Code Along - Machine Learning on AWS

By Parag Pansare

Data Science Professional & AWS Cloud Architect

Agenda

- Overview:
  - What is Cloud Computing and AWS?
  - What is Machine Learning?

- Machine Learning using AWS on Amazon SageMaker:
  - Overview and Architecture
  - Hands on problem solving using ML on Python
  - Programming Model
  - K-Means Algorithm
  - Automatic Model Tuning/Hyperparameter Tuning

- Machine Learning using other AWS services

Prerequisite

- AWS account
- Basic understanding of Python and Jupyter Notebook
- Basic understanding of Machine Learning and AWS
- Healthy lunch ;)

What is Cloud Computing and AWS?

On-demand delivery of compute power, database storage, apps, and other IT resources through a cloud services platform via internet with pay-as-you-go pricing

Providers
AWS, MS Azure, GCP, Rackspace, Salesforce

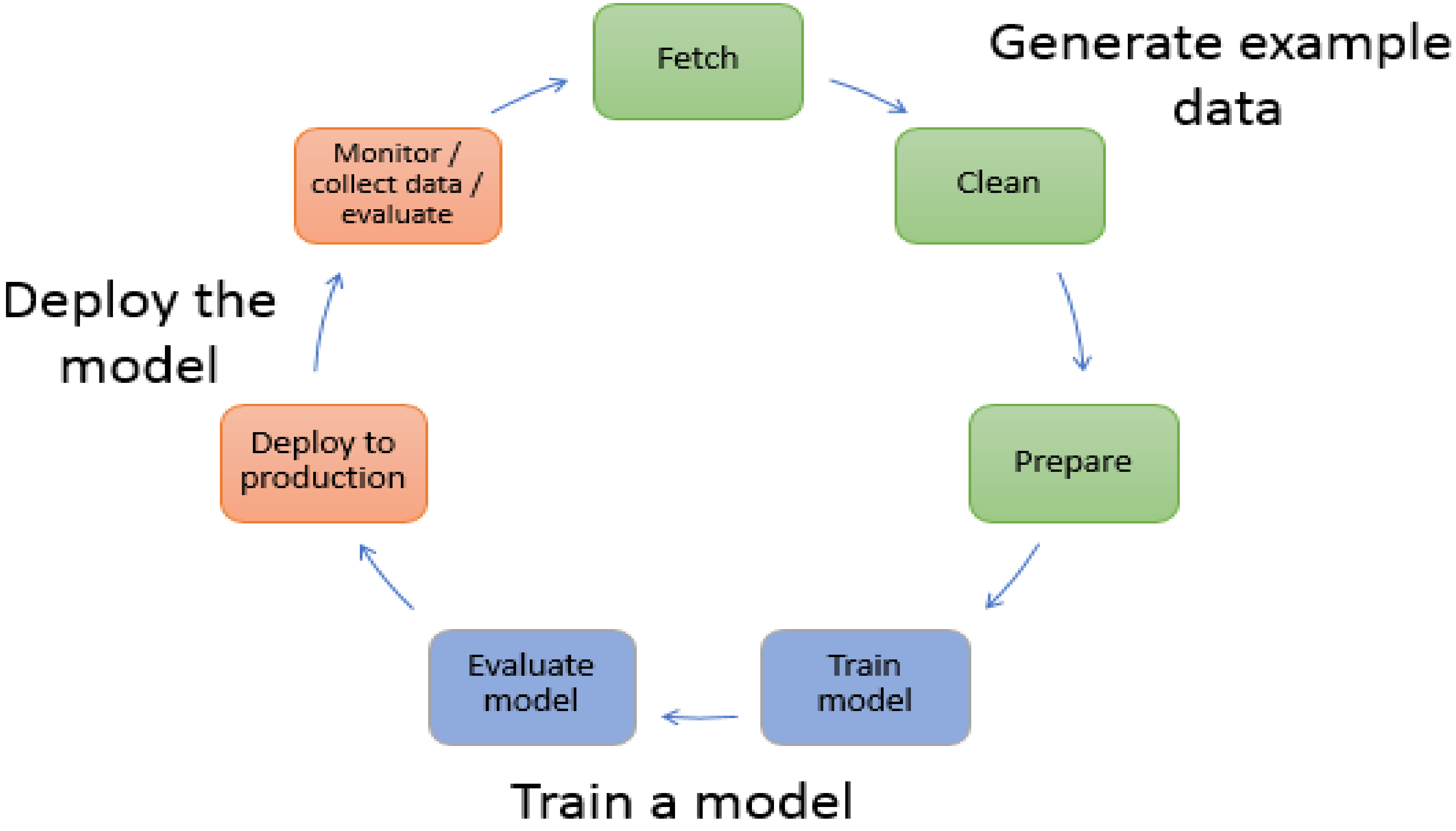
Advantages
Variable expense, scaling, HA, go global, increased speed and agility

Models
IaaS, PaaS, SaaS

Deployment Models
Cloud, Hybrid, On-premises

Solutions
Analytics, DevOps, Content Delivery, Mobile, ML, Serverless

What is Machine Learning?



