



Shortly



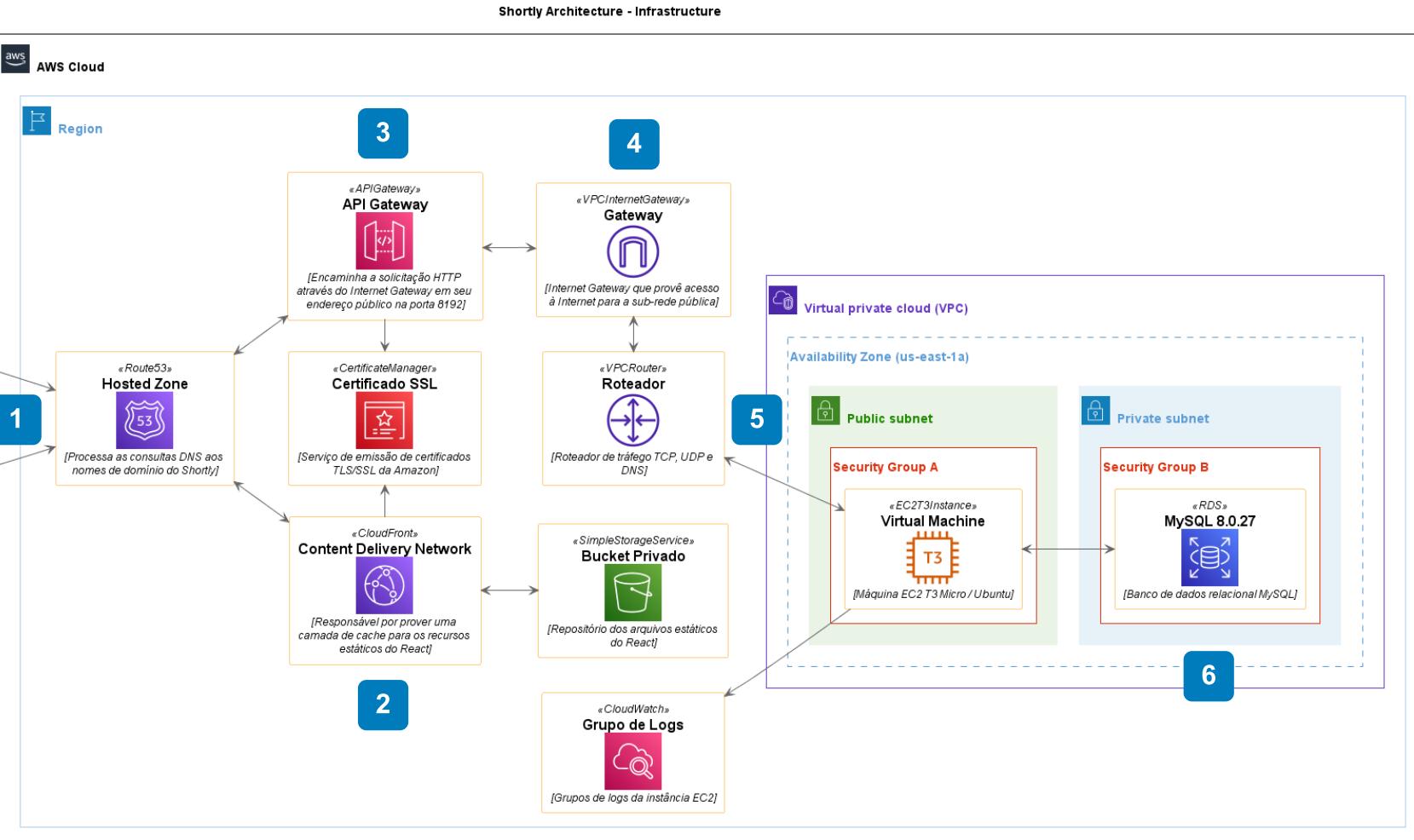
Arquitetura de Referência

Versão On-premises Elastic Compute Cloud (Amazon EC2)

