Monitoring Drupal with Sensu

John VanDyk
Iowa State University
DrupalCorn Iowa City
August 10, 2013

- What is Sensu?
- Sensu architecture
- Sensu server
- Sensu client
- Drupal and Sensu

Q: What is Sensu?

A: A monitoring router

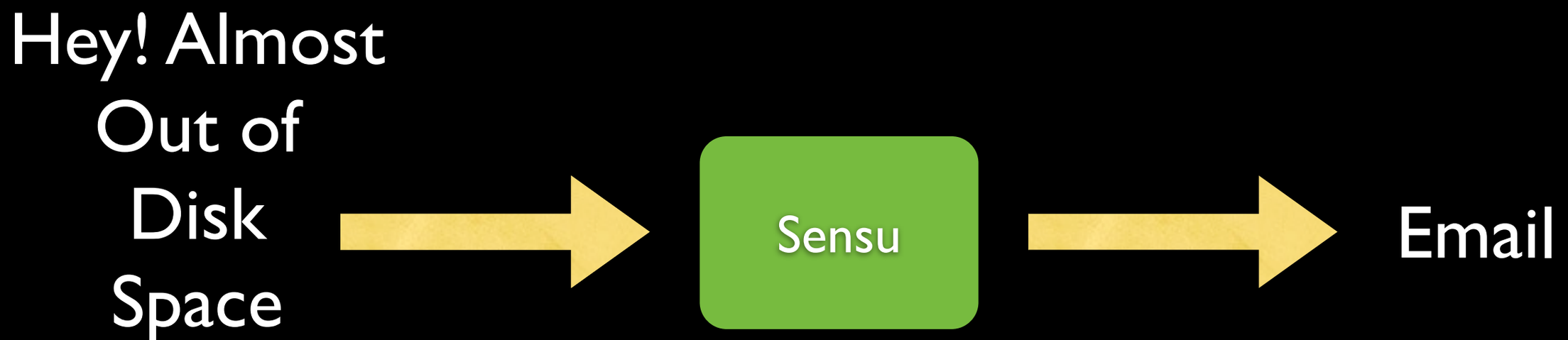
A monitoring router

Disk Space

Sensu

Sensu is focused on monitoring; that is watching something to make sure it's OK.

A monitoring router

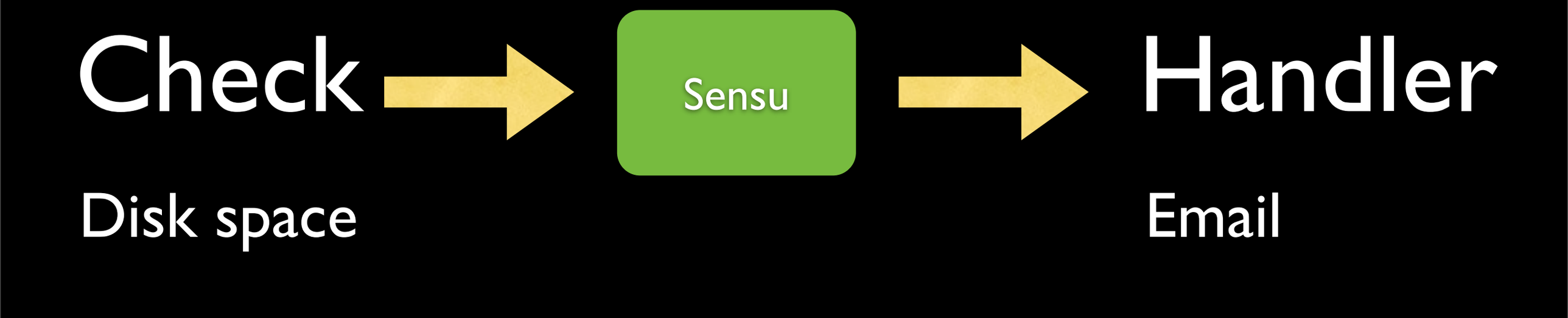


It also handles routing.

A monitoring router

Check results can be routed to the appropriate alerting system.

Sensu receives results from a check and gives them to a handler.

