

Distributed Training of Deep Networks on Amazon Web Services* (AWS)

Ravi Panchumarthy (Intel), Thomas “Elvis” Jones (AWS), Andres Rodriguez (Intel), Joseph Spisak (Intel)

Deep neural networks are capable of amazing levels of representation power resulting in state-of-the-art accuracy in areas such as computer vision, speech recognition, natural language processing, and various data analytic domains. Deep networks require large amounts of computation to train, and the time to train is often days or weeks. Intel is optimizing popular frameworks such as Caffe*, TensorFlow*, Theano*, and others to significantly improve performance and reduce the overall time to train on a single node. In addition, Intel is adding or enhancing multi. node distributed training capabilities to these frameworks to share the computational requirements across multiple nodes and further reduce time to train. A workload that previously required days can now be trained in a matter of hours. [Read more about this.](#)

Amazon Web Services (AWS) Virtual Private Cloud (VPC) provides a great environment to facilitate multinode distributed deep network training. AWS and Intel partnered to create a simple set of scripts for creating clusters that allows developers to easily deploy and train deep networks, leveraging the scale of AWS. In this article, we provide the steps to set up the AWS CloudFormation* environment to train deep networks using the Caffe network.

AWS CloudFormation Setup

The following steps create a VPC that has an Elastic Compute Cloud (EC2) t2.micro instance as the AWS CloudFormation cluster (cfnccluster) controller. The cfnccluster controller is then used to create a cluster composed of a master EC2 instance and a number of compute EC2 instances within the VPC.

Steps to deploy the Cloudformation and cfnccluster

1. Use the AWS Management Console to launch the AWS CloudFormation (Figure 1).

Amazon Web Services

Compute

- EC2**
Virtual Servers in the Cloud
- EC2 Container Service**
Run and Manage Docker Containers
- Elastic Beanstalk**
Run and Manage Web Apps
- Lambda**
Run Code in Response to Events

Storage & Content Delivery

- S3**
Scalable Storage in the Cloud
- CloudFront**
Global Content Delivery Network
- Elastic File System**
Fully Managed File System for EC2
- Glacier**
Archive Storage in the Cloud
- Snowball**
Large Scale Data Transport
- Storage Gateway**
Hybrid Storage Integration

Database

- RDS**
Managed Relational Database Service

Developer Tools

- CodeCommit**
Store Code in Private Git Repositories
- CodeDeploy**
Automate Code Deployments
- CodePipeline**
Release Software using Continuous Delivery

Management Tools

- CloudWatch**
Monitor Resources and Applications
- CloudFormation**
Create and Manage Resources with Templates
- CloudTrail**
Track User Activity and API Usage
- Config**
Track Resource Inventory and Changes
- OpsWorks**
Automate Operations with Chef
- Service Catalog**
Create and Use Standardized Products
- Trusted Advisor**
Optimize Performance and Security

Security & Identity

- Identity & Access Management**
Manage User Access and Encryption Keys

Figure 1. CloudFormation in Amazon Web Services

2. Click **Create Stack**.
3. In the section labeled, Choose a template (Figure 2), select **Specify an Amazon S3 template URL**, and then enter https://s3.amazonaws.com/caffecfncluster/1.0/intelcaffe_cfncluster.template. Click **Next**.

Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage.

Design a template Use AWS CloudFormation Designer to create or modify an existing template. [Learn more](#)

Design template

Choose a template A template is a JSON-formatted text file that describes your stack's resources and their properties.

Select a sample template

Upload a template to Amazon S3

Choose File No file chosen

Specify an Amazon S3 template URL

[View/Edit template](#)

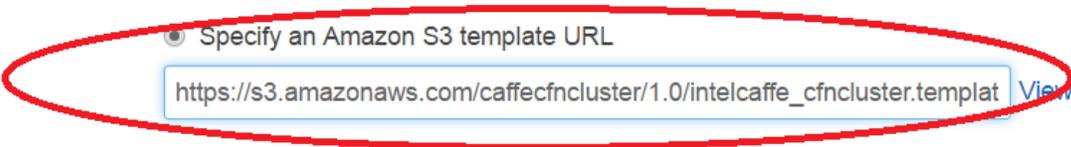


Figure 2. Entering the template URL.

4. Give the Stack a name, such as myFirstStack. Under **Select a key pair**, find the key pair you just named ([follow these instructions](#) if you need to create a key pair). Leave the rest of the Parameters as they are. Click **Next**.
5. Enter a Key, for example, name, and a Value, such as, cfnc1ustercaffe. Note that you can give any names to the key and value. The name does not have to match the key-pair from the previous step.
6. Click **Next**.
7. Review the stack, check the acknowledgement box, and then click **Create**. Creating the stacks will take a few minutes. Wait until the status of all three created stacks is CREATE_COMPLETE.

