

SCALR

Gerenciamento de infraestrutura escalável para websites

Fabiano Castro Pereira
fabiano.pereira@serpro.gov.br



Ligado nas
Comunidades

 **SERPRO**
Serviço Federal de Processamento de Dados

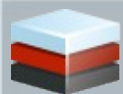


- **Introdução**
- **Funcionamento do Scalr**
 - IaaS
 - Farm
 - Role
 - Behavior
- **Funcionamento específico**
 - Apache com balanceamento
 - MySQL com replicação
- **Conclusão**



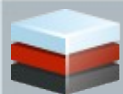


- **Objetivo do Scalr**
 - Gerenciar automaticamente a quantidade de servidores
 - Acrescenta ou retira servidores virtuais em nuvem de acordo com a variação na carga da aplicação
- **Componentes controláveis**
 - Servidor web
 - Banco de dados
 - Balanceador de carga
 - Servidor de cache
 - Serviço personalizado (via programação)





- **Formas de uso**
 - Instalação em servidor próprio
 - Download e procedimentos de instalação: <http://scalr.net>
 - Software as a Service
 - Pode-se utilizar as funcionalidades do Scalr no próprio site da ferramenta, através de assinatura do serviço
 - Custo de US\$ 99 por mês



Funcionamento



- IaaS suportadas



- Maior integração é com a Amazon

- Métricas para mudança na infraestrutura

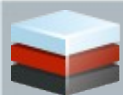
- Agente **Scalarizr** é instalado em cada VM e monitora:
 - CPU
 - Memória
 - Rede
- VMs (des)ativadas de acordo com métrica escolhida





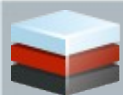
- **Componentes**

- Farm: conjunto de VMs, organizadas em Roles
- Role: cada role cumpre uma função na Farm
 - Aplicação (Apache, Tomcat)
 - Banco de dados (MySQL, PostgreSQL, Cassandra, Redis)
 - Balanceamento (nginx)
 - Cache (Memcached)
 - Mixed (mais de um role simultaneamente)
- Behavior
 - Configurações personalizadas para o role escolhido
 - Definições de como o role responde a eventos





- **Configurações genéricas do Behavior**
 - Número mínimo e máximo de VMs
 - Opções de escalação (\neq escalonamento)
 - Time: permite agendar a mudança de tamanho
 - Bandwidth: % de uso da rede
 - Load average: % de uso da CPU
 - Queue size: tamanho da fila de processamento de mensagens (quando a aplicação hospedada utiliza Amazon SQS)
 - RAM: quantidade de memória livre
 - Local e tipo de VM
 - Locais (AWS): EUA, Europa e Ásia
 - Tipo: quantidade de memória, CPU e disco



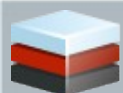


Funcionamento

- Tipo e custo mensal de VMs Amazon (US East)

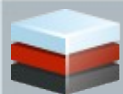
Nome	Mem	ECU*	Disco	US\$
Micro	0.6 G	2	EBS*	14
Standard Small	1.7 G	1	160 G	60
Standard Large	7.5 G	4	850 G	230
L→ RDS MySQL - 1 ou 3 Anos	7.5 G	4	850 G	76 / 39
High-Memory Extra	17.1 G	6.5	420 G	350
L→ RDS MySQL - 1 ou 3 Anos	17.1 G	6.5	420 G	110 / 55
High-CPU Medium	1.7 G	5	350 G	120
High-CPU Extra	7 G	20	1690 G	490

- 1 ECU (Elastic Compute Unit) \approx 1 Xeon 1 GHz (2007)
- EBS (Elastic Block Store) é um disco virtual
- **Elasticidade custa caro!** Servidores dedicados similares custam a metade



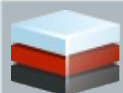


- **Configurações específicas para AWS**
 - Balanceamento HTTP
 - Pode utilizar **Elastic Load Balancer** em vez de um role específico para um balanceador HTTP (nginx, etc.)
 - Volumes EBS (Elastic Block Store)
 - Definição de tamanho do disco virtual, ponto de montagem, e se será utilizado um snapshot salvo anteriormente
 - Métricas
 - Utilização do **Amazon CloudWatch** para obtenção de métricas sobre as VMs
 - Nuvem privada
 - Pode-se configurar os parâmetros do **Virtual Private Cloud**