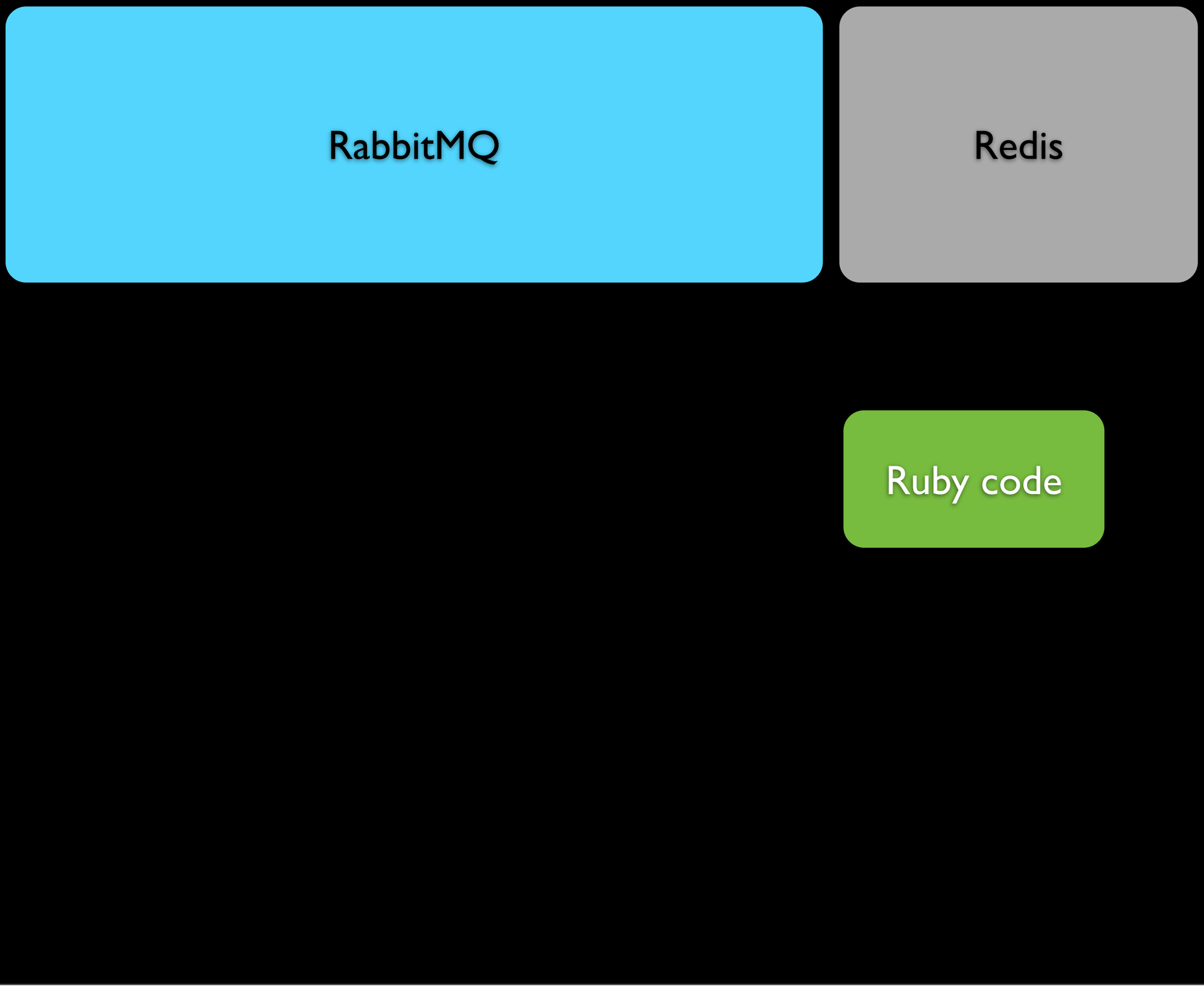

Check
Sensu
Handler
Disk space
Email

yawn

So what's special?

What's special is very little client configuration, JSON everywhere, understandable code and little of it, and route all the things.

