

Universidade Federal de São Carlos - Sorocaba

# Manual de Uso da OCD - OpenStack Connectivity Deployer

André Luiz B. Rocha  
Lucio A. Rocha  
Fabio L. Verdi

Sorocaba - SP  
2015

# Sumário

- Lista de Figuras ..... 3
- 1. Requisitos para executar a OCD ..... 4
- 2. Utilização da OCD ..... 4
- 3. Cenário Exemplo..... 7
- 4. Regras OpenFlow ..... 9

# Lista de Figuras

Figura 1: Tela inicial.....	4
Figura 2: Tela de senhas. ....	4
Figura 3: Arquivo de gerência. ....	5
Figura 4: Criação de Redes.....	5
Figura 5: Quantidade de sub-redes. ....	5
Figura 6: IP da sub-rede.....	6
Figura 7: Criação das instâncias de MVs. ....	6
Figura 8: Início da configuração.....	6
Figura 9: Cenário Exemplo.....	7
Figura 10: Regras OpenFlow utilizadas. ....	9

# 1. Requisitos para executar a OCD

Os seguintes requisitos são necessários para executar a OCD:

- 1) OpenStack instalado e rodando em uma MV.
- 2) Duas interfaces de rede, uma para o plano de dados outra para o plano de controle. Tanto nas MVs quanto nos hosts físicos.
- 3) JDK do Java SE versão 1.7.0 ou superior instalado no host físico.
- 4) Download do arquivo *OCD.zip*. Disponível em: <https://github.com/andrebeltrami/OpenStack-Connectivity-Deployer>.
- 5) Arquivo *gerencia.txt* configurado para cada host físico. Exemplo de arquivo disponível em: <https://github.com/andrebeltrami/OpenStack-Connectivity-Deployer>.
- 6) Executar o *OCD.jar* disponível no *OCD.zip* obtido no passo 4. O comando para executar a ferramenta é dado por: `#java -jar OCD.jar`

## 2. Utilização da OCD

Adiante ilustraremos cada uma das telas da ferramenta OCD e suas respectivas entradas.

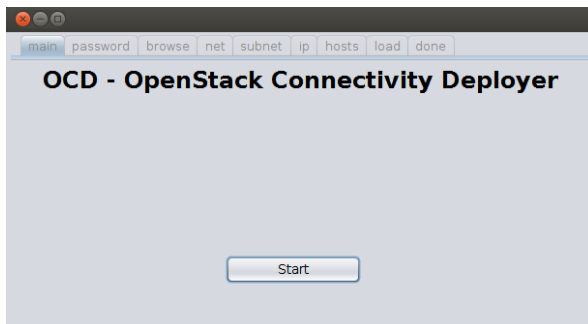


Figura 1: Tela inicial.

Tela inicial ilustrada pela Figura

1.



Figura 2: Tela de senhas.

Na segunda tela ilustrada pela Figura 2, o usuário deve fornecer a senha de root (ou usuário privilegiado) do host físico e a da MV.

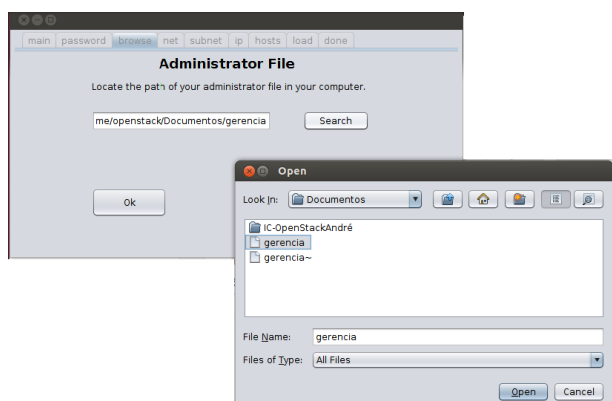


Figura 3: Arquivo de gerência.

Na terceira tela ilustrada pela Figura 3, o arquivo de gerência deve ser localizado pelo usuário. Para isso, é necessário clicar no botão *search* e navegar pelos diretórios em busca do arquivo *gerencia.txt* configurado.

