GROUP ASSIGNMENT 3 — TOPICS 4 AND 5

BUC3129 Systems and Network Administration · File Systems and Storage Management · Process, Service and System Initialization Management

Practical work in Lab 1 · the same groups of five · the same Windows Server domain you built in Assignment 2

ISSUED: THURSDAY 1ST OCTOBER 2026

TO BE SUBMITTED ON MUBSEP BY THURSDAY 22ND OCTOBER 2026, 12PM

Where this picks up

In Assignment 2 you built a school domain on Windows Server: a server, a second machine joined to it, accounts for every member of your group, an OU structure, and a Group Policy Object enforcing a security setting.

That domain is now a live client system. Everything in this assignment is built on top of it. The users are the accounts you already created. The groups are the ones you already have, or new ones you add for a stated reason.

If your Assignment 2 domain is not working, repairing it is your first week's work. Nothing below can be done without it.

There are eleven tasks. As before, each one tells you **what must exist** when you are finished, not which buttons to press. Every member will be questioned individually.

Use the **same computer number** you recorded in Assignment 2, and the same virtual machines.

How this works in the field

In this assignment your group is no longer a group of students. **It is a small IT support company that has been contracted by the school.** Give your company a name and use it on every document you produce.

Real IT support work does not happen by logging in and changing things. It follows a routine, and this assignment follows the same one:

In the field, you…	In this assignment, you…
Receive a work order from the client	Read the Head Teacher's request below
Raise a **change request** before touching a live system, with a rollback plan	Complete a change request for Tasks 1, 3, 5 and 7 before you do them
Take a backup or snapshot before every change	Snapshot the VM before every change; that snapshot is your rollback plan
Log every fault as a **ticket** and work it to closure	Record Task 10 as a ticket, from report to resolution
Send the client a short **status update** every week	Email a weekly status update (see the timeline)
Test the work against the client's requirements before handing over	Write an **acceptance test checklist** and run it
Hand over documentation so the next administrator can run the system	Submit a **handover report** with a runbook for routine jobs
Keep passwords out of documents	Submit credentials in a separate sheet, never in the report
Get the client to **sign off**	Demonstrate the system to me, acting as the client, at the oral defence

Blank templates for the change request, the ticket, the status update and the acceptance test are in the appendices at the end of this brief.

The work order

The Head Teacher has sent a second note to your company. Treat it as a real work order.

> *"Thank you for setting up the accounts. Now three more things. First, the bursar keeps the fee records on a memory stick. I want them on the server, where only she and I can see them, and where teachers cannot. Second, teachers want one shared place for lesson materials. Third, last term a student filled the server with videos and nobody could save anything for a whole day. That must never happen again. Also, the server has one hard disk. If it dies, we lose everything. And I keep being told the backup 'did not run last night'. I want it to run by itself every night, and I want to be able to see that it ran."*

Four storage requirements and two service requirements are hidden in that note. Find them. They are Tasks 1 to 5 and Tasks 6 to 9. Section 2 of your report must state each one in technical language.

Lab 1 rules — still in force

- **Your machines.** The same lab computer and the same virtual machines from Assignment 2. Record the computer number in your report again.
- **Installation media.** I will provide any media you need. Do not download it in the lab.
- **Stay out of each other's way.** Your virtual machines stay on their own internal network, and your domain keeps the name I approved. If you need a new name for any reason, mail me first at **hnamuwaya@mubs.ac.ug**.
- **Snapshots.** Take one before every significant change, and always before you add disks or change permissions. A group that loses a week's work to one mistake will get sympathy, not an extension.
- **Save your work.** Lab machines get reset. Export your virtual machines at the end of every session. Losing them is not an accepted reason for late submission.
- **Nothing runs on a machine that matters.** All of this happens inside virtual machines.

PART A — STORAGE AND FILE SHARING (TOPIC 4)

Task 1 — Storage that survives a dead disk

What must exist: Two new virtual disks attached to your server, combined into a single storage space that keeps working when one of the two disks fails, with a drive letter on it, ready for use.

Windows Server has a built-in feature for this. Find it, and choose the layout that meets the requirement. Be ready to explain what it costs you in usable space, and why that cost is worth paying for fee records.

