IBM® Storage

IBM Storage Solutions for Blockchain Platform

Version 1.2

IBM Storage Team

Contents

About this document

This Blueprint is intended to define the infrastructure that is required for a blockchain remote peer and to facilitate the deployment of IBM Blockchain Platform on IBM Cloud Private using that infrastructure. This infrastructure includes the necessary document handler components, such as IBM Blockchain Document Store, and covers the required storage for on-chain and off-chain blockchain data. To complete these tasks, you must have a basic understanding of each of the used components or have access the correct educational material to gain that knowledge.

Executive summary

IBM Blockchain enables more-efficient ledger models, streamlining business processes, and transactions, which reduces risk and increases trust. To gain a competitive advantage for your business, you need a reliable, secure, and flexible IT blockchain environment. This environment enables modern blockchain application enterprise workloads to scale as necessary to fit your needs. It also gives access to users, no matter what kind of endpoint device they are using.

Further, it should allow orchestration to suit your resource consumption requirements and minimize downtime. Your blockchain environment must provide reliable platform-as-a-service capabilities with flexible infrastructure. This means deploying a cloud-service fabric to reliably deliver containerized blockchain applications to your endpoints of choice to meet or exceed service-level expectations.

Organizations must also protect their data, whether for highly regulated industries or when building mission-critical blockchain applications. Getting to market quickly, iterating, and attracting new customers are top-of-mind for executives around the world.

Although cloud computing that uses blockchain is a major force in business innovation, challenges are everywhere. Your blockchain in the cloud is only as private and secure as the technology that protects it. As organizations implement modern blockchain application platforms, they are using technologies to deliver cloud-native workloads, provide stateful data services, and deliver enterprise-critical capabilities; from artificial intelligence and messaging to blockchain applications, DevOps, analytics, and high-performance computing.

To this end, the full-stack IBM Blockchain Platform on IBM Cloud Private cloud solution that uses IBM Blockchain Data Store as described in this Blueprint delivers a private cloud fabric for building and managing on-premises, containerized blockchain applications that can deliver scale, performance, security, and data-protection. They also can extend across hybrid and multicloud environments to fill your most critical application requirements. The possibilities are endless, and real-time decision-making is within reach.

Blockchain architectures require flexibility at all component levels. To maximize the business effectiveness of your blockchain network or peer, you want to take advantage of cloud-based and on-premises storage solutions for off-chain data.

With the heightened concerns over data privacy and controls of Personally Identifiable Information, a hybrid multicloud off-chain storage with adequate data management is the more attractive solution. Therefore, any blockchain solution should consider the needs of a hybrid cloud/onsite storage requirement. The possibilities are endless, and real-time decision-making is within reach.

Support for the Blueprint and its configurations

IBM Storage Solutions for Blockchain provides an integrated support experience for clients. The information in this document (referred to throughout as “the Blueprint”) is distributed on an “as is” basis without any warranty that is either expressed or implied. Support for the underlying components that make up this solution are provided by way of the standard procedures and processes that are available for each of those components, as governed by the support entitlement that is available for those components. For more information about these components, see “Prerequisites” on page 4.

Support of IBM Spectrum Connect

Support assistance for the use of this material is limited to situations where IBM Spectrum Connect support is entitled and where the issues are not specific to a Blueprint implementation. Support of the underlying IBM Spectrum Connect components is entitled and provided as an extension of the related Storage hardware and system software. For more information about how to request assistance and support for the IBM Spectrum Connect components, see the hardware and system software documentation.

Requesting assistance

All components of the solutions are part of this unified support structure. Support assistance of the solution that is described in this Blueprint is available by requesting assistance for any of the components in the solution and is the preferred method. Support assistance can also be requested from the IBM Blockchain Platform when the source of the issue cannot be determined.

Scope

This Blueprint provides the following features:

- A solutions architecture and the related storage endpoint capabilities that interact with the following software and hardware components:
  - IBM Cloud Private 3.1.1
  - IBM LinuxONE
  - IBM FlashSystem 9100
  - IBM Storwize V7000
  - IBM Storwize V5000
  - IBM DS8000
  - IBM Cloud Object Storage
  - IBM Spectrum Connect 3.6.1
  - IBM Storage Enabler for Containers
  - IBM Blockchain platform for IBM Cloud Private 1.1.1
- Detailed technical configuration steps for building an end-to-end blockchain peer on IBM private cloud solution with persistent storage

This technical report does *not* include the following features:

- Provide performance analysis or metrics for user consumption
- Replace any official manuals and documents that are issued by IBM for related products
- Explain the installation and configuration of VMware vSphere

What's new in Version 1.2

The documentation for the Blueprint configuration, hardware, and software requirements has been updated for the use of:

- IBM Cloud Object Storage on premises
- IBM Blockchain cloud peer for test and validation

Prerequisites

This Blueprint assumes familiarity with and basic knowledge of the following areas:

- IBM Cloud Private 3.1.1 or later
- IBM Spectrum Connect 3.6.1 or later
- IBM Storage Enabler for Containers Version 2.0 or later
- IBM LinuxONE
- IBM z/VM hypervisor
- IBM FlashSystem 9100, IBM Storwize V7000, IBM Storwize V5000, or IBM DS8000
- Linux-Ubuntu OS
- IBM Cloud Object Storage

Next, we highlight key components of the overall architecture. We suggest that you take the time to familiarize yourself with these components before you start the installation process.

IBM Blockchain Platform

The IBM Blockchain Platform for IBM Cloud Private offering is based on Kubernetes, which allows users to deploy Certificate Authorities (CAs), orderers, and peers on x86, LinuxONE, and IBM Z. IBM Blockchain Platform for IBM Cloud Private is based on Hyperledger Fabric v1.2.1 and is deployed by using Kubernetes Helm charts.

IBM Blockchain Platform for IBM Cloud Private delivers the components that you need to run a blockchain network on your own infrastructure through IBM Cloud Private. The components include Hyperledger Fabric, a Certificate Authority (CA), an orderer, and a peer, which you deploy, manage, and set up by using Kubernetes Helm charts.

This offering is intended for customers with advanced Hyperledger Fabric experience. IBM Blockchain Platform for IBM Cloud Private enables blockchain networks to be deployed on a private cloud to address data residency requirements, market regulations, and infrastructure preference. It simplifies the deployment of essential elements of a blockchain network in your own infrastructure through IBM Cloud Private, which is a Kubernetes-based application platform for developing and managing on-premises, containerized applications. Consider the following points:

- Enables clients to manage IBM Blockchain Platform networks with their own infrastructure. A free Community Edition allows customers to run in their own isolated and secure environments, but no support is provided.
- Enables clients to configure Fabric on Kubernetes by using Helm charts and detailed documentation for operations.
- Entitles clients with advanced technical support, unless you use the Community Edition.

IBM Blockchain Platform for IBM Cloud Private is a bundled product for IBM Cloud Private customers to deploy blockchain components in their local environment. After you import the Helm chart, you can find it as an IBM Blockchain Platform tile in the IBM Cloud Private Catalog.

For more information about IBM Blockchain Platform for IBM Cloud Private, see this IBM Cloud web page:

https://cloud.ibm.com/docs/services/blockchain?topic=blockchain-ibp-icp-about#ibp-icp-about

IBM Blockchain Document Store

The Blockchain Document Store is an IBM provided cloud service that allows secure sharing of documents across multiple participants on a permissioned-blockchain network. It provides an abstraction for handling documents, such as files (text, PDF, JPG, and so on) and JSON that uses APIs.

It also maintains proof of the existence of the documents by using the immutable property of blockchain and supports verification and secure sharing of these documents. The files are securely stored in the IBM Cloud Object Storage layer. The service, which is a series of APIs that is overlaid on the IBM Blockchain Platform infrastructure, demonstrates the use of IBM Cloud Object Storage as an off-chain storage medium.

IBM Cloud Private

Not all application workloads are suitable for the public cloud. In these cases, a private cloud can offer great benefits. A private cloud solution often is chosen for the following reasons:

- Some enterprises cannot tolerate the business disruption of the lengthy refactoring that is often needed to move applications off-premises.
- Many systems of record (traditional database and transactional applications) can include performance characteristics (such as less dynamic resource requirements) that do not benefit as much from cloud economics as do systems of engagement and insight (mobile, social, and analytics applications). Often, these systems feature residency, compliance, or performance needs that require them to run in dedicated on-premises infrastructure.
- Although cloud economics help save money with dynamic applications, applications with steady-state demand can cost more when they are running in public clouds.

For these and many other application workloads that operate best on-premises, IBM Cloud Private offers a leading-edge private cloud platform for developing and running workloads locally. It is an integrated environment that enables you to design, develop, deploy, and manage on-premises, containerized cloud applications behind your firewall.

It also accelerates the work of enterprise developers by providing access to valuable data and applications behind the firewall through a flexible container-based architecture and application programming interface (API)- based catalog of services. It includes a private image repository, management console, and monitoring frameworks.

IBM Cloud Private provides control of how and where applications use cloud services. It uses industry-standard open source technologies, such as Kubernetes, Docker, Helm, Terraform, Cloud Foundry, and more than 40 others.

It also provides integrated operational management and developer services, such as IBM MQ messaging for applications in distributed systems, a microservices framework builder, IBM DB2 Developer Edition, an IBM WebSphere Application Server runtime environment, and more. By using these services, enterprises can optimize older applications with cloud and containers for use with DevOps or analytics, create cloud-native applications, and open their data centers to work with cloud services.

IBM Cloud Private integrates various microservices (such as IBM Watson APIs) and middleware capabilities to help form a robust and responsive infrastructure. These capabilities can improve the overall integration and continued deployment of applications, while minimizing risks that are associated with performance bottlenecks and unpredictable scalability.

IBM Cloud Private helps drive enterprise transformation by providing developers with a choice of languages, frameworks, runtimes, and services to build cloud-native applications and microservices so that they can create their own cloud services. It accelerates innovation by facilitating the use of services, such as blockchain tracking and machine learning that developers can infuse into existing or new applications.

As of release 1.4.0, this Blueprint describes a set of other software packages and middleware support that are currently available as listed in Table 1.

