

# Getting Started with the SDAccel Environment on Nimbix Cloud

UG1240 (v2016.3) December 15, 2016

## Revision History

The following table shows the revision history for this document.

Date	Version	Changes
12/15/2016	2016.3	Updated supported board configurations in Introduction. Updated commands throughout Chapter 3: SDAccel Environment on Nimbix Cloud Tutorial. Updated Figure 18.
11/14/2016	2016.2	Initial Xilinx release.

# Table of Contents

Revision History.....	2
Chapter 1: SDAccel Environment on Nimbix Cloud Overview .....	4
Introduction .....	4
Chapter 2: Nimbix Account Creation, Login, and Setup.....	5
Account Creation .....	5
Nimbix Account Login .....	6
Transferring Data to and from Nimbix .....	8
/data Directory: Important Notes.....	10
Chapter 3: SDAccel Environment on Nimbix Cloud Tutorial .....	11
Step 1: Login to your Nimbix account.....	11
Step 2: Load SDAccel Examples from the GitHub site.....	12
Step 3: Design Compilation .....	15
Step 4: Terminating the Session .....	20
Step 5: Application Deployment.....	21
Appendix A: Legal Notices.....	24
Please Read: Important Legal Notices.....	24

# Chapter 1: SDAccel Environment on Nimbix Cloud

## Overview

---

## Introduction

This document demonstrates how to get started using SDAccel Development Environment in the Nimbix Cloud.

The SDAccel development environment, a member of the SDx™ family of development environments for systems and software engineers, enables up to 25X better performance/watt for data center application acceleration leveraging FPGAs.

Xilinx has partnered with Nimbix Inc. to deliver an opportunity to test drive SDAccel and see how FPGA based acceleration can accelerate your OpenCL C, C/C++ and RTL Kernels. The Nimbix Accelerated Compute Cloud provides a purpose built compute cloud for big data and computation. It now hosts the SDAccel development environment for companies looking to streamline evaluations and FPGA OpenCL kernel acceleration.

You can use multiple approaches when working with Nimbix Cloud, such as:

- Using Nimbix to compile and deploy your application with the Xilinx FPGA-based boards hosted on Nimbix.
- Compiling the application on your local machine, transferring the compiled files to Nimbix, and then deploying the application on Nimbix

This document describes how to use both approaches and contains a step-by-step tutorial to help familiarize you with using SDAccel under Nimbix environment.

**Important:** When compiling the application to be deployed on Xilinx FPGAs based boards hosted by Nimbix, you need specify one of the following board configurations:

FPGA Board	Supported DSA
Alpha Data V7	xilinx:adm-pcie-7v3:1ddr:3.0
Alpha Data KU3	xilinx:adm-pcie-ku3:2ddr-xpr:3.2
Xilinx KU115 Developer Board	xilinx:xil-accel-rd-pcie3-ku115:4ddr:3.2

## Chapter 2: Nimbix Account Creation, Login, and Setup

### Account Creation

Register for the SDAccel environment at the Nimbix website. Navigate with your browser to <https://www.nimbix.net/xilinx>.

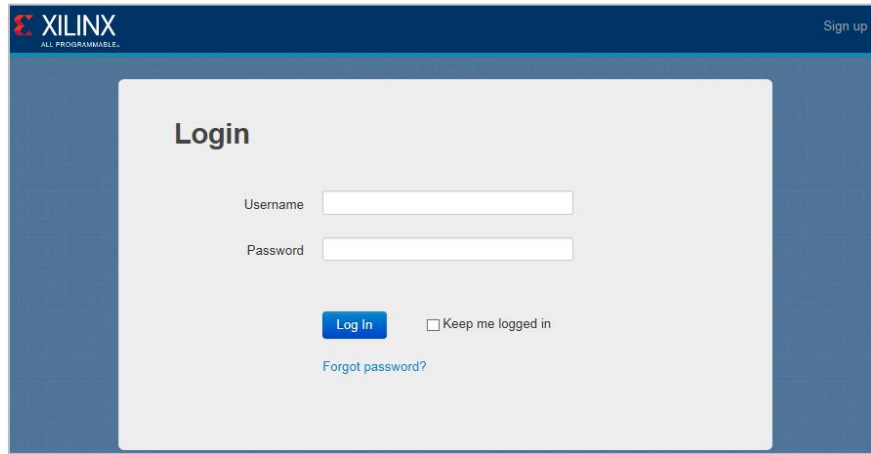


**Figure 1: Nimbix Cloud Landing Page**

Complete the registration form. Upon approval of your account, you will receive an email confirmation along with information introducing you to the SDAccel development environment.

