→ GETTING STARTED

- INTEGRATIONS:
  - DATADOG HAS OVER 400 INTEGRATIONS OFFICIALLY LISTED.
  - CUSTOM INTEGRATIONS ARE AVAILABLE VIA THE DATADOG API.
  - THE AGENT IS OPEN SOURCE.
  - ONCE INTEGRATIONS HAVE BEEN CONFIGURED, ALL DATA IS TREATED THE SAME THROUGHOUT DATADOG, WHETHER IT IS LIVING IN A DATA CENTER OR IN AN ONLINE SERVICE.

- LOG MGMT:

DATADOG MANAGEMENT (LOG) LETS YOU SEND AND PROCESS EVERY LOG PRODUCED BY YOUR APPLICATIONS AND INFRASTRUCTURE. YOU CAN OBSERVE YOUR LOGS IN REAL-TIME USING THE LIVE TAIL, WITHOUT INDEXING THEM. YOU CAN INGEST ALL OF THE LOGS FROM YOUR APPLICATIONS AND INFRASTRUCTURE, DECIDE WHAT TO INDEX DYNAMICALLY WITH FILTERS, AND THEN STORE THEM IN AN ARCHIVE.

- APM & DISTRIBUTED TRACING:

DATADOG APPLICATION PERFORMANCE MONITORING (APM OR TRACING) PROVIDES YOU WITH DEEP INSIGHT INTO YOUR APPLICATION'S PERFORMANCE — FROM AUTOMATICALLY GENERATED DASHBOARDS FOR MONITORING KEY METRICS, LIKE REQUEST VOLUME AND LATENCY, TO DETAILED TRACES OF INDIVIDUAL REQUESTS — SIDE BY SIDE WITH YOUR LOGS AND INFRASTRUCTURE MONITORING. WHEN A REQUEST IS MADE TO AN APPLICATION, DATADOG CAN SEE THE TRACES ACROSS A DISTRIBUTED SYSTEM, AND SHOW YOU SYSTEMATIC DATA ABOUT PRECISELY WHAT IS HAPPENING TO THIS REQUEST.

- INFRASTRUCTURE:
  - ALL MACHINES SHOW UP IN THE INFRASTRUCTURE LIST.
  - YOU CAN SEE THE TAGS APPLIED TO EACH MACHINE. TAGGING ALLOWS YOU TO INDICATE WHICH MACHINES HAVE A PARTICULAR PURPOSE.
  - DATADOG ATTEMPTS TO AUTOMATICALLY CATEGORIZE YOUR SERVERS. IF A NEW MACHINE IS TAGGED, YOU CAN IMMEDIATELY SEE THE STATS FOR THAT MACHINE BASED ON WHAT WAS PREVIOUSLY SET UP FOR THAT TAG.

→ AGENT

THE AGENT IS A LIGHTWEIGHT SOFTWARE INSTALLED ON YOUR HOSTS. IT REPORTS METRICS AND EVENTS FROM YOUR HOST TO DATADOG VIA INTEGRATIONS, DOGSTATSD, OR THE API. WITH ADDITIONAL SETUP, THE AGENT CAN REPORT LIVE PROCESSES, LOGS, AND TRACES.

- CONFIGURATION:

THE AGENT'S MAIN CONFIGURATION FILE IS DATADOG.YAML. THE REQUIRED PARAMETERS ARE YOUR DATADOG API KEY WHICH IS USED TO ASSOCIATE YOUR AGENT'S DATA WITH YOUR ORGANIZATION AND THE DATADOG SITE (DATADOGHQ.COM). SEE THE SAMPLE CONFIG_TEMPLATE.YAML FOR ALL AVAILABLE CONFIGURATION OPTIONS.

→ INTEGRATIONS

AN INTEGRATION, AT THE HIGHEST LEVEL, IS WHEN YOU ASSEMBLE A UNIFIED SYSTEM FROM UNITS THAT ARE USUALLY CONSIDERED SEPARATELY. AT DATADOG, YOU CAN USE INTEGRATIONS TO BRING TOGETHER ALL OF THE METRICS AND LOGS FROM YOUR INFRASTRUCTURE AND GAIN INSIGHT INTO THE UNIFIED SYSTEM AS WHOLE - YOU CAN SEE PIECES INDIVIDUALLY AND ALSO HOW INDIVIDUAL PIECES ARE IMPACTING THE WHOLE.

