

AWS-F1 ふたたび

<https://github.com/Xilinx/SDAccel-Tutorials/blob/master/docs/aws-getting-started/RTL/README.md>

このドキュメント通りだと f1 インスタンスで Vivado の合成とか動かす必要があって、あまり経済的じゃない気がする。

準備

- AWS コンソールにログイン
 - <https://console.aws.amazon.com/> から
- リージョンを選ぶ
 - US East(N.Virginia) にした。US West(Oregon), EU(Ireland) でもいいみたい
- S3 バケット作る
 - 「すべてのサービス」-> ストレージの下の S3 を選択
 - "+ バケットを作成する " ボタンをクリック
 - バケット名をつける。(aws-f1-test とか)
 - あとはデフォルトのまま " 次へ " " 作成 "
- S3 バケットにフォルダを作る
 - 作った S3 バケットをクリック
 - "+ フォルダの作成 " ボタンをクリック
 - フォルダ名をつけて保存。(SDAccel とつけた)
- プライベートキーの準備
 - <http://console.aws.amazon.com/ec2> にアクセス
 - 左ペインのキーペアをクリック
 - " キーペアの作成 " ボタンをクリック
 - 名前をつけてキーを作成 (aws-f1-test-key とつけた)
 - aws-f1-test-key.pem がローカルに保存されるので、パーミッションを 400 に。
- AWS F1 インスタンス使えるようにリクエストする必要がある
 - <http://aws.amazon.com/contact-us/ec2-request> から
 - Submit a 'Service Limit Increase' for 'EC2 Instances'
 - Select the region where you want to access F1 instances: US East (N.Virginia), US West (Oregon) or EU (Ireland)
 - Select 'f1.2xlarge' or 'f1.16xlarge' as the primary instance type
 - Set the 'New limit value' to 1 or more
 - Fill the rest of the form as appropriate and click 'Submit'

F1 インスタンスのテスト

- F1 インスタンスつくる
 - <https://console.aws.amazon.com/ec2> にアクセス
 - " インスタンスの作成 " ボタンをクリック - リージョンに注意。US East, US West, EU(アイルランド) とか。
 - 左メニューの AWS Marketplace をクリック。検索フォームに FPGA と入力して検索する。
 - FPGA Developer AMI の " 選択 " ボタンをクリック
 - 料金メニューがあるので "Continue" ボタンをクリック
 - f1.2xlarge インスタンスを探してセレクトボックスをチェック
 - " 次の手順: インスタンスの詳細の設定 " ボタンを選択 - " 確認と作成 " ではない
 - 詳細設定, ストレージの追加, タグの追加は, デフォルトのまま, 次に。
 - セキュリティグループの設定
 - " ルールの追加 " ボタンをクリックして, タイプ :RDP, プロトコル :TCP, ポート範囲 :3389, ソース :カスタム -0.0.0.0/0 を設定
 - " 確認と作成 " ボタンをクリック
 - 確認画面が表示されるので " 起動 " ボタンをクリック

- ・ 「既存のキーペアを選択するか、新しいキーペアを作成します。」 ウィンドウが開く
 - ・ "既存のキーペアの選択" を選択し、先に作ったキーペア (aws-f1-test-key) を選択
 - ・ 注意事項を読んでチェックする
 - ・ "インスタンスの作成" ボタンをクリック
- ・ "インスタンスの表示" ボタンをクリックして、インスタンス動作の確認
- ・ インスタンスを選択すると、下ペインに IP アドレスなどの情報が表示される
- ・ インスタンスにログイン
 - ・ ssh -i pem ファイル centos@IP アドレス で ログイン できる
 - ・ GUI を起動する
 - ・ source <(curl -s https://s3.amazonaws.com/aws-fpga-developer-ami/1.5.0/Scripts/setup_gui.sh)
 - ・ スクリプト終了後にパスワードが表示されるので、忘れずにメモ。
 - ・ - ssh でログインして sudo passwd centos とかしてパスワードは変更できる
 - ・ おわったら再起動
 - ・ RDP で接続
 - ・ color depth は 24-bit

SDAccel のための設定

- ・ ターミナルを開いて aws configure を実行
 - ・ アクセスキー ID とシークレットアクセスキーは https://console.aws.amazon.com/iam/home?#/security_credentials で作成
 - ・ region name: us-east-1 , output format: json で設定
- ・ 関連リソースの取得と環境設定

```
git clone https://github.com/aws/aws-fpga.git $AWS_FPGA_REPO_DIR
cd $AWS_FPGA_REPO_DIR
source sdaccel_setup.sh
```

- ・ 動作確認
 - ・ helloworld_ocl のソフトウェアエミュレーション

```
cd $SDACCEL_DIR/examples/xilinx/getting_started/host/helloworld_ocl/
make clean
make check TARGETS=sw_emu DEVICES=$AWS_PLATFORM all
```

- ・ helloworld_ocl のハードウェアエミュレーション

```
cd $SDACCEL_DIR/examples/xilinx/getting_started/host/helloworld_ocl/
make clean
make check TARGETS=hw_emu DEVICES=$AWS_PLATFORM all
```

- ・ FPGA バイナリの作成

```
cd $SDACCEL_DIR/examples/xilinx/getting_started/host/helloworld_ocl/
make clean
make TARGETS=hw DEVICES=$AWS_PLATFORM all
```

- ・ AFI イメージの作成

```
cd xclbin
$SDACCEL_DIR/tools/create_sdaccel_afi.sh ¥
-xclbin=<xclbin file name>.xclbin ¥
-s3_bucket=<bucket-name> ¥
-s3_dcp_key=<dcp-folder-name> ¥
-s3_logs_key=<logs-folder-name>
```

- ・ 今回の例だと

```
cd xclbin
$SDACCEL_DIR/tools/create_sdaccel_afi.sh ¥
-xclbin=vector_addition.hw.xilinx_aws-vu9p-f1-04261818_dynamic_5_0.xclbin ¥
-s3_bucket=aws-f1-test ¥
-s3_dcp_key=SDAccel ¥
-s3_logs_key=SDAccel
```

- ・ ホームのディスク容量が足りなかったら (足りなかったので) /tmp とかに逃がしてもいい
- ・ AFI 作成プロセス
 - ・ vector_addition.hw.xilinx_aws-vu9p-f1-04261818_dynamic_5_0.awsxclbin ができてる
 - ・ 19_04_28-082150_afi_id.txt ができてる

```
{
  "FpgaImageId": "afi-070aeb46c51e01332",
  "FpgaImageGlobalId": "agfi-059ee6445563c3fc3"
}
```

- ・ AFI 作成プロセスがおわるのをまつ

```
aws ec2 describe-fpga-images --fpga-image-ids afi-070aeb46c51e01332
```

- ・ State -> Code のところが available になればいい
- ・ S3 バケットをみると, いろいろできているのがわかる.
- ・ FPGA での実行

```
cd $SDACCEL_DIR/examples/xilinx/getting_started/host/helloworld_ocl/
sudo sh
source /opt/xilinx/xrt/setup.sh
./helloworld
```