

# MIDDLEWARE PARA A ORQUESTRAÇÃO DA GERÊNCIA DE RECURSOS E SERVIÇOS EM NUVEM

Renan Freire Tavares

Orientador: Prof. Dr. Magnos Martinello

Coorientador: Prof. Dr. Maxwell Eduardo Monteiro



UFES

# Agenda

- Panorama da Gerência de Recursos na Computação em Nuvem
- Objetivos
- Implementação
- Testes
- Conclusões

# Computação em Nuvem

- Definição do NIST
  - Características
    - Serviço automático sob demanda
    - Amplo acesso a rede
    - Conjunto de recursos
    - Flexibilidade dos recursos
    - Serviços Otimizados
  - Modelo de serviços
    - IaaS, PaaS, SaaS, XaaS
  - Modelos de desenvolvimento
    - Nuvem: Privada, Comunitária, Pública, Híbrida

# Plataforma IaaS

- Openstack
- Principais módulos utilizados neste trabalho
  - Keystone
  - Glance
  - Neutron
  - Nova
  - Horizon
- Arquitetura
  - Controlador
  - Computação
  - Armazenamento
  - Rede

# Gerência em um Ambiente em Nuvem

- Propósitos
  - Comportamento planejado.
- Características desejáveis aos Sistemas de Gerência (Aceto et al, 2013)
  - Escalabilidade
  - Flexibilidade
  - Adaptabilidade
  - Conveniência
  - Autonomia
  - Abrangência
  - Extensibilidade
  - Baixa Intromissão
  - Resiliência
  - Confiabilidade
  - Disponibilidade
  - Precisão

# Sistemas de Gerência de Recursos

- Nagios

- Plugins

- 

| Retorno do Plugin | Estado Preliminar        |
|-------------------|--------------------------|
| <i>OK</i>         | <i>UP</i>                |
| <i>WARNING</i>    | <i>UP</i> ou <i>DOWN</i> |
| <i>UNKNOWN</i>    | <i>DOWN</i>              |
| <i>CRITICAL</i>   | <i>DOWN</i>              |

- Nmap

- Escaneamento de portas

# Problemas dos Sistemas de Gerência

- Compreensão da Topologia
  - Dependências
    - Máquina Física
    - Máquina Virtual
    - Aplicações
- Inferências com o reconhecimento completo da nuvem
- Auto detecção da Nuvem (FATEMA et al., 2014)
  - Grande volume de dados

# Objetivo

- Middleware para Orquestração da Gerência de Recursos
  - Levantamento da topologia
  - Integração de dados através de um banco de dados orientado a grafo
  - Integração das Aplicações de Gerência

