

Instalação Dynatrace (APM - Kubernetes)

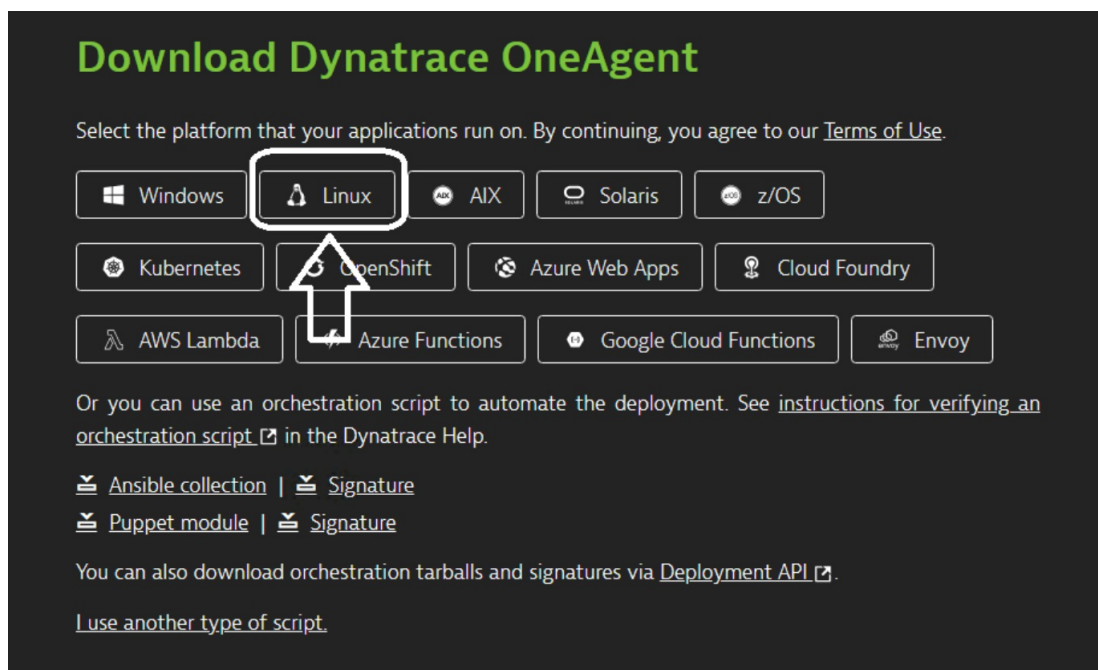
Ambiente Managed (on premise)

1) Valide se está liberado a porta 443 do lado dos nodes Kubernetes, e a porta 9999 no ActiveGate



2) Faça a instalação do OneAgent direto nos nós

Para isso vamos gerar o instalador na console UI



⚠ Aqui será necessário que você gere um token com as devidas permissões habilitadas

Download Dynatrace OneAgent for Linux

Install Dynatrace OneAgent on each Linux machine that you want to monitor.

Please enter the PaaS token you have or generate a new one

PaaS Token Create token

Choose installer type: x86/64

1. Set customized options (optional).

Monitoring mode Full stack [What is monitoring mode?](#)

Set Host Groups, Network Zones and tags.

Network zone productionk8s [What are network zones?](#)

Host group [What are host groups?](#)

Custom host name [What are custom host names?](#)

Tags [What are tags?](#)

Properties [What are properties?](#)

☒ Enable access to application log-file content on this host for problem analysis.
[Log Monitoring](#)

Em "Access Tokens" clique em "Generate access token"

Search Ambiente Producao...

Access tokens **Generate access token**

Generate access token

Give your new token a name and select only those scopes that you need. To generate an access token for PaaS or a Dynatrace module, select a token template. For details, go to [Token permissions documentation](#).

Token name

Expiration date

Template None

Select scopes from the table below

Scope name	Scope type	Permission summary
☒ Read ActiveGates activeGates.read	API v2	Grants access to GET requests of the ActiveGates API .
☒ Write ActiveGates activeGates.write	API v2	Grants access to POST and DELETE requests of the ActiveGates API .
☒ Create ActiveGate tokens activeGateTokenManagement.create	API v2	Allows to create ActiveGate tokens via the API.

Generate token with selected scopes?

