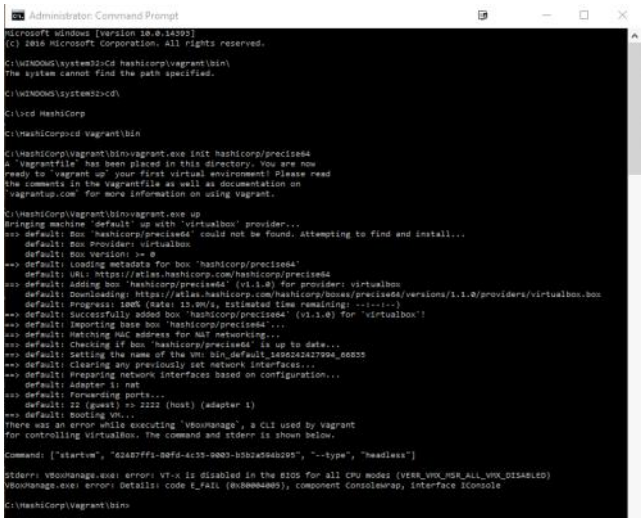
Technical Exercise

Wednesday, May 31, 2017 10:44 AM

hashicorp\vagrant\bin\

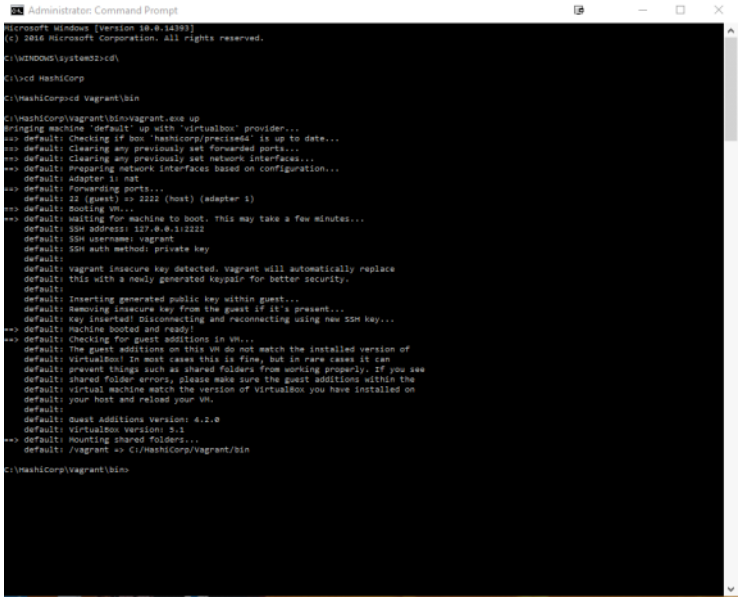
Vagrant.exe init hashicorp/precise64
Vagrant.exe up



Screen clipping taken: 5/31/2017 10:55 AM

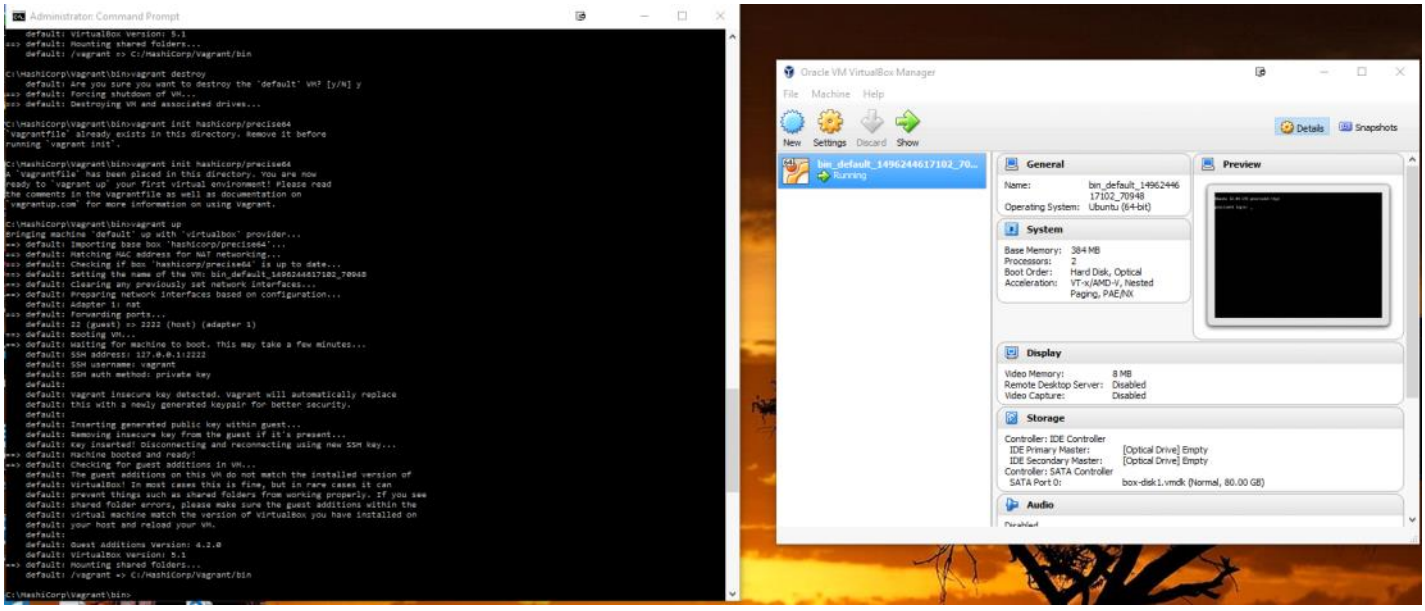
https://stackoverflow.com/questions/33304393/vt-x-is-disabled-in-the-bios-for-both-all-cpu-modes-verr-vmx-msr-all-vmx-disabl

Fixed Virtualization in BIOS and



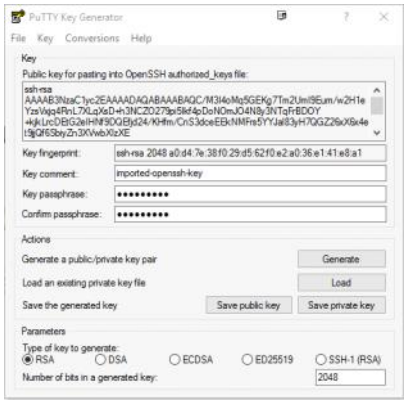
Screen clipping taken: 5/31/2017 11:04 AM

Finally!



