NVIDIA Clara Train SDK on the AWS Cloud

Quick Start Reference Deployment

April 2020

Anas Abidin, NVIDIA
Andy Schuetz and Aaron Friedman, Amazon Web Services

Visit our GitHub repository for source files and to post feedback, report bugs, or submit feature ideas for this Quick Start.

Contents

This Quick Start was created by NVIDIA in collaboration with Amazon Web Services (AWS).

Quick Starts are automated reference deployments that use AWS CloudFormation templates to deploy key technologies on AWS, following AWS best practices.

Overview

Clara Train is NVIDIA's domain-optimized application-development framework for medical-imaging researchers and artificial intelligence (AI) developers. This Quick Start reference deployment guide provides step-by-step instructions for deploying NVIDIA Clara Train SDK in a highly available (HA) configuration on the AWS Cloud. Clara Train includes an AI Assisted Annotation developer toolkit that can be integrated into existing medical viewers, accelerating the creation of AI-ready, annotated medical-imaging datasets. Clara Train also provides a TensorFlow-based training framework with domain-specific pretrained models that accelerate AI development with techniques like transfer learning, federated learning, and automated machine learning. Models trained with Clara Train are packaged as Medical Model Archives (MMARs), which provide a standardized format for training workflows and collaborations.

This Quick Start is for developers who want to use tools and APIs to carry out AI development. It's also for health IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their Clara Train SDK workloads to the AWS Cloud.

Please know that we may share who uses AWS Quick Starts with the AWS Partner Network (APN) Partner that collaborated with AWS on the content of the Quick Start.

NVIDIA Clara Train SDK on AWS

This deployment provides scalable access to NVIDIA V100 Tensor Core graphics processing units (GPUs) and the Amazon Elastic Compute Cloud (Amazon EC2) P3 instance type, with pay-as-you-go pricing. This deployment is based on Amazon Elastic Container Service (Amazon ECS) and Amazon EC2. Amazon Elastic File System (Amazon EFS) is used for storage shared between containers.

IMPORTANT: PLEASE READ

Customers are solely responsible for determining whether any federal, state, or local regulations apply to them and, if so, implement the required procedural and technical controls required of them by those regulations. AWS does not provide legal or compliance advice. Customers should consult with qualified legal counsel or consultants, as needed to ensure that their use of AWS complies with applicable laws, rules, and regulations.

This deployment of the NVIDIA Clara Train SDK is meant to be run on de-identified medical images. For additional security, customers have the ability to implement encryption in-transit throughout the AWS environment deployed as part of this Quick Start.

Cost and licenses

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There are no additional costs for using the NVIDIA Clara Train SDK or for using the Quick Start.

The AWS CloudFormation template for this Quick Start includes configuration parameters that you can customize. Some of these settings, such as instance type, affect the cost of deployment. For cost estimates, see the pricing pages for each AWS service you will use. Prices are subject to change.

Tip: After you deploy the Quick Start, we recommend that you enable the AWS Cost and Usage Report. This report delivers billing metrics to an Amazon Simple Storage Service (Amazon S3) bucket in your account. It provides cost estimates based on usage throughout each month and finalizes the data at the end of the month. For more information about the report, see the AWS documentation.

By using this Quick Start, and by using the Clara Train SDK container and download models, you accept the terms and conditions of the included licenses. Licenses are available with the NVIDIA Clara Train SDK documentation and the model application .zip files.

Architecture

Deploying this Quick Start for a new virtual private cloud (VPC) with **default parameters** builds the following NVIDIA Clara Train SDK environment in the AWS Cloud.

