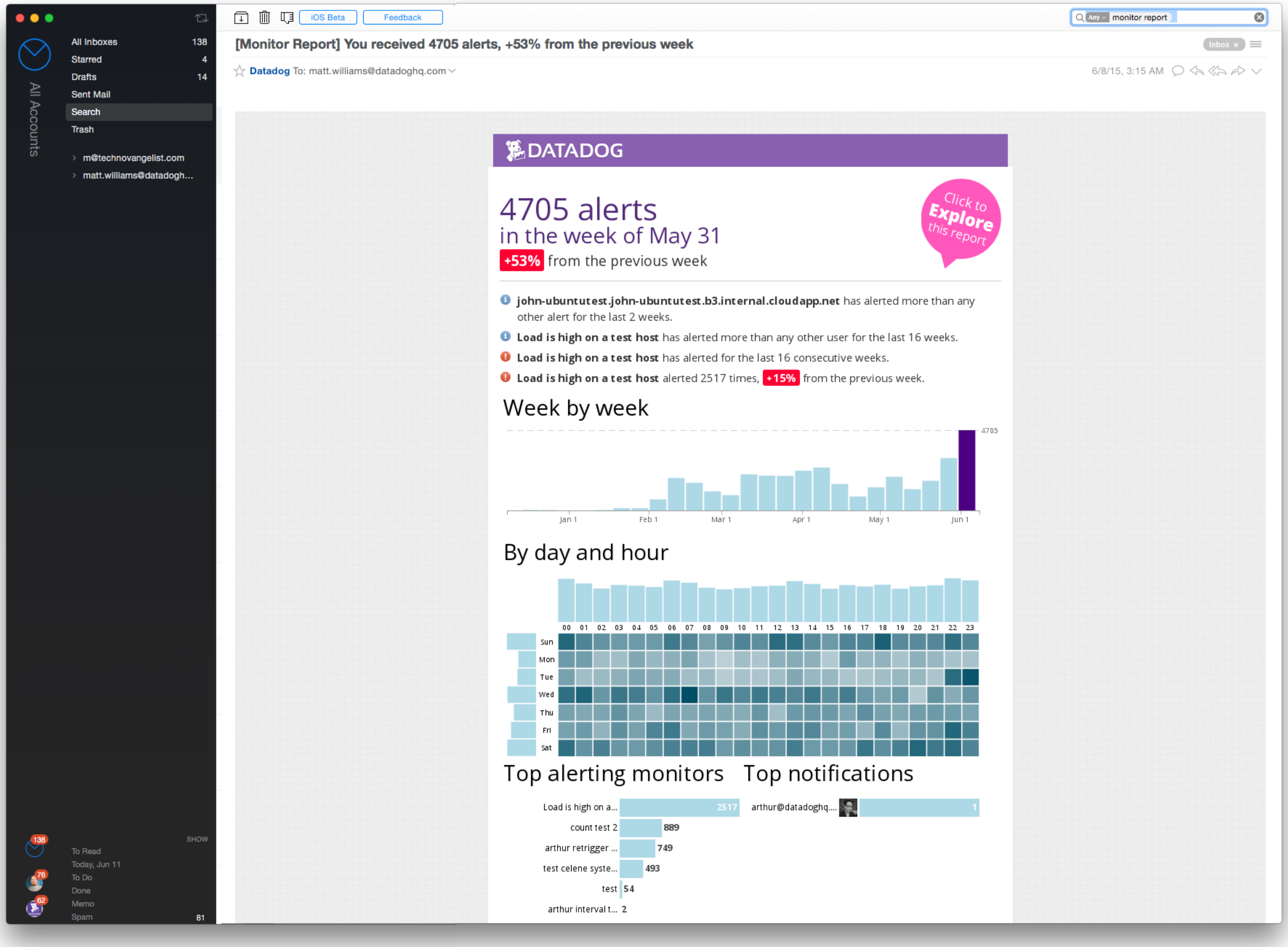
http://luigi.readthedocs.org/en/stable/example_top_artists.html

Luigi Task

```
class MyTask(luigi.Task):
  def output(self):
    return S3Target("%s/%s" % (s3_dest,end_data_date)

  def requires(self):
    return [SessionizeWebLogs(env,extract_date,start_data_date)]

  def run(self)
    curr_iteration = 0
      while curr_iteration < self.num_retries:
        try:
          self._run()
          break
        except:
          logger.exception("Iter %s of %s Failed." % (curr_iteration+1,num_retries))
          if curr_iteration < self.num_retries - 1:
            curr_iteration += 1
            time.sleep(self.sleep_time_between_retries_seconds)
          else:
            logger.error("Failed too many times.  Aborting.")
            raise
```

Why are we using it

- Understand trial account –> paid account
  - Paid account flow
  - Trends
- Free accounts >= Free services ?
- Interesting trials
- Usage by big customer
- Email reports



[Monitor Report] You received 4705 alerts, +53% from the previous week

Datadog To: matt.williams@datadoghq.com

6/8/15, 3:15 AM

DATADOG

4705 alerts
in the week of May 31

+53% from the previous week

- **john-ubuntutest.john-ubuntutest.b3.internal.cloudapp.net** has alerted more than any other alert for the last 2 weeks.
- **Load is high on a test host** has alerted more than any other user for the last 16 weeks.
- **Load is high on a test host** has alerted for the last 16 consecutive weeks.
- **Load is high on a test host** alerted 2517 times, +15% from the previous week.