Fetch

Clean

Prepare

Train model

Evaluate model

Deploy to production

Monitor / collect data / evaluate

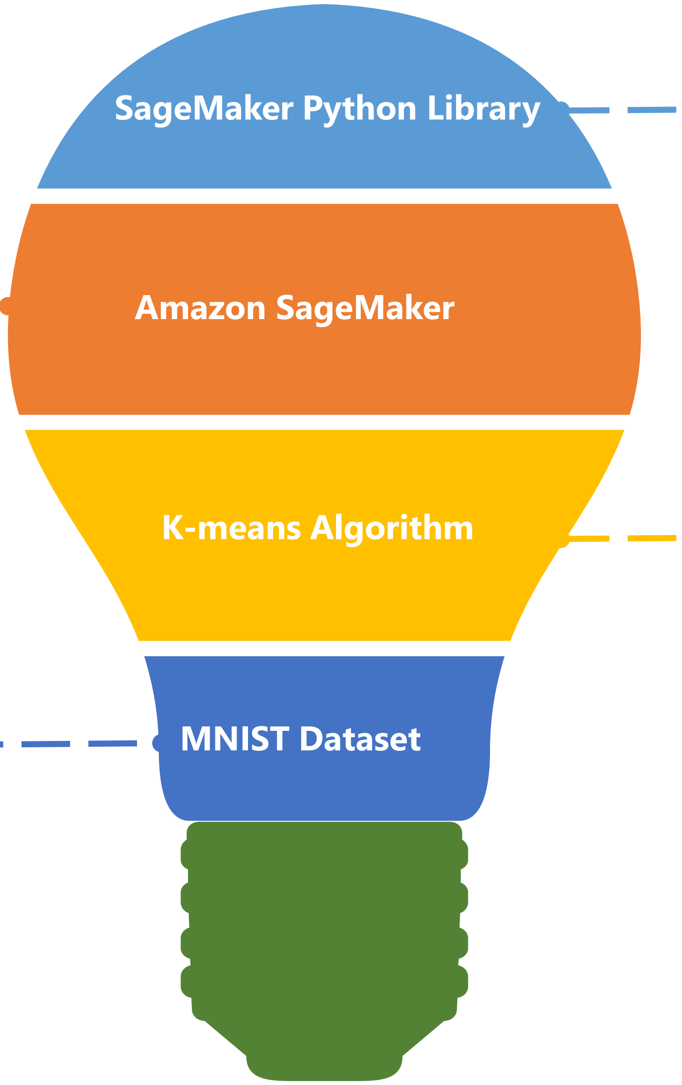
Generate example data

Train a model

Deploy the model

Problem Statement…

Classification problem to predict the number of a handwritten digit image



SageMaker Python Library

The high-level Python library provided by Amazon SageMaker

Amazon SageMaker

Fully managed machine learning service to build, train, and deploy machine learning models. Provides an integrated Jupyter authoring notebook instance.

K-Means Algorithm

A clustering algorithm to which this dataset is provided for model training. The algorithm groups the example data of handwritten numbers into K clusters, one for each number.

MNIST Dataset

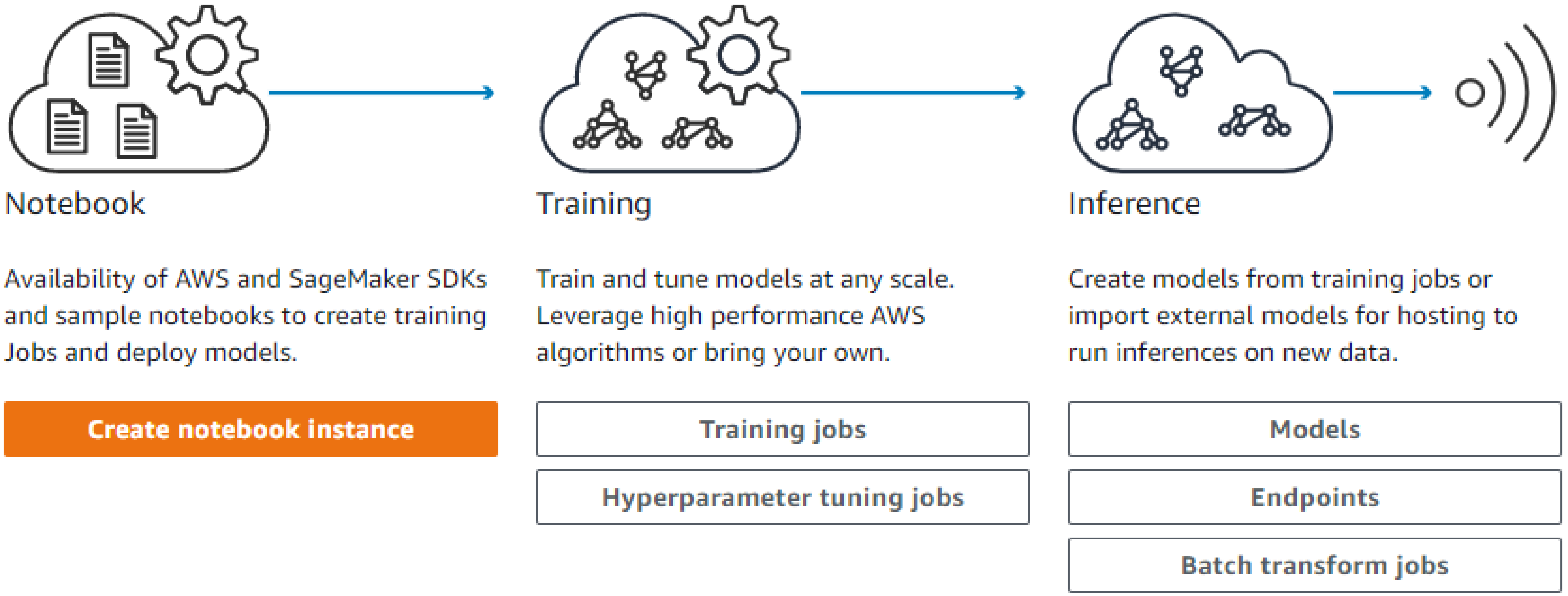
60,000 example images of handwritten single-digit numbers and a test dataset of 10,000 images

Machine Learning using Amazon SageMaker

What is Amazon SageMaker?

- Fully managed machine learning service
- Provides an integrated Jupyter authoring notebook instance
- Quickly and easily build, train, and deploy machine learning models
- Bring-your-own-algorithms and frameworks
- Provides common machine learning algorithms
- Secure and scalable environment
- Automatic Model Tuning/Hyperparameter Tuning

Amazon SageMaker - Overview



Notebook

Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.

