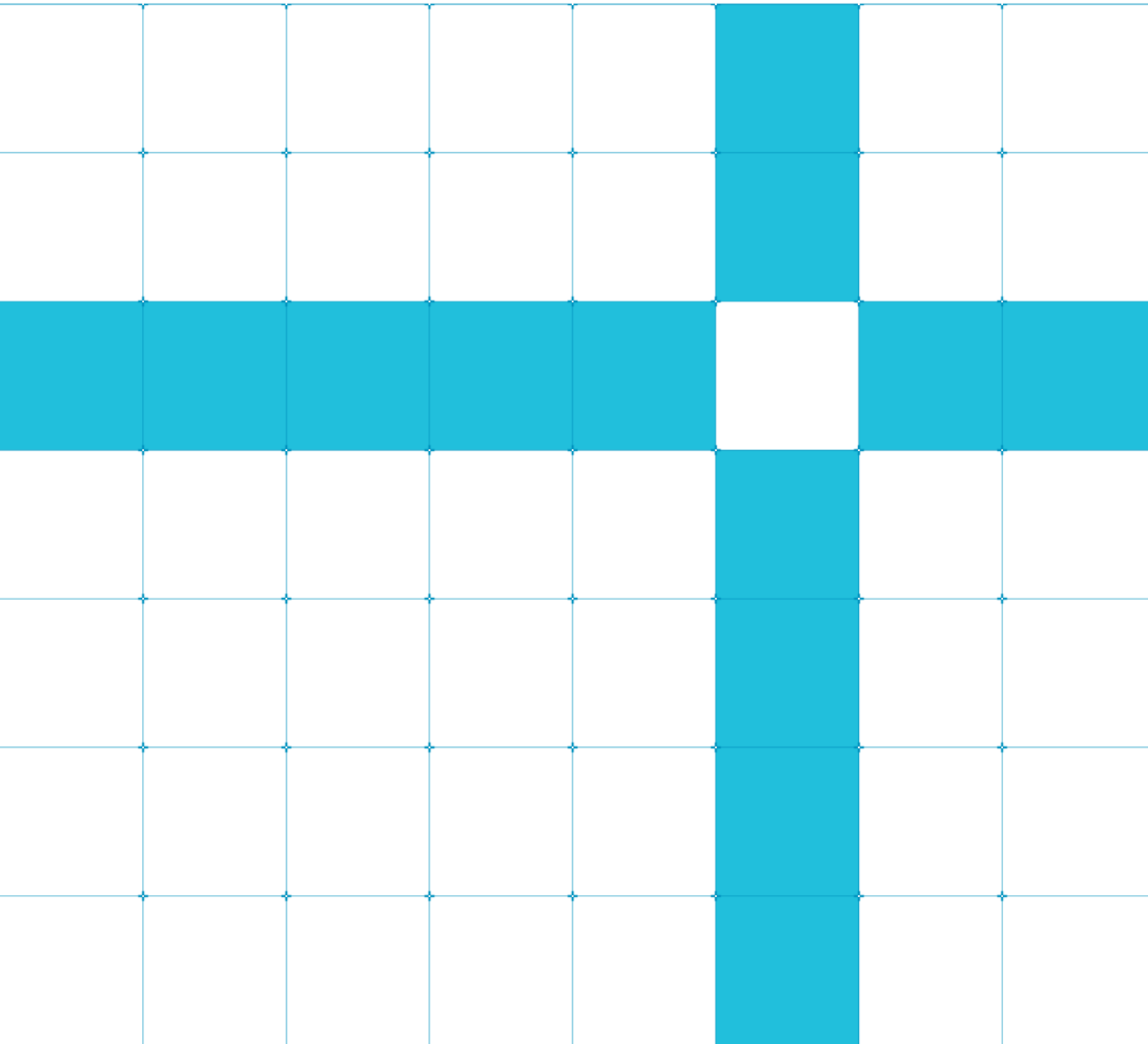




Deploying cloud-based ML for speech transcription

Version 1.0



Deploying cloud-based ML for speech transcription

Copyright © 2018 Arm Limited (or its affiliates). All rights reserved.

Release Information

Document History

Version	Date	Confidentiality	Change
1.0	18 September 2018	Non-confidential	First release

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. **No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.**

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at <http://www.arm.com/company/policies/trademarks>.

