

DANIEL NORIO TAKASU REBELO
LUCAS ARTHUR FELGUEIRAS
LUIZ GUSTAVO DOS SANTOS
VICTOR FRANÇA FERREIRA

COMPUTAÇÃO EM NUVEM: INTRODUÇÃO

São Paulo
2018

**DANIEL NORIO TAKASU REBELO
LUCAS ARTHUR FELGUEIRAS
LUIZ GUSTAVO DOS SANTOS
VICTOR FRANÇA FERREIRA**

COMPUTAÇÃO EM NUVEM: INTRODUÇÃO

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para a
disciplina de Engenharia de Sistemas.

São Paulo
2018

**DANIEL NORIO TAKASU REBELO
LUCAS ARTHUR FELGUEIRAS
LUIZ GUSTAVO DOS SANTOS
VICTOR FRANÇA FERREIRA**

COMPUTAÇÃO EM NUVEM: INTRODUÇÃO

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para a
disciplina de Engenharia de Sistemas.

Área de Concentração:
Engenharia de Sistemas

Orientador:
Prof^ª. Dr^ª. Selma Shin Shimizu
Melnikoff

RESUMO

A tecnologia de computação em nuvem é uma tendência para as novas aplicações do mundo. Grandes empresas estão migrando suas estruturas para a nuvem, outras constroem ambientes para abrigar essas novas aplicações. O objetivo do trabalho consiste em compreender melhor como funciona essas tecnologias, sua motivação e as principais opções disponíveis no mercado para uso.

Palavras-Chave – Nuvem, Arquitetura, Computação, Engenharia, Programa.

ABSTRACT

Cloud computing technology is a trend for new applications around the world. Large companies are migrating their structures to the cloud, others build environments to house these new applications. The objective of the work is to better understand how these technologies work, their motivation and the main options available in the market to use.

Keywords – Cloud, Architecture, Computing, Enginnering, Software.

LISTA DE FIGURAS

1	Arquitetura Básica do Heroku(1)	22
2	Ligação entre <i>router</i> e <i>dynos</i> do Heroku(1)	23
3	Exemplo do <i>Dashboard</i> do Heroku com seus recursos	24

LISTA DE TABELAS

SUMÁRIO

Parte I: INTRODUÇÃO	8
1 Objetivo	9
1.1 Introdução	9
2 Motivação	10
2.1 Contexto	10
3 Justificativa	11
3.1 Introdução	11
Parte II: ASPECTOS CONCEITUAIS	12
4 Necessidades	13
4.1 Introdução	13
5 Funcionamento Básico	14
5.1 Introdução	14
6 Conceitos Usados	15
6.1 Introdução	15
Parte III: ASPECTOS TÉCNICOS	16
7 Estrutura	17
7.1 Introdução	17
8 Evolução Tecnológica	18
8.1 Introdução	18

Parte IV: SOLUÇÕES DO MERCADO	19
9 Amazon Web Services	20
9.1 Introdução	20
10 Google Cloud	21
10.1 Introdução	21
11 Heroku	22
11.1 Introdução	22
11.2 Funcionamento básico	22
11.3 Tarifação	23
11.4 <i>Containers</i>	24
11.5 Conclusão	25
12 Azure	26
12.1 Introdução	26
Referências	27

PARTE I

INTRODUÇÃO

1 OBJETIVO

1.1 Introdução

2 MOTIVAÇÃO

2.1 Contexto

3 JUSTIFICATIVA

3.1 Introdução

PARTE II

ASPECTOS CONCEITUAIS

4 NECESSIDADES

4.1 Introdução

5 FUNCIONAMENTO BÁSICO

5.1 Introdução

6 CONCEITOS USADOS

6.1 Introdução

PARTE III

ASPECTOS TÉCNICOS

7 ESTRUTURA

7.1 Introdução

8 EVOLUÇÃO TECNOLÓGICA

8.1 Introdução

PARTE IV

SOLUÇÕES DO MERCADO

9 AMAZON WEB SERVICES

9.1 Introdução

10 GOOGLE CLOUD

10.1 Introdução

11 HEROKU

11.1 Introdução

O Heroku é uma plataforma de computação em nuvem conhecida no mercado, com a possibilidade de subir aplicações nas linguagens Ruby, Node.js, Java, Python, Clojure, Scala, Go e PHP. O grande destaque do Heroku está na facilidade em subir uma aplicação com facilidade e de maneira gratuita, o que possibilita testar e validar ideias básicas antes de escalar de fato.

