Proxmox Lab: Game of Active Directory - Installing the Lab

benheater.com/proxmox-lab-goad-installing-the-lab

0xBEN August 26, 2024

In this module, we'll be taking steps to provision the entire Proxmox Game of Active Directory (GOAD) v3 lab environment using the goad.sh helper script

Aug 26, 2024 5 min read

This module is part of a larger project on setting up ***Game of Active Directory (GOAD) v3*** on Proxmox alongside our existing lab infrastructure. Click here to be taken back to the project landing page.

Previous Step

Proxmox Lab: Game of Active Directory - Creating VM Templates

In this module, we'll be taking steps to create some Windows Server 2016 and Windows Server 2019 templates using Packer for use in the Proxmox Game of Active Directory lab

0xBEN0xBEN

Objectives for this Step

- Configure Terraform variables
- Create the lab environment with Terraform via the `goad.sh` script

Install the Lab

ℹ️

Even though in this step, we're not directly calling the `terraform` binary ourselves, when we use the `goad.sh` script to install the Proxmox lab, that's what's going on under the hood

Define Global Variables

```
cd /root/GOAD
```

Bash

```
nano globalsettings.ini
```

Bash

```
[all:vars]
; This is the global inventory file, data here will override all lab or provider
inventory datas
; modify this to add layouts to VMs
; https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-
language-pack-default-values
; French  : 0000040C
; US      : 00000409
; German  : 00000407
; Spanish : 0000040A
; the first in the list will be the default layout (here: FR | US)
keyboard_layouts=["00000409"]
```

Ini

Set the desired `keyboard_layouts` variable. I've set mine to `US` exclusively.

Define Proxmox Provider Variables

`nano /root/.goad/goad.ini`

Bash

```
[proxmox]
pm_api_url = https://192.168.1.1:8006/api2/json
pm_user = infra_as_code@pve
pm_node = GOAD
pm_pool = GOAD
pm_full_clone = false
pm_storage = local
pm_vlan = 10
pm_network_bridge = vmbr3
pm_network_model = e1000

[proxmox_templates_id]
winserver2019_x64 = 102
winserver2016_x64 = 103
winserver2019_x64_utd = 104
windows10_22h2_x64 = 105
```

Ini

Original File Contents

💡

If you reference the GOAD lab diagram, you'll see how the labs are in color-coded outlines indicating which hosts comprise which lab. In this guide, we're provisioning the GOAD lab (which is the default)

So even though the variables contain references to Windows 10, this lab does not deploy that host, and it would fail anyway, because we didn't download a Windows 10 ISO, nor template it with Packer.

Windows Server 2019 is at template ID `100` and Windows Server 2016 is at template ID `102`

```
[proxmox]
pm_api_url = https://172.16.1.14:8006/api2/json
pm_user = root@pam
pm_node = proxmox-um690
pm_pool = GOAD
pm_full_clone = false
pm_storage = local-lvm
pm_vlan = 10
pm_network_bridge = vmbr1
pm_network_model = e1000

[proxmox_templates_id]
winserver2019_x64 = 100
winserver2016_x64 = 102
winserver2019_x64_utd = 104
windows10_22h2_x64 = 105
```

Ini
Variables for My Environment

❗ **Changes made to the variables:**

- `pm_api_url` — changed the IP address of the PVE node to match mine
- `pm_user = root@pam`
  In the original guide written by m4yfly, we end up giving full permissions to `infrastructure_as_code@pve` anyway
- `pm_node = proxmox-um690` — this is the hostname of the target PVE node
- `pm_pool = GOAD` — this is the resource pool we created above
- `pm_storage = local-lvm` — `local` is incorrect and is not used to store guest disks
- `pm_network_bridge = vmbr1` — this is the target virtual switch for me
- `winserver2019_x64 = 100` — matches the template ID on my Proxmox VE node
- `winserver2016_x64 = 102` — matches the template ID on my Proxmox VE node

As mentioned before, we don't need to worry about the other template IDs, because the `GOAD` lab environment doesn't use either of these VM types.

