Zeeve

Release 1.6.0

Sankalp Sharma

Oct 06, 2023

CONTENTS

CHAPTER

ONE

WHAT IS ZEEVE?

Blockchain adoption is slow due to various complexities that are involved in ideating, deploying, maintaining and extending a solution built over it. IT sector is not well equipped to take on these challenges in the same manner that it has been handling deployments so far, putting big workforce onto their projects. **Zeeve** is a Blockchain management platform which acts as one's own Devops and Cloud team. It targets to provide ease to developers & enterprises by targetting 100% automation for their Blockchain solutions, indirectly solving challenges in adoption of Blockchain.

It is backed by a group of Blockchain and IOT experts, which have expertise over various protocols and understand the needs of the industry, the protocol and developers. Initially created to automate some of our personal projects, we saw the need for it externally.

Some basic needs for blockchain solutions are:-

- **Multi-cloud network deployments.** A product should be able to function over any cloud that your client requires it run on without any extra configurations.
- **Heterogeneous Cloud networks.** A Blockchain network might have some nodes running on one Cloud, while some nodes on another. This is important for the next big point which is,
- **Ability to create consortiums easily.** Building consortium is the toughest challenge for the industry. Zeeve targets technical side of this challenge and intends for users to extend their consortium to others through simple UI based interactions.
- **Making your deployments production grade.** Blockchain industry is full of complex protocols. A single left out variable of which may cause some pretty huge consequences. Zeeve helps standardize your deployments, making them ready to use in production.
- **Monitor/Scale/Optimize your deployments.** Scaling your network in count of nodes and monitor and look out for any red flags is vital to any distributed network, but its also something hard to come by as every protocol is unique in their own manner.
- **Reach your customers easily through marketplace.** Zeeve allows users to list their product in marketplace which could , if fully configured can be bought and started in a matter of minutes.

This document will help user to understand different operations onto zeeve and understand its functioning. Welcome!

CHAPTER

TWO

DESCRIPTION: GET THE LATEST DOCUMENTATION AND TECHNICAL DETAILS FOR ZEEVE'S INNOVATIVE PLATFORM. EXPLORE OUR API, INTEGRATIONS, AND TOOLS FOR BUILDING SCALABLE, RELIABLE APPLICATIONS ON THE ZEEVE PLATFORM.

CHAPTER

THREE

OVERVIEW

Zeeve is a Blockchain as a Service (BaaS) offering that allows customers to leverage cloud-based solutions to build, host and use their own Blockchain apps, Smart Contracts and functions on the blockchain while the cloud-based service provider manages all the necessary tasks and activities to keep the infrastructure agile and operational. It is an interesting development in the blockchain ecosystem that is indirectly aiding the blockchain adoption across businesses. It is based on, and works similar to, the concept of Software As A Service (SaaS) model.

3.1 Dashboard

This is the dashboard page of Zeeve where you can see your Networks, Nodes, deployed product details, Node Disk Usage, Node Statistics, and all activity logs

3.2 Networks

Networks page basically give the details of Network Name, Number of active nodes, Number of inactive nodes, Selected Blockchain Network, Running Products Name, Network Created Date, Health of Network, Action(Statistics, Restart, Stop and Delete) .

3.3 Nodes

Nodes Page contains the information like Node ID, Network Name, Health, Network's Region, Network Created Date, Online(Green colour for online and Red colour for Offline) and Action(Statistics, Restart, Stop and Delete).

3.3.1 Stats(Statistics)

This action page contains information of Node Performance Details like Network, Region, Created Date, Node ID, IP Address, Node Status, CPU Usage, RAM Usage, Storage, Recent RAM Usage, Disk Usage, *Inodes*, Network Usage and Load. Basically we can monitor our network in this action.

3.3.2 Restart

By using this action we can re-start the Node.

3.3.3 Stop

By using this action we can stop the Node.

3.3.4 Delete

By using this action we can delete the Node.

3.4 Products

The products page contains the information about selected blockchain's product. These products are fully based on Blockchain. There are two types of products we can see here one is free product and another one is paid product.

3.4.1 Free Products

Free product is nothing but it's open source project i.e. is supported by blockchain network e.g. Sawtooth Supply Chain, Sawtooth Tuna Fish. We have no need to pay for it.

3.4.2 Paid Producs

There are various paid products offering by blockchain network based on our choice we can choose it and add to our network. Sawtooth offering paid produts like MBC (Menu Based Choice) Conjoint, Lighthouse Studio etc.

3.5 API Services

3.6 Device Data

3.7 Tasks(Activity Log)

We can see all activity logs and related information of tasks here like network creation, product addition, node addition, etc.

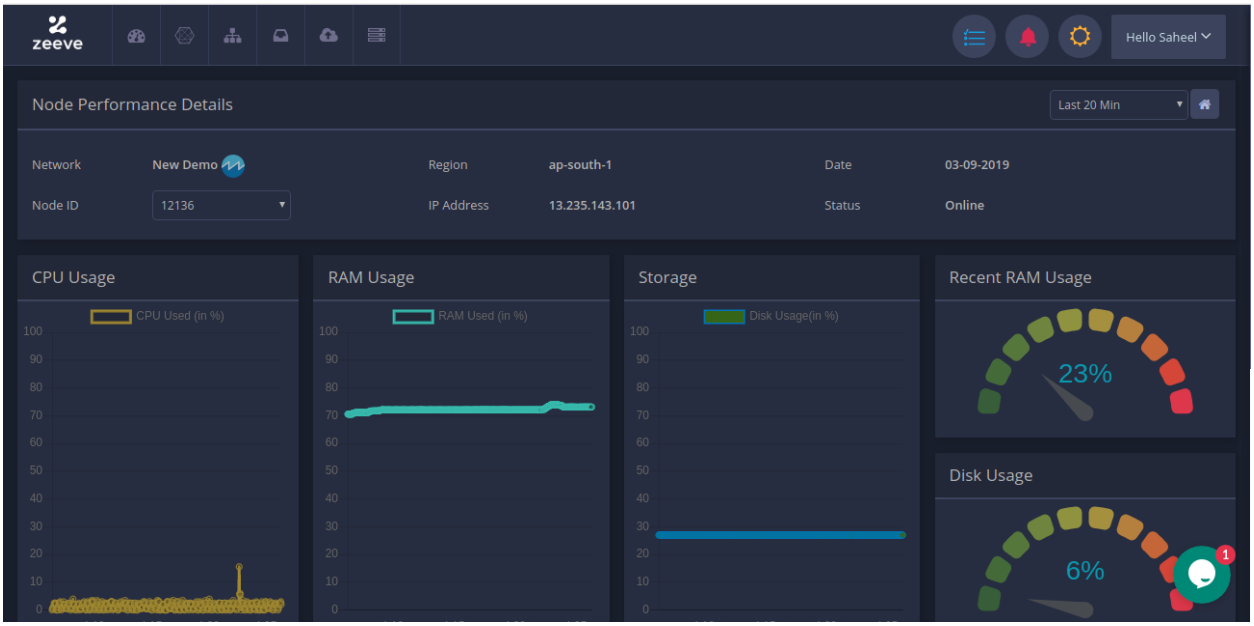
3.8 Notification

In this section we can see all notifications and invitations(refer to or receive from other people to add network).

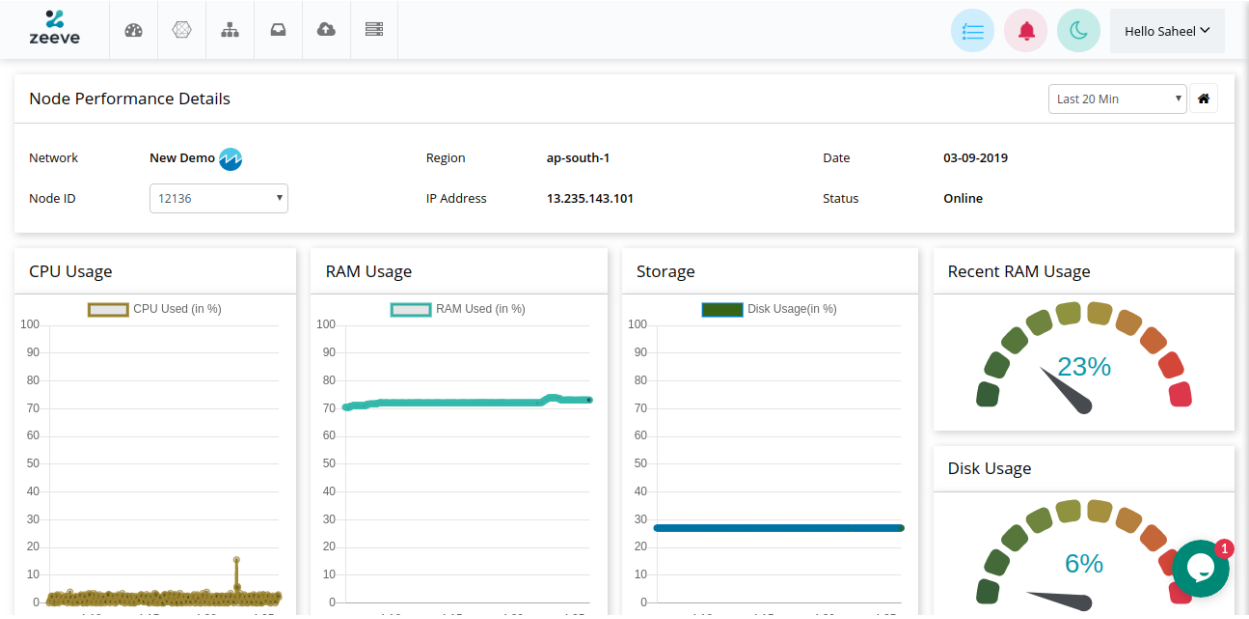
3.9 Night Mode

We can change page orientation by using this theme we can select night mode or day light mode, working as vice-versa.

3.9.1 Night Mode



3.9.2 Day Light Mode



3.10 Zeeve Assistant

For more information you can communicate with our Zeeve Assistant.

CHAPTER

FOUR

DESCRIPTION: LEARN HOW TO CREATE A NEW ACCOUNT ON ZEEVE'S PLATFORM. OUR STEP-BY-STEP GUIDE WALKS YOU THROUGH THE PROCESS OF SETTING UP A SECURE AND RELIABLE ACCOUNT FOR ACCESSING OUR API AND TOOLS.

CHAPTER

FIVE

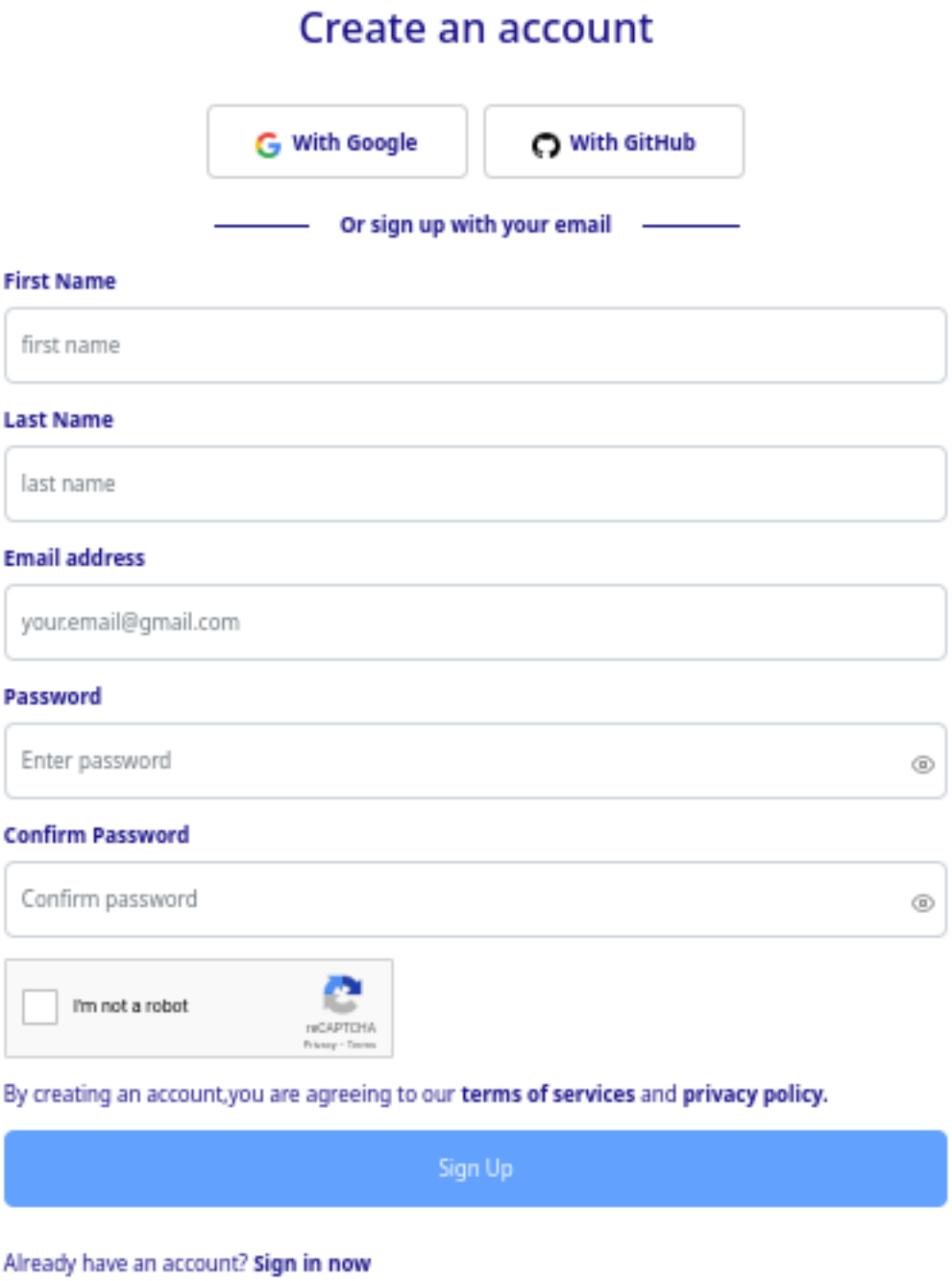
ACCOUNT CREATION

This section speaks about how to create an account on Zeeve after which user can authorize external accounts/platforms on Zeeve.

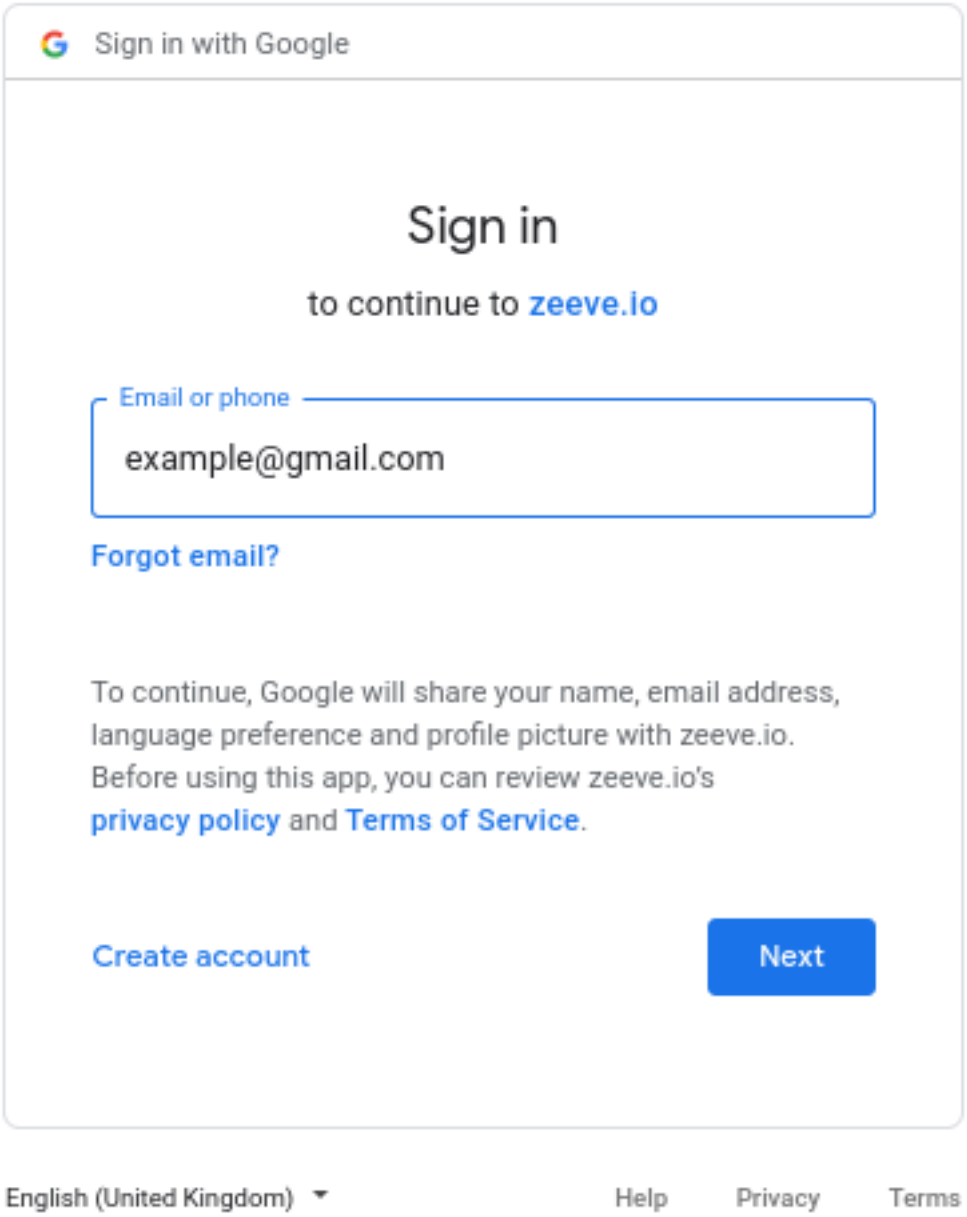
Zeeve supports multiple options for creating an account. You can choose to create account with you Google account or Github account. Also you can use your email to create an account and use the same for signing in.

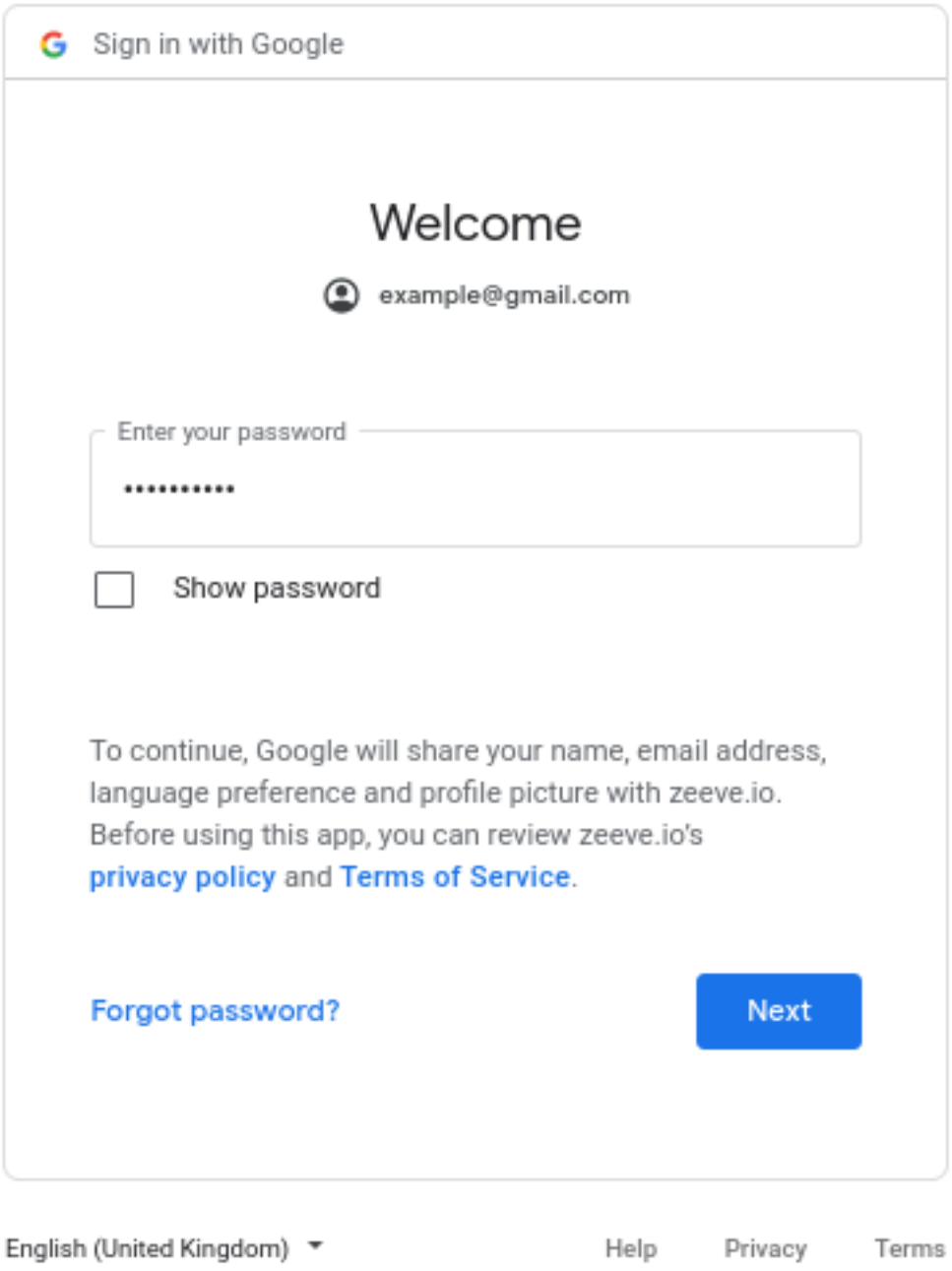
5.1 Signing up with google account

1. Click on **with Google**

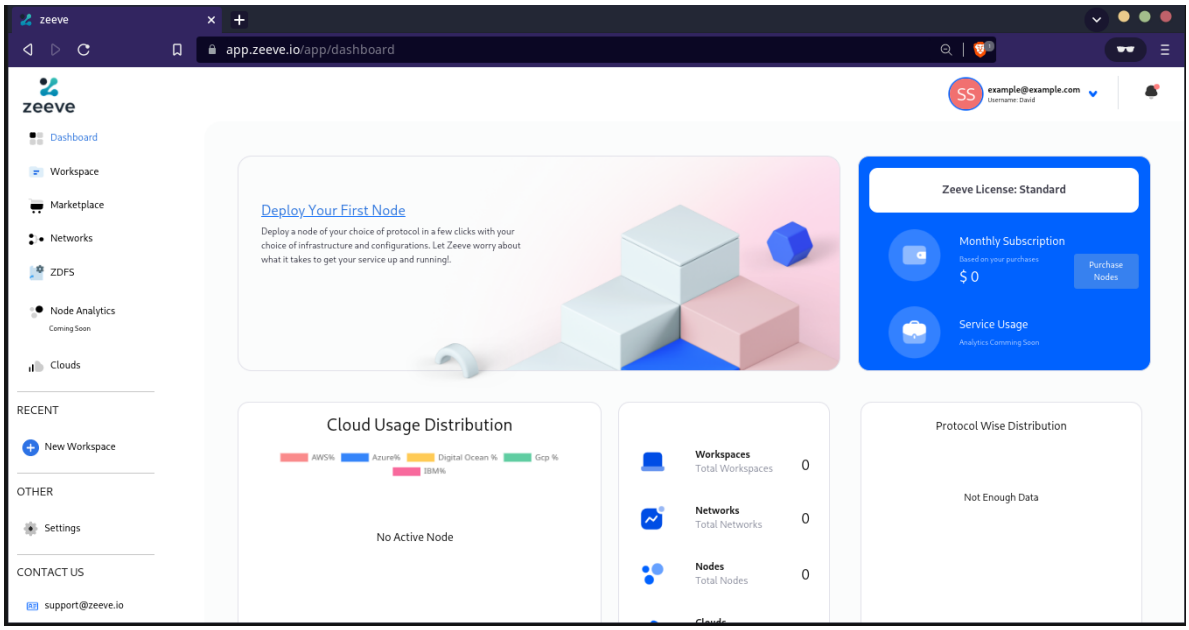


2. You will be redirected to google account page. Fill the email address, password and click **Next**


Sign in with Google
Sign in
to continue to zeeve.io
Email or phone
example@gmail.com
Forgot email?
To continue, Google will share your name, email address, language preference and profile picture with zeeve.io. Before using this app, you can review zeeve.io's privacy policy and Terms of Service.
Create account
Next
English (United Kingdom)
Help
Privacy
Terms

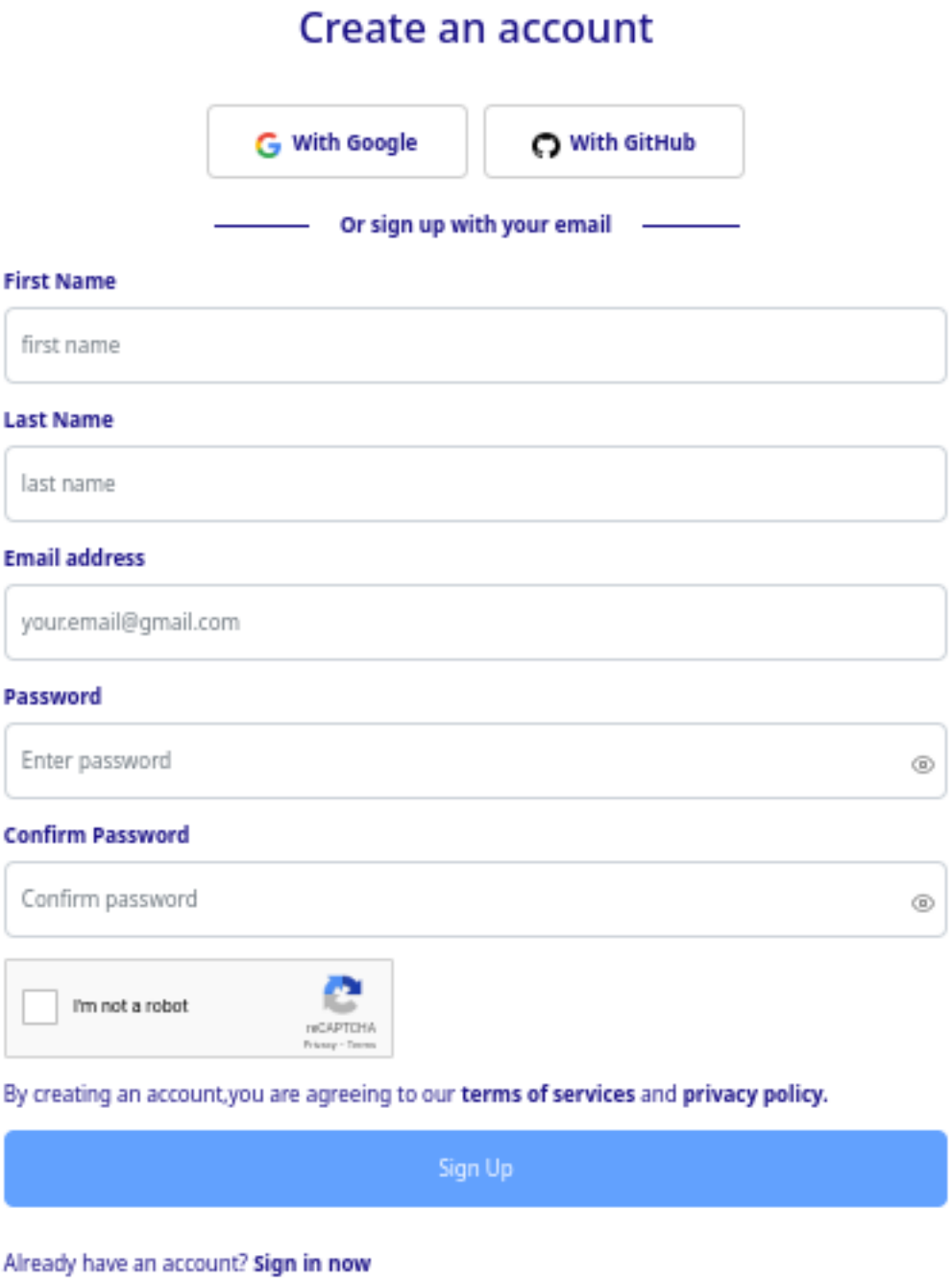


3. After successful sign in with Google, you will be landed to Zeeve's dashboard.

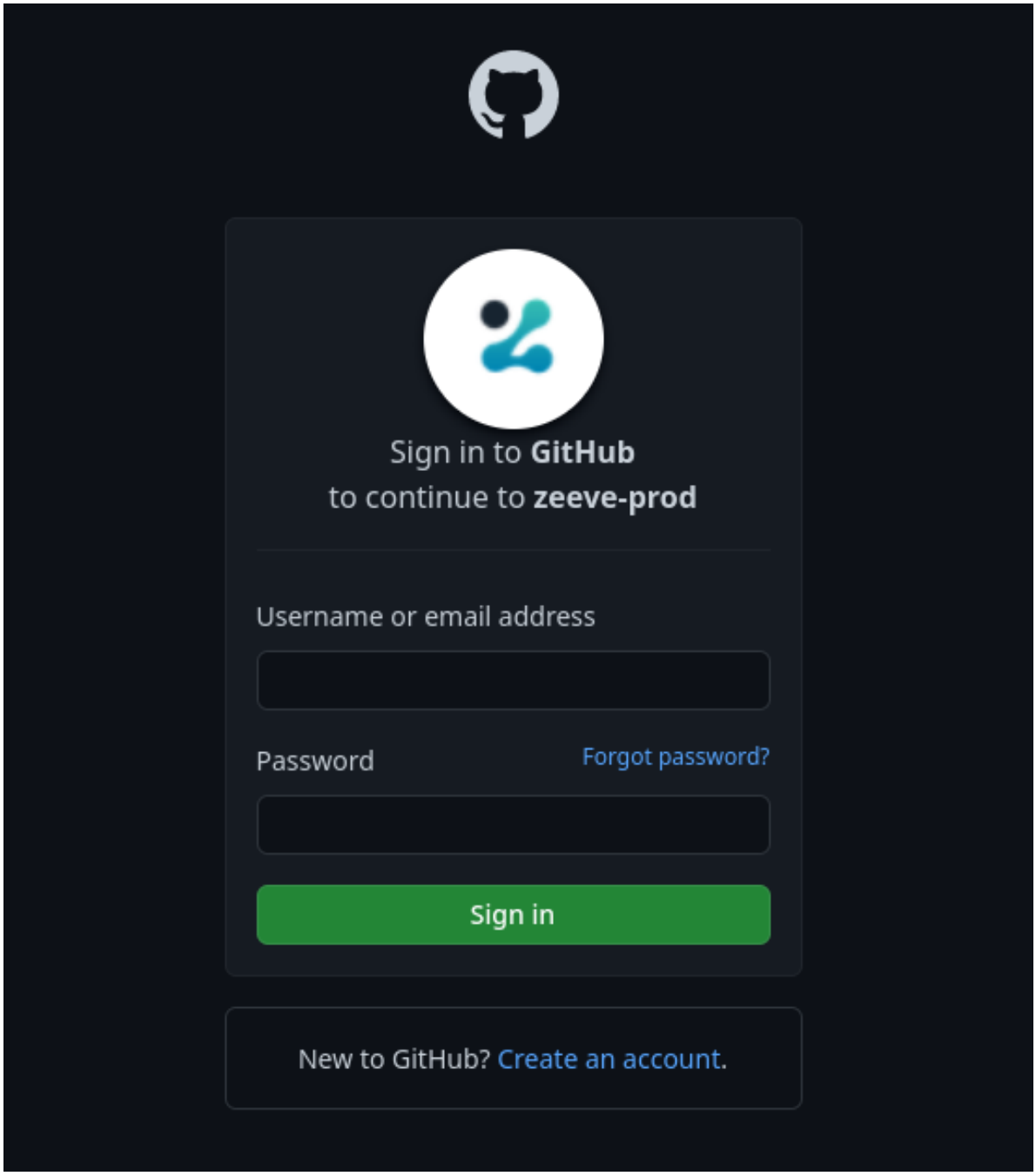


5.2 Signing up with github account

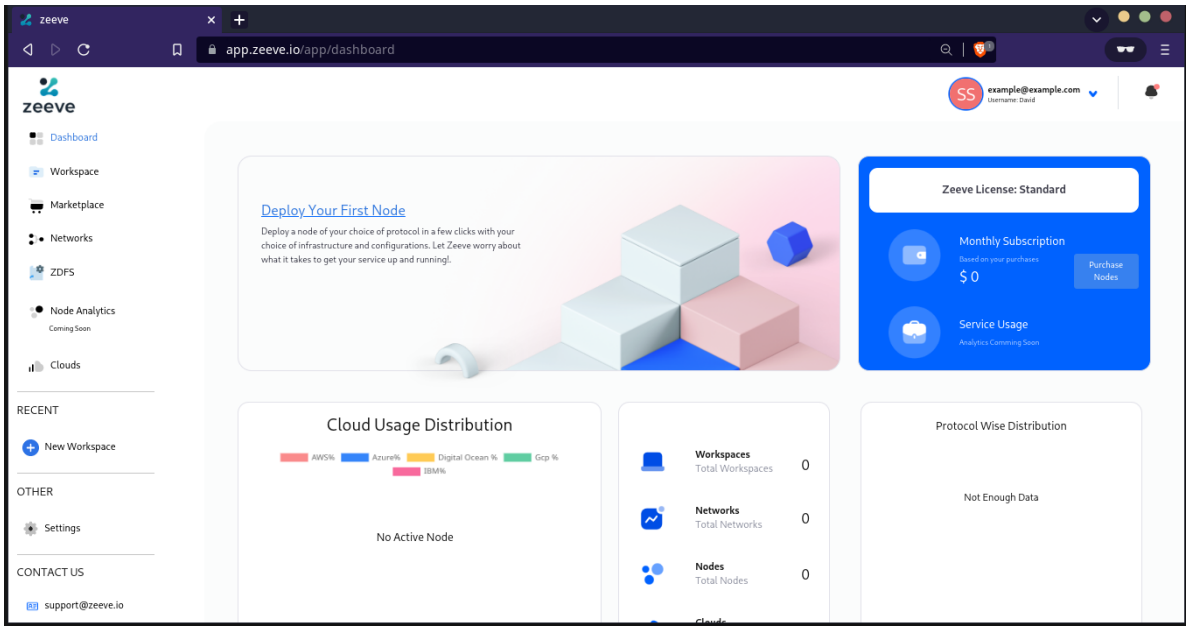
1. Click on **with Google**



2. You will be redirected to github account page. Fill the username/email address, password and click **Sign in**

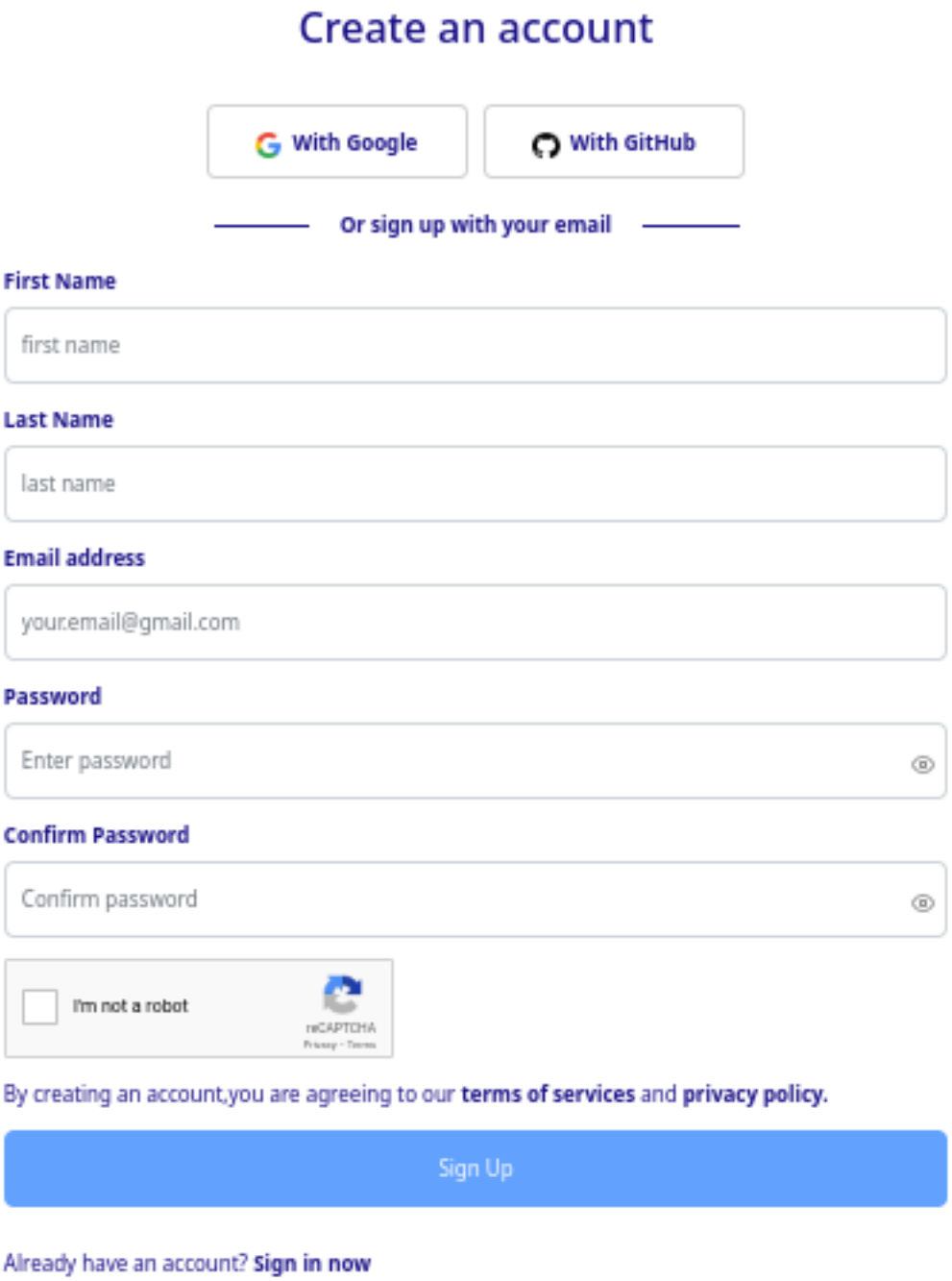


3. After successful sign in with GitHub, you will be landed to Zeeve's dashboard.

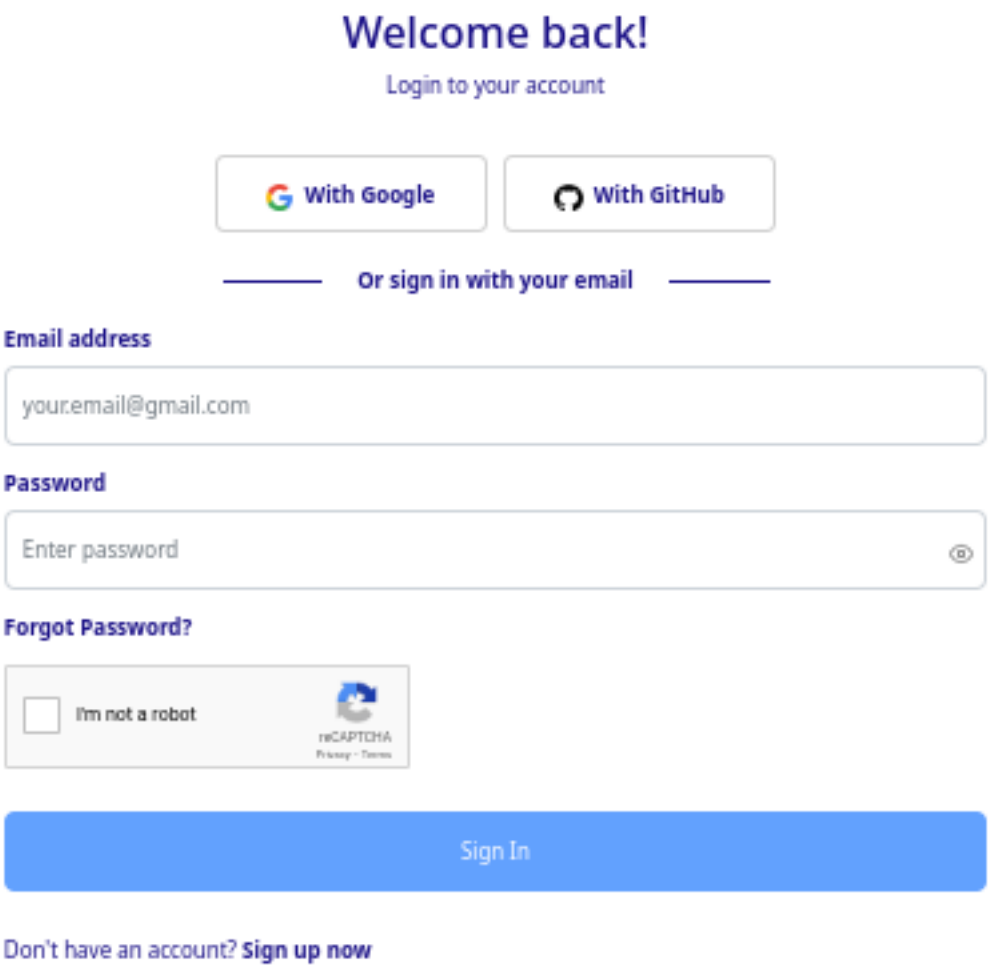


5.3 Signing up with email account

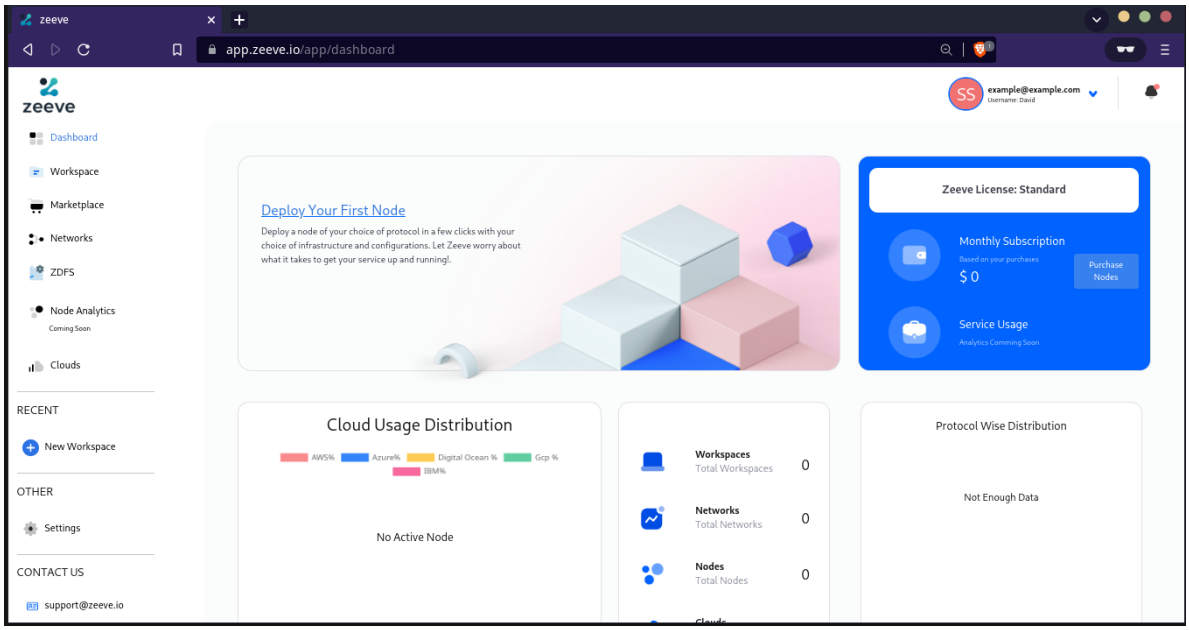
1. Click on **Sign-Up** and fill all the required details. Your email-id will act as your username for login.



2. This will give you a account verification email, clicking which will redirect you to zeeve where you can do successful login.



3. After successful login, you will be landed to Zeeve's dashboard.


zeeve
app.zeeve.io/app/dashboard
zeeve
example@example.com
Dashboard
Workspace
Marketplace
Networks
ZDFS
Node Analytics
Coming Soon
Clouds
RECENT
New Workspace
OTHER
Settings
CONTACT US
support@zeeve.io
Deploy Your First Node
Deploy a node of your choice of protocol in a few clicks with your choice of infrastructure and configurations. Let Zeeve worry about what it takes to get your service up and running!.
Zeeve License: Standard
Monthly Subscription
$ 0
Purchase Nodes
Service Usage
Cloud Usage Distribution
AWS% Azure% Digital Ocean % Gcp % IBM%
No Active Node
Workspaces
Total Workspaces
0
Networks
Total Networks
0
Nodes
Total Nodes
0
Protocol Wise Distribution
Not Enough Data

CHAPTER

SIX

DESCRIPTION: EXPLORE THE BEST PRACTICES AND TECHNICAL DETAILS FOR MANAGING AUTHORIZATION IN ZEEVE'S CLOUD-BASED PLATFORM. LEARN HOW TO SECURE YOUR DATA AND CONTROL ACCESS TO RESOURCES WITH OUR API AND TOOLS.

CHAPTER

SEVEN

CLOUD AUTHORIZATIONS

Zeeve allows you to authorize multiple cloud accounts of yours so as to create networks in the cloud of your choice. You may choose to deploy some nodes of network on one cloud and extend some nodes of the same on another. This cross cloud deployment maybe a major requirement for your usecase or clients especially for creating/expanding consortiums.

Zeeve supports a list of cloud for you to choose from. You can authorize multiple clouds and choose between them at the time of creating networks or nodes. Following is the list of currently supported clouds:-

1. AWS
2. Digital Ocean
3. Google Cloud
4. Tencent Cloud

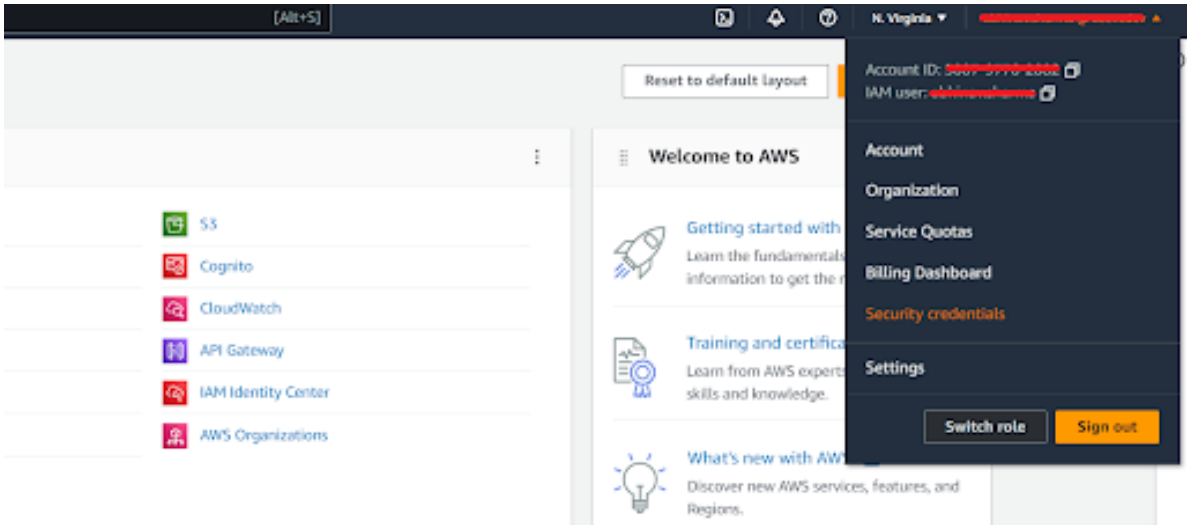
INTERESTING FACT: *Zeeve doesn't use* ***blockchain services*** *of any of the supported cloud platforms, and hence is not restricted for the level of features it can provide for a protocol on any cloud.*

7.1 AWS Authorization

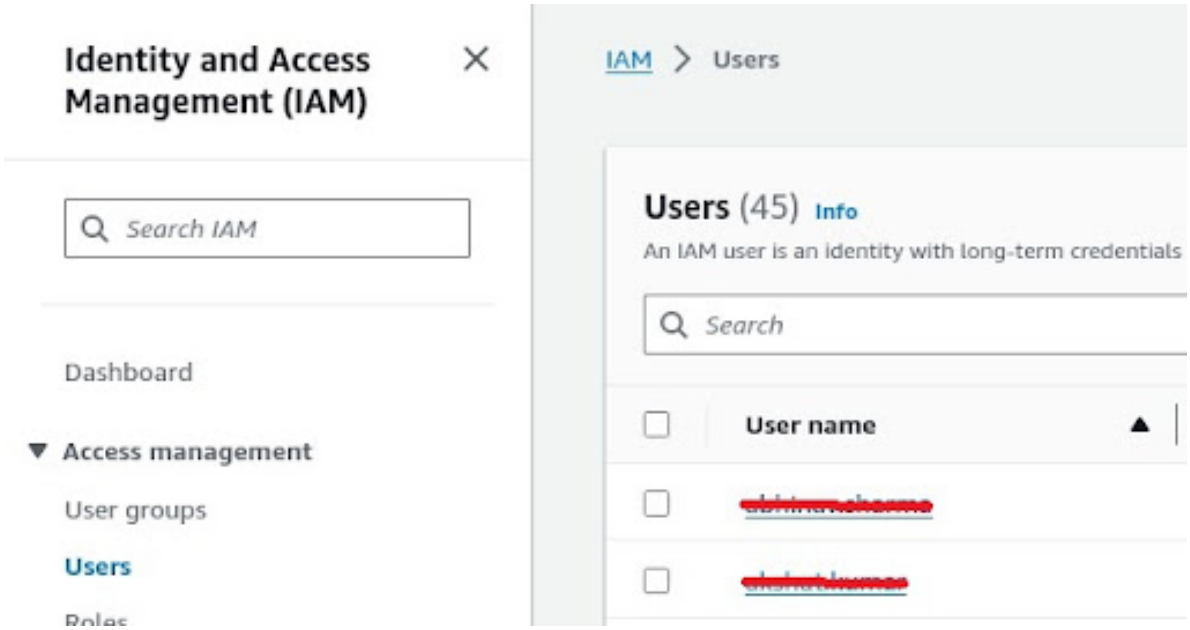
7.1.1 Configuration on AWS Portal

Before you authorize your AWS account with Zeeve, you'll need following provide IAM permissions to deploy a network:

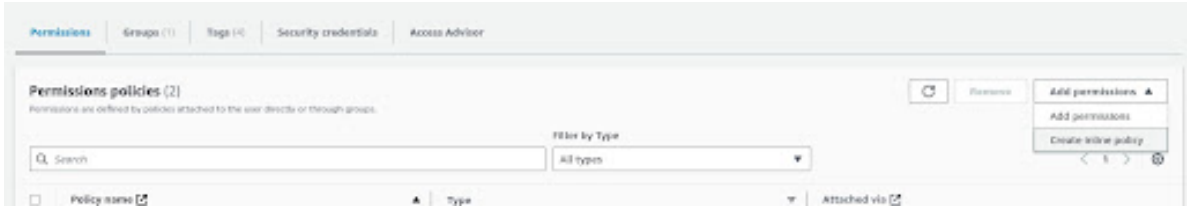
1. Login into the AWS console, go to IAM service by clicking Security Credential on the upper right corner.



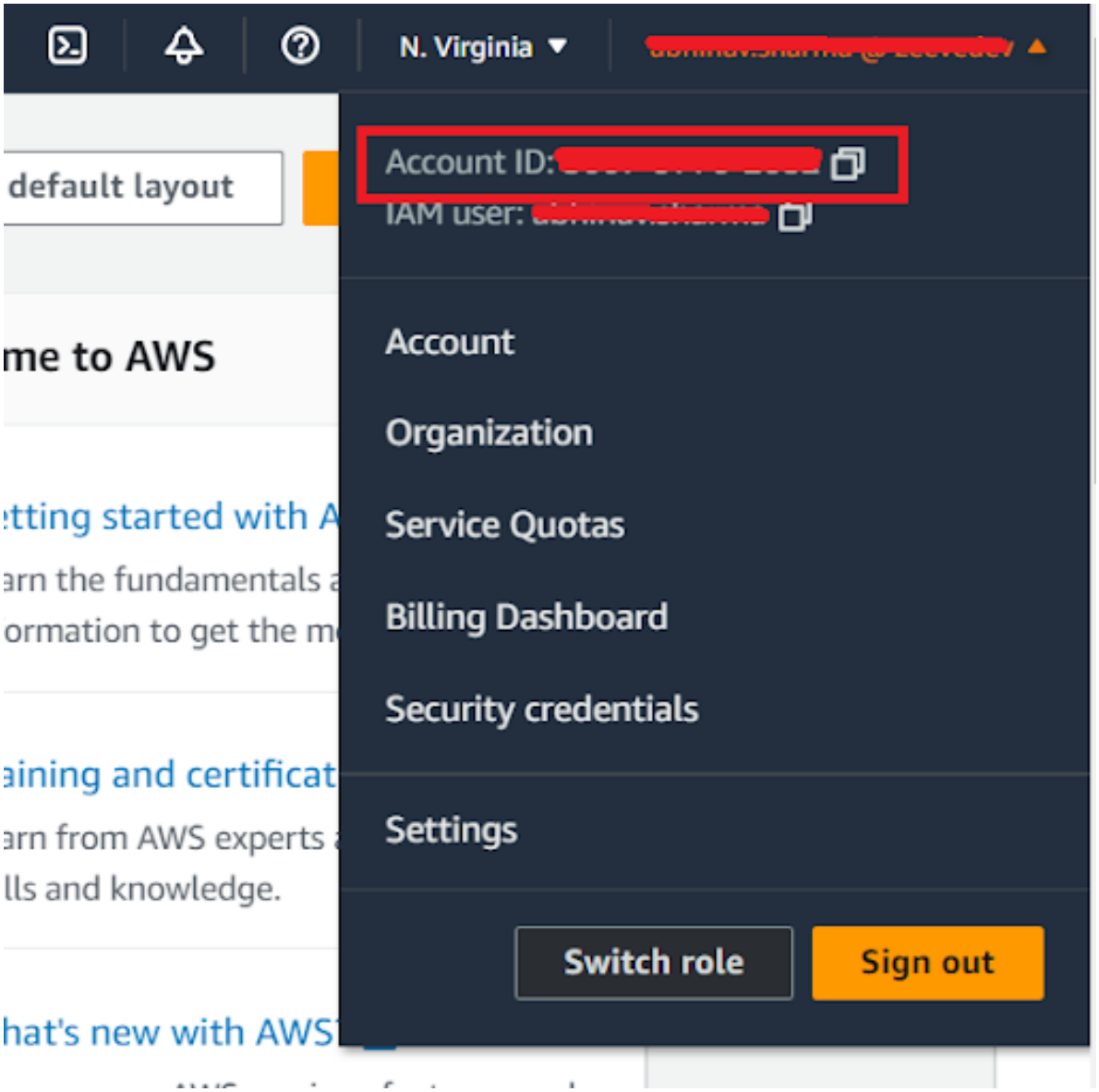
2. Select the **User** for which needs to be authorized on the Zeeve platform.



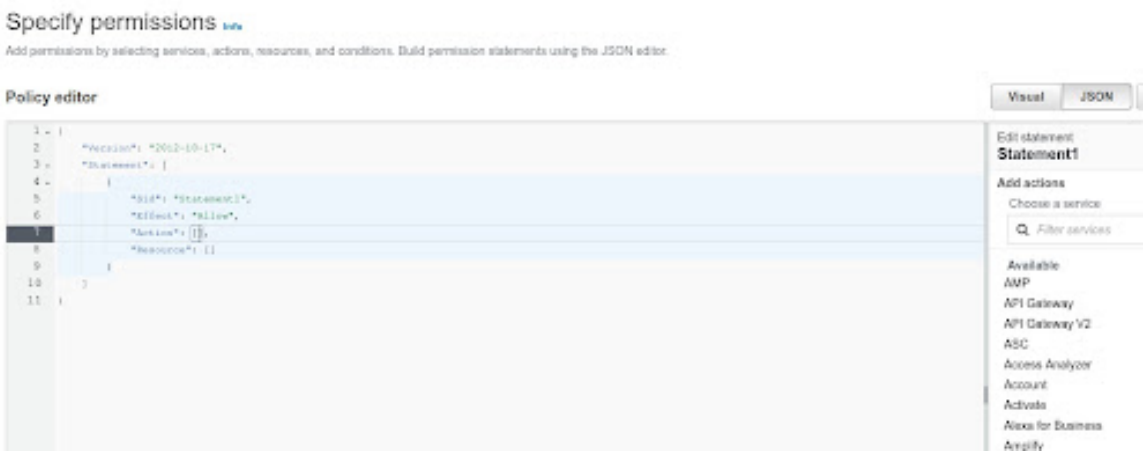
3. Click on Add Permissions button, and Create Inline Policy.



4. Copy the **ACCOUNTID** from the upper-right corner (we will need this Account ID in Further Steps).



5. Click on the JSON button, and add the below mentioned policy, (Please do replace **ACCOUNTID** with your AWS Account ID)



6. Write the Policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "logs:CreateLogStream",
               "logs:TagLogGroup",
               "logs:DeleteLogGroup",
               "logs:TagResource",
               "logs:PutRetentionPolicy",
               "logs:CreateLogGroup",
               "logs:DeleteDestination",
               "logs:TagResource",
               "logs:ListTagsLogGroup"
           ],
           "Resource": [
               "arn:aws:logs:*:ACCOUNTID:destination:*",
               "arn:aws:logs:*:ACCOUNTID:log-group:*"
           ]
       },
       {
           "Effect": "Allow",
           "Action": [
               "logs:DescribeLogGroups",
               "logs:DescribeDestinations"
           ],
           "Resource": "*"
       },
       {
            "Effect": "Allow",
            "Action": [
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:GetUserPolicy",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": [
```

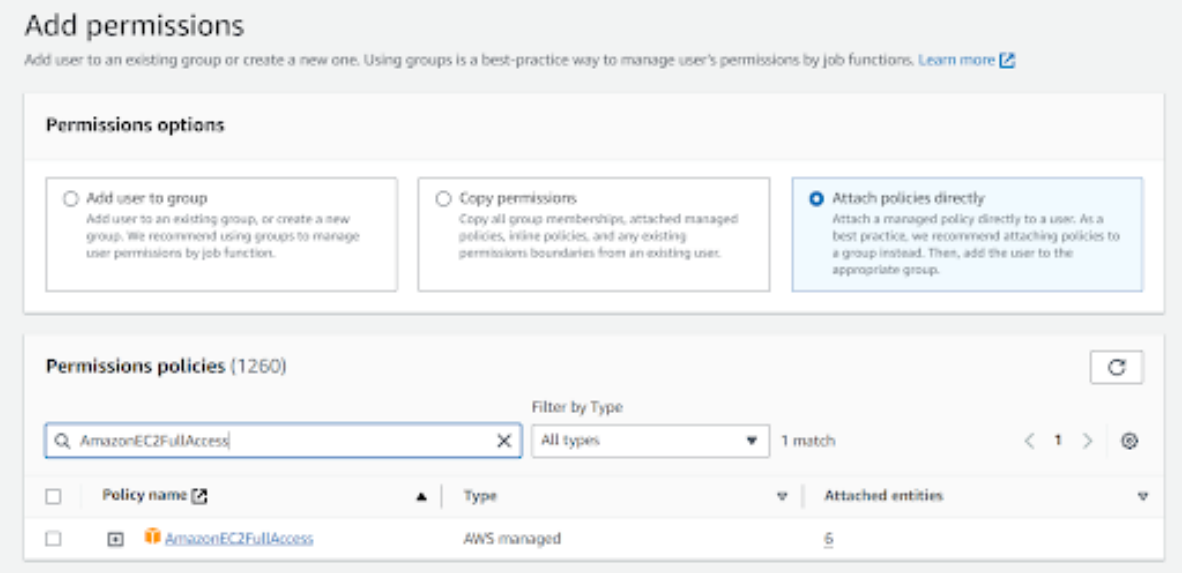
(continues on next page)

(continued from previous page)

```
                "*"
            ]
        }
    ]
}
```

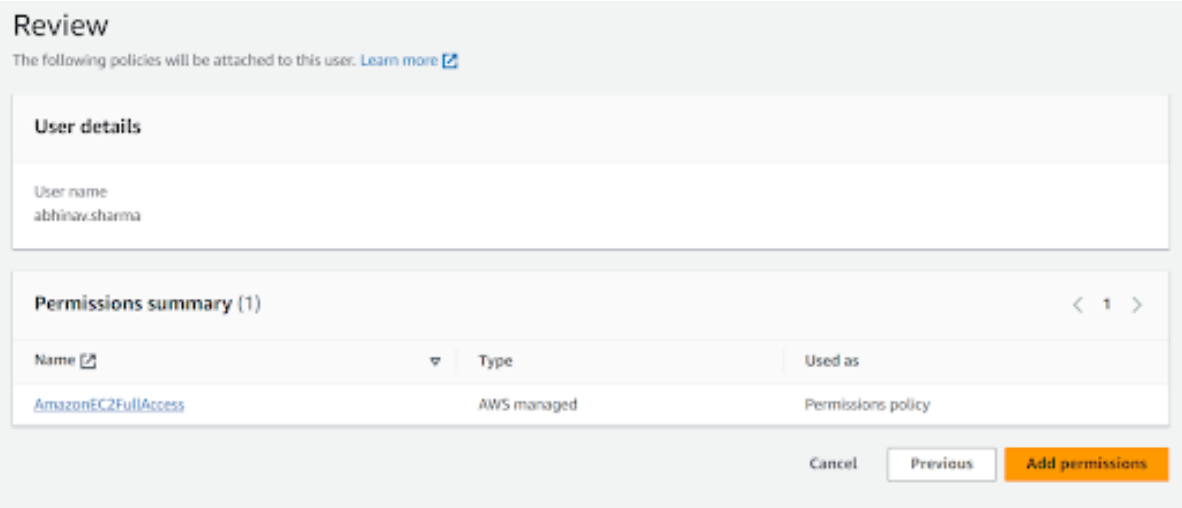
1. Name this policy as **Zeeve-IAM-Policy**, and click Save.
2. Grant AWS Managed EC2 Permissions:AmazonEC2FullAccess. a. Click on **Add Permissions -> Add Permissions**

   b. Click on Attach Policies Directly and Search for **AmazonEC2FullAccess**.



   c. Select the permission and Click on Next.

   d. On the Review Page, Review the Policy and Click on Add Permissions.



3. *(Only Permissive Protocol)* Grant below permissions for Permissive protocols.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "elasticfilesystem:CreateFileSystem",
               "eks:ListClusters",
               "eks:DescribeAddonVersions",
               "eks:RegisterCluster",
               "eks:CreateCluster"
           ],
           "Resource": "*"
       },
       {
           "Effect": "Allow",
           "Action": [
               "elasticfilesystem:DescribeMountTargets",
               "elasticfilesystem:DeleteAccessPoint",
               "elasticfilesystem:CreateMountTarget",
               "elasticfilesystem:DescribeLifecycleConfiguration",
               "elasticfilesystem:DescribeFileSystems",
               "elasticfilesystem:DeleteMountTarget",
               "elasticfilesystem:CreateAccessPoint",
               "elasticfilesystem:DeleteFileSystem",
               "elasticfilesystem:DescribeMountTargetSecurityGroups",
               "elasticfilesystem:TagResource"
           ],
           "Resource": [
               "arn:aws:elasticfilesystem:*:ACCOUNTID:file-system/*",
               "arn:aws:elasticfilesystem:*:ACCOUNTID:access-point/*"
           ]
       },
       {
           "Effect": "Allow",
           "Action": "eks:*",
           "Resource": [
               "arn:aws:eks:*:ACCOUNTID:cluster/*",
               "arn:aws:eks:*:ACCOUNTID:nodegroup/*/*/*",
               "arn:aws:eks:*:ACCOUNTID:fargateprofile/*/*/*",
               "arn:aws:eks:*:ACCOUNTID:addon/*/*/*",
               "arn:aws:eks:*:ACCOUNTID:identityproviderconfig/*/*/*/*"
           ]
       },
       {
           "Effect": "Allow",
           "Action": [
               "secretsmanager:CreateSecret",
               "secretsmanager:UpdateSecret",
               "secretsmanager:DescribeSecret",
               "secretsmanager:GetSecretValue",
               "secretsmanager:PutSecretValue",
               "secretsmanager:ReplicateSecretToRegions",
               "secretsmanager:TagResource"
           ],
           "Resource": [
               "*"
```

(continues on next page)

(continued from previous page)

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:UpdateAssumeRolePolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:DeletePolicy",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:AttachRolePolicy",
                "iam:CreateOpenIDConnectProvider",
                "iam:CreatePolicy",
                "iam:ListInstanceProfilesForRole",
                "iam:PassRole",
                "iam:DetachRolePolicy",
                "iam:ListPolicyVersions",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:GetOpenIDConnectProvider",
                "iam:DeleteOpenIDConnectProvider"
            ],
            "Resource": [
                "arn:aws:iam::ACCOUNTID:policy/*",
                "arn:aws:iam::ACCOUNTID:oidc-provider/*",
                "arn:aws:iam::ACCOUNTID:role/*"
            ]
        }
    ]
}
```

1. Follow steps 5-7, and Name this Policy as **Zeeve-Permissive-Protocol-Policy**.
2. *(Only Corda Enterprise)* Grant below permissions for Corda Enterprise.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                    "route53:GetHostedZone",
                    "route53:CreateHostedZone",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:TagRole",
                "route53:GetChange",
                "route53:ChangeResourceRecordSets",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:PutRolePolicy",
                "route53:ListTagsForResource",
                "iam:AddRoleToInstanceProfile",
                "route53:ListTagsForResources",
                "iam:DeleteRolePolicy",
```

(continues on next page)

(continued from previous page)

```
                "route53:ListResourceRecordSets",
                "route53:AssociateVPCWithHostedZone"
            ],
            "Resource": [
                "arn:aws:route53:::hostedzone/*",
                "arn:aws:route53:::healthcheck/*",
                "arn:aws:route53:::change/*",
                "arn:aws:iam::ACCOUNTID:role/*",
                "arn:aws:iam::ACCOUNTID:instance-profile/*",
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                    "kms:PutKeyPolicy",
                    "kms:DescribeKey",
                    "kms:CreateGrant",
                "kms:EnableKeyRotation",
                "kms:Decrypt",
                "kms:GetKeyRotationStatus",
                "kms:GenerateDataKey",
                "route53:DeleteHostedZone",
                "kms:GenerateDataKeyPair",
                "kms:CreateGrant",
                "kms:ScheduleKeyDeletion",
                "kms:GetKeyPolicy",
                "kms:ListResourceTags",
                "kms:TagResource"
            ],
            "Resource": [
                "arn:aws:kms:*:ACCOUNTID:key/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                    "ecr:DeleteRepository",
                "ecr:PutImage",
                    "ecr:DeleteRepository",
                    "ecr:TagResource",
                "ecr:ListTagsForResource",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:DescribeRepositories",
                "ecr:InitiateLayerUpload",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "arn:aws:ecr:*:ACCOUNTID:repository/*",
        },
        {
            "Action": "ec2:*",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "ec2:*",
            "Effect": "Allow",
```

(continues on next page)

(continued from previous page)

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "eks:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "rds:AddTagsToResource",
                "rds:DescribeDBSubnetGroups",
                "ecr:CreateRepository",
                "rds:DescribeGlobalClusters",
                "route53:ListHostedZones",
                "ecr:GetAuthorizationToken",
                "rds:CreateDBSubnetGroup",
                "rds:DeleteDBSubnetGroup",
                "rds:ListTagsForResource",
                "rds:CreateDBCluster",
                "rds:CreateDBInstance",
                "rds:DescribeDBInstances",
                "kms:CreateKey",
                "rds:DeleteDBCluster",
                "rds:DescribeDBClusters",
                "rds:DeleteDBInstance"
            ],
            "Resource": "*"
        }
    ]
}
```

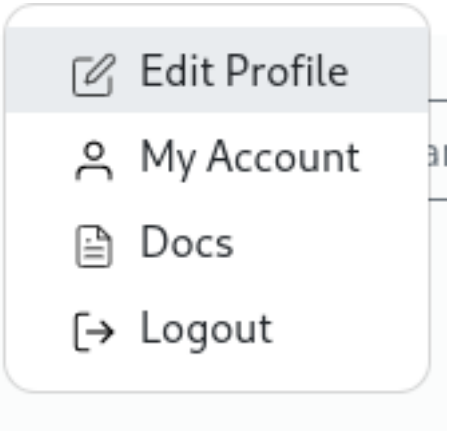
1. Follow steps 5-7, and Name this Policy as **Zeeve-Corda-Enterprise-Policy**.

7.1.2 Configuration on Zeeve Portal

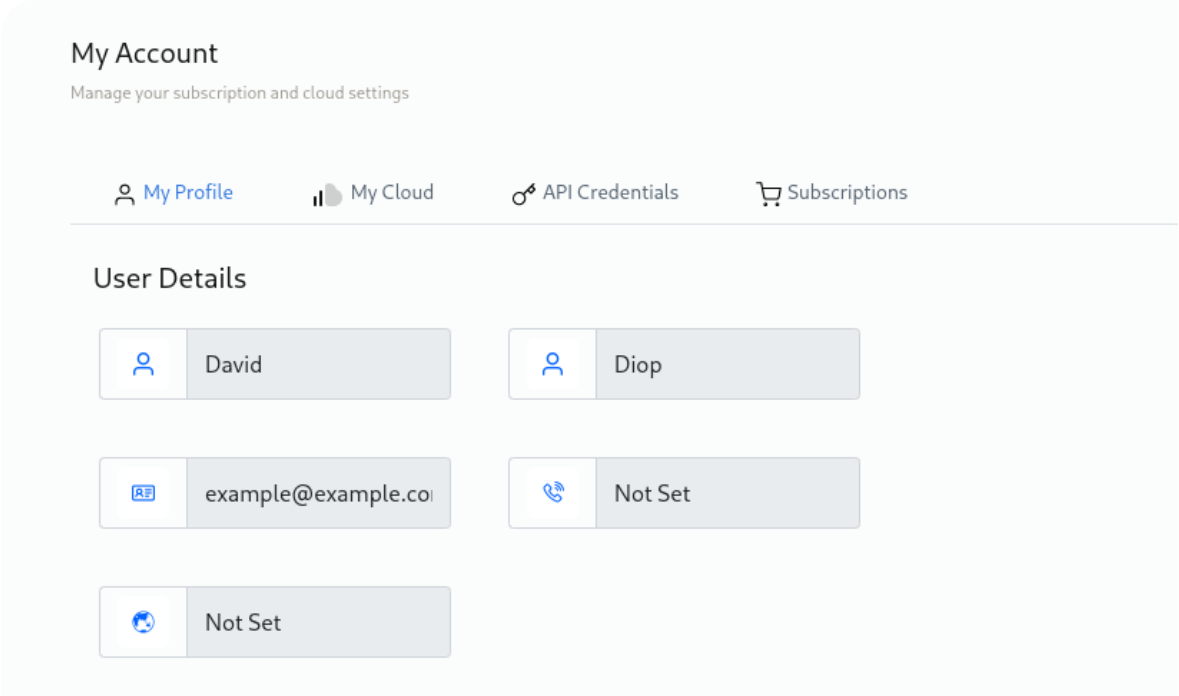
To authorize your AWS account on Zeeve:-

1. Hover on **profile**

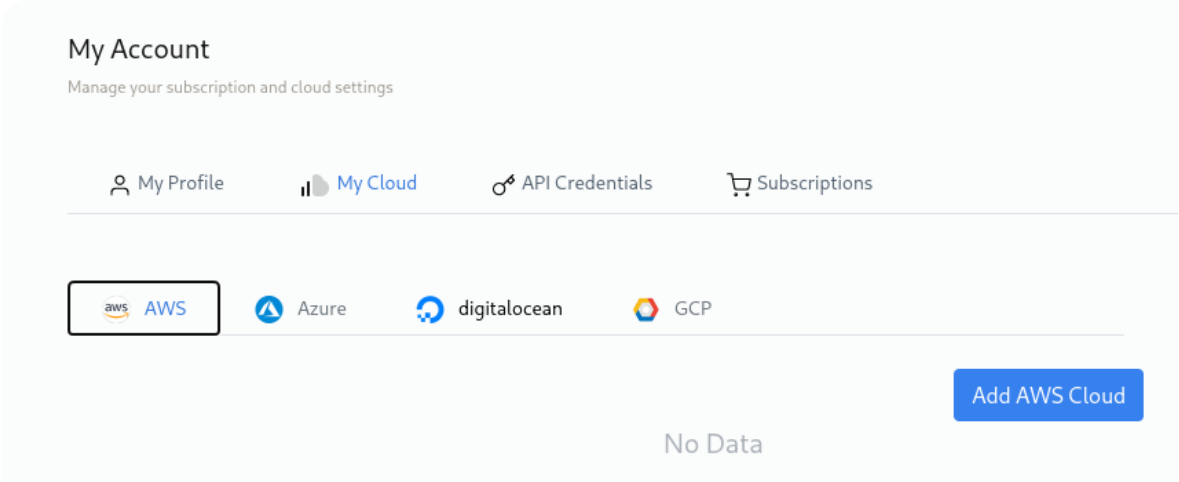
2. Click on **My Account**



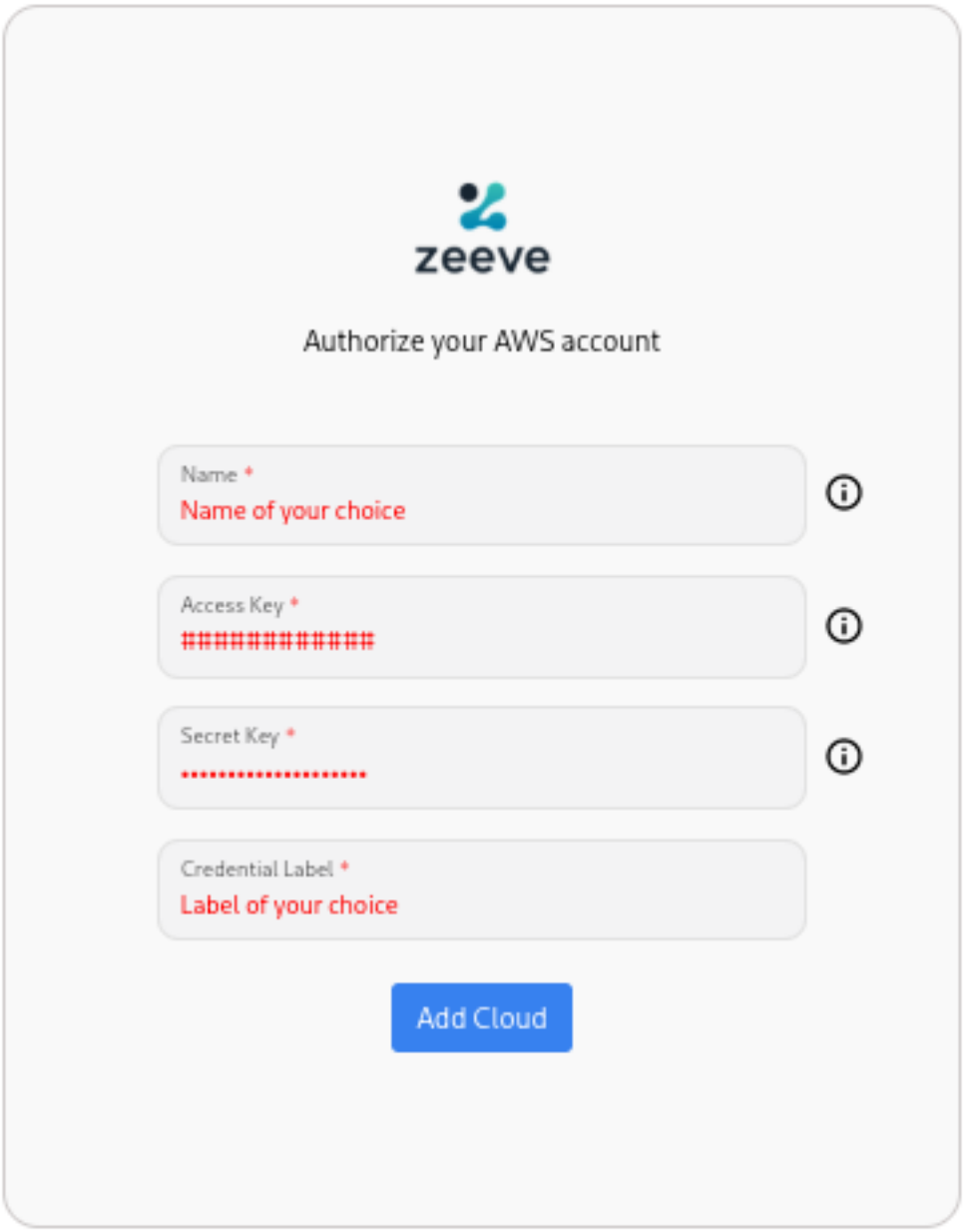
3. Click on **My Cloud**.



4. Click on **AWS** and then click on **Add AWS Cloud**.



5. You will need AWS Access Key and AWS Access Secret Key, to authenticate your AWS account with Zeeve.



7.2 Digital Ocean Authorization

To authorize your Digital Ocean account on Zeeve you'll need to ensure certain things:-

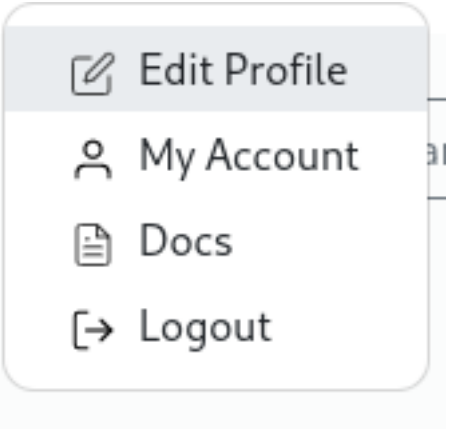
- User must have an account with enough permissions to create -
- Project
- Droplets
- Kubernetes service.
- Specific Scope in DO Account:

- Read
- Write

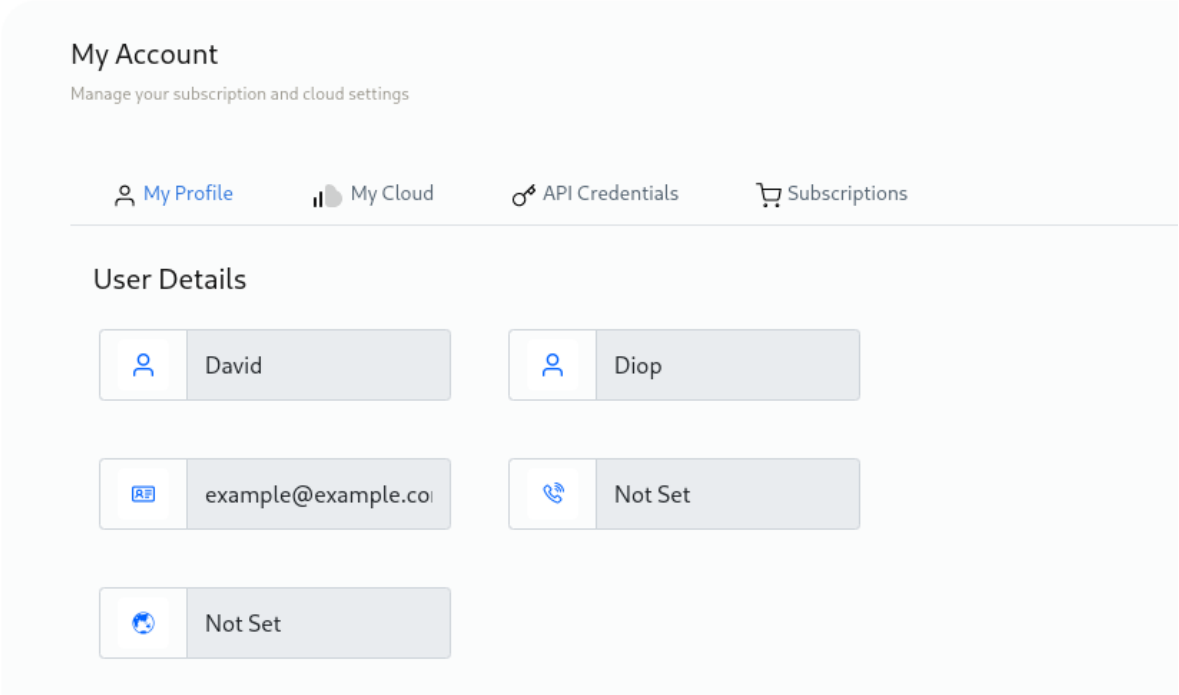
After which on Zeeve do following steps:-

1. Hover on **profile**

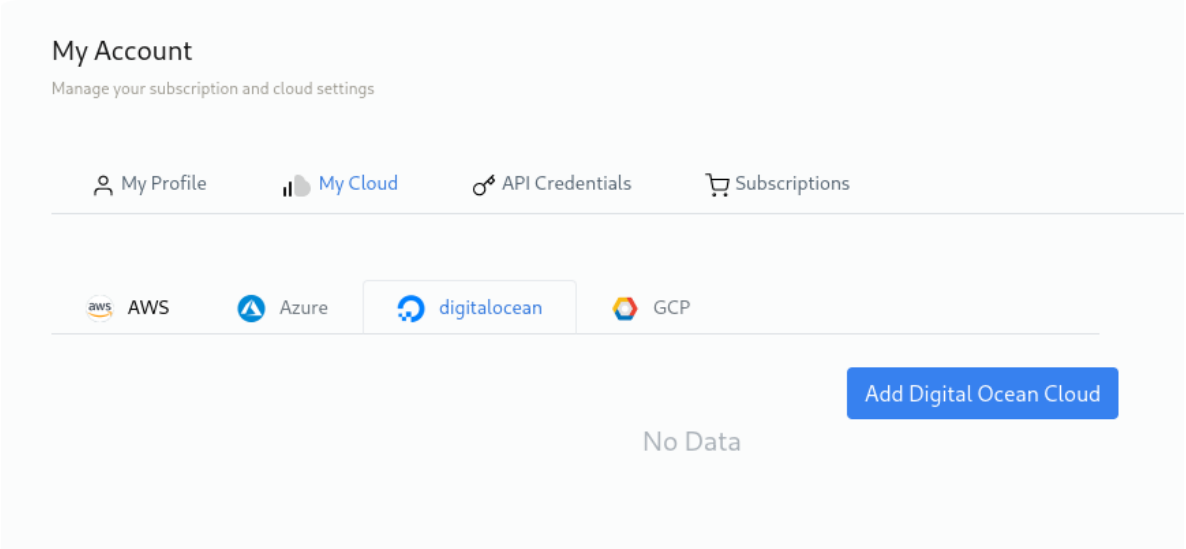
2. Click on **My Account**



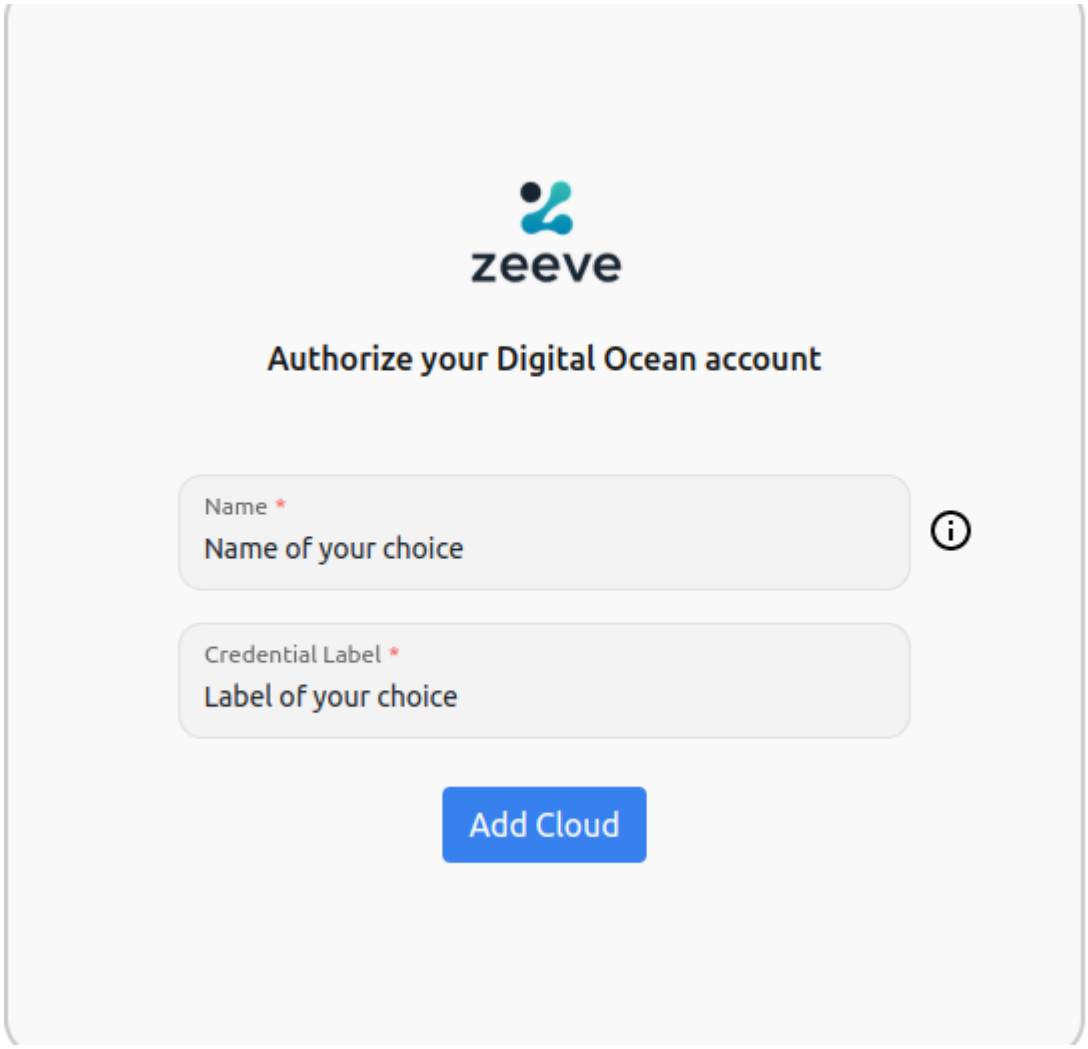
3. Click on your cloud authentication for Digital Ocean account, click on **Authorize digital Ocean**.



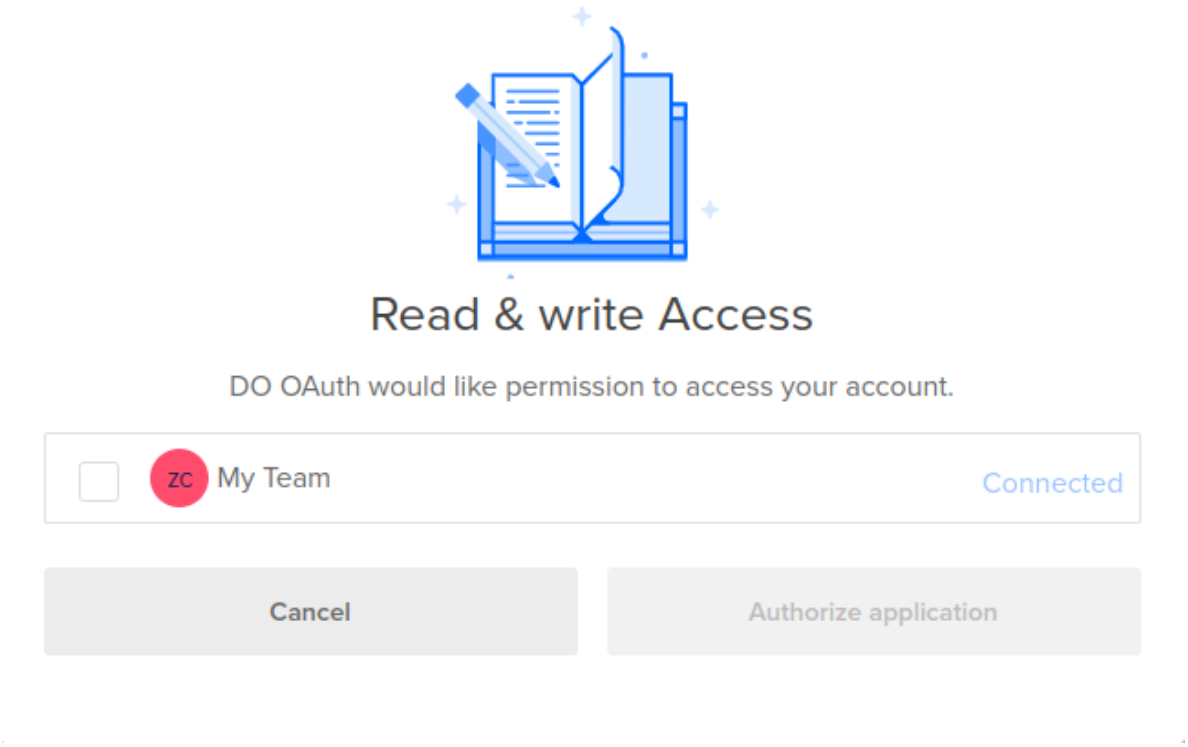
4. Click on **DigitalOcean** and then click on **Add Digital Ocean Cloud**.



5. Authorize DigitalOcean will redirect you to login page, you can add your DigitalOcean credentials and then click on **Add Cloud**.



6. After that click on the team which you want to give access and click on **Authorize Application**.



7.3 GCP Cloud Authorization

To authorize your GCP account on Zeeve you'll need to ensure certain things:-

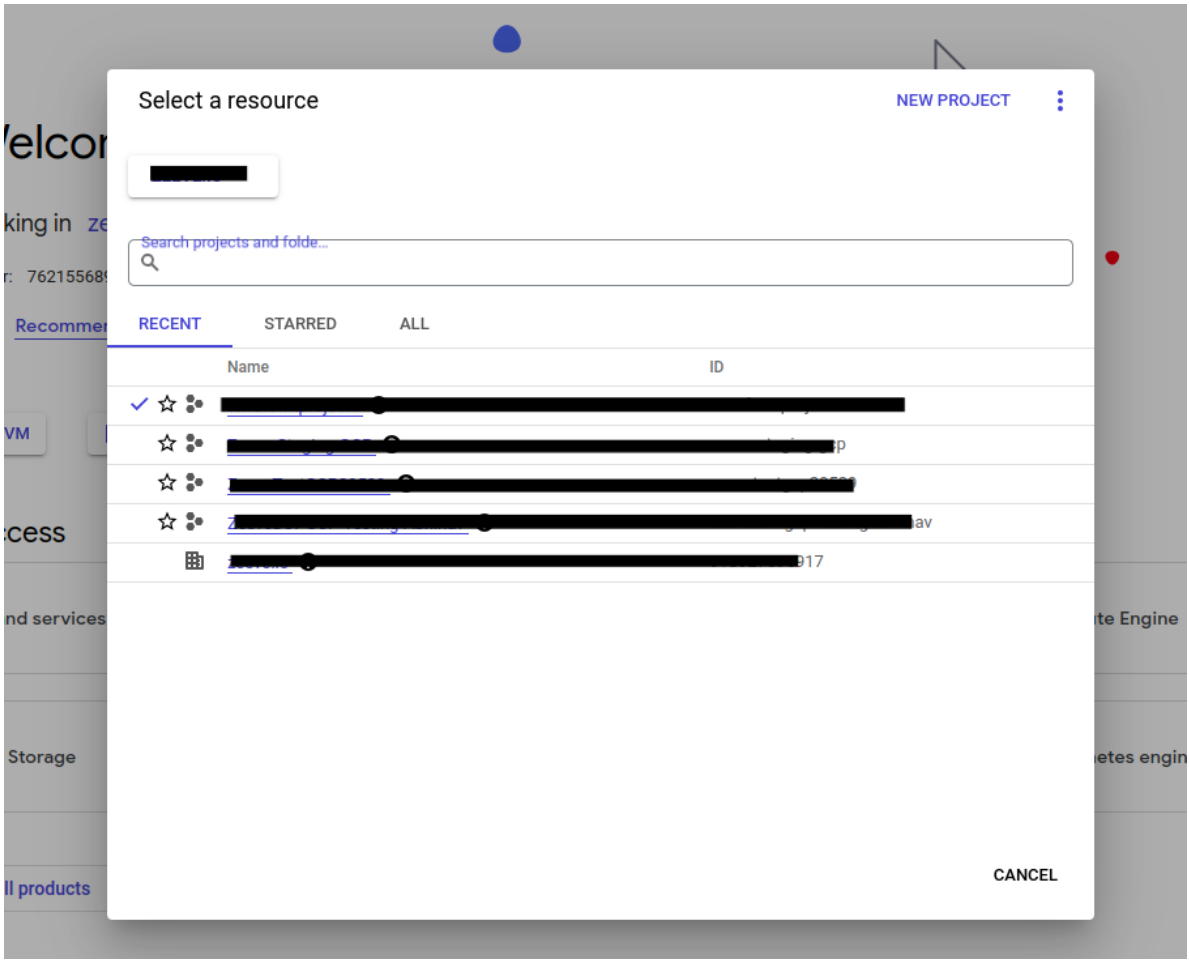
7.3.1 Configuration on GCP Portal

Enable Below APIs from Google Cloud Platform:

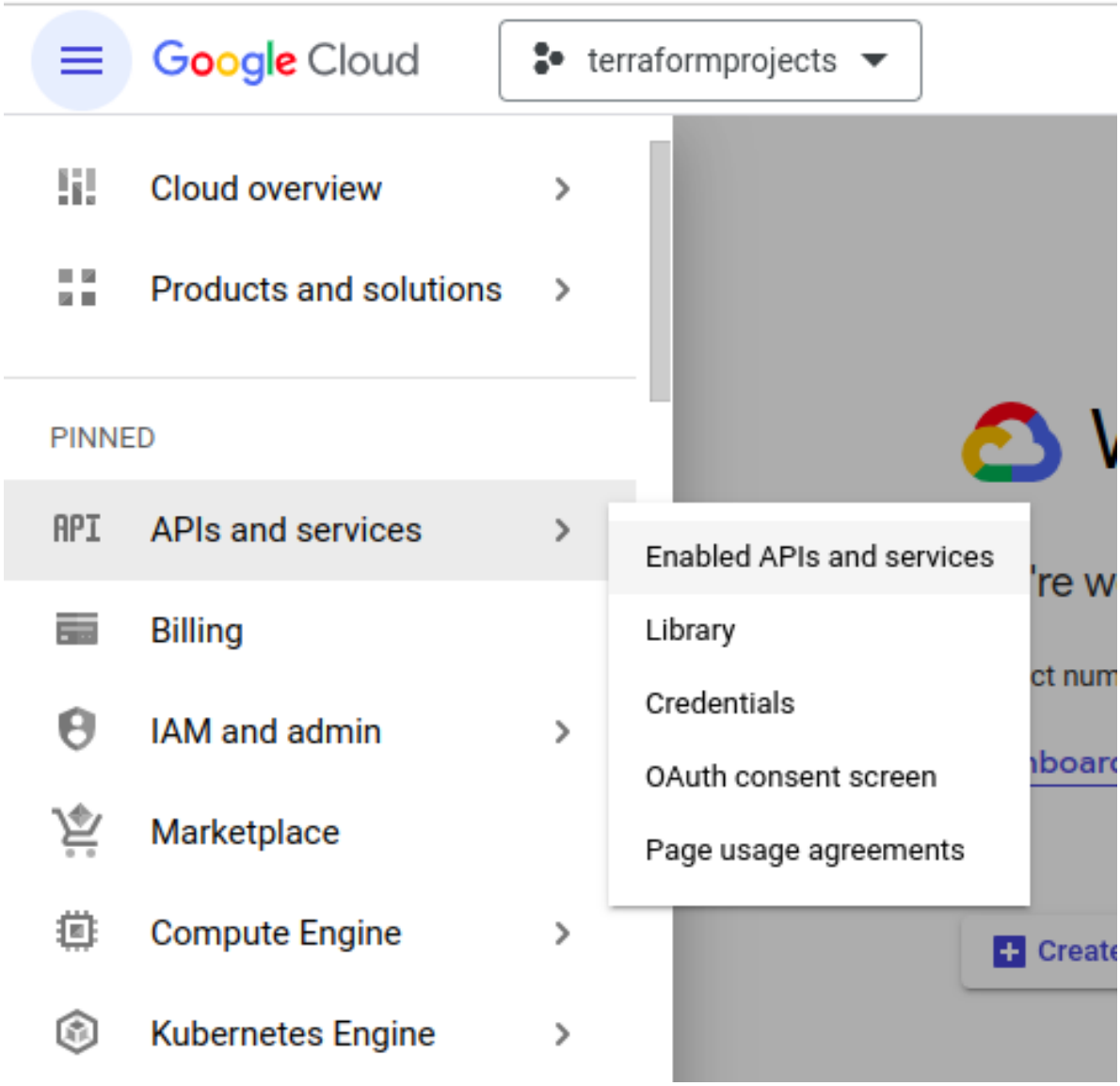
- Compute Engine API
- Kubernetes Engine API

Steps to enable APIs on GCP Platform

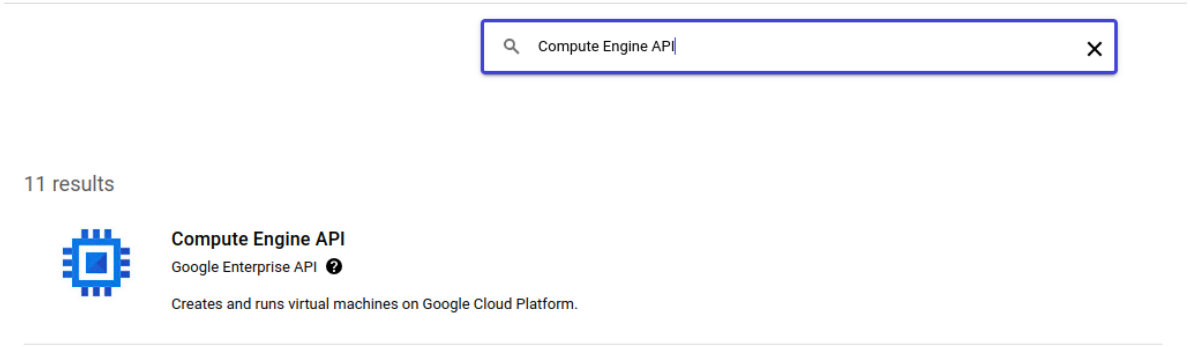
1. Go to Google Cloud console: https://console.cloud.google.com/
2. Select the project, in which you need to enable APIs.



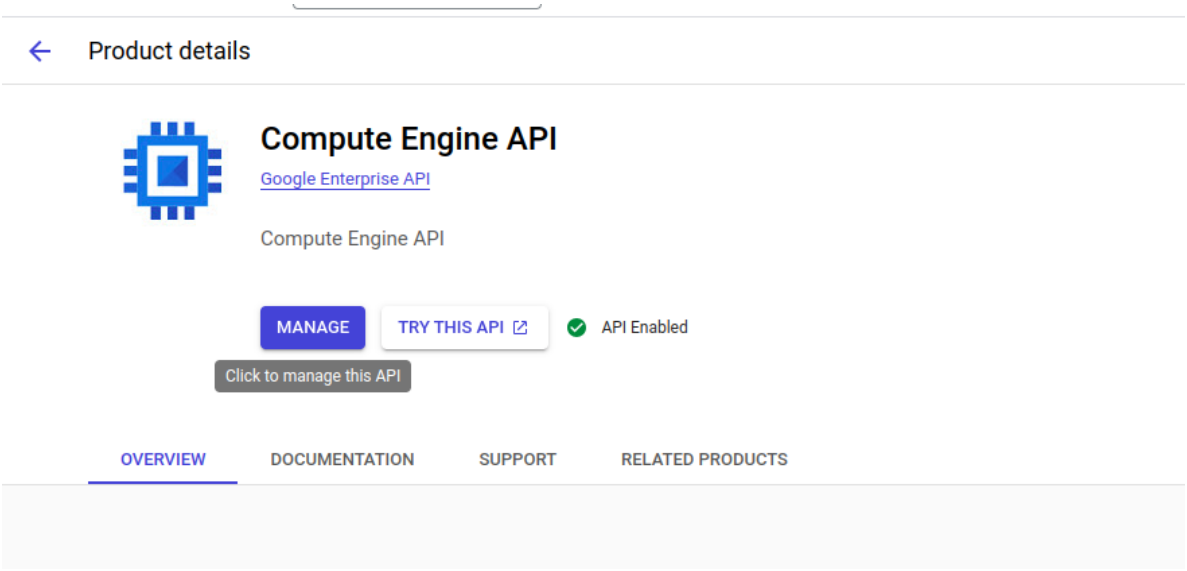
3. Click on **APIs and services** from the navigation bar and click **Enabled APIs and services**



4. Click on **+ Enable API and Services**
5. Search **Compute Engine API** in search bar.



6. Click on **Compute Engine API** -> **Enable/Manage** Option



7. Follow 5-6 step for enabling **Kubernetes Engine API**.
8. User must have an account with enough permissions to create -

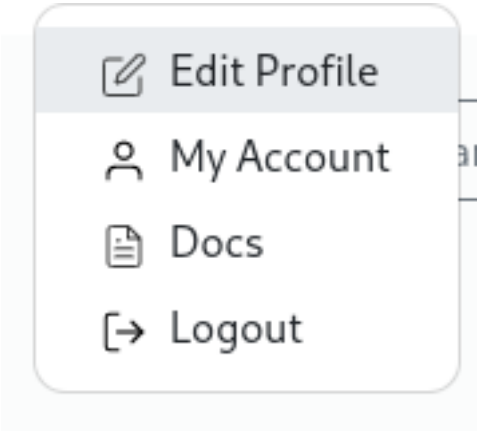
- Specific Permissions in GCP Account:
  - 'compute.globalOperations.get'
  - 'compute.machineTypes.get'
  - 'compute.networks.create'
  - 'compute.networks.delete'
  - 'compute.networks.get'
  - 'compute.networks.updatePolicy'
  - 'compute.projects.get'
  - 'compute.regionOperations.get'
  - 'compute.regions.get'
  - 'compute.routers.create'
  - 'compute.routers.delete'
  - 'compute.routers.get'
  - 'compute.routers.update'
  - 'compute.routes.create'
  - 'compute.routes.delete'
  - 'compute.routes.get'
  - 'compute.subnetworks.create'
  - 'compute.subnetworks.delete'
  - 'compute.subnetworks.get'
  - 'compute.zones.list'
  - 'resourcemanager.projects.get'

- 'compute.disks.create'
- 'compute.instances.create'
- 'compute.instances.get'
- 'compute.instances.setMetadata'
- 'iam.serviceAccounts.create'
- 'iam.serviceAccounts.delete'
- 'iam.serviceAccounts.get'
- 'iam.serviceAccountKeys.create'
- 'iam.serviceAccountKeys.delete'
- 'iam.serviceAccountKeys.get'
- 'iam.serviceAccounts.actAs',
- 'container.clusters.create'
- 'container.clusters.delete'
- 'container.clusters.get'
- 'container.clusters.getCredentials'
- 'container.clusters.update'
- 'container.operations.get'
- 'container.clusters.list'
- 'container.deployments.delete'
- 'container.deployments.get'
- 'container.namespaces.list'
- 'container.namespaces.get'
- 'container.services.get',
- 'compute.disks.createSnapshot'
- 'compute.snapshots.get'
- 'compute.snapshots.create'
- 'compute.snapshots.useReadOnly'
- 'compute.snapshots.delete'
- 'compute.zones.get'
- 'storage.objects.create'
- 'storage.objects.delete'
- 'storage.objects.get'
- 'storage.objects.list'
- 'iam.serviceAccounts.signBlob'

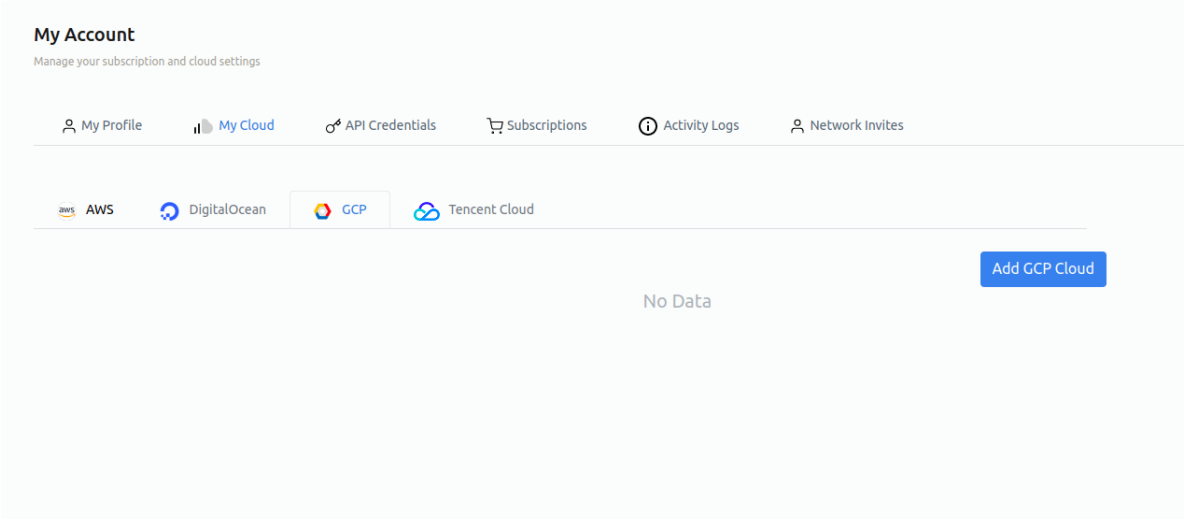
7.3.2 Configuration on Zeeve Portal

1. Hover on **profile**

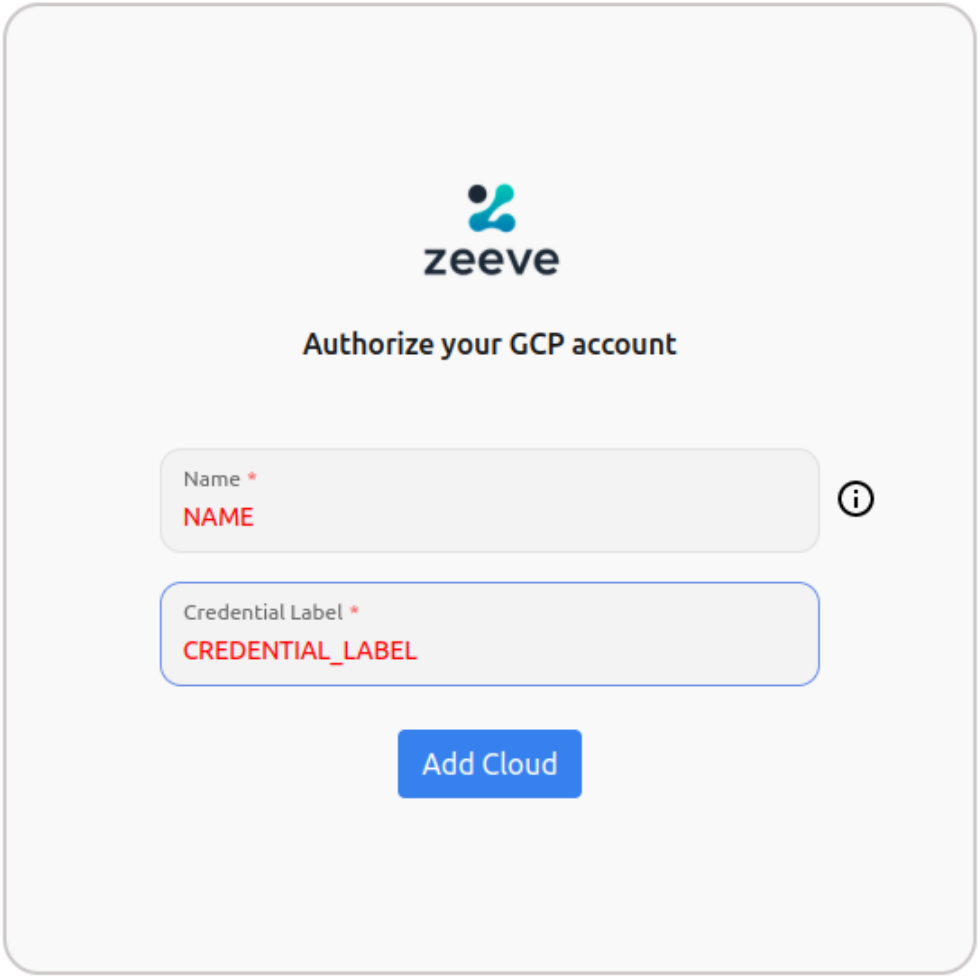
2. Click on **My Account**



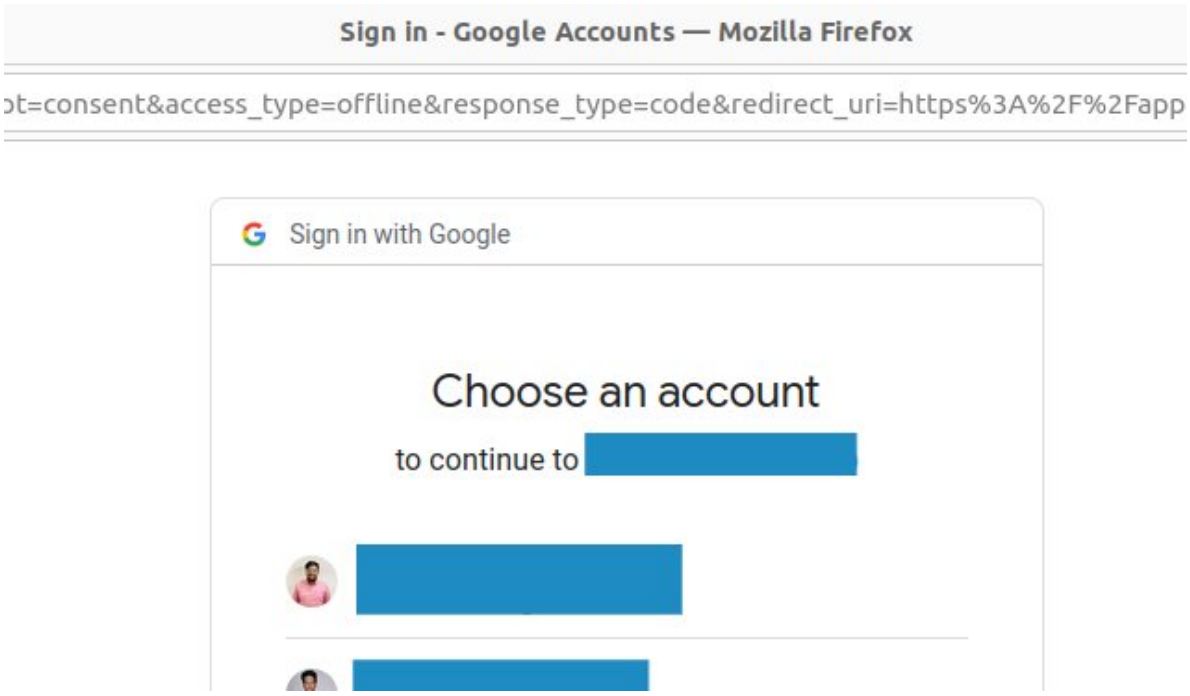
3. Click on **My Cloud**.
4. Click on **GCP** and then click on **Add GCP Cloud**.



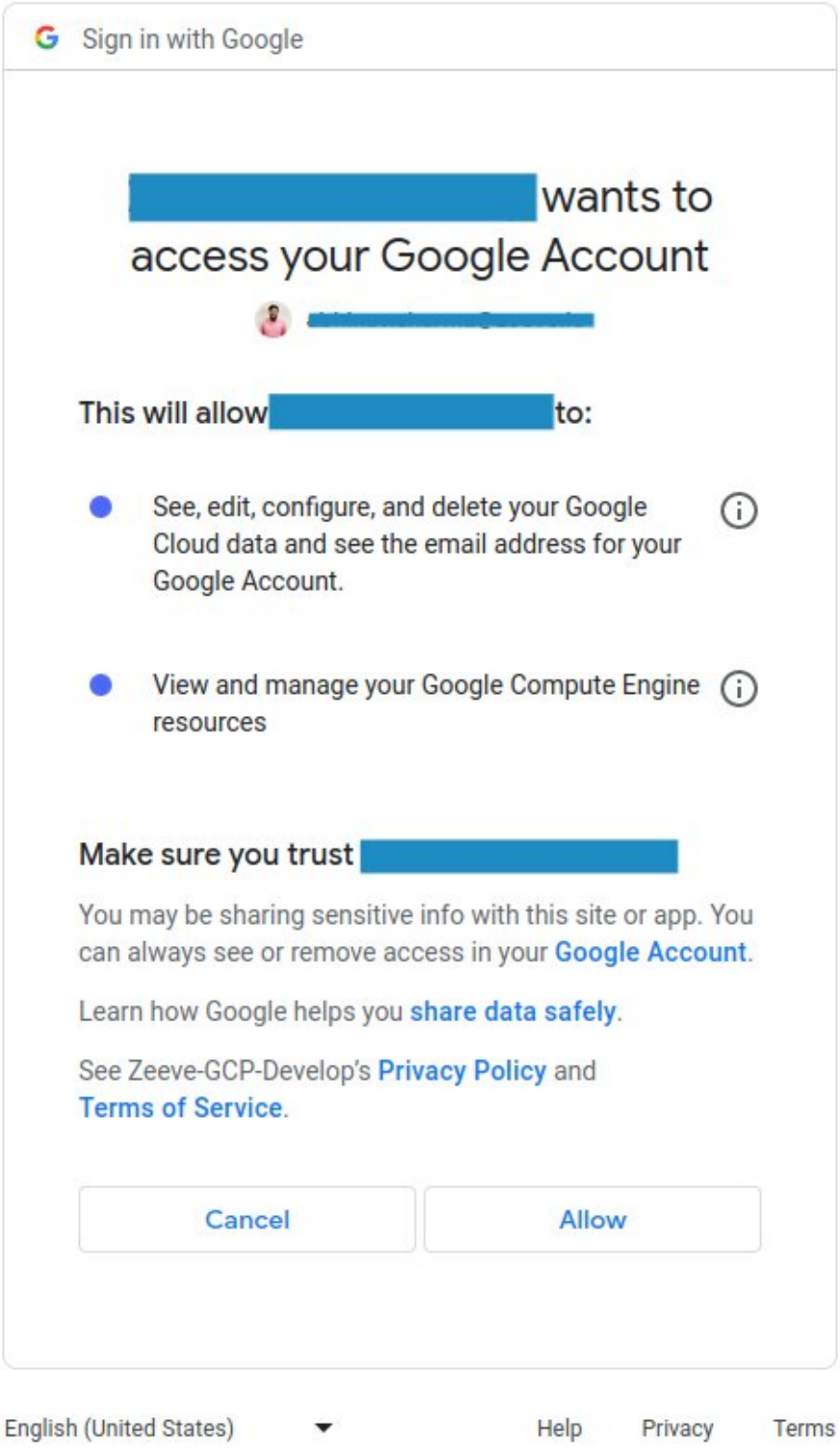
5. Authorize GCP will redirect you to login page, you can add your GCP credentials and then click on **Add Cloud**.



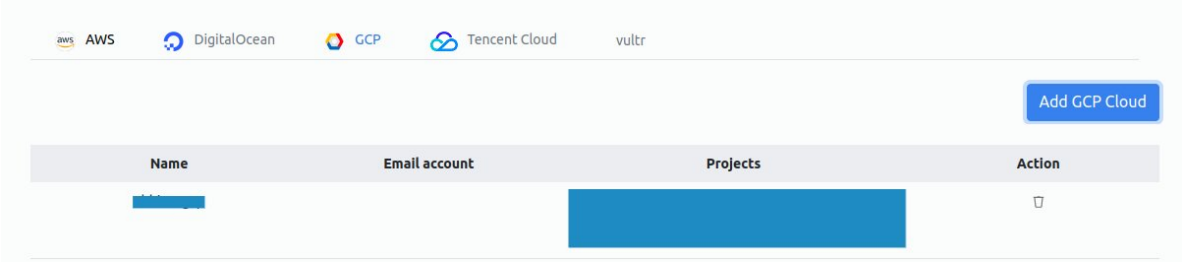
6. Login to your Google Cloud account using Google IDP.



7. Allow Zeeve to access your GCP Account.



8. You can view your creds in Zeeve Console.

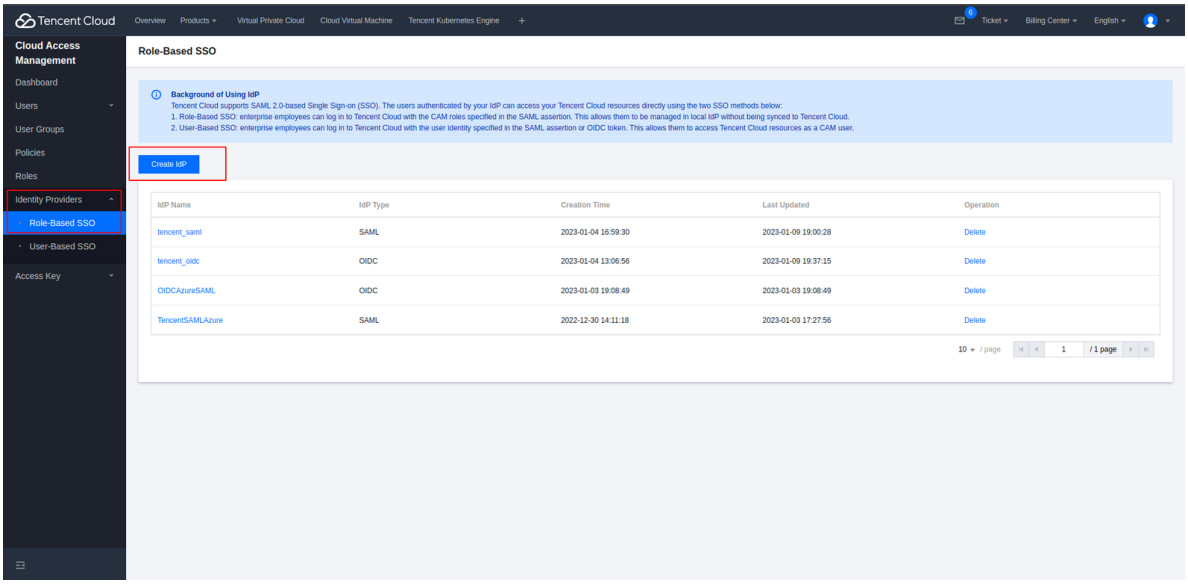


7.4 Tencent Cloud Authorization

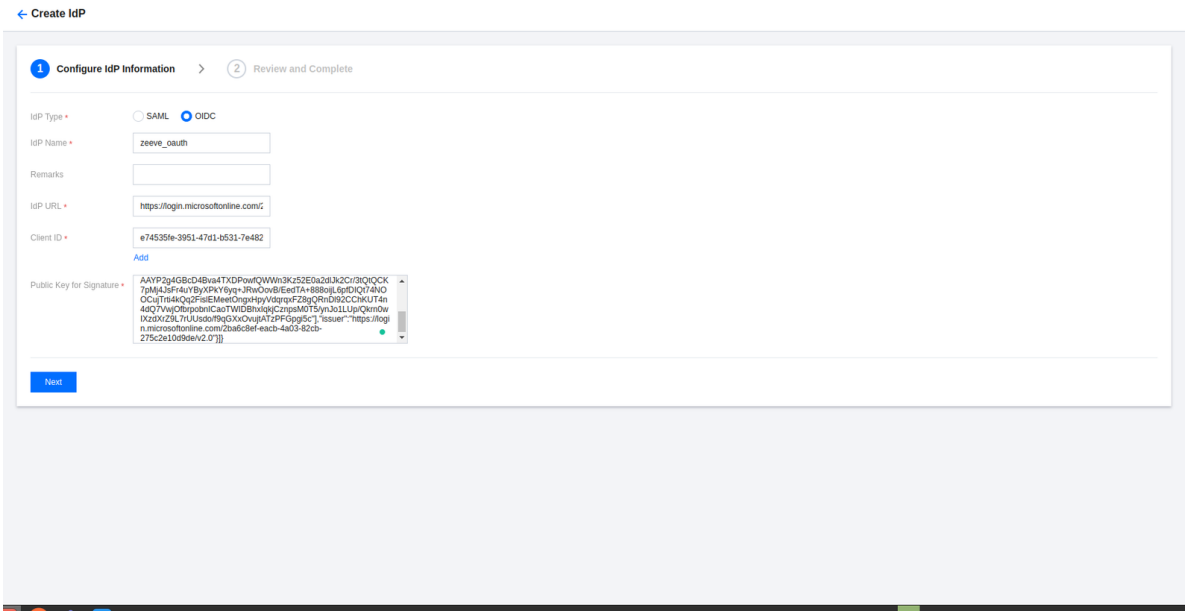
Before you authorize Tencent Cloud on Zeeve, you will need to add Zeeve's IDP into your Cloud account.

7.4.1 Creating an OIDC IdP

1. On the left sidebar in the CAM console, select **Identity Providers > Role-Based SSO**.



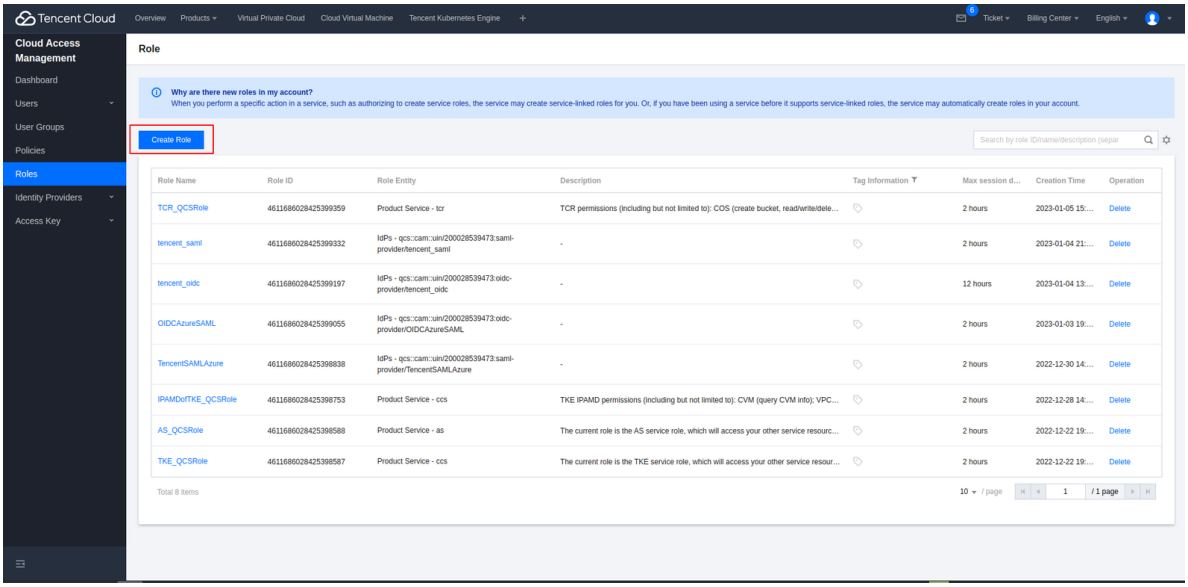
2. On the **Role-Based SSO** page, click **Create IdP**.
3. On the page you enter, select **OIDC** as the IdP type and enter the following IdP information. **IdP Name**: `zeeve_oauth` **IdP URL**: `https://login.microsoftonline.com/9188040d-6c67-4c5b-b112-36a304b66dad/v2.0` **Client ID**: `505b1146-13fe-4df6-927a-ca57321786fd` **Public Key for Signature**: For this you can click on this link (https://login.microsoftonline.com/common/discovery/v2.0/keys) then copy all the content and paste it in the column.
4. Click **Next** to enter the information review page.



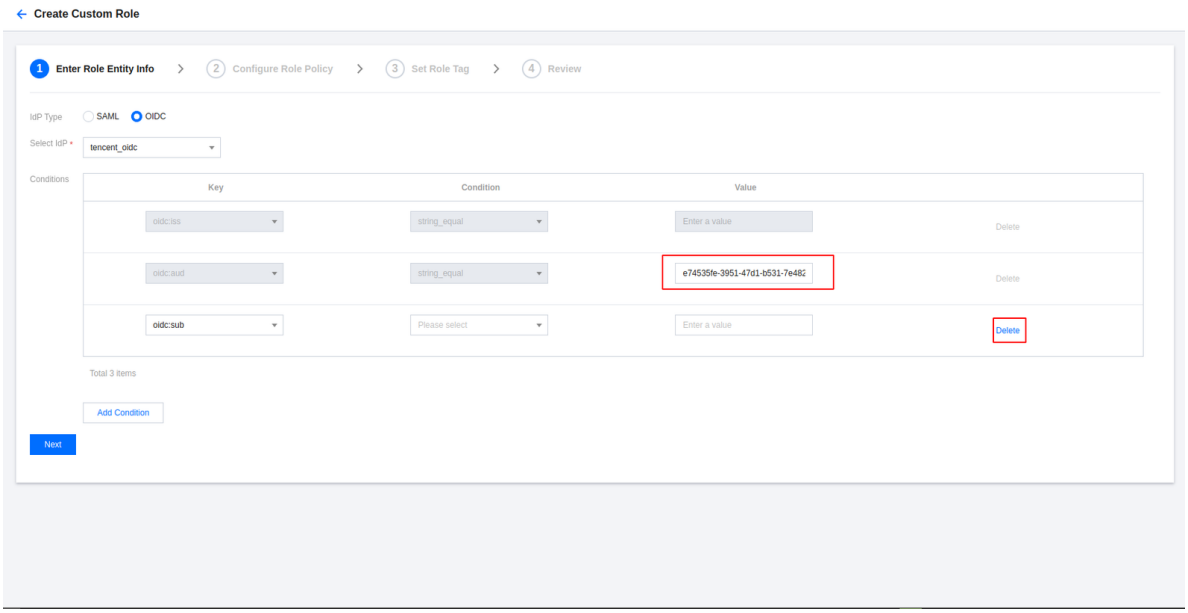
5. Confirm the information you entered and click **Complete** to save it.

7.4.2 Creating a role for the IdP

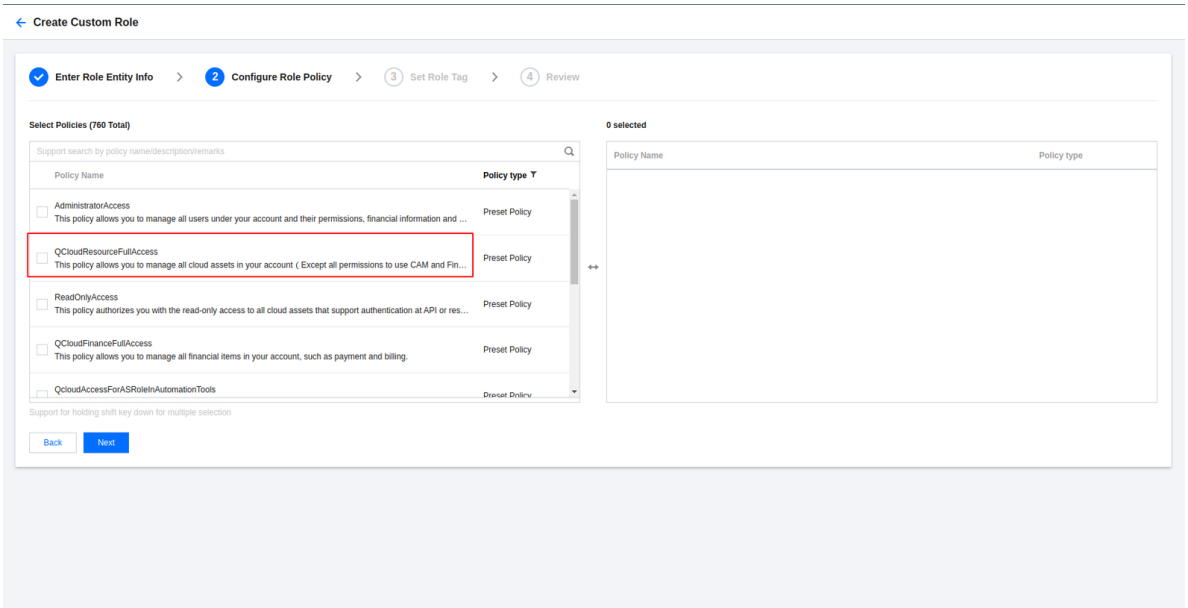
1. On the left sidebar in the CAM console, click **Roles**.



2. On the role management page, click **Create Role**.
3. Select **IdPs** as the role entity.
4. On the page you enter, select **OIDC** as the IdP type.
5. Select an IdP you created i.e **zeeve_oauth**.
6. Set conditions for the role: **oidc:aud**: `505b1146-13fe-4df6-927a-ca57321786fd` **oidc:sub**: Delete this.



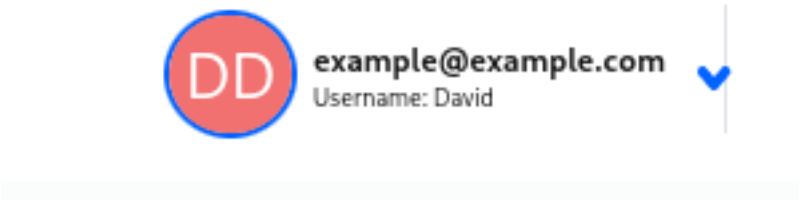
7. Click Next.
8. On the page you enter, associate the **QCloudResourceFullAccess** and the **QCloudFinanceFullAccess** policy with the role and click **Next**.



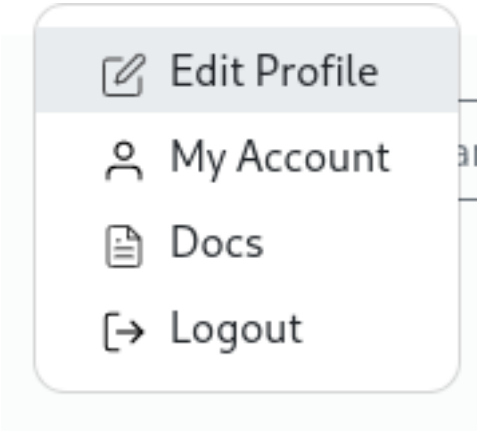
9. On the review page, enter the role name and role description (optional) and click **Complete** to save the above configurations.

7.4.3 Authorizing Cloud account

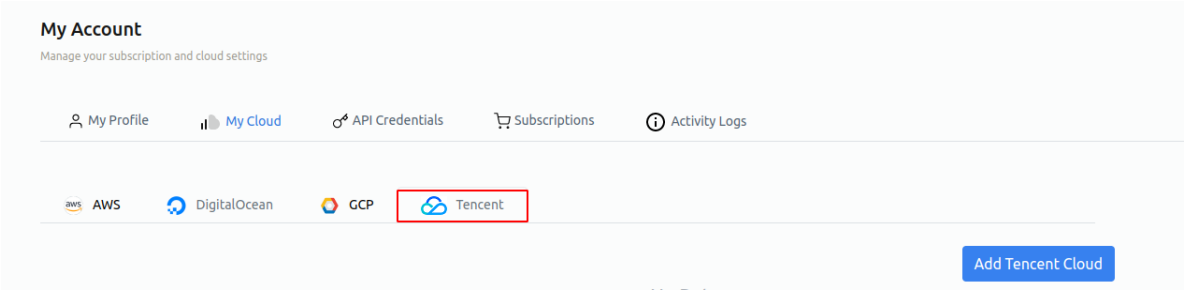
1. Hover on **profile**



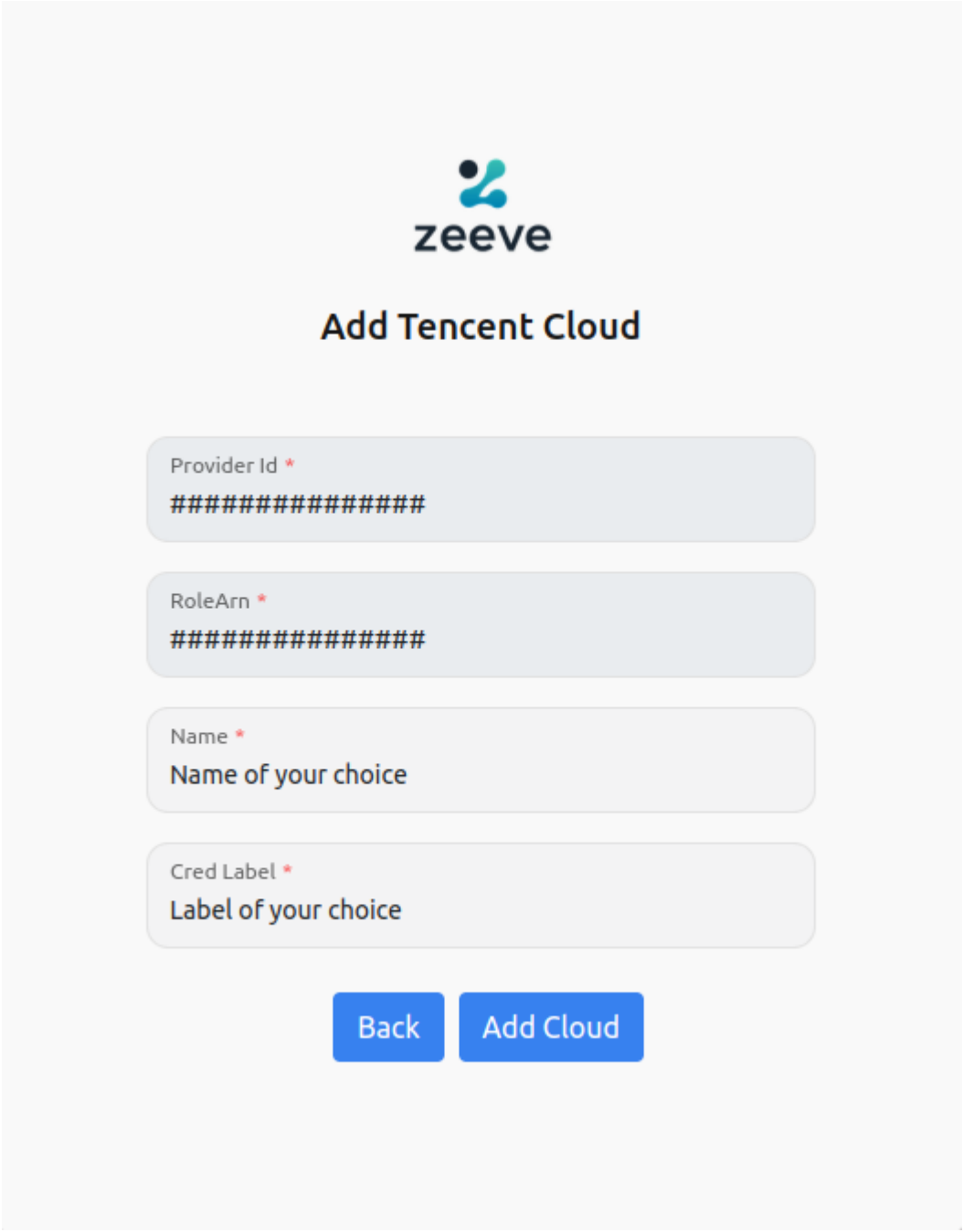
2. Click on **My Account**



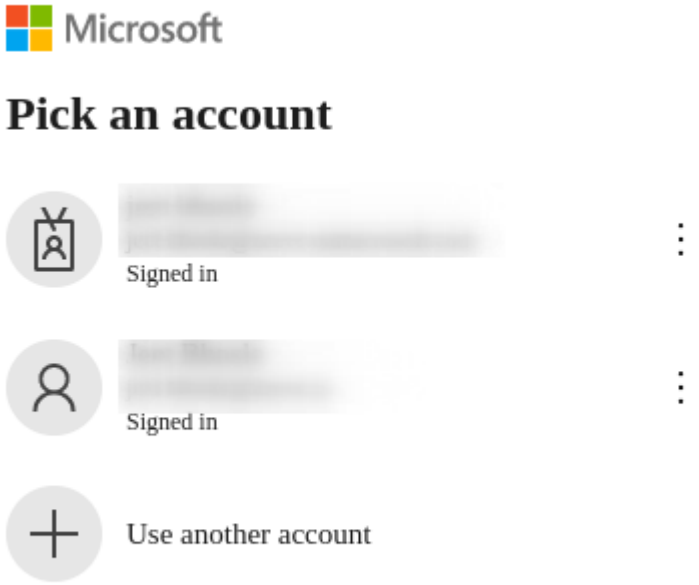
3. Click on **My Cloud**.
4. Click on **Tencent** and then click on **Add Tencent Cloud**.



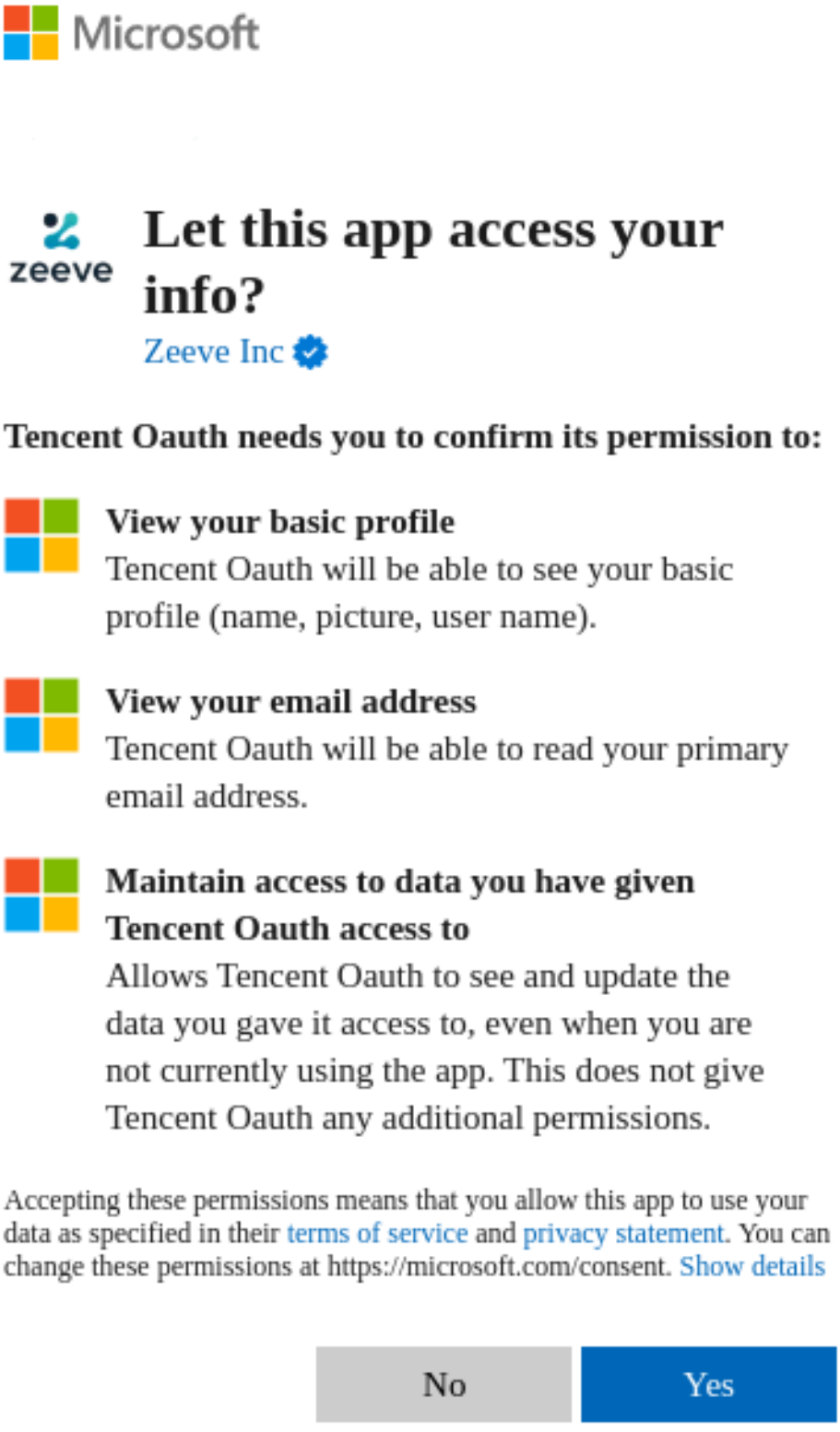
5. Add the **ProviderId** and **RoleARN** that you have created in the previous steps.



6. Login through any of your microsoft personal account, work account or you can add an account.



7. This will lead you to a consent screen where you will need to **Accept** the Terms & Conditions to allow Zeeve to use your credentials.


Microsoft
zeeve
Let this app access your info?
Zeeve Inc
Tencent Oauth needs you to confirm its permission to:
View your basic profile
Tencent Oauth will be able to see your basic profile (name, picture, user name).
View your email address
Tencent Oauth will be able to read your primary email address.
Maintain access to data you have given Tencent Oauth access to
Allows Tencent Oauth to see and update the data you gave it access to, even when you are not currently using the app. This does not give Tencent Oauth any additional permissions.
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://microsoft.com/consent. Show details
No
Yes

CHAPTER

EIGHT

DESCRIPTION: DISCOVER HOW TO MANAGE SUBSCRIPTIONS ON ZEEVE'S PLATFORM. LEARN HOW TO CREATE, UPDATE AND CANCEL SUBSCRIPTIONS, AS WELL AS HOW TO MANAGE THE BILLING, PAYMENTS AND USAGE OF THE RESOURCES IN OUR PLATFORM.

CHAPTER

NINE

MANAGE YOUR SUBSCRIPTIONS

This section talks about how one can purcahse a subscription of a service. It will also guide how one can view manage their subscriptions.

1. *Purchase Subscriptions*.
2. *View Subscriptions*.
3. *Edit Subscriptions*.
4. *Delete Subscriptions*.

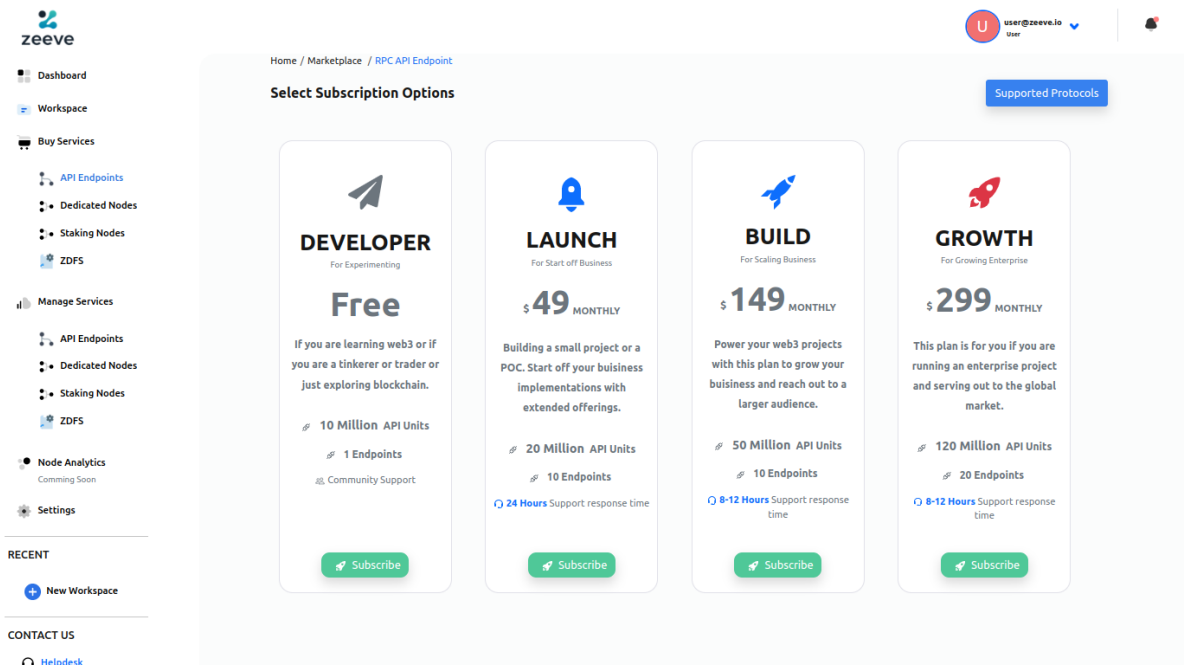
9.1 Purchase Subscriptions

This section is going to guide you how to purchase a subscription of your choice.

1. ***API Endpoint***
2. ***Staking Nodes***
3. ***Full Nodes***

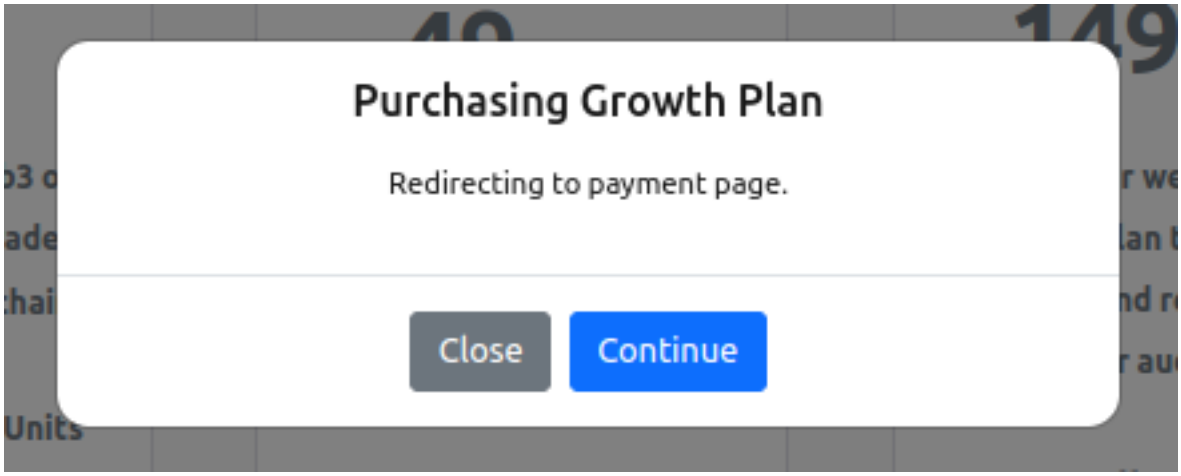
9.1.1 API Endpoint

1. Click on the **API Endpoint** under **Buy Services** from the left pane and you will be redirected to the purchase page.

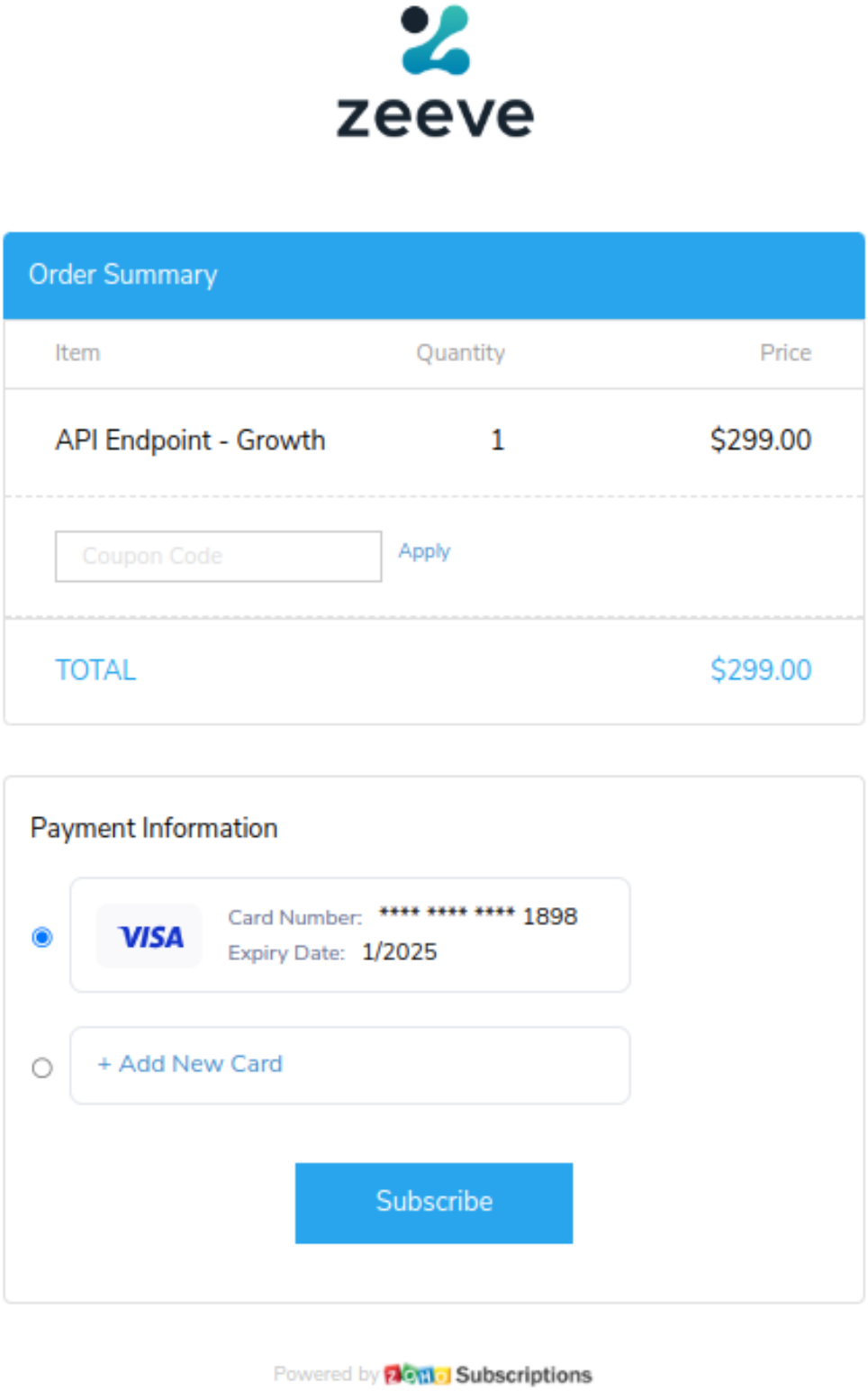


2. Click on the **Supported Protocols** button on top right and you will be able to see the protocols and their network types that Zeeve offers for creating API endpoints.

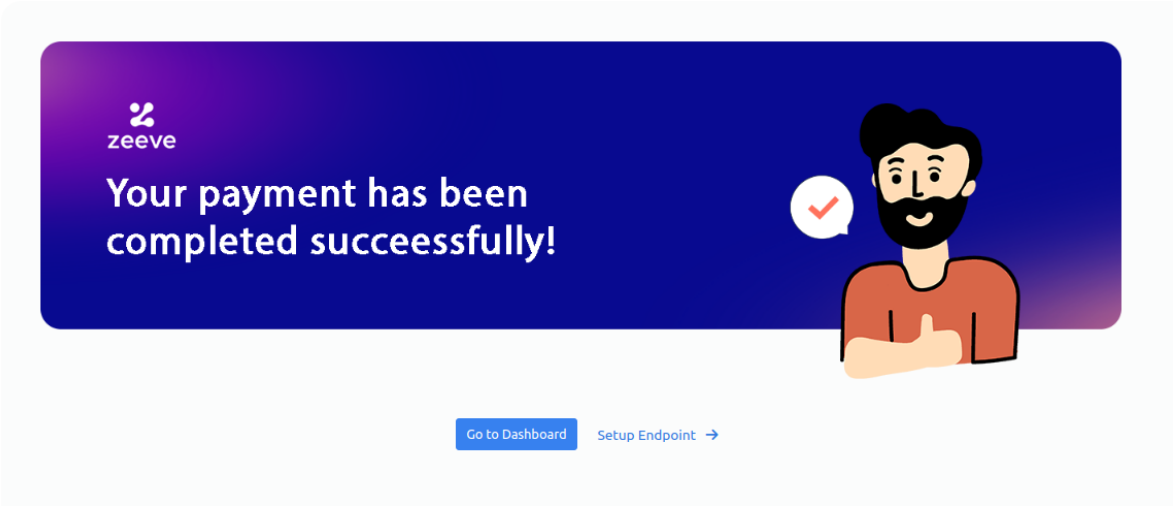
3. Choose the plan of your choice and click on the **Subscribe** button under the plan of choice. A pop up window will open which will show the plan which you are subscribing. Click on **Continue** button to proceed with your purchase.



4. You will be redirected to the payment page to complete your purchase. Fill the card details and click **Subscribe**.


zeeve
Order Summary
Item
Quantity
Price
API Endpoint - Growth
1
$299.00
Coupon Code
Apply
TOTAL
$299.00
Payment Information
VISA
Card Number: **** **** **** 1898
Expiry Date: 1/2025
+ Add New Card
Subscribe
Powered by Zoho Subscriptions

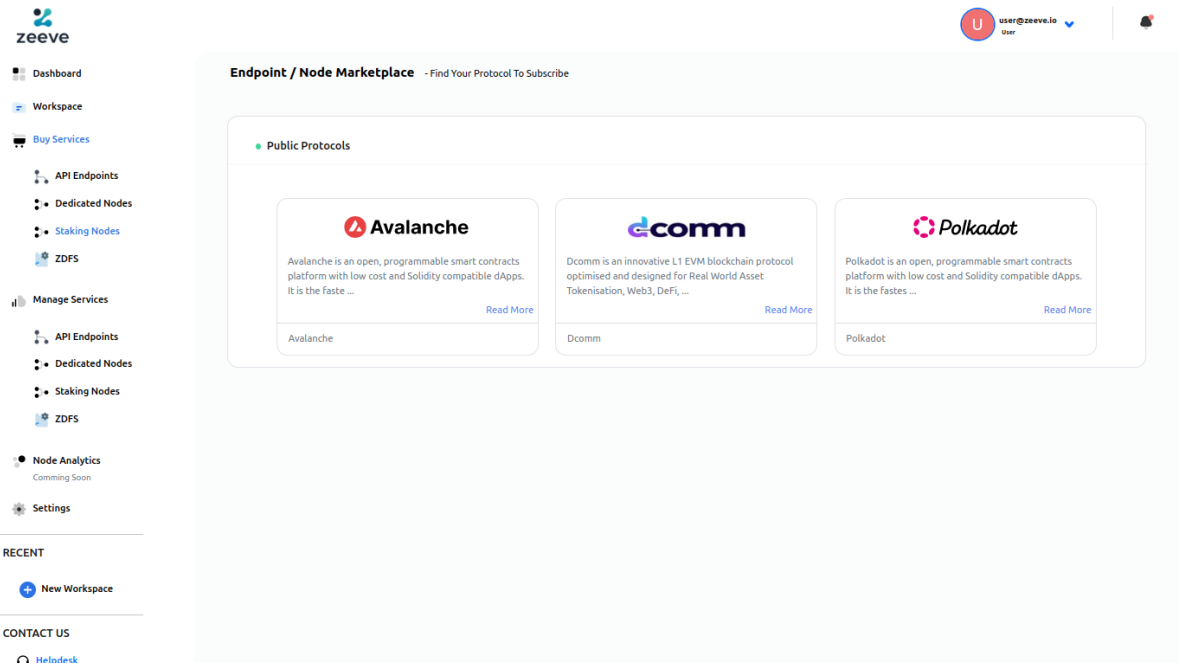
5. After successful payment you will be redirected to success page which ensures successful purchase of your subscription.



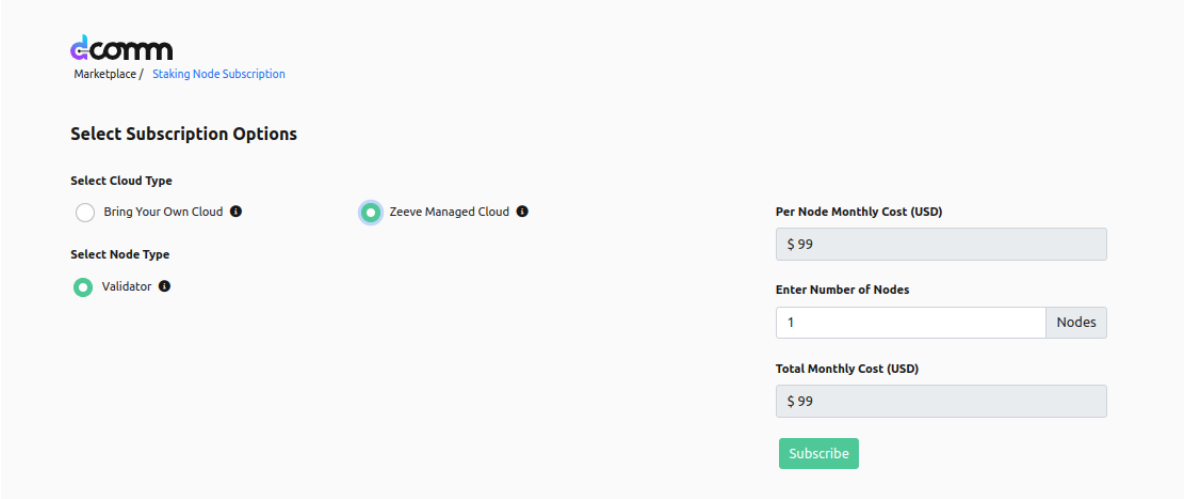
NOTE: Developer Plan is a free plan and can only be purchased once.

9.1.2 Staking Nodes

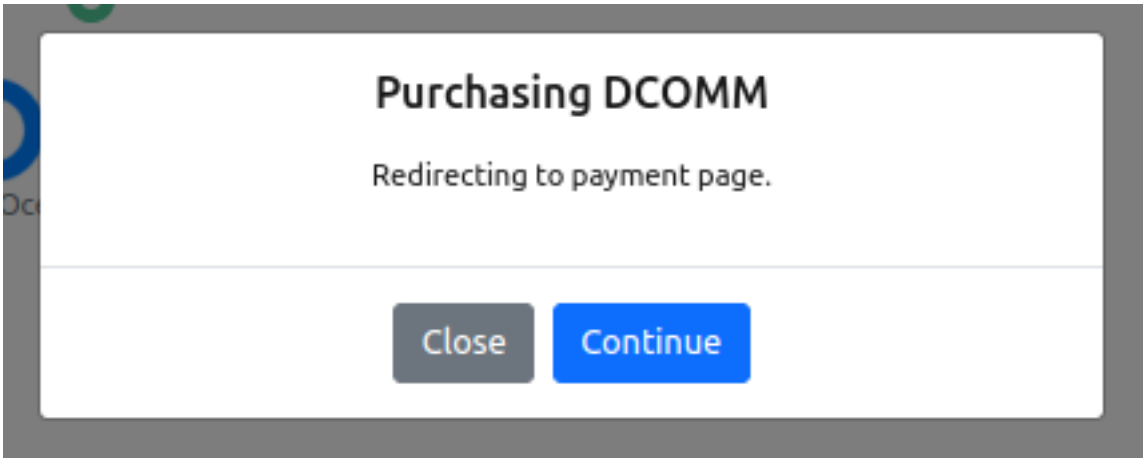
1. Click on the **Staking Nodes** under **Buy Services** from the left pane and you will be able to see the list of protocols that Zeeve offers for staking.



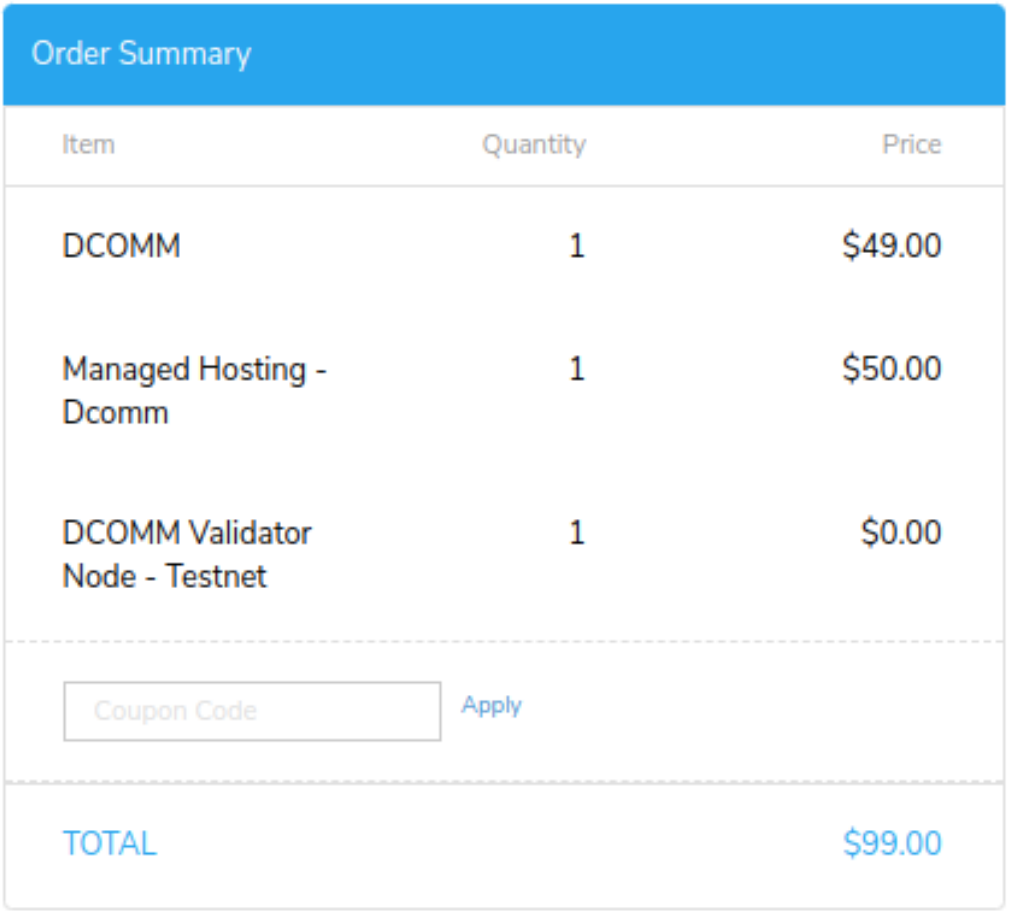
2. Click on the protocol card that you want to subscribe. You will be then redirected to the purchase page.



3. Select the cloud option from ***Bring Your Own Cloud*** or ***Zeeve Managed Cloud*** and the number of nodes you want to purchase. Based on your selection the total amount for the subscription will be shown. Click on **Subscribe** button to continue. A pop up window will open, click on **Continue** to proceed with your purchase.

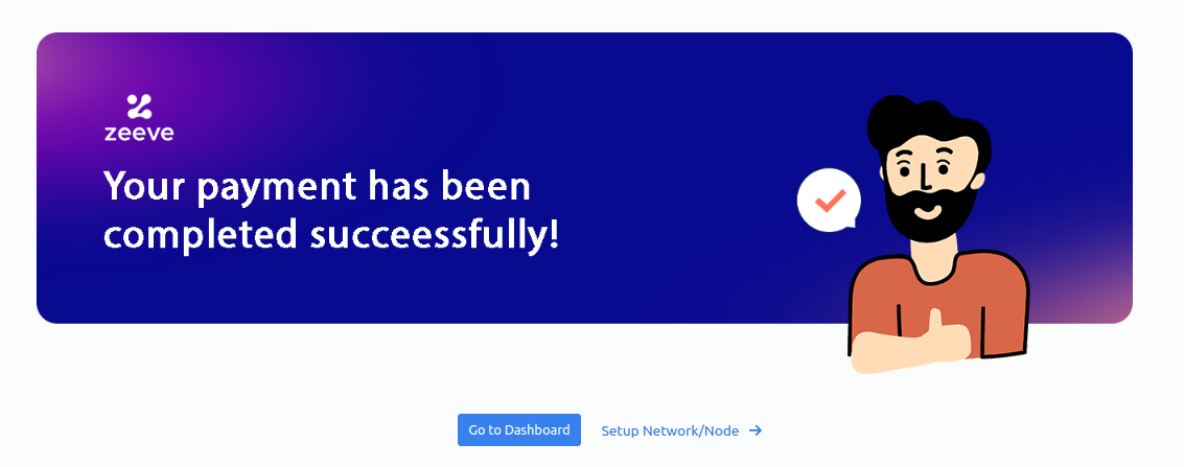


4. You will be redirected to the payment page to complete your purchase. Fill the card details and click **Subscribe**.



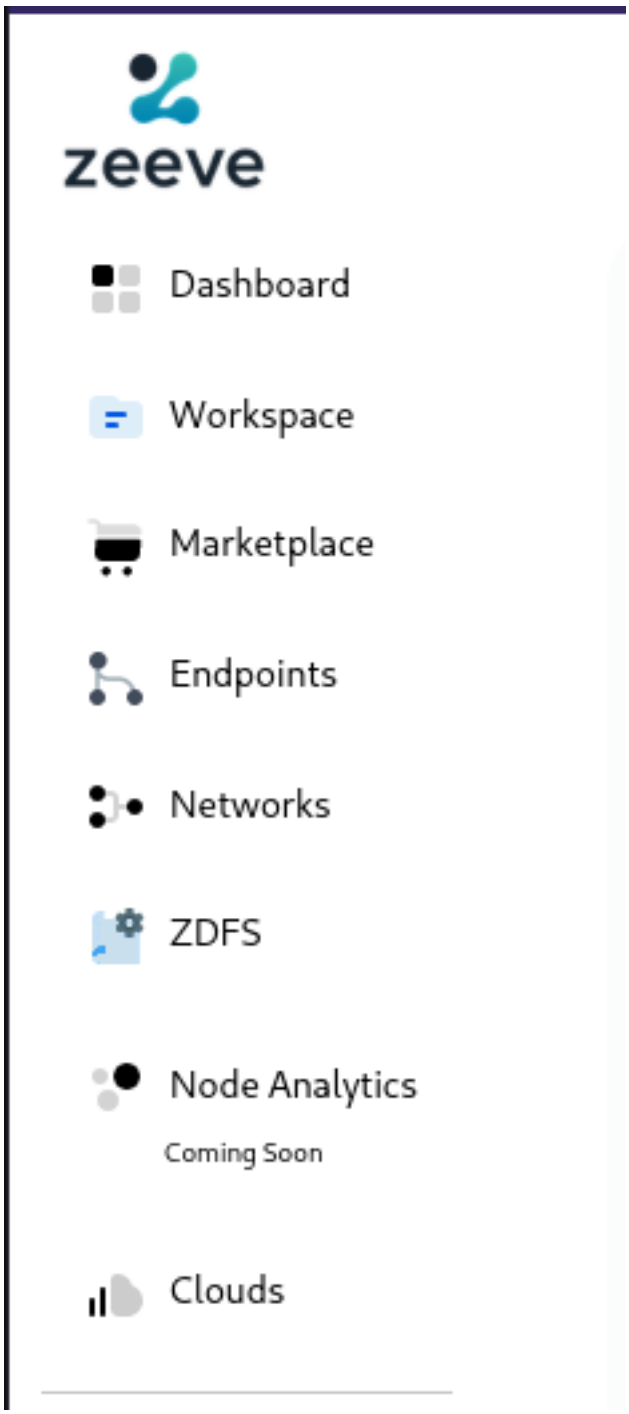
NOTE Prices on this page totally depends on the protocol and your selection of different options in previous step.

5. After successful payment you will be redirected to success page which ensures successful purchase of your subscription.

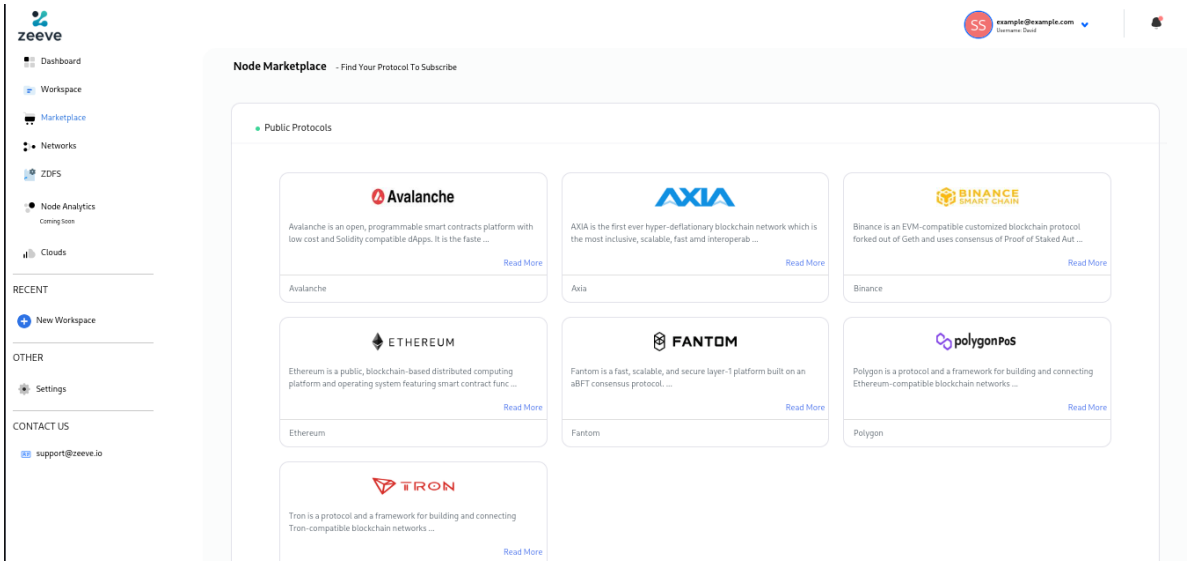


9.1.3 Full Nodes

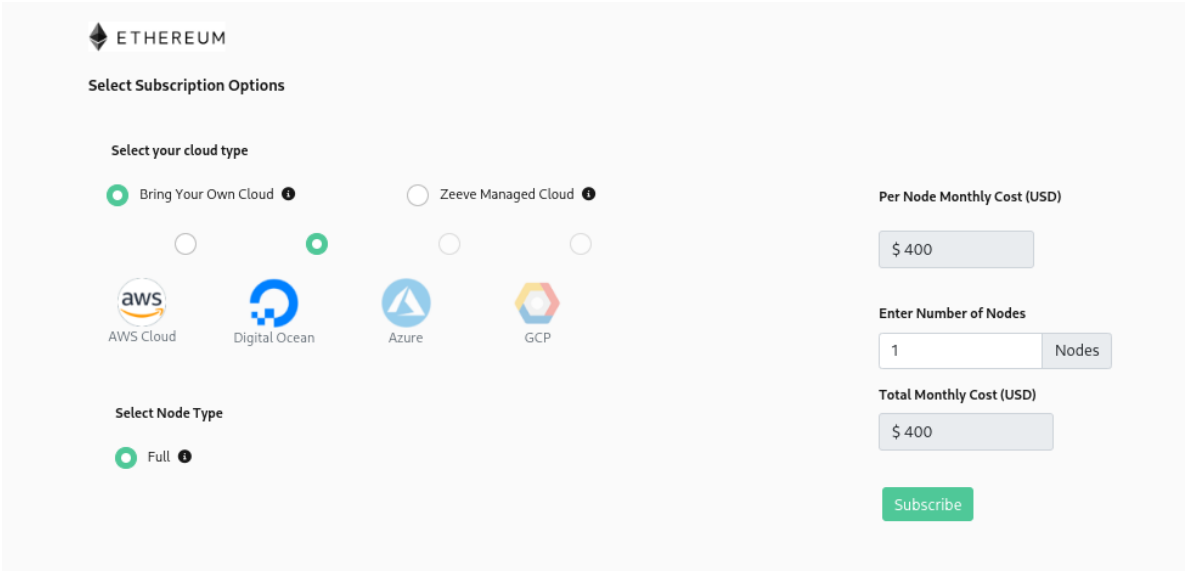
1. Click on **Marketplace** on the left side navigation bar.



2. You will be landed to zeeve's market place page. Select the protocol card of you choice to purchase subscription of the protocol.



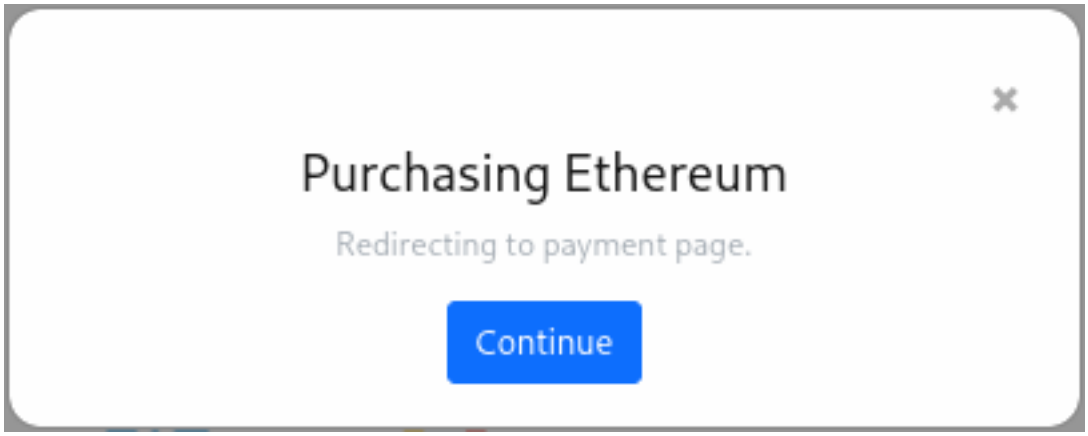
3. You will be redirected to purchase page of the protocol, which looks similar to the below image.



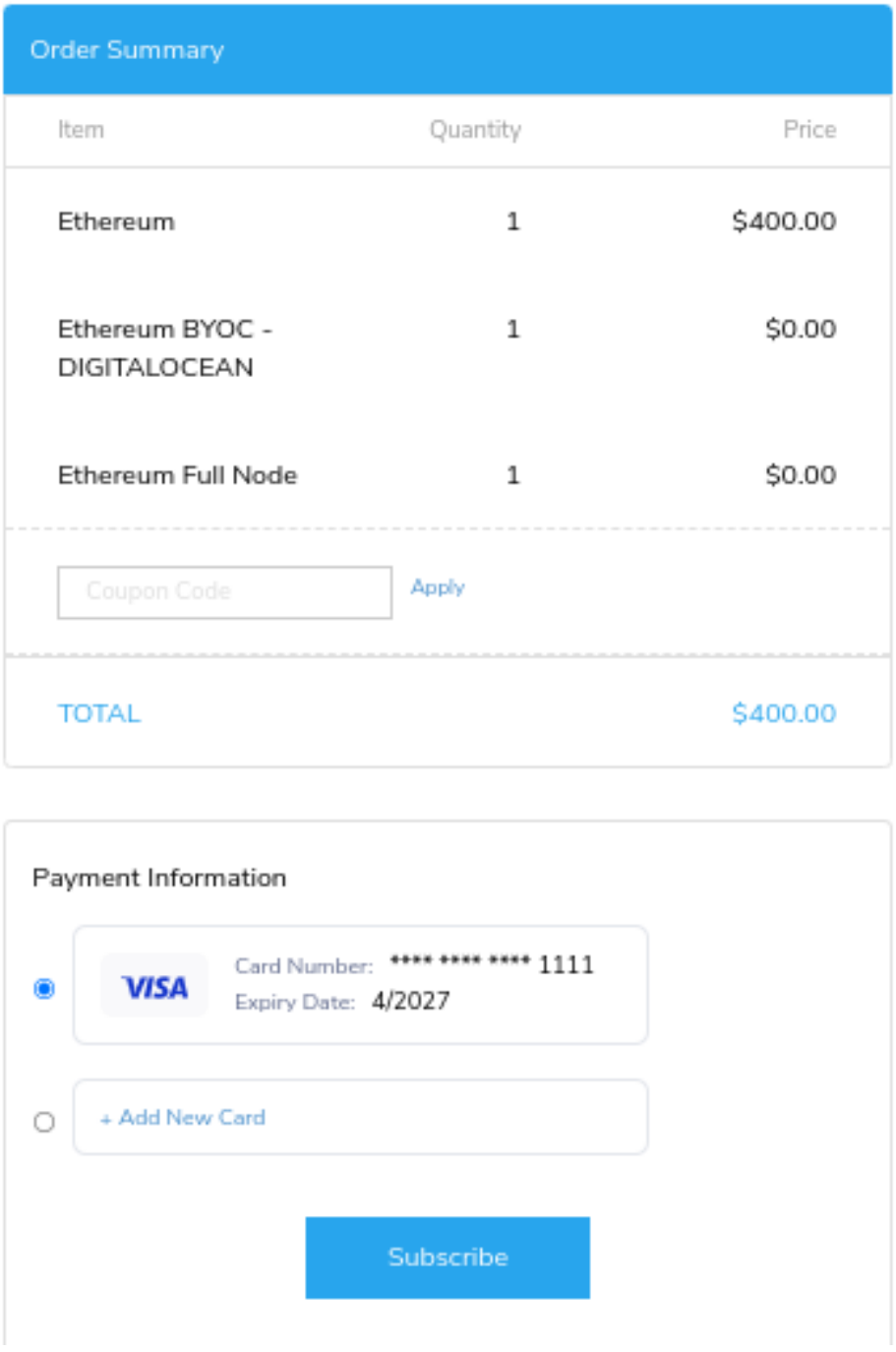
NOTE This page can be different for different protocls. Node types can vary from protocol to protocol

---

1. Select the number of nodes you want to purchase. You can also view the amount based on your selection of different options and number of nodes. After your selection, click on **subscribe** button. A pop up window will open similar to the below image. Click on **continue**.

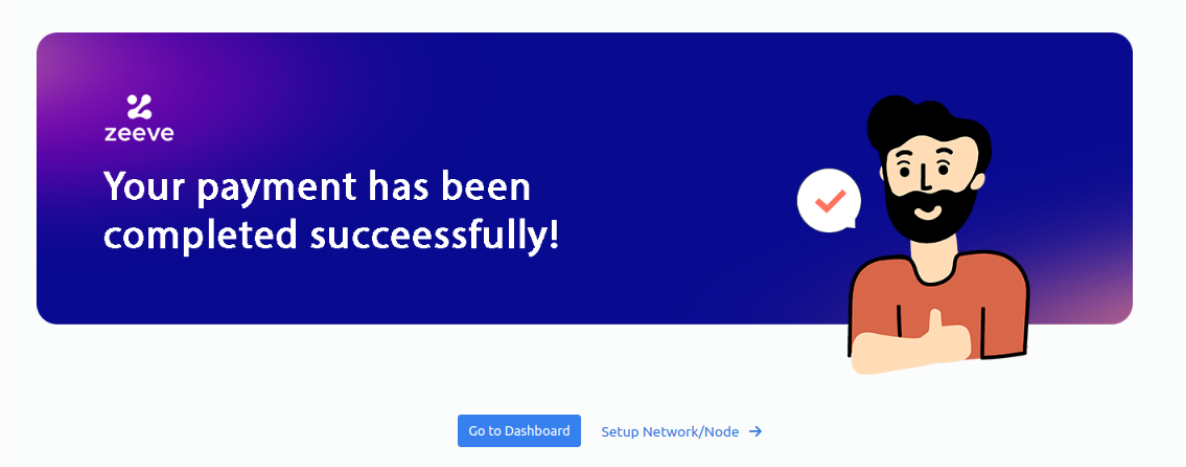


2. You will get redirect to the payment page to complete your purchase of subscription. Fill all the required details and click **subscribe**.



NOTE Prices on this page totally depends on protocol and your selection of different options in previous step.

1. After successful payment you will be redirected to payment success page which ensures successful purchase of your subscription.

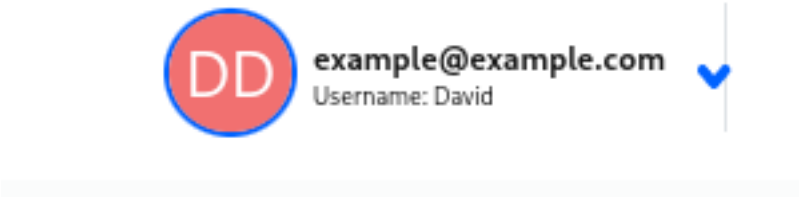


9.2 View Subscriptions

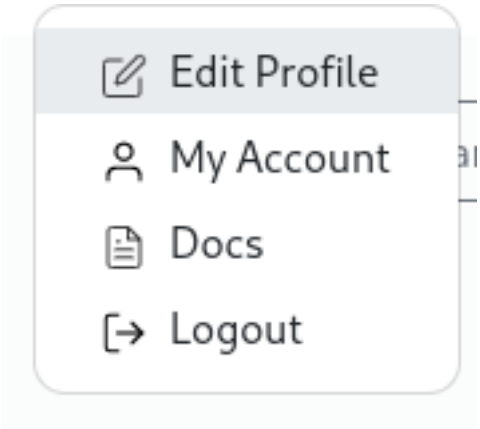
This section is going to guide you how you can view the subscriptions you purchased.

To view the subscriptions you have purchased follow the steps given below -

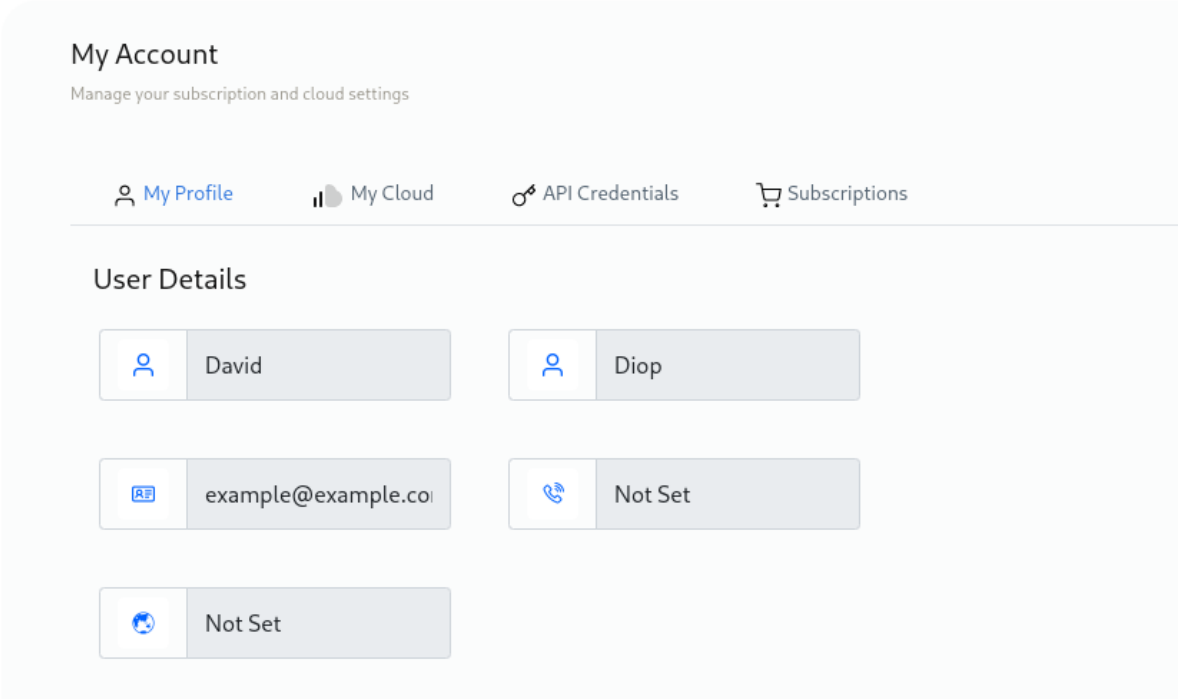
1. Hover on **profile**



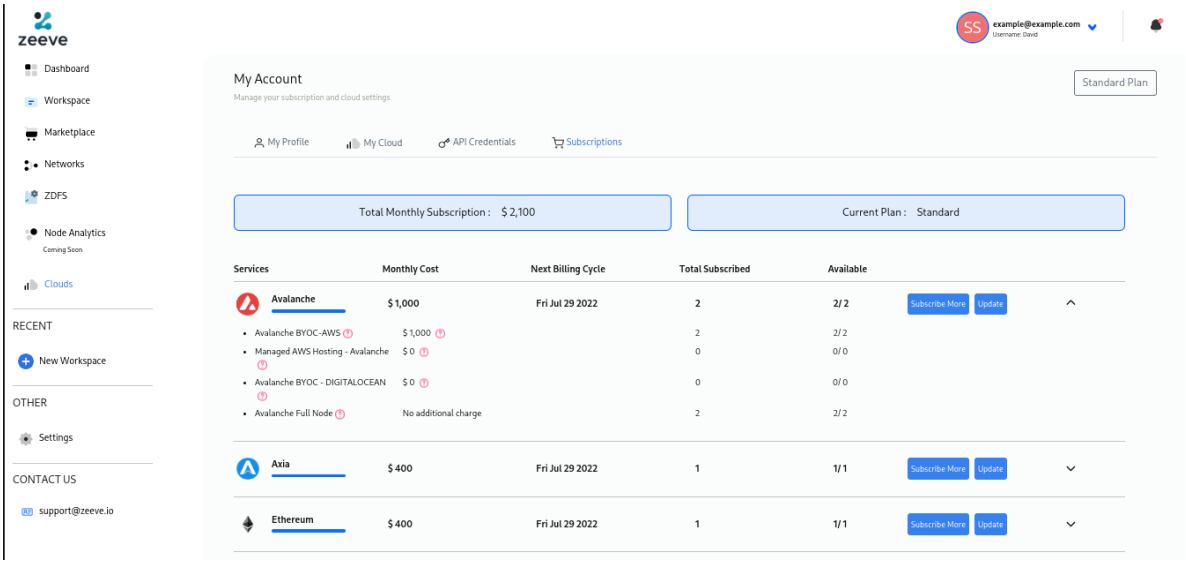
2. Click on **Edit Profile**



3. Click on **subscription** on the right most side.



4. In the subscription section you will be able to view the list of subscriptions, you currently have.



- In this section you can have a look to detailed information of your subscriptions.
  - **Monthly cost:** This tells you about the amount of each of the subscriptions.
  - **Next Billing Cycle:** It provides the next renewal date of a subscription.
  - **Total subscribed:** This talks about the quantity of each of the item you have purchased with the subscription.
  - **Available:** This tells you about available quantity of each of the item associated with the subscription. This number will increase or decrease according to the consumption of the item as you delete a network/node or create a network/adding a node respectively.This will help you to keep track of the consumption of each of the items, so that you can update your subscription as your needs.

9.3 Edit Subscriptions

This section talks about how to edit the subscriptions you purchased. You can choose to add more items to your subscription or you can also choose to decrease the quantity of the already purchased items.

1. *Increase items*.
2. *Decrease items*.

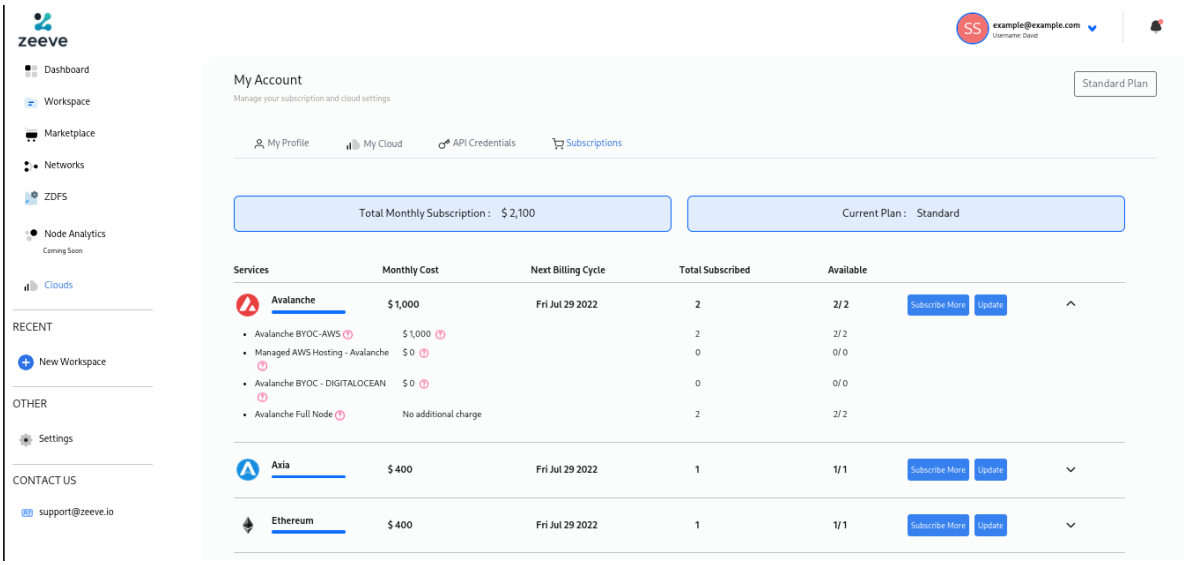
9.3.1 Increase items

You can increase items of your subscriptions in two ways. Either you can choose to add new items to the subscriotion or you can choose to increase the quantity of alredy purchased items with the subscription.

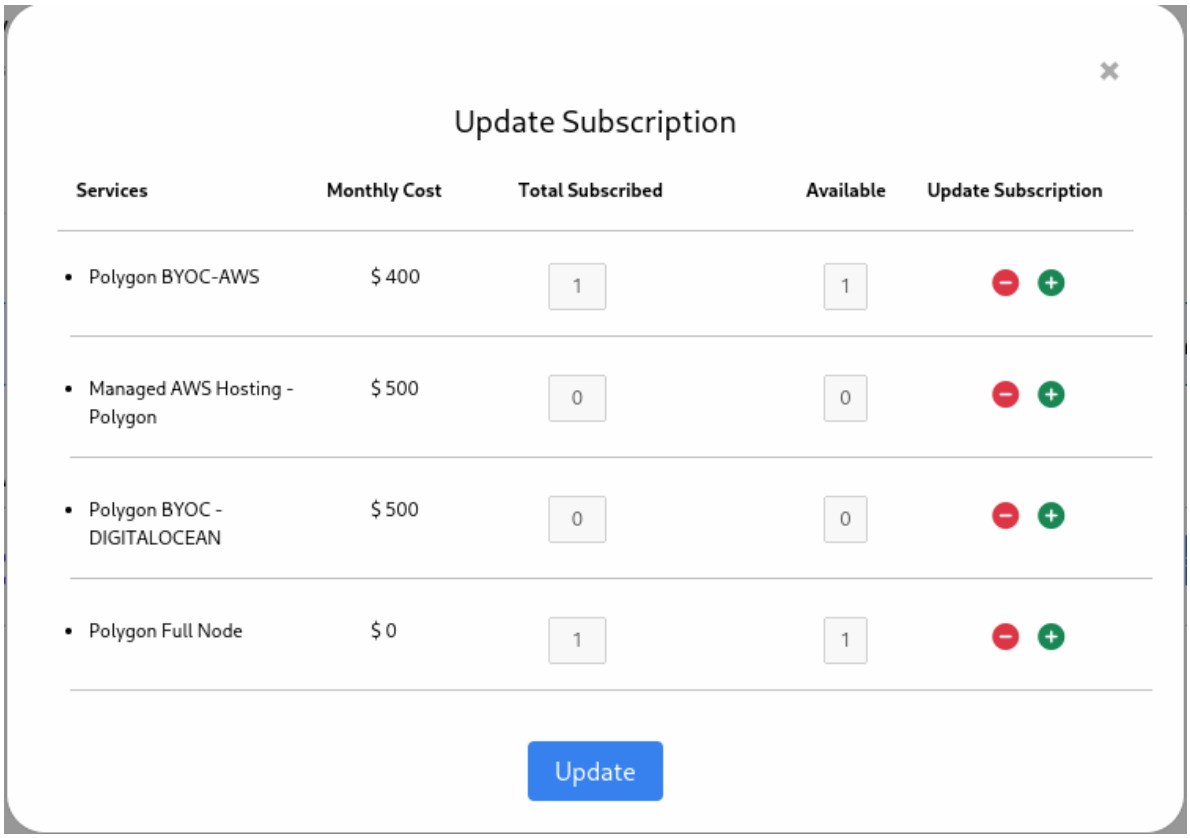
To add a new item click on the **subscribe more** button on the right most side of the subscription. Which will redirect you to the purcahse page of the selected product.

To increase the quantify of alredy purchased item follow the steps mentioned below:-

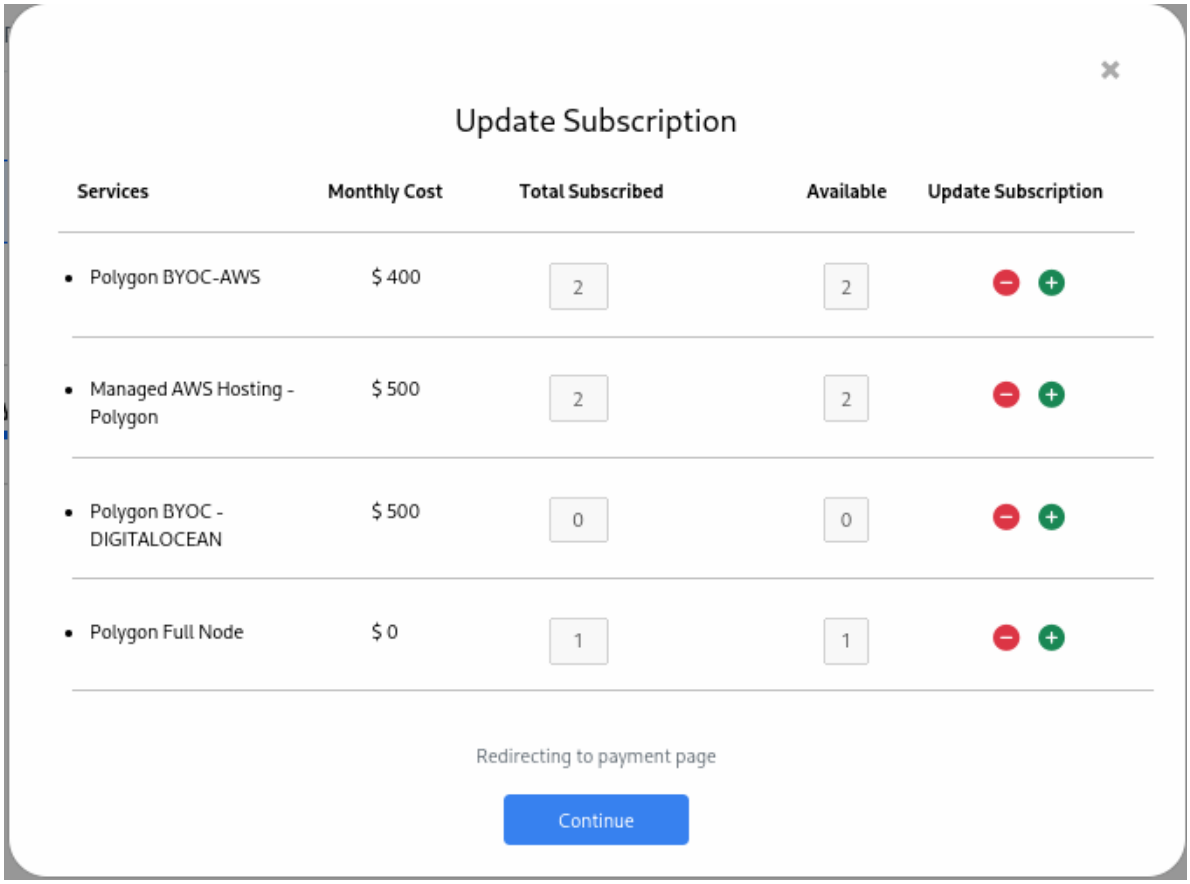
1. Go to ***subscriptions*** section.
2. Click on the **update** button on the right most side of the subscription.



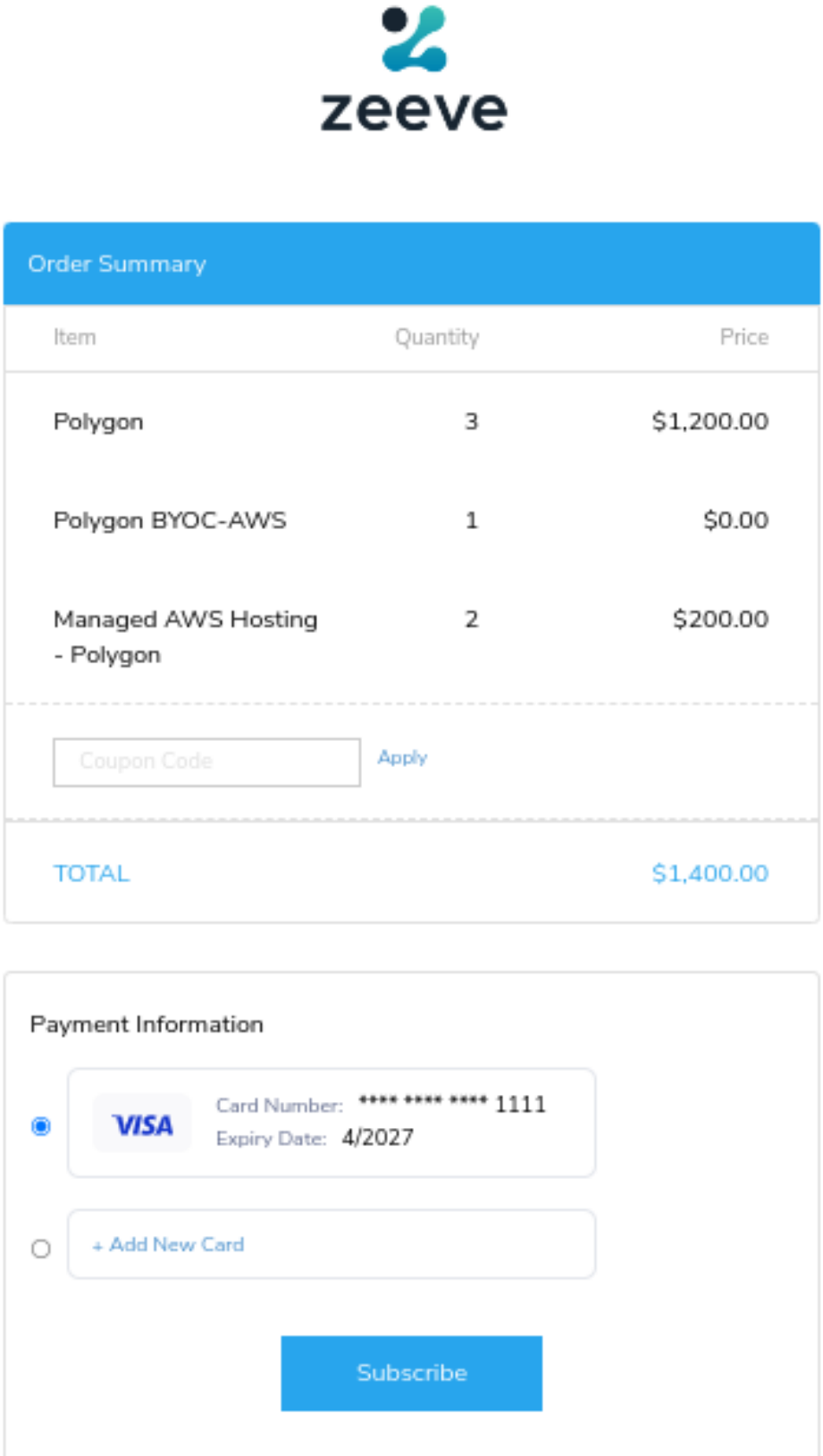
3. A pop up window will be opened similar to the image provided below.



4. Increase the quantity of the item of your choice by clicking the **green** button associated to it and click on **update** button. A **continue** button will apper on the pop up window.



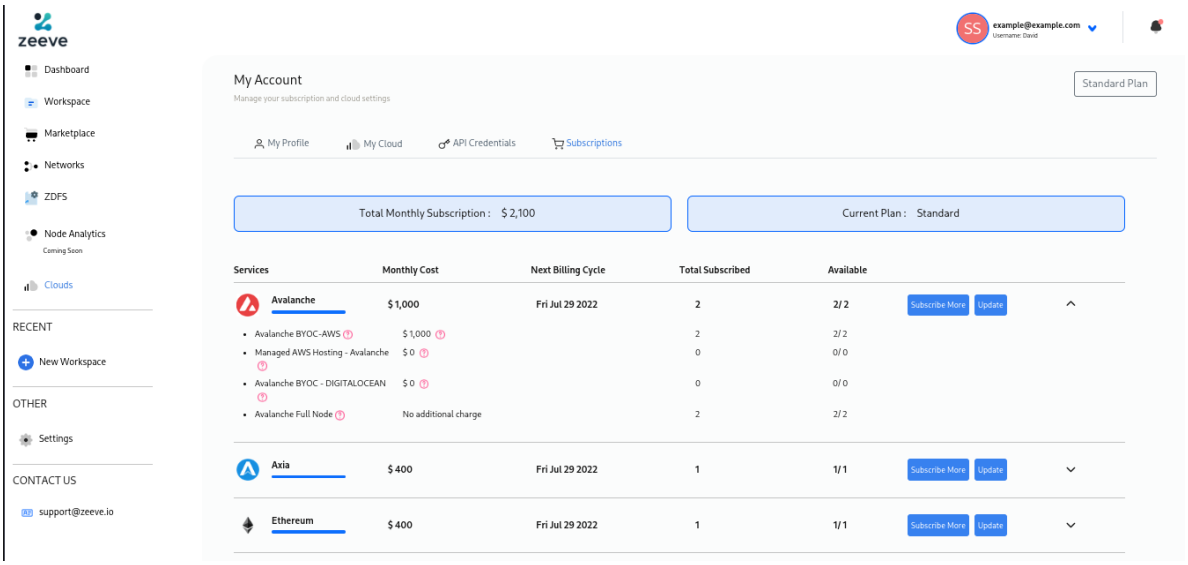
5. Clicking on the button will redirect you to the payment page, where you can view the items you have just added. Click on the **subscribe** button to complete the process of updation of your subscription.



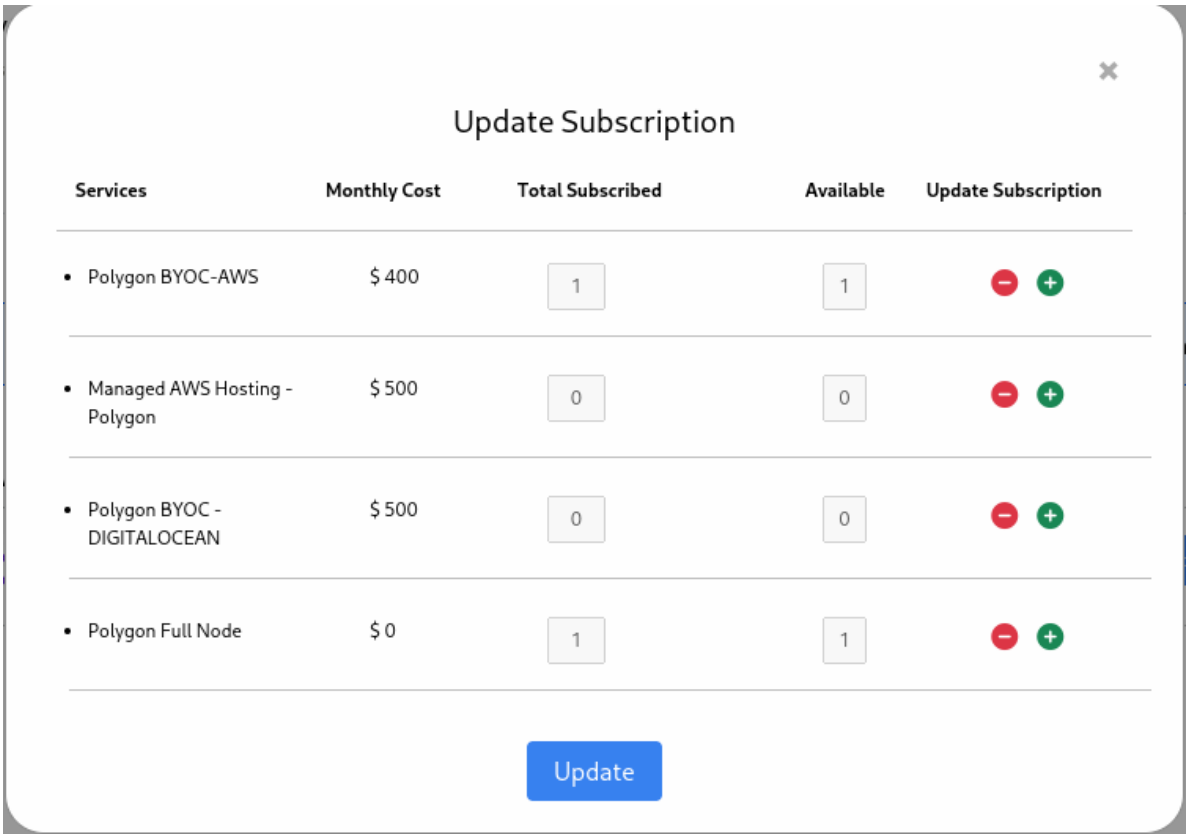
6. After successful payment of the item you can view your updated subscription in ***subscriptions*** section.

9.3.2 Decrease items

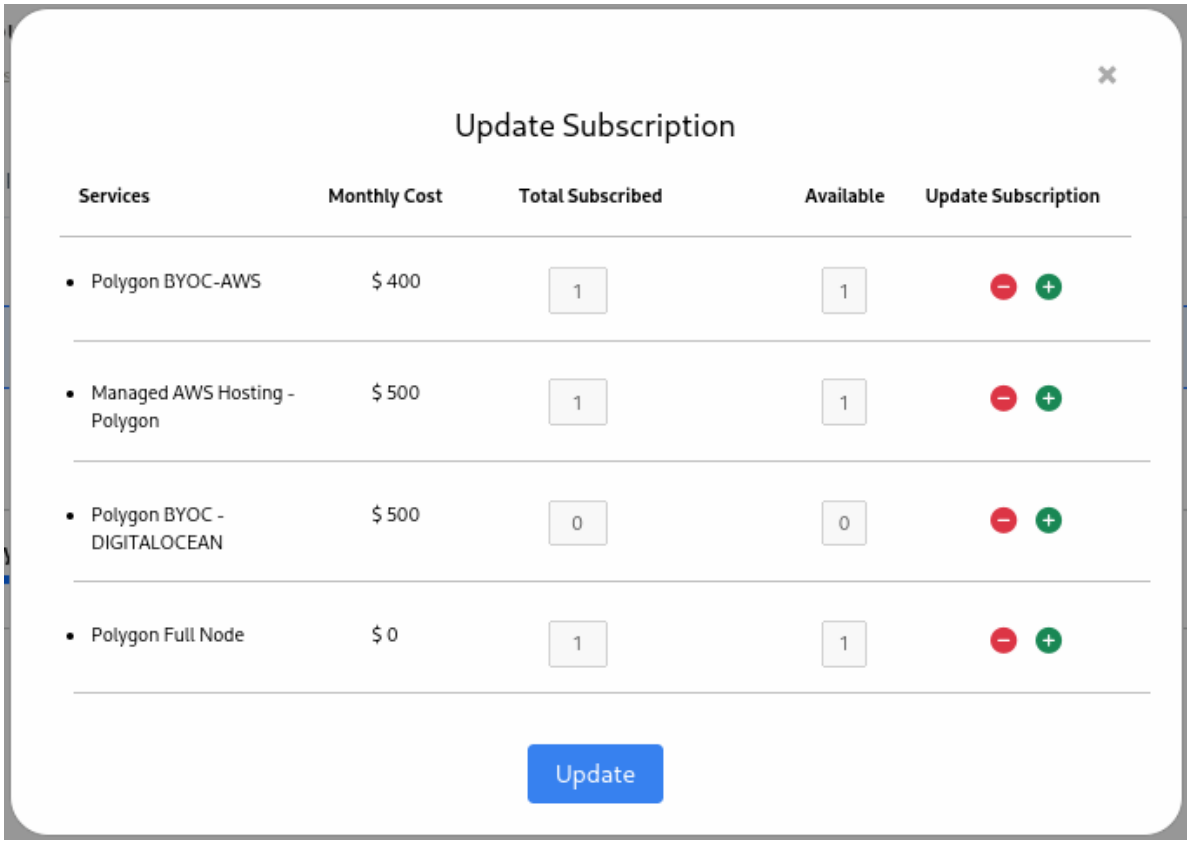
1. Go to ***subscriptions*** section.
2. Click on the **update** button on the right most side of the subscription.



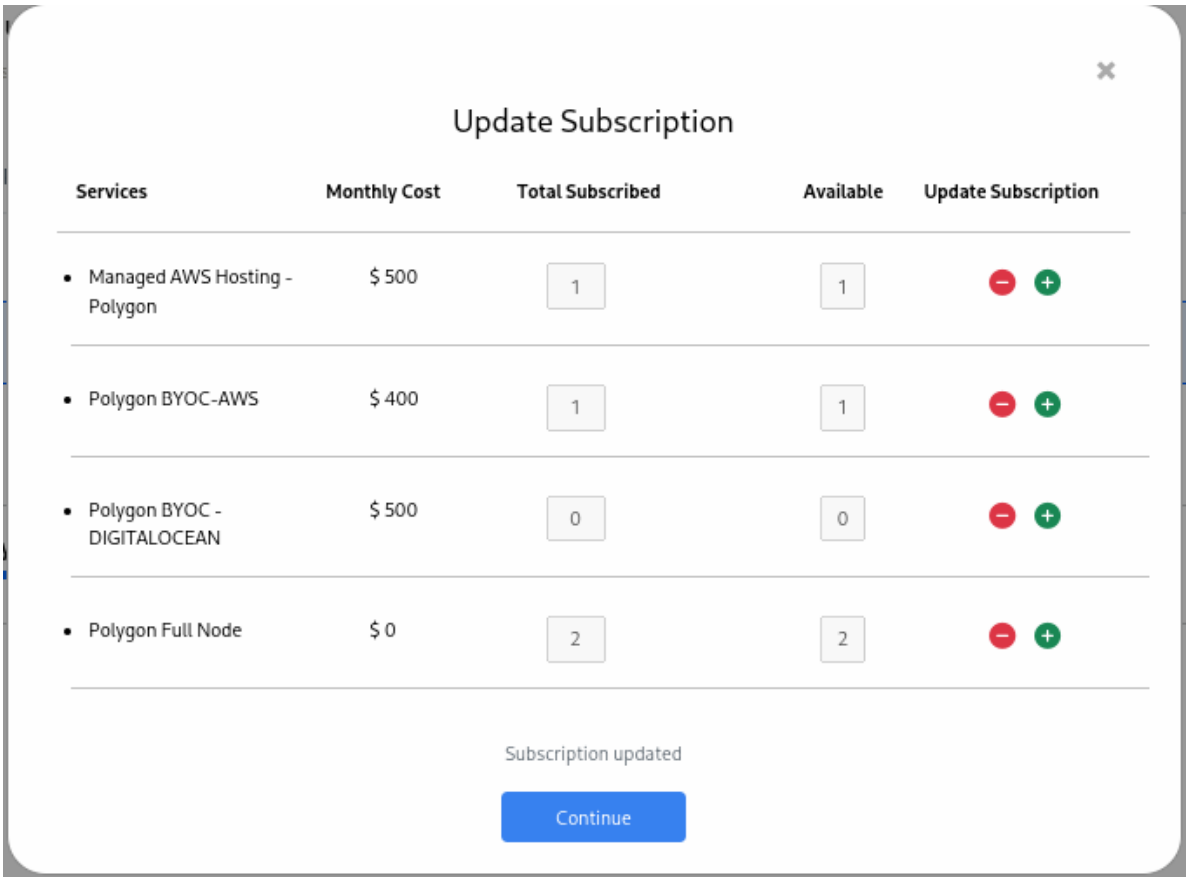
3. A pop up window will be opened similar to the image provided below.



4. Decrease the quantity of the item of your choice by clicking the **red** button associated to it and click on **update** button.



5. Clicking on the button will update your subscription. A **continue** button will apper on the pop up window.



6. After successful updation you can view your updated subscription in ***subscriptions*** section.

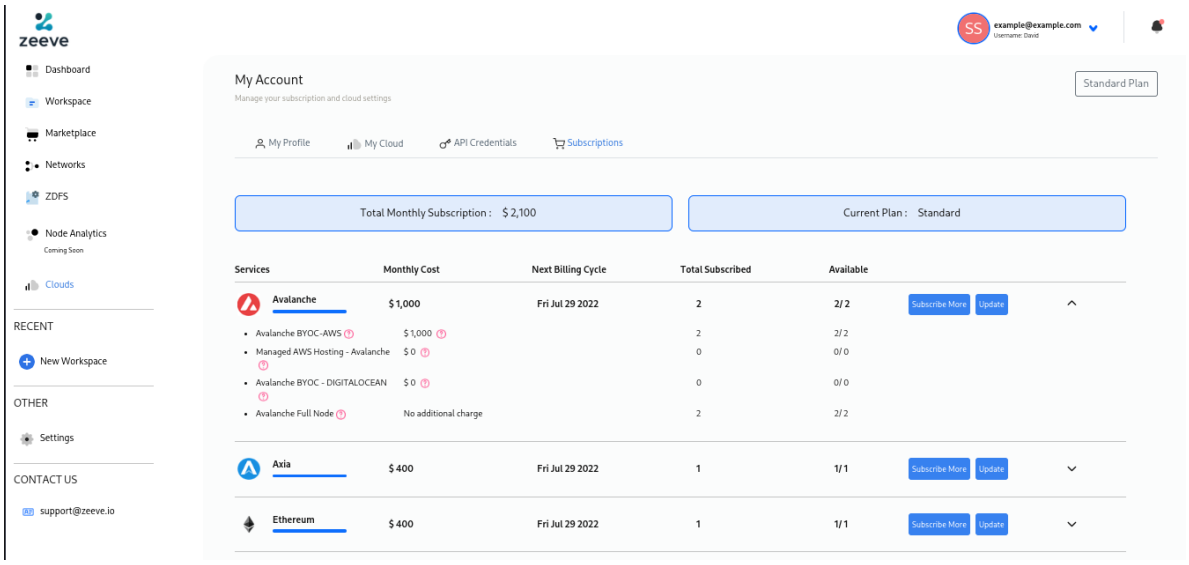
***INFO:** One can decrease the quantity of an item as much as available quantity of that item.*

9.4 Delete Subscriptions

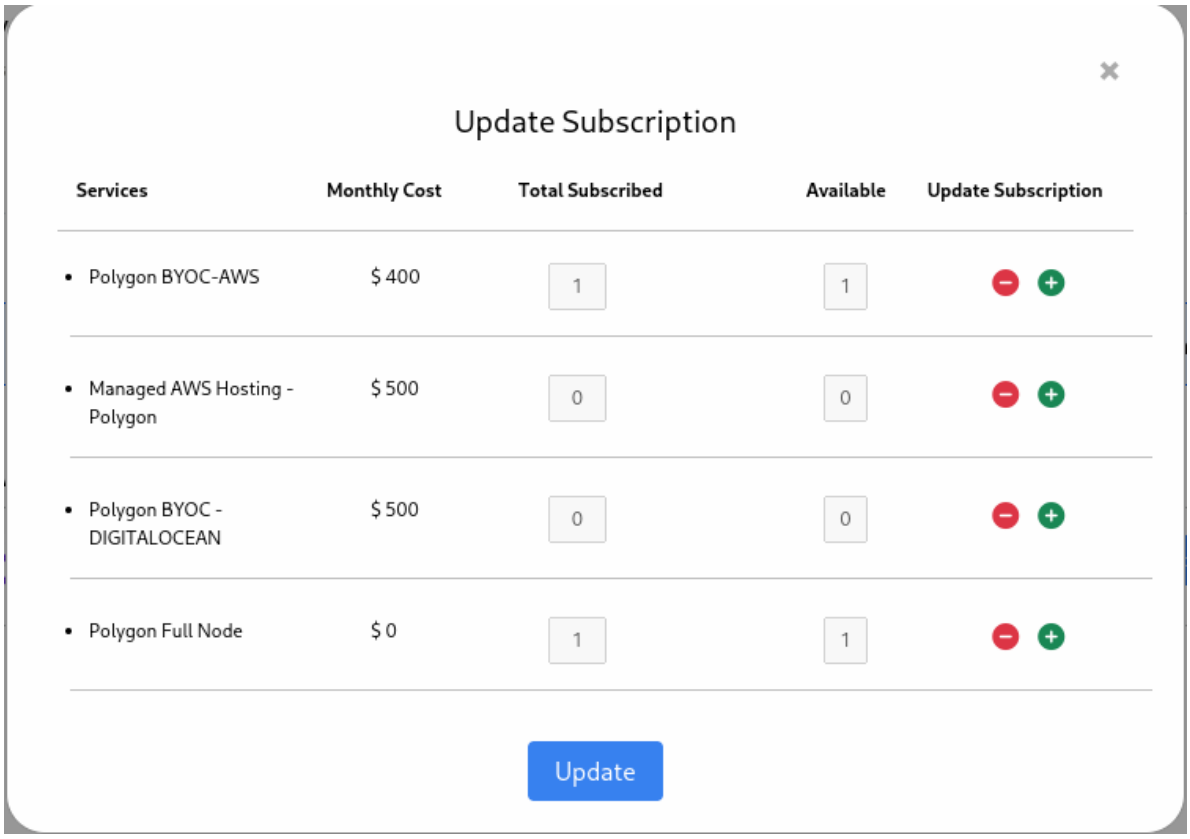
This section talks about how to delte the subscriptions you don't need.

- Before deleting a subsection, make sure that any of the items of the subscription is not consumed. Basically, make sure that the purchased quantity and available quantity is same for each of the item associated with the subscription. Otherwise you will not be able to delete a subscription.

1. Go to ***subscriptions*** section.
2. Click on the **update** button on the right most side of the subscription.



3. A pop up window will be opened similar to the image provided below.



4. Decrease the quantity of each of the item to 0 by clicking the **red** button associated to it and click on **update** button. This will delete your selected subscription.

CHAPTER

TEN

DESCRIPTION: FIND OUT HOW TO ACCESS AND USE THE API ENDPOINTS ON ZEEVE'S PLATFORM. OUR API DOCUMENTATION PROVIDES TECHNICAL DETAILS ON AUTHENTICATION, MAKING REQUESTS AND HANDLING RESPONSES FOR INTERACTING WITH OUR PLATFORM.

CHAPTER

ELEVEN

API ENDPOINTS

This page has detailed steps on how to

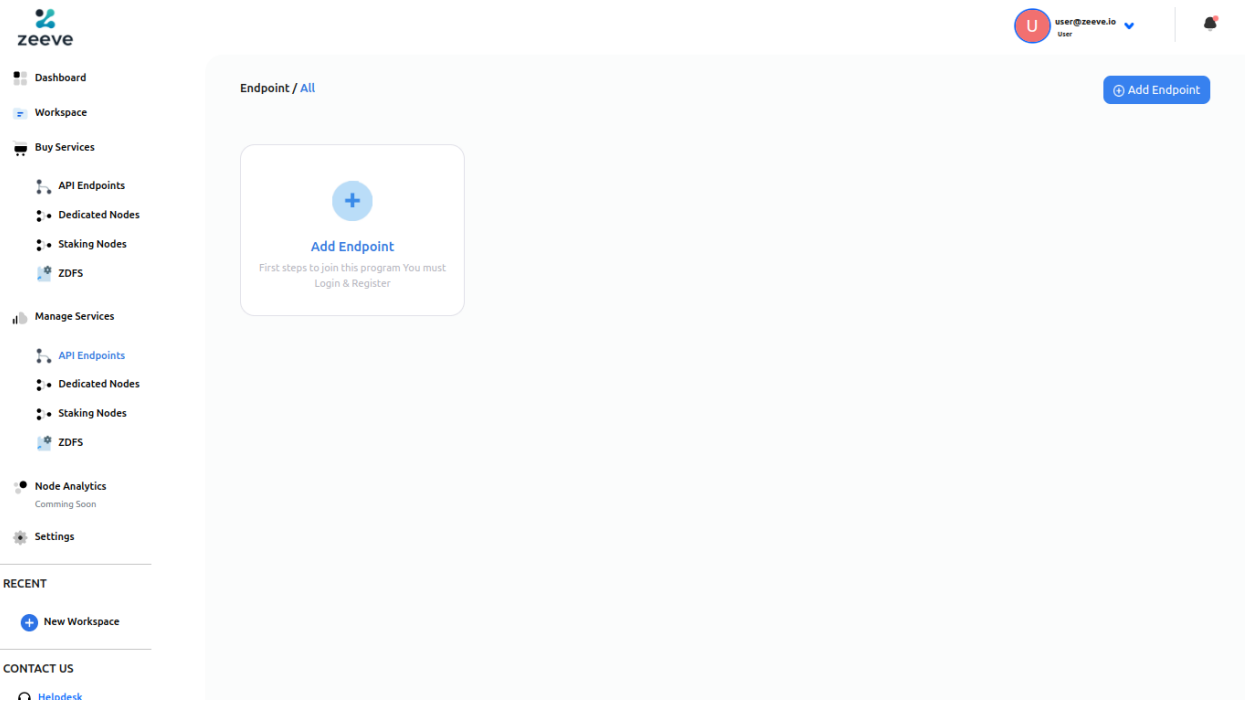
1. *Create an endpoint*
2. *Modify an endpoint*
    - Change name
    - Update security
3. *Delete an endpoint*

11.1 Create an endpoint

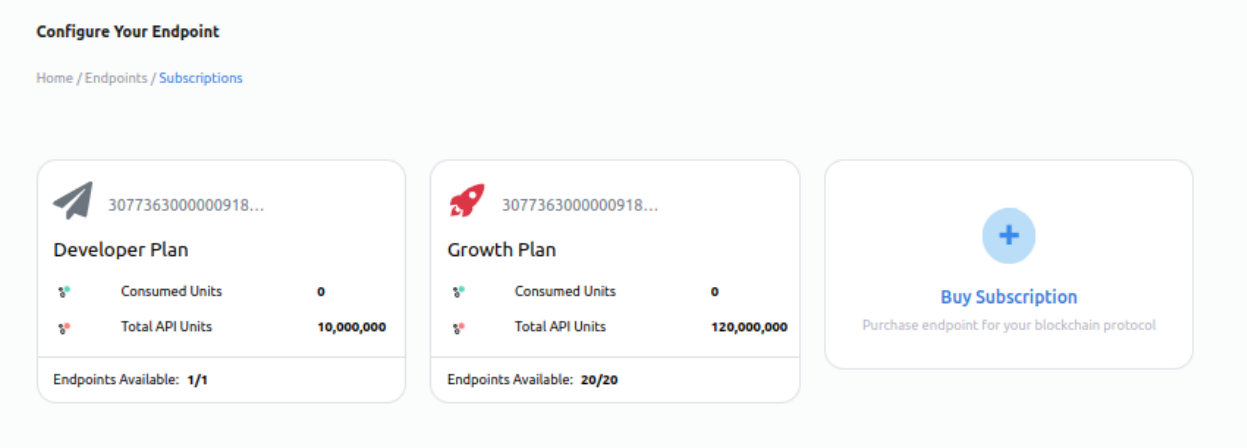
NOTE: *Purchase* a subscription plan before proceeding.

This section will provide you detailed steps for creating an API endpoint.

Visit the API Endpoints page by clicking on **API Endpoints** under **Manage Services** from the left side pane.



Click on **Add Endpoint** card or the button on top right corner. You will be able to see all the subscriptions you bought for the API endpoints.



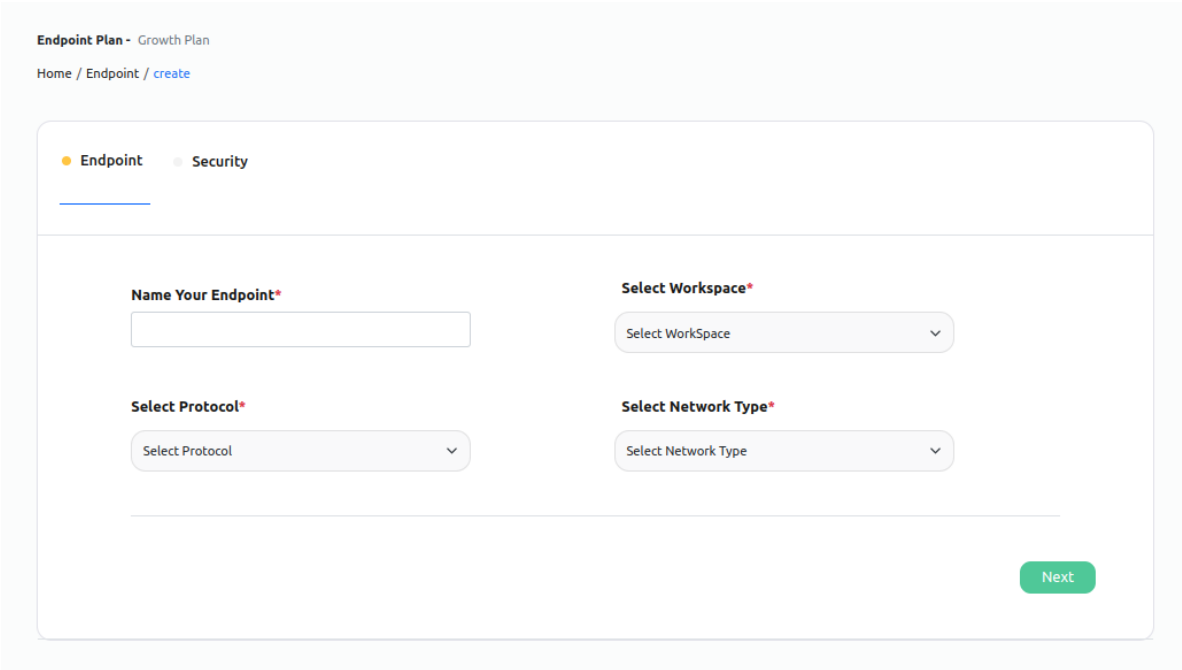
NOTE: These cards can be different based on your purchased subscriptions.

NOTE: The card will not be visible if the *API Units* or the *Endpoint* quota for that subscription has been exhausted.

Click on the card to choose the subscription in which you want add the endpoint. This will redirect you to the endpoint setup page.

1. **Endpoint Info**

   This step configures the basic and blockchain protocol settings for the endpoint.



   - **Endpoint Name:** The name of your endpoint.
   - **Workspace:** The workspace in which the endpoint will be added.
   - **Protocol:** The blockchain protocol for which the endpoint is created.

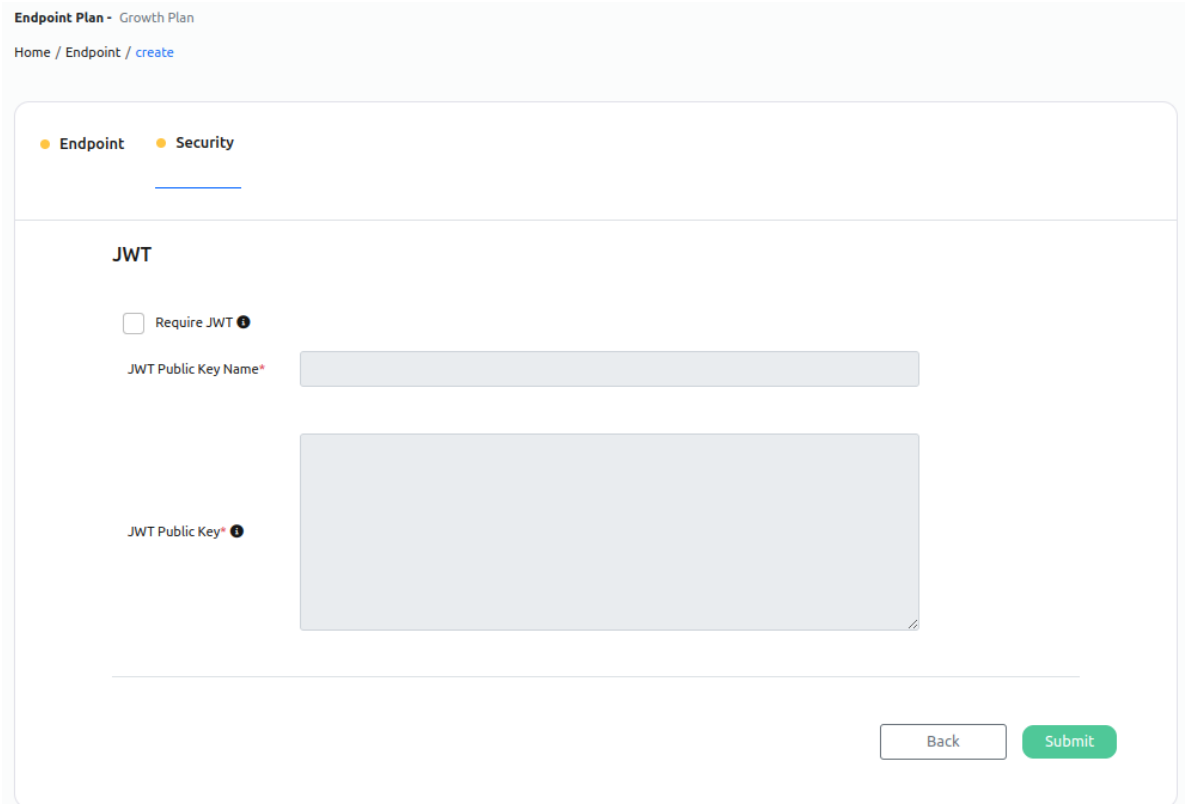
- **Network Type:** The network type of the selected blockchain protocol.

Proceed further by clicking on the **Next Step** button after providing all the details.

2. **Security Configuration**

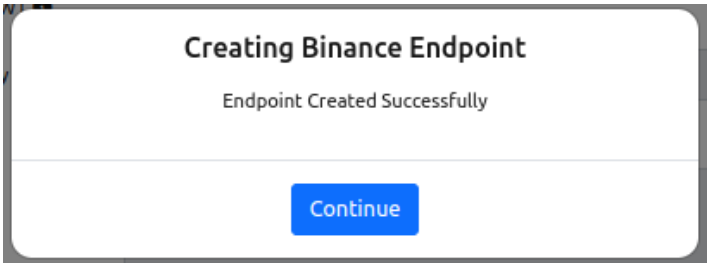
   NOTE: Adding security to the endpoint is **optional**.

   This step configures the security settings for the endpoint. An option to add a **JWT** in your API call to make your endpoint more secure.



- **Require JWT:** Enable this checkbox if you want to add a JWT security option.
- **Public Key Name:** The name associated to the *public key*.
- **Public Key:** The public key of a assymetric key-pair. Only keys generated using ***RSA*** and ***ECDSA*** algorithms are allowed.

On clicking the **Submit** button a pop-up window will open which ensures the successful creation of your endpoint.



On clicking the **Continue** button you will be redirected to the page where you can see the endpoint you created.

11.2 Modify Endpoint

This section will guide you on how you can modify an endpoint's

- Name
- Security

Visit the endpoint detail page of your endpoint (Manage Services > API Endpoints > Your Endpoint).

Click on the ***Edit*** icon in the top right corner.

- **Change Endpoint Name**

  After clicking the ***Edit*** icon the endpoint name field will become editable. Update the name as required.

  Then click the ***Save*** button beside the input field to save the name.

  A pop-up will confirm the successful updation of the endpoint name.

- **Modify Endpoint Security**

  After clicking the ***Edit*** icon the security section will become editable.

  Toggle the security toggle as per the requirement to turn on or off the JWT security option.

  Then click the ***Save*** button below to save the update in security.

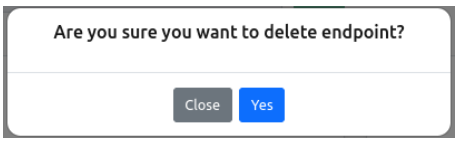
  A pop-up will confirm the successful updation of the endpoint security.

11.3 Delete Endpoint

Visit the endpoint detail page of your endpoint (Manage Services > API Endpoints > Your Endpoint).

Click on the ***Delete*** icon in the top right corner.

A confirmation window will open, click on the **Yes** button to delete the endpoint.



CHAPTER

TWELVE

DESCRIPTION: LEARN HOW TO BUILD YOUR FIRST NETWORK ON ZEEVE'S PLATFORM. OUR STEP-BY-STEP GUIDE PROVIDES DETAILED INSTRUCTIONS AND RESOURCES FOR SETTING UP AND CONFIGURING YOUR NETWORK INFRASTRUCTURE.

CHAPTER

THIRTEEN

HOW TO CREATE MY FIRST NETWORK?

Zeeve makes the process of blockchain network deployment from a long time consuming one to just a matter of few clicks whilst taking care of the most important bits.

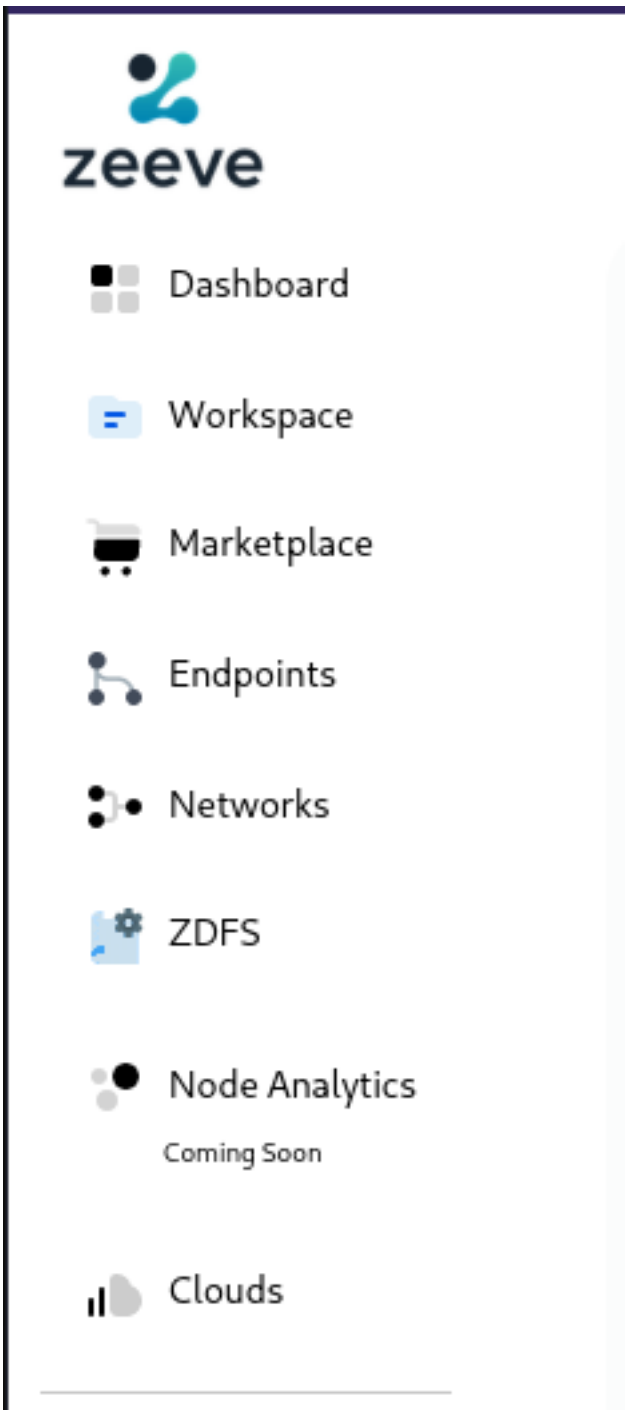
With a handful of steps using Zeeve, it has become so easy to create your own blockchain network. These networks can also be altered as per the need of the required deployment with the help of given protocol specific parameters that helps you align with your desired network performance.

So wondering upon how to begin? Just follow these easy steps:-

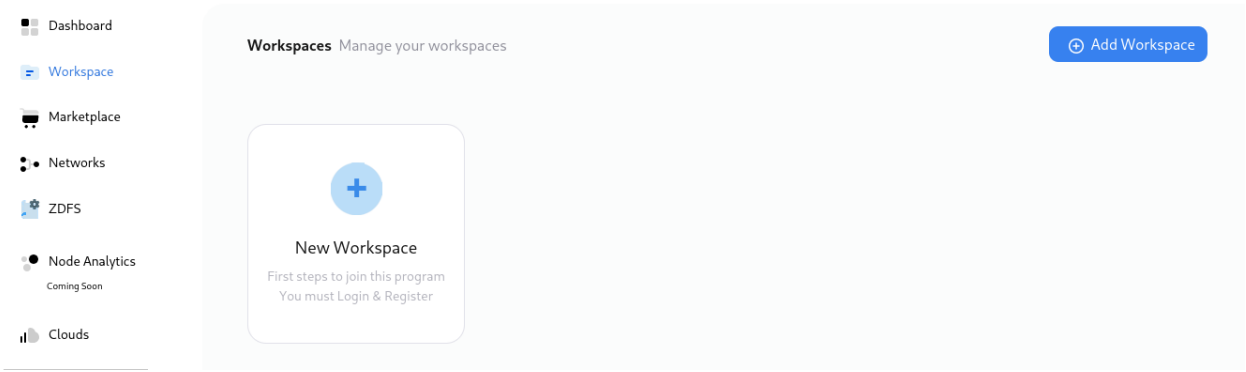
1. *Create Workspace*
2. *Create Network*

13.1 Create workspace

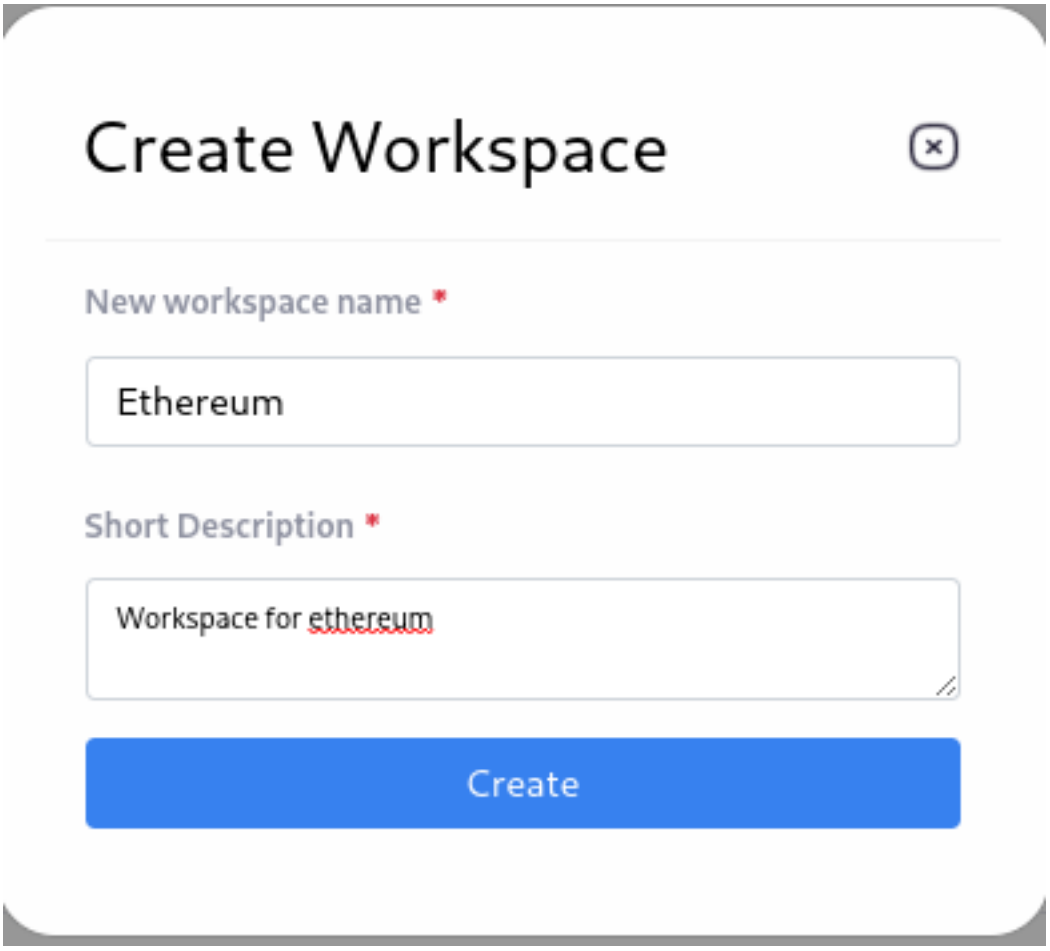
1. Click on **Workspace** on the left side navigation bar.



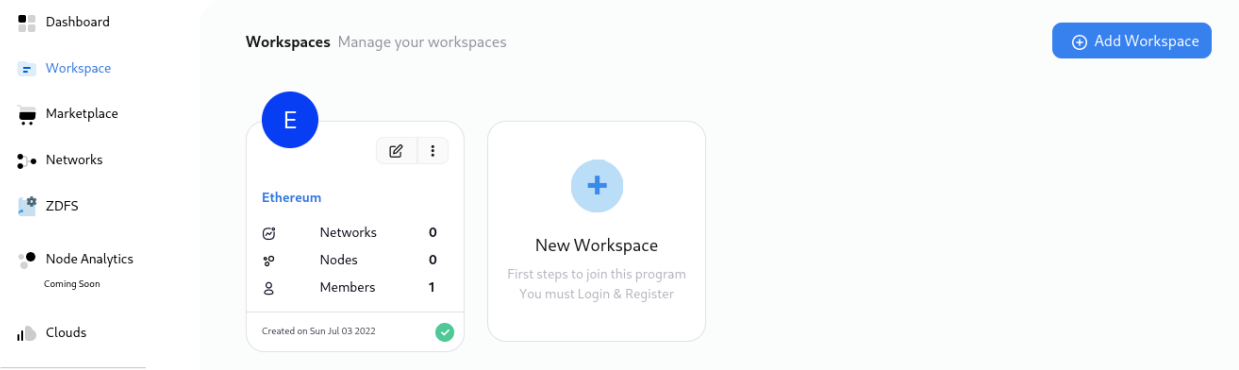
1. You will be landed to a page similar to below image.



1. Click on **New workspace**. A pop up window will appear.

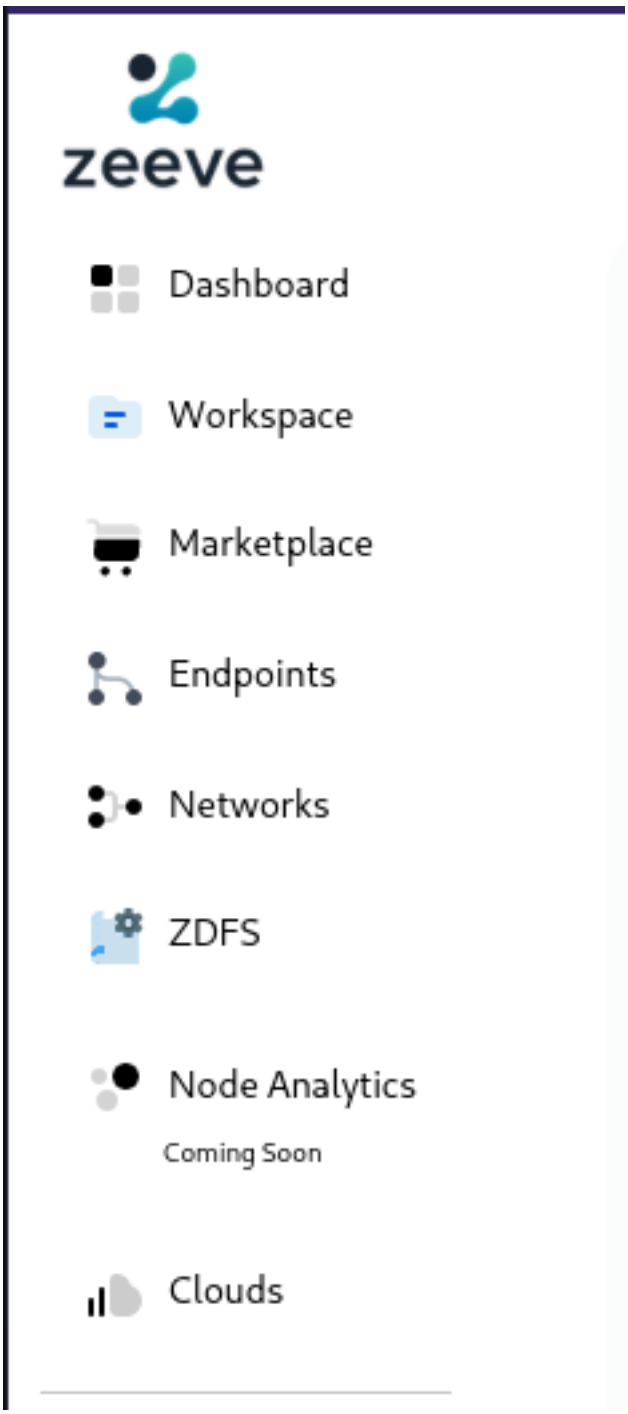


1. Give it a name of your choice(in our case we are going to name it Ethereum) and add a short description. Click **create**. You can see your newly created workspace added to the workspace tab.



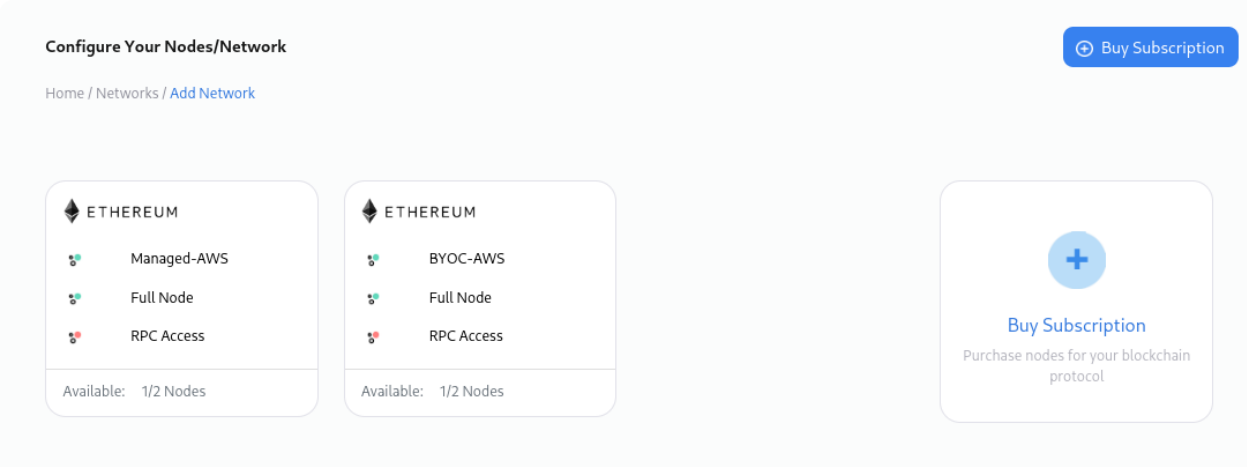
13.2 Create Network

1. Click on **Workspace** on the left side navigation bar.



1. You will be landed to **Zeeve's** network listing page where you will get the list of all of the networks you created. Click on **Add Network**.

1. You will be landed to the **Network Configuration** page which looks similar to the below image.



Here you will get to see different cards with different network configuration of diffrent protocols. These cards can be different according to your purchased subscriptions. Choose the configuration of your choice for creating a network, and click on a card accordingly. To see protocol specific configuration parameters please refer to the detailed deployment spec using the following links.

1. *Avalanche*
2. *Binance*
3. *Ethereum*
4. *Fantom*
5. *Polygon*
6. *Tron*

---

description: Get started with Hyperledger Fabric on Zeeve's platform. Our documentation provides an introduction to the key concepts, tools and techniques for building decentralized applications using Hyperledger Fabric and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

FOURTEEN

HYPERLEDGER FABRIC DEDICATED NODE SETUP

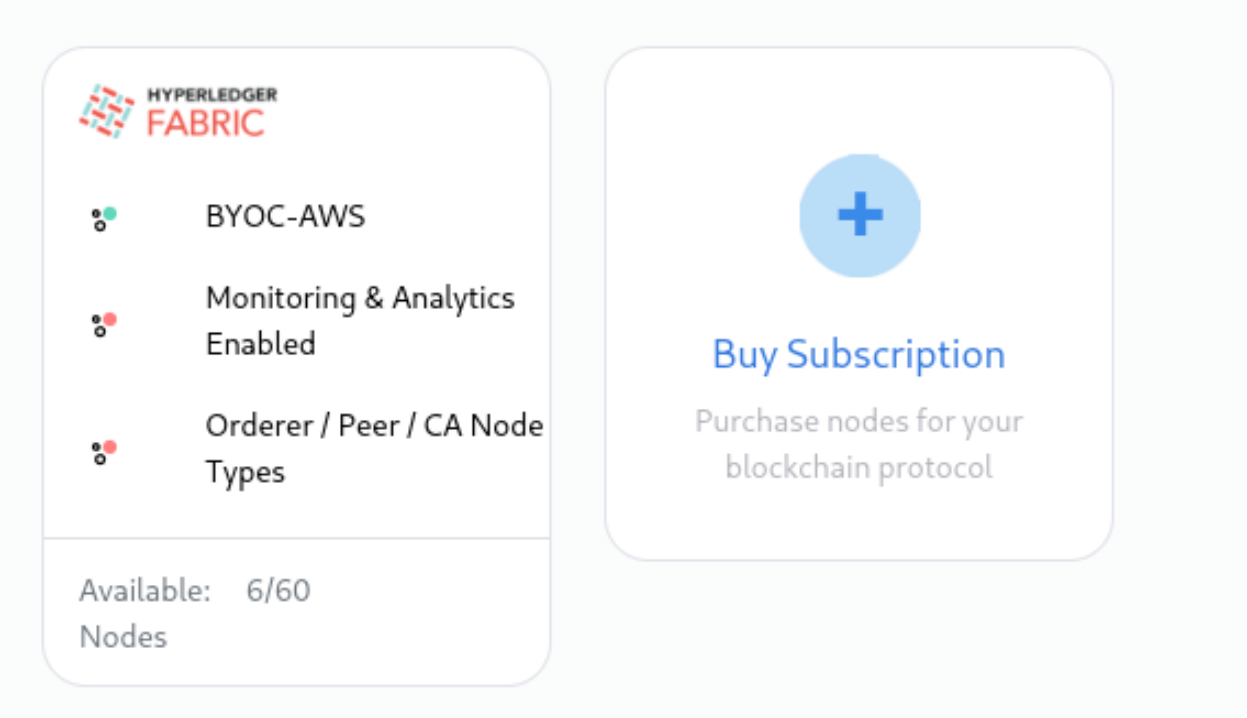
Hyperledger Fabric has one of the most exhaustive sets of available configuration parameters.

This page would help you a lot to achieve a highly customized fabric network.

1. *Create network*
2. *Add peer*
3. *Add organization*
4. *Zeeve CLI*

14.1 Create a network

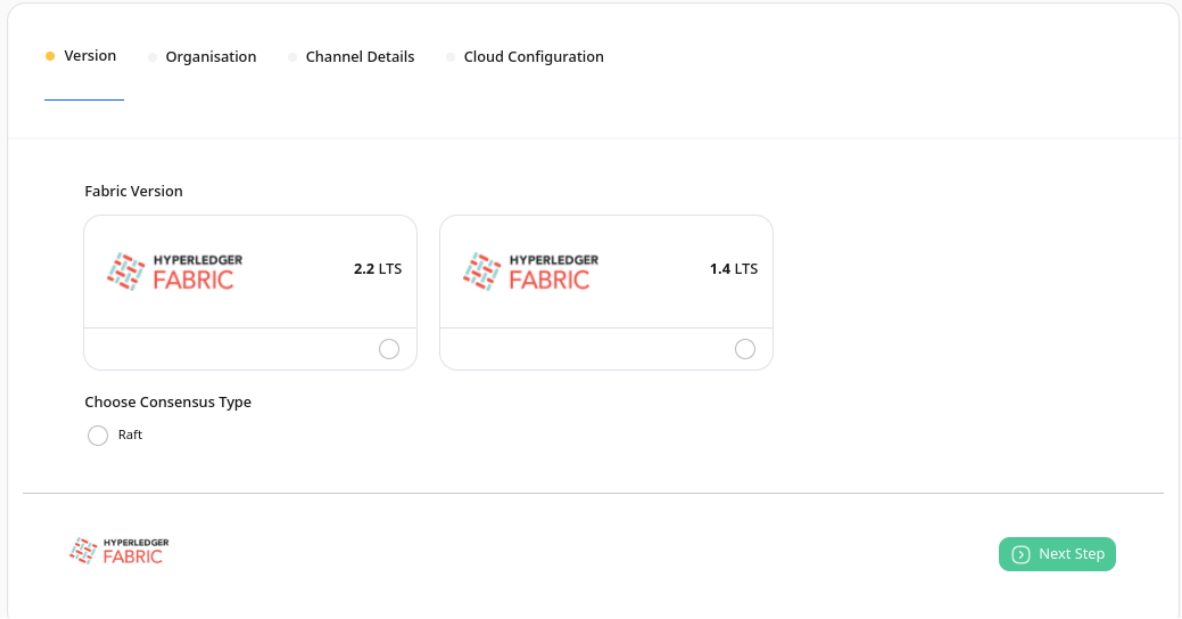
Fabric network creation is spread across 4 sections. Please read further to know about each of them. On the **Network Configuration** page you will have different cards with different network configurations for Fabric, which looks similar to the image provided below.



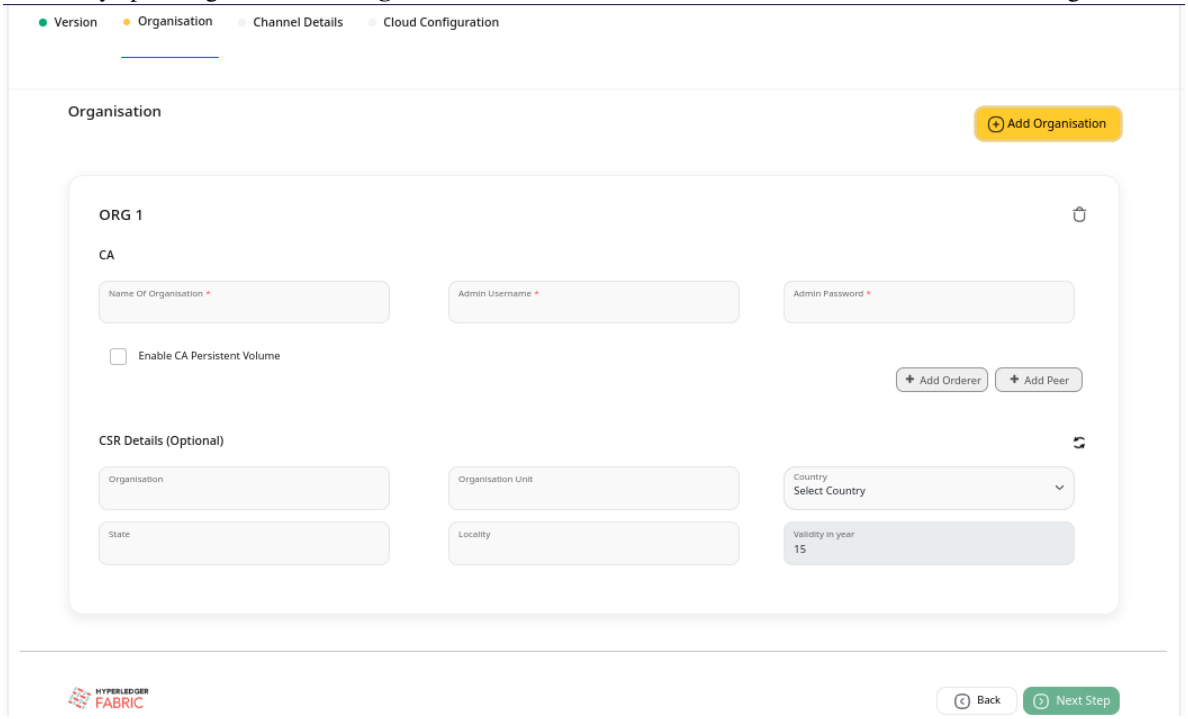
NOTE: These cards can be different for your case. Card configurations totally depend on your purchased subscription.

Choose the configuration you want. Click on the card and follow the steps accordingly.

1. In the first step choose the **Fabric version** and **Consensus** type. After that click on **Next Step** button.



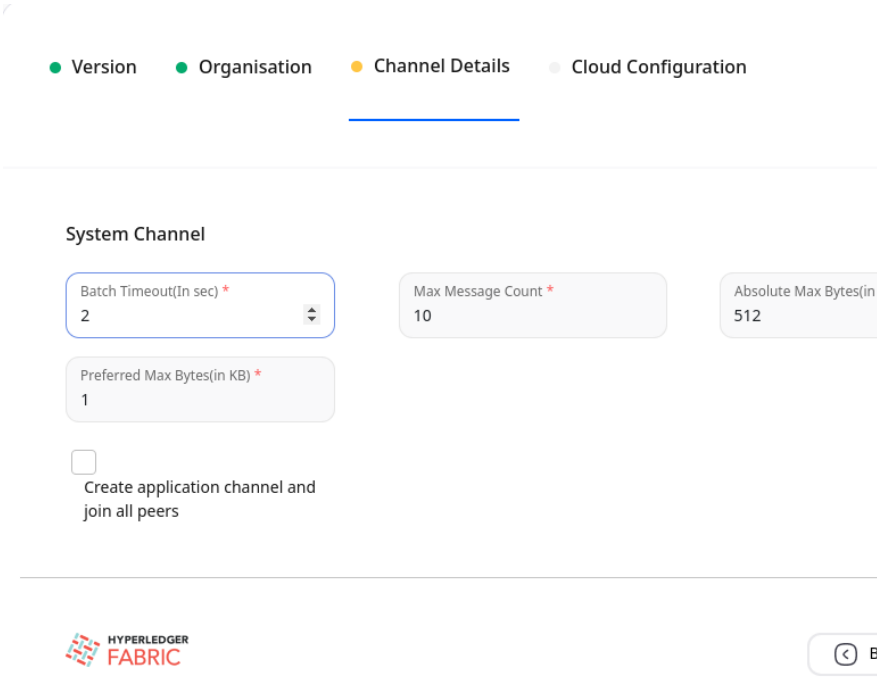
2. A fabric network is made up of a group of organizations wherein an organization is a mere stakeholder(participant) of the network, this group is called a consortium. You can add an organization by pressing the **Add organization** button and after that add a name for this organization.



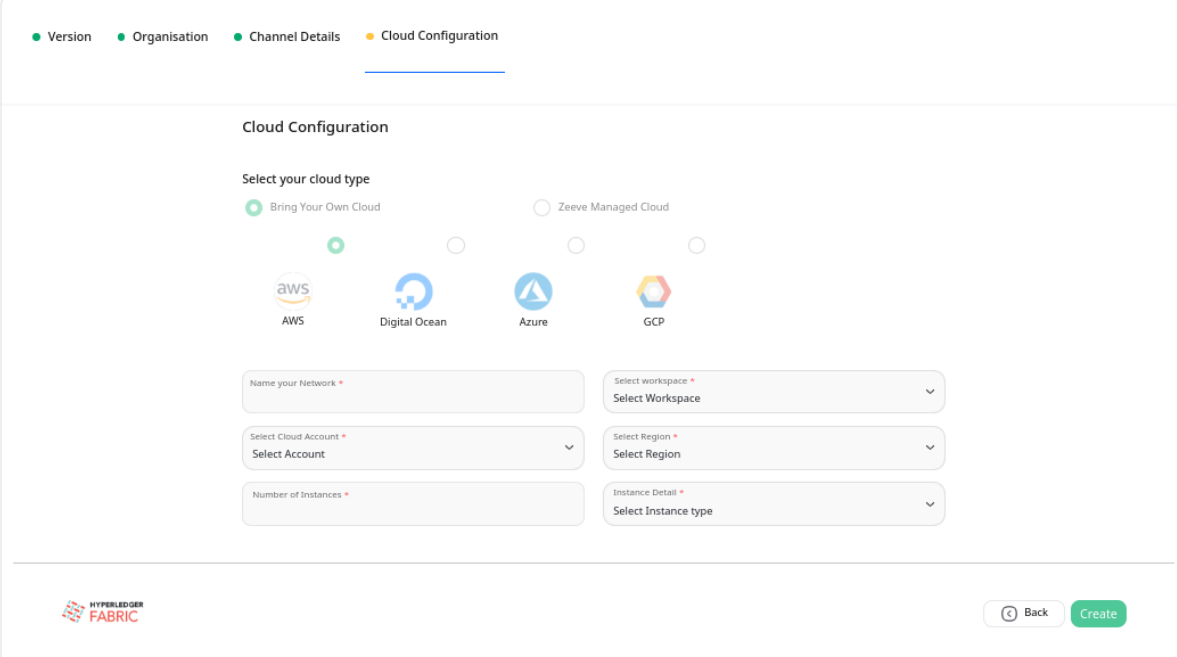
Each organization participate in the network via a few fabric specific pillars namely orderer, peer and certificate authority.

- **CA**: CA(Certificate Authority) can be configured just by providing the admin user name and password.
- **Orderer**: Zeeve supports all the three types of ordering service, which are provided by HL Fabric namely Solo (Single Orderer Network), Kafka and Raft. Making it one of the best tools for deploying fabric-based production networks.

So based upon the requirement, select the type of ordering service and just add the number of orderers using the Add Orderer button under the orderer tab of the organization section.



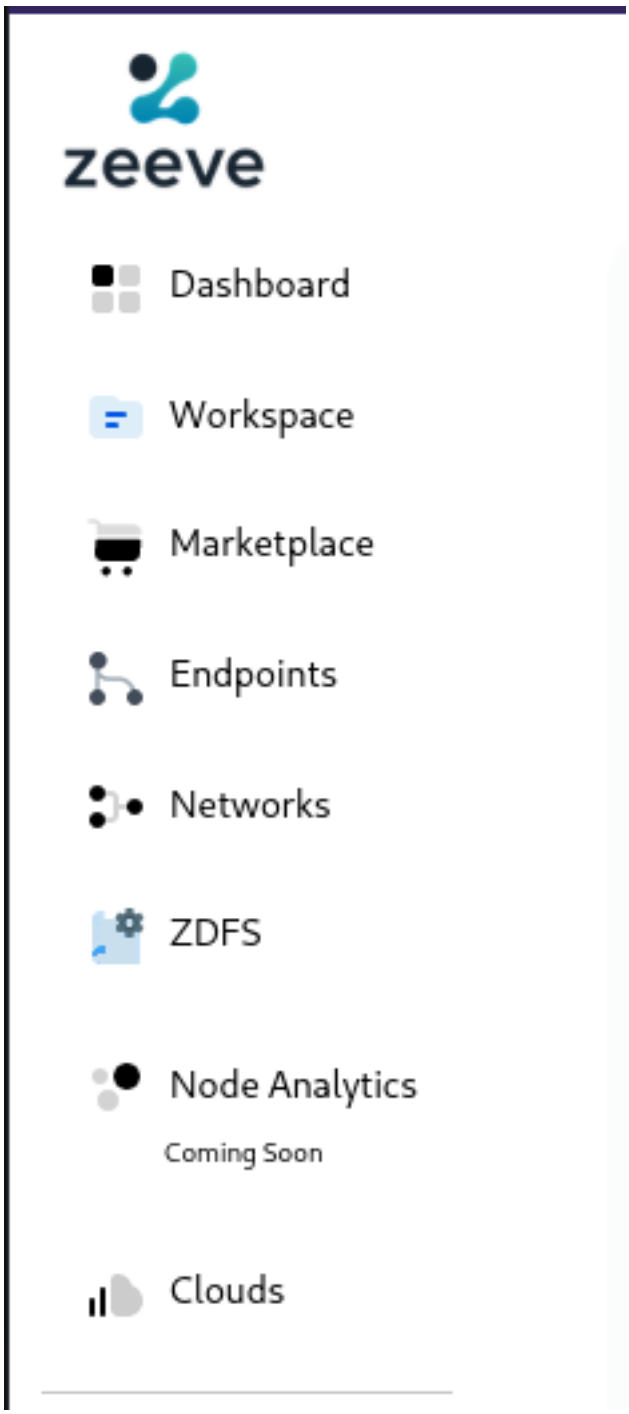
3. This is the step to configure the channel details.
   - **Batch Timeout** is the amount of time to wait after receiving the first transaction, in order to receive more transactions before cutting a block. In case we decrease this value then we get lower latency but decreasing too much will result in a decrease in throughput, as the block will not fill to its maximum capacity.
   - As indicated in the above screenshot `{ "timeout": "2s" }`
   - **Maximum Message Count** indicates the maximum number of transactions in a block. Also, if we look into
   - **Absolute Maximum Bytes**: It indicates the maximum size of a block that can be built in the channel.
   - In respect of **Absolute Maximum Bytes**, there is another parameter viz.,
   - **Preferred maximum Bytes** which is nothing but the minimum size of a block.
4. This is the last step in creating a network. select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**, and then click on the **Create** button.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

14.2 View Network

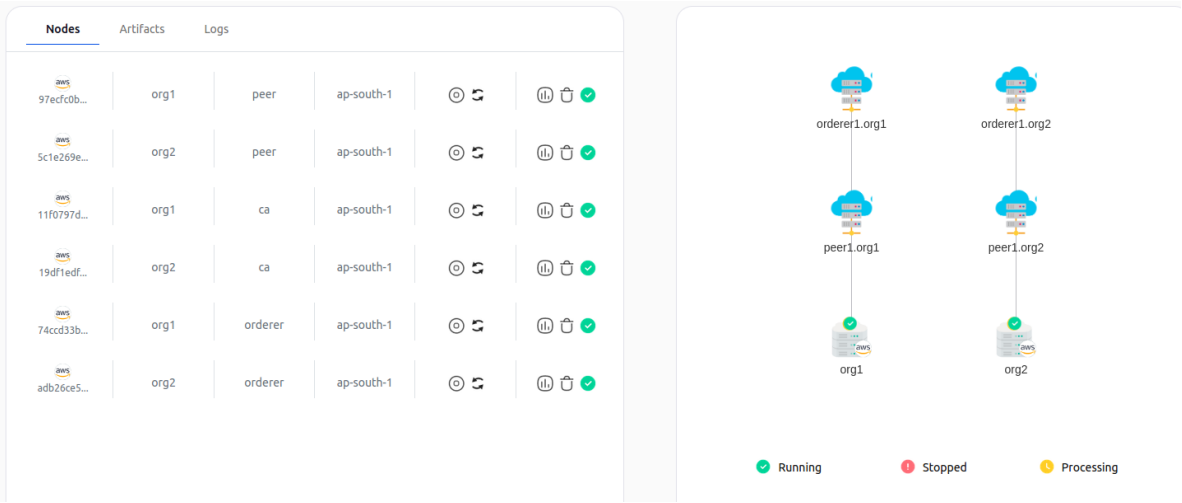
1. Click on **Networks** on the left side navigation bar.



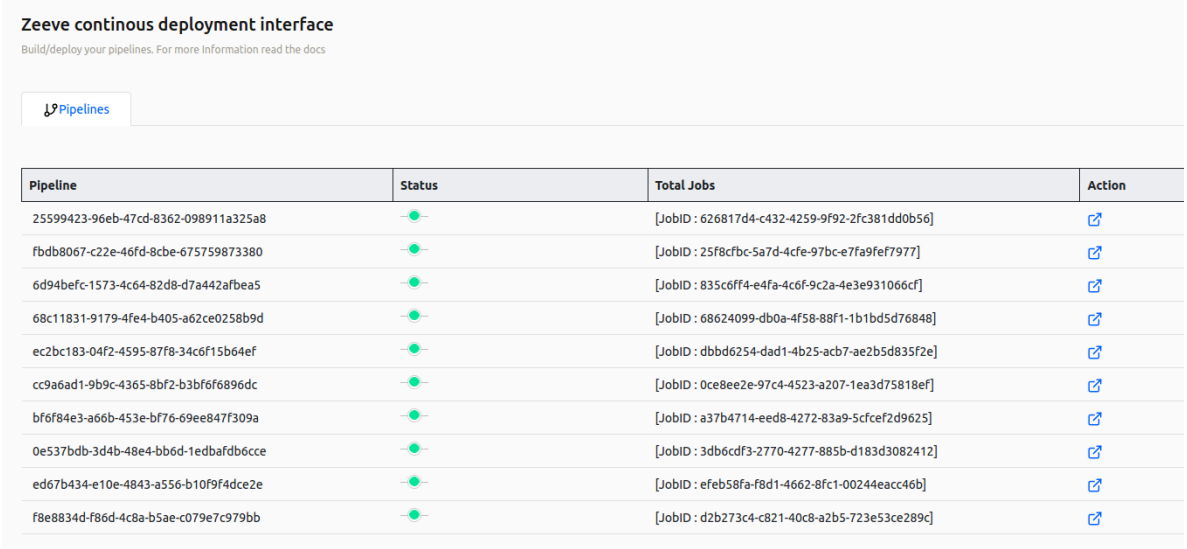
2. You will be landed on a page similar to the below image. You can view all of your networks listed on this page.

NOTE This page can be different in your case.

1. Select the Fabric network you want. After clicking on the network card you can view the nodes attached to the selected network. Network Nodes are listed with details - node's unique ID and cloud provider, an organization it is part of, node's role(peer, ca, orderer), and node's deployment region.



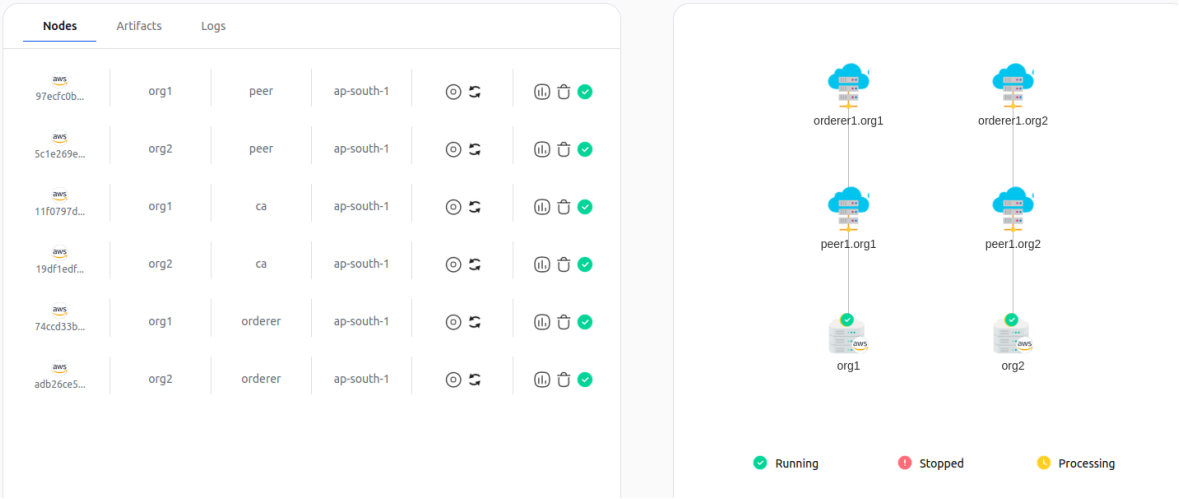
2. Pipelines are listed with details - pipeline unique ID, the status of the pipeline(completed, in progress, failed), Total Jobs that it holds, and Action (to open up jobs inside it).

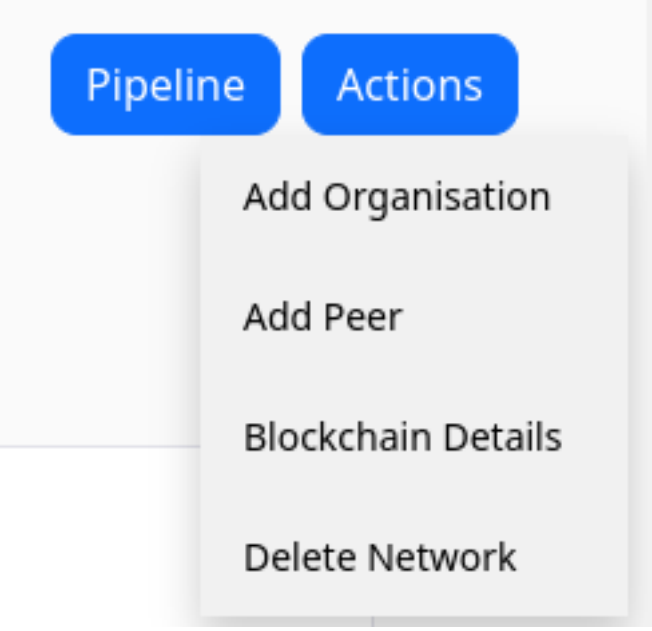


14.3 Add peer

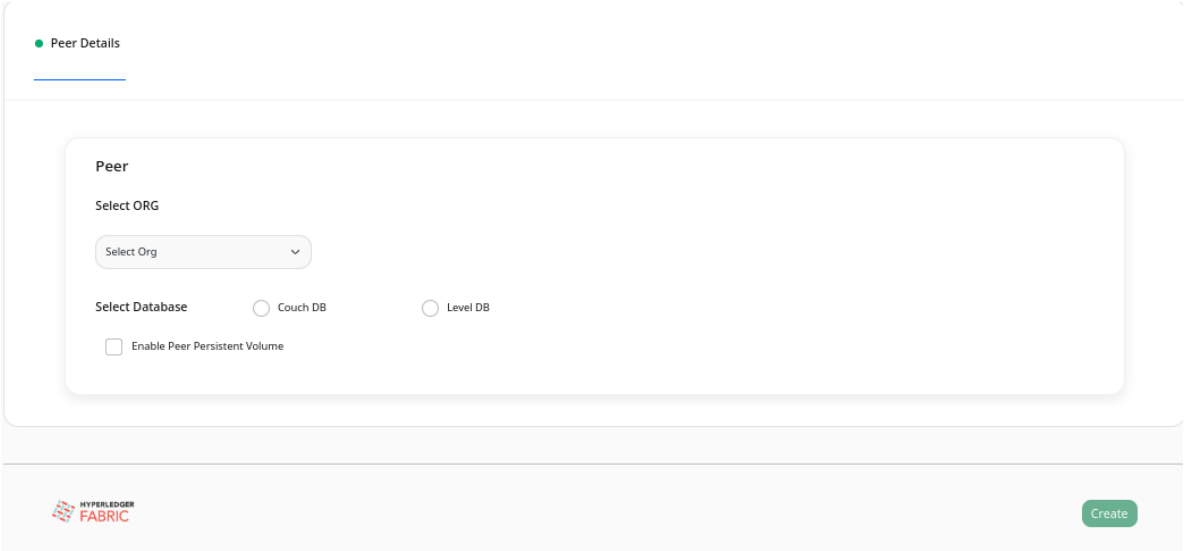
A peer can be added to the organization by only following the two steps mentioned below. You just need to choose the type of peer service for each peer you want to go with, it can be either level Db or CouchDB based, and specify whether you want to have a persistent volume for the same.

1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



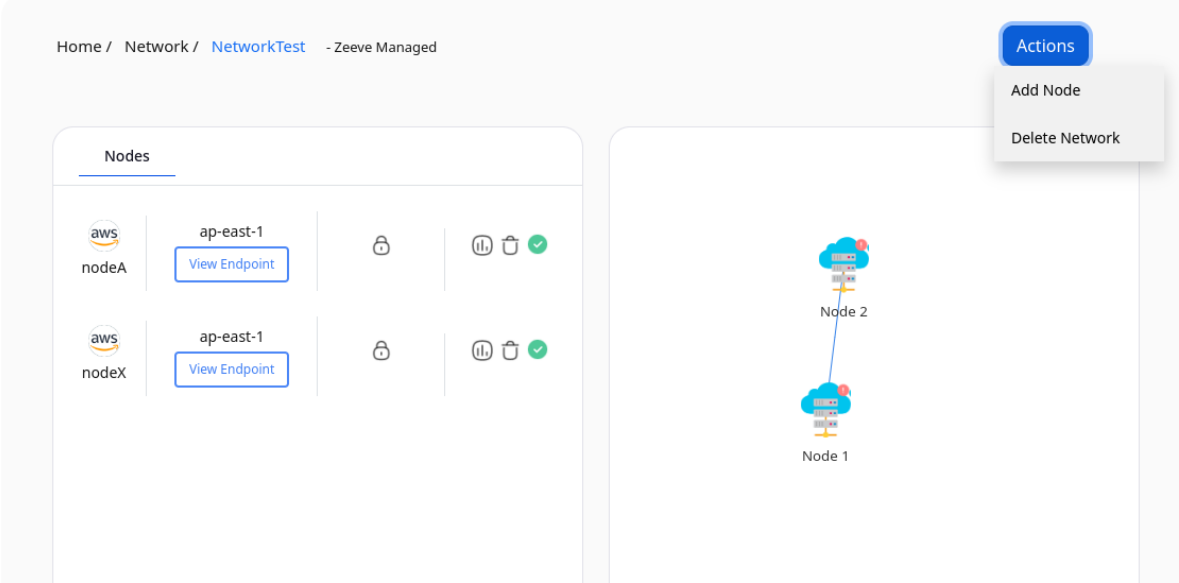


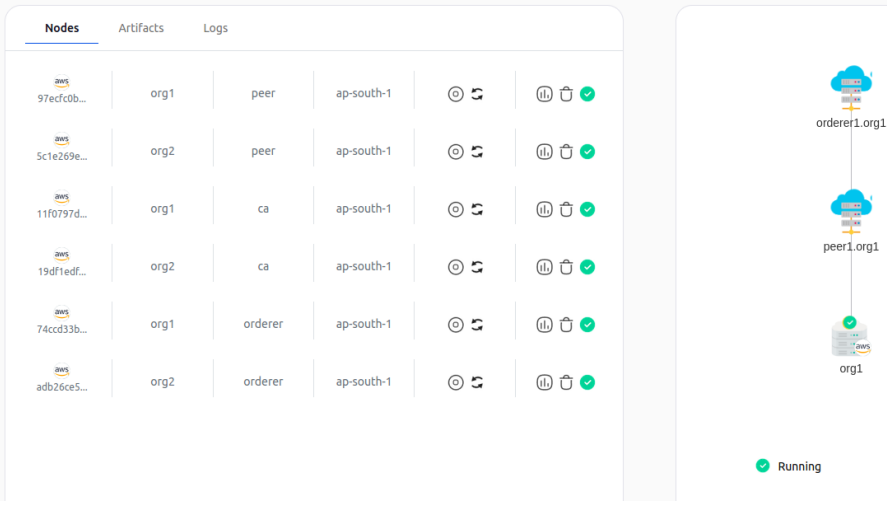
2. Click on **Actions** button and select **Add Peer**.
3. You will get to see a web page similar to the image provided below. Fill the details and click **create** button.



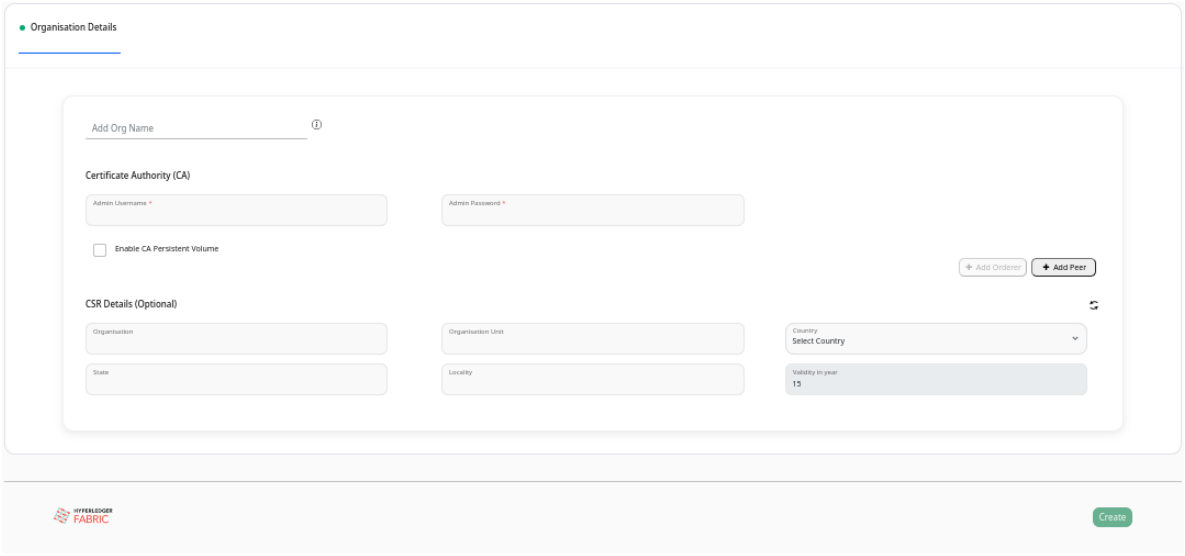
14.4 Add organization

1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.





2. Click on **Actions** button and select **Add Peer**.
3. You will get to see a web page similar to the image provided below. Fill the details and click **create** button.



14.5 Zeeve CLI

This section talks about how to use the Zeeve CLI and how to perform fabric chain code operations.

1. *Create CLI access*
2. *Chaincode pipelines*

14.5.1 Create CLI Access

1. Navigate to settings and then click on "API Credentials" to see a list of CLI/API credentials.
2. Click "Create key".
3. Provide a name, and select one or more networks to associate with the key.
4. Add appropriate permissions for the operations that the keys are being created.
5. Click "Create key" and then copy/save generated keys.

14.5.2 Chaincode Pipelines

1. Login with Zeeve CLI using the earlier created keys and add these URLs as well -

```
zeeve login -i < access-key > -s < secret-key > -ae https://app.zeeve.io/auth/
↪cli/login -fe https://app.zeeve.io/fabric-backend/
```

2. Zeeve CLI supports fabric chaincode operations - package, install, deploy, approve and commit.
3. Package: builds chaincode image, creates ECR and pushes into ECR, and creates chaincode connection package to be installed onto peers. Zeeve chaincode pipelines can deploy chaincode as a service (For more details). Following are the inputs required to run package operation
   - Chaincode zip file is a compressed file of chaincode with the name chaincode.tar.gz

```
zeeve fabric chaincode package -f /path/to/chaincode.tar.gz -n 44b28e1f-7296-
↪42a4-8904-e04341edfb27 -c marbles2 -v 1.0 -o org1 <br></br>
```

```
lakshay@PRO-IT-LAP072:~$ zeeve fabric chaincode package --help
Usage: zeeve fabric chaincode package [options]

Options:
  -f, --file-path <file-path>    File path for chaincode zip file
  -n, --network-id <network-id>  Id of the Network
  -c, --chaincode <chaincode>    chaincode name
  -v, --version <version>        chaincode version
  -o, --org <org>                chaincode organisation
  -h, --help                     display help for command
lakshay@PRO-IT-LAP072:~$
```

4. Install: runs chaincode install operation on given peer URLs.
   - The install operation can be run on a set of peers in a given org with the peer-URLs option.

```
zeeve fabric chaincode install -n 44b28e1f-7296-42a4-8904-e04341edfb27 -c␣
↪marbles2 -v 1.0 -o org1 -p peer1.org1.example.fabric.zeeve.net, peer2.org1.
↪example.fabric.zeeve.net
```

```
lakshay@PRO-IT-LAP072:~$ zeeve fabric chaincode install --help
Usage: zeeve fabric chaincode install [options]

Options:
  -n, --network-id <network-id>   Id of the Network
  -c, --chaincode <chaincode>     chaincode name
  -v, --version <version>         chaincode version
  -o, --org <org>                 chaincode organisation
  -p, --peer-urls <peer-urls...>  chaincode peer urls
  -h, --help                      display help for command
lakshay@PRO-IT-LAP072:~$
```

5. Deploy: starts chaincode pod/service in Kubernetes cluster.

```
- zeeve fabric chaincode deploy -n 44b28e1f-7296-42a4-8904-e04341edfb27 -c␣
↪marbles3 -v 1.0 -o org1 -p peer1.org1.example.fabric.zeeve.net ,peer2.org1.
↪example.fabric.zeeve.net
```

6. Approve: runs 'peer chaincode approve' on a set of given peers in an org in the channel provided.

```
 zeeve fabric chaincode approve -n 44b28e1f-7296-42a4-8904-e04341edfb27 -cc␣
↪marbles3 -v 1.0 -o org1 -p peer1.org1.example.fabric.zeeve.net, peer2.org1.
↪example.fabric.zeeve.net -ch mychannel -s 1
```

```
lakshay@PRO-IT-LAP072:~$ zeeve fabric chaincode approve --help
Usage: zeeve fabric chaincode approve [options]

Options:
  -n, --network-id <network-id>   Id of the Network
  -cc, --chaincode <chaincode>    chaincode name
  -v, --version <version>         chaincode version
  -o, --org <org>                 chaincode organisation
  -ch, --channel <channel>        channel name to approve chaincode
  -s, --sequence <sequence>       chaincode sequence
  -p, --peer-urls <peer-urls...>  chaincode peer urls
  -h, --help                      display help for command
lakshay@PRO-IT-LAP072:~$
```

7. Commit: runs 'peer chaincode commit' and also inits the chaincode with Init method on the given set of peers and organizations in the provided channel.

```
 zeeve fabric chaincode commit -n 44b28e1f-7296-42a4-8904-e04341edfb27 -cc␣
↪marbles3 -v 1.0 -o org1,org2 -p peer1.org1.example.fabric.zeeve.net,peer2.org1.
↪example.fabric.zeeve.net -ch mychannel -s 1
```

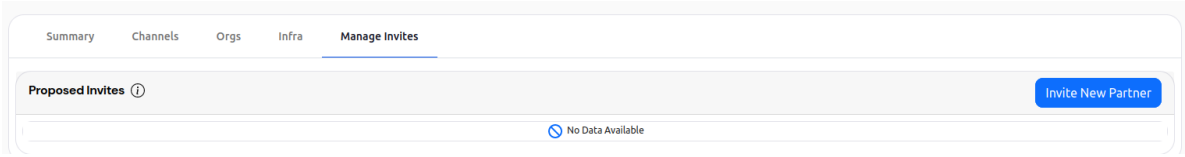
```
lakshay@PRO-IT-LAP072:~$ zeeve fabric chaincode commit --help
Usage: zeeve fabric chaincode commit [options]

Options:
  -n, --network-id <network-id>   Id of the Network
  -cc, --chaincode <chaincode>    chaincode name
  -v, --version <version>         chaincode version
  -o, --orgs <org...>             chaincode organisation
  -p, --peer-urls <peer-urls...>  chaincode peer urls
  -ch, --channel <channel>        channel name to approve chaincode
  -s, --sequence <sequence>       chaincode sequence
  -h, --help                      display help for command
lakshay@PRO-IT-LAP072:~$
```

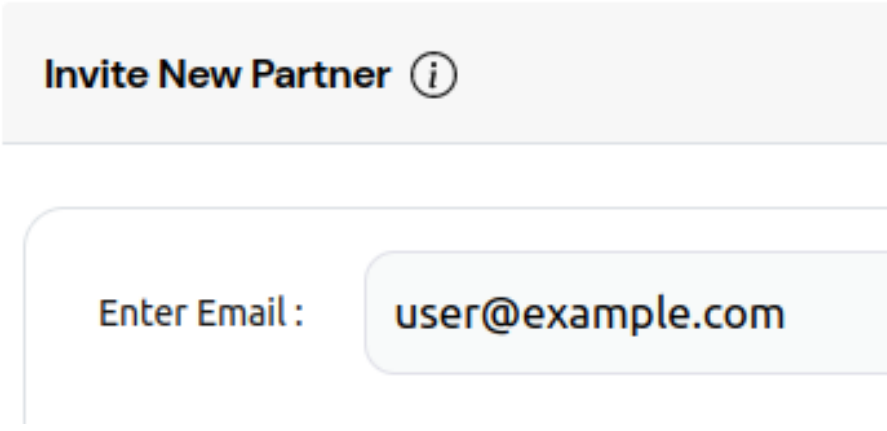
14.6 Multitenant Networks

Zeeve also supports the creation of Mulitenant networks in Hyperledger Fabric. This means that two or more users may be part of a network and may perform network operations collectively.

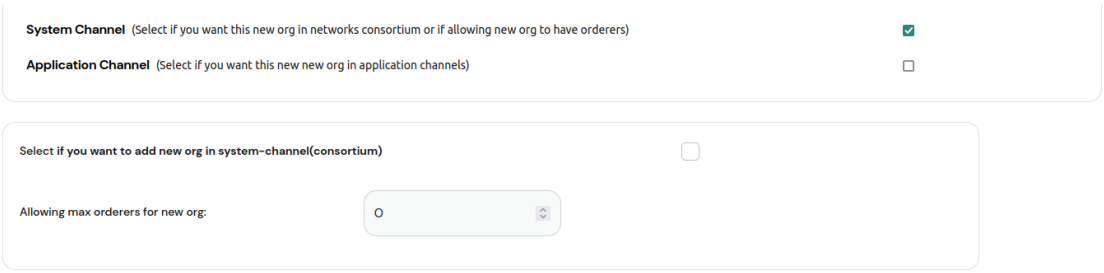
14.6.1 Inviting A User

1. Select the network to which you want to add a node, and click on the network card.
2. Click on the Manage Invites tab and click on Invite New Partner.



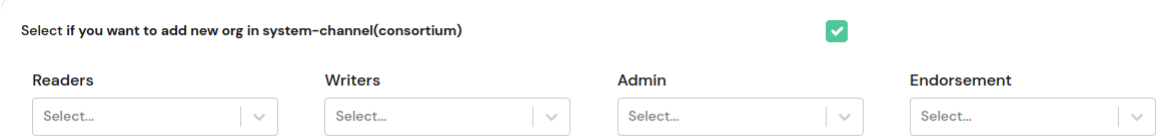


3. Enter the Email Address of the user to invite.
4. Select the system channel checkbox if you want the invited user's organization to have orderers or be a part of the



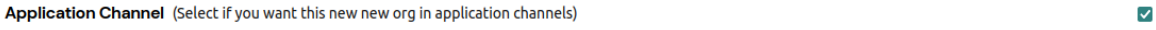
consortium.

5. Fill in the maximum number of orderers for the new user.
6. Select the consortium checkbox to add the user to consortium. Also fill in the system channel policy for the new

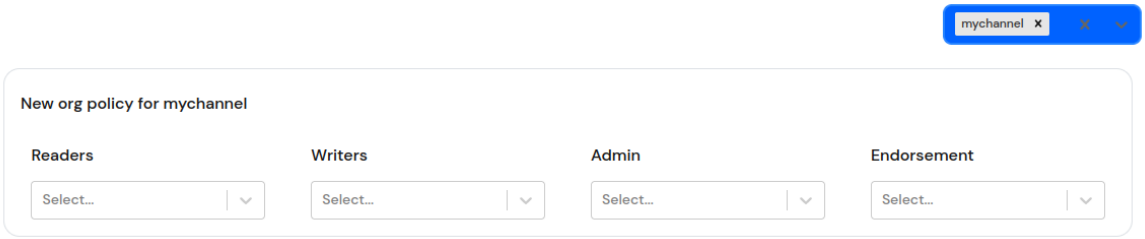


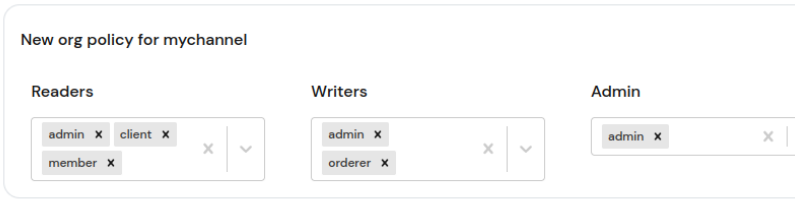
organization.

7. Select the application channel checkbox to make the new organization a part of atleast one pre-existing application channel.

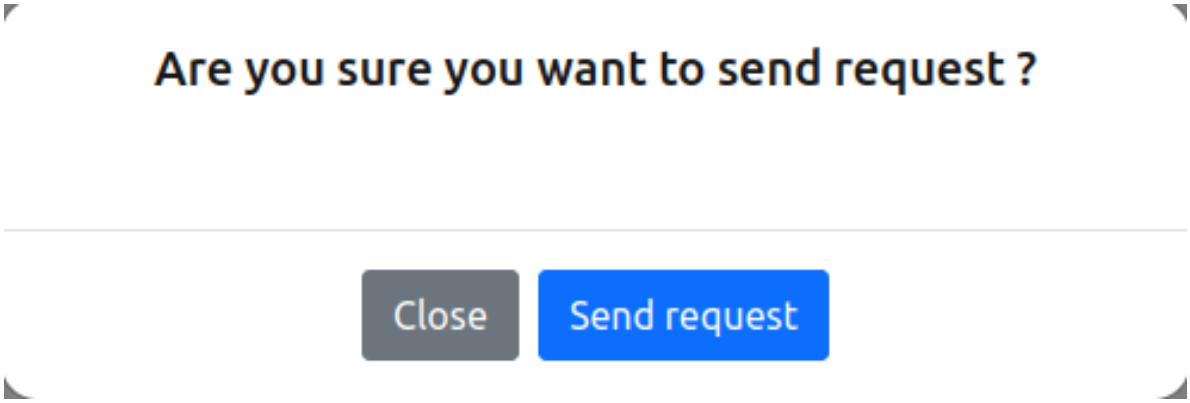


8. Click on the dropdown and select the channel you want to new organization to be a part of. You can select more than one channels but alteast one application channel is required.



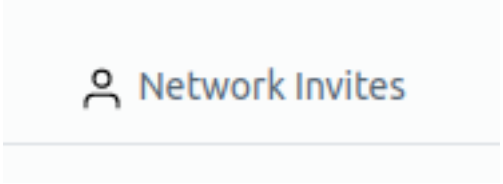


9. Fill in the channel policy for the application channel.
10. Repeat this for all application channels.
11. When all the details are filled, click on the “Submit” button. This will send an invite to the user.

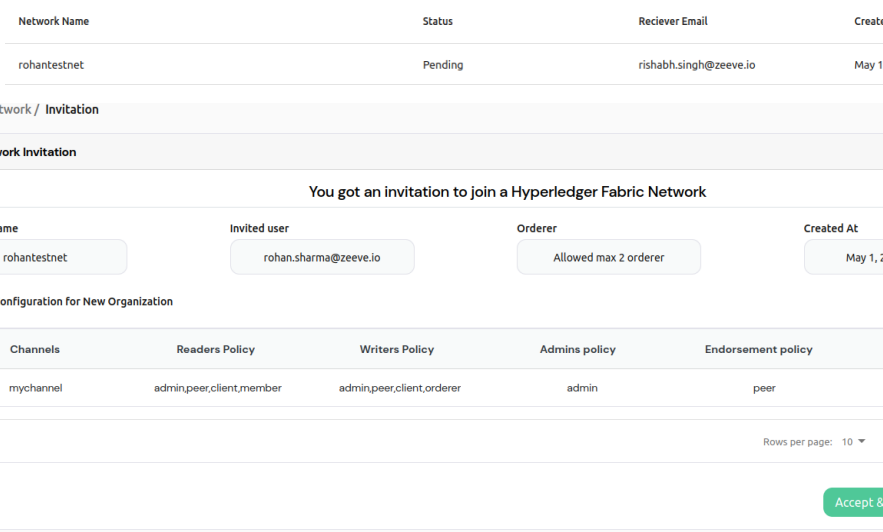


14.6.2 Accepting The Invite

1. Go to the settings page
2. Click on the Network Invite tab.



3. Click on the **View Details** button for the invite.

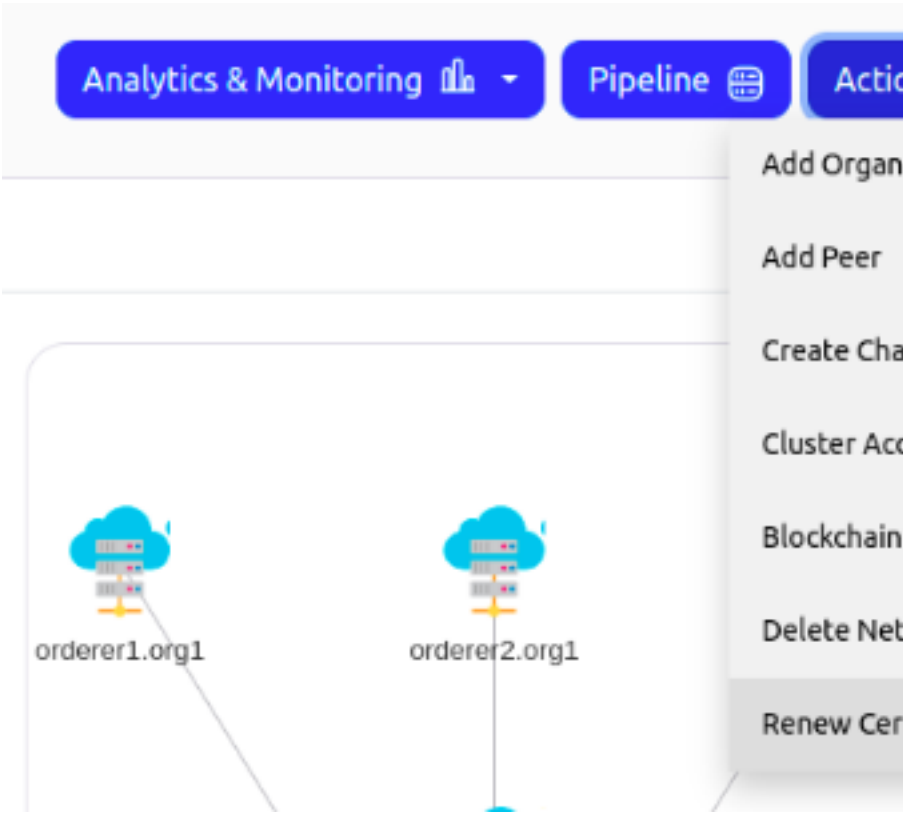


4. Verify the details and click on approve.
5. Fill in the details of the new organization.
6. Click on submit to deploy the new organization.

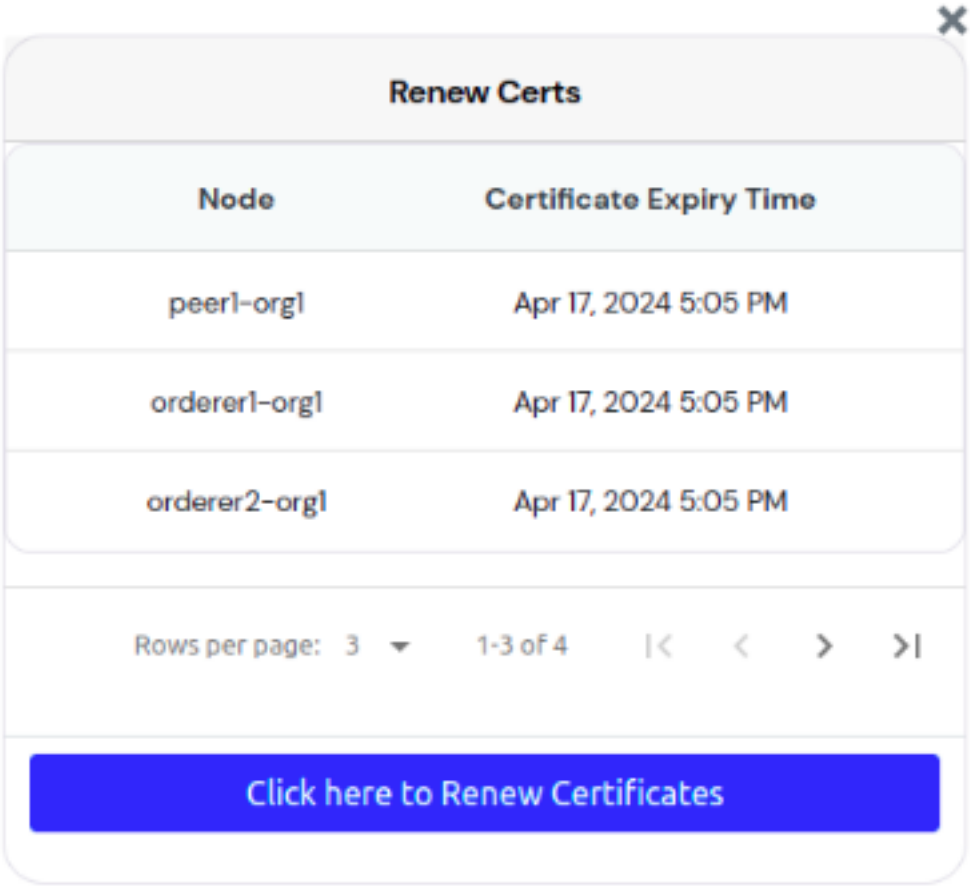
14.7 Certificate Renewal

Zeeve supports the ability to renew your fabric certificates at a click of a button. Hyperledger fabric network certificates expire in 365 days. This means that certificates need to be renewed atleast once a year.

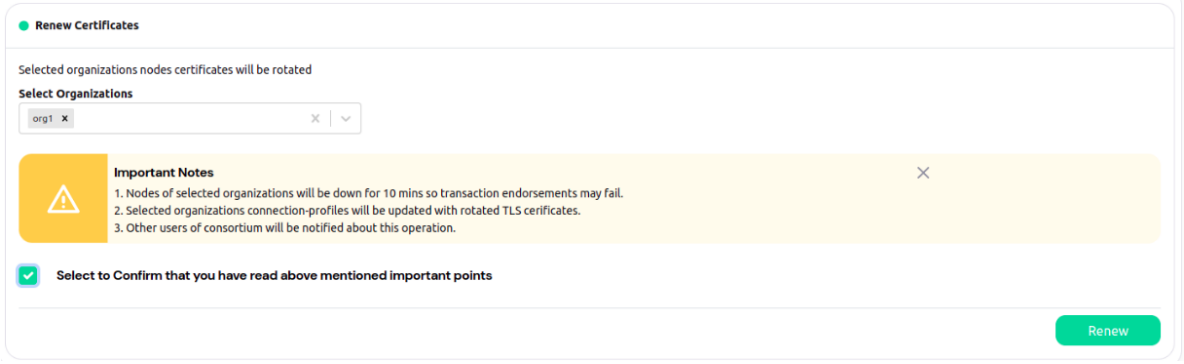
14.7.1 Renewing Certificates



1. Click on **Renew Certs** button on actions tab.
2. You can see the details of the certificate renewal with expiry date.



3. Select the organizations for which you need to renew the certificates and click on Renew button.



---

description: Explore the capabilities of Ethereum on Zeeve's platform. Our documentation provides an introduction to the key concepts, tools and techniques for building decentralized applications using Ethereum and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

FIFTEEN

ETHEREUM DEDICATED NODE SETUP

This section will guide you about different actions you can perform for Ethereum.

1. *Dedicated nodes*
2. *Api endpoints*

15.1 Dedicated nodes

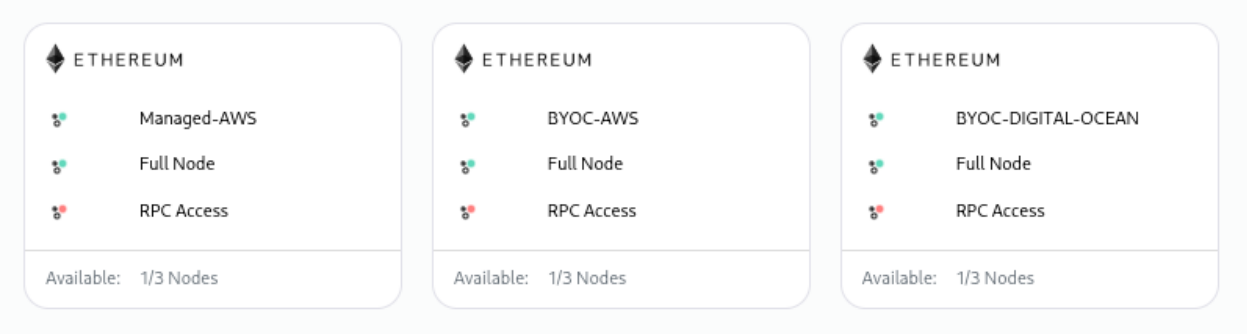
1. *Create network*
2. *Add a node*
3. *Delete a node*
4. *Delete network*

15.1.1 Create a network

This section will provide you with detailed steps for creating a network of **Ethereum**.

NOTE Please make sure to follow the steps mentioned earlier before proceeding.

On the **Network Configuration** page you will have different cards with different network configurations for Ethereum, which looks similar to the image provided below.

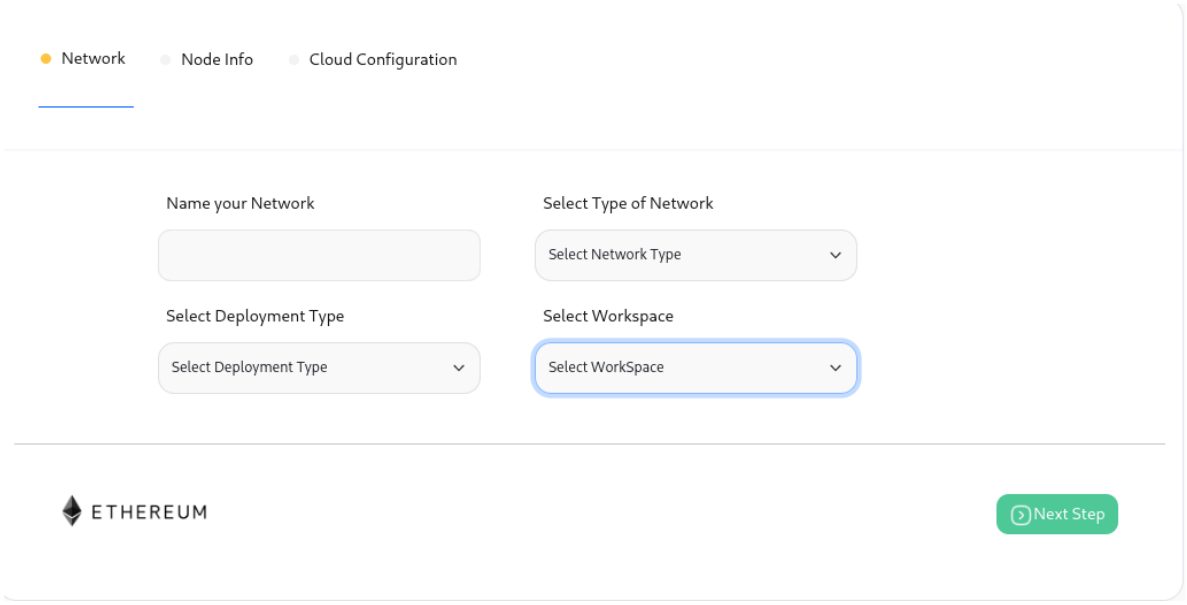


NOTE: *These cards can be different for your case. Card configurations depend on your purchased subscription.*

You can Choose **Managed-AWS** (Zeeve's managed hosting) for the infrastructure of your node or you can use your cloud account (AWS/DO) for the hosting of your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

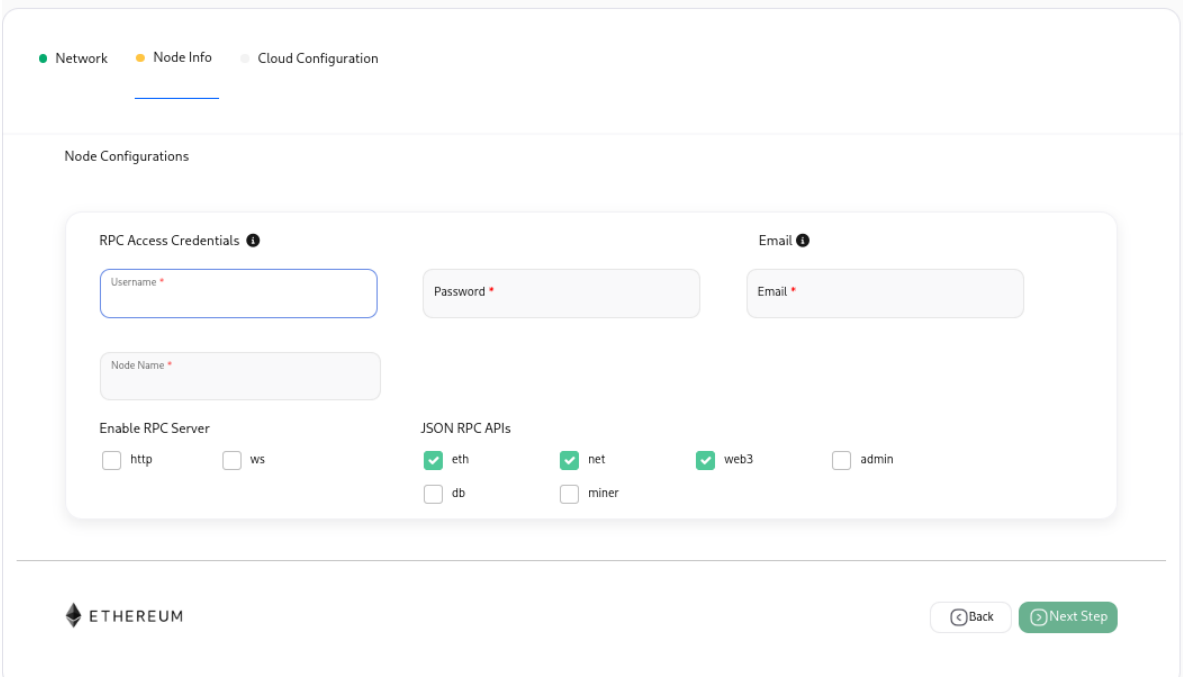
1. **Network Info:** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.



- **Name of Network**: To uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: This defines the deployment type
- **Type Of Network**
  - **MainNet**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade Ethereum dapps.
  - **Ropsten**: This is a testnet you can use for your non-production needs like testing or demonstrations.
  - **Rinkeby**: This is yet another testnet that can be used for non-production needs like testing or demonstrations, however unlike ropsten it just supports geth.
- **Workspace**:

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration:** In this section, you have to fill in the details of the nodes you want to add to your network.



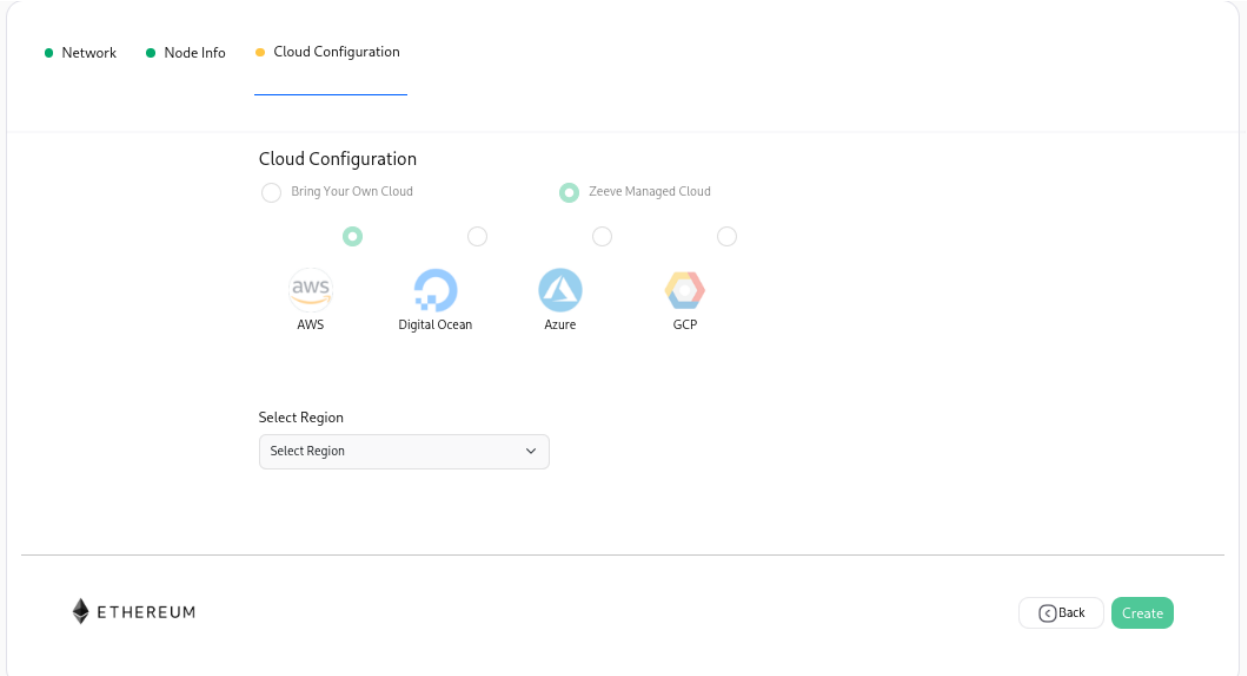
- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Email**: Fill email of yours.
- **Node Name** To identify your nodes, this field will be used.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.
- **JSON RPC APIs**: JSON RPC API is a bridge that allows dApps to connect to nodes.

After providing the details click on **Next step** to go to the last step.

1. **Cloud Configuration:** This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**.
    1. *Manged - AWS*
    2. *BYOC - AWS*
    3. *BYOC - DO*

Managed - AWS

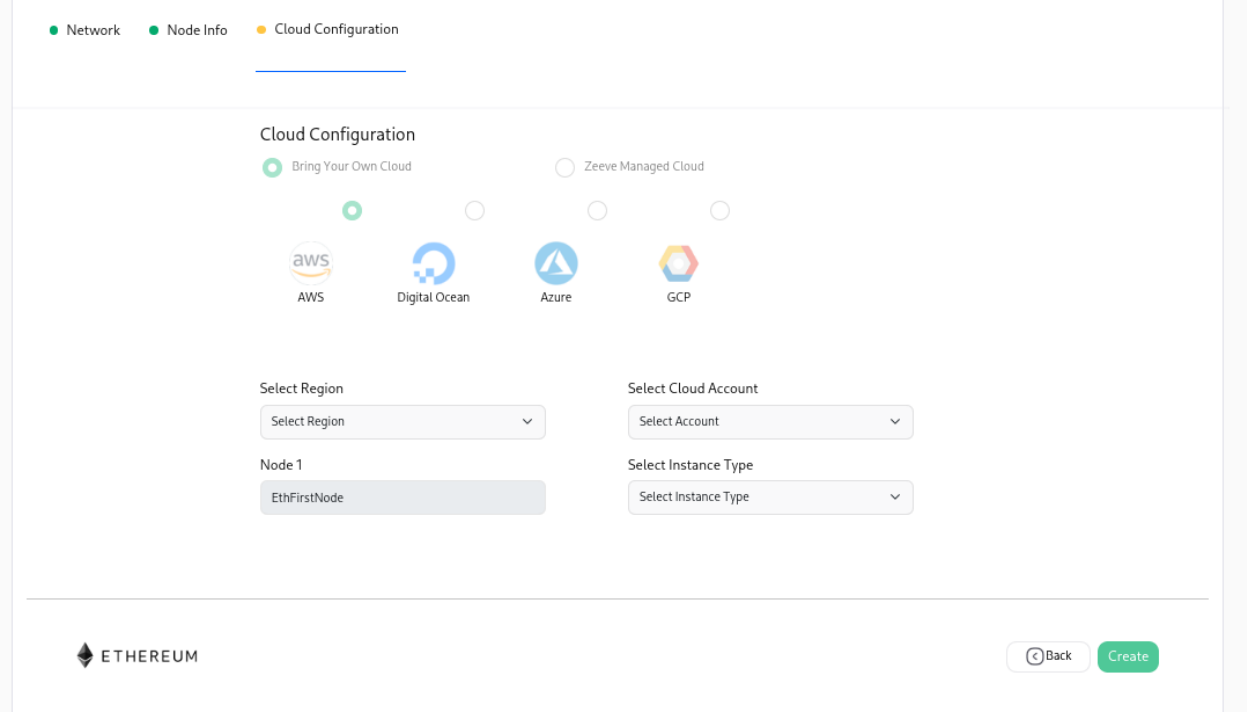
In the case of **Managed - AWS**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.

BYOC - AWS

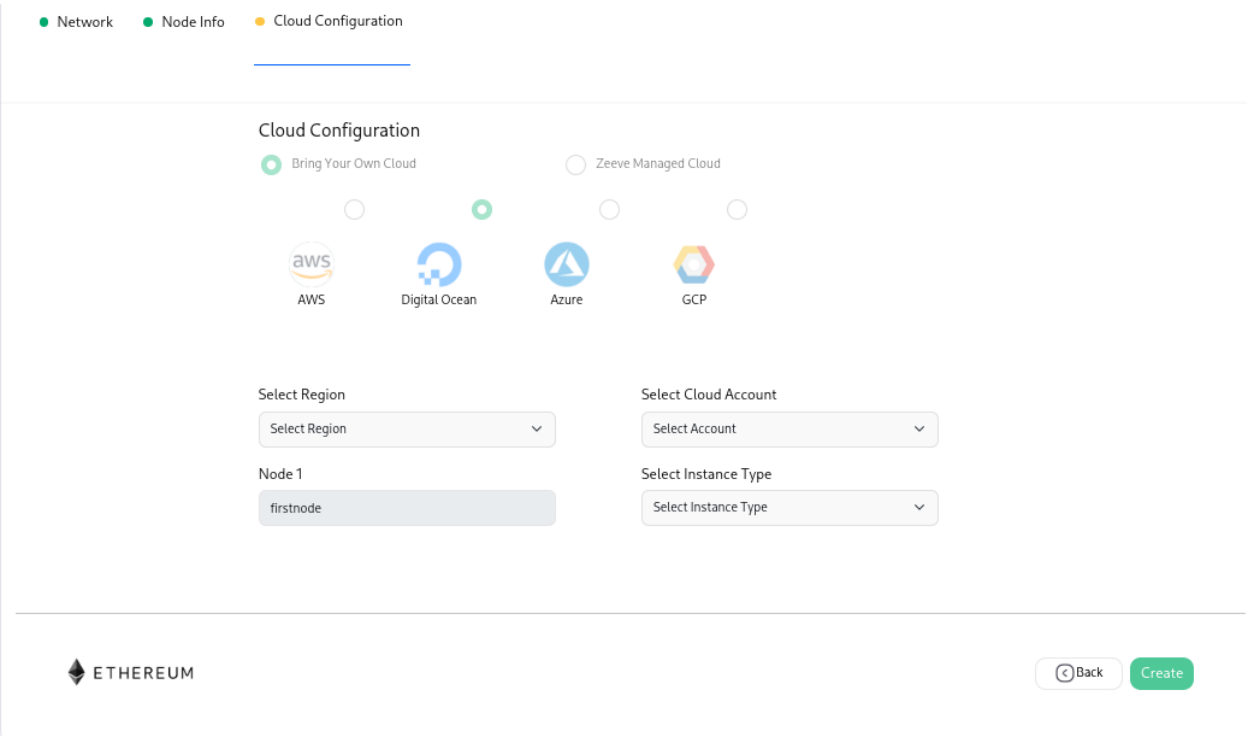
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

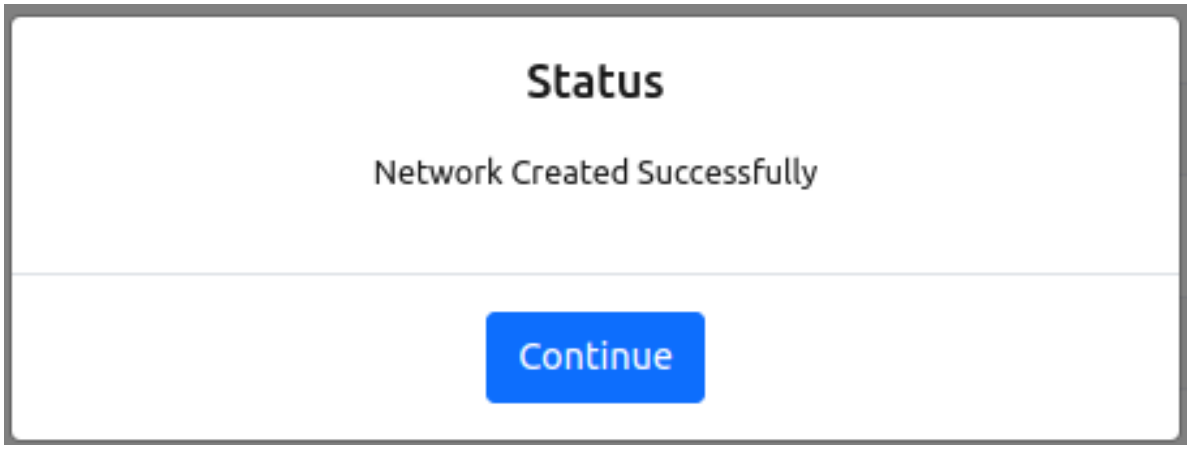
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.

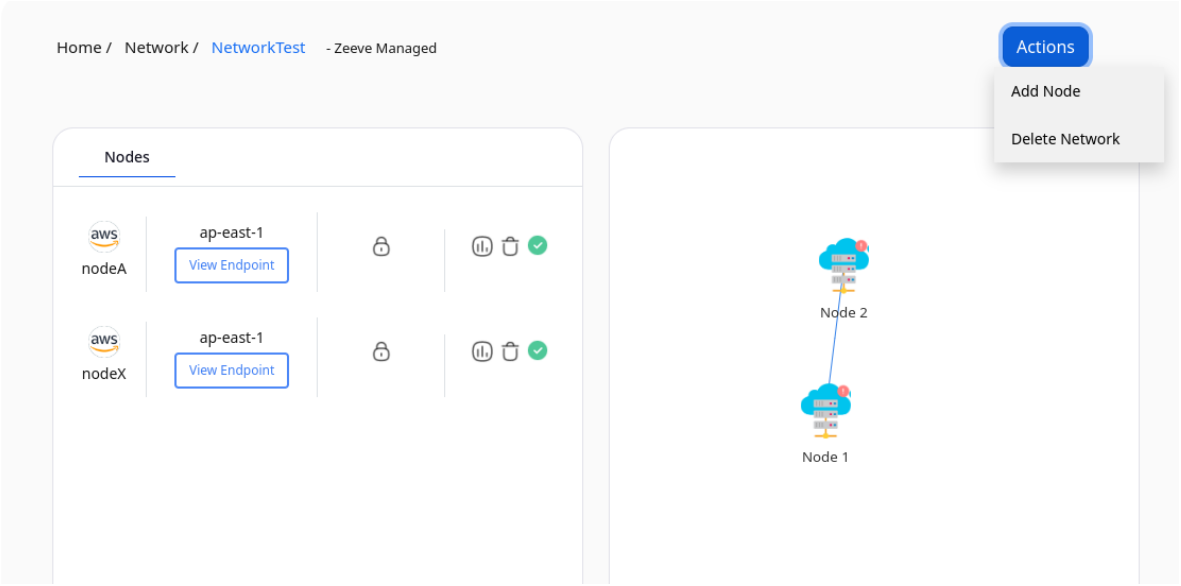


2. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes listed you've just added to the network.

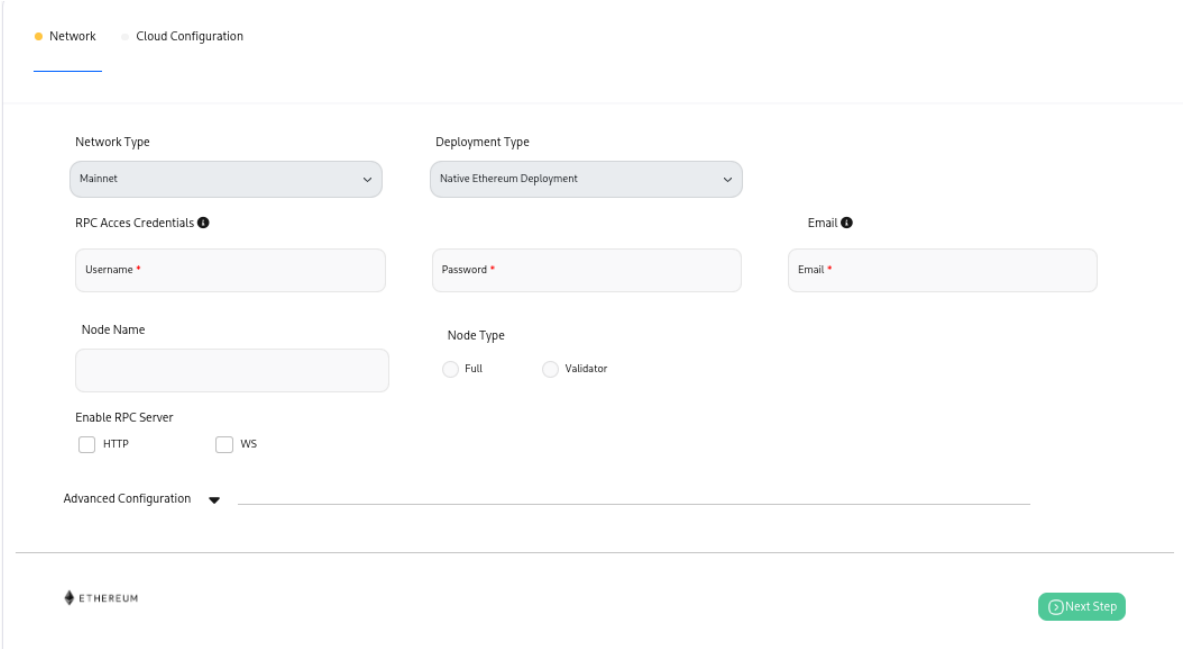
15.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

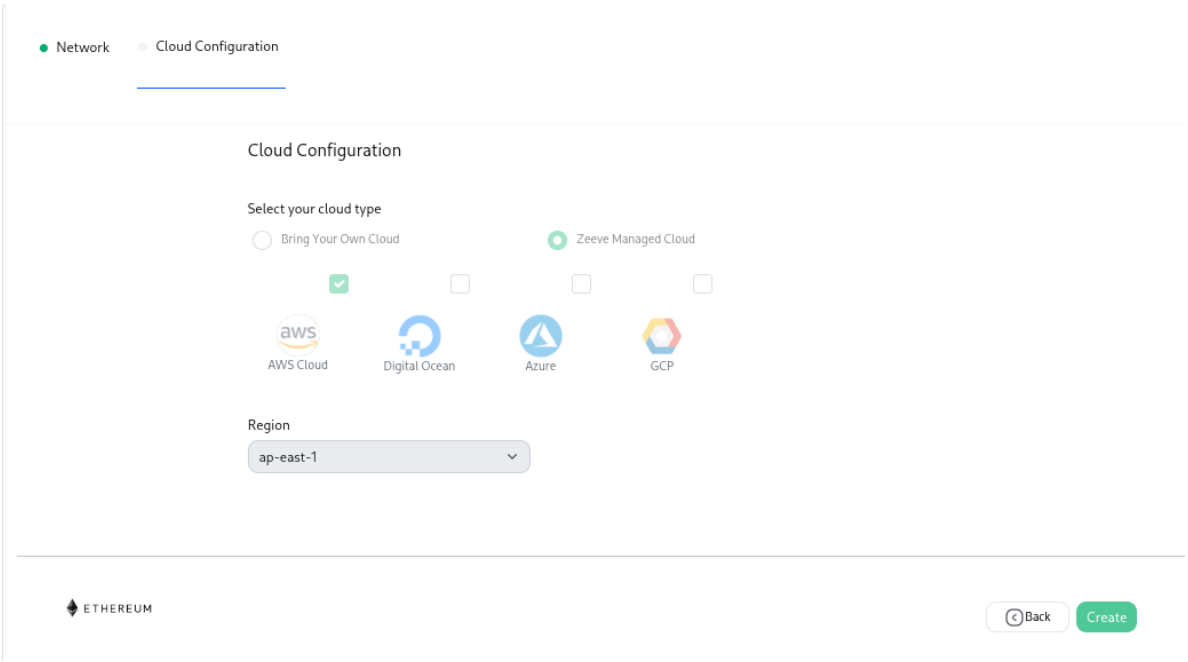
1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.

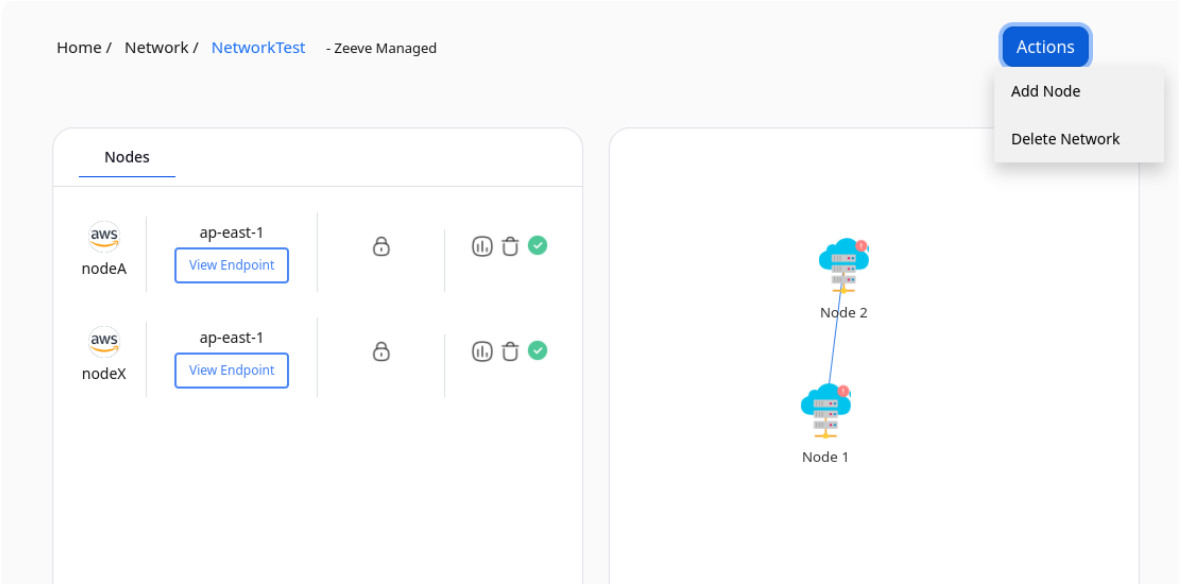


3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region will be prefilled according to the configuration of the network. Click on the **Create** button and it's done!



15.1.3 Delete node in a network

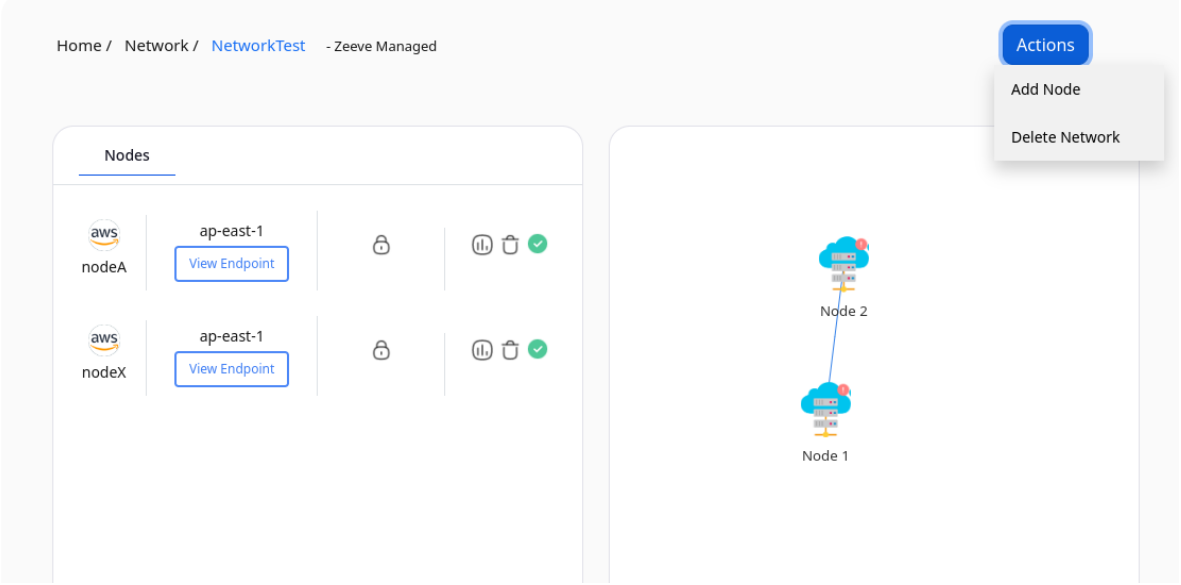
1. Select the network, you want to perform the deleted node to, and click on the network card Ref.. You will get to see similar to the below image.



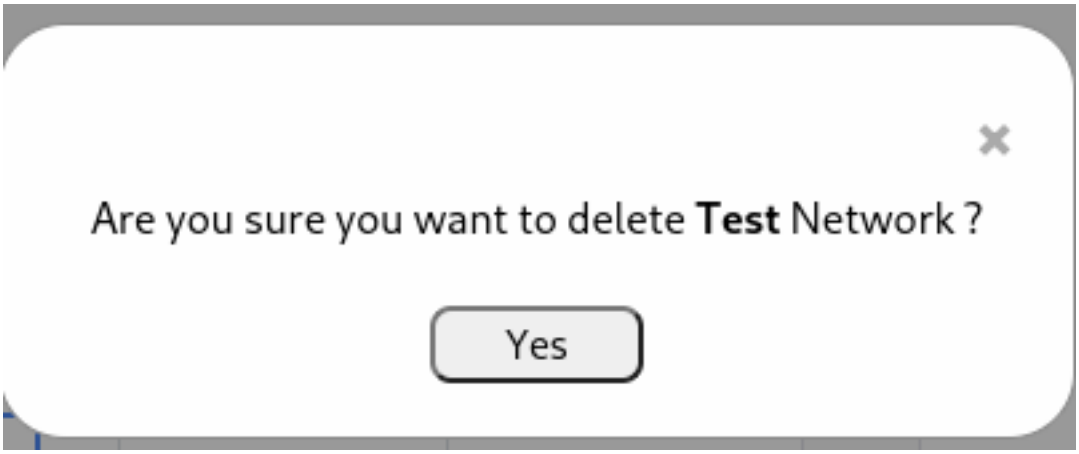
2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **yes** button to confirm.

15.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

---

description: Discover the power of Corda on Zeeve's platform. Our documentation provides an introduction to the key concepts, tools and techniques for building decentralized applications using Corda and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

SIXTEEN

CORDA

1. **Create CLI Access**
    - Navigate to settings and then click on "API Credentials" to see a list of CLI/API credentials.
    - Click "Create key".
    - Provide a name, and select one or more networks to associate with the key.
    - Add appropriate permissions for the operations that the keys are being created for.
    - Click "Create key" and then copy/save generated keys.
2. **Login with Zeeve CLI**
    - Login with Zeeve CLI using the earlier created keys and add these urls as well -

```
zeeve login -i < access-key > -s < secret-key > -ae https://app.zeeve.io/
↪auth/cli/login -ce  https://app.zeeve.io/fabric-backend/chaincode/
```

3. **Deploy**
    - Use the following command to deploy **Corda**

```
zeeve corda corda-deploy -f < cordapp tar file> -n < networkID >
```

---

description: Learn about Avalanche on Zeeve's platform. Our documentation provides an introduction to the key concepts, tools and techniques for building decentralized applications using Avalanche and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

SEVENTEEN

AVALANCHE DEDICATED NODE SETUP

This section will guide you about the different features offered by Zeeve for Avalanche.

1. *Dedicated nodes*
2. *Rpc api endpoints*

17.1 Dedicated nodes

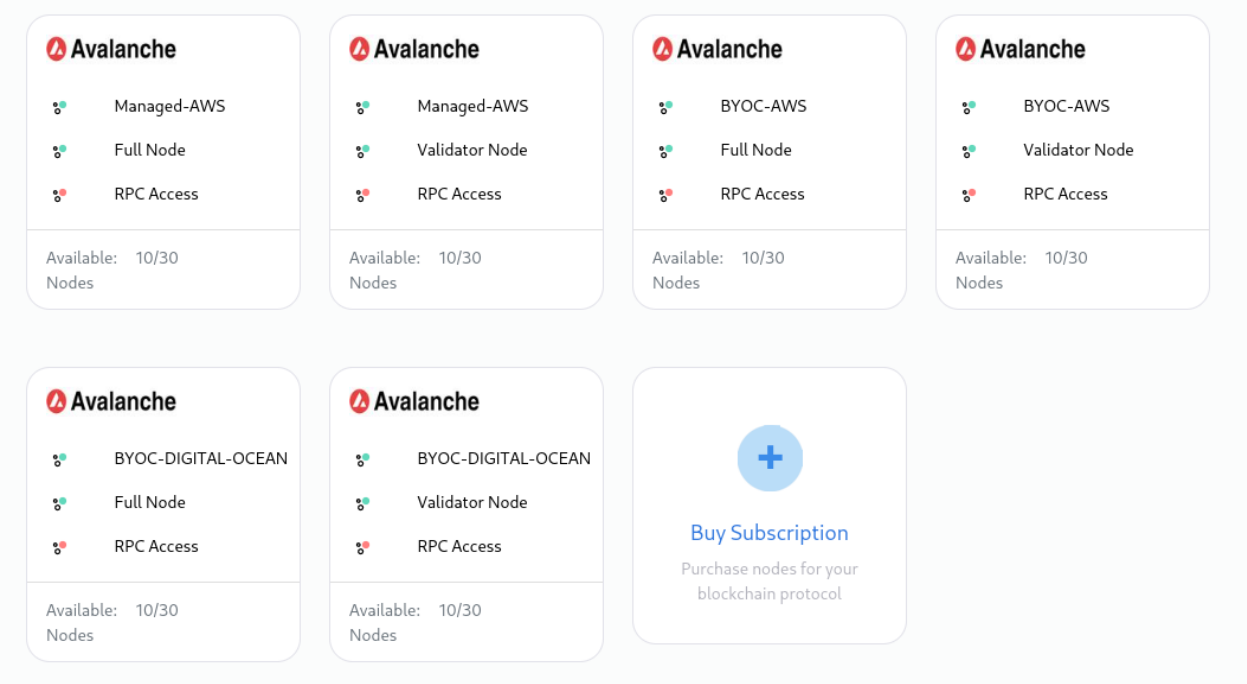
1. *Create a network*
2. *Add a node*
3. *Delete a node*
4. *Delete a network*

17.1.1 Create a network

This section will provide you with detailed steps for creating a network of **Avalanche**.

NOTE Please make sure to follow the *steps* before proceeding.

On the **Network Configuration** page you will have different cards with different network configurations for Avalanche, which looks similar to the image provided below.

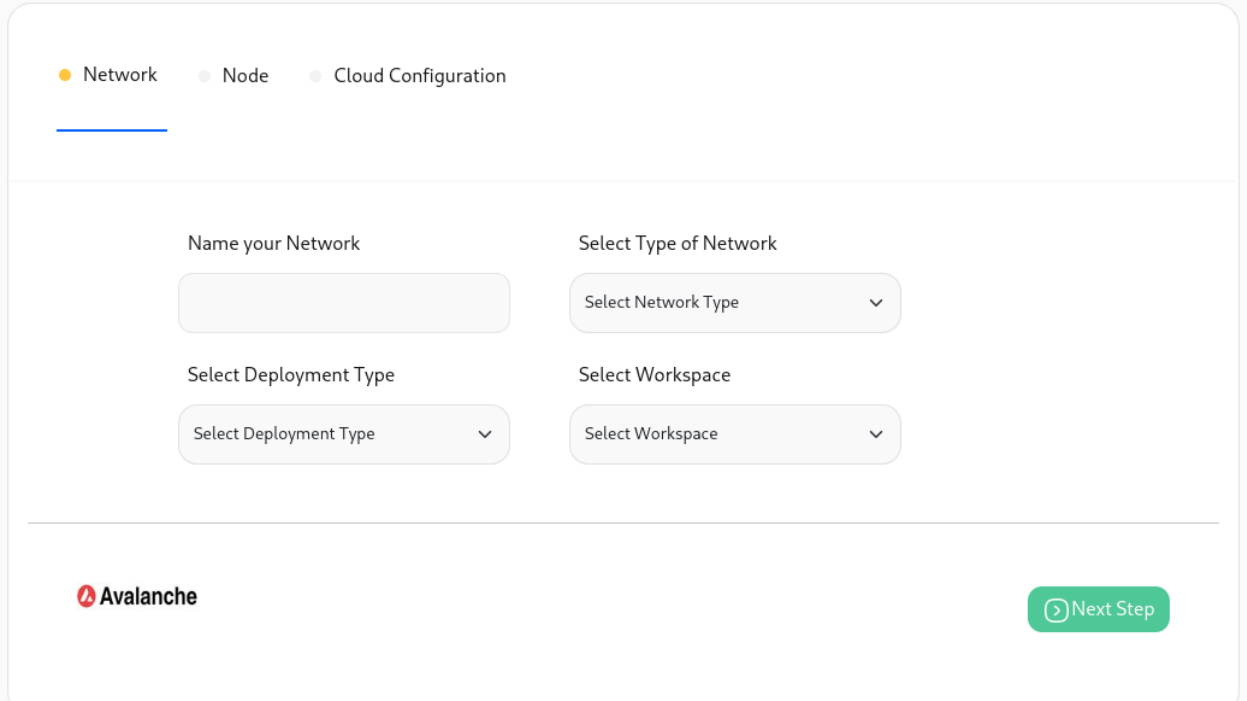


NOTE: *These cards can be different in your case. Card configurations depend on your purchased subscription.*

You can Choose **Managed-AWS** (Zeeve's managed hosting) for the infrastructure of your node or you can use your cloud account (AWS/DO) for the hosting of your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

1. **Network Info** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.

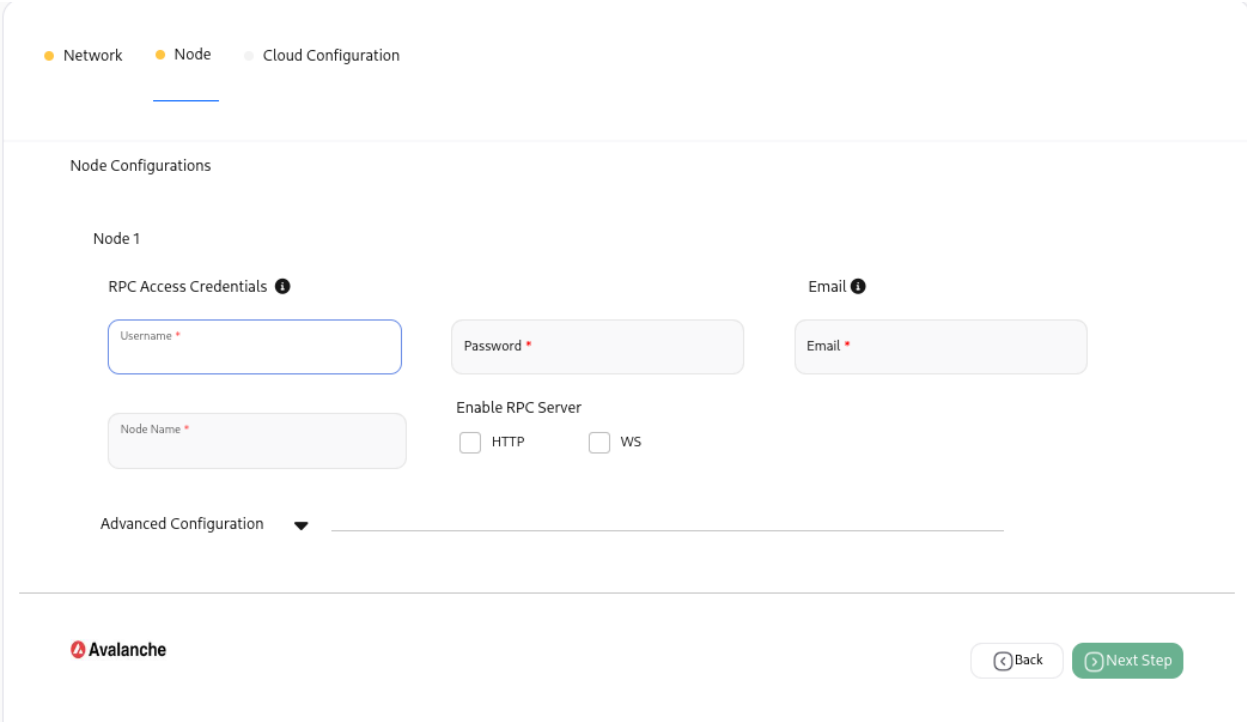


- **Name of Network**: To uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: Deployment type
- **Type Of Network**
  - **MainNet**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade avalanche dapps.
  - **TestNet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after the successful creation.

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration**

In this section, you have to fill in the details of the nodes you want to add to your network.



- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Email**: Fill email of yours.
- **Node Name**: To identify your nodes, this field will be used.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.
- **JSON RPC APIs**: JSON RPC API is a bridge that allows dApps to connect to nodes.

After providing the details click on **Next step** to go to the last step.

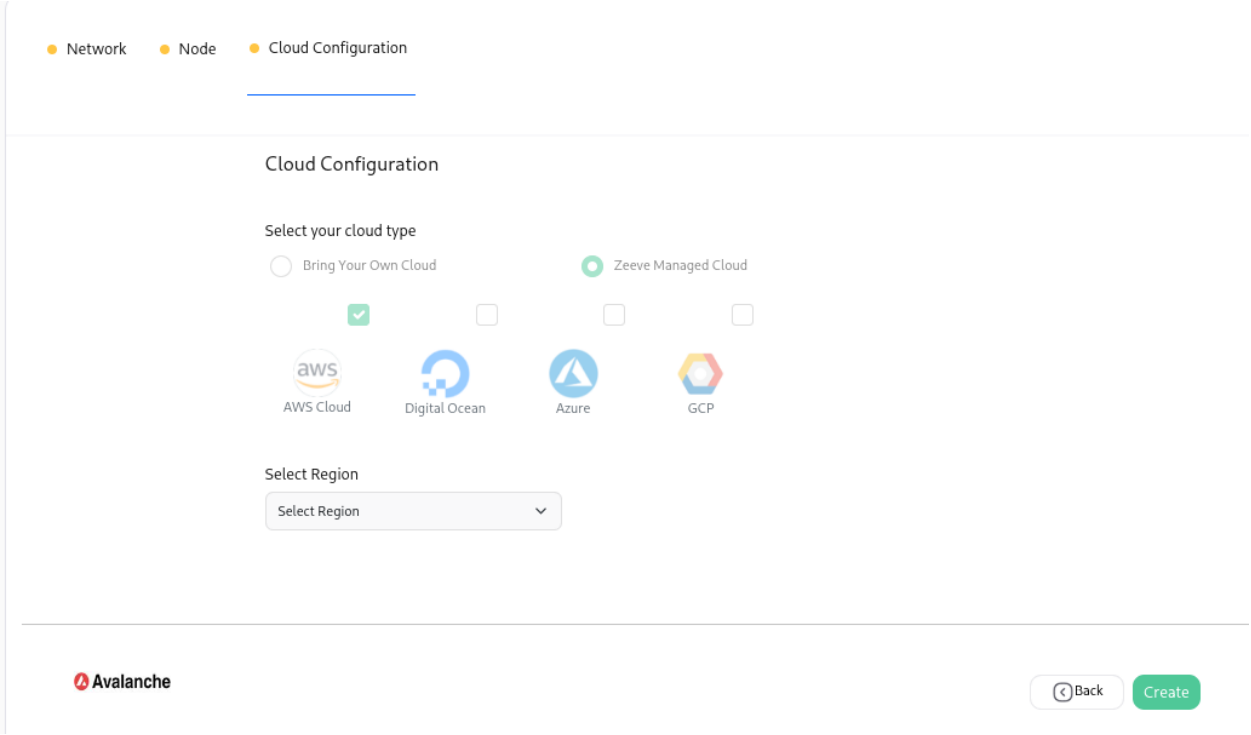
1. **Cloud Configuration**

This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**

1. *Manged - AWS*
2. *BYOC - AWS*
3. *BYOC - DO*

Managed - AWS

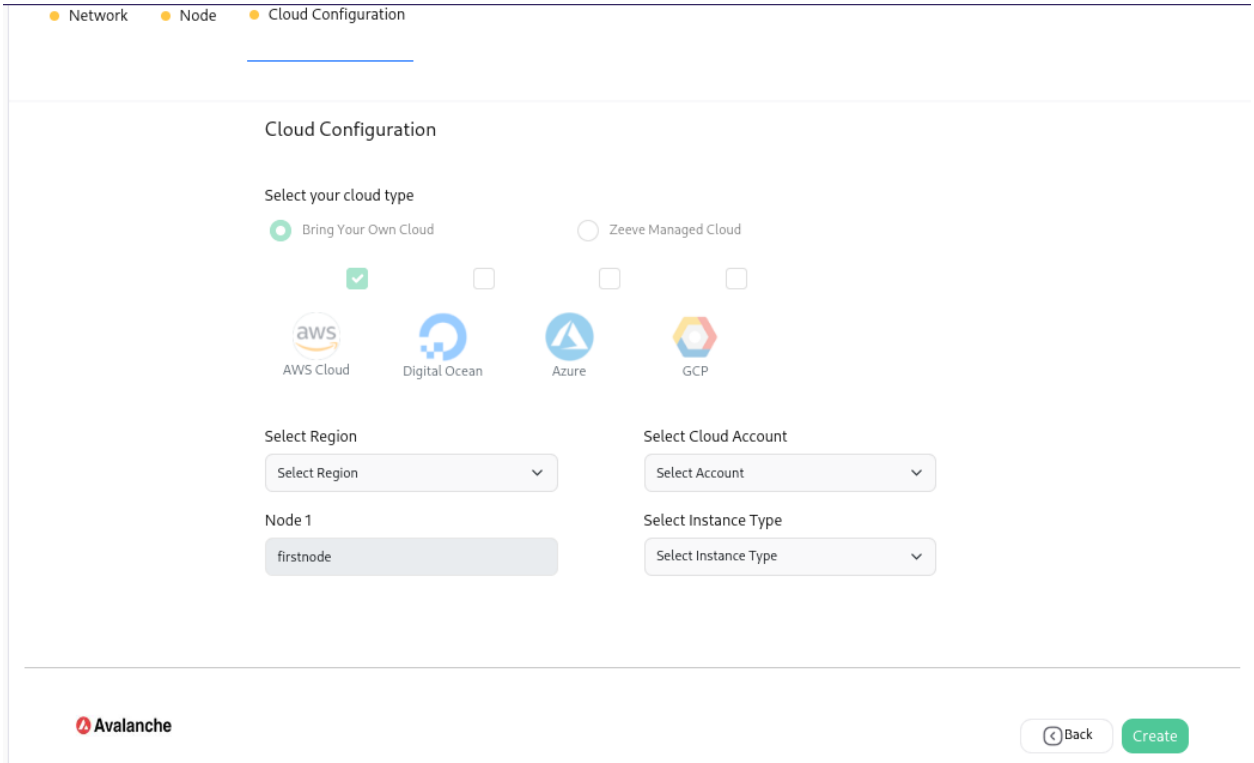
In the case of **Managed - AWS**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.

BYOC - AWS

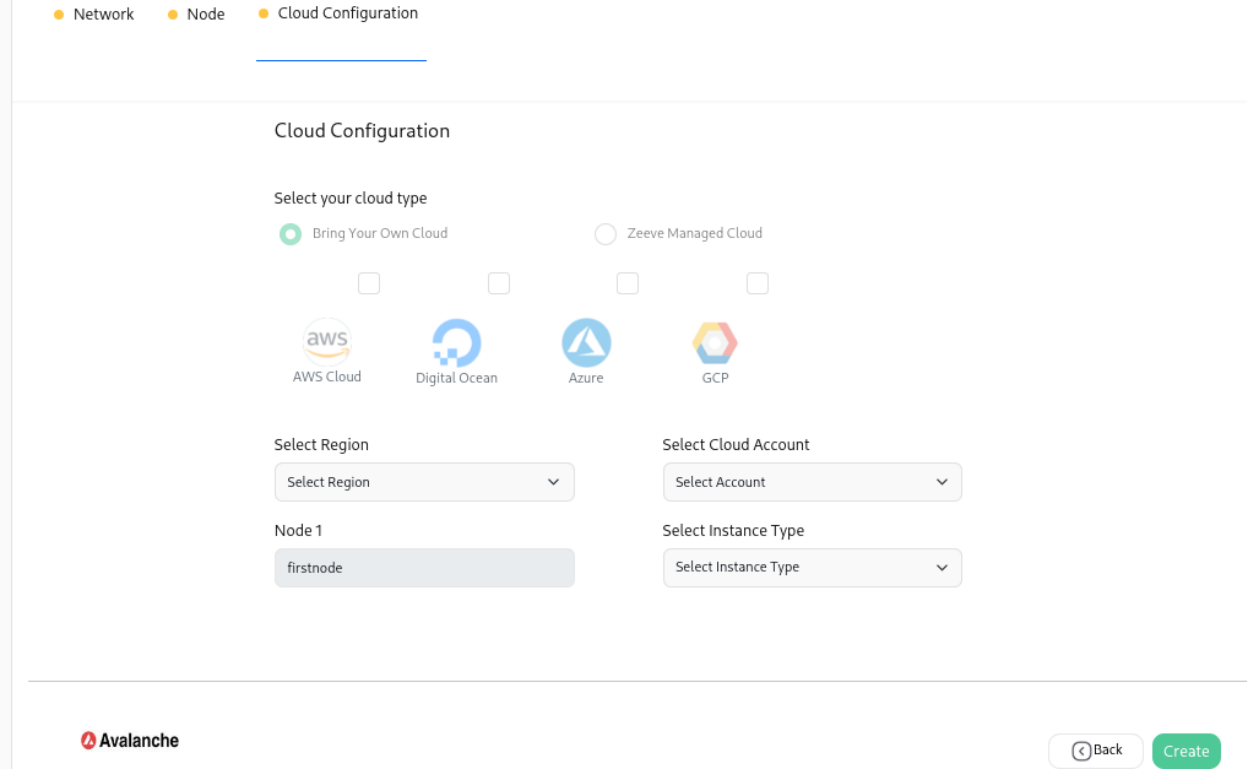
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

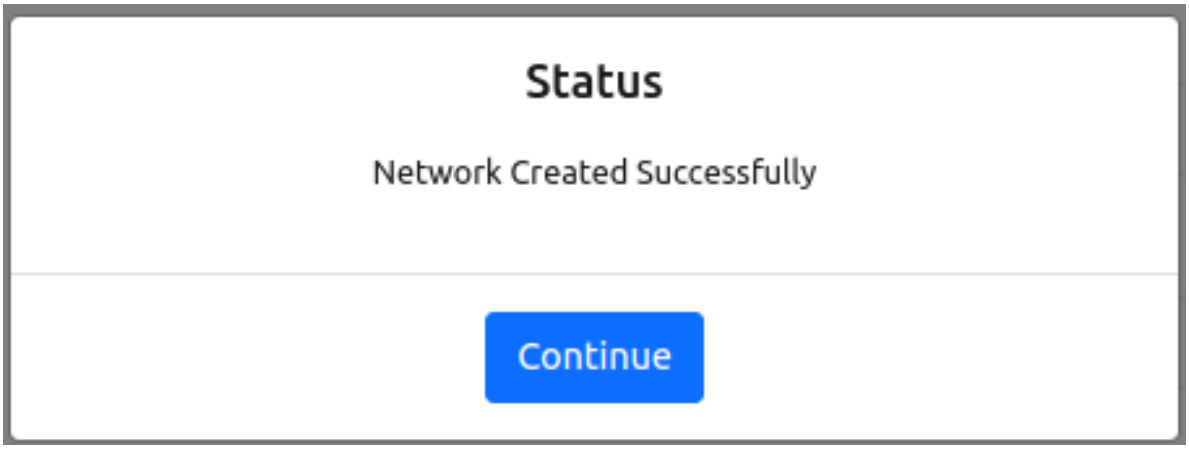
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the Digital Ocean cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.

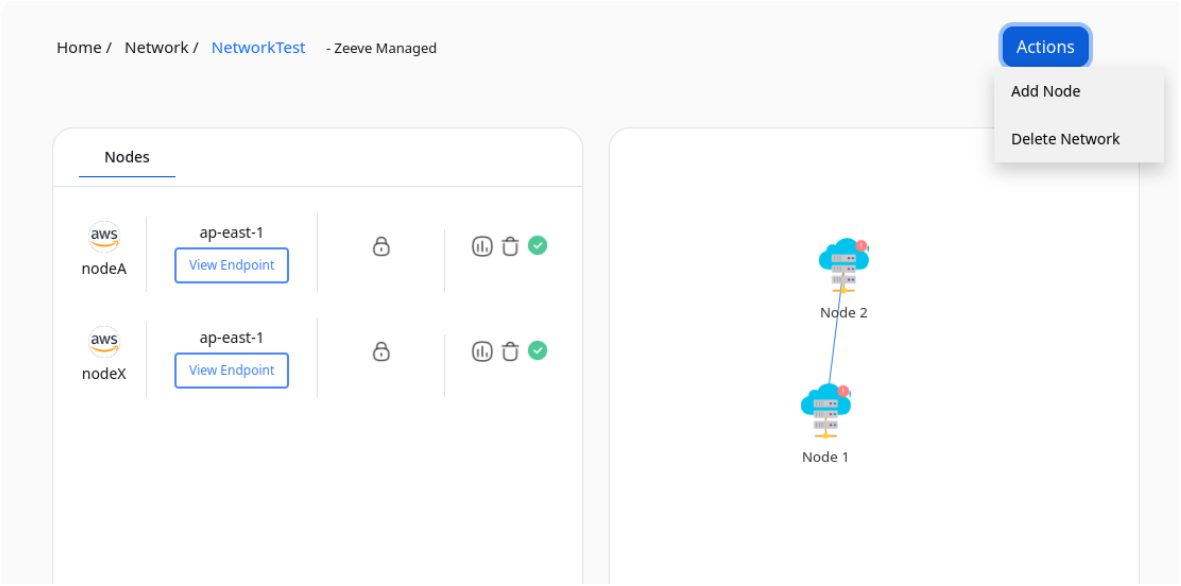


2. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes listed you've just added to the network.

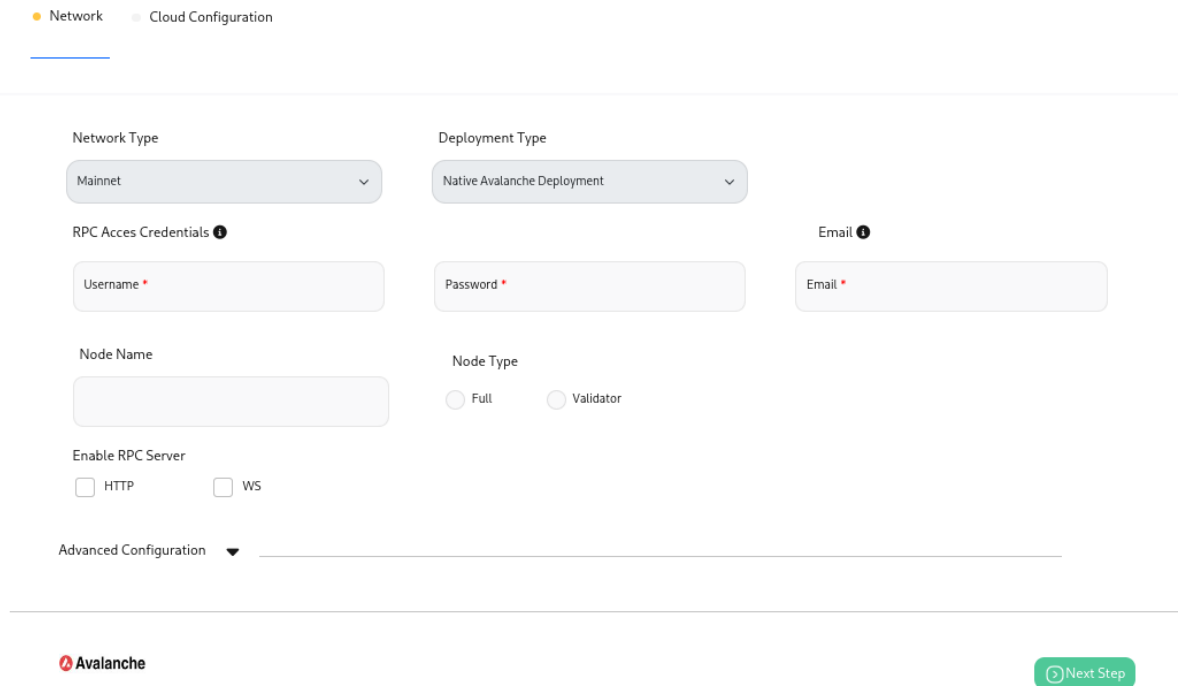
17.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

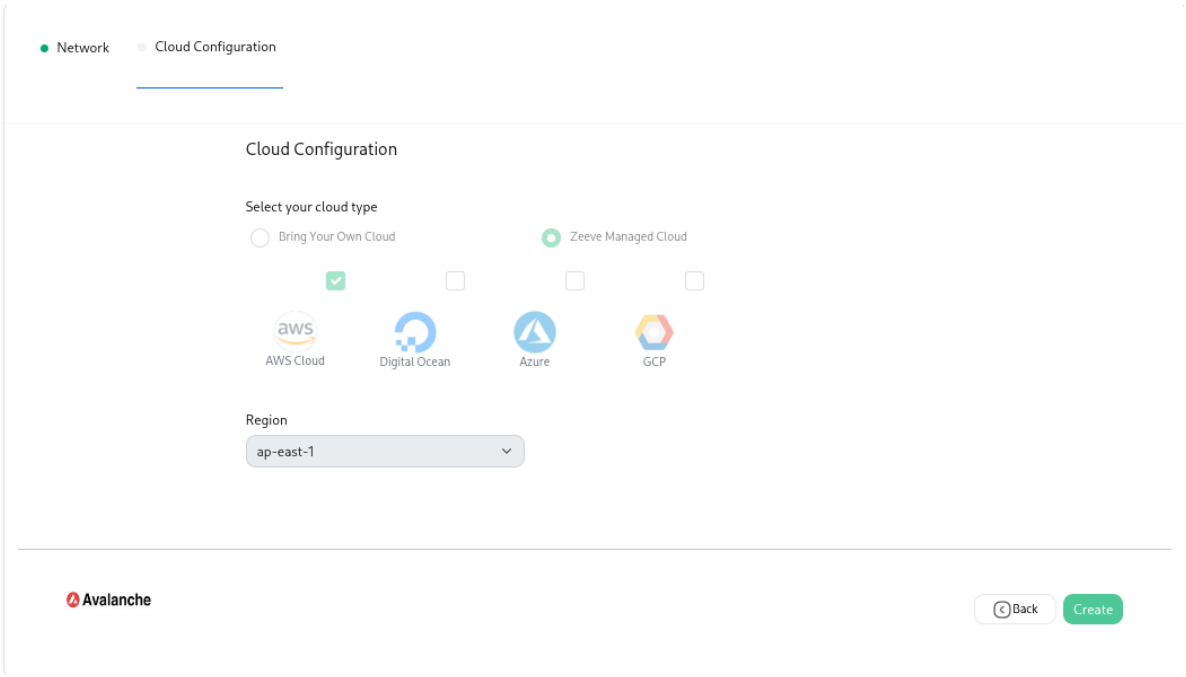
1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



2. Click on the *Actions* button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.

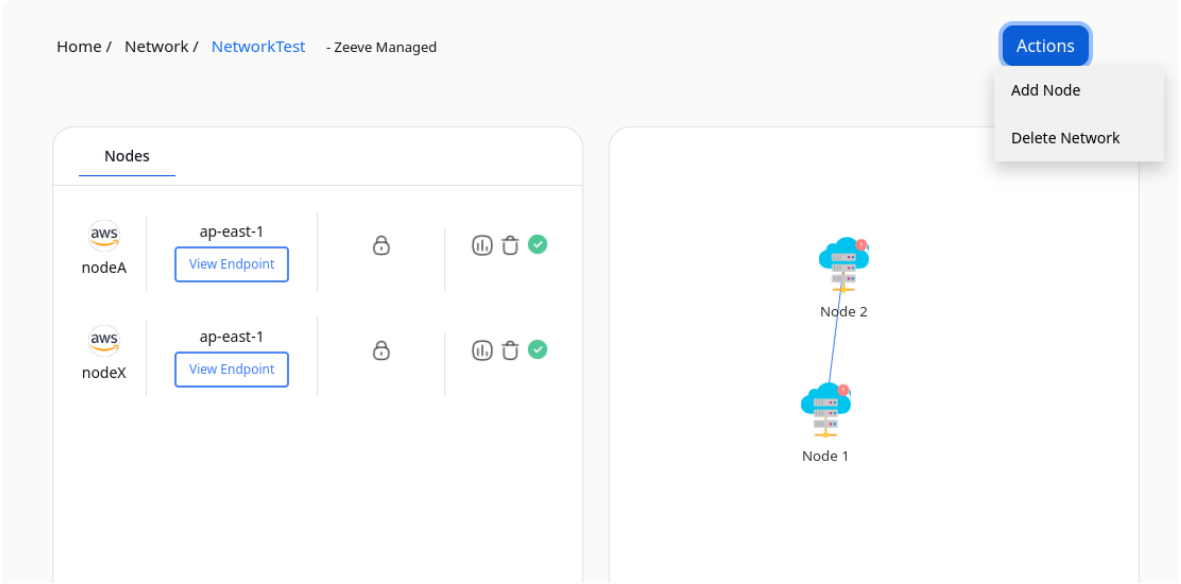


3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region will be prefilled according to the configuration of the network. Click on the **Create** button and it's done!

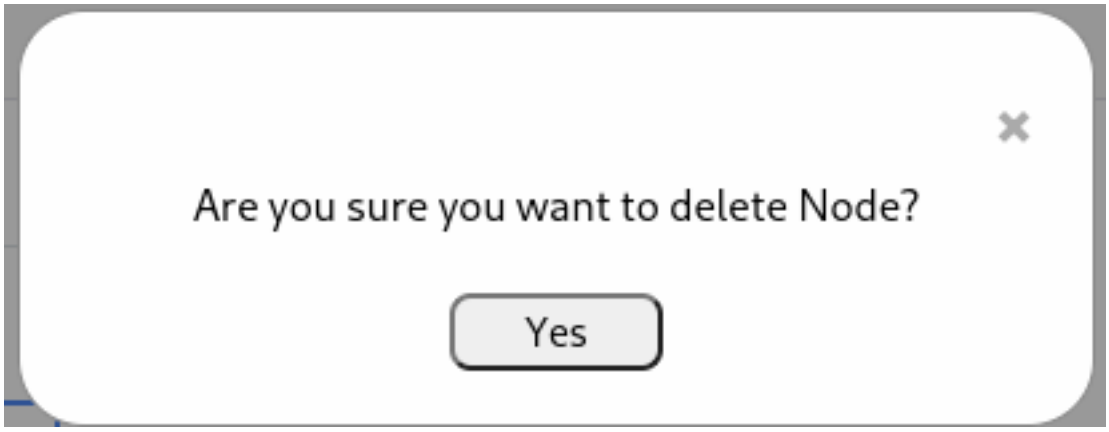


17.1.3 Delete node in a network

1. Select the network, in which you want to perform the delete node action, and click on the network card Ref.. You will get to see similar to the below image.

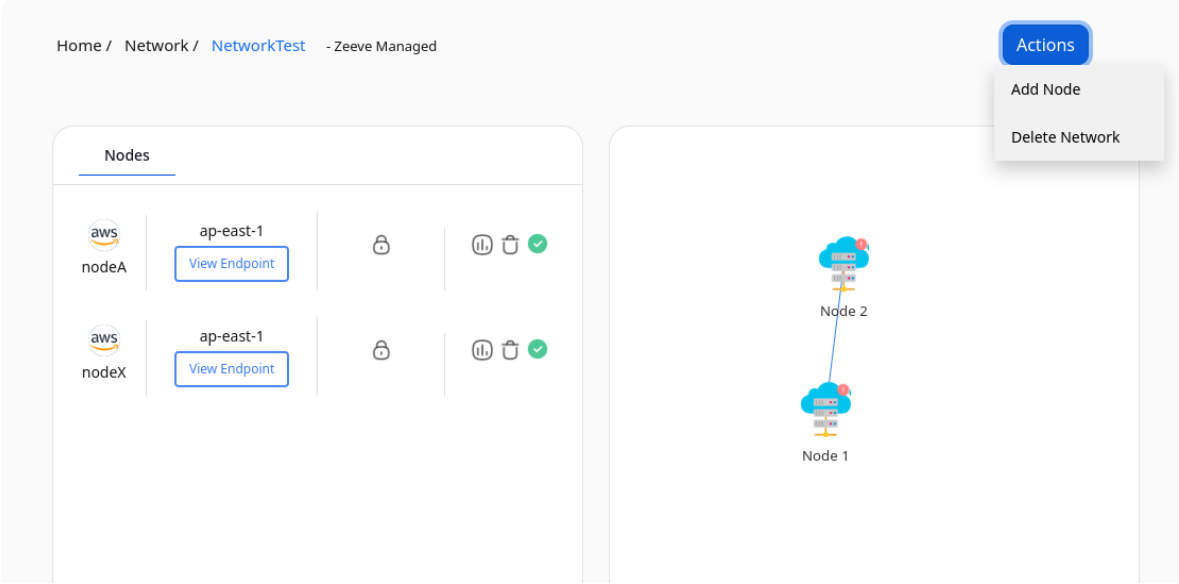


2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **yes** button to confirm.

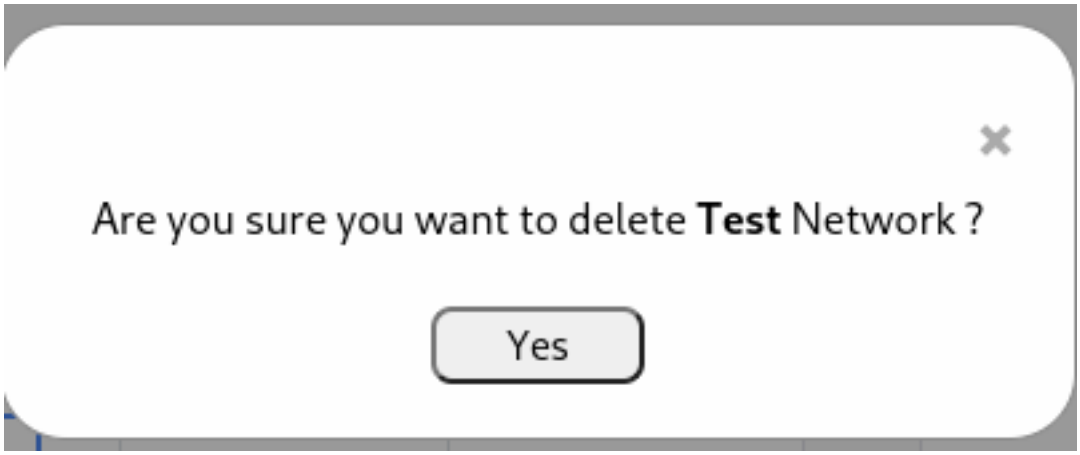


17.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the *Actions* button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

17.2 RPC API endpoints

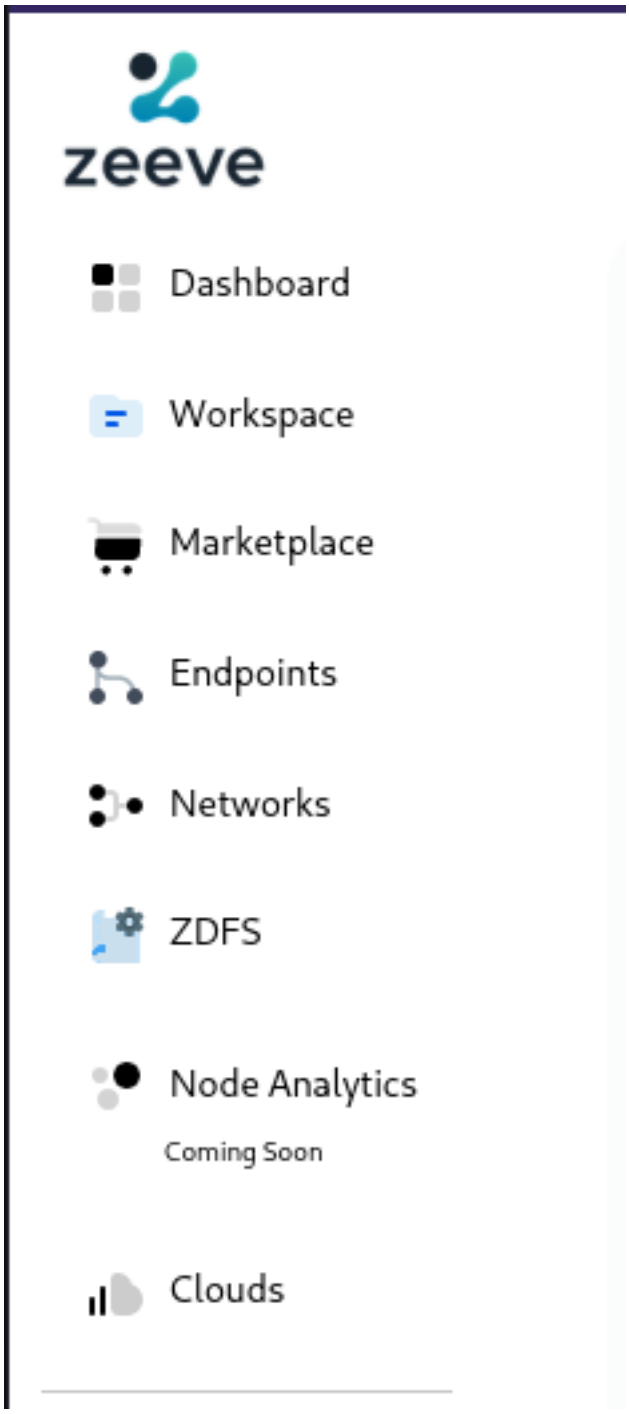
This section talks about different actions you can perform in Avalanche RPC API endpoints.

1. *Create endponits*
2. *Edit endpoints*
3. *Delete endpoint*

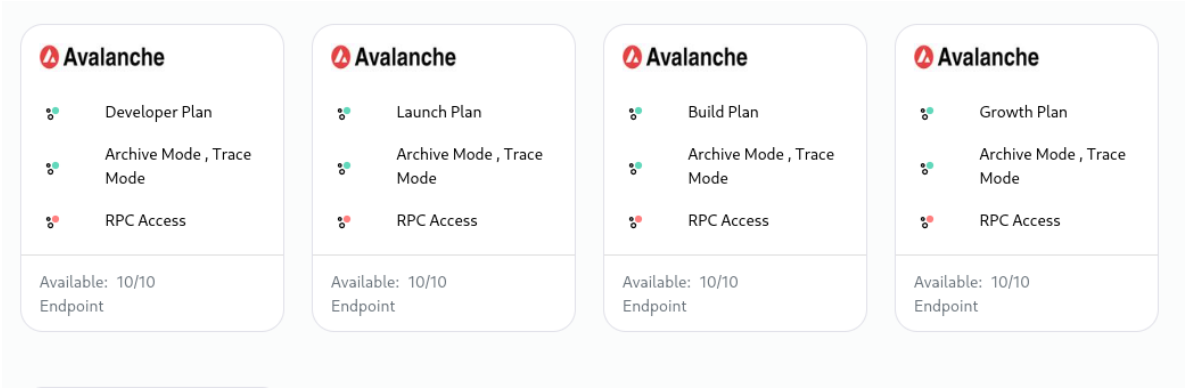
17.2.1 Create endpoint

This section provides a step-by-step guide for creating avalanche RPC API endpoints.

1. Click on **Endpoints** on the left side navigation bar and then click on **Add Endpoints**.

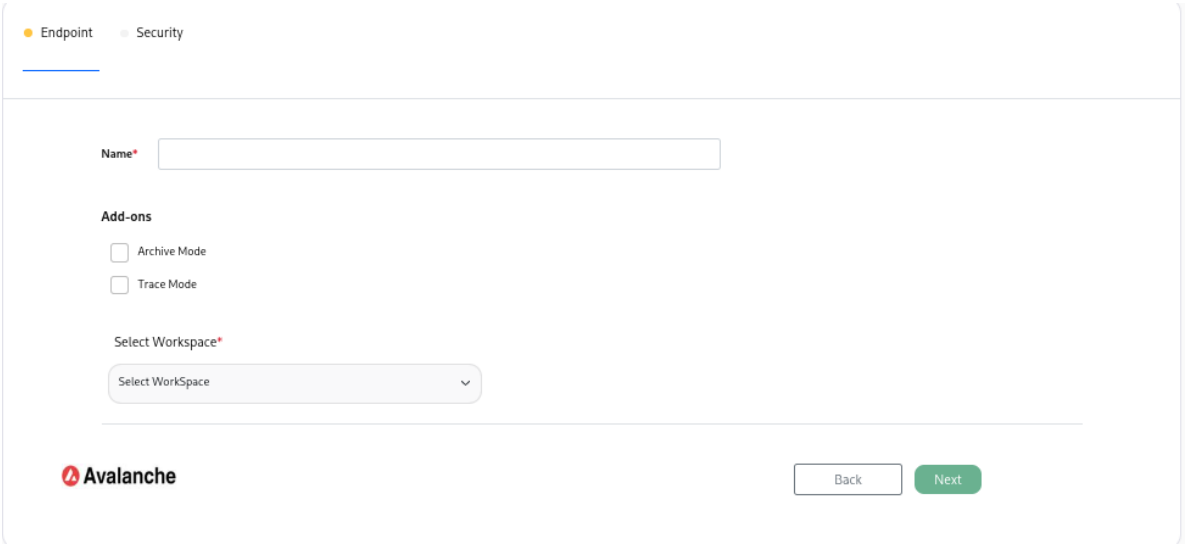


2. You will be landed on a page similar to the below image.

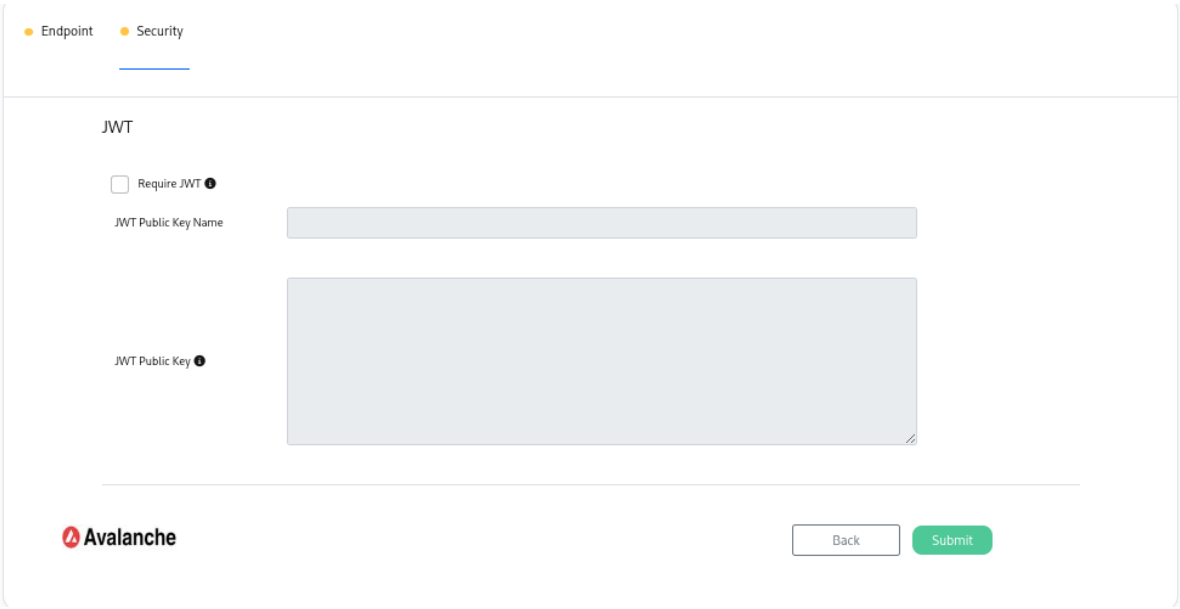


NOTE It can take be different in your case. These configuration cards depend on your subscription.

1. Select the configuration and click on the card you want. You will be landed on a page similar to the image provided below. Fill in the required details and click **next**.



2. In this step you can check the **Require JWT** and provide JWT public key details and click on **Submit**.



17.2.2 Edit endpoint

17.2.3 Delete endpoint

---

description: Integrate Binance with Zeeve's platform using our API and tools. Our documentation provides technical details on authentication, making requests and handling responses for interacting with Binance and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

EIGHTEEN

BINANCE DEDICATED NODE SETUP

This section will guide you about the different features offered by Zeeve for Binance.

1. *Dedicated nodes*
2. *Api endpoints*

18.1 Dedicated nodes

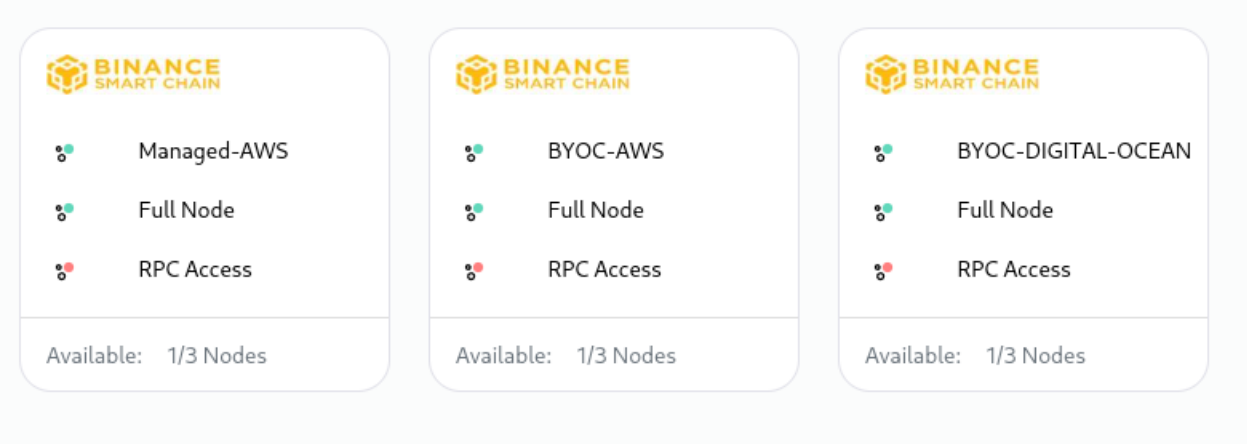
1. *Create network*
2. *Add a node*
3. *Delete a node*
4. *Delete network*

18.1.1 Create a network

This section will provide you with detailed steps for creating a network of **Binance**.

NOTE Please make sure to follow the steps mentioned earlier before proceeding.

On the **Network Configuration** page you will have different cards with different network configurations for Binance, which looks similar to the image provided below.

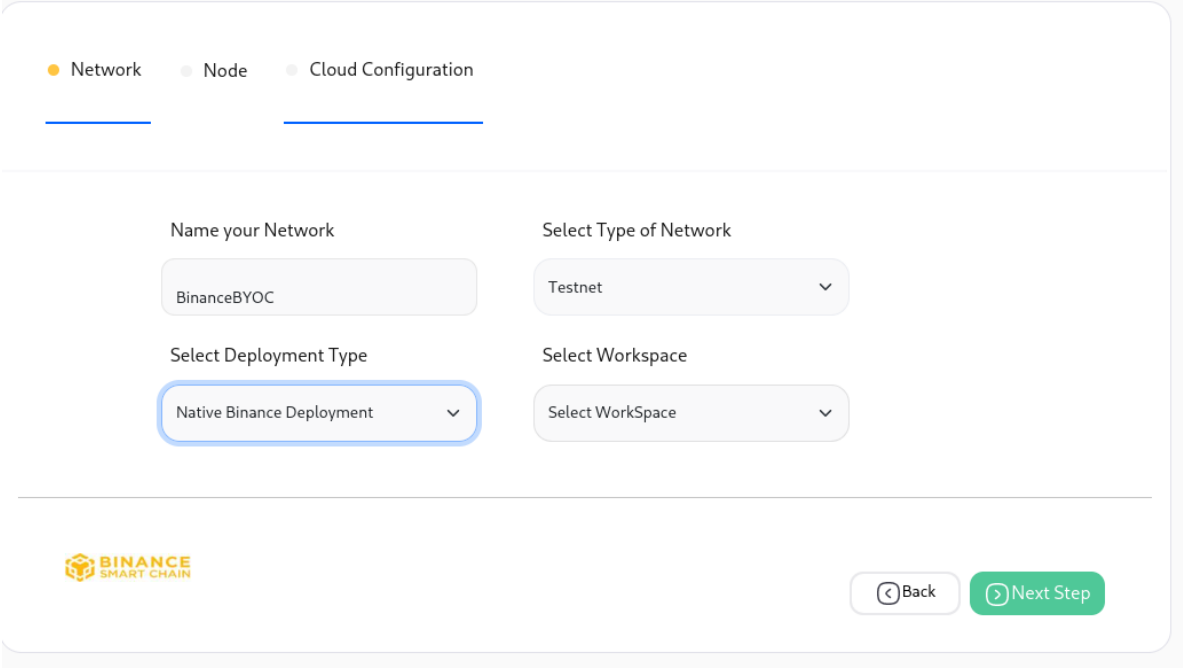


NOTE: These cards can be different for your case. Card configurations depend on your purchased subscription.

You can Choose **Managed-AWS** (Zeeve's managed hosting) for the infrastructure of your node or you can use your cloud account (AWS/DO) for the hosting of your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

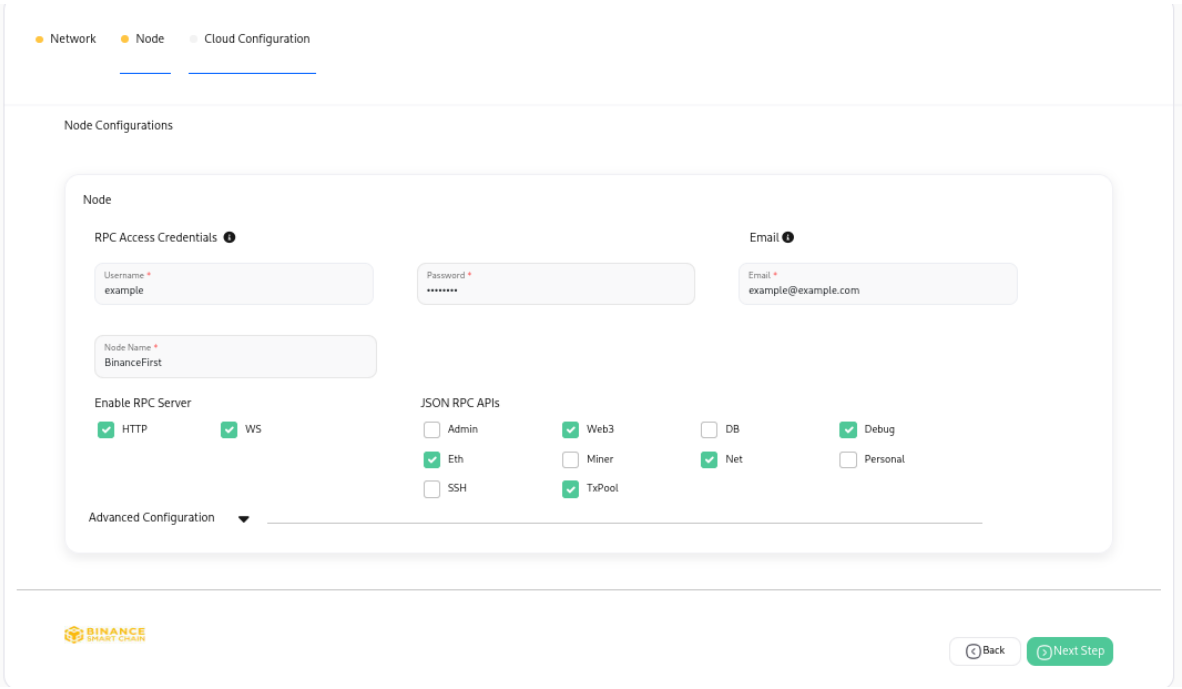
1. **Network Info:** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.



- **Name of Network**: To uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: Deployment type
- **Type Of Network**
  - **MainNet**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade Binance dapps.
  - **TestNet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after successful creation.

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration:** In this section, you have to fill in the details of the nodes you want to add to your network.



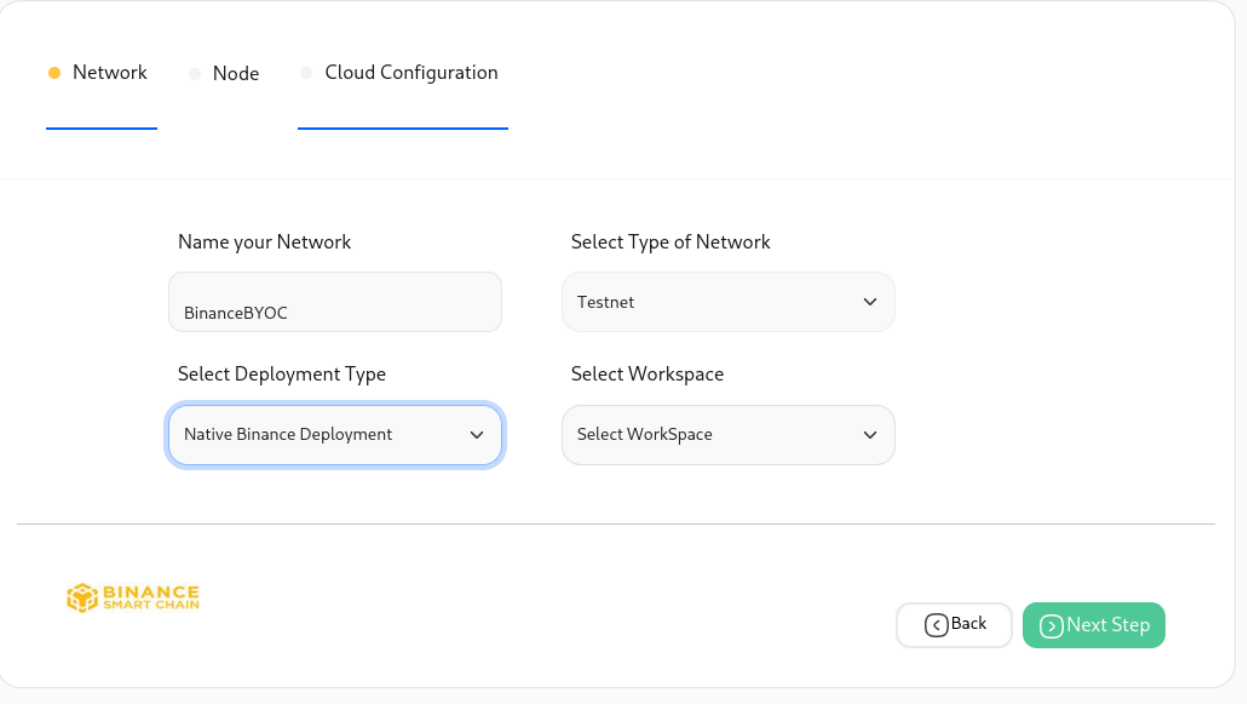
- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Email**: Fill email of yours.
- **Node Name**: To identify your nodes, this field will be used.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.
- **JSON RPC APIs**: JSON RPC API is a bridge that allows dApps to connect to nodes.

After providing the details click on **Next step** to go to the last step.

1. **Cloud Configuration:** This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**
    1. *Manged - AWS*
    2. *BYOC - AWS*
    3. *BYOC - DO*

Managed - AWS

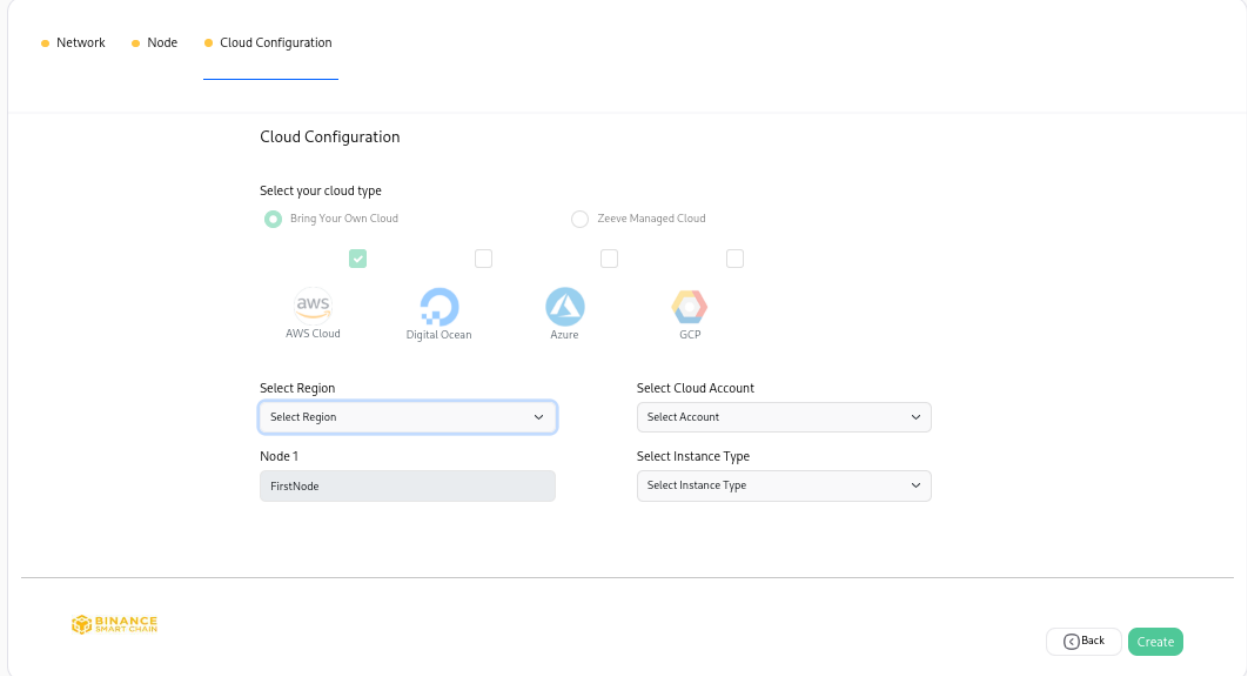
In the case of **Managed - AWS**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.

BYOC - AWS

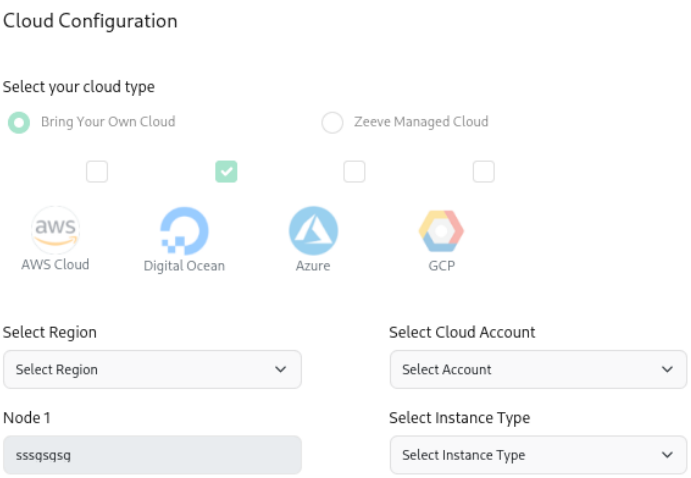
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

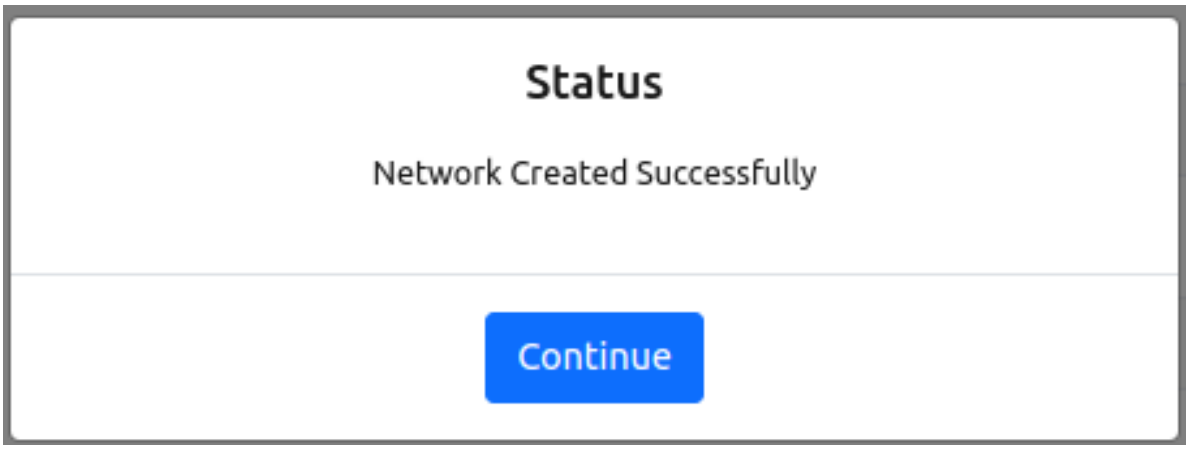
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.

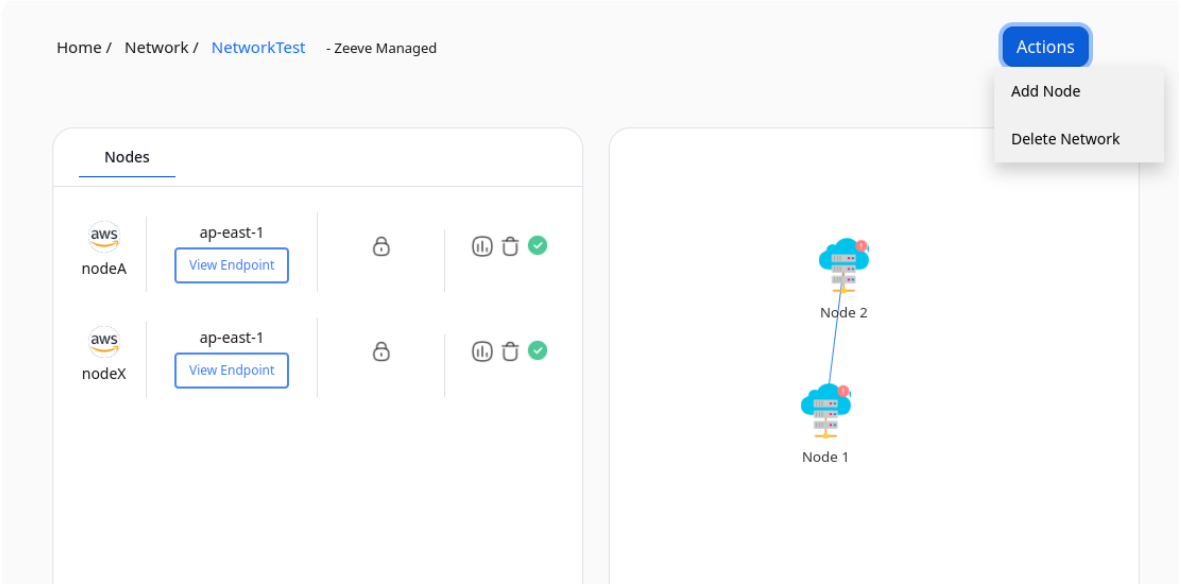


2. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes listed you've just added to the network.

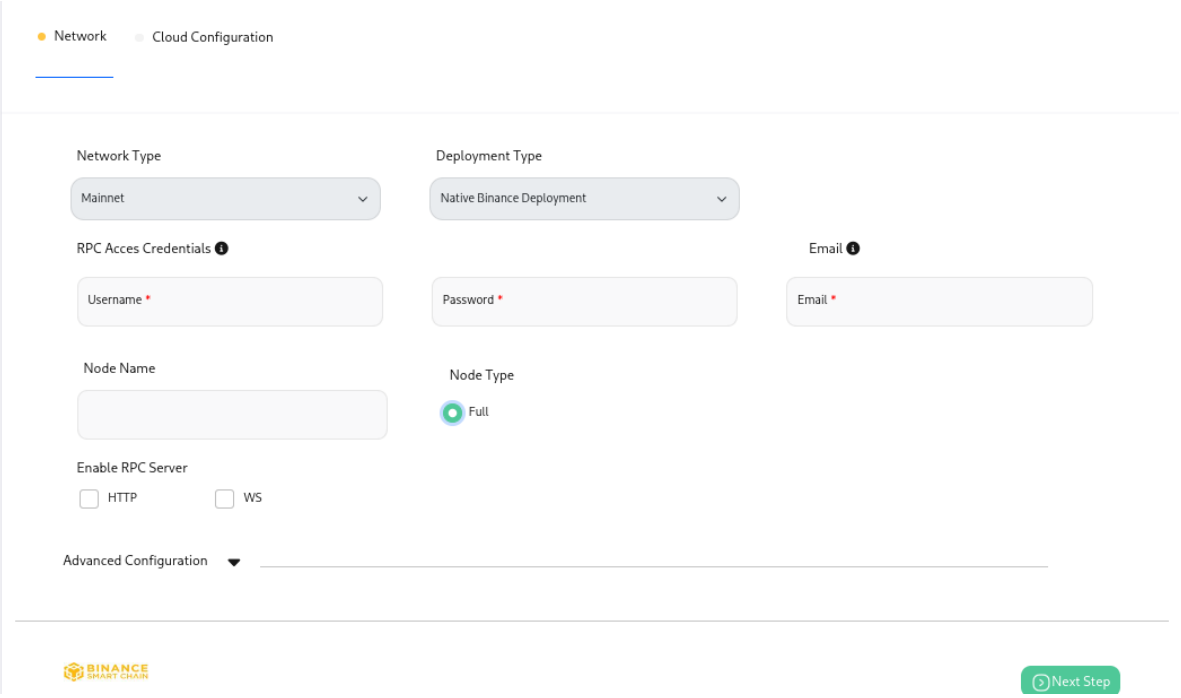
18.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

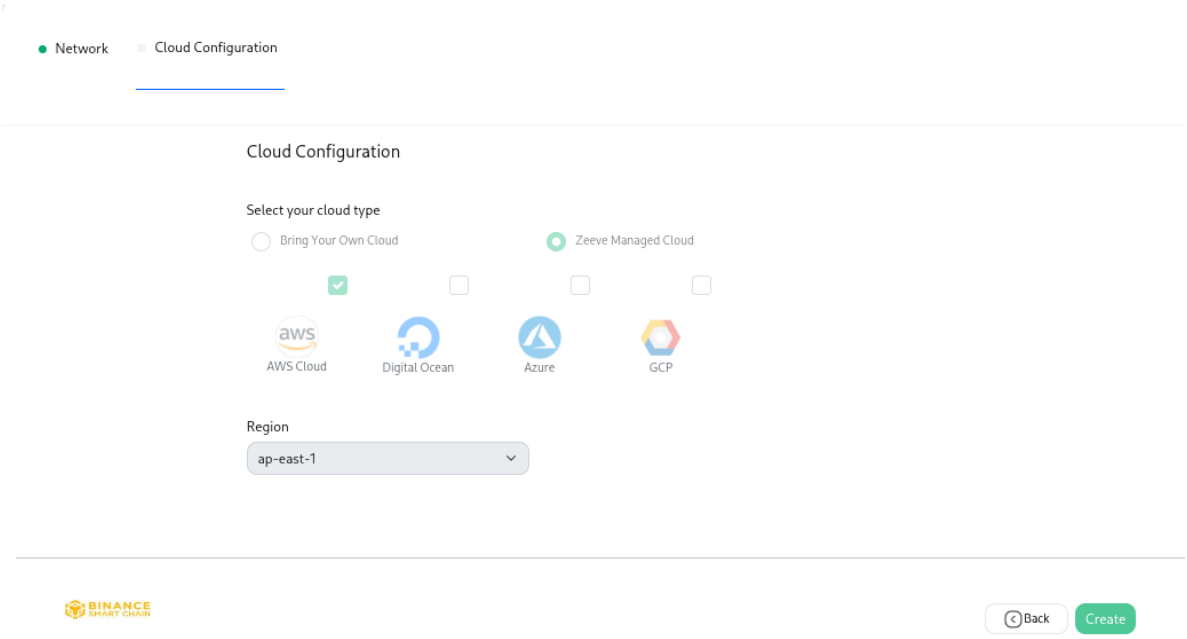
1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.

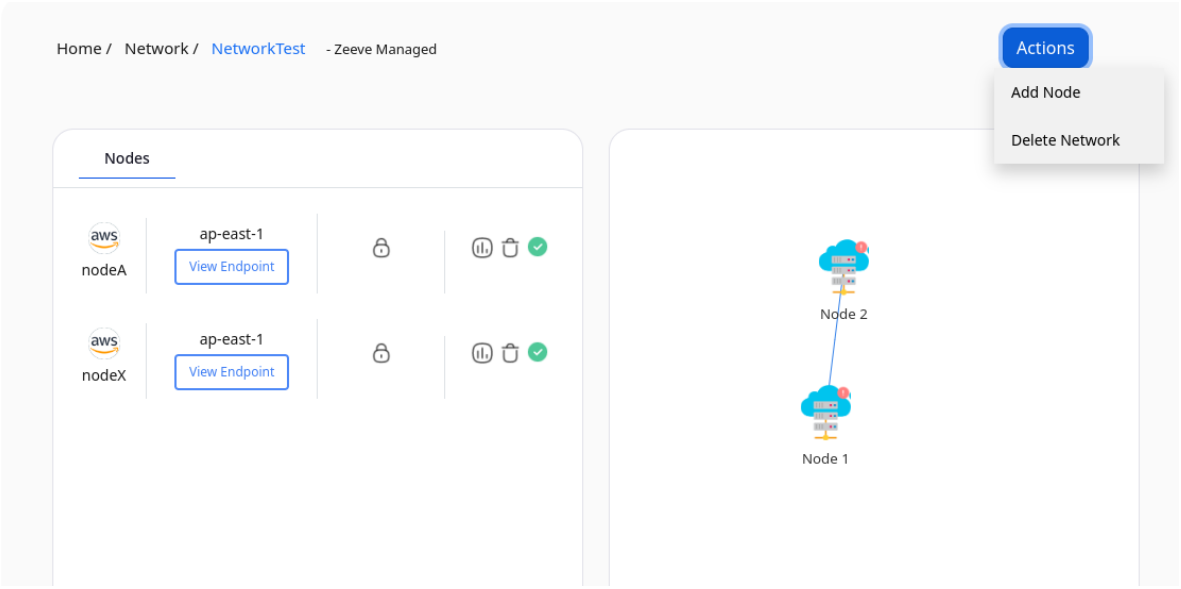


3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region will be prefilled according to the configuration of the network. Click on the **Create** button and it's done!

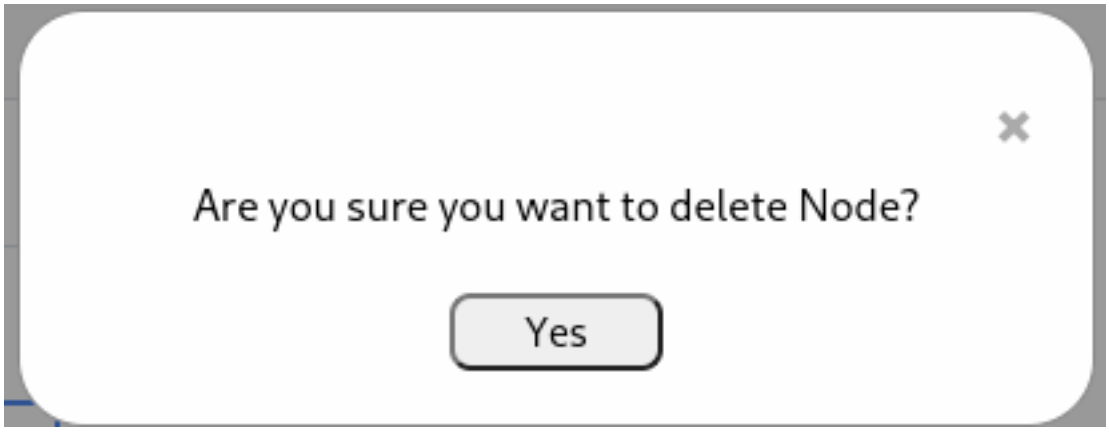


18.1.3 Delete node in a network

1. Select the network, in which you want to perform the delete node action, and click on the network card Ref.. You will get to see similar to the below image.

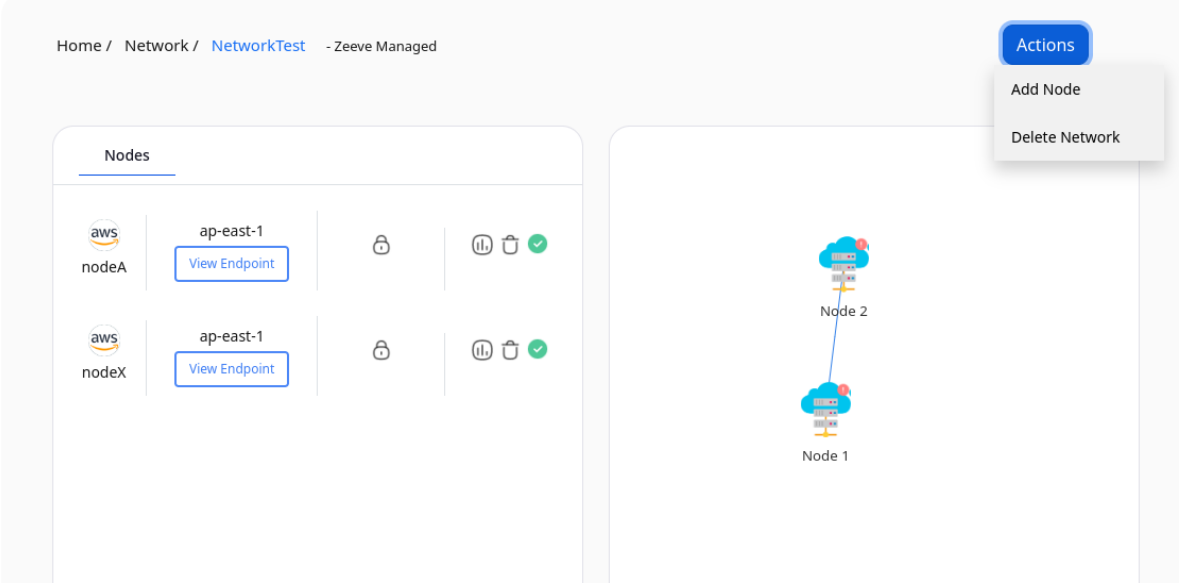


2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **yes** button to confirm.

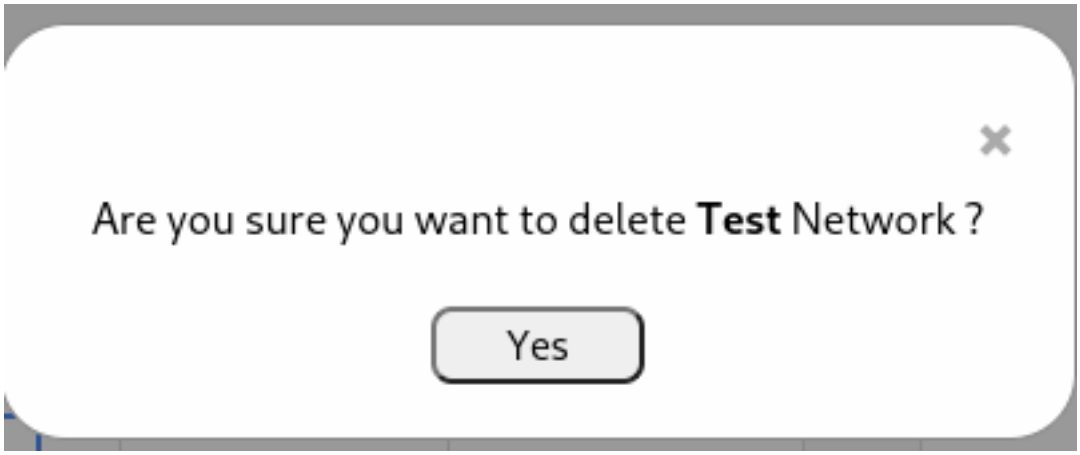


18.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the *Actions* button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

18.2 Api endpoints

CHAPTER

NINETEEN

DESCRIPTION: INTEGRATE COREUM WITH ZEEVE'S PLATFORM USING OUR API AND TOOLS. OUR DOCUMENTATION PROVIDES TECHNICAL DETAILS ON AUTHENTICATION, MAKING REQUESTS AND HANDLING RESPONSES FOR INTERACTING WITH COREUM AND THE ZEEVE PLATFORM.

CHAPTER

TWENTY

COREUM NODE SETUP

This section will guide you about the different features offered by Zeeve for Coreum.

1. *Dedicated nodes*
2. *Staking nodes*

20.1 Dedicated nodes

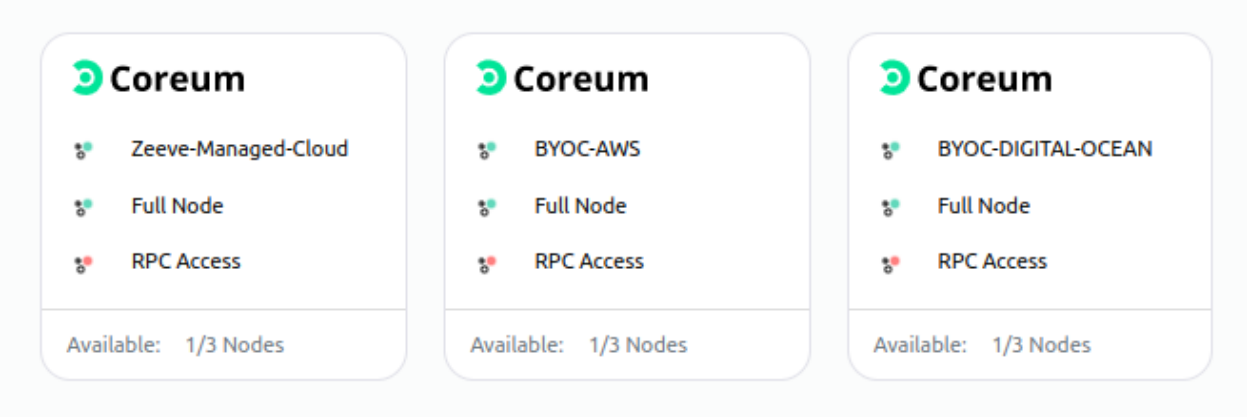
1. *Create network*
2. *Add a node*
3. *Delete a node*
4. *Delete network*

20.1.1 Create a network

This section will provide you with detailed steps for creating a network of **Coreum**.

NOTE Please make sure to follow the steps mentioned earlier before proceeding.

On the **Network Configuration** page you will have different cards with different network configurations for Coreum, which looks similar to the image provided below.

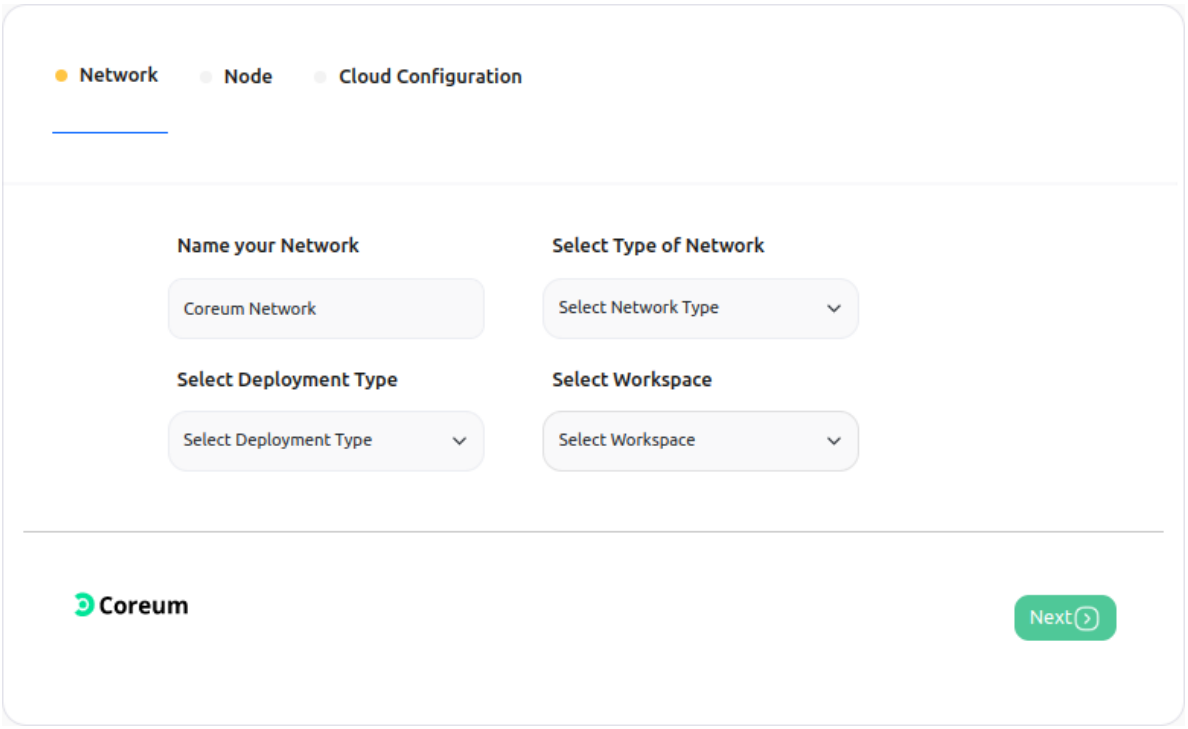


NOTE: These cards can be different for your case. Card configurations depend on your purchased subscription.

You can Choose **Zeeve-Managed-Cloud** (Zeeve's managed hosting) for the infrastructure of your node or you can use your cloud account (AWS/DO) for the hosting of your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

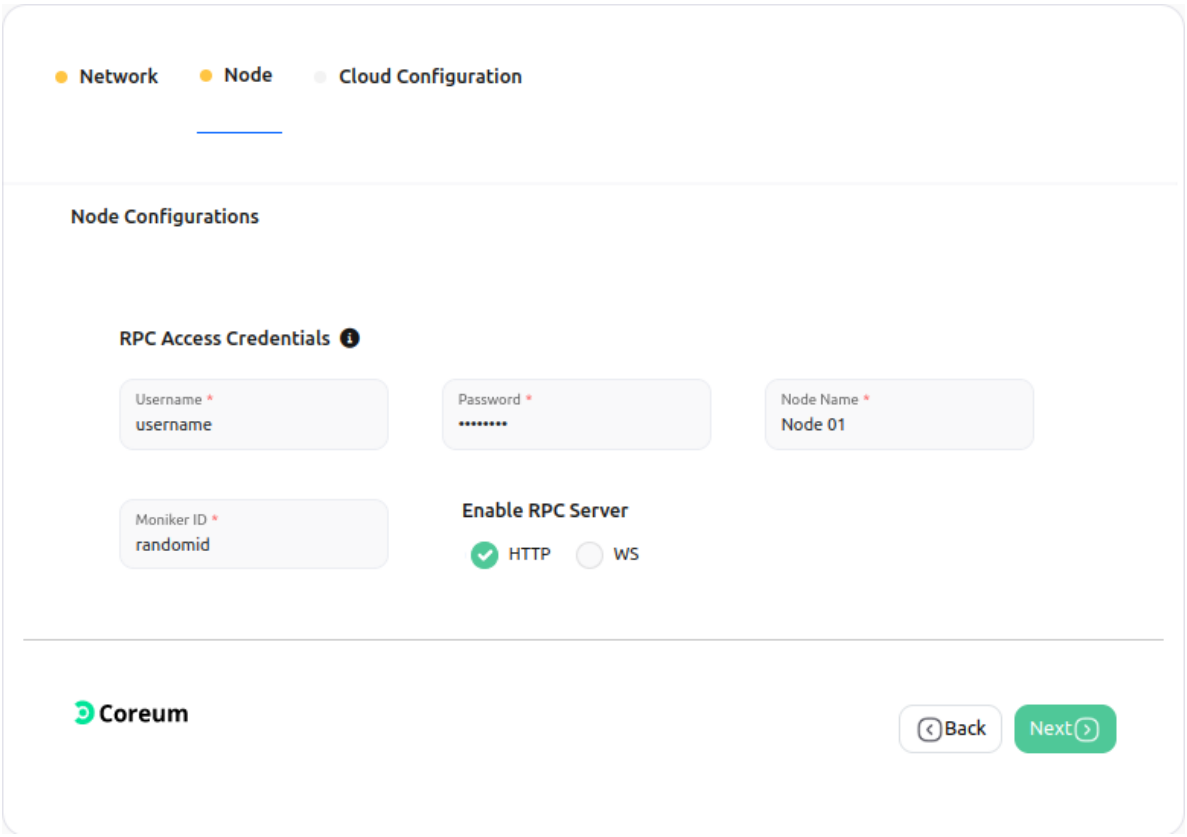
1. **Network Info:** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.



- **Name of Network**: To uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: Deployment type
- **Type Of Network**
  - **TestNet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after successful creation.

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration:** In this section, you have to fill in the details of the nodes you want to add to your network.



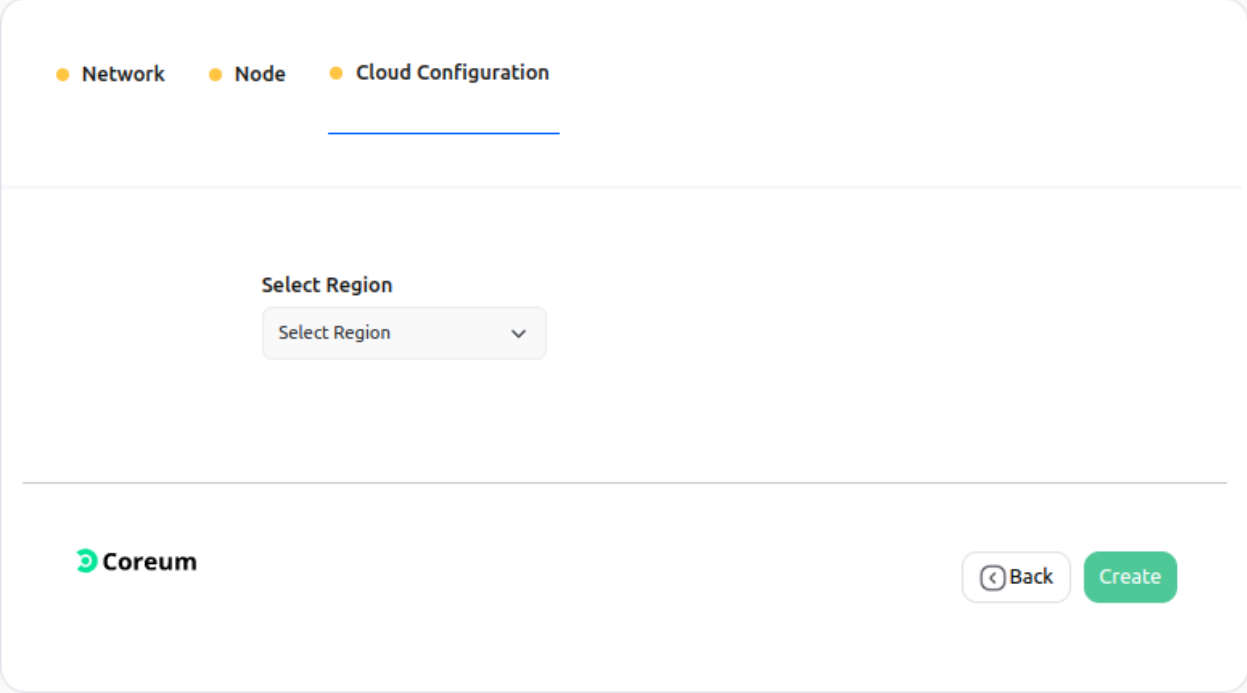
- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Node Name**: To identify your nodes, this field will be used.
- **MonikerId**: A unique ID is required.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.

After providing the details click on **Next step** to go to the last step.

1. **Cloud Configuration:** This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**
    1. *Zeeve Managed*
    2. *BYOC - AWS*
    3. *BYOC - DO*

Zeeve Managed

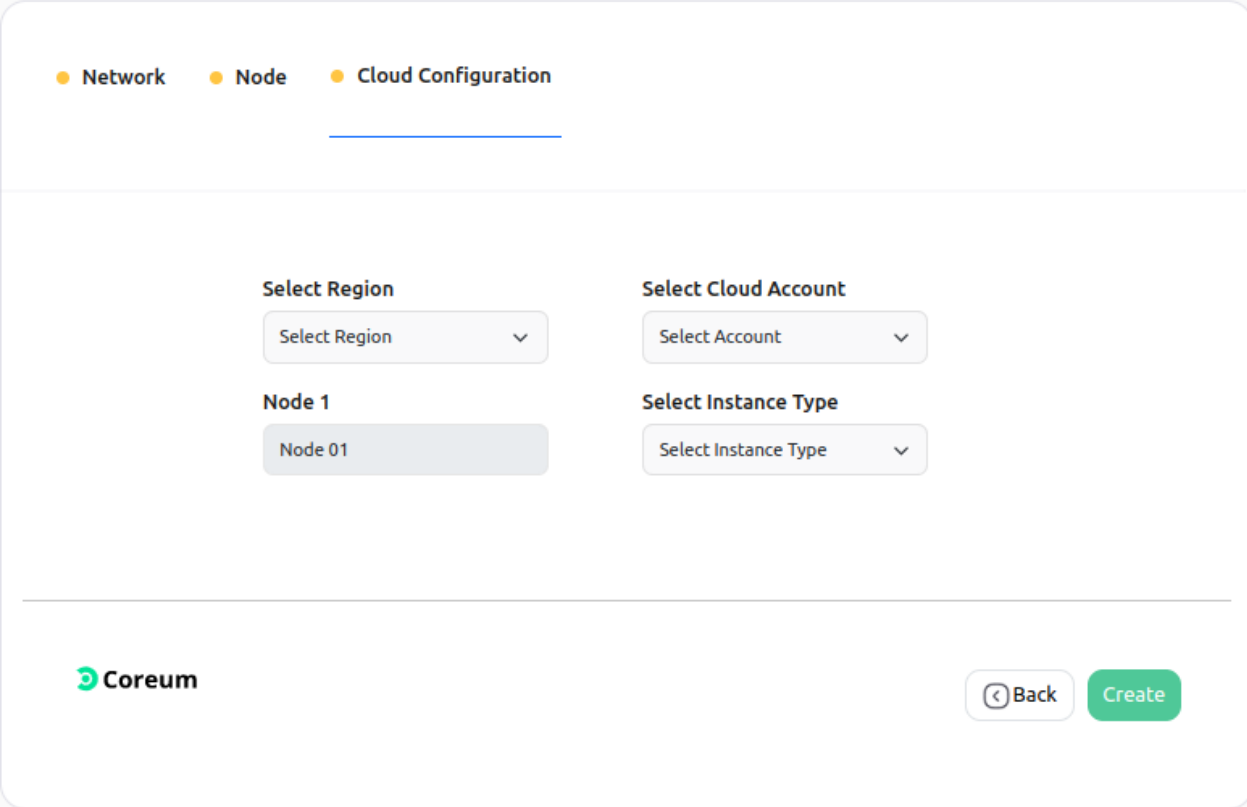
In the case of **Zeeve Managed**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted.

BYOC - AWS

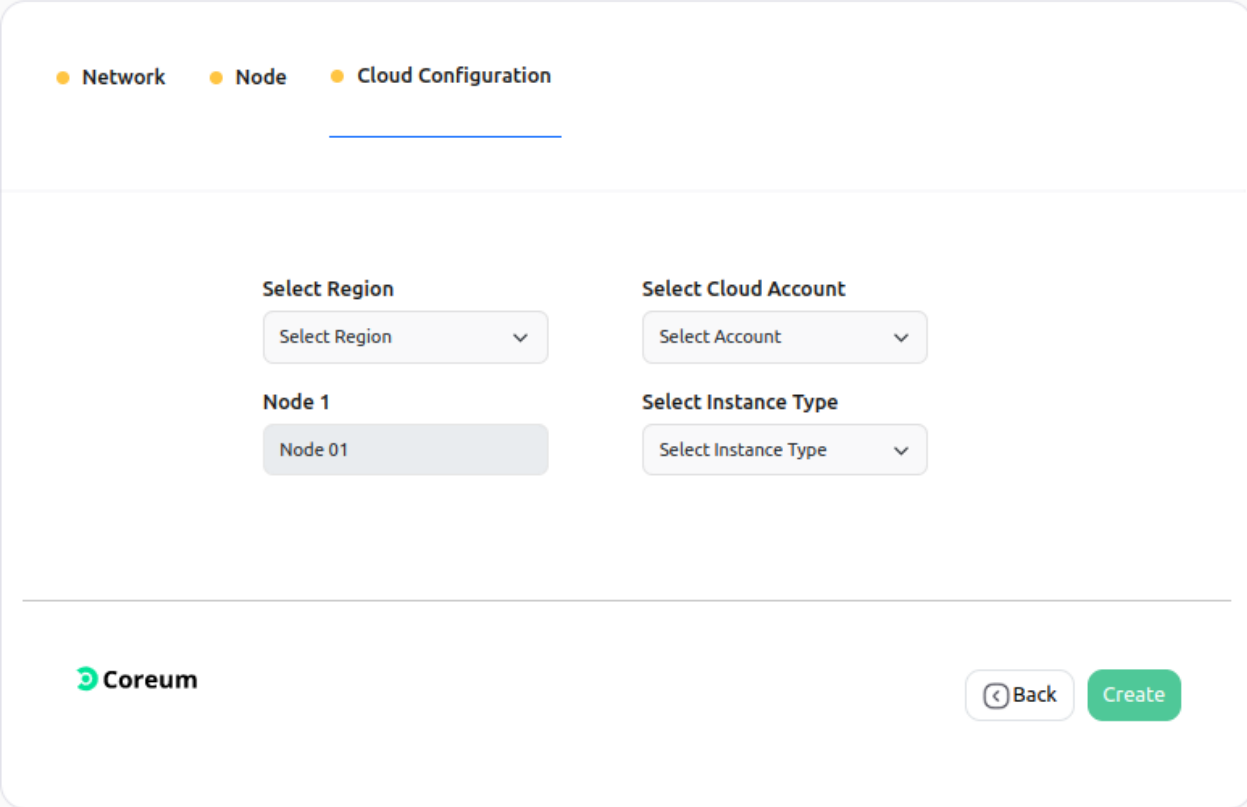
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

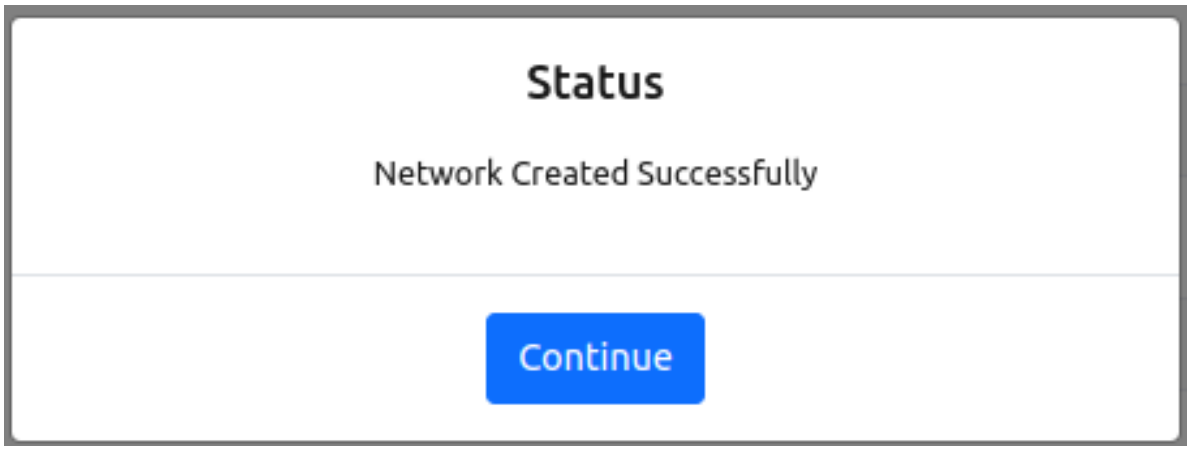
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of DigitalOcean, each of its regions has multiple, isolated locations known as Availability Zones. Digitial Ocean provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across it's Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the DO cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.



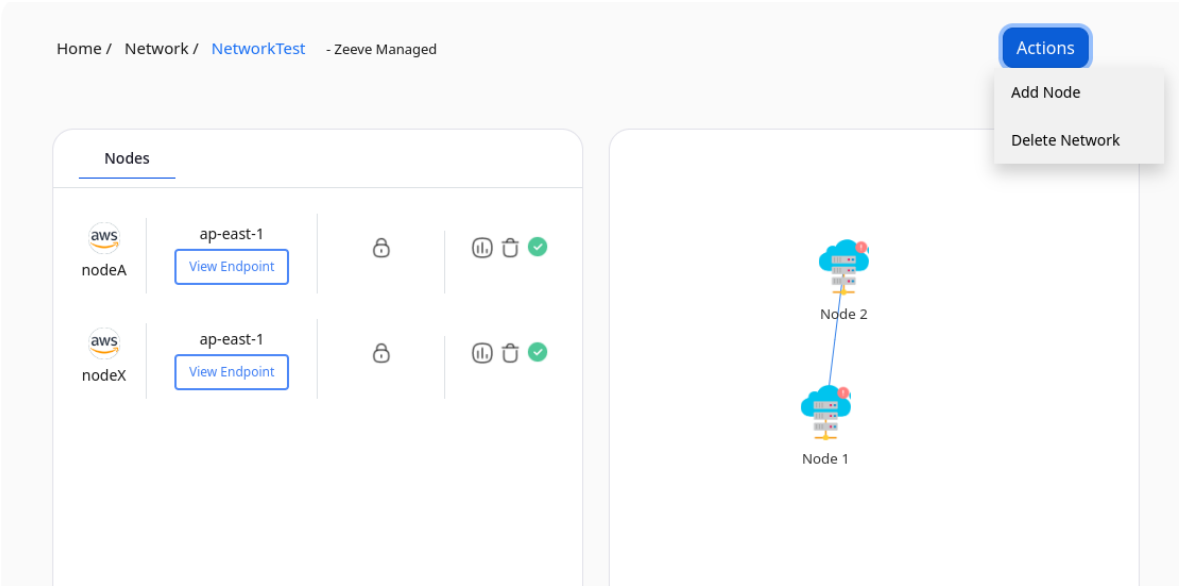
2. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes

listed you've just added to the network.

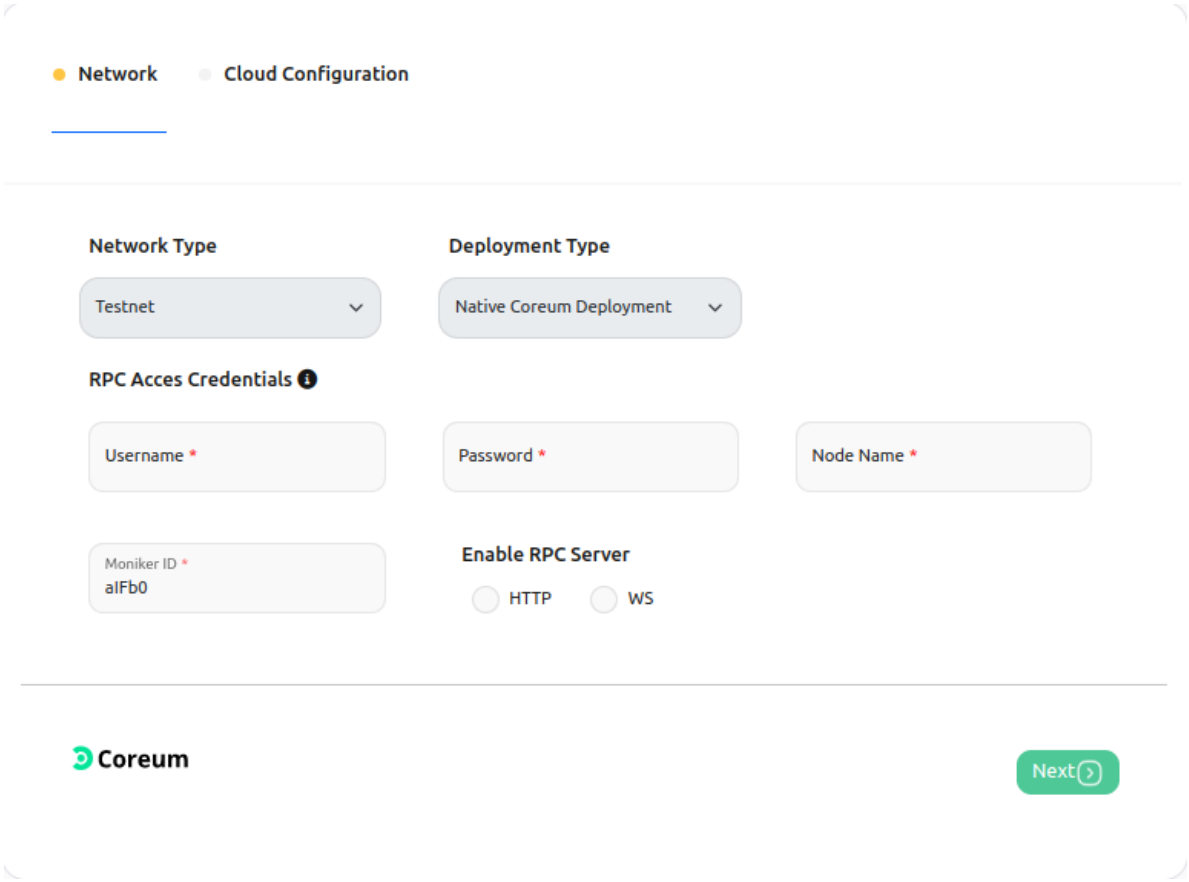
20.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

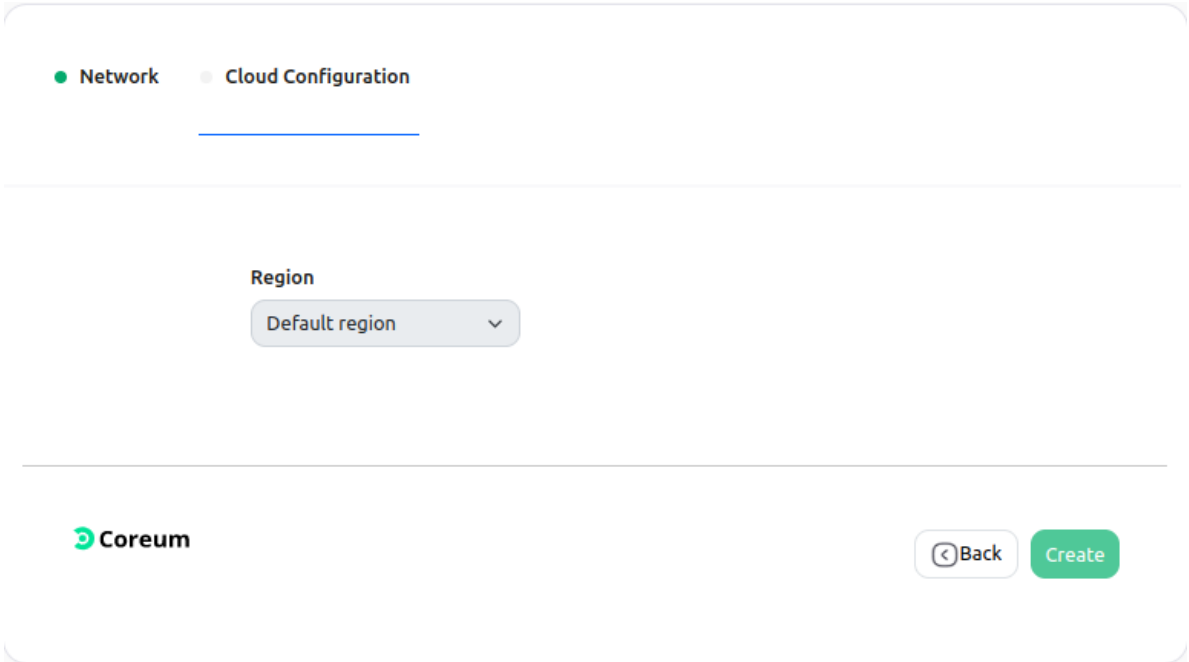
1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.

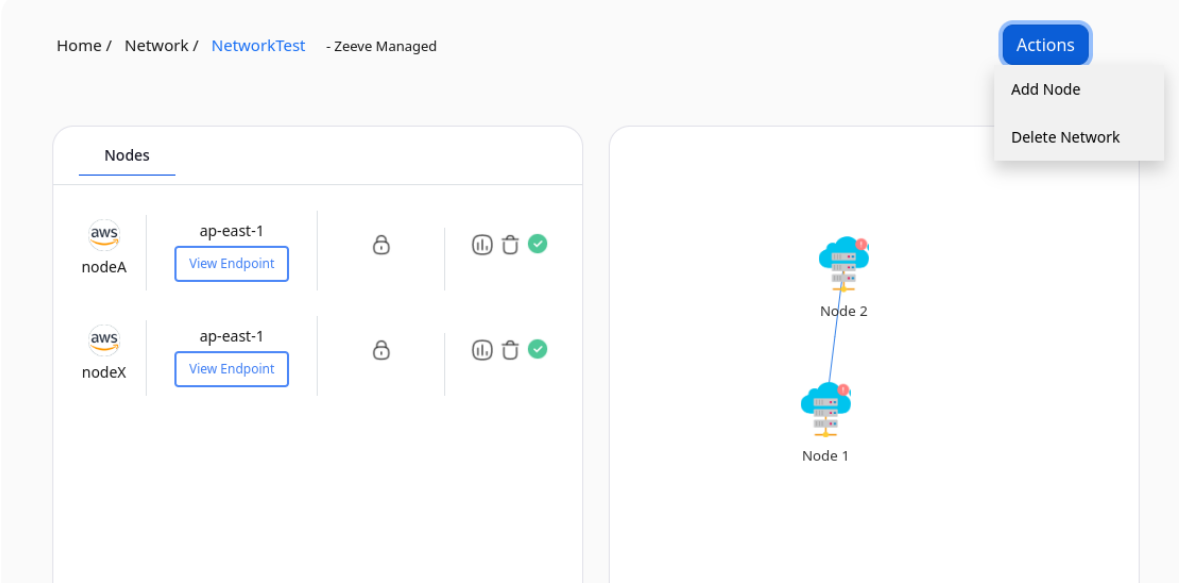


3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region will be prefilled according to the configuration of the network. Click on the **Create** button and it's done!

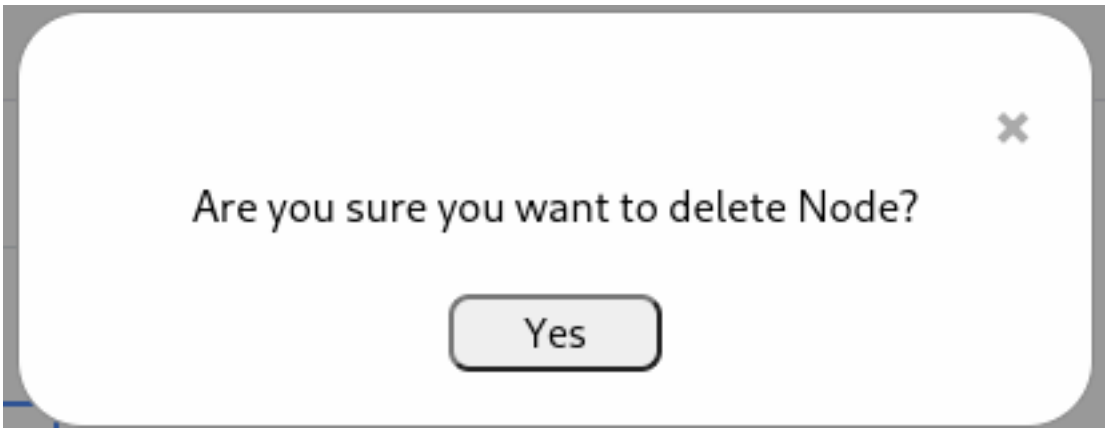


20.1.3 Delete node in a network

1. Select the network, in which you want to perform the delete node action, and click on the network card Ref.. You will get to see similar to the below image.

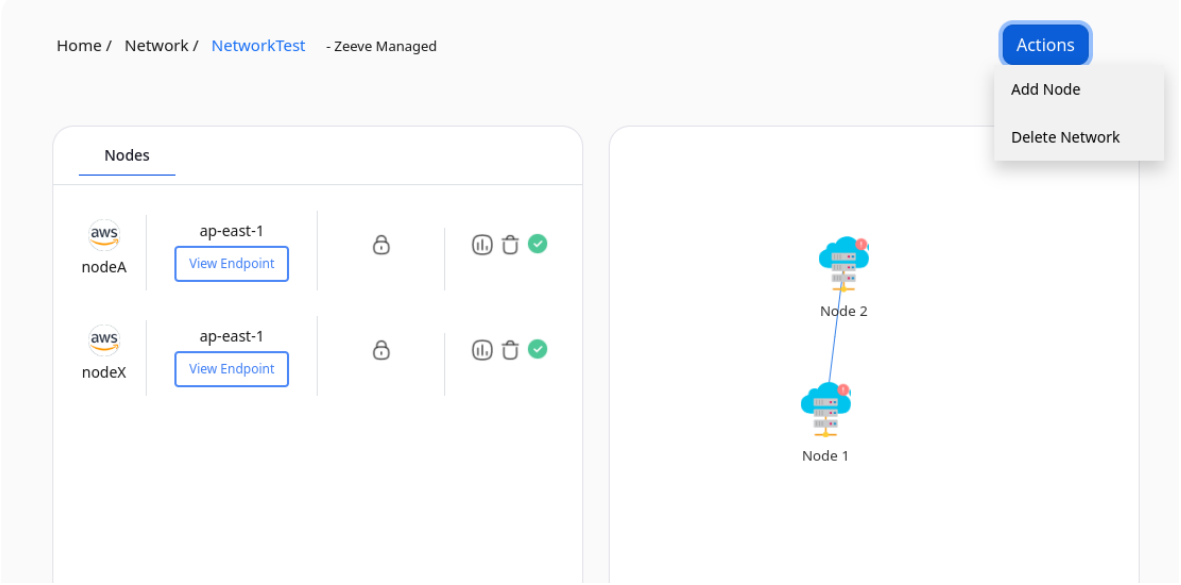


2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **yes** button to confirm.

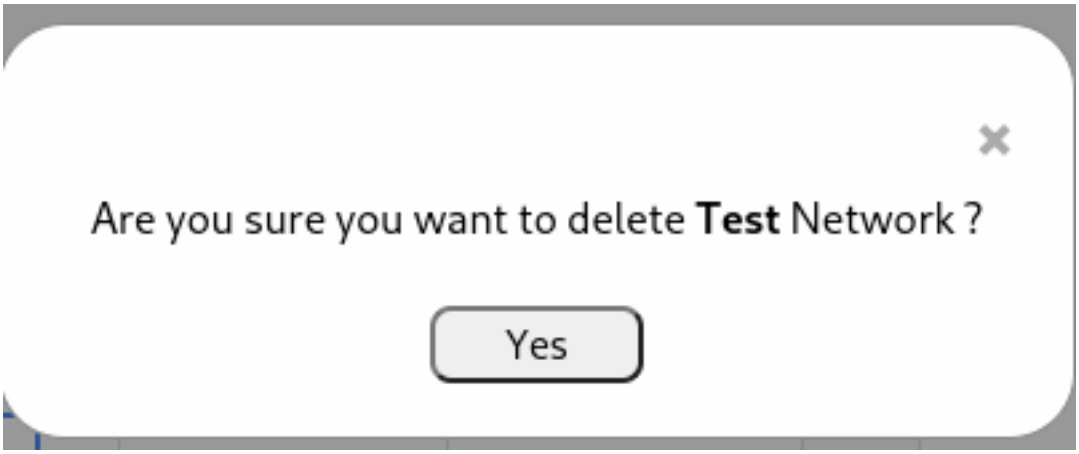


20.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the *Actions* button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

20.2 Staking nodes

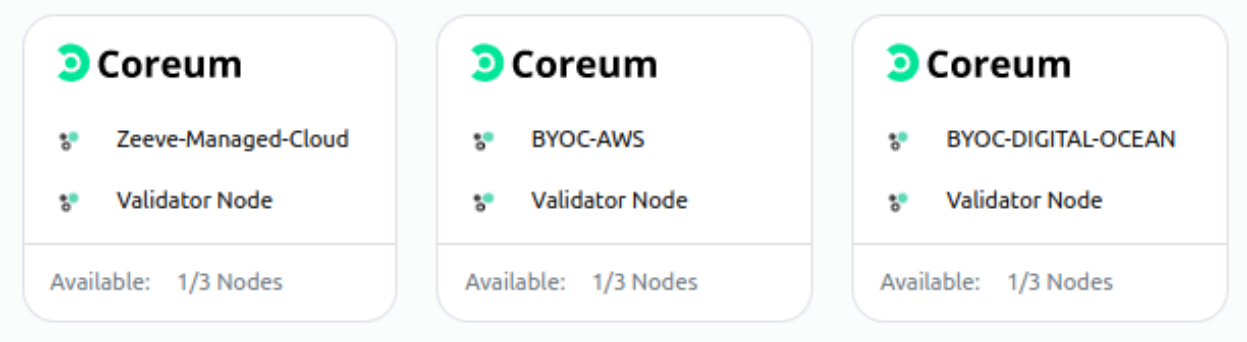
NOTE *Purchase* a subscription before proceeding.

1. *Create network*
2. *Update validator details*
3. *Unbound token*
4. *Withdraw rewards*
5. *Set rewards to a different wallet*
6. *Delete network*

20.2.1 Create a network

This section will provide you detailed steps for creating a network of **Coreum**.

On the **Network Configuration** page you will be able to see different configuration cards for Coreum, which looks similar to the image provided below.

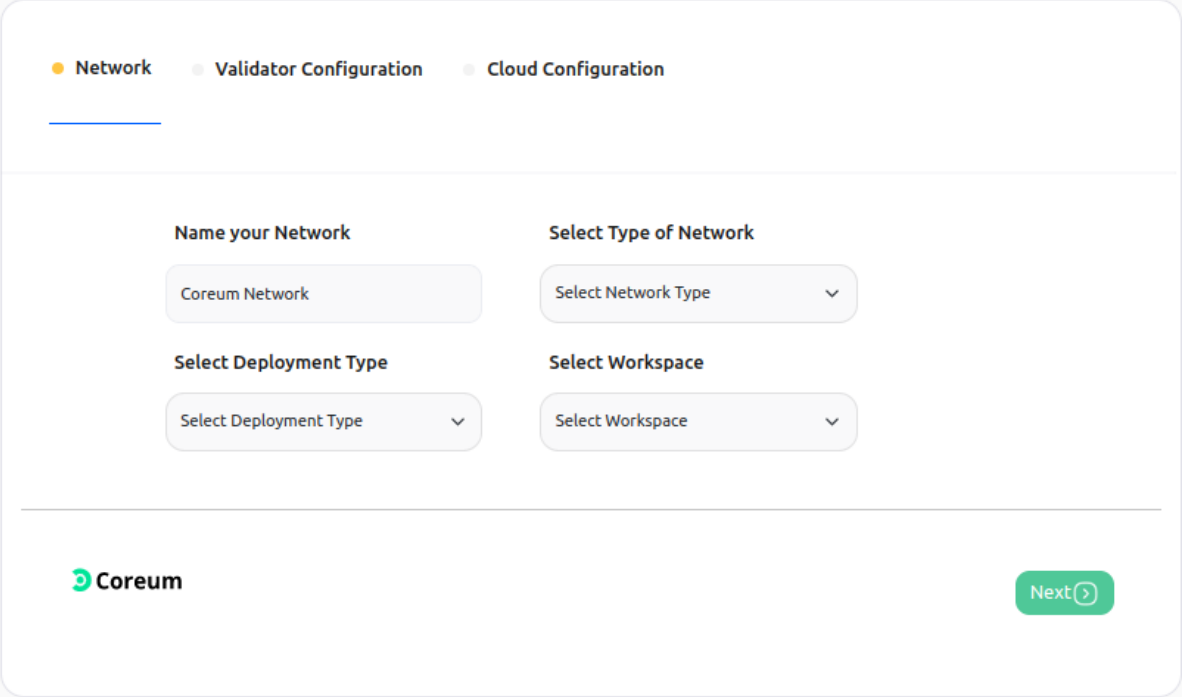


NOTE: *These cards can be different for your case. Card configurations depend on your purchased subscription.*

You can Choose **Zeeve-Managed-Cloud** (Zeeve's managed hosting) for the infrastructure of your node or you can use your cloud account (AWS/DO) for the hosting of your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

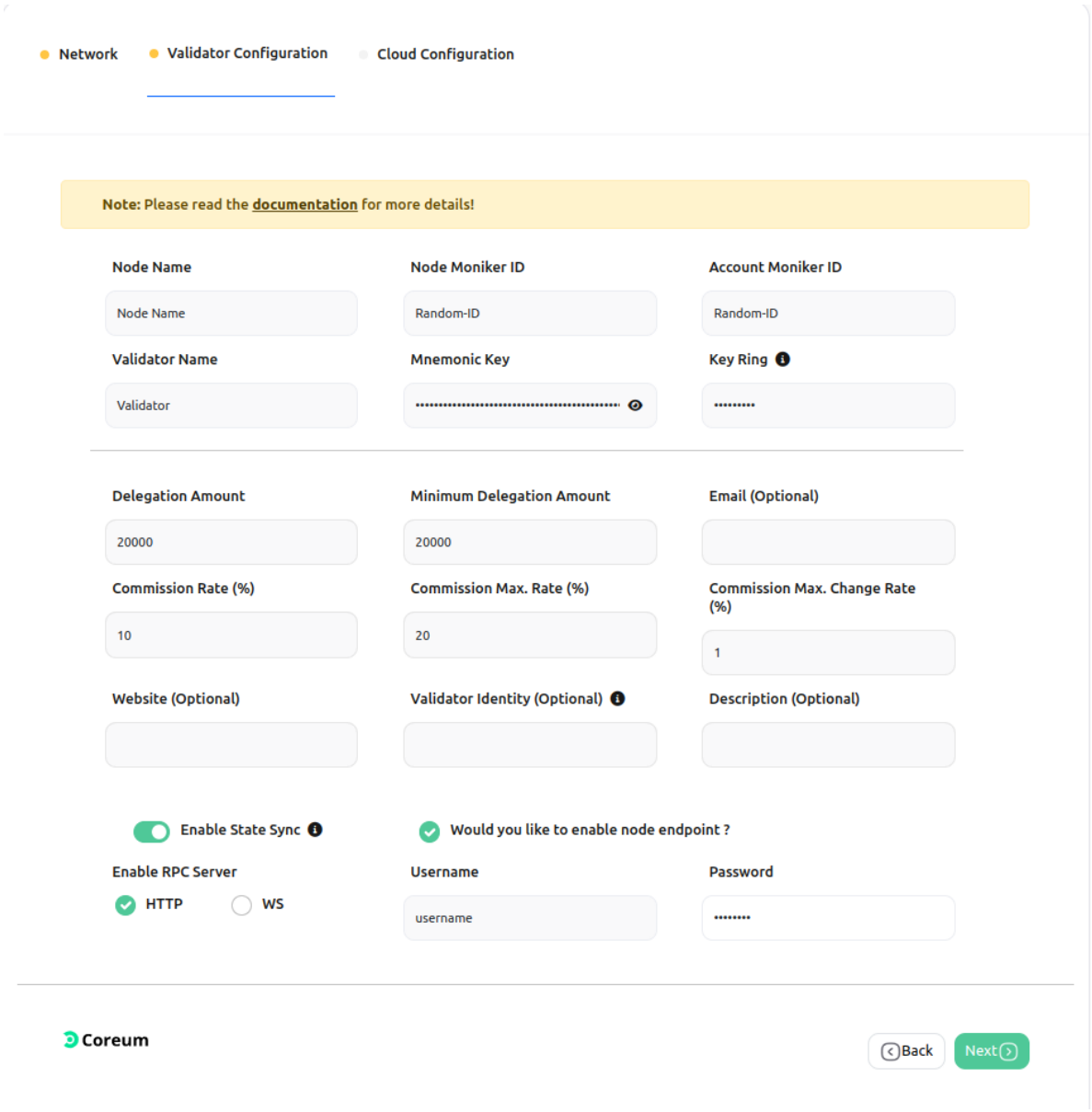
1. **Network Info:** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.



- **Name of Network**: To uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: Deployment type
- **Type Of Network**
  - **TestNet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after successful creation.

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration:** In this section, you have to fill in the details of the nodes you want to add to your network.



- **Node Name**: To identify your nodes, this field will be used.
- **Node MonikerId**: A custom human readable name for this node.
- **Account MonikerId**: Account id.
- **Validator Name**: The name which is visible on the Coreum Explorer.
- **Mnemonic Key**: bip39 mnemonic passphrase of your wallet.
- **Key Ring**: A passprase to access your private key.
- **Email**: Email will be used as a secret identity.
- **Delegation Amount**: Amount which you want to delegate/stake.
- **Minimum Delegation Amount**: The minimum delegation amount and must be grater or equal min_self_delegation.
- **Commission Rate**: The initial commission rate percentage.

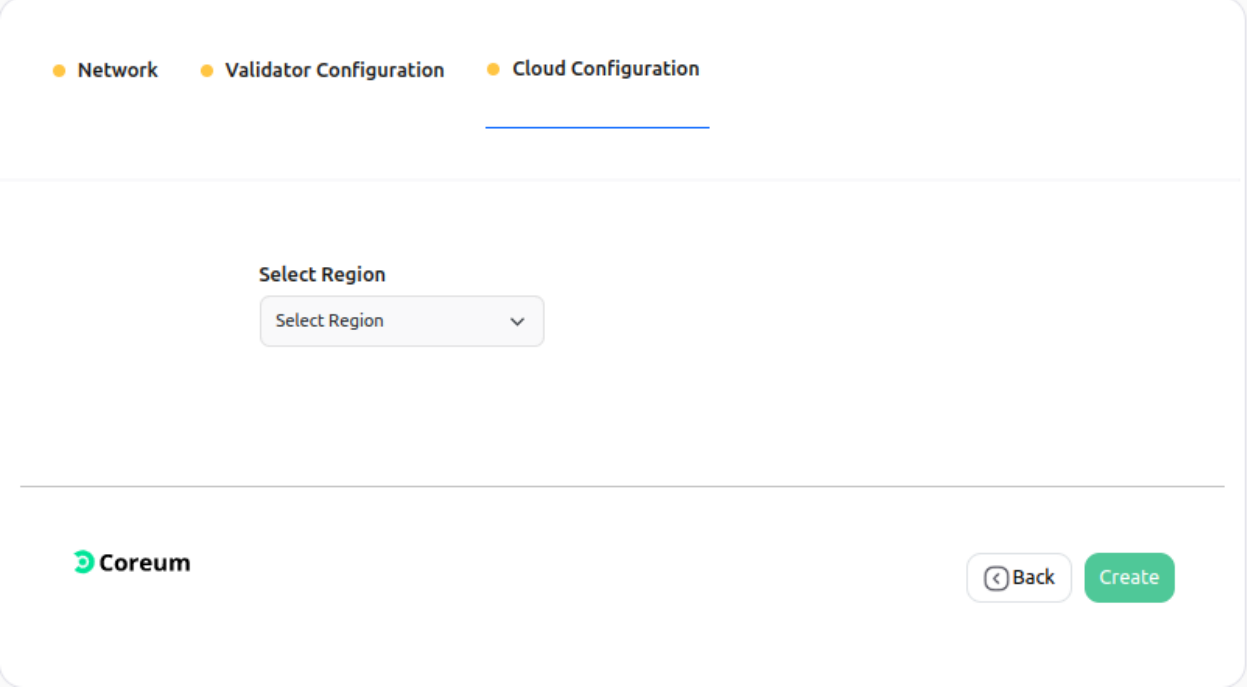
- **Commission Max. Rate**: The maximum commission rate percentage.
- **Commission Max. Change Rate**: The maximum commission change rate percentage (per day).
- **Website**: Website you want to be reflected in the.
- **Validator Identity**: The optional identity signature (ex. UPort or Keybase).
- **State Sync**: By enabling state sync your node will download data related to the head or near the head of the chain and verify the data. This leads to drastically shorter times for joining a network Read more.
- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.

After providing the details click on **Next step** to go to the last step.

1. **Cloud Configuration:** This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**
    1. *Zeeve Managed*
    2. *BYOC - AWS*
    3. *BYOC - DO*

Zeeve Managed

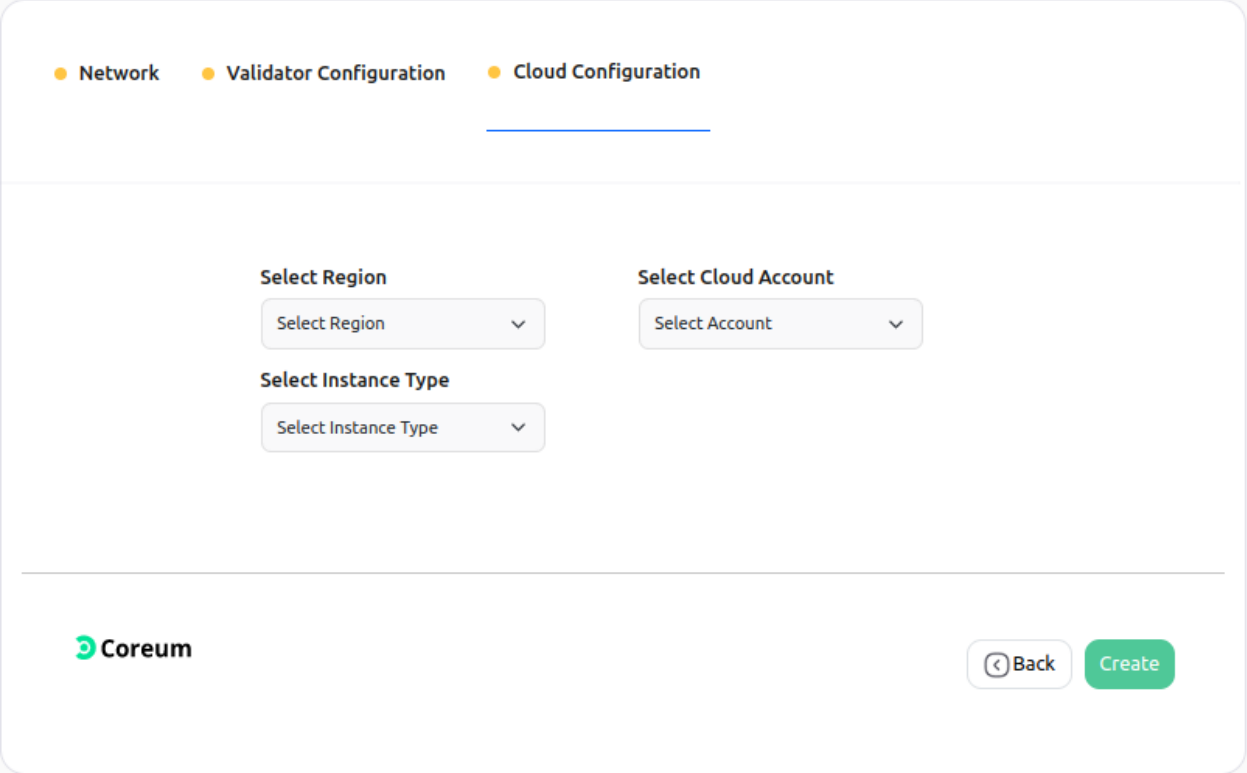
In the case of **Zeeve Managed**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted.

BYOC - AWS

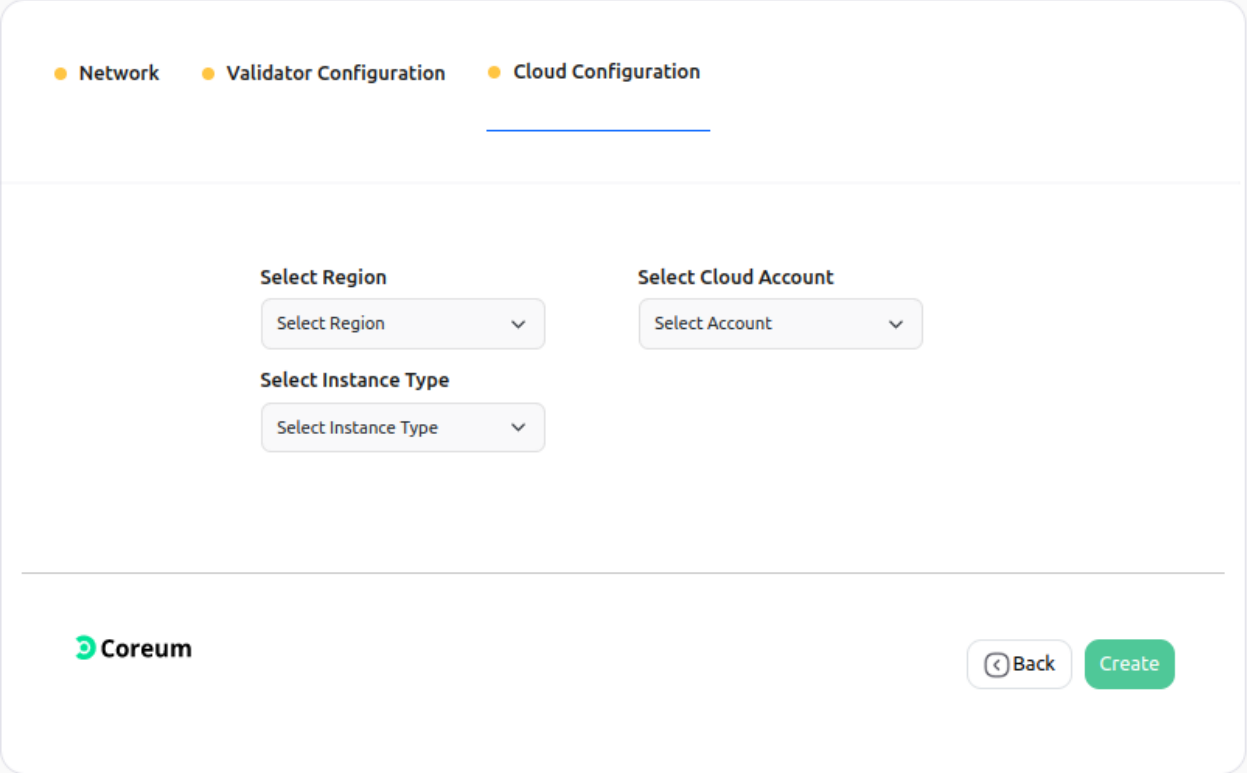
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

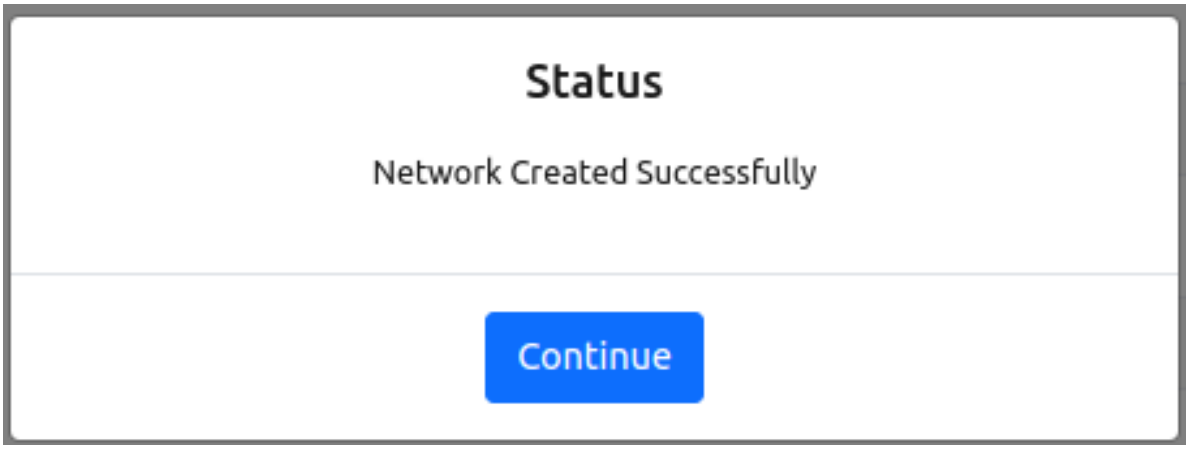
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of DigitalOcean, each of its regions has multiple, isolated locations known as Availability Zones. Digitial Ocean provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across it's Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the DO cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.



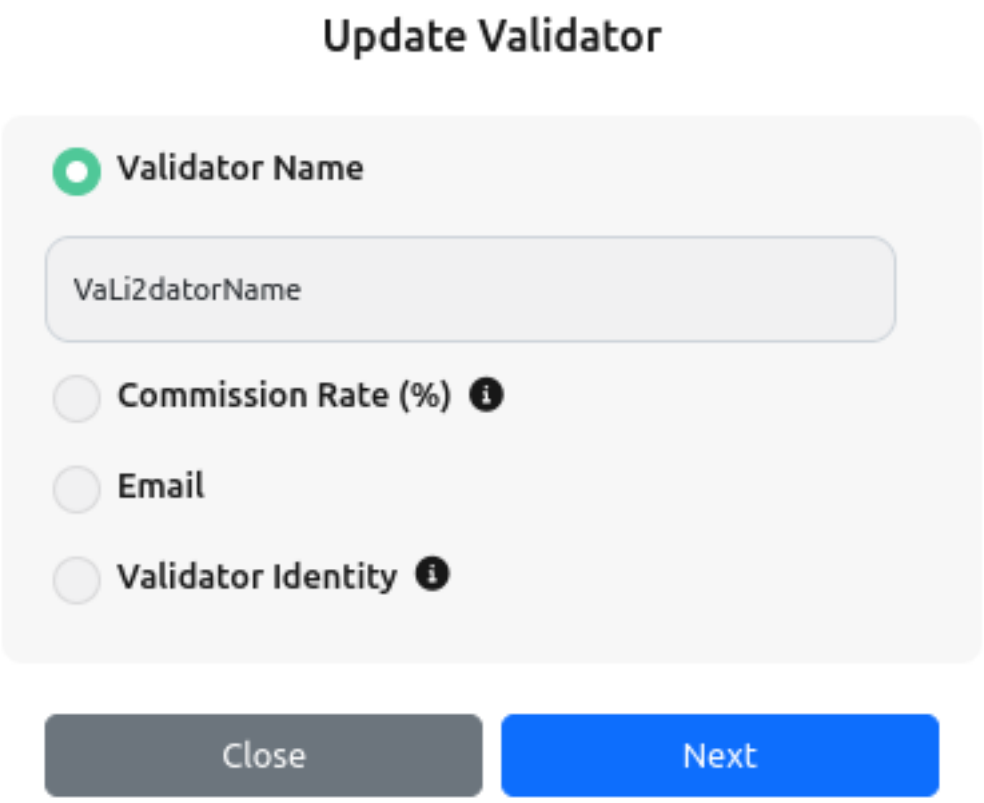
2. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes listed you've just added to the network.

20.2.2 Update Validator Details

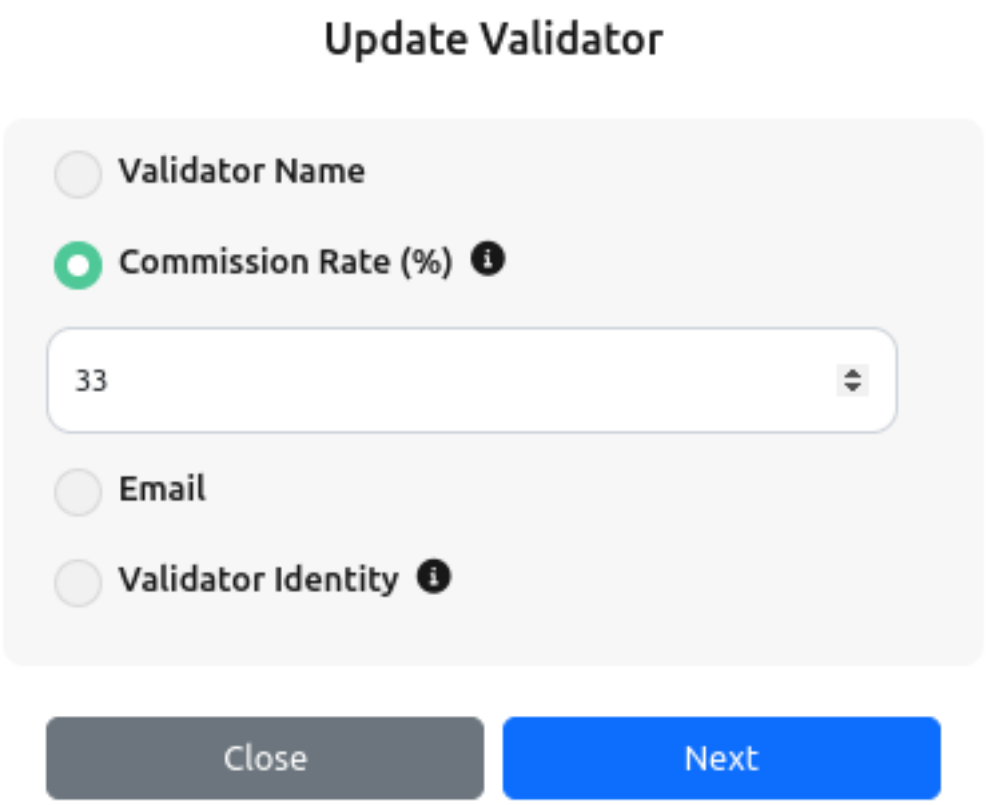
1. Find the Edit button on top right of the screen and click on it to edit and update the prefered validator detail. You will get to see similar to the below image.

   Step 01 :

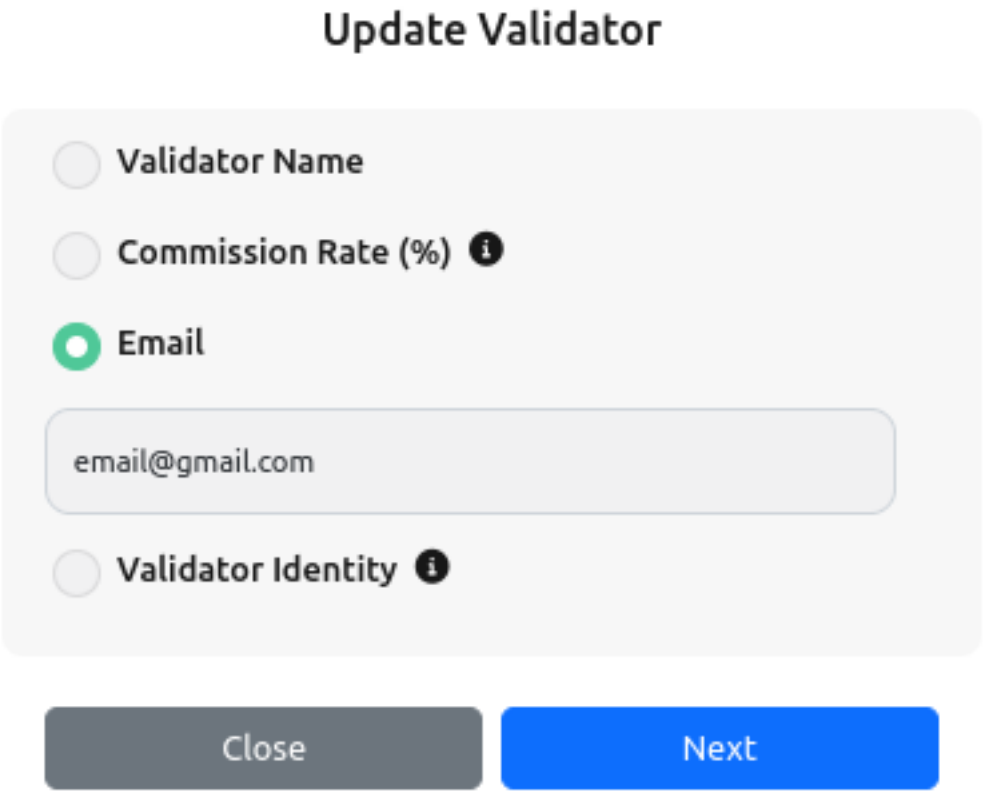
   - Validator Name



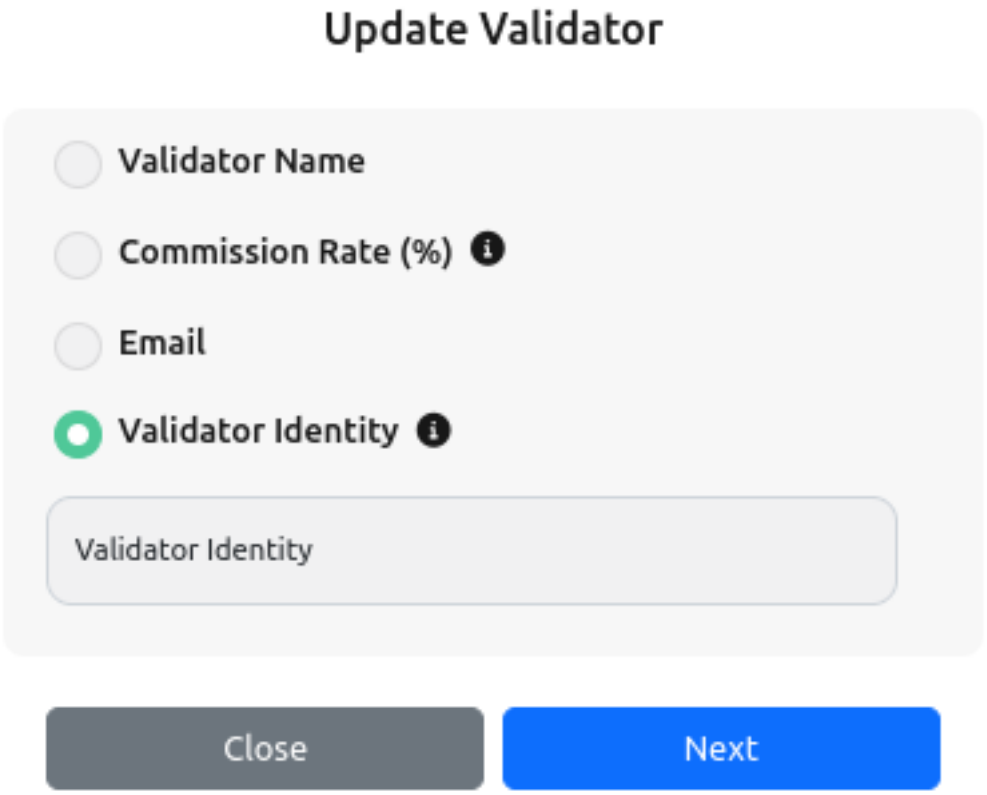
   - Commission Rate


Update Validator
Validator Name
Commission Rate (%)
33
Email
Validator Identity
Close
Next

- Email

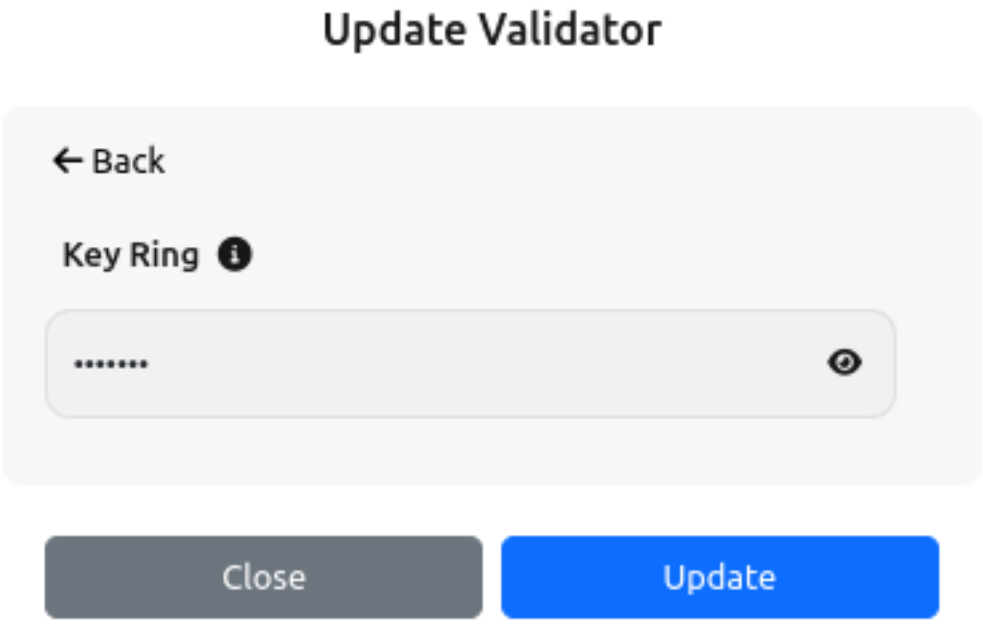

Update Validator
Validator Name
Commission Rate (%)
Email
email@gmail.com
Validator Identity
Close
Next

- Validator Identity



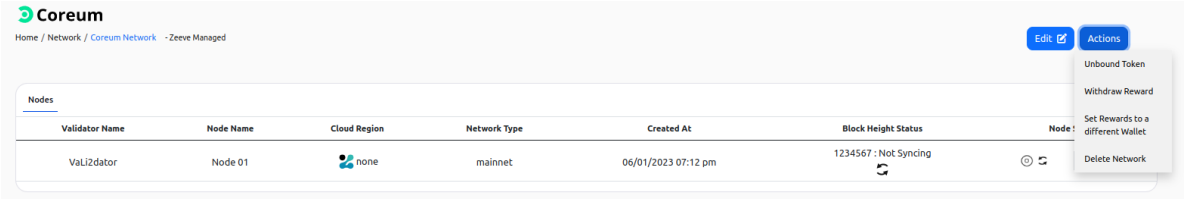
Step 02 :

- Keyring is require in order to update the validator details.

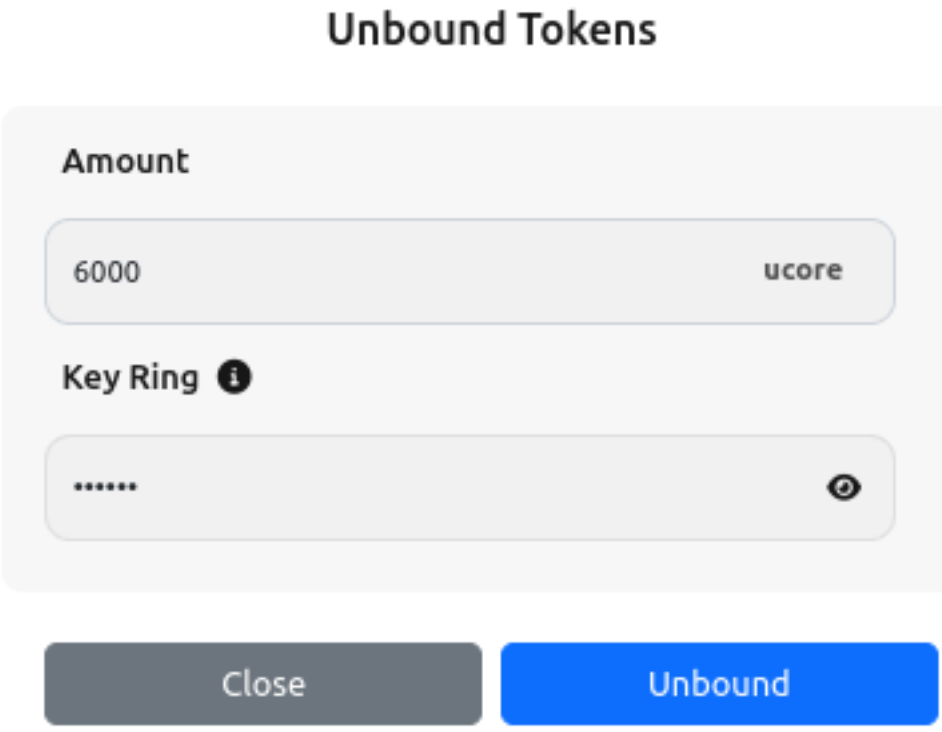


20.2.3 Ubound tokens

1. Click on the *Actions* button on the top right, and select the **Unbound Token** option.

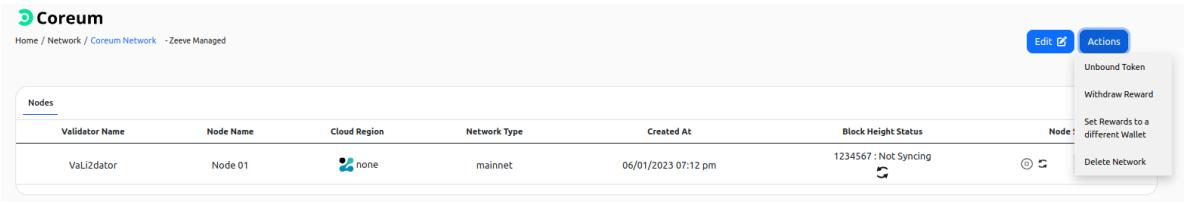


2. Enter the amount you want to unbound and the keyring, then click on Unbound button.

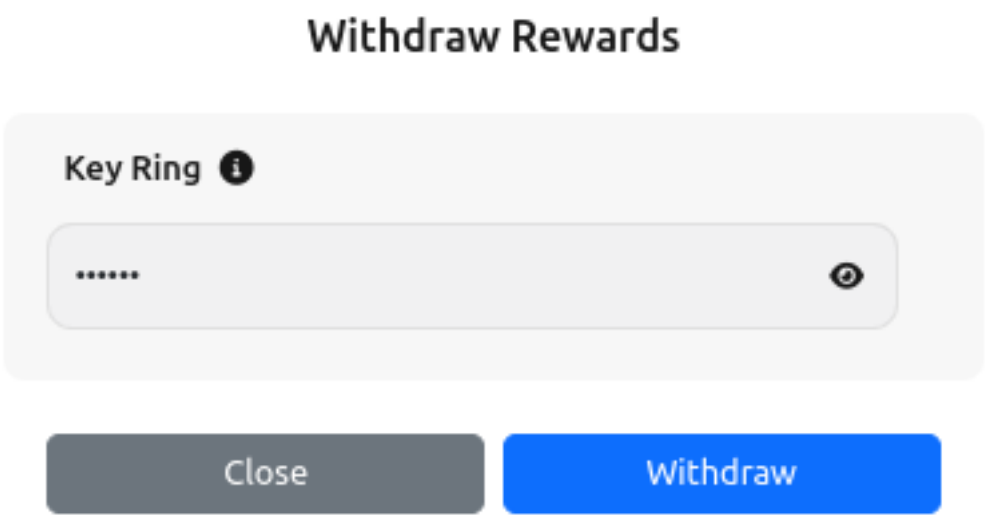


20.2.4 Withdraw rewards

1. Click on the *Actions* button on the top right, and select the **Withdraw reward** option.

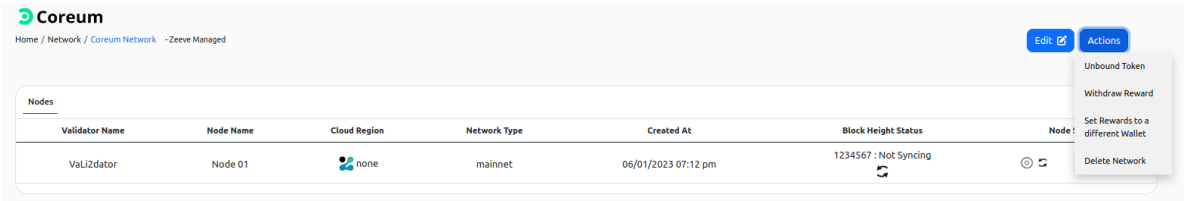


2. Enter the keyring then click on Withdraw button.

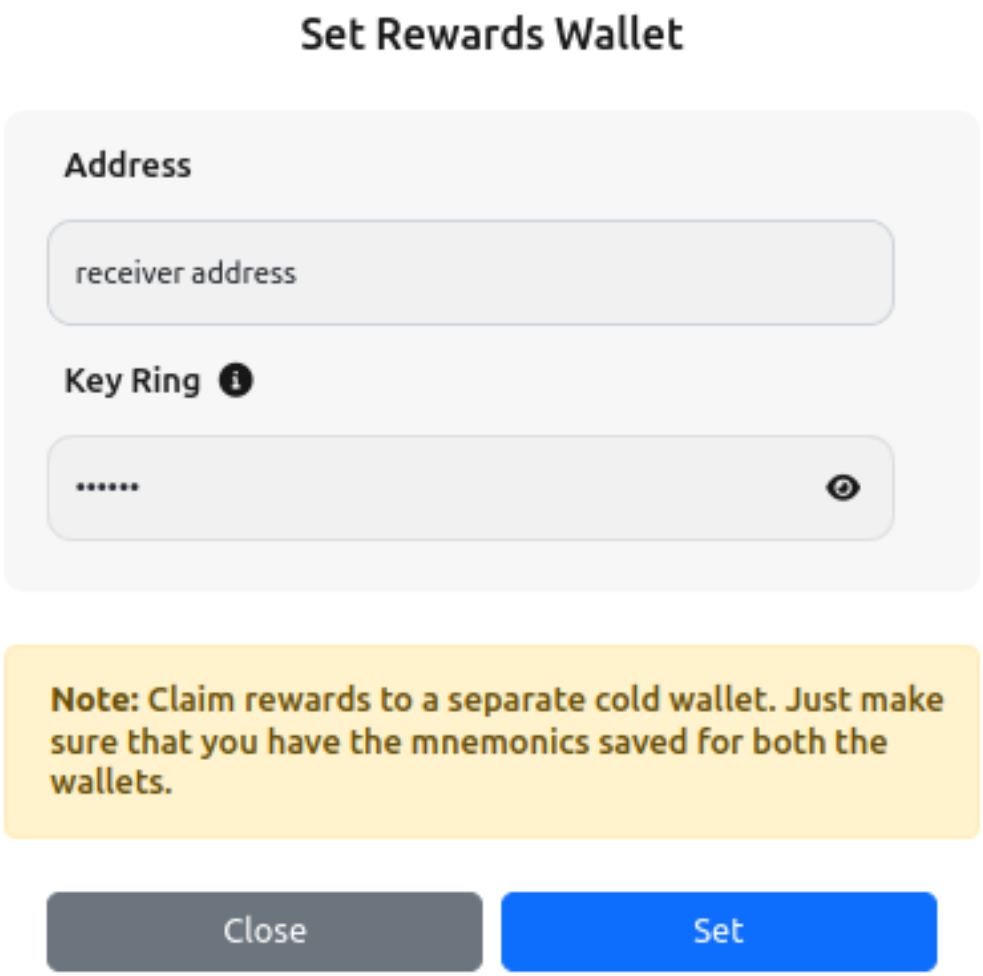


20.2.5 Set rewards to a different wallet

1. Click on the *Actions* button on the top right, and select the **Set rewards to a different wallet** option.

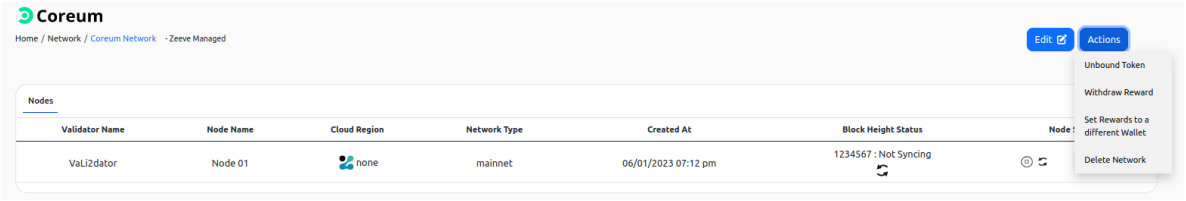


2. Enter the wallet address and keyring then click on Set button.

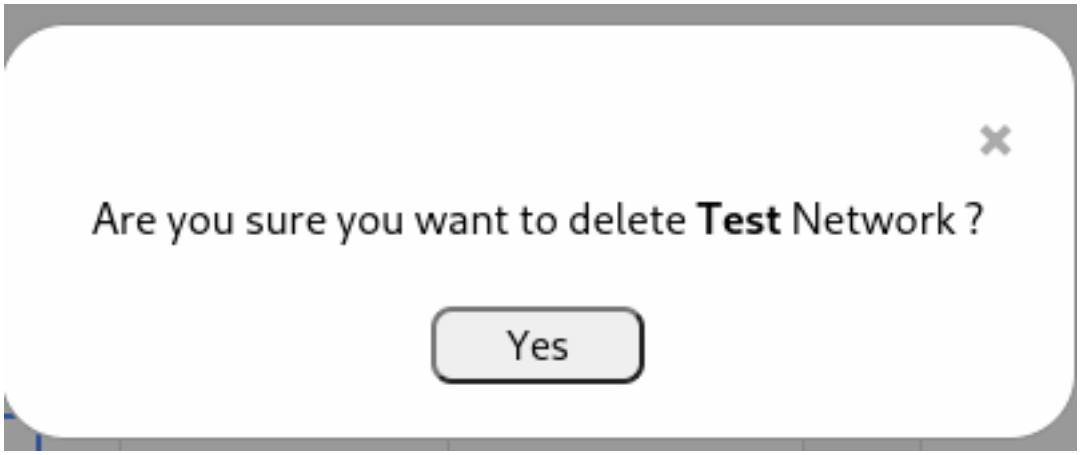


20.2.6 Delete a network

1. Select the network you want to delete, and click on the network card. You will get to see similar to the below image.



2. Click on the *Actions* button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

---

description: Use Dcomm with Zeeve's platform through our API and tools. Our documentation provides technical details on authentication, making requests and handling responses for interacting with Dcomm and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

TWENTYONE

DCOMM STAKING NODE SETUP

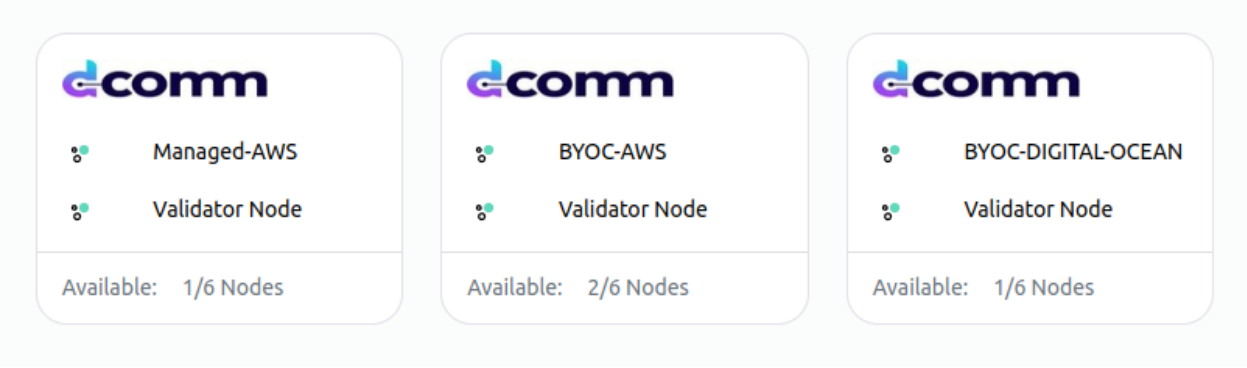
NOTE *Purchase* a subscription before proceeding.

1. *Create a network*
2. *Add an additional node*
3. *Delete a node*
4. *Delete a network*

21.1 Create a network

This section will provide you detailed steps for creating a network of **Dcomm**.

On the **Network Configuration** page you will be able to see different configuration cards for Dcomm, which looks similar to the image provided below.

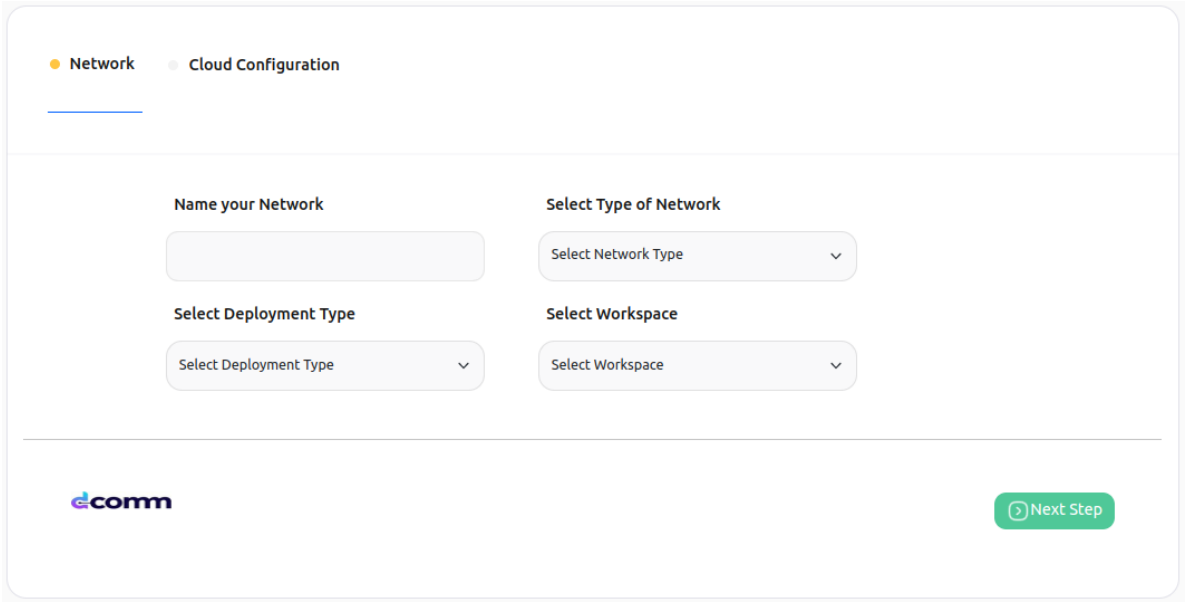


***NOTE:** These configuration cards can be different based on your purchased subscriptions.**

You can Choose **Zeeve Managed Cloud** or you can use your cloud account (AWS/DO) for the infrastructure of your node.

Choose the card with the configuration you want. Clicking on the card you will be redirected to the network setup page.

1. **Network Info**



- **Network Name**: A name to identify your network.
- **Deployment Type**: Deployment type
- **Network Type**
  - **Melbourne Testnet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after the successful creation.

Proceed further by clicking on the **Next Step** button after providing all the details.

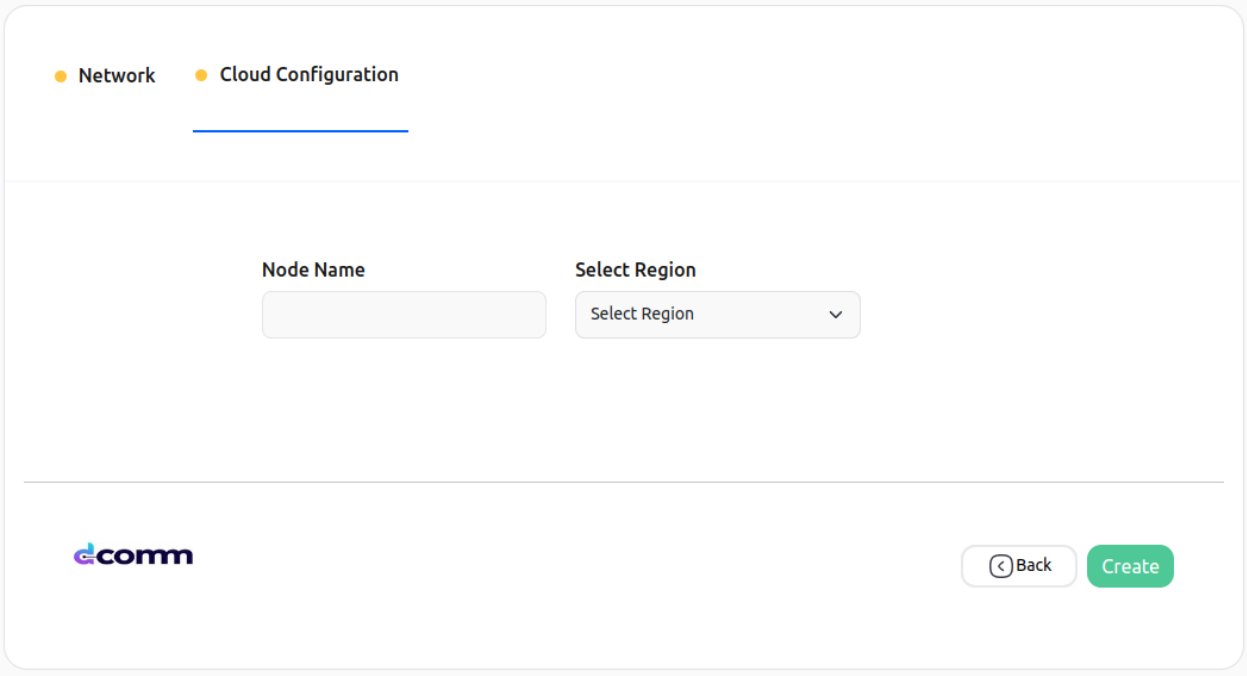
2. **Cloud Configuration**

   This step configures the cloud settings for your node. This step can vary based on your selection of **Network configuration card**

   1. *Zeeve Manged Cloud*
   2. *Bring Your Own Cloud (BYOC)*

21.1.1 Zeeve Managed Cloud

In the case of **Managed - Cloud**, select the region for the network under **Select Region** and provide a name to your node.



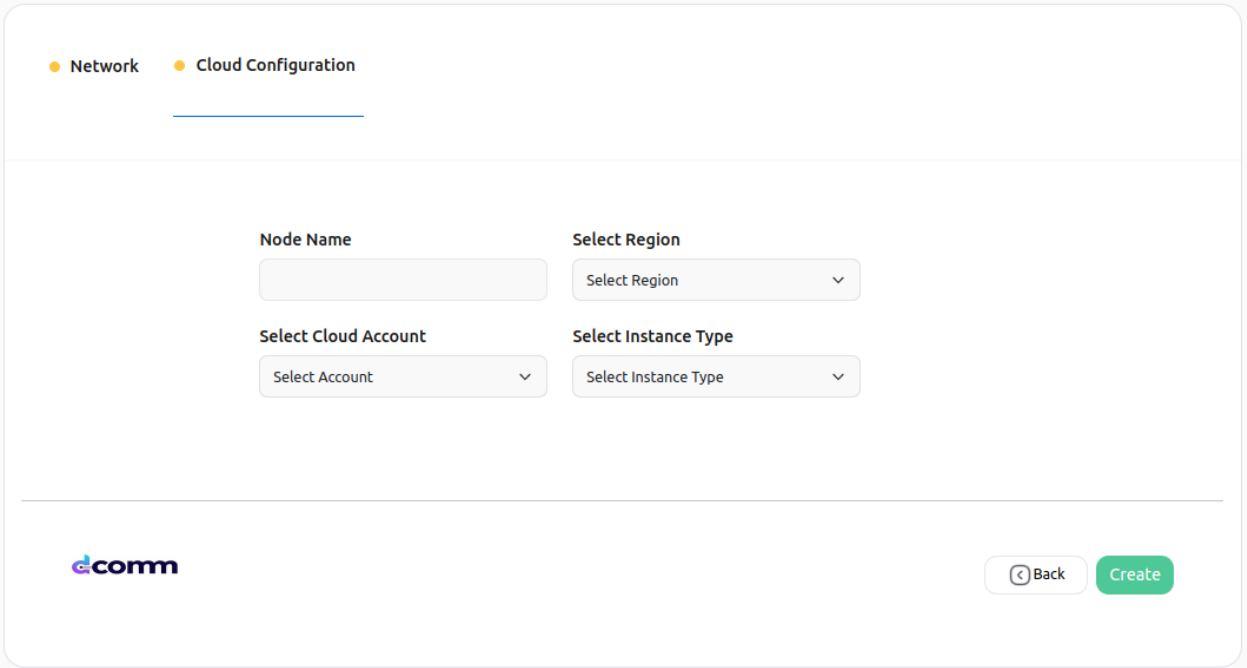
- **Node Name**: A name to identify your node, this field requires a unique name. Unique means that it should be unique in a network to which you are adding a node.
- **Region**: It indicates the region of the cloud service. These regions are the geographic locations where your network instances are going to be hosted.

For better understanding of which region is best for you please refer the following

New York City, The US: NYC1, NYC3 San Francisco, The US: SFO2 Toronto, Canada: TOR1 London, United Kingdom: LON1 Frankfurt, Germany: FRA1 Amsterdam, the Netherlands: AMS3 Bangalore, India: BLR1

21.1.2 Bring Your Own Cloud

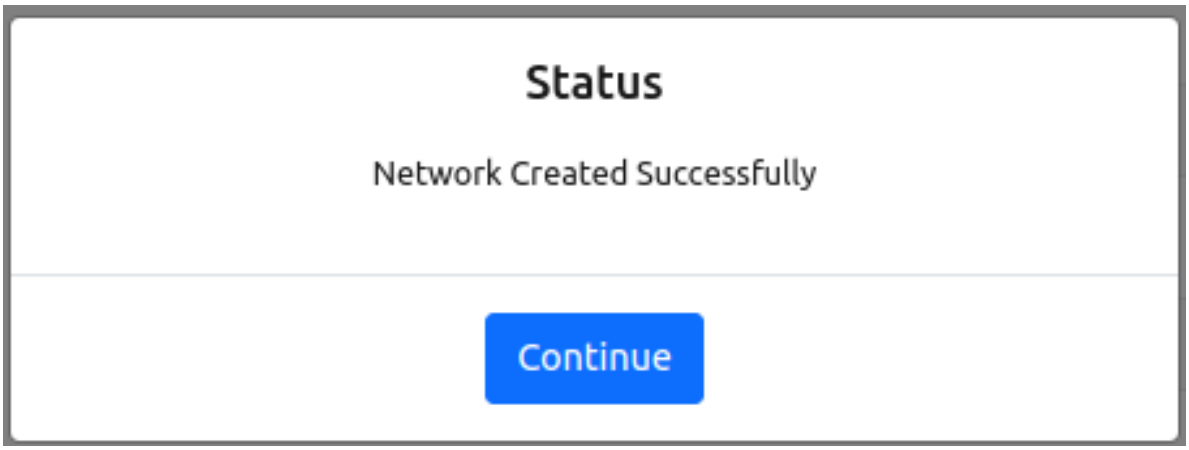
In the case of **BYOC** (AWS or Digital Ocean), select the region for the network by clicking on **Select Region**, select the *Cloud* account you want to use by clicking on **Select Cloud Account**, choose the instance type as your requirement by clicking on **Select Instance Type** and provide a name to your node.



- **Node Name**: A name to identify your node, this field requires a unique name. Unique means that it should be unique in a network to which you are adding a node.
- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted.
- **Cloud Account**: It represents the cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

---

1. On clicking the **Create** button a pop-up window will open which ensures the successful creation of your network.

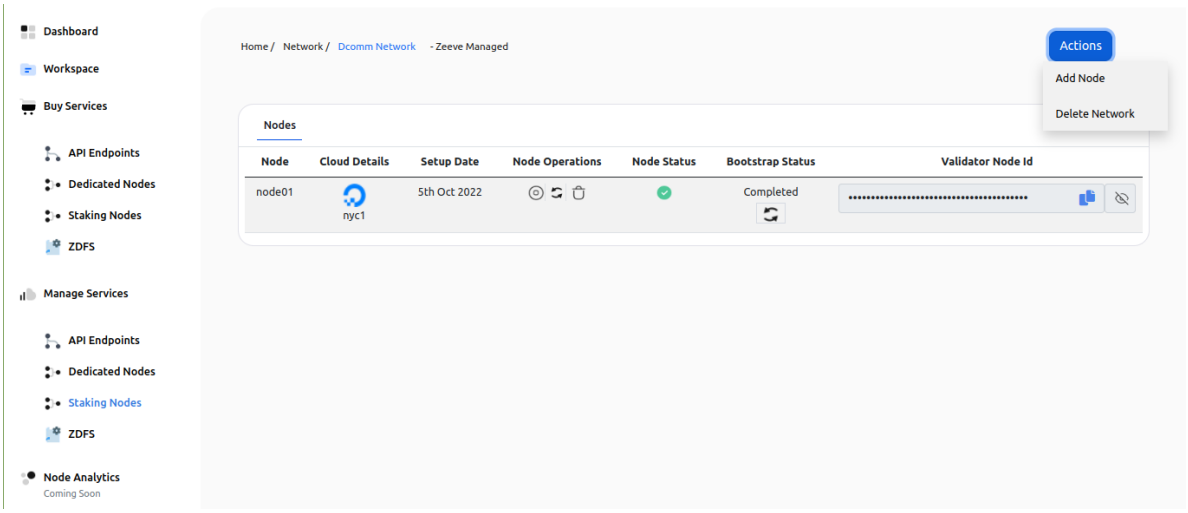


2. On clicking the **Continue** button you will be redirected to the page where you can see the network you created.

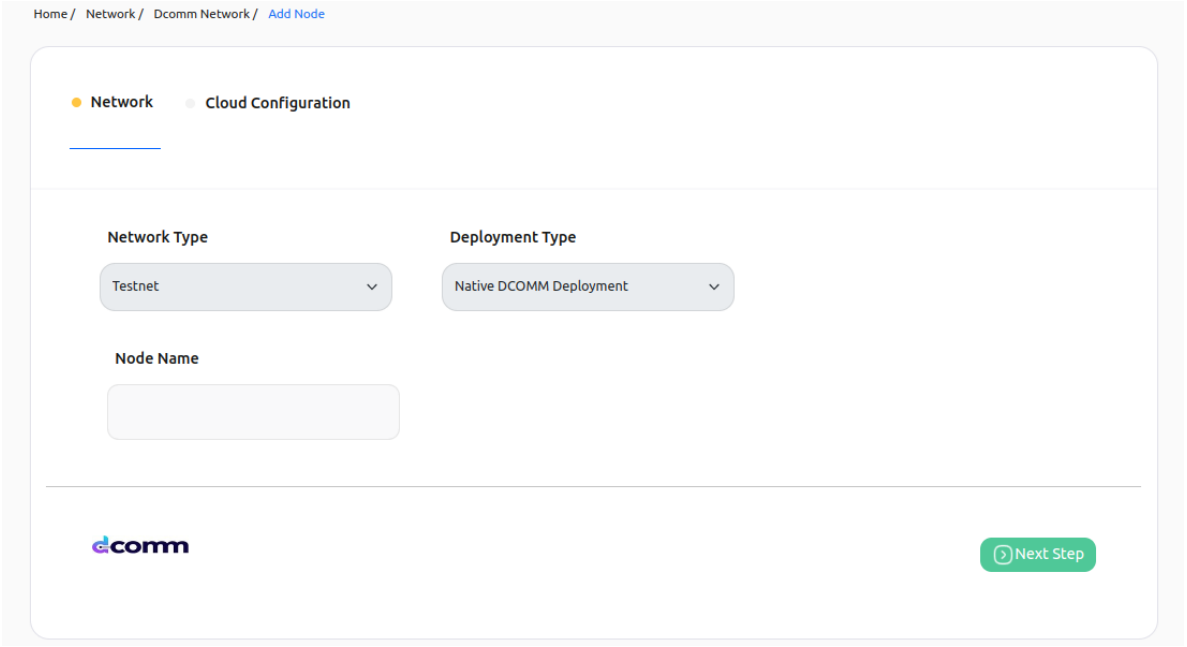
21.2 Add additional node to a network

This section will guide you on how you can add an additional node to a network. As you have already created a network, follow these steps to add more nodes to the network.

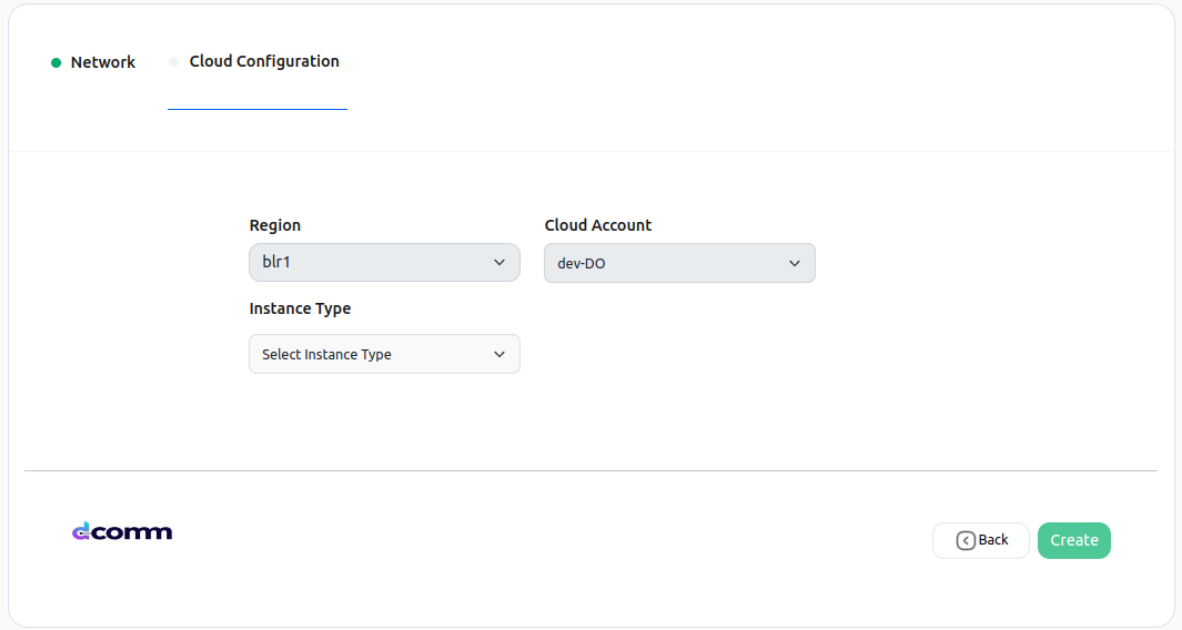
1. Visit the network detail page. Click on the *Actions* button on the top right, and select the **Add Node** option.



2. You will be redirected to the node setup page. Fill the name for the new node, network type and deployment type will be prefilled based on the network configuration. Click on the **Next** button to continue.



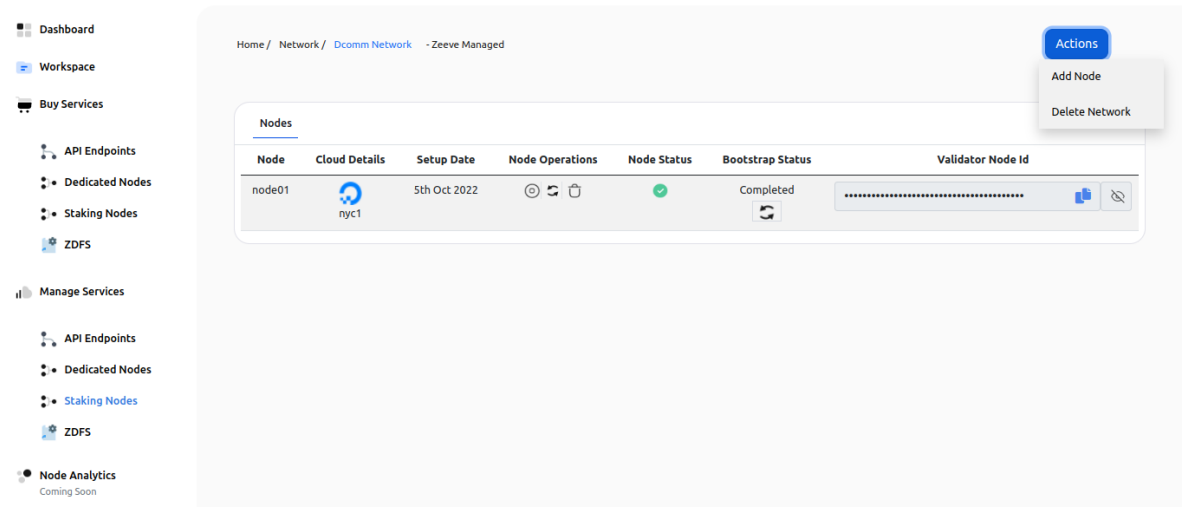
3. Select the instance type for the node, cloud account and region will be prefilled based on the network configuration. Click on the **Create** button and the node will be added.



NOTE For Zeeve Managed Cloud, the option for selecting the instance type will not be available as it will be selected by Zeeve.

21.3 Delete node in a network

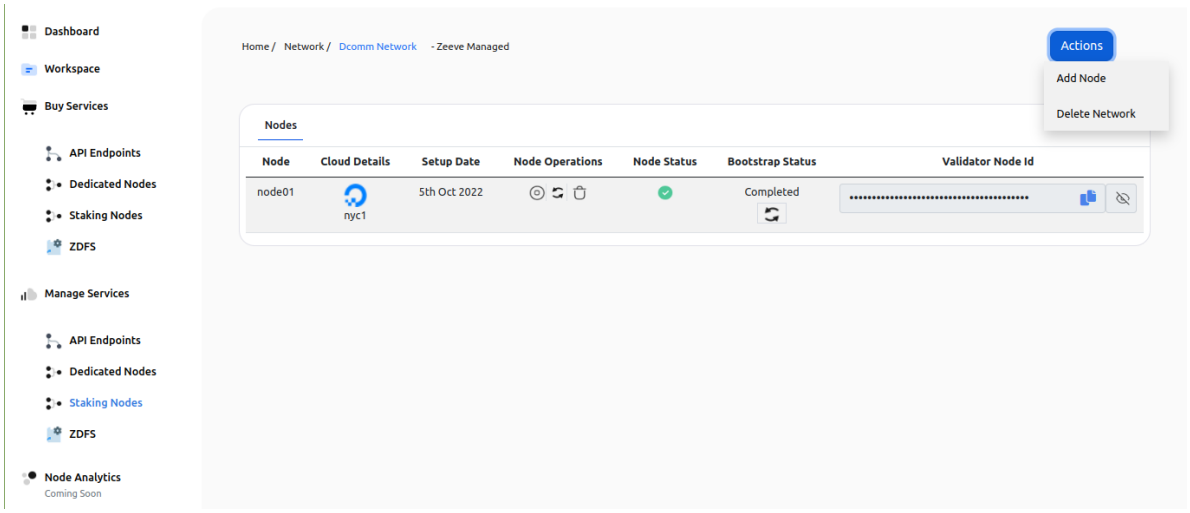
1. Select the network, in which you want to perform the delete node action, and click on the network card Ref.. You will get to see a page similar to the below image.



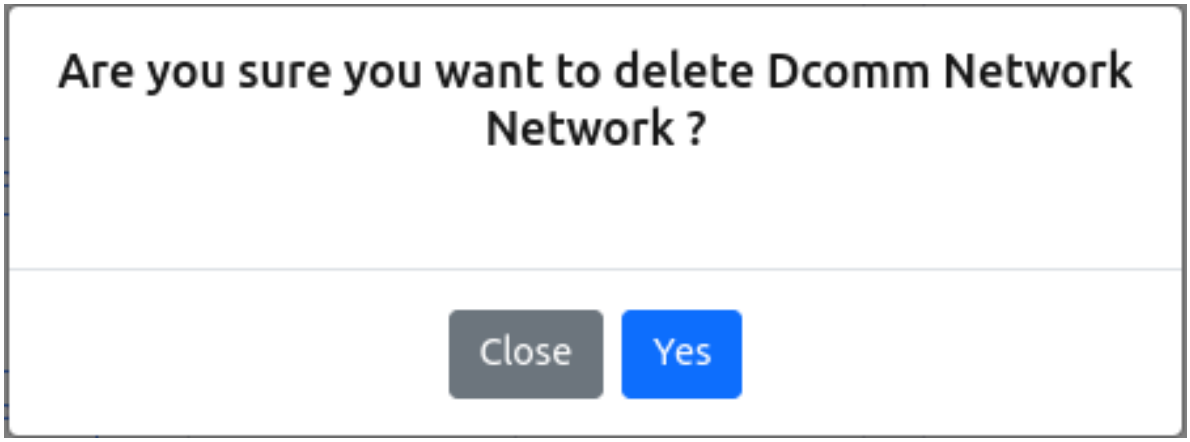
2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **Yes** button to confirm.

21.4 Delete a network

1. Visit the network detail pageRef.. Click on the *Actions* button on the top right, and select the **Delete Network** option.



2. A confirmation window will open, click on the **Yes** button to delete the network.



NOTE It will take a few minutes to delete a network.

21.5 Supported API methods

Just like any other protocol, **Dcomm** supports JSON RPC API call, which can be called to retrive the the information. **Dcomm** supports both **HTTP** as well as **WS(WebSocket)** JSON RPC methods.

21.5.1 HTTP

- ACT-Chain URL - `https://node_url/ext/bc/ACT/rpc`.
- AST-Chain URL - `https://node_url/ext/bc/AST`.
- ATH-Chain URL - `https://node_url/ext/bc/ATH`.

Example

```
import axios from "axios";

const data = JSON.stringify({
  "jsonrpc": "2.0",
  "id": 1,
  "method": "eth_blockNumber",
  "params": []
});

const config = {
  method: 'post',
  url: 'https://node_url/ext/bc/ACT/rpc',
  headers: {
    'Content-Type': 'application/json'
  },
  data : data
};

axios(config)
.then(function (response) {
  console.log(JSON.stringify(response.data));
})
.catch(function (error) {
  console.log(error);
});
```

Available HTTP methods

dvm.buildGenesis

dvm.getAddressTxs

dvm.getAllBalances

dvm.getAssetDescription

dvm.getBalance

dvm.getTx

dvm.getTxStatus

dvm.getUTXOs

health.health

eth_baseFee

eth_blockNumber

eth_call

eth_chainId

eth_getAssetBalance

eth_getBalance

eth_maxPriorityFeePerGas

eth_getTransactionCount

eth_sendRawTransaction

eth_getBlockByHash

eth_getBlockByNumber

eth_getTransactionByHash

eth_getTransactionReceipt

dcm.getAtomicTx

dcm.getAtomicTxStatus

dcm.getUTXOs

net_version

web3_clientVersion

web3_sha3

index.getLastAccepted (AST Transactions)

index.getContainerByIndex (AST Transactions)

index.getContainerByID (AST Transactions)

index.getContainerRange (AST Transactions)

index.getIndex (AST Transactions)

index.isAccepted (AST Transactions)

index.getLastAccepted (AST Vertices)

index.getContainerByIndex (AST Vertices)

index.getContainerByID (AST Vertices)

index.getContainerRange (AST Vertices)

index.getIndex (AST Vertices)

index.isAccepted (AST Vertices)

index.getLastAccepted (ATH Blocks)

index.getContainerByIndex (ATH Blocks)

index.getContainerByID (ATH Blocks)

index.getContainerRange (ATH Blocks)

index.getIndex (ACT Blocks)

index.isAccepted (ACT Blocks)

index.getLastAccepted (ACT Blocks)

index.getContainerByIndex (ACT Blocks)

index.getContainerByID (ACT Blocks)

index.getContainerRange (ACT Blocks)

index.getIndex (ATH Blocks)

index.isAccepted (ATH Blocks)

info.getBlockchainID

info.getNetworkID

info.getNetworkName

info.getNodeID

info.getNodeIP

info.getNodeVersion

info.isBootstrapped

info.getTxFee

info.getVMs

info.uptime

info.peers

authority.getBalance

authority.getBlockchains

authority.getBlockchainStatus

authority.getCurrentSupply

authority.getTotalStake

authority.getCurrentValidators

authority.getMaxStakeAmount

authority.getHeight

authority.getMinStake

authority.getRewardUTXOs

authority.getStake

authority.getTxStatus

authority.getPendingValidators

authority.getStakingAssetID

authority.getSubnets

authority.getTx

authority.getTimestamp

authority.getUTXOs

authority.getValidatorsAt

authority.sampleValidators

authority.validatedBy

authority.validates

21.5.2 WebSocket

- ACT-Chain URL - `wss://node_url/ext/bc/ACT/ws`.

***NOTE:** As of now only **Action Chain** supports WS RPC methods.

Example

```
  import WebSocket from 'ws';

  const ws = new WebSocket('wss://node_url/ext/bc/ACT/ws');

  ws.on('open', function open() {
          console.log('connected');

    // NOTE : use delay to avoid missing messages from server
          setTimeout(() => {
                  const msg = {"jsonrpc":"2.0","method":"eth_blockNumber","params
↪":[],"id":1};
          ws.send(JSON.stringify(msg));
    }, 1000);
  });

  ws.on('message', function message(data) {
    console.log('received: %s', data);
  });
```

Available WebSocket Methods

eth_baseFee

eth_blockNumber

eth_call

eth_chainId

eth_getAssetBalance

eth_getBalance

eth_maxPriorityFeePerGas

eth_getTransactionCount

eth_sendRawTransaction

eth_getBlockByHash

eth_getBlockByNumber

eth_getTransactionByHash

eth_getTransactionReceipt

dcm.getAtomicTx

dcm.getAtomicTxStatus

dcm.getUTXOs

net_version

web3_clientVersion

web3_sha3

CHAPTER

TWENTYTWO

DESCRIPTION: USE EWC WITH ZEEVE'S PLATFORM THROUGH OUR API AND TOOLS. OUR DOCUMENTATION PROVIDES TECHNICAL DETAILS ON AUTHENTICATION, MAKING REQUESTS AND HANDLING RESPONSES FOR INTERACTING WITH EWC AND THE ZEEVE PLATFORM.

CHAPTER

TWENTYTHREE

EWC VALIDATOR NODE SETUP

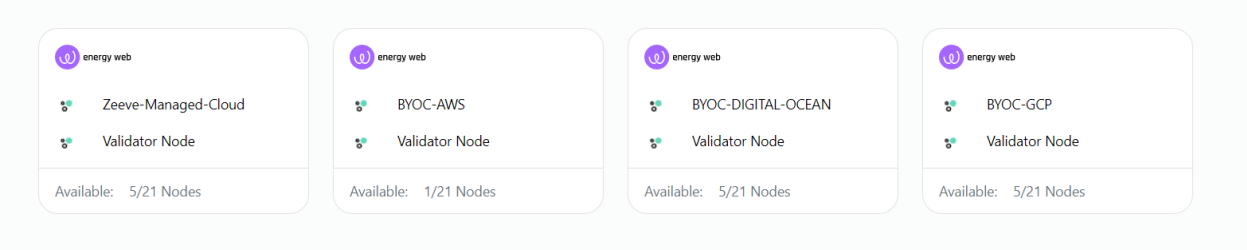
NOTE *Purchase* a subscription before proceeding.

1. *Create a network*
2. *Download Installation Summary*
3. *Withdrawal and Address Change*
4. *View Transaction*
5. *View On Explorer*
6. *Analytics*
7. *Alerts Section*
8. *Delete a network*

23.1 Create a network

This section will provide you detailed steps for creating a network of **EWC**.

On the **Network Configuration** page you will be able to see different configuration cards for EWC, which looks similar to the image provided below.

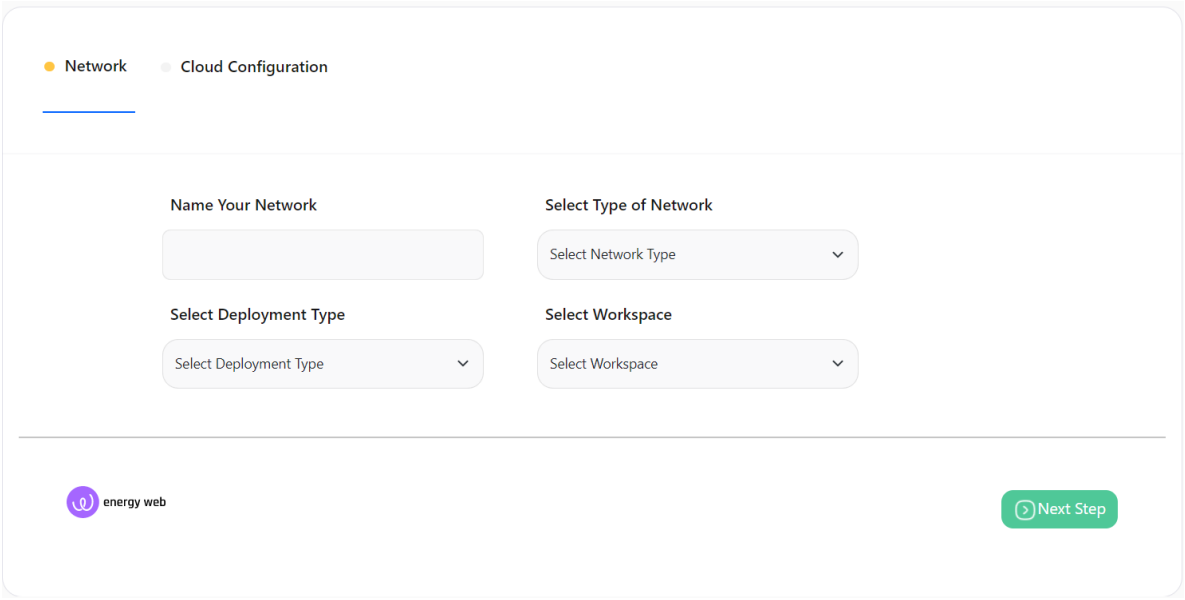


***NOTE:** These configuration cards can be different based on your purchased subscriptions.**

You can Choose **Zeeve Managed Cloud** or you can use your cloud account (AWS/DO/GCP/Tencent Cloud) for the infrastructure of your node.

Choose the card with the configuration you want. Clicking on the card you will be redirected to the network setup page.

1. **Network Info**



   - **Network Name**: A name to identify your network.
   - **Deployment Type**: Deployment type
   - **Network Type**
     - **Energy Web**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade EWC dapps.
     - **Volta**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
   - **Workspace**: This represents the workspace in which the network will be added after the successful creation.

   Proceed further by clicking on the **Next Step** button after providing all the details.

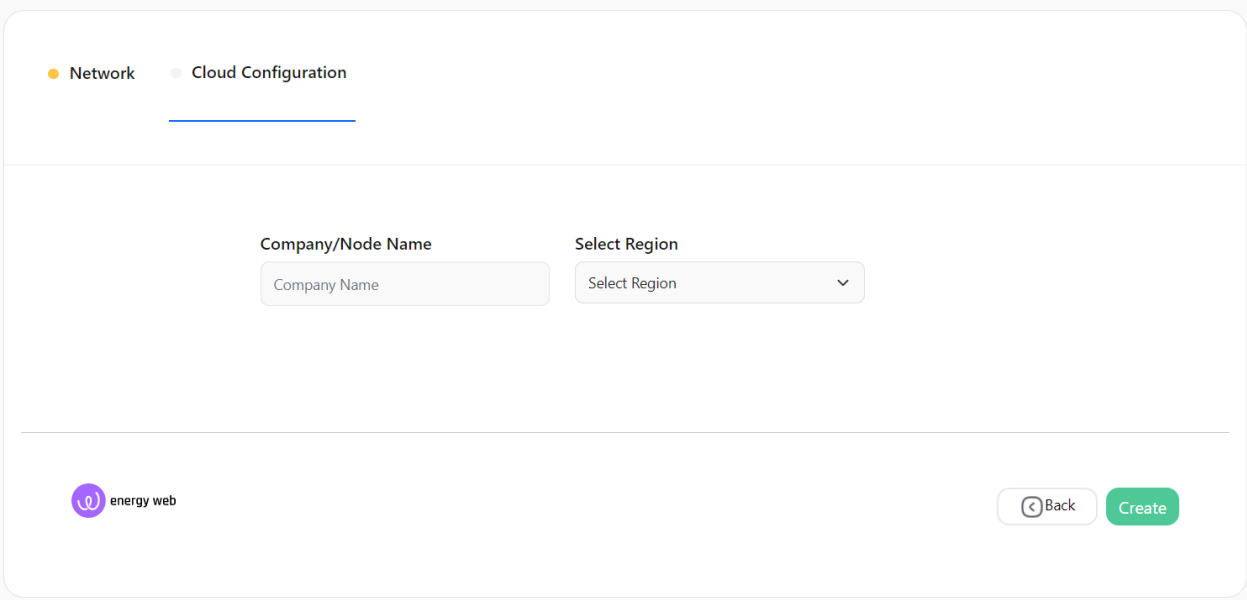
2. **Cloud Configuration**

   This step configures the cloud settings for your node. This step can vary based on your selection of **Network configuration card**

   1. *Zeeve Manged Cloud*
   2. *Bring Your Own Cloud (BYOC)*

23.1.1 Zeeve Managed Cloud

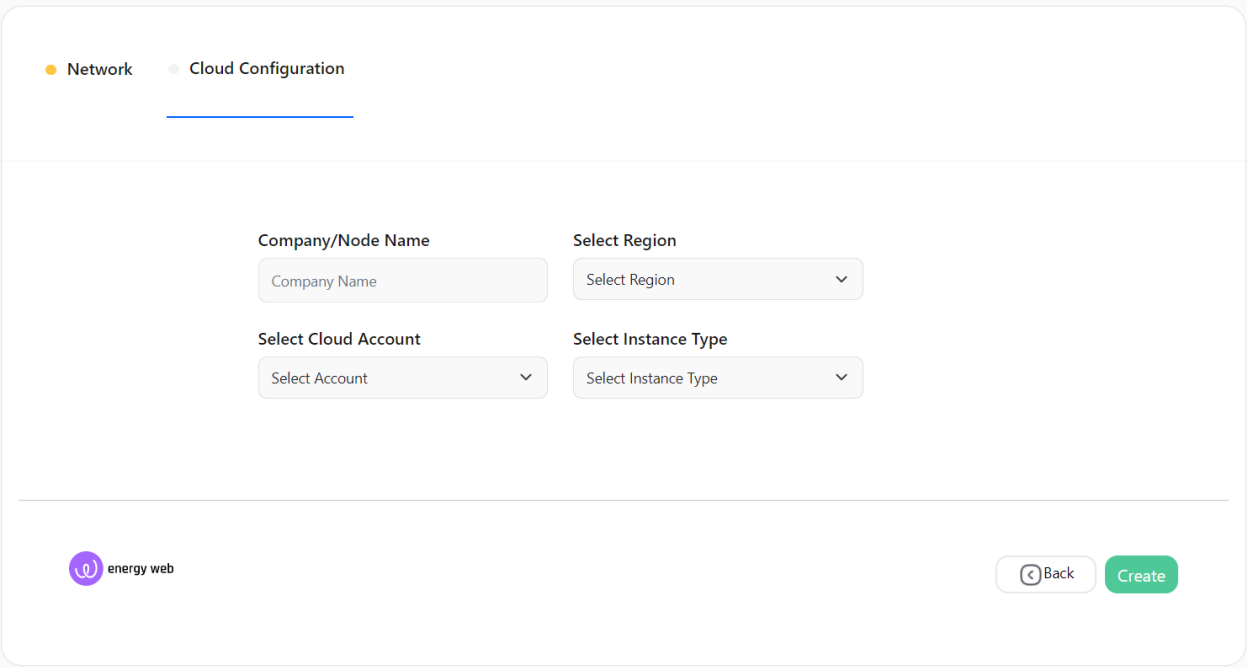
In the case of **Managed - Cloud**, select the region for the network under **Select Region** and provide a name to your node.



- **Company/Node Name**: This is the name given to the validator node.
- **Region**: It indicates the region of the cloud service. These regions are the geographic locations where your network instances are going to be hosted.

23.1.2 Bring Your Own Cloud

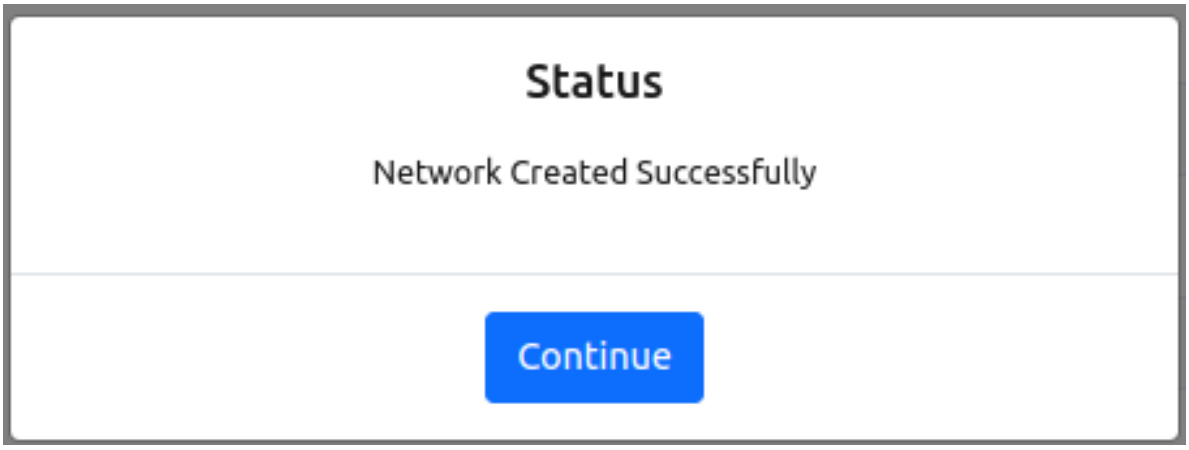
In the case of **BYOC** (AWS/Digital Ocean/GCP/Tencent Cloud), select the region for the network by clicking on **Select Region**, select the *Cloud* account you want to use by clicking on **Select Cloud Account**, choose the instance type as your requirement by clicking on **Select Instance Type** and provide a name to your node.



- **Company/Node Name**: This is the name given to the validator node.
- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted.
- **Cloud Account**: It represents the cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

---

1. On clicking the **Create** button a pop-up window will open which ensures the successful creation of your network.



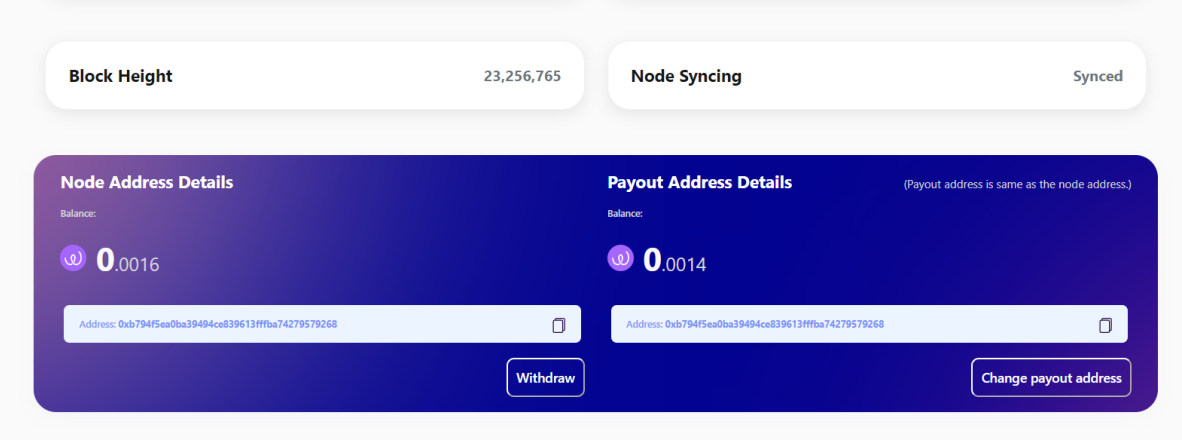
2. On clicking the **Continue** button you will be redirected to the page where you can see the network you created.

NOTE To become an EWF (Volta/EWC) validator, please proceed with the multi-sig process.

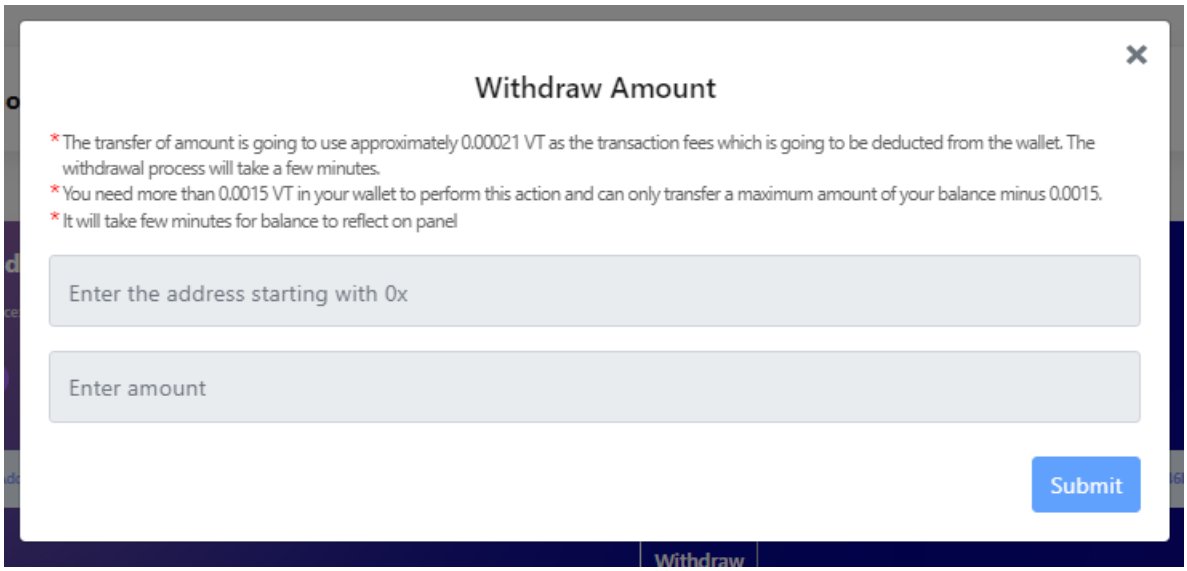
23.2 Withdrawal and Address Change

23.2.1 Amount Withdraw

1. In the case you want to withdraw amount, click on the withdraw button in the Node Address Details panel.



2. On clicking the **Withdraw** button a pop-up window will open.



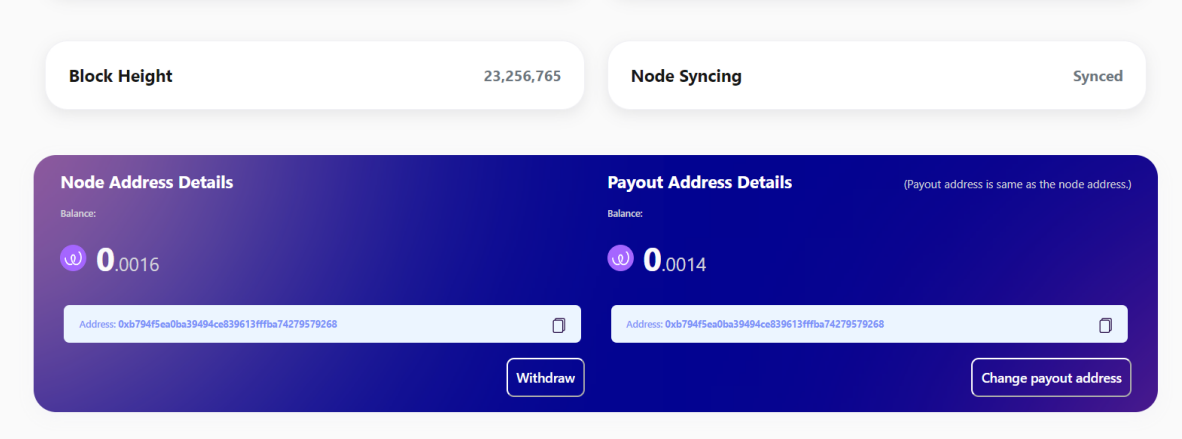
- **Address**: Enter the address where you want to withdraw the amount.
- **Amount**: Enter the amount you want to withdraw. The amount should be more than **0.0015**.

---

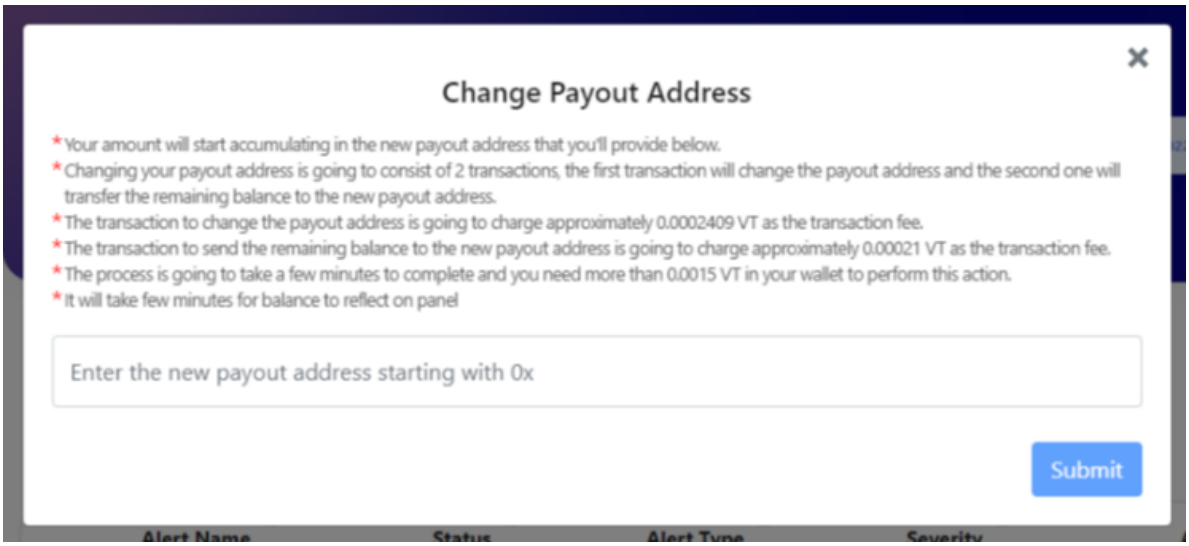
1. On clicking the **Submit** button to make the transaction. The amount withrawal will take few minutes to complete the transaction.

23.2.2 Address Change

1. In the case you want to change Address, click on the **Change Payout Address** button in the Payout Address Details panel.



2. On clicking the **Change Payout Address** button a pop-up window will open.

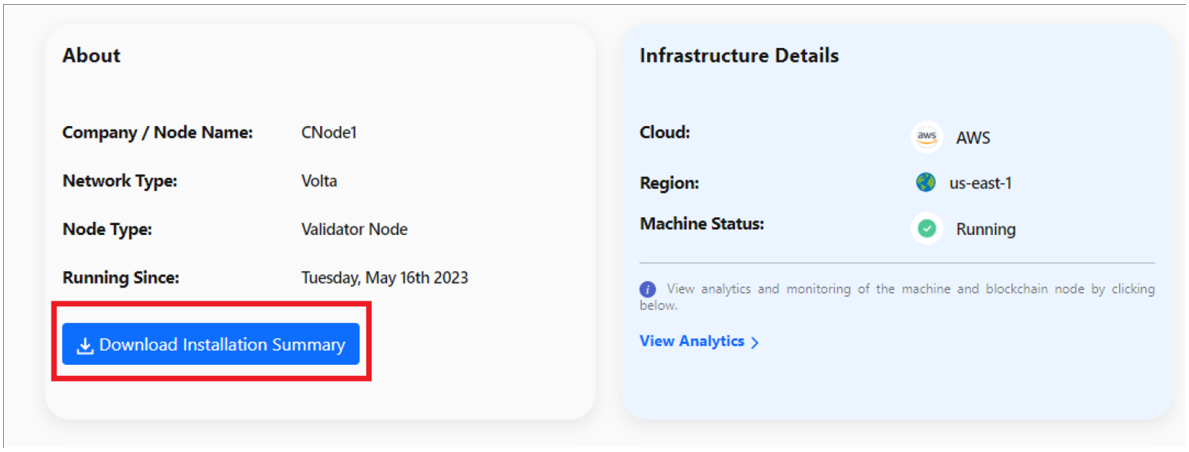


   - **Address**: Enter the new payout address.

3. On clicking the **Submit** button the Payout Address will be changed.

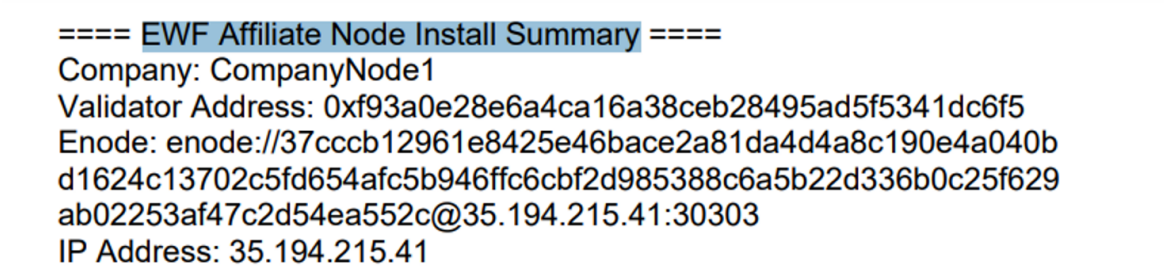
23.3 Download Installation Summary

1. To Download Installation Summary. Go to the detail page and click on the **Download installation summary** button.



2. On clicking the **Download installation summary** button, a PDF will be downloaded with the EWF Node Install information.
   - **Node Name**.
   - **Validator Address**.
   - **Enode**.
   - **IP Address**.

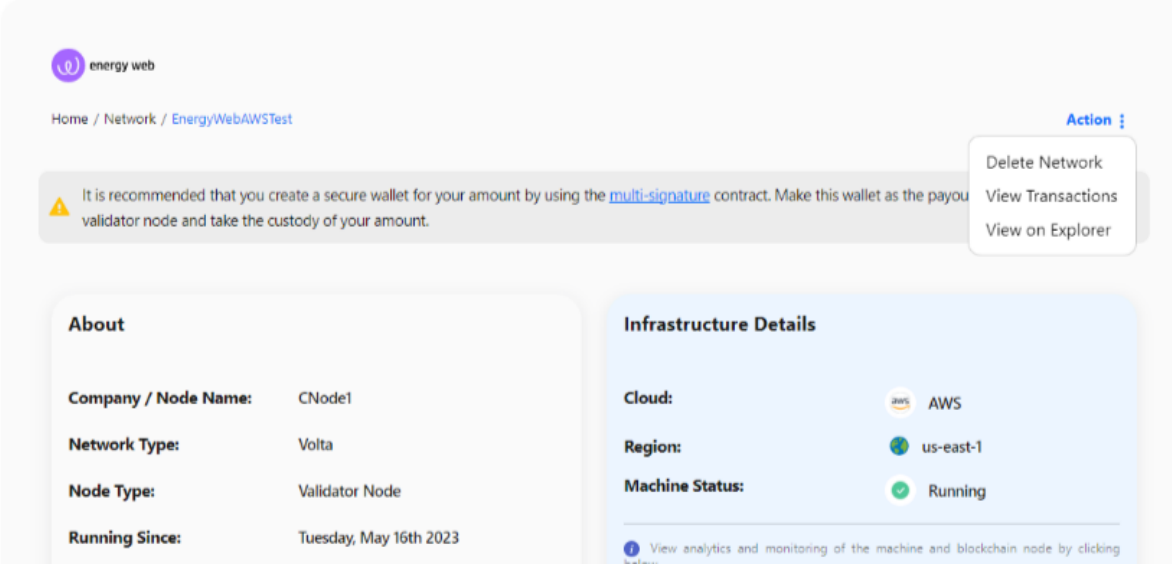
   This information will be displayed in a format similar to the below image



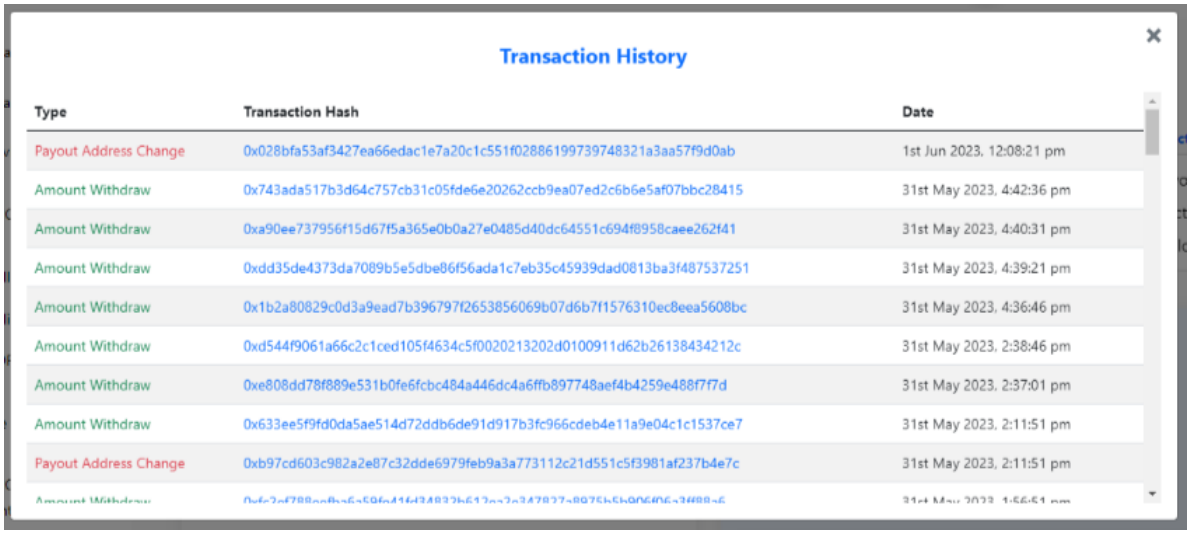
NOTE Download option will be available when a node is In the Active stage.

23.4 View Transaction

1. To view all the transaction. Click on the Actions button on the top right, and select the **View Transaction** option.

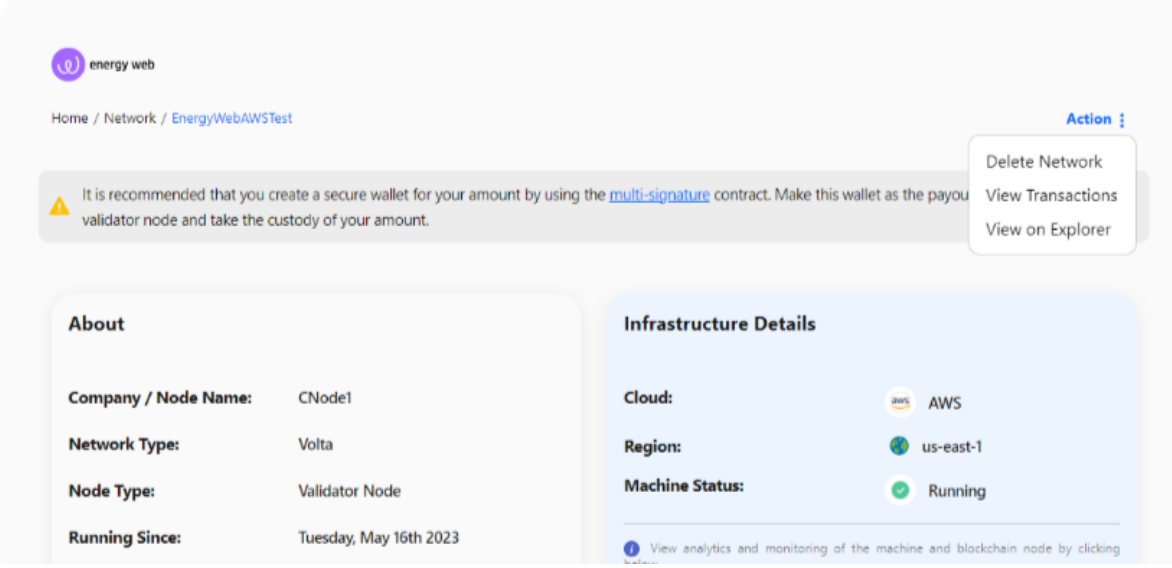


2. On clicking the **View Transaction** button a pop-up window will open.



23.5 View On Explorer

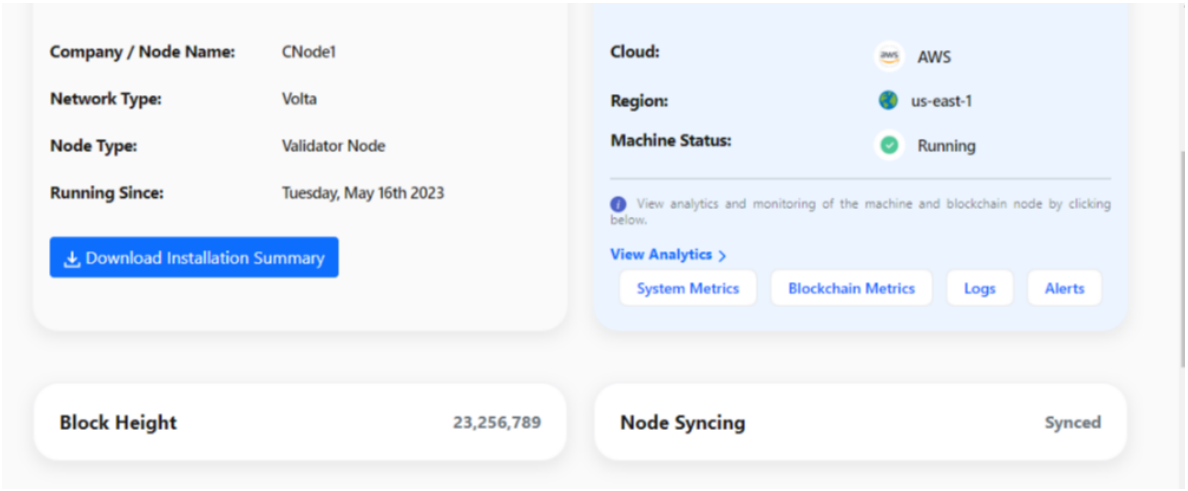
1. To view all Address details on the EWC Explorer. Click on the Actions button on the top right, and select the **View On Explorer** option.



2. On clicking the **View On Explorer** button a new tab will open.

23.6 Analytics

1. To view network analytics and monitoring details, click on the **View Analytics** in the Infrastructure Details panel.

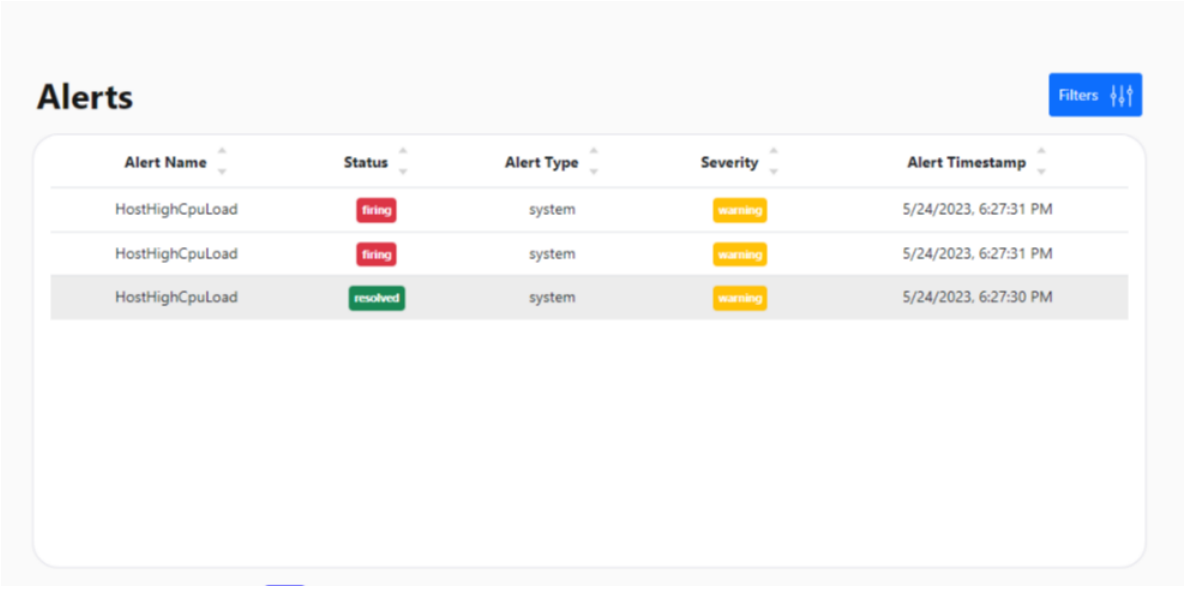


2. On clicking the **View Analytics** hyperlink, four button will be displayed.
    - System metrics

- Blockchain metrics
- Logs
- Alerts

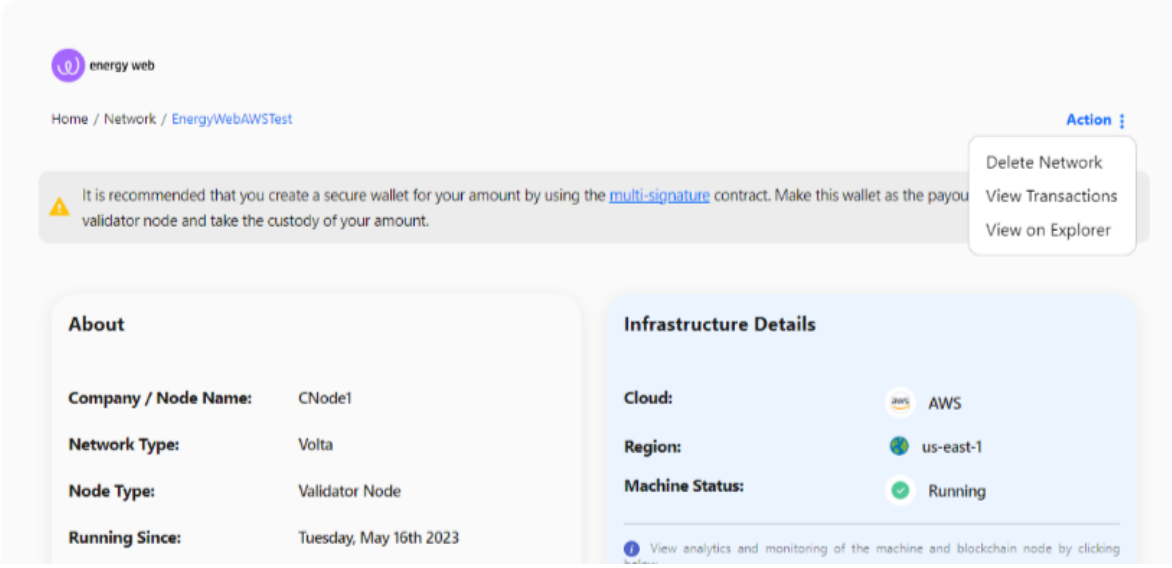
23.7 Alerts Section

1. You will see Alerts sections at the bottom of the page. Here you can see the list of Alerts related to you EWC network.

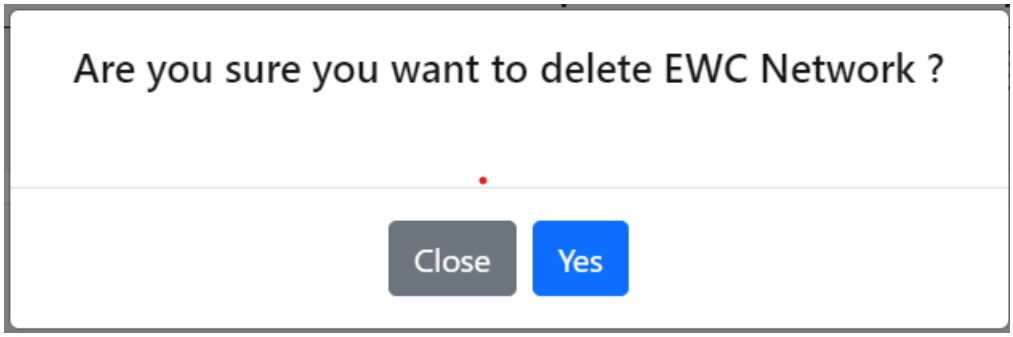


23.8 Delete a network

1. Visit the network detail page. Click on the *Actions* button on the top right, and select the **Delete Network** option.



2. A confirmation window will open, click on the **Yes** button to delete the network.



NOTE It will take a few minutes to delete a network.

---

description: Build decentralized applications using Fantom and Zeeve's platform. Our documentation provides technical details on authentication, making requests and handling responses for interacting with Fantom and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

TWENTYFOUR

FANTOM DEDICATED NODE SETUP

This section will guide you about different actions you can perform for Fantom

1. *Dedicated nodes*
2. *Api endpoints*

24.1 Dedicated nodes

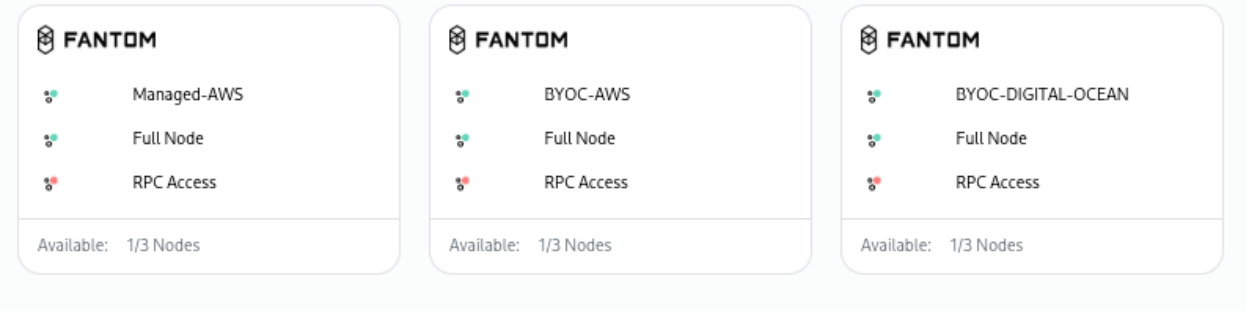
1. *Create network*
2. *Add a node*
3. *Delete a node*
4. *Delete network*

24.1.1 Create a network

This section will provide you with detailed steps for creating a network of **Fantom**.

NOTE Please make sure to follow the steps mentioned earlier before proceeding.

On the **Network Configuration** page you will have different cards with different network configurations for Fantom, which looks similar to the image provided below.

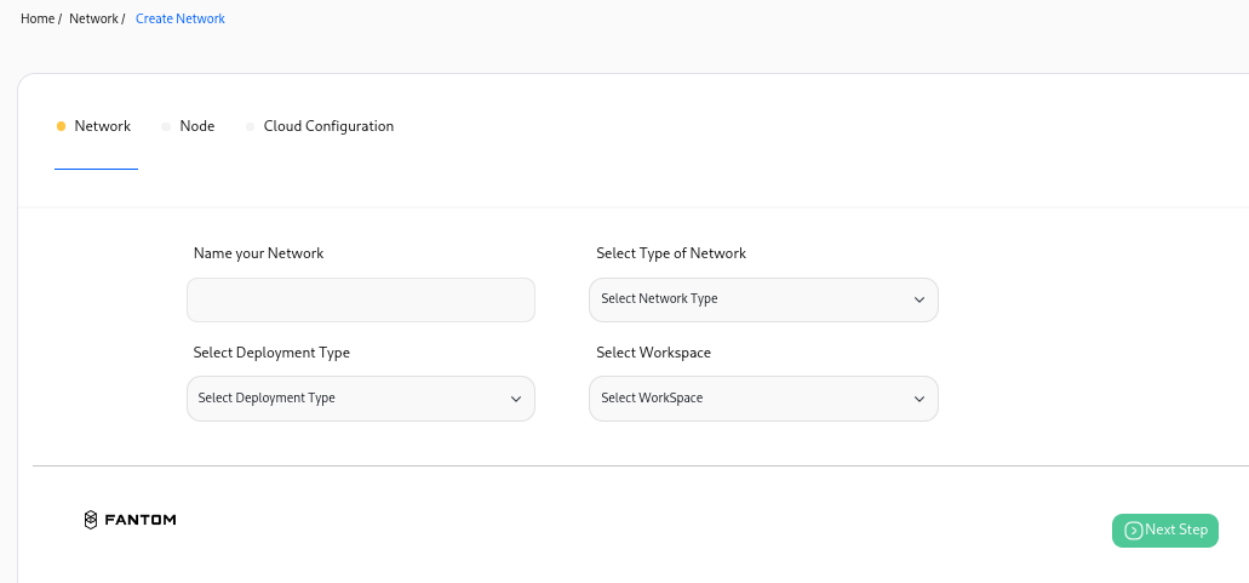


NOTE: *These cards can be different for your case. Card configurations depend on your purchased subscription.*

You can Choose **Managed-AWS** (Zeeve's managed hosting) for the infrastructure of your node or you can use your cloud account (AWS/DO) for the hosting of your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

1. **Network Info** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.

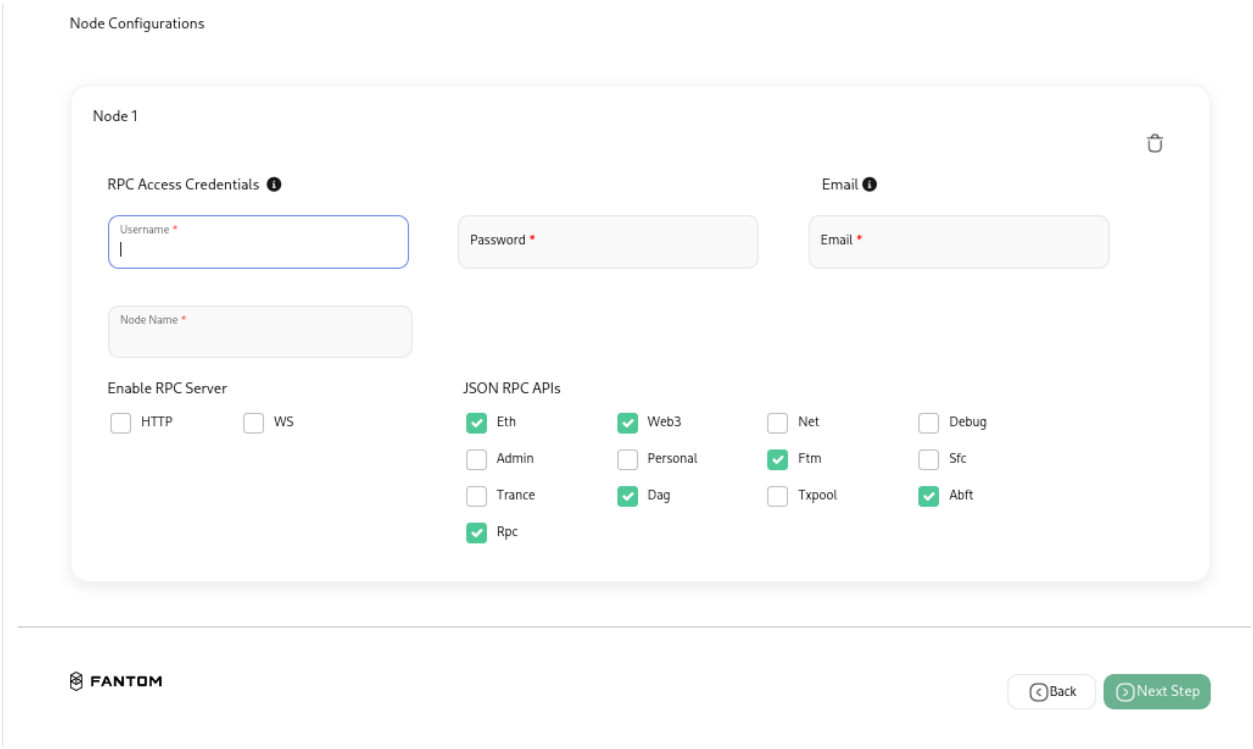


- **Name of Network**: To uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: This defines the deployment type.
- **Type Of Network**
  - **MainNet**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade Fantom dapps.
  - **Testnet**: This is a testnet you can use for your non-production needs like testing or demonstrations.
- **Workspace**:

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration**

In this section, you have to fill in the details of the nodes you want to add to your network.



- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Email**: Fill email of yours.
- **Node Name** In order to identify your nodes, this field will be used.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.
- **JSON RPC APIs**: JSON RPC API is a bridge that allows dApps to connect to nodes.

After providing the details click on **Next step** to go to the last step.

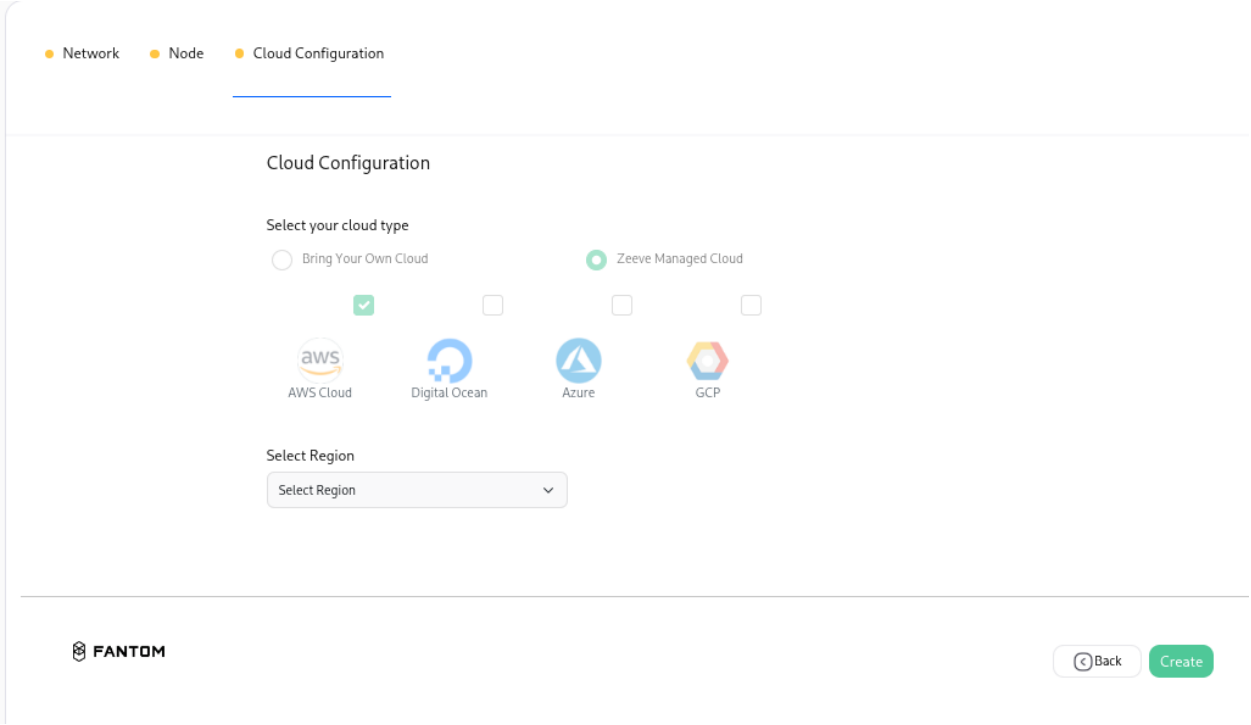
1. **Cloud Configuration**

This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**

1. *Manged - AWS*
2. *BYOC - AWS*
3. *BYOC - DO*

Managed - AWS

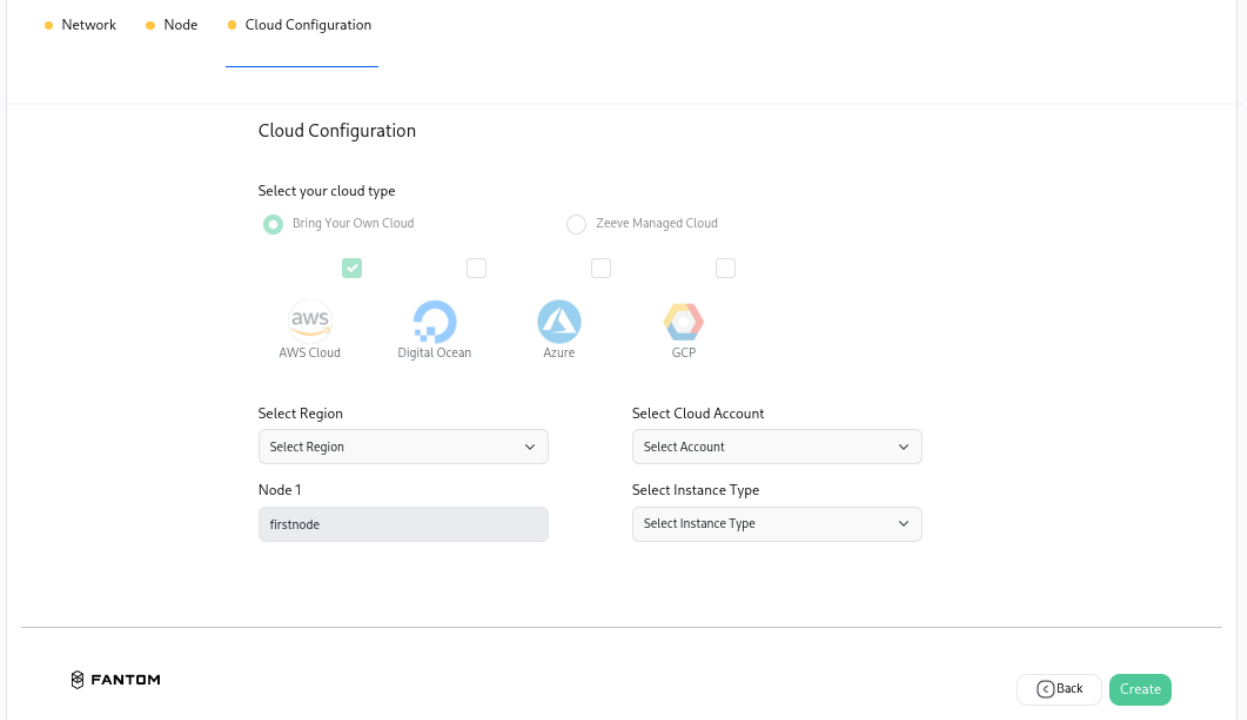
In the case of **Managed - AWS**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.

BYOC - AWS

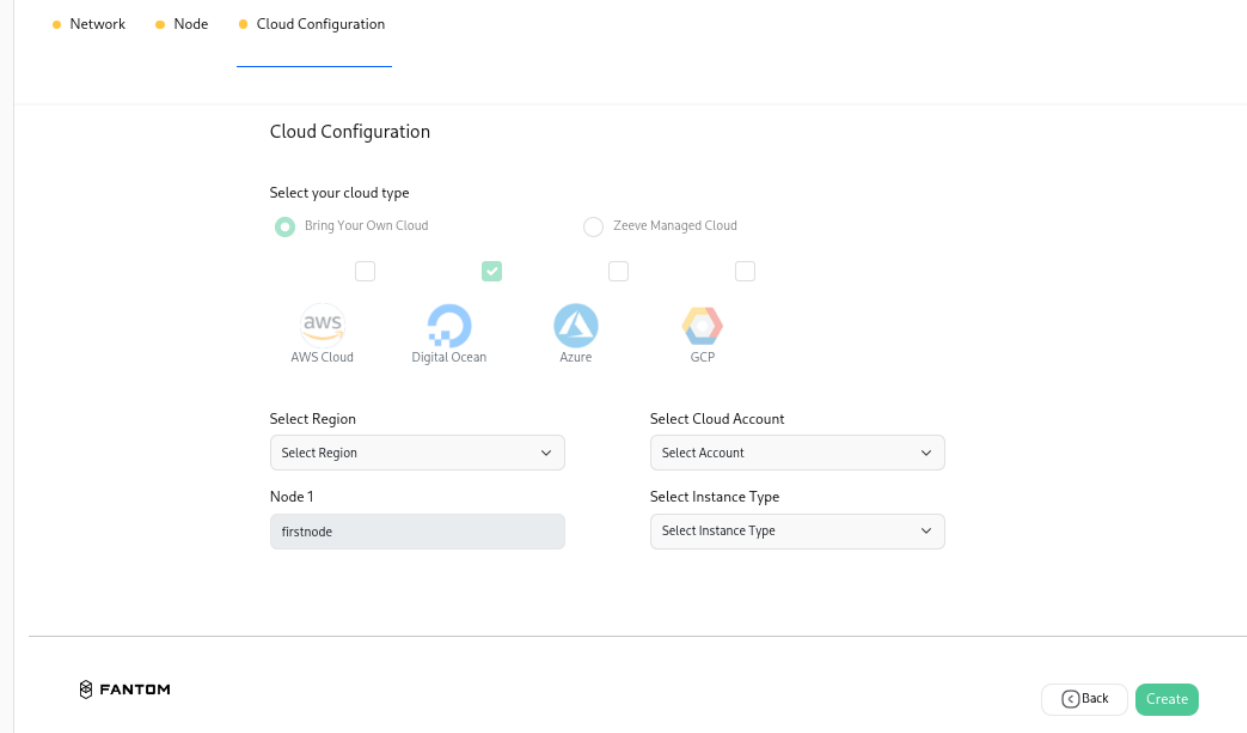
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

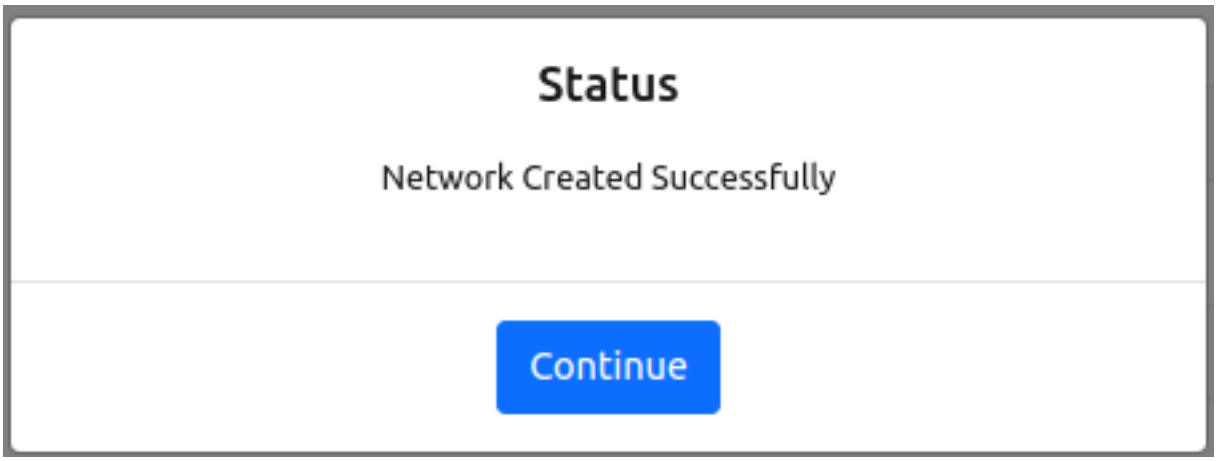
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.

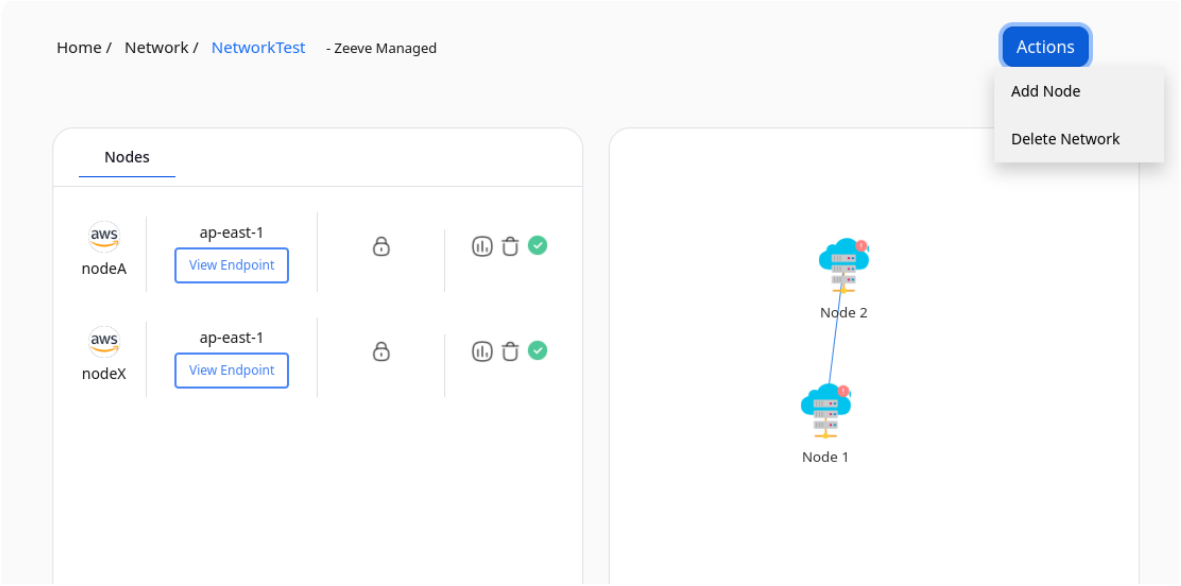


1. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes listed you've just added to the network.

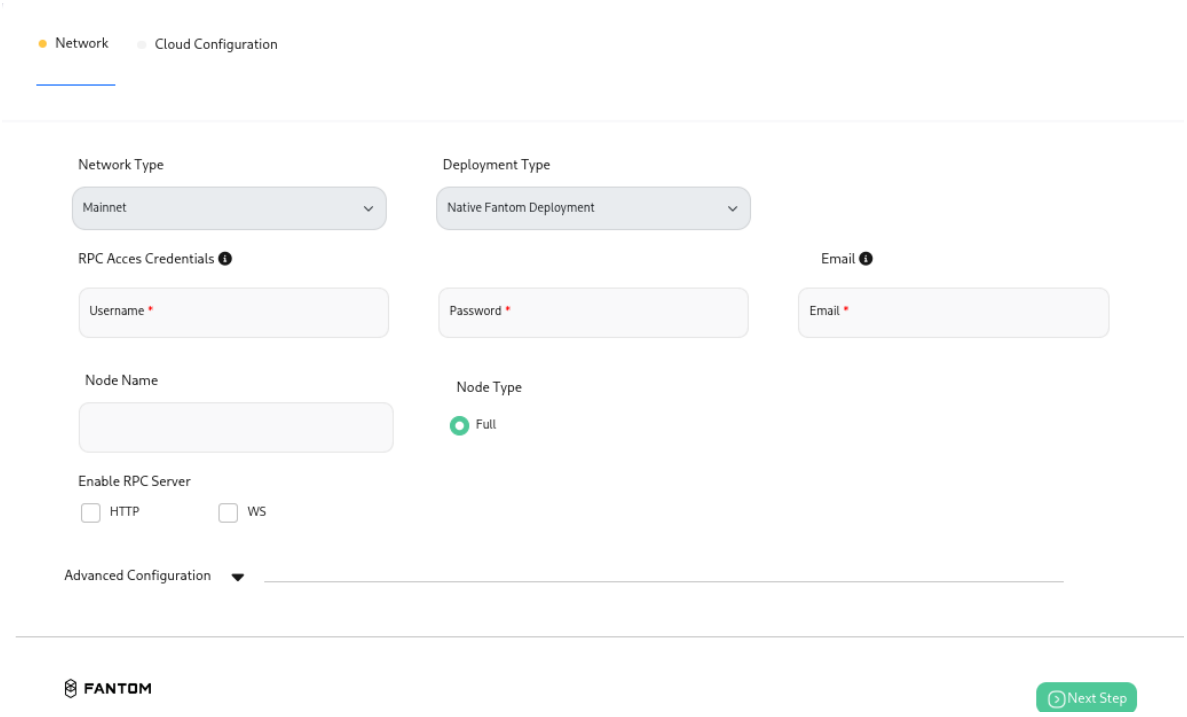
24.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

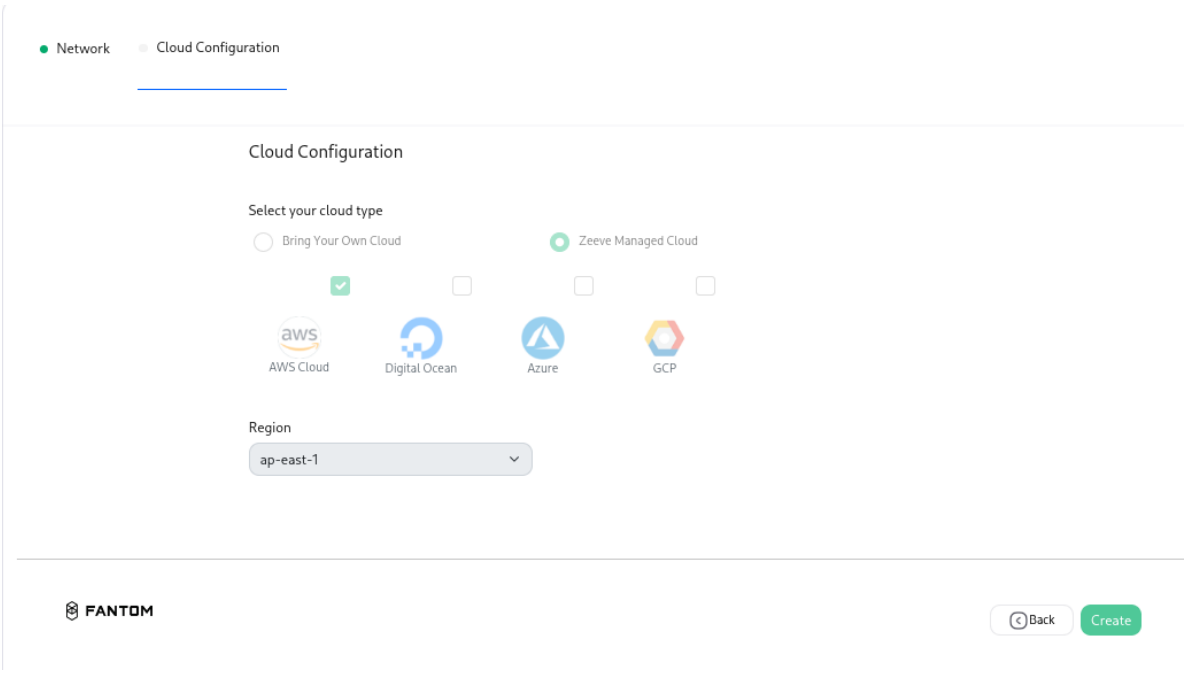
1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.

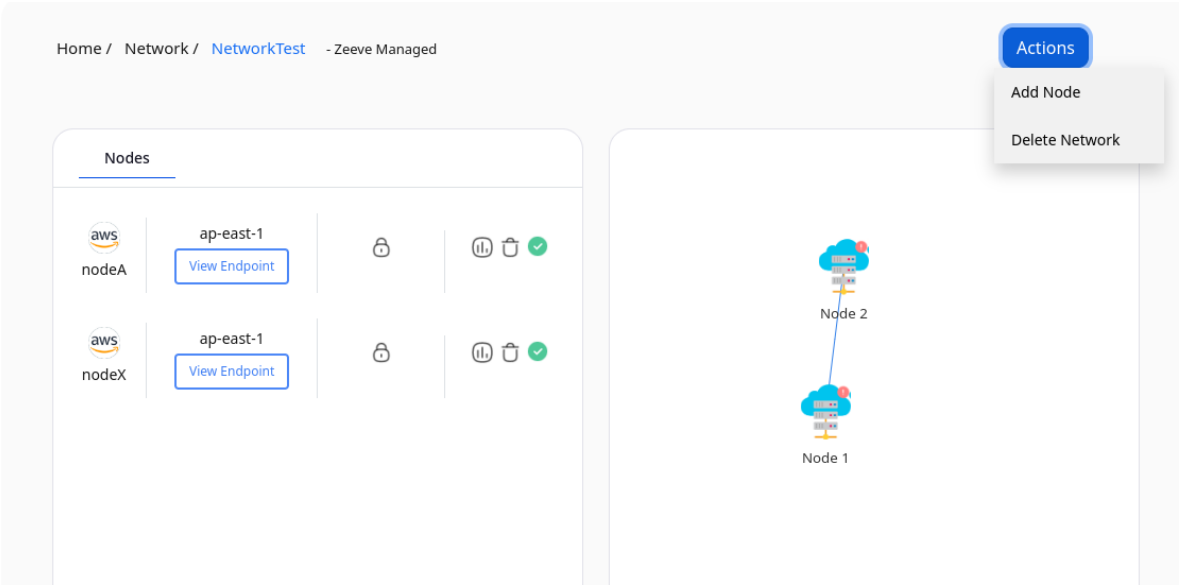


3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region will be prefilled according to the configuration of the network. Click on the **Create** button and it's done!



24.1.3 Delete node in a network

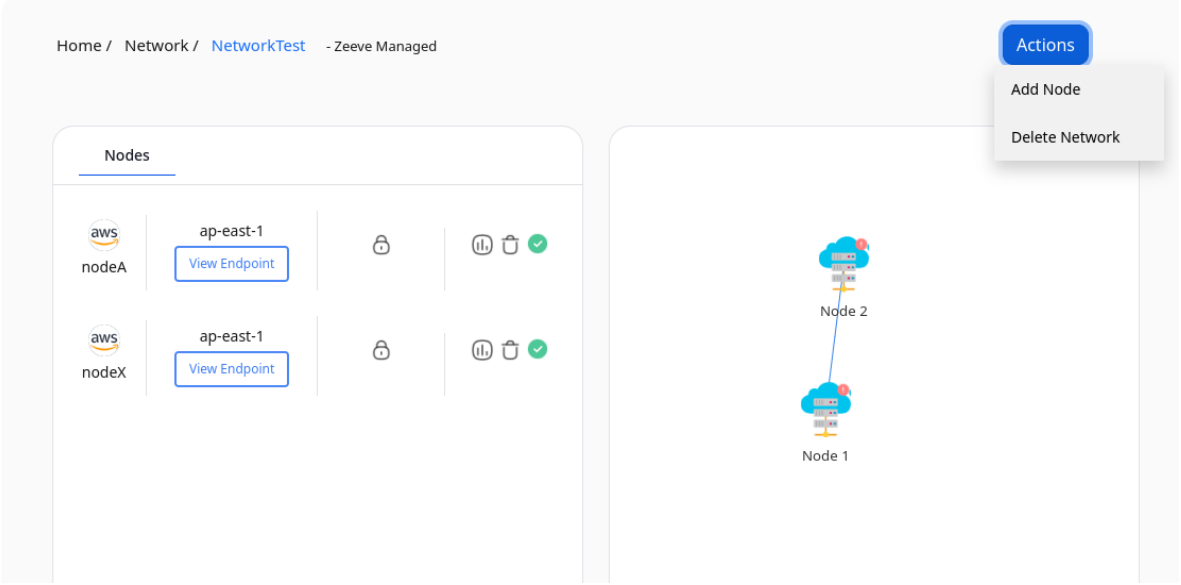
1. Select the network, in which you want to perform delete a node, and click on the network card Ref.. You will get to see similar to the below image.



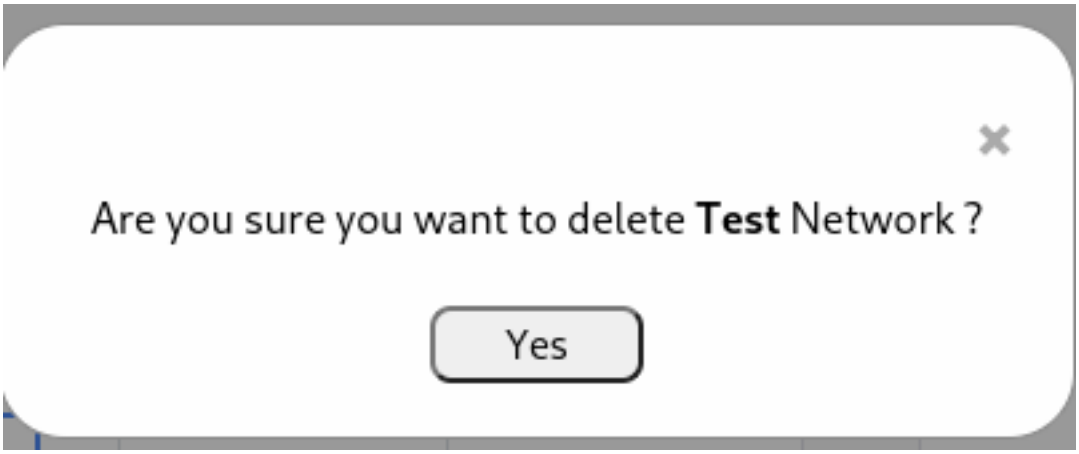
2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **yes** button to confirm.

24.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

---

description: Integrate Polygon with Zeeve's platform using our API and tools. Our documentation provides technical details on authentication, making requests and handling responses for interacting with Polygon and the Zeeve platform.

meta:

- name: robots content: noindex

CHAPTER

TWENTYFIVE

POLYGON DEDICATED NODE SETUP

This section will guide you about different actions you can perform for Polygon

1. *Dedicated nodes*
2. *Api endpoints*

25.1 Dedicated nodes

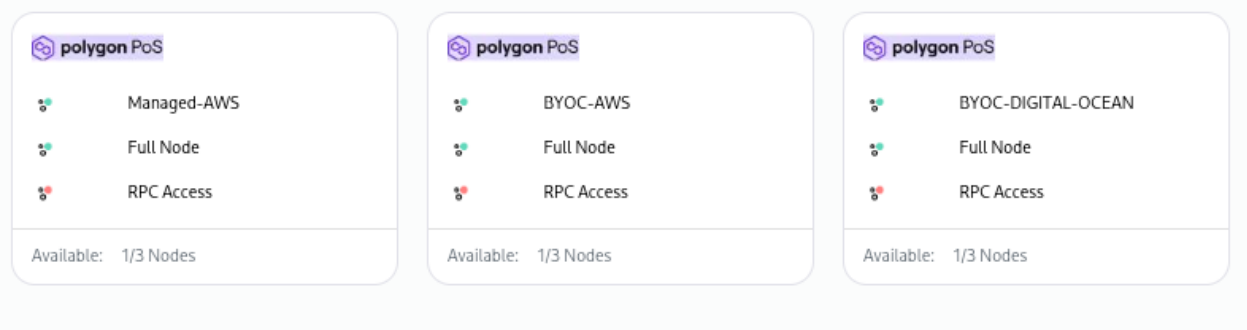
1. *Create network*
2. *Add a node*
3. *Delete a node*
4. *Delete network*

25.1.1 Create a network

This section will provide you with detailed steps for creating a network of **Polygon**.

NOTE Please make sure to follow the steps mentioned earlier before proceeding.

On the **Network Configuration** page you will have different cards with different network configurations for Polygon, which looks similar to the image provided below.

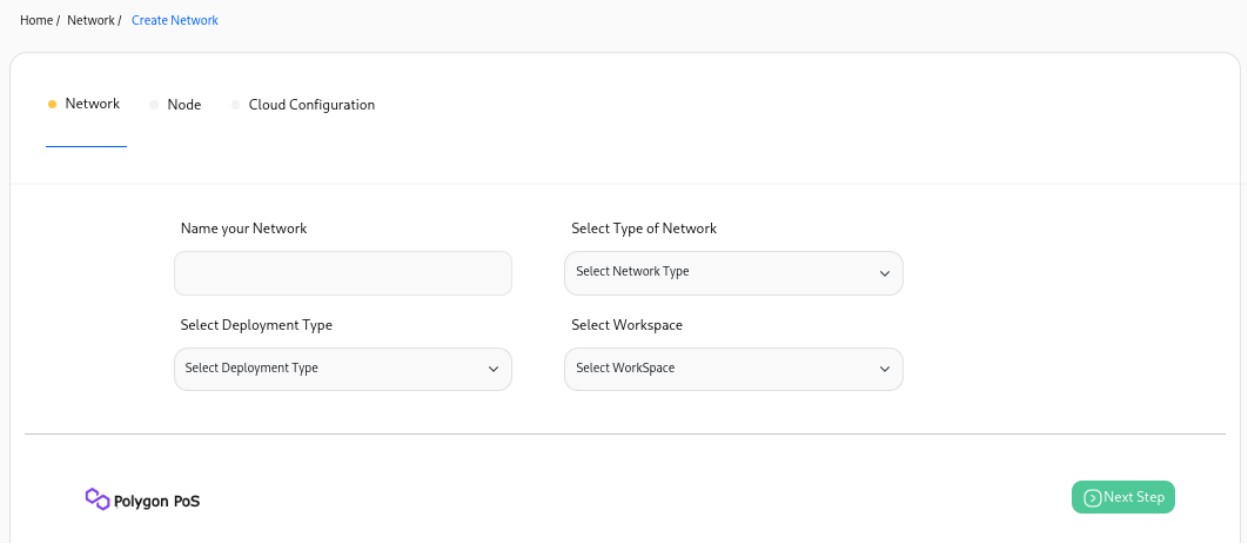


NOTE: *These cards can be different for your case. Card configurations totally depend on your purchased subscription.*

You can Choose **Managed-AWS** (Zeeve's managed hosting) for the infrastructure of your node or you can use your own cloud account (AWS/DO) for hosting your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

1. **Network Info** Clicking on the card you will be landed on a page similar to the below image. In this section, we have to provide network-related information for ex- Network Name, Network type, etc.

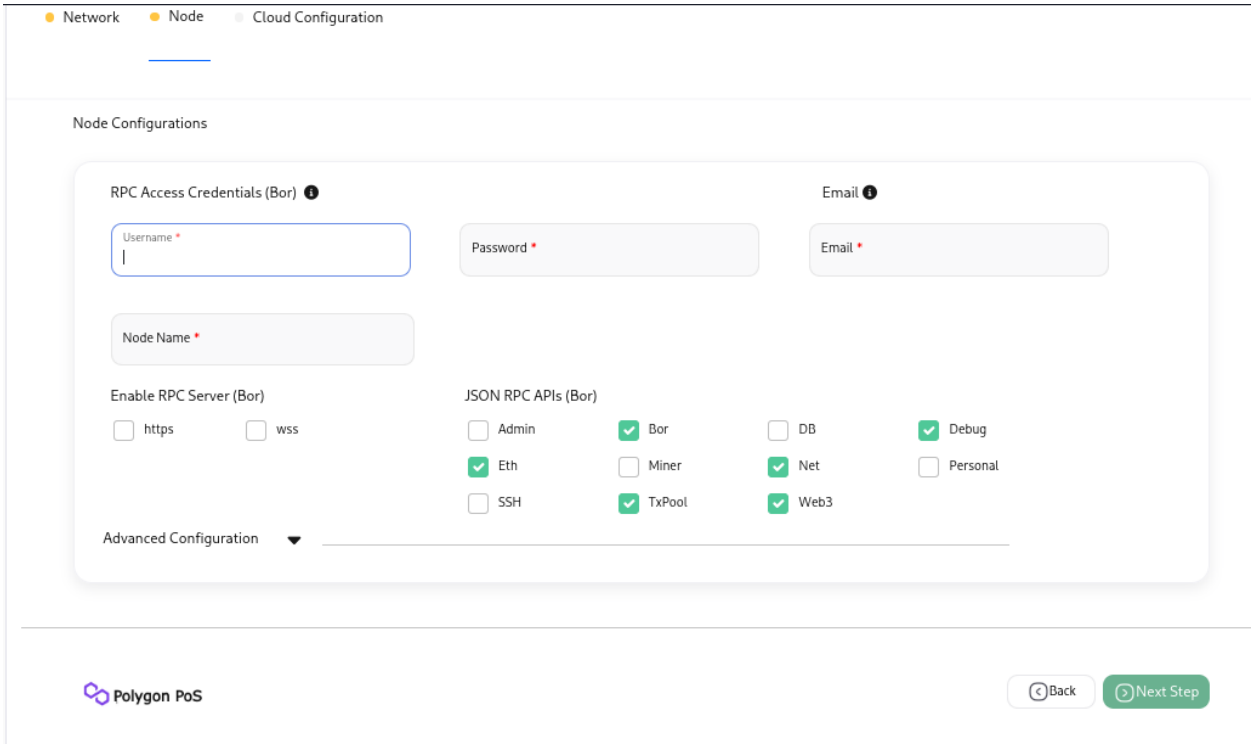


- **Name of Network**: In order to uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: Deployment type
- **Type Of Network**
  - **MainNet**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade Polygon dapps.
  - **TestNet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after successful creation.

After providing all the details correctly go to the next step by clicking on the **Next Step** button.

1. **Node Configuration**

In this section, you have to fill in the details of the nodes you want to add to your network.



- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Email**: Fill email of yours.
- **Node Name**: In order to identify your nodes, this field will be used.
- **Rpc Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.
- **JSON RPC APIs**: JSON RPC API is a bridge that allows dApps to connect to nodes.

After providing the details click on **Next step** to go to the last step.

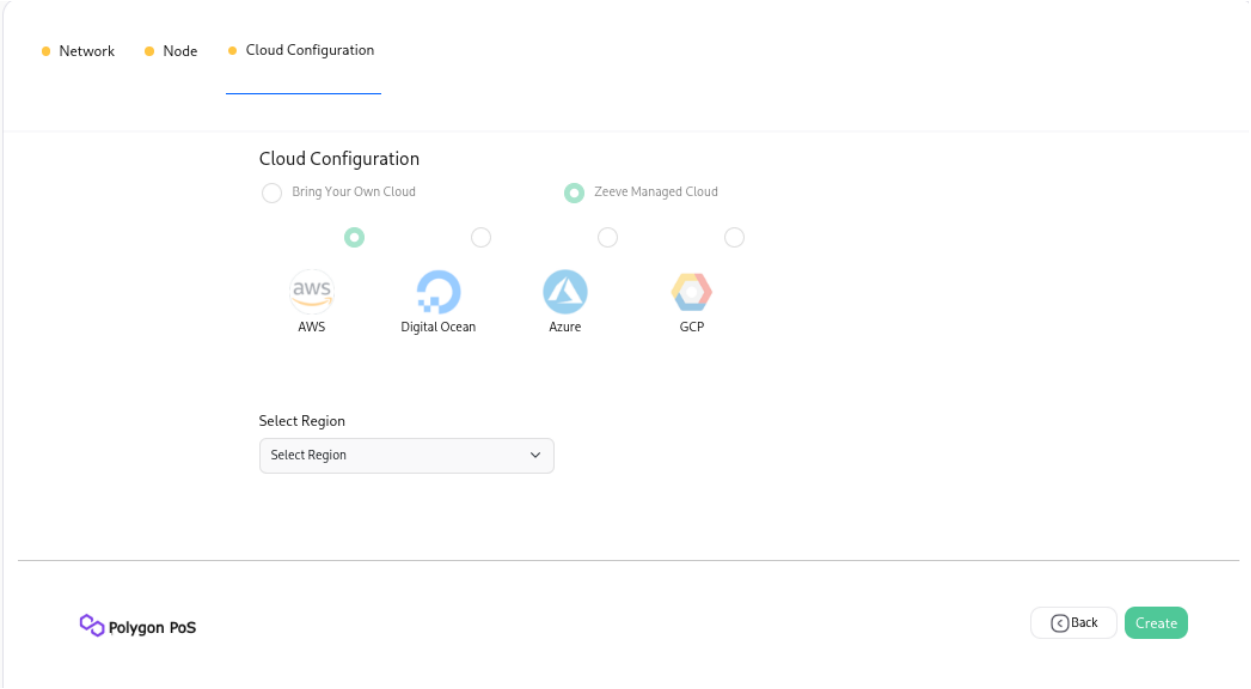
1. **Cloud Configuration**

This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**

1. *Manged - AWS*
2. *BYOC - AWS*
3. *BYOC - DO*

Managed - AWS

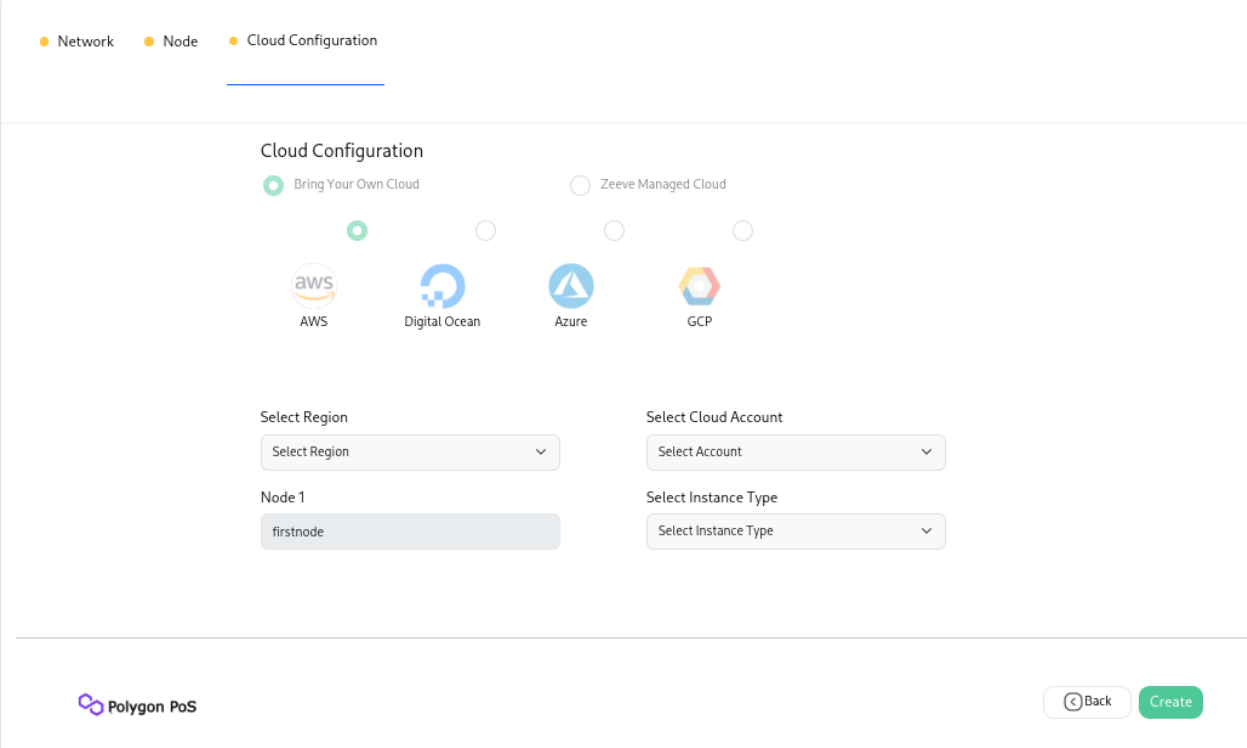
In the case of **Managed - AWS**, you don't have to bother about anything, just select the region for the network by clicking on **Select Region**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.

BYOC - AWS

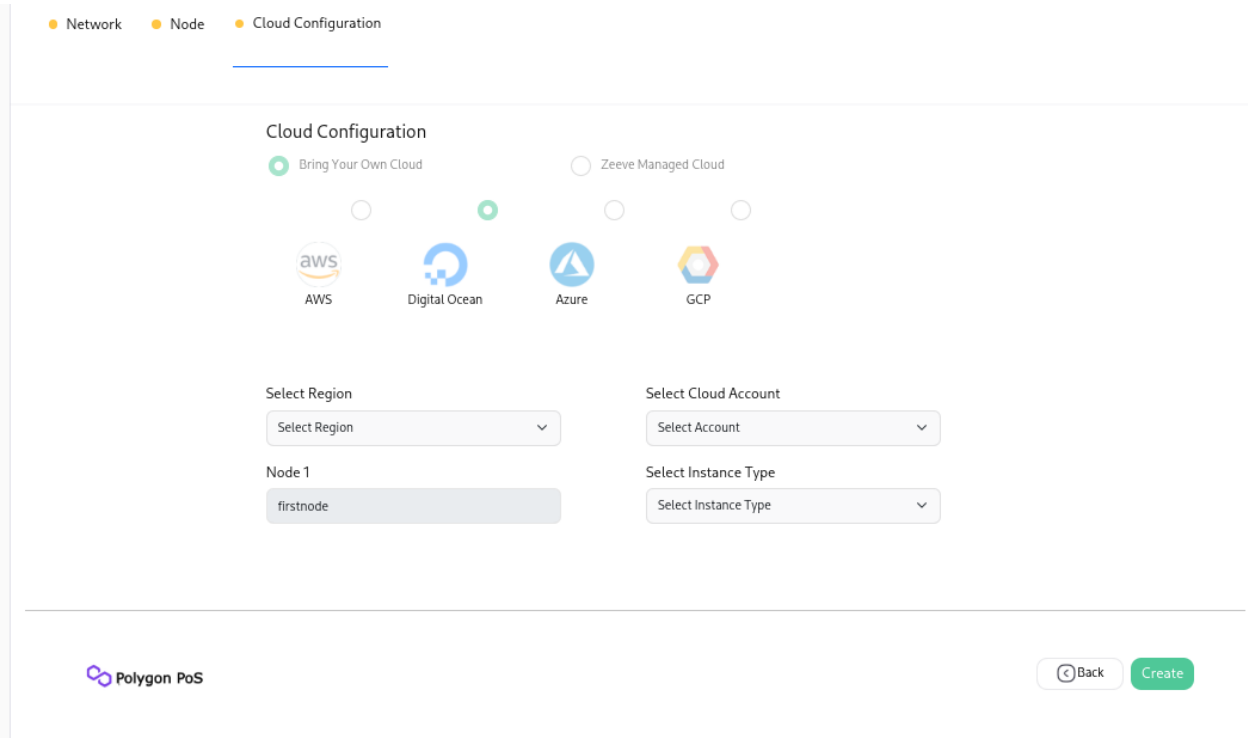
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

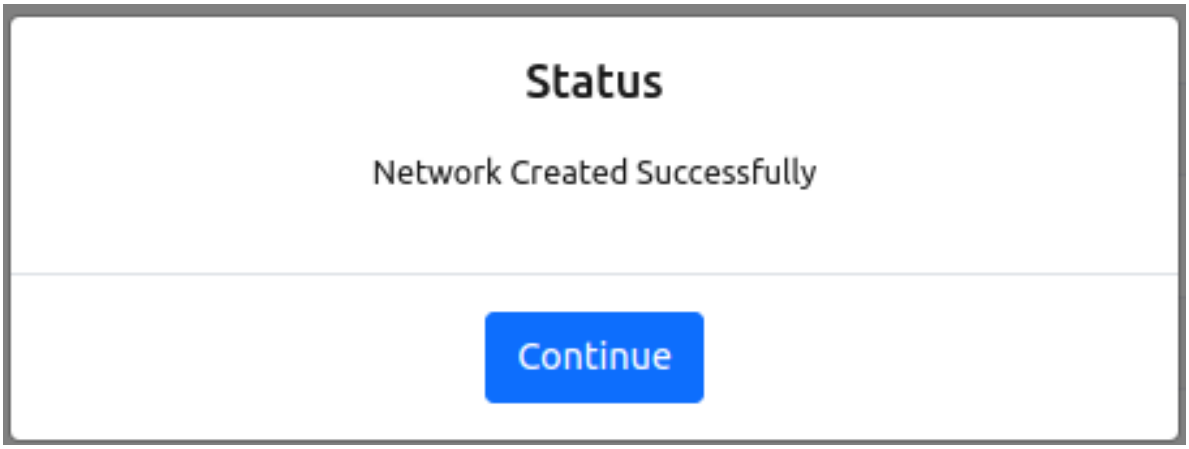
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple, isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.

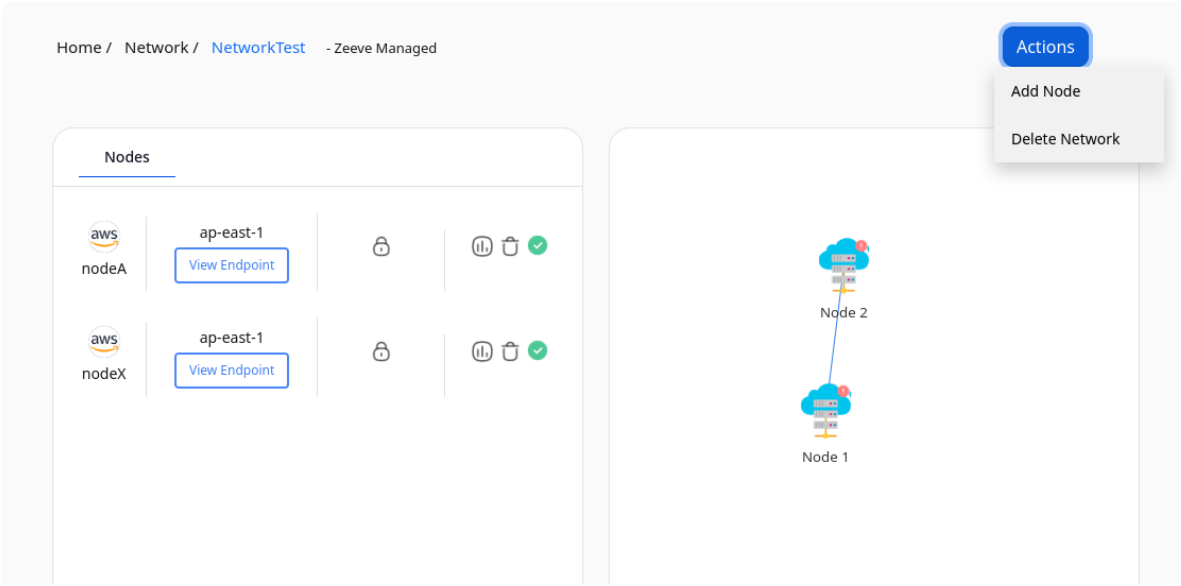


2. Click on **continue** and you will be redirected to a page similar to the below image where you can see the nodes listed you've just added to the network.

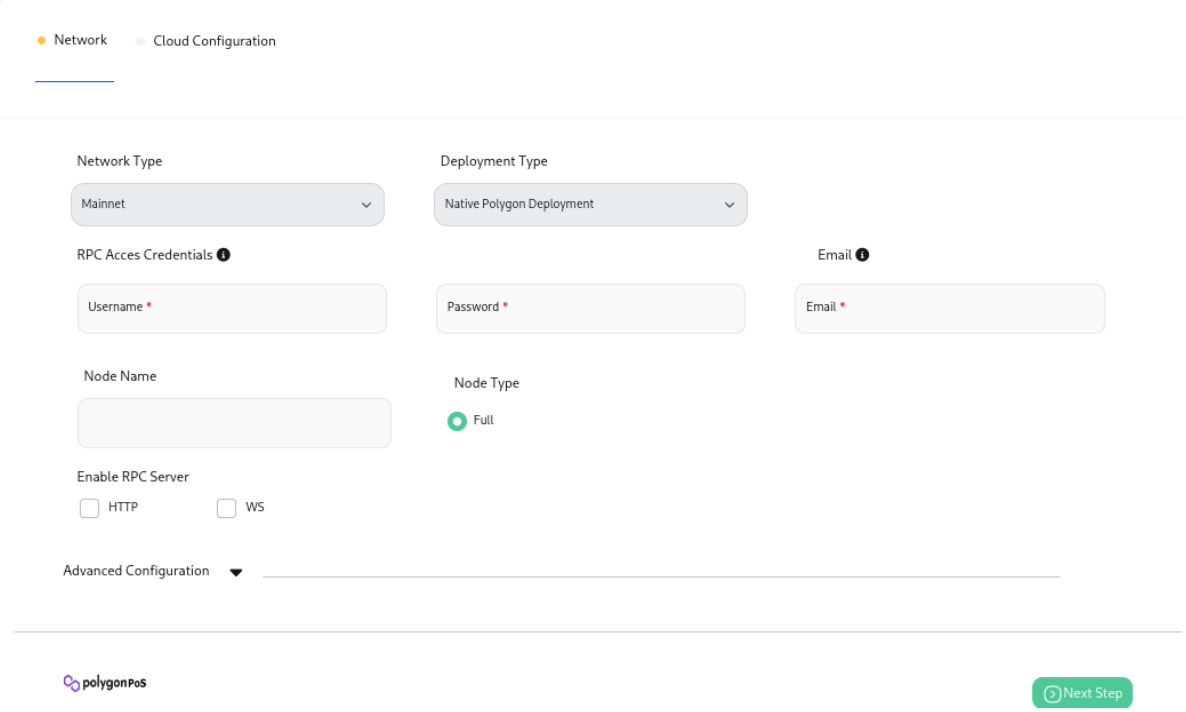
25.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

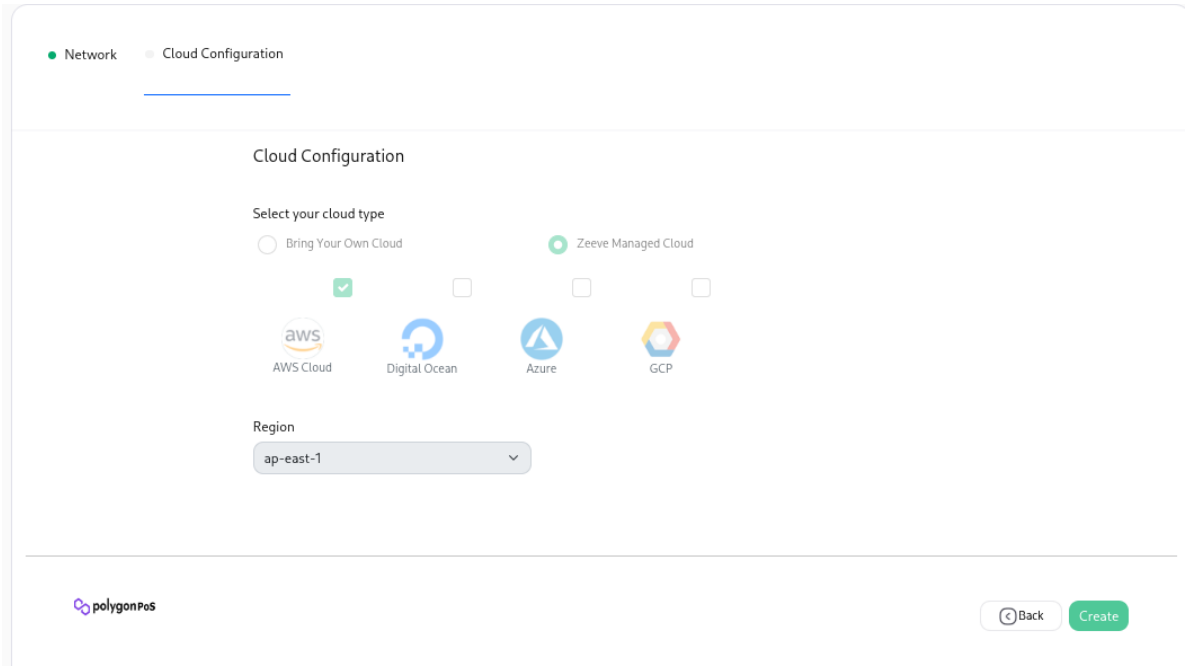
1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.

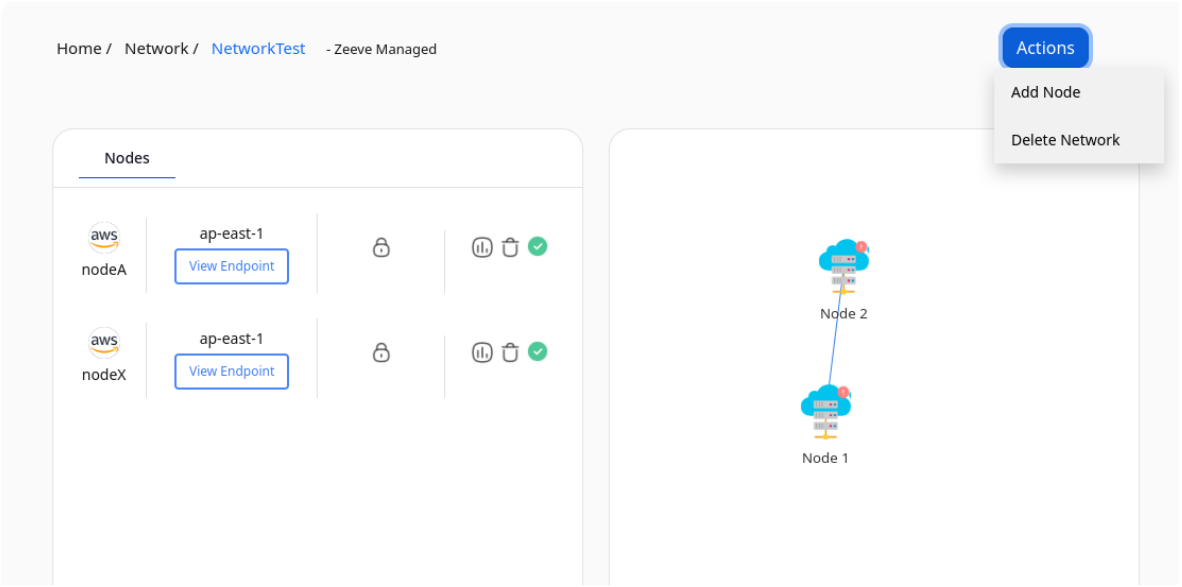


3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region will be prefilled according to the configuration of the network. Click on the **Create** button and it's done!



25.1.3 Delete node in a network

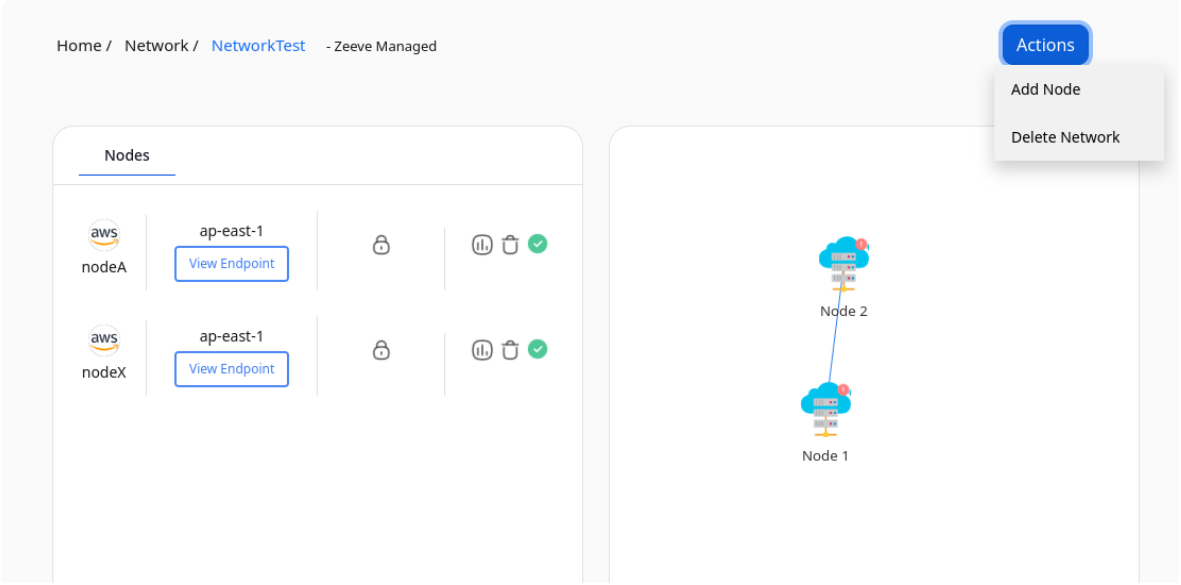
1. Select the network, you want to perform the deleted node to, and click on the network card Ref.. You will get to see similar to the below image.



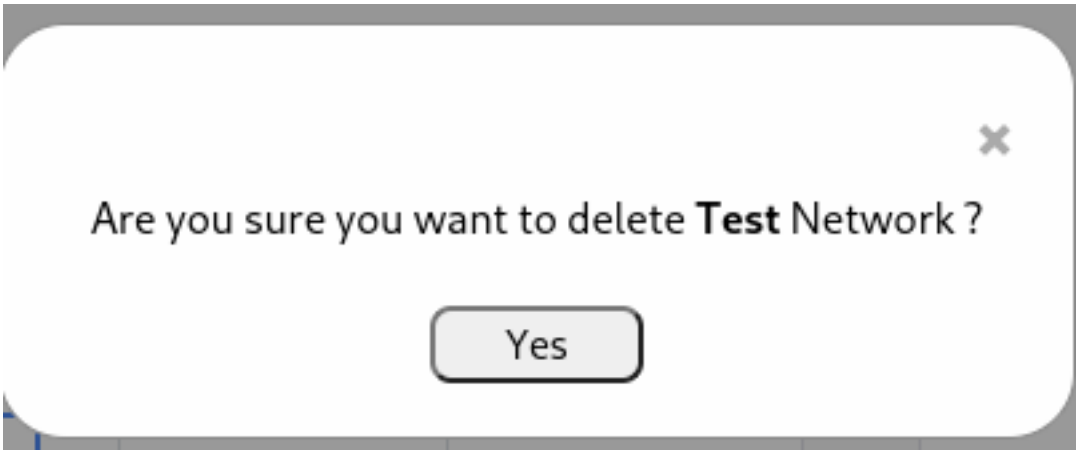
2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation, click on the **yes** button to confirm.

25.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Delete Network** option. A confirmation window will open, click on the **Yes** button, attached to it.



NOTE It can take a few minutes to delete a network.

---

meta:

- name: robots content: noindex

CHAPTER

TWENTYSIX

TRON DEDICATED NODE SETUP

This section will guide you about different actions you can perform for Tron.

1. *Dedicated nodes*
2. *API endpoints*

26.1 Dedicated nodes

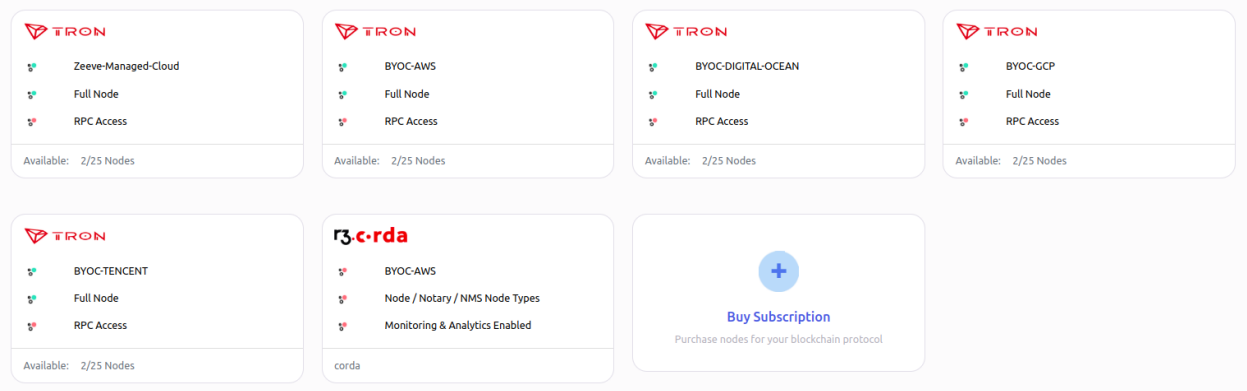
1. *Create network*
2. *Add a node*
3. *Delete a node*
4. *Delete network*

26.1.1 Create a network

This section will provide detailed steps for creating a network of **Tron**.

NOTE Please make sure to follow the steps mentioned earlier before proceeding.

On the **Network Configuration** page, you will have different cards with different network configurations for Tron, which looks similar to the image provided below.



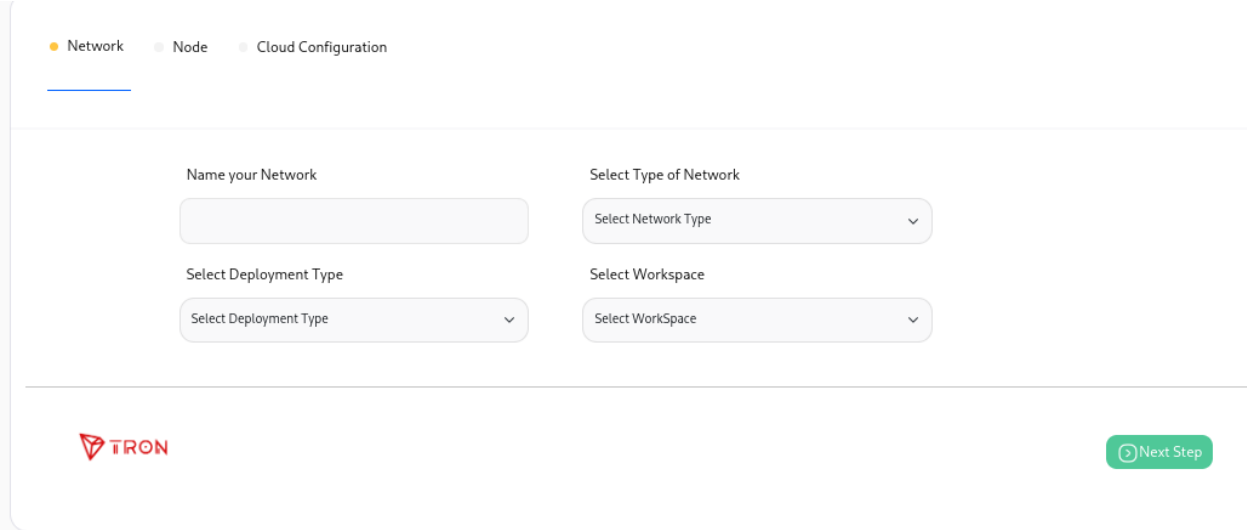
NOTE: These cards can be different for your case. Card configurations totally depend on your purchased subscription.

You can choose **Managed** (Zeeve's managed hosting) for the infrastructure of your node, or you can use your own cloud account (AWS/DO/GCP/Tencent) for hosting your node.

Choose the configuration you want. Click on the card and follow the steps accordingly.

1. **Network Info**

Clicking on the card, you will be landed on a page similar to the below image. In this section, we have to provide network-related information, ex- Network Name, Network type, etc.

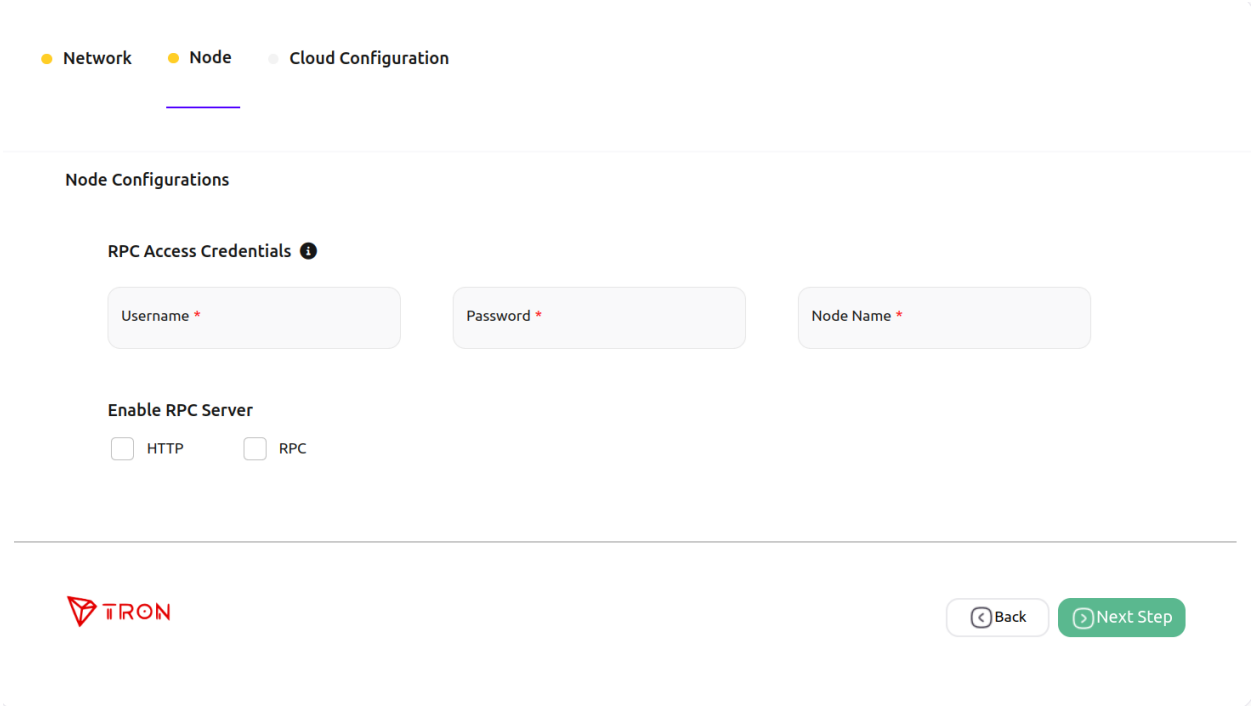


- **Name of Network**: In order to uniquely identify your network, this field requires a unique name for it. Unique over here is in terms of the account in which you are creating your network. In case you have created some network earlier, and now you are trying to create one with the same name, then the Zeeve platform won't allow you to create it.
- **Deployment Type**: Deployment type
- **Type Of Network**
  - **MainNet**: This will deploy your network on the network mainnet. This is suggested for deploying production-grade Tron dapps.
  - **TestNet**: This will deploy your network on the network testnet. you can use this for your non-production needs like testing or demonstrations.
- **Workspace**: This represents the workspace in which the network will be added after successful creation.

After providing all the details correctly, go to the next step by clicking on the **Next Step** button.

1. **Node Configuration**

In this section, you have to fill in the details of the nodes you want to add to your network.



- **Username & Password**: Choose the username and password of your choice. These will be used as RPC API credentials.
- **Node Name**: In order to identify your nodes, this field will be used.
- **RPC Server**: RPC, which stands for "Remote Procedure Call," is a group of protocols and interfaces that let us talk to the blockchain system. Through the RPC interface, we can ask for information about the blockchain (such as block number, blocks, node connection, etc.) and send a request for a transaction. > * **HTTP**: Uses individual HTTP requests and responses for each call, similar to a RESTful API. > * **ws**: WebSocket uses a persistent connection that allows the server to push data to the client.
- **JSON RPC APIs**: JSON RPC API is a bridge that allows dApps to connect to nodes.

After providing the details, click on **Next step** to go to the last step.

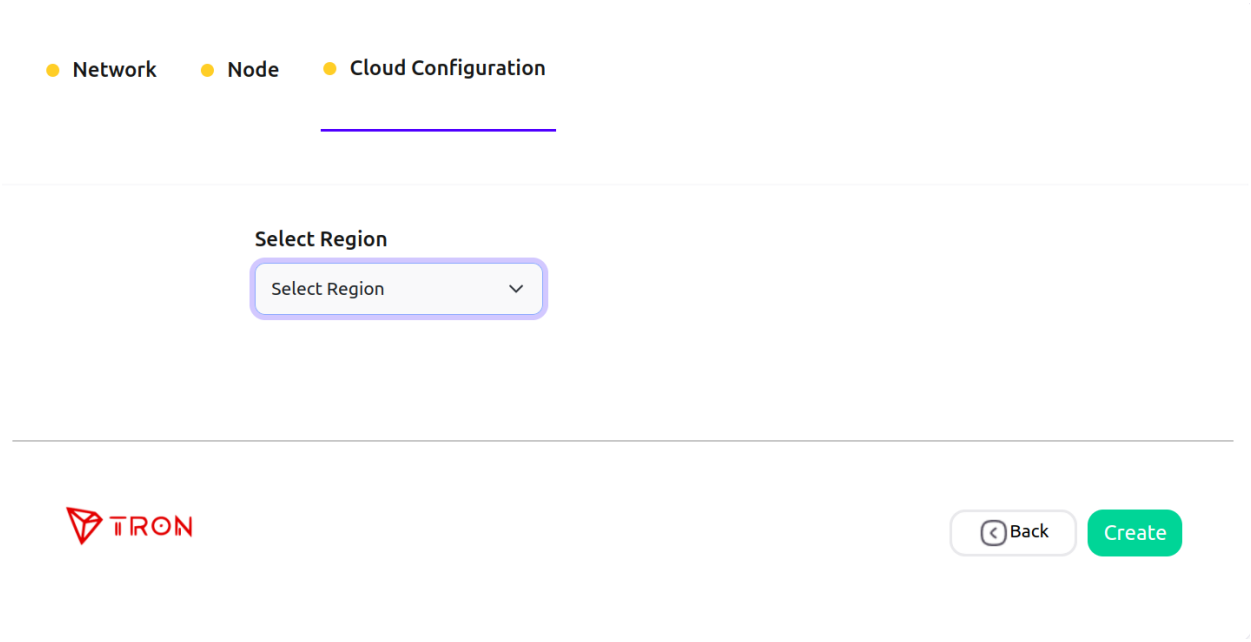
1. **Cloud Configuration**

This is the step for the configuration of the cloud for your nodes. This step can be different based on your selection of **Network configuration cards**

1. *Manged*
2. *BYOC - AWS*
3. *BYOC - GCP*
4. *BYOC - TENCENT*
5. *BYOC - DO*

Managed

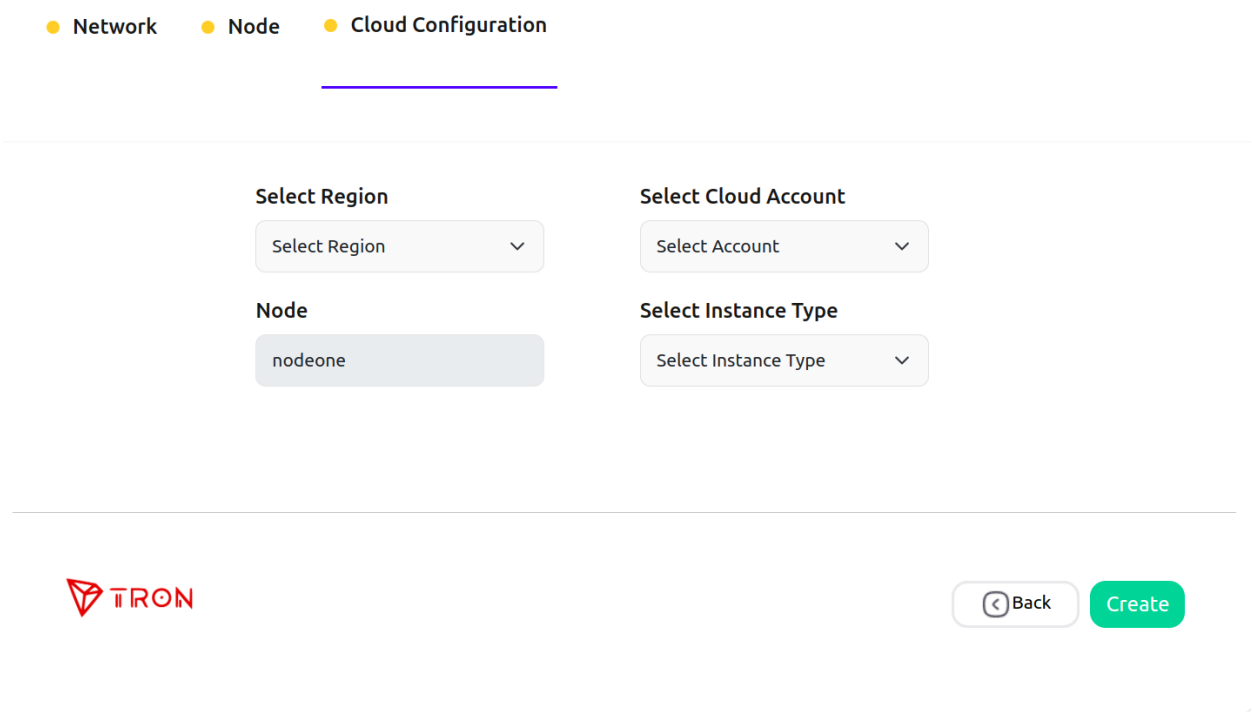
In the case of **Managed**, you don't have to bother about anything; just select the region for the network by clicking on **Select Region**.



- **Region**: In managed, you just need to select the region depending on your case and major area of operations to reduce the latency as much as possible. Ref.

BYOC - AWS

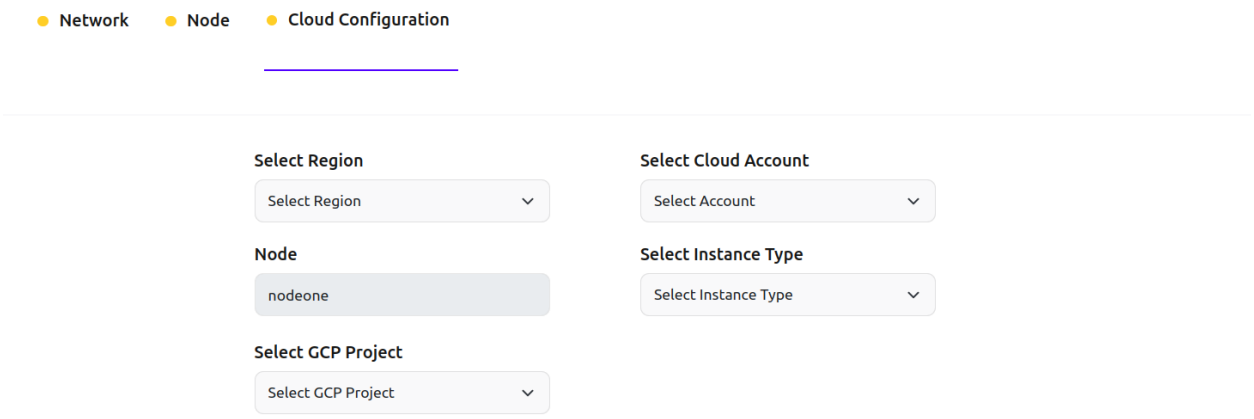
In the case of **BYOC - AWS**, select the region for the network by clicking on **Select Region**, select the *AWS* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: It indicates the region of cloud service. These regions are the geographic locations where your network instances are going to be hosted. In the case of AWS, each of its regions has multiple isolated locations known as Availability Zones. Amazon RDS provides you the ability to place resources, such as instances and data, in multiple locations. Resources aren't replicated across AWS Regions unless you do so specifically. Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

BYOC - GCP

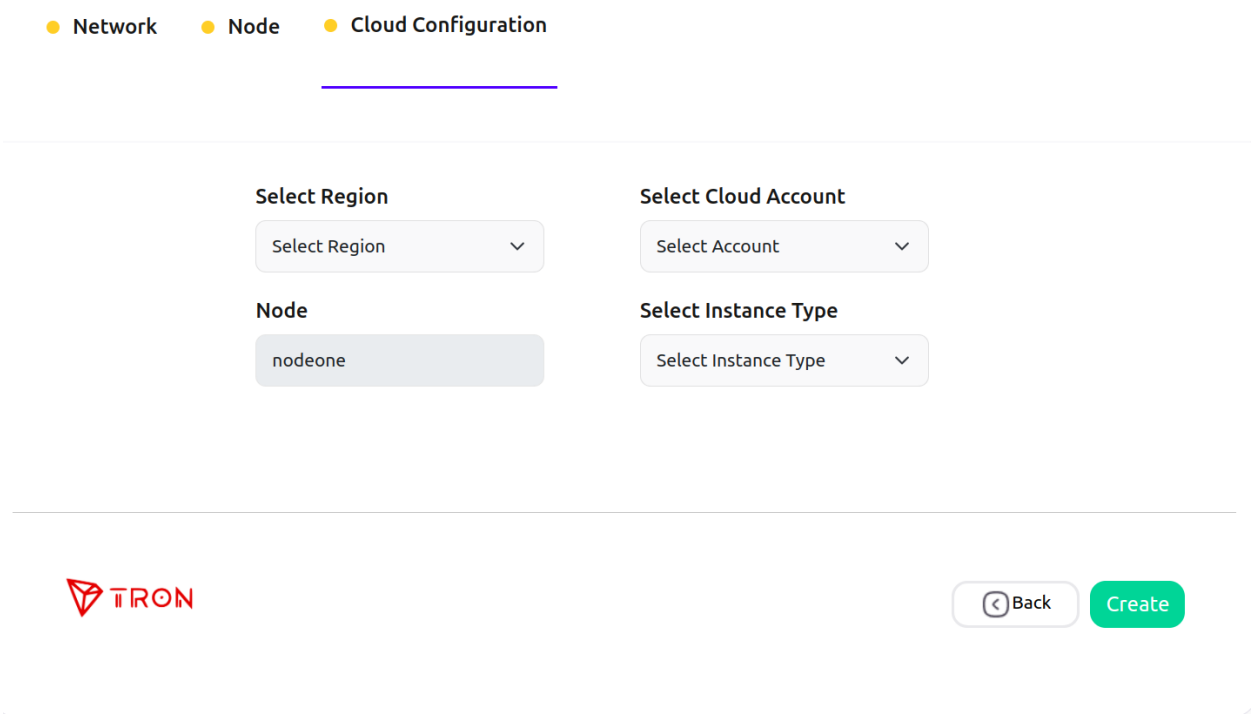
In the case of **BYOC - GCP**, select the region for the network by clicking on **Select Region**, select the *GCP* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: GCP Compute Engine resources are hosted in multiple locations worldwide. These locations are composed of regions and zones. A region is a specific geographical location where you can host your resources. Regions have three or more zones. For example, the us-west1 region denotes a region on the west coast of the United States that has three zones: us-west1-a, us-west1-b, and us-west1-c. Putting resources in different regions provides an even higher degree of failure independence. This lets you design robust systems with resources spread across different failure domains. Ref.
- **Cloud Account**: It represents the GCP cloud account that is going to be used for network creation.
- **Type of Instance**: Here, you'll get options for different virtual machines (VM). You can select the suitable VM type in Google Cloud Platform (GCP) depending on various factors, including your workload requirements, performance needs, and budget constraints. Ref.

BYOC - TENCENT

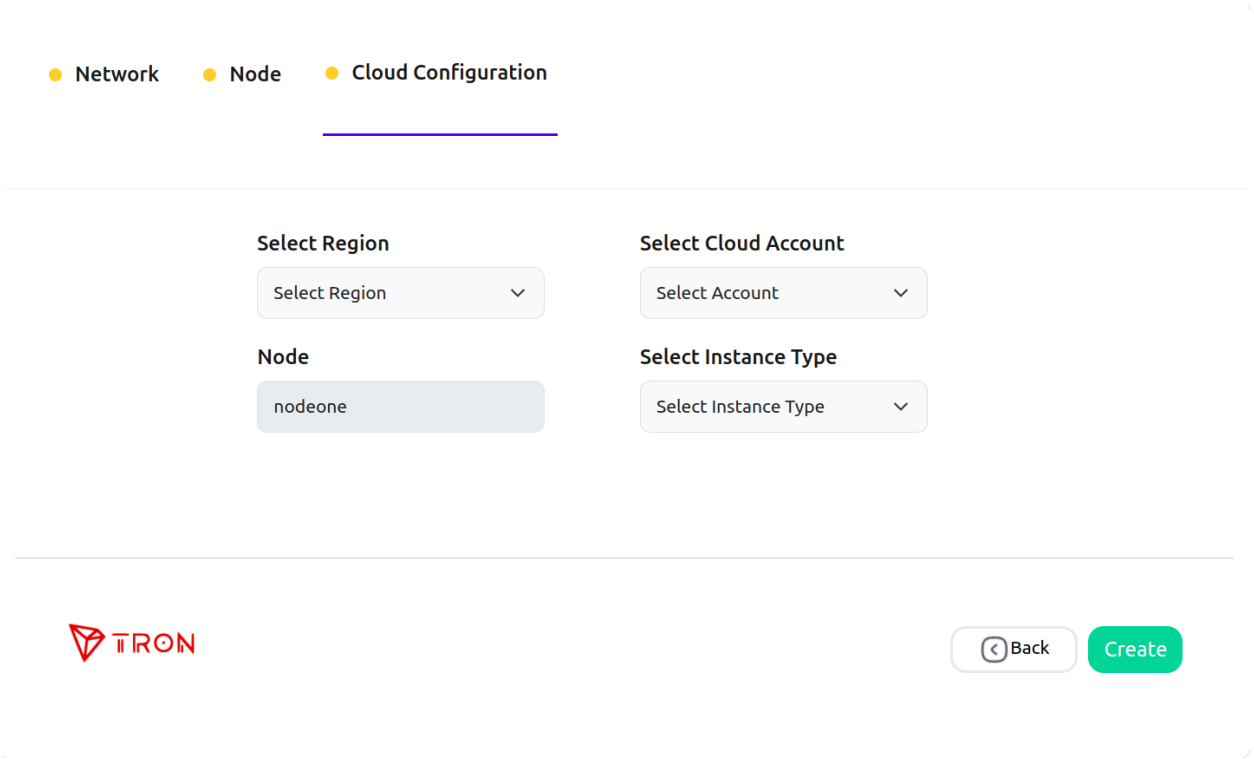
In the case of **BYOC - TENCENT**, select the region for the network by clicking on **Select Region**, select the *TENCENT* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: A region is the physical location of an IDC. In Tencent Cloud, regions are fully isolated from each other, ensuring cross-region stability and fault tolerance. We recommend that you choose the region closest to your end users to minimize access latency and improve access speed. Ref.
- **Cloud Account**: It represents the TENCENT cloud account that is going to be used for network creation.
- **Type of Instance**: Here, each instance type provides different computing, memory, and storage features. You can choose the instance type that suits your application scale. Tencent Cloud provides a range of instance families with varying combinations of CPU, memory, storage, heterogeneous hardware, and network bandwidth. This gives you flexibility in selecting the appropriate mix of resources for your applications. Ref.

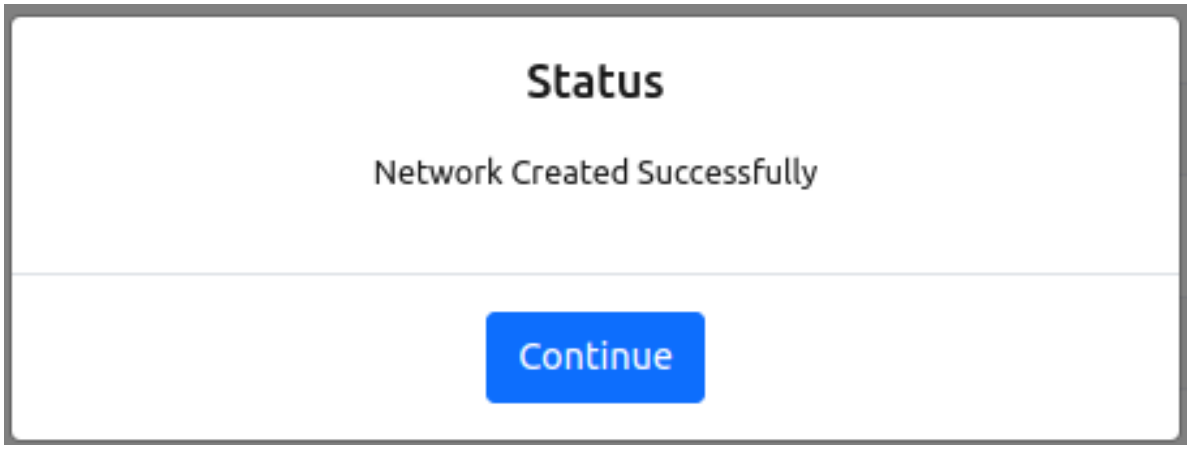
BYOC - DO

In the case of **BYOC - DO**, select the region for the network by clicking on **Select Region**, select the *Digital Ocean* account you want to use by clicking on **Select Cloud Account**, also choose the instance type as your requirement by clicking on **Select Instance Type**.



- **Region**: A region is a physical location in the world where DigitalOcean has a data center. You can always see the available regions to choose from when creating a new app, Ref.
- **Cloud Account**: It represents the AWS cloud account that is going to be used for network creation.
- **Type of Instance**: It defines the combination of CPU cores and memory. Choose the configuration which could handle loads of your network. This parameter is useful for scaling up the network. The type of Instances may vary from cloud to cloud.

1. Click on the **Create** button. A pop-up window will appear similar to the below image, which ensures the successful creation of your network.

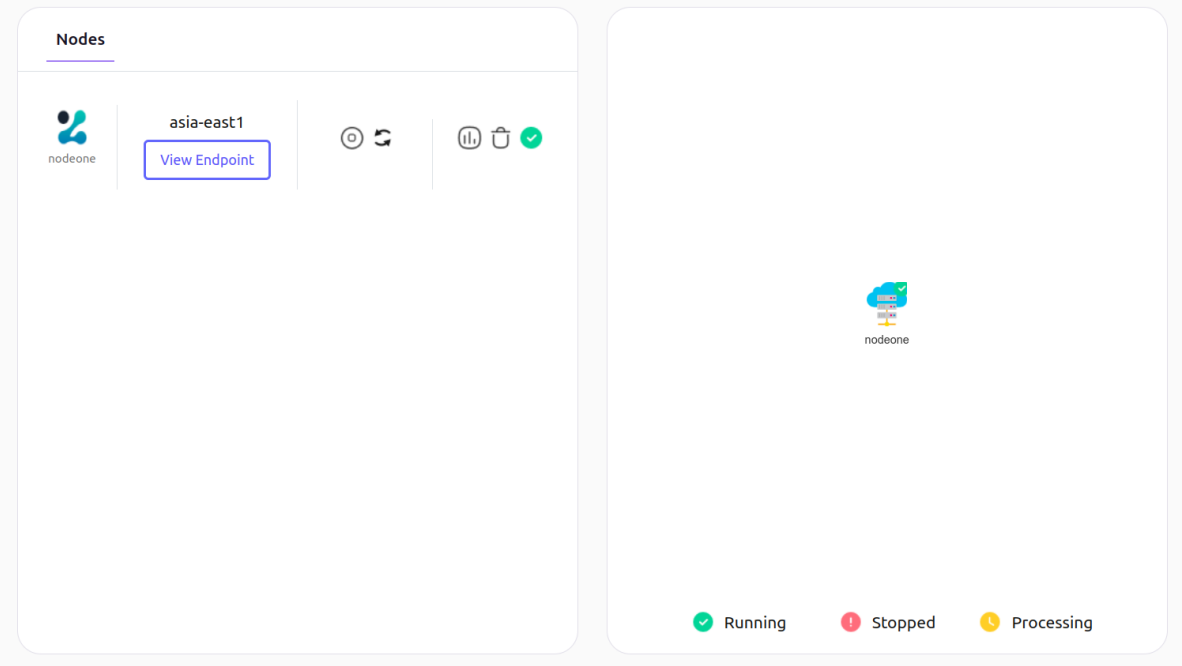


2. Click on **continue**, and you will be redirected to a page similar to the below image, where you can see the nodes listed you've just added to the network.

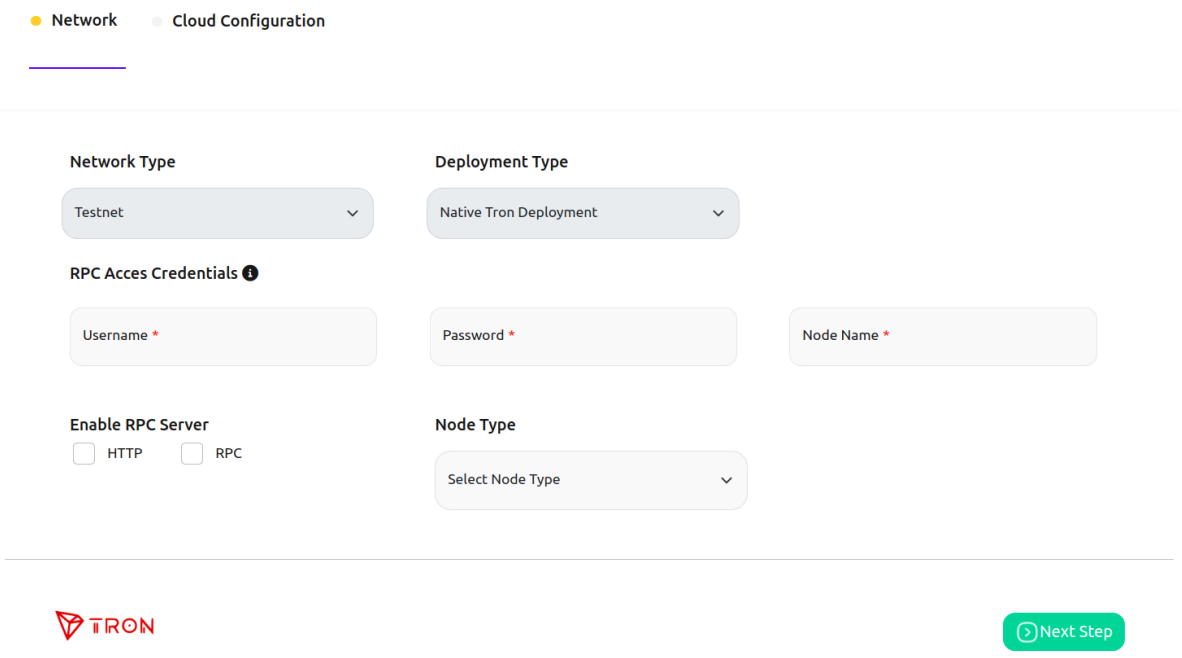
26.1.2 Add node to a network

This section will guide you on how you can add a node to a network.

1. Select the network to which you want to add a node, and click on the network card Ref.. You will get to see similar to the below image.

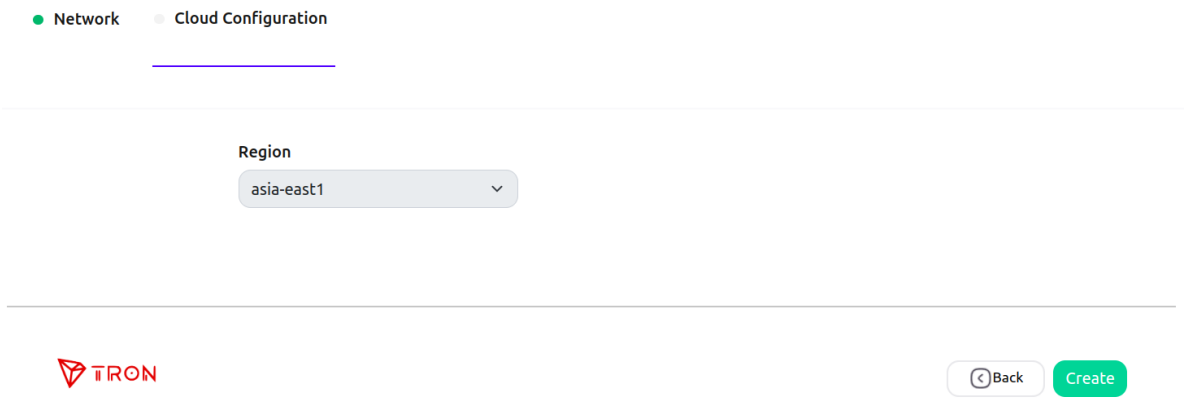


2. Click on the **Actions** button on the top right, and select the **Add Node** option. You will get to see a web page similar to that provided below.



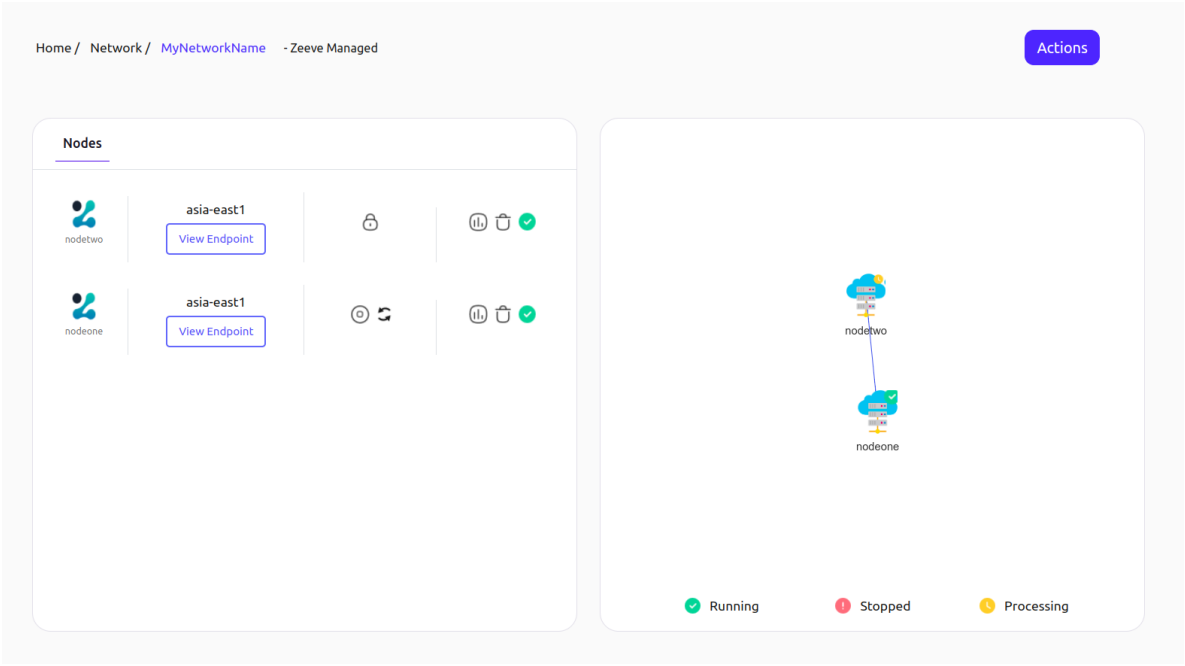
3. Fill in the details for the new node and click on the **Next** button. In this step, the cloud configuration and region

will be prefilled according to the configuration of the network. Click on the **Create** button, and it's done!

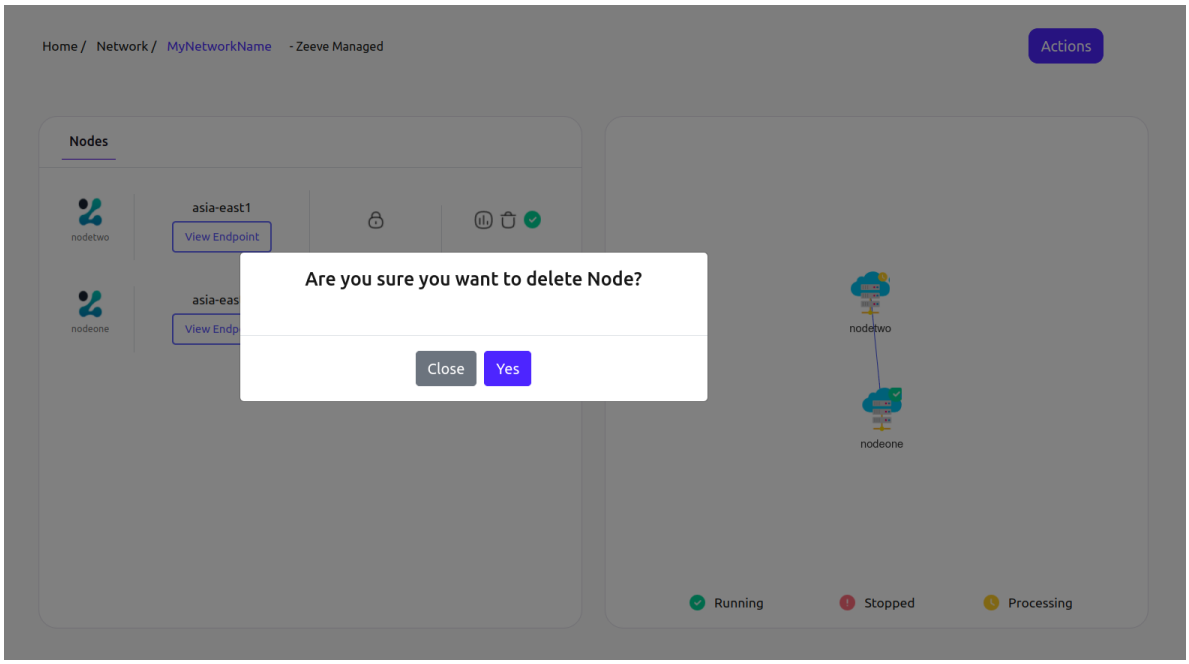


26.1.3 Delete node in a network

1. Select the network you want to perform the deleted node to, and click on the network card Ref.. You will get to see similar to the below image.

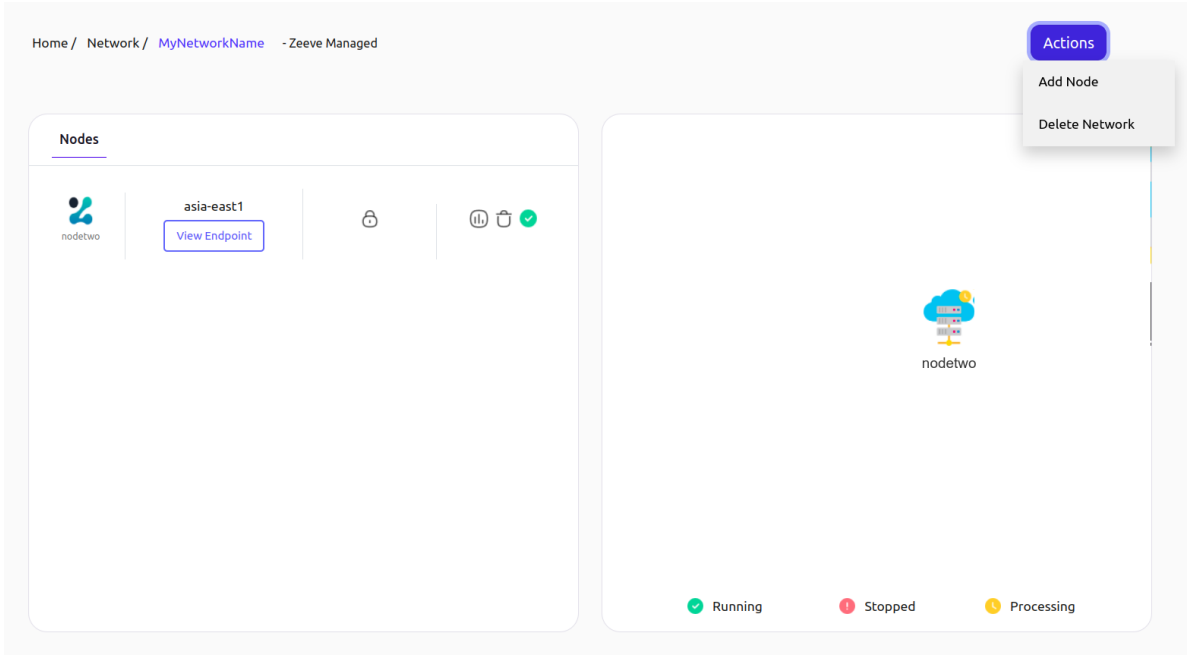


2. Click on the delete icon present alongside the node. A pop-up window will open for the confirmation; click on the **yes** button to confirm.

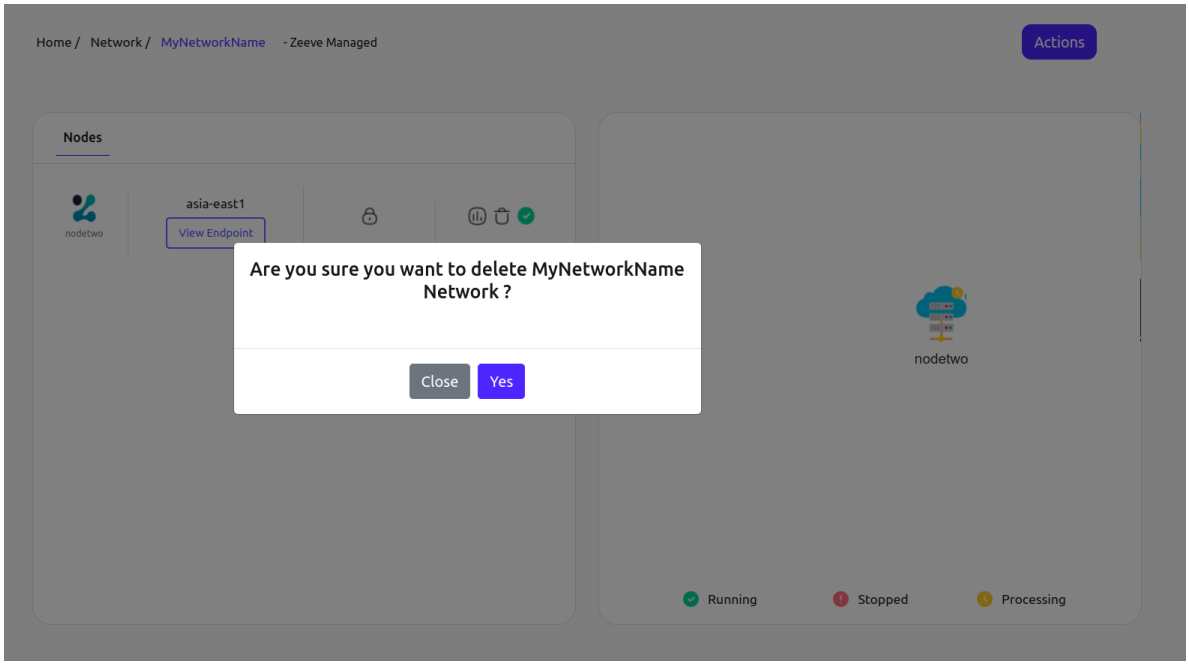


26.1.4 Delete a network

1. Select the network you want to delete, and click on the network cardRef.. You will get to see similar to the below image.



2. Click on the **Actions** button on the top right, and select the **Delete Network** option. A confirmation window will open; click on the **Yes** button attached to it.



NOTE It can take a few minutes to delete a network.

CHAPTER

TWENTYSEVEN

DESCRIPTION: BUILD DECENTRALIZED APPLICATIONS USING CREDITS AND ZEEVE'S PLATFORM. OUR DOCUMENTATION PROVIDES TECHNICAL DETAILS ON AUTHENTICATION, MAKING REQUESTS AND HANDLING RESPONSES FOR INTERACTING WITH CREDITS AND THE ZEEVE PLATFORM.

CHAPTER

TWENTYEIGHT

CREDITS DEPLOYMENT SPECIFICATIONS

Currently Credits Networks can be created using some default settings only.

Currently the Zeeve team is working to roll out more features to give you more power in your hands. Keep a watch at the releases or join our active developer community on telegram at https://t.me/ZeevePlatform.

CHAPTER

TWENTYNINE

DESCRIPTION: LEARN HOW TO CONFIGURE YOUR PRODUCTS ON ZEEVE'S PLATFORM. OUR DOCUMENTATION PROVIDES DETAILED INSTRUCTIONS AND RESOURCES FOR SETTING UP AND CUSTOMIZING YOUR PRODUCTS TO MEET YOUR SPECIFIC NEEDS

CHAPTER

THIRTY

PRODUCT CONFIGURATIONS

Zeeve allows the user to configure their projects onto the platform, which comes with in-built CI-CD pipelines and let Zeeve do the heavy lifting to make a network compatible for your product. This is ensured with some practices developer shall do in order to make the deployments one click based(mentioned later). There are some benefits to configure your product through Zeeve:

- Allow automated deployments without having to configure any web servers or domain endpoints.
- Form CI-CD pipelines. Authorizing your git account and configuring you git repository to a product.
- List your product onto marketplace to reach customers. You can either keep your products private or you can list them or marketplace as fully configured one click deployment products or as just an idea.

Following section defines how to set-up your product on the platform:

30.1 Configuring a Product

Configuring a product allows you to do one click deployments for your product in any cloud. It is based on configuring your git repo to a product and making sure certain configuration files are utilized in a standard manner such that Zeeve can automate it to 100%.

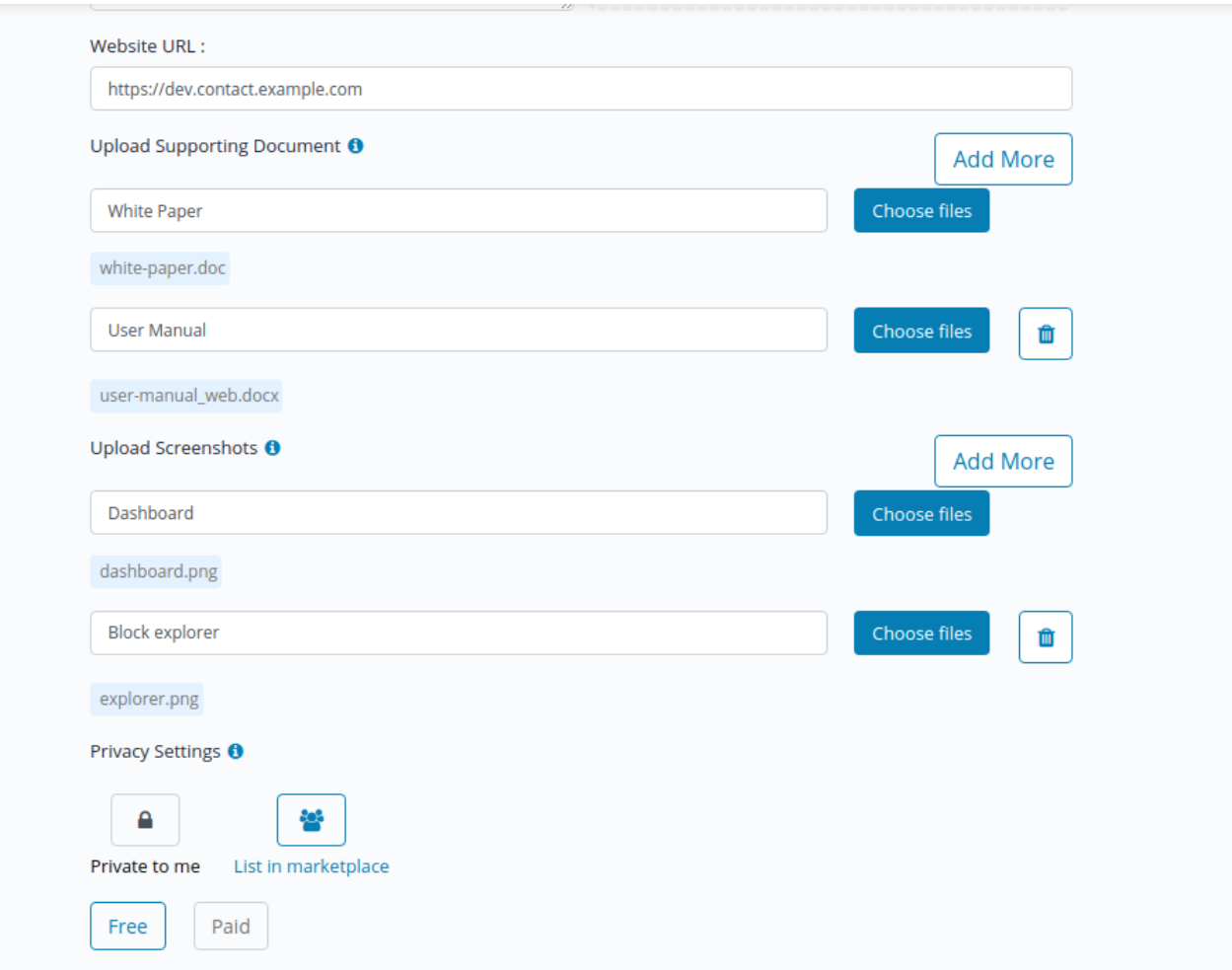
To configure a product fully as click based deployment make sure you have seen the development practices for the relevant blockchain protocol.

- *For Hyperledger Fabric*
- For Hyperledger Sawtooth

30.1.1 Steps:-

- Go to `Marketplace -> My products -> Add product.`
- Provide title and Add team members to your product.
- Select the blockchain protocol your product is built upon.
- Add a small description and logo (if any) to your product.

- Provide supporting documents(could be presentation, white paper or architecture document) for the people to understand the idea behind the product.



You can choose to keep a product private to you once you have configured it, or you can choose to list it in marketplace.

- Select the type:- Product or Idea.

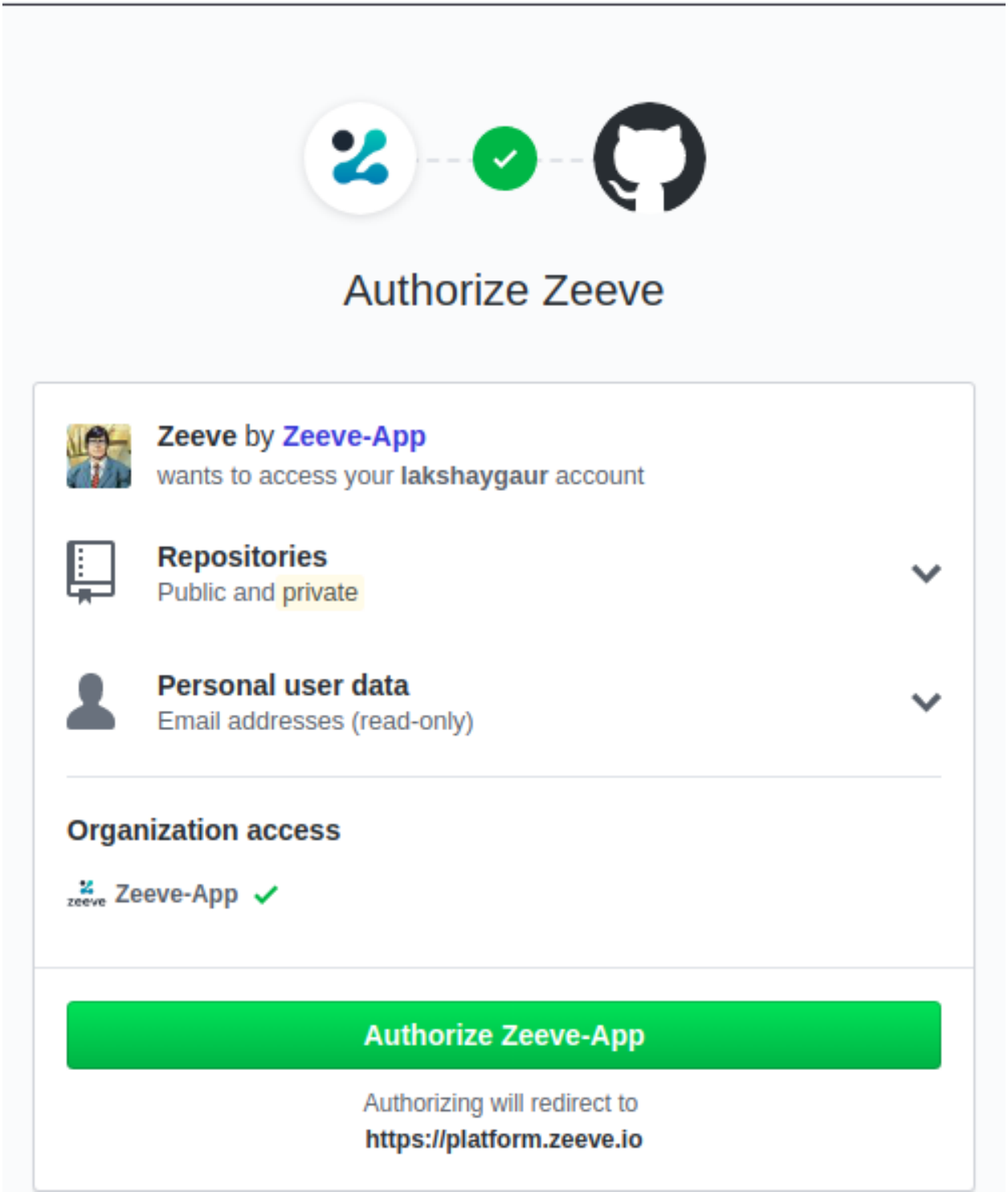
Publishing as an idea A product can be listed as an idea or as full deployable solution. Listing it as an idea allows the user to present it to marketplace before even the product is ready or has been implemented. Listing it as idea doesn't require any git authorizations or development practices to worry about.

To publish as an idea, simply select option `Publish my idea`.

Configuring as a Product A product can be configured such that it is available to the public as a free or as a paid product. Once authrized the party can themselves deploy the product anytime they want. To configure as a product you would need to

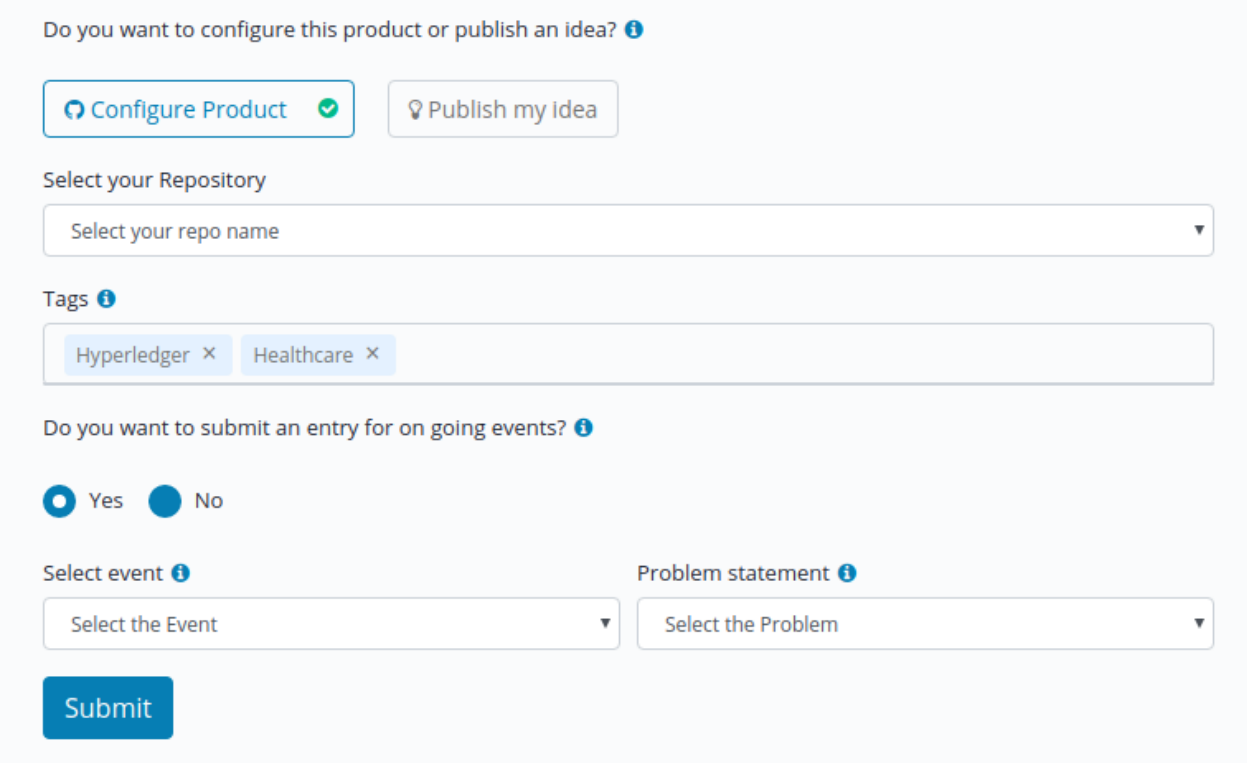
Zeeve will make use of the default git branch for setting up the CI-CD piepline for the project.

To configure as a product, select configure product option. It will ask for github authorization and ask you to select one of the projects. Make sure you have followed *development practices* if you want to make your product deployable in matter of clicks.



Once configured, you can deploy a network and choose to deploy this product on it.

- If you want to enter any running events on platform, you can select yes and select event type and problem statements.


Do you want to configure this product or publish an idea?
Configure Product
Publish my idea
Select your Repository
Select your repo name
Tags
Hyperledger
Healthcare
Do you want to submit an entry for on going events?
Yes
No
Select event
Select the Event
Problem statement
Select the Problem
Submit

CHAPTER

THIRTYONE

DESCRIPTION: EXPLORE BEST PRACTICES FOR DEVELOPING ON ZEEVE'S PLATFORM. OUR DOCUMENTATION PROVIDES GUIDELINES AND RECOMMENDATIONS FOR BUILDING SCALABLE, RELIABLE APPLICATIONS AND MAINTAINING HIGH LEVELS OF QUALITY AND SECURITY.

CHAPTER

THIRTYTWO

DEVELOPMENT PRACTICES

Every protocol has its own way of development and deployment which might vary a lot from other protocol's development practices. This make automation for your apps even tougher. Zeeve intends to allow developer to do their dvelopment as they do, and still allow Zeeve's automation to do the rest. This section speaks of how to develop you applications and make the deploy-ready for Zeeve for which developer shall keep certain practices in mind.

- *Hyperledger Fabric's application development practices*
- Hyperledger Sawtooth application development practices
- Others coming soon

CHAPTER

THIRTYTHREE

HYPERLEDGER FABRIC'S APPLICATION DEVELOPMENT PRACTICES

Refer to Sample Project

Certain practices can help the developer create applications which are one click deployable on Zeeve. They are:-

- For packaging your product to be automated by Zeeve, you will need to dockerize your project's services by creating `Dockerfiles` for them.

