

Padronização de implantação de sistema eletrônico de informações e gestão de documentos (SEI) em PaaS Kubernetes públicas e privadas

Carlos Lima¹

 orcid.org/0000-0002-8320-5452

Igor José¹

 orcid.org/0000-0001-8409-8647

Marcelo Lima²

 orcid.org/0000-0002-9756-2821

Gustavo Callou³

 orcid.org/0000-0002-7997-374X

¹Residência em Cloud Computing, Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.

²Residência em Cloud Computing, Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil. E-mail: marcelo.lima@tipe.jus.br

³Departamento de Computação, Universidade Federal Rural de Pernambuco, Recife, Brasil. E-mail: gustavo.callou@ufrpe.br

DOI: 10.25286/repa.v8i2.2184

Esta obra apresenta Licença Creative Commons Atribuição-Não Comercial 4.0 Internacional.

Como citar este artigo pela NBR 6023/2018: Carlos Lima; Igor José; Marcelo Lima; Gustavo Callou. Padronização de implantação de sistema eletrônico de informações e gestão de documentos (SEI) em PaaS Kubernetes públicas e privadas. Revista de Engenharia e Pesquisa Aplicada, v.8, n. 2, p. 11-20, 2023

RESUMO

Esse artigo visa apresentar o projeto SEI-Kubernetes que modifica o paradigma arquitetural monolítico em sistema do governo, que apesar de ser um sistema estável e funcional, possui desafios diretamente ligados a arquitetura atual. A adoção de microsserviços para o Projeto SEI-Kubernetes constitui um excelente avanço na melhoria da configuração e instalação de sistemas de mesmo tipo, além de otimizar os recursos computacionais existentes. O estudo apresentado demonstra que os tempos para configuração e instalação do ambiente podem ser reduzidos consideravelmente. Vale destacar que é possível empregar esse conceito amplamente em diversos centros de processamento de dados do Governo Brasileiro, que faz uso da mesma tecnologia de gestão de processos e arquivamento eletrônico de documentos. Esta nova abordagem na arquitetura do sistema SEI se beneficia de computação nas nuvens privadas, o que possibilita uma maior utilização dos serviços existentes.

PALAVRAS-CHAVE: microsserviços; SEI; computação nas nuvens.

ABSTACT

This paper aims introducing the project SEI-Kubernetes that changes the monolithic architectural paradigm in a government system, however the system is stable and functional, it has challenges directly related to actual architecture. The adoption of microservices into Project Sei-Kubernetes constitutes an excellent advance in how the system using the same design can improve their setup and configuration, also optimizing computational resources in place. The present study shows a sensible reduction in the time spent to setup and install the system. It is important to highlight the possibility of utilizing the same concept here adopted on multiple datacenters of Brazilian Government, since they are using the same technology of process management and document management system. This new architectural approach of SEI has benefited from private cloud computing, enabling a scale in the existing services.

KEY-WORDS: microservices; SEI; cloud computing.

1 INTRODUÇÃO

O Sistema Eletrônico de Informações (SEI) [1], desenvolvido pelo Tribunal Regional Federal da 4ª Região (TRF4), possui as funcionalidades de gestão de processos e o arquivamento eletrônico de documentos, contendo uma boa interface e buscando adotar práticas inovadoras para o dia a dia. Dentre as principais características podemos citar a substituição do papel para suporte físico em documentos institucionais e o compartilhamento de conhecimento com a comunicação de novos eventos em tempo real.

O SEI possui uma arquitetura computacional de caráter monolítico, que precisa de uma quantidade grande de recursos computacionais para garantir a disponibilidade e a confiabilidade dos dados. Apesar da estrutura atual ser funcional, a escalabilidade do sistema representa um desafio, tendo em vista que a quantidade de dados armazenados ao longo do tempo tende a aumentar, bem como quantidade de usuários acessando o sistema simultaneamente.

Neste contexto, a adoção de uma nova arquitetura, com a adoção de tecnologias amplamente usadas na atualidade, pode permitir a longevidade do sistema e enfrentando os desafios encontrados no modelo atual. Desta forma, esse projeto visa a proposição de uma arquitetura baseada em microsserviços para o sistema SEI. Esse ambiente irá utilizar o SEI configurado no Kubernetes via Helm.

Além disso, esse artigo também analisa outros fatores fundamentais relacionados ao tempo demandado para a configuração do ambiente, a otimização dos recursos computacionais utilizados e a escalabilidade do ambiente SEI. A utilização de novas tecnologias com arquitetura microsserviços visa permitir um melhor uso dos recursos computacionais existentes.