Week by week

By day and hour

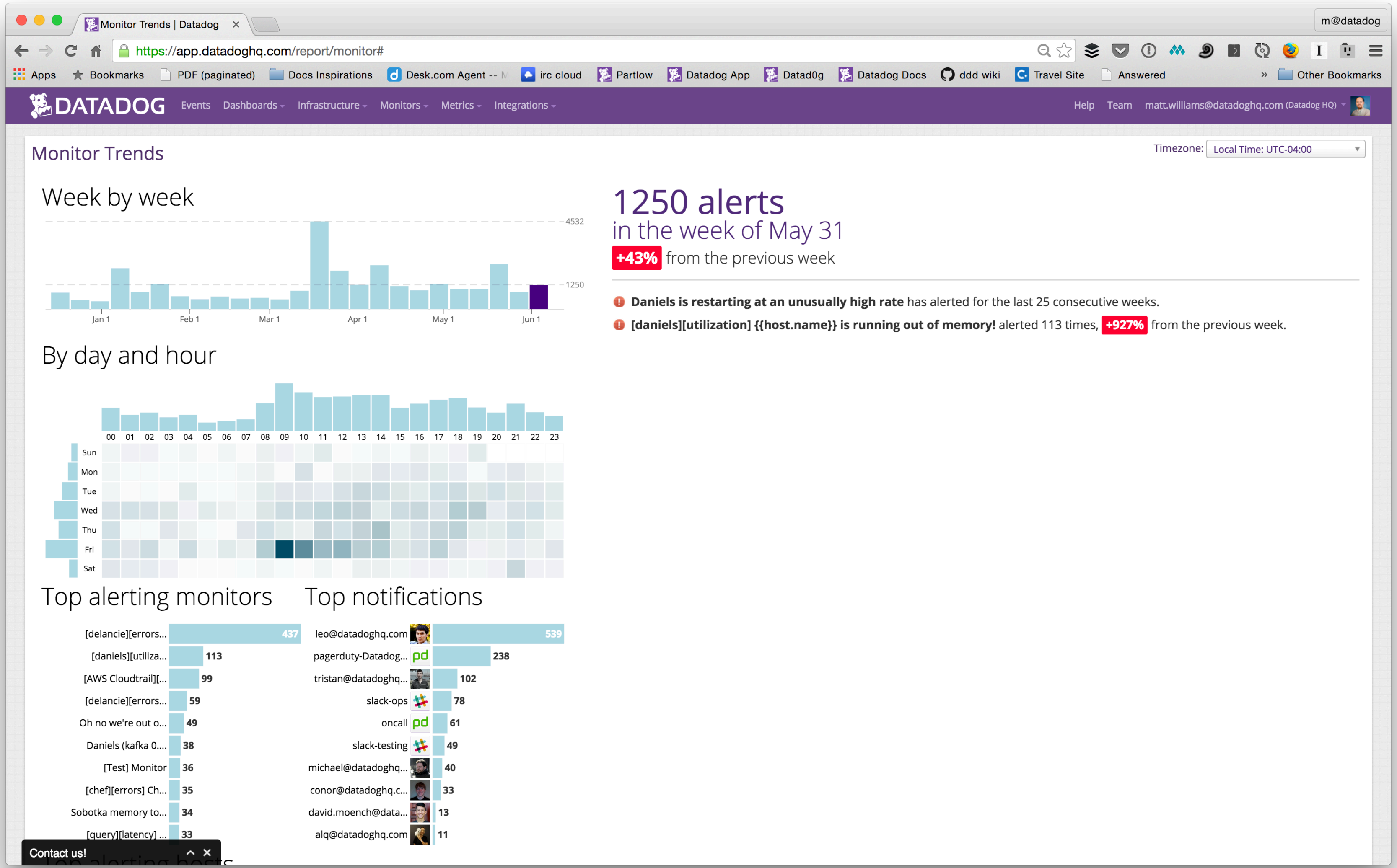
Top alerting monitors

Load is high on a...	2517
count test 2	889
arthur retrigger ...	749
test celene syste...	493
test	54
arthur interval t...	2

Top notifications

arthur@datadoghq....	1

DATADOG



Monitor Trends

Timezone: Local Time: UTC-04:00

Week by week

By day and hour

1250 alerts

in the week of May 31

+43% from the previous week

- **Daniels is restarting at an unusually high rate** has alerted for the last 25 consecutive weeks.
- **[daniels][utilization] {{host.name}} is running out of memory!** alerted 113 times, **+927%** from the previous week.

