June 11th, 2019

IBM BLOCKCHAIN PLATFORM

GETTING STARTED

CREATE A KUBERNETES SERVICE

1. Register/Login to the IBM Cloud
2. Press **Catalog** and search for Kubernetes
3. Press on **Kubernetes Service**
4. Scroll down to **Pricing Plans** and select **Free**
5. Press the **Create** button
6. Provide a **Cluster name** and select *"Dallas"* as the Service location
7. Wait for the service to get provisioned and started (may need very long)
8. Navigate to the IBM Cloud **Dashboard** and go to **Kubernetes Clusters**
9. Press on your Cluster name
10. The text "expires in a month" appears between the cluster name and the status *"Normal"*
11. Press over the word **Access**
12. Start a terminal console and follow the instructions to *"Gain access to your cluster"*

CREATE A BLOCKCHAIN SERVICE INSTANCE ON IBM CLOUD

1. On a new browser tab, go to cloud.ibm.com
2. Using the **Catalog** search for *"Blockchain"*
3. Select **Blockchain Platform**
4. Provide a *"service name"* and KEEP *"Dallas"* as the location to deploy
5. Scroll down. Under **Pricing Plans** select *"Beta trial"*
6. Press on **Create**
7. The *"Welcome to the IBM Blockchain Platform!"* page will show up
8. Press **"I have a cluster" (Skip to Link to a cluster)**
9. Select the Kubernetes Cluster you have created before
10. Click on **Deploy to Cluster**
11. Wait until the **Launch the IBM Blockchain Platform** button is highlighted
12. Press on this button
13. A new *"Welcome to the IBM Blockchain Platform"* including an architecture picture will appear
14. Press on **Let's get started**

CREATE A NETWORK TUTORIAL

Press on "Learn more about how to get the GA product". The "Build a Network Tutorial" page appears

Step One: Create a peer organization and a peer

For each organization that you want to create with the console, you should deploy at least one CA. A CA is the node that issues certificates to all network participants (peers, ordering services, clients, admins, and so on). These certificates, which include a signing certificate and private key, allow network participants to communicate, authenticate, and ultimately transact. These CAs will create all of the identities and certificates that belong to your organization, in addition to defining the organization itself. You can then use those identities to deploy nodes, create admin identities, and submit transactions.

In this tutorial, we create two organizations, one of which will own a peer, and another will own an ordering service. Each organization needs a CA to issue its certificates, therefore we need to create **two CAs**. For the purpose of this tutorial, **we will create only one CA at a time**.

Creating your peer organization's CA

1. Navigate to the **Nodes** tab on the left and click **Add Certificate Authority**. The side panels will allow you to customize the CA that you want to create and the organization that this CA will issue keys for.
2. Click **Create an IBM Cloud Certificate Authority** and **Next**.
3. Use the second side panel to give your CA a **display name**. Our recommended value for this CA is `Org1 CA`.
4. On the next panel, give your CA admin credentials by specifying a **CA administrator enroll ID** of `admin` and a secret of `adminpw`. Again, these are **recommended values**.
5. Accept all the defaults and click **Next**. You will see the **Summary** page.
6. Review the Summary page, then click **Add certificate authority**.

Using your CA to register identities

Once the CA is running, as indicated by the green box in the tile, generate these certificates by completing the following steps:

1. Click on the `Org1 CA` and ensure the `admin` identity that you created for the CA is visible in the table. Then click the **Register User** button.
2. First, we'll register the organization admin, which we can do by giving an **Enroll ID** of `org1admin` and a **secret** of `org1adminpw`. Then set the `Type` for this identity as `client` (admin identities should always be registered as `client`, while node identities should always be registered using the `peer` type). You can ignore the **Maximum enrollments** field. Click **Next**.
3. For the purpose of this tutorial, we do not need to use **Add Attribute**.
4. After the organization admin has been registered, repeat this same process for the identity of the peer (also using the `Org1 CA`). For the peer identity, give an enroll ID of `peer1` and a secret of `peer1pw`. This is a node identity, so select `peer` as the **Type**. You can ignore the **Maximum enrollments** field and, on the next panel, do not assign any **Attributes**, as before.

