

**Documentação -**

**Infraestrutura de TI**

Texto

Descrição gerada automaticamente

**www.audaztecnologia.com.br**

INDICE

[1. Informações gerais 4](#_Toc174116260)

[1.1 Responsáveis 4](#_Toc174116261)

[1.2 Dados do cliente 4](#_Toc174116262)

[1.3 Controle do documento 5](#_Toc174116263)

[2. Resumo executivo 6](#_Toc174116264)

[2.1 Objetivo 6](#_Toc174116265)

[3. Visão Geral 7](#_Toc174116266)

[3.1 Arquitetura da Aplicação 7](#_Toc174116267)

[3.2 Arquitetura da Infraestrutura 8](#_Toc174116268)

[3.2.1 Frontend 8](#_Toc174116269)

[3.2.2 Backend 8](#_Toc174116270)

[3.2.3 Containers 8](#_Toc174116271)

[3.2.4 Dados 8](#_Toc174116272)

[3.2.5 Rede 9](#_Toc174116273)

[3.2.6 Segurança 9](#_Toc174116274)

[3.2.7 Monitoramento e Observabilidade 9](#_Toc174116275)

[3.3 Infraestrutura 10](#_Toc174116276)

[3.3.1.1 Object Storage 11](#_Toc174116277)

[3.3.2 Sitebliz Backend 12](#_Toc174116278)

[3.3.2.1 Virtual machines 12](#_Toc174116279)

[3.3.3 Dados 12](#_Toc174116280)

[3.3.3.1 SQL databases 12](#_Toc174116281)

[3.3.4 Kubernetes (k3s) 13](#_Toc174116282)

[3.3.4.1 Pods 14](#_Toc174116283)

[3.3.4.2 Service 14](#_Toc174116284)

[3.3.4.3 Ingress 15](#_Toc174116285)

[3.3.5 Rede 17](#_Toc174116286)

[3.3.5.1 DNS Zone 17](#_Toc174116287)

[3.3.5.2 Rede Virtual 18](#_Toc174116288)

[3.3.5.3 Virtual network gateway 19](#_Toc174116289)

[3.3.5.4 Local network gateway 21](#_Toc174116290)

[3.3.5.5 Load balancing (Application Gateway) 21](#_Toc174116291)

[3.3.6 Segurança 23](#_Toc174116292)

[3.3.6.1 WAF 23](#_Toc174116293)

[3.3.7 Container registries 24](#_Toc174116294)

[3.3.8 VPNs 25](#_Toc174116295)

[3.3.8.1 VPN Metatron 25](#_Toc174116296)

[3.3.8.2 VPN Audaz 25](#_Toc174116297)

[3.3.8.3 GW 25](#_Toc174116298)

[3.3.8.4 Connection VPN Site-to-site (IPsec) 27](#_Toc174116299)

[3.3.8.5 Etro-MTT 27](#_Toc174116300)

[3.3.8.6 SiteblitzProduction-AudazProd 27](#_Toc174116301)

[3.3.8.7 SiteblitzProduction-MTT 28](#_Toc174116302)

[3.3.8.8 SophosFW 28](#_Toc174116303)

[3.3.8.9 OpenVPN 28](#_Toc174116304)

[4. Esteiras de integração e implantação contínua 29](#_Toc174116305)

[4.1.1 Pipeline (CI/CD) Siteblitz-Backend 29](#_Toc174116306)

[4.1.2 Estágio de BuildAndPush Integração Contínua (CI) 29](#_Toc174116307)

[4.1.3 Estágio de DeployK8SJob (Entrega Contínua - CD) 30](#_Toc174116308)

[4.1.4 Pipeline (CI/CD) Siteblitz-Frontend 30](#_Toc174116309)

[4.1.5 Estágio BuildAndDeploy 30](#_Toc174116310)

# Informações gerais

|  |  |
| --- | --- |
| AUDAZ TECNOLOGIA | |
| Avenida Jaime Poggi, 99 – BL3 # 104 - Jacarepaguá - Rio de Janeiro - RJ | |
| CEP: 22775-130 | |
| Tel.: (021) 3580-7301 |  |

## Responsáveis

|  |  |
| --- | --- |
| Consultores | Heron Brito  [heron@audaztecnologia.com.br](mailto:heron@audaztecnologia.com.br)  Resller dias  [resller@audaztecnologia.com.br](mailto:resller@audaztecnologia.com.br) |