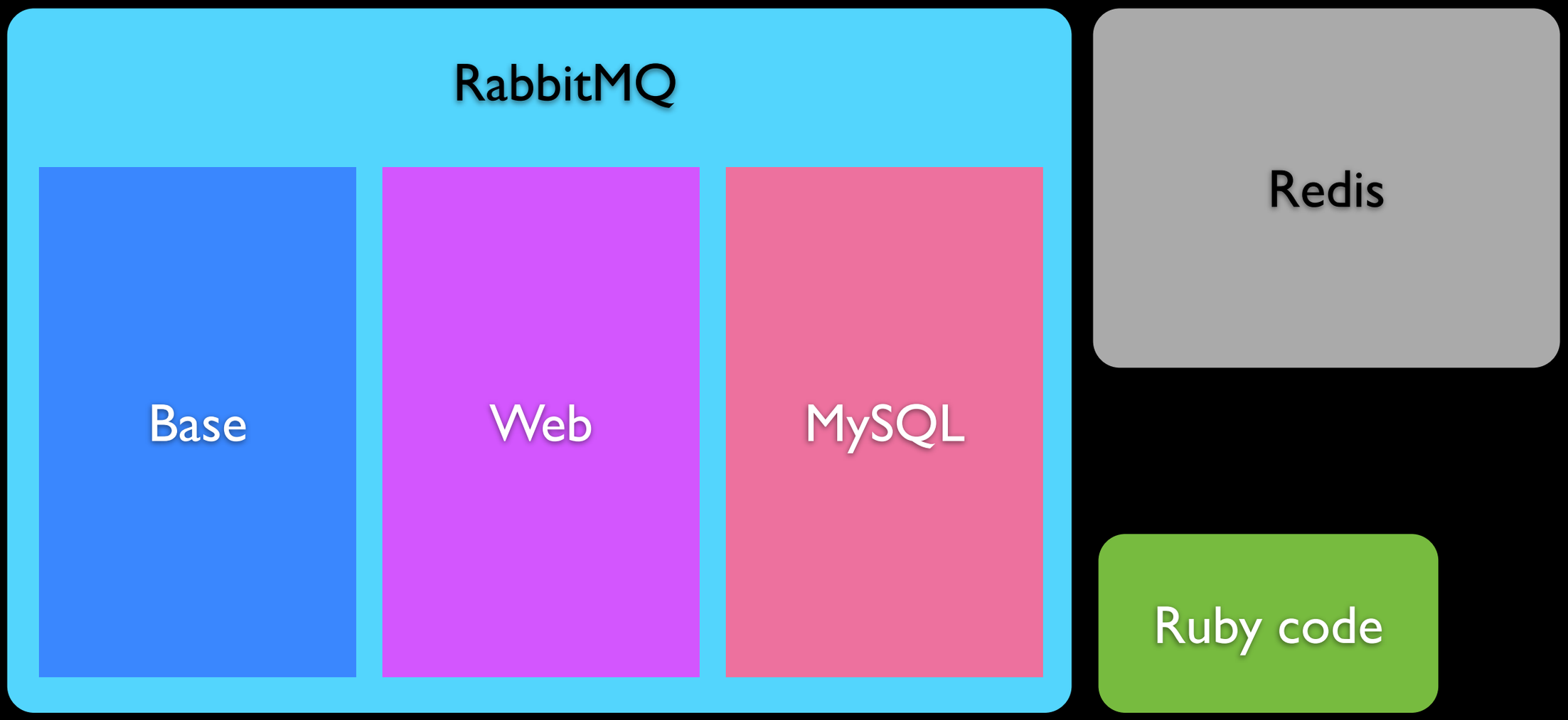
Sensu Architecture



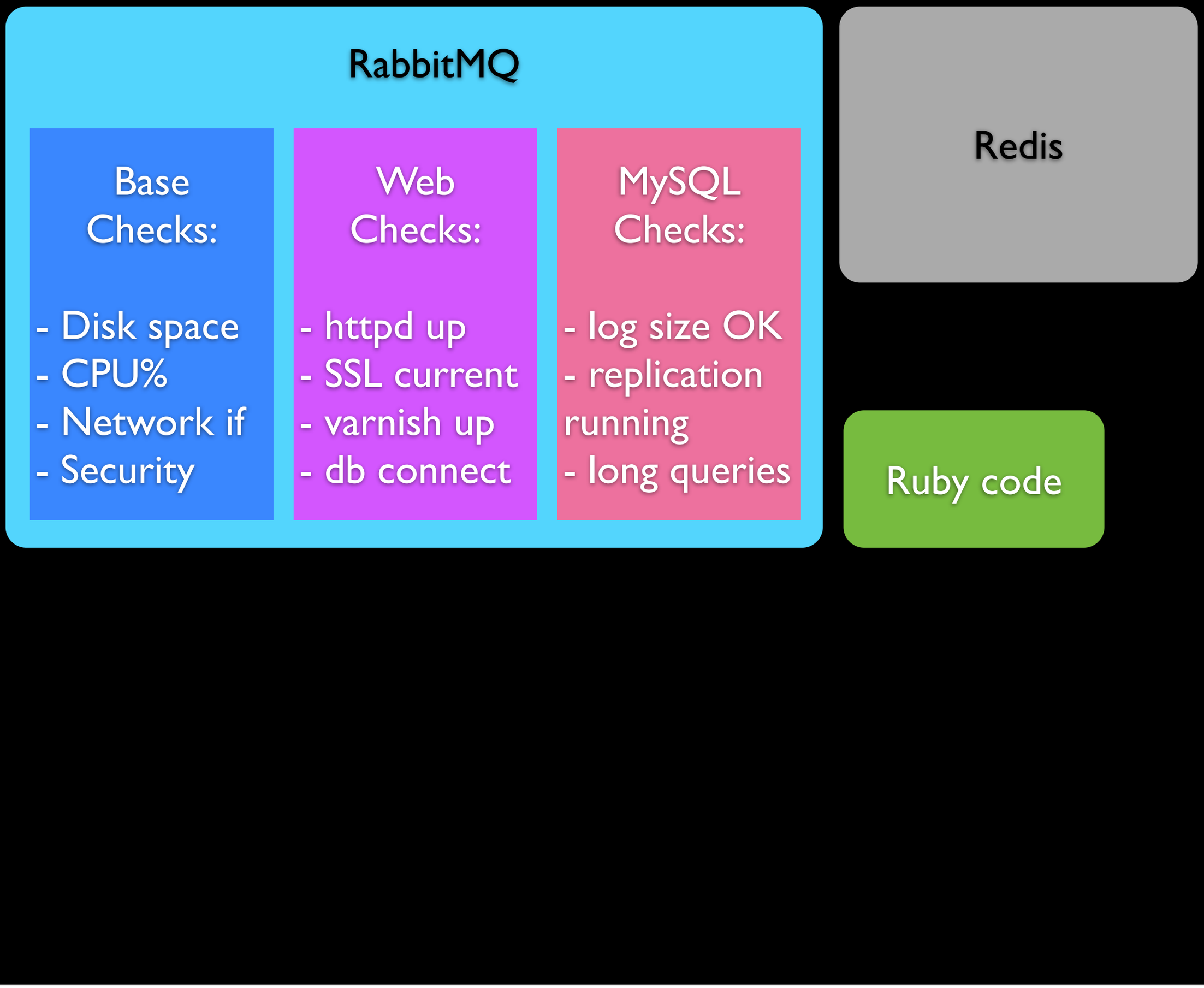
Sensu consists of a message queue, a key/value store, and ruby code.

RabbitMQ is a message broker that implements Advanced Message Queueing Protocol (AMQP). Written in Erlang, acquired by SpringSource which was then acquired by VMWare.

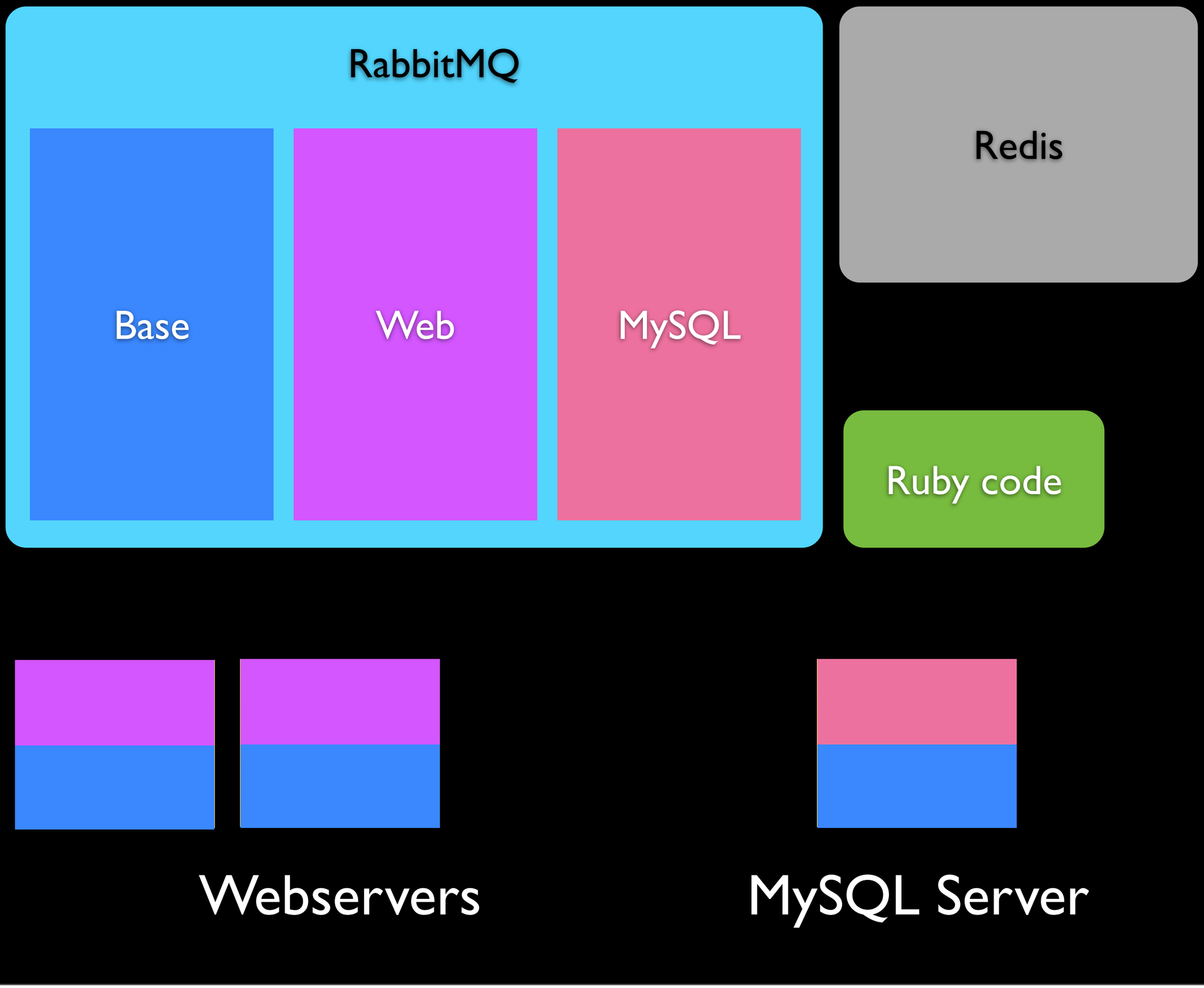
Redis is an in-memory key/value store sponsored by VMWare.



