



KubeCon



CloudNativeCon

North America 2017

Pontoon

An Enterprise grade serverless framework using Kubernetes

Kumar Gaurav, Director R&D, *VMware*

Mageshwaran R, Staff Engineer R&D, *VMware*

Serverless: a quick review

- Enables running back end logic with out managing server applications
- Mostly event driven and **instantaneously** scalable
- Constraints
 - Ephemeral – should start, process and finish in some time (Calls for small startup time)
 - Stateless – shares no process state across executions (Calls for external persistence / state management)
- Typical Use cases
 - Scheduled tasks,
 - Dynamic and burstable workloads,
 - Message driven applications

AWS



Lambda

Azure



Functions

Google Cloud



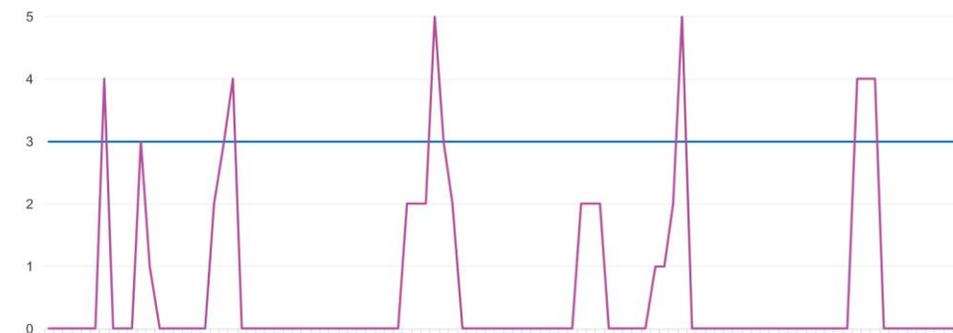
Functions

IBM OpenWhisk



Actions

projected CAGR of 32.7%



Virtual Machines

Virtual Machines run constantly at the provisioned capacity



Serverless

Function-as-a-Service respond and scale to meet demands

Agenda

- What is our need
- Requirements
- Pontoon
 - Architecture
 - Capabilities
 - Operational aspects
- Take away
- Demo

What is our need?

We work for **VMware Cloud Services**

- We ship our binaries in **Docker** container packaging, and use **Kubernetes** as the orchestration layer
- Run on AWS cloud and as well as vSphere private cloud → ~~AWS Lambda~~
- By April'17, we needed a framework to run scheduled & on-demand jobs
- We evaluated popular frameworks, but finally built one (**Pontoon**) on top of Kubernetes Jobs in about 2-3 eng months effort

** Solve for self*

** My company pays me for solving business problem only*

Serverless Functions

Our Requirements

1. Function is a process
2. Finite
3. Stateless
4. JVM, Python and Go (maybe!)
5. Runtime: few sec to few min
6. Idempotent