NOTE : IT'S BEST TO START COLLECTING METRICS ON YOUR PROJECTS AS EARLY IN THE DEVELOPMENT PROCESS AS POSSIBLE, BUT YOU CAN START AT ANY STAGE.

DATADOG PROVIDES THREE MAIN TYPES OF INTEGRATIONS:

- AGENT-BASED : INTEGRATIONS ARE INSTALLED WITH THE DATADOG AGENT AND USE A PYTHON CLASS METHOD CALLED CHECK TO DEFINE THE METRICS TO COLLECT.
- AUTHENTICATION (CRAWLER) BASED : INTEGRATIONS ARE SETUP IN DATADOG WHERE YOU PROVIDE CREDENTIALS FOR OBTAINING METRICS WITH THE API. THESE INCLUDE POPULAR INTEGRATIONS LIKE SLACK, AWS, AZURE, AND PAGERDUTY.
- LIBRARY : INTEGRATIONS USE THE DATADOG API TO ALLOW YOU TO MONITOR APPLICATIONS BASED ON THE LANGUAGE THEY ARE WRITTEN IN, LIKE NODE.JS OR PYTHON.

YOU CAN ALSO BUILD A CUSTOM CHECK TO DEFINE AND SEND METRICS TO DATADOG FROM YOUR UNIQUE IN-HOUSE SYSTEM.

- MONITORS :

MONITORS PROVIDE ALERTS AND NOTIFICATIONS BASED ON METRIC THRESHOLDS, INTEGRATION AVAILABILITY, NETWORK ENDPOINTS, AND MORE.

- USE ANY METRIC REPORTING TO DATADOG.
- SET UP MULTI-ALERTS (BY DEVICE, HOST, ETC)
- USE @ IN ALERT MESSAGES TO DIRECT NOTIFICATIONS TO THE RIGHT PEOPLE
- SCHEDULE DOWNTIMES TO SUPPRESS NOTIFICATIONS FOR SYSTEM SHUTDOWNS, OFF-LINE MAINTENANCE, ETC.

- NETWORK PERFORMANCE MONITORING :

DATADOG (NPM) GIVES YOU VISIBILITY INTO YOUR NETWORK TRAFFIC ACROSS ANY TAGGED OBJECT IN DATADOG : FROM CONTAINERS TO HOSTS, SERVICES, AND AVAILABILITY ZONES. GROUP BY ANYTHING - FROM DATACENTERS TO TEAMS TO INDIVIDUAL CONTAINERS. USE TAGS TO FILTER TRAFFIC BY SOURCE AND DESTINATION. THE FILTERS THEN AGGREGATE INTO FLOWS, EACH SHOWING TRAFFIC BETWEEN ONE SOURCE AND ONE DESTINATION, THROUGH A CUSTOMIZABLE NETWORK PAGE AND NETWORK MAP. EACH FLOW CONTAINS NETWORK METRICS SUCH AS THROUGHPUT, BANDWIDTH, RETRANSMIT COUNT, AND SOURCE/DESTINATION INFORMATION DOWN TO THE IP, PORT, AND PID LEVELS. IT THEN REPORTS KEY METRICS SUCH AS TRAFFIC VOLUME AND TCP TRANSMITS.

- REAL USER MONITORING :

DATADOG (RUM) ENABLES YOU TO VISUALIZE AND ANALIZE THE REAL-TIME ACTIVITIES AND EXPERIENCES OF INDIVIDUAL USERS TO PRIORITIZE ENGINEERING WORK ON THE FEATURES WITH THE HIGHEST BUSINESS IMPACT. YOU CAN VISUALIZE LOAD TIMES, FRONTEND ERRORS, AND PAGE DEPENDENCIES, AND THEN CORRELATE BUSINESS AND APPLICATION METRICS SO THAT YOU CAN TROUBLESHOOT QUICKLY WITH APPLICATION, INFRASTRUCTURE, AND BUSINESS METRICS IN A SINGLE DASHBOARD.

