

TREXCLOUD: Facilitando a Implantação de Aplicações Java Enterprise em Nuvens IaaS

Roberto Costa Lima Jr^{1,2}, Américo Sampaio¹, Nabor Mendonça¹

¹Programa de Pós-Graduação em Informática Aplicada (PPGIA)

Universidade de Fortaleza (UNIFOR)

Av. Washington Soares, 1321, Edson Queiroz, CEP 60811-905 Fortaleza, CE

americo.sampaio@unifor.br, nabor@unifor.br

²E-NOVAR Soluções Tecnológicas

Rua Costa Barros, 915 - 9º Andar, Centro, CEP 60160-280 Fortaleza, CE

roberto@enovar.com.br

Resumo. *Um dos grandes desafios que os atuais usuários de nuvens de infraestrutura como serviço (IaaS) enfrentam é a conhecida dificuldade para implantar e configurar suas aplicações. Este trabalho apresenta a ferramenta TREXCLOUD que automatiza a tarefa de implantar aplicações web baseadas na plataforma Java Enterprise Edition (JEE) em provedores de nuvem IaaS. O objetivo principal da ferramenta é proporcionar um ambiente intuitivo e de fácil utilização para que desenvolvedores e até usuários leigos possam implantar uma aplicação JEE na nuvem sem ter que conhecer tecnologias específicas de diversos provedores.*

Abstract. *One of the key challenges faced by current users of infrastructure-as-a-service (IaaS) clouds is the known difficult to deploy and configure their applications. This work presents the TREXCLOUD tool which automates the task of deploying web applications based on the Java Enterprise Edition (JEE) platform in providers of infrastructure-as-a-service (IaaS) clouds. The main goal of the tool is to offer an intuitive and easy-to-use environment in which developers and even non-expert users can deploy JEE applications in the cloud without having to know any provider-specific technology.*

1. Introdução

A computação em nuvem é um paradigma recente que vem transformando a forma de desenvolver, implantar e gerenciar software. Um uso popular de nuvens de infraestrutura (IaaS) dá-se no contexto de aplicações web desenvolvidas com a plataforma Java Enterprise Edition (Java EE) [Hajjat 2010]. Para implantar este perfil de aplicação na nuvem é necessário que todos os componentes da aplicação (ex: servidor web, servidor de aplicação, banco de dados e código) sejam empacotados em uma coleção de imagens de máquinas virtuais (VM) no formato específico da nuvem de escolha. Isso requer uma série de procedimentos de instalação e configuração para criar as imagens das VMs, configurar os componentes de software e configurar suas dependências. Alguns parâmetros de configuração só tornam-se conhecidos quando a nuvem inicializa a VM pois são obtidos dinamicamente, por exemplo, o endereço IP de uma VM que é atribuído pelo software controlador da nuvem via mecanismos como DHCP.

Além disso, cada provedor de nuvem IaaS possui seus próprios mecanismos de implantação, o que torna a tarefa de re-implantar aplicações em nuvens distintas um trabalho tedioso. Por exemplo, a Amazon oferece o seu mecanismo de criação de imagens proprietário, chamado Amazon Machine Image (AMI). Caso a organização desenvolvedora da aplicação decida trocar de provedor para a nuvem Rackspace, a implantação daquela aplicação deverá ser refeita quase que em sua totalidade.

Por causa dos motivos supracitados, algumas ferramentas têm sido propostas ao longo dos últimos anos, especificamente destinadas a enfrentar o desafio de implantação em nuvens IaaS. A maioria dessas ferramentas [Rodero-Merino et al. 2010, Binz et al. 2012, Heroku 2012, CloudFoundry 2012] tem um escopo amplo e permite a implantação de qualquer tipo de aplicação na nuvem. Em geral, essas ferramentas são mais difíceis de usar, pois se baseiam em linguagens *script* e, portanto, são mais indicadas para administradores de sistema experientes. Por outro lado, existem ferramentas com um âmbito mais restrito, como a Amazon Elastic Beanstalk [Beanstalk 2012], que foca em fornecer suporte para a implantação de aplicações web. No entanto, esta ferramenta sofre duas limitações principais: não suporta a configuração do banco de dados de forma automática e só permite a implantação na nuvem da Amazon.