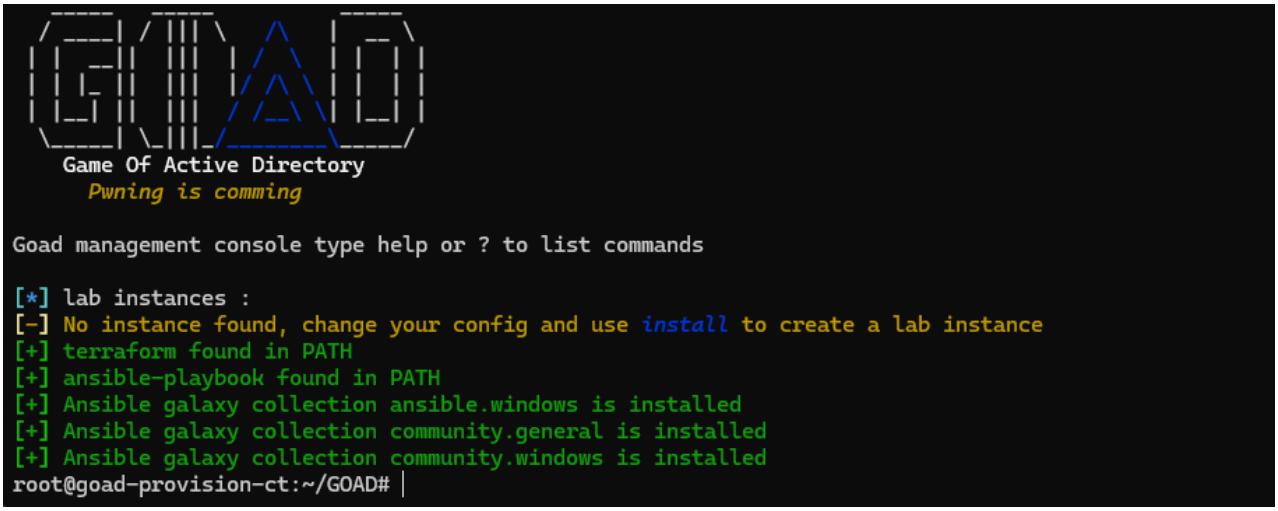
Test Configurations

```
cd /root/GOAD
```

