

Auditoria
de Logs

Sumário

1 Introdução ao sistema de logs	3
1.1 Por que registrar logs?	3
1.1.1 Conceito	3
1.1.2 syslog, syslog-ng, rsyslog	4
1.2 Logs no Linux	4
1.2.1 Conceito	4
1.3 LAB 1.1 - Acesso às máquinas virtuais	6
1.4 LAB 1.2 - Localização padrão dos logs no Linux	7
1.5 LAB 1.3 - Personalizar configuração de logs	13
1.6 Logrotate	14
1.6.1 Conceito	14
1.7 LAB 1.4 - Criar rotação de logs	15
1.8 Resolver problemas através dos logs	17
1.8.1 Conceito	17
1.9 LAB 1.5 - Resolver problemas através de logs	17
2 Auditoria de acessos no sistema	21
2.1 Auditd	21
2.1.1 Conceito	21

2.2 LAB 2.1 - Criação das regras de auditoria	21
2.3 Syscalls	26
2.3.1 Conceito	26
2.4 LAB 2.2 - Criação de regra para execução de syscalls	26
2.5 Analisar registros de auditoria	29
2.5.1 Conceito	29
2.6 LAB 2.3 - Analisar eventos com ausearch e aureport	30
2.7 Pluggable Authentication Modules	34
2.7.1 Conceito	34
2.8 LAB 2.4 - Auditoria em tempo real	35
2.9 audisdp-plugins	36
2.9.1 Conceito	36
2.10 LAB 2.5 - Gerenciar retransmissão de logs	36
2.11 Mais informações:	38
3 Gerenciar logs remotamente	39
3.1 Log remoto	39
3.1.1 Conceito	39
3.2 LAB 3.1 - Gerenciar centralização de logs	40
3.3 Criptografar o envio de logs	43
3.3.1 Conceito	43
3.4 LAB 3.2 - Logs remotos com criptografia TLS	44
3.5 Armazenamento de logs no MySQL	60
3.5.1 Conceito	60
3.6 LAB 3.3 - Armazenar logs no banco de dados MySQL	60

3.6.1	Instalação e configuração do banco no MySQL	61
3.7	Backup de logs no MySQL	65
3.7.1	Conceito	65
3.8	LAB 3.4 - Realizar backup e restore de logs no banco MySQL	66
4	Centralização de logs com Graylog	73
4.1	Instalação e configuração do Graylog (Graylog + Elastic + MongoDB)	74
4.2	Graylog	74
4.2.1	LAB 4.1 - Instalação e configuração do Graylog	75
4.3	Configurar e entender os Inputs	82
4.3.1	LAB 4.2 - Coletar logs dos hosts pelo rsyslog	83
4.3.2	LAB 4.3 - Configuração de clientes	86
4.3.3	LAB 4.4 - Coletar logs de containers	88
4.3.4	Configuração no Graylog	88
4.3.5	Configuração no Docker	90
4.4	Extratores no Graylog	94
4.4.1	LAB 4.5 - Coletar logs de containers	94
4.5	Dashboards para o Graylog	101
4.5.1	LAB 4.6 - Criar um dashboard para estatísticas de acesso	101
4.6	Criação de alertas no Graylog	115
4.6.1	LAB 4.7 - Criar alerta para envio de email	115
4.6.2	LAB 4.8 - Criar alertas para envio de mensagens via chat	126
5	Centralização de logs com ELK	146
5.1	Características do Elastic Stack	147

5.1.1	Componentes e características	147
5.1.2	Elasticsearch	148
5.1.3	Logstash	150
5.1.4	Kibana	151
5.1.5	FileBeat	152
5.2	Instalação e configuração da pilha ELK (Elastic, Logstash e Kibana)	153
5.2.1	LAB 5.1 - Instalação e configuração do Elasticsearch	153
5.2.2	Configuração do Elasticsearch	154
5.2.3	LAB 5.2 - Instalação e configuração do Logstash	156
5.2.4	Instalação do Logstash	156
5.2.5	Configuração do Logstash	156
5.2.6	LAB 5.3 - Instalação e configuração do Kibana	159
5.2.7	Instalação do Kibana e Nginx	159
5.2.8	Configuração do Kibana	159
5.2.9	Configuração do Nginx	160
5.2.10	Acessar o dashboard do Kibana	161
5.3	Utilizar o FileBeat para envio de arquivos de logs	162
5.3.1	Beats	162
5.3.2	Back-Pressure Sensitive Protocol	164
5.3.3	Beats Elastic Co.	165
5.3.4	Community Beats	165
5.3.5	Filebeat	167
5.3.6	LAB 5.4 - Instalação e configuração do Filebeat	167
5.3.7	Configuração do Filebeat em todas as VMs	168

5.3.8 Metricbeat	170
5.3.9 LAB 5.5 - Instalação e configuração do Metricbeat	170
5.3.10 Instalação e configuração o o Metricbeat	170
5.4 Construção de dashboards com Kibana	173
5.4.1 LAB 5.6 - Realizar buscas no Kibana	173
5.4.2 Primeiros passos com Kibana	174
5.4.3 Index Patterns - Filebeat	175
5.4.4 Visualizar logs	177
5.4.5 Index Patterns – Metricbeat	179
5.4.6 Construção de dashboards	181
5.4.7 LAB 5.7 - Construção do dashboard	181
5.5 Mais informações:	185
6 Gerenciar logs na AWS com o Cloudwatch	186
6.1 AWS – Conta gratuita	186
6.1.1 Amazon Web Services	186
6.2 Introdução ao CloudWatch	187
6.3 Criar função CloudWatchFullAccess	188
6.4 Gerenciar instâncias na AWS	192
6.5 Instalar e configurar o Cloudwatch Agent	202
6.6 Visualizar logs da instância no console do Cloudwatch	203
6.7 Configurar e visualizar logs de um servidor web	206
6.8 Visualizar logs do servidor Web	207
6.9 Visualizar logs de container no Docker	211

7 Gerenciar logs na GCP com o Stackdriver	215
7.1 GCP - Conta gratuita	215
7.1.1 Google Cloud Platform	215
7.2 Introdução ao Stackdriver	216
7.3 Gerenciar instâncias na GCP	217
7.4 Instalar e configurar o Cloud Logging	222
7.4.1 Configuração	222
7.5 Visualizar logs da instância da GCP	224
7.6 Configurar e visualizar logs de um servidor web	230
7.7 Visualizar logs de container no Docker	233

1

Introdução ao sistema de logs

Competências deste conteúdo

- Conhecer as diferenças entre syslog, syslog-ng, rsyslog
- Configuração padrão de rsyslog em Ubuntu, Debian e CentOS
- Entender o que são os facilities e os severity levels
- Localização padrão de arquivos de logs
- Gerencia de arquivos de logs, permissões, tamanho com o logrotate
- Verificação de problemas no gerenciador de logs
- Procurar, analisar e resolver problemas através de logs

Por que registrar logs?

Conceito

No item 12.4, que trata sobre registros e monitoramento da norma NBR ISO/IEC 27002:2013 fala-se sobre detectar e registrar atividades não autorizadas. Assim, é prioridade adotar uma política de segurança.

Além de realizar auditorias de acesso e atividades anormais nos servidores, podemos obter a saúde dos serviços e a identificação de problemas no servidor, sendo de extrema importância a sua implantação.

syslog, syslog-ng, rsyslog

Atualmente, o Linux possui três diferentes serviços para a administração de logs, os quais seguem a mesma linha. São eles:

syslog (o mais antigo), **syslog-ng** e **rsyslog**.

Os serviços têm a característica de separar o servidor de logs em:

- **Facilidade:** utilizado pra identificar o serviço para o qual o log está sendo enviado;
- **Nível:** identifica a criticidade de um log gerado;
- **Arquivo:** o nome do arquivo que será registrado no log em questão.

Comparação entre syslog, syslog-ng e rsyslog:

	syslog	syslog-ng	rsyslog
Data de criação	1997	1998	2004
TCP/UDP Remote Log	No	Yes	Yes
TLS Encrypted	No	Yes	Yes
DB SQL Storage	No	Yes	Yes

Comparando o rsyslog e syslog-ng, podemos observar que eles possuem recursos semelhantes. Por outro lado, o **syslog-ng** possui uma versão Enterprise que garante melhorias como alta disponibilidade e alertas.

Logs no Linux

Conceito

Para registrar logs no Linux é importante observar três fatores: o tipo de log (facilidade), o nível do registro do log e o local onde serão armazenados.

Exemplo de um registro de log para armazenar no arquivo **/var/log/mail.err**, todos os registros de erros que envolvam emails:

```
1 |mail.err          /var/log/mail.err
```

Facilidades

Utilizamos a facilidade para determinar o tipo de objeto que vamos registrar no sistema. Pode ser um erro no Kernel, uma informação de email ou por exemplo, um aviso sobre uma autenticação no sistema.

Abaixo podemos observar as facilidades que o Syslog utiliza:

auth	local[0-7]	user
authpriv	lpr	kern
cron	mail	syslog
ftp	news	daemon

Níveis de criticidade

Os níveis de criticidade maiores como Emerg, Alert e Critical indicam realmente algum erro ou problema no serviço. Já as criticidades menores não necessariamente são problemas, algumas delas são avisos para o funcionamento de uma verificação ou uma notificação relevante a ser cadastrada, como um envio de email com sucesso.

Abaixo podemos observar os níveis de criticidade que Syslog utiliza:

Emerg	Alert
Crit	Err
Warn	Notice
Info	Debug

Padrões

O sistema de logs syslog possui alguns arquivos padrões com funcionalidades específicas:

- **lastlog**: contém informações dos últimos logins de usuários;
- **dmesg**: contém mensagens de reconhecimento de hardware pelo kernel;
- **messages**: é o principal log do sistema, possui mensagens enviadas por aplicações e serviços;
- **syslog**: arquivo padrão do sistema para qualquer evento que não possua um arquivo específico, contendo uma diversidade.

LAB 1.1 - Acesso às máquinas virtuais

Neste laboratório vamos aprender como acessar as máquinas virtuais (VMs) que utilizaremos. É necessário seguir os procedimentos da aula **Preparando o ambiente**, que está disponível neste PDF ou através de vídeo aula no Netclass da 4Linux .

Para começar acesse a pasta 4516 que você obteve do Github da 4Linux em <https://github.com/4linux>:

```
1 | cd 4516
```

Em seguida verifique se as VMs que utilizaremos estão presentes, através do comando **vagrant status**:

```
1 | vagrant status
```

- Resultado do comando:

```
1 | Current machine states:  
2 |  
3 | webserver-audit          poweroff (virtualbox)  
4 | graylog-audit            poweroff (virtualbox)  
5 | kibana-audit              poweroff (virtualbox)  
6 |  
7 |  
8 | This environment represents multiple VMs. The VMs are all  
9 | listed  
10 | above with their current state. For more information about a  
   | specific  
10 | VM, run `vagrant status NAME`.
```

Para criar as VMs utilize o subcomando **up** do Vagrant:

```
1 | vagrant up
```

Para acessar as VMs utilize o comando **ssh suporte@<ip_da_vm>**. Exemplos:

Máquina	IP	Senha
webserver-audit	172.16.0.11	4linux
graylog-audit	172.16.0.12	4linux
kibana-audit	172.16.0.13	4linux

LAB 1.2 - Localização padrão dos logs no Linux

Neste laboratório vamos explorar os arquivos de log padrão, em distribuições baseadas no **Debian** e **RedHat**.

Localização de logs

Distribuição Debian/Ubuntu Acesse as VMs **kibana-audit** e **graylog-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.13
```

```
1 | ssh suporte@172.16.0.12
```

Na VM **kibana-audit** que usa a distribuição **Debian**, liste os arquivos de log padrão no diretório **/var/log**:

```
1 | ls -l /var/log/
2 | total 560
3 | -rw-r--r-- 1 root root      314 Aug 12 21:12 alternatives.log
4 | drwxr-xr-x 2 root root    4096 Aug 12 21:12 apt
5 | -rw-r----- 1 root adm    27113 Aug 13 18:59 auth.log
6 | -rw----- 1 root utmp      0 Nov 14 2017 btmp
7 | -rw-r----- 1 root adm   45912 Aug 13 19:04 daemon.log
8 | -rw-r----- 1 root adm   24556 Aug 13 16:55 debug
9 | -rw-r--r-- 1 root root    7214 Aug 12 21:12 dpkg.log
10 | -rw-r--r-- 1 root root   32064 Nov 14 2017 faillog
11 | drwxr-xr-x 3 root root   4096 Nov 14 2017 installer
12 | -rw-r----- 1 root adm   88051 Aug 13 16:55 kern.log
13 | -rw-rw-r-- 1 root utmp 292584 Aug 13 18:51 lastlog
14 | -rw-r----- 1 root adm 116624 Aug 13 17:00 messages
15 | -rw-r----- 1 root adm    2764 Aug 13 19:04 syslog
16 | -rw-r----- 1 root adm 208763 Aug 13 17:00 syslog.1
17 | -rw-r----- 1 root adm   3865 Aug 13 16:55 user.log
18 | -rw-rw-r-- 1 root utmp   6528 Aug 13 18:51 wtmp
```

Na VM **graylog-audit** que usa a distribuição **Ubuntu**, liste os arquivos de log padrão no diretório **/var/log**:

```
1 | ls -l /var/log
2 | total 1068
```

```

3 |-rw-r--r-- 1 root    root   21119 Aug 12 21:10 alternatives.log
4 |drwxr-xr-x 2 root    root    4096 Nov 26 2017 apt
5 |-rw-r----- 1 syslog  adm   30637 Aug 13 19:00 auth.log
6 |-rw-r--r-- 1 root    root   57457 Aug  1 2017 bootstrap.log
7 |-rw----- 1 root    utmp     0 Aug  1 2017 btmp
8 |-rw-r----- 1 root    adm    31 Aug  1 2017 dmesg
9 |-rw-r--r-- 1 root    root 359971 Aug 12 21:10 dpkg.log
10|-rw-r--r-- 1 root    root  32064 Nov 26 2017 faillog
11|drwxr-xr-x 2 root    root    4096 Nov 26 2017 fsck
12|drwxr-xr-x 3 root    root    4096 Nov 26 2017 installer
13|-rw-r----- 1 syslog  adm 132190 Aug 13 16:55 kern.log
14|-rw-rw-r-- 1 root    utmp 292584 Aug 13 16:56 lastlog
15|-rw-r----- 1 syslog  adm 176031 Aug 13 18:17 syslog
16|-rw-r--r-- 1 root    root     1 Nov 26 2017 vboxadd-install.log
17|-rw-r--r-- 1 root    root 218321 Nov 26 2017 vboxadd-setup.log
18|-rw-rw-r-- 1 root    utmp  6144 Aug 13 16:56 wtmp

```

Distribuição CentOS Acesse a VM **webserver-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.11
```

Liste os arquivos de log padrão no diretório **/var/log**:

```

1 |ls -l /var/log
2 |total 400
3 |drwxr-xr-x. 2 root    root      20 Nov 14 2017 anaconda
4 |drwx----- 2 root    root      23 Aug 12 21:13 audit
5 |-rw----- 1 root    root   19862 Aug 13 2020 boot.log
6 |-rw----- 1 root    utmp     0 Nov 14 2017 btmp
7 |drwxr-xr-x. 2 chrony chrony      6 Aug  3 2017 chrony
8 |-rw----- 1 root    root   2369 Aug 13 16:01 cron
9 |-rw-r--r-- 1 root    root  29044 Aug 13 2020 dmesg
10|-rw-r--r-- 1 root    root  29254 Aug 12 21:13 dmesg.old
11|-rw-r--r-- 1 root    root   193 Nov 14 2017
     grubby_prune_debug
12|-rw-r--r-- 1 root    root 292584 Aug 13 15:54 lastlog
13|-rw----- 1 root    root    588 Aug 13 13:56 maillog
14|-rw----- 1 root    root 268057 Aug 13 16:01 messages
15|drwxr-xr-x. 2 root    root      6 Aug  4 2017 qemu-ga
16|drwxr-xr-x. 2 root    root      6 Nov 14 2017 rhsm
17|-rw----- 1 root    root 13253 Aug 13 15:59 secure
18|-rw----- 1 root    root     0 Nov 14 2017 spooler
19|-rw----- 1 root    root     0 Nov 14 2017 tallylog

```

```
20 | drwxr-xr-x. 2 root    root      23 Aug 12 18:13 tuned
21 | -rw-rw-r--. 1 root    utmp      6528 Aug 13 15:54 wtmp
22 | -rw-----. 1 root    root      229 Aug 12 18:15 yum.log
```

Conteúdo do arquivos de logs

Na VM **webserver-audit** e visualize as 10 últimas linhas dos seguintes arquivos:

lastlog

```
1 | sudo tail /var/log/lastlog
2 | ?*?_pts/010.0.2.2[vagrant@webserver ~]$
```

dmesg

```
1 | sudo tail /var/log/dmesg
2 | [ 4.954851] [TTM] Zone kernel: Available graphics memory:
  941148 kiB
3 | [ 4.955392] [TTM] Initializing pool allocator
4 | [ 4.955844] [TTM] Initializing DMA pool allocator
5 | [ 4.957769] fbcon: vboxdrmfb (fb0) is primary device
6 | [ 4.983915] Console: switching to colour frame buffer device
  100x37
7 | [ 4.994282] vboxvideo 0000:00:02.0: fb0: vboxdrmfb frame
  buffer device
8 | [ 4.994553] [drm] Initialized vboxvideo 1.0.0 20130823 for
  0000:00:02.0 on minor 0
9 | [ 5.207143] type=1305 audit(1605118055.850:4): audit_pid=2143
  old=0 auid=4294967295 ses=4294967295 subj=system_u:system_r
  :auditd_t:s0 res=1
10 | [ 5.481640] e1000 0000:00:08.0 eth1: (PCI:33MHz:32-bit)
  08:00:27:59:fe:f0
11 | [ 5.481991] e1000 0000:00:08.0 eth1: Intel(R) PRO/1000
  Network Connection
```

messages

```
1 | sudo tail /var/log/messages
2 | Nov 11 18:10:58 localhost kernel: docker_gwbridge: port 4(
  vethb6c8919) entered blocking state
3 | Nov 11 18:10:58 localhost kernel: docker_gwbridge: port 4(
  vethb6c8919) entered forwarding state
4 | Nov 11 18:10:58 localhost firewalld[2251]: WARNING:
  COMMAND_FAILED: '/usr/sbin/iptables -w2 -t nat -C
  POSTROUTING -m ipvs --ipvs -d 10.0.1.0/24 -j SNAT --to-
```

```
source 10.0.1.4' failed: iptables: No chain/target/match by
that name.
5 Nov 11 18:13:06 localhost systemd-logind: Removed session 2.
6 Nov 11 18:13:06 localhost systemd: Removed slice User Slice of
vagrant.
7 Nov 11 18:14:49 localhost dockerd: time="2020-11-11T18
:14:49.403966254Z" level=info msg="NetworkDB stats webserver
(0e3d06e8cc89) - netID:jxtxllswszku3kilxttrcnbwk leaving:
false netPeers:1 entries:4 Queue qLen:0 netMsg/s:0"
8 Nov 11 18:14:49 localhost dockerd: time="2020-11-11T18
:14:49.404465677Z" level=info msg="NetworkDB stats webserver
(0e3d06e8cc89) - netID:hbwai3mvp9pzzt5xrmmmsvdwun leaving:
false netPeers:1 entries:7 Queue qLen:0 netMsg/s:0"
9 Nov 11 18:16:40 localhost systemd: Created slice User Slice of
vagrant.
10 Nov 11 18:16:40 localhost systemd: Started Session 3 of user
vagrant.
11 Nov 11 18:16:40 localhost systemd-logind: New session 3 of user
vagrant.
```

Alterne para a VM **kibana-audit** e visualize as 10 últimas linha do seguinte arquivo syslog:

```
1 sudo tail /var/log/syslog
2 Aug 13 18:51:03 graylog systemd[1077]: Reached target Sockets.
3 Aug 13 18:51:03 graylog systemd[1077]: Reached target Basic
System.
4 Aug 13 18:51:03 graylog systemd[1077]: Reached target Default.
5 Aug 13 18:51:03 graylog systemd[1077]: Startup finished in 5ms.
6 Aug 13 18:51:03 graylog systemd[1]: Started User Manager for UID
1000.
7 Aug 13 19:04:46 graylog systemd[1]: Started Run anacron jobs.
8 Aug 13 19:04:46 graylog anacron[1169]: Anacron 2.3 started on
2020-08-13
9 Aug 13 19:04:46 graylog anacron[1169]: Normal exit (0 jobs run)
10 Aug 13 19:04:46 graylog systemd[1]: anacron.timer: Adding 3min
10.228878s random time.
11 Aug 13 19:17:01 graylog CRON[1174]: (root) CMD (    cd / && run-
parts --report /etc/cron.hourly)
```

Comandos que geram logs

No diretório **/var/log** existem arquivos que não devemos visualizar o seu conteúdo através do comando **cat**. O ideal é executar um comando que irá exibir o seu conteúdo de forma legível. Vamos explorar esses comandos.

dmesg: exibe o conteúdo do arquivo **/var/log/dmesg**.

Alterne para a VM **webserver-audit** e execute o comando **dmesg**:

```
1 | dmesg
2 | ....
3 | [ 206.633565] br0: port 5(veth3) entered blocking state
4 | [ 206.634046] br0: port 5(veth3) entered disabled state
5 | [ 206.634721] device veth3 entered promiscuous mode
6 | [ 206.636556] IPv6: ADDRCONF(NETDEV_UP): veth3: link is not
   ready
7 | [ 206.643761] docker_gwbridge: port 4(vethb6c8919) entered
   blocking state
8 | [ 206.644169] docker_gwbridge: port 4(vethb6c8919) entered
   disabled state
9 | [ 206.644560] device vethb6c8919 entered promiscuous mode
10 | [ 206.645000] IPv6: ADDRCONF(NETDEV_UP): vethb6c8919: link is
    not ready
11 | [ 206.645398] docker_gwbridge: port 4(vethb6c8919) entered
   blocking state
12 | [ 206.645740] docker_gwbridge: port 4(vethb6c8919) entered
   forwarding state
13 | [ 206.771281] IPVS: Creating netns size=2048 id=7
14 | [ 207.001769] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not
   ready
15 | [ 207.003043] docker_gwbridge: port 4(vethb6c8919) entered
   disabled state
16 | [ 207.003536] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes
   ready
17 | [ 207.004040] IPv6: ADDRCONF(NETDEV_CHANGE): veth3: link
   becomes ready
18 | [ 207.004440] br0: port 5(veth3) entered blocking state
19 | [ 207.004850] br0: port 5(veth3) entered forwarding state
20 | [ 207.009690] IPv6: ADDRCONF(NETDEV_CHANGE): vethb6c8919: link
   becomes ready
21 | [ 207.010106] docker_gwbridge: port 4(vethb6c8919) entered
   blocking state
22 | [ 207.010576] docker_gwbridge: port 4(vethb6c8919) entered
   forwarding state
```

lastlog: exibe o conteúdo do arquivo **/var/log/lastlog**. O comando **lastlog** exibe informações referentes aos últimos logins realizados no sistema.

```
1 | sudo lastlog
```

	Username	Port	From	Latest
2	root			**Never logged in**
3	daemon			**Never logged in**
4	bin			**Never logged in**
5	sys			**Never logged in**
6	sync			**Never logged in**
7	games			**Never logged in**
8	man			**Never logged in**
9	lp			**Never logged in**
10	mail			**Never logged in**
11	news			**Never logged in**
12	uucp			**Never logged in**
13	proxy			**Never logged in**
14	www-data			**Never logged in**
15	backup			**Never logged in**
16	list			**Never logged in**
17	irc			**Never logged in**
18	gnats			**Never logged in**
19	nobody			**Never logged in**
20	systemd-timesync			**Never logged in**
21	systemd-network			**Never logged in**
22	systemd-resolve			**Never logged in**
23	systemd-bus-proxy			**Never logged in**
24	syslog			**Never logged in**
25	_apt			**Never logged in**
26	messagebus			**Never logged in**
27	sshd			**Never logged in**
28	vagrant	pts/0	10.0.2.2 +0000 2020	Thu Aug 13 16:56:23
29	vboxadd			**Never logged in**
30	suporte			**Never logged in**

last: exibe o conteúdo do arquivo **/var/log/wtmp**. O comando **last** exibe informações referentes a login e logout de usuários do sistema.

```
1 sudo last
2 vagrant pts/0          10.0.2.2           Wed Nov 11 18:16 still
   logged in
3 reboot  system boot 3.10.0-957.12.1. Wed Nov 11 18:07 - 18:21
   (00:14)
4 reboot  system boot 3.10.0-957.12.1. Wed May  1 13:41 - 13:52
   (00:10)
5
6 wtmp begins Wed May  1 13:41:47 2019
```

lastb: exibe o conteúdo do arquivo **/var/log/btmp**. O comando **lastb** exibe informações sobre as tentativas mal sucedidas de se logar no sistema..

```
1 | sudo lastb  
2 | btmp begins Wed Nov 11 18:07:35 2020
```

LAB 1.3 - Personalizar configuração de logs

Neste laboratório vamos criar uma configuração personalizada, para gerar logs sobre a ferramenta Cron (utilizada para agendar tarefas no sistema).

Acesse a VM **graylog-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.12
```

Crie o arquivo **cron.conf** para gerar logs sobre a ferramenta Cron.

```
1 | sudo vim /etc/rsyslog.d/cron.conf  
2 | cron.\*          /var/log/cron.log
```

Explicando a configuração:

- **Facilidade**: com o primeiro valor estando na configuração anterior, “**cron**”;
- **Criticidade**: o nível de criticidade do log, sendo o ****_*_*_*** qualquer valor de **debug** à **emerg**;
- **Arquivo de destino do log**: podemos utilizar um arquivo como **/var/log/cron.log**, ou um servidor remoto via **udp** ou **tcp** (representados por **@** ou **@@**, respectivamente).

O próximo passo é reiniciar o serviço do **rsyslog**, para aplicar as alterações:

```
1 | sudo systemctl restart rsyslog
```

Antes de visualizar o arquivo de log, gere um conteúdo listando tarefas agendadas no Cron.

```
1 | crontab -l  
2 | no crontab for suporte
```

Para terminar, visualize os logs gerados no arquivo **/var/log/cron.log**:

```
1 | sudo cat /var/log/cron.log
2 | Aug 13 20:17:01 graylog CRON[1534]: (root) CMD (    cd / && run-
|   parts --report /etc/cron.hourly)
3 | Aug 13 20:17:03 graylog crontab[1536]: (suporte) LIST (suporte)
```

Logrotate

Conceito

Com o aumento gradativo dos logs, podemos ter alguns problemas. Um deles é o tamanho de um arquivo de log, aumentando o tempo de procura de logs em um arquivo específico. Outro problema é o espaço em disco utilizado por estes arquivos.

Uma solução simples e padrão do sistema é o **logrotate**, realizando a quebra de logs por tamanho ou tempo, além da possibilidade de compressão do arquivo, diminuindo o espaço em disco utilizado.

Exemplo de uma configuração padrão do Logrotate:

```
1 | # rotate log files weekly
2 | weekly
3 |
4 | # keep 4 weeks worth of backlogs
5 | rotate 4
6 |
7 | # create new (empty) log files after rotating old ones
8 | create
9 |
10 | # uncomment this if you want your log files compressed
11 | #compress
```

Explicando:

- **weekly**: define a rotação de logs semanais.
- **rotate**: mantém os últimos 4 arquivos de logs.
- **create**: cria arquivos **.old** após o **rotate**.
- **compress**: realiza a compressão gz nos arquivos de logs.

LAB 1.4 - Criar rotação de logs

Neste laboratório vamos criar uma configuração personalizada, para rotacionar os logs do arquivo **/var/log/cron.log**.

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do root.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Crie o arquivo **/etc/logrotate.d/cron** o seguinte conteúdo:

```
1 | vim /etc/logrotate.d/cron
2 | /var/log/cron.log {
3 |     daily
4 |     rotate 4
5 |     compress
6 |     delaycompress
7 |     size 1M
8 |     missingok
9 |     notifempty
10 |    create 644 root root
11 | }
```

DICA: Use como base o arquivo **/etc/logrotate.d/dpkg**.

Explicando:

- **daily**: define a rotação de logs diários.
- **rotate**: mantém os últimos 4 arquivos de logs.
- **compress**: realiza a compressão gzip nos arquivos de logs.
- **delaycompress**: comprime o arquivo de log rotacionado apenas no próximo rotacionamento.
- **size 1M**: define para 1M o tamanho que o arquivo de log deve ter para ser rotacionado.
- **missingok**: se o arquivo de log não existir o **logrotate** não gera mensagens de erro.
- **notifempty**: não rotaciona o arquivo de log se o mesmo estiver vazio.
- **create 644 root root**: cria novos arquivo com permissão **644** onde o dono é o **root** e grupo dono também é o **root**.

Para não conflitar com nossa configuração, edite o arquivo **/etc/logrotate.d/rsyslog** e comente a linha referente ao cron:

```
1 | vim /etc/logrotate.d/rsyslog
2 | ....
3 | /var/log/lpr.log
4 | #/var/log/cron.log
5 | /var/log/debug
```

Antes de aplicar a rotação de logs, aumente o tamanho do arquivo **/etc/logrotate.d/cron.log** através dos seguintes comandos:

```
1 | cat /var/log/* >> /var/log/cron.log
```

```
1 | cat /var/log/* >> /var/log/cron.log
```

Verifique se o arquivo **/var/log/cron.log** possui um valor maior que 1M.

```
1 | du -h /var/log/cron.log
2 | 1.1M    /var/log/cron.log
```

DICA: Caso o arquivo possua um valor menor que 1M, repita o comando `cat /var/log/* >> /var/log/cron.log` algumas vezes.

Para aplicar a rotação de logs, execute o seguinte comando:

```
1 | logrotate -f /etc/logrotate.conf
```

Para terminar, verifique se no diretório **/var/log** existem no mínimo 2 arquivo do cron.

```
1 | ls /var/log/cron*
2 |
3 | /var/log/cron.log /var/log/cron.log.1
```

Resolver problemas através dos logs

Conceito

A partir do registro de logs no sistema, podemos identificar quais são os erros em serviços e acessos feitos pelos usuários.

Quando o sistema detecta erros de digitação em arquivos de configurações, o serviço não consegue iniciar.

Quando o sistema não consegue autenticar devido ao nome do usuário e/ou senha que estão errados, o login não é completado.

LAB 1.5 - Resolver problemas através de logs

Neste laboratório vamos criar 2 cenários onde um serviço não irá conseguir iniciar, e um usuário não irá conseguir logar no sistema.

Acesse a VM **graylog-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.12
```

Cenário 1: erro de configuração no serviço

Vamos editar o arquivo de configuração do Rsyslog e adicionar na primeira linha do arquivo, a palavra “curso”.

```
1 | sudo vim /etc/rsyslog.conf
2 | curso
3 | # /etc/rsyslog.conf      Configuration file for rsyslog.
```

Reinicie o serviço do Rsyslog para aplicar a configuração:

```
1 | sudo systemctl restart rsyslog
```

Vamos exibir o estado do serviço e verificar o erro apresentado:

```
1 | sudo systemctl status rsyslog●
```

```
2 | rsyslog.service - System Logging Service
3 |   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled;
4 |     vendor preset: enabled)
5 |   Active: active (running) since Thu 2020-08-13 23:18:24 UTC;
6 |     48s ago
7 |     Docs: man:rsyslogd(8)
8 |           http://www.rsyslog.com/doc/
9 |   Main PID: 3166 (rsyslogd)
10 |    Tasks: 4 (limit: 4915)
11 |   CGroup: /system.slice/rsyslog.service
12 |           3166 /usr/sbin/rsyslogd -n
13 |
14 | Aug 13 23:18:24 graylog systemd[1]: Starting System Logging
15 |   Service...
16 | Aug 13 23:18:24 graylog liblogging-stdlog[3166]: [origin
17 |     software="rsyslogd" swVersion="8.24.0" x-pid="3166" x-info="
18 |       htt
19 | Aug 13 23:18:24 graylog systemd[1]: Started System Logging
20 |   Service.
21 | Aug 13 23:18:24 graylog liblogging-stdlog[3166]: action 'curso'
22 |     treated as ':omusrmmsg:curso' - please use ':omusrmmsg:curs
23 | Aug 13 23:18:24 graylog liblogging-stdlog[3166]: error during
24 |     parsing file /etc/rsyslog.conf, on or before line 1: warnin
```

Podemos identificar o erro através do arquivo **/var/log/syslog**:

```
1 | sudo tail /var/log/syslog
2 | Aug 13 23:04:30 graylog liblogging-stdlog: error during parsing
3 |     file /etc/rsyslog.conf, on or before line 1: warnings
4 |     occurred in file '/etc/rsyslog.conf' around line 1 [v8.24.0
5 |     try http://www.rsyslog.com/e/2207 ]
6 | Aug 13 23:17:01 graylog CRON[3127]: (root) CMD (    cd / && run-
7 |     parts --report /etc/cron.hourly)
8 | Aug 13 23:18:24 graylog systemd[1]: Stopping System Logging
9 |   Service...
10 | Aug 13 23:18:24 graylog liblogging-stdlog: [origin software="
11 |     rsyslogd" swVersion="8.24.0" x-pid="3099" x-info="http://www
12 |     .rsyslog.com"] exiting on signal 15.
13 | Aug 13 23:18:24 graylog systemd[1]: Stopped System Logging
14 |   Service.
15 | Aug 13 23:18:24 graylog systemd[1]: Starting System Logging
16 |   Service...
17 | Aug 13 23:18:24 graylog liblogging-stdlog: [origin software="
18 |     rsyslogd" swVersion="8.24.0" x-pid="3166" x-info="http://www
19 |     .rsyslog.com"] start
```

```
9 | Aug 13 23:18:24 graylog systemd[1]: Started System Logging
  | Service.
10 | Aug 13 23:18:24 graylog liblogging-stdlog: action 'curso'
   | treated as ':omusrmmsg:curso' - please use ':omusrmmsg:curso'
   | syntax instead, 'curso' will not be supported in the future
   | [v8.24.0 try http://www.rsyslog.com/e/2184 ]
11 | Aug 13 23:18:24 graylog liblogging-stdlog: error during parsing
   | file /etc/rsyslog.conf, on or before line 1: warnings
   | occurred in file '/etc/rsyslog.conf' around line 1 [v8.24.0
   | try http://www.rsyslog.com/e/2207 ]
```

Como solução, edite o arquivo de configuração do Rsyslog e remova a palavra “curso”.

```
1 | sudo vim /etc/rsyslog.conf
2 |
3 | # /etc/rsyslog.conf      Configuration file for rsyslog.
```

Não esqueça de reiniciar o serviço para aplicar as configurações

```
1 | sudo systemctl restart rsyslog
```

Vamos exibir o estado do serviço e verificar se o mesmo voltou a funcionar:

```
1 | sudo systemctl status rsyslog●
2 |   rsyslog.service - System Logging Service
3 |     Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled;
4 |       vendor preset: enabled)
5 |     Active: active (running) since Thu 2020-08-13 23:23:31 UTC; 3
6 |           s ago
7 |       Docs: man:rsyslogd(8)
8 |             http://www.rsyslog.com/doc/
9 |     Main PID: 3182 (rsyslogd)
10 |        Tasks: 4 (limit: 4915)
11 |        CGroup: /system.slice/rsyslog.service
12 |                 3182 /usr/sbin/rsyslogd -n
13 |
14 | Aug 13 23:23:31 graylog systemd[1]: Starting System Logging
  | Service...
15 | Aug 13 23:23:31 graylog liblogging-stdlog[3182]: [origin
  | software="rsyslogd" swVersion="8.24.0" x-pid="3182" x-info="
  | htt
16 | Aug 13 23:23:31 graylog systemd[1]: Started System Logging
  | Service.
```

Cenário 2: erro de autenticação no sistema

Acesse a VM **kibana-audit** com o usuário **suporte**.

```
1 | ssh suporte@172.16.0.13
```

Em seguida tente acessar a VM **webserver-audit** via SSH através do usuário **suporte** e senha **123456**.

```
1 | sshpass -p "123456" ssh -l suporte webserver
```

Alterne para a VM **webserver-audit** e identifique o erro através do arquivo **/var/log/auth.log**:

```
1 | sudo grep Failed /var/log/secure --color
2 | Nov 11 18:39:27 localhost sudo: suporte : TTY=pts/0 ; PWD=/home/
    suporte ; USER=root ; COMMAND=/bin/grep Failed /var/log/
    secure --color
```

2

Auditoria de acessos no sistema

Competências deste conteúdo

- Instalação e configuração do **auditd**
- Criação de regras de auditoria para acesso e alteração de arquivos
- Criação de regras para execução de syscalls
- Visualização e busca de eventos com o **ausearch** e **aureport**
- Auditando o que foi digitado em um terminal com **pam_tty_audit**
- Encaminhando eventos para o **syslog** através do **audispd-plugins**

Auditd

Conceito

Um serviço para gerar logs de sistema muito interessante é o **auditd**, padrão no CentOS 7, o qual possui a capacidade de gerar logs personalizados através de regras, assim como a configuração de acesso a arquivos importantes do sistema, como **/etc/passwd** e outros.

LAB 2.1 - Criação das regras de auditoria

Neste laboratório vamos aprender como criar regras de auditoria para acesso e alteração de arquivos.

Acesse a VM **graylog-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.12
```

Antes de gerenciar as regras de auditoria, precisamos instalar a ferramenta **auditd**:

```
1 | sudo apt install auditd -y
```

Comando **auditctl**

Para começar, visualize as opções de utilização do comando **auditctl**:

```
1 | sudo auditctl --help
2 | usage: auditctl [options]
3 |   -a <l,a>           Append rule to end of <l>ist with <a>
4 |             ction
5 |   -A <l,a>           Add rule at beginning of <l>ist with <a>
6 |             ction
7 |   -b <backlog>       Set max number of outstanding audit
8 |             buffers
9 |
10 |   -c                  Continue through errors in rules
11 |   -C f=f              Compare collected fields if available:
12 |             Field name, operator(=,!), field name
13 |   -d <l,a>           Delete rule from <l>ist with <a>ction
14 |             l=task,exit,user,exclude
15 |             a=never,always
16 |   -D                  Delete all rules and watches
17 |   -e [0..2]            Set enabled flag
18 |   -f [0..2]            Set failure flag
19 |             0=silent 1=printk 2=panic
20 |   -F f=v              Build rule: field name, operator
21 |             (=, !=, <, >, <=,
22 |                           >=, &, &=) value
23 |   -h                  Help
24 |   -i                  Ignore errors when reading rules from
25 |             file
26 |   -k <key>            Set filter key on audit rule
27 |   -l                  List rules
28 |   -m text              Send a user-space message
29 |   -p [r|w|x|a]         Set permissions filter on watch
30 |             r=read, w=write, x=execute, a=attribute
31 |   -q <mount,subtree>  make subtree part of mount point's dir
```

watches	
27	-r <rate> Set limit in messages/sec (0=none)
28	-R <file> read rules from file
29	-s Report status
30	-S syscall Build rule: syscall name or number
31	-t Trim directory watches
32	-v Version
33	-w <path> Insert watch at <path>
34	-W <path> Remove watch at <path>
35	--loginuid-immutable Make loginuids unchangeable once set
36	--backlog_wait_time Set the kernel backlog_wait_time

Para verificar os status do ambiente, utilize o parâmetro **-s**:

```

1 | sudo auditctl -s
2 | enabled 1
3 | failure 1
4 | pid 1008
5 | rate_limit 0
6 | backlog_limit 8192
7 | lost 0
8 | backlog 0
9 | backlog_wait_time 0
10 | loginuid_immutable 0 unlocked

```

Para listar regras de auditoria, utilize o parâmetro **-l**:

```

1 | sudo auditctl -l
2 | No rules

```

Criando regras

Para criar uma regra de auditoria, utilize os parâmetros **-w** e **-p**:

```

1 | sudo auditctl -w /etc/passwd -p rwx

```

No exemplo apresentado, criamos uma regra para auditar qualquer tipo de acesso ao arquivo **/etc/passwd**.

Verifique se a regra foi criada.

```
1 | sudo auditctl -l
2 | -w /etc/passwd -p rwx
```

Descrição das opções utilizadas

- **-w /etc/passwd**: inicia um observador (watcher) no arquivo **passwd**. Quando o arquivo for acessado, o watcher irá gerar eventos.
- **-p rwx**: define o tipo de permissão a ser observado. A opção **rwx** acrescenta leitura(r), gravação(w), execução(x) e alteração de atributos(a).

Testando regras

Para verificar se o Auditd esta monitorando acessos ao arquivo **/etc/passwd**, vamos visualizá-lo através do comando **cat**.

```
1 | cat /etc/passwd
2 | root:x:0:0:root:/root:/bin/bash
3 | daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
4 | bin:x:2:2:bin:/bin:/usr/sbin/nologin
5 | sys:x:3:3:sys:/dev:/usr/sbin/nologin
6 | sync:x:4:65534:sync:/bin:/bin/sync
7 | games:x:5:60:games:/usr/games:/usr/sbin/nologin
8 | man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
9 | lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
10 | mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
11 | news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
12 | uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
13 | proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
14 | www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
15 | backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
16 | list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
17 | irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
18 | gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats
   | :/usr/sbin/nologin
19 | nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
20 | systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/
   | systemd:/bin/false
21 | systemd-network:x:101:103:systemd Network Management,,,:/run/
   | systemd/netif:/bin/false
22 | systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/
   | resolve:/bin/false
23 | systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
24 | _apt:x:104:65534::/nonexistent:/bin/false
25 | avahi-autoipd:x:105:109:Avahi autoip daemon,,,:/var/lib/avahi-
```

```
1 autoipd:/bin/false
26 messagebus:x:106:110::/var/run/dbus:/bin/false
27 sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
28 statd:x:108:65534::/var/lib/nfs:/bin/false
29 vagrant:x:1000:1000:vagrant,,,:/home/vagrant:/bin/bash
30 vboxadd:x:999:1::/var/run/vboxadd:/bin/false
31 suporte:x:1001:1001::/home/suporte:/bin/bash
```

Pesquise a string **passwd** no arquivo de log do Auditd, para confirmar se o acesso foi registrado.

```
1 sudo grep -i passwd /var/log/audit/audit.log
2 type=PATH msg=audit(1598037882.938:77): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
3 type=PATH msg=audit(1598037883.818:78): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
4 type=PATH msg=audit(1598037884.458:79): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
5 type=PATH msg=audit(1598037898.849:80): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
6 type=PATH msg=audit(1598037906.557:81): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
7 type=PATH msg=audit(1598037907.542:82): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
8 type=PATH msg=audit(1598037909.047:83): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
9 type=PATH msg=audit(1598037910.471:84): item=0 name="/etc/passwd"
   " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
10 type=PATH msg=audit(1598037912.848:85): item=0 name="/etc/passwd"
    " inode=131847 dev=fe:00 mode=0100644 ouid=0 ogid=0 rdev
   =00:00 nametype=NORMAL
```

Removendo regras

Para remover uma regra de auditoria, utilize o parâmetro **-W**:

```
1 | sudo auditctl -W /etc/passwd
```

Verifique se a regra foi removida.

```
1 | sudo auditctl -l  
2 | No rules
```

Syscalls

Conceito

O Syscall (chamada do sistema) é a interface entre uma aplicação e o kernel do Linux. As chamadas de sistema não são invocadas diretamente, mas sim via funções wrapper na **glibc**. O Auditd permite gerar eventos relacionados a chamadas de sistema, através do comando **auditctl**.

LAB 2.2 - Criação de regra para execução de syscalls

Neste laboratório vamos aprender como criar regras de auditoria para chamada do sistema.

Acesse a VM **graylog-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.12
```

Crie uma regra para auditar mudanças no horário do sistema:

```
1 | sudo auditctl -a exit,always -F arch=b64 -S clock_settime -k  
mudarhora
```

Verifique se a regra foi criada.

```
1 | sudo auditctl -l  
2 | -a always,exit -F arch=b64 -S clock_settime -F key=mudarhora
```

Descrição das opções utilizadas

- **-a always,exit**: permite adicionar uma regra. A opção **always** marca o tempo e grava o registro. A opção **exit** permite criar um evento, quando a chamada do sistema for encerrada.

- **-F arch=b64**: define um filtro para chamadas de arquitetura 64 bits.
- **-S clock_settime**: define uma chamada do sistema para registrar eventos, relacionados ao relogio do sistema. Para ter acesso a lista completas de todas as chamadas de sistema, acesse a man page do syscalls [1].
- **-F key=mudarhora**: define uma string para identificar a regra.

Para verificar se o Auditd esta monitorando alterações na hora do sistema, vamos utilizar o comando **date**.

```
1 | sudo date --set "2020/01/01 20:00"
2 | Wed Jan  1 20:00:00 UTC 2020
```

Pesquise a string **mudarhora** no arquivo de log do Auditd, para confirmar se a data e/ou hora foi alterado.

```
1 | sudo grep -i mudarhora /var/log/audit/audit.log
2 | type=CONFIG_CHANGE msg=audit(1598049164.590:119): auid=1000 ses
   =3 op="add_rule" key="mudarhora" list=4 res=1
3 | type=SYSCALL msg=audit(1598049800.506:120): arch=c000003e
   syscall=227 success=yes exit=0 a0=0 a1=7ffdc4453470 a2=0 a3
   =57d items=0 ppid=1281 pid=1372 auid=1000 uid=0 gid=0 euid=0
   suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty pts0 ses=3 comm=
   date" exe="/bin/date" key="mudarhora"
```

É possível identificar que o syscall registrado foi o de numero **227**. Para exibir a lista dos syscalls e o código correspondente, utilize o comando **ausyscall**.

```
1 | sudo ausyscall --dump
2 | Using x86_64 syscall table:
3 | ....
4 | 220 semtimedop
5 | 221 fadvise64
6 | 222 timer_create
7 | 223 timer_settime
8 | 224 timer_gettime
9 | 225 timer_getoverrun
10 | 226 timer_delete
11 | 227 clock_settime
12 | 228 clock_gettime
13 | 229 clock_getres
14 | 230 clock_nanosleep
```

Para remover uma regra de auditoria syscalls, utilize o parâmetro **-d**:

```
1 | sudo auditctl -d exit,always -F arch=b64 -S clock_settime -k  
    mudarhora
```

Verifique se a regra foi removida.

```
1 | sudo auditctl -l  
2 | No rules
```

Definir regras na inicialização do sistema

Todas as regras criadas no **auditd** são apagadas na reinicialização do serviço. Para manter as configurações na inicialização, é preciso adicionar as configurações no arquivo **/etc/audit/rules.d/audit.rules**.

Para começar crie as regras de auditoria para o arquivo **/etc/passwd**, e também para o syscalls de alteração de data/hora do sistema.

```
1 | sudo auditctl -w /etc/passwd -p rwxa -k listadeusuários
```

```
1 | sudo auditctl -a exit,always -F arch=b64 -S clock_settime -k  
    mudarhora
```

Em seguida logue com o usuário *root*, e envie as regras criadas para o final do arquivo **/etc/audit/rules.d/audit.rules**.

```
1 | sudo su -  
2 | auditctl -l >> /etc/audit/rules.d/audit.rules
```

Verifique se as regras estão armazenadas no arquivo **audit.rules**:

```
1 | cat /etc/audit/rules.d/audit.rules  
2 | ## First rule - delete all  
3 | -D  
4 |  
5 | ## Increase the buffers to survive stress events.  
6 | ## Make this bigger for busy systems  
7 | -b 8192  
8 |  
9 | ## This determine how long to wait in burst of events
```

```
10 | --backlog_wait_time 0
11 |
12 | ## Set failure mode to syslog
13 | -f 1
14 |
15 | -w /etc/passwd -p rwx -k listadeusuarios
16 | -a always,exit -F arch=b64 -S clock_settime -F key=mudarhora
```

Limpe e liste todas as regras antes de testar nossa implementação:

```
1 | auditctl -D
2 | No rules
```

```
1 | auditctl -l
2 | No rules
```

Reinic peace o serviço do **auditd** para aplicas as alterações:

```
1 | systemctl restart auditd
```

Liste as regras para confirmar as alterações:

```
1 | auditctl -l
2 | -w /etc/passwd -p rwx -k listadeusuarios
3 | -a always,exit -F arch=b64 -S clock_settime -F key=mudarhora
```

Analisar registros de auditoria

Conceito

No Linux podemos contar com comandos como **grep**, **egrep**, **awk**, **cut**, entre outros para filtrar e analisar logs de auditoria. O Auditd permite analisar logs de auditoria, através dos comandos **ausearch** e **aureport**.

LAB 2.3 - Analisar eventos com ausearch e aureport

Neste laboratório vamos aprender como visualizar e busca de eventos com as ferramentas **ausearch** e **aureport**.

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário root.

```
1 | ssh suporte@172.16.0.12  
2 | sudo su -
```

Antes de começar a utilizar as ferramentas, crie as regras de auditoria para o arquivo **/etc/passwd**, e também para o syscalls de alteração de data/hora do sistema.

```
1 | auditctl -w /etc/passwd -p rwxa -k listadeusuarios  
  
1 | auditctl -a exit,always -F arch=b64 -S clock_settime -k  
    mudarhora
```

Caso as regras já existam, o sistema irá exibir a seguinte mensagem:

Error sending add rule data request (Rule exists)

Em seguida visualize o conteúdo do arquivo **/etc/passwd** e altere a data e hora para o atual.

```
1 | cat /etc/passwd  
  
1 | date --set "2020/08/21 20:00"  
2 | Fri Aug 21 20:00:00 UTC 2020
```

Comando ausearch

Filtre eventos de visualização do arquivo **/etc/passwd**, através da chave **listadeusuarios**.

```
1 | ausearch -k listadeusuarios  
2 | ----  
3 | time->Fri Aug 21 22:58:11 2020  
4 | type=CONFIG_CHANGE msg=audit(1598050691.079:122): auid=1000 ses  
    =3 op="add_rule" key="listadeusuarios" list=4 res=1
```

```

5  -----
6 time->Fri Aug 21 22:59:49 2020
7 type=PROCTITLE msg=audit(1598050789.383:124): proctitle
   =636174002F6574632F706173737764
8 type=PATH msg=audit(1598050789.383:124): item=0 name="/etc/
   passwd" inode=131847 dev=fe:00 mode=0100644 uid=0 ogid=0
   rdev=00:00 nametype=NORMAL
9 type=CWD msg=audit(1598050789.383:124): cwd="/root"
10 type=SYSCALL msg=audit(1598050789.383:124): arch=c000003e
    syscall=2 success=yes exit=3 a0=7ffd71f61f19 a1=0 a2=
      ffffffff0400 a3=69f items=1 ppid=1281 pid=1388 auid=1000
      uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty
      =pts0 ses=3 comm="cat" exe="/bin/cat" key="listadeusuarios"
```

Filtre eventos de visualização do arquivo **/etc/passwd**, através da chave **mudarhora**.

```

1 ausearch -k mudarhora
2 -----
3 time->Fri Aug 21 22:32:44 2020
4 type=CONFIG_CHANGE msg=audit(1598049164.590:119): auid=1000 ses
   =3 op="add_rule" key="mudarhora" list=4 res=1
5 -----
6 time->Fri Aug 21 22:43:20 2020
7 type=PROCTITLE msg=audit(1598049800.506:120): proctitle
   =64617465002D2D73657400323032302F30312F30312032303A3030
8 type=SYSCALL msg=audit(1598049800.506:120): arch=c000003e
   syscall=227 success=yes exit=0 a0=0 a1=7ffdc4453470 a2=0 a3
   =57d items=0 ppid=1281 pid=1372 auid=1000 uid=0 gid=0 euid=0
   suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=3 comm="
   date" exe="/bin/date" key="mudarhora"
9 -----
10 time->Fri Aug 21 22:50:11 2020
11 type=CONFIG_CHANGE msg=audit(1598050211.535:121): auid=1000 ses
   =3 op="remove_rule" key="mudarhora" list=4 res=1
12 -----
13 time->Fri Aug 21 22:58:14 2020
14 type=CONFIG_CHANGE msg=audit(1598050694.943:123): auid=1000 ses
   =3 op="add_rule" key="mudarhora" list=4 res=1
15 -----
16 time->Fri Aug 21 23:00:23 2020
17 type=PROCTITLE msg=audit(1598050823.183:125): proctitle
   =64617465002D2D73657400323032302F30382F32312032303A3030
18 type=SYSCALL msg=audit(1598050823.183:125): arch=c000003e
   syscall=227 success=yes exit=0 a0=0 a1=7ffcd77316a0 a2=0 a3
   =57d items=0 ppid=1281 pid=1390 auid=1000 uid=0 gid=0 euid=0
   suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=3 comm="
```

```
| date" exe="/bin/date" key="mudarhora"
```

Execução de comandos

Para visualizar eventos sobre execução de comandos, execute o comando **ausearch** com o parâmetro **-x**. Antes não esqueça de executar algum comando no sistema.

```
1 | crontab -l
2 | no crontab for root
```

```
1 | ausearch -x /usr/bin/crontab
2 |
3 | time->Fri Aug 21 23:34:52 2020
4 | type=PROCTITLE msg=audit(1598052892.473:1153187): proctitle
   =63726F6E746162002D6C
5 | type=PATH msg=audit(1598052892.473:1153187): item=0 name="/etc/
   passwd" inode=131841 dev=fe:00 mode=0100644 ouid=0 ogid=0
   rdev=00:00 nametype=NORMAL
6 | type=CWD msg=audit(1598052892.473:1153187): cwd="/root"
7 | type=SYSCALL msg=audit(1598052892.473:1153187): arch=c000003e
   syscall=2 success=yes exit=3 a0=7fd808c1d7a4 a1=80000 a2=1b6
   a3=80000 items=1 ppid=1281 pid=3010 auid=1000 uid=0 gid=0
   euid=0 suid=0 egid=107 sgid=107 fsgid=107 tty=pts0
   ses=3 comm="crontab" exe="/usr/bin/crontab" key="
   listadeusuarios"
```

Comando aureport

Para exibir um sumário de todos os eventos, execute o comando **aureport** com a opção **--summary**:

```
1 | aureport --summary
2 |
3 | Summary Report
4 | =====
5 | Range of time in logs: 08/21/2020 19:07:55.515 - 08/21/2020
   23:08:10.133
6 | Selected time for report: 08/21/2020 19:07:55 - 08/21/2020
   23:08:10.133
7 | Number of changes in configuration: 13
8 | Number of changes to accounts, groups, or roles: 0
9 | Number of logins: 0
```

```
10 | Number of failed logins: 0
11 | Number of authentications: 1
12 | Number of failed authentications: 0
13 | Number of users: 3
14 | Number of terminals: 6
15 | Number of host names: 1
16 | Number of executables: 8
17 | Number of commands: 6
18 | Number of files: 1
19 | Number of AVC's: 0
20 | Number of MAC events: 0
21 | Number of failed syscalls: 0
22 | Number of anomaly events: 0
23 | Number of responses to anomaly events: 0
24 | Number of crypto events: 0
25 | Number of integrity events: 0
26 | Number of virt events: 0
27 | Number of keys: 3
28 | Number of process IDs: 31
29 | Number of events: 71445
```

Para exibir um sumário dentro de uma faixa de data e hora, use as flags `--start` e `--end`. Exemplo:

```
1 | aureport -s --start 01/01/2020 00:00 --end 12/31/2020
    23:59
```

Para visualizar eventos sobre arquivos, execute o comando **aureport** com o parâmetro `-f`:

```
1 | aureport -f | tail
2 | 71135. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270418
3 | 71136. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270419
4 | 71137. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270420
5 | 71138. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270421
6 | 71139. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270422
7 | 71140. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270423
8 | 71141. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000
   270424
```

```
9 | 71142. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000  
    270425  
10 | 71143. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000  
    270426  
11 | 71144. 08/21/2020 23:11:13 /etc/passwd 2 yes /sbin/aureport 1000  
    270427
```

Contas de usuários

Para visualizar eventos sobre contas de usuários, execute o comando **aureport** com o parâmetro **-m**. Antes não esqueça de criar um novo usuário.

```
1 | adduser linus
```

Preencha as informações do usuário linus.

```
1 | aureport -m  
2  
3 Account Modifications Report  
4 =====  
5 # date time auid addr term exe acct success event  
6 =====  
7 1. 08/21/2020 23:26:13 1000 ? pts/0 /usr/sbin/groupadd ? yes  
    710453  
8 2. 08/21/2020 23:26:13 1000 ? pts/0 /usr/sbin/groupadd ? yes  
    710454  
9 3. 08/21/2020 23:26:13 1000 ? pts/0 /usr/sbin/groupadd ? yes  
    710455  
10 4. 08/21/2020 23:26:14 1000 ? pts/0 /usr/sbin/useradd ? yes  
    710462  
11 5. 08/21/2020 23:26:18 1000 ? pts/0 /usr/bin/passwd linus yes  
    710478
```

Pluggable Authentication Modules

Conceito

O Linux PAM (*Pluggable Authentication Modules*) é um método altamente flexível para implementar serviços de autenticação em aplicativos e vários serviços de sistema.

A ferramenta **auditd** usa o módulo PAM **pam_tty_audit** para habilitar ou desabilitar a

auditoria de entrada TTY para usuários específicos. Depois que um usuário é configurado para ser auditado, **pam_tty_audit** funciona em conjunto com o **auditd** para rastrear as ações do usuário no terminal e, se configurado, captura as teclas exatas que o usuário faz e, em seguida, registra no arquivo **/var/log/audit/audit**.

LAB 2.4 - Auditoria em tempo real

Neste laboratório vamos aprender como monitorar todos os comandos de usuários pela biblioteca do pam autenticada pelo **auditd**:

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário **root**.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Adicione no final do arquivo **/etc/pam.d/common-password**, a linha que ativa o módulo **pam_tty_audit.so**.

```
1 | vim /etc/pam.d/common-password +$
2 | ....
3 | session required pam_tty_audit.so disable=*
   enable=root
   log_passwd
```

Com esta linha configurada, qualquer comando que o **root** executar será gravado no arquivo de logs **audit.log**.

Envie para o final do arquivo **/etc/audit/rules.d/audit.rules**, regras para auditar comandos executados pelo usuário **root**:

```
1 | echo '-a exit,always -F arch=b64 -F euid=0 -S execve' >> /etc/
   audit/rules.d/audit.rules
2 | echo '-a exit,always -F arch=b32 -F euid=0 -S execve' >> /etc/
   audit/rules.d/audit.rules
```

Reinicie o serviço do **auditd** para aplicar as alterações:

```
1 | service auditd restart
```

Para testar execute um comando para listar arquivos do diretório **/opt**.

```
1 | ls /opt
```

Verifique se o auditd registrou o comando no arquivo **/var/log/audit/audit.log**:

```
1 | grep '/opt' /var/log/audit/audit.log
2 | type=EXECVE msg=audit(1598055542.469:1685468): argc=2 a0="ls" a1
   ="/opt/"
3 | type=EXECVE msg=audit(1598055752.621:1685470): argc=3 a0="grep"
   a1="/opt" a2="/var/log/audit/audit.log"
```

audisdp-plugins

Conceito

O pacote **audisdp-plugins** fornece extensões para a interface em tempo real do sistema de auditoria, audisdp. Estas extensões podem fazer coisas como retransmitir eventos para máquinas remotas ou analisar eventos em busca de comportamentos suspeitos.

LAB 2.5 - Gerenciar retransmissão de logs

Neste laboratório vamos aprender como interromper entradas de registro de auditoria gravadas em logs no arquivo **/var/log/messages**.

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário root.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Antes de gerenciar retransmissão de logs, precisamos instalar o plugin **audisdp-plugins**:

```
1 | apt install audisdp-plugins -y
```

Em seguida edite o arquivo **/etc/audisp/plugins.d/syslog.conf** e altere a linha **args = LOG_INFO** para **args = LOG_LOCAL6**.

```
1 | vim /etc/audisp/plugins.d/syslog.conf
2 | # This file controls the configuration of the syslog plugin.
3 | # It simply takes events and writes them to syslog. The
4 | # arguments provided can be the default priority that you
5 | # want the events written with. And optionally, you can give
6 | # a second argument indicating the facility that you want events
7 | # logged to. Valid options are LOG_LOCAL0 through 7, LOG_AUTH,
8 | # LOG_AUTHPRIV, LOG_DAEMON, LOG_SYSLOG, and LOG_USER.
9 |
10 | active = yes
11 | direction = out
12 | path = builtin_syslog
13 | type = builtin
14 | args = LOG_LOCAL6
15 | format = string
```

O próximo passo é editar o arquivo **/etc/rsyslog.d/50-default.conf**, onde vamos bloquear todos os logs de auditoria enviados para o arquivo **/var/log/messages**:

```
1 | vim /etc/rsyslog.d/50-default.conf
```

Modifique:

```
1 | *.=info;*.=notice;*.=warn;\ \
2 |     auth,authpriv.none;\ \
3 |     cron,daemon.none;\ \
4 |     mail,news.none -/var/log/messages
```

Para:

```
1 | *.=info;*.=notice;*.=warn;\ \
2 |     auth,authpriv.none;\ \
3 |     cron,daemon.none;\ \
4 |     mail,news.none;local6.none -/var/log/messages
```

Reinic peace os serviços do auditd e rsyslog para aplicas as alterações:

```
1 | service auditd restart
2 | service rsyslog restart
```

Testar auditoria de comandos

Para testar execute um comando para listar arquivos do diretório **/opt**.

```
1 |ls /opt
```

Verifique se o auditd registrou o comando no arquivo **/var/log/audit/audit.log**:

```
1 grep '/opt' /var/log/audit/audit.log
2 type=EXECVE msg=audit(1598055542.469:1685468): argc=2 a0="ls" a1
   ="/opt/"
3 type=EXECVE msg=audit(1598055752.621:1685470): argc=3 a0="grep"
   a1="/opt" a2="/var/log/audit/audit.log"
4 type=EXECVE msg=audit(1598057166.789:1685888): argc=2 a0="ls" a1
   ="/opt/"
5 type=EXECVE msg=audit(1598057175.341:1685893): argc=3 a0="grep"
   a1="/opt" a2="/var/log/messages"
6 type=EXECVE msg=audit(1598057183.805:1685896): argc=3 a0="grep"
   a1="/opt" a2="/var/log/audit/audit.log"
```

Agora verifique que o arquivo **/var/log/messages**, não registrou o evento:

```
1 |grep '/opt' /var/log/messages
```

Mais informações:

[1] Man page do Syscalls: <https://man7.org/linux/man-pages/man2/syscalls.2.html>

3

Gerenciar logs remotamente

Competências deste conteúdo

- Configurar servidor de logs remoto
- Logs remotos utilizando a criptografia TLS
- Configuração de armazenamento de logs
- Planejamento de capacidade e backup
- Armazenando logs no MySQL
- Planejamento de capacidade e backup do mysql

Log remoto

Conceito

Para a melhor administração de logs em um ambiente corporativo, é necessário a utilização de ferramentas para a centralização de logs, de forma a facilitar o gerenciamento e garantir que os dados fiquem salvos.

O serviço padrão de Linux para o registro de logs, o **rsyslog**, já possui suporte tanto ao cliente quanto ao servidor de logs remotos. É preciso realizar ajustes nas configurações do arquivo **/etc/rsyslog.conf**, para que a máquina possa receber logs via TCP e/ou UDP na porta 514, o que é padrão do servidor de logs remoto.

LAB 3.1 - Gerenciar centralização de logs

Neste laboratório vamos aprender como criar um servidor de logs no Linux.

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário root.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Para preparar o servidor de logs, edite o arquivo **/etc/rsyslog.conf** para ativar os protocolos UDP e TCP e a porta do servidor de logs.

```
1 | vim /etc/rsyslog.conf
```

Descomente as linhas das diretivas **imudp** e **imtcp**, conforme o exemplo abaixo:

```
1 | # provides UDP syslog reception
2 | module(load="imudp")
3 | input(type="imudp" port="514")
4 |
5 | # provides TCP syslog reception
6 | module(load="imtcp")
7 | input(type="imtcp" port="514")
```

Descrição das opções utilizadas:

- **module(load=“imudp”)**: oferece a capacidade de receber mensagens syslog via UDP;
- **input(type=“imudp” port=“514”)**: especifica a porta na qual o servidor deve escutar via UDP;
- **module(load=“imtcp”)**: oferece a capacidade de receber mensagens syslog via TCP;
- **input(type=“imtcp” port=“514”)**: especifica a porta na qual o servidor deve escutar via TCP.

É preciso reiniciar o serviço do rsyslog, para aplicar as configurações:

```
1 | systemctl restart rsyslog
```

Verifique se o serviço do Rsyslog esta atendendo requisições na porta 514/TCP e 514/UDP:

```
1 | ss -nlptu | grep 514
```

```

2 | udp    UNCONN    0      0          *:514          *:*
     |           users:(("rsyslogd",pid=1194,fd=5))
3 | udp    UNCONN    0      0          :::514          :::*
     |           users:(("rsyslogd",pid=1194,fd=6))
4 | tcp    LISTEN     0      25         *:514          *:*
     |           users:(("rsyslogd",pid=1194,fd=7))
5 | tcp    LISTEN     0      25         :::514          :::*
     |           users:(("rsyslogd",pid=1194,fd=8))

```

Redirecionando arquivo de logs

Na configuração atual, qualquer envio de log será redirecionado para os arquivos padrões do servidor, dificultando a administração e gerenciamento.

Para realizar a configuração de arquivos distintos de logs, será necessário reconfigurar o rsyslog:

```
1 | vim /etc/rsyslog.d/template.conf
```

Adicione no arquivo:

```

1 | template (name="LogRemoto" type="string" string="/srv/log/%
HOSTNAME%/%PROGRAMNAME%.log")
2 | *.* ?LogRemoto

```

Descrição das opções utilizadas:

- **%HOSTNAME%**: mostra o nome do host que enviou o log;
- **%PROGRAMNAME%**: mostra o nome do programa do log;
- **%SYSLOGFACILITY%**: envia a facilidade do log;
- **%SYSLOGTAG%**: mostra a tag da mensagem.

Existem diversas variáveis do rsyslog com funções de separar os arquivos de log para a melhor gerência.

Acesse: <http://www.rsyslog.com/doc/master/configuration/properties.html> para verificar a lista completa.

Em seguida crie o diretório de base para armazenar os logs remotos e altere a permissão de acesso:

```

1 | mkdir /srv/log
2 | chown syslog:syslog -R /srv/log

```

É preciso reiniciar o serviço do **rsyslog**, para aplicar as configurações:

```
1 | systemctl restart rsyslog
```

Verifique se o serviço do Rsyslog esta em execução:

```
1 | systemctl status rsyslog
2 * rsyslog.service - System Logging Service
3   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled;
4         vendor preset: enabled)
4   Active: active (running) since Wed 2020-11-11 19:41:31 UTC;
5         10s ago
5     Docs: man:rsyslogd(8)
6           http://www.rsyslog.com/doc/
7   Main PID: 3090 (rsyslogd)
8     Tasks: 10 (limit: 2945)
9    CGroup: /system.slice/rsyslog.service
10       `-- 3090 /usr/sbin/rsyslogd -n
11
12 Nov 11 19:41:31 graylog systemd[1]: Starting System Logging
13 Service...
13 Nov 11 19:41:31 graylog rsyslogd[3090]: imuxsock: Acquired UNIX
14 socket '/run/systemd/journal/syslog' (fd 3) from systemd.
14 Nov 11 19:41:31 graylog systemd[1]: Started System Logging
15 Service.
15 Nov 11 19:41:31 graylog rsyslogd[3090]: rsyslogd's groupid
16 changed to 106
16 Nov 11 19:41:31 graylog rsyslogd[3090]: rsyslogd's userid
17 changed to 102
17 Nov 11 19:41:31 graylog rsyslogd[3090]: [origin software="
      rsyslogd" swVersion="8.32.0" x-pid="3090" x-info="http://www
      .r
```

Preparação dos clientes para enviar logs

Em outros terminais acesse as VMs **webserver-audit** e **kibana-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.11
```

```
1 | ssh suporte@172.16.0.13
```

Nas duas VMs adicione no final do arquivo **/etc/rsyslog.conf**, a configuração de envio de logs para o servidor **graylog**:

```
1 | sudo vim /etc/rsyslog.conf +$
```

```
1 | *.* @@graylog
```

| Atente-se que “@” envia logs via protocolo UDP, já “@@” envia via TCP.

É preciso reiniciar o serviço do rsyslog, para aplicar as configurações:

```
1 | sudo systemctl restart rsyslog
```

Alterne para a VM **graylog-audit**, e verifique se as outras máquinas estão enviando logs para o nosso servidor:

```
1 | sudo ls -R /srv/log/
2 | /srv/log/:
3 | graylog  kibana  webserver
4 |
5 | /srv/log/graylog:
6 | rsyslogd.log  sudo.log  systemd.log
7 |
8 | /srv/log/kibana:
9 | rsyslogd.log  sudo.log  systemd.log
10 |
11 | /srv/log/webserver:
12 | dockerd.log  polkitd.log  rsyslogd.log  sudo.log  systemd.log
```

Criptografar o envio de logs

Conceito

Ao realizar o envio de logs sem criptografia, podemos ficar suscetíveis a ataques do tipo **Man In The Middle**, ou seja, a obtenção destes arquivos em texto puro e até mesmo a alteração do mesmo.

A criptografia TLS é um tipo de criptografia assimétrica onde as chaves de cliente e servidor

são diferentes, dificultando a descriptografia.

LAB 3.2 - Logs remotos com criptografia TLS

Neste laboratório vamos aprender como gerenciar logs remotos, utilizando criptografia TLS.

Para começar acesse as VMs através do comando **ssh** e alterne para conta do usuário **root**.

```
1 | ssh suporte@172.16.0.11
2 | sudo su -
```

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

```
1 | ssh suporte@172.16.0.13
2 | sudo su -
```

Etapa 1 – Preparar servidor e clientes

Na VM **graylog-audit** instale os pacotes para gerenciar certificados.

```
1 | apt install rsyslog-gnutls gnutls-bin -y
```

Em seguida crie e acesse a pasta **/etc/rsyslog-keys**, onde vamos armazenar chaves e certificados.

```
1 | mkdir /etc/rsyslog-keys
2 | cd /etc/rsyslog-keys
```

Na VM **kibana-audit** instale o pacote que da suporte TLS ao Rsyslog.

```
1 | apt install rsyslog-gnutls -y
```

Em seguida crie a pasta **/etc/rsyslog-keys**, onde vamos armazenar chaves e certificados.

```
1 |mkdir /etc/rsyslog-keys
```

Na VM **webserver-audit** instale o pacote que da suporte TLS ao Rsyslog.

```
1 |yum install rsyslog-gnutls -y
```

Em seguida crie a pasta **/etc/rsyslog-keys**, onde vamos armazenar chaves e certificados.

```
1 |mkdir /etc/rsyslog-keys
```

Etapa 2 – Gerar certificado autoassinado para o servidor

Na VM **graylog-audit** gere uma chave privada de nome **ca-key.pem**.

```
1 |certtool --generate-privkey --outfile ca-key.pem
```

E defina a permissão onde somente o **root** terá acesso.

```
1 |chmod 400 ca-key.pem
```

Gere o certificado autoassinado de nome **ca.pem**, utilizando a chave privada **ca-key.pem**.

```
1 |certtool --generate-self-signed --load-privkey ca-key.pem --  
      outfile ca.pem
```

Acompanhe o preenchimento das informações do certificado:

```
1 Generating a self signed certificate...  
2 Please enter the details of the certificate's distinguished name  
   . Just press enter to ignore a field.  
3 Common name: graylog <--- Digite o HOSTNAME do servidor  
4 UID: <Tecle Enter>  
5 Organizational unit name: <Tecle Enter>  
6 Organization name: <Tecle Enter>  
7 Locality name: <Tecle Enter>  
8 State or province name: <Tecle Enter>  
9 Country name (2 chars): <Tecle Enter>  
10 Enter the subject's domain component (DC): <Tecle Enter>  
11 This field should not be used in new certificates.
```

```
12 | E-mail: <Tecle Enter>
13 | Enter the certificate's serial number in decimal (default:
14 |   6680410231240074733): <Tecle Enter>
15 |
16 | Activation/Expiration time.
17 | The certificate will expire in (days): 3650 <--- Digite a
18 |   quantidade de dias em que o certificado irá expirar
19 |
20 | Extensions.
21 | Does the certificate belong to an authority? (y/N): y
22 | Path length constraint (decimal, -1 for no constraint): -1
23 | Is this a TLS web client certificate? (y/N): n
24 | Will the certificate be used for IPsec IKE operations? (y/N): n
25 | Is this a TLS web server certificate? (y/N): n
26 | Enter a dnsName of the subject of the certificate: graylog <---
27 |   Digite o HOSTNAME do servidor
28 | Enter a dnsName of the subject of the certificate: <Tecle Enter>
29 | Enter a URI of the subject of the certificate: <Tecle Enter>
30 | Enter the IP address of the subject of the certificate: <Tecle
31 |   Enter>
32 | Enter the e-mail of the subject of the certificate: <Tecle Enter
33 |   >
34 | Will the certificate be used to sign OCSP requests? (y/N): n
35 | Will the certificate be used to sign code? (y/N): n
36 | Will the certificate be used for time stamping? (y/N): n
37 | Will the certificate be used for email protection? (y/N): n
38 | Will the certificate be used to sign other certificates? (y/N):
39 |   y
40 | Will the certificate be used to sign CRLs? (y/N): y
41 | Enter the URI of the CRL distribution point: <Tecle Enter>
42 | X.509 Certificate Information:
43 |   Version: 3
44 |   Serial Number (hex): 5f4844150180c684
45 |   Validity:
46 |     Not Before: Thu Aug 27 23:39:01 UTC 2020
47 |     Not After: Sun Aug 25 23:39:06 UTC 2030
48 |   Subject: CN=graylog
49 |   Subject Public Key Algorithm: RSA
50 |   Algorithm Security Level: High (3072 bits)
51 |   Modulus (bits 3072):
52 |     00:b8:0d:57:bc:49:2f:d2:a6:f6:2b:52:52:8a:ee:18
53 |     2c:92:0e:df:95:e1:e2:5d:d6:c3:af:5f:fa:4d:c9:46
54 |     cb:3e:50:f0:fc:e1:b2:3b:ec:74:84:2e:cf:c3:db:b5
55 |     cb:01:09:dc:00:90:fd:70:ab:4b:27:6b:7a:64:4d:27
56 |     09:b1:d7:60:65:51:eb:cd:3c:3d:0e:b9:cb:02:58:d6
```

```
53      d3:80:39:b0:6f:ee:fc:39:40:61:92:59:fb:77:71:83
54      8a:7d:75:28:e0:6a:79:6e:af:d3:ff:4b:af:84:71:6b
55      7c:f6:30:a7:42:92:63:02:e6:55:8e:6f:78:f6:e1:42
56      2e:21:10:40:ab:67:ba:0e:22:fb:b7:8d:c2:42:31:fd
57      2d:5c:73:f7:af:d3:7e:12:13:71:f3:46:c4:0c:16:07
58      09:c1:22:b9:5f:61:55:b0:5f:61:43:2d:92:6c:de:d4
59      fa:37:a5:b1:f6:35:66:20:06:c6:4c:61:28:f5:79:ae
60      f3:37:9c:a0:91:14:6a:d8:4f:a2:84:74:cb:f3:76:95
61      4e:e7:bc:29:33:a1:2a:8e:2d:65:e0:59:92:96:8b:61
62      b9:0b:8b:66:6c:9c:b8:0e:8e:d4:a0:a6:d6:93:8e:c7
63      7c:3c:96:27:b3:4e:6f:15:c8:08:73:29:7b:1c:b7:82
64      a6:ec:be:73:ed:59:d5:a7:cf:15:9d:f7:b9:8d:1c:50
65      4e:3d:aa:75:de:a4:96:3d:0e:1f:02:92:54:fd:a2:dd
66      3a:83:3e:f2:19:8a:70:49:18:73:23:14:74:be:bb:7e
67      48:cf:c8:14:0d:68:3e:6a:da:2c:ee:b5:a9:bc:a7:d8
68      7f:78:00:63:4f:7a:10:d1:72:91:c6:5a:75:12:da:8b
69      20:fb:19:f3:19:72:3a:24:13:1e:92:5c:33:76:ee:ff
70      fc:dd:82:71:18:0f:7d:73:4d:f1:50:54:d3:c6:f7:83
71      f0:f8:c9:5b:28:33:f3:a2:d8:e1:8c:2a:7d:fa:bd:6e
72      d5
73      Exponent (bits 24):
74          01:00:01
75      Extensions:
76          Basic Constraints (critical):
77              Certificate Authority (CA): TRUE
78          Subject Alternative Name (not critical):
79              DNSname: graylog
80          Key Usage (critical):
81              Certificate signing.
82              CRL signing.
83          Subject Key Identifier (not critical):
84              977295b3fa656a7b731df55c38d377b269f58811
85      Other Information:
86          Public Key ID:
87              sha1:977295b3fa656a7b731df55c38d377b269f58811
88              sha256:842265
89                  a3d16ed740c8a2ff8062bb27f616bbaa66f7c520672c6b166b39cfa3d2
90          Public key's random art:
91              +- [ RSA 3072 ] ---+
92              |                   |
93              |                   E. |
94              |                   +. o |
95              |                   o.o= B|
96              |                   S + .o XB|
97              |                   + .. =.+|
98              |                   . .o o|
```

```
98 |           ..+o o|
99 |           .+o o |
100 +-----+
101
102 Is the above information ok? (y/N): y
103
104
105 Signing certificate...
```

Etapa 3 – Gerar chaves para os clientes

Gere uma chave privada de nome **webserver-key.pem** para a máquina **webserver-audit**.