Change request required before you start.

Evidence: The storage pool and its virtual disk, showing the layout you chose and the usable capacity. Then, with a file saved on the new drive, **remove one of the two disks from the virtual machine** and show that the file is still readable. Screenshot that. Re-attach the disk afterwards and show the storage returning to a healthy state.

> *If the drive is still readable with a disk missing, you have met the Head Teacher's request. If it is not, you chose the wrong layout.*

Task 2 — A folder structure that reflects the school

What must exist: On the new drive, a folder tree for the school. At minimum: somewhere for finance records, somewhere for teaching materials, and somewhere for student work. Sensibly named and deliberately organised.

Evidence: A screenshot of the tree, and one paragraph on why it is arranged that way.

> *Think ahead. The permissions in Task 3 will be far easier if the folders are grouped by **who needs them**, not by subject.*

Task 3 — Permissions: who may see what

What must exist:

- The finance folder readable and writable by the bursar and the Head Teacher, and **nobody else**, not even a teacher browsing casually.
- The teaching materials folder readable by all staff, where **each teacher can change only what they created**. Another teacher may read it but not alter it.
- Permissions assigned to **groups**, never to individual accounts. If you need a new group, create it in the correct OU from Assignment 2.

Then prove it, using the accounts from Assignment 2. A teacher must be shown **failing** to open the finance folder. A second teacher must be shown **failing** to alter the first teacher's file.

Change request required before you start.

Evidence: The permission settings on both folders, and the two refusals, with the account name visible in each. A claim that it works is not proof. **The refusal is the proof.**

> *Windows has a tool that shows you exactly what one user can do to one folder without you having to sign in as them. Find it. It will save you an hour.*

Task 4 — Share it on the network

What must exist: The teaching materials folder and the finance folder both shared, so that a member of staff on the second machine can reach them as a mapped drive using their domain account.

A share on Windows has **two** sets of permissions: the share's own, and the folder's. Understand how they combine before you set either. Get this wrong and either everyone gets in or nobody does.

Evidence: The second machine with the share mapped as a drive letter, signed in as a domain user, with the account name visible. Then one paragraph: which of the two permission sets is doing the real work in your design, and why.

Optional, for extra credit: connect to the same share from a Linux machine. One screenshot. This is what "cross-platform file share" means in your course outline.

Task 5 — Nobody fills the disk again

What must exist: A limit on how much space each student can use in the student work folder. Choose a limit and defend it. A student who reaches it must be stopped, and must see a clear message.

Windows Server offers more than one way to do this. Research them, choose one, and say in the report why you chose it over the other.

Change request required before you start.

Evidence: The limit as configured, and a screenshot of a student account being **stopped** when it tries to exceed it. **The error message is the evidence.**

PART B — SERVICES, SCHEDULING AND PROCESSES (TOPIC 5)

Task 6 — Know what is already running

What must exist: An inventory of the services running on your server, and an understanding of which ones the school depends on.

In the field, the first thing a new administrator does on an unfamiliar server is find out what is running and what matters. Choose **three** services running on your domain controller. For each one, state: what it does, whether it starts automatically, which account it runs as, and what would stop working if it were disabled.

Evidence: A screenshot of the service list with your three marked, and the four answers for each, in a table.

> *At least one of your three must be a service that the domain itself depends on. If you pick three you have never heard of, say so, and find out what they are.*

Task 7 — The backup that runs by itself

What must exist: A scheduled job that copies the finance folder to a backup location, **automatically, every night**, without anybody signed in.

The Head Teacher asked for two things, not one: the backup must run by itself, **and** she must be able to see that it ran. So the job must also leave a record behind every time it runs, with the date and time.

The job must run whether or not a user is signed in. That is a setting, and finding it is part of the task.

Change request required before you start.

Evidence: The job as configured, showing its schedule and the account it runs as. Then **proof that it actually ran on its own**: the job's run history, and the backup files or log it produced, with timestamps visible.

> *Do not test a nightly job by waiting for the night. Set it to run every few minutes first, prove it works, then change it to the real schedule. Say in your report that you did this. Real administrators do exactly the same.*

In your report, answer this honestly: a copy of the finance folder kept on the same server is better than nothing, but it is not a real backup. Why not, and what would a school actually need?