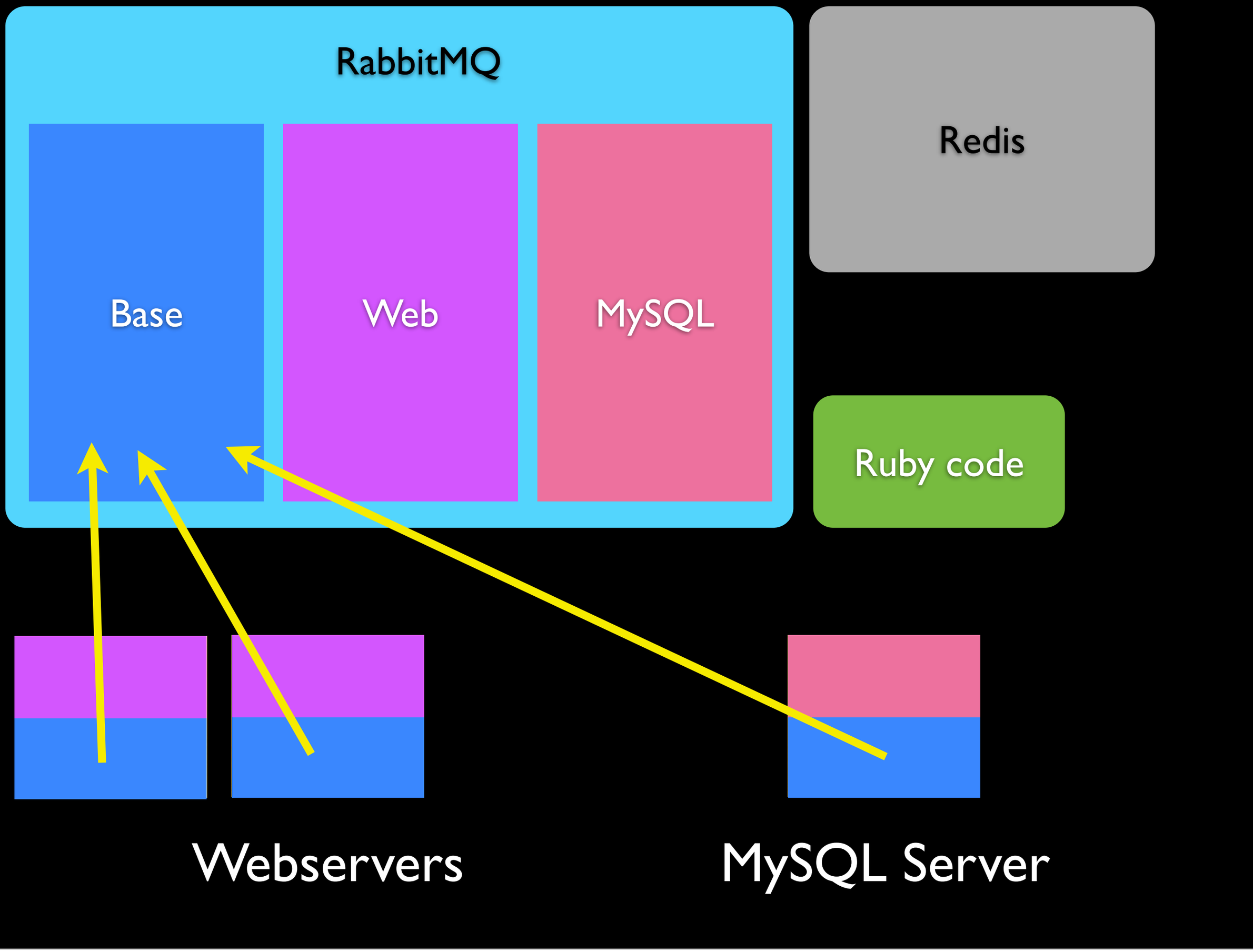
Sensu creates queues in RabbitMQ.