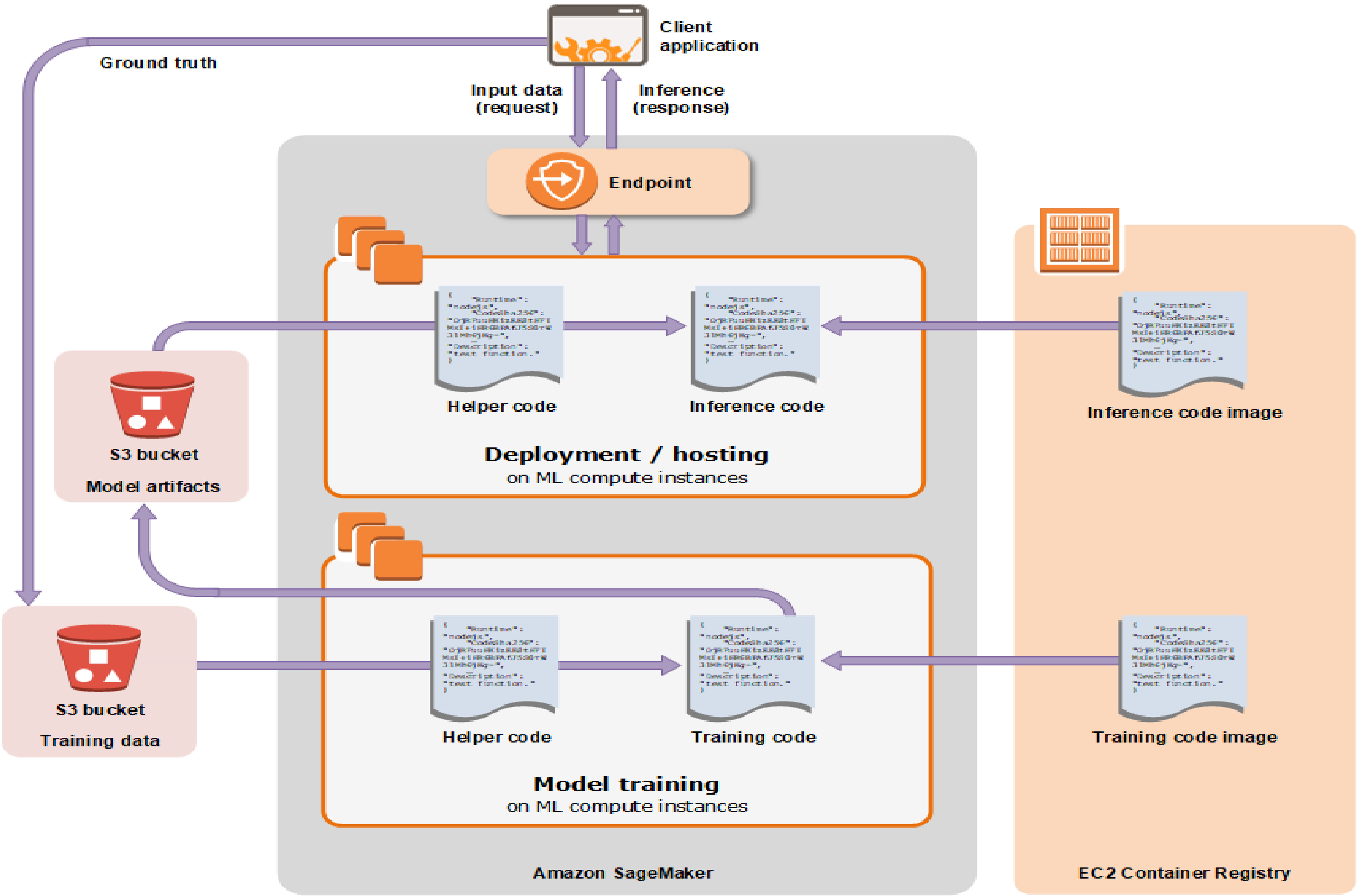
Training

Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.