## Nimbix Account Login

To login to your account navigate to <https://xilinx-cloud.jarvice.com/login-page>

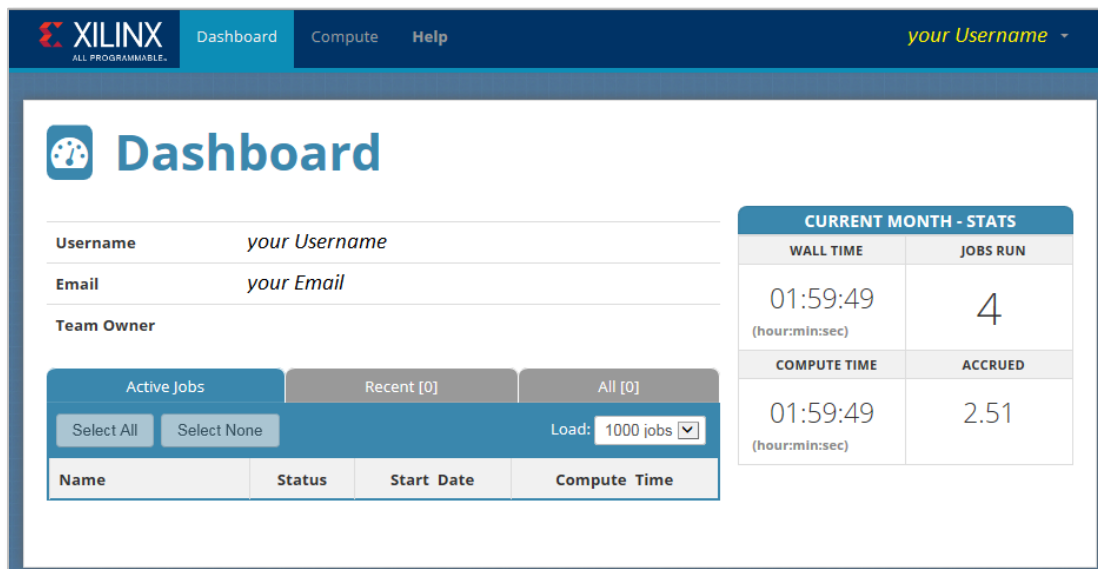


The login page features the Xilinx logo and 'ALL PROGRAMMABLE™' tagline in the top left, and a 'Sign up' link in the top right. The main content area is titled 'Login' and contains two input fields for 'Username' and 'Password'. Below these fields is a blue 'Log In' button, a checkbox for 'Keep me logged in', and a link for 'Forgot password?'.

**Figure 2: Nimbix Login Page**

On this page enter your user name and a password and then click **Log In** to login into your account.

When you login to your account, the Nimbix interface **Dashboard** opens.



The dashboard page has a top navigation bar with the Xilinx logo, 'Dashboard' (active), 'Compute', and 'Help' tabs, and a user profile dropdown showing 'your Username'. The main content area is titled 'Dashboard' and includes a user information section with fields for 'Username' (your Username), 'Email' (your Email), and 'Team Owner'. Below this is a section for 'Active Jobs' with buttons for 'Select All' and 'Select None', a 'Load' dropdown set to '1000 jobs', and a table with columns: Name, Status, Start Date, and Compute Time. To the right is a 'CURRENT MONTH - STATS' table.

CURRENT MONTH - STATS	
WALL TIME	JOBS RUN
01:59:49 (hour:min:sec)	4
COMPUTE TIME	ACCRUED
01:59:49 (hour:min:sec)	2.51

**Figure 3: Dashboard Page**

The **Dashboard** provides you information about:

- The jobs you are currently running and their status
- Previously run jobs

The **Compute** page allows you to compile and deploy your applications.

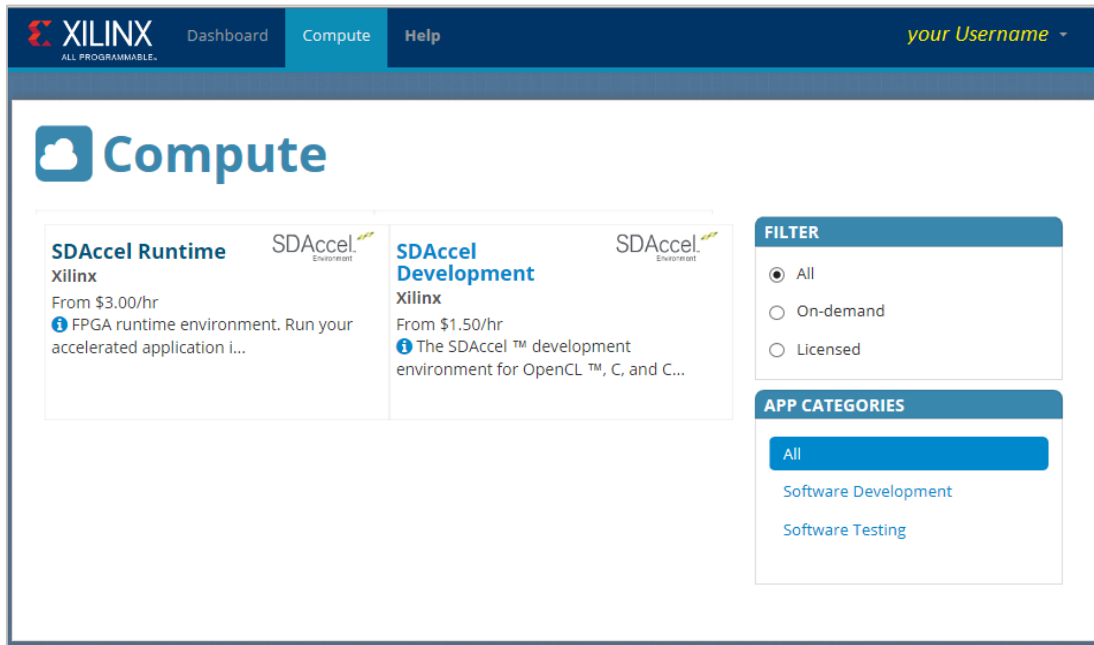
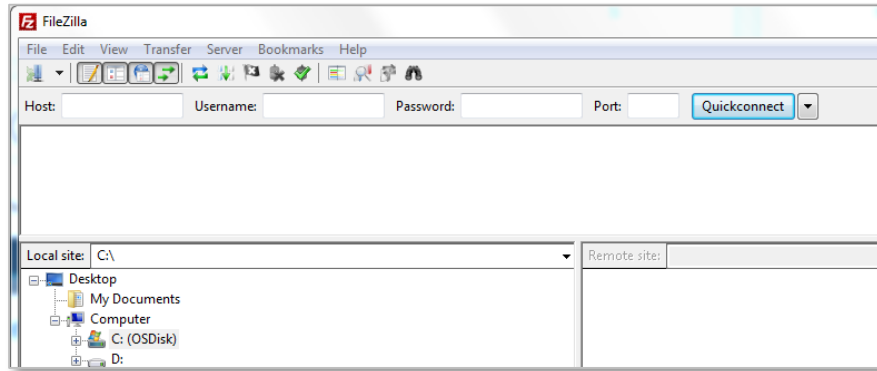


