OJoyent[®] FULL STACK METRICS:

Richard Kiene @shmeeny Tim Gross @0x74696d









Modern Applications and Operations Made Easy





Bloyent®



TRITON Containers as a Service

Manta **Object Storage**



Public Cloud

Dector
Stronge
Dector
Dect

Triton Elastic Container Service. We run our

customer's mission critical applications on container native infrastructure

Private Cloud

our **Tri**

🕂 Inu

Triton Elastic Container Infrastructure is an on-premise, container run-time environment used by some of the world's most recognizable brands





Public Cloud Triton Elastic C

customer's missic container native in

Joyent Contore Contore Contor Contor Storage Detrag Node.je	Dashboard Instances 14 0 Instance	Docker 3 9 Jann Strong contenen	Marta Storage 271 oz	0,000 (1.5) 50.00 (1.5) 9.00 (1.5) 1.5 1.5 1.5 1.5 1.5 1.5
O Bayert Para A Annora	Control Computer Teacouse Desport Ministration View Respert Varyables	e Canto Data inat Optione Data de Angeres etca Teor Date Typese Typese	-044	
Seres d Series Name Grayment Al			JOYTERT BLOG Jur Dr. Prednime 2013 Jur Jur J. Hanne Johnson Dr. Srill Benning Joynet the Band Dr. Srill Benning Joynet the Band Dr. Srill Dr. Strandbarter and T Dr. Srill Benningbarter and T Dr. Srill Dr. Strandbarter and T Dr. Srill Dr. Lettert Roards Loss Constants	n Mang I Pasar ta Rac Banase I an Manthag Tagathar Rachard nan agar sanas Rag ta an Fad and Talana at

♦ Code Issues 28 Pull requ Ain Triton DataCenter project I 132 commits Branch: master • New pull request Issues I ools puldate server-setup doe I ools puldate server I ools Added s I tools Added s I tools<	joyent / triton				
Aain Triton DataCenter project T 32 commits Pranch: master New pull request joshwilsdon update server-setup dow assets PUBAPI-1 adocs update server- docs update server- etc RELENG i tools Added s jeitignore update i LICENSE TOOLS Makefile joyent i README.md joyent package.json TOO	() ()	ode () Issues 28	ິ່ງ Pull requ		
 Tat commits Branch: master ▼ New pull request ioshwilsdon update server-setup dot assets PUBAPI-1 docs update se etc RELENSE LICENSE Internse Update Internse <l< td=""><td>Nain</td><td>Triton DataCenter</td><td>project</td></l<>	Nain	Triton DataCenter	project		
Branch: master • New pull request ioshwilsdon update server-setup date ioassets PUBAPI-11 ioacs update server-setup date iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		T 132 commits			
isition joshwilsdon update server-setup dor i assets PUBAPI-1 i docs update se i docs i docs<	B	ranch: master - Nev	w pull request		
Image: Public product of the set		ioshwilsdon update	server-setup do		
 assets docs etc RELENG tools Added s .gitignore update .gitignore LICENSE Makefile Makefile pockage.json TOOLS 		Joern	PUBAPI-1		
 indocs indo		assets	update se		
Image: with the section Image: wit		docs	RELENG		
 tools ∴ gitignore LICENSE Makefile Makefile README.md joyent package.json TOOLS 		etc	Added s		
 Provide Provide Prov		tools	update		
ICENSE Image: Addition of the state of		.gitignore	TOOLS		
Image: Makefile Image: Makefile Image: Read Me.md Image: package.json Image: Read Me.md			iovent		
README.md joyen		Makefile	ioven		
E Package.json 100		README.md	10301		
E README.md		package.json	100		
		E README.md	fc		