## Dados do cliente

|  |  |  |
| --- | --- | --- |
| Empresa | ETRO Construction  https://www.etroconstruction.com/  4727 Hastings St, Burnaby, British Columbia V5A, CA | |
| Informações de Contato | Marcel Rohrs  Gerente de TI  [marcelo.rohrs@etroconstruction.com](mailto:marcelo.rohrs@etroconstruction.com)  Rafael Souza  Analista  [rafael.souza@metatron.com.br](mailto:rafael.souza@metatron.com.br) | Luiz Steinberg  Diretor de TI  [luizst@metatron.com.br](mailto:luizst@metatron.com.br) |

## Controle do documento

Localização do Documento

Este documento pode ser encontrado na rede corporativa da AUDAZ, contate o Gerente de Projetos para maiores informações.

Histórico

|  |  |  |  |
| --- | --- | --- | --- |
| Data da revisão | Ver/Rev. | Resumo da mudança | Autor |
| 18/04/2024 | rev01 | Esqueleto inicial | Resller Dias |
| 26/04/2024 | rev02 | Versão inicial | Heron Brito |
|  |  |  |  |
|  |  |  |  |

Revisões

|  |  |  |
| --- | --- | --- |
| Data da revisão | Ver/Rev. | Nome |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Resumo executivo

## Objetivo

escrever

# Visão Geral

A aplicação web Siteblitz é um sistema de engenharia projetado para gerenciar materiais e ferramentas em obras. A infraestrutura na Azure foi cuidadosamente planejada e implementada para garantir escalabilidade, disponibilidade e segurança.

## Arquitetura da Aplicação

Diagrama, Aplicativo, Word

Descrição gerada automaticamente

A aplicação Siteblitz composta por três camadas principais:

1. Camada de FrontEnd: Desenvolvida usando o framework JavaScript, React, esta camada fornece a interface de usuário da aplicação.
2. Camada de BackEnd: Implementada como uma API REST utilizando o framework Dotnet em Docker. Esta camada lida com a lógica de negócios da aplicação.
3. Camada de Dados: Utiliza o SQL Server como banco de dados para armazenar informações sobre materiais e Object Storage para armazenar arquivos.

## Arquitetura da Infraestrutura

### Frontend

ESCREVER.

A Azure FrontDoor é utilizada como serviço de roteamento global e balanceamento de carga para a camada de apresentação. Ele fornece escalabilidade automática, alta disponibilidade e proteção contra ataques DDoS. Blob Storage armazena os arquivos estáticos da aplicação, como imagens, vídeos e scripts.

### Backend

A Azure Load Balancer distribui o tráfego entre as instâncias do serviço da camada lógica para garantir alta disponibilidade e desempenho. Já a, Azure Gateway fornece conectividade segura entre a infraestrutura na Azure e redes locais ou outras nuvens.

O componente Traefik é utilizado como um proxy reverso e controlador de tráfego para aplicativos em contêineres Docker. E tambem, Docker e Kubernetes são utilizados para implantar e orquestrar contêineres na nuvem. Isso proporciona escalabilidade automática, fácil gerenciamento e isolamento de aplicativos.

### Containers

ESCREVER.

### Dados

ESCREVER.

SQL SERVER

S3-Like

### Rede

Azure CDN (Content Delivery Network) é utilizado para distribuir conteúdo estático da aplicação para usuários finais com baixa latência e alta disponibilidade. Além disso, Azure DNS gerencia o sistema de nomes de domínio para roteamento eficiente de solicitações de usuário.

VPN.

OPENVPN.

### Segurança

O JWT (JSON Web Token) é usado para autenticação entre as camadas da aplicação, garantindo que apenas usuários autorizados acessem recursos específicos. O API de Integração com Google Login Authenticator oferece uma camada adicional de autenticação segura, permitindo que usuários façam login usando credenciais do Google.

### Monitoramento e Observabilidade

A Elasticsearch e Kibana são usados para armazenar, pesquisar e visualizar logs da aplicação e infraestrutura, permitindo a análise de dados e identificação de problemas. Já o Prometheus e Grafana são utilizados para monitorar métricas de desempenho da aplicação, como uso de CPU, memória e tráfego de rede.

