# Getting Started with the SDAccel Environment on Nimbix Cloud

UG1240 (v201809) October, 2018



### **Revision History**

The following table shows the revision history for this document.

Date	Version	Changes	
09/17/2018	201809	Updated figures throughout	
		Updated board configurations	
		Changes to login URL	
08/16/2017	2017.2	Changes are: Modified Login URL.	
06/20/2017	2017.1	Changes are:	
		Updated for 2017.1 release.	
		Changed Nimbix to Material Compute flow.	
		Updated figures throughout.	
		Added reference to the Xilinx VU9P Developer Board DSA file.	
12/15/2016	2016.3	Updated supported board configurations.	
		Updated commands throughout.	
		Updated Figure 15.	
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	6
Creating an Account	6
Logging into Your Nimbix Account	7
Transferring Data to and from Nimbix	8
/data Directory: Important Notes	10
Chapter 3: Using the SDAccel Environment in Nimbix Cloud	11
Introduction	11
Step 1: Login to your Nimbix account	11
Step 2: Load SDAccel Examples from the GitHub Site	13
Step 3: Compile the Design	16
Step 4: Terminate the Session	19
Step 5: Deploy the Application	21



Chapter 1: SDAccel Environment on Nimbix Cloud Overview

### Introduction

This document demonstrates how to get started using SDAccel<sup>™</sup> Environment in the Nimbix Cloud.

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

Nimbix Inc<sup>®</sup> has partnered with Xilinx<sup>®</sup> to deliver an opportunity to test drive SDAccel and see how FPGA-based acceleration can speed-up your OpenCL C, C/C++ and RTL Kernels. The Nimbix Cloud provides a purpose-built compute cloud for big data and computation. It now hosts the SDAccel development environment for companies who want to streamline evaluations and accelerate FPGA OpenCL kernels.

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

- Using Nimbix to compile and deploy your application with Xilinx FPGAs
- 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 in the Nimbix environment.



**IMPORTANT:** When compiling the application to be deployed on Xilinx FPGAs available in the Nimbix Cloud, you need specify one of the following board configurations:

SDAccel development app name	Development toolkit installed	Built-in FPGA Board supports (DSA)
Xilinx SDAccel Development 2017.4_op	SDx 2017.4_op	Xilinx KCU1500 Accelerator (DSA: xilinx_kcu1500_dynamic_5_0) Xilinx VCU1525 Accelerator (DSA: xilinx_vcu1525_dynamic_5_1)
Xilinx SDAccel Development 2018.2	SDx 2018.2	Xilinx KCU1500 Accelerator (DSA: xilinx_kcu1500_dynamic_5_0) Xilinx VCU1525 Accelerator (DSA: xilinx_vcu1525_dynamic_5_1)
Xilinx SDAccel Development 2018.2 XDF	SDx 2018.2	Xilinx Alveo U200 Accelerator (DSA: xilinx_u200_xdma_201820_1) Xilinx Alveo U250 Accelerator (DSA: xilinx_u250_xdma_201820_1)

Nimbix Supported Xilinx SDx (SDAccel) Development



#### Nimbix Supported Xilinx SDx (SDAccel) Runtime and Developer Boards

Xilinx FPGA Board Acces App	Runtime/Driver installed	Machine Type	FPGA Board (DSA)
Xilinx FPGA (with	SDx 2017.4_op	nx4a	Xilinx KCU1500 Accelerator (DSA: xilinx_kcu1500_dynamic_5_0)
2017.4_op Runtime)		nx5a	Xilinx VCU1525 Accelerator (DSA: xilinx_vcu1525_dynamic_5_1)
Xilinx FPGA (with 2018.2	SDx 2018.2	nx4b	Xilinx KCU1500 Accelerator (DSA: xilinx_kcu1500_dynamic_5_0)
Runtime)		nx5b	Xilinx VCU1525 Accelerator (DSA: xilinx_vcu1525_dynamic_5_1)
Xilinx Alveo FPGA (with	XRT	nx5u	Xilinx Alveo U200 Accelerator (DSA: xilinx_u200_xdma_201820_1)
XRT Runtime)		nx6u	Xilinx Alveo U250 Accelerator (DSA: xilinx_u250_xdma_201820_1)



Chapter 2: Nimbix Account Creation, Login, and Setup

### **Creating an Account**

To create an account, do the following:

- 1. Register for the SDAccel<sup>®</sup> Environment at the Nimbix website. Navigate with your browser to https://www.nimbix.net/fpga-developer-program/
- 2. The following figure shows the Login page.



