Every AWS Service You'll EVER Need

Quick Start Links

- **AWS Free Tier Signup**
- **AWS Console**
- **AWS Skill Builder (Main)** - filter by service name and set the access tier to **Free**.
- **AWS re:Post**
- **AWS Docs (Landing)**
- **aws-samples GitHub org**

The 7 Buckets Overview

1. Foundation & Security
2. Compute & Containers
3. Storage & Databases
4. Networking & Content Delivery
5. DevOps & Automation
6. AI & Machine Learning
7. Monitoring & Cost

BUCKET 1: Foundation & Security

Learning Links

- IAM basics:
- MFA setup:
- Billing alarms (CloudWatch billing):
- Secrets Manager:
- Parameter Store:
- IAM roles for services (EC2 → S3 pattern):

Day One Setup Checklist

- ☐ Create AWS account

    - Setup guide:
- ☐ Enable MFA on root
    - Tutorial:
- ☐ Billing alert + alarm ($10/$20/$50)
    - Tutorial:
- ☐ Create IAM user (Admin for learning)
    - Tutorial:

Hands-on Tasks

- ☐ Create an IAM role and attach policy
    - Example policy reference:
- ☐ Store a secret in Secrets Manager and retrieve it
    - Code sample reference:
- ☐ Store config in Parameter Store and retrieve it
    - Code sample reference:

Resources:

- Account Setup Learning Plan
- AWS IAM Builder Labs (includes troubleshooting labs)

BUCKET 2: Compute & Containers

Learning Links

- EC2 fundamentals:
- SSH + Linux basics:
- Lambda fundamentals:
- Docker basics:
- ECS basics:
- Fargate basics:
- EKS intro (optional):
- ECR basics:
- GPU instances (P/G/Inf):

Core Services Notes

EC2

- EC2 guide:

- User data scripts:
- GPU instance overview:

Lambda

- Lambda guide:
- Lambda triggers examples:

ECS + Fargate

- ECS getting started:
- Fargate overview:
- ECR push/pull tutorial:

EKS (optional)

- EKS getting started:
- Kubernetes basics:

Hands-on Tasks

- ☐ Launch EC2 and SSH in
    - Walkthrough:
- ☐ Deploy a Lambda with a trigger
    - Walkthrough:
- ☐ Build Docker image + push to ECR
    - Walkthrough:
- ☐ Run container on ECS/Fargate
    - Walkthrough:

Resources:

- Compute Services Courses & Labs

BUCKET 3: Storage & Databases

Learning Links

- S3 basics + policies:
- S3 events (S3 → Lambda):
- EBS volumes + snapshots:

- EFS basics:
- RDS basics:
- DynamoDB basics (keys, indexes):
- ElastiCache overview:
- Aurora overview:

Hands-on Tasks

- ☐ Create S3 bucket + upload/download using SDK
  - Boto3 example:
- ☐ Attach EBS volume to EC2 + snapshot it
  - Walkthrough:
- ☐ Mount EFS on 2 EC2 instances
  - Walkthrough:
- ☐ Create RDS instance + connect from app
  - Walkthrough:
- ☐ Create DynamoDB table + basic CRUD
  - Walkthrough:

BUCKET 4: Networking & Content Delivery

Learning Links

- VPC basics (subnets, routes):
- Security groups vs NACLs (basic):
- ALB vs NLB:
- Route 53 basics:
- CloudFront basics:
- API Gateway basics:

Hands-on Tasks

- ☐ Create VPC with public + private subnets
  - Walkthrough:
- ☐ Add an ALB in public subnet
  - Walkthrough:
- ☐ Point domain to ALB using Route 53
  - Walkthrough:
- ☐ Put CloudFront in front of S3/ALB

    - Walkthrough:
- ☐ Create API Gateway endpoints for Lambda
    - Walkthrough:

Resources:

- Configuring VPC
- Content Delivery Setup

BUCKET 5: DevOps & Automation

Learning Links

- CloudFormation basics:
- CDK getting started (Python):
- CDK getting started (TypeScript):
- Systems Manager (Session Manager):
- CodePipeline basics:
- CodeBuild buildspec.yml:
- CodeDeploy overview:
- EventBridge basics:

Hands-on Tasks

- ☐ Deploy infra with CloudFormation OR CDK
    - Template/examples:
- ☐ Use SSM Session Manager to access EC2
    - Walkthrough:
- ☐ Build CI/CD pipeline: GitHub → build → ECR → ECS deploy
    - Walkthrough:
- ☐ EventBridge rule triggering Lambda
    - Walkthrough:

BUCKET 6: AI & Machine Learning

Learning Links

- Bedrock overview + models:

- Bedrock API examples:
- SageMaker overview:
- Deploying endpoints in SageMaker:
- Amazon Q Developer:
- Textract docs:
- Rekognition docs:
- Comprehend docs:

Hands-on Tasks

- ☐ Call Bedrock from Lambda
  - Example repo:
- ☐ Textract pipeline: S3 upload → Lambda → Textract → DB
  - Example repo:
- ☐ Comprehend sentiment/entity extraction
  - Example:

BUCKET 7: Monitoring & Cost

Learning Links

- CloudWatch metrics/logs/alarms:
- CloudWatch dashboards:
- Cost Explorer:
- Budgets:
- CloudTrail:

Hands-on Tasks

- ☐ Add CloudWatch alarm for CPU or error rate
  - Walkthrough:
- ☐ Enable CloudTrail and confirm logs in S3
  - Walkthrough:
- ☐ Set monthly budget and alerts
  - Walkthrough:

The 7 Portfolio Projects

Project 1: Three-Tier Web Application

GitHub: https://github.com/aws-samples/aws-three-tier-web-architecture-workshop

Services Used: VPC, ALB, EC2, RDS
Goal: Build classic 3-tier infra
Coding: Minimal app code

Project 2: Serverless REST API

GitHub: https://github.com/aws-samples/serverless-samples

Services Used: API Gateway, Lambda, DynamoDB
Goal: Serverless backend
Coding: Python/Node.js

Project 3: CI/CD Pipeline

Youtube: **Build a CI/CD Pipeline with AWS CodePipeline | Training and Demo**
GitHub: https://github.com/krishnaik06/AWS-CI-CD-Projects

Services Used: CodePipeline, CodeBuild, ECR, ECS
Goal: Automated deployments
Coding/config: Dockerfile + buildspec.yml

Project 4: Event-Driven File Processing

Resource:
https://docs.aws.amazon.com/lambda/latest/dg/concepts-event-driven-architectures.html
Github:

- https://github.com/aws-samples/event-driven-architecture-with-amazon-sns-fifo
- https://github.com/aws-samples/appmod-partners-serverless/blob/main/event-driven-architecture/README.md

Services Used: S3, Lambda, EventBridge, SNS, Step Functions
Goal: Event-driven architecture
Coding: Lambda + state machine JSON

Project 5: AI Document Analysis

GitHub:

- https://github.com/aws-samples/aws-ai-intelligent-document-processing
- https://github.com/aws-samples/aws-generative-ai-document-processing-solution

Services Used: S3, Lambda, Textract, Comprehend, DynamoDB
Goal: AI workflow pipeline
Coding: Python + basic UI

Project 6: Infrastructure as Code (CDK)

CDK Examples: **Getting Started With Infrastructure as Code (AWS CDK, CloudForma…**
CDK Ref: https://github.com/aws/aws-cdk

Services Used: VPC, ECS, RDS, ALB
Goal: Infra fully in code
Coding: Python/TypeScript

Project 7: AI Chatbot with Bedrock

Youtube: **Build a RAG based Generative AI Chatbot in 20 mins using Amazon Bedroc…**
GitHub: https://github.com/aws-samples/bedrock-chat

Services Used: API Gateway, Lambda, Bedrock, DynamoDB, S3, CloudFront
Goal: End-to-end GenAI app
Coding: Backend + frontend

Learning Approach (Milestones + Link Slots)

Phase 1: Foundation (Buckets 1 + 3)

Recommended path link:
Milestone: S3 static site + IAM + SDK scripting

Phase 2: Infrastructure (Buckets 2 + 4)

Milestone: Three-tier app (Project 1)

Phase 3: Automation (Buckets 5 + 7)

Milestone: CI/CD pipeline (Project 3)

Phase 4: AI (Bucket 6)

Milestone: Project 5 or 7

Cost Management (Link Slots)

Free Tier reference:

https://aws.amazon.com/free/

Billing alarms walkthrough:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/monitor_estimated_charges_with_cloudwatch.html

Cleanup checklist / teardown guide:

https://docs.aws.amazon.com/general/latest/gr/aws-general-reference.html

(Use with service-specific delete guides — standard AWS teardown reference)

Monthly budget sheet (AWS Budgets):

https://docs.aws.amazon.com/cost-management/latest/userguide/budgets-managing-costs.html

---

Communities

- **r/aws**
- **r/devops**
- **AWS re:Post**

Quick Tips:

For beginners:

- Learn on **Skill Builder (Free tier)**
- Build using **aws-samples GitHub**
- Ask questions on **re:Post + r/aws**
- Follow **AWS Events** for architecture clarity