

# **The Knowledgebase of Interatomic Models (OpenKIM)**

**&**

# **The Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)**

**Ryan S. Elliott**

Department of Aerospace Engineering & Mechanics  
The University of Minnesota

Ellad B. Tadmor (U. Minn), James P. Sethna (Cornell U.)

# OUTLINE

1. The Knowledgebase of Interatomic Models
  - User Signup
  - KIM API
  - “KIM Compliant” Simulators
  - Setup VirtualBox Ubuntu VM
  - KIM Model performance within LAMMPS
  - KIM Model pages
2. Installing the KIM API on Ubuntu (ppa:openkim/openkim)
3. Using LAMMPS with the KIM API

Appendix: Installing the KIM API by hand

# OUTLINE

1. The Knowledgebase of Interatomic Models
  - User Signup
  - KIM API
  - “KIM Compliant” Simulators
  - Setup VirtualBox Ubuntu VM
  - KIM Model performance within LAMMPS
  - KIM Model pages
2. Installing the KIM API on Ubuntu (ppa:openkim/openkim)
3. Using LAMMPS with the KIM API

Appendix: Installing the KIM API by hand

# OpenKIM with LAMMPS

OpenKIM

+ -

Member Login

?



## Welcome to the Knowledgebase of Interatomic Models!

OpenKIM is an online suite of open source tools for molecular simulation of materials. These tools help to make molecular simulation more accessible and more reliable. Within OpenKIM, you will find an online resource for standardized testing and long-term warehousing of interatomic models and data, and an *application programming interface* (API) standard for coupling atomistic simulation codes and interatomic potential subroutines.



### Models

**How do atoms interact?** KIM Models (interatomic potentials and force fields) are software packages for describing atomic interactions that can be used with a variety of simulation codes, including LAMMPS, DL\_POLY, IMD, ASE and GULP, that are compatible with the KIM API standard.

Get interatomic potentials



### Properties

**What do the models predict?** KIM Properties are standardized definitions for material properties uploaded by the materials research community. Predictions of KIM Models for these properties are stored in the OpenKIM Repository and can be viewed, visualized, and compared with first principles and experimental reference data.

Get model predictions



### Tests

**How are properties computed?** KIM Tests are robust, standardized calculations (stand-alone computer code or input files to supported simulators), uploaded by the materials research community, that couple with KIM Models to make predictions for well-defined material properties.

Get property simulators



### Participate

**What can I do?** KIM is an international standards organization for molecular simulations. We invite you to learn more, join as a member, contribute content (Models, Tests, Reference Data and Visualizers), help define standard material properties, contribute to API code development, and help document!

Get involved

# OpenKIM with LAMMPS

OpenKIM

Ryan



## Welcome to the Knowledgebase of Interatomic Models!

OpenKIM is an online suite of open source tools for molecular simulation of materials. These tools help to make molecular simulation more accessible and more reliable. Within OpenKIM, you will find an online resource for standardized testing and long-term warehousing of interatomic models and data, and an *application programming interface* (API) standard for coupling atomistic simulation codes and interatomic potential subroutines.



### Models

**How do atoms interact?** KIM Models (interatomic potentials and force fields) are software packages for describing atomic interactions that can be used with a variety of simulation codes, including LAMMPS, DL\_POLY, IMD, ASE and GULP, that are compatible with the KIM API standard.

Get interatomic potentials



### Properties

**What do the models predict?** KIM Properties are standardized definitions for material properties uploaded by the materials research community. Predictions of KIM Models for these properties are stored in the OpenKIM Repository and can be viewed, visualized, and compared with first principles and experimental reference data.

Get model predictions



### Tests

**How are properties computed?** KIM Tests are robust, standardized calculations (stand-alone computer code or input files to supported simulators), uploaded by the materials research community, that couple with KIM Models to make predictions for well-defined material properties.

Get property simulators




### Participate

**What can I do?** KIM is an international standards organization for molecular simulations. We invite you to learn more, join as a member, contribute content (Models, Tests, Reference Data and Visualizers), help define standard material properties, contribute to API code development, and help document!






Get involved




# OpenKIM with LAMMPS

 OpenKIM

Ryan






## Welcome to the Knowledgebase of Interatomic Models

OpenKIM is an online suite of open source tools for molecular simulation. It helps to make molecular simulation more accessible and more reliable. Within OpenKIM, you can find standardized testing and long-term warehousing of interatomic models. OpenKIM provides a *programming interface* (API) standard for coupling atomistic simulation codes to subroutines.


- KIM Items
- Query
- KIM API**
- Guide to KIM IDs
- EDN Format
- kimspec.edn Reference
- KIM Properties Framework
- OpenKIM Virtual Machine (VM)
- Pipeline Queue



### Models

**How do atoms interact?** KIM Models (interatomic potentials and force fields) are software packages for describing atomic interactions that can be used with a variety of simulation codes, including LAMMPS, DL\_POLY, IMD, ASE and GULP, that are compatible with the KIM API standard.


**Get interatomic potentials**



### Properties

**What do the models predict?** KIM Properties are standardized definitions for material properties uploaded by the materials research community. Predictions of KIM Models for these properties are stored in the OpenKIM Repository and can be viewed, visualized, and compared with first principles and experimental reference data.


**Get model predictions**



### Tests

**How are properties computed?** KIM Tests are robust, standardized calculations (stand-alone computer code or input files to supported simulators), uploaded by the materials research community, that couple with KIM Models to make predictions for well-defined material properties.

**Get property simulators**



### Participate

**What can I do?** KIM is an international standards organization for molecular simulations. We invite you to learn more, join as a member, contribute content (Models, Tests, Reference Data and Visualizers), help define standard material properties, contribute to API code development, and help document!

**Get involved**

## KIM Application Programming Interface (API)

### Overview

The KIM API is an Application Programming Interface for atomistic simulations. The API provides a standard for exchanging information between atomistic simulation codes (molecular dynamics, molecular statics, lattice dynamics, Monte Carlo, etc.) and interatomic models (potentials or force fields). It also includes a set of library routines for using the API with bindings for

- FORTRAN 77
- Fortran 90/95
- Fortran 2003
- C
- C++

By conforming to this API, an atomistic simulation code will seamlessly work with any KIM-compliant interatomic model written in any supported language. The interface is computationally efficient and often requires relatively minor changes to existing codes.

NOTE: It has been determined that **GCC 4.6.x** (and in particular, gfortran 4.6.x) has a number of insidious bugs associated with the use of Cray pointers *AND* the iso\_c\_binding interface. These bugs cause KIM Models to produce incorrect numerical results without generating any warnings or crashes. **Thus, gfortran 4.6.x SHOULD NOT BE USED with the openkim-api software package.**

### Downloading the Most Recent API Package

The most recent packaged download of the KIM API is available here:

[kim-api-latest](#)

# OpenKIM with LAMMPS

<http://groups.google.com/group/openkim>

Members of the OpenKIM development team actively monitor this forum and will do their best to respond to questions in a timely fashion. This forum is also used to announce minor new releases and bug fixes.

It is highly recommended for users who plan to work with the KIM API to become members of the openkim group. (Just go to the above link and click on "Join this group" on the right of the screen.)

## About

- About KIM
- Getting Started with KIM
- What People Are Saying
- Funding
- Citing the KIM Project
- Guide to KIM IDs
- The OpenKIM Brand Guide
- Licensing
- Team Members
- Governance

## KIM API

- Overview
- Download
- OS Setup for the KIM API
- Boot Camp
- Help
- Features

## Resources

- Frequently Asked Questions
- Software and Projects using KIM
- KIM Publications
- News Archive
- Reading List
- Requirements Document

## OpenKIM Repository

- KIM Items
- Visualizations
- Pipeline
- Pipeline — Query
- kimspec.edn Content Guide
- Properties
- KIM Properties Framework
- EDN Format

## Other

- Forums
- Contact Us



The OpenKIM project is supported by NSF funding



The OpenKIM project collaborates with the NIST



## OS Setup for the KIM API - Overview

Overview

Ubuntu

Macintosh

Windows

### Description

These pages provide simple steps to setup the Ubuntu, Mac, and Window Operating Systems (OS's) for use with the KIM API. In addition, instructions for installing these OS's as Virtual Machines (VMs) within VirtualBox (VB) are provided.

If you simply want to make your OS ready for use with the KIM API, click on the appropriate OS tab above and follow the instructions.

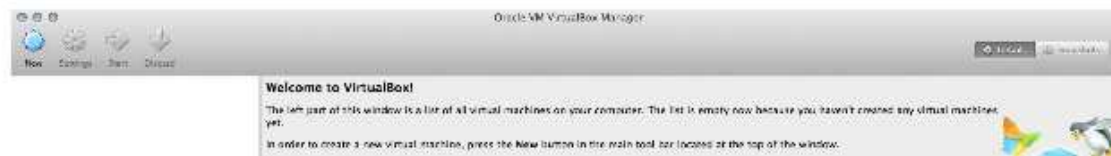
If you want to create a VM with your favorite OS, you will first need to install VirtualBox. See below.

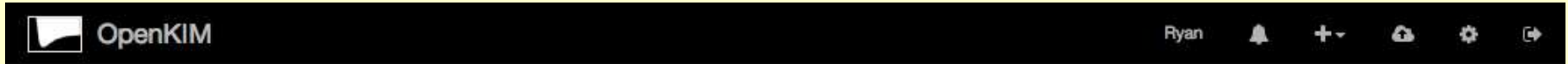
### VirtualBox Setup

VirtualBox (VB) is a great tool that allows virtualization. That is, running another OS on the host machine in a window on the host desktop. To download and to find out more about VirtualBox click [here](#).

### Starting up VB

Once you have downloaded and installed VB, start it up. It will open up screen that looks like this:





## OS Setup for the KIM API - Ubuntu

Overview

Ubuntu

Macintosh

Windows

### Contents:

- [Ubuntu disk image](#)
- [Ubuntu Virtual Machine](#)
- [Installing the Ubuntu OS](#)
- [Ubuntu KIM API Configuration file](#)

### Downloading the disk image

To set up the Ubuntu machine you first need a Ubuntu install disk image. This can be downloaded from [here](#). Choose which version you want (64 bit, 32 bit or 64 bit Mac; 64 bit is recommended for the KIM API). Then click Download.


### Creating the Ubuntu Virtual Machine (VM)

If you want to create a Ubuntu VM follow [these instructions](#).







### Installing the Ubuntu OS

Using the disk image, boot your machine. You should see this screen:



 OpenKIM

Ryan



## Software and Projects using KIM

### Software

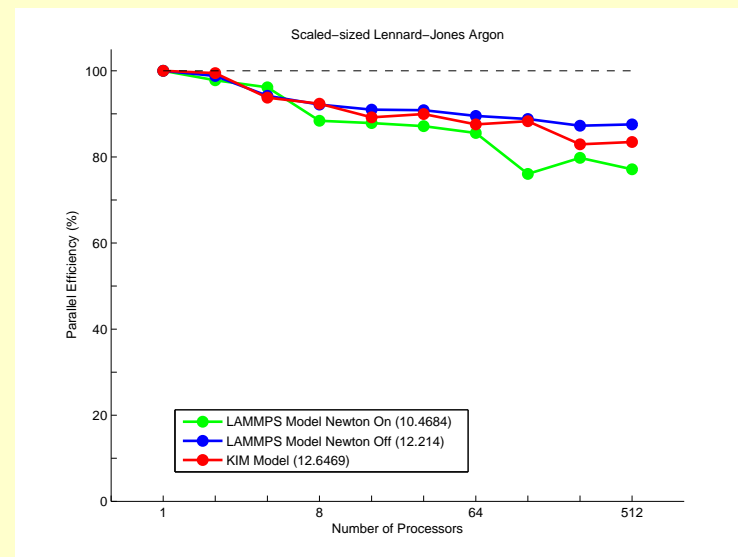
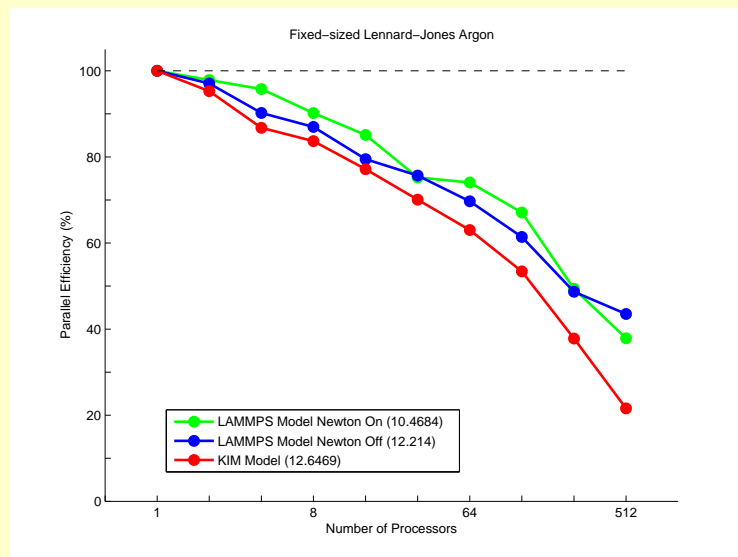
- **LAMMPS** – The molecular dynamics program LAMMPS has full support for the KIM API. This support is implemented in the [pair-kim](#) package which is integrated directly into the LAMMPS package. The KIM API must first be installed. Then, LAMMPS has to be compiled with the "make yes-kim" directive (see instructions [here](#))  
  
To run LAMMPS with a KIM model, download and build/install the Model Driver and/or Model archives (you can find these under [KIM Items](#)) by following the instructions provided in the KIM API package's INSTALL file. Then run LAMMPS as usual using "[pair\\_style kim](#)" and providing the name of the KIM model in the LAMMPS input file.
- **ASAP** – A calculator for doing large-scale classical molecular dynamics within the Campos Atomic Simulation Environment (ASE). The ASAP manual can be found [here](#) and installation instructions [here](#).
- **ASE** – KIM currently maintains an unofficial interface to the Atomic Simulation Environment (ASE) through a Python module called "kimcalculator." This module implements a calculator class much like all of the other calculators in the standard release, although it calculates quantities using the KIM API. Follow the [instructions on how to install and use the KIM calculator](#).
- **IMD** – The molecular dynamics program IMD currently supports KIM in beta mode. (KIM Models run about 50% slower than native potentials.) See the [instructions on how to use KIM Models with IMD](#).
- **GULP** – The lattice dynamics and molecular dynamics program GULP will provide support for the use of KIM Models beginning with Version 4.2. To use KIM with GULP you must add the flag "-DKIM" to DEFS in getmachine so that the code that supports KIM is enabled during compilation. Models are then specified using the "kim\_model" option. At present only a single type of each element is supported when using KIM and so species types are ignored when being passed to KIM models.
- **libAtoms+QUIP** – The libAtoms+QUIP molecular dynamics package has implemented partial support for KIM and is currently working towards full KIM compliance.
- **DL\_POLY** – KIM support for the DL\_POLY molecular dynamics program is currently under development.
- **Virtual Fab** – The Virtual Fab simulation laboratory provides an interactive platform to construct, carry out, and analyze simulations pertaining to nanoscale devices, with an emphasis on semiconductors. Virtual Fab currently offers full support for the use of KIM Models along with a visualization interface available for plotting and comparing KIM Test Results for Models relevant to a given application. Moreover, there is development underway to autogenerate KIM Tests from simulations carried out in Virtual Fab.
- **Quasicontinuum (QC) Method** – The version of the multiscale QC method program that fully supports the KIM API is currently in testing. A new release is forthcoming.

### Projects

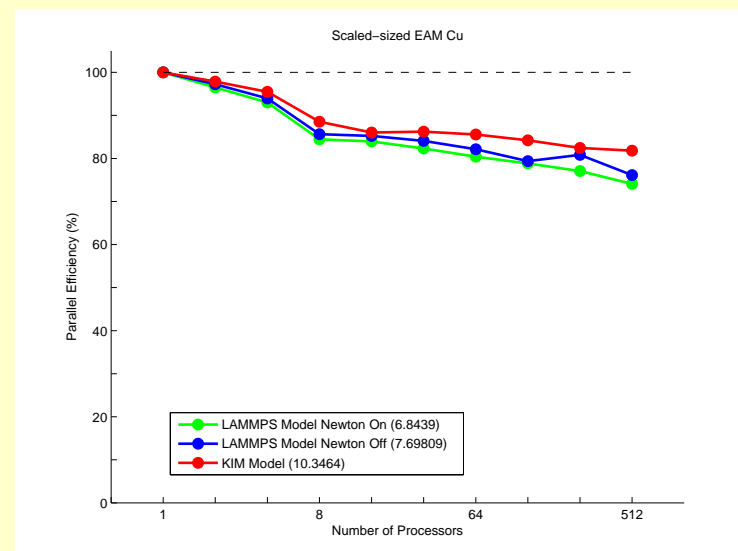
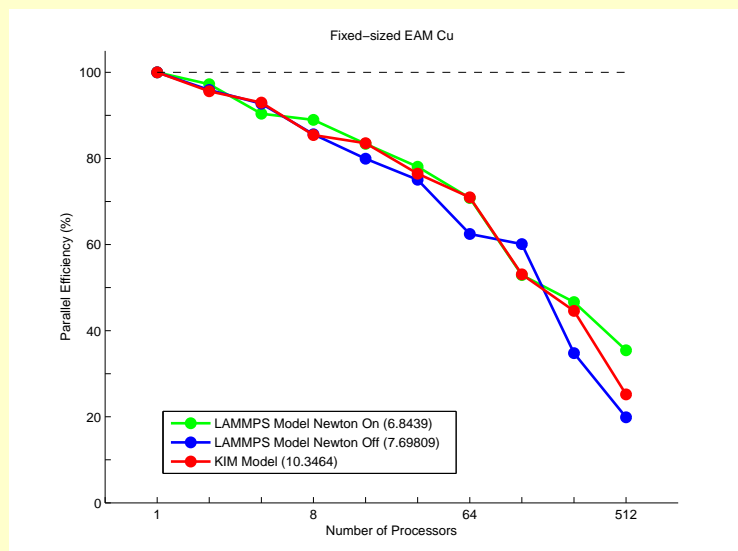
- KIM is collaborating with the [Computational Materials Repository \(CMR\)](#) project led by the Technical University of Denmark (DTU) on automatically importing first principles data into KIM and auto-generating Tests."
- KIM is collaborating with the National Institute of Standards and Technology (NIST) on developing reproducible [scientific workflows](#) for use with KIM Tests.