Screen clipping taken: 5/31/2017 11:35 AM

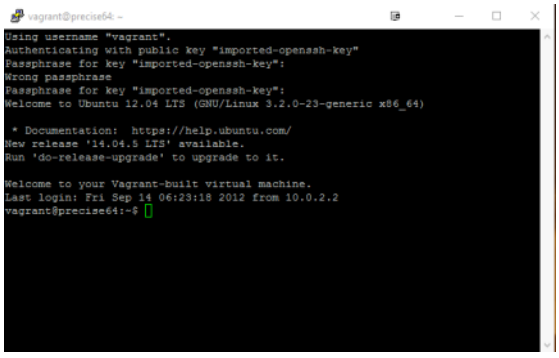
Putty KeyGen to support Private Key Auth on Windows host



Screen clipping taken: 5/31/2017 11:57 AM

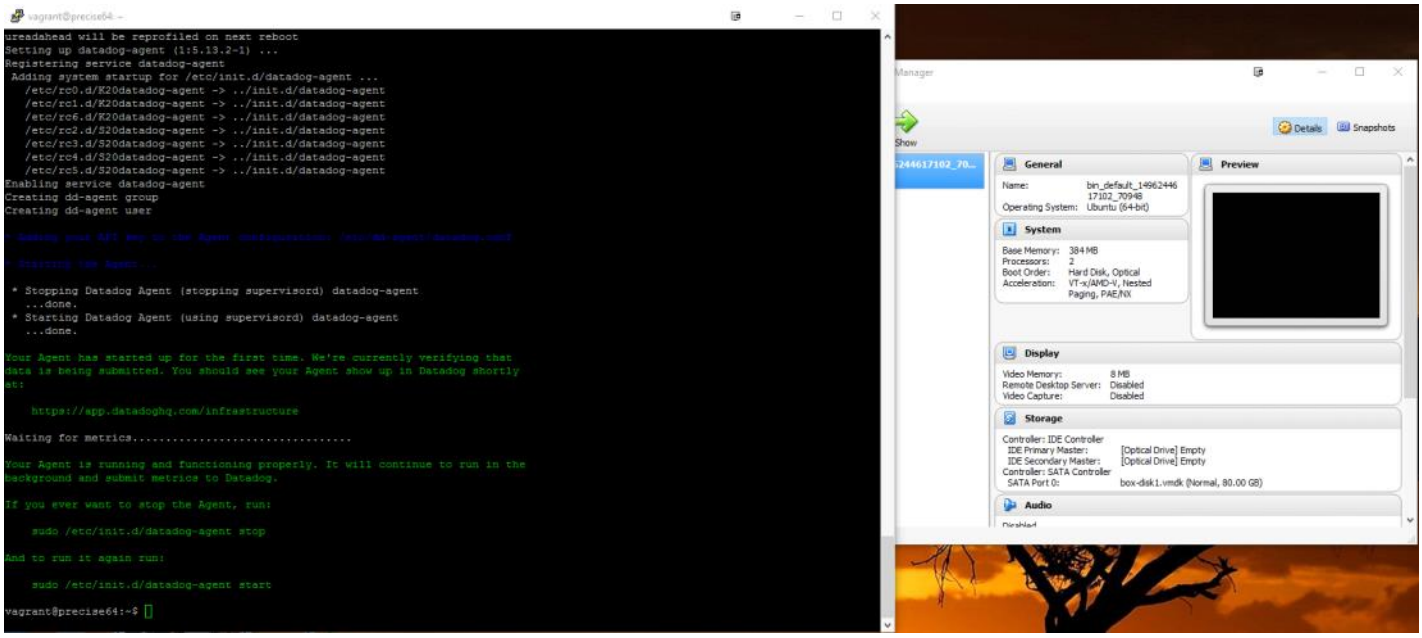
https://support.rackspace.com/how-to/logging-in-with-an-ssh-private-key-on-windows/

Success!



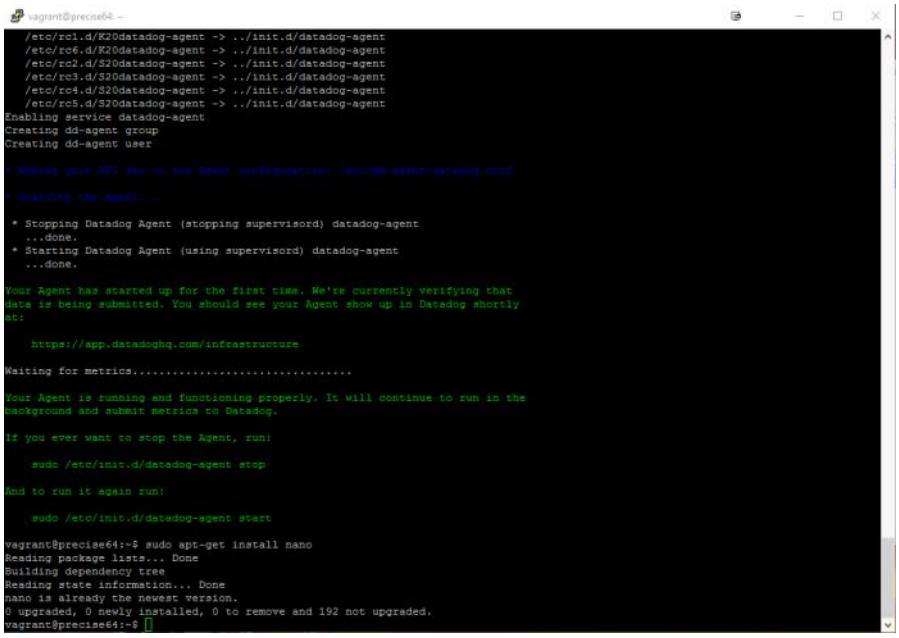
Screen clipping taken: 5/31/2017 12:00 PM