<https://shortly.digital/>

Visão Geral da Arquitetura do Shortly, Parte 1 de 10

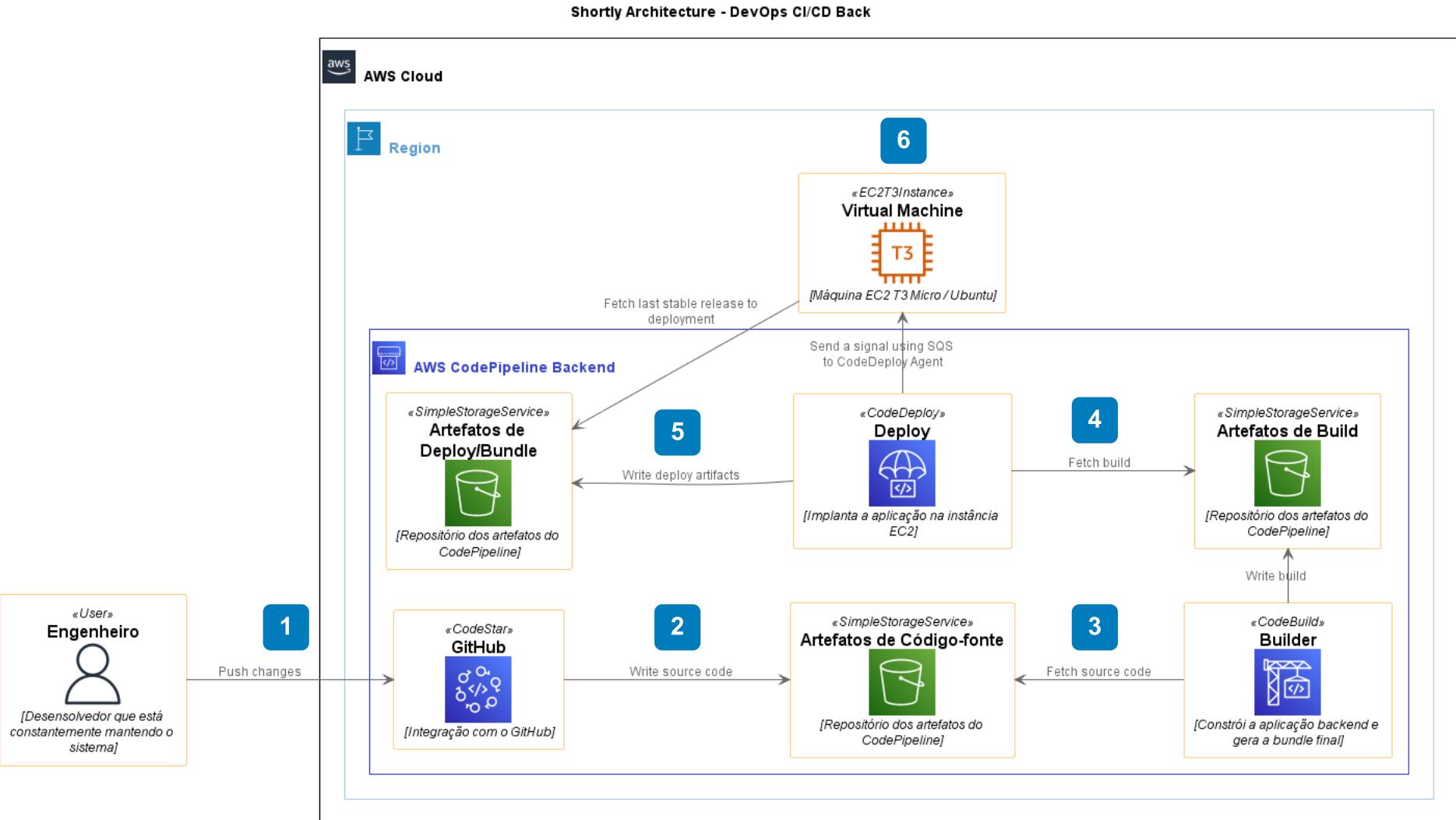
O Shortly é um sistema que utiliza os serviços EC2, RDS/MySQL e API Gateway da AWS. O EC2 fornece capacidade de computação escalável, o RDS/MySQL armazena dados eficientemente e o API Gateway atua como ponto de entrada seguro para as solicitações dos usuários. Essa combinação garante uma arquitetura eficiente e escalável, proporcionando uma experiência de baixa latência e alta disponibilidade, além de baixos custos de infraestrutura.



- O usuário envia uma solicitação para o sistema, de alguma funcionalidade específica como encurtamento de links.
- A solicitação de DNS do usuário é resolvida pelo serviço Route53, que encaminha a solicitação para o CloudFront. O CloudFront é uma rede de distribuição de conteúdo (CDN) que fornece uma camada de cache para os recursos estáticos do aplicativo (arquivos do React).
- O Route53 também encaminha a solicitação para o API Gateway. O API Gateway é responsável por receber, gerenciar, filtrar e encaminhar as solicitações para o Internet Gateway da VPC em que a instância EC2 executa.
- O Internet Gateway recebe a conexão TCP proveniente do API Gateway e encaminha para o Router que encaminha os pacotes TCP, UDP e DNS para seu destino apropriado.
- O Router lê a tabela de roteamento especificada no Terraform e encaminha as conexões TCP/UDP que estejam sendo feitas para a porta 8192 da instância EC2. A instância EC2, por sua vez, atende à solicitação e devolve uma resposta para o Router que devolve para o Internet Gateway e assim por diante.
- O banco de dados MySQL provido pelo serviço Amazon RDS é utilizado pela instância EC2 a partir de uma sub-rede pública. O banco de dados foi propositalmente implantado em uma sub-rede privada para evitar a exposição da porta 3306 à exploits e tentativas arbitrárias de violação, reforçando o compromisso com a proteção dos dados do usuário e a adequação à LGPD.

