ISSN 1677-8464

Foto:Pexels

COMUNICADO TÉCNICO

131

Campinas, SP
Dezembro/2018

Embrapa

IPAgrIDados: Tutorial para Criação de Cluster Virtual R

Alan Massaru Nakai
Jorge Luiz Correa
Renato José Santos Maciel

IPAgrIDados: Tutorial para Criação de Cluster Virtual R[1]

1. Introdução

A Infraestrutura Computacional para Pesquisa Agropecuária Intensiva em Dados (IPAgrIDados) é sediada na Embrapa Informática Agropecuária. Seu foco é a disponibilização de dois tipos de infraestrutura computacional: para processamento e para armazenamento de dados científicos. A infraestrutura é composta por diversos servidores de alta capacidade para processamento de dados, bem como de equipamentos específicos para armazenamento. Sua utilização está condicionada à realização de parcerias em projetos de pesquisa com a Empresa Brasileira de Pesquisa Agropecuária (Embrapa). A IPAgrIDados visa oferecer autonomia e otimização no uso de recursos computacionais para pesquisadores com demandas por estes dois tipos de recursos.

Para tanto, sua implementação é baseada na suíte de ferramentas OpenStack (Open..., 2018). O OpenStack é um conjunto de softwares que, operando controlada e colaborativamente, estabelece uma nuvem computacional no modelo Infrastructure As A Service (IaaS). Este modelo de infraestrutura como serviço permite não só a otimização do uso de recursos computacionais e autonomia, mas também, como consequência, uma maior disponibilidade dos recursos, agilidade para a execução dos projetos de pesquisas e diminuição de custos financeiros. Os recursos são gerenciados de forma compartilhada, podendo ser alocados e desalocados conforme forem necessários. Projetos que necessitem de um alto poder computacional para processar certa quantidade de dados não mais necessitam adquirir servidores para isso, que possivelmente ficariam ociosos grande parte do tempo. A característica da sazonalidade no uso dos recursos permite que sua capacidade seja melhor utilizada e por um maior número de projetos.

O R (The R Foundation, 2018) é um software estatístico, gratuito, amplamente adotado pela comunidade científica. Permite a criação de programas a partir de uma linguagem de programação própria e possui uma grande variedade de pacotes com ferramentas para diversas áreas do

[1] Alan Massaru Nakai, cientista da computação, doutor em Ciência da Computação, analista da Embrapa Informática Agropecuária, Campinas, SP. Jorge Luiz Corrêa, cientista da Computação, mestre em Ciência da Computação, analista da Embrapa Informática Agropecuária, Campinas, SP. Renato José Santos Maciel, cientista da Computação, analista da Informação da Embrapa Informática Agropecuária, Campinas, SP.

conhecimento. O R pode ser instalado em uma série de sistemas operacionais sendo muito utilizado em análises de dados, geração de gráficos e afins. Além disso, possui facilidades que auxiliam o processamento distribuído em múltiplos servidores de processamento.

Este documento exemplifica o uso da IPAgrIDados, apresentando o passo a passo da criação de um cluster virtual pré-configurado para processamento distribuído com o software R. São apresentados scripts de inicialização para o ambiente virtual R, além de um exemplo de processamento utilizando o pacote Simple Network of Workstations (snow) (Tierney et al., 2018). Para acompanhar este tutorial, o leitor deve ter conhecimentos básicos em ferramentas e tecnologias comumente utilizados em ambientes como este, tais como Secure SHell (SSH), Network File System (NFS), Advanced Packaging Tool (APT), Secure Copy Protocol (SCP) e FIle Transfer Protocol (FTP), assim como na configuração de ambientes Linux.

2. Descrição do Cluster Virtual

A Figura 1 ilustra o cluster virtual a ser configurado. Um nó principal (mestre) servirá de ponto de acesso externo para os usuários, tanto por meio de SSH quanto via web, por meio do RStudio Server (Pylvainen, 2016), uma Integrated Development Environment (IDE) para programação e execução de R em servidores remotos. Para simplificar o exemplo, o nó principal também acumulará a função de servidor NFS, compartilhando uma área de

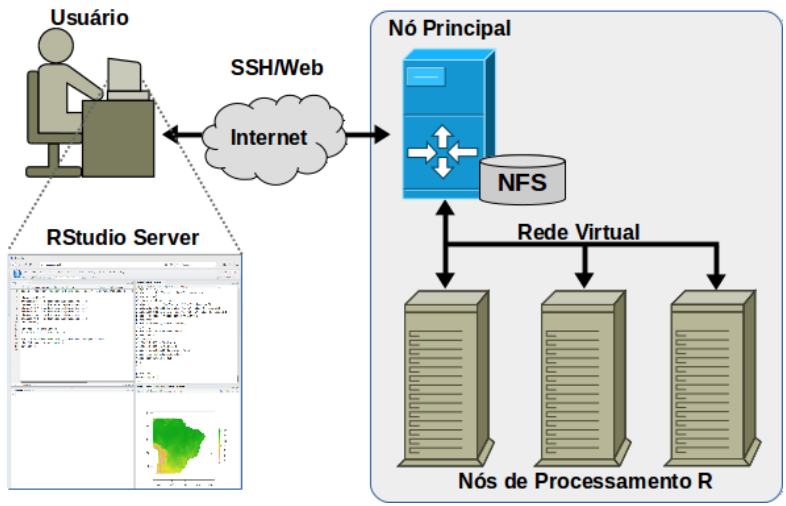

Figura 1. Esquema do cluster virtual a ser criado.

armazenamento que será acessível por todos os nós do cluster.

O nó principal será conectado a um número arbitrário de nós processadores (escravos) por meio de uma rede virtual interna, criada pelo OpenStack, de forma que qualquer interação do usuário com nós processadores deve ocorrer via nó principal. Todos os hosts são pré-configurados com chaves de autenticação que permitem que todos possam acessar uns aos outros, via SSH, sem a necessidade de senhas. Esta característica é necessária para utilização do pacote R snow, voltado para processamento paralelo em clusters. Além disso, os nós de processamento têm o software R pré-configurado e a área de armazenamento NFS montada automaticamente.

3. Passo a Passo

Para criar e gerenciar seu ambiente virtual na IPAgridados, o usuário deve acessar o OpenStack Dashboard[2]. A autenticação é realizada utilizando-se a matrícula e a senha corporativa da Embrapa e o usuário deve estar cadastrado em algum projeto da IPAgrIDados.

A Seção 3.1 apresenta os scripts de inicialização, baseados na ferramenta Cloud-init (Cloud-init, 2018), para configuração automatizada do cluster virtual R. Em seguida, a Seção 3.2 mostra o passo a passo para instanciação do nó principal. Finalmente, na Seção 3.3, é abordada a instanciação dos nós processadores. É importante que a instanciação dos nós processadores seja disparada apenas depois que a configuração do nó principal esteja completa, pois a configuração daqueles depende de que o servidor NFS esteja operacional.

Para facilitar as explicações, nas próximas seções, convenciona-se que as etiquetas que destacam determinados elementos das figuras serão referenciadas pelo nome da figura seguido de um traço e o número da etiqueta. Exemplo: “Figura 2 – 1” referencia a etiqueta 1 da Figura 2.

3.1. Scripts de Inicialização

O Código Fonte 1 apresenta o script de inicialização do nó principal. Observe que a indentação do código define as seções do script e deve ser respeitada (os scripts cloud-init utilizam a sintaxe YAML - YAML Ain’t Markup Language). Nas linhas 3 a 18, encontra-se a definição de usuários. Segue a descrição dos campos desta seção do código:

- – name: Nome do usuário. No exemplo, utilizar-se-á pcade, o nome do projeto no OpenStack. O traço indica que se iniciou uma nova definição de usuário, a única neste exemplo;
- gecos: Nome completo do usuário;
- groups: Lista de grupos aos quais o usuário será adicionado. No exemplo, o

[2] Disponível em: <www.openstack.cnptia.embrapa.br>.

usuário é adicionado ao grupo sudo, para poder realizar tarefas administrativas;

• lock_passwd: Define se o usuário pode (false) ou não (true) se autenticar por senha;

• passwd: Senha encriptada do usuário gerada a partir do comando:

$> mkpasswd –method=SHA-512

Neste exemplo, utilizou-se a senha "pcade".

