Docker for Machine Learning!

Paniraja Koppa
Technical Marketing Engineer

Haseeb Niazi
Technical Marketing Engineer

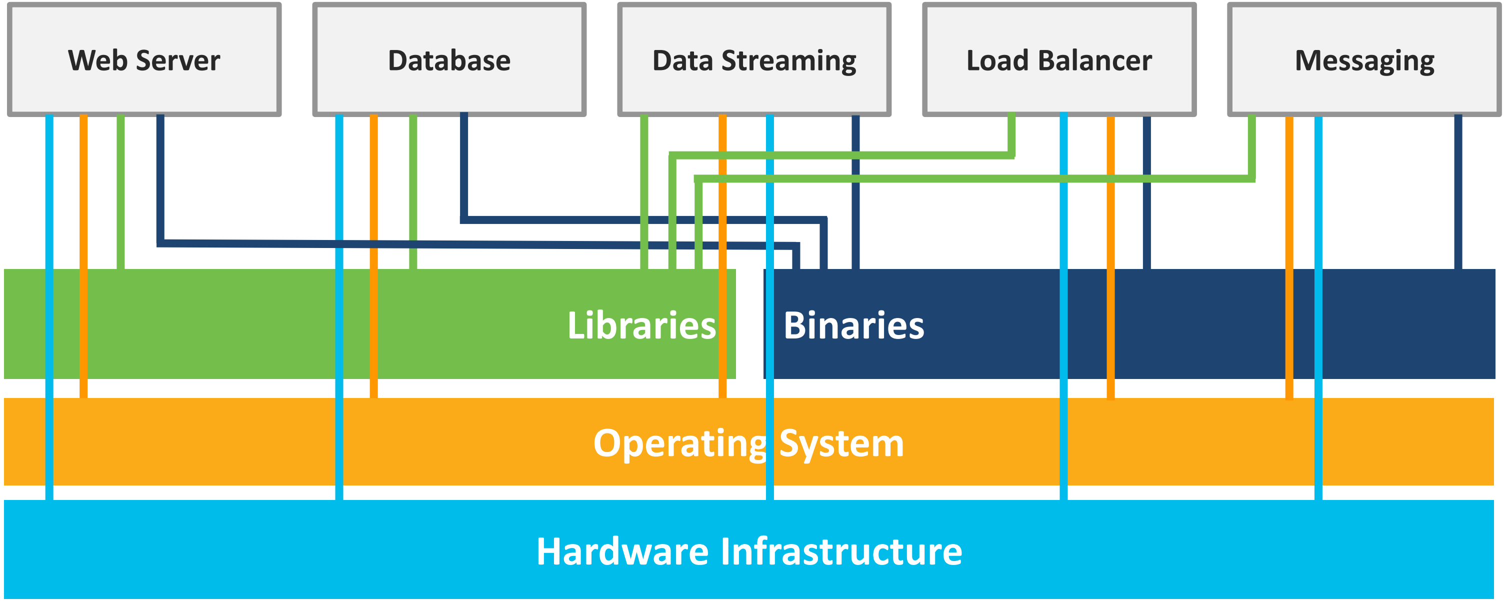
DGTL-BRKPRG-2438

Agenda

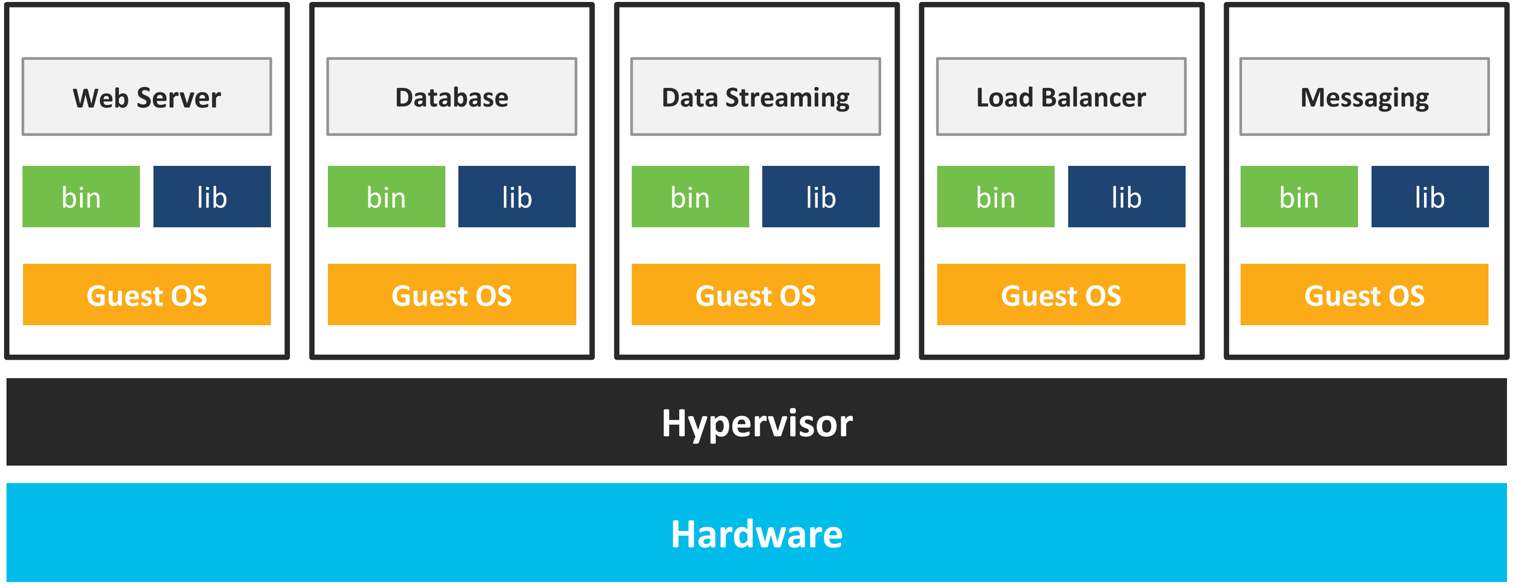
- Machine Learning Workflows using Docker Containers
  - Docker basics
  - Networking and storage options in Docker
- Docker during machine learning model development
  - Building Docker image for a simple ML problem
  - Initialize and running container
- Deploy Machine Learning Models at Edge using Docker
- Docker Containers with GPU Support
  - Cisco UCS GPU enabled platforms: C480 ML M5, Cisco UCS C220/C240 M5
  - Setting up the platforms to support Docker environment with GPU support
  - Downloading and running Tensorflow containers
- Cisco Converged Infrastructure Solutions for AI/ML
  - FlexPod and FlashStack AI/ML solutions
  - GPU support in the VMware environments – NVIDIA vComputeServer