Figure 4: Compute Page

## Transferring Data to and from Nimbix

When working with SDAccel, you might need to transfer files between your machine and Nimbix. SFTP is the preferred file transfer method. The instructions below demonstrate file transfer using the [FileZilla](#) SFTP client software.

**Note:** Follow this [link](#) for more information regarding file transfers.

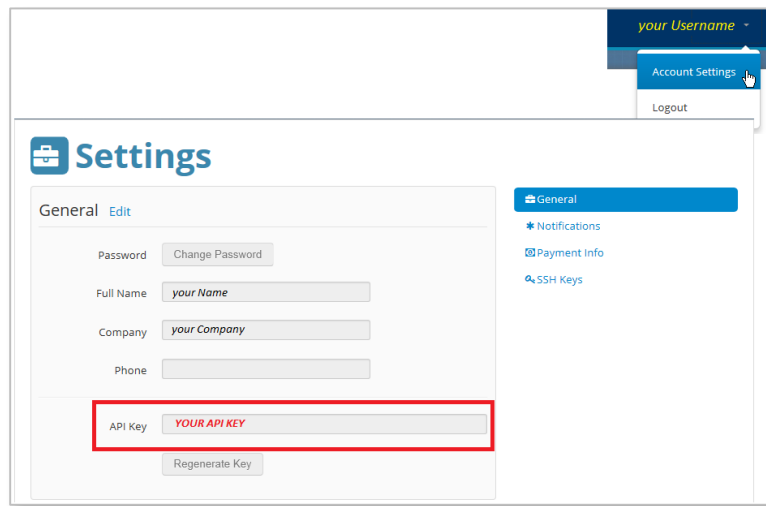


**Figure 5: FileZilla SFTP client**

1. Download and install FileZilla.
2. Open FileZilla and configure the following:
  - a. In the **Host:** field specify `drop.jarvice.com`
  - b. In the **Port:** field specify the value 22
  - c. In the **Username:** field specify your Nimbix Username
  - d. The **Password:** field should contain the your JARVICE API key

To obtain your API key, login to your Nimbix account, go to your account settings and copy the API Key from the Settings page, as shown in Figure 6.

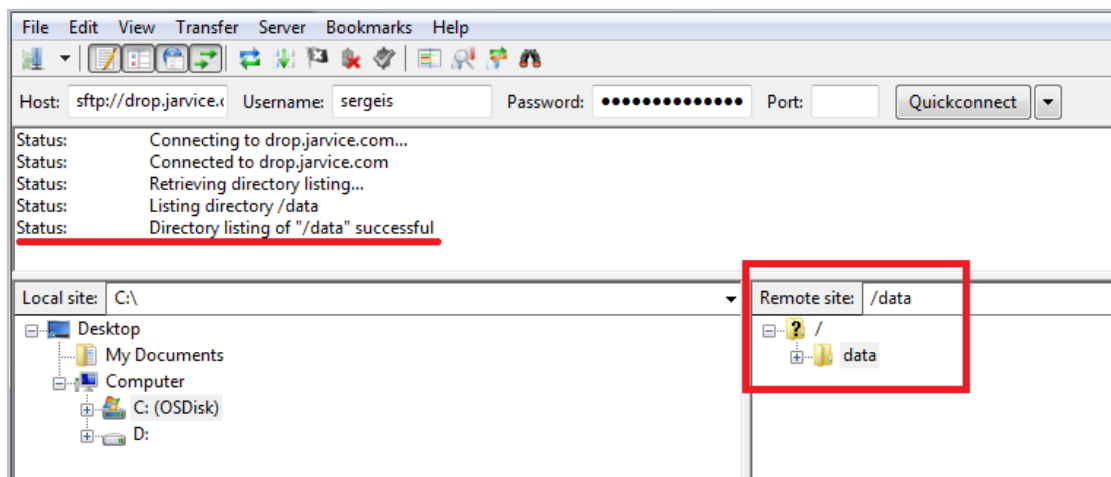




**Figure 6: API Key on the Settings Page**

3. Select the **Quickconnect** button to connect to Nimbix.

If the connection was successfully established you should see the **data** directory on the Remote site, as shown in Figure 7.