• ssh-authorized-keys: chave pública do par de chaves criptográficas utilizado para que o usuário possa acessar os outros hosts do cluster sem a necessidade de senhas. O par de chaves pode ser gerado a partir do comando:

$> ssh-keygen -t rsa

O comando solicitará a definição do nome do arquivo a ser gerado. Este arquivo conterá a chave privada, enquanto a versão do arquivo com extensão .pub conterá a chave pública;

• shell: define o interpretador de comandos padrão.

A diretiva ssh_pwauth, na linha 20, configura o servidor SSH da máquina virtual para aceitar autenticação via senha. Esta característica é interessante para que o usuário externo possa acessar o nó principal sem possuir a chave criptográfica.

A seção write_files, nas linhas 22 a 58, permite escrever arquivos arbitrários no sistema de arquivos da máquina virtual. Esta funcionalidade é útil, por exemplo, quando é necessário criar arquivos de configuração para determinados pacotes. No exemplo, são criados dois arquivos. A criação de cada arquivo é iniciada com a diretiva "- content".

O primeiro arquivo gerado é a versão privada do par de chaves criptográficas. Neste exemplo, é necessário que todos os hosts possuam as duas chaves para permitir que o usuário padrão possa se conectar de um host para outro sem a necessidade de senha. O conteúdo da chave privada é definido nas linhas 24 a 50 e o arquivo é salvo como /etc/ssh/id_rsa com permissão apenas de leitura, conforme definido pelas diretivas path (linha 51) e permissions (linha 52) respectivamente.

O segundo arquivo gerado (linhas 53 a 58) configura o servidor SSH para não exigir a confirmação da identidade dos *hosts* destino, o que levaria à

Código Fonte 1: Script de inicialização do nó principal.

```
1 #cloud-config
2
3   users:
4   - name: pcade
5     gecos: pcade
6     groups: sudo
```

```
    lock_passwd: false
    passwd:
$6$Cy92yOi50w5.ng$yxa/ESNb9RuZb9am/jsPGmOp4e7QL8Dz6ApcMWtWHJfHgqf3xffz7ZY5KZNMQv
IgWfRtzP5xG2.rTynZj0MrA/
    ssh-authorized-keys:
      - ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQClY7SHzJsMwRQVPVhk72NudvsmTKT73Vom9p3vFIP6IOjQgHRe
FjG5xWXlcyF4dsoH3LLO8Pq0m2rUmdMYBaaQD4hucr8+Dc4EhDEyDj94A8GC8KBX2SgmgWlYLZpMbcgZ
cbv0p1D18KP9HFU2rnJfBzEh8k+DzrNxMhDkuavh9ywda7xKj5xoeXlPfEjv7NhVJQZt/pVFifYm6iOC
JdwnMyEqpfutUys+lgfQkpp0LLe32vJq3qHXnOhS39j596lbSv7iFcq6IYEaRn98RID0DDEYRjwTTI4q
VYE/2/tL3DOUDwlsUGYrM3dmnPlSkU84InuEYZLgr8Bq+wCW8oBH Generated-by-Nova
    shell: /bin/bash


  ssh_pwauth: yes


  write_files:
    - content: |
        -----BEGIN RSA PRIVATE KEY-----
        MIIEqQIBAAKCAQEApWO0h8ybDMEUFT1YZO9jbnb7Jkyk+91aJvad7xSD+iDo0IB0
        XhYxucVl5XMheHbKB9yyzvD6tJtq1JnTGAWmkA+IbnK/Pg3OBIQxMg4/eAPBgvCg
        V9koJoFtWC2aTG3IGXG79KdQ9fCj/RxVNq5yXwcxIfJPg86zcTIQ5Lmr4fcsHWu8
        So+caHl5T3xI7+zYVSUGbf6VRYn2JuojgiXcJzMhKqX7rVMrPpYH0JKadCy3t9ry
        at6h15zoUt/Y+fepW0r+4hXKuiGBGkZ/fESA9AwxGEY8E0yOKlWBP9v7S9wzlA8J
        bFBmKzN3Zpz5UpFPOCJ7hGGS4K/AavsAlvKARwIDAQABAoIBADPVt/MH0WCB8wVl
        IJsh3Av3BmfOhbbafTLRAFa6G1L9lXojiFUkp5kNoQ30s3zJ3i/wn6n0JN8OBsFz
        JJHW04k7FQtosz+DnBoWRETn6kR3COM8/0QJL2hCED8eq8lgbuJqnvnipqEuNndR
        tfNZYUcfPerhwsIQunSzuE+3UYOULybdFJ3bojAy/Wbdn7ySZePi5NX6Goz0VX2q
        dfojnm/KUCDv3fhuv/RCwX42ureZ4qJ/BXK77tmkt2adOrm9QdjP2a4y9q7ycgUw
        CoCKSVZIExyZti9XKEK9jQjBmrL97hI9jtU+IiNqXdgkI3XlLSwW9YtAhXSsddya
        ldF/0+ECggCBAMvYxBaO7lO73aBvl5sHzWbCr8GduWLiUMJIj+RkM5A/eCy1vYVG
        TjMB6138UwbaMhFaXfhp64YEr4fCAmnpJL72+r8Ao9PBcYe6P1iEtWm0qMGSbt6r
        y9uSrNc9YZQccNHv8+WLiBJE7PVptI7S4HxkCJhE7XNxOj0FgOVlF3t1AoIAgQDP
        tCAZ2yJq8ray0BoKW8HjD6Gt/pkLad4wTdOYcqDgugE//qeQ+Ii6lHm8HQhG5aZe
        a9NTBV6S9IjCSbct3p8F1ejSRrDFCzgQNnlSVSzBZFzJpnw/OsRaueLL7nQtmLQB
        XvaVddVycNAwS8CJrphoIBOveM9L/WtPvASR/uJhSwKCAIEAnkt6GjpsB6KRfCCs
        DguFCaNtcGDWsV8lb3mBo83b7Mwcujt3LKNn3lxDNfzXTJ7r+mW+S0WVS8EfvcB+
        t1gA3PpVxTTuA2ZYjxmmT0rexXMxCQB5cBjgeWXxKkPUHnvncF9xenNb+zPi/sBT
        IgXHpVDZC8WazubCXtiDO5/BGHECggCBAMaPcFDHiPOs7LOs9fgJGCW1f98SkcSz
        hNGw1SPcEahEqYVAMXkJ38xevLH1H0CHsrB5bCDl9S+Pzl3Okuuq82Nkx5qeMcdt
        nj8BlKH9AzgudfXgzs6zo5sIMaS7ZR1h6lOVMS5Y46Dv8Fb7VClLXC/O2oLZf3Mp
        5TjoiYX4oPpZAoIAgC2ploiYVR47GXddfTyJ7I81MBY634HXT5DHp9VKjLj8uxyh
        dLGPaIlOwc6w+7VnO9dKG7Jonn82/BJct+lY/AmY45/tl9TupS4ubCNFJHH+2f6P
```

```
49          yNvhgYA8SmH0F3CJSl6fKBKoIdfHY6LF82Ghbp2hifxW1dpl9Kh47AGBq+DY
50          -----END RSA PRIVATE KEY-----
51        path: /etc/ssh/id_rsa
52        permissions: '400'
53      - content: |
54            Host *
55               StrictHostKeyChecking no
56               UserKnownHostsFile=/dev/null
57        path: /etc/ssh/ssh_config
58        permissions: '644'
59
60    packages:
61      - r-base-core
62      - gdebi-core
63      - nfs-kernel-server
64      - nfs-common
65
66    runcmd:
67      - cp /etc/ssh/id_rsa /home/pcade/.ssh/id_rsa
68      - chown pcade:pcade /home/pcade/.ssh/id_rsa
69      - mkdir /opt/dados
70      - chown pcade:pcade /opt/dados
71      - echo '/opt/dados  192.168.0.0/24(rw,sync,no_subtree_check)' >>
72  /etc/exports
73      - /etc/init.d/nfs-kernel-server restart
74      - wget https://download2.rstudio.org/rstudio-server-1.1.383-amd64.deb
75      - gdebi -n rstudio-server-1.1.383-amd64.deb
76      - rm rstudio-server-1.1.383-amd64.deb
77      - Rscript -e 'install.packages("snow", repos="https://cran.rstudio.com")'
78
79    final_message: "Instalacao completa apos $UPTIME segundos!"
```

necessidade de uma resposta interativa do usuário durante a conexão SSH. O conteúdo do arquivo é definido nas linhas 54 a 56 e o arquivo é salvo como /etc/ssh/ssh_config com permissão 664, conforme definido pelas diretivas path (linha 57) e permissions (linha 58) respectivamente.