Docker Basics

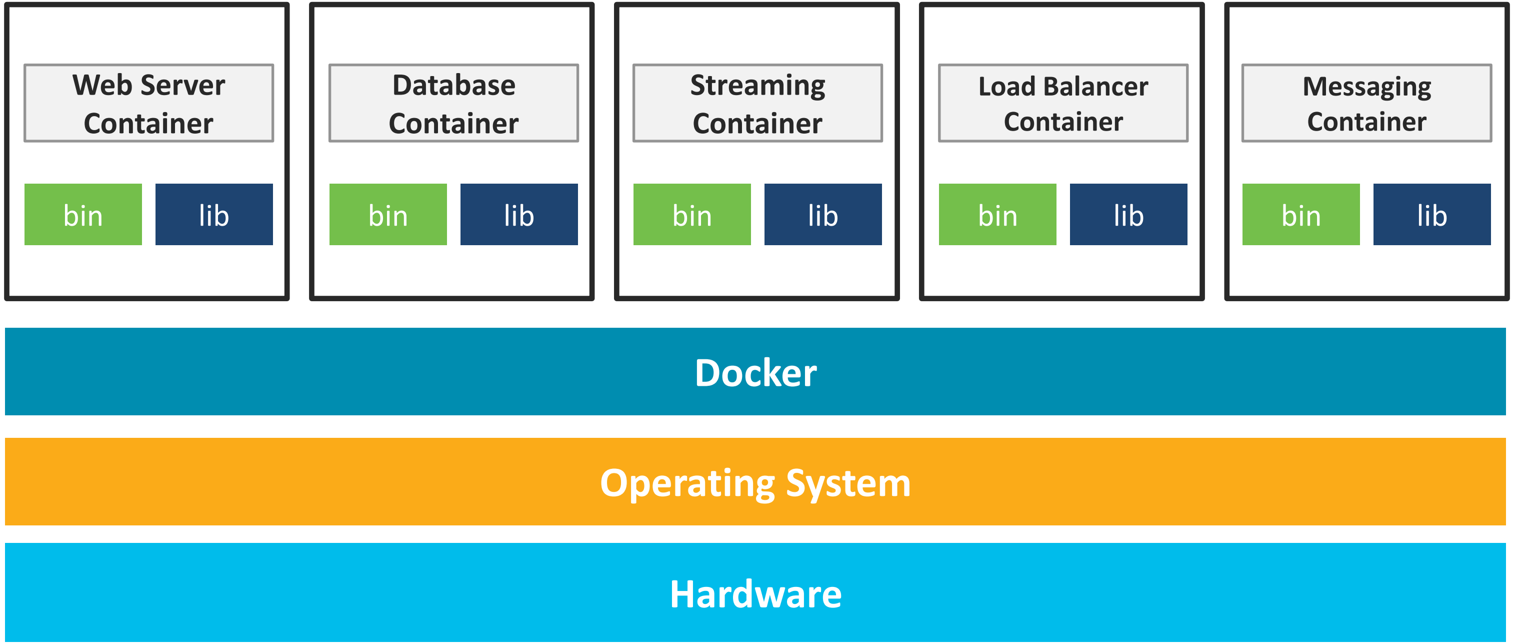
Deployments on Physical Infrastructure



Deployment on Virtual Machine



Application deployment with Container



What is Docker ??

Docker allows us to package and run applications in an isolated environment

Develop and share layered applications

Package code + its dependencies to enable application to run in an isolated

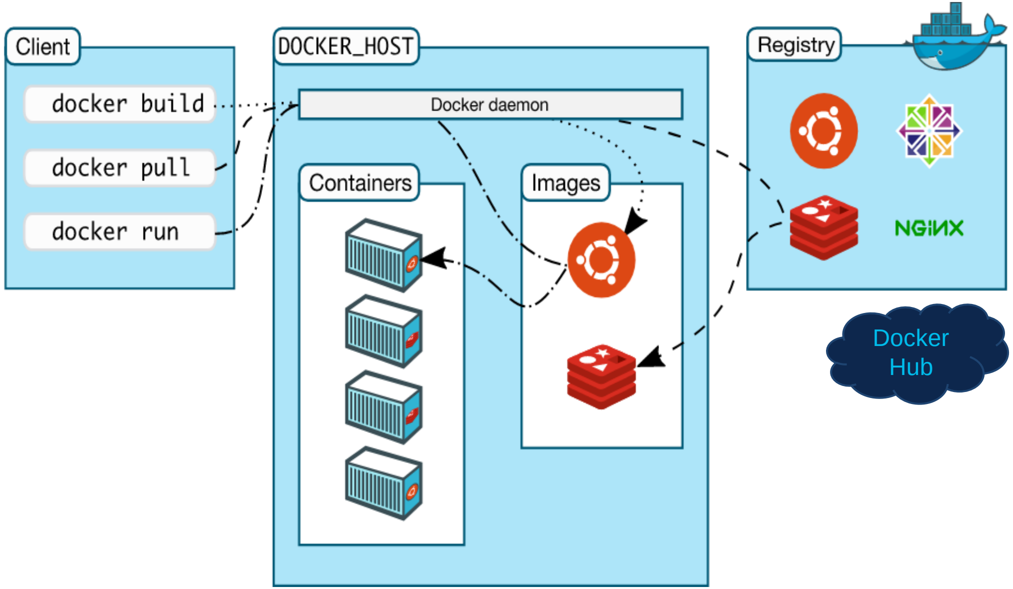
Share the same Operating System Kernel

Uses kernel features: namespaces and cgroups

"By 2022, more than 75% of global organizations will be running containerized applications in production, which is a significant increase from fewer than 30% today"