Table 1 Operating systems that are supported by IBM Cloud Private

Vendor	Operating system
Red Hat	Enterprise Linux (RHEL) 7.3, 7.4 and 7.5 (64-bit)
Canonical	Ubuntu 18.04 LTS and 16.04 LTS
SUSE	Linux Enterprise Server (SLES) 12 SP3

IBM Spectrum Connect 3.6.0

Today's organizations demand easy and fast integration of storage in multiple cloud environments. IBM Spectrum Connect empowers storage teams and other stakeholders by enabling provisioning, monitoring, automating, and orchestrating IBM block storage in containerized, VMware, and Microsoft PowerShell environments. It offers the same UI for many solutions and environments for a consistent experience. It also helps organizations simplify cloud complexity and is available by entitlement to every IBM block storage customer.

For more information about the supported IBM Storage Systems and respective microcode levels, see IBM Knowledge Center for IBM Spectrum Connect:

https://www.ibm.com/support/knowledgecenter/en/SS6JWS/landing/IBM_Spectrum_Connect_welcome_page.html

IBM Storage Enabler for Containers

IBM Storage Enabler for Containers allows IBM storage systems to be used as persistent volumes for stateful applications that are running in IBM Cloud Private clusters. IBM Storage Enabler for Containers v2.0 extends IBM Spectrum Connect v3.6 for IBM block storage and IBM Spectrum Scale for file storage, respectively, to Kubernetes-orchestrated container environments. For more information about supported operating systems tables, see IBM Storage Enabler for Containers Release Notes.

IBM Cloud Object Storage

IBM Cloud Object Storage (COS) provides a highly flexible set of architectures that allow for local, metro, and geo sharding architectures to be built. Whether you want a local configuration that provides sharding across local storage, a shard datastore that is configured across a metropolitan or campus area, or a full-fledged geo-sharded data store that guarantees data cannot be lost, IBM COS is the solution of choice.

For more information about IBM Cloud Object Storage, see the following website:

https://www.ibm.com/cloud/object-storage

Blockchain solution reference architecture

The solution that is shown in Figure 1 uses Kubernetes containers on IBM Cloud Private on IBM LinuxONE to provide worker nodes in which to install IBM Blockchain Platform. Storage enabler for containers provides creation, attachment, and mounting of storage to containers through interfacing with Spectrum Connect to communicate to the IBM Block Storage. A standard S3 interface is used to access the provided Cloud Object Storage buckets.

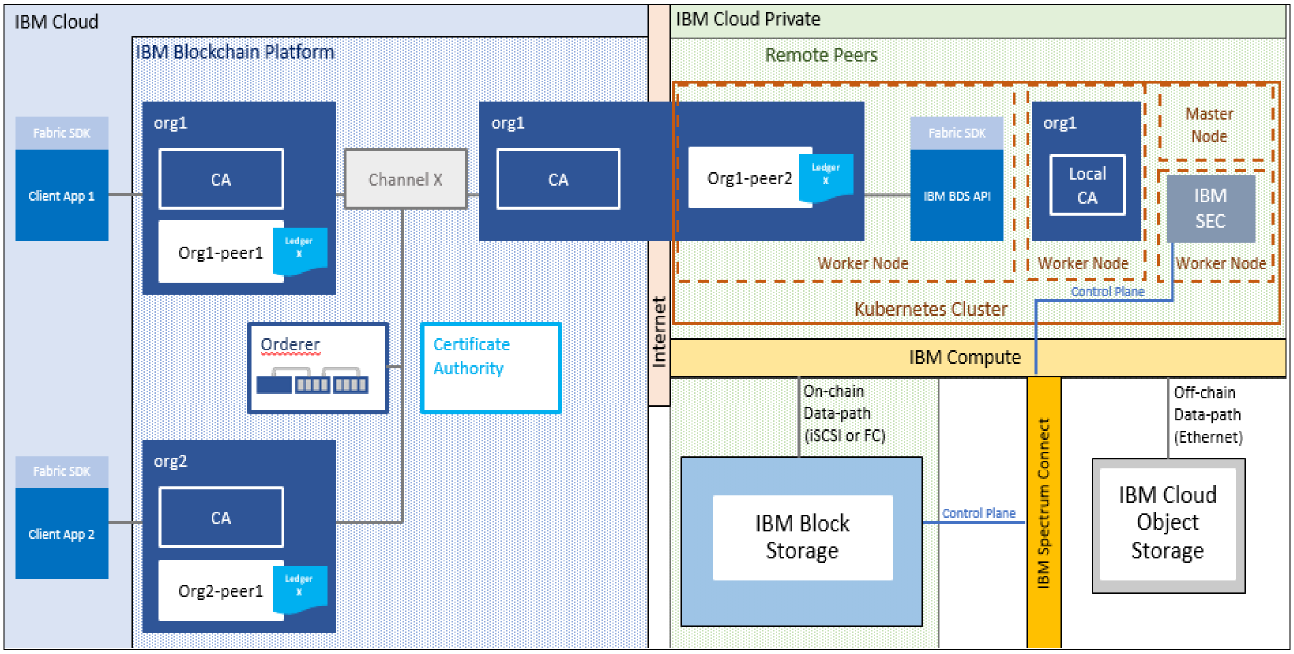


Figure 1 The end-to-end multicloud solution architecture that is illustrated in this Blueprint

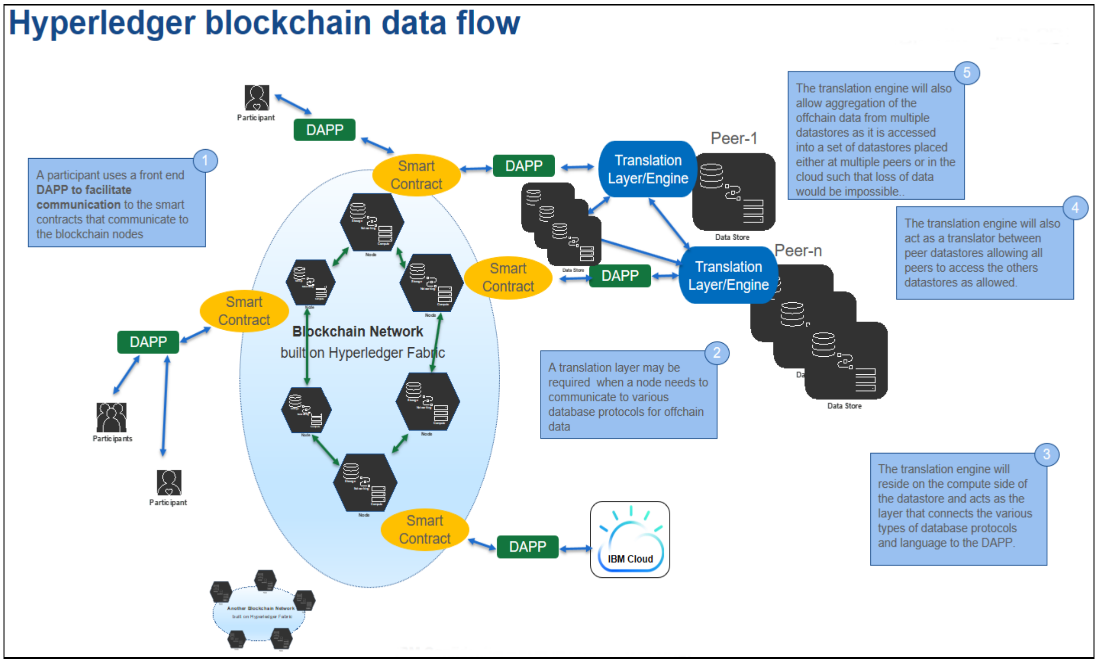
Sample Configuration

The architecture features the following supported components:

- Software:
  - IBM Cloud Private (version 3.1.1)
  - IBM Spectrum Connect 3.6.0
  - IBM Storage Enabler for Containers 2.0
  - IBM Blockchain Platform 1.1.1
  - IBM Blockchain Document Store
- Hardware:
  - IBM LinuxONE Rockhopper II (IBM Compute)
  - IBM FlashSystem 9100 (IBM Block Storage)
  - IBM Cloud Object Storage
- Network:
  - 16 Gbps Fibre Channel
  - 40 GB Ethernet

Solution architecture data paths

The complex nature of data flows in the blockchain environment is shown in Figure 2. At the blockchain internal or on-chain level, data flows from the participant, to a node, and to the consensus nodes. After it is approved, a node combines the transaction with other approved transactions into a block, which is crypto-signed and added to the blockchain that is in the cloud and, if the node is distributed, on local storage at the node.



1 A participant uses a front end **DAPP to facilitate communication** to the smart contracts that communicate to the blockchain nodes

2 A translation layer may be required when a node needs to communicate to various database protocols for offchain data

3 The translation engine will reside on the compute side of the datastore and acts as the layer that connects the various types of database protocols and language to the DAPP.

4 The translation engine will also act as a translator between peer datastores allowing all peers to access the others datastores as allowed.

5 The translation engine will also allow aggregation of the offchain data from multiple datastores as it is accessed into a set of datastores placed either at multiple peers or in the cloud such that loss of data would be impossible..

Figure 2 Data paths of the solution architecture

The off-chain or sidedb dataflow goes from the participant to the node and is held there while the transaction goes through consensus. After the transaction is approved, the off-chain data (which might be block or file, depending on the application), is sent to off-chain storage.

Off-chain storage can be Cloud Object Storage in the cloud or local, or block type storage in a distributed database solution. For local storage, S3 compatible connections are provided or Storage Enablement for Containers is used to provide connection through Spectrum Connect.

Getting started: IBM Storage Solutions for Blockchain Platform on IBM Cloud Private

This section describes the end-to-end private cloud solution architecture to facilitate a smooth deployment experience.

