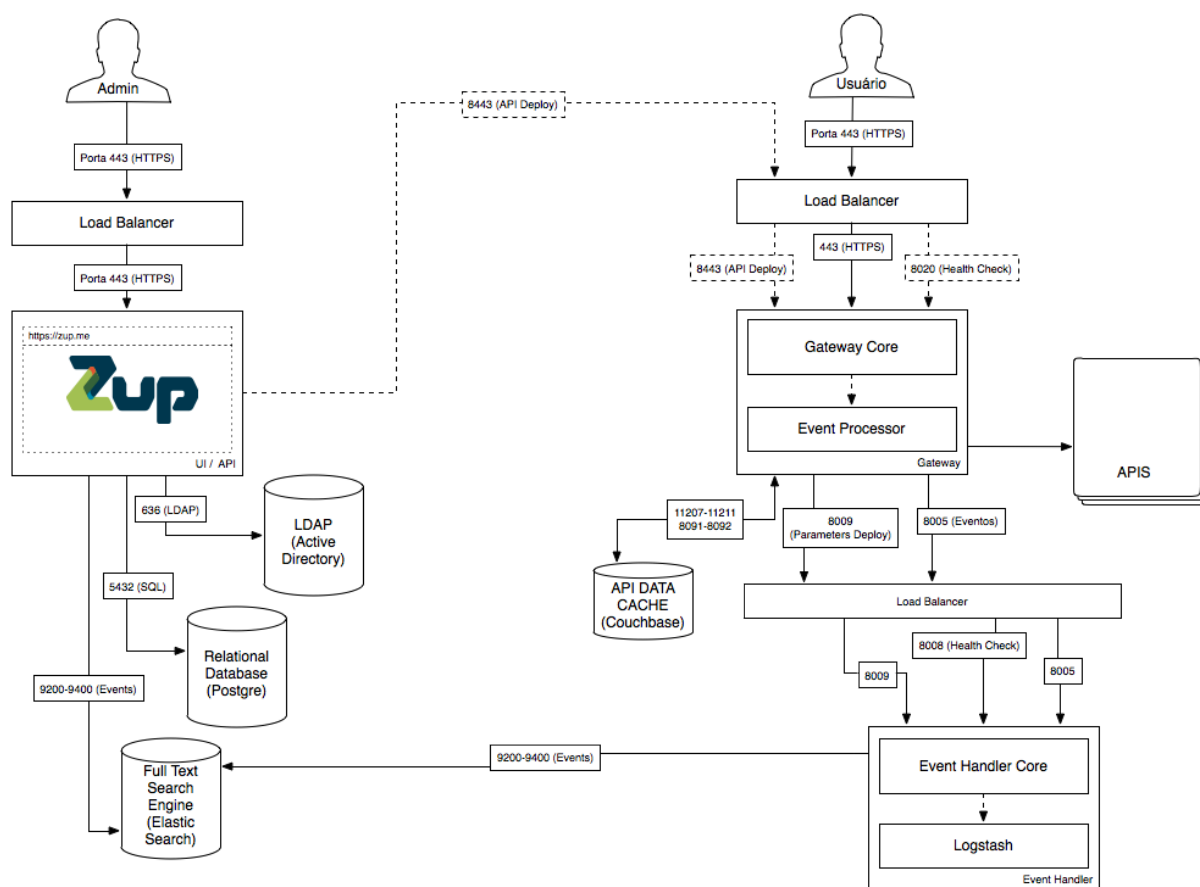


1. Introdução

Este documento tem o objetivo de instruir as equipes que irão instalar e operar o **Zup.me API Manager** em ambiente produtivo.

1.1. Arquitetura



1.2. Componentes

1.2.1. Api Manager

Componente através do qual os usuários interagem com a solução. Fazem parte desse componente o portal de configuração de APIs e pelo portal de desenvolvedores.

1.2.1.1. zupme-api-manager-api

Conjunto de APIs REST responsáveis pelo gerenciamento, criação e manutenção das APIs expostas pelo Gateway.

1.2.1.2. zupme-user-manager-api

Conjunto de APIs REST responsáveis pela integração com diferentes provedores de autenticação.

1.2.1.3. zupme-ui

Interface gráfica dos portais de configuração de APIs e do desenvolvedor. É composta apenas por arquivos estáticos (HTML, CSS e Javascript) que utilizam as APIs fornecidas pelos **api-manager-api** e **user-manager-api**.

1.2.2. PostgreSQL

Base de dados *design time*, onde se localiza toda a configuração de APIs e gerenciamento de usuários. Utilizado pelo API Manager API e User Manager API.

1.2.3. Gateway

1.2.3.1. zupme-gateway

Responsável pelo processamento, encaminhamento e modificação das requisições referentes às APIs configuradas no ambiente.

1.2.3.2. event-processor

Processamento assíncrono e envio de eventos para o componente Event Handler.

1.2.4. Couchbase

Base de dados não relacional de *runtime* onde é armazenada a estrutura de dados dos recursos de APIs e Cache de respostas.

1.2.5. Event Handler

1.2.5.1. zupme-event-handler

Recebe os eventos do gateway e redireciona para uma fila in-memory que será consumida pelo Logstash.

1.2.5.2. logstash

Trata, ofusca e persiste os eventos no Elasticsearch.

1.2.6. ElasticSearch

Plataforma de análise e pesquisa de dados em tempo real.

1.2.7. Commons Lib

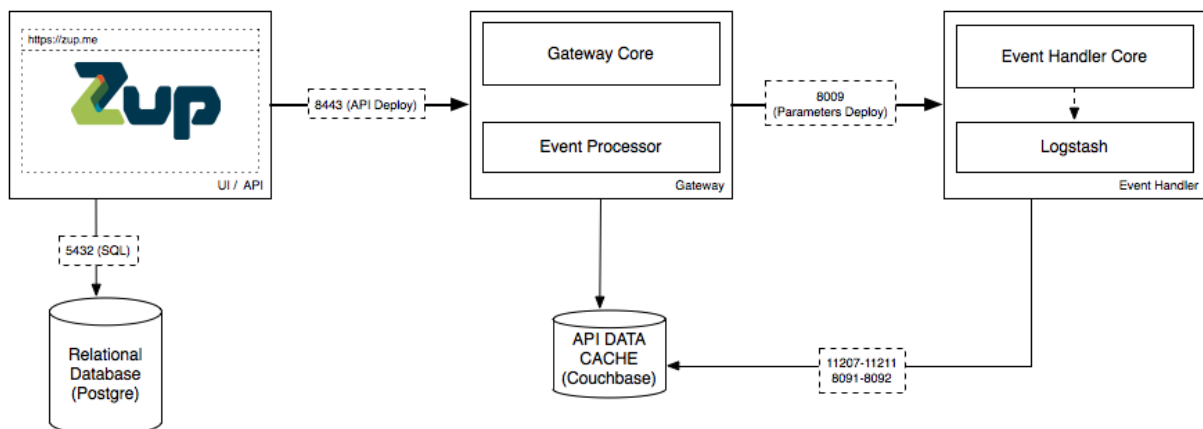
Biblioteca comum a todos os componentes responsável por prover operações como: extrações de dados, asserções lógicas e ofuscação.

1.3. Fluxos de Comunicação

Apresentamos a seguir os principais fluxos de comunicação da ferramenta, a partir dos quais é possível ter um melhor entendimento de como os elementos arquiteturais interagem para formar uma solução completa.

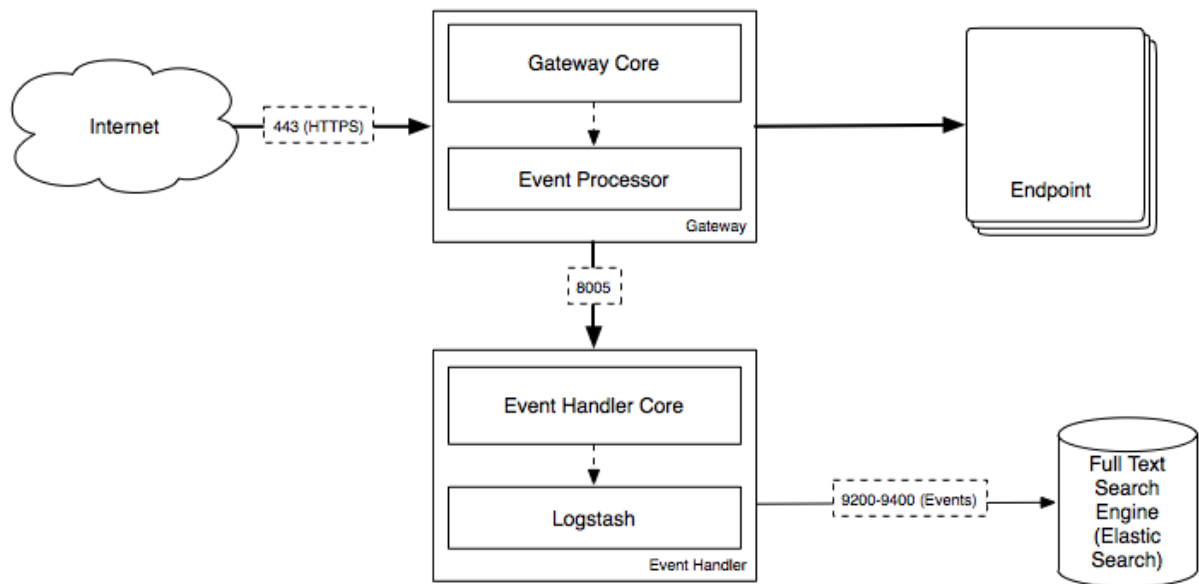
1.3.1. Deploy de APIs

1. Usuário cria API que é armazenada na base de dados PostgreSQL;
2. Usuário realiza o Deploy no Gateway (Entrypoint) desejado;
3. Configuração de APIs é enviada para o Gateway na porta de deploy configurada;
4. Gateway recebe configurações e armazena no Couchbase;
5. Gateway envia informações de ofuscação para o Event Handler;
6. Event Handler recebe configurações e armazena no Couchbase;
7. API está publicada e pronta para ser consumida.



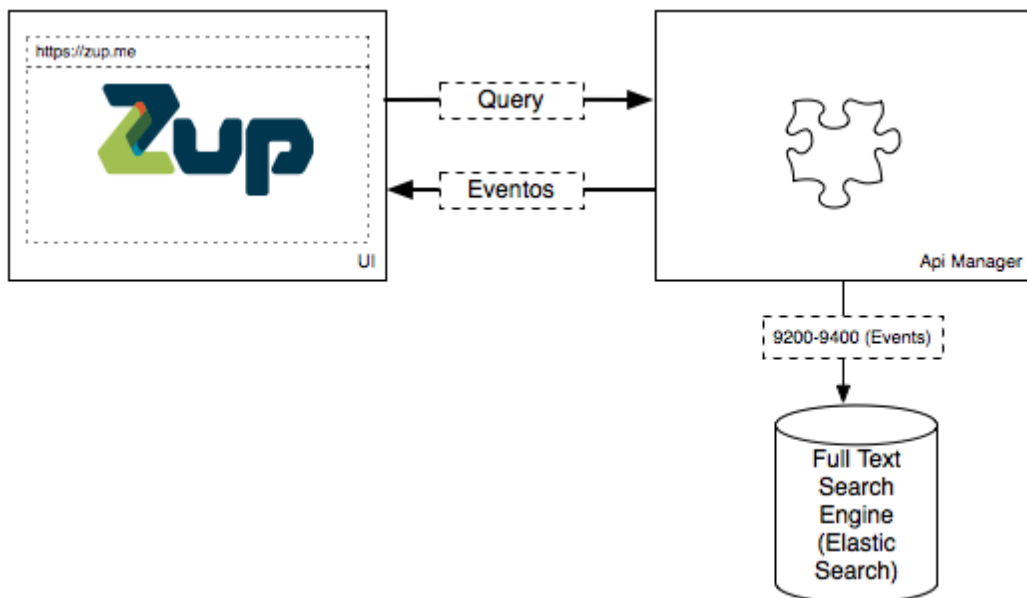
1.3.2. Processamento de Requisições

1. Aplicativo envia request para o Gateway;
2. Gateway aplica possíveis transformações no request e envia para o Endpoint;
3. Gateway envia response para o aplicativo após aplicar possíveis transformações;
4. Gateway envia evento para o Event Handler;
5. Event Handler envia evento para Logstash que aplica possíveis ofuscações e persiste no Elasticsearch.