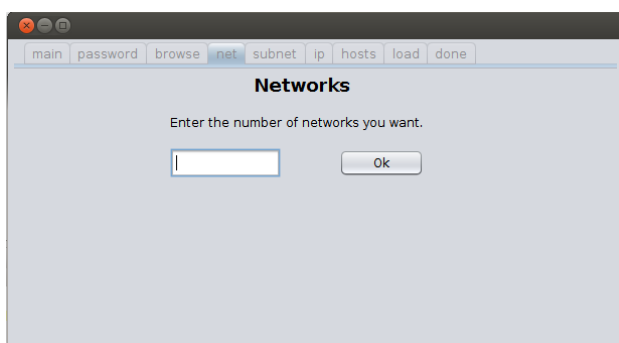


Figura 4: Criação de Redes.

Na quarta tela ilustrada pela Figura 4, o número de redes que o usuário deseja deve ser inserido.

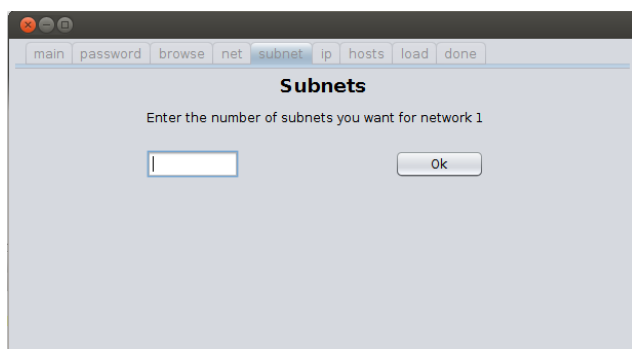


Figura 5: Quantidade de sub-redes.

Na quinta tela ilustrada pela Figura 5, o usuário deve inserir o número de sub-redes que pretende instanciar, esse número será pedido para cada rede instanciada.

Obs: De acordo com o número informado pelo usuário nas Figuras 4 e 5, as próximas telas serão exibidas de forma dinâmica, ou seja, a quantidade de vezes que elas irão aparecer será equivalente ao número que o usuário informou nas Figuras 4 e 5.

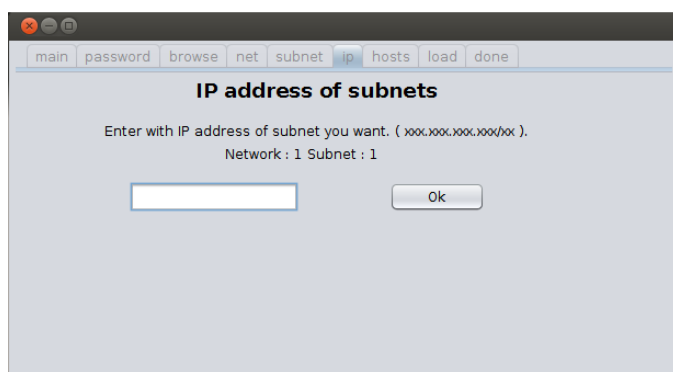


Figura 6: IP da sub-rede.

redes com máscara /24. Na Figura 6, pode ser observado que a ferramenta exibe para o usuário a sub-rede que está sendo configurada o IP. Essa tela será exibida repetidas vezes até que todos os IPs de cada uma das sub-redes tenham sido obtidos.

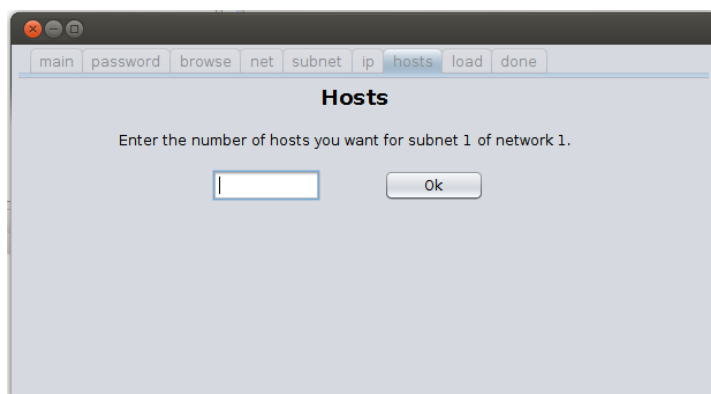


Figura 7: Criação das instâncias de MVs.

Na sétima tela ilustrada pela Figura 7, deve ser informado pelo usuário o número de instâncias de MV que serão atribuídas a cada rede configurada. Essa tela também será repetida para cada sub-rede existente.