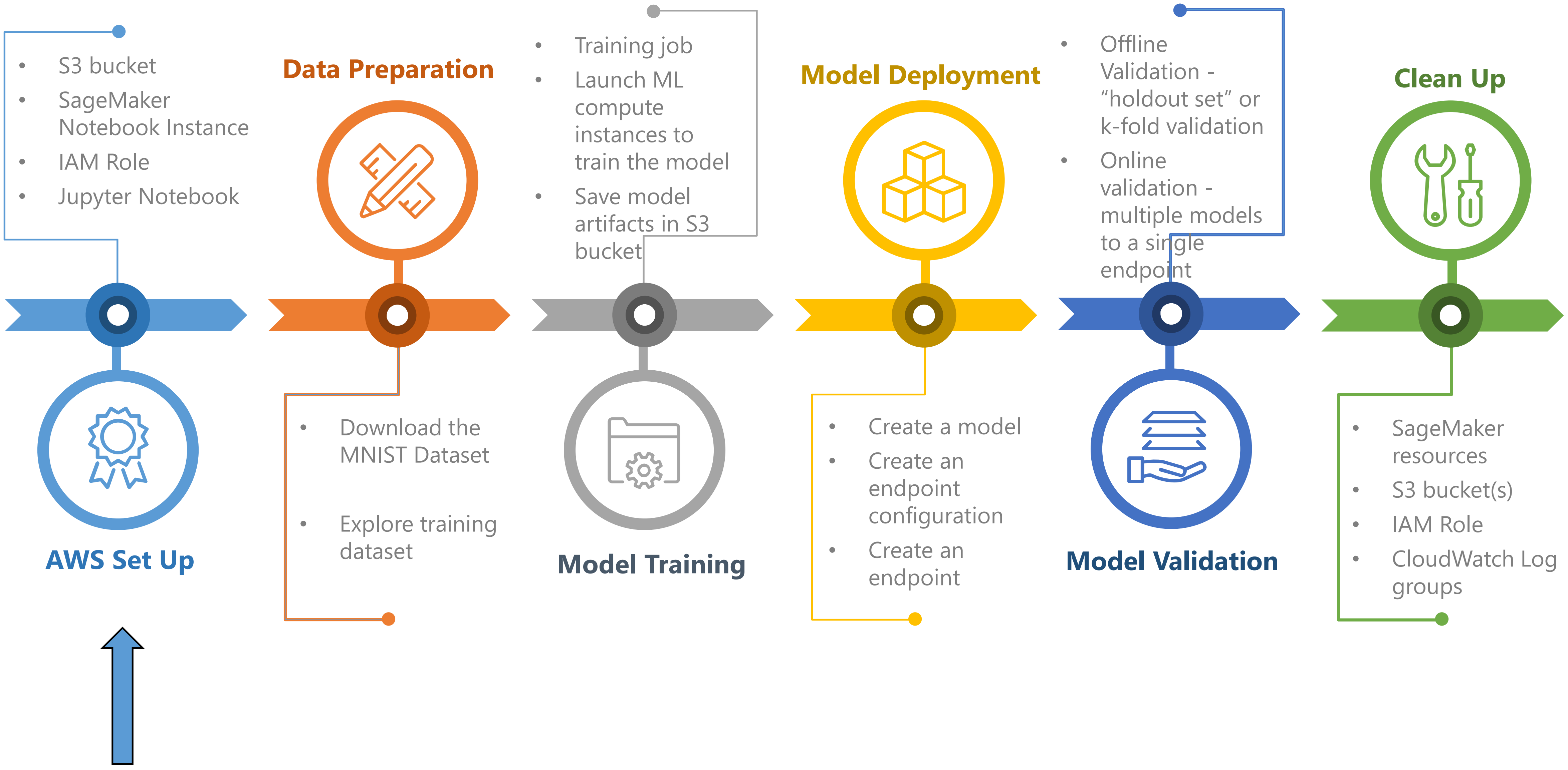
Inference

Create models from training jobs or import external models for hosting to run inferences on new data.

ML using Amazon SageMaker – Architectural Diagram



Steps in Machine Learning using Amazon SageMaker



AWS Set Up

- S3 bucket
- SageMaker Notebook Instance
- IAM Role
- Jupyter Notebook

Data Preparation

- Download the MNIST Dataset
- Explore training dataset

Model Training

- Training job
- Launch ML compute instances to train the model
- Save model artifacts in S3 bucket

Model Deployment

- Create a model
- Create an endpoint configuration
- Create an endpoint

Model Validation

- Offline Validation - "holdout set" or k-fold validation
- Online validation - multiple models to a single endpoint

Clean Up

- SageMaker resources
- S3 bucket(s)
- IAM Role
- CloudWatch Log groups

Step 1

Step 1 – AWS Set up…

- AWS S3 bucket:
  - to save model training data and artifacts
  - Include “sagemaker” in the bucket name

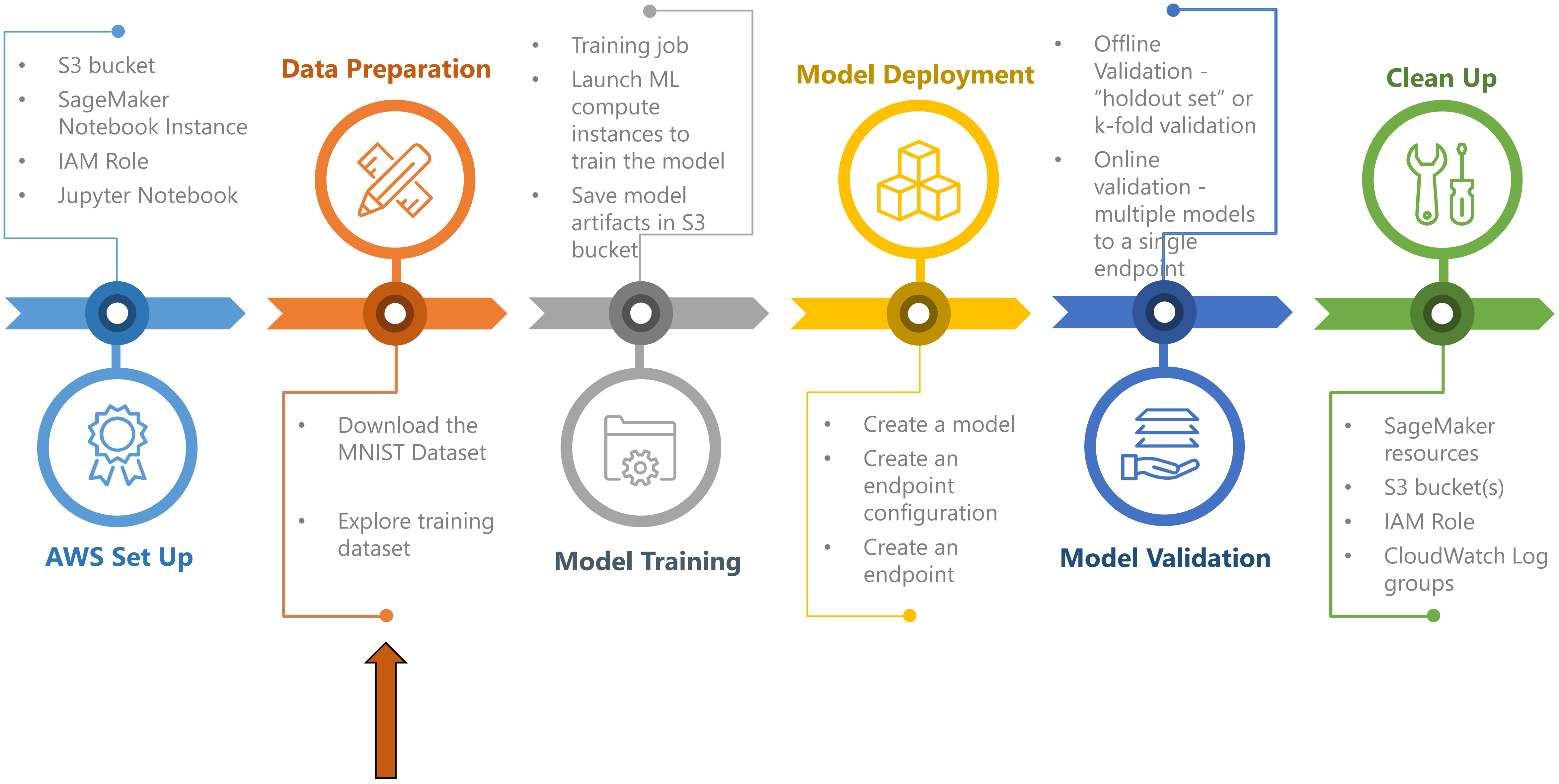
- Amazon SageMaker Notebook Instance:
  - fully managed machine learning (ML) EC2 compute instance running Jupyter Notebook App

