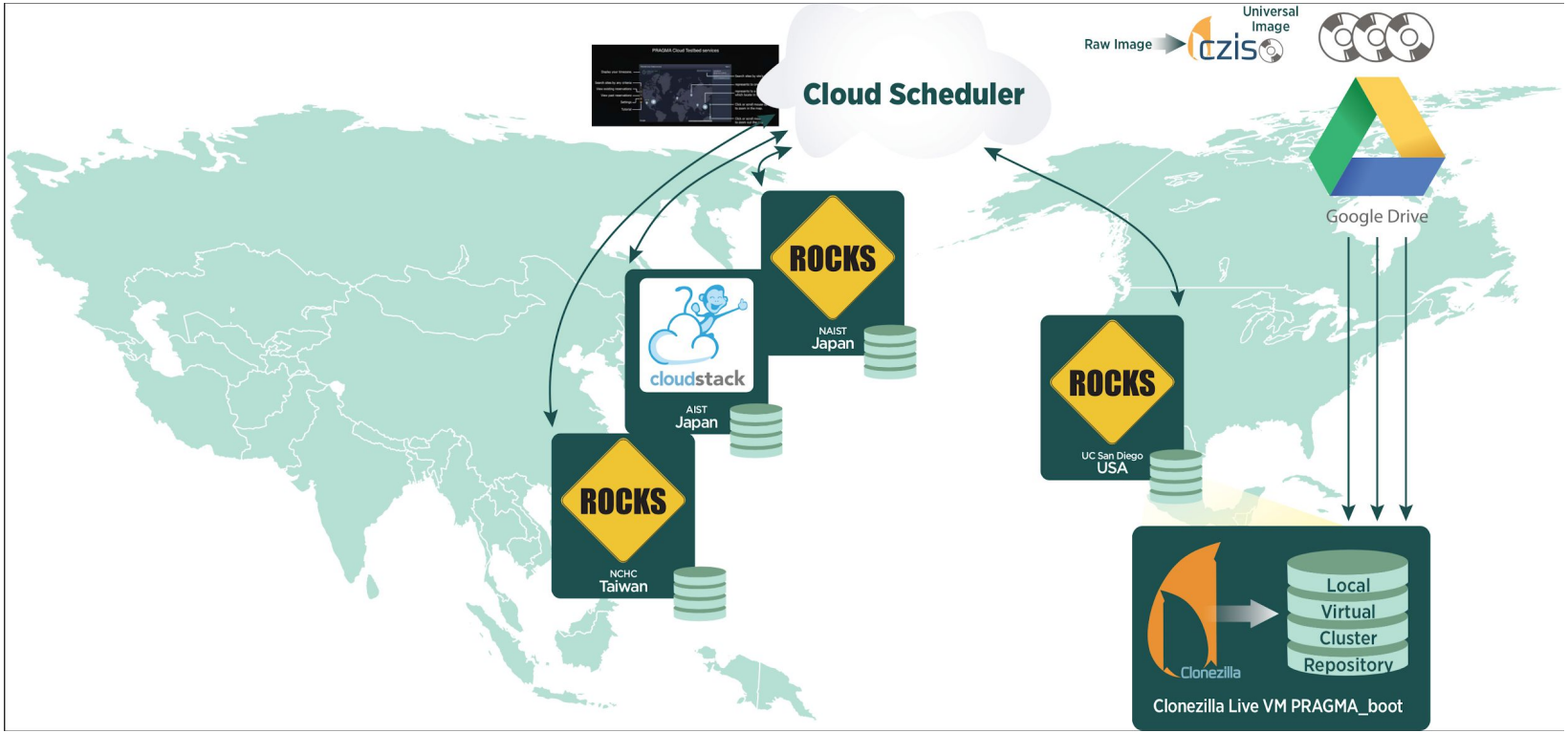
A real-time performance monitoring system for distributed cloud applications