Task 8 — A service that watches the server

What must exist: A job of your own creation that checks the health of the server and records it every few minutes: for example, the free space on the storage you built, or whether last night's backup exists.

It must start by itself after a restart, without anybody touching it.

Evidence: Restart the server. Then show the record still being written afterwards, with timestamps from before **and** after the restart clearly visible.

Task 9 — A runaway process, diagnosed

What must exist: A demonstration that you can find and deal with a process that is misbehaving.

Have one member start a process that uses the processor heavily, signed in as an ordinary domain user. Then, **as the administrator**, find it without being told what it is, identify who started it and how much it is using, and stop it safely.

Evidence: The tool you used, showing the process at the top of the list with its owner; the steps you took, in order; and the system afterwards, back to normal.

In your report, answer this: what is the difference between ending a process politely and forcing it to stop, and when would forcing it be the wrong choice?

PART C — FAULTS AND HANDOVER

Task 10 — A fault, ticketed and resolved

One member of your group deliberately changes **one** thing so that something built in Tasks 1 to 9 stops working. That member tells nobody else what they changed. They then **report it as a user would**, in one sentence: for example, *"I cannot open the teaching materials drive"*.

The rest of the group handles it as a real support ticket: log it, give it a priority, investigate, fix, test, and close it.

There are many possible single changes: a permission, a share, a quota, a stopped service, a disabled scheduled job. Which one it was is not the point.

Evidence: The completed ticket (template in Appendix B): the user's report, the priority you gave it and why, the checks you made **in the order you made them**, the root cause, the fix, and the time it took from opening to closing.

The order of the checks matters more than the answer. A group that checked systematically and took longer scores above a group that guessed.

Task 11 — Export and hand over the machine

What must exist: Your working server, exported as a single file, handed over in a way the client can check has arrived intact.

1. Shut the virtual machine down properly. Do not export a running or suspended machine.
2. Export it from your hypervisor as a single appliance file. Name it `BUC3129_Group<number>_Server.ova`.
3. Calculate the file's **SHA-256 checksum** and record it. In the field, this is how the receiver proves the file was not damaged in transfer.
4. The file will be several gigabytes, which is too large for MUBSEP. Upload it to a cloud drive, set sharing to *anyone with the link can view*, and test the link from a different account.
5. On **MUBSEP**, submit four things in the Assignment 3 area: the handover report, the credentials sheet, the link to the exported machine, and the checksum.

A link I cannot open counts as no submission. The link must still work on the day of marking.

> *Start this on the last weekend, not in the last hour. Exporting is slow, the file is large, and the upload is slower still.*

Dividing the work

Five people, eleven tasks. Every member must **lead** at least two tasks and assist on the others. **Rotate the leads from Assignment 2**: whoever led the domain build should not lead the storage build.

One member is the **team leader**, who sends the weekly status updates and approves change requests together with one other member. In the field, no change to a live system is approved by the person making it.

Task	Led by	Signature
1 — Resilient storage		
2 — Folder structure		
3 — Permissions		
4 — Network share		
5 — Quotas		

6 — Service inventory		
7 — Scheduled backup		
8 — Health monitor		
9 — Runaway process		
10 — Fault ticket		
11 — Export and handover		
Team leader		

A member whose name appears against no task cannot pass this assignment. Fill this in as you go, not the night before.

Weekly status updates

Every Thursday, the team leader sends one short email to **hnamuwaya@mubs.ac.ug**, with the subject line *BUC3129 Group <number> — Status Week <1/2>*. Five lines at most:

- What was completed this week
- What is planned for next week
- Anything blocking you, and what you need to unblock it

Update	Due
Week 1 status update	Thursday 8th October 2026
Week 2 status update	Thursday 15th October 2026
Final submission on MUBSEP	Thursday 22nd October 2026, 12pm

These are marked, and they are how real clients know a contractor is in control of a job. A group that is stuck in week one and says so will get help. A group that is stuck and says nothing until week three will not.