```
1 | certtool --generate-privkey --outfile webserver-key.pem --bits
   | 2048
```

Gere uma chave privada de nome **kibana-key.pem** para a máquina **kibana-audit**.

```
1 | certtool --generate-privkey --outfile kibana-key.pem --bits 2048
```

Verifique se as chaves que criamos estão presentes no servidor.

```
1 | ls -l *key.pem
2 | -r----- 1 root root 8167 Nov 11 21:02 ca-key.pem
3 | -rw----- 1 root root 5628 Nov 11 21:06 kibana-key.pem
4 | -rw----- 1 root root 5628 Nov 11 21:05 webserver-key.pem
```

Etapa 4 – Gerar requisição de assinatura do certificado para os clientes

Gere uma requisição de assinatura do certificado de nome **webserver-request.pem**, utilizando a chave privada que criamos para a máquina **webserver-audit**.

```
1 | certtool --generate-request --load-privkey webserver-key.pem --
   | outfile webserver-request.pem
```

Acompanhe o preenchimento das informações da requisição de assinatura do certificado:

```
1 | Generating a PKCS #10 certificate request...
2 | Common name: webserver <--- Digite o HOSTNAME do servidor
   | webserver
```

```
3 |Organizational unit name: <Tecle Enter>
4 |Organization name: <Tecle Enter>
5 |Locality name: <Tecle Enter>
6 |State or province name: <Tecle Enter>
7 |Country name (2 chars): <Tecle Enter>
8 |Enter the subject's domain component (DC): <Tecle Enter>
9 |UID: <Tecle Enter>
10 |Enter a dnsName of the subject of the certificate: webserver
     <--- Digite o HOSTNAME do servidor webserver
11 |Enter a dnsName of the subject of the certificate: <Tecle Enter>
12 |Enter a URI of the subject of the certificate: <Tecle Enter>
13 |Enter the IP address of the subject of the certificate: <Tecle
     Enter>
14 |Enter the e-mail of the subject of the certificate: <Tecle Enter
     >
15 |Enter a challenge password: <Tecle Enter>
16 |Does the certificate belong to an authority? (y/N): n
17 |Will the certificate be used for signing (DHE ciphersuites)? (Y/
     n): y
18 |Will the certificate be used for encryption (RSA ciphersuites)?
     (Y/n): n
19 |Will the certificate be used to sign code? (y/N): n
20 |Will the certificate be used for time stamping? (y/N): n
21 |Will the certificate be used for email protection? (y/N): n
22 |Will the certificate be used for IPsec IKE operations? (y/N): n
23 |Will the certificate be used to sign OCSP requests? (y/N): n
24 |Is this a TLS web client certificate? (y/N): n
25 |Is this a TLS web server certificate? (y/N): n
```

Gere uma requisição de assinatura do certificado de nome **kibana-request.pem**, utilizando a chave privada que criamos para a máquina **kibana-audit**.

```
1 |certtool --generate-request --load-privkey kibana-key.pem --
     outfile kibana-request.pem
```

Acompanhe o preenchimento das informações da requisição de assinatura do certificado:

```
1 |Generating a PKCS #10 certificate request...
2 |Common name: kibana <--- Digite o HOSTNAME do servidor kibana
3 |Organizational unit name: <Tecle Enter>
4 |Organization name: <Tecle Enter>
5 |Locality name: <Tecle Enter>
6 |State or province name: <Tecle Enter>
7 |Country name (2 chars): <Tecle Enter>
```

```
8 | Enter the subject's domain component (DC): <Tecle Enter>
9 | UID: <Tecle Enter>
10 | Enter a dnsName of the subject of the certificate: kibana <---
     |   Digite o HOSTNAME do servidor kibana
11 | Enter a dnsName of the subject of the certificate: <Tecle Enter>
12 | Enter a URI of the subject of the certificate: <Tecle Enter>
13 | Enter the IP address of the subject of the certificate: <Tecle
     |   Enter>
14 | Enter the e-mail of the subject of the certificate: <Tecle Enter
     |   >
15 | Enter a challenge password: <Tecle Enter>
16 | Does the certificate belong to an authority? (y/N): n
17 | Will the certificate be used for signing (DHE ciphersuites)? (Y/
     |   n): y
18 | Will the certificate be used for encryption (RSA ciphersuites)? (Y/n): n
19 | Will the certificate be used to sign code? (y/N): n
20 | Will the certificate be used for time stamping? (y/N): n
21 | Will the certificate be used for IPsec IKE operations? (y/N): n
22 | Will the certificate be used to sign OCSP requests? (y/N): n
23 | Is this a TLS web client certificate? (y/N): n
24 | Is this a TLS web server certificate? (y/N): n
```

Verifique se as requisições que criamos estão presentes no servidor.

```
1 | ls -l *request.pem
2 | -rw----- 1 root root 2520 Nov 11 21:09 kibana-request.pem
3 | -rw----- 1 root root 2535 Nov 11 21:08 webserver-request.pem
```

Etapa 5 – Gerar certificado autoassinado para o cliente

Gere o certificado autoassinado de nome **webserver-cert.pem**, utilizando a requisição de assinatura **webserver-request.pem**, através da chave privada **ca-key.pem**.

```
1 | certtool --generate-certificate --load-request webserver-request
     | .pem --outfile webserver-cert.pem --load-ca-certificate ca.
     | pem --load-ca-privkey ca-key.pem
```

Acompanhe o preenchimento das informações do certificado:

```
1 | Generating a signed certificate...
2 | Enter the certificate's serial number in decimal (default:
     | 6865818189034885834): <Tecle Enter>
```

```
3
4
5 Activation/Expiration time.
6 The certificate will expire in (days): 1000
7
8
9 Extensions.
10 Do you want to honour all the extensions from the request? (y/N)
    : <Tecle Enter>
11 Does the certificate belong to an authority? (y/N): <Tecle Enter
    >
12 Is this a TLS web client certificate? (y/N): y
13 Will the certificate be used for IPsec IKE operations? (y/N): <
    Tecle Enter>
14 Is this a TLS web server certificate? (y/N): y
15 Enter a dnsName of the subject of the certificate: webserver
    <--- Digite o HOSTNAME do servidor webserver
16 Enter a dnsName of the subject of the certificate: <Tecle Enter>
17 Enter a URI of the subject of the certificate: <Tecle Enter>
18 Enter the IP address of the subject of the certificate: <Tecle
    Enter>
19 Will the certificate be used for signing (DHE ciphersuites)? (Y/
    n): n
20 Will the certificate be used for encryption (RSA ciphersuites)?
    (Y/n): n
21 Will the certificate be used to sign OCSP requests? (y/N): n
22 Will the certificate be used to sign code? (y/N): n
23 Will the certificate be used for time stamping? (y/N): n
24 Will the certificate be used for email protection? (y/N): n
25 X.509 Certificate Information:
    Version: 3
    Serial Number (hex): 5f48494429d712ca
    Validity:
        Not Before: Fri Aug 28 00:01:20 UTC 2020
        Not After: Thu May 25 00:01:25 UTC 2023
    Subject: CN=security
    Subject Public Key Algorithm: RSA
    Algorithm Security Level: Medium (2048 bits)
    Modulus (bits 2048):
        00:e0:c7:a9:b1:0a:5d:4d:b5:ed:87:b4:15:74:0e:0a
        d6:e7:a6:b2:10:2e:7a:db:4a:f3:d3:e1:0a:1c:4b:ef
        82:a2:9c:f7:6b:d3:26:4b:a2:89:96:a9:b0:ba:39:fc
        e2:47:ab:3b:21:9d:e2:7f:39:7e:bd:e4:64:ad:66:e8
        36:3a:41:ad:79:35:61:c4:c0:0c:69:89:29:8e:03:1f
        42:18:8d:a0:b1:8d:24:db:ac:b2:ef:b2:d2:ed:8a:1b
        26:0f:0e:2f:de:4a:02:6f:f6:5c:2e:16:49:b0:c9:e9
        4b:bc:99:64:a4:8b:51:29:49:9e:97:60:18:bd:87:40
```

```
43      68:37:c5:cb:df:42:a5:4b:6d:d7:46:ea:43:cb:2b:83
44      52:9c:99:58:eb:f6:6b:e3:a2:78:e9:c3:c3:49:86:a7
45      06:5c:9b:72:ff:2a:5f:75:67:eb:79:da:a8:04:48:ee
46      f7:3b:32:59:e4:80:48:75:af:02:05:80:c1:8a:09:4d
47      bf:39:d7:4b:35:21:42:55:a4:29:55:81:a3:93:6e:17
48      d4:38:ff:41:13:c2:7a:25:ed:32:e0:85:85:98:62:91
49      2e:73:48:a2:bd:89:44:b0:00:66:16:c9:fe:a3:12:ce
50      5b:57:1d:96:84:8a:98:0c:30:43:b9:1f:7e:dd:df:cd
51      e3
52      Exponent (bits 24):
53      01:00:01
54  Extensions:
55      Basic Constraints (critical):
56      Certificate Authority (CA): FALSE
57      Key Purpose (not critical):
58      TLS WWW Client.
59      TLS WWW Server.
60      Subject Alternative Name (not critical):
61      DNSname: security
62      Subject Key Identifier (not critical):
63      f1d7ef094b6bc59f9dfa1adf8625ffab9ca6c925
64      Authority Key Identifier (not critical):
65      977295b3fa656a7b731df55c38d377b269f58811
66 Other Information:
67      Public Key ID:
68      sha1:f1d7ef094b6bc59f9dfa1adf8625ffab9ca6c925
69      sha256:
70          a52fa9ce139f61e420d8be2fec79daf6d42adb79f43726209de8ecc90c5cb372
71      Public key's random art:
72      +---[ RSA 2048]---+
73      |
74      |
75      |
76      |       o   .
77      |       S   .   o
78      |       .   .+.
79      |       E   =.=*
80      |       .   *o0+B|
81      |       ++0=**|
82      +-----+
83 Is the above information ok? (y/N): y
84
85
86 Signing certificate...
```

Gere o certificado autoassinado de nome **kibana-cert.pem**, utilizando a requisição de assinatura **kibana-request.pem**, através da chave privada **ca-key.pem**.

```
1 certtool --generate-certificate --load-request kibana-request.
  pem --outfile kibana-cert.pem --load-ca-certificate ca.pem
  --load-ca-privkey ca-key.pem
```

Acompanhe o preenchimento das informações do certificado:

```
1 Generating a signed certificate...
2 Enter the certificate's serial number in decimal (default:
  6865819065109212558): <Tecla Enter>
3
4
5 Activation/Expiration time.
6 The certificate will expire in (days): 1000
7
8
9 Extensions.
10 Do you want to honour all the extensions from the request? (y/N)
    : <Tecla Enter>
11 Does the certificate belong to an authority? (y/N): <Tecla Enter
    >
12 Is this a TLS web client certificate? (y/N): y
13 Will the certificate be used for IPsec IKE operations? (y/N): <
    Tecla Enter>
14 Is this a TLS web server certificate? (y/N): y
15 Enter a dnsName of the subject of the certificate: kibana <---
    Digite o HOSTNAME do servidor kibana
16 Enter a dnsName of the subject of the certificate: <Tecla Enter>
17 Enter a URI of the subject of the certificate: <Tecla Enter>
18 Enter the IP address of the subject of the certificate: <Tecla
    Enter>
19 Will the certificate be used for signing (DHE ciphersuites)? (Y/
    n): n
20 Will the certificate be used for encryption (RSA ciphersuites)? (Y/n):
    n
21 Will the certificate be used to sign OCSP requests? (y/N): n
22 Will the certificate be used to sign code? (y/N): n
23 Will the certificate be used for time stamping? (y/N): n
24 Will the certificate be used for email protection? (y/N): n
25 X.509 Certificate Information:
    Version: 3
    Serial Number (hex): 5f484a1023f06d8e
```

```
28 |     Validity:
29 |         Not Before: Fri Aug 28 00:04:33 UTC 2020
30 |         Not After: Thu May 25 00:04:35 UTC 2023
31 |     Subject: CN=kibana
32 |     Subject Public Key Algorithm: RSA
33 |     Algorithm Security Level: Medium (2048 bits)
34 |         Modulus (bits 2048):
35 |             00:de:7c:5f:28:ef:e2:f1:b0:5a:08:a6:7b:9e:6c:06
36 |             e9:0d:fb:27:11:f7:a3:cb:e1:a0:71:1e:10:80:9b:16
37 |             52:a4:58:92:f0:83:c4:ab:ca:2a:e3:fc:a8:05:22:83
38 |             6c:28:5a:89:dc:6d:a0:49:b5:6a:06:b2:9f:ad:79:f9
39 |             39:cf:80:49:8b:69:a8:13:ff:85:1f:82:9a:84:6c:42
40 |             6f:1a:01:df:a9:75:88:0d:e5:d3:27:d0:e2:f2:9b:f3
41 |             03:08:a4:55:4d:6b:67:05:27:bd:0f:5e:5f:4d:fc:70
42 |             2a:4a:d7:97:5a:6f:39:0e:46:df:d7:90:b6:2c:70:56
43 |             fb:20:f8:9b:f8:7e:e0:35:67:00:80:42:dc:5d:64:0c
44 |             63:af:ef:45:38:48:c2:d5:c8:b9:c5:09:df:98:25:93
45 |             23:65:ce:ec:95:62:8e:a1:92:e7:cc:49:92:b1:08:61
46 |             ca:2a:cd:b3:4d:35:89:89:93:27:05:b7:9a:c1:40:3e
47 |             c5:86:9d:10:84:96:ac:0d:91:c7:bf:50:c8:1f:db:40
48 |             d9:21:c0:b1:4e:13:4d:15:54:83:f6:48:68:72:8b:d8
49 |             52:69:d9:e7:be:28:e4:58:e2:18:d6:c1:d6:e0:c4:ba
50 |             ca:78:c4:8d:20:ce:55:d0:69:2e:77:66:32:c1:9d:8c
51 |             db
52 |         Exponent (bits 24):
53 |             01:00:01
54 |     Extensions:
55 |         Basic Constraints (critical):
56 |             Certificate Authority (CA): FALSE
57 |         Key Purpose (not critical):
58 |             TLS WWW Client.
59 |             TLS WWW Server.
60 |         Subject Alternative Name (not critical):
61 |             DNSname: kibana
62 |         Subject Key Identifier (not critical):
63 |             aefa84a1fb45b69b141da00e674bf08ab72fdf73
64 |         Authority Key Identifier (not critical):
65 |             977295b3fa656a7b731df55c38d377b269f58811
66 |     Other Information:
67 |         Public Key ID:
68 |             sha1:aefa84a1fb45b69b141da00e674bf08ab72fdf73
69 |             sha256:29
70 |                 c281250393c351486aedcfe5fa9fd376315c462c05e348256dbf48f438cd5a
71 |             Public key's random art:
72 |                 +---[ RSA 2048 ]---+
73 |                 | . . . |
```

```
73 |      o . .
74 |      * .
75 |      . B . .
76 |      . o + + S
77 |      . o = +
78 |      o . = .
79 |      .o =.+E
80 |      .+=o*o
81 +-----+
82
83 Is the above information ok? (y/N): y
84
85
86 Signing certificate...
```

Verifique se os certificados dos clientes estão presentes no servidor.

```
1 | ls -l *cert.pem
2 | -rw-r--r-- 1 root root 1330 Nov 11 21:14 kibana-cert.pem
3 | -rw-r--r-- 1 root root 1338 Nov 11 21:13 webserver-cert.pem
```

Antes de continuar, remova as requisições, pois não são mais necessárias.

```
1 | rm *request.pem
```

Etapa 6 – Enviar chaves e certificados para os clientes

Envie a chave do servidor, chave e certificado do cliente **webserver** para a máquina **webserver-audit**.

```
1 | scp ca.pem webserver* suporte@webserver:/tmp
2 | suporte@webserver's password:
3 | ca.pem
4 |
5 |          100% 1444      1.3MB/s  00:00 webserver-cert.pem
6 |
7 |          100% 1338      1.6MB/s  00:00
8 | webserver-key.pem
9 |
10 |         100% 5628      5.4MB/s  00:00
```

Envie a chave do servidor, chave e certificado do cliente **kibana** para a máquina **kibana-audit**.

```
1 | scp ca.pem kibana* suporte@kibana:/tmp
2 | suporte@kibana's password:
3 | ca.pem
4 |
5 |          100% 1444      1.7MB/s   00:00
4 | kibana-cert.pem
5 |
6 |          100% 1330      2.1MB/s   00:00
5 | kibana-key.pem
7 |
8 |          100% 5628      7.8MB/s   00:00
```

Alterne para a máquina **webserver-audit** e mova a chave do servidor, chave e certificado do cliente para o diretório **/etc/rsyslog-keys**.

```
1 | mv /tmp/ca.pem /etc/rsyslog-keys/
2 | mv /tmp/webserver*.pem /etc/rsyslog-keys/
```

Alterne para a máquina **kibana-audit** e mova a chave do servidor, chave e certificado do cliente para o diretório **/etc/rsyslog-keys**.

```
1 | mv /tmp/ca.pem /etc/rsyslog-keys/
2 | mv /tmp/kibana*.pem /etc/rsyslog-keys/
```

Etapa 7 - Configurar Rsyslog para enviar logs de forma segura

Alterne para a máquina **graylog-audit** e crie um arquivo de configuração para o servidor Rsyslog.

```
1 | vim /etc/rsyslog.d/syslog-tls.conf
2 | $DefaultNetstreamDriver gtls
3 |
4 | $DefaultNetstreamDriverCAFile /etc/rsyslog-keys/ca.pem
5 |
6 | $DefaultNetstreamDriverCertFile /etc/rsyslog-keys/webserver-cert
7 | .pem
8 | $DefaultNetstreamDriverKeyFile /etc/rsyslog-keys/webserver-key.
9 | pem
10 | $DefaultNetstreamDriverCertFile /etc/rsyslog-keys/kibana-cert.
11 | pem
```

```
10 $DefaultNetstreamDriverKeyFile /etc/rsyslog-keys/kibana-key.pem
11
12 $ModLoad imtcp
13 $InputTCPServerStreamDriverMode 1
14 $InputTCPServerStreamDriverAuthMode anon
15 $InputTCPServerRun 6514
16
17 :fromhost, isEqual, "webserver"      /var/log/webserver/messages
18 :fromhost, isEqual, "webserver"      ~
19 :fromhost, isEqual, "kibana"         /var/log/kibana/messages
20 :fromhost, isEqual, "kibana"         ~
```

O arquivo modelo esta disponível na pasta **/opt** na VM graylog, ou no Github do curso:

```
1 | git clone https://github.com/4linux/4516.git
```

Descrição das opções utilizadas:

- **\$DefaultNetstreamDriver**: define o driver utilizado. A opção **gtls** utiliza o driver para GnutTLS, que é uma biblioteca TLS. É usado para transferência de mensagem criptografada. A outra opção é a **ptcp**, que é usado para transferência de mensagens não criptografadas.
- **\$DefaultNetstreamDriverCAFile**: define a localização e nome do arquivo de chave privada do servidor.
- **\$DefaultNetstreamDriverCertFile**: define a localização e nome do arquivo de certificado do cliente.
- **\$DefaultNetstreamDriverKeyFile**: define a localização e nome do arquivo de chave privada do cliente.
- **\$ModLoad**: define qual protocolo o Rsyslog vai utilizar para receber os logs.
- **\$InputTCPServerStreamDriverMode**: permite executar o driver somente no modo TLS.
- **\$InputTCPServerStreamDriverAuthMode**: define que se cliente precisa ou não estar autenticado.
- **\$InputTCPServerRun**: define a porta na qual o servidor deve escutar requisições.

Antes de continuar, é preciso ajustar as permissões da pasta **/etc/rsyslog-keys**:

```
1 | chown syslog:syslog -R /etc/rsyslog-keys
```

Edita o arquivo **/etc/rsyslog.conf** e comente as linhas das diretivas **imudp** e **imtcp**, conforme o exemplo abaixo:

```
1 | vim /etc/rsyslog.conf
2 | ....
3 |
4 | # provides UDP syslog reception
5 | #module(load="imudp")
6 | #input(type="imudp" port="514")
7 |
8 | # provides TCP syslog reception
9 | #module(load="imtcp")
10 | #input(type="imtcp" port="514")
```

É preciso reiniciar o serviço do rsyslog, para aplicar as configurações:

```
1 | systemctl restart rsyslog
```

Verifique se o serviço do Rsyslog esta atendendo requisições na porta 6514/TCP:

```
1 | ss -nlptu | grep 6514
2 | tcp      LISTEN    0          25          0.0.0.0:6514
      0.0.0.0:*      users:(("rsyslogd",pid=3278,fd=6))
3 | tcp      LISTEN    0          25          [:]:6514
      [:]:*        users:(("rsyslogd",pid=3278,fd=7))
```

Etapa 8 – Configurar Rsyslog para receber logs de forma segura

Nas máquinas **webserver-audit**, e **kibana-audit** e crie um arquivo de configuração para o servidor Rsyslog.

```
1 | vim /etc/rsyslog.d/syslog-tls.conf
2 | $DefaultNetStreamDriverCAFile /etc/rsyslog-keys/ca.pem
3 |
4 | $DefaultNetStreamDriver gtls
5 | $ActionSendStreamDriverMode 1
6 | $ActionSendStreamDriverAuthMode anon
7 |
8 | *.*      @@(o)graylog:6514
```

O arquivo modelo esta disponível na pasta **/opt** nas VMs security e kibana, ou no Github do curso:

```
1 |git clone https://github.com/4linux/4516.git
```

Ainda nas máquinas **security**, e **kibana**, edite o arquivo **/etc/rsyslog.conf** e comente a última linha.

```
1 |vim /etc/rsyslog.conf
2 |....
3 |#*.* @@graylog
```

É preciso reiniciar o serviço do rsyslog, para aplicar as configurações:

```
1 |systemctl restart rsyslog
```

Etapa 9 – Testando o envio de logs

Na máquina **graylog-audit**, verifique se os logs da VM **webserver** estão disponíveis no arquivo **/var/log/webserver/messages**.

```
1 |cat /var/log/webserver/messages
2 |Nov 11 21:32:28 webserver polkitd[2171]: Registered
  Authentication Agent for unix-process:26173:1229764 (system
  bus name :1.156 [/usr/bin/pktyagent --notify-fd 5 --
  fallback], object path /org/freedesktop/PolicyKit1/
  AuthenticationAgent, locale en_US.UTF-8)
3 |Nov 11 21:32:28 webserver systemd: Stopping System Logging
  Service...
4 |Nov 11 21:32:28 webserver rsyslogd: [origin software="rsyslogd"
  swVersion="8.24.0-52.el7_8.2" x-pid="26132" x-info="http://
  www.rsyslog.com"] exiting on signal 15.
5 |Nov 11 21:32:28 webserver systemd: Stopped System Logging
  Service.
6 |Nov 11 21:32:28 webserver systemd: Starting System Logging
  Service...
7 |Nov 11 21:32:28 webserver rsyslogd: [origin software="rsyslogd"
  swVersion="8.24.0-52.el7_8.2" x-pid="26179" x-info="http://
  www.rsyslog.com"] start
```

Aproveite e verifique se os logs da VM **kibana-audit** estão disponíveis no arquivo **/var/log/kibana/messages**.

```
1 |cat /var/log/kibana/messages
```

```
2 | Nov 11 21:31:14 kibana rsyslogd: [origin software="rsyslogd"
|   swVersion="8.1901.0" x-pid="2514" x-info="https://www.
|   rsyslog.com"] exiting on signal 15.
3 | Nov 11 21:31:14 kibana systemd[1]: Stopping System Logging
|   Service...
4 | Nov 11 21:31:14 kibana systemd[1]: rsyslog.service: Succeeded.
5 | Nov 11 21:31:14 kibana systemd[1]: Stopped System Logging
|   Service.
6 | Nov 11 21:31:14 kibana systemd[1]: Starting System Logging
|   Service...
7 | Nov 11 21:31:14 kibana systemd[1]: Started System Logging
|   Service.
8 | Nov 11 21:31:14 kibana rsyslogd: imuxsock: Acquired UNIX socket
|   '/run/systemd/journal/syslog' (fd 3) from systemd. [v8
|   .1901.0]
9 | Nov 11 21:31:14 kibana rsyslogd: [origin software="rsyslogd"
|   swVersion="8.1901.0" x-pid="2563" x-info="https://www.
|   rsyslog.com"] start
```

Armazenamento de logs no MySQL

Conceito

Para o melhor desempenho de gravação e leitura de logs, é necessário a inserção deles em um banco de dados. O rsyslog possui suporte aos bancos de dados MySQL, MariaDB, PostgreSQL, Elasticsearch, MongoDB e outros. Neste ambiente será utilizado o MySQL.

LAB 3.3 - Armazenar logs no banco de dados MySQL

Neste laboratório vamos aprender como armazenar logs no banco de dados MySQL.

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário root.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Antes de gerenciar retransmissão de logs, precisamos instalar o plugin **audisdp-plugins**.

Instalação e configuração do banco no MySQL

Instale os pacotes necessários para o MySQL, sem precisar de nenhuma interação durante a instalação.

```
1 | DEBIAN_FRONTEND=noninteractive apt install mysql-server mysql-client mysql-common rsyslog-mysql
```

O próximo passo é criar o banco de dados **Syslog**:

```
1 | mysql -u root -e 'CREATE DATABASE Syslog;' syst
```

Verifique se o banco **Syslog** foi criado corretamente:

```
1 | mysql -u root -D Syslog -e 'SHOW DATABASES;'
```

```
1 | +-----+  
2 | Database |  
3 | +-----+  
4 | Syslog |  
5 | information_schema |  
6 | mysql |  
7 | performance_schema |  
8 | +-----+  
9 | 4 rows in set (0.00 sec)
```

Utilize o comando **mysql** para criar as tabelas no banco **Syslog**:

```
1 | mysql -u root -D Syslog < /usr/share/dbconfig-common/data/rsyslog-mysql/install/mysql
```

Verifique se as tabelas **SystemEvents** e **SystemEventsProperties** foram criadas no banco **Syslog**

```
1 | mysql -u root -D Syslog -e 'SHOW TABLES;'
```

```
1 | +-----+  
2 | Tables_in_Syslog |
```

```
3 |-----+  
4 | SystemEvents  
5 | SystemEventsProperties  
6 |-----+
```

Para exibir a estrutura dos campos da tabela SystemEvents, utilize o comando **DESC**:

```
1 |mysql -u root -D Syslog -e 'DESC SystemEvents;'
```

1	Field	Type	Null	Key	Default	
2	Extra					
3	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
4	ID auto_increment	int(10) unsigned	NO	PRI	NULL	
5	CustomerID	bigint(20)	YES		NULL	
6	ReceivedAt	datetime	YES		NULL	
7	DeviceReportedTime	datetime	YES		NULL	
8	Facility	smallint(6)	YES		NULL	
9	Priority	smallint(6)	YES		NULL	
10	FromHost	varchar(60)	YES		NULL	
11	Message	text	YES		NULL	
12	NTSeverity	int(11)	YES		NULL	
13	Importance	int(11)	YES		NULL	
14	EventSource	varchar(60)	YES		NULL	
15	EventUser	varchar(60)	YES		NULL	
16	EventCategory	int(11)	YES		NULL	
17	EventID	int(11)	YES		NULL	
18	EventBinaryData	text	YES		NULL	

```

19 | MaxAvailable      | int(11)          | YES   |       | NULL    |
20 | CurrUsage         | int(11)          | YES   |       | NULL    |
21 | MinUsage          | int(11)          | YES   |       | NULL    |
22 | MaxUsage          | int(11)          | YES   |       | NULL    |
23 | InfoUnitID        | int(11)          | YES   |       | NULL    |
24 | SysLogTag         | varchar(60)     | YES   |       | NULL    |
25 | EventLogType      | varchar(60)     | YES   |       | NULL    |
26 | GenericFileName   | varchar(60)     | YES   |       | NULL    |
27 | SystemID          | int(11)          | YES   |       | NULL    |
28 +-----+-----+-----+-----+
29 24 rows in set (0.00 sec)

```

Configuração de usuário e senha

Uma vez criado o banco e tabelas, é preciso configurar o acesso a um usuário com uma senha:

```

1 | mysql -u root -D Syslog -e "CREATE USER rsysloguser@localhost
                                IDENTIFIED BY 'rsyslogpw';"

1 | mysql -u root -D Syslog -e 'GRANT ALL ON Syslog.* TO
                                rsysloguser@localhost;'
```

O comando CREATE USER esta criando o usuário rsysloguser com a senha rsyslogpw, onde o mesmo terá acesso via localhost. O comando GRANT ALL ON esta permitindo acesso total ao usuário rsysloguser, em todas as tabelas do banco Syslog.

Feito os ajustes de acesso no MySQL, edite o arquivo **/etc/rsyslog.d/mysql.conf** para informar o nome de usuário e senha.

```

1 | vim /etc/rsyslog.d/mysql.conf
2 | ### Configuration file for rsyslog-mysql
```

```
3 |### Changes are preserved
4 |
5 |$ModLoad ommysql
6 |.* action(type="ommysql" server="localhost" db="Syslog" uid=
|    rsysloguser" pwd="rsyslogpw")
```

É preciso reiniciar o serviço do rsyslog, para aplicar as configurações:

```
1 |systemctl restart rsyslog
```

Testar o armazenamento de logs no banco MySQL

Alterne para as máquinas **webserver-audit** e **kibana-audit** e reinicie o serviço do rsyslog, para aplicar as configurações:

```
1 |systemctl restart rsyslog
```

Alterne para a máquina **graylog-audit** e visualize os dados da tabela nos campos **ID**, **fromHost** e **Message**:

```
1 |mysql -u root -D Syslog -e 'SELECT ID,fromHost,Message FROM
|    SystemEvents';
```

```
1 +-----+
2 | ID   | fromHost | Message
3 |       |
4 | 1   | graylog  | Stopping System Logging Service...
5 |       |
6 | 2   | graylog  | [origin software="rsyslogd" swVersion
|     = "8.24.0" x-pid="4292" x-info="http://www.rsyslog.com"]
|     exiting on signal 15.
7 |       |
8 | 3   | graylog  | Stopped System Logging Service.
9 |       |
```

```
7 | 4 | graylog | Starting System Logging Service...
|
8 | 5 | graylog | node=graylog type=SERVICE_START msg=audit
(1598625468.312:7817): pid=1 uid=0 auid=4294967295 ses
=4294967295 msg='unit=rsyslog comm="systemd" exe="/lib/
systemd/systemd" hostname=? addr=? terminal=? res=success'
|
9 | 6 | graylog | node=graylog type=SERVICE_STOP msg=audit
(1598625468.312:7818): pid=1 uid=0 auid=4294967295 ses
=4294967295 msg='unit=rsyslog comm="systemd" exe="/lib/
systemd/systemd" hostname=? addr=? terminal=? res=success'
|
10 | 7 | graylog | node=graylog type=SYSCALL msg=audit
(1598625468.312:7819): arch=c000003e syscall=59 success=yes
exit=0 a0=7f386f107cc0 a1=7ffc2a613c80 a2=7ffc2a614228 a3=7
ffc2a613ee8 items=2 ppid=4722 pid=4723 auid=1000 uid=0 gid=0
euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=1
comm="systemd-tty-ask" exe="/bin/systemd-tty-ask-password-
agent" key=(null) |
11 | 8 | graylog | node=graylog type=EXECVE msg=audit
(1598625468.312:7819): argc=2 a0="/bin/systemd-tty-ask-
password-agent" a1="--watch"
|
12 | 9 | graylog | node=graylog type=CWD msg=audit
(1598625468.312:7819): cwd="/root"
```

Backup de logs no MySQL

Conceito

É de extrema importância manter cópias de segurança, ou seja, o backup de um banco de dados MySQL. O objetivo final é restaurar algum dado, em nosso caso, dados de logs, caso haja algum tipo de falha no sistema.

No MySQL podemos realizar backup e rode dados através da ferramenta **mysqldump**, ou backup binário do banco através da ferramenta **mysqlbinlog**.

LAB 3.4 - Realizar backup e restore de logs no banco MySQL

Neste laboratório vamos aprender como gerenciar backup de dados e binário de logs no MySQL.

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário root.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Backup dos dados

Para realizar o backup do banco do banco de dados **Syslog**, utilize a ferramenta **mysqldump**:

```
1 | mysqldump Syslog > backup-banco-syslog.sql
```

Verifique se o arquivo **backup-banco-syslog.sql** foi criado. Visualize o conteúdo do DUMP através do comando cat:

```
1 | cat backup-banco-syslog.sql
```

```
1 | -- MySQL dump 10.16 Distrib 10.1.45-MariaDB, for debian-linux-
2 | gnu (x86_64)
3 | --
4 | -- Host: localhost      Database: Syslog
5 | -----
6 | -- Server version     10.1.45-MariaDB-0+deb9u1
7 | 
8 | /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
9 |   */;
10 | /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
11 |   */;
12 | /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
13 |   */;
14 | /*!40101 SET NAMES utf8mb4 */;
15 | /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
16 | /*!40103 SET TIME_ZONE='+00:00' */;
17 | /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0
18 |   */;
19 | /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
20 |   FOREIGN_KEY_CHECKS=0 */;
21 | /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='
```

```
16 | NO_AUTO_VALUE_ON_ZERO' *';  
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

Para realizar o backup da tabela **SystemEvents**, utilize a ferramenta **mysqldump**:

```
1 |mysqldump Syslog SystemEvents > backup-tabela-system-events.sql
```

Verifique se o arquivo **backup-tabela-system-events.sql** foi criado. Visualize o conteúdo do DUMP através do comando **cat**:

```
1 |cat backup-tabela-system-events.sql
```

```
1 |-- MySQL dump 10.16 Distrib 10.1.45-MariaDB, for debian-linux-gnu (x86_64)  
2 |--  
3 |-- Host: localhost      Database: Syslog  
4 |-----  
5 |-- Server version      10.1.45-MariaDB-0+deb9u1  
6 |  
7 |/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT  
8 |*/;  
9 |/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS  
10| */;  
11|/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION  
12| */;  
13|/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0  
14| */;  
15|/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
16| FOREIGN_KEY_CHECKS=0 */;  
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='  
16| NO_AUTO_VALUE_ON_ZERO' */;  
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;  
17 |  
18 |--  
19 |-- Table structure for table `SystemEvents`  
20 |--  
21 |  
22 |DROP TABLE IF EXISTS `SystemEvents`;  
23 |/*!40101 SET @saved_cs_client      = @@character_set_client */;  
24 |/*!40101 SET character_set_client = utf8 */;
```

```
25 | CREATE TABLE `SystemEvents` (
26 |   `ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
27 |   `CustomerID` bigint(20) DEFAULT NULL,
28 |   `ReceivedAt` datetime DEFAULT NULL,
29 |   `DeviceReportedTime` datetime DEFAULT NULL,
30 |   `Facility` smallint(6) DEFAULT NULL,
31 |   `Priority` smallint(6) DEFAULT NULL,
32 |   `FromHost` varchar(60) DEFAULT NULL,
33 |   `Message` text,
34 |   `NTSeverity` int(11) DEFAULT NULL,
35 |   `Importance` int(11) DEFAULT NULL,
36 |   `EventSource` varchar(60) DEFAULT NULL,
37 |   `EventUser` varchar(60) DEFAULT NULL,
38 |   `EventCategory` int(11) DEFAULT NULL,
39 |   `EventID` int(11) DEFAULT NULL,
40 |   `EventBinaryData` text,
41 |   `MaxAvailable` int(11) DEFAULT NULL,
42 |   `CurrUsage` int(11) DEFAULT NULL,
43 |   `MinUsage` int(11) DEFAULT NULL,
44 |   `MaxUsage` int(11) DEFAULT NULL,
45 |   `InfoUnitID` int(11) DEFAULT NULL,
46 |   `SysLogTag` varchar(60) DEFAULT NULL,
47 |   `EventLogType` varchar(60) DEFAULT NULL,
48 |   `GenericFileName` varchar(60) DEFAULT NULL,
49 |   `SystemID` int(11) DEFAULT NULL,
50 |   PRIMARY KEY (`ID`)
51 | ) ENGINE=InnoDB AUTO_INCREMENT=697 DEFAULT CHARSET=utf8mb4;
52 | /*!40101 SET character_set_client = @saved_cs_client */;
```

Para realizar o backup da tabela **SystemEventsProperties**, utilize a ferramenta **mysqldump**:

```
1 | mysqldump Syslog SystemEventsProperties > backup-tabela-system-
    events-properties.sql
```

Verifique se o arquivo **backup-tabela-system-events-properties.sql** foi criado. Visualize o conteúdo do DUMP através do comando **cat**:

```
1 | cat backup-tabela-system-events-properties.sql
```

```
1 | -- MySQL dump 10.16 Distrib 10.1.45-MariaDB, for debian-linux-
    gnu (x86_64)
2 |
3 | -- Host: localhost      Database: Syslog
```

```
4 | -- -----
5 | -- Server version      10.1.45-MariaDB-0+deb9u1
6 |
7 | /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
8 |      */;
9 | /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
10 |      */;
11 | /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
12 |      */;
13 | /*!40101 SET NAMES utf8mb4 */;
14 | /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
15 | /*!40103 SET TIME_ZONE='+00:00' */;
16 | /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0
17 |      */;
18 | /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
19 |      FOREIGN_KEY_CHECKS=0 */;
20 | /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE=
21 |      'NO_AUTO_VALUE_ON_ZERO' */;
22 | /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

23 | --
24 | -- Table structure for table `SystemEventsProperties`
25 | --
26 |
27 | DROP TABLE IF EXISTS `SystemEventsProperties`;
28 | /*!40101 SET @saved_cs_client      = @@character_set_client */;
29 | /*!40101 SET character_set_client = utf8 */;
30 | CREATE TABLE `SystemEventsProperties` (
31 |     `ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
32 |     `SystemEventID` int(11) DEFAULT NULL,
33 |     `ParamName` varchar(255) DEFAULT NULL,
34 |     `ParamValue` text,
35 |     PRIMARY KEY (`ID`)
36 | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
37 | /*!40101 SET character_set_client = @saved_cs_client */;
```

Restore dos dados

Antes de restaurar os dados, remova a tabela **SystemEvents**:

```
1 | mysql -u root -D Syslog -e 'DROP TABLE SystemEvents;'
```

Confirme se a tabela **SystemEvents** foi removida:

```
1 |mysql -u root -D Syslog -e 'SHOW TABLES;'
```

Utilize o comando **mysql** para importar o backup da tabela **SystemEvents** no banco Syslog:

```
1 |mysql -u root -D Syslog < backup-tabela-system-events.sql
```

Confirme se a tabela **SystemEvents** foi restaurada:

```
1 |mysql -u root -D Syslog -e 'SHOW TABLES;'  
2 +-----+  
3 | Tables_in_Syslog |  
4 +-----+  
5 | SystemEvents |  
6 | SystemEventsProperties |  
7 +-----+
```

Visualize os dados da tabela nos campos **ID**, **fromHost** e **Message**:

```
1 |mysql -u root -D Syslog -e 'SELECT ID,fromHost,Message FROM  
SystemEvents';
```

```
1 +-----+-----+  
2 | ID   | fromHost | Message  
3 |       |          |  
4 +-----+-----+  
4 | 1   | graylog  | Stopping System Logging Service...  
5 |     |          |  
5 | 2   | graylog  | [origin software="rsyslogd" swVersion  
|      |          | = "8.24.0" x-pid="4292" x-info="http://www.rsyslog.com"]  
|      |          | exiting on signal 15.  
6 |     |          |  
6 | 3   | graylog  | Stopped System Logging Service.  
7 |     |          |  
7 | 4   | graylog  | Starting System Logging Service...  
8 |     |          |
```

```
8 | 5 | graylog | node=graylog type=SERVICE_START msg=audit
(1598625468.312:7817): pid=1 uid=0 auid=4294967295 ses
=4294967295 msg='unit=rsyslog comm="systemd" exe="/lib/
systemd/systemd" hostname=? addr=? terminal=? res=success'
|
9 | 6 | graylog | node=graylog type=SERVICE_STOP msg=audit
(1598625468.312:7818): pid=1 uid=0 auid=4294967295 ses
=4294967295 msg='unit=rsyslog comm="systemd" exe="/lib/
systemd/systemd" hostname=? addr=? terminal=? res=success'
|
10 | 7 | graylog | node=graylog type=SYSCALL msg=audit
(1598625468.312:7819): arch=c000003e syscall=59 success=yes
exit=0 a0=7f386f107cc0 a1=7ffc2a613c80 a2=7ffc2a614228 a3=7
ffc2a613ee8 items=2 ppid=4722 pid=4723 auid=1000 uid=0 gid=0
euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=1
comm="systemd-tty-ask" exe="/bin/systemd-tty-ask-password-
agent" key=(null) |
11 | 8 | graylog | node=graylog type=EXECVE msg=audit
(1598625468.312:7819): argc=2 a0="/bin/systemd-tty-ask-
password-agent" a1="--watch"
|
12 | 9 | graylog | node=graylog type=CWD msg=audit
(1598625468.312:7819): cwd="/root"
```

Agendamento de backup

Vamos criar um script de backup para o banco **Syslog**:

```
1 | vim /usr/local/bin/bkp-banco.sh
2 |#!/bin/bash
3 |mysqldump --user="rsysloguser" --password="rsyslogpw" "$@" "
Syslog" > "/opt/backup/syslog-$(date '+%d-%m-%Y').sql 2> /
dev/null
```

Defina permissão de acesso ao script **/usr/local/bin/bkp-banco.sh**:

```
1 | chmod u+x /usr/local/bin/bkp-banco.sh
```

Antes de testar o script, crie a pasta backup no diretório **/opt**:

```
1 |mkdir /opt/backup
```

Execute o script **/usr/local/bin/bkp-banco.sh** para testar o backup:

```
1 |bash /usr/local/bin/bkp-banco.sh
```

Verifique se o script criou o arquivo de nome syslog com o dia, mês e ano:

```
1 |ls -l /opt/backup/
2 |total 36
3 |-rw-r--r-- 1 root root 33242 Nov 11 21:52 syslog-11-11-2020.sql
```

Para criar um agendamento de backups diários, copie o script **bkp-banco.sh** para o diretório **/etc/cron.daily/**.

```
1 |cp /usr/local/bin/bkp-banco.sh /etc/cron.daily/bkp-banco
```

Antes de testar o agendamento, remova todos os backups criados no diretório **/opt/backup**:

```
1 |rm /opt/backup/*
```

Execute o comando **run-parts** para força a execução de todos os scripts diários:

```
1 |run-parts /etc/cron.daily
```

Verifique se o agendamento diário criou o arquivo de nome syslog com o dia, mês e ano:

```
1 |ls -l /opt/backup/
2 |total 36
3 |-rw-r--r-- 1 root root 33242 Nov 11 21:53 syslog-11-11-2020.sql
```

4

Centralização de logs com Graylog

Competências deste conteúdo

- Instalação e configuração do Graylog (Graylog + Elastic + MongoDB)
- Configuração e entendimento sobre Inputs
- Realização de buscas
- Criação de extractores baseados em Grok Pattern e Regex
- Criação de dashboards
- Criação de alertas personalizados

Instalação e configuração do Graylog (Graylog + Elastic + Mon

Fig. 4.1: Graylog

Graylog

Basicamente, o Graylog é um software para gerência de logs. O projeto foi iniciado em 2009 por Lennart Koopmann que após receber a cotação de um software para gestão de logs proprietário com custo elevado, decidiu desenvolver sua própria ferramenta. O Graylog funciona em conjunto com os softwares MongoDB e Elasticsearch, viabilizando a criação de filtros e aumentando a velocidade de pesquisa de mensagens.

Dentre suas características, destacam-se:

- Coleção de logs em diferentes formatos
- Dashboards personalizados para os logs
- Filtros em registros
- Gerador de alertas

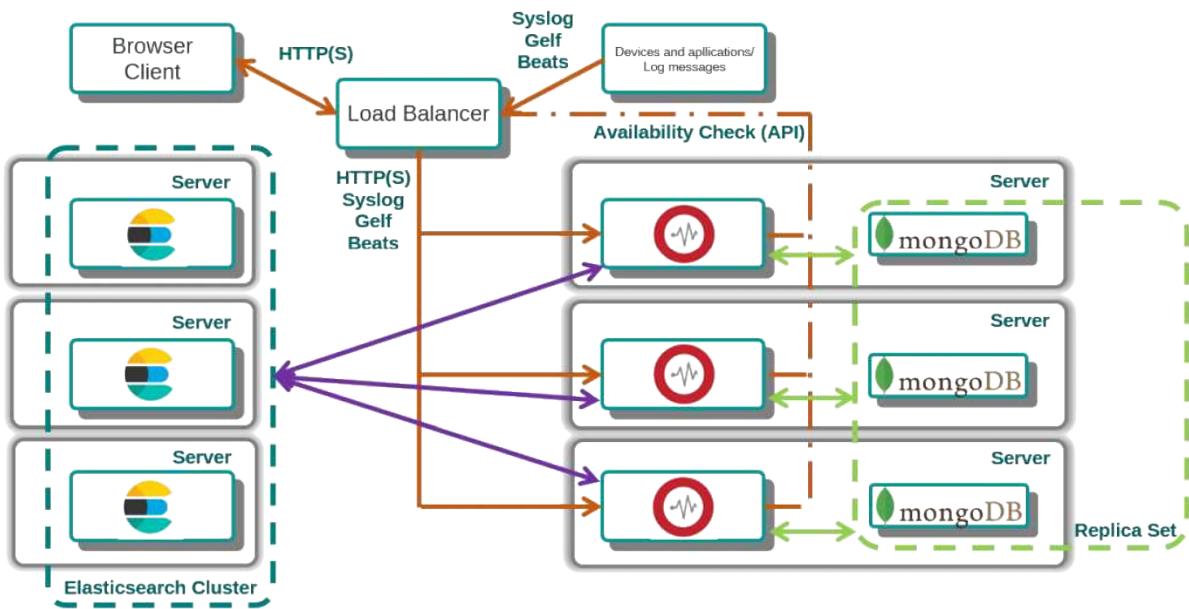


Fig. 4.2: Estrutura do Graylog

Na imagem acima, podemos ver um exemplo de estrutura que pode ser utilizado para instalar o Graylog em grandes ambientes. Como pode ser visto, o Graylog é composto por três ferramentas, o **Graylog-server**, **MongoDB** para armazenamento de configurações do Graylog-server e **Elasticsearch** para armazenamento dos logs.

LAB 4.1 - Instalação e configuração do Graylog

Neste laboratório vamos aprender como instalar e configurar o Graylog.

Acesse a VM **graylog-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.12
```

Para realizar a instalação dos pré-requisitos, execute os comandos:

```
1 | sudo apt update
2 | sudo apt install -y apt-transport-https openjdk-8-jre-headless
      uuid-runtime pwgen
```

Em seguida configure o **JAVA_HOME** que sera utilizado pelo Elasticsearch:

```
1 | sudo vim /etc/environment +$
2 | PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
```

```
3 | JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/"
```

Recarregue o arquivo para aplicar as mudanças em sua sessão atual e teste a variável **JAVA_HOME**:

```
1 | source /etc/environment
```

```
1 | echo $JAVA_HOME  
2 | /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/
```

Vamos instalar o **MongoDB**:

1. Adicione a chave do repositório;
2. Adicione o repositório do MongoDB no arquivo `mongodb-org-4.0.list`;
3. Atualize a lista de pacotes;
4. Instale o pacote `mongodb-org`.

```
1 | sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --  
    recv 9DA31620334BD75D9DCB49F368818C72E52529D4  
2 |  
3 | echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu  
      bionic/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/  
      sources.list.d/mongodb-org-4.0.list  
4 |  
5 | sudo apt update  
6 | sudo apt-get install -y mongodb-org
```

Habilite o MongoDB na inicialização e inicie o serviço:

```
1 | sudo systemctl enable mongod.service  
2 | sudo systemctl start mongod.service  
3 | sudo systemctl status mongod.service
```

```
1 | * mongod.service - MongoDB Database Server  
2 |   Loaded: loaded (/lib/systemd/system/mongod.service; enabled;  
        vendor preset: enabled)  
3 |     Active: active (running) since Wed 2020-11-11 22:15:03 UTC; 1  
        s ago  
4 |       Docs: https://docs.mongodb.org/manual  
5 |     Main PID: 8434 (mongod)  
6 |       CGroup: /system.slice/mongod.service  
7 |             `-- 8434 /usr/bin/mongod --config /etc/mongod.conf
```

```
8 |  
9 | Nov 11 22:15:03 graylog systemd[1]: Started MongoDB Database  
   Server.
```

Em seguida pare e remova da inicialização o serviço do MySQL:

```
1 | sudo systemctl stop mysql.service  
2 | sudo systemctl disable mysql.service
```

Agora vamos realizar a instalação do Elasticsearch:

1. Adicione a chave do repositório;
2. Adicione o repositório do Elasticsearch no arquivo **elastic-6.x.list**;
3. Atualize a lista de pacotes;
4. Instale o pacote `elasticsearch-oss`.

```
1 | wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |  
   sudo apt-key add -  
2 |  
3 | echo "deb https://artifacts.elastic.co/packages/oss-6.x/apt  
      stable main" | sudo tee -a /etc/apt/sources.list.d/elastic  
      -6.x.list  
4 |  
5 | sudo apt-get update  
6 | sudo apt-get install elasticsearch-oss -y
```

Vamos trocar o nome do cluster do Elasticsearch para Graylog. Para isso, abra para edição o arquivo de configuração do Elasticsearch:

```
1 | sudo vim /etc/elasticsearch/elasticsearch.yml
```

E deixe a linha **cluster.name** da seguinte maneira:

```
1 | cluster.name: graylog
```

Altere para 512m o tamanho inicial e máximo do espaço total de heap:

```
1 | sudo vim /etc/elasticsearch/jvm.options  
2 | ....  
3 |  
4 | -Xms512m
```

```
5 | -Xmx512m
```

Habilite o Elasticsearch na inicialização e inicie o serviço:

```
1 | sudo systemctl enable elasticsearch.service
2 | sudo systemctl start elasticsearch.service
3 | sudo systemctl status elasticsearch.service
```

```
1 | * elasticsearch.service - Elasticsearch
2 |   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service
2 |           ; enabled; vendor preset: enabled)
3 |     Active: active (running) since Wed 2020-11-11 22:19:35 UTC;
3 |             939ms ago
4 |       Docs: http://www.elastic.co
5 |     Main PID: 9417 (java)
6 |       Tasks: 13 (limit: 2945)
7 |      CGroup: /system.slice/elasticsearch.service
7 |             `--9417 /usr/bin/java -Xms512m -Xmx512m -XX:+
7 |                   UseConcMarkSweepGC -XX:
7 |                   CMSInitiatingOccupancyFraction=75 -XX:+UseC
9 |
10 | Nov 11 22:19:35 graylog systemd[1]: Started Elasticsearch.
11 | Nov 11 22:19:35 graylog elasticsearch[9417]: warning: Falling
11 | back to java on path. This behavior is deprecated. Specify
```

E por fim, vamos instalar o Graylog:

1. Baixe o arquivo .deb do repositório;
2. Instale o repositório do Graylog através do arquivo .deb;
3. Atualize a lista de pacotes;
4. Instale o pacote graylog-server.

```
1 | wget https://packages.graylog2.org/repo/packages/graylog-3.3-
1 |   repository_latest.deb
2 |
3 | sudo dpkg -i graylog-3.3-repository_latest.deb
4 |
5 | sudo apt-get update
6 | sudo apt-get install graylog-server -y
```

Habilite o Graylog na inicialização e inicie o serviço:

```
1 | sudo systemctl enable graylog-server.service
2 | sudo systemctl start graylog-server.service
3 | sudo systemctl status graylog-server.service

1 | * graylog-server.service - Graylog server
2 |   Loaded: loaded (/usr/lib/systemd/system/graylog-server.
|     service; enabled; vendor preset: enabled)
3 |   Active: active (running) since Wed 2020-11-11 22:22:32 UTC;
|     884ms ago
4 |     Docs: http://docs.graylog.org/
5 |   Main PID: 10216 (graylog-server)
6 |     Tasks: 14 (limit: 2945)
7 |   CGroup: /system.slice/graylog-server.service
8 |         |-10216 /bin/sh /usr/share/graylog-server/bin/graylog
|             -server
9 |         `-10228 /usr/bin/java -Xms1g -Xmx1g -XX:NewRatio=1 -
|             server -XX:+ResizeTLAB -XX:+UseConcMarkSweepGC -
|             XX:+CMSCon
10 |
11 | Nov 11 22:22:32 graylog systemd[1]: Started Graylog server.
```

Crie as senhas para autenticação no Graylog Server, para isso, execute o comando abaixo:

```
1 | sudo pwgen -N 1 -s 96
2 | EWYD0QKzY1X9WAKXGmSR6lAEPtI728RXy2IPDvJmj01zdgtNlC0Pr9fh09zAeHU6S5D9x00dowG
```

```
1 | echo -n 4linux | sha256sum
2 | 212e018e6377125c10c19d912a5bb537898ccd86d9e847131de1695330f69f7d
```

OBS: **Guarde esses valores** pois iremos utiliza-los no próximo passo.

No arquivo de configuração do Graylog, altere os valores dos parâmetros no arquivo **/etc/graylog/server/server.conf**:

```
1 | sudo vim /etc/graylog/server/server.conf
```

```
1 | password_secret =
|   EWYD0QKzY1X9WAKXGmSR6lAEPtI728RXy2IPDvJmj01zdgtNlC0Pr9fh09zAeHU6S5D9x00
```

```
2
3 root_password_sha2 = 212
e018e6377125c10c19d912a5bb537898ccd86d9e847131de1695330f69f7d
4
5 root_timezone = America/Sao_Paulo
6
7 http_bind_address = 172.16.0.12:9000
8
9 http_publish_uri = http://172.16.0.12:9000/
10
11 http_external_uri = http://172.16.0.12:9000/
```

Parâmetros alterados no arquivo de configuração do Graylog:

Parâmetro	Descrição
password_secret	Colar o resultado do comando pwgen -N 1 -s 96
root_password_sha2	Colar o resultado do comando echo -n 4linux
root_timezone	Configuração de fuso horário do usuário root
http_bind_address	Interface de rede usada pela interface HTTP do Graylog.
http_publish_uri	Usado para se comunicar com os outros nós do Graylog no cluster e por todos os clientes que usam a interface web do Graylog.
http_external_uri	Usado pela interface web do Graylog para se comunicar com a API REST do Graylog, interface da web do Graylog.

Agora que todas as configurações foram realizadas, vamos reiniciar os serviços do Graylog e Elasticsearch:

```
1 sudo systemctl restart graylog-server
2 sudo systemctl status graylog-server
3 sudo systemctl restart elasticsearch.service
4 sudo systemctl status elasticsearch.service
```

Pronto! Agora que os serviços estão configurados e foram iniciados, podemos acessar a interface gráfica do Graylog pela primeira vez. Acesse:

<http://172.16.0.12:9000/>

Username: admin

Password: 4linux

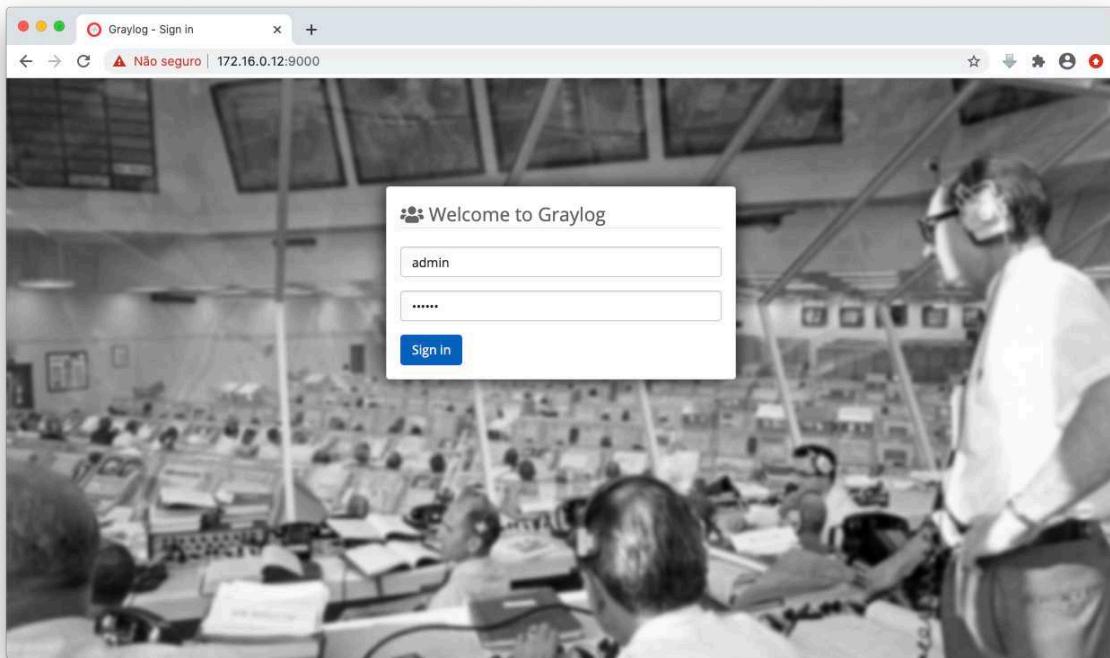


Fig. 4.3: Tela de login do Graylog

Realizado o login, podemos ver a tela inicial do Graylog:

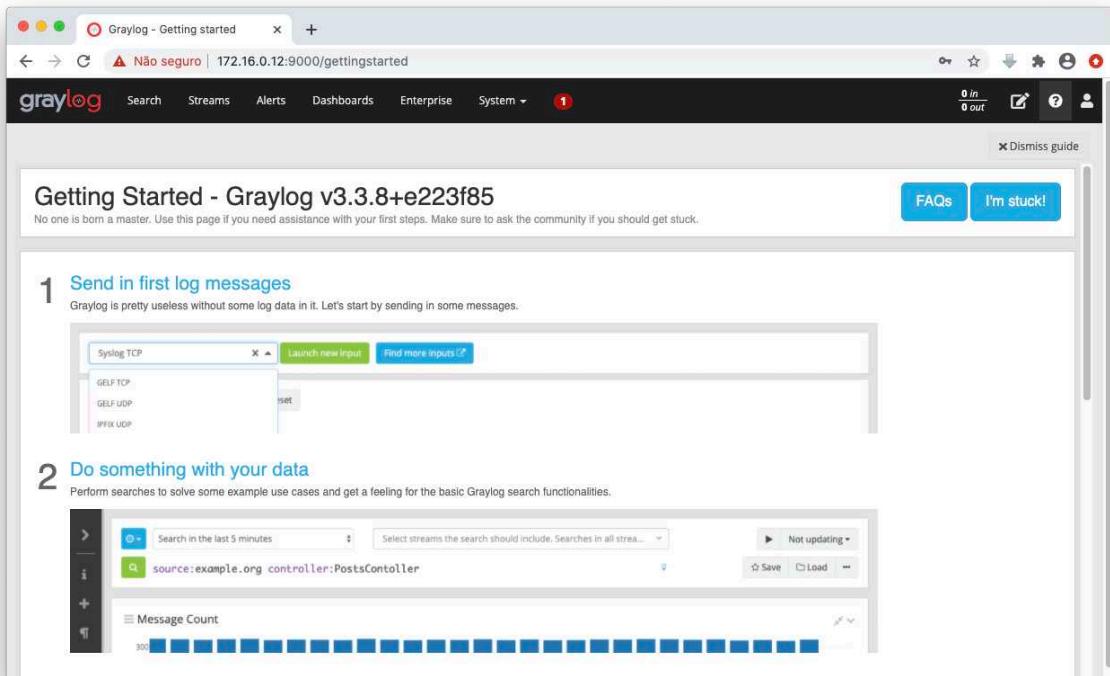


Fig. 4.4: Tela inicial do Graylog

Configurar e entender os Inputs

Os Inputs de mensagens são responsáveis por aceitar mensagens de log. Depois de escolher o tipo de entrada na interface da web Graylog em System -> Input, a entrada é iniciada sem reiniciar o Graylog. A maioria dos ambientes usará os padrões para as entradas, mas a maioria das entradas tem uma configuração granular disponível.

Fonte: https://docs.graylog.org/en/3.3/pages/sending_data.html

Inputs disponíveis no Graylog

Input	Descrição
AWS CloudTrail Input	Permite registrar logs de como sua conta da AWS está sendo usada, em todas as atividades do console da web, SDKs e APIs
Beats	Permite registrar logs do Elastic através de beats.
CEF AMQP Input	Permite registrar logs do CEF (Common Event Format) usando o AMQP como sistema de enfileiramento.
CEF Kafka Input	Permite registrar logs do CEF (Common Event Format) usando o Kafka como sistema de enfileiramento.
CEF TCP Input	Permite registrar logs do CEF (Common Event Format) TCP.
CEF UDP Input	Permite registrar logs do CEF (Common Event Format) UDP.
GELF AMQP	Permite registrar logs do GELF (Graylog Extended Log Format) usando o AMQP como sistema de enfileiramento.
GELF HTTP	Permite registrar logs do GELF (Graylog Extended Log Format) HTTP.
GELF Kafka	Permite registrar logs do GELF (Graylog Extended Log Format) usando o Kafka como sistema de enfileiramento.
GELF TCP	Permite registrar logs do GELF (Graylog Extended Log Format) TCP.
GELF UDP	Permite registrar logs do GELF (Graylog Extended Log Format) UDP.
JSON path from HTTP API	Permite registrar resposta JSON de um recurso REST e armazena um valor de campo dele como um log no Graylog.
NetFlow UDP	Permite registrar fluxo de rede como um log no Graylog.
Random HTTP message generator	Permite registrar mensagens HTTP aleatórias no Graylog.
Raw/Plaintext AMQP	Permite registrar logs de mensagens de texto simples, usando o AMQP como sistema de enfileiramento.

Input	Descrição
Raw/Plaintext Kafka	Permite registrar logs de mensagens de texto simples, usando o Kafka como sistema de enfileiramento.
Raw/Plaintext TCP	Permite registrar logs de mensagens de texto simples TCP.
Raw/Plaintext UDP	Permite registrar logs de mensagens de texto simples UDP.
Syslog AMQP	Permite registrar logs do Syslog, usando o AMQP como sistema de enfileiramento.
Syslog Kafka	Permite registrar logs do Syslog, usando o Kafka como sistema de enfileiramento.
Syslog TCP	Permite registrar logs do Syslog TCP.
Syslog UDP	Permite registrar logs do Syslog UDP.

LAB 4.2 - Coletar logs dos hosts pelo rsyslog

Vamos configurar o Graylog para receber logs do rsyslog.

Criando um input para rsyslog

Primeiramente precisamos criar um input para o rsyslog. Acesse **System > Inputs** em **Select input**

The screenshot shows the Graylog web interface with the following details:

- Header:** Graylog - Inputs, 172.16.0.12:9000/system/inputs
- Top Bar:** Search, Streams, Alerts, Dashboards, Enterprise, System / Inputs (with a red notification dot), 0 in, 0 out.
- Inputs Section:**
 - Global inputs:** 0 configured. Message: "There are no global inputs."
 - Local inputs:** 0 configured. Message: "There are no local inputs."
- Bottom Footer:** Graylog 3.3.8+e223f85 on graylog (Private Build 1.8.0_272 on Linux 4.15.0-121-generic)

Fig. 4.5: Criando input Rsyslog - ETAPA 1

Em seguida selecione **Syslog UDP** e clique em **Launch new input**.

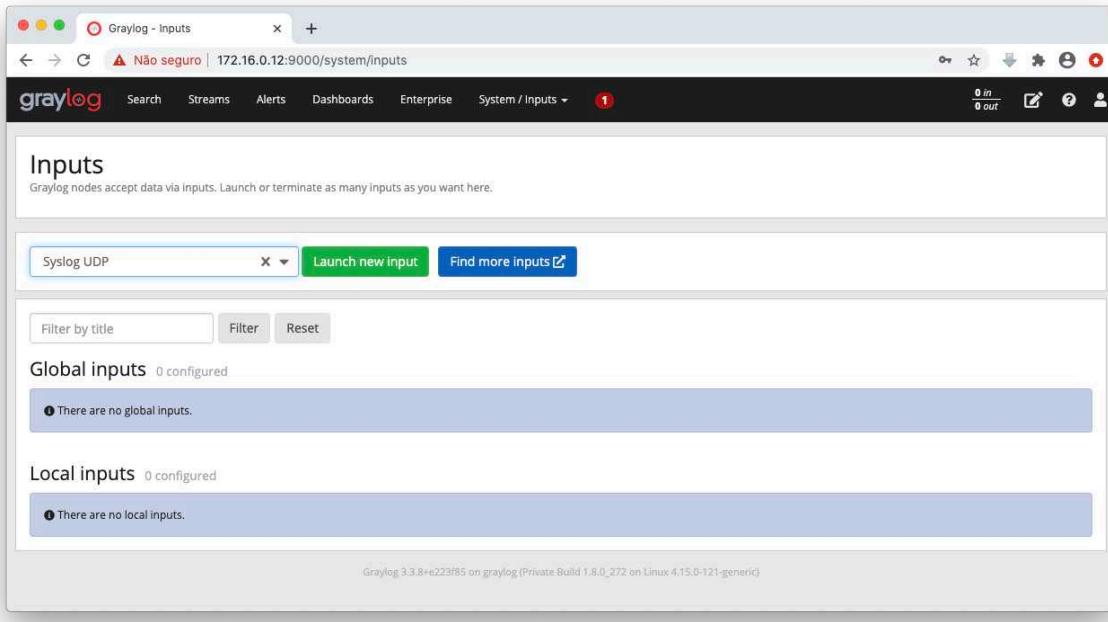


Fig. 4.6: Criando input Rsyslog - ETAPA 2

Preencha os seguintes campos:

Node: Selecione o node **Title:** SYSLOG **Bind address:** 172.16.0.12 **Port:** 1514

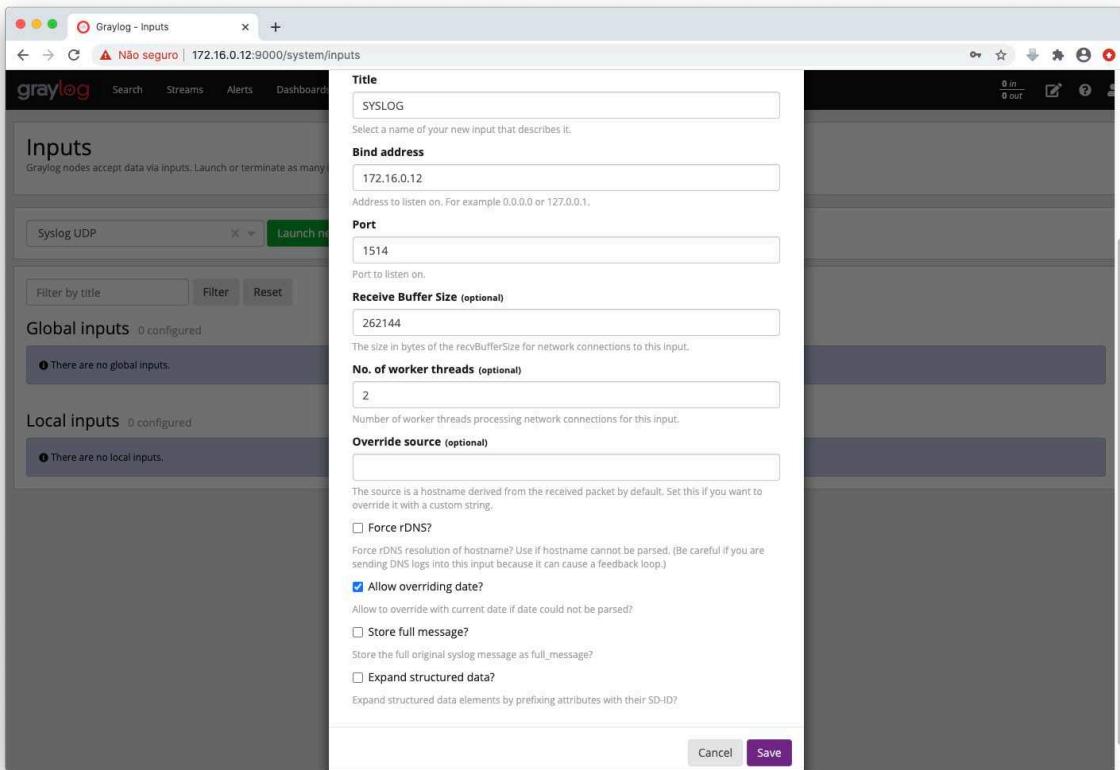


Fig. 4.7: Criando input Rsyslog - ETAPA 3

Feito o preenchimento, clique em **Save**.

Verifique se o status do Input mudou para **RUNNING**

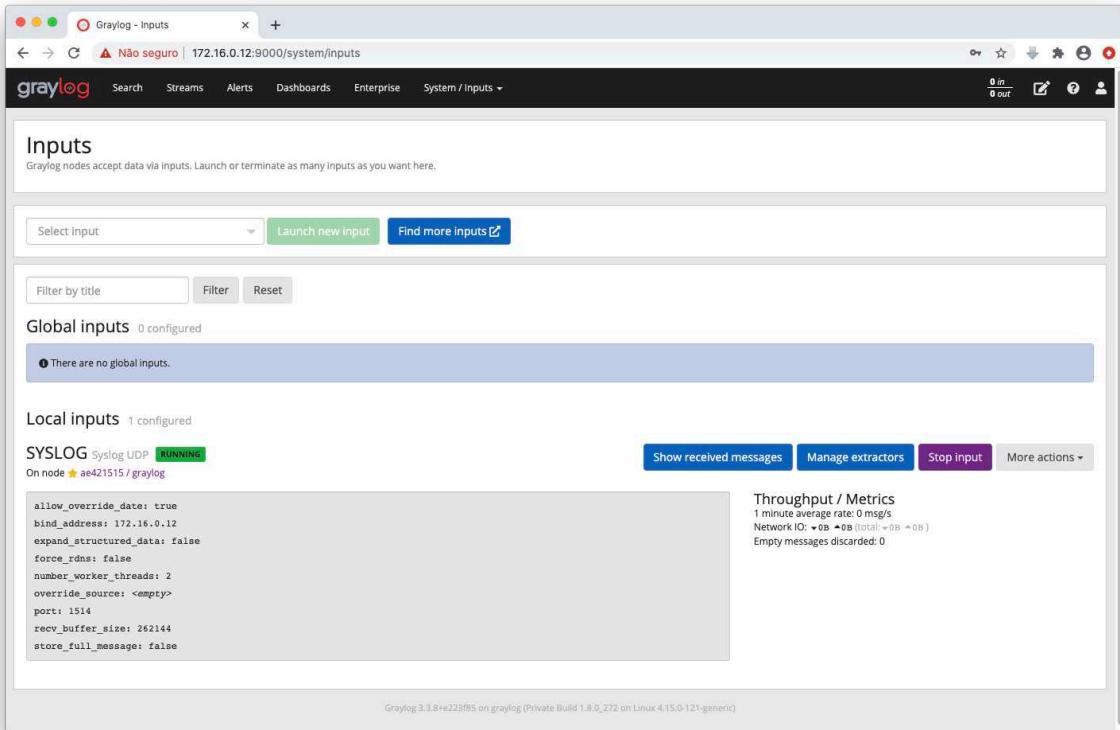


Fig. 4.8: Criando input Rsyslog - ETAPA 4

LAB 4.3 - Configuração de clientes

Acesse as VMs **webserver-audit** e **kibana-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.11
2 | ssh suporte@172.16.0.13
```

Nas 2 VMs habilite o envio de logs para o servidor do Graylog. Para isso, edite o arquivo **/etc/rsyslog.conf**:

```
1 | sudo vim /etc/rsyslog.conf
```

E adicione a seguinte linha no final do arquivo:

```
1 | *.* @graylog:1514;RSYSLOG_SyslogProtocol23Format
```

Após editar e salvar o arquivo, reinicie o serviço do rsyslog:

```
1 | sudo systemctl restart rsyslog
```

Volte para o ambiente gráfico do Graylog. Para verificar se as mensagens realmente estão chegando e conseguir ler essas mensagens, clique em **Search**.