# OpenKIM with LAMMPS

LennardJones612\_Universal\_MO\_826355984548\_000



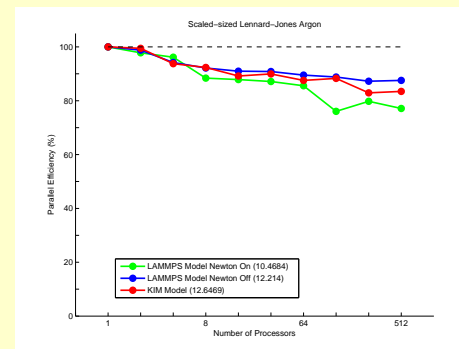
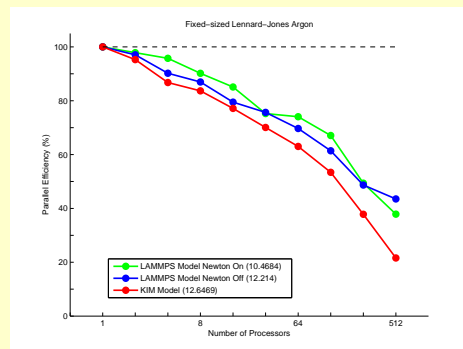
EAM\_Dynamo\_Foiles\_Baskes\_Universal3\_Cu\_MO\_666348409573\_000



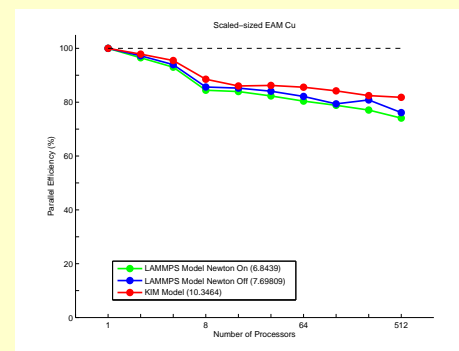
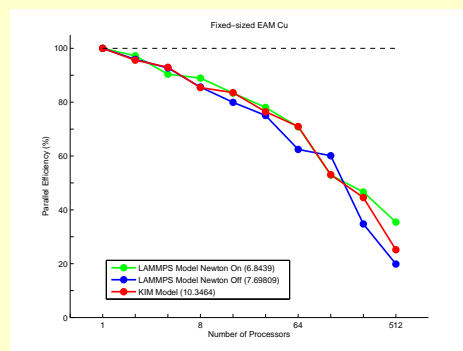


# OpenKIM with LAMMPS

LennardJones612\_Universal\_\_MO\_826355984548\_000



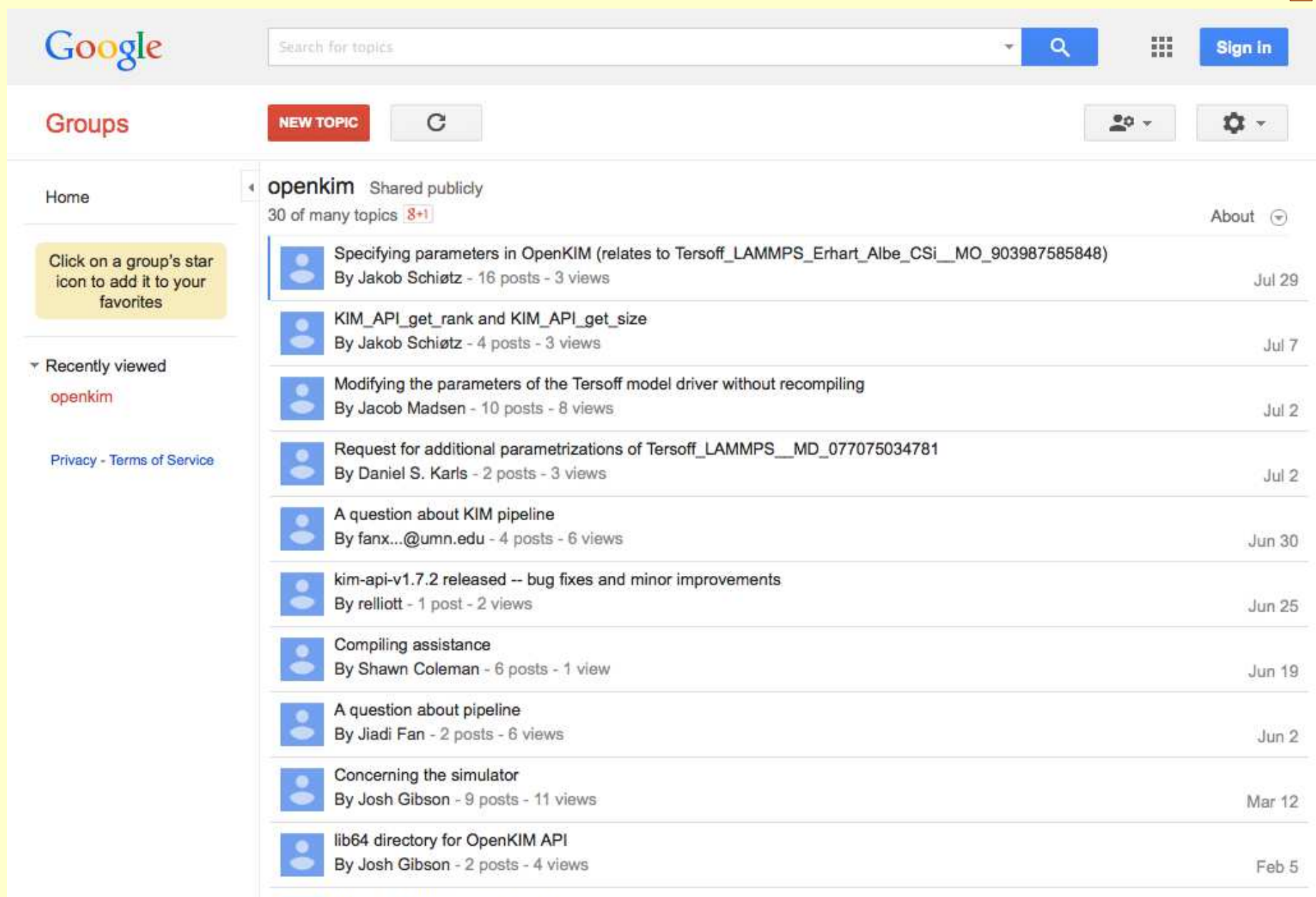
EAM\_Dynamo\_Foiles\_Baskes\_Universal3\_Cu\_\_MO\_666348409573\_000



Compiled using the Intel compiler suite (13.1.3) and -O3 optimization and executed on the Minnesota Supercomputing Institute's Itasca machine:

Itasca is an HP Linux cluster with 1,091 HP ProLiant BL280c G6 blade servers, each with two-socket, quad-core 2.8 GHz Intel Xeon X5560 "Nehalem EP" processors sharing 24 GB of system memory, with a 40-gigabit QDR InfiniBand (IB) interconnect. In total, Itasca consists of 8,728 compute cores and 24 TB of main memory.

# OpenKIM with LAMMPS



The screenshot shows the OpenKIM Google Group page. At the top is a Google search bar with the text "Search for topics". Below the search bar are navigation links: "Groups", "NEW TOPIC", and a refresh icon. On the left sidebar, there is a "Home" link, a yellow box with the text "Click on a group's star icon to add it to your favorites", a "Recently viewed" section showing "openkim", and links for "Privacy" and "Terms of Service". The main content area displays a list of topics under the "openkim" group, which is marked as "Shared publicly" and has "30 of many topics" (8+1). Each topic entry includes a user profile icon, the topic title, the author's name, the number of posts and views, and the date. The topics listed are:

- Specifying parameters in OpenKIM (relates to Tersoff\_LAMMPS\_Erhart\_Albe\_CSi\_MO\_903987585848) By Jakob Schiøtz - 16 posts - 3 views Jul 29
- KIM\_API\_get\_rank and KIM\_API\_get\_size By Jakob Schiøtz - 4 posts - 3 views Jul 7
- Modifying the parameters of the Tersoff model driver without recompiling By Jacob Madsen - 10 posts - 8 views Jul 2
- Request for additional parametrizations of Tersoff\_LAMMPS\_\_MD\_077075034781 By Daniel S. Karls - 2 posts - 3 views Jul 2
- A question about KIM pipeline By fanx...@umn.edu - 4 posts - 6 views Jun 30
- kim-api-v1.7.2 released -- bug fixes and minor improvements By relliott - 1 post - 2 views Jun 25
- Compiling assistance By Shawn Coleman - 6 posts - 1 view Jun 19
- A question about pipeline By Jiadi Fan - 2 posts - 6 views Jun 2
- Concerning the simulator By Josh Gibson - 9 posts - 11 views Mar 12
- lib64 directory for OpenKIM API By Josh Gibson - 2 posts - 4 views Feb 5

# OpenKIM with LAMMPS

OpenKIM

Ryan



## Welcome to the Knowledgebase of Interatomic Models!

OpenKIM is an online suite of open source tools for molecular simulation of materials. These tools help to make molecular simulation more accessible and more reliable. Within OpenKIM, you will find an online resource for standardized testing and long-term warehousing of interatomic models and data, and an *application programming interface* (API) standard for coupling atomistic simulation codes and interatomic potential subroutines.



### Models

**How do atoms interact?** KIM Models (interatomic potentials and force fields) are software packages for describing atomic interactions that can be used with a variety of simulation codes, including LAMMPS, DL\_POLY, IMD, ASE and GULP, that are compatible with the KIM API standard.

[Get interatomic potentials](#)



### Properties

**What do the models predict?** KIM Properties are standardized definitions for material properties uploaded by the materials research community. Predictions of KIM Models for these properties are stored in the OpenKIM Repository and can be viewed, visualized, and compared with first principles and experimental reference data.

[Get model predictions](#)



### Tests

**How are properties computed?** KIM Tests are robust, standardized calculations (stand-alone computer code or input files to supported simulators), uploaded by the materials research community, that couple with KIM Models to make predictions for well-defined material properties.

[Get property simulators](#)



### Participate

**What can I do?** KIM is an international standards organization for molecular simulations. We invite you to learn more, join as a member, contribute content (Models, Tests, Reference Data and Visualizers), help define standard material properties, contribute to API code development, and help document!

[Get involved](#)

# OpenKIM with LAMMPS

## KIM Models

Click on an element in the periodic table for which you need an interatomic model.

KIM Models (interatomic potentials and force fields) are software packages for describing atomic interactions that can be used with a variety of simulation codes, including LAMMPS, DL\_POLY, IMD, ASE and GULP, that are compatible with the KIM API standard.

H																	He
Li	Be	<div><div></div></div>										B	C	N	O	F	Ne
Na	Mg	<div><div>↑</div><div>0 Models</div></div>										Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra		Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Uut	Fl	Uup	Lv	Uus	Uuo

La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr

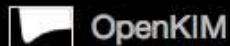


# OpenKIM with LAMMPS

Cu

Extended KIM ID	Title
<a href="#">EAM_Dynamo_Ackland_Tichy_Cu__MO_179025990738_001</a>	Finnis Sinclair potential for Cu
<a href="#">EAM_Dynamo_Bonny_Pasianot_FeCuNi__MO_469343973171_001</a>	FeCuNi potential to model reactor pressure vessel steels
<a href="#">EAM_Dynamo_Cai_Ye_AlCu__MO_942551040047_001</a>	EAM potential for Al-Cu binary system
<a href="#">EAM_Dynamo_Folles_Baskes_Universal3_Cu__MO_666348409573_000</a>	Third universal Cu potential of Folles, Baskes, and Daw; obtained from LAMMPS
<a href="#">EAM_Dynamo_Hoyt_Garvin_PbCu__MO_119135752160_001</a>	Embedded Atom Method parametrization of the Pb-Cu system
<a href="#">EAM_Dynamo_Mendelev_King_Cu__MO_748636486270_001</a>	FS potential for Cu
<a href="#">EAM_Dynamo_Mendelev_Kramer_Cu__MO_945691923444_001</a>	FS/EAM potential for Cu
<a href="#">EAM_Dynamo_Mendelev_Kramer_CuZr__MO_600021860456_001</a>	FS potential for Cu-Zr
<a href="#">EAM_Dynamo_Mendelev_Sordelet_CuZr__MO_120596890176_001</a>	FS potential for Cu-Zr
<a href="#">EAM_Dynamo_Mishin_Mehl_Cu__MO_346334655118_001</a>	EAM Cu Potential
<a href="#">EAM_Dynamo_Onat_Durukanoglu_CuNi__MO_592013496703_001</a>	An optimized EAM potential for Cu-Ni alloys
<a href="#">EAM_Dynamo_Williams_Mishin_CuAg__MO_128703483589_001</a>	EAM alloy potential for the Cu-Ag system.
<a href="#">EAM_Dynamo_Wu_Trinkle_CuAg__MO_270337113239_001</a>	EAM potential for Cu/Ag(111) Surface Diffusion.
<a href="#">EAM_Dynamo_Zhou_Johnson_Cu__MO_127245782811_001</a>	EAM alloy potential set table, compatible with LAMMPS
<a href="#">EAM_Johnson_NearestNeighbor_Cu__MO_887933271505_001</a>	This is an analytical NN EAM model for Cu by Johnson.
<a href="#">EMT_Asap_MetalGlass_CuMgZr__MO_655725647552_002</a>	Effective Medium Theory potential for CuMg and CuZr alloys, in particular metallic glasses.
<a href="#">EMT_Asap_Standard_Jacobsen_Stoltze_Norskov_AlAgAuCuNiPdPt__MO_118428466217_002</a>	Standard Effective Medium Theory potential for face-centered cubic metals as implemented in ASE/Asap.
<a href="#">MEAM_2NN_Fe_to_Ga__MO_145522277939_001</a>	Model parameterization of 2NN MEAM model
<a href="#">Pair_Morse_Modified_MacDonaldMacDonald_Cu__MO_034823476734_000</a>	Modified Morse pair potential for copper due to MacDonald and MacDonald
<a href="#">Pair_Morse_Shifted_GirifalcoWeizer_HighCutoff_Cu__MO_151002396060_001</a>	This is a Cu Morse Model Parameterization by Girifalco and Weizer using a high accuracy cutoff distance.

# OpenKIM with LAMMPS



Ryan



Jump to: [Tests](#) | [Visualizers](#) | [Files](#)

Contributor: [gbonny \(username default\)](#)

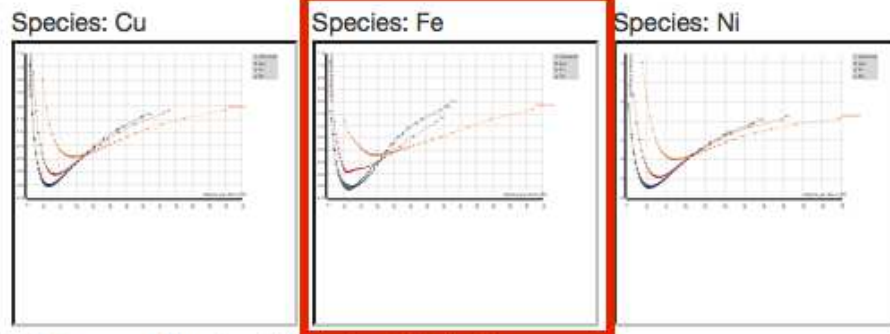
Maintainer: [gbonny \(username default\)](#)

## EAM\_Dynamo\_Bonny\_Pasianot\_FeCuNi\_MO\_469343973171\_001

Title ?	FeCuNi potential to model reactor pressure vessel steels
Short KIM ID ?	<a href="#">MO_469343973171_001</a>
Extended KIM ID ?	<a href="#">EAM_Dynamo_Bonny_Pasianot_FeCuNi_MO_469343973171_001</a>
KIM Item Type ?	Parameterized Model using Model Driver <a href="#">EAM_Dynamo_MD_120291908751_001</a>
Species ?	Cu, Fe, Ni
Description ?	<p>Ternary FeCuNi EAM-type potential.</p> <p>The fitting was focussed on solute-point defect interaction in the bcc Fe matrix. With respect to thermodynamics the following was accounted for: experimentally observed intermetallic compounds in the FeNi alloys, the Cu solubility in the FeCu binary and the CuNi miscibility gap. The potential is designed to model radiation damage in the FeCuNi model alloy which represents reactor pressure vessel steels. FeNi cross potential is taken from [Bonny et. al., Modelling Simul. Mater. Sci. Eng. 17 (2009) 025010]. FeCu cross potential is taken from [Pasianot and Malerba, J. Nucl. Mater. 360 (2007) 118]. Fe potential is taken from [Mendelev et al., Philos. Mag. 83 (2003) 3977]. Ni potential is taken from [Voter and Chen, Mater. Res. Soc. Symp. Proc. 82 (1987) 175]. Cu potential is taken from [Mishin et al., Phys. Rev. B 63 (2001) 224106].</p>
Disclaimer ?	The potential is stiffened.
Source Citations ?	FeCuNi potential: G. Bonny, R.C. Pasianot, N. Castin and L. Malerba, Philos. Mag. 89 (2009) 3531

## Cohesive Energy Graph

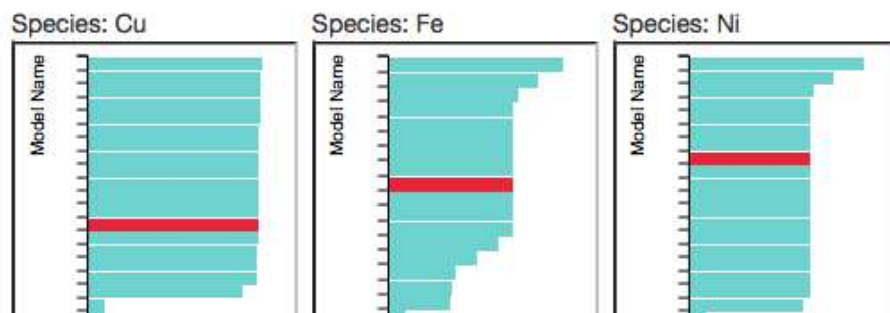
This graph shows the cohesive energy versus volume-per-atom for the current mode for four mono-atomic cubic phases (body-centered cubic (bcc), face-centered cubic (fcc), simple cubic (sc), and diamond). The curve with the lowest minimum is the ground state of the crystal if stable. (The crystal structure is enforced in these calculations, so the phase may not be stable.) Graphs are generated for each species supported by the model.



Click on any thumbnail to get a full size image.

## Horizontal Chart

This bar chart plot shows the mono-atomic face-centered cubic (fcc) lattice constant predicted by the current model (shown in red) compared with the predictions for all other models in the OpenKIM Repository that support the species. The vertical bars show the average and standard deviation (one sigma) bounds for all model predictions. Graphs are generated for each species supported by the model.





# OpenKIM with LAMMPS

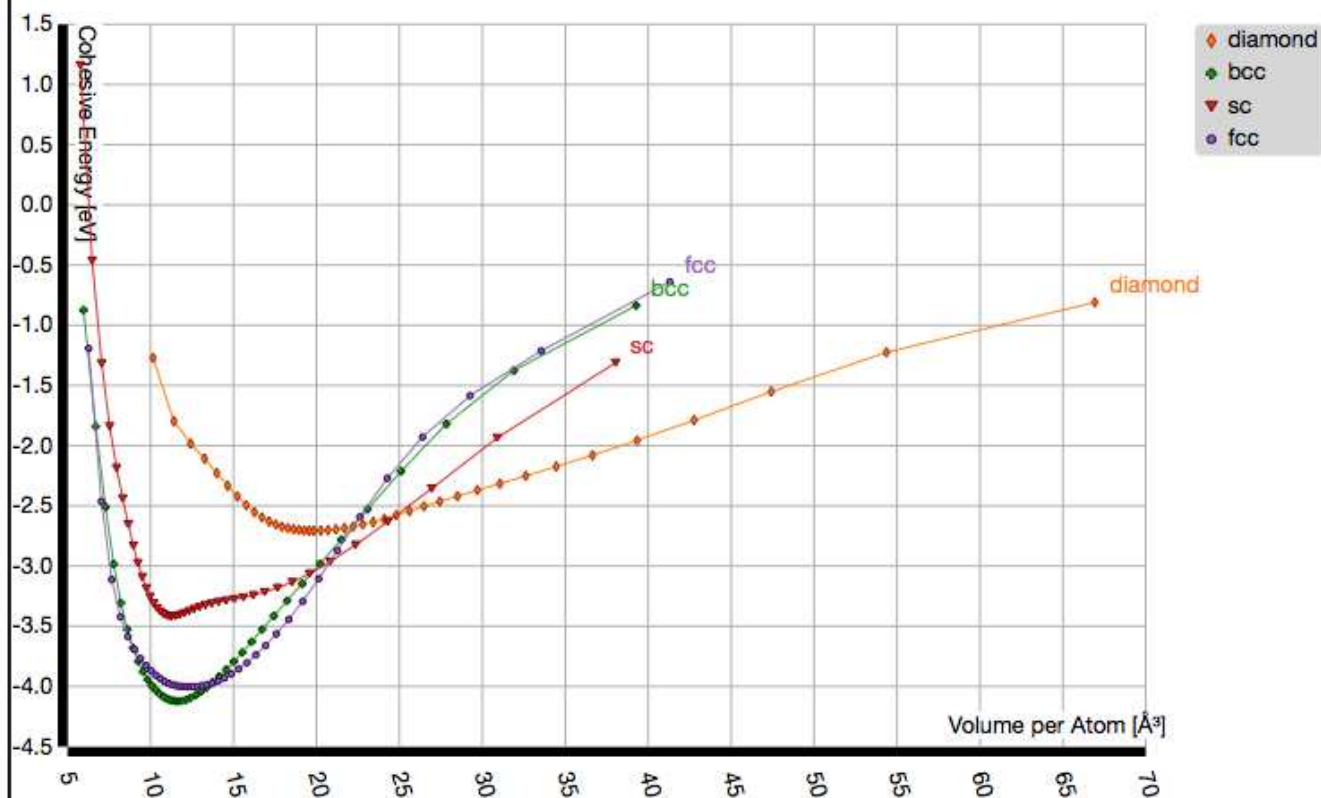
OpenKIM

Ryan



Model: EAM\_Dynamo\_Bonny\_Pasianot\_FeCuNi\_MO\_469343973171\_001

Species: Fe



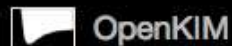


# OpenKIM with LAMMPS

Cu

Extended KIM ID	Title
<a href="#">EAM_Dynamo_Ackland_Tichy_Cu__MO_179025990738_001</a>	Finnis Sinclair potential for Cu
<a href="#">EAM_Dynamo_Bonny_Pasianot_FeCuNi__MO_469343973171_001</a>	FeCuNi potential to model reactor pressure vessel steels
<a href="#">EAM_Dynamo_Cai_Ye_AlCu__MO_942551040047_001</a>	EAM potential for Al-Cu binary system
<a href="#">EAM_Dynamo_Folles_Baskes_Universal3_Cu__MO_666348409573_000</a>	Third universal Cu potential of Folles, Baskes, and Daw; obtained from LAMMPS
<a href="#">EAM_Dynamo_Hoyt_Garvin_PbCu__MO_119135752160_001</a>	Embedded Atom Method parametrization of the Pb-Cu system
<a href="#">EAM_Dynamo_Mendelev_King_Cu__MO_748636486270_001</a>	FS potential for Cu
<a href="#">EAM_Dynamo_Mendelev_Kramer_Cu__MO_945691923444_001</a>	FS/EAM potential for Cu
<a href="#">EAM_Dynamo_Mendelev_Kramer_CuZr__MO_600021860456_001</a>	FS potential for Cu-Zr
<a href="#">EAM_Dynamo_Mendelev_Sordelet_CuZr__MO_120596890176_001</a>	FS potential for Cu-Zr
<a href="#">EAM_Dynamo_Mishin_Mehl_Cu__MO_346334655118_001</a>	EAM Cu Potential
<a href="#">EAM_Dynamo_Onat_Durukanoglu_CuNi__MO_592013496703_001</a>	An optimized EAM potential for Cu-Ni alloys
<a href="#">EAM_Dynamo_Williams_Mishin_CuAg__MO_128703483589_001</a>	EAM alloy potential for the Cu-Ag system.
<a href="#">EAM_Dynamo_Wu_Trinkle_CuAg__MO_270337113239_001</a>	EAM potential for Cu/Ag(111) Surface Diffusion.
<a href="#">EAM_Dynamo_Zhou_Johnson_Cu__MO_127245782811_001</a>	EAM alloy potential set table, compatible with LAMMPS
<a href="#">EAM_Johnson_NearestNeighbor_Cu__MO_887933271505_001</a>	This is an analytical NN EAM model for Cu by Johnson.
<a href="#">EMT_Asap_MetalGlass_CuMgZr__MO_655725647552_002</a>	Effective Medium Theory potential for CuMg and CuZr alloys, in particular metallic glasses.
<a href="#">EMT_Asap_Standard_Jacobsen_Stoltze_Norskov_AlAgAuCuNiPdPt__MO_118428466217_002</a>	Standard Effective Medium Theory potential for face-centered cubic metals as implemented in ASE/Asap.
<a href="#">MEAM_2NN_Fe_to_Ga__MO_145522277939_001</a>	Model parameterization of 2NN MEAM model
<a href="#">Pair_Morse_Modified_MacDonaldMacDonald_Cu__MO_034823476734_000</a>	Modified Morse pair potential for copper due to MacDonald and MacDonald
<a href="#">Pair_Morse_Shifted_GirifalcoWeizer_HighCutoff_Cu__MO_151002396060_001</a>	This is a Cu Morse Model Parameterization by Girifalco and Weizer using a high accuracy cutoff distance.