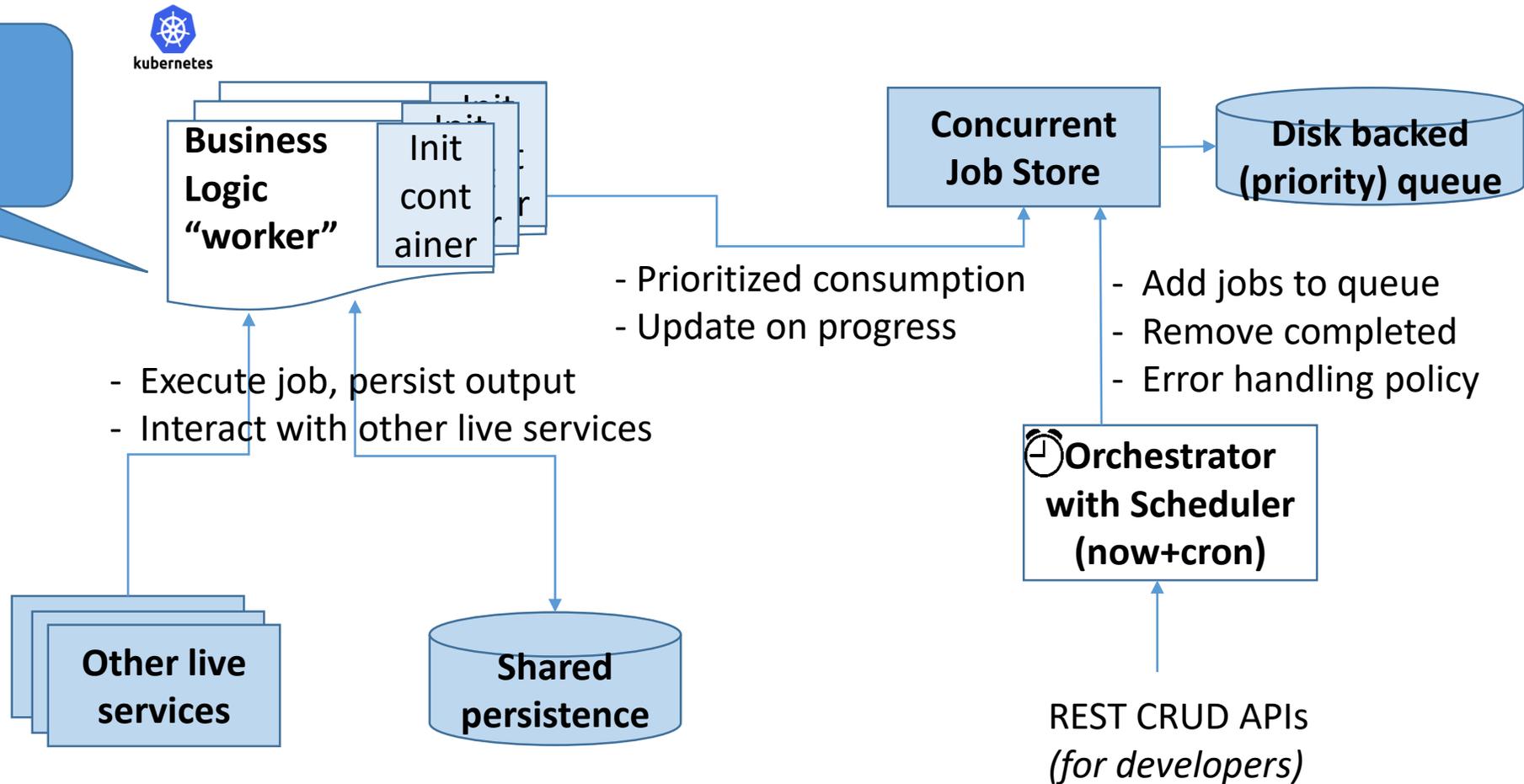
Serverless Framework

Our Requirements

1. Native to Kubernetes
2. Persist logs
3. Functions can access other k8s resources
4. Retry limit, and max parallelism
5. Should be highly available
6. Visibility on function execution and status
7. At least once execution
8. "Register" function by API
9. Not needed: "chaining" of functions

Requirement translation

- * How to package function?
- * Spec for developers?



Fitment of existing frameworks

Requirements	Iron IO Functions	Fabric8 Funktion	Fission	OpenWhisk
K8S Integration	Yes, not native	K8s only	K8s only	Yes, not native
Access to K8S cluster resources	No	Yes	Yes	No
Language agnostic	Yes, Container	Yes, Container	Yes, Environment	Yes, Container
Visibility on event queue/ monitoring	No	Yes	No	-
Persisted Logs	No	Yes	Yes	-
Cons	Error handling; k8s network; Logs	Stores code in config map; SANDBOXED ☹️ CLOSEST MATCH	JVM not supported	Deploy on k8s wasn't fully supported at time of eval

This might be a bit outdated. Comprehensive study was done in early April 2017. New entrants: kubeless

Enters Pontoon

Pontoons are airtight hollow structures, similar to [pressure vessels](#), designed to provide [buoyancy](#) in water. Their principal applications are in [watercraft hulls](#) and aircraft floats, [floating pier](#) and [pontoon bridge](#) construction, and marine engineering applications such as [salvage](#).