Generate token Discard changes

Você vai criar seu token, não coloque data de expiração para esse token específico. Marque todas as permissões necessárias para o OneAgent.

Depois de validar se está tudo certo, clique em "Generate Token".

☒ Write ActiveGate tokens activeGateTokenManagement.write

Generate token with selected scopes?

Copie o token gerado e cole em um local de segurança, vamos precisar dele para a próxima parte.

🔍 Search Ambiente Producao...

Access tokens **Generate access token**

🔧 Give your new token a name and select only those scopes that you need. To generate an access token for PaaS or a Dynatrace module, select a token template. For details, go to [Token permissions documentation](#).

PaaS integration - Installer download PaaS integration - Support alert

Dynatrace modules scopes

Dynatrace module integration - Synthetic Classic

Please copy and store your token in a password manager!
You see the newly generated token only once upon its creation.

"SEU TOKEN GERADO VAI APARECER AQUI" **Copy** Done

Volte a aba "Download Dynatrace OneAgent for Linux", vamos inserir o token que acabamos de criar.

Download Dynatrace OneAgent for Linux

Install Dynatrace OneAgent on each Linux machine that you want to monitor.

Please enter the PaaS token you have or generate a new one

PaaS Token **Create token**

Choose installer type: **x86/64**

1. Set customized options (optional).

Monitoring mode **Full stack** [What is monitoring mode?](#)

Set Host Groups, Network Zones and tags.

Network zone **productionk8s** [What are network zones?](#)

Host group [What are host groups?](#)

Custom host name [What are custom host names?](#)

Tags [What are tags?](#)

Properties [What are properties?](#)

☒ Enable access to application log-file content on this host for problem analysis.
Log Monitoring

Após inserir, scrole com o mouse para baixo, você vai ver três passos que precisam ser feito do lado do Sistema operacional lá nos nodes.

2. Download the installer using this command on the target host.

```
wget -O Dynatrace-OneAgent-Linux-productionk8s-1.289.154.20240531-083726.sh "https://[redacted]/deployment/installer/agent/unix/default/latest?arch=x86&networkZone=productionk8s" --header="Authorization: Api-Token [redacted]"
```

3. Verify signature:

```
wget https://ca.dynatrace.com/dt-root.cert.pem ; ( echo 'Content-Type: multipart/signed; protocol="application/x-pkcs7-signature"; micalg="sha-256"; boundary="--SIGNED-INSTALLER"' ; echo ; echo ; echo '---SIGNED-INSTALLER' ; cat Dynatrace-OneAgent-Linux-productionk8s-1.289.154.20240531-083726.sh ) | openssl cms -verify -CAfile dt-root.cert.pem > /dev/null
```

4. Run the installer with root rights.

```
/bin/sh Dynatrace-OneAgent-Linux-productionk8s-1.289.154.20240531-083726.sh --set-monitoring-mode=fullstack --set-app-log-content-access=true --set-network-zone=productionk8s
```

5. Restart the processes that you want to monitor.

2. Download do instalador:

- Baixa o instalador do Dynatrace OneAgent com wget, utilizando um token de API e parâmetros como arquitetura e zona de rede.

3. Verificação da assinatura:

- Baixa o certificado raiz da Dynatrace e verifica a assinatura do instalador com openssl, garantindo sua autenticidade.

4. Execução do instalador como root:

- Executa o script de instalação com /bin/sh, configurando o modo de monitoramento como fullstack, habilitando acesso a logs de aplicativos e definindo a zona de rede.

5. Reinício de processos:

- Orienta a reiniciar os processos que devem ser monitorados para que o agente comece a coletar dados.

Depois desse processo realizado em todos os nodes você já deve começar a visualizar os hosts na console UI

Hosts

Start typing to select a filter category or filter by name

Quick filters

All filter options available in the filter field

Problem Impact

☒ Any

☐ Impacted

☐ Not impacted

State

☒ Any

☐ Running

6 Hosts

Name	Operating system	Virtualization	CPU usage	Memory usage	Disk latency	Network traffic
EXEMPLO-NODE01	Linux	VMware	48 %	32 % of 19.5 GiB	2.34 ms	14.4 Mbit/s
EXEMPLO-NODE02	Linux	VMware	4.61 %	22 % of 19.5 GiB	2.22 ms	6.34 Mbit/s
EXEMPLO-NODE03	Linux	VMware	4.92 %	24 % of 19.5 GiB	800 µs	7.43 Mbit/s
	Linux	VMware	4.42 %	31 % of 19.5 GiB	-	12.4 Mbit/s
	Linux	VMware	11 %	59 % of 19.5 GiB	2.13 ms	17.3 Mbit/s
	Linux	VMware	5.87 %	35 % of 19.5 GiB	2.02 ms	8.2 Mbit/s