Ayuth Mangmesap[1], Nitipat Wuttisasiwat[1]
Prapaporn Rattanatamrong[1], Jason Haga[2], Nadya Williams[3], Shava Smallen[3], Vahid Daneshmand[4]

[1]Thammasat University
[2]National Institute of Advanced Industrial Science and Technology (AIST)
[3]University of California, San Diego
[4]University of Florida

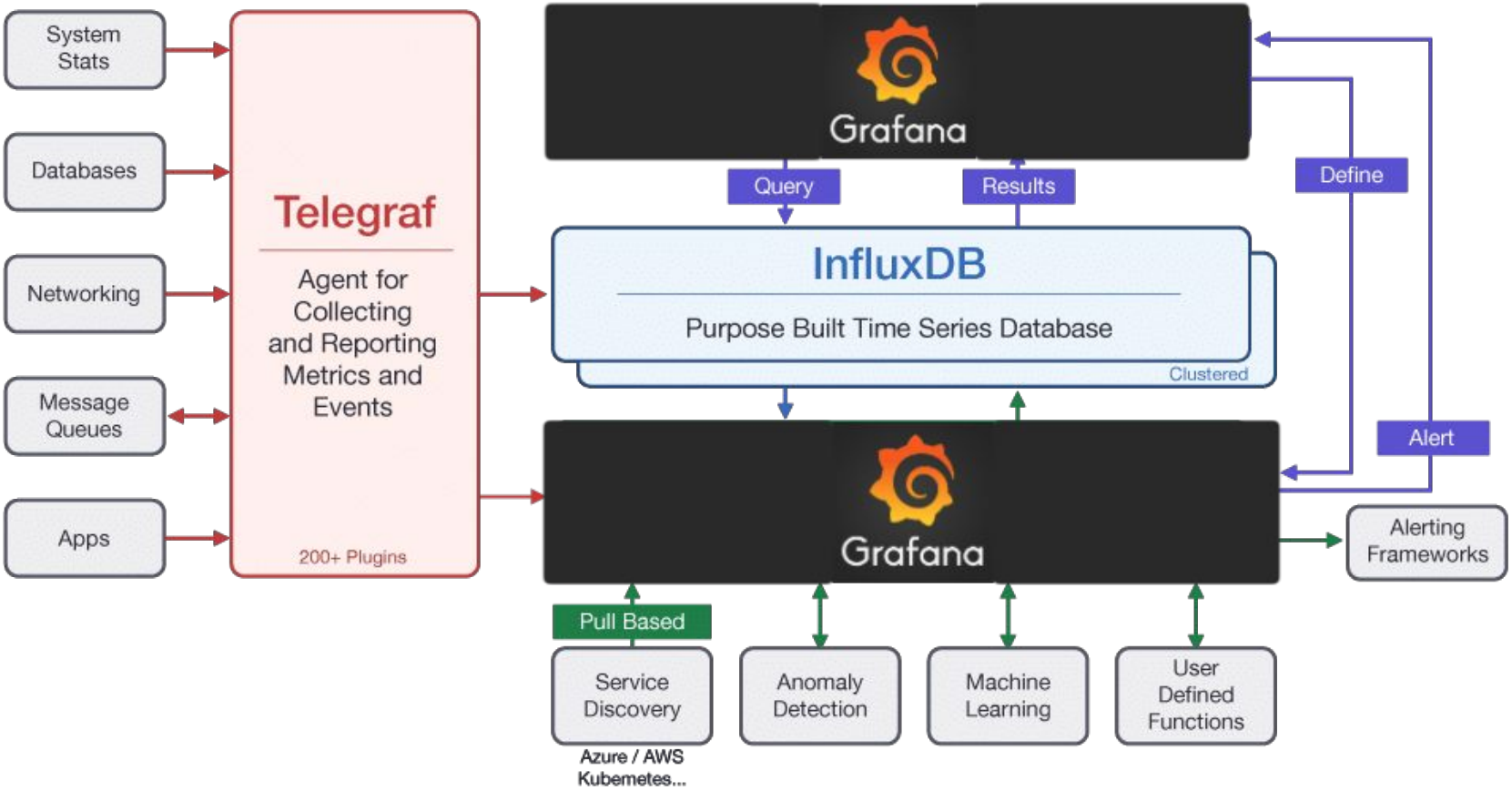
PRAGMA Cloud



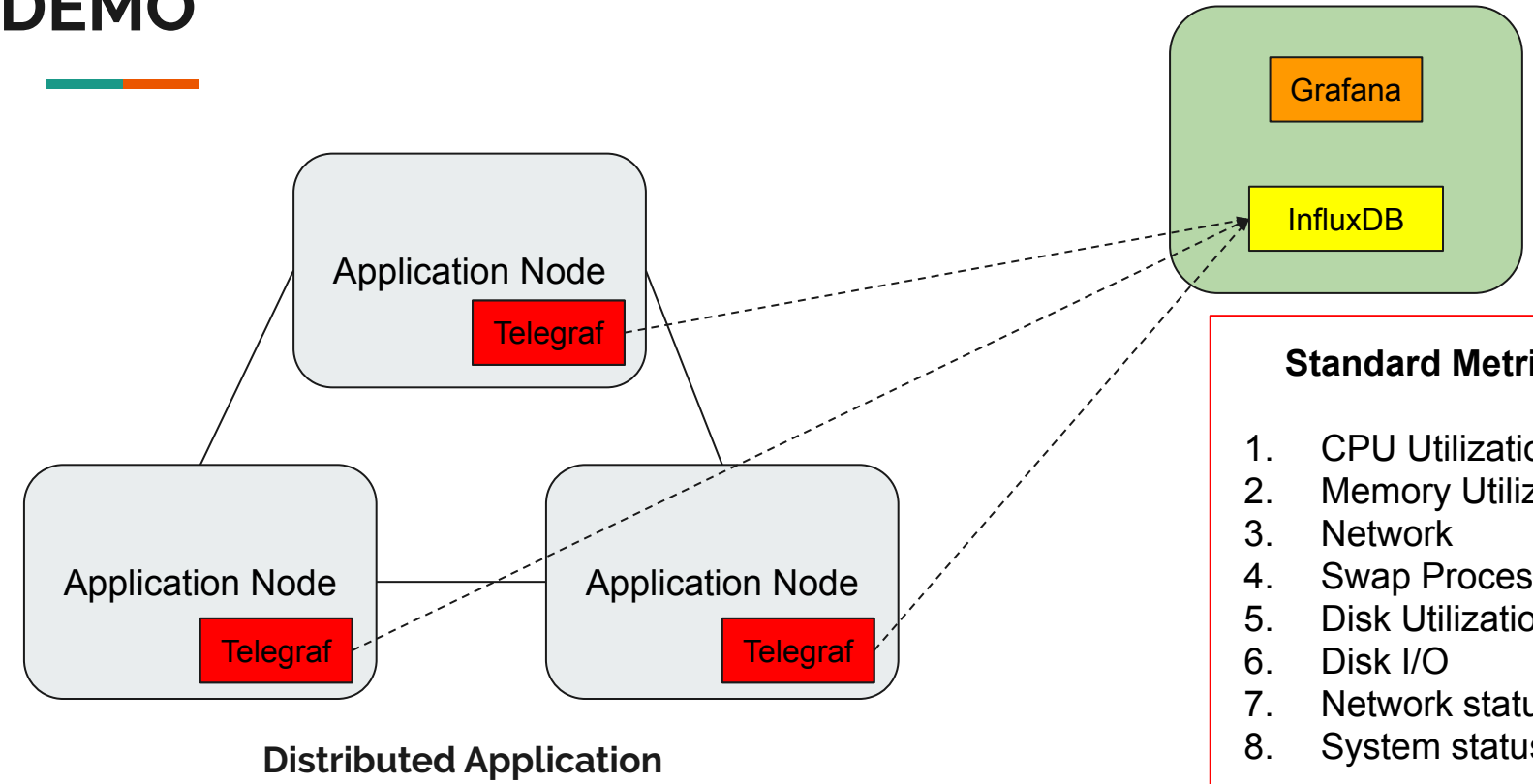
Challenges of Monitoring Distributed Apps.

