makeithappen.nyc

web based sites & applications

Managing Your AWS Infrastructure

You have two choices when managing cloud infrastructure: build a serverless architecture or run on traditional EC2 instances. Developers love building services and deploying them to the public cloud. It reduces the overhead of managing infrastructures, such as networking and operating systems.

Running on traditional EC2 instances, however, brings some limitations. For example, there's limited control over memory and storage. If you want to use more memory or storage, your EC2 instances have to be scaled up, resulting in increased costs. Fortunately, AWS offers another option: Lambda. Developers can now build serverless architectures by integrating AWS Lambda with AWS Elastic Beanstalk and IAM. With serverless apps, they can build apps without worrying about server infrastructure.

Your choice of environment will depend on which type you are most comfortable with and your project's requirements.

1. AWS Cloud - The first option is to deploy live in Amazon's cloud servers for maximum performance. If this was not selected during development, it is done before deployment by selecting Elastic Beanstalk and choosing "Deploy" from the drop-down menu. This can have drawbacks as any changes made require a redeployment process.
2. AWS Cloud using EC2 - Using EC2 provides more flexibility than deploying directly into Amazon's cloud services but less than storing your code safely offline in S3 storage buckets. This way, you can use AWS's resources without having any hardware or software requirements upfront. It's also easy and convenient because it abstracts away a lot of the complexity by allowing users to pay only for what they need as they go along instead of committing upfront with significant costs. In this case, performance is not an issue since there are no scalability limitations.
3. Private Environment – One advantage of managing everything yourself within a private server is greater control over how things work, including faster updates without waiting on someone else's timelines or processes. The downside of a private environment is you have to find different solutions to many problems yourself. It also can be more work when there's no team available if something goes wrong.

The serverless computing model allows developers to build applications without worrying about the servers on which those applications will run. Instead of worrying about configuring, maintaining, and scaling servers, developers can now stay focused on building their applications. With AWS Lambda and AWS Elastic Beanstalk, developers are empowered to build serverless architectures, enabling developers to focus on business value and not infrastructure. With AWS Lambda, developers can define their business logic in code and deploy it to AWS Lambda, and developers can elastically scale their functions.

In AWS Lambda, developers create functions and code units executed on a server in response to an event. An event can be anything from a user navigating a web page to selecting a button to make a payment. Developers define their functions using the AWS Lambda console, and after submitting code, AWS Lambda automatically scales, configures, and manages the servers on which their functions execute. Lambda functions have a simple structure:

1. A name: The function's name is usually the name of the event triggering the function.
2. An event source: The event source is an AWS service that sends the events to the Lambda function. You can specify a function but not an AWS service.
3. A function body: The function body contains the code executed when the event is received.
4. An event type: The event type. The event type is the event source's name
5. A role: a role is defined as a collection of permissions to resources. It defines a collection of operations performed in AWS. The AWSLambdaRole resource represents this collection.

Why work with us

We offer comprehensive on-demand design and development services—ideal for companies that do not have a design and development team in-house. Working with us means you have a partner to help manage your design and development needs so that you can focus on running your core business. By partnering with us, you can have our dedicated team help and guide you with:

- Data Visualization
- Custom web development
- Website creation and maintenance
- Database development
- User Experience and User Interface design

Company Experience:

- Ten years of experience developing real-time, performance-critical applications
- Five years of Angular-js and React-js development experience
- Proficient in cloud platforms such as Amazon Web Services
- Proficient in AGILE methods such as TDD, continuous integration, scrum, user stories, iterative development, and domain-driven design.
- Skilled at delivering modular, decoupled, user-friendly applications optimized for performance.
- Expert in Design Patterns and real-time messaging systems.
- Excellent written and verbal communication skills.

Technical Proficiency:

ReactJS
Redux
MVC Architecture
AngularJS
Node.js
Axios
Javascript
Stripe integration

HTML5
Responsive UI
GIT
SQL Server
Postgres SQL
Bootstrap
C#.NET
REST Web Services

Amazon Web Services
EC2
RDS
Lambda
SNS
Elastic Beanstalk
s3
EFS