**Figure 7: Remote Site data Directory**

Using FileZilla interface, you can now start to transfer files to and from Nimbix.



**IMPORTANT:** If you work behind a company firewall and are unable to connect to Nimbix, speak with your IT department and request the access to the 22 (TCP) port.

---

## /data Directory: Important Notes

Working in the Nimbix environment, you will need to use the **/data** directory in different situations, so it is important to understand all of its features.

- All data located in the Nimbix **/data** directory is preserved after you logout from your Nimbix account. Copy critical files to the **/data** directory if you want to preserve them for later reuse. You can also create sub-directories under **/data** if necessary.
- In order to exchange data between Nimbix and your local machine, use the **/data** directory. For instance, if you need use an OpenCL located in your local machine for SDAccel compilation on Nimbix, transfer this file to the **/data** directory first and then copy from **/data** to the final folder.
- To deploy the compiled application on the FPGA boards hosted by Nimbix, copy all necessary files to the **/data** directory first. This is described later on in Step 5: Application Deployment.

## Chapter 3: SDAccel Environment on Nimbix Cloud Tutorial

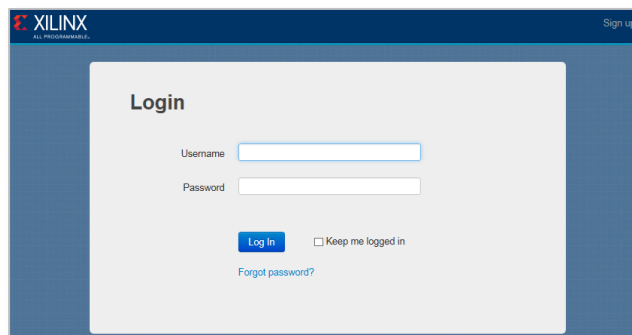
The tutorial uses the `hello` example available in the `getting_started` folder at this location:

[https://github.com/Xilinx/SDAccel\\_Examples/](https://github.com/Xilinx/SDAccel_Examples/)

---

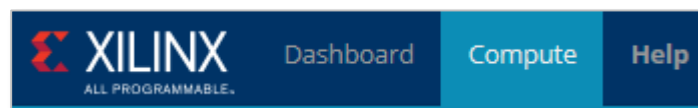
### Step 1: Login to your Nimbix account

1. Go to: <https://xilinx-cloud.jarvice.com/login-page>



**Figure 8: Nimbix Account Login**

2. Provide your Username and Password and select **Log In**.
3. In the opened page, select **Compute** to open the **Compute** page.



**Figure 9: Compute**

- a. The **Compute** page will open, as shown in Figure 10.

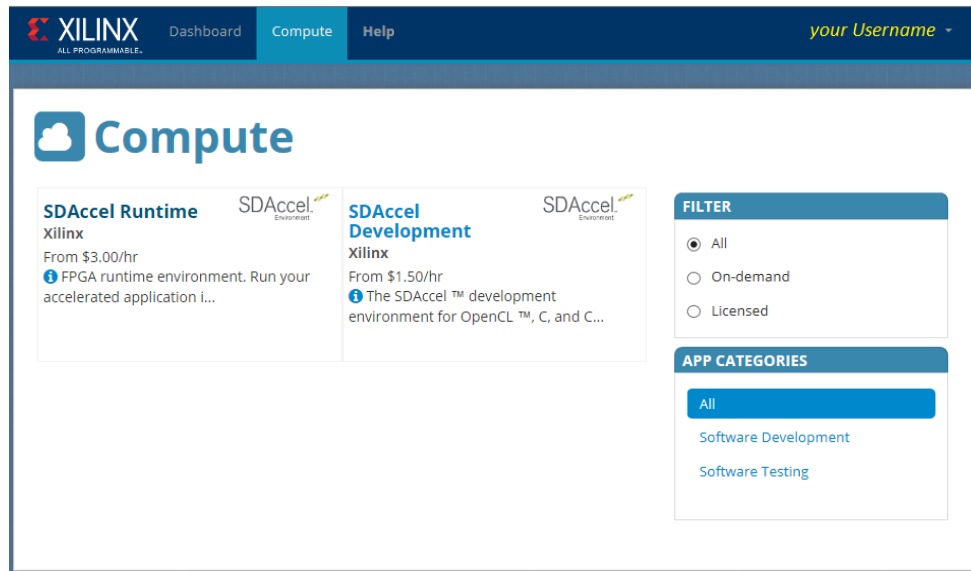


Figure 10: The Compute Page

- b. Use **SDAccel Development** to:
  - Compile and run your application in CPU and Hardware Emulation modes
  - Compile your application and transfer the compiled design to the **/data** directory for Hardware deployment using FPGA boards hosted by Nimble.
- c. Use **SDAccel Runtime** to:
  - Deploy your application on the FPGA Boards hosted by Nimble.
  - For application deployment, the compiled application and data files should be located in the **/data** directory.

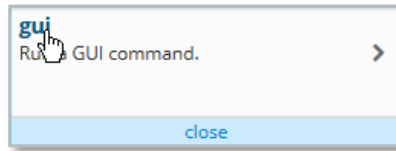


## Step 2: Load SDAccel Examples from the GitHub site

### *Launch SDAccel Development environment*

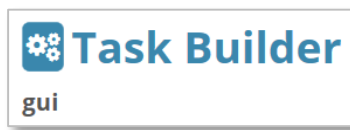
1. On the **Compute** page, select **SDAccel Development**.