Agent installed successfully! Needed to install 'curl'

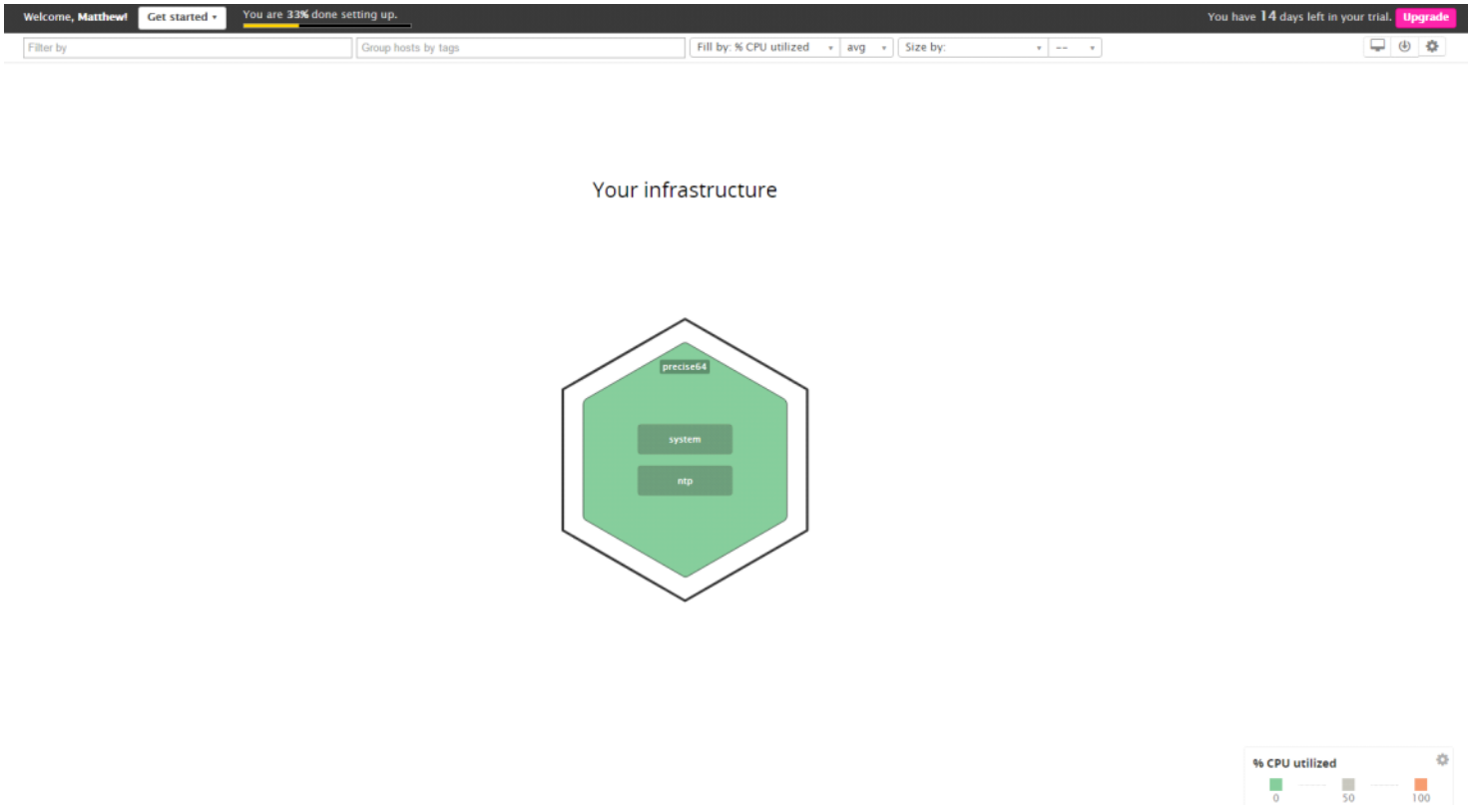


Screen clipping taken: 5/31/2017 12:08 PM

Nano terminal text editor



Screen clipping taken: 5/31/2017 12:15 PM



Screen clipping taken: 5/31/2017 12:33 PM

Changed hostname in config file prior to changing tags - which duplicated the hosts on the infrastructure list. Then saw a note saying it would take about 3 hours for them to disappear if they had been removed.

Screen clipping taken: 5/31/2017 1:28 PM

From sudo /etc/init.d/datadog-agent info saw - so I know the update worked

```
Hostnames
=========

  agent-hostname: ubuntuweb
  hostname: ubuntuweb
  socket-fqdn: precise64
```