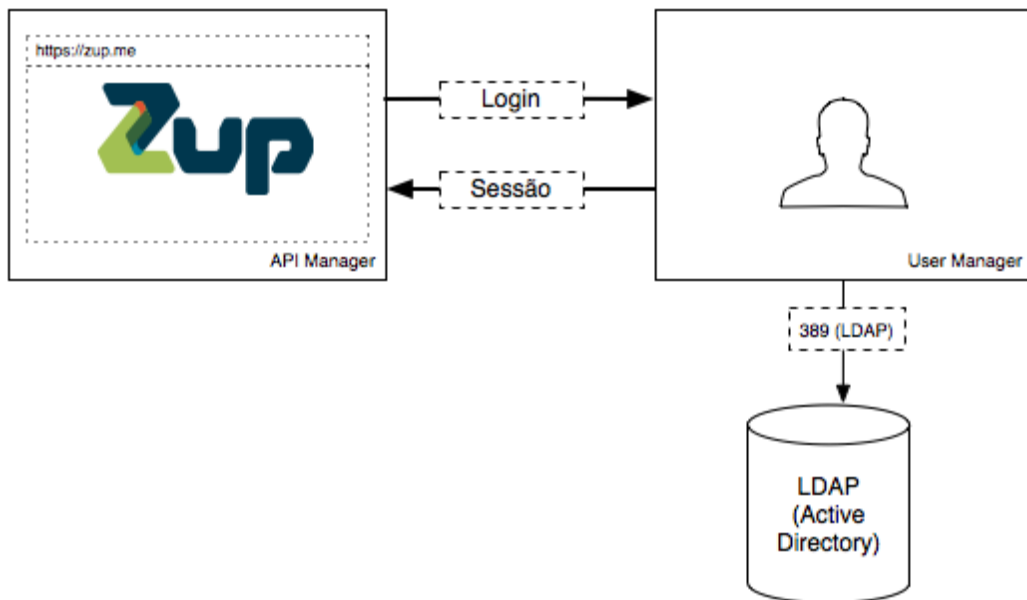
1.3.3. Visualização de Logs

1. Usuário cria query através da interface de usuário (UI);
2. UI envia query para API Manager;
3. API Manager realiza consulta no Elasticsearch;
4. Elasticsearch retorna eventos para o API Manager que exibe através da UI



1.3.4. Controle de Acesso

1. Usuário informa credenciais de acesso;
2. UI do API Manager envia credenciais informadas para User Manager (Login);
3. User Manager consulta o provedor de autenticação (LDAP, AD, etc..) e retorna uma sessão de acesso caso as credenciais sejam válidas.



2. Instalação

2.1. Pré-requisitos

É necessário pelo menos um servidor virtual ou bare-metal com pelo menos 4 núcleos, 20GB de disco e 4GB de memória ram para cada componente da solução. O sistema operacional necessário para a execução de todos os componentes é o Red Hat versão 6, 7 ou derivados, com exceção do componente *Forum*, que deve ser executado exclusivamente em sistemas operacionais Red Hat versão 7 ou derivados.

2.1.1. Máquina Cliente

Para realizar a instalação é necessário uma máquina que tenha acesso a todos os servidores, com um usuário root ou que tenha permissões de root (sudo).

Recomendamos que essa máquina esteja executando o sistema operacional Linux pois será necessário utilizar a ferramenta ansible, que não possui suporte para Windows.

2.1.2. Repositório RPM

Todos os servidores deverão ter acesso a internet ou repositórios locais do tipo Satellite/Spacewalk com acesso ao repositório da Zup (além de EPEL e repositórios padrões).

É necessário que seja informado o IP público do(s) servidores que irão utilizar ou sincronizar este repositório para que possa ser liberado o acesso.

https://s3-sa-east-1.amazonaws.com/zupme-packages-repository/CentOS/6/x86_64/production

ou

https://s3-sa-east-1.amazonaws.com/zupme-packages-repository/CentOS/7/x86_64/production

2.2. Entendendo a estrutura dos scripts

Utilizamos a ferramenta [Ansible](#), para automatizar o processo de instalação / atualização de componentes da plataforma. Escolhemos essa ferramenta pois já utilizamos outras ferramentas com o mesmo propósito no passado, e verificamos que o Ansible é mais simples e poderoso em recursos e possui uma curva de aprendizado menor.

Basicamente precisamos saber que existem os arquivos de **playbook**, onde esses contém tarefas (*tasks*) que são executadas sequencialmente através de uma conexão SSH, e os arquivos de **inventário** onde são informadas variáveis relativas aos componentes e ambientes específicos além dos nomes ou IPs dos hosts que serão acessados durante a execução do playbook.

Os arquivos de playbook tem a extensão **.yml** e normalmente não são alterados, pois eles são comuns a todos os ambientes. Os parâmetros que são dinâmicos e variam de acordo com ambiente devem ser especificados nos **inventários**.

Na raiz do diretório existem algumas pastas, sendo as mais relevantes para nós as que contém os playbooks e os inventários:

- ansible/
 - componentes/
 - playbook.yml
- customers/
 - ambientes/
 - componentes/
 - inv/
 - hosts

O Ansible funciona da seguinte maneira:

```
$ ansible-playbook --parametros /caminho_do_playbook/playbook.yml -i /caminho_do_inventário/hosts
```

Os parâmetros podem ser diversos. Os que são geralmente utilizados:

--ask-pass

Pergunta a senha do usuário que será utilizada na conexão SSH.

`--ask-sudo-pass`

Pergunta a senha para elevar o privilégio do usuário.

`--user`

Especifica o usuário que será utilizado para conectar no SSH.

Para não ser necessário especificar longos caminhos e parâmetros relativos ao Ansible, nós abstraímos a linha de comando do Ansible com os shell-scripts ***zupme.sh***, ***deploy.sh*** e ***provision.sh***.

A primeira vez em que o script ***zupme.sh*** é executado ele pergunta qual usuário vai ser utilizado para a conexão SSH e cria um arquivo oculto ***.zupme-deploy.conf*** no diretório */home* do usuário que estiver executando. É possível forçar a execução do script com outro usuário usando o parâmetro `--user`.

Os *playbooks* de provisionamento são executados em ambientes novos, onde a plataforma nunca foi configurada e suas tasks apenas preparam o ambiente, instalando dependências e certificando-se de configurações, para que então possamos executar os *playbooks* de deploy para instalar pacotes dos componentes desenvolvidos pela Zup e suas respectivas versões.

A abstração funciona da seguinte maneira:

```
$ /<zupme-deploy-root>/bin zupme deploy --module=<componente>  
--customer=<customer> --env=<ambiente> --version=<versão>
```

2.3. Local dos scripts de deploy

Os scripts de deploy do Ansible podem ser encontrados no endereço:

[https://s3-sa-east-1.amazonaws.com/zupme-packages-repository/files/zupme-\[customer\]-deploy-latest.tar.bz2](https://s3-sa-east-1.amazonaws.com/zupme-packages-repository/files/zupme-[customer]-deploy-latest.tar.bz2)

Esse é o endereço do Object Storage da Zup na Amazon, e é necessário fazer a liberação do IP de origem. O IP liberado geralmente é aquele utilizado pelo SpaceWalk / Satellite para fazer o sincronismo dos repositório.

2.4. commons-lib

2.4.1. Provision

```
$ bin/zupme deploy --module=cockpit-commons-lib  
--customer=<customer> --env=<env> --version=<version>
```

2.5. ElasticSearch

2.5.1. Provision

```
$ bin/zupme provision --module=elasticsearch --customer=<customer>  
--env=<env> --skip-tags=running_cluster
```

2.6. PostgreSQL

2.6.1. Provision

```
bin/zupme provision --module=postgresql --customer=<customer>  
--env=<env>
```

2.6.2. Create User

Logar na máquina primária do PostgreSQL e criar o usuário master utilizando o usuário de sistema postgres:

```
sudo -s -u postgres  
psql -c "create role <username> with createdb login password  
'<user_password>'"
```

2.7. Api Manager Api

```
bin/zupme provision --module=zupme-api-manager-api  
--customer=<customer> --env=<env>
```


2.7.1. Certificado SSL

Verificar em `customers/<customer>/<env>/zupme-api-manager-api/conf/nginx/nginx.conf` o caminho esperado para o certificado SSL. Procure por linhas assim:

...

```
ssl_certificate /etc/ssl/certs/zupme.crt;  
ssl_certificate_key /etc/ssl/certs/zupme.key;
```

...

Caso esteja utilizando selinux, será necessário rodar o seguinte comando no diretório para onde foram copiados os certificados:

```
sudo restorecon -v -R /etc/ssl/certs
```

Copie o certificado e a chave para os locais esperados antes de realizar o deploy.