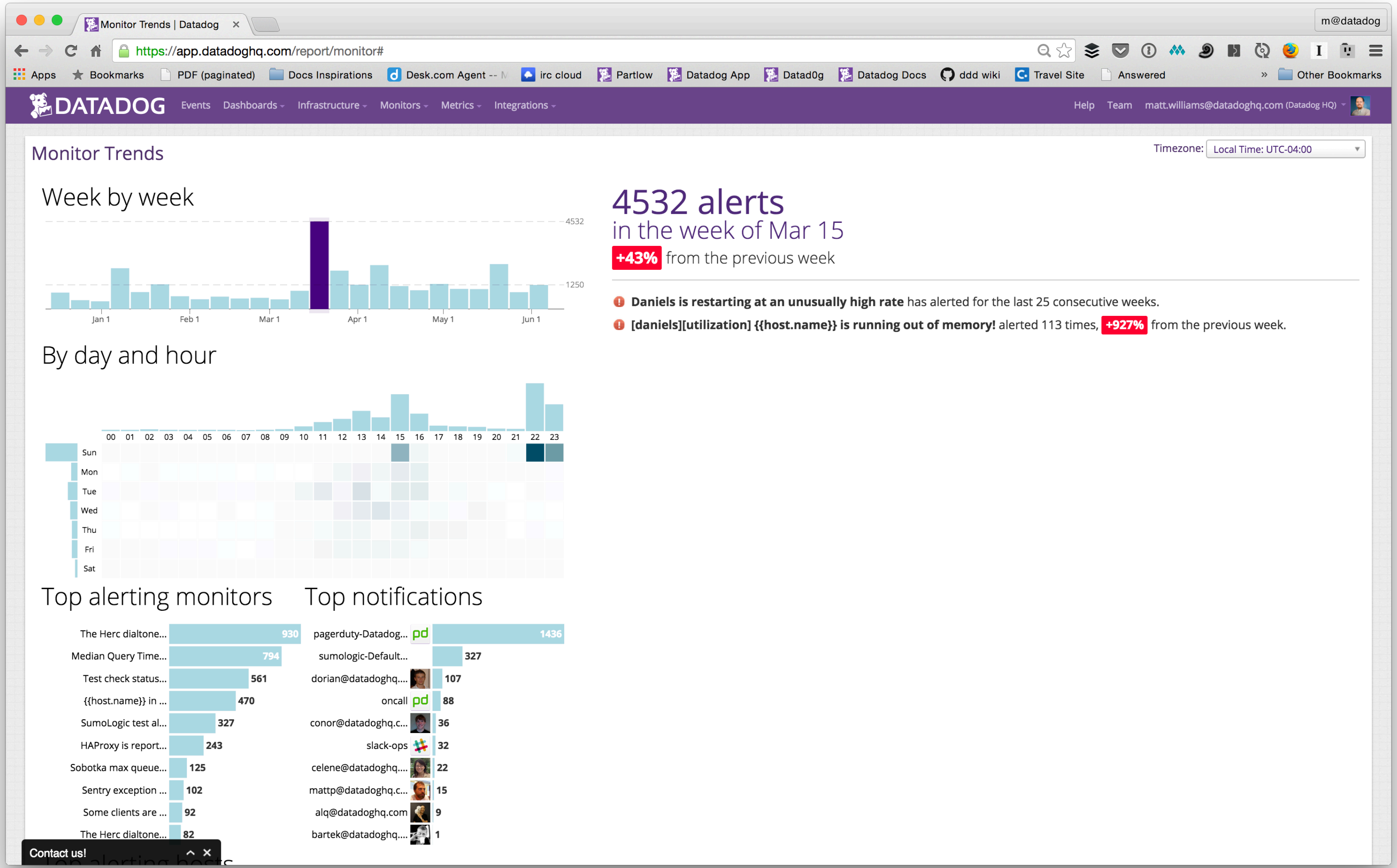
Top alerting monitors

[delancie][errors...	437
[daniels][utiliza...	113
[AWS Cloudtrail][...	99
[delancie][errors...	59
Oh no we're out o...	49
Daniels (kafka 0....	38
[Test] Monitor	36
[chef][errors] Ch...	35
Sobotka memory to...	34
[query][latency] ...	33

Top notifications

leo@datadoghq.com	539
pagerduty-Datadog...	238
tristan@datadoghq...	102
slack-ops	78
oncall	61
slack-testing	49
michael@datadoghq...	40
conor@datadoghq.c...	33
david.moench@data...	13
alq@datadoghq.com	11

Top alerting hosts



Monitor Trends

Timezone: Local Time: UTC-04:00

Week by week

By day and hour

Top alerting monitors

The Herc dialtone...	930
Median Query Time...	794
Test check status...	561
{{host.name}} in ...	470
SumoLogic test al...	327
HAProxy is report...	243
Sobotka max queue...	125
Sentry exception ...	102
Some clients are ...	92
The Herc dialtone...	82

Top notifications

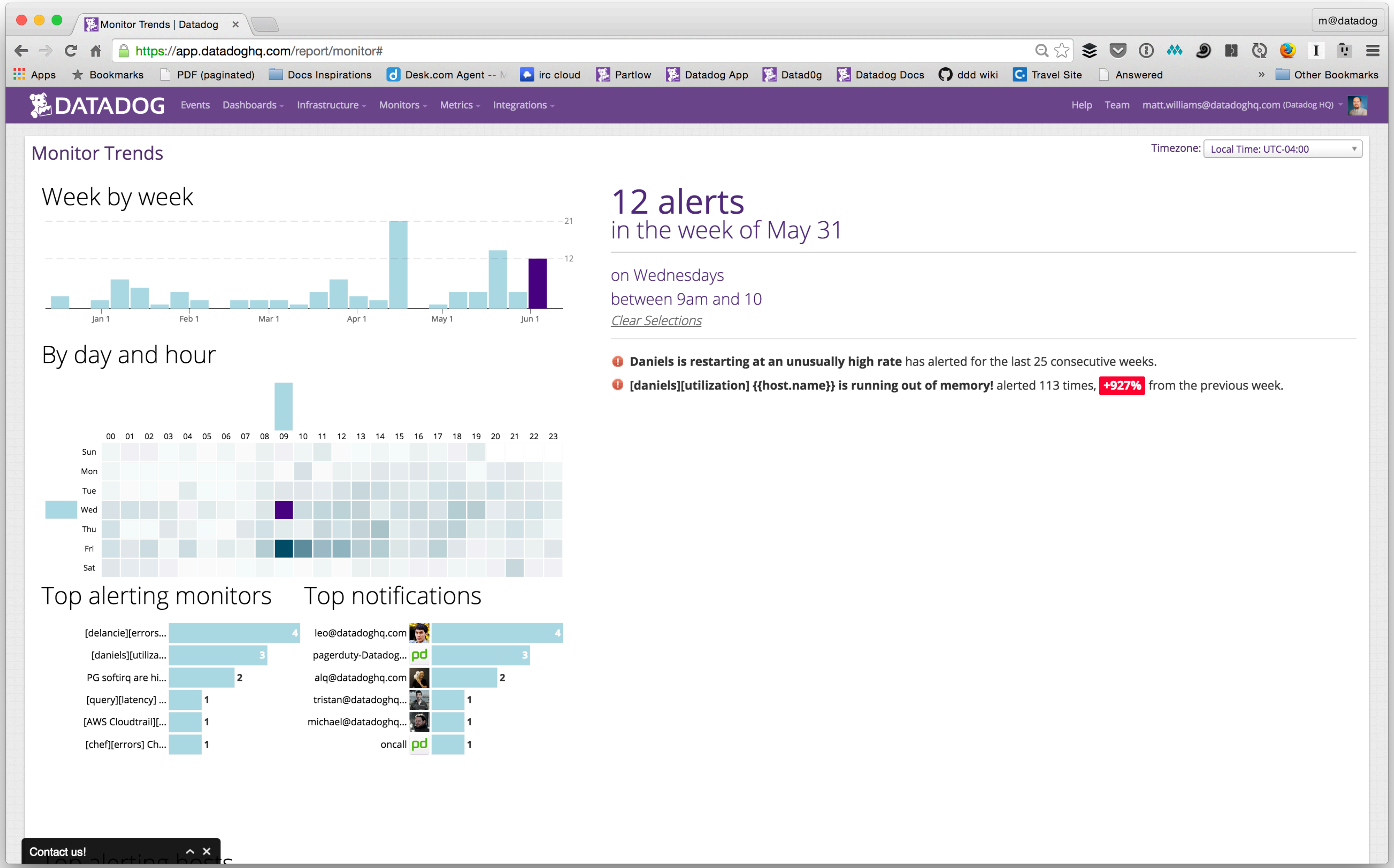
pagerduty-Datadog... pd	1436
sumologic-Default...	327
dorian@datadoghq....	107
oncall pd	88
conor@datadoghq.c...	36
slack-ops	32
celene@datadoghq....	22
mattp@datadoghq.c...	15
alq@datadoghq.com	9
bartek@datadoghq....	1

4532 alerts

in the week of Mar 15

+43% from the previous week

- **Daniels is restarting at an unusually high rate** has alerted for the last 25 consecutive weeks.
- **[daniels][utilization] {{host.name}} is running out of memory!** alerted 113 times, **+927%** from the previous week.



