About

- Customer Success Engineer at Sensu
- Avid Banjo player
- Homebrewer
- Twitter: asachs01

Objectives

- Components of Sensu deployment
- Sensu configuration building blocks
  - Checks
  - Filters
  - Mutators
  - Handlers
- Put the pieces together and have your own monitoring solution
- Follow along at: http://bit.ly/intro2sensu

Before We Start...

```
git clone https://github.com/sensu/training-vagrant.git
cd training-vagrant/workshops/intro-to-sensu
vagrant up
```

Monitoring

- What is monitoring?
- How is it different than observability?
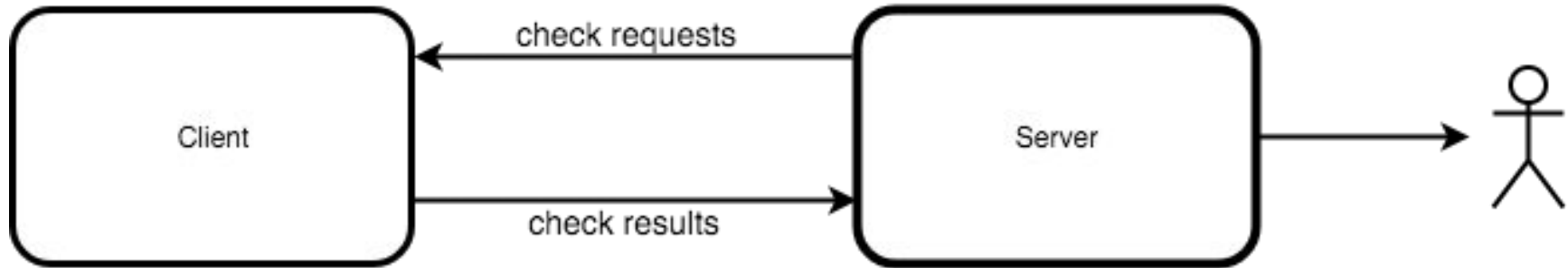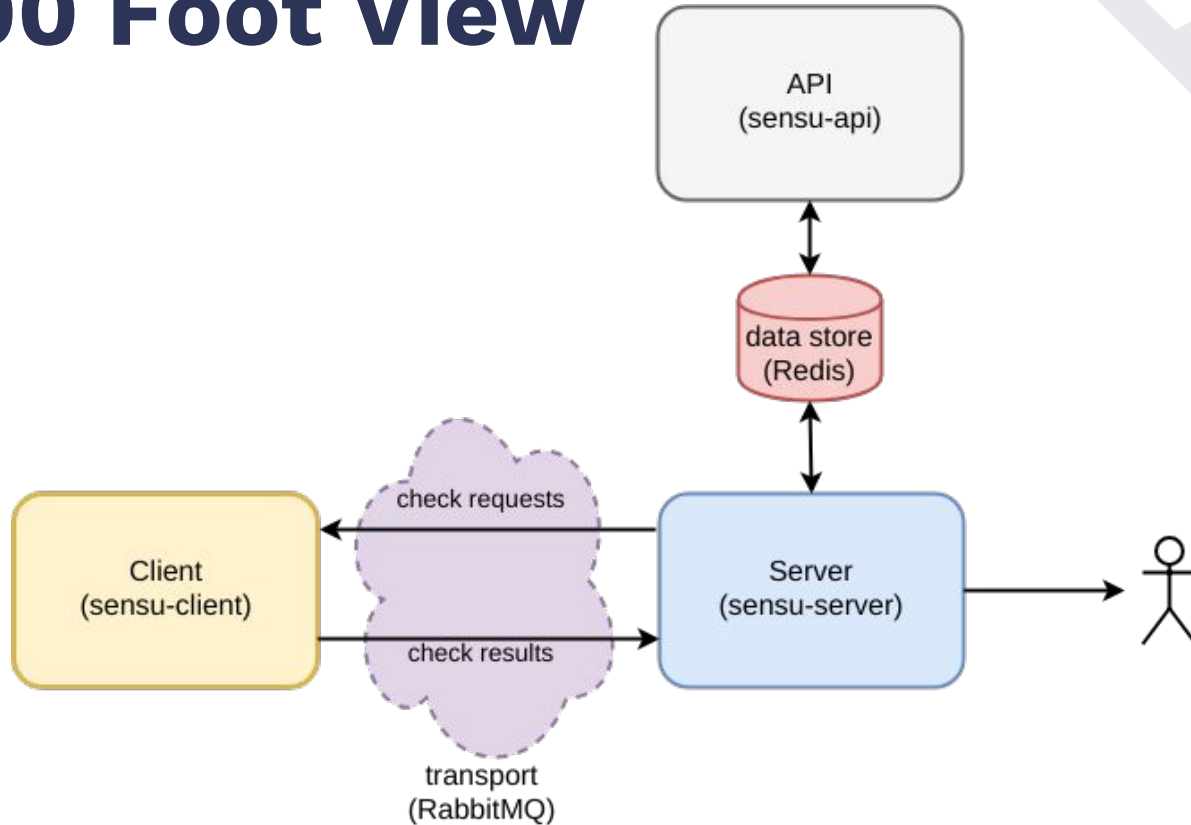- Why should I monitor something?