- IAM Role (Identity and Access Management):
  - Allows SageMaker to perform actions in other AWS services on your behalf
  - Associate “AmazonSageMakerFullAccess” IAM policy to the IAM role you create

- When notebook instance is “InService”, open the Jupyter dashboard

- Create a Jupyter Notebook
  - Files tab → New → conda_python3 → Name the notebook

Steps in Machine Learning using Amazon SageMaker



AWS Set Up

- S3 bucket
- SageMaker Notebook Instance
- IAM Role
- Jupyter Notebook

Data Preparation

- Download the MNIST Dataset
- Explore training dataset

Model Training

- Training job
- Launch ML compute instances to train the model
- Save model artifacts in S3 bucket

Model Deployment

- Create a model
- Create an endpoint configuration
- Create an endpoint

Model Validation

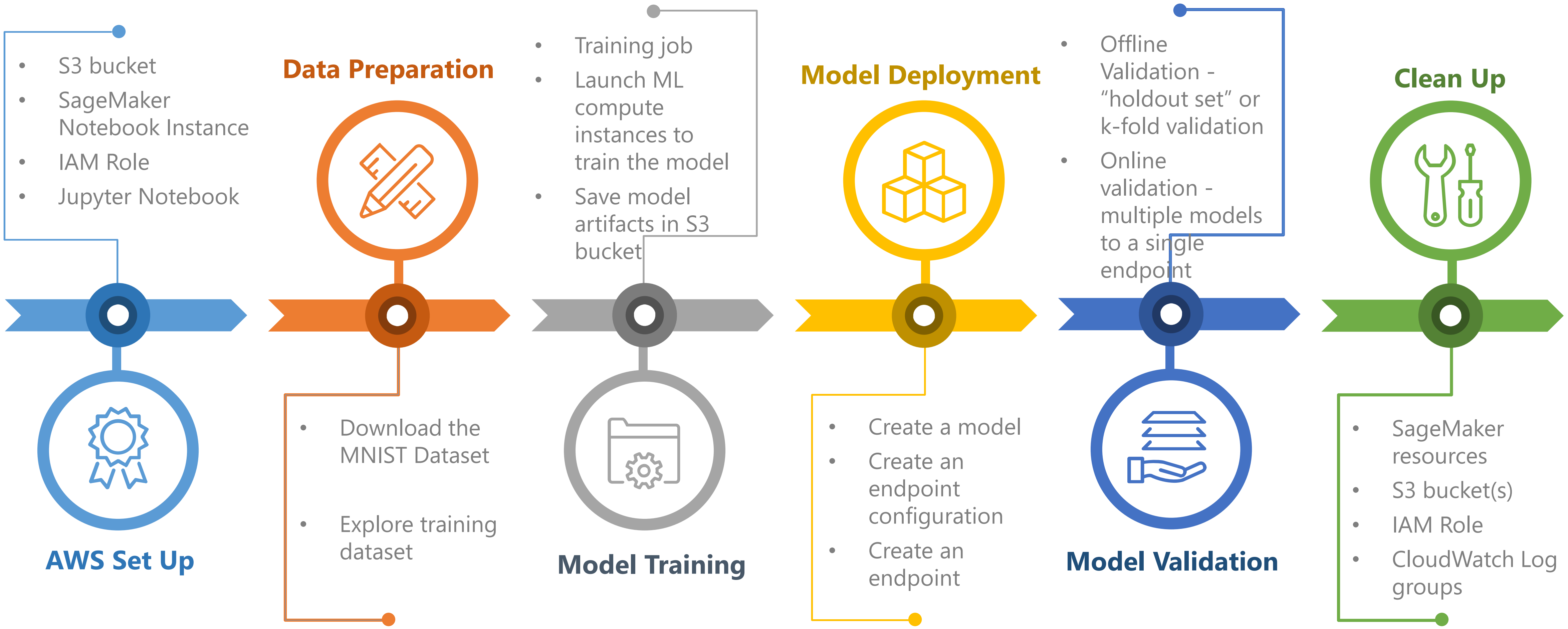
- Offline Validation - “holdout set” or k-fold validation
- Online validation - multiple models to a single endpoint