IBM Cloud Private on IBM Z installation

To install IBM Cloud Private on IBM Z, follow the instructions in the most recent version of the document that is available at this web page:

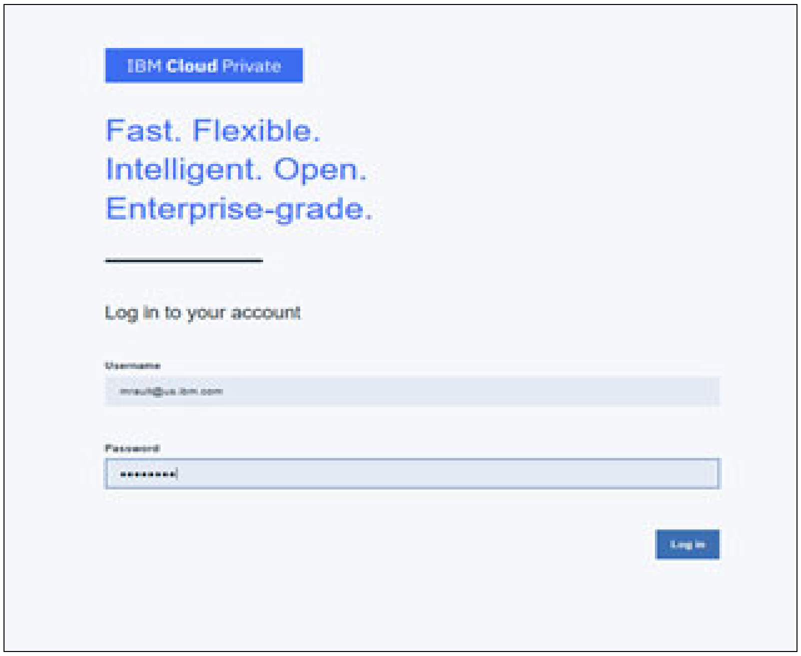
https://www.ibm.com/account/reg/us-en/signup?formid=urx-33814

Configure two worker nodes for the certificate authority (CA) and peer nodes of the IBM Blockchain Platform. If this peer is a stand-alone peer, you might need to configure another orderer node.

IBM Blockchain Platform on IBM Cloud Private installation

Complete the following steps:

1. Start the IBM Cloud Private console as a user with Cluster Administrator privileges. For example, the IBM Cloud Private Dashboard URL for our test installation is shown in Figure 3 (https://x.xx.xx.xx:8443):



Fast. Flexible.
Intelligent. Open.
Enterprise-grade.

Log in to your account

Username: mrault@us.ibm.com

Password: ••••••••

Figure 3 Logging in to IBM Cloud Private

2. From the Console, select **Manage** →**Namespace**.
3. Select the **Create Namespace** option (see Figure 4) to create a namespace to install the IBM Blockchain Platform.

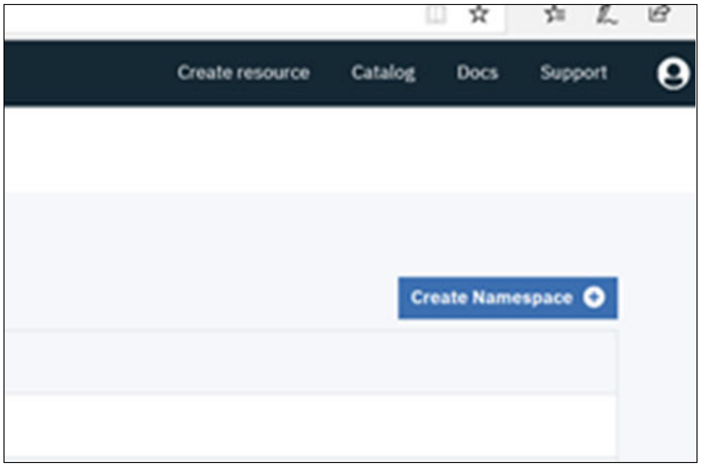
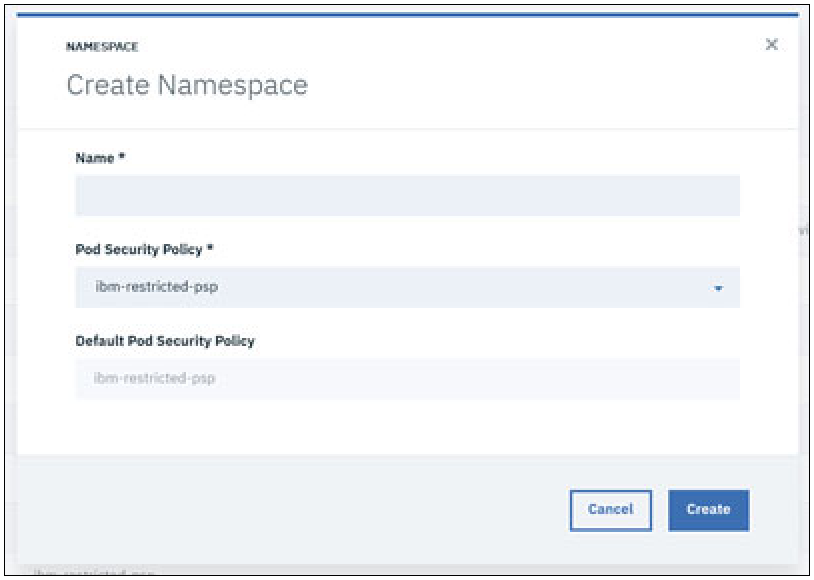


Figure 4 Creating Namespace option

Namespace names must be lowercase. Special characters, such as “-” can be used. The namespace must have ibm-privileged-psp privilege or the package does not install. In our test, we installed a namespace that is called `ibp_on_icp`, as shown in Figure 5.



Name *

Pod Security Policy * ibm-restricted-psp

Default Pod Security Policy ibm-restricted-psp

Figure 5 Creating a Namespace

4. Select the **Catalog** option from the upper menu bar. From the right-side menu, select the **Blockchain** option (see Figure 6).

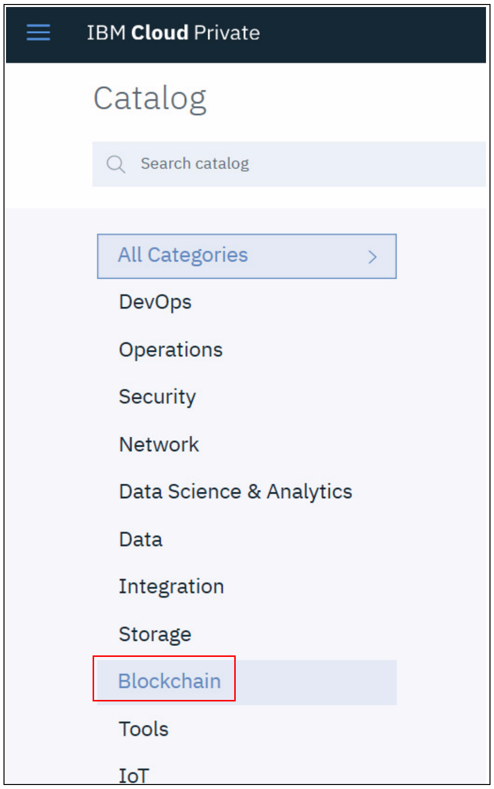


Figure 6 Selecting Blockchain

5. From the Blockchain versions that are provided, select the most recently released remote peer option, as shown in Figure 7.

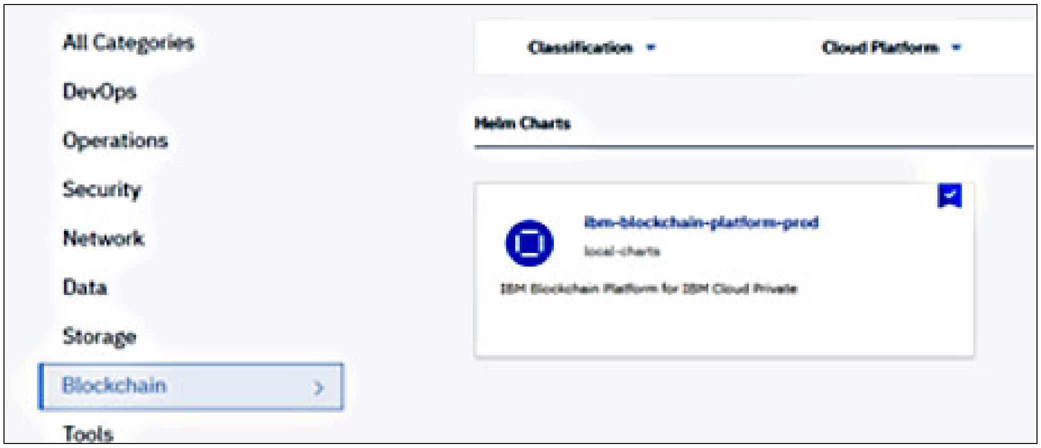
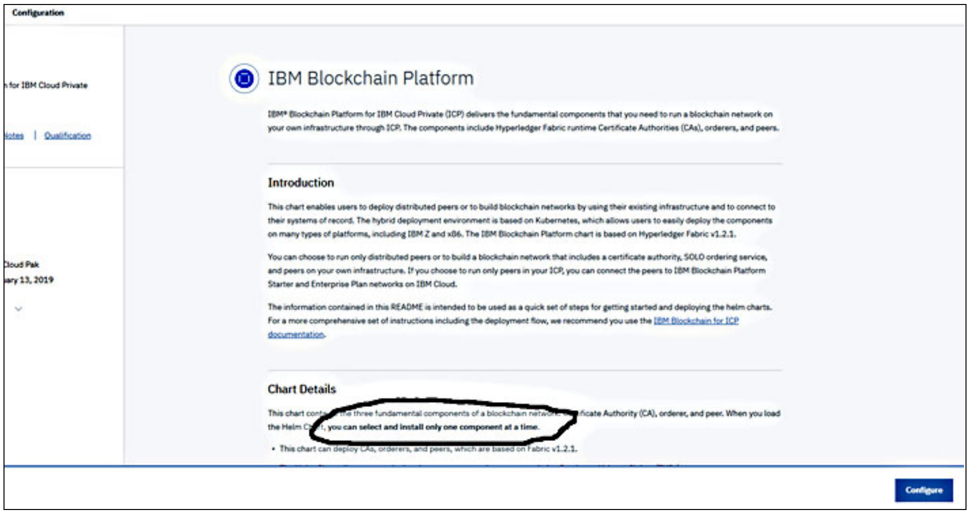


Figure 7 Remote peer option

6. Click the **Configuration** tab at the top of the panel or click **Configure** in the lower right corner (see Figure 8).