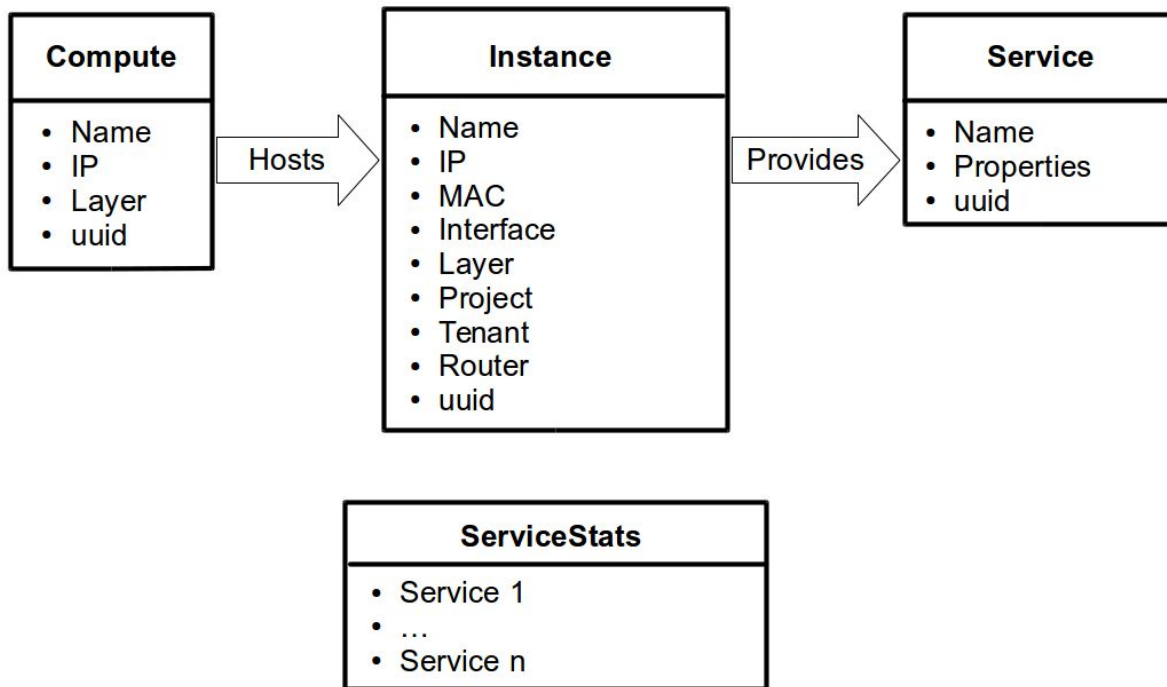


# Banco de Dados Orientado a Grafo

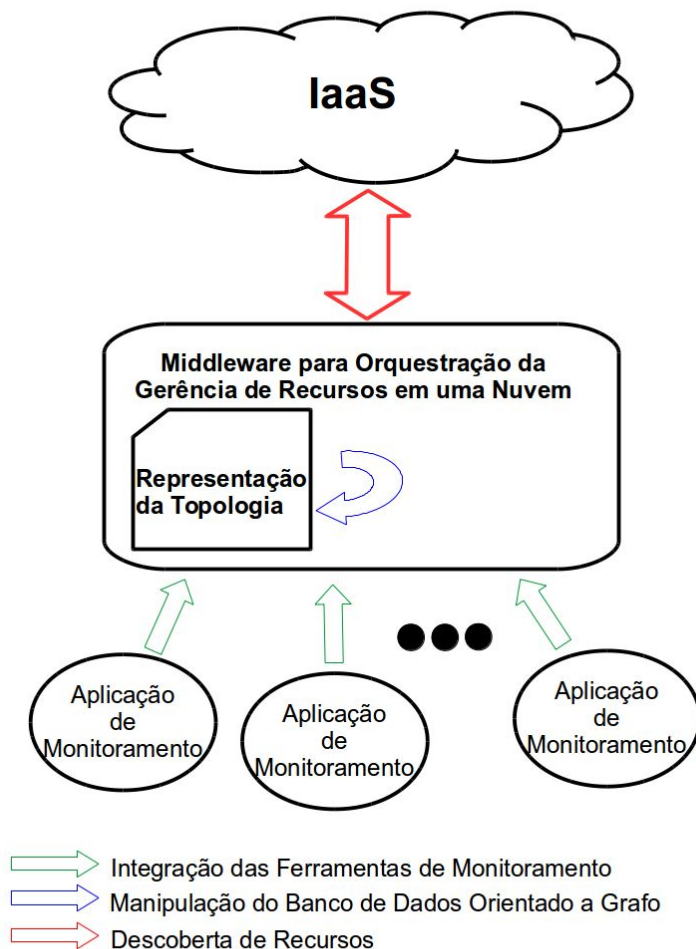
- Representação única da topologia da infraestrutura
  - Grafo
  - Representação Semântica
- Catálogo de Ativos
- Visualização das dependências
- Neo4j
  - SGBD orientado a grafo robusto
  - Interface Web
  - Linguagem: Cypher
  - Conexão
    - JDBC
      - URL
        - Ex.: *https://localhost:7474*