Clean Up

- SageMaker resources
- S3 bucket(s)
- IAM Role
- CloudWatch Log groups

Step 2

Steps in Machine Learning using Amazon SageMaker



AWS Set Up

- S3 bucket
- SageMaker Notebook Instance
- IAM Role
- Jupyter Notebook

Data Preparation

- Download the MNIST Dataset
- Explore training dataset

Model Training

- Training job
- Launch ML compute instances to train the model
- Save model artifacts in S3 bucket

Model Deployment

- Create a model
- Create an endpoint configuration
- Create an endpoint

Model Validation

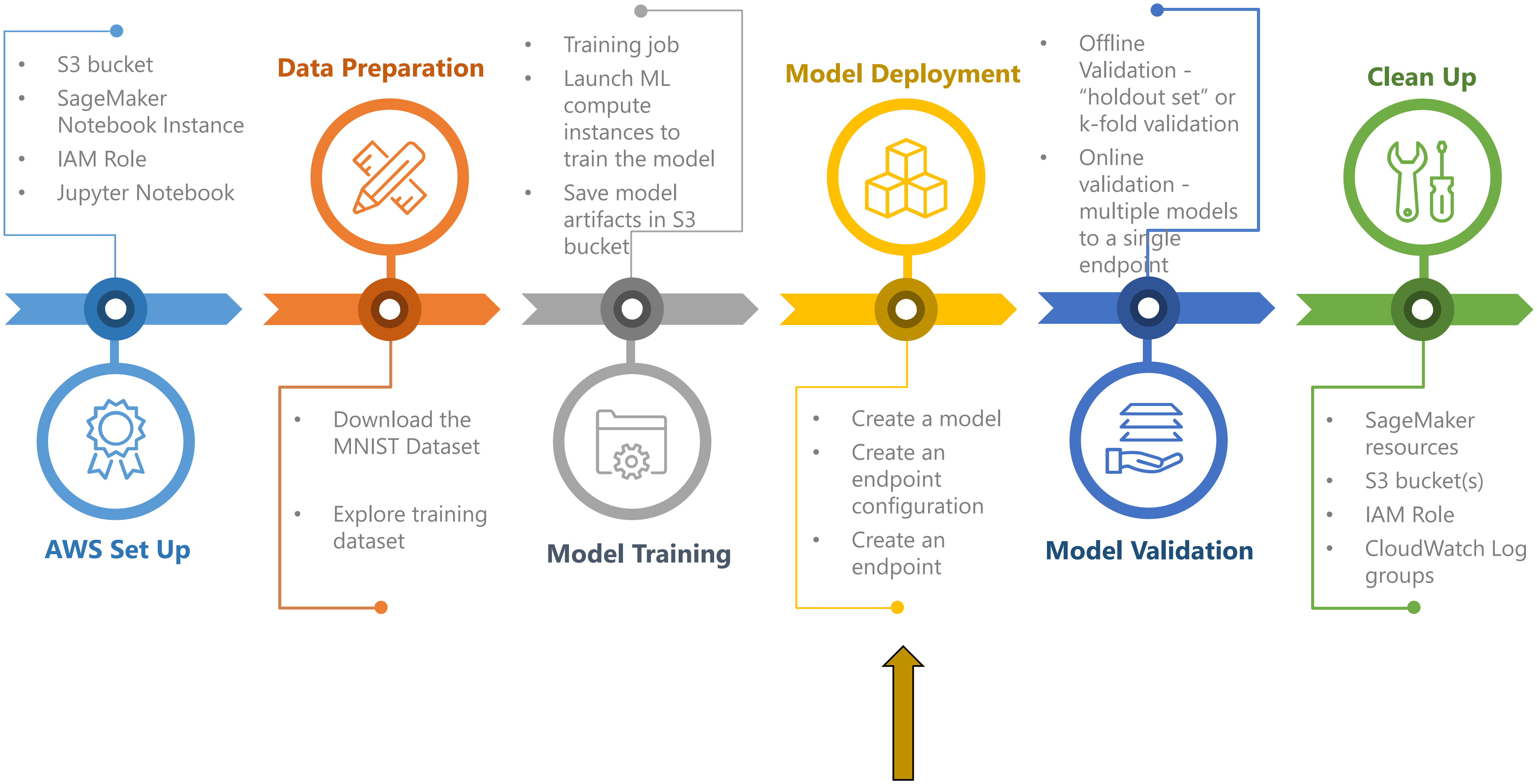
- Offline Validation - “holdout set” or k-fold validation
- Online validation - multiple models to a single endpoint

Clean Up

- SageMaker resources
- S3 bucket(s)
- IAM Role
- CloudWatch Log groups

Step 3

Steps in Machine Learning using Amazon SageMaker



AWS Set Up

- S3 bucket
- SageMaker Notebook Instance
- IAM Role
- Jupyter Notebook

Data Preparation

- Download the MNIST Dataset
- Explore training dataset

Model Training

- Training job
- Launch ML compute instances to train the model
- Save model artifacts in S3 bucket