Veja que o Graylog já está recebendo as mensagens de log dos servidores **webserver**:

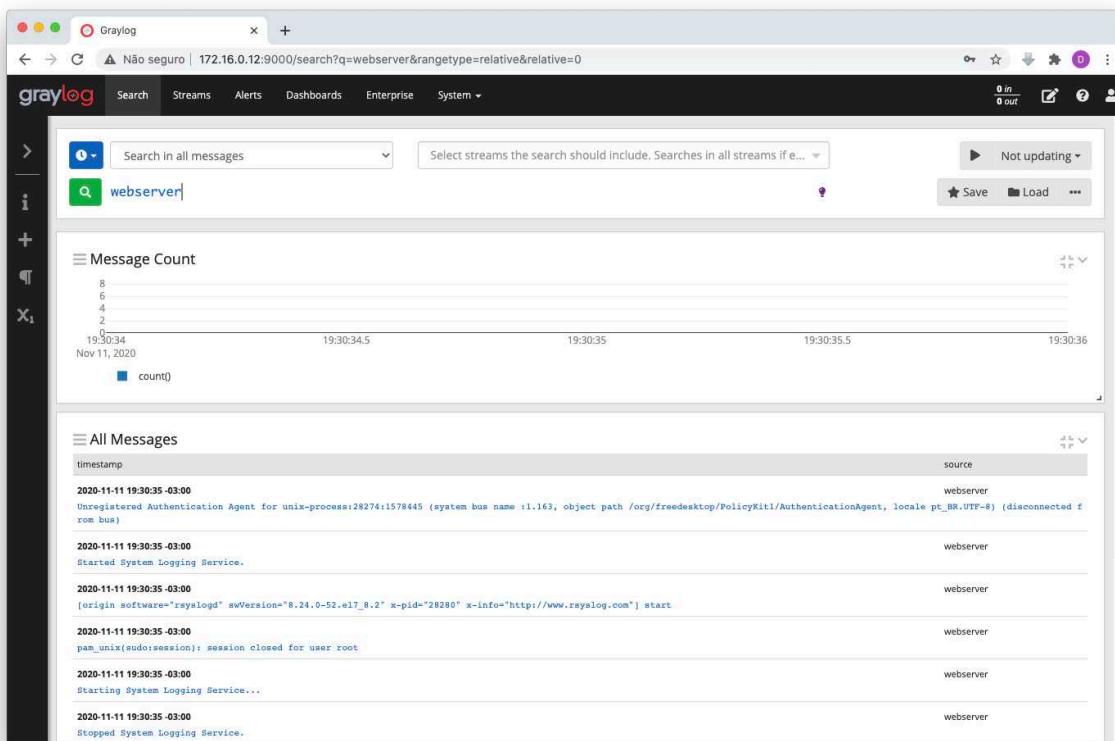


Fig. 4.9: Visualizar logs da VM webserver no Graylog

Veja que o Graylog já está recebendo as mensagens de log dos servidores **Kibana**:

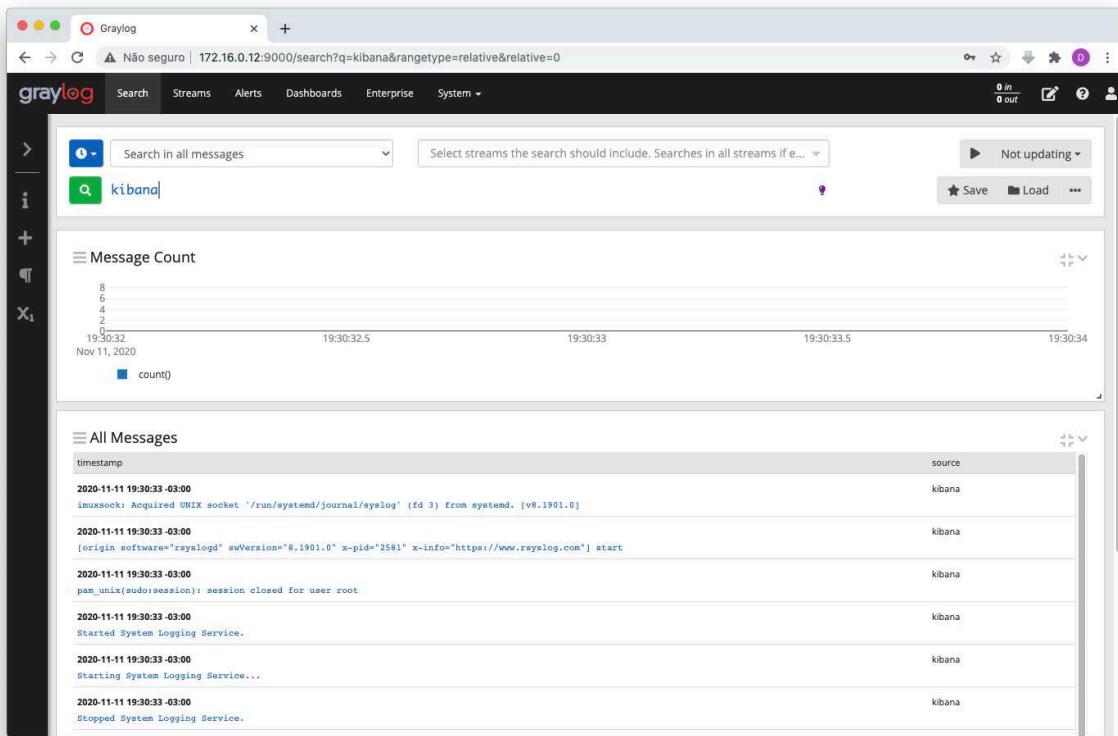


Fig. 4.10: Visualizar logs da VM webserver no Kibana

LAB 4.4 - Coletar logs de containers

Vamos configurar o Graylog para receber logs do Docker.

Configuração no Graylog

Primeiramente precisamos criar um input do tipo GELF, acesse **System > Inputs** em **Select input** selecione **GELF UDP** e clique em **Launch new input**, e preencha os seguintes campos:

Node: Selecione o node disponível **Title:** Docker **Bind address:** 172.16.0.12 **Port:** 12201

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

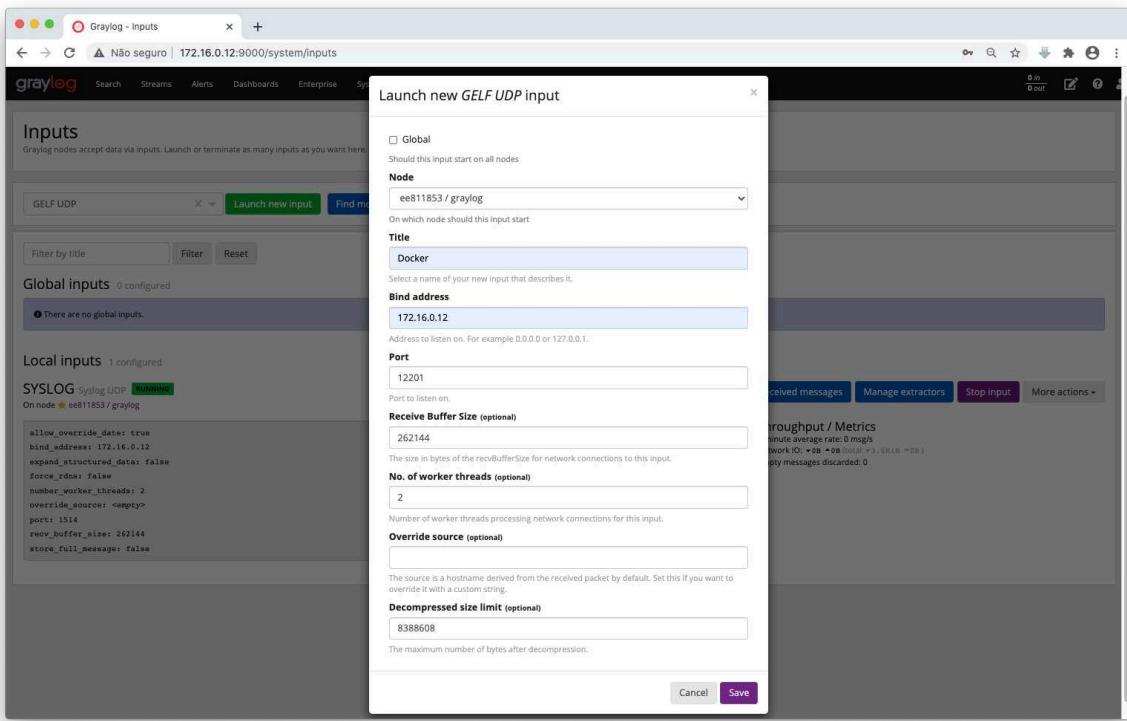


Fig. 4.11: Criando input Docker - ETAPA 1

Feito o preenchimento clique em **Save**.

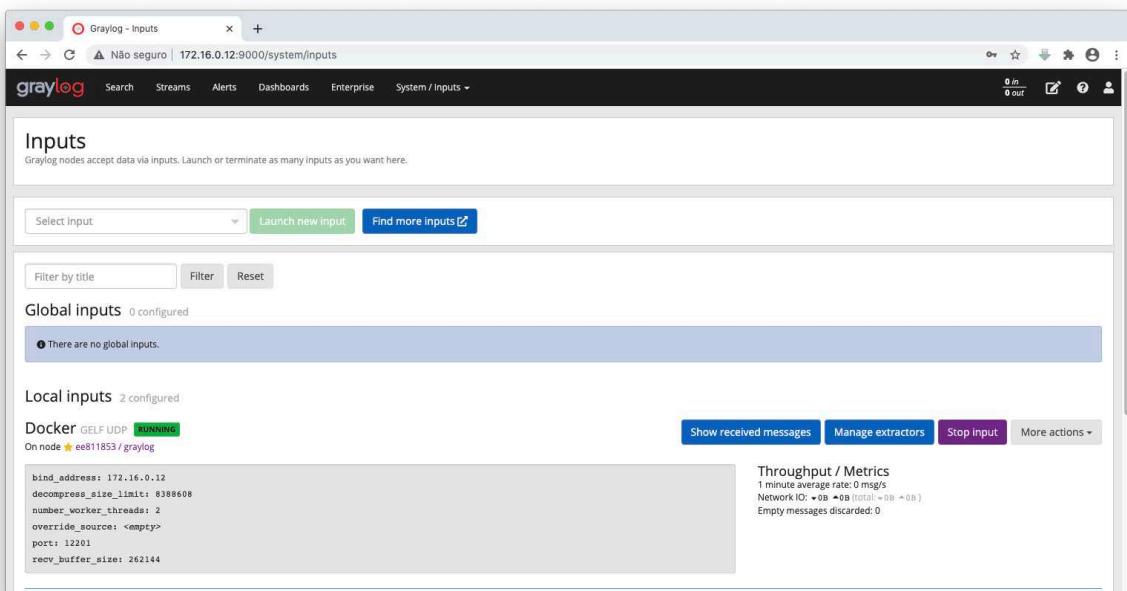


Fig. 4.12: Criando input Docker - ETAPA 2

Configuração no Docker

O Docker possui diversos drivers de logs e entre eles o GELF, que significa Graylog Extended Log Format. Para conhecer outros drivers de log, acesse a documentação oficial do Docker:

<https://docs.docker.com/config/containers/logging/configure/>

1 - Provisionamento do Wordpress

Acesse a VM **webserver-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.11
```

E visualize o arquivo **wordpress-deploy/wordpress-deploy.yaml** que provisiona o ambiente do Wordpress, através do Docker Compose.

```
1 | cat wordpress-deploy/docker-compose.yaml
```

```
1 | ....
2 |   wordpress-mysql:
3 |     image: "mysql:5.6"
4 |     logging:
5 |       driver: "gelf"
6 |       options:
7 |         gelf-address: "udp://172.16.0.12:12201"
8 |
9 | ....
10 |   wordpress:
11 |     image: "wordpress:4.8-apache"
12 |     logging:
13 |       driver: "gelf"
14 |       options:
15 |         gelf-address: "udp://172.16.0.12:12201"
```

Em seguida acesse a pasta **wordpress-deploy** e provisione o Wordpress, através do Docker Compose.

```
1 | cd wordpress-deploy
2 | docker-compose up -d
```

Verifique se os containers do **Traefik**, **MYSQL** e **Wordpress**, estão em execução:

```
1 | docker-compose ps
```

- Resultado:

1	Name	State	Command
2			Ports
3	wordpress-deploy_traefik_1	Up	/traefik --web --docker -- ... 0.0.0.0:443->443/tcp, 0.0.0.0:80->80/ tcp,
4			
5	wordpress-deploy_wordpress-mysql_1	Up	docker-entrypoint.sh mysqld 3306/tcp
6	wordpress-deploy_wordpress_1	Up	docker-entrypoint.sh apach 80/tcp

Acessar a aplicação Wordpress

Adicione uma entrada no arquivo **/etc/hosts** na máquina física apontando para o IP **172.16.0.11**, o domínio **wordpress.4labs.example**:

No Linux e Mac

- Arquivo **/etc/hosts**:

```
1 | ....  
2 | 172.16.0.11 wordpress.4labs.example
```

No Windows

- Arquivo: C:\Windows\System32\Drivers\etc\hosts

```
1 | ....  
2 | 172.16.0.11 wordpress.4labs.example
```

No navegador de sua máquina física acesse o domínio **wordpress.4labs.example**, e instale o Wordpress.

Acesso: <http://wordpress.4labs.example>

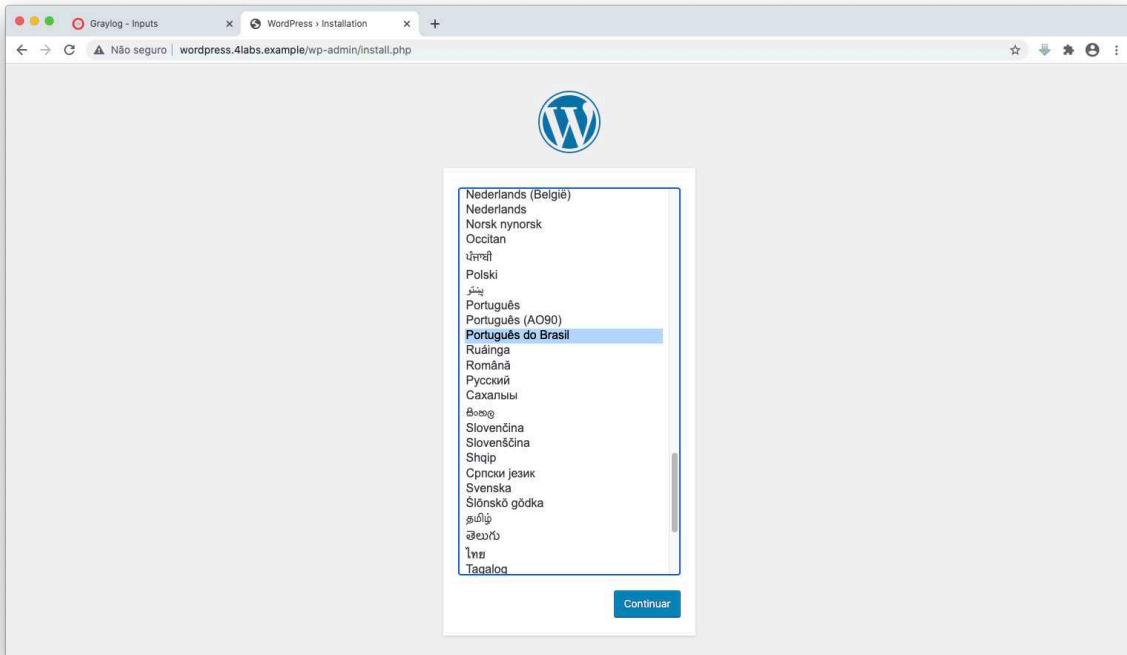


Fig. 4.13: Acessando Wordpress - ETAPA 1

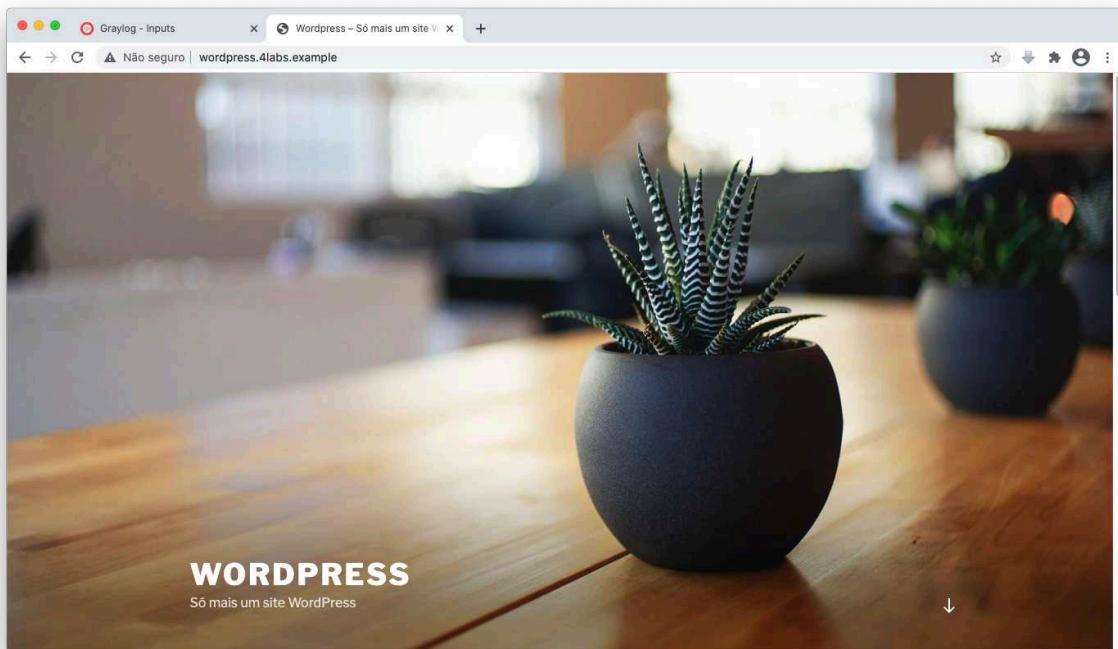


Fig. 4.14: Acessando Wordpress - ETAPA 2

Para acompanhar os logs do Docker, clique no botão **Show received messages**.

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

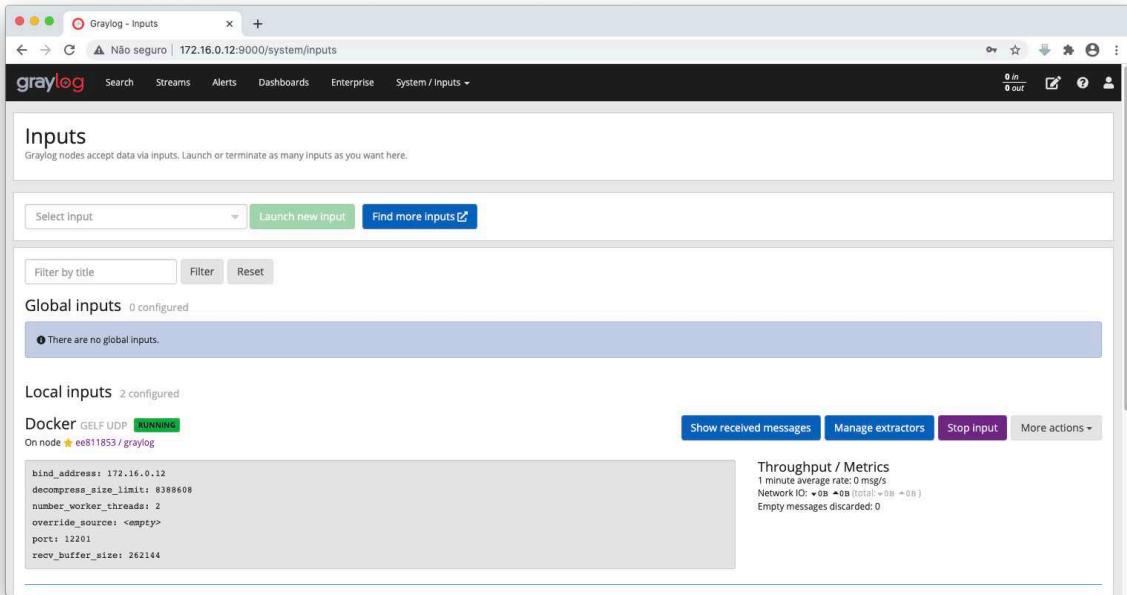


Fig. 4.15: Visualizar logs do Docker - ETAPA 1

Pesquise a string **wordpress** e veja que o Graylog já está recebendo as mensagens de log dos containers:

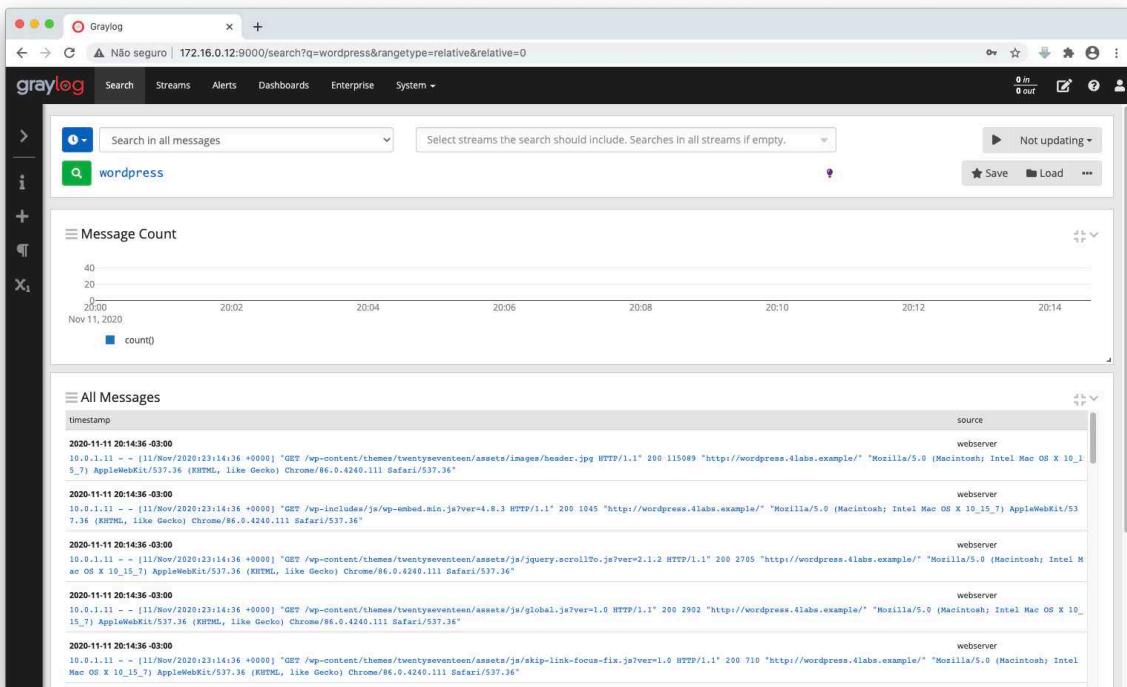


Fig. 4.16: Visualizar logs do Docker - ETAPA 2

Extratores no Graylog

Os extratores permitem que você instrua os nós Graylog sobre como extrair dados de qualquer texto na mensagem recebida (não importa de qual formato ou se um campo já foi extraído) para os campos de mensagem.

Um extrator é um filtro que pode ser construído dentro do Graylog, para extraír dados de qualquer texto na mensagem recebida, facilitando a criação de uma área de filtros.

LAB 4.5 - Coletar logs de containers

Vamos configurar um extrator para filtrar código de resposta HTTP.

Acesse a VM **kibana-audit** via **ssh** e faça um acesso à página do Wordpress via comando **curl**:

```
1 ssh suporte@172.16.0.13
2
3 for cont in $(seq 1 10); do curl http://wordpress.4labs.example;
done
```

Para criar um extrator acesse no Graylog **System > Inputs > Docker > Manage extractors**.

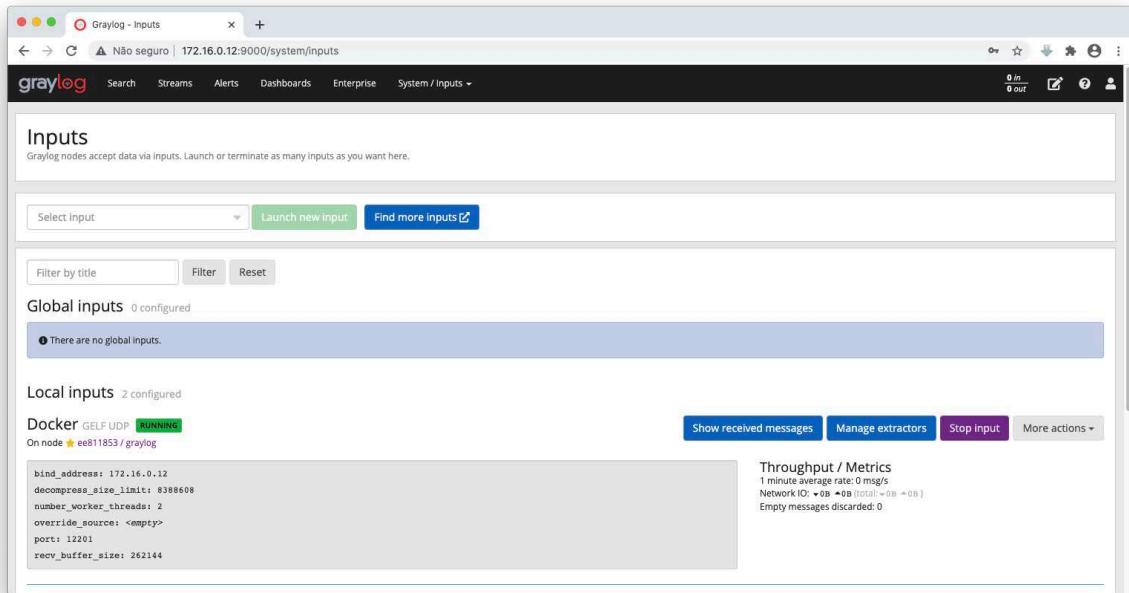


Fig. 4.17: Configurar Manage extractors - ETAPA 1

Em seguida clique em **Get started** e **Load Message**, será apresentado a última mensagem registrada do input do Docker.

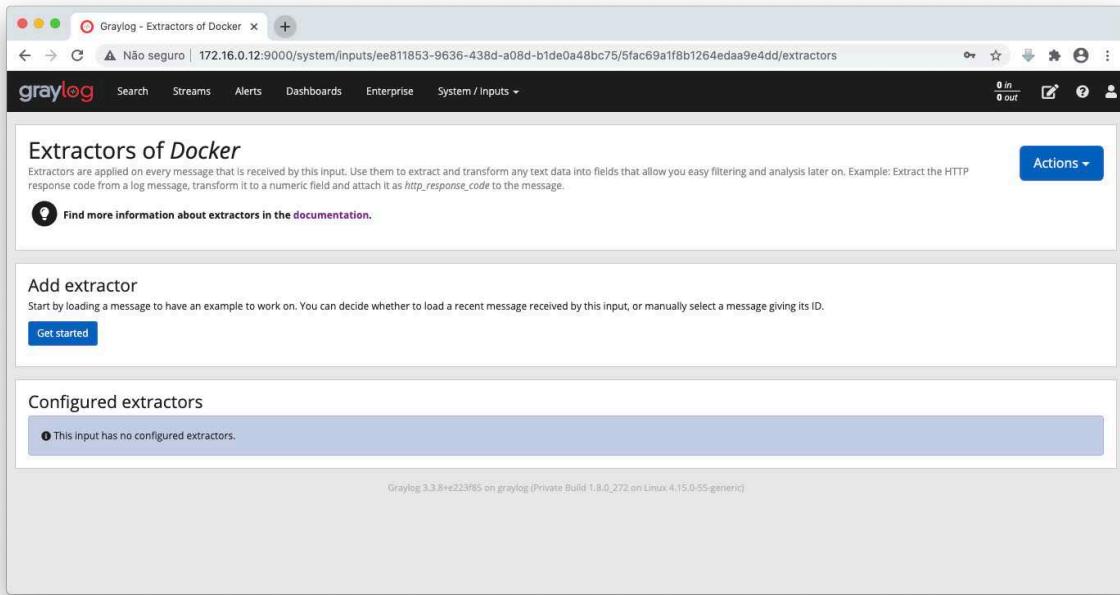


Fig. 4.18: Configurar Manage extractors - ETAPA 2

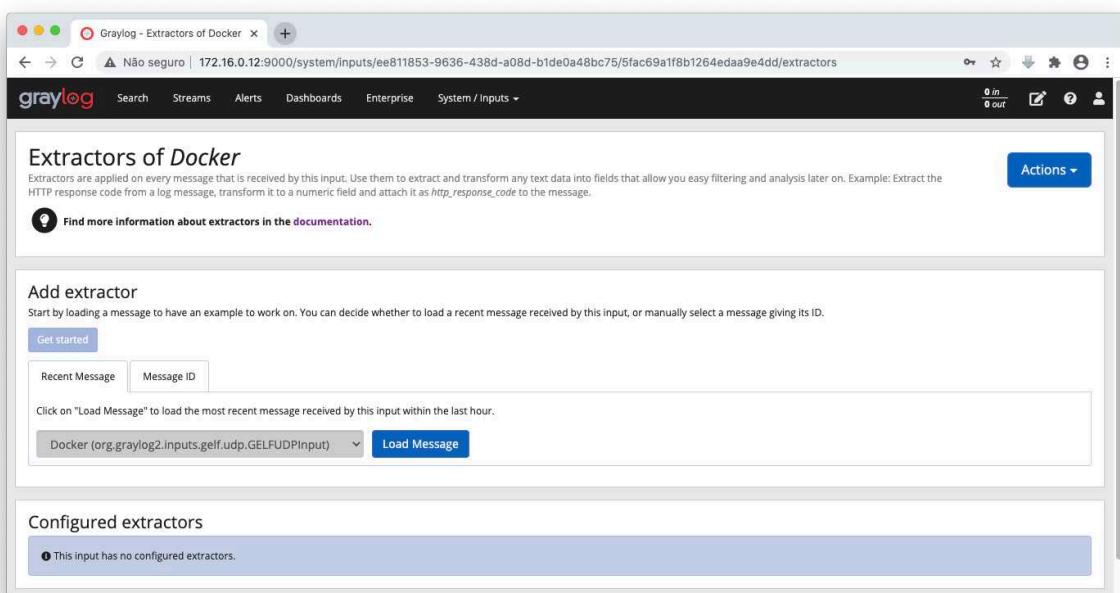


Fig. 4.19: Configurar Manage extractors - ETAPA 3

Selecione no campo **message** a opção Select extractor type -> **Regular expression**

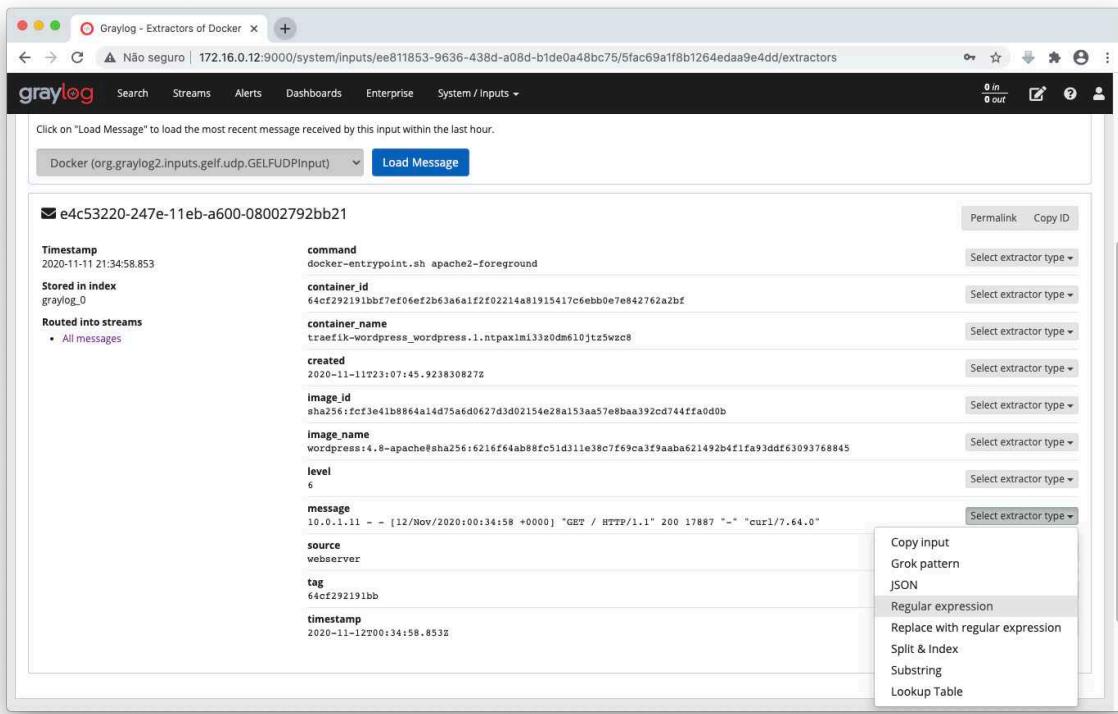


Fig. 4.20: Configurar Manage extractors - ETAPA 4

E preencha os seguintes campos:

Regular expression: `HTTP/1.1\" (.+?)\s.*` **Condition:** Always try to extract **Store as field:** http_reponse_code **Extraction strategy:** Copy **Extractor title:** HTTP_RESPONSE_CODE

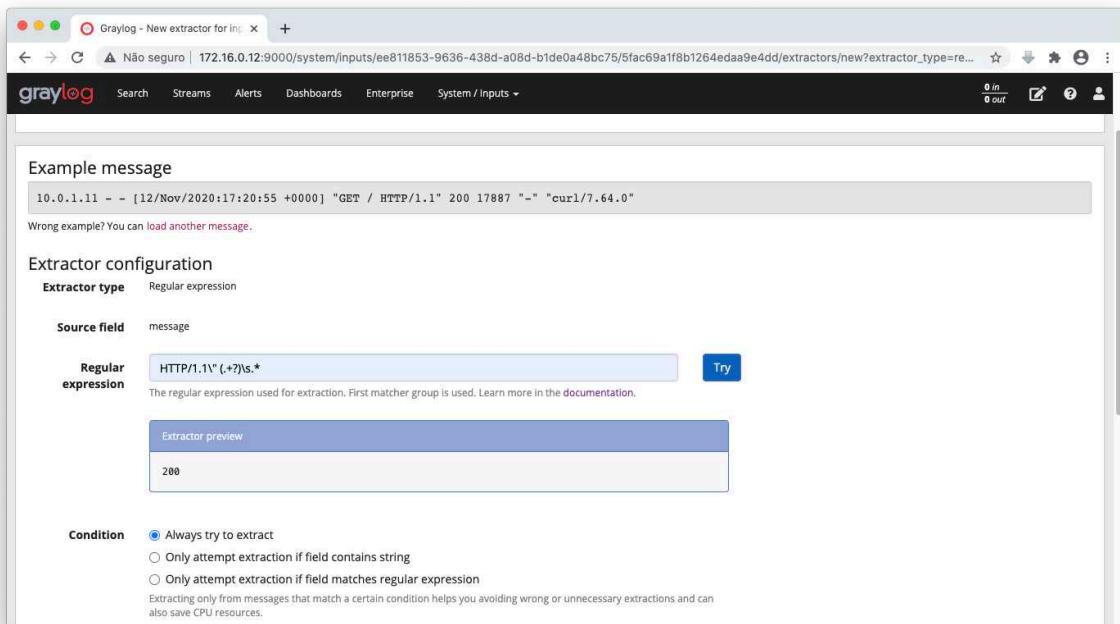


Fig. 4.21: Configurar Manage extractors - ETAPA 5

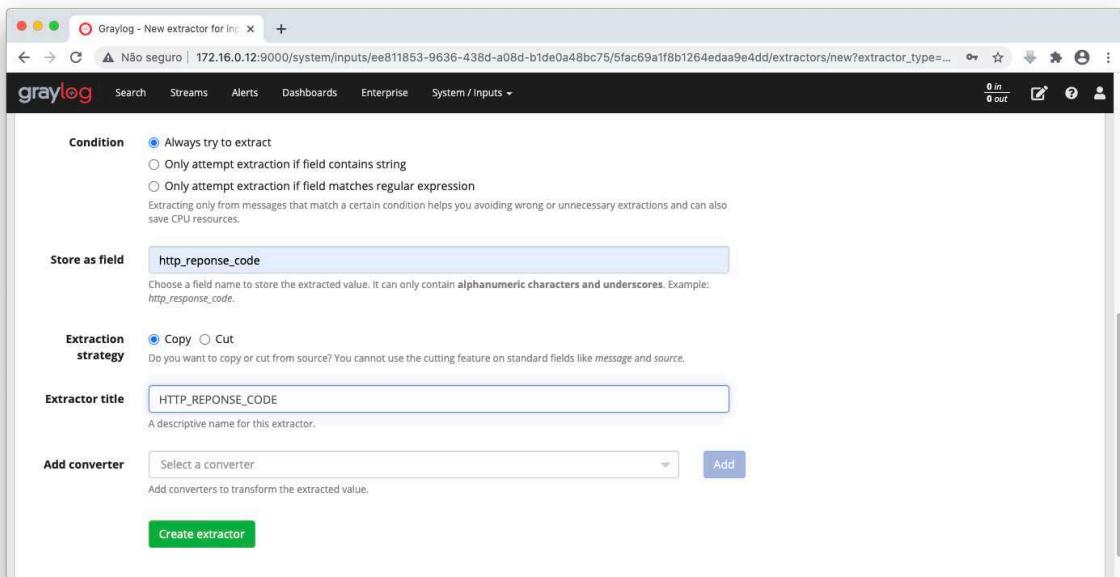


Fig. 4.22: Configurar Manage extractors - ETAPA 6

Em seguida clique em **Create extractor**.

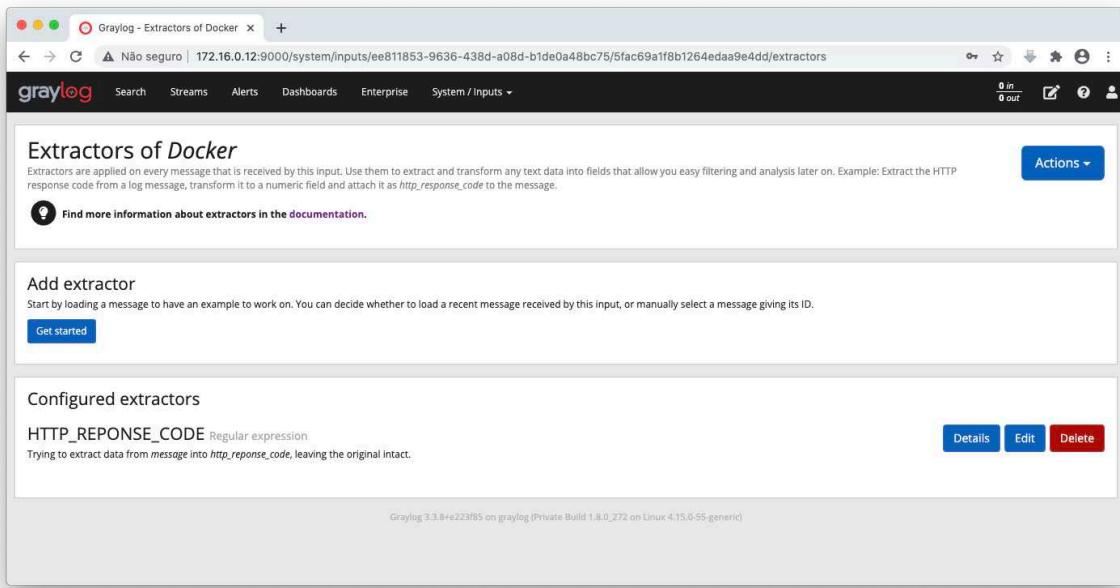


Fig. 4.23: Configurar Manage extractors - ETAPA 7

Através da VM **Kibana-audit**, gere o código **200**:

```
1 | for cont in $(seq 1 10); do curl http://wordpress.4labs.example;
   | done
```

Faça um segundo um teste de acesso para gerar o código **404**:

```
1 | for cont in $(seq 1 10); do curl http://wordpress.4labs.example/
   | teste; done
```

Faça um terceiro um teste de acesso para gerar o código **302**:

```
1 | for cont in $(seq 1 10); do curl http://wordpress.4labs.example/
   | wp-admin/themes.php; done
```

No Graylog clique em **Search** e no campo de busca aplique o filtro **http_reponse_code: 200**, é possível ver as mensagens com o código de acesso **200**.

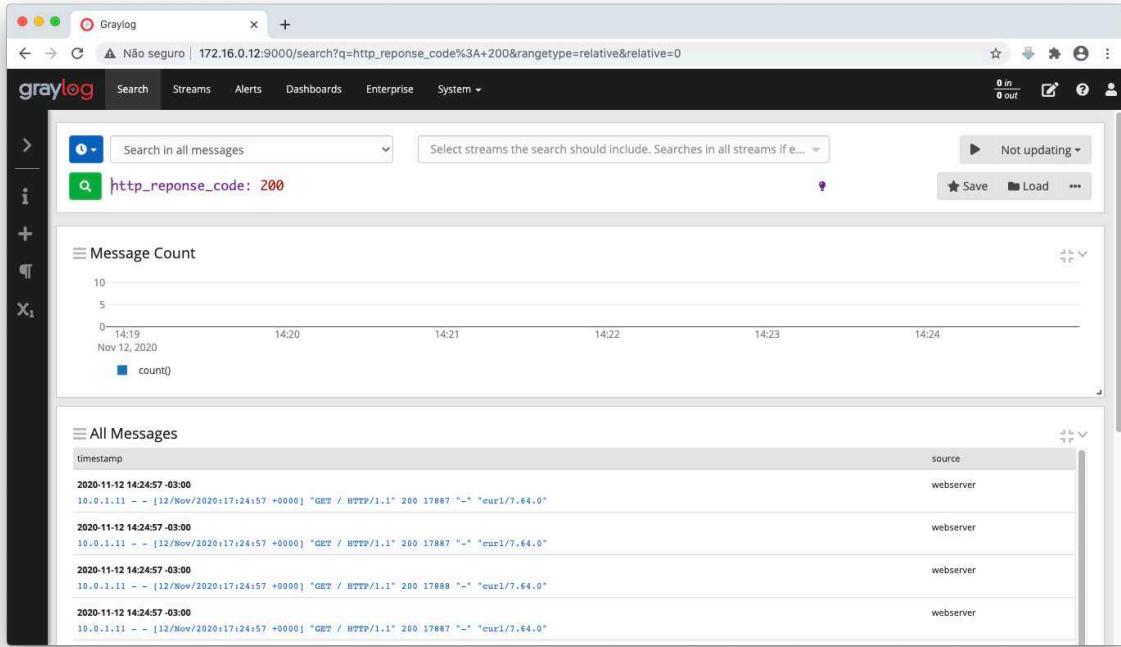


Fig. 4.24: Visualizar Logs - Code 200

No Graylog clique em **Search** e no campo de busca aplique o filtro **http_reponse_code: 404**, é possível ver as mensagens com o código de acesso **404**.

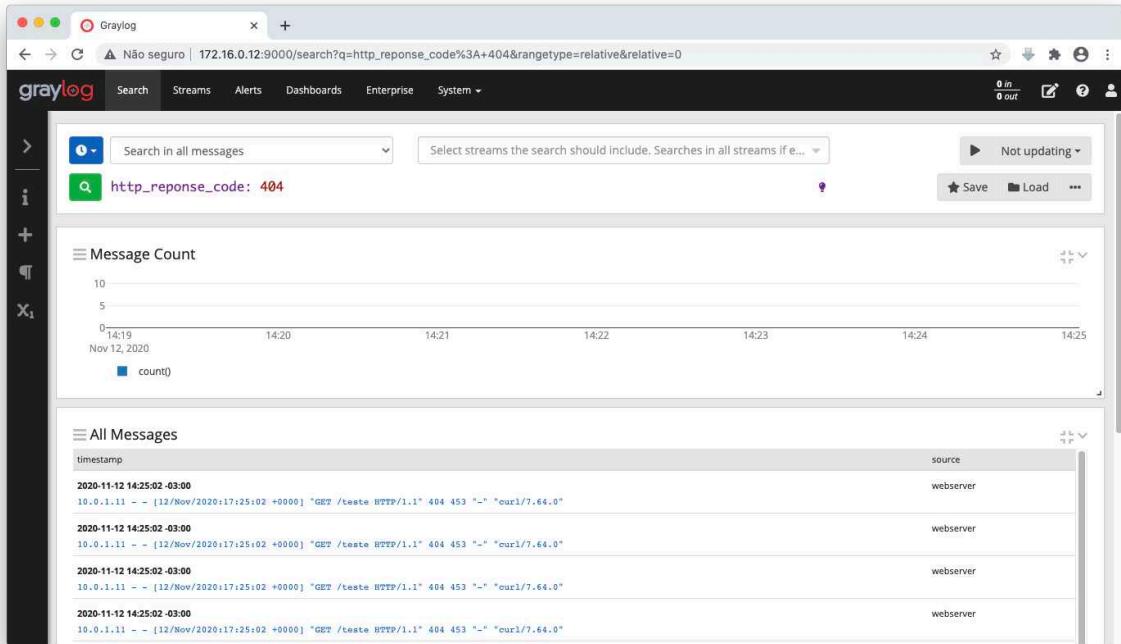


Fig. 4.25: Visualizar Logs - Code 400

No Graylog clique em **Search** e no campo de busca aplique o filtro **http_reponse_code: 302**, é possível ver as mensagens com o código de acesso **302**.

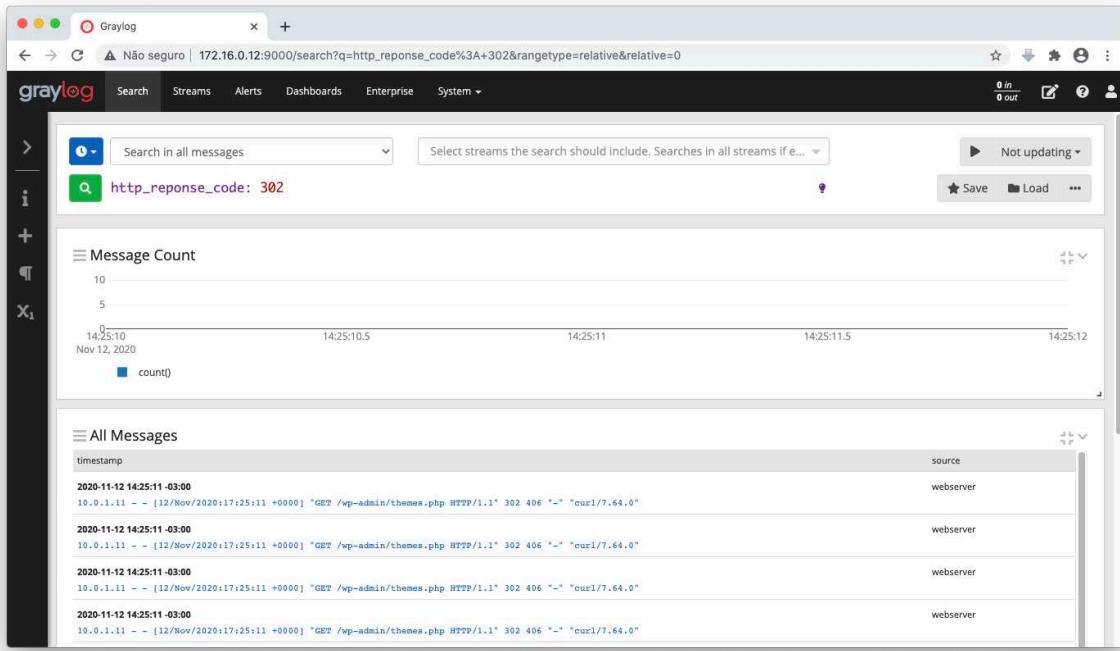


Fig. 4.26: Visualizar Logs - Code 302

Ao clicar em cima da última mensagem para expandi-la, é possível ver o campo do extrator **http_reponse_code** que foi criado.

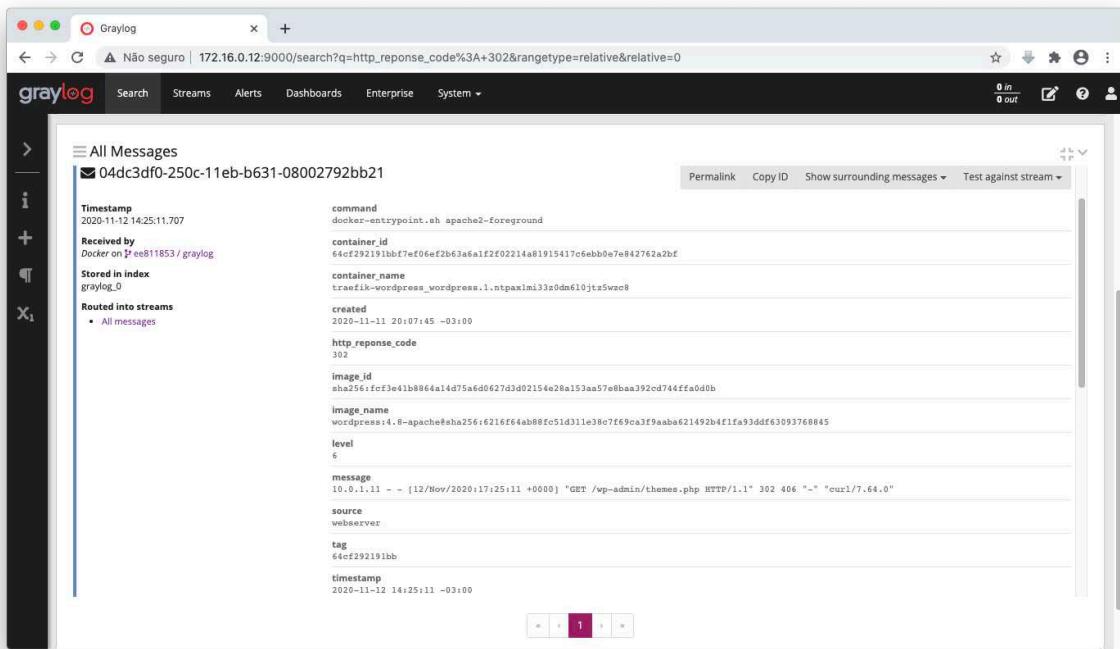


Fig. 4.27: Visualizar logs

Dashboards para o Graylog

O uso de painéis permite que você crie pesquisas predefinidas em seus dados de forma gráfica. Em comparação com as pesquisas salvas, os painéis incluem uma variedade de recursos adicionais. A principal diferença é a possibilidade de definir critérios de pesquisa específicos do widget, como a consulta ou o intervalo de tempo.

Os painéis também oferecem suporte à criação de várias guias para diferentes casos de uso, exibindo o resultado em modo de tela inteira e, conforme descrito, compartilhando com outras pessoas.

Fonte: <https://docs.graylog.org/en/3.3/pages/dashboards.html>

LAB 4.6 - Criar um dashboard para estatísticas de acesso

Vamos criar um dashboard para gerar as estatísticas de acesso a página do Wordpress.

1 - Criando o Dashboard de Pizza (Pie Chart)

Para criar um dashboard no Graylog, clique em **Dashboards** -> **Create new dashboard**

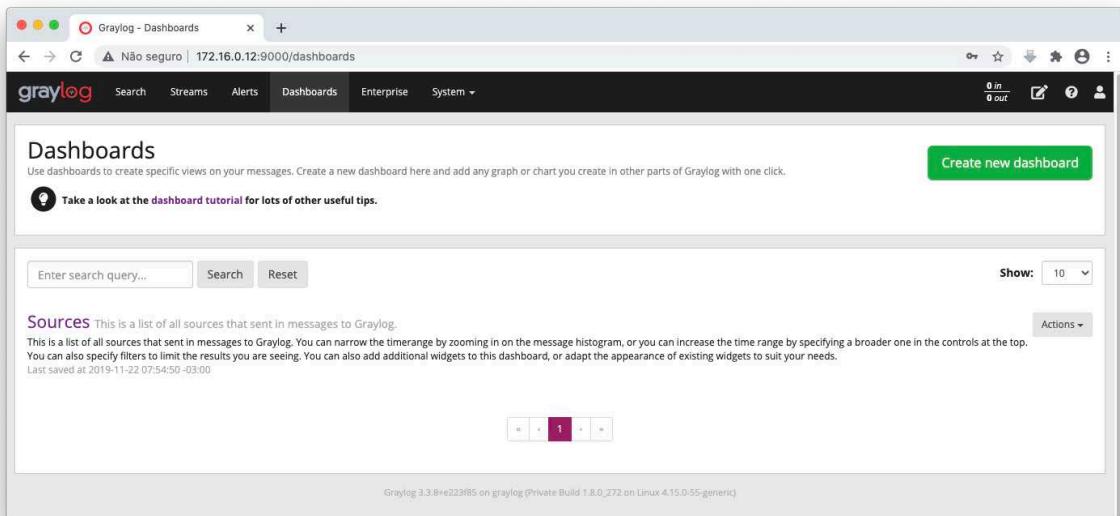


Fig. 4.28: Criar dashboards - ETAPA 1

Em seguida clique no botão **Save as**

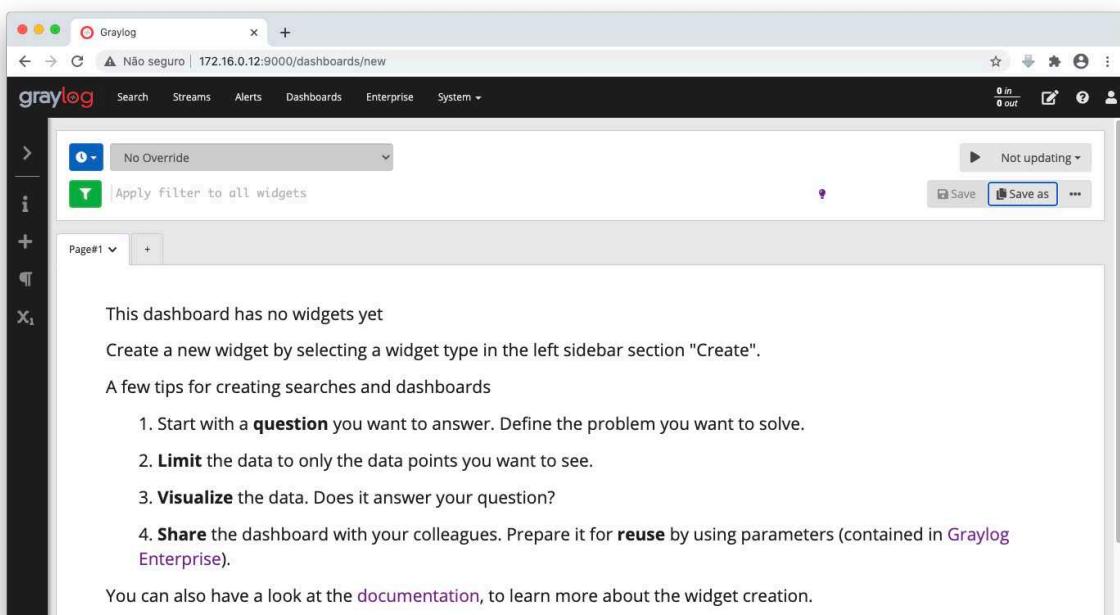


Fig. 4.29: Criar dashboards - ETAPA 2

E preencha os campos:

Title: Wordpress Dashboard **Summary:** Wordpress Dashboard **Description:** Dashboard para estatísticas do site Wordpress.

Em seguida clique em **Save**.

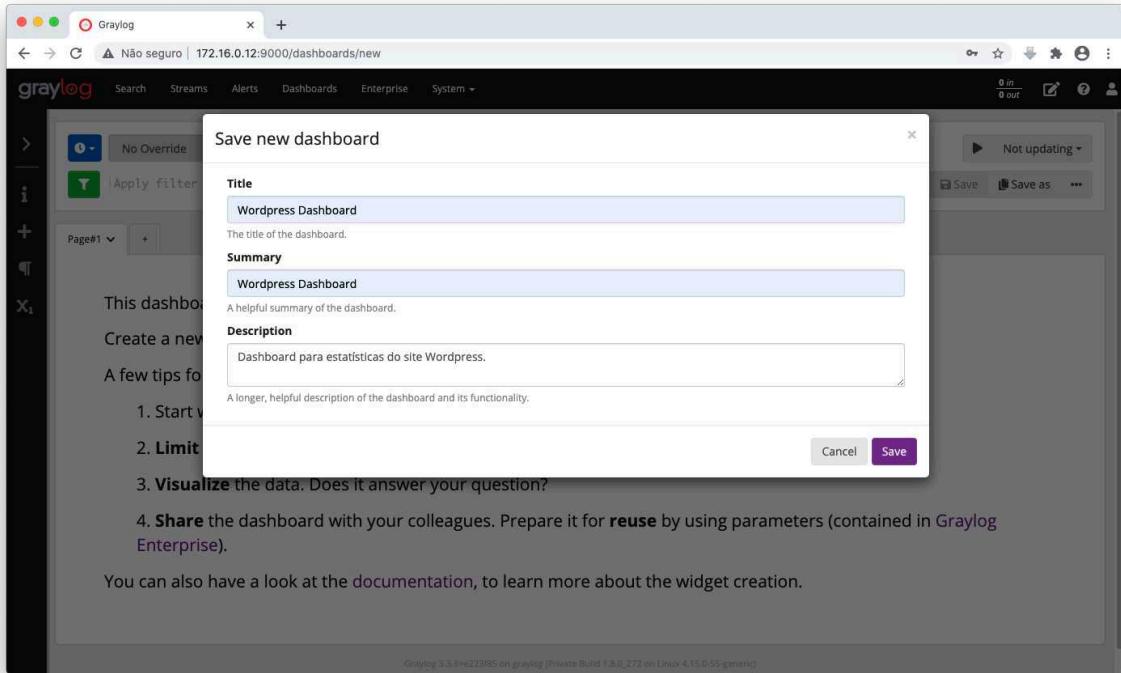


Fig. 4.30: Criar dashboards - ETAPA 3

Retorne ao menu **Dashboards** para confirmar a criação do Wordpress Dashboard.

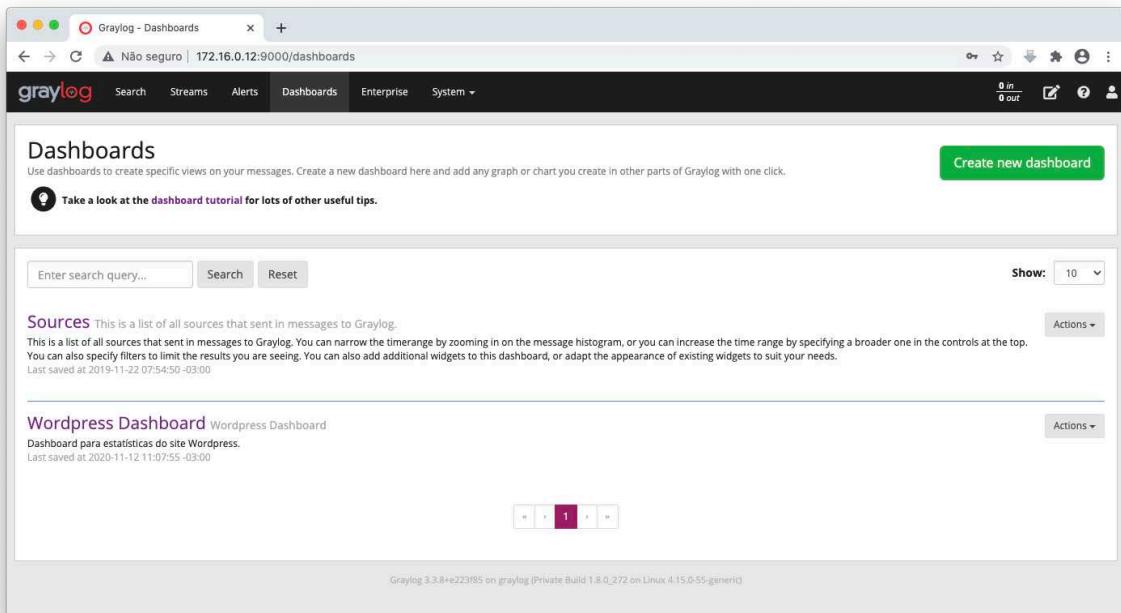


Fig. 4.31: Criar dashboards - ETAPA 4

Para popular esse dashboard com dados, clique em **System > Inputs > Docker > Show received messages** e realize uma busca alterando os seguintes campos:

Selecione por **Search in all messages**

Not updating: 5 seconds **No campo de filtro:** `_exists_:http_reponse_code`

Na área **Message Count > Actions** clique em **Edit**

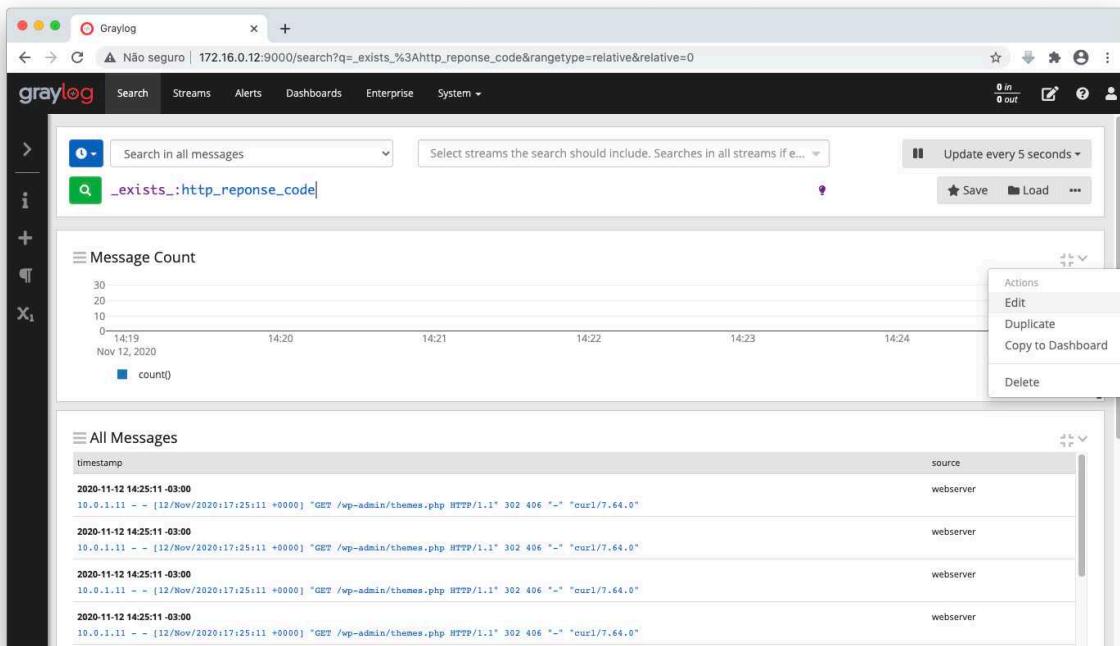


Fig. 4.32: Criar dashboards - ETAPA 5

Altere as seguintes informações:

VISUALIZATION TYPE: Pie Chart **ROWS:** `http_reponse_code`

Em seguida clique em **Save**.

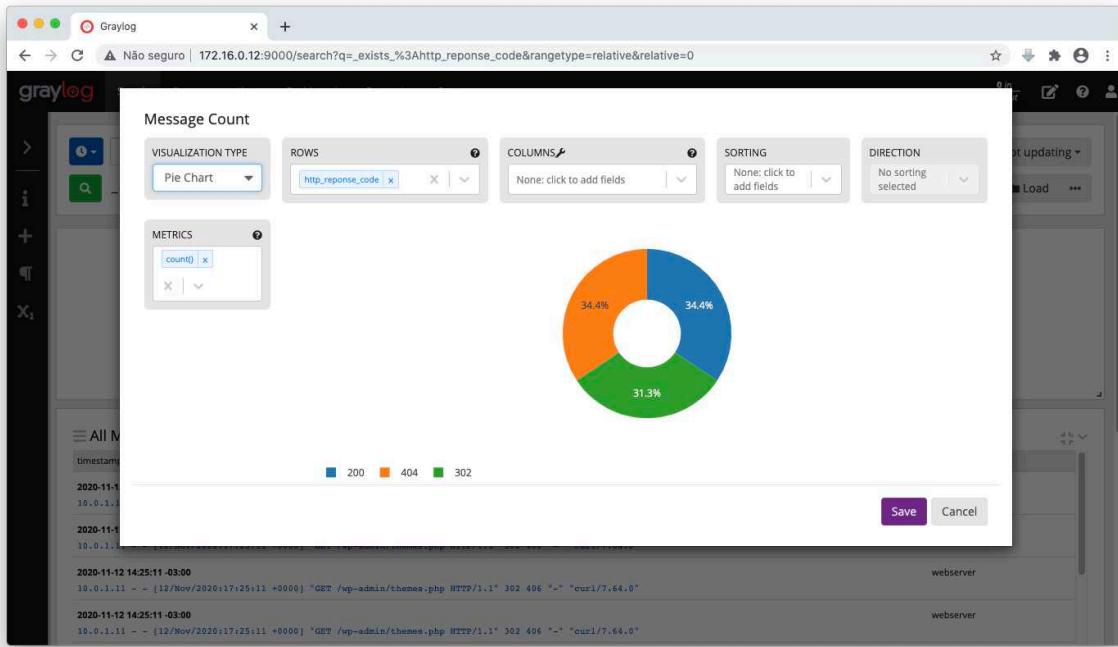


Fig. 4.33: Criar dashboards - ETAPA 6

Em seguida confirme o resultado:

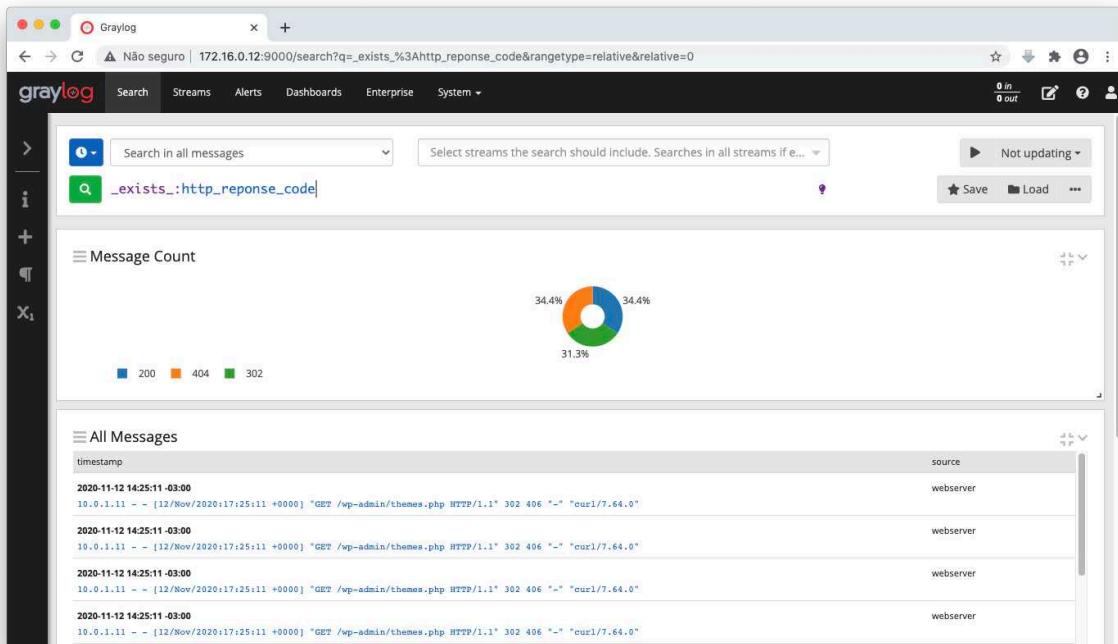


Fig. 4.34: Criar dashboards - ETAPA 7

Na área **Message Count** > **Actions** clique em **Edit** > **Copy to Dashboard**

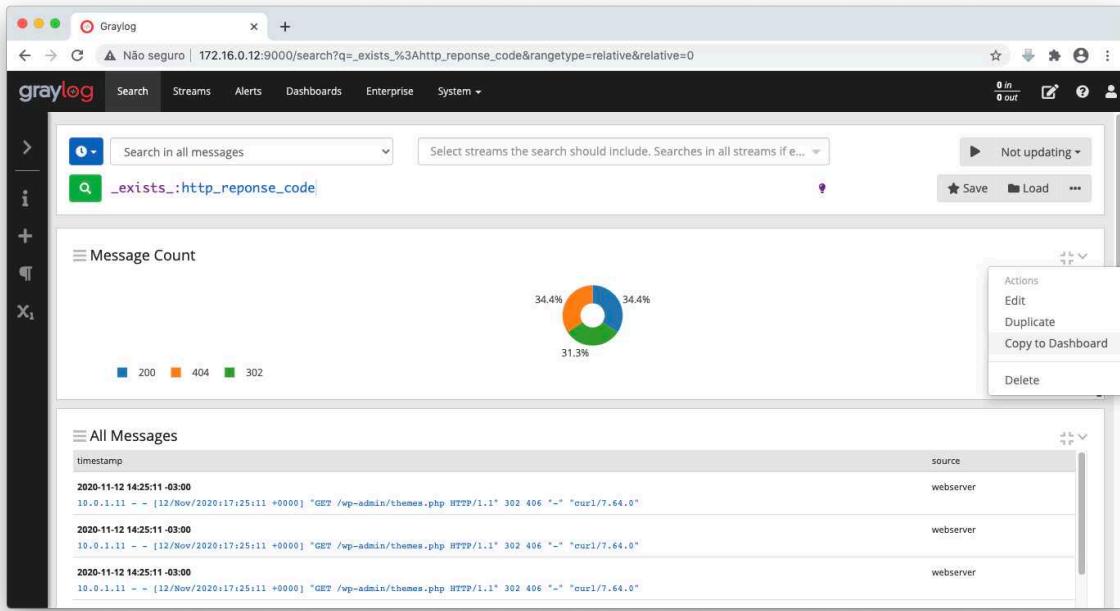


Fig. 4.35: Criar dashboards - ETAPA 8

Selecione o dashboard **Wordpress Dashboard** e clique no botão **Select**

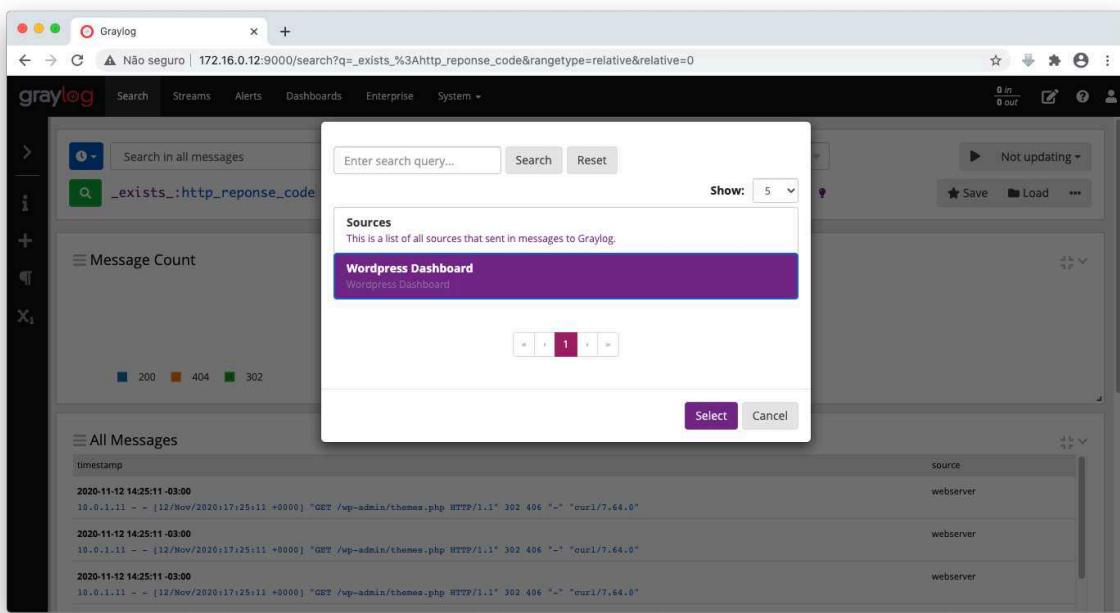


Fig. 4.36: Criar dashboards - ETAPA 9

No Dashboard clique em **Actions > Edit** para renomear o painel criado

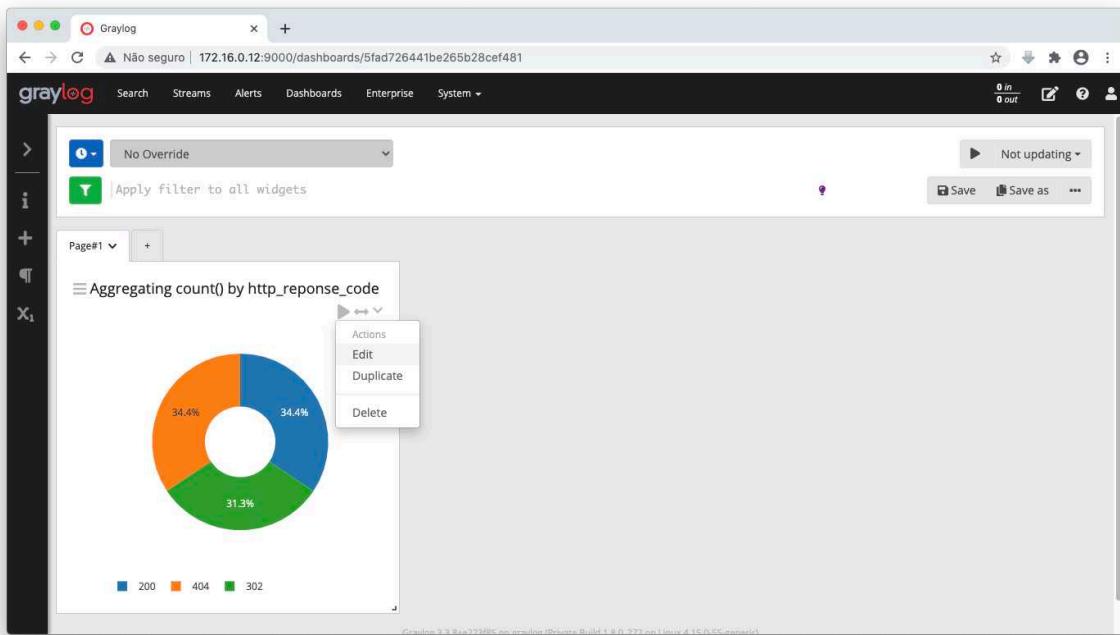


Fig. 4.37: Criar dashboards - ETAPA 10

Altere o título para **Estatísticas de Acesso - Wordpress** e clique no botão **Save**

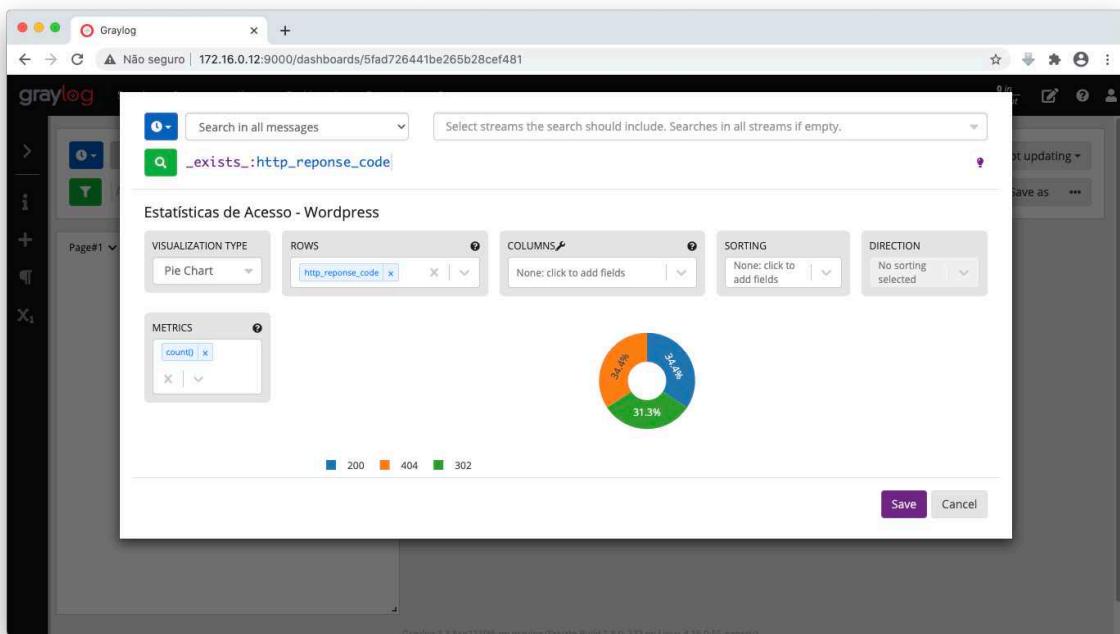


Fig. 4.38: Criar dashboards - ETAPA 11

Em seguida confirme o resultado e clique no botão **Save**:

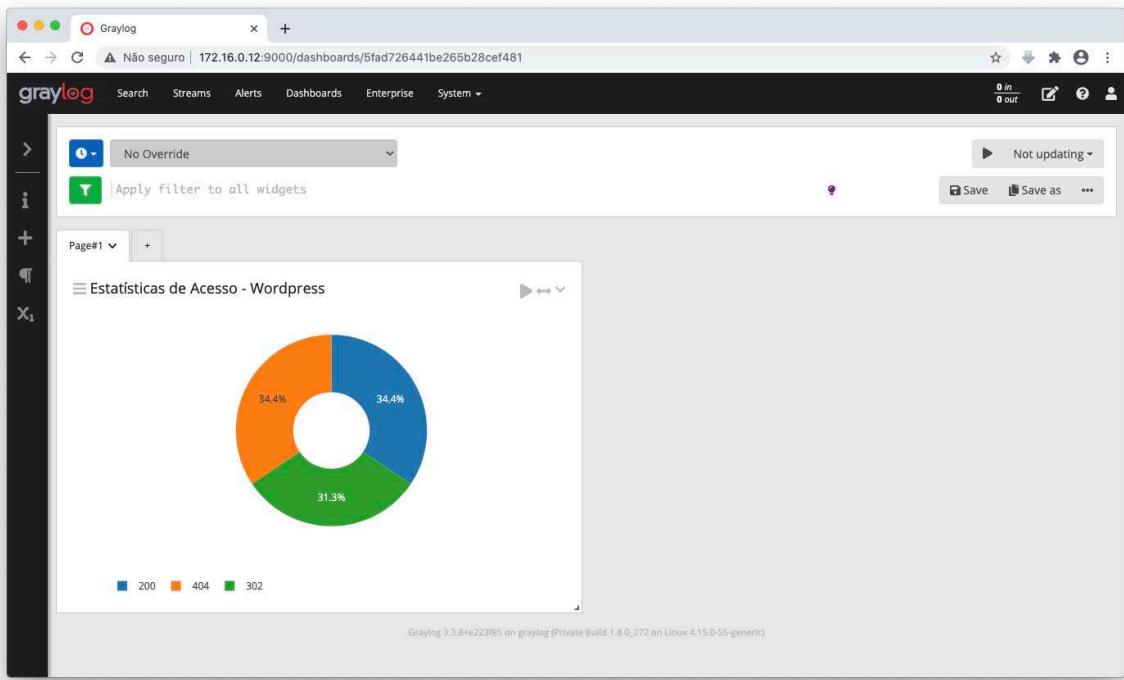


Fig. 4.39: Criar dashboards - ETAPA 12

2 - Dashboard de contagem de acesso

Clique em **System > Inputs > Docker > Manage extractors > Show received messages** e realize uma busca alterando os seguintes campos:

Selecione por **Search in all messages**

Not updating: 5 seconds **No campo de filtro:** _exists_:*http_reponse_code

Na área **All Messagens > Actions** clique em **Edit**

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

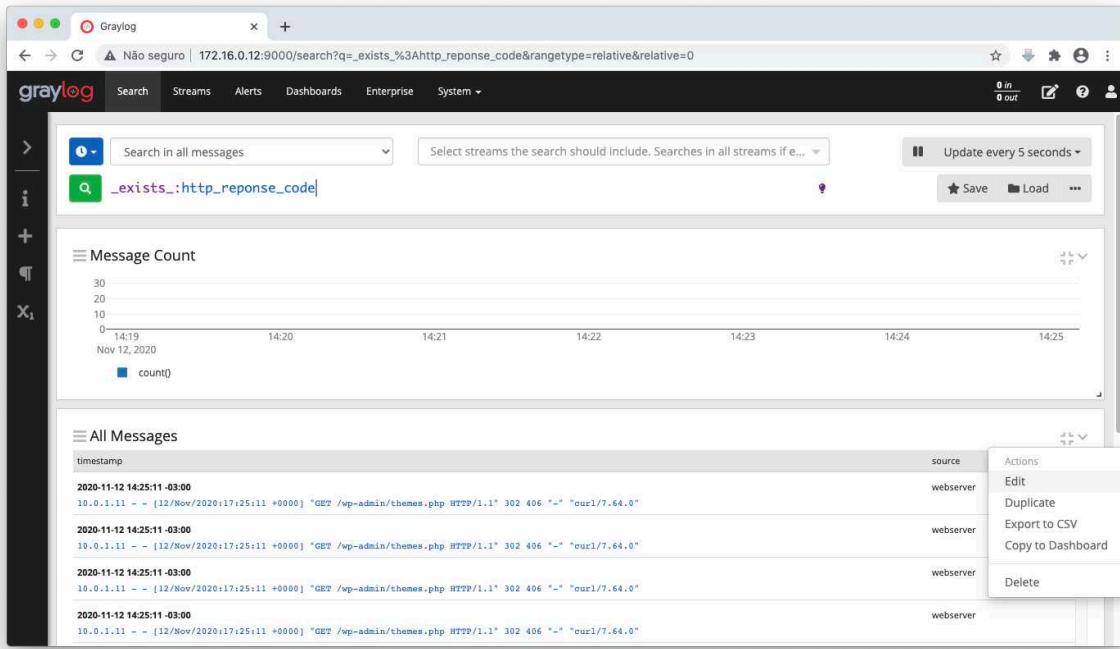


Fig. 4.40: Dashboard de Contagem de Acesso - ETAPA 1

Ao lado esquerdo da tela na caixa FIELDS, selecione o campo **http_reponse_code** e clique no botão **Save**

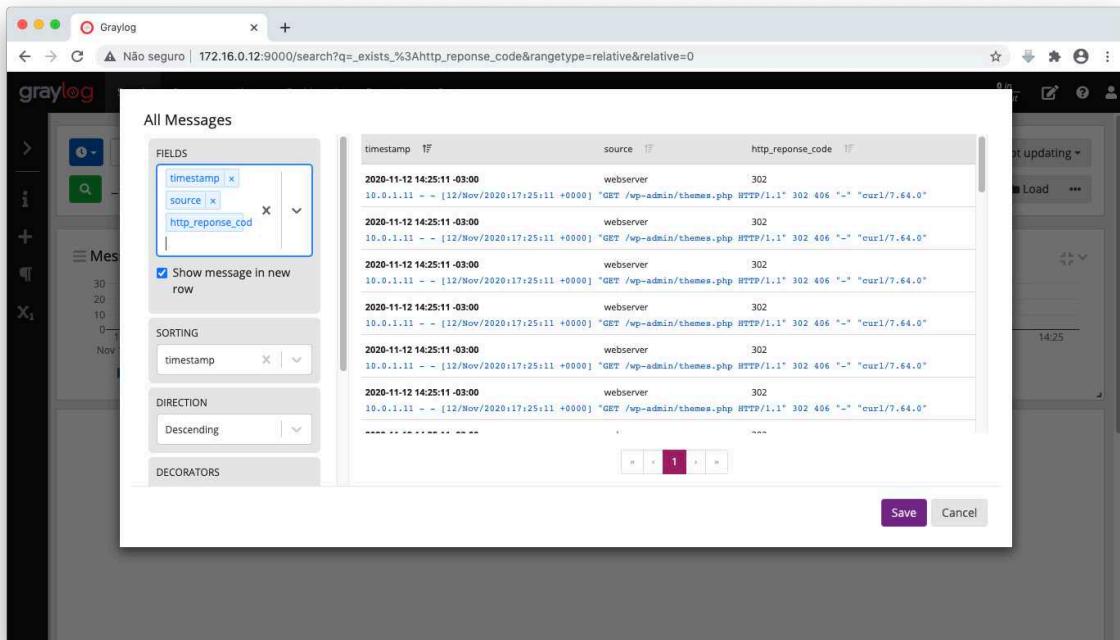


Fig. 4.41: Dashboard de Contagem de Acesso - ETAPA 2

Na área **All Messages** selecione **Actions** no campo **http_reponse_code** e clique na opção **Show top values**

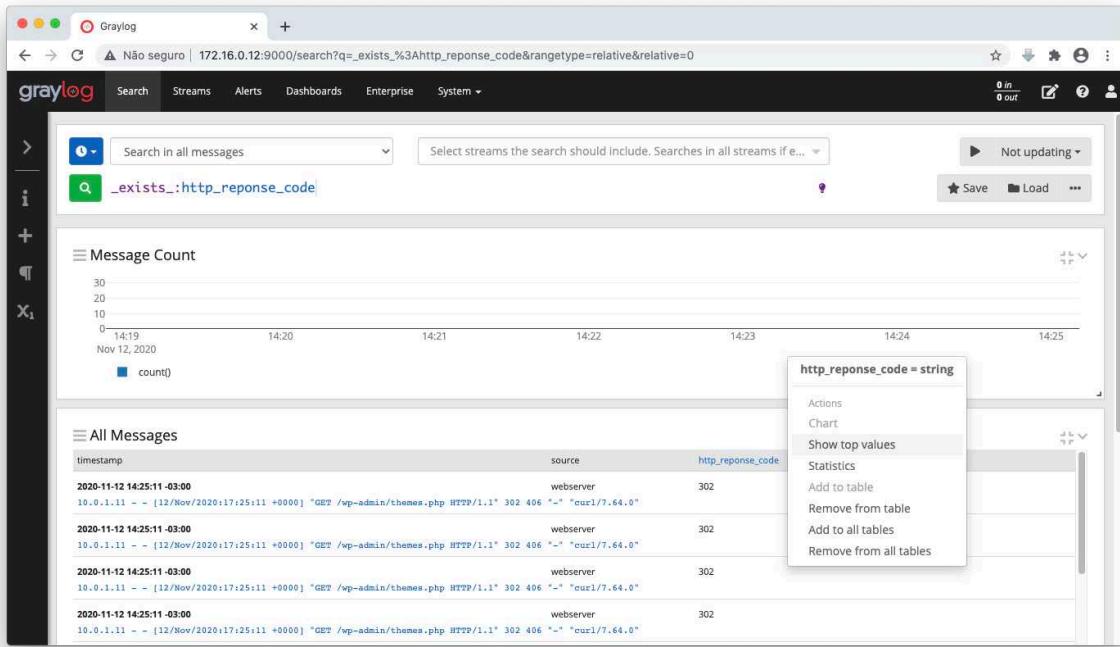


Fig. 4.42: Dashboard de Contagem de Acesso - ETAPA 3

No painel clique em **Actions** > **Edit** para renomear

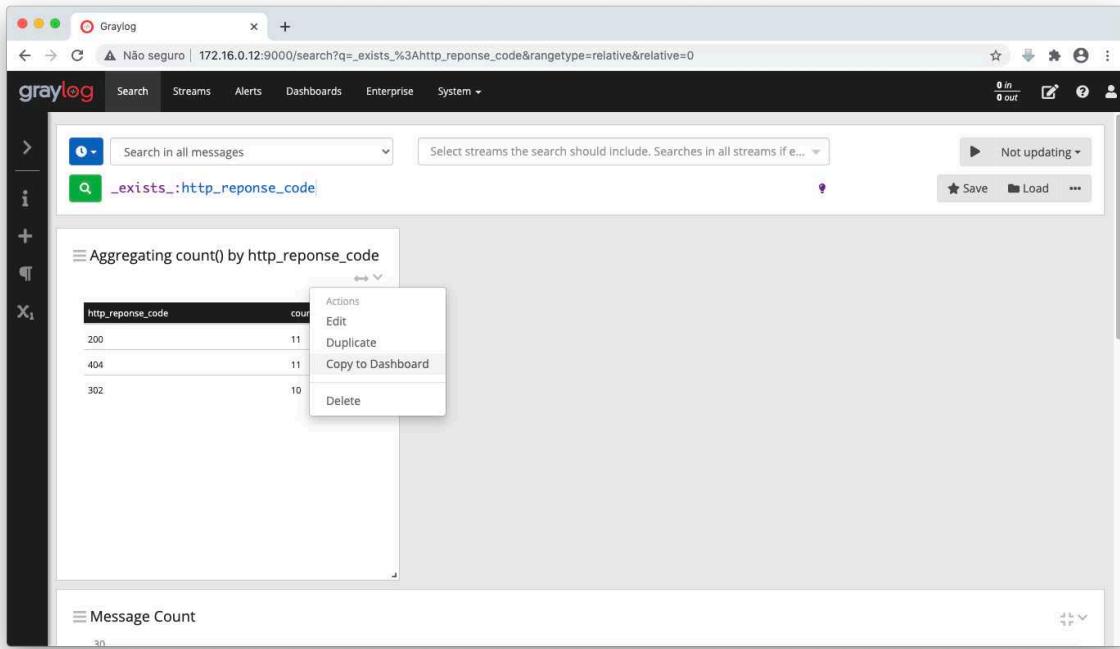


Fig. 4.43: Dashboard de Contagem de Acesso - ETAPA 4

Altere o título para **Contagem de acesso por HTTP REONSE CODE** e clique no botão **Save**

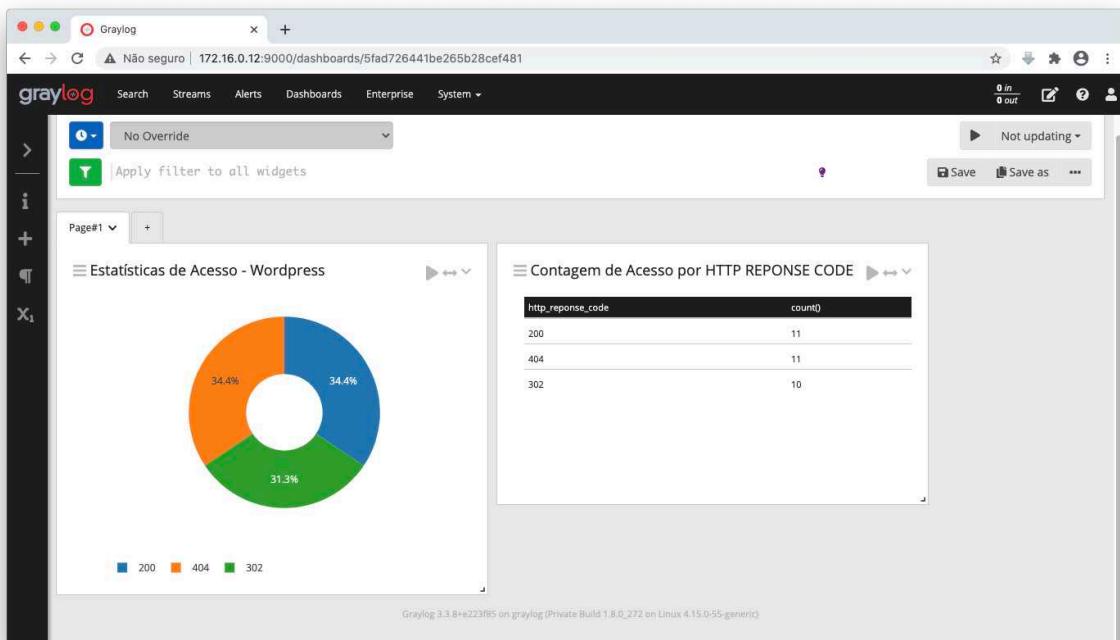


Fig. 4.44: Dashboard de Contagem de Acesso - ETAPA 5

3 - Dashboard de contagem total de acessos

Reita a criação do Dashboard de Pizza e na etapa para selecionar o tipo, escolha **Single Number** e clique no botão **Save**.

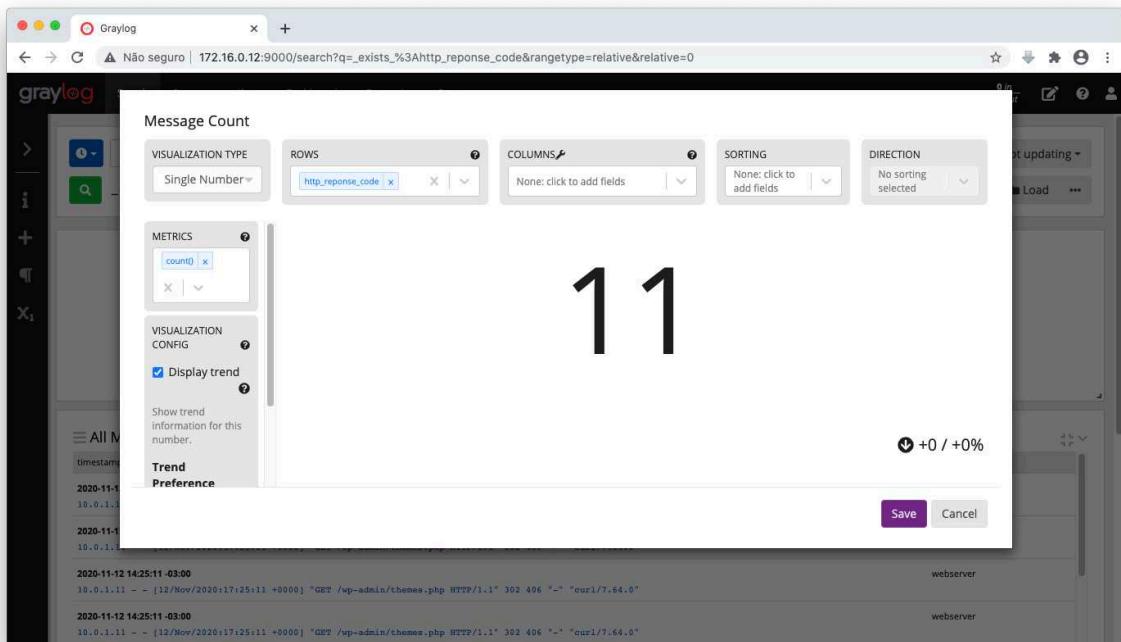


Fig. 4.45: Dashboard de Contagem total de Acessos - ETAPA 1

Envie para o **Dashboard Wordpress** o novo painel.

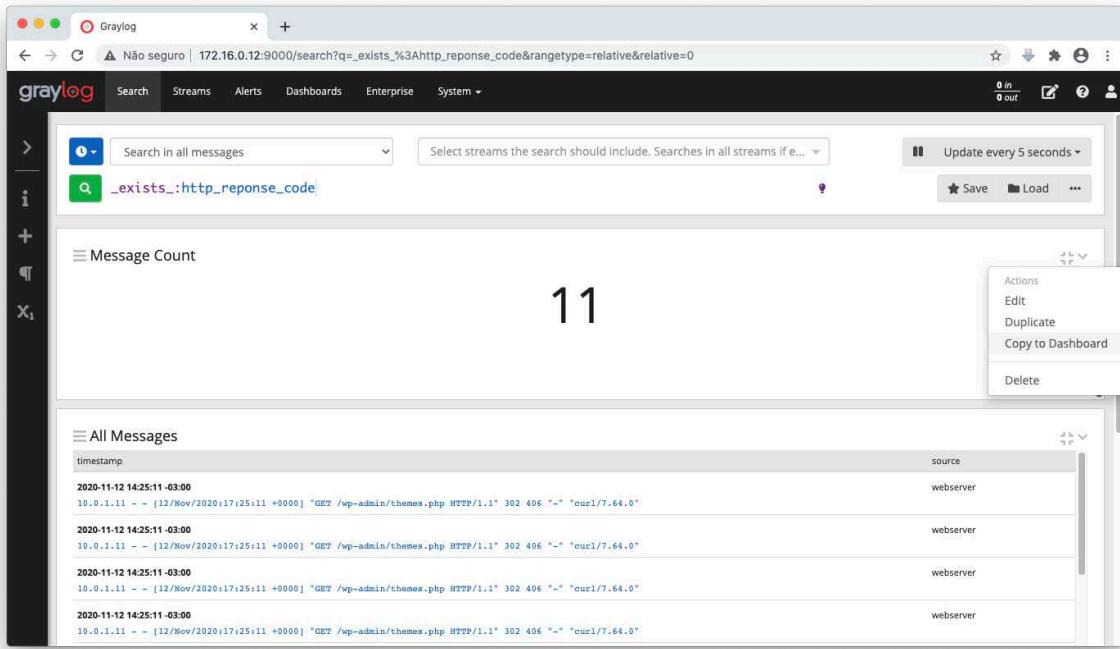


Fig. 4.46: Dashboard de Contagem total de Acessos - ETAPA 2

Altere o título para **Numero total de acesso Wordpress** e clique no botão **Save**

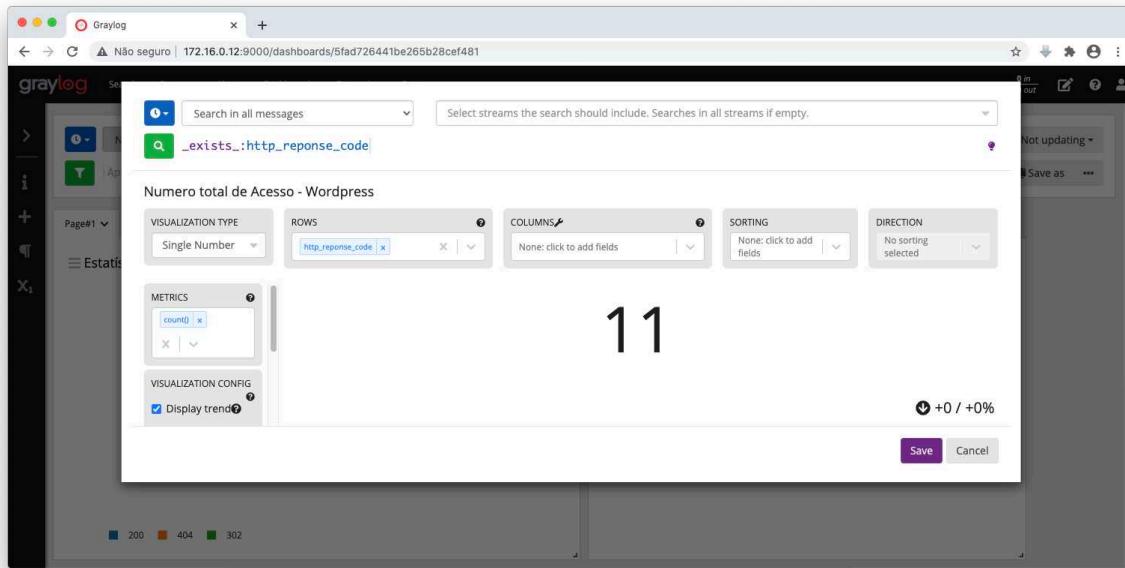


Fig. 4.47: Dashboard de Contagem total de Acessos - ETAPA 3

4 - Preencher os Dashboards

Através da VM **kibana-audit**, gera alguns acessos na aplicação web Wordpress:

```
1 | for cont in $(seq 1 10); do curl http://wordpress.4labs.example;  
| done
```

Faça um segundo um teste de acesso para gerar o código **404**:

```
1 | for cont in $(seq 1 10); do curl http://wordpress.4labs.example/  
| teste; done
```

Faça um terceiro um teste de acesso para gerar o código **302**:

```
1 | for cont in $(seq 1 10); do curl http://wordpress.4labs.example/  
| wp-admin/themes.php; done
```

Faça um quarto um teste de acesso para gerar o código **200**:

```
1 | for cont in $(seq 1 100); do curl http://wordpress.4labs.example  
| ; done
```

Retorne ao Graylog e mude para atualizar a cada 1 segundo e veja o resultado

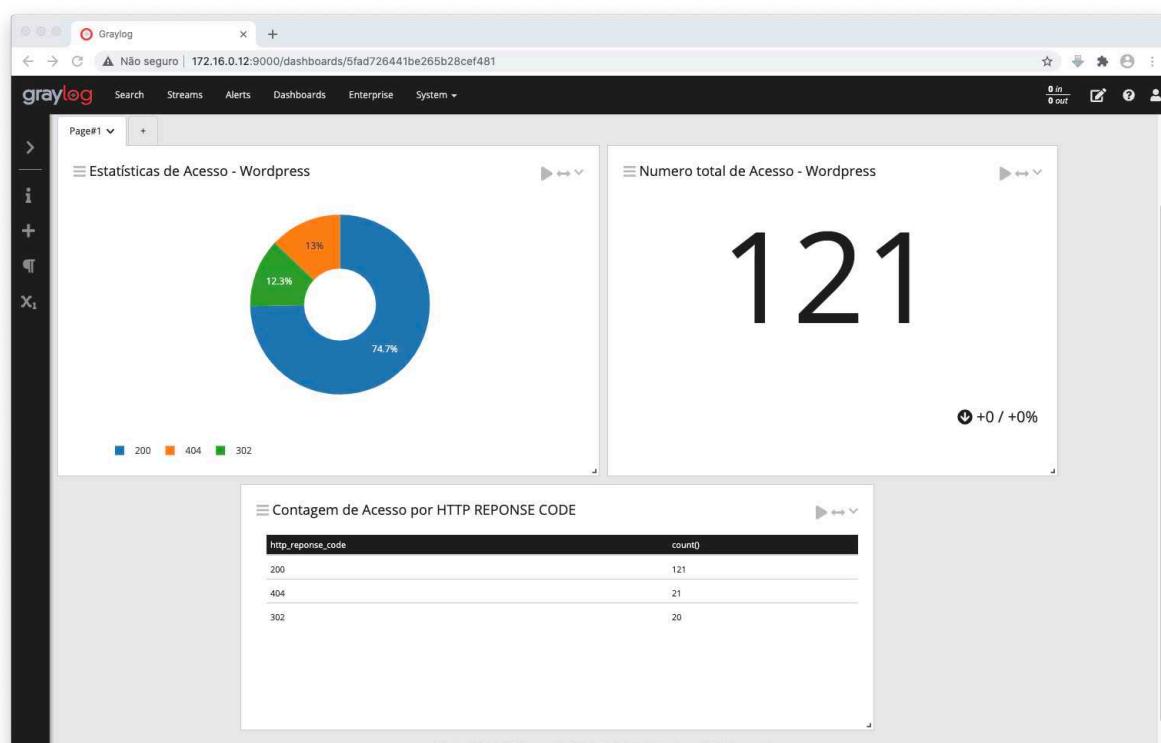


Fig. 4.48: Dashboard de Contagem total de Acessos - ETAPA 4

Criação de alertas no Graylog

Os alertas são criados usando definições de eventos que consistem em condições. Quando uma determinada condição for atendida, ela será armazenada como um evento e pode ser usada para acionar uma notificação. Se o seu sistema tiver uma licença corporativa, os eventos podem ser combinados para criar correlações.

O Graylog vem com condições de alerta padrão e notificações de alerta, e ambos podem ser estendidos com Plugins.

Fonte: <https://docs.graylog.org/en/3.3/pages/alerts.html>

LAB 4.7 - Criar alerta para envio de email

Vamos criar um alerta para enviar email quando o acesso a página do Wordpress gerar o código 404.

1 - Preparação do Graylog

Acesse a VM **graylog-audit**, através do comando **ssh** e alterne para a conta do usuário **root**.

```
1 | ssh suporte@172.16.0.12
2 | sudo su -
```

Em seguida configure o arquivo **/etc/graylog/server/server.conf** com as informações de seu servidor de email.

```
1 | vim /etc/graylog/server/server.conf
2 |
3 |
4 | # Email transport
5 | transport_email_enabled = true
6 | transport_email_hostname = smtp.gmail.com
7 | transport_email_port = 587
8 | transport_email_use_auth = true
9 | transport_email_auth_username = seuemail@gmail.com
10 | transport_email_auth_password = suasenhaqui
11 | transport_email_subject_prefix = [graylog]
12 | transport_email_from_email = graylog@4labs.example
13 | transport_email_use_ssl = false
```

No exemplo apresentando, estamos utilizando as informações do servidor Gmail. Mude as informações para o servidor de email de sua preferencia.

Reinicie o serviço do Graylog para aplicar as alterações:

```
1 | systemctl restart graylog-server
```

2 - Criação do alerta

Para criar um alerta no Graylog, clique em **Alerts > Alerts & Events > Get Started!**

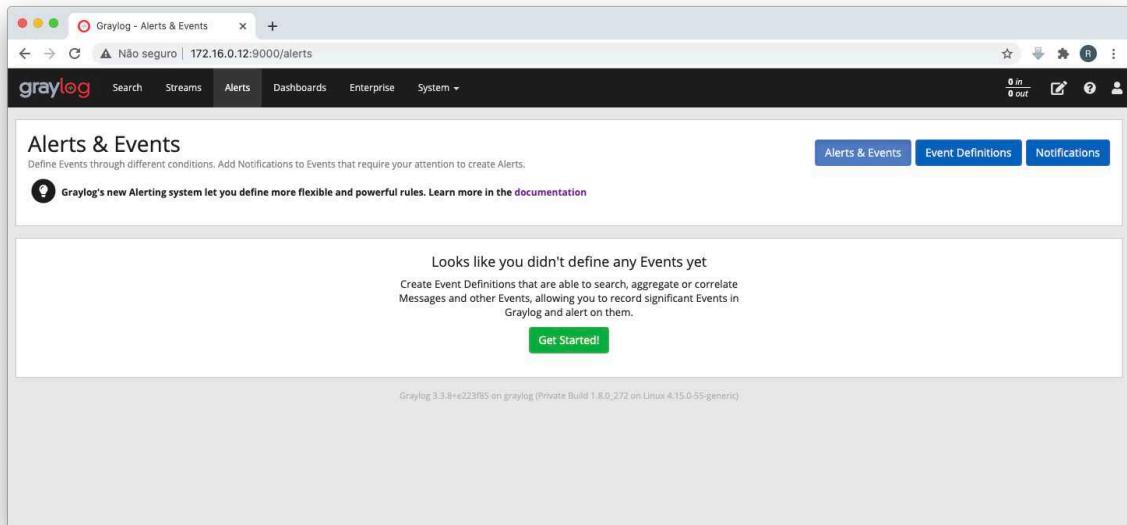


Fig. 4.49: Criar alerta para envio de email - ETAPA 1

Na aba **Event Details**, preencha as seguintes informações:

- **Title:** acesso ao site Wordpress
- **Description:** o acesso ao site do Wordpress gerou o código 404.
- **Priority:** Normal

Clique no botão **Next** para continuar.

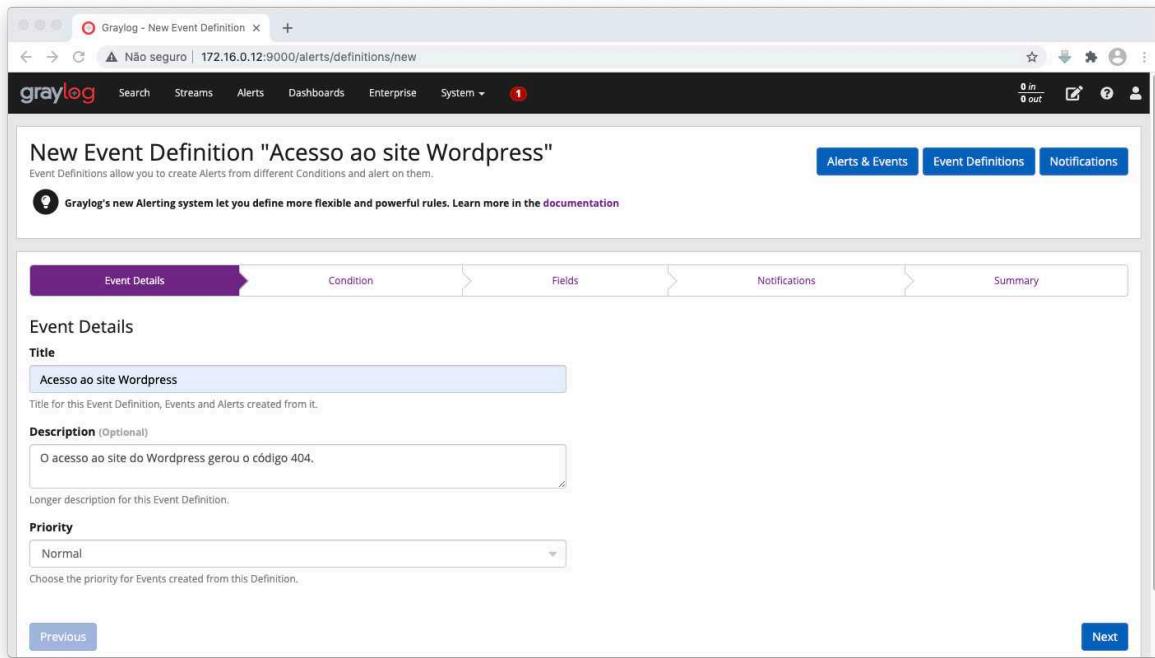


Fig. 4.50: Criar alerta para envio de email - ETAPA 2

Na aba **Filter & Aggregation**, preencha as seguintes informações:

- **Condition Type:** Filter & Aggregation
- **Search Query:** 404
- **Streams:** All messages
- **Search within the last:** 1
- **Search within every:** 1
- **Create Events for Definition if...:** Filter has results

Clique no botão **Next** para continuar.

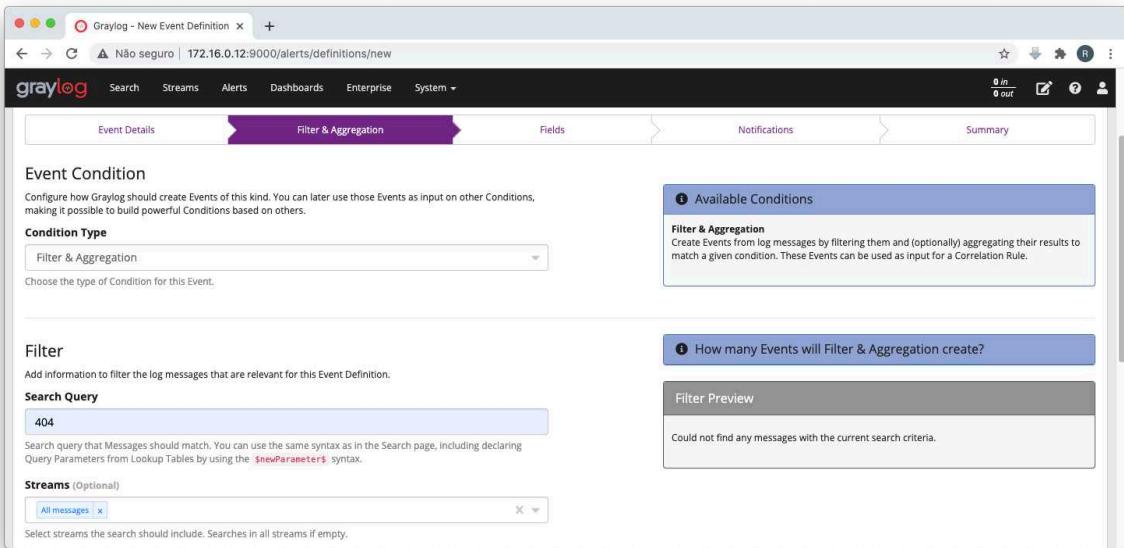


Fig. 4.51: Criar alerta para envio de email - ETAPA 3

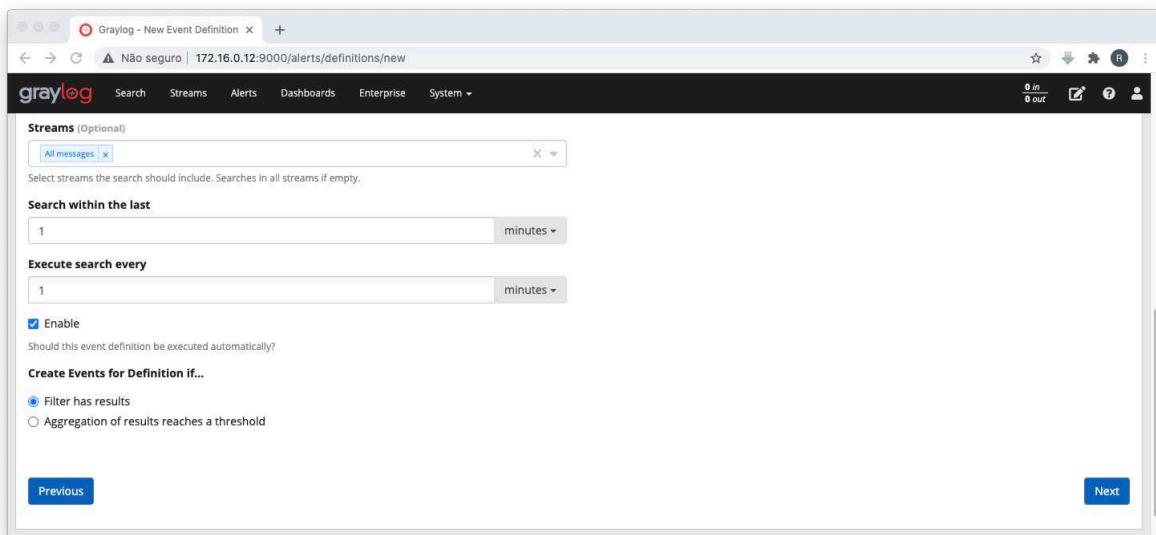


Fig. 4.52: Criar alerta para envio de email - ETAPA 4

A aba **Fields**, clique no botão **Add Custom Field**

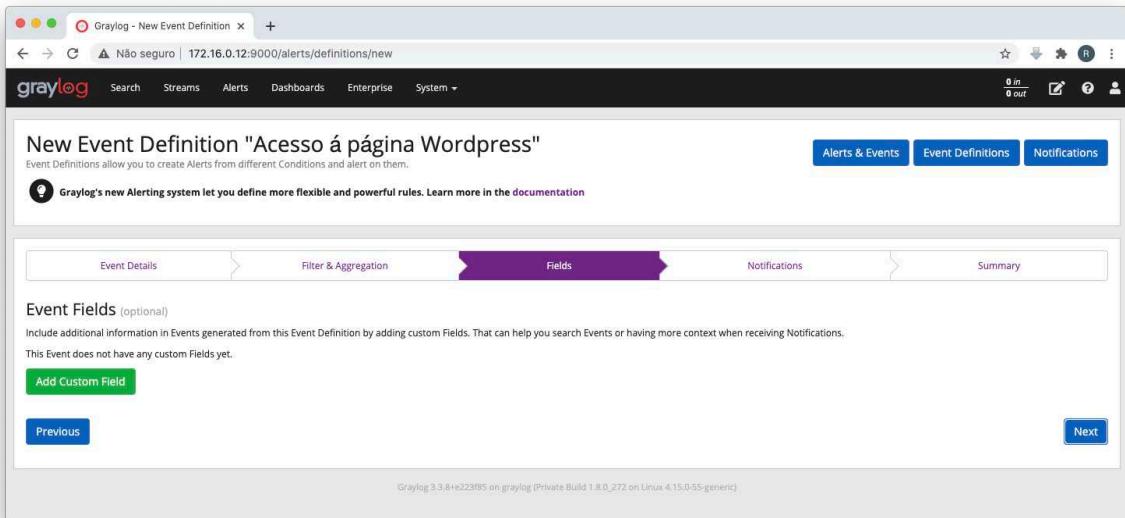


Fig. 4.53: Criar alerta para envio de email - ETAPA 5

E preencha as seguintes informações:

- **Name:** http_reponse_code
- **Set Value From:** Template
- **Template:** Filter

Clique no botão **Done** para criar o **Custom Field**

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

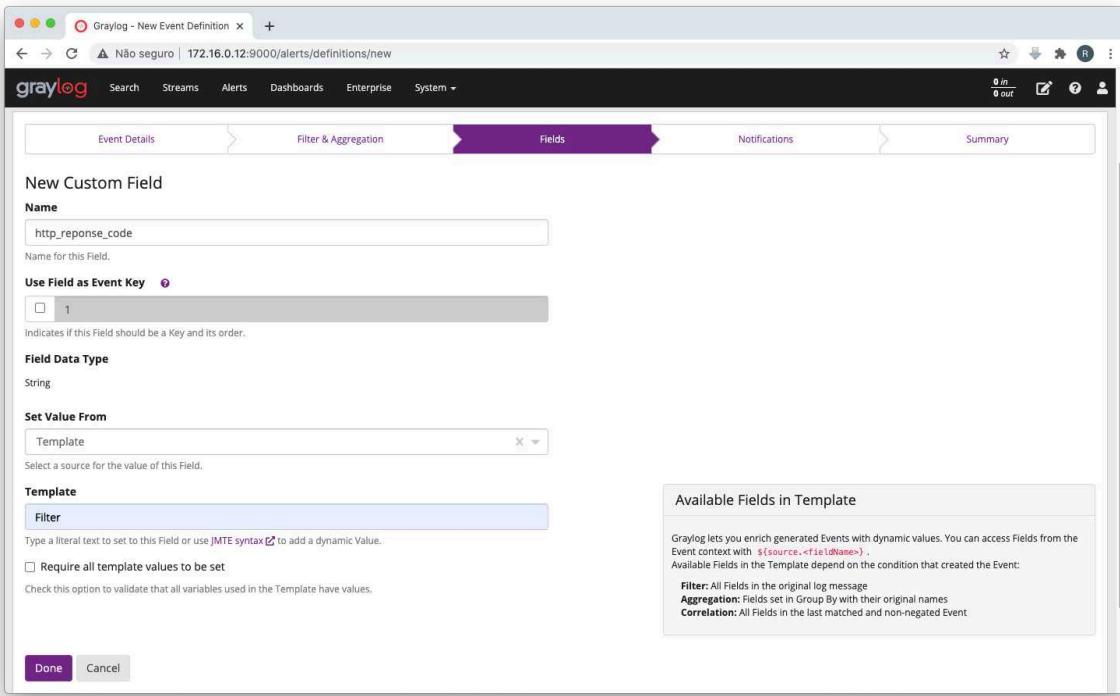


Fig. 4.54: Criar alerta para envio de email - ETAPA 6

Confirme a criação do **Custom Field** e clique no botão **Next** para continuar.

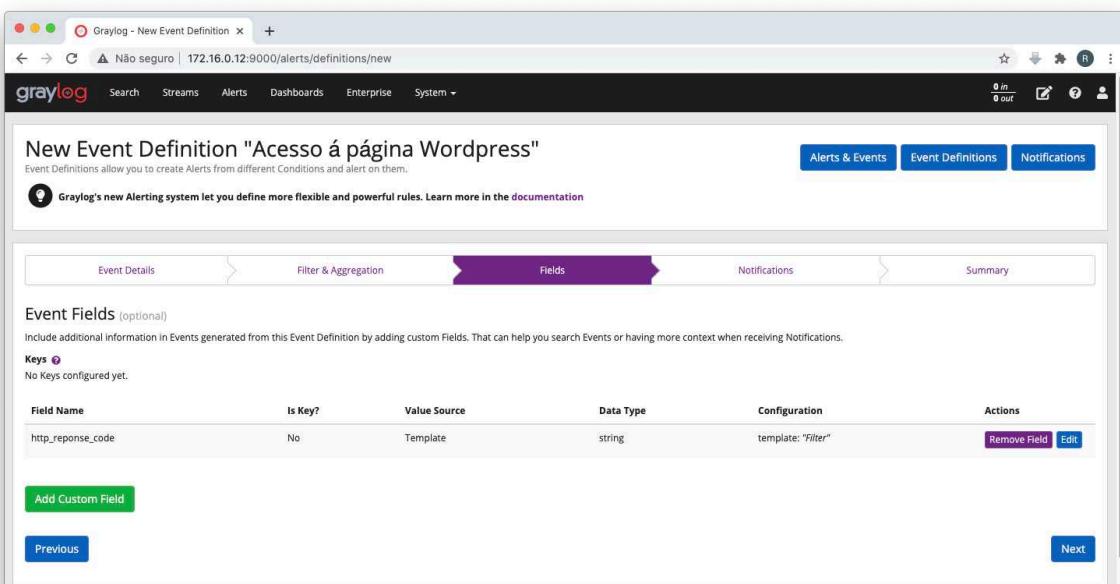


Fig. 4.55: Criar alerta para envio de email - ETAPA 7

A aba **Notifications**, clique no botão **Add Notification**

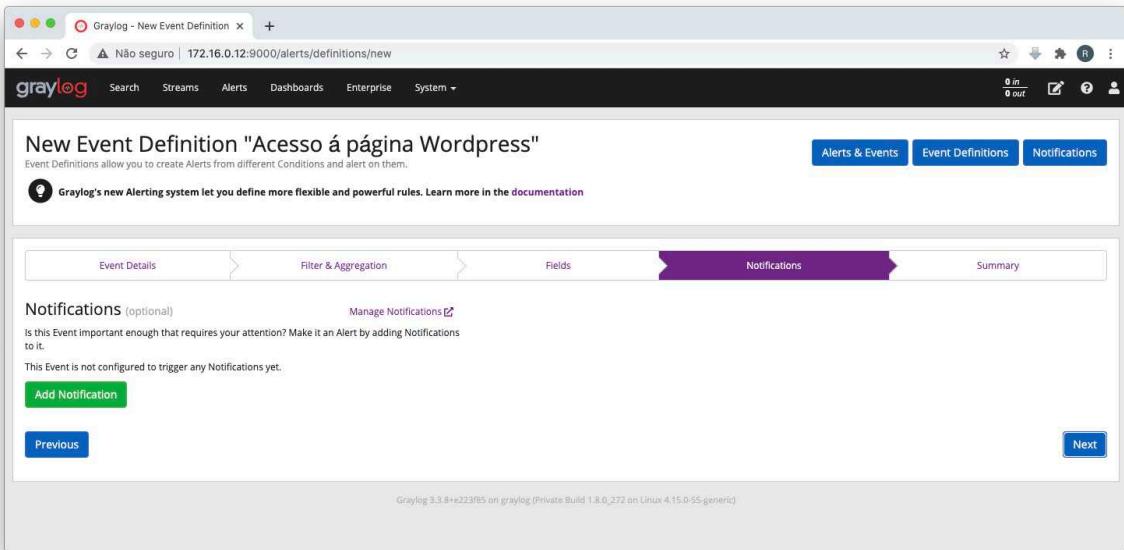


Fig. 4.56: Criar alerta para envio de email - ETAPA 8

E preencha as seguintes informações:

- **Choose Notification:** Create New Notification...
- **Title:** acesso ao site Wordpress
- **Description:** o acesso ao site do Wordpress gerou o código 404.
- **Notification Type:** Email Notification
- **Sender:** seuemail@dominio
- **User recipient(s):** admin (administrator)
- **Email recipient(s):** seuemail@dominio

Clique no botão **Done** para criar a **Notification**

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

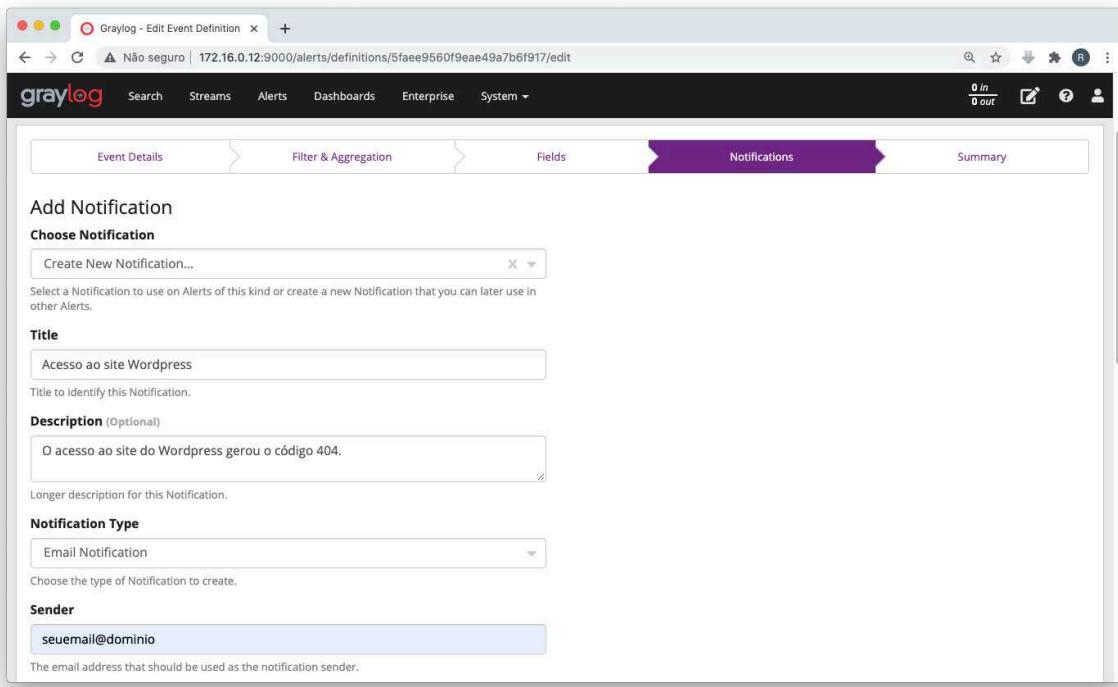


Fig. 4.57: Criar alerta para envio de email - ETAPA 9

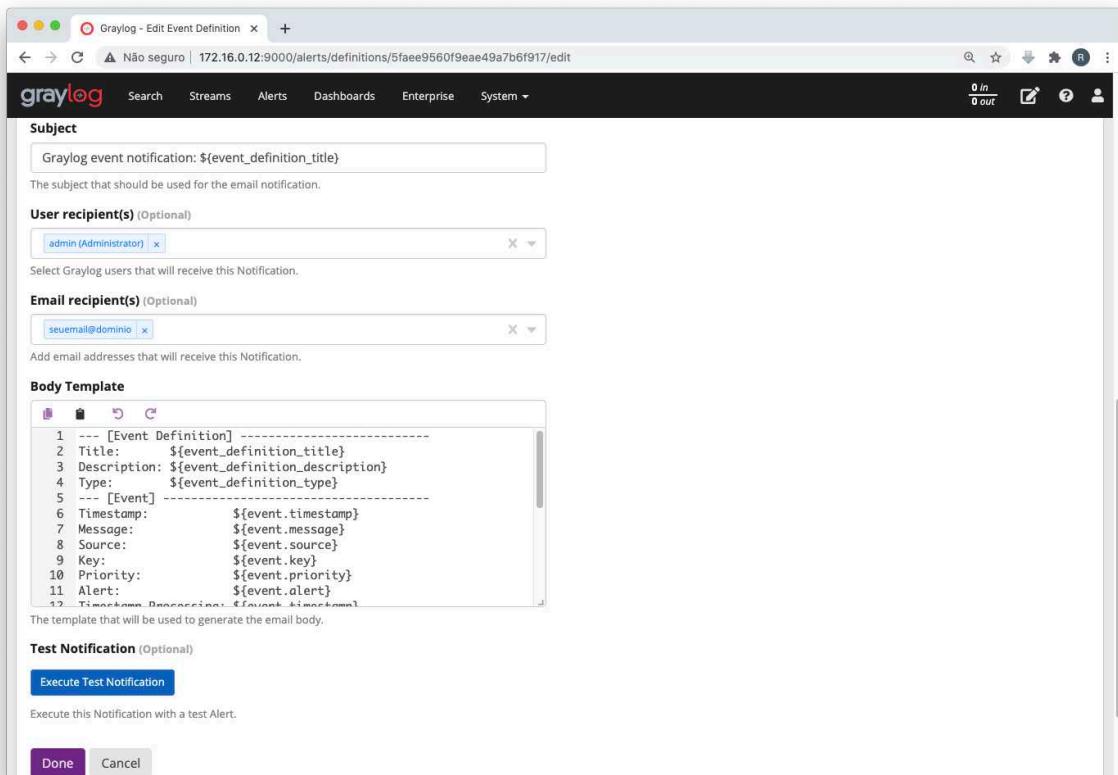


Fig. 4.58: Criar alerta para envio de email - ETAPA 10

Confirme a criação do **Notifications** e clique no botão **Next** para continuar.

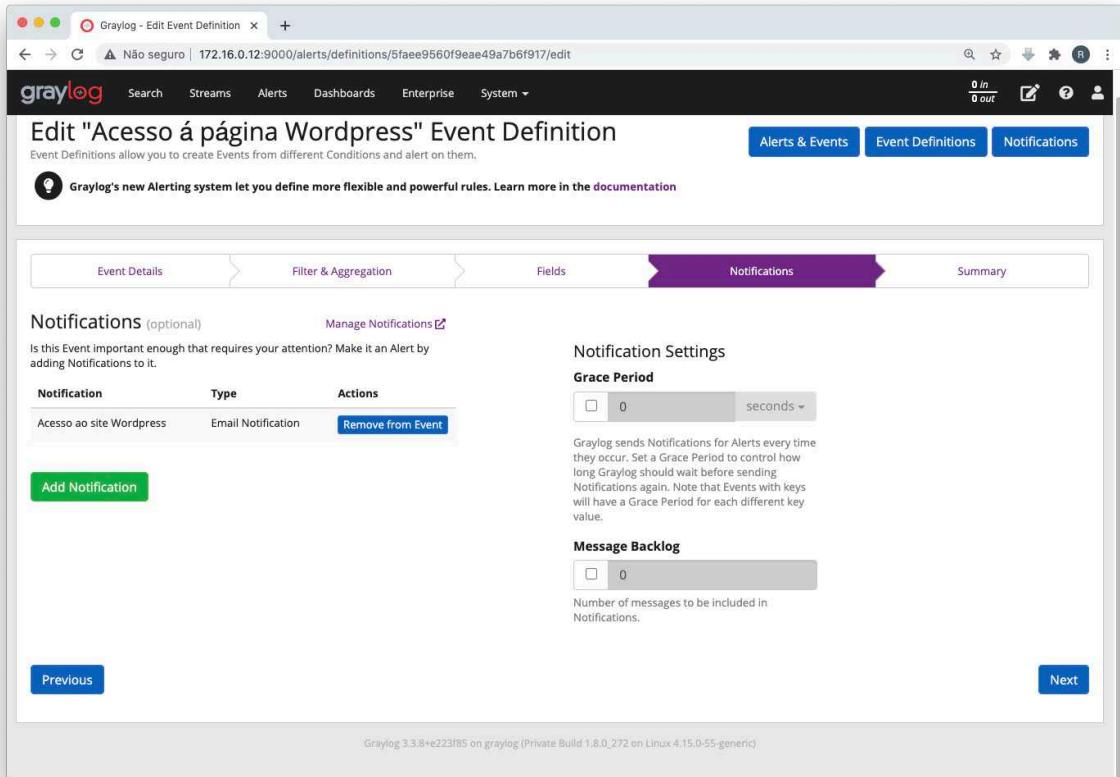


Fig. 4.59: Criar alerta para envio de email - ETAPA 11

Faça um checklist no alerta e clique no botão **Done** para finalizar.

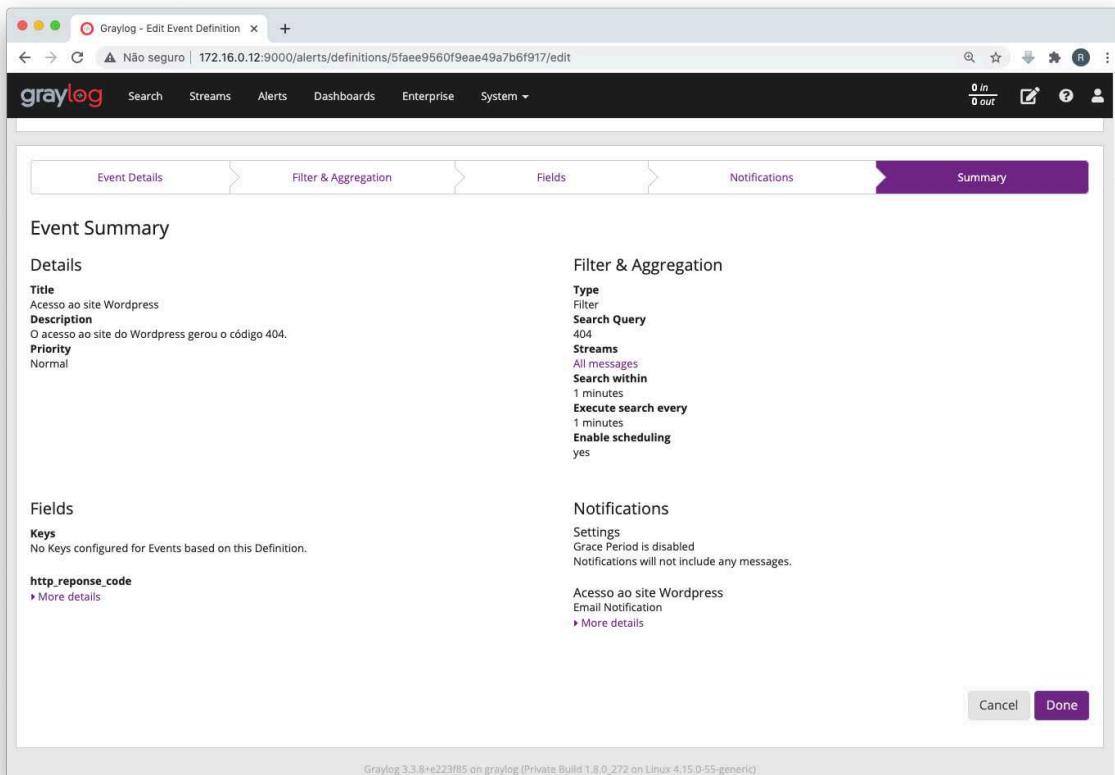


Fig. 4.60: Criar alerta para envio de email - ETAPA 12

3 - Teste do alerta

Para testar um alerta no Graylog, clique em **Alerts > Notifications > More > Teste Notification**

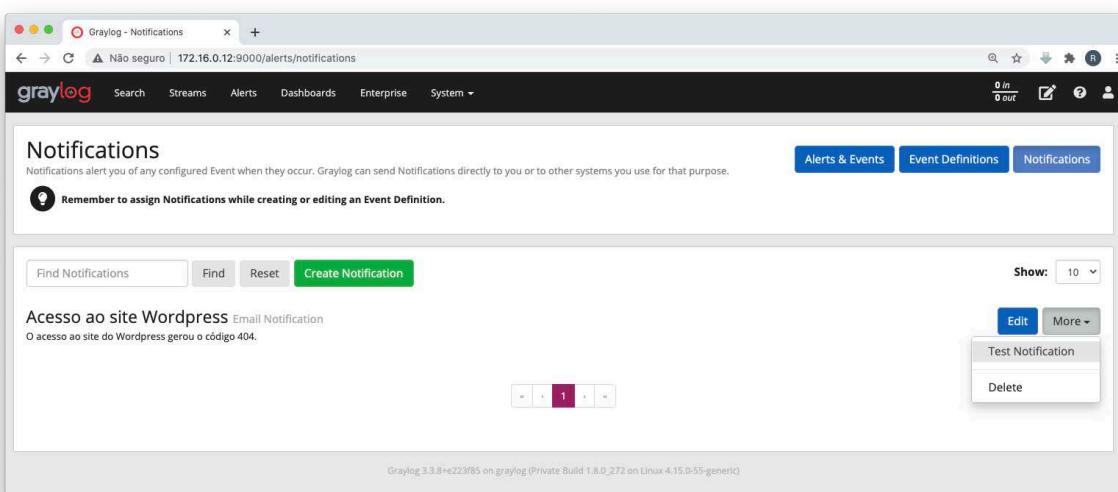


Fig. 4.61: Testando o alerta - ETAPA 1

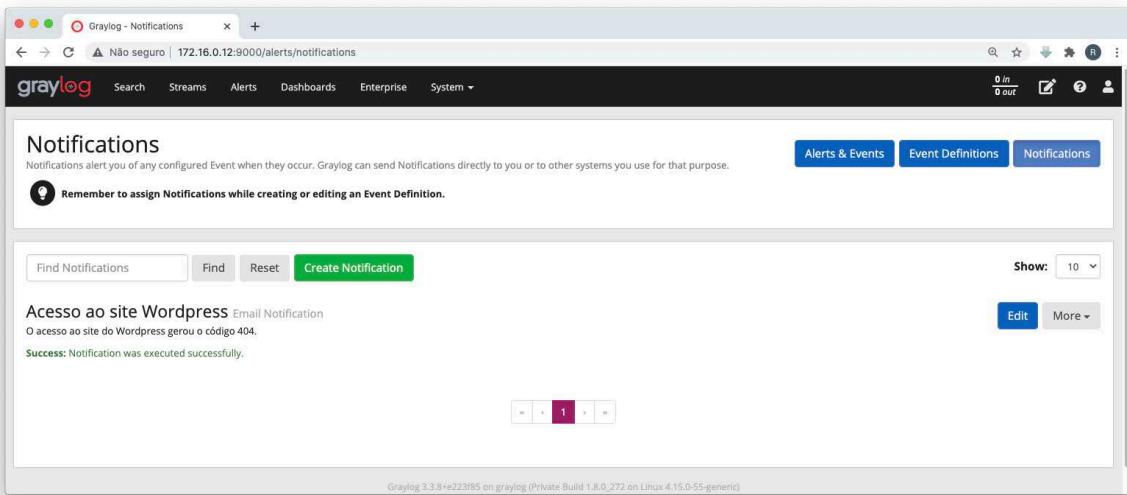


Fig. 4.62: Testando o alerta - ETAPA 2

Se você estiver usando o **Gmail**, é preciso acessar o endereço abaixo e mudar a opção **Permitir aplicativos menos seguros para ATIVADA**

<https://myaccount.google.com/lesssecureapps?>

← Acesso a app menos seguro

Alguns apps e dispositivos usam tecnologias de login menos seguras, o que deixa sua conta vulnerável. Você pode desativar o acesso desses apps, o que recomendamos, ou ativá-lo se optar por usá-los apesar dos riscos. O Google desativará essa configuração automaticamente se ela não estiver sendo usada. [Saiba mais](#)

Permitir aplicativos menos seguros: **ATIVADA**



Fig. 4.63: Testando o alerta - ETAPA 3

4 - Gerar código 404 para testar o alerta

Acesse em seu navegador o endereço <http://wordpress.4labs.example/teste> para gerar o código **404**

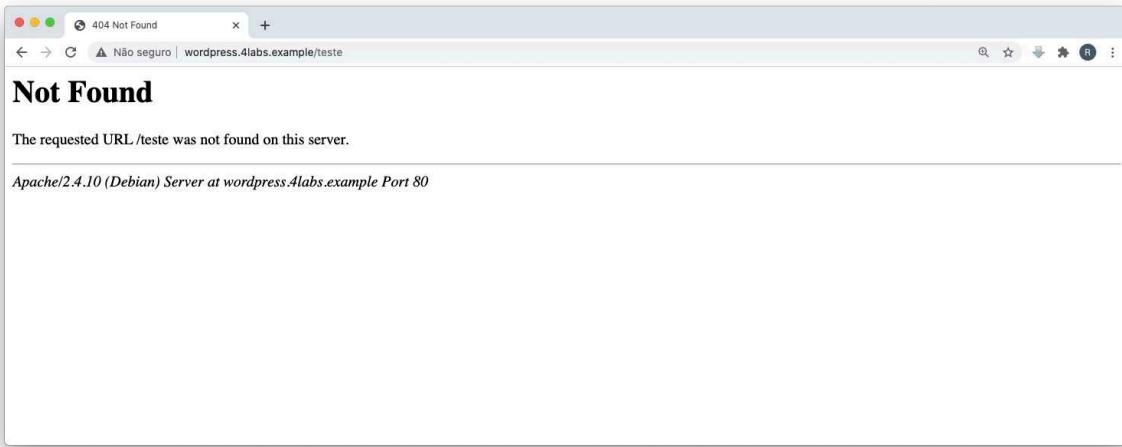


Fig. 4.64: Testando o alerta - ETAPA 4

Aguarde de 1 a 2 minutos e verifique seu email!

Exemplo de email recebido no Gmail:

```
-- [Event Definition] -----
Title: Acesso ao site Wordpress
Description: O acesso ao site do Wordpress gerou o código 404.
Type: aggregation-v1
-- [Event] -----
Timestamp: 2020-11-13T21:46:11.969Z
Message: Acesso ao site Wordpress
Source: graylog
Key:
Priority: 2
Alert: true
Timestamp Processing: 2020-11-13T21:46:11.969Z
Timerange Start:
Timerange End:
Fields:
http_reponse_code: Filter
```

Fig. 4.65: Testando o alerta - ETAPA 5

LAB 4.8 - Criar alertas para envio de mensagens via chat

Vamos criar um alerta para enviar mensagem ao Rocket Chat, quando o acesso a página do Wordpress gerar o código 404.

O Rocket.chat é uma plataforma de bate-papo em equipe de código aberto

1 - Provisionamento do Rocket Chat

Acesse a VM **kibana-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.11
```

Em seguida acesse a pasta **rocketchat-deploy** e provisione o servidor Rocket Chat, através do Docker Compose.

```
1 | cd rocketchat-deploy  
2 | docker-compose up -d
```

Verifique se os containers do **MongoDB** e **Rocket Chat**, estão em execução:

```
1 | docker-compose ps
```

- Resultado:

1	Name	State	Command	Ports
2				
3	suporte_mongo-init-replica_1	... Exit 0	docker-entrypoint.sh bash	
4	suporte_mongo_1	... Up	docker-entrypoint.sh mongo	27017/tcp
5	suporte_rocketchat_1	... Up	docker-entrypoint.sh bash	0.0.0.0:3000->3000/tcp

2 - Configuração do Rocket Chat

Acesse em seu navegador o endereço da VM Webserver na porta 3000:

<http://172.16.0.11:3000>

Passo 1: informação de administração

Preencha as seguintes informações:

- **Nome:** admin
- **Nome do usuário:** admin
- **E-mail da Organização:** admin@4labs.example
- **Senha:** 4linux

Clique no botão **Continuar**

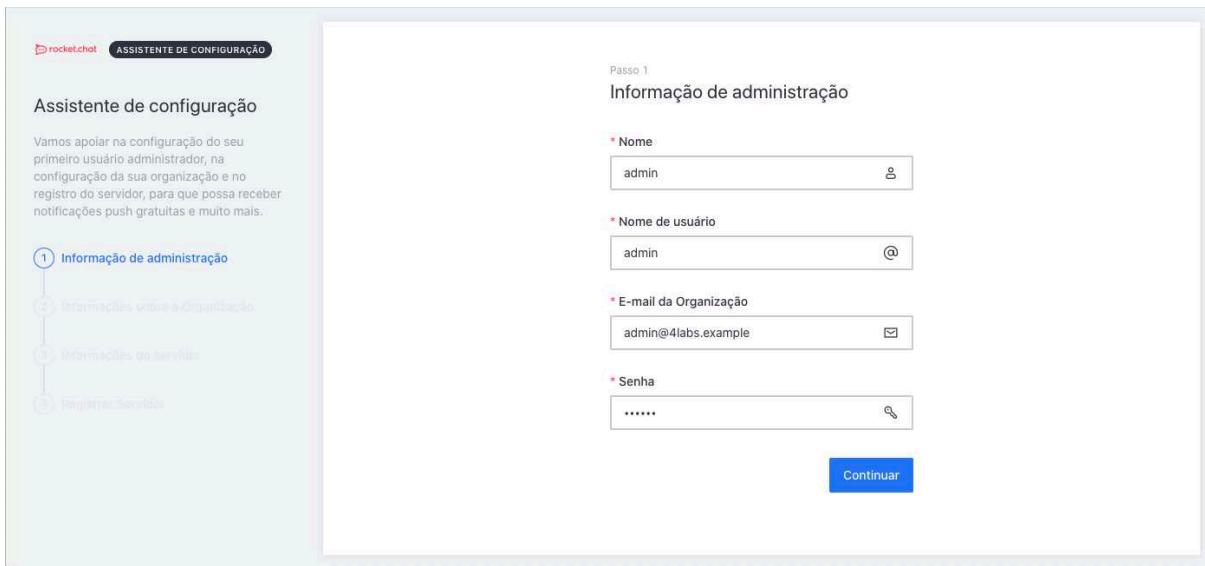


Fig. 4.66: Configurando o Rocket Chat - ETAPA 1

Passo 2: informações sobre a organização (empresa)

Preencha as seguintes informações:

- **Tipo de Organização:** Comunidade
- **Nome da Organização:** 4labs
- **Indústria:** Provedor de tecnologia
- **Tamanho:** 11-50 people
- **País:** Brasil
- **Site:** 4labs.example

Clique no botão **Continuar**

The screenshot shows the second step of the Rocket Chat setup wizard, titled "Passo 2 Informações sobre a Organização". On the left, a sidebar lists steps: 1. Informação de administração (selected), 2. Informações sobre a Organização (current step), 3. Informações do servidor, and 4. Registros do Servidor. The main form fields are:

- Tipo de Organização:** Comunidade
- Nome da Organização:** 4labs
- Indústria:** Provedor de tecnologia
- Tamanho:** 11-50 people
- País:** Brasil
- Site:** 4labs.example

A blue "Continuar" button is at the bottom right.