- SERVERLESS :

SERVERLESS LETS YOU WRITE EVENT-DRIVEN CODE AND UPLOAD IT TO A CLOUD PROVIDER, WHICH MANAGES ALL OF THE UNDERLYING COMPUTE RESOURCES. DATADOG SERVERLESS BRINGS TOGETHER METRICS, TRACES, AND LOGS FROM YOUR AWS LAMBDA FUNCTIONS RUNNING SERVERLESS APPLICATIONS INTO ONE VIEW, SO THAT YOU CAN OPTIMIZE PERFORMANCE BY FILTERING TO FUNCTIONS THAT ARE GENERATING ERRORS, HIGH LATENCY, OR COLD STARTS.

- HOST MAP :

THE HOST MAP CAN BE FOUND UNDER THE INFRASTRUCTURE MENU. IT OFFERS THE ABILITY TO :

- QUICKLY VISUALIZE YOUR ENVIRONMENT
- IDENTIFY OUTLIERS
- DETECT USAGE PATTERNS
- OPTIMIZE RESOURCES

- EVENTS :

THE EVENT STREAM IS BASED ON THE SAME CONVENTIONS AS A BLOG :

- ANY EVENT IN THE STREAM CAN BE COMMENTED ON.
- CAN BE USED FOR DISTRIBUTED TEAMS AND MAINTAINING THE FOCUS OF AN INVESTIGATION.
- YOU CAN FILTER BY USER, SOURCE, TAG, HOST, STATUS, PRIORITY, AND INCIDENT.

FOR EACH INCIDENT, USERS CAN :

- INCREASE / DECREASE PRIORITY
- COMMENT
- SEE SIMILAR INCIDENTS
- @ NOTIFY TEAM MEMBERS, WHO RECEIVE AN EMAIL
- @ SUPPORT - DATADOG TO ASK FOR ASSISTANCE

- DASHBOARDS :

DASHBOARDS CONTAIN GRAPHS WITH REAL-TIME PERFORMANCE METRICS

- SYNCHRONOUS MOUSING ACROSS ALL GRAPHS IN A SCREENBOARD.
- VERTICAL BARS ARE EVENTS. THEY PUT A METRIC INTO CONTEXT.
- CLICK AND DRAG ON A GRAPH TO ZOOM IN ON A PARTICULAR TIMEFRAME.
- AS YOU HOVER OVER THE GRAPH, THE EVENT STREAM MOVES WITH YOU.
- DISPLAY BY ZONE, HOST, OR TOTAL USAGE.
- DATADOG EXPOSES A JSON EDITOR FOR THE GRAPH, ALLOWING FOR ARITHMETIC AND FUNCTIONS TO BE APPLIED TO METRICS.
- SHARE A GRAPH SNAPSHOT THAT APPEARS IN THE STREAM.
- GRAPHS CAN BE EMBEDDED IN AN IFRAME. THIS ENABLES YOU TO GIVE A 3RD PARTY ACCESS TO A LIVE GRAPH WITHOUT ALSO GIVING ACCESS TO YOUR DATA OR ANY OTHER INFORMATION.

- SECURITY MONITORING:

DATADOG SECURITY MONITORING AUTOMATICALLY DETECTS THREATS TO YOUR APPLICATION OR INFRASTRUCTURE. FOR EXAMPLE, A TARGET ATTACK, AN IP COMMUNICATING WITH YOUR SYSTEMS MATCHING A THREAT INTEL LIST, OR AN INSECURE CONFIGURATION. THESE THREATS ARE SURFACED IN DATADOG AS SECURITY SIGNALS AND CAN BE CORRELATED AND TRIAGED IN THE SECURITY EXPLORER.

- TAG:

THE TAG ENDPOINT ALLOWS YOU TO ASSIGN TAGS TO HOSTS, FOR EX: ROLE:DATABASE. THOSE TAGS ARE APPLIED TO ALL METRICS SENT BY THE HOST. REFER TO HOSTS BY NAME (YOURHOST.EXAMPLE.COM) WHEN FETCHING AND APPLYING TAGS TO A PARTICULAR HOST.

THE COMPONENT OF YOUR INFRASTRUCTURE RESPONSIBLE FOR A TAG IS IDENTIFIED BY A SOURCE. FOR EX, SOME VALID SOURCES INCLUDE NAGIOS, HUDSON, JENKINS, USERS, FEED, CHEF, PUPPET, GIT, BITBUCKET, FABRIC, CAPISTRANO, ETC.

* TAGGING IS A METHOD TO OBSERVE AGGREGATE DATA POINTS.