Visão Geral da Arquitetura do Shortly, Parte 2 de 10

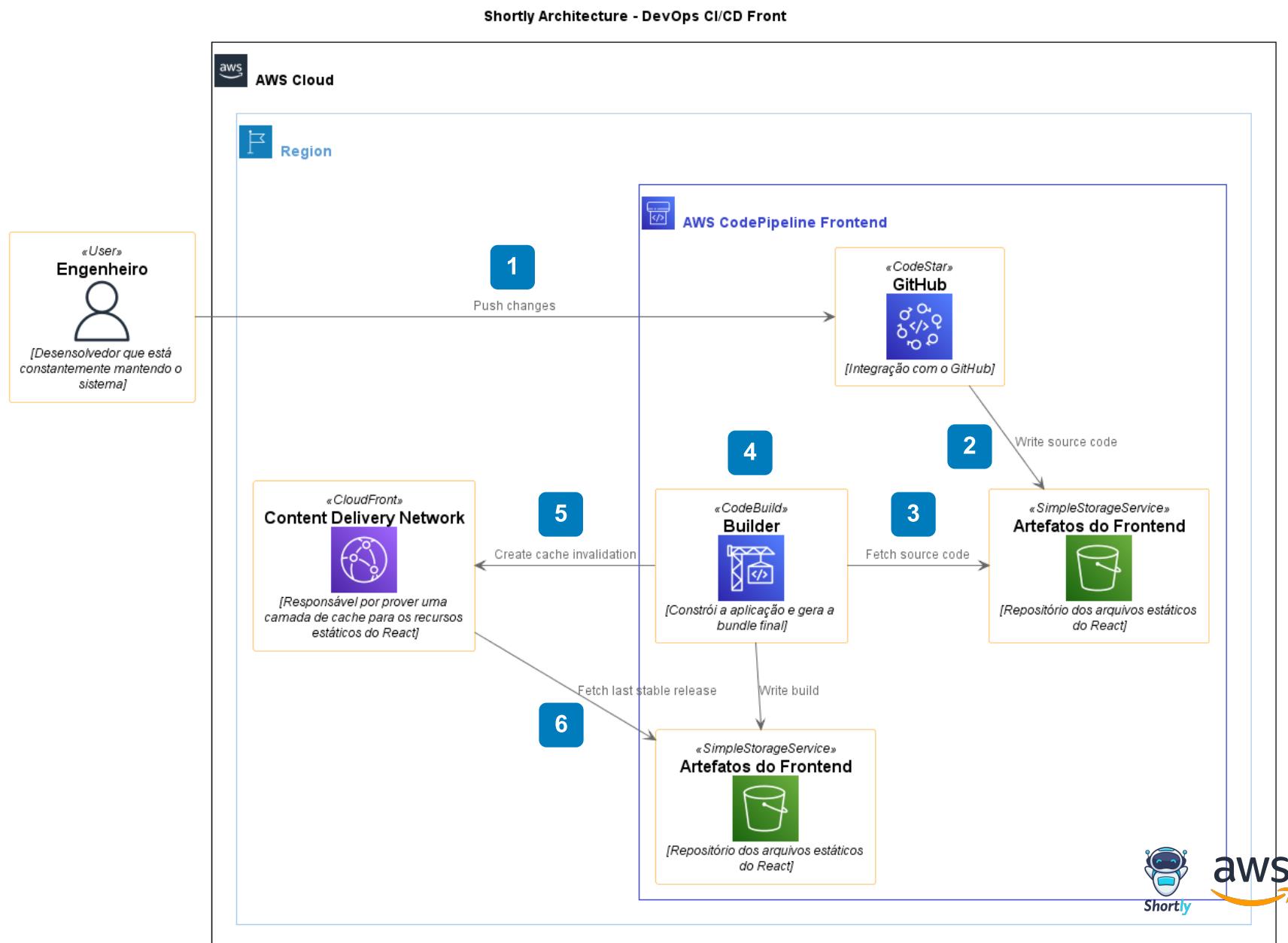
O Shortly utiliza o CodePipeline, CodeStar, CodeBuild e o CodeDeploy para realizar as implantações no ambiente On-premises do Amazon EC2. Devido a essa combinação de serviços é possível montar um fluxo de trabalho completamente personalizado para executar as mais diversas atividades que variam desde o processo de build até a implantação dentro da instância Amazon EC2. As variáveis de ambiente do processo de build até a implantação são armazenadas no AWS Systems Manager Parameter Store.



- O engenheiro envia alterações para o repositório do GitHub, acionando a integração do CodePipeline. Essas alterações forçam o CodeStar a realizar o clone do repositório.
- Os artefatos gerados pelo CodeStar após a clonagem do repositório são então escritos no Bucket S3 do CodePipeline. Esse Bucket S3 de artefatos é utilizado ao longo do processo de DevOps para repassar os artefatos de uma etapa anterior para a etapa seguinte.
- O CodeBuild realiza a leitura dos artefatos gerados pelo CodeStar no processo de clonagem do repositório. O CodeBuild então constrói a aplicação NestJS e escreve os artefatos de saída novamente no Bucket S3 de artefatos de build.
- O CodeDeploy realiza a leitura dos artefatos gerados pelo CodeBuild no processo de construção da aplicação NestJS. O CodeDeploy então interpreta os scripts de implantação, executa ações no projeto que incluem baixar as dependências do sistema.
- O CodeDeploy escreve os artefatos de saída no Bucket S3 do CodePipeline. Após a escrita dos artefatos o CodeDeploy envia um sinal SQS para o CodeDeploy Agent que está instalado no sistema operacional da instância EC2. O CodeDeploy Agent então inicia o processo de cópia dos artefatos do Bucket S3 para dentro da instância.
- Após ter recebido o sinal SQS, o CodeDeploy Agent copia os artefatos do Bucket e implanta a aplicação com base nas especificações do arquivo **appspec.yml** presente no diretório raiz do projeto.