### Get started with FPGA accelerated applications today on the Nimbix Cloud

Nimbix has partnered with Xilinx to provide developers and engineers a trial account that provides up to 100 hours of free time on the Nimbix Cloud using Xilinx FPGA Tools and Accelerators

Register for access to the FPGA Developer Program to get started with FPGA accelerated applications today. Applicant must be part of a valid company or university. Free trial time is limited to FPGA enabled resources and Xilinx tools and applications. A valid credit card is required for access to non-FPGA resources.

First	Last
Email	
Company or Univ	versity email address required. No ٤
City *	
State *	

Figure 1: Nimbix Cloud Landing Page

3. Complete the form.

Upon approval of your account, you receive an email confirmation along with information introducing to complete the sign up and registration process



### Logging into Your Nimbix Account

To login to your account:

1. Navigate to <u>https://xilinx-cloud.jarvice.com</u>.

N Login	
User name	
Password	
Keep me logged in	
Forgot Password	LOGIN

Figure 2: Nimbix Login Page

2. 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 **Compute** opens.

≡ Compute					NIMBIX
Filter Applications	Q Search			×	Compute
OT License Included	Xilinx				Dashboard
Nimbix File Manager	Xilinx GEMX E XILINX	Xilinx ML Suite <b>£</b> XILINX.	Xilinx SDAccel SDAccel."	Xilinx SDAccel SDAccel.	PushToCompute™
E All Applications	Xilinx	Allinx	2017.4_op Xilinx	2018.2 Xilinx	Account
A My Applications	NULL OD Local and an	Niller OD tool attact	Willow OD Local and and	NULL NEEDE OF MILLIN	
Categories	Runtime 2017.1 Xilinx	Runtime 2017.4_op Xilinx	Runtime 2018.2	XIIIIX WEBP & ALLINX Xilinx	Xilinx WEBP & XILINX
Community	from \$3.00/hr	from \$3.00/hr	from \$3.00/hr	from \$3.00/hr	Xilinx
Vendors V					from \$3.00/hr
					Nimbix File Manager Nimbix, Inc.
					from \$0.36/hr
					Xilinx SDAccel SDAccel Development 2018.2 Xilinx
					from \$1.25/hr
					HPC Test Environment Nimbix, Inc.
					from \$2.42/hr

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.

### **Transferring Data to and from Nimbix**

When working with SDAccel, you might need to transfer files from your machine. Nimbix. SFTP is the preferred file transfer method. The following instructions demonstrate file transfer using the <u>FileZilla</u> SFTP client software.

FileZilla				
File Edit View Transfe	er Server Bookmark	s Help		
M - 17: **	🔁 🐰 🎮 🌸 🛷	🗉 🕵 😚 🚜		
Host:	Username:	Password:	Port:	Quickconnect
Local site: C:\			<ul> <li>Remote site:</li> </ul>	
Desktop				
My Documents				
🗄 🚛 Computer				
🗄 🚢 C: (OSDisk)				

*Note*: Follow this <u>link</u> for more information regarding file transfers.

#### Figure 4: FileZilla SFTP client

- 1. Download and install FileZilla.
- 2. Open FileZilla and configure the following:
  - a. In Host: Specify drop.jarvice.com
  - b. In Port: Specify the value 22
  - c. In **Username:** Specify your Nimbix Username
  - d. The Password: Specify 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 5.