A escolha do SEI para este estudo se deve a grande adoção do sistema em diversos entes públicos no Brasil, por fazer parte da plataforma de software público governamental, e por estar no cotidiano dos usuários. Contudo, outras tecnologias, governamentais ou não, podem se aproveitar da adoção de Kubernetes com Helm e obter resultados satisfatórios como os que são demonstrados nesse trabalho.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os conceitos necessários para um melhor entendimento do trabalho. A Seção 3 mostra os trabalhos relacionados existentes na literatura. A Seção 4 apresenta a metodologia utilizada. A Seção 5 mostra um comparativo entre a solução proposta e a atual. Por fim, a Seção 6 conclui o trabalho

e fornece os direcionamentos para a continuidade dessa pesquisa.

2 CONTÊINERES, COMPUTAÇÃO NAS NUVENS E O SEI

Contêineres e a computação nas nuvens são termos comumente utilizados na atualidade, e a adoção desses paradigmas representa uma nova forma de pensar [2], explorando tecnologias de ritmo rápido para a redução de riscos e melhorando a qualidade do serviço oferecido. A seguir, serão descritas as tecnologias utilizadas na abordagem microsserviços contêineres, computação nas nuvens e o SEI.

2.1. Contêineres

Antes de definir contêineres, é preciso compreender o que são microsserviços. KOCHER [3] define que microsserviços são independentes, possuem funcionalidades padrões que são desenhadas na forma de executáveis, ou ainda na forma de processos, e que se comunicam uns com os outros através de padrões de baixo custo computacional. Esse autor estabelece ainda que microsserviços são únicos e diferentes de uma aplicação padrão, pois cada microsserviço é desenvolvido, testado, executado e escalado por demanda e de forma independente.

Os contêineres representam uma parte importante na arquitetura proposta para o SEI, sendo responsáveis por uma mudança na forma como são desenhadas soluções de software. Além disso, a distribuição e a execução dos softwares também diferem, pois esse paradigma permite a abstração do ambiente em que a aplicação será executada. MOUAT [4] define que containers são o encapsulamento de uma aplicação e suas dependências, enaltecendo as características como a natureza leve, que permite a execução de vários contêineres ao mesmo tempo sob o mesmo sistema operacional. Fora isso, outro fator a se destacar é a portabilidade, que evita problemas relacionados à mudança no ambiente da aplicação.

2.1.1. Kubernetes

A evolução na utilização dos microsserviços pode ser observada com o advento do Kubernetes, que faz a denominada orquestração de containers. O Kubernetes [5] foi criado a partir de um projeto feito pelo Google, em 2014, onde o objetivo era o de

implementar um orquestrador de contêineres para ser utilizado pelo próprio Google.

Sobre o papel do Kubernetes é importante destacar que a administração de um sistema inclui diversas atividades, dentre elas a atualização de servidores, instalações para correção de segurança, configuração e administração de redes computacionais e execução de backups. Com o Kubernetes [5] é possível automatizar essas tarefas para que a equipe de suporte se concentre em fazer o trabalho que lhe for essencial. Esse ambiente permite ainda o balanceamento de carga e a escalabilidade de forma automática, uma vez que tais recursos estão incluídos no núcleo de sua implementação. Dessa forma, pode-se observar que o Kubernetes facilita o processo de utilização dos contêineres.

2.1.2. Helm

Apesar das facilidades trazidas pelo Kubernetes existem algumas repetições nos arquivos de configuração para lançamento e uso dos contêineres. O Helm [5] permite definir os parâmetros uma única vez e referenciá-los sempre que preciso. O Helm pode ser definido então como um gerenciador de pacotes para Kubernetes. Nesse caso, o Helm é utilizado para instalar e configurar as aplicações definidas pelos pacotes do Kubernetes.

2.2. Computação nas Nuvens

A Computação nas nuvens possui um papel importante na definição arquitetural do SEI e, também, para o uso do Kubernetes no projeto proposto. CHANDRASEKARAN [6] descreve que, em termos simples, a computação nas nuvens significa armazenar e acessar dados e programas através da internet, de locais remotos ou computadores, ao invés de acessar a partir do disco de armazenamento local.

Uma outra definição, mais formal, é fornecida pelo NIST (*National Institute of Standards and Technology*) [6], onde computação nas nuvens é definida como "um modelo que habilita de forma ubíqua e conveniente o acesso sob demanda a um conjunto de recursos computacionais compartilhados que podem ser rapidamente provisionados e lançados com o mínimo esforço no gerenciamento ou ajuda do provedor de serviços". A existência da computação nas nuvens permite a utilização do SEI, que será definido a seguir.

2.3. SEI

O Sistema Eletrônico de Informações (SEI), desenvolvido pelo Tribunal Regional Federal da 4ª Região (TRF4), é um sistema de gestão de processos e documentos arquivísticos eletrônicos. O SEI [1] é um dos produtos do projeto Processo Eletrônico Nacional (PEN), iniciativa conjunta de órgãos e entidades de diversas esferas da Administração Pública, visando a construção de uma infraestrutura pública de processos e documentos administrativos eletrônicos. O objetivo do SEI é possibilitar a melhoria no desempenho dos processos do Governo Brasileiro e fornecer, assim, ganhos em agilidade, produtividade, transparência, satisfação do usuário e a redução de custos.