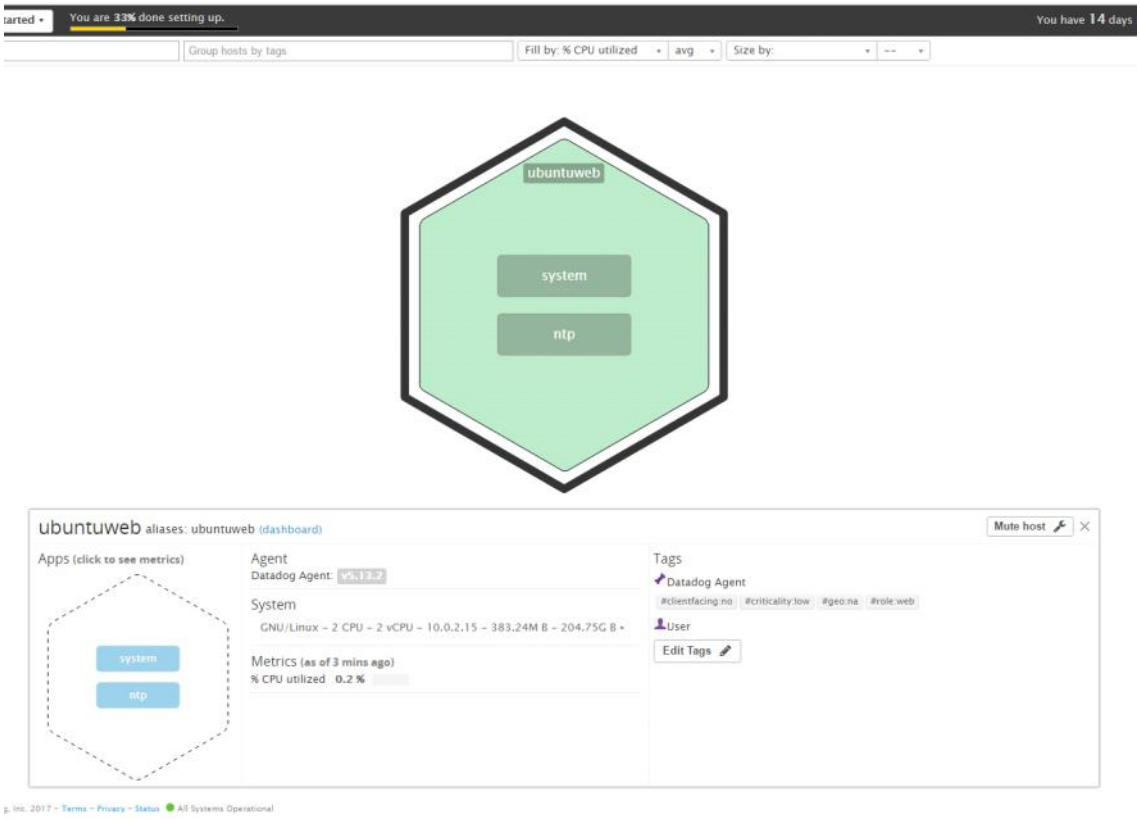
Screen clipping taken: 5/31/2017 1:29 PM

Extra machines removed by 2

Added Tags

```
  GNU nano 2.2.6                                          File: /etc/dd-agent/datadog.conf

[
[Main]

# The host of the Datadog intake server to send Agent data to
dd_url: https://app.datadoghq.com

# If you need a proxy to connect to the Internet, provide the settings here (default: disabled)
# proxy_host: my-proxy.com
# proxy_port: 3128
# proxy_user: user
# proxy_password: password
# To be used with some proxys that return a 302 which make curl switch from POST to GET
# See http://stackoverflow.com/questions/8156073/curl-violate-rfc-2616-10-3-2-and-switch-from-post-to-get
# proxy_forbid_method_switch: no

# If you run the agent behind haproxy, you might want to enable this
# skip_ssl_validation: no

# The Datadog api key to associate your Agent's data with your organization.
# Can be found here:
# https://app.datadoghq.com/account/settings
# (default: None, the agent doesn't start without it)
api_key: 2b12d61240ff877b086e15569db01abb

# Force the hostname to whatever you want. (default: auto-detected)
 hostname: ubuntuweb

# Enable the trace agent.
# apm_enabled: false

# Set the host's tags (optional)
 tags: role:web, geo:na, criticality:low, clientfacing:no

# Set timeout in seconds for outgoing requests to Datadog. (default: 20)
# When a request timeout, it will be retried after some time.
# It will only be deleted if the forwarder queue becomes too big. (30 MB by default)
# forwarder_timeout: 20

# Set timeout in seconds for integrations that use HTTP to fetch metrics, since
# unbounded timeouts can potentially block the collector indefinitely and cause
# problems!
# default_integration_http_timeout: 9

# Add one "dd_check:checkname" tag per running check. It makes it possible to slice
# and dice per monitored app (= running Agent Check) on Datadog's backend.
# create_dd_check_tags: no

# Collect AWS EC2 custom tags as agent tags (requires an IAM role associated with the instance)
# collect_ec2_tags: no
# Incorporate security-groups into tags collected from AWS EC2
# collect_security_groups: no
```

Screen clipping taken: 5/31/2017 2:10 PM



Screen clipping taken: 5/31/2017 2:10 PM

Deployed MySQL
Deployed Database just to make sure

create database datadogmysql;
grant all on datadogmysql.* to 'matt' identified by 'datadog';

```
mysql> create database datadogmysql;
Query OK, 1 row affected (0.01 sec)

mysql> grant all on datadogmysql.* to 'matt' identified by 'datadog';
Query OK, 0 rows affected (0.00 sec)
```

Configured MySQL Integration

DataDog User

```
mysql> show grants for 'datadog'@'localhost';
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
---+
| Grants for datadog@localhost

   |
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
---+
| GRANT PROCESS, REPLICATION CLIENT ON *.* TO 'datadog'@'localhost' IDENTIFIED B
Y PASSWORD '*2647B4EE68112D02461EE0510C66DF7E8DACD918' WITH MAX_USER_CONNECTIONS
 5 |
| GRANT SELECT ON `performance_schema`.* TO 'datadog'@'localhost'

   |
+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
---+
2 rows in set (0.00 sec)
```