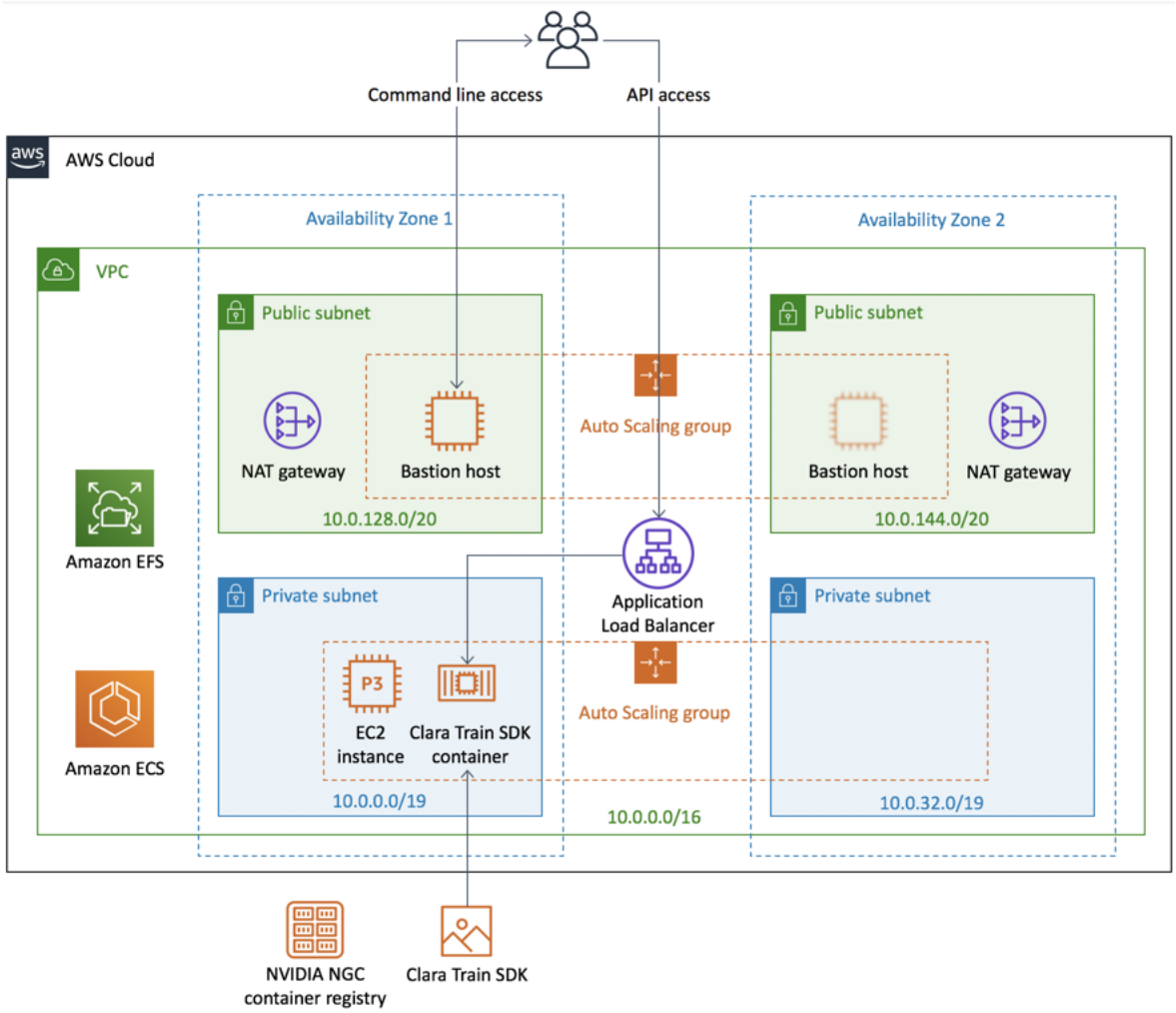


Figure 1: Quick Start architecture for NVIDIA Clara Train SDK on AWS

As shown in Figure 1, the Quick Start sets up the following:

- A highly available architecture that spans two Availability Zones.*
- A VPC configured with public and private subnets, according to AWS best practices, to provide you with your own virtual network on AWS.*
- In the public subnets:
  - Managed network address translation (NAT) gateways to allow outbound internet access for resources in the private subnets.*
  - A Linux bastion host in an Auto Scaling group to allow inbound Secure Shell (SSH) access to Amazon EC2 host instances in the private subnets.*
- In the private subnets:
  - One instance of the NVIDIA Clara Train SDK container deployed with the Amazon EC2 launch type, on a GPU-enabled Amazon EC2 instance, in an Auto Scaling group.
  - One P3 Amazon EC2 instance.
- Amazon EFS, a fully managed elastic network file system (NFS) to persist and share data across container instances.
- Amazon ECS, a fully managed container orchestration service for running and managing Docker containers on a cluster.
- An Application Load Balancer to route traffic to the NVIDIA Clara Train APIs and over HTTPS.

* The template that deploys the Quick Start into an existing VPC skips the components marked by asterisks and prompts you for your existing VPC configuration.

Planning the deployment

Specialized knowledge

This Quick Start assumes familiarity with containers and with Amazon ECS.