# Banco de Dados Orientado a Grafo

- Modelo de dados

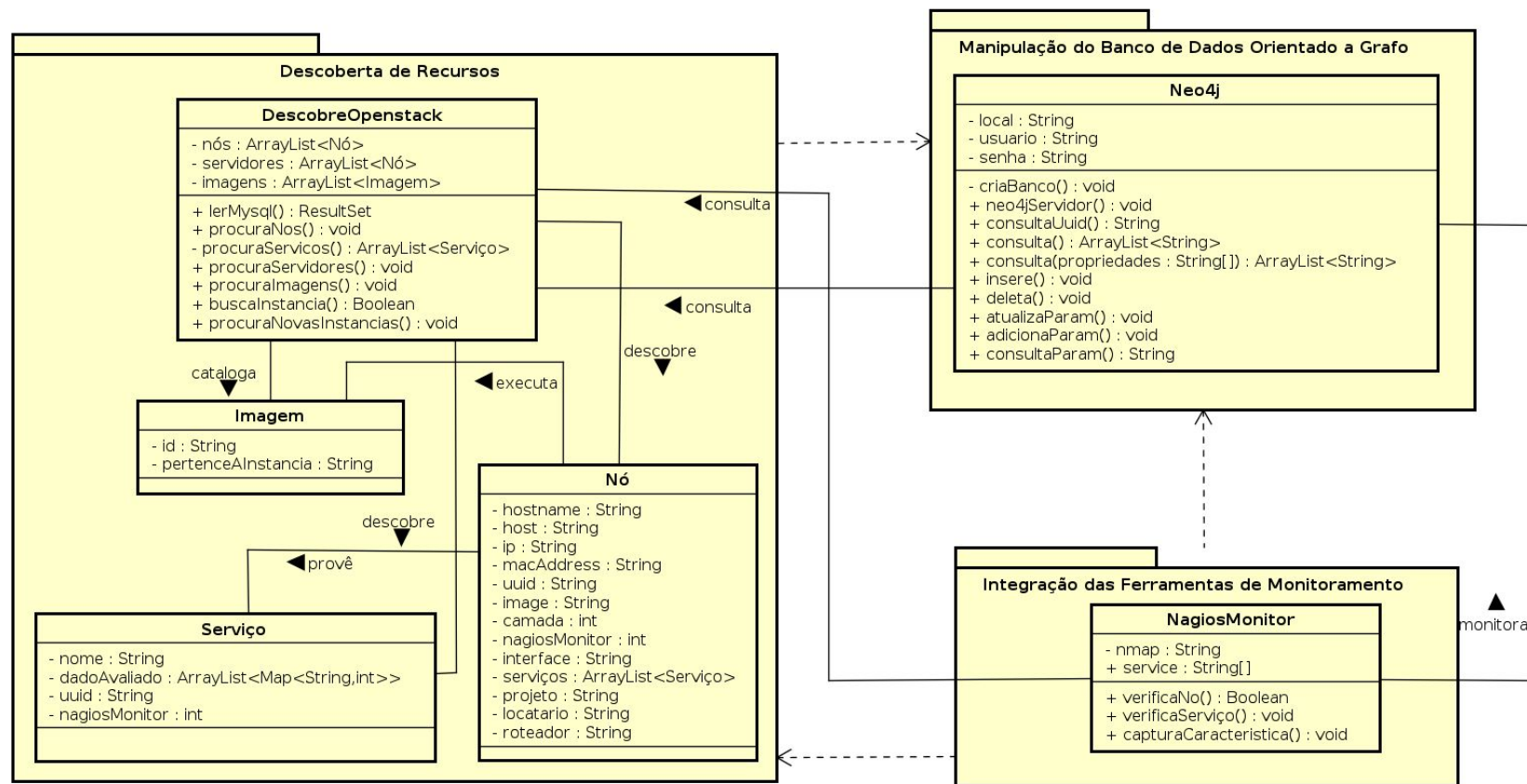


# Arquitetura da Solução



# Diagrama de Classes

pkg



# Descoberta de Recursos

- Coleta de dados da base dados
  - Função *lerMysql*

```
//Propriedades da Conexão
Properties connectionProps = new Properties();
connectionProps.put("user", user);
connectionProps.put("password", passwd);

// Carrega o Driver MySQL
Class.forName("com.mysql.jdbc.Driver");

// Configuração da Conexão com o Banco de Dados
connect = DriverManager.getConnection(
    "jdbc:" + "mysql" + "://" +
    "localhost" +
    ":" + "3306" + "/" + database,
    connectionProps);

// Statements Permitem Emitir Consultas SQL ao Banco de Dados
statement = connect.createStatement();

// Resultset Captam o Resultado da Consulta SQL
resultSet = statement.executeQuery(selectSql);
return resultSet;
```