A estrutura básica do Heroku funciona com o uso de *dynos*, que servem tanto para hospedar sua aplicação principal quanto máquinas auxiliares para serviços externos e/ou paralelos. Porém, recentemente o Heroku disponibilizou também a estrutura de *containers* em Docker (tecnologia de gerenciamento de *containers*) (assim como seus principais concorrentes), gerando maior flexibilidade para o serviço.

11.2 Funcionamento básico

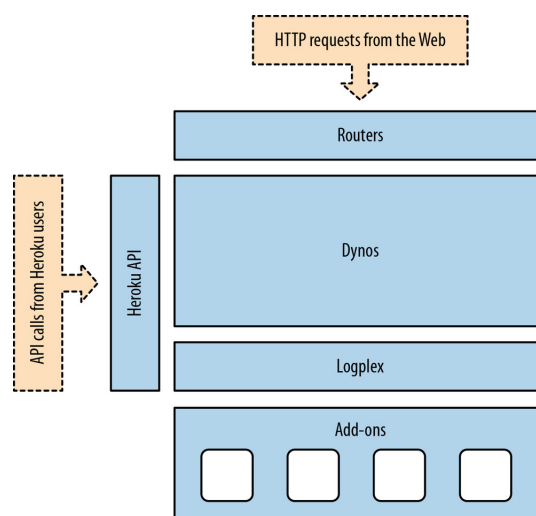


Figura 1: Arquitetura Básica do Heroku(1)

O funcionamento do Heroku consiste no uso de *dynos* para hospedar as aplicações e nos *routers* para tratar e encaminhar as requisições dos usuários. Além disso, o próprio Heroku disponibiliza extensões para gerenciar sua aplicação, como por exemplo o gerenciador de IP estático, ou o banco de dados necessário para a aplicação.

Um *dyno* é um *container* pronto, com 512MB de memória RAM, responsável por abrigar uma ou mais instâncias de sua aplicação, permitindo escalabilidade e tolerância de erros. Além disso, os *dynos* permitem rodar tarefas de sua aplicação à parte, como filas, requisições assíncronas, entre outros.

Já os *routers* são responsáveis por gerenciar os acessos dos usuários às aplicações correspondentes, dado que não é padrão do Heroku estruturar IP estático para cada *dyno*. Ou seja, a estrutura é responsável por fazer sua aplicação funcionar corretamente, sem acessar aplicações alheias.

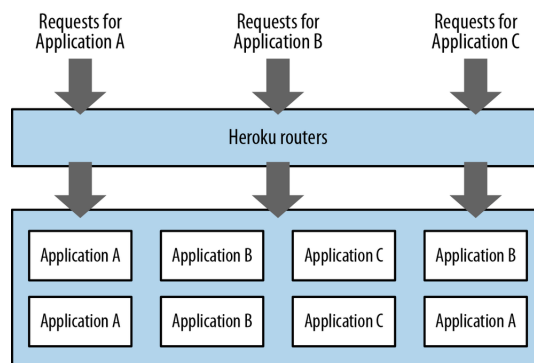


Figura 2: Ligação entre *router* e *dynos* do Heroku(1)

Graças a essa estrutura, é possível, com facilidade, gerenciar versionamento e escalar a aplicação de maneira horizontal, com novos *dynos* gerenciados pelo *router*, dando uma popularidade considerável ao Heroku, em especial nas aplicações onde velocidade de validar hipóteses é o principal foco.

11.3 Tarifação

O Heroku possui um diferencial em relação à tarifação: possui um plano básico gratuito que possibilita testar aplicações de maneira fácil e sem dificuldades de expansão. Essencialmente, a cobrança do Heroku ocorre via uso de seus *dynos*-hora. Além disso, há a cobrança pelas extensões usadas dentro da aplicação, onde, no geral, há um plano gratuito de testes ou até para projetos pequenos funcionarem com tranquilidade sem necessidades de escalabilidade.

Para qualquer usuário do Heroku, é disponibilizado um pacote de 750 *dynos*-hora durante um mês, ou seja, uma aplicação gratuita usando um simples *dyno* pode durar tranquilamente. Além disso, o Heroku de uso gratuito limita o tempo em que um *dyno* fica ligado de maneira ociosa: passando trinta minutos desde a última requisição feita pelo cliente (e repassada pelo *router*), o *dyno* é derrubado, sendo religado após uma nova requisição.