The handover report

One report per group, covering the WHOLE project: Assignment 2 and Assignment 3 together.

This is not a second short report. It is the handover pack you would give a client, so that the next administrator can run the system without calling you. **No more than twenty pages, excluding screenshots and appendices.**

1. **Cover page:** company name, group number, all members' names and registration numbers, domain name, computer number, and the date.
2. **Executive summary:** one page. What the client asked for across both assignments, and what now exists. Written so the Head Teacher, who is not technical, can read it and understand what she has.
3. **The work order, decoded:** the Head Teacher's note rewritten as six numbered technical requirements.
4. **The environment as built:** the whole system, described and drawn: the server, the second machine, the domain, the OUs, the storage, the shares and the scheduled jobs. Include one simple diagram.
5. **Assignment 2 work, revisited:** two or three pages. What you built then, and **what you would now do differently**, having learned more. Honest reflection scores here; repeating the old report does not.
6. **Assignment 3, Tasks 1 to 11:** one section each. For every task: what you built, the evidence, and your reasoning. Explain. Do not paste lists of commands.

7. **Problems and how you solved them:** at least **five** genuine problems across the whole project. This section is marked. A report without it looks dishonest.
8. **Self-evaluation:** see the next section.
9. **Recommendations to the client:** if this school actually had forty staff and three hundred students, what would you change? Name three things and say why.
10. **Who did what:** the signed tables from Assignment 2 and Assignment 3.
11. **Sources:** everything you used.

Appendices, required:

- **Appendix A:** the four completed change requests (Tasks 1, 3, 5, 7)
- **Appendix B:** the completed ticket from Task 10
- **Appendix C:** an **administrator's runbook**: step-by-step instructions, in plain language, for five routine jobs. *Add a new member of staff and give them the right access. A teacher leaves. Restore a file that was deleted yesterday. Check that last night's backup ran. A user says the disk is full.* Written so that someone who has never seen your system could follow them.
- **Appendix D:** the acceptance test checklist, with every test run and its result
- **Appendix E:** copies of your two weekly status updates
- **Appendix F:** the signed self-evaluation forms

Every screenshot must show your domain name or a group member's account name. Screenshots that could belong to any group score nothing.

Never put passwords in the report. Submit them on a separate one-page credentials sheet on MUBSEP. In the field, documentation is shared widely and credentials are not.

Acceptance and the oral defence

In the session after submission, your group demonstrates the live system to me, **acting as the client**, using your own acceptance test checklist. I will choose which tests to run. This is the sign-off: a requirement only counts as delivered if it passes in front of the client.Presentation schedule will be shared In the due course.

Appendix templates

Copy these into your report and fill them in. Use one change request per task.

Change request template

Field	Entry
Change request number	CR-G<group>-<01, 02, …>
Company and group	
Requested by (client)	Head Teacher
What will change	
Why (which requirement it meets)	
Systems affected	
Risk if it goes wrong	Low / Medium / High, and why
Rollback plan	How you will undo it, including the snapshot name
How you will test it	
Planned date and time	

Approved by (team leader and one other member)	
Outcome after the change	Successful / Rolled back, and notes

Support ticket template

Field	Entry
Ticket number	TKT-G<group>-001
Reported by and time	
The user's report, word for word	
Priority	P1 Whole school stopped · P2 A department stopped · P3 One person affected · P4 Request or question
Why that priority	
Checks made, in order, with the result of each	1. … 2. … 3. …
Root cause	
Fix applied	
How the fix was tested	
Time opened, time closed, total time	
Closed by	

Weekly status update template

Field	Entry
Subject line	BUC3129 Group <number> — Status Week <1/2>
Completed this week	
Planned for next week	
Blocked by, and what we need	
Sent by (team leader)	

Acceptance test checklist template

Requirement	Test	Expected result	Actual result	Pass or fail
Storage survives a dead disk	Remove one disk; open a file on the drive	File opens		
Only bursar and Head Teacher reach finance	Sign in as a teacher; open the finance share	Access denied		
Teachers cannot alter each other's files				
Students are limited to their quota				

Backup runs nightly by itself				
Backup leaves a record				

Add a row for every requirement you delivered. The first two rows are done for you as examples.