A arquitetura do SEI é descrita na Seção 5 (Arquiteturas dos Sistemas). Essa seção irá descrever em detalhes a abordagem que foi adotada e analisar os benefícios obtidos, como o de ampliar a escalabilidade, a disponibilidade e facilitar a administração na implantação do sistema.

3 TRABALHOS RELACIONADOS

O governo e entidades brasileiras oferecem diversos serviços diretamente aos cidadãos para os mais variados fins. Esses serviços oferecidos pelos governos podem se beneficiar da adoção de computação nas nuvens e containers. De acordo com BRETSCHNEIDER e MERGEL [7], em um estudo sobre a influência das novas tecnologias no *e-Government* (Governo Eletrônico), as iniciativas referentes aos serviços do Governo aos cidadãos nos Estados Unidos se encontram nos estágios iniciais e, assim, é difícil prever quando essas tecnologias irão se encontrar capazes de reduzir a burocracia e as estruturas hierárquicas complexas dos serviços públicos. Para isso, os autores fizeram uma análise histórica das tendências ao longo do tempo, relacionando sete propostas que podem ajudar na evolução de sistemas governamentais. No entanto, não houve uma abordagem mencionando sistemas ou arquiteturas, mas sim contextos que podem levar a uma evolução do *e-Government*.

A tendência e a busca por novas tecnologias podem ajudar na melhoria dos serviços governamentais ao cidadão. POPOV e KHRIPUNOV [8] fazem uma analogia à adoção de novas tecnologias como *Cloud Computing*, Big Data e microsserviços para a melhoria no atendimento dos serviços ao cidadão, com a designação do denominado "*Government-as-a-Platform*". Essa transformação digital passa

principalmente pelo uso de *Cloud Computing*. A abordagem da arquitetura baseada em microsserviços visa assegurar a entrega e o suporte de sistemas com a qualidade e a velocidade que os negócios na atualidade requerem. Este tipo de arquitetura possibilita atualizações com rapidez e uma redução de tempo na preparação e lançamento de novas versões do sistema. Os autores sugerem que a criação de uma camada de arquitetura denominada "*Dataware layer*" - camada de armazenagem de dados - é necessária para a evolução dos sistemas, visto que irá permitir uma abstração maior do sistema e uma melhor gestão dos dados.

Os sistemas *e-Government* podem evoluir como, por exemplo, o Onnara BPS [9] do governo da Coreia do Sul. Nesse caso, a análise de requisitos e a listagem de funcionalidades do sistema possibilitaram a definição de uma nova arquitetura baseada em microsserviços. Isso só foi possível, pois na fase de desenho de solução as entidades do sistema monolítico puderam ser agrupadas em categorias, e em seguida foram classificadas como possíveis candidatas a serem implementadas como microsserviços.

A aplicação de arquiteturas microsserviços não é uma exclusividade a ser aplicada aos serviços públicos. CARDOSO [10] faz referência a uma plataforma de faturamento eletrônica de estrutura monolítica com diversos problemas de desempenho, escalabilidade e manutenção. Estudos de casos foram realizados, dentre eles uma plataforma de serviços de vídeo (Netflix), explicando o funcionamento da plataforma e como os microsserviços estão na arquitetura apresentada. Ao final do estudo, CARDOSO [10] realiza a medição através de indicadores de desempenho, escalabilidade, manutenção e o processo na comparação do sistema de faturamento eletrônico na arquitetura monolítica com a de microsserviços.

SOUZA [11] propõe a modernização do *Data Center* do Instituto Federal de Santa Catarina através de uma migração para o modelo baseado em microsserviços. A proposta, baseada na preocupação com os serviços de tecnologia da informação centralizados, busca evidenciar as possibilidades e de como a computação nas nuvens pode aumentar a escalabilidade, e ser mais tolerante a falhas através da orquestração de containers com a adoção do Kubernetes.

Diferente dos trabalhos anteriores, esse artigo propõe um Projeto SEI Kubernetes através da adoção de uma nova arquitetura, ou seja, não consiste apenas em bases teóricas e sugestão de uma nova abordagem, mas sim mudança de paradigma arquitetural do sistema de monolítico para microsserviços. Além disso,

esse artigo também demonstra a preparação e construção do sistema utilizando Kubernetes com Helm que está apta ao uso para instalação de um ambiente SEI de acordo com as especificações existentes para o sistema [1]. Há, assim, uma abstração maior no processo de instalação e configuração inicial do sistema, o que permite ganhos na adoção, redução possíveis erros humanos, e permitindo a redução de recursos computacionais. É importante destacar que a ênfase desse artigo é na mudança de paradigma de estrutura monolítica para microsserviços, de forma a melhorar a escalabilidade e facilidade na adoção. Além disso, apresenta-se uma solução prática e concreta de Kubernetes com Helm, bem como a modernização de serviços Governamentais, e com a abrangência de utilização do SEI por várias entidades do Governo brasileiro. Isso significa na possibilidade de adoção com a modernização da instalação e configuração, de forma prática, e na implementação concreta no contexto específico do SEI, a redução da burocracia governamental e a adaptação das tecnologias de microsserviços.

