

// Jivops

Private AI en VCF

Guía completa — 20 lecciones

Fundamentos · Plataforma de IA · Gobierno y seguridad · Operación

Contenido

Módulo 1: Fundamentos y hardware

- Lección 01 — Qué es Private AI y por qué en VCF
- Lección 02 — Arquitectura de VMware Private AI Foundation
- Lección 03 — Hardware: GPU, red y almacenamiento
- Lección 04 — Compartir y asignar GPU: vGPU, MIG y passthrough
- Lección 05 — Licencias y controladores NVIDIA

Módulo 2: Plataforma de IA

- Lección 06 — Deep Learning VM
- Lección 07 — Clústeres Kubernetes con GPU
- Lección 08 — Almacén y catálogo de modelos
- Lección 09 — Servidores de inferencia
- Lección 10 — Bases de datos de vectores y RAG

Módulo 3: Gobierno y seguridad

- Lección 11 — Privacidad y soberanía de datos
- Lección 12 — Acceso y multitenencia
- Lección 13 — Entornos desconectados
- Lección 14 — Automatización y autoservicio
- Lección 15 — Seguridad de modelos y prompts

Módulo 4: Operación y capacidad

- Lección 16 — Monitorización de GPU
- Lección 17 — Rendimiento y optimización
- Lección 18 — Planificación de capacidad y costes
- Lección 19 — Ciclo de vida y actualizaciones
- Lección 20 — Proyecto final: plataforma de IA privada para la empresa

Módulo 1: Fundamentos y hardware

Lección 01 — Qué es Private AI y por qué en VCF

Ejecutar cargas de IA generativa y aprendizaje automático (entrenamiento, ajuste fino e inferencia) en infraestructura propia o controlada, usando la plataforma VCF con GPU, de modo que los datos y modelos permanecen bajo el control de la organización, con la gestión, seguridad y operación ya conocidas de vSphere.

Privacidad y soberanía de los datos, cumplimiento normativo y control de costes y rendimiento.

Lección 02 — Arquitectura de VMware Private AI Foundation

VCF como base (vSphere, vSAN, NSX, automatización y operaciones), hosts con GPU NVIDIA, el software NVIDIA AI Enterprise, plantillas de Deep Learning VM, clústeres Kubernetes con GPU, almacén de modelos, bases de datos con vectores y servicios de automatización de autoservicio.

Reutiliza las herramientas, operación y seguridad que ya se conocen de vSphere.

Lección 03 — Hardware: GPU, red y almacenamiento

Las GPU (modelo, memoria y generación), el ancho de banda entre GPU (NVLink/NVSwitch), una red rápida de baja latencia (100 Gb o más, con RDMA donde proceda) para entrenamiento distribuido y almacenamiento rápido para datos y modelos, además de CPU, memoria, alimentación y refrigeración.

Porque el modelo debe caber en la memoria de la GPU, que depende del número de parámetros y la precisión.

Lección 04 — Compartir y asignar GPU: vGPU, MIG y passthrough

Por passthrough (DirectPath I/O), donde la VM usa la GPU completa; o con vGPU, que reparte la GPU en perfiles (por time-slicing, o con MIG para particiones aisladas por hardware en las GPU compatibles), permitiendo compartirla entre varias VM.

Cuando la carga necesita la GPU completa y el máximo rendimiento, como entrenar o servir modelos grandes.

Lección 05 — Licencias y controladores NVIDIA

El controlador de host de vGPU instalado en cada ESXi (gestionado con vSphere Lifecycle Manager), el controlador invitado en las VM, y licencias de NVIDIA AI Enterprise servidas por un servidor de licencias, con versiones compatibles entre ESXi, controlador y GPU.

La GPU funciona con rendimiento o funciones reducidas hasta obtener la licencia.

Módulo 2: Plataforma de IA

Lección 06 — Deep Learning VM

Una máquina virtual preconfigurada con controladores de GPU, CUDA, marcos como PyTorch o TensorFlow y herramientas de ciencia de datos (por ejemplo JupyterLab y contenedores de NGC), que se despliega con rapidez para que los científicos de datos trabajen sin montar el entorno a mano.

Es rápido y reproducible: no se monta el entorno a mano y todos usan versiones coherentes.

Lección 07 — Clústeres Kubernetes con GPU

Con clústeres VKS cuyos nodos worker tienen GPU (vGPU o passthrough) y el GPU Operator de NVIDIA, que instala controladores y plugin de dispositivos para que los pods soliciten recursos `nvidia.com/gpu` en sus manifiestos.

Declarando el recurso `nvidia.com/gpu` en los límites de sus contenedores.

Lección 08 — Almacén y catálogo de modelos

Para guardar y versionar modelos aprobados (por ejemplo en un registro tipo Harbor o una galería de modelos), con control de acceso y escaneo, de modo que los equipos despliegan solo modelos validados y la procedencia queda documentada.

Da control y trazabilidad de la procedencia, con despliegues reproducibles y más seguros.