> **Note:** You can install only one component at a time. If you plan to build a blockchain network with all of these components, you must install a CA before you install an orderer and a peer. For more information about deploying these components, see the IBM Cloud Docs deployment guide *Getting Started with IBM Blockchain Platform for IBM Cloud Private*.



IBM Blockchain Platform

IBM® Blockchain Platform for IBM Cloud Private (ICP) delivers the fundamental components that you need to run a blockchain network on your own infrastructure through ICP. The components include Hyperledger Fabric runtime Certificate Authorities (CAs), orderers, and peers.

Introduction

This chart enables users to deploy distributed peers or to build blockchain networks by using their existing infrastructure and to connect to their systems of record. The hybrid deployment environment is based on Kubernetes, which allows users to easily deploy the components on many types of platforms, including IBM Z and x86. The IBM Blockchain Platform chart is based on Hyperledger Fabric v1.2.1.

You can choose to run only distributed peers or to build a blockchain network that includes a certificate authority, SOLO ordering service, and peers on your own infrastructure. If you choose to run only peers in your ICP, you can connect the peers to IBM Blockchain Platform Starter and Enterprise Plan networks on IBM Cloud.

The information contained in this README is intended to be used as a quick set of steps for getting started and deploying the helm charts. For a more comprehensive set of instructions including the deployment flow, we recommend you use the IBM Blockchain for ICP documentation.

Chart Details

This chart conta the three fundamental components of a blockchain netwo ficate Authority (CA), orderer, and peer. When you load the Helm C t, **you can select and install only one component at a time.**

- This chart can deploy CAs, orderers, and peers, which are based on Fabric v1.2.1.

Figure 8 IBM Blockchain platform

Configuration

Select the component to install and complete the parameter fields. The tables that are described next list the configuration parameters for each component and their default values.

General and global configuration parameters

Complete the parameter configurations that are listed in Table 2 for either component to install.

Table 2 General and global configuration parameters

Parameter	Description	Default	Required
Helm release name	The name of your helm release.	None	Yes
Target namespace	Choose the Kubernetes namespace to install the Helm Chart.	None	Yes
Service account name	Enter the name of the service account that you use to run the pods.	Default	No

CA configuration parameters

Complete the parameter configurations that are listed in Table 3 for either component to install.

For more information about the CA configuration parameters, see the IBM Blockchain Platforms documentation.

Table 3 CA configuration parameters

Parameter	Description	Default	Required
Install CA	Select to install a CA	Cleared	No
CA name	Specify a name to use for the certificate authority. **Important**: Make a note of this value. It is required later when you configure an orderer or peer.	SampleOrgCA	Yes
CA worker node architecture	Select your cloud platform architecture (ADM64 or S390X).	AMD64	Yes
CA database type	The type of database to store CA data. Only SQLite is supported.	SQLite	Yes
CA data persistence enabled	If checked, data is available when the container restarts. Otherwise, all data is lost if a failover or pod restart occurs.	Selected	No
CA use dynamic provisioning	Check to enable dynamic provisioning for storage volumes.	Selected	No
CA storage class name	Specify a unique storage class name. Otherwise, the default storage class in the cluster is used.	None	No
CA existing volume claim	Specify the name of a Volume Claim and leave all other fields blank.	None	No
CA selector label	Specify the Selector label for your Persistent Volume Claim (PVC).	None	No
CA selector value	Specify the Selector value for your PVC.	None	No
CA storage access mode	Specify the storage access mode for the PVC.	ReadWriteMany	Yes

CA volume claim size	Choose the size of disk to use.	2 Gi	Yes
CA image repository	Location of the CA Helm Chart.	ibmcom/ibp-fabric-ca	Yes
CA Docker image tag	Value of the tag that is associated with the CA image. This field is autofilled to the image version. Do not change it.	1.2.1	Yes
CA service type	This field specifies whether external ports should be exposed on the CA. Select NodePort to expose the ports externally (recommended); select ClusterIP to not expose the ports. LoadBalancer and ExternalName are not supported in this release.	NodePort	Yes
CA secret (Required)	Enter the name of the Kubernetes secret object that you created for your `ca-admin-name` and `ca-admin-password`.	None	Yes
CA CPU request	Specify the minimum number of CPUs to allocate to the CA.		Yes
CA CPU limit	Specify the maximum number of CPUs to allocate to the CA.		Yes
CA memory request	Specify the minimum amount of memory to allocate to the CA.	1 Gi	Yes
CA memory limit	Specify the maximum amount of memory to allocate to the CA.	4 Gi	Yes
CA TLS instance name	Specify a name of the CA TLS instance that is used to enroll an orderer or peer.	None	Yes
CSR common name	Specify the Common Name (CN) that the generated CA root cert presents when contacted.	tlsca-common	Yes
Proxy IP	Enter the Proxy Node IP for the cluster where the CA is deployed.	127.0.0.1	No

Orderer configuration parameters

Complete the parameter configurations that are listed in Table 4 for either component to install.

For more information about the orderer configuration parameters, see the IBM Blockchain Platforms documentation.

Table 4 Orderer configuration parameters

Parameter	Description	Default	Required
Install Orderer	Select to install an orderer.	Cleared	No
Orderer worker node architecture	Select your IBM Cloud Private worker node architecture (AMD64 or S390X).	Autodetected architecture based on your master node	Yes

Orderer configuration	You can customize the configuration of the orderer. This information overwrites the content in the orderer configuration file; that is, `orderer.yaml`. Waiting on more instructions from dev.	None	No
Organization MSP secret (Required)	Specify the name of the secret object that contains organization MSP certificates and keys.	None	No
Orderer data persistence enabled	Data is available when the container restarts. If cleared, all data is lost if a failover or pod restart occurs.	Selected	No
Orderer use dynamic provisioning	Check to enable dynamic provisioning for storage volumes.	Selected	No
Orderer image repository	Location of the Orderer Helm Chart. This field is autofilled to the installed path. If you use the Community Edition and do not have internet access, change this field to the location from where you downloaded the Fabric orderer image.	ibmcom/ibp-fabric-orderer	No
Orderer Docker image tag	Autofilled to the version of the Orderer image.	1.2.1	Yes
Orderer consensus type	The consensus type of the ordering service.	SOLO	Yes
Orderer organization name	Specify the name that you want to use for the orderer organization.	None	No
Orderer Org MSP ID	Specify the name that you want to use for the MSP ID of the orderer organization.	None	No
Orderer storage class name	Specify a storage class name for the orderer.	None	No
Orderer existing volume claim	Specify the name of a Volume Claim and leave all other fields blank.	None	No
Orderer selector label	Specify the Selector label for your PVC.	None	No
Orderer selector value	Specify the Selector value for your PVC.	None	No
Orderer storage access mode	Specify the storage access mode for the PVC.	ReadWriteMany	Yes
Orderer volume claim size	Choose the size of disk to use, which must be at least 2 Gi.	8 Gi	Yes

Orderer service type`	This field specifies whether external ports should be exposed on the orderer. Select NodePort to expose the ports externally (recommended), and ClusterIP to expose the ports on a cluster-internal IP.	NodePort	Yes
Orderer CPU request	Specify the minimum number of CPUs to allocate to the Orderer.		Yes
Orderer CPU limit	Specify the maximum number of CPUs to allocate to the Orderer.		Yes
Orderer memory request	Specify the minimum amount of memory to allocate to the Orderer.	1 Gi	Yes
Orderer memory limit	Specify the maximum amount of memory to allocate to the Orderer.	2 Gi	Yes

Peer configuration parameters

Complete the parameter configuration that is listed in Table 5 for either component to install.

For more information about the peer configuration parameters, see the IBM Blockchain Platform documentation.

Table 5 Peer configuration parameters

Parameter	Description	Default	Required
Install Peer	Select to install a peer.	Cleared	No
Peer worker node architecture	Select your cloud platform architecture (AMD64 or S390x)	AMD64	Yes
Peer image repository	Location of the Peer Helm Chart. This field is autofilled to the installed path.	ibmcom/ibp-fabric-peer	Yes
Peer Docker image tag	Autofilled to the version of the Peer image.	1.2.1	Yes
Peer configuration	You can customize the configuration of the peer. This information overwrites the content in the peer configuration file; that is, `core.yaml`.	None	No
Peer configuration secret (Required)	Name of the Peer configuration secret that you created in IBM Cloud Private.	None	Yes
Organization MSP (Required)	This value can be found in Network Monitor (IBP UI) by clicking Remote Peer Configuration in the Overview window. If you are not connecting to an IBP network, you can create an Organization MSP value, such as “org1” or specify an Organization MSP of which the peer is a part.	None	Yes

Peer service type	Used to specify whether external ports should be exposed on the peer. Select NodePort to expose the ports externally (recommended), and ClusterIP to not expose the ports. LoadBalancer and ExternalName are not supported in this release.	NodePort	Yes
State database	The state database that is used to store your channel ledger. The peer must use the same database as your blockchain network.	None	Yes
CouchDB image repository	Applies only if CouchDB is selected as the ledger database. This field is autofilled to the installed path.	ibmcom/ibp-couchdb	Yes
CouchDB Docker image tag	Applies only if CouchDB is selected as the ledger database. This field is autofilled to the version of the CouchDB image.	0.4.10	Yes
Peer Data persistence enabled	Enable the ability to persist data after cluster restarts or fails (for more information, see storage in Kubernetes). **Note:** If cleared, all data is lost if a failover or pod restart occurs.	Checked	No
Peer use dynamic provisioning	Check to enable dynamic provisioning for storage volumes.	Checked	No
Peer persistent volume claim	For new claim only. Enter a name for your new PVC to be created.	my-data-pvc	No
Peer storage class name	Specify a storage class name for the peer.	Blank if you want to create a new PVC; otherwise, specify the storage class that is associated with the existing PVC.	No
Peer existing volume claim	Specify the name of an existing Volume Claim and leave all other fields blank.	New claim name	No
Peer selector label	Specify the Selector label for your PVC.	Default	No
Peer selector value	Specify the Selector value for your PVC.	Default	No
Peer storage access mode	Specify the storage access mode for the PVC.	ReadWriteMany	No
Peer volume claim size	Size of the Volume Claim. This value must be larger than 2 Gi.	8 Gi	Yes
State database persistent volume claim	For new claim only. Enter a name for your new PVC to be created.	statedb-pvc	No