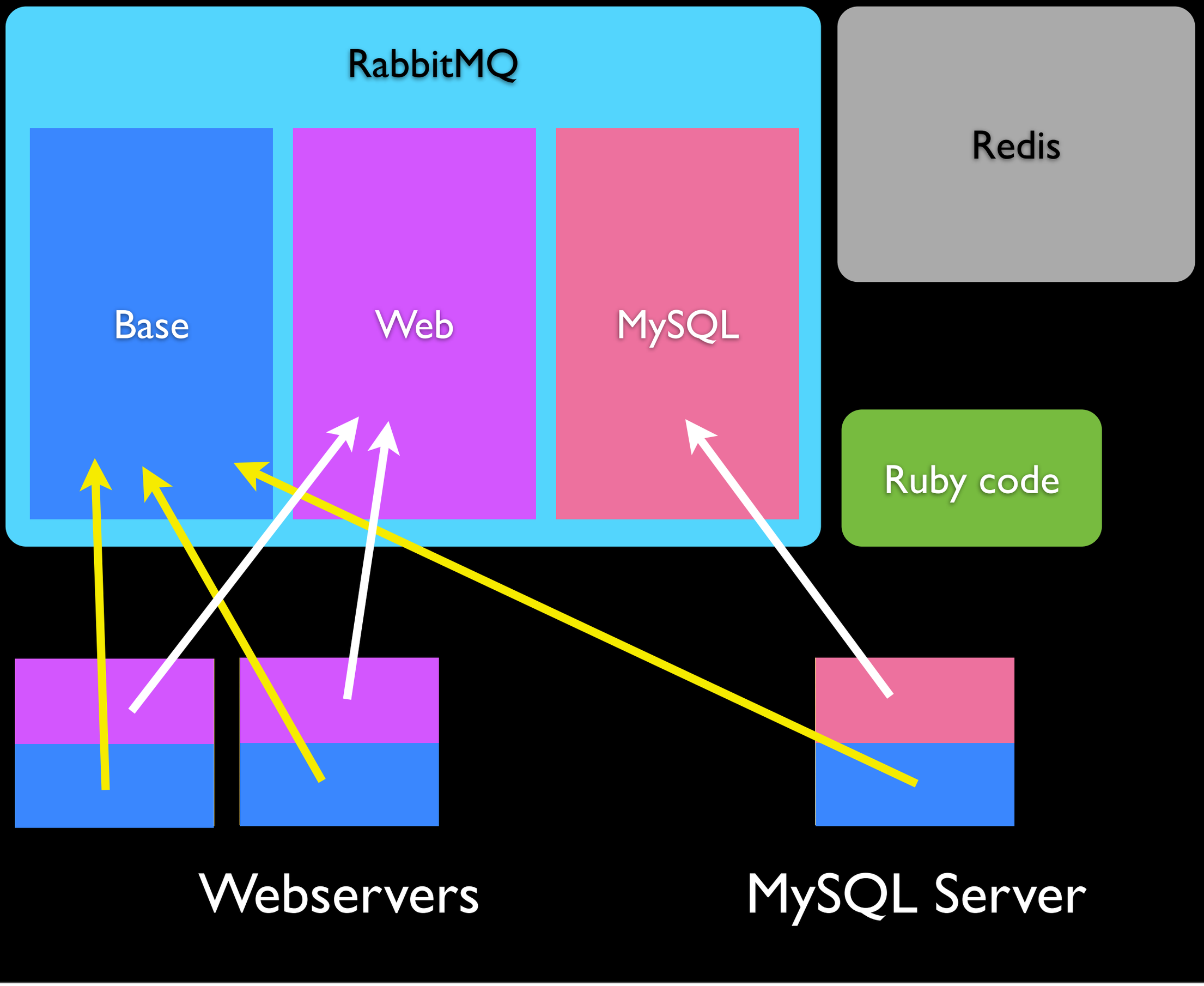
Sensu server sends checks into the queues.



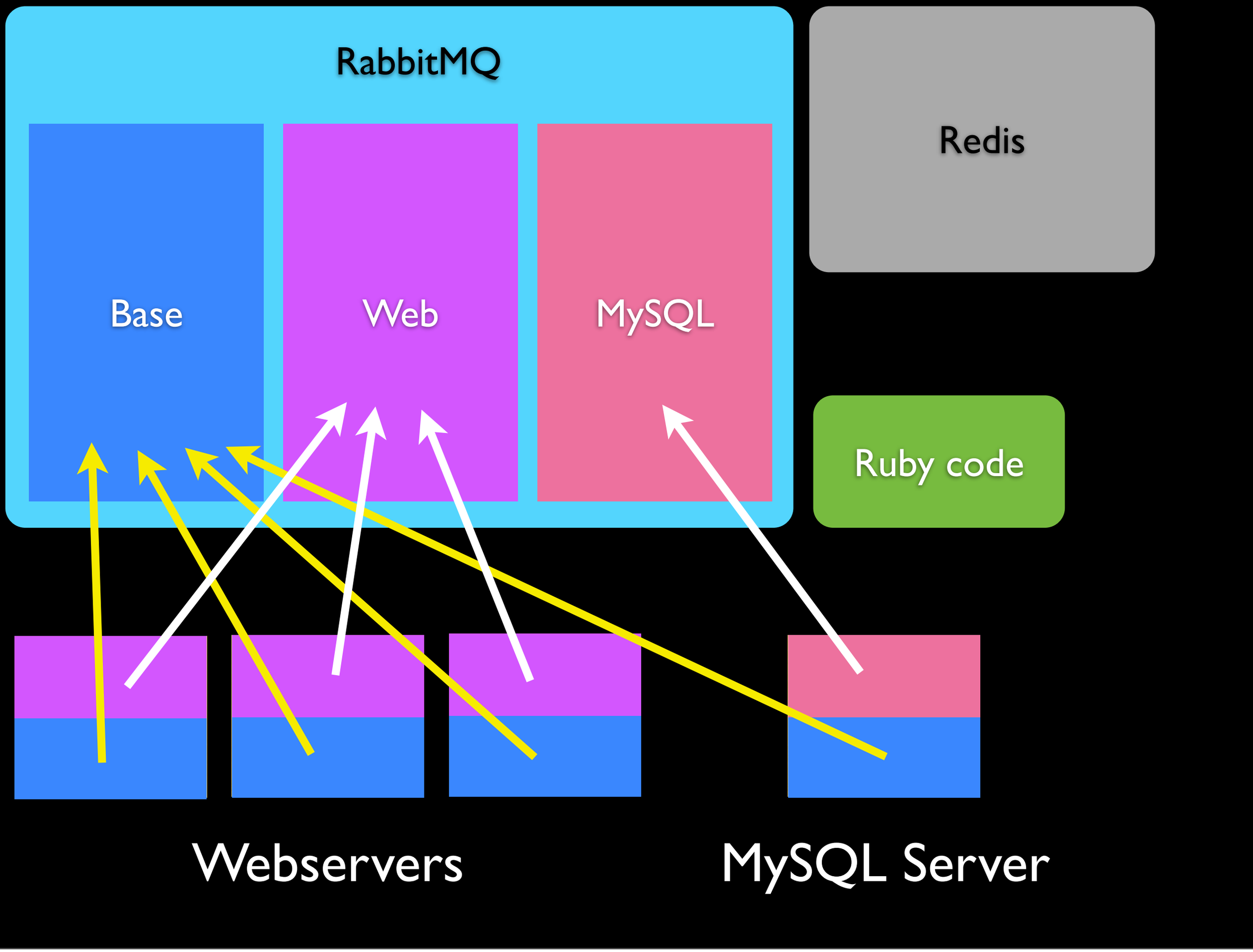
Servers subscribe to one more more queues.