# Descoberta de Recursos

- Execução a partir do Nó controlador
  - Função *procuraServidores*

```
//Consulta SQL para a Coleta das Informações dos Nós de Computação  
String selectSql = "select host, host_ip from compute_nodes";
```

- Função *procuraImagens*

```
//Consulta SQL para a Coleta das Informações das Imagens pelo Método procuraImagens  
String selectSql = "select image_id, value from glance.image_properties "+  
    "where name='instance_uuid';";
```

- Função *procuraNos*

```
//Consulta SQL para a Coleta das Informações dos Nós Feita pelo Método procuraNos  
String selectSql = "select host, hostname, uuid, image_ref, project_id,"+  
    "user_id, network_info from instances i "+  
    "join instance_info_caches ic on i.uuid = ic.instance_uuid "+  
    "where vm_state='active';";
```

# Descoberta de Recursos

- Método *procuraServicos*

```
//Execução da Biblioteca NMap4j do Java
Nmap4j nmap4j = new Nmap4j ( "/usr" );
nmap4j.addFlags ( "-T3 -sV" );
nmap4j.includeHosts( ip );
nmap4j.execute();
```

- Método *buscaInstancia*

```
//Consulta SQL para a Coleta do UUID das Instancias pelo Método buscaInstancia
String selectSql = "select uuid from instances "+
    "where uuid='"+uuid+"' AND vm_state='active'";
```

- Método *procuraNovasInstancias*

```
//Consulta SQL para a Coleta do UUID das Instancias ativas pelo Método procuraNovasInstancias
String selectSql = "select uuid from instances "+
    "where vm_state='active'";
```

# Manipulação do Banco de Dados Orientado a Grafo

- Função *insere*
  - JDBC

```
// Configuração da Conexão com o Banco de Dados
Neo4jConnection con=null;
try {
    con = new Driver().connect(local, props);
} catch (SQLException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}

// Resultsets Captam o Resultado da Consulta Cypher
ResultSet rs;
try {
    rs = con.createStatement().executeQuery(query);
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```



# Manipulação do Banco de Dados Orientado a Grafo

- Função *criaBanco*

- Criação do nó de computação

```
insere(["CREATE (a:Compute{Name:'"+server.getHostname()+  
    "','IP:'"+server.getIp()+  
    "','Layer:'"+server.getCamada()+  
    "','uuid:'"+server.getUuid()+"}")"]);
```

- Criação das Instâncias

```
insere("CREATE (a:Instance{Name:'"+virtual.getHostname()+  
    "','IP:'"+virtual.getIp()+ "','MAC:'"+virtual.getMac()+  
    "','Interface:'"+virtual.getInterface()+  
    "','Layer:'"+virtual.getCamada()+ "','uuid:'"+virtual.getUuid()+  
    "','Project:'"+virtual.getProject()+  
    "','Tenant:'"+virtual.getTenant()+  
    "','Router:'"+virtual.getRouter()+"}")");
```

- Criação do relacionamento entre Computação e Instância

```
insere("MATCH (a:Compute),(b:Instance) WHERE a.uuid='"+server.getUuid()+  
    "' AND b.uuid='"+virtual.getUuid()+"' CREATE (a)-[r:Hosts]->(b)");
```

# Manipulação do Banco de Dados Orientado a Grafo

- Função *criaBanco*

- Criação dos Serviços

```
insere("CREATE (a:Service{Name:''+service.getName()+  
'',uuid:''+service.getUuid()+''})");
```

- Criação do Relacionamento entre Instância e Serviço

