Pavan Weekly Reports

Week 1

	Date	Tasks	Next Week
1	09/16/24		
2	09/17/24	Discussion with Advisor	
3	09/18/24	Connected with team mates	
4	09/19/24		
5	09/20/24	I refreshed my knowledge of Linux.	
6	09/21/24	I updated Linux in VirtualBox again, as we had already installed it during our data engineering course	
7	09/22/24		

Week 2

	Date	Tasks	Next Week
1	09/23/24	I installed WSL as it is better than VirtualBox.	
2	09/24/24	I experimented with Docker and kubernetes using ChatGPT and watched some YouTube videos.	
3	09/25/24		
4	09/26/24	I believe LinkedIn Learning will be beneficial, so I started learning on the platform.  CertificateOfCo... ker.pdf 03 Oct 2024, 01:09 AM	
5	09/27/24	I experimented with Kubernetes and Nvidia	

6	09/28/24	Our advisor suggested that we need to install Linux via dual boot and started working by installing	
7	09/29/24	I decided to work on some projects, so I started with Docker through LinkedIn Learning.  Docker: Your First Project **CertificateOfCo… ect.pdf** 03 Oct 2024, 01:11 AM	

Week 3

	Date	Tasks	Next Week
1	09/30/24	Experimented with NVIDIA drivers and created two users to simulate	
2	10/1/24		
3	10/2/24	working with NVIDIA drivers and how to use them to stimulate gpus simulate	
4	10/3/24		
5	10/4/24		
6	10/5/24	Gained practical experience with Kubernetes and load balancing.  Learning Kubernetes **CertificateOfCo… tes.pdf** 09 Oct 2024, 07:36 PM	
7	10-/6/24		

Week 4

	Date	Tasks	Next Week
1	10/7/24	started working on creating VM to create GPU in personal system for testing	

2	10/8/24	Expermenting with VM's and with the containers in it.	
3	10/9/24	-	
4	10/10/24	-Ran a Container: Successfully created and entered a Docker container running Ubuntu. -Tried to Install NVIDIA Docker: Attempted to set up NVIDIA Docker to allow our container to access the GPU resources on our host machine. *Verified NVIDIA Driver Installation.* - Working on ubuntu Adding Correct NVIDIA Docker Repository:* - We got some errors here *Install NVIDIA Docker:* - Pending	
5	10/11/24	Experimented and got some issues with Nvidia container toolkit	
6	10/12/24	1.Docker Container Creation: We created and ran a Docker container using the Ubuntu 22.04 image with GPU support enabled. 2.NVIDIA Driver Check: We confirmed that the NVIDIA GPU is accessible within the container by successfully running the nvidia-smi command, which displays GPU details.	
7	10/13/24	Experimented with two systems(remote and host) . We now have access to the GPU on the remote system from our own machine	Research more on docker and Kubernetes.Now we have access to our own system GPU and need to work on how to manage and implement those GPU with multiple users.

Week 5:

	Date	Tasks	Next Week
1	10/14/24	Continued focusing on Docker deployment and began integrating Docker with Kubernetes.	-
2	10/15/24	I’ve set up JupyterHub and gained an understanding of how it operates.	
3	10/16/24	Held a meeting with the advisor to discuss progress.	
4	10/17/24	I've deployed JupyterHub in a container, and while it's functioning, the GPU isn't accessible within the Jupyter notebooks.	-
5	10/18/24	-	-
6	10/19/24	I’ve tried deploying multiple containers with different images, but none of them have successfully enabled GPU access within Jupyter notebooks.	-
7	10/20/24	I worked with the Kubernetes cluster dashboard and then shifted focus to enabling GPU support within the notebook, experimenting with different approaches.	The GPU is not yet accessible within the Jupyter notebooks, so the next step will be to troubleshoot and resolve the issue with GPU access in the notebook environment.

Week 6

	Date	Tasks	Next Week
1	10/21/24	-	-
2	10/22/24	-	-
3	10/23/24	We had a conversation with advisor and gained some valuable insights.	