Gartner

Architecture



Docker Daemon

Daemon (dockerd) which interacts with OS and performs all kind of services

Docker Client

CLI tool to interact with Daemon

Image

Is the application package

Container

Running instance of image

Registry

Repository of images

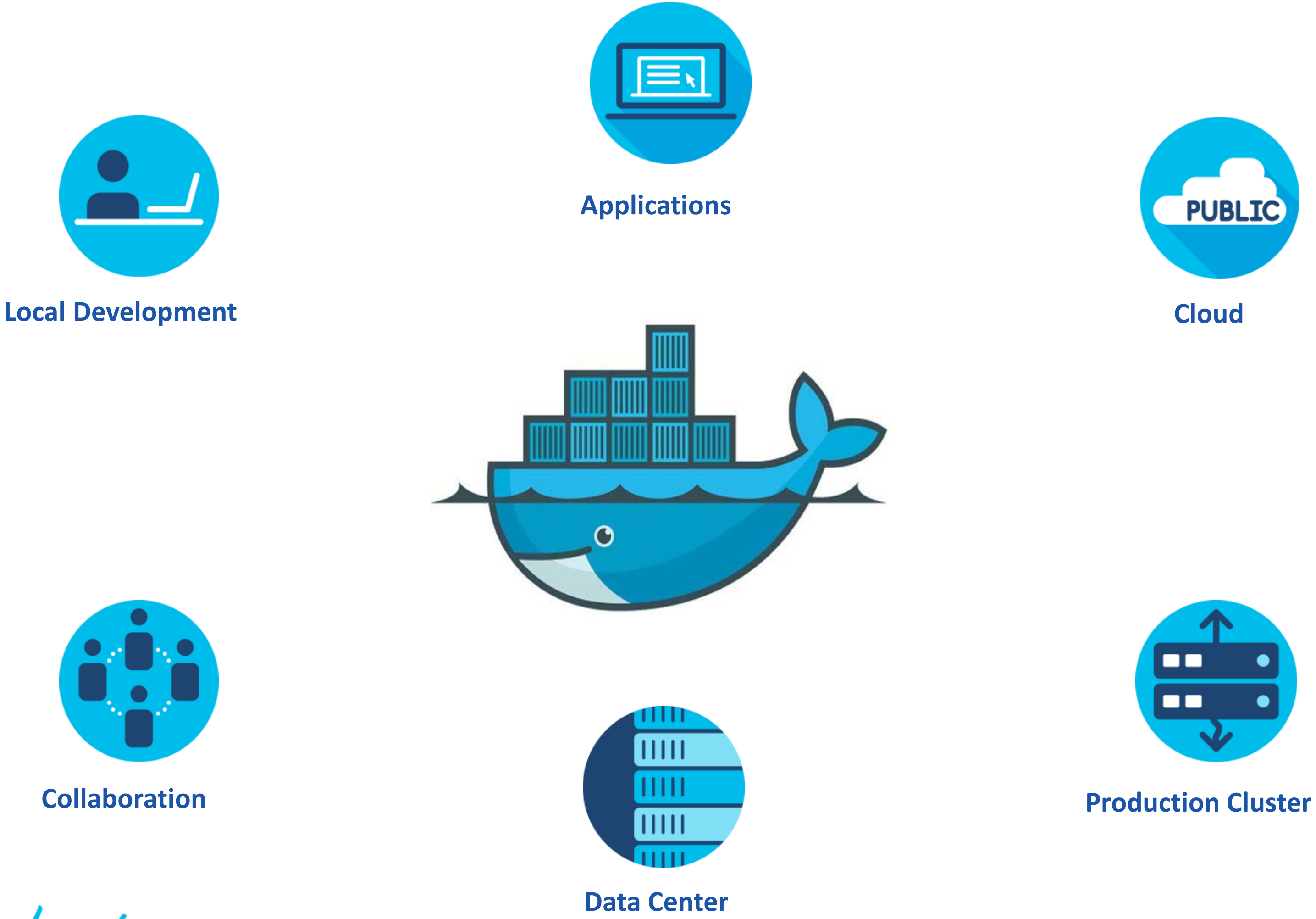
Default: Docker Hub

Analogy

Object Oriented paradigm

Why containerize ML workflow ?

- Reproduce experiments easily!
- Solves -- it works on my machine problem!
- Reduced Complexity to develop and deploy
- Easy sharing - No complex software dependencies.
- Dev -> Test -> Production easier and faster
- Easier to clear data in large scale
- Speed…. Speed…. Speed….

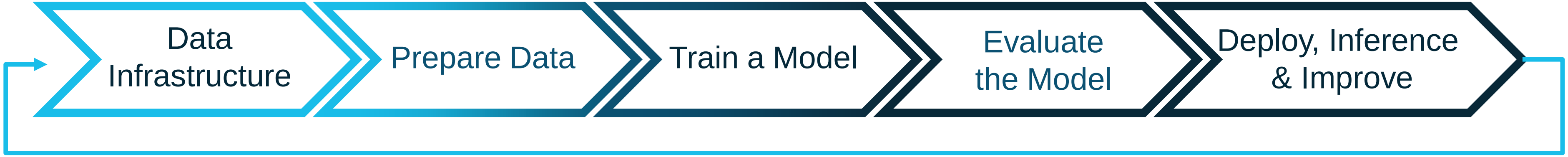

Applications
Local Development
Cloud
PUBLIC
Collaboration
Production Cluster
Data Center