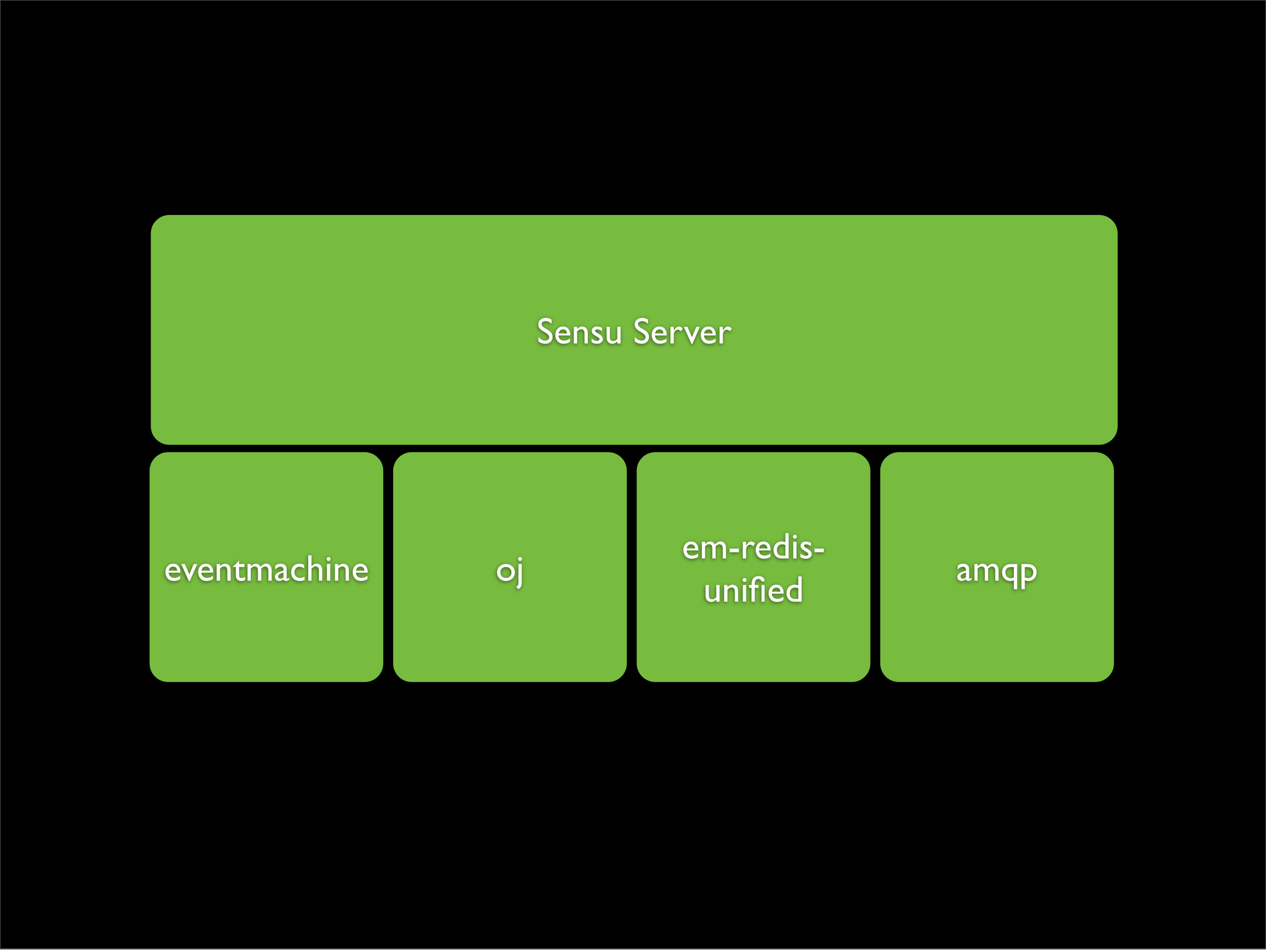
All servers can subscribe to a base queue.



A server may subscribe to any number of appropriate queues.

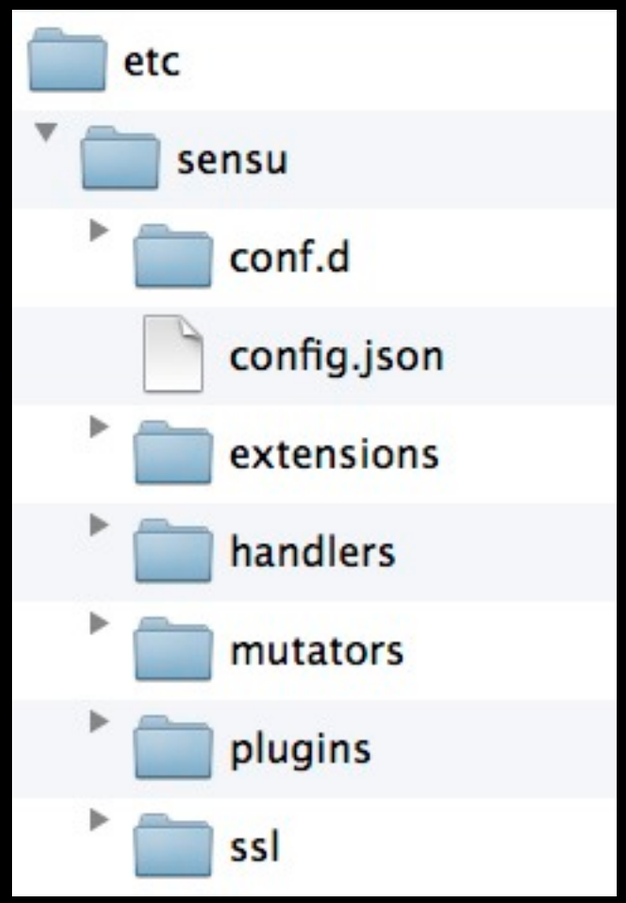


Adding monitoring for a new server is as simple as subscribing it to the proper queues.