Sensu - Overview

- A composable monitoring framework aimed at obviating the need to (re)build custom monitoring solutions.
- Translation?
- Difference from other monitoring software?

50,000 foot view
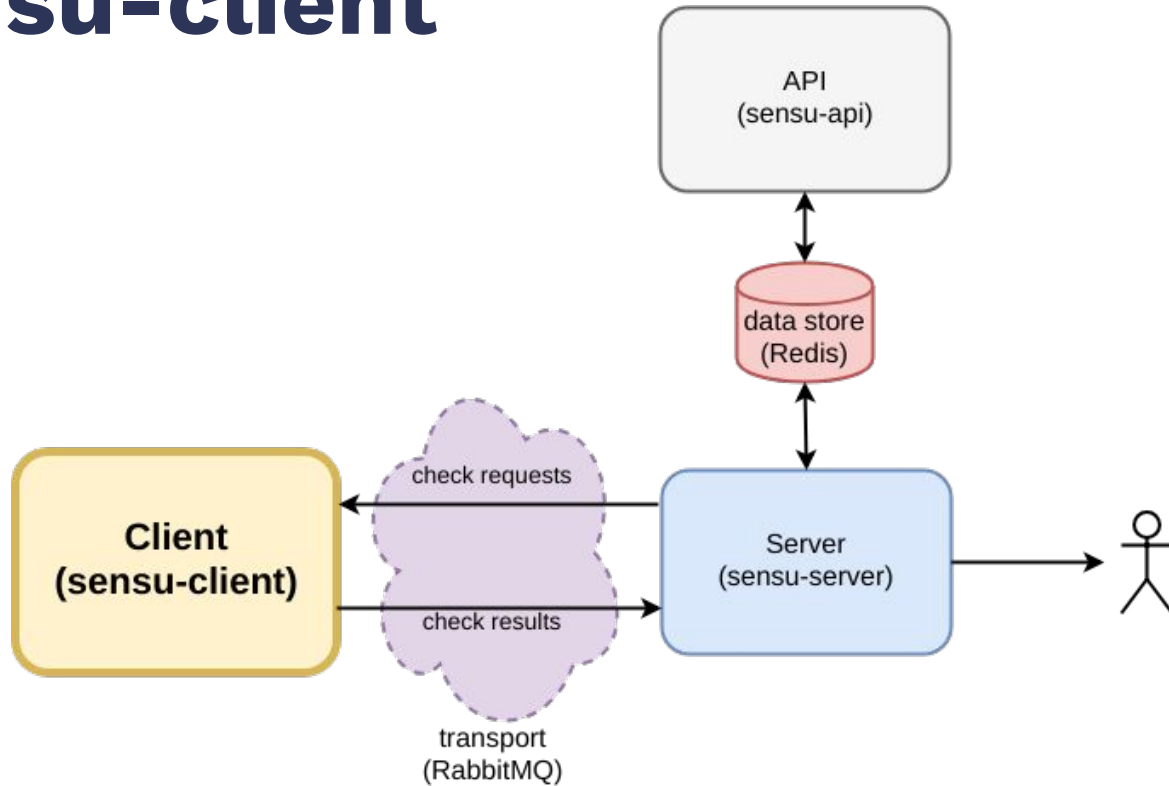
15,000 Foot View

