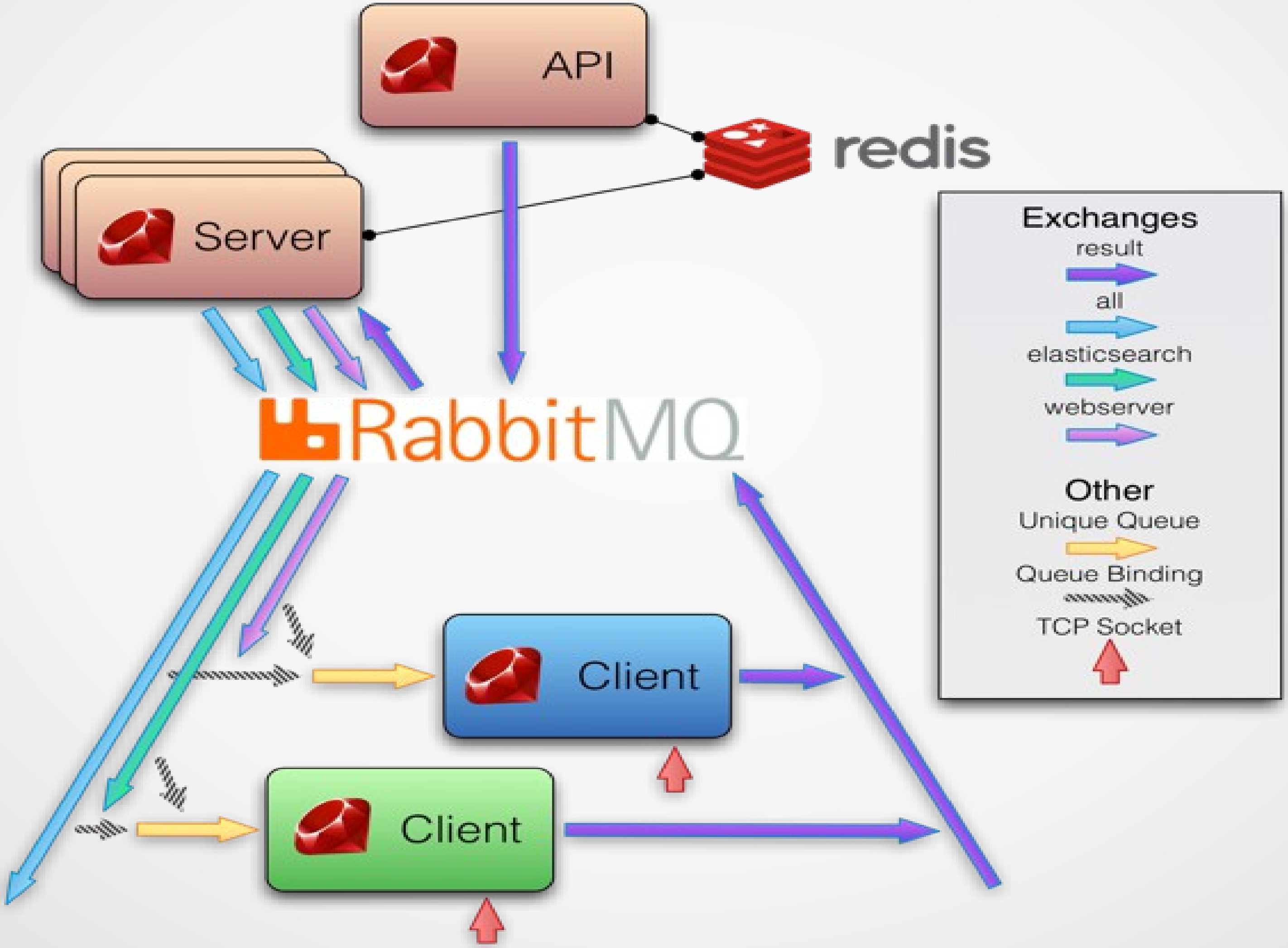
Sensu Monitoring framework

Sensu, a monitoring framework that aims to be simple, malleable, and scalable

Motivation

- **Cloud adoption** – servers come and go → discovery of new hosts, removal of “dead” hosts can be pain on old monitoring systems
- **Scalability** – old monitoring systems get easily overwhelmed with growing number of monitored nodes
- **Configuration** – not easy to understand; not designed with CM in mind
- **Difficult to extend** – we rarely hack in C these days, right ? :)