O Zabbix é implementado para monitoramento proativo da infraestrutura, alertando sobre falhas e anomalias.

## Infraestrutura

**Sitebliz Frontend**

A arquitetura do siteblitz frontend é composta por três componentes principais: Azure Front Door, Azure Blob Storage e Azure DNS Zone. Essa arquitetura oferece uma solução poderosa e altamente escalável para a entrega de conteúdo estático na web, permitindo criar sites estáticos com desempenho global, alta disponibilidade e segurança aprimorada.

**Front Door**

O Azure Front Door é um serviço de rede de entrega de conteúdo (CDN) global e seguro que oferece escalabilidade, alta disponibilidade e desempenho rápido para seus aplicativos da web. Ele funciona como um gateway de entrada que otimiza a entrega de conteúdo, balanceando a carga de tráfego e roteando o usuário para o ponto de extremidade mais próximo e com melhor desempenho.

Interface gráfica do usuário, Texto, Email

Descrição gerada automaticamente

**Azure KeyVault**

ESCVREVER

#### Object Storage

O Blob Storage é usado para armazenar os arquivos do site estático, como HTML, CSS, JavaScript e imagens. Ele fornece uma solução econômica e eficaz para hospedar sites estáticos na nuvem, com escalabilidade automática para lidar com grandes volumes de tráfego.

**Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente**

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

### Sitebliz Backend

A arquitetura do Siteblitz Backend é projetada para oferecer uma solução robusta e escalável para suportar os requisitos de processamento e armazenamento de dados da aplicação. Utilizando uma combinação de recursos do Azure, incluindo Virtual Machines, Load Balancing (Application Gateway), WAF (Web Application Firewall), SQL Databases e Container Registries, o Siteblitz Backend é capaz de fornecer alto desempenho, segurança e confiabilidade para a aplicação.

Recursos do Azure:

#### Virtual machines

As Virtual Machines são utilizadas para hospedar o ambiente de execução do k3s, um sistema de orquestração de contêineres leve e eficiente. O k3s é responsável por gerenciar os contêineres que executam a aplicação do Siteblitz Backend, garantindo escalabilidade e disponibilidade.

**Configuração da Máquina Virtual**: A VM é do tipo Standard B2ms, com 2 vCPUs, 8 GiB de memória, e executa o sistema operacional Linux (Ubuntu 22.04)

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Dados

#### SQL databases

Azure SQL Database é uma solução poderosa e flexível para hospedar de bancos de dados SQL na nuvem, oferecendo desempenho, segurança e escalabilidade líderes do setor

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Kubernetes (k3s)

Kubernetes (k3s) é uma plataforma poderosa e flexível para gerenciar aplicativos em contêineres, oferecendo recursos avançados de orquestração, escalonamento automático,

Aplicações em execução no kubernets:

* **siteblitz-backend-test-:** aplicação siteblitz-backend no ambiente de Test.
* **siteblitz-backend-staging** aplicação siteblitz-backend do ambiente de Staging.
* **cert-manager:** responsável por gerenciar certificados SSL/TLS para o cluster Kubernetes.
* **svclb-traefik:** responsável por balancear o tráfego para o Traefik.
* **local-path-provisioner:** fornecer armazenamento persistente local para o cluster Kubernetes.
* **prometheus-operator:** é responsável por operar o servidor de monitoramento Prometheus.
* **blackbox-exporter:** coleta métricas de disponibilidade para endpoints HTTP, HTTPS, DNS, TCP e ICMP.
* **kube-state-metrics:** coleta métricas do estado do Kubernetes.
* **metrics-server:** fornece métricas de utilização de recursos do cluster Kubernetes.
* **coredns**: é responsável por resolver nomes de DNS no cluster Kubernetes.
* **Grafana:** é responsável pela interface de usuário do Grafana para visualização de métricas e dashboards.
* **traefik**: o Traefik em execução, que atua como um controlador de Ingress para o cluster Kubernetes.
* **prometheus-k8s:**Instâncias do servidor de monitoramento Prometheus no namespace monitoring.
* **Fluentd:** é responsável pela coleta e envio de logs do cluster Kubernetes.