Monitor Trends

Timezone: Local Time: UTC-04:00

Week by week

By day and hour

Top alerting monitors

[delancie][errors...	4
[daniels][utiliza...	3
PG softirq are hi...	2
[query][latency] ...	1
[AWS Cloudtrail][...	1
[chef][errors] Ch...	1

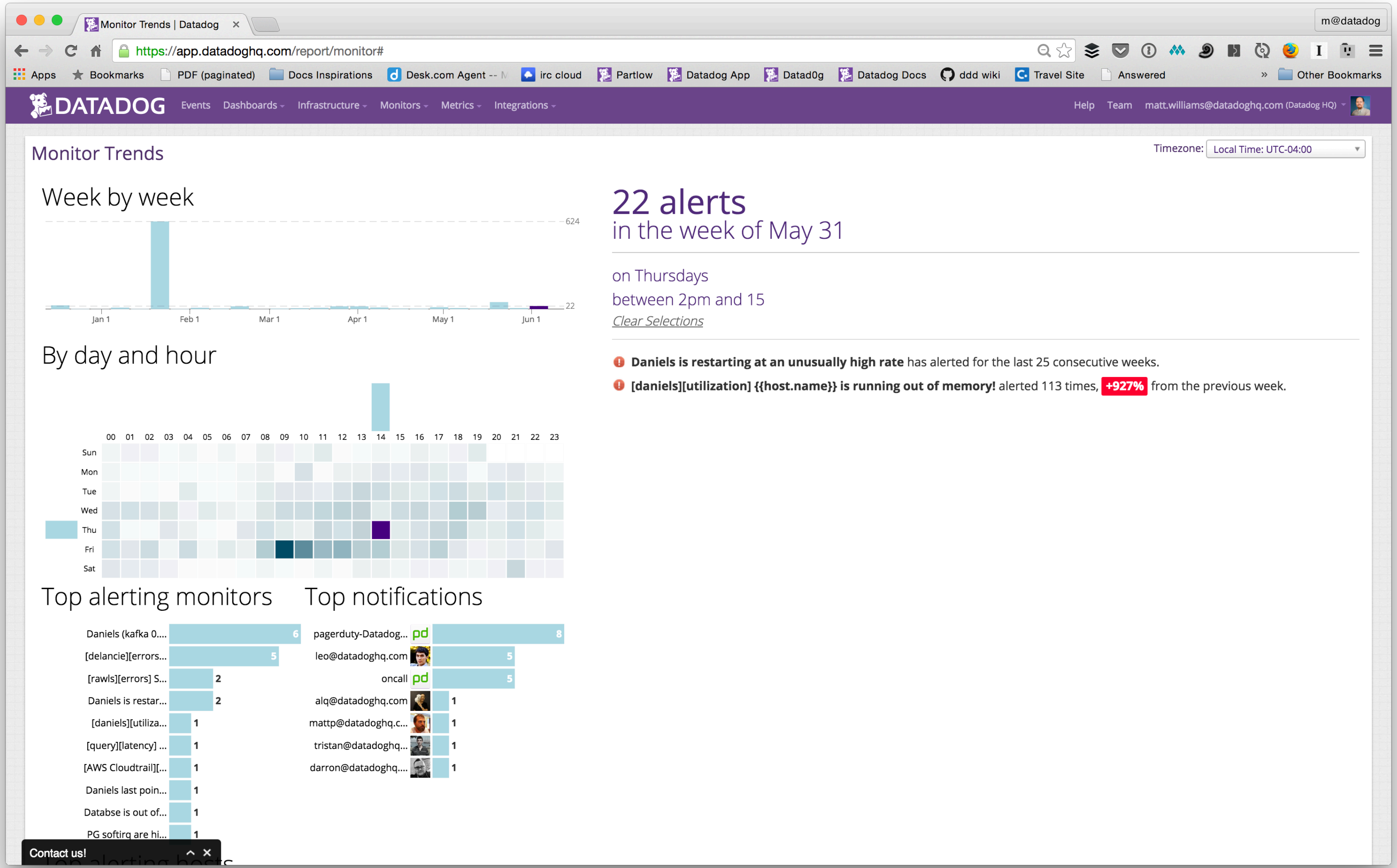
Top notifications

leo@datadoghq.com	4
pagerduty-Datadog... pd	3
alq@datadoghq.com	2
tristan@datadoghq...	1
michael@datadoghq...	1
oncall pd	1

12 alerts
in the week of May 31

on Wednesdays
between 9am and 10
Clear Selections

- **Daniels is restarting at an unusually high rate** has alerted for the last 25 consecutive weeks.
- **[daniels][utilization] {{host.name}} is running out of memory!** alerted 113 times, **+927%** from the previous week.