Este trabalho apresenta a ferramenta TREXCLOUD, que permite a implantação e configuração de aplicações Java EE completas (incluindo todos os componentes de software) em diferentes nuvens IaaS de uma maneira simplificada, sem a necessidade de executar procedimentos manuais e complexos. A ferramenta atualmente suporta dois provedores de nuvem IaaS (Amazon e Rackspace) e oferece mecanismos de extensibilidade para programaticamente introduzir novos provedores. Uma avaliação empírica da ferramenta apresenta resultados promissores, indicando que usuários com variados níveis de experiência com tecnologia da informação conseguiram rapidamente (em poucos minutos) implantar uma aplicação Java EE completa na nuvem da Amazon. Os resultados demonstraram que os usuários conseguem concluir a tarefa de implantação, em média, muito mais rápido do que quando utilizando as ferramentas fornecidas pela própria Amazon.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute as principais etapas e dificuldades envolvidas na implantação de aplicações em nuvens IaaS. A Seção 3 apresenta a ferramenta TREXCLOUD e seus mecanismos de apoio ao processo de implantação na nuvem. A Seção 4 descreve o estudo empírico conduzido para avaliar a ferramenta. A Seção 5 compara a ferramenta com diversos trabalhos relacionados. Por fim, a Seção 6 oferece algumas conclusões e direções para trabalhos futuros.

2. Implantação em Nuvens IaaS

O processo de implantação de aplicações em nuvens IaaS pode ser organizado em duas etapas principais: (i) criação de imagens e (ii) instalação e configuração de software.

2.1. Criação de Imagens

Esta etapa consiste em criar as imagens das máquinas virtuais que vão ser utilizadas para instalar os componentes da aplicação. Por exemplo, no caso de uma aplicação Java EE típica, uma configuração mínima envolveria a criação de duas imagens: uma para o servidor de aplicações e uma para o banco de dados.

A imagem do servidor de aplicação teria o *container* Java necessário para o software a ser entregue (ex: Spring, JBoss, Tomcat, etc.). Ao considerar uma nuvem IaaS específica, seria necessário construir uma imagem de VM a partir do zero (um processo manual custoso) ou reutilizar uma imagem pré-construída disponibilizada pelo provedor IaaS. Por exemplo, a Amazon oferece muitas AMIs públicas para vários *containers* conhecidos Java (em várias distribuições de sistema operacional) que podem ser personalizadas. Mesmo que a reutilização de uma imagem já existente seja mais fácil, o usuário ainda precisa seguir um conjunto de passos para registrar a imagem e configurar os certificados de segurança do provedor.

A imagem do banco de dados deve conter o sistema de gerenciamento de banco de dados específico utilizado pela aplicação que está sendo implantada (ex: MySQL, Postgres, etc.). Para facilitar esta tarefa, os provedores IaaS também oferecem imagens pré-construídas para os mais populares sistemas comerciais e *open source* de banco de dados em várias distribuições de sistema operacional.

2.2. Instalação e Configuração de Software

Depois que as imagens são criadas e registradas para uso no provedor IaaS, o próximo passo é instalar em cada imagem a pilha de software necessária à execução da aplicação que será implantada. Considerando a aplicação exemplo anterior, o usuário precisa executar os seguintes passos.

Na imagem do servidor de aplicação, o usuário precisa copiar e instalar a camada de apresentação e lógica de negócios da aplicação. Em geral, isso é feito iniciando-se a máquina virtual, e depois transferindo os arquivos (ex: arquivo *.war*) que contêm a lógica da aplicação. Se necessário, alguns arquivos de configuração do servidor de aplicação também terão de ser criados ou modificados. Na imagem do banco de dados, a configuração é similar. O usuário precisará criar um novo banco de dados a partir do zero, ou carregar um já existente através da cópia de um arquivo de *backup*. Além disso, esta é também a etapa em que algumas configurações de ajuste podem ser aplicadas ao banco de dados.

Além de configurar cada imagem, o usuário também tem que configurar as dependências entre cada componente. No exemplo acima, um passo comum é configurar o servidor de aplicação com o endereço IP do servidor de banco de dados. Vale ressaltar que este passo normalmente só pode ser feito depois que o servidor de banco de dados foi implantado e inicializado, pois máquinas virtuais em nuvens IaaS geralmente possuem endereços IP atribuídos dinamicamente. Portanto, em cada nova implantação, o servidor de banco de dados teria um novo endereço IP que deve ser informado durante a inicialização da VM que contém o servidor de aplicação para a correta implantação e configuração da aplicação como um todo.