Simplify ML Workflows using Docker

Demo -1

Docker during ML Model Development

Machine learning Workflow Summary

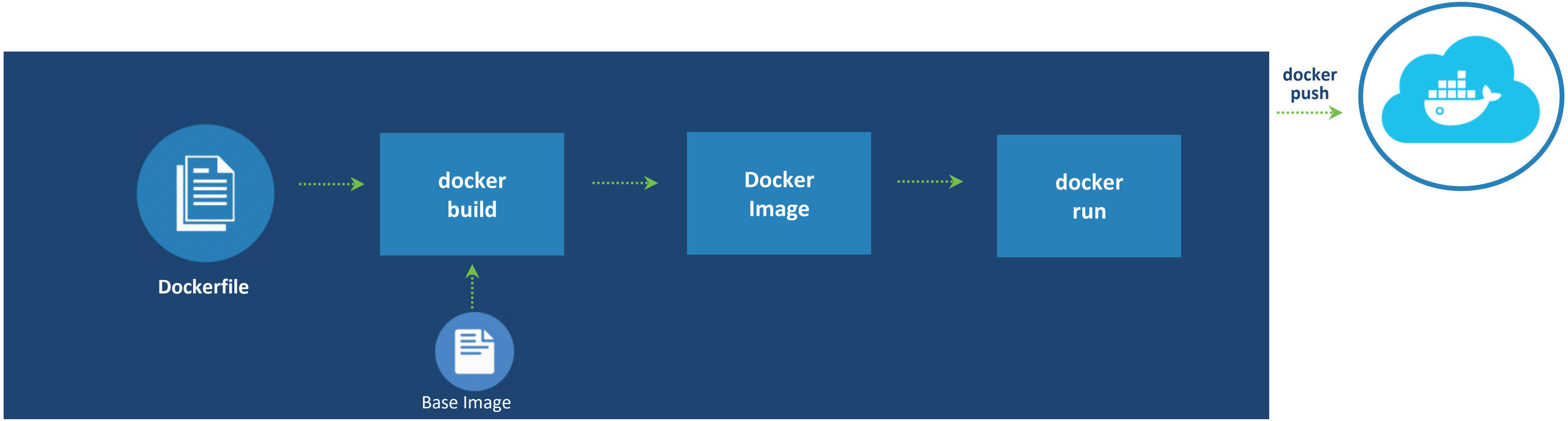


Data Infrastructure

ML/DL Framework / Infrastructure

Inferencing & Ingestion End Point

Docker – Build, run and upload process



Dockerfile

```
FROM     python:latest

LABEL    maintainer="Paniraj Koppa <pkoppa@cisco.com>" \
         description="Docker for machine learning demonstration Cisco Live!”

RUN      pip --no-cache-dir install \
             pandas==0.24.2 \
             jupyter \
             seaborn==0.9.0 \
             matplotlib==3.0.3 \
             missingno==0.4.1 \
             numpy==1.16.3 \
             sklearn

WORKDIR /ml_space

EXPOSE  8888

COPY     titanic.ipynb titanic.csv /ml_space

CMD      ["jupyter", "notebook", "--ip='0.0.0.0'", "--port=8888", "--no-browser", "--allow-root"]
```

Summary of Commands

Format of the command

$ docker <management_commands> <commands> <option> <image_name>

Samples

```
$ docker container run -d --rm -p 4321:8888 --name my_trial ml_trials
$ docker image ls
$ docker image inspect ml_trials
$ docker container ls
$ docker container logs my_trial
$ docker container inspect my_trial
$ docker container exec -it my_trial bash
```

Sharing your research

Step 1: You build the image

$ docker image build -t ml_trials .

Step 2: You "push" it to Docker Hub

$ docker login

$ docker image tag ml_trials pkoppa/ml_trials

$ docker image push pkoppa/ml_trials

Step 3: Others will "pull" from Docker Hub

$ docker image pull pkoppa/ml_trials

Step 4: Running a container

docker container run -d --rm -p 4321:8888 --name my_trial ml_trials

Deploying Machine Learning Models

Demo - 2

ML @ edge!

