

Anarion Technologies – Apache TVM

Note: Search product on Azure marketplace and click on "Get it now"

 Microsoft | Marketplace

Products ▾ 🔍 Search Marketplace

☰ All Products ▾ Categories ▾ Industries ▾ Partners ▾

Products > Apache TVM on Ubuntu26



Apache TVM on Ubuntu26

by Anarion Technologies

Virtual Machines

Starting at \$0.03 [Get it now](#) ☆ Save to my list

[Overview](#) [Plans + Pricing](#) [Ratings + reviews](#) [Details + support](#)

Production Ready VM on Ubuntu26.04 LTS + Free Support

Apache TVM is a high-performance open-source machine learning compiler stack designed to optimize, compile, and deploy AI models across a wide range of hardware platforms. It enables developers to improve the performance of deep learning workloads while reducing deployment complexity for production environments.

This virtual machine comes pre-installed with Apache TVM 0.23.0 on Ubuntu 26.04 LTS, providing a ready-to-use environment for AI engineers, machine learning developers, researchers, and enterprises. The image includes Python 3.14, LLVM support, essential build tools, and all required dependencies, allowing users to begin developing and optimizing machine learning applications immediately without manual installation.

Apache TVM supports model compilation and optimization for popular machine learning frameworks such as TensorFlow, PyTorch, ONNX, and Apache MXNet. It provides automatic operator tuning, graph optimization, code generation, and hardware-specific optimizations to maximize inference performance across CPUs, GPUs, and other supported

Click on Continue

Confirm your details to continue



Apache TVM on Ubuntu26

By Anarion Technologies

Plan

Apache TVM on Ubuntu26



Pricing

Starting at \$0.03/hour

Description

Apache TVM on Ubuntu26

You're signed in as **Market Place Products (mp@anariontech.com)**.

[Edit your details](#)

ⓘ By clicking **Get it now**, I give Microsoft permission to use or share my account information so that the provider can contact me regarding this product. I agree to the provider's [terms of use](#) and [privacy policy](#) and understand that the rights to use this product do not come from Microsoft, unless Microsoft is the provider. Use of Marketplace is governed by separate [terms](#) and [privacy](#).

Get it now

Click on Create

Home

Apache TVM on Ubuntu26

Anarion Technologies



Apache TVM on Ubuntu26

 Add to Favorites

Anarion Technologies | Virtual Machine

Subscription

Azure subscription 1

Plan

Apache TVM on Ubuntu26

Create

Start with a pre-set configuration

Want to deploy programmatically? [Get started](#)

[Overview](#) [Plans + Pricing](#) [Usage Information + Support](#) [Ratings + Reviews](#)

Apache TVM is a high-performance open-source machine learning compiler stack designed to optimize, compile, and deploy AI models across a wide range of hardware platforms. It enables developers to improve the performance of deep learning workloads while reducing deployment complexity for production environments.

This virtual machine comes pre-installed with Apache TVM 0.23.0 on Ubuntu 26.04 LTS, providing a ready-to-use environment for AI engineers, machine learning developers, researchers, and enterprises. The image includes Python 3.14, LLVM support, essential build tools, and all required dependencies, allowing users to begin developing and optimizing machine learning applications immediately without manual installation.

Apache TVM supports model compilation and optimization for popular machine learning frameworks such as TensorFlow, PyTorch, ONNX, and Apache MXNet. It provides automatic operator tuning, graph optimization, code generation, and hardware-specific optimizations to maximize inference performance across CPUs, GPUs, and other supported accelerators.

This image is configured with Ubuntu 26.04 LTS and the latest available security updates, providing a secure, reliable, and production-ready platform for AI development, research, model optimization, and deployment. Developers can integrate TVM into existing machine learning pipelines while benefiting from its flexible compiler architecture and extensive hardware support.

Creating a virtual machine, enter or select appropriate values for zone, machine type, resource group and so on as per your choice.