Components

sensu-client

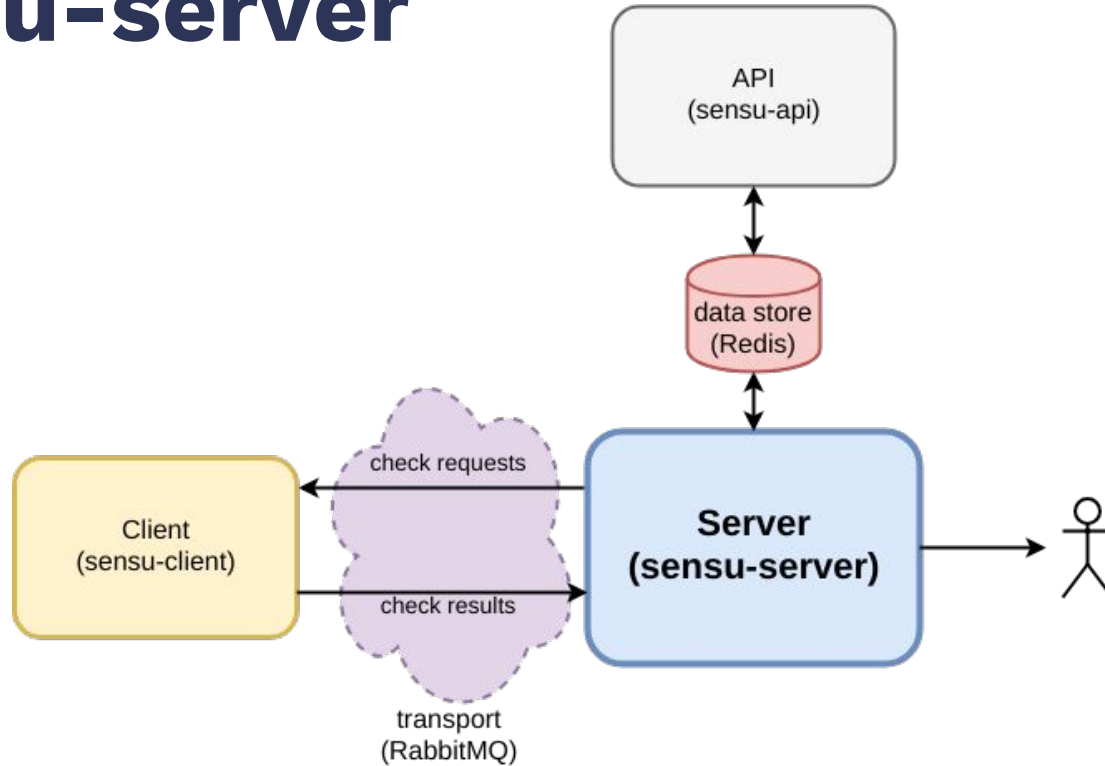# Example Client Config

```json
{
  "client": {
    "name": "i-274135e0",
    "address": "54.188.127.142",
    "subscriptions": [
      "production",
      "load_balancer"
    ]
  }
}
```