3. TREXLOUD

TREXLOUD é uma ferramenta de implantação de software que visa proporcionar suporte para a implantação de aplicações web em nuvens IaaS. A ferramenta suporta atualmente a implantação de aplicações Java EE em dois conhecidos provedores de nuvem comerciais: Amazon e Rackspace. O objetivo da ferramenta é libertar o usuário de ter qualquer conhecimento específico sobre tecnologias proprietárias de nuvens IaaS, tais

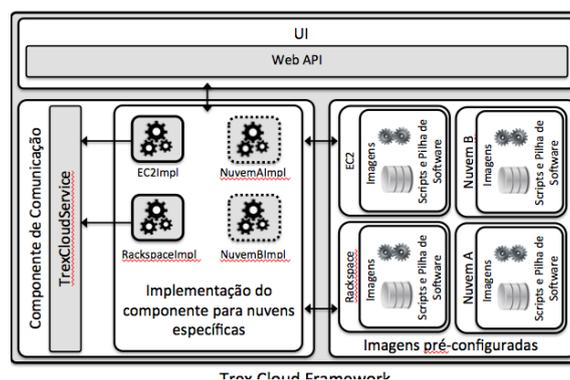


Figura 1. Visão geral da arquitetura da ferramenta TREXCLOUD.

como os exigidos para a criação das imagens das VMs e suas configurações. Além disso, o usuário não precisa ter nenhum conhecimento sobre linguagens de *script*, de modo que ele/ela pode implantar um aplicativo completo na nuvem com apenas alguns cliques. A Figura 1 mostra a arquitetura da ferramenta.

A ferramenta está estruturada em três componentes:

- *API Web* — Aplicativo web que recebe dos usuários todas as informações necessárias para a realização da implantação e repassa as informações de forma estruturada para o componente seguinte.
- *Componente de comunicação com a nuvem* — Componente responsável por executar os comandos necessários para realizar a implantação da aplicação no provedor de nuvem escolhido. A ferramenta poderá disponibilizar múltiplas instâncias desse componente, sendo cada instância responsável pela comunicação com um provedor de nuvem específico.
- *Imagens de instâncias pré-configuradas* — Imagens de instâncias específicas de cada provedor de nuvem IaaS com as configurações básicas necessárias para a implantação automatizada de componentes típicos da plataforma Java EE (ex.: servidor de aplicação, banco de dados) e seus respectivos *scripts* de configuração.

Cada um desses componentes será detalhado nas subseções a seguir.

3.1. API Web

Um pré-requisito para o uso da ferramenta é que antes de realizar qualquer nova implantação, o usuário precisa ter criado uma conta com um dos fornecedores de IaaS suportados. Uma vez que o usuário tenha criado essa conta, ele/ela tem que registrar suas credenciais de nuvem com a ferramenta, como mostrado na Figura 2.

Após fornecer as credenciais do provedor de nuvem, o usuário poderá proceder com a implantação. A API web é a camada que interage com os usuários e recebe todas as informações necessárias. Poucas informações são requeridas para que a implantação aconteça, tais como (ver Figura 3): provedor de nuvem a ser escolhido, propriedades de configuração do servidor de aplicação e do banco de dados.

3.2. Componente de Comunicação com a Nuvem

Esse componente implementa a interface TRexCloudService, provida pela ferramenta (ver Figura 4). O objetivo desta interface é definir um conjunto de operações de

Figura 2. Tela de configuração das credenciais de usuário no provedor de nuvem.

Figura 3. Tela de configuração de uma aplicação a ser implantada na nuvem.

implantação que são comuns a vários provedores de nuvem IaaS, desacoplando, assim, o cliente desta interface (camada de UI) dos detalhes de comunicação específicos de cada provedor. Para cada provedor de nuvem suportado pela ferramenta deve existir uma implementação diferente desta interface.

No caso da nuvem Amazon EC2, esta implementação se baseou nas APIs Java fornecidas pela própria Amazon. Já no caso da nuvem Rackspace, foram utilizados recursos de invocação de serviços web REST em Java para comunicação com os serviços deste provedor.

3.3. Imagens de Instâncias Pré-configuradas

Para implantar uma aplicação web Java EE é necessário que alguns softwares estejam instalados e configurados na máquina em que será realizada a implantação, tais como o ambiente de execução Java (JRE), o servidor web e o servidor de banco de dados, sendo este último opcional, dependendo do tipo da aplicação. Além disso, alguns comandos precisam ser executados de forma ordenada para a completa realização do processo de implantação, configuração do banco de dados e inicialização da aplicação.