4 METODOLOGIA

Essa seção apresentada a metodologia proposta para a realização de um estudo comparativo entre a arquitetura do SEI na configuração do sistema na atualidade, e uma estrutura utilizando tecnologias de contêineres. A comparação da arquitetura recomendada [12] para o SEI com a nova proposta foi realizada confrontando o desenho de ambas as soluções, da quantidade de recursos computacionais exigidos para cada abordagem, e registrando o tempo demandado para configuração e instalação.

Os recursos computacionais em ambos os ambientes são descritos de acordo com a documentação de instalação e configuração do ambiente SEI [12], e seguem os parâmetros segundo uma abordagem de ambiente de microsserviços. É fundamental expressar que a escalabilidade representa o fator principal para adoção deste tipo de ambiente baseado em microsserviços. Todavia, não compete neste estudo a comparação durante o uso dos sistemas, ou seja, não existe a comparação no uso dos sistemas instalados e como eles se comportam em ambientes simulados de produção. Entretanto, foi possível coletar algumas informações relacionadas ao tempo de configuração e instalação de alguns componentes, demonstrando a abordagem positiva da arquitetura proposta utilizando Kubernetes

O ambiente SEI em sua estrutura atual é complexo, demanda grandes recursos computacionais, e todos os parâmetros e informações aqui descritos foram extraídos dos documentos oficiais para a implantação do sistema SEI. Para as questões referentes ao SEI em formas de containers, existe a reengenharia de fato, disponível para instalação em qualquer ambiente, já estruturado em ambiente de repositório acessível a aqueles que desejam conhecer e implementar a nova abordagem. Desta forma, as descrições de arquiteturas a seguir, bem como as temáticas abordadas no que diz respeito a estrutura de implantação atual são documentais, e baseada nas observações que foram possíveis do ambiente atual. No que diz respeito a estrutura microsserviços, existe uma implementação real e disponível que se faz beneficiária das tecnologias já descritas.

A Figura 1 demonstra os componentes da arquitetura do SEI na atualidade. Nessa figura, pode-se observar quais são os itens que precisam ser instalados e configurados para se ter o ambiente em plena funcionalidade. Destaca-se que cada componente deve ser instalado e configurado manualmente, um por vez, o que demanda tempo e esforço humano para se ter o pleno funcionamento da aplicação do sistema SEI.

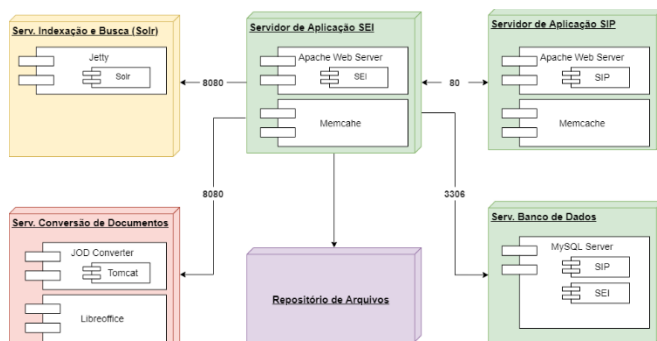


Figura 1: Componentes do Sistema SEI.
Fonte: Site do Software Público [12].

A Figura 2 apresenta a execução em separado de três aplicações (App A, Y e Z) com máquinas virtuais, onde é necessário um supervisor (*Hypervisor*) para comunicação e controle entre o sistema operacional e o hardware existente. Além disso, cada máquina virtual precisa do seu sistema operacional, o que resulta em uma grande demanda para alocação de recursos. Quando comparado com a Figura 3, pode-se perceber as mesmas três aplicações na estrutura de container. Nesse cenário, não há a necessidade de replicação dos sistemas operacionais e os processos são executados de forma nativa, o que dispensa a adoção do supervisor.

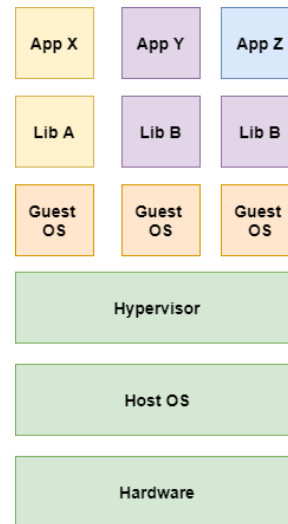


Figura 2: Aplicações na estrutura de máquinas virtuais.
Fonte: *Using Docker: Developing and deploying software with containers* [13].