Screen clipping taken: 5/31/2017 2:43 PM

Deleted users and started over - needed to understand running mysql commands from command line more.

```
vagrant@precise64:~$ sudo mysql -u root -p -e "CREATE USER 'datadog'@'localhost' IDENTIFIED BY 'LtzM0lscIJEenteaPJIUUvBJ';"
Enter password:
vagrant@precise64:~$ sudo mysql -u root -p -e "GRANT REPLICATION CLIENT ON *.* TO 'datadog'@'localhost' WITH MAX_USER_CONNECTIONS 5;"
Enter password:
vagrant@precise64:~$ sudo mysql -u root -p  -e "GRANT PROCESS ON *.* TO 'datadog'@'localhost';"
Enter password:
vagrant@precise64:~$ sudo mysql -u root -p  -e "GRANT SELECT ON performance_schema.* TO 'datadog'@'localhost';"
Enter password:
vagrant@precise64:~$ mysql -u root -p -u datadog --password=LtzM0lscIJEenteaPJIUUvBJ -e "show status" | \
> grep Uptime && echo -e "\033[0;32mMySQL user - OK\033[0m" || \
> echo -e "\033[0;31mCannot connect to MySQL\033[0m"
Uptime  3264
Uptime_since_flush_status       3264
MySQL user - OK
vagrant@precise64:~$ mysql -u datadog --password=LtzM0lscIJEenteaPJIUUvBJ -e "SELECT * FROM performance_schema.threads" && \
> echo -e "\033[0;32mMySQL SELECT grant - OK\033[0m" || \
> echo -e "\033[0;31mMissing SELECT grant\033[0m"
MySQL SELECT grant - OK
vagrant@precise64:~$ mysql -u datadog --password=LtzM0lscIJEenteaPJIUUvBJ -e "SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST" && \
> echo -e "\033[0;32mMySQL PROCESS grant - OK\033[0m" || \
> echo -e "\033[0;31mMissing PROCESS grant\033[0m"
+----+---------+-----------+------+---------+------+-----------+-----------------------------------------------+
| ID | USER    | HOST      | DB   | COMMAND | TIME | STATE     | INFO                                          |
+----+---------+-----------+------+---------+------+-----------+-----------------------------------------------+
| 55 | datadog | localhost | NULL | Query   |    0 | executing | SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST |
+----+---------+-----------+------+---------+------+-----------+-----------------------------------------------+
MySQL PROCESS grant - OK
vagrant@precise64:~$
vagrant@precise64:~$
```

Screen clipping taken: 5/31/2017 3:23 PM

Checks Successful!

Could not find file
/etc/dd-agent/conf.d/mysql.yaml - but found mysql.yaml.example, modified the values to match the screenshot and then saved it as mysql.yaml - restarted agent, ran info

```
Checks
======

  ntp (5.13.2)
  ------------
    - instance #0 [OK]
    - Collected 1 metric, 0 events & 1 service check

  disk (5.13.2)
  -------------
    - instance #0 [OK]
    - Collected 32 metrics, 0 events & 0 service checks

  network (5.13.2)
  ----------------
    - instance #0 [OK]
    - Collected 0 metrics, 0 events & 0 service checks

  mysql (unknown)
  ---------------
    - initialize check class [ERROR]: 'mapping values are not allowed in this context\n  in "<byte string>", line 5, column 10'
```

Screen clipping taken: 5/31/2017 3:53 PM

Initialization failure. - Turned on line numbers in nano

```
  GNU nano 2.2.6                                                File: mysql.yaml

init_config:

instances:
  - server: localhost
    user: datadog
    pass: LtzM0lscIJEenteaPJIUUvBJ
    # port: 3306            # Optional
    # sock: /path/to/sock   # Connect via Unix Socket
    # defaults_file: my.cnf  # Alternate configuration mechanism
    # connect_timeout: None  # Optional integer seconds
    tags:                 # Optional
      - optional_tag1
      - optional_tag2
    options:              # Optional
      replication: 0
    #   replication_non_blocking_status: false  # grab slave count in non-blocking manner (req. performance_schema)
      galera_cluster: false
    #   extra_status_metrics: true
    #   extra_innodb_metrics: true
    #   extra_performance_metrics: true
    #   schema_size_metrics: false
    #   disable_innodb_metrics: false
```

Screen clipping taken: 5/31/2017 3:59 PM

Tried again - same problem, found this - https://botbot.me/freenode/datadog/2015-07-19/?page=1
Changed formatting, tried again

Malformed yaml, modified the formatting and used https://yaml-online-parser.appspot.com/ and http://www.yamllint.com/ to validate YAML

Finally - this YAML worked
init_config:

instances:
 - server: localhost
   user: datadog
   pass: LtzM0lscIJEenteaPJIUUvBJ

   tags:

```
    - role:db
    - geo:na
  options:
    replication: 0
    galera_cluster: 1
    extra_status_metrics: true
    extra_innodb_metrics: true
    extra_performance_metrics: true
    schema_size_metrics: false
    disable_innodb_metrics: false
```

```
Checks
======

  ntp (5.13.2)
  ------------
    - Collected 0 metrics, 0 events & 0 service checks

  disk (5.13.2)
  -------------
    - instance #0 [OK]
    - Collected 32 metrics, 0 events & 0 service checks

  network (5.13.2)
  ----------------
    - instance #0 [OK]
    - Collected 15 metrics, 0 events & 0 service checks

  mysql (5.13.2)
  --------------
    - instance #0 [OK]
    - Collected 168 metrics, 0 events & 1 service check
    - Dependencies:
        - pymysql: 0.6.6.None
```

Screen clipping taken: 5/31/2017 6:38 PM

Writing Custom Agent Check

Hello World Example
/etc/dd-agent/checks.d/hello.py
/etc/dd-agent/conf.d/hello.yaml

```
Checks
======

  network (5.13.2)
  ----------------
    - instance #0 [OK]
    - Collected 0 metrics, 0 events & 0 service checks

  mysql (5.13.2)
  --------------
    - instance #0 [OK]
    - Collected 62 metrics, 0 events & 1 service check

  ntp (5.13.2)
  ------------
    - instance #0 [OK]
    - Collected 1 metric, 0 events & 1 service check

  disk (5.13.2)
  -------------
    - instance #0 [OK]
    - Collected 32 metrics, 0 events & 0 service checks

  hello (5.13.2)
  --------------
    - instance #0 [OK]
    - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 5/31/2017 6:52 PM

HTTP Check

YAML Config File
/etc/dd-agent/conf.d/hello.yam