Enter Sensu (http://portertech.ca)



Sensu Stack

- **Ruby**
- **RabbitMQ**
- **Redis**
- **JSON**

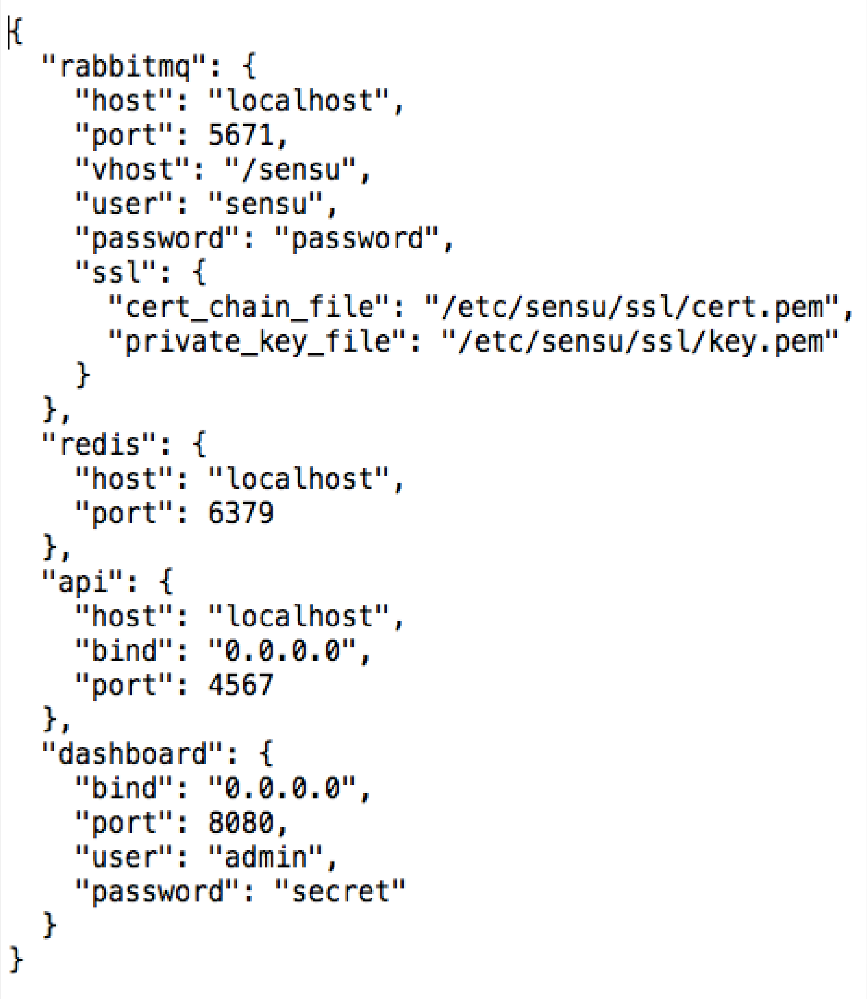
Sensu Components

- **Sensu server -** publishes check requests; receives check results and acts on them via handlers
- **Sensu client** – receives check requests; executes checks and reports back to the server
- **Sensu API** – allows to integrate with other tools in Infrastructure; can be used for lots of cool tricks and hacks
- **Sensu dashboard –** WebUI to check and act on alerts; simple auditing