Model Deployment

- Create a model
- Create an endpoint configuration
- Create an endpoint

Model Validation

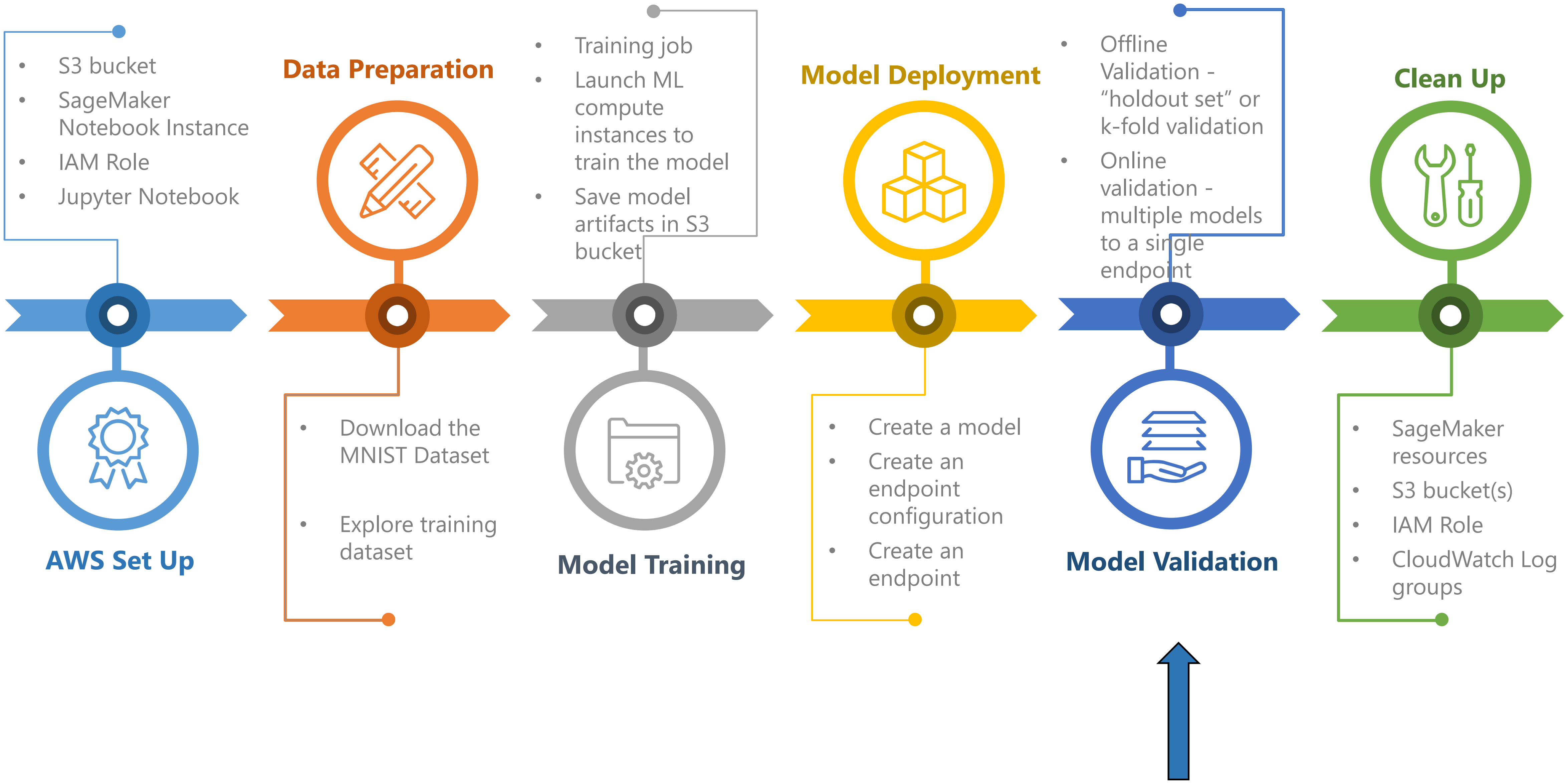
- Offline Validation - "holdout set" or k-fold validation
- Online validation - multiple models to a single endpoint

Clean Up

- SageMaker resources
- S3 bucket(s)
- IAM Role
- CloudWatch Log groups

Step 4

Steps in Machine Learning using Amazon SageMaker



AWS Set Up

- S3 bucket
- SageMaker Notebook Instance
- IAM Role
- Jupyter Notebook

Data Preparation

- Download the MNIST Dataset
- Explore training dataset

Model Training

- Training job
- Launch ML compute instances to train the model
- Save model artifacts in S3 bucket

Model Deployment

- Create a model
- Create an endpoint configuration
- Create an endpoint

Model Validation

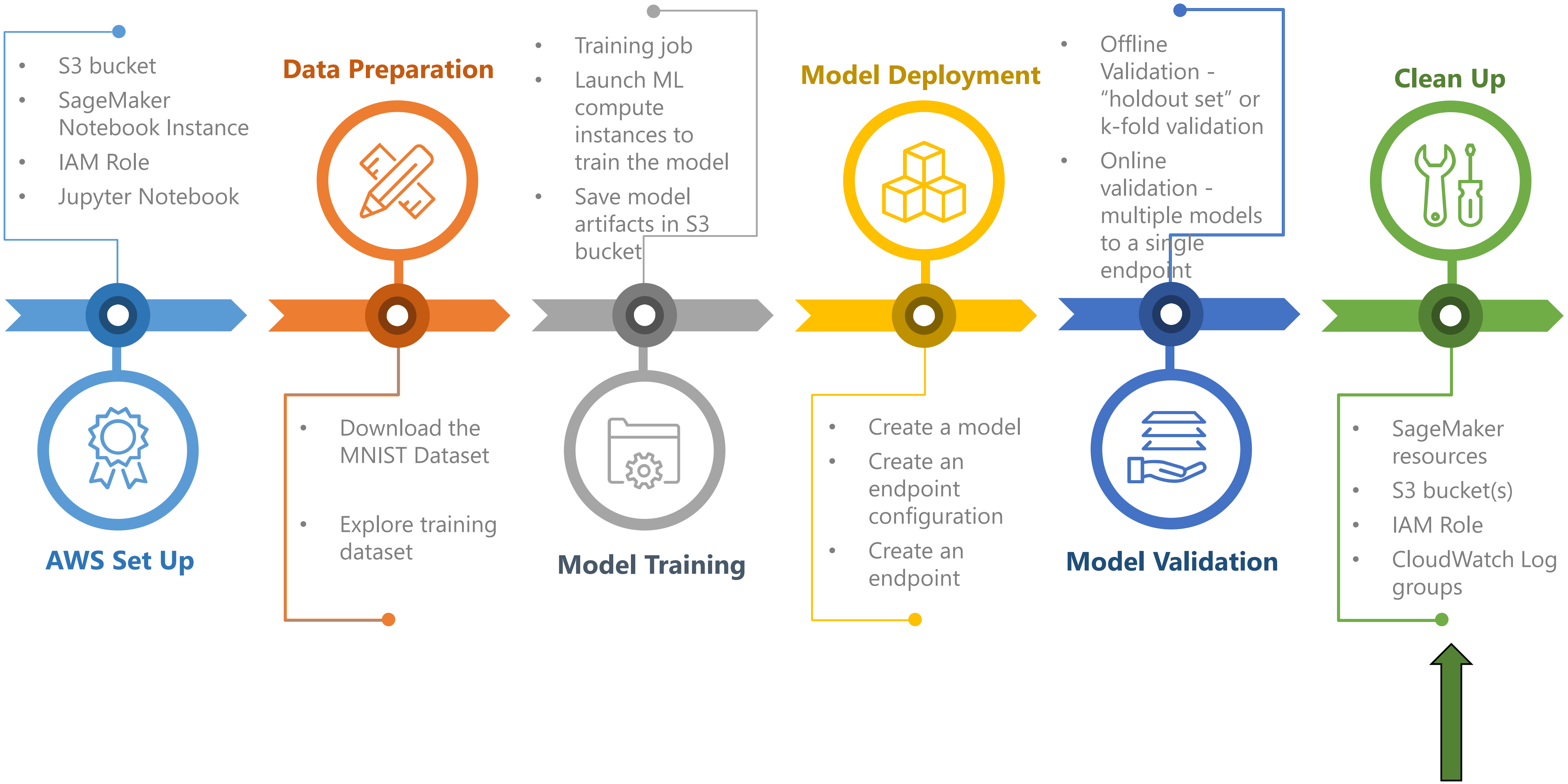
- Offline Validation - “holdout set” or k-fold validation
- Online validation - multiple models to a single endpoint