#### Pods

Os Pods são uma abstração poderosa que simplifica a implantação e o gerenciamento de aplicativos em Kubernetes. Eles fornecem um ambiente isolado e compartilhado para a execução de contêineres, permitindo que os desenvolvedores implantem e dimensionem aplicativos de forma eficiente e confiável

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Essa lista representa os diferentes pods em execução em um cluster Kubernetes, distribuídos em vários namespaces. Cada pod corresponde a uma instância de um contêiner que desempenha um papel específico no cluster, como gerenciamento de certificados, coleta de métricas, provisionamento de armazenamento e execução de aplicativos. Os namespaces ajudam a organizar e isolar os recursos do cluster, permitindo que diferentes aplicativos e componentes coexistam e sejam gerenciados de forma mais eficiente

#### Service

Os serviços são uma parte fundamental da arquitetura de rede do Kubernetes, permitindo que os aplicativos se comuniquem uns com os outros de forma transparente e eficiente.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

#### Ingress

O Ingress é uma ferramenta poderosa para gerenciar o acesso externo aos serviços dentro do cluster Kubernetes, permitindo uma configuração flexível e escalável para roteamento de tráfego e balanceamento de carga.

Texto, Aplicativo, Email

Descrição gerada automaticamente

### Rede

#### DNS Zone

A zona de DNS no Azure é um serviço que permite hospedar e gerenciar o DNS (Domain Name System) de um domínio no Azure. Ele fornece controle sobre a resolução de nomes de domínio para seus recursos no Azure, como máquinas virtuais e aplicativos da web

#### Rede Virtual

o Azure Virtual Network é um componente fundamental para criar redes virtuais isoladas e seguras na nuvem do Azure, oferecendo controle, segurança e conectividade para seus aplicativos e recursos na nuvem.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Subnet**

Uma sub-rede (subnet) em uma rede virtual (VNet) do Azure é uma divisão lógica de uma rede virtual em segmentos menores para melhor gerenciamento e controle do tráfego de rede. As sub-redes permitem que você agrupe recursos em uma rede virtual com base em necessidades específicas de segurança, organização ou conectividade.

Texto

Descrição gerada automaticamente com confiança média

#### Virtual network gateway

A Virtual Network Gateway (Gateway de Rede Virtual) é um tipo de recurso do Azure que permite a você criar uma conexão segura entre sua rede virtual no Azure e outra rede, seja ela uma rede local, outra rede virtual no Azure ou até mesmo a internet. Esse gateway atua como uma ponte entre essas redes, permitindo a comunicação segura e o tráfego de dados entre elas.

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

#### Local network gateway

O Local Network Gateway (Gateway de Rede Local) é um recurso do Azure que representa a configuração da sua rede local para estabelecer uma conexão segura com a sua rede virtual no Azure. Ele define os detalhes da rede local, como o endereço IP público da sua rede local, os intervalos de endereços da sua rede local e as configurações de segurança necessárias para a conexão.

#### Load balancing (Application Gateway)

O Azure Load Balancer é um serviço essencial para distribuir o tráfego de rede de entrada para clusters Kubernetes . Ele garante alta disponibilidade e confiabilidade dos aplicativos, atuando como um ponto de entrada único e distribuindo o tráfego de maneira equilibrada entre as instâncias disponíveis. Isso evita a sobrecarga de máquinas ou serviços individuais, melhorando o desempenho, a escalabilidade e a resiliência dos aplicativos. Dessa forma, o Azure Load Balancer proporciona uma experiência consistente e confiável para os usuários finais.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Backend pools**

Um pool de back-end é uma coleção de recursos aos quais o gateway de aplicativo pode enviar o tráfego. Ele conter as máquinas virtuais,

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Backend settings**

Se referem às configurações que especificam como um gateway ou balanceador de carga se comunica com os servidores de backend que hospedam os aplicativos ou serviços que estão sendo expostos

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Backend health**

São verificações regulares feitas pelo gateway ou balanceador de carga para verificar se os servidores de backend estão funcionando corretamente.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Segurança

#### WAF

O WAF (Firewall de Aplicativo Web) do Azure, integrado ao Gateway de Aplicativo do Azure, protege ativamente os aplicativos web contra explorações e vulnerabilidades. Ele utiliza o Conjunto de Regras Principais (CRS) do Open Web Application Security Project (OWASP) para identificar e bloquear ataques conhecidos, como injeção de SQL e scripts entre sites. O WAF é configurado por meio de políticas,