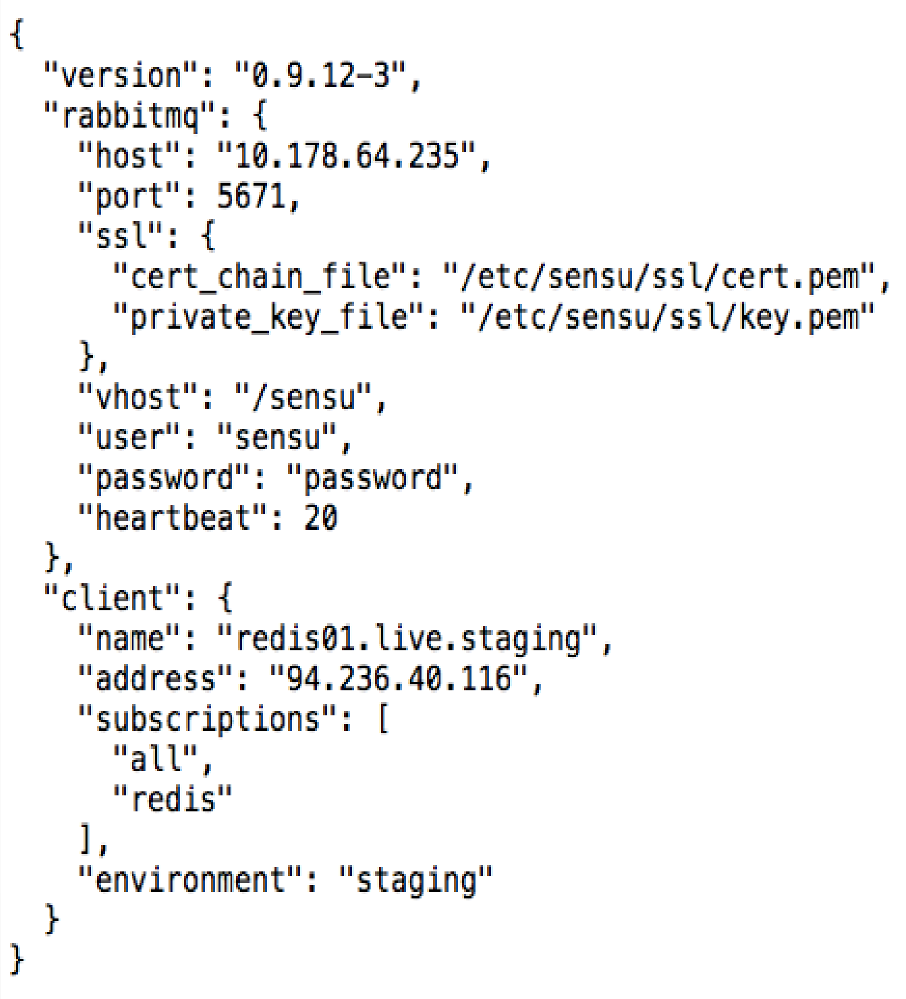
Core Features

- **Very lightweight – server and client are tiny ruby daemons**
- **CM friendly**
- **Auto-discovery and auto-registration of nodes**
- **Simple plugin interface**
- Available as **Rubygem** or as **OS package** (.deb and .rpm) which ships Sensu as **omnibus install**
- **JSON logging**
- **Secure communication**
- **Can be used to ship metrics**
- **Lots of other goodies...**