Para automatizar os procedimentos descritos acima, a ferramenta TREXCLOUD disponibiliza imagens de máquinas virtuais contendo toda a pilha de aplicativos necessária e *scripts* de configuração (ver detalhes a seguir) para cada um dos provedores de nuvem IaaS suportados. Ressalta-se que o intuito é facilitar a implantação pelo usuário, que não precisará se preocupar nem com a criação de imagens nem com configurações complexas,

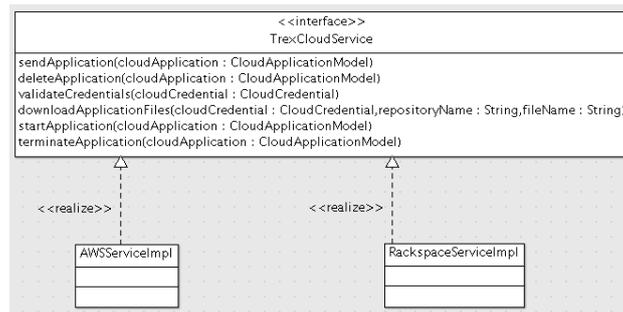


Figura 4. Camada de comunicação com os provedores de nuvem.

bastando informar os parâmetros solicitados pela ferramenta.

3.3.1. Pilha de aplicativos

A pilha de aplicativos web Java EE disponibilizada pela ferramenta contém os seguintes softwares: JRE, servidor web e banco de dados relacional. Diferentes imagens com combinações diversas dessa pilha de aplicativos e de *scripts* de configuração foram projetadas para dar suporte a uma grande variedade de aplicações e possíveis variações na sua arquitetura.

3.3.2. Scripts de configuração

Além da pilha de aplicativos necessária para implantação e funcionamento da aplicação, alguns comandos precisam ser executados para que a implantação seja efetuada com sucesso, como a cópia do arquivo da aplicação para a pasta correta do servidor web, a restauração do banco de dados, a inicialização do servidor web, inicialização do banco de dados, além de outras tarefas que precisam ser executadas de forma ordenada. No contexto da ferramenta proposta, esses comandos, escritos em uma linguagem Shell, são disparados logo após a inicialização da máquina virtual. Dependendo da aplicação e da implantação que está sendo efetuada, diferentes *scripts* serão executados, garantindo que a implantação ocorra corretamente.

3.4. Exemplo de uso

Afim de ilustrar a forma como o processo de implantação inteiro funciona, usaremos uma aplicação como exemplo. Suponha que o desenvolvedor de um aplicativo de Business Intelligence (BI), chamado *myDashboard*, quer implantá-lo na nuvem da Amazon. Se ele/ela decide usar a ferramenta TREXCLOUD (a partir de agora nós chamamos essa pessoa o usuário) o usuário precisa primeiro criar uma conta na Amazon (caso já não possua) e depois registra suas credenciais com a ferramenta (Figura 2).

O usuário, em seguida, tem de fornecer um conjunto de propriedades de configuração e alguns arquivos de entrada, como mostrado na Figura 3. Neste ponto, quando o usuário pressiona o botão "enviar", o que acontece "nos bastidores" é que todas essas propriedades são salvas em um arquivo que será enviado e armazenado no Amazon Simple Storage Service (S3). Além disso, os arquivos contendo a lógica da aplicação

(.WAR) e o *backup* do banco de dados (.TAR) também são transferidos do computador do usuário para o S3.

Uma vez que o usuário implanta a aplicação, todos os componentes são instanciados na nuvem de escolha (ex: Amazon). A implementação da Amazon para a interface da ferramenta de comunicação (AWSServiceImpl) vai invocar as APIs necessárias para inicializar as instâncias de VM (com base nas imagens pré-construídas) de todos os componentes de aplicação na ordem correta. Por exemplo, ele vai começar com a instância de banco de dados. Quando essa instância é inicializada, o *script* incorporado nela irá "baixar" o arquivo de propriedades e o *backup* do banco de dados (.TAR) do S3 e realizar a completa configuração da base de dados.