Creating the peer organization MSP definition

Now that we have created the peer's CA and used it to **register** identities for the `Org1` admin and for the peer we'll be associating with `Org1`, we need to create a formal definition of the peer's organization, which is known as an MSP. Note that many peers can belong to an organization. **You do not need to create a new organization every time you create a peer**. Because this is the first time that we go through the tutorial, we will create the MSP ID for this organization. During the process of creating the MSP, we will enroll the `org1admin` identity and add it to our Wallet.

1. Navigate to the **Organizations** tab in the left navigation and click **Create MSP definition**.
2. Give your MSP the display name `Org1 MSP` and an MSP ID of `org1msp`
3. Under **Root Certificate Authority details**, specify the CA you used to register the identities in the previous step. If this is your first time through this tutorial, you should only see one: `Org1 CA`.
4. The **Enroll ID** and **Enroll secret** fields below this will auto populate with the enroll ID and secret for the first user that you created with your CA: `admin` and `adminpw`. However, using this identity would give your organization the same admin identity as your CA, which for security reasons is not recommended. Instead, select the enroll ID you created for your organization admin from the drop-down list, `org1admin`, and enter its associated secret, `org1adminpw`. Then, give this identity a display name, `Org1 Admin`.
5. Click the **Generate** button to enroll this identity as the admin of your organization and export the identity to the Wallet, where it will be used when creating the peer and creating channels.
6. Click **Export** to export the admin certificates to your file system. As we said above, this identity is not stored in your console or managed by IBM. It is only stored in local browser storage. If you change browsers, you will need to import this identity into your Wallet to be able to administer the peer.

7. Click **Create MSP definition**.

After you have created the MSP, you should be able to see the peer organization admin in your **Wallet**, which can be accessed by clicking on the **Wallet** in the left navigation.

Creating a peer

After you have created the Org1 CA, used it to register Org1 identities, and created the Org1 MSP, you're ready to create a peer for Org1.

What role do peers play?

It's important to remember that organizations themselves do not maintain ledgers. Peers do. Organizations also use peers to sign transaction proposals and approve channel configuration updates. Because having at least two peers per organization on a channel makes them highly available, having three peers per organization joined to a channel is considered a best practice for production level implementations because it ensures high availability even while a peer is down for maintenance. In this tutorial though, we'll only show the process for creating a single peer. You can replicate the process to suit your own business needs.
From a resource allocation perspective, it is possible to join the same peers to multiple channels. The design of the peer ensures that the data from one channel cannot pass to another through the peer. However, because the peer will store a separate ledger for each channel, it is necessary to ensure that the peer has enough processing power and storage to handle the transaction and data load.

Deploying your peer

1. On the **Nodes** page, click **Add peer**.
2. Click **Create an IBM Cloud peer** and **Next**.
3. Give your peer a **Display name** of `Peer Org1`.
4. On the next screen, select `Org1 CA`, as this is the CA you used to register the peer identity. Select the **Enroll ID** for the peer identity that you created for your peer from the drop-down list, `peer1`, and enter its associated **secret**, `peer1pw`. Then, select `Org1 MSP` from the drop-down list and click **Next**.
5. The next side panel asks for TLS CA information. When you created the CA, a TLSCA was created alongside it. This CA is used to create certificates for the secure communication layer for nodes. Therefore, select the **Enroll ID** for the peer identity that you created for your peer from the drop-down list, `peer1`, and enter the associated **secret**, `peer1pw`. The **TLS Certificate Signing Request (CSR) hostname** is an option available to advanced users who want to specify a custom domain name that can be used to address the peer endpoint. Custom domain names are not a part of this tutorial, so leave the **TLS CSR hostname** blank for now.
6. The next side panel asks you to **Associate an identity** to make it the admin of your peer. For the purpose of this tutorial, make your organization admin, `Org1 Admin`, the admin of your peer as well. It is possible to register and enroll a different identity with the `Org1 CA` and make that identity the admin of your peer, but this tutorial uses the `Org1 Admin` identity.
7. If you are using a free cluster, you see the **Summary** page.
8. Review the summary and click **Add peer**.