-- wikipedia



Light weight. Purpose built. Versatile.

Pontoon - Design choices

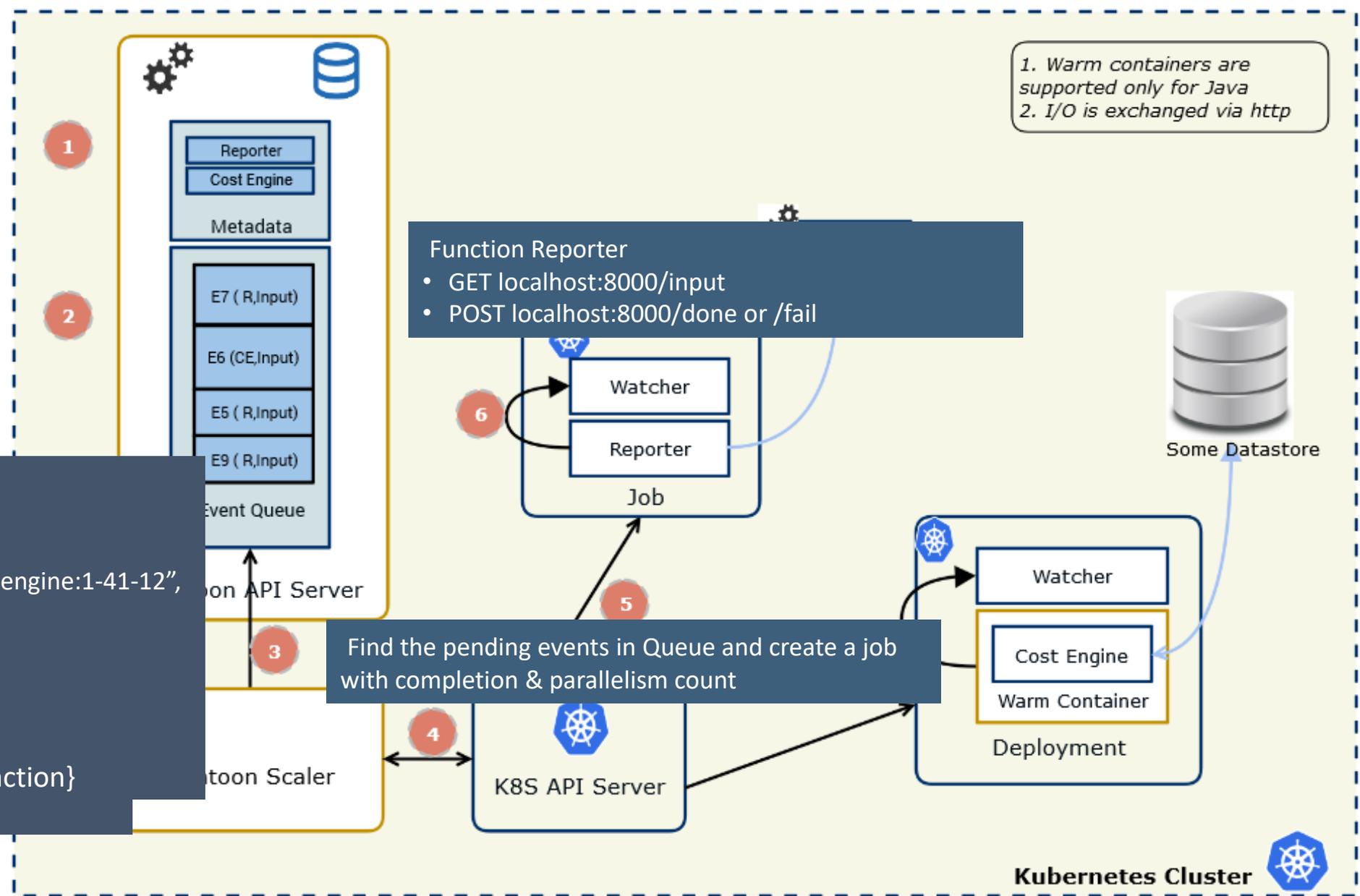
- Do not interact with docker daemon directly
 - Need to handle image cleanup
 - Might not work with K8S cluster monitoring tools
 - Requires custom CLI for monitoring
- Embrace Kubernetes Job & Deployments
 - Cold startup - Job
 - Ability to control parallel executions
 - Easier optimization using no. of completions
 - Deployment - Warm nodes (not yet done)
- We don't aspire for pure play *FaaS*. If you were thinking of enhanced auth/security