```
insere("MATCH (a:Instance),(b:Service) WHERE a.uuid='' +virtual.getUuid()+  
'' AND b.uuid='' +service.getUuid()+'' CREATE (a)-[r:Provides]->(b)");
```

- Criação do nó *ServiceStats*

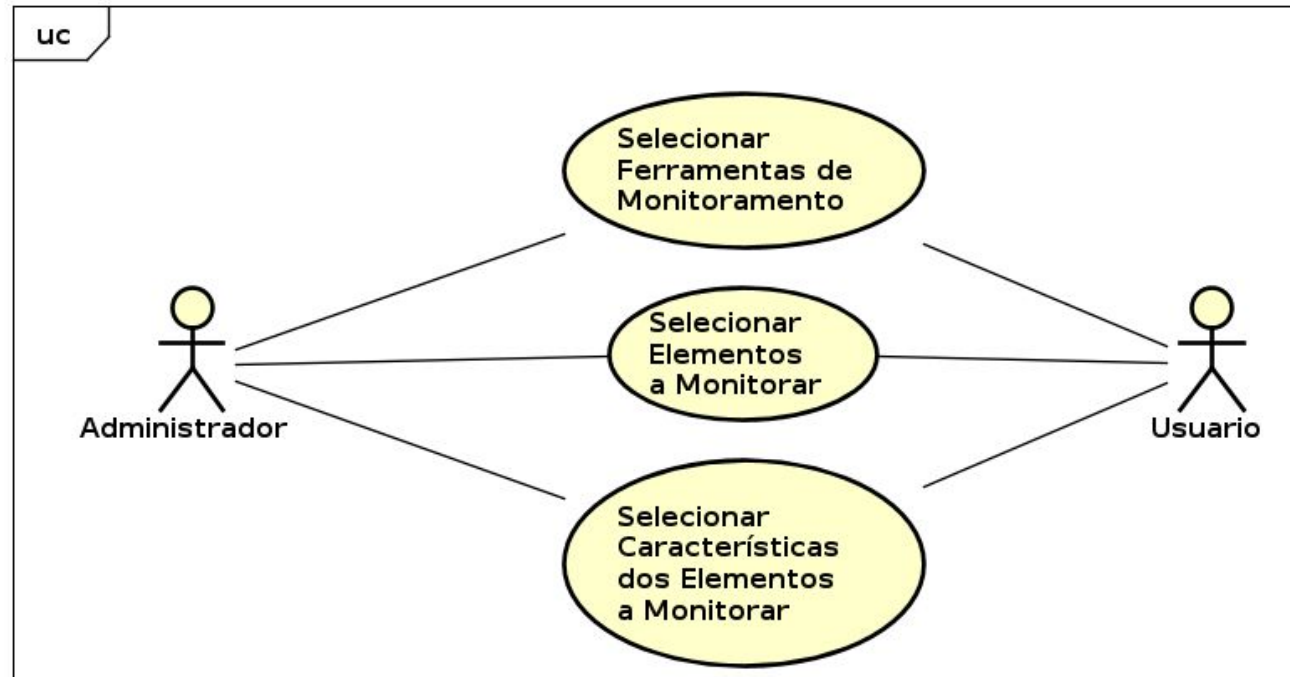
```
insere("create(n:ServiceStats)");
```

# Integração das Ferramentas de Monitoramento

- Métodos para consulta ao Neo4j
  - Leitura
    - Funções consulta, consulta com sobrecarga, consultaParam e consultaUuid
      - Ex.: ***MATCH (n:"+RÓTULO+") RETURN n." + PROPRIEDADE***
  - Escrita
    - Métodos deleta, adiciona e adiciona com sobrecarga
      - Ex.: ***MATCH ()-[r1:"+RÓTULO+"]-(n:"+RÓTULO+") MATCH (n:"+RÓTULO+")-[r2:"+RÓTULO+"]-(s:"+RÓTULO+") WHERE "+CONDIÇÃO+" DELETE n,r1,r2,s***
      - Ex.: ***MATCH (n:"+RÓTULO+") WHERE "+CONDIÇÃO+" SET n." + PROPRIEDADE + "=' "+NOVOVALOR+"'***

# Integração das Ferramentas de Monitoramento

- Orquestração da Gerência de Recursos



# Integração das Ferramentas de Monitoramento

- NagiosMonitor
  - Plugins do Nagios
  - Funções
    - verificaNo
      - Tolerância
      - Evento de Tratamento
    - verificaServico
      - Tolerância
      - Evento de Tratamento
    - capturaCaracterística
      - Acesso via SSH

# Ambiente de Testes

- Configurações do Ambiente
  - Processador: Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz.
  - Memória (RAM): 4GB.
  - Disco Rígido: Capacidade - 750GB.
  - Sistema Operacional: Ubuntu 14.04.
  - Rede local.
  - Neo4j 2.2.5.
  - Openstack Kilo.
  - Nagios Core 3.

# Devstack

- Rede Local