State database storage class name	Specify a storage class name for state database.	None	No
State database that is in volume claim	Specify the name of an existing Volume Claim and leave all other fields blank.	None	No
State database selector label	Specify the Selector label for your PVC.	None	No
State database selector value	Specify the Selector value for your PVC.	None	No
State database storage access mode	Specify the storage access mode for the PVC.	ReadWriteMany	No
State database volume claim size	Choose the size of disk to use.	8 Gi	Yes
CouchDB - Data persistence enabled	For CouchDB container, ledger data is available when the container restarts. **Note:** If cleared, all data is lost if a failover or pod restart occurs.	Selected	No
CouchDB - Use dynamic provisioning	For CouchDB container use Kubernetes dynamic storage.	Selected	No
Peer CPU request	Minimum number of CPUs to allocate to the peer.		Yes
Peer CPU limit	Maximum number of CPUs to allocate to the peer.		Yes
Peer Memory request	Minimum amount of memory to allocate to the peer.	1 Gi	Yes
Peer Memory limit	Maximum amount of memory to allocate to the peer.	4 Gi	Yes
CouchDB CPU request	Minimum number of CPUs to allocate to CouchDB.		Yes
CouchDB CPU limit	Maximum number of CPUs to allocate to CouchDB.		Yes
CouchDB Memory request	Minimum amount of memory to allocate to CouchDB.	1 Gi	Yes
CouchDB Memory limit	Maximum amount of memory to allocate to CouchDB.		

Select and configure only one type of node at a time: CA, Orderer, or Peer.

If you do not use x86 architecture for all nodes, dynamic provisioning cannot be used. Because we are installing on the IBM z/Architecture, we cannot use dynamic provisioning.

If you do not specify storage class names, the default cluster storage class is used. If you do not use dynamic provisioning, Persistent Volumes must be created and set up with labels that can be used to refine the Kubernetes PVC bind process.

7. Using the values that you gathered in step 6 on page 11, select the Configuration option on the upper menu bar and complete the values that are required for your installation.

   All values that are marked as “required” need an entry or the installation fails.

8. Complete the following steps to install the certificate authority (for more information, see this IBM Developer web page):

   a. Attach the FlashSystem 9100 to the master node and install NFS server. Set up the other nodes in the cluster to use the master node as NFS server to which to mount.

      The following nodes are used in the installation:

```
ordererca_user=ord-ca-admin
namespace=ibp-on-icp
```

   b. Create persistent volumes that use the following names:

```
blockchain-pv01
blockchain-pv02
blockchain-pv03
blockchain-pv04
blockchain-pv05
blockchain-pv06
blockchain-pv07
```

      Your volumes should follow your naming conventions.

   c. Run the following shell commands:

```
root # ===> export ibp4icp_install_dir=$HOME/fabric-ca-client
root # ===> echo "export ibp4icp_install_dir=$HOME/fabric-ca-client" >>
~/.bashrc
root # ===> echo $ibp4icp_install_dir
/root/fabric-ca-client
root # ===> export ordererca_user=ord-ca-admin
root # ===> export ordererca_password=secure_password
(secure_password should be replaced by something actually secure)
```

   d. Log in to the cloud control console:

```
root # ===> cloudctl login -a
https://ibp-icp-blueprint.wsclab.endicott.ibm.com:8443 --skip-ssl-validation
Username> username@xx.ibm.com
Password> ********
Authenticating...
OK
Targeted account ibp-icp-blueprint Account (id-ibp-icp-blueprint-account)
```

   e. Select a namespace:

      i. cert-manager

      ii. default

      iii. ibmcom

      iv. ibp-on-icp

      v. istio-system

      vi. kube-public

vii. kube-system

viii.platform

ix. services

```
Enter a number> 4
Targeted namespace ibp-on-icp
Configuring kubectl ...
Property "clusters.ibp-icp-blueprint" unset.
Property "users.ibp-icp-blueprint-user" unset.
Property "contexts.ibp-icp-blueprint-context" unset.
Cluster "ibp-icp-blueprint" set.
User "ibp-icp-blueprint-user" set.
Context "ibp-icp-blueprint-context" created.
Switched to context "ibp-icp-blueprint-context".
OK
```

f. Configure helm:

```
/root/.helm
OK
```

g. Log in to kubectl:

```
root # ===> kubectl config view --minify | grep namespace
    namespace: ibp-on-icp
```

h. Create a secret for CA:

```
root # ===> kubectl create secret generic ibp4icp-orderer-ca
--from-literal=ca-admin-name=$ordererca_user
--from-literal=ca-admin-password=$ordererca_password
secret/ibp4icp-orderer-ca created
```

i. Get the proxy for the CA node:

```
root # ===> kubectl get nodes -l "proxy=true" -o
jsonpath="{.items[0].status.addresses[0].address}"
9.60.87.24
```

j. Issue the shell commands to configure logicals for:

```
root # ===> export release=ibp4icp-orderer-ca
root # ===> helm get values $release –tls
```

k. Enter the configuration values for ca in the GUI window:

```
  name: OrdererCA
  enabled: true
  proxyIP: 9.60.87.24
  app:
    arch: s390x
  tlsca:
    name: orderer-tlsca
    cname: orderer-tlsca-common
  dataPVC:
    existingClaimName: blockchain-pv01
  ca:
    caAdminSecret: ibp4icp-orderer-ca
license: accept
peer:
  enabled: false
orderer:
  enabled: false
global:
  multiarch: false
```

l. Issue the following root commands:

```
root # ===> export NODE_IP=9.60.87.24
root # ===> export ord_caname=OrdererCA
root # ===> export ord_tlscaname=orderer-tlsca
root # ===> helm status $release --tls
```

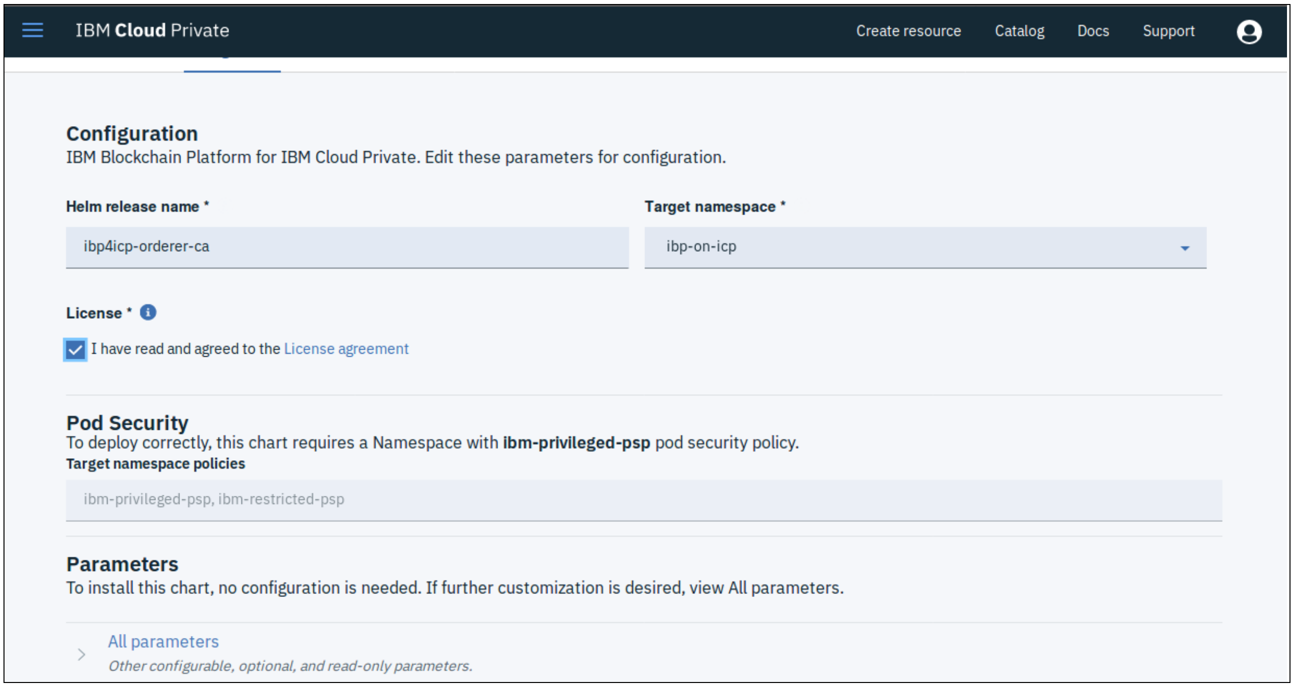
Re-creating as Table

The following values were used for the Blueprint CA:

- Service account name: Default
- CA Name: OrdererCA
- CA Storage class name: Local-storage

  Storage classes are used mostly for dynamic provisioning. Dynamic provisioning is supported under GlusterFS on AMD64/i86_84 only; therefore, it is not a consideration for use on s390x. Although the use of dynapro with NFS works, it is not suggested because it is not supported.
- CA Existing volume claim: ibp4icp-orderer-ca-pvc (mapped to persistent volume blockchain-pv01)
- CA worker node architecture: s390x
- CA Selector label: {leave blank}
- CA selector value: {leave blank}
- CA Secret: Required field. Enter the name of the Kubernetes secret object that you created for your `ca-admin-name` and `ca-admin-password` ibp4icp-orderer-ca.
- CA TLS Instance Name: orderer-tlsca
- CSR Common Name: orderer-tlsca-common
- ProxyIP: 9.60.87.24

The parameters are entered in the configuration windows that are shown in Figure 9, Figure 10, Figure 11 on page 22, and Figure 12 on page 22.



Configuration
IBM Blockchain Platform for IBM Cloud Private. Edit these parameters for configuration.