2.7.2. Deploy

Apenas no primeiro deploy configure as seguintes variáveis no inventário, `customers/<customer>/<env>/zupme-api-manager-api/inv/hosts`:

```
change_database_url=y  
create_database=y
```

No primeiro deploy é necessário informar a URL de conexão com o banco de dados, que será perguntada pelo script. O formato é:

```
postgres://<username>:<password>@<host>/<database_name>.
```

```
bin/zupme deploy --module=zupme-api-manager-api  
--customer=<customer> --env=<env> --version=<version>
```

Após o deploy volte as variáveis:

```
change_database_url=n  
create_database=n
```

2.8. User Manager Api

Normalmente

2.8.1. Deploy

```
bin/zupme deploy --module=zupme-user-manager-api  
--customer=<customer> --env=<env> --version=<version>
```

Apenas no primeiro deploy configure as seguintes variáveis no inventário,
customers/<customer>/<env>/zupme-api-manager-api/inv/hosts:

```
change_database_url=y  
create_database=y
```

No primeiro deploy é necessário informar a URL de conexão com o banco de dados,
que será perguntada pelo script. O formato é:

```
postgres://<username>:<password>@<host>/<database_name>.
```

Após o deploy volte os valores das variáveis:

```
change_database_url=n  
create_database=n
```

Agora é preciso configurar a Organização padrão:

```
sudo zupme-user-manager-api run bin/config/organization create_default  
--name Organization --slug organization_slug
```

```
sudo zu      pme-user-manager-api run bin/config/organization  
auth_provider --strategy active_directory \  
--user_creation_allowed false --password_reset_allowed false  
--user_confirmation_required false \  
--settings host:bcocorporate.ad port:636 base_dn:"OU=Internal  
Users,DC=bcocorporate,DC=ad" filter_attribute:sAMAccountName \  
auth_method:simple  
auth_username:"CN=ORG,OU=Users,DC=corp,DC=ad"  
auth_password:<auth_password> \  
encryption: simple_tls \  
--user_mapping uid:sAMAccountName name:givenname  
displayname:displayName email:mail \  
--organization_slug organization_slug
```

```
sudo zupme-user-manager-api run bin/config/admin_user create
--admin-user-name=Admin \
--organization-slug=organization_slug
--admin-user-role=general_administrator --admin-user-uid=admin \
--admin-user-display-name=Admin
--admin-user-email=admin@organization.com.br
```

2.9. Zupme UI

2.9.1. Deploy

```
bin/zupme deploy --module=zupme-ui --customer=<customer>
--env=<env> --version=<version>
```

2.10. CouchBase

2.10.1. Provision Cluster

```
ansible-playbook --ask-pass --ask-sudo-pass -u <user>
ansible/couchbase/provision-cluster.yml \
-i customers/<customer>/<env>/couchbase/inv/hosts --extra-vars
"ansible_distribution=RedHat"
```

2.10.2. Provision Buckets

```
ansible-playbook --ask-pass --ask-sudo-pass -u <user>
ansible/couchbase/provision-buckets.yml \
-i customers/<customer>/<env>/couchbase/inv/hosts --extra-vars
"ansible_distribution=RedHat"
```

2.10.3. Provision Views

```
ansible-playbook --ask-pass --ask-sudo-pass -u <user>
ansible/couchbase/provision-views.yml \
-i customers/<customer>/<env>/couchbase/inv/hosts --extra-vars
"ansible_distribution=RedHat"
```

2.11. Event Handler

2.11.1. Provision

```
$ bin/zupme provision --module=zupme-event-handler  
--customer=<customer> --env=<env>
```

2.11.2. Deploy

```
$ bin/zupme deploy --module=zupme-event-handler  
--customer=<customer> --env=<env> --version=<version>
```

2.12. Logstash

Instalar git na máquina.

2.12.1. Provision

```
$ bin/zupme provision --module=logstash --customer=<customer>  
--env=<env>
```

2.13. Gateway

2.13.1. Provision

```
bin/zupme provision --module=zupme-gateway --customer=<customer>  
--env=<env>
```

2.13.2. Certificado SSL

Verificar em `customers/<customer>/<env>/zupme-gateway/conf/environments/config.json` o caminho esperado para o certificado SSL. Procure por linhas assim:

...

```
"cert": "/etc/ssl/certs/zupme.crt",  
"key": " /etc/ssl/certs/zupme.key",
```

...

Copie o certificado e a chave para os locais esperados antes de realizar o deploy.

Pode ser necessário rodar o seguinte comando na pasta para onde foram copiados os certificados:

```
sudo restorecon -v -R /etc/ssl/certs
```

2.13.3. Deploy

```
bin/zupme deploy --module=zupme-gateway --customer=<customer>  
--env=<env> --version=<version>
```

3. Atualização

3.1. Introdução

O processo de atualização é executado em ambientes que já foram previamente provisionados e que já estão em operação, portanto é esperado que as máquinas estejam atendendo os requisitos descritos na seção 2.1.

3.2. Componentes

3.2.1. cockpit-commons-lib

Executando pelo shell-script de abstração:

```
$ bin/zupme deploy --module=cockpit-commons-lib --customer=<customer>  
--env=<ambiente> --version=<version>
```

Executando diretamente pelo Ansible:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <usuario>  
ansible/cockpit-commons-lib/deploy.yml -i  
customers/<customer>/<ambiente>/cockpit-commons-lib/inv/hosts
```

As versões serão informadas previamente ao cliente quando agendarmos a atualização.

3.2.2. Api Manager Api

Como esse componente interage com a base de dados PostgreSQL, certifique-se que as variáveis abaixo estejam configuradas com 'n':

- `change_database_url=n`
- `create_database=n`

Antes de iniciar a execução do playbook, o script pergunta o caminho, usuario, senha e nome da base dados para fazer alterações, apenas aperte ENTER sem escrever nada.

Executando pelo shell-script de abstração:

```
$ bin/zupme deploy --module=zupme-api-manager-api --customer=<customer>
--env=<ambiente> --version=<version>
```

Executando diretamente pelo Ansible:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <usuario>
ansible/zupme-api-manager-api/deploy.yml -i
customers/<customer>/<ambiente>/zupme-api-manager-api/inv/hosts
```

As versões serão informadas previamente ao cliente quando agendarmos a atualização.

3.2.3. User Manager Api

Como esse componente interage com a base de dados PostgreSQL, certifique-se que as variáveis abaixo estejam configuradas com 'n':

- `change_database_url=n`
- `create_database=n`

Antes de iniciar a execução do playbook, o script pergunta o caminho, usuario, senha e nome da base dados para fazer alterações, apenas aperte ENTER sem escrever nada.

Executando pelo shell-script de abstração:

```
$ bin/zupme deploy --module=zupme-user-manager-api --customer=<customer>
--env=<ambiente> --version=<version>
```

Executando diretamente pelo Ansible:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <usuario>
ansible/zupme-user-manager-api/deploy.yml -i
```

```
customers/<customer>/<ambiente>/zupme-user-manager-api/inv/hosts
```

As versões serão informadas previamente ao cliente quando agendarmos a atualização.

3.2.4. Zupme UI

Executando pelo shell-script de abstração:

```
$ bin/zupme deploy --module=zupme-ui --customer=<customer>  
--env=<ambiente> --version=<version>
```

Executando diretamente pelo Ansible:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <usuario>  
ansible/zupme-ui/deploy.yml -i  
customers/<customer>/<ambiente>/zupme-ui/inv/hosts
```

As versões serão informadas previamente ao cliente quando agendarmos a atualização.

3.2.5. Event Handler

Executando pelo shell-script de abstração:

```
$ bin/zupme deploy --module=zupme-event-handler --customer=<customer>  
--env=<ambiente> --version=<version>
```

Executando diretamente pelo Ansible:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <usuario>  
ansible/zupme-event-handler/deploy.yml -i  
customers/<customer>/<ambiente>/zupme-event-handler/inv/hosts
```

As versões serão informadas previamente ao cliente quando agendarmos a atualização.

3.2.6. Gateway

Executando pelo shell-script de abstração:

```
$ bin/zupme deploy --module=zupme-gateway --customer=<customer>
--env=<ambiente> --version=<version>
```

Executando diretamente pelo Ansible:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <usuario>
ansible/zupme-gateway/deploy.yml -i
customers/<customer>/<ambiente>/zupme-gateway/inv/hosts
```

As versões serão informadas previamente ao cliente quando agendarmos a atualização.

3.2.7. Logstash

Eventualmente pode ser necessário atualizar algum *plugin* ou configuração do Logstash.

Nesse caso basta rodar o provisionamento novamente:

```
$ bin/zupme provision --module=logstash --customer=<customer>
--env=<env>
```

4. Troubleshooting Guide

4.1. Como reiniciar o Api Manager Api

```
$ sudo service zupme-api-manager-api stop
$ sudo service zupme-api-manager-api start
```

Aguarde alguns instantes e verifique o log:

```
$ sudo zupme-api-manager-api logs -tail
```

4.2. Como reiniciar o User Manager Api

```
$ sudo service zupme-user-manager-api stop
$ sudo service zupme-user-manager-api start
```

Aguarde alguns instantes e verifique o log:


```
$ sudo zupme-user-manager-api logs -tail
```

4.3. Como reiniciar o Gateway

```
$ sudo service zupme-gateway stop  
$ sudo rm -rf /var/run/zupme-gateway/*.ipc  
$ sudo service zupme-gateway start
```

Aguarde alguns instantes e verifique o log:

```
$ sudo zupme-gateway logs -tail
```

4.4. Como reiniciar o Event Handler

```
$ sudo service cockpit-event-handler stop  
$ sudo rm -rf /opt/cockpit-event-handler/*.ipc  
$ sudo service cockpit-event-handler start
```

Aguarde alguns instantes e verifique o log:

```
$ sudo cockpit-event-handler logs -tail
```

Também pode ser necessário reiniciar o Logstash, que sempre fica na mesma máquina do Event Handler:

```
$ sudo service logstash restart
```

Os logs do Logstash estão em: `/var/log/logstash`

4.5. Problemas comuns

4.5.1. Erro ao logar

Quando um usuário tenta usar suas credenciais de acesso para entrar na plataforma, e um erro desconhecido é retornado, geralmente a causa é a falha na consulta do User Manager com o servidor LDAP.

Verificar status do serviço do User Manager:

```
$ sudo service zupme-user-manager-api stop
$ sudo service zupme-user-manager-api start
```

Se o serviço estiver UP o problema pode ser a consulta no LDAP, verificar se houve alteração na senha do usuário que faz o bind com o LDAP. Executando tcpdump com origem o IP do User Manager e destino servidor LDAP na porta de acesso (389 ou 689) é possível ver e confirmar essa situação.

4.5.2. Ao logar não é exibido nenhum produto

Quando as credenciais do usuário são aceitas, mas ao logar não aparece nenhum produto para interagir, pode ser por que o usuário em questão não possui privilégios cadastrados para acesso.

Em raras situações isso pode ocorrer por algum erro ao realizar o bind no LDAP, nesse caso é necessário revisar os atributos de bind.

4.5.3. Erro 500 ao realizar deploy

Erro 500 (Internal Server Error) pode estar relacionado a muitos motivos. Nesse caso é preciso verificar os logs para tentar diagnosticar. Os motivos mais frequentes são:

4.5.3.1. Falha de comunicação entre o zupme-api-manager-api e o zupme-gateway

Verificar no log do zupme-api-manager-api, utilizando :

```
# zupme-api-manager-api logs --tail
```

na máquina de api-manager-api.

4.5.3.2. Falha de comunicação entre o zupme-gateway e o zupme-event-handler

Verificar se há conectividade na porta 8005 do Gateway para o Event-Handler.

Verificar no log do zupme-gateway, utilizando :

```
# zupme-gateway logs --tail
```

na máquina de Gateway.

4.5.3.3. Falha de comunicação entre o zupme-gateway e o couchbase

Verificar se há conectividade na porta 11211, 8091 e 8092 do Gateway para o Couchbase.

Verificar no log do zupme-gateway, utilizando :

```
# zupme-gateway logs --tail
```

na máquina de Gateway.

4.5.3.4. Falha de comunicação entre o zupme-event-handler e o couchbase

Verificar se há conectividade na porta 11211, 8091 e 8092 do Event Handler para o Couchbase.

Verificar no log do zupme-event-handler, utilizando :

```
# cockpit-event-handler logs --tail
```

na máquina de Event Handler.

4.5.4. Os logs não são exibidos

Quando não são exibidos Logs na sessão de Logs da aplicação de um dia para o outro, pode ser que os índices e aliases do Elasticsearch não tenham sido criados.

O processo de criação de índices e aliases funciona com dois Jobs que rodam na máquina do Api Manager Api:

- O processo job_scheduler agenda a criação de índices e aliases diariamente às 19:00;
- O processo es_worker executa os jobs agendados.