A seção packages, linhas 60 a 64, define uma lista de pacotes a serem instalados durante a primeira inicialização da máquina virtual. No caso do nó principal, os seguintes pacotes são instalados:

- r-base-core: Pacote base do software R;
- gdebi-core: Gerenciador de instalação de pacotes utilizado para instalar o RStudio Server, que

não é instalável via gerenciador APT;
- nfs-kernel-server: Servidor NFS;
- nfs-common: Cliente NFS.

A seção runcmd, linhas 66 a 77, define comandos de shell que são executados na primeira inicialização da máquina virtual. Nas linhas 67 e 68, o arquivo da chave criptográfica privada é copiado para o diretório home do usuário pcade. Isso é necessário para que a conexão SSH sem senhas seja possível. Nas linhas 69 e 70, um novo diretório é criado e o usuário pcade é definido como seu dono. As linhas 71 a 73 configuram e reiniciam o servidor NFS de forma que o novo diretório seja compartilhado na rede local do *cluster* virtual. Nas linhas 74 a 76, o software RStudio Server é descarregado e instalado e a linha 77 instala o pacote *snow* na biblioteca do R. Finalmente, a diretiva final_message, na linha 79, imprime uma mensagem no log de inicialização da máquina virtual, ao término do processo.

A versão do script de inicialização dos nós processadores, apresentada no Código Fonte 2, tem poucas modificações em relação ao script do nó principal. São elas:
- ssh_pwauth: diretiva removida, pois não há necessidade de permitir acesso SSH com senha nos nós processadores;
- packages: nesta seção, os pacotes relacionados à instalação do RStudio Server e do servidor NFS foram removidos da lista, já que tais softwares são hospedados apenas pelo nó principal;
- runcmd: nesta seção, os comandos relacionados à instalação do RStudio Server e à configuração do servidor NFS foram removidos. Foram adicionados comandos para montar o diretório NFS no nó processador (linhas 65 a 67).

Código Fonte 2: Script de inicialização para nós processadores.

```
1  #cloud-config
2
3    users:
4    - name: pcade
5      gecos: pcade
6      groups: sudo
7      lock_passwd: false
8      passwd:
9  $6$Cy92yOi50w5.ng$yxa/ESNb9RuZb9am/jsPGmOp4e7QL8Dz6ApcMWtWHJfHgqf3xffz7ZY5KZNMQv
10 IgWfRtzP5xG2.rTynZj0MrA/
11     ssh-authorized-keys:
12       - ssh-rsa
13 AAAAB3NzaC1yc2EAAAADAQABAAABAQClY7SHzJsMwRQVPVhk72NudvsmTKT73Vom9p3vFIP6IOjQgHRe
14 FjG5xWXlcyF4dsoH3LLO8Pq0m2rUmdMYBaaQD4hucr8+Dc4EhDEyDj94A8GC8KBX2SgmgWlYLZpMbcgZ
15 cbv0p1D18KP9HFU2rnJfBzEh8k+DzrNxMhDkuavh9ywda7xKj5xoeXlPfEjv7NhVJQZt/pVFifYm6iOC
16 JdwnMyEqpfutUys+lgfQkpp0LLe32vJq3qHXnOhS39j596lbSv7iFcq6IYEaRn98RID0DDEYRjwTTI4q
17 VYE/2/tL3DOUDwlsUGYrM3dmnPlSkU84InuEYZLgr8Bq+wCW8oBH Generated-by-Nova
18     shell: /bin/bash
19
20   write_files:
21     - content: |
```

```
22          -----BEGIN RSA PRIVATE KEY-----
23          MIIEqQIBAAKCAQEApWO0h8ybDMEUFT1YZO9jbnb7Jkyk+91aJvad7xSD+iDo0IB0
24          XhYxucVl5XMheHbKB9yyzvD6tJtq1JnTGAWmkA+IbnK/Pg3OBIQxMg4/eAPBgvCg
25          V9koJoFtWC2aTG3IGXG79KdQ9fCj/RxVNq5yXwcxIfJPg86zcTIQ5Lmr4fcsHWu8
26          So+caHl5T3xI7+zYVSUGbf6VRYn2JuojgiXcJzMhKqX7rVMrPpYH0JKadCy3t9ry
27          at6h15zoUt/Y+fepW0r+4hXKuiGBGkZ/fESA9AwxGEY8E0yOKlWBP9v7S9wzlA8J
28          bFBmKzN3Zpz5UpFPOCJ7hGGS4K/AavsAlvKARwIDAQABAoIBADPVt/MH0WCB8wVl
29          IJsh3Av3BmfOhbbafTLRAFa6G1L91XojiFUkp5kNoQ30s3zJ3i/wn6n0JN8OBsFz
30          JJHW04k7FQtosz+DnBoWRETn6kR3COM8/0QJL2hCED8eq8lgbuJqnvnipqEuNndR
31          tfNZYUcfPerhwsIQunSzuE+3UYOULybdFJ3bojAy/Wbdn7ySZePi5NX6Goz0VX2q
32          dfojnm/KUCDv3fhuv/RCwX42ureZ4qJ/BXK77tmkt2adOrm9QdjP2a4y9q7ycgUw
33          CoCKSVZIExyZti9XKEK9jQjBmrL97hI9jtU+IiNqXdgkI3XlLSwW9YtAhXSsddya
34          ldF/0+ECggCBAMvYxBaO7l073aBvl5sHzWbCr8GduWLiUMJIj+RkM5A/eCy1vYVG
35          TjMB6138UwbaMhFaXfhp64YEr4fCAmnpJL72+r8Ao9PBcYe6P1iEtWm0qMGSbt6r
36          y9uSrNc9YZQccNHv8+WLiBJE7PVptI7S4HxkCJhE7XNxOj0FgOVlF3t1AoIAgQDP
37          tCAZ2yJq8ray0BoKW8HjD6Gt/pkLad4wTdOYcqDgugE//qeQ+Ii6lHm8HQhG5aZe
38          a9NTBV6S9IjCSbct3p8F1ejSRrDFCzgQNnlSVSzBZFzJpnw/OsRaueLL7nQtmLQB
39          XvaVddVycNAwS8CJrphoIBOveM9L/WtPvASR/uJhSwKCAIEAnkt6GjpsB6KRfCCs
40          DguFCaNtcGDWsV81b3mBo83b7Mwcujt3LKNn3lxDNfzXTJ7r+mW+S0WVS8EfvcB+
41          t1gA3PpVxTTuA2ZYjxmmT0rexXMxCQB5cBjgeWXxKkPUHnvncF9xenNb+zPi/sBT
42          IgXHpVDZC8WazubCXtiDO5/BGHECggCBAMaPcFDHiPOs7LOs9fgJGCW1f98SkcSz
43          hNGw1SPcEahEqYVAMXkJ38xevLH1H0CHsrB5bCDl9S+Pzl3Okuuq82Nkx5qeMcdt
44          nj8BlKH9AzgudfXgzs6zo5sIMaS7ZR1h6lOVMS5Y46Dv8Fb7VClLXC/O2oLZf3Mp
45          5TjoiYX4oPpZAoIAgC2ploiYVR47GXddfTyJ7I81MBY634HXT5DHp9VKjLj8uxyh
46          dLGPaI1Owc6w+7VnO9dKG7Jonn82/BJct+lY/AmY45/tl9TupS4ubCNFJHH+2f6P
47          yNvhgYA8SmH0F3CJSl6fKBKoIdfHY6LF82Ghbp2hifxW1dpl9Kh47AGBq+DY
48          -----END RSA PRIVATE KEY-----
49        path: /etc/ssh/id_rsa
50        permissions: '400'
51      - content: |
52            Host *
53                StrictHostKeyChecking no
54                UserKnownHostsFile=/dev/null
55        path: /etc/ssh/ssh_config
56        permissions: '644'
57
58    packages:
59      - r-base-core
60      - nfs-common
61
62    runcmd:
63      - cp /etc/ssh/id_rsa /home/pcade/.ssh/id_rsa
64      - chown pcade:pcade /home/pcade/.ssh/id_rsa
65      - mkdir /opt/dados
66      - chown pcade:pcade /opt/dados
67      - sudo mount cluster-r-master:/opt/dados /opt/dados
68      - Rscript -e 'install.packages("snow", repos="https://cran.rstudio.com")'
69
70    final_message: "Instalacao completa apos $UPTIME segundos!"
```

3.3. Instanciando o Nó Principal

Os seguintes passos devem ser seguidos para instanciar o nó principal:

• **Passo 1:** Após acessar a interface principal do OpenStack Dashboard[3], no painel lateral esquerdo, clique em Compute/Instances (Figura 2 – 1) para acessar a tela de gerenciamento de instâncias de máquinas virtuais. Em seguida, clique no botão Launch Instance (Figura 2 – 2) para iniciar uma nova máquina virtual e uma janela de diálogo com o mesmo nome será aberta.