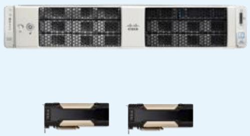
GitHub Repo:
https://github.com/pkoppa/docker_for_ml

Docker Containers with GPU support

Cisco AI/ML Compute Portfolio – Addressing All Aspect

Test & Dev and Model Training

C240



2 x V100

HyperFlex 240

2 x V100 Per Node

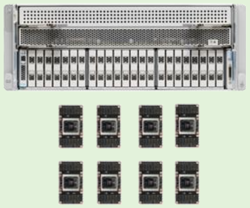
Option of GPU Only Nodes

Deep Learning/ Training

C480

6 x PCIe V100

C480 ML



8 x V100 with NVLink

Inferencing

C/HX 220
C/HX 240

2 x T4
6 x T4

Unified Management

CISCO INTERSIGHT

CISCO UCS Manager

Cisco IMC

XML API

python SDK

Simplified Management, Customer Choice, Cisco Validated Design

Cisco Portfolio Alignment With Deployment Lifecycle

Use Case Conception → Feasibility Study → Model Design → Model Deployment

Use Case Conception

cloudera HORTONWORKS hadoop APACHE Spark

C240 M5

Optimized Big Data Solutions for Data Collection and Preparation

Feasibility Study

NVIDIA V100

C/HX-240

Quick Installation of Hardware and Software for AI Exploration and Experimentation