# OpenKIM with LAMMPS



Ryan



Jump to: [Tests](#) | [Visualizers](#) | [Files](#)

Contributor: [Ryan](#)

Maintainer: [Ryan](#)

## EAM\_Dynamo\_Foiles\_Baskes\_Universal3\_Cu\_\_MO\_666348409573\_000

Title ?	Third universal Cu potential of Foiles, Baskes, and Daw; obtained from LAMMPS
Short KIM ID ?	<a href="#">MO_666348409573_000</a>
Extended KIM ID ?	<a href="#">EAM_Dynamo_Foiles_Baskes_Universal3_Cu__MO_666348409573_000</a>
KIM Item Type ?	Parameterized Model using Model Driver <a href="#">EAM_Dynamo__MD_120291908751_001</a>
Species ?	Cu
Description ?	This Cu EAM potential parameter file was obtained from the LAMMPS distribution and is dated 2007-06-11. It is the "universal 3" potential from the paper by Foiles, Baskes, and Daw.
Disclaimer ?	
Source Citations ?	"Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys," by S. M. Foiles, M. I. Baskes, and M. S. Daw, Phys. Rev. B 33, 7983 (1986); Erratum Phys. Rev. B 37, 10378 (1988) <a href="http://dx.doi.org/10.1103/PhysRevB.33.7983">http://dx.doi.org/10.1103/PhysRevB.33.7983</a>
Programming Language(s) ?	N/A

## Visualizers (in-page)

# OpenKIM with LAMMPS

## Files

 [Cu\\_u3.eam](#)

 [LICENSE](#)

 [Makefile](#)

 [kimspec.edn](#)

## Download

<a href="#">EAM_Dynamo_Folles_Baskes_Universal3_Cu__MO_666348409573_000.txz</a>	Tar+XZ	Linux and OS X archive (modern compression)
<a href="#">EAM_Dynamo_Folles_Baskes_Universal3_Cu__MO_666348409573_000.tgz</a>	Tar+Gzip	Linux and OS X archive (legacy compression)
<a href="#">EAM_Dynamo_Folles_Baskes_Universal3_Cu__MO_666348409573_000.zip</a>	Zip	Windows archive

## ⚠ Download Dependency

This Model requires a Model Driver. Archives for the Model Driver [EAM\\_Dynamo\\_\\_MD\\_120291908751\\_001](#) appear below.

<a href="#">EAM_Dynamo__MD_120291908751_001.txz</a>	Tar+XZ	Linux and OS X archive (modern compression)
<a href="#">EAM_Dynamo__MD_120291908751_001.tgz</a>	Tar+Gzip	Linux and OS X archive (legacy compression)
<a href="#">EAM_Dynamo__MD_120291908751_001.zip</a>	Zip	Windows archive

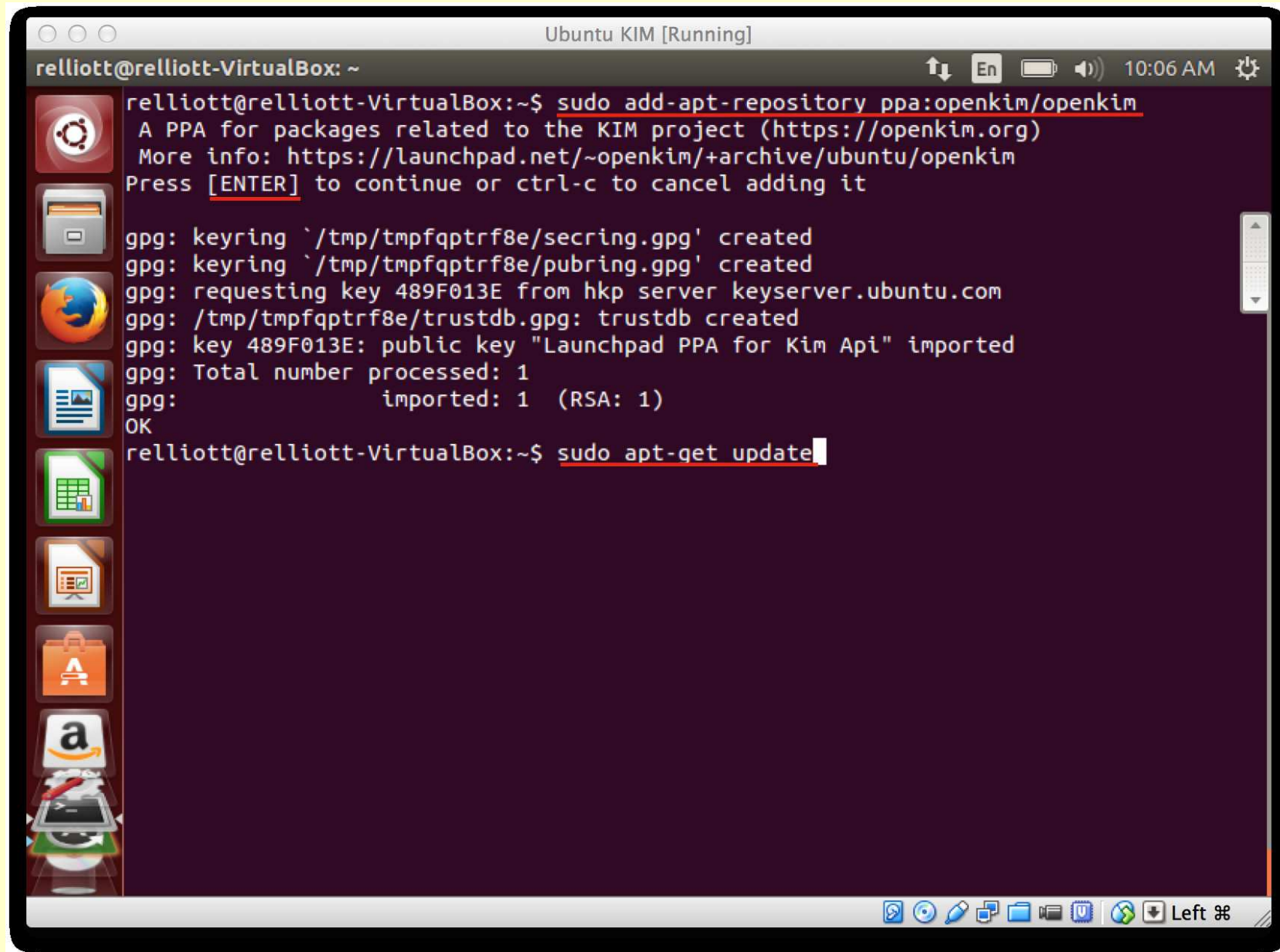
# OUTLINE

1. The Knowledgebase of Interatomic Models
  - User Signup
  - KIM API
  - “KIM Compliant” Simulators
  - Setup VirtualBox Ubuntu VM
  - KIM Model performance within LAMMPS
  - KIM Model pages
2. Installing the KIM API on Ubuntu ([ppa:openkim/openkim](https://ppa.launchpad.net/openkim/openkim/ubuntu))
3. Using LAMMPS with the KIM API

Appendix: Installing the KIM API by hand

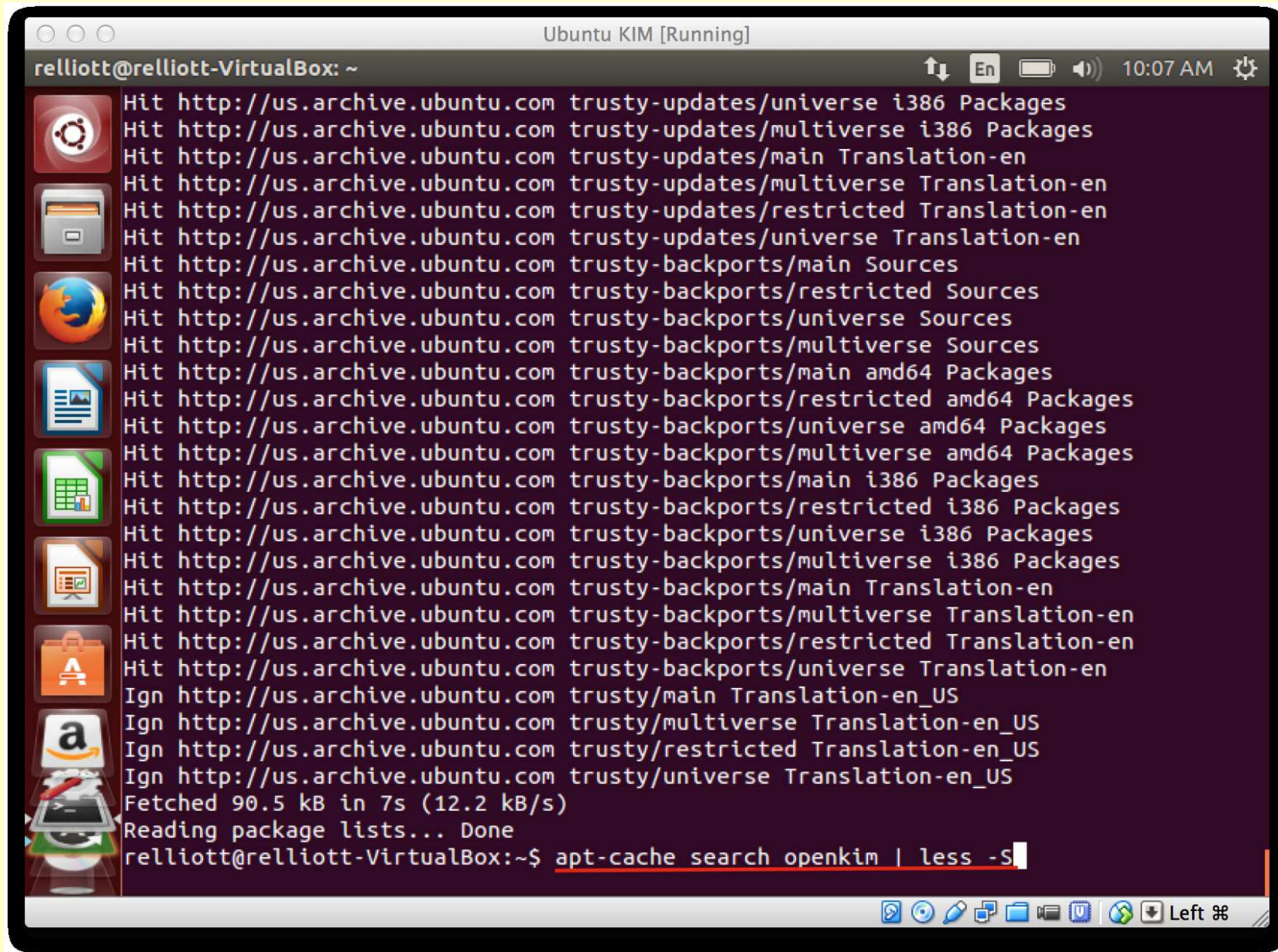


# OpenKIM with LAMMPS



```
reliott@reliott-VirtualBox: ~  
reliott@reliott-VirtualBox:~$ sudo add-apt-repository ppa:openkim/openkim  
A PPA for packages related to the KIM project (https://openkim.org)  
More info: https://launchpad.net/~openkim/+archive/ubuntu/openkim  
Press [ENTER] to continue or ctrl-c to cancel adding it  
  
gpg: keyring `/tmp/tmpfqptrf8e/secring.gpg' created  
gpg: keyring `/tmp/tmpfqptrf8e/pubring.gpg' created  
gpg: requesting key 489F013E from hkp server keyserver.ubuntu.com  
gpg: /tmp/tmpfqptrf8e/trustdb.gpg: trustdb created  
gpg: key 489F013E: public key "Launchpad PPA for Kim Api" imported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)  
OK  
reliott@reliott-VirtualBox:~$ sudo apt-get update
```

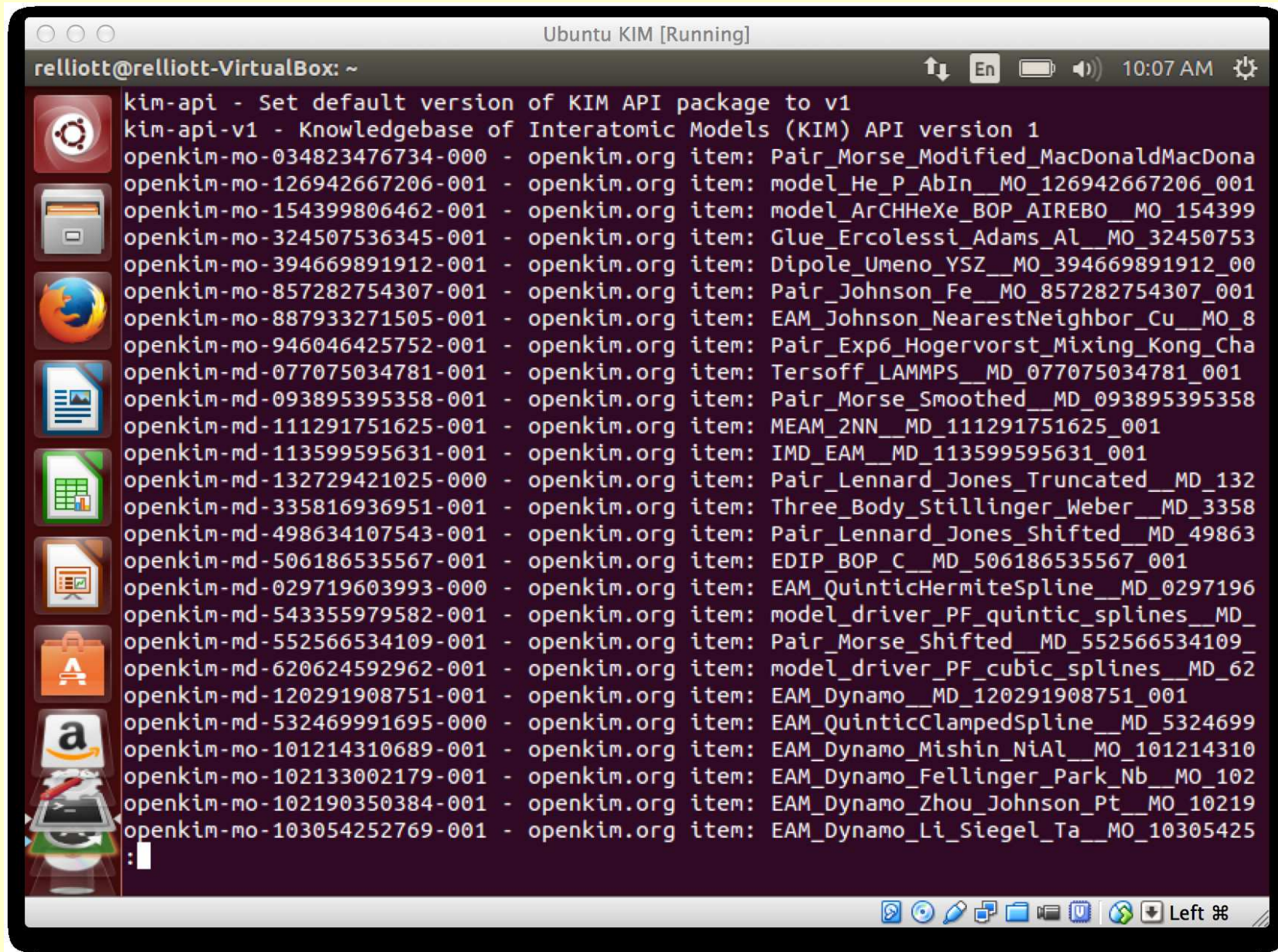
# OpenKIM with LAMMPS



```
Ubuntu KIM [Running]
relliott@relliott-VirtualBox: ~
Hit http://us.archive.ubuntu.com trusty-updates/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty-updates/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty-updates/main Translation-en
Hit http://us.archive.ubuntu.com trusty-updates/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-updates/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty-updates/universe Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/main Sources
Hit http://us.archive.ubuntu.com trusty-backports/restricted Sources
Hit http://us.archive.ubuntu.com trusty-backports/universe Sources
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Sources
Hit http://us.archive.ubuntu.com trusty-backports/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Fetched 90.5 kB in 7s (12.2 kB/s)
Reading package lists... Done
relliott@relliott-VirtualBox:~$ apt-cache search openkim | less -s
```