```
###NETWORK
HOST_IP=10.0.0.1
SERVICE_HOST=10.0.0.1
MYSQL_HOST=10.0.0.1
RABBIT_HOST=10.0.0.1
GLANCE_HOSTPORT=10.0.0.1:9292

# Do not use Nova-Network
disable_service n-net

# Enable Neutron
ENABLED_SERVICES+=,q-svc,q-dhcp,q-meta,q-agt,q-l3
```

# Devstack

- Opções do Neutron e Open vSwitch

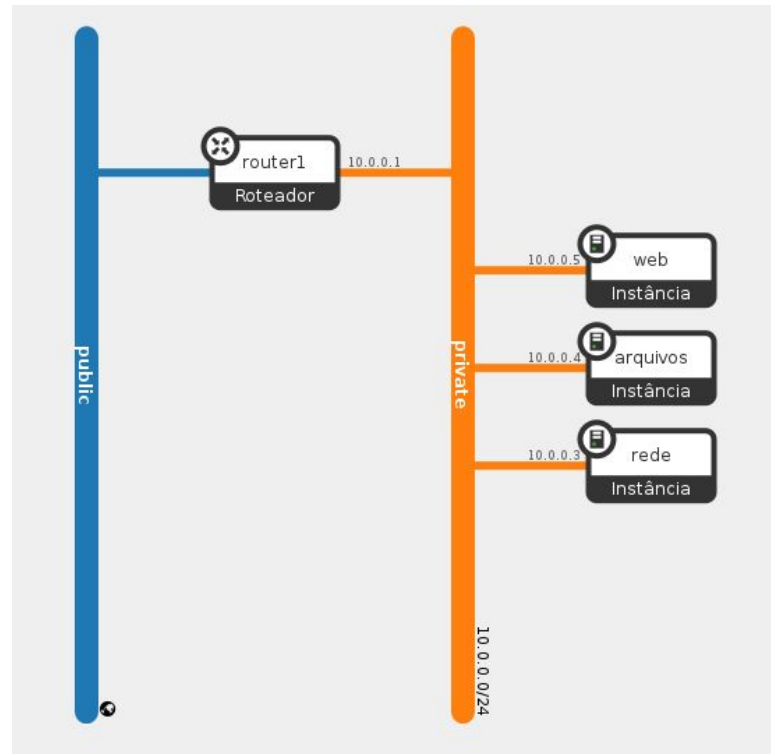
```
## Opções do Neutron |
Q_USE_SECGROUP=True
FLOATING_RANGE="10.0.10.0/24"
FIXED_RANGE="10.0.0.0/24"
Q_FLOATING_ALLOCATION_POOL=start=10.0.10.40,end=10.0.10.50
PUBLIC_NETWORK_GATEWAY="10.0.10.1"
Q_L3_ENABLED=True
PUBLIC_INTERFACE=p5p1