» Create a virtual machine

Basics

Disks

Networking

Management

Advanced

Tags

Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Azure subscription 1

Resource group * ⓘ

Demo

Create new

Instance details

Virtual machine name * ⓘ

Demo

Region * ⓘ

(US) East US

Review + create

< Previous

Next : Disks >

Create a virtual machine ...

Basics Disks Networking Management Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

Disk options

OS disk type * ⓘ Premium SSD (locally-redundant storage) ▼

Encryption type * (Default) Encryption at-rest with a platform-managed key ▼

Enable Ultra Disk compatibility ⓘ ☐
Ultra disk is available only for Availability Zones in eastus.

Data disks

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

i Adding unmanaged data disks is currently not supported at the time of VM creation. You can add them after the VM is

Review + create

< Previous

Next : Networking >

Create a virtual machine ...

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * ⓘ (new) Demo-vnet ▼
[Create new](#)

Subnet * ⓘ (new) default (10.1.0.0/24) ▼

Public IP ⓘ (new) Demo-ip ▼
[Create new](#)

NIC network security group ⓘ ☐ None
☒ Basic

Review + create

< Previous

Next : Management >

Create a virtual machine ...

Basics Disks Networking Management Advanced Tags Review + create

Configure monitoring and management options for your VM.

Azure Security Center

Azure Security Center provides unified security management and advanced threat protection across hybrid cloud workloads. [Learn more](#)

✓ Your subscription is protected by Azure Security Center basic plan.

Monitoring

Boot diagnostics ⓘ ☐ On
☒ Off

Enable OS guest diagnostics ⓘ ☐

Identity

[Review + create](#)

[< Previous](#)

[Next : Advanced >](#)

Create a virtual machine ...

Basics Disks Networking Management Advanced Tags Review + create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. [Learn more about tags](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name ⓘ	Value ⓘ	Resource
	:	12 selected v

[Review + create](#)

[< Previous](#)

[Next : Review + create >](#)

[Home](#) >

Create a virtual machine ...

✓ Validation passed

Basics Disks Networking Management Monitoring Advanced Tags **Review + create**

i Cost given below is an estimate and not the final price. For all your pricing needs, please use the pricing calculator. [↗](#)

Price

1 X Standard B1s
by Microsoft
[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ

0.8652 INR/hr
[Pricing for other VM sizes](#)

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

< Previous

Next >

Create

After Process of Create Virtual Machine. You have got an Option Go to Resource Group

Click Go to Resource Group

Copy the Public IP Address

SSH into Terminal and Run these Commands:

```
$ sudo su
```

```
$ cd ../../
```

```
$ sudo apt update && sudo apt upgrade -y
```



```

testing@testing:~$ sudo su
root@testing:/home/testing# cd ../../
root@testing:/# sudo apt update && sudo apt upgrade -y
Hit:1 http://azure.archive.ubuntu.com/ubuntu resolute InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu resolute-updates InRelease [137
kB]
Hit:3 http://azure.archive.ubuntu.com/ubuntu resolute-backports InRelease
Get:4 http://azure.archive.ubuntu.com/ubuntu resolute-security InRelease [13
7 kB]
Get:5 https://aquasecurity.github.io/trivy-repo/deb resolute InRelease [3238
B]
Get:6 http://azure.archive.ubuntu.com/ubuntu resolute-updates/main amd64v3 P
ackages [282 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu resolute-updates/main Translati
on-en [74.0 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu resolute-updates/main amd64 Com
ponents [49.4 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu resolute-updates/main amd64 c-n
-f Metadata [5244 B]
Get:10 http://azure.archive.ubuntu.com/ubuntu resolute-updates/universe amd6
4v3 Packages [159 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu resolute-updates/universe Tran
slation-en [50.1 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu resolute-updates/universe amd6

```

Step 1: Activate the Python Virtual Environment

```
$ source /opt/tvm-env/bin/activate
```