Para identificar porque os índices e aliases não foram criados deve-se analisar os logs desses dois processos:

```
$ tail -f /var/log/zupme-api-manager-api/job_scheduler-1.log
$ tail -f /var/log/zupme-api-manager-api/es_worker-1.log
```

4.5.4.1. Como verificar se os índices e aliases do Elasticsearch foram criados

```
$ curl http://<elasticsearch-host>:9200/_cat/indices | grep 20160524
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
124   372   124   372    0    0  26620      0 --:--:-- --:--:-- --:--:-- 33818
green open events-org_slug-20160524 2 1 445 0 14.8mb 14.8mb
```

O correto e retornar pelo menos um índice, como mostrado acima.

Agora verificando os aliases:

```
$ curl http://<elasticsearch-host>:9200/_cat/aliases | grep 20160524
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
 0      0    0    0    0    0    0      0 --:--:-- --:--:-- --:--:--
0events-org_slug-10-1-201605 events-org_slug-20160524 * - -
events-org_slug-10-1-20160524 events-org_slug-20160524 * - -
events-org_slug-11-4-20160524 events-org_slug-20160524 * - -
```

events-org_slug-2-3-20160524	events-org_slug-20160524 * - -
events-org_slug-9-2-201605	events-org_slug-20160524 * - -
events-org_slug-2-4-201605	events-org_slug-20160524 * - -
events-org_slug-11-3-20160524	events-org_slug-20160524 * - -
events-org_slug-2-1-201605	events-org_slug-20160524 * - -
events-org_slug-10-4-201605	events-org_slug-20160524 * - -
events-org_slug-9-2-20160524	events-org_slug-20160524 * - -
events-org_slug-11-2-201605	events-org_slug-20160524 * - -
events-org_slug-2-2-20160524	events-org_slug-20160524 * - -
events-org_slug-2-3-201605	events-org_slug-20160524 * - -
events-org_slug-9-3-201605	events-org_slug-20160524 * - -
events-org_slug-9-3-20160524	events-org_slug-20160524 * - -
events-org_slug-11-3-201605	events-org_slug-20160524 * - -
events-org_slug-10-3-201605	events-org_slug-20160524 * - -
events-org_slug-11-1-201605	events-org_slug-20160524 * - -
events-by-dev-portal-org_slug-1-1-201605	events-org_slug-20160524 * - -
events-org_slug-11-1-20160524	events-org_slug-20160524 * - -
events-by-dev-portal-org_slug-1-1-20160524	events-org_slug-20160524 * - -
events-org_slug-9-4-201605	events-org_slug-20160524 * - -
events-org_slug-2-4-20160524	events-org_slug-20160524 * - -
events-org_slug-10-3-20160524	events-org_slug-20160524 * - -
events-org_slug-9-4-20160524	events-org_slug-20160524 * - -
events-org_slug-9-1-201605	events-org_slug-20160524 * - -
events-org_slug-2-2-201605	events-org_slug-20160524 * - -
events-org_slug-10-2-201605	events-org_slug-20160524 * - -
events-org_slug-2-1-20160524	events-org_slug-20160524 * - -
events-org_slug-10-2-20160524	events-org_slug-20160524 * - -
events-org_slug-9-1-20160524	events-org_slug-20160524 * - -
events-org_slug-11-2-20160524	events-org_slug-20160524 * - -
events-org_slug-10-4-20160524	events-org_slug-20160524 * - -
events-org_slug-11-4-201605	events-org_slug-20160524 * - -

O correto é retornar vários aliases como mostrado acima.

Para forçar a criação de índices e aliases, deve-se executar uma task do zupme-api-manager-api. Entre na máquina do Api Manager Api e execute para criar os índices e aliases do dia:

```
$ sudo zupme-api-manager-api run rake zupme:create_es_indexes
```

Para criar os índices e aliases dos últimos 5 dias:

```
$ sudo zupme-api-manager-api run rake zupme:create_last_5_days_indexes
```

Faça a verificação novamente para se certificar que tudo foi criado corretamente.

Quando repentinamente os logs não estão sendo exibidos mas os índices e aliases foram criados, pode ser alguma falha de comunicação do Gateway com o Elasticsearch. Nesse caso, verifique a comunicação na porta 9200 e 9300 do Gateway para o Elasticsearch.

Caso não seja possível estabelecer conexão, verificar o status do serviço do Elasticsearch e do Logstash.

5. Configurações

5.1. Configurar listeners dos componentes em IPv4 e IPv6

5.1.1. Api Manager

O padrão do componente é escutar IPv4 e IPv6 simultaneamente, configurações adicionais não são necessárias.

5.1.2. Gateway

No arquivo de configuração do mesmo, para IPv4, se espera:

```
{
  "server": {
    "gateway": {
      "name": "gateway",
      "address": "0.0.0.0",
      "port": 443,
    (... )
  }
}
```

Para IPv6, se espera:

```
{
  "server": {
    "gateway": {
      "name": "gateway",
      "address": ":::",
      "port": 443,
    (... )
  }
}
```

5.2. Alterar certificados utilizados no HTTPS

5.2.1. API Manager

Verificar em `customers/<customer>/<env>/zupme-api-manager-api/conf/nginx/nginx.conf` o caminho esperado para o certificado SSL. Procure por linhas assim:

```
(...)  
    ssl_certificate /etc/ssl/certs/zupme.crt;  
    ssl_certificate_key /etc/ssl/certs/zupme.key;  
(...)
```

Copie o certificado e a chave para os locais esperados e reinicie o serviço através do comando:

```
$ sudo service zupme-api-manager-api restart
```

5.2.2. Gateway

Verificar em customers/<customer>/<env>/zupme-gateway/conf/environments/config.json o caminho esperado para o certificado SSL. Procure por linhas assim:

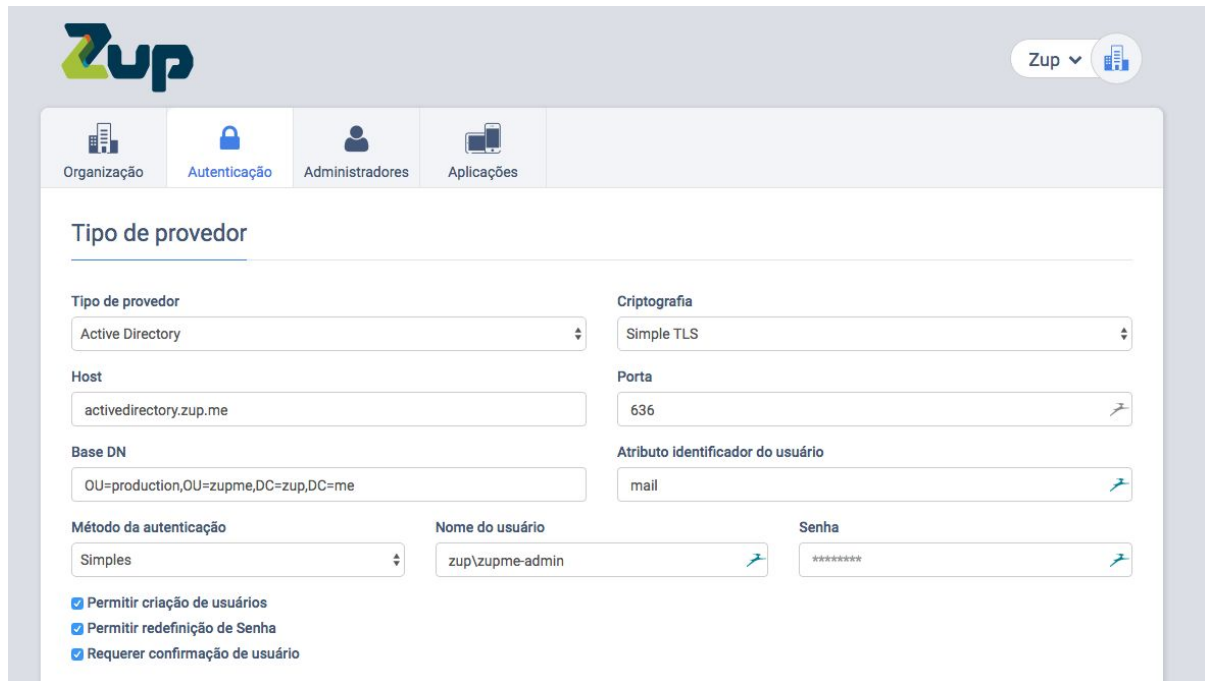
```
...  
    "cert": "/etc/ssl/certs/zupme.crt",  
    "key": " /etc/ssl/certs/zupme.key",  
...
```

Copie o certificado e a chave para os locais esperados e reinicie o serviço através do comando:

```
$ sudo service zupme-gateway restart
```

5.3. Alterar conexão ao LDAP

5.3.1. Configuração do provider



The screenshot shows the Zup web interface with the 'Autenticação' (Authentication) tab selected. The 'Tipo de provedor' (Provider Type) section is active, displaying configuration fields for an Active Directory provider. The fields are as follows:

Tipo de provedor	
Tipo de provedor	Criptografia
Active Directory	Simple TLS
Host	Porta
activedirectory.zup.me	636
Base DN	Atributo identificador do usuário
OU=production,OU=zupme,DC=zup,DC=me	mail
Método da autenticação	Nome do usuário
Simples	zup\zupme-admin
	Senha

Below the fields, there are three checked checkboxes:

- ☒ Permitir criação de usuários
- ☒ Permitir redefinição de Senha
- ☒ Requerer confirmação de usuário

Tipo do provedor: LDAP ou Active Directory

Criptografia: Simple TLS ou Start TLS

Host: Nome do host do servidor

Porta: Porta em que o serviço escuta

Base DN: DN base para pesquisa

Atributo identificador do usuário: atributo que identifica unicamente o usuário

Método de autenticação: Simples ou Anônimo

Nome do usuário: usuário de bind

Senha: senha do usuário de bind

5.3.2. Mapeamento de usuários e grupos

Mapear campos LDAP

Usuários

Nome	Atributo identificador (uid)
displayName	mail
Nome público (display_name)	Email
givenname	mail
Grupos do usuário (memberof)	
	memberof

Grupos

Nome (name)	Atributo identificador (distinguishedname)	Membros do grupo (members)
name	distinguishedname	member

Salvar

5.3.2.1. Usuários

Nome: atributo que contém o nome do usuário

Atributo Identificador: atributo que identifica unicamente o usuário

Nome público: atributo que contém o nome do usuário

Email: atributo que contém o email do usuário

Grupos do usuário: atributo que contém os grupos do usuário

5.3.2.2. Grupos

Nome: atributo que contém o nome do grupo

Atributo Identificador: atributo que identifica unicamente o grupo

Membros do grupo: atributo que contém os membros do grupo

5.4. Alteração de servidor SMTP

1. Alterar as configurações a seguir no inventário do componente zupme-api-manager-api em <customers>/<customer>/<environment>/zupme-api-manager-api/conf/environments/config.yml.

```
smtp_address:
smtp_port:
smtp_username:
smtp_password:
smtp_from:
```

2. Realizar o deploy do componente através do comando:


```
$ bin/zupme deploy --module=zupme-api-manager-api  
--version=<versao_executada> --customer=<customer> --env=<ambiente>
```

5.5. zupme-api-manager-api

As configurações do **zupme-api-manager-api** feitas através de variáveis de ambiente. O arquivo

/<zupme-deploy-root>/customers/<customer>/<env>/zupme-api-manager-api/config/environments/config.yml contém todas as configurações, que são aplicadas em momento de *deploy*, por isso recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo que todas instâncias do componente tenham as alterações aplicadas.