Depois que o banco de dados é configurado, a ferramenta inicializa a instância do servidor de aplicação (com base na imagem pré-construída com o Tomcat), a qual, durante a inicialização, copia os arquivos contendo a lógica da aplicação do S3 (.WAR) e faz a configuração correta do endereço IP do banco de dados baseado no IP da VM do banco de dados que foi inicializada previamente. O que é importante destacar é que todos estes passos de configuração ficam completamente transparentes para o usuário da ferramenta, o qual apenas tem de fornecer um conjunto de propriedades de configuração e arquivos. Esta é uma das principais contribuições da ferramenta TREXCLOUD, uma vez que torna a implantação da aplicação na nuvem em uma questão de poucos cliques para o usuário.

4. Avaliação

Afim de avaliar a eficácia da ferramenta TREXCLOUD, realizamos um estudo empírico em que nove usuários utilizaram três abordagens distintas para implantar dois aplicativos web diferentes na nuvem da Amazon. Nosso objetivo foi investigar como TREXCLOUD se compara, em termos de esforço do usuário, com outras abordagens comumente utilizadas para a implantação de aplicações web na nuvem da Amazon.

Especificamente, investigamos as seguintes questões:

(Q1) Como a nossa abordagem se compara, em termos do tempo gasto para concluir a implantação (i.e., esforço), com outras abordagens comumente utilizadas por clientes da Amazon?

(Q2) Que abordagem é mais adequada para qual perfil de usuário (considerando usuários com diferentes níveis de experiência em Java e tecnologias relacionadas com a nuvem)?

4.1. Planejamento

Para realizar a avaliação, foram selecionados nove usuários com os seguintes perfis (três usuários de cada perfil):

- leigo – usuários que nunca fizeram uma implantação de um aplicativo Java antes e que nunca usaram uma nuvem IaaS.
- intermediário – usuários que possuem alguma experiência anterior com a implantação de uma aplicação Java EE completa (incluindo servidor de aplicação e banco de dados), mas nenhuma experiência com nuvens IaaS.

- avançado – usuários que possuem as mesmas habilidades que os usuários do perfil intermediário mais alguma experiência anterior com implantação em nuvens IaaS.

Cada usuário foi solicitado a realizar seis implantações na nuvem da Amazon, sendo três implantações por aplicação. Em cada implantação, o usuário selecionou uma das três abordagens abaixo para implantar a aplicação:

Amazon Elastic Compute Cloud (EC2). Nesta abordagem, o usuário executa a implantação usando apenas as APIs da Amazon e seu *Dashboard*, que fornecem um conjunto de operações básicas para implantação e gerenciamento de instâncias na nuvem EC2. Portanto, com esta abordagem a maioria dos passos de implantação descritos na Seção 2 é executada manualmente pelo usuário da nuvem. Esta ainda é a abordagem mais comum quando se utiliza a nuvem da Amazon.

Amazon Elastic Beanstalk (EB). Nesta abordagem, o usuário executa a implantação usando o Amazon Elastic Beanstalk [Beanstalk 2012], uma ferramenta oferecida pela Amazon para implantar aplicações web em sua nuvem EC2. Uma limitação importante desta ferramenta é que ela não permite a implantação do banco de dados e se restringe apenas à lógica da aplicação.

TREXCLOUD (TREN). Nesta abordagem, o usuário executa a implantação usando a ferramenta TREXCLOUD, descrita neste trabalho.

As duas aplicações selecionadas para o estudo foram *Calendar* e *MyDashboard*. A primeira é uma aplicação de calendário simples, baseada em tecnologias Java para a web, fornecida pela Google. Uma característica interessante desse aplicativo é que ele não tem um banco de dados, contendo apenas a lógica de negócio empacotada em um arquivo (.WAR). A segunda aplicação é uma ferramenta de BI desenvolvida pela mesma equipe que desenvolveu TREXCLOUD. Este aplicativo contém tanto a lógica do aplicativo quanto o banco de dados. A lógica da aplicação é empacotada em um arquivo .WAR juntamente com um arquivo de contexto que possui propriedades de configuração. O banco de dados é distribuído através de um arquivo de *backup* (.TAR) e utiliza o Postgres.

Durante o estudo, os nove usuários foram instruídos a realizar as implantações em ordem aleatória, para evitar que a ordem pudesse influenciar nos resultados. Além disso, eles foram obrigados a completar as seis implantações. A implantação foi considerada completa quando o usuário: (i) teve a aplicação funcionando corretamente na nuvem da Amazon, o que foi verificado acessando a URL da aplicação na Amazon a partir de um navegador, ou (ii) quando o usuário explicitamente assumia que não tinha conseguido realizar a implantação. Para proporcionar um ambiente familiar para os experimentos, os usuários foram autorizados a realizar as implantações utilizando seus próprios computadores. No entanto, para facilitar a comparação dos esforços dos usuários configuramos todos os computadores utilizados no estudo para gravar um vídeo de sua tela durante cada sessão de implantação. Estes vídeos foram posteriormente analisados para se calcular com precisão o tempo que cada usuário necessitou para completar cada implantação.