≡	Account		Your Usern	
Setti	ngs	Profile	^	Compute
₽ ♠ ⊖	Profile Notifications SSH Keys	Full Name Your Name Company Your Company Phone		Dashboard PushToCompute™ Account Log Out
		EDIT Authentication API Key Your API Key REGENERATE	_	Recent

Figure 5: API Key on the Settings Page

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

When the connection establishes successfully, you see the /data directory on the remote site, as shown in the following figure:

File	Edit View Transfer Server E	Bookmarks Help					
1	1 🕂 😫 🗲 🎲 🛱	🎉 🛷 🔳 🕵	🚰 🕺				
Host:	sftp://drop.jarvice.d Username:	sergeis	Password:	•••••	Port:	Quickco	onnect 💌
Status:	Connecting to drop.jarv	ice.com					
Status:	Connected to drop.jarvi	ce.com					
Status:	Retrieving directory listi	ng					
Status:	tatus: Listing directory /data						
Status:	itatus: Directory listing of "/data" successful						
Local	site: C:\			-	Remote site:	/data	
<b>-</b>	Desktop				B <b>2</b> /		
	My Documents				dat	ta	
				H			
	🕀 🟭 C: (OSDisk)						

#### Figure 6: 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 the following features:

- Data located in the Nimbix /data directory is preserved after you shut down running systems and 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.
- The /data directory is where you need to exchange data between Nimbix and your local machine. For example, if you need code 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.
- The /data directory is the location from which to deploy compiled applications on FPGAs in the Nimbix Cloud. Copy all necessary files to the /data directory first.

Chapter 3: Using the SDAccel Environment in Nimbix Cloud describes these activities.



Chapter 3: Using the SDAccel Environment in Nimbix Cloud

### Introduction

The tutorial uses the vadd example available in the /getting\_started/misc folder at this location:

https://github.com/Xilinx/SDAccel Examples/.

### Step 1: Login to your Nimbix account

1. Go to: https://xilinx-cloud.jarvice.com

N Login	
User name	
Password	
Keep me logged in	
Forgot Password	LOGIN

Figure 7: Nimbix Account Login

- 2. Provide your Username and Password and select Log In.
- 3. In the opened page, select **Compute**, shown in the following figure, to open the **Compute** page.

NIMBIX
Compute
Dashboard
PushToCompute™
Account
Log Out

Figure 8: Compute

The **Compute** page opens, as shown in the following figure:



≡ Compute					tan en de Angeland	NIMBIX
Filter Applications		Q Search			×	Compute
OT License Included		Xilinx				Dashboard
Nimbix File Manager		Xilinx GEMX E XILINX	Xilinx ML Suite E XILINX.	Xilinx SDAccel SDAccel."	Xilinx SDAccel SDAccel	PushToCompute™
		Xilinx	Xilinx	Development 2017.4_op Xilinx	Development 2018.2 Xilinx	Account
		from \$3.00/hr	from \$3.00/hr	from \$1.50/hr	from \$1.25/hr	Log Out
My Applications		Xilinx SDAccel SDAccel	Xilinx SDAccel SDAccel	Xilinx SDAccel SDAccel	Xilinx WEBP E XILINX	Recent
Categories	~	Xilinx	2017.4_op Xilinx	Xilinx		Xilinx WEBP E XILINX
Community	~	from \$3.00/hr	from \$3.00/hr	from \$3.00/hr	from \$3.00/hr	
Vendors	~					from \$3.00/hr
						Nimbix File Manager Nimbix, Inc.
						from \$0.36/hr
						Xilinx SDAccel SDAccel."" Development 2018.2 Xilinx
						from \$1.25/hr
						HPC Test Environment Nimbix, Inc.
						from \$2.42/hr

Figure 9: The Compute Page

#### a. 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 Nimbix.

#### b. Use SDAccel Runtime to:

 Deploy your application on the FPGA machines in the Nimbix Cloud.
 For application deployment, the compiled application and data files should be located in the /data directory. SDAccel Development Xilinx

