**Implementação de IaC com Template ARM**

**Traffic Manager**

**27/03/2020 / Versão 1.0**

Sumário

[Tabela de versionamento 3](#_Toc36030605)

[Introdução 4](#_Toc36030606)

[Criação visual via Portal 4](#_Toc36030607)

[Utilização do Pipeline do Azure Devops via ARM Templates 4](#_Toc36030608)

[Pré-requisitos (Dependências) 4](#_Toc36030609)

[Variáveis 4](#_Toc36030610)

[Scripts JSONs - REPOS 4](#_Toc36030611)

[Criação PipeLine 4](#_Toc36030612)

[Variáveis 4](#_Toc36030613)

[Tasks 5](#_Toc36030614)

# Tabela de versionamento

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Data | Autor | Descrição |
| 0.1 | 24/03/2020 | Daniel Huanca | Criação do Template. |
| 1.0 | 27/03/2020 | Daniel Huanca | Inseridos prints e procedimento. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introdução

Documentação de implementação do componente Traffic Manager utilizando ARM Templates do Azure no Pipeline do Azure Devops.

Com o intuito de provisionar a infraestrutura necessária para migração das aplicações On-premises para a Cloud Azure de maneira automatizada. ( CI / CD)

# Criação visual via Portal

Seguem telas coletadas na interface gráfica do Portal contendo todos os campos necessários para criação do serviço no Azure.

Componentes básicos para provisionamento do Serviço.

**-Name**

**- Routing Method**

**- Subscription**

**- Resource Group**

**- Region**

# 

Texto alternativo gerado por máquina:
Create Traffic Manager pr... 
Name 
Trafficmanagerdhg01 
.trafficmanager.net 
Routing method 
Priority 
Subscription 
Azure subscription 1 
Resource group * 
Create new 
Resource group location O 
(LIS) East US 2 
x 
Create 
Automation optians 

# Utilização do Pipeline do Azure Devops via ARM Templates

Descrição de pré requisitos, configuração de variáveis necessárias e explicação dos Scripts JSONs.

## Pré-requisitos (Dependências)

Os pré-requisitos da aplicação são os seguintes:

- **Resource Group**

Sem a prévia criação do(s) componente(s) acima não é possível o deploy do serviço.

## Variáveis

Particularidades do serviço para correta implementação via PipeLine

## Scripts JSONs - REPOS

Componentes:

**- Template:**

O Script utilizado para coleta genérica e que foi utilizado para testes.

- TrafficManager\_Template.json

**- Parameters**

JSON de Parâmetros utilizado para testes. O qual deve ser utilizado como base para eventuais customizações ou levantamento de variáveis.

TraficManager\_Template\_Paramenters.json

Devem estar corretamente no REPOS para serem utilizados como artefato do Pipeline:

## Texto alternativo gerado por máquina: TrafficManager_ARM Contents History Committed 7ff6c35a: Deleted 123 TrafficManager_Template_Parameters.json TrafficManager_Template.json Last change 2m ago 2m ago Commits eeb338a8 Added 2 files to /TrafficManager_ARMdaniel.huanca eeb338a8 Added 2 files to /TrafficManager_ARMdaniel.huanca

## Criação PipeLine

Na pipeline da aplicação a ser migrada, criar uma task separada e antes dos pacotes de aplicação para provisionamento da Infraestrutura como código. (IaC)

Cada aplicação possui as suas particularidades que devem ser levantadas no planejamento da migração para definir os componentes a serem provisionados.