2. Select **gui**.



**Figure 11: Selecting gui**

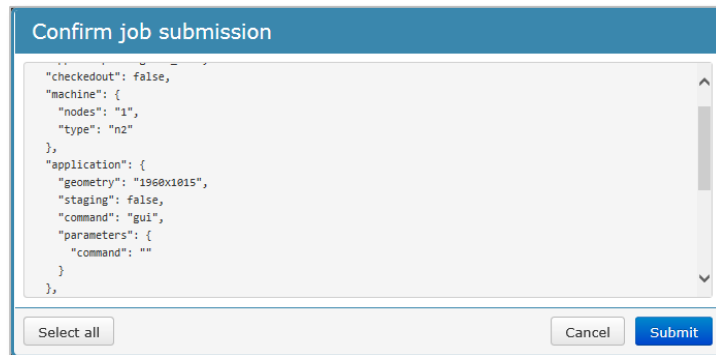
3. The **Builder** page will open. You can change default options, such as selecting a machine configuration to be used, Window Size, etc. if necessary.



**Figure 12: Task Builder**

After that, select **Continue**.

4. Confirm job submission by pressing **Submit**, followed by **OK**.



**Figure 13: Confirm Job Submission**

This will automatically make the **Dashboard** page active, as shown in Figure 14, where you can see that you have a new active job.

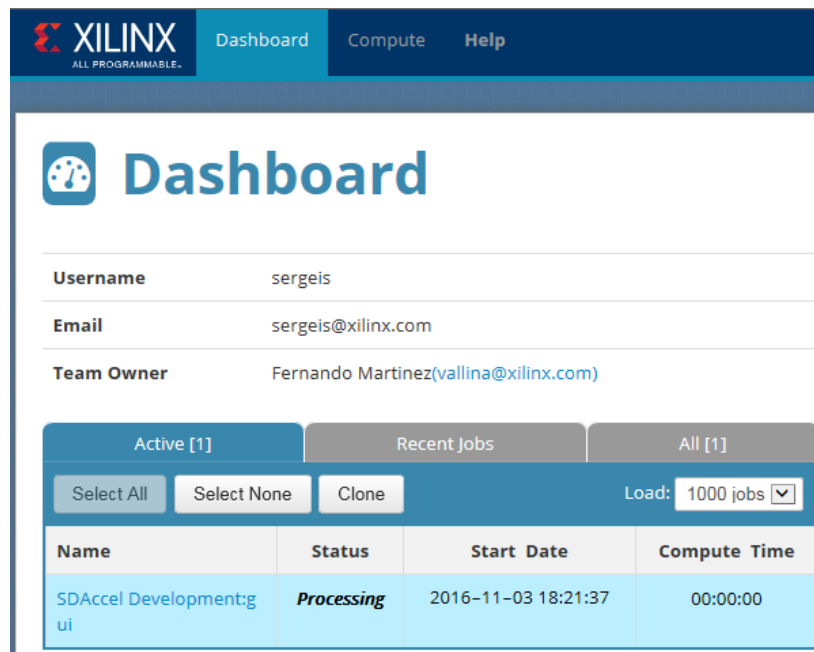


Figure 14: Dashboard Displaying an Active Job

5. Select **SDAccel™ Desktop:gui** job name to expand its content, and then select **Connect**.

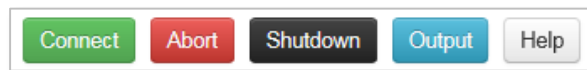


Figure 15: Instance Management Buttons

**Note:** it may take several seconds before the **Connect** button appears.

6. After that, the ready-to-use Linux desktop environment becomes available, as shown in Figure 16.

Two icons are on the desktop:

- **Terminal.** This opens a new terminal with a ready-to-use SDAccel environment
- **data:** This is a **/data** directory which you will use to transfer the compiled application for the Hardware deployment. See **/data Directory: Important Notes** for more information about the **/data** folder.

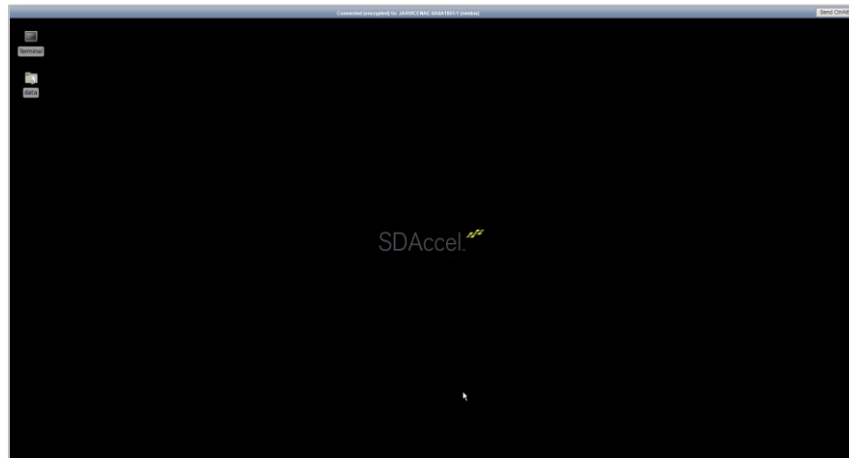


Figure 16: Ready-to-use Linux Desktop

## Download the Latest SDAccel Examples from GitHub

1. Open the **Terminal**



2. In the Terminal, enter the following command to get the latest version of the SDAccel examples hosted on GitHub:

```
git clone https://github.com/Xilinx/SDAccel\_Examples examples
```