Triton DataCenter

	Watch 111 ★ Star 599 Star 599 Star Star
ests 2 🗉 Wiki 🦾 Pulse 🕕 Graphs	
	eleases
ته 2 branches	Find file Clone or download -
	Latest commit 31c1988 21 days age
	8 months ag
based on Orlando's feedback	21 days a
eq: restdown BE GONE! index.restdown >	a year a
ver-setup doc based on Orlando's feedback	2 months
e13: deprecate sdc-zookeeper	2 years
tup script for using CoaL on linux with KVM and liby	2 years
and start reviewing notes	2 year
607 I for one, welcome my new open source license	a mon
dc#33: make clone-all-repos	2 ye
sdc#203 first pass at sdc -> triton naming changing	

LS-623 Add repos list

its open source! ork me, pull me: <u>https://github.com/joyent/triton</u>

Triton is proven at scale; it is the software that runs the

DeteCenter (just "Triton" for short, formerly "SmartDataCenter" and "SDC") is an open-source cloud management



---- Come in... NE'REHRRIGE

https://joyent.com/about/careers



"We have built mind-bogglingly complicated systems that we cannot see, allowing glaring performance problems to hide in broad daylight in our systems."

Bryan Cantrill, Joyent CTO

ACM Queue Vol 4, Issue 1, 2006 Feb 23 http://queue.acm.org/detail.cfm?id=1117401

"System performance problems are typically introduced at the highest layers of abstraction, but they are often first encountered and attributed at the lowest layers of abstraction."

Bryan Cantrill, Joyent CTO

ACM Queue Vol 4, Issue 1, 2006 Feb 23 http://queue.acm.org/detail.cfm?id=1117401

MONTORING IN PRODUCTION Hardest problems appear in production Must be able to observe safely in production:

- - No risk of crashing
 - observed environment

Dynamic instrumentation: no performance hit on

VALUE OF OBSERVABILITY Observability is the key to being production-ready Much of Joyent's value over our competitors is our best-in-class observability and debugging tooling

TRITON ARCHITECTURE

- Customer applications run as containers
- SmartOS or Linux (LX) infrastructure containers, or Docker application containers, running as Solaris Zones
- Proven battle-tested multi-tenant security
- Bare-metal performance
- Isolation provides observability w/o interference



CLOUD ANALYTICS V1

- Historical data is cumbersome to use
- API is awkward for high-dimensionality
- Want to improve scalability w/ aggregation
- Want better availability
- No path for end users to application-level metrics

DESIGN CONSTRAINTS Multi-tenant: One customer can't cause brown-outs for other customers! their existing monitoring

Operators of Triton provide an API for customers (endusers, developers, etc.) to deploy their containers.

Give customers a sane migration path or let them use



WHY PUL 2

We don't drop metrics for overloaded target (collection happens outside the zone) Can easily throttle customer requests Pushing to a customer collector that's down requires implementing back-off/buffering for every customer in metrics agent End-users can have multiple consumers



WHY PROMETHEUS? Pull not push Agnostic to storage: end with the metrics afterwa their existing metrics so

Agnostic to storage: end-users can do what they want with the metrics afterwards (including push them into their existing metrics solution if they want!)

CONTAINER MONITOR: ARCHITECTURE



METRIC AGENT

Instance on each physical machine ("compute node") Collects metrics from all containers via kstat, zfs list, etc.

Triton compute node: SmartOS Many customer containers Metric Agent



Noun Project icon by Aneeque Ahmed

ς	

Customer	Customer	Customer
Container	Container	Container
Customer	Customer	Customer
Container	Container	Container
Metric	Customer	Customer
Agent	Container	Container
Customer	Customer	Customer
Container	Container	Container

SmartOS Container Hypervisor

Triton data center: Many compute nodes Each has its own Metric Agent



METRIC AGENT PROXY

- Stateless and horizontally scalable
- Routes Prometheus server requests to appropriate Metric Agent

HA across data center: 1 on head node + min 2 per DC

Responsible for rate-limiting and authentication

DISCOVERY: TRITON CNS Triton Container Name Service (CNS): automated container-native DNS service instances (and services) Proxy's IP for each container

- Containers are automatically assigned A-Records for
- Container Monitor provides CNAME to Metric Agent

Metric Agent Proxy: Prometheus API to each Metric Agent Metric Metric Agent Agent Metric Metric Agent Agent



METRICS COLLECTION

- Customer-owned Prometheus server(s)
- metrics to existing monitoring systems

Optional customer-owned Metrics Forwarders: forward

Customer-owned



Metrics Forwarder
Customer-owned
Translate from
Prometheus API to
Influx, Graphite, etc.