4	10/24/24	We set up a container that operates within the same network, allowing multiple users to access JupyterHub without any authorization requirements.	-
5	10/25/24	-	-
6	10/26/24	Experimented with Helm charts, focusing on user management and resource allocation.	-
7	10/27/24	Explored the compatibility between Docker and Kubernetes regarding resource scaling.	Need to work with MetalLB for load balancing and using NFS for resource allocation in my Kubernetes cluster.

Week 7:

	Date	Tasks	Next Week
1	10/28/24	-	-
2	10/29/24	-	-
3	10/30/24	**Set Up Minikube for GPU Access**: • Installed Minikube and initially configured it with the **Docker driver** to explore GPU access within a local Kubernetes environment. • Encountered issues with Minikube not detecting the GPU resources.	
4	10/31/24	**Attempted GPU Integration with Minikube**: • Applied the **NVIDIA device plugin** in Kubernetes to enable GPU resource detection. • Encountered NVML (NVIDIA Management Library) errors indicating that Minikube's environment could not	

		recognize the NVML library, preventing Kubernetes from detecting the GPU.	
5	11/01/24		
6	11/02/24	-	-
7	11/03/24	-	**-**

November month:

	Date	Tasks	Next week
1	November month	1. **Kubernetes and JupyterHub Refinements:** ◦ Decided to move forward with the setup using **CPU and storage resources** only, omitting GPU for the time being. ◦ Adjusted Kubernetes pod resource configurations by modifying `limits` and `requests` settings for **CPU and memory** to ensure optimal performance. 2. **System Optimization:** ◦ Focused on efficient resource management for Kubernetes pods, enhancing scalability and stability within the cluster. ◦ Continued development of the **dashboard for cluster monitoring**, emphasizing resource allocation insights for multi-tenancy. 3. **Load Balancing and Storage Management:** ◦ Integrated and tested **MetalLB for load balancing**, ensuring seamless distribution of network traffic across cluster nodes. 4. **Authentication and User Management:** ◦ Streamlined **JupyterHub's native authentication**, prioritizing	There is an issue with the Storage and GPU. So, started with CPU and storage for the time being which caused errors. Need to work on it

		simplicity with **user ID/password login** (no token requirement). ◦ Ensured the login page allowed seamless access to Jupyter Notebooks for users on the same network. 5. **Networking and Accessibility:** ◦ Verified that the **dashboard and JupyterHub** can be accessed from any device on the local network, aligning with project goals for flexibility. 6. **Experimentation and Testing:** ◦ Conducted tests to validate the **dashboard functionality**, including the allocation of CPU/memory resources and cluster availability metrics. ◦ Continued fine-tuning the deployment to align with your overall objectives for efficient multi-user and resource management.	

December:

	Date	Tasks	Next Week
1	**12/02/24**	-	-
2	12/03/24	-	-
3	**12/04/24**	-	
4	12/05/24	-	
5	12/06/24	-	-
6	12/07/24	Working on the storage issue for the pods	-
7	12/08/24	**-**	**--**

December 2nd week:

	Date	Tasks	Next Week
1	**12/09/24**	**Persistent Volume (PV) and Persistent Volume Claim (PVC)** • Created PVCs for pods and faced an issue with them.	The issue likely arose during PV-PVC binding or due to specific NFS settings.
2	12/10/24	The NFS storage successfully handled concurrent user workloads, with Persistent Volume Claims (PVCs) providing reliable, consistent, and scalable access to shared storage across multiple pods in the Kubernetes cluster. Data consistency and performance metrics under multi-user scenarios confirmed the robustness of the setup, ensuring it meets our requirements	-
3	**12/11/24**	The Kubernetes cluster has been successfully configured with a fully operational NFS storage system, ensuring seamless handling of user workloads. All pods, including user-specific pods, are being created dynamically and are functioning as expected. JupyterHub is fully operational, providing multi-user access with secure authentication, and the Kubernetes dashboard is accessible, offering comprehensive visibility and management of the cluster. The project is now stable.	
4	12/12/24	Worked on report creation(Helped the team)	
5	12/13/24	-	-
6	12/14/24		-
7	12/15/24		**--**