- The template used in Step 3 calls two other nested templates, creating a VPC with an EC2 t2.micro instance (Figure 3). Select the stack with the EC2 instance, and then select **Resources**. Click the Physical ID of the cfncclusterMaster.

The screenshot shows the AWS CloudFormation console. At the top, there are navigation tabs for 'AWS', 'Services', and 'Edit'. Below these are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter section shows 'Filter: Active' and a search box 'By Name:'. The main table lists stacks with columns for 'Stack Name', 'Created Time', 'Status', and 'Description'. The first stack, 'myFirstStack-InstanceStack-V8', is selected, indicated by a red circle around its checkbox. Below the stack list, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', and 'Change Set'. The 'Resources' tab is active and shows a table with columns for 'Logical ID', 'Physical ID', and 'Type'. The first row shows 'cfncclusterMaster' with a Physical ID of 'i-2b402f84', which is circled in red. The second row shows 'cfncclusterMasterSecurityGro...' with a Physical ID of 'sg-e7e9de81'.

Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/> myFirstStack-InstanceStack-V8	2016-08-09 22:07:53 UTC-0700	CREATE_COMPLETE	Step 3 of 3: Cre
<input type="checkbox"/> myFirstStack-VPCStack-1A16F	2016-08-09 22:06:47 UTC-0700	CREATE_COMPLETE	Step 2 of 3: Cre
<input type="checkbox"/> myFirstStack	2016-08-09 22:06:41 UTC-0700	CREATE_COMPLETE	Step 1 of 3: This

Logical ID	Physical ID	Type
cfncclusterMaster	i-2b402f84	AWS::EC2::Instanc
cfncclusterMasterSecurityGro...	sg-e7e9de81	AWS::EC2::Securit

Figure 3. Selecting the Physical ID from the Resources tab.

- This will take you to AWS EC2 console (Figure 4). Under Description, note the VPC ID and the Subnet ID as you'll need them in a later step. Right-click on the instance, select **Connect** and follow the instructions.