1. Gathering data from different node, network and operating systems.
2. Visualize data to the right group of people (System admin, Scientist and Anonymous)
3. Easy to setup and scalable.

Software Stack (TIG)



DEMO



Standard Metrics

1. CPU Utilization
2. Memory Utilization
3. Network
4. Swap Process
5. Disk Utilization
6. Disk I/O
7. Network status
8. System status

Telegraf, normally installation steps

1. Add the influxdata repository

Add the influxdata repository.

```
source /etc/lsb-release
echo "deb https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" | sudo tee /et
c/apt/sources.list.d/influxdb.list
```

2. Update package manager repository

```
sudo yum update
```

3. Install telegraf

```
sudo yum install telegraf -y
```

4. Generate configuration file from input metrics

```
sh -c "telegraf --input-filter $TELEGRAF_INPUTS
    --output-filter influxdb config | tee >
        /etc/telegraf/telegraf.conf"
```

```
vim /etc/telegraf/telegraf.conf"
```

5. Configure another value such as database url, username, password, port by vim

6. Start/Restart telegraf service

```
systemctl restart telegraf
```

Telegraf, simplify installation steps

1. Download telegraf installation script

```
curl -O https://raw.githubusercontent.com/pragmagrid/tstat/master/TIG/telegraf-install/telegraf-install.sh
```

2. Prepare configuration file

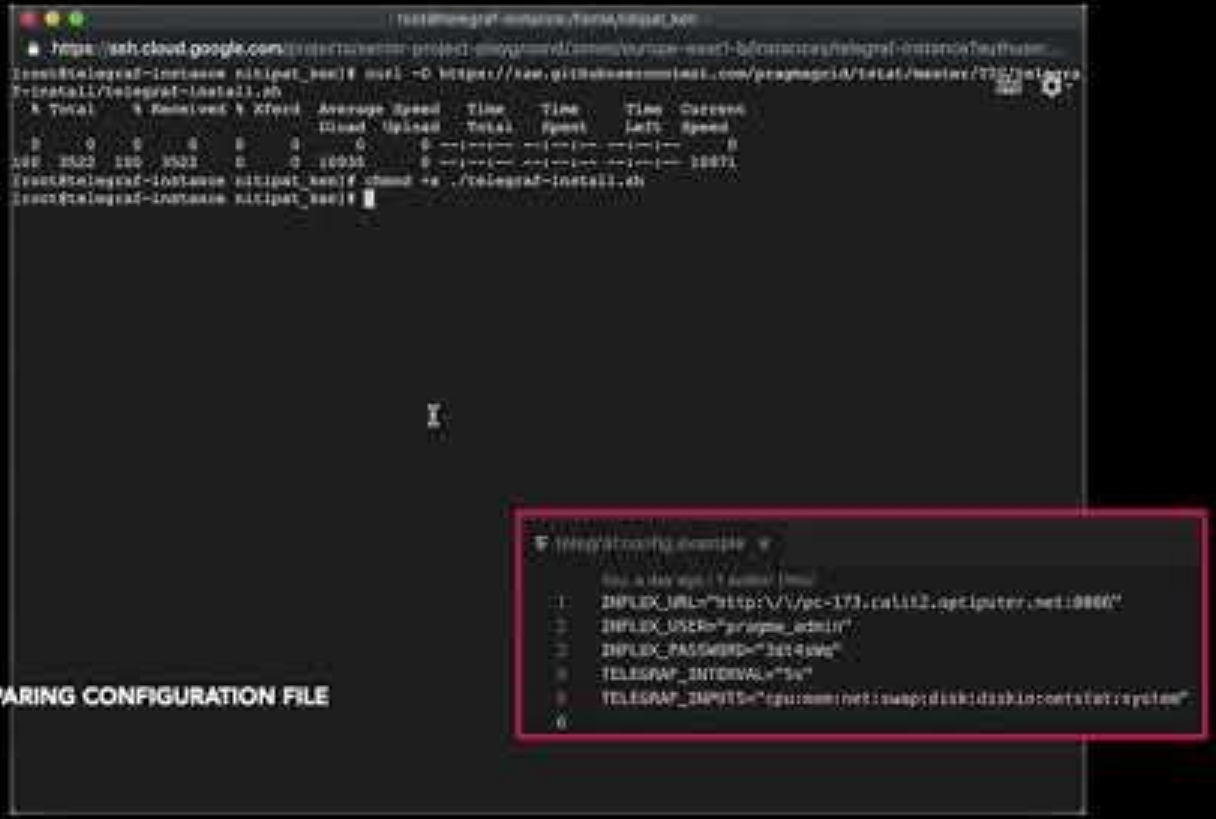
telegraf.config.example

```
You, a day ago | 1 author (You)
1  INFLUX_URL="http:\/\/pc-173.calit2.optiputer.net:8086"
2  INFLUX_USER="pragma_admin"
3  INFLUX_PASSWORD="3dt4sWq"
4  TELEGRAF_INTERVAL="5s"
5  TELEGRAF_INPUTS="cpu:mem:net:swap:disk:diskio:netstat:system"
```