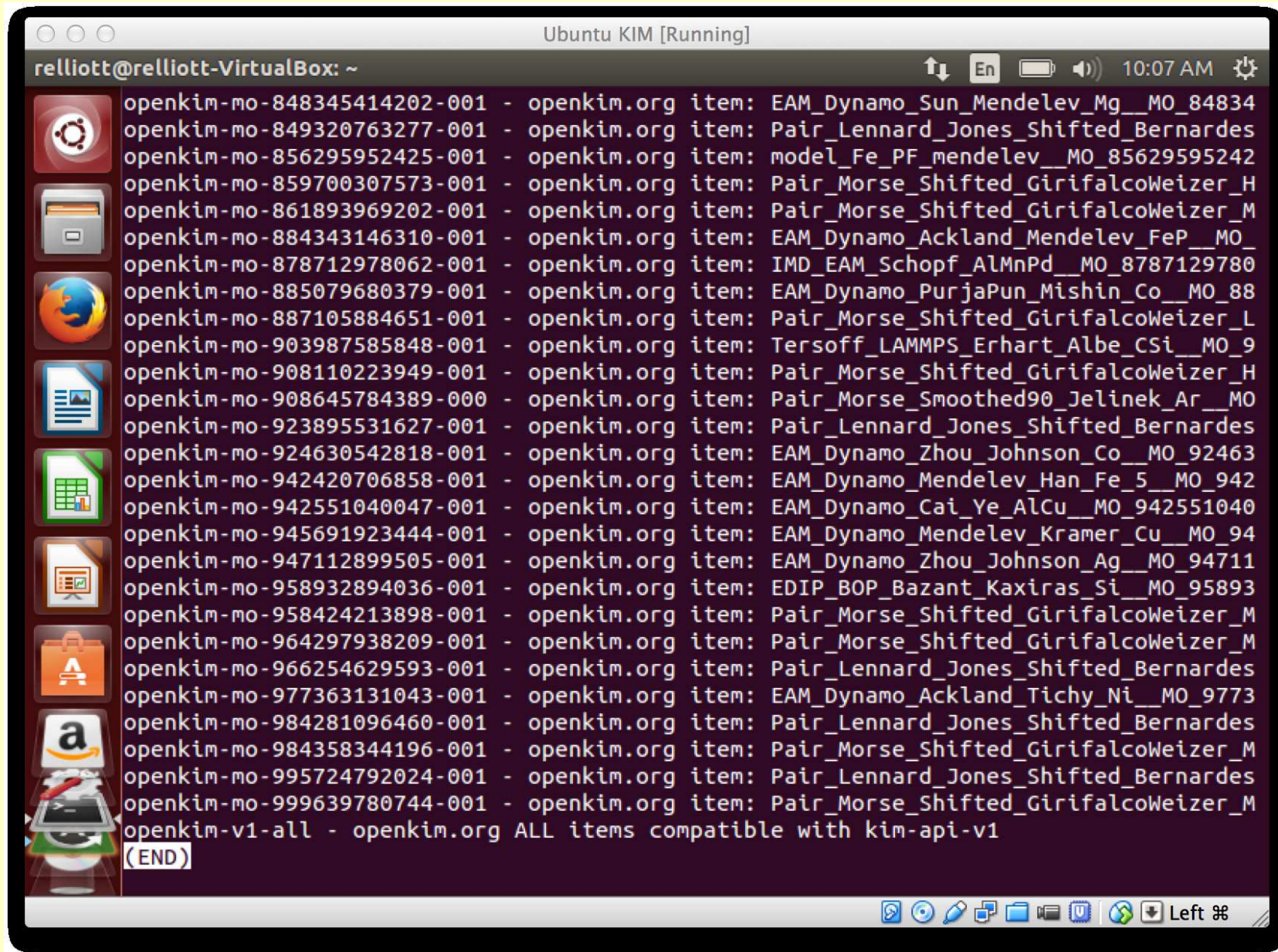
# OpenKIM with LAMMPS



```
rellyott@rellyott-VirtualBox: ~
kim-api - Set default version of KIM API package to v1
kim-api-v1 - Knowledgebase of Interatomic Models (KIM) API version 1
openkim-mo-034823476734-000 - openkim.org item: Pair_Morse_Modified_MacDonaldMacDona
openkim-mo-126942667206-001 - openkim.org item: model_He_P_AbIn__MO_126942667206_001
openkim-mo-154399806462-001 - openkim.org item: model_ARCHHeXe_BOP_AIREBO__MO_154399
openkim-mo-324507536345-001 - openkim.org item: Glue_Ercolessi_Adams_Al__MO_32450753
openkim-mo-394669891912-001 - openkim.org item: Dipole_Umeno_YSZ__MO_394669891912_00
openkim-mo-857282754307-001 - openkim.org item: Pair_Johnson_Fe__MO_857282754307_001
openkim-mo-887933271505-001 - openkim.org item: EAM_Johnson_NearestNeighbor_Cu__MO_8
openkim-mo-946046425752-001 - openkim.org item: Pair_Exp6_Hogervorst_Mixing_Kong_Cha
openkim-md-077075034781-001 - openkim.org item: Tersoff_LAMMPS__MD_077075034781_001
openkim-md-093895395358-001 - openkim.org item: Pair_Morse_Smoothed__MD_093895395358
openkim-md-111291751625-001 - openkim.org item: MEAM_2NN__MD_111291751625_001
openkim-md-113599595631-001 - openkim.org item: IMD_EAM__MD_113599595631_001
openkim-md-132729421025-000 - openkim.org item: Pair_Lennard_Jones_Truncated__MD_132
openkim-md-335816936951-001 - openkim.org item: Three_Body_Stillinger_Weber__MD_3358
openkim-md-498634107543-001 - openkim.org item: Pair_Lennard_Jones_Shifted__MD_49863
openkim-md-506186535567-001 - openkim.org item: EDIP_BOP_C__MD_506186535567_001
openkim-md-029719603993-000 - openkim.org item: EAM_QuinticHermiteSpline__MD_0297196
openkim-md-543355979582-001 - openkim.org item: model_driver_PF_quintic_splines__MD_
openkim-md-552566534109-001 - openkim.org item: Pair_Morse_Shifted__MD_552566534109_
openkim-md-620624592962-001 - openkim.org item: model_driver_PF_cubic_splines__MD_62
openkim-md-120291908751-001 - openkim.org item: EAM_Dynamo__MD_120291908751_001
openkim-md-532469991695-000 - openkim.org item: EAM_QuinticClampedSpline__MD_5324699
openkim-mo-101214310689-001 - openkim.org item: EAM_Dynamo_Mishin_NiAl__MO_101214310
openkim-mo-102133002179-001 - openkim.org item: EAM_Dynamo_Fellinger_Park_Nb__MO_102
openkim-mo-102190350384-001 - openkim.org item: EAM_Dynamo_Zhou_Johnson_Pt__MO_10219
openkim-mo-103054252769-001 - openkim.org item: EAM_Dynamo_Li_Siegel-Ta__MO_10305425
:
```



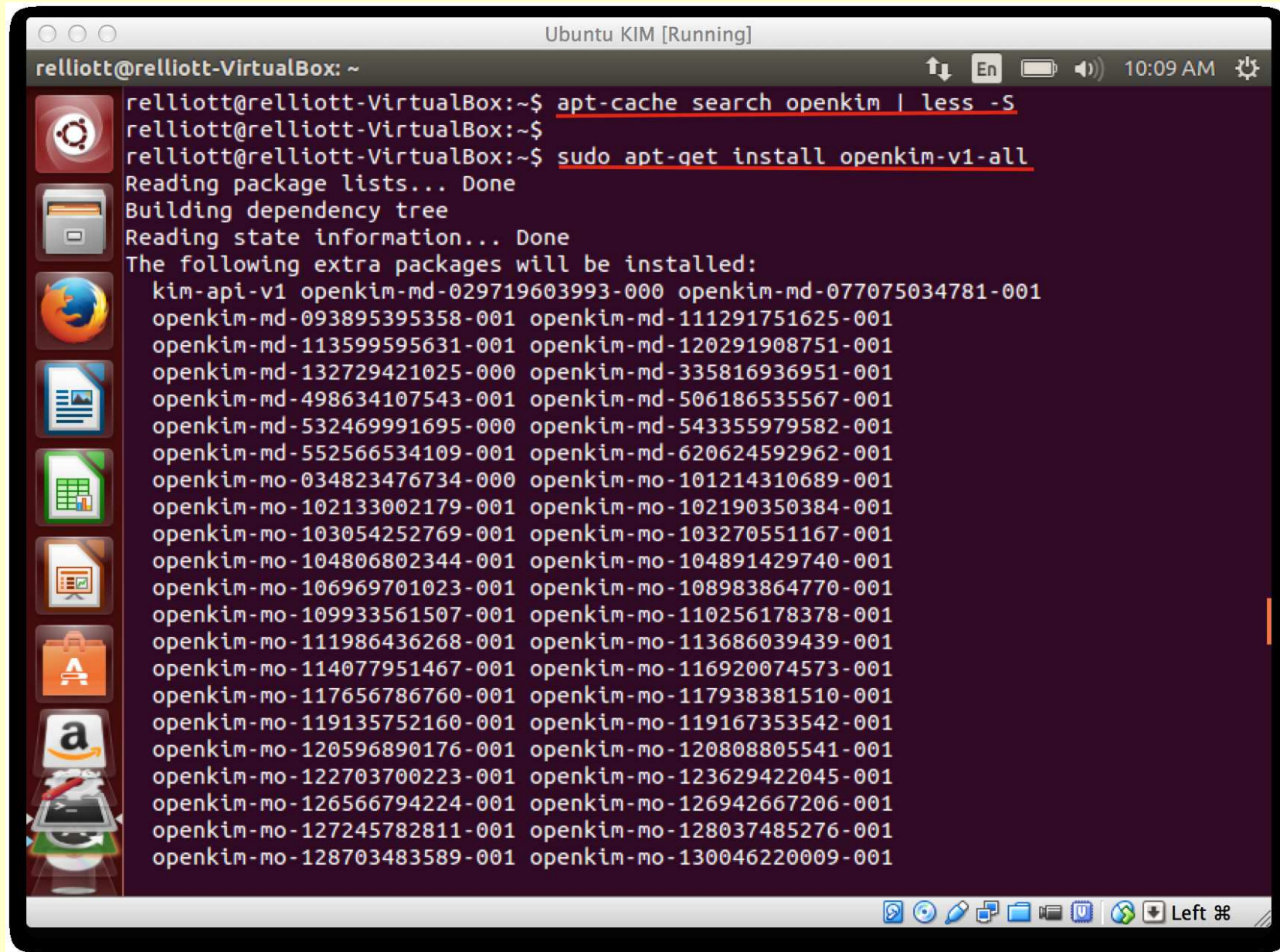
# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~
openkim-mo-848345414202-001 - openkim.org item: EAM_Dynamo_Sun_Mendelev_Mg__MO_84834
openkim-mo-849320763277-001 - openkim.org item: Pair_Lennard_Jones_Shifted_Bernardes
openkim-mo-856295952425-001 - openkim.org item: model_Fe_PF_mendelev__MO_85629595242
openkim-mo-859700307573-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_H
openkim-mo-861893969202-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_M
openkim-mo-884343146310-001 - openkim.org item: EAM_Dynamo_Ackland_Mendelev_FeP__MO_
openkim-mo-878712978062-001 - openkim.org item: IMD_EAM_Schopf_ALMnPd__MO_8787129780
openkim-mo-885079680379-001 - openkim.org item: EAM_Dynamo_PurjaPun_Mishin_Co__MO_88
openkim-mo-887105884651-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_L
openkim-mo-903987585848-001 - openkim.org item: Tersoff_LAMMPS_Erhart_Albe_CSi__MO_9
openkim-mo-908110223949-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_H
openkim-mo-908645784389-000 - openkim.org item: Pair_Morse_Smoothed90_Jelinek_Ar__MO
openkim-mo-923895531627-001 - openkim.org item: Pair_Lennard_Jones_Shifted_Bernardes
openkim-mo-924630542818-001 - openkim.org item: EAM_Dynamo_Zhou_Johnson_Co__MO_92463
openkim-mo-942420706858-001 - openkim.org item: EAM_Dynamo_Mendelev_Han_Fe_5__MO_942
openkim-mo-942551040047-001 - openkim.org item: EAM_Dynamo_Cai_Ye_AlCu__MO_942551040
openkim-mo-945691923444-001 - openkim.org item: EAM_Dynamo_Mendelev_Kramer_Cu__MO_94
openkim-mo-947112899505-001 - openkim.org item: EAM_Dynamo_Zhou_Johnson_Ag__MO_94711
openkim-mo-958932894036-001 - openkim.org item: EDIP_BOP_Bazant_Kaxiras_Si__MO_95893
openkim-mo-958424213898-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_M
openkim-mo-964297938209-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_M
openkim-mo-966254629593-001 - openkim.org item: Pair_Lennard_Jones_Shifted_Bernardes
openkim-mo-977363131043-001 - openkim.org item: EAM_Dynamo_Ackland_Tichy_Ni__MO_9773
openkim-mo-984281096460-001 - openkim.org item: Pair_Lennard_Jones_Shifted_Bernardes
openkim-mo-984358344196-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_M
openkim-mo-995724792024-001 - openkim.org item: Pair_Lennard_Jones_Shifted_Bernardes
openkim-mo-999639780744-001 - openkim.org item: Pair_Morse_Shifted_GirifalcoWeizer_M
openkim-v1-all - openkim.org ALL items compatible with kim-api-v1
(END)
```

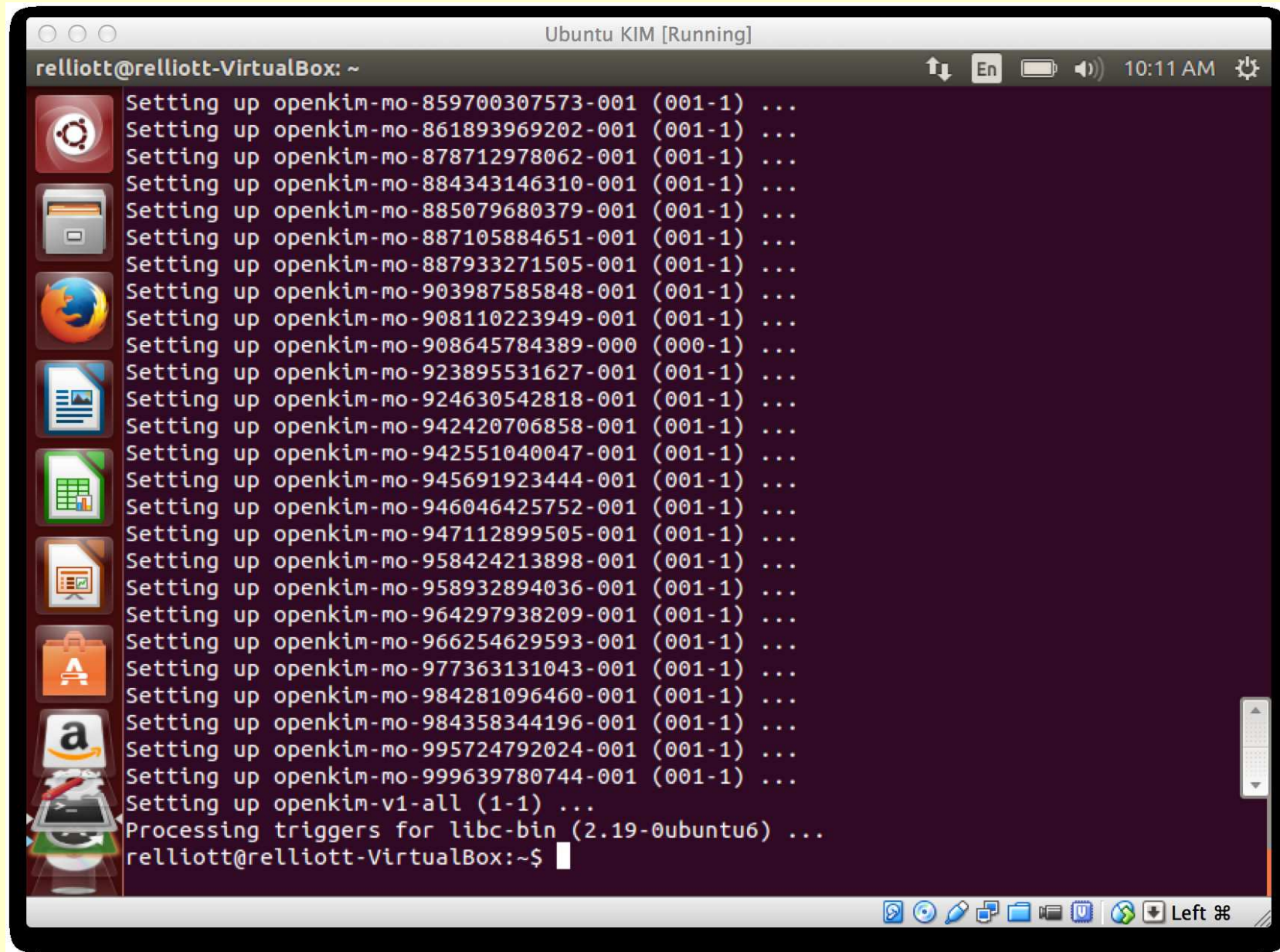


# OpenKIM with LAMMPS



```
reliott@reliott-VirtualBox: ~  
reliott@reliott-VirtualBox:~$ apt-cache search openkim | less -S  
reliott@reliott-VirtualBox:~$ sudo apt-get install openkim-v1-all  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
kim-api-v1 openkim-md-029719603993-000 openkim-md-077075034781-001  
openkim-md-093895395358-001 openkim-md-111291751625-001  
openkim-md-113599595631-001 openkim-md-120291908751-001  
openkim-md-132729421025-000 openkim-md-335816936951-001  
openkim-md-498634107543-001 openkim-md-506186535567-001  
openkim-md-532469991695-000 openkim-md-543355979582-001  
openkim-md-552566534109-001 openkim-md-620624592962-001  
openkim-mo-034823476734-000 openkim-mo-101214310689-001  
openkim-mo-102133002179-001 openkim-mo-102190350384-001  
openkim-mo-103054252769-001 openkim-mo-103270551167-001  
openkim-mo-104806802344-001 openkim-mo-104891429740-001  
openkim-mo-106969701023-001 openkim-mo-108983864770-001  
openkim-mo-109933561507-001 openkim-mo-110256178378-001  
openkim-mo-111986436268-001 openkim-mo-113686039439-001  
openkim-mo-114077951467-001 openkim-mo-116920074573-001  
openkim-mo-117656786760-001 openkim-mo-117938381510-001  
openkim-mo-119135752160-001 openkim-mo-119167353542-001  
openkim-mo-120596890176-001 openkim-mo-120808805541-001  
openkim-mo-122703700223-001 openkim-mo-123629422045-001  
openkim-mo-126566794224-001 openkim-mo-126942667206-001  
openkim-mo-127245782811-001 openkim-mo-128037485276-001  
openkim-mo-128703483589-001 openkim-mo-130046220009-001
```

# OpenKIM with LAMMPS



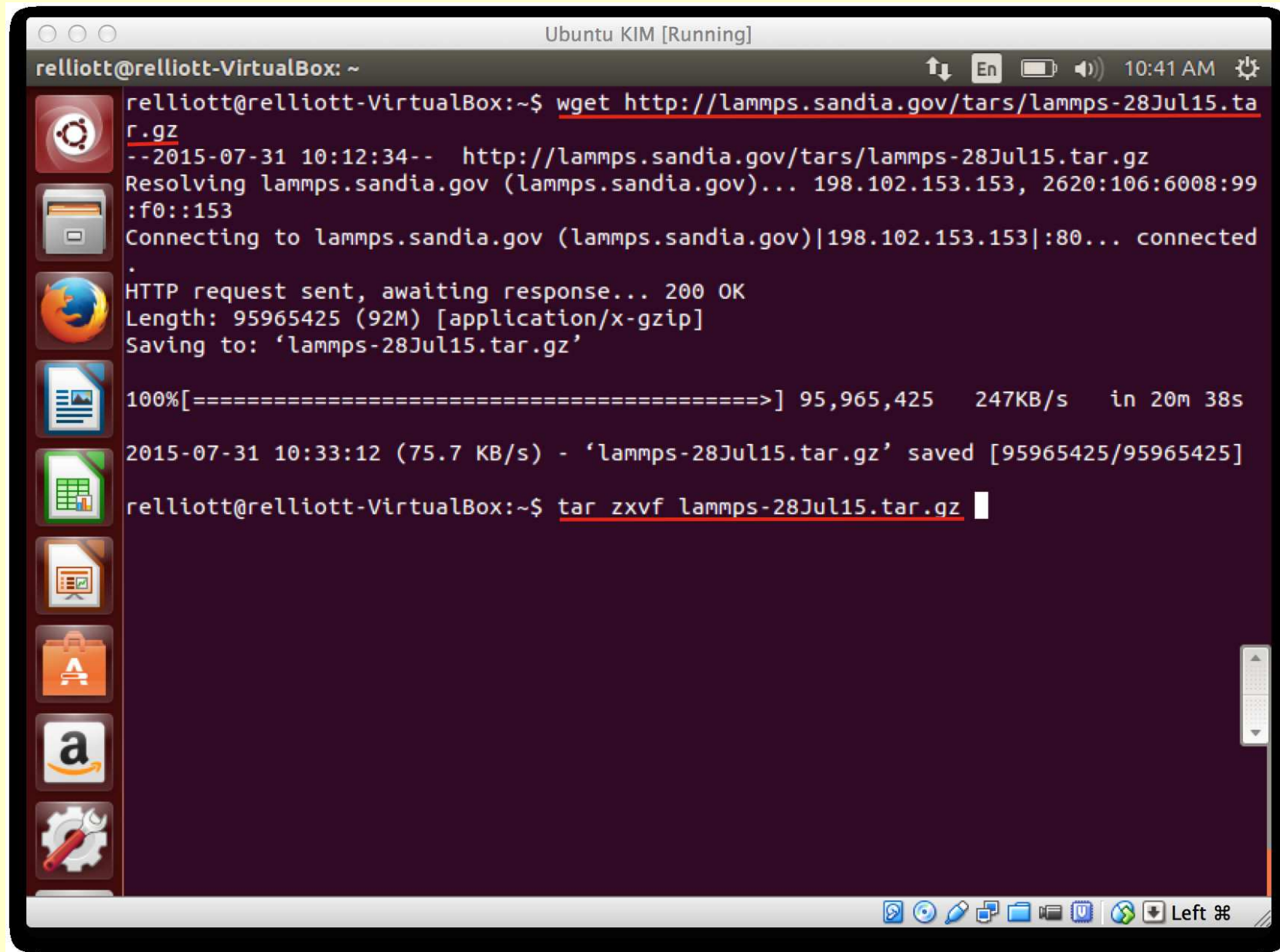
```
Ubuntu KIM [Running]
relliott@relliott-VirtualBox: ~
Setting up openkim-mo-859700307573-001 (001-1) ...
Setting up openkim-mo-861893969202-001 (001-1) ...
Setting up openkim-mo-878712978062-001 (001-1) ...
Setting up openkim-mo-884343146310-001 (001-1) ...
Setting up openkim-mo-885079680379-001 (001-1) ...
Setting up openkim-mo-887105884651-001 (001-1) ...
Setting up openkim-mo-887933271505-001 (001-1) ...
Setting up openkim-mo-903987585848-001 (001-1) ...
Setting up openkim-mo-908110223949-001 (001-1) ...
Setting up openkim-mo-908645784389-000 (000-1) ...
Setting up openkim-mo-923895531627-001 (001-1) ...
Setting up openkim-mo-924630542818-001 (001-1) ...
Setting up openkim-mo-942420706858-001 (001-1) ...
Setting up openkim-mo-942551040047-001 (001-1) ...
Setting up openkim-mo-945691923444-001 (001-1) ...
Setting up openkim-mo-946046425752-001 (001-1) ...
Setting up openkim-mo-947112899505-001 (001-1) ...
Setting up openkim-mo-958424213898-001 (001-1) ...
Setting up openkim-mo-958932894036-001 (001-1) ...
Setting up openkim-mo-964297938209-001 (001-1) ...
Setting up openkim-mo-966254629593-001 (001-1) ...
Setting up openkim-mo-977363131043-001 (001-1) ...
Setting up openkim-mo-984281096460-001 (001-1) ...
Setting up openkim-mo-984358344196-001 (001-1) ...
Setting up openkim-mo-995724792024-001 (001-1) ...
Setting up openkim-mo-999639780744-001 (001-1) ...
Setting up openkim-v1-all (1-1) ...
Processing triggers for libc-bin (2.19-0ubuntu6) ...
relliott@relliott-VirtualBox:~$
```

1. The Knowledgebase of Interatomic Models
  - User Signup
  - KIM API
  - “KIM Compliant” Simulators
  - Setup VirtualBox Ubuntu VM
  - KIM Model performance within LAMMPS
  - KIM Model pages
2. Installing the KIM API on Ubuntu (ppa:openkim/openkim)
3. **Using LAMMPS with the KIM API**

Appendix: Installing the KIM API by hand



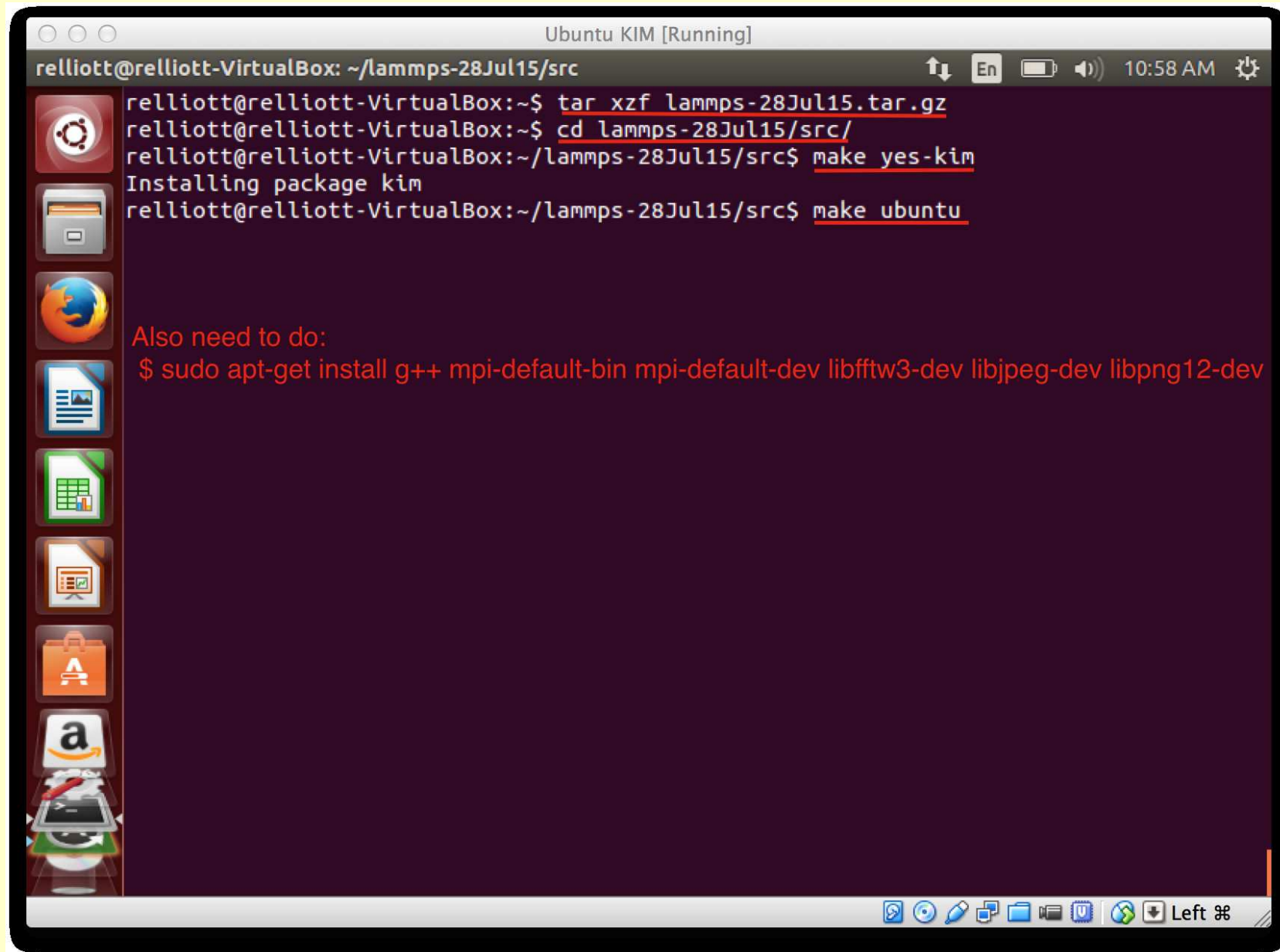
# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~  
relliott@relliott-VirtualBox:~$ wget http://lammps.sandia.gov/tars/lammps-28Jul15.tar.gz  
--2015-07-31 10:12:34-- http://lammps.sandia.gov/tars/lammps-28Jul15.tar.gz  
Resolving lammps.sandia.gov (lammps.sandia.gov)... 198.102.153.153, 2620:106:6008:99:f0::153  
Connecting to lammps.sandia.gov (lammps.sandia.gov)|198.102.153.153|:80... connected  
HTTP request sent, awaiting response... 200 OK  
Length: 95965425 (92M) [application/x-gzip]  
Saving to: 'lammps-28Jul15.tar.gz'  
  
100%[=====>] 95,965,425 247KB/s in 20m 38s  
2015-07-31 10:33:12 (75.7 KB/s) - 'lammps-28Jul15.tar.gz' saved [95965425/95965425]  
relliott@relliott-VirtualBox:~$ tar zxvf lammps-28Jul15.tar.gz
```