Model Design

NVIDIA V100 NVLink

C480 ML

Offload and Accelerate AI Training at Scale with Performance Optimized Systems

Model Deployment

NVIDIA T4

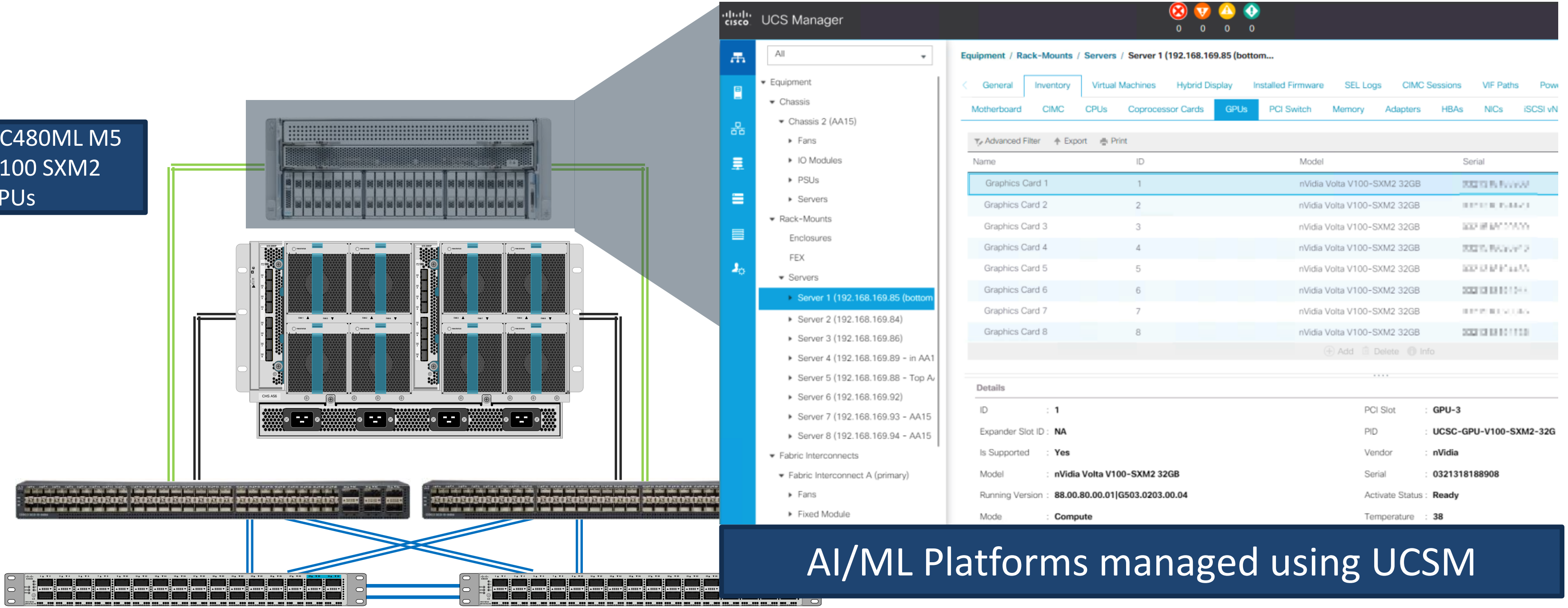
intel OpenVINO™

C/HX- 220/240

Distribute and Scale AI Inference from the Data Center to the Edge

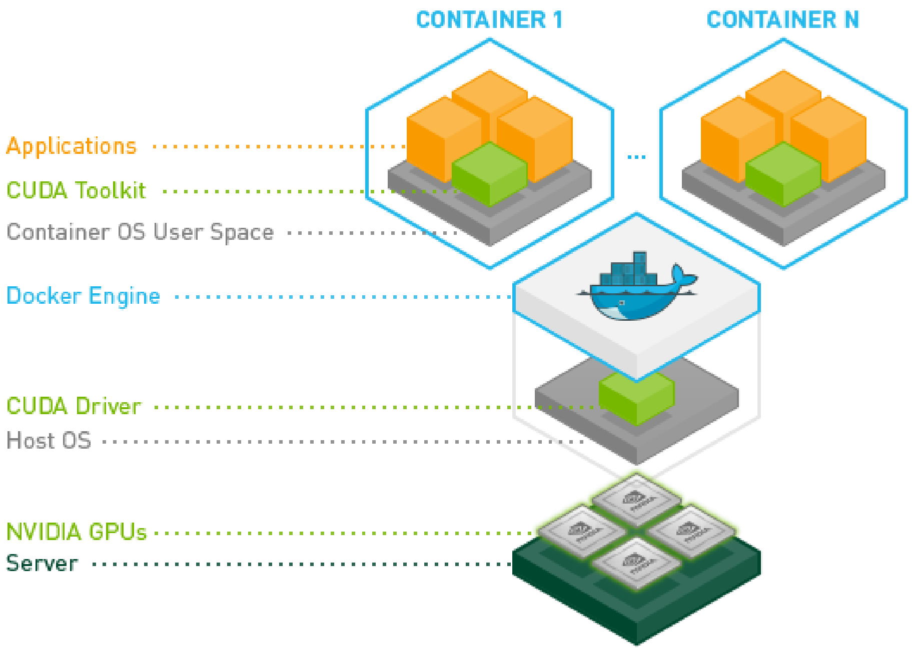
Unified Management for Cisco UCS Platforms

Cisco UCS C480ML M5 with 8 V100 SXM2 GPUs



AI/ML Platforms managed using UCSM

AI/ML – Software and Workload



NVIDIA GPU Cloud (NGC)	NVIDIA GPU Cloud (NGC) provides containerized versions of deep learning frameworks
NVIDIA-Docker	NVIDIA designed NVIDIA-Docker to enable portability in Docker images that leverage NVIDIA GPUs.
NVIDIA CUDA Toolkit	Using CUDA, developers can dramatically speed up applications by harnessing the power of GPUs
RHEL 7.6	Docker installed on RHEL 7.6 running on GPU equipped UCS C-Series servers

NGC Containers for AI/ML

- Eliminate time consuming complex builds and simply pull and run the NVIDIA GPU enabled AI/ML frameworks
- Support multi-GPU and multi-Node systems for scale up and scale out environments
- Support both Bare-Metal (BM) deployment and vSphere environments
- Flexible customer deployment options:
  - For maximum performance, deploy BM servers
  - For flexible GPU configuration, such as fractional GPUs, deploy in VMware environment