Example Transport Config

```json
{
  "rabbitmq": {
    "host": "127.0.0.1",
    "port": 5672,
    "vhost": "/sensu",
    "user": "sensu",
    "password": "secret"
  }
}
```
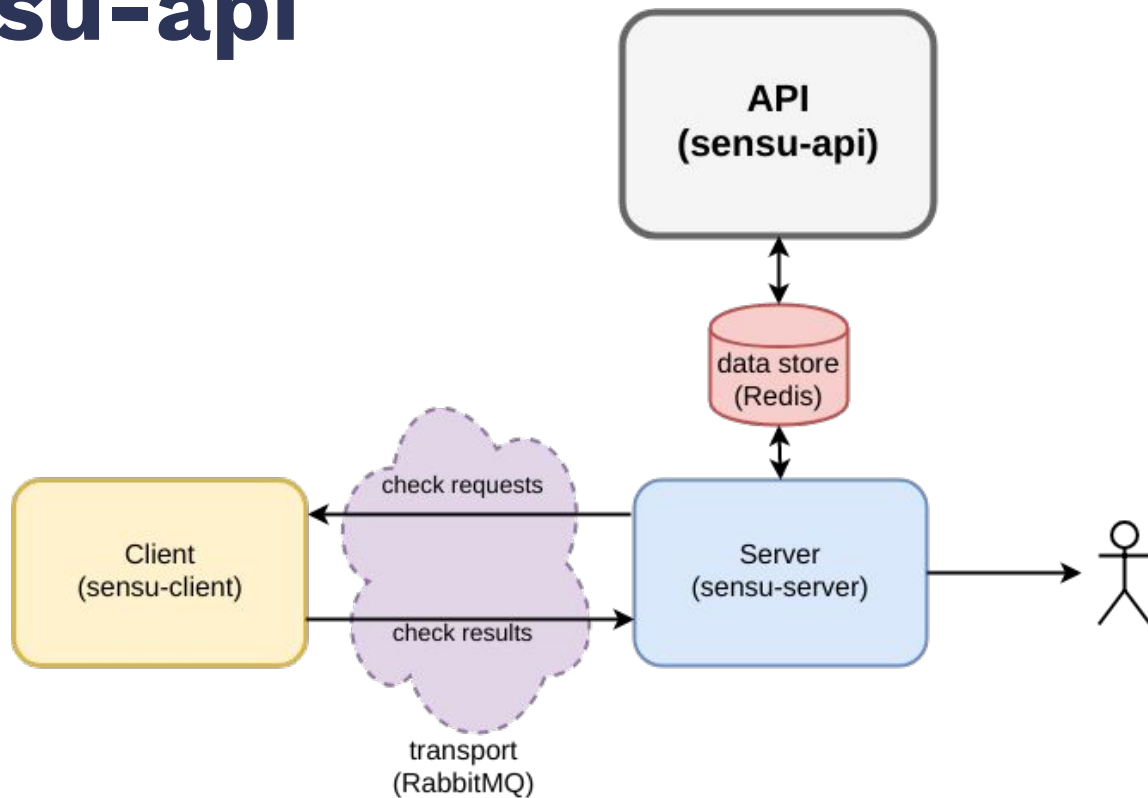
sensu-server

Example Server Config

```json
{
  "api": {
    "host": "127.0.0.1",
    "port": 4567
  },
  "transport": {
    "name": "rabbitmq"
  }
}
```

sensu-api

Example API Config

```
{
  "api": {
    "host": "57.43.53.22",
    "bind": "0.0.0.0",
    "port": 4567
  }
}
```

Transport

Transport – ctd

Check Results
(Server as Consumer)

Results
Queue

Results
Exchange

Check Requests
(Client as Consumer)

Sensu Client A
webserver, linux

"webserver"
Exchange

Client A
Queue

Check Requests
(Server as Publisher)

"smtp"
Exchange

Client B
Queue

Sensu Client B
webserver, linux

Sensu Server

"linux"
Exchange

Client C
Queue

Sensu Client C
smtp, linux

Keepalives
Queue

Keepalives
Exchange

Keepalives
(Server as
Consumer)

RabbitMQ Broker

Keepalives
(Client as Publisher)

Data Store

Dashboard - Uchiwa

- The dashboard for Sensu Core
- Provides a way graphical to interact with Sensu
- It's an optional component, but it makes interacting with Sensu easier

