



# Aplicando JClouds para Computação em Alto Desempenho

II Escola Regional de Alto Desempenho de São Paulo  
ERAD-SP 2011

Orientador: Prof. Dr. Cesar A. C. Marcondes

Aluno: Bruno Katekawa

Curso: Engenharia da Computação

# JClouds

~ Multi-cloud library ~



# jclouds

multi - cloud library

Compute



BlobStore



# API Compute



- \* É um meio portátil de gerenciar nós nas nuvens, isto é, através dela podem-se executar vários nós como um conjunto, independentemente da API da nuvem subjacente.
- \* Possui um recurso de criação de modelos (Templates) que permite a busca de configurações que correspondem aos parâmetros como núcleos de CPU, memória RAM ou sistema operacional, permitindo uma alta customização durante a criação de um nó

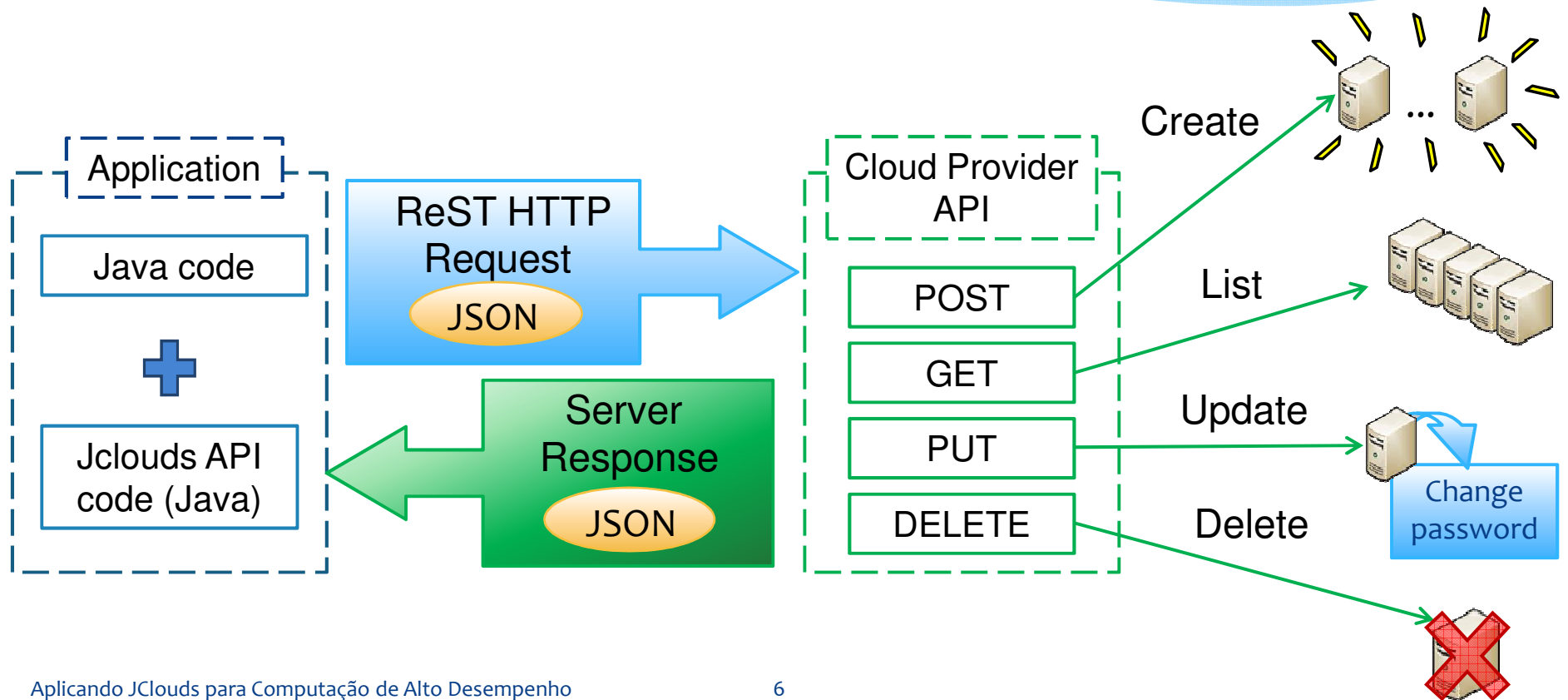
# API BlobStore



- \* É um mapa de chave-valor, onde há uma conta existente e pode-se criar recipientes (Containers) para armazenar dados como um objeto binário cada um sendo referenciado por um nome.
- \* É um meio portátil que os provedores de armazenamento em nuvem utilizam para gerenciar mapas de chave-valor, como o Microsoft Azure Blob Service, Amazon S3 e Rackspace Cloud Files.