Demo 4
Running Tensorflow Container on NVIDIA Docker

Cisco Converged Infrastructure solutions for AI/ML

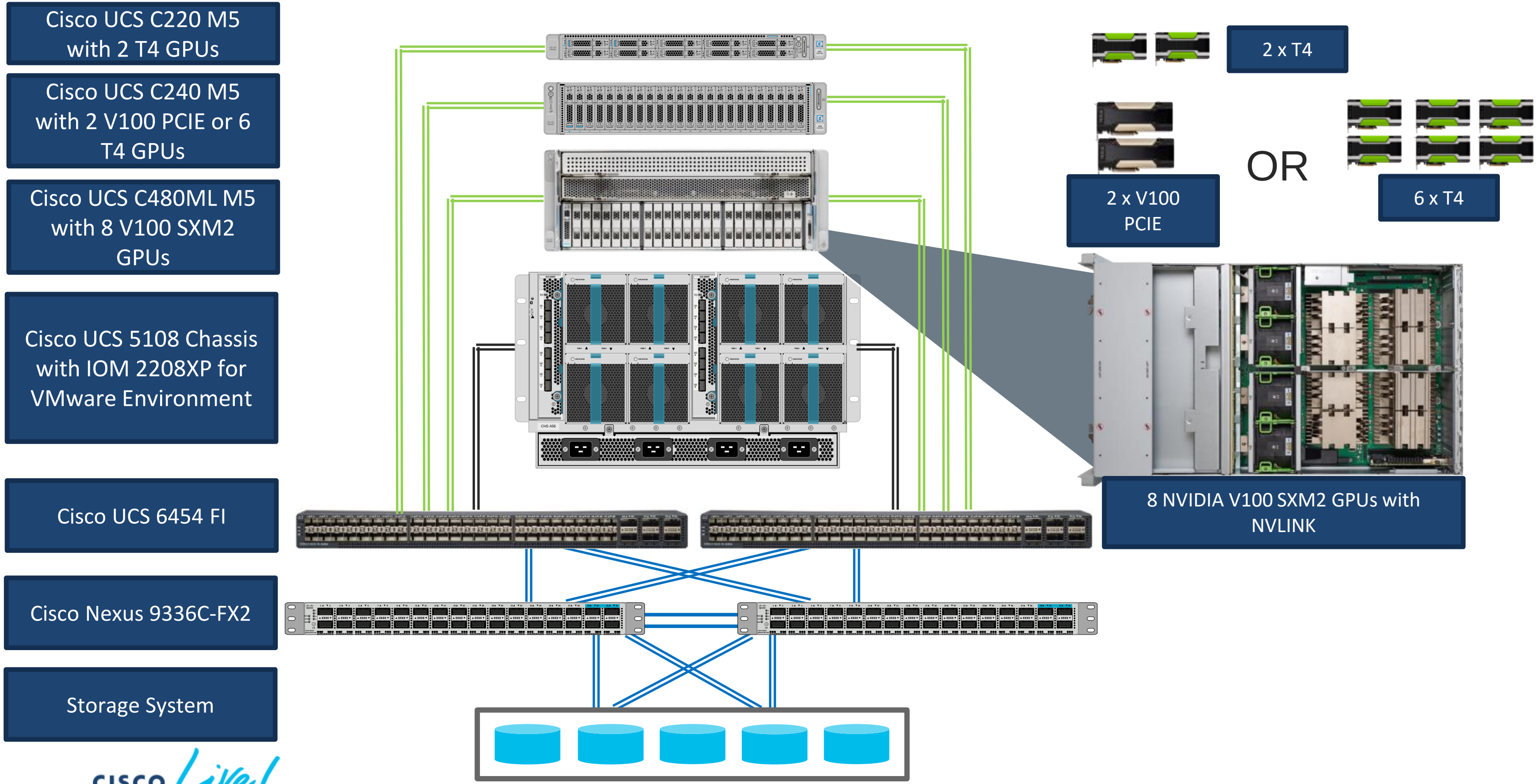
Cisco Converged Infrastructure for AI

Fast, efficient, easy and scalable

- Simplified Management: Extend your existing designs to seamlessly support AI/ML. Manage the AI/ML platforms like any other UCS Server
- Consistent operation and support model
- Repeatable building blocks to increase the scale of the environment, including GPUs, allowing you to start small and grow non-disruptively
- Easily deploy AI Frameworks with GPU support
- Close partnership with leading storage vendors to develop Cisco validated designs and solutions

mxnet TensorFlow Caffe2 theano PYTORCH

Cisco UCS Platforms for AI - Integration



NVIDIA GPUs for vComputeServer

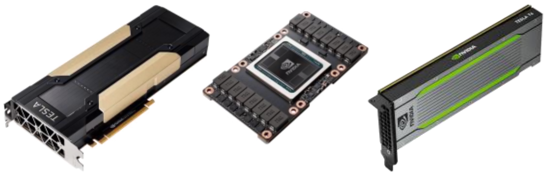
NVIDIA recommends **T4** and **V100** GPUs for vComputeServer deployments

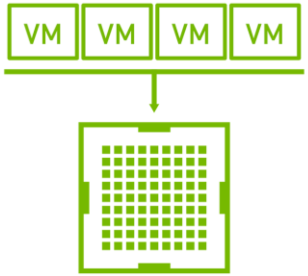
Fractional GPU: assign more than 1 VM to a GPU

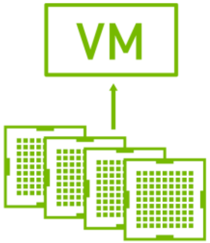
- Optimize GPU utilization
- Upto 8 VMs to a single GPU
- Minimum profile size of 4GB
- Maximum profile size of 32 GB

Aggregate GPUs: assign more than 1 GPU to a VM

- Scaling for higher performance
- Upto 4 vGPU to a VM (ESXi 6.7 U3)







Demo 5
NVIDIA vComputeServer
Fractional GPU support

Thank you

Possibilities

#CiscoLive