Step Two: Create the Ordering Service

In other distributed blockchains, such as Ethereum and Bitcoin, there is no central authority that orders transactions and sends them out to peers. Hyperledger Fabric, the blockchain that the IBM Blockchain Platform is based on, works differently. It features a node, or a cluster of nodes, called an **ordering service**.
The ordering service is a key component in a network because it performs a few essential functions:

- They literally **order** the blocks of transactions that are sent to the peers to be written to their ledgers. This process is called "ordering".
- They maintain the **ordering system channel**, the place where the **consortium**, the list of peer organizations permitted to create channels, resides. A consortium is essentially a multi-tenancy vehicle, and a single ordering service by design can host multiple consortia.
- They **enforce the policies** decided by the consortium or the channel administrators. These policies dictate everything from who gets to read or write to a channel, to who can create or modify a channel. For example, when a network participant asks to modify a channel or consortium policy, the ordering service processes the request to see if the participant has the proper administrative rights for that configuration update, validates it against the existing configuration, generates a new configuration, and relays it to the peers.

Creating your ordering service organization CA

The process for creating a CA for an ordering service is identical to creating it for a peer.

1. Navigate to the **Nodes** tab and click **Add Certificate Authority**.
2. Click **Create an IBM Cloud Certificate Authority** and **Next**
3. Give this CA a unique display name, `Ordering Service CA`.
4. You're free to reuse the **CA administrator enroll ID** of `admin` and a secret of `adminpw`. As this is a different CA, this identity is distinct from the CA admin identity for created for the `Org1 CA`, even though the ID and secret are identical.
5. If you are using a free cluster, you will see the **Summary** page.
6. Review the Summary page, then click **Add certificate authority**.

Using your CA to register ordering service node and ordering service admin identities

As we did with the peer, we need to register two identities with our ordering service CA. After selecting your CA, you will need to register an admin for our ordering service organization and an identity for the ordering service itself. As before, you should see an identity on the `Ordering Service CA` tab; it's the admin that you created for the CA.

Once the CA is running, as indicated by the green box in the tile for the `Ordering Service CA`, generate these certificates by completing the following steps:

1. Click on the `Ordering Service CA` in the **Nodes** tab and ensure the `admin` identity that you created for the CA is visible in the table. Then click the **Register User** button.
2. First, we'll register the organization admin, which we can do by giving an **Enroll ID** of `OSadmin` and a **secret** of `OSadminpw`. Then set the `Type` for this identity as `client` (admin identities should always be registered as `client`, while node identities should always be registered using the `peer` type). You can ignore the **Maximum enrollments** field. If you want to learn more about enrollments, see Registering identities. Click **Next**.
3. For the purpose of this tutorial, we do not need to use **Add Attribute**. If you want to learn more about identity attributes, see Registering identities.

4. After the organization admin has been registered, repeat this same process for the identity of the ordering service (also using the `Ordering Service CA`). For the ordering service node identities, give an enroll ID of `OS1` and a secret of `OS1pw`. This is a node identity, so select `peer` as the **Type**. You can ignore the **Maximum enrollments** field and, on the next panel, do not assign any **Attributes**, as before.

Creating the ordering service organization MSP definition

Create your ordering service organization MSP definition and specify the admin identity for the organization. After we have registered the ordering service admin and ordering service users, we need to create the MSP ID and enroll the `OSadmin` user that we registered as the admin of our organization.