Visão Geral da Arquitetura do Shortly, Parte 3 de 10

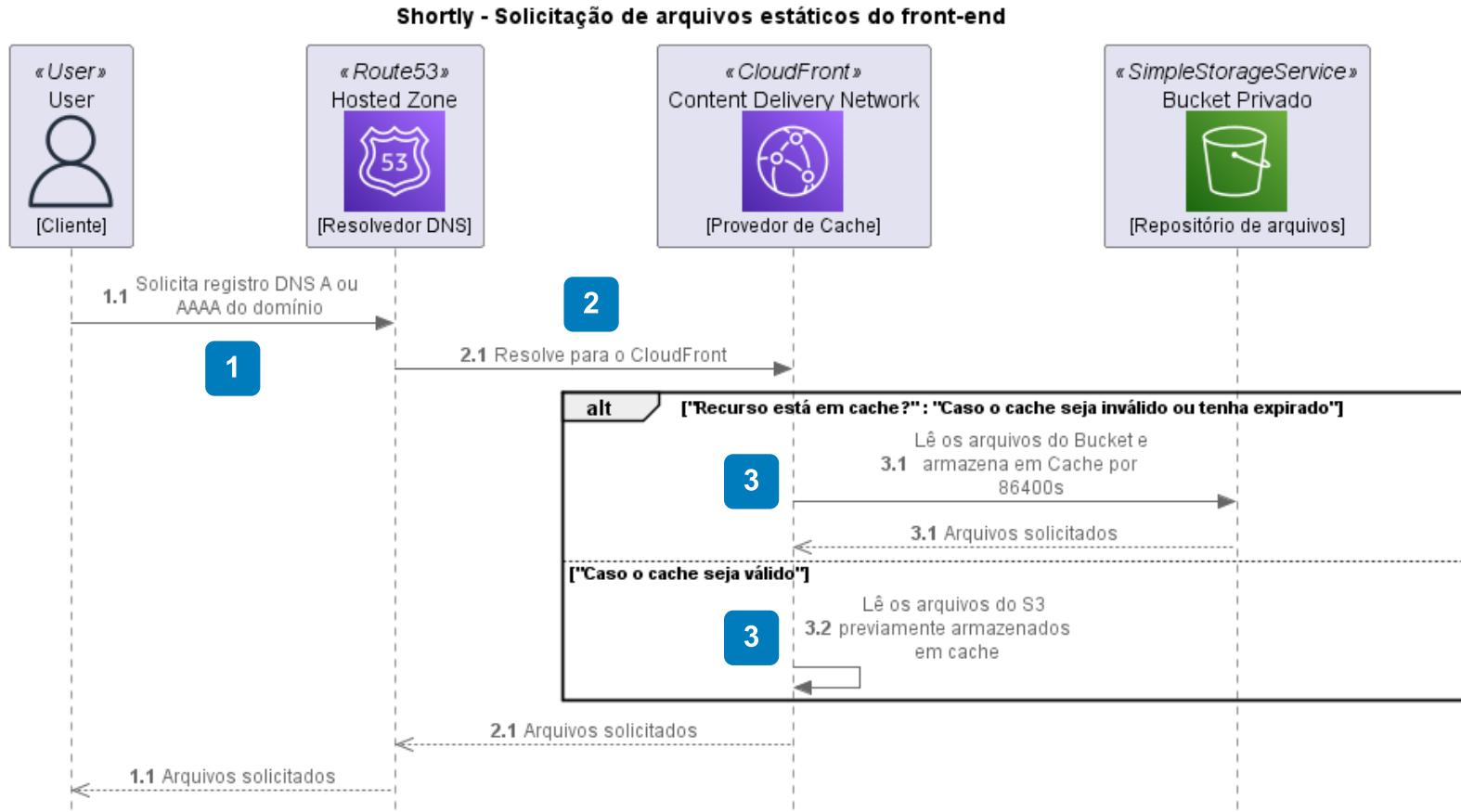
O Shortly utiliza também o CodePipeline, CodeStar e CodeBuild para a construção e implantação da aplicação React utilizando Vite.js. Devido à configuração personalizada do build e às boas práticas adotadas na construção do Pipeline, a aplicação React pode ser rapidamente atualizada por meio de merges com a ramificação master do repositório no GitHub. Esse processo leva tipicamente 2 minutos para ser executado. Assim como no build da API RESTful, as variáveis de ambiente são armazenadas no AWS Systems Manager Parameter Store.



- 1 O engenheiro envia alterações para o repositório do GitHub, acionando a integração do CodePipeline.
- 2 O processo de construção do Frontend inicia com o CodeStar que realiza o clone do repositório de origem e escreve os arquivos do clone em um Bucket S3 de artefatos do Frontend que será utilizado na etapa sucessora de construção da aplicação React pelo CodeBuild.
- 3 O CodeBuild lê os arquivos do clone do repositório de origem para começar o processo de Build do Frontend.
- 4 O CodeBuild inicia a construção do Frontend React executando os scripts NPM que invocam o Vite para gerar a build do projeto. Após construir o Frontend, os artefatos são novamente escritos no Bucket S3 de artefatos do Frontend.
- 5 O CodeBuild cria o que se chama de invalidação de cache na distribuição CloudFront que provê os arquivos do Bucket S3. A invalidação de cache serve para forçar o CloudFront a reconhecer que os arquivos devem ser colocados em cache novamente.
- 6 Uma vez que o CloudFront recebe uma invalidação de cache, os arquivos estáticos do Frontend são lidos novamente do Bucket S3 na próxima vez que o CloudFront necessitar provê-los. Esse processo chama-se re-hidratação do cache. Os arquivos também são lidos e armazenados em cache quando há uma falta compulsória.