Helm release name *: ibp4icp-orderer-ca

Target namespace *: ibp-on-icp

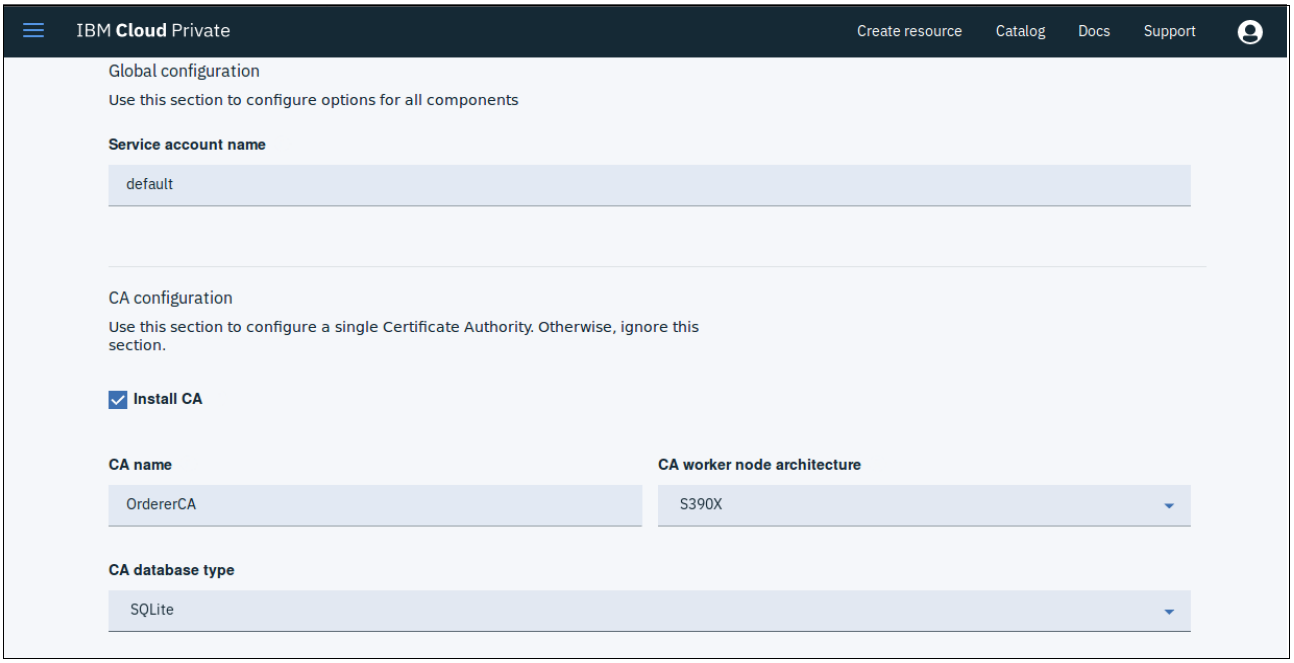
License *
☑ I have read and agreed to the License agreement

Pod Security
To deploy correctly, this chart requires a Namespace with **ibm-privileged-psp** pod security policy.
Target namespace policies: ibm-privileged-psp, ibm-restricted-psp

Parameters
To install this chart, no configuration is needed. If further customization is desired, view All parameters.

All parameters
Other configurable, optional, and read-only parameters.

Figure 9 First Configuration window



Global configuration
Use this section to configure options for all components

Service account name: default

CA configuration
Use this section to configure a single Certificate Authority. Otherwise, ignore this section.

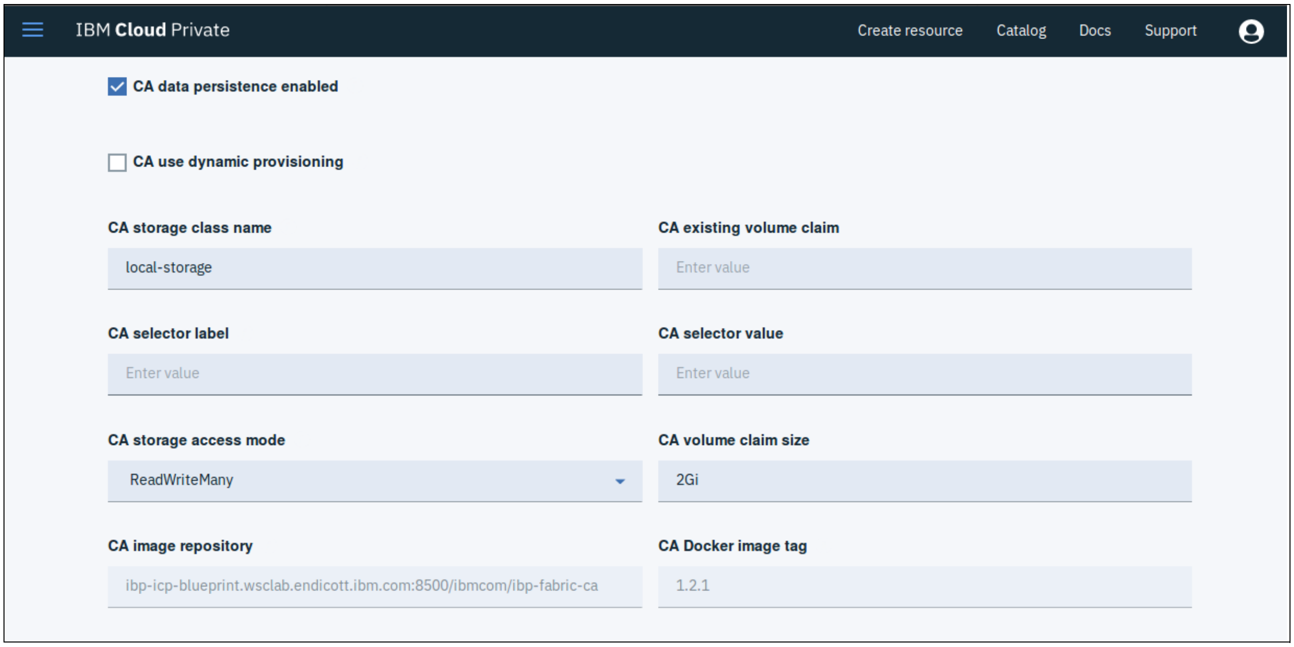
☑ Install CA

CA name: OrdererCA

CA worker node architecture: S390X

CA database type: SQLite

Figure 10 Second Configuration window



☑ CA data persistence enabled

□ CA use dynamic provisioning

CA storage class name: local-storage

CA existing volume claim: Enter value

CA selector label: Enter value

CA selector value: Enter value

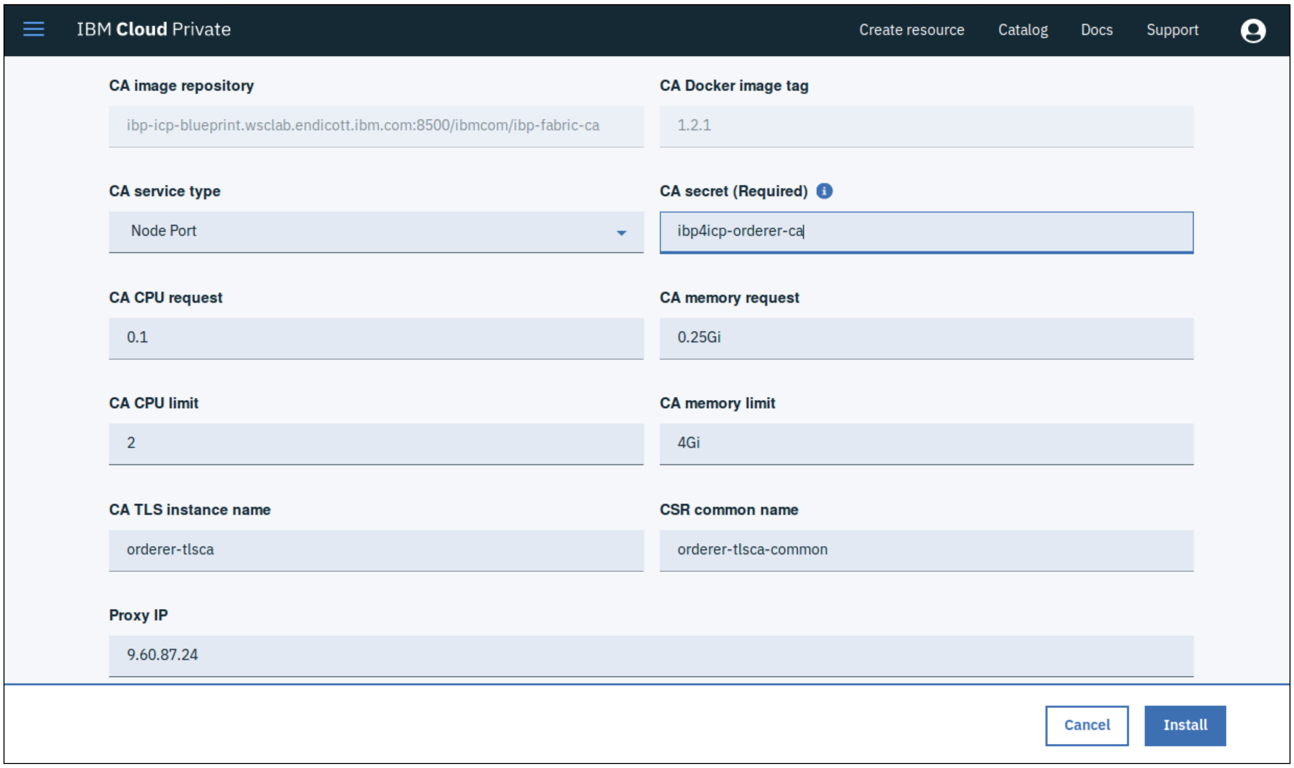
CA storage access mode: ReadWriteMany

CA volume claim size: 2Gi

CA image repository: ibp-icp-blueprint.wsclab.endicott.ibm.com:8500/ibmcom/ibp-fabric-ca

CA Docker image tag: 1.2.1

Figure 11 Third Configuration window



CA image repository: ibp-icp-blueprint.wsclab.endicott.ibm.com:8500/ibmcom/ibp-fabric-ca