3) Parte Kubernetes

Primeiro vamos fazer a instalação do Dynatrace operator na ultima versão direto do repositório oficial

```
helm install dynatrace-operator oci://public.ecr.aws/dynatrace/dynatrace-operator \
--set csidriver.enabled="true" \
--create-namespace \
--namespace dynatrace \
--atomic
```

Explicação dos parâmetros

- `helm install dynatrace-operator oci://...`
Instala o Dynatrace Operator a partir de um repositório OCI (container registry).
- `--set csidriver.enabled="false"`
Desativa o driver CSI, usado para gerenciamento de volumes, caso não seja necessário.
- `--set csidriver.enabled="true"`
Ativa o driver CSI, usado para gerenciamento de volumes, caso seja necessário.
- `--create-namespace`
Cria automaticamente o namespace `dynatrace` se ele ainda não existir no cluster.
- `--namespace dynatrace`
Define o namespace `dynatrace` como o local onde os recursos serão instalados.
- `--atomic`
Garante que a instalação seja atômica: se algum passo falhar, tudo será revertido automaticamente.

Criar o secret

Depois de instalado, vamos criar o secret

```
kubectrl create secret generic productionk8s \
--from-literal="apiToken=SEU_API_TOKEN" \
--from-literal="paasToken=SEU_PAAS_TOKEN" \
-n dynatrace
```

Substitua `SEU_API_TOKEN` e `SEU_PAAS_TOKEN` pelos valores reais gerados na interface do Dynatrace.

Verificar versões suportadas pelo CRD

```
kubectl get crd dynakubes.dynatrace.com -o=jsonpath='{.spec.versions[*].name}'
```

```
# Run kubectl commands inside here
# e.g. kubectl get all
> kubectl get crd dynakubes.dynatrace.com -o=jsonpath='{.spec.versions[*].name}'
v1beta1 v1beta2 v1beta3 v1beta4 v1beta5
```

Configuração do Dynakube

```
apiVersion: dynatrace.com/v1beta5
kind: DynaKube
metadata:
  name: productionk8s
  namespace: dynatrace
  annotations:
    feature.dynatrace.com/metadata-enrichment: "true"
    feature.dynatrace.com/automatic-kubernetes-api-monitoring: "true"
spec:
  apiUrl: https://SEU-AMBIENTE-DYNATRACE.NET/e/1004594-4d8d-4671d-86f4-3189005dd8da35/api
  skipCertCheck: true
  tokens: productionk8s
  networkZone: productionk8s

  activeGate:
    capabilities:
      - routing
      - kubernetes-monitoring
      - dynatrace-api
    group: productionk8s
  resources:
    requests:
      cpu: 500m
      memory: 512Mi
    limits:
      cpu: 1000m
      memory: 1.5Gi
```

DynaKube Manifest - Ambiente `productionk8s`

Este manifesto define os componentes necessários para integrar o cluster Kubernetes ao Dynatrace usando o operador oficial.

Estrutura do Manifesto

- **apiVersion:** `dynatrace.com/v1beta1`
Versão da API do CRD (Custom Resource Definition) do Dynatrace.
- **kind:** `DynaKube`
Tipo de recurso customizado que representa a configuração da integração Dynatrace no cluster.
- **metadata.name:** `productionk8s`
Nome da instância do DynaKube.
- **metadata.namespace:** `dynatrace`
Namespace onde os recursos Dynatrace serão instalados.

Annotations

- `feature.dynatrace.com/metadata-enrichment: "true"`
Habilita enriquecimento automático de metadados das entidades monitoradas.
- `feature.dynatrace.com/automatic-kubernetes-api-monitoring: "true"`
Ativa a monitoração automática da API do Kubernetes.