4.2. Resultados do tempo de implantação

A Tabela 1 mostra os resultados que representam o tempo gasto por cada usuário para implantar o aplicativo *Calendar*. Pode-se perceber que, para todos os usuários (exceto para o usuário 2 por alguns segundos) TREXCLOUD superou as outras duas ferramentas

Usuário	Perfil	EC2	EB	TREX
Usuário 1	Leigo	-	0:22:26	0:04:22
Usuário 2	Leigo	-	0:03:51	0:04:02
Usuário 3	Leigo	-	0:24:34	0:04:29
Usuário 4	Intermediário	0:35:10	0:11:24	0:01:57
Usuário 5	Intermediário	0:51:05	0:05:33	0:02:48
Usuário 6	Intermediário	0:56:12	0:35:39	0:01:51
Usuário 7	Avançado	0:29:08	0:15:27	0:02:15
Usuário 8	Avançado	0:27:55	0:28:27	0:01:47
Usuário 9	Avançado	0:30:43	0:04:34	0:02:19

Tabela 1. Tempo total de implantação do aplicativo *Calendar*

por uma margem grande. Além disso, nenhum dos usuários leigos foi capaz de implantar o aplicativo com o EC2. Em contraste, todos os usuários intermediários e avançados foram capazes de concluir com sucesso a implantação com o EC2; no entanto, essas implantações necessitaram de uma quantidade considerável de tempo – cerca de 28 minutos no caso mais rápido (para o usuário 8). O maior esforço exigido pelo EC2 pode ser explicado pelo fato desta abordagem ser essencialmente manual.

Outro resultado interessante é o tempo de implantação obtido com o Amazon Elastic Beanstalk (EB). Como o aplicativo *Calendar* não contém um banco de dados, tecnicamente implantá-lo com esta abordagem deveria envolver um nível semelhante de dificuldade ao necessário para fazê-lo com a ferramenta TREXCLOUD. Isso foi observado para os usuários 2, 5 e 9, que foram capazes de completar a tarefa com EB em menos de 6 minutos. No entanto, pode-se ver que a maioria dos usuários levou muito mais tempo para implantar o aplicativo com EB do que com TREXCLOUD.

Na verdade, a implantação do aplicativo com TREXCLOUD sempre foi muito rápida, levando menos de dois minutos (para usuários de 4, 6 e 8), até um máximo de quatro minutos e 29 segundos (para o usuário 3). A maior parte desse tempo é gasto com a transferência dos arquivos de configuração (mencionado na Seção 3) para a nuvem e com a inicialização das instâncias de máquina virtual na Amazon, e não com interação do usuário com a ferramenta.

A Tabela 2 mostra os resultados do tempo gasto por cada usuário para implantar o aplicativo *MyDashboard*. Mais uma vez, somente os usuários intermediários e avançados foram capazes de completar a tarefa com o EC2, embora eles tenham levado muito tempo para fazê-lo (cerca de duas horas, em média).

No caso do Amazon Elastic Beanstalk, desta vez apenas um usuário (usuário 5) foi capaz de executar a implantação, levando cerca de uma hora para completar a tarefa. A razão para a elevada taxa de insucesso observada com esta abordagem foi a dificuldade para implantar o banco de dados da aplicação. Ao analisar os vídeos gravados durante as sessões de implantação realizadas com o EB, percebemos que os usuários de alguma forma ficaram confusos sobre a maneira correta de configurar a aplicação, uma vez que a ferramenta não apóia explicitamente a implantação do banco de dados.