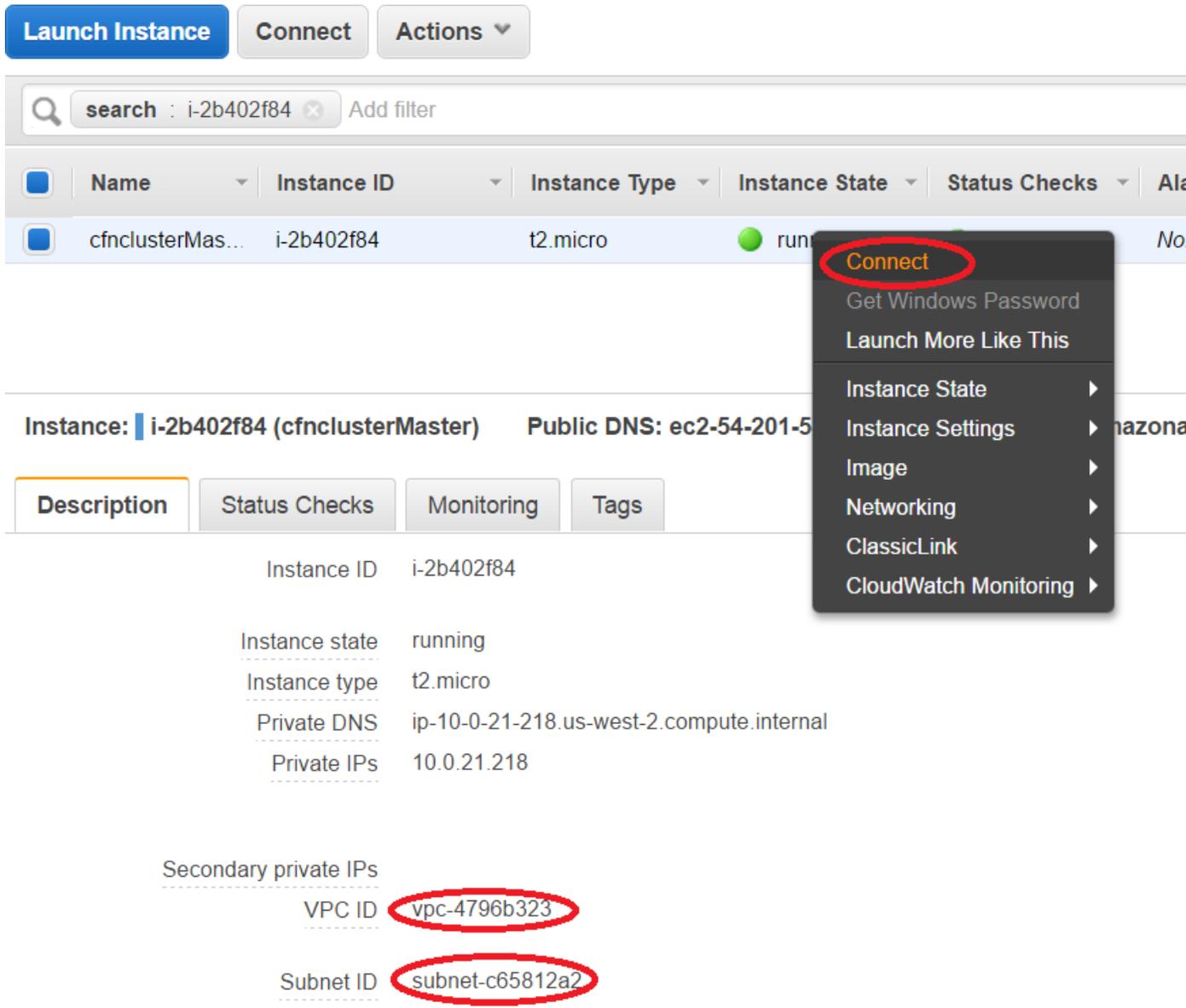


Figure 4. AWS EC2 console.

10. Once you ssh into the instance, prepare to modify the cluster’s configuration with the following commands:

```
cd .cfnccluster
cp config.edit_this_cfnccluster_config config
vi config
```

11. Follow the comments in the config file (opened with the final command in Step 9) to fill in the appropriate information.

- Obtain your `AWS_ACCESS_KEY` and `AWS_SECRET_ACCESS_KEY` from your AWS system administrator if you don’t have one already or [follow these steps to obtain them](#).

Note that while the master node is not labelled as a compute node, it also acts as a compute node. Therefore, if the total number of nodes to be used in training is 32, then choose a `queue_size = 31` compute nodes.

- Use the VPC ID and Subnet ID obtained in Step 8.
- The latest `custom_ami` to use should be `ami-77aa6117`; this article will be updated when newer AMI are provided.

12. Launch a cluster with the command: `cfnc1uster create <vpc_name_chosen_in_config_file>`. This will launch more AWS CloudFormation templates. You can see them via the AWS CloudFormation page in the AWS Management Console.

Sample Scripts to Train a Few Popular Networks

After the cloud-formation-setup is complete, if you configured the size of the cluster to be N , there will be $N+1$ instances created (1 master node and N compute nodes). Note that the master node is also treated as a compute node. The created cluster has a shared drive among all $N+1$ instances. The instances contain intelcaffe, Intel® Math Kernel Library (Intel® MKL) and sample scripts to train CIFAR-10 and GoogLeNet. To start training a sample network, login into the master node.

To start training a CIFAR-10 model with provided solver and `train_val` prototxt files, run:

```
cd ~/scripts/  
./aws_ic_mn_run_cifar.sh
```

To start training a GoogLeNet model, you should download ImageNet dataset and configure the variables `path_to_imagenet_train_folder`, `batchsize_pernode` and others if required in the script and run the `./aws_ic_mn_run_googlenet.sh` script:

```
cd ~/scripts/  
#Edit variables path_to_imagenet_train_folder, batchsize_pernode and  
others if required  
vi ./aws_ic_mn_run_googlenet.sh  
./aws_ic_mn_run_googlenet.sh
```

The script `aws_ic_mn_run_cifar.sh` creates a hosts file (`~/hosts.aws`) which contains all the IP addresses of the instances in your VPC. It then updates the solver and `train_val` prototxt files located in `~/models/cifar10/`. You could modify these prototxt files to suit your training requirements. The `aws_ic_mn_run_cifar.sh` script will start the data server, which will provide data to the compute nodes. There will be a little overhead on the master with data server running along with the compute. After the data server is launched, the distributed training is launched using the `mpirun` command.

The script `aws_ic_mn_run_googlenet.sh` creates a hosts file (`~/hosts.aws`) which contains all the IP addresses of the instances in your VPC. Unlike, the CIFAR-10 example where

the data server provides the data, in GoogLeNet training, each worker will read its own data. The script will create separate solver, train_val prototxt files and train.txt files for each worker based on the template solver and train_val prototxt located in `~/models/googlenet/`. You could modify these template prototxt files to suit your training requirements. The `aws_ic_mn_run_googlenet.sh` script will then launch the job using the mpirun command.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others. © Intel Corporation.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

For more information go to <http://www.intel.com/performance>.