



# Cloud Developer Nanodegree Program

*Become an AWS cloud developer*

---

## COURSE 1: Cloud Fundamentals

The cloud has become a key enabler for innovation with beneficial features like high availability, unlimited capacity, and on-demand scalability and elasticity. Learn the fundamentals of cloud computing while being introduced to compute power, security, storage, networking, messaging, and management services in the cloud. While learning the fundamentals, you will explore tools and services offered by Amazon Web Services (AWS) through interactive hands-on exercises. By the end of the course, you will have deployed your first website to AWS.

## PROJECT 1: Deploy Static Website on AWS

The cloud is perfect for hosting static websites that only include HTML, CSS, and JavaScript files that require no server-side processing. In this project, you will deploy a static website to AWS. First, you will create a S3 bucket, configure the bucket for website hosting, and secure it using IAM policies. Next, you will upload the website files to your bucket and speed up content delivery using AWS's content distribution network service, CloudFront. Lastly, you will access your website in a browser using the unique S3 endpoint.

## SUPPORTING LESSON CONTENT

Lesson Title	Learning Outcomes
<b>Cloud Overview</b>	<ul style="list-style-type: none"><li>→ Learn the basics of cloud computing including cloud deployment models, benefits, and popular options</li><li>→ Explore services provided by Amazon Web Services(AWS)</li></ul>
<b>Foundational and Compute Services</b>	<ul style="list-style-type: none"><li>→ Learn why we need servers, compute power, and security</li><li>→ Explore AWS compute services like Elastic Cloud Compute (EC2), Virtual Private Cloud (VPC), Lambda for serverless framework, and Elastic Beanstalk in action</li><li>→ Launch a secure EC2 instance, create and execute a Lambda, and deploy an application to Elastic Beanstalk</li></ul>
<b>Storage and content delivery</b>	<ul style="list-style-type: none"><li>→ Learn why we need storage and content delivery in the cloud</li><li>→ Learn storage services like S3, DynamoDB, Relational Database Service (RDS), and CloudFront</li><li>→ Create a DynamoDB table, launch a MySQL database instance,</li></ul>

---

and create a CloudFront distribution

---

### Security

- Learn the importance of security in the cloud
  - See Identity & Access Management (IAM) in action
  - Secure applications using IAM users, groups, and policies
- 

### Networking & Elasticity

- Learn the basics of networking and elasticity in the cloud
  - Examine services like Route 53, EC2 Auto Scaling, and Elastic Load Balancing
  - Add an auto scaling policy to your EC2 instance
- 

### Messaging & Containers

- Learn the basics of messaging and containers in the cloud
  - Explore services like Simple Notification Service (SNS), Simple Queue Service (SQS), and Elastic Container Service (ECS)
  - Create cloud notifications using SNS
- 

### AWS Management

- Learn why we need logging, auditing, and resource management in the cloud
  - Understand services like Cloud Watch, Cloud Trail, Cloud Formation, and the AWS Command Line Interface (CLI)
  - Explore the CLI
- 

## COURSE 2: Full Stack Apps on AWS

Explore the foundational concepts of designing and deploying scalable, extendable, and maintainable full stack applications using modern cloud architecture. All concepts are covered at a fundamental level and motivated with practical, real world programming exercises. Through the course, you'll have built and deployed a multi-service cloud stack. By the end of this course, you'll understand key design decisions and useful tools to maintain your application.

## PROJECT 2: Udagram: your own Instagram on AWS

In this project, you will develop a cloud-based application for uploading, listing, and filtering images. You will use Node.js/Express, a popular javascript framework for networked application development to develop this application. You will implement a REST API to issue commands using HTTP, store data in Amazon Web Services Relational Data Service (RDS) and S3, extend the codebase with secure authentication signon features, and deploy to Amazon Web Services Elastic Beanstalk. These are the hard skills you'll need in any Cloud developer role.

## SUPPORTING LESSON CONTENT

Lesson Title	Learning Outcomes
<b>CLOUD BASICS</b>	<ul style="list-style-type: none"><li>→ Learn key terminology and building blocks of a cloud system</li><li>→ Understand design paradigm of modern cloud applications</li></ul>
<b>SET UP BEST PRACTICES</b>	<ul style="list-style-type: none"><li>→ Implement a process so you write quality code, working alone or on teams.</li></ul>

---

	→ Learn unit and integration testing, a better way to git, and how to use packaged dependencies.
<b>STORING DATA IN THE CLOUD</b>	<ul style="list-style-type: none"> <li>→ Set up and start using a cloud-based relational database for storing user data using AWS RDS</li> <li>→ Implement a filestore for media like images using AWS S3</li> </ul>
<b>DEPLOYING TO THE CLOUD</b>	<ul style="list-style-type: none"> <li>→ Consume cloud data services (database and filestore) within your server application</li> <li>→ Deploy your application using AWS Elastic Beanstalk.</li> </ul>
<b>USER AUTHENTICATION AND SECURITY</b>	→ Learn common mistakes and modern techniques for dealing with security and new set of cloud authentication challenges
<b>SCALING AND FIXING</b>	→ Cloud systems need to be maintained as dependencies are updated and there is more demand for your service. Explore tools and process to minimize growing pains

## COURSE 3: Monolith to Microservices at Scale

Microservices are becoming the default mode of developing and deploying applications at scale. The microservices architecture makes it easier to scale an application to a large system and is a great enabler for continuous integration and delivery. Microservices architecture allows independent scaling, independent releases and deployments and independent development so that each service has its own codebase. In this course we will cover the best practices on how to develop and deploy microservices. You will learn topics such as different microservice architecture patterns, independent scaling, resiliency, service replication, service registration and discovery. By the end of this course, you should be able to design and build an application using a microservice architecture.