uchiwa

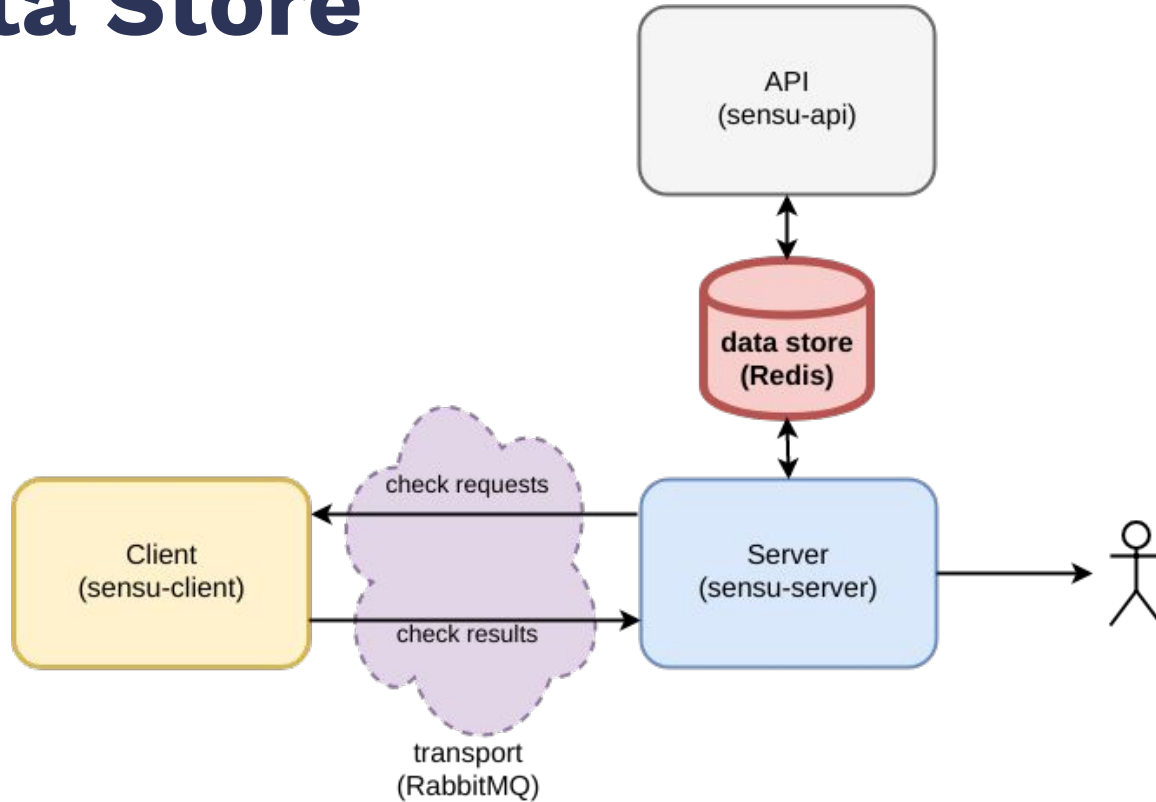|		
2		
46		
87		
2		
3		
1		

⊘ a few seconds ago
👤 CNBEBOP

_id	CNBEBOP/
address	
environment	prod
haproxy	{
	"host": "localhost",
	"password": "REDACTED",
	"port": 9090,
	"user": "cnmon"
	}
keepalive	{
	"handlers": [
	"slack",
	"email"
	]
	"refresh": 30,
	"thresholds": {
	"critical": 180,
	"warning": 180
	}
	}
name	
safe_mode	false
socket	{
	"bind": "127.0.0.1",
	"port": 3030
	}
subscriptions	all, linux, fail2ban, haproxy, keepalived
timestamp	2016-12-12 13:56:34
version	0.23.3