This deployment guide also requires a moderate level of familiarity with AWS services. If you're new to AWS, visit the Getting Started Resource Center and the AWS Training and Certification website. These sites provide materials for learning how to design, deploy, and operate your infrastructure and applications on the AWS Cloud. For more information about AWS services used in this Quick Start, see the Additional resources section.

To learn more about deep learning tools and AI-enabled medical-imaging workflows, see the NVIDIA Clara documentation.

For information about compliance on AWS, see the AWS Compliance page on the AWS website.

AWS account

If you don't already have an AWS account, create one at https://aws.amazon.com by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use.

Technical requirements

Before you launch the Quick Start, your account must be configured as specified in the following table. Otherwise, deployment might fail.

Resources	If necessary, request service quota increases for the following resources. You might need to do this if an existing deployment uses these resources, and you might exceed the default quotas with this deployment. The Service Quotas console displays your usage and quotas for some aspects of some services. For more information, see the AWS documentation.

Resource	This deployment uses
VPCs	1
AWS Identity and Access Management (IAM) security groups	4
IAM roles	3
Auto Scaling groups	2
Application Load Balancers	1
t2.micro instances	1
p3.2xlarge instances	1

Regions	This deployment includes Amazon EFS, which isn't currently supported in all AWS Regions. For a current list of supported Regions, see the endpoints and quotas webpage.

Key pair	Make sure that at least one Amazon EC2 key pair exists in your AWS account in the Region where you plan to deploy the Quick Start. Make note of the key pair name. You need it during deployment. To create a key pair, follow the instructions in the AWS documentation. For testing or proof-of-concept purposes, we recommend creating a new key pair instead of using one that's already being used by a production instance.
IAM permissions	Before launching the Quick Start, you must log in to the AWS Management Console with IAM permissions for the resources and actions the templates deploy. The *AdministratorAccess* managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions.

Deployment options

This Quick Start provides two deployment options:

- **Deploy NVIDIA Clara Train SDK into a new VPC (end-to-end deployment)**. This option builds a new AWS environment consisting of the VPC, subnets, NAT gateways, security groups, bastion hosts, and other infrastructure components. It then deploys NVIDIA Clara Train SDK into this new VPC.
- **Deploy NVIDIA Clara Train SDK into an existing VPC**. This option provisions NVIDIA Clara Train SDK in your existing AWS infrastructure.

The Quick Start provides separate templates for these options. It also lets you configure Classless Inter-Domain Routing (CIDR) blocks, instance types, and the number of NVIDIA Clara Train SDK service instances, as discussed later in this guide.

Deployment steps

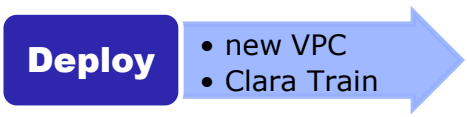
Step 1. Sign in to your AWS account

1. Sign in to your AWS account at https://aws.amazon.com with an IAM user role that has the necessary permissions. For details, see Planning the deployment earlier in this guide.
2. Make sure that your AWS account is configured correctly, as discussed in the Technical requirements section.

Step 2. Launch the Quick Start

> **Note:** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. For full details, see the pricing pages for each AWS service used by this Quick Start. Prices are subject to change.

1. Sign in to your AWS account, and choose one of the following options to launch the AWS CloudFormation template. For help with choosing an option, see Deployment options earlier in this guide.



Deploy NVIDIA Clara Train SDK into a new VPC on AWS

Deploy NVIDIA Clara Train SDK into an existing VPC on AWS

> **Important:** If you're deploying NVIDIA Clara Train SDK into an existing VPC, make sure that your VPC has two private subnets in different Availability Zones for the EC2 instances, and that the subnets aren't shared. This Quick Start doesn't support shared subnets. These subnets require NAT gateways in their route tables to allow the instances to download packages and software without exposing them to the internet.
>
> Also, make sure that the domain name option in the DHCP options is configured as explained in the Amazon VPC documentation. You provide your VPC settings when you launch the Quick Start.

   Each deployment takes about 30 minutes to complete.

2. Check the AWS Region that's displayed in the upper-right corner of the navigation bar, and change it if necessary. This is where the network infrastructure for NVIDIA Clara Train SDK will be built. The template is launched in the US East (Ohio) Region by default.

> **Note:** This deployment includes Amazon EFS, which isn't currently supported in all AWS Regions. For a current list of supported Regions, see the endpoints and quotas webpage.

