ROCKET.CHAT

Technical Assessment Challenge

Matheus Paiva_April2021

Summary

Deployment Environment

- Installed a snap of Rocket.chat on Ubuntu 16.04 LTS as per documentation available on https://docs.rocket.chat

- Created an Ubuntu VM running on Microsoft Azure platform following the recommended Hardware resources. (Public IP: 23.97.102.57)

  Rocket.chat server credentials:

  URL : https://rocketchatmatheuspaiva.brazilsouth.cloudapp.azure.com
  User: mmcp108@gmail.com
  Pass: RC$_17042021

Rocket.chat Installation steps

1- Connected to the Ubuntu VM via terminal (on this case I used putty) and entered the following command line:

```
sudo snap install rocketchat-server
```

2- On my computer, browse to http://localhost:3000 to setup Rocket.chat.

Note: On this step I noticed that there was no access to the page, then I enabled the inbound port role for port 3000 on Azure Networking settings, then I was able to browse to the URL and setup Rocket.chat.

Configure Auto SSL (Caddy service)

Note: As this Ubuntu VM was created on MS Azure platform it provided me a domain to use.

1- Entered the following command line via terminal:

```
sudo snap set rocketchat-server caddy-url=https:// rocketchatmatheuspaiva.brazilsouth.cloudapp.azure.com
sudo snap set rocketchat-server caddy=enable
sudo snap set rocketchat-server https=enable
sudo snap run rocketchat-server.initcaddy
```

2- Restarted Rocket.chat and Caddy services:

```
sudo systemctl restart snap.rocketchat-server.rocketchat-server.service
sudo systemctl restart snap.rocketchat-server.rocketchat-caddy.service
```

In order to proof that configuration was applied, I browsed again to the URL https:\\ rocketchatmatheuspaiva.brazilsouth.cloudapp.azure.com and we can see that the padlock icon is being shown in the browser and if we click on that we can see that there is a valid SSL certificate and the server can use HTTPS protocol.

Then, disabled the port 3000 in MS Azure Networking settings.

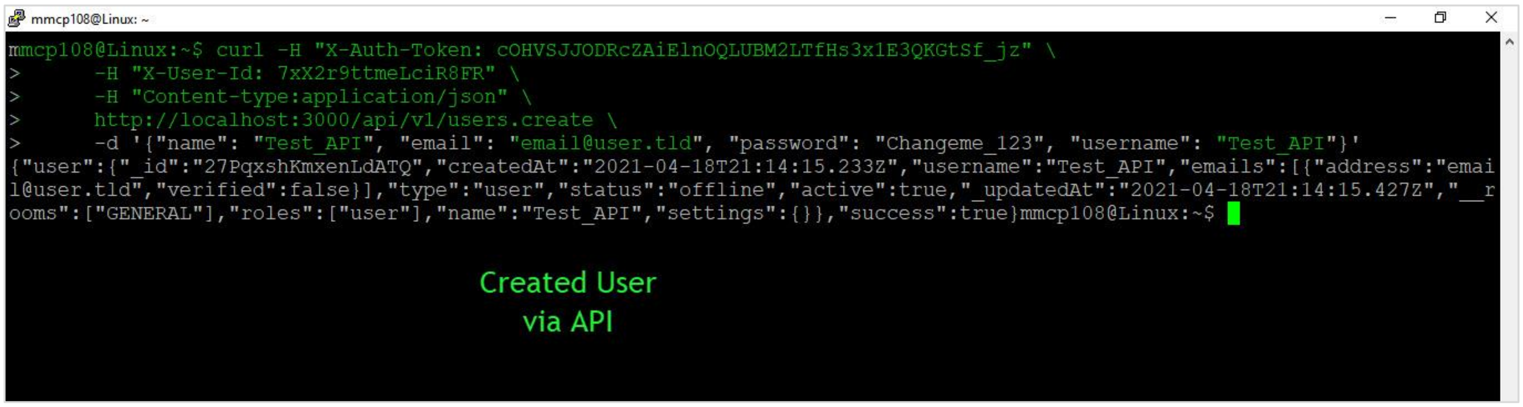
Rocket.chat APIs Test Results

NOTE: In order to perform the queries/ updates via rest API, firstly you must have to generate a personal access token in your profile page: *Profile -> My Account -> Security -> Personal Access Tokens*