## PROJECT 3: Refactor Udagram app into Microservices and deploy using Kubernetes

In this project, you will reuse their existing Udagram application and convert and extend into a microservice architecture. After the application is divided into smaller service, you will containerize it and deploy it to a Kubernetes cluster. This includes the deployment pipeline, scalability, observability, services, networking, and deployment strategies to service the system. You will then implement and interpret performance, usage, and logs to solve real problems similar to those they would encounter in the field.

## SUPPORTING LESSON CONTENT

Lesson Title	Learning Outcomes
<b>Microservices design principles and best practices</b>	→ Learn different microservices architecture designs how to divide an application into microservices

<b>Independent releases and deployments</b>	<ul style="list-style-type: none"> <li>→ Understand CI/CD benefits and use Travis to build CI/CD pipeline</li> <li>→ Integrate github and CI/CD and automate testing with CI</li> </ul>
<b>Containers using Docker</b>	<ul style="list-style-type: none"> <li>→ Build and run your first container image using Docker</li> <li>→ Debug container and store these images using container registry</li> </ul>
<b>Service orchestration with Kubernetes</b>	<ul style="list-style-type: none"> <li>→ Learn the fundamentals of Kubernetes</li> <li>→ Configure and launch a auto-scaling, self-healing Kubernetes cluster</li> <li>→ Deploy your microservices using Kubernetes cluster</li> </ul>
<b>Service registration, discovery and scaling</b>	<ul style="list-style-type: none"> <li>→ Implement service discovery and</li> </ul>
<b>Debugging, Monitoring and Logging</b>	<ul style="list-style-type: none"> <li>→ Learn best practices for debugging microservices</li> <li>→ Implement monitoring and logging for microservices</li> <li>→ Understand how to build resilience in your application</li> </ul>

## COURSE 4: Develop & Deploy Serverless App

Serverless technologies have become very popular recently because they can increase the speed of development and drastically reduce the cost of running a cloud infrastructure. This course combines the theory of using serverless technologies with the practice of developing a complex serverless application. You will learn advanced serverless features such as implementing WebSockets and stream processing, and learn about serverless best practices throughout the course.

### PROJECT 4: Serverless Application

In this project you will develop an Instagram-like serverless service for uploading, listing, and filtering images. You will begin with building serverless REST APIs using API Gateway and AWS Lambda, a stack of serverless technologies on AWS. You will then implement an API to interact with this application, store data in AWS DynamoDB, S3, and Elasticsearch, secure your application with authentication, and deploy to Amazon Web Services using a Serverless framework.

### SUPPORTING LESSON CONTENT

Lesson Title	Learning Outcomes
<b>Introduction to Serverless</b>	<ul style="list-style-type: none"> <li>→ Learn the main components of a serverless application</li> <li>→ Implement simple application using Function as a Service (FaaS)</li> </ul>
<b>REST APIs for Serverless</b>	<ul style="list-style-type: none"> <li>→ Build a simple REST API using serverless technologies such as API Gateway, AWS Lambda, and AWS DynamoDB and use it in React based web application</li> </ul>

---

<b>Serverless framework</b>	<ul style="list-style-type: none"><li>→ Build, package and deploy serverless applications using Serverless framework</li><li>→ Implement additional features using advanced DynamoDB features</li></ul>
<b>Events processing with Serverless</b>	<ul style="list-style-type: none"><li>→ Improve our application using FaaS ability to execute custom logic when particular events occur in the system</li><li>→ Add WebSockets support and full-text search to your app</li></ul>
<b>Implementing Authentication</b>	<ul style="list-style-type: none"><li>→ Implement authentication in a serverless application using popular serverless service Auth0 in your app API and client application.</li><li>→ Learn to store secrets for our serverless application using AWS Secrets Manager.</li></ul>
<b>Serverless best practices</b>	<ul style="list-style-type: none"><li>→ Learn how to test serverless applications, minimize risk of vendor lock-in, create multiple stages of our API and improve security and observability of our application</li></ul>
<b>Serverless GraphQL</b>	<ul style="list-style-type: none"><li>→ Implement Udagram application using another serverless component called AWS AppSync that allows to easily create GraphQL API for cloud resources without writing much backend code.</li></ul>

---

---

## Capstone Project

The purpose of the cloud development capstone project is to give you a chance to combine what you've learned throughout the program. This project will be an important part of your portfolio that will help you achieve your cloud development-related career goals.

In the capstone project, each project is unique to the student. You'll build an application on AWS based on predefined criteria. Students will define the scope of the project, come up feature list and decide which AWS services to use to meet availability and performance criteria.