• **Passo 2:** Na janela Launch Instance, seção Details, preencha o nome da máquina virtual a ser criada (Figura 3 – 1) e clique em Next (Figura 3 – 2).

• **Passo 3:** Na janela Launch Instance, seção Source, preencha os seguintes campos:

◦ Select Source Boot (Figura 4 – 1): Selecione a opção Image, indicando que a máquina virtual será criada a partir de uma imagem já existente;

◦ Create New Volume (Figura 4 – 2): Selecione a opção Yes, indicando que um novo disco virtual será criado para a nova máquina;

◦ Volume Size (GB) (Figura 4 – 3): Escolha um tamanho para o novo disco virtual;

◦ Delete Volume on Instance Delete (Figura 4 – 4): Defina se o disco será apagado (Yes) ou não (No) caso a máquina virtual seja apagada;

◦ Escolha a imagem que será usada como base para criação da nova máquina virtual. No exemplo, utilizar-se-á a imagem Ubuntu Server 16.04 LTS (Figura 4 – 5);

◦ Clique em Next (Figura 4 – 6).

• **Passo 4:** Na janela Launch Instance, seção Flavor, escolha dentre as configurações disponíveis, qual será utilizada para a criação da nova máquina virtual. Esta escolha deve levar em consideração a cota de hardware disponível para o projeto que está sendo utilizado. No exemplo, será utilizado o tipo m1.large (Figura 5 – 1), com 4 CPUs e 8 GB de memória. Clique em Next (Figura 5 – 2);

• **Passo 5:** Na janela Launch Instance, seção Networks, selecione a rede virtual que será utilizada para conectar as máquinas do cluster. No exemplo, será utilizada a rede pcade-net (Figura 6 – 1). Em seguida, clique em Next (Figura 6 – 2). Todo projeto criado na IPAgrIDados tem uma rede virtual padrão pré-configurada, porém, caso seja necessário, o usuário pode criar outras redes virtuais. Para aprender como criar uma rede virtual, consulte a documentação de rede do OpenStack (Self-service…, 2018).

• **Passo 6:** Na janela Launch Instance, seção Configuration, selecione o script de inicialização para o nó principal

[3] Disponível em: <www.openstack.cnptia.embrapa.br>.

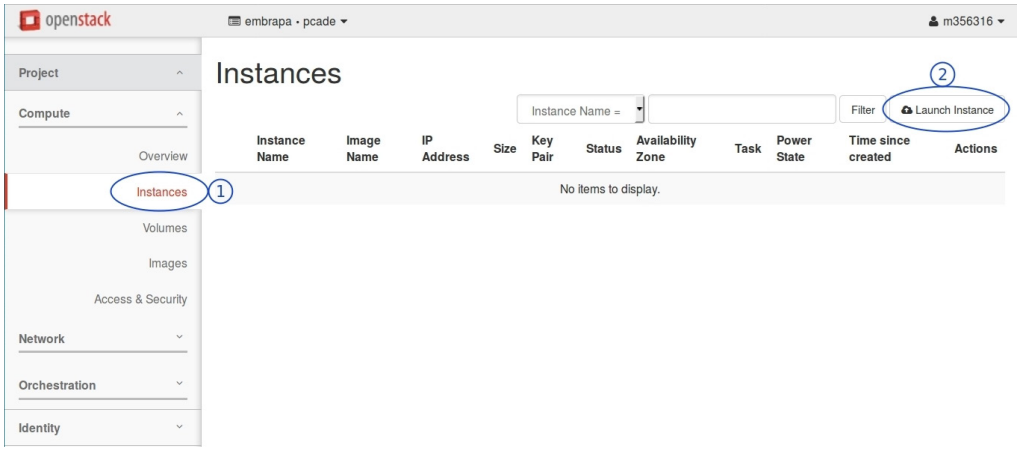

Figura 2. Tela de gerenciamento de instâncias de máquinas virtuais.

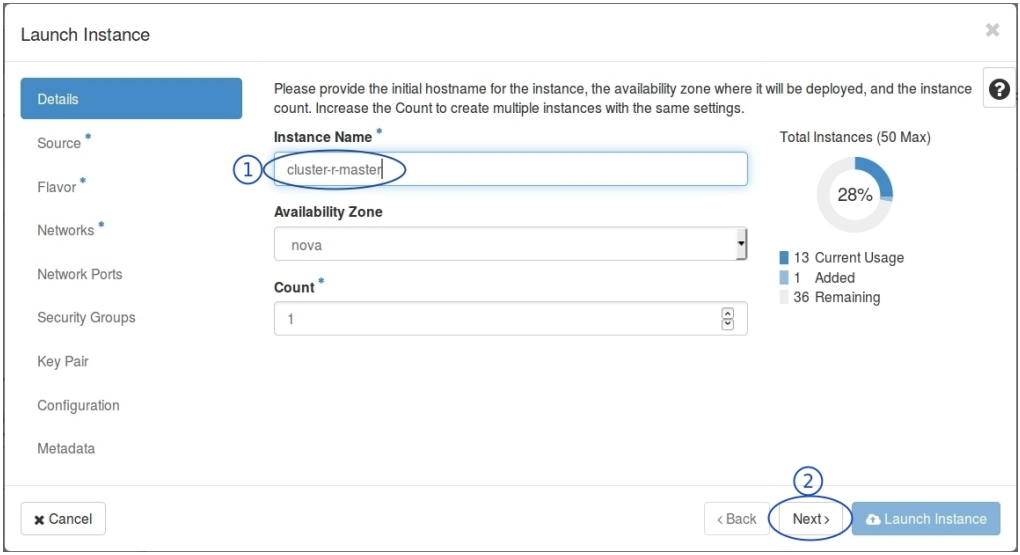

Figura 3. Janela de diálogo para instanciação de uma nova máquina virtual.

