1. How would you centralize access and management of the 3 AWS accounts?

Since the use case is for large enterprise, I recommend AWS Control Tower to set up and govern a secure, multi-account AWS environment. I would use AWS Organizations to segregate accounts into OU and apply right privileges via Service Control Policies. Integrating AWS IAM Identity Canter with Identity provider like Microsoft AD provides single source of truth for user management.

2. How would you aggregate the collected data from all accounts into a single, easily

digestible format?

My proposed solution uses AWS Systems Manager Ansible playbooks that deploys playbooks by a specific tag , for eg: “OS = Linux”. The output logs are pushed to a centralized S3 bucket in Logs Archive account.

3. How would your solution scale if the company acquires more companies and AWS

accounts in the future?

The solution is using AWS Systems Manager and S3 which are both serverless components. Hence there is no issue of scaling.

I also propose using AWS Transit Gateway for cross account communication. This easier to maintain and scale than VPC peering.

Deliverables :

> Provide a high-level architectural diagram of your proposed solution.

Github repo has Management Architecture diagram and Network Architecture diagram

> Provide the ansible playbook

I have made separate playbook for Linux and Windows in github for faster execution

> Summarize the various component involved in the process

AWS Solutions Manager has a cronjob once a day to run an ansible playbook. This playbook is run on instances with specific tag for eg: “OS = Linux”. This runs a task to collect disk wise usage details and stores it in centralized S3 bucket in Logs Archive account.

Note:

1)There are no access keys and secret keys involved here, the EC2 gets the permission via IAM role for SSM. The Instances can be in private subnet an still work.

2) S3 bucket is private with restrictive permission on “PutObject” for the EC2 and bucket level policy allowing only certain AWS accounts to write to the bucket.

Solution demo is available here: <https://drive.google.com/file/d/1vu0kVN4e2AYvfEdvP11U6IOoJtkOIV1g/view?usp=drive_link>