(1)

- IT IS RECOMMENDED TO CONSTRUCT TAGS IN THE <KEY>:<VALUE> FORMAT.

TAG	KEY	VALUE
ENV:STAGING:EAST	ENV	STAGING:EAST
ENV_STAGING:EAST	ENV_STAGING	EAST

- TAGGING METHODS:

METHOD	ASSIGN TAGS
CONFIGURATION FILES	MANUALLY IN YOUR MAIN AGENT OR INTEGRATION CONFIGURATION FILES.
UI	IN THE DATADOG APP.
API	WHEN USING DATADOG'S API.
DOGSTATSD	WHEN SUBMITTING METRICS VIA DOGSTATSD

- ACCESSING AGENT MAIN CONFIG FILE:

MACOS ~/.DATADOG-AGENT/DATADOG.YAML

UBUNTU /ETC/DATADOG-AGENT/DATADOG.YAML

AGENT LEVEL ~~LAUNCHCTL~~ START COM.DATADOGHQ.AGENT

INTEGRATION LEVEL ~~START SUDO START~~ SUDO SERVICE DATADOG-AGENT ~~START~~

GREP -N TAGS: /ETC/DATADOG-AGENT/DATADOG.YAML

SUDO SYSTEMCTL START DATADOG-AGENT

→ SUDO VI DATADOG.YAML → :66 → ESC+i → ESC → :WQ

/ETC/DATADOG-AGENT/CONF.D/

- INSTALL MYSQL ON UBUNTU

(2)

$ SUDO APT UPDATE
$ SUDO APT INSTALL
$ SUDO MYSQL_SECURE_INSTALLATION

- CONFIG A CHECK FOR AN AGENT RUNNING ON A HOST

- TO ACTIVATE A GIVEN INTEGRATION

(I) RENAME CONF.YAML.EXAMPLE > CONF.YAML
(II) UPDATE THE REQUIRED PARAMETERS INSIDE THE FILE
(III) RESTART THE DATADOG AGENT

$ MV CONF.YAML.EXAMPLE CONF.YAML
$ SUDO VI CONF.YAML
$ SUDO SERVICE DATADOG-AGENT RESTART
$ SUDO DATADOG-AGENT STATUS

- OVERWRITE FILES

755
$ SUDO CHMOD 644 MY.CNF
$ SUDO VI MY.CNF

- $ SUDO SERVICE MYSQL RESTART

- ACCESS DATADOG.YAML AND CHANGE LOGS_ENABLED: TRUE

- CD /ETC/~~MYSQL~~ DATADOG-AGENT / CONF.D / MYSQLD/ CONF.YAML

- WRITING A CUSTOM AGENT CHECK

CUSTOMS CHECKS ARE WELL SUITED TO COLLECT METRICS FROM CUSTOM APPLICATIONS OR UNIQUE SYSTEMS

* THE NAMES OF THE CONFIGURATION ~~FILE~~ AND CHECK FILES MUST MATCH. IF YOUR CHECK IS CALLED MYCHECK.PY YOUR CONFIGURATION FILE MUST BE NAMED MYCHECK.YAML

$ SUDO -S
$ SUDO -U VAGRANT -S

MY_METRIC

BONUS: YES, YOU CAN!

THE COLLECTION INTERVAL DEFAULT VALUE IS 155 BUT YOU CAN EDIT IT AS YOU WISH.
THE MIN VALUE OF 30S DOES NOT MEAN THE METRICS WILL BE COLLECTED EVERY 30S, BUT RATHER IT COULD BE COLLECTED AS OFTEN AS 30S.

- METRIC MONITOR

METRIC MONITORS ARE USEFUL FOR A CONTINUOUS STREAM OF DATA. ANY METRIC SENT TO DATADOG CAN BE ALERTED UPON IF THEY CROSS A THRESHOLD OVER A GIVEN PERIOD OF TIME.

MONITORS > NEW MONITOR > METRIC

• THRESHOLD: A THRESHOLD ALERT COMPARES METRICS VALUES TO A STATIC THRESHOLD.

ON EACH ALERT EVALUATION, DATADOG CALCULATES THE AVERAGE / MIN / MAX / SUM OVER THE SELECTED PERIOD AND CHECKS IF IT IS ABOVE OR BELOW THE THRESHOLD. THIS IS THE STANDARD ALERT CASE WHERE YOU KNOW THE EXPECTED VALUES.