Os resultados para TREXCLOUD são excelentes. Todos os usuários de todos os perfis foram capazes de completar a tarefa em aproximadamente sete minutos, em média. Tais resultados favoráveis são justificados porque TREXCLOUD torna o processo de implantação muito mais fácil, bastando ao usuário fornecer os arquivos necessários

Usuário	Perfil	EC2	EB	TREX
Usuário 1	Leigo	–	–	0:07:48
Usuário 2	Leigo	–	–	0:06:27
Usuário 3	Leigo	–	–	0:07:06
Usuário 4	Intermediário	1:57:17	–	0:06:49
Usuário 5	Intermediário	2:09:01	1:08:06	0:05:41
Usuário 6	Intermediário	2:21:55	–	0:06:15
Usuário 7	Avançado	1:45:33	–	0:07:31
Usuário 8	Avançado	1:55:06	–	0:06:32
Usuário 9	Avançado	2:02:58	–	0:08:17

Tabela 2. Tempo total de implantação do aplicativo *MydashBoard*

para implantar a lógica da aplicação e o banco de dados. Todos os aspectos relacionados com a configuração dos componentes da aplicação e suas dependências são tratados automaticamente pela ferramenta, como explicado na Seção 3.

Com base nas tabelas 1 e 2, foi calculada a redução do esforço conseguido por TREXCLOUD quando comparada com EC2 e EB. Para o aplicativo *Calendar* (considerando apenas implantações bem-sucedidas) TREXCLOUD conseguiu uma redução média de 94% e 83% no tempo de implantação do aplicativo, quando comparado, respectivamente, com EC2 e EB. Em relação ao aplicativo *myDashboard*, os dados mostram uma redução média de 94% e 91% no tempo de implantação em relação ao EC2 e EB, respectivamente (considerando apenas as implantações bem-sucedidas).

4.3. Resultados do Questionário de Avaliação

Além da análise quantitativa do esforço de implantação, também foi realizada uma análise qualitativa para obter a impressão dos usuários sobre cada uma das três ferramentas. Pedimos aos usuários para preencher um questionário simples, que continha os seguintes itens:

(i) Atribuir uma nota para cada implantação, em uma escala de 0 a 5, em que 0 representa que o usuário não conseguiu realizar a implantação e {1, 2, 3, 4, 5} representam, respectivamente, que a implantação foi {muito difícil, difícil, moderada, fácil, muito fácil} de realizar.

(ii) Sugerir algumas melhorias na funcionalidade de cada uma das ferramentas utilizadas.

(iii) Comentar sobre as vantagens da utilização de cada ferramenta no contexto de cada implantação.

A Figura 5 mostra a avaliação dos questionários para a aplicação *MyDashboard*. Com relação aos resultados do EC2, como os usuários leigos não conseguiram completar a tarefa, a nota atribuída por todos foi zero. Os resultados mais interessantes são os atribuídos à ferramenta Amazon Elastic Beanstalk (EB). Somente o usuário 5 conseguiu completar a tarefa com o EB, embora com bastante dificuldade, levando-o a atribuir a essa ferramenta a nota 1 (muito difícil). Conforme explicado anteriormente, a razão disso foi a dificuldade de configurar o banco de dados da aplicação por falta de suporte da ferramenta EB. Com relação à TREXCLOUD, todos os usuários consideraram a implantação com essa ferramenta fácil ou muito fácil, devido ao seu suporte facilitado para a implantação de todos os componentes da aplicação.

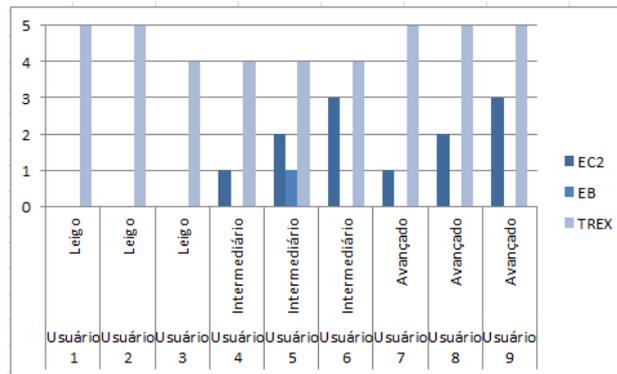


Figura 5. Notas atribuídas pelos usuários para a implantação do *MyDashboard*.

Algumas das respostas dadas pelos usuários para as questões abertas incluídas no questionário podem fornecer mais evidências dos benefícios da ferramenta TRES-CLOUD. Por exemplo, a seguinte resposta foi dada por um dos usuários leigos: “TRES-CLOUD é muito mais rápida e prática. Eu só preciso fornecer os arquivos de entrada e dar alguns cliques para fazer tudo. A ferramenta faz todo o trabalho por mim.”