3. After completion of this command, the terminal should look like

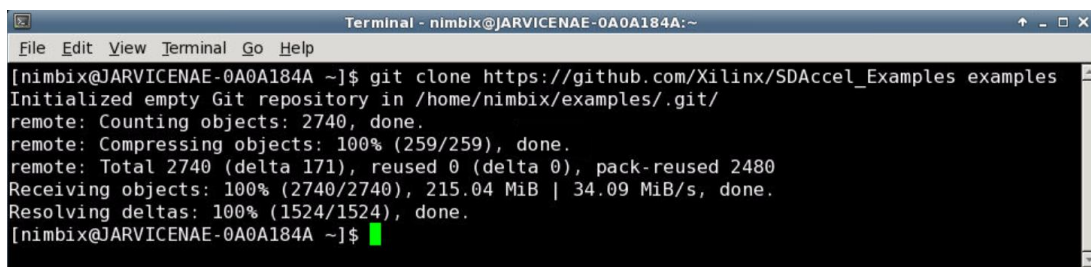


Figure 17: Terminal with Latest Version of SDAccel Examples

## Step 3: Design Compilation

As mentioned before, this tutorial uses a simple "hello" example.

In the opened Terminal, go to the Downloads/SDAccel\_Examples-master/getting\_started/hello directory by typing the following command:

```
cd Downloads/SDAccel_Examples-master/getting_started/hello
```

## Compile and run the application in CPU Emulation mode

The CPU Emulation mode verifies functional correctness of the application.

1. To compile the **hello** example for a CPU Emulation mode, run the following command in an opened Terminal from the **getting\_started/hello** directory:

```
make TARGETS=sw_emu all
```

This command generates:

- A host executable, named **hello** in the current directory.
- A **krnl\_hello.sw\_emu.xilinx\_adm-pcie-ku3\_2ddr\_xpr\_3\_2.xclbin** XCLBIN file located in the **xclbin** directory.

2. Before running the application in CPU Emulation mode, run the following commands:

```
export LD_LIBRARY_PATH=$XILINX_SDAccel/runtime/lib/x86_64/:$LD_LIBRARY_PATH
export XCL_EMULATION_MODE=sw_emu
emconfigutil --xdevice 'xilinx:adm-pcie-ku3:2ddr-xpr:3.2' --nd 1
```

3. Now you can launch the application using the following command:

```
./hello
```

Please note that the application is built in a way it is capable to select a correct XCLBIN file based on the value of the XCL\_EMULATION\_MODE variable.

4. Upon successful execution of the application, the following output appears:

```
export LD_LIBRARY_PATH=$XILINX_SDAccel/runtime/lib/x86_64/:$LD_LIBRARY_PATH
export XCL_EMULATION_MODE=<sw_emu|hw_emu>
emconfigutil --xdevice 'xilinx:adm-pcie-ku3:2ddr-xpr:3.2' --nd 1
```

## Compile and run the application in Hardware Emulation mode

The Hardware Emulation mode checks the correctness of the logic generated for the custom compute units and estimates the application performance.

1. To compile the example for a Hardware Emulation mode, run the following command in an opened Terminal from the **getting\_started/hello** directory:

```
make TARGETS=sw_emu all
```

This command generates:

- A host executable, named **hello** in the current directory.
- A **krnl\_hello.hw\_emu.xilinx\_adm-pcie-ku3\_2ddr\_xpr\_3\_2.xclbin** XCLBIN file located in the **xclbin** directory.



---

**IMPORTANT:** If you:

- Continue to use the same Terminal you used to run CPU Emulation mode, then before running the application in a Hardware Emulation mode, run the following command:

```
export XCL_EMULATION_MODE=hw_emu
```



- Compiled the application for Hardware Emulation mode using a different terminal, run the following commands first:

```
export LD_LIBRARY_PATH=$XILINX_SDAccel/runtime/lib/x86_64/:$LD_LIBRARY_PATH
export XCL_EMULATION_MODE=sw_emu
```

- 
2. Now you can launch the application using the following command:

```
./hello
```

Note that the application is built in a way it is capable to select a correct XCLBIN file based on the value of the XCL\_EMULATION\_MODE variable.

3. Upon successful execution of the application, the following output appears:

```
export LD_LIBRARY_PATH=$XILINX_SDAccel/runtime/lib/x86_64/:$LD_LIBRARY_PATH
export XCL_EMULATION_MODE=<sw_emu|hw_emu>
emconfigutil --xdevice 'xilinx:adm-pcie-ku3:2ddr-xpr:3.2' --nd 1
```

4. After the Hardware emulation has been completed, you can explore the application performance by opening the **sdaccel\_profile\_summary.html** file in the Firefox Web Browser with the following command:

```
firefox sdaccel_profile_summary.html
```

You should see the report shown in Figure 18.

