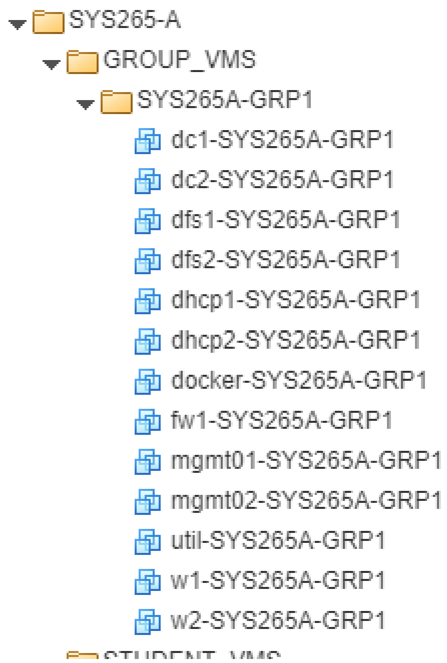
Systems Administration 2: Final Group Project

Your team is going to build a medium-sized enterprise from scratch for a client over a period of some weeks. Your team will demonstrate your integration efforts to a client manager as part of user acceptance testing in a series of milestones. (In real life, you would get paid if you pass the user acceptance testing, and admonished if you don't.)

Prerequisites:

- Your File Services Project is complete.
  - These resources will be turned off and likely removed, and the teams likely reorganized.
- Individual Lab environments for GPO-AD will be shut down shortly after its due date.

Environment



```
SYS265-A
  GROUP_VMS
    SYS265A-GRP1
      dc1-SYS265A-GRP1
      dc2-SYS265A-GRP1
      dfs1-SYS265A-GRP1
      dfs2-SYS265A-GRP1
      dhcp1-SYS265A-GRP1
      dhcp2-SYS265A-GRP1
      docker-SYS265A-GRP1
      fw1-SYS265A-GRP1
      mgmt01-SYS265A-GRP1
      mgmt02-SYS265A-GRP1
      util-SYS265A-GRP1
      w1-SYS265A-GRP1
      w2-SYS265A-GRP1
  STUDENT_VMS
```

Milestones

Milestone 1

- Routing works (systems can talk to one another and route to the internet)
- Domain Users are created

- Domain Joined ws1, ws2
- Redundant ADDS (this means that either can be down and that domain services are still provided)
- Documentation Updated

Milestone 2

- Redundant Linux DHCP works
- Ansible Controller and nodes are configured
- Util is domain joined
- MGMT02 configured
- Documentation Updated

Final Project

- Docker service is deployed
- DFS based profiles
- Ansible Services
- Ansible Users
- Group Policy completed
- Final Documentation Complete

System	Description	Notes
FW1	PFSense Firewall between SYS265-WAN and GROUP-LAN	● WAN on 10.0.17.X ● LAN on 172.16.1.2 ● WAN Assignments
docker	Ubuntu 20.04 Cloud Server	● 172.16.1.5 ● Docker ● Dockerized CMS
DHCP1, DHCP2	Centos 7	● 172.16.1.10,11 ● Redundant DHCP ● Controlled by Ansible via SSH/RSA Keys
DC1,DC2	Server 2019 Core	● Redundant ADDS and DNS Services ● 172.16.1.12,13
MGMT1	Server 2019 GUI	● 172.16.1.14
W1,W2	Windows 10	● DHCP Client/Domain Joined .100-.150
MGMT2	Ubuntu 20.04 Cloud Server	● Ansible/Network Management ● Pick a free IP outside of the DHCP scope

DFS1,DFS2	Server 2019 Core	● Distributed File System that Stores Domain User Profiles & Desktop
util	CentOS 7	● Domain Joined ● 172.16.1.15

