TensorFlow 2.15 Deployment Guide

Environment: Python 3.8 Server + CUDA 12.2 Driver

> **Environment Status:** - **System Python:** 3.8 (Incompatible with TF 2.15) - **CUDA Driver:** 12.2 (Compatible) - **Goal:** Deploy TensorFlow 2.15 (Requires Python 3.9-3.11)

---

1. Constraints & Prerequisites

Why we must not upgrade the system Python:

Risk	Impact
System stability	Core OS utilities depend on the specific system Python version
Permissions	Requires root access which is often restricted
Dependencies	Shared infrastructure may break for other users

Solution: Use user-space virtualization (Miniconda/Virtualenv) to install Python 3.11 independently.

---

2. Recommended: Miniconda Installation

This method requires **no root access** and creates an isolated environment.

Online Installation (Standard)

```
# 1. Download installer
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

# 2. Install to home directory
bash Miniconda3-latest-Linux-x86_64.sh -b -p $HOME/miniconda3

# 3. Initialize conda
$HOME/miniconda3/bin/conda init bash
source ~/.bashrc

# 4. Create Python 3.11 environment
conda create -n ml-vision python=3.11 -y
conda activate ml-vision
python --version  # Confirms Python 3.11.x
```

Offline / Air-Gapped Installation

If the server has no internet access, prepare files on a connected machine first.

A. Preparation (On Internet-Connected Machine)

```
# 1. Download Installer
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

# 2. Download Wheels
mkdir tf_offline && cd tf_offline
pip download tensorflow[and-cuda]==2.15.0 --dest .

# 3. Option: Pre-pack full environment (requires conda-pack)
# conda install -c conda-forge conda-pack
# conda-pack -n ml-vision -o ml-vision-portable.tar.gz
```

B. Transfer

Transfer `Miniconda3-latest-Linux-x86_64.sh` and the `tf_offline/` folder (or `.tar.gz` archive) to the server via **FileZilla (SFTP)**

C. Installation (On Server)

```
# Install Miniconda
bash ~/Miniconda3-latest-Linux-x86_64.sh -b -p $HOME/miniconda3
source ~/.bashrc

# Create Environment
conda create -n ml-vision python=3.11 -y
conda activate ml-vision

# Install Packages
pip install --no-index --find-links=~/tf_offline/ tensorflow
```

---

3. GPU Configuration

Verification: Run `nvidia-smi` to confirm the driver version is **≥ 525.60.13** (required for CUDA 12). Your system shows **12.2**, which is compatible.

Installation Check

TensorFlow 2.15 will automatically use the bundled CUDA libraries if installed via `pip install` `tensorflow[and-cuda]`.

```
import tensorflow as tf

print(f"TF Version: {tf.__version__}")

print(f"GPU: {tf.config.list_physical_devices('GPU')}")
```

Troubleshooting: If `libcudart.so` errors occur, verify `LD_LIBRARY_PATH` includes the conda environment's lib directory:

```
export LD_LIBRARY_PATH=$CONDA_PREFIX/lib:$LD_LIBRARY_PATH
```

---

4. Alternative: Container Deployment

If conda is not an option, use a container runtime.

Docker

```
docker run --gpus all -it -v <PROJECT_DIR>:/workspace tensorflow/tensorflow:2.15.0-gpu bash
```

Apptainer / Singularity

Common in research environments without root Docker access.

```
apptainer build tf215.sif docker://tensorflow/tensorflow:2.15.0-gpu

apptainer exec --nv tf215.sif python <YOUR_SCRIPT.py>
```

---

5. Escalation: Technical Request Specifications

If user-space solutions are blocked, providing these exact technical details to IT will speed up the process.

Objective: Install Python 3.11 alongside system Python 3.8 (Non-destructive "Altinstall").

Technical Requirements:

- **Target Version:** Python 3.11.x
- **Installation Path:** `/usr/local/bin/python3.11` (or `/opt/python3.11`)
- **Method:** Source compilation with `make altinstall` to prevent overwriting `/usr/bin/python3`.

Safe Installation Command Sequence for IT:

```
# 1. Download & Extract

wget https://www.python.org/ftp/python/3.11.7/Python-3.11.7.tgz

tar xzf Python-3.11.7.tgz && cd Python-3.11.7


# 2. Configure (Enable optimizations for ML performance)

./configure --enable-optimizations --prefix=/usr/local
```

```
# 3. Build & Alt-Install
make -j$(nproc)
sudo make altinstall  # CRITICAL: Use 'altinstall', not 'install'
```

Why this is safe: 1. **No Symlink Changes:** Does NOT touch the `python3` command or system links. 2. **Isolated Binary:** Only accessible via explicit `python3.11` command. 3. **No Yum/Apt Conflicts:** Completely independent of the system package manager.

Quick Reference

Action	**Command**
New Env	`conda create -n ml-vision python=3.11`
Activate	`conda activate ml-vision`
GPU Check	`nvidia-smi`
TF Install	`pip install tensorflow[and-cuda]==2.15.0`