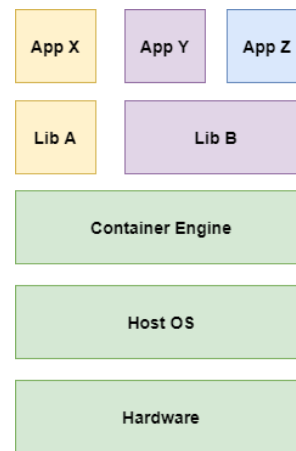


Figura 3: Aplicações na estrutura de máquinas virtuais.
Fonte: *Using Docker: Developing and deploying software with containers* [13].

A comparação das Figuras 2 e 3 demonstram de forma clara os ganhos para a mudança de paradigma arquitetural com máquinas virtuais para uma estrutura de contêineres, nesse caso com Kubernetes.

5 ARQUITETURAS DOS SISTEMAS

Esta seção descreve as arquiteturas dos sistemas apresentados nas Figuras 4 e 5. A primeira figura considerada o padrão atual, chamado aqui por arquitetura monolítica [12]. A seguir, é descrita a estrutura em microsserviços do SEI.

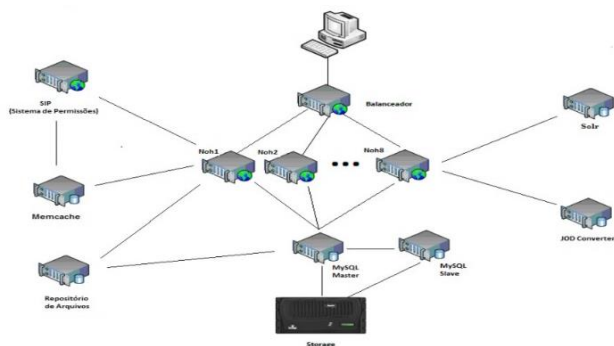


Figura 4: SEI – Infraestrutura de Hardware Sugerida.
Fonte: Site do Software Público [12].

A arquitetura do SEI é composta pelos seguintes componentes [12]: o balanceador de aplicação, um total de oito nós para a aplicação, um servidor para cache (Memcache), mecanismo de busca (Apache Solr), geração de arquivos de extensão .pdf (JOD Converter), repositório de arquivos, sistema de permissões (SIP), e o banco de dados (MySQL *master* e *slave*). Na estrutura monolítica, o ambiente faz uso do sistema operacional Red Hat Enterprise Linux 7.1, envolvendo máquinas físicas e virtuais conforme mostrado na Tabela 1.

Tabela 1: Infraestrutura de Hardware.

Componente	Classificação
Balanceador de Aplicação	Software.
Nó de aplicação	Máquina Virtual
Serviço de Cachê	Máquina Virtual
Mecanismo de busca	Máquina Virtual
Geração de Arquivos .pdf	Máquina Virtual
Repositório de Arquivos	Máquina Física
Sistema de Permissões	Máquina Virtual
Banco de Dados	Máquina Física

Além dos componentes descritos, pode-se verificar na Tabela 2 os recursos computacionais [12] necessários no sistema para a configuração do SEI na atualidade.

Tabela 2: Recursos computacionais.

Recurso Computacional	Total
Memória RAM	262 GB
Capacidade de Disco	337 GB
CPUs	15 Unidades

A estrutura atual do SEI requer uma grande disponibilidade de recursos computacionais. Para se ter uma ideia, são necessários oito nós de aplicação, que consistem em máquinas virtuais com o sistema operacional Red Hat Enterprise Linux para a hospedagem das aplicações. Em termos de ganhos, apenas esta parte da arquitetura já é extremamente vantajosa na arquitetura microsserviços, pois os containers não necessitam de uma máquina virtual para cada aplicação, o que resulta em grande redução no uso de recursos computacionais. Além dos recursos computacionais, existem também a economia de recursos financeiros para instituições, pois não há a necessidade de tantos nós de aplicação utilizando o sistema operacional. Logo, as quantidades de licenças podem ser reduzidas, o que representa uma boa utilização dos recursos públicos.

Um outro aspecto a ser considerado na estrutura com máquinas virtuais do SEI é a infraestrutura mínima para atender o repositório de arquivos. De acordo com o Manual do SEI [12] são necessários 48 GB de memória e espaço inicial de 1.2 Terabytes apenas para esta componente da arquitetura, ratificando a quantidade de recursos computacionais necessários para adoção de um sistema SEI, na atualidade.

Nas orientações para instalação e configuração do ambiente SEI já existia a preocupação com a escalabilidade do sistema [12], apontando com pontos a serem observados o crescimento excessivo do banco de dados e do repositório de arquivos. Quanto ao crescimento em si, o mesmo pode ocorrer em qualquer tipo de arquitetura, mas a forma como o sistema na arquitetura em microsserviços é lançado permite a economia de recursos computacionais no ponto inicial da operacionalização do sistema SEI.