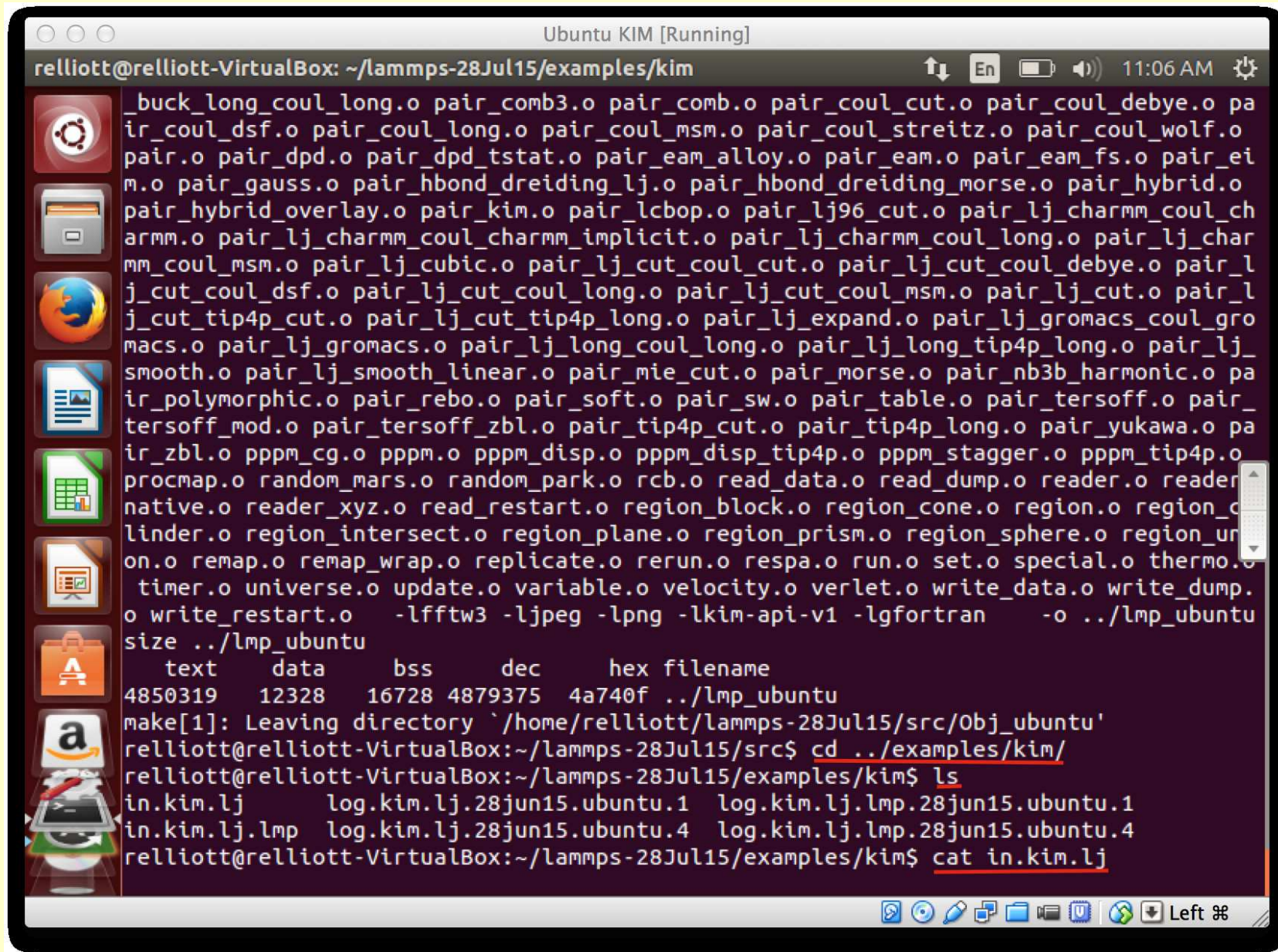
# OpenKIM with LAMMPS



```
reliott@reliott-VirtualBox: ~/lammps-28Jul15/src
reliott@reliott-VirtualBox:~$ tar xzf lammps-28Jul15.tar.gz
reliott@reliott-VirtualBox:~$ cd lammps-28Jul15/src/
reliott@reliott-VirtualBox:~/lammps-28Jul15/src$ make yes-kim
Installing package kim
reliott@reliott-VirtualBox:~/lammps-28Jul15/src$ make ubuntu
```

Also need to do:  
\$ sudo apt-get install g++ mpi-default-bin mpi-default-dev libfftw3-dev libjpeg-dev libpng12-dev

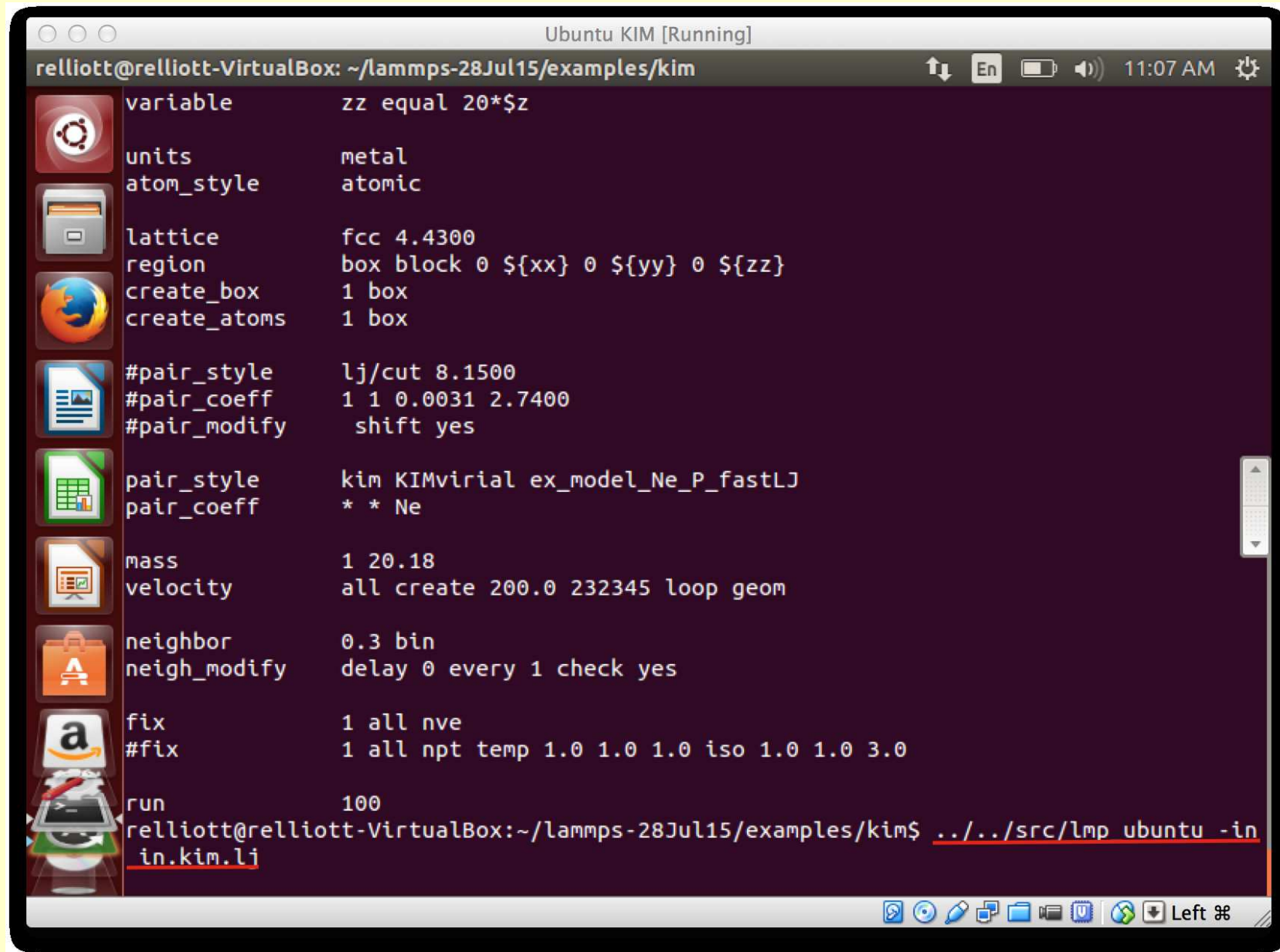
# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/lammps-28Jul15/examples/kim
_buck_long_coul_long.o pair_comb3.o pair_comb.o pair_coul_cut.o pair_coul_debye.o pa
ir_coul_dsf.o pair_coul_long.o pair_coul_msm.o pair_coul_streitz.o pair_coul_wolf.o
pair.o pair_dpd.o pair_dpd_tstat.o pair_eam_alloy.o pair_eam.o pair_eam_fs.o pair_ei
m.o pair_gauss.o pair_hbond_dreiding_lj.o pair_hbond_dreiding_morse.o pair_hybrid.o
pair_hybrid_overlay.o pair_kim.o pair_lcbop.o pair_lj96_cut.o pair_lj_charmm_coul_ch
armm.o pair_lj_charmm_coul_charmm_implicit.o pair_lj_charmm_coul_long.o pair_lj_char
mm_coul_msm.o pair_lj_cubic.o pair_lj_cut_coul_cut.o pair_lj_cut_coul_debye.o pair_l
j_cut_coul_dsf.o pair_lj_cut_coul_long.o pair_lj_cut_coul_msm.o pair_lj_cut.o pair_l
j_cut_tip4p_cut.o pair_lj_cut_tip4p_long.o pair_lj_expand.o pair_lj_gromacs_coul_gro
macs.o pair_lj_gromacs.o pair_lj_long_coul_long.o pair_lj_long_tip4p_long.o pair_lj_
smooth.o pair_lj_smooth_linear.o pair_mie_cut.o pair_morse.o pair_nb3b_harmonic.o pa
ir_polymorphic.o pair_rebo.o pair_soft.o pair_sw.o pair_table.o pair_tersoff.o pair_
tersoff_mod.o pair_tersoff_zbl.o pair_tip4p_cut.o pair_tip4p_long.o pair_yukawa.o pa
ir_zbl.o ppm.o ppm_cg.o ppm_disp.o ppm_disp_tip4p.o ppm_stagger.o ppm_tip4p.o
procmap.o random_mars.o random_park.o rcb.o read_data.o read_dump.o reader.o reader
native.o reader_xyz.o read_restart.o region_block.o region_cone.o region.o region_c
ylinder.o region_intersect.o region_plane.o region_prism.o region_sphere.o region_un
ion.o remap.o remap_wrap.o replicate.o rerun.o respa.o run.o set.o special.o thermo.o
timer.o universe.o update.o variable.o velocity.o verlet.o write_data.o write_dump.
o write_restart.o -lfftw3 -ljpeg -lpng -lkim-api-v1 -lgfortran -o ../lmp_ubuntu
size ../lmp_ubuntu
      text      data      bss      dec      hex filename
4850319    12328    16728 4879375 4a740f ../lmp_ubuntu
make[1]: Leaving directory `/home/relliott/lammps-28Jul15/src/Obj_ubuntu'
relliott@relliott-VirtualBox:~/lammps-28Jul15/src$ cd ../examples/kim/
relliott@relliott-VirtualBox:~/lammps-28Jul15/examples/kim$ ls
in.kim.lj      log.kim.lj.28jun15.ubuntu.1  log.kim.lj.lmp.28jun15.ubuntu.1
in.kim.lj.lmp  log.kim.lj.28jun15.ubuntu.4  log.kim.lj.lmp.28jun15.ubuntu.4
relliott@relliott-VirtualBox:~/lammps-28Jul15/examples/kim$ cat in.kim.lj
```

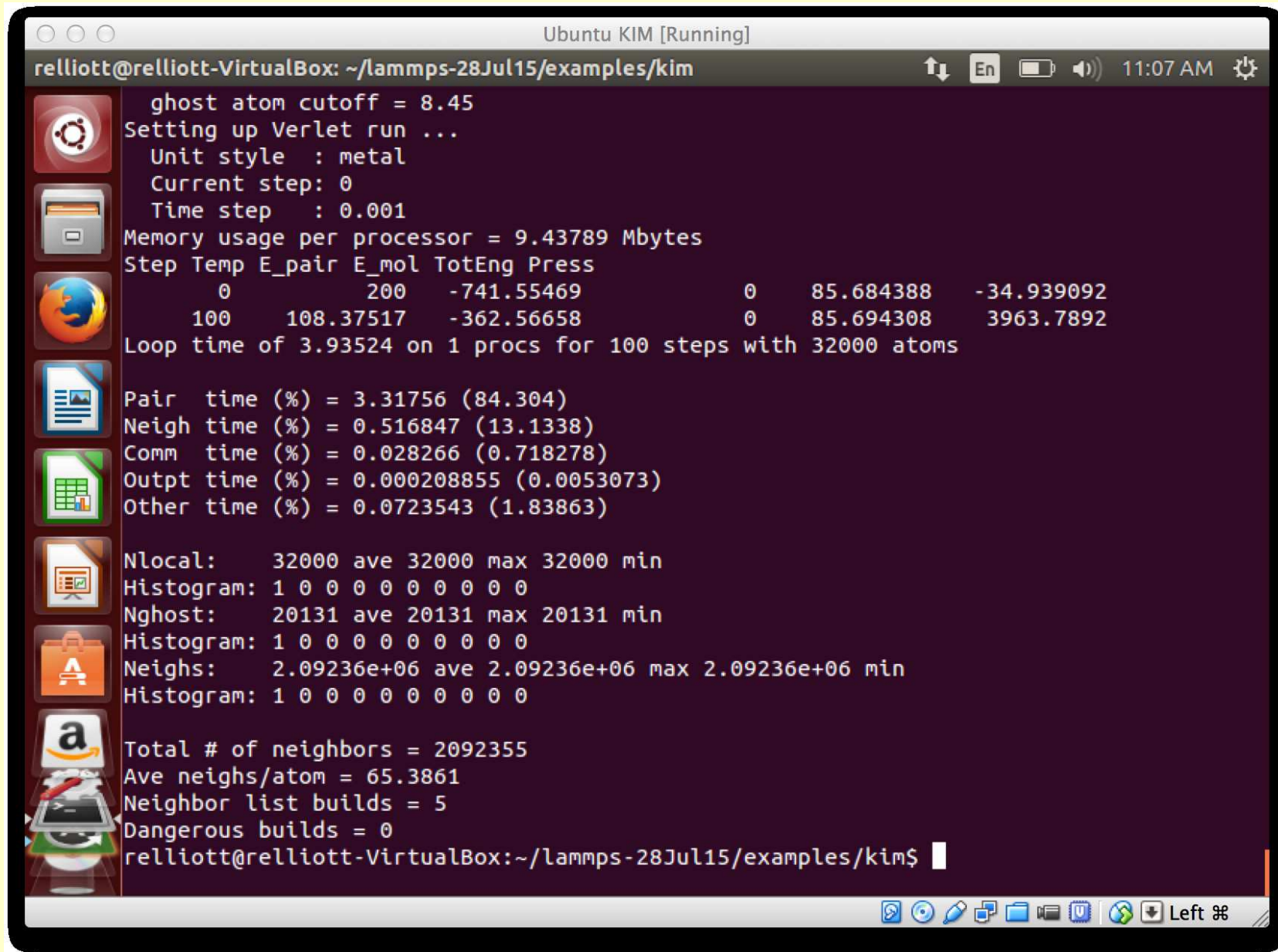


# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/lammps-28Jul15/examples/kim
variable          zz equal 20*$z
units              metal
atom_style         atomic
lattice            fcc 4.4300
region             box block 0 ${xx} 0 ${yy} 0 ${zz}
create_box         1 box
create_atoms       1 box
#pair_style        lj/cut 8.1500
#pair_coeff         1 1 0.0031 2.7400
#pair_modify       shift yes
pair_style         kim KIMvirial ex_model_Ne_P_fastLJ
pair_coeff         * * Ne
mass               1 20.18
velocity           all create 200.0 232345 loop geom
neighbor           0.3 bin
neigh_modify       delay 0 every 1 check yes
fix                1 all nve
#fix               1 all npt temp 1.0 1.0 1.0 iso 1.0 1.0 3.0
run                100
relliott@relliott-VirtualBox:~/lammps-28Jul15/examples/kim$ ../../src/lmp ubuntu -in
in.kim.lj
```