Para listar as configurações em vigor:

```
$ sudo zupme-api-manager-api config
```

Para ver uma configuração específica:

```
$ sudo zupme-api-manager-api config:get CONFIG_NAME
```

Para modificar uma configuração específica:

```
$ sudo zupme-api-manager-api config:set CONFIG_NAME=NEW_VALUE
```

Chaves de configuração:

URLs dos nós do Elasticsearch configurados como data_nodes

```
elasticsearch_urls: "http://es_host1:9200,http://es_host2:9200"
```

Certificados utilizados para realizar deploy de APIs para o Gateway

```
client_cert: "/opt/zupme-api-manager-api/certs/zupme-api-manager-api.crt"
```

```
client_key: "/opt/zupme-api-manager-api/certs/zupme-api-manager-api.key"
```

```
ca_file: "/opt/zupme-api-manager-api/certs/zupme-ca.crt"
```

Certificado e CA utilizados para comunicação entre apis (api-manager-api <> user-manager-api)

```
zupme_client_application_certificate:
```

```
/opt/zupme-api-manager-api/certs/api_communication.crt
```

```
zupme_api_communication_public_key_path:
```

```
/opt/zupme-api-manager-api/certs/zupme-ca.pub
```

Chave pública utilizada para validar sessões

```
zupme_user_manager_public_key:
```

```
/opt/zupme-api-manager-api/certs/user_manager_session.pub
```

Configurações de SMTP

```
smtp_url_options:
```

```
smtp_address:
```

```
smtp_port:
```

```
smtp_username:
smtp_password:
smtp_domain:
smtp_from:
```

Configurações de número máximo de threads e conexões com o Banco de Dados Postgre

```
max_threads: 50
db_pool: 50
```

URL pública do zupme-user-manager-api

```
zupme_user_manager_api_url: "https://user-manager-api.domain.corp"
zupme_client_application_id: "3"
```

Domínios onde se é permitido CORS por padrão

```
api_cors_default_domains: "api-manager.zup.desenv.bb.com.br,*.zup.desenv.bb.com.br"
cors_origins: "api-manager.domain.corp,*.domain.corp"
```

Configurações para envio de emails de Alarmes de SLA.

```
sla_alarm_cron: "30m"
sla_alarm_organizations:
sla_default_email_receiver: ""
```

Assinatura para atributos de contexto de End Points:

```
end_point_context_attributes_secret:
```

Configurações para o SSO do Fórum

```
discourse_sso_secret:
discourse_sso_login_url: "https://forum.domain.corp/session/sso_login"
```

Chave para sign in através do github

```
github_key:
github_secret:
```

Configuração do tempo de vida dos tokens de autenticação no developer portal (em segundos):

```
auth_token_batch_timeout: 10
```

5.6. zupme-user-manager-api

As configurações do **zupme-User-manager-api** feitas através de variáveis de ambiente. O arquivo

/<zupme-deploy-root>/customers/<customer>/<env>/zupme-user-manager-api/config/environments/config.yml contém todas as configurações, que são aplicadas em momento de *deploy*, por isso recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo que todas instâncias do componente tenham as alterações aplicadas.

Para listar as configurações em vigor:

```
$ sudo zupme-user-manager-api config
```

Para ver uma configuração específica:

```
$ sudo zupme-user-manager-api config:get CONFIG_NAME
```

Para modificar uma configuração específica:

```
$ sudo zupme-user-manager-api config:set CONFIG_NAME=NEW_VALUE
```

Chaves de configuração:

Porta a se fazer bind local

port: 6010

Slugs que não serão permitidos se criar (Configuração não relevante em ambientes *On Premises*)

slug_blacklist: eventhandler,apimanager,usermanager,gateway,ldap

Parametros de controle de falhas de login

max_login_failure_attempts: 10

blocking_period_after_max_failure_attempts: 900

login_failure_aware_interval: 600

Chaves utilizadas para assinar sessões

zupme_user_manager_private_key_path:

/opt/zupme-user-manager-api/certs/user_manager_session.key

auth_provider_settings_secret:

Chave pública utilizada para validar sessões

zupme_user_manager_public_key:

/opt/zupme-user-manager-api/certs/user_manager_session.pub

Tempo máximo de validade de sessão

zupme_session_expiration: 7

Configurações de número máximo de threads e conexões com o Banco de Dados Postgre

max_threads: 50

db_pool: 25

Chaves utilizadas para comunicação entre apis

zupme_api_communication_public_key_path:

/opt/zupme-user-manager-api/certs/zupme-ca.pub

zupme_ca_file: "/opt/zupme-user-manager-api/certs/zupme-ca.crt"

zupme_api_communication_client_application_certificate:

/opt/zupme-user-manager-api/certs/api_communication.crt

Domínios onde se é permitido CORS por padrão

cors_origins: "api-manager.domain.corp,*.domain.corp"

5.7. zupme-gateway

As configurações do **zupme-gateway** são feitas no arquivo **/<zupme-deploy-root>/customers/<customer>/<env>/zupme-gateway/conf/environments/config.json**. Recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo assim que todas instâncias do componente tenham as alterações aplicadas.

No momento do *deploy* o arquivo de configuração é copiado para **/opt/zupme-gateway/lib/config/<env>.json**. Caso seja feita alguma modificação diretamente no servidor é necessário reiniciar o serviço para que a configuração atualizada seja aplicada.

JSONSchema do arquivo de configuração:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "server": {
      "type": "object",
      "title": "Server schema.",
      "description": "Configurações das interfaces http do gateway.",
      "properties": {
        "gateway": {
          "type": "object",
          "title": "Gateway schema.",
          "description": "Configurações http do serviço do gateway.",
          "properties": {
            "name": {
              "type": "string",
              "description": "Nome interno do serviço."
            },
            "address": {
              "type": "string",
              "description": "Endereço de rede que onde o serviço irá escutar. Utilize 0.0.0.0 para escutar em qualquer endereço"
            },
            "port": {
              "type": "integer",
              "description": "Porta que o serviço irá escutar."
            },
            "listener": {
              "type": "object",
              "description": "Configurações do listener http.",
              "properties": {
                "tls": {
                  "type": "object",
                  "description": "Configurações de TLS.",
                  "properties": {
                    "key": {
                      "type": "string",
                      "description": "Caminho para a chave pública do certificado SSL."
                    },
                    "cert": {
                      "type": "string",

```

```

        "description": "Caminho para o do certificado SSL."
    },
    "ciphers": {
        "type": "string",
        "description": "Cifras suportadas."
    },
    "ecdhCurve": {
        "type": "string",
        "description": "Curva ecdh."
    },
    "honorCipherOrder": {
        "type": "boolean",
        "description": "Deve ou não honrar a ordem das cifras configuradas."
    }
},
"required": [
    "key",
    "cert"
]
},
"contexts": {
    "type": "array",
    "description": "Configuração de contextos SNI.",
    "items": {
        "type": "object",
        "description": "Array de contextos.",
        "properties": {
            "hostnameRegex": {
                "type": "string",
                "description": "Regex a aplicar no hostname da requisição."
            },
            "context": {
                "type": "object",
                "properties": {
                    "key": {
                        "type": "string",
                        "description": "Caminho para a chave pública do certificado SSL."
                    },
                    "cert": {
                        "type": "string",
                        "description": "Caminho para o do certificado SSL."
                    },
                    "ciphers": {
                        "type": "string",
                        "description": "Cifras suportadas."
                    },
                    "ecdhCurve": {
                        "type": "string",
                        "description": "Curva ecdh."
                    },
                    "honorCipherOrder": {
                        "type": "boolean",
                        "description": "Deve ou não honrar a ordem das cifras configuradas."
                    }
                }
            },
            "required": [
                "key",
                "cert"
            ]
        }
    },
    "required": [

```

```

        "hostnameRegex",
        "context"
    ]
    }
}
},
"required": [
    "tls"
]
},
"tls": {
    "type": "boolean",
    "description": "Ativa/Desativar TLS/SSL."
},
"routes": {
    "type": "object",
    "description": "Configurações da rota.",
    "properties": {
        "payload": {
            "type": "object",
            "properties": {
                "maxBytes": {
                    "description": "Tamanho máximo do payload."
                }
            }
        },
        "required": [
            "maxBytes"
        ]
    },
    "security": {
        "type": "object",
        "description": "Configurações de segurança.",
        "properties": {
            "hsts": {
                "type": "object",
                "description": "Configurações de HSTS (HTTP Strict Transport Security). Ativa o envio do header Strict-Transport-Security na resposta.",
                "properties": {
                    "maxAge": {
                        "type": "integer",
                        "description": "Valor da propriedade max-age no header Strict-Transport-Security."
                    },
                    "includeSubdomains": {
                        "type": "boolean",
                        "description": "Ativar/Desativar envio da propriedade includeSubDomains no header Strict-Transport-Security."
                    }
                }
            },
            "required": [
                "maxAge",
                "includeSubdomains"
            ]
        },
        "xframe": {
            "type": "boolean",
            "description": "Ativar/Desativar envio do header X-FRAME-OPTIONS com o valor SAMEORIGIN (https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options).",
        },
        "noSniff": {
            "type": "boolean",
            "title": "NoSniff schema.",

```

```

        "description": "Ativar/Desativar envio do header X-Content-Type-Options com o
valor nosniff (https://www.owasp.org/index.php/List\_of\_useful\_HTTP\_headers)"
    },
    "required": [
        "hsts",
        "xframe",
        "noSniff"
    ]
},
    "required": [
        "payload",
        "security"
    ]
},
    "proxyProtocolEnabled": {
        "type": "integer",
        "description": "Ativar/Desativar o uso de proxy protocol pelo load balancer que está
na frente do gateway."
    }
},
    "required": [
        "name",
        "address",
        "port",
        "listener",
        "tls",
        "routes",
        "proxyProtocolEnabled"
    ]
},
    "deploy": {
        "type": "object",
        "title": "Gateway schema.",
        "description": "Configurações http do serviço do gateway.",
        "properties": {
            "name": {
                "type": "string",
                "description": "Nome interno do serviço."
            },
            "address": {
                "type": "string",
                "description": "Endereço de rede que onde o serviço irá escutar. Utilize 0.0.0.0 para
escutar em qualquer endereço"
            },
            "port": {
                "type": "integer",
                "description": "Porta que o serviço irá escutar."
            },
            "listener": {
                "type": "object",
                "description": "Configurações do listener http.",
                "properties": {
                    "tls": {
                        "type": "object",
                        "description": "Configurações de TLS.",
                        "properties": {
                            "key": {
                                "type": "string",
                                "description": "Caminho para a chave pública do certificado SSL."
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    "cert": {
      "type": "string",
      "description": "Caminho para o do certificado SSL."
    },
    "ciphers": {
      "type": "string",
      "description": "Cifras suportadas."
    },
    "ecdhCurve": {
      "type": "string",
      "description": "Curva ecdh."
    },
    "honorCipherOrder": {
      "type": "boolean",
      "description": "Deve ou não honrar a ordem das cifras configuradas."
    }
  },
  "required": [
    "key",
    "cert"
  ]
},
"contexts": {
  "type": "array",
  "description": "Configuração de contextos SNI.",
  "items": {
    "type": "object",
    "description": "Array de contextos.",
    "properties": {
      "hostnameRegex": {
        "type": "string",
        "description": "Regex a aplicar no hostname da requisição."
      },
      "context": {
        "type": "object",
        "properties": {
          "key": {
            "type": "string",
            "description": "Caminho para a chave pública do certificado SSL."
          },
          "cert": {
            "type": "string",
            "description": "Caminho para o do certificado SSL."
          },
          "ciphers": {
            "type": "string",
            "description": "Cifras suportadas."
          },
          "ecdhCurve": {
            "type": "string",
            "description": "Curva ecdh."
          },
          "honorCipherOrder": {
            "type": "boolean",
            "description": "Deve ou não honrar a ordem das cifras configuradas."
          }
        }
      },
      "required": [
        "key",
        "cert"
      ]
    }
  ]
}

```