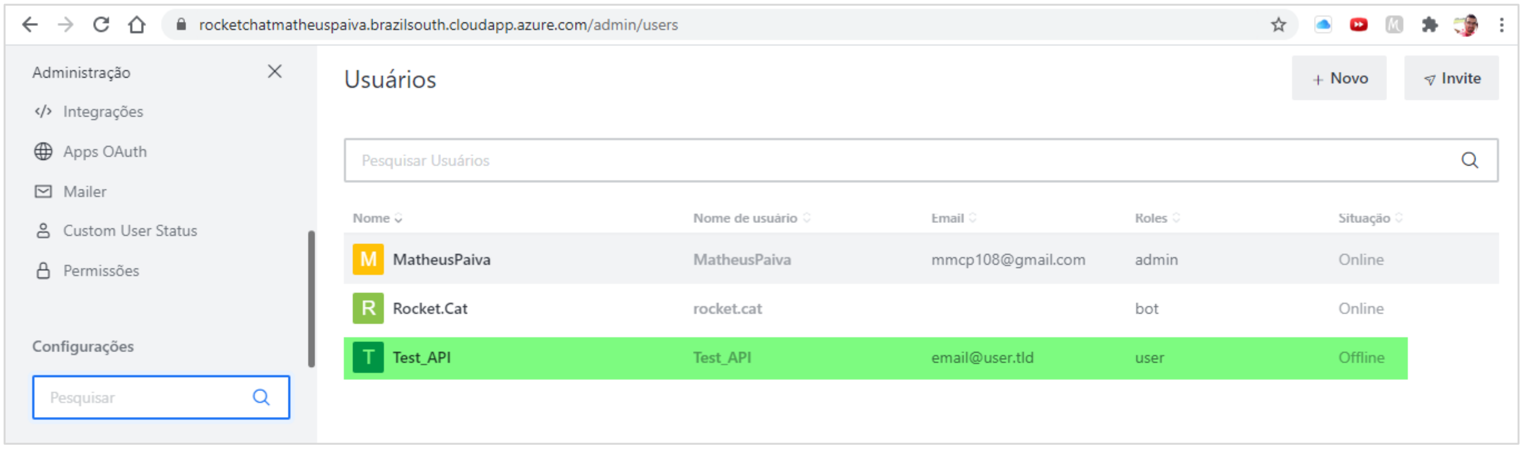
Token: cOHVSJJODRcZAiElnOQLUBM2LTfHs3x1E3QKGtSf_jz