Fig. 4.67: Configurando o Rocket Chat - ETAPA 2

Passo 3: informações do servidor

Preencha as seguintes informações:

- **Tipo de Organização:** Comunidade
- **Nome do Site:** Rocket.Chat
- **Idioma:** Português do Brasil
- **Tipo de servidor:** Comunidade
- **Auto ativar a autenticação de duas etapas via email para novos usuários:** Sim

Clique no botão **Continuar**

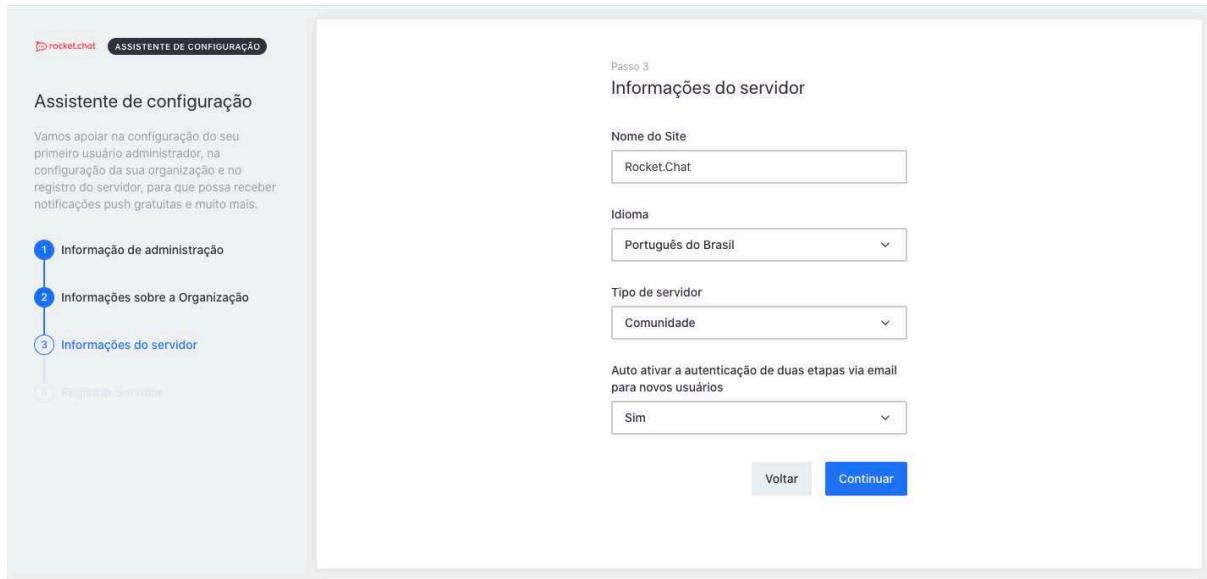


Fig. 4.68: Configurando o Rocket Chat - ETAPA 3

Passo 4: registrar servidor

Selecione a opção **Mantenha-se autônomo, você precisará** e clique no botão **Continuar**

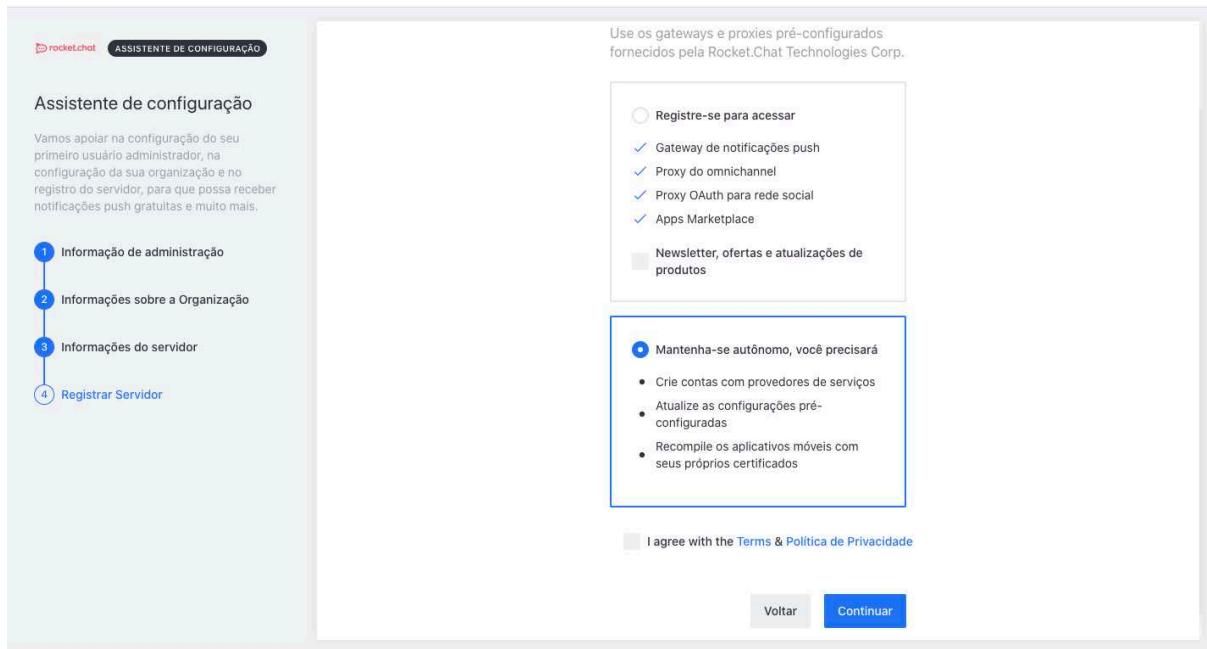


Fig. 4.69: Configurando o Rocket Chat - ETAPA 4

Para terminar a configuração, clique no botão **Vá para o seu espaço de trabalho**

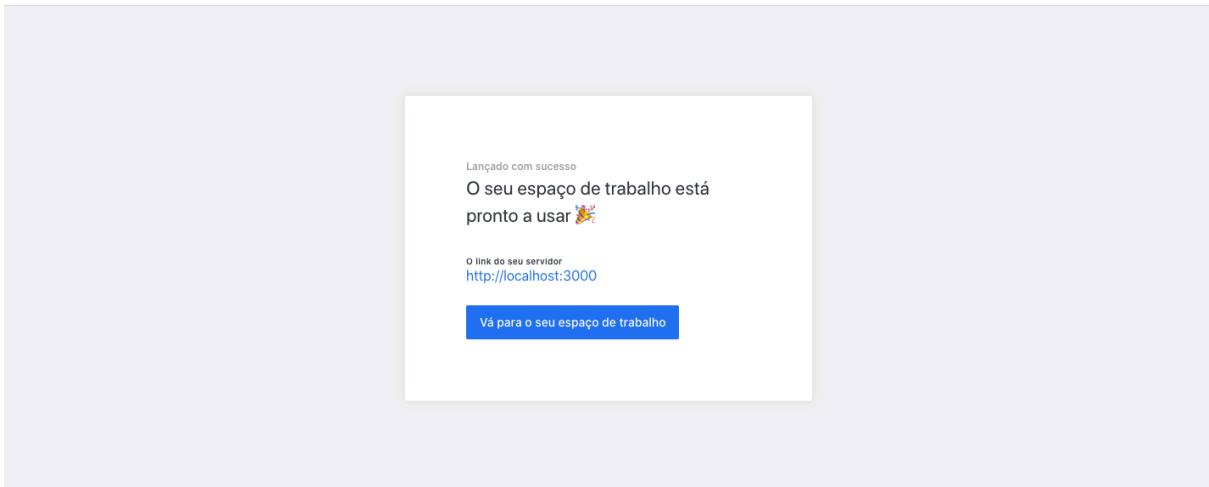


Fig. 4.70: Configurando o Rocket Chat - ETAPA 5

E pronto! Você agora possui uma plataforma de comunicação de equipes de código aberto.

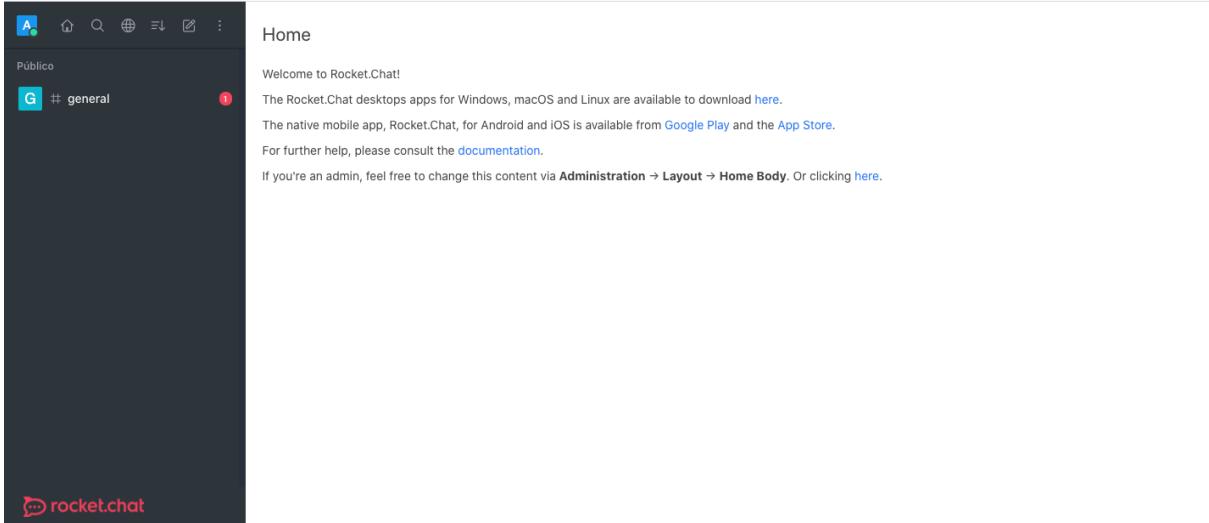


Fig. 4.71: Configurando o Rocket Chat - ETAPA 6

3 - Criar canal no Rocket Chat

Vamos criar um canal (sala de bate papo) para receber as mensagens do Graylog. Clique no ícone do lápis > # Canal

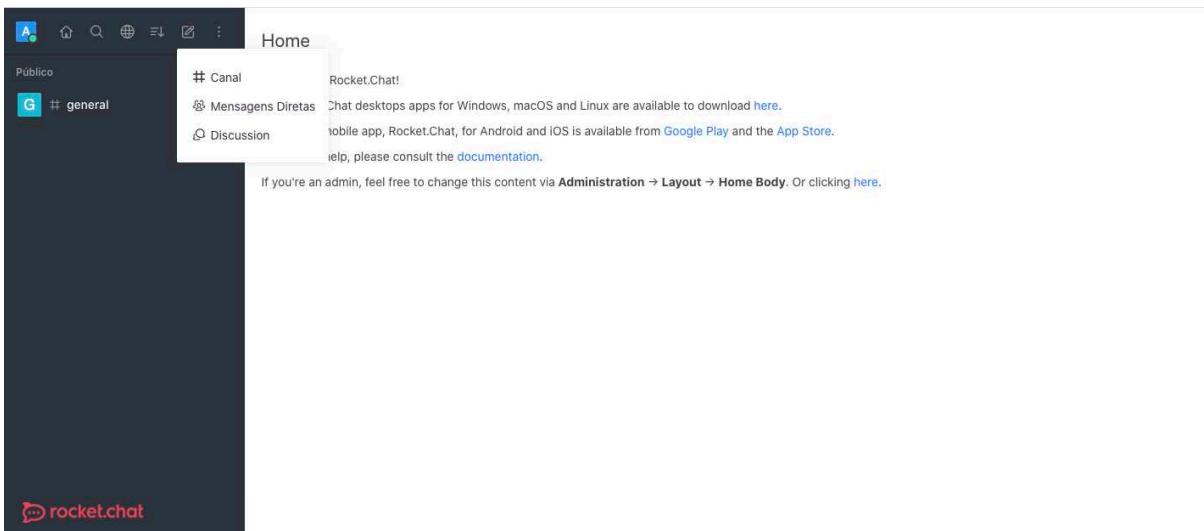


Fig. 4.72: Configurando o Rocket Chat - ETAPA 7

Selecione e preencha as seguintes informações:

- **Canal público:** Não
- **Canal Somente Leitura:** Não
- **Canal de Transmissão:** Não
- **Nome do Canal:** graylog
- **Convidar Usuários:** rocket.cat

Clique no botão **Criar**

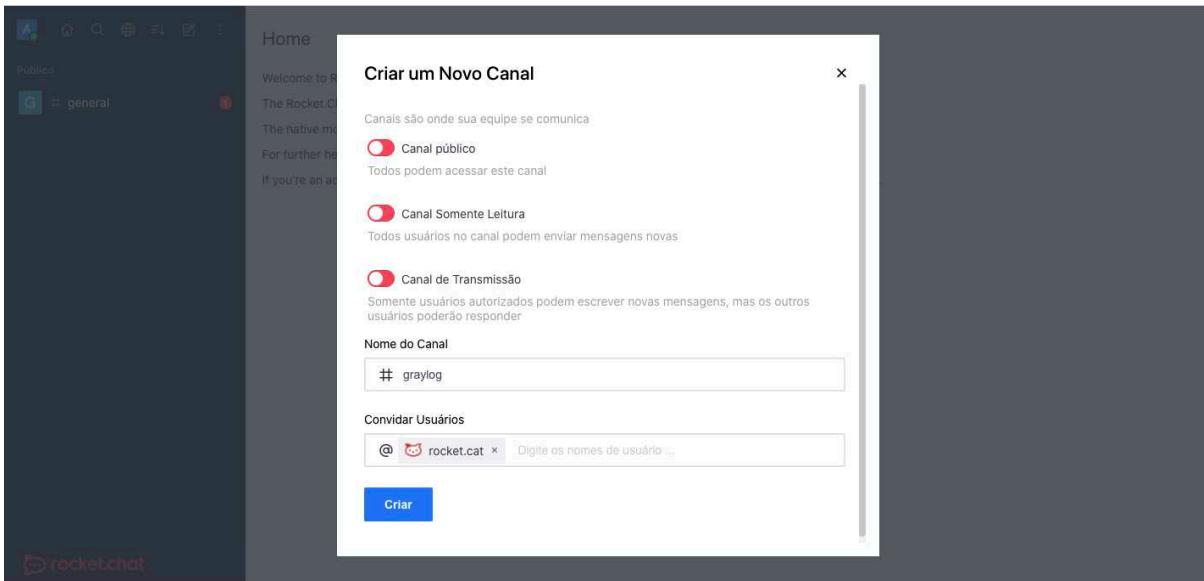


Fig. 4.73: Configurando o Rocket Chat - ETAPA 8

Confirme que o canal **graylog** foi criado.

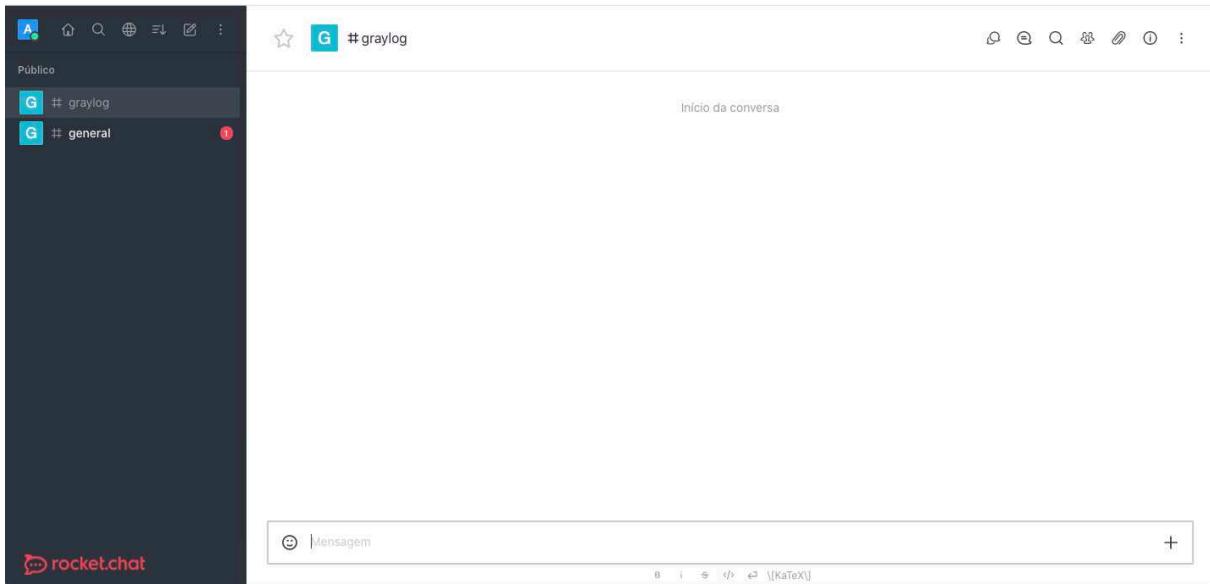


Fig. 4.74: Configurando o Rocket Chat - ETAPA 9

4 - Criar integração WebHook no Rocket Chat

Vamos criar uma integração WebHook para receber as mensagens do Graylog. Clique no ícone de três pontos > Administração

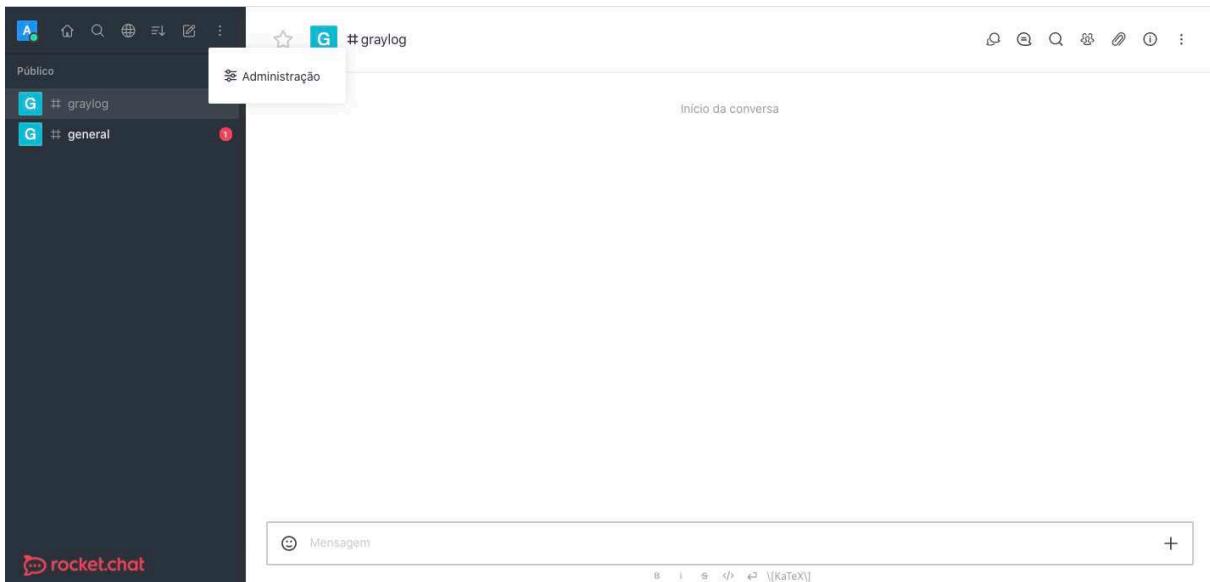


Fig. 4.75: Criar integração WebHook no Rocket Chat - ETAPA 1

Em seguida clique em **Integrações** > Incoming > botão + New

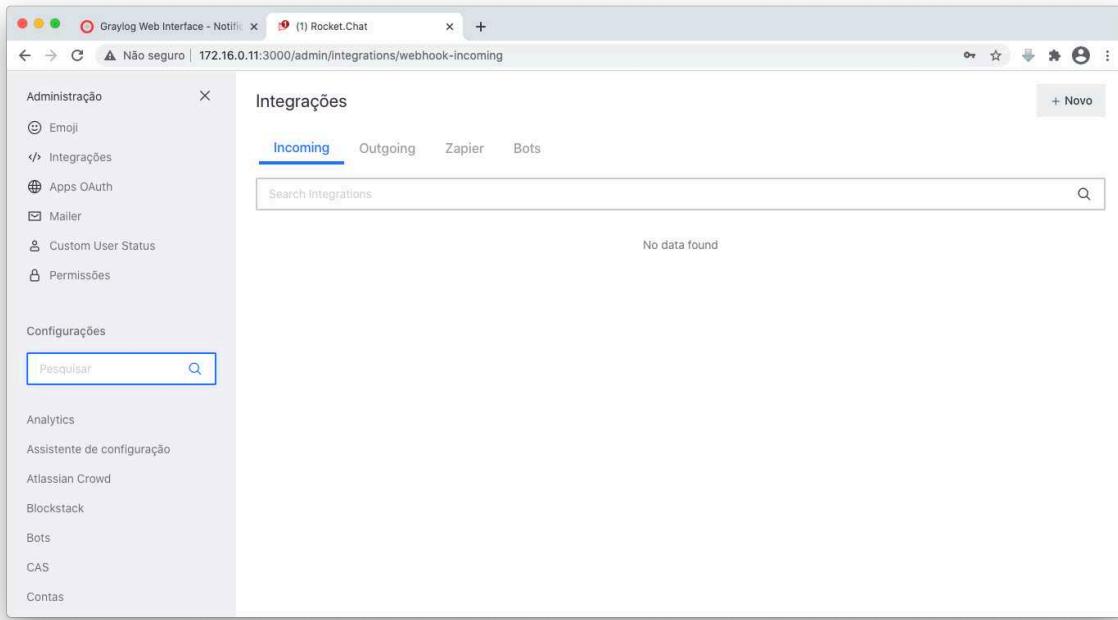


Fig. 4.76: Criar integração WebHook no Rocket Chat - ETAPA 2

Selecione e preencha as seguintes informações:

- **Ativado:** Sim
- **Nome (opcional):** alertgraylog
- **Postar no Canal:** #graylog
- **Postar como:** rocket.cat
- **Apelido (opcional):** Graylog Alert

Fig. 4.77: Criar integração WebHook no Rocket Chat - ETAPA 3

Ainda na janela de configuração do WebHook, selecione a opção **Script Ativado** e adicione na janela **Script**, o conteúdo do script no seguinte endereço:

<https://raw.githubusercontent.com/jeanmorais/rocketchat-graylog-hook/master/graylog-rocketchat.hooks.js>

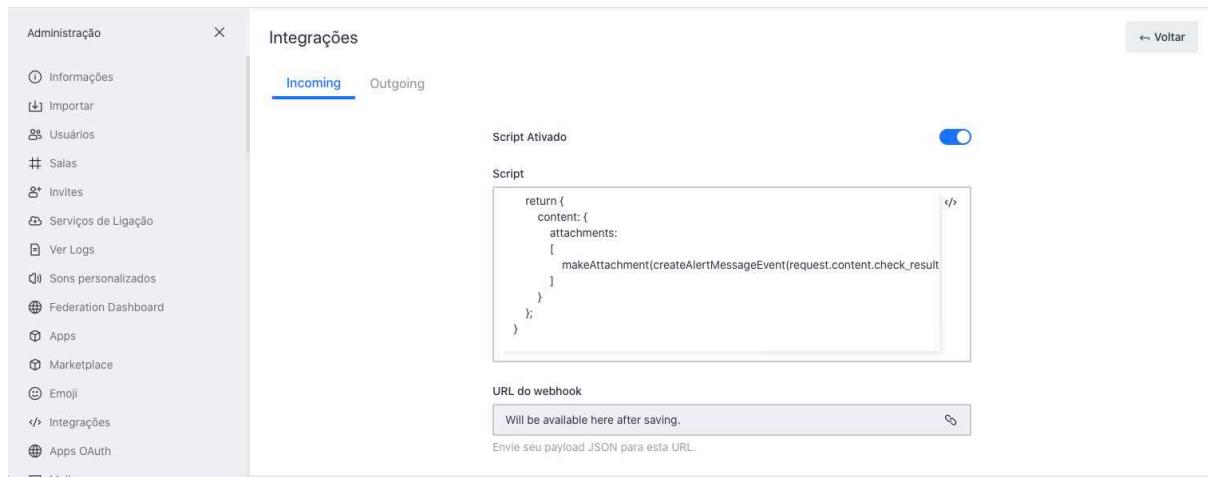


Fig. 4.78: Criar integração WebHook no Rocket Chat - ETAPA 4

Clique no botão **Salvar** para concluir a criação do WebHook.

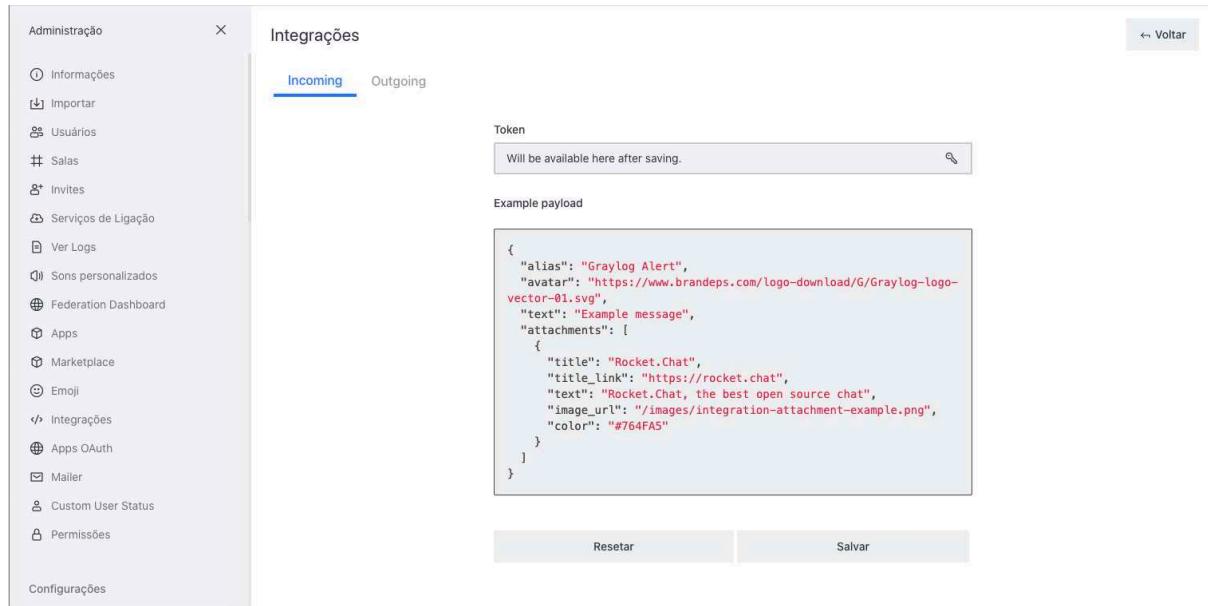


Fig. 4.79: Criar integração WebHook no Rocket Chat - ETAPA 5

Após salvar o WebHook, uma URL será criada na caixa **URL do webhook**. Copie o endereço completo!

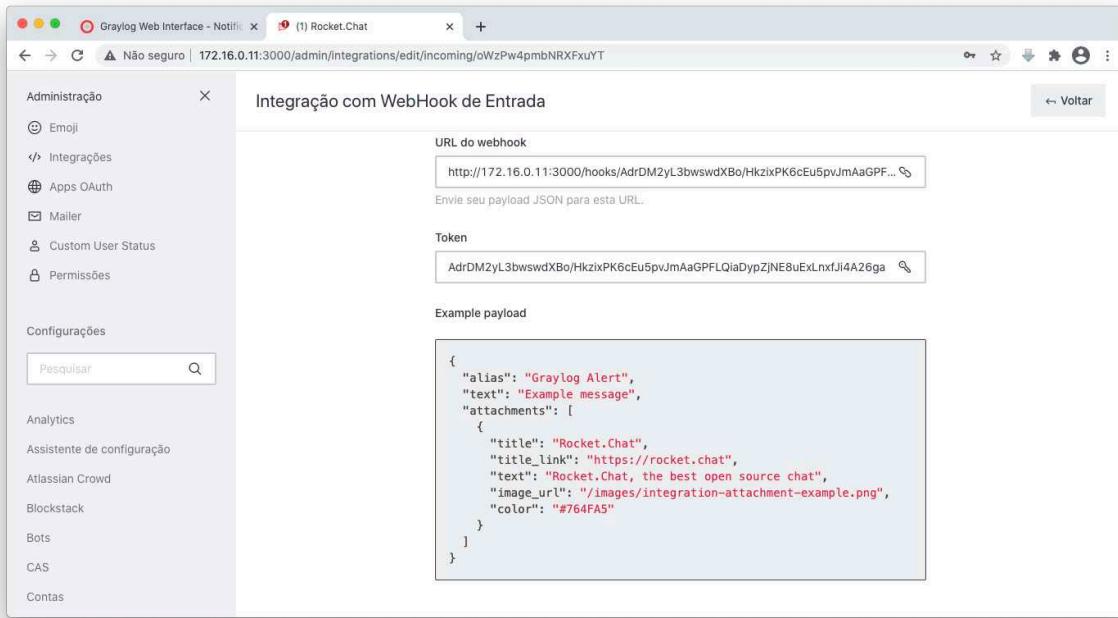


Fig. 4.80: Criar integração WebHook no Rocket Chat - ETAPA 6

5 - Criar notificação via Rocket Chat no Graylog

Antes de adicionar uma notificação via Rocket Chat, é preciso adicionar a URL de webhook como endereço confiável no Graylog. Clique no menu **System > Configurations > URL Whitelist Configuration >** botão **Update**.

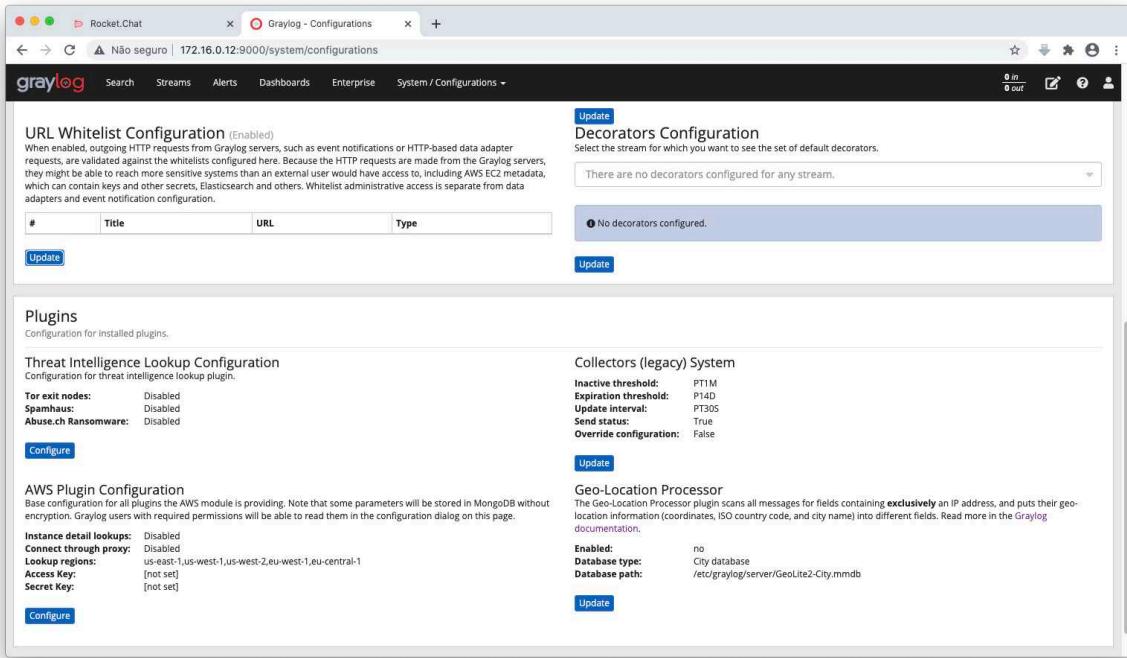


Fig. 4.81: Criar notificação via Rocket Chat no Graylog - ETAPA 1

Clique no botão **Add Url** e preencha as seguintes informações:

- **Title:** Acesso ao site Wordpress
- **URL:** Cole aqui a URL do webhook que você copiou no Rocket Chat

Clique no botão **Save** para continuar.

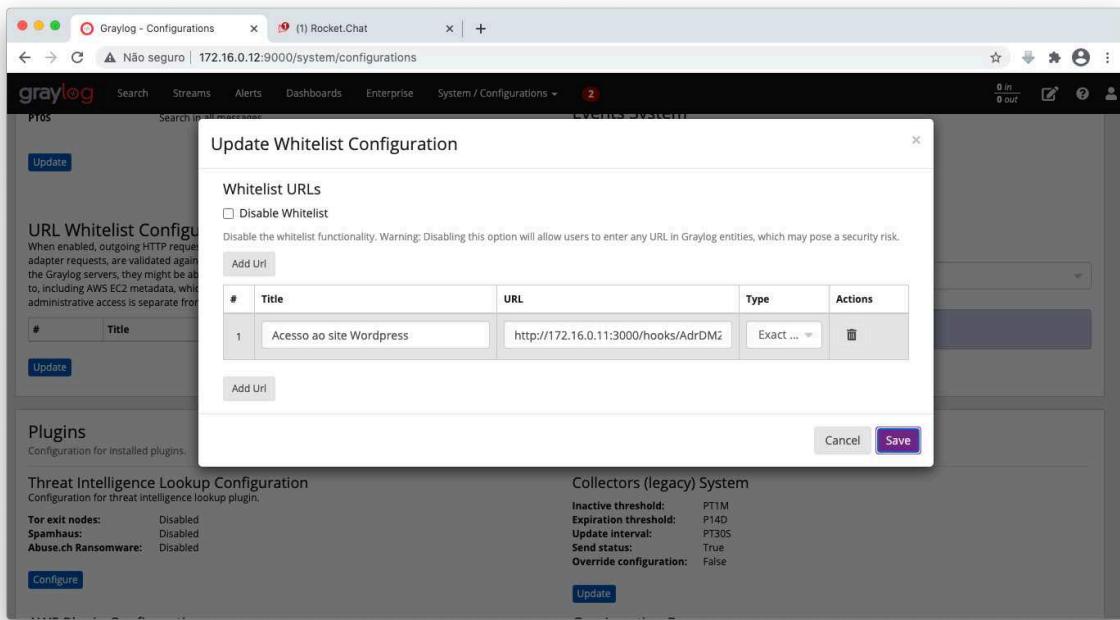


Fig. 4.82: Criar notificação via Rocket Chat no Graylog - ETAPA 2

Confirme a adição da URL do Webhook.

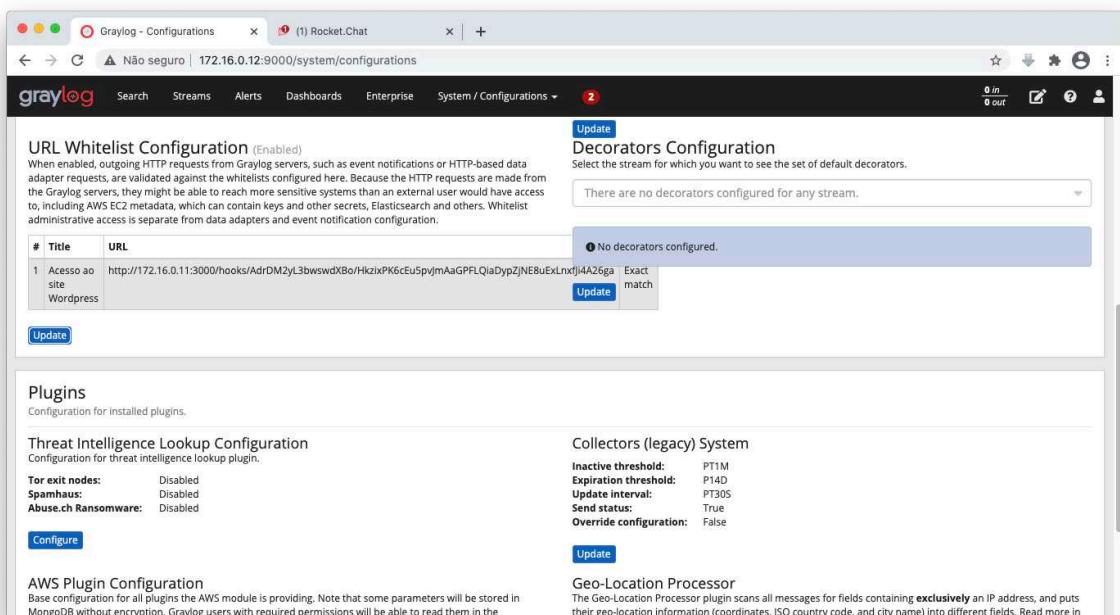


Fig. 4.83: Criar notificação via Rocket Chat no Graylog - ETAPA 3

6 - Configurar notificação via Rocket Chat no Graylog

Vamos adicionar uma notificação via Rocket Chat no Graylog. Clique no menu **Alerts > Notifications** > botão **Create Notification**.

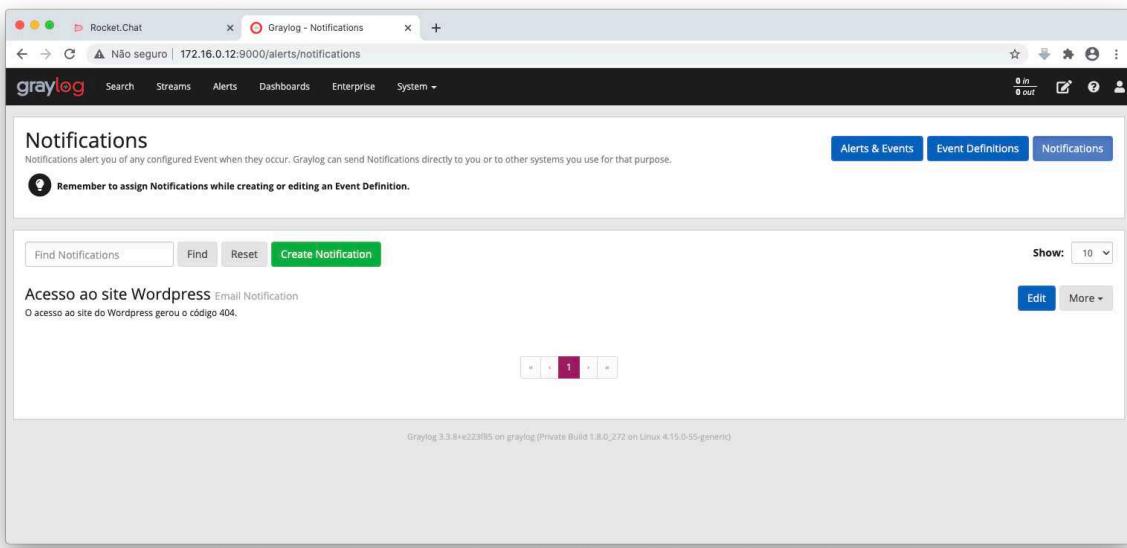


Fig. 4.84: Configurar notificação via Rocket Chat no Graylog - ETAPA 1

E preencha as seguintes informações:

- **Title:** Acesso ao site Wordpress - Chat
- **Description:** O acesso ao site do Wordpress gerou o código 404.
- **Notification Type:** Legacy Alarm Callbacks
- **Choose Legacy Notification:** Legacy HTTP Alarm Callback
- **URL:** Cole aqui a URL do webhook que você copiou no Rocket Chat

Clique no botão **Create** para continuar.

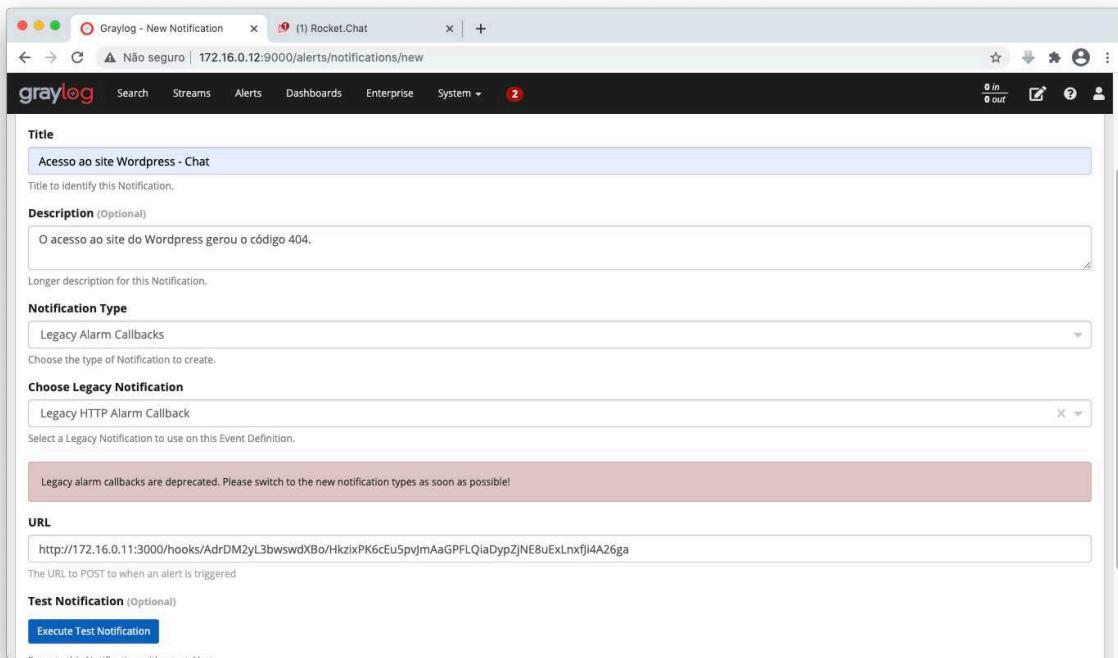


Fig. 4.85: Configurar notificação via Rocket Chat no Graylog - ETAPA 2

Para testar a notificação **Acesso ao site Wordpress - Chat**, clique em **More > Test Notification**.

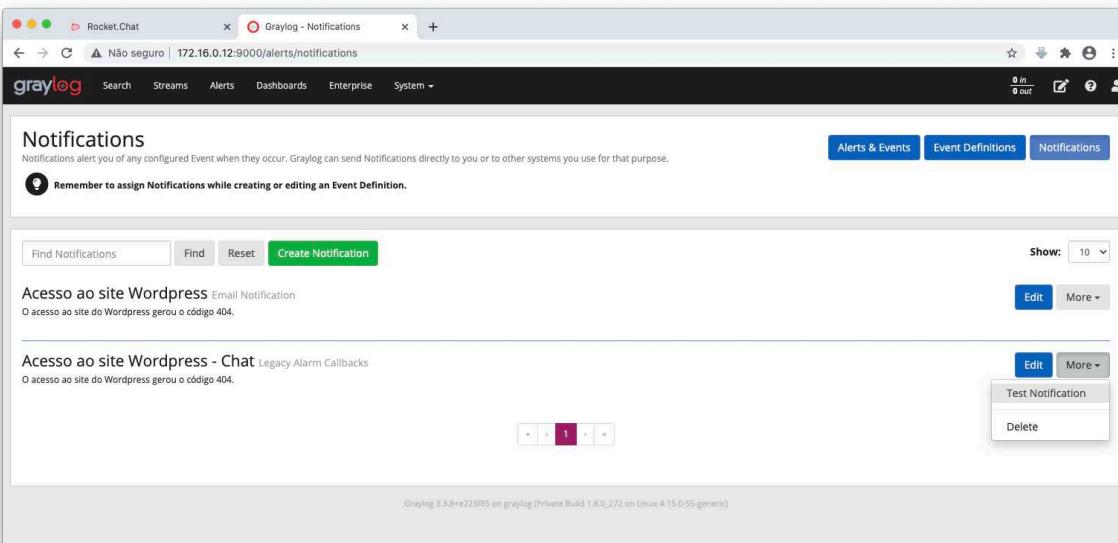


Fig. 4.86: Configurar notificação via Rocket Chat no Graylog - ETAPA 3

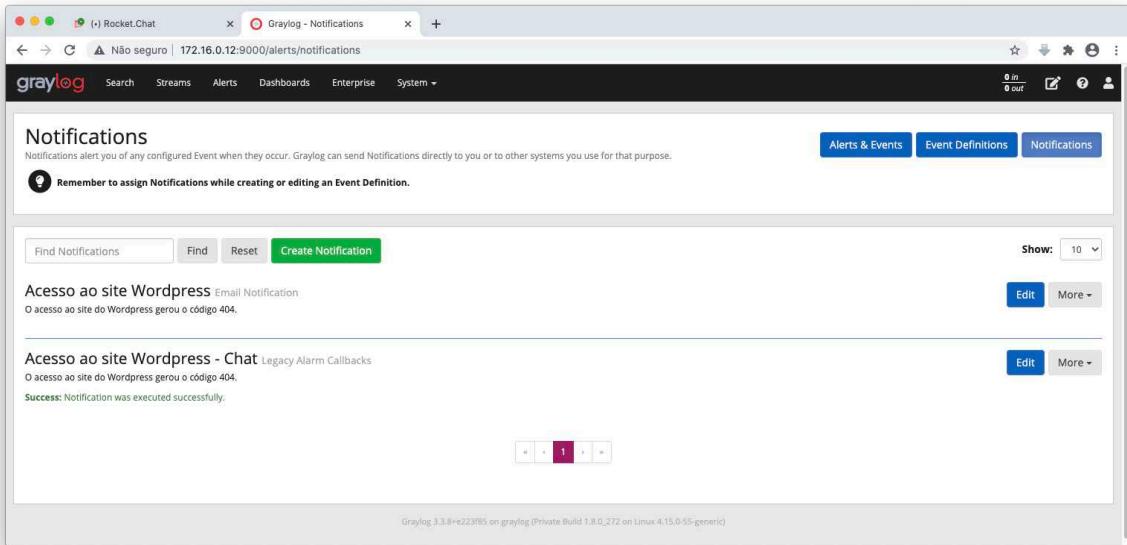


Fig. 4.87: Configurar notificação via Rocket Chat no Graylog - ETAPA 4

Alterne para o Dashboard do Rocket Chat e verifique se a mensagem de teste foi entregue.

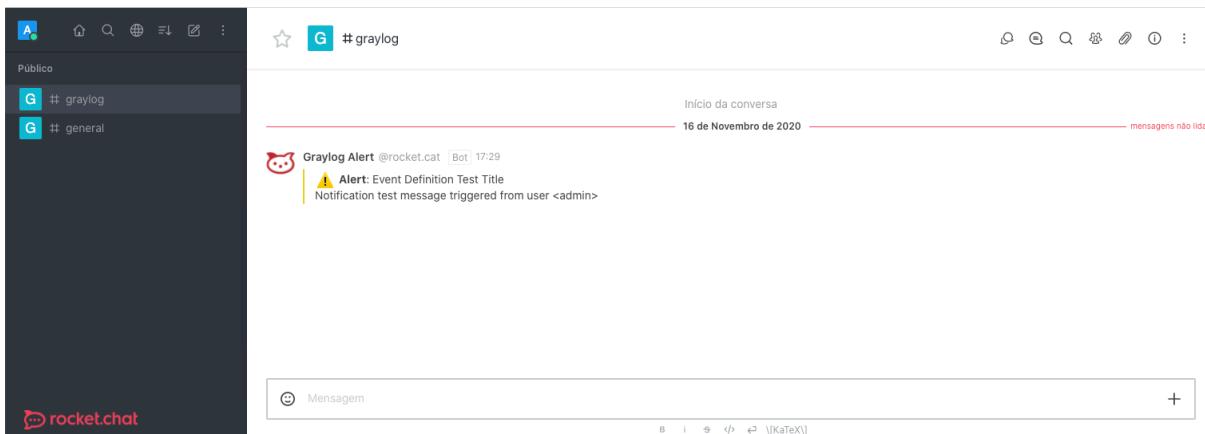


Fig. 4.88: Configurar notificação via Rocket Chat no Graylog - ETAPA 5

7 - Atualizar alerta no Graylog

Vamos atualizar o alerta que criamos no Graylog, adicionando a notificação via Rocket Chat. Clique no menu **Alerts > Event Definitions > botão Edit > botão Add Notification**

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

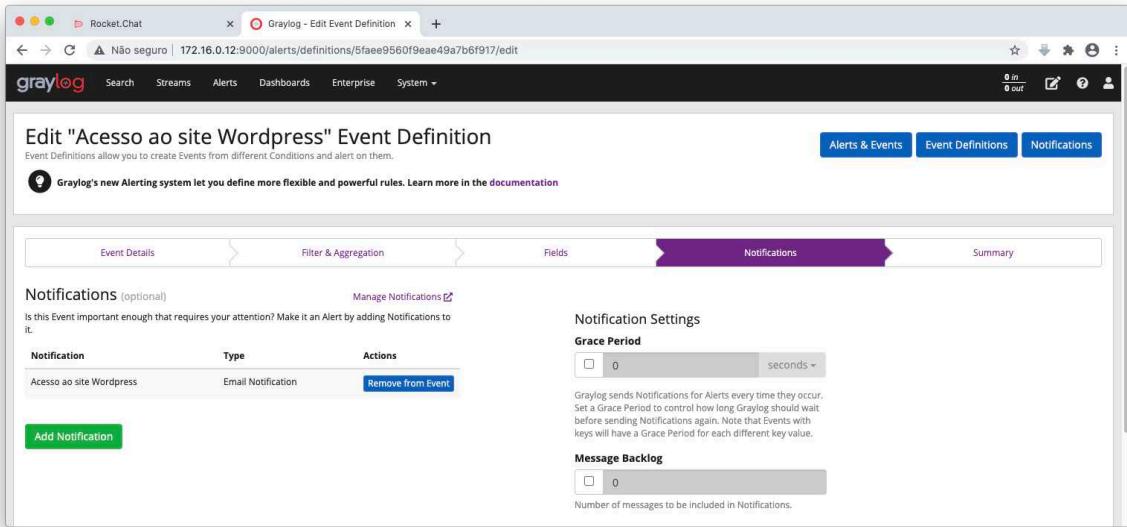


Fig. 4.89: Atualizar alerta no Graylog - ETAPA 1

Na caixa **Choose Notification** selecione **Acesso ao site Wordpress - Chat** > botão **Done**

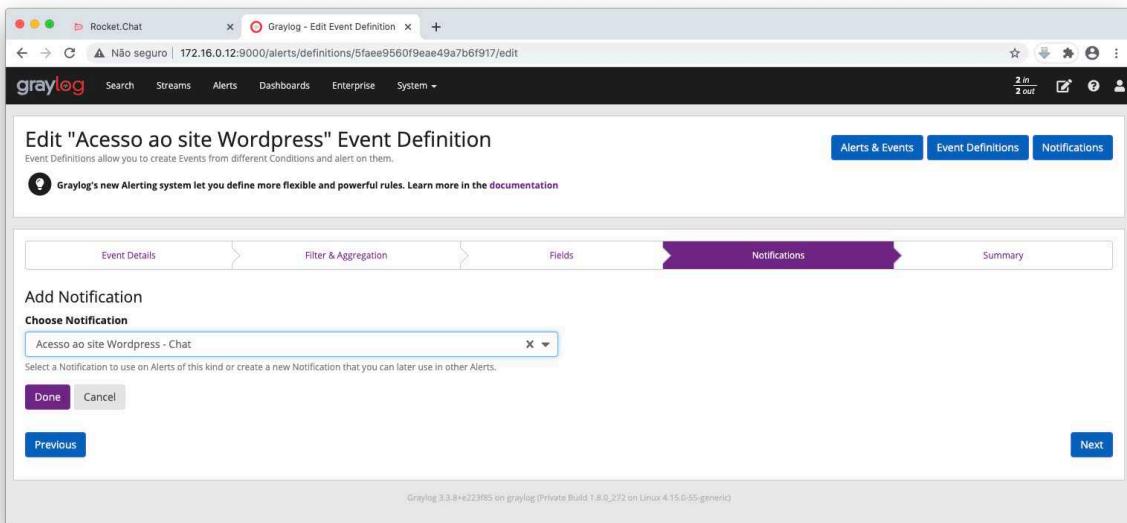


Fig. 4.90: Atualizar alerta no Graylog - ETAPA 2

Confirme a nova notificação adicionada.

4. CENTRALIZAÇÃO DE LOGS COM GRAYLOG

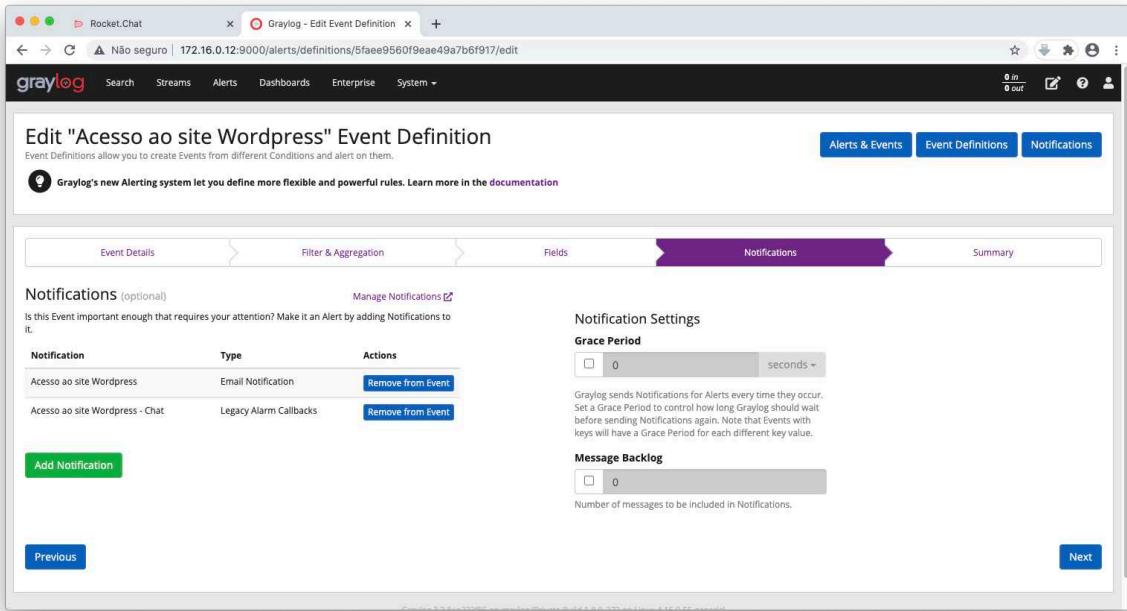


Fig. 4.91: Atualizar alerta no Graylog - ETAPA 3

Antes de continuar, selecione o tempo de **5 segundos** para envio da mensagem e clique no botão **Next**.

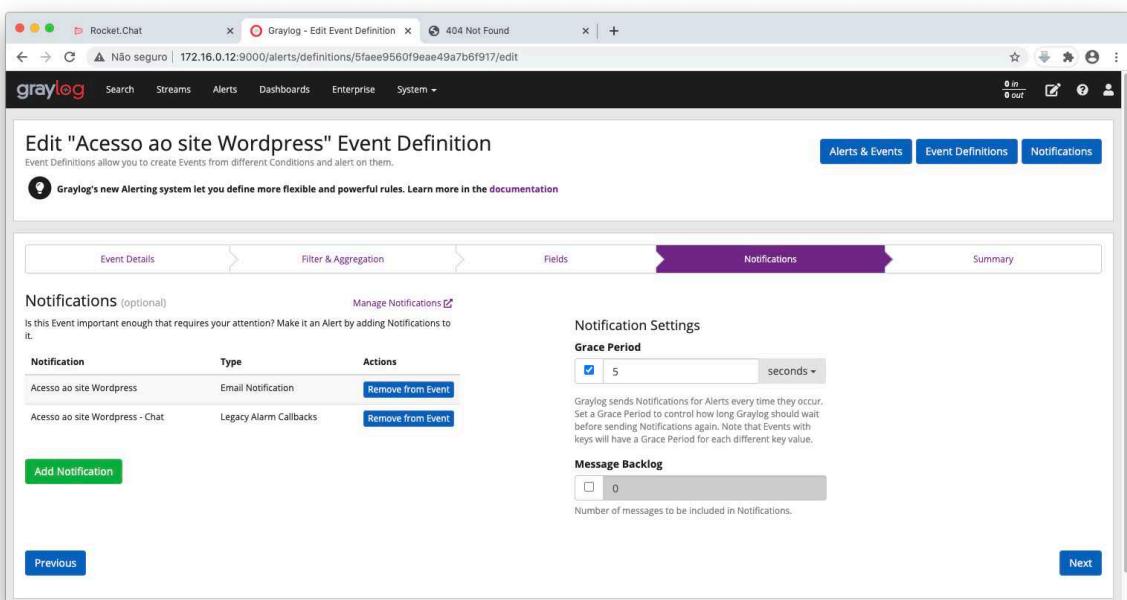


Fig. 4.92: Atualizar alerta no Graylog - ETAPA 4

Faça um checklist no alerta e clique no botão **Done** para finalizar.

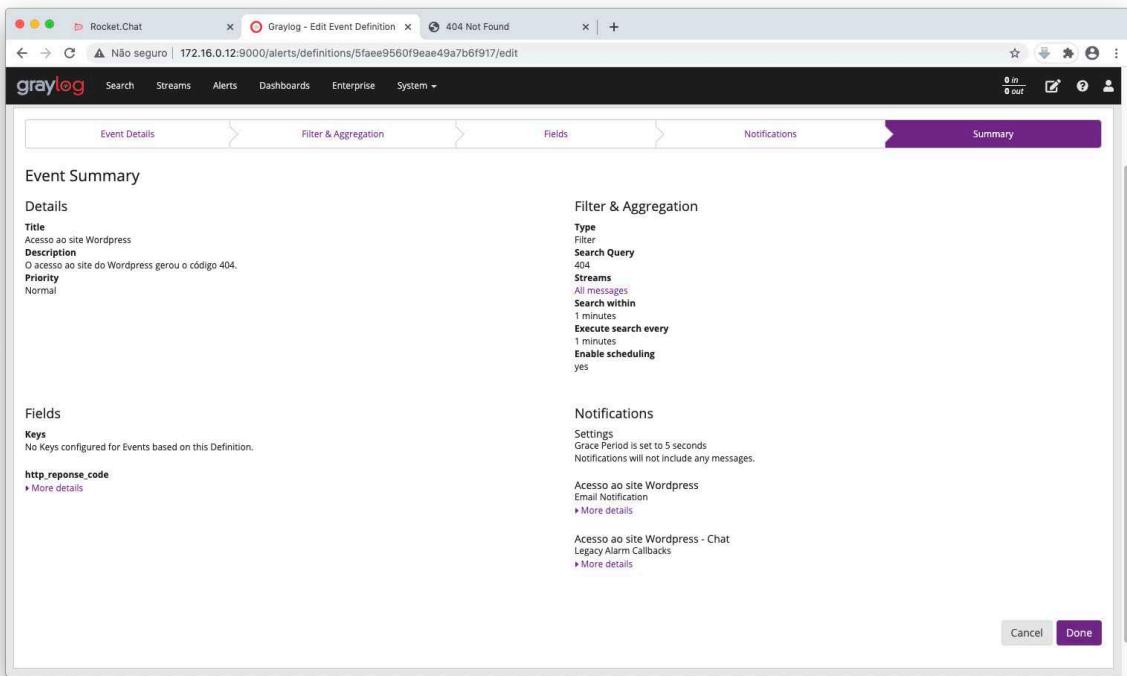


Fig. 4.93: Atualizar alerta no Graylog - ETAPA 5

Para testar a integração entre o Graylog e Rocket Chat, acesse em seu navegador o endereço <http://wordpress.4labs.example/teste> para gerar o código **404**

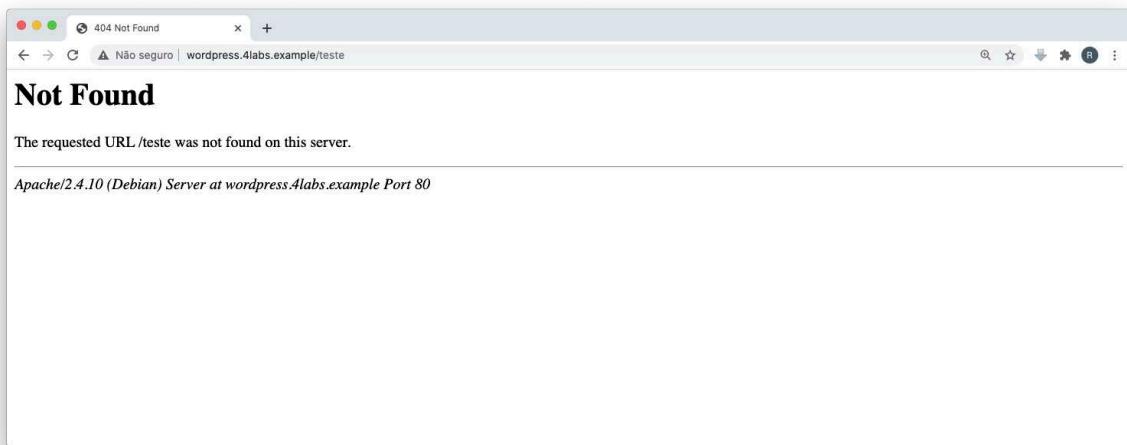


Fig. 4.94: Atualizar alerta no Graylog - ETAPA 6

Aguarde 1 minuto e verifique no Rocket Chat, a mensagem enviada!

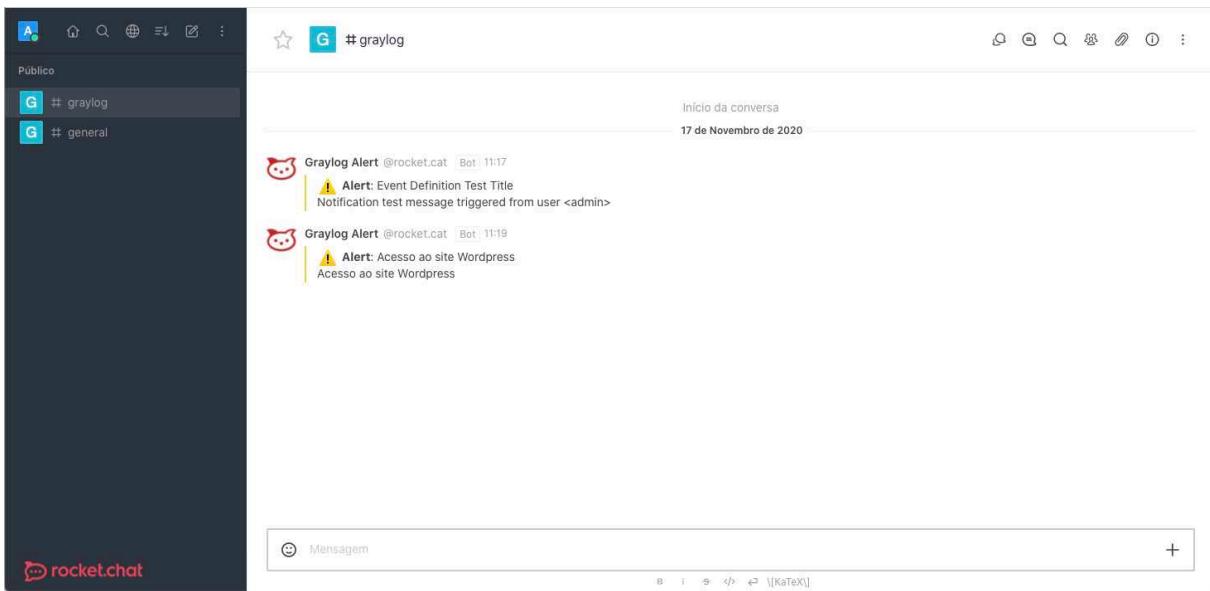


Fig. 4.95: Atualizar alerta no Graylog - ETAPA 7

5

Centralização de logs com ELK

Competências deste conteúdo

- Características do Elastic Stack
- Instalação e configuração da pilha ELK (Elastic, Logstash e Kibana)
- Filtro de logs com o Logstash
- Utilizando o FileBeat para envio de arquivos logs
- Realização de buscas
- Criação de dashboards

Características do Elastic Stack

Componentes e características

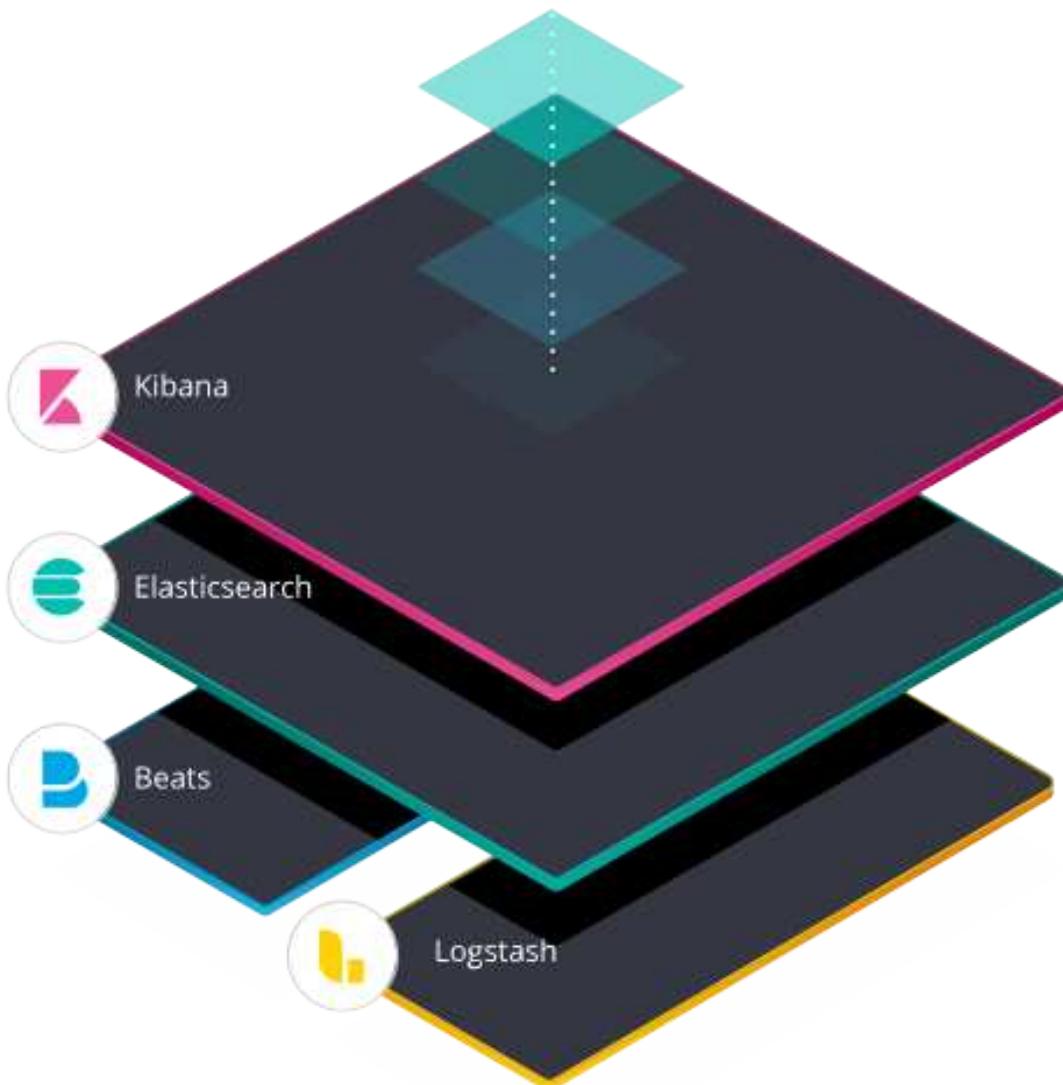


Fig. 5.1: Elastic Stack

O **Elastic Stack**, anteriormente conhecido como **elk stack** é um conjunto de três projetos Open Source: **Elasticsearch**, **Logstash** e **Kibana**.

Em 2015 foi introduzido o **Beats** e a solução passou a ser conhecida como **ELKB**.

Hoje em dia quando falamos sobre **ELKB**, **ELK Stack** ou **Elastic Stack** estamos nos referindo a mesma solução.

*É possível encontrar referencias ao **ELKB** em outros acrônimos como o **ELKB**, **ELBK**, **LKEB** etc.*

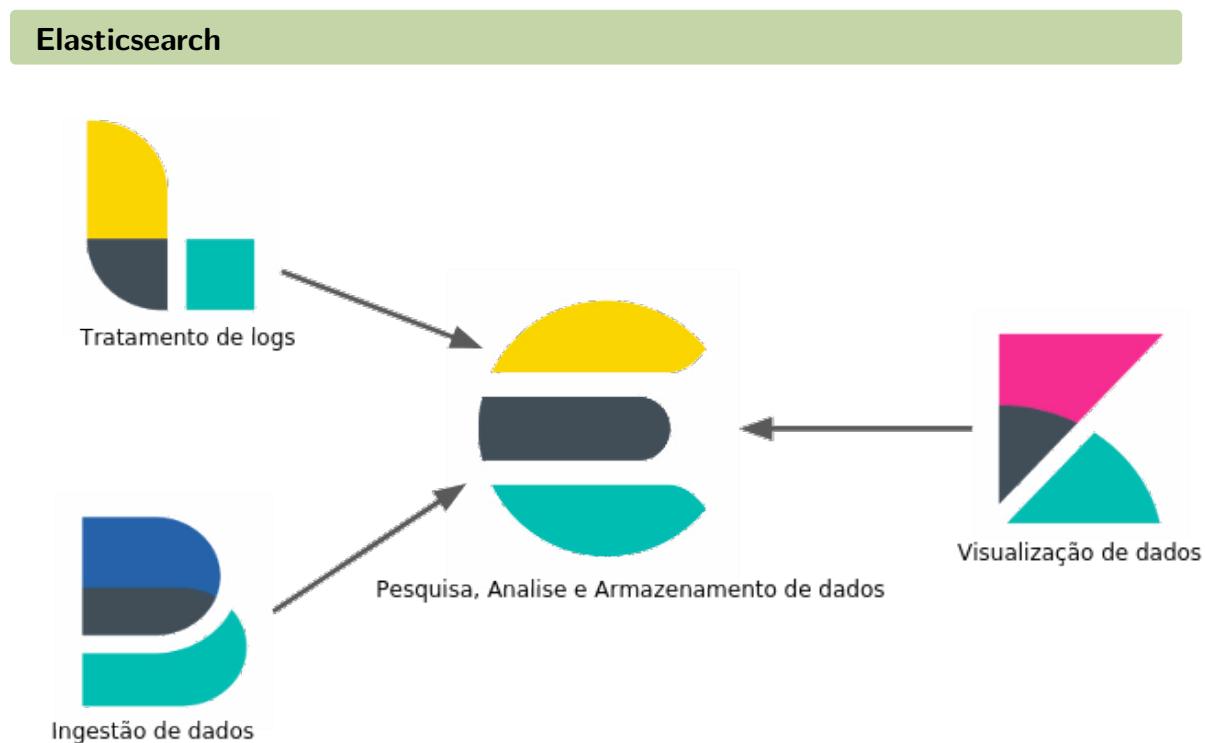


Fig. 5.2: Elastic Stack

O Elastic Stack é uma solução completa com a integração de todas as ferramentas.



Fig. 5.3: Elasticsearch

O Elasticsearch é um motor de pesquisa por texto completo e analítico muito utilizado por habitar a funcionalidade de pesquisa em diferentes aplicações e é o coração do Elastic Stack.

Podemos utilizar o Elasticsearch por exemplo em um blog, no qual desejamos que os usuários possam pesquisar por vários tipos de dados que poderiam ser posts, produtos, categorias etc.

Através do Elasticsearch podemos implementar uma pesquisa por texto completo levando em conta outros fatores como tamanho de posts ou avaliações.

Tecnicamente o Elasticsearch pode fazer tudo que gostaríamos de ter em uma poderosa ferramenta de pesquisa e não ficar limitado a apenas pesquisas de texto completo. É possível escrever consultas para dados estruturados e utilizá-los para criar gráficos utilizando o Elasticsearch como plataforma de análise

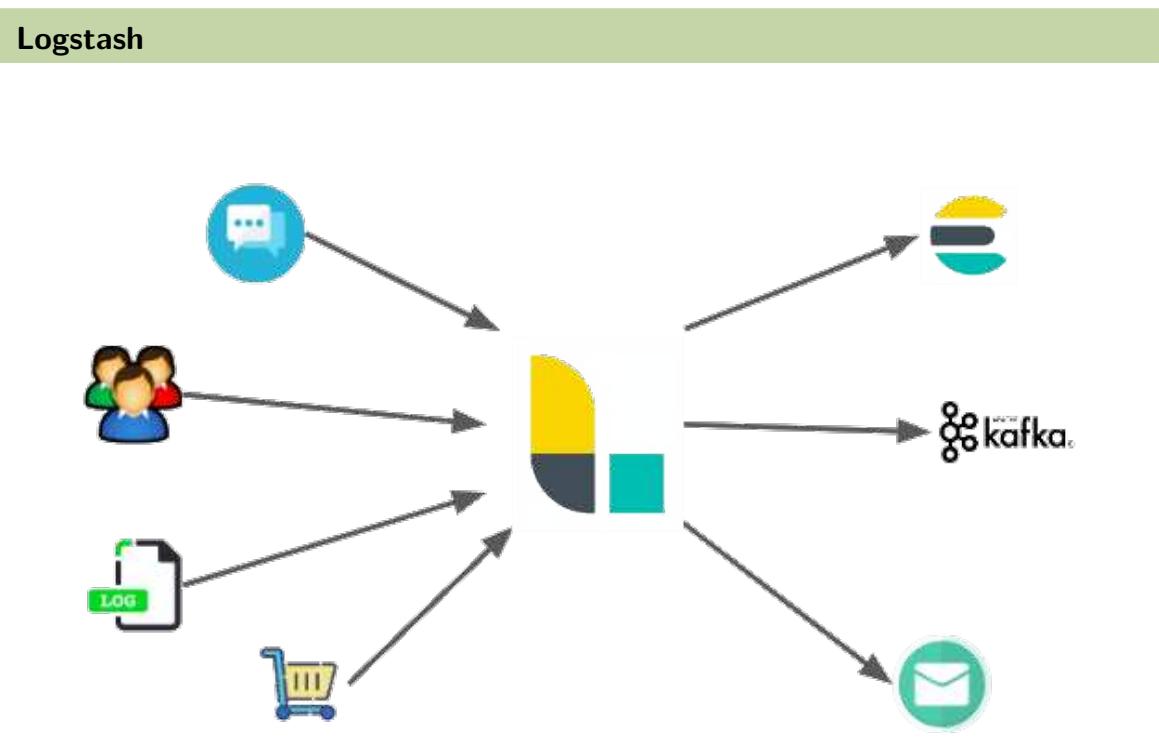


Fig. 5.4: Logstash

O Logstash é utilizado para processar os logs de aplicações e enviá-los para o Elasticsearch.

Os dados coletados pelo logstash são gerenciados como eventos que podem ser de qualquer tipo, como por exemplo, entradas de arquivos, pedidos de e-commerce, clientes, mensagens de chat etc.

O logstash desempenha seu papel através de 3 estágios:

- **Entrada** – Como o logstash recebe os dados, pode ser um endpoint HTTP, uma base de dados relacional ou até mesmo um arquivo.
- **Filtro** – Como o logstash processa os eventos recebidos na etapa de Entrada, aqui podemos analisar arquivos no formato CSV, XML ou até mesmo JSON. Também é possível fazer o enriquecimento dos dados.
- **Saída** – Para onde são enviados os dados processados, podem ser uma instância do Elasticsearch, uma base de dados, um arquivo etc.

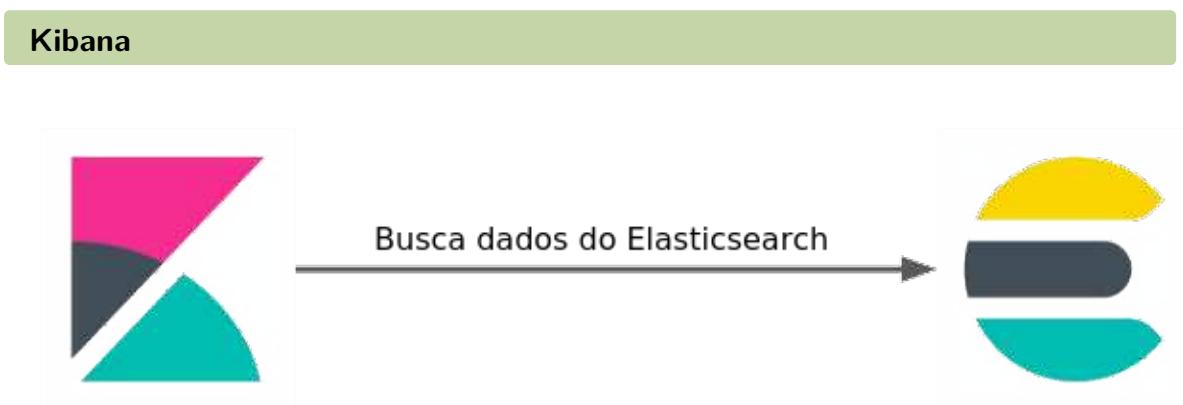


Fig. 5.5: Kibana - Elasticsearch

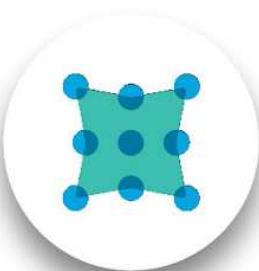
O Kibana é uma plataforma de análise e visualização de dados que possibilita visualizar os dados do Elasticsearch e analisá-los. Podemos dizer que o Kibana é um Dashboard do Elasticsearch onde podemos criar visualizações e também gráficos.

O Kibana fornece uma interface para construir pesquisas e configurar a exibição de resultados através dos dados do Elasticsearch e envia solicitações para o Elasticsearch através de uma REST API.

Existe uma website da própria Elastic com um DEMO online do dashboard do kibana com exemplos de dashboards.

FileBeat

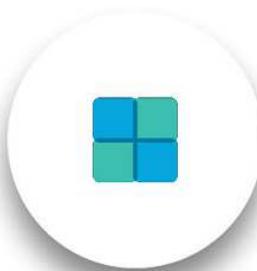
The Beats family

**Packetbeat**

Network data

**Metricbeat**

Metrics

**Winlogbeat**

Windows Event Logs

**Auditbeat**

Audit data

**Filebeat**

Log files

**Heartbeat**

Uptime monitoring

Fig. 5.6: Beats

O Beats é uma coleção de exportadores de dados, que são agentes com algum propósito em particular.

