



ANSIBLE

# Ansible para SysAdmin

por

***Phillipe Costa***



# 127.0.0.1 | CHANGED | rc=0 >>

## Tipos de autenticação WinRM



## Tipos de autenticação WinRM

Option	Local Accounts	Active Directory Accounts	Credential Delegation	HTTP Encryption
Basic	Yes	No	No	No
Certificate	Yes	No	No	No
Kerberos	No	Yes	Yes	Yes
NTLM	Yes	Yes	No	Yes
CredSSP	Yes	Yes	Yes	Yes

## Autenticação básica

- Método de autenticação mais simples (apenas usuários locais) e menos seguro (Credenciais codificados em Base64).
- **RECOMENDA-SE** a utilização de um canal seguro (HTTPS) para evitar que as credenciais de acessos sejam decodificadas.
- Por padrão vem desabilitadas, porém podem ser habilitadas através do comando **Set-Item -Path WSMan:\localhost\Service\Auth\Basic -Value \$true** no PowerShell.
- Abaixo, as variáveis de inventário que podem ser utilizadas para autenticação básica:

```
ansible_user: LocalUsername  
ansible_password: Password  
ansible_connection: winrm  
ansible_winrm_transport: basic
```



## Autenticação por Certificado

- Método de autenticação **similar** a autenticação por chaves de SSH (SSH Key Pairs), porém com formatos e processo de geração diferentes.
- Não vem habilitados por padrão. Porém, podem ser habilitados executando o comando **Set-Item -Path WSMan:\localhost\Service\Auth\Certificate -Value \$true** no PowerShell.
- Abaixo, as variáveis de inventário que podem ser utilizadas para autenticação básica:

```
ansible_connection: winrm
ansible_winrm_cert_pem: /path/to/certificate/public/key.pem
ansible_winrm_cert_key_pem: /path/to/certificate/private/key.pem
ansible_winrm_transport: certificate
```

## Autenticação por Certificado

- Para gerar o certificado, pode-se utilizar:
  - Active Directory Certificate Service (ADCS)
  - OpenSSL (necessário **conversão de PFX para PEM**, utilizado pelo Ansible:

```
# Set the name of the local user that will have the key mapped to
USERNAME="username"

cat > openssl.conf << EOL
distinguished_name = req_distinguished_name
[req_distinguished_name]
[v3_req_client]
extendedKeyUsage = clientAuth
subjectAltName = otherName:1.3.6.1.4.1.311.20.2.3;UTF8:$USERNAME@localhost
EOL

export OPENSSL_CONF=openssl.conf
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -out cert.pem -outform PEM -keyout cert_key.pem -subj "/CN=$USERNAME" -extensions v3_req_client
rm openssl.conf
```

## Autenticação por Certificado

- Para gerar o certificado, pode-se utilizar:
  - PowerShell, using the New-SelfSignedCertificate cmdlet (**somente para Windows 10, Windows Server 2012 ou superior**)

```
# Set the name of the local user that will have the key mapped
$username = "username"
$output_path = "C:\temp"

# Instead of generating a file, the cert will be added to the personal
# LocalComputer folder in the certificate store
$cert = New-SelfSignedCertificate -Type Custom `
    -Subject "CN=$username" `
    -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2", "2.5.29.17={text}upn=$username@localhost") `
    -KeyUsage DigitalSignature, KeyEncipherment `
    -KeyAlgorithm RSA `
    -KeyLength 2048

# Export the public key
$pem_output = @()
$pem_output += "-----BEGIN CERTIFICATE-----"
$pem_output += [System.Convert]::ToBase64String($cert.RawData) -replace ".{64}", "$&`n"
$pem_output += "-----END CERTIFICATE-----"
[System.IO.File]::WriteAllLines("$output_path\cert.pem", $pem_output)

# Export the private key in a PFX file
[System.IO.File]::WriteAllBytes("$output_path\cert.pfx", $cert.Export("Pfx"))
```

## Autenticação NTLM

- Mecanismo mais antigo de autenticação da Microsoft, disponível para autenticação local e/ou em domínio.
- É habilitado por padrão no serviço do WinRM, não havendo necessidades de configurações prévias.
- É mais seguro do que o método de autenticação básica, mas não suporta novos protocolos de criptografia.
- Abaixo, as variáveis de inventário que podem ser utilizadas para autenticação básica:

```
ansible_user: LocalUsername  
ansible_password: Password  
ansible_connection: winrm  
ansible_winrm_transport: ntlm
```



## Autenticação Kerberos

- Opção mais **RECOMENDADA** para ambientes de domínios, em especial, por ser a mais segura em utilização com WinRM.
- Possui criptografia de mensagens HTTP e delegação de credenciais.
- Necessita da instalação do pacote **pywinrm[kerberos]** e pacotes adicionais para configuração do kerberos na máquina onde esta sendo executado o Ansible (gerar a configuração do **/etc/krb5.conf**).
- Abaixo, as variáveis de inventário que podem ser utilizadas para autenticação básica:

```
ansible_user: username@DOMAIN.LOCAL
ansible_password: Password
ansible_port: 5685
ansible_connection: winrm
ansible_winrm_transport: kerberos
```

## Autenticação Kerberos


- Pontos importantes:
  - Em versões do Ansible 2.3+, os tickets do Kerberos são gerados automaticamente baseando-se nas variáveis de inventário `ansible_user` e `ansible_password`.
  - Em versões antigas do Ansible ou quando a variável `ansible_winrm_kinit_mode` estiver com o valor `manual`, o ticket do Kerberos já deverá existir. Para gerar manualmente o ticket do kerberos basta executar o comando `kinit username@DOMAIN.LOCAL`. Em seguida, execute `klist` para obter a lista de tickets.

## Autenticação CredSSP

- Novo método de autenticação que possibilita a delegação de credenciais.
- Possui criptografia de mensagens HTTP e delegação de credenciais.
- Envia as credenciais de acesso (usuário e senha) criptografados para o servidor através do protocolo credSSP.
- Pode ser utilizado tanto em autenticação local quanto de domínio.
- Abaixo, as variáveis de inventário que podem ser utilizadas para autenticação básica:

```
ansible_user: username  
ansible_password: Password  
ansible_connection: winrm  
ansible_winrm_transport: credssp
```

```
ok: [localhost] => {  
    "output.stdout_info": [  
        'Obrigado e até a próxima aula'  
    ]  
}
```

A decorative network diagram at the bottom of the slide, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue and white, set against a dark blue background.