mijin Deployment Guide

Release 1.3 (English)

Techbureau, Corp

Jul 28, 2025

Contents:

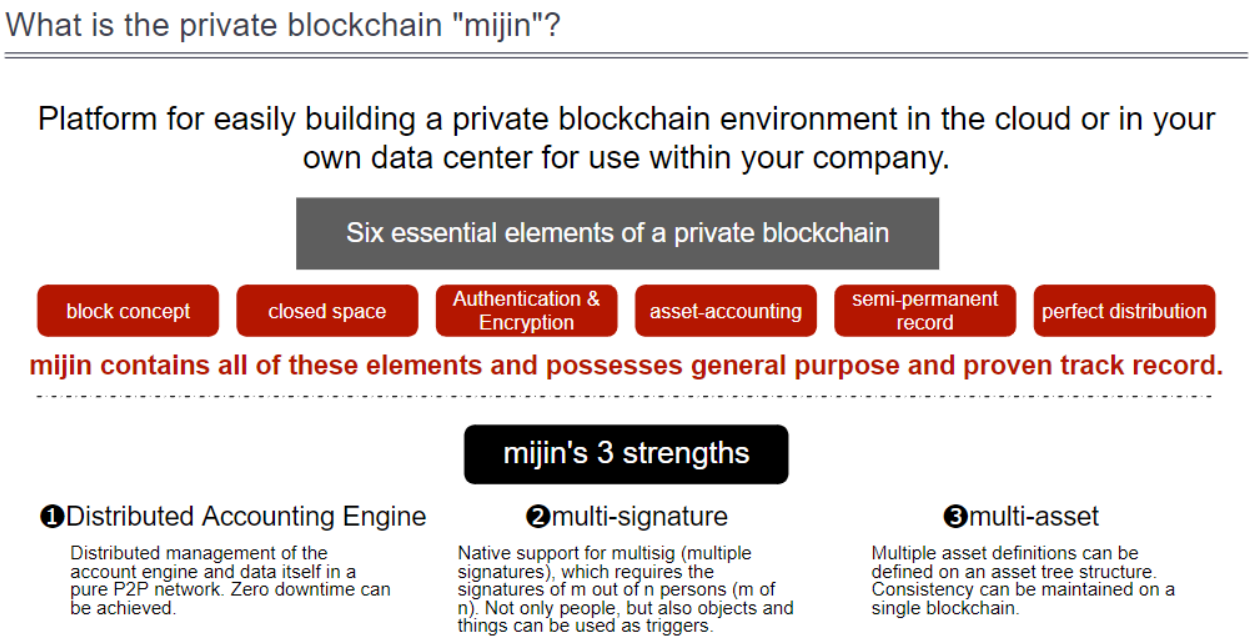
About

1.1 About mijin Catapult(v.2)

1.1.1 What is mijin Catapult(v.2)?

mijin Catapult(v.2) allows you to build a viable private blockchain with minimal resources.

As a private blockchain product that replaces existing databases, it utilizes multiple cryptographic signature functions and is used in many projects.
Without compromising the atomicity of transactions, mijin's account engine operates as a fully distributed, zero-downtime network that assumes high security, minimizing the need for application development and maintenance.
Tokenize any data or asset with a simple definition and process all transactions as secure, one-time smart contracts, with speeds of thousands of transactions per second.



mijin Catapult(v.2) will offer new features such as **aggregate transactions** (1,000 transactions

simultaneously with atomicity) and Fully distributed atomic swap technology with guaranteed atomicity through **Multilayer Signature** (3 levels of multisig authentication).
All of these advanced features can now be implemented with a minimum of knowledge and man-hours, thanks to the feedback we have received from our customers over more than five years of practical cases in V1.
The combination of this experience and technology with NEM, a public blockchain that has been in operation for many years, is now being offered for the first time as a commercial blockchain product from Japan on the AWS Marketplace.
The following site also provides a detailed explanation of mijin Catapult(v.2).

mijin Catapult(v.2) Description Page
https://mijin.io/product/

1.1.2 Use Cases

mijin Catapult(v.2) can be used effectively in a variety of fields.

Note:

If you would like to know more details, you can obtain actual case studies from the various application fields section of the mijin Site page or by contacting us.


Used in Various Industries
Financial Services
Settlements, currency exchange, remittances, savings, etc.; stock trading, BITCOIN transactions, overseas remittances, social banking
Authentication
ID, copyright, ownership, various certifications
Healthcare
Medical information
Point Systems
Gift certificates, rewards for artists, prepaid cards, reward tokens
Sharing
Ride sharing, space sharing
Data Storage
Fundraising
Artist equity transactions, crowdfunding
Commercial Distribution / Distribution Management
Supply chain, tracking management, marketplaces, management of digital assets, transfers
Asset management
Communication
SNS, messenger transactions
Content
Games, electronic books, streaming
Education / Human Resources
Scholastic history, curriculum vitae, work history
IoT
Manufacturing, sensing, mining chips
Public Sector
Budget transparency, voting, INDIVIDUAL NUMBER management
Other
Media, infrastructure, etc.

1.1.3 Differences from the public blockchain Symbol

Since mijin Catapult(v.2) uses the same core engine (Catapult) as Symbol, which is launched as a public blockchain, what can be done with Symbol can also be done with mijin Catapult(v.2).
This section mainly shows the difference from Symbol.

Table 1: mijin vs. symbol

item	mijin Catapult(v.2)	Symbol
chain	private blockchain	public blockchain
network name	MIJIN, MIJIN_TEST	MAIN_NET, TEST_NET
Genesis Block Epoch	1560294000s (Tue 11 Jun 2019 23:00:00 UTC)	1615853185s (Tue 16 Mar 2021 00:06:25 AM UTC)
Base Currency NAME	cat.currency	symbol.xym
Base Currency ID	Create ID when building mijin Catapult(v.2)	6BED913FA20223F8
Base Currency Issue Volume	8,998,999,998.000000	8,164,233,299.724038
Harvest Currency Name	cat.harvest	symbol.xym（基軸通貨と兼用）
Harvest Currency ID	Create ID when building mijin Catapult(v.2)	6BED913FA20223F8
Harvesting Currency Issue Volume	15,000,000	8,164,233,299.724038（symbol.xym）
block generation interval	10〜60 秒（構築時にカスタム可能）	30 秒
1 ブロックあたりの最大トランザクション数	6,000 10,000 20,000 50,000 100,000	5,000
transaction fee	required なし（構築時に指定可能）	required
Mosaic Rental Fee	あり（50cat.currency） なし（構築時に指定可能）	あり（50symbol.xym）
Namespace Rental Fees	あり（期間によって変動） なし（構築時に指定可能）	あり（期間によって変動）
VotingKey ファイルの期限	約 547〜3285 日（ブロック生成間隔により異なる）	Approx. 180 days
Maximum number of signatures per account	25、50、100、1000（選択可能）	25
Finalization support	決定的、確率的（構築時に選択可能）	Deterministic

1.2 Comparison with major database products and applications

1.2.1 Comparison with major database products

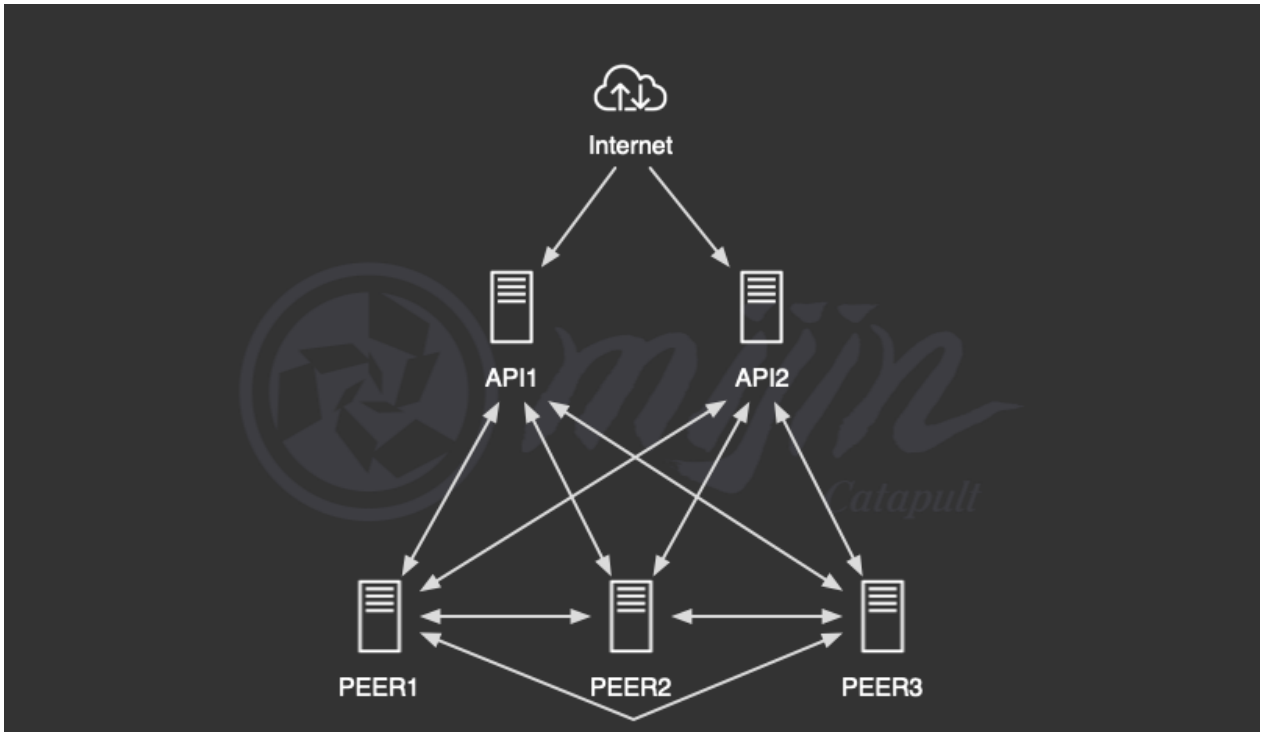
Table 2: 各種データベースの比較

Item	mijin v.1	mijin Catapult (v.2)	MySQL	Redis	Mongo	Neo4j	Apache Hbase
type	blockchain	blockchain	RDBMS	NoSQL（KV 型）	NoSQL (document type)	NoSQL (graph type)	NoSQL (wide column type)
Overview	Data is stored in blocks, DB to be connected like a chain	ditto	Consistency is maintained in tabular form, Easily searchable by query	With a key-value structure High speed access available	Schemaless JSON format, etc. DB that can be stored and operated	The relationship is represented by a graph structure, Fast merging is possible	Each column has a different structure, Fast tabulation is possible
ライセンス	Enterprise	Enterprise	GPL / Enterprise	BSD / Enterprise	SSPL / Enterprise	GPL / AGPL / Enterprise	Apache 2.0
Release Date	September 2015	June 2019	1995 年	2009 年	2009 年	2007	2010
development language	Java	C++	C++	C	C++	Java	Java
Supported OS	Linux, Windows	Linux, Windows (要 Docker)	Linux, Windows	Linux, Windows	Linux, Windows	Linux, Windows	Linux
client	REST API, nem-sdk	REST API, symbol-sdk (Java, TS)	mysql-client, each language library	redis-client, each language library	mongo-client, each language library	REST API, WEB UI each language library	HBase client, RPC-enabled library
advantage	Highly tamper-resistant, Easy construction of account assets	Same as above	Powerful query, integrity guaranteed informative	Simple API, Memory-resident and fast	Schemaless, 水平スケーラブル	horizontally scalable Query support	Column oriented and fast tabulation, ビッグデータに強い
主な用途	Point management, audit log	Point management, P2P transactions	Customer Management	Session/message relay	Logs, games, surveys	Recommendation, relationship analysis	Analytical, statistical, and tabulation applications

1.3 Architecture and Specification Requirements

1.3.1 mijin Catapult(v.2) configuration

The mijin Catapult (v.2) comprises a county of servers, called nodes, that make up a private blockchain network.
The minimum recommended configuration for mijin is two API nodes and three PEER nodes.



1.3.2 PEER node roles and specification requirements

PEER nodes generate blockchain data and provide block consensus functionality.
The PEER node alone cannot be accessed by clients; an API node is required.

CPU	Processor around 3.1 GHz (CPU core 2 or higher)
memory	4GB RAM or more
disk	root 30GB or more block 500GB or more 3000IOPS or more recommended
OS	Linux running Docker (recommended Ubuntu 20.04 or higher)

Note:

AWS MarketPlace's mijin Catapult(v.2) allows you to select only available specs that are above recommended.
Disk space increases in proportion to the amount of storage on the blockchain.

1.3.3 API node roles and specification requirements

The API node provides an API to write and read blockchain data to mongodb at high speed.
There is also blockchain data on the API node, but it is treated as a simple backup with no block generation function.
This blockchain data generation function can also be enabled and can function as a DUAL mode with PEER node functionality.

CPU	Processor around 3.1 GHz (CPU core 2 or higher)
memory	8GB RAM or more
disk	root 30GB or more mongo 300GB 3000IOPS or more recommended block 500GB 3000IOPS 以上推奨
OS	Linux running Docker (recommended Ubuntu 20.04 or higher)

Note:

AWS MarketPlace's mijin Catapult(v.2) allows you to select only available specs that are above recommended.
Disk space increases in proportion to the amount of storage on the blockchain.

Deploy

2.1 mijin Catapult(v.2) How to Deploy

2.1.1 How to deploy mijin

There are currently two ways to build mijin Catapult(v.2).

1. Deployment using AWS MarketPlace
2. Built by Tech Bureau

2.1.1.1 Deployment using AWS MarketPlace

We offer products that make it easy to deploy|mijin| on the Amazon Web Service (AWS) marketplace, one of the largest cloud vendors.

AWS MarketPlace
https://aws.amazon.com/marketplace

For more information, see Preparing to Deploy with AWS MarketPlace for more information.

2.1.1.2 Built by Tech Bureau

mijin construction will be performed at Tech Bureau.
For more information, please contact us through the contact page of mijin Site.

2.2 AWS MarketPlace

AWS MarketPlace will be an AWS service to source and provision third-party software, services, and data.
This chapter describes mijin Catapult(v.2) deployed in AWS MarketPlace.

2.2.1 Preparing to Deploy with AWS MarketPlace

An Amazon Web Services (AWS) account and knowledge of its operation are required to use mijin Catapult(v.2), which is available on AWS Marketplace.
This chapter will help you understand what you will need to prepare in advance of deployment.

2.2.1.1 Prepare AWS account

Amazon Web Services (AWS) will be the platform for creating infrastructure technologies such as computing, storage, and databases in the cloud.
To perform the operation, you will need to create an AWS account, so please refer to the following site and create an AWS account.

https://aws.amazon.com/jp/register-flow/

Warning:

mijin Catapult(v.2), it is recommended to create and deploy an administrative IAM account instead of the root user.
To create an administrative IAM user, see below.
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/getting-started_create-admin-group.html
If you want to know more about root users, please refer to the following
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_root-user.html

2.2.1.2 Knowledge of AWS

The main AWS services used by mijin are
The following will be automatically created when deploying mijin Catapult(v.2) provided by Marketplace.

- **Amazon VPC**

  Create a virtual network. mijin Catapult(v.2) will be placed in a network on this VPC.
  For more information, check the following documents
  https://docs.aws.amazon.com/ja_jp/vpc/latest/userguide/what-is-amazon-vpc.html

- **Amazon EC2**

  Create a computing machine. mijin Catapult(v.2) nodes (servers) are created.
  For more information, check the following documents
  https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/concepts.html

- **Amazon EBS**

  Create storage on the computing machine.
  mijin Catapult(v.2) places blockchain and mongo data on EBS.

For more information, check the following documents
https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/AmazonEBS.html

- **Elastic Load Balancing**

  Create a load balancer to relay to the computing machine.
  mijin Catapult(v.2) ensures redundancy by routing access to API nodes through a load balancer.
  For more information, check the following documents
  https://docs.aws.amazon.com/ja_jp/elasticloadbalancing/latest/userguide/what-is-load-balancing.html

- **Amazon Route 53**

  Create a DNS service to be used for name resolution of computing machines.
  mijin Catapult(v.2) connects each node by name resolution using DNS.
  For more information, check the following documents
  https://docs.aws.amazon.com/ja_jp/Route53/latest/DeveloperGuide/Welcome.html

- **AWS IAM**

  Specify users and groups that have access to AWS services and resources, such as between AWS services, and
  Centralized management of fine-grained access permissions.
  mijin Catapult(v.2) When deploying, the AWS account must have IAM privileges. (Explained in Account authorization to deploy to AWS)

  mijin Catapult(v.2) to resources created after deployment.
  ・Authority from EC2 instance to Secrets Manager
  ・SSM login privileges from EC2 instances
  to the profile of the EC2 instance that is the API,PEER node.

  For EC2 IAM roles, see below.
  https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

  For more information on SSM login, please see below.
  https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/session-manager-getting-started-instance-profile.html

- **AWS Systems Manager Parameter Store**

  Use as secure tiered storage for configuration data management and confidentiality control.
  mijin Catapult(v.2) stores configuration information and common settings for each node.
  By storing them in the parameter store, they can be restored in the event of a failure.
  For more information on the parameter store, see
  https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/systems-manager-parameter-store.html

- **AWS CloudFormation**

Provides orchestration to automatically build AWS services.
AWS Marketplace uses AWS CloudFormation to build mijin Catapult(v.2) on AWS.
For more information, check the following documents
https://docs.aws.amazon.com/ja_jp/AWSCloudFormation/latest/UserGuide/Welcome.html

2.2.1.3 Account authorization to deploy to AWS

To deploy mijin Catapult(v.2) on AWS, you need to grant privileges to the account you prepared with Prepare AWS account.
Permissions to create Knowledge of AWS AWS resources and to subscribe to use the mijin Catapult(v.2) images deployed in Marketplace are required.

An IAM policy focused only on AWS Marketplace and Deploy permissions is as follows
Refer to the following to create an IAM policy and grant it to the IAM account to be used for deployment.
https://docs.aws.amazon.com/ja_jp/apigateway/latest/developerguide/api-gateway-create-and-attach-iam-policy.html

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "iam:*RolePolicy",
                "route53:*HostedZone",
                "iam:List*",
                "aws-marketplace:*",
                "elasticloadbalancing:RegisterTargets",
                "ec2:*RouteTable*",
                "iam:CreateRole",
                "elasticloadbalancing:DeleteLoadBalancer",
                "ec2:DescribeInternetGateways",
                "elasticloadbalancing:DescribeLoadBalancers",
                "ec2:*KeyPairs",
                "lambda:GetFunction*",
                "ec2:DescribeAccountAttributes",
                "elasticloadbalancing:ModifyTargetGroupAttributes",
```

(continues on next page)

(continued from previous page)

```
                "cloudformation:ListStackResources",
                "iam:GetRole",
                "elasticloadbalancing:DescribeTarget*",
                "elasticloadbalancing:CreateTargetGroup",
                "route53:ChangeResourceRecordSets",
                "iam:DeleteRole",
                "cloudformation:GetTemplate*",
                "ec2:RunInstances",
                "elasticloadbalancing:*Listener",
                "ec2:*Addresses",
                "cloudformation:GetStackPolicy",
                "cloudformation:DeleteStack",
                "ec2:DeleteVpc",
                "iam:GetAccountSummary",
                "ec2:*Tags",
                "route53:GetChange",
                "iam:*InstanceProfile*",
                "ec2:*InternetGateway",
                "ec2:CreateVpc",
                "sns:ListTopics",
                "route53:ListQueryLoggingConfigs",
                "cloudformation:*ChangeSet*",
                "cloudformation:EstimateTemplateCost",
                "elasticloadbalancing:DescribeListeners",
                "ec2:DescribeAvailabilityZones",
                "ec2:ModifyVpcAttribute",
                "ec2:*Route",
                "ec2:ReleaseAddress",
                "cloudformation:ListStacks",
                "elasticloadbalancing:CreateLoadBalancer",
                "ec2:TerminateInstances",
                "elasticloadbalancing:DeleteTargetGroup",
                "ec2:AllocateAddress",
                "cloudformation:DescribeStack*",
                "lambda:AddPermission",
                "cloudformation:CreateStack",
                "ec2:DescribeVpcs",
                "lambda:*Function",
                "ec2:*NatGateway*",
                "lambda:RemovePermission",
                "ec2:*SecurityGroup*",
                "ec2:*Subnet*",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

2.2.1.4 mijin Catapult(v.2) license in AWS

AWS Marketplace licenses for the mijin Catapult(v.2) product version are made available by agreeing to the EULA for mijin Catapult(v.2) in Marketplace.
For the EULA, please refer to the following
https://d7umqicpi7263.cloudfront.net/eula/product/d6b2653b-ee61-4a62-8fef-a9fa7930892e/c255cb3f-6c72-412a-a899-42fa3f83fd71.pdf

The license fee for the AWS Marketplace version of mijin Catapult(v.2) will be added to your AWS usage fee and charged on a pay-as-you-go **per-hour** basis.
It does not matter what the specifications or number of CPU cores are, it is per node.

item	Value
mijin license fee	$0.40 per hour

Note:

AWS MarketPlace Trial Version is not subject to the mijin Catapult(v.2) license fee.
AWS MarketPlace Enterprise x86_64 Version has a minimum of 5 node activations, so the approximate license cost per month is calculated as follows

Hourly
$0.4/1h * 5(units) = $2.0/1h
by the day
$2.0/1h * 24(h) = $48.0/1d
In units of 1 month (30 days)
$48/1d * 30(h) = $1,440.0/1m

2.2.1.5 AWS Usage Fees

In addition to the mijin license fee, AWS resource usage fees are

- Amazon EC2
- Amazon EBS
- Elastic Load Balancing
- Amazon Route53
- Amazon VPC(Nat Gateway)
- parameter store

The cost of the pay-as-you-go service will be charged on a pay-as-you-go basis.
The fee also varies depending on the amount of data transferred and other factors.
For information on AWS usage fees, please refer to the following

https://aws.amazon.com/jp/pricing/

Note:

The configuration of mijin Catapult(v.2) offered on AWS Marketplace changes depending on the parameters you specify, and the cost will vary.
Note that the following will vary, depending primarily on the parameter specification.

- instance type
- Number of nodes (instances)
- Whether Elastic Load Balancing is required or not
- Whether VPC is created or not
- Block size and IOPS of EBS

AWS Marketplace Product Description

The products currently offered by Tech Bureau are as follows
Each product has its own product page on AWS Marketplace.

- Trial Version (AWS MarketPlace Trial Version)
- Product version (AWS MarketPlace Enterprise x86_64 Version)
- Product version (AWS MarketPlace Enterprise arm64 Version)

The mijin Catapult(v.2) offered in the AWS Marketplace does not require any complex operations to build.
Depending on the parameters chosen at deployment time, it automatically builds an environment that includes a network and builds a secure and robust blockchain network from an image that includes mijin.

The contents of the environment to be created are explained on each deployment method page.

2.2.1.6 Trial Version

The trial version is available free of charge as a simple to start mijin Catapult(v.2).
There is no redundancy and it will be a single operation mijin Catapult(v.2).
Compared to the commercial version, it is a single-AZ arrangement with no customizability, but it is suitable for a trial or development environment.

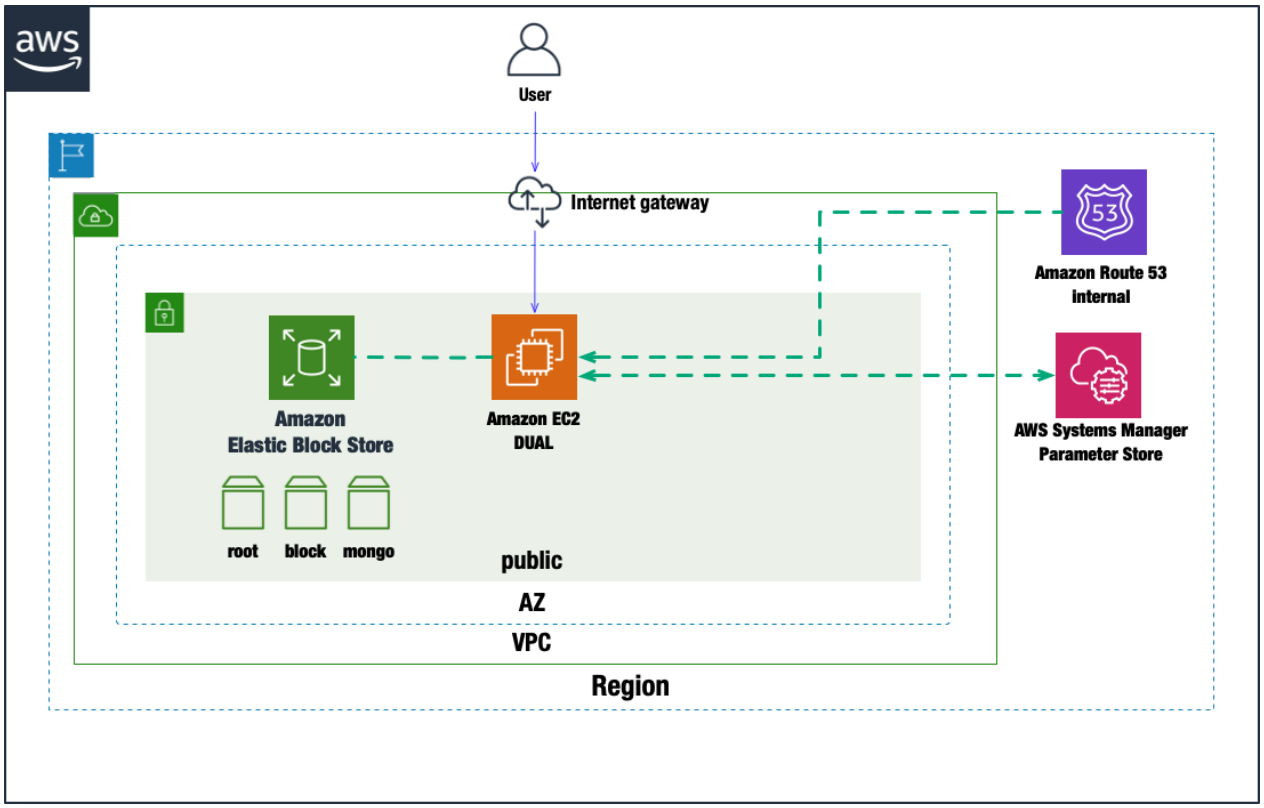


Table 1: Free Trial Environment Overview

item	Description
Marketplace Product Page	AWS MarketPlace Trial Version
デプロイ方法	Build a new VPC with mijin Catapult(v.2). Deploying a trial version of mijin を参照
Deployment time	約 15 分
Support	基本サポートなし AWS に関する問題は以下で問い合わせ： https://aws.amazon.com/jp/premiumsupport/
available region	Available in 21 regions worldwide ap-northeast-1, us-west-1, us-west-2, us-east-1, us-east-2, eu-north-1, eu-west-1, eu-west-2, eu-west-3, eu-south-1, af-south-1, ap-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-southeast-1, ap-southeast-2, sa-east-1, ca-central-1, eu-central-1, me-south-1
Number of nodes	1 台（DUAL モード）
placement availability zone	シングル AZ（1 つ）
load balancer	None

2.2.1.7 Production version

The full version is customizable for enterprise production, with fault-tolerant and high-availability configurations as standard, and secure API access can be easily configured.
In addition, you can set up a new or existing environment with mijin Catapult(v.2) and apply it to various environments, such as load balancing using load balancers.

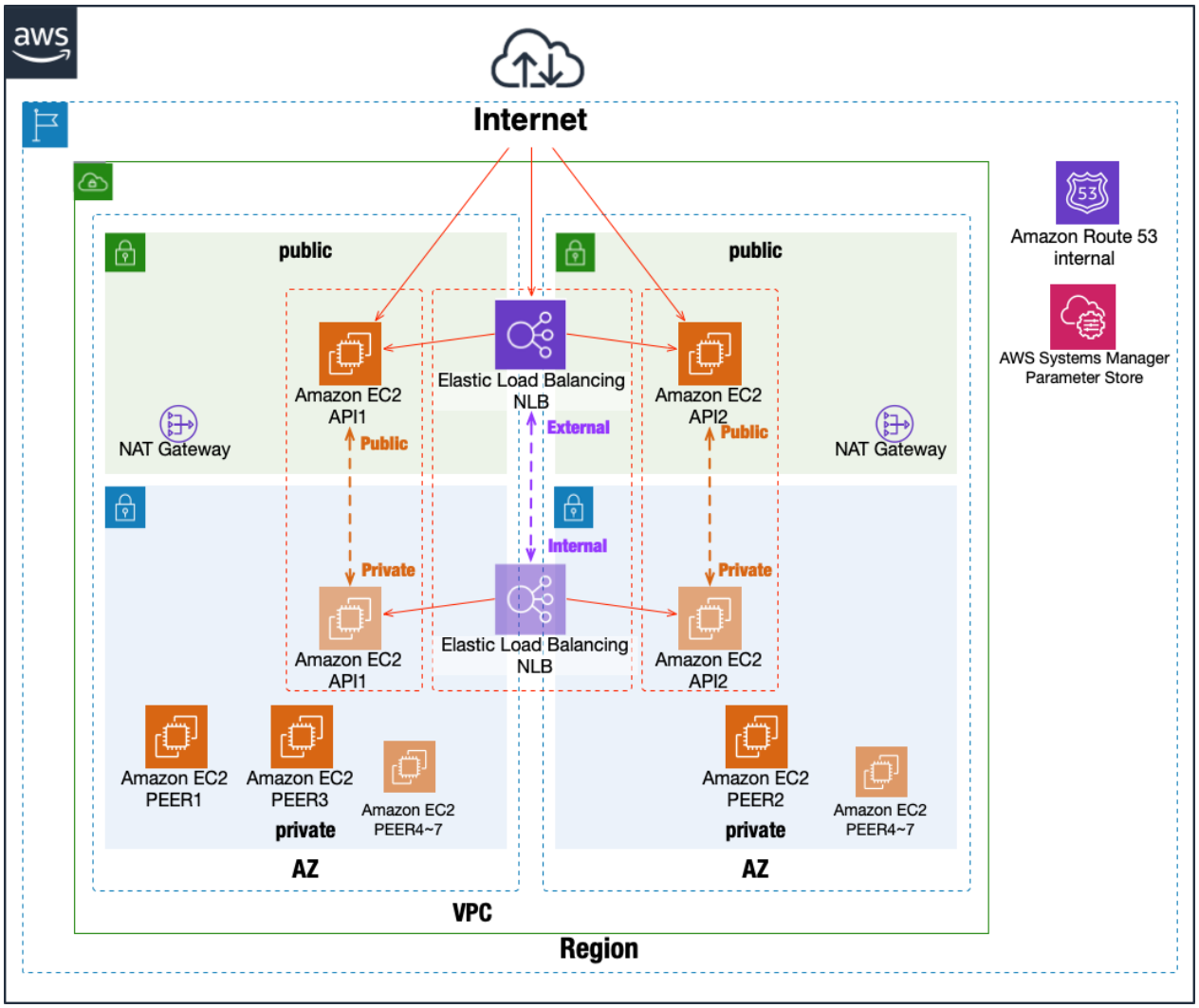


Table 2: AWS デプロイ情報

item	Description
Marketplace Product Page	x86_64 版：AWS MarketPlace Enterprise x86_64 Version arm64 版：AWS MarketPlace Enterprise arm64 Version
既存環境へのデプロイ方法	すでに既存サービスがあり、 同じ VPC 内で mijin Catapult(v.2) を管理したい場合は Deploy mijin on an existing VPC を参照
新規環境へのデプロイ方法	mijin Catapult(v.2) Technical Inquiries 初めて AWS を使う場合は Create a new VPC and deploy mijin を参照
Deployment time	約 30 分
Support	デプロイ関連のお問い合わせは https://mijin.io/aws_contact/ ※ mijin Catapult(v.2) に関する技術的な質問は Paid support (ticket) For AWS-related information, go to AWS Support Case: https://aws.amazon.com/jp/premiumsupport/
available region	Available in 21 regions worldwide ap-northeast-1, us-west-1, us-west-2, us-east-1, us-east-2, eu-north-1, eu-west-1, eu-west-2, eu-west-3, eu-south-1, af-south-1, ap-south-1, ap-east-1, ap-northeast-2, ap-northeast-3, ap-southeast-1, ap-southeast-2, sa-east-1, ca-central-1, eu-central-1, me-south-1
Number of nodes	2 API nodes, 3 to 7 PEER nodes (5 to 9 total)
配置 AZ（アベイラビリティゾーン）	Multi AZ (2) ※ Single AZ cannot be set.
load balancer	必要に応じて設定可能

2.2.1.8 About Paid Support

Support for deploying mijin Catapult(v.2) will be provided free of charge, but if you need technical inquiries about mijin Catapult(v.2), you can purchase paid support (ticket-based) from Tech Bureau to assist you.
Please contact us below to inquire about purchasing support.

https://mijin.io/aws_contact/

Support Contents	mijin Catapult(v.2) Technical Inquiries Node failure support (during business hours) Version upgrade notification and procedure release Infrastructure support in an AWS environment. etc

2.2.1.9 Limitations due to AWS service quotas

AWS Marketplace mijin Catapult(v.2) uses the following AWS services, so if you use an AWS account that already has an environment built, there is a possibility of startup failure due to service quotas. For service quotas, see the explanation below.

https://docs.aws.amazon.com/ja_jp/general/latest/gr/aws_service_limits.html

Below is a description of the number of services created by mijin Catapult(v.2).

Table 3: AWS Services and Restrictions

AWS Services	Content related to restrictions
Amazon VPC	VPC: 1 サブネット: 4（トライアルは 1） インターネットゲートウェイ: 1 NAT Gateway: 2（トライアルは 0） ルートテーブル: 1 セキュリティグループ: 5 参考: https://docs.aws.amazon.com/ja_jp/vpc/latest/userguide/amazon-vpc-limits.html
Amazon EC2	EC2 インスタンス: 5〜9（トライアルは 1） インスタンスタイプによりクォータ変動あり EC2 制限: https://docs.aws.amazon.com/ja_jp/general/latest/gr/ec2-service.html オンデマンド制限: https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html#ec2-on-demand-instances-limits EBS: 630GB〜 https://docs.aws.amazon.com/ja_jp/general/latest/gr/ebs-service.html
Elastic Load Balancing	Network Load Balancer: 1（有効時） https://docs.aws.amazon.com/ja_jp/elasticloadbalancing/latest/network/load-balancer-limits.html
Amazon Route53	Internal ゾーン: 1 https://docs.aws.amazon.com/ja_jp/Route53/latest/DeveloperGuide/DNSLimitations.html
AWS IAM	IAM ロール: 2（トライアルは 1） IAM ポリシー: 2（トライアルは 1） https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/reference_iam-quotas.html
Systems Manager Parameter Store	パラメーター数: 48〜(トライアルは 15) https://docs.aws.amazon.com/ja_jp/general/latest/gr/ssm.html

2.2.2 Create a new VPC and deploy mijin

This chapter shows how to deploy mijin Catapult(v.2) on a new network (VPC).
In the commercial version, you can flexibly build a network that fits your environment by changing parameters.

2.2.2.1 List of services to be built on AWS by deployment

- Amazon EC2 (APINode x 2 PEERNode x 3~)
- Amazon EBS
- Elastic Load Balancing
- Amazon Route53
- Amazon VPC(Nat Gateway)
- parameter store

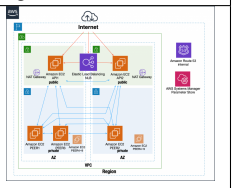
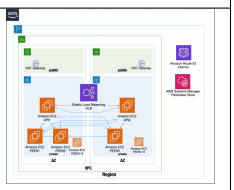
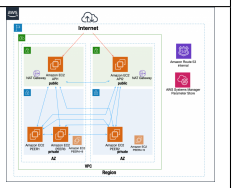
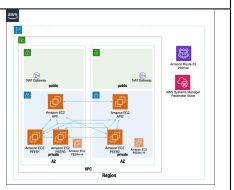
Note:

Elastic Load Balancing may not be created in some cases, depending on the parameter settings specified during deployment.

2.2.2.2 View Network

In the commercial version, it is possible to build a network that fits your environment by changing parameters. An example pattern is shown below.

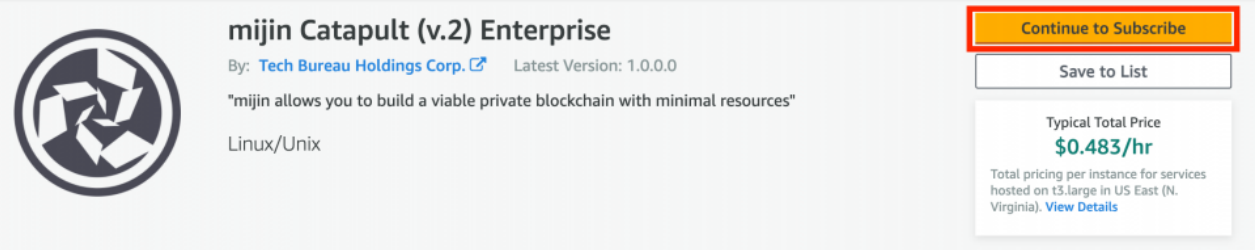
Table 4: ロードバランサーとノード配置パターン

No	load balancer	API node placement	PEER node placement	figure
1	** Yes / Open NW** UseLoadBalancer: `Yes` LoadBalancerType: `external`	**公開 NW** ApiPlacementNetwork: `Public`	Private NW	
2	**あり / 非公開 NW** UseLoadBalancer: `Yes` LoadBalancerType: `internal`	**Private NW** ApiPlacementNetwork: `Private`	Private NW	
3	**No** UseLoadBalancer: `No` LoadBalancerType は設定無効	**公開 NW** ApiPlacementNetwork: `Public`	Private NW	
4	**No** UseLoadBalancer: `No` LoadBalancerType は設定無効	**Private NW** ApiPlacementNetwork: `Private`	Private NW	

Note:

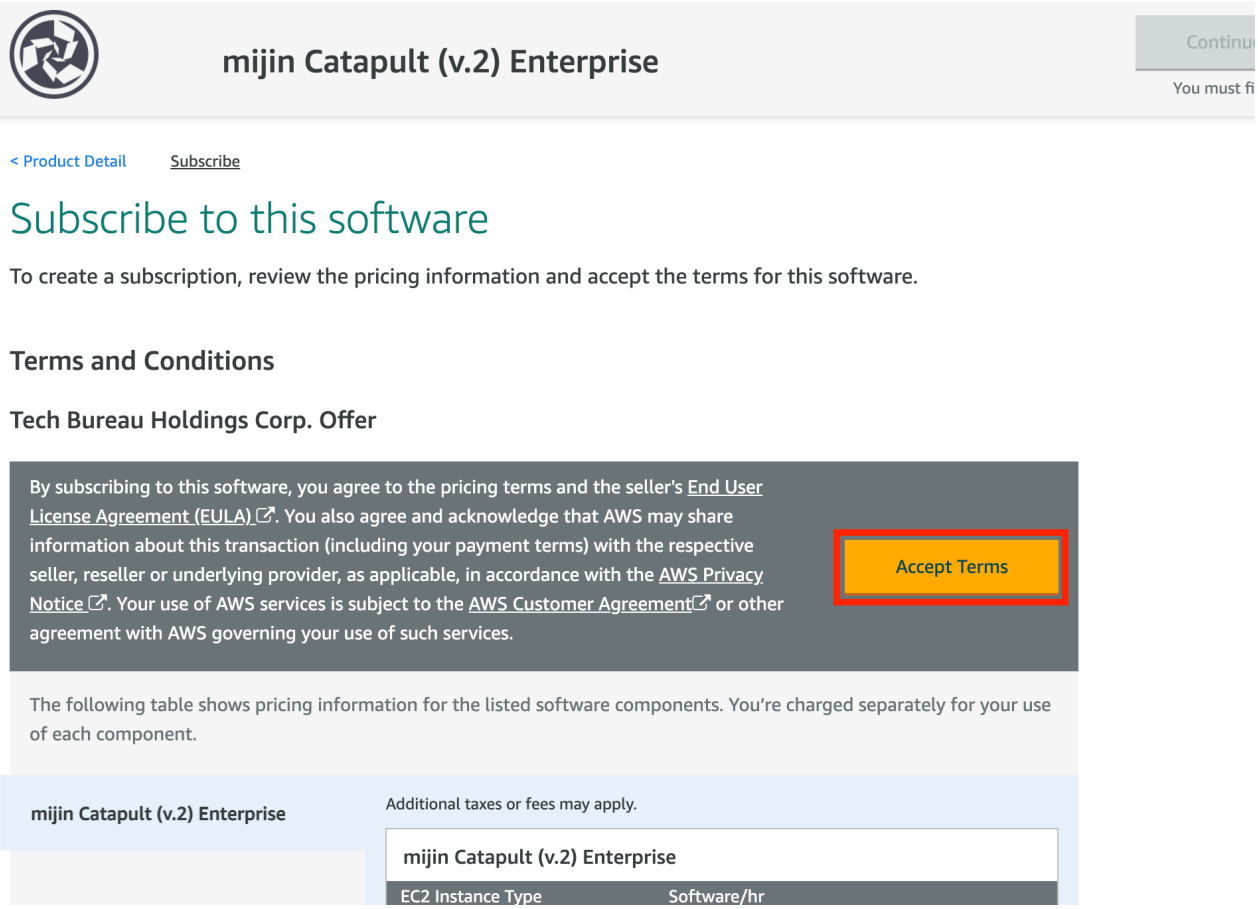
The commercial version cannot be configured with a single AZ to achieve high availability, but only with multiple AZs.
Note that you will always need two public networks and two private networks.
Deployment is single region, but can be deployed in each of the world's 20 regions.

2.2.2.3 Step.1



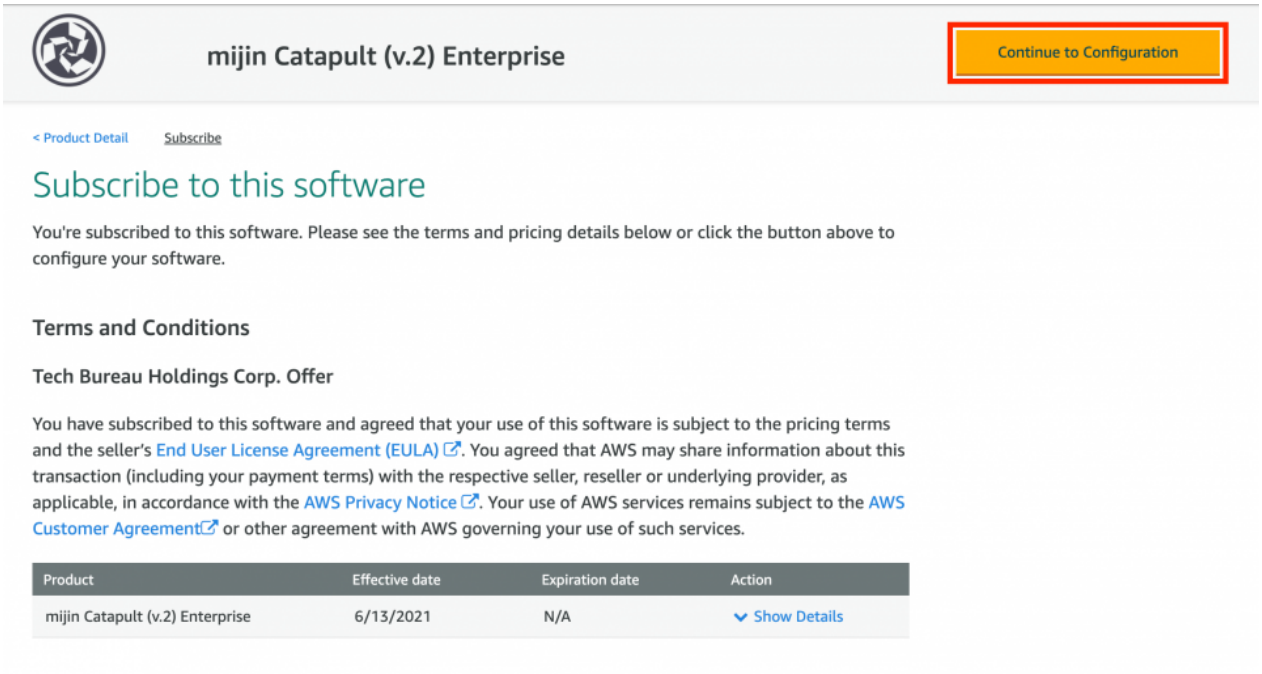
you need to subscribe to use the AMI of mijin Catapult Enterprise. Please press the button in the red frame.

2.2.2.4 Step.2



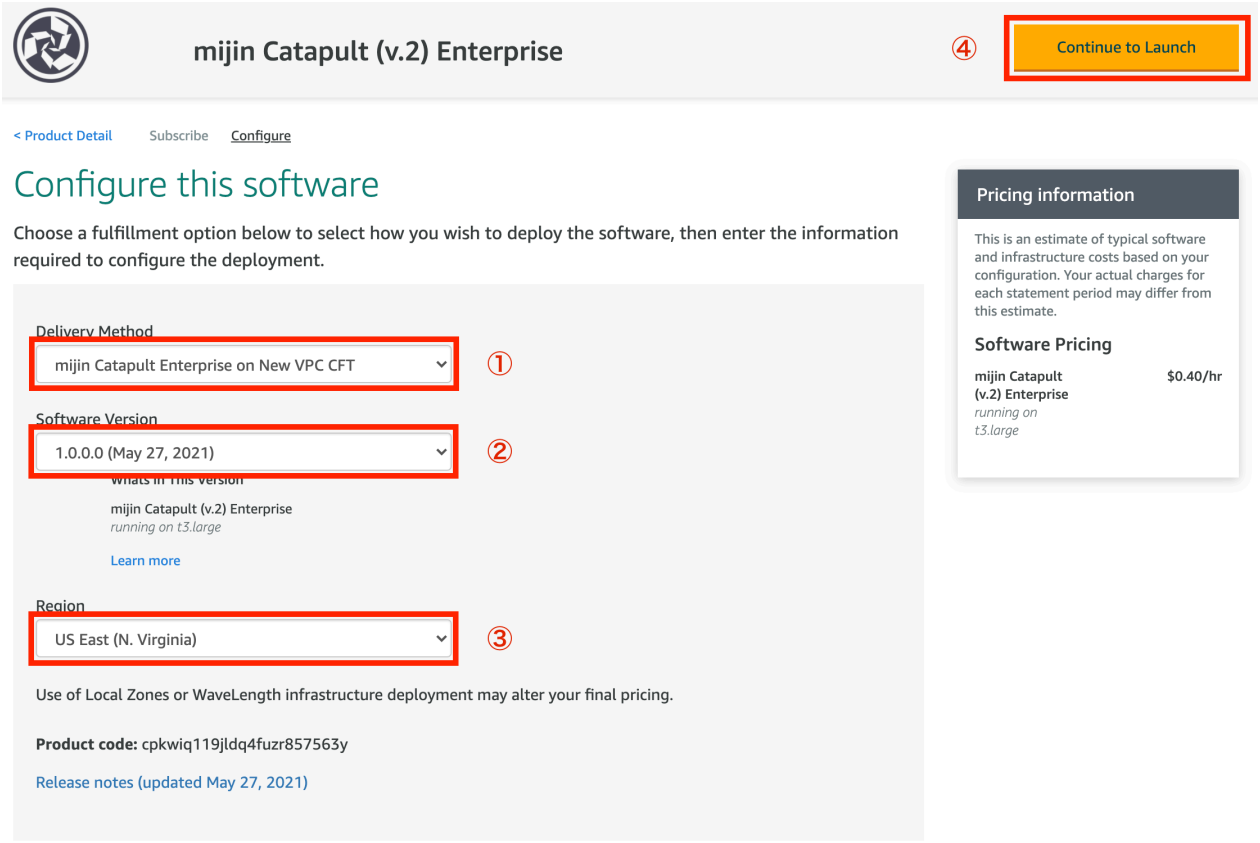
mijin Catapult Enterprise AMI, please approve its use.

2.2.2.5 Step.3



To start from Config, click on the text in the red box.

2.2.2.6 Step.4



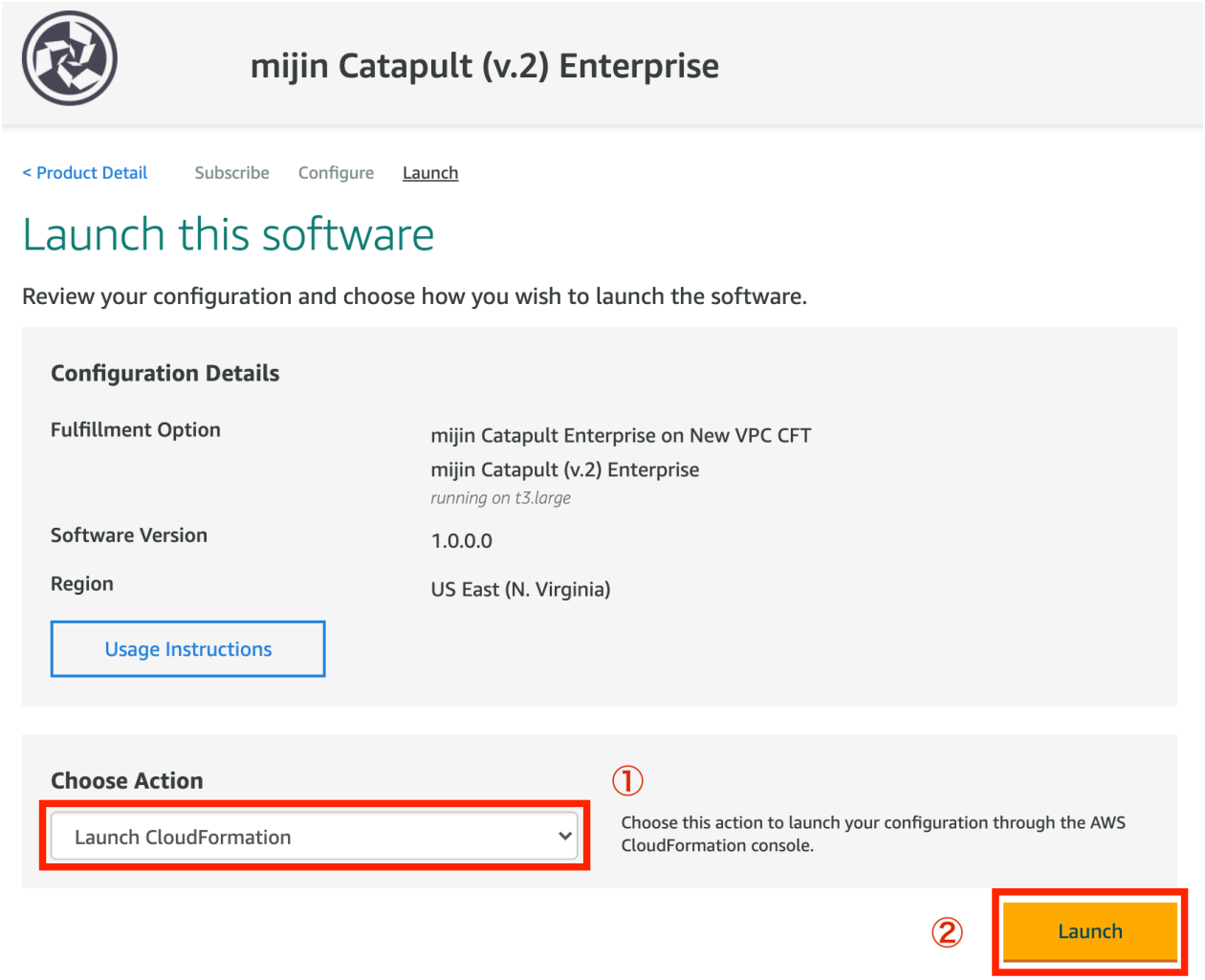
Select the template to be used in the red frame in (1). In this example, 'mijin Catapult Enterprise on New VPC CFT'is selected to create a new network.

Specify the version of mijin in the red frame in ②

Specify the region where mijin will be deployed

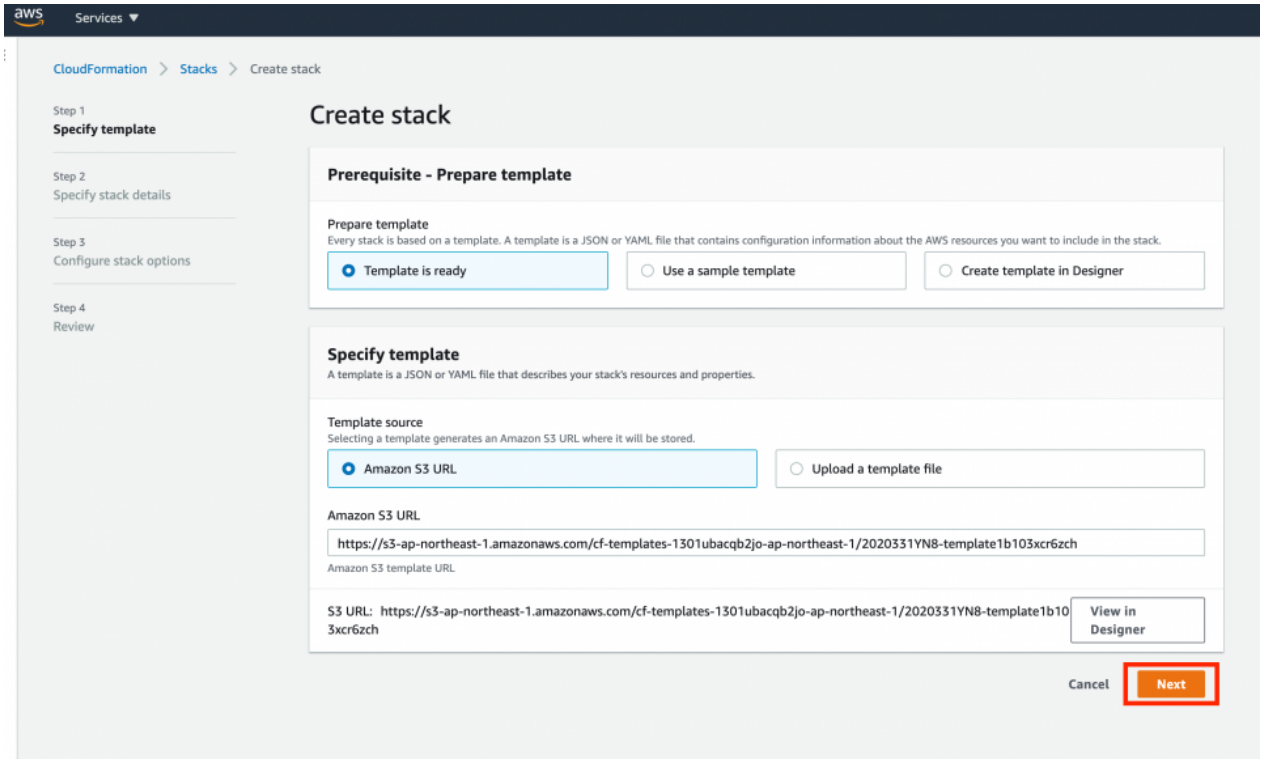
Click 'Continue to Launch'in the red frame of ④.

2.2.2.7 Step.5



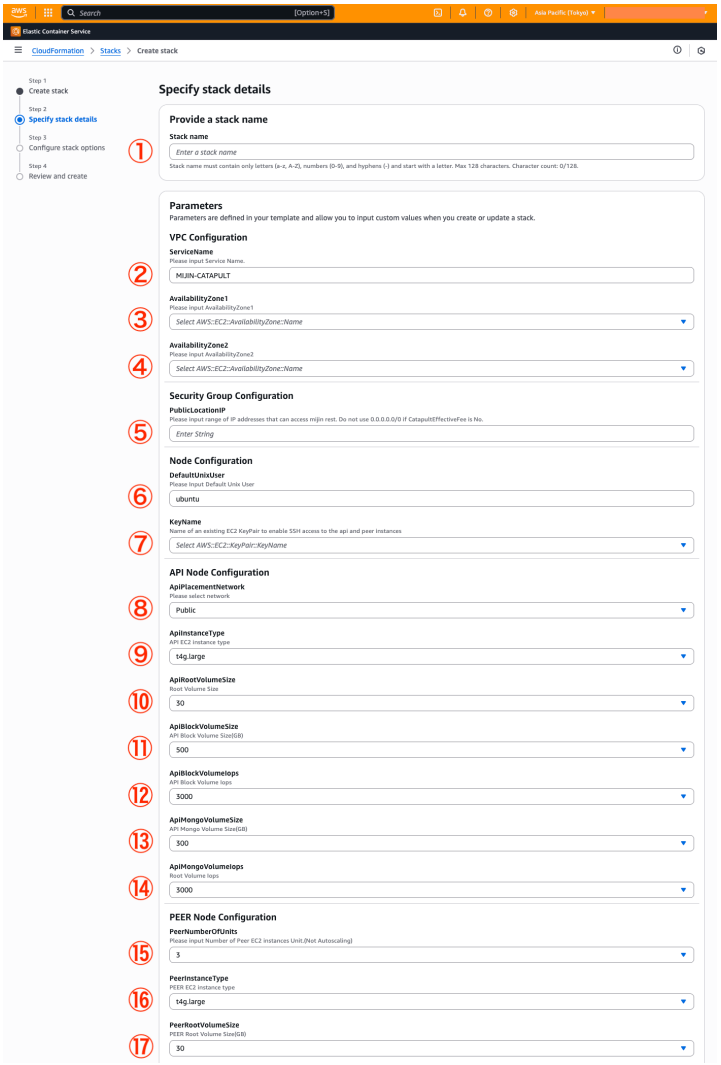
In the red frame in (1), specify the service for which mijin is to be built. In this case, 'Launch CloudFormation'is specified. Click 'Launch'in the red frame in ②.

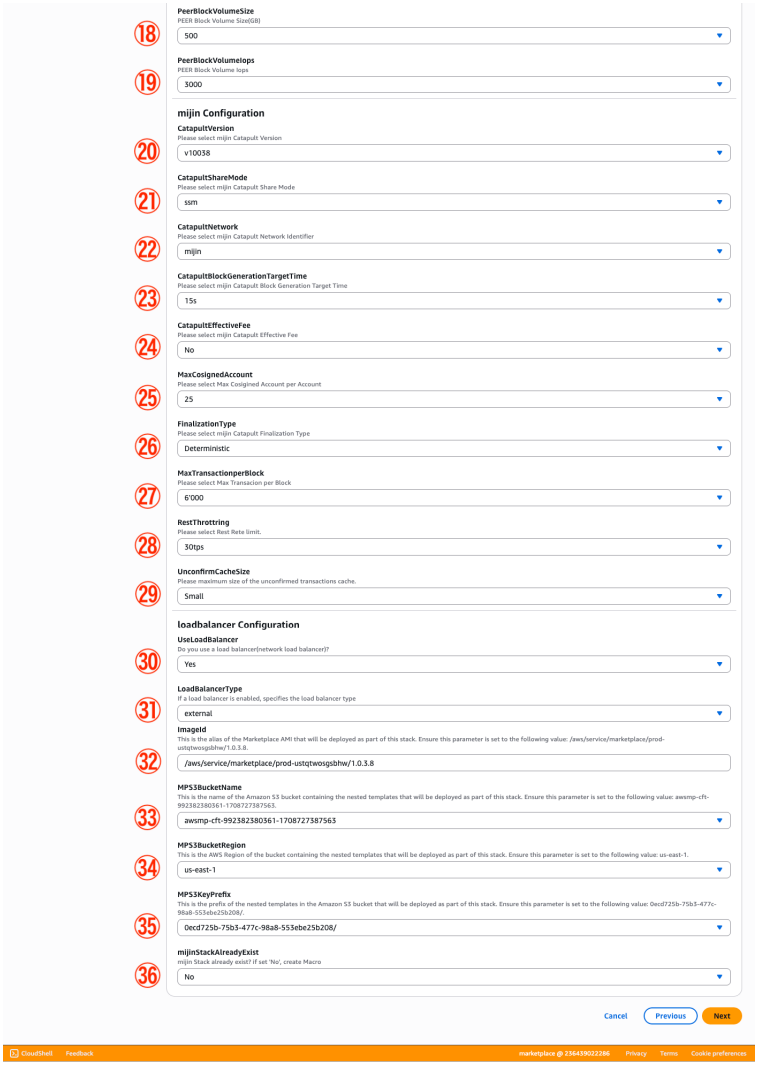
2.2.2.8 Step.6



Without editing anything in particular, press 'Next'in the red frame.