Step 2: Verify Apache TVM Version

```
$ python -c "import tvm; print(tvm.__version__)"
```

```

root@testing:/# source /opt/tvm-env/bin/activate
(tvm-env) root@testing:/# python -c "import tvm; print(tvm.__version__)"
0.23.0
(tvm-env) root@testing:/#

```

Step 3: Verify TVM Runtime

```
$ python -c "import tvm.runtime; print('TVM Runtime Loaded Successfully')"
```

```

(tvm-env) root@testing:/#
(tvm-env) root@testing:/# python -c "import tvm.runtime; print('TVM Runtime
Loaded Successfully')"
TVM Runtime Loaded Successfully
(tvm-env) root@testing:/#

```

Step 4: Verify LLVM Support

```
$ python -c "import tvm; print(tvm.target.codegen.llvm_version_major())"
```

```
(tvm-env) root@testing:/#
(tvm-env) root@testing:/# python -c "import tvm; print(tvm.target.codegen.ll
vm_version_major())"
21
(tvm-env) root@testing:/#
```

Step 5: Verify TVM Build Libraries

\$ ls -lh /opt/tvm/build/libtvm*

```
(tvm-env) root@testing:/#
(tvm-env) root@testing:/# ls -lh /opt/tvm/build/libtvm*
-rwxr-xr-x 1 root root 66M Jun 29 09:07 /opt/tvm/build/libtvm.so
-rwxr-xr-x 1 root root 4.9M Jun 29 08:04 /opt/tvm/build/libtvm_runtime.so
(tvm-env) root@testing:/#
```

Step 6: Verify TVM Build Configuration

\$ python -c "import tvm; print(tvm.support.libinfo())"