3. On the **Create stack** page, keep the default setting for the template URL, and then choose **Next**.
4. On the **Specify stack details** page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary.

   In the following tables, parameters are listed by category and described separately for the two deployment options:

   - Parameters for deploying NVIDIA Clara Train SDK into a new VPC
   - Parameters for deploying NVIDIA Clara Train SDK into an existing VPC

   When you finish reviewing and customizing the parameters, choose **Next**.

OPTION 1: PARAMETERS FOR DEPLOYING NVIDIA CLARA TRAIN SDK INTO A NEW VPC

View template

VPC network configuration:

Parameter label (name)	Default	Description
Availability Zones (`AvailabilityZones`)	*Requires input*	Choose the Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list.
VPC CIDR (`VPCCIDR`)	10.0.0.0/16	CIDR block for the VPC.
Private subnet 1 CIDR (`PrivateSubnet1CIDR`)	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
Private subnet 2 CIDR (`PrivateSubnet2CIDR`)	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.
Public subnet 1 CIDR (`PublicSubnet1CIDR`)	10.0.128.0/20	CIDR block for the public subnet located in Availability Zone 1.
Public subnet 2 CIDR (`PublicSubnet2CIDR`)	10.0.144.0/20	CIDR block for the public subnet located in Availability Zone 2.

Linux bastion configuration:

Parameter label (name)	Default	Description
Bastion key pair name (BastionKeyPairName)	*Requires input*	Enter the public/private key pair you created in your preferred AWS Region; see the Technical requirements section.
Allowed bastion external access CIDR (`BastionAccessCidr`)	*Requires input*	Enter the CIDR IP range that is permitted to access NVIDIA Clara Train SDK via the bastion host. This provides command line interface (CLI) access to the SDK. We recommend that you set this value to a trusted IP range. For example, you might want to grant only your corporate network access to the software.

EFS configuration:

Parameter label (name)	Default	Description
EFS deletion policy (`EFSDeletionPolicy`)	Delete	Retain or delete the Amazon EFS resources after CloudFormation stack deletion. The permitted values are Delete and Retain.
EFS performance mode (`PerformanceMode`)	generalPurpose	Select the performance mode of the EFS file system. The permitted values are generalPurpose and maxIO. The default, generalPurpose, is recommended for most applications.

ECS EC2 instance configuration:

Parameter label (name)	Default	Description
ECS EC2 Key pair name (`KeyName`)	*Requires input*	Enter the public/private key pair you created in your preferred AWS Region; see the Technical requirements section.
ECS EC2 instance type (`ECSHostInstanceType`)	p3.2xlarge	Choose the GPU instance type to use for the ECS hosts.

ECS cluster configuration:

Parameter label (name)	Default	Description
Allowed load balancer external access CIDR (`SourceCidrIP`)	*Requires input*	Enter the CIDR IP range that is permitted to access NVIDIA Clara Train SDK via the Application Load Balancer. We recommend that you set this value to a trusted IP range. For example, you might want to grant only your corporate network access to the software.
Enable end-to-end encryption of	HTTPS	Create an HTTPS listener on the Application Load Balancer, and launch the Clara Train API in Secure Sockets Layer (SSL)

Parameter label (name)	Default	Description
connections to the AI-Assisted Annotation service (`UseHTTPS`)		mode. This will provide end-to-end encryption of connections to the Clara Train API. Permitted values are **HTTP** (not recommended for production use) and **HTTPS**.
ARN for the ACM Certificate to use for TLS. Leave blank for HTTP connections (`CertificateID`)	*Optional*	Enter the Amazon Resource Name (ARN) of the server certificate to use with the Application Load Balancer. The Application Load Balancer uses an AWS Certificate Manager (ACM) SSL/TLS (Transport Layer Security) server certificate to terminate the front-end connection and then forward requests from clients to the targets. If `UseHTTPS` was set to **HTTPS**, encryption will be maintained to the targets. If `UseHTTPS` was set to **HTTP**, `CertificateID` should be an empty string, and API connections will not be encrypted.
Number of ECS hosts (`ClusterSize`)	1	Enter the number of ECS hosts to initially deploy.
Number of instances of this Clara Train task (`DesiredServiceCount`)	1	Enter the number of instances of the Clara Train SDK task to run across the cluster.