2.2.2.9 Step.7


Search
[Option+S]
Asia Pacific (Tokyo)
Elastic Container Service
CloudFormation
Stacks
Create stack
Step 1
Create stack
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review and create
Specify stack details
Provide a stack name
Stack name
Enter a stack name
Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 0/128.
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
VPC Configuration
ServiceName
Please input Service Name.
MIJIN-CATAPULT
AvailabilityZone1
Please input AvailabilityZone1
Select AWS::EC2::AvailabilityZone::Name
AvailabilityZone2
Please input AvailabilityZone2
Select AWS::EC2::AvailabilityZone::Name
Security Group Configuration
PublicLocationIP
Please input range of IP addresses that can access mijin rest. Do not use 0.0.0.0/0 if CatapultEffectiveFee is No.
Enter String
Node Configuration
DefaultUnixUser
Please Input Default Unix User
ubuntu
KeyName
Name of an existing EC2 KeyPair to enable SSH access to the api and peer instances
Select AWS::EC2::KeyPair::KeyName
API Node Configuration
ApiPlacementNetwork
Please select network
Public
ApiInstanceType
API EC2 instance type
t4g.large
ApiRootVolumeSize
Root Volume Size
30
ApiBlockVolumeSize
API Block Volume Size(GB)
500
ApiBlockVolumeIops
API Block Volume Iops
3000
ApiMongoVolumeSize
API Mongo Volume Size(GB)
300
ApiMongoVolumeIops
Root Volume Iops
3000
PEER Node Configuration
PeerNumberOfUnits
Please input Number of Peer EC2 instances Unit.(Not Autoscaling)
3
PeerInstanceType
PEER EC2 instance type
t4g.large
PeerRootVolumeSize
PEER Root Volume Size(GB)
30
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
⑭
⑮
⑯
⑰



Enter parameters.

Table 5: mijin デプロイ用パラメーター覧（簡易版）

No	パラメータ	Description	Recommended value
①	Stack Name	Name of this stack	•
②	Service Name	全リソースの冠名として利用されるサービス名	•
③	Availability Zone1	Availability zone to be used (Multi-AZ configuration)	•
④	Availability Zone2	Specify an AZ different from AZ1 (Multi-AZ configuration)	•
⑤	Public Location IP	Allowed IP address for API connection (/24, etc. is also acceptable)	例: XX.XX.XX.XX/32
⑥	Default UnixUser	Standard Unix user for EC2	ubuntu
⑦	KeyName	Key name for EC2 SSH connection	•
⑧	ApiPlacementNetwork	Network placement location of API node	Public
⑨	ApiInstanceType	Instance type of API node	c5n.2xlarge 以上
⑩	ApiRootVolumeSize	Root disk space for API node (used for Docker, etc.)	30GB or more
⑪	ApiBlockVolumeSize	Disk space for mijin block data storage	500GB or more
⑫	ApiBlockVolumeIops	IOPS settings for the above disks	3000 以上
⑬	ApiMongoVolumeSize	Disk space for mongo data storage	300GB or more

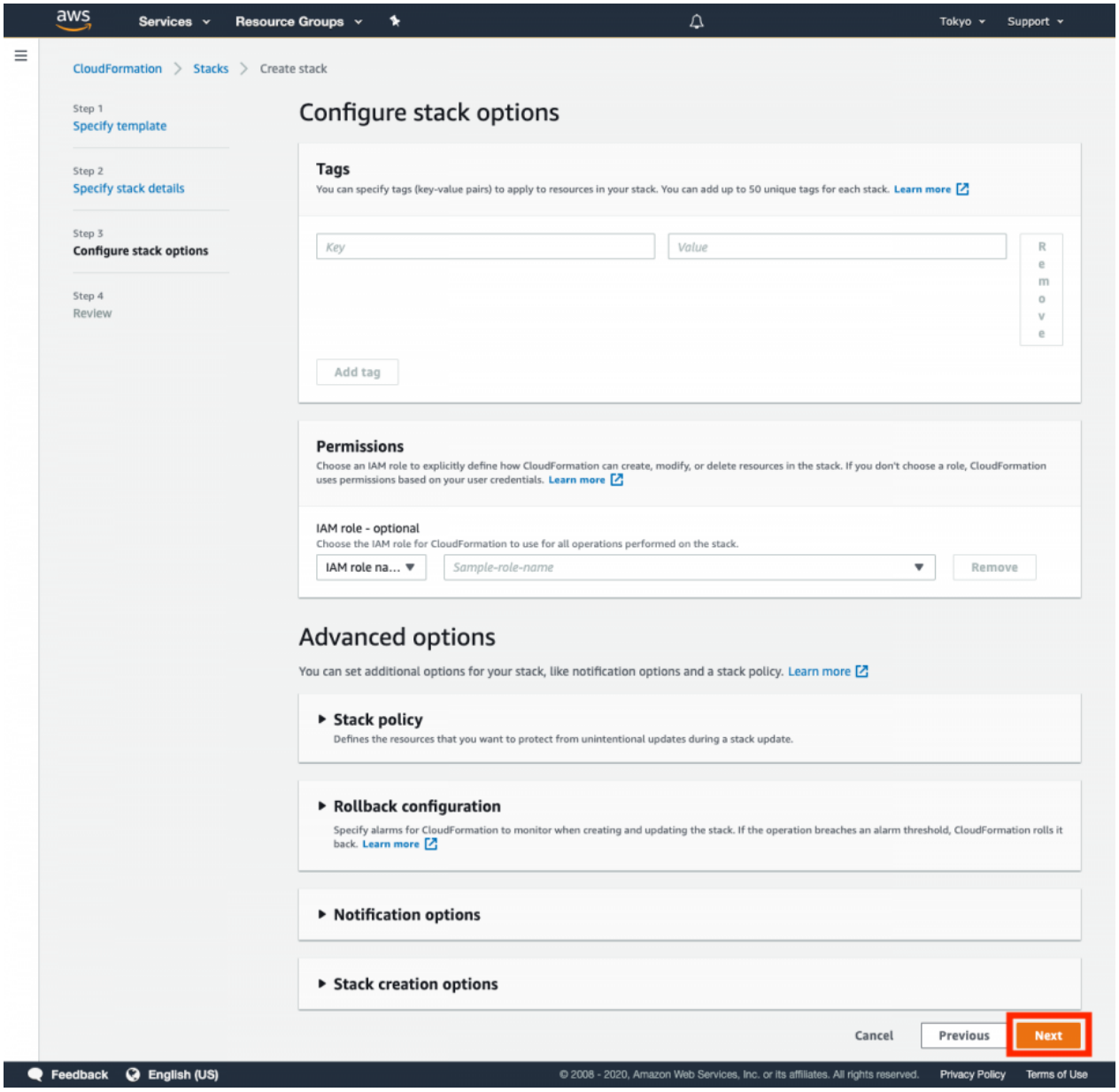
continues on next page

Table 5 – continued from previous page

No	パラメータ	Description	Recommended value
⑭	ApiMongoVolumeIops	IOPS settings for mongo disks	3000 以上
⑮	PeerNumberOfUnits	Number of PEER nodes	3
⑯	PeerInstanceType	Instance type of PEER node	c5n.xlarge 以上
⑰	PeerRootVolumeSize	Root disk space of PEER node	30GB or more
⑱	PeerBlockVolumeSize	Disk for block data in PEER node	500GB or more
⑲	PeerBlockVolumeIops	IOPS settings for the above disks	3000 以上
⑳	CatapultVersion	mijin のバージョン	v10038
㉑	CatapultShareMode	Setting the save method (example: SSM)	ssm
㉒	CatapultNetwork	network name	mijin
㉓	BlockGenerationTargetTime	block generation interval	15s
㉔	EffectiveFee	Commission setting (with/without)	No
㉕	MaxCosignedAccount	Maximum number of accounts that can cosign	25
㉖	FinalizationType	finalization system	Deterministic
㉗	MaxTransactionperBlock	Maximum number of transactions (1 block)	6000
㉘	RestThrottling	API connection limit	30tps
㉙	UnconfirmCacheSize	Unauthorized transaction cache size	•
㉚	UseLoadBalancer	Whether NLB is used or not	Yes
㉛	LoadBalancerType	ロードバランサー種別	External
㉜	ImageId	ID of AMI for management (cannot be changed)	Unchangeable
㉝	MPS3BucketName	S3 bucket name (cannot be changed)	Unchangeable
㉞	MPS3BucketRegion	S3 Region (cannot be changed)	Unchangeable
㉟	MPS3KeyPrefix	S3 prefix (cannot be changed)	Unchangeable
㊱	mijinStackAlreadyExist	Existing stack or not	No

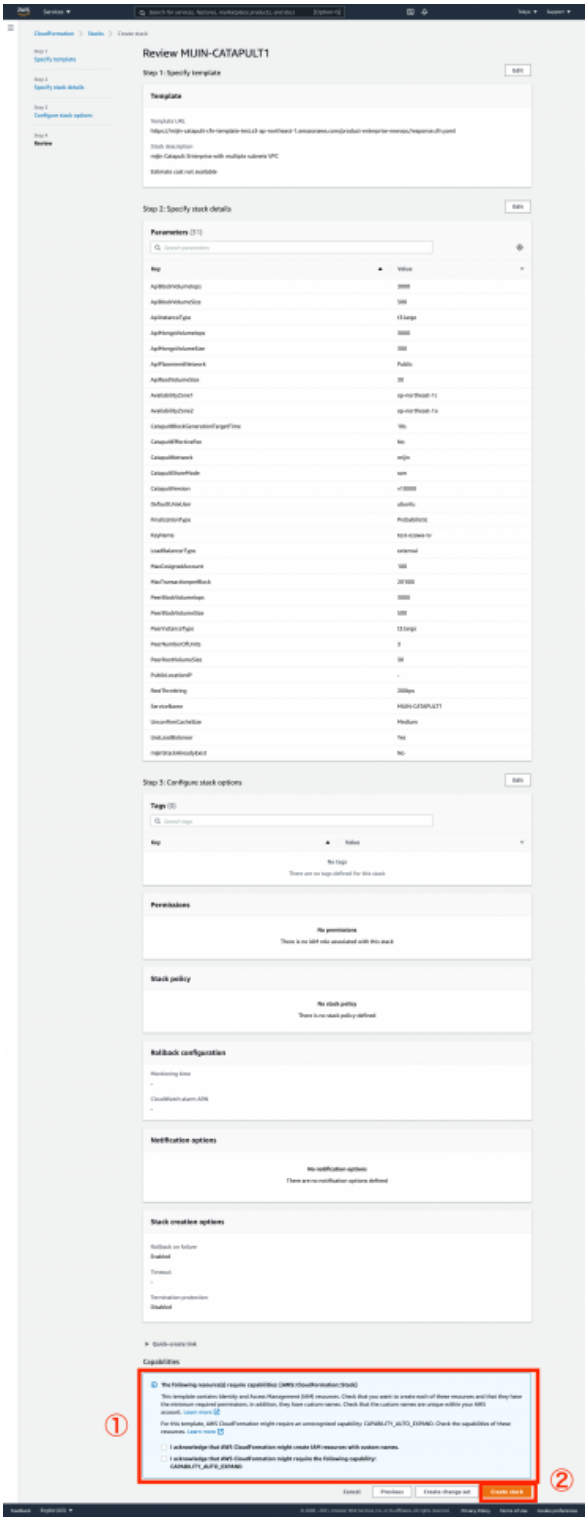
Press ‘Next’after completing the parameter entry.

2.2.2.10 Step.8



Without editing anything in particular, press 'Next'in the red frame.

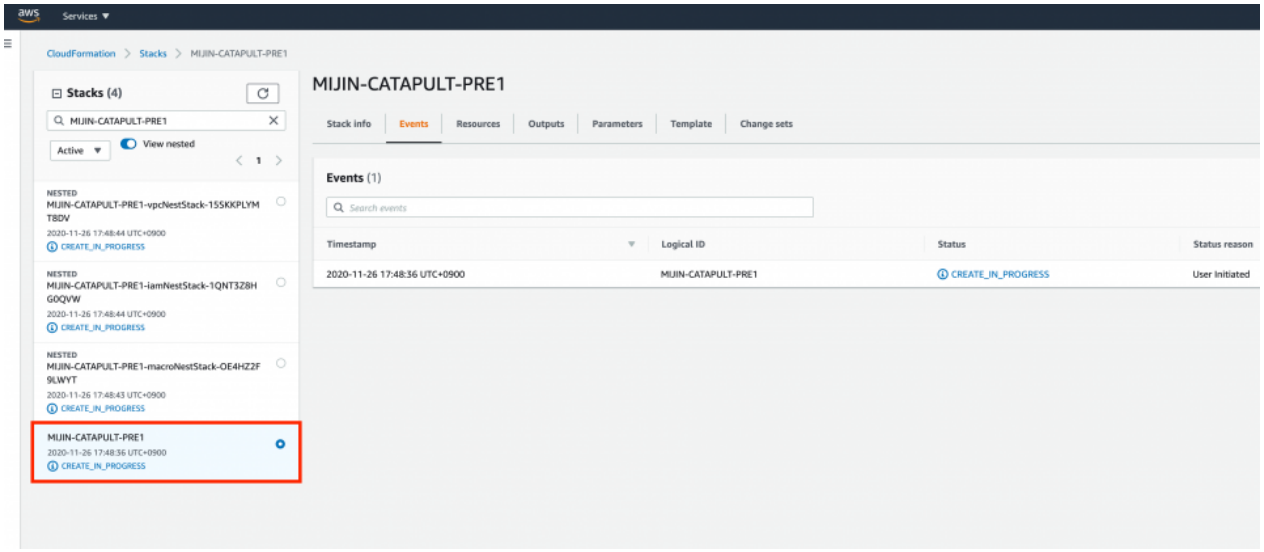
2.2.2.11 Step.9



Check the two items in the red frame in ①.

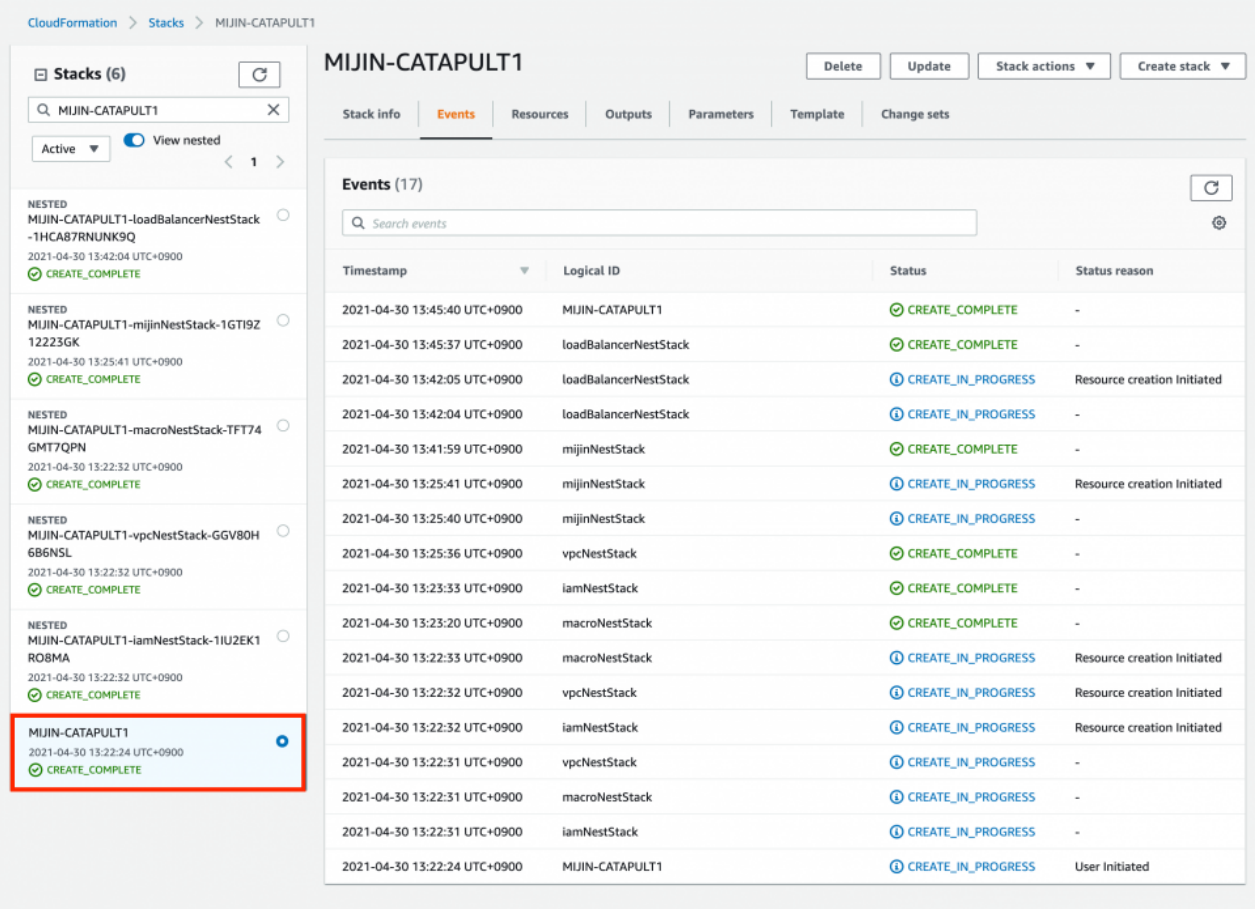
Click ‘Create Stack’in the red frame in ②. If there are no errors, creation will begin.

2.2.2.12 Step.10



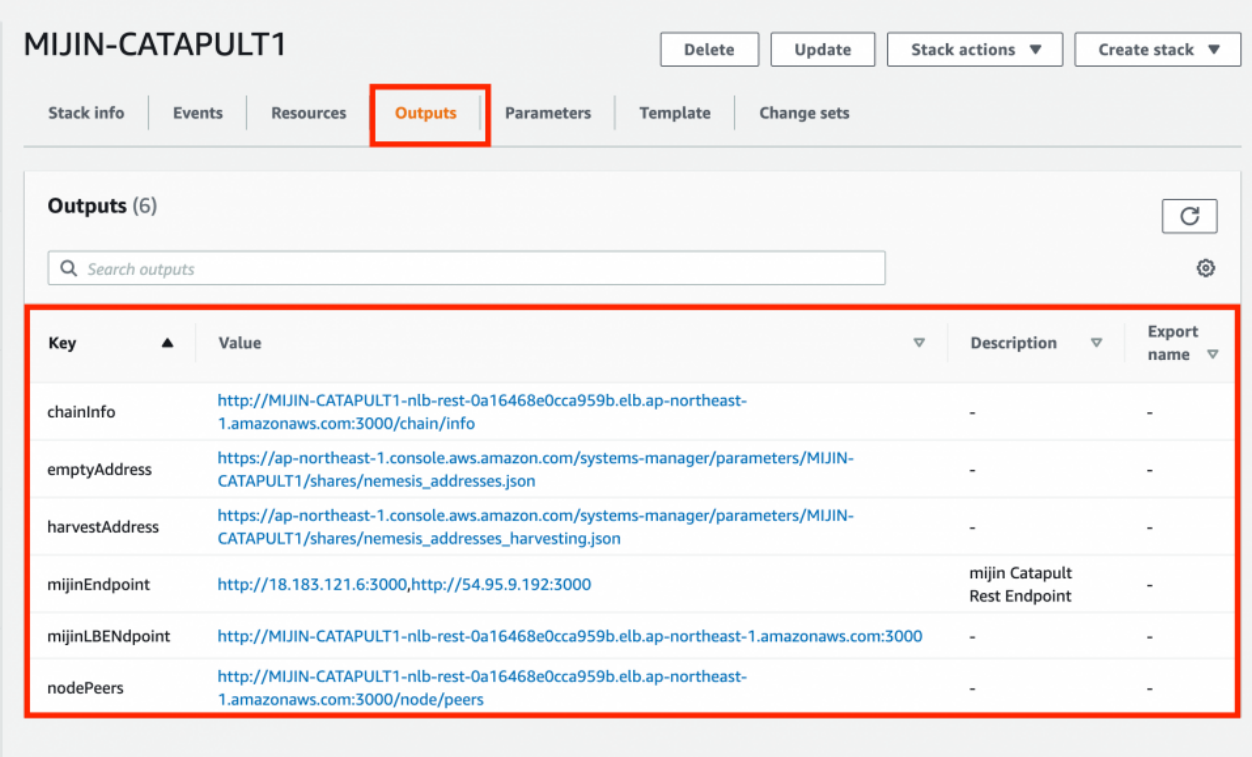
Confirm that Stack has started and is in 'CREATE_IN_PROGRESS'state. This state will take approximately 20~30 minutes.

2.2.2.13 Step.11



If the status is 'CREATE_COMPLATE', the creation of mijin is complete.

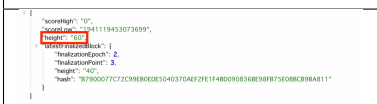
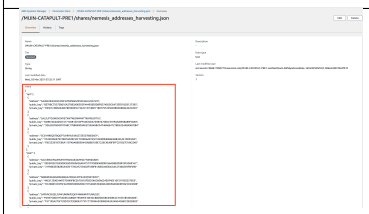
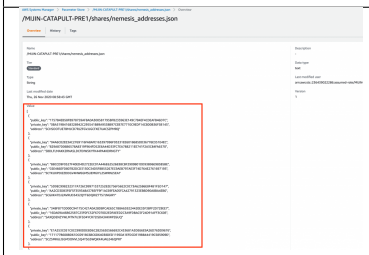
2.2.2.14 Step.12



Press ‘Outputs’of the created Stack to see the created mijin configuration information.

2.2.2.15 mijin endpoints and checkpoints

Table 6: mijin endpoints and checkpoints

	mijinLBEndpoint ロードバランサー経由の mijin API エンドポイントです。セッション維持設定が有効で、ソース IP に基づいたスティッキーセッションとなります。詳細はこちら
	mijinEndpoint mijin API endpoints for direct access to API nodes (EC2).
	**chainInfo** mijin の現在のブロック数を確認できます。ブロック数が「2」以上であれば正常です。
	**harvestAddress** AWS Systems Manager パラメータストアに登録された通貨分配用アドレスのリンクです。
	**emptyAddress** AWS Systems Manager パラメータストアに登録された未使用アドレスのリンクです。
	**nodePeers** mijin API からノードの接続状態が確認できます。API ノード 1 台と、設定済みの PEER 台数が表示されていれば OK です。

You are now ready to use mijin Catapult. Let's start the operation in the next section!

2.2.3 Deploy mijin on an existing VPC

This chapter shows how to deploy mijin Catapult(v.2) on an existing network (VPC).
In the commercial version, you can flexibly build a network that fits your environment by changing parameters.

2.2.3.1 List of services to be built on AWS by deployment

- Amazon EC2 (APINode x 2 PEERNode x 3~)
- Amazon EBS
- Elastic Load Balancing
- Amazon Route53
- parameter store

Note:

Elastic Load Balancing may not be created in some cases, depending on the parameter settings specified during deployment.

2.2.3.2 Creation of subnets for existing VPCs

In deploying mijin, two subnets, one for the Public network and one for the Private network, are required to achieve high availability.
If the number of subnets is not sufficient, please refer to the following for creation.

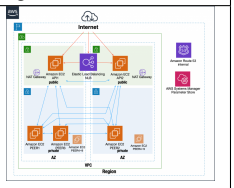
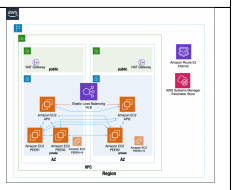
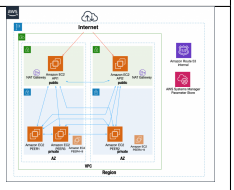
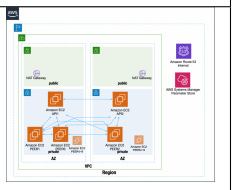
<https://docs.aws.amazon.com/ja_jp/vpc/latest/userguide/working-with-subnets.html#create-subnets>

Note: Create two subnets of multiple availability zones (AZs) so that service can continue even if one AZ fails.

2.2.3.3 View Network

For existing networks, it is possible to build on an already existing environment by changing parameters.
An example pattern is shown below.

Table 7: API/PEER ネットワーク構成一覧

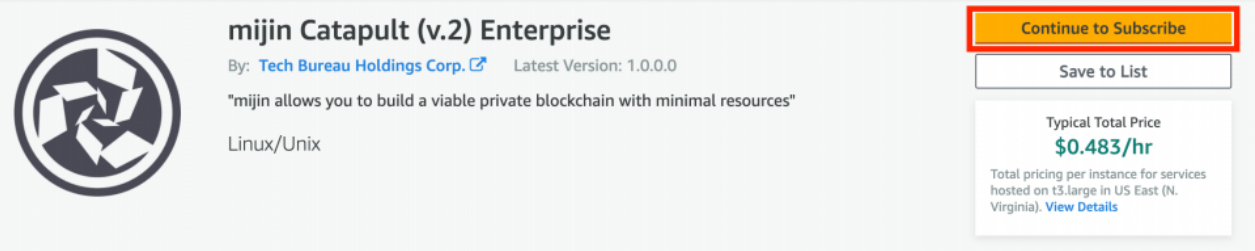
No	load balancer	API node placement	PEER node placement	figure
1	**あり（公開ネットワーク）** VPC: Your vpc VpcCidrBlock: xx.xx.xx.xx/xx Public1/2, Private1/2 UseLoadBalancer: `Yes` LoadBalancerType: `external`	public NW ApiPlacementNetwork: `Public`	Private NW	
2	**あり（非公開ネットワーク）** UseLoadBalancer: `Yes` LoadBalancerType: `internal`	Private NW ApiPlacementNetwork: `Private`	Private NW	
3	**No** UseLoadBalancer: `No` LoadBalancerType は設定効果なし	public NW ApiPlacementNetwork: `Public`	Private NW	
4	**No** UseLoadBalancer: `No` LoadBalancerType は設定効果なし	Private NW ApiPlacementNetwork: `Private`	Private NW	

Note:

The commercial version cannot be configured with a single AZ to achieve high availability, but only with multiple AZs.
Note that you will always need two public networks and two private networks.

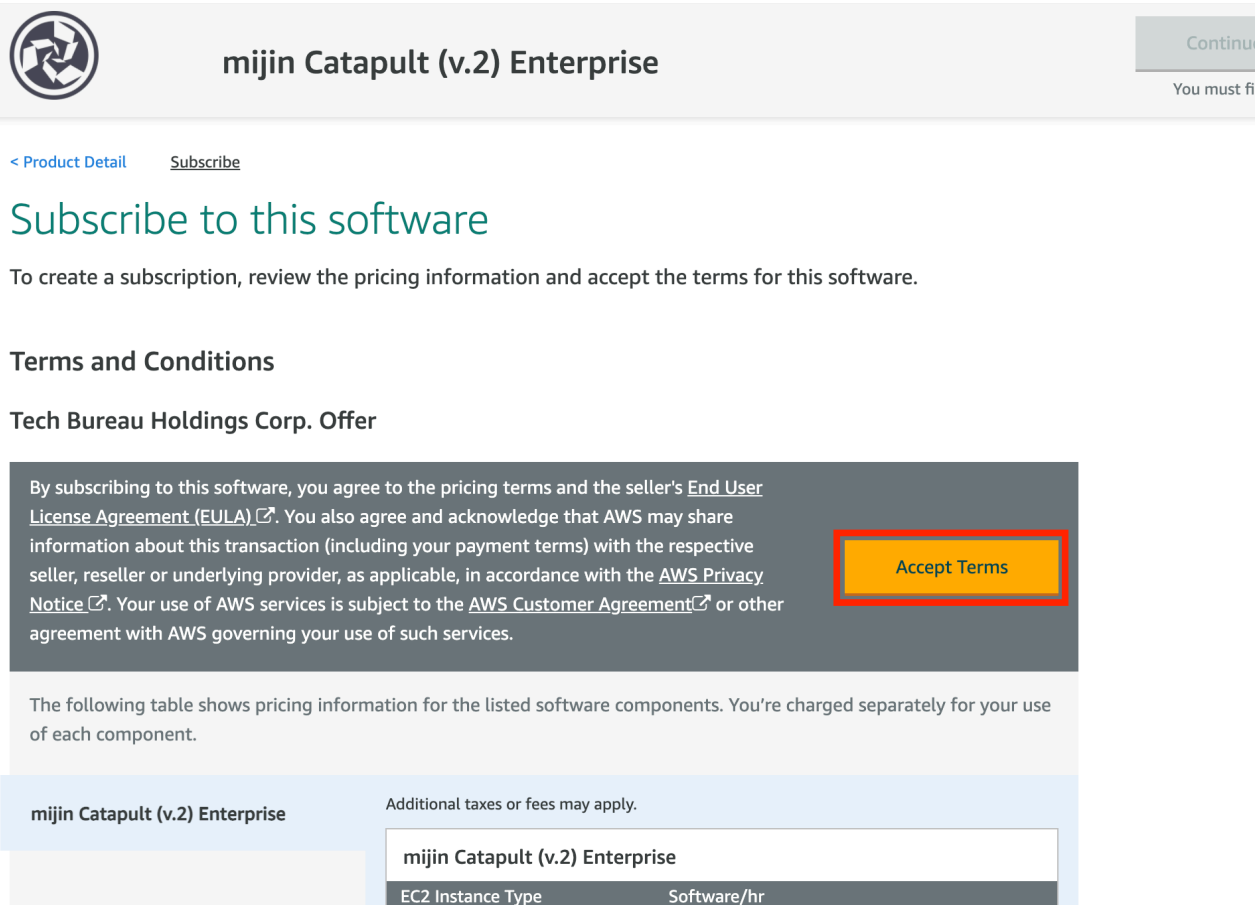
Deployment is single region, but can be deployed in each of the world's 21 regions.

2.2.3.4 Step.1



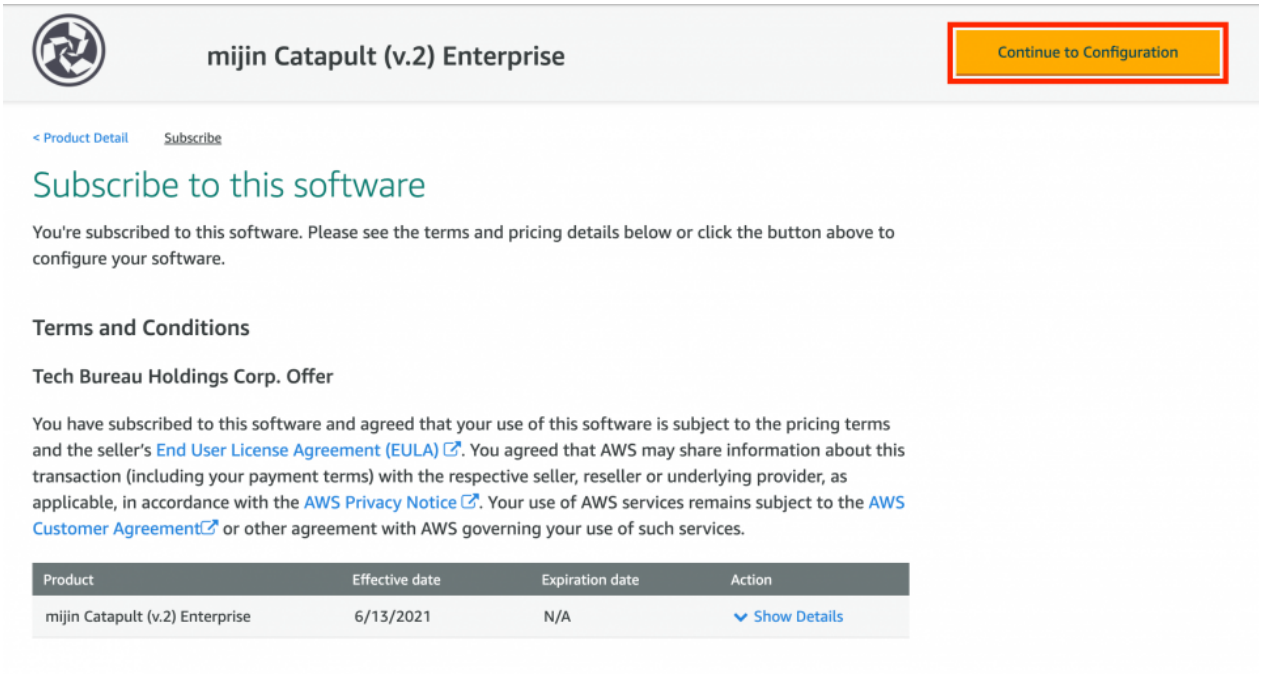
you need to subscribe to use the AMI of mijin Catapult Enterprise. Please press the button in the red frame.

2.2.3.5 Step.2



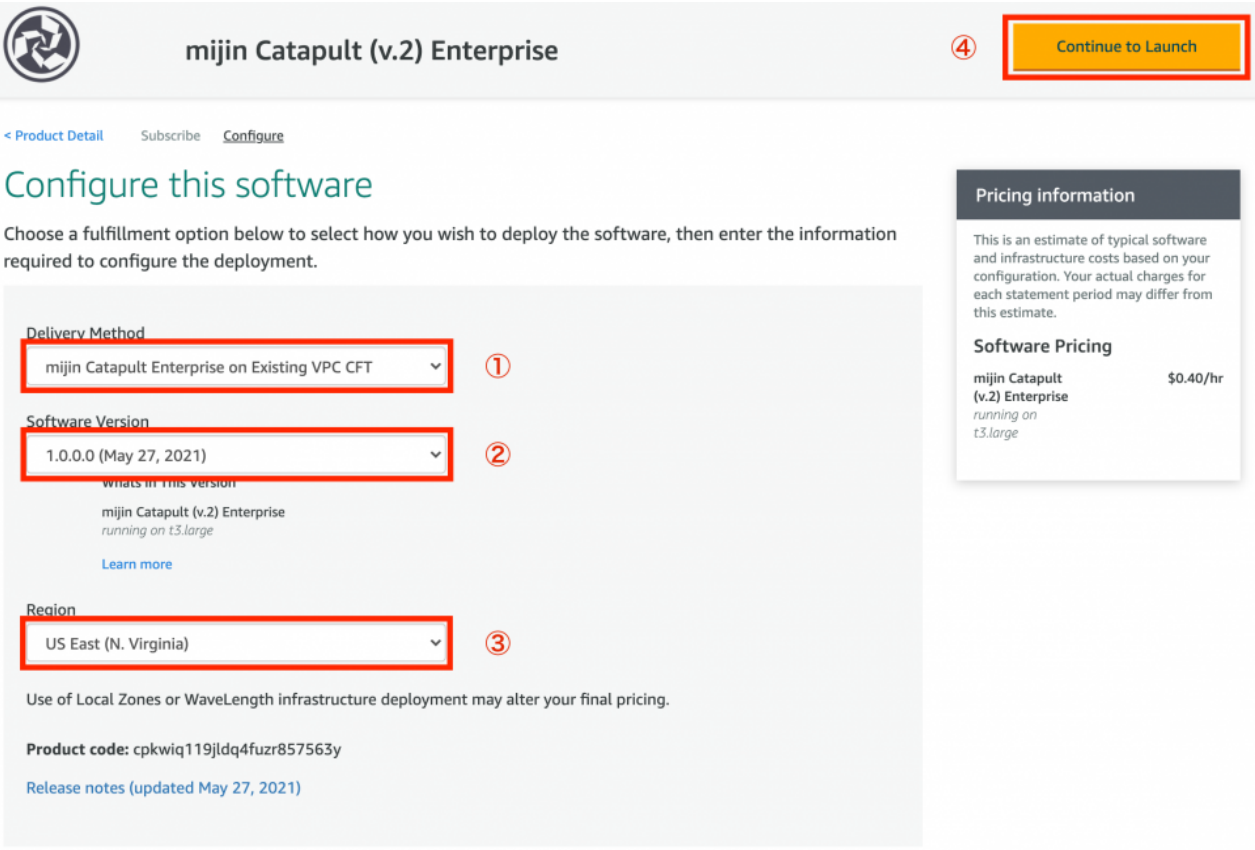
mijin Catapult Enterprise AMI, please approve its use.

2.2.3.6 Step.3



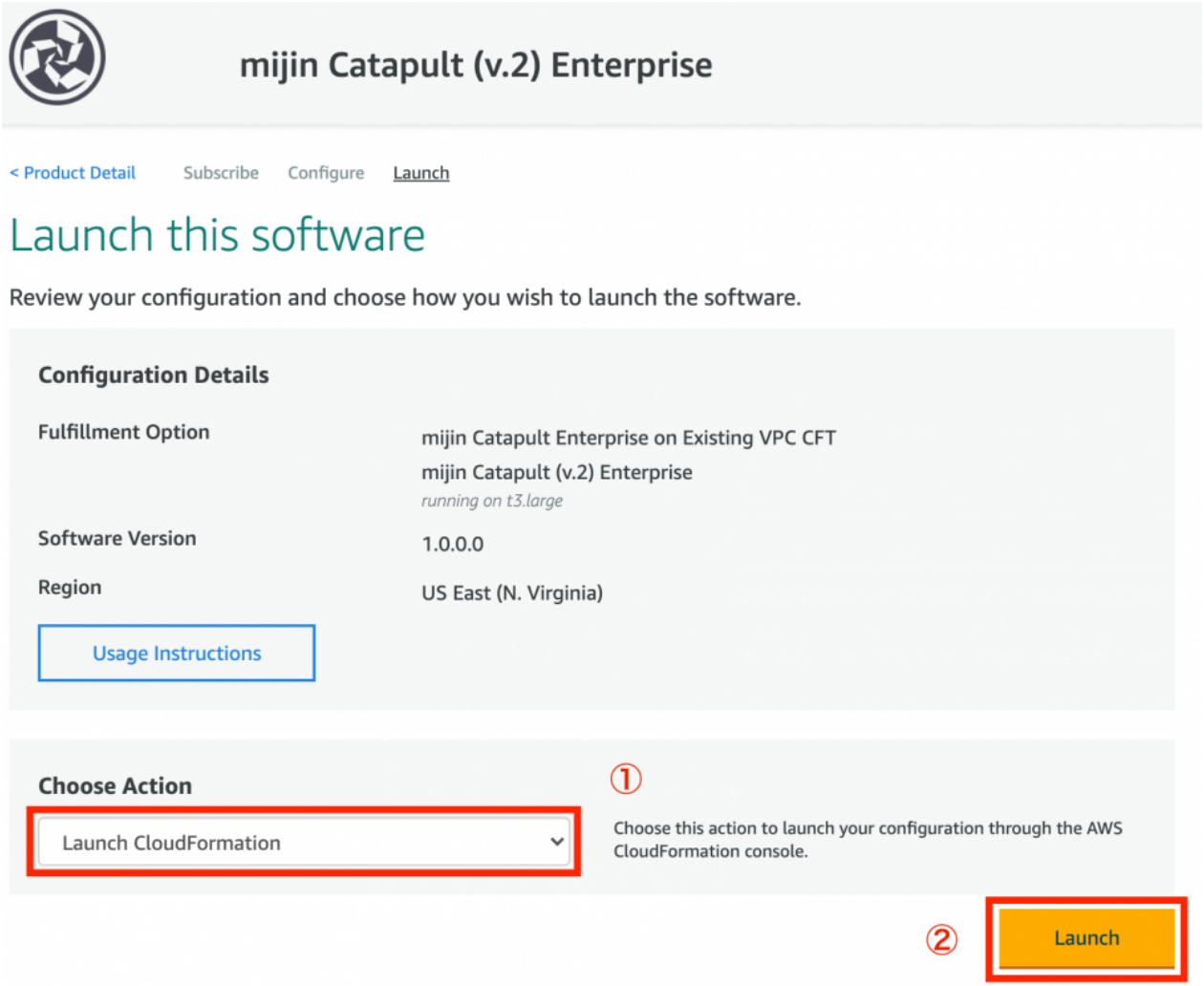
To start from Config, click on the text in the red box.

2.2.3.7 Step.4



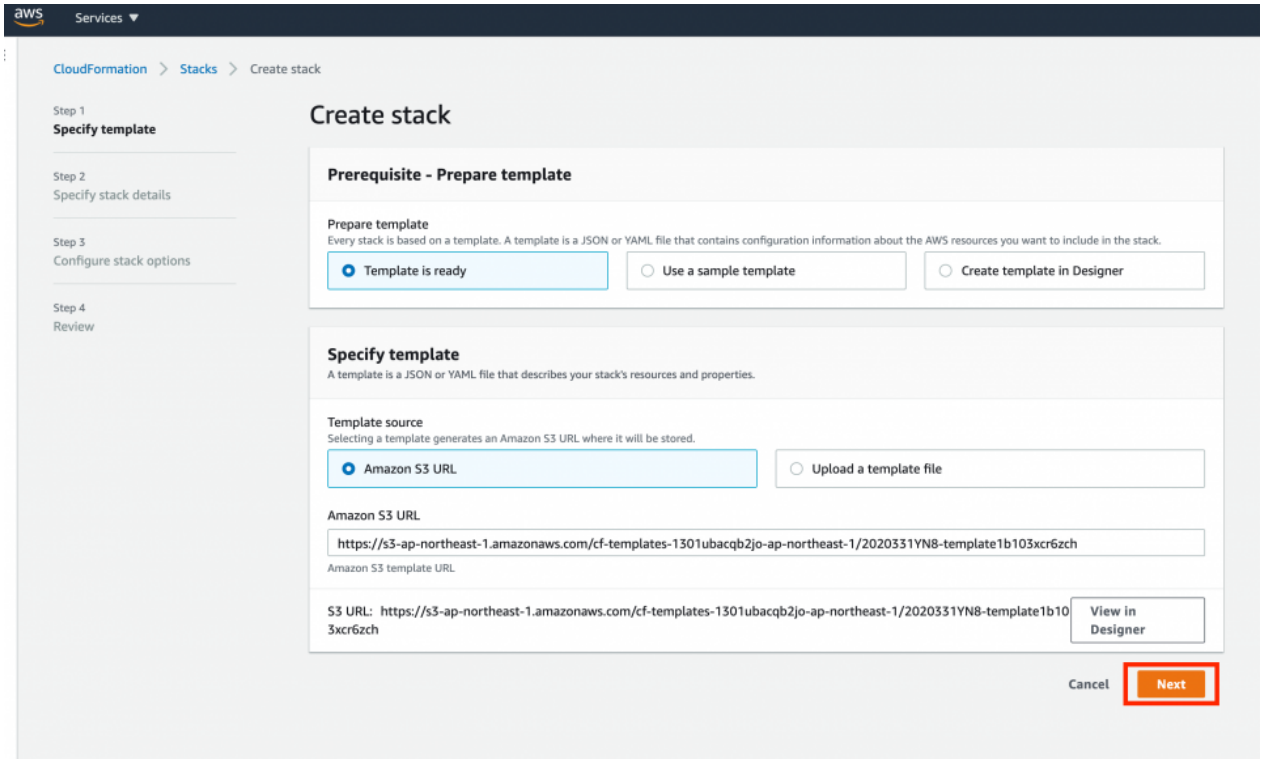
Select the template to be used in the red frame in (1). In this case, we specify 'mijin Catapult Enterprise on Existing VPC CFT'because it will be created on an existing network. Specify the version of mijin in the red frame in ②. Specify the region where mijin will be deployed. Click 'Continue to Launch'in the red frame of ④.

2.2.3.8 Step.5



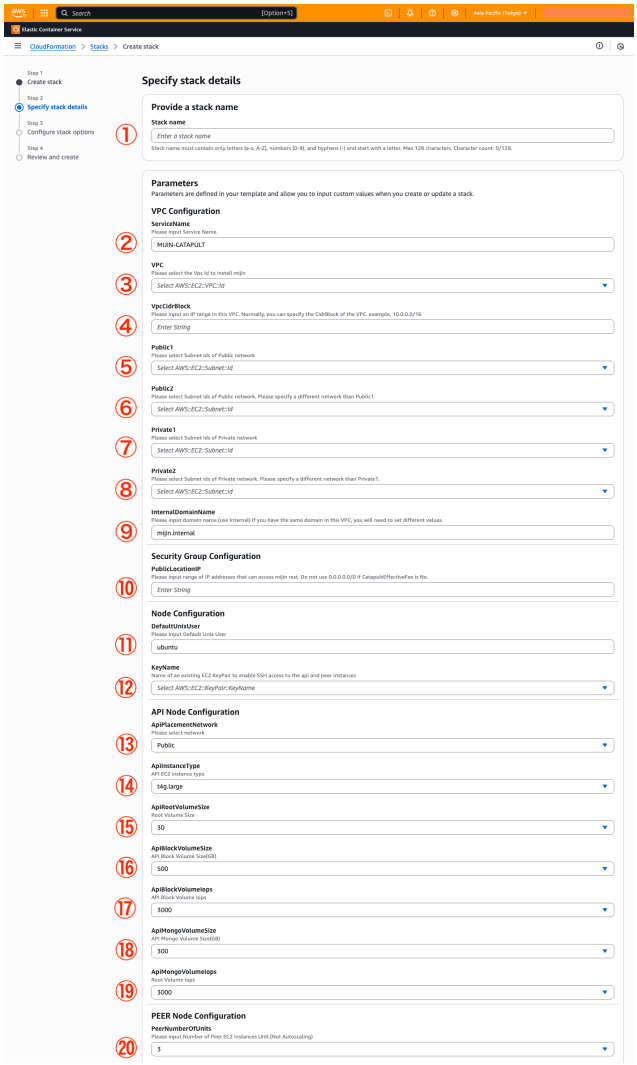
In the red frame in (1), specify the service for which mijin is to be built. In this case, 'Launch CloudFormation'is specified. Click 'Launch'in the red frame in ②.

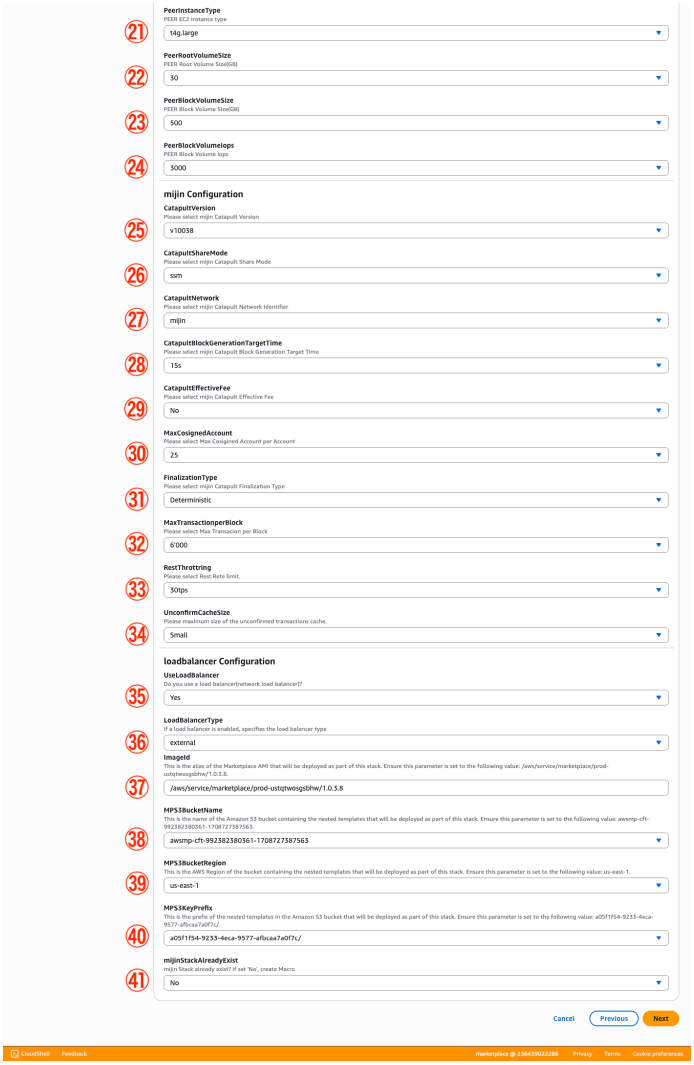
2.2.3.9 Step.6



Without editing anything in particular, press 'Next'in the red frame.

2.2.3.10 Step.7


Search
[Option+S]
Asia Pacific (Tokyo)
Elastic Container Service
CloudFormation > Stacks > Create stack
Step 1
Create stack
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review and create
Specify stack details
Provide a stack name
Stack name
Enter a stack name
Stack name must contain only letters (a–z, A–Z), numbers (0–9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 0/128.
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
VPC Configuration
ServiceName
Please input Service Name.
MIJIN-CATAPULT
VPC
Please select the Vpc Id to install mijin
Select AWS::EC2::VPC::Id
VpcCidrBlock
Please input an IP range in this VPC. Normally, you can specify the CidrBlock of the VPC. example, 10.0.0.0/16
Enter String
Public1
Please select Subnet ids of Public network
Select AWS::EC2::Subnet::Id
Public2
Please select Subnet ids of Public network. Please specify a different network than Public1.
Select AWS::EC2::Subnet::Id
Private1
Please select Subnet ids of Private network
Select AWS::EC2::Subnet::Id
Private2
Please select Subnet ids of Private network. Please specify a different network than Private1.
Select AWS::EC2::Subnet::Id
InternalDomainName
Please input domain name.(use Internal) If you have the same domain in this VPC, you will need to set different values.
mijin.internal
Security Group Configuration
PublicLocationIP
Please input range of IP addresses that can access mijin rest. Do not use 0.0.0.0/0 if CatapultEffectiveFee is No.
Enter String
Node Configuration
DefaultUnixUser
Please input Default Unix User
ubuntu
KeyName
Name of an existing EC2 KeyPair to enable SSH access to the api and peer instances
Select AWS::EC2::KeyPair::KeyName
API Node Configuration
ApiPlacementNetwork
Please select network
Public
ApiInstanceType
API EC2 instance type
t4g.large
ApiRootVolumeSize
Root Volume Size
30
ApiBlockVolumeSize
API Block Volume Size(GB)
500
ApiBlockVolumeIops
API Block Volume Iops
3000
ApiMongoVolumeSize
API Mongo Volume Size(GB)
300
ApiMongoVolumeIops
Root Volume Iops
3000
PEER Node Configuration
PeerNumberOfUnits
Please input Number of Peer EC2 instances Unit.(Not Autoscaling)
3
①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
⑬
⑭
⑮
⑯
⑰
⑱
⑲
⑳



Enter parameters.

Table 8: CloudFormation Parameter List

No	Parameter	Describe	Recommended value
①	Stack Name	Provide the name in this stack.	•
②	Service Name	Provide the name of the service created by the stack. Used as a crown name for all resources.	•
③	VPC	Please specify your VPC.	•
④	VpcCidrBlock	Please list the IP address range of the VPC specified in (3). (CIDR Block) The IP address range in () selected in (3) is fine.	•
⑤	Public1	Specify the public network in the VPC specified in (3).	•

continues on next page

Table 8 – continued from previous page

No	Parameter	Describe	Recommended value
⑥	Public2	Specify the public network in the VPC specified in (3). ⑤ と別のネットワークを指定する必要があります。	•
⑦	Private1	Specify the private network in the VPC specified in (3).	•
⑧	Private2	Specify the private network in the VPC specified in (3). ⑦ とは別のネットワークを指定する必要があります。	•
⑨	InternalDomainName	List the DNS name for name resolution to be used between nodes. 非公開ネットワーク用で、公開はされません。 複数スタック構築時には一意にしてください。	mijin.internal
⑩	Public Location IP	Specify the IP address that is allowed to connect to mijin Catapult's API. IP レンジ指定可 (/24 など)	XX.XX.XX.XX/32 etc.
⑪	Default UnixUser	作成する EC2 の標準 Unix ユーザー	ubuntu
⑫	KeyName	EC2 の SSH 鍵を指定してください。 表示されない場合は事前作成が必要です。 こちら	•
⑬	ApiPlacementNetwork	API ノードの配置ネットワークを選択してください。	•
⑭	ApiInstanceType	API ノードのインスタンスタイプ Architecture and Specification Requirements 参照	c5n.2xlarge 以上
⑮	ApiRootVolumeSize	API ノードのルートディスクサイズ (GB) Docker ログや System log に使用	30GB or more
⑯	ApiBlockVolumeSize	mijin ブロックデータ格納用ディスクサイズ (GB) Use gp3 disk	500GB or more
⑰	ApiBlockVolumeIops	IOPS for mijin block data	Over 3000
⑱	ApiMongoVolumeSize	Disk size for mongo data storage (GB) For Blockchain data call	300GB or more
⑲	ApiMongoVolumeIops	IOPS for mongo data	3000 IOPS or more
⑳	PeerNumberOfUnits	PEER ノードの台数	3 or more
㉑	PeerInstanceType	Instance type of PEER node Architecture and Specification Requirements 参照	c5n.xlarge 以上
㉒	PeerRootVolumeSize	PEER node root disk size (GB) Docker ログや System log に使用	30GB or more
㉓	PeerBlockVolumeSize	Disk size for PEER node block data storage	500GB or more
㉔	PeerBlockVolumeIops	IOPS for block data in PEER node	3000 IOPS or more
㉕	CatapultVersion	mijin のバージョン	v10038

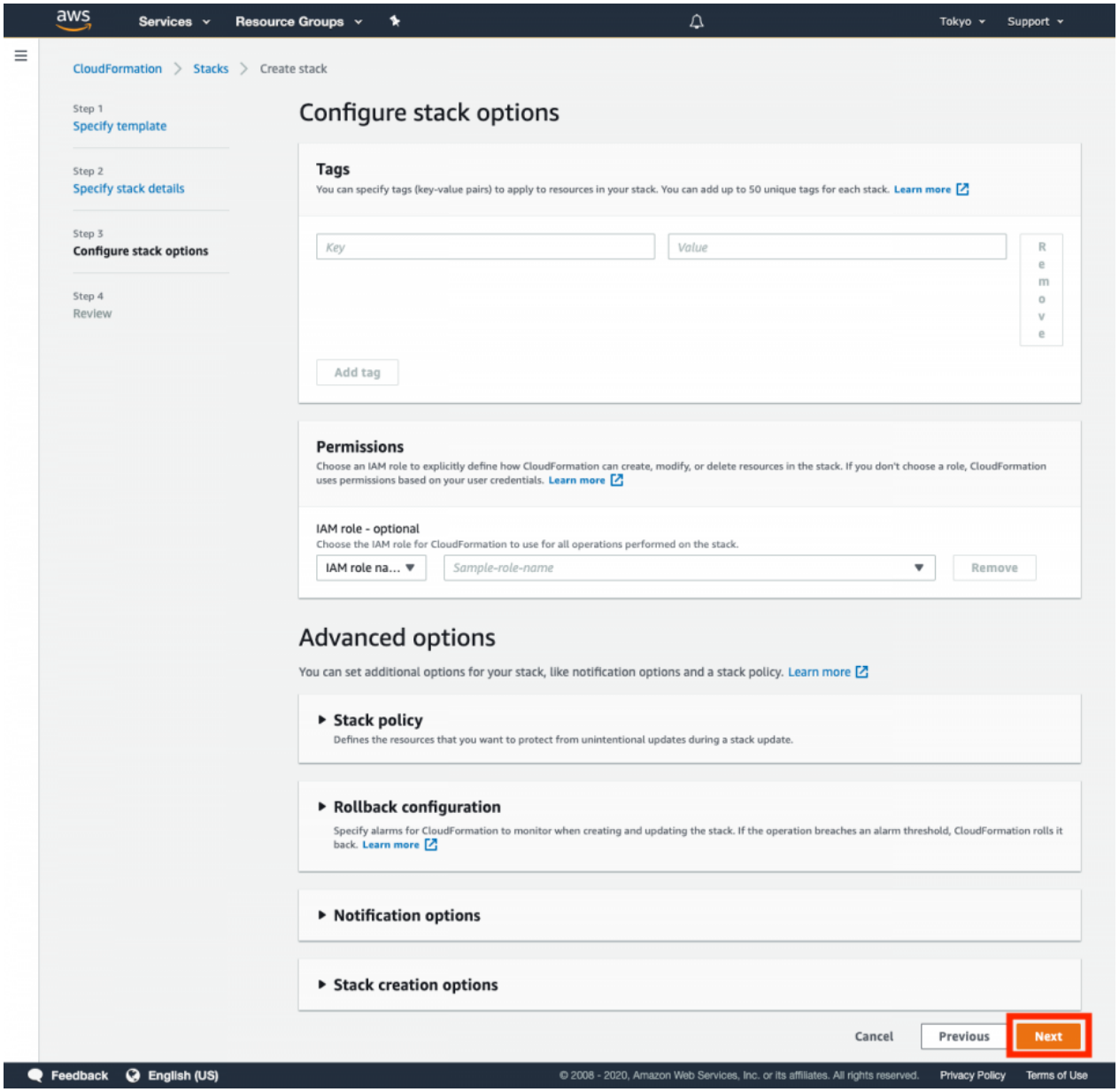
continues on next page

Table 8 – continued from previous page

No	Parameter	Describe	Recommended value
㉖	CatapultShareMode	Specify where to save block generation information Currently fixed to AWS Systems Manager	ssm
㉗	CatapultNetwork	mijin ネットワーク指定	mijin
㉘	Catapult BlockGenerationTargetTime	ブロック生成間隔	15s
㉙	CatapultEffectiveFee	手数料有無の設定	No
㉚	MaxCosignedAccount	最大署名数	25
㉛	FinalizationType	finalization system	Deterministic
㉜	MaxTransactionperBlock	1 ブロックの最大トランザクション数	6000
㉝	RestThrottring	API 接続数上限	30tps
㉞	UnconfirmCacheSize	Number of unauthorized transactions stored	Small
㉟	UseLoadBalancer	NLB use or not (fixed Yes for preview)	Yes
㊱	LoadBalancerType	ロードバランサー配置タイプ	External
㊲	ImageId	AMI ID (cannot be changed)	Unchangeable
㊳	MPS3BucketName	S3 bucket name (cannot be changed)	Unchangeable
㊴	MPS3BucketRegion	S3 Region (cannot be changed)	Unchangeable
㊵	MPS3KeyPrefix	S3 Key Prefix (cannot be changed)	Unchangeable
㊶	mijinStackAlreadyExist	If there are other mijin stacks, select “Yes”when recreating	No

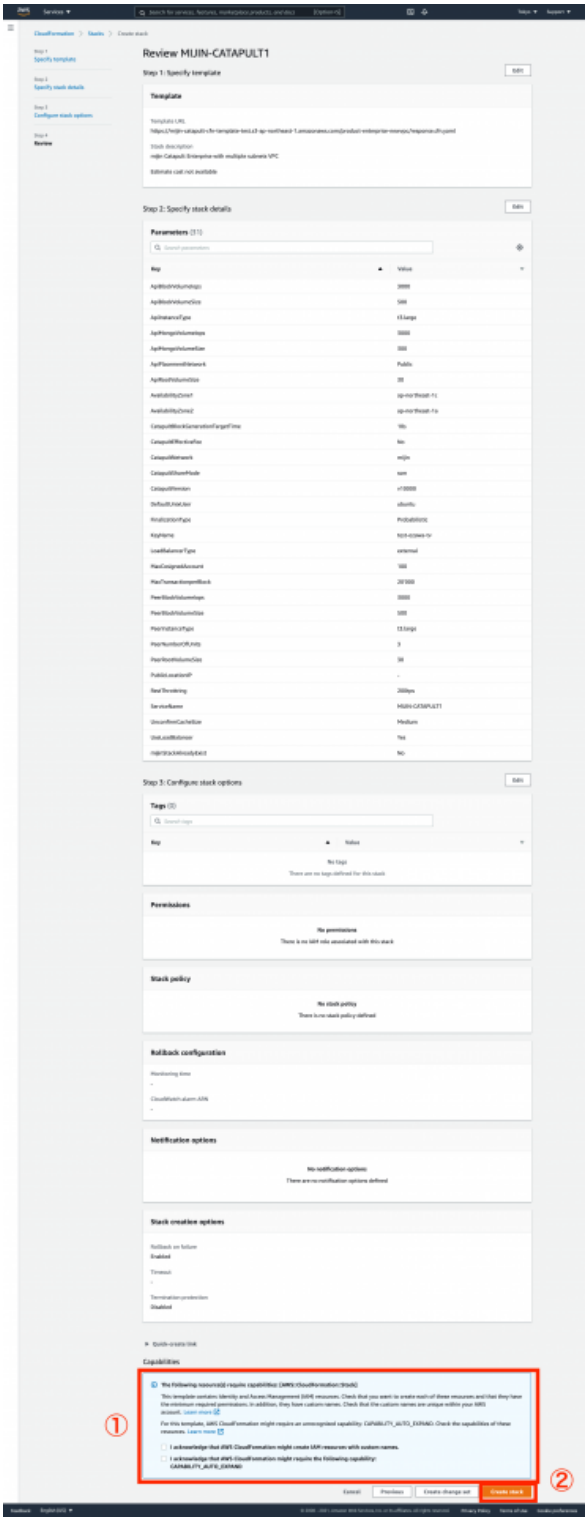
Press ‘Next’after completing the parameter entry.