Monitor Trends

Timezone: Local Time: UTC-04:00

Week by week

By day and hour

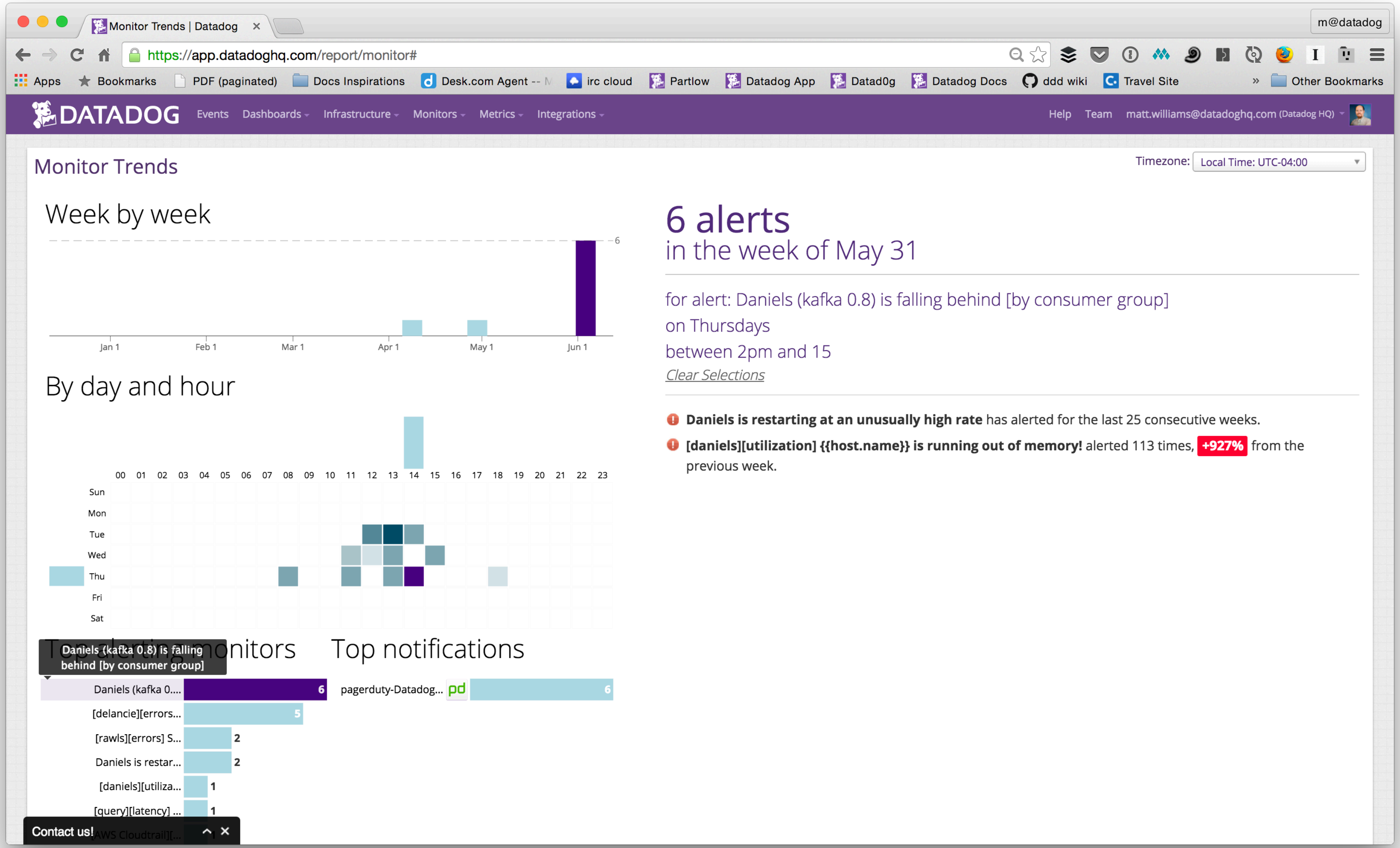
Top alerting monitors

Top notifications

22 alerts
in the week of May 31

on Thursdays
between 2pm and 15
Clear Selections

- **Daniels is restarting at an unusually high rate** has alerted for the last 25 consecutive weeks.
- **[daniels][utilization] {{host.name}} is running out of memory!** alerted 113 times, **+927%** from the previous week.



Monitor Trends

Week by week

By day and hour

Top alerting monitors

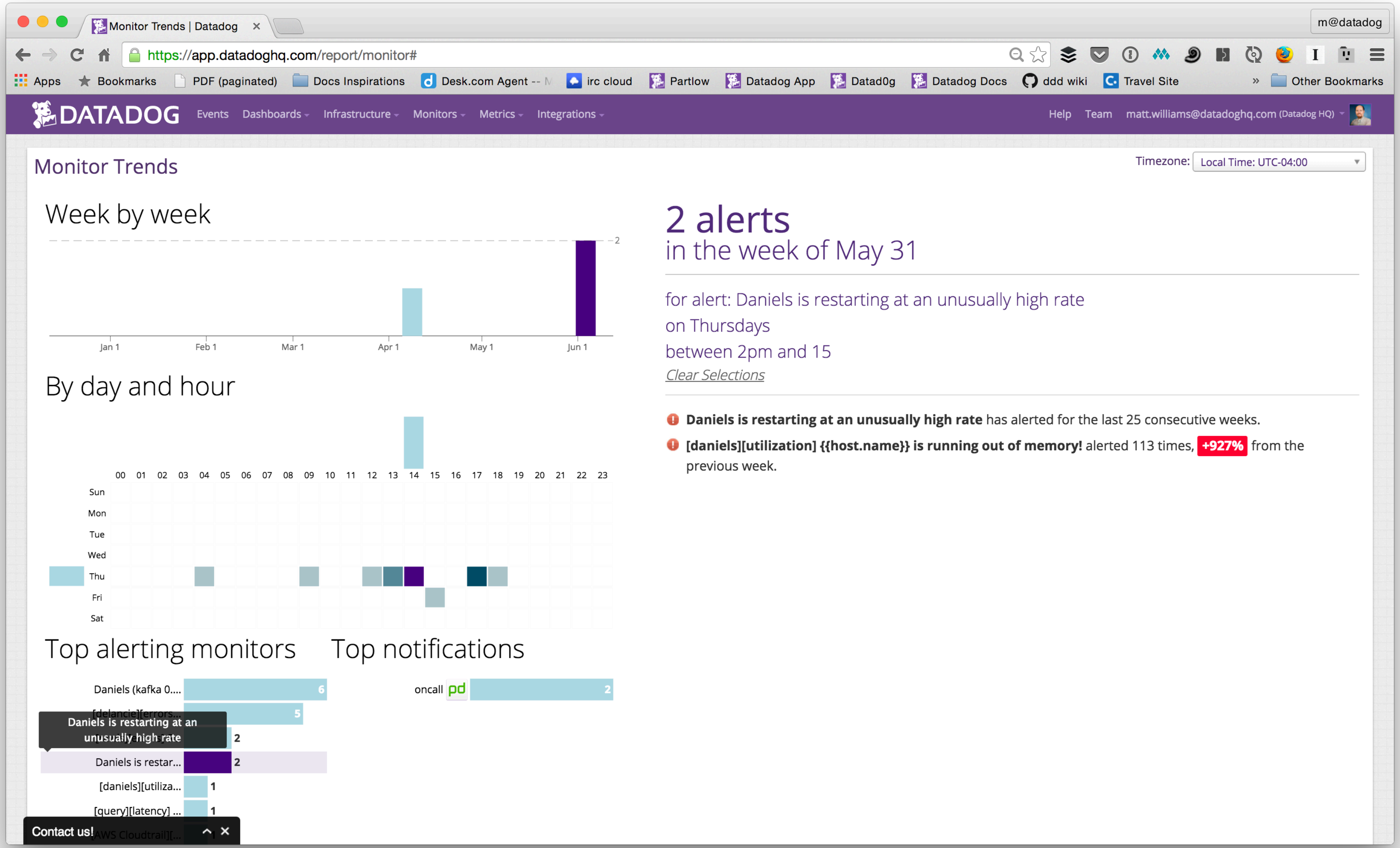
Top notifications

6 alerts
in the week of May 31

for alert: Daniels (kafka 0.8) is falling behind [by consumer group]
on Thursdays
between 2pm and 15