AWS Quick Start configuration:

Note: We recommend keeping the default settings for the following three parameters, unless you are customizing the Quick Start templates for your own deployment projects. Changing these parameter settings automatically updates code references to point to a new Quick Start location. For additional details, see the AWS Quick Start Contributor's Guide.

Parameter label (name)	Default	Description
Quick Start S3 bucket name (`QSS3BucketName`)	aws-quickstart	The S3 bucket you created for your copy of Quick Start assets if you decide to customize the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.
Quick Start S3 bucket region (`QSS3BucketRegion`)	us-east-1	The AWS Region where the Quick Start S3 bucket (QSS3BucketName) is hosted. When using your own bucket, you must specify this value.
Quick Start S3 key prefix (`QSS3KeyPrefix`)	quickstart-nvidia-imaging-clara-train/	The S3 key name prefix that is used to simulate a folder for your copy of Quick Start assets. You need to use this if you want to customize the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.

OPTION 2: PARAMETERS FOR DEPLOYING NVIDIA CLARA TRAIN SDK INTO AN EXISTING VPC

View template

Network configuration:

Parameter label (name)	Default	Description
VPC ID (VPC)	*Requires input*	Enter the ID of your existing VPC (e.g., vpc-0343606e).
VPC CIDR (VPCCIDR)	*Requires input*	CIDR block of your existing VPC (e.g. 10.180.0.0/16).
Private subnet 1 ID (PrivateSubnet1AID)	*Requires input*	Enter the ID of the private subnet in Availability Zone 1 in your existing VPC (e.g., subnet-a0246dcd).
Private subnet 2 ID (PrivateSubnet1AID)	*Requires input*	Enter the ID of the private subnet in Availability Zone 2 in your existing VPC (e.g., subnet-b58c3d67).
Public subnet 1 ID (PublicSubnet1ID)	*Requires input*	Enter the ID of the public subnet in Availability Zone 1 in your existing VPC (e.g., subnet-a0246dc9).
Public subnet 2 ID (PublicSubnet2ID)	*Requires input*	Enter the ID of the public subnet in Availability Zone 2 in your existing VPC (e.g., subnet-a0246dc1).

EFS configuration:

Parameter label (name)	Default	Description
EFS deletion policy (EFSDeletionPolicy)	Delete	Retain or delete the Amazon EFS resources after CloudFormation stack deletion. The permitted values are Delete and Retain.
EFS performance mode (PerformanceMode)	generalPurpose	Select the performance mode of the EFS file system. The permitted values are generalPurpose and maxIO. The default, generalPurpose, is recommended for most applications.

ECS EC2 instance configuration:

Parameter label (name)	Default	Description
Key pair name (KeyName)	*Requires input*	Enter the public/private key pair you created in your preferred AWS Region; see the Technical requirements section.
ECS EC2 instance type (ECSHostInstanceType)	p3.2xlarge	Choose the GPU instance type to use for the ECS hosts.

ECS cluster configuration:

Parameter label (name)	Default	Description
Allowed load balancer external access CIDR (SourceCidrIP)	*Requires input*	Enter the CIDR IP range that is permitted to access NVIDIA Clara Train SDK via the Application Load Balancer. We recommend that you set this value to a trusted IP range. For example, you might want to grant only your corporate network access to the software.
Enable end-to-end encryption of connections to the AI-Assisted Annotation service (UseHTTPS)	HTTPS	Create an HTTPS listener on the Application Load Balancer, and launch the Clara Train API in Secure Sockets Layer (SSL) mode. This will provide end-to-end encryption of connections to the Clara Train API. Permitted values are **HTTP** (not recommended for production use) and **HTTPS**.
ARN for the Certificate to use for TLS. Otherwise leave blank for HTTP (CertificateID)	*Optional*	Enter the Amazon Resource Name (ARN) of the server certificate to use with the Application Load Balancer. The Application Load Balancer uses an AWS Certificate Manager (ACM) SSL/TLS (Transport Layer Security) server certificate to terminate the front-end connection and then forward requests from clients to the targets. If UseHTTPS was set to **HTTPS**, encryption will be maintained to the targets. If UseHTTPS was set to **HTTP**, CertificateID should be an empty string, and API connections will not be encrypted.
Number of ECS hosts (ClusterSize)	1	Enter the number of ECS hosts to initially deploy.
Number of Clara services instances (DesiredServiceCount)	1	Enter the number of instances of the Clara Train SDK task to run across the cluster.
Linux bastion security group (BastionSecurityGroupID)	*Requires input*	Enter the security group of the bastion host, allowed to connect to private IP addresses of Amazon ECS hosts.