2.2.3.11 Step.8



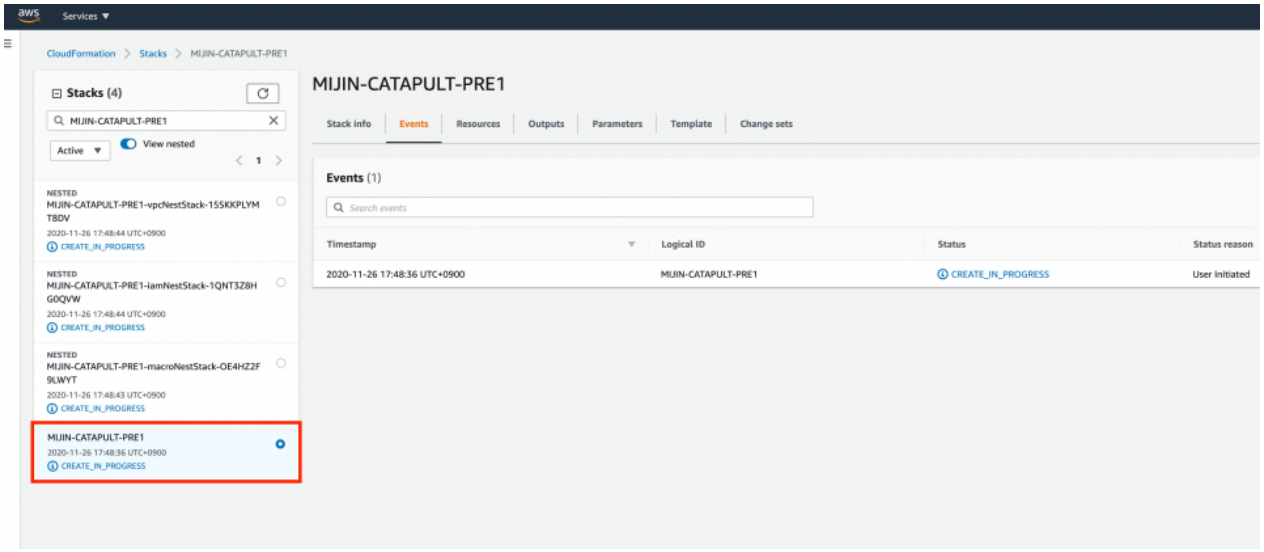
Without editing anything in particular, press 'Next'in the red frame.

2.2.3.12 Step.9



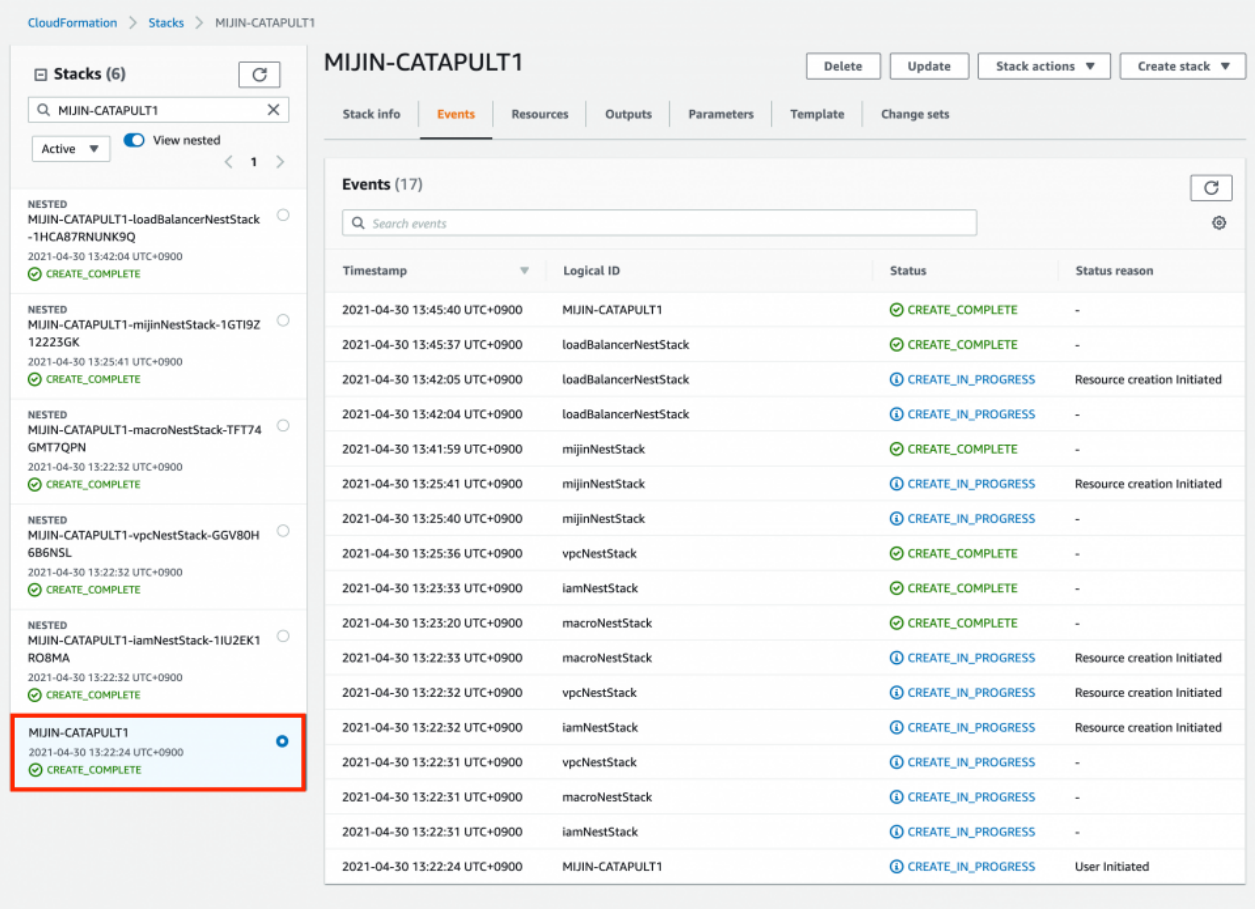
Check the two items in the red frame in ①. Click 'Create Stack'in the red frame in ②. If there are no errors, creation will begin.

2.2.3.13 Step.10



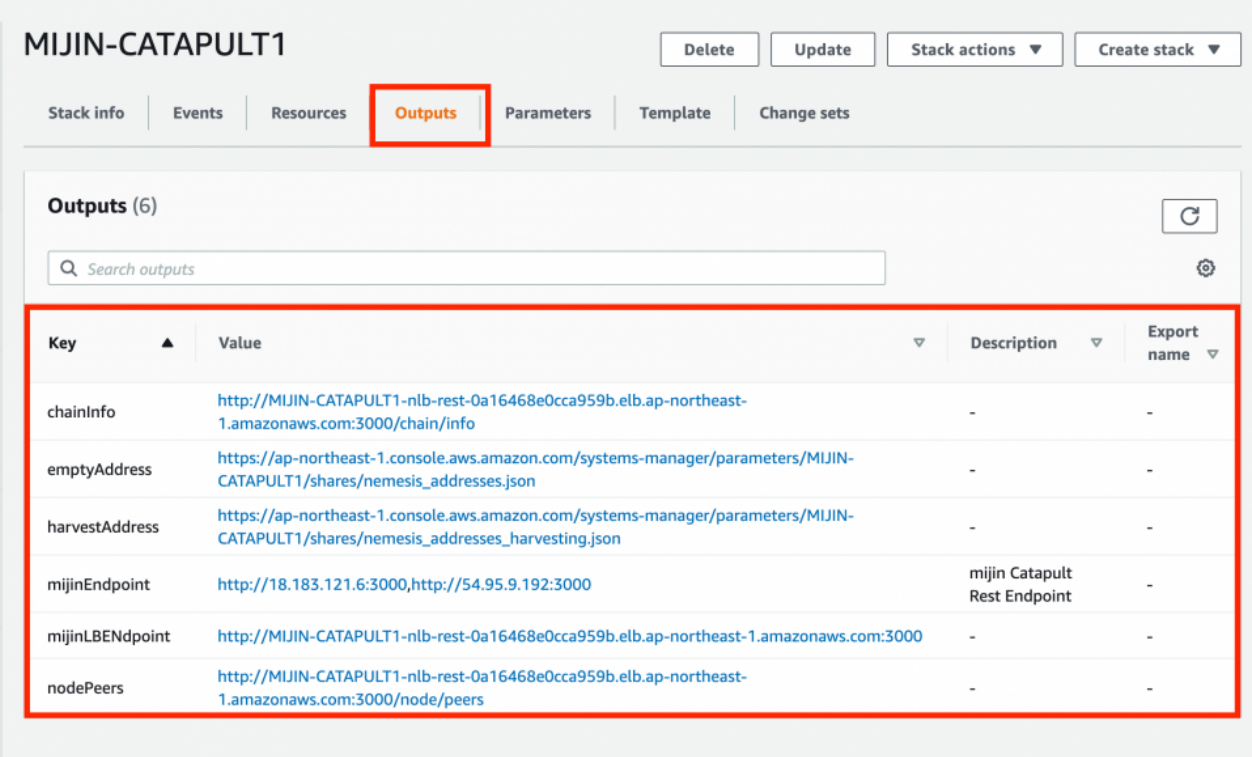
Confirm that Stack has started and is in 'CREATE_IN_PROGRESS'state. This state will take approximately 20~30 minutes.

2.2.3.14 Step.11



If the status is 'CREATE_COMPLATE', the creation of mijin is complete.

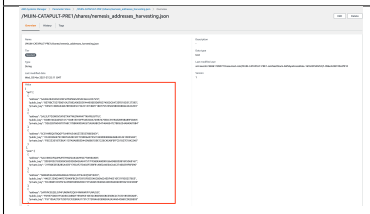
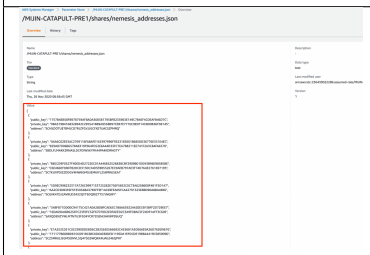
2.2.3.15 Step.12



Press ‘Outputs’of the created Stack to see the created mijin configuration information.

※The table below is omitted because it is the same as the new network version.

Table 9: mijin endpoints and checkpoints

	mijinLBEndpoint ロードバランサーを通した mijin の API エンドポイントです。API ノードを負荷分散しますが、ソース IP によるスティッキーセッションが有効です。詳細はこちら
	mijinEndpoint API endpoint for direct access to API nodes (EC2 instances). It can be connected to without a load balancer.
	**chainInfo** You can check the current number of blocks in mijin. Make sure that the number of blocks is “2”or more.
	**harvestAddress** A link to a currency distribution address registered in the AWS Systems Manager parameter store.
	**emptyAddress** A link to an unused address registered in the AWS Systems Manager parameter store.
	**nodePeers** mijin API からノードの接続状態を確認できます。API ノード 1 台と設定された PEER ノード数が表示されていれば正常です。

You are now ready to use mijin Catapult. Let’s start the operation in the next section!

2.2.4 Deploying a trial version of mijin

In the trial version, it is possible to build a network that fits your environment by changing parameters. An example pattern is shown below.

This page will guide you through the process of launching the mijin Catapult trial version from cloudformation in the AWS Marketplace.

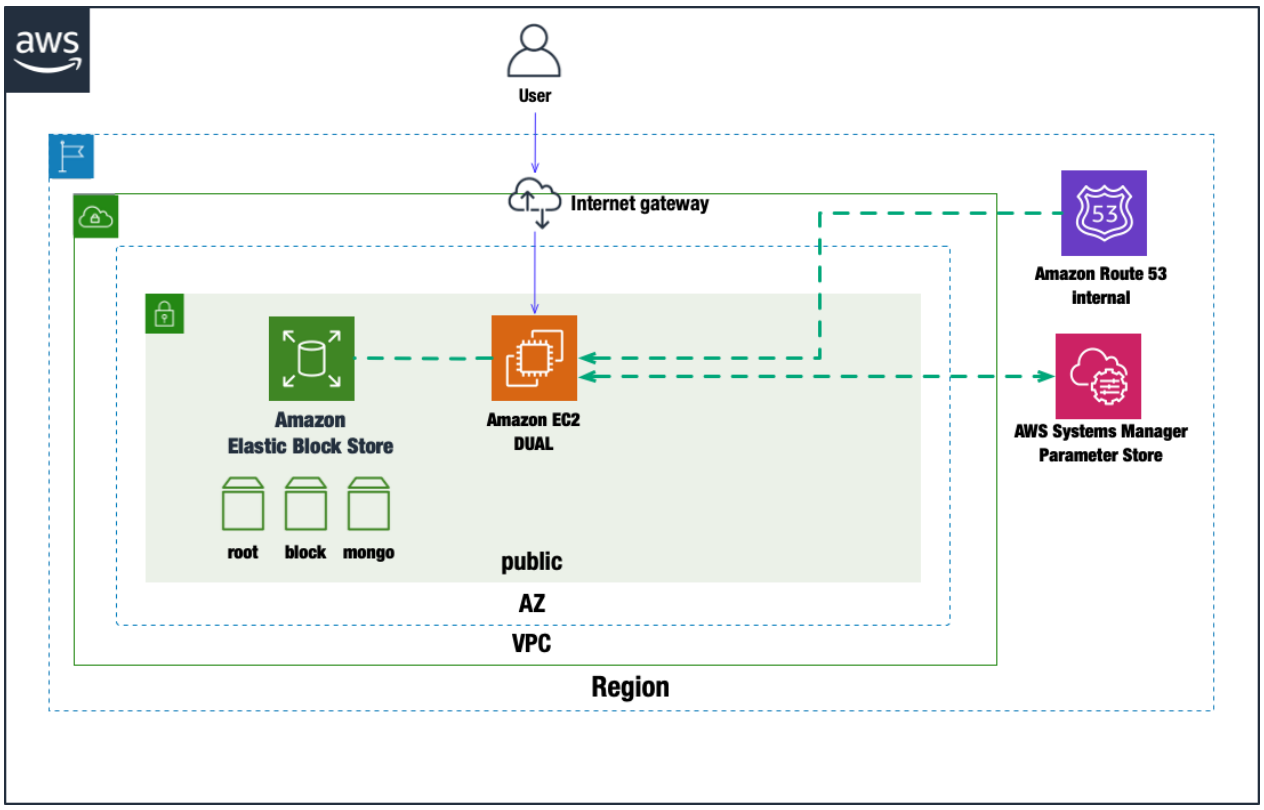
2.2.4.1 Services using AWS

- Amazon EC2
- Amazon EBS
- Amazon Route53
- Amazon VPC(Nat Gateway)
- parameter store

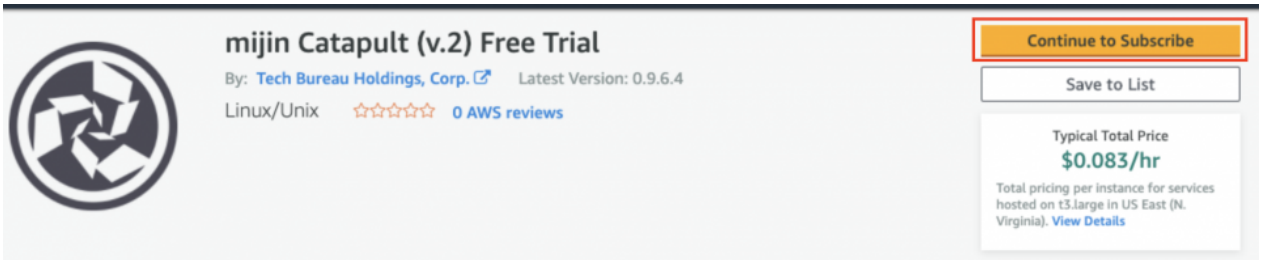
2.2.4.2 View Network

Note:

The trial version consists of only a single AZ.
Deployment is single region, but can be deployed in each of the world's 20 regions.

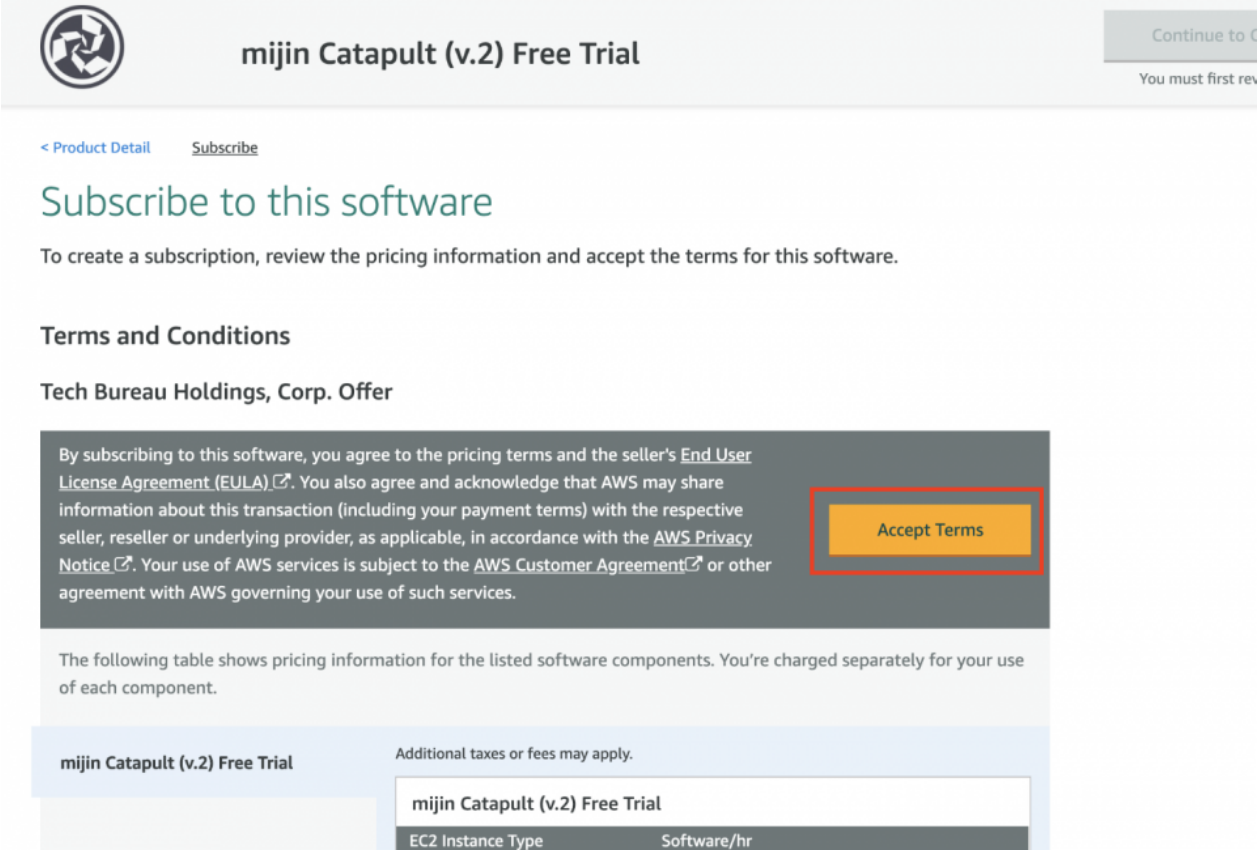


2.2.4.3 Step.1



You must subscribe to use the AMI for mijin Catapult Free Trial. Please press the button in the red frame.

2.2.4.4 Step.2



mijin Catapult AMI, so please approve its use.

2.2.4.5 Step.3

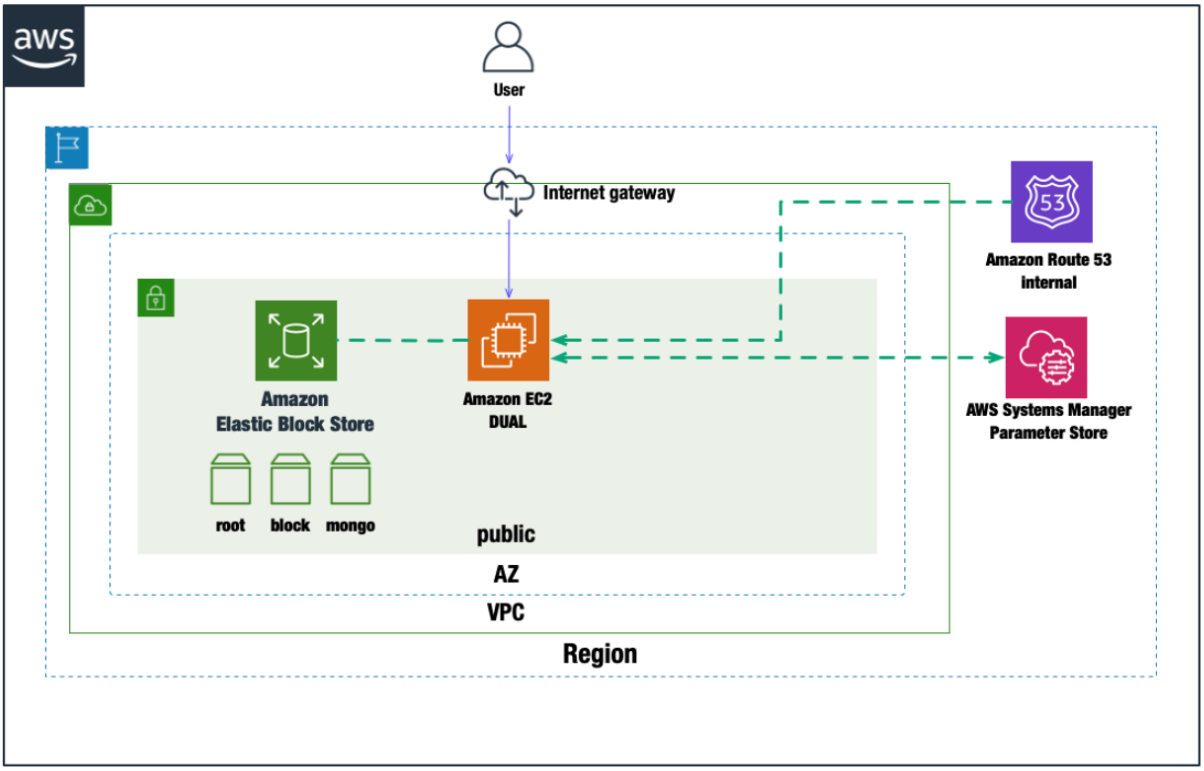
mijin Catapult Free Trial CFT 0.9.6.4
CloudFormation Template

This Cloudformation Stack makes it easy to build mijin.
Users create a new VPC and place a single EC2 instance in a public network.
After the Cloudformation execution is complete, the mijin address data is stored in the System Manager's Parameter Store. Using symbol-cli without having to remotely log in to the server, users are able to use the mijin immediately.

View Template Components

View Usage Instructions

Close CloudFormation Template

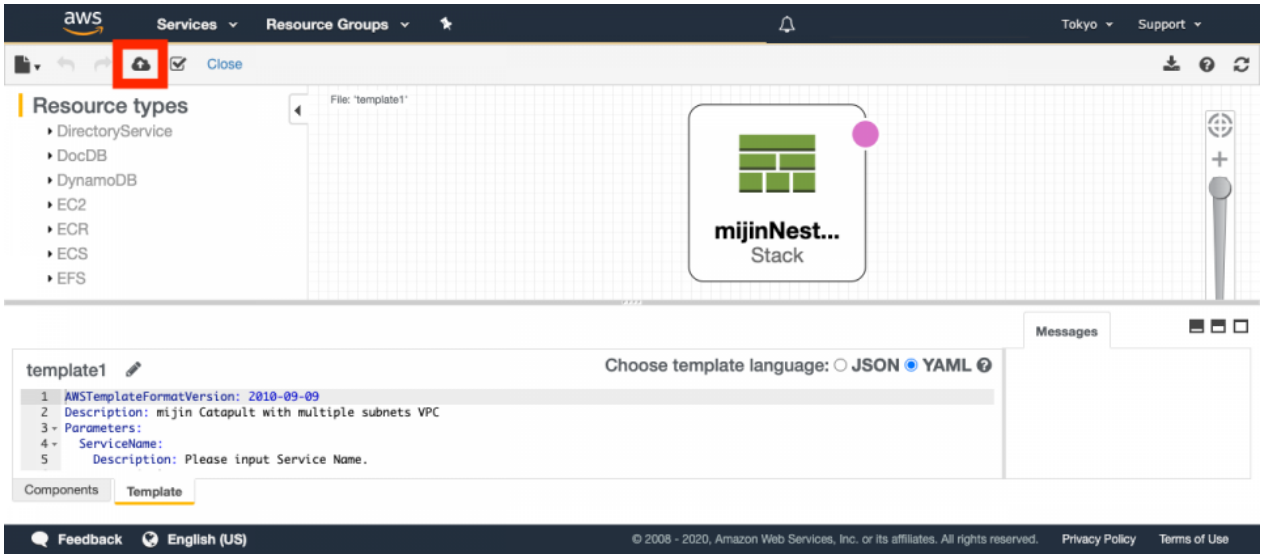


Download CloudFormation Template

View Template in CloudFormation Designer

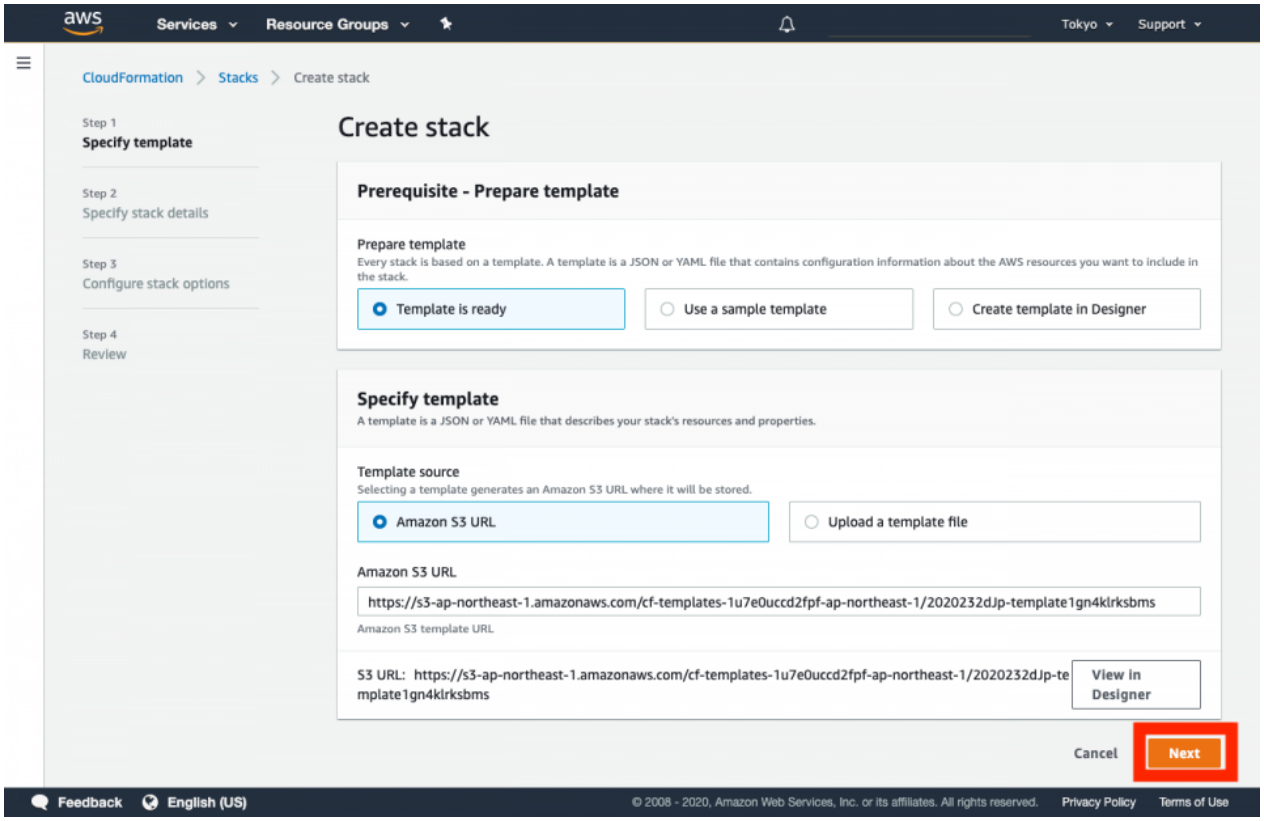
To create the network shown in the figure by AWS CloudFormation, click on the text in the red box.

2.2.4.6 Step.4



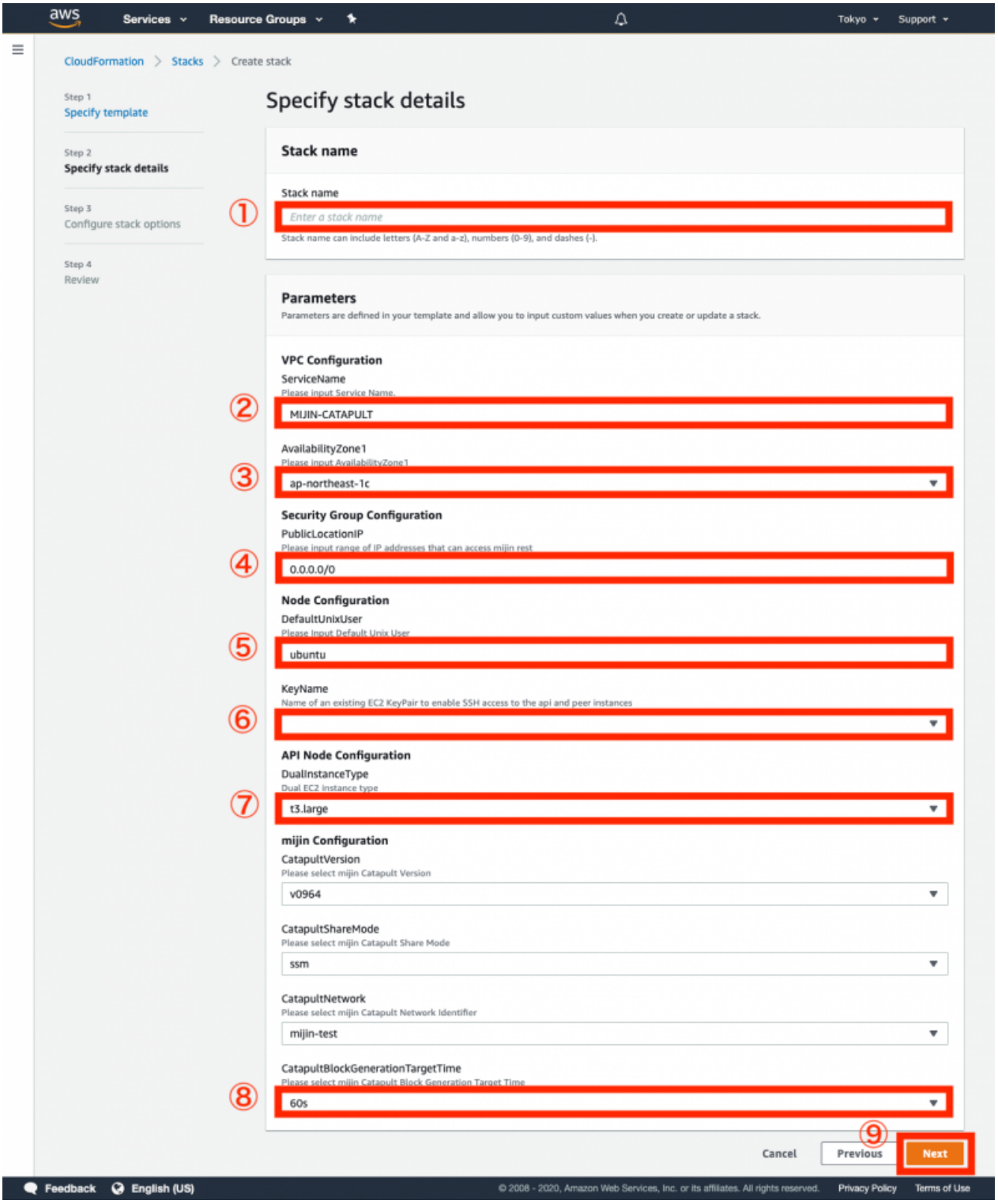
Press CreateStack on the cloud symbol without editing anything in particular.

2.2.4.7 Step.5



Without editing anything in particular, press 'Next'in the red frame.

2.2.4.8 Step.6

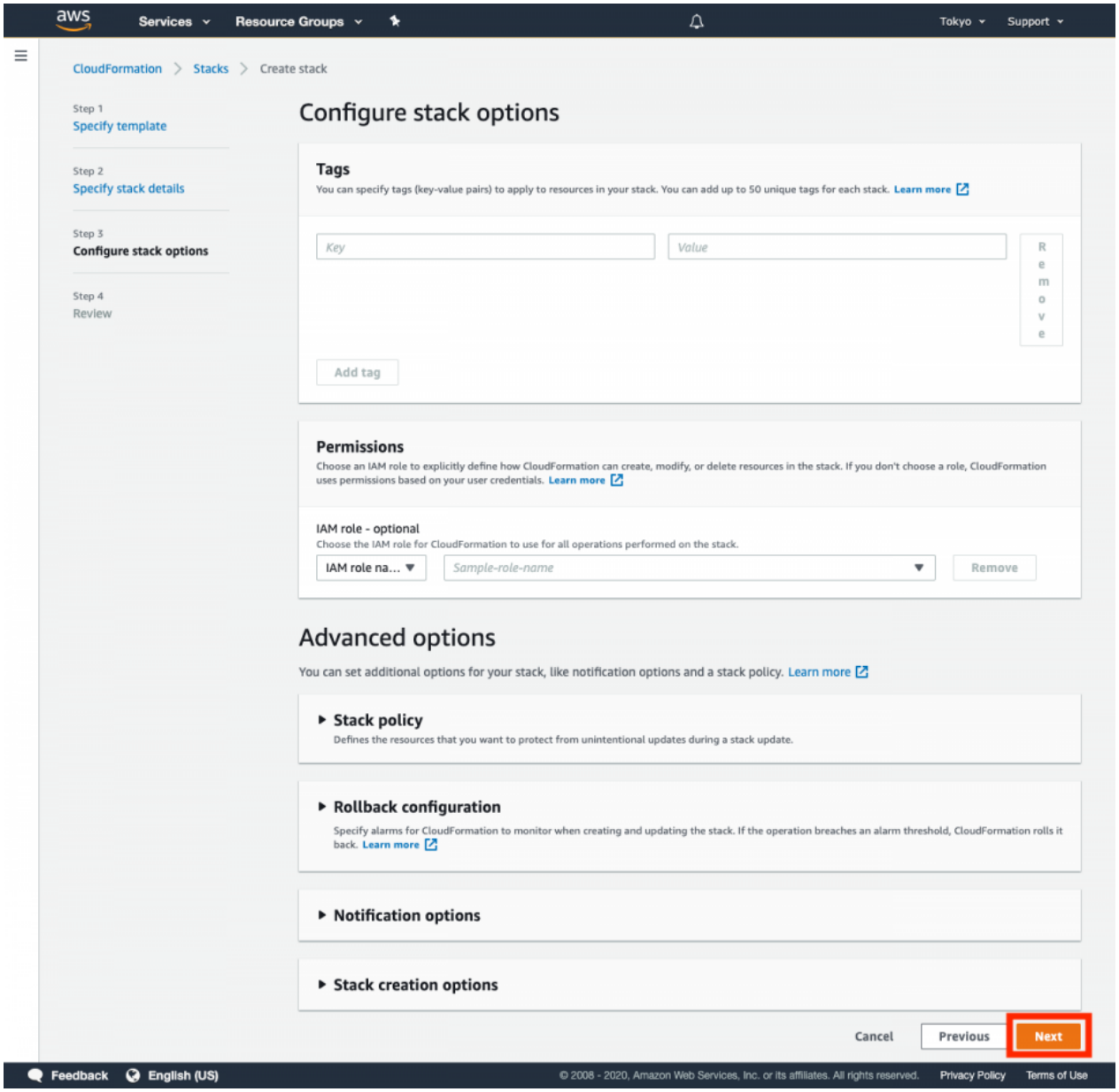


Enter parameters.

No	Parameter	Describe
①	Stack Name	Provide the name in this stack.
②	Service Name	Describe the name of the service created by the stack. This will be used as the crown name for all resources.
③	Availability Zone	Select the availability zone for your region.
④	Public Location IP	Specify the IP address that is allowed to connect to mijin Catapult's API. IP アドレスはレンジでも可能です（/24 など）
⑤	Default UnixUser	List the standard Unix user for the EC2 instance you are creating.
⑥	KeyName	Select the SSH key for the remote connection for the EC2 instance you are creating. 表示されていない場合、事前に鍵を作成する必要があります。 鍵の作成方法は こちら をご確認ください。
⑦	Dual InstanceType	Select the specifications of the EC2 instance to be created.
⑧	Catapult BlockGenerationTargetTime	Select the mijin block generation time; you can choose 30 or 60 seconds only.

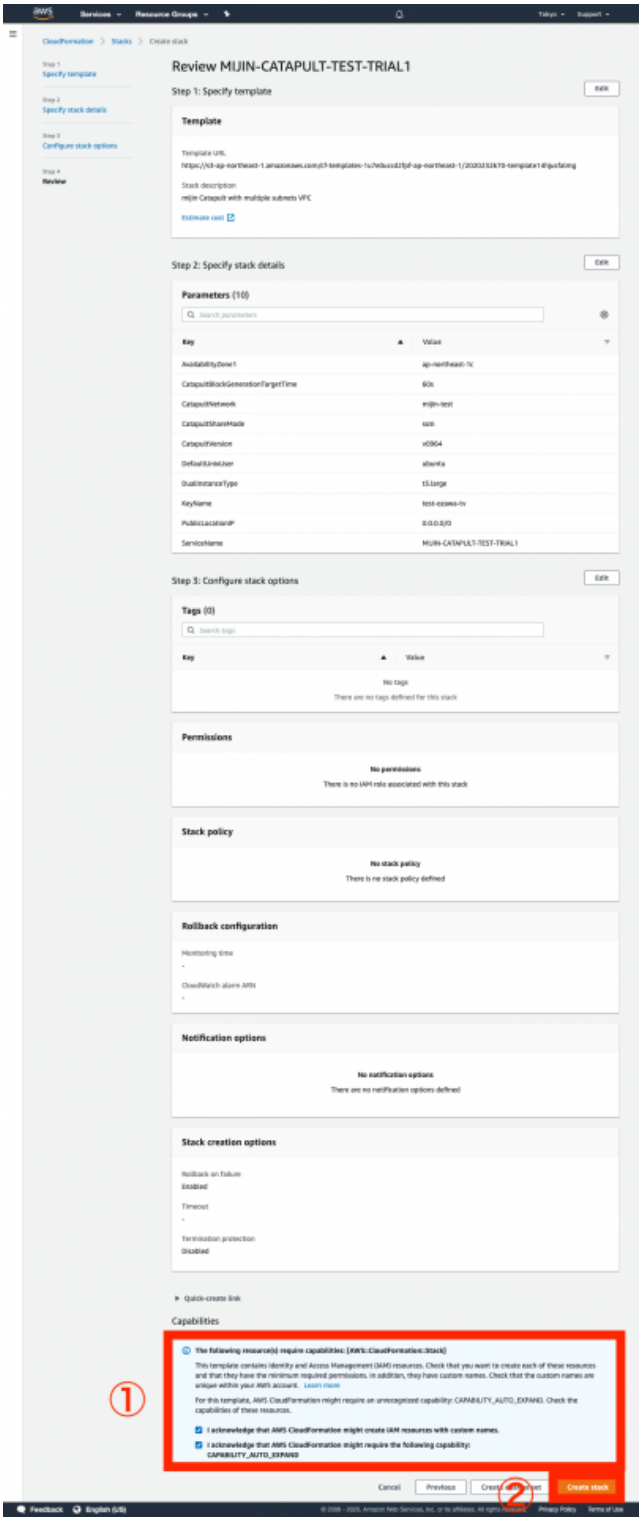
After completing the parameter entry, press 'Next'(⑨).

2.2.4.9 Step.7



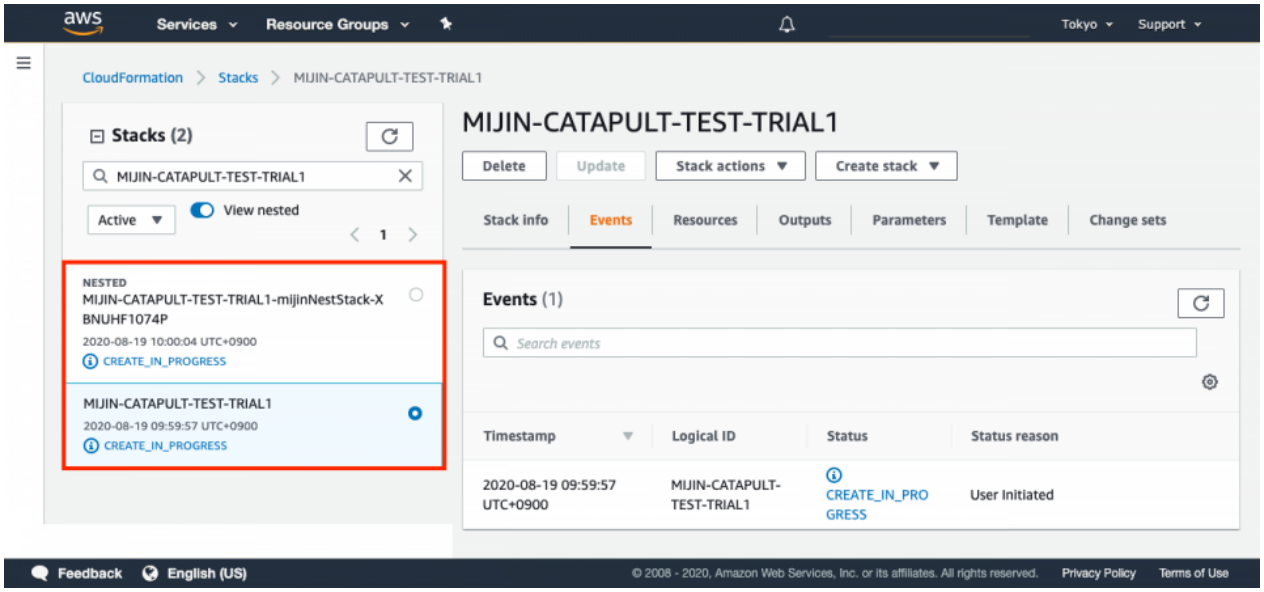
Without editing anything in particular, press 'Next'in the red frame.

2.2.4.10 Step.8



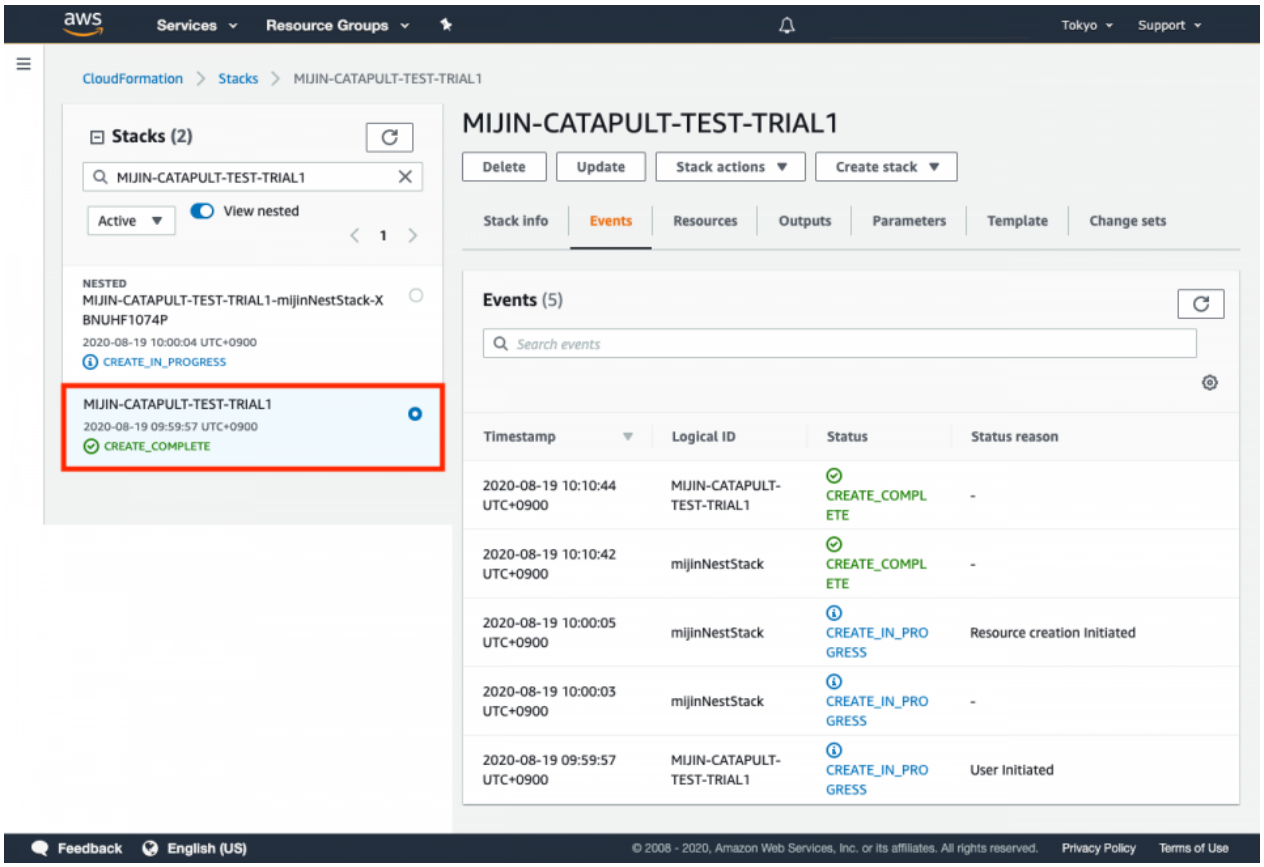
Check the two items in the red frame in ①. Day: Click 'Create Stack'in the red frame in ②. If there are no errors, creation will begin.

2.2.4.11 Step.9



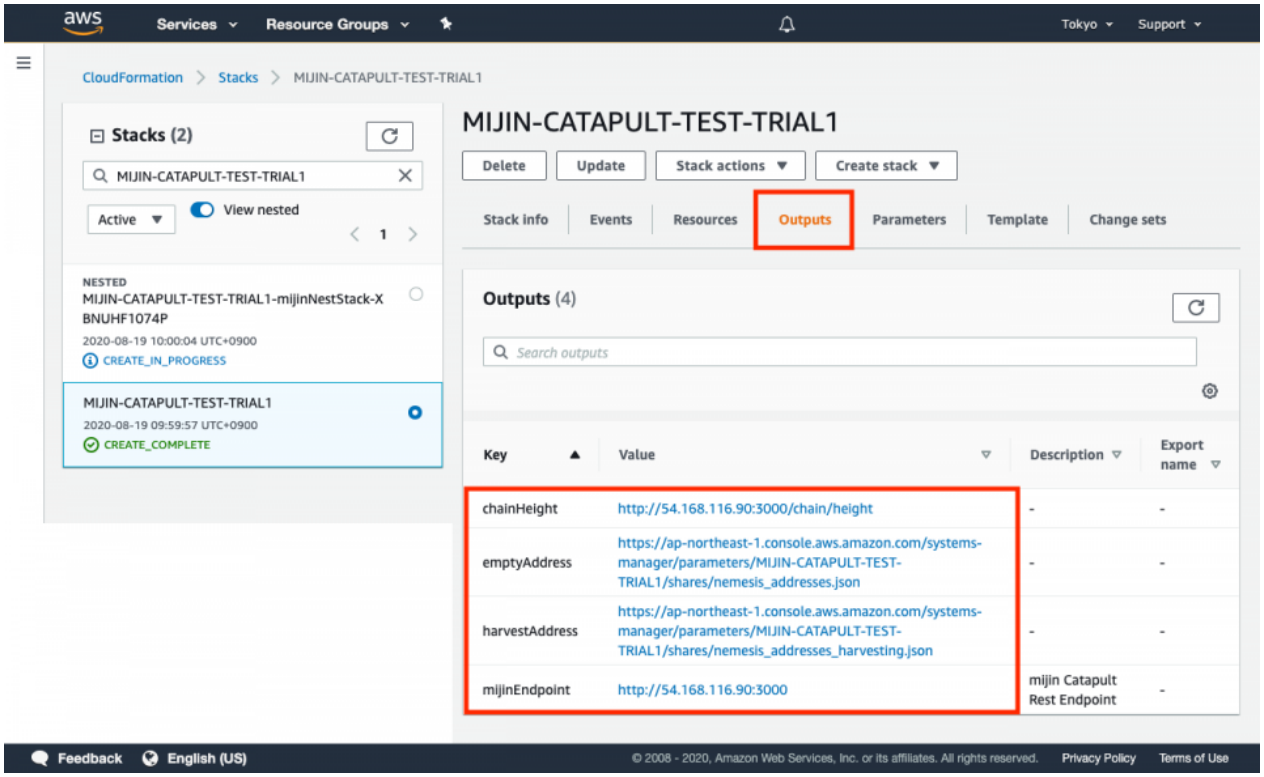
Confirm that Stack has started and is in 'CREATE_IN_PROGRESS'state. This state will take approximately 15~20 minutes.

2.2.4.12 Step.10



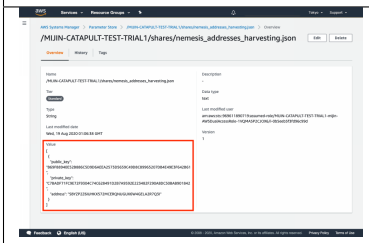
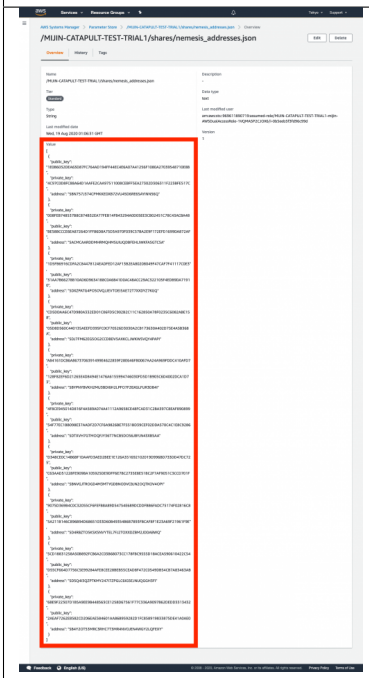
If the status is ‘CREATE_COMPLATE’, the creation of mijin is complete.

2.2.4.13 Step.11



Press ‘Outputs’of the created Stack to see the created mijin configuration information.

Table 10: mijin エンドポイント確認項目（トライアル）

	mijinEndpoint mijin の API エンドポイントです。
	**chainHeight** mijin の現在のブロック数の確認ができます。ブロック数が「2」以上になっていることを確認してください。
	**harvestAddress** AWS Systems Manager パラメータストアに登録された、基軸通貨（トライアル版のため 2000cat.currency）を分配したアドレスのリンクです。
	**emptyAddress** A link to an unused address registered in the AWS Systems Manager parameter store.

You are now ready to use mijin Catapult. Let's start the operation in the next section!

2.2.5 AWS MarketPlace Technical Documentation

Add various technical documents in AWS MarketPlace.

2.2.5.1 AWS Marketplace Cloudformation Parameter Comparison Table

This chapter describes the parameters and default values for deploying mijin Catapult(v.2) in AWS Marketplace.
The two commercial versions and the trial version have different default values.

Cloudformation Parameter Comparison Table

Table 11: Trial / Product 設定一覧

No	Categories	Configuration Name	Description	set value (e.g. of a function, parameter, etc.)	Limits	Trial setting available	Trial デフォルト値	Product NewVPC デフォルト値	Product ExistsVPC デフォルト値
1	VPC Configuration	ServiceName	Specify the name of the service that will be the crown name of the resource.	String	記号始まりNG、大文字小文字英数字、ダッシュ (-) 使用可能	○	MIJIN-CATAPULT	MIJIN-CATAPULT	MIJIN-CATAPULT
2		AvailabilityZone1	Specify the AZ to be used in the VPC.	List	Region's AZ	○	Region-dependent	Region-dependent	•
3		AvailabilityZone2	Specify the AZ to be used in the VPC (specify different from AZ1)	List	Region's AZ	•	Region-dependent	Region-dependent	•
4		VPC	Specifies a VPC Id that already exists.	List	VpcId in the region	•	•	•	○
5		VpcCidrBlock	Specify IP range of VPC (e.g., vpc-xxxx (xx.xx.xx.xx/16))	List	Regular expression: (d{1,3}.{3}d{1,3}/d{1,2}	•	•	•	○
6		Public1	Specify the subnet of the public network (AZ1)	List		•	•	•	○
7		Public2	Specify the subnet of the public network (AZ2) (different from Public1)	List		•	•	•	○
8		Private1	Specify subnet for non-public network (AZ1)	List		•	•	•	○
9		Private2	Specify a subnet for the private network (AZ2) (different from Private1)	List		•	•	•	○
10		InternalDomainName	Specify an internal DNS name (e.g. mijin.internal)	String		•	•	mijin.internal	mijin.internal
11	Security Group Configuration	PublicLocationIP	Permitted IP range to connect to mijin endpoints	String	Regular expression: (d{1,3}.{3}d{1,3}/d{1,2}	○			

continues on next page

Table 11 – continued from previous page

No	Categories	Configuration Name	Description	set value (e.g. of a function, parameter, etc.)	Limits	Trial setting available	Trial デフォルト値	Product NewVPC デフォルト値	Product ExistsVPC デフォルト値
12	Node Configuration	DefaultUnixUser	EC2 remote login user name	String	“catapult” 以外	○	ubuntu	ubuntu	ubuntu
13		KeyName	SSH key used in EC2	List		○	Region-dependent	Region-dependent	Region-dependent
14	API Node Configuration	ApiPlacementNetwork	API node placement (Public or Private)	List	Public, Private	•	•	Public	Public
15		ApiInstanceType	API インスタンスのスペック	List	several	△	t3.large	t3.large	t3.large
16		ApiRootVolumeSize	ルートボリュームの容量（GB）	List	30, 100	•	30	30	30
17		ApiBlockVolumeSize	ブロックデータ用のディスク容量（GB）	List	50, 300, 500, 800, 1000	•	50	500	500
18		ApiBlockVolumeIops	ブロックデータ用 IOPS（gp3）	List	3000, 5000, 10000	•	100	3000	3000
19		ApiMongoVolumeSize	MongoDB 用のディスク容量（GB）	List	50, 300, 500, 800, 1000	•	50	300	300
20		ApiMongoVolumeIops	MongoDB 用 IOPS（gp3）	List	3000, 5000, 10000	•	300	3000	3000
21	PEER Node Configuration	PeerNumberOfUnits	PEER ノード台数（固定）	Int	3–10	•	•	3	3
22		PeerInstanceType	PEER ノードのインスタンスタイプ	List		•	•	t3.large	t3.large
23		PeerRootVolumeSize	PEER ノードのルートディスク容量	List	30, 100	•	•	30	30
24		PeerBlockVolumeSize	PEER ノードのブロックデータ用容量	List	50, 300, 500, 800, 1000	•	•	500	500
25		PeerBlockVolumeIops	PEER ノードの IOPS（io1）	List	3000, 5000, 10000	•	•	3000	3000
26	mijin Configuration	CatapultVersion	起動時の Catapult バージョン	List	v10037,v10038	△	v10038	v10038	v10038
27		CatapultShareMode	初期データ（アドレス等）の保存場所（SSM 推奨）	List	ssm	△	ssm	ssm	ssm
28		mijinDataDiretoryName	データマウント先のパスを指定	String	絶対パス	•	/mnt/mijin	/mnt/mijin	/mnt/mijin
29		CatapultNetwork	mijin のネットワークタイプ	List	mijin, mijin-test	△	mijin-test	mijin	mijin
30		CatapultBlockGenerationTargetTime	ブロック生成間隔（目安）	List	5s, 15s, 30s, 60s	△	60s	15s	15s
31		CatapultEffectiveFee	トランザクション手数料有無	Boolean	Yes, No	•	Yes	No	No
32		MaxCosignedAccount	最大連署アカウント数	List	25, 50, 100, 1000	•	25	25	25

continues on next page

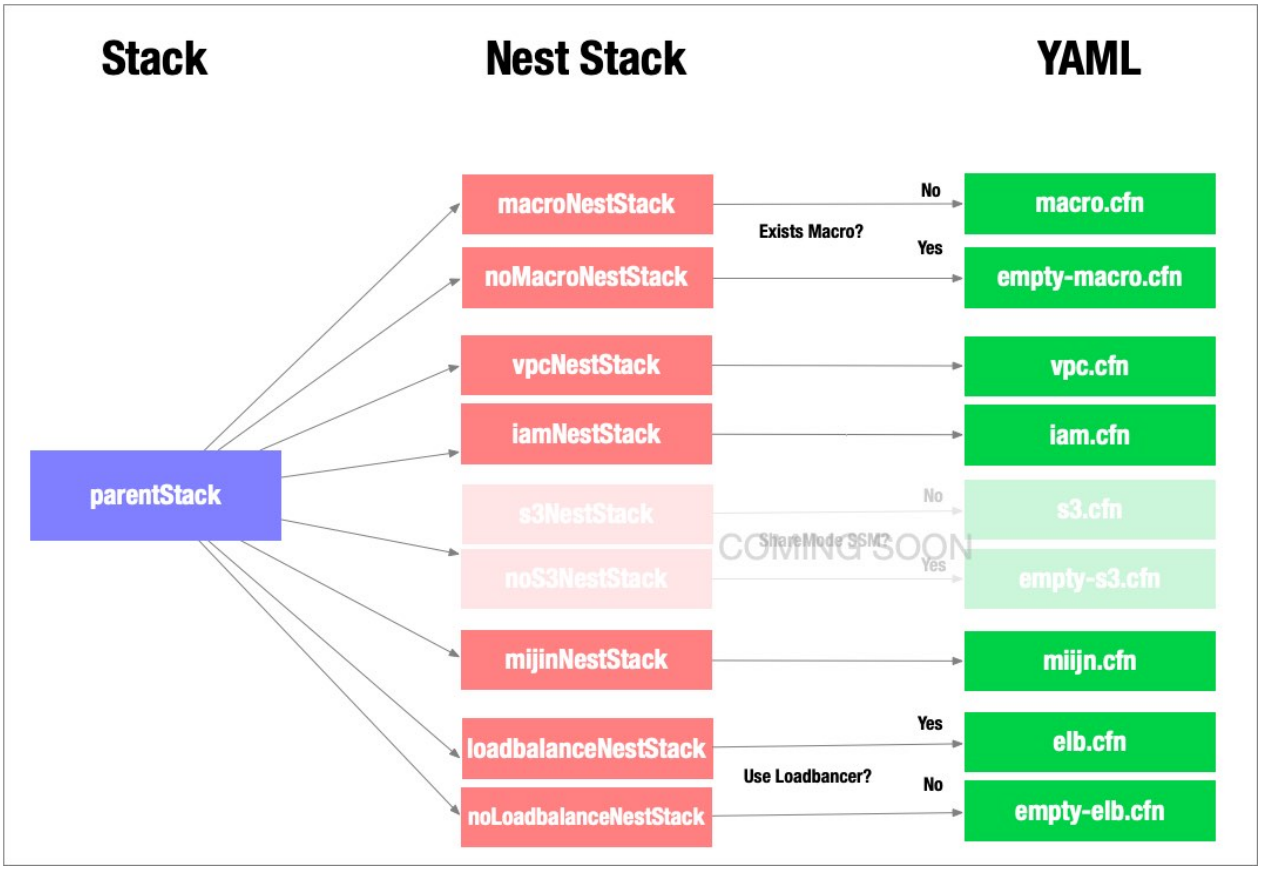
Table 11 – continued from previous page

No	Categories	Configuration Name	Description	set value (e.g. of a function, parameter, etc.)	Limits	Trial setting available	Trial デフォルト値	Product NewVPC デフォルト値	Product ExistsVPC デフォルト値
33		FinalizationType	ファイナライズ方式		Deterministic, Probabilistic	•	Deterministic	Deterministic	Deterministic
34		MaxTransactionperBlock	ブロック内最大トランザクション数	List	6'000, 10'000, 20'000, 50'000, 100'000	•	6'000	6'000	6'000
35		RestThrottring	API 接続数（バースト時+100）	List	30tps, 100tps, 200tps, 500tps, NoLimit	•	30tps	30tps	30tps
36		UnconfirmCacheSize	未承認トランザクションのキャッシュサイズ	List	Small, Medium, Large	•	Small	Small	Small
37	LoadBalancer Configuration	UseLoadBalancer	NLB を使用するか	Boolean	Yes, No	•	•	Yes	Yes
38		LoadBalancerType	NLB の配置場所	List	external, internal	•	•	external	external
39	Other	mijinStackAlreadyExist	他スタックが存在するかの指定	Boolean	Yes, No	•	•	No	No

2.2.5.2 AWS MarketPlace Cloudformation Specifications

The AWS MarketPlace service, mijin Catapult(v.2), is deployed using Cloudformation, an orchestration tool.
This chapter describes the AWS resources created using Cloudformation.

The Cloudformation Template (CFT) consists of multiple files, where the parent Stack calls each of its child Stacks. The child Stacks to be invoked depend on the parameters of the parent Stack.



macroNestStack

macroNestStack creates a Cloudformation Macro. If the parameter mijinStackAlreadyExist is Yes, a stack that does not create a Macro will call (empty-macro). This is because the Cloudformation Macro is created with a unique name, so it is not possible to create multiple Macro's with the same name.

Macro is created in Lambda (Node.js) and converts the CFT of mijinNestStack that comes after this by specified parameters.
Assign the following IAM roles and policies to allow the created Lambda to read and write to Amazon CloudWatch Logs.

```
Resources:
  PeerUnitsExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - lambda.amazonaws.com
```

(continues on next page)

(continued from previous page)

```
            Action:
              - 'sts:AssumeRole'
        Path: /
        Policies:
          - PolicyName: root
            PolicyDocument:
              Version: 2012-10-17
              Statement:
                - Effect: Allow
                  Action:
                    - 'logs:CreateLogGroup'
                    - 'logs:CreateLogStream'
                    - 'logs:PutLogEvents'
                  Resource:
                    - Fn::Join:
                      - ':'
                      -
                        - 'arn:aws:logs'
                        - Ref: 'AWS::Region'
                        - Ref: 'AWS::AccountId'
                        - 'log-group:/aws/lambda/*:*:*'
```

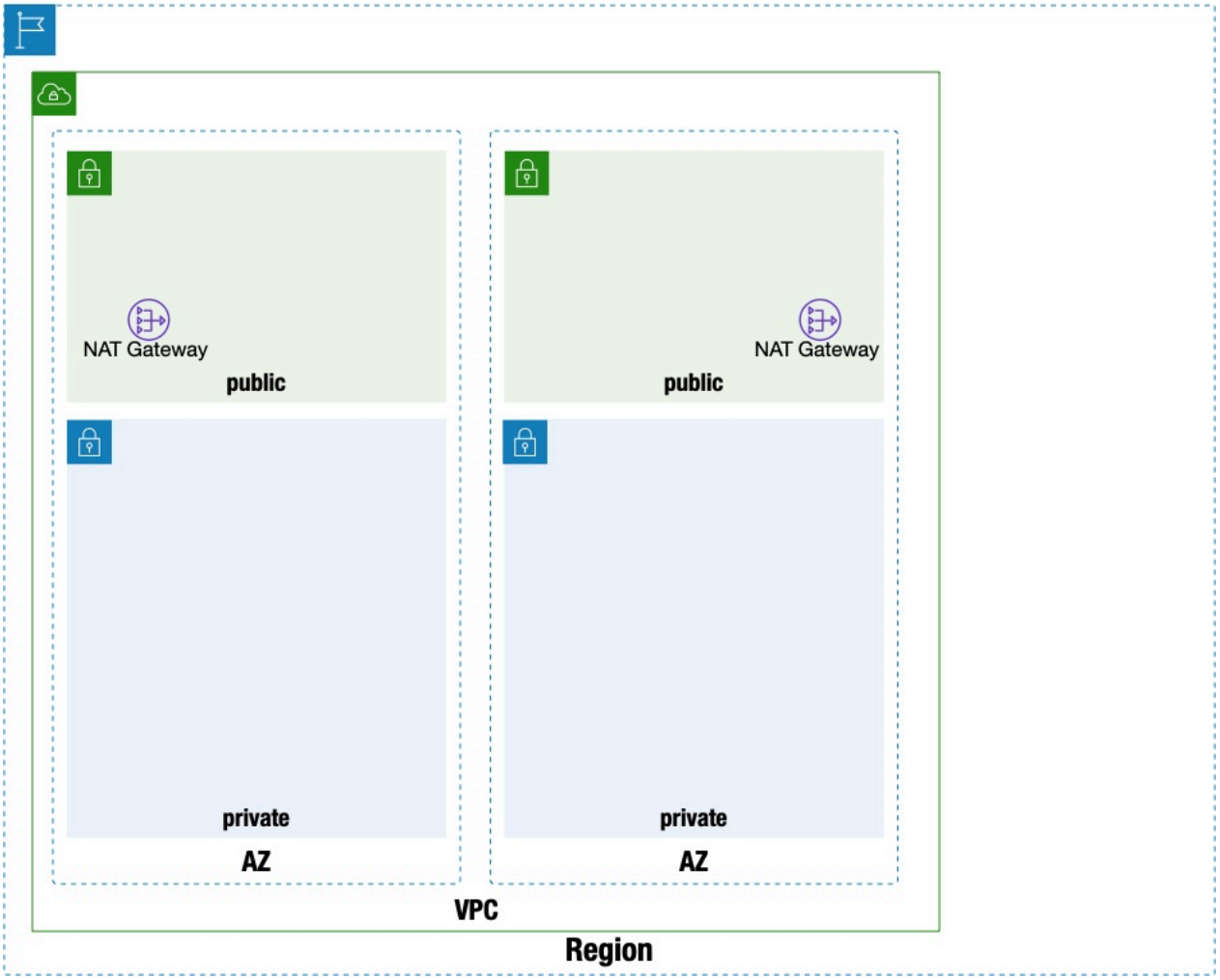
The Cloudformation Macro is created in Lambda (Node.js) and converts the CFT of the mijinNestStack that follows by the specified parameters.The Macro can replicate the stack of EC2 instances and dynamically change the EC2 instances to be launched, depending on the number of the parameter PeerNumberOfUnits.

Warning:

To create multiple mijin Catapult(v.2) in the same region, mijinStackAlreadyExist must be YES.

vpcNestStack

vpcNestStack creates a new VPC. A multi-AZ environment is created, with public and private subnets placed in each AZ. The default gateway for private network routing can also go out to the Internet using the Nat Gateway placed on the public network in the same AZ.



Note: This stack is not used when deploying mijin in an existing network.

iamNestStack

iamNestStack creates IAM roles and the IAM policies associated with them for use by EC2 instances.

The following are roles to be assigned to each API node and PEER node.

```
AWSApiAccessRole:
  Type: 'AWS::IAM::Role'
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
```

(continues on next page)

(continued from previous page)

```
          Action:
            - 'sts:AssumeRole'
    Path: /
AWSPeerAccessRole:
  Type: 'AWS::IAM::Role'
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - 'sts:AssumeRole'
    Path: /
```

Grant permissions to AWS Systems Manager Session Manager
AWS Systems Manager Session Manager grants privileges to EC2 instances to allow remote login with IAM privileges and ties them to IAM roles.

```
AWSSSMRolePolicies:
  Type: 'AWS::IAM::Policy'
  Properties:
    PolicyName: AWSSSMAccessPolicy
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Action:
            - 'ssm:DescribeAssociation'
            - 'ssm:GetDeployablePatchSnapshotForInstance'
            - 'ssm:GetDocument'
            - 'ssm:GetManifest'
            - 'ssm:GetParameters'
            - 'ssm:ListAssociations'
            - 'ssm:ListInstanceAssociations'
            - 'ssm:PutInventory'
            - 'ssm:PutComplianceItems'
            - 'ssm:PutConfigurePackageResult'
            - 'ssm:UpdateAssociationStatus'
            - 'ssm:UpdateInstanceAssociationStatus'
            - 'ssm:UpdateInstanceInformation'
          Resource: '*'
        - Effect: Allow
          Action:
            - 'ssmmessages:CreateControlChannel'
            - 'ssmmessages:CreateDataChannel'
            - 'ssmmessages:OpenControlChannel'
            - 'ssmmessages:OpenDataChannel'
          Resource: '*'
```

Grant read/write permissions to AWS Systems Manager Parameter Store.
The AWS Systems Manager parameter store stores dynamically generated data for the first node of mijin Catapult(v.2), grants permissions that can be referenced by each of the other nodes, and ties them to IAM roles.

```
AWSPSAccessRolePolicies:
  Type: 'AWS::IAM::Policy'
  Properties:
    PolicyName: AWSPSAccessRole
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Action:
            - 'ssm:PutParameter'
            - 'ssm:GetParameter'
            - 'ssm:GetParametersByPath'
```

Grant Security Token Service (STS) from the resource of the same account same service name IAM role and link the IAM role.
With the STS granted, the AWSApiAccessRole and AWSPeerAccessRole IAM roles can operate the AWS services specified in the policy.

```
AWSAssumeAccessRolePolicies:
  Type: 'AWS::IAM::Policy'
  Properties:
    PolicyName: AWSAssumeAccessRole
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Action:
            - 'sts:AssumeRole'
          Resource:
            - Fn::Join:
                    - ':'
                    -
                      - 'arn:aws:iam:'
                      - Ref: 'AWS::AccountId'
                      - !Sub "role/${ServiceName}*"
    Roles:
      - !Ref AWSApiAccessRole
      - !Ref AWSPeerAccessRole
```

s3NestStack

Warning: This feature is currently disabled and only the parameter store can be saved.

s3NestStack allows you to choose whether to place mijin data in the AWS Systems Manager parameter store or in an S3 bucket, and if you choose S3, it creates an S3 bucket.

mijinNestStack

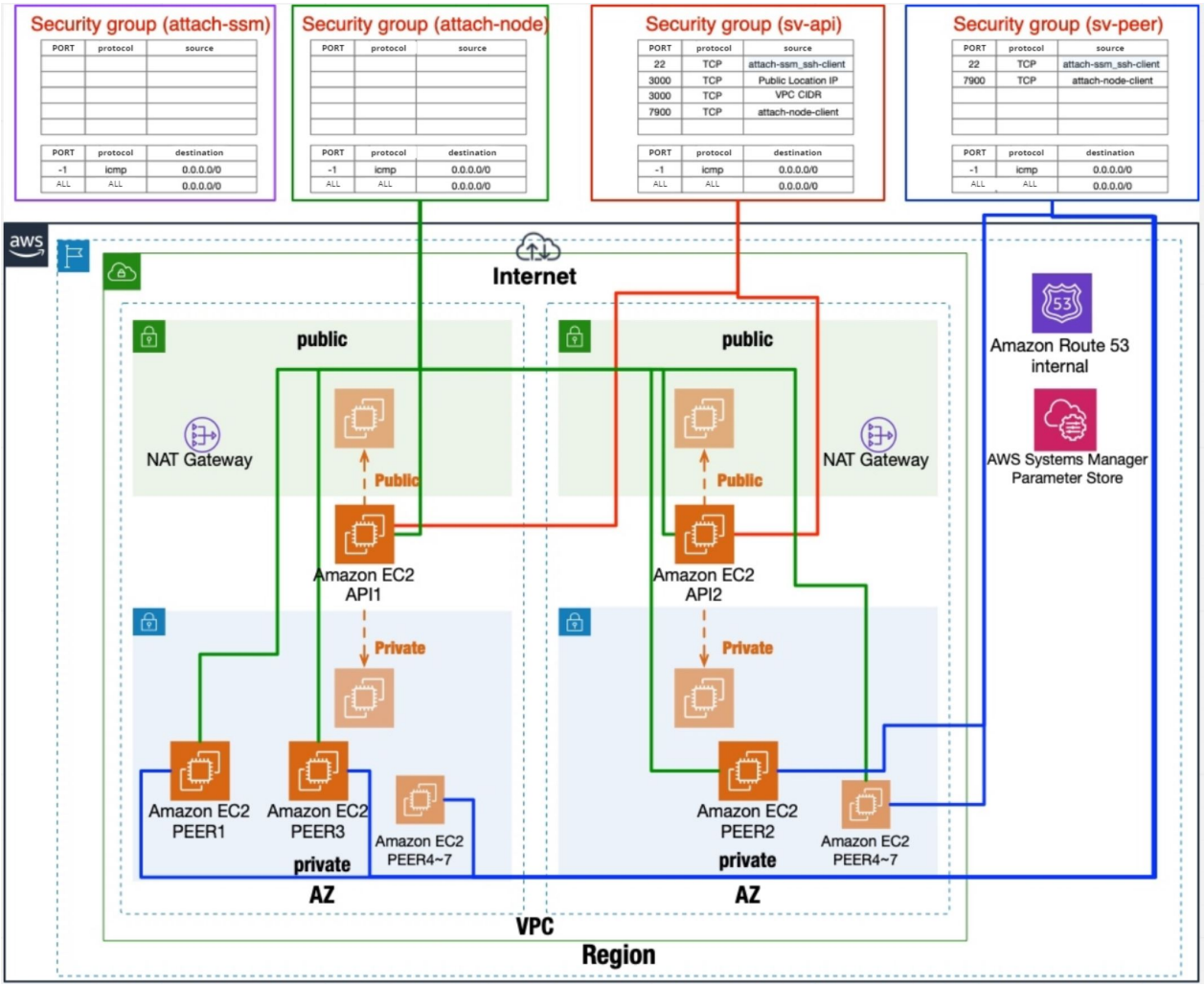
mijinNestStack creates private DNS, security groups, EBS, and EC2 instances with Route53.

mijin uses DNS names to communicate between each EC2 instance. The domain is fixed at mijin.internal, and each instance name is set to the A record as follows.

1. api1.mijin.internal
2. api2.mijin.internal
3. peer1.mijin.internal
4. peer2.mijin.internal
5. peer3.mijin.internal
6. peer4.mijin.internal
7. peer5.mijin.internal
8. peer6.mijin.internal
9. peer7.mijin.internal
10. peer8.mijin.internal
11. peer9.mijin.internal

Security groups are applied below, with the minimum security group used for communication as shown in the image.

Security group name	Description
attach-ssm_ssh-client	This security group can be set up as a stepping stone, etc. to allow SSH login. It is not used for new VPCs. (If created for an existing VPC, assign it to an existing stepping stone, etc.)
attach-node-client	For inter-node communication.
sv-api	This is for API nodes. 3000 port / for REST access. 7900 / For mijin node communication.
sv-peer	For PEER node. 7900 port / for mijin node communication.

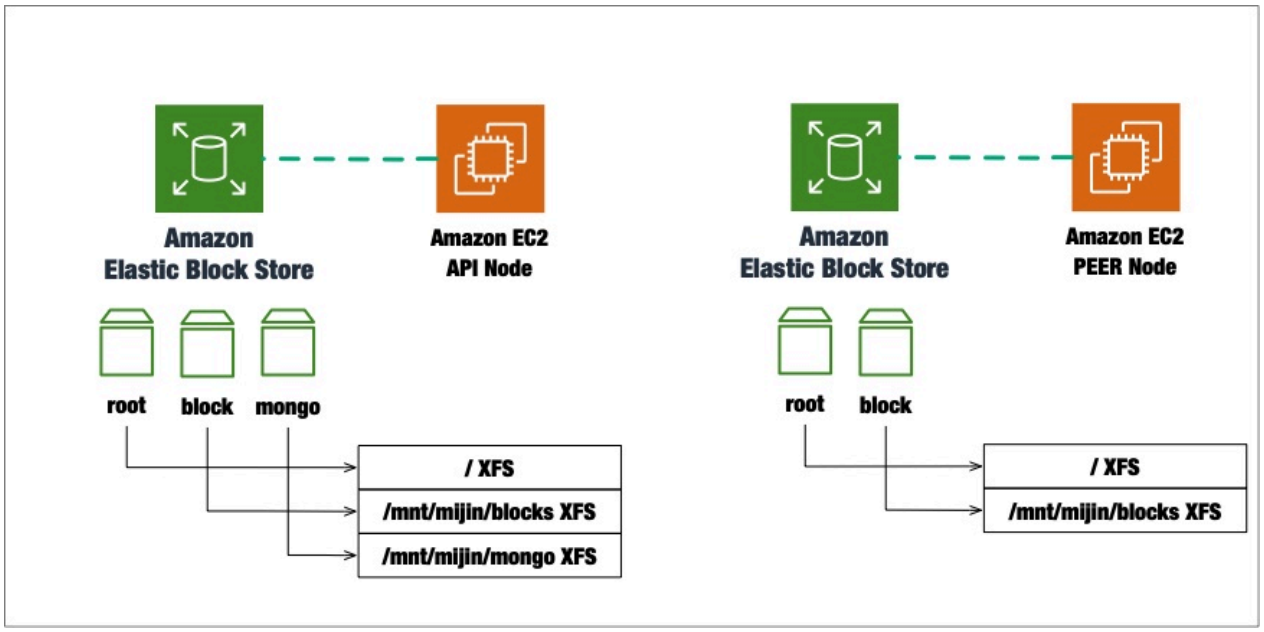


Since EBS stores mijin's data area, two EBSs are attached to the API node in addition to the root partition, and one EBS is attached to the PEER node.

VolumeType is fixed GP3 and no disk encryption.

The EBS is designed to be XFS formatted and disk mounted using cloud-init only when the EC2 instance is built for the first time.

The mijin data is stored in block, the API node stores the block and the mongo data used by rest, but can restore the mongo data if there is data in block.



The EC2 instance is started using a custom AMI with the mijin package installed. UserData runs cloud-init and executes the initial package configuration. It receives the results of the UserData execution and rolls back if the configuration fails.

1. Unix user settings specified by parameter
2. Host Name Settings
3. OS package updates
4. Installing pip
5. Installing cloudformation helper script
6. **Setup of mijin**
    1. EBS Format
    2. mijin setup (api1 uploads data to parameter store)

In addition, the order in which instances are created is as follows

1. ApiInstance1
2. ApiInstance2 PeerInstanceX Concurrency

At ApiInstance1, create the configuration to be used for all nodes and store the data in the AWS Systems Manager parameter store. Other instances will retrieve data from this parameter store and create mijin.
The contents stored in the parameter store are as follows

Parameter name	Description
/Crown name specified at deploy time/shares/api_node.json	Public key used by the API node
/Crown name specified at deploy time/shares/generation_hash.json	mijin Catapult(v.2) blockchain's Genesis Hash (GenerationHash)
/Crown name specified at deploy time/shares/harvest_fee_sink_public_key.json	Address to receive Harvest (not required for mijin)
/Crown name specified at deploy time/shares/init_host_count.json	Number of nodes created during deployment
/Crown name specified at deploy time/shares/mosaic_rental_fee_sink_public_key.json	Address to receive Mosaic rental fees
/Crown name specified at deploy time/shares/namespace_rental_fee_sink_public_key.json	Address to receive Namespace rental fees
/Crown name specified at deploy time/shares/nemesis_addresses.json	Empty address not used in particular (can be used)
/Crown name specified at deploy time/shares/nemesis_addresses_harvesting.json	Address to receive harvest, etc.
/Crown name specified at deploy time/shares/nemesis_addresses_harvesting_cosig.json	Address used for authorization to finalize
/Crown name specified at deploy time/shares/nemesis_addresses_harvesting_vrf.json	Address for enhanced security (to obscure the state in which blocks can be generated)
/Crown name specified at deploy time/shares/peer_node.json	Public key used by the PEER node
/Crown name specified at deploy time/shares/rest_gateway_private_key.json	Address for REST used by API node
/Crown name specified at deploy time/shares/signer_private_key.json	Address to sign Nemesis (Genesis) block
/Crown name specified at deploy time/shares/new-cert/each node/CA/[*].pem	SSL certificate to encrypt communications between nodes

Note:

The values in this parameter store are stored as the values to be created in the first block of the blockchain and are not called from the parameter store after deployment.
Therefore, if you want to delete data as security, etc., you can delete this data.
In addition, if there is a failure or a need for a new expansion, etc., the data can be recovered from this data.

loadbalanceNestStack

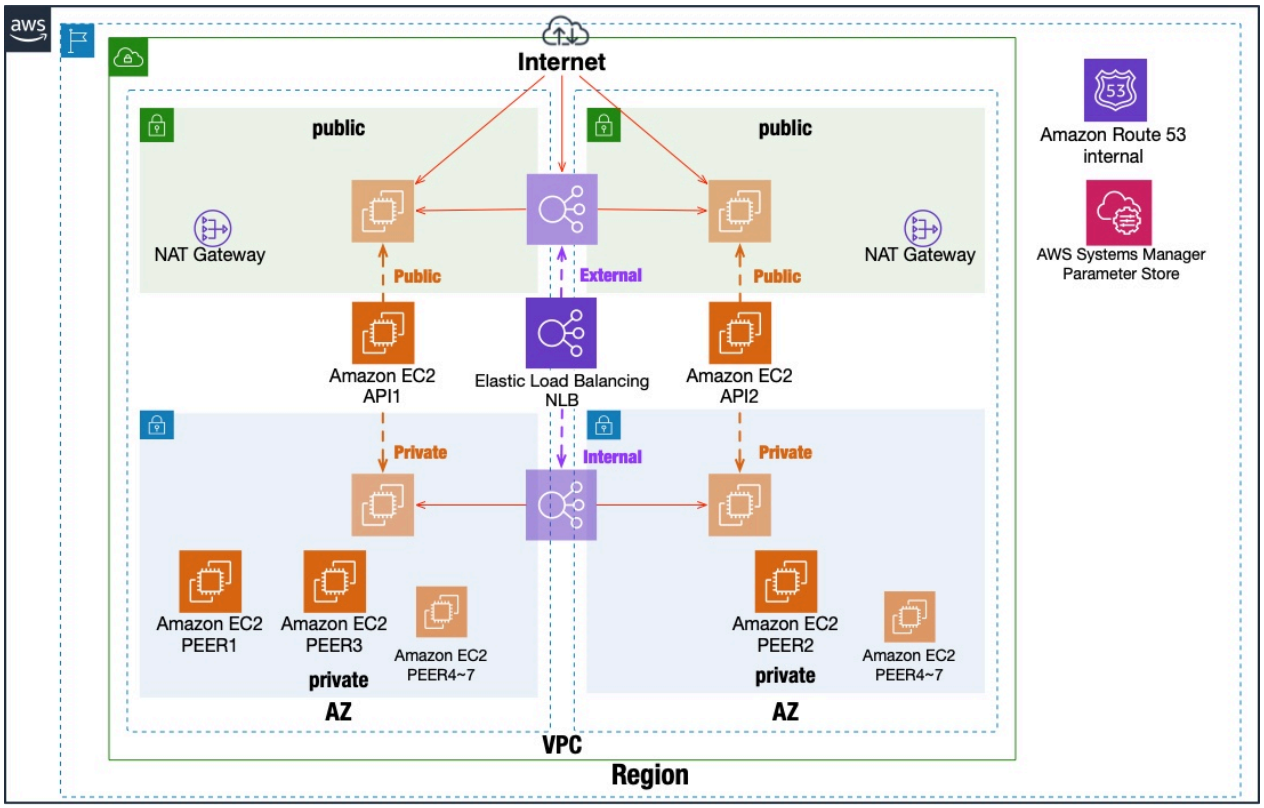
loadbalanceNestStack creates an ELB (load balancer).
If the parameter UseLoadBalancer is No, the stack (empty-elb) is called without creating the ELB.
The ELB distributes to the REST access port 3000 port of the API node.
The ELB Type uses NLB (Network Load Balancer) and the connection to the API node is configured to use the same node for a certain period of time for the same session by sticky session.

Note:

ELBs are created with Network Load Balancer (NLB) only.
To understand the NLB, please refer to the following
https://docs.aws.amazon.com/ja_jp/elasticloadbalancing/latest/network/introduction.html

```
TargetGroupAttributes:
  - Key: stickiness.enabled
    Value: 'true'
```

NLBs can be parameterized for inward placement for private use only or outward placement for connection via the Internet.



Health checks are performed on the REST access port 3000 of the API node under the following conditions
Health checks are monitoring http://API-NODE:3000/chain/info for dead/ alive.

```
Properties:
  HealthCheckIntervalSeconds: 10 # 10 秒間隔でチェックする Check at 10-second intervals.
  UnhealthyThresholdCount: 3 # 異常とみなす回数 Number of times considered abnormal
  HealthyThresholdCount: 3 # 正常とみなす回数 Number of times considered normal
  HealthCheckPath: /chain/info # ヘルスチェックをする URL URL for health check
  HealthCheckProtocol: HTTP
  Port: 3000 # ヘルスチェックポート health check port
```

2.2.5.3 AWS MarketPlace mijin Catapult(v.2) Recovery Strategy with Architectural Patterns

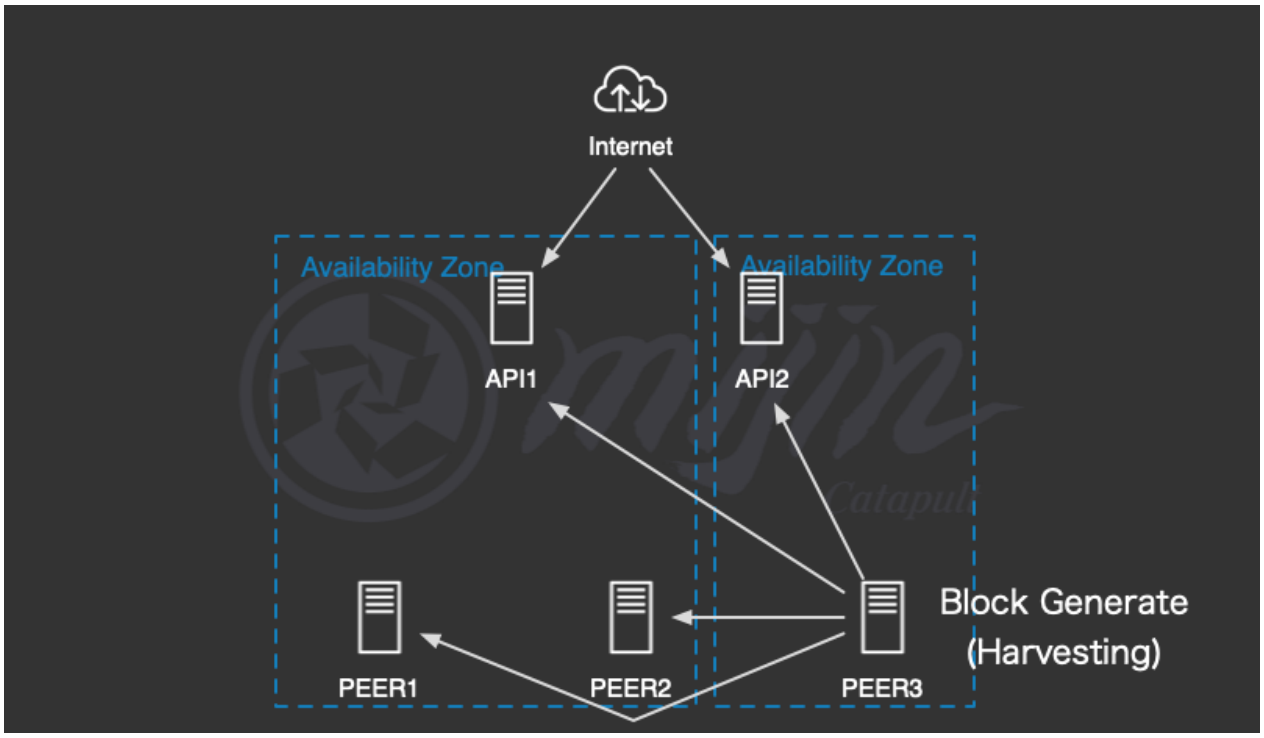
Deployed in AWS MarketPlace, mijin Catapult(v.2) has various architectural patterns depending on the parameters used for deployment.
In this chapter, you will learn architectural patterns and describe recovery strategies through disaster recovery and other BCP measures.

mijin Catapult(v.2) high availability and fault tolerance

mijin Catapult(v.2) keeps the same blockchain data on all nodes, so as long as there is at least one PEER node, the blockchain will continue to update. (Decentralized Fault Tolerance)
Blockchain is characterized by the fact that data is not stored in units of records, such as RDB, but is stored collectively in units of blocks.
Therefore, depending on the circumstances of the failure, the most recent blockchain data may not always be the most current and correct data.
The reason is that the block data of the blockchain is generated in the approximate number of seconds set during deployment, and is divided into nodes that generate blocks (harvesting nodes) and nodes that receive the generated blocks.
The nodes that create blockchain data are pruned by a consensus algorithm (Proof-of-Stake Plus), and the nodes that generate blocks (harvesting nodes) are pruned by a consensus algorithm (Proof-of-Stake Plus).
To learn more about the consensus algorithm, see Symbol's Document.

https://docs.symbol.dev/ja/concepts/consensus-algorithm.html

Although rewards through the block generation mechanism exist in mijin, they are not required in a private blockchain, so the reward process is only running functionally.
The following diagram shows the blockchain being generated by PEER3 and the blockchain data being sent to each node.

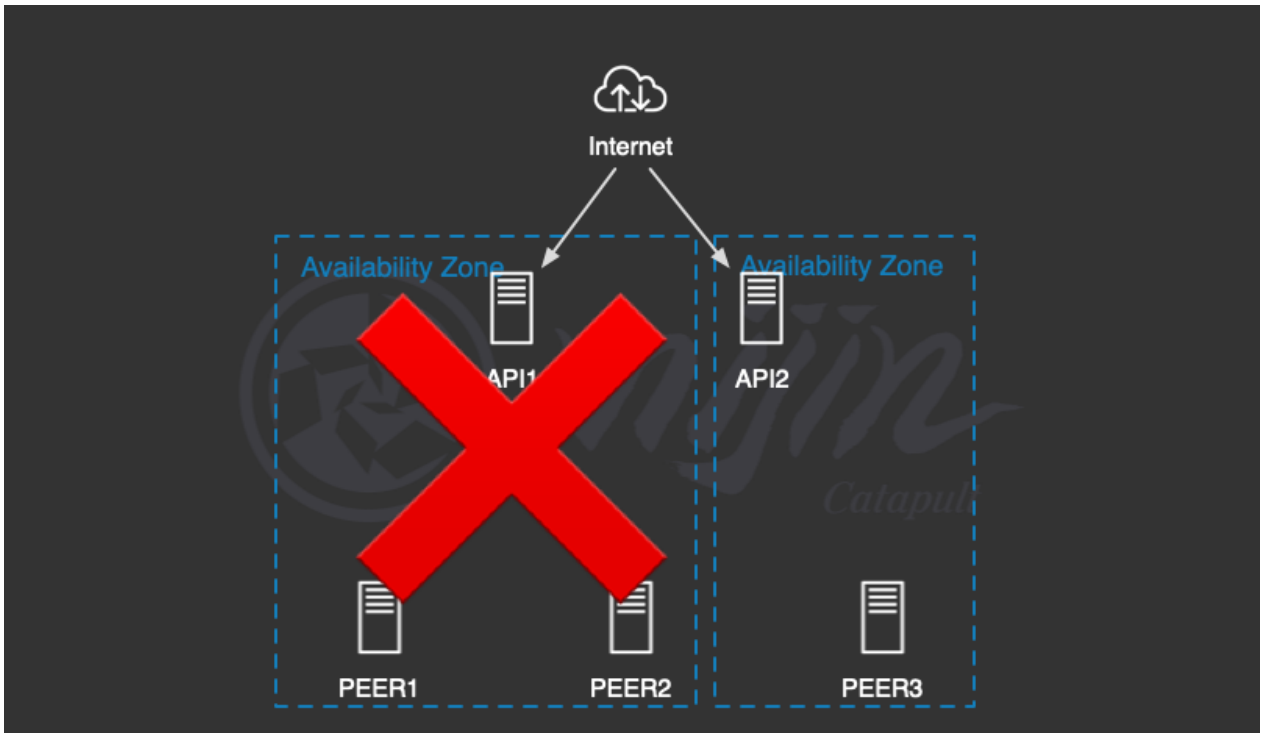


If a failure occurs in a situation where only PEER2 remains without receiving the block data generated from PEER3, there may be a difference of at least one block.
At that time, PEER2 cannot recognize PEER1 and PEER3, so it generates new blockchain data and turns into an independent node.
However, if PEER1 and PEER3 are recovered immediately, PEER3 with the correct block data is rolled back as positive and recovered to the normal blockchain data.
If all the API nodes are down, they cannot be accessed by programs, etc., but block generation will proceed as long as the PEER nodes are present.

Multi-AZ Recovery Strategy

Recovery Time Objective (RTO)	Almost 0
Recovery Point Objective (RPO)	Almost 0

Deploying in Marketplace mijin Catapult(v.2) deploys each node in a multi-AZ environment.
Even if one AZ side fails, the distributed fault tolerance of mijin Catapult(v.2) allows the service to continue.
If you want to keep the connection to the API node, enable Elastic Load Balance to improve availability.



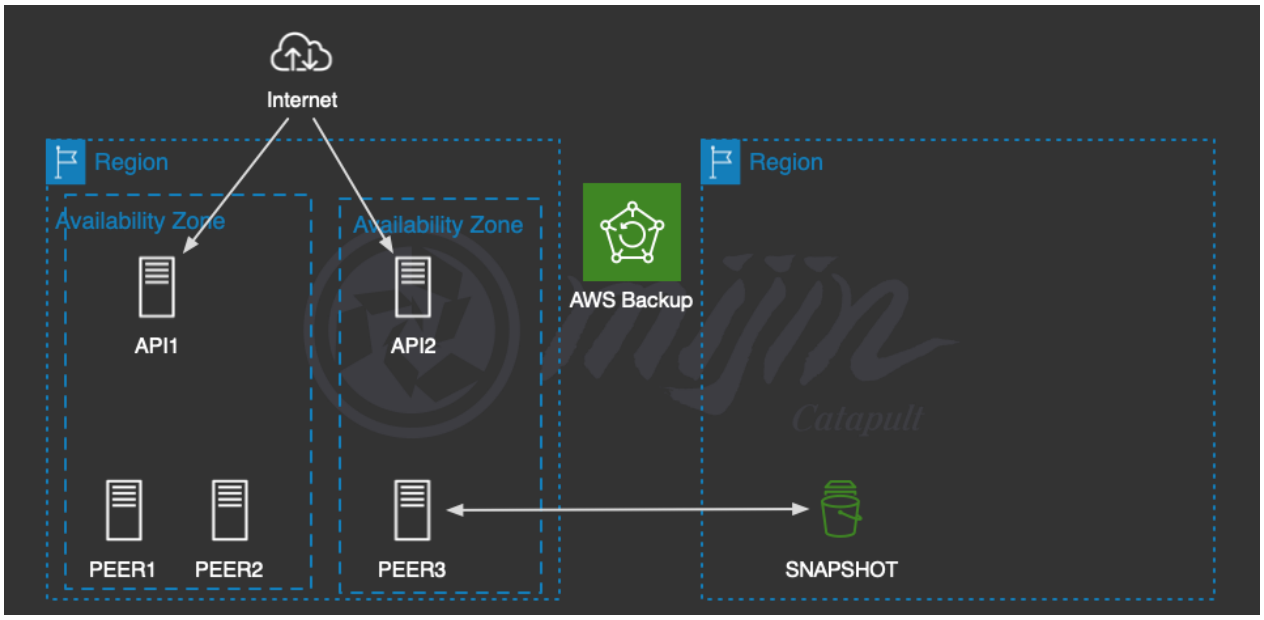
Multi-region and recovery strategies

Inter-region backup

Recovery Time Objective (RTO)	one day
Recovery Point Objective (RPO)	2 hours

All of the mijin Catapult(v.2) deployed in Marketplace can be recovered by backing up the blockchain data from one of the nodes.
Using AWS Backup, blockchain data on a node can be easily backed up to another region.
For backup instructions, see . /aws_tips_ebs_backup.
See below for information on how to restore until backed up.

https://docs.aws.amazon.com/ja_jp/aws-backup/latest/devguide/restore-resource.html



Consortium chain with active/active

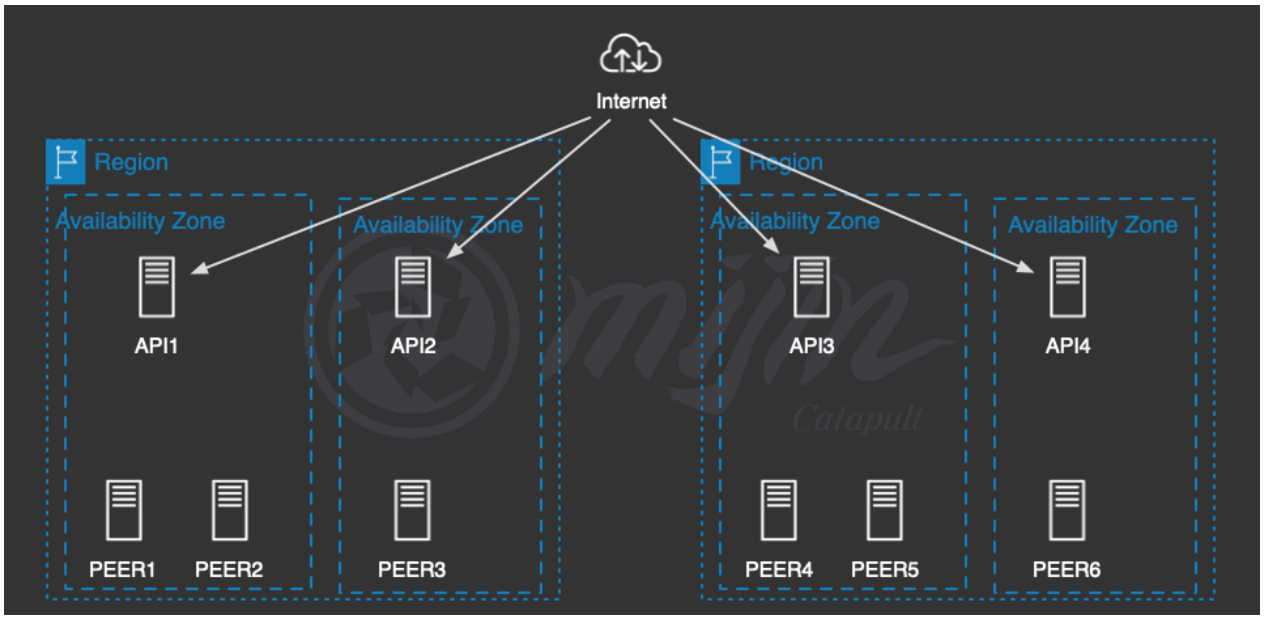
Recovery Time Objective (RTO)	Almost 0
Recovery Point Objective (RPO)	Almost 0

Warning:

Manually build mijin in a separate region to achieve **0 downtime** by realizing a consortium chain. The following diagram shows an example, but since the construction procedure varies depending on the environment, we can provide paid support.

https://mijin.io/aws_contact/

The mijin Catapult(v.2) deployed on Marketplace can be located in 21 regions around the world, but there is no mechanism for building nodes across regions.
By manually building mijin nodes outside of the main region, a multi-region disaster recovery strategy can be established.
The API nodes are also installed together to form a consortium chain, which is an active/active configuration even between regions.



2.2.6 mijin Catapult(v.2) AWS configuration after deployment

This chapter describes how to configure the mijin Catapult(v.2) provided by AWS MarketPlace after deploying it.

2.2.6.1 mijin Catapult(v.2) EC2 instance login how to

This chapter describes how to log in to a node on mijin Catapult(v.2) on AWS.

mijin Catapult(v.2) is running on a Linux server, so this is the procedure for Linux login method. However, since AWS has 'Session Manager'that allows easy remote login from the management console, we set up remote login with Session Manager when deploying.
To learn more about Session Manager, please see below.
https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/session-manager.html

Note:

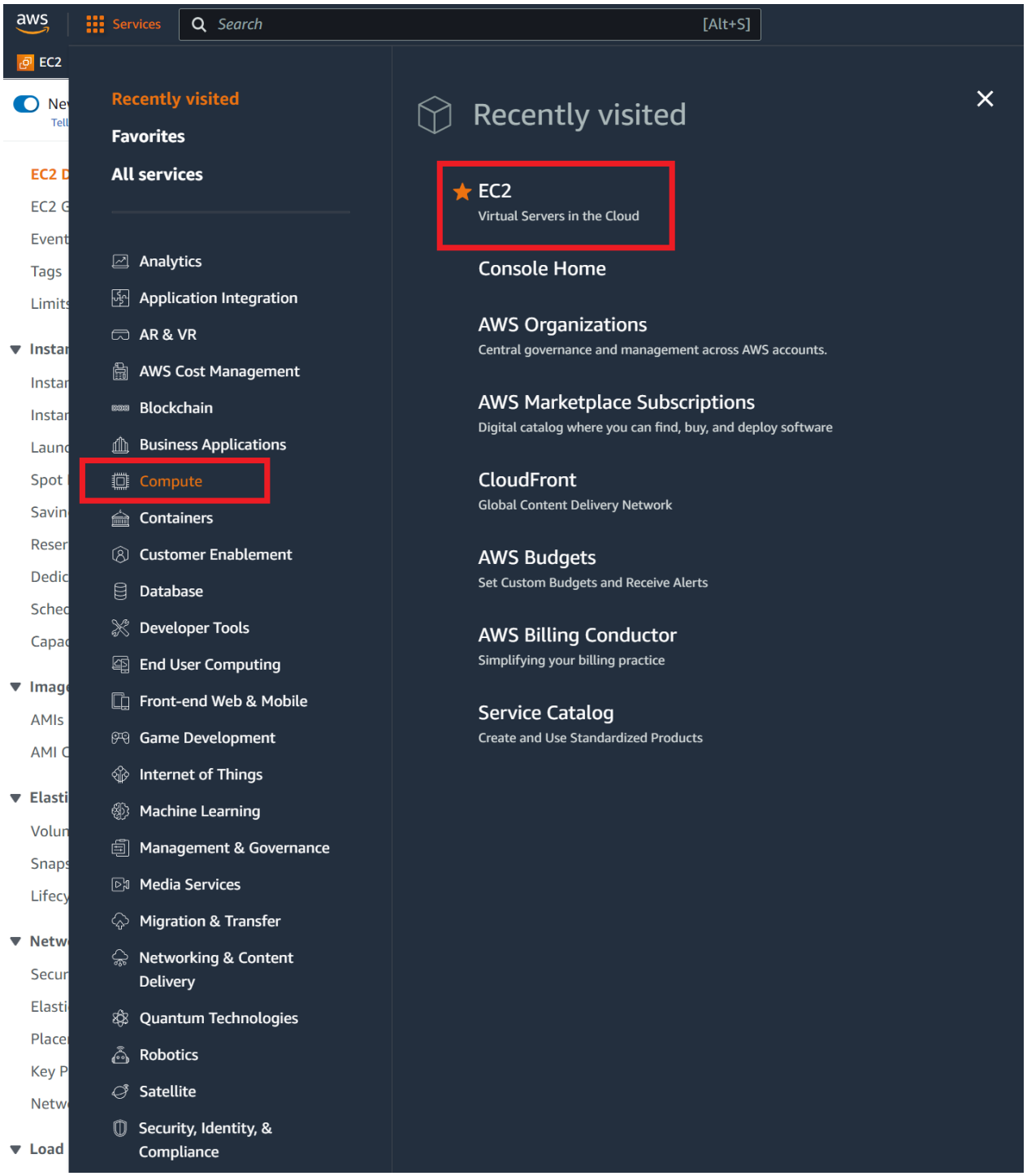
This chapter is an example of the login procedure to EC2 when deploying mijin Catapult(v.2) in AWS MarketPlace.
The connection can be established by changing the security group, etc. using SSH or other conventional remote login methods.

Log in to AWS Management Console

Log in at the AWS Managed Console
https://aws.amazon.com/jp/console/

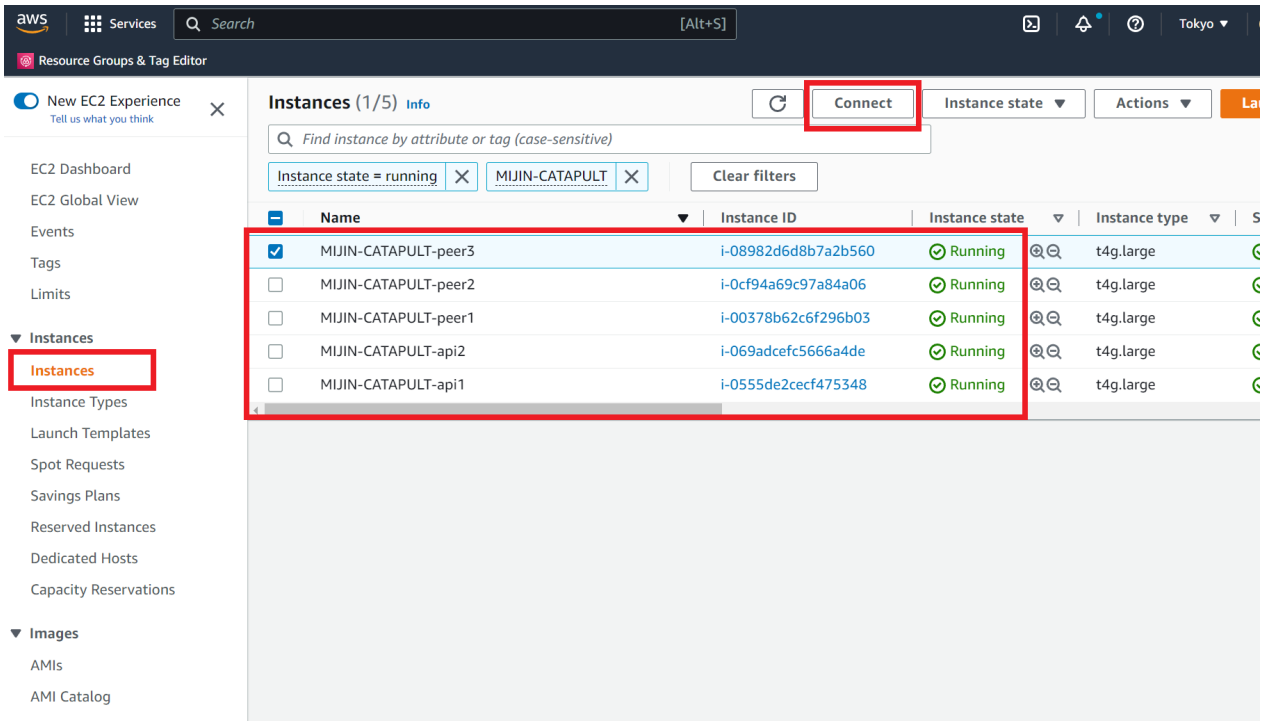
Move to EC2 Service

1. Click on ‘Services’at the top
2. Click on ‘Compute’from the menu that appears
3. Click on ‘EC2.’


aws
Services
Search
[Alt+S]
EC2
Recently visited
Favorites
All services
Analytics
Application Integration
AR & VR
AWS Cost Management
Blockchain
Business Applications
Compute
Containers
Customer Enablement
Database
Developer Tools
End User Computing
Front-end Web & Mobile
Game Development
Internet of Things
Machine Learning
Management & Governance
Media Services
Migration & Transfer
Networking & Content Delivery
Quantum Technologies
Robotics
Satellite
Security, Identity, & Compliance
Recently visited
EC2
Virtual Servers in the Cloud
Console Home
AWS Organizations
Central governance and management across AWS accounts.
AWS Marketplace Subscriptions
Digital catalog where you can find, buy, and deploy software
CloudFront
Global Content Delivery Network
AWS Budgets
Set Custom Budgets and Receive Alerts
AWS Billing Conductor
Simplifying your billing practice
Service Catalog
Create and Use Standardized Products

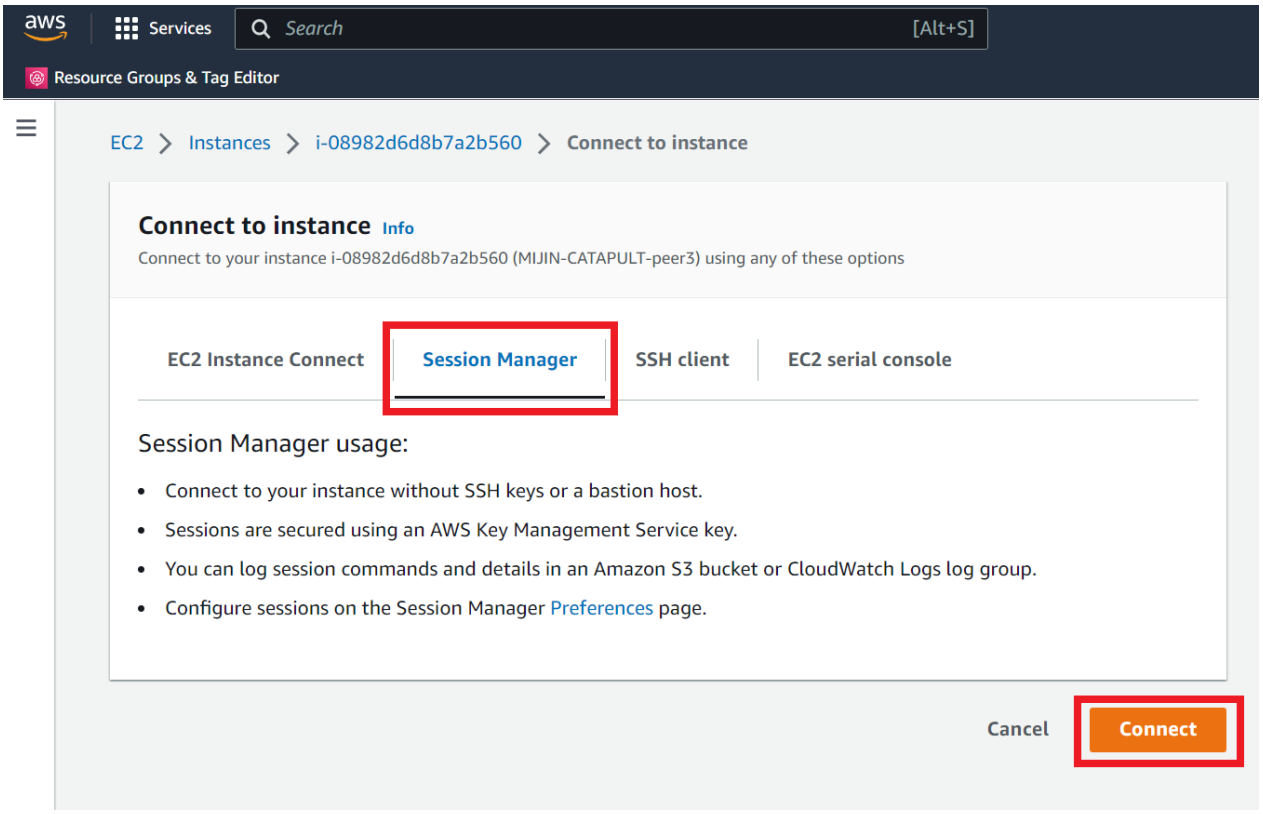
Select the instance you wish to log in to and connect

1. Click on 'Instances'from the left menu
2. From the list of instances, click the check box for the node you wish to log in.
3. Press 'Connect.'



Select Session Manager and connect

1. Make sure it is 'Session Manager.'(If not selected, click on it)
2. Press 'Connect.'
3. A separate console screen window opens.



Console screen operation

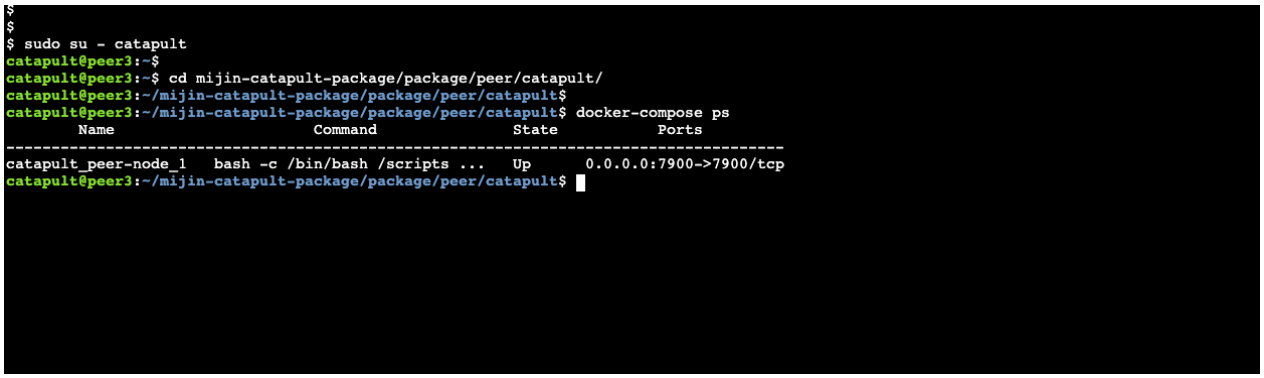
1. Confirm that '$'appears in the console screen window.
2. mijin Catapult(v.2) switch to the UNIX user **catapult** running

```
sudo su - catapult
```

3. Check to see if mijin Catapult(v.2) is working.

```
# PEER ノードに接続した時 (本章では PEER ノード接続時) When connected to a PEER
node (in this chapter, when connected to a PEER node)
cd mijin-catapult-package/package/peer/catapult/
docker-compose ps

# API ノードに接続した時 When connected to an API node
cd mijin-catapult-package/package/api/catapult/
docker-compose ps
```



Note:

mijin Catapult(v.2) runs as one of the containers on docker.
Knowledge of docker is required for operation.
For knowledge of Docker, please refer to the following documents
https://docs.docker.jp/
Official (English)
https://docs.docker.com/get-started/overview/

2.2.6.2 mijin Catapult(v.2) encryption of node storage

This chapter describes the storage encryption used on the nodes of mijin Catapult(v.2) on AWS.

At the time of deployment, the EBS volumes mounted by each node are not encrypted.
To make it more secure, blockchain data and Mongo data can be encrypted.
This section describes the procedure for encrypting the EBS volume that contains mijin Catapult(v.2) using PEER node 3 as an example.

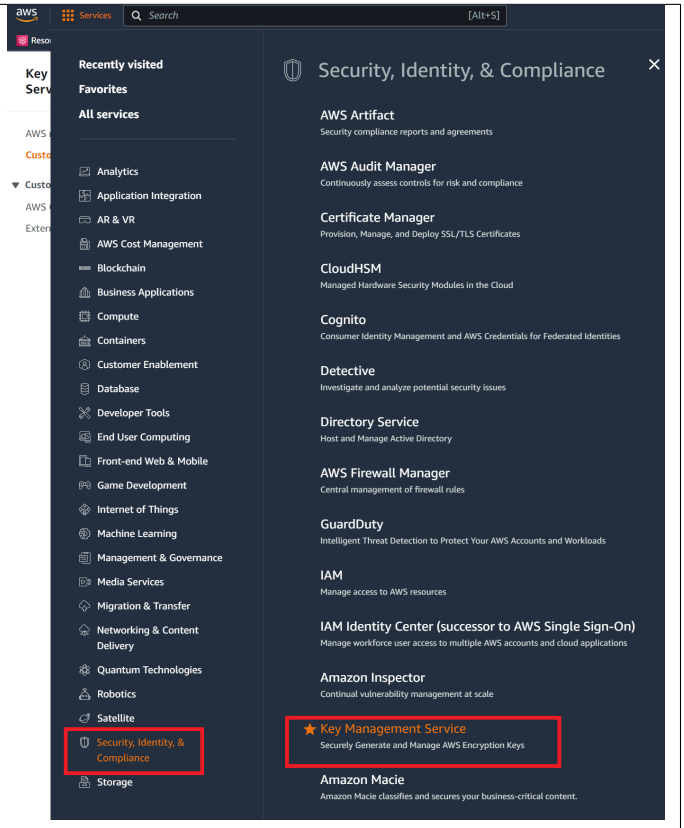
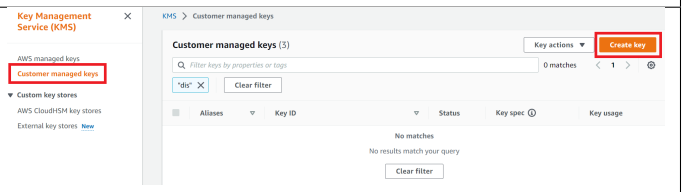
Flow of encrypting a node's blockchain data

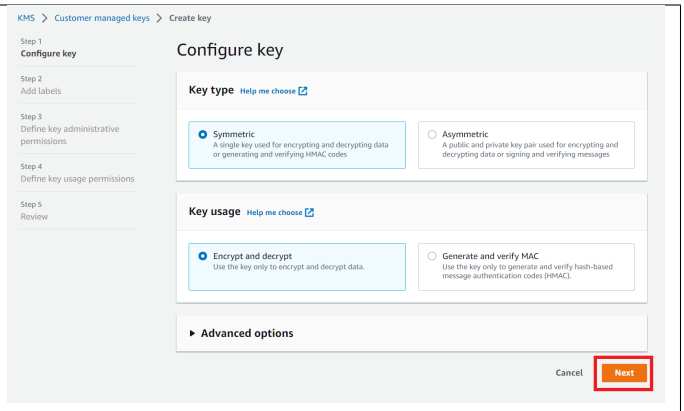
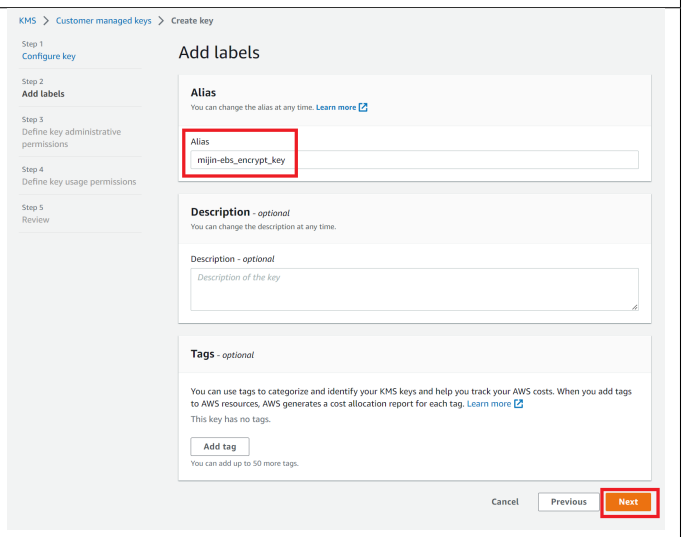
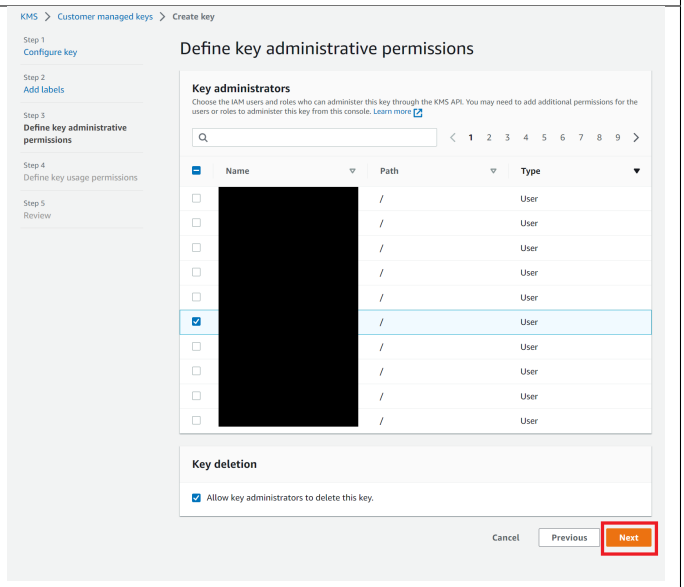
1. Create KMS Key
2. Stop one of the nodes. (In the production version, availability is maintained even if one node is stopped.)
3. Obtain a snapshot of a stopped node
4. Create an encrypted snapshot by copying the snapshot created in 3.
5. Create a volume from the encrypted snapshot created in step 4.
6. Detach the volume of blockchain data on the PEER node stopped in 2.
7. Attach the volume created in step 5 to the PEER node stopped in step 2
8. Start the PEER node stopped in 2.

Note: The same process is followed on the API node, but in the case of the API node, the mongo data is also mounted, so the two volumes can be encrypted.

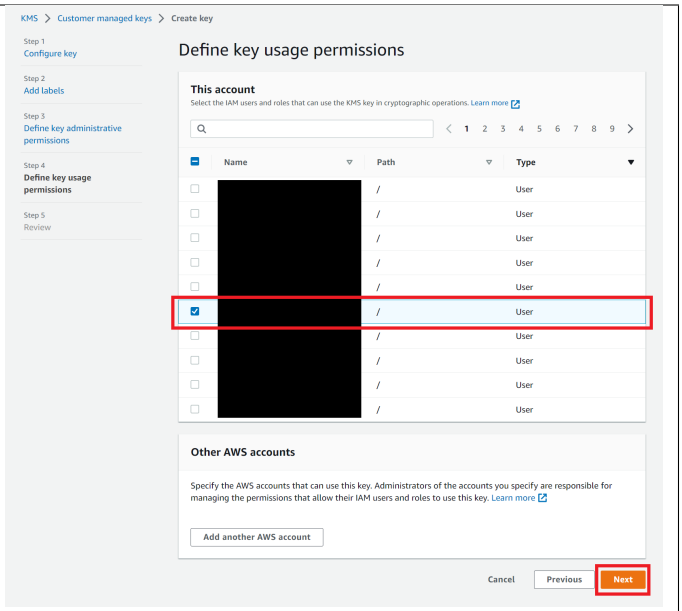
Create KMS Key

To encrypt storage, use KMS to create encryption keys.
To learn more about key creation with KMS, please refer to the following
https://docs.aws.amazon.com/ja_jp/kms/latest/developerguide/create-keys.html

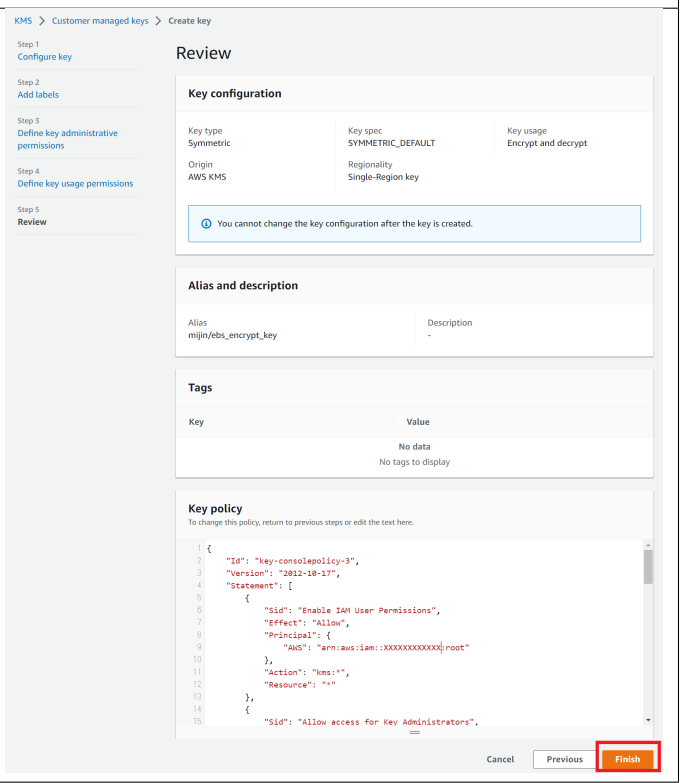
From Services, click Security, Identity, & Compliance, then Key Management Service.	
Click on a ‘Customer managed keys’and click on ‘Create Key’.	

Set any name for the alias and click ‘Next’.	
Specify a key name (alias) and click ‘Next’.	
Select your own account as the key administrator and click ‘Next’. (In this case, specify the account you are currently logged in to)	

Select the account for which you want to use the key and click ‘Next’. (In this case, specify the account you are currently logged in to)



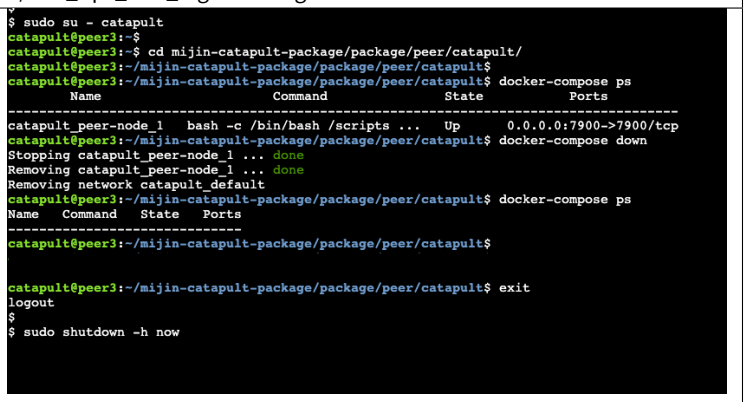
Confirm the values and click ‘Finish’.



Stop one of the nodes

Stopping one PEER or API node will not stop the mijin Catapult(v.2) blockchain network.
Here, we stop at PEER node 3 as an example.

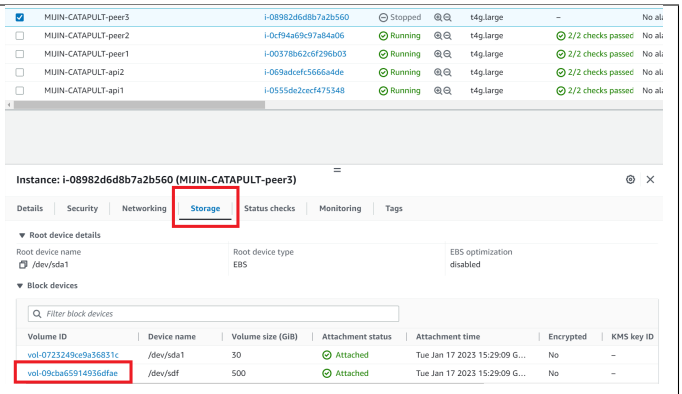
Log in to PEER node 3.	. /aws_tips_ssm_login and log in to the node.

mijin Catapult(v.2) and stop the EC2 instance.
 	
`sudo su - catapult`
`cd mijin-catapult-package/package/peer/catapult/`
`docker-compose down`
`docker-compose ps`
`exit`
`sudo shutdown -h now`
	

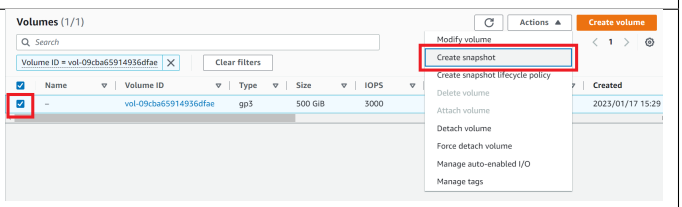
Obtain a snapshot of a stopped node

To create an encrypted volume, you must first create a snapshot of the target volume.

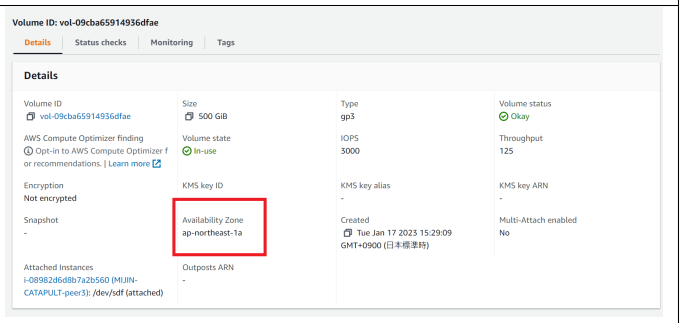
Select the stopped PEER node 3, click the Storage tab, and click the target volumeId.
Here, the volumeId and device name (/dev/sdf) should be noted.



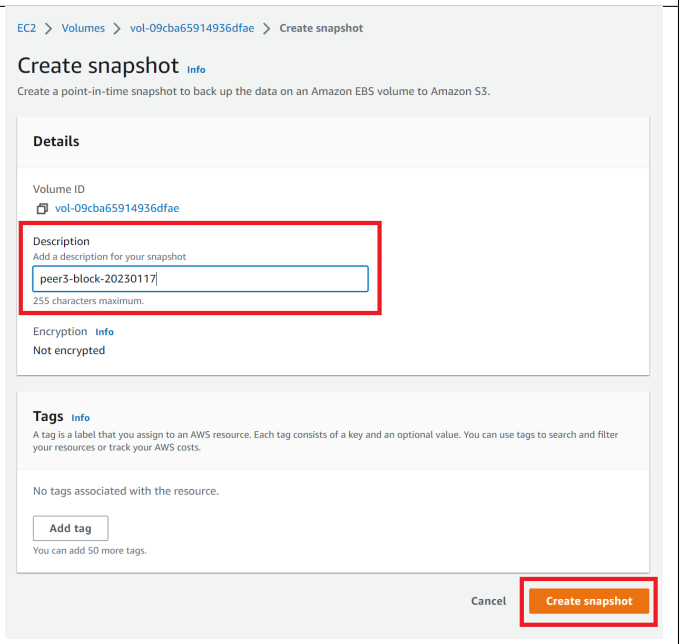
Click the check box for the target volume and click 'Create snapshot'.



We will now check which availability zone this volume belongs to.



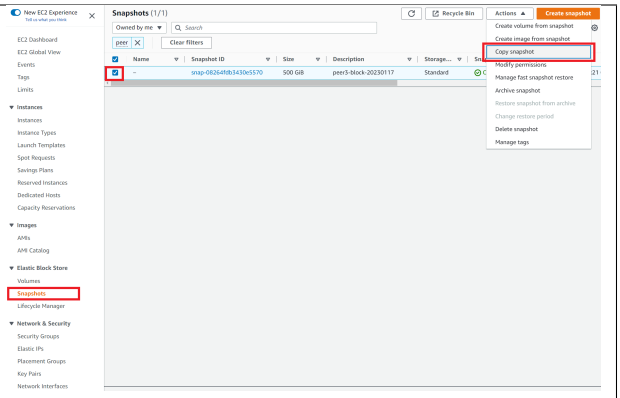
Provide a name in the description that is easy to understand when searching, and click 'Create snapshot'.



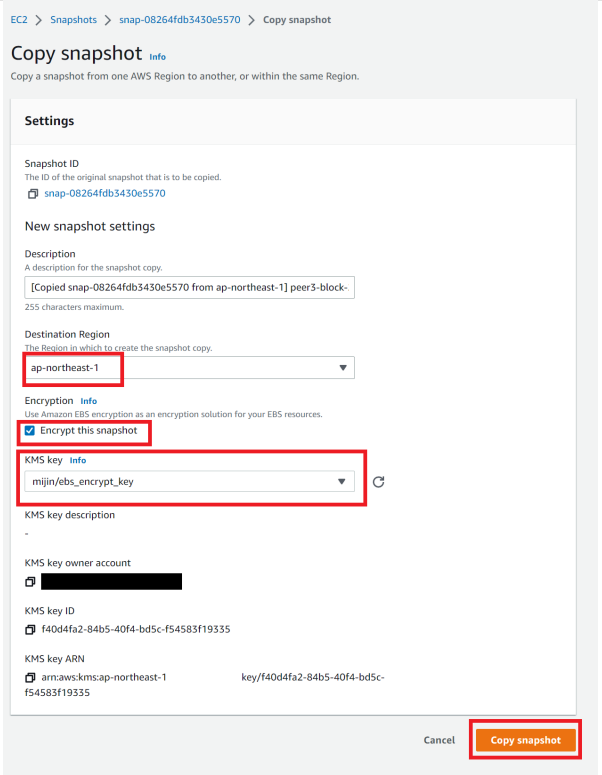
Create an encrypted snapshot by copying the created snapshot

To create encrypted volumes from encrypted snapshots, encryption is performed when the snapshots are copied.

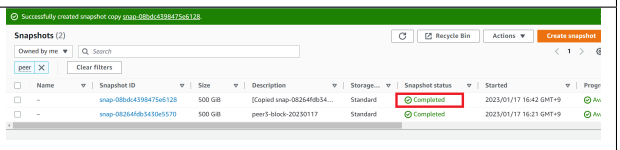
Click Snapshot in the menu, check the snapshot you created, and click ‘Copy snapshot’.



For the destination region, select the same region as the node from which the snapshot was taken.
Check the ‘Encrypt this snapshot’and specify the KMS created.
Click on ‘Copy Snapshot.’



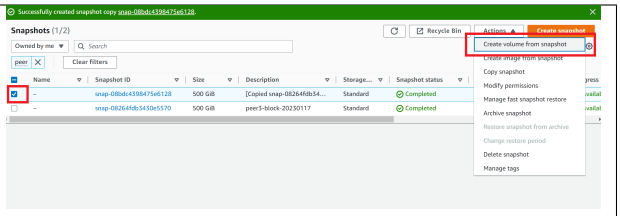
Make sure the snapshot has been completed.



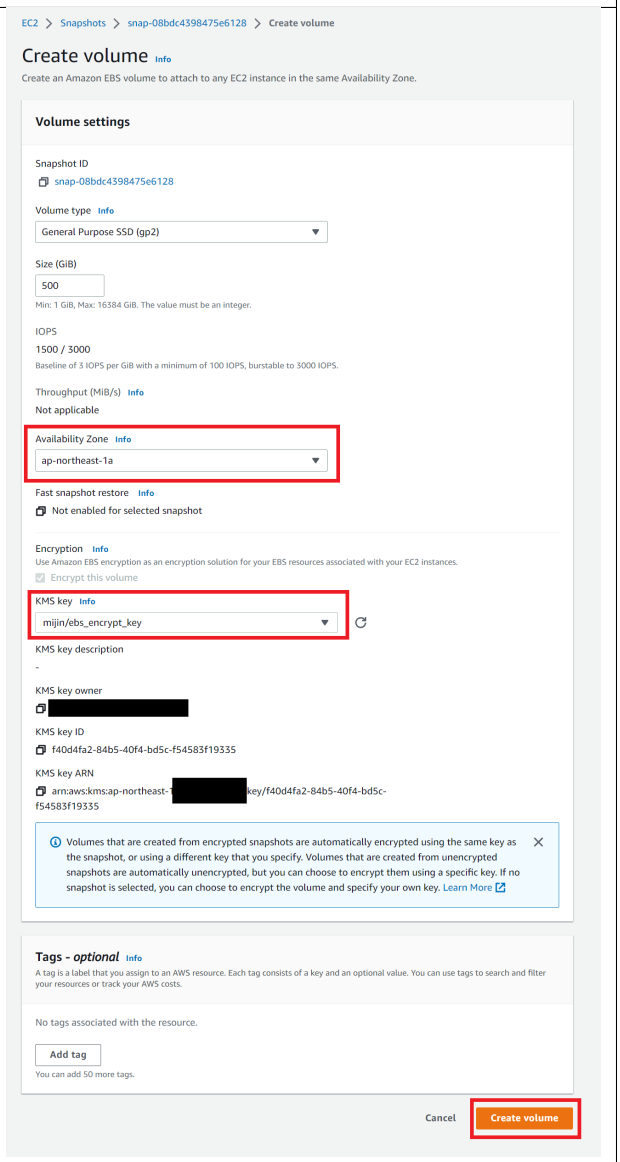
Create a volume from an encrypted snapshot

Create encrypted volumes from encrypted snapshots.
At this point, a volume is created that has not yet been mounted.

Click on the encrypted snapshot and click ‘Create volume from snapshot’.



Select the availability zone as the same availability zone as the node from which the snapshot was taken.
Select the KMS key you created.
Click on ‘Create volume.’

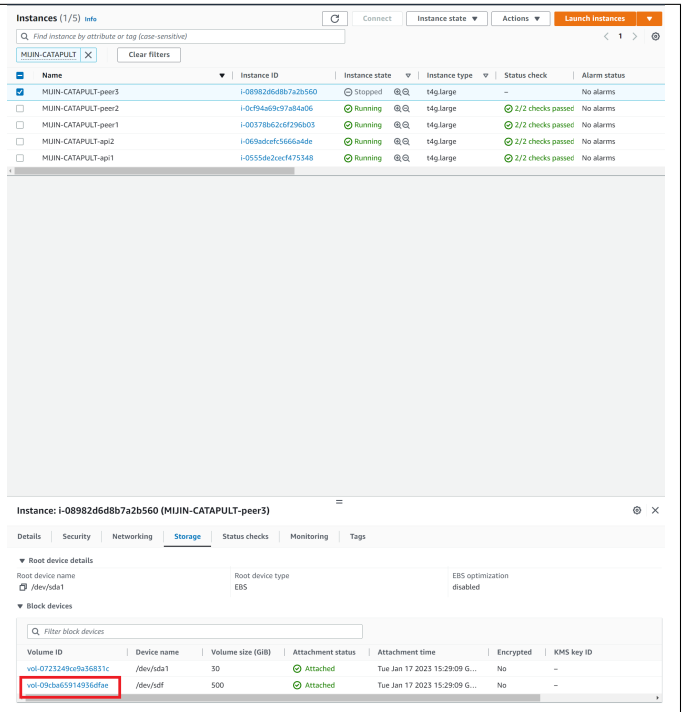
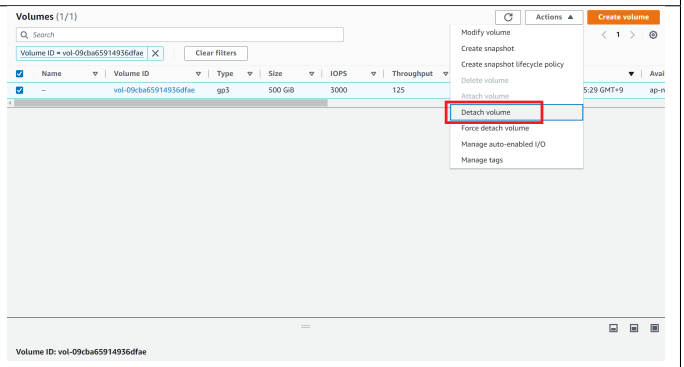


Note:

Note that if you select a different availability zone than the node here, the volume will not appear when attached to the node.
Check Obtain a snapshot of a stopped node for availability zone confirmation of the node.
If you would like to understand availability zones, please refer to the following
https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-availability-zones

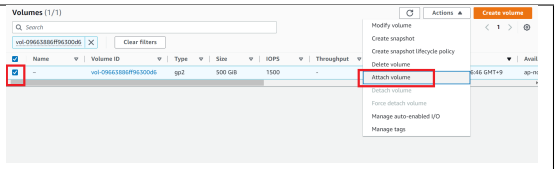
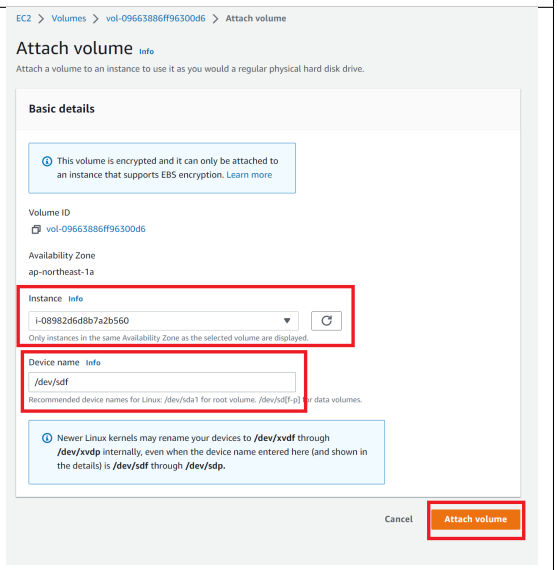
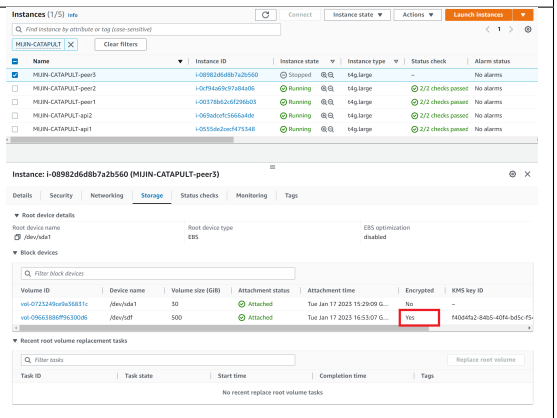
Detach the volume of blockchain data on the stopped PEER node

Remove any unencrypted volumes attached to the node.

Select the volume from PEER node 3 again.	
From Volume, click on ‘Detach volume’and press OK on the screen that pops up.	

Attach encrypted volume to PEER node

Attach the encrypted volume to the node

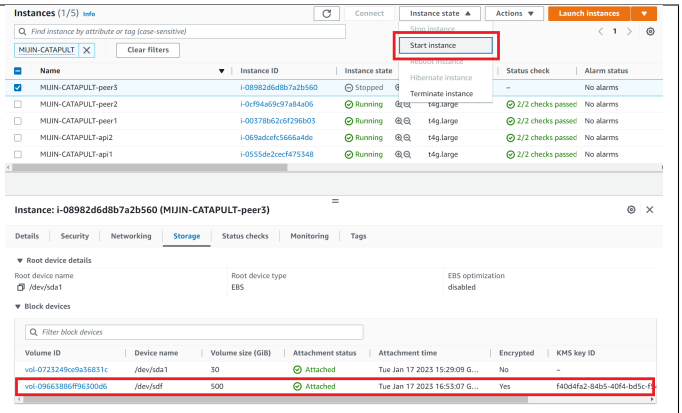
Select the encrypted volume and click ‘Attach volume’.	
Select instance as PEER node 3 and specify the device name you have noted down as the same path as the volume before encryption. Click on ‘Attach volume.’	
Verify that the encrypted volume is attached.	

Note: The device name must be the name of the device noted in Obtain a snapshot of a stopped node.

Start up a stopped PEER node

Start up the stopped node and start up mijin Catapult(v.2) as well.

From the list of instances, check PEER node 3 and click ‘Start Instance’.



Log in to PEER node 3.

. /aws_tips_ssm_login and log in to the node.

Make sure the disk is mounted, and start up mijin Catapult(v.2).

```
df -h
sudo su - catapult
cd mijin-catapult-package/package/
↪peer/catapult/
docker-compose up -d
docker-compose ps
```

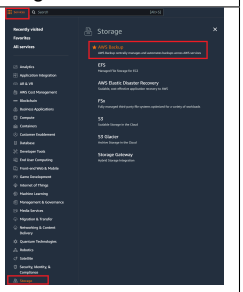
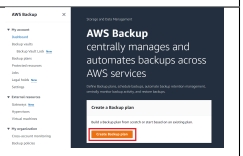
2.2.6.3 mijin Catapult(v.2) periodic backup of nodes

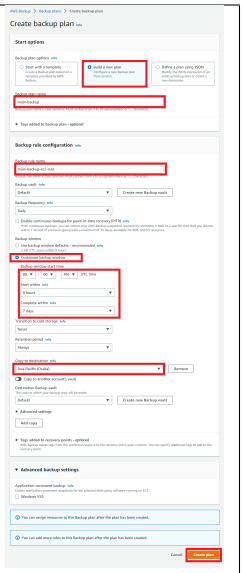
This chapter describes the procedure for backing up data on a node of mijin Catapult(v.2) on AWS.
By backing up your nodes, you can recover from the blockchain data in the event of a region failure, for example.

About AWS Backup Service

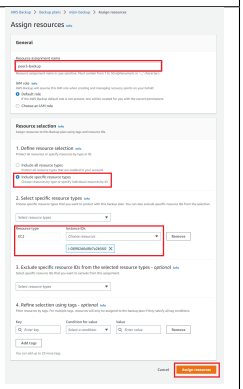
Easily back up the mijin Catapult(v.2) running on your EC2 instance Use 'AWS Backup'to perform backups on a regular basis.
To learn more about AWS Backup, please refer to the following
https://docs.aws.amazon.com/ja_jp/aws-backup/latest/devguide/whatisbackup.html

Creating a Backup Plan

Description	diagram
Click on Services on the top tab. Click Storage. Click on the 'AWS Backup'service.	
「バックアッププランの作成」をクリックします。	



[起動オプション]
Check Build a new plan.
Describe the Backup plan name.
[バックアップルールの設定]
Describe the Backup rule name.
Check Customize backup window.
バックアップの開始時間を設定します（UTC 時間）。
コピー先にコピーで任意のリージョンを指定します（※1）。
「プランの作成」をクリックします。



[全般]
リソース割り当て名を記載します。
[リソース選択]
「特定のリソースタイプを含める」をチェックします。
リソースタイプ：EC2
インスタンス ID：ここでは PEER3 ノードを選択します。
「リソース選択」をクリックします。

Note:

Although you can specify multiple nodes (EC2 instances) to be backed up
mijin Catapult(v.2) can recover everything from blockchain data on a single node, so regular backups

are no problem if only one node is backed up, thus reducing costs.
*1 If the destination region is the same region, it can be recovered at Restore from a backed-up snapshot.

Separate regions can also be used for disaster recovery and other anti-disaster measures.

2.2.6.4 mijin Catapult(v.2) Balance transfer of balance account when commissioned mode is enabled

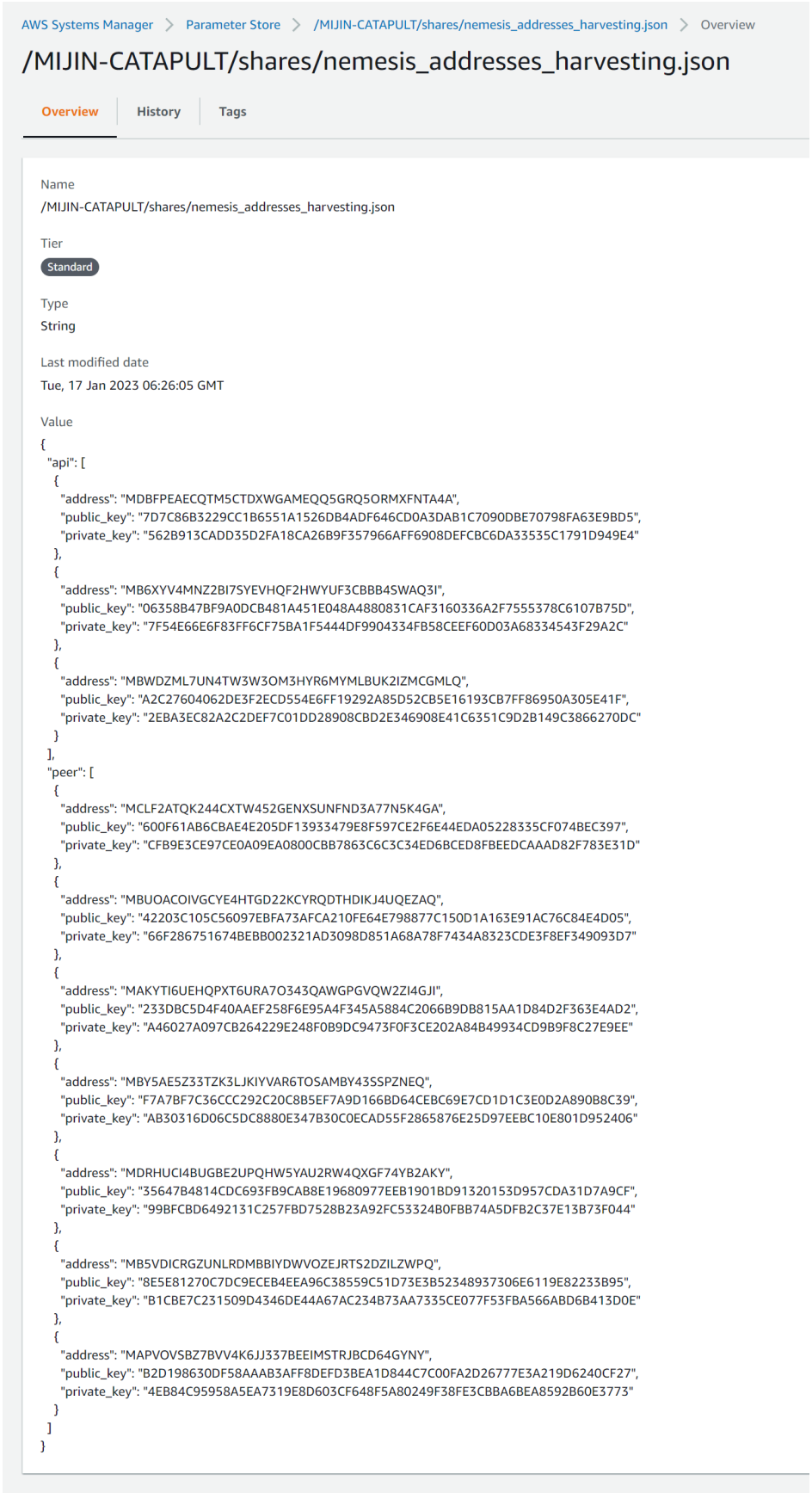
This chapter explains how to move balances from one account with a balance of a mijin Catapult(v.2) node on AWS to another.
The moved account is recommended as the account for the mijin Catapult(v.2) operation.

Note:

In a blockchain, there is always a base currency and transaction fees must be paid in operating the blockchain.
mijin Catapult(v.2) also requires a fee to create a transaction if you enable the with fee mode, and you must operate with an account that has a base currency balance for all transaction fees, Mosaic rental fees, etc.
Conversely, the no-fee mode allows you to create transactions using accounts that do not carry a balance.

Verify accounts with balance

Outstanding accounts are stored in **nemesis_addresses_harvesting.json** in the AWS Systems Manager parameter store.
The list of accounts in this file is all tied to each node.



AWS Systems Manager > Parameter Store > /MIJIN-CATAPULT/shares/nemesis_addresses_harvesting.json > Overview

/MIJIN-CATAPULT/shares/nemesis_addresses_harvesting.json

Overview | History | Tags

Name
/MIJIN-CATAPULT/shares/nemesis_addresses_harvesting.json

Tier
Standard

Type
String

Last modified date
Tue, 17 Jan 2023 06:26:05 GMT

Value


```
{
 "api": [
  {
   "address": "MDBFPEAECQTM5CTDXWGAMEQQ5GRQ5ORMXFNTA4A",
   "public_key": "7D7C86B3229CC1B6551A1526DB4ADF646CD0A3DAB1C7090DBE70798FA63E9BD5",
   "private_key": "562B913CADD35D2FA18CA26B9F357966AFF6908DEFCBC6DA33535C1791D949E4"
  },
  {
   "address": "MB6XYV4MNZ2BI7SYEVHQF2HWYUF3CBBB4SWAQ3I",
   "public_key": "06358B47BF9A0DCB481A451E048A4880831CAF3160336A2F7555378C6107B75D",
   "private_key": "7F54E66E6F83FF6CF75BA1F5444DF9904334FB58CEEF60D03A68334543F29A2C"
  },
  {
   "address": "MBWDZML7UN4TW3W3OM3HYR6MYMLBUK2IZMCGMLQ",
   "public_key": "A2C27604062DE3F2ECD554E6FF19292A85D52CB5E16193CB7FF86950A305E41F",
   "private_key": "2EBA3EC82A2C2DEF7C01DD28908CBD2E346908E41C6351C9D2B149C3866270DC"
  }
 ],
 "peer": [
  {
   "address": "MCLF2ATQK244CXTW452GENXSUNFND3A77N5K4GA",
   "public_key": "600F61AB6CBAE4E205DF13933479E8F597CE2F6E44EDA05228335CF074BEC397",
   "private_key": "CFB9E3CE97CE0A09EA0800CBB7863C6C3C34ED6BCED8FBEEDCAAAD82F783E31D"
  },
  {
   "address": "MBUOACOIVGCYE4HTGD22KCYRQDTHDIKJ4UQEZAQ",
   "public_key": "42203C105C56097EBFA73AFCA210FE64E798877C150D1A163E91AC76C84E4D05",
   "private_key": "66F286751674BEBB002321AD3098D851A68A78F7434A8323CDE3F8EF349093D7"
  },
  {
   "address": "MAKYTI6UEHQPXT6URA7O343QAWGPGVQW2ZI4GJI",
   "public_key": "233DBC5D4F40AAEF258F6E95A4F345A5884C2066B9DB815AA1D84D2F363E4AD2",
   "private_key": "A46027A097CB264229E248F0B9DC9473F0F3CE202A84B49934CD9B9F8C27E9EE"
  },
  {
   "address": "MBY5AE5Z33TZK3LJKIYVAR6TOSAMBY43SSPZNEQ",
   "public_key": "F7A7BF7C36CCC292C20C8B5EF7A9D166BD64CEBC69E7CD1D1C3E0D2A890B8C39",
   "private_key": "AB30316D06C5DC8880E347B30C0ECAD55F2865876E25D97EEBC10E801D952406"
  },
  {
   "address": "MDRHUCI4BUGBE2UPQHW5YAU2RW4QXGF74YB2AKY",
   "public_key": "35647B4814CDC693FB9CAB8E19680977EEB1901BD91320153D957CDA31D7A9CF",
   "private_key": "99BFCBD6492131C257FBD7528B23A92FC53324B0FBB74A5DFB2C37E13B73F044"
  },
  {
   "address": "MB5VDICRGZUNLRDMBBIYDWVOZEJRTS2DZILZWPQ",
   "public_key": "8E5E81270C7DC9ECEB4EEA96C38559C51D73E3B52348937306E6119E82233B95",
   "private_key": "B1CBE7C231509D4346DE44A67AC234B73AA7335CE077F53FBA566ABD6B413D0E"
  },
  {
   "address": "MAPVOVSBZ7BVV4K6JJ337BEEIMSTRJBCD64GYNY",
   "public_key": "B2D198630DF58AAAB3AFF8DEFD3BEA1D844C7C00FA2D26777E3A219D6240CF27",
   "private_key": "4EB84C95958A5EA7319E8D603CF648F5A80249F38FE3CBBA6BEA8592B60E3773"
  }
 ]
}
```

All accounts have balances, but here we will check the following data at the top using mijin-catapult-cli.

```
{
  "address": "MDBFPEAECQTM5CTDXWGAMEQQ5GRQ5ORMXFNTA4A",
  "public_key": "7D7C86B3229CC1B6551A1526DB4ADF646CD0A3DAB1C7090DBE70798FA63E9BD5",
  "private_key": "562B913CADD35D2FA18CA26B9F357966AFF6908DEFCBC6DA33535C1791D949E4"
},
```

Preparation Install nodejs and yarn

Install nodejs to use mijin-catapult-tools. Install nodejs using NodeSource

```
$ curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash - && sudo apt-get
install -y nodejs
$ node --version
$ sudo npm install -g yarn
```

Installing mijin-catapult-tools

Use yarn to install mijin-catapult-tools.

```
$ yarn global add  @tech-bureau/mijin-catapult-tools
$ echo 'export PATH="$HOME/.yarn/bin:$PATH"' >> ~/.bashrc && source ~/.bashrc
```

Import of balance accounts

Import balance accounts.
-u specifies the mijinLBENdpoint or mijinEndpoint URL in the CloudFormation output.
-p specifies the private_key of the balance account.

```
$ mijin-catapult-tools account generate -w aws.json -u http://xxxxxxxxxxxxxxxxxxxx.
↪elb.ap-northeast-1.amazonaws.com:3000 -s -p
562B913CADD35D2FA18CA26B9F357966AFF6908DEFCBC6DA33535C1791D949E4
2023-01-17T07:35:12.266Z [info] : mijin URL: http://xxxxxxxxxxxxxxxxxxxx.elb.ap-
↪northeast-1.amazonaws.com:3000
2023-01-17T07:35:12.268Z [info] : Network: 96
2023-01-17T07:35:12.268Z [info] : Mosaic Currency Id: 769E11974E2CAD76
2023-01-17T07:35:12.269Z [info] : Mosaic Harvest Id: 792022E7945425E4
2023-01-17T07:35:12.269Z [info] : Start Account Generate...
2023-01-17T07:35:12.343Z [info] : Write Config File: aws.json
2023-01-17T07:35:12.345Z [info] : New Account: {
  "url": "http://xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.amazonaws.com:3000",
  "workAccount": {
    "publicKey": "E28BF2A27FE64DF392CBF6D6883BE4858CF26790B4EABC3BCA8E08854BC6A9BF",
    "privateKey": "C3BE65EB9055405ED8CCB7AD568D1368067174F969E9321C0AB4379A7565C9C2",
    "address": "MB2ZQXQQQHOVYU4GX2TKNNJK6XLZPIZV6LK62MY"
  },
```

(continues on next page)

(continued from previous page)


```
  "balanceAccount": {
    "publicKey": "7D7C86B3229CC1B6551A1526DB4ADF646CD0A3DAB1C7090DBE70798FA63E9BD5",
    "privateKey": "562B913CADD35D2FA18CA26B9F357966AFF6908DEFCBC6DA33535C1791D949E4",
    "address": "MDBFPEAECQTM5CTDXWGAMEQQ5GRQ5ORMXFNTA4A"
  },
  "mainAccount": {
    "publicKey": "DAF95081E2D816062108424CF2404B9C3B7C4C7CD1DF6E1446158CC2A2D9B29B",
    "privateKey": "ACF704D53457DF418E1784BACB6D0977B2626BC2847506C7D91B39CDD0515F0F",
    "address": "MD4J2ZVP2AW3BCH6UKZOJNNBGS35DBKTNS4KGVI"
  },
  "keylink": {
    "vrf": {
      "publicKey": "E3D05474D23B57EEFCF953EB7A1AF7A44F9BA338C83900AB0A72927D933CD56A",
      "privateKey": "D0F48B1926ECA6C32C1D3A61AE25A7A483EF47B6DE96B9B4E4970773F904CB73",
      "address": "MBR5NJKKF66GVIOFQKCJOAOTR5KWFE5AKWTVLOQ"
    },
    "voting": {
      "publicKey": "8A615FD6E66CBCB6361FAE0156CB6E22E0932F7157F04849551FCAE9CC6E494F",
      "privateKey": "1615F3484A2123131A624FA065E4B17A0CE916FF6D5A772501FBC83876B3B912",
      "address": "MDWPSE2HLW4HFFTLSECCIMWQPF6IKFPQAI76V3Y"
    }
  },
  "test1Account": {
    "publicKey": "889E1705185A2138F4408D70C28A015536F05A69185392F8C683BB39A0BDB951",
    "privateKey": "8636D903270275A3A6459B41E73E8E7365A6A55ED2437AB25F3321230FD64C35",
    "address": "MBRU2UKYC5C7J6MNQU7F3KYXFVMSIUKUZTPWFHI"
  },
  "test2Account": {
    "publicKey": "24323D2D3594BF0A1993E018571EBD1175BD4B461EF3D15FE4EB09FADAB95834",
    "privateKey": "2A4F1B1E98BF4F91F6FE6D6F90844D9FCA78BCF198028AD8778CC85C409F6B5E",
    "address": "MAWOT4JSX3OYBMNGRM47ZQDDUHCJ3LLQKKO6RKA"
  }
}
```


Verify account information.
Mosaic Id **769E11974E2CAD76** with “currency”: true is the base currency.
Currency with “harvest”: true must be owned by the node entitled to generate the block.

Warning: Note that the Mosaic Id for the base currency is created at the time of deployment and is not the same Id.

```
$ mijin-catapult-tools account info -r aws.json -t balance
2023-01-17T07:36:16.432Z [info] : mijin URL: http://xxxxxxxxxxxxxxxxxxxx.elb.ap-
↪northeast-1.amazonaws.com:3000
2023-01-17T07:36:16.433Z [info] : Network: 96
2023-01-17T07:36:16.433Z [info] : Mosaic Currency Id: 769E11974E2CAD76
2023-01-17T07:36:16.434Z [info] : Mosaic Harvest Id: 792022E7945425E4
2023-01-17T07:36:16.434Z [info] : Start Account Info
2023-01-17T07:36:16.603Z [info] : balance Account: {
  "publicKey": "7D7C86B3229CC1B6551A1526DB4ADF646CD0A3DAB1C7090DBE70798FA63E9BD5",
  "address": "MDBFPEAECQTM5CTDXWGAMEQQ5GRQ5ORMXFNTA4A",
```

(continues on next page)

(continued from previous page)

```
  "mosaics": [
    {
      "id": "769E11974E2CAD76",
      "amount": "1799799999600000",
      "currency": true,
      "harvest": false
    },
    {
      "id": "792022E7945425E4",
      "amount": "3000000",
      "currency": false,
      "harvest": true
    }
  ],
  "keylink": {
    "vrf": {
      "publicKey": "9CD207F9A6DE6D485D350C29B749590251924A29C0EFD8E38DDE24866D71F160"
    },
    "voting": {
      "publicKey": "E3822AA0720F610847E4BE2B740F8FFF9130BEC4E9140BA845150DC2D591D86D",
      "startEpoch": 1,
      "endEpoch": 26280
    }
  }
}
```

Create a new account

Create a new account with no balance.
Move the balance to address **MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ** later.

```
$ mijin-catapult-tools account generate -r aws.json
2023-01-17T07:38:41.738Z [info] : mijin URL: http://xxxxxxxxxxxxxxxxxxxx.elb.ap-
↪northeast-1.amazonaws.com:3000
2023-01-17T07:38:41.740Z [info] : Network: 96
2023-01-17T07:38:41.740Z [info] : Mosaic Currency Id: 769E11974E2CAD76
2023-01-17T07:38:41.740Z [info] : Mosaic Harvest Id: 792022E7945425E4
2023-01-17T07:38:41.740Z [info] : Start Account Generate...
2023-01-17T07:38:41.768Z [info] : New Account: {
  "publicKey": "7437EB45A39AF335F08CABD203503632115CA1793902F5106BC03963C96AEE4F",
  "privateKey": "708AB4973F37B89195340AEA7EBD733ED16AE51B99EB648E7A3885869CBAF3C9",
  "address": "MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ"
```

Query the node for account information.
Verify that this account has a non-existent error because there is no record of a balance in the node.

```
$ mijin-catapult-tools account info -r aws.json -t other -a
MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ
2023-01-17T07:39:44.832Z [info] : mijin URL: http://xxxxxxxxxxxxxxxxxxxx.elb.ap-
```

(continues on next page)

(continued from previous page)

```
↪northeast-1.amazonaws.com:3000
2023-01-17T07:39:44.834Z [info] : Network: 96
2023-01-17T07:39:44.834Z [info] : Mosaic Currency Id: 769E11974E2CAD76
2023-01-17T07:39:44.834Z [info] : Mosaic Harvest Id: 792022E7945425E4
2023-01-17T07:39:44.834Z [info] : Start Account Info
2023-01-17T07:39:45.061Z [error] : Address Not Found
```

Balance transfers

Transfer the balance from the balance account to a new account (MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ).
Here we will transfer 100,000 cat.currency.

Announce a transfer transaction from your balance account (mijin-have-currency-account).

```
$ mijin-catapult-tools transaction transfer -r aws.json -f balance -d
MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ -a 100000
2023-01-17T07:41:40.559Z [info] : mijin URL: http://xxxxxxxxxxxxxxxxxxxx.elb.ap-
↪northeast-1.amazonaws.com:3000
2023-01-17T07:41:40.562Z [info] : Network: 96
2023-01-17T07:41:40.562Z [info] : Start Transfer Account...
2023-01-17T07:41:40.585Z [info] : From Account Address:
MDBFPEAECQTM5CTDXWGAMEQQ5GRQ5ORMXFNTA4A
2023-01-17T07:41:40.585Z [info] : Dest Account Address:
MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ
2023-01-17T07:41:40.585Z [info] : Currecny Amount: 100000000000
2023-01-17T07:41:40.594Z [info] : Start Transfer Transaction...
2023-01-17T07:41:55.775Z [info] : End Transfer Transaction
2023-01-17T07:41:55.775Z [info] : http://xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.
↪amazonaws.com:3000/transactionStatus/
↪EC5FE12DBEFD1DF7DDE2D49287EC4DA1649546BB1EC43DE75641D5D4A7BEE770
2023-01-17T07:41:55.775Z [info] : http://xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.
↪amazonaws.com:3000/transactions/confirmed/
↪EC5FE12DBEFD1DF7DDE2D49287EC4DA1649546BB1EC43DE75641D5D4A7BEE770
```

Verify that the new account has a balance.
The account information that was in error earlier is recognized by mijin Catapult(v.2) and you can confirm that you have a balance.

```
$ mijin-catapult-tools account info -r aws.json -t other -a
MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ
2023-01-17T07:42:26.802Z [info] : mijin URL: http://xxxxxxxxxxxxxxxxxxxx.elb.ap-
↪northeast-1.amazonaws.com:3000
2023-01-17T07:42:26.803Z [info] : Network: 96
2023-01-17T07:42:26.804Z [info] : Mosaic Currency Id: 769E11974E2CAD76
2023-01-17T07:42:26.804Z [info] : Mosaic Harvest Id: 792022E7945425E4
2023-01-17T07:42:26.804Z [info] : Start Account Info
2023-01-17T07:42:26.927Z [info] : get Account: {
```

(continues on next page)

(continued from previous page)


```
  "publicKey": "0000000000000000000000000000000000000000000000000000000000000000",
  "address": "MDJMNVU47CWHTZBMX7B6M6WWT5NEEY4GTG66GLQ",
  "mosaics": [
    {
      "id": "769E11974E2CAD76",
      "amount": "100000000000",
      "currency": true,
      "harvest": false
    }
  ],
  "keylink": {
    "vrf": {
      "publicKey": ""
    },
    "voting": {
      "publicKey": "",
      "startEpoch": "",
      "endEpoch": ""
    }
  }
}
```


2.2.6.5 mijin Catapult(v.2) How to update a node's voting rights file

This chapter describes how to deal with the expiration of node voting rights on a mijin Catapult(v.2) node on AWS.
Voting rights for a node expire approximately 547~3285 days, depending on the block generation interval.

Warning:

When the voting rights file expires, the finalization block stops.
Therefore, if you are using finalize blocks, you must update them.

How to calculate the expiration date of a voting node

The expiration date of a single voting file can be determined by the following formula

Warning: Note that blockGenerationTargetTime (block generation interval) is an approximate value, for example, setting 15s does not mean that a block will be generated in a certain amount of time.

```
(VotingSetGroup * maxVotingKeyLifetime) / ( 60 / blockGenerationTargetTime * 60 * 24)
```

VotingSetGroup and maxVotingKeyLifetime are fixed values, so the value of
blockGenerationTargetTime
The expiration date will vary depending on the value of blockGenerationTargetTime.

```
# blockGenerationTargetTime 10s
(180 * 26280) / (60 /10 * 60 * 24) = 547 days
# blockGenerationTargetTime 15s
(180 * 26280) / (60 /15 * 60 * 24) = 821 days
# blockGenerationTargetTime 60s
(180 * 26280) / (60 /60 * 60 * 24) = 3285 days
```

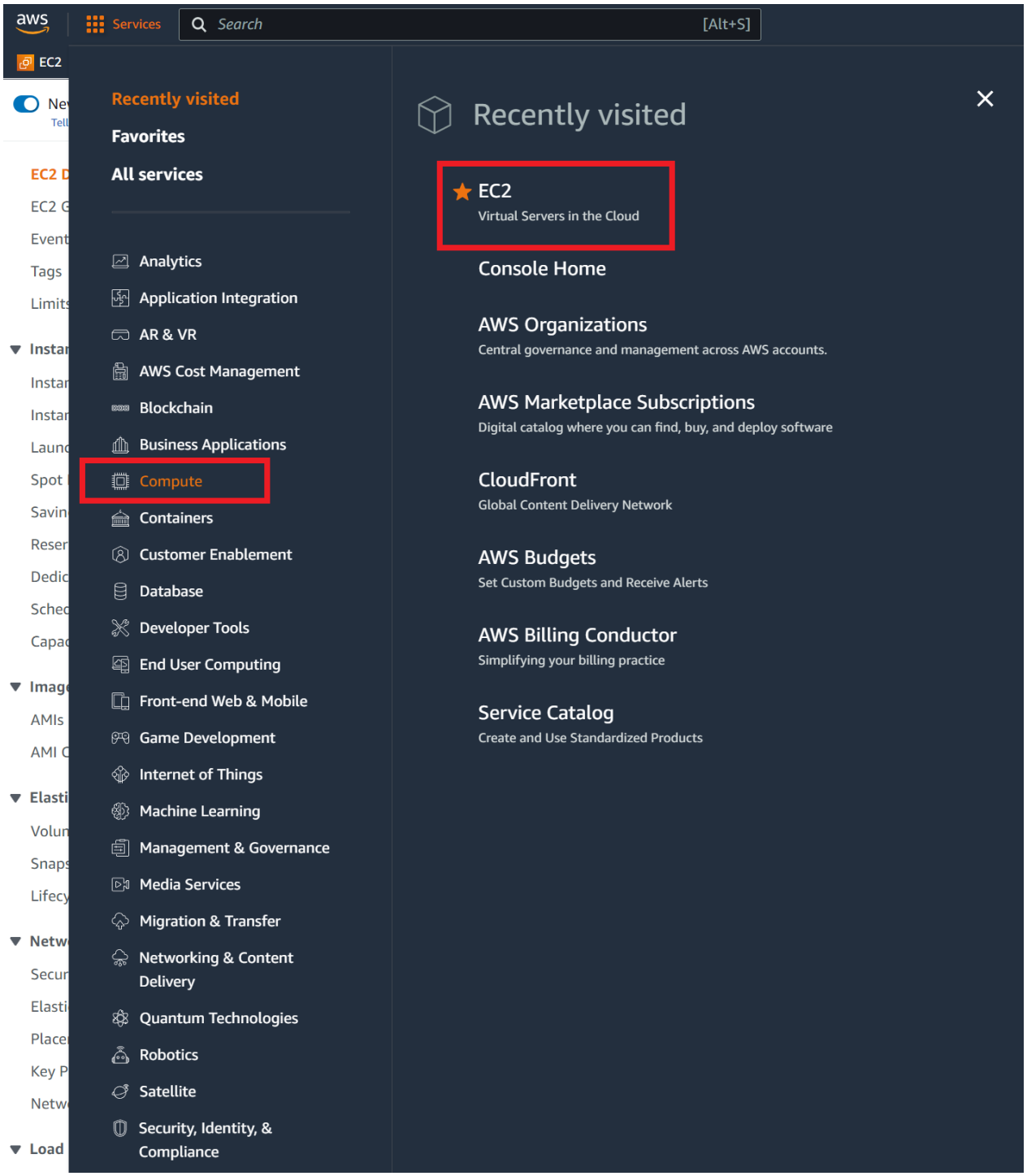
Log in to AWS Management Console

Log in at the AWS Managed Console.

https://aws.amazon.com/jp/console/

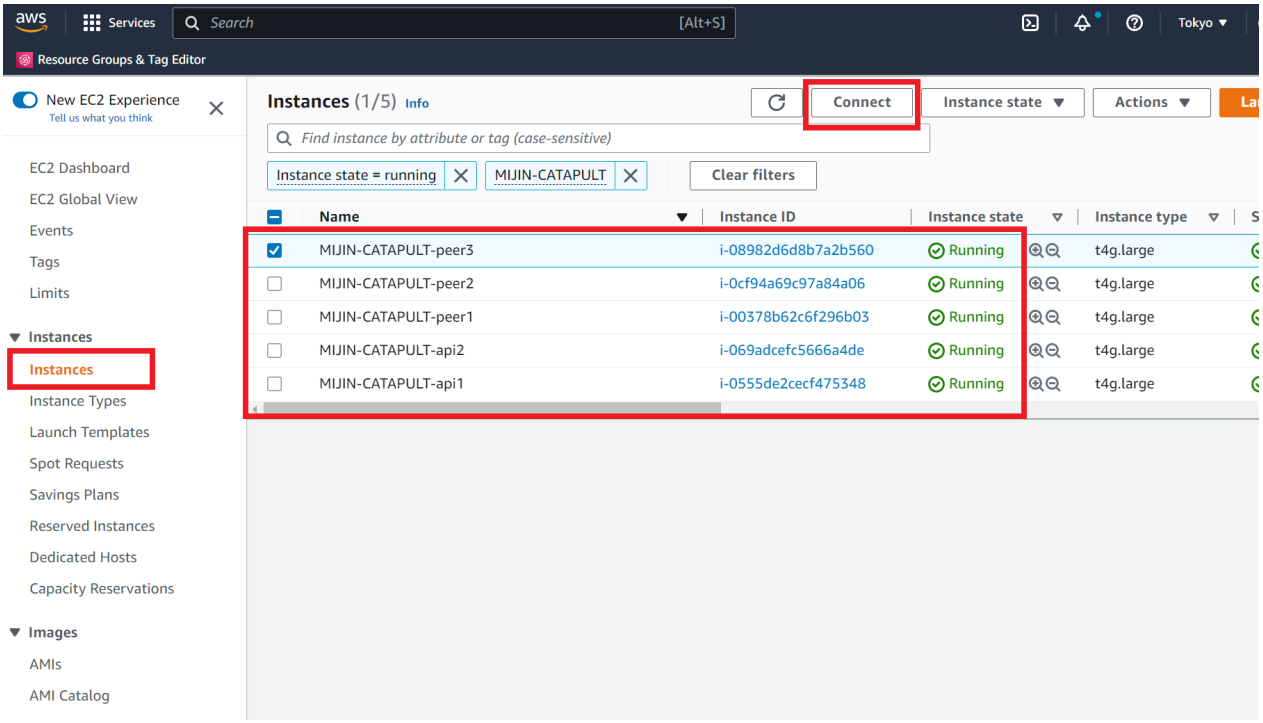
Move to EC2 Service

1. 上部「サービス」をクリックします
2. 表示されたメニューから「コンピューティング」をクリックします
3.「EC2」をクリックします。


aws
Services
Search
[Alt+S]
EC2
Recently visited
Favorites
All services
Analytics
Application Integration
AR & VR
AWS Cost Management
Blockchain
Business Applications
Compute
Containers
Customer Enablement
Database
Developer Tools
End User Computing
Front-end Web & Mobile
Game Development
Internet of Things
Machine Learning
Management & Governance
Media Services
Migration & Transfer
Networking & Content Delivery
Quantum Technologies
Robotics
Satellite
Security, Identity, & Compliance
Recently visited
EC2
Virtual Servers in the Cloud
Console Home
AWS Organizations
Central governance and management across AWS accounts.
AWS Marketplace Subscriptions
Digital catalog where you can find, buy, and deploy software
CloudFront
Global Content Delivery Network
AWS Budgets
Set Custom Budgets and Receive Alerts
AWS Billing Conductor
Simplifying your billing practice
Service Catalog
Create and Use Standardized Products

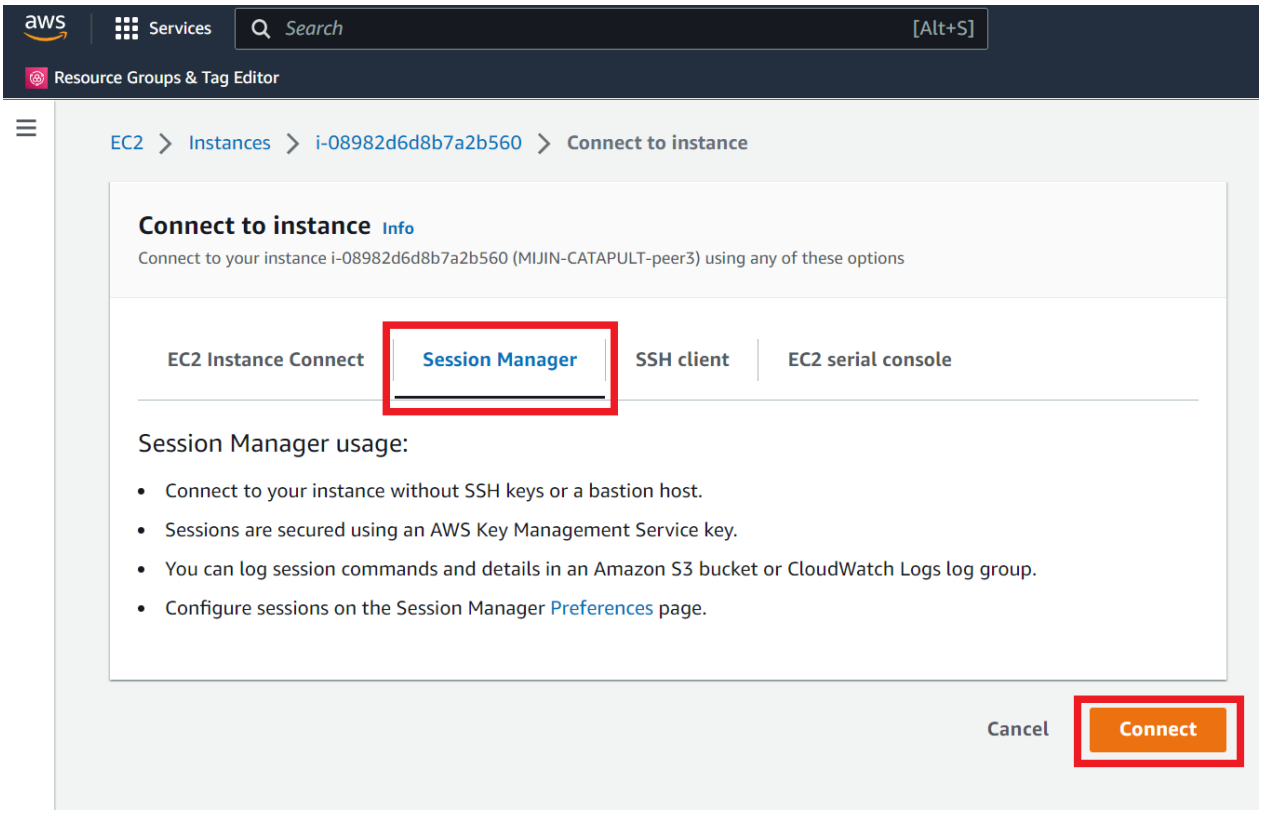
Select the instance you wish to log in to and connect

1. 左側メニューから「インスタンス」をクリックします
2. インスタンス一覧からログインしたいノードのチェックをクリックします。
3.「接続」を押します。



Select Session Manager and connect

1.「セッションマネージャー」であることを確認します。(選択されていなければクリック)
2.「接続」を押します。
3. コンソール画面のウィンドウが別で開きます。



Console screen operation

1. Confirm that '$'appears in the console screen window.

```
$
```

Installing mijin-catapult-tools

1. If nodejs is not installed, install it.

```
$ curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash - && sudo
apt-get install -y nodejs
```

2. Switch to the catapult user and install mijin-catapult-tools.

```
$ yarn global add  @tech-bureau/mijin-catapult-tools
$ echo 'export PATH="$HOME/.yarn/bin:$PATH"' >> ~/.bashrc && source ~/.
↪bashrc
```

Check status of current voting rights file

Refers to the public key.
Here it is **402B6ECE0D1CF99A7F07B832477048C56F213A4F54ED4AEB35AE829507FBC4A6**.
We can confirm that this voting file is valid from 1 to 26280.

```
$ mijin-catapult-tools votingkey info -u http://localhost:3000 -d mijin-
↪catapult-package/package/api/catapult/userconfig/resources/votingkey
2023-01-26T05:22:29.833Z [info] : Start Voting Key Check Dir... mijin-
↪catapult-package/package/api/catapult/userconfig/resources/votingkey
2023-01-26T05:22:29.836Z [info] : votingfile: index0:  {"publicKey":
↪"402B6ECE0D1CF99A7F07B832477048C56F213A4F54ED4AEB35AE829507FBC4A6",
↪"startEpoch":1,"endEpoch":26280,"filename":"private_key_tree1.dat"}
```

Note:

Note that the finalizationEpoch must be updated before it reaches 26280.
$ curl -Ss http://localhost:3000/chain/info | jq -r
{
“scoreHigh”: “0”,
“scoreLow”: “16875391960469924”,
“height”: “310”,
“latestFinalizedBlock”: {
“finalizationEpoch”: 3,
“finalizationPoint”: 7,
“height”: “296”,
“hash”: “AC19CA6C89F87F70470BD84649A31E4FA0E0C5DD71A55E9ADAE25C1AED47882F”
}
}

Voting Rights File Update

The private key is retrieved from the public key obtained above, and a voting rights file is created.
A sequentially numbered file private_key_tree2.dat is created.

```
$ mijin-catapult-tools votingkey update \
    -u http://localhost:3000 \
    -d mijin-catapult-package/package/api/catapult/userconfig/resources/
↪votingkey \
    -p $(cat /mnt/mijin/shares/nemesis_addresses_harvesting_voting.json | jq -
↪r '.[]|.[]|select(.public_key ==
"402B6ECE0D1CF99A7F07B832477048C56F213A4F54ED4AEB35AE829507FBC4A6")|.private_
↪key')

2023-01-26T06:06:16.472Z [info] : Start Voting Key Update...
2023-01-26T06:06:16.476Z [info] : votingSetGroup: 160
2023-01-26T06:06:16.476Z [info] : votingMaxEpoch: 26280
2023-01-26T06:06:16.477Z [info] : votingStartEpoch: 26281
```

(continues on next page)

(continued from previous page)

```
2023-01-26T06:06:16.477Z [info] : votingEndEpoch: 52560
2023-01-26T06:06:16.477Z [info] : blockGenerationTargetTime: 15
2023-01-26T06:06:42.716Z [info] : Voting Key file Create: SUCCESS mijin-
↪catapult-package/package/api/catapult/userconfig/resources/votingkey/
↪private_key_tree2.dat
```

Check status of voting rights file again.

If you see a voting rights file (private_key_tree2.dat) created with the same public key, the update file has been created.
Next time you need to update the finalizationEpoch before it reaches 52560.

```
$ mijin-catapult-tools votingkey info -u http://localhost:3000 -d mijin-
↪catapult-package/package/api/catapult/userconfig/resources/votingkey
2023-01-26T06:10:00.625Z [info] : Start Voting Key Check Dir... mijin-
↪catapult-package/package/api/catapult/userconfig/resources/votingkey
2023-01-26T06:10:00.630Z [info] : votingfile: index0:  {"publicKey":
↪"402B6ECE0D1CF99A7F07B832477048C56F213A4F54ED4AEB35AE829507FBC4A6",
↪"startEpoch":1,"endEpoch":26280,"filename":"private_key_tree1.dat"}
2023-01-26T06:10:00.630Z [info] : votingfile: index1:  {"publicKey":
↪"402B6ECE0D1CF99A7F07B832477048C56F213A4F54ED4AEB35AE829507FBC4A6",
↪"startEpoch":26281,"endEpoch":52560,"filename":"private_key_tree2.dat"}
```

2.2.6.6 [Archive] mijin Catapult(v.2) Balance transfer of balance account when commissioned mode is enabled

This chapter explains how to move balances from one account with a balance of a mijin Catapult(v.2) node on AWS to another.
The moved account is recommended as the account for the mijin Catapult(v.2) operation.

Warning:

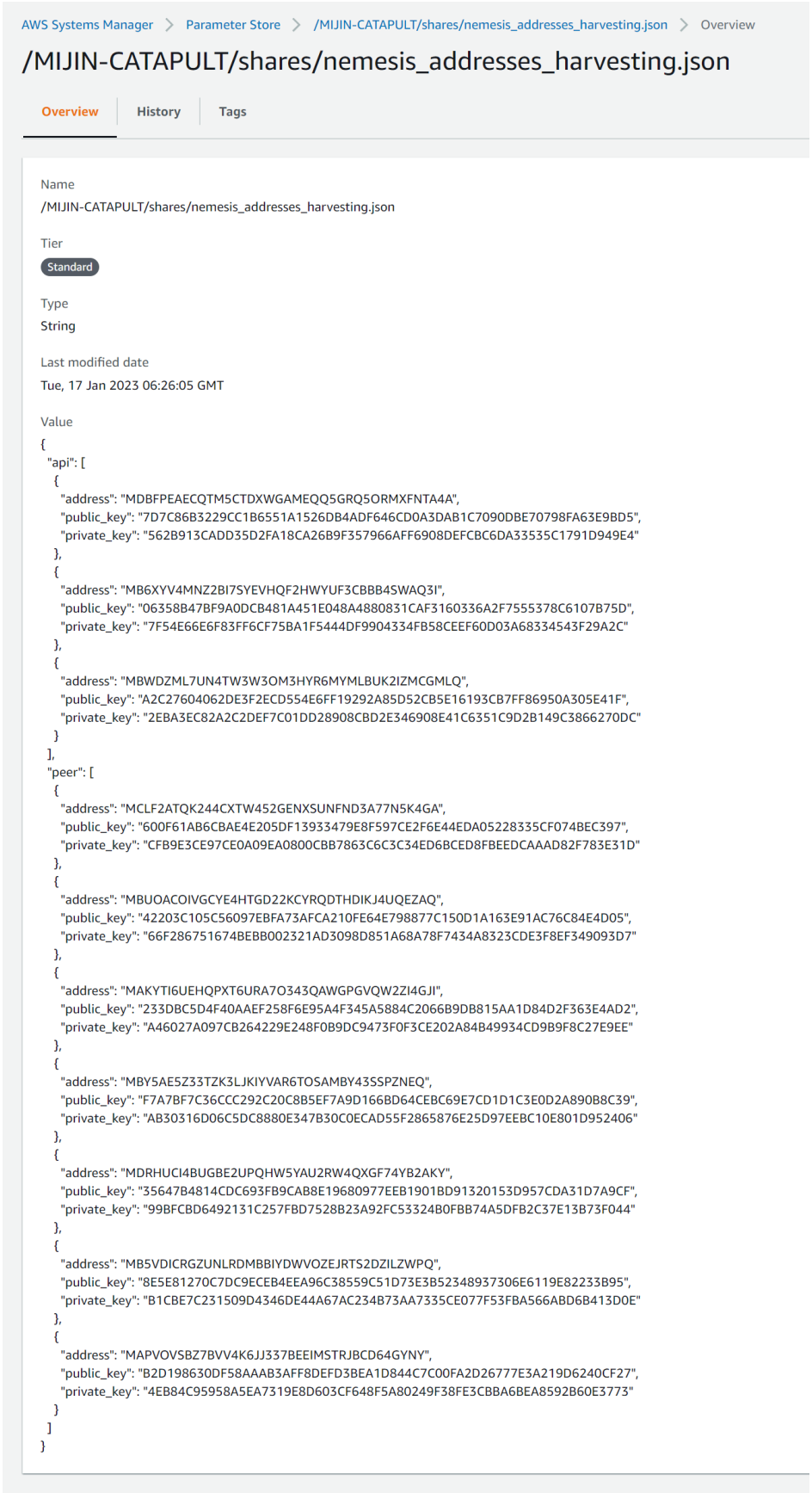
Symbol-cli has been archived, so symbol-cli may not be available.
From 1.0.3.4 and onwards, please refer to mijin Catapult(v.2) Balance transfer of balance account when commissioned mode is enabled.

Note:

In a blockchain, there is always a base currency and transaction fees must be paid in operating the blockchain.
mijin Catapult(v.2) But if you enable the with-fee mode, you will need to pay a fee to create a transaction, and you will need to operate with an account that has a base currency balance for all transaction fees, Mosaic rental fees, etc.
Conversely, the no-fee mode allows you to create transactions using accounts that do not carry a balance.

Verify accounts with balance

Outstanding accounts are stored in **nemesis_addresses_harvesting.json** in the AWS Systems Manager parameter store.
The list of accounts in this file is all tied to each node.



AWS Systems Manager > Parameter Store > /MIJIN-CATAPULT/shares/nemesis_addresses_harvesting.json > Overview

/MIJIN-CATAPULT/shares/nemesis_addresses_harvesting.json

Overview | History | Tags

Name
/MIJIN-CATAPULT/shares/nemesis_addresses_harvesting.json

Tier
Standard

Type
String

Last modified date
Tue, 17 Jan 2023 06:26:05 GMT

Value


```
{
 "api": [
  {
   "address": "MDBFPEAECQTM5CTDXWGAMEQQ5GRQ5ORMXFNTA4A",
   "public_key": "7D7C86B3229CC1B6551A1526DB4ADF646CD0A3DAB1C7090DBE70798FA63E9BD5",
   "private_key": "562B913CADD35D2FA18CA26B9F357966AFF6908DEFCBC6DA33535C1791D949E4"
  },
  {
   "address": "MB6XYV4MNZ2BI7SYEVHQF2HWYUF3CBBB4SWAQ3I",
   "public_key": "06358B47BF9A0DCB481A451E048A4880831CAF3160336A2F7555378C6107B75D",
   "private_key": "7F54E66E6F83FF6CF75BA1F5444DF9904334FB58CEEF60D03A68334543F29A2C"
  },
  {
   "address": "MBWDZML7UN4TW3W3OM3HYR6MYMLBUK2IZMCGMLQ",
   "public_key": "A2C27604062DE3F2ECD554E6FF19292A85D52CB5E16193CB7FF86950A305E41F",
   "private_key": "2EBA3EC82A2C2DEF7C01DD28908CBD2E346908E41C6351C9D2B149C3866270DC"
  }
 ],
 "peer": [
  {
   "address": "MCLF2ATQK244CXTW452GENXSUNFND3A77N5K4GA",
   "public_key": "600F61AB6CBAE4E205DF13933479E8F597CE2F6E44EDA05228335CF074BEC397",
   "private_key": "CFB9E3CE97CE0A09EA0800CBB7863C6C3C34ED6BCED8FBEEDCAAAD82F783E31D"
  },
  {
   "address": "MBUOACOIVGCYE4HTGD22KCYRQDTHDIKJ4UQEZAQ",
   "public_key": "42203C105C56097EBFA73AFCA210FE64E798877C150D1A163E91AC76C84E4D05",
   "private_key": "66F286751674BEBB002321AD3098D851A68A78F7434A8323CDE3F8EF349093D7"
  },
  {
   "address": "MAKYTI6UEHQPXT6URA7O343QAWGPGVQW2ZI4GJI",
   "public_key": "233DBC5D4F40AAEF258F6E95A4F345A5884C2066B9DB815AA1D84D2F363E4AD2",
   "private_key": "A46027A097CB264229E248F0B9DC9473F0F3CE202A84B49934CD9B9F8C27E9EE"
  },
  {
   "address": "MBY5AE5Z33TZK3LJKIYVAR6TOSAMBY43SSPZNEQ",
   "public_key": "F7A7BF7C36CCC292C20C8B5EF7A9D166BD64CEBC69E7CD1D1C3E0D2A890B8C39",
   "private_key": "AB30316D06C5DC8880E347B30C0ECAD55F2865876E25D97EEBC10E801D952406"
  },
  {
   "address": "MDRHUCI4BUGBE2UPQHW5YAU2RW4QXGF74YB2AKY",
   "public_key": "35647B4814CDC693FB9CAB8E19680977EEB1901BD91320153D957CDA31D7A9CF",
   "private_key": "99BFCBD6492131C257FBD7528B23A92FC53324B0FBB74A5DFB2C37E13B73F044"
  },
  {
   "address": "MB5VDICRGZUNLRDMBBIYDWVOZEJRTS2DZILZWPQ",
   "public_key": "8E5E81270C7DC9ECEB4EEA96C38559C51D73E3B52348937306E6119E82233B95",
   "private_key": "B1CBE7C231509D4346DE44A67AC234B73AA7335CE077F53FBA566ABD6B413D0E"
  },
  {
   "address": "MAPVOVSBZ7BVV4K6JJ337BEEIMSTRJBCD64GYNY",
   "public_key": "B2D198630DF58AAAB3AFF8DEFD3BEA1D844C7C00FA2D26777E3A219D6240CF27",
   "private_key": "4EB84C95958A5EA7319E8D603CF648F5A80249F38FE3CBBA6BEA8592B60E3773"
  }
 ]
}
```

All accounts have balances, but here we will check the following data at the top using symbol-cli.

```
{
  "address": "MAL4SPKWUI3WGSNOWSDA3KKIBJG7QHMCXD7GZVA",
  "public_key": "FDA90ACB0B4DA564FBA3D9D3A3E67A7146A77D2F5C246BC67AC044AAD578E161",
  "private_key": "F36139408F597D2F0DA0C5E3CB1162E3D80EFEF188E21089284F57723676CC5C"
},
```

Installation of symbol-cli

Install symbol-cli from npm.

```
$ sudo npm i -g symbol-cli@1.0.0
/usr/local/bin/symbol-cli -> /usr/local/lib/node_modules/symbol-cli/bin/symbol-cli
+ symbol-cli@1.0.0
updated 1 package in 8.724s
```

Import of balance accounts

Import balance accounts.

item	Description	value
Select the network type	Specify the network. CatapultNetwork value specified at build time	MIJIN/MIJIN_TEST
Enter the Symbol node URL.	Specify the mijinLBENdpoint or mijinEndpoint URL on the Outouts tab in the Cloudformation Stack.	<http://xxxxxxx:300>
Enter a profile name	Specify a profile name to invoke the account.	optional
Enter your wallet password	Specify the password for your account	optional
Do you want to set the account as the default profile?	Specify whether this account is to be used by Default.	optional
Select an import type	Specify the import method for saving again.	PrivateKey
Enter your account private key	private_key in **nemesis_addresses_harvesting.json**.	Optional **where the value starts with F3613**.

```
$ symbol-cli profile import
✓ Select the network type: › MIJIN
✓ Enter the Symbol node URL. (Example: http://localhost:3000): ⋯ http://
↪xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.amazonaws.com:3000
✓ Enter a profile name: ⋯ mijin-have-currency-account
✓ Enter your wallet password: ⋯ ********
✓ Do you want to set the account as the default profile? ⋯ no
✓ Select an import type: › PrivateKey
✓ Enter your account private key: ⋯
****************************************************************
```

(continues on next page)

(continued from previous page)

```
Account
┌──────────────┬──────────────────────────────────────────────────────────────────┐
│ Property     │ Value                                                            │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Address      │ MAL4SP-KWUI3W-GSNOWS-DA3KKI-BJG7QH-MCXD7G-ZVA                    │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key   │ FDA90ACB0B4DA564FBA3D9D3A3E67A7146A77D2F5C246BC67AC044AAD578E161  │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Private Key  │ F36139408F597D2F0DA0C5E3CB1162E3D80EFEF188E21089284F57723676CC5C  │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Password     │ Test1234                                                         │
└──────────────┴──────────────────────────────────────────────────────────────────┘

SUCCESS Stored mijin-have-currency-account profile
```

Verify account information.
The base currency is **1D8350FA8D4830FA** with a high value of Amount in the Mosaic Id in the Balance Information.

Warning: Note that the Mosaic Id for the base currency is created at the time of deployment and is not the same Id.

```
$ symbol-cli account info --profile mijin-have-currency-account
⁝ Processing
Account Information
┌──────────────────┬──────────────────────────────────────────────────────────────────┐
│ Property         │ Value                                                            │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address          │ MAL4SP-KWUI3W-GSNOWS-DA3KKI-BJG7QH-MCXD7G-ZVA                    │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address Height   │ 1                                                                │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key       │ FDA90ACB0B4DA564FBA3D9D3A3E67A7146A77D2F5C246BC67AC044AAD578E161 │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key Height │ 1                                                               │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance       │ 2850000                                                          │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance Height │ 2600                                                            │
└──────────────────┴──────────────────────────────────────────────────────────────────┘
Balance Information
┌──────────────────┬──────────────────┬──────────────────┬──────────────────┐
│ Mosaic Id        │ Relative Amount  │ Absolute Amount  │ Expiration Height │
├──────────────────┼──────────────────┼──────────────────┼──────────────────┤
│ 1D8350FA8D4830FA │ 1,799,799,999.6  │ 1799799999600000 │ Never            │
├──────────────────┼──────────────────┼──────────────────┼──────────────────┤
│ 01964E14621F06F6 │ 3,000            │ 3000000          │ Never            │
└──────────────────┴──────────────────┴──────────────────┴──────────────────┘
```

Create a new account

Create a new account with no balance.
Move the balance to address **MADIFG-N27CKA-6DY42J-UMEFJA-7OKXLO-NXLAEQ-XII** later.

item	Description	value
Select the network type	Specify the network. CatapultNetwork value specified at build time	MIJIN/MIJIN_TEST
Do you want to save the account?	Save this account.	yes
Select an import type	Specify the import method for saving again.	PrivateKey
Enter the Symbol node URL.	Specify the mijinLBENdpoint or mijinEndpoint URL on the Outouts tab in the Cloudformation Stack.	<http://xxxxxxx:300>
Enter a profile name	Specify a profile name to invoke the account.	optional
Enter your wallet password	Specify the password for your account	optional

```
$ symbol-cli account generate
✓ Select the network type: › MIJIN
✓ Do you want to save the account? ··· yes
✓ Select an import type: › PrivateKey
✓ Enter the Symbol node URL. (Example: http://localhost:3000): ··· http://
↪xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.amazonaws.com:3000
✓ Enter a profile name: ··· mijin-no-currency-account
✓ Enter your wallet password: ··· ********
✓ Do you want to set the account as the default profile? ··· no

Account
┌──────────────┬──────────────────────────────────────────────────────────────────┐
│ Property     │ Value                                                            │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Address      │ MADIFG-N27CKA-6DY42J-UMEFJA-7OKXLO-NXLAEQ-XII                    │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key   │ B86CDD63C3BA820C4659CF7FC3D53DA035CF8370AC3E0DBF025BEE691AED7DFA │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Private Key  │ E911E779671BD33B26A9D424DB331A36BDD497BA62D91B27ADAA4B1350A52D43 │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Password     │ Test1234                                                         │
└──────────────┴──────────────────────────────────────────────────────────────────┘

SUCCESS Stored mijin-no-currency-account profile
```

Query the node for account information.
Please confirm that this account is in error because the node does not have a record of the balance.

```
$ symbol-cli account info --profile mijin-no-currency-account
```

(continues on next page)

(continued from previous page)

```
ERR {"statusCode":404,"statusMessage":"Not Found","body":"{\"code\":\"ResourceNotFound\
↪",\"message\":\"no resource exists with id 'MADIFGN27CKA6DY42JUMEFJA7OKXLONXLAEQXII'\
↪"}"}

TIP The account has to receive at least one transaction to be recorded on the network
```

Balance transfers

Transfers balances from a balance account (mijin-have-currency-account) to a new account (mijin-no-currency-account).
Here we will transfer 100,000 cat.currency.

Announce a transfer transaction from your balance account (mijin-have-currency-account).

item	Description	value
Enter your wallet password	Please specify the password you have set	optional
Mosaics to transfer in the format (mosaicId(hex)|@aliasName)::absoluteAmount	The base currency (cat.currency) is sent 100,000. The base currency has a divisibility of 6, so it has a 6-digit decimal point, so add 6 zeros.	**3BF3AF8B22CB53D8::1**
Enter the recipient address or @alias	Specify the forwarding address for the new account	MADIFGN27CKA6DY42JUMEFJA7OKXLONXLAEQXII
Enter a message	A message can be added to the transfer transaction	optional
Enter the maximum fee (absolute amount)	Specifies the transaction fee. This changes with the commission mode. With commission 20000 or so (0.2cat.currency) Without fee 0 No fee 0	0
Select the transaction announce mode	Specifies how transactions are announced.	normal

```
$ symbol-cli transaction transfer --profile mijin-have-currency-account
✓ Enter your wallet password: ⋯ ********
✓ Mosaics to transfer in the format (mosaicId(hex)|@aliasName)::absoluteAmount, (Ex:
sending 1 symbol.xym, @symbol.xym::1000000). Add multiple mosaics separated by commas:
⋯ @cat.currency::100000000000
✓ Enter the recipient address or @alias: ⋯ MADIFGN27CKA6DY42JUMEFJA7OKXLONXLAEQXII
✓ Enter a message: ⋯ test
✓ Enter the maximum fee (absolute amount): ⋯ 200000
✓ Select the transaction announce mode: › normal
┌──────────────────────────────────────────────────────────────────────────────┐
│                                   TRANSFER                                   │
├──────────────┬───────────────────────────────────────────────────────────────┤
│ Max fee:     │ 200,000                                                       │
├──────────────┼───────────────────────────────────────────────────────────────┤
│ Network type: │ MIJIN                                                        │
├──────────────┼───────────────────────────────────────────────────────────────┤
```

(continues on next page)

(continued from previous page)

```
│ Deadline:     │ 2022-10-01 23:57:37.489                                         │
├───────────────┼─────────────────────────────────────────────────────────────────┤
│ Recipient:    │ MADIFG-N27CKA-6DY42J-UMEFJA-7OKXLO-NXLAEQ-XII                   │
├───────────────┼─────────────────────────────────────────────────────────────────┤
│ Message:      │ test                                                            │
├───────────────┼─────────────────────────────────────────────────────────────────┤
│ Mosaic (1/1): │ 100,000,000,000 cat.currency (85BBEA6CC462B244)                 │
├───────────────┴─────────────────────────────────────────────────────────────────┤
│                               Signature details                                 │
├───────────────┬─────────────────────────────────────────────────────────────────┤
│ Payload:      │ B5000000000000000496383B0C2AF6B3295D615336F48B2C299AAF38619399C40 │
│               │ 85AC7F6CF58092EAE743D5C754DC3C149E4E5EFA8E6038519F8BAAFDDF3B05BC │
│               │ 41B355638528AE03FDA90ACB0B4DA564FBA3D9D3A3E67A7146A77D2F5C246BC6 │
│               │ 7AC044AAD578E16100000000001605441400D030000000000517F484B18000000 │
│               │ 60068299BAF8940F0F1CD268C21520FB9575B9B758090BA10500010000000000 │
│               │ 44B262C46CEABB8500E87648170000000074657374                      │
├───────────────┼─────────────────────────────────────────────────────────────────┤
│ Hash:         │ 24FACA961CB1DF4D3F76DFFBE302D2CAA512F7D1BD424CC4E0D14ACFA7221FA4 │
├───────────────┼─────────────────────────────────────────────────────────────────┤
│ Signer:       │ FDA90ACB0B4DA564FBA3D9D3A3E67A7146A77D2F5C246BC67AC044AAD578E161 │
└───────────────┴─────────────────────────────────────────────────────────────────┘
✔ Do you want to announce this transaction? … yes

SUCCESS Transaction announced correctly

TIP To check if the network confirms or rejects the transaction, run the command
↪'symbol-cli transaction status'
```

Ensure that the new account has a balance.
The account information that was in error earlier is recognized by mijin Catapult(v.2) and you can confirm that you have a balance.

```
$ symbol-cli account info --profile mijin-no-currency-account
⠹ Processing
Account Information

┌───────────────────┬──────────────────────────────────────────────────────────────────┐
│ Property          │ Value                                                            │
├───────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address           │ MADIFG-N27CKA-6DY42J-UMEFJA-7OKXLO-NXLAEQ-XII                    │
├───────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address Height    │ 2706                                                             │
├───────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key        │ 0000000000000000000000000000000000000000000000000000000000000000 │
├───────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key Height │ 0                                                                │
├───────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance        │ 0                                                                │
├───────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance Height │ 0                                                                │
└───────────────────┴──────────────────────────────────────────────────────────────────┘
Balance Information

┌───────────────────┬─────────────────┬─────────────────┬───────────────────┐
│ Mosaic Id         │ Relative Amount │ Absolute Amount │ Expiration Height │
```

(continues on next page)

(continued from previous page)

```
┌──────────────────┬──────────────┬──────────────┬──────────────┐
│ 1D8350FA8D4830FA │ 100,000      │ 100000000000 │ Never        │
└──────────────────┴──────────────┴──────────────┴──────────────┘
```

2.2.7 AWS Troubleshooting

Summarize the troubleshooting of mijin Catapult(v.2) launched on AWS Marketplace.

2.2.7.1 Restore from a backed-up snapshot

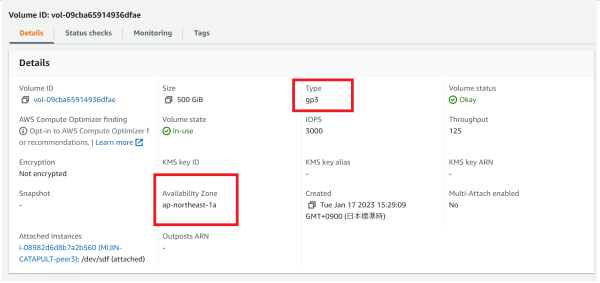
This chapter describes the procedure for restoring data on a mijin Catapult (v.2) node on AWS from a backup from a Snapshot obtained with AWS Backup.

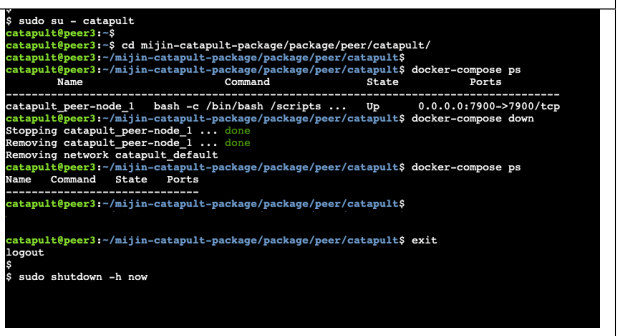
Note:

If you want to erase the blockchain data and restore it from another node instead of from Snaoshot, you can use mijin Catapult(v.2) node resynchronization.

Stop the node completely

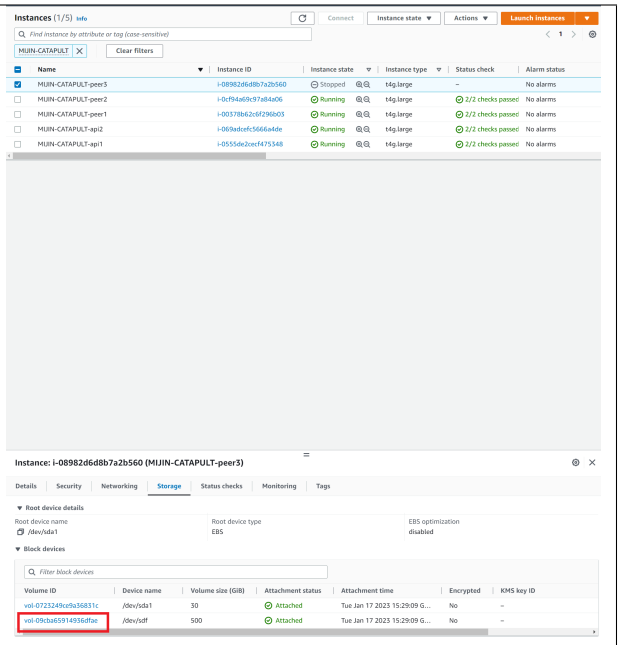
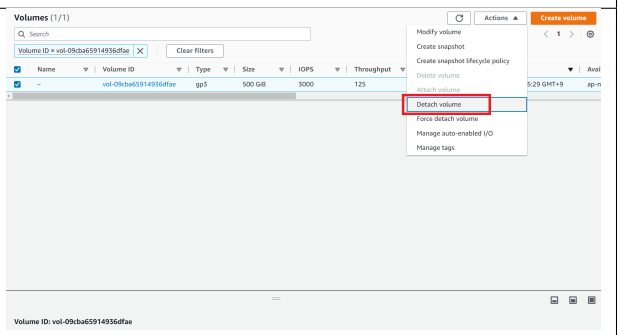
Blockchain data.
Here, we stop at PEER node 3 as an example.

Login to PEER node 3.	mijin Catapult(v.2) EC2 instance login how to and log in to the node.
Note the availability zone where the volume is located. Make sure it is GP3	

mijin Catapult(v.2) and stop the EC2 instance.
 	
`sudo su - catapult`
`cd mijin-catapult-package/package/peer/`
`↪catapult/`
`docker-compose down`
`docker-compose ps`
`exit`
`sudo shutdown -h now`
	

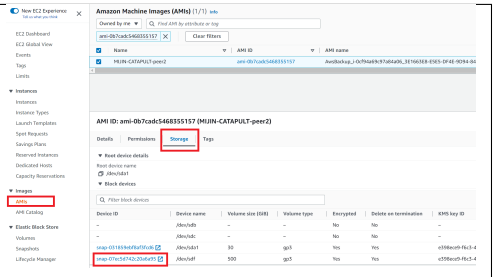
Detach volume of stopped PEER node

Remove the volume attached to the node.

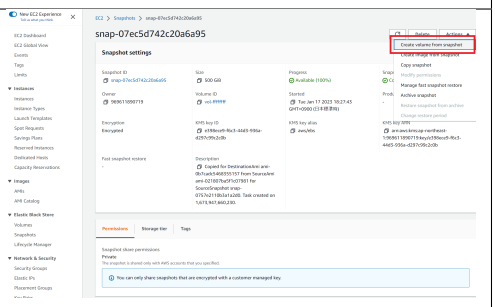
Select the volume from PEER node 3 again.	
From Volume, click on ‘Detach volume’and press OK on the screen that pops up.	

Create a volume from an AWS Backup Snapshot

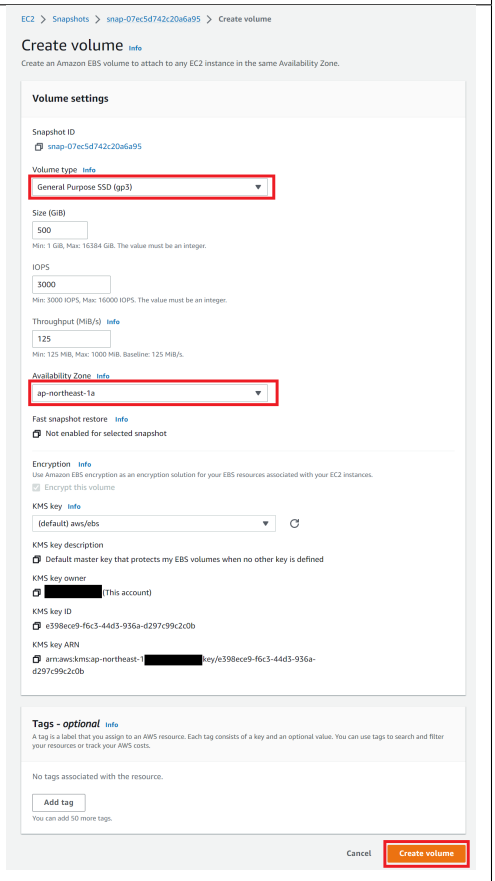
For AWS Backup, select a Snapshot of mijin data from AMI.
Skip this step if you want to select a manually acquired Snapshot.



Click on ‘Create volume from snapshot’.

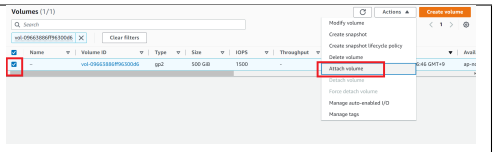
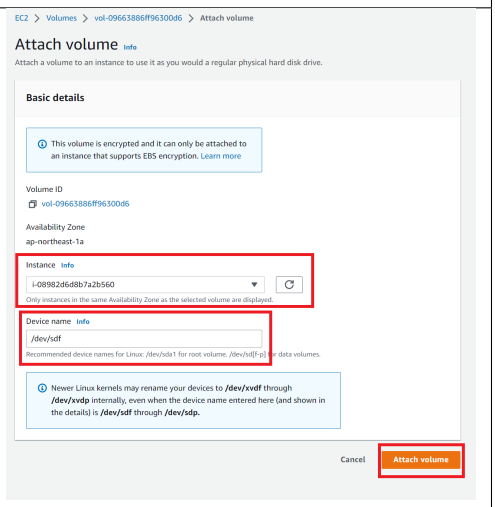
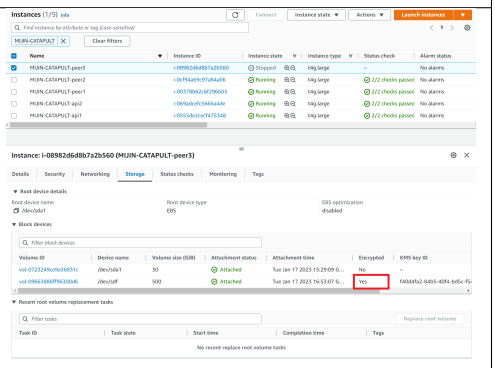


Select volume type gp3
Select availability zone Stop the node completely.



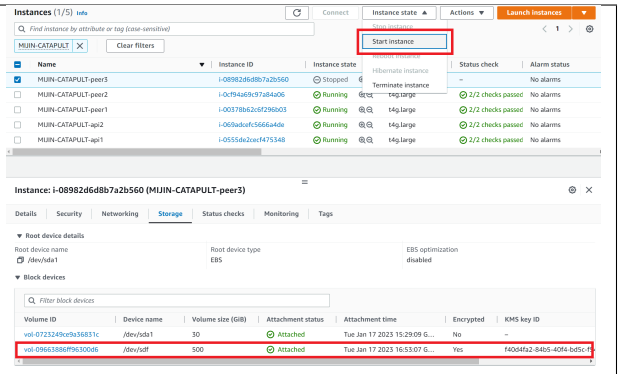
Attach the restored volume to the PEER node

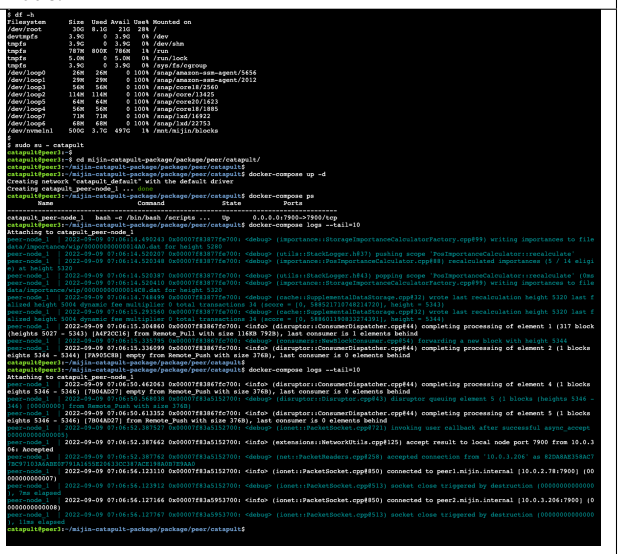
Attach the encrypted volume to the node

Select the encrypted volume and click ‘Attach volume’.	
Select instance PEER node 3 and specify the same path as the detached volume. Click on ‘Attach volume.’	
Verify that the restored volume is attached.	

Start up a stopped PEER node

Start up the stopped node and start up mijin Catapult(v.2) as well.

From the list of instances, check PEER node 3 and click 'Start Instance'.	
Login to PEER node 3.	mijin Catapult(v.2) EC2 instance login how to and log in to the node.

Make sure the disk is mounted, and start up mijin Catapult(v.2).
 	
`df -h`
`sudo su - catapult`
`cd mijin-catapult-package/package/peer/`
`↪catapult/`
`rm -rf /mnt/mijin/blocks/data/*.lock`
`docker-compose up -d`
`docker-compose ps`
	

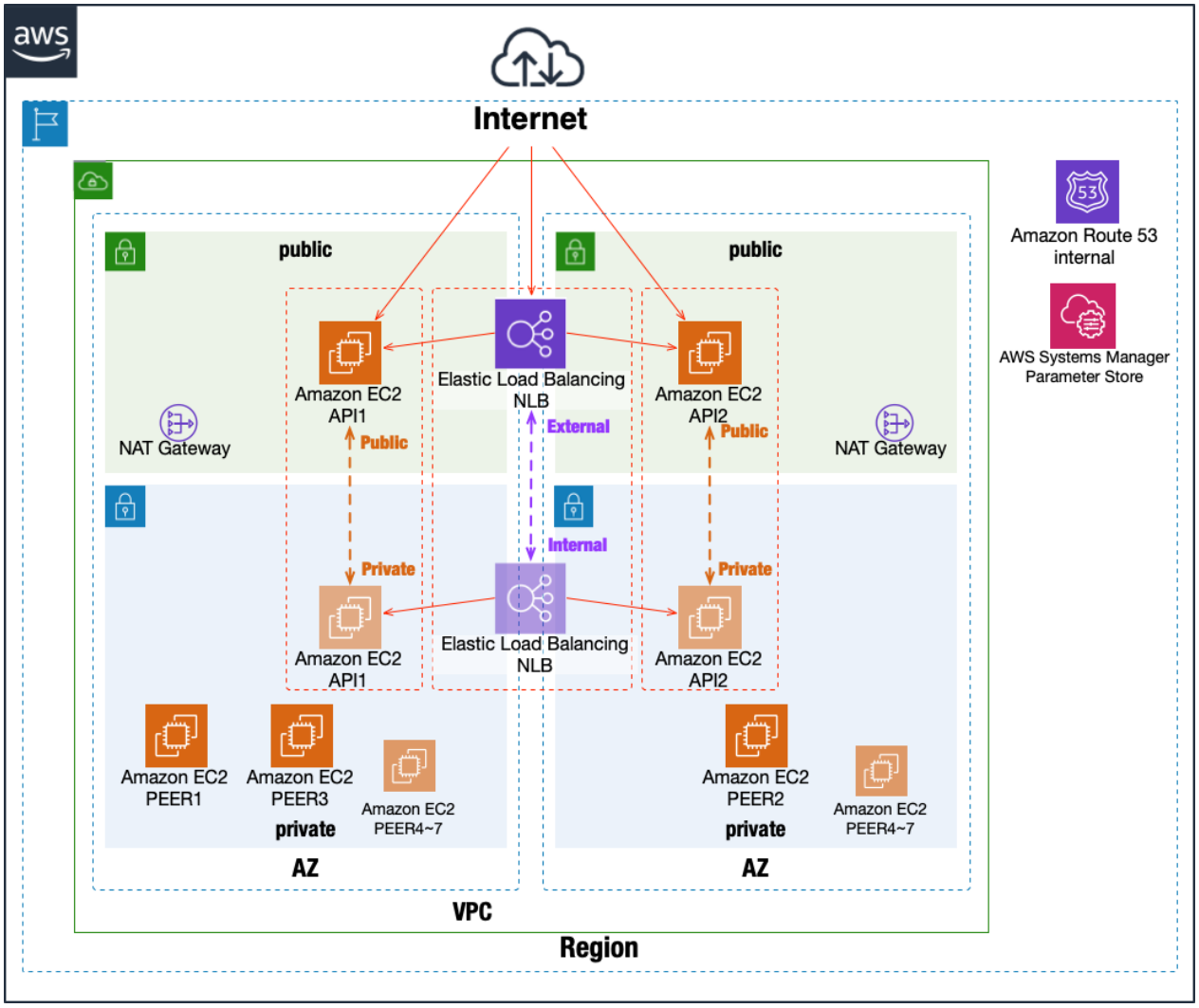
2.2.7.2 How to respond to availability zone (AZ) failures

This chapter explains how to respond to a failure in the availability zone (AZ) of a mijin Catapult(v.2) VPC on AWS.

mijin Catapult(v.2) Product Edition Configuration

mijin Catapult(v.2) In the production version, the VPC is located in one region and the nodes are distributed across two availability zones (AZs).
By enabling ELB installation in the deployment parameters, service can continue even if one of the AZs fails.
mijin Catapult(v.2) can update blockchain data with a single PEER node.
If you want to access it from a program, etc., a single API node is all you need to continue accessing it.

For example, even if the entire service surrounded by the left AZ stops, as shown in the figure below, mijin Catapult(v.2) will not stop.



How to respond after AZ restoration

After AZ restoration, the response is simple.
mijin Catapult(v.2) automatically connects to the node with the running blockchain data after recovery (after the EC2 instance is started) and starts synchronizing the data.
Therefore, no special work is required for restoration.
To check if a node is restored, you can check by command or browser as follows.

http://{mijin endpoint}:3000/node/peers

```
$ curl http://mijin エンドポイント:3000/node/peers
[
    {
        "version": 0,
        "publicKey": "9073B0A623934996A9AAAC85C6DEC8540AE17258D6997E42E00100CCFE6848EF",
        "networkGenerationHashSeed":
↪"B319300B02B12264B7DF867F0EFD583CC3C6E65ED2732E3FD77BBC1DE8E00E85",
        "roles": 70,
        "port": 7900,
        "networkIdentifier": 96,
        "host": "api1.mijin.internal",
        "friendlyName": "api1.mijin.internal"
    },
    {
        "version": 0,
        "publicKey": "82DA8AE358AC7DF7BC97103A6ABE0F791A1655E20633CC387ACE198A0B7E9AA0",
        "networkGenerationHashSeed":
↪"B319300B02B12264B7DF867F0EFD583CC3C6E65ED2732E3FD77BBC1DE8E00E85",
        "roles": 69,
        "port": 7900,
        "networkIdentifier": 96,
        "host": "peer2.mijin.internal",
        "friendlyName": "peer2.mijin.internal"
    },
    {
        "version": 0,
        "publicKey": "4EE257A9DD6D3F19331A467C6C76BA86B50B1297181E32C7A83C1184B666996C",
        "networkGenerationHashSeed":
↪"B319300B02B12264B7DF867F0EFD583CC3C6E65ED2732E3FD77BBC1DE8E00E85",
        "roles": 69,
        "port": 7900,
        "networkIdentifier": 96,
        "host": "peer1.mijin.internal",
        "friendlyName": "peer1.mijin.internal"
    },
    {
        "version": 0,
        "publicKey": "C158D513266B2C04216CDC03AD99036757A41AD2AFDF59D2A67F6D2D4F8CC84F",
        "networkGenerationHashSeed":
↪"B319300B02B12264B7DF867F0EFD583CC3C6E65ED2732E3FD77BBC1DE8E00E85",
        "roles": 69,
        "port": 7900,
        "networkIdentifier": 96,
        "host": "peer3.mijin.internal",
        "friendlyName": "peer3.mijin.internal"
```

(continues on next page)

(continued from previous page)

```
    }
]
```

If a node cannot be confirmed above, the blockchain data may be corrupted by a sudden node.
In that case, the mijin Catapult(v.2) node resynchronization or Restore from a backed-up snapshot to recover the node.

2.2.8 AWS MarketPlace mijin Catapult(v.2) FAQ List

This is a list of FAQs on mijin Catapult(v.2) deployed in AWS MarketPlace.

2.2.8.1 Product Version FAQ List

—

Q.Is it possible to carry over data from the Free Trial version?
A.Data cannot be transferred.

—

Q.Can I use it for commercial purposes?
A.Possible.

—

Q.Where is the mijin license fee included?
A.mijin license fees are charged by AWS.
As with the cost of EC2 instances, billing is added as an hourly usage fee.

—

Q.Is there an initial cost to start up?
A.There is no initial cost. However, there is a pay-as-you-go mijin license fee and AWS usage fee.

—

Q.Is there a free trial period for the product version?
A.There is no free trial period for the product version.
We have a separate free trial version available on the . /about_aws trial board.

—

Q.Who can I contact for support?
A.Please refer to the following for contact information, which varies depending on the case.

Technical inquiries about mijin or problems during construction (paid support)
For technical inquiries regarding mijin, please contact us from the following link, as it is necessary to purchase a paid support ticket.

https://mijin.io/en/aws_contact

■ Trouble on AWS.

https://aws.amazon.com/jp/premiumsupport/tech-support-guidelines/

—

Q.What is the version of mijin Catapult (v.2) product version?
A.

catapult-server: 1.0.3.8
catapult-rest: 2.5.0

(2025/6/10 時点)

—

Q.Is it possible to upgrade?
A.It is possible to upgrade.

However, if there is a large version difference, it may not be possible to upgrade.
By purchasing paid support, you will receive support and announcements for version upgrades.

https://mijin.io/en/aws_contact

—

Q.I tried to set up a second mijin Catapult(v.2) in the same region and it failed.
A.Deployment parameters when launching a second or later version of mijin Catapult(v.2)

The value of '**mijinStackAlreadyExist**'should be set to **YES**.

—

Q.I would like to build a disaster recovery environment.

A.The product version is deployed in a multi-AZ environment with nodes distributed as standard.
See AWS MarketPlace mijin Catapult(v.2) Recovery Strategy with Architectural Patterns for more information.

—

Q.I want to rotate the data in the AWS Systems Manager Parameter Store.
A.Data in the AWS Systems Manager Parameter Store is only used during initial deployment and is not Saved as **backup**.
The Parameter Store contains the certificate data used for encrypted communication of the node, so if you want to update it, use . /…/tech/troubles/replace_node_key.

—

Q.Does mijin work with arm(Graviton) instances?
A.AWS Marketplace provides both x86_64 and arm versions.
AWS MarketPlace Enterprise x86_64 Version
AWS MarketPlace Enterprise arm64 Version

Q.Is there any difference between the arm(Graviton) version and the x86_64 version?
A.As for mijin, it is optimized for each CPU architecture, but the speed is the same and the license fee is the same.
Starting up in arm will reduce the running cost of AWS usage fees.
For price differences, please refer to the simulated data in each instance of the minimum requirements.
(* Simulation data does not include mijin license fee)

x86_64 構成 https://calculator.aws/#/estimate?id=3df2b4611ffde3cc598ffc6fec9aff49b8a986b2
arm64 configuration
https://calculator.aws/#/estimate?id=c3bdc61df9a07f9760fdb790680cec8d3807b0dc

—

2.2.8.2 Free Trial Version FAQ List

—

Q.Are there any functional limitations?
A.There are no restrictions on mijin functions, but the following restrictions apply.
1.the base currency is limited to a small amount of '2,000 cat.currency'.
2.Mosaic, Namespace, and transaction issuance fees are required.

—

Q.Is there a limit to the length of use?
A.There is no set usage period in the system, but since this is a test license, it cannot be used for commercial purposes, support inquiries cannot be made, and so on.

—

Q.Is there a commercial version available?
A.The commercial version is available for commercial use.
For more information, check . /about_aws for more information.

—

Q.To what extent can you provide support for mijin's features?
A.The free trial version is a test license and does not provide support for mijin functions or development.
If you need support, you can do so by using the commercial version and signing a support contract.

—

Q.What is the version of mijin Catapult (v.2) free trial?
A.
catapult-server: 1.0.3.6
catapult-rest: 2.4.3

—

2.2.9 AWS MarketPlace mijin Catapult(v.2) 利用料金比較表

AWS MarketPlace で展開する mijin Catapult(v.2) では、AWS の EC2 インスタンスを自由に選択することができます。
Comparative table of server usage fees incurred outside of license fees.

Processor	EC2	EBS	VPC	Region	Instance Payment Methods	12 months($)	monthly amount($)	3 months approximate($)	Monthly Amount (yen)	Approximate 3 months (yen)	estimate
x86	t3.large * 5	GP2 30GB	New	us-east-1	Ondemand	4914.00	409.50	1227	¥57,330	¥171,780	見積
				us-east-1	Reserved 1 year	3399.84	283.32	849	¥39,665	¥118,860	見積

continues on next page

Table 12 – continued from previous page

Proces-sor	EC2	EBS	VPC	Region	Instance Payment Methods	12 months($)	monthly amount($)	3 months approxi-mate($)	Monthly Amount (yen)	Approx-imate 3 months (yen)	estimate
				ap-north-east1	Onde-mand	6385.32	532.11	1596	¥74,495	¥223,440	見積
				ap-north-east1	Re-served 1 year	4419.88	368.32	1104	¥51,565	¥154,560	見積
			既存	us-east-1	Onde-mand	4124.64	343.72	1029	¥48,121	¥144,060	見積
				us-east-1	Re-served 1 year	2610.48	217.54	651	¥30,456	¥91,140	見積
				ap-north-east1	Onde-mand	5297.64	441.47	1323	¥61,806	¥185,220	見積
				ap-north-east1	Re-served 1 year	3332.20	277.68	831	¥38,876	¥116,340	見積
	API c5n.2xlage * 2 PEER c5n.xlarge * 3	IO1 IOPS100 130GB * 2 IO1 IOPS100 80GB * 3	New	us-east-1	Onde-mand	15474.96	1289.58	3867	¥180,541	¥541,380	見積
				us-east-1	Re-served 1 year	10018.84	834.90	2502	¥116,886	¥350,280	見積
				ap-north-east1	Onde-mand	19378.92	1614.91	4842	¥226,087	¥677,880	見積
				ap-north-east1	Re-served 1 year	12506.88	1042.24	3126	¥145,914	¥437,640	見積
			既存	us-east-1	Onde-mand	14685.60	1223.80	3669	¥171,332	¥513,660	見積
				us-east-1	Re-served 1 year	9229.48	769.12	2307	¥107,677	¥322,980	見積
				ap-north-east1	Onde-mand	18291.24	1524.27	4572	¥213,398	¥640,080	見積
				ap-north-east1	Re-served 1 year	11419.20	951.60	2853	¥133,224	¥399,420	見積
Arm	t4g.large	GP2 30GB	New	us-east-1	Onde-mand	4213.20	351.10	1053	¥49,154	¥147,420	見積
				us-east-1	Re-served 1 year	2994.84	249.57	747	¥34,940	¥104,580	見積
				ap-north-east1	Onde-mand	5404.20	450.35	1350	¥63,049	¥189,000	見積
				ap-north-east1	Re-served 1 year	3844.88	320.41	960	¥44,857	¥134,400	見積
			既存	us-east-1	Onde-mand	3423.84	285.32	855	¥39,945	¥119,700	見積
				us-east-1	Re-served 1 year	2205.48	183.79	549	¥25,731	¥76,860	見積
				ap-north-east1	Onde-mand	4316.52	359.71	1077	¥50,359	¥150,780	見積

continues on next page

Table 12 – continued from previous page

Proces-sor	EC2	EBS	VPC	Region	Instance Payment Methods	12 months($)	monthly amount($)	3 months approxi-mate($)	Monthly Amount (yen)	Approx-imate 3 months (yen)	estimate
				ap-north-east1	Re-served 1 year	2987.20	248.93	744	¥34,851	¥104,160	見積
	API c6g.2xlage * 2 PEER c6g.xlarge * 3	IO1 IOPS100 130GB * 2 IO1 IOPS100 80GB * 3	New	us-east-1	Onde-mand	10569.36	880.78	2640	¥123,309	¥369,600	見積
				us-east-1	Re-served 1 year	7134.84	594.57	1782	¥83,240	¥249,480	見積
				ap-north-east1	Onde-mand	13197.86	1099.82	3297	¥153,975	¥461,580	見積
				ap-north-east1	Re-served 1 year	8873.88	739.49	2217	¥103,529	¥310,380	見積
			既存	us-east-1	Onde-mand	9780.00	815.00	2445	¥114,100	¥342,300	見積
				us-east-1	Re-served 1 year	6345.48	528.79	1584	¥74,031	¥221,760	見積
				ap-north-east1	Onde-mand	12110.18	1009.18	3027	¥141,285	¥423,780	見積
				ap-north-east1	Re-served 1 year	7786.20	648.85	1944	¥90,839	¥272,160	見積

Technical

3.1 mijin Catapult(v.2) Basics

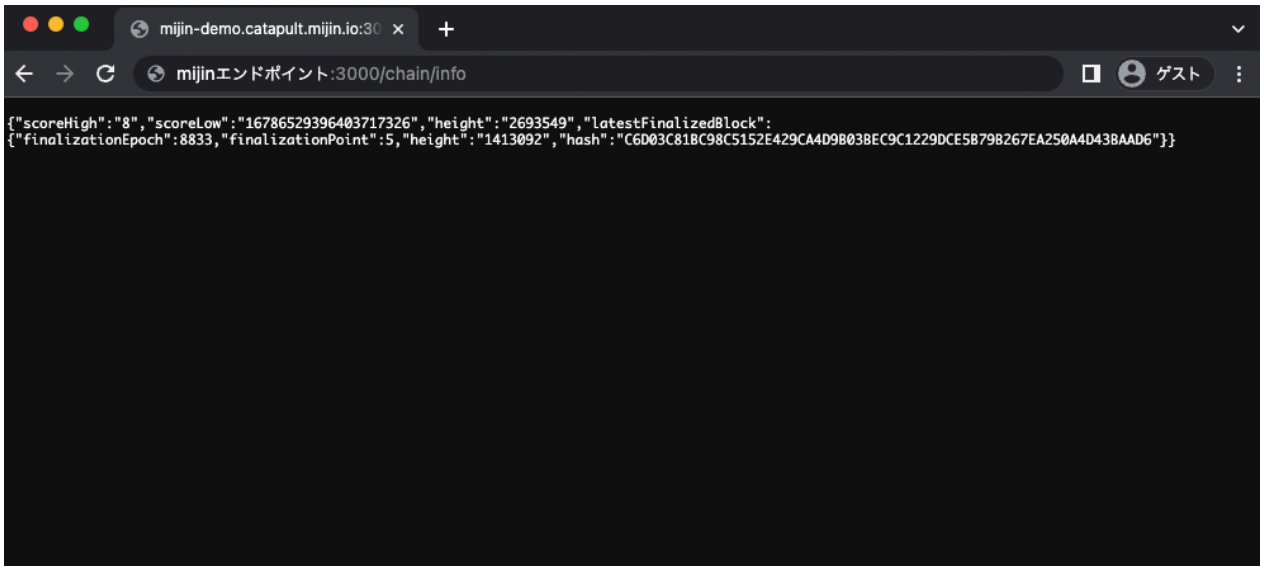
3.1.1 mijin Catapult(v.2) How to access

mijin Catapult(v.2) access to the API Rest endpoint of the API node allows blockchain operations.
Below you can check the current number of blocks in the blockchain with the command

```
$ curl -Ss http://mijin エンドポイント:3000/chain/info | jq -r
{
  "scoreHigh": "8",
  "scoreLow": "16778237146341708801",
  "height": "2693365",
  "latestFinalizedBlock": {
    "finalizationEpoch": 8833,
    "finalizationPoint": 5,
    "height": "1413092",
    "hash": "C6D03C81BC98C5152E429CA4D9B03BEC9C1229DCE5B79B267EA250A4D43BAAD6"
  }
}
```

item	Description
height	Current block height of blockchain
latestFinalizedBlock.height	Block height of the determined blockchain

You can also check it with Chrome or other browsers.



Others can also be accessed via sdk or cli.

For access to CLI, please refer to mijin Catapult(v.2) manipulation.

3.1.2 mijin Catapult(v.2) How to check status

The status of each mijin node can be obtained from REST.
You can check it directly in your browser or curl.
You can also check the same in sdk.

As for the REST response, it will be the same as Symbol from Nem.

https://symbol.github.io/symbol-openapi/v1.0.0/

3.1.2.1 Check block height

/chain/info

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/chain/info | jq -r
{
  "scoreHigh": "0",
  "scoreLow": "1400415221207545868",
  "height": "12318",
  "latestFinalizedBlock": {
    "finalizationEpoch": 78,
    "finalizationPoint": 8,
    "height": "12300",
    "hash": "7DE8B8052D35E29D0020EE7DB65BE075F0B6CEC69F17018447470E205B68175D"
  }
}
```

item	Description
height	Current block height
latestFinalizedBlock	finalize block
finalizationEpoch	•
finalizationPoint	•
height	•

3.1.2.2 Check the REST version

/node/server

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/node/server | jq -r
{
  "serverInfo": {
    "restVersion": "2.3.5",
    "sdkVersion": "2.3.5"
  }
}
```

item	Description
restVersion	version of rest
sdkVersion	version of catapult-sdk used by rest

3.1.2.3 Check node information

/node/info

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/node/info | jq -r
{
  "version": 16777216,
  "publicKey": "03ECD9C1929E26ED53BEBCCCF17E6F32F37ED9C6474397F592C883F771AB6A05",
  "networkGenerationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
  "roles": 70,
  "port": 7900,
  "networkIdentifier": 96,
  "host": "api2.mijin.internal",
  "friendlyName": "api2.mijin.internal",
  "nodePublicKey": "5958AE940208CF8FD0D7FF2A584F8B234A3814AFC4D93F304A5CEA926EF6A747"
}
```

item	Description
publicKey	Public key used in Harvest
networkGenerationHashSeed	Blockchain-specific GenerationHash settings created
roles	Node Roles (api/peer/dual/voting)
port	Communication port between nodes
networkIdentifier	Network Type
host	Host name of the node on which it is running
friendlyName	Friendly name of node
nodePublicKey	Public key for node

3.1.2.4 Check the connected node

/node/peers

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/node/peers | jq -r
[
  {
    "version": 0,
    "publicKey": "DB8D9DD59D78AE62E157824305DE31B9D415AA217EFE1DF14A7361E9D20E7456",
    "networkGenerationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
    "roles": 69,
    "port": 7900,
    "networkIdentifier": 96,
    "host": "peer1.mijin.internal",
    "friendlyName": "peer1.mijin.internal"
  },
  {
    "version": 0,
    "publicKey": "22722F1534AE77DA44A065C0E2ACB125CB66FB45E80403A183EFEBE222BF3D90",
    "networkGenerationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
    "roles": 69,
    "port": 7900,
    "networkIdentifier": 96,
    "host": "peer2.mijin.internal",
    "friendlyName": "peer2.mijin.internal"
  },
  {
    "version": 0,
    "publicKey": "239CC13A2B3D112C4146415EE532146D5338614BBBAD1A1E2E8E4690638F07D9",
    "networkGenerationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
    "roles": 69,
    "port": 7900,
    "networkIdentifier": 96,
    "host": "peer3.mijin.internal",
    "friendlyName": "peer3.mijin.internal"
  },
  {
    "version": 0,
    "publicKey": "E4BF3706483B4D42243F3DCB2625021C3E3AE7C253CC466154EEDF9775012C20",
    "networkGenerationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
```

(continues on next page)

(continued from previous page)

```
    "roles": 70,
    "port": 7900,
    "networkIdentifier": 96,
    "host": "api1.mijin.internal",
    "friendlyName": "api1.mijin.internal"
  }
]
```

item	Description
publicKey	Public key used in Harvest
networkGenerationHashSeed	Blockchain-specific GenerationHash settings created
roles	Node Roles (api/peer/dual/voting)
port	Communication port between nodes
networkIdentifier	Network Type
host	Host name of the node on which it is running
friendlyName	Friendly name of node
nodePublicKey	Public key for node

3.1.2.5 Check the total number of transactions and total number of accounts

/node/storage

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/node/storage | jq -r
{
  "numBlocks": 12322,
  "numTransactions": 34,
  "numAccounts": 14
}
```

item	Description
numBlocks	Current block height
numTransactions	Total number of transactions issued in the past
numAccounts	Total number of accounts used in the past

3.1.2.6 Check network type

` /network `

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/network | jq -r
{
  "name": "mijin",
  "description": "mijin network"
}
```

item	Description
name	The network name used mijin or mijin-test.
description	Network Description

3.1.2.7 Check the status of a node's container

/node/health

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/node/health | jq -r
{
  "status": {
    "apiNode": "up",
    "db": "up"
  }
}
```

item	Description
apiNode	api-node container status up or down.
db	db container status up or down.

3.1.2.8 Check the settings for the entire blockchain

/network/properties

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/network/properties | jq -r
{
  "network": {
    "identifier": "mijin",
    "nemesisSignerPublicKey":
↪"12086D4CB80CB6461887427BD49ED22D3914117526F573CC6F9937FC19DB2F73",
    "nodeEqualityStrategy": "host",
    "generationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
    "epochAdjustment": "1560294000s"
  },
  "chain": {
    "enableVerifiableState": true,
    "enableVerifiableReceipts": true,
    "currencyMosaicId": "0x61D0'A72B'3C62'5448",
    "harvestingMosaicId": "0x1248'680A'CB99'E205",
    "blockGenerationTargetTime": "15s",
    "blockTimeSmoothingFactor": "3000",
    "importanceGrouping": "40",
    "importanceActivityPercentage": "5",
    "maxRollbackBlocks": "0",
    "maxDifficultyBlocks": "60",
    "defaultDynamicFeeMultiplier": "0",
    "maxTransactionLifetime": "24h",
    "maxBlockFutureTime": "500ms",
    "initialCurrencyAtomicUnits": "8'998'999'998'000'000",
    "maxMosaicAtomicUnits": "9'000'000'000'000'000",
    "totalChainImportance": "15'000'000",
    "minHarvesterBalance": "1'000'000",
    "maxHarvesterBalance": "15'000'000",
    "minVoterBalance": "1'000'000",
    "votingSetGrouping": "160",
    "maxVotingKeysPerAccount": "3",
```

(continues on next page)

(continued from previous page)

```
    "minVotingKeyLifetime": "72",
    "maxVotingKeyLifetime": "26280",
    "harvestBeneficiaryPercentage": "10",
    "harvestNetworkPercentage": "5",
    "harvestNetworkFeeSinkAddress": "MBVF6QLFNKAXDBZLJYBPBT2YYKMJW7UE7GH7RTY",
    "maxTransactionsPerBlock": "6'000"
  },
  "plugins": {
    "accountlink": {
      "dummy": "to trigger plugin load"
    },
    "aggregate": {
      "maxTransactionsPerAggregate": "1'000",
      "maxCosignaturesPerAggregate": "25",
      "enableStrictCosignatureCheck": false,
      "enableBondedAggregateSupport": true,
      "maxBondedTransactionLifetime": "48h"
    },
    "lockhash": {
      "lockedFundsPerAggregate": "0",
      "maxHashLockDuration": "2d"
    },
    "locksecret": {
      "maxSecretLockDuration": "30d",
      "minProofSize": "1",
      "maxProofSize": "1000"
    },
    "metadata": {
      "maxValueSize": "1024"
    },
    "mosaic": {
      "maxMosaicsPerAccount": "1'000",
      "maxMosaicDuration": "3650d",
      "maxMosaicDivisibility": "6",
      "mosaicRentalFeeSinkAddress": "MBKRTIOKHE34GF7J5WZDW6VLXEDYFRFFURN2EZA",
      "mosaicRentalFee": "0"
    },
    "multisig": {
      "maxMultisigDepth": "3",
      "maxCosignatoriesPerAccount": "25",
      "maxCosignedAccountsPerAccount": "25"
    },
    "namespace": {
      "maxNameSize": "64",
      "maxChildNamespaces": "256",
      "maxNamespaceDepth": "3",
      "minNamespaceDuration": "1m",
      "maxNamespaceDuration": "3650d",
      "namespaceGracePeriodDuration": "30d",
      "reservedRootNamespaceNames": "xem, nem, user, account, org, com, biz, net, edu,
mil, gov, info",
      "namespaceRentalFeeSinkAddress": "MBWRFMKEJRDUZC5WEW2PFYG374AI444HL2WQX6A",
      "rootNamespaceRentalFeePerBlock": "1",
      "childNamespaceRentalFee": "0"
    },
    "restrictionaccount": {
      "maxAccountRestrictionValues": "512"
```

(continues on next page)

(continued from previous page)

```
    },
    "restrictionmosaic": {
      "maxMosaicRestrictionValues": "20"
    },
    "transfer": {
      "maxMessageSize": "1024"
    }
  }
}
```

For this set value, see the

3.1.2.9 Check transaction fees

/network/fees/transaction

```
$ curl -Ss http://mijin-catapult-1-nlb-rest-eef0ebffe49c4aa3.elb.ap-northeast-1.
↪amazonaws.com:3000/network/fees/transaction | jq -r
{
  "averageFeeMultiplier": 0,
  "medianFeeMultiplier": 0,
  "highestFeeMultiplier": 0,
  "lowestFeeMultiplier": 0,
  "minFeeMultiplier": 0
}
```

item	Description
averageFeeMultiplier	Multiplier value of average (automatic)
medianFeeMultiplier	Median multiplier value (automatic)
highestFeeMultiplier	Maximum multiplier value used
lowestFeeMultiplier	Minimum multiplier value used
minFeeMultiplier	Minimum required multiplier value set for the node; if 0, no fee mode

3.2 mijin Catapult(v.2) manipulation

Symbol, a public blockchain and mijin, are originally the same and are OSS under a dual license. Therefore, Symbol's tools can also be used with mijin. In this example, we will operate a test for transferring money with mijin Catapult(v.2).

Warning:

Symbol-cli is a Public Archive and may not work properly. Symbol-sdk must be a forked version from the original symbol.
symbol-sdk is a custom sdk forked for mijin.

3.2.1 mijin account creation

This chapter describes the account operations that must be performed in order to operate mijin. Operation will be performed on Linux (Ubuntu 20.04), so a minimum understanding of Linux operation is assumed.

Note: mijin Catapult(v.2) to operate, you create an account and use that account to submit transactions.Also note that you must have a base currency (cat.currency) balance in your account if you are in commissionable mode as specified during deployment.

3.2.1.1 Install nodejs and yarn

Install nodejs to use mijin-catapult-tools. Install nodejs using NodeSource

```
$ curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash - && sudo apt-get
install -y nodejs
$ node --version
$ sudo npm install -g yarn
```

3.2.1.2 Installing mijin-catapult-tools

Use yarn to install mijin-catapult-tools.

```
$ yarn global add  @tech-bureau/mijin-catapult-tools
$ echo 'export PATH="$HOME/.yarn/bin:$PATH"' >> ~/.bashrc && source ~/.bashrc
```

Verify that mijin-catapult-tools is available.

```
$ mijin-catapult-tools
```

3.2.1.3 Create an account

First, let's create an account to be used for testing. Here we will use **test1Account test2Account** and store them in mijin.json.

Create Account

```
$ mijin-catapult-tools account generate -u http://localhost:3000 -w mijin.json -s
2023-01-17T06:34:59.412Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:34:59.413Z [info] : Network: 96
2023-01-17T06:34:59.413Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:34:59.413Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:34:59.413Z [info] : Start Account Generate...
2023-01-17T06:34:59.475Z [info] : Write Config File: mijin.json
2023-01-17T06:34:59.476Z [info] : New Account: {
  "url": "http://localhost:3000",
  "workAccount": {
```

(continues on next page)

(continued from previous page)


```
    "publicKey": "425F06A8870381A00BD83E2D1083BB690F9FCB815F0919DFFA1A53A68E144D06",
    "privateKey": "9E9C660164AB344030DF5E77539952D9F5B380311C978369BA08923F577D8DAB",
    "address": "MC5VTRZO7IGJUG2VPQXBD4GDO4A74YLFUCVZZGI"
  },
  "balanceAccount": {
    "publicKey": "",
    "privateKey": "",
    "address": ""
  },
  "mainAccount": {
    "publicKey": "F2985009341A526B17E954EB6EA3EC77E1A0B33AE31EB29F86A69D9BB283AF67",
    "privateKey": "FF9A552ED35D930378F1E6B349A677771F19A06BCA0C4D5DCA621F354F5C6956",
    "address": "MCV66SABR6MBWT2I56YI3ULCPYWREJFX5DHHGDQ"
  },
  "keylink": {
    "vrf": {
      "publicKey": "C074A57A0EDF633414DDD764C6771926E91596481E0C2CAD4D9C3EFAAA432BCE",
      "privateKey": "C07D9A63B8DD8B0E639AAAE6059D7FA6C554082677926B7CACF0514E3940692C",
      "address": "MBLYBGODVGLT3HED6EXAQAXJE4CHHVRWGIAS45Y"
    },
    "voting": {
      "publicKey": "F6571CBC420A4EBA09E027AA53E5DD9486642911CDC11E0D3A6D1B2E4BC228D1",
      "privateKey": "FD5744779348F77F6EA6288232D0C8944E2F3C9E24D2E0FB1E93A8C4F9DD9BA5",
      "address": "MD2M6SDIE64O6ZWF2IJ7R4R5RDKOT2FNJDUIYOQ"
    }
  },
  "test1Account": {
    "publicKey": "CB51613497A40D9A256B17932579BC64D5037A04B29737B944965C1ADADD6E04",
    "privateKey": "8D5969EF1796F5F90256C92B5017396E40786ED87995169D4E26C9E5E01D7F8C",
    "address": "MAS36UGDCOGG6GYCBDPX3ROISABSNPZ6JQXMJSA"
  },
  "test2Account": {
    "publicKey": "5CC14799D5B2643914C8E574C8D073A7EE9AE2A405F1339A53612B566498AB1D",
    "privateKey": "FDE625C1D53AF04533FEB06A3556679FC02C4C6246952D3A534EB3E19CF83C56",
    "address": "MC5AH4UGBPPHNCMPTVNSY6LRDPIMEDODS2O373A"
  }
}
```


At this point, test1Account and test2Account are checked and do not exist.
This is because there is no exchange of this address on the blockchain, so its existence cannot be confirmed.

```
$ catapult@catapult:~$ mijin-catapult-tools account info -r mijin.json -t test1
2023-01-17T06:36:26.600Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:36:26.601Z [info] : Network: 96
2023-01-17T06:36:26.601Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:36:26.601Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:36:26.601Z [info] : Start Account Info
2023-01-17T06:36:26.620Z [error] : Address Not Found
```

```
$ mijin-catapult-tools account info -r mijin.json -t test2
2023-01-17T06:36:43.387Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:36:43.387Z [info] : Network: 96
```

(continues on next page)

(continued from previous page)

```
2023-01-17T06:36:43.387Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:36:43.387Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:36:43.387Z [info] : Start Account Info
2023-01-17T06:36:43.405Z [error] : Address Not Found
```

About Fees and Commissions

Fees are charged by transactions issued in the following items and must be paid in the base currency (cat.currency).

- transaction fee
- Mosaic Rental Fee
- Namespace Rental Fees
- Locked guarantee deposit

For mijin offered in the marketplace, the commission mode is set to none as standard.
With no fees, there is no need to have a base currency balance in the account, so there is nothing to be aware of, but you will need to set the fee setting to **0** each time in the sdk, for example.
If you created a mijin with fees, refer to the (Extra)How to move the base currency when in commissionable mode field and **test1Account** to send the base currency to the account in **test1Account** ".

3.2.1.4 Create and transfer Mosaic

First, let's create and transfer the basic Mosaic.

Create Mosaic

Issue one issue of Mosaic(Token) with the test1 account as the owner.

```
$ mijin-catapult-tools transaction mosaic create -r mijin.json -o test1 -s 1 -d 0
2023-01-17T06:42:51.942Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:42:51.942Z [info] : Network: 96
2023-01-17T06:42:51.942Z [info] : Create Mosaic...
2023-01-17T06:42:51.955Z [info] : Mosaic Owener Account:
MAS36UGDCOGG6GYCBDPX3ROISABSNPZ6JQXMJSA
2023-01-17T06:42:51.955Z [info] : MosaicId: 3D86C9FE5D52DE6F
2023-01-17T06:42:51.955Z [info] : Mosaic Flags: supplymutable:true, transferable:true,
restrictable:true, revokable:false
2023-01-17T06:42:51.955Z [info] : Mosaic Supply: divisibility:0, supply:1
2023-01-17T06:42:51.964Z [info] : Transaction Fee: 0
2023-01-17T06:42:51.964Z [info] : Mosaic Rental Fee: 0
2023-01-17T06:42:51.964Z [info] : Start Aggregate Transaction...
2023-01-17T06:43:00.967Z [info] : End Aggregate Transaction
2023-01-17T06:43:00.967Z [info] : http://localhost:3000/transactionStatus/
↪39BBE7E083460C3B85EB7D5FA2FB486F9786CA3FF1E0F976214F5753E527383A
2023-01-17T06:43:00.967Z [info] : http://localhost:3000/transactions/confirmed/
↪39BBE7E083460C3B85EB7D5FA2FB486F9786CA3FF1E0F976214F5753E527383A
```

Checking the information on account A, we can see that it has only 1 MosaicId 3D86C9FE5D52DE6F.

```
$ mijin-catapult-tools account info -r mijin.json -t test1
2023-01-17T06:43:43.349Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:43:43.350Z [info] : Network: 96
2023-01-17T06:43:43.350Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:43:43.350Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:43:43.350Z [info] : Start Account Info
2023-01-17T06:43:43.365Z [info] : test1 Account: {
  "publicKey": "CB51613497A40D9A256B17932579BC64D5037A04B29737B944965C1ADADD6E04",
  "address": "MAS36UGDCOGG6GYCBDPX3ROISABSNPZ6JQXMJSA",
  "mosaics": [
    {
      "id": "3D86C9FE5D52DE6F",
      "amount": "1",
      "currency": false,
      "harvest": false
    }
  ],
  "keylink": {
    "vrf": {
      "publicKey": ""
    },
    "voting": {
      "publicKey": "",
      "startEpoch": "",
      "endEpoch": ""
    }
  }
}
```

Transfer Mosaic

Transfer the Mosaic **3D86C9FE5D52DE6F** you just created from **test1** to **test2**.

```
$ mijin-catapult-tools transaction transfer -r mijin.json -f test1 -d test2 -m
3D86C9FE5D52DE6F -a 1
2023-01-17T06:44:46.983Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:44:46.984Z [info] : Network: 96
2023-01-17T06:44:46.984Z [info] : Start Transfer Account...
2023-01-17T06:44:46.996Z [info] : From Account Address:
MAS36UGDCOGG6GYCBDPX3ROISABSNPZ6JQXMJSA
2023-01-17T06:44:46.996Z [info] : Dest Account Address:
MC5AH4UGBPPHNCMPTVNSY6LRDPIMEDODS2O373A
2023-01-17T06:44:47.005Z [info] : Start Transfer Transaction...
2023-01-17T06:44:52.197Z [info] : End Transfer Transaction
2023-01-17T06:44:52.197Z [info] : http://localhost:3000/transactionStatus/
↪454DFDC48F00852A3DE355D7D2AD4B581D718D999BA177E1BB392AECD1656C6A
2023-01-17T06:44:52.197Z [info] : http://localhost:3000/transactions/confirmed/
↪454DFDC48F00852A3DE355D7D2AD4B581D718D999BA177E1BB392AECD1656C6A
```

Check the status of the test1 account.
The test2 account does not have Mosaic **3D86C9FE5D52DE6F**.

```
$ mijin-catapult-tools account info -r mijin.json -t test1
2023-01-17T06:45:51.931Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:45:51.931Z [info] : Network: 96
2023-01-17T06:45:51.931Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:45:51.931Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:45:51.931Z [info] : Start Account Info
2023-01-17T06:45:51.946Z [info] : test1 Account: {
  "publicKey": "CB51613497A40D9A256B17932579BC64D5037A04B29737B944965C1ADADD6E04",
  "address": "MAS36UGDCOGG6GYCBDPX3ROISABSNPZ6JQXMJSA",
  "mosaics": [],
  "keylink": {
    "vrf": {
      "publicKey": ""
    },
    "voting": {
      "publicKey": "",
      "startEpoch": "",
      "endEpoch": ""
    }
  }
}
```

Then check the status of the **test2 account**.
Confirm that the test2 account owns Mosaic **3D86C9FE5D52DE6F**.

```
$ mijin-catapult-tools account info -r mijin.json -t test2
2023-01-17T06:46:41.737Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:46:41.738Z [info] : Network: 96
2023-01-17T06:46:41.738Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:46:41.738Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:46:41.738Z [info] : Start Account Info
2023-01-17T06:46:41.754Z [info] : test1 Account: {
  "publicKey": "0000000000000000000000000000000000000000000000000000000000000000",
  "address": "MC5AH4UGBPPHNCMPTVNSY6LRDPIMEDODS2O373A",
  "mosaics": [
    {
      "id": "3D86C9FE5D52DE6F",
      "amount": "1",
      "currency": false,
      "harvest": false
    }
  ],
  "keylink": {
    "vrf": {
      "publicKey": ""
    },
    "voting": {
      "publicKey": "",
      "startEpoch": "",
      "endEpoch": ""
    }
  }
}
```

3.2.1.5 (Extra)How to move the base currency when in commissionable mode

If you create a mijin in commissioned mode, all transaction transmissions, etc. will be charged a fee from the base currency.
For this reason, we will explain the procedure for submitting the base currency balance when creating an account.

Register for a key currency holding account

Note:

If you use | mijin | in AWS Marketplace, select the Outouts tab in the AWS Cloudformation Stack.
Please note the URL of the Key 'mijinLBEndpint'or 'mijinbEndpoint'.
Next, click on the URL for the Value of the Key 'HarvestAddress'.
Please note the value of the private_key in the very first Vaule in the api. (In this case, the value begins with 055E)
For more information, see Deploy mijin on an existing VPC or Create a new VPC and deploy mijin for details.

Import the above private_key in symbol-cli.If you already have a mijin.json file, only balanceAccount can be appended.

```
$ mijin-catapult-tools account generate -r mijin.json -w mijin.json -s -p
90EEBCB77A767F8F5CCCE9D0F89A60CB2D7FCD5FD8F469E2F8BDFC0CDD8B8A2F
2023-01-17T06:55:52.167Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:55:52.167Z [info] : Network: 96
2023-01-17T06:55:52.167Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:55:52.167Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:55:52.167Z [info] : Start Account Generate...
2023-01-17T06:55:52.229Z [info] : Write Config File: mijin.json
2023-01-17T06:55:52.233Z [info] : New Account: {
  "url": "http://localhost:3000",
  "workAccount": {
    "publicKey": "425F06A8870381A00BD83E2D1083BB690F9FCB815F0919DFFA1A53A68E144D06",
    "privateKey": "9E9C660164AB344030DF5E77539952D9F5B380311C978369BA08923F577D8DAB",
    "address": "MC5VTRZO7IGJUG2VPQXBD4GDO4A74YLFUCVZZGI"
  },
  "balanceAccount": {
    "publicKey": "4FAAC9BF9881893CB31BC2065E8A8D0B12364423E2A08DAF4F77E1FEF5D5B2E8",
    "privateKey": "90EEBCB77A767F8F5CCCE9D0F89A60CB2D7FCD5FD8F469E2F8BDFC0CDD8B8A2F",
    "address": "MCPYNELDE5QS63QBRZ2L7OLNZ63YPQJSOCEWXQI"
  },
  "mainAccount": {
    "publicKey": "F2985009341A526B17E954EB6EA3EC77E1A0B33AE31EB29F86A69D9BB283AF67",
    "privateKey": "FF9A552ED35D930378F1E6B349A677771F19A06BCA0C4D5DCA621F354F5C6956",
    "address": "MCV66SABR6MBWT2I56YI3ULCPYWREJFX5DHHGDQ"
  },
  "keylink": {
    "vrf": {
      "publicKey": "C074A57A0EDF633414DDD764C6771926E91596481E0C2CAD4D9C3EFAAA432BCE",
      "privateKey": "9E9C660164AB344030DF5E77539952D9F5B380311C978369BA08923F577D8DAB",
      "address": "MBLYBGODVGLT3HED6EXAQAXJE4CHHVRWGIAS45Y"
    },
```

(continues on next page)

(continued from previous page)

```
    "voting": {
      "publicKey": "F6571CBC420A4EBA09E027AA53E5DD9486642911CDC11E0D3A6D1B2E4BC228D1",
      "privateKey": "FD5744779348F77F6EA6288232D0C8944E2F3C9E24D2E0FB1E93A8C4F9DD9BA5",
      "address": "MD2M6SDIE64O6ZWF2IJ7R4R5RDKOT2FNJDUIYOQ"
    }
  },
  "test1Account": {
    "publicKey": "CB51613497A40D9A256B17932579BC64D5037A04B29737B944965C1ADADD6E04",
    "privateKey": "8D5969EF1796F5F90256C92B5017396E40786ED87995169D4E26C9E5E01D7F8C",
    "address": "MAS36UGDCOGG6GYCBDPX3ROISABSNPZ6JQXMJSA"
  },
  "test2Account": {
    "publicKey": "5CC14799D5B2643914C8E574C8D073A7EE9AE2A405F1339A53612B566498AB1D",
    "privateKey": "FDE625C1D53AF04533FEB06A3556679FC02C4C6246952D3A534EB3E19CF83C56",
    "address": "MC5AH4UGBPPHNCMPTVNSY6LRDPIMEDODS2O373A"
  }
}
```

You can now import into the balanceAccount.
Verify account information.
This account has two Mosaics initially.
Balance Information 268CF9B2D33FBD22 is the base currency (cat.currency) and 4C39D26C386E3182 is Mosaic for Harvest validity.

Warning: This ID will be a different value for each mijin created, and the base currency will be the ID displayed as “currency”: true

```
$ mijin-catapult-tools account info -r mijin.json -t balance
2023-01-17T06:56:51.299Z [info] : mijin URL: http://localhost:3000
2023-01-17T06:56:51.299Z [info] : Network: 96
2023-01-17T06:56:51.299Z [info] : Mosaic Currency Id: 268CF9B2D33FBD22
2023-01-17T06:56:51.299Z [info] : Mosaic Harvest Id: 4C39D26C386E3182
2023-01-17T06:56:51.299Z [info] : Start Account Info
2023-01-17T06:56:51.315Z [info] : balance Account: {
  "publicKey": "4FAAC9BF9881893CB31BC2065E8A8D0B12364423E2A08DAF4F77E1FEF5D5B2E8",
  "address": "MCPYNELDE5QS63QBRZ2L7OLNZ63YPQJSOCEWXQI",
  "mosaics": [
    {
      "id": "268CF9B2D33FBD22",
      "amount": "8998977498000000",
      "currency": true,
      "harvest": false
    },
    {
      "id": "4C39D26C386E3182",
      "amount": "15000000",
      "currency": false,
      "harvest": true
    }
  ],
```

(continues on next page)

(continued from previous page)

```
  "keylink": {
    "vrf": {
      "publicKey": "2A88BA2689D584B03A3D4B829347F0A8B63AF55A8E9F176F1D2327F9E87E22D8"
    },
    "voting": {
      "publicKey": "22BB9DCA05D483E4D4DDE764E8742E741ADA676F461D5F3E6663840C5290320F",
      "startEpoch": 1,
      "endEpoch": 26280
    }
  }
}
```

3.2.2 mijin Catapult(v.2) version up

This chapter describes how to do a minor upgrade of the mijin Catapult(v.2) node.
mijin Catapult(v.2) programs can be upgraded by updating the docker container version and, if necessary, the Config file.
Version upgrade announcements are sent out for paid support customers.

Warning:

Synchronization may fail if there is a major version upgrade or version differences.
As of 2022/9/20 There will be no version upgrade.

3.2.2.1 Step.1

Remote login to the node.
If you want to log in to AWS MarketPlace's mijin Catapult(v.2), you can use mijin Catapult(v.2) EC2 instance login how to.

3.2.2.2 Step.2

Switch to the 'catapult'user running mijin.

```
$ sudo su - catapult
catapult@api1:~$
```

3.2.2.3 Step.3

Go to the directory where the mijin startup files are located.
Note that the directories are different for API and PEER nodes.

Node	Directories
API/Dual	mijin-catapult-package/package/**api**/catapult/
PEER	mijin-catapult-package/package/**peer**/catapult/

For API nodes, move as follows

```
catapult@api1:~$ cd mijin-catapult-package/package/api/catapult/
catapult@api1:~/mijin-catapult-package/package/api/catapult$
```

3.2.2.4 Step.4

Mijin launches multiple containers by docker.
Check multiple containers.

For API nodes

Four containers are running. Make sure that all States are Up. If there is something wrong, the state will be Exit instead of Up and you will be in a down state.

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose ps
          Name                        Command               State           Ports
-------------------------------------------------------------------------------------
 ↪-----
catapult_api-node-broker_1   bash -c /bin/bash /scripts ...   Up
catapult_api-node_1          bash -c perl /scripts/wait ...   Up      0.0.0.0:7900->
 ↪7900/tcp
catapult_db_1                docker-entrypoint.sh bash  ...   Up      27017/tcp
catapult_rest-gateway_1      docker-entrypoint.sh ash - ...   Up      0.0.0.0:3000->
 ↪3000/tcp
```

For PEER node

One container is activated.
Make sure that all States are Up.
If there is some abnormality, it will be Exit instead of Up and will be in a down state.

```
catapult@peer1:~/mijin-catapult-package/package/peer/catapult$ docker-compose ps
        Name                      Command               State           Ports
-------------------------------------------------------------------------------------
catapult_peer-node_1   bash -c /bin/bash /scripts ...   Up      0.0.0.0:7900->7900/tcp
```

3.2.2.5 Step.5

Stop all mijin containers using docker-compose.

All containers stopped

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose down
Stopping catapult_rest-gateway_1 ... done
Stopping catapult_api-node_1     ... done
Stopping catapult_db_1           ... done
Removing catapult_api-node-broker_1 ... done
Removing catapult_rest-gateway_1    ... done
Removing catapult_api-node_1        ... done
Removing catapult_db_1              ... done
Removing network catapult_default
```

After stopping, check the lock file and delete it if any.

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ ls -la /mnt/mijin/blocks/
↪data/*.lock
---------- 1 catapult catapult 0 Jul 14 02:17 /mnt/mijin/blocks/data/broker.lock
---------- 1 catapult catapult 0 Jul 14 02:17 /mnt/mijin/blocks/data/server.lock
```

If a lock file exists after the stop as shown above, it is considered to have stopped abnormally.
Therefore, delete the lock file.

```
$ rm -rf /mnt/mijin/blocks/data/broker.lock /mnt/mijin/blocks/data/server.lock
```

3.2.2.6 Step.6

Modify the docker-compose file.
As an example, we will replace **1.0.0.0** with **1.0.0.1**.

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ sed -i -e s/gcc-1.0.0.0/
↪gcc-1.0.0.1/g docker-compose.yml
```

Note:

Config file may need to be modified.
The procedure will be made available at the time of the upgrade announcement. (No version as of 9/20/20/2022)

3.2.2.7 Step.7

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose up -d
Creating network "catapult_default" with the default driver
Creating catapult_db_1 ... done
Creating catapult_rest-gateway_1    ... done
Creating catapult_api-node-broker_1 ... done
Creating catapult_api-node_1        ... done
```

After startup, execute the items in Step.4 and make sure all containers are Up.

3.2.3 [Archive] mijin account creation (>=1.0.0.0)

This chapter describes the account operations that must be performed in order to operate mijin.

Warning:

Symbol-cli has been archived, so symbol-cli may not be available.
From 1.0.3.4 and onwards, please refer to ./aws_tips_new_move_currency.

Note:

mijin Catapult(v.2) to operate, you create an account and use that account to submit transactions.
Also note that you must have a base currency (cat.currency) balance in your account if you are in commissionable mode as specified during deployment.

3.2.3.1 Installation of symbol-cli

Install symbol-cli from npm.

```
$ sudo npm i -g symbol-cli@1.0.0
/usr/local/bin/symbol-cli -> /usr/local/lib/node_modules/symbol-cli/bin/symbol-cli
+ symbol-cli@1.0.0
updated 1 package in 8.724s
```

3.2.3.2 Create an account

First, let's create two accounts to be used for testing.

item	Description	value
Select the network type	Specify the network. CatapultNetwork value specified at build time	MIJIN/MIJIN_TEST
Do you want to save the account?	Save this account.	yes
Select an import type	Specify the import method for saving again.	PrivateKey
Enter the Symbol node URL.	Specify the mijinLBENdpoint or mijinEndpoint URL on the Outouts tab in the Cloudformation Stack.	<http://xxxxxxx:300>
Enter a profile name	Specify a profile name to invoke the account.	optional
Enter your wallet password	Specify the password for your account	optional

Create the first account (Profile mijin-a)

```
$ symbol-cli account generate

? Select the network type: › - Use arrow-keys. Return to submit.
✓ Select the network type: › MIJIN
✓ Do you want to save the account? ··· yes
✓ Select an import type: › PrivateKey
✓ Enter the Symbol node URL. (Example: http://localhost:3000): ··· http://
↪xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.amazonaws.com:3000
✓ Enter a profile name: ··· mijin-a
✓ Enter your wallet password: ··· ********
✓ Do you want to set the account as the default profile? ··· yes

Account
┌──────────────┬──────────────────────────────────────────────────────────────────┐
│ Property     │ Value                                                            │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Address      │ MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI                    │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key   │ 707902962A0A2E32226243D1E7B98D2DD40261E9D3649543E7C28A0F024D4A38 │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Private Key  │ 2515EDCAAA3985F30D6E758ED139823290DAB11034BF4113849FF5CB9355B9C9 │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Password     │ Test1234                                                         │
```

(continues on next page)

(continued from previous page)

```
└──────────────┴──────────────────────────────────────────────────────────────────┘

SUCCESS Stored mijin-a profile
```

At this point, a check of the mijin-a account shows that it does not exist.
This is because there is no exchange of this address on the blockchain, so its existence cannot be confirmed.

```
$ symbol-cli account info --profile mijin-a
⠏ Processing(node:7718) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
(node:7718) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:7718) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.

ERR {"statusCode":404,"statusMessage":"Not Found","body":"{\"code\":\"ResourceNotFound\
↪",\"message\":\"no resource exists with id 'MA36BR7DCFZT65BQZPTM5QNDEZKSB7HNE4DU6TI'\
↪"}"}

TIP The account has to receive at least one transaction to be recorded on the network
```

Create second account (Profile mijin-b)

```
$ symbol-cli account generate

✓ Select the network type: › MIJIN
✓ Do you want to save the account? ··· yes
✓ Select an import type: › PrivateKey
✓ Enter the Symbol node URL. (Example: http://localhost:3000): ··· http://
↪xxxxxxxxxxxxxxxxxxxx.elb.ap-northeast-1.amazonaws.com:3000
✓ Enter a profile name: ··· mijin-b
✓ Enter your wallet password: ··· ********
✓ Do you want to set the account as the default profile? ··· no

Account
┌──────────────┬──────────────────────────────────────────────────────────────────┐
│ Property     │ Value                                                            │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Address      │ MCLO63-LBWG6V-PLJD4O-MADZ37-W6QXQE-DPC3H3-EGQ                    │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key   │ 2D2AC0FF30FABEFC12CB3FBB2323F8CD079ED1055FAAF2581CA29697130292FA │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Private Key  │ 654065E33D00446F1FAAF2CF7D72CC287BDD91E55E9489AEC42769EDDB7A9759 │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Password     │ Test1234                                                         │
└──────────────┴──────────────────────────────────────────────────────────────────┘

SUCCESS Stored mijin-b profile
```

Same as the first, no mijin-b account exists.

```
$ symbol-cli account info --profile mijin-b

⠸ Processing(node:53) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
(node:53) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:53) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.

ERR {"statusCode":404,"statusMessage":"Not Found","body":"{\"code\":\"ResourceNotFound\
↪",\"message\":\"no resource exists with id 'MCLO63LBWG6VPLJD4OMADZ37W6QXQEDPC3H3EGQ'\
↪"}"}

TIP The account has to receive at least one transaction to be recorded on the network
```

About Fees and Commissions

Fees are charged by transactions issued in the following items and must be paid in the base currency (cat.currency).

- transaction fee
- Mosaic Rental Fee
- Namespace Rental Fees
- Locked guarantee deposit

For mijin offered in the marketplace, the commission mode is set to none as standard.
Without commissions, there is no need to have a base currency balance in your account, so there is nothing to be aware of, but you will need to set the commission setting to 0 in symbol-cli or sdk each time.
If you have created a mijin with commissions, please refer to the section 'How to move the base currency in the mode with commissions'and send the base currency to Profile mijin-a.

3.2.3.3 Create and transfer Mosaic

First, let's create and transfer the basic Mosaic.

Create Mosaic

Issue 1amount of Mosaic(Token) in profile mijin-a account.

item	Description	value
Enter your wallet password	Please specify the password you have set	optional
Do you want a non-expiring mosaic	Specify whether Mosaic is to expire or indefinitely; Yes to indefinitely; No to indefinitely; No to indefinitely.	yes
Enter the mosaic divisibility	Specifies the divisibility of Mosaic, 0 to none.	0
Do you want this mosaic to have a mutable supply?	Specifies whether the maximum number of Mosaic issues can be changed. None.	no
Do you want this mosaic to be transferable?	Specifies whether Mosaic forwarding is allowed.	yes
Do you want this mosaic to be restrictable?	Specifies whether Mosaic restrictions are allowed.	yes
Amount of mosaics units to create	Specify the number of Mosaic issues. In this case, 1 is issued.	1
Enter the maximum fee (absolute amount)	Specifies the transaction fee. This changes with the commission mode. With commission 20000 or so (0.2cat.currency) No commission 0	0
Select the transaction announce mode	Specifies how transactions are announced.	normal

Listing 1: Symbol CLI によるモザイク定義

```
$ symbol-cli transaction mosaic --profile mijin-a

✓ Enter your wallet password: ⋯ ********
✓ Do you want a non-expiring mosaic? ⋯ yes
✓ Enter the mosaic divisibility: ⋯ 0
✓ Do you want this mosaic to have a mutable supply? ⋯ no
✓ Do you want this mosaic to be transferable? ⋯ yes
✓ Do you want this mosaic to be restrictable? ⋯ yes
✓ Amount of mosaics units to create:  ⋯ 1
✓ Enter the maximum fee (absolute amount): ⋯ 0
✓ Select the transaction announce mode: › normal
✓ Do you want to announce this transaction? ⋯ yes

SUCCESS Transaction announced correctly

TIP To check if the network confirms or rejects the transaction, run the command
↪'symbol-cli transaction status'
```

When I check the mijin-a account again, I can see that it has only one MosaicId 3BF3AF8B22CB53D8 in the Balance Information.

```
$ symbol-cli account info --profile mijin-a

⠸ Processing(node:7795) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
(node:7795) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:7795) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
```

(continues on next page)

(continued from previous page)

```
↪SecureContext instead.
⋮ Processing
Account Information
```

Property	Value
Address	MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI
Address Height	959
Public Key	707902962A0A2E32226243D1E7B98D2DD40261E9D3649543E7C28A0F024D4A38
Public Key Height	959
Importance	0
Importance Height	0

```
Balance Information
```

Mosaic Id	Relative Amount	Absolute Amount	Expiration Height
3BF3AF8B22CB53D8	1	1	Never

Transfer Mosaic

Transfer the Mosaic 3BF3AF8B22CB53D8 you just created from Profile mijin-a to mijin-b.

item	value
Mosaic	3BF3AF8B22CB53D8
transfer amount	1
Destination address (Profile mijin-b address)	MCLO63-LBWG6V-PLJD4O-MADZ37-W6QXQE-DPC3H3-EGQ

item	Description	value
Enter your wallet password	Please specify the password you have set	optional
Mosaics to transfer in the format (mosaicId(hex)|@aliasName)::absoluteAmount	転送するモザイク ID（またはエイリアス）にコロンを二つ追加した後、転送 amount を指定します。	**3BF3AF8B22CB53D8::1**
Enter the recipient address or @alias	Specify the forwarding address	MCLO63-LBWG6V-PLJD4O-MADZ37-W6QXQE-DPC3H3-EGQ
Enter a message	A message can be added to the transfer transaction	optional
Enter the maximum fee (absolute amount)	トランザクション手数料を指定します（モードにより異なります）。 - 手数料あり: 約``20000`` (= 0.2cat.currency) - 手数料なし: 0	0
Select the transaction announce mode	Specifies how transactions are announced.	normal

```
$ symbol-cli transaction transfer --profile mijin-a

? Enter your wallet password: › (node:97) [DEP0091] DeprecationWarning: crypto.DEFAULT_
↪ENCODING is deprecated.
(node:97) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:97) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
✓ Enter your wallet password: ⋯ ********
✓ Mosaics to transfer in the format (mosaicId(hex)|@aliasName)::absoluteAmount, (Ex:
sending 1 symbol.xym, @symbol.xym::1000000). Add multiple mosaics separated by commas:
⋯ 3BF3AF8B22CB53D8::1
✓ Enter the recipient address or @alias: ⋯ MCLO63-LBWG6V-PLJD4O-MADZ37-W6QXQE-DPC3H3-
↪EGQ
✓ Enter a message: ⋯
✓ Enter the maximum fee (absolute amount): ⋯ 0
✓ Select the transaction announce mode: › normal
```

TRANSFER	
Max fee:	0
Network type:	MIJIN
Deadline:	2021-05-17 16:40:45.212
Recipient:	MCLO63-LBWG6V-PLJD4O-MADZ37-W6QXQE-DPC3H3-EGQ
Message:	N/A
Mosaic (1/1):	1 3BF3AF8B22CB53D8
Signature details	
Payload:	B00000000000000639B3F893989DC69FA3DF8D9BA294FD787F7644D68918FA8 F5E6A5162FE57CD34B03E944AA48F4B22790A50ECEA4A130EBB89299BDB173A4 556BBAFF8092A20B707902962A0A2E32226243D1E7B98D2DD40261E9D3649543 E7C28A0F024D4A380000000001605441000000000000000009C1F80300E000000 6096EF6D61B1BD57AD23E39801E77FB7A178106F16CFB21A0000010000000000 D853CB228BAFF33B0100000000000000
Hash:	0CAB966B0E7090AA19AE4D4F2BD2334A7F7466E5661107A15F8831EA48A5CE88
Signer:	707902962A0A2E32226243D1E7B98D2DD40261E9D3649543E7C28A0F024D4A38

```
✓ Do you want to announce this transaction? ⋯ yes

SUCCESS Transaction announced correctly

TIP To check if the network confirms or rejects the transaction, run the command
↪'symbol-cli transaction status'
```

Check the status of the profile **mijin-a**.
We can verify that it does not have Mosaic **3BF3AF8B22CB53D8**.

```
symbol-cli account info --profile mijin-a
⠏ Processing(node:108) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
(node:108) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:108) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
⠇ Processing
Account Information
┌──────────────────┬──────────────────────────────────────────────────────────────────┐
│ Property         │ Value                                                            │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address          │ MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI                    │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address Height   │ 959                                                              │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key       │ 707902962A0A2E32226243D1E7B98D2DD40261E9D3649543E7C28A0F024D4A38 │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key Height │ 959                                                             │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance       │ 0                                                                │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance Height │ 0                                                               │
└──────────────────┴──────────────────────────────────────────────────────────────────┘
```

Then check the status of Profile **mijin-b**.
We can verify that it has Mosaic **3BF3AF8B22CB53D8**.

```
symbol-cli account info --profile mijin-b
⠏ Processing(node:119) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
(node:119) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:119) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
⠇ Processing
Account Information
┌──────────────────┬──────────────────────────────────────────────────────────────────┐
│ Property         │ Value                                                            │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address          │ MCLO63-LBWG6V-PLJD4O-MADZ37-W6QXQE-DPC3H3-EGQ                    │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address Height   │ 17912                                                            │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key       │ 0000000000000000000000000000000000000000000000000000000000000000 │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key Height │ 0                                                               │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance       │ 0                                                                │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance Height │ 0                                                               │
└──────────────────┴──────────────────────────────────────────────────────────────────┘
Balance Information
┌──────────────────┬──────────────────┬──────────────────┬──────────────────┐
```

(continues on next page)

(continued from previous page)

```
| Mosaic Id          | Relative Amount | Absolute Amount | Expiration Height |
|--------------------|-----------------|-----------------|-------------------|
| 3BF3AF8B22CB53D8   | 1               | 1               | Never             |
└────────────────────┴─────────────────┴─────────────────┴───────────────────┘
```

3.2.3.4 (Extra)How to move the base currency when in commissionable mode

If you create a mijin in commissioned mode, all transaction transmissions, etc. will be charged a fee from the base currency.
For this reason, we will explain the procedure for submitting the base currency balance when creating an account.

Register for a key currency holding account

Note:

If you use | mijin | in AWS Marketplace, select the Outouts tab in the AWS Cloudformation Stack.
Please note the URL of the Key 'mijinLBEndpint'or 'mijinbEndpoint'.
Next, click on the URL for the Value of the Key 'HarvestAddress'.
Please note the value of the private_key in the very first Vaule in the api. (In this case, the value begins with 055E)
For more information, see Deploy mijin on an existing VPC or Create a new VPC and deploy mijin for details.

Import the above private_key in symbol-cli.

item	Input value
Select the network type	Specify the network. 構築時に指定した `CatapultNetwork` の値 `MIJIN` または `MIJIN_TEST`
Enter the Symbol node URL	控えていた `mijinEndpoint` もしくは ロードバランサーが有効であれば `mijinLBEndpoint` の URL を入力
Enter a profile name	Enter any profile name
Enter your wallet password	Enter any password
Do you want to set the account as the default profile	DefaultProfile にするかどうか、ここでは `Yes` を選択
Select an import type	`PrivateKey`
Enter your account private key	控えていた `private_key` を入力

```
$ symbol-cli profile import

✓ Select the network type: › MIJIN
✓ Enter the Symbol node URL. (Example: http://localhost:3000): … http://MIJIN-
↪CATAPULT-E1-nlb-rest-XXXXXXXXX.elb.ap-northeast-1.amazonaws.com:3000
```

(continues on next page)

(continued from previous page)

```
✓ Enter a profile name: ··· mijin-harvest
✓ Enter your wallet password: ··· ********
✓ Do you want to set the account as the default profile? ··· no
✓ Select an import type: › PrivateKey
✓ Enter your account private key: ···
************************************************************

Account
┌──────────────┬──────────────────────────────────────────────────────────────────┐
│ Property     │ Value                                                            │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Address      │ MAQUY5-KOJVPE-DDCTD6-3SZYHM-EQOFF4-HTUYZU-3WQ                    │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key   │ 29800CB9DF988622AD4B940F578569321F4B7F08127C478A0C0C28ACC61B8A2C │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Private Key  │ 2EC8FF52B5B922E0F509FBEE6CE3C4B3512E9347DB800A76A6EF993C43C0D5BC │
├──────────────┼──────────────────────────────────────────────────────────────────┤
│ Password     │ Test1234                                                         │
└──────────────┴──────────────────────────────────────────────────────────────────┘

SUCCESS Stored mijin-harvest profile
```

Now you are ready to import.
Verify account information.
This account has two Mosaics initially.
Balance Information 04A125F887094D2A is the base currency (cat.currency) and 49DB43B9FA374EF2 is Mosaic for Harvest validity.

Warning: This ID will be different for each creation, and the base currency will be the ID with the largest Amount.

```
$ symbol-cli account info --profile mijin-harvest

⠏ Processing(node:141) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
(node:141) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:141) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
⠇ Processing
Account Information
┌──────────────────┬──────────────────────────────────────────────────────────────────┐
│ Property         │ Value                                                            │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address          │ MAQUY5-KOJVPE-DDCTD6-3SZYHM-EQOFF4-HTUYZU-3WQ                    │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address Height   │ 1                                                                │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key       │ 29800CB9DF988622AD4B940F578569321F4B7F08127C478A0C0C28ACC61B8A2C │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
```

(continues on next page)

(continued from previous page)

```
│ Public Key Height  │ 1                                                                   │
├────────────────────┼─────────────────────────────────────────────────────────────────────┤
│ Importance         │ 2850000                                                             │
├────────────────────┼─────────────────────────────────────────────────────────────────────┤
│ Importance Height  │ 18080                                                               │
└────────────────────┴─────────────────────────────────────────────────────────────────────┘
Balance Information
┌──────────────────┬─────────────────┬──────────────────┬────────────────────┐
│ Mosaic Id        │ Relative Amount │ Absolute Amount  │ Expiration Height  │
├──────────────────┼─────────────────┼──────────────────┼────────────────────┤
│ 04A125F887094D2A │ 1,799,799,999.6 │ 1799799999600000 │ Never              │
├──────────────────┼─────────────────┼──────────────────┼────────────────────┤
│ 49DB43B9FA374EF2 │ 3,000           │ 3000000          │ Never              │
└──────────────────┴─────────────────┴──────────────────┴────────────────────┘
```

Confirmation of base currency

item	Description	value
Enter the mosaic id in hexadecimal format	Information on the target mosaic can be obtained by specifying the mosaic ID.	In the example here 04A125F887094D2A

Divisibility can be checked at Divisibility.
In the following, divisibility is 6.

```
symbol-cli mosaic info
? Enter the mosaic id in hexadecimal format: › (node:163) [DEP0091] DeprecationWarning:
crypto.DEFAULT_ENCODING is deprecated.
(node:163) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:163) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
✓ Enter the mosaic id in hexadecimal format: … 04A125F887094D2A

Mosaic Information
┌────────────────────┬───────────────────────────────────────────────┐
│ Property           │ Value                                         │
├────────────────────┼───────────────────────────────────────────────┤
│ Record Id          │ 609DEB554D4B851AA429AE2C                      │
├────────────────────┼───────────────────────────────────────────────┤
│ Mosaic Id          │ 04A125F887094D2A                              │
├────────────────────┼───────────────────────────────────────────────┤
│ Divisibility       │ 6                                             │
├────────────────────┼───────────────────────────────────────────────┤
│ Transferable       │ true                                          │
├────────────────────┼───────────────────────────────────────────────┤
│ Supply Mutable     │ false                                         │
├────────────────────┼───────────────────────────────────────────────┤
│ Height             │ 1                                             │
```

(continues on next page)

(continued from previous page)

```
┌──────────────────┬──────────────────────────────────────────────────┐
│ Expiration       │ Never                                            │
├──────────────────┼──────────────────────────────────────────────────┤
│ Owner            │ MAO5AR-GSMLGK-ZCDV35-IJWDVL-JFOCZT-XHM3KJ-RHA    │
├──────────────────┼──────────────────────────────────────────────────┤
│ Supply (Absolute)│ 8998999998000000                                 │
├──────────────────┼──────────────────────────────────────────────────┤
│ Supply (Relative)│ 8,998,999,998                                    │
└──────────────────┴──────────────────────────────────────────────────┘
```

Send the base currency

Try to send the base currency 1000cat.currency to the registered account (Profile mijin-a).

item	value
Mosaic	cat.currency
transfer amount	1000000000(1000.000000 since divisibility is 6)
Forwarding address (Profile mijin-a)	MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI

item	Description	value
Enter your wallet password	Please specify the password you have set	optional
Mosaics to transfer in the format (mosaicId(hex)|@aliasName)::absoluteAmount	After adding two colons to the mosaic ID (or alias) to be transferred, specify the transfer AMOUNT.	@cat.currency::1000000000
Enter the recipient address or @alias	Specify the forwarding address	MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI
Enter a message	A message can be added to the transfer transaction	optional
Enter the maximum fee (absolute amount)	Specifies the transaction fee. This changes with the commission mode. With commission 20000 or so (0.2cat.currency) No commission 0	0
Select the transaction announce mode	Specifies how transactions are announced.	normal

Here the transaction fee is set to 0, but if there is a fee, the transaction fee will be charged, so you will need about 200000 instead of 0.
The fee calculation method will be the same as for Symbol.

<https://docs.symbol.dev/concepts/fees.html#transaction-fee>

```
$ symbol-cli transaction transfer --profile mijin-harvest

? Enter your wallet password: › (node:196) [DEP0091] DeprecationWarning: crypto.
↪DEFAULT_ENCODING is deprecated.
(node:196) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
```

(continues on next page)

(continued from previous page)

```
tls.createSecureContext instead.
(node:196) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
✓ Enter your wallet password: ··· ********
✓ Mosaics to transfer in the format (mosaicId(hex)|@aliasName)::absoluteAmount, (Ex:
sending 1 symbol.xym, @symbol.xym::1000000). Add multiple mosaics separated by commas:
··· @cat.currency::1000000000
✓ Enter the recipient address or @alias: ··· MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-
↪6TI
✓ Enter a message: ···
✓ Enter the maximum fee (absolute amount): ··· 0
✓ Select the transaction announce mode: › normal
```

TRANSFER	
Max fee:	0
Network type:	MIJIN
Deadline:	2021-05-17 17:46:42.643
Recipient:	MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI
Message:	N/A
Mosaic (1/1):	1,000,000,000 cat.currency (85BBEA6CC462B244)
Signature details	
Payload:	B00000000000000071E0E34240C563025C466C31BFA995D6DA4CADFFB0AD29A3 B2EFF280AD596BDA4CF5DC01B3654D9C34F56346B5DF41112EF27858C0ED1FCE CF409B5A450FC30E29800CB9DF988622AD4B940F578569321F4B7F08127C478A 0C0C28ACC61B8A2C0000000001605441000000000000000005382BC300E000000 6037E0C7E311733F7430CBE6CEC1A3265520FCED27074F4D00000100000000000 44B262C46CEABB8500CA9A3B00000000
Hash:	864EF99D58E8DA837879D85DE08DF29398766E04F967F09732A2FD02115469FB
Signer:	29800CB9DF988622AD4B940F578569321F4B7F08127C478A0C0C28ACC61B8A2C

```
✓ Do you want to announce this transaction? ··· yes

SUCCESS Transaction announced correctly

TIP To check if the network confirms or rejects the transaction, run the command
↪'symbol-cli transaction status'
```

Check your profile mijin-a account.
You can confirm that you have 1000cat.currency.

```
symbol-cli account info --profile mijin
⠏ Processing(node:207) [DEP0091] DeprecationWarning: crypto.DEFAULT_ENCODING is
deprecated.
```

(continues on next page)

(continued from previous page)

```
(node:207) [DEP0010] DeprecationWarning: crypto.createCredentials is deprecated. Use
tls.createSecureContext instead.
(node:207) [DEP0011] DeprecationWarning: crypto.Credentials is deprecated. Use tls.
↪SecureContext instead.
⁝ Processing
Account Information
┌──────────────────┬──────────────────────────────────────────────────────────────────┐
│ Property         │ Value                                                            │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address          │ MA36BR-7DCFZT-65BQZP-TM5QND-EZKSB7-HNE4DU-6TI                    │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Address Height   │ 959                                                              │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key       │ 707902962A0A2E32226243D1E7B98D2DD40261E9D3649543E7C28A0F024D4A38 │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Public Key Height │ 959                                                             │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance       │ 0                                                                │
├──────────────────┼──────────────────────────────────────────────────────────────────┤
│ Importance Height │ 0                                                               │
└──────────────────┴──────────────────────────────────────────────────────────────────┘
Balance Information
┌──────────────┬─────────────────┬─────────────────┬─────────────────┐
│ Mosaic Id    │ Relative Amount │ Absolute Amount │ Expiration Height │
├──────────────┼─────────────────┼─────────────────┼─────────────────┤
│ 04A125F887094D2A │ 1,000       │ 1000000000      │ Never           │
└──────────────┴─────────────────┴─────────────────┴─────────────────┘
```

3.3 Troubleshooting

mijin Catapult(v.2) troubleshooting is summarized here.

3.3.1 mijin Catapult(v.2) node resynchronization

mijin Catapult(v.2) node has stopped synchronizing for some reason, etc. This section describes the resynchronization procedure.

3.3.1.1 target

- Node blocking is not progressing.
- No longer able to check in /node/peers.
- Some containers are Exit in the docker-compose ps command and cannot be recovered.

3.3.1.2 Step.1

Remote login to the node.
If you want to log in to AWS MarketPlace's mijin Catapult(v.2), you can use mijin Catapult(v.2) EC2 instance login how to.

3.3.1.3 Step.2

Switch to the 'catapult'user running mijin.

```
$ sudo su - catapult
catapult@api1:~$
```

3.3.1.4 Step.3

Go to the directory where the mijin startup files are located.
Note that the directories are different for API and PEER nodes.

Node	Directory
API/Dual	mijin-catapult-package/package/**api**/catapult/
PEER	mijin-catapult-package/package/**peer**/catapult/

For API nodes, move as follows

```
catapult@api1:~$ cd mijin-catapult-package/package/api/catapult/
catapult@api1:~/mijin-catapult-package/package/api/catapult$
```

3.3.1.5 Step.4

Mijin launches multiple containers by docker.
Check multiple containers.

For API nodes

Four containers are running. Make sure that all States are Up. If there is something wrong, the state will be Exit instead of Up and you will be in a down state.

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose ps
          Name                        Command               State           Ports
-------------------------------------------------------------------------------------
 ↪-----
catapult_api-node-broker_1   bash -c /bin/bash /scripts ...   Up
catapult_api-node_1          bash -c perl /scripts/wait ...   Up      0.0.0.0:7900->
 ↪7900/tcp
catapult_db_1                docker-entrypoint.sh bash  ...   Up      27017/tcp
catapult_rest-gateway_1      docker-entrypoint.sh ash - ...   Up      0.0.0.0:3000->
 ↪3000/tcp
```

In the case of PEER node

One container is activated.
Make sure that all States are Up.
If there is some abnormality, it will be Exit instead of Up and will be in a down state.

```
catapult@peer1:~/mijin-catapult-package/package/peer/catapult$ docker-compose ps
        Name                      Command               State           Ports
-------------------------------------------------------------------------------------
catapult_peer-node_1   bash -c /bin/bash /scripts ...   Up      0.0.0.0:7900->7900/tcp
```

3.3.1.6 Step.5

Stop and re-start all mijin containers using docker-compose.

All containers stopped

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose down
Stopping catapult_rest-gateway_1 ... done
Stopping catapult_api-node_1     ... done
Stopping catapult_db_1           ... done
Removing catapult_api-node-broker_1 ... done
Removing catapult_rest-gateway_1    ... done
Removing catapult_api-node_1        ... done
Removing catapult_db_1              ... done
Removing network catapult_default
```

After stopping, check the lock file and delete it if any.

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ ls -la /mnt/mijin/blocks/
↪data/*.lock
---------- 1 catapult catapult 0 Jul 14 02:17 /mnt/mijin/blocks/data/broker.lock
---------- 1 catapult catapult 0 Jul 14 02:17 /mnt/mijin/blocks/data/server.lock
```

If a lock file exists after the stop as shown above, it is considered to have stopped abnormally.
Therefore, delete the lock file.

```
$ rm -rf /mnt/mijin/blocks/data/broker.lock /mnt/mijin/blocks/data/server.lock
```

re-start

```
$ docker-compose up -d
Creating network "catapult_default" with the default driver
Creating catapult_db_1 ... done
Creating catapult_rest-gateway_1    ... done
Creating catapult_api-node-broker_1 ... done
Creating catapult_api-node_1        ... done
```

After startup, execute the items in Step.4 and make sure all containers are Up.
If you are in a similar situation, go to Step.9.

3.3.1.7 Step.6

Reset and recover block data on a node. If in a redundant state, data can be automatically retrieved and recovered from other nodes.

