

Documentação sobre instalação da ferramenta de monitoramento de métricas de CI

Atualizado em 02/01/2024

1. Configurações da Ferramenta:

Edite o arquivo **src/frontend/.env.prod**, é preciso colocar o endereço da API do backend de produção, como nos exemplos abaixo:

```
# colocar o endereco do backend de producao
# ao fazer o build do backend para um jar, automaticamente ele ja v
# a porta serah a porta que aplicação irah irá subir externamente:

VUE_APP_NAME = holter-devops
VUE_APP_PUBLIC_PATH = '/holter/'

# LOCAL TEST
VUE_APP_BACKEND_BASE_URL = http://localhost:8080/holter/api

# DTI PROJECT
#VUE_APP_BACKEND_BASE_URL = https://xxxxxx.imd.ufrn.br/holter/api

# LENOVO PROJECT
#VUE_APP_BACKEND_BASE_URL = http://XXX.XX.0.13:8200/holter/api

# ASSEMBLEIA PROJECT
#VUE_APP_BACKEND_BASE_URL = http://XXX.XX.70.165:8200/holter/api
```

Após esse passo, abra o arquivo **src/main/resources/application.properties** e altere as 5 propriedades abaixo de acordo com sua preferências:

```

#
# In the case study we will use login to check information of specific users
# default = false
#
enable.login-page=true

#
# IF the tool will access each access to the dashboard, this can take a lot of space.
# default = false
#
log.dashboard-access=true

#
# For the study we will use just CI metric, after the study other metric will be unlocked to the company
# default = false
#
lock.no-ci-metrics=true

#
# We will send alert by e-mail when automatic collect execute.
# we will make this feature optional. let's see if they ask for it.
# default = true
#
send.email-alert=true

#
# Disable ssl verification ONLY if the CI server do not use HTTPS ou its certificate if invalid.
# Local Gitlab and Sonar instances
# default = false
#
disable.ssl-verification=false

```

- **enable.login-page:** habilita ou desabilita o login no sistema, se login a ferramenta terá acesso público. **default = false**
- **log.dashboard-access:** Se a ferramenta registrará cada acesso ao dashboard. Desabilite essa opção, que foi usada apenas para o estudo de caso. **default = false**
- **lock.no-ci-metrics:** Limita as métricas apenas a métricas de CI. Desabilite essa opção para liberar todas as métricas da ferramenta. **default = false**
- **send.email-alert:** Se a ferramenta irá enviar e-mails de alerta sempre que rodar uma monitoramento. **default = true**
- **disable.ssl-verification:** Desativa a verificação SSL. utilize apenas SOMENTE se o seu servidor CI não usar HTTPS ou seu certificado for inválido ou para instâncias locais do Gitlab e Sonar. Caso contrário, mantenha habilitado. **default = false**

Altere também a propriedade com o endereço da aplicação.

```
#  
# Used just to send by e-mail  
# default value: http://localhost:8080/holter  
#  
app.url = https://xxxxxx.imd.ufrn.br/holter
```

2. Build da ferramenta

Para fazer o build da ferramenta basta digitar o comando docker baixo dentro do diretório principal da ferramenta, onde se encontrar o arquivo Dockerfile:

```
docker build . -t <instituicao>/holter-devops:vX
```

Onde:

- <instituicao>: substituir pelo nome da sua instituição
- vX: A versão da imagem

Exemplo:

```
docker build . -t ufrn/holter-devops:v1
```

3. Execução da ferramenta

Para executar a ferramenta digitar o comando baixo (em uma única linha):

```
docker container run -d -p 8200:8080 --env  
DATABASE_PATH=/data --env DATABASE_PASSWORD=sa123456 --env  
MAIL_ACCOUNT=xxxx@ufrn.br --env MAIL_PASSWORD='xxxx xxxx xxxx  
xxxx' --env TOKEN_SECRET=aaaaaaaa00000 -v /data/holter:/data  
--name holter-devops <instituicao>/holter-devops:vX
```

Onde:

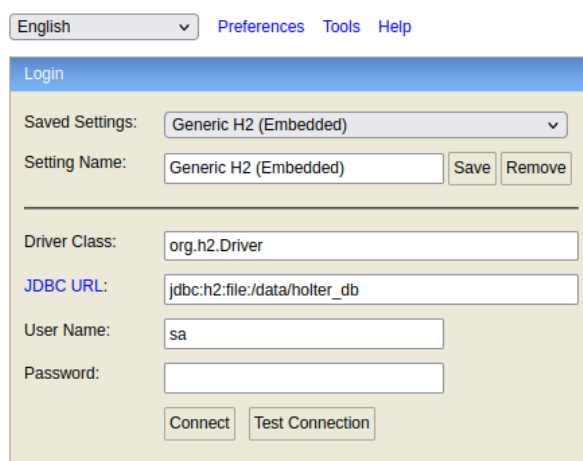
- **8200**: A porta onde a ferramenta executará.
- **sa123456**: A senha do banco de dados embutido na ferramenta.
- **xxxx@ufrn.br**: A conta de e-mail do google para enviar os alertas
- **'xxxx xxxx xxxx xxxx'**: A senha de app para enviar os emails (substitui a senha da conta do google).
<https://support.google.com/accounts/answer/185833?hl=pt-BR>
- **aaaaaaaa00000**: Hash sha512 (128 caracteres) aleatório que pode ser gerado nesse endereço: <https://emn178.github.io/online-tools/sha512.html>
- **/data/holter**: O diretório do servidor onde estará localizado o banco de dados da ferramenta (criado na primeira execução).
- **<instituicao>/holter-devops:vX**: O nome da imagem docker gerada no passo anterior.