```
FROM node:8.9.0

WORKDIR /balance-transfer

COPY . .
ENV PORT=4000
RUN npm install
CMD node app
```

- Make use of `network-config.yaml` (connections profiles) and `org.yaml` (if required) for all blockchain related configurations. At the time of deployment for a network, Zeeve creates these files and allows the developer to download them along with the other artifacts. You can consider this file to develop your applications. All other application configs should be part of the Docker image itself.
- Create a `.env` file containing an array of domain prefixes corrosponding to Ingress resource definitions. The syntax should be of the form: `("<domain_prefix_1>:<ingress_resource_name_1>" "<domain_prefix_2>:<ingress_resource_name_2> ...")`

```
EXT_EXPOSED_SERVICES=("balancetranfer:balance-transfer-ingress")
```

- Create a `docker-compose-build.yaml` file for creating images of all services that your application requires. This will help Zeeve create relevant images, push them to the container-registry and later use them.
  - Each service definition whose container image would be created needs to have an `image` keyword, the associated value needs to same as that of image name (Deployment.contianers.spec.containers.image) in k8_application.yaml.template.

```
version: "2.0"

services:
  balance-transfer:
    build:
      context: .
      dockerfile: Dockerfile
    image: balance-transfer:latest
    container_name: balance-transfer-default
```

(continues on next page)

(continued from previous page)

```
    ports:
      - '4000:4000'
    command: |
      bash -c "PORT=4000 node app"
    restart: always
```

- Create a yaml file `k8_application.yaml.template` and keep it at your project's root folder. The file needs to keep the following points in account:-
  - The image name for containers needs to adhere to the guideline outlined in the step above.
  - Define an `imagePullSecrets` named container-registry-cred. Creation and updation of this secret is handled by Zeeve, but the definition is developers responsibility.
  - For mounting relevant crypto data and channel artifacts in a deployment, Zeeve will create secrets and mount them on `/crypto-data` path. The deployment/s on which this mounting takes place is identified by special character string `@@replace_my_crypto_artifacts@@`.
  - Host for each Ingress resource will be an amalgamation of information specified in .env file and domain assigned to your Kubernetes cluster.
  - Take special care to mount relevant persistent volumes as pods will be recreated whenever there is an application update.

```
apiVersion: v1
kind: Service
metadata:
  name: balance-transfer-svc
spec:
  type: ClusterIP
  ports:
    - port: 4000
      targetPort: 4000
      protocol: TCP
  selector:
    name: balance-transfer-dep

 ---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: balance-transfer-dep
spec:
  replicas: 1
  selector:
    matchLabels:
      name: balance-transfer
  template:
    metadata:
      labels:
        name: balance-transfer
    spec:
      volumes:
        #- name: balance-transfer-data
        #  persistentVolumeClaim:
        #    claimName: balance-transfer-data-pvc
            @@replace_my_crypto_artifacts@@
      imagePullSecrets:
```

(continues on next page)

(continued from previous page)

```
      - name: container-registry-cred
      containers:
        - name: balance-transfer
          image: balance-transfer:12
          imagePullPolicy: Always
          ports:
            - containerPort: 4000
              protocol: TCP
          #livenessProbe:
          #  httpGet:
          #    path: /
          #    port: 4000
          command:
            - bash
            - -c
            - |
              echo "installing GO"
              cd /usr/local
              curl -O https://dl.google.com/go/go1.10.3.linux-amd64.tar.gz
              tar -xvf go1.10.3.linux-amd64.tar.gz
              echo "export PATH=$PATH:/usr/local/go/bin" >> /root/.bashrc
              source /root/.bashrc

              echo "configuring application"
              mkdir /application
              cd /application
              cp -r /balance-transfer/* ./
              cp -Lr /crypto-data/* ./artifacts/
              #npm install
              node app
          volumeMounts:
            #- mountPath: /application
            #  name: balance-transfer-data
              @@replace_my_crypto_artifacts@@
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: balance-transfer-data-pvc
spec:
  accessModes:
    - "ReadWriteOnce"
  resources:
    requests:
      storage: "5Gi"

---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: balance-transfer-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/backend-protocol: "http"
spec:
  rules:
    - host:
```

(continues on next page)

(continued from previous page)

```
    http:
      paths:
      - backend:
          serviceName: balance-transfer-svc
          servicePort: 4000
```

- While uploading the project attach some supporting documents to explain the organisation names and other details that shall be put while creating networks for the product. These documents will help users to create and deploy networks and products on their own.

---

description: Learn about the different blockchain protocols available on Zeeve's platform. Our documentation provides an overview of the key features, benefits and use cases for Hyperledger Fabric, Ethereum, Corda, and other leading blockchain technologies.

meta:

- name: robots content: noindex

CHAPTER

THIRTYFOUR

MAJOR BLOCKCHAIN PROTOCOLS

This page of this documentation is here just to provide you with a gist of various major protocols and help you give a rough idea around them. So you can choose the one which suits your usecase well.

34.1 HyperLedger Fabric

Hyperledger Fabric is an open source enterprise-grade permissioned distributed ledger technology (DLT) platform, designed for use in enterprise contexts, that delivers some key differentiating capabilities over other popular distributed ledger or blockchain platforms. Ref.

34.2 Ethereum

Ethereum is an open-source, blockchain based distributed computing platform which involves smart contract (business logic coded script) functionality. Ether is a currency used for transactions between accounts and also used to compensate participating mining nodes for computations performed. Ref.

34.3 Corda

Corda is an open source blockchain project, designed for businesses. Only Corda allows you to build interoperable blockchain networks that transact in strict privacy. Corda's smart contract technology allows businesses to transact directly, with value. Ref.

34.4 Avalanche

 Avalanche is an open, programmable smart contracts platform with low cost and Solidity compatible dApps. It is the fastest platform as measured by time-to-finality.

34.5 Axia

 Axia is an open, programmable smart contracts platform with low cost and Solidity compatible dApps. It is the fastest platform as measured by time-to-finality.

34.6 Binance

Binance is an EVM-compatible customized blockchain protocol forked out of Geth and uses consensus of Proof of Staked Authority (PoSA).

34.7 Polygon

 Polygon is a protocol and a framework for building and connecting Ethereum-compatible blockchain networks.

34.8 Fantom

 Fantom is a fast, scalable, and secure layer-1 platform built on an aBFT consensus protocol.

34.9 Tron

Tron is a protocol and a framework for building and connecting Tron-compatible blockchain networks.

CHAPTER

THIRTYFIVE

DESCRIPTION: FIND RESOURCES AND REFERENCES FOR USING ZEEVE'S PLATFORM. OUR DOCUMENTATION INCLUDES TECHNICAL GUIDES, API DOCUMENTATION, AND OTHER USEFUL MATERIALS FOR DEVELOPERS BUILDING APPLICATIONS ON THE ZEEVE PLATFORM.

CHAPTER

THIRTYSIX

REFERENCES

In order to understand various terminologies used in the knowledge base and different releases of Zeeve, they have been provided under below sections.

- *Glossary*
- *Release*

---

description: Discover Zeeve's distributed file system for storing and managing large amounts of data. Our documentation provides an introduction to the key features and benefits of the Zeeve distributed file system, as well as technical details on how to use it with our platform.

meta:

- name: robots content: noindex

CHAPTER

THIRTYSEVEN

ZEEVE DISTRIBUTED FILE SYSTEM

37.1 Overview

ZDFS is your secure and decentralised storage for your digital assets whether it be NFTs or other digital assets. It has been built by developers for developers and feature the most secure, easy to use and easy to integrate decentralised storage service. Now you don't need to hustle around IPFS complexities because ZDFS offers you the complete control over your contents using interactive *ZDFS GUI*. ZDFS allows you to upload your file, folder and content using an IPFS hash of 25GB in size. ZDFS ensures your content is always online in the IPFS network without worrying about unusual downtime and IPFS node management. ZDFS supports *IPFS CLI* out of the box and also provides Public *HTTP APIs* to integrate with your application and perform various actions on your content.

There is a series of steps you need to follow in order to access ZDFS service.

1. *Purchase Subscription*.
2. *Create Access*.
3. *Usage*.

37.2 Purchase Subscription

ZDFS offers you different subscription plans and you can optimize cost as per your requirement using the plan calculator. You need to purchase the endpoints of a particular plan to enjoy seamless services of ZDFS. You can purchase 'N' number of endpoints of a particular plan on a monthly basis.

NOTE: *We are referring* ***1 ENDPOINT*** *as* ***1 Access Key****.*

There are two plan types offered by ZDFS currently:

- *Basic Plan*.
- *Standard Plan*.

37.2.1 Basic Plan

This plan is meant for experimenting purpose where you can enjoy the following features:

- 1 GB Storage.
- 5 GB Bandwidth.
- API Based Access.
- Console Management.
- Pinning Service.
- Community Support.

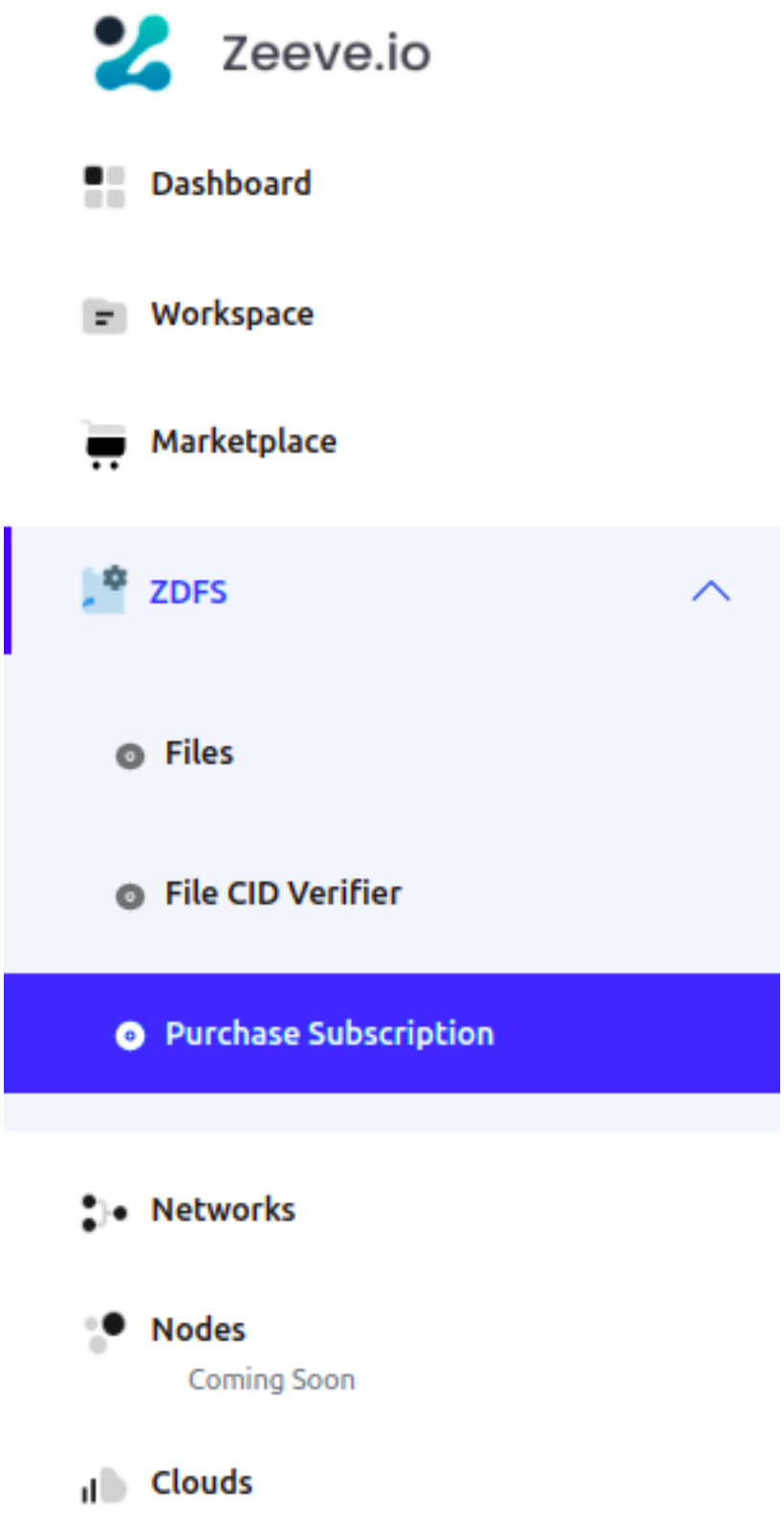
37.2.2 Standard Plan

This plan is meant for small-scale purpose where you can enjoy the following features:

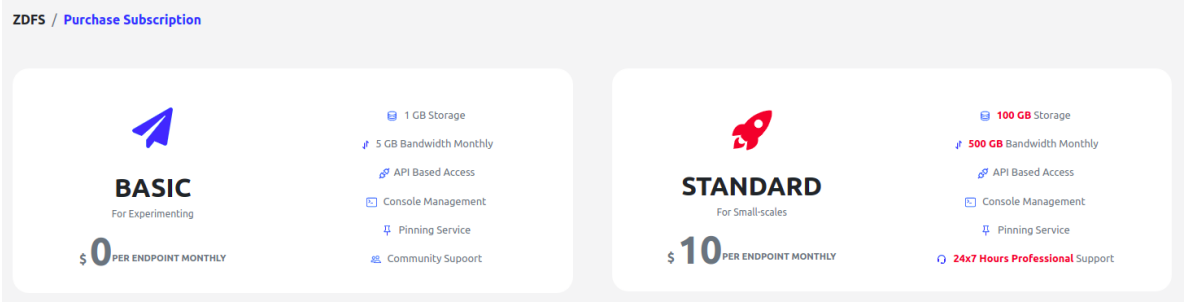
- 100 GB Storage.
- 500 GB Bandwidth.
- API Based Access.
- Console Management.
- Pinning Service.
- 24x7 Hours Professional Support.

INFO: *ZDFS also offers* ***ENTERPRISE PLAN****, you can contact us at* ***support@zeeve.io*** *to get the quotation.*

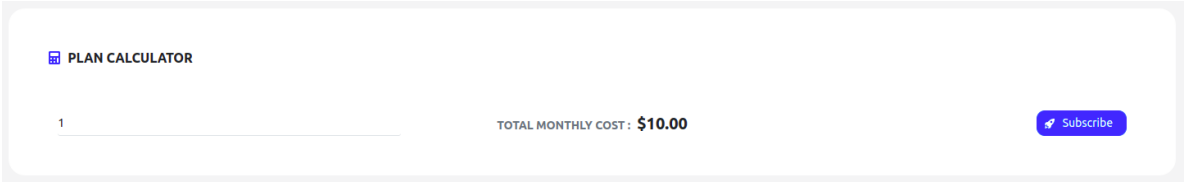
- Go to `ZDFS -> Purchase Subscription`.



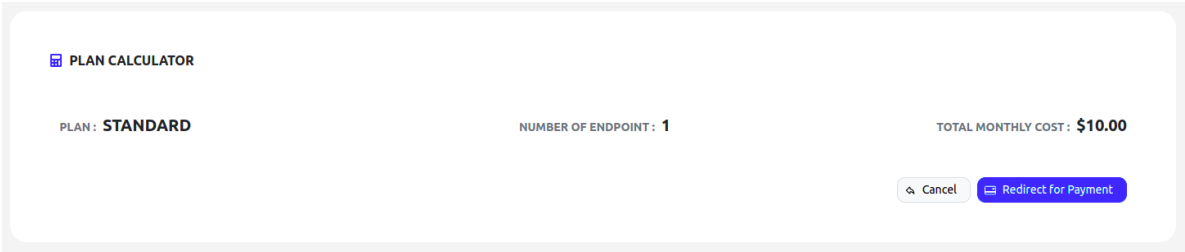
- Select the plan by clicking on it.



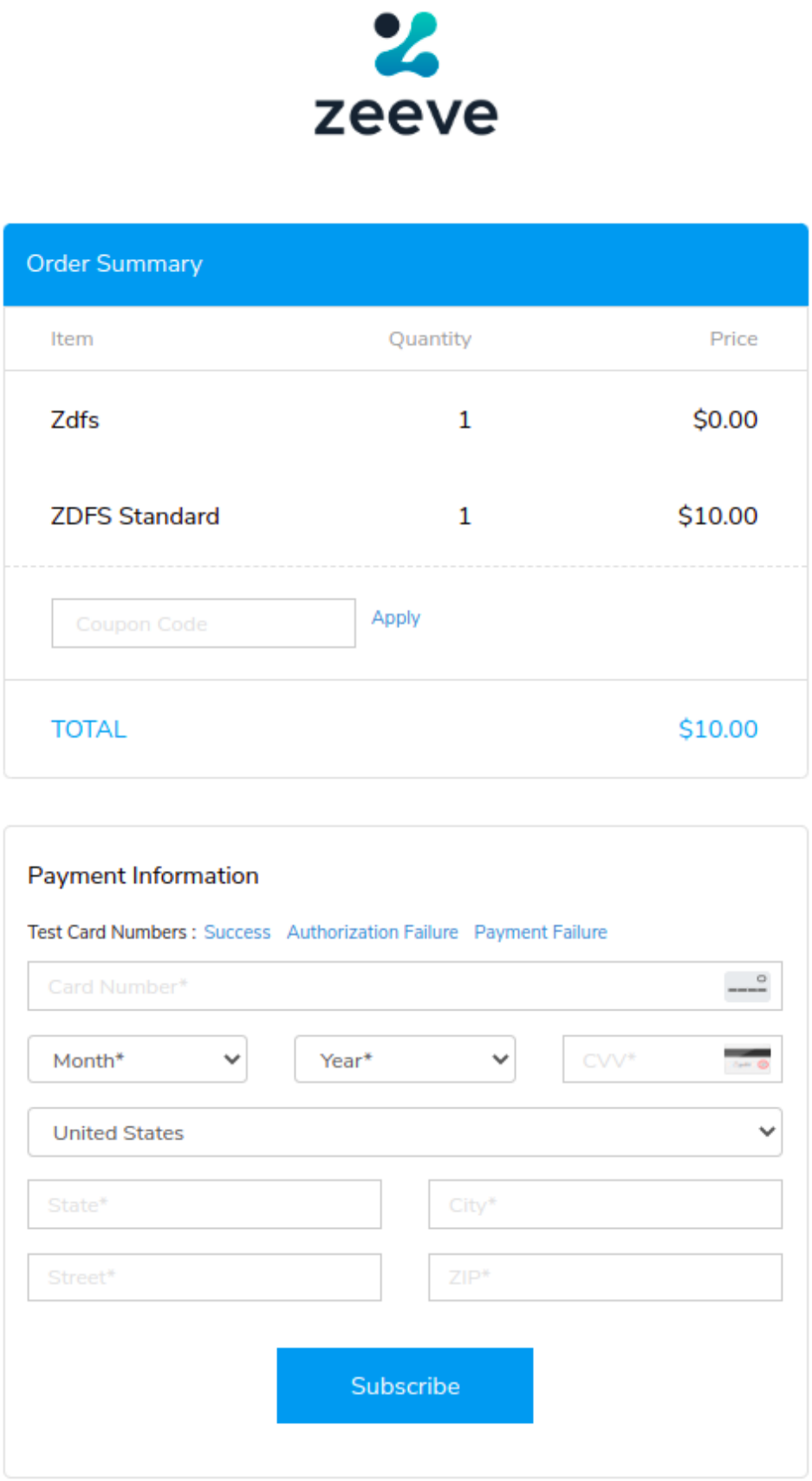
- You can enter the 'N' number of endpoints you want to purchase.
- Click on the `Subscribe` button to finally crosscheck your purchase details.



- Click on the `Redirect for Payment` button to get redirected to the payment gateway.

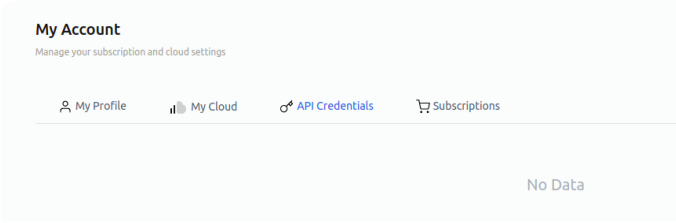


- You can make the payment by entering your card details and applying Coupon Code if you have any.

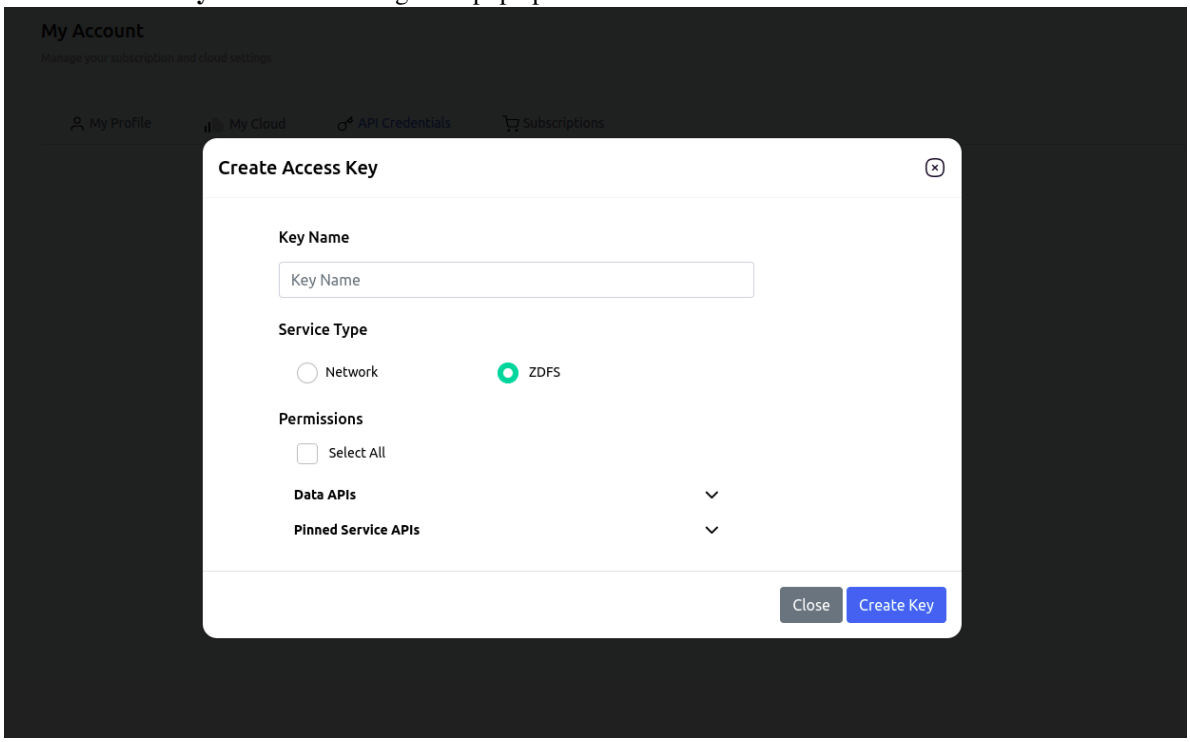

zeeve
Order Summary
Item
Quantity
Price
Zdfs
1
$0.00
ZDFS Standard
1
$10.00
Coupon Code
Apply
TOTAL
$10.00
Payment Information
Test Card Numbers : Success Authorization Failure Payment Failure
Card Number*
Month*
Year*
CVV*
United States
State*
City*
Street*
ZIP*
Subscribe

37.3 Create Access

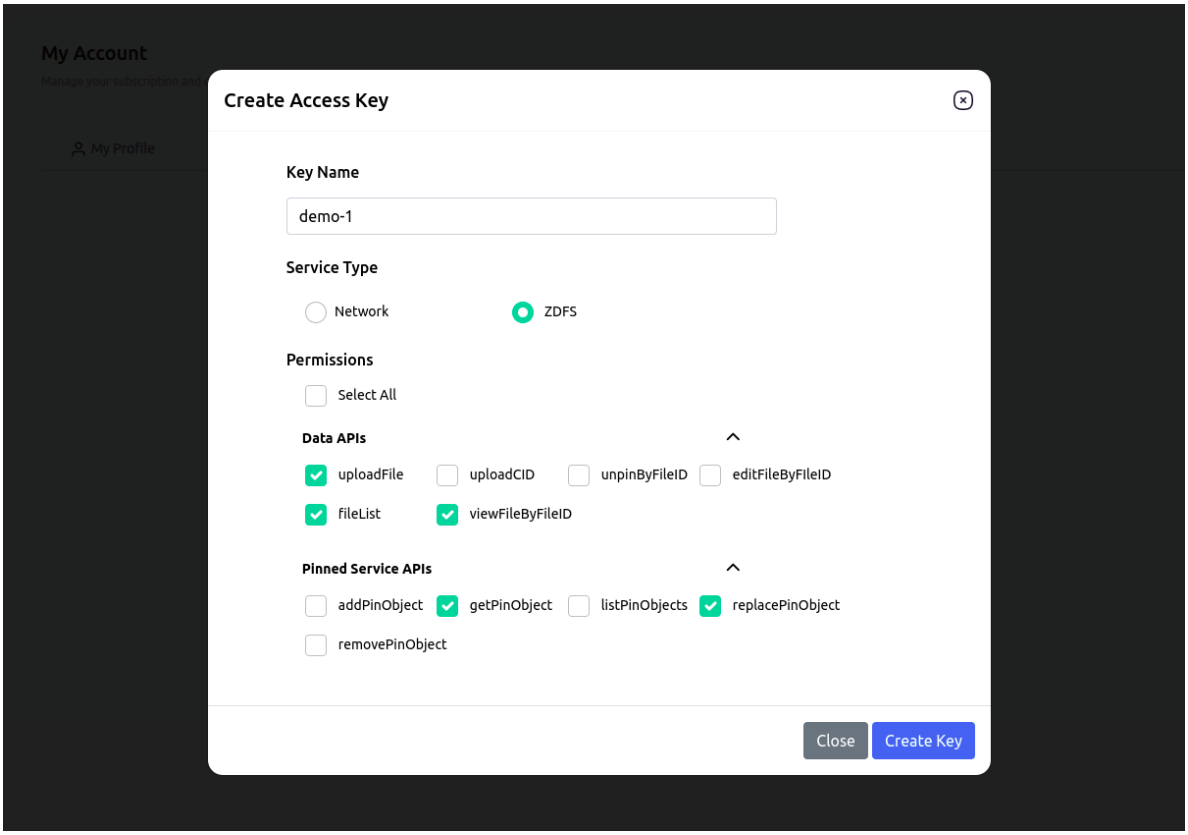
You need to generate an access token in order to use ZDFS APIs. You need to have `AccessKey` & `AccessSecret` or `AccessToken` for the same.



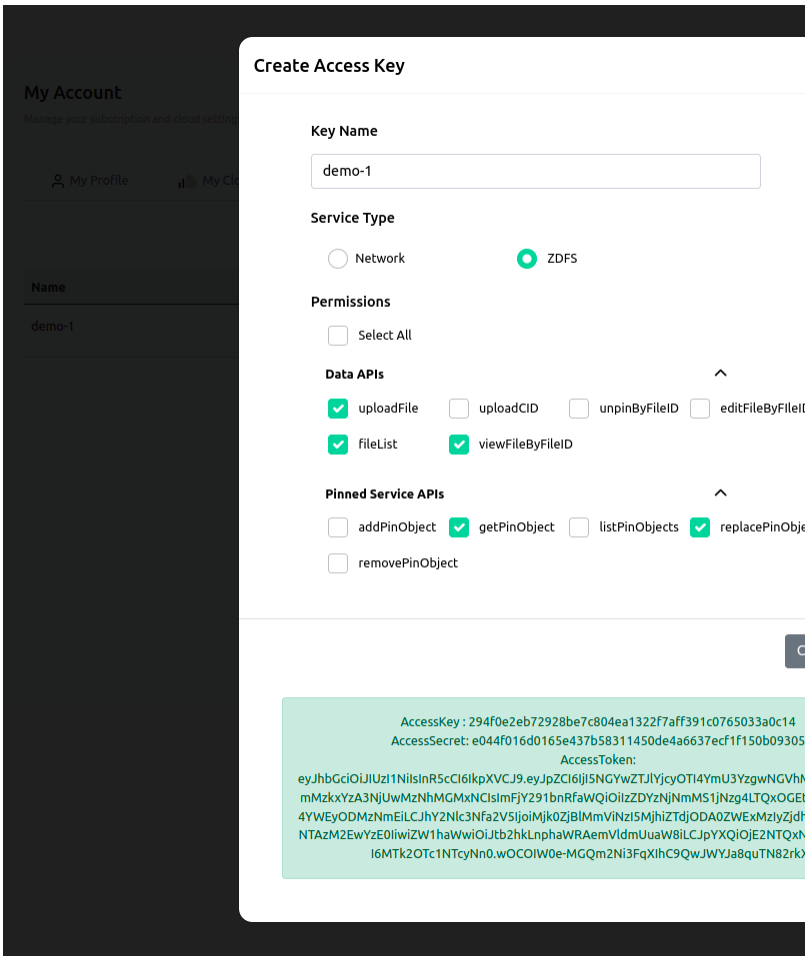
- Go to `Settings/My Accounts -> API Credentials`.
- Click on the **Create Key** button. You will get the pop up form to take some information as described in the next step.



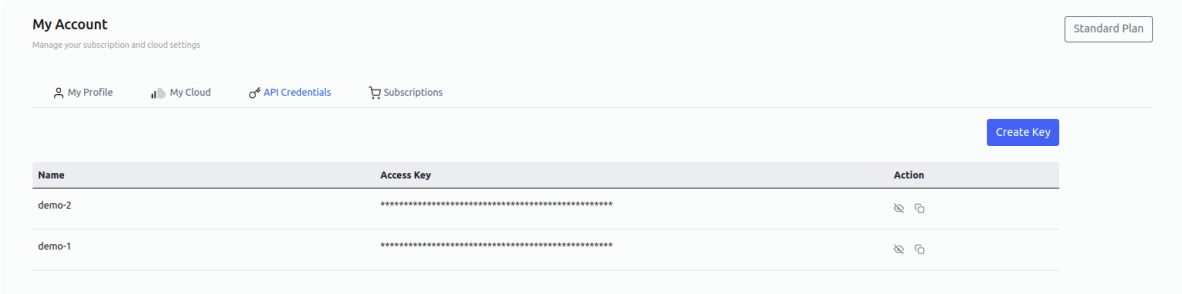
- Provide **Key Name** as per your choice.
- Select `ZDFS` as **Service Type**.
- Select **Permissions** as per your requirement. Actions are having direct relation with API methods present in the



API Document.



- Click on the **Create Key** button in order to generate it.
- Please write down `AccessKey`, `AccessSecret` and `AccessToken` at a safe place.
- Close the pop by clicking on the Cancel button.
- You can see all your generated access keys within the same section.



37.4 Usage

You can utilize ZDFS services in ample ways as per your use case. There are 3 ways in which you can use ZDFS as follows:

1. *IPFS CLI*.
2. *HTTP APIs*.
3. *ZDFS GUI*.

37.4.1 IPFS CLI

Command-line users benefit from `ipfs pin remote` commands, which simplify remote pinning operations. The built-in pinning service API client executes all the necessary remote calls under the hood.

Configure ZDFS in IPFS CLI

You need to have an *access token* with **Pinned Service APIs** permissions.

Open a terminal and execute the following command:

```
ipfs pin remote service add zdfs https://app.zeeve.io/zdfs-api/api/psa ACCESS_TOKEN
```

For more information please refer to API documentation.

37.4.2 HTTP APIs

ZDFS offers you public APIs to integrate in your application. An *access token* is required to be sent with each request in the HTTP authorization header. Learn More

There are two types of HTTP APIs:

1. *Pinning Service API (PSA)*.
2. *Data APIs*.

1. Pinning Service API (PSA)

Pinning Service API is meant for the purpose of performing operation on remote IPFS node of ZDFS. Pinning Service API spec is standardized specification for the developers and pinning service providers. Pinning Service API reduces the learning curve because of standardization. Learn More.

2. Data APIs

ZDFS goes beyond in order to make the remote operation and integration of content with your application more effective. That's why your control on your content is not limited to just a few pinning methods but there are multiple other ways in which you fully access your content and make a seamless pipeline with your application. Learn More

37.4.3 ZDFS GUI

ZDFS provides you with a good and interactive console to manage your content on a click of a button. You can visualise your content and play around. Perform all the operations graphically which are possible through *Data APIs*. You can even check file CID and availability on the IPFS Public Network without being uploaded on it.

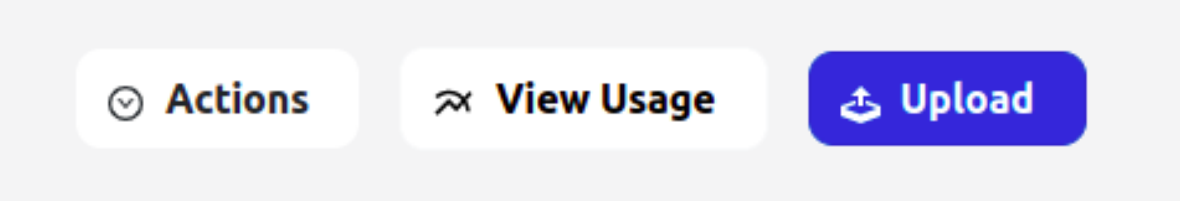
You can perform multiple file based operations using ZDFS console manager in the following optimized ways.

1. *File Upload*.
2. *File List*.
3. *File Detail*.
4. *File Edit*.
5. *File Delete*.
6. *File Export*.
7. *File Unpin*.
8. *File Pin*.
9. *File Preview*.

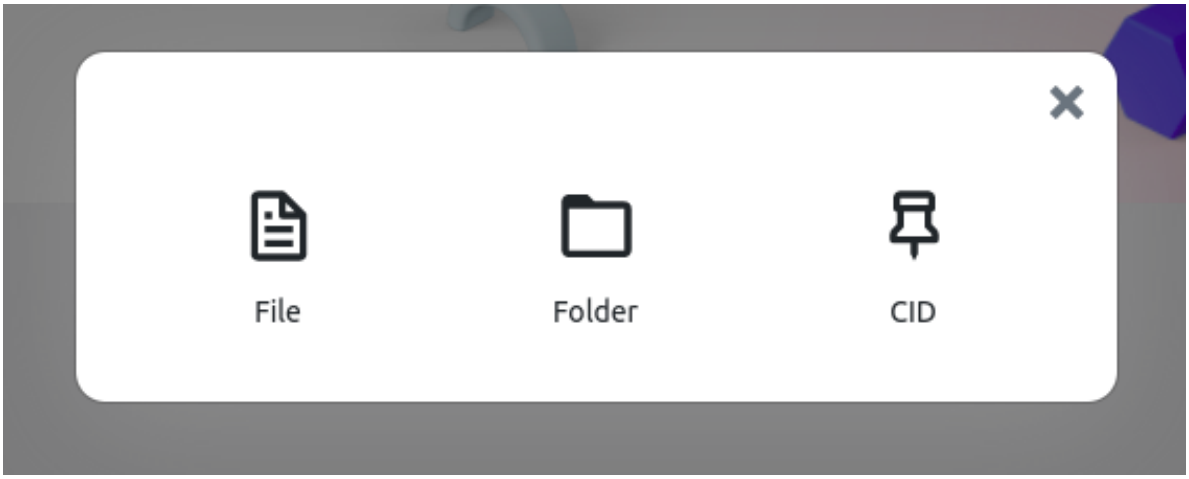
File Upload

You can upload your content using file, folder or by file CID upto individual file size 25GB.

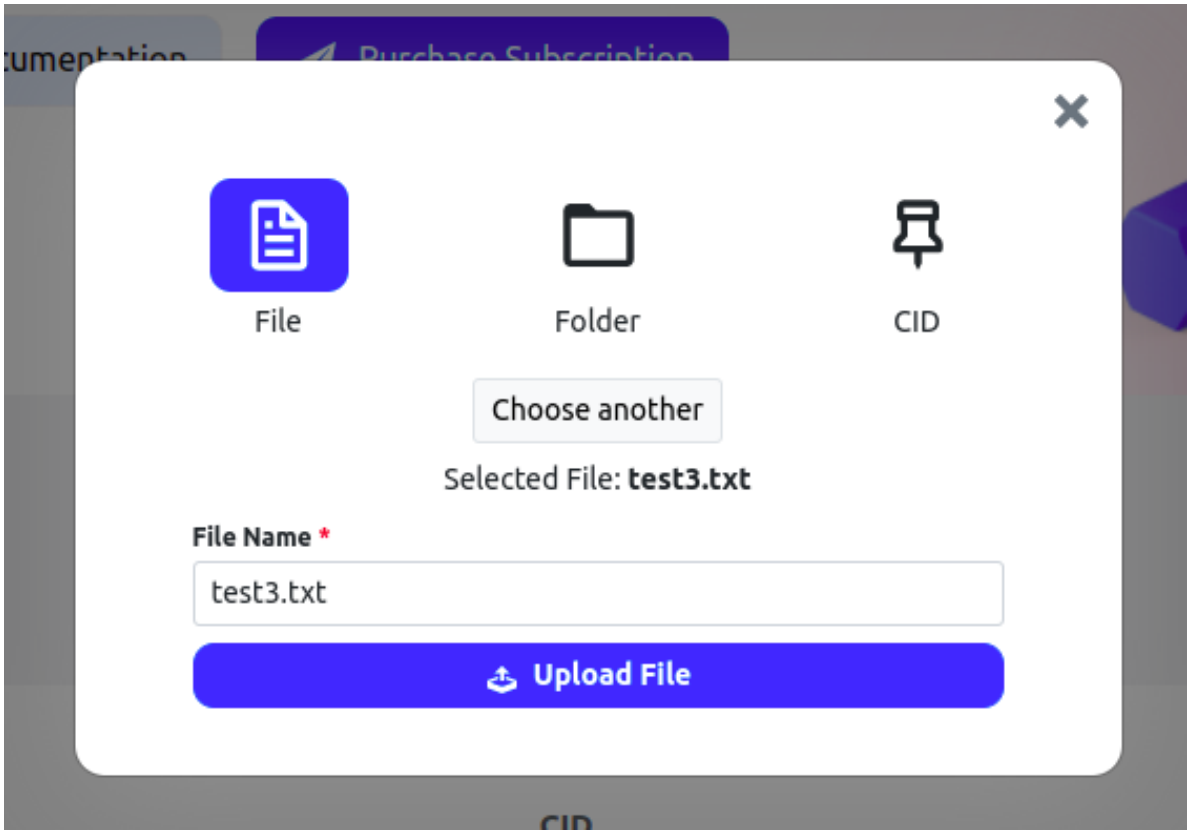
- Go to `ZDFS -> Files`.
- Click on **Upload** button.



- Choose from the provided upload method.



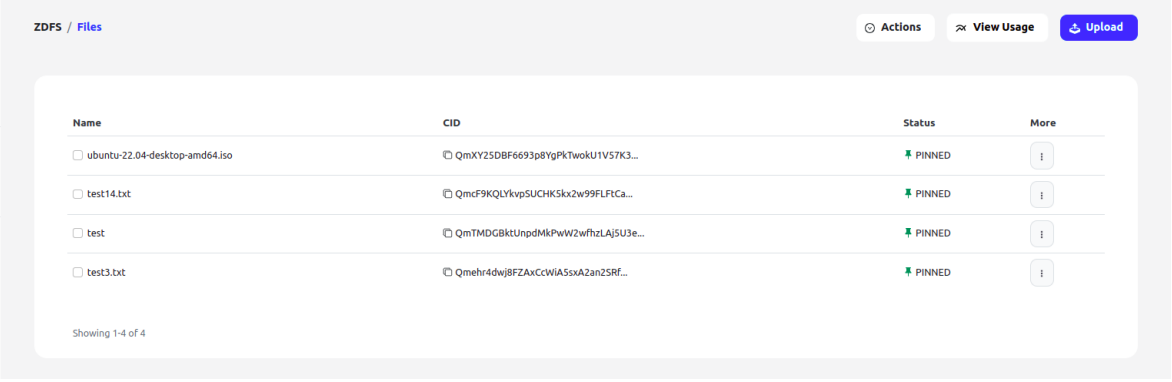
- Select a file and click on **Upload File** button to complete the process.



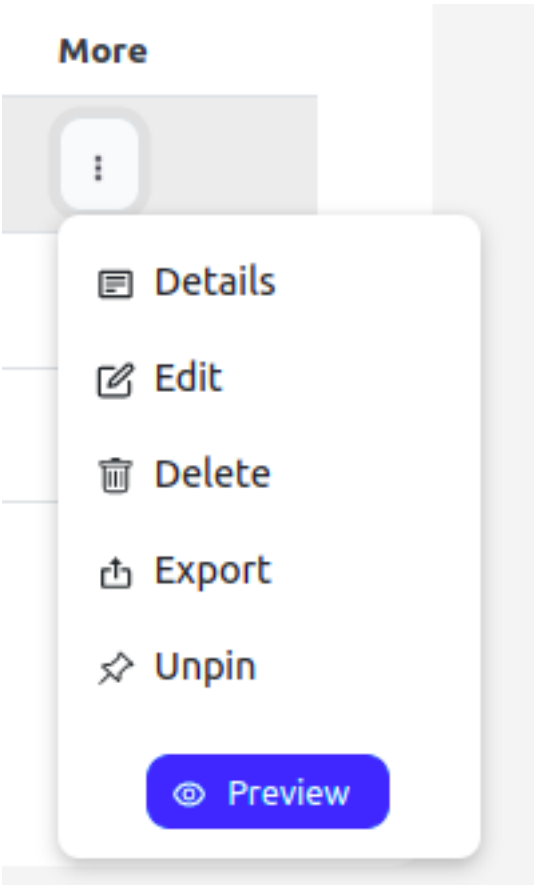
File List

You can organize your file or folder at the same place so that you can perform different operations individually.

- Go to `ZDFS -> Files`.



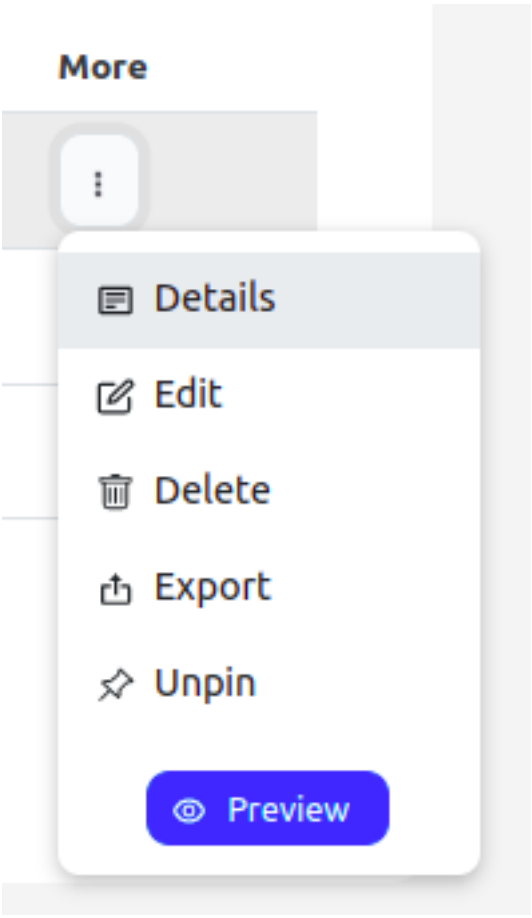
- Now, you can perform the following operations by clicking on the More (three vertical dot) button.



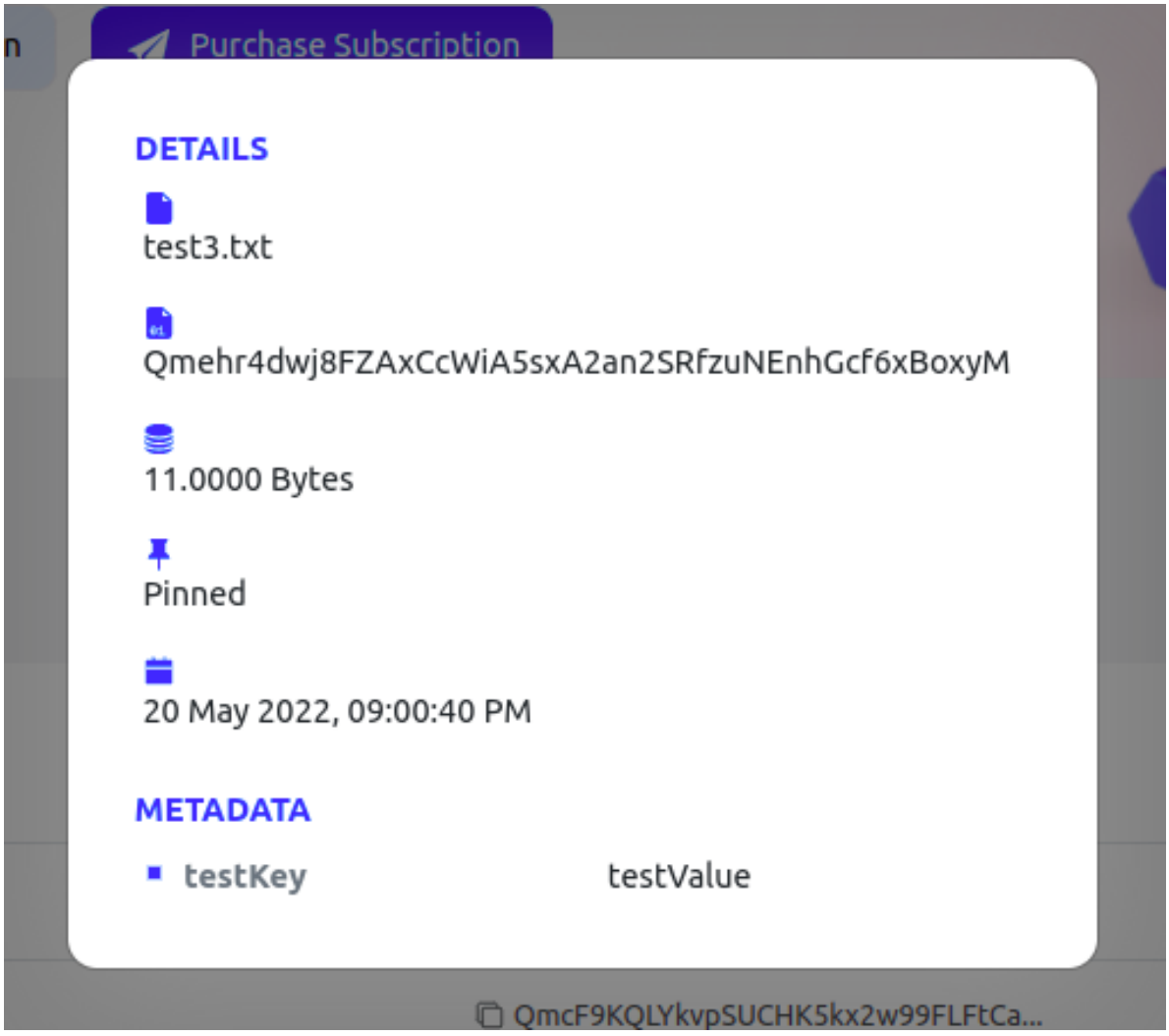
File Detail

You can check the details of your file in a more readable manner.

- Go to `ZDFS -> Files -> More -> Details`.



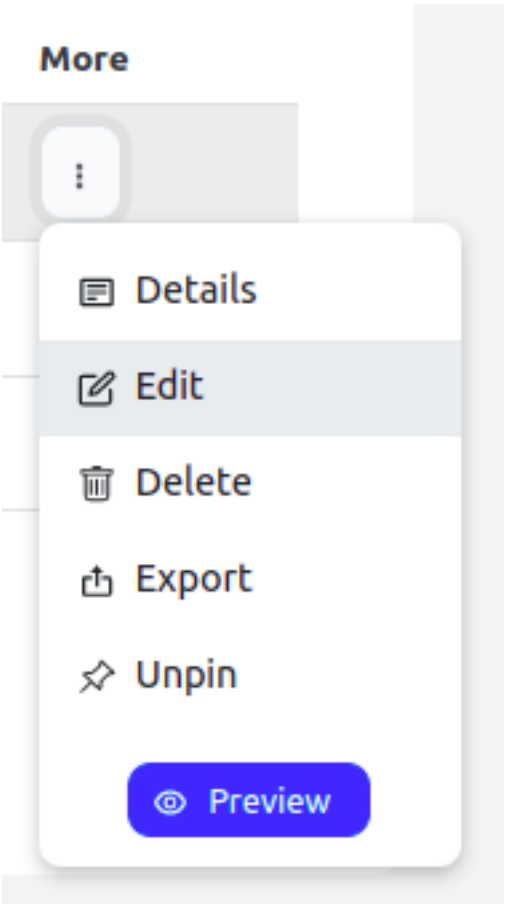
- Click on **Detail** button to view file detail.



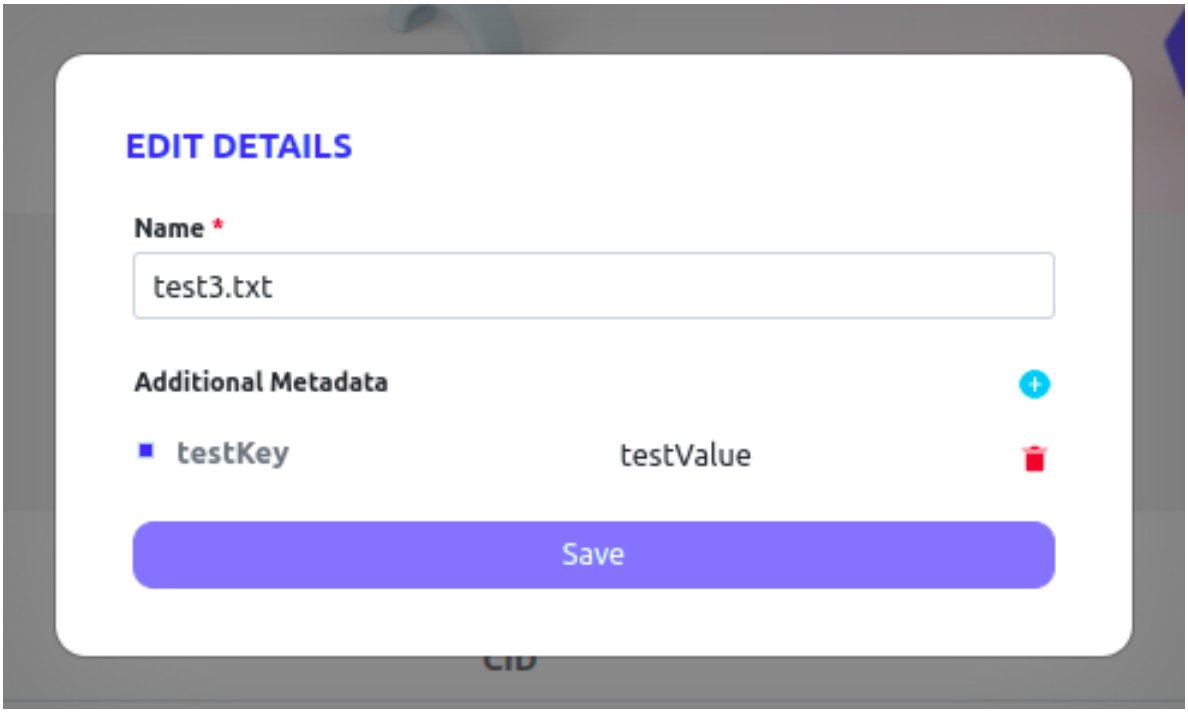
File Edit

You can change the file or folder name along with the meta data.

- Go to `ZDFS -> Files -> More -> Edit`.



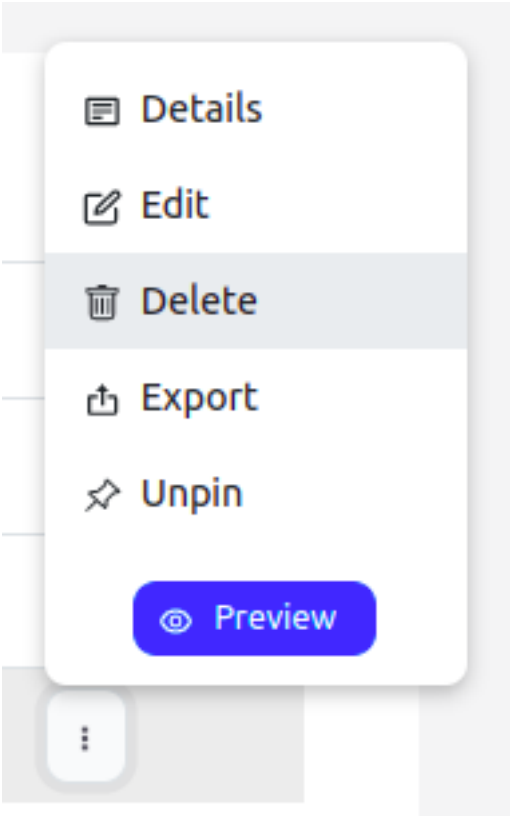
- Click on **Edit** button to edit the file detail.



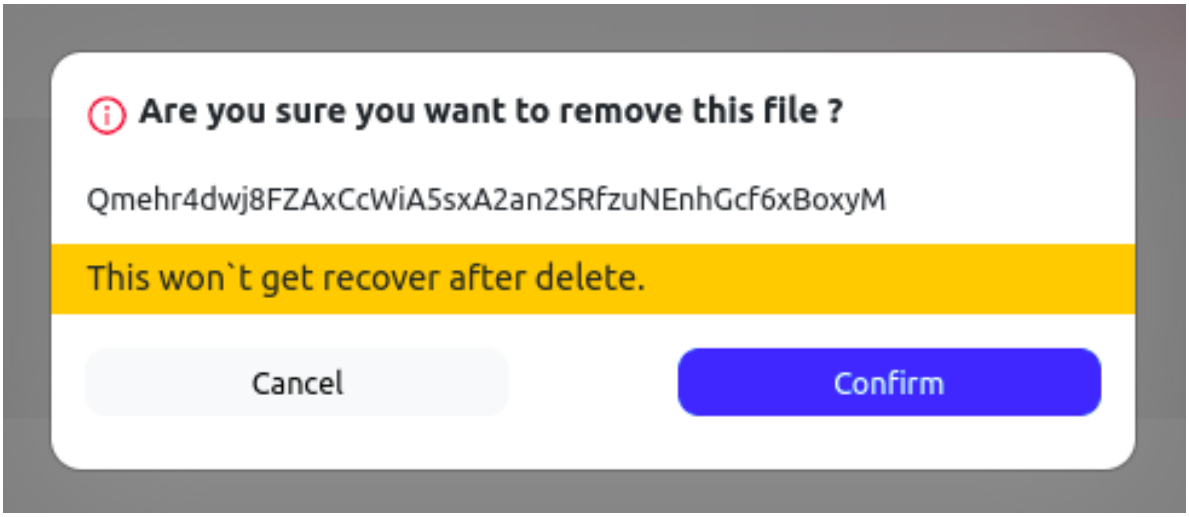
File Delete

You can delete any content such that it will get automatically unpinned while removing its entry in ZDFS.

- Go to `ZDFS -> Files -> More -> Delete`.



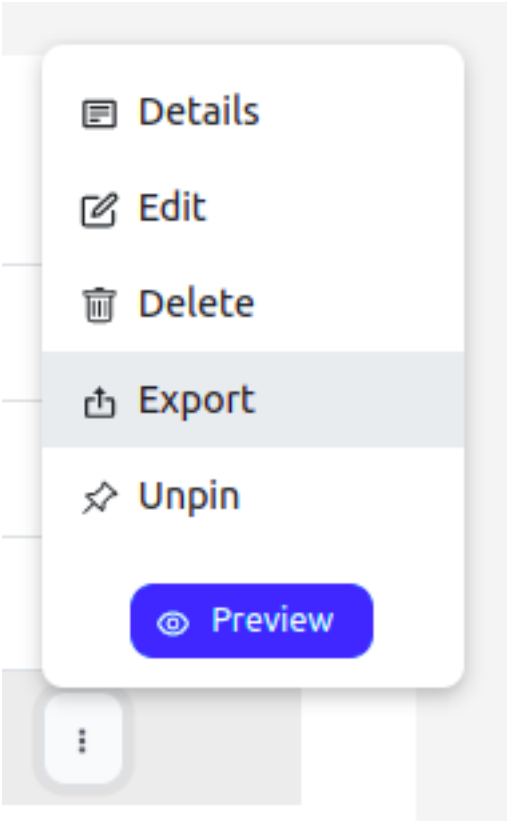
- Now you will see the confirmation dialogue box to reinsure delete.



File Export

You can download your content by simply following the steps.

- Go to `ZDFS -> Files -> More -> Export`.

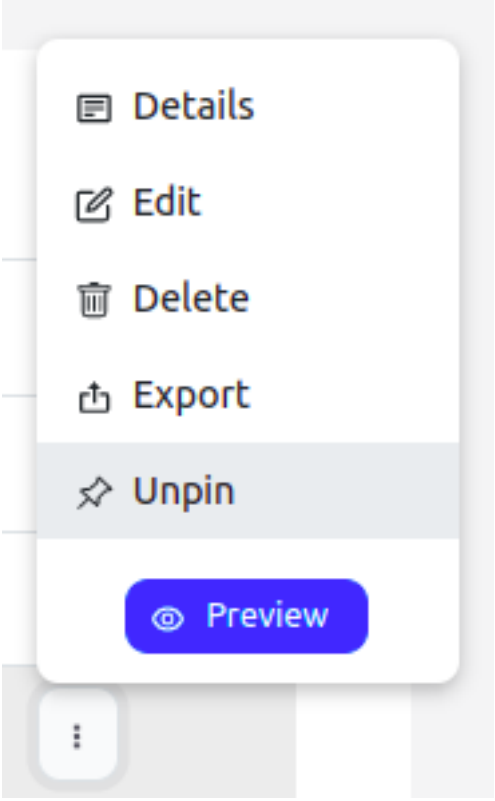


- Click on **Export** button to start downloading.

File Unpin

You can unpin the already pinned file or folder.

- Go to `ZDFS -> Files -> More -> Unpin`.

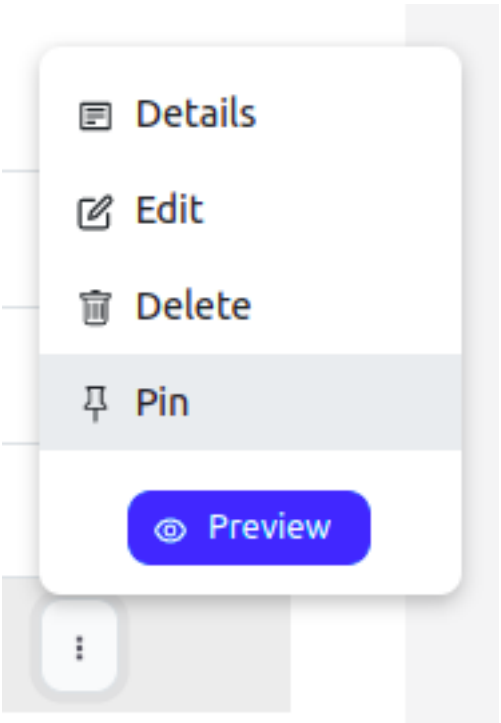


- Click on **Unpin** button to make the file unpin.

File Pin

You can pin the unpinned file or folder.

- Go to `ZDFS -> Files -> More -> Pin`.

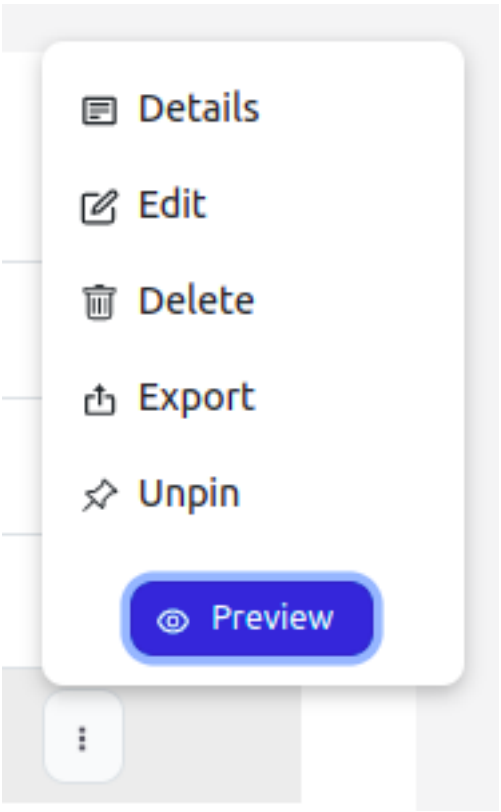


- Click on **Pin** button to make the file pin.

File Preview

You can see your file directly on the IPFS gateway.

- Go to `ZDFS -> Files -> More -> Preview`.

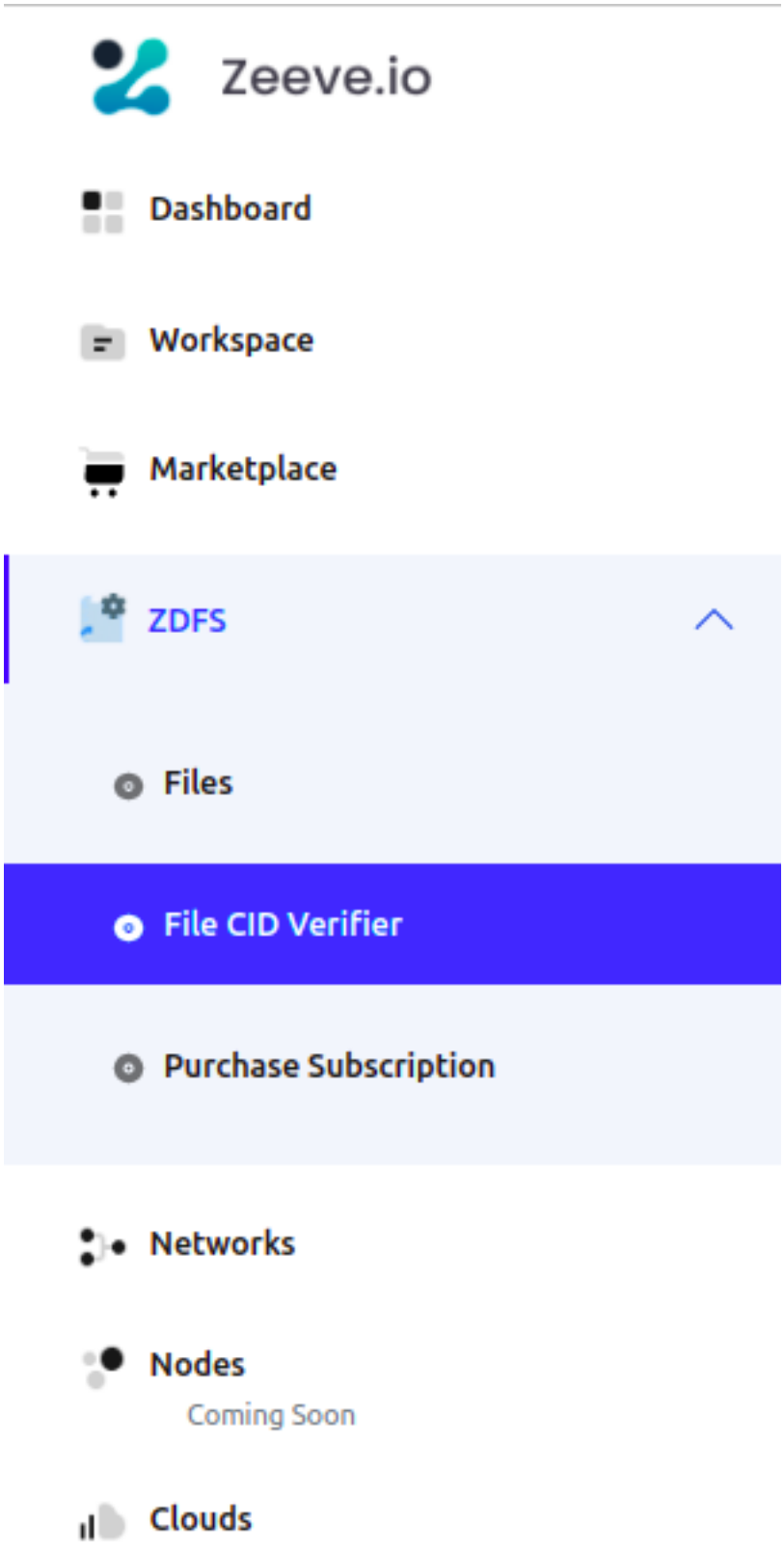


- Click on **Preview** button to see your file.

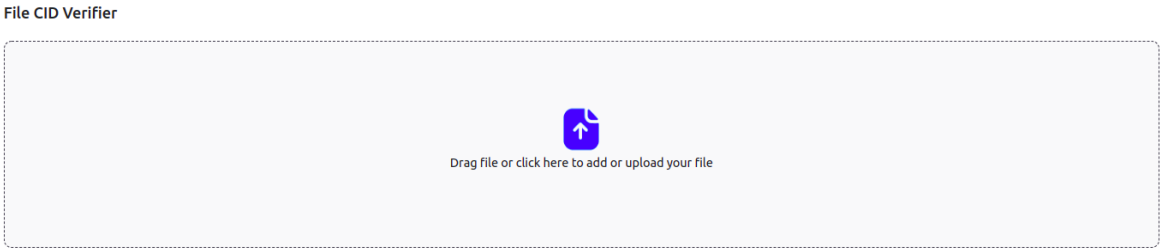
File CID Verifier

ZDFS offers you to check the IPFS file CID IPFS hash of version 0 and 1 hashing schema. And also provides you to check its availability on IPFS Public Gateway. ZDFS ensures that you can check file CID IPFS hash of a particular content without being uploaded to the IPFS Public Gateway.

- Go to `ZDFS -> File CID Verifier`.



- Drag file or click inside the dotted area.



- Now you will be able to see file CID hash and its availability for both the cases.

- Click on **Clear** button to check for another file.

NOTE: *File might get available late on IPFS Public Network after upload because it takes some time to propagate on IPFS Public Network*

CHAPTER

THIRTYEIGHT

DESCRIPTION: GET SUPPORT FOR USING ZEEVE'S PLATFORM. OUR DOCUMENTATION PROVIDES RESOURCES FOR TROUBLESHOOTING AND RESOLVING COMMON ISSUES, AS WELL AS INFORMATION ON HOW TO CONTACT OUR SUPPORT TEAM FOR FURTHER ASSISTANCE. ROBOTS: NOINDEX

CHAPTER

THIRTYNINE

ZEEVE SUPPORT

Stuck anywhere? Need help about anything? Want some kind of clarification? Want to pour in your suggestions? Want to raise feature requests? We are easily reachable and are all ears, eyes and mind for you on the following channels :

Chat Client : You can connect with us via our chat client. This client can be found at the bottom right corner across the Zeeve platform. Talk to us, by just hitting that green button you find there.

Telegram : You can also join our active developer community on telegram.

CHAPTER

FORTY

GLOSSARY

40.1 Access Key

Access Key ID helps in accessing and controlling the cloud account and services like EC2, S3, SimpleDB, CloudFront, SQS, EMR, RDS, etc. In case of AWS its length is 20 alphanumeric characters long like ABCDGHJK1234CBDG123C . It can be shared with others as well.

40.2 Bitcoin

 Bitcoin is an open source public blockchain in which anyone can participate. Currency exchange in this is termed as Bitcoin, which is having value in stock exchange as well. It is a peer-to-peer technology to operate with no central authority or banks; managing transactions and the issuing of bitcoins is carried out collectively by the network. Ref.

40.3 Blockchain

Blockchain is considered as a chain of blocks. Blocks are nothing but collection of data. Blockchain is an immutable time-stamped series of data that is maintained within a distributed network of peer nodes. These nodes each maintain a copy of the ledger by applying transactions that have been validated by a consensus protocol, grouped into blocks that include a hash that bind each block to the preceding block.

40.4 Certificate Authority

Certificate Authority accorss this documentation refers to HyperLedger Fabric's key component. This component is responsible for issuing identification material like certificates and keys for an organization.

40.5 Cloud

40.6 Distributed Ledger Technology (DLT)

Distributed Ledger is a backbone of Blockchain. Literal meaning of Ledger says that a book contains different records, which in turn is distributed in nature. DLT is a digital system which is recording the transactions and their details in distributed manner. Distributed ledger don't have any central data store.

40.7 Inode

40.8 Instance Type

In cloud instance is considered as a node termed as EC2 (Elastic Compute Cloud). Instance type is a term given which varies based upon the hardware configuration of instance created. In Zeeve we have various instance type both in AWS as well as in Azure. For more information click for AWS Instance Type and Azure Instance Type.

40.9 IOT

Internet of Things

40.10 Kafka

Kafka is a message handling system which uses Publish-Subscribe model. Consumers subscribe to the topic to receive new messages, that are published by a Producer. Topics are nothing but messages, so when they become huge in number, then they are split into partitions, and Kafka guarantees that all messages inside a partition are sequentially ordered. Hyperledger Fabric ordering service nodes (OSNs) use the Kafka cluster and provide an ordering service to your blockchain network. Kafka is permissioned voting based consensus type, here leader does the ordering, only in-sync can be voted as leader.

40.11 Node

Node is an instance of a network. It basically a connection point which is helping in receiving, creating, storing and sending data.

40.12 Network

Network is a collection of nodes/instances. In Zeeve, when you create a network over the cloud then it asks you for providing the number of nodes and other configuration details. Once all the installation and configuration gets over, it results in [[How_to_create_my_first_network|creating]] your Blockchain network.

40.13 Orderer

Orderers are considered as the special nodes, which are helping each peer nodes to have consistent ledger by enabling the interaction of peer nodes and applications participating in the network.

40.14 Raft

Raft based Ordering service in fabric.

40.15 Secret Key

Secret key is used along with Access key in order to access and control the cloud account and services like EC2, S3, SimpleDB, CloudFront, SQS, EMR, RDS, etc. In case of AWS its length is 40 alphanumeric-slash-plus characters long like ******KOsJW/******/c*********+***wy . It can't be shared with others.

40.16 Smart Contract

Smart Contract is a piece of code that contains the business logic. It's execution is done in a secured environment. This is the part of the blockchain that ensures validity of data going into it.

40.17 Zookeeper

Zookeeper is a distributed key-value store, most commonly used to store metadata and handle the mechanics of clustering. It allows clients of the service (the Kafka brokers) to subscribe and have changes sent to them once they happen. This is how brokers know when to switch partition leaders. Zookeeper is also extremely fault-tolerant as it ought to be, since Kafka heavily depends on it.

CHAPTER

FORTYONE

RELEASES

41.1 Zeeve 1.5.0 IBM Cloud Release

Now use your favourite Cloud to Deploy your favourite Blockchain. This update includes IBM cloud support, which allows you to deploy the choice of Blockchain deployments in a few clicks.

Key Highlights of this release, Now deploy on IBM cloud:

- Hyperledger Fabric
- Hyperledger Sawtooth
- R3 Corda
- Ethereum
- Credits

What's up next

- Support for private networks deployment for Corda.
- One click CorDapps deployment using Marketplace
- Google Cloud support.

Did someone say bugs? We have restocked the bug repellent to allow seamless experience.

- Night mode is shinier then it was.

41.2 Zeeve 1.4.0 R3 Corda Release

We continuously upgrade Zeeve with smaller updates having bigger impacts. This release will be the addition of one of the leading Consortium protocol to the platform. Zeeve will now allow you to deploy Corda Production, Pre-Production and Test net on demand.

Key Highlights of this release:

- Corda as a protocol choice to deploy
- Option for selecting your public zone:-

  - R3 production zone
  - R3 pre-production zone
  - R3 test-net
- Selecting version for your corda nodes/notaries.
- Specifying detailed configurations for your corda nodes and notaries.
  - Also allowing options for doing initial registration or to upload your keystore/SSLstore files with custom passwords and other CSR details.
- Currently supports deployment on AWS, Azure, Digital Ocean, IBM Cloud and Google cloud.

What's up next

- Support for private networks deployment for Corda.
- One click CorDapps deployment using Marketplace
- Google Cloud support.
- IBM Cloud (BlueMix) support.

Did someone say bugs? We now have made Zeeve stable more than ever. And these as well:

- Project creation bug fixed for Digital Ocean.
- Day/Night visual mode is persistent now.

41.3 Zeeve Digital Ocean Follow-up Release

We are committed to continuously update Zeeve and provide you all the Blockchain essentials required to power up your Blockchain Business needs. In the latest Follow-up Release, Zeeve now allows you to Deploy and Manage Hyperledger Fabric Networks on DigitalOcean Cloud using Kubernetes.

What Support for Fabric allows you to do on DO:

• Create & scale Fabric networks on Digital Ocean Cloud • Create & download cryptographic artifacts • Channel configurations • Choice of consensus - RAFT, Solo & Kafka • Organization wise Orderers • Choice between LevelDb and CouchDb for peer • Configure CAs with admin's user-name and password • Persistent volume option for every service • Option to create and join application channels • Option to add more nodes to existing networks

Did someone say bugs? We now have made Zeeve stable more than ever.

Check now @ https://www.zeeve.io

41.4 Zeeve 1.3.0 Major Release

Another upgrade to upgrade your Blockchain experience. Zeeve 1.3.0 has been released and is available for use. We now support DigitalOcean, this allows you to deploy your Blockchain networks in your own DigitalOcean cloud infrastructure and its just a few clicks away.

Key Highlights of this release:

- Deploy Ethereum, Sawtooth and Credits networks on DigitalOcean.
- Fabric Networks are more configurable now with
- Support for RAFT consensus
- Organization wise Orderers
- Choice between LevelDb and CouchDb for peer.
- Configure CAs with admin's username and password.
- Persistent volume option for every service.
- Option to create and join application channel added.
- Option to add more nodes to the network.

What's up next

- R3 Corda network deployments
- Zeeve managed deployments
- Fabric support on DigitalOcean

Did someone say bugs? We now have made Zeeve stable more than ever. And these as well:

- Error notifications are more detailed
- You can see cloud VPC limit errors in Zeeve

Check now @ https://www.zeeve.io

41.5 Zeeve 1.2.0 Major Release

Credits Public Blockchain deployment is now Supported

If you are a developer then you must be familiar with Credits already as an open-source, fully decentralized blockchain software protocol. Now it is simpler then ever to do Credits. If you are a developer working on credits or a business running a production grade solution, you can now sip a coffee, deploy, monitor and manage your Credits deployment on cloud with ease.

Credits is now integrated with worlds leading Blockchain as a Service platform Zeeve (https://zeeve.io). Zeeve will power up your credits development and production environments. With a few clicks, it will enable you to deploy credits on your own cloud infrastructure.

1 2 3 Credits with Zeeve:

- Choose among your choice of cloud providers, i.e. AWS and Azure
- Sign up with Zeeve and authorize your cloud account through your profile
- Launch a Credits MainNet Node through create a network option
- Monitor resources, start/stop or delete on demand
- Continue to create more nodes in different cloud data centers

Whats coming up for Credits

- TestNet support to be available soon
- Support for hosted infrastructure for seamless and managed deployments
- Wallet support for Credits
- Smart Contract deployments
- We hear you! You can submit feedback/features to us to improve

We have also squashed a few bugs and fixed some annoying issues. Please feel free to test the beta, we are open to suggestions, issues and feature requests.

Check now @ https://www.zeeve.io

41.6 Zeeve 1.1.1 Minor Release

IoT as a Service is the latest offering.

Zeeve 1.1.1 has been released and is available for use. Platform already had the support to deploy Hyperledger Sawtooth, Fabric and Ethereum. Now we have added support for IoT Services to it too.

Key highlights of 1.1.1

- Create data publish/consume APIs
- Create and Manage API service keys for IoT data access
- Fabric stability and improvements
- Now you can delete/start/stop/restart fabric nodes.
- Azure cloud support is now enabled for Sawtooth and Ethereum.
- Fabric Network now allows users to download the blockchain artifacts.
- Live logs are now available while you watch your Tasks.

And there is more

- More enhancements to Dark Mode
- Several performance improvements

We have also squashed a few bugs and fixed some annoying issues. Please feel free to test the beta, we are open to suggestions, issues and feature requests.

Check now @ https://www.zeeve.io

41.7  Zeeve 1.1.0 Released

Zeeve 1.1.0 has been released and is live.

The Major change you can enjoy is addition of Hyperledger Fabric, while it already supported Hyperledger Sawtooth and Ethereum.

Fabric release allows

- Creation of cryptographic artifacts
- Supports Channel configurations
- Kubernetes based cloud service on your own AWS account
- Deploy production grade multi-ordered network Choice of consensus
- Solo orderer
- Kafka based ordering

And there is more

- Dark mode is made even better
- Activity and logs are more detailed
- Dashboard now auto updates
- Several performance improvements

We have also squashed a few bugs and fixed some annoying issues. Please feel free to test the beta, we are open to suggestions, issues and feature requests.

Check now @ https://www.zeeve.io