Pontoon - Architecture

```
POST /api/v1/function/{app}
{
```

```
  "name" : "CostEngine",
  "image" : "**/pontoon/cost-engine:1-41-12",
  "timeout": 30,
  "memory": 1500,
  "maxRetryCount": 3,
  "priority": 1,
  "parallelism" : 3
```

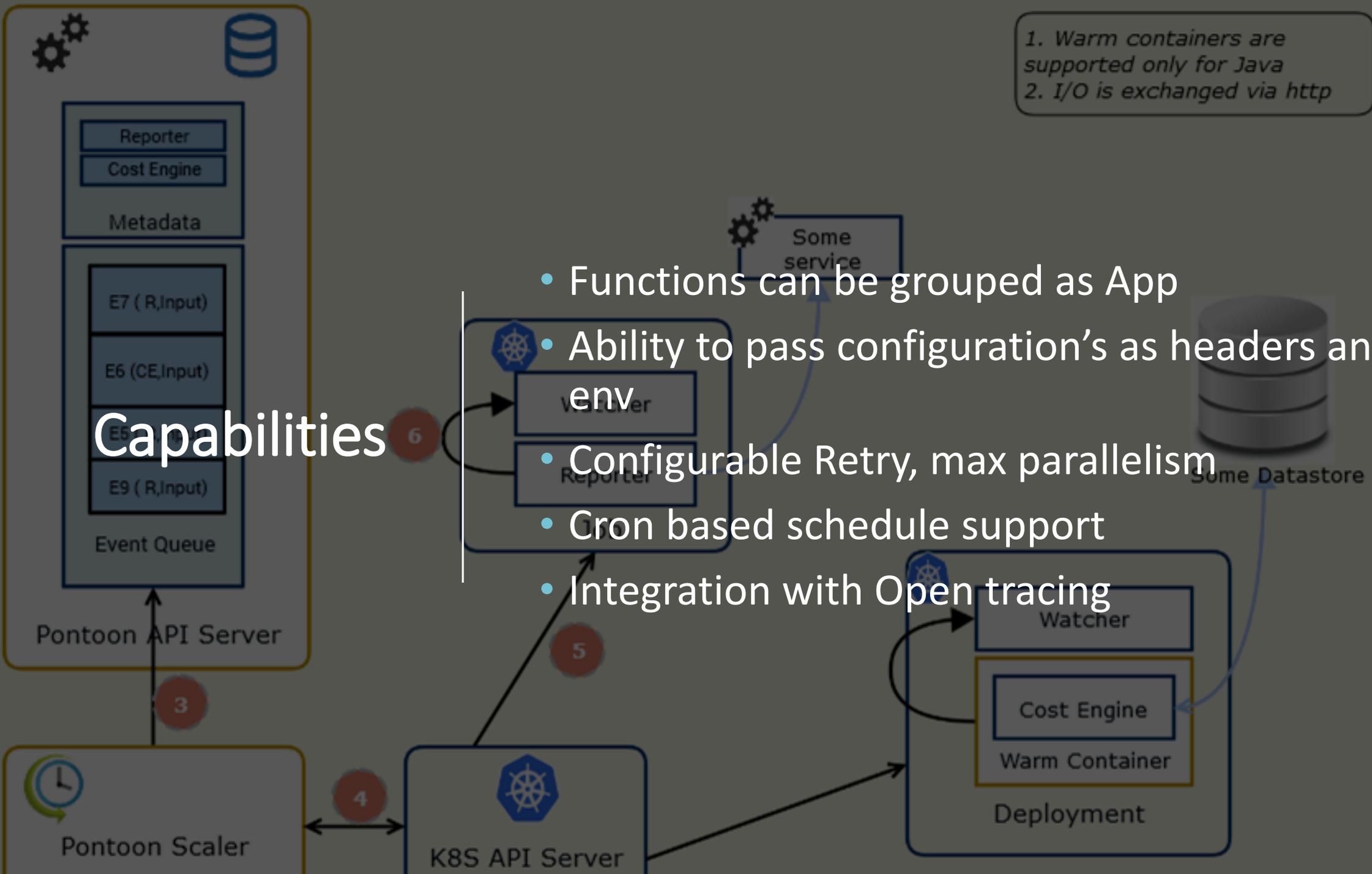
```
POST /api/v1/event/{app}/{function}
[ "Event payload in JSON format" ]
```



