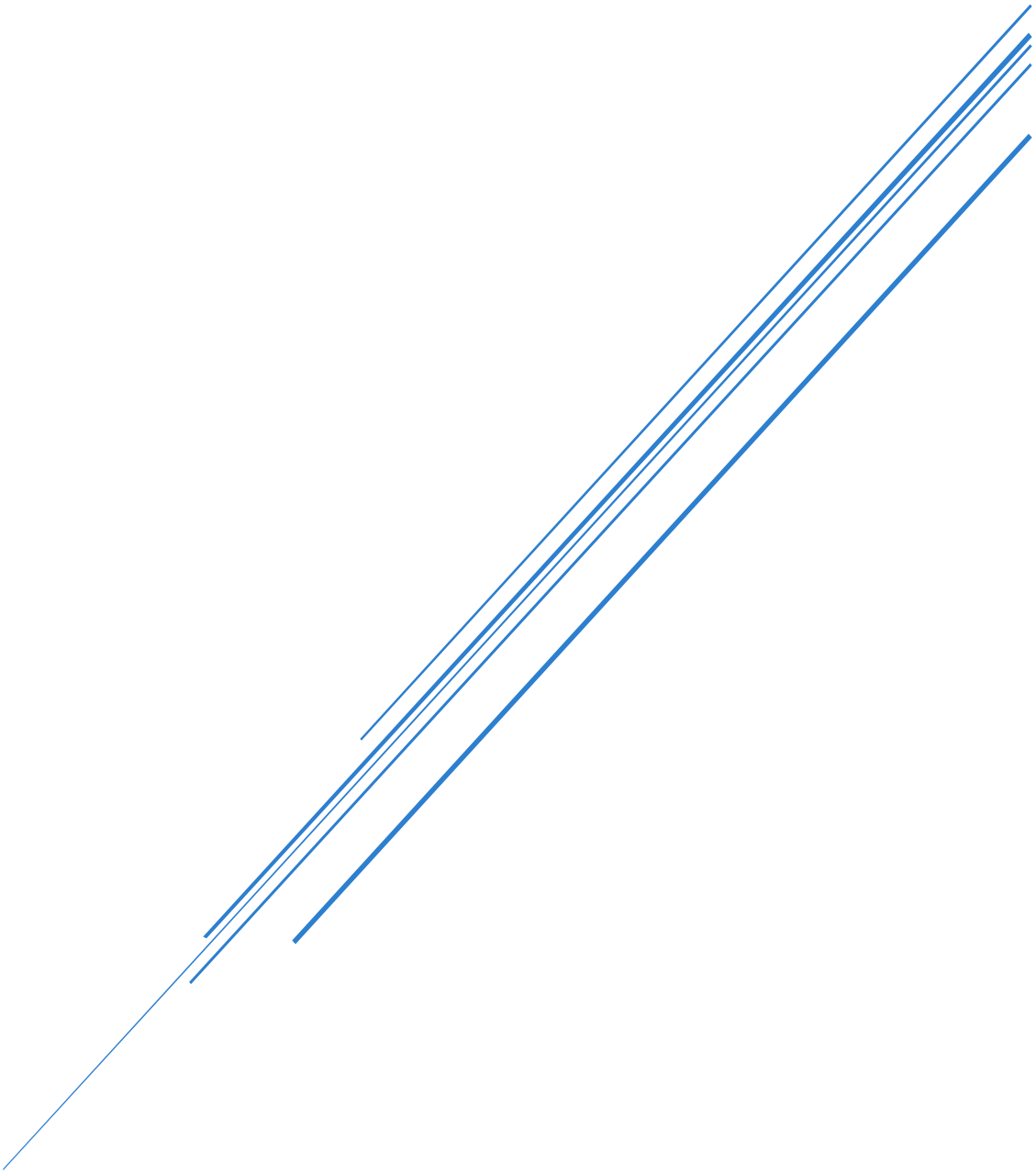
RESEARCH QUESTIONS

Assessment 2



Eugene Parker
DevOps Engineer

This following research questions acts a guide to understand the key requirements of the implementation process & configuration involved before committing to it .

1. GPU Slicing & Virtualization Techniques

- What are the key differences between **NVIDIA Time-Slicing** and **Multi-Instance GPU (MIG)?**
- How does **time-slicing** impact **latency and throughput** for AI workloads?
- What are the **hardware and software requirements** for enabling **MIG** on AWS GPU instances?
- Which **GPU instance types (G4, G5, P4, etc.)** support **MIG vs. time-slicing** on AWS?
- What are the **performance trade-offs** of GPU slicing compared to dedicated GPU allocation?

2. NVIDIA GPU Operator and Kubernetes Configuration

- How does the **NVIDIA GPU Operator** work within **EKS**?
- What is the correct **installation and configuration** process for **NVIDIA GPU drivers** on EKS?
- How does the **NVIDIA device plugin** handle **GPU allocation** within Kubernetes?
- How can we configure **time-slicing** for fractional GPU allocation in Kubernetes?
- What are the **best practices for monitoring GPU utilization** in Kubernetes?

3. AI Workload Optimization for GPU Sharing

- How can AI workloads be **optimized for fractional GPU allocation**?
- What **Kubernetes resource requests and limits** should be set for shared GPUs?
- How does **batch size, inference time, and parallel processing** affect GPU utilization when sliced?
- What **AI models and frameworks** (TensorFlow, PyTorch, etc.) support **GPU sharing efficiently**?
- How can **GPU memory fragmentation** be minimized when using **time-slicing**?

4. Karpenter Autoscaler & GPU-Aware Scheduling

- How does **Karpenter Autoscaler** handle **GPU node provisioning** in EKS?
- What configurations are needed to ensure **Karpenter provisions GPU instances dynamically**?
- How does **Karpenter compare with Cluster Autoscaler** for managing GPU nodes?
- Can Karpenter automatically **scale GPU instances based on fractional GPU requests**?
- What are the **best instance types** for GPU auto-scaling with Karpenter?

---

5. AWS GPU Instance Cost Optimization

- How do **AWS pricing models** (On-Demand, Spot, Savings Plans) impact **GPU costs**?
- How can **Spot Instances** be leveraged for AI workloads while maintaining reliability?
- What are the **cost differences between G4, G5, P4 instances** for GPU workloads?
- Can AWS **Savings Plans** or **Reserved Instances** help in reducing GPU costs for EKS clusters?
- What are the **cost savings projections** when using **GPU slicing vs. dedicated GPUs**?

6. Infrastructure as Code (IaC) & Automation

- How can **Terraform** or **Helm charts** be used to automate **NVIDIA GPU Operator deployment**?
- How do we implement **GPU slicing configurations** using Kubernetes manifests?
- What **CI/CD pipelines** can be used for **deploying AI workloads** on GPU-enabled EKS clusters?
- How can **custom Karpenter NodePool policies** be managed using Infrastructure as Code (IaC)?
- What are the **best automation practices** for maintaining **GPU workload efficiency**?

7. Monitoring and Troubleshooting GPU Workloads

- What **monitoring tools** (Prometheus, Grafana, NVIDIA DCGM) can track **GPU utilization** in EKS?
- How can **GPU performance bottlenecks** be identified and mitigated?
- What are the **most common GPU allocation issues** in Kubernetes, and how can they be solved?
- How can **nvidia-smi and DCGM metrics** be used to optimize **GPU performance**?
- How do we implement **alerting for underutilized or idle GPUs** in Kubernetes?