# OpenKIM with LAMMPS



```
Ubuntu KIM [Running]
relliott@relliott-VirtualBox: ~/lammps-28Jul15/examples/kim

ghost atom cutoff = 8.45
Setting up Verlet run ...
Unit style : metal
Current step: 0
Time step : 0.001
Memory usage per processor = 9.43789 Mbytes
Step Temp E_pair E_mol TotEng Press
    0         200   -741.55469          0   85.684388   -34.939092
   100    108.37517  -362.56658          0   85.694308   3963.7892
Loop time of 3.93524 on 1 procs for 100 steps with 32000 atoms

Pair time (%) = 3.31756 (84.304)
Neigh time (%) = 0.516847 (13.1338)
Comm time (%) = 0.028266 (0.718278)
Outpt time (%) = 0.000208855 (0.0053073)
Other time (%) = 0.0723543 (1.83863)

Nlocal:      32000 ave 32000 max 32000 min
Histogram:  1 0 0 0 0 0 0 0 0 0
Nghost:      20131 ave 20131 max 20131 min
Histogram:  1 0 0 0 0 0 0 0 0 0
Neighs:      2.09236e+06 ave 2.09236e+06 max 2.09236e+06 min
Histogram:  1 0 0 0 0 0 0 0 0 0

Total # of neighbors = 2092355
Ave neighs/atom = 65.3861
Neighbor list builds = 5
Dangerous builds = 0
relliott@relliott-VirtualBox:~/lammps-28Jul15/examples/kim$
```

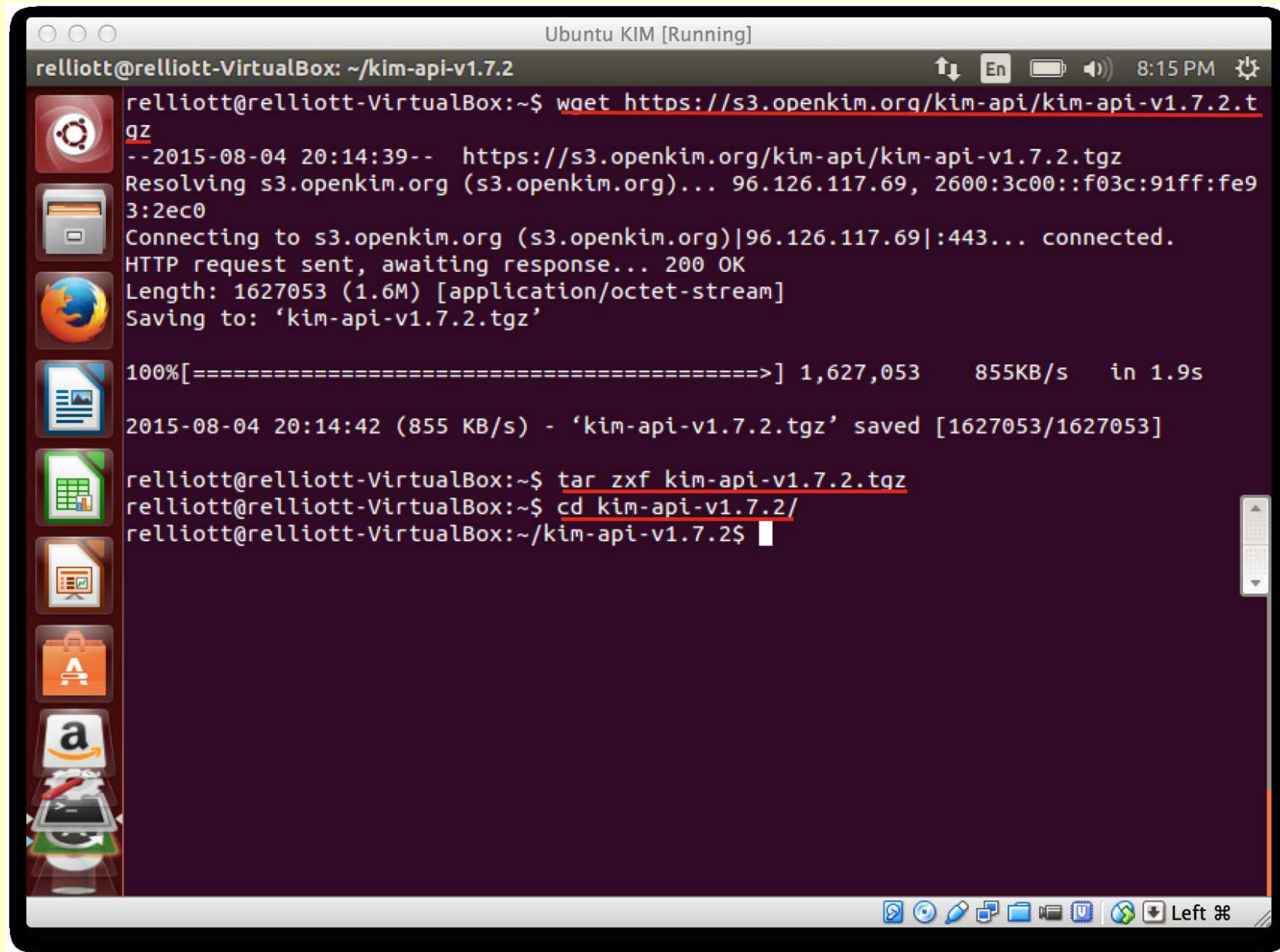


# OUTLINE

1. The Knowledgebase of Interatomic Models
  - User Signup
  - KIM API
  - “KIM Compliant” Simulators
  - Setup VirtualBox Ubuntu VM
  - KIM Model performance within LAMMPS
  - KIM Model pages
2. Installing the KIM API on Ubuntu (ppa:openkim/openkim)
3. Using LAMMPS with the KIM API

Appendix: **Installing the KIM API by hand**

# OpenKIM with LAMMPS



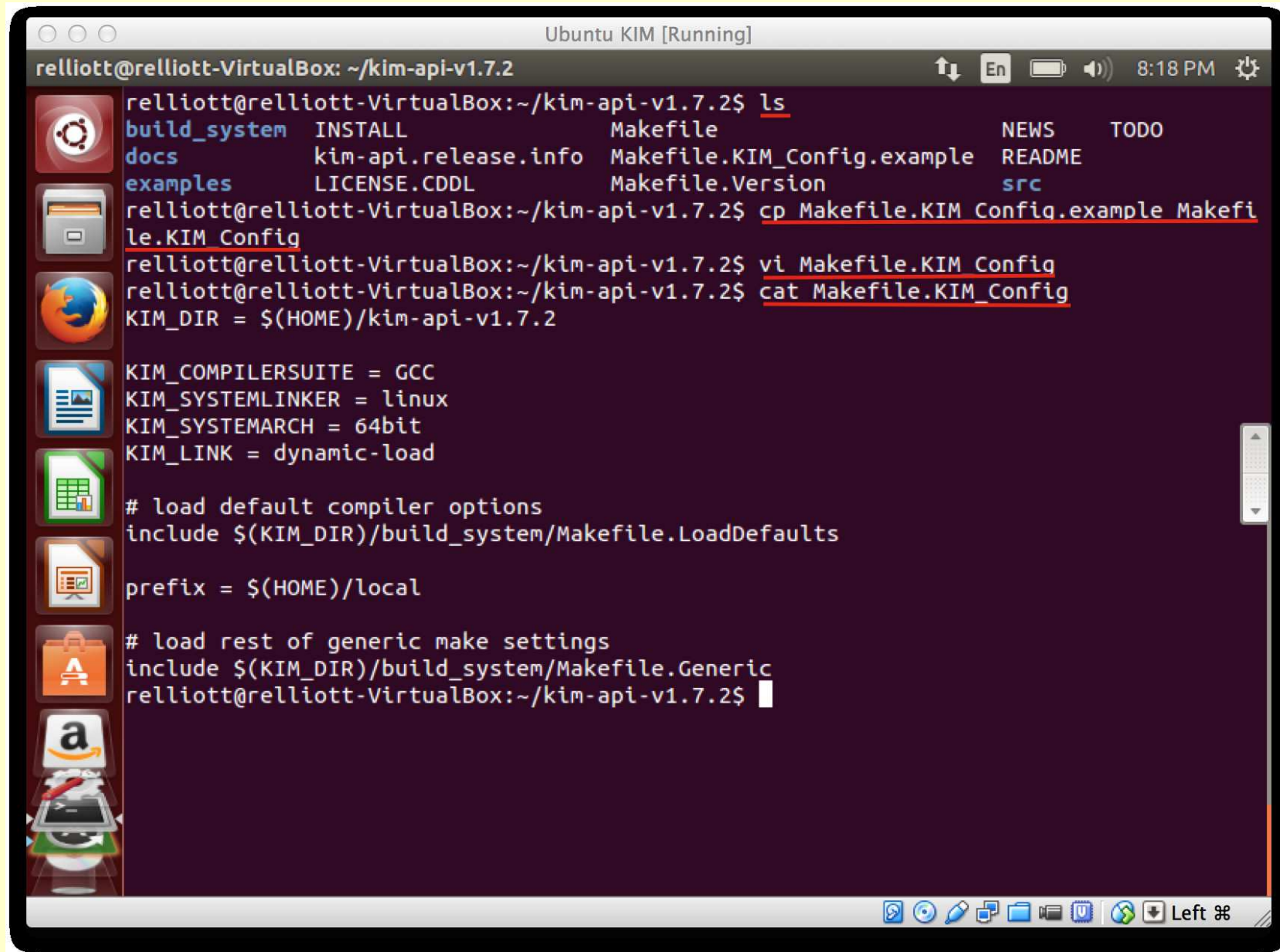
```
reliott@reliott-VirtualBox: ~/kim-api-v1.7.2
reliott@reliott-VirtualBox:~$ wget https://s3.openkim.org/kim-api/kim-api-v1.7.2.tgz
--2015-08-04 20:14:39-- https://s3.openkim.org/kim-api/kim-api-v1.7.2.tgz
Resolving s3.openkim.org (s3.openkim.org)... 96.126.117.69, 2600:3c00::f03c:91ff:fe93:2ec0
Connecting to s3.openkim.org (s3.openkim.org)|96.126.117.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1627053 (1.6M) [application/octet-stream]
Saving to: 'kim-api-v1.7.2.tgz'

100%[=====] 1,627,053 855KB/s in 1.9s

2015-08-04 20:14:42 (855 KB/s) - 'kim-api-v1.7.2.tgz' saved [1627053/1627053]

reliott@reliott-VirtualBox:~$ tar xzf kim-api-v1.7.2.tgz
reliott@reliott-VirtualBox:~$ cd kim-api-v1.7.2/
reliott@reliott-VirtualBox:~/kim-api-v1.7.2$
```

# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ ls
build_system  INSTALL          Makefile          NEWS          TODO
docs          kim-api.release.info  Makefile.KIM_Config.example  README
examples      LICENSE.CDDL      Makefile.Version  src
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ cp Makefile.KIM_Config.example Makefile.KIM_Config
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ vi Makefile.KIM_Config
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ cat Makefile.KIM_Config
KIM_DIR = $(HOME)/kim-api-v1.7.2

KIM_COMPILERSUITE = GCC
KIM_SYSTEMLINKER = linux
KIM_SYSTEMARCH = 64bit
KIM_LINK = dynamic-load

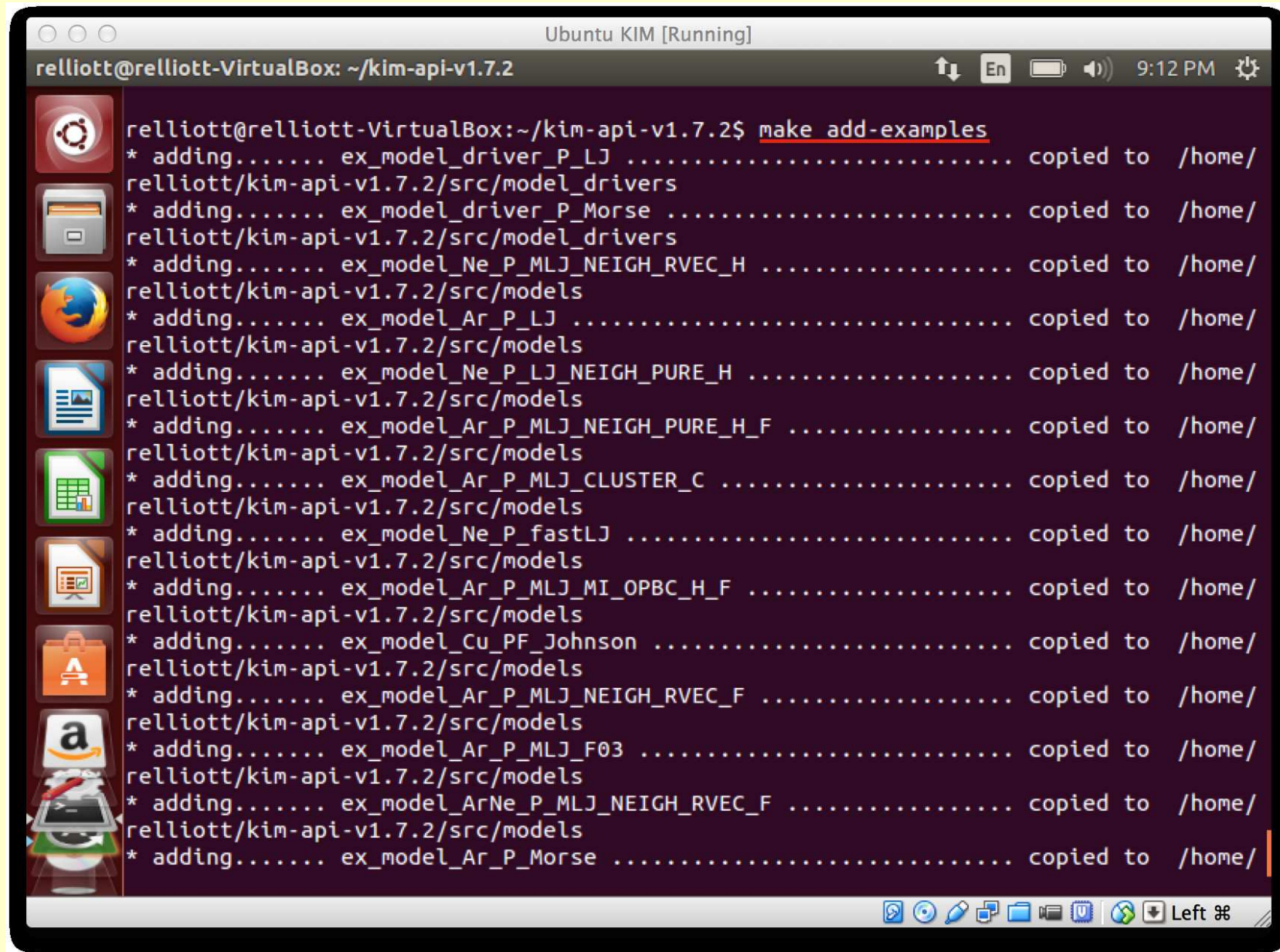
# load default compiler options
include $(KIM_DIR)/build_system/Makefile.LoadDefaults

prefix = $(HOME)/local

# load rest of generic make settings
include $(KIM_DIR)/build_system/Makefile.Generic
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$
```



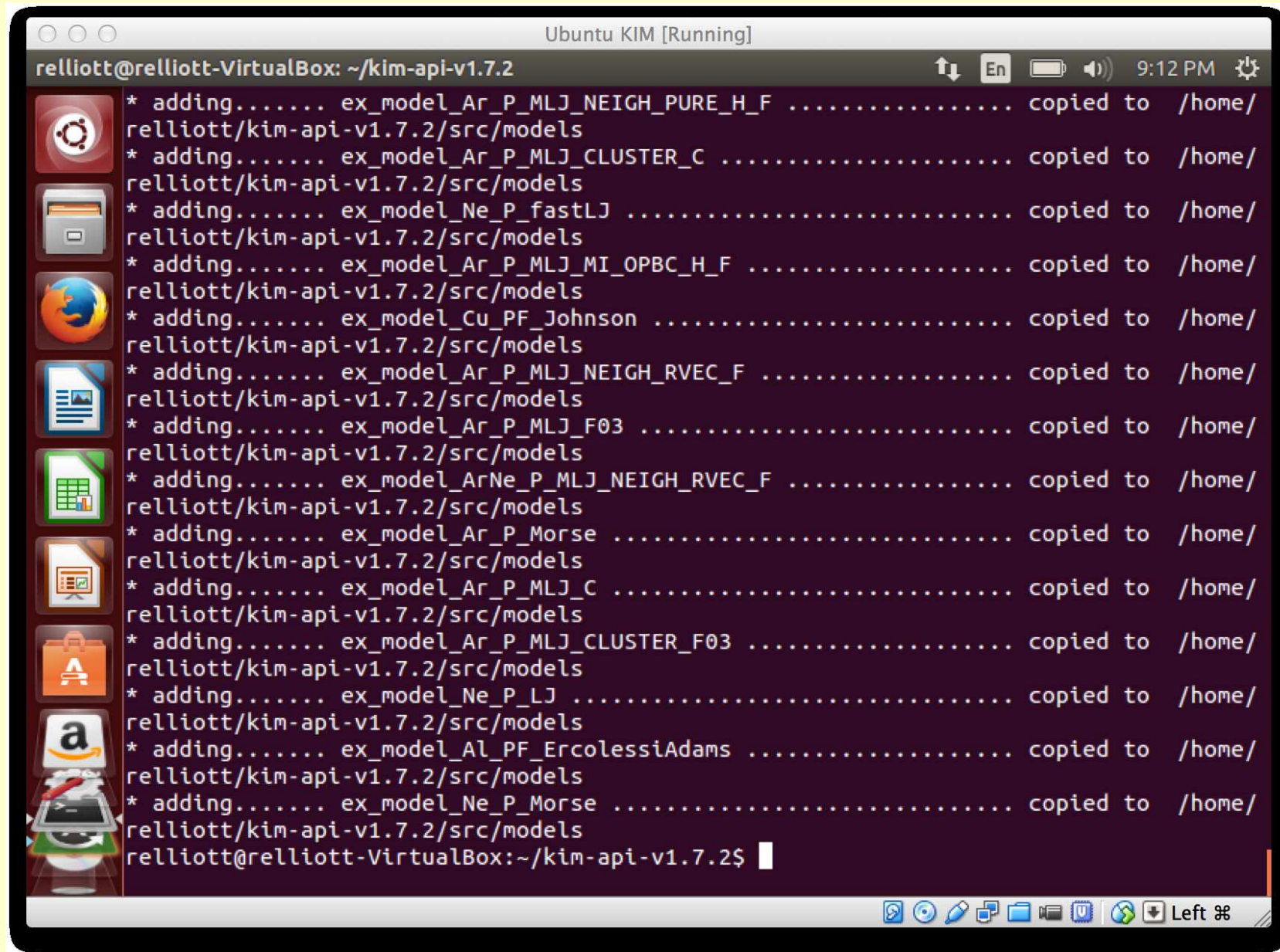
# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ make add-examples
* adding..... ex_model_driver_P_LJ ..... copied to /home/
relliott/kim-api-v1.7.2/src/model_drivers
* adding..... ex_model_driver_P_Morse ..... copied to /home/
relliott/kim-api-v1.7.2/src/model_drivers
* adding..... ex_model_Ne_P_MLJ_NEIGH_RVEC_H ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_LJ ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ne_P_LJ_NEIGH_PURE_H ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_NEIGH_PURE_H_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_CLUSTER_C ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ne_P_fastLJ ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_MI_OPBC_H_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Cu_PF_Johnson ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_NEIGH_RVEC_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_F03 ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_ArNe_P_MLJ_NEIGH_RVEC_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_Morse ..... copied to /home/
```