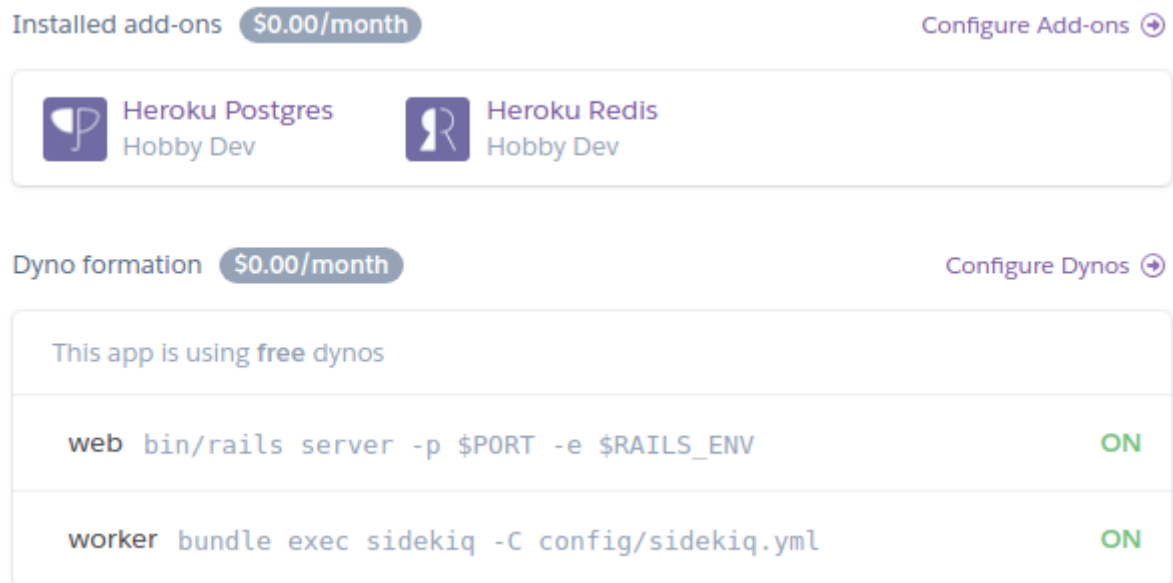


Figura 3: Exemplo do *Dashboard* do Heroku com seus recursos

Além da cobrança do uso dos *dynos*, há também a cobrança para cada extensão utilizada por sua aplicação, o que de fato encarece o custo final para escalar uma aplicação. Por exemplo, o banco de dados relacional padrão do Heroku é gratuito até certo limite de dados, passando desse limite, há a extensão do plano que permite, além de mais dados armazenados, aumenta o fluxo possível de acesso ao banco.

11.4 *Containers*

Recentemente, o Heroku possibilitou em seu catálogo de soluções o uso de estruturas isoladas customizadas: os *containers*. *Containers*, segundo o site do Docker(2) (em inglês):

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

Ou seja, são imagens autossuficientes e virtualizadas que permitem rodar aplicações prontas nos mais diversos dispositivos, nos moldes da virtualização.

O Heroku possibilitou, em 2017, uma maneira eficiente de receber *containers* em seu serviço, com o **Heroku Container Registry**(3). Esse serviço facilita colocar em produção as máquinas isoladas contruídas com o uso do *Docker*, aumentando as possibilidades de operação com a plataforma. A tarifação do serviço segue o mesmo padrão do Heroku tradicional, dado o fato de que *dynos* são uma espécie de *containers*.

11.5 Conclusão

O Heroku é uma solução que se destaca pela velocidade e facilidade de colocar uma ideia em produção, de maneira gratuita, com possibilidades eficientes de escalabilidade. Porém, financeiramente, não é a melhor solução, com uma tarifação elevada para escalar a aplicação. Além disso, sua estrutura atende apenas aplicações prontas, não permitindo outras estruturas avançadas como o uso de *containers*, sendo incorporado ao catálogo de produtos recentemente.

12 AZURE

12.1 Introdução

REFERÊNCIAS

- 1 SCHNEEMAN, R.; MIDDLETON, N. *Heroku: Up and Running*. 2013. Disponível em: <https://www.safaribooksonline.com/library/view/heroku-up-and/9781449341381/ch02.html>.
- 2 DOCKER. *What is a Container*. 2018. Disponível em: <https://www.docker.com/what-container>.
- 3 HEROKU. *Container Registry & Runtime (Docker Deploys)*. 2018. Disponível em: <https://devcenter.heroku.com/articles/container-registry-and-runtime>.