A estrutura em microsserviços proposta [14] é demonstrada conforme pode ser visto na Figura 5. Nessa figura, pode-se observar que os componentes são os mesmos conforme a recomendação do projeto SEI [12]. É possível observar que não há perda de componentes, conforme mencionado anteriormente, mas como o nível de abstração aumentou, a

intervenção humana é menor, mitigando erros e melhorando a instalação e a configuração do SEI.

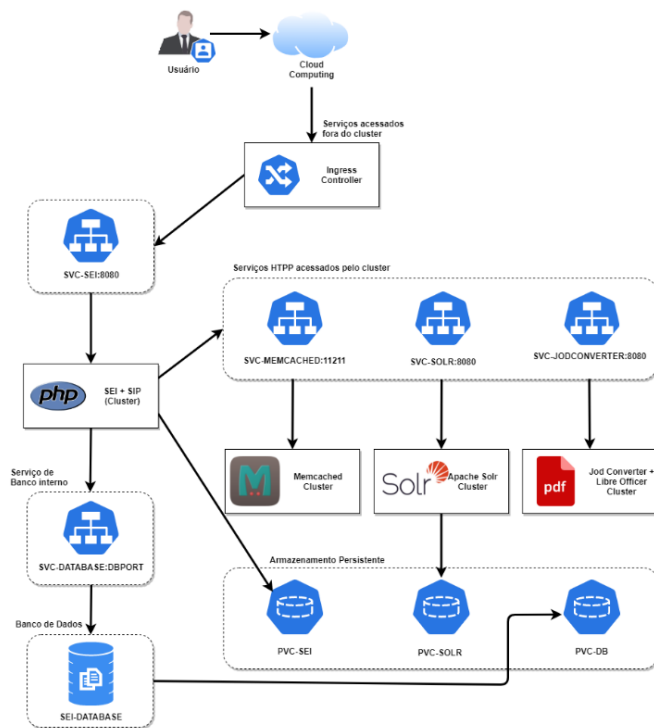


Figura 5: SEI – Infraestrutura de Hardware Sugerida. Fonte: GitHub – Projeto SEI Kubernetes [14].

Os recursos computacionais utilizados na proposta arquitetural são diferentes, pois trata-se de estruturas em microsserviços em forma de contêineres. E, assim, essa estrutura não precisa de uma reserva de recurso inicial mínima, pois o ambiente está apto para as configurações mínimas descritas na Tabela 3.

Tabela 3: Recursos computacionais microsserviços.

Recurso Computacional	Total
Memória RAM	8 GB
Capacidade de Disco	5 GB
CPUs	2 Unidades

A primeira etapa na escolha da migração do ambiente SEI para a utilização da estrutura em microsserviços se deve pela redução do tempo de instalação e configuração do ambiente SEI. Como não há a necessidade de máquinas virtuais, e de tantas configurações que exijam a intervenção de forma manual, o ambiente em microsserviços proposto possui uma grande parte automatizada, o que reduz

consideravelmente a complexidade da configuração e o tempo para o serviço estar disponível.

A arquitetura existente foi descrita na Figura 4, demonstrando os componentes existentes e que são os mesmos observados na arquitetura do projeto SEI em microsserviços (Figura 5). A arquitetura em microsserviços é composta pelos seguintes *containers*: banco de dados (MySQL), serviço de conversão para arquivos em formato .pdf (JodConverter), serviço web (Apache, PHP e Memcache), e serviço de indexação (Solr). Dentre os pré-requisitos, a tecnologia Helm precisa estar previamente instalada. Após isso, o processo é simples, basta realizar a instalação do sistema utilizando um repositório, conforme é demonstrado nos passos a seguir:

1. Realize o clone do repositório localmente:

```
git clone https://github.com/seikubernetes/projeto-sei.git
```

2. Crie um namespace em seu ambiente kubernetes:

```
kubectl create namespace projeto-sei
```

3. Defina a variável de namespace name no arquivo de nome values.yaml conforme nome do namespace criado anteriormente.

4. Configure as variáveis do values.yaml correspondentes ao seu ambiente kubernetes.

5. Instale o Helm Chart do SEI para criar o ambiente completo baseado nas definições estabelecidas:

```
helm install projeto-sei ./projeto-sei/sei
```

Após seguir esses passos, o SEI na arquitetura microsserviços estará instalado e pronto para uso. Para visualizar os passos para desinstalação, visualizar os pacotes de dependência e verificar a licença, existe um repositório existente no GitHub [14] com todas essas informações.

6 ANÁLISE DE RESULTADOS

Com o intuito de comparar os ganhos da adoção do sistema SEI utilizando microsserviços, foi realizada de forma repetitiva e seguindo o mesmo procedimento a instalação de alguns componentes do sistema SEI na atualidade. Desta forma, foi possível entender o quanto a automação do processo de instalação e configuração do sistema com a adoção do Kubernetes é benéfica.