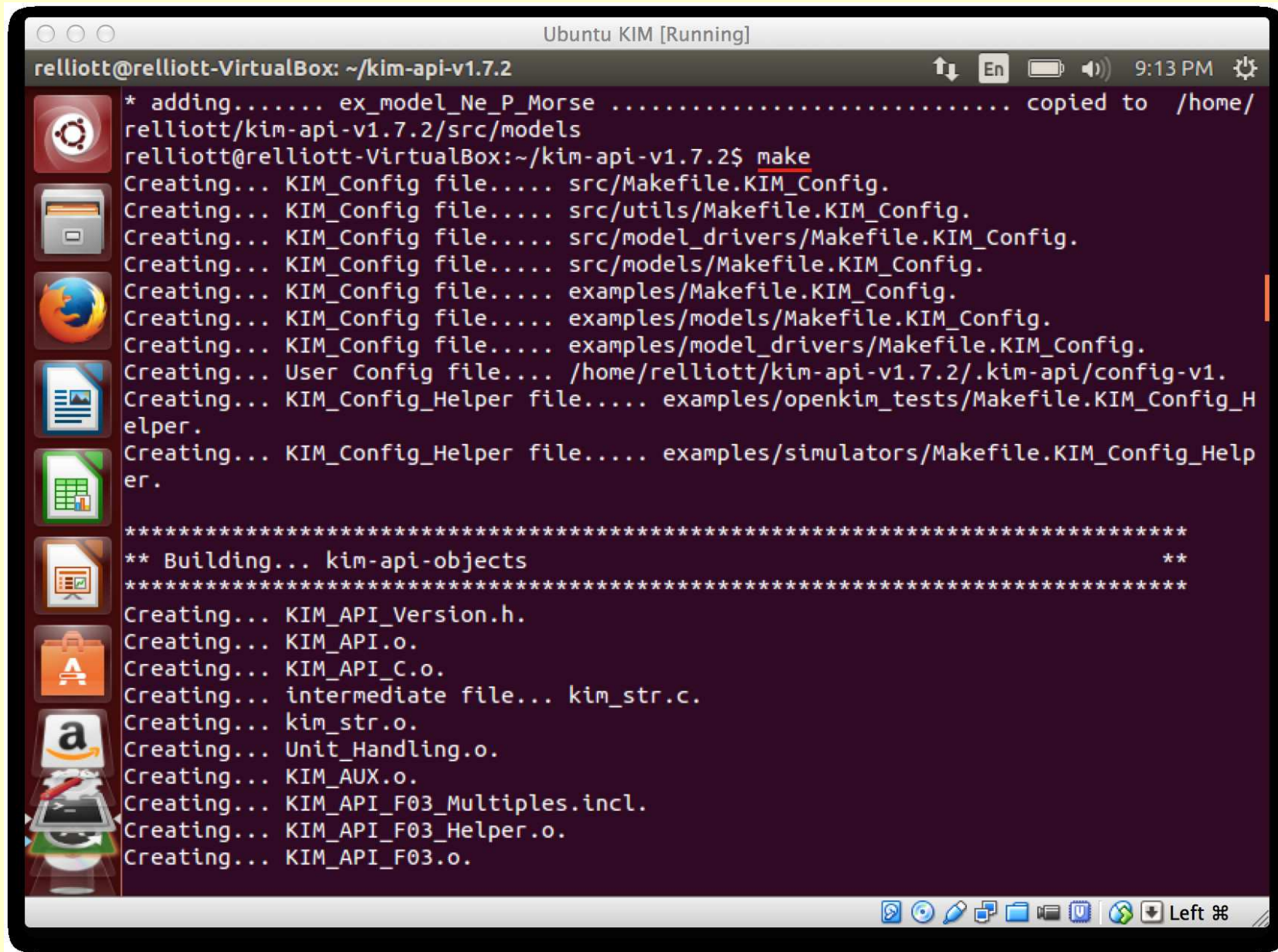


# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2
* adding..... ex_model_Ar_P_MLJ_NEIGH_PURE_H_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_CLUSTER_C ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ne_P_fastLJ ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_MI_OPBC_H_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Cu_PF_Johnson ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_NEIGH_RVEC_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_F03 ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_ArNe_P_MLJ_NEIGH_RVEC_F ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_Morse ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_C ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ar_P_MLJ_CLUSTER_F03 ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ne_P_LJ ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Al_PF_ErcolessiAdams ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
* adding..... ex_model_Ne_P_Morse ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$
```

# OpenKIM with LAMMPS

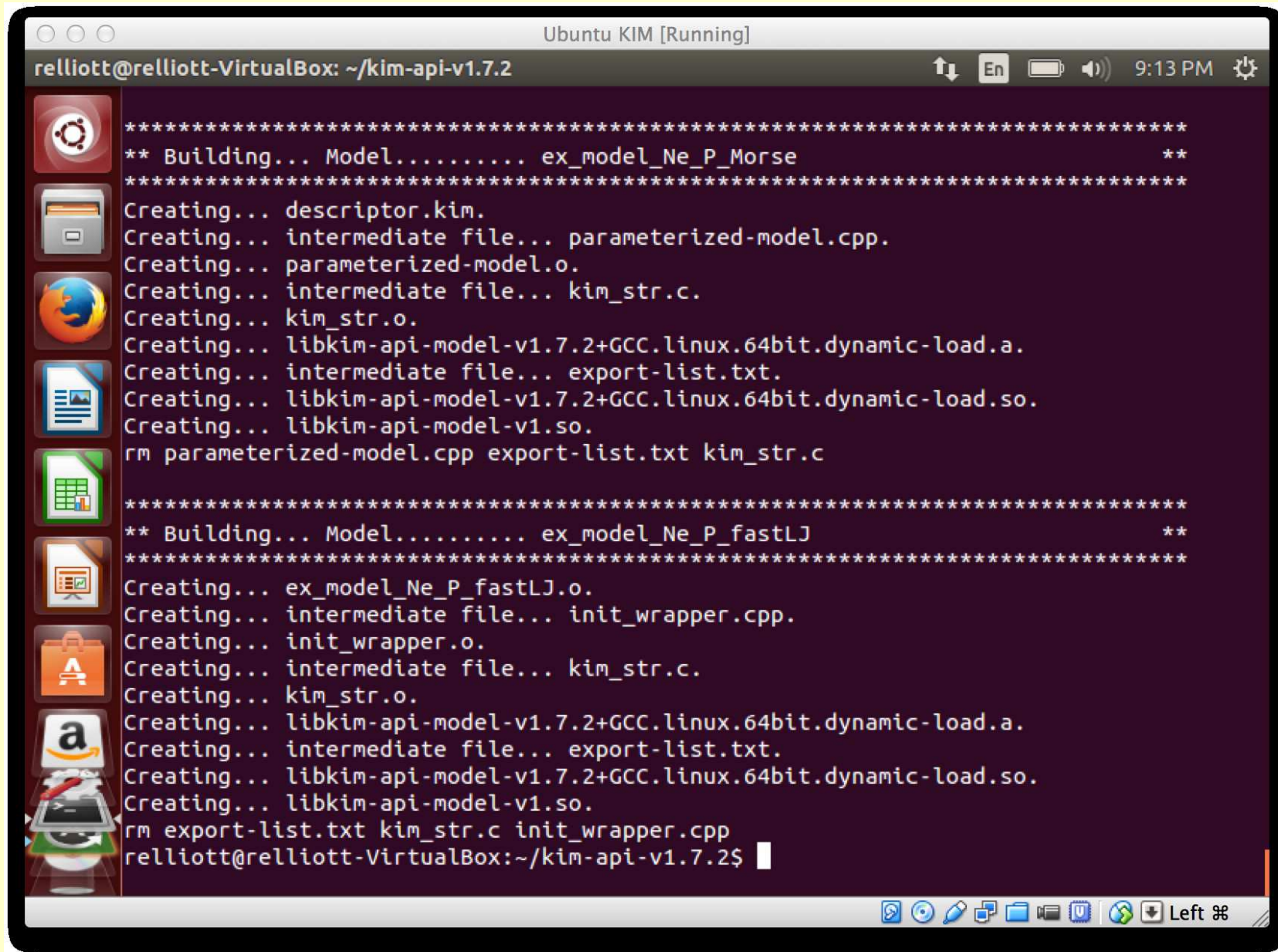


```
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2
* adding..... ex_model_Ne_P_Morse ..... copied to /home/
relliott/kim-api-v1.7.2/src/models
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ make
Creating... KIM_Config file.... src/Makefile.KIM_Config.
Creating... KIM_Config file.... src/utils/Makefile.KIM_Config.
Creating... KIM_Config file.... src/model_drivers/Makefile.KIM_Config.
Creating... KIM_Config file.... src/models/Makefile.KIM_Config.
Creating... KIM_Config file.... examples/Makefile.KIM_Config.
Creating... KIM_Config file.... examples/models/Makefile.KIM_Config.
Creating... KIM_Config file.... examples/model_drivers/Makefile.KIM_Config.
Creating... User Config file.... /home/relliott/kim-api-v1.7.2/.kim-api/config-v1.
Creating... KIM_Config_Helper file.... examples/openkim_tests/Makefile.KIM_Config_H
elper.
Creating... KIM_Config_Helper file.... examples/simulators/Makefile.KIM_Config_Help
er.

*****
** Building... kim-api-objects
*****
Creating... KIM_API_Version.h.
Creating... KIM_API.o.
Creating... KIM_API_C.o.
Creating... intermediate file... kim_str.c.
Creating... kim_str.o.
Creating... Unit_Handling.o.
Creating... KIM_AUX.o.
Creating... KIM_API_F03_Multiples.incl.
Creating... KIM_API_F03_Helper.o.
Creating... KIM_API_F03.o.
```



# OpenKIM with LAMMPS



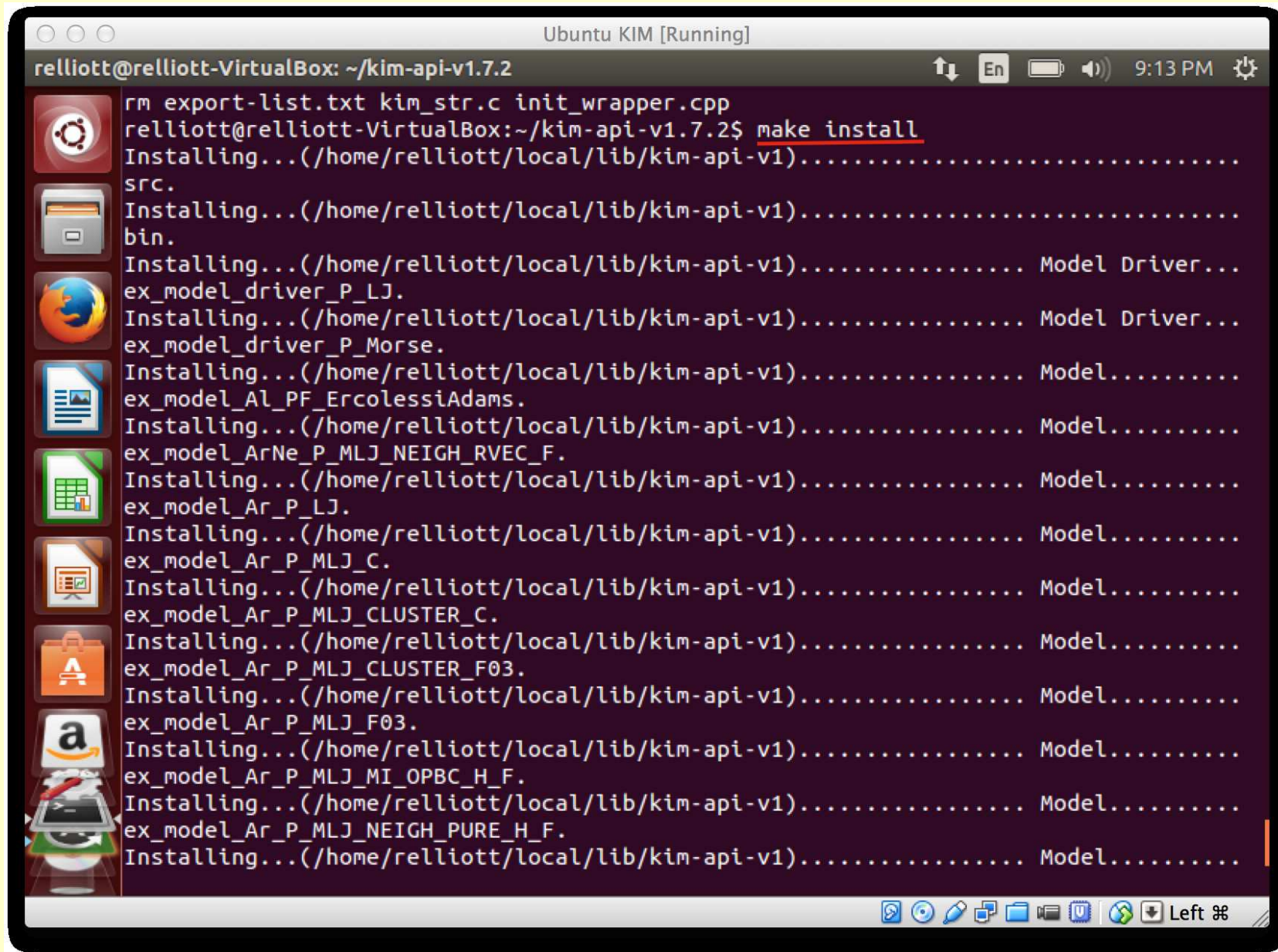
```
Ubuntu KIM [Running]
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2

*****
** Building... Model..... ex_model_Ne_P_Morse **
*****
Creating... descriptor.kim.
Creating... intermediate file... parameterized-model.cpp.
Creating... parameterized-model.o.
Creating... intermediate file... kim_str.c.
Creating... kim_str.o.
Creating... libkim-api-model-v1.7.2+GCC.linux.64bit.dynamic-load.a.
Creating... intermediate file... export-list.txt.
Creating... libkim-api-model-v1.7.2+GCC.linux.64bit.dynamic-load.so.
Creating... libkim-api-model-v1.so.
rm parameterized-model.cpp export-list.txt kim_str.c

*****
** Building... Model..... ex_model_Ne_P_fastLJ **
*****
Creating... ex_model_Ne_P_fastLJ.o.
Creating... intermediate file... init_wrapper.cpp.
Creating... init_wrapper.o.
Creating... intermediate file... kim_str.c.
Creating... kim_str.o.
Creating... libkim-api-model-v1.7.2+GCC.linux.64bit.dynamic-load.a.
Creating... intermediate file... export-list.txt.
Creating... libkim-api-model-v1.7.2+GCC.linux.64bit.dynamic-load.so.
Creating... libkim-api-model-v1.so.
rm export-list.txt kim_str.c init_wrapper.cpp
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$
```



# OpenKIM with LAMMPS



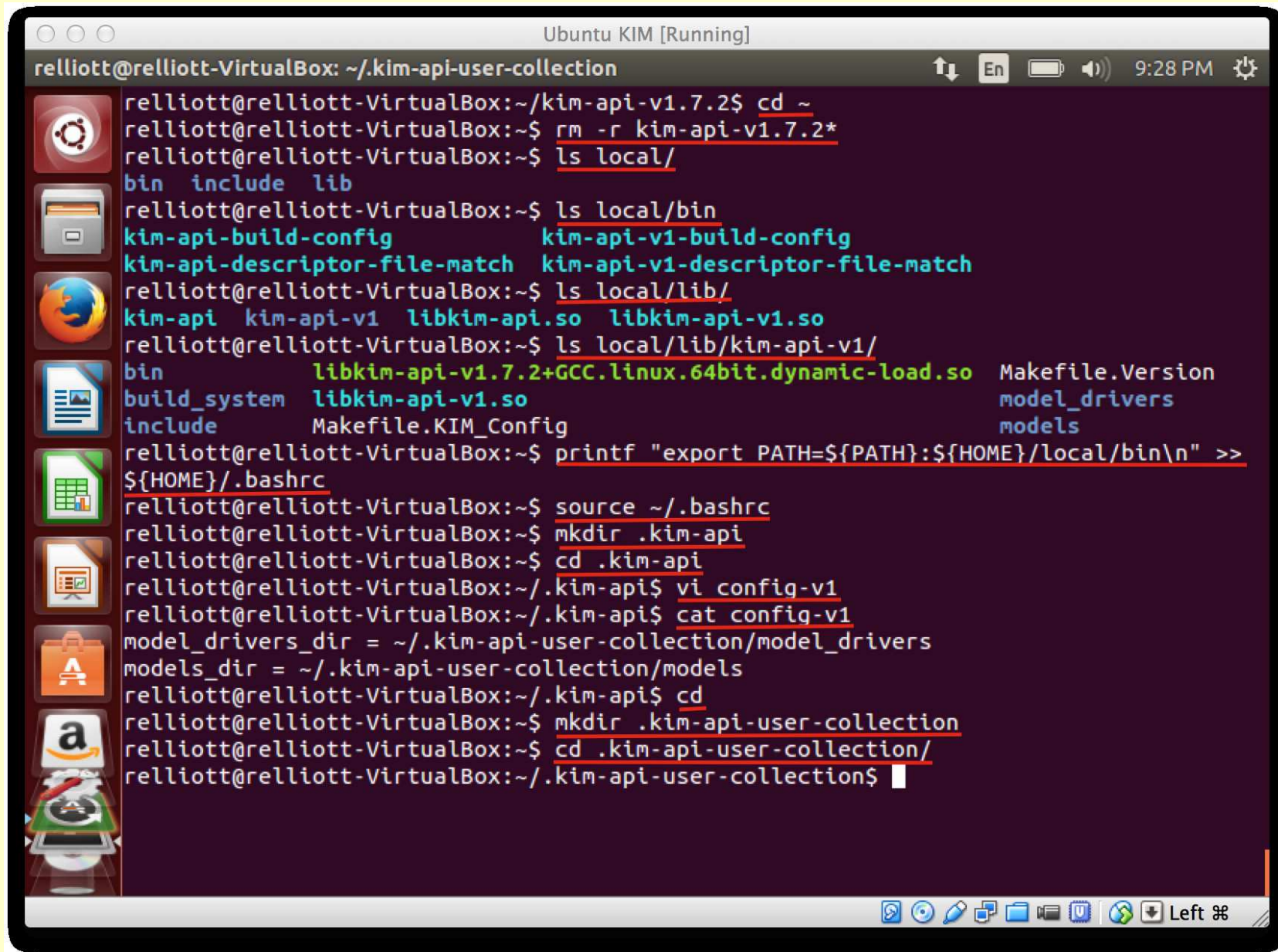
```
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2
rm export-list.txt kim_str.c init_wrapper.cpp
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ make install
Installing...(/home/relliott/local/lib/kim-api-v1).....
src.
Installing...(/home/relliott/local/lib/kim-api-v1).....
bin.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model Driver...
ex_model_driver_P_LJ.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model Driver...
ex_model_driver_P_Morse.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Al_PF_ErcolessiAdams.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_ArNe_P_MLJ_NEIGH_RVEC_F.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_LJ.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_C.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_CLUSTER_C.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_CLUSTER_F03.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_F03.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_MI_OPBC_H_F.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_NEIGH_PURE_H_F.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
```

# OpenKIM with LAMMPS