User ID: 7xX2r9ttmeLciR8FR

Create a new user via an API endpoint



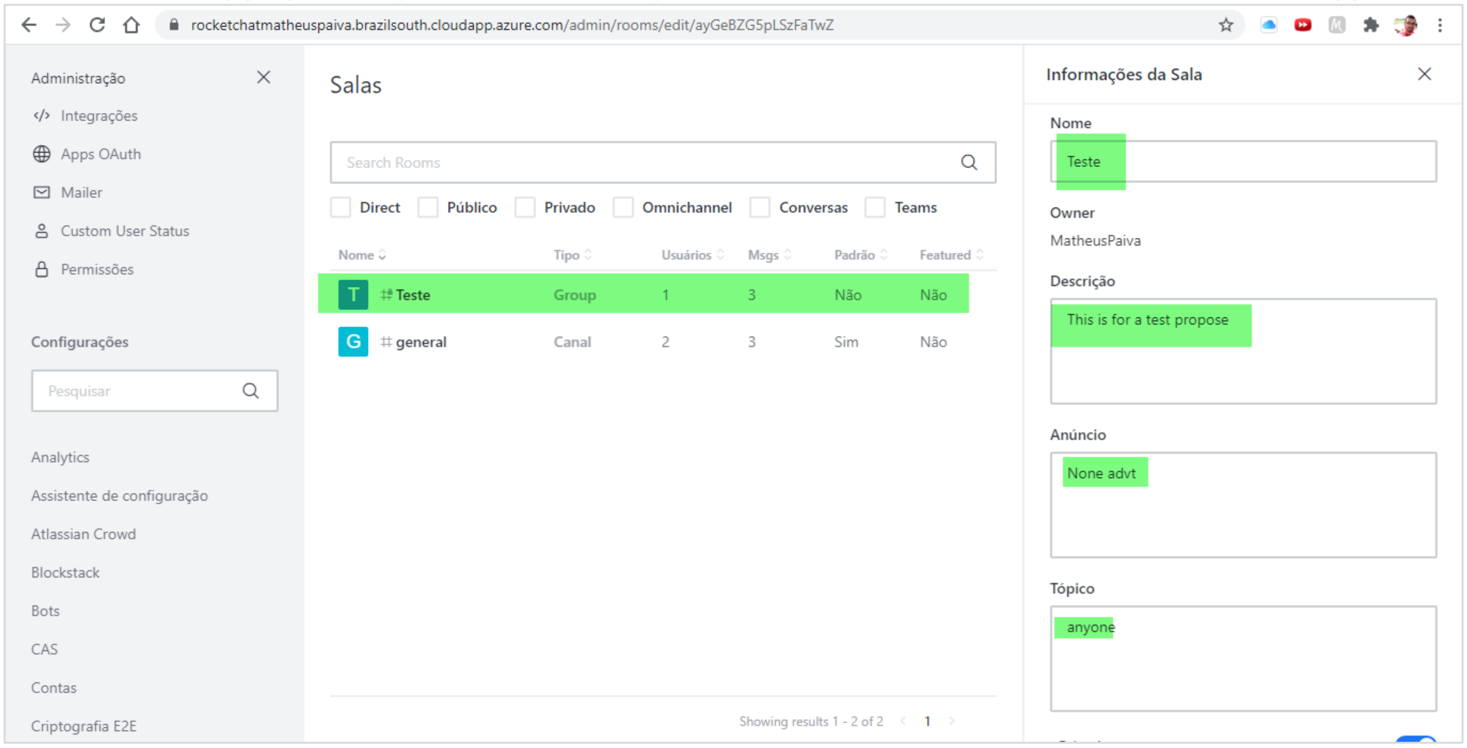
```
mmcp108@Linux:~$ curl -H "X-Auth-Token: cOHVSJJODRcZAiElnOQLUBM2LTfHs3x1E3QKGtSf_jz" \
>       -H "X-User-Id: 7xX2r9ttmeLciR8FR" \
>       -H "Content-type:application/json" \
>       http://localhost:3000/api/v1/users.create \
>       -d '{"name": "Test_API", "email": "email@user.tld", "password": "Changeme_123", "username": "Test_API"}'
{"user":{"_id":"27PqxshKmxenLdATQ","createdAt":"2021-04-18T21:14:15.233Z","username":"Test_API","emails":[{"address":"emai
l@user.tld","verified":false}],"type":"user","status":"offline","active":true,"_updatedAt":"2021-04-18T21:14:15.427Z","__r
ooms":["GENERAL"],"roles":["user"],"name":"Test_API","settings":{}},"success":true}mmcp108@Linux:~$
```



Nome	Nome de usuário	Email	Roles	Situação
M MatheusPaiva	MatheusPaiva	mmcp108@gmail.com	admin	Online
R Rocket.Cat	rocket.cat		bot	Online
T Test_API	Test_API	email@user.tld	user	Offline

Get the room information via an API endpoint

For testing propose only, I created manually a new Room named “Teste” via Rocket.chat web application:



Nome	Tipo	Usuários	Msgs	Padrão	Featured
T # Teste	Group	1	3	Não	Não
G # general	Canal	2	3	Sim	Não