AWS Quick Start configuration:

> **Note:** We recommend keeping the default settings for the following three parameters, unless you are customizing the Quick Start templates for your own deployment projects. Changing these parameter settings automatically updates code references to point to a new Quick Start location. For additional details, see the AWS Quick Start Contributor's Guide.

Parameter label (name)	Default	Description
Quick Start S3 bucket name (`QSS3BucketName`)	aws-quickstart	The S3 bucket you have created for your copy of Quick Start assets if you decide to customize the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.
Quick Start S3 bucket region (`QSS3BucketRegion`)	us-east-1	The AWS Region where the Quick Start S3 bucket (QSS3BucketName) is hosted. When using your own bucket, you must specify this value.
Quick Start S3 key prefix (`QSS3KeyPrefix`)	quickstart-nvidia-imaging-clara-train/	The S3 key name prefix that is used to simulate a folder for your copy of Quick Start assets. You need to use this if you want to customize the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.

5. On the options page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you're done, choose **Next**.
6. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the two check boxes to acknowledge that the template creates IAM resources and might require the ability to automatically expand macros.
7. Choose **Create stack** to deploy the stack.
8. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the NVIDIA Clara Train SDK cluster is ready.
9. Use the URLs displayed in the **Outputs** tab for the stack, as shown in Figure 2, to view the Application Load Balancer and Amazon EFS that were created.