A Tabela 4 mostra os registros dos tempos demandados na instalação e configuração de alguns componentes na arquitetura monolítica. Os registros na Tabela 4 não consideram o tempo com possíveis erros ou testes, mas sim a instalação e configuração

de acordo com as especificações existentes e consideradas em conformidade para uso do sistema. A desconsideração de tempos de ajustes ou correções no setup ocorre devido ao fato de que a adição desses tempos causaria uma variação maior entre os números de instalações realizadas, dificultando uma aproximação do tempo médio para cada procedimento, já que o procedimento foi repetido para ter uma maior fidelidade nos dados registrados.

Os tempos foram registrados utilizando o mesmo dispositivo, e foram repetidos por um profissional experiente já habituado com o ambiente do SEI. Sendo assim, buscou-se uma melhor exatidão possível para o estudo em análise. É importante observar que a cada instalação de um componente do SEI, no ambiente monolítico, é preciso obrigatoriamente a instalação do sistema operacional. Em outras palavras, cada componente só pode ser instalado em um ambiente onde já existe a instalação do sistema operacional, o que torna a instalação monolítica com um tempo de instalação e configuração muito maior em comparação com o a abordagem proposta nesse artigo com microsserviços. O tempo médio de instalação do sistema operacional foi de 8 (oito) minutos, conforme podemos visualizar na Tabela 4. Essa tabela traz o registro também do tempo médio de instalação de alguns componentes (ex., MySQL, e Apache).

Tabela 4: Registro de tempos da instalação de componentes no ambiente SEI monolítico.

Descrição do Componente	Tempo Médio (minutos)	
	Sistema Operacional	Instalação Componente
MySQL	8 min 15 s	15 min 07 s
Apache	7 min 58 s	13 min 18 s
Memcache	8 min 27 s	3 min 20 s
Solr	8 min 36 s	20 min 47 s
JOD Converter	8 min 07 s	14 min 38 s

Foi utilizado o sistema operacional CentOS 8 x86_64 Stream para facilitar o registro e a simulação. Contudo, a referência para o projeto SEI solicita um sistema operacional que demanda mais tempo para a instalação e configuração, além dos custos de licenciamento.

É importante destacar ainda que os números apresentados na Tabela 4 são referentes apenas a alguns componentes no sistema SEI na arquitetura

monolítica, e não contempla a instalação do sistema completo. Comparando esse processo com a instalação do sistema completo do SEI utilizando Kubernetes com Helm, foi registrado o tempo total de 3 (três) minutos. Dessa forma, fica evidente o ganho com a adoção da estratégia em microsserviços proposta neste trabalho. E, assim, demonstra-se a real importância da migração, pois o tempo demandado para instalação de um único componente no sistema monolítico (ex., Memcache, assumindo o sistema operacional já instalado) pode ser equiparado ao tempo demandado para a configuração de todo o sistema SEI na arquitetura em microsserviços. Fica então, demonstrado os ganhos fundamentais para a mudança do paradigma arquitetural.

É importante destacar que não é objetivo deste trabalho tratar das questões como disponibilidade de serviço. Contudo, a adoção de tecnologias de microsserviços como Kubernetes já permite uma alta disponibilidade, pois caso um componente da arquitetura venha a apresentar falha, o container é recriado automaticamente. Isso pode ser feito através de uma configuração para tal fim, permitindo assim uma estabilidade melhor da aplicação como um todo.

Desta forma, são evidentes as vantagens da migração do ambiente SEI para uma arquitetura em microsserviços. Isso deve ser feito não somente pela facilidade na instalação e configuração do ambiente, mas também pela automação e redução do tempo de configuração, demandando ainda uma menor intervenção humana. Sendo assim, a adoção da arquitetura em microsserviços reduz erros e cria possibilidades de escalabilidade de forma simplificada, tendo em vista a maior abstração na arquitetura do sistema para aqueles que irão proceder com a configuração e instalação.

7 CONCLUSÕES

A abordagem de mudanças em arquiteturas em sistemas são sempre um desafio. Além da preocupação com a manutenção de funcionalidades, manutenção e integridade, há também a necessidade de visualização de quais benefícios essas mudanças podem proporcionar, principalmente em softwares públicos, que possuem um papel relevante para a sociedade.

Não obstante disso, o projeto SEI Kubernetes proposto traz a mudança de um ambiente já promulgado e utilizado em diversas entidades públicas. Essa mudança busca resolver a

escalabilidade do sistema, bem como abstrair a adoção, pois tecnologias em microsserviços trazem consigo ganhos em termos da otimização dos recursos computacionais existentes e, também, possibilitam um processo de automação que minimiza erros e reduz a complexidade para a adoção do sistema SEI. Os comparativos da instalação de componentes do SEI nos dois tipos de arquiteturas aqui apresentados são mais do que suficientes para demonstrar a importância dessa nova abordagem em microsserviços.