Capabilities

1. Warm containers are supported only for Java
2. I/O is exchanged via http

- Functions can be grouped as App
- Ability to pass configuration's as headers and env
- Configurable Retry, max parallelism
- Cron based schedule support
- Integration with Open tracing



1. Warm containers are supported only for Java
2. I/O is exchanged via http

Operational aspects

- Deployment using a single yaml
- Integrate your Logging solutions
- Event Audit – Configurable history
- Monitoring UI
- Duplicate event detection (configurable)
- Runs everywhere on K8S
- Unit testability using HTTP Mock Server

1

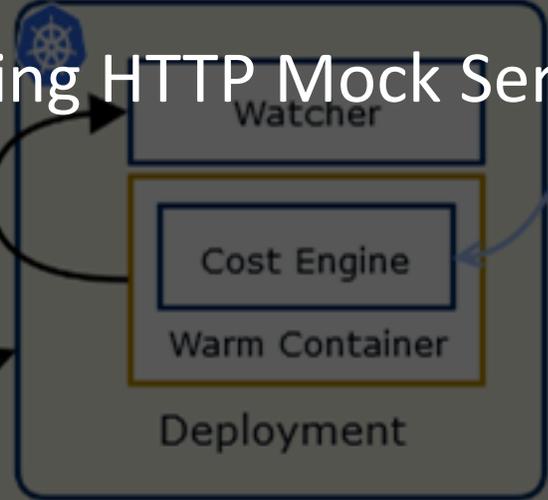
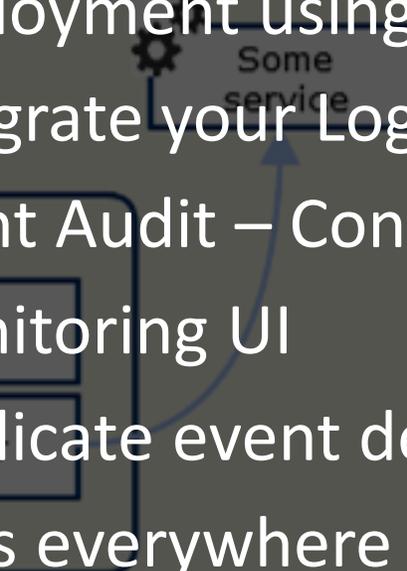
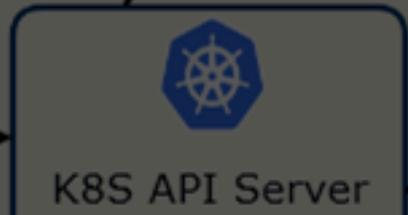
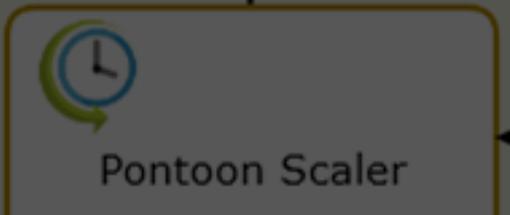
2