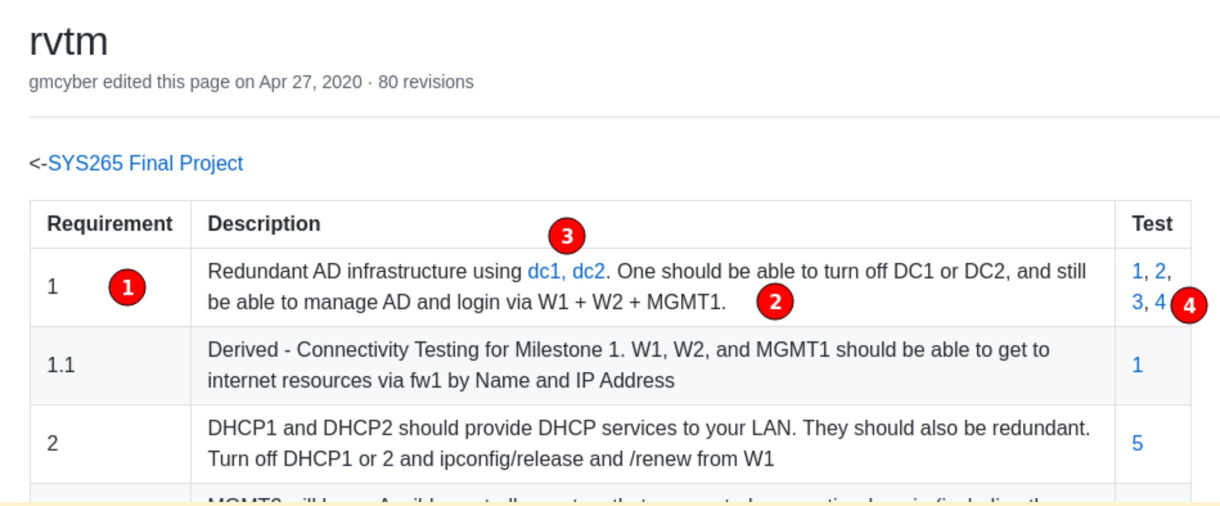
Requirements

For each task, create and document a test that demonstrates to the client that the requirement has been met. Before you demonstrate, the first three columns should be completed. This needs to be part of your Final Project Journal's wiki, linking the videos validating the listed required tests.

💡This is often documented with a Requirements Verification Traceability Matrix (**RVTM**). A simplified version is shown below. This implies every requirement is testable. Your job will be to develop an effective and efficient test that quickly shows you have met each and every requirement. It is possible that a single command and screenshot may deal with more than 1 requirement. The excerpt below shows:

1. The requirement number
2. The description
3. Links to the appropriate system documentation
4. Most importantly, links to the individual tests.

- See the following video for a quick walk through.



rvtm

gmcyber edited this page on Apr 27, 2020 · 80 revisions

<-SYS265 Final Project

Requirement	Description	Test
1	Redundant AD infrastructure using dc1, dc2. One should be able to turn off DC1 or DC2, and still be able to manage AD and login via W1 + W2 + MGMT1.	1, 2, 3, 4
1.1	Derived - Connectivity Testing for Milestone 1. W1, W2, and MGMT1 should be able to get to internet resources via fw1 by Name and IP Address	1
2	DHCP1 and DHCP2 should provide DHCP services to your LAN. They should also be redundant. Turn off DHCP1 or 2 and ipconfig/release and /renew from W1	5

1. Redundant AD infrastructure using DC1 + DC2. One should be able to turn off DC1 or DC2, and still be able to manage AD and login via W1 + W2 + MGMT1.
2. DHCP1 and DHCP2 should provide DHCP services to your LAN. They should also be redundant. Turn off DHCP1 or 2 and ipconfig/release and /renew from W1.

3. MGMT2 will be an Ansible controller system that can control your entire domain, with the exception of Windows workstations and the firewall. You should be able to run interactive commands against all these systems.
4. Util will be a system that you can provision with a new application via MGMT2 and Ansible. It can also be used to assist in meeting other requirements. Deploy an application that has not been covered in class to util, using an Ansible playbook.
5. MGMT1, DC1, DC2, DFS1, DFS2 and your workstations represent your Active Directory Infrastructure. Your domain should be your groupname.local. Join all Windows systems to the domain and at least one of your Linux systems.
6. Create an AD security group called linux-admins. Members of this group should be able to sudo to root on one of your Linux systems (this has some implied sub-requirements).
7. Install docker and a wiki/application of your choice on docker. (NOT WORDPRESS!)
8. Create a Domain Group Policy that allows W1 + W2 to remote desktop between one another.
9. Create a Domain Group Policy that applies corporate wallpaper to W1 + W2 + MGMT1.
10. Create a Domain group policy that moves W1 and W2 user profiles and home directories to a DFS share.
11. Use Ansible to install an apt-package.
12. Use Ansible to install a yum package.
13. Use Ansible to add a new Linux local user can be an SSH user or one with a password.
14. Use Ansible to add a new Windows domain user.

Deliverable 1. Inventory with linked Comprehensive Build documentation (see video). This documentation should be useful to you in the remainder of your time in the program:

- To the extent possible, use Powershell and Bash to configure Windows Systems as opposed to mouse clicks.
- Document this in your wiki!
- Each system in your environment should have its dedicated page where its configuration is clearly stated, and any configuration file that you touch is also stored in the respective git repo and linked from the wiki.
- One team member should create the repo for this final project (GROUPX-Final Project), and then share that wiki with their teammate.
- The documentation should evolve through each milestone as changes are made.

Deliverable 2. Your Wiki should also contain your Test Plan in RVTM format. Ensure that the person responsible for deploying the service is documenting both the build procedures as well as the test procedures. It is an excellent idea for other team members to run the test to ensure the validity of the test procedure.

Deliverable 3. A professional demonstration recording with the entire team (with voice) using the screencasting software of your choice validating the requirements from the RVTM above. You will need to experiment with this to determine the best way to present and capture as a remote team. Show the successful testing of each requirement. Negative points will be assessed for a sloppy presentation.

💡If you do this right, this could be done with three videos:

- 1 that shows your milestone 1 deliverables
- another for milestone 2
- and the final project that shows the deliverables that were not handled before.

Be sure to include (in order) all shared video links. Teamwork can be demonstrated by switching who takes charge of milestone1 and 2 deliverables, and collaborating on the final video.