Existem diversos tipos de exportadores de dados e todos são chamados de beats:

- Packetbeat
- Metricbeat
- Winlogbeat
- Auditbeat
- Filebeat
- Heartbeat

É possível instalar os beats por servidor ou requerimento. Os beats enviam os dados para o Elasticsearch e/ou Logstash

Em nosso laboratório iremos utilizar o Filebeat que é basicamente um beat para coletar arquivos de logs e enviar suas entradas para o Elasticsearch e/ou Logstash

O Filebeat é muito útil para coleta de logs como logs de acesso ou erro e possui diversos módulos para leitura de logs tais como nginx, Apache web server, Mysql etc.

Os beats não estão limitados aos citados como exemplo, existem inúmeros tipos de beats diferentes desenvolvidos pela comunidade, cada um com sua particularidade porém todos com o mesmo propósito.

Instalação e configuração da pilha ELK (Elastic, Logstash e Kibana)

LAB 5.1 - Instalação e configuração do Elasticsearch

Neste laboratório vamos aprender como instalar e configurar o Elasticsearch.

Acesse a VM **kibana-audit**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.13
```

O Elasticsearch requer OpenJDK ou Oracle JDK. Realize a instalação do pacote **openjdk-11-jre**:

```
1 | sudo apt update  
2 | sudo apt install -y openjdk-11-jre
```

Em seguida teste a variável **JAVA_HOME**:

```
1 | java -version
```

Resultado:

```
1 | openjdk version "11.0.9.1" 2020-11-04  
2 | OpenJDK Runtime Environment (build 11.0.9.1+1-post-Debian-1  
   deb10u2)  
3 | OpenJDK 64-Bit Server VM (build 11.0.9.1+1-post-Debian-1deb10u2,  
   mixed mode, sharing)
```

O próximo passo é realizar a instalação dos pré-requisitos,

```
1 | sudo apt install -y wget apt-transport-https curl
```

Adicione a chave e configuração do repositório do Elasticsearch.

```
1 | wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |  
    sudo apt-key add -  
2 |  
3 | echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt  
      stable main" | sudo tee -a /etc/apt/sources.list.d/elastis  
      -7.x.list
```

Para terminar esta primeira etapa, atualize a lista de pacotes e instala o Elasticsearch.

```
1 | sudo apt update  
2 | sudo apt install -y elasticsearch-oss
```

Configuração do Elasticsearch

Para o Elasticsearch fizemos algumas alterações no arquivo **elasticsearch.yml** localizado no diretório **/etc/elasticsearch**:

```
1 | sudo vim /etc/elasticsearch/elasticsearch.yml  
2 | cluster.name: auditlogs  
3 | path.data: /var/lib/elasticsearch  
4 | path.logs: /var/log/elasticsearch  
5 | network.host: 172.16.0.13  
6 | http.port: 9200  
7 | cluster.initial_master_nodes: "172.16.0.13"
```

O arquivo **elasticsearch.yml** é responsável por todas as configurações da aplicação Elasticsearch.

É possível definir através dos campos:

- **path.data** – Caminho para armazenamento dos dados do elasticsearch;
- **path.logs** – Caminho para armazenamento dos logs do elasticsearch;
- **network.host** – Endereço de rede no qual o Elasticsearch estará escutando. Em nossa aplicação utilizamos o localhost, caso fossemos utilizar um cluster de elasticsearch devriam alterar o campo para o endereço IP;
- **http.port** – Porta padrão na qual o elasticsearch estará escutando e aguardando conexões (por padrão a porta 9200 é utilizada).

- **cluster.initial_master_nodes** – Define o IP do nó que sera qualificado para ser o mestre do Cluster.

Em seguida, altere para **512m** o tamanho inicial e máximo do espaço total de heap:

```
1 | sudo vim /etc/elasticsearch/jvm.options
2 | ....
3 |
4 | -Xms512m
5 | -Xmx512m
```

Inicie e ative na inicialização do sistema o serviço do **Elasticsearch**.

```
1 | sudo systemctl start elasticsearch
2 | sudo systemctl enable elasticsearch
```

Teste a comunicação com o Elasticsearch na porta 9200:

```
1 | curl -X GET http://172.16.0.13:9200
```

Resultado:

```
1 | {
2 |   "name" : "kibana",
3 |   "cluster_name" : "auditlogs",
4 |   "cluster_uuid" : "m04kggnAQ SqTUBV6vK_0DA",
5 |   "version" : {
6 |     "number" : "7.10.1",
7 |     "build_flavor" : "oss",
8 |     "build_type" : "deb",
9 |     "build_hash" : "1c34507e66d7db1211f66f3513706fdf548736aa",
10 |    "build_date" : "2020-12-05T01:00:33.671820Z",
11 |    "build_snapshot" : false,
12 |    "lucene_version" : "8.7.0",
13 |    "minimum_wire_compatibility_version" : "6.8.0",
14 |    "minimum_index_compatibility_version" : "6.0.0-beta1"
15 |  },
16 |  "tagline" : "You Know, for Search"
17 | }
```

LAB 5.2 - Instalação e configuração do Logstash

Neste laboratório vamos aprender como instalar e configurar o Logstash.

Instalação do Logstash

Execute o comando necessário para instalar o Logstash.

```
1 | sudo apt install -y logstash-oss=7.10.2
```

Configuração do Logstash

Para o logstash, não efetuamos alterações no arquivo **logstash.yml** uma vez que o mesmo aponta para o diretório **/etc/logstash/conf.d** para leitura dos arquivos de configuração.

Neste diretório criamos 3 arquivos para as etapas da pipeline de logs:

- filebeat-input.conf
- output-elasticsearch.conf
- syslog-filter.conf

Input Stage

O arquivo **filebeat-input.conf** é responsável por informar qual a porta no qual o logstash estará escutando dados enviados pelo filebeat no Input Stage.

```
1 | sudo vim /etc/logstash/conf.d/filebeat-input.conf
2 | input {
3 |   beats {
4 |     port => 5044
5 |     type => syslog
6 |   }
7 | }
```

As tags e seus significados:

- **input** – Informa qual o estágio a ser trabalhado por meio da configuração;
- **beats** – Define que a entrada será de um módulo de beats;
- **port** – Define a porta de escuta para o serviço;
- **type** – Define o tipo do evento que será recebido.

Outras opções podem ser vistas na documentação oficial em [1].

Filter Stage

O arquivo **syslog-filter.conf** é responsável por informar quais filtros serão aplicados no arquivo recebido pelo filebeat no Filter Stage

```
1 | sudo vim /etc/logstash/conf.d/syslog-filter.conf
2 | filter {
3 |   if [type] == "syslog" {
4 |     grok {
5 |       match => { "message" => "%{SYSLOGTIMESTAMP:
6 |                     syslog_timestamp} %{SYSLOGHOST:syslog_hostname} %{DATA
7 |                     :syslog_program}(?:\[ %{POSINT:syslog_pid}\])?: %{"
8 |                     GREEDYDATA:syslog_message}" }
9 |       add_field => [ "received_at", "%{@timestamp}" ]
10 |      add_field => [ "received_from", "%{host}" ]
11 |    }
12 |  }
13 | }
```

- **filter** – Define que o arquivo trata um estágio de filtragem de dados
- **if [type] == “syslog”** – Verifica se o tipo da entrada recebida pelo input stage é do tipo syslog e aplica as configurações
- **grok** – Plugin para analistar e estruturar texto. Atualmente o **grok** é a melhor maneira de analisar e organizar dados de log não estruturados em algo estruturado e consultável. Possui mais de 120 padrões incorporados ao logstash. Mais informações podem ser encontradas na documentação oficial em [2].
- **match** - É o parâmetro do plugin **grok** no qual são combinados padrões de texto em algo que possa ser comparado aos logs. Utilizamos o match no padrão **%{SYNTAX:SEMANTIC}** onde:
 - **SYNTAX** – É o nome do parâmetro no qual corresponderá com o texto;
 - * Exemplos:
 - **3,44** – É correspondido pelo padrão **NUMBER**;
 - **123.234.12.1** – É correspondido pelo padrão **IP**;
 - **SEMANTIC** – É o identificador dado a fração do texto que foi correspondido
 - Exemplos:
 - * **3,44** – Poderia ser a duração de um evento em segundos, poderíamos chamar de **duration**
 - * **123.234.12.1** – Pode ser o identificador de um **client** efetuando um request.
 - Exemplo de filtro: **%{NUMBER:durantion} %{IP:client}**

Outras opções podem ser encontradas na documentação oficial em [3].

Output Stage

O arquivo **output-elasticsearch.conf** é o responsável por informar no Output Stage o endpoint do Elasticsearch o qual receberá os dados capturados e filtrados (no Filter Stage).

```
1 | sudo vim /etc/logstash/conf.d/output-elasticsearch.conf
2 | output {
3 |   elasticsearch { hosts => ["172.16.0.13:9200"] }
4 |   hosts => "172.16.0.13:9200"
5 |   manage_template => false
6 |   index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
7 | }
8 | }
```

- **output** – Define que o arquivo trata um estágio de saída de dados;
- **elasticsearch** – Define quais hosts/portas do Elasticsearch receberão os dados, caso não deseje que os dados sejam enviados para o elasticsearch, podemos utilizar no lugar de *elasticsearch* o parâmetro **file** o qual gravará os dados de evento no disco, **graphite** que enviará os dados para o graphite (uma ferramenta popular para armazenamento e geração de gráficos de métricas) ou **statsd** (um serviço que escuta por estatísticas como contadores e timers via UDP);
- **hosts** – Configura o host da instância remota do Elasticsearch;
- **manage_template** – Parâmetro para aplicar um template ao elasticstash durante o inicio do logstash;
- **index** – Formato no qual serão indexados os dados enviados ao Elasticsearch.

Outras opções podem ser vistas na documentação oficial em [4].

Em seguida, altere para **512m** o tamanho inicial e máximo do espaço total de heap:

```
1 | sudo vim /etc/logstash/jvm.options
2 | ....
3 |
4 | -Xms512m
5 | -Xmx512m
```

Inicie e ative na inicialização do sistema o serviço do **Logstash**.

```
1 | sudo systemctl start logstash
2 | sudo systemctl enable logstash
```

LAB 5.3 - Instalação e configuração do Kibana

Neste laboratório vamos aprender como instalar e configurar o Kibana.

Instalação do Kibana e Nginx

Execute o comando necessário para instalar o Kibana e Nginx.

```
1 | sudo apt install -y kibana-oss nginx
```

Configuração do Kibana

Para o Kibana, fizemos alterações no arquivo **kibana.yml** localizado em **/etc/kibana**

```
1 | sudo vim /etc/kibana/kibana.yml
2 | server.port: 5601
3 | server.host: "172.16.0.13"
4 | elasticsearch.hosts: ["http://172.16.0.13:9200"]
```

O arquivo **kibana.yml** é responsável por todas as configurações da aplicação Kibana.

É possível definir através dos campos:

- **server.port** – Porta padrão na qual o kibana estará escutando e publicando sua interface web;
- **server.host** – Endereço de rede no qual o kibana estará escutando. Como em nossa aplicação utilizamos o **nginx** para efetuar o redirecionamento através de proxy reverso, utilizamos a configuração como localhost, caso fossemos publicar sem o nginx, poderíamos alterar para que ele ouça todas as conexões alterando o campo para *0.0.0.0*;
- **elasticsearch.hosts** – Endereços do servidores Elasticsearch no qual o Kibana se conectará para geração das dashboards e índices.

Inicie e ative na inicialização do sistema o serviço do Logstash.

```
1 | sudo systemctl start kibana
2 | sudo systemctl enable kibana
```

Configuração do Nginx

Vamos configurar o Nginx para realizar Proxy Reverso e adicionar autenticação básica para acessar o Kibana. Crie o arquivo **kibana.4labs.example** no diretório **/etc/nginx/sites-available**.

```
1 | sudo vim /etc/nginx/sites-available/kibana.4labs.example
2 | server {
3 |   listen 80;
4 |
5 |   server_name kibana.4labs.example;
6 |
7 |   auth_basic "Acesso Restrito";
8 |   auth_basic_user_file /etc/nginx/htpasswd.users;
9 |
10|   location / {
11|     proxy_pass http://172.16.0.13:5601/;
12|     proxy_http_version 1.1;
13|     proxy_set_header Upgrade $http_upgrade;
14|     proxy_set_header Connection 'upgrade';
15|     proxy_set_header Host $host;
16|     proxy_cache_bypass $http_upgrade;
17|   }
18| }
```

Em seguida crie um arquivo, para armazenar um usuário e senha do usuário administrativo do Kibana:

```
1 | echo "admin:`openssl passwd 4linux`" | sudo tee -a /etc/nginx/
      htpasswd.users
```

Ative a nova configuração criando um link simbólico para o diretório **sites-enabled**.

```
1 | sudo ln -s /etc/nginx/sites-available/kibana.4labs.example /etc/
      nginx/sites-enabled/kibana.4labs.example
```

E remova a configuração padrão do Nginx:

```
1 | sudo rm /etc/nginx/sites-enabled/default
```

Reinic peace e ative na inicialização do sistema o serviço do Nginx.

```
1 | sudo systemctl restart nginx  
2 | sudo systemctl enable nginx
```

Acessar o dashboard do Kibana

Adicione uma entrada no arquivo **/etc/hosts** em máquina física apontando para o IP **172.16.0.13**, o domínio **kibana.4labs.example**:

No Linux e Mac

- Arquivo **/etc/hosts**:

```
1 | ....  
2 | 172.16.0.13 kibana.4labs.example
```

No Windows

- Arquivo: C:\Windows\System32\Drivers\etc\hosts

```
1 | ....  
2 | 172.16.0.13 kibana.4labs.example
```

No navegador da sua máquina física acesse o domínio **kibana.4labs.example**:

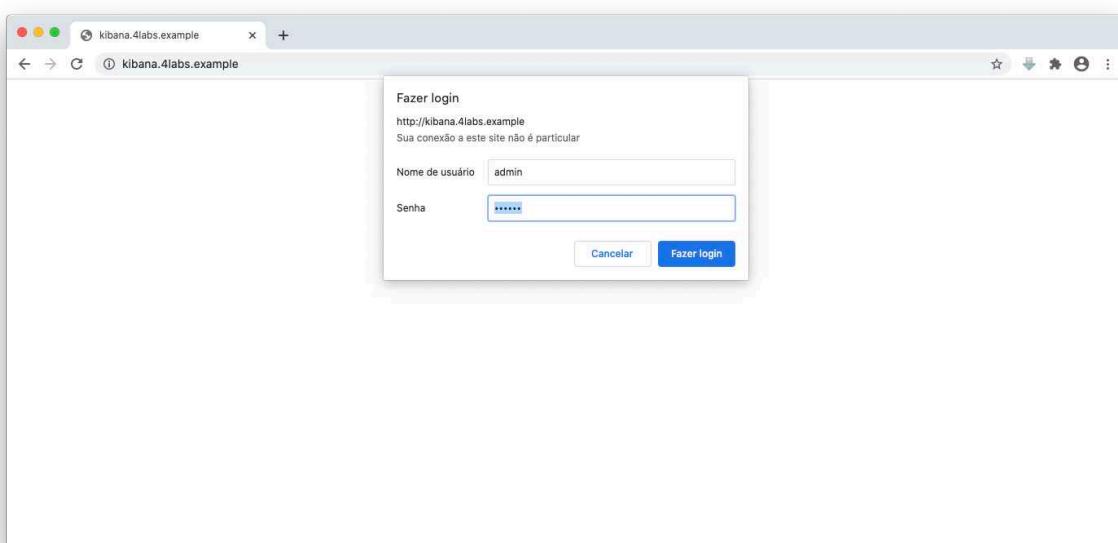


Fig. 5.7: Acessando a página do kibana

Após o sistema carregar, será exibida a tela inicial do **Kibana**.

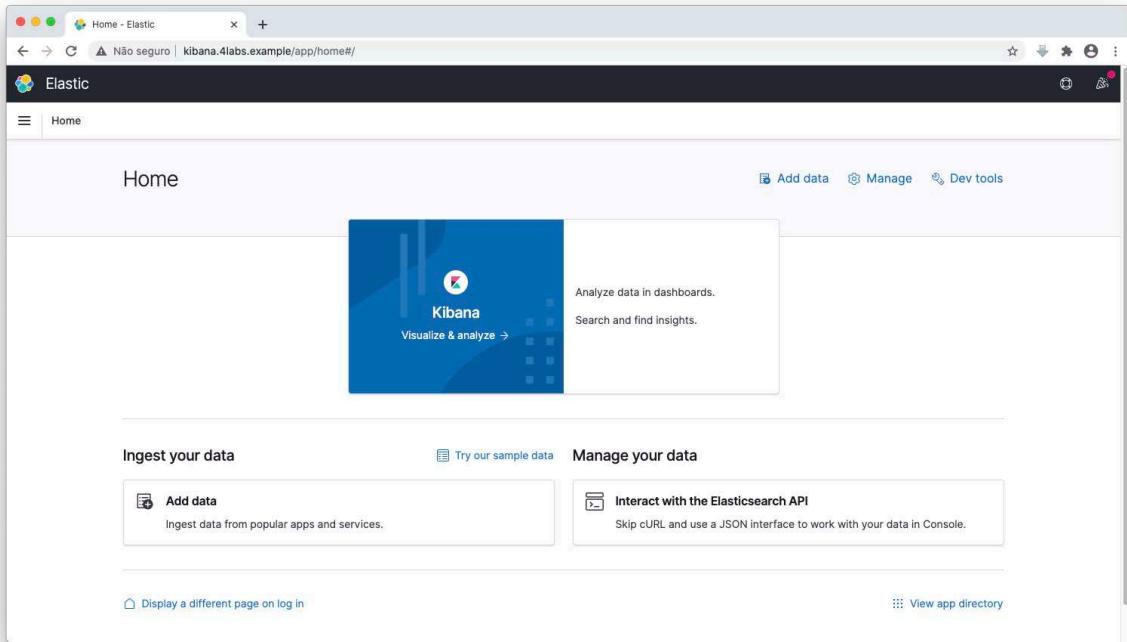


Fig. 5.8: Página inicial do Kibana

Utilizar o FileBeat para envio de arquivos de logs

Beats

Beats são ferramentas leves desenvolvidas para exportar todos os tipos de dados para o Elasticsearch e/ou Logstash.

Através de Beats é possível exportar dados de servidores, containers e até mesmo funções.



Fig. 5.9: Beats

O principal ganho na utilização de beats conectados diretamente ao Logstash ou Elasticsearch é a utilização de um recurso chamado **Back-Pressure Sensitive Protocol**, onde o Filebeat e o Logstash/Elasticsearch conversam entre si para evitar o travamento e a perda de informações ao receber os logs.

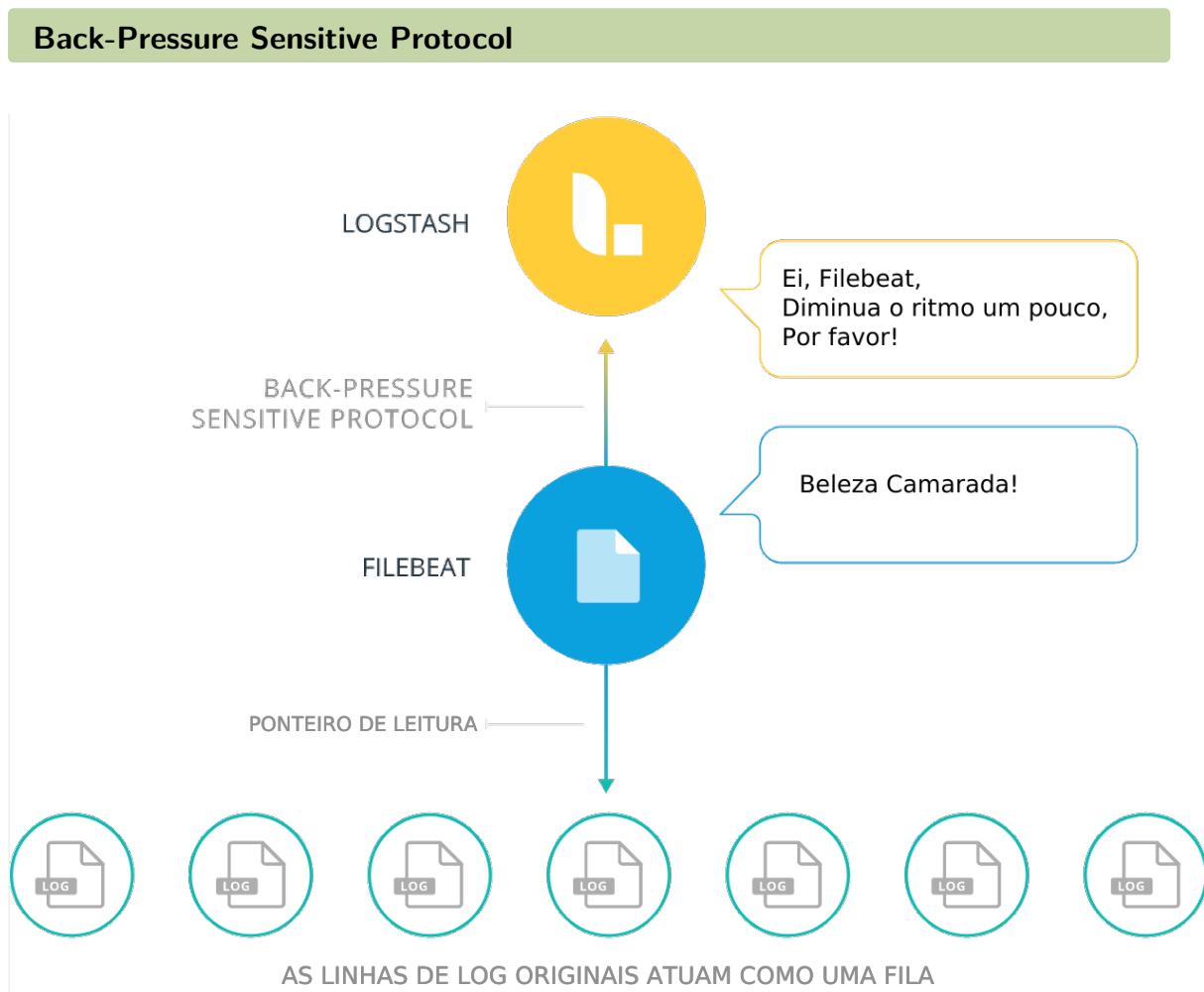


Fig. 5.10: Back-Pressure Sensitive Protocol

O **Filebeat** possui um ponteiro de leitura que opera sob uma velocidade constante, quando o **Logstash** estiver ocupado tratando os arquivos ele informa ao Filebeat para que o mesmo diminua sua taxa de leitura para que os dados não sejam perdidos, uma vez descongestionado o Filebeat continua em seu ritmo original de leitura/envio de logs.

Beats Elastic Co.

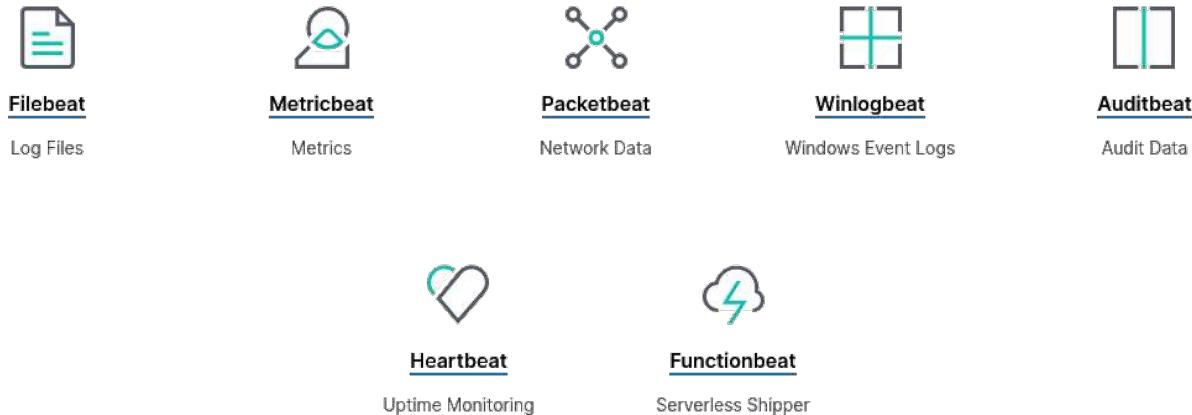


Fig. 5.11: Beats Family

Beats desenvolvidos pela Elastic Co:

- **Filebeat** – Beat para leitura de arquivos de log, com módulos internos para programas como auditd, Apache, Nginx, System, MySQL etc.
- **Metricbeat** – Beat para monitoramento a nível de sistema como nível de uso de CPU, memória, sistemas de arquivos, I/O, redes, processos etc.
- **Packetbeat** – Beat para monitoramento de protocolos de redes e aplicações;
- **Winlogbeat** – Beat para leitura de logs de eventos do Windows;
- **Auditbeat** – Beat para monitoramento de atividade de usuário e processos. Coleta os mesmos dados que o Auditd;
- **Heartbeat** – Beat para monitoramento de uptime de serviços e hosts.
- **Functionbeat** – Beat para monitoramento de infraestrutura de cloud, seu deploy é feito como função em frameworks serverless, como por exemplo AWS Lambda.

Community Beats

Beats não são desenvolvidos somente pela Elastic, uma vez que são Open Source, toda a comunidade contribui no desenvolvimento de novos beats chamados de **Community Beats**

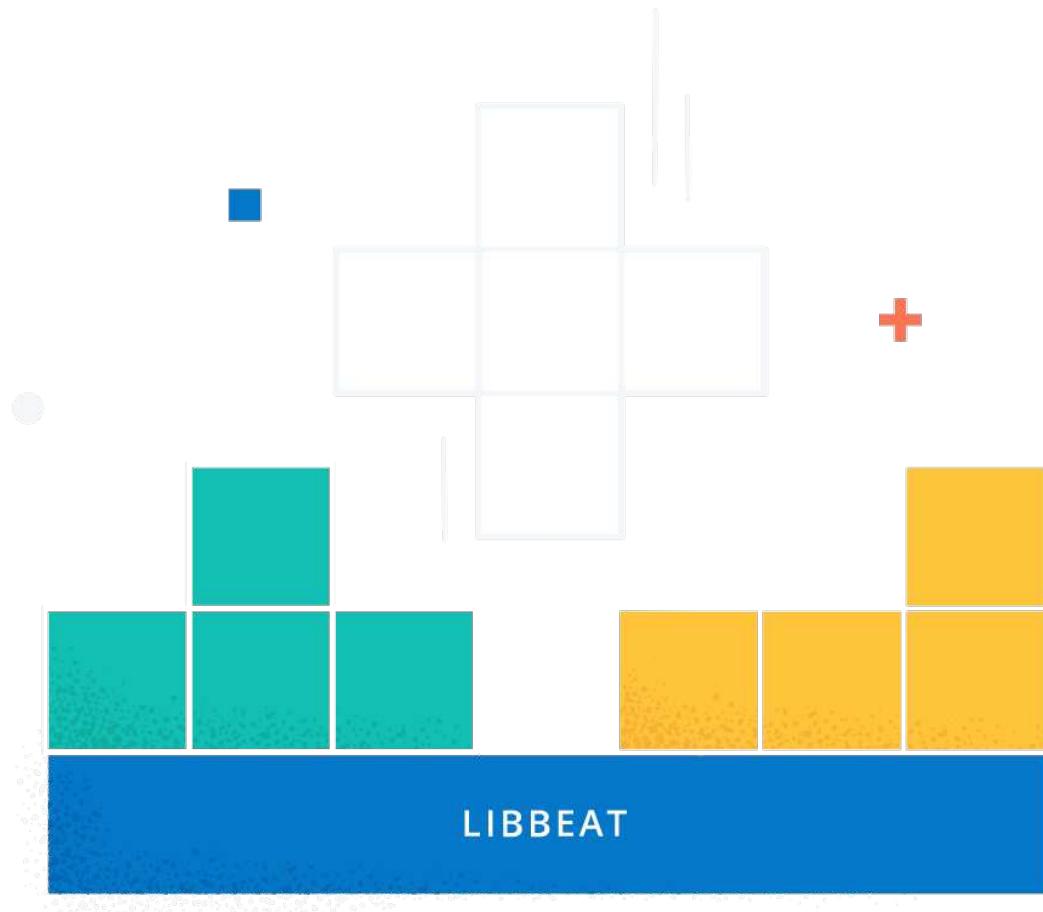


Fig. 5.12: libbeat

Atualmente existem mais de 90 Community Beats os quais atendem diversas áreas não cobertas pelos Beats tradicionais da Elastic Co.

A lista completa de community beats pode ser vista na documentação oficial em [5].

Filebeat

Fig. 5.13: Filebeat

O Filebeat foi desenvolvido para atender aqueles usuários que queriam apenas executar um **tail** em um arquivo de log de maneira simples, agregando o mesmo a possibilidade de pesquisa e centralizando o mesmo.

Iremos utilizar o Filebeat para capturar logs de sistema de nossos servidores da nossa infraestrutura e visualizar no Kibana.

LAB 5.4 - Instalação e configuração do Filebeat

Neste laboratório vamos aprender como instalar e configurar o Filebeat em todas as VMs.

Execute o comando necessário para instalar o Filebeat na VM **Kibana**.

```
1 | sudo apt install -y filebeat=7.10.2
```

Acesse a VM **Graylog**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.12
```

Execute o comando necessário para instalar o Filebeat na VM **Graylog**.

```
1 | curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat  
     /filebeat-7.10.1-amd64.deb  
2 | sudo dpkg -i filebeat-7.10.1-amd64.deb
```

Acesse a VM **Webserver**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.11
```

Execute o comando necessário para instalar o Filebeat na VM **Webserver**.

```
1 | curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat  
     /filebeat-7.10.1-x86_64.rpm  
2 | sudo rpm -vi filebeat-7.10.1-x86_64.rpm
```

Configuração do Filebeat em todas as VMs

O arquivo **filebeat.yml** é responsável por todas as configurações da aplicação Filebeat. Vamos realizar um backup do arquivo padrão antes de iniciar a configuração:

```
1 | sudo mv /etc/filebeat/filebeat.yml /etc/filebeat/filebeat.bkp
```

Crie um novo arquivo **filebeat.yml**, seguindo o modelo:

```
1 | sudo vim /etc/filebeat/filebeat.yml  
2 | filebeat.inputs:  
3 |   - type: log  
4 |     enabled: true  
5 |     paths:  
6 |       - /var/log/*.log  
7 |  
8 | filebeat.config.modules:  
9 |   path: ${path.config}/modules.d/*.yml  
10 |  reload.enabled: false  
11 |  
12 | setup.template.settings:  
13 |   index.number_of_shards: 1  
14 |  
15 | setup.kibana:  
16 |   host: "172.16.0.13:5601"  
17 |  
18 | output.logstash:
```

```
19 |     hosts: ["172.16.0.13:5044"]
20 |
21 |processors:
22 |  - add_host_metadata:
23 |    when.not.contains.tags: forwarded
24 |  - add_cloud_metadata: ~
25 |  - add_docker_metadata: ~
26 |  - add_kubernetes_metadata: ~
```

Realize a configuração em todas as VMs.

Vamos conhecer as configurações:

- **filebeat.inputs** – Define uma configuração INPUT para o filebeat;
- - **type** – Define o tipo de entrada.
- **enabled** – Mude para *true* para habilitar esta configuração de entrada.
- **paths** – Define o caminhos em que os logs devem ser rastreados.
- **filebeat.config.modules** – Define uma configuração de módulos do filebeat;
- **output.logstash** – Define a configuração do módulo logstash;
- **hosts** – Define o IP e porta do servidor logstash.
- **processors** – Define os processadores para hosts, cloud, docker e kubernetes.

O próximo passo é ativar os módulos **logstash** e **system**.

```
1 | sudo filebeat modules enable logstash system
```

Verifique quais são os módulos ativados e desativados.

```
1 | sudo filebeat modules list
```

Para finalizar a configuração, precisamos carregar o template de índice.

```
1 | sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts =["172.16.0.13:9200"]'
```

Se você está logado na máquina, onde o serviço do Elasticsearch está ouvindo somente em localhost, execute o comando sem passar parâmetros:

```
1 | sudo filebeat setup
```

Inicie e ative na inicialização do sistema o serviço do Filebeat.

```
1 | sudo systemctl start filebeat  
2 | sudo systemctl enable filebeat
```

Metricbeat

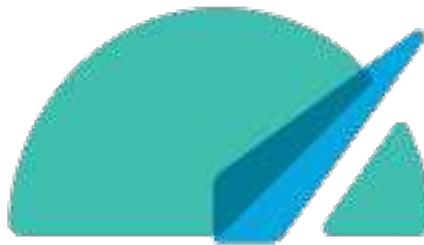


Fig. 5.14: Metricbeat

O Metricbeat foi desenvolvido para atender os usuários que queriam monitorar servidores em nível de sistema, coletando dados como: uso de CPU, memória, sistema de arquivos, I/O de disco, I/O e estatísticas de redes.

Iremos utilizar o Metricbeat para visualizar as métricas do nosso servidor de logs

LAB 5.5 - Instalação e configuração do Metricbeat

Neste laboratório vamos aprender como instalar e configurar o Metricbeat.

Instalação e configuração o o Metricbeat

Acesse a VM **Kibana**, através do comando **ssh**.

```
1 | ssh suporte@172.16.0.13
```

Execute o comando necessário para instalar o Metricbeat na VM **Kibana**.

```
1 | sudo apt install -y metricbeat=7.10.2
```

Vamos realizar um backup do arquivo padrão antes de iniciar a configuração:

```
1 | sudo mv /etc/metricbeat/metricbeat.yml /etc/metricbeat/  
metricbeat.bkp
```

Crie um novo arquivo **metricbeat.yml**, seguindo o modelo:

```
1 | sudo vim /etc/metricbeat/metricbeat.yml
2 | metricbeat.config.modules:
3 |   path: ${path.config}/modules.d/*.yml
4 |   reload.enabled: false
5 |
6 | setup.template.settings:
7 |   index.number_of_shards: 1
8 |   index.codec: best_compression
9 |
10 | setup.kibana:
11 |   host: "172.16.0.13:5601"
12 |
13 | output.elasticsearch:
14 |   hosts: ["172.16.0.13:9200"]
15 |
16 | processors:
17 |   - add_host_metadata: ~
18 |   - add_cloud_metadata: ~
19 |   - add_docker_metadata: ~
20 |   - add_kubernetes_metadata: ~
21 |
22 | logging.to_files: true
23 | logging.files:
24 |   path: /var/log/metricbeat
25 |   name: metricbeat
26 |   keepfiles: 7
27 |   permissions: 0644
```

Dentro da pasta **modules.d** existem mais de 40 modelos de módulos do metricbeat. Podemos utilizar qualquer um destes modelos para nossas máquinas, mas por padrão, apenas o módulo **system** vem habilitado, o qual monitora diversos dados relacionados ao sistema.

Execute a configuração inicial do ambiente:

```
1 | sudo metricbeat setup --dashboards
```

O comando **metricbeat setup --dashboards** faz a configuração inicial do ambiente bem como a configuração dos dashboards do Kibana.

Habilite e inicie o serviço do metricbeats:

```
1 | sudo systemctl enable metricbeat
2 | sudo systemctl start metricbeat
```

Verifique se as métricas estão sendo exportadas

```
1 | curl http://172.16.0.13:9200/_cat/indices/metricbeat*?v
```

Resultado:

```
1 | health status index          uuid                   pri
    rep docs.count docs.deleted store.size pri.store.size
2 | yellow open   metricbeat-2020.12.11 ET-jsum5Tx-lKrMcxojLsA 1
    1      135022           0      50.6mb      50.6mb
```

```
1 | curl -XGET 'http://172.16.0.13:9200/metricbeat-*/_search?pretty'
```

Resultado:

```
1 | {
2 |     "took" : 1397,
3 |     "timed_out" : false,
4 |     "_shards" : {
5 |         "total" : 1,
6 |         "successful" : 1,
7 |         "skipped" : 0,
8 |         "failed" : 0
9 |     },
10 |     "hits" : {
11 |         "total" : {
12 |             "value" : 10000,
13 |             "relation" : "gte"
14 |         },
15 |         "max_score" : 1.0,
16 |         "hits" : [
17 |             {
18 |                 "_index" : "metricbeat-2020.12.11",
19 |                 "_type" : "_doc",
20 |                 "_id" : "qJv6UXYB41i7UQK-jFsR",
21 |                 "_score" : 1.0,
22 |                 "_source" : {
23 |                     "tags" : [
24 |                         "beats_input_raw_event"
```