Configuration – server and client

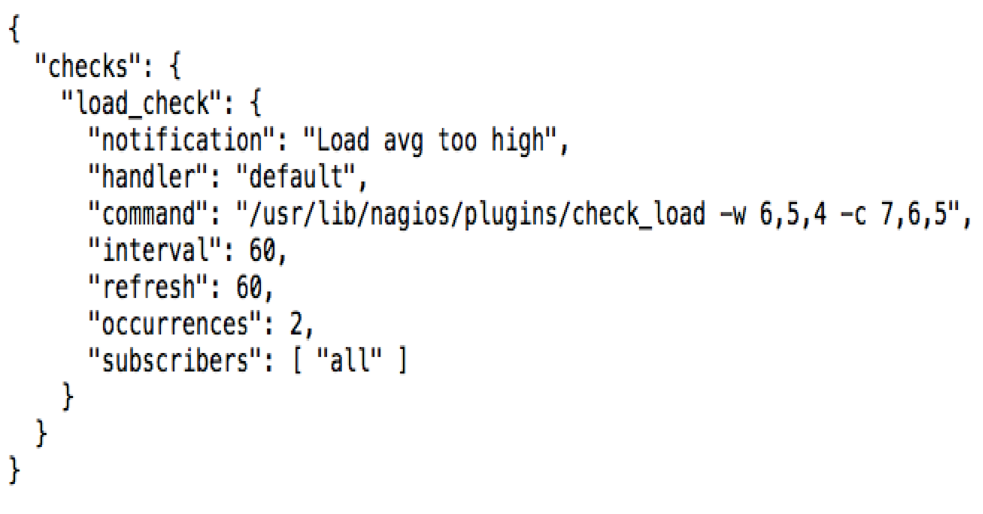


```
{
  "rabbitmq": {
    "host": "localhost",
    "port": 5671,
    "vhost": "/sensu",
    "user": "sensu",
    "password": "password",
    "ssl": {
      "cert_chain_file": "/etc/sensu/ssl/cert.pem",
      "private_key_file": "/etc/sensu/ssl/key.pem"
    }
  },
  "redis": {
    "host": "localhost",
    "port": 6379
  },
  "api": {
    "host": "localhost",
    "bind": "0.0.0.0",
    "port": 4567
  },
  "dashboard": {
    "bind": "0.0.0.0",
    "port": 8080,
    "user": "admin",
    "password": "secret"
  }
}
```

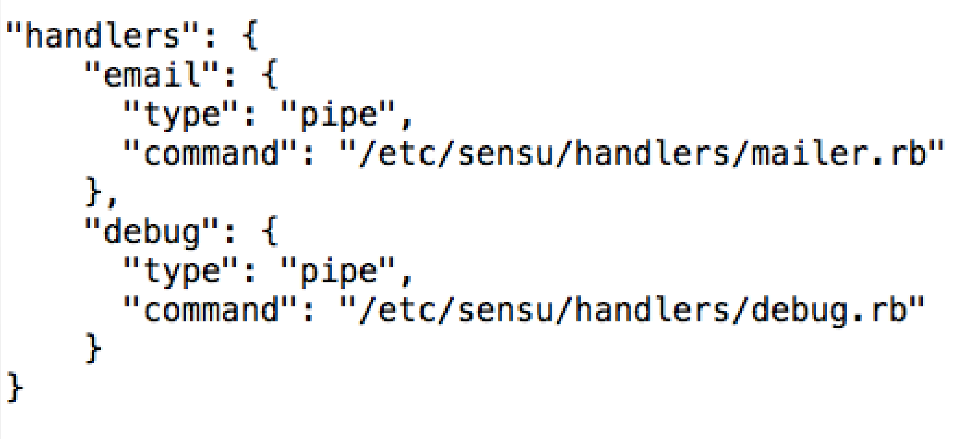


```
{
  "version": "0.9.12-3",
  "rabbitmq": {
    "host": "10.178.64.235",
    "port": 5671,
    "ssl": {
      "cert_chain_file": "/etc/sensu/ssl/cert.pem",
      "private_key_file": "/etc/sensu/ssl/key.pem"
    },
    "vhost": "/sensu",
    "user": "sensu",
    "password": "password",
    "heartbeat": 20
  },
  "client": {
    "name": "redis01.live.staging",
    "address": "94.236.40.116",
    "subscriptions": [
      "all",
      "redis"
    ],
    "environment": "staging"
  }
}
```