```

        },
        "required": [
            "hostnameRegex",
            "context"
        ]
    }
}
},
"required": [
    "tls"
]
},
"tls": {
    "type": "boolean",
    "description": "Ativa/Desativar TLS/SSL."
},
"routes": {
    "type": "object",
    "description": "Configurações da rota.",
    "properties": {
        "payload": {
            "type": "object",
            "properties": {
                "maxBytes": {
                    "description": "Tamanho máximo do payload."
                }
            }
        },
        "required": [
            "maxBytes"
        ]
    },
    "security": {
        "type": "object",
        "description": "Configurações de segurança.",
        "properties": {
            "hsts": {
                "type": "object",
                "description": "Configurações de HSTS (HTTP Strict Transport Security). Ativa o envio do header Strict-Transport-Security na resposta.",
                "properties": {
                    "maxAge": {
                        "type": "integer",
                        "description": "Valor da propriedade max-age no header Strict-Transport-Security."
                    },
                    "includeSubdomains": {
                        "type": "boolean",
                        "description": "Ativar/Desativar envio da propriedade includeSubDomains no header Strict-Transport-Security."
                    }
                }
            },
            "required": [
                "maxAge",
                "includeSubdomains"
            ]
        },
        "xframe": {
            "type": "boolean",
            "description": "Ativar/Desativar envio do header X-FRAME-OPTIONS com o valor SAMEORIGIN (https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options).
        },
        "noSniff": {

```

```

        "type": "boolean",
        "title": "NoSniff schema.",
        "description": "Ativar/Desativar envio do header X-Content-Type-Options com o
valor nosniff (https://www.owasp.org/index.php/List\_of\_useful\_HTTP\_headers)"
    }
},
    "required": [
        "hsts",
        "xframe",
        "noSniff"
    ]
},
    "required": [
        "payload",
        "security"
    ]
},
    "proxyProtocolEnabled": {
        "type": "integer",
        "description": "Ativar/Desativar o uso de proxy protocol pelo load balancer que está
na frente do gateway."
    }
},
    "required": [
        "name",
        "address",
        "port",
        "listener",
        "tls",
        "routes",
        "proxyProtocolEnabled"
    ]
},
    "health": {
        "type": "object",
        "title": "Gateway schema.",
        "description": "Configurações http do serviço do gateway.",
        "properties": {
            "name": {
                "type": "string",
                "description": "Nome interno do serviço."
            },
            "address": {
                "type": "string",
                "description": "Endereço de rede que onde o serviço irá escutar. Utilize 0.0.0.0 para
escutar em qualquer endereço"
            },
            "port": {
                "type": "integer",
                "description": "Porta que o serviço irá escutar."
            },
            "listener": {
                "type": "object",
                "description": "Configurações do listener http.",
                "properties": {
                    "tls": {
                        "type": "object",
                        "description": "Configurações de TLS.",
                        "properties": {
                            "key": {
                                "type": "string",

```

```

        "description": "Caminho para a chave pública do certificado SSL."
    },
    "cert": {
        "type": "string",
        "description": "Caminho para o do certificado SSL."
    },
    "ciphers": {
        "type": "string",
        "description": "Cifras suportadas."
    },
    "ecdhCurve": {
        "type": "string",
        "description": "Curva ecdh."
    },
    "honorCipherOrder": {
        "type": "boolean",
        "description": "Deve ou não honrar a ordem das cifras configuradas."
    }
},
"required": [
    "key",
    "cert"
]
},
"contexts": {
    "type": "array",
    "description": "Configuração de contextos SNI.",
    "items": {
        "type": "object",
        "description": "Array de contextos.",
        "properties": {
            "hostnameRegex": {
                "type": "string",
                "description": "Regex a aplicar no hostname da requisição."
            },
            "context": {
                "type": "object",
                "properties": {
                    "key": {
                        "type": "string",
                        "description": "Caminho para a chave pública do certificado SSL."
                    },
                    "cert": {
                        "type": "string",
                        "description": "Caminho para o do certificado SSL."
                    },
                    "ciphers": {
                        "type": "string",
                        "description": "Cifras suportadas."
                    },
                    "ecdhCurve": {
                        "type": "string",
                        "description": "Curva ecdh."
                    },
                    "honorCipherOrder": {
                        "type": "boolean",
                        "description": "Deve ou não honrar a ordem das cifras configuradas."
                    }
                }
            }
        }
    },
    "required": [
        "key",
        "cert"
    ]
}

```

```

        ]
      },
      "required": [
        "hostnameRegex",
        "context"
      ]
    }
  },
  },
  "required": [
    "tls"
  ]
},
"tls": {
  "type": "boolean",
  "description": "Ativa/Desativar TLS/SSL."
},
"routes": {
  "type": "object",
  "description": "Configurações da rota.",
  "properties": {
    "payload": {
      "type": "object",
      "properties": {
        "maxBytes": {
          "description": "Tamanho máximo do payload."
        }
      }
    },
    "required": [
      "maxBytes"
    ]
  },
},
"security": {
  "type": "object",
  "description": "Configurações de segurança.",
  "properties": {
    "hsts": {
      "type": "object",
      "description": "Configurações de HSTS (HTTP Strict Transport Security). Ativa o envio do header Strict-Transport-Security na resposta.",
      "properties": {
        "maxAge": {
          "type": "integer",
          "description": "Valor da propriedade max-age no header Strict-Transport-Security."
        },
        "includeSubdomains": {
          "type": "boolean",
          "description": "Ativar/Desativar envio da propriedade includeSubDomains no header Strict-Transport-Security."
        }
      }
    },
    "required": [
      "maxAge",
      "includeSubdomains"
    ]
  },
},
"xframe": {
  "type": "boolean",
  "description": "Ativar/Desativar envio do header X-FRAME-OPTIONS com o valor SAMEORIGIN (https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options)."
```

```

    },
    "noSniff": {
      "type": "boolean",
      "title": "NoSniff schema.",
      "description": "Ativar/Desativar envio do header X-Content-Type-Options com o
valor nosniff (https://www.owasp.org/index.php/List\_of\_useful\_HTTP\_headers)"
    }
  },
  "required": [
    "hsts",
    "xframe",
    "noSniff"
  ]
},
"required": [
  "payload",
  "security"
]
},
"proxyProtocolEnabled": {
  "type": "integer",
  "description": "Ativar/Desativar o uso de proxy protocol pelo load balancer que está
na frente do gateway."
}
},
"required": [
  "name",
  "address",
  "port",
  "listener",
  "tls",
  "routes",
  "proxyProtocolEnabled"
]
}
},
"required": [
  "gateway",
  "deploy",
  "health"
]
},
"couchbase": {
  "type": "object",
  "description": "Configurações do couchbase.",
  "properties": {
    "hosts": {
      "type": "string",
      "description": "Lista de hosts dos nós do cluster de Couchbase."
    },
    "apisBucket": {
      "type": "string",
      "description": "Nome do bucket de APIs."
    },
    "apiDocsBucket": {
      "type": "string",
      "description": "Nome do bucket de documentação de APIs."
    },
    "quotaBucket": {
      "type": "string",
      "description": "Nome do bucket de cotas."
    }
  }
}

```

```

    },
    "identityBucket": {
      "type": "string",
      "description": "Nome do bucket de endpoint identity."
    },
    "sessionsBucket": {
      "type": "string",
      "description": "Nome do bucket de sessão."
    },
    "tokensBucket": {
      "type": "string",
      "description": "Nome do bucket de tokens."
    },
    "developerPortalsBucket": {
      "type": "string",
      "description": "Nome do bucket de developer portal."
    },
    "cacheBucket": {
      "type": "string",
      "description": "Nome do bucket de cache."
    },
    "oAuthScopesBucket": {
      "type": "string",
      "description": "Nome do bucket de oauth scopes."
    },
    "designDocuments": {
      "type": "object",
      "description": "Configuração dos designDocuments para as views do couchbase.",
      "properties": {
        "api": {
          "type": "string",
          "description": "Nome do designDocument de api."
        },
        "apiDocs": {
          "type": "string",
          "description": "Nome do designDocument de apiDocs."
        }
      },
      "required": [
        "api",
        "apiDocs"
      ]
    },
    "views": {
      "type": "object",
      "description": "Configuração das views do couchbase.",
      "properties": {
        "byApiVersion": {
          "type": "string",
          "description": "Nome da view de busca por apiVersion."
        },
        "byClientApplicationKey": {
          "type": "string",
          "description": "Nome da view de busca por clientApplicationKey."
        },
        "byApiFullPath": {
          "type": "string",
          "description": "Nome da view de busca por caminho completa do api."
        }
      },
      "required": [
        "byApiVersion",

```

```

        "byClientApplicationKey",
        "byApiFullPath"
    ]
},
"timeouts": {
    "type": "object",
    "properties": {
        "operation": {
            "type": "integer",
            "description": "Timeout de operações."
        },
        "connection": {
            "type": "integer",
            "description": "Timeout de conexão."
        }
    },
    "required": [
        "operation",
        "connection"
    ]
},
"required": [
    "hosts",
    "apisBucket",
    "apiDocsBucket",
    "quotaBucket",
    "identityBucket",
    "sessionsBucket",
    "tokensBucket",
    "developerPortalsBucket",
    "cacheBucket",
    "oAuthScopesBucket",
    "designDocuments",
    "views",
    "timeouts"
]
},
"logger": {
    "type": "object",
    "title": "Logger schema.",
    "description": "An explanation about the purpose of this instance.",
    "properties": {
        "gateway": {
            "type": "object",
            "properties": {
                "consoleEnabled": {
                    "type": "boolean",
                    "description": "Ativar/Desativar log na saída padrão."
                },
                "appendTimestampToConsole": {
                    "type": "boolean",
                    "description": "Ativar/Desativar timestamp na saída padrão."
                },
                "fileEnabled": {
                    "type": "boolean",
                    "description": "Ativar/Desativar log em arquivo."
                },
                "file": {
                    "type": "string",
                    "description": "Caminho para o arquivo de log."
                }
            }
        }
    }
},

```

```

    "maxFileSizeInBytes": {
      "type": "integer",
      "description": "Tamanho máximo do arquivo de log."
    },
    "maxFiles": {
      "type": "integer",
      "description": "Número máximo de arquivos de log."
    },
    "gelfEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar log no formato gelf (Graylog)."
    },
    "gelfHost": {
      "type": "string",
      "description": "Host do servidor Graylog."
    },
    "gelfPort": {
      "type": "string",
      "description": "Porta do servidor Graylog."
    },
    "gelfFacility": {
      "type": "string",
      "description": "Nome da facility a ser enviada para o Graylog."
    }
  },
  "required": []
},
"eventProcessor": {
  "type": "object",
  "properties": {
    "consoleEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar log na saída padrão."
    },
    "appendTimestampToConsole": {
      "type": "boolean",
      "description": "Ativar/Desativar timestamp na saída padrão."
    },
    "fileEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar log em arquivo."
    },
    "file": {
      "type": "string",
      "description": "Caminho para o arquivo de log."
    },
    "maxFileSizeInBytes": {
      "type": "integer",
      "description": "Tamanho máximo do arquivo de log."
    },
    "maxFiles": {
      "type": "integer",
      "description": "Número máximo de arquivos de log."
    },
    "gelfEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar log no formato gelf (Graylog)."
    },
    "gelfHost": {
      "type": "string",
      "description": "Host do servidor Graylog."
    }
  },

```