ON AVERAGE	THE SERIES IS AVERAGED TO PRODUCE A SINGLE VALUE THAT IS CHECKED AGAINST THE THRESHOLD. IT ADD THE AVG() FUNCTION TO YOUR MONITOR QUERY.
NO DATA	NOTIFY IF DATA IS MISSING FOR MORE THAN N MINUTES. IF A HOST WITH THE AGENT MUST BE UP CONTINUOUSLY, YOU CAN EXPECT THE METRIC SYSTEM.CPU.IDLE TO ALWAYS REPORT DATA

- COLLECTING APM DATA

TRACING WITHOUT LIMITS: JOURNEY OF A TRACE

YOUR INSTRUMENTED APPLICATION > 15 MIN LIVE SEARCH & ANALYTICS > INTELLIGENT RETENTION & CUSTOM FILTERS > 15 DAY INDEXED SEARCH & ANALYTICS

• GET UP DATADOG APM

IN MOST ENVIRONMENTS, CONFIGURING YOUR APPLICATION TO SEND TRACES TO DATADOG INVOLVES TWO STEPS:

(I) CONFIGURING THE DATADOG AGENT FOR APM.

(II) ADDING THE DATADOG TRACING LIBRARY TO YOUR CODE.

TRACES ARE SENT FROM YOUR APPLICATION INSTRUMENTED WITH A DATADOG TRACING LIBRARY TO THE DATADOG AGENT, AND FROM THE DATADOG AGENT TO DATADOG.

- INSTALL DATADOG TRACING LIBRARY — $ PIP INSTALL DDTRACE
- INSTRUMENT YOUR PYTHON APP — $ DDTRACE-RUN FLASKAPP.PY
- CONFIGURE THE DATADOG AGENT FOR APM

INSTALL AND CONFIGURE THE DATADOG AGENT TO RECEIVE TRACES FROM YOUR NOW INSTRUMENTED APPLICATION. BY DEFAULT THE DATADOG AGENT IS ENABLED IN YOUR DATADOG.YAML FILE UNDER APM_ENABLED: TRUE AND LISTENS FOR TRACE TRAFFIC AT LOCALHOST:8126

DINAMICALLY SET SERVICE, ENV, AND VERSION TAGS

- ENVIRONMENT NAME:

IT IS RECOMMENDED TO USE THE ENVIRONMENT VARIABLE DD_ENV TO CONFIGURE ENV THROUGH YOUR SERVICE'S TRACER.

ALTERNATIVELY, NAME YOUR ENVIRONMENT BY UPDATING DATADOG.YAML TO SET ENV UNDER APM_CONFIG.

HOSTTAG ENV:<ENVIRONMENT>

PRIMARY TAGS APPEAR AT THE TOP OF APM PAGES.

$PS -FA | GREP PYTHON

$ KILL

$ SUDO KILL -9 <PROCESS_ID>

METRIC TRACE.REQUES

TRACE.<SPAN_NAME>.<METRIC_SUFFIK>

TRACE.<SPAN_NAME>.<METRIC_SUFFIK>.<2ND_PRIM_TAG>_SERVICE

LOCAL MACHINE

APP : WEB APP.PY

FROM: SERVICE : FLASKAPP1

ENV : PROD1

VM

FLASKAPP.PY

SERVICE : FLASKAPP

PROD.

- SERVICE VS RESOURCE:

SERVICE SERVICES ARE THE BUILDING BLOCKS OF MODERN MICROSERVICES ARCHITECTURES — BROADLY A SERVICE GROUPS TOGETHER ENDPOINTS, QUERIES, OR JOBS FOR THE PURPOSES OF BUILDING YOUR APPLICATION.

RESOURCE RESOURCES REPRESENT A PARTICULAR DOMAIN OF A CUSTOMER APPLICATION — THEY ARE TYPICALLY AN INSTRUMENTED WEB ENDPOINT, DATABASE QUERY, OR BACKGROUND JE

- GITHUB
  - FORK REPO
  - CLONE REPO: $GIT CLONE "REPO HTTPS"
  - STAGE FILES FOR COMMIT: $GIT ADD .
  - COMMIT THE FILES: $GIT COMMIT -M "COMMIT DESCRIPTION"
  - PUSH THE CHANGES: $GIT PUSH ORIGIN "REPO"
  - PULL REQUEST: