

AWSMahjong Ver1.1

JAWS-UG

Rule (Mahjong)

- Basically same with Mahjong's rule
- ポン・チー・カン are allowed
- 喰いタン・後付け are not allowed
- Only 東風 ※No 親の連荘
- Start with 25000 Points. 30000 Points should be return.
 - Do not 符計算
 - Game is over when we played 4 games or if someone lost all of his points.
- If someone 上がる or 18 times ツモ will be the end of the game (流局)
- Consider JAWS-UG Pai as ドラ(Dora)
- If there are no CDP, it is 0 Point ※exception for 役満
- If there are CDP, One 翻 for one CDP ※Can not use Pai several times for making CDP

Rule (Score calculating)

	子(ロン)	子(ツモ)	親(ロン)	親(ツモ)
1翻	1000	500(500)	1500	1000オール
2翻	2000	1000(1000)	3000	1500オール
3翻	3000	1000(2000)	5000	2000オール
4翻	5000	1500(3000)	8000	3000オール
5翻(満貫)	8000	2000(4000)	12000	4000オール
6,7翻(跳満)	12000	3000(6000)	18000	6000オール
8,9,10翻(倍満)	16000	4000(8000)	24000	8000オール
11,12翻(三倍満)	24000	6000(12000)	36000	12000オール
13翻(役満)	32000	8000(16000)	48000	18000オール

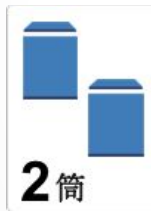
Rule (Combination)

● 刻子



Gather two Pai of the same type. カン is available with four Pai.

● 順子



Gather number of the same type ※but, something like 891 are not allowed.

● 対子

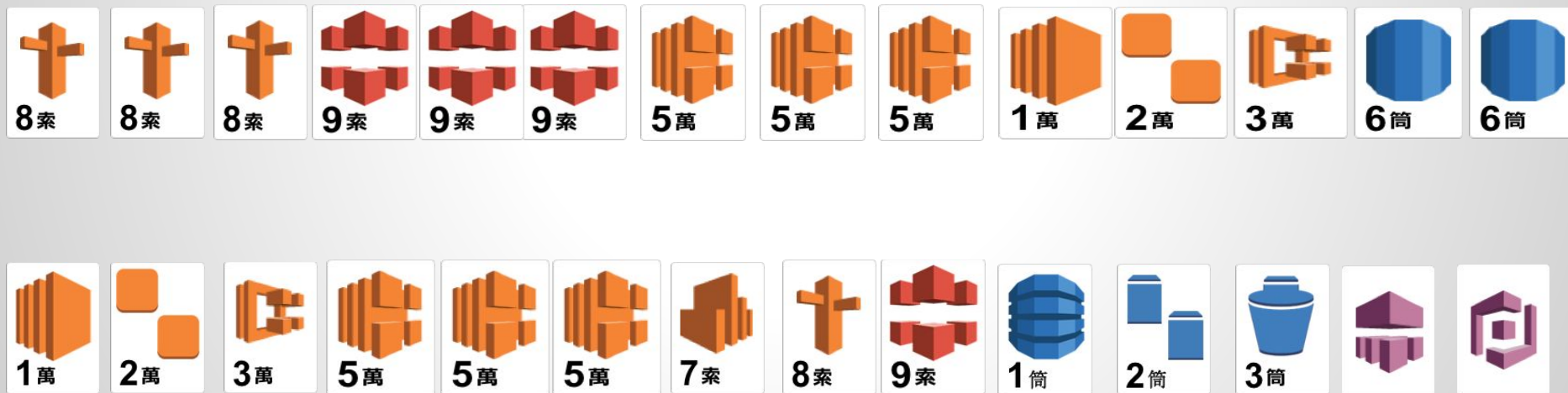


Gather two Pai of the same type.

Rule (for win)

● Win by making Four 刻子 or 順子, One 对子

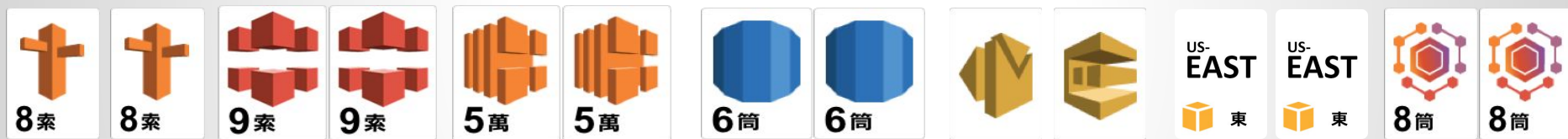
例)



Rule (exception)

● exception pattern

例) 七対子



例) 国士無双



役 (Common役)

● 1翻

- 立直 (門前)、一発 (門前)、門前清模和 (門前)、平和 (門前)、断ヤオ、一盃口 (門前)、役牌 (白、發、中、門風牌、莊風牌)、嶺上開花、海底撈月、河底撈魚、槍槓、ドラ

● 2翻

- ダブル立直、全帯 (鳴き1翻)、混老頭、三色同順 (鳴き1翻)、一気通貫 (鳴き1翻)、対々和、三色同刻、三暗刻、三槓子、小三元、七対子 (門前)

● 3翻

- 二盃口 (門前)、純全帯 (鳴き2翻)、混一色 (鳴き2翻)

● 6翻

- 清一色 (鳴き5翻)

役(役満)

- 四暗刻
- 四暗刻単騎(double)
- 清老頭
- 四槓子
- 大三元
- 字一色
- 小四喜
- 大四喜
- 国士無双
- 国士無双十三面待ち(double)
- 天和
- 地和

※ There are no 九蓮宝燈、緑一色、大車輪.

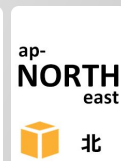
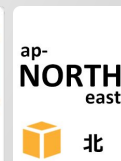
AWS役(役満)

● Reference Web architecture(役満)

Route53



● Sushiro一(役満)



Special あがり (Mahjong limited)

● Reference Web architecture (役満)

Route53

CloudFront

ELB

EC2

RDS
(Multi-AZ)



● Sushiro一 (役満)



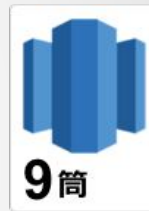
Special役

● Kinesushi
2翻 with 暗カン、1翻 with 明カン

● Docomo
2翻 with 暗カン、1翻 with 明カン

● Security Can
2翻 with 暗カン、1翻 with 明カン

● AWS Robo
2翻 with 面前



AWS Mahjong ▪ Donjara

CDP list Ver1.1

JAWS-UG

Type of card



Details about each card 萬子 (Manzu)



EC2



Elastic
Beanstalk



Auto Scaling



Instances



Elastic Load
Balancing



Amazon Lambda



EC2
ContainerService



AMI



Amazon Kinesis

※青字は変更

Details about each card 筒子 (Pinzu)



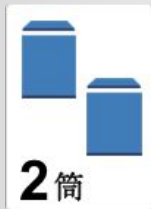
Dynamo DB



bucket



AWS IoT



Amazon EBS



ElastiCache



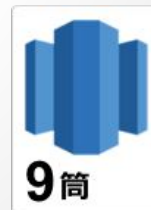
Mobile Hub



snapshot



RDS



Amazon Redshift

※青字は変更

Details about each card

索子 (So-zu)



Direct Connect



EMR



QuickSight



CloudWatch



Machine Learning



Route53



elastic network
instance



S3



Cloud Front

※青字は変更

Details about each card 三元牌(Sangenhai)



WAF



Cognito



Amazon
CloudSearch



CloudTrail



Device Farm



Amazon SES



Inspector



MobileAnalytics



Amazon SQS



IAM



SNS



API Gateway

※correspond to 白

※correspond to 撥

※correspond to 中

Details about each card 風牌 (Fanpai)



リージョン
バージニア



JAWS-UG



リージョン
シンガポール



JAWS-UG
Enterprise



リージョン
カリフォルニア



JAWS-UG
Middle Line



リージョン
東京



JAWS-UG
Girls Party

About JAWS-UG card
JAWS-UG card is treated as Dora(ドラ) in Mahjong.
But, in Donjara, it's treated as Almighty (オールマイティ)

List of CDP Ver1.1

CDP



Snapshot

Create snapshot of the data of certain moment(to back up) and save it in S3. So we can restore data at any moment. Auto Backup with API is commonly used.



Stamp Pattern

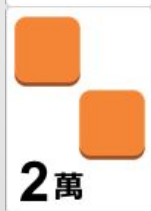
Can make same enviroment through AMI. It is very helpful when we need several system with same enviroment setting.

CDP



Web Storage

Distribute Big files or static contents through S3 prevent overload in EC2. But when it comes to Dynamic contents, use EC2.



Cache Distribution

No delay in distribution by using Cloudfront. The sites will be more fast and get better response to user and reduce EC2's access load.

CDP



Direct Hosting

It is possible to make error-free static web site by using R53、Cloudfront、S3.



Job Observer

By using SQS, do autoscaling when it is overpass certain level setted in CloudWatch. Control EC2's number(Scale in/out) according to the load in EC2.

CDP



Back Net

By preparing two of ENI (virtual network interface) in EC2, make and use the public network interface and another network interface for management.



State Sharing

By saving state information (such as session information, user information) to DyanamoDB(or to Redis), prevent the loss of state information when the server access is increased or decreased rapidly.



CDP



Inmemory DB Cache

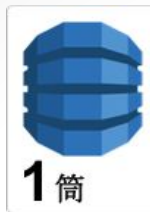
By making cache of the frequently used data in Redis, we can use those data from Redis, not DB.



Scheduled Autoscaling

When it is already known when the access increase rapidly, it is possible to prevent the server error by doing auto scale out at scheduled timing.

CDP



Storage Index

When storing data in the Internet storage, store the meta-information simultaneously to KVS (Dynamo DB, which has high performance in searching), to use that information as an index. When we search the data, use KVS (Dynamo DB) first, and on the basis of the results from KVS, access to the Internet storage.



Multi Load Balancer

By preparing multiple ELB, it is possible to change the behavior for each ELB at the same web site. It is possible to change the access destination in web by using the ELB.

CDP added



Samurai IoT Tsuji

The main pattern that Mr.Tsuji loves. (but, this pattern is easy to get in the trouble When performing the demo LT) Process The sensor data that Kinesis get with the Lanbda service and save it to the Dynbamo DB. And deal with the data as we want.



IoT starter pack pattern

The most desired configuration if this is the first time to start IoT in WS. AWS IoT directly save the sensor data that is from divice through MQTT protocol to the Dynamo DB. And visualize those data with QuickSite. But unfortunately QuickSite is still in preview.

CDP added



Multi region pattern

You can keep a low latency because they distribute loads automatically to close Region when balancing the system which are distributed over a wide area.



The well-used pattern for cloud migration

This is very similar to the configuration of on-pre-systems. And this is also the major CDP for enterprise. It is hybrid but it is okay for a while. The good results will come.

CDP added



Brooklyn Deployment pattern

The safe release is possible by using the Route53 and Beanstalk. If there are any trouble on it, all you need to do is doing roll back. So, let's develop more freely.

Brooklyn Deployment ECS pattern



The more advanced system than the system written above. This system make it possible to release product more safely by using Route53 and ECS. If release success, It discard the older one automatically.

CDP added



BI pattern

The major pattern of BI configuration which use AWS. All data are scraped by Kinesis and saved to Redshift. And then, QuickSight will show the results.



Deep learning pattern

The advanced version of BI configuration which use AWS. All data are saved to Redshift and the machine learning algorithm will run. And we will see the future...

CDP added



Combination for web site security

The strongest combination for security. Don't be afraid of DDos, XSS and etc.



Combination for mobile environment

The most powerful combination for the mobile development. With this combination, any mobile application development goes well.

CDP added



the pattern of observation

LLimited edition of CDP in AWS Mahjong.

CloudWatch monitor each services and if it is needed use SES for alarming. And it is also possible to get history of the usage of AWS API.

And it is also possible to analysis operational status by using the log in S3.



server-less api pattern

WWS Mahjong limited CDP. This configuration is becoming a new common sense of the REST AP. With this configuration you can easily publish the code running on the Lambda as API. And you can manage it very easily.

CDP added



Schedule backup pattern

Make the backup of EC2 or RDS by using Lambda and save it to snapshot. This easily solve the trouble of making backup.



A certain app release pattern

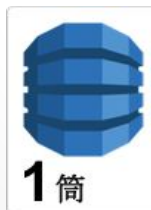
It is possible to search and register the word by using the Mobile Hub and the CloudSearch via the mobile app .

CDP added



The pattern for proper analysis

This is the CDP for the analysis system engineers who think Redshift is not enough and who think Hadoop and machine learning is needed.



Cloud-native pattern

There are a lot of cloud native engineers who do not use EC2. But, The thing is they do not hate EC2.

Special (Mahjong limited)

- reference Web architecture (役満)

Route53



8索



8索



8索



9索



9索



9索



5萬

CloudFront



5萬

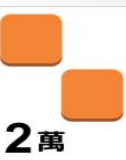


5萬

ELB



1萬



2萬

EC2



3萬

RDS
(Multi-AZ)



6筒



6筒

- Sushiro (役満)



1萬



2萬



3萬



9萬



9萬



9萬



4筒



4筒



4筒



9筒



9筒



9筒



ap-NORTH
east
北



ap-NORTH
east
北

Special役

- Kinesushi
- Docomo
- Security can
- AWS robo