Bash

```
./goad.sh -t check -l GOAD -p proxmox -ip 192.168.10
```

Bash



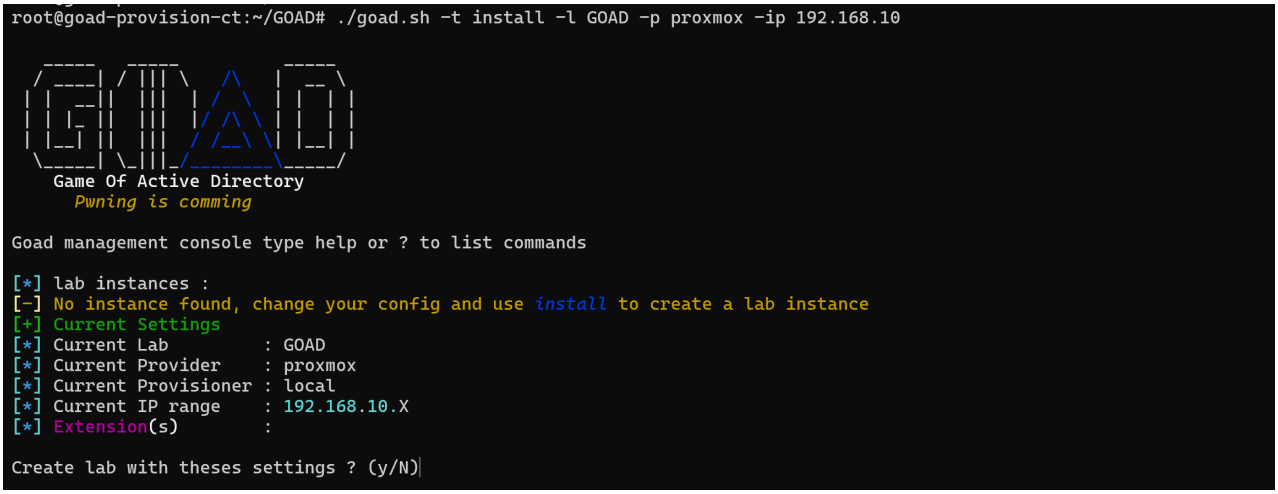
Provision GOADv3

```
cd /root/GOAD
```

Bash

```
./goad.sh -t install -l GOAD -p proxmox -ip 192.168.10
```

Bash



```
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[*] CWD: /workspace/3a66dc-goad-proxmox/provider
[*] Running command : terraform plan
var.pm_password
  Enter a value:
```

Enter the password for `root@pam` on your Proxmox VE node

```
Plan: 5 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actio
[*] CWD: /workspace/3a66dc-goad-proxmox/provider
[*] Running command : terraform apply
var.pm_password
  Enter a value:
```

Once again, enter the password for `root@pam` on your Proxmox VE node

```
Plan: 5 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

Enter `yes` when ready

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

proxmox_virtual_environment_vm.bgp["dc01"]: Creating...
proxmox_virtual_environment_vm.bgp["srv02"]: Creating...
proxmox_virtual_environment_vm.bgp["srv03"]: Creating...
proxmox_virtual_environment_vm.bgp["dc03"]: Creating...
proxmox_virtual_environment_vm.bgp["dc02"]: Creating...
proxmox_virtual_environment_vm.bgp["dc01"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["srv02"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["dc03"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["srv03"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["dc02"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["dc01"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["srv02"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["srv03"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["dc03"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["dc02"]: Still creating... [20s elapsed]
```

Be patient while the hosts are provisioned by Terraform

```
[*] Loading playbook list
[+] build.yml file found
[+] ad-servers.yml file found
[+] ad-parent_domain.yml file found
[+] ad-child_domain.yml file found
[+] wait5m.yml file found
[+] ad-members.yml file found
[+] ad-trusts.yml file found
[+] ad-data.yml file found
[+] ad-gmsa.yml file found
[+] laps.yml file found
[+] ad-relations.yml file found
[+] adcs.yml file found
[+] ad-acl.yml file found
[+] servers.yml file found
[+] security.yml file found
[+] vulnerabilities.yml file found
[*] Run playbook : build.yml with inventory file(s) : /root/GOAD/ad/GOAD/data/inventory, /root/GOAD/workspace/4c8c8c-goad-proxmox/inventory,
/root/GOAD/globalsettings.ini
[*] CWD: /ansible/
[*] Running command : ansible-playbook -i /root/GOAD/ad/GOAD/data/inventory -i /root/GOAD/workspace/4c8c8c-goad-proxmox/inventory -i
/root/GOAD/globalsettings.ini build.yml

PLAY [Read data files] *********************************************************************************************************************
 [started TASK: Gathering Facts on dc01]
 [started TASK: Gathering Facts on dc02]
 [started TASK: Gathering Facts on dc03]
 [started TASK: Gathering Facts on srv02]
 [started TASK: Gathering Facts on srv03]
```

Once provisioned, the build script automatically triggers the Ansible playbook to configure the environment

Troubleshooting

Ansible Hosts Unreachable

```
)", "unreachable": true}
fatal: [srv03]: UNREACHABLE! => {"changed": false, "msg": "ssl: HTTPSConnectionPool(host='192.168.10.23', port=5986): Max retries
used by NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x78da9980d300>: Failed to establish a new connection: [
)", "unreachable": true}

PLAY RECAP *********************************************************************************************************************************
dc01                       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
dc02                       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
dc03                       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
srv02                      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
srv03                      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

[-] 3 fails abort.
[-] Something wrong during the provisioning task : build.yml
root@goad-provision-ct:~/GOAD#
```

I know the correct IP address range was applied to the GOAD hosts — `192.168.10.x`. And, I know pfSense is allowing `TCP/5985` and `TCP/5986` from the provisioning container to the GOAD subnet.