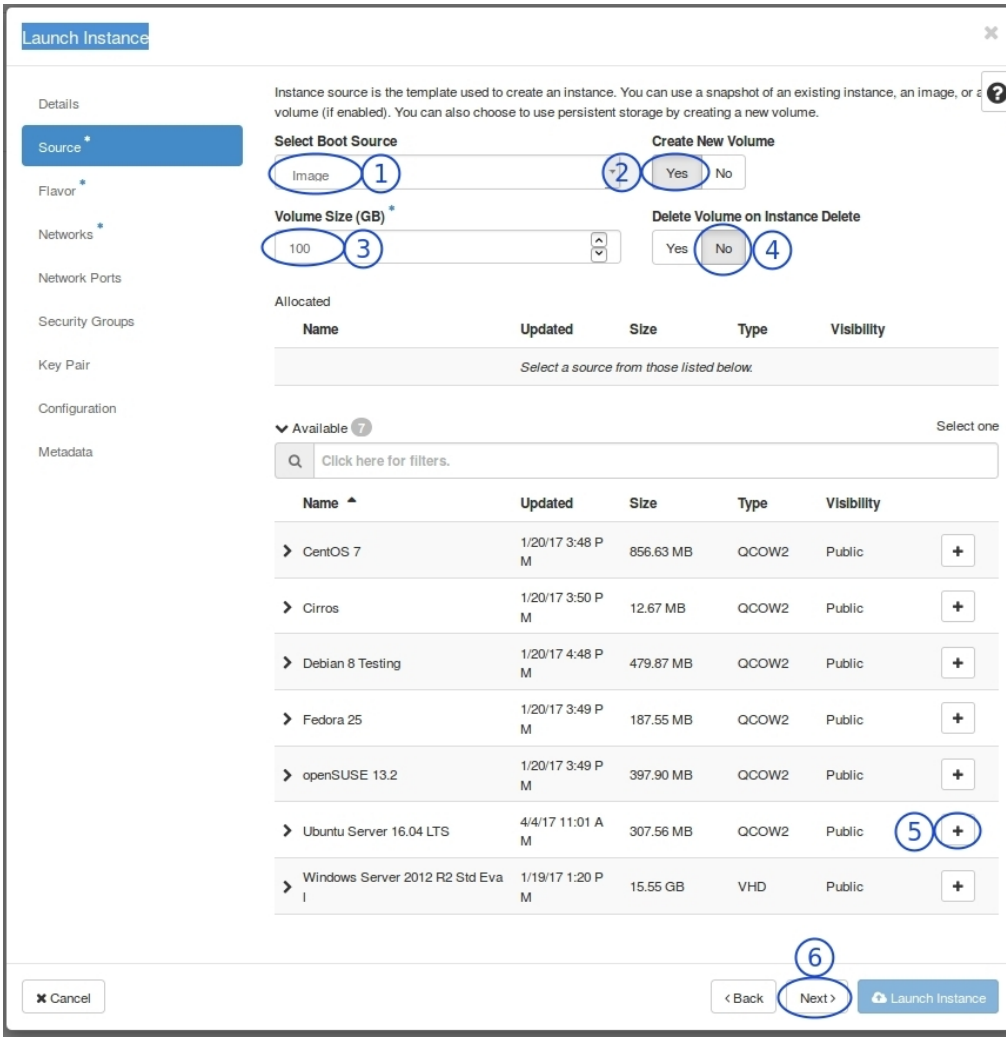

Figura 4. Definição do disco virtual e da imagem a ser utilizada como base para criação da nova máquina virtual.

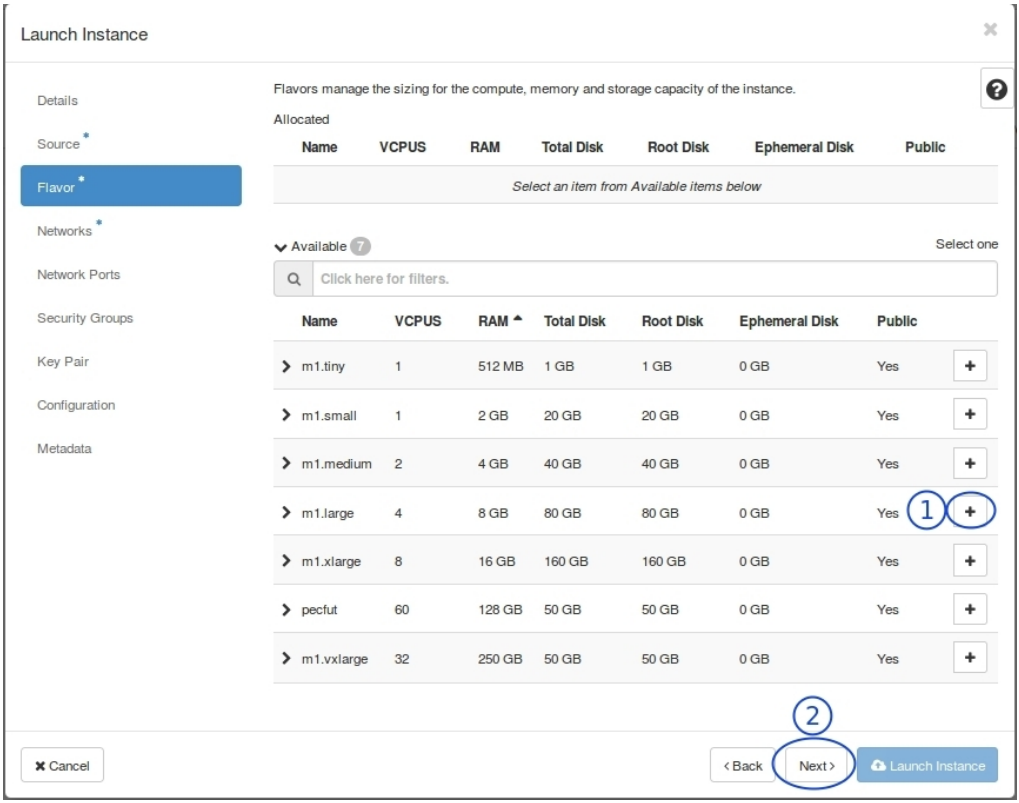

Figura 5. Dimensionamento da máquina virtual.

– no exemplo: script_inicialização_R_master.txt (Figura 7 – 1). Para criar o arquivo, utilize o exemplo apresentado no Código Fonte 1. Clique no botão Launch Instance (Figura 7 – 2).

- **Passo 7:** Após criada a instância do nó principal, deve-se atribuir um endereço IP externo a ele. Cada projeto da IPAgriDados terá disponível um endereço IPv4 externo. Para atribuir o endereço externo ao nó principal, na lista de instâncias, clique no menu de ações da máquina virtual e escolha a opção Associate Floating IP (Figura 8 – 1).

Em seguida, na janela de diálogo, selecione o IP externo disponível (Figura 9 – 1) e em seguida clique em Associate (Figura 9 – 2).

- **Passo 8:** Verifique se a instanciação da máquina virtual foi finalizada antes de prosseguir com o tutorial. Para isso, na lista de instâncias, clique sobre o nome da instância (cluster-r-master, no exemplo). Em seguida, clique na aba Log e no botão View Full Log. Uma nova aba do navegador será aberta com o log completo da inicialização. Ao rolar o log até o fim, se a mensagem de

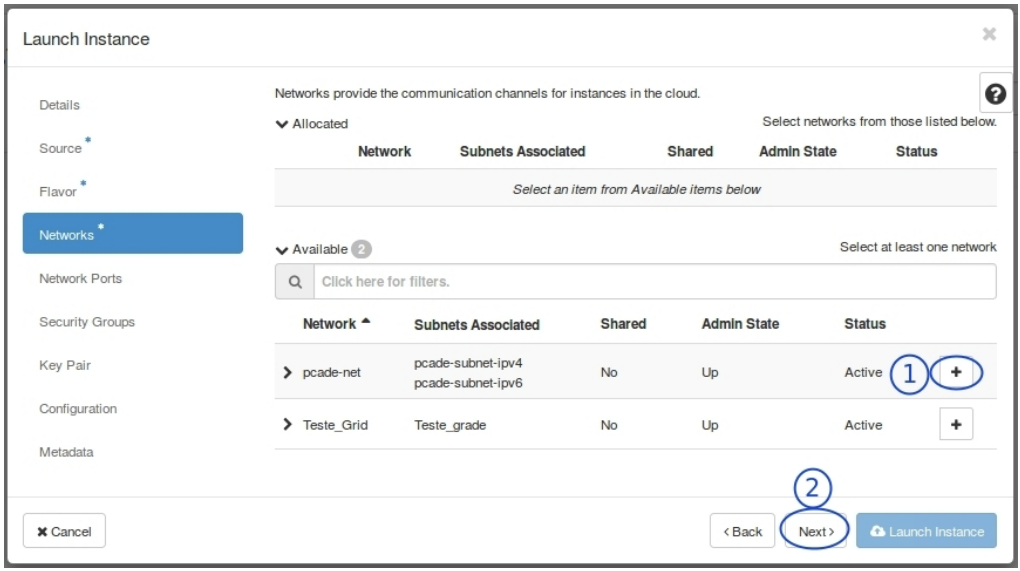

Figura 6. Definição da rede virtual ao qual a máquina estará conectada.