1. Navigate to the **Organizations** tab in the left navigation and click **Create MSP definition**.
2. Give your MSP definition the display name `Ordering Service MSP` and an MSP ID of `osmsp`.
3. Under **Root Certificate Authority details**, select the `Ordering Service CA` we created.
4. The **Enroll ID** and **Enroll secret** fields below this will auto populate with the enroll ID and secret for the first user that you created with your CA: `admin` and `adminpw`. However, using this identity would make your organization the same identity as your CA identity, which for security reasons is not recommended. Instead, select the enroll ID that you created for your organization admin from the drop-down list, `OSadmin`, and enter its associated secret, `OSadminpw`. Then, give this identity a display name, `Ordering Service Admin`.
5. Click the **Generate** button to enroll this identity as the admin of your organization and export the identity to the Wallet.
6. Click **Export** to export the admin certificates to your file system. As we said above, this identity is not stored in your console or managed by IBM. It is only stored in your browser. If you change browsers, you will need to import this identity to be able to administer the ordering service.
7. Click **Create MSP definition**.

After you have created the MSP, you should be able to see the ordering service organization admin in your **Wallet**, which can be accessed by clicking on the **Wallet** in the left navigation.

Deploy the ordering nodes

Perform the following steps from your console:

1. On the **Nodes** page, click **Add ordering service**.
2. Click **Create an IBM Cloud Ordering service** and click **Next**.
3. Give your ordering service a **Display name** of `Ordering Service`. Since this is a free cluster, only one ordering node can be created. and, if in a paid cluster, choose whether you want your ordering service to have one node (sufficient for testing) or five nodes (good for production). Choose **five nodes**. And do not choose to use an external CA. This is an advanced option.
4. On the next panel, select `Ordering Service CA` as your CA. Then, select the **enroll ID** for the node identity that you created for your ordering service from the drop-down list, `OS1`, and enter the associated **secret**, `OS1pw`. Then, select your MSP, `Ordering Service MSP` from the drop-down list. For the purpose of this tutorial, do not choose to use an external CA for your ordering service, though if you want more information, see Using certificates from an external CA. Click **Next**.
5. The next side panel asks for TLS CA information. When you created the CA, a TLS CA was created alongside it. This CA is used to create certificates for the secure communication layer for nodes. Therefore, select the **enroll ID** for the ordering service identity that you created from the drop-down list, `OS1`, and enter its associated **secret**, `OS1pw`. The **TLS Certificate Signing Request (CSR) hostname** is an option available to advanced users who want to specify a custom domain name that can be used to address the ordering service endpoint. Custom domain names are not a part of this tutorial, so leave the **TLS CSR hostname** blank for now.
6. The **Associate identity** step allows you to choose an admin for your ordering service. Select `Ordering Service Admin` as before and click **Next**.
7. Review the Summary page and click **Add ordering service**.

After the ordering service has been created, you are able to see it on the **Nodes** panel.

Step Three: Join the consortium hosted by the ordering service

As we noted earlier, a peer organization must be known to the ordering service before it can create or join a channel (this is also known as joining the "consortium", the list of organizations known to the ordering service). This is because channels are, at a technical level, **messaging paths** between peers through the ordering service.

Just as a peer can be joined to multiple channels without information passing from one channel to another, so too can an ordering service have multiple channels running through it without exposing data to organizations on other channels.
Because only ordering service admins can add peer organizations to the consortium, you will either need to **be** the ordering service admin or **send** MSP information to the ordering service admin.

Because you created the ordering service admin using the console, this process is relatively straightforward:

1. Navigate to the **Nodes** tab.
2. Scroll down to the ordering service you created and click on it to open it.
3. Under **Consortium Members**, click **Add organization**.
4. From the drop-down list, select `Org1 MSP`, as this is the MSP that represents the peer's organization: `Org1`.
5. Click **Add organization**.

When this process is complete, it is possible for `Org1` to create or join a channel hosted on your `Ordering Service`.
In this tutorial, we can easily access the `Org1 MSP` because both the peer organization and the ordering service organization were created in the same console. In a production scenario, the MSP definitions of other organization would be created by different network operators in their own cluster using their own IBM Blockchain console. In those cases, when the organization wants to join your consortium, the organization MSP definition of the organization will need to be sent to your console in an out of band operation. Additionally, you will need to export your ordering service and send it to them so they can import it into their console and join a peer to a channel (or create a new channel). This process is described in the Join a network tutorial under Exporting your organization information.