# Modelo de Funcionamento da API

~JClouds API ~



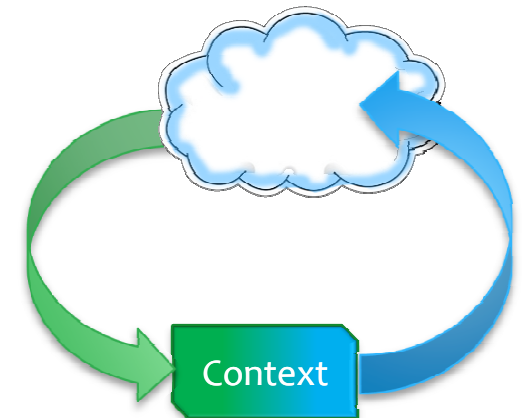
# JClouds API

~ Criando o contexto ~

```
ComputeServiceContext context =  
new ComputeServiceContextFactory().createContext(  
"cloudservers-us", wiring, overrides  
);
```

```
CloudServersClient rackspaceClient =  
CloudServersClient.class.cast(  
context.getProviderSpecificContext().getApi()  
);
```

```
BaseComputeService client =  
BaseComputeService.class.cast(context.getComputeService());
```



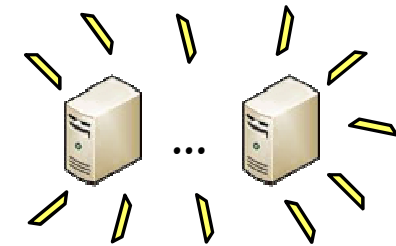
# JClouds API

~ Criando Servidores ~

```
Template nodeToCreate = context.getClient().templateBuilder()  
.osFamily(OsFamily.UBUNTU)  
.osVersionMatches("10.10")  
.minCores(2)  
.minRam(minRam)  
.options(options).build();
```

```
System.out.println("Creating nodes ...");
```

```
Set<? extends NodeMetadata> nodes =  
context.getClient().createNodesInGroup(  
nodeGroupName,  
numberOfNodes,  
nodeToCreate);
```





# JClouds API

~ Outras operações ~

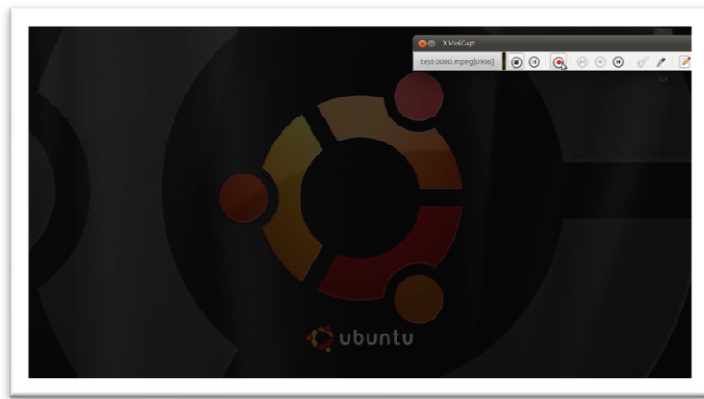
```
context.getClient().rebootNode(node);
```

```
context.getRackspaceClient().resizeServer(serverid, 3);
```

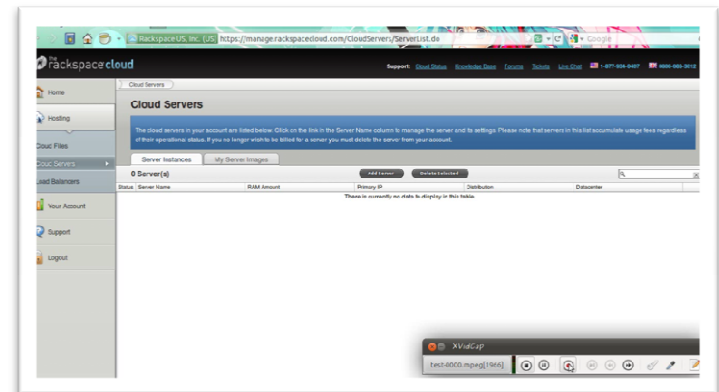
```
context.getClient().destroyNode(nodeToDelete);
```



# API em Ação



Criando Servidores  
(Nós) no Rackspace



Mostrando Dashboard  
do Rackspace

# API em Ação

```
root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

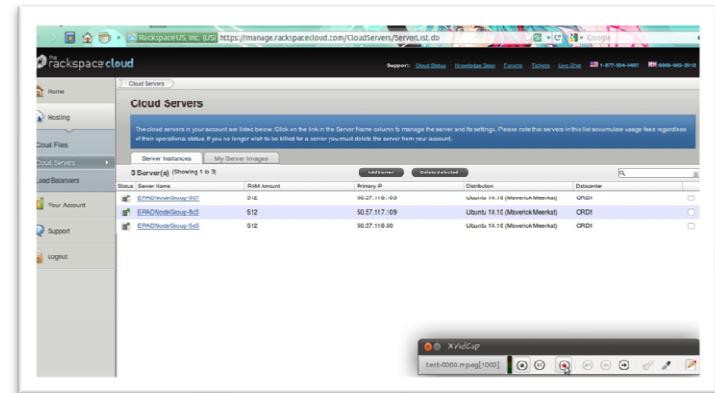
root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help

root@ubuntu:~# rcli --help
rcli 0.0.1
  do you wish?
  * Create
  * List
  * Delete
  * Miscellaneous
  * Exit Program
  --help
```

Apagando Servidores (Nós) no Rackspace



Mostrando Dashboard do Rackspace

# Referências

- \* Site do Projeto da API JClouds, <http://code.google.com/p/jclouds/>, acesso: 22 de Maio de 2011.
- \* Rackspace Hosting, <http://www.rackspace.com/cloud/>, acesso: 25 de Julho de 2011.

# Perguntas?



Contato:  
[brunokatekawa@gmail.com](mailto:brunokatekawa@gmail.com)  
[marcondes@dc.ufscar.br](mailto:marcondes@dc.ufscar.br)