```
25  ],
26  "metricset" : {
27    "period" : 10000,
28    "name" : "load"
29  },
30  "@timestamp" : "2020-12-11T13:25:41.946Z",
31  "@version" : "1",
32  "host" : {
33    "mac" : [
34      "08:00:27:93:93:48",
35      "08:00:27:b8:7f:8a
```

Os comandos acima fazem a listagem dos índices de métricas e a coleta de eventos no formato raw.

Construção de dashboards com Kibana

LAB 5.6 - Realizar buscas no Kibana

Neste laboratório vamos aprender como configurar o Kibana para visualizar os logs das VMs.



Fig. 5.15: kibana

O Kibana é uma ferramenta que nos permite visualizar os dados do Elasticsearch e navegar no Elastic Stack. Através dele podemos rastrear dados, gerar gráficos e dashboards para entender a maneira que as requisições e dados caminham sobre nossas aplicações.

Primeiros passos com Kibana

Após o sistema carregar, será exibida a tela inicial do **Kibana**. Em seguida clique no menu **Manage**.

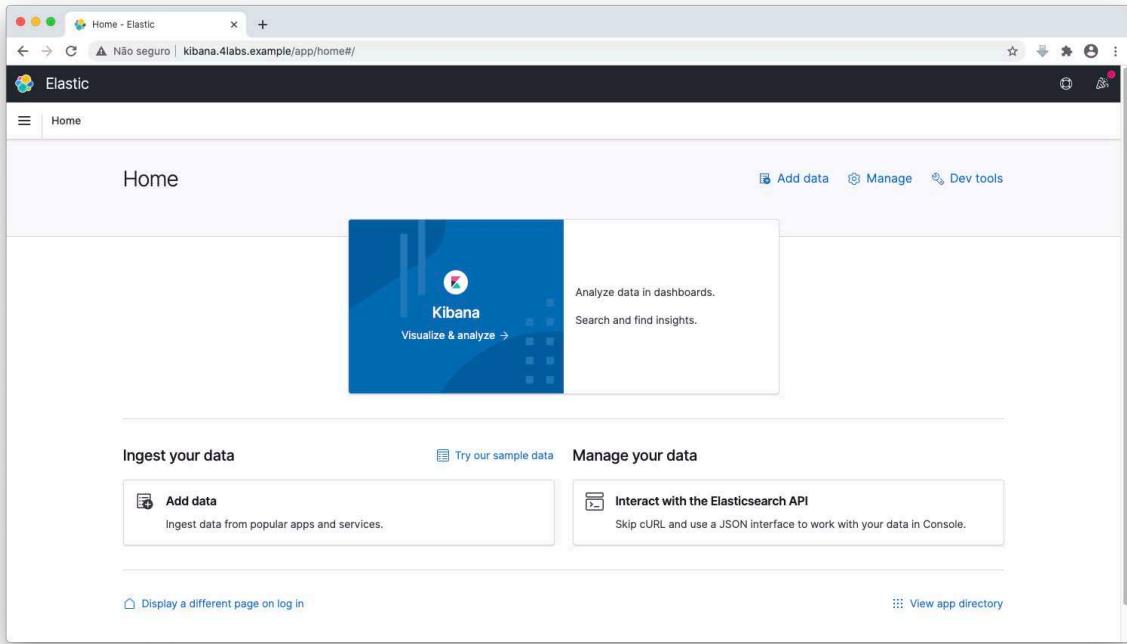


Fig. 5.16: Página inicial do Kibana

Index Patterns - Filebeat

Clique em **Index Patterns** > botão **Create index pattern**

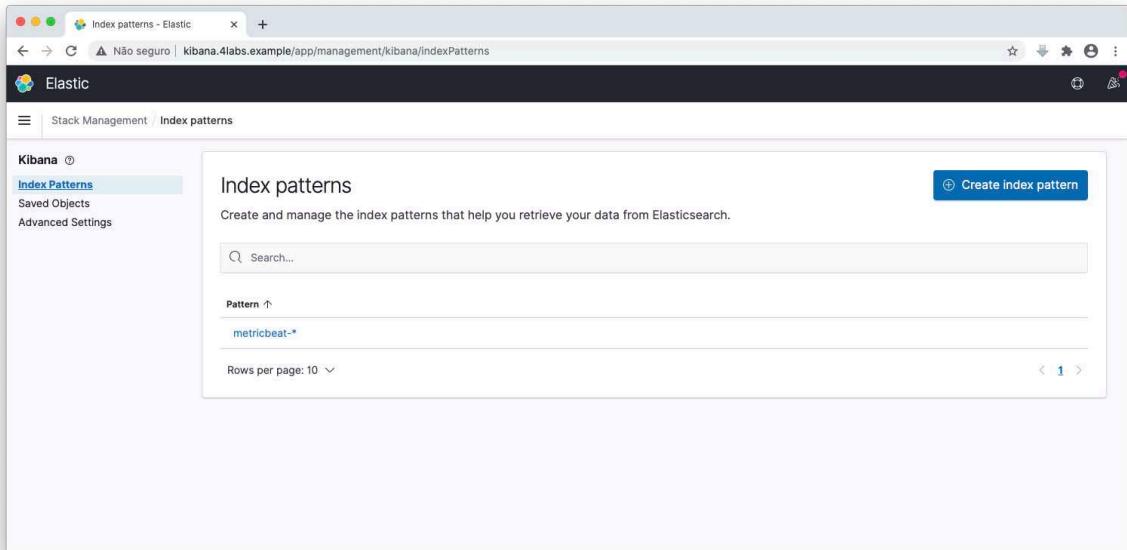


Fig. 5.17: Criando Index Patterns - Filebeat - ETAPA 1

Preencha o Pattern com o valor **filebeat-*** e clique em **Next Step**

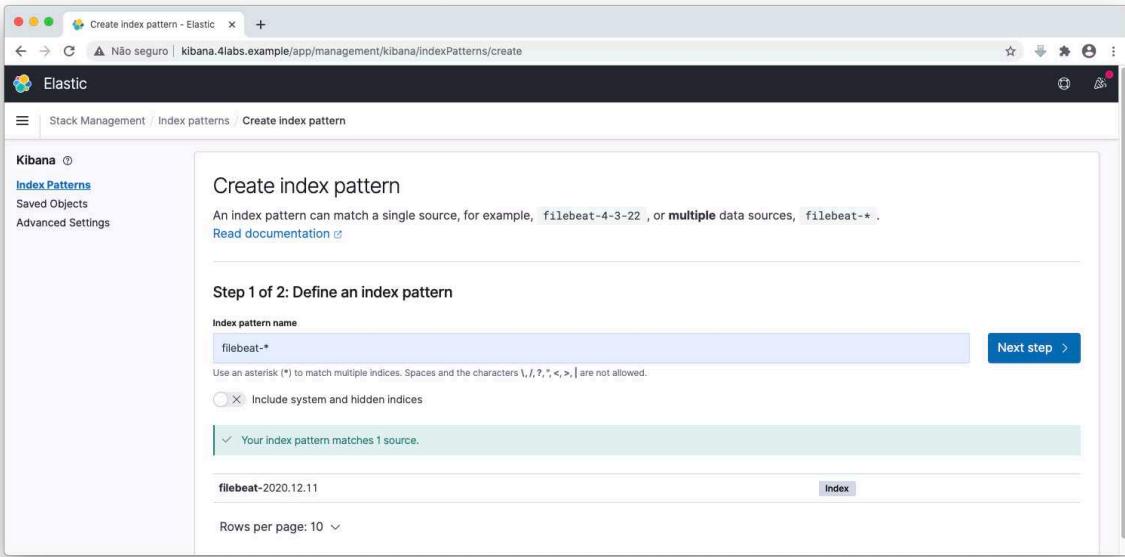


Fig. 5.18: Criando Index Patterns - Filebeat - ETAPA 2

Selecione no Dropbox o parâmetro **@timestamp** e clique em **Create Index Pattern**

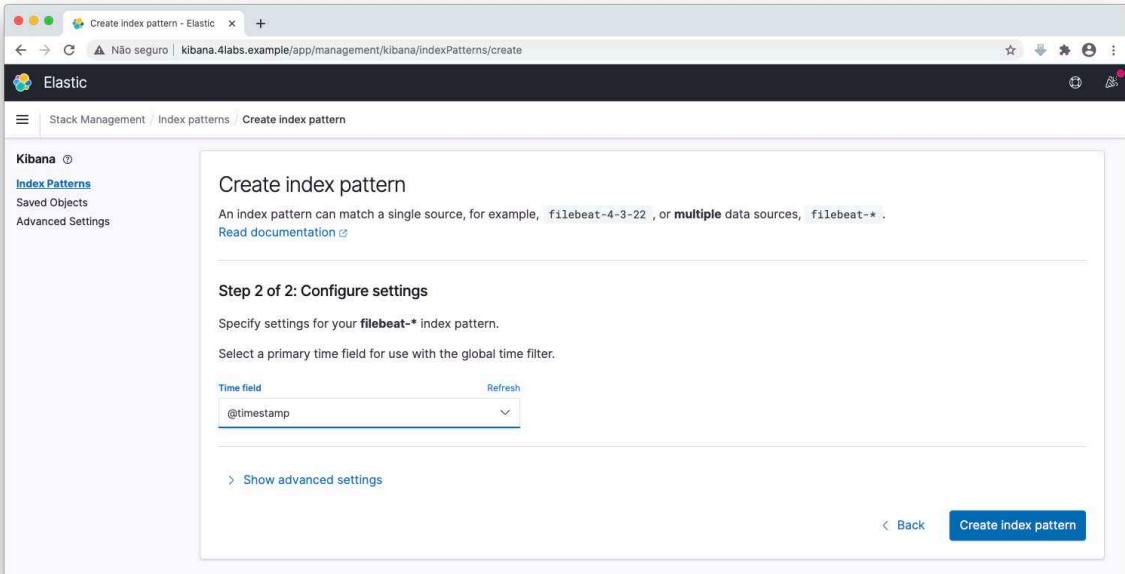


Fig. 5.19: Criando Index Patterns - Filebeat - ETAPA 3

Será exibida uma tela com os valores listados pelo índice **filebeat-***

The screenshot shows the Kibana interface for managing index patterns. The left sidebar has 'Index Patterns' selected under 'Stack Management'. The main area is titled 'filebeat-*' and shows a table of fields. The table includes columns for Name, Type, Format, Searchable, Aggregatable, and Excluded. Fields listed include @timestamp, @version, @version.keyword, _id, _index, _score, _source, and _type.

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	
@version	string		●		
@version.keyword	string		●	●	
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	

Fig. 5.20: Criando Index Patterns - Filebeat - ETAPA 4

Visualizar logs

Para visualizar logs das VMs, clique no menu ao lado esquerdo da tela em **Kibana > Discover**.

The screenshot shows the Kibana interface with the 'Discover' tab selected from the left sidebar. The main area is titled 'filebeat-*' and shows a table of fields. The table includes columns for Name, Type, Format, Searchable, Aggregatable, and Excluded. Fields listed include @timestamp, @version, @version.keyword, _id, _index, _score, _source, and _type.

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	
@version	string		●		
@version.keyword	string		●	●	
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	

Fig. 5.21: Visualizar logs no Kibana

5. CENTRALIZAÇÃO DE LOGS COM ELK

Serão exibidos os logs de todas as VMs.

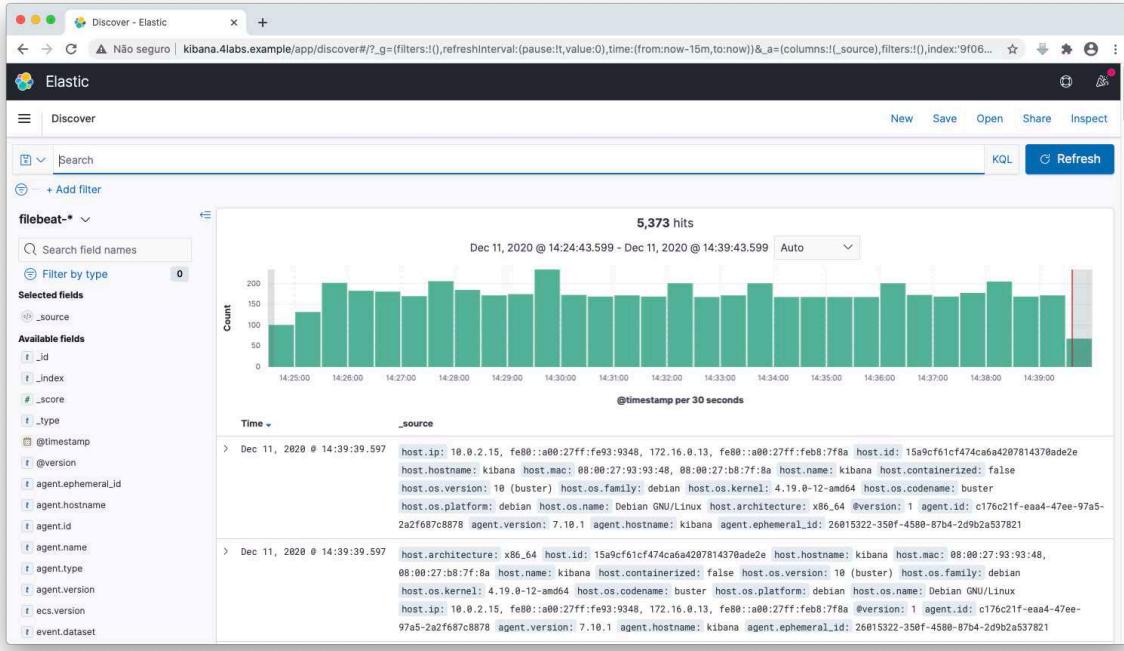


Fig. 5.22: Visualizando logs no Kibana

Filtre uma string no canto superior da tela usando o **filebeat-***. Como exemplo vamos pesquisar logs sobre a VM **graylog**.

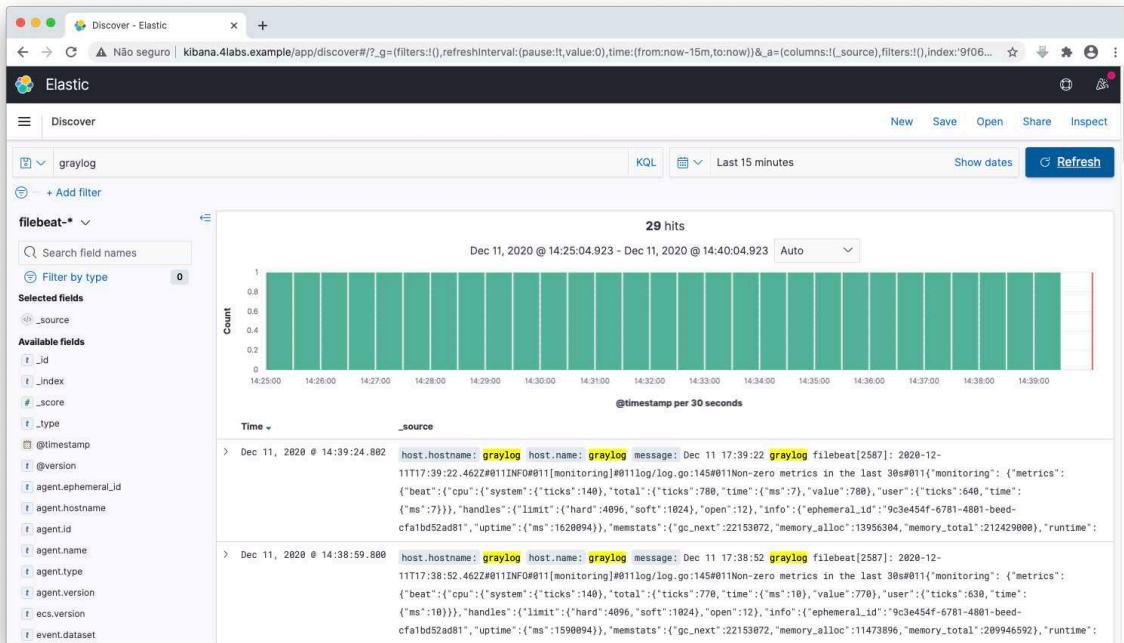


Fig. 5.23: Pesquisar logs da VM graylog

Aproveite e pesquise logs sobre a VM webserver.

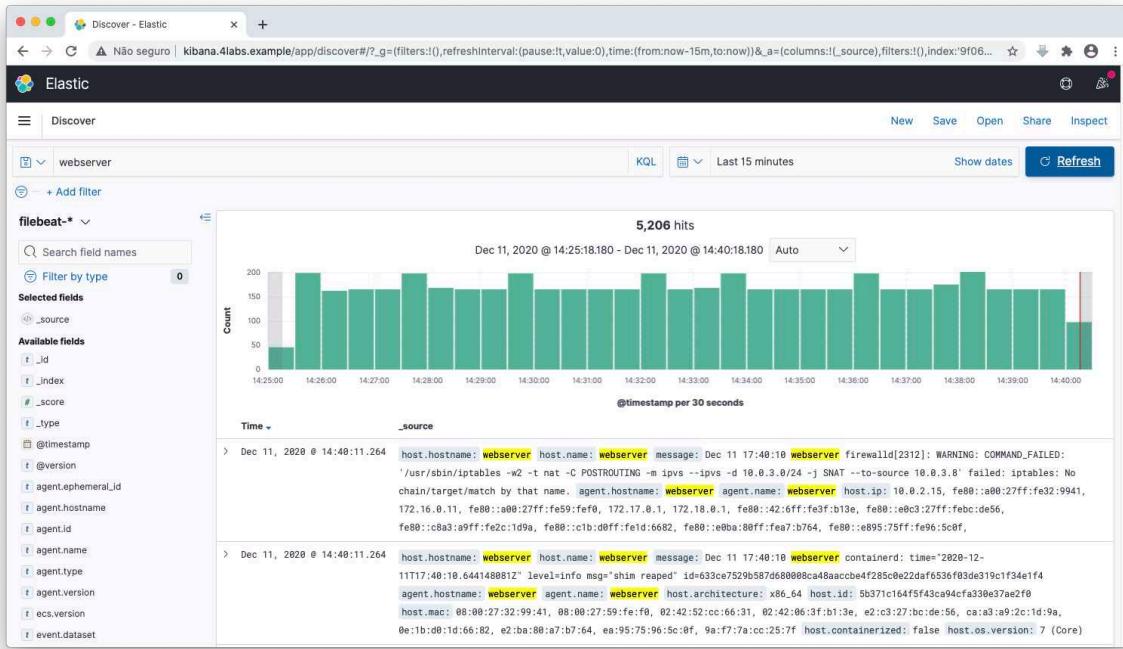


Fig. 5.24: Pesquisar logs da VM webserver

Index Patterns – Metricbeat

Clique em **Index Patterns** > botão **Create index pattern** e preencha o Pattern com o valor **metricbeat-*** e clique em **Next Step**

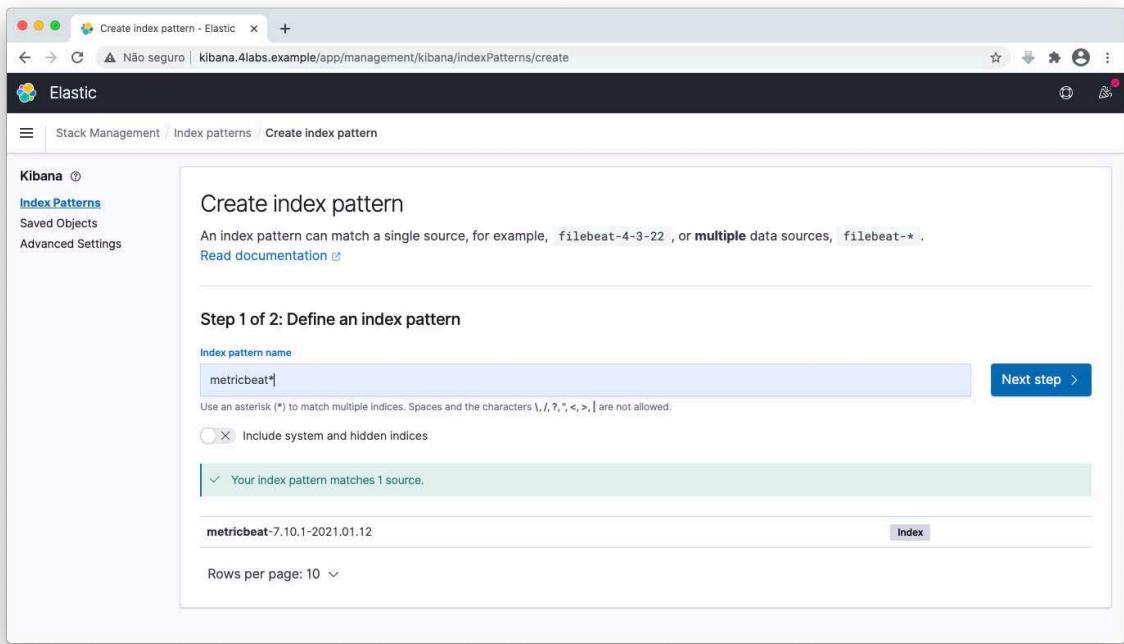


Fig. 5.25: Criando Index Patterns - Metricbeat - ETAPA 1

Selecione no Dropbox o parâmetro **@timestamp** e clique em **Create Index Pattern**

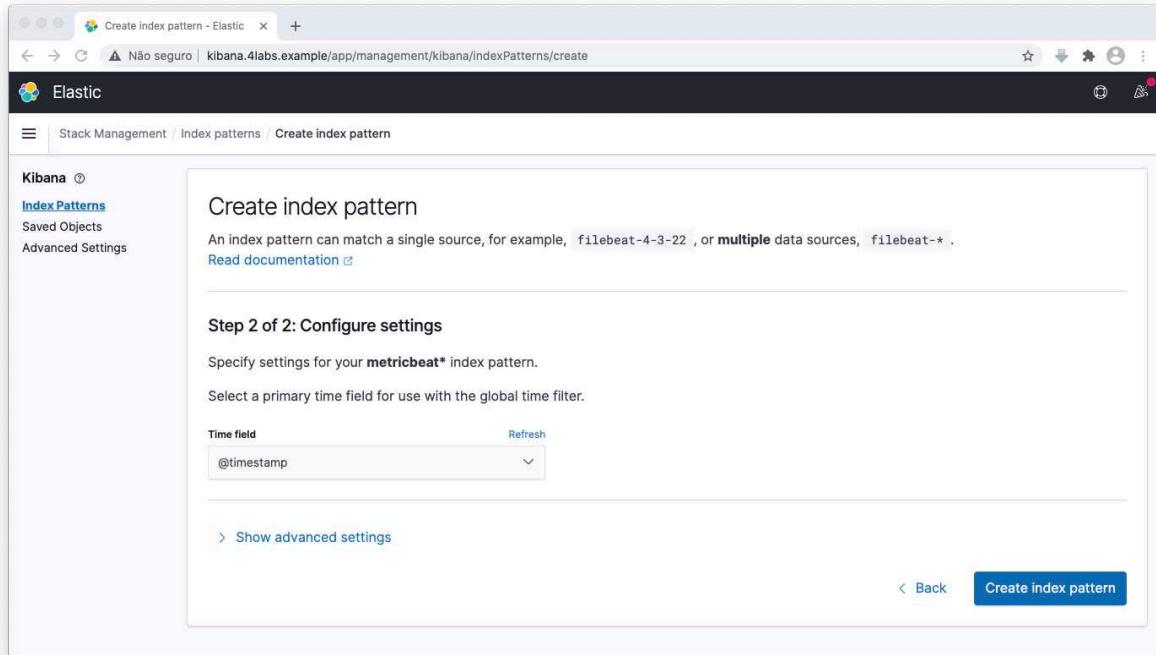


Fig. 5.26: Criando Index Patterns - Metricbeat - ETAPA 2

Será exibida uma tela com os valores listados pelo índice **metricbeat**.

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	✎
_id	string		●	●	✎
_index	string		●	●	✎
_score	number				✎
_source	_source				✎
_type	string		●	●	✎
aerospike.namespace.client.delete.error	number		●	●	✎
aerospike.namespace.client.delete.not_found	number		●	●	✎
aerospike.namespace.client.delete.success	number		●	●	✎
aerospike.namespace.client.delete.timeout	number		●	●	✎

Fig. 5.27: Criando Index Patterns - Metricbeat - ETAPA 3

Construção de dashboards

Um dos recursos mais interessantes do Kibana é a possibilidade de criar dashboards personalizados com as informações que sejam importantes para o usuário final, podemos criar gráficos de acordo com a nossa necessidade.

LAB 5.7 - Construção do dashboard

Neste laboratório vamos aprender como criar Dashboard no Kibana para visualizar métricas.

Para criar um dashboard, clique no menu ao lado esquerdo da tela em **Kibana > Dashboard**.

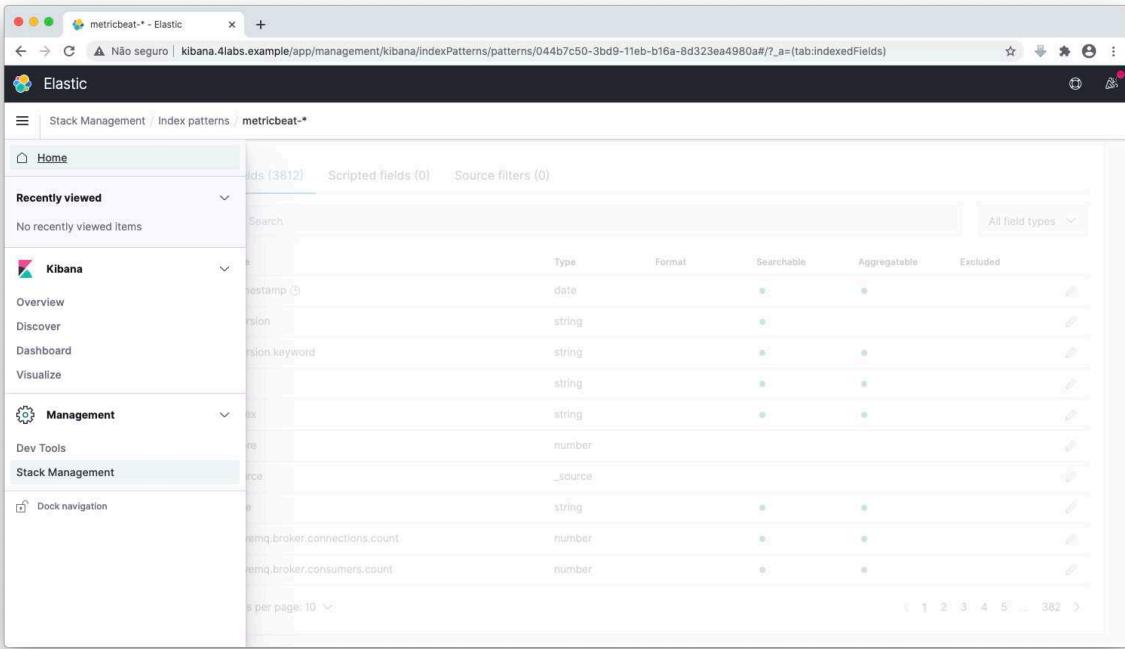


Fig. 5.28: Criando Dashboard ETAPA 1

Será exibida uma tela com diversos dashboards disponíveis e um campo de pesquisa

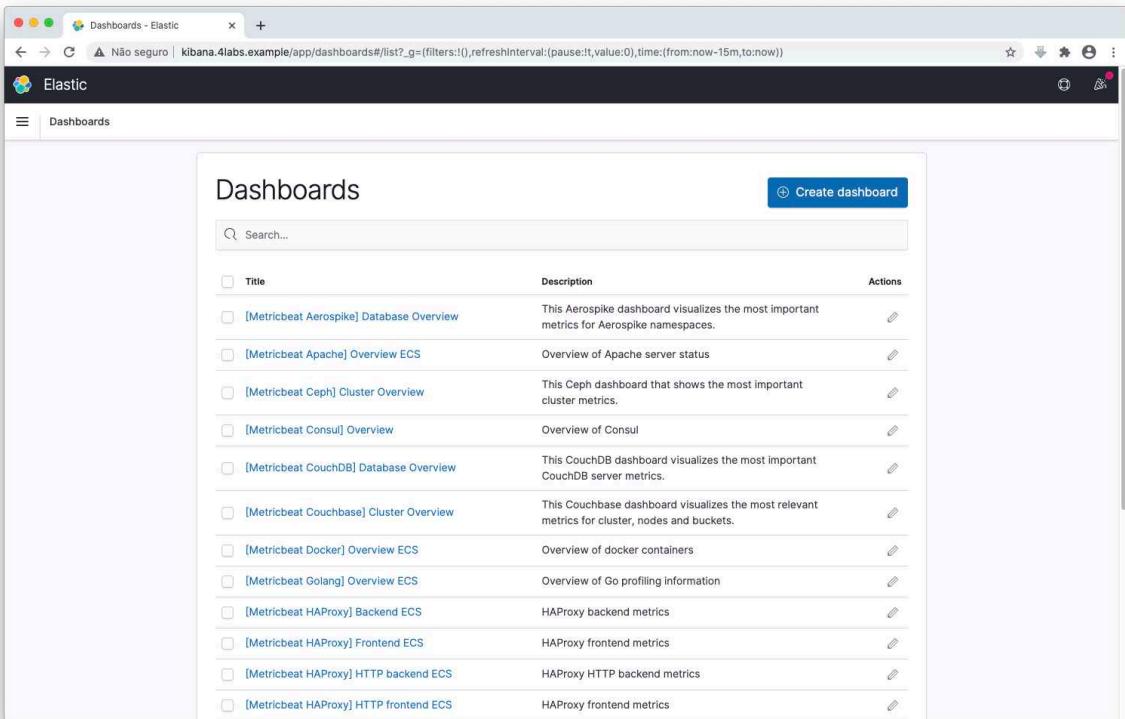


Fig. 5.29: Criando Dashboard ETAPA 2

Pequise a string ‘Host’ e selecione o modelo **[Metricbeat System] Host overview ECS**

Fig. 5.30: Criando Dashboard ETAPA 3

Visualize o Dashboard criado.

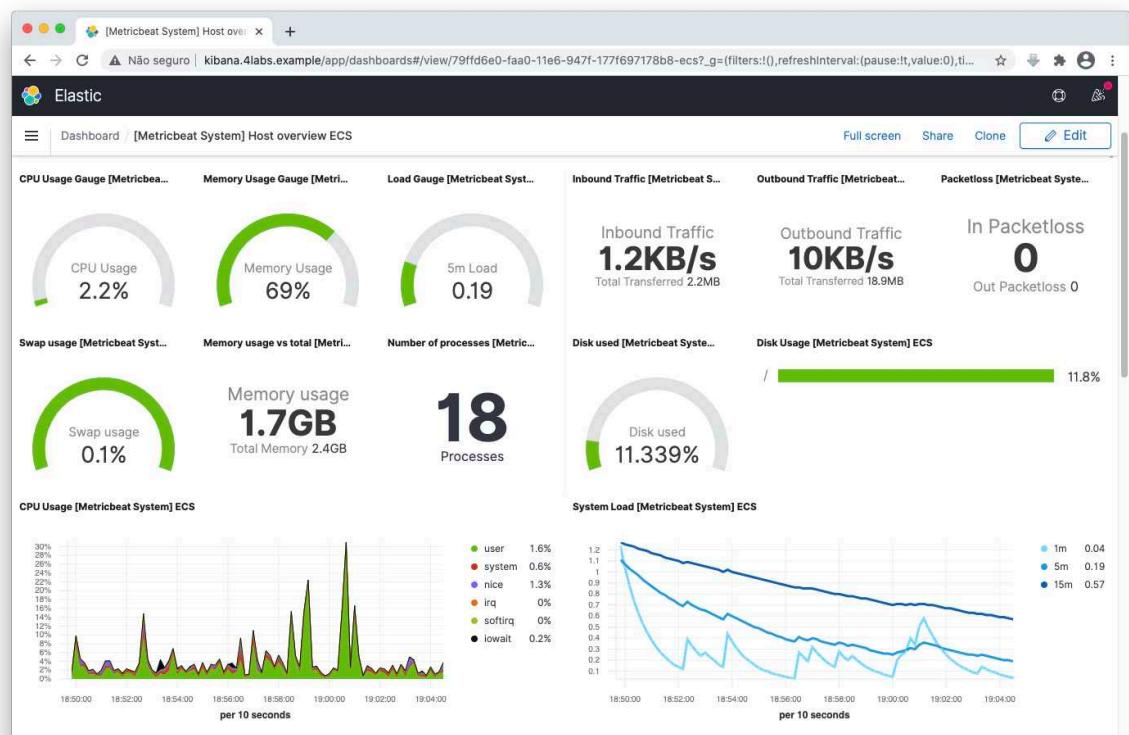


Fig. 5.31: Criando Dashboard ETAPA 4

Ao lado esquerdo da tela, clique em SAVED QUERIES e no botão **Save current query** e dê um nome ao mesmo.

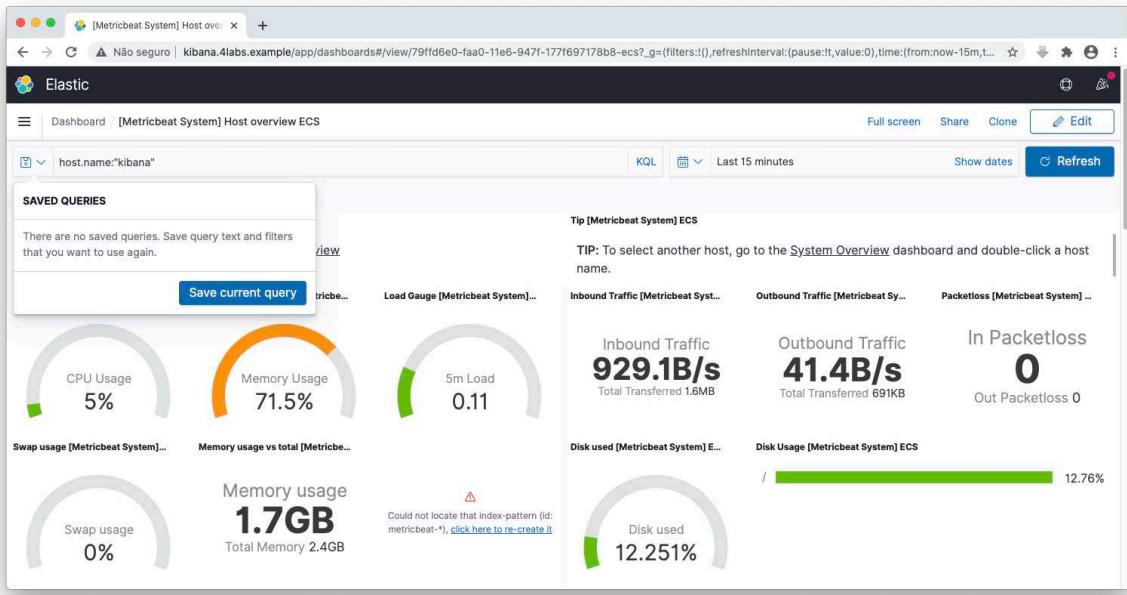


Fig. 5.32: Criando Dashboard ETAPA 5

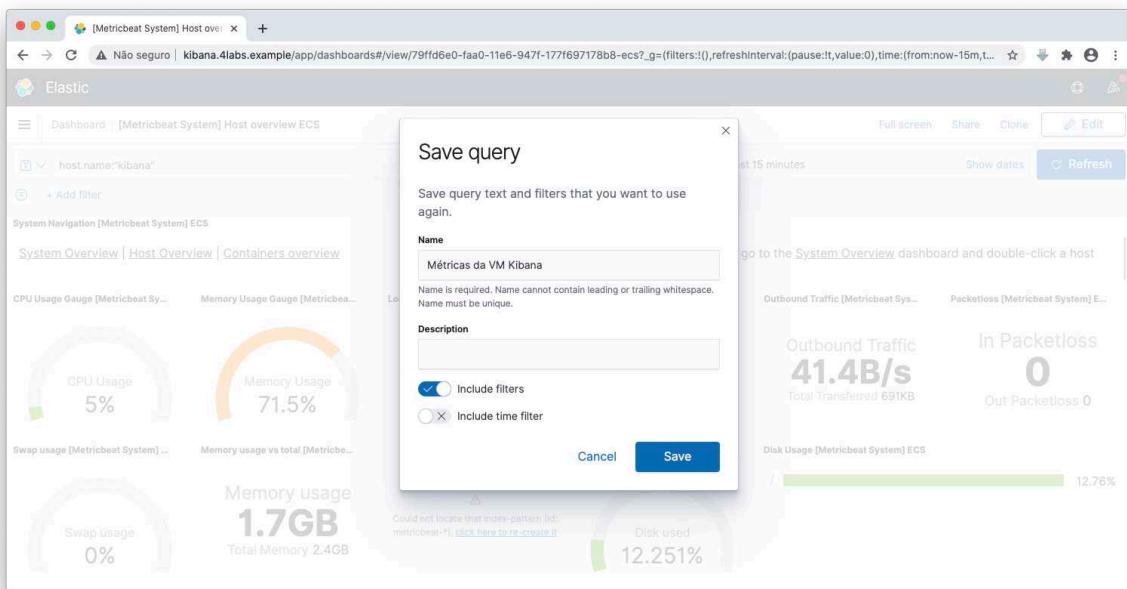


Fig. 5.33: Criando Dashboard ETAPA 6

Mais informações:

[1] Inputs: <https://elastic.co/guide/en/logstash/current/input-plugins.html> [2] Grok: <https://elastic.co/guide/en/logstash/current/filters-grok.html> [3] Filter plugins: <https://elastic.co/guide/en/logstash/current/filter-plugins.html> [4] Plugins de saída do Elasticsearch: <https://elastic.co/guide/en/logstash/current/plugins-outputs-elasticsearch.html> [5] Beats da comunidade: <https://www.elastic.co/guide/en/beats/libbeat/current/beats.html>

6

Gerenciar logs na AWS com o Cloudwatch

Competências deste conteúdo

- Criar uma conta gratuita na AWS
- Introdução ao Cloudwatch
- Criar função CloudWatchFullAccess
- Criar instância na AWS
- Acessar instância na AWS
- Instalar e configurar Cloudwatch Agent
- Visualizar logs da instância no console do Cloudwatch
- Configurar e visualizar logs de um servidor Web
- Configurar e visualizar logs de containers

AWS – Conta gratuita

Amazon Web Services

Para criar uma conta no AWS, é necessário: - Possuir um cartão de crédito, de preferência internacional - Link para a conta grátis por 1 ano do Amazon Web Services: <https://aws.amazon.com/pt/free> - Clique no botão **Crie uma conta gratuita**.

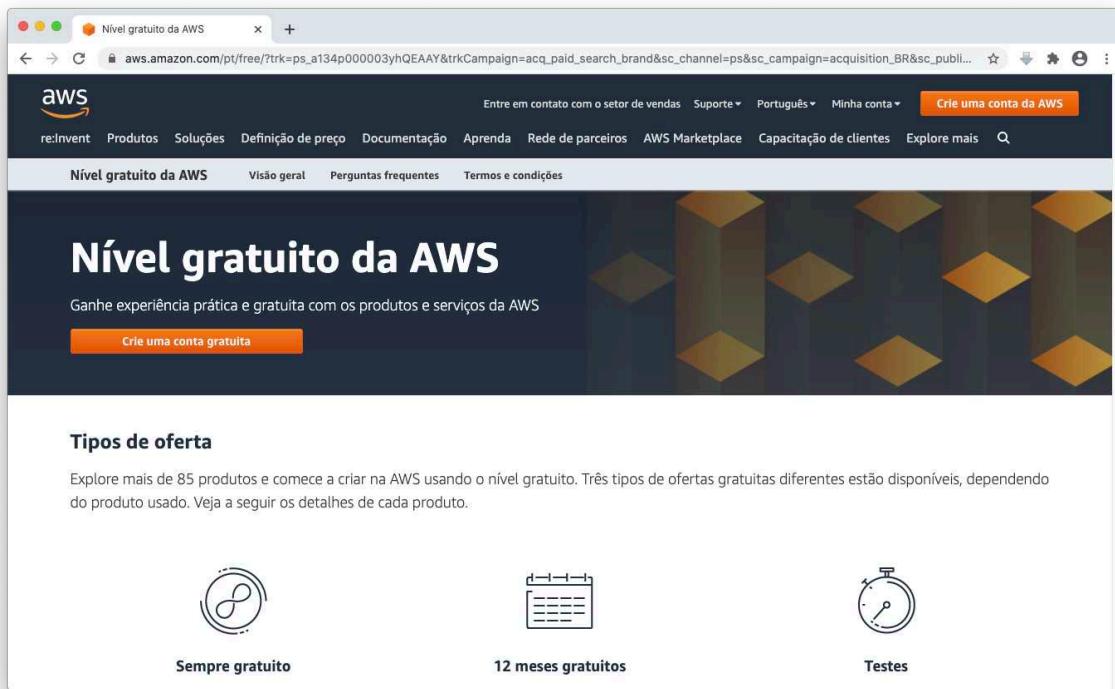


Fig. 6.1: AWS - Conta gratuita

Introdução ao CloudWatch

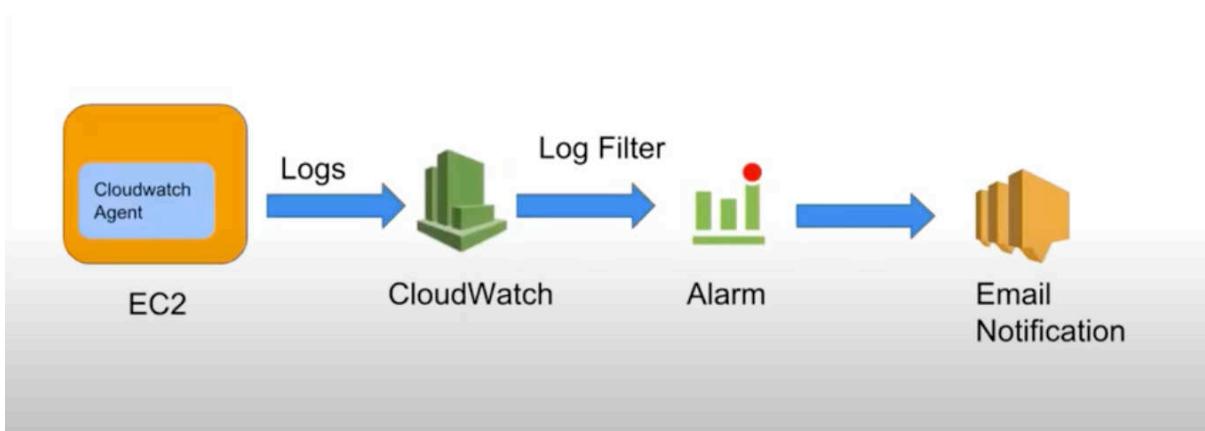


Fig. 6.2: Arquitetura do Cloudwatch

O Amazon CloudWatch é um serviço de monitoramento e observação de instâncias na cloud da AWS. O CloudWatch permite realizar as seguintes tarefas:

- Fornecer dados e insights práticos para monitorar aplicativos;
- Permite responder às alterações de performance em todo o sistema;

- Otimizar a utilização de recursos e obter uma visualização unificada da integridade operacional;
- Coleta dados de monitoramento e operações na forma de logs;
- Permite o uso de alarmes e notificação via e-mail;
- Visualização unificada dos recursos de aplicativos e dos serviços da AWS executados na AWS e em servidores locais.

Fonte: <https://aws.amazon.com/>

Criar função CloudWatchFullAccess

Criar função para Cloudwatch

Uma função do IAM permite de forma segura, conceder permissões para entidade que você confia.

Acesse a console da AWS e clique em **Services > Segurança, Identidade E Conformidade > IAM**

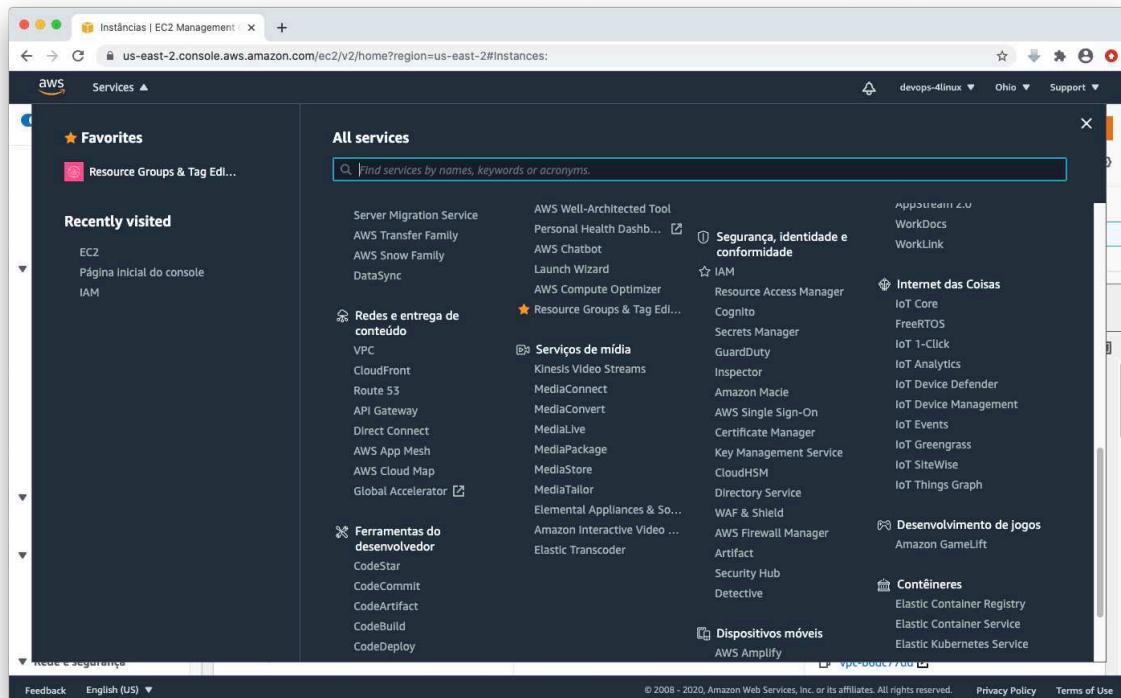


Fig. 6.3: Criar função Cloudwatch - ETAPA 1

Clique em **Funções** ao lado esquerdo da tela.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

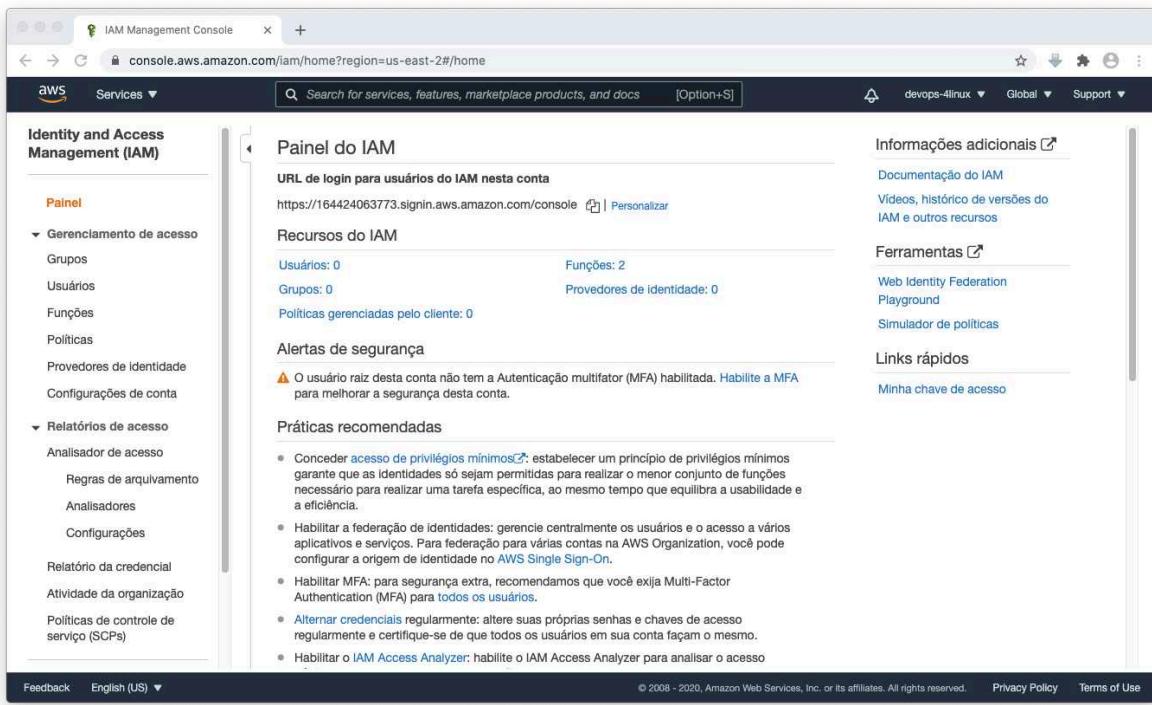


Fig. 6.4: Criar função Cloudwatch - ETAPA 2

Em seguida clique no botão **Criar função**.

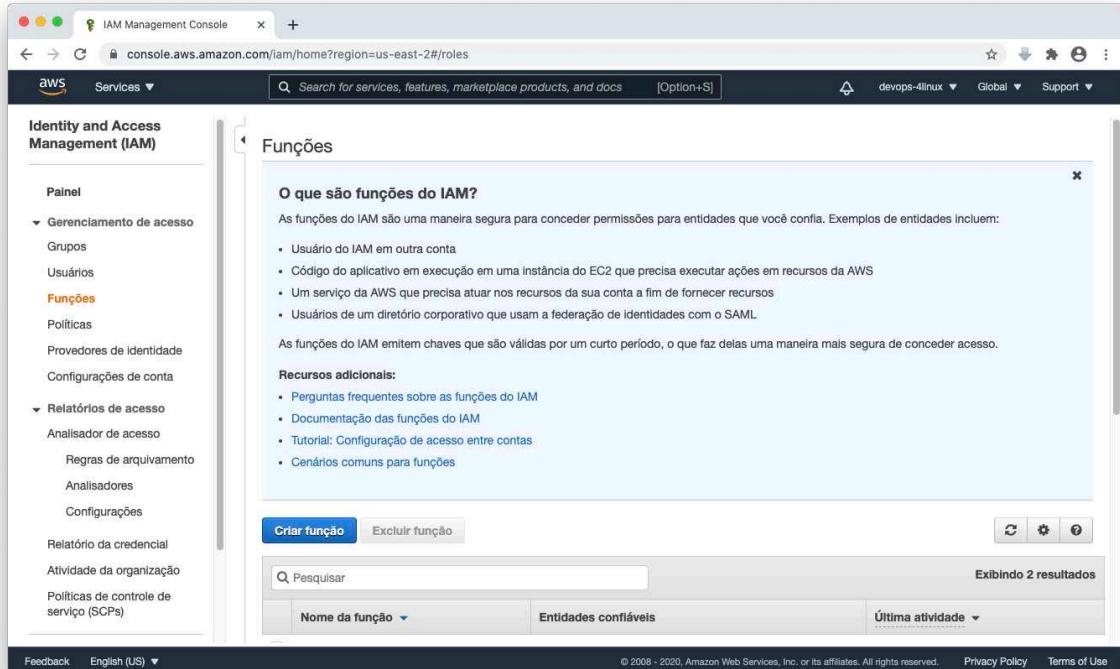


Fig. 6.5: Criar função Cloudwatch - ETAPA 3

Selecione em **Caso de uso comuns**, a opção **EC2** e no botão **Próximo: Permissões**.

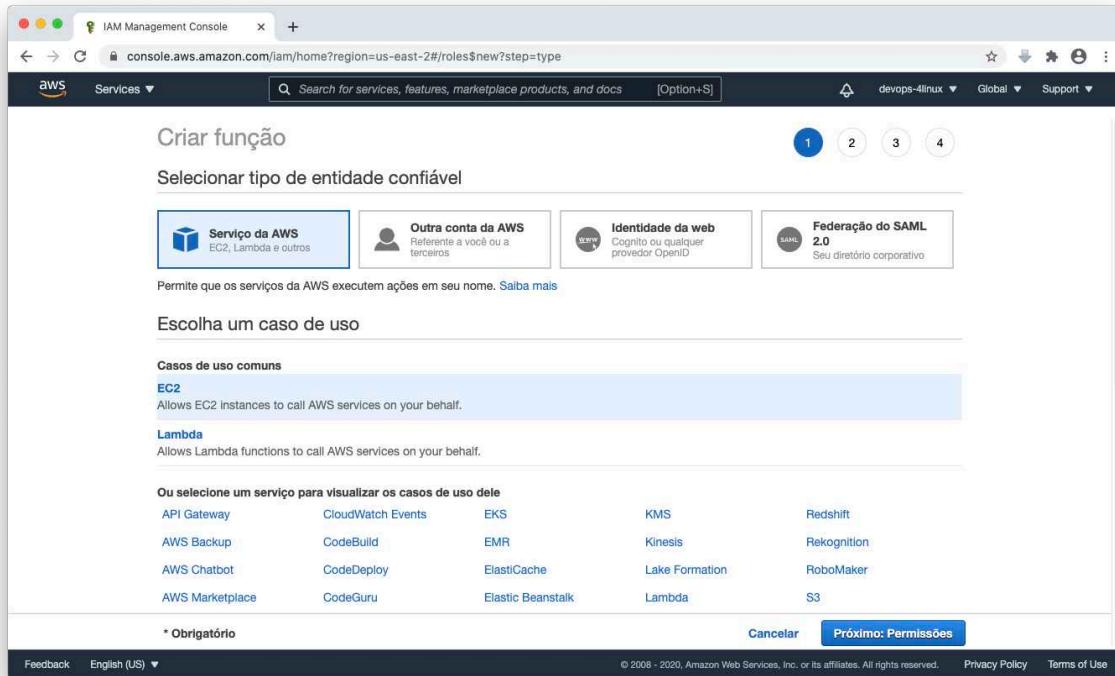


Fig. 6.6: Criar função Cloudwatch - ETAPA 4

Na caixa **Filtrar política**, digite **cloudwatchfull** e selecione a política **CloudWatchFullAccess** e clique em **Próximo: Tags**

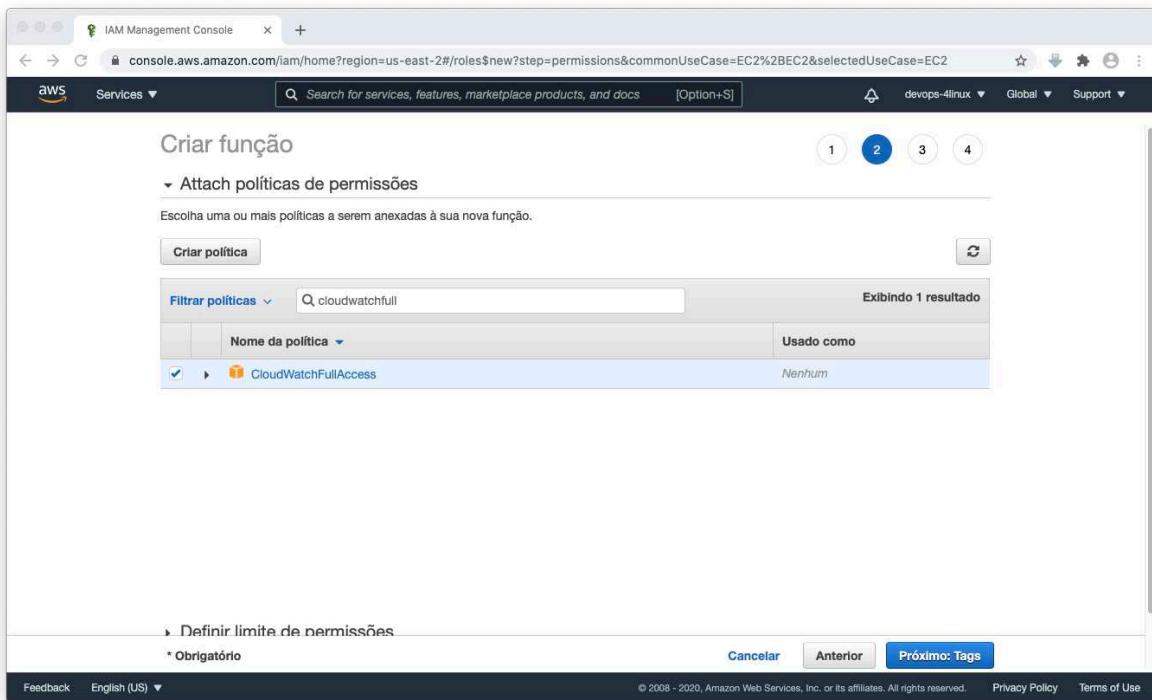


Fig. 6.7: Criar função Cloudwatch - ETAPA 5

Em seguida clique em **Próximo: Revisar**.

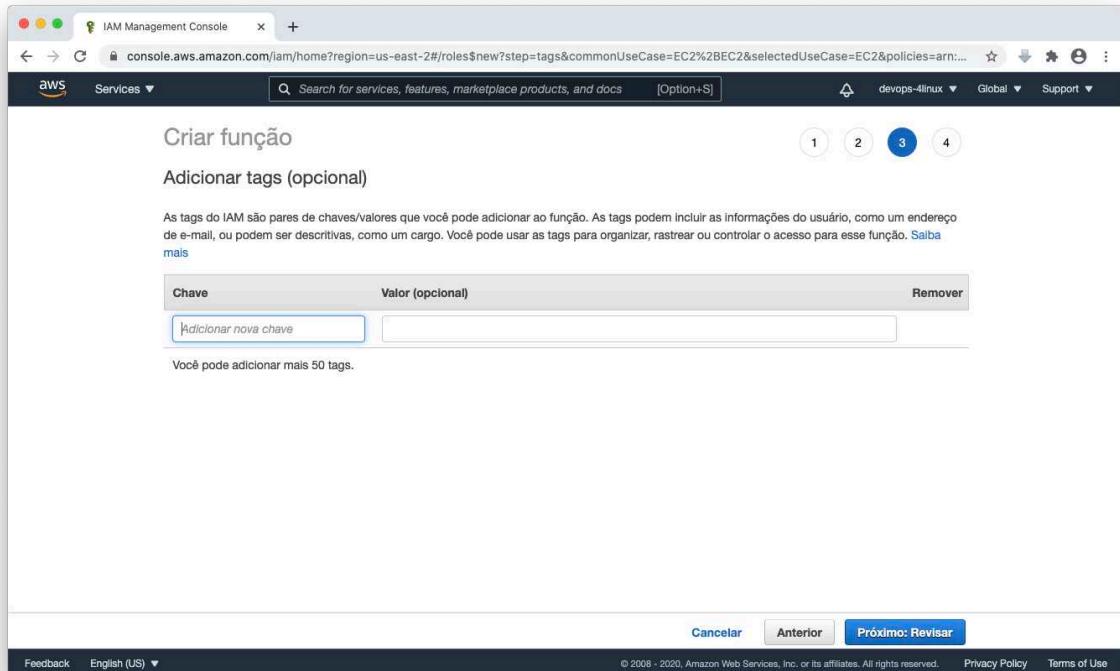


Fig. 6.8: Criar função Cloudwatch - ETAPA 6

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

E preencha a caixa **Nome da função**, o nome **ec2-role-cloudwatch**. Para terminar clique no botão **Criar função**.

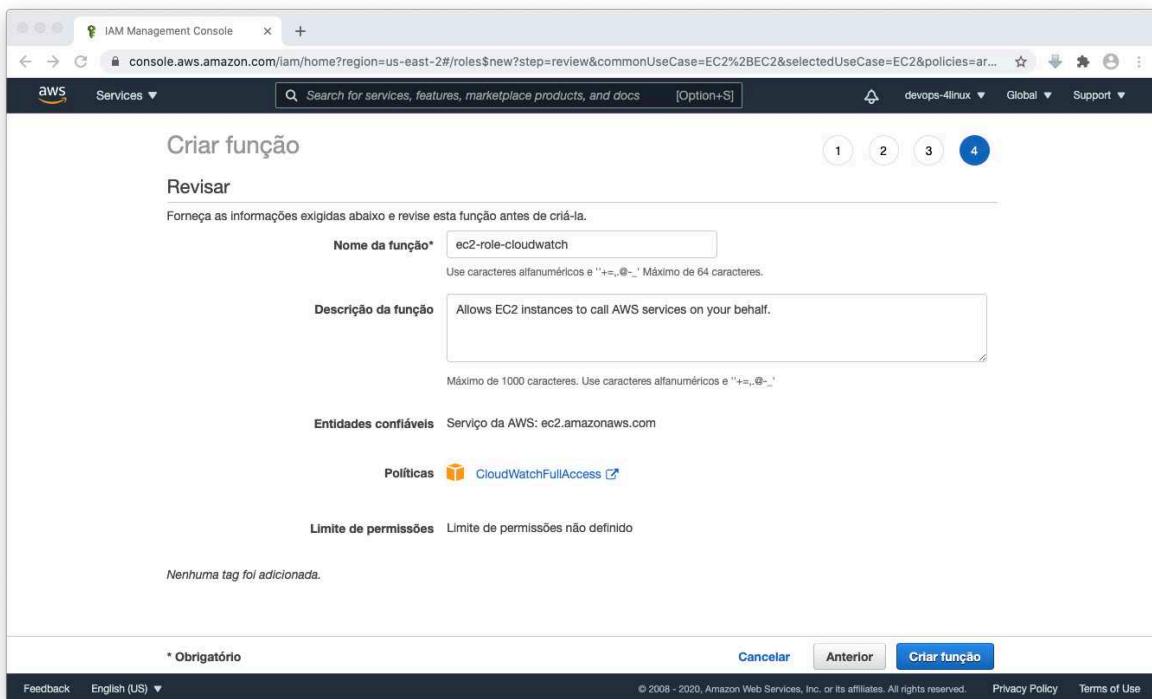


Fig. 6.9: Criar função Cloudwatch - ETAPA 7

Gerenciar instâncias na AWS

Criar uma instância na AWS

Vamos criar uma instância EC2, para isso clique em **Painel EC2** ao lado esquerdo da tela.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

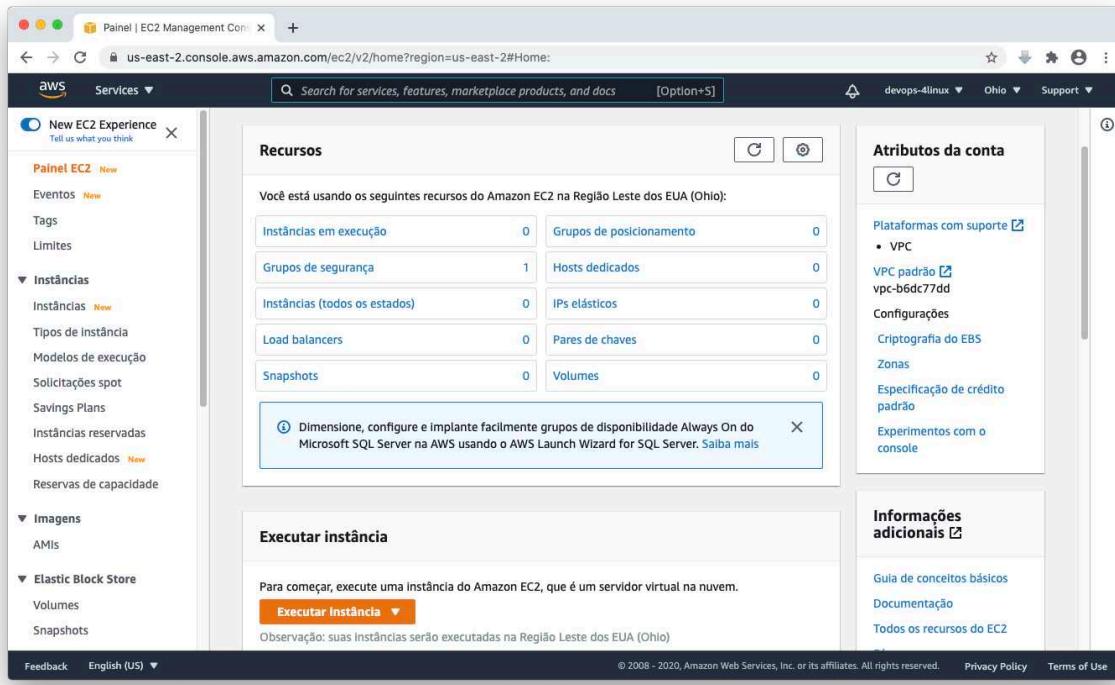


Fig. 6.10: Gerenciar instâncias na AWS - ETAPA 1

Clique em **Executar instância > Executar instância.**

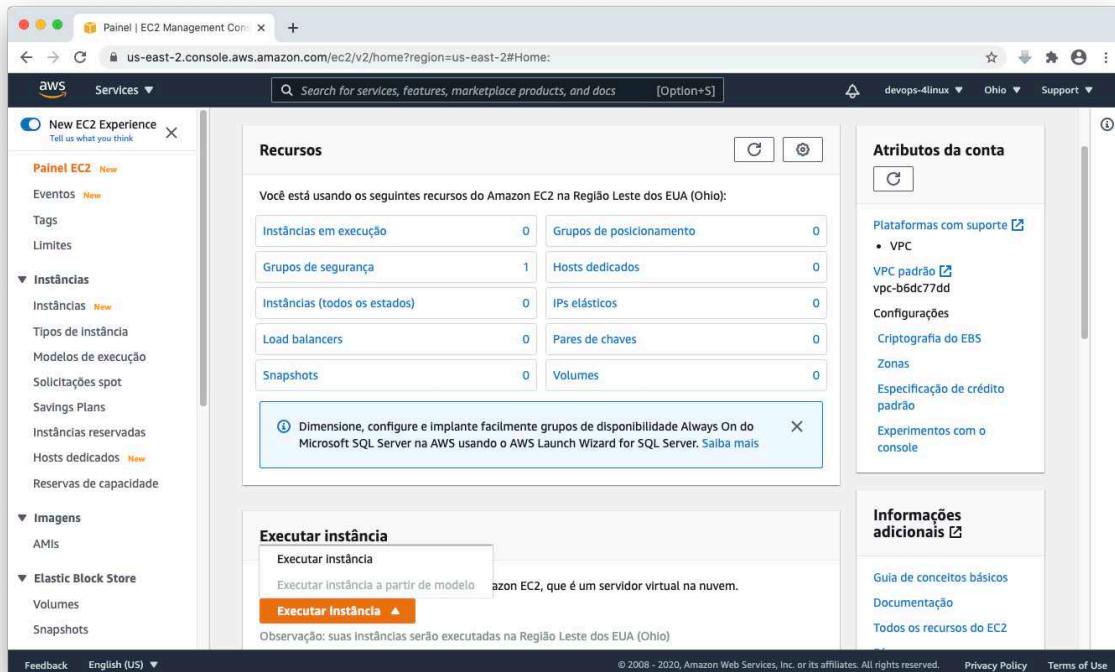


Fig. 6.11: Gerenciar instâncias na AWS - ETAPA 2

Selecione a imagem **Amazon Linux 2 AMI (HVM), SSD Volume Type**, e clique em **Select**:

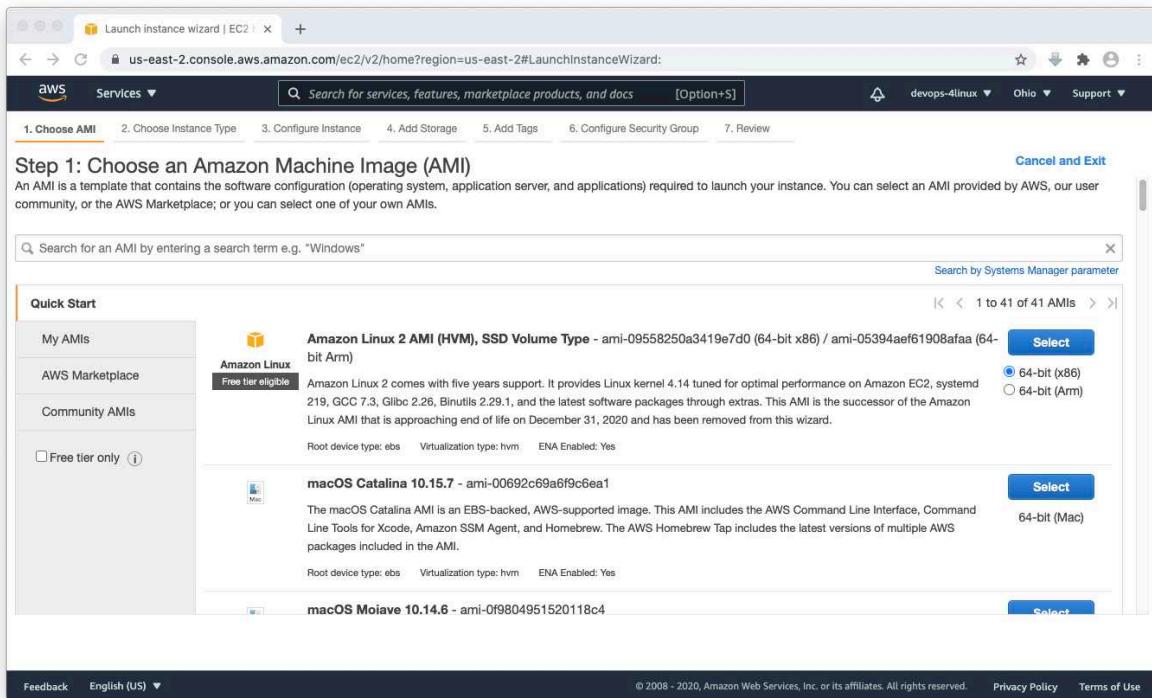


Fig. 6.12: Gerenciar instâncias na AWS - ETAPA 3

Na próxima tela basta clicar em **Next: Configure Instance Details**.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All Instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (~ ECUs, 1 vCPUs, 2.5 GHz, ~ 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) ▾ © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Fig. 6.13: Gerenciar instâncias na AWS - ETAPA 4

Em seguida selecione em **IAM role**, a role **ec2-role-cloudwatch**. Agora basta clicar em **Next: Add Storage**.

Step 3: Configure Instance Details

Subnet (No preference (default subnet in any Availability Zone)) Create new subnet

Auto-assign Public IP (Use subnet setting (Enable))

Placement group (checkbox) Add instance to placement group

Capacity Reservation (Open)

Domain join directory (No directory) Create new directory

IAM role (ec2-role-cloudwatch) Create new IAM role

CPU options (checkbox) Specify CPU options

Shutdown behavior (Stop)

Stop - Hibernate behavior (checkbox) Enable hibernation as an additional stop behavior

Enable termination protection (checkbox) Protect against accidental termination

Monitoring (checkbox) Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy (Shared - Run a shared hardware instance) Additional charges will apply for dedicated tenancy.

Cancel Previous Review and Launch Next: Add Storage

Feedback English (US) ▾ © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Fig. 6.14: Gerenciar instâncias na AWS - ETAPA 5

Próxima tela basta clicar em **Next: Add Tags**.

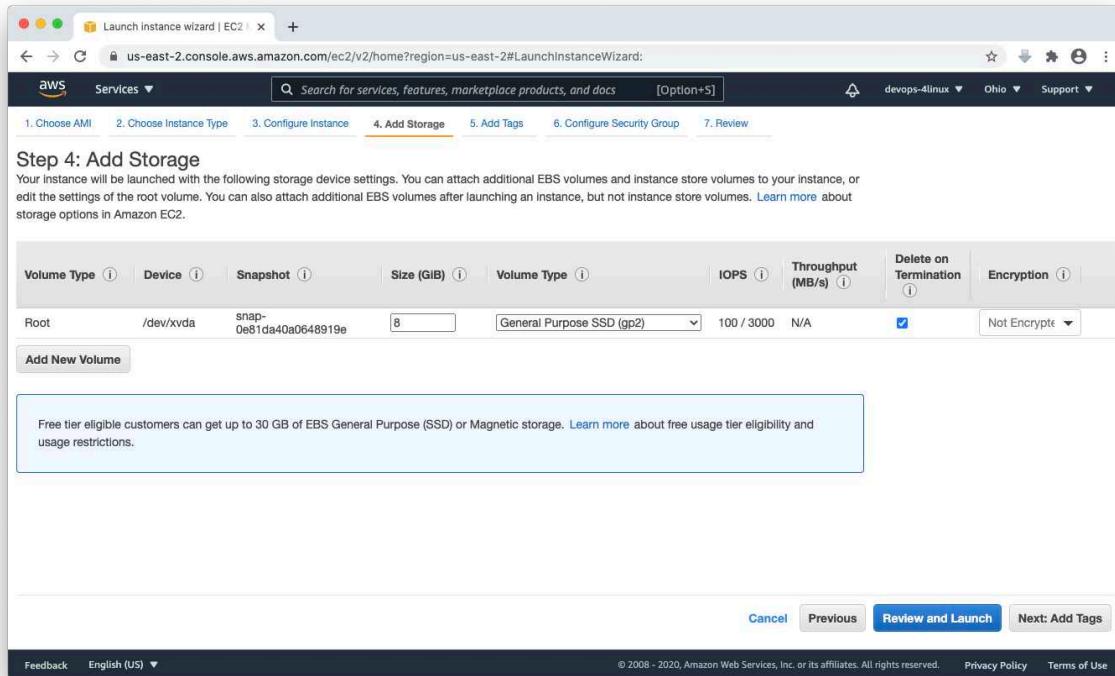


Fig. 6.15: Gerenciar instâncias na AWS - ETAPA 6

Próxima tela basta clicar em **Next: Configure Security Group**.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

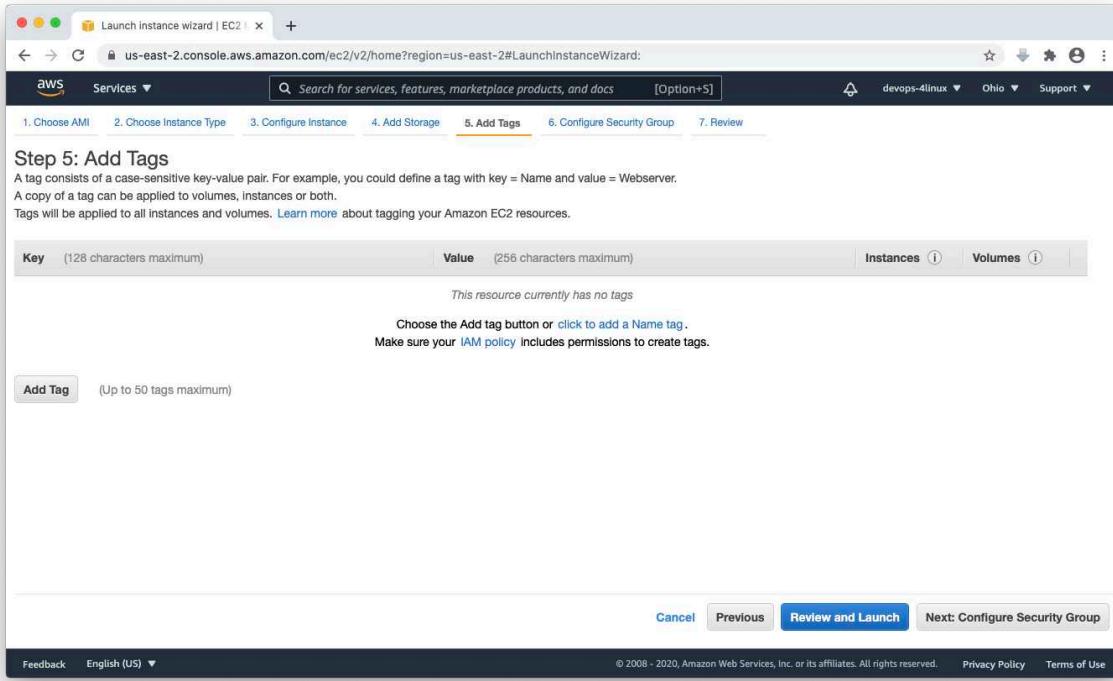


Fig. 6.16: Gerenciar instâncias na AWS - ETAPA 7

Através do botão **Add Rule**, selecione **HTTP** e clique no botão **Review and Launch**.

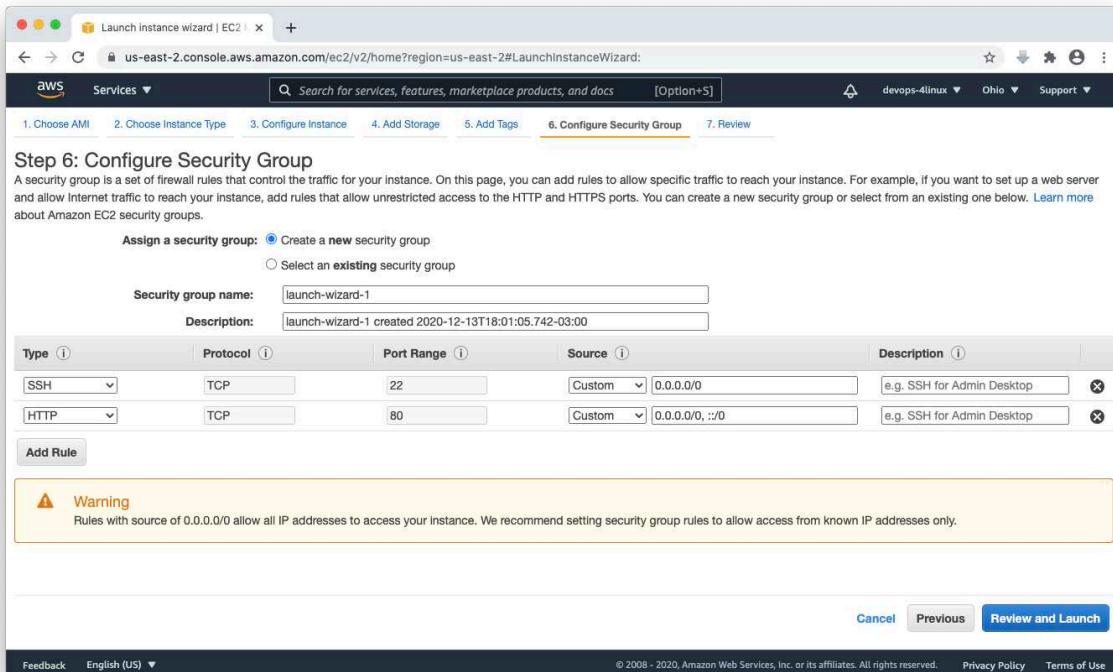


Fig. 6.17: Gerenciar instâncias na AWS - ETAPA 8

Próxima tela basta clicar em **Launch**:

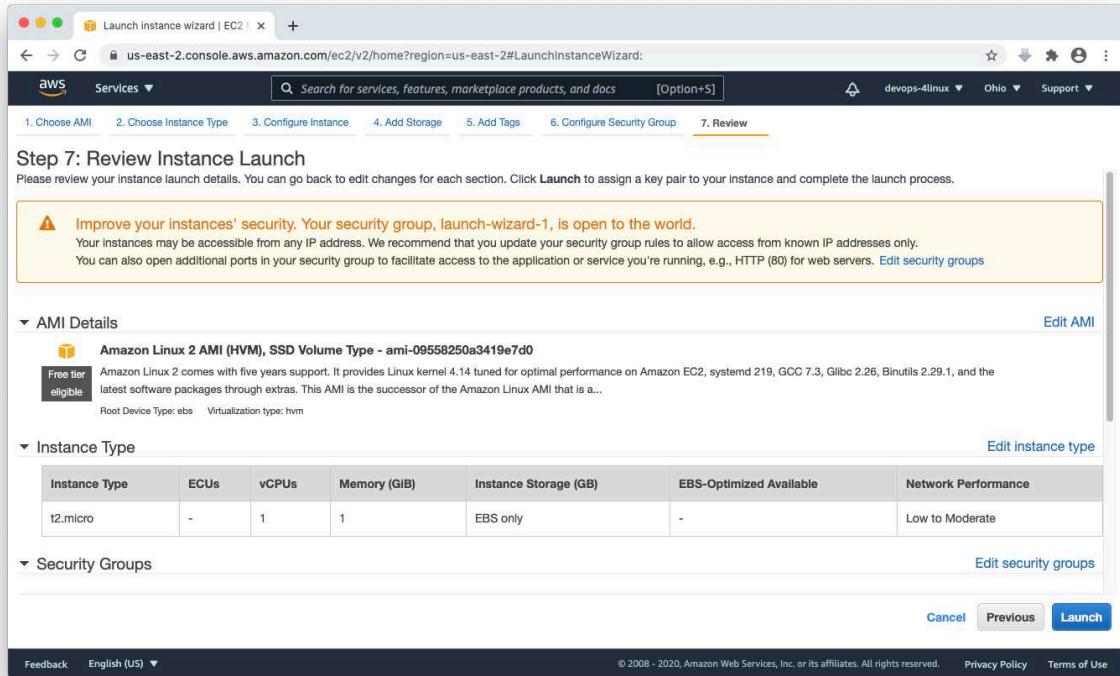


Fig. 6.18: Gerenciar instâncias na AWS - ETAPA 9

No próximo passo, vamos criar uma **key pair**, selecione **Create a new key pair** e em Key pair name preencha **cloudwatch**. Clique em **Download Key Pair** e em seguida clique em **Launch Instances**:

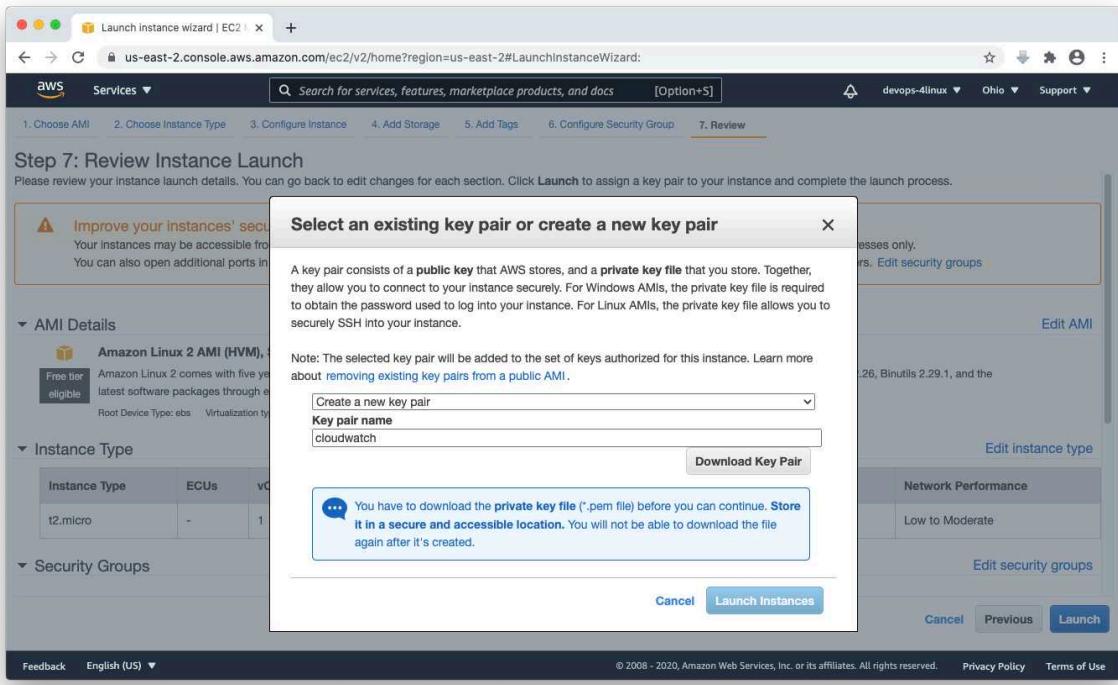


Fig. 6.19: Gerenciar instâncias na AWS - ETAPA 10

Para visualizar o estado da instância criada, clique em **View Instances**:

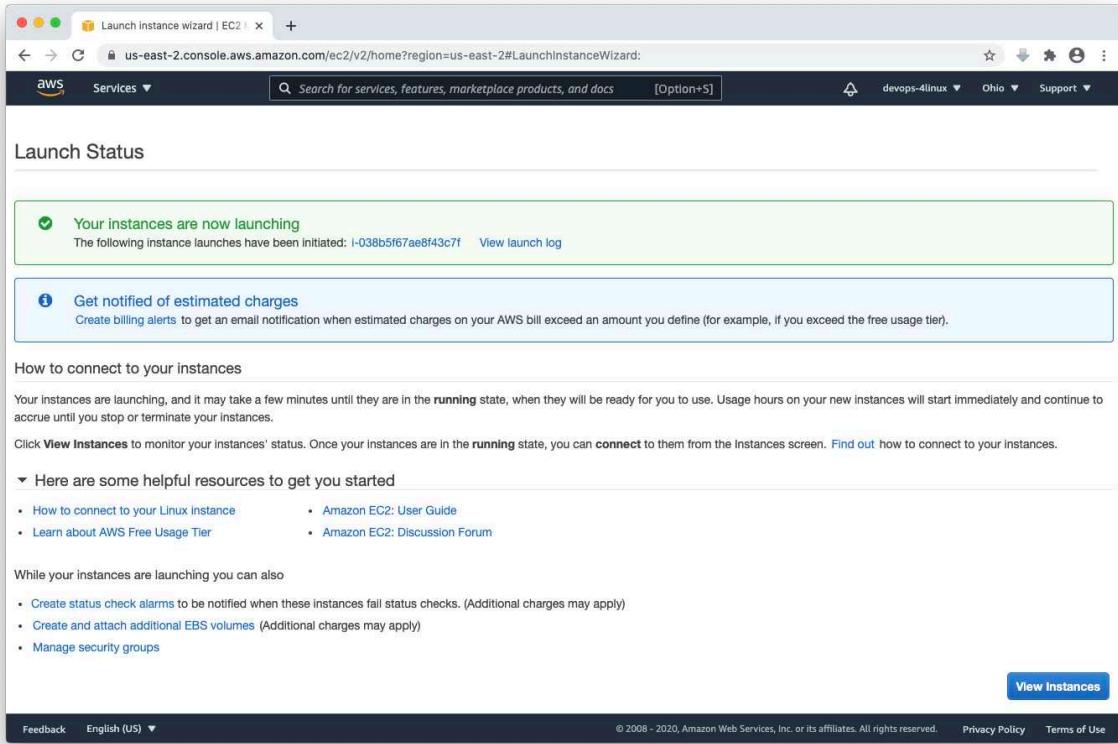


Fig. 6.20: Gerenciar instâncias na AWS - ETAPA 11

Acessar uma instância da AWS

Selecione a instância criada e clique em **Conectar**.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

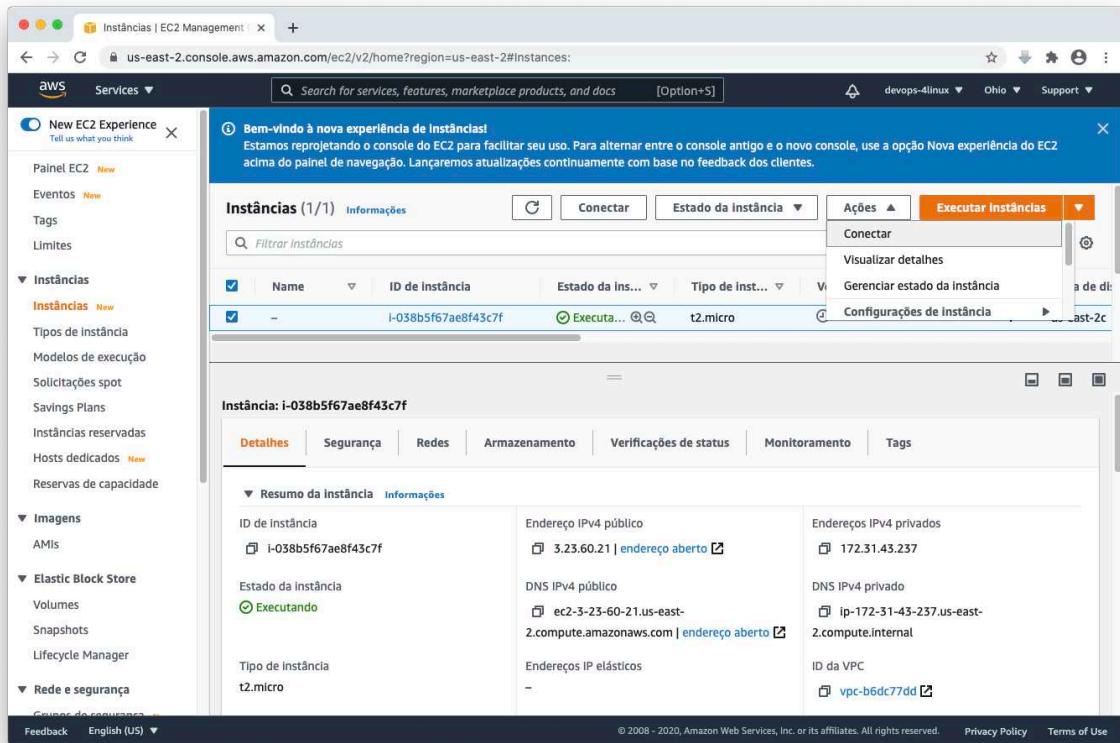


Fig. 6.21: Acessar instância na AWS - ETAPA 1

Selecione **Conexão de Instância do EC2** e clique no botão **Conectar**.

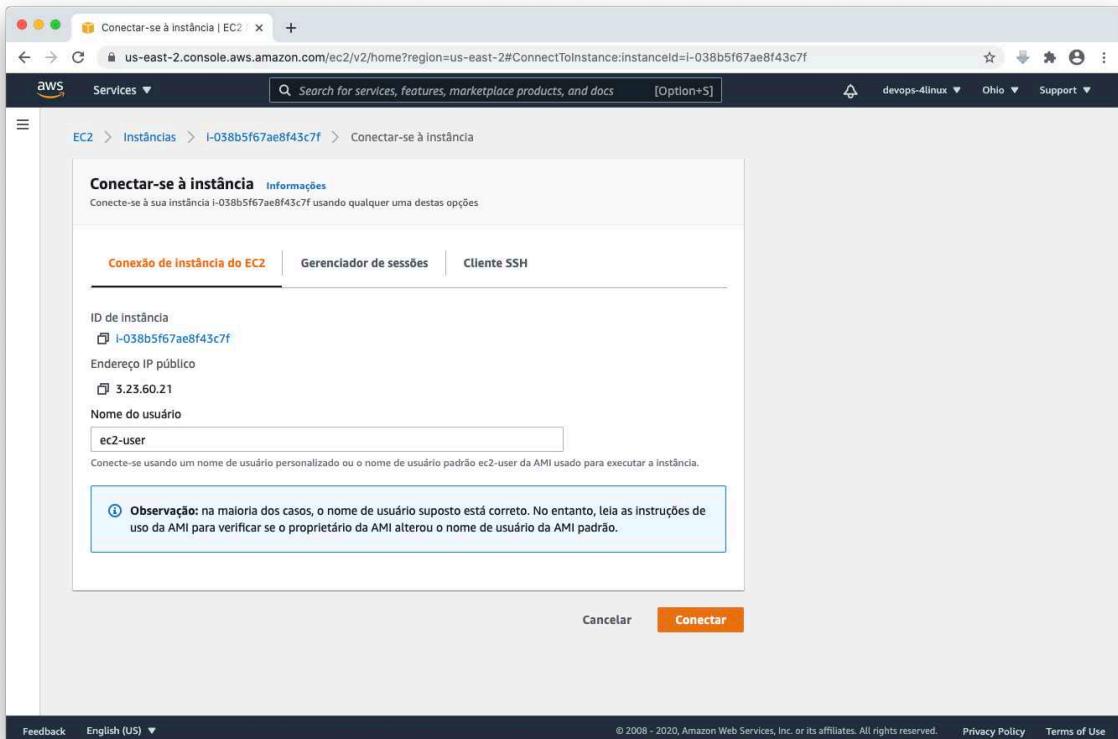


Fig. 6.22: Acessar instância na AWS - ETAPA 2

Instalar e configurar o Cloudwatch Agent

Instale o pacote **awslogs**:

```
1 | sudo yum install awslogs -y
```

Em seguida edite o arquivo **awscli.conf** na pasta **/etc/awslogs**, para definir em qual região sua instância esta sendo executada.

```
1 | sudo vim /etc/awslogs/awscli.conf
2 | [plugins]
3 | cwlogs = cwlogs
4 | [default]
5 | region = us-east-2
```

O próximo arquivo está configurado por padrão, para visualizar logs do arquivo **/var/log/messages**. Precisamos editar para definir o ID se nossa instância:

```

1 | sudo vim /etc/awslogs/awslogs.conf +$  

2 | ....  

3 | [/var/log/messages]  

4 | datetime_format = %b %d %H:%M:%S  

5 | file = /var/log/messages  

6 | buffer_duration = 5000  

7 | log_stream_name = i-03a946baa04b49efb  

8 | initial_position = start_of_file  

9 | log_group_name = /var/log/messages

```

Copie o ID que aparece na parte inferior da tela:

```

# %z      UTC offset in the form +HHMM or -HHMM (empty string if the object is naive).          (empty), +0000, -0
# 400, +1030
#
# %j      Day of the year as a zero-padded decimal number.                                     001, 002, ..., 365
#
# %U      Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0.   00, 01, ..., 53
#
# %W      Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0.           00, 01, ..., 53
#
# %c      Locale's appropriate date and time representation.                                Tue Aug 16 21:30:0
0 1988 (en_US)
#



[ /var/log/messages ]
datetime_format = %b %d %H:%M:%S
file = /var/log/messages
buffer_duration = 5000
log_stream_name = i-03a946baa04b49efb
initial_position = start_of_file
log_group_name = /var/log/messages

i-03a946baa04b49efb
Public IPs: 18.223.15.89  Private IPs: 172.31.37.28

```

Fig. 6.23: Configuração do arquivo /etc/awslogs/awslogs.conf

Inicie e ative na inicialização do sistema o serviço do **Cloudwatch Agent**.

```

1 | sudo systemctl start awslogsd
2 | sudo systemctl enable awslogsd

```

Visualizar logs da instância no console do Cloudwatch

Para visualizar os logs de nossa instância, clique em **Services** > Gerenciamento e governança > **CloudWatch**. Ao lado esquerdo da tela, clique em **Grupos de logs**.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

The screenshot shows the AWS CloudWatch Management Console with the 'CloudWatch: Visão geral' (General View) selected. The left sidebar is open, showing categories like CloudWatch, Painéis, Alarmes, Logs, Métricas, and Eventos. The main content area displays service alarms for CloudWatch Logs, EC2, Elastic Block Store, and Usage. It also shows a section for recent alarms and default dashboards.

Fig. 6.24: Acessar logs da instância no CloudWatch - ETAPA 1

Selecione o grupo **/var/log/messages**.

The screenshot shows the AWS CloudWatch Management Console with the 'CloudWatch Logs > Log groups' selected. The left sidebar is open, showing categories like CloudWatch, Painéis, Alarmes, Logs, and Grupos de logs. The main content area displays a list of log groups, with one group named '/var/log/messages' visible.

Fig. 6.25: Acessar logs da instância no CloudWatch - ETAPA 2

E selecione o **ID de nossa instância** na coluna **Log stream**.

The screenshot shows the AWS CloudWatch Management Console interface. On the left, the navigation menu is expanded to show the 'Logs' section, specifically the 'Grupos de logs' (Log Groups) option. The main content area displays the details for the '/var/log/messages' log group. At the top of this section, there are buttons for 'Ações' (Actions), 'Visualizar no Logs Insights' (View in Logs Insights), and a search bar labeled 'Search log group'. Below this, a table provides detailed information about the log group, including retention settings (never expire, 3 minutes ago), metrics filters (0), and log storage (0 bytes). An ARN (Amazon Resource Name) is also listed. A 'Streams de log' (Log Streams) tab is selected, showing one stream named 'Log stream' with the ID 'I-038b5f67ae8f43c7f' and a timestamp of '2020-12-13 18:14:55 (UTC-03:00)'.

Fig. 6.26: Acessar logs da instância no CloudWatch - ETAPA 3

Como resultado final é possível visualizar os logs do arquivo **/var/log/messages**.

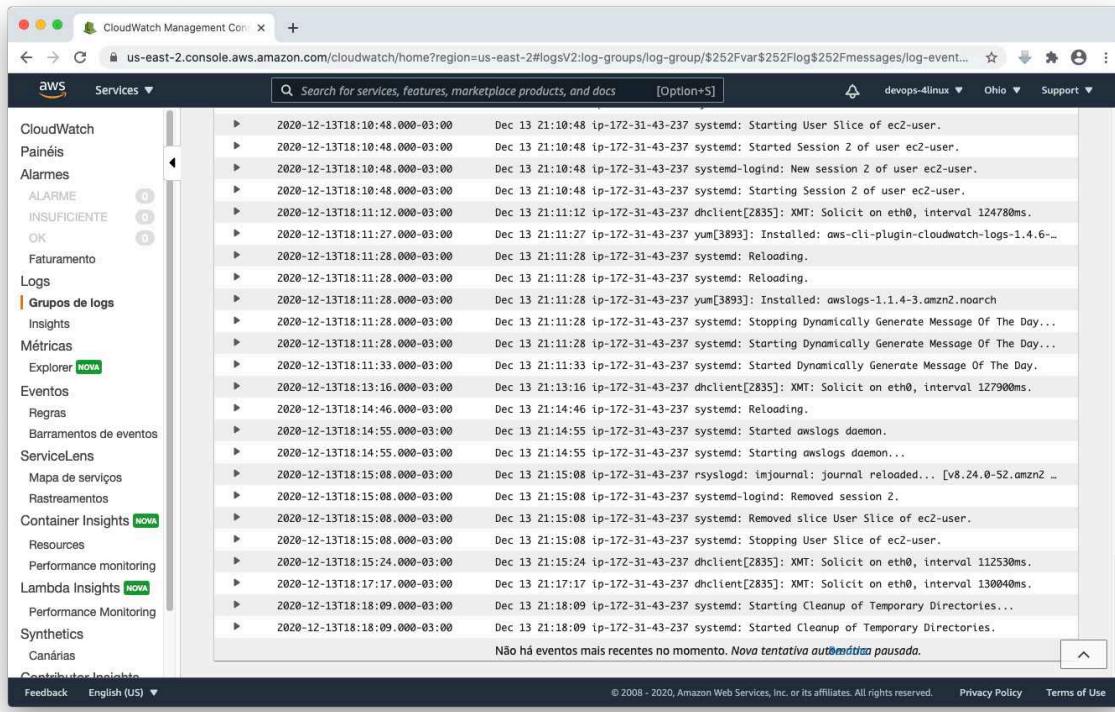


Fig. 6.27: Acessar logs da instância no CloudWatch - ETAPA 4

Configurar e visualizar logs de um servidor web

Instale o pacote **httpd**, inicie e ative na inicialização do sistema o seu serviço:

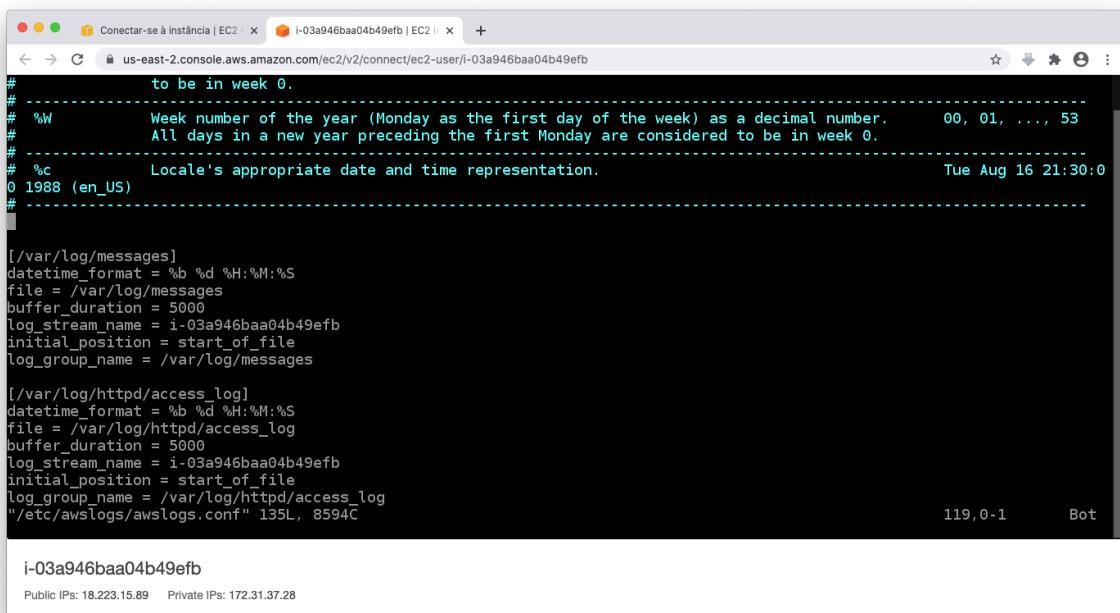
```
1 | sudo yum install httpd -y
2 |
3 | sudo systemctl start httpd
4 | sudo systemctl enable httpd
```

Em seguida edite e adicione no final do arquivo **/etc/awslogs/awslogs.conf**, o bloco de configuração para visualizar os logs de acesso de nosso servidor web.

```
1 | sudo vim /etc/awslogs/awslogs.conf +$
2 | ....
3 |
4 | [ /var/log/httpd/access_log ]
5 | datetime_format = %b %d %H:%M:%S
6 | file = /var/log/httpd/access_log
7 | buffer_duration = 5000
```

```
8 | log_stream_name = i-03a946baa04b49efb
9 | initial_position = start_of_file
10| log_group_name = /var/log/httpd/access_log
```

Copie o ID que aparece na parte inferior da tela:



```
# to be in week 0.
#-----#
# %W Week number of the year (Monday as the first day of the week) as a decimal number. 00, 01, ..., 53
# All days in a new year preceding the first Monday are considered to be in week 0.
#-----#
# %c Locale's appropriate date and time representation. Tue Aug 16 21:30:00
#-----#
# 1988 (en_US)

[ /var/log/messages ]
datetime_format = %b %d %H:%M:%S
file = /var/log/messages
buffer_duration = 5000
log_stream_name = i-03a946baa04b49efb
initial_position = start_of_file
log_group_name = /var/log/messages

[ /var/log/httpd/access_log ]
datetime_format = %b %d %H:%M:%S
file = /var/log/httpd/access_log
buffer_duration = 5000
log_stream_name = i-03a946baa04b49efb
initial_position = start_of_file
log_group_name = /var/log/httpd/access_log
"/etc/awsLogs/awslogs.conf" 135L, 8594C
```

i-03a946baa04b49efb
Public IPs: 18.223.15.89 Private IPs: 172.31.37.28

Fig. 6.28: Adicionar captura de logs do servidor Web no arquivo /etc/awslogs/awslogs.conf

Reinicie o serviço do **Cloudwatch Agent** para aplicar as novas configurações.