Clear Selections

- **Daniels is restarting at an unusually high rate** has alerted for the last 25 consecutive weeks.
- **[daniels][utilization] {{host.name}} is running out of memory!** alerted 113 times, **+927%** from the previous week.



Monitor Trends

Timezone: Local Time: UTC-04:00

Week by week

By day and hour

Top alerting monitors

Top notifications

2 alerts
in the week of May 31

for alert: Daniels is restarting at an unusually high rate
on Thursdays
between 2pm and 15

Clear Selections

- **Daniels is restarting at an unusually high rate** has alerted for the last 25 consecutive weeks.
- **[daniels][utilization] {{host.name}} is running out of memory!** alerted 113 times, +927% from the previous week.

Why are we using it

- Similar questions solved before with various solutions
    - Complex SQL queries
    - Shell scripts
        - Can’t easily be restarted (idempotency was rarely thought about)
        - Failure checking is manual

Lets look at how we use it in detail

Org-day

1. Get source data from S3
2. Generate a list of all orgs with new trials (100s)
3. Get metrics
4. Rollup metrics with lots of joins, groups, and flattens
5. Save that
6. Parse the application log files grouped by org
7. Get all org activity
8. Save to S3
9. Copy it all to Redshift

Lets look at how we use it in detail

Org-Trial-Metrics

1. Get the source data from S3
2. Calculate key trial metrics
   # of hosts, integrations, dashboards, metrics
3. Create target metrics
   Median hosts, integrations, dashboards, metrics, etc
4. Prep to push to Redshift, Salesforce
5. Push everything to Redshift (looker), S3, and Salesforce (sales to followup on)

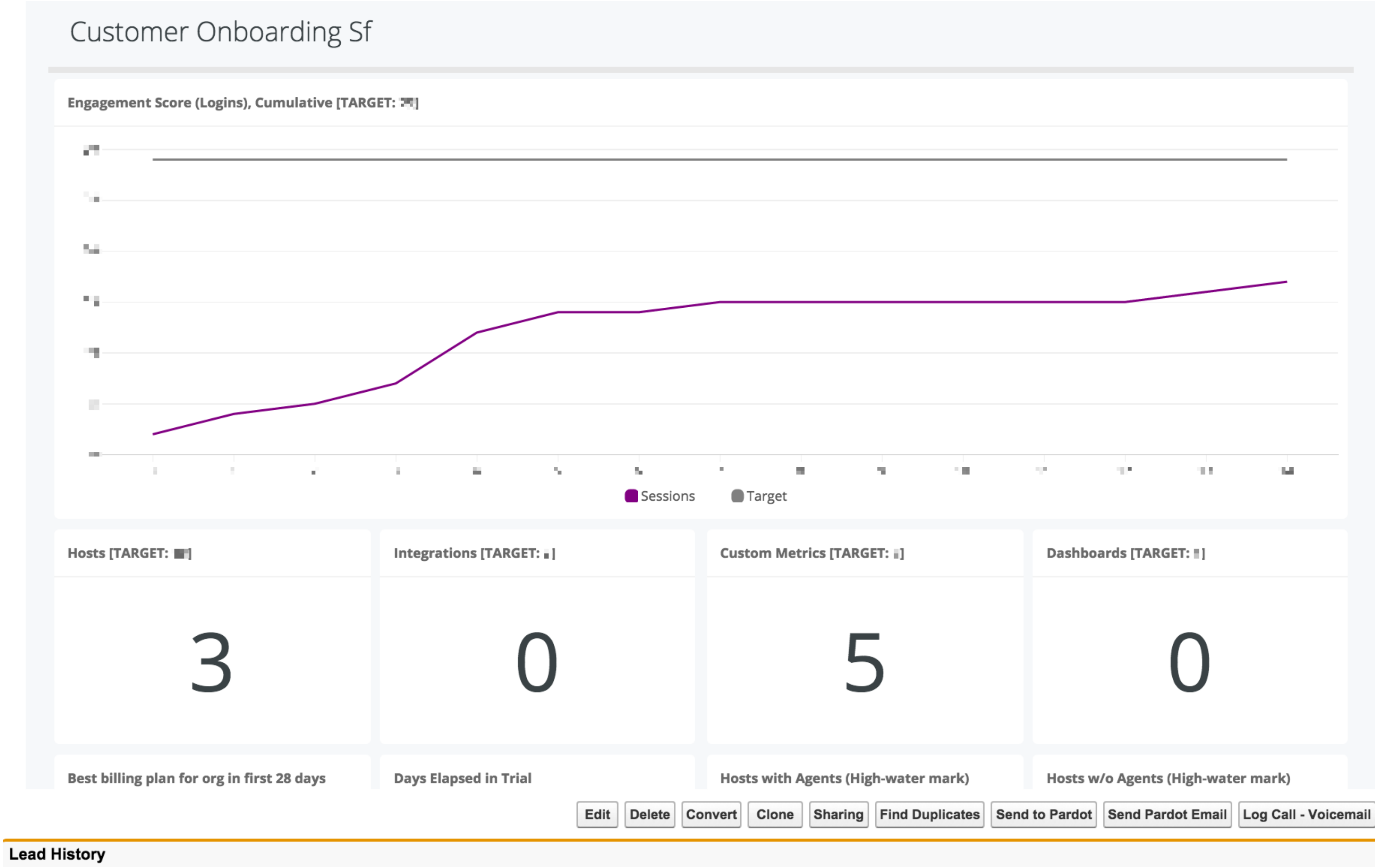
1 task in more detail

```
class CreateOrgTrialMetrics(MortarPigscriptTask):
    cluster_size = luigi.IntParameter(default=3)

    def requires(self):
        return [ S3PathTask(dd_utils.get_base_org_day_path(
                        self.env, self.version, self.data_date)) ]

    def script_output(self):
        return [ S3Target(dd_utils.get_base_org_trial_metrics_path_for_redshift(
                        self.env, self.version, self.data_date)),
                 S3Target(dd_utils.get_base_org_trial_metrics_path_for_salesforce(
                        self.env, self.version, self.data_date)),
                 S3Target(dd_utils.get_base_org_trial_metrics_path(
                        self.env, self.version, self.data_date)) ]

    def output(self):
        return self.script_output()

    def script(self):
        return 'org-trial-metrics/010-generate_org_trial_metrics.pig'
```