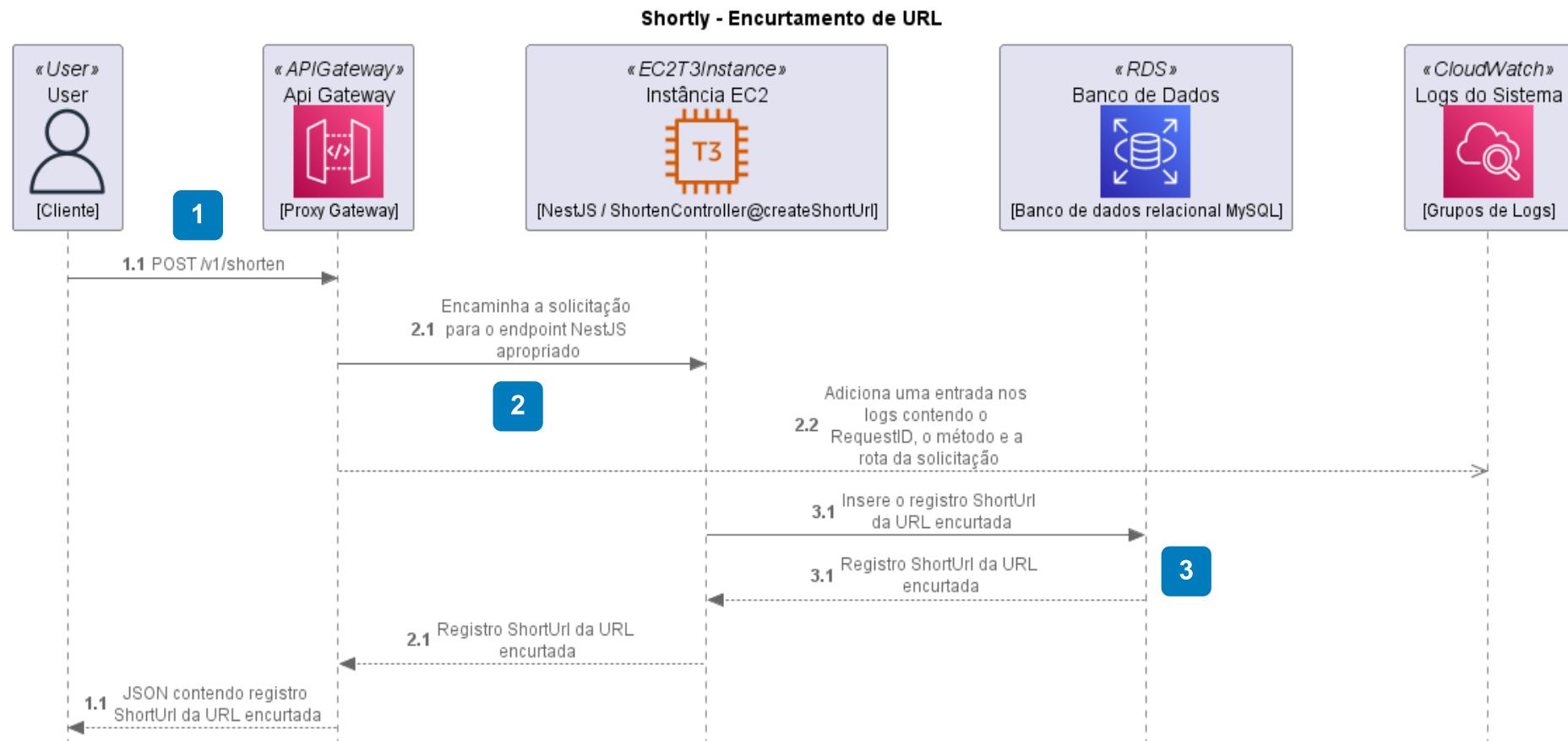
Visão Geral da Arquitetura do Shortly, Parte 4 de 10

O diagrama de sequência a seguir representa a interação de um usuário com um front-end web hospedado em um Bucket S3 privado, fornecido por uma distribuição CloudFront. O processo envolve quatro entidades principais: o ator (usuário), o resolvedor DNS do Route53, o Amazon CloudFront e o Bucket Privado do Amazon S3.



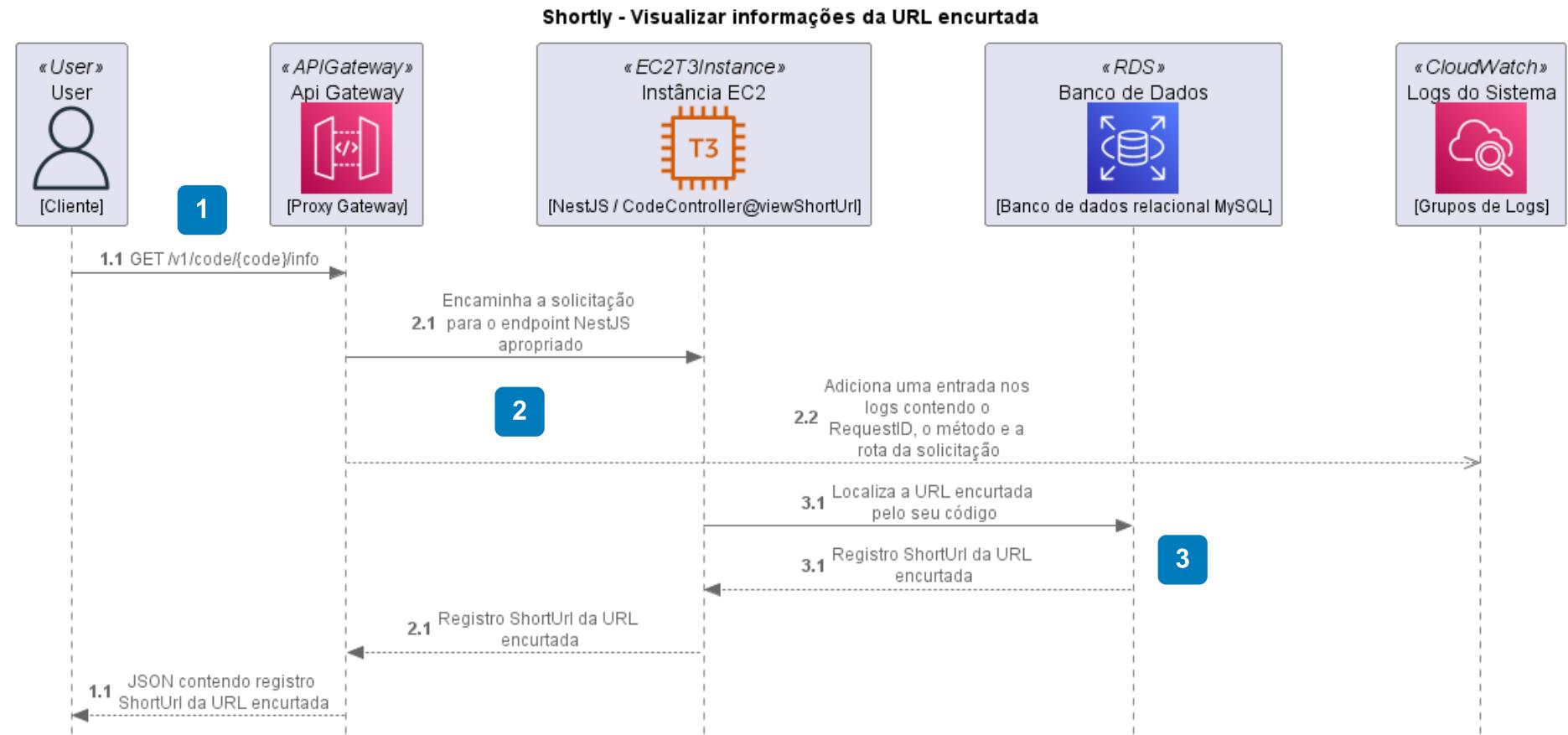
Visão Geral da Arquitetura do Shortly, Parte 5 de 10

O diagrama de sequência abaixo ilustra a interação entre um usuário, um endpoint fornecido pelo API Gateway e uma arquitetura composta por serviços EC2, RDS/MySQL e API Gateway. A interação ocorre através de uma instância EC2 On-premises que se comunica com um banco de dados MySQL hospedado no RDS. No esquema abaixo vemos a interação do usuário para **Encurtar uma URL** no Shortly.