```
1 | sudo systemctl restart awslogsd
```

Visualizar logs do servidor Web

Antes de visualizar os logs do servidor web, é preciso acessá-lo através do **IP público de sua instância**.

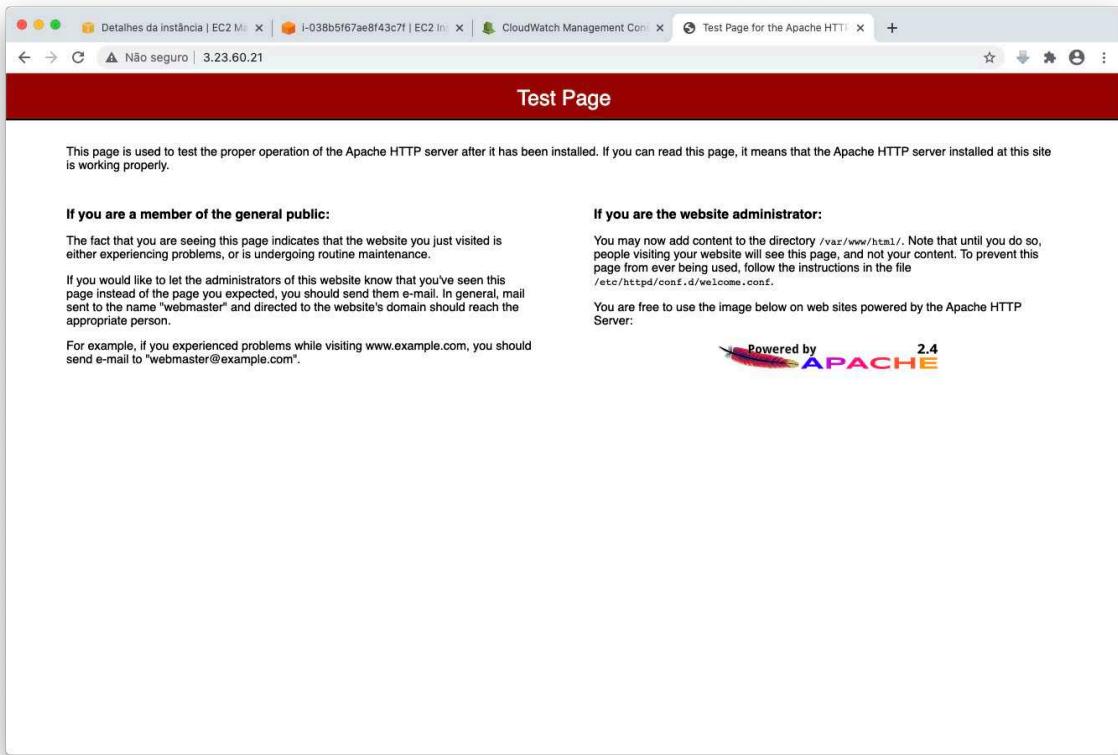


Fig. 6.29: Acessar servidor Web na instância da AWs

Retorne para a página do CloudWatch e clique novamente em **Grupos de logs**. A página será atualizada com o novo grupo **/var/log/httpd/acess_log**, onde devemos selecionar.

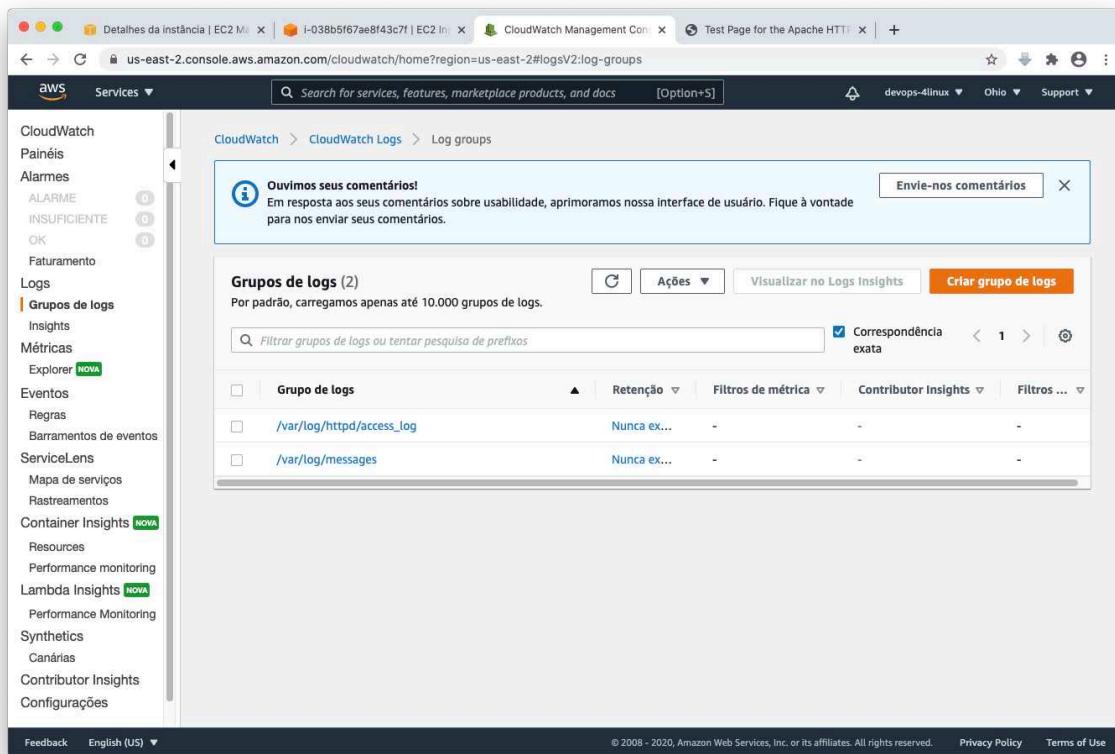


Fig. 6.30: Acessar logs da servidor Web no CloudWatch - ETAPA 1

E selecione o **ID de nossa instância** na coluna **Log stream**.

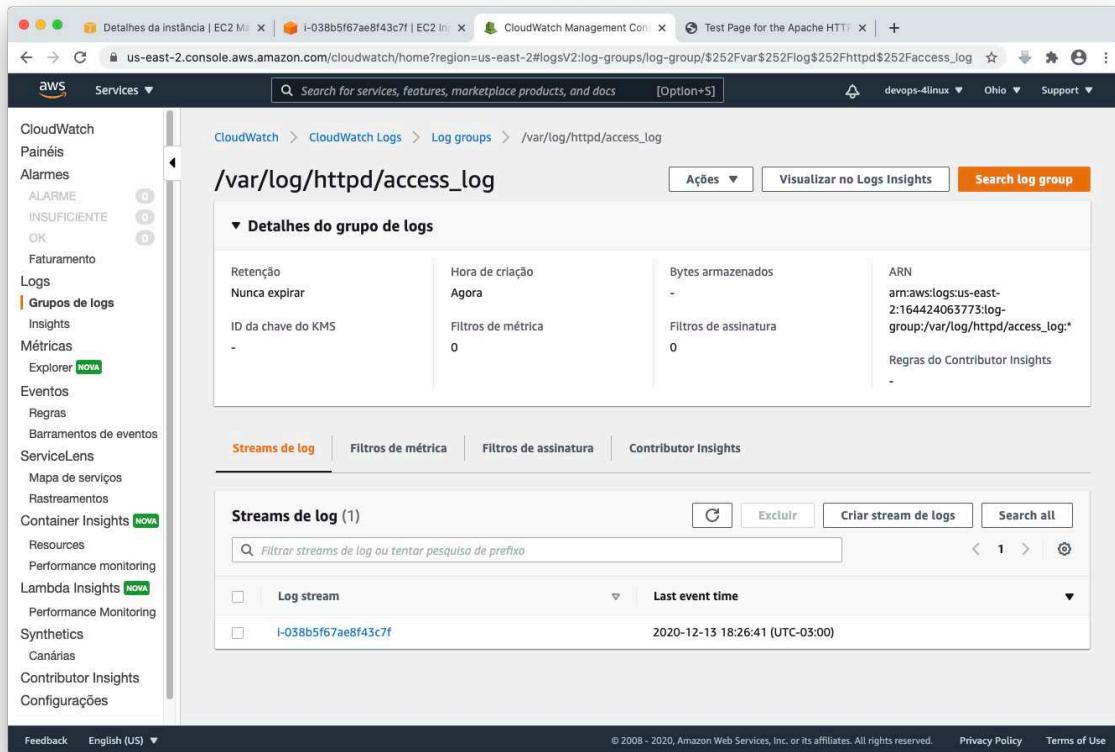


Fig. 6.31: Acessar logs da servidor Web no CloudWatch - ETAPA 2

Como resultado final é possível visualizar os **logs de acesso de nosso servidor web**.

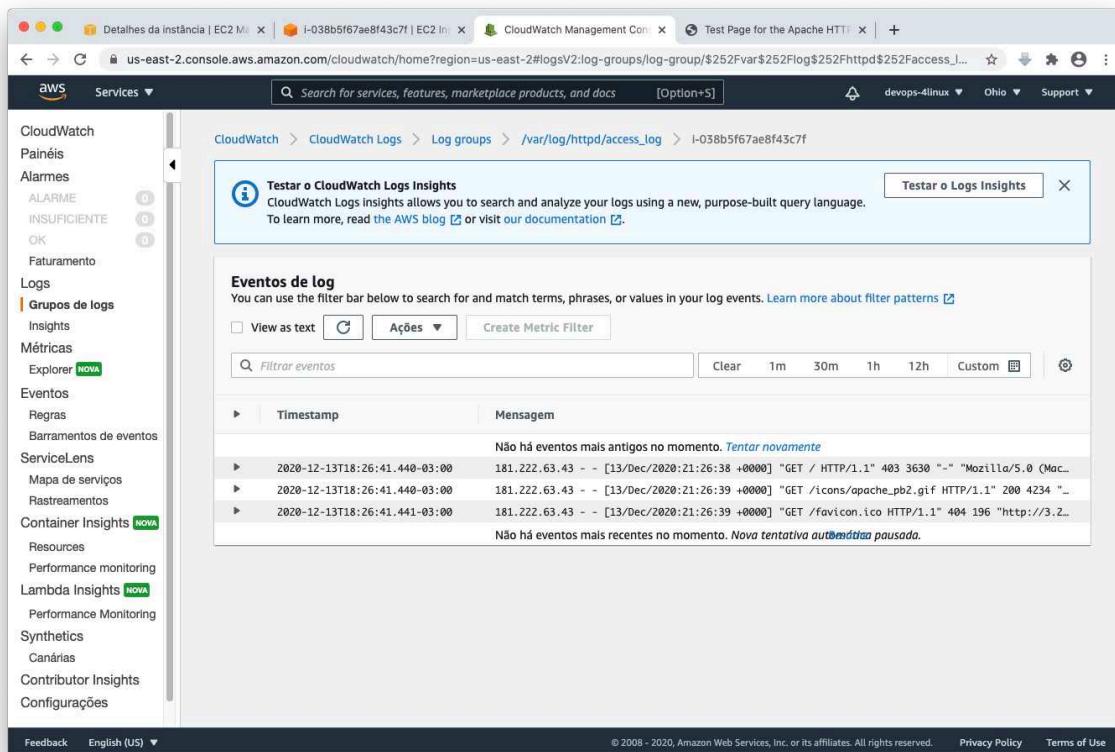


Fig. 6.32: Acessar logs da servidor Web no CloudWatch - ETAPA 3

Visualizar logs de container no Docker

Para a instalação do Docker na instância é preciso seguir os seguintes passos:

1. Atualizar lista de pacotes repositórios do sistema
2. Instalar o Docker
3. Iniciar o serviço do Docker

Conforme a sequência de comandos a seguir:

```
1 | sudo yum update -y
2 | sudo yum install docker -y
3 | sudo service docker start
```

Verifique qual e a versão instalada do Docker:

```
1 | sudo docker version
```

Antes de executar um container do **Nginx**, pare o serviço do Apache que está utilizando a **porta 80**:

```
1 | sudo systemctl stop httpd
```

Em seguida execute um container do **Nginx** informando o driver de log **awslogs**:

```
1 | sudo docker container run -d --name nginx --log-driver=awslogs  
    --log-opt awslogs-region=us-east-2 --log-opt awslogs-group=  
    nginx-logs --log-opt awslogs-create-group=true -p 80:80  
    nginx
```

Acesse a página do Nginx através do IP público da VM, com o intuito de gerar logs de acesso do container.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Fig. 6.33: Visualizar logs de Containers no Docker - ETAPA 1

Retorne para a página do CloudWatch e clique novamente em **Grupos de logs**. A página sera atualizada com o novo grupo **nginx-logs**, onde devemos selecionar.

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

The screenshot shows the AWS CloudWatch Management Console interface. On the left, there's a sidebar with various monitoring services like CloudWatch Metrics, CloudWatch Logs, CloudWatch Events, and CloudWatch ServiceLens. The 'Logs' section is selected. The main content area is titled 'Log groups' and shows a list of three log groups: '/var/log/httpd/access_log', '/var/log/messages', and 'nginx-logs'. Each log group entry includes a checkbox, retention settings ('Nunca expirar'), metric filters, and contributor insights filters. A search bar at the top allows filtering by prefix. At the bottom of the list, there are buttons for 'Ações' (Actions), 'Visualizar no Logs Insights' (View in Logs Insights), and 'Criar grupo de logs' (Create log group).

Fig. 6.34: Visualizar logs de Containers no Docker - ETAPA 2

E selecione o **ID** mais recente na coluna **Log stream**.

This screenshot shows the 'nginx-logs' log group details page. The top part displays general information such as retention (Never expire), creation time (2 minutes ago), and storage bytes (0). It also shows the ARN of the log group and contributor insights rules. Below this, the 'Streams de log' section is shown, containing a single log stream entry with the ID '3341eb7ee00b77fb2964688c7da50ee9a5b504dcdbaa42aa33...' and a timestamp of '2020-12-14 14:53:11 (UTC-03:00)'. There are tabs for 'Streams de log', 'Filtros de métrica', 'Filtros de assinatura', and 'Contributor Insights'.

Fig. 6.35: Visualizar logs de Containers no Docker - ETAPA 3

Como resultado final é possível visualizar os logs de acesso de nosso servidor **Nginx**, que está sendo executado dentro de um container no Docker:

6. GERENCIAR LOGS NA AWS COM O CLOUDWATCH

The screenshot shows the AWS CloudWatch Management Console interface. On the left, a sidebar lists various services: CloudWatch (selected), Painéis, Alarms, ALARME, INSUFICIENTE, OK, Faturamento, Logs (selected), Grupos de logs (highlighted in orange), Insights, Métricas, Explorer (NEW), Eventos, Regras, Barramentos de eventos, ServiceLens, Mapa de serviços, Rastreamentos, Container Insights (NEW), Resources, Performance monitoring, and Lambda Insights (NEW). The main area is titled "Eventos de log" and contains a table of log events. The table has two columns: "Timestamp" and "Mensagem". The "Timestamp" column shows dates from December 14, 2020, at 11:14:53 to 11:46:32. The "Mensagem" column displays log messages related to Docker and nginx configuration. A message at the top states: "Não há eventos mais antigos no momento. Tentar novamente". A message at the bottom states: "Não há eventos mais recentes no momento. Nova tentativa autenticação pausada." The interface includes a search bar, filter buttons, and time range selection (Clear, 1m, 30m, 1h, 12h, Custom).

Fig. 6.36: Visualizar logs de Containers no Docker - ETAPA 4

7

Gerenciar logs na GCP com o Stackdriver

Competências deste conteúdo

- Criar conta gratuita na GCP
- Introdução ao Stackdriver
- Criar instância na GCP
- Acessar instância na GCP
- Instalar e configurar o agente do Cloud Logging
- Visualizar logs da instância no console do Stackdriver
- Configurar e visualizar logs de containers

GCP - Conta gratuita

Google Cloud Platform

Para criar uma conta na GCP é necessário:
- Possuir um cartão de crédito, de preferência internacional
- Link para conta grátis do Google Cloud com crédito de \$300: <https://cloud.google.com/>
- Clique no botão **Comece a usar gratuitamente**

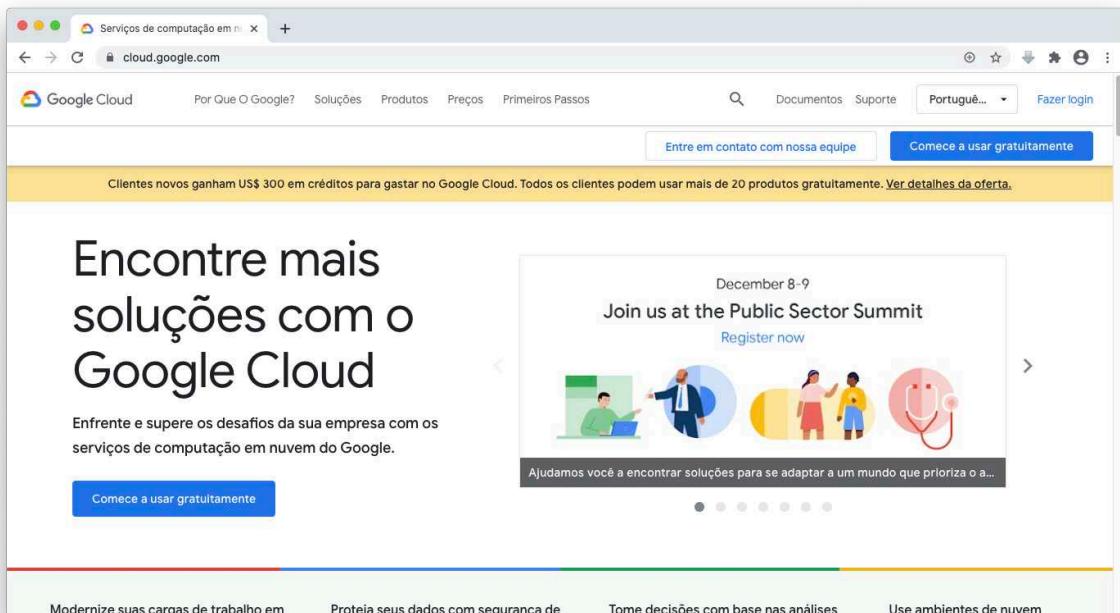


Fig. 7.1: GCP - Conta gratuita

Introdução ao Stackdriver



Fig. 7.2: Stackdriver GCP

O Google Stackdriver é um serviço freemium de gerenciamento de sistemas de computação em nuvem gratuito oferecido pelo Google. Ele fornece dados de desempenho e diagnóstico para usuários de nuvem pública. O Stackdriver é uma solução com várias nuvens, fornecendo suporte para os ambientes de nuvem Google Cloud e AWS.

Fonte: <https://cloud.google.com/>

Gerenciar instâncias na GCP

Criar uma instância na GCP

Vamos criar uma instância na GCP, para isso selecione um projeto.

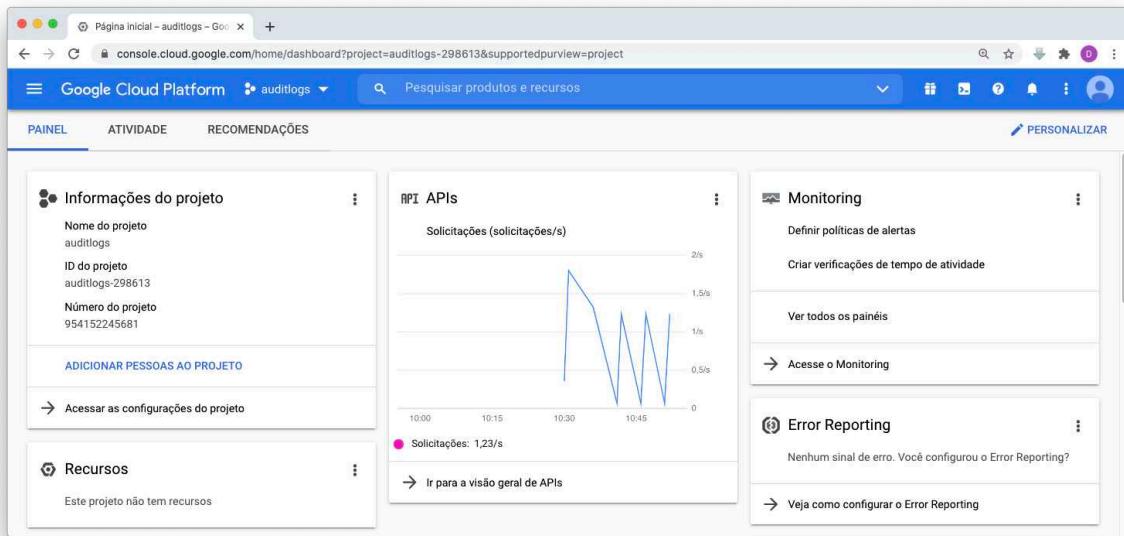


Fig. 7.3: Gerenciar instâncias na GCP - ETAPA 1

Em seguida clique no menu principal que esta no canto superior esquerdo, selecione **Compute Engine > Instâncias de VM**.

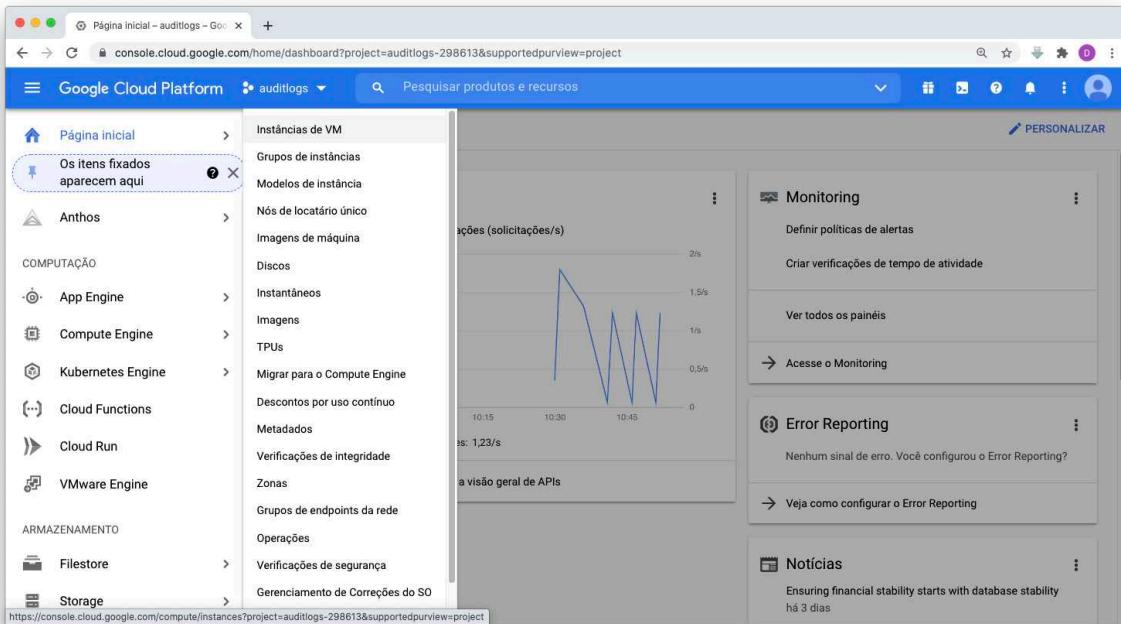


Fig. 7.4: Gerenciar instâncias na GCP - ETAPA 2

O próximo passo é clicar no botão **Criar**.

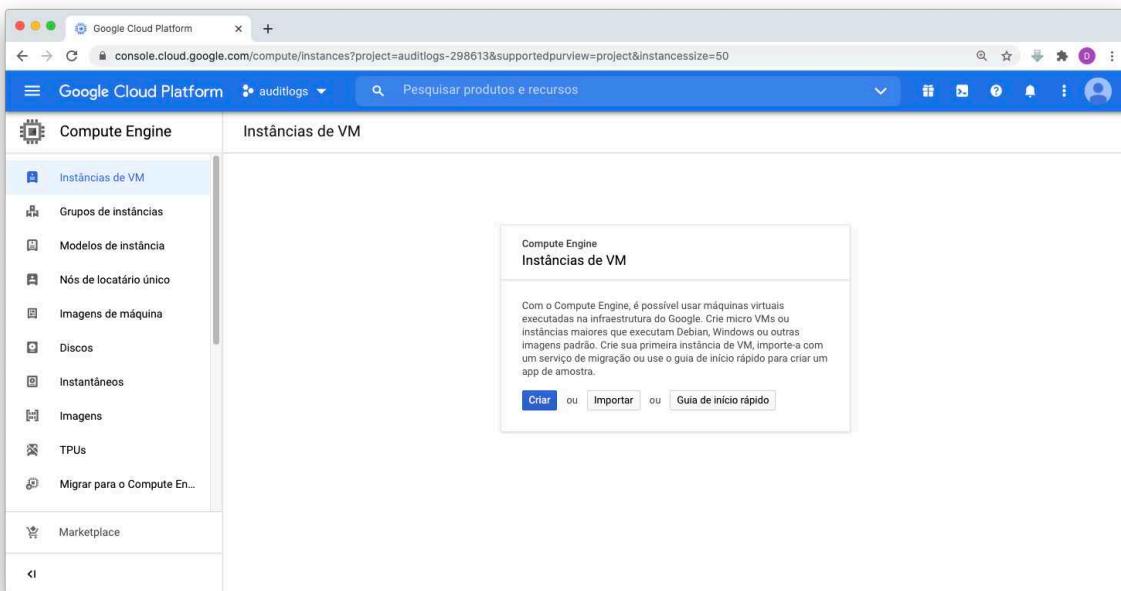


Fig. 7.5: Gerenciar instâncias na GCP - ETAPA 3

Defina um **nome** para a instância, a **região** e o **tipo de máquina**.

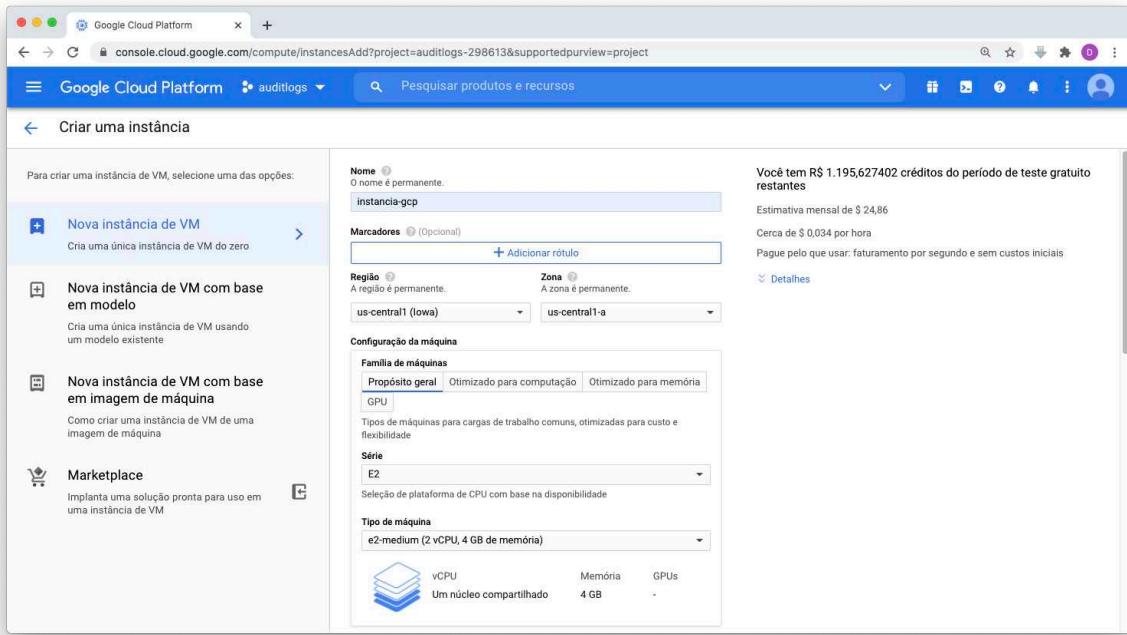


Fig. 7.6: Gerenciar instâncias na GCP - ETAPA 4

Em **Disco de inicialização**, clique no botão **Alterar**.

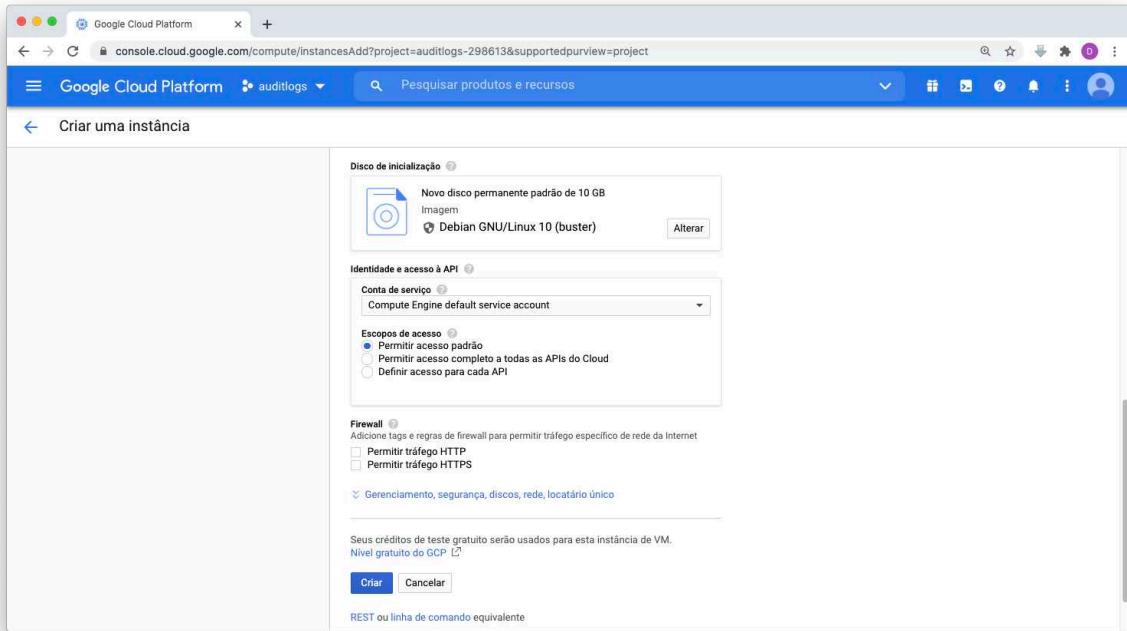


Fig. 7.7: Gerenciar instâncias na GCP - ETAPA 5

Seleciona:

- Sistema Operacional: **Ubuntu**
- Versão: 20.04 LTS
- Tipo de disco de inicialização: **Disco permanente SSD**
- Tamanho (GB): **100**

Clique no botão **Selecionar** para continuar.

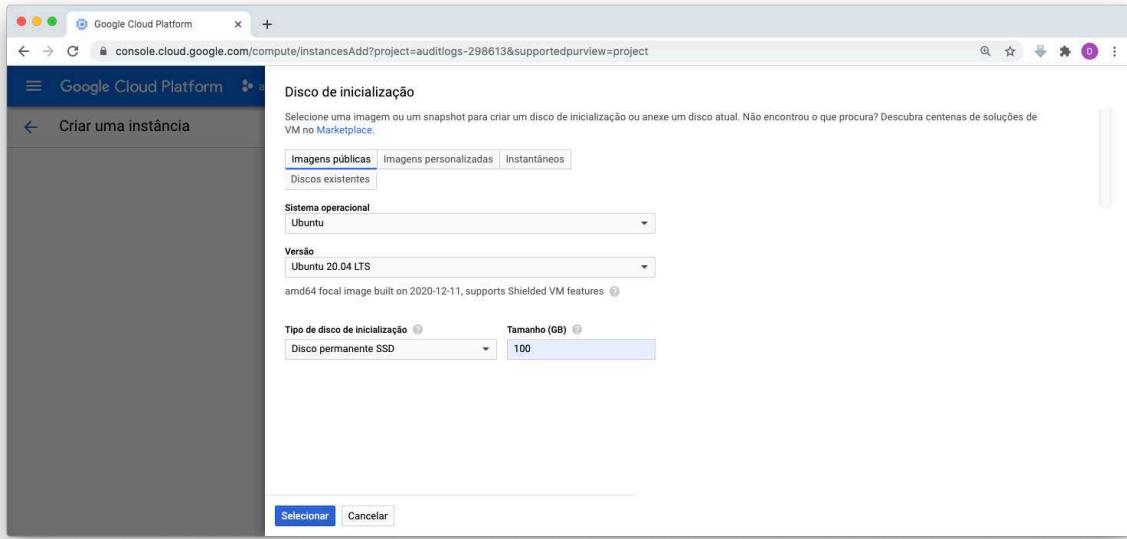


Fig. 7.8: Gerenciar instâncias na GCP - ETAPA 6

Para terminar, clique e, **Permitir tráfego HTTP e HTTPS** e clique no botão Criar.

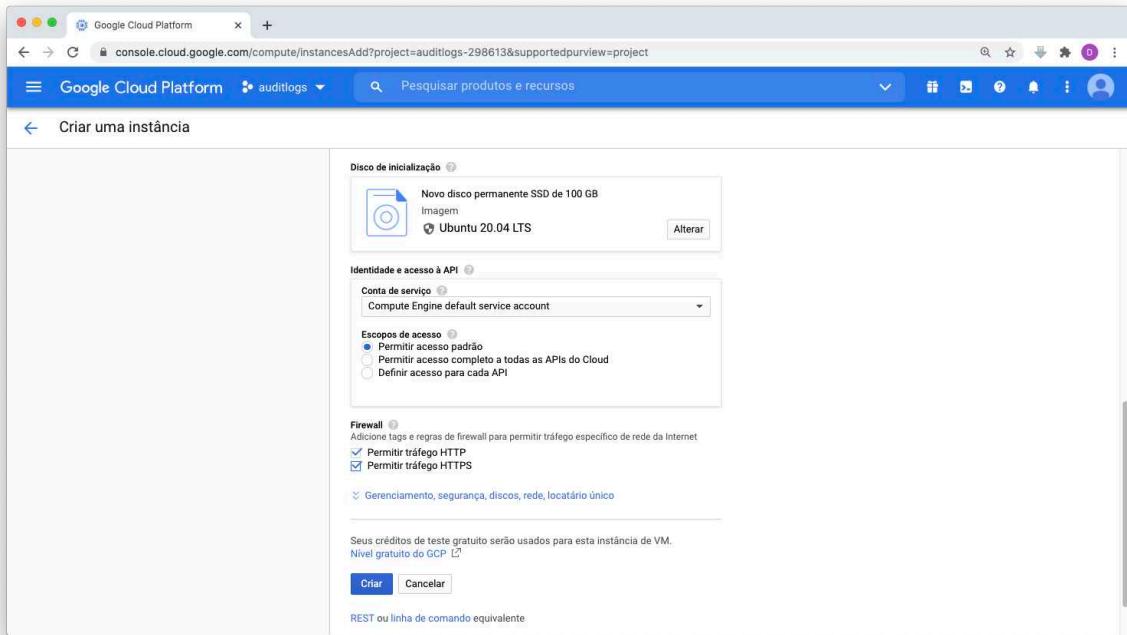


Fig. 7.9: Gerenciar instâncias na GCP - ETAPA 7

Acessar uma instância na GCP

Através da coluna **SSH**, selecione a opção **Abrir na Janela do navegador**.

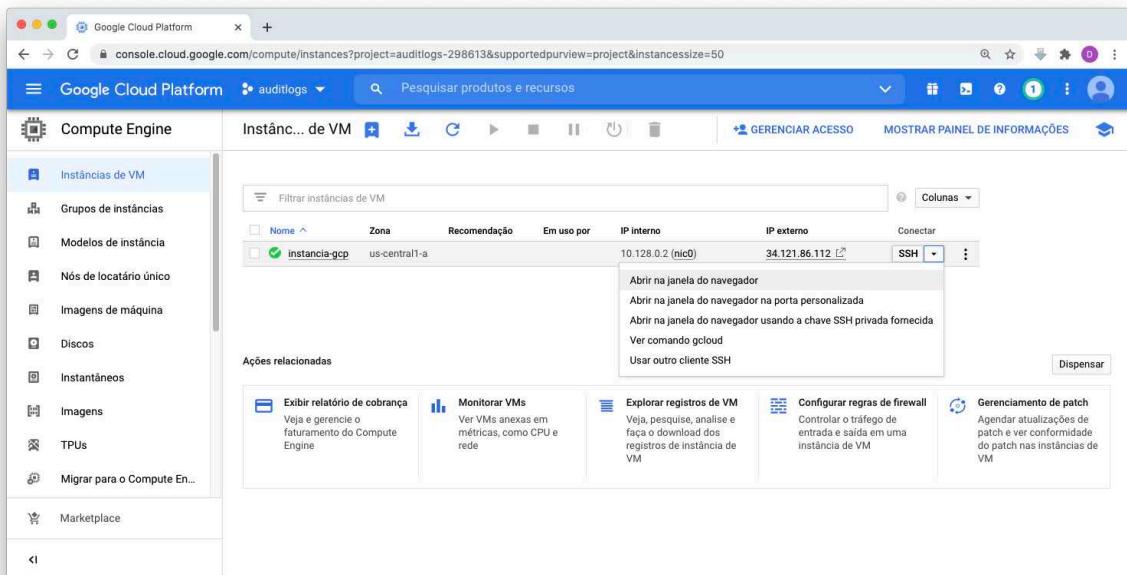


Fig. 7.10: Gerenciar instâncias na GCP

Instalar e configurar o Cloud Logging

Uma conectado na instância da GCP, alterne para a conta do **root**:

```
1 | sudo su -
```

Baixe e execute o script que configura o repositório do Cloud Logging:

```
1 | curl -sS0 https://dl.google.com/cloudagents/add-logging-agent-  
repo.sh  
2 | bash add-logging-agent-repo.sh
```

Atualize a lista de pacotes e instale o pacote **google-fluentd**:

```
1 | apt-get update  
2 | apt-get install -y google-fluentd
```

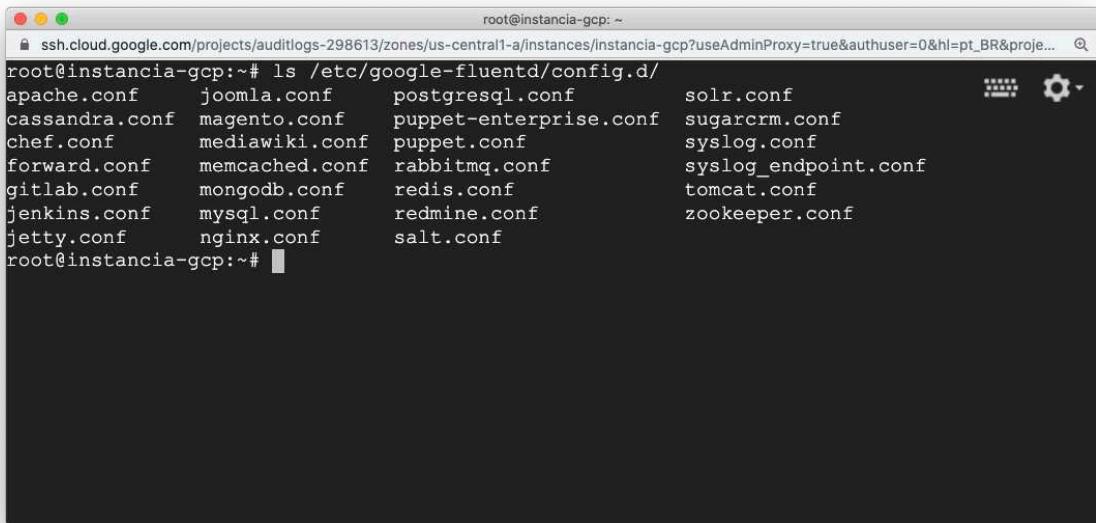
Para terminar instale as configurações modelo do **google-fluentd**:

```
1 | apt-get install -y google-fluentd-catch-all-config
```

Configuração

O **google-fluentd** já possui diversas configurações prontas que estão disponíveis no diretório **/etc/google-fluentd/config.d/**:

```
1 | ls /etc/google-fluentd/config.d/
```

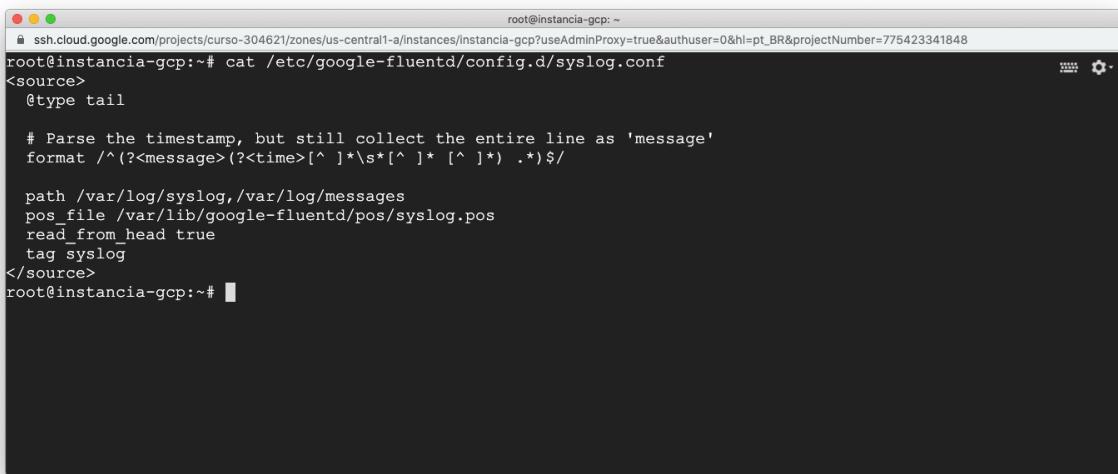


```
root@instancia-gcp:~# ls /etc/google-fluentd/config.d/
apache.conf      joomla.conf      postgresql.conf      solr.conf
cassandra.conf   magento.conf     puppet-enterprise.conf sugarcrm.conf
chef.conf        mediawiki.conf   puppet.conf          syslog.conf
forward.conf     memcached.conf   rabbitmq.conf       syslog_endpoint.conf
gitlab.conf      mongodb.conf    redis.conf         tomcat.conf
jenkins.conf     mysql.conf      redmine.conf      zookeeper.conf
jetty.conf       nginx.conf     salt.conf
root@instancia-gcp:~#
```

Fig. 7.11: Gerenciar instâncias na GCP

Para acompanhar os logs do Rsyslog na GCP, é preciso que o arquivo **syslog.conf** exista no diretório **/etc/google-fluentd/config.d/**.

```
1 | cat /etc/google-fluentd/config.d/syslog.conf
```



```
root@instancia-gcp:~#
ssh.cloud.google.com/projects/cr...?useAdminProxy=true&authuser=0&hl=pt_BR&projectNumber=775423341848
root@instancia-gcp:~# cat /etc/google-fluentd/config.d/syslog.conf
<source>
  @type tail
  # Parse the timestamp, but still collect the entire line as 'message'
  format /^(?<message>(?<time>[^ ]*\s*[^ ]* [^ ]*) .*)$/
  path /var/log/syslog,/var/log/messages
  pos_file /var/lib/google-fluentd/pos/syslog.pos
  read_from_head true
  tag syslog
</source>
root@instancia-gcp:~#
```

Fig. 7.12: Gerenciar instâncias na GCP

Inicie e ative na inicialização do sistema o serviço do **Cloud Logging**.

```
1 | systemctl start google-fluentd
2 | systemctl enable google-fluentd
```

Visualizar logs da instância da GCP

Clique no menu principal que esta no canto superior esquerdo, selecione **Registros > Explorador de registros**.

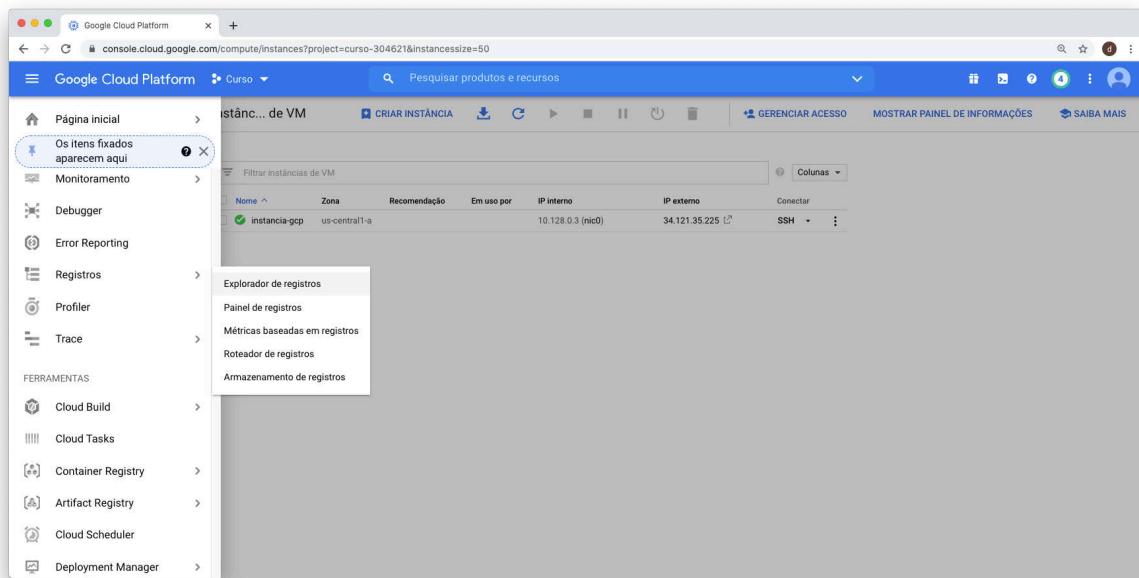


Fig. 7.13: Acessar Explorador de registros

Em seguida selecione **Roteador de registro > CRIAR COLETOR**.

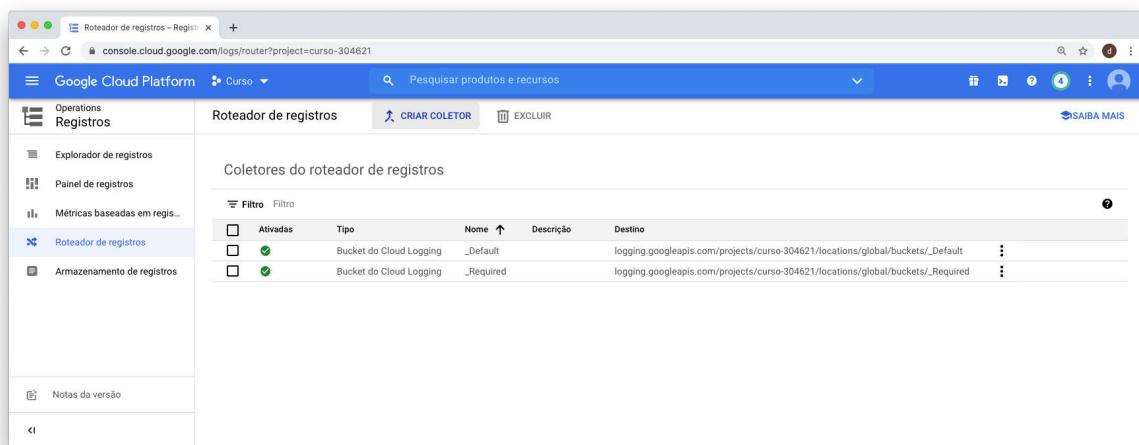


Fig. 7.14: Criar coletor de roteamento de registros - ETAPA 1

Preencha o nome do coletor, em nosso exemplo vamos usar **bucket-audit**. Em seguida clique no botão **NEXT**.

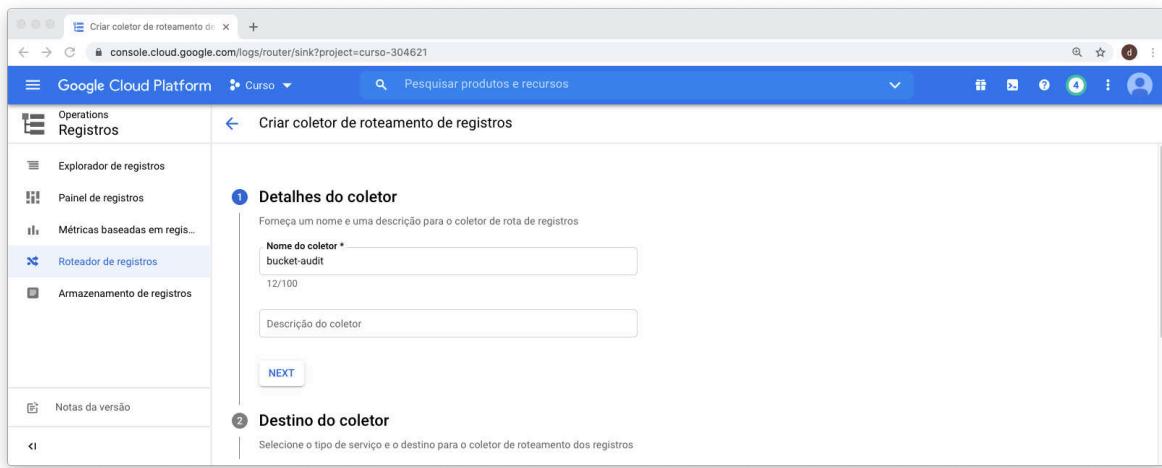


Fig. 7.15: Criar coletor de roteamento de registros - ETAPA 2

Em **Destino do coletor** selecione **Bucket do Cloud Logging**.



Fig. 7.16: Criar coletor de roteamento de registros - ETAPA 3

Em **Selecionar um bucket de registros** selecione **Criar novo bucket de registros**.

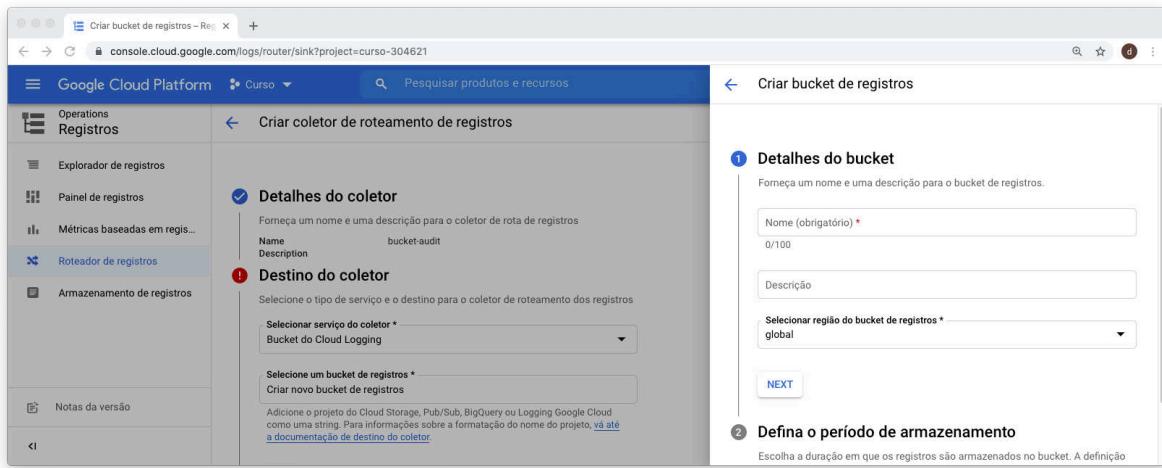


Fig. 7.17: Criar coletor de roteamento de registros - ETAPA 4

Digite em Nome **bucket-logs** e clique no botão **CRIAR BUCKET**.

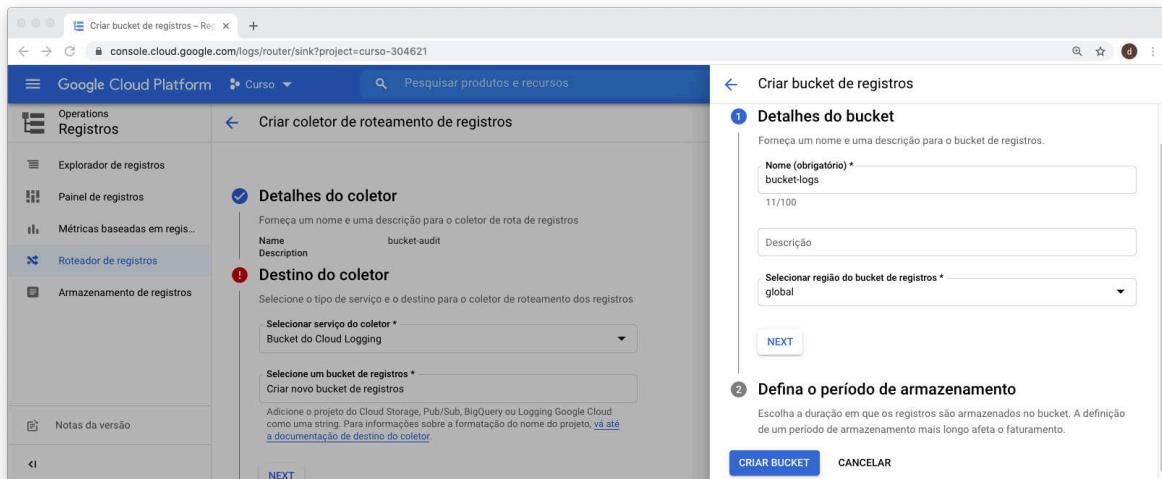


Fig. 7.18: Criar coletor de roteamento de registros - ETAPA 5

Com o novo Bucket criado, clique no botão **NEXT**.

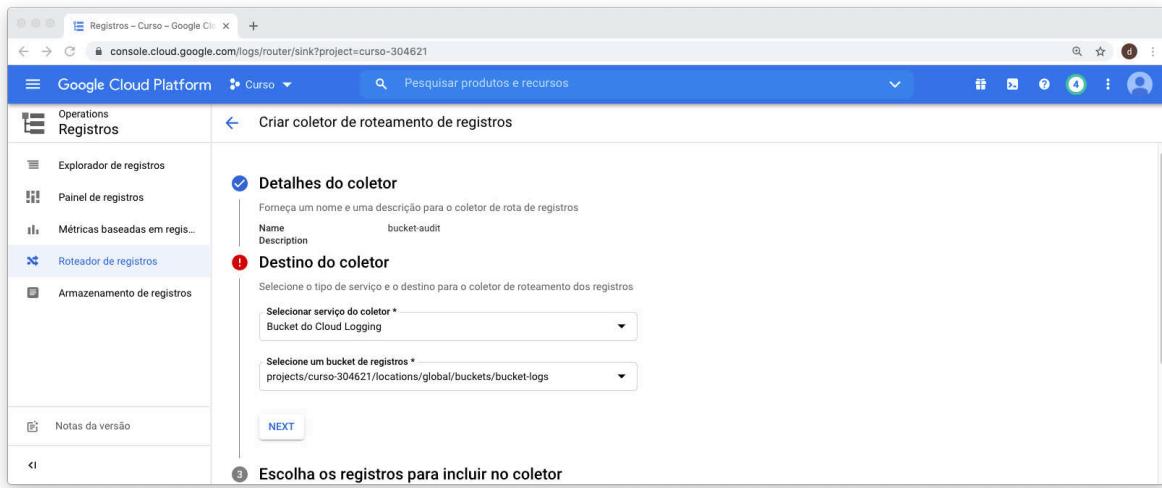


Fig. 7.19: Criar coletor de roteamento de registros - ETAPA 6

Para terminar clique no botão **CRIAR COLETOR**.

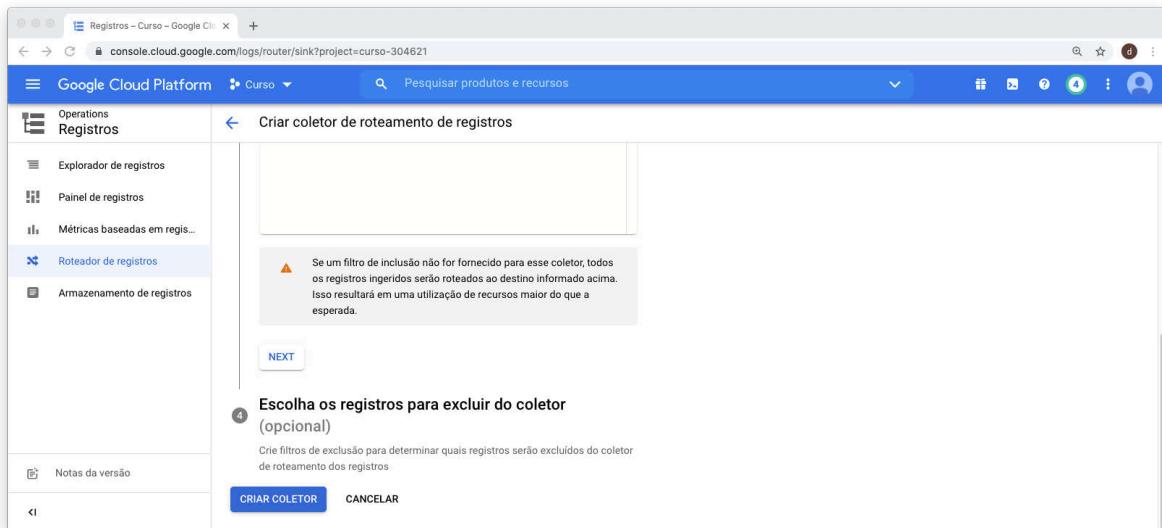


Fig. 7.20: Criar coletor de roteamento de registros - ETAPA 7

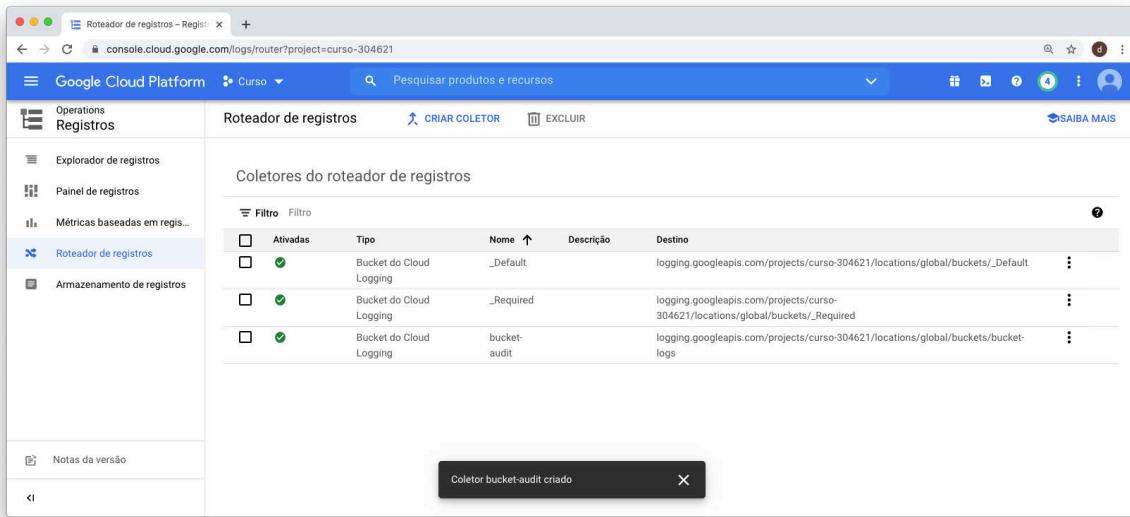


Fig. 7.21: Criar coletor de roteamento de registros - ETAPA 8

Retorne ao terminal da instância e reinicie o serviço do **Cloud Logging**.

```
1 | systemctl restart google-fluentd
```

Retorne ao painel de registros e clique em Explorador de Registros.

Selecione **instância da VM > Nome da instância** (nome exemplo: *instancia-gcp*) > **Todos os registros** > Qualquer nível de registro.

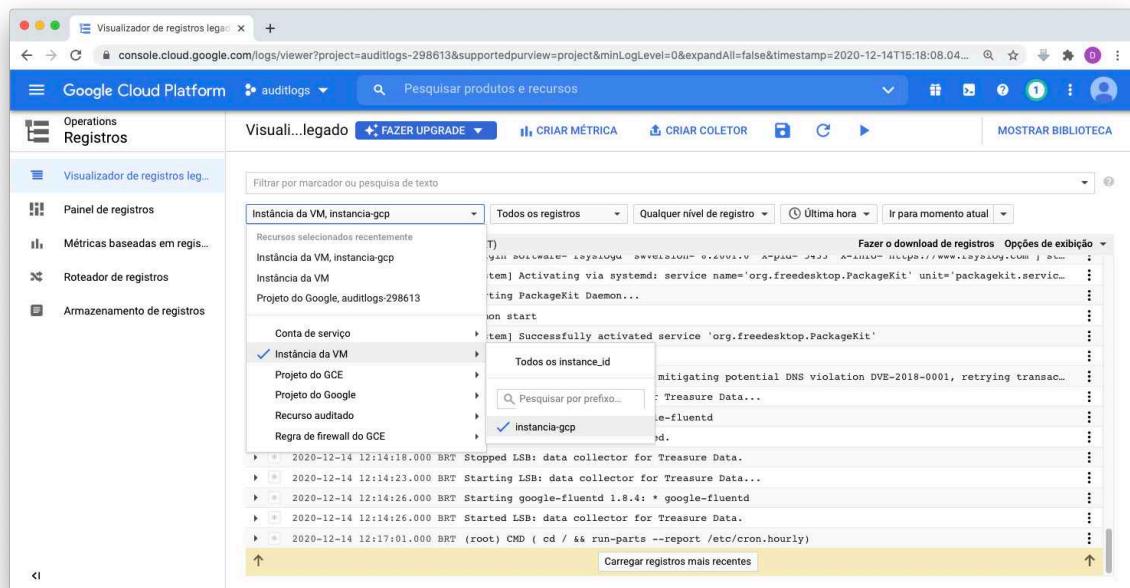


Fig. 7.22: Acessar logs da instância no Stackdriver - ETAPA 2

Como resultado final é possível visualizar os logs do arquivo **/var/log/syslog**.

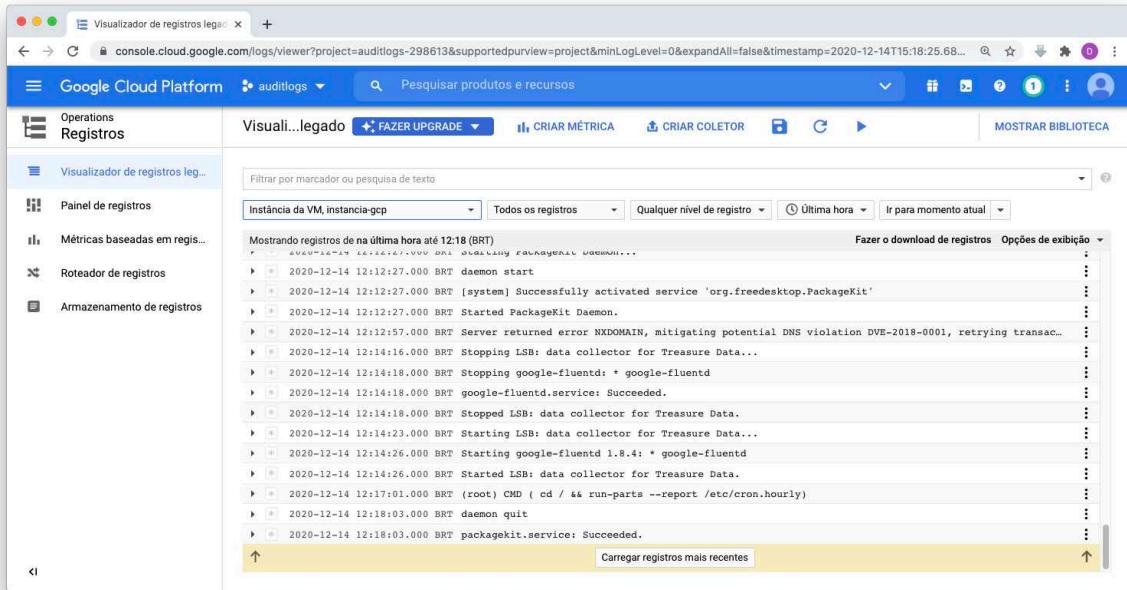


Fig. 7.23: Acessar logs da instância no Stackdriver - ETAPA 3

Alterne para a janela de acesso se sua instância, e através do comando **logger** crie um log de exemplo:

```
1 | logger 'Curso Auditoria de Logs 4Linux'
```

Como resultado é possível visualizar o novo log no arquivo **/var/log/syslog**.

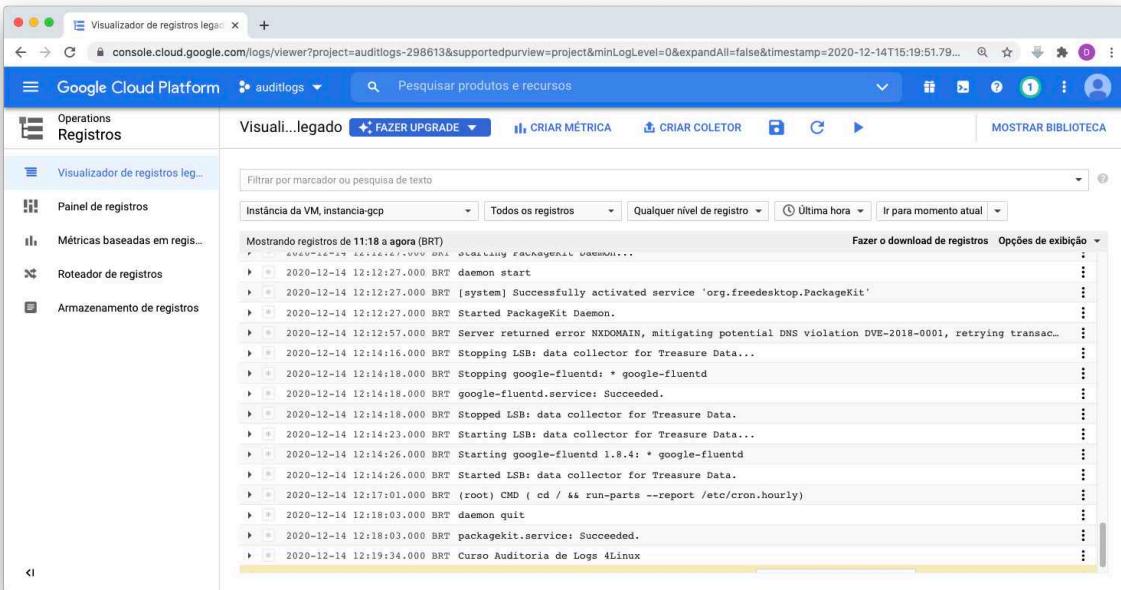


Fig. 7.24: Acessar logs da instância no Stackdriver - ETAPA 4

Configurar e visualizar logs de um servidor web

Para começar instale o pacote **nginx**:

```
1 | apt install nginx -y
```

E inicie o serviço do **nginx**.

```
1 | systemctl start nginx
```

Através do **Compute Engine > Instâncias de VM**, verifique qual é IP público da VM:

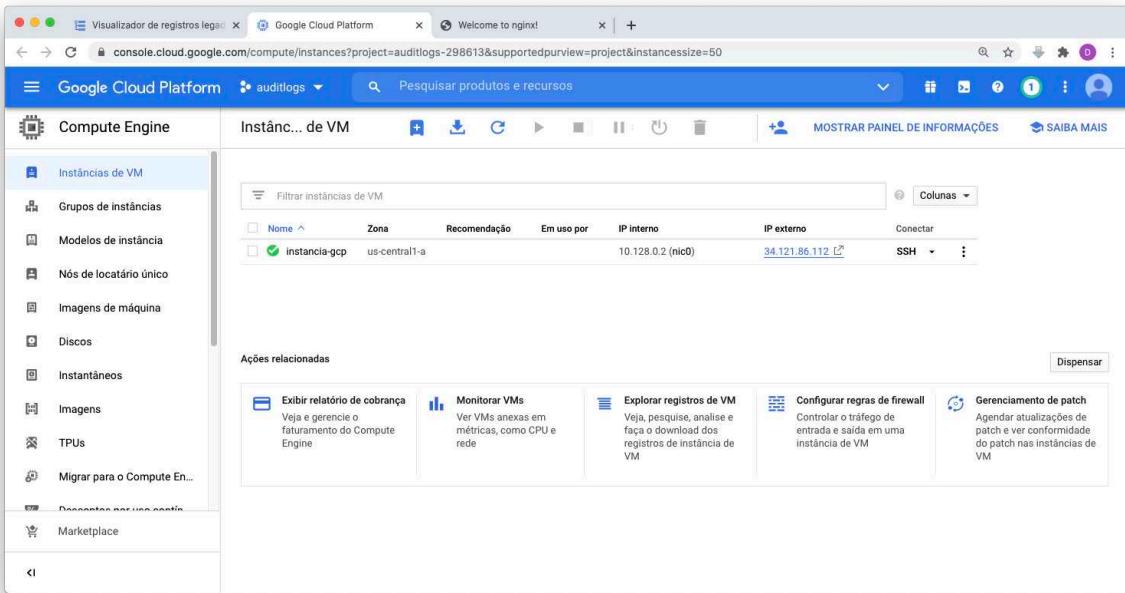


Fig. 7.25: Visualizar logs do servidor Web - ETAPA 1

Acesse a página do Nginx através do IP público da VM, com o intuito de gerar logs de acesso.

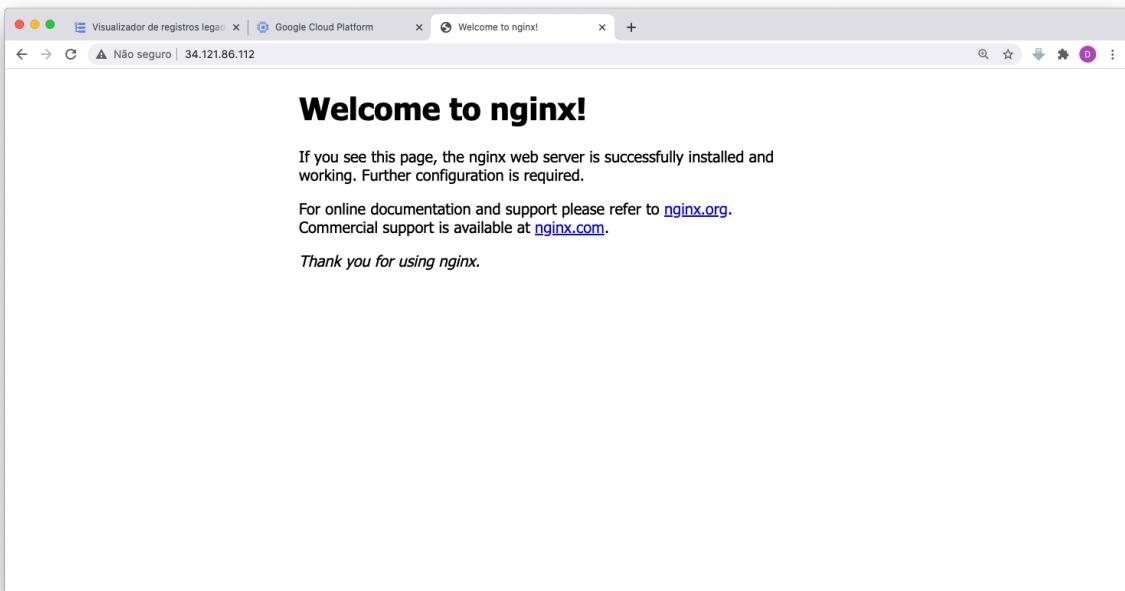


Fig. 7.26: Visualizar logs do servidor Web - ETAPA 2

Para acompanhar os logs do **Nginx** na GCP, é preciso que o arquivo **nginx.conf** exista no diretório **/etc/google-fluentd/config.d/**.

```
1 | cat /etc/google-fluentd/config.d/nginx.conf
```

```
root@instancia-gcp:~# cat /etc/google-fluentd/config.d/nginx.conf
<source>
  @type tail
  format none
  path /var/log/nginx/access.log
  pos_file /var/lib/google-fluentd/pos/nginx-access.pos
  read_from_head true
  tag nginx-access
</source>

<source>
  @type tail
  format none
  path /var/log/nginx/error.log
  pos_file /var/lib/google-fluentd/pos/nginx-error.pos
  read_from_head true
  tag nginx-error
</source>
root@instancia-gcp:~#
```

Fig. 7.27: Gerenciar instâncias na GCP

Retorne a janela de registros e selecione na caixa Todos os registros, a opção **nginx-access**.

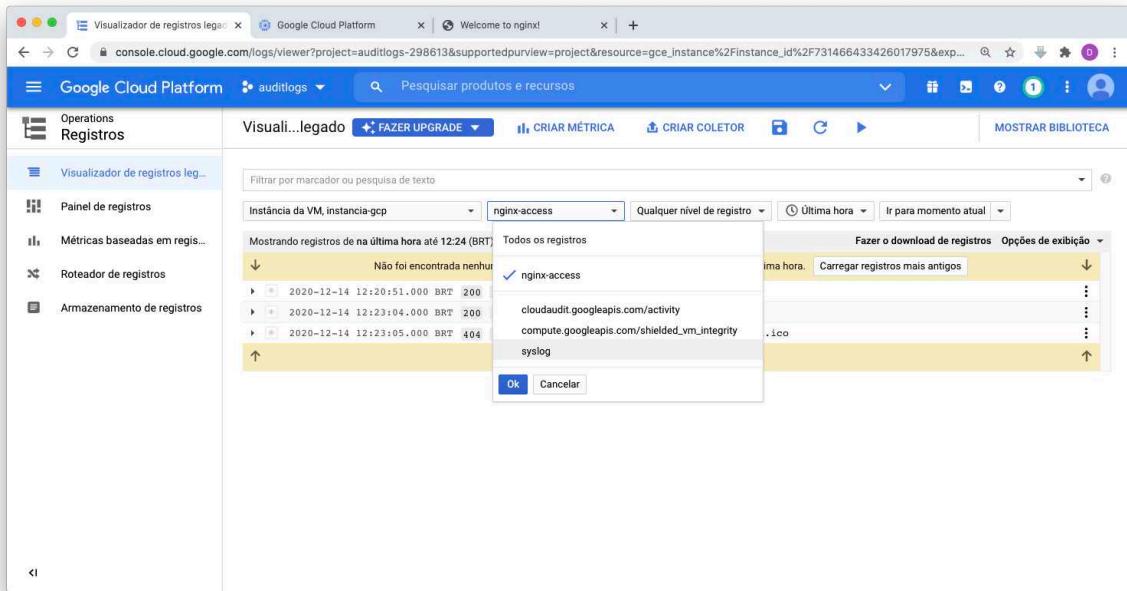


Fig. 7.28: Visualizar logs do servidor Web - ETAPA 3

Como resultado final é possível visualizar os logs de acesso de nosso servidor **Web Nginx**.

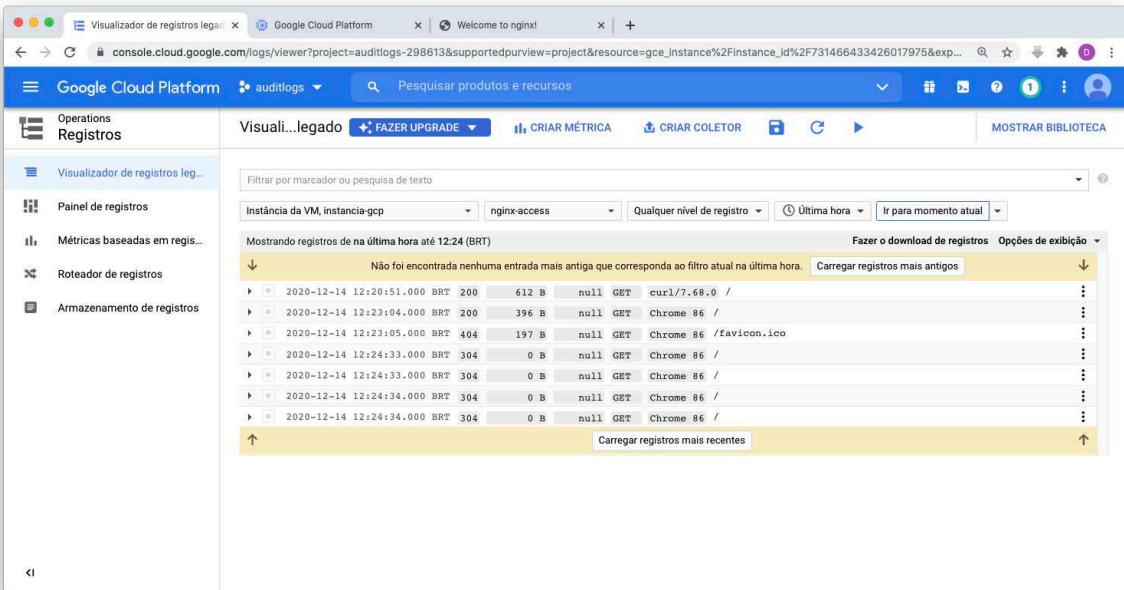


Fig. 7.29: Visualizar logs do servidor Web - ETAPA 4

Visualizar logs de container no Docker

Para a instalação do Docker na instância é preciso seguir os seguintes passos:

1. Atualizar lista de pacotes repositórios do sistema
2. Instalar as dependências para o Docker
3. Baixar e configurar a chave pública do repositório.
4. Configurar o repositório estável do Docker
5. Atualizar novamente a lista de pacotes dos repositórios configurados.
6. Instalar o Docker

Conforme a sequência de comandos a seguir:

```

1 apt-get update
2 apt-get install apt-transport-https ca-certificates curl gnupg-
   agent software-properties-common -y
3 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-
   key add -
4 add-apt-repository "deb [arch=amd64] https://download.docker.com
   /linux/ubuntu $(lsb_release -cs) stable"
5 apt-get update
6 apt-get install docker-ce docker-ce-cli containerd.io -y

```

Verifique qual é a versão instalada do Docker:

```
1 | docker version
```

Antes de executar um container do **Apache**, pare o serviço do Nginx que está utilizando a **porta 80**:

```
1 | systemctl stop nginx
```

Em seguida execute um container do **Apache** informando o driver de log **gcplogs**:

```
1 | docker container run -d --name httpd --log-driver=gcplogs --log-opt gcp-meta-name=`hostname` -p 80:80 httpd
```

Acesse a página do Apache através do IP público da VM, com o intuito de gerar logs de acesso do container.

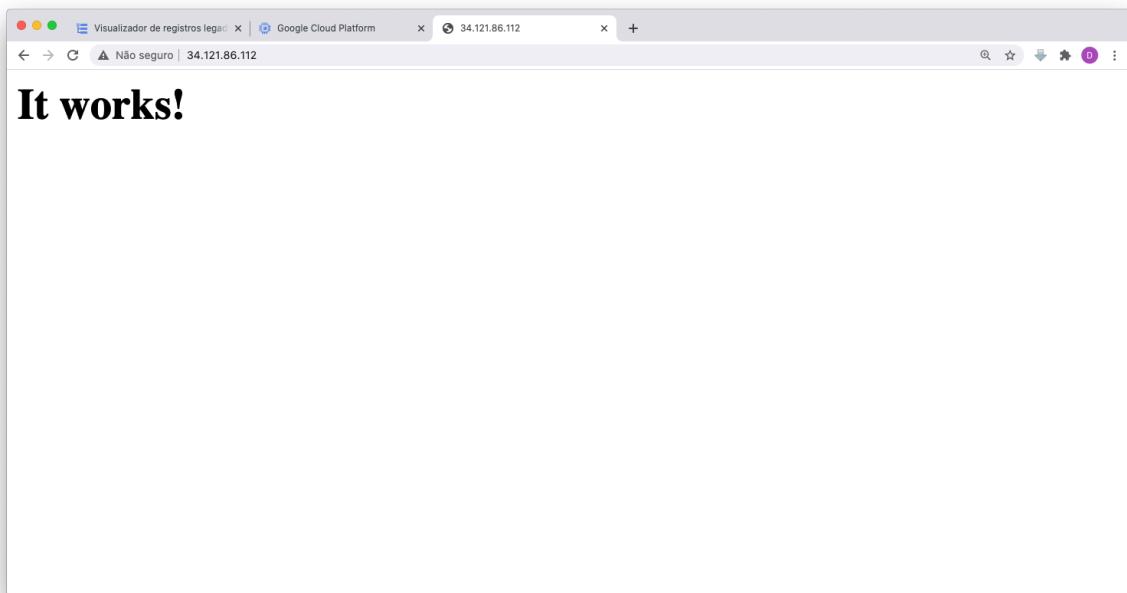


Fig. 7.30: Visualizar logs de Containers no Docker - ETAPA 1

Retorne a janela de registros e selecione na caixa *Todos os registros*, a opção **gcplogs-docker-driver**.

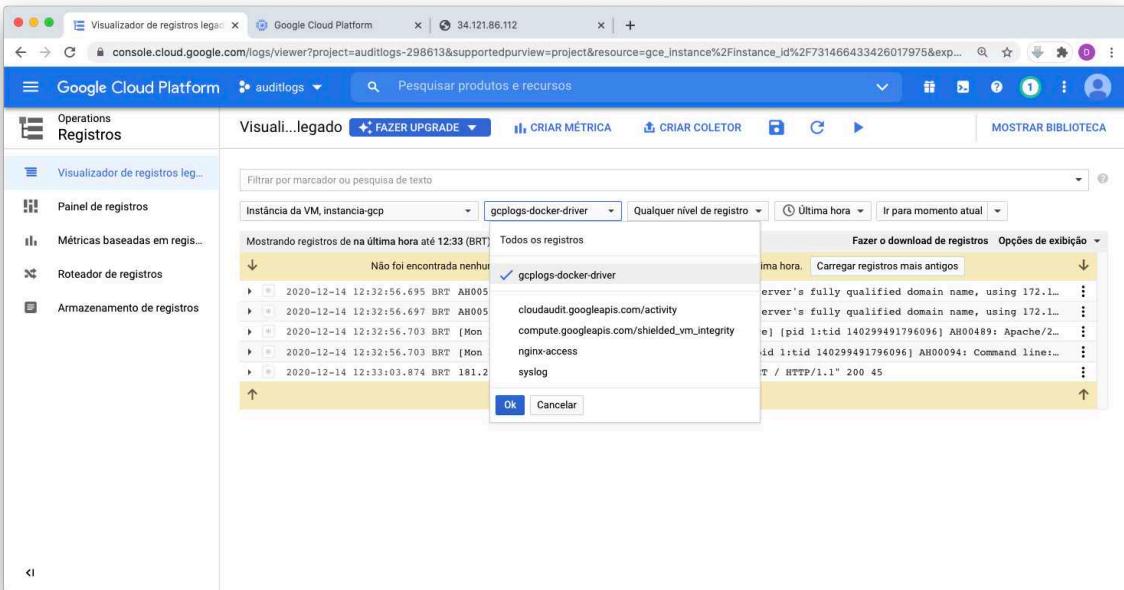


Fig. 7.31: Visualizar logs de Containers no Docker - ETAPA 2

Como resultado final é possível visualizar os logs de acesso de nosso servidor **Apache**, que está sendo executado dentro de um container no Docker:

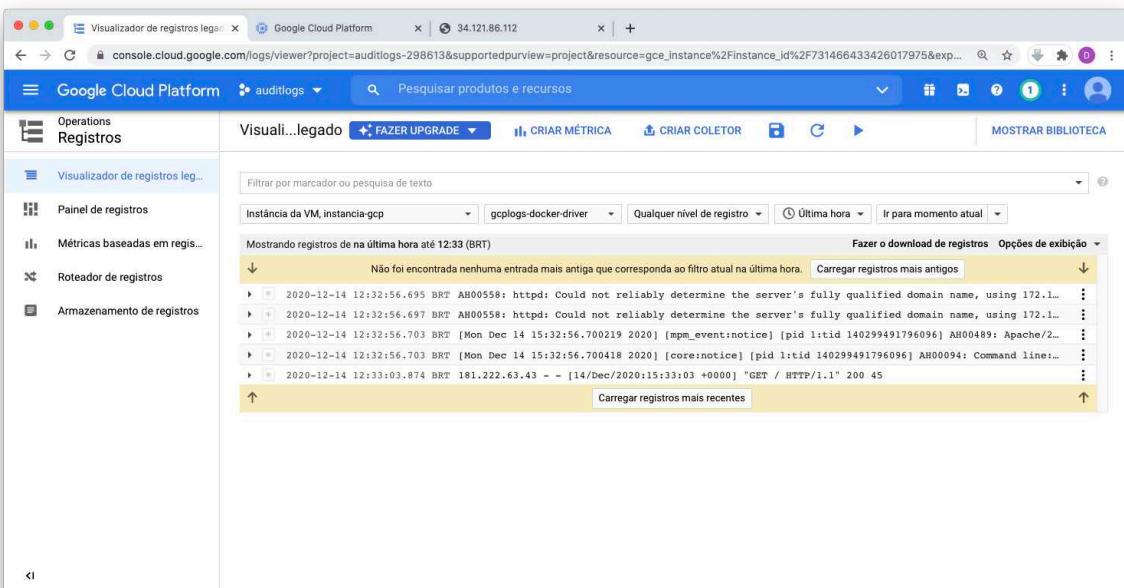


Fig. 7.32: Visualizar logs de Containers no Docker - ETAPA 3