Para acessar a ferramenta digite: <https://servidor.ufrn.br:8200/holter> (mesmo endereço que está na propriedade app.url no application.properties)

4. Configurando o Usuário inicial

Caso tenha configurado a ferramenta para ter login, para realizar o primeiro login na ferramenta é preciso cadastrar 1 usuário inicial via comando SQL, para isso acesse o console do banco de dados h2:

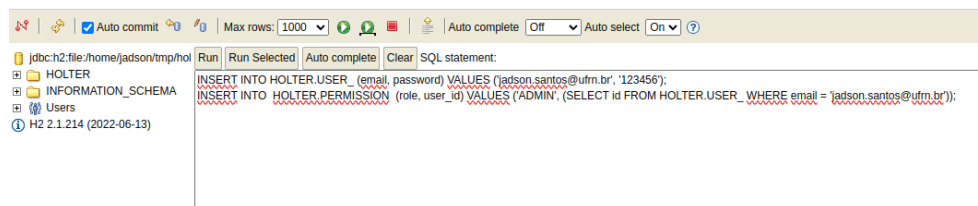
<https://servidor.ufrn.br:8200/holter/h2-console/>



No campo “Password” informe a senha do banco definido no comando docker run (`--env DATABASE_PASSWORD=sa123456`) e no campo JDBC URL digite o nome do arquivo do banco de dados (holter_db) definido também no comando docker run pelo parâmetro (`--env DATABASE_PATH=/data`), este último é o diretório dentro do docker, não é o diretório no servidor local: /data/holter que foi mapeado via: `-v /data/holter:/data`. Os 2 possuem os mesmos arquivos, mas no console do H2 só funciona o caminho do banco dentro do docker.

Agora insira o usuário inicial ADMIN inicial com os comandos SQL abaixo:

```
INSERT INTO HOLTER.USER_ (email, password) VALUES ('xxxxx.xxxx@ufrn.br', 'xxxxxx');
INSERT INTO HOLTER.PERMISSION (role, user_id) VALUES ('ADMIN', (SELECT id FROM HOLTER.USER_ WHERE email = 'xxxxx.xxxx@ufrn.br'));
```



Acesse a tela inicial da ferramenta que será possível realizar o login.

5. Considerações sobre a coleta de métricas

Algumas métricas, principalmente as que envolvem deploy do projeto só serão corretamente coletadas se os desenvolvedores seguirem um fluxo padronizado de desenvolvimento para a ferramenta conseguir coletar e calcular corretamente as métricas.

1 - Cálculo de métricas que envolvem erros/falhas em produção

É preciso ter uma clara definição do que seria uma issue relacionada a um erro/problema no sistema (normalmente marca-se um label indicando que ele é relacionada a erro). Os labels que definem issues de erro bem como a branch que contém o código de produção do projeto devem ter sido cadastrados nas configurações do projeto (**Projects -> Projects Configurations -> Campos "Issues Erros Labels" e "Production Branch"**).

Justificativa:

No GitLab, a taxa de falha é medida como a porcentagem de deploys que causam um incidente em produção no período determinado. O GitLab calcula isso pelo número de incidentes dividido pelo número de deploys em um ambiente de produção.

Isso pressupõe:

- *Os incidentes do GitLab são rastreados.*
- *Todos os incidentes estão relacionados a um ambiente de produção.*
- *Incidentes e deploys têm um relacionamento estritamente individual. Um incidente está relacionado a apenas um deploy em produção, e qualquer deploy em produção está relacionada a no máximo um incidente.*

Para simplificar calcularemos o número de incidentes em produção como "issues de erro" abertas para um projeto. Pois não sabemos se toda equipe utiliza esse conceito de "incidentes" e "ambiente de produção" existente no Gitlab. Além de não englobar outros repositórios com GitHub onde não existe esses conceitos.

2 - Cálculo de métricas que envolvem a frequência de deploy

Para calcular métricas que envolvem a frequência de deploy, a ferramenta considera como release do projeto as tags criadas no repositório. Sempre que uma nova tag é criada, a ferramenta considera que uma nova release foi lançada. 1 deploy = 1 tag criada no projeto.

Justificativa:

No GitLab, a frequência de deploy é medido pelo número médio de deploys por dia em um determinado ambiente, com base no horário de término do deploy (finished_at). O GitLab calcula a frequência de deploy como o número de deploys concluídos em um determinado dia. Somente implantações bem-sucedidas (Deployment.statuses = sucesso) são contadas. O GitLab usa o conceito de "Deployment". Para simplificar, consideramos uma tag como um deploy. Pois não sabemos se toda equipe utiliza esse conceito de "deployment" existente no Gitlab.

3 - Cálculo de métricas de CycleTime (tempo entre primeiro commit e fechamento da issue)

Para calcular a métrica de CycleTime que envolve o tempo entre o primeiro commit e a conclusão de uma tarefa, é preciso existir uma relação entre os commits e issues no Gitlab. A ferramenta usa o comentários dos commits para fazer essa relação, então para o cálculo correto dessa métrica, os comentários dos commits devem possuir o número da issue no formato #NNN - xxxxxxxx. Onde NNN é o número da issue. Exemplo de comentário de commit:

#98 - Issue de testes unitários