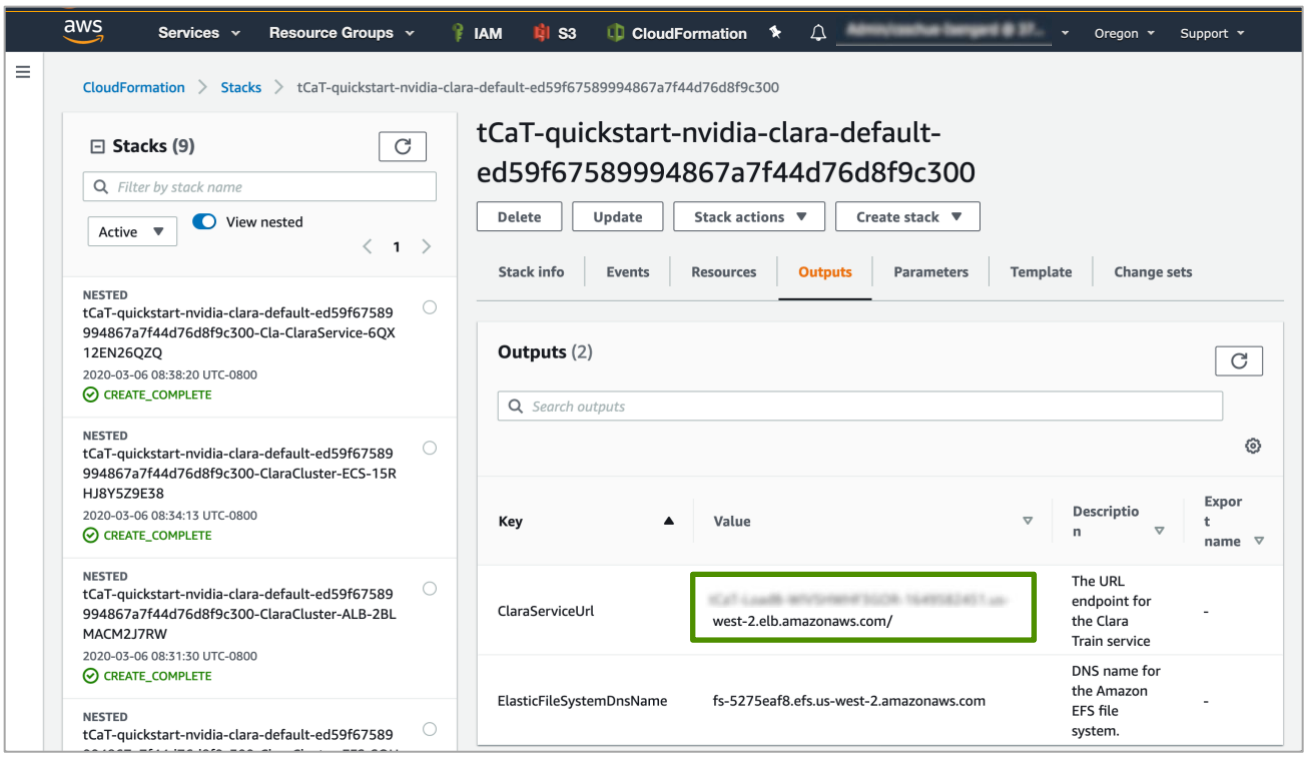


Figure 2: NVIDIA Clara Train SDK outputs after successful deployment

Step 3. Test the deployment

To test the NVIDIA Clara Train SDK deployment, point your browser to the URL of the Application Load Balancer displayed in the **Outputs** tab of the AWS CloudFormation console. If the deployment was successful, you will see the AI-Assisted Annotation API documentation, as shown in Figure 3.

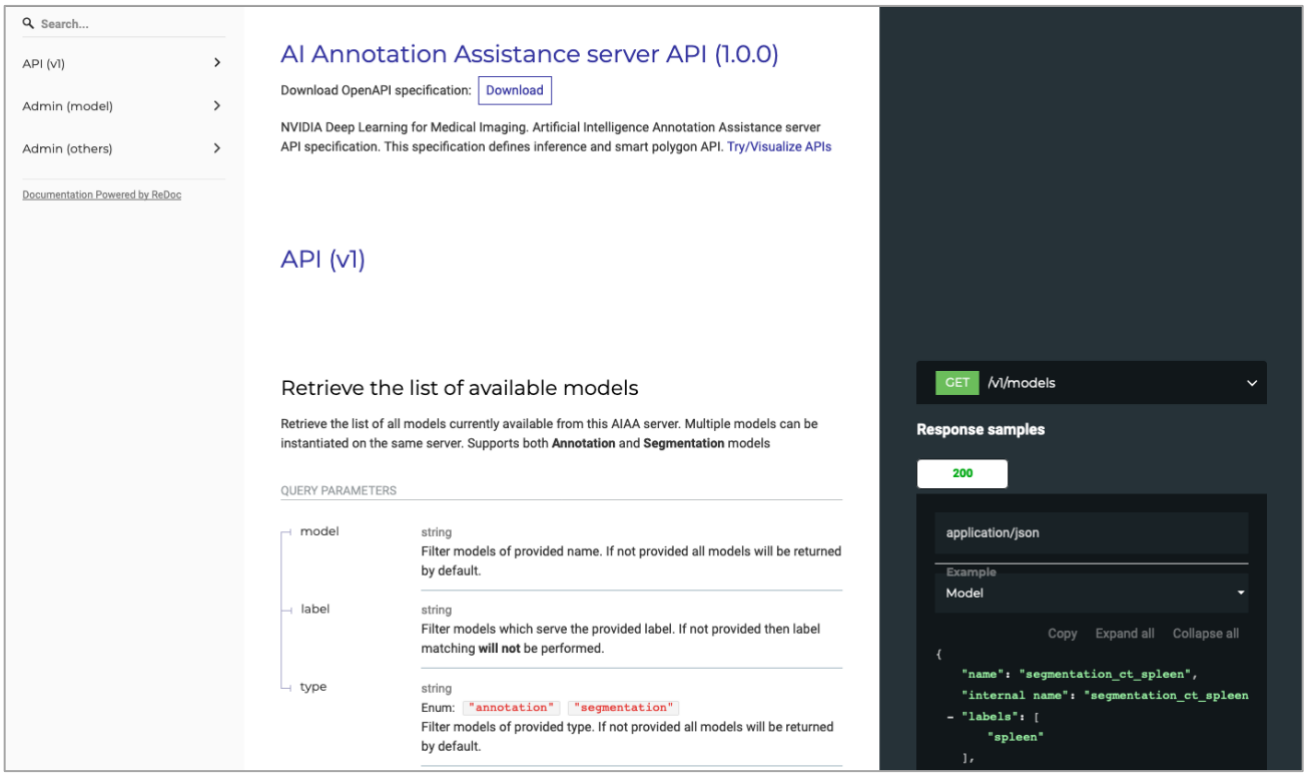


Figure 3. NVIDIA Clara Train SDK AI-Assisted documentation page after a successful deployment

Best practices for using NVIDIA Clara Train SDK on AWS

The NVIDIA Clara Train SDK deployed with this Quick Start provides web services and command-line tools to build and adapt medical-imaging AI models. The SDK requires an NVIDIA GPU, and the Quick Start provides a selection of suitable Amazon EC2 accelerated-computing instance types with compatible NVIDIA GPUs. The Quick Start will automatically deploy the latest Amazon ECS-optimized, GPU-enabled Amazon Machine Image (AMI) on the hosts, including suitable NVIDIA drivers.