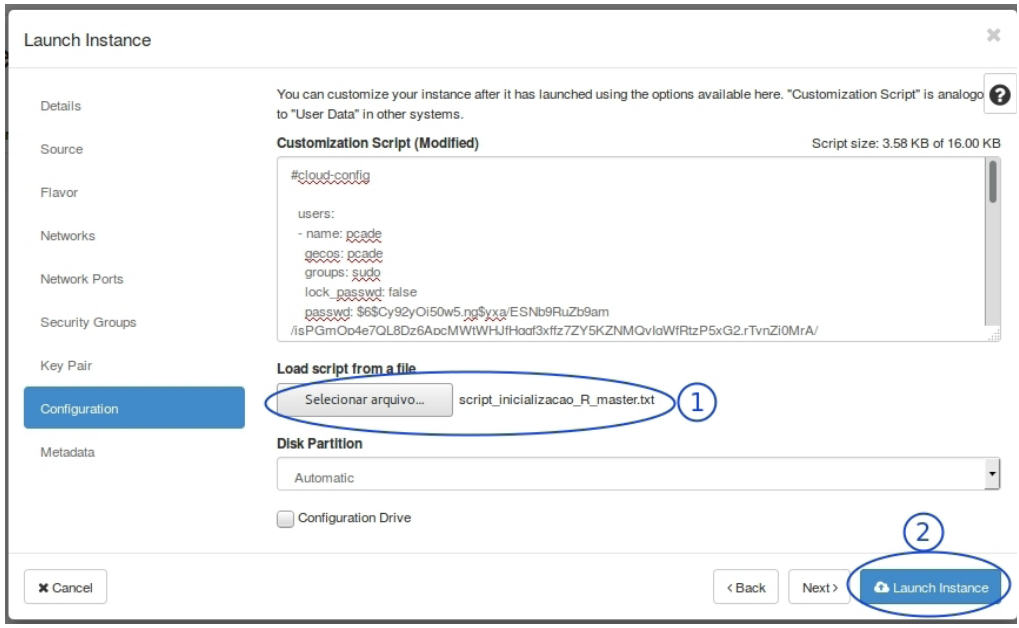

Figura 7. Seleção de script de inicialização para configuração do nó principal.

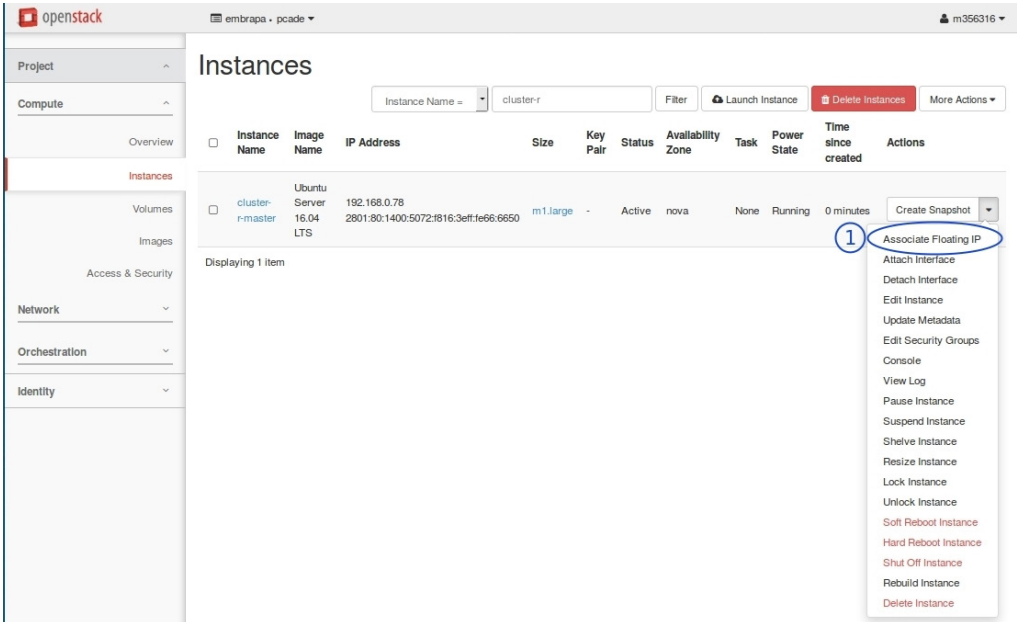

Figura 8. Atribuição de IP externo ao nó principal.

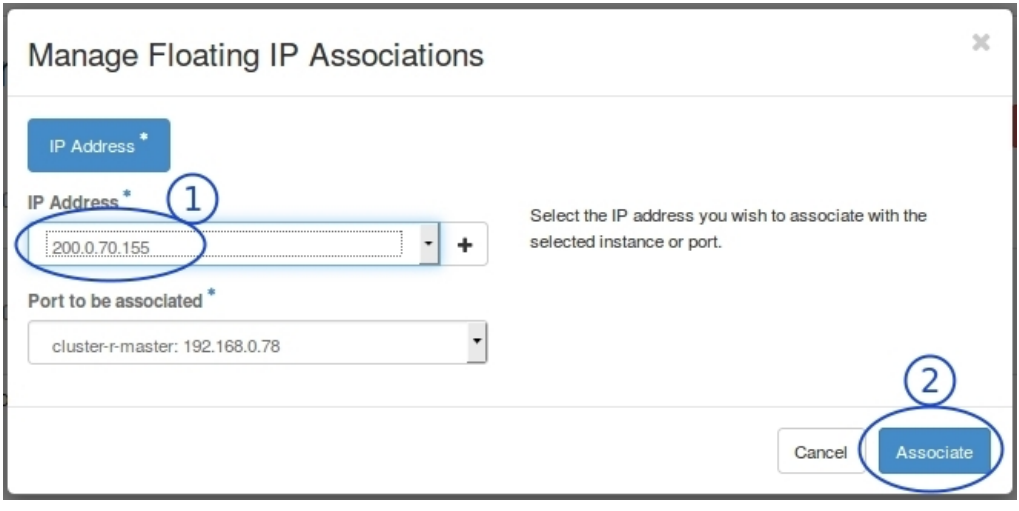

Figura 9. Atribuição de IP externo ao nó principal - janela de diálogo.

finalização tiver sido impressa (Figura 10 – 1), a instanciação foi finalizada com sucesso. Caso contrário, o script de inicialização pode ainda estar em execução. Neste caso, atualize a aba do navegado web após alguns minutos e verifique novamente.

3.3. Instanciação dos Nós Processadores

A instanciação dos nós processadores é mais simples do que a do nó principal, sendo a maior parte dos dois processos idêntica. A seguir, são descritos os passos para a instanciação dos nós processadores. Os detalhes dos passos já explicados na instanciação do nó principal serão omitidos.

• **Passo 1:** Acesse a interface principal do OpenStack Dashboard[4] e, no painel lateral esquerdo, clique em Compute/ Instances para acessar a tela de gerenciamento de instâncias de máquinas virtuais. Em seguida, clique no botão Launch Instance para iniciar o assistente de instanciação de máquinas virtuais.

• **Passo 2:** Na janela Launch Instance, seção Details, preencha o nome que servirá de base para os nomes das máquinas virtuais (Figura 11 – 1) e o número de nós processadores a serem instanciados (Figura 11 – 2). Os nomes das instâncias serão compostos pelo nome definido seguido de um

[4] Disponível em: <www.openstack.cnptia.embrapa.br>.

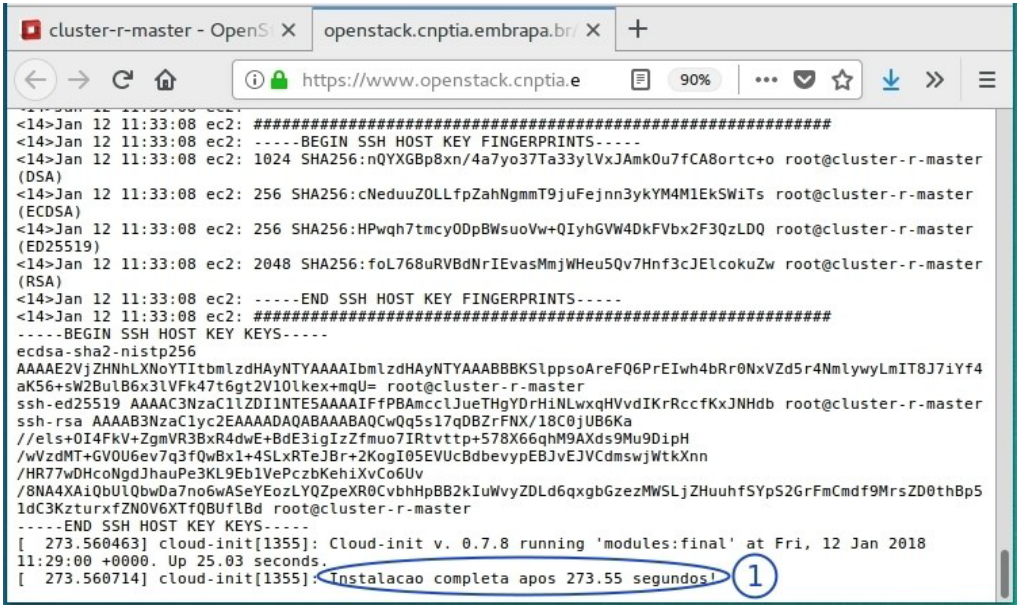

Figura 10. Log da inicialização da máquina virtual.

traço e de um número incremental. No exemplo, serão criadas as instâncias cluster-r-node-1, cluster-r-node-2 e cluster-r-node-3. Clique em Next (Figura 11 – 3).