Step four: Create a channel

Although the members of a network are usually related business entities that want to transact with each other, there might be instances when subsets of the members want to transact without the knowledge of the others. This is possible by creating a **channel** on which these transactions will take place. Channels replicate the structure of a blockchain network in that they contain members, peers, an ordering service, a ledger, policies, and smart contracts. But by restricting the membership, and even the knowledge of the channel, to particular subsets of the network membership, channels ensure that network members can leverage the overall structure of the network while maintaining privacy, where needed.
As noted above, to join a peer from `Org1` to a channel, `Org1` must first be added to the consortium. If the organization is not a member of the consortium at channel creation time, it's possible to create the channel and add the organization later by clicking the **Settings** button on the page of the relevant channel and going through the **Update Channel** flow.

Creating a channel: `channel1`

Because the console uses peers to gather information about the channels that the peers belong to, **unless an organization has joined a peer to a channel, it cannot interact with the channel**.
When you have created your CAs, identities, MSPs, ordering service, a peer, and have added your peer organization to the consortium, navigate to the **Channels** tab in the left navigation. This is where channel creation and management are handled.
When you first navigate to this tab, it will be empty except for the **Create channel** and **Join channel** buttons. This is because you haven't created a channel and joined a peer to it yet.

Creating the Channel

Perform the following steps from your console:

1. Navigate to the **Channels** tab.
2. Click **Create channel**. A side panel will open.

3. Give the channel a **name**, `channel1`. Make a note of this value, as you will need to share it with anyone who wants to join this channel.
4. Select `Ordering Service` from the drop-down list.
5. Choose the **Organizations** who will be a part of this channel. As we have only created one organization, this will be `Org1 MSP (org1msp)`. Make this organization an **Operator**. Note: do not use the `Ordering Service MSP` here.
6. Choose a **Channel update policy** for the channel. This is the policy that will dictate how many organizations will have to approve updates to the channel configuration. As this tutorial only involves creating a single organization, this policy should be `1 out of 1`. As you add organizations to the channel, you should change this policy to reflect the needs of your use case. A sensible standard is to use a majority of organizations. For example, `3 out of 5`.
7. DO NOT specify any **Access control** limitations you want to make. Note: this is an **advanced option**. If you set the access to a resource to a particular organization, it will restrict access to that resource for every other organization in the channel. For example, if the default access to a particular resource is the `Readers` of all organizations, and that access is changed to the `Admin` of `Org1`, then **only** the admin of Org1 will have access to that resource. Because access to certain resources is fundamental to the smooth operation of a channel, it is highly recommended to make access control decisions carefully. If you decide to limit access to a resource, make sure that the access to that resource is added, as needed, for each organization.
8. Select the **Channel creator organization**. Because the console allows multiple organizations to be owned by a single user, it is necessary to specify which organization is creating the channel. Because this tutorial is limited to the creation of a single organization, choose `Org1 MSP` (org1msp) from the drop-down list. Likewise, choose `Org1 Admin` as the identity creating the channel.

When you are ready, click **Create channel**. You will be taken back to the **Channels** tab and you can see a pending tile of the channel that you just created.

Step five: Join your peer to the channel

We are almost done. Joining the peer to the channel is the last step in setting up the basic infrastructure for your network. If you are not already there, navigate to the **Channels** tab in the left navigation.

Perform the following steps from your console:

1. Click the pending tile for `channel1` to launch the side panel.
2. Select which peers you want to join to the channel. For purposes of this tutorial, click `Peer Org1`.
3. Click **Join channel**.

JOIN A NETWORK TUTORIAL

This current tutorial guides you through the process of joining an existing network by creating a peer and joining it to a channel.
Follow the instructions under:
https://cloud.ibm.com/docs/services/blockchain/howto?topic=blockchain-ibp-console-join-network#ibp-console-join-network