Spec

- **apiUrl**
URL do ambiente Dynatrace usado para se conectar à API.
- **skipCertCheck: true**
Ignora verificação de certificado SSL (útil para ambientes internos ou certificados customizados).
- **tokens: productionk8s**
Nome do `Secret` contendo os tokens `apiToken` e `paasToken`.
- **networkZone: productionk8s**
Zona de rede lógica usada para segmentação de comunicação dentro do Dynatrace.

activeGate

- **capabilities**
Lista de funcionalidades ativadas:
 - `routing` : roteamento de tráfego para o Dynatrace
 - `kubernetes-monitoring` : monitoramento da API do Kubernetes
 - `dynatrace-api` : acesso à API Dynatrace
- **resources**
Requisições e limites de CPU e memória para o ActiveGate.

Importante: certifique-se de criar o `Secret` chamado `productionk8s` no namespace `dynatrace` contendo os tokens `apiToken` e `paasToken`.

Aplique a configuração

```
kubectl apply -f dynakube.yaml
```

OU:

```
.\kubectl.exe --kubeconfig=.\kubeconfig-homolog.yaml apply -f C:\Users\ANDESI3\Downloads\kubectl\dynakube.yaml
```

Instrumentação (auto-injection)

Para facilitar essa parte, foi realizado a criação de um script para auxiliar a criação das labels

```
# Caminho do kubectl e do kubeconfig
$kubectl = ".\kubectl.exe"
$kubeconfig = "--kubeconfig=.\kubeconfig-homolog.yaml"

# Lista de namespaces do sistema a ignorar
$excludedNamespaces = @(
    "kube-system", "kube-public", "kube-node-lease",
    "cattle-system", "cattle-monitoring-system",
    "cattle-fleet-system", "cattle-impersonation-system",
    "calico-system", "tigera-operator",
    "longhorn-system", "cert-manager",
    "local-path-storage", "system-upgrade",
    "default"
)

# Obter todos os namespaces
$namespaces = & $kubectl $kubeconfig get ns -o jsonpath="{.items[*].metadata.name}" | Out-String
$namespaces = $namespaces.Trim().Split()

foreach ($ns in $namespaces) {
    if ($excludedNamespaces -notcontains $ns) {
        Write-Host "Aplicando label em namespace: $ns"
        & $kubectl $kubeconfig label ns $ns dynatrace.com/injection=true --overwrite
    } else {
        Write-Host "Ignorando namespace de sistema: $ns"
    }
}
```

Para validar se se o label (`dynatrace.com/injection=true`) foi realmente aplicado aos namespaces corretos, você pode seguir alguns dos passos abaixo:

```
kubectl get ns --show-labels
```

ou:

```
.\kubect1.exe --kubeconfig=.\kubeconfig-homolog.yaml get ns --show-labels | findstr "dynatrace.com/inject=true"
```

```
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-2617c,kubernetes.io/metadata.name=alert
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-t52dg,kubernetes.io/metadata.name=apicur
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-2617c,kubernetes.io/metadata.name=argocd
app.kubernetes.io/instance=rancher-monitoring,app.kubernetes.io/managed-by=Helm,app.kubernetes.io/part-of=rancher-monitorin
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-fg4fn,heritage=Helm,kubernetes.io/metadata

dynatrace.com/inject=true,dynatrace.com/injection=true,kubernetes.io/metadata.name=dynatrace
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-2617c,kubernetes.io/metadata.name=elasti
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-2617c,kubernetes.io/metadata.name=elasti
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-fg4fn,kubernetes.io/metadata.name=ingres
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-t52dg,kubernetes.io/metadata.name=kafka,
dynatrace.com/inject=true,dynatrace.com/injection=true,field.cattle.io/projectId=p-fg4fn,kubernetes.io/metadata.name=local
```

Deep Monitoring

⚠ Um ponto importante não esqueça de habilitar o "Deep Monitoring" na console UI, se faz necessário para que funcione os traces distribuídos

Process group settings

General

Real User Monitoring (internal)

Deep monitoring

OneAgent features

OneAgent Side Masking

Availability monitoring

Connectivity alerts

URL-based sampling

Process group monitoring

Enable or disable monitoring for certain process groups

Monitoring state

Monitor

Default

Do not monitor

Monitor