SDAccel Runtime Xilinx

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

- 1. Launch SDAccel Development Environment
- 2. On the Compute page, select SDAccel Development 2018.2 XDF.
- 3. Select the **Desktop Mode**, shown in the following figure:

The SDAccel <sup>™</sup> development environment for OpenCL <sup>™</sup>, C, and C++, enables up to 25X better performance/watt for data center application acceleration leveraging FPGAs. SDAccel, member of the SDx <sup>™</sup> family, combines the industry's first architecturally optimizing compiler supporting any combination of OpenCL, C, and C++ kernels, along with libraries, deve ...

Desktop Mode Batch Mode

#### Figure 10: Selecting Desktop Mode

4. The **Builder** page opens. You can change default options, such as the machine configuration to use, window size, and so forth, if necessary. The following figure shows the top-level Desktop Mode page. When your configurations are set, continue the job submission by pressing **Submit**.

GENERAL	OPTIONAL	STORAGE	PREVIEW SUBM	AISSION
Machine				
Machine type	8 core, 64GB RAM (CPU only) (n2)		v	
Cores	8		\$1.50/hr	
				SURM

Figure 11: Task Builder

This automatically makes the **Dashboard** page active, as shown in Figure 12, where you can see that you have a new active job.



**NOTE:** If you wish to copy and paste commands from this document directly to the desktop environment, you can connect with a VNC client or by using ssh to log in to a terminal with the credentials provided in the Dashboard Current view as in Figure 12.



#### Current



Figure 12: Dashboard Displaying an Active Job

5. Click in the desktop preview space to open a new tab with an interactive desktop session.

*Note*: *It can take several seconds before the* desktop preview opens.

The ready-to-use Linux desktop environment becomes available, as shown in the following figure:



Figure 13: Ready-to-Use Linux Desktop



Two icons are on the desktop:

- **Terminal**: Opens a new terminal with a ready-to-use SDAccel environment.
- data: The directory in which you transfer the compiled applications for hardware deployment. See /data Directory: Important Notes for more information about the /data directory.

### 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 --single-branch --branch 2018.2_xdf --depth=1 https://github.com/
Xilinx/SDAccel_Examples /data/examples
```

After the command executes, the terminal should look like the following figure:



Figure 14: Terminal with Latest Version of SDAccel Examples

### **Step 3: Compile the Design**

This tutorial uses a simple "vadd" example.

In the opened Terminal, go to the /data/examples/getting\_started/misc/vadd directory by typing the following command:

cd /data/examples/getting\_started/misc/vadd

### Compile and Run the Application in CPU Emulation Mode

The CPU Emulation mode verifies functional correctness of the application. To compile the vadd example for a CPU Emulation mode, do the following:

1. Run the following command in an opened Terminal from the vadd directory:

```
make TARGETS=sw_emu DEVICES=xilinx_u200_xdma_201820_1 all
```

This command generates:

- o A host executable, named vadd in the current directory.
- o A krnl\_vadd.sw\_emu.xilinx\_u200\_xdma\_201820\_1.xclbin file (XCLBIN) located
  in the xclbin directory.
- 2. Before running the application in CPU Emulation mode, run the following commands:

export XCL\_EMULATION\_MODE=sw\_emu

emconfigutil --platform 'xilinx\_u200\_xdma\_201820\_1' --nd 1

These commands set the library path, the XCL EMULATION MODE variable, and the DSA file.

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

./vadd

**Note:** This application uses the XCL\_EMULATION\_MODE variable and discovered emulated devices to select the correct XCLBIN file.

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

```
Result Match: i = 1021 CPU result = 3063 Krnl Result = 3063
Result Match: i = 1022 CPU result = 3066 Krnl Result = 3066
Result Match: i = 1023 CPU result = 3069 Krnl Result = 3069
TEST PASSED
```

### 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. To compile the example for a Hardware Emulation mode:

1. Run the following command in an opened Terminal from the vadd directory:

```
make TARGETS=hw_emu DEVICES=xilinx_u200_xdma_201820_1 check
```

This command generates:

- o A host executable, named **vadd** in the current directory.
- o A krnl\_vadd.hw\_emu.xilinx\_u200\_xdma\_201820\_1.xclbin (XCLBIN) file located in the xclbin directory.

IMPORTANT: If you:

• Continue to use the <u>same</u> 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 <u>different</u> terminal, run the following commands first:

export XCL\_EMULATION\_MODE=hw\_emu

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

./vadd

**Note**: This application uses the XCL\_EMULATION\_MODE variable and discovered emulated devices to select the correct XCLBIN file.

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

```
Result Match: i = 1021 CPU result = 3063 Krnl Result = 3063
Result Match: i = 1022 CPU result = 3066 Krnl Result = 3066
Result Match: i = 1023 CPU result = 3069 Krnl Result = 3069
TEST PASSED
```

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 using the following command:

firefox sdaccel\_profile\_summary.html

You should see the report shown in Figure 15.

### **SDAccel Profile Summary**

Application: vadd Created: 2018-09-18 00:06:12 Devices: xilinx\_vcu1525\_dynamic\_5\_1-0 Msec: 1537229172543 Report name: Profile Summary Target: Hardware Emulation Tool version: 2018.2

### OpenCL API Calls

API Name	Number Of Calls	Total Time (ms)	Minimum Time (ms)	Average Time (ms)	Maximum Time (ms)
clCreateProgramWithBinary	1	11693.800	11693.800	11693.800	11693.800
clFinish	1	3004.000	3004.000	3004.000	3004.000
clEnqueueMigrateMemObjects	2	18.576	1.257	9.288	17.319
clReleaseProgram	1	2.144	2.144	2.144	2.144
clCreateBuffer	3	1.715	0.556	0.572	0.591
clEnqueueTask	1	1.274	1.274	1.274	1.274
clCreateKernel	1	0.025	0.025	0.025	0.025
clSetKernelArg	4	0.014	0.001	0.003	0.009
clReleaseKernel	1	0.011	0.011	0.011	0.011
clRetainMemObject	3	0.011	0.002	0.004	0.006
clGetExtensionFunctionAddress	1	0.011	0.011	0.011	0.011
clGetDeviceIDs	2	0.010	0.001	0.005	0.009
clReleaseMemObject	6	0.009	0.001	0.002	0.004
clGetPlatformInfo	4	0.007	0.001	0.002	0.003
clCreateCommandQueue	1	0.006	0.006	0.006	0.006
clCreateContext	1	0.006	0.006	0.006	0.006
clRetainDevice	2	0.005	0.002	0.002	0.003
clGetDeviceInfo	2	0.005	0.002	0.002	0.003
clReleaseCommandQueue	1	0.004	0.004	0.004	0.004
clReleaseDevice	2	0.003	0.001	0.002	0.002
clReleaseContext	1	0.003	0.003	0.003	0.003

#### Figure 15: 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/vadd directory:

make TARGETS=hw DEVICES=xilinx\_u200\_xdma\_201820\_1 all

This command generates:

- o A host executable named **vadd** in the current directory.
- 2. To simplify the application deployment on Nimbix, the host's executable should have the **.exe** file extension. Copy the **vadd** executable to **vadd.exe** with the following command:

mv vadd vadd.exe



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

### **Step 4: Terminate the Session**

To terminate the current session:

1. From the **Dashboard**, select shutdown.



Figure 16: Shutdown Option

#### 2. Select Yes.

You should see in the Dashboard page that the process status is **Completed**. You should also receive a corresponding E-mail.



Figure 17: Instance Status Check: Completed



**WARNING!** After terminating the session, you might see that the job is still **Processing**, as shown in the following figure.



### Current

SDAccel. SDAcce	cel Development(278023)	Ē 🖸 🛈
Command	batch	
Status	Processing	

Figure 18: Instance Status Check: Processing

3. If the job is still processing, check **Force Stop**, select **Yes**, then select **Shutdown** to terminate the job.



Figure 19: Instance Management Buttons