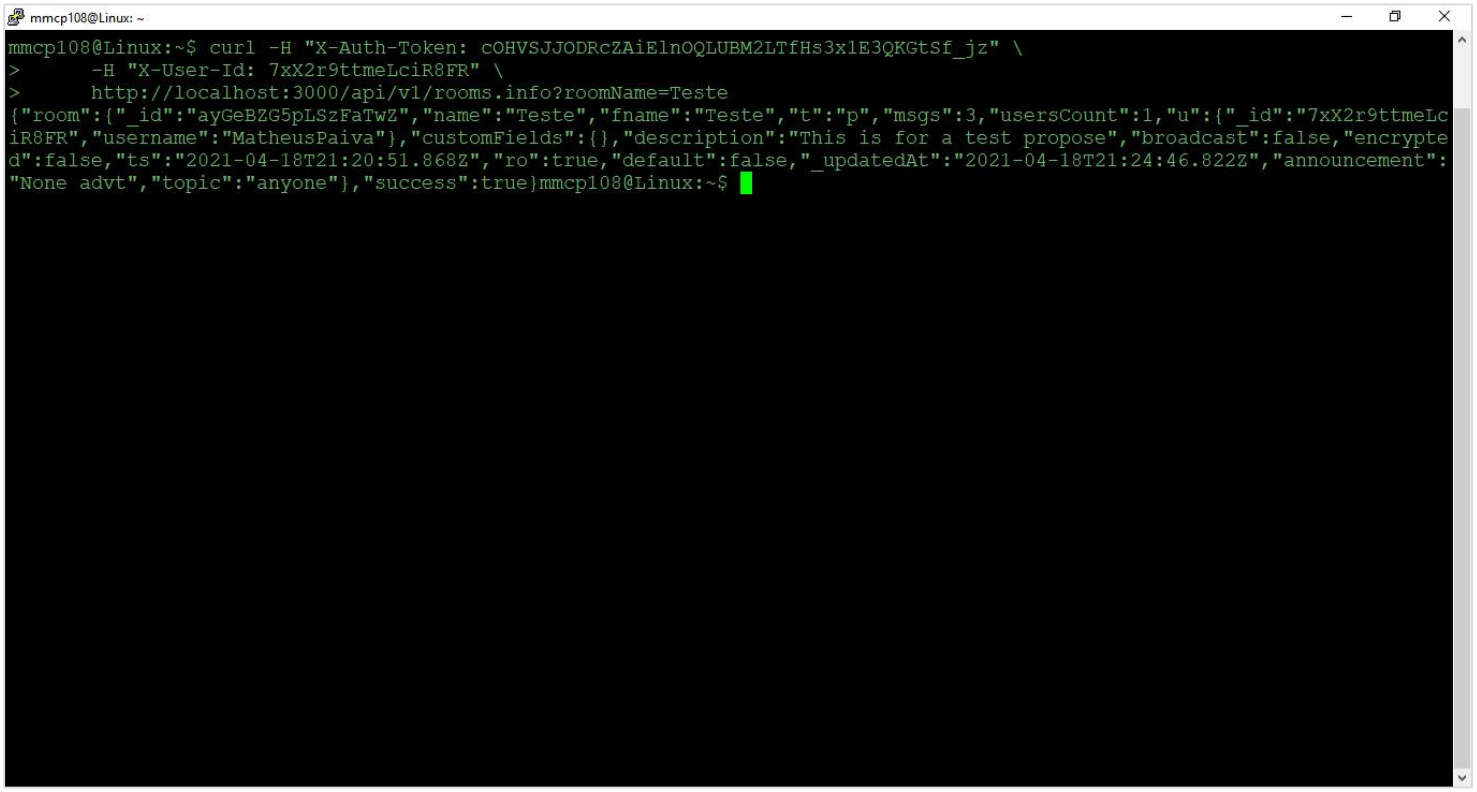
Nome: Teste

Owner: MatheusPaiva

Descrição: This is for a test propose

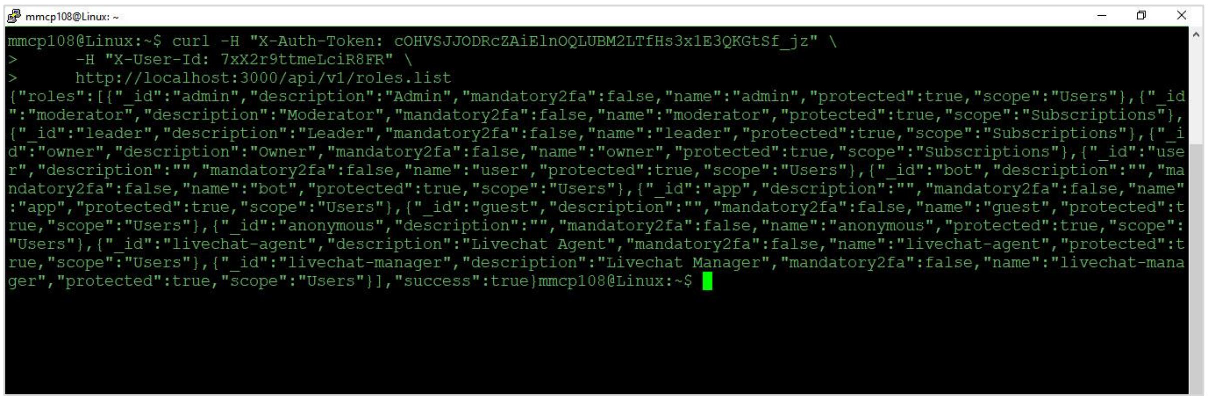
Anúncio: None advt

Tópico: anyone



```
mmcp108@Linux:~$ curl -H "X-Auth-Token: cOHVSJJODRcZAiElnOQLUBM2LTfHs3x1E3QKGtSf_jz" \
>      -H "X-User-Id: 7xX2r9ttmeLciR8FR" \
>      http://localhost:3000/api/v1/rooms.info?roomName=Teste
{"room":{"_id":"ayGeBZG5pLSzFaTwZ","name":"Teste","fname":"Teste","t":"p","msgs":3,"usersCount":1,"u":{"_id":"7xX2r9ttmeLc
iR8FR","username":"MatheusPaiva"},"customFields":{},"description":"This is for a test propose","broadcast":false,"encrypte
d":false,"ts":"2021-04-18T21:20:51.868Z","ro":true,"default":false,"_updatedAt":"2021-04-18T21:24:46.822Z","announcement":
"None advt","topic":"anyone"},"success":true}mmcp108@Linux:~$
```

Get a list of all user roles in the system via an API endpoint



```
mmcp108@Linux: ~
mmcp108@Linux:~$ curl -H "X-Auth-Token: cOHVSJJODRcZAiElnOQLUBM2LTfHs3x1E3QKGtSf_jz" \
>      -H "X-User-Id: 7xX2r9ttmeLciR8FR" \
>      http://localhost:3000/api/v1/roles.list
{"roles":[{"_id":"admin","description":"Admin","mandatory2fa":false,"name":"admin","protected":true,"scope":"Users"},{"_id