• **Passo 3:** Na janela Launch Instance, seção Source, preencha os seguintes campos:

◦ Select Source Boot: Mantenha a opção Image, indicando que a máquina virtual será criada a partir de uma imagem já existente;

◦ Create New Volume: Selecione a opção No, indicando que não serão criados discos virtuais permanentes para os nós processadores. Isto significa que ao remover as instâncias dos nós processadores, seus sistemas de arquivo serão perdidos, o que não é um problema, pois os resultados gerados poderão ser armazenados no diretório NFS do nó principal;

◦ Escolha a imagem que será usada como base para criação da nova máquina virtual. No exemplo, utilizar-se-á a imagem Ubuntu Server 16.04 LTS;

◦Clique em Next.

• **Passo 4:** Na janela Launch Instance, seção Flavor, escolha a configuração que será utilizada para a criação da nova máquina virtual. No exemplo, será utilizado o mesmo tipo utilizado no nó principal: m1.large, com 4 CPUs e 8 GB de memória. Clique em Next;

• **Passo 5:** Na janela Launch Instance, seção Networks, selecione a rede virtual que será utilizada para conectar as máquinas do cluster. No exemplo, será utilizada a rede

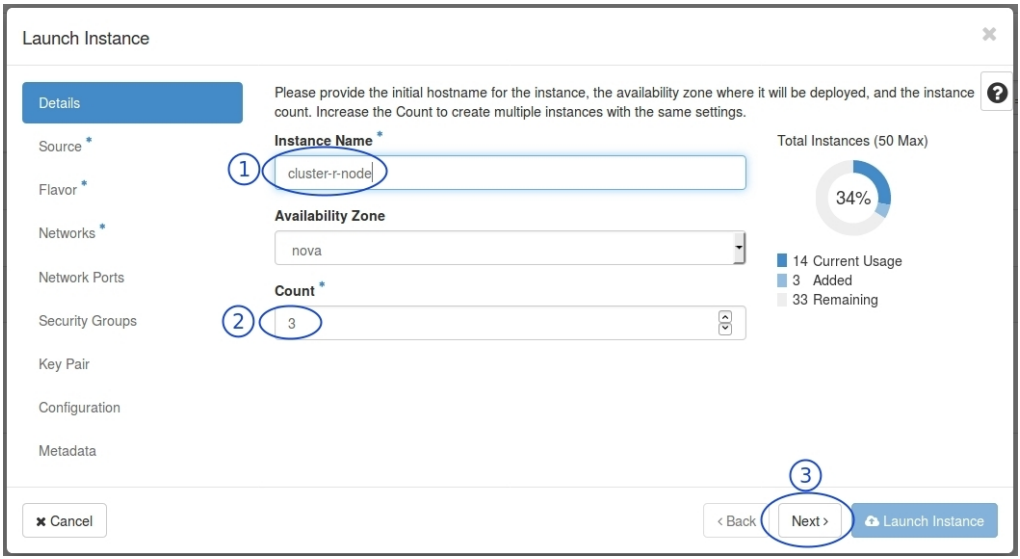

Figura 11. Iniciando a instanciação de nós processadores.

pcade-net. Em seguida, clique em Next.

• **Passo 6:** Na janela Launch Instance, seção Configuration, selecione o script de inicialização para os nós processadores. Para criar o arquivo, utilize o exemplo apresentado no Código Fonte 2. Clique no botão Launch Instance

• **Passo 7:** Verifique se a instanciação das máquinas virtuais foi finalizada antes de prosseguir com o tutorial.

4. Exemplo de Uso

Esta seção apresenta um exemplo de processamento distribuído no *cluster* virtual utilizando o pacote *snow*. Não é objetivo deste tutorial detalhar o uso do software R para computação distribuída, de forma que apenas um teste ilustrativo será apresentado.

O Código Fonte 3 apresenta o código R para o exemplo. Na linha 1, o pacote *snow* é carregado. Nas linhas 3 e 4, o comando makeSOCKcluster configura um *cluster* (objeto cl) do pacote *snow*. O array que parametriza esta função define a lista de *hosts* processadores. Cada ocorrência do nome de um host na lista define um slot de processamento, ou seja, um potencial processo paralelo. Isto significa que, neste exemplo, até seis tarefas poderão ser executadas simultaneamente, uma no nó principal (localhost), uma no nó cluster-r-node 1, duas no nó cluster-r-node-2 e duas no nó cluster-r-node-3.

As linhas 6 a 9 definem uma função, denominada funcao_teste, que recebe um valor como parâmetro e simplesmente imprime esse valor em um novo arquivo no diretório NFS (/opt/dados). Para facilitar a verificação do resultado, o nome do arquivo terá como sufixo a concatenação do nome do *host* que o gerou com o valor do parâmetro.

Na linha 11, o comando clusterApply executa a função funcao_teste para todos os valores da sequência de 1 a 20 passada como parâmetro. Cada chamada da função será submetida a um slot livre de processamento do *cluster* (cl). Após o término da execução, a conexão com o cluster é fechada na linha 13 com o comando stopCluster.

Para testar o programa no cluster, siga os seguintes passos:

• **Passo 1:** Em um navegador web, acesse a porta 8787 do endereço IP externo atribuído ao nó principal (Figura 12 – 1). O endereço IP aparece na lista de instâncias do OpenStack Dashboard. A tela de login do RStudio Server será carregada. Utilize o usuário e a senha criados pelo script de inicialização do nó principal (Figura 12 – 2) para acessar o sistema.

• **Passo 2:** Na tela principal do RStudio (Figura 13), clique na opção Novo/R Script (Figura 13 – 1) e uma nova aba de scripts em branco será aberta. Copie o código do exemplo nessa aba (Figura 13 – 2) e pressione o botão source (Figura 13 – 3). Esta ação carregará e executará o código do programa no terminal interativo (Figura 13 – 4). Observe que algumas mensagens são lançadas alertando o usuário de que os

Código Fonte 3. Exemplo de código R com pacote *snow*.

```
1  library(snow)
2
3  cl = makeSOCKcluster(c('localhost', 'cluster-r-node-1', 'cluster-r-node-2',
4  'cluster-r-node-2', 'cluster-r-node-3', 'cluster-r-node-3'))
5
6  funcao_teste = function(x) {
7    cat(x, file=paste0('/opt/dados/teste_', Sys.info()['nodename'], '_',x))
8    return('ok')
9  }
10
11 clusterApply(cl, 1:20, funcao_teste)
12
13 stopCluster(cl)
```

nós processadores foram adicionados à lista de *hosts* conhecidos do nó principal devido ao acesso SSH realizado pelo pacote *snow*. Tais alertas podem ser ignorados. Quando o *prompt* interativo (caractere “>”) aparecer no terminal, o processamento terá terminado.

• **Passo 3:** Para verificar os resultados do teste, na aba Files, localizada no painel inferior direito do RStudio Server, utilize o botão de navegação de diretórios (Figura 14 – 1) para acessar o diretório NFS (/opt/dados) e visualizar os arquivos gerados. Note que os nomes dos arquivos mostram que foram gerados pelos diferentes *hosts* do *cluster* virtual, indicando que o processamento foi paralelizado com sucesso.