Metric

Agent

Metric

Agent



Metric Proxy

> Metric Agent

Metric Agent



HOW A CONTAINER GETS MONITORED End-user launches container VMAPI pushes change feed event to CNS New CNAME record for each container to Metric Agent Proxy IP address



HOW A CONTAINER GETS MONITORED. CONT.

- plugin to poll metric agent proxy endpoints for all containers associated with that account
 - metric agent

Customer's Prometheus server uses Triton discovery Metric Agent Proxy forwards requests to appropriate

APPLICATION METRICS: CONTAINERPLOT



AUTOPLOT PATTERN

- applications
- coordinate their actions thru globally shared state Platform agnostic

Design pattern for self-operating and self-managing

Containers adapt to changes in their environment and

CONTAINERPILOT

App-centric micro-orchestrator that enables the Autopilot Pattern cycle hooks metrics endpoint

Acts as PID1 in the container and fires user-defined life-

Telemetry "sensor" hooks feed data to a Prometheus



CONTAINERPILOT METRICS ON TRITON Containers have a CNS name ContainerPilot exposes Prometheus endpoint Add discovery catalog (ex. Consul, etcd) to Prometheus server config



```
"consul": "consul:8500",
 "preStart": "/usr/local/bin/reload.sh preStart",
 "logging": {"level": "DEBUG"},
"services":
     "name": "nginx",
     "port": 80,
     "health": "/usr/bin/curl --fail -s http://localhost/health",
     "poll": 10,
    "ttl": 25
 ٦
 "backends": [
     "name": "example",
     "poll": 7,
     "onChange": "/usr/local/bin/reload.sh"
"telemetry": {
   "port": 9090,
  "sensors":
       "name": "tb_nginx_connections_unhandled_total",
      "help": "Number of accepted connnections that were not handled",
      "type": "gauge",
       "poll": 5,
       "check": ["/usr/local/bin/sensor.sh", "unhandled"]
    },
       "name": "tb_nginx_connections_load",
       "help": "Ratio of active connections (less waiting) to the maximum worker connections",
        type": "gauge",
       "poll": 5,
       "check": ["/usr/local/bin/sensor.sh", "connections_load"]
```

ContainerPilot config file

```
"consul": "consul:8500",
"preStart": "/usr/local/bin/reload.sh preStart",
"logging": {"level": "DEBUG"},
"services":
    "name": "nginx",
     "port": 80,
    "health": "/usr/bin/curl --fail -s http://localhost/health",
    "poll": 10,
    "ttl": 25
 ر ا
"backends":
    "name": "example",
     "poll": 7,
    "onChange": "/usr/local/bin/reload.sh"
 و
"telemetry": {
    "port": 9090,
    "sensors":
        "name": "tb nginx connections unhandled total",
```

ContainerPilot config file

"help": "Number of accepted connections that were not handled".

```
"name": "example",
     "poll": 7,
                                                             ContainerPilot
      "onChange": "/usr/local/bin/reload.sh"
                                                             config file
"telemetry": {
   "port": 9090,
    "sensors":
       "name": "tb_nginx_connections_unhandled_total",
       "help": "Number of accepted connnections that were not handled",
       "type": "gauge",
       "poll": 5,
       "check": ["/usr/local/bin/sensor.sh", "unhandled"]
     , {
       "name": "tb_nginx_connections_load",
       "help": "Ratio of active connections (less waiting) to the maximum worker connections",
       "type": "gauge",
       "poll": 5,
        "check": ["/usr/local/bin/sensor.sh", "connections_load"]
```



OJoyent[®] FULL STACK METRICS:

Richard Kiene @shmeeny Tim Gross @0x74696d