### Container registries

Azure Container Registry (ACR) é um serviço de registro de contêineres privados na nuvem do Azure, projetado para hospedar e gerenciar imagens de contêineres. Com o Azure Container Registry, você pode criar, armazenar, gerenciar e proteger imagens de contêineres do Docker de maneira eficiente e segura.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

### VPNs

#### VPN Metatron

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

#### VPN Audaz

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### GW

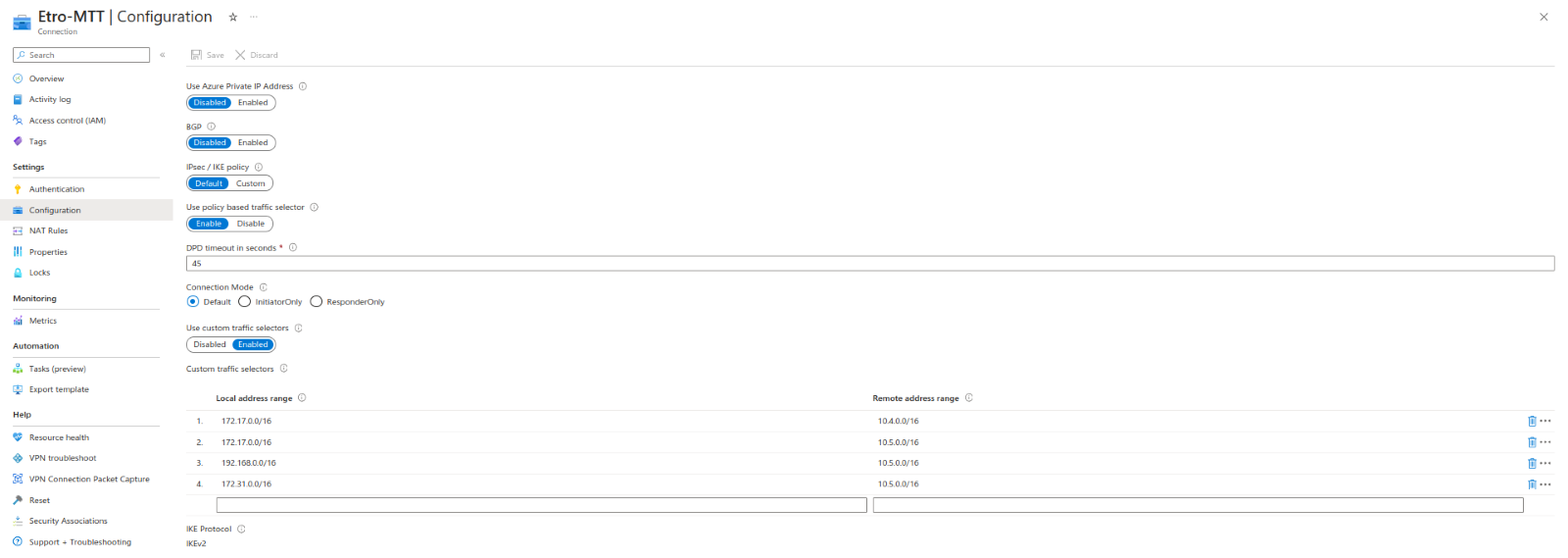
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

#### Connection VPN Site-to-site (IPsec)

A conexão VPN Site-to-Site (IPsec) é uma forma segura e confiável de conectar sua rede local à sua rede virtual no Azure, permitindo a comunicação segura entre os recursos das duas redes por meio da Internet pública.

#### Etro-MTT



#### SiteblitzProduction-AudazProd

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

#### SiteblitzProduction-MTT

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

#### SophosFW

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### OpenVPN

TELAS DO OPENVPN

# Esteiras de integração e implantação contínua

Nulla lacus odio, bibendum sed ante eget, eleifend venenatis metus. Quisque suscipit ex vel urna laoreet, a pharetra ipsum viverra. Sed ac fermentum ex. Nunc ut urna tincidunt nulla lobortis ullamcorper. Praesent nec mollis purus, sit amet tempus nunc. Sed efficitur velit nec dolor rhoncus laoreet. Donec imperdiet scelerisque justo condimentum interdum. Donec mollis urna ac libero tincidunt tincidunt. Donec scelerisque sit amet purus vitae molestie. Maecenas fermentum nec dolor id efficitur.