6

5

3

4



A quick DEMO!

```
public static void main(String[] args) {
    log.info("Starting prime generator");
    Optional<Long[]> input = watcher.getInputAsJson(Long[].class);
    try {
        input.ifPresent(limits -> {
            log.info("Find primes between {} and {}" + limits[0], limits[1]);
            watcher.done(Utils.toJson(findPrimesBetween(limits[0], limits[1])));
        });
    } catch (Exception e) {
        watcher.fail(e.getMessage());
    }
}

static List<Long> findPrimesBetween(long start, long end) {...}

private static boolean isPrime(long n) {...}
```

Monitoring: Logs

vm Log Insight Dashboards Interactive Analytics Technology Preview Night

Count of events + over time

Latest 24 hours of data

Match all of the following filters: 2017-12-04 16:47:24.347 to 2017-12-05 16:47:24.347

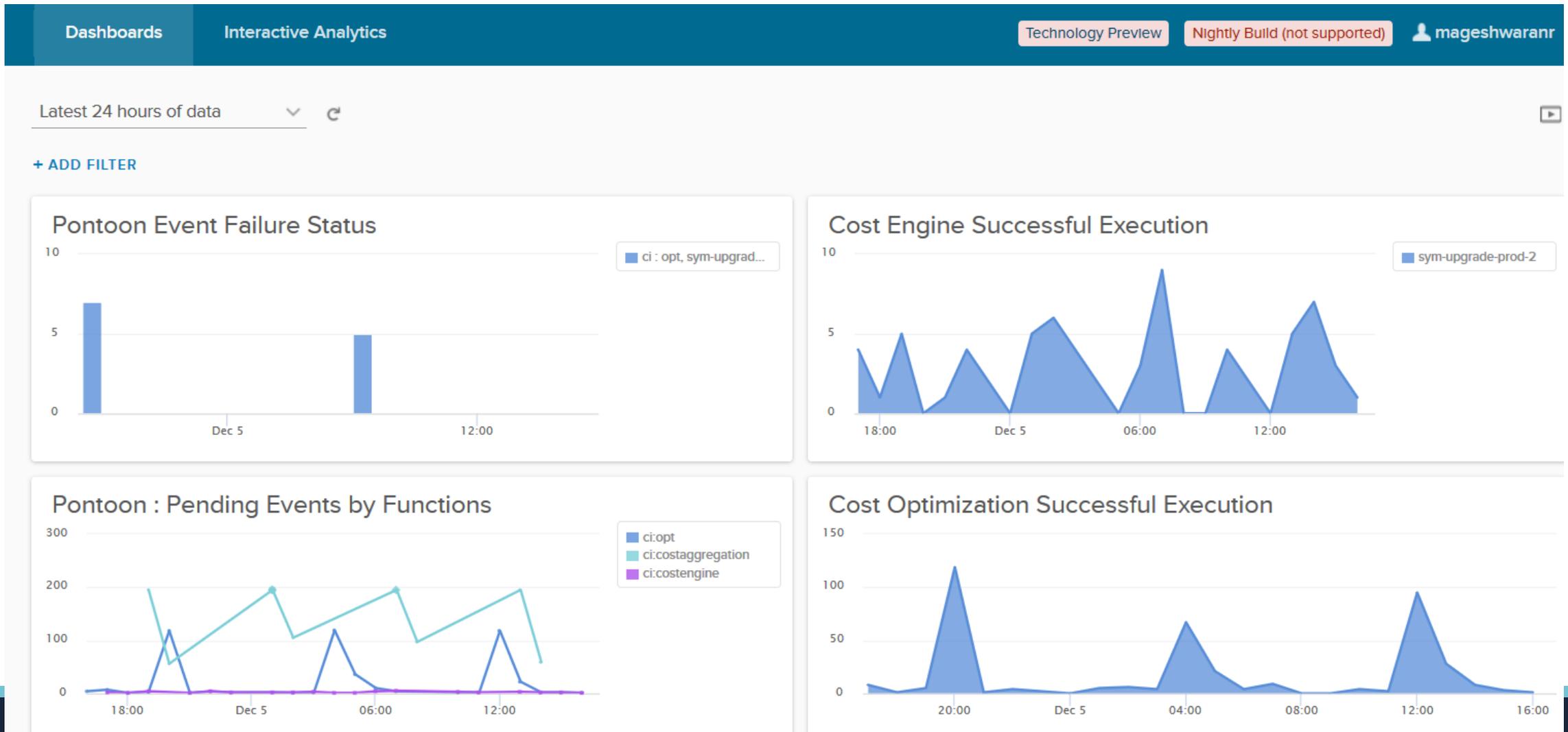
- environment contains sym-upgrade-prod-2
- product contains vrbc-daemon-lagent
- text contains OptimizationDriverFunction, EnumerateAwsComputeInstance, InstanceEnumerator, EnumerateAwsDiskInstance, EnumerateVsphereVms, ComputeId, ComputePoweredOff, DiskIdle, DiskUnused, PoweredOffCompute, EnumerateEndpoints, OptimizationAnalysisFunction, OptimizationUtils
- filepath contains optimization-driver-function-logs