DATADOG

the pig file it relies on

```
import ....

org_day_data = cached_org_day('*');

conversion_period_data = filter org_day_data
    by org_day < ($TRIAL_PERIOD_DAYS + $EXTRA_CONVERSION_PERIOD_DAYS)
    and ToDate(metric_date) <= ToDate('$DATA_DATE', 'yyyy-MM-dd');

current_final_billing_plans = foreach (group conversion_period_data by org_id) {
    decreasing_days = order conversion_period_data by org_day DESC;
    cf_day = limit decreasing_days 1;

    generate group                                    as org_id,
             FLATTEN(cf_day.org_billing_plan_id)   as org_billing_plan_id,
             FLATTEN(cf_day.org_billing_plan_name) as org_billing_plan_name;
};

days_in_trial =  filter conversion_period_data
                         by org_day <= $TRIAL_PERIOD_DAYS;

org_trial_data =  group days_in_trial by org_id;
org_data = join org_trial_data by group, current_final_billing_plans by org_id;

results =  foreach org_data {
    decreasing_days = order org_trial_data::days_in_trial by org_day DESC;
    cf_day = limit decreasing_days 1;

    generate group                                        as org_id,
             FLATTEN(cf_day.org_name)                     as org_name,
             ToDate('$DATA_DATE', 'yyyy-MM-dd')       as generated_date,
```



Customer Onboarding Sf

Engagement Score (Logins), Cumulative [TARGET: [redacted]]

Hosts [TARGET: [redacted]]	Integrations [TARGET: [redacted]]	Custom Metrics [TARGET: [redacted]]	Dashboards [TARGET: [redacted]]
3	0	5	0

Best billing plan for org in first 28 days

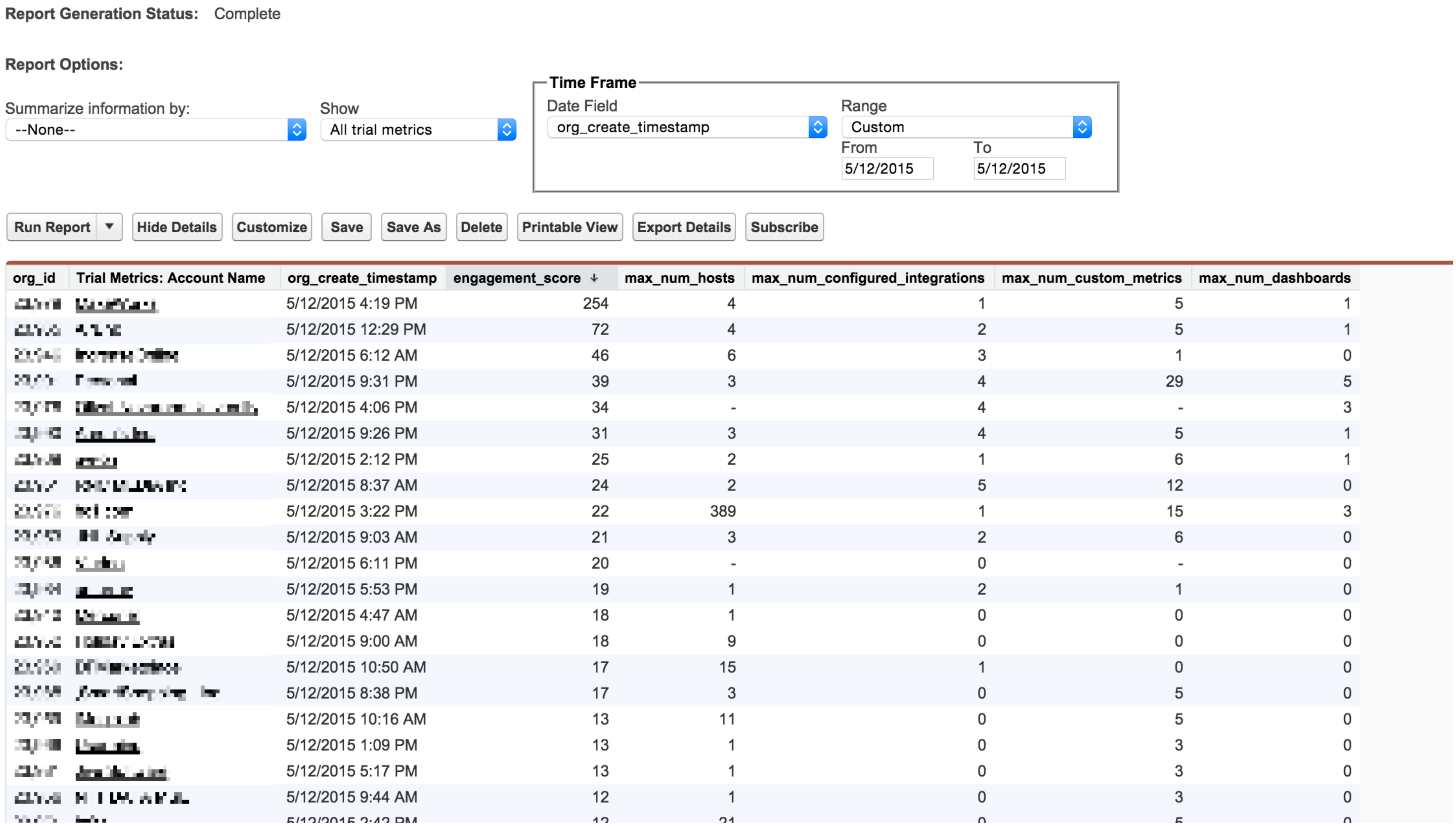
Days Elapsed in Trial

Hosts with Agents (High-water mark)