Checks and Handlers



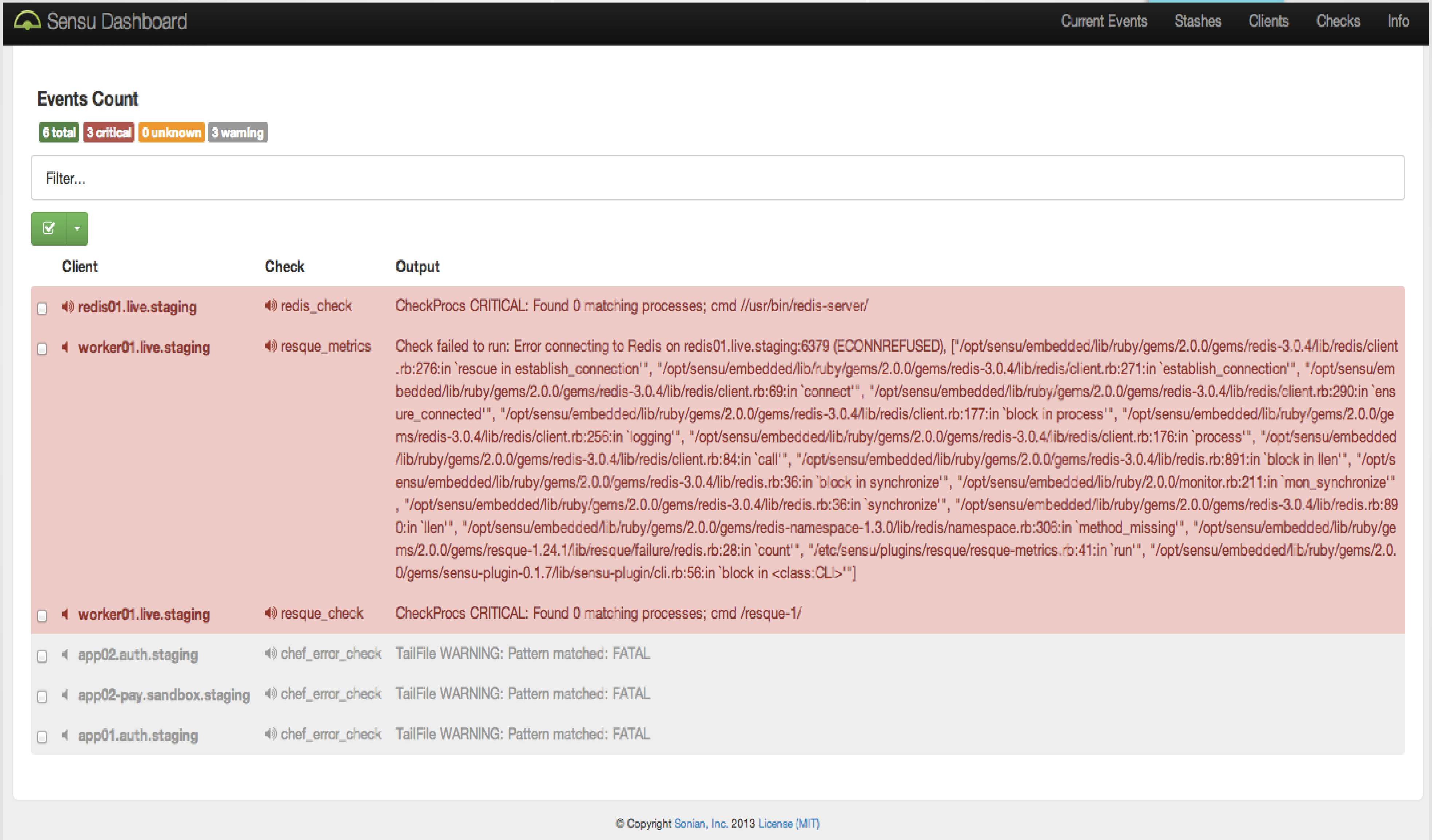
```
{
  "checks": {
    "load_check": {
      "notification": "Load avg too high",
      "handler": "default",
      "command": "/usr/lib/nagios/plugins/check_load -w 6,5,4 -c 7,6,5",
      "interval": 60,
      "refresh": 60,
      "occurrences": 2,
      "subscribers": [ "all" ]
    }
  }
}
```



```
"handlers": {
    "email": {
      "type": "pipe",
      "command": "/etc/sensu/handlers/mailer.rb"
    },
    "debug": {
      "type": "pipe",
      "command": "/etc/sensu/handlers/debug.rb"
    }
}
```