```
init_config:
  default_timeout: 5

instances:
  - url: https://google.com

  - url: http://httpbin.org/delay/10
    timeout: 8

  - url: http://httpbin.org/status/400
```

PY Script
/etc/dd-agent/checks.d/http.py

```
import time
import requests

from checks import AgentCheck
from hashlib import md5

class HTTPCheck(AgentCheck):
  def check(self, instance):
    if 'url' not in instance:
      self.log.info("Skipping instance, no url found.")
      return

    # Load values from the instance config
    url = instance['url']
    default_timeout = self.init_config.get('default_timeout', 5)
    timeout = float(instance.get('timeout', default_timeout))

    # Use a hash of the URL as an aggregation key
    aggregation_key = md5(url).hexdigest()

    # Check the URL
    start_time = time.time()
    try:
      r = requests.get(url, timeout=timeout)
      end_time = time.time()
    except requests.exceptions.Timeout as e:
      # If there's a timeout
      self.timeout_event(url, timeout, aggregation_key)
      return

    if r.status_code != 200:
      self.status_code_event(url, r, aggregation_key)

    timing = end_time - start_time
    self.gauge('http.reponse_time', timing, tags=['http_check'])

  def timeout_event(self, url, timeout, aggregation_key):
    self.event({
      'timestamp': int(time.time()),
```

```
            'event_type': 'http_check',
            'msg_title': 'URL timeout',
            'msg_text': '%s timed out after %s seconds.' % (url, timeout),
            'aggregation_key': aggregation_key
        })

    def status_code_event(self, url, r, aggregation_key):
        self.event({
            'timestamp': int(time.time()),
            'event_type': 'http_check',
            'msg_title': 'Invalid reponse code for %s' % url,
            'msg_text': '%s returned a status of %s' % (url, r.status_code),
            'aggregation_key': aggregation_key
        })

if __name__ == '__main__':
    check, instances = HTTPCheck.from_yaml('/path/to/conf.d/http.yaml')
    for instance in instances:
        print "\nRunning the check against url: %s" % (instance['url'])
        check.check(instance)
        if check.has_events():
            print 'Events: %s' % (check.get_events())
        print 'Metrics: %s' % (check.get_metrics())
```

Restarted Agent and ran -info - new checks show up as ok for all three instances

```
  Checks
  ======

    http (5.13.2)
    -------------
      - instance #0 [OK]
      - instance #1 [OK]
      - instance #2 [OK]
      - Collected 1 metric, 2 events & 0 service checks

    network (5.13.2)
    ----------------
      - instance #0 [OK]
      - Collected 0 metrics, 0 events & 0 service checks

    mysql (5.13.2)
    --------------
      - instance #0 [OK]
      - Collected 62 metrics, 0 events & 1 service check

    ntp (5.13.2)
    ------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 1 service check

    disk (5.13.2)
    -------------
      - instance #0 [OK]
      - Collected 32 metrics, 0 events & 0 service checks

    hello (5.13.2)
    --------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 6/1/2017 9:01 AM

Ran the check also by running (sudo -u dd-agent dd-agent check http) just to see the output and got the below

```
vagrant@precise64:~$ sudo -u dd-agent dd-agent check http
2017-06-01 13:04:22,945 | INFO | dd.collector | config(config.py:1139) | initialized checks.d checks: ['http', 'network', 'mysql', 'ntp', 'disk', 'hello']
2017-06-01 13:04:22,945 | INFO | dd.collector | config(config.py:1140) | initialization failed checks.d checks: []
2017-06-01 13:04:22,946 | INFO | dd.collector | checks.collector(collector.py:542) | Running check http
Metrics:
[('http.reponse_time',
  1496322271,
  0.04824185371398926,
  {'hostname': 'ubuntuweb', 'tags': ['http_check'], 'type': 'gauge'})]
Events:
[{'aggregation_key': '03c6a41ca1b67234e00d5164a384540b',
  'event_type': 'http_check',
  'msg_text': 'http://httpbin.org/delay/10 timed out after 8.0 seconds.',
  'msg_title': 'URL timeout',
  'timestamp': 1496322271},
 {'aggregation_key': 'b6f5dcea9029b7f92151ca070ef1d367',
  'event_type': 'http_check',
  'msg_text': 'http://httpbin.org/status/400 returned a status of 400',
  'msg_title': 'Invalid reponse code for http://httpbin.org/status/400',
  'timestamp': 1496322271}]
Service Checks:
[]
Service Metadata:
[{}, {}, {}]
    http (5.13.2)
    -------------
      - instance #0 [OK]
      - instance #1 [OK]
      - instance #2 [OK]
      - Collected 1 metric, 2 events & 0 service checks
```

Screen clipping taken: 6/1/2017 9:06 AM

Creating 'Random Value Check'

Config File - /etc/dd-agent/conf.d/random.yam
Started with basically blank config
init_config:

instances:
  [{}]

Script File - /etc/dd-agent/checks.d/random.py
Started with

```
    Import random

    from checks import AgentCheck
    class MattCheck(AgentCheck):
        def check(self, instance):
            self.gauge('test.support.random', print(random.random())
```