3. Execute the installation with configuration file

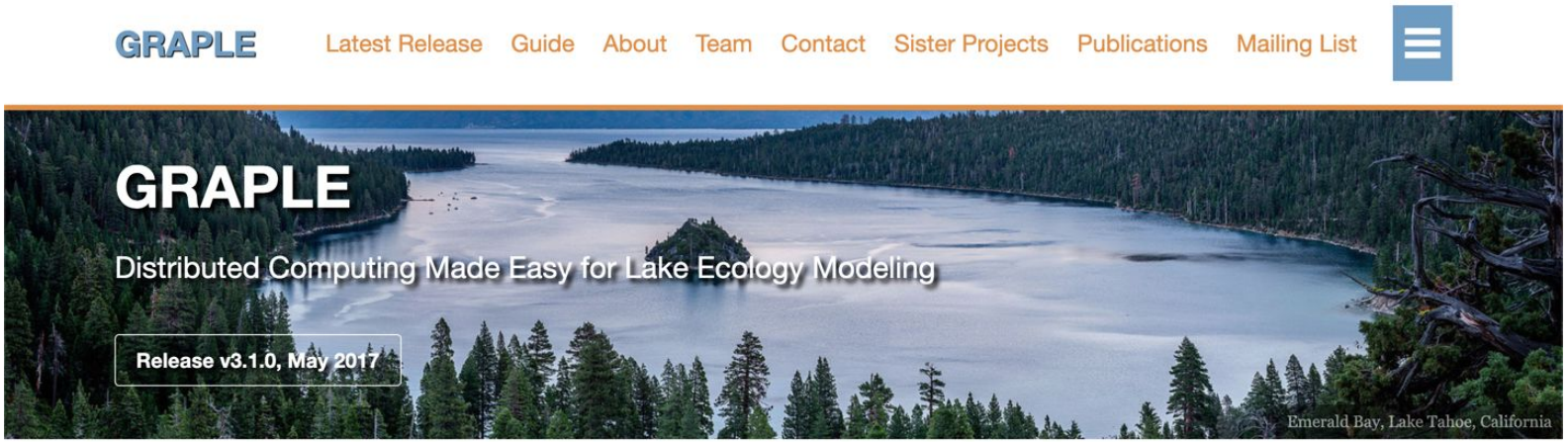
```
telegraf-install.sh telegraf.config.example
```