- Same check can have multiple subscribers → 1:N relationship
- Familiar check API – exit status indicates success or failure
- Check parameters are accessible in handlers via *event* hash (*event['check']['param']*)
- Handler sets – send check result to multiple handlers
- Loads of community handlers and plugins up for grabs (IRC,Pagerduty, etc.) or write your own very easily !

Sensu Dashboard



Events Count

6 total 3 critical 0 unknown 3 warning

Filter...

☑ ▾

	Client	Check	Output
□	redis01.live.staging	redis_check	CheckProcs CRITICAL: Found 0 matching processes; cmd //usr/bin/redis-server/
□	worker01.live.staging	resque_metrics	Check failed to run: Error connecting to Redis on redis01.live.staging:6379 (ECONNREFUSED), ["/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:276:in `rescue in establish_connection'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:271:in `establish_connection'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:69:in `connect'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:290:in `ensure_connected'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:177:in `block in process'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:256:in `logging'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:176:in `process'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis/client.rb:84:in `call'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis.rb:891:in `block in llen'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis.rb:36:in `block in synchronize'", "/opt/sensu/embedded/lib/ruby/2.0.0/monitor.rb:211:in `mon_synchronize'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis.rb:36:in `synchronize'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-3.0.4/lib/redis.rb:890:in `llen'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/redis-namespace-1.3.0/lib/redis/namespace.rb:306:in `method_missing'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/resque-1.24.1/lib/resque/failure/redis.rb:28:in `count'", "/etc/sensu/plugins/resque/resque-metrics.rb:41:in `run'", "/opt/sensu/embedded/lib/ruby/gems/2.0.0/gems/sensu-plugin-0.1.7/lib/sensu-plugin/cli.rb:56:in `block in <class:CLI>'"]
□	worker01.live.staging	resque_check	CheckProcs CRITICAL: Found 0 matching processes; cmd /resque-1/
□	app02.auth.staging	chef_error_check	TailFile WARNING: Pattern matched: FATAL
□	app02-pay.sandbox.staging	chef_error_check	TailFile WARNING: Pattern matched: FATAL
□	app01.auth.staging	chef_error_check	TailFile WARNING: Pattern matched: FATAL

Cool use cases

- Raise an alert from your app - POST JSON to local sensu-client socket:

  ```
  echo '{"handlers": ["default"], "name" : "my_check", "output" : "SOMETHING WENT WRONG DUDE !", "status" : 2}'|nc -w1 127.0.0.1 3030
  ```

- “Dump” metrics from your app – POST JSON to local sensu-client socket (similar to the above)
- Force check from command line on the host which has access to sensu-api:

  ```
  curl -XPOST http://sensu-api-host:4567/check/request -d '{ "check": "redis_process", "subscribers": [ "all"] }'
  ```

- Distribute alerts to different teams (add extra parameters to check definitions and do the logic in handler)
- Clear/Raise alert chains from within handler itself

  ```
  curl -XPOST http://sensu-api-host:4567/event/resolve -d '{ "client": "some_client_host", "check": "redis_process"}'
  ```

- Many other including the ones you will come up with! :-)

Links + Live Demo

- http://portertech.ca/2011/11/01/sensu-a-monitoring-framework/
- http://joemiller.me/2012/01/19/getting-started-with-the-sensu-monitoring-framework/
- https://github.com/sensu
- http://sensuapp.org/
- http://vimeo.com/59417953
- http://www.youtube.com/watch?v=aJGQSXZrxeY
- IRC: freenode #sensu