Restarted - Info. Appears something blew up on the collector - 'Uncaught error running the agent'
    Looked at Collector log -/var/log/datadog/collector.log

```
2017-06-01 13:24:22 UTC | INFO | dd.collector | daemon(daemon.py:157) | Starting
2017-06-01 13:24:22 UTC | ERROR | dd.collector | config(config.py:833) | Unable to import check module random.py from checks.d
Traceback (most recent call last):
  File "/opt/datadog-agent/agent/config.py", line 829, in _get_check_class
    check_module = imp.load_source('checksd_%s' % check_name, check_path)
  File "/etc/dd-agent/checks.d/random.py", line 1
    Import random
    ^
IndentationError: unexpected indent
2017-06-01 13:24:22 UTC | INFO | dd.collector | config(config.py:1139) | initialized checks.d checks: ['http', 'network', 'mysql', 'ntp', 'disk', 'hello']
2017-06-01 13:24:22 UTC | INFO | dd.collector | config(config.py:1140) | initialization failed checks.d checks: ['random']
2017-06-01 13:24:26 UTC | INFO | dd.collector | checks.collector(collector.py:403) | Running check http
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:403) | Running check network
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:403) | Running check mysql
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:403) | Running check ntp
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:403) | Running check disk
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:403) | Running check hello
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:685) | Hostnames: {'agent-hostname': 'ubuntuweb', 'hostname': 'ubuntuweb', 'socket-fqdn': 'precise64', 'timezones': ('UTC', 'UTC')}, tags: {'system':
2017-06-01 13:24:35 UTC | INFO | dd.collector | checks.collector(collector.py:530) | Finished run #1. Collection time: 12.75s. Emit time: 0.03s
2017-06-01 13:24:36 UTC | ERROR | dd.collector | collector(agent.py:604) | Uncaught error running the Agent
Traceback (most recent call last):
  File "/opt/datadog-agent/agent/agent.py", line 600, in <module>
    sys.exit(main())
  File "/opt/datadog-agent/agent/agent.py", line 526, in main
    return Agent.info(verbose=options.verbose)
  File "/opt/datadog-agent/agent/agent.py", line 218, in info
    return CollectorStatus.print_latest_status(verbose=verbose)
  File "/opt/datadog-agent/agent/checks/check_status.py", line 281, in print_latest_status
    message = module_status.render()
  File "/opt/datadog-agent/agent/checks/check_status.py", line 179, in render
    ] + ["", ""]
  File "/opt/datadog-agent/agent/checks/check_status.py", line 545, in body_lines
```

Unexpected indent in my python script - took out the space between import random and from checks and restarted. This time it loaded, but the check failed for the random check I created

```
vagrant@precise64:~$ sudo -u dd-agent dd-agent check
Usage: /usr/bin/dd-agent check <check_name> [check_rate]
Add check_rate as last argument to compute rates
vagrant@precise64:~$ sudo -u dd-agent dd-agent check random
2017-06-01 13:35:47,095 | ERROR | dd.collector | config(config.py:833) | Unable to import check module random.py from checks.d
Traceback (most recent call last):
  File "/opt/datadog-agent/agent/config.py", line 829, in _get_check_class
    check_module = imp.load_source('checksd_%s' % check_name, check_path)
  File "/etc/dd-agent/checks.d/random.py", line 1
    Import random
    ^
```

Screen clipping taken: 6/1/2017 9:37 AM

Apparently random.py is reserved for the module I'm importing, so I need to change the name, I just switched the files to matt_random.py and matt_random.yaml and deleted my existing files, restarted the agent and ran -info

```
vagrant@precise64:~$ sudo nano /etc/dd-agent/conf.d/random.yaml
vagrant@precise64:~$ sudo nano /etc/dd-agent/checks.d/random.py
vagrant@precise64:~$ sudo nano /etc/dd-agent/checks.d/matt_random.py
vagrant@precise64:~$ sudo rm /etc/dd-agent/conf.d/random.yaml /etc/dd-agent/checks.d/random.py
vagrant@precise64:~$ ls
ddagent-install.log  postinstall.sh
vagrant@precise64:~$ sudo nano /etc/dd-agent/checks.d/random.py
vagrant@precise64:~$ sudo /etc/init.d/datadog-agent restart
 * Stopping Datadog Agent (stopping supervisord) datadog-agent
 * Starting Datadog Agent (using supervisord) datadog-agent
vagrant@precise64:~$
```

Screen clipping taken: 6/1/2017 10:47 AM

Collector blew up again on info command - 'uncaught error running the agent' - Realized my python syntax was wrong and it keeps complaining because a null value is being returned.

Trying

```
import random
mattrandomvalue = random.random()

from checks import AgentCheck
class MattCheck(AgentCheck):
   def check(self, instance):
      self.gauge('test.support.random', mattrandomvalue)
```

Same error about unexpected indent, modified the formatting to take out initial spaces, restarted Info - SUCCESS

```
  Checks
  ======

    matt_random (5.13.2)
    --------------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks

    http (5.13.2)
    -------------
      - instance #0 [OK]
      - instance #1 [OK]
      - instance #2 [OK]
      - Collected 1 metric, 2 events & 0 service checks

    network (5.13.2)
    ----------------
      - instance #0 [OK]
      - Collected 15 metrics, 0 events & 0 service checks

    mysql (5.13.2)
    --------------
      - instance #0 [OK]
      - Collected 168 metrics, 0 events & 1 service check
      - Dependencies:
          - pymysql: 0.6.6.None

    ntp (5.13.2)
    ------------
      - Collected 0 metrics, 0 events & 0 service checks

    disk (5.13.2)
    -------------
      - instance #0 [OK]
      - Collected 32 metrics, 0 events & 0 service checks

    hello (5.13.2)
    --------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks
```