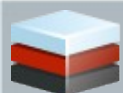


- Configurações específicas por role
 - Parâmetros do Apache
 - Diversas configurações, principalmente relacionados a Virtual Host, permitindo que vários websites sejam hospedados
 - Configurações específicas para MySQL
 - Criação de snapshots do volume EBS
 - Snapshots são usados para replicação da base para slaves
 - Definir periodicidade e quantos guardar antes do descarte
 - Backup
 - O objetivo aqui é diferente do snapshot
 - Visa ter uma cópia completa dos dados utilizando **mysqldump**



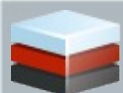


- Instalação de aplicações hospedadas
 - Duas opções
 - Usar ferramenta **Scalr-tools**, caso a aplicação esteja em um repositórios de código (svn, git, etc.)
 - Criar cópia da máquina virtual
 - Usando Scalr-tools
 - Criar cópia do código localmente (desktop do desenvolvedor)
 - Registrar e publicar aplicação na farm
 - Criando cópia da VM
 - Conectar na VM via SSH e fazer instalação da aplicação
 - Acessar os dados da VM na interface do Scalr e escolher a opção **Create server snapshot**





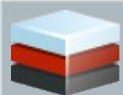
- **Atualização de aplicações hospedadas**
 - Como atualizar se podem existir cópias da aplicação em VMs que estão desligadas?
 - Scalr permite associar scripts a determinados eventos
 - Scripts rodam em cada VM do role configurado
 - Pode-se associar um script de atualização da aplicação ao evento de inicialização de host
 - Existem eventos relativos à:
 - Inicialização e encerramento de VM
 - Mudança de endereço IP
 - Volumes EBS (Amazon)
 - Mudanças no DNS



Apache com balanceamento



- **Role Apache**
 - Pode ser usado como servidor web, ou de aplicações PHP, Ruby, Python, etc.
 - Para aplicações Java recomenda-se a role Tomcat
- **Configurações necessárias**
 - Dados de Virtual Host (podendo servir vários sites)
- **Dimensionamento da nuvem**
 - Ao verificar a necessidade de nova VM, o Scalr inicia a VM e acrescenta o novo Apache no balanceador
 - Quando a carga diminui, o Scalr retira um integrante do balanceador e desliga a VM escolhida





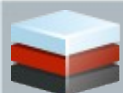
- **Role MySQL**

- Utiliza a funcionalidade de replicação do MySQL

- Opera com esquema master/slave
 - Dados são gravados apenas no master, e copiados para cada slave que esteja configurado
 - Slaves só podem ser lidos (SELECT)

- Armazenamento

- Os dados da base são armazenados em volumes EBS
 - Isto é útil para o início rápido de slaves, no momento de acrescentar servidores ao cluster





- **Role MySQL**

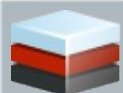
- Acrescentando um slave

- Scalr inicia VM
 - Monta volume EBS copiando o último snapshot do master
 - Configura replicação no slave, informando local do master

- **Failover**

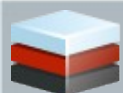
- Caso o master se torne indisponível

- Scalr detecta e promove um slave para que seja novo master
 - Volume EBS do master antigo é reutilizado





- **Ferramenta muito boa**
 - Várias aplicações suportadas
 - Fácil configuração
 - Poupa muito trabalho
- **Melhorias**
 - <http://scalr.uservice.com/>
 - Em andamento: CloudStack, OpenStack, CloudFoundry, MongoDB, Chef
 - Planejado: Alertas, RAID com EBS, FarmTemplates (MediaWiki, Drupal, Wordpress, etc.), Dashboard widgets
 - JBoss?





Perguntas?

fabiano.pereira@serpro.gov.br