All containers stopped

```
$ docker-compose down
Stopping catapult_rest-gateway_1 ... done
Stopping catapult_api-node_1     ... done
Stopping catapult_db_1           ... done
Removing catapult_api-node-broker_1 ... done
Removing catapult_rest-gateway_1    ... done
Removing catapult_api-node_1        ... done
Removing catapult_db_1              ... done
Removing network catapult_default
```

Delete block data and mongo data

The catapult user that mijin is running as does not have sudo privileges and cannot delete directories.
Therefore, grant the catapult user sudo privileges.
This sudo setting is required only for the first time.

```
# catapult ユーザーからログアウト Log out of catapult user
$ logout
# root ユーザーにスイッチ Switch to root user
$ sudo su -
# catapult ユーザーに sudo 権限を付与する Grant sudo privileges to the catapult user
# echo "catapult ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/catapult
# catapult ユーザーにスイッチ Switch to catapult user
# su - catapult
```

Delete block data with sudo privileges.

```
$ sudo rm -rf /mnt/mijin/blocks/data
```

If API nodes are targeted, mongo data deletion is also required

```
$ sudo rm -rf /mnt/mijin/mongo/db
```

Execute recovery commands

Go to the file with docker-compose and run the RECOVER script.

When the API node is the target

```
catapult@api1:~$ cd mijin-catapult-package/package/api/catapult/
catapult@api1:~/mijin-catapult-package/package/api/catapult$
```

When the PEER node is the target

```
catapult@peer1:~$ cd mijin-catapult-package/package/peer/catapult/
catapult@peer1:~/mijin-catapult-package/package/peer/catapult$
```

Execute scripts. (Common)

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ bash scripts/recover.sh
2021/07/14 02:52 Start: mijin Recovery
2021/07/14 02:52 Check: /home/catapult/mijin-catapult-package/package/api/catapult
2021/07/14 02:52 Check: /home/catapult/mijin-catapult-package/package/api/catapult OK
2021/07/14 02:52 Check: Started docker?
2021/07/14 02:52 Check: Started docker Stop OK
2021/07/14 02:52 Check: Block Directory
2021/07/14 02:52 Check: Block Directory Empty OK
2021/07/14 02:52 Start: Make Block Directory
2021/07/14 02:52 Check: mongo Directory
2021/07/14 02:52 Check: mongo Directory OK
2021/07/14 02:52 Start: Make mongo Directory
2021/07/14 02:52 Start: Create mongo Init Data
about to fork child process, waiting until server is ready for connections.
forked process: 10
child process started successfully, parent exiting
[+] Preparing db
MongoDB shell version v4.2.5
connecting to: mongodb://localhost:27017/catapult?compressors=disabled&
↪gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c296d026-dfdf-4ea2-ba03-91d7404e21c9") }
MongoDB server version: 4.2.5
Loading LockHash
Loading LockSecret
Loading Metadata
Loading Mosaic
Loading Multisig
Loading Namespace
Loading RestrictionAccount
Loading RestrictionMosaic
===== accountRestrictions INDEXES =====
{ "_id" : 1 }
{ "accountRestrictions.address" : 1 }
===== accounts INDEXES =====
{ "_id" : 1 }
{ "account.publicKey" : 1 }
{ "account.address" : 1 }
===== addressResolutionStatements INDEXES =====
```

(continues on next page)

(continued from previous page)

```
{ "_id" : 1 }
{ "statement.height" : 1, "statement.unresolved" : 1 }
===== blocks INDEXES =====
{ "_id" : 1 }
{ "block.signerPublicKey" : 1 }
{ "block.timestamp" : -1 }
{ "block.height" : -1 }
{ "block.type" : 1, "block.height" : -1 }
{ "block.signerPublicKey" : 1, "block.height" : -1 }
{ "block.beneficiaryAddress" : 1, "block.height" : -1 }
===== finalizedBlocks INDEXES =====
{ "_id" : 1 }
{ "block.finalizationEpoch" : -1 }
{ "block.height" : -1 }
===== hashLocks INDEXES =====
{ "_id" : 1 }
{ "lock.hash" : 1 }
{ "lock.ownerAddress" : 1 }
===== metadata INDEXES =====
{ "_id" : 1 }
{ "metadataEntry.compositeHash" : 1 }
{
        "metadataEntry.sourceAddress" : 1,
        "metadataEntry.metadataType" : 1,
        "metadataEntry.scopedMetadataKey" : 1
}
{
        "metadataEntry.targetAddress" : 1,
        "metadataEntry.metadataType" : 1,
        "metadataEntry.scopedMetadataKey" : 1
}
===== mosaicResolutionStatements INDEXES =====
{ "_id" : 1 }
{ "statement.height" : 1, "statement.unresolved" : 1 }
===== mosaicRestrictions INDEXES =====
{ "_id" : 1 }
{ "mosaicRestrictionEntry.compositeHash" : 1 }
===== mosaics INDEXES =====
{ "_id" : 1 }
{ "mosaic.id" : 1 }
{ "mosaic.ownerAddress" : 1 }
===== multisigs INDEXES =====
{ "_id" : 1 }
{ "multisig.accountAddress" : 1 }
===== namespaces INDEXES =====
{ "_id" : 1 }
{ "namespace.level0" : 1, "meta.index" : 1, "namespace.depth" : 1 }
{
        "meta.latest" : -1,
        "meta.index" : 1,
        "namespace.level0" : 1,
        "namespace.depth" : 1
}
{ "meta.latest" : -1, "namespace.level1" : 1, "namespace.depth" : 1 }
{ "meta.latest" : -1, "namespace.level2" : 1, "namespace.depth" : 1 }
{ "meta.latest" : -1, "namespace.ownerAddress" : 1 }
===== partialTransactions INDEXES =====
```

(continues on next page)

(continued from previous page)

```
{ "_id" : 1 }
{ "transaction.signerPublicKey" : 1, "_id" : -1 }
{ "transaction.recipientAddress" : 1, "_id" : -1 }
{ "meta.hash" : 1 }
{ "meta.addresses" : 1 }
{ "meta.aggregateId" : 1 }
{ "meta.aggregateHash" : 1 }
===== secretLocks INDEXES =====
{ "_id" : 1 }
{ "lock.compositeHash" : 1 }
{ "lock.ownerAddress" : 1 }
===== system.profile INDEXES =====
===== transactionStatements INDEXES =====
{ "_id" : 1 }
{
        "statement.height" : 1,
        "statement.source.primaryId" : 1,
        "statement.source.secondaryId" : 1
}
===== transactionStatuses INDEXES =====
{ "_id" : 1 }
{ "status.hash" : 1 }
{ "status.deadline" : -1 }
===== transactions INDEXES =====
{ "_id" : 1 }
{ "transaction.signerPublicKey" : 1, "_id" : -1 }
{ "transaction.recipientAddress" : 1, "_id" : -1 }
{ "meta.hash" : 1 }
{ "meta.addresses" : 1 }
{ "meta.aggregateId" : 1 }
{ "meta.height" : -1 }
{ "transaction.deadline" : -1 }
{ "transaction.cosignatures.signerPublicKey" : 1 }
{ "transaction.id" : 1, "transaction.type" : 1 }
===== unconfirmedTransactions INDEXES =====
{ "_id" : 1 }
{ "transaction.signerPublicKey" : 1, "_id" : -1 }
{ "transaction.recipientAddress" : 1, "_id" : -1 }
{ "meta.hash" : 1 }
{ "meta.addresses" : 1 }
{ "meta.aggregateId" : 1 }
{ "meta.aggregateHash" : 1 }
bye
[.] (exit code: 0)
/
[+] db prepared, checking account indexes
MongoDB shell version v4.2.5
connecting to: mongodb://localhost:27017/catapult?compressors=disabled&
↪gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a9d9d654-d0c9-4760-bf52-94138f4c6871") }
MongoDB server version: 4.2.5
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
```

(continues on next page)

(continued from previous page)

```
                "name" : "_id_",
                "ns" : "catapult.accounts"
        },
        {
                "v" : 2,
                "key" : {
                        "account.publicKey" : 1
                },
                "name" : "account.publicKey_1",
                "ns" : "catapult.accounts"
        },
        {
                "v" : 2,
                "unique" : true,
                "key" : {
                        "account.address" : 1
                },
                "name" : "account.address_1",
                "ns" : "catapult.accounts"
        }
]
2021-07-14T02:52:54.345+0000 I  CONTROL  [main] Automatically disabling TLS 1.0, to
force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2021-07-14T02:52:54.349+0000 W  ASIO     [main] No TransportLayer configured during
NetworkInterface startup
killing process with pid: 10
2021/07/14 02:52 Start: mijin docker-compose
Creating network "catapult_default" with the default driver
Creating catapult_db_1 ... done
Creating catapult_api-node-broker_1 ... done
Creating catapult_rest-gateway_1    ... done
Creating catapult_api-node_1        ... done
2021/07/14 02:52 End: ALL Success
```

3.3.1.8 Step.7

Check the operation.

Confirmation of container operation

Make sure that all States are Up.

```
$ docker-compose ps
          Name                        Command               State           Ports
-------------------------------------------------------------------------------------
 ↪-----
catapult_api-node-broker_1   bash -c /bin/bash /scripts ...   Up
catapult_api-node_1          bash -c perl /scripts/wait ...   Up      0.0.0.0:7900->
 ↪7900/tcp
catapult_db_1                docker-entrypoint.sh bash  ...   Up      27017/tcp
catapult_rest-gateway_1      docker-entrypoint.sh ash - ...   Up      0.0.0.0:3000->
 ↪3000/tcp
```

Check to see if the block is progressing.

Check to see if the blocks are stacked.
Run it on other nodes to see if they are progressing in the same block.

For API nodes

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose logs --
↪tail=20 api-node| grep heights
api-node_1          | 2021-07-14 02:59:36.958509 0x00007f95631c8700: <info>
(chain::ChainSynchronizer.cpp@217) peer returned 1 blocks (heights 400 - 400)
api-node_1          | 2021-07-14 02:59:36.958645 0x00007f95631c8700: <debug>
(disruptor::Disruptor.cpp@43) disruptor queuing element 27 (1 blocks (heights 400 -
400) [00000000] from Remote_Pull with size 424B)
api-node_1          | 2021-07-14 02:59:37.046493 0x00007f953bfff700: <info>
(disruptor::ConsumerDispatcher.cpp@44) completing processing of element 27 (1 blocks
(heights 400 - 400) [F15ACAA0] from Remote_Pull with size 424B), last consumer is 0
elements behind
```

In the case of PEER node

```
$ docker-compose logs --tail=20 peer-node| grep heights
peer-node_1   | 2021-07-14 03:00:26.168343 0x00007fb1c396a700: <debug>
(disruptor::Disruptor.cpp@43) disruptor queuing element 942 (1 blocks (heights 403 -
403) [00000000] from Remote_Push with size 376B)
peer-node_1   | 2021-07-14 03:00:26.257224 0x00007fb1a57fa700: <info>
(disruptor::ConsumerDispatcher.cpp@44) completing processing of element 942 (1 blocks
(heights 403 - 403) [5C675B6B] empty from Remote_Push with size 376B), last consumer is
0 elements behind
peer-node_1   | 2021-07-14 03:00:26.317777 0x00007fb1c3169700: <debug>
(disruptor::Disruptor.cpp@43) disruptor queuing element 943 (1 blocks (heights 403 -
403) [00000000] from Remote_Push with size 376B)
peer-node_1   | 2021-07-14 03:00:26.368263 0x00007fb1a57fa700: <info>
(disruptor::ConsumerDispatcher.cpp@44) completing processing of element 943 (1 blocks
(heights 403 - 403) [5C675B6B] from Remote_Push with size 376B), last consumer is 0
elements behind
```

For API nodes, check if rest can connect

Check if node information can be obtained.

```
$ curl -Ss http://localhost:3000/node/info | jq -r
{
  "version": 16777216,
  "publicKey": "E4BF3706483B4D42243F3DCB2625021C3E3AE7C253CC466154EEDF9775012C20",
  "networkGenerationHashSeed":
↪"2DE20B93EBE048A3BA132CC9874BCABBC21C87E18FE9836B8D5D002E57640D4B",
  "roles": 70,
  "port": 7900,
  "networkIdentifier": 96,
  "host": "api1.mijin.internal",
  "friendlyName": "api1.mijin.internal",
  "nodePublicKey": "27E7EEAF5819493D60CA848BAA48145A1A97DF63596ED41394563C791303C778"
}
```

3.3.2 Check the node log for mijin Catapult(v.2).

mijin Catapult(v.2) This is the procedure up to checking the logs of a node.

3.3.2.1 target

- Node blocking is not progressing.
- There is an error and I can't figure out the cause.

3.3.2.2 Step.1

Remote login to the node.
If you want to log in to AWS MarketPlace's mijin Catapult(v.2), you can use mijin Catapult(v.2) EC2 instance login how to.

3.3.2.3 Step.2

Switch to the 'catapult'user running mijin.

```
$ sudo su - catapult
catapult@api1:~$
```

3.3.2.4 Step.3

Go to the directory where the mijin startup files are located.
Note that the directories are different for API and PEER nodes.

Node	Directory
API/Dual	mijin-catapult-package/package/**api**/catapult/
PEER	mijin-catapult-package/package/**peer**/catapult/

For API nodes, move as follows

```
catapult@api1:~$ cd mijin-catapult-package/package/api/catapult/
catapult@api1:~/mijin-catapult-package/package/api/catapult$
```

3.3.2.5 Step.4

Mijin launches multiple containers by docker. Check multiple containers.

For API nodes

Four containers are activated.
Make sure that all States are Up.

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose ps
          Name                        Command               State           Ports
-------------------------------------------------------------------------------------
↪-----
catapult_api-node-broker_1   bash -c /bin/bash /scripts ...   Up
catapult_api-node_1          bash -c perl /scripts/wait ...   Up      0.0.0.0:7900->
↪7900/tcp
catapult_db_1                docker-entrypoint.sh bash  ...   Up      27017/tcp
catapult_rest-gateway_1      docker-entrypoint.sh ash - ...   Up      0.0.0.0:3000->
↪3000/tcp
```

For PEER node

One container is activated.
Make sure that all States are Up.

```
catapult@peer1:~/mijin-catapult-package/package/peer/catapult$ docker-compose ps
        Name                      Command               State           Ports
-------------------------------------------------------------------------------
catapult_peer-node_1   bash -c /bin/bash /scripts ...   Up      0.0.0.0:7900->7900/tcp
```

3.3.2.6 Step.5

The logs can be checked with the following command using docker-compose.

```
docker-compose logs
```

To learn more about docker-compose, please visit the following sites
https://docs.docker.com/compose/

Below is an example of how to check.

Check the last XX line logs for each container

```
docker-compose logs --tail=10
```

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose logs --
↪tail=10
Attaching to catapult_rest-gateway_1, catapult_api-node_1, catapult_api-node-broker_1,
catapult_db_1
api-node-broker_1  | 2021-06-14 00:41:32.447154 0x00007f59efb66700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 1 messagesfrom /data/
↪spool/block_change
api-node-broker_1  | 2021-06-14 00:41:32.447137 0x00007f59ef365700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 2 messagesfrom /data/
↪spool/state_change
api-node-broker_1  | 2021-06-14 00:41:50.955992 0x00007f59efb66700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 1 messagesfrom /data/
↪spool/block_change
api-node-broker_1  | 2021-06-14 00:41:50.955975 0x00007f59ef365700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 2 messagesfrom /data/
↪spool/state_change
api-node-broker_1  | 2021-06-14 00:42:05.963816 0x00007f59efb66700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 2 messagesfrom /data/
↪spool/state_change
api-node-broker_1  | 2021-06-14 00:42:05.966239 0x00007f59ef365700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 1 messagesfrom /data/
↪spool/block_change
api-node-broker_1  | 2021-06-14 00:42:23.972546 0x00007f59efb66700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 2 messagesfrom /data/
↪spool/state_change
api-node-broker_1  | 2021-06-14 00:42:23.972928 0x00007f59ef365700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 1 messagesfrom /data/
↪spool/block_change
api-node-broker_1  | 2021-06-14 00:42:41.982065 0x00007f59ef365700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 1 messagesfrom /data/
↪spool/block_change
api-node-broker_1  | 2021-06-14 00:42:41.982048 0x00007f59efb66700: <debug>
(subscribers::BrokerMessageReaders.h@90) preparing to process 2 messagesfrom /data/
↪spool/state_change
api-node_1         | 2021-06-14 00:42:41.523496 0x00007f1bda429700: <debug>
(chain::CompareChains.cpp@119) comparing chain scores: 7676859692801638166 (local) vs
7676974281840495032 (remote)
api-node_1         | 2021-06-14 00:42:41.523569 0x00007f1bda429700: <debug>
(chain::CompareChains.cpp@145) comparing hashes with local height 67006,starting height
66976, max hashes 1440
api-node_1         | 2021-06-14 00:42:41.529042 0x00007f1bdac2a700: <debug>
(chain::ChainSynchronizer.cpp@309) pulling blocks from remote with common height 67006
(fork depth = 0) from DEC1EF1767E76BC31DF2FDADC75C23F6FDA6ECCB22554E4F4C790F81F869F797
@ 10.0.3.199
api-node_1         | 2021-06-14 00:42:41.618591 0x00007f1bdac2a700: <info>
(chain::ChainSynchronizer.cpp@217) peer returned 1 blocks (heights 67007 - 67007)
api-node_1         | 2021-06-14 00:42:41.618764 0x00007f1bdac2a700: <debug>
(chain::ChainSynchronizer.cpp@223) completing chain synchronization with1 blocks (fork
depth = 0)
api-node_1         | 2021-06-14 00:42:41.618805 0x00007f1bdac2a700: <debug>
(disruptor::Disruptor.cpp@43) disruptor queuing element 66554 (1 blocks (heights 67007
- 67007) [00000000] from Remote_Pull with size 376B)
api-node_1         | 2021-06-14 00:42:41.618960 0x00007f1bdac2a700: <info>
```

(continues on next page)

(continued from previous page)

```
(chain::RemoteApiForwarder.h@69) completed 'synchronizer task' (peer2.mijin.internal @
peer2.mijin.internal:7900) with result Success
api-node_1          | 2021-06-14 00:42:41.676588 0x00007f1bb0ff9700: <debug>
(cache::SupplementalDataStorage.cpp@32) wrote last recalculation height 67000 last
finalized height 66976 dynamic fee multiplier 0 total transactions 34 (score = [0,
7676974281840495032], height = 67007)
api-node_1          | 2021-06-14 00:42:41.714555 0x00007f1bb07f8700: <info>
(disruptor::ConsumerDispatcher.cpp@44) completing processing of element 66554 (1 blocks
(heights 67007 - 67007) [833B95DA] from Remote_Pull with size 376B), last consumer is 0
elements behind
api-node_1          | 2021-06-14 00:42:50.247964 0x00007f1bdac2a700: <debug>
(chain::RoundContext.cpp@89) not completable - Erv == g(Vrv) and descendant can reach
g(Crv) (total weight 15000000, cumulative precommit weight 3000000)
db_1                | 2021-06-02T09:37:24.913+0000 I  NETWORK  [listener] connection
accepted from 172.20.0.9:47924 #16 (15 connections now open)
db_1                | 2021-06-02T09:37:24.913+0000 I  NETWORK  [conn16] received client
metadata from 172.20.0.9:47924 conn16: { driver: { name: "nodejs", version: "3.6.0" },
os: { type: "Linux", name: "linux", architecture: "x64", version: "5.4.0-1029-aws" },
platform: "'Node.js v12.18.1, LE (legacy)" }
db_1                | 2021-06-02T09:39:05.968+0000 I  NETWORK  [listener] connection
accepted from 172.20.0.9:47942 #17 (16 connections now open)
db_1                | 2021-06-02T09:39:05.969+0000 I  NETWORK  [conn17] received client
metadata from 172.20.0.9:47942 conn17: { driver: { name: "nodejs", version: "3.6.0" },
os: { type: "Linux", name: "linux", architecture: "x64", version: "5.4.0-1029-aws" },
platform: "'Node.js v12.18.1, LE (legacy)" }
db_1                | 2021-06-02T09:42:43.866+0000 I  NETWORK  [listener] connection
accepted from 172.20.0.9:47948 #18 (17 connections now open)
db_1                | 2021-06-02T09:42:43.869+0000 I  NETWORK  [conn18] received client
metadata from 172.20.0.9:47948 conn18: { driver: { name: "nodejs", version: "3.6.0" },
os: { type: "Linux", name: "linux", architecture: "x64", version: "5.4.0-1029-aws" },
platform: "'Node.js v12.18.1, LE (legacy)" }
db_1                | 2021-06-08T05:37:00.594+0000 I  NETWORK  [listener] connection
accepted from 172.20.0.9:58874 #19 (18 connections now open)
db_1                | 2021-06-08T05:37:00.596+0000 I  NETWORK  [conn19] received client
metadata from 172.20.0.9:58874 conn19: { driver: { name: "nodejs", version: "3.6.0" },
os: { type: "Linux", name: "linux", architecture: "x64", version: "5.4.0-1029-aws" },
platform: "'Node.js v12.18.1, LE (legacy)" }
db_1                | 2021-06-12T02:52:21.305+0000 I  NETWORK  [listener] connection
accepted from 172.20.0.9:37814 #20 (19 connections now open)
db_1                | 2021-06-12T02:52:21.310+0000 I  NETWORK  [conn20] received client
metadata from 172.20.0.9:37814 conn20: { driver: { name: "nodejs", version: "3.6.0" },
os: { type: "Linux", name: "linux", architecture: "x64", version: "5.4.0-1029-aws" },
platform: "'Node.js v12.18.1, LE (legacy)" }
rest-gateway_1      | > node _build/index.js "/userconfig/rest.json"
rest-gateway_1      |
rest-gateway_1      | [winston] Attempt to write logs with no transports {"message":
↪"loading config from /userconfig/rest.json","level":"info"}
rest-gateway_1      | info: loading config from /userconfig/rest.json
rest-gateway_1      | verbose: finished loading rest server config {"network":{"name":
↪"mijin","description":"mijin network"},"port":3000,"crossDomain":{"allowedHosts":["*
↪"],"allowedMethods":["GET","POST","PUT","OPTIONS"]},"extensions":["accountLink",
↪"aggregate","lockHash","lockSecret","mosaic","metadata","multisig","namespace",
↪"receipts","restrictions","transfer"],"db":{"url":"mongodb://db:27017/","name":
↪"catapult","pageSizeMin":10,"pageSizeMax":100,"maxConnectionAttempts":7,
↪"baseRetryDelay":750},"apiNode":{"host":"api-node","port":7900,
↪"tlsClientCertificatePath":"/userconfig/resources/cert/node.crt.pem",
↪"tlsClientKeyPath":"/userconfig/resources/cert/node.key.pem","tlsCaCertificatePath":
```

(continues on next page)

(continued from previous page)

```
↪"/userconfig/resources/cert/ca.cert.pem","timeout":1000,"networkPropertyFilePath":"/
↪api-node-config/config-network.properties","nodePropertyFilePath":"/api-node-config/
↪config-node.properties"},"websocket":{"mq":{"host":"api-node-broker","port":7902,
↪"monitorInterval":500,"connectTimeout":10000,"monitorLoggingThrottle":60000},
↪"allowOptionalAddress":true},"throttling":{"burst":100,"rate":30},"logging":{"console
↪":{"formats":["colorize","simple"],"level":"verbose","handleExceptions":true},"file":
↪{"formats":["prettyPrint"],"level":"verbose","handleExceptions":true,"filename":
↪"catapult-rest.log","maxsize":20971520,"maxFiles":100}},"numBlocksTransactionFeeStats
↪":300,"timestamp":"2021-06-02T09:27:19.336Z"}
rest-gateway_1     | info: connecting to mongodb://db:27017/ (database:catapult) {
↪"timestamp":"2021-06-02T09:27:19.407Z"}
rest-gateway_1     | (node:24) DeprecationWarning: current Server Discovery and
Monitoring engine is deprecated, and will be removed in a future version. To use the new
Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the
MongoClient constructor.
rest-gateway_1     | verbose: connected to mongo at mongodb://db:27017/catapult {
↪"timestamp":"2021-06-02T09:27:19.443Z"}
rest-gateway_1     | info: registering routes {"timestamp":"2021-06-02T09:27:19.445Z"}
rest-gateway_1     | info: listening on port 3000 {"timestamp":"2021-06-02T09:27:19.462Z
↪"}
```

Checks for each specified container

By specifying the hostname of a container, the log output is focused on a specific container.
Host names are fixed.

- api-node
- peer-node
- api-node-broker
- db
- rest-gateway

```
docker-compose logs [コンテナ名 Container name]
```

```
catapult@api1:~/mijin-catapult-package/package/api/catapult$ docker-compose logs rest-
↪gateway
Attaching to catapult_rest-gateway_1
rest-gateway_1     |
rest-gateway_1     | > catapult-api-rest@0.0.0 start /app/catapult-rest/rest
rest-gateway_1     | > node _build/index.js "/userconfig/rest.json"
rest-gateway_1     |
rest-gateway_1     | [winston] Attempt to write logs with no transports {"message":
↪"loading config from /userconfig/rest.json","level":"info"}
rest-gateway_1     | info: loading config from /userconfig/rest.json
rest-gateway_1     | verbose: finished loading rest server config {"network":{"name":
↪"mijin","description":"mijin network"},"port":3000,"crossDomain":{"allowedHosts":["*
↪"],"allowedMethods":["GET","POST","PUT","OPTIONS"]},"extensions":["accountLink",
↪"aggregate","lockHash","lockSecret","mosaic","metadata","multisig","namespace",
↪"receipts","restrictions","transfer"],"db":{"url":"mongodb://db:27017/","name":
↪"catapult","pageSizeMin":10,"pageSizeMax":100,"maxConnectionAttempts":7,
↪"baseRetryDelay":750},"apiNode":{"host":"api-node","port":7900,
```

(continues on next page)

(continued from previous page)

```
↪"tlsClientCertificatePath":"/userconfig/resources/cert/node.crt.pem",
↪"tlsClientKeyPath":"/userconfig/resources/cert/node.key.pem","tlsCaCertificatePath":
↪"/userconfig/resources/cert/ca.cert.pem","timeout":1000,"networkPropertyFilePath":"/
↪api-node-config/config-network.properties","nodePropertyFilePath":"/api-node-config/
↪config-node.properties"},"websocket":{"mq":{"host":"api-node-broker","port":7902,
↪"monitorInterval":500,"connectTimeout":10000,"monitorLoggingThrottle":60000},
↪"allowOptionalAddress":true},"throttling":{"burst":100,"rate":30},"logging":{"console
↪":{"formats":["colorize","simple"],"level":"verbose","handleExceptions":true},"file":
↪{"formats":["prettyPrint"],"level":"verbose","handleExceptions":true,"filename":
↪"catapult-rest.log","maxsize":20971520,"maxFiles":100}},"numBlocksTransactionFeeStats
↪":300,"timestamp":"2021-06-02T09:27:19.336Z"}
rest-gateway_1     | info: connecting to mongodb://db:27017/ (database:catapult) {
↪"timestamp":"2021-06-02T09:27:19.407Z"}
rest-gateway_1     | (node:24) DeprecationWarning: current Server Discovery and
Monitoring engine is deprecated, and will be removed in a future version. To use the new
Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the
MongoClient constructor.
rest-gateway_1     | verbose: connected to mongo at mongodb://db:27017/catapult {
↪"timestamp":"2021-06-02T09:27:19.443Z"}
rest-gateway_1     | info: registering routes {"timestamp":"2021-06-02T09:27:19.445Z"}
rest-gateway_1     | info: listening on port 3000 {"timestamp":"2021-06-02T09:27:19.462Z
↪"}
```

3.3.3 Update encrypted communication between nodes

This chapter describes how to update certificates used for inter-node communication.

Note:

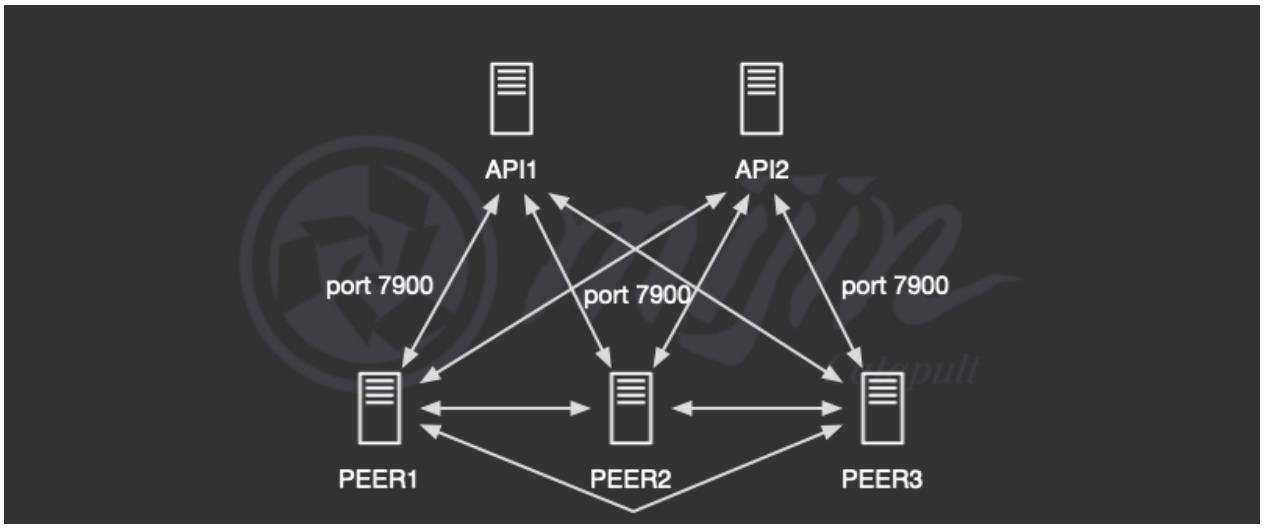
When you deploy mijin Catapult(v.2) via AWS MarketPlace, the initial data is backed up in the AWS Systems Manager Parameter Store.
Note that the following parameter values are subject to update and will differ between the node and the AWS Systems Manager Parameter Store.
The difference does not affect the operation.

- /Crown name specified at deploy time/shares/new-cert/each node/CA/[*].pem
- /Crown name specified at deploy time/shares/nemesis_addresses_harvesting.json
- /Crown name specified at deploy time/shares/nemesis_addresses_harvesting_voting.json
- /Crown name specified at deploy time/shares/nemesis_addresses_harvesting_vrf.json

---

3.3.3.1 mijin Catapult(v.2) encrypted communication between nodes

mijin Catapult(v.2) communicates between nodes using SSL encryption with TLS1.3 at **TCP port/7900**, and only recognized nodes are communicated with as correct nodes.
For SSL communication, a self-signed certificate is applied to each node, and the node pre-registers the KeyPair public key of every node's self-signed certificate.
The KeyPair created from this self-signed certificate is also used as the authorized account that can generate the blockchain.



3.3.3.2 How to renew a node's SSL certificate

The procedure for updating SSL certificates used between nodes is as follows

1. CA and signing node SSL certificate creation (used for communication between nodes)
2. Retrieve the private key from the KeyPair of the SSL certificate of 1 and issue a transaction to enable block generation for the node on the mijin Catapult(v.2) blockchain
3. Create a private key and a dat file for finalization tied to the private key of 2, and issue a transaction to tie it to the private key of 2.
4. Replace the SSL certificate and dat file on the relevant node.
5. Replace the public key of the target node in the configurations of all nodes with the key of 2.

Warning:

As of 2022/10, the procedure for renewing SSL certificates is complex and varies from environment to environment, so please contact mijin Support.
In the future, we plan to provide an easy way to update the data with tools.

3.4 mijin Catapult(v.2) data directory structure

This section describes the data structure of mijin.

3.4.1 Directory of data placement

Directory	Description
/home/catapult/mijin-catapult-package	Directories related to mijin config files and other packages
/mnt/mijin/blocks	mijin's block data directory
/mnt/mijin/mongo	mijin's mongo data directory

3.4.2 Structure of mijin package

3.4.2.1 API Node

```
/home/catapult/mijin-catapult-package
├── default # mijin 構築時に使用するディレクトリ (データとしては不要) Directory to be used when
building mijin (not required as data)
|   ├── catapult
|   |   ├── bin-mount
|   |   |   ├── await
|   |   |   ├── wait
|   |   |   └── waitmongo
|   |   ├── mongo
|   |   |   ├── mongoDbDrop.js
|   |   |   ├── mongoDbPrepare.js
|   |   |   ├── mongoDeploy.sh
|   |   |   ├── mongoLockHashDbPrepare.js
|   |   |   ├── mongoLockSecretDbPrepare.js
|   |   |   ├── mongoMetadataDbPrepare.js
|   |   |   ├── mongoMosaicDbPrepare.js
|   |   |   ├── mongoMultisigDbPrepare.js
|   |   |   ├── mongoNamespaceDbPrepare.js
|   |   |   ├── mongoRestrictionAccountDbPrepare.js
|   |   |   ├── mongoRestrictionMosaicDbPrepare.js
|   |   |   └── mongors.sh
|   |   ├── nemgen
|   |   |   └── nemgen.sh
|   |   ├── scripts
|   |   |   ├── prepare.sh
|   |   |   ├── runServerRecover.sh
|   |   |   ├── startApiServer.sh
|   |   |   ├── startBroker.sh
|   |   |   └── startServer.sh
|   |   └── tools
|   |       ├── clean-all.sh
|   |       └── clean-data.sh
|   └── usr
|       └── catapult
|           └── bin-mount
├── package # mijin 構築パッケージ Package for building mijin
|   └── api # api ノード用のパッケージ Package for api node
|       └── catapult
|           ├── docker-compose.yml # docker コンテナを起動する compose ファイル compose file
to start docker container
|           ├── scripts
|           |   ├── prepare.sh
|           |   ├── recover.sh # ブロックリセット時に使用するスクリプト Script to be used when
```

(continues on next page)

(continued from previous page)

```
resetting the block
|            |    ├── runServerRecover.sh
|            |    ├── startApiServer.sh
|            |    ├── startBroker.sh
|            |    ├── startServer.sh
|            |    └── waitmongo.pl
|            └── userconfig
|                ├── block-properties-file.properties # ネメシス（ジェネシス）ブロックを作成する際に使用するプロパティファイル Property file used to create the Nemesis (Genesis) block
|                ├── resources
|                |    ├── cert # ノードの証明書 Node Certificates
|                |    |    ├── ca.cert.pem
|                |    |    ├── ca.pubkey.pem
|                |    |    ├── node.crt.pem
|                |    |    ├── node.full.crt.pem
|                |    |    └── node.key.pem
|                |    ├── config-database.properties # データベースに関する設定 Database Settings
|                |    ├── config-extensions-broker.properties # broker ノードに関する設定 Broker node settings
|                |    ├── config-extensions-recovery.properties # ブロックリカバリに関する設定 Settings for block recovery
|                |    ├── config-extensions-server.properties # ノードで起動する拡張領域関する設定 Settings related to the extension area to be activated in the node
|                |    ├── config-finalization.properties # ファイナライズに関する設定 Finalization Settings
|                |    ├── config-inflation.properties # インフレーションに関する設定 Inflation-related settings
|                |    ├── config-logging-broker.properties # broker のログに関する設定 Settings for broker logging
|                |    ├── config-logging-recovery.properties # リカバリのログに関する設定 Recovery log settings
|                |    ├── config-logging-server.properties # サーバログに関する設定 Server log settings
|                |    ├── config-messaging.properties # MQ に関する設定 MQ-related settings
|                |    ├── config-network.properties # mijin ネットワークに関する設定 mijin network settings
|                |    ├── config-networkheight.properties # ネットワークのブロック高を知るためのノード数の設定 Set the number of nodes to know the block height of the network
|                |    ├── config-node.properties # ノードに関する設定 Node-related settings
|                |    ├── config-pt.properties # Partial トランザクションに関する設定 Partial Transaction Settings
|                |    ├── config-task.properties # タスクに関する設定 Task-related settings
|                |    ├── config-timesync.properties # 時間に関する設定 Time-related settings
|                |    ├── config-user.properties # ストレージに関する設定 Storage Settings
|                |    ├── peers-api.json # API ノードに関する設定 API Node Settings
|                |    ├── peers-p2p.json # PEER ノードに関する設定 PEER Node Settings
|                |    └── votingkey # Voting に使用するファイル Files used for Voting
|                |        └── private_key_tree1.dat
|                ├── rest.json # rest に関する設定 Settings for REST
|                ├── transactions # Nemesis ブロック作成時にアナウンスするトランザクション Transaction to be announced at the time of Nemesis block creation
|                |    ├── voting_tx0.bin
|                |    ├── voting_tx1.bin
|                |    ├── voting_tx2.bin
|                |    ├── voting_tx3.bin
|                |    ├── voting_tx4.bin
```

(continues on next page)

(continued from previous page)

```
|                 |   ├── voting_tx5.bin
|                 |   ├── voting_tx6.bin
|                 |   ├── voting_tx7.bin
|                 |   ├── voting_tx8.bin
|                 |   ├── voting_tx9.bin
|                 |   ├── vrf_tx0.bin
|                 |   ├── vrf_tx1.bin
|                 |   ├── vrf_tx2.bin
|                 |   ├── vrf_tx3.bin
|                 |   ├── vrf_tx4.bin
|                 |   ├── vrf_tx5.bin
|                 |   ├── vrf_tx6.bin
|                 |   ├── vrf_tx7.bin
|                 |   ├── vrf_tx8.bin
|                 |   └── vrf_tx9.bin
|                 └── votingkeys # 全ノードの Voting ファイル Voting files for all nodes
|                     ├── api
|                     |   ├── 0
|                     |   |   └── private_key_tree1.dat
|                     |   ├── 1
|                     |   |   └── private_key_tree1.dat
|                     |   └── 2
|                     |       └── private_key_tree1.dat
|                     └── peer
|                         ├── 0
|                         |   └── private_key_tree1.dat
|                         ├── 1
|                         |   └── private_key_tree1.dat
|                         ├── 2
|                         |   └── private_key_tree1.dat
|                         ├── 3
|                         |   └── private_key_tree1.dat
|                         ├── 4
|                         |   └── private_key_tree1.dat
|                         ├── 5
|                         |   └── private_key_tree1.dat
|                         └── 6
|                             └── private_key_tree1.dat
└── tools
    └── address.py # アドレス作成時に変換するツール Tools to convert when creating addresses
```

3.4.2.2 PEER Node

```
/home/catapult/mijin-catapult-package
├── default # mijin 構築時に使用するディレクトリ（データとしては不要）Directory to be used when
building mijin (not required as data)
|   └── catapult
|       ├── bin-mount
|       |   ├── await
|       |   ├── wait
|       |   └── waitmongo
|       ├── mongo
|       |   ├── mongoDbDrop.js
|       |   ├── mongoDbPrepare.js
|       |   ├── mongoDeploy.sh
```

(continues on next page)

(continued from previous page)

```
|       |   ├── mongoLockHashDbPrepare.js
|       |   ├── mongoLockSecretDbPrepare.js
|       |   ├── mongoMetadataDbPrepare.js
|       |   ├── mongoMosaicDbPrepare.js
|       |   ├── mongoMultisigDbPrepare.js
|       |   ├── mongoNamespaceDbPrepare.js
|       |   ├── mongoRestrictionAccountDbPrepare.js
|       |   ├── mongoRestrictionMosaicDbPrepare.js
|       |   └── mongors.sh
|       ├── nemgen
|       |   └── nemgen.sh
|       ├── scripts
|       |   ├── prepare.sh
|       |   ├── runServerRecover.sh
|       |   ├── startApiServer.sh
|       |   ├── startBroker.sh
|       |   └── startServer.sh
|       └── tools
|           ├── clean-all.sh
|           └── clean-data.sh
├── package
|   └── peer
|       └── catapult
|           ├── docker-compose.yml # docker コンテナを起動する compose ファイル compose file
to start docker container
|           ├── scripts
|           |   ├── prepare.sh
|           |   ├── recover.sh # ブロックリセット時に使用するスクリプト Script to be used when
resetting the block
|           |   ├── runServerRecover.sh
|           |   ├── startApiServer.sh
|           |   ├── startBroker.sh
|           |   ├── startServer.sh
|           |   └── waitmongo.pl
|           └── userconfig
|               ├── block-properties-file.properties # ネメシス（ジェネシス）ブロックを作成す
る際に使用するプロパティファイル Property file used to create the Nemesis (Genesis) block
|               ├── resources
|               |   ├── cert # ノードの証明書 Node Certificates
|               |   |   ├── ca.cert.pem
|               |   |   ├── ca.pubkey.pem
|               |   |   ├── node.crt.pem
|               |   |   ├── node.full.crt.pem
|               |   |   └── node.key.pem
|               |   ├── config-database.properties # データベースに関する設定 Database
Settings
|               |   ├── config-extensions-recovery.properties # ブロックリカバリに関する設定
Settings for block recovery
|               |   ├── config-extensions-server.properties # ノードで起動する拡張領域関する
設定 Settings related to the extension area to be activated in the node
|               |   ├── config-finalization.properties # ファイナライズに関する設定
Finalization Settings
|               |   ├── config-harvesting.properties # ハーベストに関する設定 Harvest-
↪related settings
|               |   ├── config-inflation.properties # インフレーションに関する設定 Inflation-
↪related settings
|               |   ├── config-logging-recovery.properties # リカバリのログに関する設定
```

(continues on next page)

(continued from previous page)

```
Recovery log settings
|               |   ├── config-logging-server.properties # サーバログに関する設定 Server log
settings
|               |   ├── config-messaging.properties # MQ に関する設定 MQ-related settings
|               |   ├── config-network.properties # mijin ネットワークに関する設定 mijin
network settings
|               |   ├── config-networkheight.properties # ネットワークのブロック高を知るための
ノード数の設定 Set the number of nodes to know the block height of the network
|               |   ├── config-node.properties # ノードに関する設定 Node-related settings
|               |   ├── config-pt.properties # Partial トランザクションに関する設定 Partial
Transaction Settings
|               |   ├── config-task.properties # タスクに関する設定 Task-related settings
|               |   ├── config-timesync.properties # 時間に関する設定 Time-related settings
|               |   ├── config-user.properties # ストレージに関する設定 Storage Settings
|               |   ├── peers-api.json # API ノードに関する設定 API Node Settings
|               |   ├── peers-p2p.json # PEER ノードに関する設定 PEER Node Settings
|               |   └── votingkey # Voting に使用するファイル Files used for Voting
|               |       └── private_key_tree1.dat
|               ├── transactions # Nemesis ブロック作成時にアナウンスするトランザクション
Transaction to be announced at the time of Nemesis block creation
|               |   ├── voting_tx0.bin
|               |   ├── voting_tx1.bin
|               |   ├── voting_tx2.bin
|               |   ├── voting_tx3.bin
|               |   ├── voting_tx4.bin
|               |   ├── voting_tx5.bin
|               |   ├── voting_tx6.bin
|               |   ├── voting_tx7.bin
|               |   ├── voting_tx8.bin
|               |   ├── voting_tx9.bin
|               |   ├── vrf_tx0.bin
|               |   ├── vrf_tx1.bin
|               |   ├── vrf_tx2.bin
|               |   ├── vrf_tx3.bin
|               |   ├── vrf_tx4.bin
|               |   ├── vrf_tx5.bin
|               |   ├── vrf_tx6.bin
|               |   ├── vrf_tx7.bin
|               |   ├── vrf_tx8.bin
|               |   └── vrf_tx9.bin
|               └── votingkeys # 全ノードの Voting ファイル Voting files for all nodes
|                   ├── api
|                   |   ├── 0
|                   |   |   └── private_key_tree1.dat
|                   |   ├── 1
|                   |   |   └── private_key_tree1.dat
|                   |   └── 2
|                   |       └── private_key_tree1.dat
|                   └── peer
|                       ├── 0
|                       |   └── private_key_tree1.dat
|                       ├── 1
|                       |   └── private_key_tree1.dat
|                       ├── 2
|                       |   └── private_key_tree1.dat
|                       ├── 3
|                       |   └── private_key_tree1.dat
```

(continues on next page)

(continued from previous page)

```
│                         ├── 4
│                         │   └── private_key_tree1.dat
│                         ├── 5
│                         │   └── private_key_tree1.dat
│                         └── 6
│                             └── private_key_tree1.dat
└── tools
    └── address.py # アドレス作成時に変換するツール Tools to convert when creating addresses
```

3.4.3 Block Data Structure

```
/mnt/mijin/blocks/
├── data # ブロックチェーンのデータディレクトリ Blockchain Data Directory
│   ├── 00000
│   ├── audit
│   ├── catapult_server0000.log
│   ├── commit_step.dat
│   ├── importance
│   ├── index.dat
│   ├── logs
│   ├── proof.index.dat
│   ├── server.lock
│   ├── spool
│   ├── startup
│   ├── state
│   ├── statedb
│   ├── summary.txt
│   ├── transfer_message
│   ├── voting
│   └── voting_status.dat
└── seed # Nemesis ブロック Nemesis block
    ├── 00000
    ├── index.dat
    ├── proof.index.dat
    └── summary.txt
```

3.4.4 Mongo Data Structure

```
/mnt/mijin/mongo/
└── db # mongo のデータディレクトリ mongo data directory
    ├── WiredTiger
    ├── WiredTiger.lock
    ├── WiredTiger.turtle
    ├── WiredTiger.wt
    ├── WiredTigerLAS.wt
    ├── _mdb_catalog.wt
    ├── collection-0--1310205274663118138.wt
    ├── collection-0--3714664905013916938.wt
    ├── collection-108--1310205274663118138.wt
    ├── collection-115--1310205274663118138.wt
    ├── collection-122--1310205274663118138.wt
    ├── collection-129--1310205274663118138.wt
```

(continues on next page)

(continued from previous page)

```
    ├── collection-138--1310205274663118138.wt
    ├── collection-145--1310205274663118138.wt
    ├── collection-150--1310205274663118138.wt
    ├── collection-162--1310205274663118138.wt
    ├── collection-167--1310205274663118138.wt
    ├── collection-2--1310205274663118138.wt
    ├── collection-28--1310205274663118138.wt
    ├── collection-35--1310205274663118138.wt
    ├── collection-4--1310205274663118138.wt
    ├── collection-56--1310205274663118138.wt
    ├── collection-61--1310205274663118138.wt
    ├── collection-66--1310205274663118138.wt
    ├── collection-71--1310205274663118138.wt
    ├── collection-78--1310205274663118138.wt
    ├── collection-8--1310205274663118138.wt
    ├── collection-9--1310205274663118138.wt
    ├── collection-93--1310205274663118138.wt
    ├── diagnostic.data
    ├── index-1--1310205274663118138.wt
    ├── index-1--3714664905013916938.wt
    ├── index-10--1310205274663118138.wt
    ├── index-102--1310205274663118138.wt
    ├── index-104--1310205274663118138.wt
    ├── index-106--1310205274663118138.wt
    ├── index-109--1310205274663118138.wt
    ├── index-11--1310205274663118138.wt
    ├── index-110--1310205274663118138.wt
    ├── index-113--1310205274663118138.wt
    ├── index-116--1310205274663118138.wt
    ├── index-117--1310205274663118138.wt
    ├── index-120--1310205274663118138.wt
    ├── index-123--1310205274663118138.wt
    ├── index-124--1310205274663118138.wt
    ├── index-127--1310205274663118138.wt
    ├── index-13--1310205274663118138.wt
    ├── index-130--1310205274663118138.wt
    ├── index-131--1310205274663118138.wt
    ├── index-134--1310205274663118138.wt
    ├── index-136--1310205274663118138.wt
    ├── index-139--1310205274663118138.wt
    ├── index-140--1310205274663118138.wt
    ├── index-143--1310205274663118138.wt
    ├── index-146--1310205274663118138.wt
    ├── index-147--1310205274663118138.wt
    ├── index-151--1310205274663118138.wt
    ├── index-152--1310205274663118138.wt
    ├── index-154--1310205274663118138.wt
    ├── index-156--1310205274663118138.wt
    ├── index-158--1310205274663118138.wt
    ├── index-16--1310205274663118138.wt
    ├── index-160--1310205274663118138.wt
    ├── index-163--1310205274663118138.wt
    ├── index-164--1310205274663118138.wt
    ├── index-168--1310205274663118138.wt
    ├── index-169--1310205274663118138.wt
    ├── index-19--1310205274663118138.wt
    ├── index-22--1310205274663118138.wt
```

(continues on next page)

(continued from previous page)

```
    ├── index-25--1310205274663118138.wt
    ├── index-29--1310205274663118138.wt
    ├── index-3--1310205274663118138.wt
    ├── index-30--1310205274663118138.wt
    ├── index-32--1310205274663118138.wt
    ├── index-36--1310205274663118138.wt
    ├── index-37--1310205274663118138.wt
    ├── index-39--1310205274663118138.wt
    ├── index-41--1310205274663118138.wt
    ├── index-44--1310205274663118138.wt
    ├── index-46--1310205274663118138.wt
    ├── index-48--1310205274663118138.wt
    ├── index-5--1310205274663118138.wt
    ├── index-50--1310205274663118138.wt
    ├── index-52--1310205274663118138.wt
    ├── index-54--1310205274663118138.wt
    ├── index-57--1310205274663118138.wt
    ├── index-58--1310205274663118138.wt
    ├── index-6--1310205274663118138.wt
    ├── index-62--1310205274663118138.wt
    ├── index-63--1310205274663118138.wt
    ├── index-67--1310205274663118138.wt
    ├── index-68--1310205274663118138.wt
    ├── index-72--1310205274663118138.wt
    ├── index-73--1310205274663118138.wt
    ├── index-75--1310205274663118138.wt
    ├── index-79--1310205274663118138.wt
    ├── index-80--1310205274663118138.wt
    ├── index-82--1310205274663118138.wt
    ├── index-84--1310205274663118138.wt
    ├── index-87--1310205274663118138.wt
    ├── index-89--1310205274663118138.wt
    ├── index-91--1310205274663118138.wt
    ├── index-94--1310205274663118138.wt
    ├── index-95--1310205274663118138.wt
    ├── index-97--1310205274663118138.wt
    ├── index-99--1310205274663118138.wt
    ├── journal
    ├── mongod.lock
    ├── sizeStorer.wt
    └── storage.bson
```

3.5 mijin Catapult(v.2) environment building options table

We typically use Ansible to build mijin environments, and the ansible option in the Playbook is shown here.

No	item name	Default value	Input value	説明
1	service	peer	peer,api	Specify the mode in which ansible is to be built. [peer] Build in peer mode [api] Build in api mode

continues on next page

Table 1 – continued from previous page

No	item name	Default value	Input value	説明
2	share_mode	dir	dir,ssm,s3	Places the initial address used when creating the Nemesis block. [dir] Stored only in share_directory. [ssm] Store in AWS SSM parameter store. [s3] Save to AWS S3.
3	aws_region	ap-northeast-1	String	Specify the AWS region. Specifying the region will be used to retrieve to S3 and SSM. <https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-available-regions>
4	ssm_ps_name			share_mode: For ssm, specify the path to be stored in the SSM parameter store.
5	s3_bucket_name			share_mode: For s3, specify the S3 bucket name.
6	api_dual_mode	FALSE	true,false	service: api の場合、dual モードで起動するかを指定します。 [Yes] Start in dual mode. (Enable Harvest) [No] Starts in normal mode.
7	api_hosts	[192.168.96.131]	Array[String, String]	Specify the host name or IP address of the API node as an array.
8	peer_hosts	[192.168.96.132, 192.168.96.133]	Array[String, String]	Specify the hostname or IP address of the PEER node as an array.
9	unix_user	catapult	String	Specify the user who will launch mijin.
10	repo_name	mijin-catapult-package	String	Specify the directory where the mijin application is located. /home/[unix_user]/[repo_name]
11	catapult_version	v10038	v10038,v10037	Specifies the version of catapult. v10038
12	network_identifier	mijin	mijin,mijin-test	Specify the network type to be built with mijin.
13	mongo_docker_version	4.2.5	String	Specifies the version of the mongo container image used by the API node. <https://hub.docker.com/_/mongo?tab=tags>
14	mongo_host	db	String	Specify the mongo container name for the API node.
15	mongo_max_connection	7	Int	Specify the API node's connection limit to mongo.
16	mongo_base_retry_delay	750	Int	Specify the connection retry time to mongo for the API node.
17	python_docker_version	3.9.0	String	Specify the docker version of python.
18	enable_pip_install	FALSE	true,false	Specify whether to run pip install.
19	share_directory	/opt/mijin/shares	String	Nemesis ブロック作成時にアドレスデータを保管するディレクトリを指定します。 s3/ssm を指定しても、一時保存場所として使用されます。
20	block_directory	/opt/mijin/blocks	String	Specify the directory where block data is stored for each node.
21	mongo_directory	/opt/mijin/mongo	String	Specify the directory where mongo data is stored for the API node.
22	docker_network_range	172.20.0.0/24	String	Specifies the network range used by docker.

continues on next page

Table 1 – continued from previous page

No	item name	Default value	Input value	説明
23	rest_ip_address	172.20.0.9	String	docker ネットワーク上での rest-gateway コンテナの IP アドレスです。
24	domain	mijin.internal	String	各ノード間の通信で使用する SSL 証明書の CN を指定します。
25	node_port	7900	Int	Specifies the TCP port number to be used for communication between each node.
26	enable_cache_database_storage	TRUE	true,false	キャッシュデータ保存を有効化します。 Disabling it may improve processing performance.
27	enable_auto_sync_cleanup	TRUE	true,false	Specifies automatic deletion of temporary synchronization files.
28	base_namespace	cat	String	Specifies the root namespace to be used for the base currency.
29	base_currency_name	currency	String	Specifies the subnamespace to be used for the base currency.
30	base_harvest_name	harvest	String	Harvest モザイクのネームスペースを指定します。
31	currency_supply	8’998’999’998’000’000	String	Specifies the amount of base currency issued. With quotes.
32	harvest_supply	15’000’000	String	Specifies the amount of Harvest mosaic issued.
33	block_generation_target_time	15s	String	ブロック生成間隔を指定します（例: 15s）
34	rest_gateway_private_key_num	1	Int	rest_gateway のアドレス作成数を指定します。
35	nemesis_generation_hash_num	1	Int	Nemesis 用の GenerationHash アドレス作成数。
36	nemesis_addresses_harvesting_num	4	Int	Number of addresses created for Harvest (same number as number of nodes required)
37	nemesis_signer_private_key_num	1	Int	Number of addresses created for cosign (usually 1)
38	nemesis_addresses_num	10	Int	Number of empty (unused) addresses created.
39	transaction_selections_tratego	oldest	String	Select from oldest, maximize-fee, minimize-fee, etc.
40	max_time_behind_pull_transaction_start	5m	String	Maximum allowable delay for Pull transactions.
41	min_fee_multiplier	100	Int	Minimum commission multiplier. 0 for free.
42	default_dynamic_fee_multiplier	1’000	String	Base coefficient for dynamic fees.
43	root_namespace_rental_fee_per_block	1	Int	Rental fee per block of root namespace.
44	child_namespace_rental_fee	100	Int	子ネームスペースのレンタル料。
45	mosaic_rental_fee	500	Int	モザイクのレンタル料。
46	rest_throttling_burst	100	Int	Maximum number of REST connections during burst.
47	rest_throttling_rate	30	Int	Normal REST connection processing rate.
48	unconfirmd_cache_max_response_size	20MB	String	未承認トランザクション受信の最大サイズ。
49	unconfirmd_cache_max_size	5MB	String	未承認トランザクションのキャッシュ最大サイズ。
50	cache_max_response_size	20MB	String	（旧バージョン向け）未承認 Tx の応答最大サイズ。
51	cache_max_size	50’000	String	（旧バージョン向け）キャッシュ Tx の最大数。
52	block_disruptor_slot_count	4096	Int	ブロック処理用スロット数。
53	block_element_trace_interval	1	Int	ブロック要素のトレース間隔。
54	block_disruptor_max_memory_size	300MB	String	ブロックディスラプターの最大メモリ。
55	transaction_disruptor_slot_count	8192	Int	トランザクション処理用スロット数。
56	transaction_element_trace_interval	10	Int	トランザクション要素のトレース間隔。
57	max_transaction_per_block	6’000	String	Maximum number of transactions in one block.
58	min_transaction_failures_count_for_ban	8	Int	トランザクション失敗回数の BAN 閾値。
59	min_transaction_failures_percent_for_ban	10	Int	トランザクション失敗率の BAN 閾値。
60	partial_cache_max_response_size	5MB	String	Maximum partial transaction response size.
61	partial_cache_max_size	20MB	String	部分トランザクションキャッシュ全体の最大サイズ。
62	enable_finalization	TRUE	true,false	ファイナライゼーションプラグインを有効化するか。
63	max_rollback_blocks	0	Int	Maximum number of blocks that can be rolled back (definitive at 0).
64	enable_voting	TRUE	true,false	Voting 機能を有効にするか。
65	voting_set_grouping	160	Int	Voting ラウンドのブロック数（importance_grouping の倍数）。

continues on next page

Table 1 – continued from previous page

No	item name	Default value	Input value	説明
66	votingkey_start_epoch	1	Int	VotingKey の最小有効期間（エポック単位）。
67	votingkey_end_epoch	26280	Int	Maximum validity period of the VotingKey (e.g., approximately 821 days).
68	voting_key_dilution	128	Int	Voting key dilution level (reuse interval).
69	enable_revote_on_boot	FALSE	true,false	Do they automatically re-poll on startup
70	importance_grouping	40	Int	Number of Importance rounds (impact score update interval).
71	max_transaction_lifetime	24h	String	Transaction validity period (e.g., 24h).
72	max_block_future_time	500ms	String	Maximum allowable time for future block reception.
73	max_transactions_per_aggregate	1'000	String	Maximum number of Tx to be included in the aggregate Tx (100 in Symbol).
74	max_cosignatures_per_aggregate	25	Int	Maximum number of signatures that can cosign an aggregate Tx.
75	max_bonded_transaction_lifetime	48h	String	Aggregate bonded Tx validity period.
76	locked_funds_per_aggregate	10'000'000	String	Aggregate Tx lock deposit.
77	max_hash_lock_duration	2d	String	The validity period of the hash lock.
78	max_secret_lock_duration	30d	String	The maximum validity period of the Secret Lock.
79	min_proof_size	1	Int	シークレットプルーフの最小バイト数。
80	max_proof_size	1000	Int	シークレットプルーフの最大バイト数。
81	max_meta_value_size	1024	Int	Maximum size (in bytes) of the metadata Value.
82	max_cosignatories_per_account	25	Int	アカウントの連署者数の上限。
83	max_cosigned_accounts_per_account	25	Int	1 アカウントが連署できるアカウント数の上限。
84	max_multisig_depth	3	Int	Upper limit on the depth of the multisig hierarchy.
85	max_mosaics_per_account	1'000	String	The number of mosaics an account can hold.
86	max_mosaic_duration	3650d	String	Maximum validity period of the mosaic (in days)
87	max_mosaic_divisibility	6	Int	Maximum number of decimal places in the mosaic.
88	max_name_size	64	Int	Maximum length (characters) of the namespace name.
89	max_child_namespaces	256	Int	Number of child namespaces that the parent namespace can have.
90	max_namespace_depth	3	Int	Maximum depth of the namespace hierarchy.
91	min_namespace_duration	1m	String	Minimum validity period of the namespace.
92	max_namespace_duration	3650d	String	Maximum validity period of the namespace.
93	namespace_grace_period_duration	30d	String	Grace period after the namespace expiration date.
94	max_account_restriction_values	512	Int	Maximum number of values that can be set for account limits.
95	max_mosaic_restriction_values	20	Int	Maximum number of values that can be set for the mosaic limit.
96	max_message_size	1024	Int	トランザクションメッセージの最大バイト数。