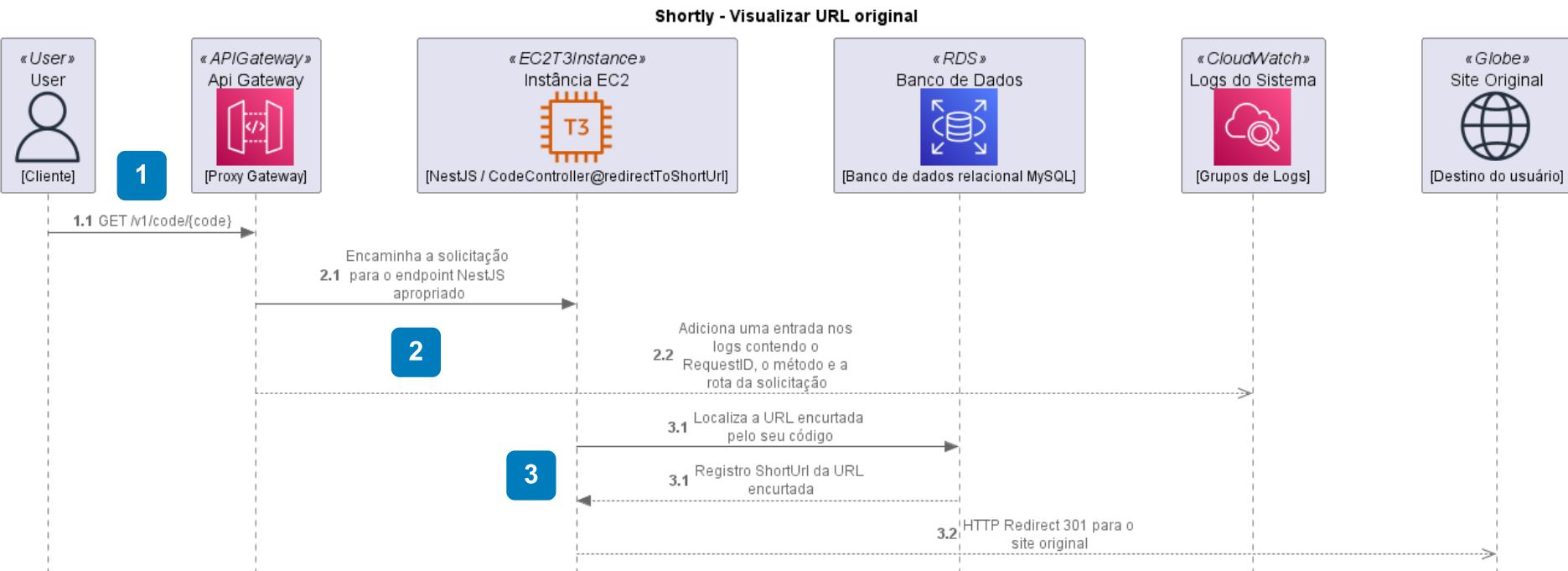
Visão Geral da Arquitetura do Shortly, Parte 6 de 10

O diagrama de sequência abaixo ilustra a interação entre um usuário, um endpoint fornecido pelo API Gateway e uma arquitetura composta por serviços EC2, RDS/MySQL e API Gateway. A interação ocorre através de uma instância EC2 On-premises que se comunica com um banco de dados MySQL hospedado no RDS. No esquema abaixo vemos a interação do usuário para **Visualizar informações da URL encurtada** no Shortly.