# Open vSwitch provider networking configuration
Q_USE_PROVIDERNET_FOR_PUBLIC=True
OVS_PHYSICAL_BRIDGE=br-ex
PUBLIC_BRIDGE=br-ex
OVS_BRIDGE_MAPPINGS=public:br-ex
```



# Horizon

- Configuração do Ambiente
  - Criação de Instâncias



# Orquestrador

- Descoberta de Recursos
- Escrita no banco de dados orientado a grafo

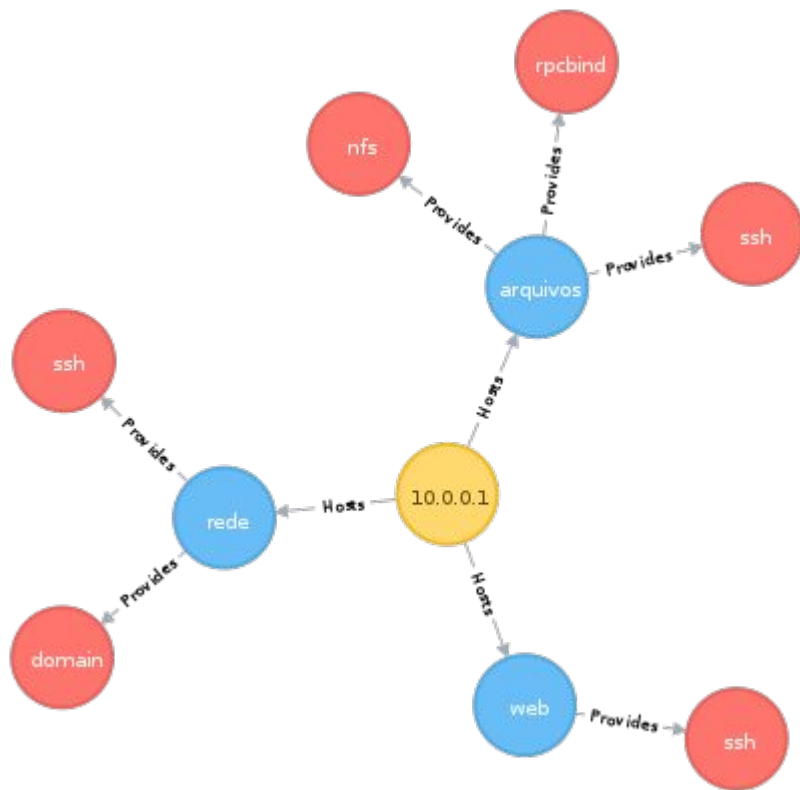
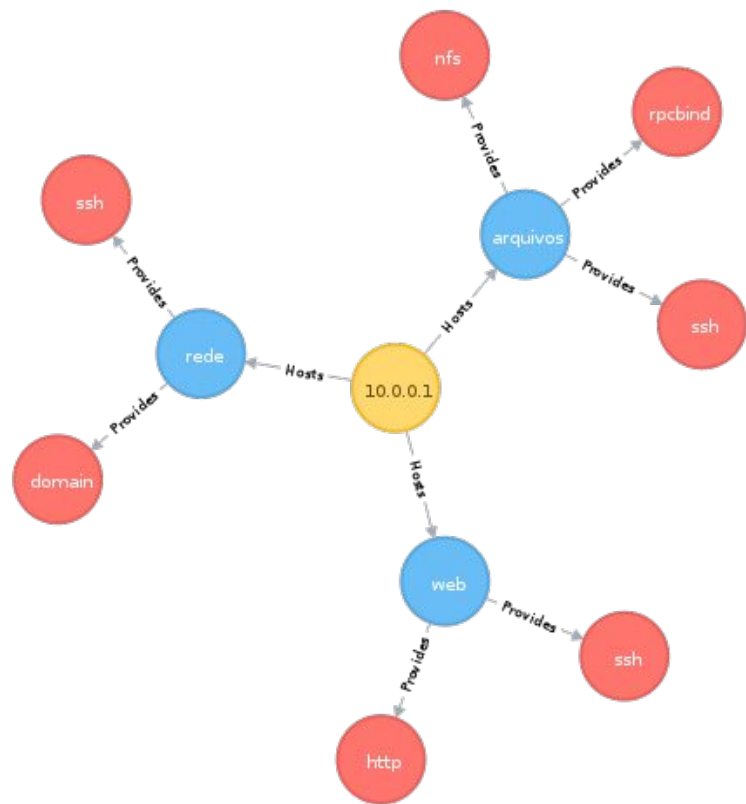
```
public class Orquestrador {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Selecione a IaaS...");  
        System.out.println("No momento, somente OPENSTACK é suportado");  
        new DescobreOpenstack();  
        Neo4j.neo4jServidor("start");  
        new Neo4j();  
        System.out.println("NAGIOS MONITOR");  
        //A execução do programa termina com um ctrl+c  
        createShutdownHook();  
    }  
}
```

# Orquestrador

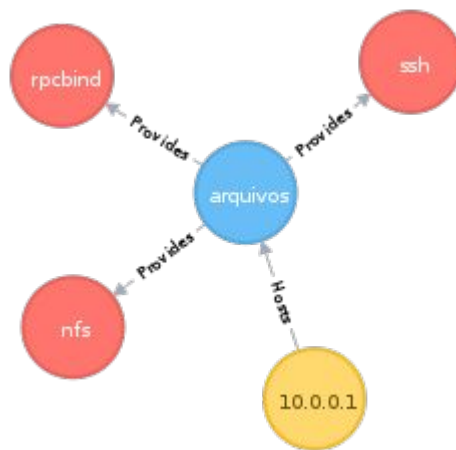
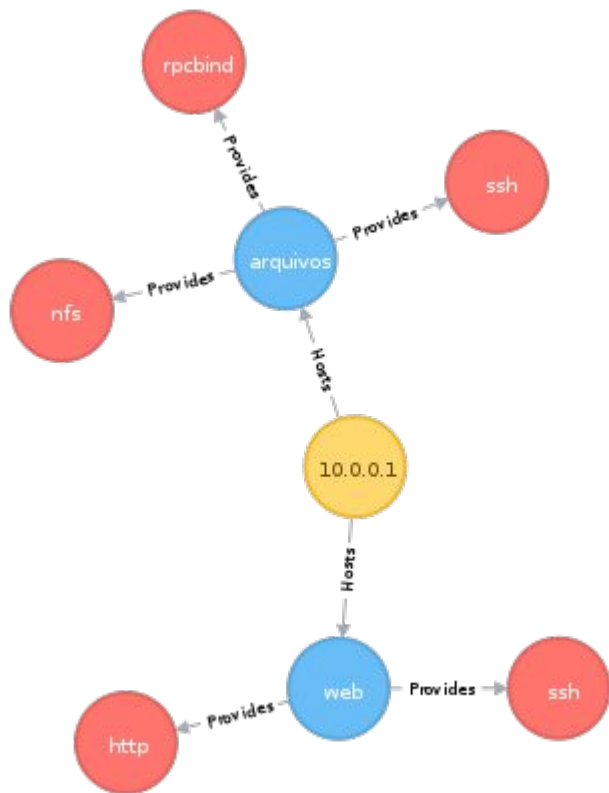
- Integração das Ferramentas de Monitoramento
  - Possibilidade da orquestração