Copyright © 2018 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Web Address

<http://www.arm.com>

Contents

1 About this document	5
2 Overview	5
3 Before you begin.....	5
4 Deploy an Arm server	6
5 Build an ML framework for Arm	7
5.1. Install dependencies	7
5.2. Build PaddlePaddle	7
5.3. Install PaddlePaddle	8
6 Install DeepSpeech 2 for Arm	8
6.1. Install dependencies	8
6.2. Build DeepSpeech	8
6.3. Download models while building	8
6.4. Build speech manifest	9
7 Install the speech-to-text demo	9
7.1. Install demo on client	9
7.2. Prepare demo server	9
7.3. Start demo server	10
8 Run the speech-to-text demo	10
8.1. Run demo on client	10
9 Next steps	10

1 About this document

This document contains a how-to guide for setting up a client-server speech transcription deployed as a service running on cloud-hosted Arm servers.

2 Overview

Not every machine learning task runs on an edge device. Some tasks, such as offline video captioning or podcast transcription, are not time-critical and are therefore particularly well-suited to running in the data center; the increase in compute performance available that significantly speeds up such tasks.

This guide shows you how to set up client-server speech transcription deployed as a service running on [cloud-hosted Arm servers](#). Here, you record an audio file to your client machine then send it to the server. The Arm-based server utilises a speech recognition service that uses machine learning, and then sends the text back to your client machine.

Important

Deploying a server in the cloud is not free, and you will need to pay a small amount to [packet.net](#) to complete this guide.

3 Before you begin

This is a technical deployment walkthrough using Ubuntu 16.04, so some familiarity with the command-line, Linux package managers, and SSH is assumed. No knowledge of machine learning is necessary.

The installations and builds that are described in this guide can take several hours to undertake, but once installed the service will be up and running very quickly.

Ensure that your PC has a working microphone as you will need to record your voice for the transcription service to work.

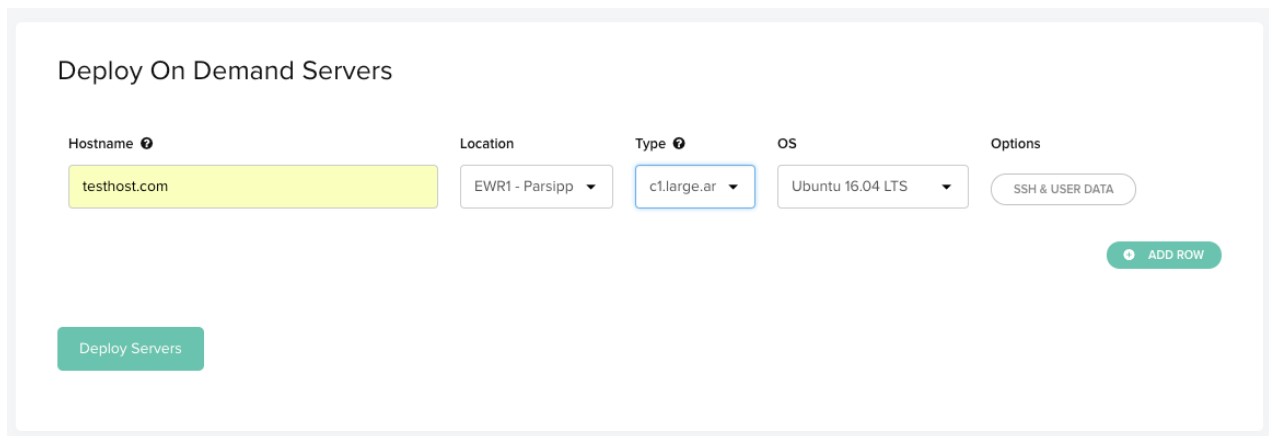
Before you start this guide, you need to create an account at [packet.net](#). Your account may take some time to be verified and costs \$1 to create. Once your account is verified, you need to do the following before you can deploy an Arm server:

1. Create a new [packet.net](#) project.
2. Generate an SSH keys pair and add the public key to your project. This allows you to login securely to the server. Follow these [packet.net instructions](#) on how to do this.

Packet is a paid-for cloud-based computing service which provides bare-metal servers. You will need to provide payment details on sign-up. We'll be using the Type 2A server (Cavium Thunder X), which costs \$0.50 as of April 2018. The computing cost for this guide is approximately \$3.