OJ is the ruby Optimized JSON parser, a C extension to Ruby.

Eventmachine is a library providing the Reactor design pattern which demultiplexes incoming events, dispatching them to individual handlers.



Sensu configuration directory structure.

/etc/sensu/conf.d/rabbitmq.json

```
{
  "rabbitmq": {
    "user": "sensu",
    "port": 5671,
    "password": "secret",
    "vhost": "/sensu",
    "host": "sensuserver.example.com",
    "ssl": {
      "private_key_file": "/etc/sensu/ssl/key.pem",
      "cert_chain_file": "/etc/sensu/ssl/cert.pem"
    }
  }
}
```

Example of Sensu client configuration. Sets location of RabbitMQ so client can find queues to receive checks and publish results.

/etc/sensu/conf.d/client.json

```
{
  "client": {
    "name": "www.example.com",
    "address": "203.0.113.5",
    "safe_mode": false,
    "subscriptions": [
      "base",
      "drupal"
    ]
  }
}
```

Example of Sensu client configuration. Client identifies itself and says which queues it wants to subscribe to.

Safe mode means checks must be defined on client as well as server, preventing an evil server from running arbitrary code as checks on the client.

Example:

Checking that Drupal version is current

1. Create plugin (code that check will execute on client when check runs)

2. Define check configuration

3. Determine appropriate handler(s)

General methodology.

1. Create plugin (code that check will execute on client when check runs)

```
drush @nrem.live core-status drupal-version --pipe
```

Output:

```
7.20
```

1. Create plugin (code that check will execute when check runs)

/etc/sensu/plugins/check_drupal_current.sh

```
#!/bin/bash

# Sensu check for Drupal current-ness.
#
# Example: check_drupal_current.sh @example.alias

DRUPAL_VERSION=`drush $@ core-status drupal-version --pipe`
DRUPAL_MAJOR_VERSION=${DRUPAL_VERSION:0:1}

CURRENT_VERSION=$(</etc/drush/constants/currentdrupal${DRUPAL_MAJOR_VERSION}.txt)

if [ $DRUPAL_VERSION != $CURRENT_VERSION ]
then
  echo "VERSION WARNING - $@ $DRUPAL_VERSION"
  exit 1
else
  echo "VERSION OK - $@ $DRUPAL_VERSION"
  exit 0
fi
```

2. Define check configuration on server

```
{
  "checks": {
    "check_drupal_current": {
      "handlers": [ "mailer" ],
      "command":
"/etc/sensu/plugins/check_drupal_current.sh @nrem.live",
      "subscribers": [ "drupal" ],
      "interval": 3600
      "refresh": 86400
    }
  }
}
```

Interval means "run this check every this-many seconds."
Refresh means "number of seconds handlers should wait before taking second action."

3. Determine appropriate handler(s)

```
{
  "checks": {
    "check_drupal_current": {
      "handlers": [ "mailer" ],
      "command":
"/etc/sensu/plugins/check_drupal_current.sh @nrem.live",
      "subscribers": [ "drupal" ],
      "interval": 3600
      "refresh": 86400
    }
  }
}
```

Avoid hardcoding drush alias

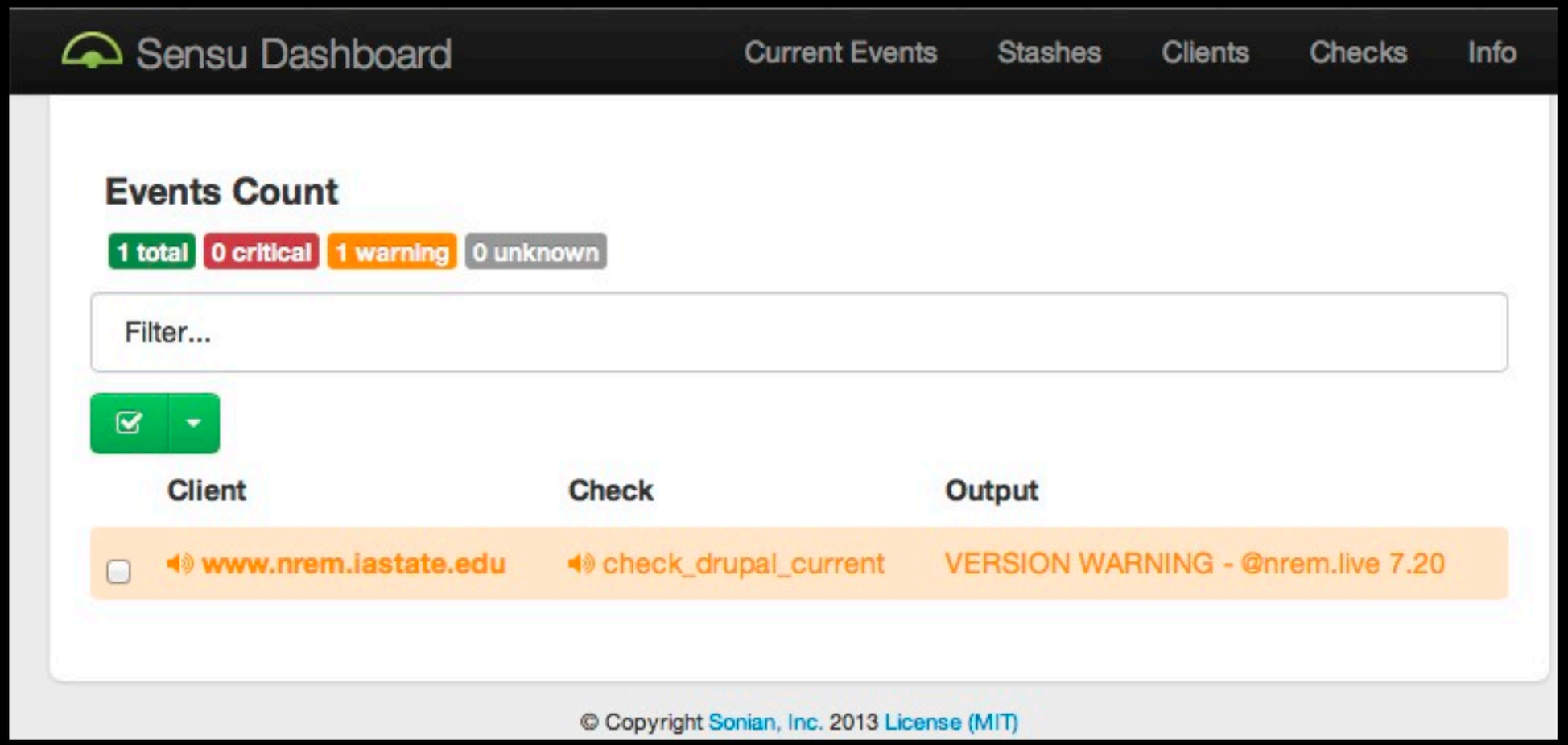
```
{
  "checks": {
    "check_drupal_current": {
      "handlers": [ "mailer" ],
      "command":
"/etc/sensu/plugins/check_drupal_current.sh :::drushalias:::",
      "subscribers": [ "drupal" ],
      "interval": 3600
      "refresh": 86400
    }
  }
}
```

