Ferramenta tecnológica para implantação do Balcão Virtual

Introdução

Em atenção à necessidade de implantação do denominado “Balcão Virtual” no prazo estabelecido em resolução aprovada pelo Plenário do CNJ na sessão do dia 9 de fevereiro do corrente ano, é preciso que os Tribunais disponibilizem em seu sítio eletrônico, ferramenta de videoconferência que permita imediato contato com o setor de atendimento de cada unidade judiciária, popularmente denominado como “balcão” durante o horário de atendimento ao público.

Em razão do acelerado processo de transformação digital ocorrido no último ano, e das Resoluções pretéritas editadas pelo CNJ, percebe-se que diversos Tribunais já possuem, dentre o rol de serviços providos, ferramenta de videoconferência para realização de reuniões, audiências e sessões de julgamento, que poderão atender às necessidades oriundas da disponibilização do Balcão Virtual.

Para os Tribunais que não possuem solução própria ou contratada para atendimento a essa demanda, ou para aqueles que possuem limitações de licenças, o CNJ, conforme a mesma resolução, sugere a adoção de solução gratuita, para imediata criação do Balcão Virtual.

Dessa forma, disponibiliza-se abaixo um manual de instalação da ferramenta Jitsi Meet (https://meet.jit.si/) na infraestrutura do Tribunal. Tal aplicação, em avaliação técnica e negocial, atende aos requisitos necessários para a criação do Balcão Virtual em cada Tribunal.

Saiba mais sobre o Jitsi Meet

Jitsi Meet é um aplicativo de videoconferência e mensagens instantâneas de código aberto para a plataforma da web, Windows, Linux, macOS, iOS e Android. Além disso, é baseado principalmente em WebRTC (Web Real-Time Communication). Ele fornece salas de videoconferência para várias pessoas, que você pode acessar usando seu navegador. Além disso, oferece funcionalidade comparável a uma chamada de conferência Zoom ou Skype.

Instalação da ferramenta Jitsi.me

1. Configurando o nome do host do sistema

Primeiro, precisamos alterar o nome do host do sistema para corresponder ao nome de domínio que pretendemos usar para a instância Jitsi Meet. Em seguida, resolva esse nome de host para o IP do host local, 127.0.0.1.

Execute o comando abaixo para definir o nome do host atual e modificar o / etc / hostname que conterá o nome do host do sistema entre as reinicializações.

```
$ sudo hostnamectl set-hostname jitsi.seu_dominio
```

Confirme se foi bem sucedido ou não executando o comando abaixo. Ele retornará o nome do host definido com o comando hostnamectl.

```
$ hostname
```

Em seguida, defina um mapeamento local do nome de host do servidor para o endereço IP de loopback, 127.0.0.1. Você pode fazer isso abrindo o / etc / hosts com um editor de texto:

```
$ sudo nano /etc/hosts
```

Adicione a linha abaixo ao arquivo.

```
127.0.0.1 jitsi.seu_dominio
```

O mapeamento local do nome de domínio do servidor Jitsi Meet para 127.0.0.1 é necessário porque seu servidor usa vários processos em rede no servidor que aceitam conexões locais no endereço IP 127.0.0.1.

2. Configurando o Firewall

Já habilitamos o firewall UFW. Agora o servidor Jitsi precisa de algumas portas abertas para que possa se comunicar com os clientes de chamada. As portas que precisamos abrir são 80/tcp, 443/tcp, 4443/tcp 10000/udp.

Execute os comandos abaixo para abrir as portas.

```
$ sudo ufw allow 80 / tcp
$ sudo ufw allow 443 / tcp
$ sudo ufw allow 4443 / tcp
$ sudo ufw allow 10000 / udp
```

Você pode verificar se eles foram adicionados ou não executando o comando abaixo.

```
$ sudo ufw status
```

3. Instale o Jitsi Meet

Primeiro, baixamos a chave Jitsi GPG com o utilitário de download wget.

```
$ wget https://download.jitsi.org/jitsi-key.gpg.key
```

Em seguida, adicione a chave GPG ao chaveiro do apt usando o utilitário apt-key.

```
$ sudo apt-key add jitsi-key.gpg.key
```

Agora exclua o arquivo de chave GPG, pois ele não é mais necessário. Para isso, execute o comando abaixo.

```
$ rm jitsi-key.gpg.key
```

Em seguida, adicione o repositório Jitsi ao servidor criando um novo arquivo de origem que contém o repositório Jitsi. Portanto, abra e crie um novo arquivo.

```
$ sudo nano /etc/apt/sources.list.d/jitsi-stable.list
```

Agora adicione a linha abaixo no arquivo para o repositório Jitsi.

```
deb https://download.jitsi.org stable /
```

Em seguida, salve o arquivo e saia do editor.