CA Docker image tag: 1.2.1

CA service type: Node Port

CA secret (Required): ibp4icp-orderer-ca

CA CPU request: 0.1

CA memory request: 0.25Gi

CA CPU limit: 2

CA memory limit: 4Gi

CA TLS instance name: orderer-tlsca

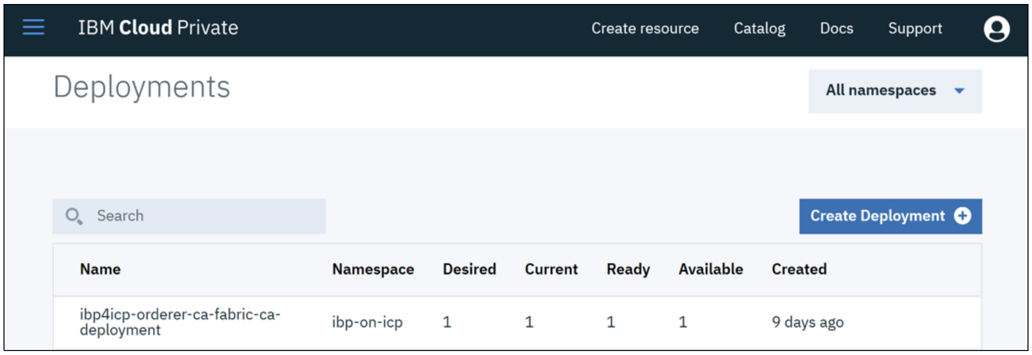
CSR common name: orderer-tlsca-common

Proxy IP: 9.60.87.24

Figure 12 Fourth Configuration window

9. After all of the configuration parameters are entered, click **Install**.

> **Note:** The following error can occur:
>
> ```
> Failed to create default configuration file: open
> /etc/hyperledger/fabric-ca-server/fabric-ca-server-config.yaml: permission
> denied
> ```
>
> This error can be fixed by adding read access to the group in the pv by using the **`chmod -R g+w pv05`** command.
>
> Check that the CA is configured by using your Cloud Control console and review the deployments, as shown in Figure 13.



Name	Namespace	Desired	Current	Ready	Available	Created
ibp4icp-orderer-ca-fabric-ca-deployment	ibp-on-icp	1	1	1	1	9 days ago

Figure 13 Configured certificate authority

10. You can create your peer after the CA is configured.

Creating a peer

Complete the following steps to create a peer:

1. Log in to the cloudctl CLI:

```
[root@ibpzms03 ~]#  cloudctl login -a
https://ibp-icp-blueprint.wsclab.endicott.ibm.com:8443 --skip-ssl-validation

Username> mrault@us.ibm.com
Password> xxxxxxxxx

Authenticating...
OK
Targeted account ibp-icp-blueprint Account (id-ibp-icp-blueprint-account)
```

2. Select the same namespace that is used for the CA (ibp-on-icp):
   - cert-manager
   - default
   - ibm-blockchain-platform
   - ibmcom
   - ibp-on-icp
   - istio-system
   - kube-public

- kube-system
- platform
- services

```
Enter a number> 5
Targeted namespace ibp-on-icp
Configuring kubectl ...
Property "clusters.ibp-icp-blueprint" unset.
Property "users.ibp-icp-blueprint-user" unset.
Property "contexts.ibp-icp-blueprint-context" unset.
Cluster "ibp-icp-blueprint" set.
User "ibp-icp-blueprint-user" set.
Context "ibp-icp-blueprint-context" created.
Switched to context "ibp-icp-blueprint-context".
OK
Configuring helm: /root/.helm
OK
```

3. Verify your name:

```
[root@ibpzms03 ~]# helm ls -m 10 -dr --tls
   next: metering

NAME           REVISION UPDATED              STATUS  CHART
NAMESPACE
ibp4icp-orderer-ca 1       Thu Feb 21 12:24:56
2019DEPLOYEDibm-blockchain-platform-prod-1.0.1 ibp-on-icp
```

4. Move your certification into the `tls.pem` file:

```
root@ibpzms03 ~]# kubectl exec $POD_NAME – cat
/etc/hyperledger/fabric-ca-server/ca-cert.pem > tls.pem && cat
```

5. Create the needed directories and perform needed exports:

```
[root@ibpzms03 ~]# mkdir fabric-ca-client/catls/
[root@ibpzms03 ~]# mkdir fabric-ca-client/ca_admin/
[root@ibpzms03 ~]# export $HOME/fabric-ca-client/ca_admin/
[root@ibpzms03 ~]# export
FABRIC_CA_CLIENT_HOME=$HOME/fabric-ca-client/ca_admin/
```

6. Use kubctl to get your service details; you need the port (in this example, 30722):

```
[root@ibpzms03 ~]# kubectl get service
NAME                   TYPE       CLUSTER-IP   EXTERNAL-IP   PORT(S)
AGE
ibp4icp-orderer-ca-ca   NodePort   10.0.0.162   <none>        7054:30722/TCP
11d
```

7. Create your base64 cert (it must be in base64 or it does not work):

```
[root@ibpzms03 ~]# export POD_NAME=$(kubectl get pods --namespace ibp-on-icp -l
"app=ibm-ibp, release=ibp4icp-orderer-ca" -o
jsonpath="{.items[0].metadata.name}")
[root@ibpzms03 catls]# kubectl exec $POD_NAME -- cat
/etc/hyperledger/fabric-ca-server/ca-cert.pem > tls.pem && cat tls.pem | base64
$FLAG
```

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUNGakNDQWIyZ0F3SUJBZ0lVU1JtMjRlZjRo
dO0rNFBxRWNQaWpGSGRHQ1VBd0NnWUlLb1pJemowRUF3SXcKYURFTE1Ba0dBMVVFQmhNQ1ZWTXhG
ekFWQmd0VkJBZ1REazV2Y25Sb0lFTmhjbTlzYVc1aE1SUXdFZ1lEVlFRSwpFd3RJZVhCbGNteGxa
R2RsY2pFUE1BMEdBMVVFQ3hNR1JtRmljbWxqTVJrd0Z3WURWUVFERXhCbVlXSnlhV01OClkyRXRj

```
MlZ5ZG1WeU1CNFhEVEU1TURJeU56QXhNemt3TUZvWERUTTBNREl5TXpBeE16a3dNRm93YURFTE1B
aOcKQTFVRUJoTUNWVk14RnpBVkJnTlZCQWdURGs1dmNuUm9JRU5oY2O5c2FXNWhNUlF3RWdZRFZR
UUtFd3RJZVhCbApjbXhsWkdkbGNqRVBNQTBHQTFVRUN4TUdSbUZpY21sak1Sa3dGd1lEVlFRREV4
Qm1ZVOp5YVdNdFkyRXRjMlZ5CmRtVnlNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl6ajBEQVFjRFFn
QUVTK3lMcTBzQOE5c3R1cjRUQWlieUtQNzQKSWJJVUdpeC83MWxLRzZKelpUNOxuQkNuZORmREFL
UnFjd2sOaElMYTVKd3VQRWdwUjlhMXJHUi9WVjMzRnFORgpNRU13RGdZRFZSMFBBUUgvQkFRREFn
RUdNQklHQTFVZEV3RUIvd1FJTUFZQkFmOENBUUV3SFFZRFZSME9CQllFCkZBMOhSMkJkTW1kbHdC
UUZmbDVuUkQORWNOZmJNQW9HQONxR1NNNDlCQU1DQTBjQU1FUUNJRW4vYVJ1TkdGVTAKWmh6YzMO
YU8Obkg2ODhCVTdyRONHZjRqdXp4K3dUam1BaUFwSjFqd2F1L1lwTlR6Qk1zRVF5R3A5SDNXVHdL
aQprM2VpUDlqM3RGbO9PUTO9CiOtLSOtRU5EIENFUlRJRklDQVRFLSOtLSOK
```

8. If not yet not done so, download and install the needed fabric binaries:

```
[root@ibpzms03 ~]# mkdir $HOME/fabric-ca-client

[root@ibpzms03 ~]# cd $HOME/fabric-ca-client

[root@ibpzms03 fabric-ca-client]# mkdir fabric-binaries
[root@ibpzms03 fabric-ca-client]# cd fabric-binaries

[root@ibpzms03 fabric-binaries]# curl -sSL http://bit.ly/2ysbOFE | bash -s 1.2.1 
1.2.1 -d -s
```

9. Add the path to the binaries to your $PATH logical so that the binaries are searchable:

```
export PATH=$PATH:$HOME/fabric-ca-client/fabric-binaries/bin
```

10. Enroll the certificate authority. You need the following information:

   ◦ Casecname: The name of your CA secret user: ord-ca-admin (Step 9 c of installing the CA, page 18).

   ◦ Capassword: The password for that user: secure_password (Step 9 c of installing the CA, page 18).

   ◦ CAip: The IP address of the CA node: 9.60.87.24 (Step 9 i of installing the CA, page 19).

   ◦ CAport: The port of the CA: 30722 (Step 6 page 24).

   ◦ Tls.certfiles: The full directory of the tls.pem file (step 4 page 24).

   ◦ Caname: The name of the CA: OrdererCA (Step 9 j of installing the CA, page 19):

```
[root@ibpzms03 ~]# cd $HOME/fabric-ca-client
[root@ibpzms03 fabric-ca-client]# mkdir peer-admin
[root@ibpzms03 fabric-ca-client]# mkdir tls-ibp
[root@ibpzms03 fabric-ca-client]# export 
FABRIC_CA_CLIENT_HOME=$HOME/fabric-ca-client/peer-admin
[root@ibpzms03 fabric-ca-client]# cd fabric-binaries
[root@ibpzms03 fabric-binaries]# fabric-ca-client enroll -u 
https://ord-ca-admin:secure_password@9.60.87.24:30722 --caname OrdererCA 
-tls.certfiles $ibp4icp_install_dir/catls/tls.pem
2019/03/05 11:27:42 [INFO] Created a default configuration file at 
/root/fabric-ca-client/ca-admin/fabric-ca-client-config.yaml
2019/03/05 11:27:42 [INFO] TLS Enabled
2019/03/05 11:27:42 [INFO] generating key: &{A:ecdsa S:256}
2019/03/05 11:27:42 [INFO] encoded CSR
2019/03/05 11:27:43 [INFO] Stored client certificate at 
/root/fabric-ca-client/ca-admin/msp/signcerts/cert.pem
2019/03/05 11:27:43 [INFO] Stored root CA certificate at 
/root/fabric-ca-client/ca-admin/msp/cacerts/9-60-87-24-30722-OrdererCA.pem
```

At this point, the certificate authority is installed and running, you can now deploy a peer.