":"moderator","description":"Moderator","mandatory2fa":false,"name":"moderator","protected":true,"scope":"Subscriptions"},
{"_id":"leader","description":"Leader","mandatory2fa":false,"name":"leader","protected":true,"scope":"Subscriptions"},{"_i
d":"owner","description":"Owner","mandatory2fa":false,"name":"owner","protected":true,"scope":"Subscriptions"},{"_id":"use
r","description":"","mandatory2fa":false,"name":"user","protected":true,"scope":"Users"},{"_id":"bot","description":"","ma
ndatory2fa":false,"name":"bot","protected":true,"scope":"Users"},{"_id":"app","description":"","mandatory2fa":false,"name"
:"app","protected":true,"scope":"Users"},{"_id":"guest","description":"","mandatory2fa":false,"name":"guest","protected":t
rue,"scope":"Users"},{"_id":"anonymous","description":"","mandatory2fa":false,"name":"anonymous","protected":true,"scope":
"Users"},{"_id":"livechat-agent","description":"Livechat Agent","mandatory2fa":false,"name":"livechat-agent","protected":t
rue,"scope":"Users"},{"_id":"livechat-manager","description":"Livechat Manager","mandatory2fa":false,"name":"livechat-mana
ger","protected":true,"scope":"Users"}],"success":true}mmcp108@Linux:~$ 
```

Integrate Rocket.chat with ELK:

In order to have an observability of the Rocket.chat server, I created a trial account on Elastic Cloud.