4 Deploy an Arm server

1. Log into packet.net and either create a new project or open an existing one.
2. Click on the *Servers* tab and then select the *Deploy servers* button.
3. Enter a suitable hostname. Note that the hostname is for your reference only and does not need to be tied to a registered domain.
4. From the Location dropdown, choose one of these options:
 - NRT1
 - SJC1
 - EWR1Arm-based servers are only available in these locations.
5. From the Type dropdown, select a c1.large.arm server.
6. From the OS dropdown, select Ubuntu 16.04 LTS, as shown here:



The screenshot shows the 'Deploy On Demand Servers' interface. It features a form with the following fields and controls:

- Hostname:** A text input field containing 'testhost.com'.
- Location:** A dropdown menu showing 'EWR1 - Parsipp'.
- Type:** A dropdown menu showing 'c1.large.ar'.
- OS:** A dropdown menu showing 'Ubuntu 16.04 LTS'.
- Options:** A button labeled 'SSH & USER DATA'.
- ADD ROW:** A green button with a plus icon and the text 'ADD ROW'.
- Deploy Servers:** A green button at the bottom left of the form.

7. Select *Deploy Servers*. This process takes approximately 5-10 minutes.

New servers will be created with the SSH public key you provided in your project settings. You will need this to log in after it boots. Check this now, or add an extra one by clicking the "SSH & user data" options button.

Once your server has booted, its IP address is shown on the Servers page.

8. Open a command line and replace **<ip address>** with the one provided to log in to the server using this command:

```
ssh root<ip address>
```

If you can login successfully, then we will revisit this command further on in this guide to deploy and run a machine learning demo on your server.

5 Build an ML framework for Arm

The framework you choose may depend on the application you wish to run. This example uses Baidu's [DeepSpeech 2](#), a state-of-the-art speech recognition system that provides very high-quality models for both English and Chinese.

DeepSpeech 2 is built on Baidu's [PaddlePaddle](#) framework. Although less well-known than [TensorFlow](#), it is just as easy to build and configure on an Armv8 system.

The following instructions provided here work on Ubuntu 16.04 LTS running on a [packet.net](#) type 2A server. For reference, the official guide for building PaddlePaddle from source is here: <https://github.com/PaddlePaddle/DeepSpeech>.

5.1. Install dependencies

Most dependencies are already pre-built for Armv8. Logged in to the server, enter the following to a command line to obtain and install the dependencies from standard repositories:

```
apt-get update
```

```
apt-get -y install python-dev python-pip python-numpy python-scipy python-wheel git cmake swig  
golang libfreetype6-dev libpng12-dev libopenblas-dev
```

```
pip install protobuf
```

```
git clone https://github.com/PaddlePaddle/recordio.git
```

```
cd recordio/python
```

```
./build.sh
```

```
pip install -e .
```

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

5.2. Build PaddlePaddle

Building from source can take several hours. Once complete, you can copy the *.whl file and deploy it directly onto subsequent servers. To build from source:

```
git clone https://github.com/PaddlePaddle/Paddle.git
```

```
cd Paddle
```

```
mkdir build
```

```
cd build
```

```
cmake -DWITH_GPU=OFF -DWITH_TESTING=OFF ..
```

```
make -j96
```

```
cd ..
```

5.3. Install PaddlePaddle

Installing the built package is straightforward.

1. Display the contents of the python/dist directory by entering the ls command. Note the version number in the name of the .whl file.
2. Enter this command and add the version number that you noted in step 1:

```
pip install Paddle/python/dist/paddlepaddle-0.11.0-cp27-cp27mu-linux_aarch64.whl
```

6 Install DeepSpeech 2 for Arm

Baidu's DeepSpeech network provides state-of-the-art speech-to-text capabilities. Their PaddlePaddle-based implementation comes with state-of-the-art models that have been trained on their internal >8000 hour English speech dataset. Mandarin versions are also available.

Mozilla host a TensorFlow-based version of DeepSpeech, but the model files available for it are trained on small public datasets and offer significantly lower accuracy than Baidu's internally-trained ones.

The remainder of this section provides a condensed guide on installing DeepSpeech2, tested on Ubuntu 16.04 LTS running on a [packet.net](#) Type 2A server. To install it on another platform, follow [Baidu's general installation guide](#).