## SDAccel Profile Summary

**Generated on:** 2016-12-07 16:45:44

**Profiled application:** hello

**Target platform:** Xilinx

**Target devices:** xilinx:adm-pcie-ku3:2ddr:3.2-0

**Flow mode:** System Run

**Tool version:** 2016.3

### OpenCL API Calls

API Name	Number Of Calls	Total Time (ms)	Minimum Time (ms)	Average Time (ms)	Maximum Time (ms)
clCreateProgramWithBinary	1	5403.92	5403.92	5403.92	5403.92
clCreateContext	1	4023.5	4023.5	4023.5	4023.5
clFinish	1	0.624681	0.624681	0.624681	0.624681
clEnqueueWriteBuffer	1	0.205888	0.205888	0.205888	0.205888
clEnqueueNDRangeKernel	1	0.051195	0.051195	0.051195	0.051195
clCreateBuffer	2	0.018401	0.004903	0.0092005	0.013498
clGetPlatformInfo	4	0.01546	0.000857	0.003865	0.011733
clGetPlatformIDs	1	0.013113	0.013113	0.013113	0.013113
clCreateKernel	1	0.013026	0.013026	0.013026	0.013026
clSetKernelArg	1	0.00656	0.00656	0.00656	0.00656
clBuildProgram	1	0.00599	0.00599	0.00599	0.00599
clGetEventProfilingInfo	2	0.004448	0.000868	0.002224	0.00358
clCreateCommandQueue	1	0.004378	0.004378	0.004378	0.004378
clGetDeviceInfo	2	0.004177	0.001687	0.0020885	0.00249
clGetDeviceIDs	1	0.00213	0.00213	0.00213	0.00213
clReleaseMemObject	1	0.001935	0.001935	0.001935	0.001935
clReleaseContext	1	0.001638	0.001638	0.001638	0.001638
clReleaseCommandQueue	1	0.001337	0.001337	0.001337	0.001337
clReleaseKernel	1	0.001275	0.001275	0.001275	0.001275

**Figure 18: SDAccel Profile Summary Report**

## ***Compile and prepare the application for Hardware Deployment***

1. To compile the example for hardware deployment, run the following command in an opened Terminal from the **getting\_started/hello** directory:

```
make TARGETS=hw all
```

This command generates:

- A host executable named **hello** in the current directory.
  - A **krnl\_hello.hw.xilinx\_adm-pcie-ku3\_2ddr\_xpr\_3\_2.xclbin** XCLBIN file located in the **xclbin** directory.
2. To simplify the application deployment on Nimbix, the host's executable should have the **.exe** file extension. Copy the **hello** executable to **hello.exe** with the following command:

```
mv hello hello.exe
```

3. Afterwards, run the **xbinst** utility to generate all necessary (additional) files for application deployment:

```
xbinst -x 'xilinx:adm-pcie-ku3:2ddr-xpr:3.2' -d .
```

It generates the **xbinst** folder in the current directory.

4. Copy the host executable (**hello.exe**) and XCLBIN (**krnl\_hello.hw.xilinx\_adm-pcie-ku3\_2ddr\_xpr\_3\_2.xclbin**) files to the **xbinst/pkg/pcie** directory in the following way:

```
cp hello.exe xbinst/pkg/pcie
mkdir xbinst/pkg/pcie/xclbin
cp xclbin/krnl_hello.hw.xilinx_adm-pcie-ku3_2ddr_xpr_3_2.xclbin xbinst/pkg/pcie/xclbin
```

You now have all the necessary files for hardware deployment.

5. The last step is to copy the **xbinst** folder to the **/data** directory with the following command:

```
cp -R xbinst* /data
```




---

**IMPORTANT:** At this step, you can close the current SDAccel Development session. However, all files which are not located in the **/data** directory will be lost. Therefore, copy any files you wish to preserve to the **/data** directory before you continue.

---

## Step 4: Terminating the Session

To terminate the current session:

1. Select **Log Out**.

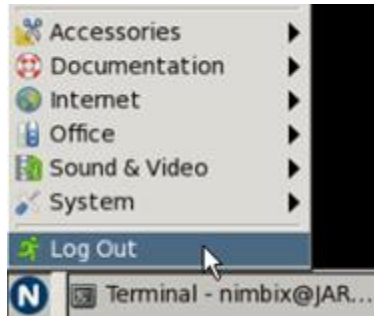


Figure 19: Log Out

2. Select **Shut Down**.

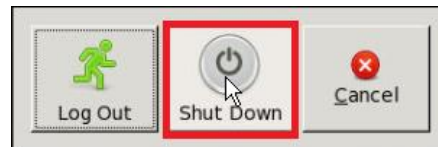


Figure 20: Shut Down

3. You should see in the **Dashboard** page that the process has been **Completed**. You should also receive a corresponding E-mail.

Name	Status	Start Date	Compute Time
SDAccel Development:gui	Completed	2016-11-03 18:21:37	01:34:42

Figure 21: Instance Status Check: Completed

**WARNING:** After terminating the session you might see that the job is still **Processing**, as shown in Figure 22.

Name	Status	Start Date	Compute Time
SDAccel Development:gui	Processing	2016-11-03 18:21:37	00:00:00

Figure 22: Instance Status Check: Processing

If so, click on the **SDAccel Development:gui** to expand the job and then select **Abort** to terminate the job.

