**Beginner-Level Projects**

1. **Static Website Hosting on S3**
   * Host a static HTML/CSS/JS website on an S3 bucket.
   * Configure static website hosting, set up permissions, and enable versioning.
2. **EC2-Based Web Application**
   * Launch an EC2 instance and deploy a basic web application (e.g., a Python Flask or Node.js app).
   * Configure security groups and key pairs for SSH access.
3. **IAM User and Policy Management**
   * Create IAM users, groups, and roles with custom policies.
   * Implement least-privilege access for specific AWS resources.
4. **RDS with a Simple Database**
   * Set up an RDS (MySQL/PostgreSQL) instance.
   * Connect to it using a local or EC2-hosted application to perform CRUD operations.
5. **CloudWatch Monitoring and Alerts**
   * Monitor an EC2 instance or S3 bucket with CloudWatch metrics.
   * Create alarms for high CPU usage or storage thresholds.

**Intermediate-Level Projects**

1. **Serverless REST API with Lambda and API Gateway**
   * Build a REST API using AWS Lambda and API Gateway.
   * Connect it to DynamoDB for a fully serverless architecture.
2. **Autoscaling EC2 with Load Balancer**
   * Set up an Auto Scaling group with a Load Balancer for a web application.
   * Simulate traffic to test the scaling functionality.
3. **VPC Configuration and Networking**
   * Design a VPC with public and private subnets.
   * Deploy resources in each subnet and configure a NAT Gateway.
4. **AWS CLI Automation**
   * Automate common AWS tasks like launching EC2 instances, managing S3 buckets, and setting up IAM roles using the AWS CLI.
5. **AWS Systems Manager for EC2 Automation**
   * Use AWS Systems Manager to automate patching and configuration management on EC2 instances.
   * Enable session manager for remote shell access.

**Advanced-Level Projects**

1. **Containerized Application Deployment with ECS and Fargate**
   * Deploy a Dockerized application using AWS ECS and Fargate.
   * Set up service discovery and auto-scaling for tasks.
2. **CloudFormation for Infrastructure as Code (IaC)**
   * Create a CloudFormation template to provision a full-stack environment, including VPC, EC2, RDS, and S3.
   * Automate deployment using the AWS Management Console or CLI.
3. **Kubernetes Cluster with EKS**
   * Set up an EKS cluster and deploy a containerized application.
   * Configure load balancing and auto-scaling with Kubernetes.
4. **Real-Time Data Processing with Kinesis**
   * Set up Kinesis Data Streams to process real-time data.
   * Use Lambda for processing and store results in DynamoDB.
5. **AWS Glue and Athena for Data Analytics**
   * Use AWS Glue to catalog data stored in S3.
   * Query the data using Athena and visualize it with QuickSight.
6. **CI/CD Pipeline with CodePipeline and CodeBuild**
   * Set up a CI/CD pipeline using CodePipeline, CodeBuild, and CodeDeploy.
   * Deploy a sample application to an EC2 instance or Lambda function.
7. **Disaster Recovery Setup**
   * Design a disaster recovery plan using S3 cross-region replication and RDS multi-AZ deployments

**18. Hybrid Cloud Architecture with AWS VPN**

* Set up a hybrid cloud environment by connecting an on-premises network to AWS using AWS Site-to-Site VPN.
* Deploy resources in a private subnet and test connectivity from the on-premises environment.

**19. AWS Step Functions for Orchestration**

* Create a serverless workflow using AWS Step Functions.
* Integrate multiple AWS services like Lambda, DynamoDB, and S3 to automate a multi-step process (e.g., image processing or data pipeline).

**20. Machine Learning Deployment with SageMaker**

* Build, train, and deploy a machine learning model using SageMaker.
* Use the model for real-time predictions through an API endpoint and integrate it with a front-end application.

Let me know if you'd like detailed guidance on any of these projects!