Lección 09 — Servidores de inferencia

Con servidores de inferencia como vLLM o Triton, o microservicios NVIDIA NIM empaquetados en contenedores, que cargan el modelo en GPU y exponen una API compatible (por ejemplo tipo OpenAI) con procesamiento por lotes y gestión eficiente de memoria.

Mejora el rendimiento (throughput) y aprovecha mejor la GPU.

Lección 10 — Bases de datos de vectores y RAG

Generación aumentada por recuperación: se convierten los documentos en embeddings guardados en una base de datos de vectores (por ejemplo PostgreSQL con pgvector gestionada con Data Services Manager); ante una pregunta se recuperan los fragmentos relevantes y se pasan al modelo como contexto.

Usa datos actualizados sin reentrenar, con menor coste y con fuentes citables.

Módulo 3: Gobierno y seguridad

Lección 11 — Privacidad y soberanía de datos

Manteniendo datos, modelos e inferencia dentro de la infraestructura de la organización, clasificando los datos, cifrando en reposo y en tránsito, aplicando minimización y anonimización cuando proceda, y cumpliendo la normativa aplicable de protección de datos y residencia.

Los datos personales o sensibles, que conviene anonimizar o minimizar antes.

Lección 12 — Acceso y multitenencia

Con proyectos o namespaces por equipo, cuotas de GPU, permisos por roles con identidad corporativa, y aislamiento de red y de almacenamiento, de modo que cada equipo ve y consume solo lo suyo.

Porque la GPU es un recurso escaso y caro: hay que repartirla de forma justa y controlar el gasto.

Lección 13 — Entornos desconectados

Trayendo modelos, contenedores, controladores y paquetes por un proceso controlado a registros y repositorios internos (air-gap), verificando firmas y escaneando todo antes de importarlo, y gestionando las actualizaciones por lotes.

Su origen y licencia, la firma y un escaneo de seguridad.

Lección 14 — Automatización y autoservicio

Con un catálogo de autoservicio en la automatización de VCF donde los usuarios solicitan una Deep Learning VM o un clúster Kubernetes de IA con GPU, con parámetros, aprobaciones, cuotas y plantillas definidas por la plataforma.

Rapidez y autonomía: obtienen entornos estándar sin esperas.

Lección 15 — Seguridad de modelos y prompts

Inyección de instrucciones (prompt injection), fuga de datos en las respuestas, modelos o dependencias manipulados, y salidas no fiables; se mitigan con validación de entradas y salidas, filtros, mínimo privilegio para herramientas del agente, modelos de origen verificado y registro de uso.

Para limitar el daño si el modelo es manipulado, aplicando mínimo privilegio.

Módulo 4: Operación y capacidad

Lección 16 — Monitorización de GPU

Con métricas de GPU (utilización, memoria, temperatura, potencia) mediante DCGM y su exportador a Prometheus, junto con las métricas de vSphere y Kubernetes, paneles en Grafana o en las herramientas de operaciones de VCF y alertas por saturación o fallos.

Utilización baja con la GPU reservada pero ociosa, lo que supone coste desperdiciado.

Lección 17 — Rendimiento y optimización

Eligiendo el tamaño de modelo adecuado, usando cuantización, procesamiento por lotes continuo, caché de claves y valores, paralelismo de tensores entre GPU y ajustando el tamaño del contexto y la concurrencia, midiendo latencia y tokens por segundo.

Latencia, tiempo hasta el primer token, tokens por segundo y rendimiento global.

Lección 18 — Planificación de capacidad y costes

Estimando los modelos y usuarios concurrentes, la memoria necesaria por modelo y por contexto, el reparto entre entrenamiento e inferencia, el margen para picos y fallos, y midiendo la utilización real para decidir si compartir con vGPU o MIG o dedicar GPU completas.

Da responsabilidad y permite optimizar y repartir costes entre equipos.

Lección 19 — Ciclo de vida y actualizaciones

Siguiendo la matriz de compatibilidad entre VCF, ESXi, controlador de vGPU, NVIDIA AI Enterprise, Kubernetes y los modelos; actualizando por fases con vSphere Lifecycle Manager, probando antes en un entorno previo y manteniendo copias de configuraciones y modelos.

Hay que migrarlas (con vMotion de vGPU y una pausa breve, si se soporta) o apagarlas antes del mantenimiento.

Lección 20 — Proyecto final: plataforma de IA privada para la empresa

Dominio de carga de trabajo de VCF con hosts GPU y red rápida, vGPU o MIG para compartir y passthrough para cargas grandes, licencias NVIDIA AI Enterprise, Deep Learning VM y clústeres VKS con GPU en autoservicio, almacén de modelos, vLLM o NIM, pgvector para RAG, cuotas y RBAC, monitorización con DCGM y actualizaciones por fases.

Validar el modelo, guardarlo en el almacén, desplegarlo en un servidor de inferencia, probarlo y monitorizarlo.