You can reach the APIs and AI-Assisted Annotation Server at the Application Load Balancer URL displayed in the **Outputs** tab of the AWS CloudFormation console. Optionally, you can address the APIs and AI-Assisted Annotation Server with a static IP address by pointing an AWS Global Accelerator service accelerator to the Application Load Balancer. For the steps, see Getting Started with AWS Global Accelerator. We recommend using the default UseHTTPS setting to establish end-to-end encryption of connections to the API and AI-Assisted Annotation Server.

You can reach the command-line tools of the NVIDIA Clara Train SDK via SSH through the bastion host. (See Securely Connect to Linux Instances Running in a Private Amazon VPC.) To do this, follow these steps:

1. Connect by using SSH to the public IP address of the bastion host, using `ssh-add -K myPrivateKey.pem` to manage your private keys.
2. Connect by using SSH from the bastion host to the private IP address of the ECS host running the NVIDIA Clara Train SDK container. You can find the private IP address of the ECS host in the EC2 console.
3. Identify the NVIDIA Clara Train SDK container by running `docker ps`. Note the `CONTAINER ID`.
4. Connect to the container by running the command `docker exec -it <CONTAINER ID> /bin/bash`.

Security

If data containing PHI will be analyzed, the workload may fall within the scope of the U.S. Health Insurance Portability and Accountability Act (HIPAA). AWS does not provide legal or compliance advice. Customers should consult with qualified legal counsel or consultants, as needed, to ensure that their use of AWS complies with HIPAA, the terms of the AWS BAA, and other applicable laws, rules, and regulations.

The default deployment of this infrastructure provides HTTPS access to the APIs and AI-Assisted Annotation Server for improved security. It is possible, to permit unencrypted connections to the APIs and AI-Assisted Annotation Server for compatibility with existing tools, like 3D Slicer and the corresponding NVIDIA plugin. Unencrypted network access should only be permitted if non-PHI data is being analyzed.

Network access to the bastion host and Application Load Balancer should be restricted to CIDR blocks corresponding to your network. These controls are described above.

Data used by the NVIDIA Clara Train SDK containers is stored on an Amazon EFS file system that is encrypted by default, and we recommend not changing this setting.

The EC2 ECS hosts are deployed in a private subnet for improved security. We do not recommend permitting direct access from the internet to the Amazon ECS cluster. You can access the command-line tools of the SDK through the bastion host, as described above.

FAQ

Q. I encountered a **CREATE_FAILED** error when I launched the Quick Start.

A. If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack's state is retained and the instance is left running, so you can troubleshoot the issue.

> **Important:** When you set **Rollback on failure** to **No**, you continue to incur AWS charges for this stack. Please make sure to delete the stack when you finish troubleshooting.

For additional information, see Troubleshooting AWS CloudFormation on the AWS website.

Q. I encountered a size limitation error when I deployed the AWS CloudFormation templates.

A. We recommend that you launch the Quick Start templates from the links in this guide or from another S3 bucket. If you deploy the templates from a local copy on your computer or from a location other than an S3 bucket, you might encounter template size limitations. For more information about AWS CloudFormation quotas, see the AWS documentation.

Send us feedback

To post feedback, submit feature ideas, or report bugs, use the **Issues** section of the GitHub repository for this Quick Start. If you'd like to submit code, please review the Quick Start Contributor's Guide.

Additional resources

AWS resources

- Getting Started Resource Center
- AWS General Reference
- AWS Glossary

AWS services

- AWS CloudFormation

- Amazon EFS
- Amazon EC2
- Amazon ECS
- IAM
- AWS Certificate Manager
- Amazon VPC

NVIDIA Clara Train SDK documentation

- Clara-Train-SDK Release notes
- Clara Train SDK User Guide
- Clara Train SDK Blog

Other Quick Start reference deployments

- AWS Quick Start home page
- Reference Architecture for HIPAA on AWS
- Reference Architecture for HITRUST on AWS

Document revisions

Date	Change	In sections
April 2020	Initial publication	—

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The software included with this paper is licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at http://aws.amazon.com/apache2.0/ or in the "license" file accompanying this file. This code is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.