[+ ADD FILTER](#) [X CLEAR ALL FILTERS](#)

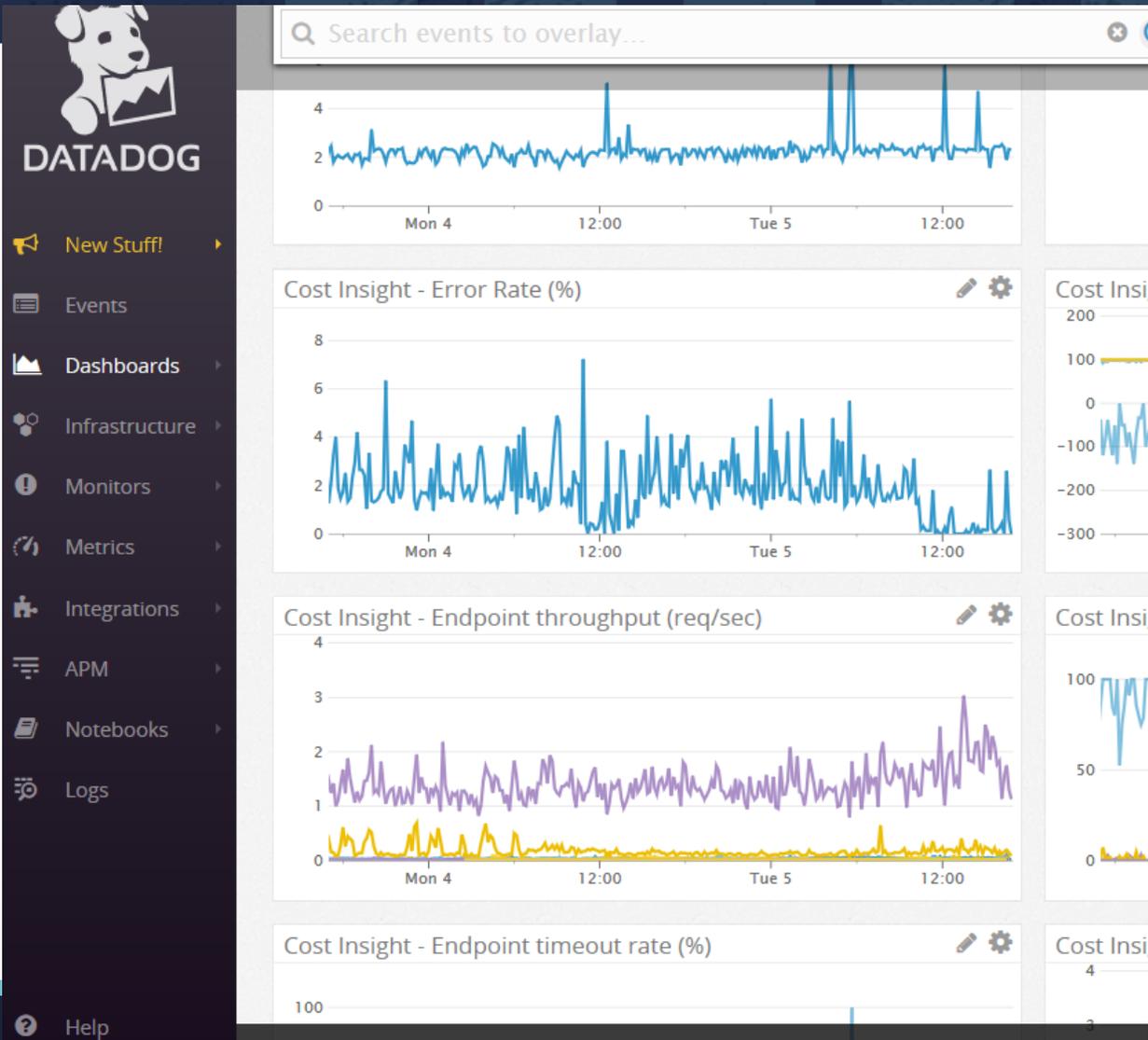
Events 1 to 50 out of 525,559+ events

2017-12-05 16:32:15.157	[1903][I][2017-12-05T21:32:15.105Z][1][EnumerateEndpoints][findEndpoints][Recommendation function completed successfully for endpointLink: /resources/endpoints /19d85662dafa407555607644714ae] source environment event_type filepath hostname product service symphony tenant symphonyEndPointLink
2017-12-05 16:32:14.733	[1807][I][2017-12-05T21:32:14.635Z][752][AggregatorService][lambda\$aggregate\$1][Going to create EndpointSummary for Readiness Readiness(id=1512509527476000, loadNo=1512509527476000, endpointLink=/resources/endpoints/19d85662dafa407555607644714ae, tenantLink=/tenants/project/3ae8d9e1e704a875-7f703c5265a63d87, status=DONE, function=com.vmware.vrbc.optimization.OptimizationAnalysisFunction, criteria={endpointType=vsphere-on-prem, executionDateStamp=1512432000000000, enumerationType=com.vmware.vrbc.optimization.enumerators.vsphere.EnumerateVsphereVms}, customProperties=null, lastUpdatedOn=1512509527476) - Found 30] source environment event_type filepath hostname product service symphony tenant symphonyEndPointLink