After that, I downloaded the heartbeat monitoring and installed the monitoring agent on my VM.

Configured the heartbeat settings file (heartbeat.yml) to point to the URL of my Rocket.chat server.

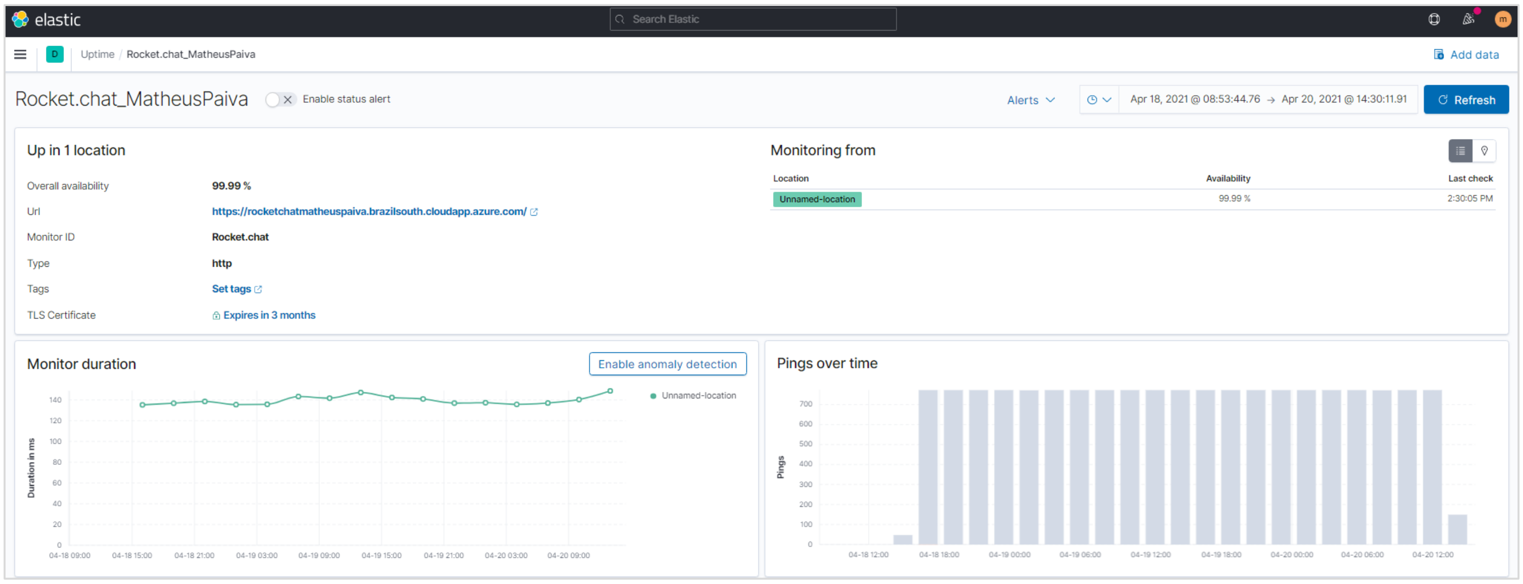
The heartbeat monitoring in ELK is now available and now I can monitor the Uptime of my Rocket.chat server.

https://i-o-optimized-deployment-2e5e8c.kb.us-west1.gcp.cloud.es.io:9243/app/uptime/monitor/Um9ja2V0LmNoYXQ=/