```

        "gelfPort": {
            "type": "string",
            "description": "Porta do servidor Graylog."
        },
        "gelfFacility": {
            "type": "string",
            "description": "Nome da facility a ser enviada para o Graylog."
        }
    },
    "required": []
},
"required": [
    "gateway",
    "eventProcessor"
]
},
"ipc": {
    "type": "object",
    "description": "Configurações de comunicação com o event-broker.",
    "properties": {
        "pushSocketFile": {
            "type": "string",
            "description": "Caminho para o socket de entrada."
        },
        "pullSocketFile": {
            "type": "string",
            "description": "Caminho para o socket de saída."
        }
    },
    "required": [
        "pushSocketFile",
        "pullSocketFile"
    ]
},
"eventProcessor": {
    "type": "object",
    "description": "Configurações do event-processor.",
    "properties": {
        "host": {
            "type": "string",
            "description": "Host do event-handler."
        },
        "port": {
            "type": "integer",
            "description": "Porta em que o event-handler está escutando os eventos."
        },
        "deploymentPort": {
            "type": "integer",
            "description": "Porta em que o event-handler está escutando os deploys."
        },
        "key": {
            "type": "string",
            "description": "Caminho para a chave pública do certificado SSL."
        },
        "cert": {
            "type": "string",
            "description": "Caminho para o do certificado SSL."
        },
        "ca": {
            "type": "string",
            "description": "Caminho para o ca, utilizado para validar o 2-way-ssl."
        }
    }
}

```

```

    },
    "maxSockets": {
      "type": "integer",
      "description": "Numero máximos de sockets na comunicação com o event-handler."
    },
    "postTimeout": {
      "type": "integer",
      "description": "Timeout da comunicação com o event-handler."
    }
  },
  "required": [
    "host",
    "port",
    "deploymentPort",
    "key",
    "cert",
    "ca",
    "maxSockets",
    "postTimeout"
  ]
},
"gateway": {
  "type": "object",
  "description": "Configurações do endpoint.",
  "properties": {
    "defaultOrganizationSlug": {
      "type": "string",
      "description": "Slug da organização do endpoint. No caso de um endpoint que sirva a
multiplas organizações, usar false."
    },
    "eventsEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar envio de eventos."
    },
    "timeout": {
      "type": "integer",
      "description": "Timeout da conexão com o Endpoint."
    },
    "endpointConnectionKeepAlive": {
      "type": "boolean",
      "description": "Ativar/Desativar keepalive na conexão com o Endpoint."
    },
    "cors": {
      "type": "object",
      "description": "Configurações de cors.",
      "properties": {
        "requestHeaders": {
          "type": "array",
          "description": "Headers de request permitidos em CORS.",
          "items": {
            "type": "string"
          }
        },
        "exposeHeaders": {
          "type": "array",
          "description": "Headers de response permitidos em CORS.",
          "items": {
            "type": "string"
          }
        }
      }
    }
  },
  "required": [

```

```

        "requestHeaders",
        "exposeHeaders"
    ]
}
},
"required": [
    "defaultOrganizationSlug",
    "eventsEnabled",
    "timeout",
    "endpointConnectionKeepAlive",
    "cors"
]
},
"naught": {
    "type": "object",
    "description": "Configurações do naught process manager",
    "properties": {
        "exitTimeoutSeconds": {
            "type": "integer",
            "description": "Timeot de terminação do serviço."
        }
    }
},
"required": [
    "exitTimeoutSeconds"
]
},
"cache": {
    "type": "object",
    "description": "Configurações de cache in-memory.",
    "properties": {
        "api": {
            "type": "object",
            "properties": {
                "enabled": {
                    "type": "boolean",
                    "description": "Ativar/Desativar cache de APIs."
                }
            },
            "max": {
                "type": "integer",
                "description": "Número maximo de entradas no cache de APIs."
            },
            "maxAge": {
                "type": "integer",
                "description": "Tempo de expiração das entradas no cache de APIs."
            }
        },
        "required": [
            "enabled",
            "max",
            "maxAge"
        ]
    },
    "identity": {
        "type": "object",
        "description": "Configurações de cache de identidade do entrypoint.",
        "properties": {
            "max": {
                "type": "integer",
                "description": "Número maximo de entradas no cache de identidade do entrypoint."
            },
            "maxAge": {
                "type": "integer",
            }
        }
    }
}

```

```

        "description": "Tempo de expiração das entradas no cache de identidade do
entrypoint."
    },
    "required": [
        "max",
        "maxAge"
    ]
},
"required": [
    "api",
    "identity"
]
},
"security": {
    "type": "object",
    "description": "Configurações de segurança interna.",
    "properties": {
        "tokenCipherSecretKey": {
            "type": "string",
            "description": "Secret da chave de criptografia de tokens."
        },
        "contextAttributesSecretKey": {
            "type": "string",
            "description": "Secret da chave de criptografia de atributos."
        }
    },
    "required": [
        "tokenCipherSecretKey",
        "contextAttributesSecretKey"
    ]
},
"required": [
    "server",
    "couchbase",
    "logger",
    "ipc",
    "eventProcessor",
    "gateway",
    "naught",
    "cache",
    "security"
]
}

```

5.8. zupme-event-handler

As configurações do **zupme-event-handler** são feitas no arquivo **/<zupme-deploy-root>/customers/<customer>/<env>/zupme-event-handler/conf/environments/config.json**. Recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo assim que todas instâncias do componente tenham as alterações aplicadas.

No momento do *deploy* o arquivo de configuração é copiado para **/opt/cockpit-event-handler/lib/config/<env>.json**. Caso seja feita alguma modificação

diretamente no servidor é necessário reiniciar o serviço para que a configuração atualizada seja aplicada.

JSONSchema do arquivo de configuração:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "server": {
      "type": "object",
      "description": "Configurações das interfaces http do event-handler.",
      "properties": {
        "handler": {
          "type": "object",
          "description": "Configurações http do serviço que recebe os eventos do gateway.",
          "properties": {
            "name": {
              "type": "string",
              "description": "Nome interno do serviço."
            },
            "address": {
              "type": "string",
              "description": "Endereço de rede que onde o serviço irá escutar. Utilize 0.0.0.0 para escutar em qualquer endereço"
            },
            "port": {
              "type": "integer",
              "description": "Porta que o serviço irá escutar."
            },
            "key": {
              "type": "string",
              "description": "Caminho para a chave pública do certificado SSL."
            },
            "cert": {
              "type": "string",
              "description": "Caminho para o do certificado SSL."
            },
            "ca": {
              "type": "string",
              "description": "Caminho para o ca, utilizado para validar o 2-way-ssl."
            },
            "maxPayloadSizeInBytes": {
              "type": "integer",
              "description": "Tamanho máximo do payload http."
            }
          }
        },
        "required": [
          "name",
          "address",
          "port",
          "key",
          "cert",
          "ca",
          "maxPayloadSizeInBytes"
        ]
      },
      "health": {
        "type": "object",
        "description": "Configurações http do serviço de health-check.",

```

```

    "properties": {
      "name": {
        "type": "string",
        "description": "Nome interno do serviço."
      },
      "address": {
        "type": "string",
        "description": "Endereço de rede que onde o serviço irá escutar. Utilize 0.0.0.0 para
escutar em qualquer endereço"
      },
      "port": {
        "type": "integer",
        "description": "Porta que o serviço irá escutar."
      },
    },
    "required": [
      "name",
      "address",
      "port"
    ]
  },
  "deploy": {
    "type": "object",
    "description": "Configurações http do serviço de deploy de apis.",
    "properties": {
      "name": {
        "type": "string",
        "description": "Nome interno do serviço."
      },
      "address": {
        "type": "string",
        "description": "Endereço de rede que onde o serviço irá escutar. Utilize 0.0.0.0 para
escutar em qualquer endereço"
      },
      "port": {
        "type": "integer",
        "description": "Porta que o serviço irá escutar."
      },
    },
    "required": [
      "name",
      "address",
      "port"
    ]
  },
  "required": [
    "handler",
    "health",
    "deploy"
  ]
},
"logger": {
  "type": "object",
  "description": "Configurações de log.",
  "properties": {
    "eventHandler": {
      "type": "object",
      "properties": {
        "consoleEnabled": {
          "type": "boolean",
          "description": "Ativar/Desativar log na saída padrão."
        }
      }
    }
  }
}

```

```

    },
    "fileEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar log em arquivo."
    },
    "file": {
      "type": "string",
      "description": "Caminho para o arquivo de log."
    },
    "maxFileSizeInBytes": {
      "type": "integer",
      "description": "Tamanho máximo do arquivo de log."
    },
    "maxFiles": {
      "type": "integer",
      "description": "Número máximo de arquivos de log."
    },
    "gelfEnabled": {
      "type": "boolean",
      "description": "Ativar/Desativar log no formato gelf (Graylog)."
    },
    "gelfHost": {
      "type": "string",
      "description": "Host do servidor Graylog."
    },
    "gelfPort": {
      "type": "string",
      "description": "Porta do servidor Graylog."
    },
    "gelfFacility": {
      "type": "string",
      "description": "Nome da facility a ser enviada para o Graylog."
    }
  },
  "required": []
}
},
"required": [
  "eventHandler"
]
},
"ipc": {
  "type": "object",
  "description": "Configurações de comunicação com o event-broker.",
  "properties": {
    "pushSocketFile": {
      "type": "string",
      "description": "Caminho para o socket de entrada."
    },
    "pullSocketFile": {
      "type": "string",
      "description": "Caminho para o socket de saída."
    }
  },
  "required": [
    "pushSocketFile",
    "pullSocketFile"
  ]
},
"naught": {
  "type": "object",
  "description": "Configurações do naught process manager",

```

```

    "properties": {
      "exitTimeoutSeconds": {
        "type": "integer",
        "description": "Timeot de terminação do serviço."
      }
    },
    "required": [
      "exitTimeoutSeconds"
    ]
  },
  "couchbase": {
    "type": "object",
    "description": "Configurações do couchbase.",
    "properties": {
      "hosts": {
        "type": "string",
        "description": "Lista de hosts dos nós do cluster de Couchbase."
      },
      "actionsBucket": {
        "type": "string",
        "description": "Nome do bucket de actions."
      },
      "timeouts": {
        "type": "object",
        "properties": {
          "operation": {
            "type": "integer",
            "description": "Timeout de operações."
          },
          "connection": {
            "type": "integer",
            "description": "Timeout de conexão."
          }
        },
        "required": [
          "operation",
          "connection"
        ]
      },
      "required": [
        "hosts",
        "actionsBucket",
        "timeouts"
      ]
    },
    "required": [
      "server",
      "logger",
      "ipc",
      "naught",
      "couchbase"
    ]
  }
}

```


5.9. Logstash

As configurações do **logstash** são feitas no arquivo

/<zupme-deploy-root>/customers/<customer>/<env>/logstash/conf/logstash.conf.

Recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo assim que todas instâncias do componente tenham as alterações aplicadas.

As duas configurações relevantes que podem ser alteradas nesse arquivo de configuração são os *hosts* dos *cluster* do Couchbase e do Elasticsearch que se encontram nas propriedades `dynamic_couchbase.cluster` e `elasticsearch.host` respectivamente. No caso do Couchbase, todos os nós do *cluster* devem ser informados. No caso do Elasticsearch, todos os nós que contarem com a configuração de `http_enabled=true` (vide inventário do componente Elasticsearch).

5.10. PostgreSQL

As configurações do **postgre** são feitas no arquivo **\$PG_DATA/postgresql.conf** e **\$PG_DATA/pg_hba.conf**. Recomenda-se a leitura da documentação oficial do componente para detalhamento dos parâmetros de configuração (<https://www.postgresql.org/docs/9.5/static/runtime-config.html>).

5.11. Couchbase

As configurações do **couchbase** são feitas no arquivo

/<zupme-deploy-root>/customers/<customer>/<env>/couchbase/inv/hosts.

Recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo assim que todas instâncias do componente tenham as alterações aplicadas.

```
# Memória total do Couchbase por Host
couchbase_server_ram=
# Porta de administração do Couchbase
couchbase_server_admin_port=
# Tamanho em memória do bucket do tipo small
couchbase_server_bucket_ram_small=
# Tamanho em memória do bucket do tipo medium
couchbase_server_bucket_ram_medium=
# Tamanho em memória do bucket do tipo large
couchbase_server_bucket_ram_large=
# Número de réplicas por bucket
couchbase_server_bucket_number_of_replica=
# Permitir a deleção manual de dados dos buckets
```

```
couchbase_server_bucket_flush=  
# Memória RAM dedicada a operações de Indexação  
couchbase_server_index_ram=
```

5.12. Elasticsearch

As configurações do **elasticsearch** são feitas no arquivo
/<zupme-deploy-root>/customers/<customer>/<env>/elasticsearch/inv/hosts.

Recomenda-se que qualquer alteração necessária seja feita nesse arquivo, garantindo assim que todas instâncias do componente tenham as alterações aplicadas.

```
# Hosts que serão utilizados como master_nodes  
elasticsearch_unicast_host=  
# Número de nodes que são esperados que estejam no cluster  
expected_nodes=  
# Número de nodes esperados para iniciar o processo de recovery, caso necessário  
recover_after_nodes=  
# Número mínimo de master_nodes para que o cluster fique operacional, recomenda-se  
# mínimo 3 para evitar situações de split-brain  
minimum_master_nodes=  
# Memória RAM dedicada ao processo do elasticsearch, recomenda-se que seja até no  
# máximo 50% da memória total da máquina.  
elasticsearch_heap_size=  
# True caso host seja data_node  
data_node=  
# True caso host seja master_node  
master_node=  
# True caso host deva receber requisições HTTP na porta 9200  
http_enabled=
```

6. Monitoramento

6.1. Api Manager

6.1.1. Portas

- Nginx: Portas 80 e 443
- Api Manager API: Porta 6000
- User Manager API: Porta 6010

6.1.2. Processos

- Puma (zupme-api-manager-api)
- Puma (zupme-user-manager-api)
- Nginx

6.1.3. Jobs

- developer_portal_deployment
- organization_dirty_check
- es_index
- organization_destroy
- auto_deployment
- mailers

6.1.4. Logs

6.1.4.1. api-manager-api e user-manager-api

O alarme é disparado quando chegam mais de 10 mensagens que contenham a palavra "error" no seu texto nos últimos 60 minutos. O tempo entre um alarme e outro é de 60 minutos.

6.1.4.2. nginx

O alarme é disparado quando chegam mais de 10 eventos no log de erro do nginx nos últimos 60 minutos. O tempo entre um alarme e outro é de 10 minutos.

6.2. PostgreSQL

6.2.1. Portas

- 5432

6.2.2. Processos

- postgres

6.2.3. Data Volumes

- /data/pqsql

6.3. Gateway

6.3.1. Portas

- Gateway: Portas 443 e 8443
- Health Check: 8020

6.3.2. Processos

- Node (zupme-gateway.js)
- Node (event-processor.js)

6.3.3. Health Check

- <http://localhost:8020/health-check>

6.3.4. Logs

O alarme é disparado quando chegam mais de 10 mensagens que tenham o início parecido com "2016-09-29T14:18:39.576Z - error:" (o que muda é só a data) nos últimos 60 minutos. O tempo entre um alarme e outro é de 60 minutos.

6.4. Couchbase

6.4.1. Portas

- Node to node: 8091, 8092, 11209, 11210, 4369, 21100-21299
- Node to client: 8091, 8092, 11207, 11210, 11211
- Admin: 8091

6.4.2. Processos

- beam.smp

6.4.3. Logs

Os logs verificados ficam em `"/opt/couchbase/var/lib/couchbase/logs"` com a seguinte nomenclatura:

`{couchdb,info,http_access,error,xcdr_error,xdc,tmppfail,mapreduce_errors,views,stats,memcached,reports}.log,bin`

Cada mensagem é definida pelo padrão `"^\\[. *?\\.: *?\\,\\d{4}-\\d{2}-\\d{2}T"`

O alarme é disparado quando chegam mais de 10 mensagens com o padrão `"(?i)error|emergency|alert|critical"` nos últimos 60 minutos. O tempo entre um alarme e outro é de 60 minutos.

6.5. Event Handler

6.5.1. Portas

- Event Handler: Portas 8005 e 8009
- ZMQ: 127.0.0.1:8006
- Health Check: 8008

6.5.2. Processos

- `lojs (cockpit-event-handler.js)`
- `lojs (event-broker.js)`
- Java (`logstash`)

6.5.3. Health Check

- `http://localhost:8008/health-check`

6.5.4. Logs

O alarme é disparado quando chegam mais de 10 mensagens que o início tenha o padrão `"[Ee]rror:"` nos últimos 60 minutos. O tempo entre um alarme e outro é de 60 minutos.

6.6. ElasticSearch

6.6.1. Portas

- Client Nodes: 9200 e 9300
- Master e Data Nodes: 9300

6.6.2. Data Volumes

- /data/elasticsearch

6.6.3. Processos

- Java (elasticsearch)

6.6.4. Cluster Status

- `curl -XGET 'http://localhost:9200/_cluster/health?pretty=true'`

6.6.5. Logs

Os logs verificados ficam em:

/var/log/elasticsearch/zupme.log

/var/log/elasticsearch/zupme_index_indexing_slowlog.log

/var/log/elasticsearch/zupme_index_search_slowlog.log

Cada mensagem é definida pelo padrão "`^\[\d{4}-\d{2}-\d{2}`"

O alarme é disparado quando chegam mais de 10 mensagens com o padrão `(?i)error|emergency|alert|critical` nos últimos 60 minutos. O tempo entre um alarme e outro é de 60 minutos.

7. Performance e tuning

A ferramenta zupme-deploy já contempla parâmetros de performance e tuning de todos os componentes instalados pela mesma, caso sejam necessárias configurações específicas para o ambiente, deve-se adaptar o número de workers e memória utilizada para cada componente de acordo com o hardware utilizado, dentro do inventário dos mesmos.

Para componentes de terceiros, recomenda-se a leitura dos documentos relacionados a performance de cada um.

Elasticsearch:

<https://www.elastic.co/guide/en/elasticsearch/guide/1.x/indexing-performance.html>

Couchbase:

<http://www.couchbase.com/nosql-resources/presentations/tuning-couchbase-server-the-os-and-the-network-for-maximum-performance.html>

PostgreSQL:

https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server

8. Escalabilidade

8.1. Componentes Zup.me

Para os componentes proprietários da solução, sendo eles zupme-gateway, zupme-event-handler, zupme-api-manager, zupme-user-manager, zupme-connector-bb e zupme-connector-recorder, a escala horizontal deve ser feita da seguinte forma:

1. Introdução dos novos hosts a serem adicionados no inventário de cada componente dentro do zupme-deploy em customers/<customer>/<ambiente>/<componente>/inv/hosts.
2. Provisionamento dos novos hosts através do comando:
`$ bin/zupme provision --module=<componente> --customer=<customer> --env=<ambiente> --limit novos_host`
3. Deploy do componente nos hosts recém provisionados através do comando:
`$ bin/zupme deploy --module=<componente> --version=<versao_utilizada> --customer=<customer> --env=<ambiente> --limit novos_host`

A retirada dos componentes pode ser feita através da remoção do componente do balancer ou VIP e shutdown do Sistema Operacional.

8.2. Componentes de terceiros

8.2.1. Elasticsearch

A inclusão de novos hosts deve ser feita através da ferramenta zupme-deploy, da seguinte forma:

1. Introdução dos novos hosts a serem adicionados no inventário do elasticsearch do zupme-deploy em `customers/<customer>/<ambiente>/elasticsearch/inv/hosts`, dentro do grupo de hosts ao qual deve ser adicionado (`data_nodes`, `master_nodes` ou `client_nodes`)
2. Ainda no inventário, Incluir o host na lista de `elasticsearch_unicast_host`, caso o mesmo seja um master node.
3. Provisionamento dos novos hosts através do comando:
`$ bin/zupme provision --module=elasticsearch --customer=<customer> --env=<ambiente> --limit ново_host`

A retirada de componentes do tipo data node deve ser feita da seguinte forma:

1. Descomissionamento do host através do comando:

```
curl -XPUT 127.0.0.1:9200/_cluster/settings -d '{ "transient" :{  
  "cluster.routing.allocation.exclude._ip" : "host_a_ser_removido" } }'
```

2. Após a realocação dos dados no host a ser removido, é seguro desligar a instância.

A retirada de master nodes deve ser realizada apenas caso houver alguma falha no componente que seja entendida como irrecoverável. Caso seja necessário, assegurar que há pelo menos o número mínimo de master nodes estipulado no inventário na chave `minimum_master_nodes` antes de remover algum master node.

A remoção pode ser feita a qualquer momento desligando a instância onde o mesmo está sendo executado.

8.2.2. Couchbase

A introdução de novos hosts deve ser feita da seguinte forma:

1. Adicionar novos hosts no arquivo de inventário do componente em `<customer>/<environment>/couchbase/inv/hosts`.
2. Provisionar o novo host através do seguinte comando:

```
$ ansible-playbook --ask-pass --ask-sudo-pass -u <user>  
ansible/couchbase/provision-cluster.yml -i  
customers/<customer>/<env>/couchbase/inv/hosts --extra-vars  
"ansible_distribution=RedHat" --tags "installation" --limit=ново_host
```


3. Executar o seguinte comando para adicionar um novo host:

```
$ couchbase-cli rebalance -c host_existente:8091 --server-add=novo_host:8091  
--server-add-username=Administrator --server-add-password=password
```

A remoção de nodes pode ser feita executando o comando abaixo:

```
$ couchbase-cli rebalance --cluster=host_existente:8091 -u Administrator -p  
Password --server-remove=host_a_ser_removido
```

8.2.3. PostgreSQL

Devido à natureza *Design Time* do componente no ecossistema da solução, escalabilidade horizontal não se faz necessária nesse componente.

9. Referencias

Lista de comandos por componentes