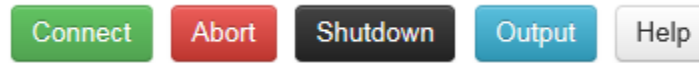


Figure 23: Instance Management Buttons

## Step 5: Application Deployment

The application is now compiled and all the necessary files for Hardware Deployment are copied to the **/data** directory. To deploy the application:

1. From the **Compute** page of the Nimble environment, select **SDAccel Runtime > batch**

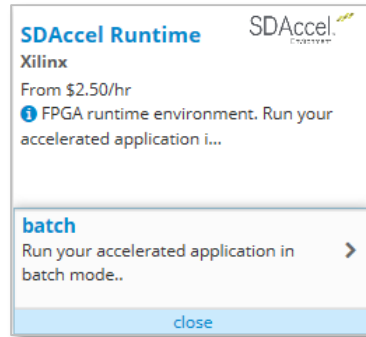


Figure 24: Selecting batch

2. On the opened page:
  - a. Select the target platform from the **Select Machine** drop-down menu.  
For this example, select KU3 board:

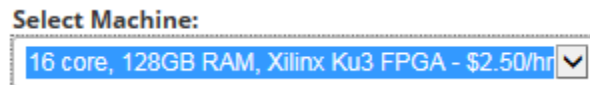


Figure 25: Selecting the Board

- b. Select the **hello.exe** host executable file located in the **xbinst/pkg/pcie** directory from the "Executable" dialog box.

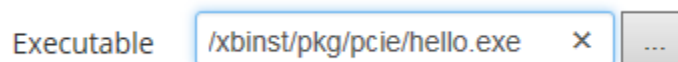


Figure 26: Selecting the Host Executable File

- c. **Arguments** field: The application automatically detects the XCLBIN file for deployment and therefore does not need any additional arguments specified.

Arguments

**Figure 27: Arguments Field**

**Note:** the directory where the host executable file is located is considered a current directory by Nimble.

- d. To obtain a graphical TRACE report, check one of the following options:

Timeline trace ☐ Download results in sdaccel\_timeline\_trace.csv when the job completes.

Device profiling ☐ Turn on device profiling and include in sdaccel\_timeline\_trace.csv file. Note that the profile summary (sdaccel\_profile\_summary.csv) is always created then downloaded when the job completes.

**Figure 28: TRACE Report Options**

- e. Select **Continue**, followed by **Submit** and **OK**, to launch the deployment and obtain the final results.

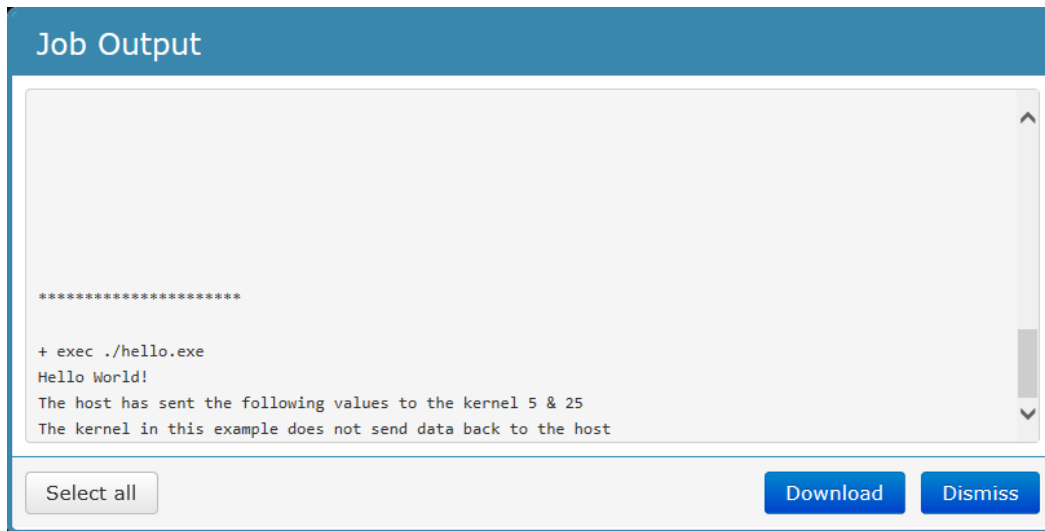
If the execution of the application is successful, the following status appears:

Name	Status	Start Date	Compute Time
SDAccel Runtime:batch	Completed	2016-11-03 20:36:43	00:00:20

**Figure 29: Successful Application Execution Status**

In addition, Nimble will automatically send you the application Log file by E-mail.

- f. Select the **SDAccel Runtime:batch** job to expand it and then select **Output** to see the LOG file generated by the application. You should see LOG messages similar to the CPU and Hardware Emulation modes, as shown in Figure 30.



**Figure 30: Job Output**

In addition to the E-mail with the LOG file you will receive from Nimble, you can save the LOG file on your local machine by clicking the Download button.

The application automatically generates the performance report in the **sdaccel\_profile\_summary.html** file, located in the **xbinst/pkg/pcie** directory. To explore this report you can either:

- Restart the **SDAccel Development** environment and then open this file using the Firefox Web Browser.
- Or copy the file to your local machine using the method described in Transferring Data to and from Nimble and then opening it using your Web Browser.

## Appendix A: Legal Notices

---

### Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

#### **AUTOMOTIVE APPLICATIONS DISCLAIMER**

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2012-2016 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.