### Pipeline (CI/CD) Siteblitz-Backend

Este pipeline visa automatizar a construção de uma imagem Docker e a implantação de uma aplicação em um ambiente de Kubernetes.

Configuração Inicial:

* Trigger: O pipeline é acionado quando ocorrem alterações no branch "Test,Staging,Master".
* Pool de Agentes: Utiliza o pool padrão do Azure Pipelines.
* Grupo de Variáveis: Utiliza um grupo de variáveis, para definir variáveis de ambiente compartilhadas.
* Variáveis: Define variáveis de ambiente adicionais, como a configuração de build, diretório temporário para CLI do .NET, nome do branch e versão da image

### Estágio de BuildAndPush Integração Contínua (CI)

BuildAndPushJob: Este job realiza a construção da imagem Docker e a publicação no registro de contêineres.

Passo do BuildAndPushJob:

* Checkout: Baixa o código fonte do repositório.
* Login to Docker Registry: Realiza o login no registro de contêineres e constrói a imagem Docker com base no Dockerfile.
* Build and Push Docker image: Realiza o push da imagem Docker para o registro de contêineres, utilizando uma tag baseada na versão da build. Esta ação utiliza um serviço de conexão previamente configurado no Azure Pipelines para autenticar e conectar-se ao registro de contêineres, garantindo o correto armazenamento da imagem no repositório.

### Estágio de DeployK8SJob (Entrega Contínua - CD)

**DeployK8SJob**: Este job implanta a imagem Docker no cluster Kubernetes.

**Passo do BuildAndPushJob**:

* **Dependência**: Dependente do job BuildAndPushJob para garantir que a imagem esteja disponível.
* **Checkout**: Baixa novamente o código fonte do repositório (pode ser desnecessário e poderia ser removido para otimização).
* **Deploy k8s**: Utiliza variáveis de ambiente para definir parâmetros de implantação e executa um script Bash para implantar a imagem no Kubernetes.
* **Substituição de Variáveis**: O script Bash substitui variáveis no arquivo de configuração do Kubernetes antes de aplicar a implantação.
* **Comandos kubectl**: Usa o kubectl para listar recursos do Kubernetes e aplicar a implantação.

**Observações:**

O pipeline segue o padrão CI/CD, onde a Integração Contínua (CI) é representada pela construção da imagem Docker e a Entrega Contínua (CD) é representada pela implantação no Kubernetes.

A Integração Contínua garante que a imagem Docker seja construída e disponível sempre que houver uma alteração no código fonte, enquanto a Entrega Contínua garante que a nova versão da aplicação seja implantada automaticamente no Kubernetes.

### Pipeline (CI/CD) Siteblitz-Frontend

Este pipeline é responsável por automatizar a construção e implantação de uma aplicação web. Ele utiliza o Azure Pipelines para realizar várias etapas, desde a instalação das dependências até o upload dos arquivos para o Azure Blob Storage e a invalidação do cache no Azure Front Door.

**Configuração Inicial**

* **Trigger:** O pipeline é acionado quando ocorrem alterações no branch "Test,Staging,Master".
* **Agent Pool:** Utiliza a imagem de máquina virtual "windows-latest".
* **Variáveis:** Usa um grupo de variáveis para definir variáveis de ambiente compartilhadas.

### Estágio BuildAndDeploy

Este estágio contém os jobs e as etapas necessárias para construir e implantar a aplicação web.

**Job BuildAndDeployjob:**

* **NodeTool:** Instala o Node.js na versão especificada.
* **Script Install Dependencies:** Instala as dependências do projeto usando o Yarn.
* **Script Build:** Constrói a aplicação web usando o Yarn e define variáveis de ambiente necessárias para a construção.
* **Clean Azure Blob Container:** Limpa o conteúdo do contêiner de Blob no Azure Storage.
* **Upload to Azure Blob:** Faz o upload dos arquivos da aplicação web para o contêiner de Blob no Azure Storage.
* **Invalidate Front Door:** Invalida o cache no Azure Front Door para refletir as alterações feitas na aplicação.