### **Step 5: Deploy the Application**

The application is now compiled and all the necessary files for hardware deployment are located in the **/data** directory. To deploy the application:

1. From the **Compute** page of the Nimbix Environment, select **SDAccel Runtime 2018.2 XDF> batch**, shown in the following figure.

FPGA runtime environment. Run your accelerated application in batch mode.



Figure 20: Selecting Batch Workflow

- 2. On the opened page:
  - a. Select the target platform from the **Select Machine** drop-down menu.

For this example, select **Alveo U200 FPGA**:

Machine type 16 core, 128GB RAM, Xilinx U200 FPGA (nx5u)

#### Figure 21: Selecting the Board

b. Select the vadd.exe host executable file located in the /examples/getting\_started/misc/vadd/ directory from the Executable dialog box, shown in the following figure.

Executable	/examples/getting_started/misc/vadd/vadd.exe	
Select executable to run (*.exe)		

#### Figure 22: Selecting the Host Executable File

The application automatically detects the XCLBIN file for deployment; no further arguments need to be specified.

Arguments
ommand-line arguments for

#### Figure 23: Arguments Field

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



c. To obtain a graphical Trace report, check one of the following options:



#### Figure 24: TRACE Report Options

d. Select **Submit** to launch the deployment and obtain the results.

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

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

#### Figure 25: Successful Application Execution Status

Nimbix sends you the application log file by email automatically.

e. 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 26.



#### Current

SDAccel SDAcce	cel Runtime(278687) batch	r 🖸 🗘	
Status Result Match: Result Result	<pre>Processing i = 1004 CPU result = 3012 Krnl Result = 3012 i = 1006 CPU result = 3015 Krnl Result = 3015 i = 1006 CPU result = 3021 Krnl Result = 3021 i = 1007 CPU result = 3021 Krnl Result = 3024 i = 1009 CPU result = 3027 Krnl Result = 3027 i = 1010 CPU result = 3030 Krnl Result = 3033 i = 1011 CPU result = 3038 Krnl Result = 3033 i = 1012 CPU result = 3038 Krnl Result = 3033 i = 1014 CPU result = 3034 Krnl Result = 3034 i = 1016 CPU result = 3042 Krnl Result = 3034 i = 1016 CPU result = 3048 Krnl Result = 3034 i = 1016 CPU result = 3048 Krnl Result = 3048 i = 1016 CPU result = 3048 Krnl Result = 3046 i = 1017 CPU result = 3054 Krnl Result = 3054 i = 1018 CPU result = 3057 Krnl Result = 3056 i = 1020 CPU result = 3066 Krnl Result = 3066 i = 1022 CPU result = 3066 Krnl Result = 3066 i = 1022 CPU result = 3066 Krnl Result = 3069 i = 1021 CPU result = 3066 Krnl Result = 3069 i = 1021 CPU result = 3066 Krnl Result = 3069 i = 1021 CPU result = 3066 Krnl Result = 3069 i results match cpu results.</pre>		

Figure 26: Job Output

In addition to the email with the log file you receive from Nimbix, you can save the log file on your local machine by clicking the **Download** button in the output windows, as shown in the following figure:

Output:	Opening Vault Generating SSH key pair for this job Initializing NAE CPU cores count: 8	(L)
	CPU thread count: 8	
	Locking password for user root. passwd: Success	
	Locking password for user nimbix. passwd: Success Starting NAE	
	NAE Started in I second(s).	

Figure 27: Download Log File

The application automatically generates the performance report in the

sdaccel\_profile\_summary.html file, located in the xbinst directory. To explore this report
you can either:

- Restart the SDAccel Environment and open this file using the Firefox Web Browser.
- Copy the file to your local machine using the method described in Transferring Data to and from Nimbix and then opening it using your Web Browser.