4.4. Limitações

O estudo realizado neste trabalho compartilha de algumas limitações comumente encontradas em experimentos empíricos envolvendo software [Easterbrook et al. 2008] como, por exemplo, depender de pessoas com variados níveis de experiência para realizar as implantações. Para os objetivos do nosso estudo, era importante que os usuários possuíssem perfis diferenciados, uma vez que queríamos investigar o quanto isso poderia influenciar nos resultados para as diferentes ferramentas avaliadas. Conforme mencionado anteriormente, tentamos garantir que os usuários seguissem ordens de implantações diferentes para evitar beneficiar uma ferramenta em detrimento de outras, visto que os usuários aprendem mais sobre as ferramentas após realizar cada implantação.

5. Trabalhos relacionados

A maioria das ferramentas disponíveis para implantação de aplicativos em nuvens IaaS não se limita apenas a esta funcionalidade. Elas também oferecem outros serviços, tais como a possibilidade de controlar e dimensionar automaticamente a aplicação na nuvem, que está fora do âmbito do presente trabalho. Como a maioria dessas abordagens visa suportar implantar qualquer tipo de aplicação, geralmente utilizam mecanismos de implantação mais genéricos. Por exemplo, a fim de implantar na plataforma Claudia [Rodero-Merino et al. 2010], o usuário tem que especificar a implantação em um arquivo descritor baseado em XML que contém informações sobre as imagens usadas, assim como as definições das propriedades de configuração da aplicação e das dependências dos componentes.

A abordagem TOSCA [Binz et al. 2012] funciona de uma forma semelhante à Claudia. As aplicações que serão implantadas na nuvem são especificadas em um documento XML onde o usuário descreve os componentes de software a serem utilizados, as suas relações e propriedades de configuração. Além disso, o usuário também especifica os fluxos de trabalho (chamados planos de execução) que descrevem o fluxo de

gerenciamento da aplicação. Já com o serviço Rightscale [RightScale 2012], as propriedades, configuração e dependências de componentes têm que ser especificados em uma linguagem de *script* chamada “Right Scripts” que têm o propósito de ajudar a configurar automaticamente a aplicação. Abordagens semelhantes, também baseadas em *script*, são oferecidas por CloudFoundry [CloudFoundry 2012] e Heroku [Heroku 2012]. A principal lacuna dessas abordagens é que elas exigem um conhecimento mais profundo de administração de sistemas e linguagens de *script*, tornando sua utilização limitada a usuários mais avançados.

A abordagem que mais se assemelha a este trabalho seria a Amazon Elastic Beanstalk [Beanstalk 2012]. Sua principal limitação é que, atualmente, ela não suporta a transferência e configuração automática do banco de dados da aplicação, o que torna mais difícil de implantar e configurar completamente aplicações web “empresariais”, como foi demonstrado em nosso estudo empírico.

6. Conclusão e Trabalhos Futuros

Neste trabalho, apresentamos TREXCLOUD, uma ferramenta criada para facilitar a implantação de aplicações Java EE em nuvens IaaS. TREXCLOUD permite que os usuários rapidamente implantem uma aplicação web na nuvem, exigindo, como entrada, um pequeno conjunto de arquivos que são automaticamente enviados para uma nuvem IaaS e usados pela ferramenta para instalar e configurar todas as máquinas virtuais que representam a pilha de software completa da aplicação. Nosso trabalho futuro se concentrará em apoiar outras plataformas de desenvolvimento (ex: .NET, PHP, Ruby on Rails), bem como outros sistemas de banco de dados (ex: MySQL, NoSQL).

Agradecimentos

Este trabalho é parcialmente financiado com Recursos obtidos da FINEP.

Referências

- Beanstalk (2012). Aws elastic beanstalk. <http://aws.amazon.com/elasticbeanstalk/>.
- Binz, T. et al. (2012). Portable cloud services using toska. *IEEE Internet Computing*, 16(3):80–85.
- CloudFoundry (2012). Cloud foundry. <http://www.cloudfoundry.com/>.
- Easterbrook, S. et al. (2008). Selecting Empirical Methods for Software Engineering . *Guide to Advanced Empirical Software Engineering*.
- Hajjat, M. (2010). Cloudward bound: planning for beneficial migration of enterprise applications to the cloud. In *ACM SIGCOMM 2010*. ACM.
- Heroku (2012). Heroku cloud application platform. <http://www.heroku.com/>.
- RightScale (2012). Right scale cloud management plataform. <http://www.rightscale.com/>.
- Rodero-Merino, L. et al. (2010). From infrastructure delivery to service management in clouds. *Future Generation Comp. Syst.*, 26(8):1226–1240.