• **Passo 4:** Para transferir os arquivos gerados no *cluster virtual* para o seu *desktop*, o usuário pode utilizar alguma ferramenta de transferência de dados,

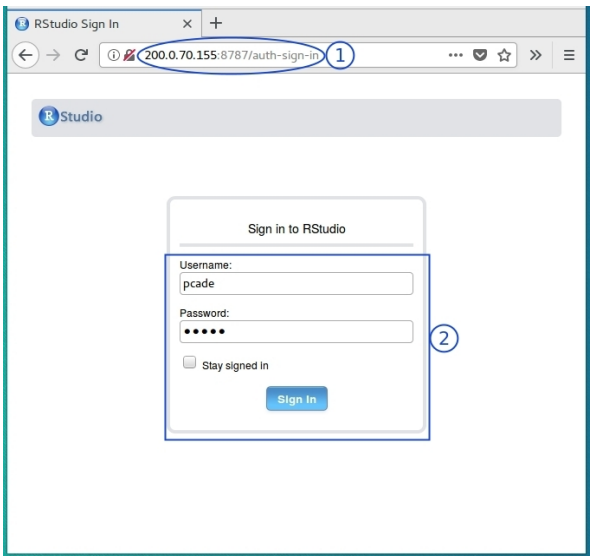

Figura 12. Tela de login do RStudio Server.

como SCP ou FTP. Por exemplo, o comando:

```
$>  scp -r pcade@200.0.70.155:/opt/dados/ /
tmp/dados
```

utiliza SCP para copiar o diretório /opt/dados do nó principal para o diretório local /tmp/dados, por meio do endereço IP externo atribuído a ele.

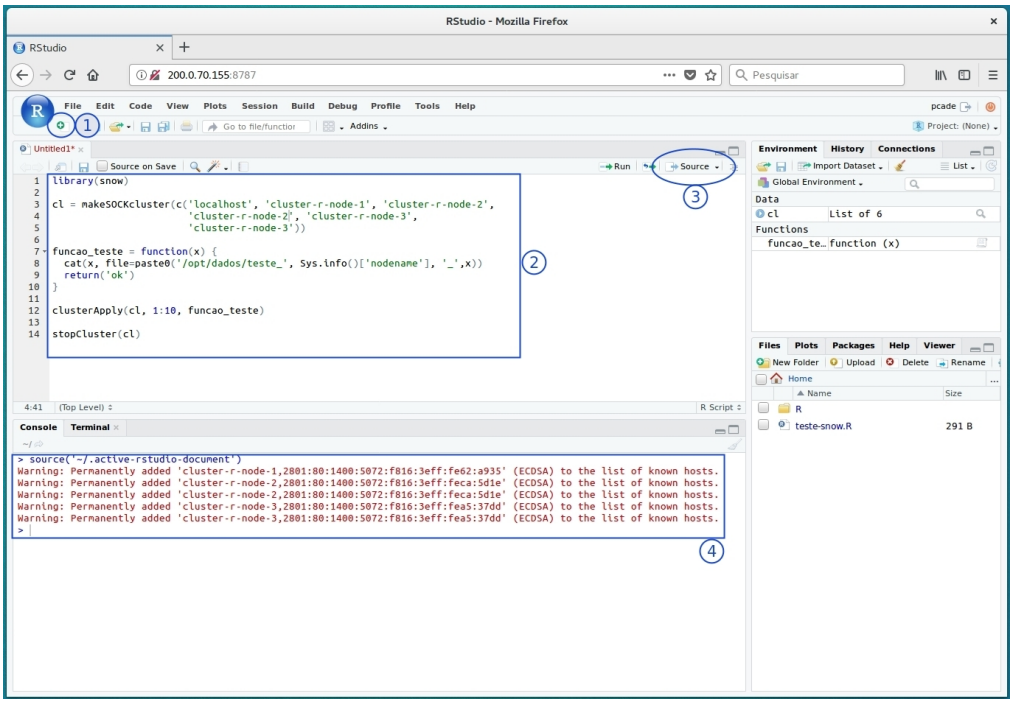

Figura 13. Tela principal do RStudio Server.

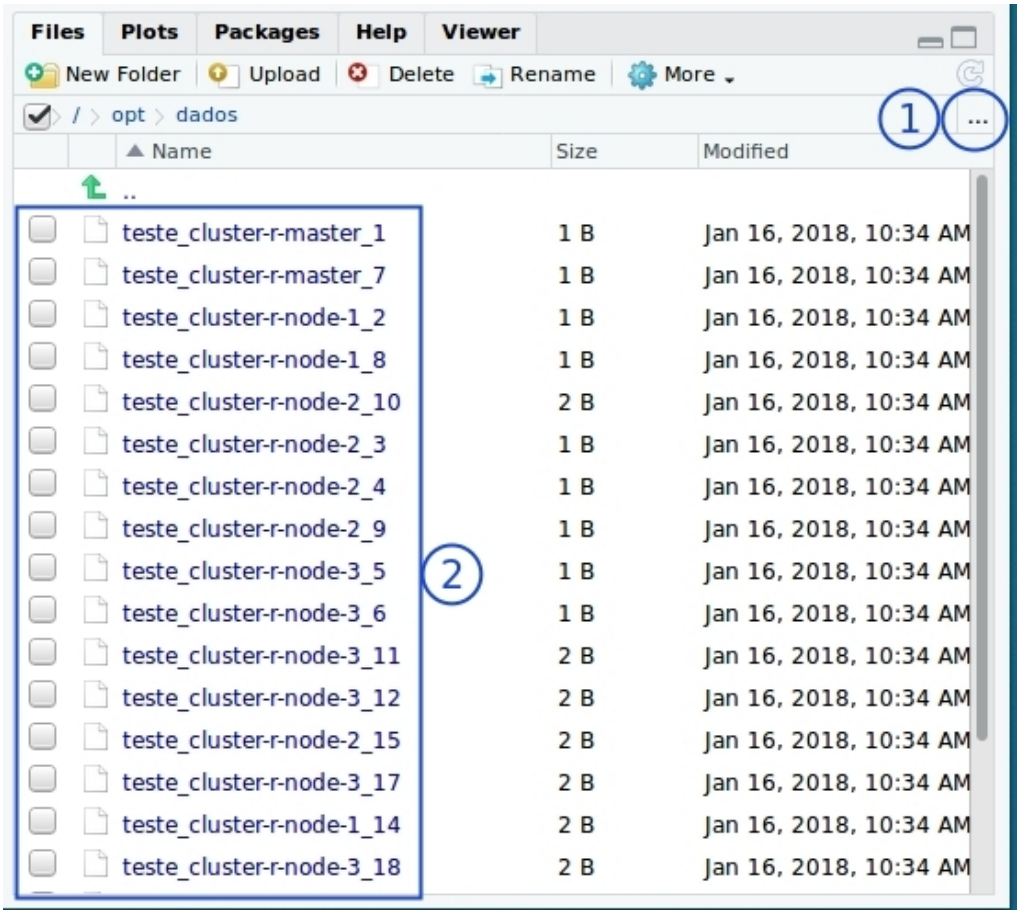

Figura 14. Visualização dos resultados.

Referências

CLOUD-INIT. Documentation. <https://cloudinit.readthedocs.io/en/latest/>. Acesso em: 11 dez. 2018.

OPEN source software for creating private and public clouds. 2018. Disponível em: <https://www.openstack.org/>. Acesso em: 12 jan 2018.

PYLVAINEN, I. **Getting started with RStudio Server**. 2016. Disponível em: <https://support.rstudio.com/hc/en-us/articles/234653607-Getting-Started-with-RStudio-Server>. Acesso em: 12 jan 2018.

SELF-SERVICE network. Disponível em: <https://docs.openstack.org/mitaka/install-guide-ubuntu/launch-instance-networks-selfservice.html>. Acesso em: 12 dez. 2018.

THE R FOUNDATION. **The R Project for statistical computing**. 2018. Disponível em: <https://www.r-project.org/>. Acesso em: 12 jan 2018.

TIERNEY, L.; ROSSINI, A. J.; LI, N.; SEVCIKOVA, H. **Package 'snow'**. 2018. 9 p. Disponível em: <https://cran.r-project.org/web/packages/snow/snow.pdf>. Acesso em: 12 jan 2018.

Exemplares desta edição
podem ser adquiridos na:

Embrapa Informática Agropecuária
Av. Dr. André Tosello, 209 - Cidade Universitária
Campinas, SP, Brasil
CEP. 13083-886
Fone: (19) 3211-5700

www.embrapa.br
www.embrapa.br/fale-conosco/sac

1ª edição
Versão digital (2018)

CGPE 14962