However, when I looked at the hosts in Proxomx, ***the Windows VMs were not fully booted***. Ansible was triggered by the Terraform apply being completed, but the ***hosts were not ready yet to be managed over WinRM***.

```
cd /root/GOAD
```

Bash

```
./goad.sh -h
```

Bash
Help output for the script

```
./goad.sh -t show
```

Bash

```
root@goad-provision-ct:~/GOAD# ./goad.sh -t show

   _____   _____        _____
  / ____| / ||| \   /\   |  __ \
 | |  __||  |||  |  /  \  | |  | |
 | | |_ ||  |||  | /  /\ \ | |  | |
 | |__| ||  ||| / /  __\ \| |__| |
  \_____| \_|||_/_/      \_\_____/
    Game Of Active Directory
      Pwning is comming

Goad management console type help or ? to list commands

[*] lab instances :
```

Instance ID	Lab	Provider	IP Range	Status	Is Default	Extensions
eacbdb-goad-proxmox	GOAD	proxmox	192.168.10.0/24	ready for provisioning	No	

```
root@goad-provision-ct:~/GOAD#
```

Make a note of the instance ID

```
./goad.sh -t install -i eacbdb-goad-proxmox -a eacbdb-goad-proxmox
```

Bash

`-i` specifies the target GOAD lab instance and `-a` specifies to only run Ansible with the `-t install` task. You can find these script options and more using `./goad.sh -h`

```
TASK [Gathering Facts] *********************************************************************************
ok: [srv03]
ok: [dc03]
ok: [dc02]
ok: [srv02]
ok: [dc01]
 [started TASK: save the Json data to a Variable as a Fact on dc01]

TASK [save the Json data to a Variable as a Fact] ******************************************************
ok: [dc01]
 [started TASK: find domain_adapter on dc01]

TASK [find domain_adapter] *****************************************************************************
ok: [dc01] => (item={'dns_domain': None, 'connection_name': 'Ethernet 2', 'default_gateway': '192.168.10.1',
MT Network Connection', 'ipv6': {'address': 'fe80::704a:e212:b63a:1291%2', 'prefix': '64'}, 'macaddress': 'BC
: 2, 'ipv4': {'address': '192.168.10.10', 'prefix': '24'}})
 [started TASK: find nat_adapter on dc01]
 [started TASK: find number of interfaces on dc01]
```

Now, we're gathering Ansible host facts just fine

Current State of the Lab

Type ↑	Description	Disk usage...	Memory us...	CPU usage	Uptime	Host CPU ...	Host Mem...	Tags
lxc	6040 (goad-provision-ct)	12.9 %	11.7 %	3.1% of 4 ...	6 days 21:26...	0.8% of 16 ...	0.4 %	
qemu	104 (SRV02)	0.0 %	71.2 %	30.3% of 2 ...	00:11:40	3.8% of 16 ...	7.0 %	
qemu	106 (SRV03)	0.0 %	69.2 %	80.8% of 2 ...	00:11:38	10.1% of 1...	5.6 %	
qemu	107 (DC01)	0.0 %	54.0 %	99.8% of 2 ...	00:11:39	12.5% of 1...	2.6 %	
qemu	108 (DC02)	0.0 %	52.2 %	99.4% of 2 ...	00:11:36	12.4% of 1...	2.5 %	
qemu	109 (DC03)	0.0 %	52.2 %	82.4% of 2 ...	00:11:37	10.3% of 1...	2.5 %	
qemu	100 (WinServer2019x64-clo...				-			
qemu	102 (WinServer2016x64-clo...				-			

Next Step

Proxmox Lab: Game of Active Directory - Attacking GOAD

In the final module of the lab, we'll be taking steps to ensure that we can access our attack box in certain conditions and successfully ensure connectivity to Game of Active Directory targets in the lab.

 0xBEN0xBEN