Hosts w/o Agents (High-water mark)

Lead History

DATADOG



Report Generation Status: Complete

Report Options:

Summarize information by: --None--

Show: All trial metrics

Time Frame

Date Field: org_create_timestamp

Range: Custom

From: 5/12/2015 To: 5/12/2015

org_id	Trial Metrics: Account Name	org_create_timestamp	engagement_score ↓	max_num_hosts	max_num_configured_integrations	max_num_custom_metrics	max_num_dashboards
[redacted]	[redacted]	5/12/2015 4:19 PM	254	4	1	5	1
[redacted]	[redacted]	5/12/2015 12:29 PM	72	4	2	5	1
[redacted]	[redacted]	5/12/2015 6:12 AM	46	6	3	1	0
[redacted]	[redacted]	5/12/2015 9:31 PM	39	3	4	29	5
[redacted]	[redacted]	5/12/2015 4:06 PM	34	-	4	-	3
[redacted]	[redacted]	5/12/2015 9:26 PM	31	3	4	5	1
[redacted]	[redacted]	5/12/2015 2:12 PM	25	2	1	6	1
[redacted]	[redacted]	5/12/2015 8:37 AM	24	2	5	12	0
[redacted]	[redacted]	5/12/2015 3:22 PM	22	389	1	15	3
[redacted]	[redacted]	5/12/2015 9:03 AM	21	3	2	6	0
[redacted]	[redacted]	5/12/2015 6:11 PM	20	-	0	-	0
[redacted]	[redacted]	5/12/2015 5:53 PM	19	1	2	1	0
[redacted]	[redacted]	5/12/2015 4:47 AM	18	1	0	0	0
[redacted]	[redacted]	5/12/2015 9:00 AM	18	9	0	0	0
[redacted]	[redacted]	5/12/2015 10:50 AM	17	15	1	0	0
[redacted]	[redacted]	5/12/2015 8:38 PM	17	3	0	5	0
[redacted]	[redacted]	5/12/2015 10:16 AM	13	11	0	5	0
[redacted]	[redacted]	5/12/2015 1:09 PM	13	1	0	3	0
[redacted]	[redacted]	5/12/2015 5:17 PM	13	1	0	3	0
[redacted]	[redacted]	5/12/2015 9:44 AM	12	1	0	3	0
[redacted]	[redacted]	5/12/2015 2:42 PM	12	21	0	5	0

DATADOG

The Salesforce Task

```
class UploadOrgTrialMetricsToSalesforce(luigi.UploadToSalesforceTask):
    sf_external_id_field_name=luigi.Parameter(default="org_id__c")
    sf_object_name=luigi.Parameter(default="Trial_Metrics__c")
    sf_sandbox_name=luigi.Parameter(default="adminbox")

    # Common parameters
    env = luigi.Parameter()
    version = luigi.Parameter()
    data_date = luigi.DateParameter()

    def upload_file_path(self):
        return self.get_local_path()

    def requires(self):
        return [ CreateOrgTrialMetrics(
                                env=self.env,
                                version=self.version,
                                data_date=self.data_date,
                               ) ]
```

The Salesforce Task (pt2)

- https://github.com/spotify/luigi/pull/981/commits

Tips & Tricks

Save often

- Save the results of each step
  - They may be useful later on
  - Its super useful for debugging

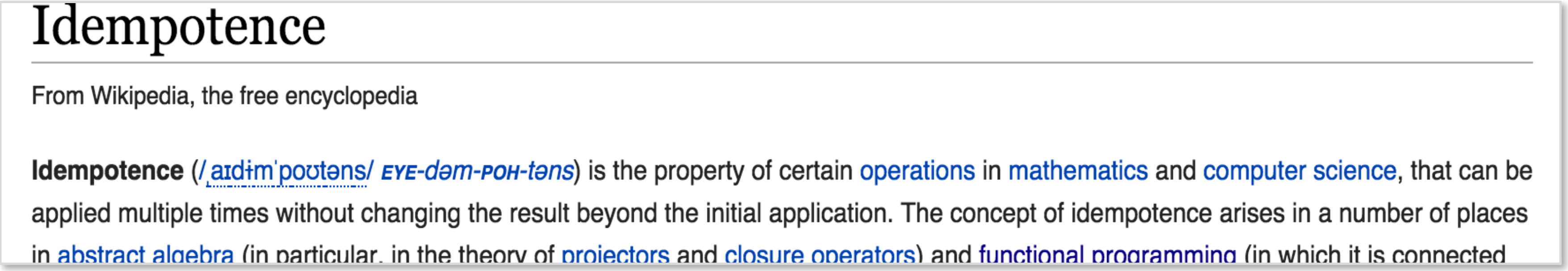
- but be ok with regenerating when needed
  - Spotify accidentally deleted massive output directory, but was easy (though time consuming) to recreate only what was needed.

Aim small miss small
(code small retry small)

Shoot for relatively small units of work

- The pipeline will be easier to understand
- If there is a task that takes a long time and might fail, easier to deal with

Idempotency – think it, live it, love it



Idempotence

From Wikipedia, the free encyclopedia

Idempotence (/ˌaɪdɨmˈpoʊtəns/ *EYE-dəm-POH-təns*) is the property of certain operations in mathematics and computer science, that can be applied multiple times without changing the result beyond the initial application. The concept of idempotence arises in a number of places in abstract algebra (in particular, in the theory of projectors and closure operators) and functional programming (in which it is connected

- Again, keep things small
- Write to somewhere else and don’t update the source data
- Tasks should only be changing one thing (if possible)
- Use atomic writes (where possible)

Parallelization can be your friend

- Luigi can parallelize your workflows
- But you need to tell it that you want that
- Default number of workers is 1
- Use --workers to specify more

How to get started

http://blog.mortardata.com/post/107531302816/building-data-pipelines-using-luigi-with-erik

- the Livestream has a weird password, but the transcript is great

- https://vimeo.com/63435580
- https://github.com/spotify/luigi

Questions?

Matt Williams
mattw@datadoghq.com
@technovangelist