É importante mencionar que não foram abordadas questões relacionadas à segurança da informação. Destaca-se, que este pode ser um futuro direcionamento deste trabalho, bem como um comparativo no uso dos ambientes em produção (microsserviço e monolítico) no intuito de identificar e avaliar o comportamento do sistema em termos de escalabilidade e de recursos computacionais em utilização. Desta forma, há a possibilidade em estender o trabalho atual para se realizar a análise através de comparação entre o ambiente SEI na arquitetura monolítica em contraste com a microsserviços. Além disso, essa pesquisa não teve como foco a análise de questões como latência e desempenho, sendo esses possíveis direcionamentos futuros. Outras limitações é que não foram consideradas evolução do sistema e a manutenção dele, bem como a necessidade de possíveis adaptações em casos de mudanças por legislação ou inovações tecnológicas.

As possibilidades de trabalho futuro podem ainda serem estendidas as questões de governança, adesão a Legislação vigente para proteção e integridade dos dados, bem como aspectos que visem testar e avaliar a integridade, a confidencialidade e a disponibilidade dos dados existentes no sistema. Como um futuro direcionamento dessa pesquisa, pode haver a análise dos impactos da adoção do Kubernetes em um ambiente legado, no mesmo cenário analisado por este artigo. É possível ainda estudar e avaliar outras ferramentas de uso governamental ou não, sendo o sistema objeto deste estudo uma referência para outras migrações das estruturas arquiteturais.

AGRADECIMENTOS

Os autores gostariam de agradecer ao programa de Capacitação para Consolidação de Novas Tecnologias de Computação promovido pela Universidade de Pernambuco (UPE) em parceria com o suporte financeiro da Superintendência Nacional de Desenvolvimento do Nordeste (SUDENE) que possibilitou o desenvolvimento acadêmico e profissional na Especialização em Cloud Computing,

área tão crescente e importante para o Estado de Pernambuco, e conseqüentemente pela possibilidade de realização desta pesquisa.

REFERÊNCIAS

[1] Tribunal de Justiça do Estado de Pernambuco (TJPE). **Manual do Usuário SEI**. Disponível em: <https://www.tjpe.jus.br/documents/1592722/0/Manual_do_usuario_SEI_3.1.4.pdf> Acesso em 15 jul. 2021, 20:08.

[2] MORRIS, Kief. **Infrastructure as Code: dynamic systems for the cloud age**. O'Reilly. 2ª Edição. California – US, 2020.

[3] KOCHER, Parminder Singh. **Microservices and Containers**. Pearson Education, 2018.

[4] MOUAT, Adrian. **Using Docker: Developing and Deploying Software with Containers**. United States: O'Reilly Media, 2015.

[5] ARUNDEL, John; DOMINGUS, Justin. **DevOps nativo de nuvem com Kubernetes**. São Paulo - SP: O'Reilly Media, 2019.

[6] CHANDRASEKARAN, K.. **Essentials of Cloud Computing**. Philippines, Taylor & Francis, 2014.

[7] BRETSCHEIDER, Stuart I.; MERGEL, Ines. **Technology And Public Management Information Systems: Where We Have Been and Where We Are Going**. Disponível em <<https://www.researchgate.net/publication/233993585>>. Acesso em 24 jun. 2021, 18:45:05

[8] POPOV, S. B.; KHRIPUNOV, P. V.. **Digital Transformation Legacy Social Service Information System**. 2019 J. Phys.: Conf. Ser. 1368 052019.

[9] SRIKAEW, Pankamol; KIM, Inkyu. **A Microservice Development for Document Management System**. 2017. Publicado em: 4th International Conference on Computer Applications and Information Processing Technology (CAIPT).

[10] CARDOSO, Hélio José Almeida. **eDocuments as Microservices**. Porto - PT, 2020. Instituto Superior de Engenharia do Porto.

[11] SOUZA, Gabriel de. **Proposta de nuvem privada multicâmpus para o IFSC usando orquestração de contêineres**. São José - SC. 2018. Instituto Federal de Santa Catarina.

[12] Site do Software Público. **Treinamento SEI Implantar**. Disponível em:
<https://softwarepublico.gov.br/social/articles/0005/3867/Treinamento_SEI_Implantar_20170323_vSEGES.pdf>. Acesso em 15 de Jul. 2021, 17:25:00

[13] MOUAT, Adrian. **Using Docker: Developing and deploying software with containers**. California - USA: O'Reilly Media, 2016.

[14] Site GitHub. **Projeto SEI Kubernetes**. Disponível em:
<<https://github.com/seikubernetes/projeto-sei>>. Acesso em 18 de jul. 2021, 22:08:45