Ran check directly against this to see results - VICTORY!

```
vagrant@precise64:~$ sudo -u dd-agent dd-agent check matt_random
2017-06-01 15:27:43,572 | INFO | dd.collector | config(config.py:1139) | initialized checks.d checks: ['matt_random', 'http', 'network', 'mysql', 'ntp', 'disk', 'hello']
2017-06-01 15:27:43,572 | INFO | dd.collector | config(config.py:1140) | initialization failed checks.d checks: []
2017-06-01 15:27:43,572 | INFO | dd.collector | checks.collector(collector.py:542) | Running check matt_random
Metrics:
[('test.support.random',
  1496330863,
  0.6703302721766303,
  {'hostname': 'ubuntuweb', 'type': 'gauge'})]
Events:
[]
Service Checks:
[]
Service Metadata:
[{}]
    matt_random (5.13.2)
    --------------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 6/1/2017 11:28 AM

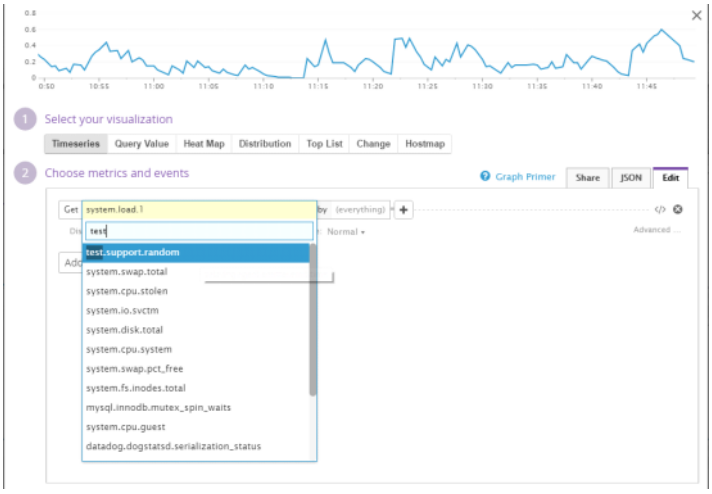
Ran again just to make sure that my script didn't inadvertently store the same random value - it did not. Just paranoid mode confirmation!

```
vagrant@precise64:~$ sudo -u dd-agent dd-agent check matt_random
2017-06-01 15:36:06,847 | INFO | dd.collector | config(config.py:1139) | initialized checks.d checks: ['matt_random', 'http', 'network', 'mysql', 'ntp', 'disk', 'hello']
2017-06-01 15:36:06,848 | INFO | dd.collector | config(config.py:1140) | initialization failed checks.d checks: []
2017-06-01 15:36:06,848 | INFO | dd.collector | checks.collector(collector.py:542) | Running check matt_random
Metrics:
[('test.support.random',
  1496331366,
  0.747074070988056,
  {'hostname': 'ubuntuweb', 'type': 'gauge'})]
Events:
[]
Service Checks:
[]
Service Metadata:
[{}]
    matt_random (5.13.2)
    --------------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 6/1/2017 11:37 AM

Visualizing Data

Cloned Dashboard - test.support.random showing up as an available metric.



v

Hmmmm - flatline graph. I ran the check again directly to see if the result was different than previous and it was.

Screen clipping taken: 6/1/2017 1:00 PM

```
vagrant@precise64:~$ sudo nano /etc/dd-agent/checks.d/matt_random.py
vagrant@precise64:~$ sudo -u dd-agent dd-agent check matt_random
2017-06-01 15:57:48,833 | INFO | dd.collector | config(config.py:1139) | initialized checks.d checks: ['matt_random', 'http', 'network', 'mysql', 'ntp', 'disk', 'hello']
2017-06-01 15:57:48,833 | INFO | dd.collector | config(config.py:1140) | initialization failed checks.d checks: []
2017-06-01 15:57:48,834 | INFO | dd.collector | checks.collector(collector.py:542) | Running check matt_random
Metrics:
[('test.support.random',
  1496332668,
  0.3501909963152954,
  {'hostname': 'ubuntuweb', 'type': 'gauge'})]
Events:
[]
Service Checks:
[]
Service Metadata:
[{}]
    matt_random (5.13.2)
    --------------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 6/1/2017 12:04 PM

Every time the service is restarted the value is changed because I set this as a global variable.
Changed to below.

```
import random
from checks import AgentCheck
class MattCheck(AgentCheck):
   def check(self, instance):
      self.gauge('test.support.random', random.random())
```

Restarted, no errors, ran info

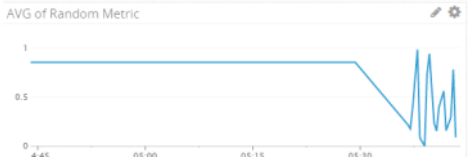
```
Checks
======

  matt_random (5.13.2)
  --------------------
    - instance #0 [OK]
    - Collected 1 metric, 0 events & 0 service checks

  http (5.13.2)
  -------------
    - instance #0 [OK]
    - instance #1 [OK]
    - instance #2 [OK]
    - Collected 1 metric, 2 events & 0 service checks

  network (5.13.2)
  ----------------
    - instance #0 [OK]
    - Collected 0 metrics, 0 events & 0 service checks

  mysql (5.13.2)
  --------------
    - instance #0 [OK]
    - Collected 62 metrics, 0 events & 1 service check

  ntp (5.13.2)
  ------------
    - instance #0 [OK]
    - Collected 1 metric, 0 events & 1 service check

  disk (5.13.2)
  -------------
    - instance #0 [OK]
    - Collected 32 metrics, 0 events & 0 service checks

  hello (5.13.2)
  --------------
    - instance #0 [OK]
    - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 6/2/2017 5:42 AM
Ran check itself

```
vagrant@precise64:~$ sudo -u dd-agent dd-agent check matt_random
2017-06-02 09:40:59,683 | INFO | dd.collector | config(config.py:1139) | initialized checks.d checks: ['matt_random', 'http', 'network', 'mysql', 'ntp', 'disk', 'hello']
2017-06-02 09:40:59,684 | INFO | dd.collector | config(config.py:1140) | initialization failed checks.d checks: []
2017-06-02 09:40:59,684 | INFO | dd.collector | checks.collector(collector.py:542) | Running check matt_random
Metrics:
[('test.support.random',
  1496396459,
  0.556564752684846,
  {'hostname': 'ubuntuweb', 'type': 'gauge'})]
Events:
[]
Service Checks:
[]
Service Metadata:
[{}]
    matt_random (5.13.2)
    --------------------
      - instance #0 [OK]
      - Collected 1 metric, 0 events & 0 service checks
```

Screen clipping taken: 6/2/2017 5:43 AM

Confirmed in DataDog HQ and am now showing varied results!



Screen clipping taken: 6/2/2017 5:44 AM

Other Agents

Custom Metric - crazyscale