2017-12-05 16:32:14.733	[1806][I][2017-12-05T21:32:14.611Z][24][AggregatorService][aggregate][Going to create optimisation aggregate summary for OrgLink: /tenants/organization/3ae8d9e1e704a875, Readiness:Readiness(id=1512509527476000, loadNo=1512509527476000, endpointLink=/resources/endpoints/19d85662dafa407555607644714ae, tenantLink=/tenants/project/3ae8d9e1e704a875-7f703c5265a63d87, status=DONE, function=com.vmware.vrbc.optimization.OptimizationAnalysisFunction, criteria={endpointType=vsphere-on-prem, executionDateStamp=1512432000000000, enumerationType=com.vmware.vrbc.optimization.enumerators.vsphere.EnumerateVsphereVms}, customProperties=null, lastUpdatedOn=1512509527476)] source environment event_type filepath hostname product service symphony tenant symphonyOrgLink symphonyEndPointLink

Monitoring: Status reporting



Monitoring: Job Execution



Our Experience – take away

- Look at your requirements first
- Then scout community projects
- If needed, build your own: it's easy!

- Watch out for scale issues not in function but on its dependency (DB, other services etc.,)
- Doesn't mean no-ops. Needs observability & monitoring
- Product becomes more robust
 - transition to 12factor app

Saves us ~1,000 USD a day

Expect 10X more scale by end-2018

Next: Pontoon running on AWS Spot Instances



Questions ???