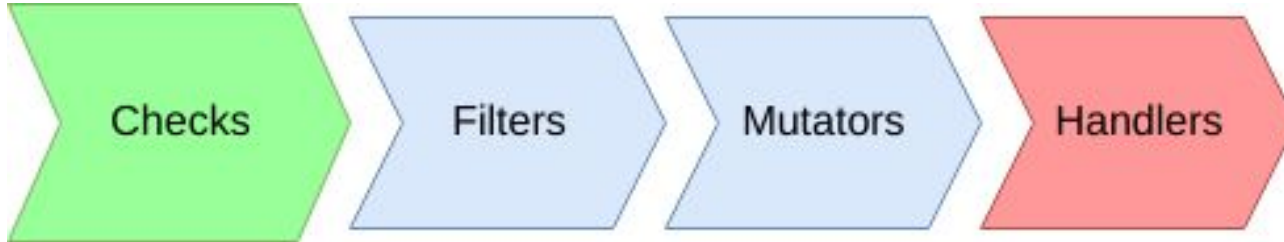| Check ⇕	Output ⇕	⊙ ⇕
check_crond	CheckProcess OK: Found 1 matching processes; cmd /crond/	a few seconds ago
check_keepalived_proc	CheckProcess OK: Found 3 matching processes; cmd /keepalived/	a few seconds ago
metrics-haproxy	haproxy.broker00.session_current 1567 148154720..	3 minutes ago
metrics-proc-mem-us...	proc_mem_usage.sensu.sensu-client 1030.92 1481..	5 minutes ago
check-cpu	CheckCPU TOTAL OK: total=10.06 user=1.44 nice=0.0 system=6.06 idle=89.94 ...	a minute ago
check-load	CheckLoad OK: Load average: 0.16, 0.21, 0.19	a few seconds ago
metrics-uptime	uptime.uptime 12367988.73 1481547207 ...	3 minutes ago
check-ntp	CheckNTP OK: NTP offset by 0.01ms	a few seconds ago
metrics-netif	lo.rx_kB_per_sec 2.93 1470207163 ...	4 months ago
check_failedaccess	OK: 0 failure login attempts(warning at 10, critical at 20)	a few seconds ago
check-hardware-fail	CheckHardwareFail OK: Hardware OK	a few seconds ago
metrics-iostat-extended	iostat.avg-cpu.pct_user 9.69 1481547147 ...	4 minutes ago
metrics-load	load_avg.one 0.36 1481547206 ...	3 minutes ago
metrics-puppet-run	puppet.resources.scheduled 0 1481547177 ...	4 minutes ago
check_ntpd	CheckProcess OK: Found 1 matching processes; cmd /ntpd/	a few seconds ago
check_haproxy_broker...	CheckHAProxy OK: UP: 100% of 3 /broker00/ services	a few seconds ago
metrics-memory	memory.total 1034092544 1481547149 ...	4 minutes ago
check_sshd	CheckProcess OK: Found 1 matching processes; cmd /sshd/	a few seconds ago
check-puppet-last-run	PuppetLastRun OK: Puppet last run 3342 seconds ago	a few seconds ago
check-disk-usage	CheckDisk OK: All disk usage under 85% and inode usage under 85%	a few seconds ago
metrics-memory-perc...	memory_percent.free 23.460770644527535 148154...	4 minutes ago
metrics-proc-cpu-usage	proc_cpu_usage.haproxy.haproxy 8.1 1481547206 ...	3 minutes ago

Event Processing Pipeline
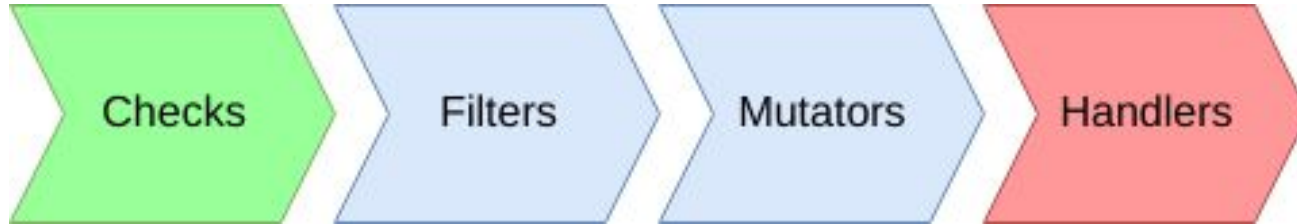
Checks

Event Processing Pipeline

Check Components

- Check plugin
- Check definition

Check Plugins

- Adhere to Nagios API
  - Exit status indicates status
  - Output to STDOUT provides context
- Simple to write and understand
- Can be in any language
- Thousands of existing plugins available

0	Normal/OK
1	Warning
2	Critical
3+	Unknown

Check Definition Attributes

Required

- `command`
- `subscribers` **or** `standalone`
- `interval` **or** `cron`

Frequently used

- `type`
- `handlers`

Advanced

- `source`
- `aggregates`
- `subdue`
- `hooks`
- `proxy_requests`

Example Check Config

```json
{
  "checks": {
    "check_haproxy": {
      "command": "check_haproxy.rb -s app01 -w40 -c25",
      "subscribers": ["load_balancer"],
      "interval": 30,
      "handlers": ["slack"]
    }
  }
}
```

Check Results

Example Check Result Message

```
{
  "client": {"name": "i-274135e0" },
  "check": {
    "name": "check_haproxy",
    "command": "check_haproxy.rb -s app01 -w 40 -c 25",
    "subscribers": ["load_balancer"],
    "interval": 30,
    "issued": 1445569640,
    "executed": 1445569641,
    "output": "UP - 85% of 10 /app01/ services",
    "status": 0,
    "duration": 0.87
  }
}
```