### Variáveis

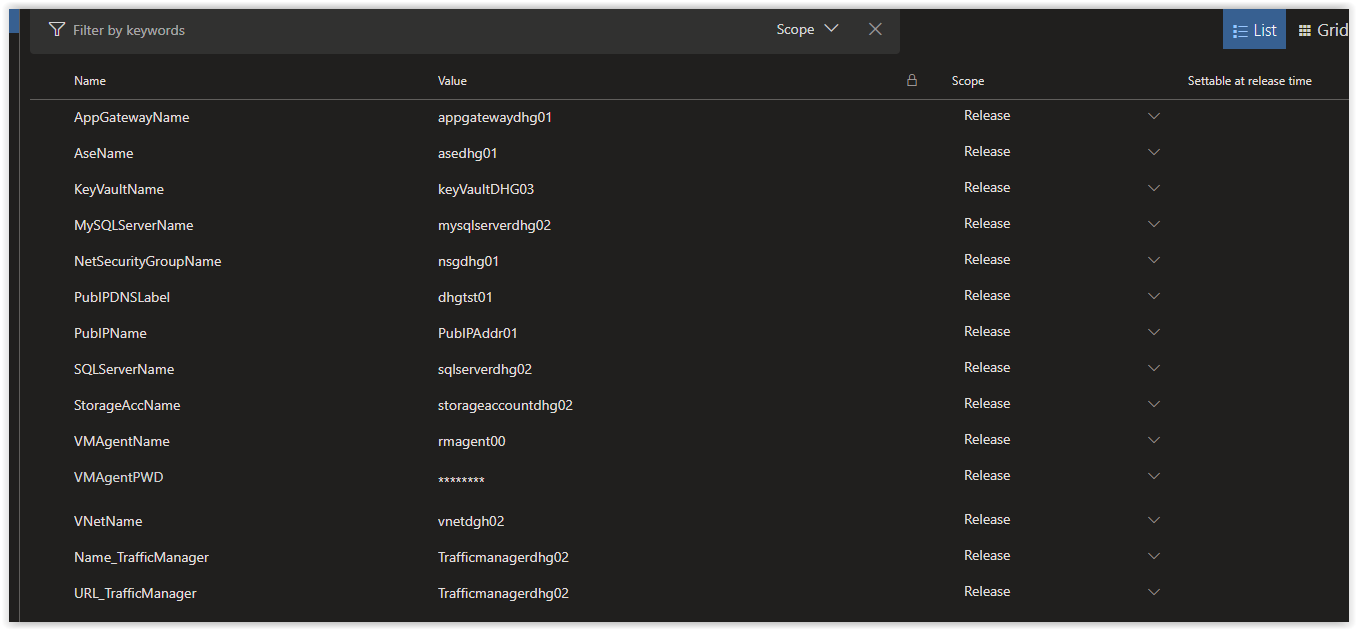
Cadastrar na Pipeline previamente as variáveis que serão utilizadas pelo Deploy da infra, seguindo nomenclatura pré definida.

Variaveis utilizadas pelo Serviço TrafficManager

- **name** (Name\_TrafficManager)

-**relativeName** (URL\_TrafficManager)

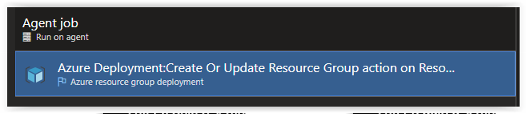
Conforme evidência abaixo as variáveis de cada aplicação deverão ser cadastradas no



### Tasks

Para Deploy da infra para o TrafficManager são necessárias as Tasks abaixo:

- Tipo de Task – Azure resource group deployment



**Campos a serem utilizados:**

**- Subscription:** Subscrição aonde será realizado o deploy da infra.

**- Resource Group:** Resource group aonde será feito o deploy da infra.

**- Location:** Região onde será realizado o deploy da infra.

**- Template:** Local no REPOS onde se encontra o Template.

**- Parameters:** Local no REPOs onde se encontra o arquivo de Parameters.

**- Override template parameters:** Campo utilizado para sobrescrever parâmetros descritos no arquivo de parâmetros, ideal realizar a alteração nesse campo, evitando alterações no template/parameters que serão utilizados em outras aplicações.

Deve ser utilizado no seguinte formato:

**-PARAMETRO VALOR**

Onde **PARAMETRO** está ligado ao existente no arquivo JSON de parâmetros como abaixo:

INSERIR PRINT PARAMETERS do SERVIÇO

E **VALOR** o nome estabelecido por convenção de nomenclatura ou configuração.