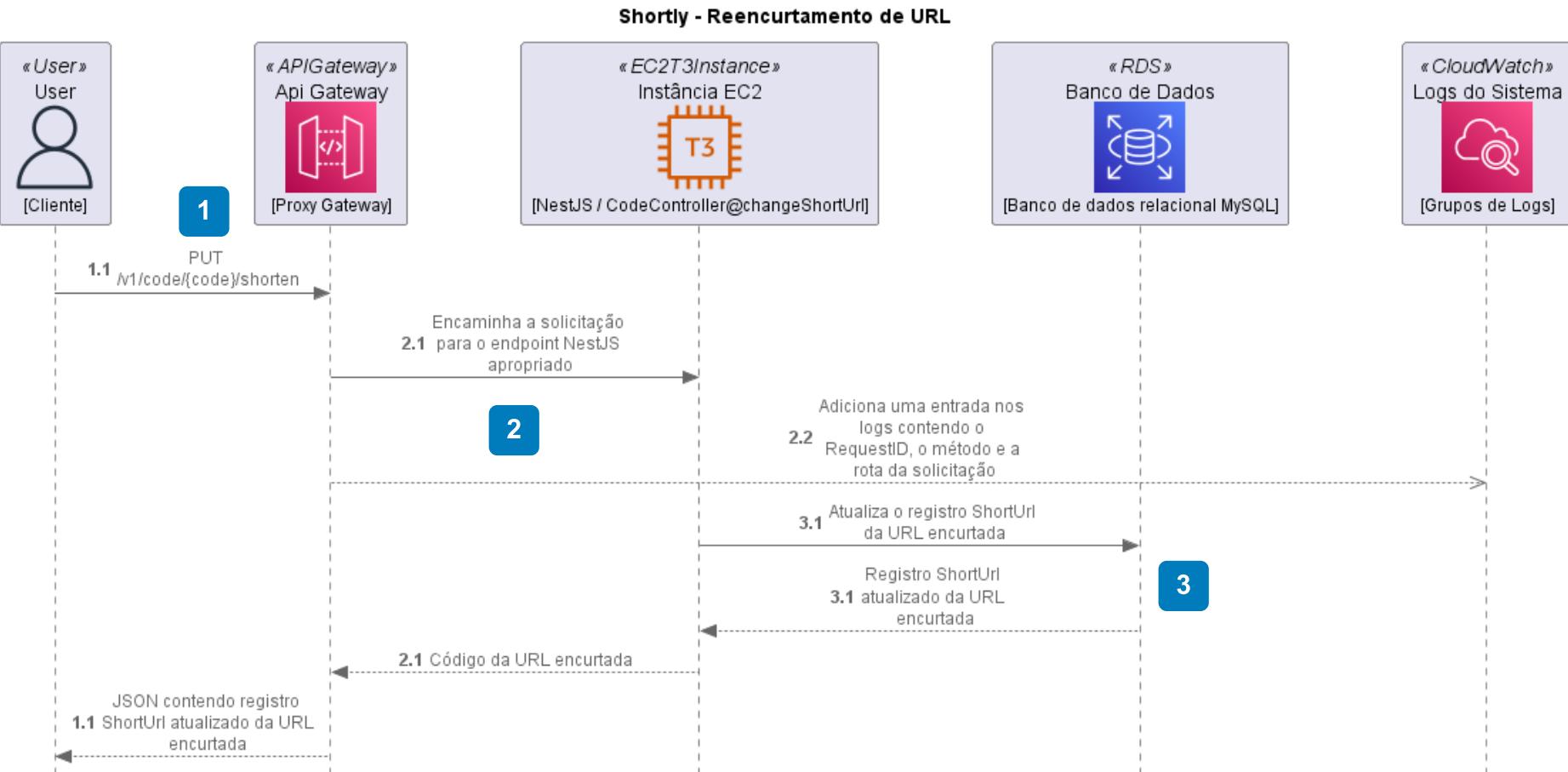
Visão Geral da Arquitetura do Shortly, Parte 7 de 10

O diagrama de sequência abaixo ilustra a interação entre um usuário, um endpoint fornecido pelo API Gateway e uma arquitetura composta por serviços EC2, RDS/MySQL e API Gateway. A interação ocorre através de uma instância EC2 On-premises que se comunica com um banco de dados MySQL hospedado no RDS. No esquema abaixo vemos a interação do usuário para **Visualizar a URL original** no Shortly.



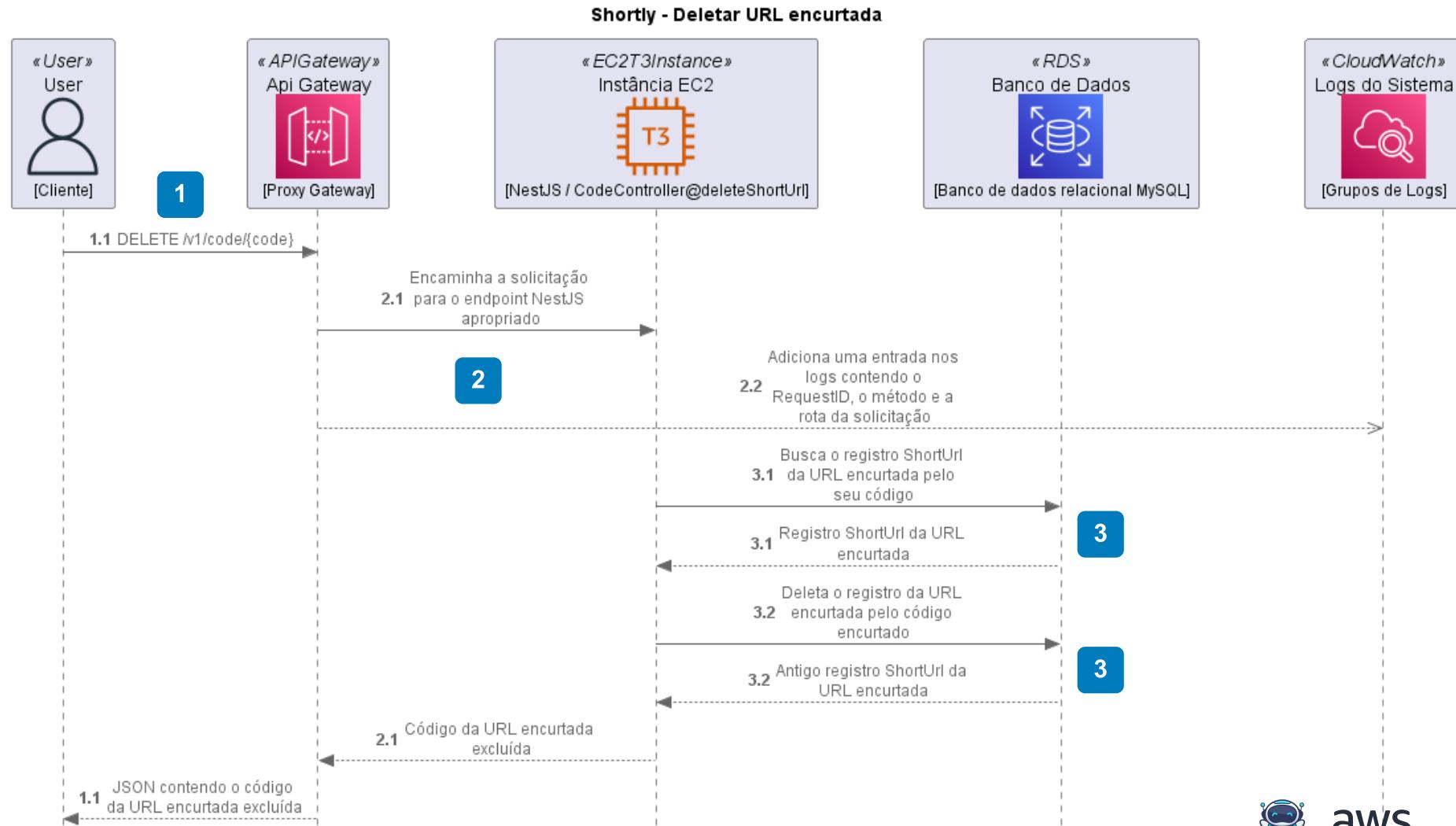
Visão Geral da Arquitetura do Shortly, Parte 8 de 10