This command displays the TVM build configuration and confirms that the runtime libraries are loaded successfully.

```
(tvm-env) root@testing:/#
(tvm-env) root@testing:/# python -c "import tvm; print(tvm.support.libinfo())"
{'BUILD_STATIC_RUNTIME': 'OFF', 'BUILD_DUMMY_LIBTVM': 'OFF', 'COMPILER_RT_PATH': '3rdparty/compiler-rt', 'CUDA_VERSION': 'NOT-FOUND', 'DMLC_PATH': '3rdparty
/dmlc-core/include', 'GIT_COMMIT_HASH': '75ce1b4d46853b28fa926a6e29aa715b2463de92', 'GIT_COMMIT_TIME': '2026-01-27 13:55:36 +0800', 'HIDE_PRIVATE_SYMBOLS':
'OFF', 'INDEX_DEFAULT_I64': 'ON', 'INSTALL_DEV': 'OFF', 'LLVM_VERSION': '21.1.8', 'MLIR_VERSION': 'NOT-FOUND', 'PICOJSON_PATH': '3rdparty/picojson', 'RANG_P
ATH': '3rdparty/rang/include', 'ROCM_PATH': '/opt/rocm', 'SUMMARIZE': 'OFF', 'TVM_CXX_COMPILER_PATH': '/usr/bin/c++', 'USE_ALTERNATIVE_LINKER': 'AUTO', 'USE
_ARM_COMPUTE_LIB_GRAPH_EXECUTOR': 'OFF', 'USE_ARM_COMPUTE_LIB': 'OFF', 'USE_BLAS': 'none', 'USE_BNNS': 'OFF', 'USE_BYODT_POSIT': 'OFF', 'USE_COREML': 'OFF',
'USE_CPP_RPC': 'OFF', 'USE_CPP_RTVM': 'OFF', 'USE_CUBLAS': 'OFF', 'USE_CUDA': 'OFF', 'USE_NVTX': 'OFF', 'USE_NCCL': 'OFF', 'USE_MSCCL': 'OFF', 'USE_CUDNN':
'OFF', 'USE_CUSTOM_LOGGING': 'OFF', 'USE_CUTLASS': 'OFF', 'USE_AMX': 'OFF', 'USE_DNNL': 'OFF', 'USE_FALLBACK_STL_MAP': 'OFF', 'USE_GTEST': 'AUTO', 'USE_HEX
AGON': 'OFF', 'USE_HEXAGON_RPC': 'OFF', 'USE_HEXAGON_SDK': '/path/to/sdk', 'USE_HEXAGON_GTEST': '/path/to/hexagon/gtest', 'USE_HEXAGON_EXTERNAL_LIBS': 'OFF',
'USE_IOS_RPC': 'OFF', 'USE_KHRONOS_SPIRV': 'OFF', 'USE_LIBBACKTRACE': 'AUTO', 'USE_LIBTORCH': 'OFF', 'USE_LLVM': 'llvm-config', 'USE_MLIR': 'OFF', 'USE_ME
TAL': 'OFF', 'USE_MIOPEN': 'OFF', 'USE_MKL': 'OFF', 'USE_MRVL': 'OFF', 'USE_MSVC_MT': 'OFF', 'USE_NNPACK': 'OFF', 'USE_OPENCL': 'OFF', 'USE_OPENCL_ENABLE_HO
ST_PTR': 'OFF', 'USE_OPENCL_EXTN_QCOM': 'NOT-FOUND', 'USE_OPENCL_GTEST': '/path/to/opencl/gtest', 'USE_OPENMP': 'none', 'USE_PAPI': 'OFF', 'USE_RANDOM': 'ON
', 'TVM_DEBUG_WITH_ABI_CHANGE': 'OFF', 'TVM_LOG_BEFORE_THROW': 'OFF', 'USE_ROCBLAS': 'OFF', 'USE_HIPBLAS': 'OFF', 'USE_ROCM': 'OFF', 'USE_RCCL': 'OFF', 'USE
_RPC': 'ON', 'TVM_BUILD_PYTHON_MODULE': 'OFF', 'USE_RTTI': 'ON', 'USE_RUST_EXT': 'OFF', 'USE_SORT': 'ON', 'USE_SPIRV_KHR_INTEGER_DOT_PRODUCT': 'OFF', 'USE_T
ENSORFLOW_PATH': 'none', 'USE_TENSORRT_CODEGEN': 'OFF', 'USE_TENSORRT_RUNTIME': 'OFF', 'USE_TFLITE': 'OFF', 'USE_THREADS': 'ON', 'USE_THRUST': 'OFF', 'USE_C
URAND': 'OFF', 'USE_VULKAN': 'OFF', 'USE_CLML': 'OFF', 'TVM_CLML_VERSION': '', 'USE_CLML_GRAPH_EXECUTOR': 'OFF', 'USE_UMA': 'OFF', 'USE_MSC': 'OFF', 'USE_CC
ACHE': 'AUTO', 'USE_NVSHMEM': 'OFF', 'USE_NNAPI_CODEGEN': 'OFF', 'USE_NNAPI_RUNTIME': 'OFF', 'BACKTRACE_ON_SEGFAULT': 'OFF'}
(tvm-env) root@testing:/#
(tvm-env) root@testing:/#
```

Step 7: Verify CPU Target

\$ python -c "import tvm; print(tvm.cpu())"

```
(tvm-env) root@testing:/#
(tvm-env) root@testing:/#
(tvm-env) root@testing:/# python -c "import tvm; print(tvm.cpu())"
cpu:0
(tvm-env) root@testing:/#
(tvm-env) root@testing:/#
```

Step 8: Verify Installation Using Python

\$python

Run the following commands:


```
>>> import tvm
```

```
>>> print("TVM Version:", tvm.__version__)
```

```
>>> print("CPU Device:", tvm.cpu())
```

```
(tvm-env) root@testing:/#  
(tvm-env) root@testing:/# python  
Python 3.14.4 (main, Apr 8 2026, 04:02:31) [GCC 15.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import tvm  
>>> print("TVM Version:", tvm.__version__)  
TVM Version: 0.23.0  
>>> print("CPU Device:", tvm.cpu())  
CPU Device: cpu:0  
>>> quit  
(tvm-env) root@testing:/# ls
```

Thanks!!!