Use a placeholder instead of hardcoding.

/etc/sensu/conf.d/client.json

```
{
  "client": {
    "name": "www.example.com",
    "address": "203.0.113.5",
    "safe_mode": false,
    "subscriptions": [
      "base",
      "drupal"
    ],
    "drushalias": "@nrem.live" ,
  }
}
```

Arbitrary tokens can be set on the client in the client configuration JSON.


Sensu Dashboard
Current Events
Stashes
Clients
Checks
Info
Events Count
1 total
0 critical
1 warning
0 unknown
Filter...
Client
Check
Output
www.nrem.iastate.edu
check_drupal_current
VERSION WARNING - @nrem.live 7.20
© Copyright Sonian, Inc. 2013 License (MIT)

Example:

Checking a Drupal Metric

1. Create plugin (code that check will execute on client when check runs)

2. Define check configuration

3. Determine appropriate handler(s)

Same methodology for metrics.

1. Create plugin (code that check will execute on client when check runs)

```
drush @nrem.live sql-query
"SELECT COUNT(lid) FROM linkchecker_link WHERE
fail_count > 0" | sed -n 2p
```

Output:

157

1. Create plugin (code that check will execute on client when check runs)

/etc/sensu/plugins/count_broken_drupal_links.sh

```
#!/bin/bash

# Sensu check for broken links in a Drupal site. Requires linkchecker Drupal
# module.
#
# Example: count_broken_drupal_links.sh @example.alias

BROKEN_LINK_COUNT=`drush $@ sql-query "SELECT COUNT(lid) AS broken FROM
linkchecker_link WHERE fail_count > 0" | sed -n 2p`
if [ $BROKEN_LINK_COUNT -gt 0 ]
then
  echo "LINK WARNING - $@ $BROKEN_LINK_COUNT broken links"
  exit 1
else
  echo "LINK OK - $@ no broken links"
  exit 0
fi
```

2. Define check configuration on server

```
{
  "checks": {
    "check_broken_drupal_links": {
      "type": "metric",
      "handlers": [ "mailer" ],
      "command": "/etc/sensu/plugins/
count_broken_drupal_links.sh :::drushalias:::",
      "subscribers": [ "drupal" ],
      "interval": 3600
      "refresh": 86400
    }
  }
}
```

If you declare the check as a metric the results will be sent to the server even if the exit value is 0.

3. Determine appropriate handler(s)

```
{
  "checks": {
    "check_broken_drupal_links": {
      "type": "metric",
      "handlers": [ "graphite" ],
      "command": "/etc/sensu/plugins/
count_broken_drupal_links.sh :::drushalias:::",
      "subscribers": [ "drupal" ],
      "interval": 3600
    }
  }
}
```

You might want to send your metrics to graphite.

Sensu Dashboard

Current Events | Stashes | Clients | Checks | Info

Events Count

2 total | 0 critical | 2 warning | 0 unknown

Filter...

Client	Check	Output
www.nrem.iastate.edu	check_drupal_current	VERSION WARNING - @nrem.live 7.20
www.nrem.iastate.edu	count_broken_drupal_links	LINK WARNING - @nrem.live 157 broken links

Pushing Info from Drupal to Sensu

(demo of logtosensu
Drupal module)

```
{
"timestamp":"2013-08-09T16:43:56.060628-0500",
"level":"info",
"message":"publishing check result",
  "payload":
  { "client":"www.example.com",
    "check":
    { "name":"watchdog",
      "output":"page: updated Foo.",
      "status":0,
      "handler":["default", "mailer"],
      "drupal_base_url":"http://local.dev/d723",
      "drupal_timestamp":1376084636,
      "drupal_type":"content",
      "drupal_ip":"127.0.0.1",
      "drupal_request_uri":"http://local.dev/d723/node/1/edit",
      "drupal_referer":"http://local.dev/d723/node/1/edit",
      "drupal_uid":"1",
      "drupal_link":"view",
      "type":"metric",
      "issued":1376084636
      }
  }
}
```


JSON event resulting from updating a page in Drupal.

http://sensuapp.org

Much, much more:

Any test, including Nagios scripts

Any Drupal metric

Any handler

Mutators (not covered here) may change data before handler gets it

http://drupal.org/project/logtosensu