Telegraf installation demonstration video



Default dashboard
after installed all parts

GRAPLEr



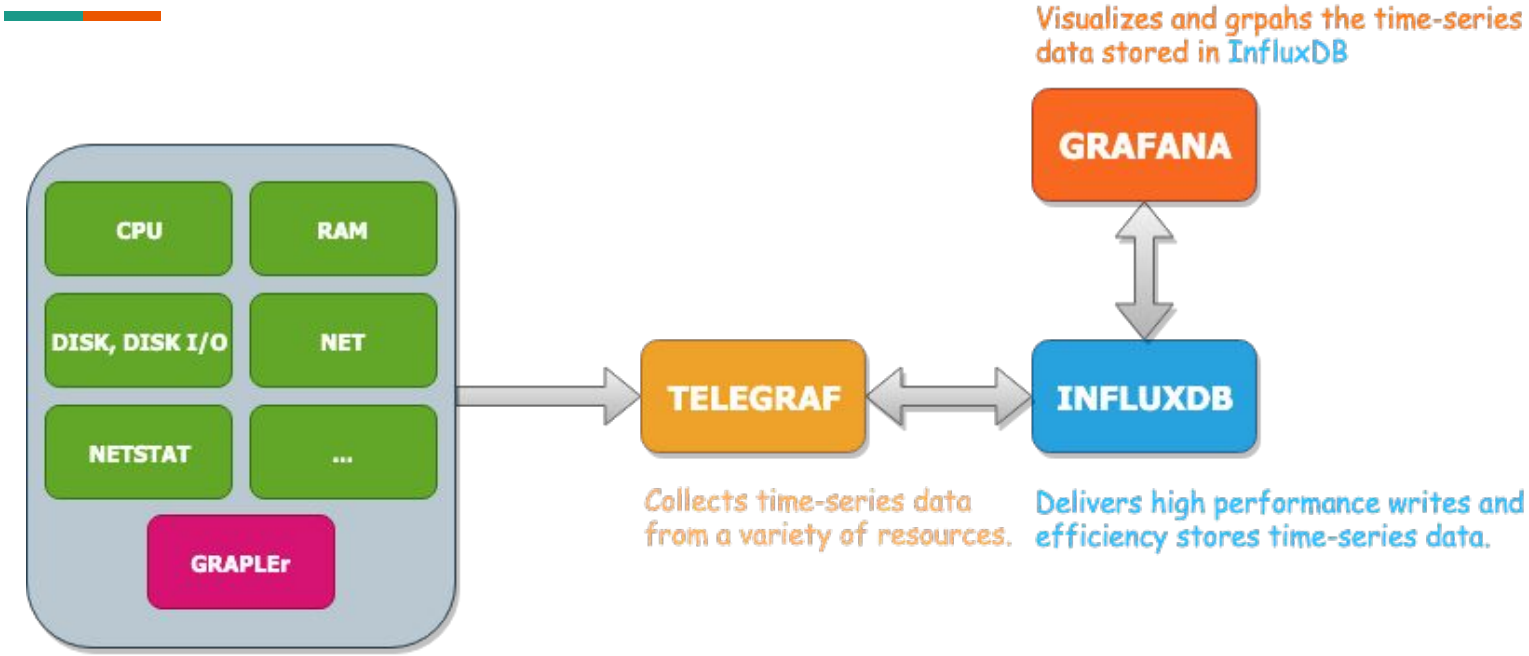
GRAPLE is an inter-disciplinary collaboration between computer scientists and lake modelers with the *GLEON Research and PRAGMA Lake Expedition*.

The *GRAPLE* collaboration's main software product is *GRAPLEr*, an R-based open-source software that brings the power of distributed computing to the fingertips of lake ecology modelers.