Check Result Processing

- Check results update state in the data store
  - History registry
  - Results registry
  - Events registry (non-OK results)
- Check results are combined with additional state to create events
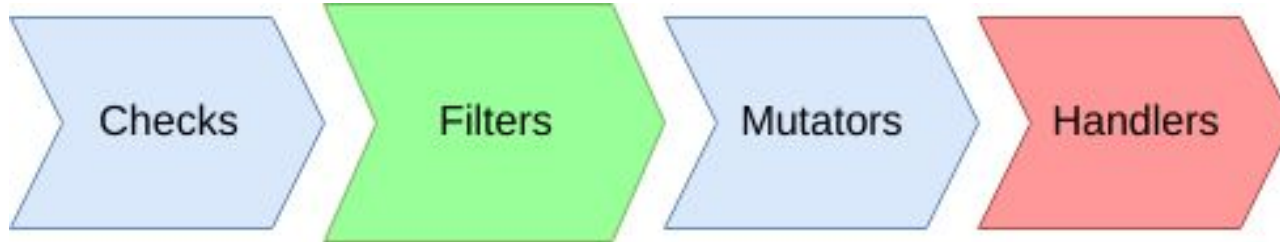- Events are processed by a configurable pipeline

Event Data

```json
{
  "id": "ef6b87d2-1f89-439f-8bea-33881436ab90",
  "action": "create",
  "timestamp": 1460172828,
  "occurrences": 1,
  "check": {
    "...": "...",
    "total_state_change": 0,
    "history": ["0","0","0","0","0","0"],
  },
  "client": { "...": "..."}
}
```

Filters

Event Processing Pipeline

Filters

Native filters

- User-defined via configuration
- Exact matching
- `eval()` matching

Filter extensions

- Implemented as Sensu extensions
- Some examples built into Sensu Core
  - occurrences
  - check_dependencies

Filter Definition Attributes

Native filter definitions may

- Combine eval and exact matching
- Apply only during specific time windows using `when` attribute
- Invert their logic using `negate` attribute

Exact Matching Native Filter

```json
{
  "filters": {
    "env_production": {
      "attributes": {
        "client": {"environment": "production"}
      },
      "when": { "days": {
          "all": [{
            "begin": "5:00 PM",
            "end": "8:00 AM"}]
        }
      }
    }
  }
}
```

Attribute Evaluating Filter

```json
{
  "filters": {
    "state_change_only": {
      "negate": true,
      "attributes": {
        "check": {
          "history": "eval: value.last == value[-2]"
        }
      }
    }
  }
}
```
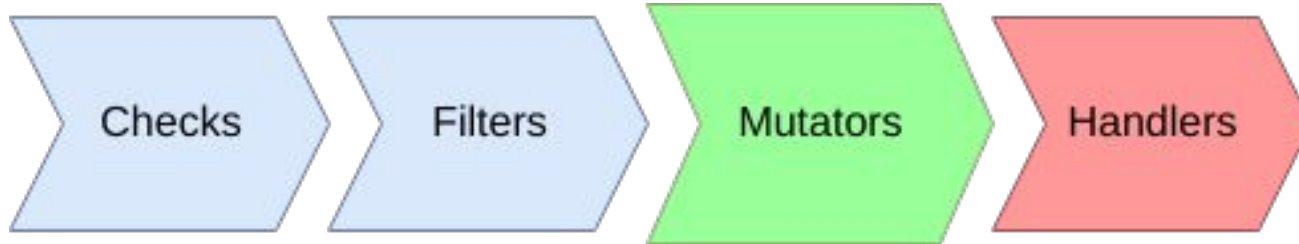
Mutators