6.1. Install dependencies

Once you have a working PaddlePaddle installation, install the additional DeepSpeech dependencies. These are mostly audio codecs:

```
apt-get install -y pkg-config libflac-dev libogg-dev libvorbis-dev libboost-dev libffi-dev
```

6.2. Build DeepSpeech

DeepSpeech's requirements.txt file specifies particular scipy and Cython versions, which will automatically be built from source. These builds can take longer than an hour, so while this is happening, download the models, which also takes a long time.

```
git clone https://github.com/PaddlePaddle/DeepSpeech.git
```

```
cd DeepSpeech
```

```
bash setup.sh
```

6.3. Download models while building

These two files are large (400MB and 8GB) so it is useful to start downloading these while the previous build step is in progress. To do this, open a new command line, login to the server using SSH as before, navigate to the Paddle/build/python/dist directory and enter:

```
cd DeepSpeech/models/baidu_en8k
```

```
bash download_model.sh
```

```
cd ../lm
```

```
bash download_lm_en.sh
```



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

6.4. Build speech manifest

The librispeech manifest is used by the demo server to provide warmup examples. Scripts to download it are provided:

```
cd data/librispeech
ln -s ../../data_utils data_utils
python librispeech.py --full_download=False
cd ../../
```

7 Install the speech-to-text demo

7.1. Install demo on client

You need install the demo on your local machine. This is the guide for a MacBook Pro installation of the client. Although DeepSpeech must be cloned, it does not need to be built or installed on the client.

To install the demo, enter:

```
brew install portaudio
pip install pyaudio
pip install pynput
git clone https://github.com/PaddlePaddle/DeepSpeech.git
cd DeepSpeech
```

The client listens for keypresses from the keyboard space and escape keys. This does not work on all devices including the MacBook Pro. To amend this, you can modify `deploy/demo_client.py` to use `ctrl` for record and `shift` for exit:

```
sed -i '' s/space/ctrl/g deploy/demo_client.py
sed -i '' 's/Key\.esc/Key\.shift/g' deploy/demo_client.py
```

7.2. Prepare demo server

On the server, set port 8000 to listen for connections:

```
apt-get -y install ufw
ufw allow ssh
ufw allow 8000
```

It is good practice to block ports that are not in use, which UFW does automatically.

7.3. Start demo server

Again, on the packet.net server enter the following to start the demo server and replace **SERVER_IP** below with the IP address of the server and run this from the DeepSpeech/ directory:

```
CUDA_VISIBLE_DEVICES=0 \  
python -u deploy/demo_server.py \  
--host_ip='SERVER_IP' \  
--host_port=8000 \  
--num_conv_layers=2 \  
--num_rnn_layers=3 \  
--rnn_layer_size=1024 \  
--alpha=1.15 \  
--beta=0.15 \  
--cutoff_prob=1.0 \  
--cutoff_top_n=40 \  
--use_gru=True \  
--use_gpu=False \  
--share_rnn_weights=False \  
--speech_save_dir='demo_cache' \  
--mean_std_path='models/baidu_en8k/mean_std.npz' \  
--vocab_path='models/baidu_en8k/vocab.txt' \  
--model_path='models/baidu_en8k/params.tar.gz' \  
--lang_model_path='models/lm/common_crawl_00.prune01111.trie.klm' \  
--decoding_method='ctc_beam_search' \  
--specgram_type='linear'
```

8 Run the speech-to-text demo

8.1. Run demo on client

On your host machine, replace **SERVER_IP** with the IP address of the server:

```
python -u deploy/demo_client.py --host_ip '<SERVER_IP>' --host_port 8000
```

After the client has connected, press and hold space (or ctrl if you modified the client demo) to talk. Once you release the space bar, a transcription of your speech will be sent to the server, processed, returned, and then printed. This takes around 4x the length of the speech itself. Press escape (or shift) to exit.

9 Next steps

The [Arm ecosystem](#) provides robust support for many state-of-the-art machine learning frameworks and applications. This demo would not be suitable for interactive assistant speech recognition, but with 96 cores available on a Cavium Thunder X server such as the one used here, 24 hours of English or Mandarin speech can be transcribed with state-of-the-art accuracy for just \$0.50!

More exciting use cases will continue to develop as an increasingly wide range of next-generation Arm servers become available on the cloud.

Watch this space!