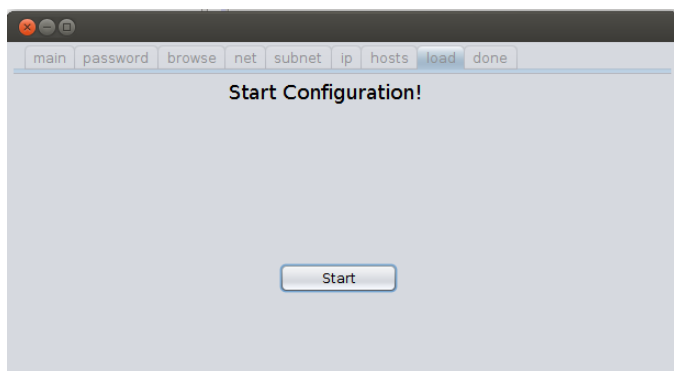


Figura 8: Início da configuração.

Na oitava tela ilustrada pela Figura 8, após todas as entradas terem sido obtidas basta que o usuário pressione start para que a ferramenta inicie as devidas configurações.

### 3. Cenário Exemplo

Para validarmos a ferramenta OCD utilizamos o cenário descrito no artigo para o salão de ferramentas, segue o link: <https://github.com/andrebeltrami/OpenStack-Connectivity-Deployer> . Para relembrar, nosso objetivo é obter a comunicação entre as instâncias H, conforme Figura 9.

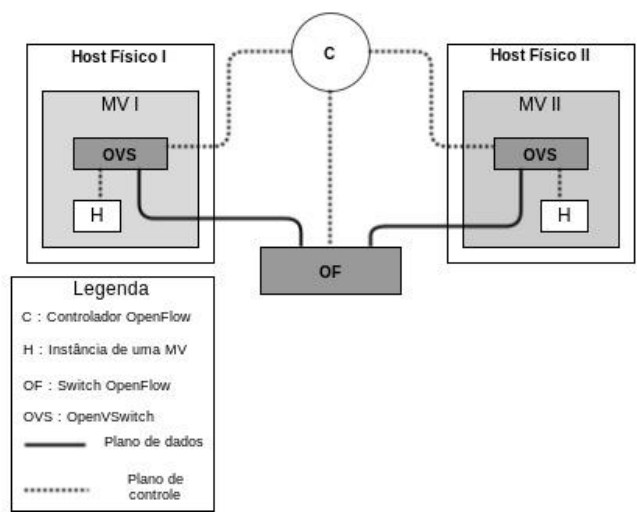


Figura 9: Cenário Exemplo

O cenário utilizado para testar a ferramenta possui duas interfaces de rede; uma interface wireless USB em cada host físico para o plano de controle, uma interface física em cada host físico para o plano de dados. A ferramenta deve ser executada em cada um dos hosts físicos. Portanto dois arquivos de gerencia devem ser configurados, segue as configurações utilizadas em cada um dos hosts físicos.

Arquivo gerencia.txt do host físico do lado esquerdo e arquivo de gerencia.txt do host físico II do lado direito.

Data Plane:20.20.20.0/24	Data Plane:20.20.20.0/24
Control Plane:192.168.1.0/24	Control Plane:192.168.1.0/24
IP of MV:192.168.1.115	IP of MV:192.168.1.116
Login of MV:openstack-slave	Login of MV:openstack-slave2
IP of other Subnet:30.30.30.0/24	IP of other Subnet:40.40.40.0/24

As entradas utilizadas para obter a conectividade nesse cenário estão descritas adiante:

Executando a ferramenta OCD para o host físico 1.

- Senhas de root para o host físico I e senha de root para MV I.
- Localizar o arquivo gerencia.txt no host físico I.
- Quantidade de redes = 1.
- Quantidade de sub-redes = 1.
- IP da sub-rede = 40.40.40.0/24.
- Quantidade de Instâncias de MV = 1.

Executando a ferramenta OCD para o host físico 2.

- Senhas de root para o host físico II e senha de root para MV II.
- Localizar o arquivo gerencia.txt no host físico II.
- Quantidade de redes = 1.
- Quantidade de sub-redes = 1.
- IP da sub-rede = 30.30.30.0/24.
- Quantidade de Instâncias de MV = 1.

Após finalizar a configuração de cada host físico, para o cenário estar operante é necessário a criação de regras OpenFlow.



## 4. Regras OpenFlow

Regras OpenFlow devem ser configuradas pelo Controlador em cada um dos três switches OpenFlow presentes no cenário. O Controlador utilizado para validar o cenário foi o Floodlight, para o envio de regras utilizamos a API REST Static Flow Pusher do Floodlight.