```
while(true){
    DescobreOpenstack.procuraNovasIntancias();
    try {
        //Executa nagios monitor
        ArrayList <String> ips;
        ips= Neo4j.consulta("INSTANCE", "IP");
        for(String ipNeo4j: ips)
            new NagiosMonitor(ipNeo4j);
        Thread.sleep(30000); //1000 milisegundos = 1 segundo
    }catch (InterruptedException ex) {
        Thread.currentThread().interrupt();
    }
}
```

# Testes



# Testes



# Conclusão

Com o middleware desenvolvido foi possível criar um ambiente de orquestração da gerência de recursos. Assim, as aplicações não necessitam do esforço de reconhecer a topologia, ao mesmo tempo que, com os dados integrados ao banco de dados orientado a grafo, podem tomar inferências mais precisas.

# Limitações e Trabalhos Futuros

- Eficiência em grande escala
- Detecção automática do tipo de IaaS
- Técnicas auxiliares a descoberta de recursos
- Detalhamento mais completo da topologia
- Utilização de variável para indicação do estado do recurso
- Restrições ao Banco de dados orientado a grafo

# Código do Projeto

- <https://github.com/orenanft/projetoFinal/>



# Agradecimentos

Os conhecimentos adquiridos através do curso de Ciência da Computação foram imprescindíveis para a realização deste trabalho. Assim, gostaria de agradecer a todos os professores e colegas que me ajudaram ao longo deste caminho. De forma especial, desejo agradecer aos integrantes desta banca, pela atenção e contribuição ao trabalho.

Obrigado!!!