**Parâmetros que podem ser alterados do VNET:**

- name

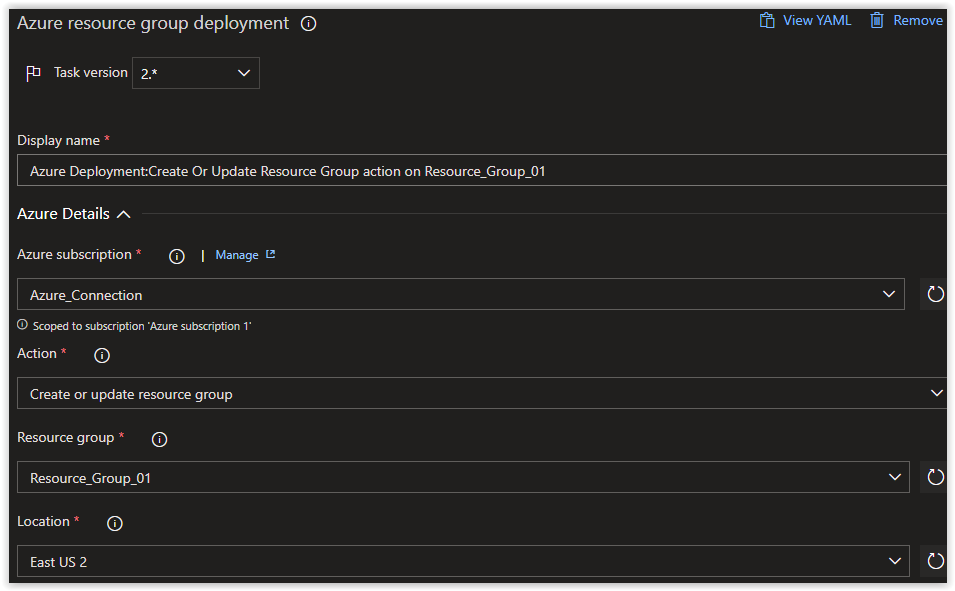
- relativeName

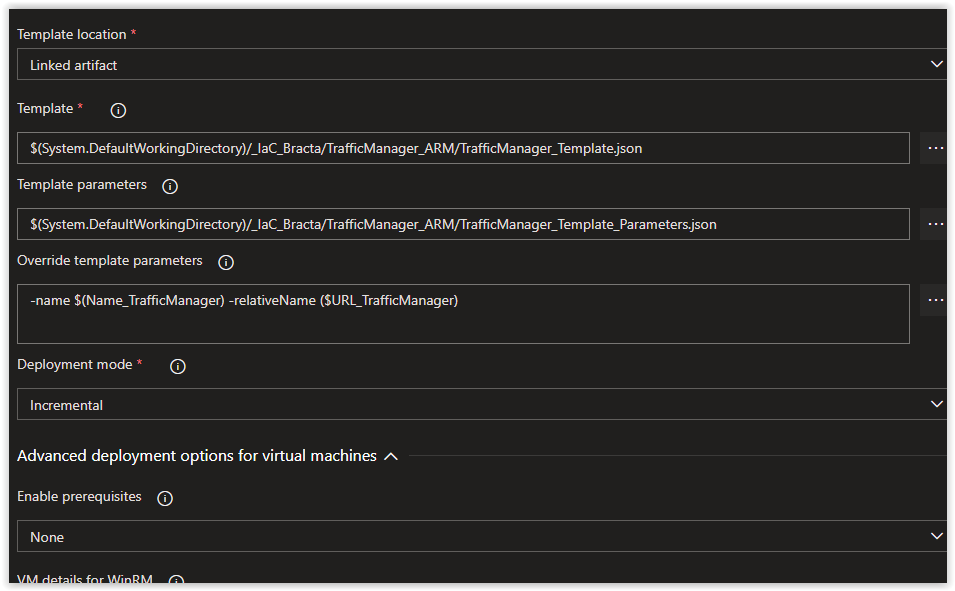
- trafficRoutingMethod

O único obrigatório seria o **name e relativeName** que no exemplo é trazido da variável cadastrada anteriormente estando no formato **$(VARIAVEL\_CADASTRADA)**.

Dessa maneira ao executar o provisionamento é utilizado o nome cadastrado.

**Tela de config da task:**





Após a configuração da task como descrito acima é possível provisionar uma VNET conforme necessidade da aplicação.