Login as Elastic Search user

User: Test

Password: RC$_17042021



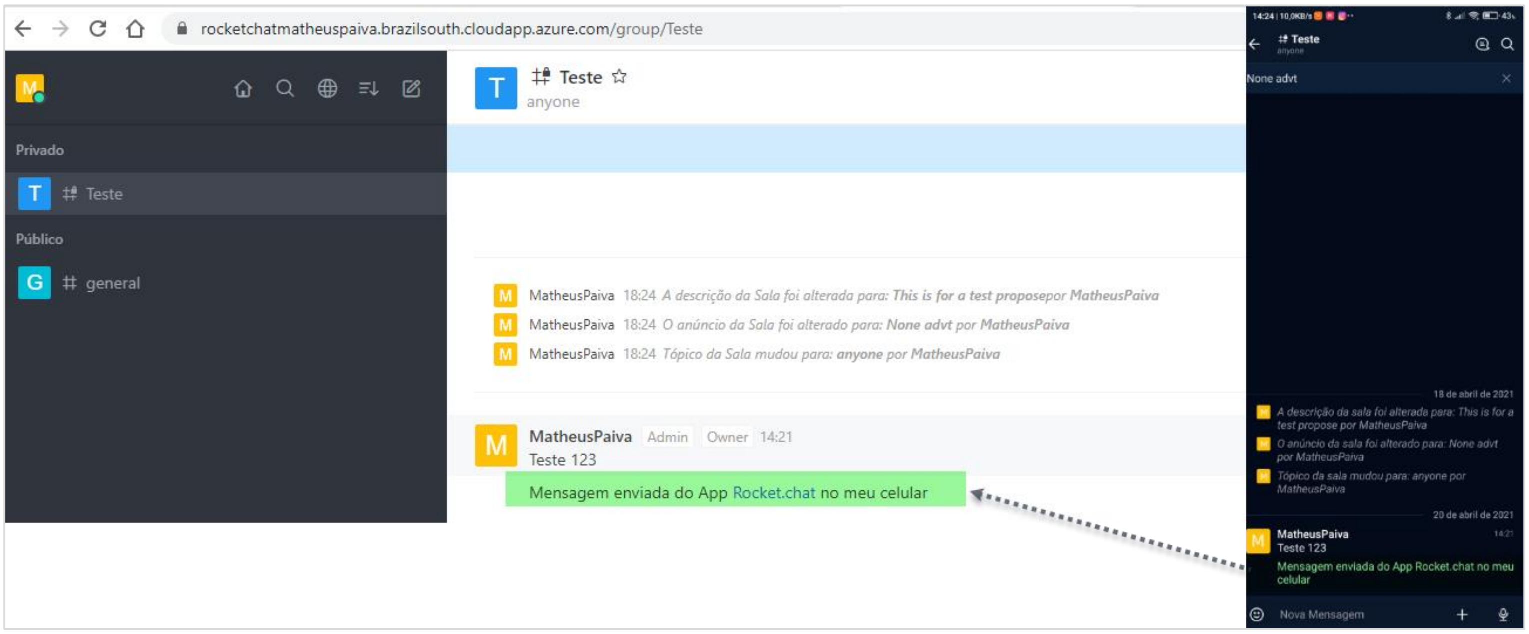
Up in 1 location

Overall availability	99.99 %
Url	https://rocketchatmatheuspaiva.brazilsouth.cloudapp.azure.com/
Monitor ID	Rocket.chat
Type	http
Tags	Set tags
TLS Certificate	Expires in 3 months

Monitoring from

Location	Availability	Last check
Unnamed-location	99.99 %	2:30:05 PM

Connect to my Rocket.chat server via Rocket.chat App (Android)



rocketchatmatheuspaiva.brazilsouth.cloudapp.azure.com/group/Teste

MatheusPaiva 18:24 *A descrição da Sala foi alterada para: **This is for a test propose**por **MatheusPaiva***

MatheusPaiva 18:24 *O anúncio da Sala foi alterado para: **None advt** por **MatheusPaiva***

MatheusPaiva 18:24 *Tópico da Sala mudou para: **anyone** por **MatheusPaiva***

MatheusPaiva Admin Owner 14:21
Teste 123
Mensagem enviada do App Rocket.chat no meu celular

18 de abril de 2021

A descrição da sala foi alterada para: This is for a test propose por MatheusPaiva

O anúncio da sala foi alterado para: None advt por MatheusPaiva

Tópico da sala mudou para: anyone por MatheusPaiva

20 de abril de 2021

MatheusPaiva 14:21
Teste 123
Mensagem enviada do App Rocket.chat no meu celular