A Figura 10 ilustra cada uma das seis regras OpenFlow que foram criadas para este cenário.

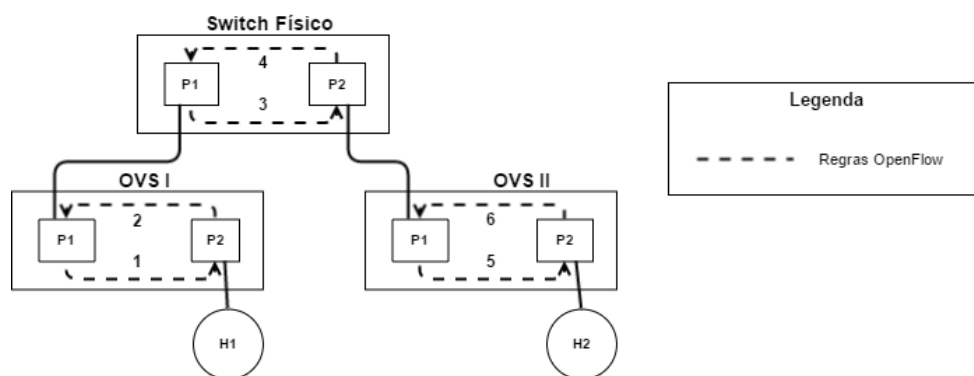


Figura 10: Regras OpenFlow utilizadas.

**Em 1:** Tudo que entra na porta P1 do Open vSwitch I sai pela porta P2 do Open vSwitch I. Comando curl utilizado: `curl -d '{"switch": "00:00:00:00:00:00:00:01", "name": "flow-mod-1", "cookie": "0", "priority": "32768", "ingress-port": "<P1 OVS I>", "active": "true", "actions": "output=<P2 OVS I>"}`  
`http://<controller_ip>:8080/wm/staticflowentrypusher/json .`

**Em 2:** Tudo que entra na porta P2 do Open vSwitch I sai pela porta P1 do Open vSwitch I. Comando curl utilizado: `curl -d '{"switch": "00:00:00:00:00:00:00:01", "name": "flow-mod-2", "cookie": "0", "priority": "32768", "ingress-port": "<P2 OVS I>", "active": "true", "actions": "output=<P1 OVS I>"}`  
`http://<controller_ip>:8080/wm/staticflowentrypusher/json.`

**Em 3:** Tudo que entra na porta P1 do switch físico sai pela porta P2 do switch físico. Comando curl utilizado: `curl -d '{"switch": "00:00:00:00:00:00:00:02", "name": "flow-mod-3", "cookie": "0", "priority": "32768", "ingress-port": "<P1 do Switch Fís I>", "active": "true", "actions": "output=<P2 do Switch Fís II>"}'`  
`http://<controller_ip>:8080/wm/staticflowentrypusher/json.`

**Em 4:** Tudo que entra na porta P2 do switch físico sai pela porta P1 do switch físico. Comando curl utilizado: `curl -d '{"switch": "00:00:00:00:00:00:00:02", "name": "flow-mod-4", "cookie": "0", "priority": "32768", "ingress-port": "<P2 do Switch Fís I>", "active": "true", "actions": "output=<P1 do Switch Fís II>"}'`  
`http://<controller_ip>:8080/wm/staticflowentrypusher/json.`

**Em 5:** Tudo que entra na porta P1 do Open vSwitch II sai pela porta P2 do Open vSwitch II. Comando curl utilizado: `curl -d '{"switch": "00:00:00:00:00:00:00:03", "name": "flow-mod-5", "cookie": "0", "priority": "32768", "ingress-port": "<P1 OVSI-I>", "active": "true", "actions": "output=<P2 OVS II>"}'`  
`http://<controller_ip>:8080/wm/staticflowentrypusher/json.`

**Em 6:** Tudo que entra na porta P2 do Open vSwitch II sai pela porta P1 do Open vSwitch II. Comando curl utilizado: `curl -d '{"switch": "00:00:00:00:00:00:00:03", "name": "flow-mod-6", "cookie": "0", "priority": "32768", "ingress-port": "<P2 OVSI-I>", "active": "true", "actions": "output=<P1 OVS II>"}'`  
`http://<controller_ip>:8080/wm/staticflowentrypusher/json.`

Após configurar as regras em seus respectivos switches a comunicação do plano de dados estará funcional. Para verificar, um teste de conectividade foi realizado através do ping entre as instâncias de MV H1 e H2.