Hello, I am Alex, a programmer.

Thank you so much for visiting my tech blog, I am so excited to introduce this blog to you! I developed it with Node.js, and MongoDB, you can find the source code of this blog on my GitHub.

In this little space, I am going to present and share my knowledge, skills, and personal understanding of programming with all of you. Currently, there are several blogs on topics of HTML, CSS/SASS, Javascript, Typescript, Node.js, EJS, React, MongoDB, MS SQL Server, AWS Cloud Service, C#, .Net, HTTP, API, Unity3D, and Data Structure and Algorithms, and I will keep updating new blogs in the coming days. I also look forward to your comments and communication with me. Your support is always what keeps me going. If you have any questions, please do not hesitate to leave comments on the message board of this blog or email me, and I will reply to you as soon as possible. Thank you!

Alex

AWS Multi-factor Authentication(MFA) and Identity Access Management(IAM)

Today I just start my new journey on AWS, and my aim is to get a AWS Architect Associate Certificate and good AWS skills after one months’ learning and practice. Today I have got some introduction knowledge of AWS, and I created multiple AWS accounts with the instruction of Adrian. As the first days lessons are all about AWS account configuration, and security of accounts is really important, the tutor introduce two ways that enhance security of AWS account, which impressed me a lot:

The first thing is Multi-factor Authentication(MFA).

In our daily life, it is very common to use username and password to get authenticated by many authentication systems. However, it is really risky because once someone else gets your username and password, it’s easily to make breach in your account. If more different pieces of evidence could be provided and required to prove identity in the authentication process, it can be more secure. Those evidence is called factors.

Knowledge such as passwords, some specific question and answers, possession such as bank card, MFA device/app, inherent such as fingerprint voice and facial expression and even physical location, and network IP are able to be used as factors. In AWS, we use MFA app to implement the MFA process, it is really convenient to get verified code via mobile apps.

The second thing is Identity and Access Management(IAM).

We know that AWS root account has the highest level of controlling of AWS account, there is no restriction on the root account. As a result, it is very dangerous to make manipulation with the root account and the IAM comes.

IAM has full trust from the AWS Account, and it can create three different type identities, IAM user, IAM group, IAM Role, and I will talk those three identities in the coming blogs.

Five characteristics of Cloud computing

With the rapid development of cloud computing technologies in giant IT companies such as Amazon, Microsoft and Google, more and more small companies are able to share the benefits from the big cloud computing companies’ infrastructure.

In recent years cloud computing is getting popular among the public, but rare people know what the exact cloud computing is. There are five crucial characteristics of cloud computing, and without everyone of them, it could not be cloud computing.

1. Clients of cloud computing should be able to provision capabilities as needed without requiring human interaction. On cloud platform, every client with specific cloud skills should be able to use service of the platform on his demand all by himself, and in the whole process, a guild from a real platform staff is not necessary.
2. All the service is available over the network and it’s accessible through standard mechanisms. It’s not necessary to use special network or port to use cloud service, clients are able to visit cloud service everywhere with standard protocols and standard network.
3. Resource Pooling. There is a sense of location independence. Clients of cloud service may only be able to select a very large scale of region to settle their resources, but they have no control or knowledge over the exact location of the resources. Resources are pooled to serve multiple consumers using a multi-tenant model.
4. Rapid Elasticity. Cloud computing capabilities can be elastically provisioned and released to scale rapidly outward and inward with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited. It’s not necessary to make a fixed capability plan in advance, as all demand is elastic in the real production environment, cloud computing must be adjustable and rapidly elastically.
5. Measured Service. Resource usage can be monitored, controlled, reported and billed. The fifth characteristic is the premise of “pay what you consume”. And that can promise the good quality of service to customers and decrease cost of customers.

29/September/2022

Use nested object to save and display blogs with readability

When I was developing my personal blog website, it was confused me that how to make my blog body more readable. In Yu’s course it is just simply saving a plain string into MongoDB as the text body, and as a result, when the data was collected from database and rendered on the browser, all the format was gone and the blog is not readable at all.

My previous solution is changing the structure of the blog object. At first my model of blog only has two properties such as title and content, and the content is also a plain string. Then I changed the blog object to a nested object. The content had been changed to an array instead of a plain string, and every element in that array is a object which has following properties: subtitle, first paragraphs, second paragraphs, third paragraphs fourth paragraphs, and a code paragraph. Those four paragraphs are all plain string saved as the properties in database, and the code paragraph is also save as string, but the format will be displayed.

When I render those strings of a single blog body in web page, I designed 3 basic elements in a blog content array. As a result, a blog body is able to contain as many as 15 paragraphs, and I used <pre> tag to display the formatted string in the code paragraph. This presents a readable structure of an blog article, and it also provide some flexible to write a small blog which does’t contain 15 paragraphs, as nothing will display in <p></p> if there is nothing between those <p> tag.

But the disadvantage is also obvious.

Firstly, the nested and complicated blog model makes the blog data take more storage room, and every time it cost more network transmit load. It is also more complicated to display each paragraph in a ejs file.

Secondly, it is not flexible enough to edit a customized blog. Even 15 paragraphs are enough for a normal blog, but is not enough for some ingenious and fancy design. The paragraphs displayed as string in <p></p> cannot change color and size, and it is not possible to display a image. Those all decreases the readability of the blog.

Thirdly, it is hard edit an old blog or compose a new blog since there are 15 blanks of paragraphs to fill, all users have to remember the structure of the 15 paragraphs. Those blog is not maintainable since every string in different paragraph saved as different properties in database.

This solution is partly solve the readability issue that using plain string as a blog, but the disadvantages of it are forcing me to look for some other ways to replace it. And I have found the Markdown language which is more flexible and it’s a better solution to solve this. The next blog I will present the markdown solution in my blog website.