```
Ubuntu KIM [Running]
relliott@relliott-VirtualBox: ~/kim-api-v1.7.2
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_CLUSTER_F03.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_F03.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_MI_OPBC_H_F.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_NEIGH_PURE_H_F.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_MLJ_NEIGH_RVEC_F.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ar_P_Morse.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Cu_PF_Johnson.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ne_P_LJ.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ne_P_LJ_NEIGH_PURE_H.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ne_P_MLJ_NEIGH_RVEC_H.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ne_P_Morse.
Installing...(/home/relliott/local/lib/kim-api-v1)..... Model.....
ex_model_Ne_P_fastLJ.
Installing...(/home/relliott/local/lib/kim-api-v1).....
KIM_Config files.
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ make install-set-default-to-v1
Setting default kim-api to kim-api-v1
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$
```



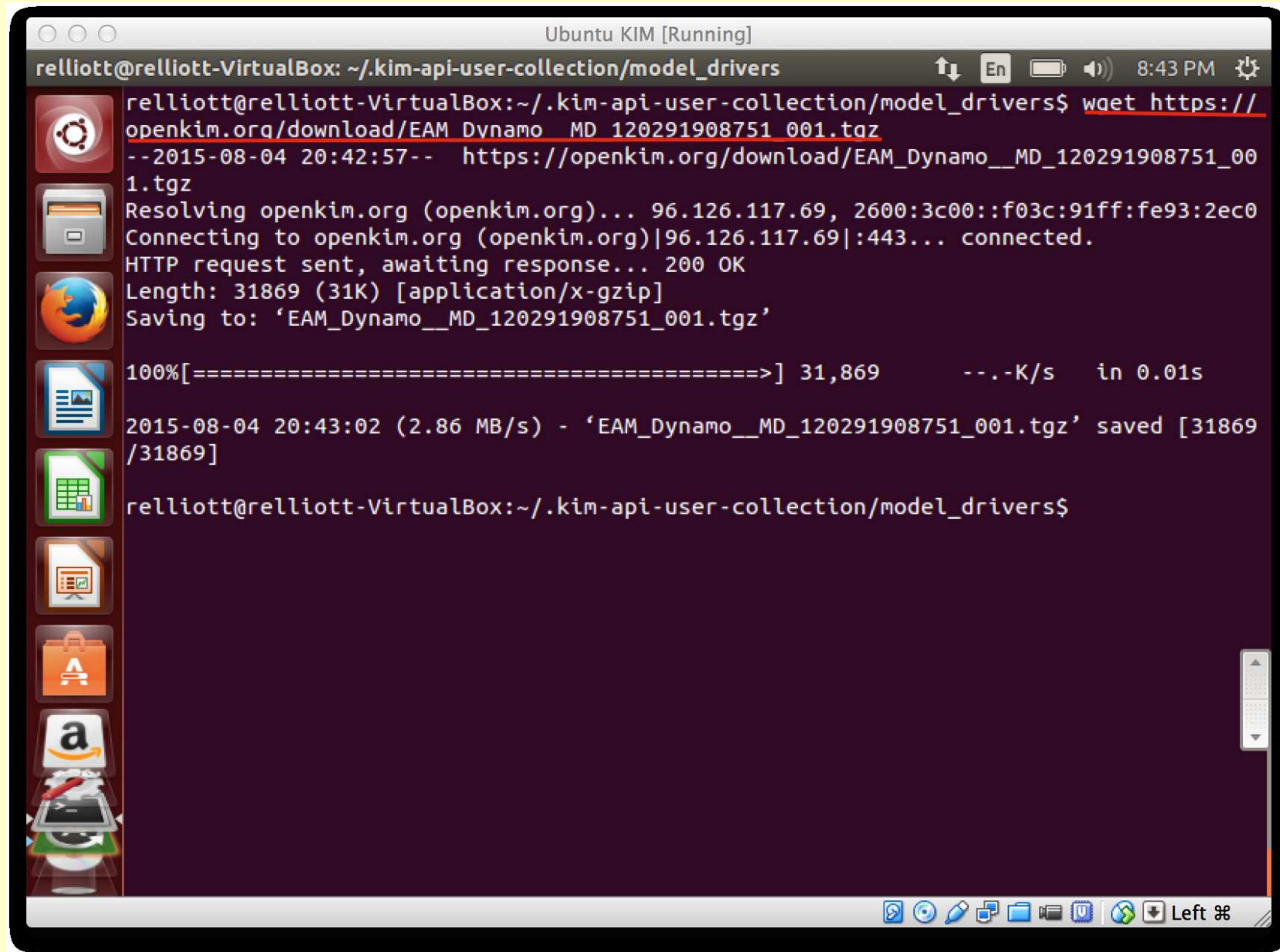
# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/.kim-api-user-collection
relliott@relliott-VirtualBox:~/kim-api-v1.7.2$ cd ~
relliott@relliott-VirtualBox:~$ rm -r kim-api-v1.7.2*
relliott@relliott-VirtualBox:~$ ls local/
bin  include  lib
relliott@relliott-VirtualBox:~$ ls local/bin
kim-api-build-config      kim-api-v1-build-config
kim-api-descriptor-file-match  kim-api-v1-descriptor-file-match
relliott@relliott-VirtualBox:~$ ls local/lib/
kim-api  kim-api-v1  libkim-api.so  libkim-api-v1.so
relliott@relliott-VirtualBox:~$ ls local/lib/kim-api-v1/
bin          libkim-api-v1.7.2+GCC.linux.64bit.dynamic-load.so  Makefile.Version
build_system libkim-api-v1.so                                model_drivers
include      Makefile.KIM_Config                          models
relliott@relliott-VirtualBox:~$ printf "export PATH=${PATH}:${HOME}/local/bin\n" >>
${HOME}/.bashrc
relliott@relliott-VirtualBox:~$ source ~/.bashrc
relliott@relliott-VirtualBox:~$ mkdir .kim-api
relliott@relliott-VirtualBox:~$ cd .kim-api
relliott@relliott-VirtualBox:~/.kim-api$ vi config-v1
relliott@relliott-VirtualBox:~/.kim-api$ cat config-v1
model_drivers_dir = ~/.kim-api-user-collection/model_drivers
models_dir = ~/.kim-api-user-collection/models
relliott@relliott-VirtualBox:~/.kim-api$ cd
relliott@relliott-VirtualBox:~$ mkdir .kim-api-user-collection
relliott@relliott-VirtualBox:~$ cd .kim-api-user-collection/
relliott@relliott-VirtualBox:~/.kim-api-user-collection$
```



# OpenKIM with LAMMPS



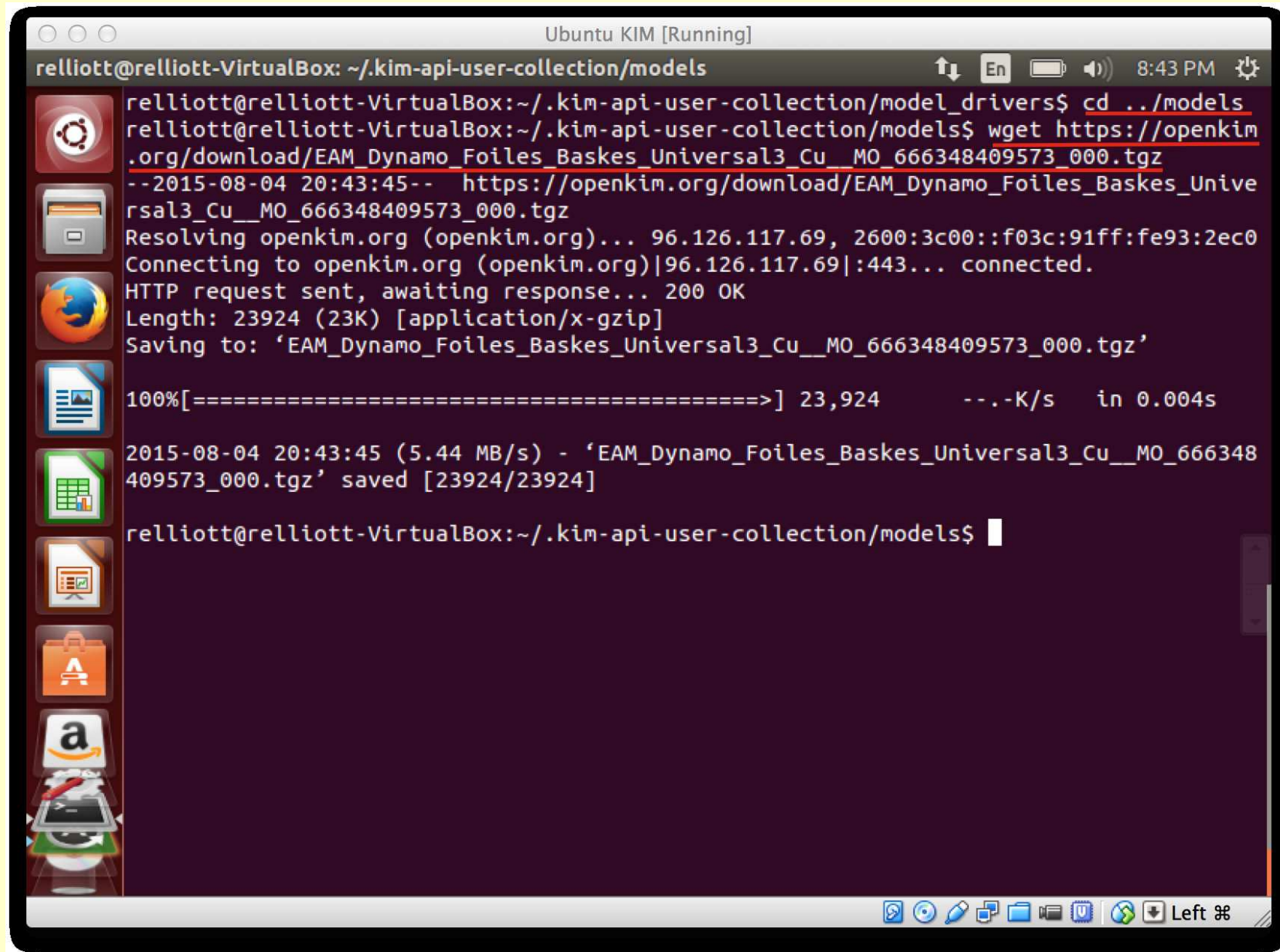
```
Ubuntu KIM [Running]
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/model_drivers
relliott@relliott-VirtualBox:~/.kim-api-user-collection/model_drivers$ wget https://
openkim.org/download/EAM_Dynamo_MD_120291908751_001.tgz
--2015-08-04 20:42:57-- https://openkim.org/download/EAM_Dynamo__MD_120291908751_00
1.tgz
Resolving openkim.org (openkim.org)... 96.126.117.69, 2600:3c00::f03c:91ff:fe93:2ec0
Connecting to openkim.org (openkim.org)|96.126.117.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31869 (31K) [application/x-gzip]
Saving to: 'EAM_Dynamo__MD_120291908751_001.tgz'

100%[=====] 31,869 ---K/s in 0.01s

2015-08-04 20:43:02 (2.86 MB/s) - 'EAM_Dynamo__MD_120291908751_001.tgz' saved [31869
/31869]

relliott@relliott-VirtualBox:~/.kim-api-user-collection/model_drivers$
```

# OpenKIM with LAMMPS



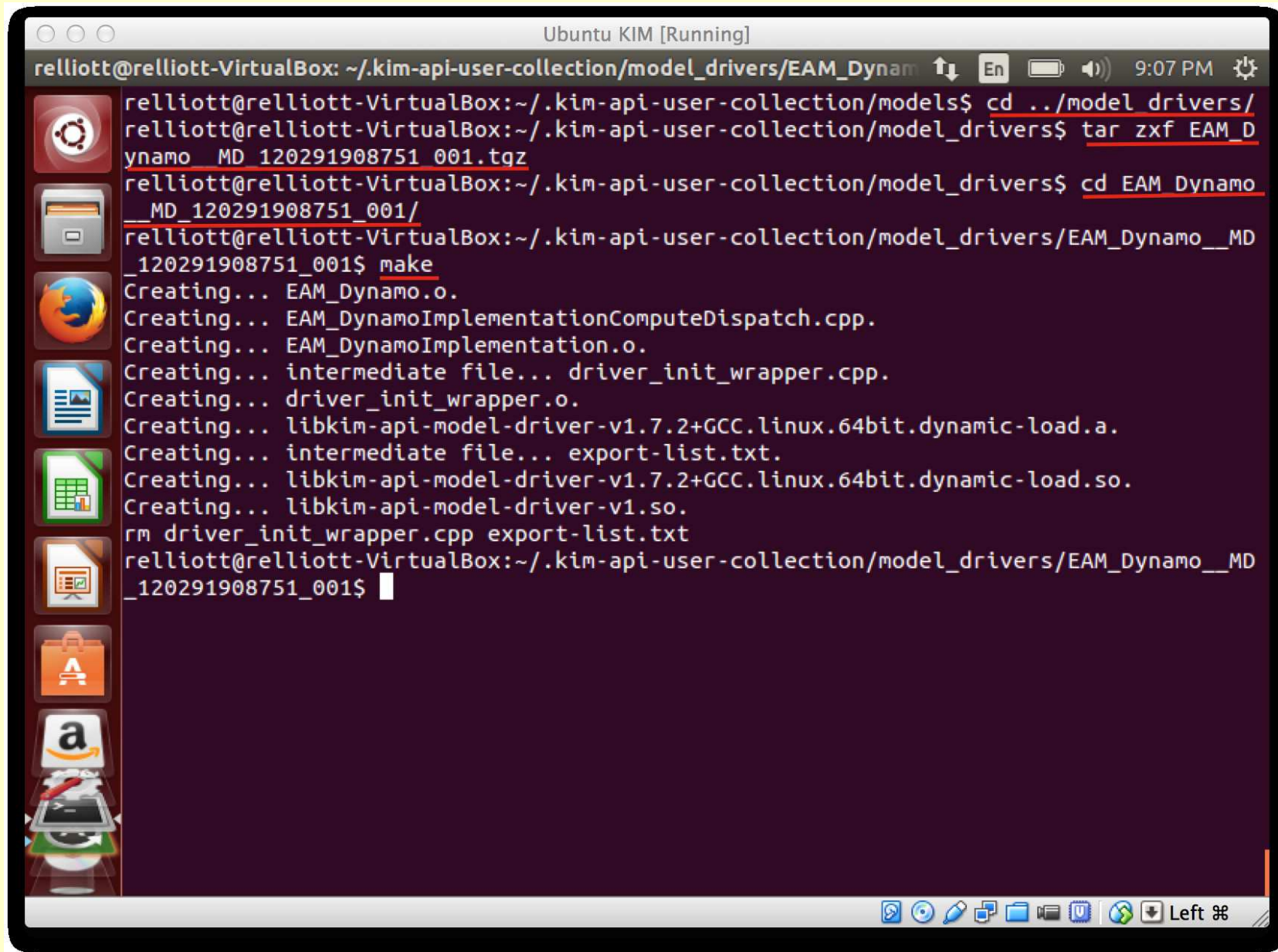
```
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/models
relliott@relliott-VirtualBox:~/.kim-api-user-collection/model_drivers$ cd ../models
relliott@relliott-VirtualBox:~/.kim-api-user-collection/models$ wget https://openkim.org/download/EAM_Dynamo_Foiles_Baskes_Universal3_Cu__MO_666348409573_000.tgz
--2015-08-04 20:43:45-- https://openkim.org/download/EAM_Dynamo_Foiles_Baskes_Universal3_Cu__MO_666348409573_000.tgz
Resolving openkim.org (openkim.org)... 96.126.117.69, 2600:3c00::f03c:91ff:fe93:2ec0
Connecting to openkim.org (openkim.org)|96.126.117.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23924 (23K) [application/x-gzip]
Saving to: 'EAM_Dynamo_Foiles_Baskes_Universal3_Cu__MO_666348409573_000.tgz'

100%[=====>] 23,924      ---K/s   in 0.004s

2015-08-04 20:43:45 (5.44 MB/s) - 'EAM_Dynamo_Foiles_Baskes_Universal3_Cu__MO_666348409573_000.tgz' saved [23924/23924]

relliott@relliott-VirtualBox:~/.kim-api-user-collection/models$
```

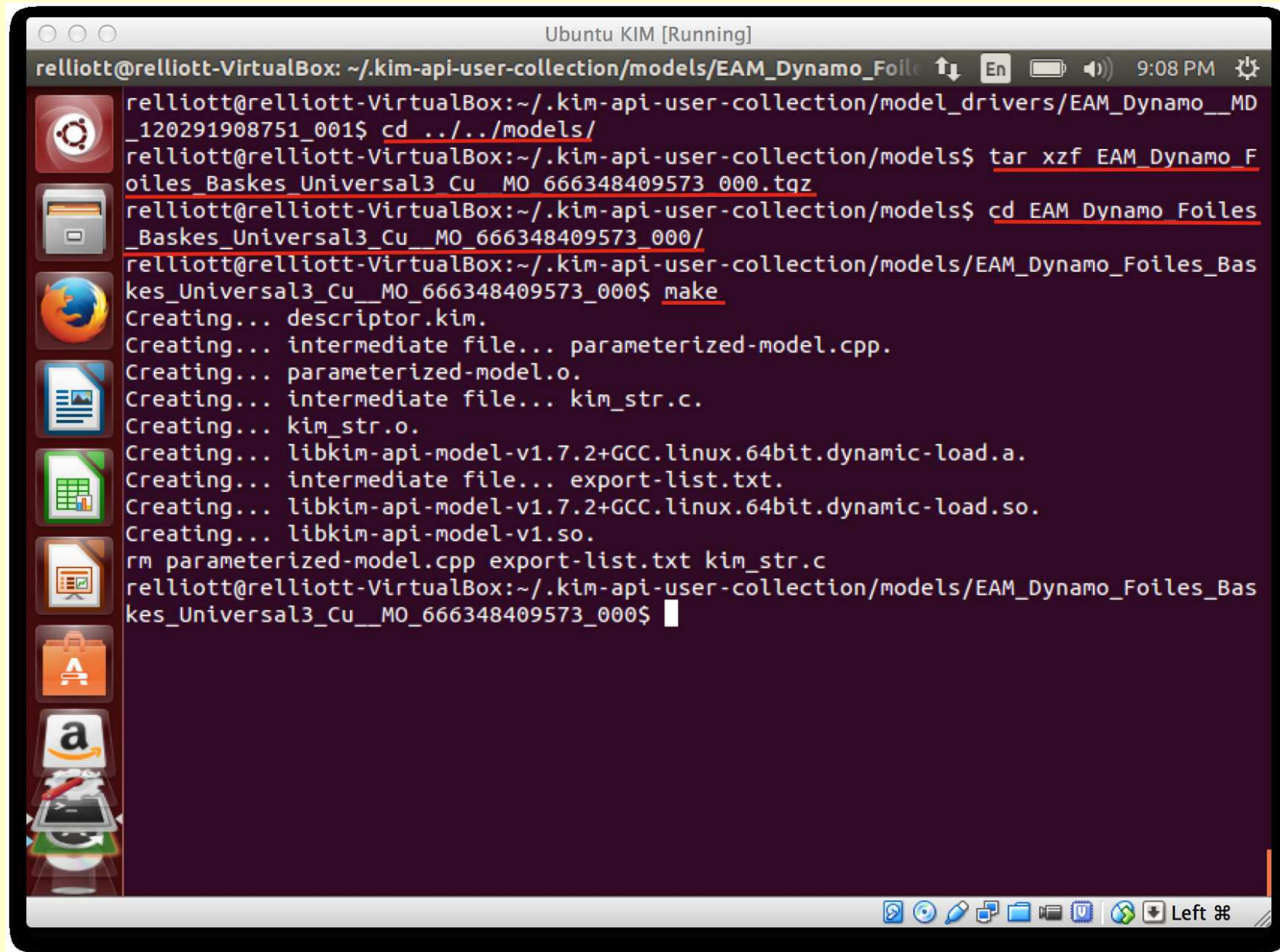
# OpenKIM with LAMMPS



```
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/model_drivers/EAM_Dynamo
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/models$ cd ../model_drivers/
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/model_drivers$ tar xzf EAM_Dynamo_MD_120291908751_001.tgz
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/model_drivers$ cd EAM_Dynamo_MD_120291908751_001/
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/model_drivers/EAM_Dynamo_MD_120291908751_001$ make
Creating... EAM_Dynamo.o.
Creating... EAM_DynamoImplementationComputeDispatch.cpp.
Creating... EAM_DynamoImplementation.o.
Creating... intermediate file... driver_init_wrapper.cpp.
Creating... driver_init_wrapper.o.
Creating... libkim-api-model-driver-v1.7.2+GCC.linux.64bit.dynamic-load.a.
Creating... intermediate file... export-list.txt.
Creating... libkim-api-model-driver-v1.7.2+GCC.linux.64bit.dynamic-load.so.
Creating... libkim-api-model-driver-v1.so.
rm driver_init_wrapper.cpp export-list.txt
relliott@relliott-VirtualBox: ~/.kim-api-user-collection/model_drivers/EAM_Dynamo_MD_120291908751_001$
```



# OpenKIM with LAMMPS



```
reliott@reliott-VirtualBox: ~/.kim-api-user-collection/models/EAM_Dynamo_Foile
reliott@reliott-VirtualBox: ~/.kim-api-user-collection/models/EAM_Dynamo__MD
_120291908751_001$ cd ../../models/
reliott@reliott-VirtualBox: ~/.kim-api-user-collection/models$ tar xzf EAM_Dynamo_F
oiles_Baskes_Universal3_Cu_MO_666348409573_000.tgz
reliott@reliott-VirtualBox: ~/.kim-api-user-collection/models$ cd EAM_Dynamo_Foiles
_Baskes_Universal3_Cu_MO_666348409573_000/
reliott@reliott-VirtualBox: ~/.kim-api-user-collection/models/EAM_Dynamo_Foiles_Bas
kes_Universal3_Cu_MO_666348409573_000$ make
Creating... descriptor.kim.
Creating... intermediate file... parameterized-model.cpp.
Creating... parameterized-model.o.
Creating... intermediate file... kim_str.c.
Creating... kim_str.o.
Creating... libkim-api-model-v1.7.2+GCC.linux.64bit.dynamic-load.a.
Creating... intermediate file... export-list.txt.
Creating... libkim-api-model-v1.7.2+GCC.linux.64bit.dynamic-load.so.
Creating... libkim-api-model-v1.so.
rm parameterized-model.cpp export-list.txt kim_str.c
reliott@reliott-VirtualBox: ~/.kim-api-user-collection/models/EAM_Dynamo_Foiles_Bas
kes_Universal3_Cu_MO_666348409573_000$
```