Clean Up

- SageMaker resources
- S3 bucket(s)
- IAM Role
- CloudWatch Log groups

Step 5

Steps in Machine Learning using Amazon SageMaker



AWS Set Up

- S3 bucket
- SageMaker Notebook Instance
- IAM Role
- Jupyter Notebook

Data Preparation

- Download the MNIST Dataset
- Explore training dataset

Model Training

- Training job
- Launch ML compute instances to train the model
- Save model artifacts in S3 bucket

Model Deployment

- Create a model
- Create an endpoint configuration
- Create an endpoint

Model Validation

- Offline Validation - “holdout set” or k-fold validation
- Online validation - multiple models to a single endpoint

Clean Up

- SageMaker resources
- S3 bucket(s)
- IAM Role
- CloudWatch Log groups

Step 6

Step 6 – Clean up...

To avoid incurring unnecessary charges, use the AWS Console to delete following resources

- Amazon SageMaker console:
  - Endpoint (auto deletes ML compute instances)
  - Endpoint configuration
  - Model
  - Notebook instance (stop the instance before deleting it)

- Amazon S3 console:
  - S3 buckets created for storing model artifacts and training dataset

- IAM console:
  - Delete IAM Role
  - Delete customized IAM Policy, if created

- Amazon CloudWatch console:
  - Log groups starting with “/aws/sagemaker/”

Amazon SageMaker Programming Model

- Use the Amazon SageMaker console
    - Works well for simple jobs

- Modify the example Jupyter notebooks
    - Start with a notebook that has a suitable algorithm
    - Modify it to accommodate your data source and specific needs

- Write model training and inference code from scratch:
    - High-level Python library:
        - Simplifies model training and deployment
        - Handles various operations implicitly
    - AWS SDK:
        - Available in multiple languages and platforms
        - Provides methods that corresponds to SageMaker APIs
        - No need to write authentication code. Uses access keys to implicitly authenticate the requests.

- Integrate Amazon SageMaker into your Apache Spark workflow
    - SageMaker provides a library for calling its APIs from Apache Spark
    - With it, Amazon SageMaker-based estimators can be used in an Apache Spark pipeline

K-Means Algorithm

- Unsupervised classification machine learning algorithm
- Finds discrete groupings within data by training a model that groups similar objects together
- Maps each observation to a point in the n-dimensional space
- Choose the number of clusters (k) to create
- Determines the initial K cluster centers: random or k-means++ approach
- Iterates over the training dataset and calculates cluster centers: moves these centers toward the true cluster centers
- K-Means Hyperparameters
    - **K**: number of required clusters
    - **feature_dim**: number of features in the input data
    - **mini_batch_size**: number of observations per mini-batch
    - **init_method**: random or kmeans++
    - **extra_center_factor**: K centers = num of clusters (k) * extra_center_factor (x)
    - **objective metric**: msd (Mean squared distances)/ssd (Sum of the squared distances)

Automatic Model Tuning/Hyperparameter Tuning

- Supervised machine learning regression problem
- Finds the best version of a model
- Specify the range of hyperparameters values to search to find the best values
- Launches training jobs on your dataset using the algorithm and ranges of hyperparameters
- Chooses the hyperparameter values that result in a model that performs the best, as measured by an objective metric that you choose
- Deploy the best trained model that training job created

Machine Learning using other AWS Services

Demo of Machine Learning using other AWS Services

- **Amazon Comprehend**: Natural Language Processing and Text Analytics
- **Amazon Rekognition**: Deep learning-based visual analysis service for images and videos
- **Amazon Translate**: Natural and fluent translation -automatic detection and translation of a language text
- **Amazon Polly**: Text-to-Speech
- **Amazon Lex**: Build voice and text natural language chatbots
- **AWS DeepLens**: Deep learning-enabled video camera
- **Amazon Transcribe**: Automatic Speech Recognition

Questions ??

Thanks for coming

Have a nice day!