Implements new plugin

TELEGRAF

```go
package htcondor

.
.
.

// Gather outputs.
func (htc *HTCondor) Gather(acc telegraf.Accumulator) error {
    c := exec.Command("condor_q")
    out, _ := c.Output()

    var regexGroupMatch = condorOutputRegex.FindAllStringSubmatch(string(out), -1)
    fields := make(map[string]interface{})
    tags := make(map[string]string)

    for i := 1; i < len(regexGroupMatch[0]); i++ {
        var matched = strings.Split(regexGroupMatch[0][i], " ") // "1 jobs" --> ["1", "jobs"]
        var fieldKey = matched[1]
        var fieldvalue, _ = strconv.ParseInt(matched[0], 10, 64)
        fields[fieldKey] = fieldvalue
    }
    acc.AddFields(measurement, fields, tags)

    return nil
}
```

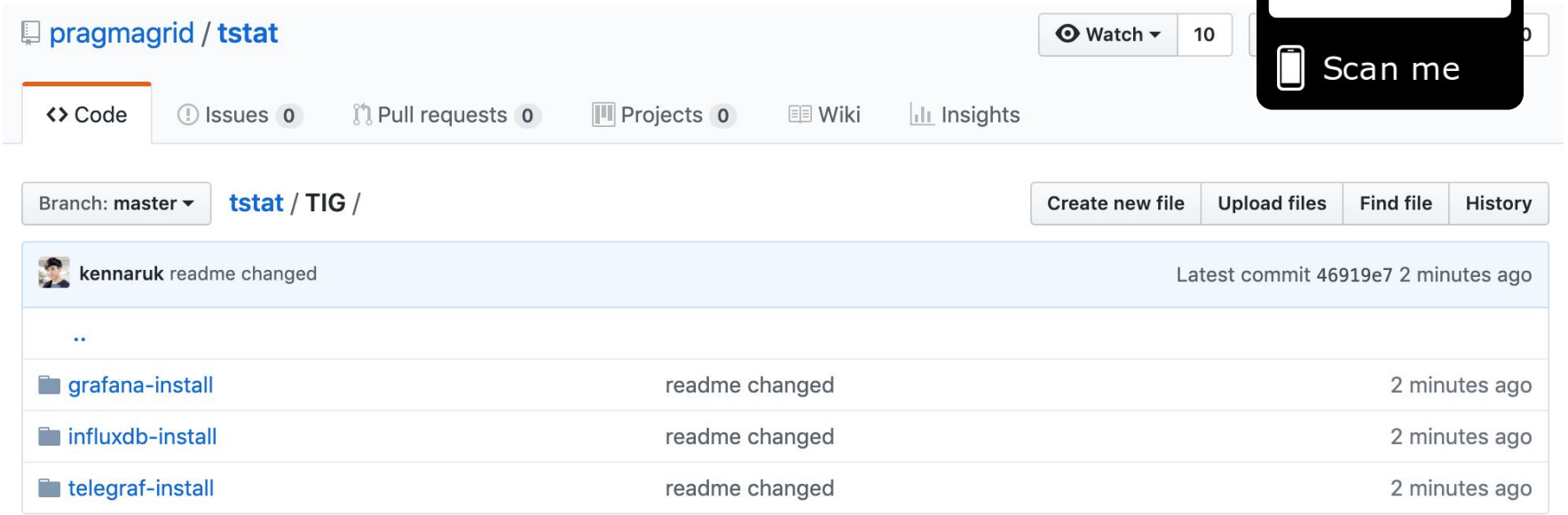


Inspired from https://wiki.zimbra.com/wiki/File:Tig-monitor-logic.png

Demo - Real-time monitoring dashboard

Link to installation repository

You can explore more here https://github.com/pragmagrid/tstat/tree/master/TIG



Thank you!

Any questions?