Finalmente, atualize o sistema para coletar a lista de pacotes do repositório Jitsi e, em seguida, instale o pacote jitsi-meet.

```
$ sudo apt update

$ sudo apt install jitsi-meet
```

Ao instalar o Jitsi Meet, você será solicitado a inserir o nome de domínio que deseja usar para sua instância Jitsi Meet.

Em seguida, você será solicitado a criar e usar um certificado TLS autoassinado ou usar um existente, se tiver um. Nesse caso, se você não tiver um certificado TLS para seu domínio Jitsi, selecione a opção ‘Gerar um novo certificado autoassinado’.

Agora, a instância Jitsi Meet está instalada usando um certificado TLS autoassinado.

4. Obtenção de um certificado TLS assinado

Para baixar automaticamente um certificado TLS para o domínio, o Jitsi Meet fornece um script. Execute o comando abaixo para executar o script de instalação.

```
$ sudo /usr/share/jitsi-meet/scripts/install-letsencrypt-cert.sh
```

Este script pedirá um endereço de e-mail. Insira o endereço de e-mail e ele será enviado ao emissor do certificado https://letsencrypt.org e será usado para notificar sobre questões de segurança.

5. Bloqueando a criação da conferência

Primeiro, abra o arquivo:

```
/etc/prosody/conf.avail/jitsi.seu_dominio.cfg.lua
```

```
$ sudo nano /etc/prosody/conf.avail/seu_dominio.cfg.lua
```

Em seguida, edite a linha abaixo.

```
autenticação = "anônimo"
```

Para

```
autenticação = "internal_plain"
```

Isso forçará o Jitsi Meet a forçar a autenticação do nome de usuário e senha antes de permitir a criação de uma sala de conferência por um novo visitante.

Então, no mesmo arquivo, adicione as linhas abaixo na parte inferior do arquivo.

```
VirtualHost "guest.jitsi.your_domain"

autenticação = "anônimo"

c2s_require_encryption = false
```

Esta configuração permitirá que usuários anônimos ingressem em salas de conferência criadas por um usuário autenticado. Porém, para entrar no quarto, o hóspede deve ter um endereço único e uma senha opcional.

Agora abra outro arquivo de configuração em:

```
 /etc/jitsi/meet/jitsi.your_domain-config.js

$ sudo nano /etc/jitsi/meet/jitsi.your_domain-config.js
```

Em seguida, edite esta linha:

```
// domínio anônimo: 'guest.jitsi.seu_domínio',
```

Para

```
domínio anônimo: 'guest.jitsi.seu_domínio',
```

Em seguida, abra /etc/jitsi/jicofo/sip-communicator.properties

```
$ sudo nano /etc/jitsi/jicofo/sip-communicator.properties
```

Em seguida, adicione a linha abaixo para concluir as alterações de configuração.

```
org.jitsi.jicofo.auth.URL = XMPP: jitsi.seu_dominio
```

Esta configuração aponta um dos processos Jitsi Meet para o servidor local que executa a autenticação do usuário que agora é necessária.Agora a configuração do Jitsi Meet está concluída. Então, agora precisamos registrar os usuários e suas senhas.

Aqui está o comando para adicionar um usuário ao servidor.

```
$ sudo prosodyctl registrar usuário sua senha de domínio
```

Este não é um usuário do sistema. Em vez disso, eles só poderão criar uma sala de conferências. Por último, reinicie os processos do Jitsi Meet para carregar a nova configuração.

```
$ sudo systemctl restart prosody.service
$ sudo systemctl restart jicofo.service
$ sudo systemctl restart jitsi-videobridge2.service
```

Finalmente, agora o servidor Jitsi Meet está configurado e configurado com segurança.

Documentação de uso

A própria comunidade de desenvolvimento da plataforma mantém documentação atualizada para configuração e uso do sistema, em seu sítio eletrônico (https://jitsi.github.io/handbook/docs/user-guide/user-guide-start).