Event Processing Pipeline

Mutator Components

- Mutator plugin
- Mutator definition

Mutator Plugins

- Receive event data on STDIN
- Emit mutated event data on STDOUT

0	Normal/OK

1+	Critical

Example Mutator Config
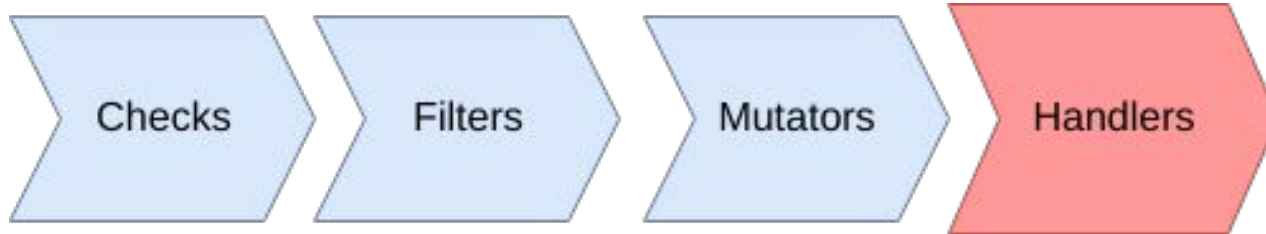
```
{
  "mutators": {
    "graphite": {
      "command": "mutator-graphite"
    }
  }
}
```

Handlers

Event Processing Pipeline

Checks → Filters → Mutators → Handlers

Handler Components

- Handler definition
- Handler plugin (maybe)

Handler Types

Pipe

- Executed by Sensu server
- Event data sent to stdin
- Most powerful type

Set

- List of handlers
- Pipeline spawned for each
- Tip: consider making your "default" a set

TCP, UDP

- Sends event data to a remote host/service
- Each event spawns a new connection

Transport

- Reuses connection to message bus
- Queues data for an external process to consume

Handler Attributes

These attributes are common across handler types

- filters
- severities
- mutator
- timeout
- handle_flapping
- handle_silenced

Handler Plugins

- Plugins are executables run by server
- Very simple to understand
- Very simple to write in any language
- Receive event data on STDIN
- Exit status indicates success/failure

| 0 | Normal/OK |

| 1+ | Critical |

UDP Handler Example

```json
{
  "handlers": {
    "graphite": {
      "type": "udp",
      "socket": {
        "host": "graphite.example.com",
        "port": 2003
      },
      "mutator": "graphite_events"
    }
  }
}
```

Pipe Handler Example

```json
{
  "handlers": {
    "pagerduty": {
      "type": "pipe",
      "command": "pagerduty.rb",
      "severities": ["critical"],
      "filters": ["occurrences"]
    }
  }
}
```

Resources

- Sensu
  - https://docs.sensu.io
  - https://docs.uchiwa.io
  - https://github.com/sensu-plugins
  - https://github.com/sensu/training-vagrant
  - https://slack.sensu.io
- Other Resources
  - Monitoring and Observability - Cindy Sridharan
  - Practical Monitoring - Mike Julian
  - The Art of Monitoring - James Turnbull

Q & A

BREAK TIME

Workshop Time

Getting Started

- Virtualbox
- Vagrant
- Git
- https://github.com/sensu/training-vagrant

Clone the Repo & Start the VM

```
git clone https://github.com/sensu/training-vagrant.git
cd training-vagrant/workshops/intro-to-sensu
vagrant up
```

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/centos-7.4"
  config.vm.hostname = "sensu-up-and-running-01"
  config.vm.network "private_network", type: "dhcp"
  config.vm.provision "shell", path: "../../scripts/sensu-and-graphite.sh"
  config.vm.provider "virtualbox" do |vb|
    vb.memory = 1024
  end
end
```

Access Vagrant Guest

```
vagrant ssh
```

For Starters

- Take a moment and explore
- Everything lives in /etc/sensu
- Head over to the dashboard and poke around.

Write your own…

- Check
  - Monitors a website of your choosing
- Filter
  - Only notifies you if **"workshop": true**
- Handler
  - Use mailer handler
  - http://bit.ly/goog-app-pass

Resources

- Sensu
  - https://docs.sensu.io
  - https://docs.uchiwa.io
  - https://github.com/sensu-plugins
  - https://github.com/sensu/training-vagrant
  - https://slack.sensu.io
- Other Resources
  - [Monitoring and Observability](#) - Cindy Sridharan
  - [Practical Monitoring](#) - Mike Julian
  - [The Art of Monitoring](#) - James Turnbull

Reach Out

Me:
- Aaron Sachs
  - Twitter: asachs01

Sensu:
- Sensu.slack.io - Community Slack Channel
- Feedback on the workshop/training questions
  - training@sensu.io

FIN