O diagrama de sequência abaixo ilustra a interação entre um usuário, um endpoint fornecido pelo API Gateway e uma arquitetura composta por serviços EC2, RDS/MySQL e API Gateway. A interação ocorre através de uma instância EC2 On-premises que se comunica com um banco de dados MySQL hospedado no RDS. No esquema abaixo vemos a interação do usuário para **Reencurtar uma URL** no Shortly.



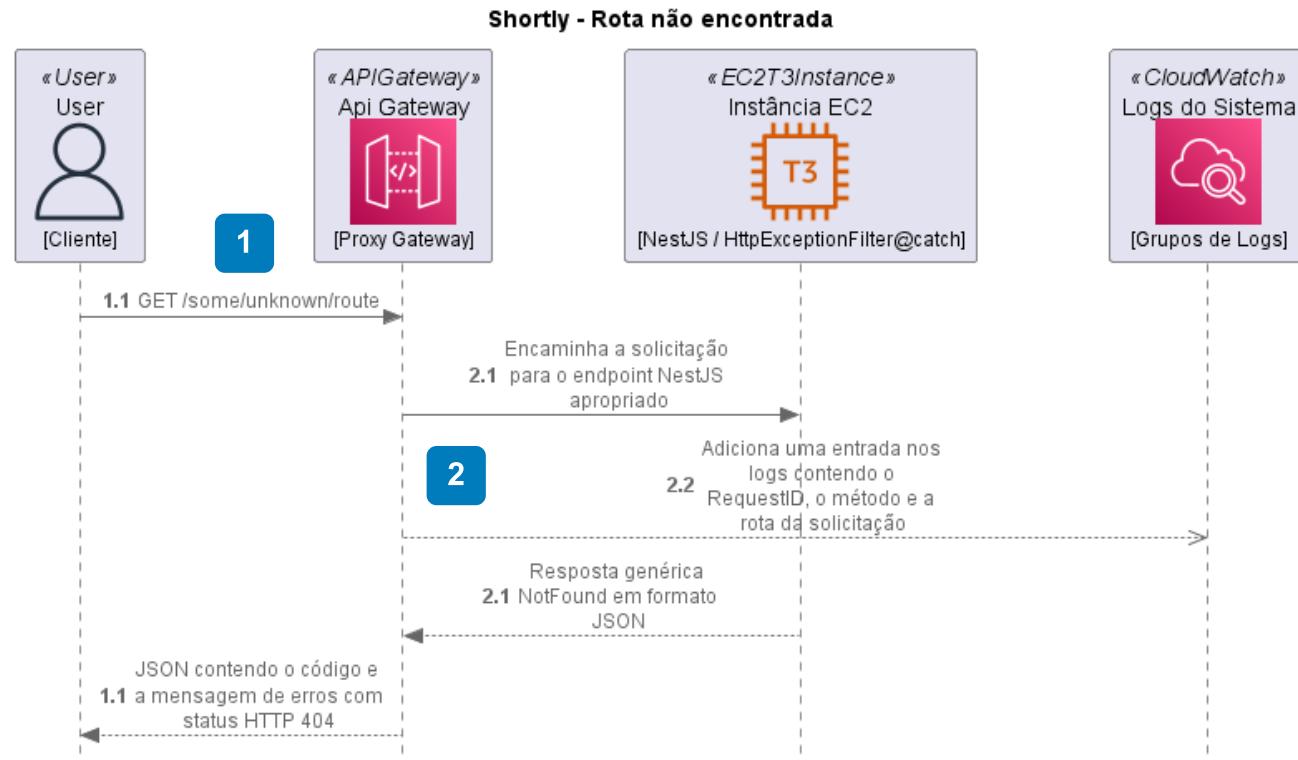
Visão Geral da Arquitetura do Shortly, Parte 9 de 10

O diagrama de sequência abaixo ilustra a interação entre um usuário, um endpoint fornecido pelo API Gateway e uma arquitetura composta por serviços EC2, RDS/MySQL e API Gateway. A interação ocorre através de uma instância EC2 On-premises que se comunica com um banco de dados MySQL hospedado no RDS. No esquema abaixo vemos a interação do usuário para **Deletar uma URL encurtada** no Shortly.



Visão Geral da Arquitetura do Shortly, Parte 10 de 10

O diagrama de sequência abaixo ilustra a interação entre um usuário, um endpoint fornecido pelo API Gateway e uma arquitetura composta por serviços EC2, RDS/MySQL e API Gateway. A interação ocorre através de uma instância EC2 On-premises que se comunica com um banco de dados MySQL hospedado no RDS. No esquema abaixo vemos a interação do usuário para **Acessar uma rota não encontrada** no Shortly.





Arquitetura de Referência

Versão On-premises Elastic Compute Cloud (Amazon EC2)