Deploying the peer

Complete the following steps to deploy the peer:

1. Complete the CA portion of your JSON document by using the following required entries:
   - CAname (Step 9 j of installing the CA, page 19).
   - CAPort (Step 6 of Installing the CA, page 24).
   - CAHost (Step 9 i of installing the CA, page 19).
   - CACert (see next section).
2. Complete the following steps to generate a CACert:

   a. Go to your Starter or Enterprise IBC Console, select **Overview →Connection Profile**, as shown in Figure 14.

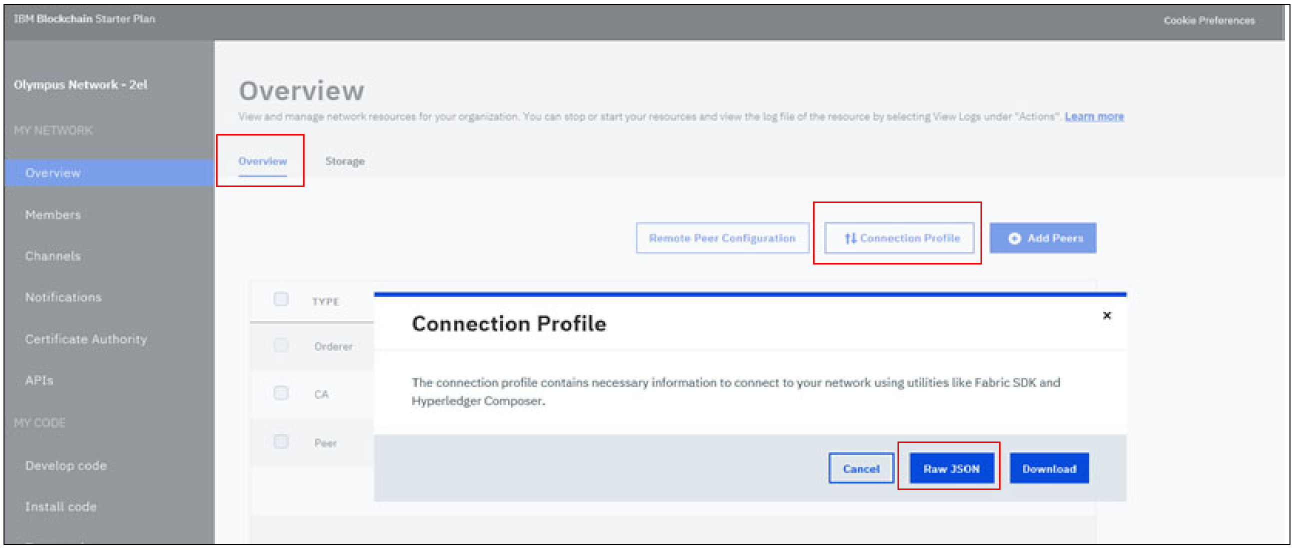

IBM Blockchain Starter Plan
Cookie Preferences
Olympus Network - 2el
MY NETWORK
Overview
Members
Channels
Notifications
Certificate Authority
APIs
MY CODE
Develop code
Install code
Overview
View and manage network resources for your organization. You can stop or start your resources and view the log file of the resource by selecting View Logs under "Actions". Learn more
Overview
Storage
Remote Peer Configuration
Connection Profile
Add Peers
TYPE
Orderer
CA
Peer
Connection Profile
×
The connection profile contains necessary information to connect to your network using utilities like Fabric SDK and Hyperledger Composer.
Cancel
Raw JSON
Download

Figure 14 Connections Profile

   b. Click **Raw JSON** from the JSON connection profile that is displayed. Select the appropriate cert and insert it into the following command:

```
echo -e 'paste in Certificate Authority (CA) TLS Certificate' | base64 $FLAG
```

For example:

```
[root@ibpzms03 dev]# echo -e ?-----BEGIN
CERTIFICATE-----\nMIIFajCCBFKgAwIBAgISA4zRdubZCc/b8B7dxjGDFEOJMA0GCSqGSIb3DQ
EBCwUA\nMEoxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQD\nEx
pMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0xODEyMDIyMTI0NDFaFw0x\nOTAzMDIyMTI0
NDFaMCQxIjAgBgNVBAMMGSoudXMwNy5ibG9ja2NoYWluLmlibS5j\nb20wggEiMA0GCSqGSIb3DQ
EBAQUAA4IBDwAwggEKAoIBAQDiKzoRLmg23pHklFjX\nJIe+J0TqotOUM37zrWGgcbBtSR3sXbLT
N5v4mOAHPtR2AgGEW/BEt57f9tqURS86\n4vUu8s2ANxBXEpo/HsYGfxeiF4gu6T8jit3LKttUEY
8+h5ZJ1uIzU7I371/GSOM5\nxebZ6PSLQE+fbxO8rx6gOsQ+TxIbuTkY/wUPNxiVdEhp9n+M/i7b
1cXweKNOTSac\nwZgB7tSBVDtcnYyUzxNxcEPgxrxyYwccyC9JGg9P9/n+IfqJxUAF2oENfZ36kl
OK\nWfhTbOrpCbEmhexKsxyMSvHMwUOKpWsVInZ1UC9PWJ8sUH8NjRuIxnk+saJCtYTx\nRCynAg
MBAAGjggJuMIICajAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYB\nBQUHAwEGCCsGAQUF
BwMCMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFC65dBQY8kqc\n5fB1e3KHHlRsukBWMB8GA1UdIw
QYMBaAFKhKamMEfd265tE5t6ZFZe/zqOyhMG8G\nCCsGAQUFBwEBBGMwYTAuBggrBgEFBQcwAYYi
aHR0cDovL29jc3AuaW50LXgzLmxl\ndHNlbmNyeXB0Lm9yZzAvBggrBgEFBQcwAoYjaHR0cDovL2
NlcnQuaW50LXgzLmxl\ndHNlbmNyeXB0Lm9yZy8wJAYDVR0RBB0wG4IZKi51czA3LmJsb2NrY2hh
aW4uaWJt\nLmNvbTBMBgNVHSAERTBDMAgGBmeBDAECATA3BgsrBgEEAYLfEwEBATAoMCYGCCsG\n
```

```
AQUFBwIBFhpodHRwOi8vY3BzLmxldHNlbmNyeXB0Lm9yZzCCAQQGCisGAQQB1nkC\nBAIEgfUEgf
IA8AB2AOJpS64m6OlACeiGG7Y7g9Q+5/5OiPukjyiTAZ3d8dv+AAAB\nZ3EGDzMAAAQDAEcwRQIh
ALHD7emLC16lVvYjlP2ZdDxeRdORxpLuOwUyPTtkZwBw\nAiAUGNF+IYinZge/LZzgEnoIcdue6D
BQPHhN3Vsdw7uUEQB2AGPy283oO8wszwty\nhCdXazOkjWF3j711pjixx2hUS9iNAAABZ3EGDZQA
AAQDAEcwRQIgDJcVVQp6omI6\nW2d8udUBHxTjcZdCOTRK23Ux9MRVuGkCIQDCrEcEFpxlNN6zZa
6ZBl8A/ihbyIav\ncn2nIP9Xv1aNOjANBgkqhkiG9w0BAQsFAAOCAQEAJL44fTpUa6qp8CsrgOLA
HeEw\n/LSONVXmbsWxO8NZPkqaEO5Qpvntx+enbvr/ZRqm76ooGQJqgNuYkRJUcpsWMBES\nhOYj
YB+OjWtF7PumohHcANO3rZUGoAONa+vcpgn+oK2ub/SaGno2AkYIdtggejbF\nrjXXaPagKcvaLt
DZ4QwiLkm5HbfahVy1M/oXAsr2uRsL/JU7C5z3dBEmtaNGhoiA\ncbXodLxx8cJSpQeDrt8PSWoC
CPoMOQrOCVJOxUs/qQJNgVZmQ/O1E3EoyXcGFgab\nr3K+fVLjLlVp2NeAT5WToz2NKIuhITABS3
4+JZVOUZ+rNJGX7OhG1eBYBHoWTg==\n-----END CERTIFICATE-----\n-----BEGIN
CERTIFICATE-----\nMIIEkjCCA3qgAwIBAgIQCgFBQgAAAVOFc2oLheynCDANBgkqhkiG9w0BAQ
sFADA/\nMSQwIgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBAMT\nDk
RTVCBSb290IENBIFgzMB4XDTE2MDMxNzE2NDA0NloXDTIxMDMxNzE2NDA0Nlow\nSjELMAkGA1UE
BhMCVVMxFjAUBgNVBAoTDUxldCdzIEVuY3J5cHQxIzAhBgNVBAMT\nGkxldCdzIEVuY3J5cHQgQX
V0aG9yaXR5IFgzMIIBIjANBgkqhkiG9w0BAQEFAAOC\nAQ8AMIIBCgKCAQEAnNMM8FrlLke3cl03
g7NoYzDq1zUmGSXhvb418XCSL7e4S0EF\nq6meNQhY7LEqxGiHC6PjdeTm86dicbp5gWAf15Gan/
PQeGdxyGkOlZHP/uaZ6WA8\nSMx+yk13EiSdRxta67nsHjcAHJyse6cF6s5K671B5TaYucv9bTyW
aN8jKkKQDIZ0\nZ8h/pZq4UmEUEz9l6YKHy9v6Dlb2honzhT+Xhq+w3Brvaw2VFn3EK6BlspkENn
WA\na6xK8xuQSXgvopZPKiAlKQTGdMDQMc2PMTiVFrqoM7hD8bEfwzB/onkxEzOtNvjj\n/PIzar
k5McWvxI0NHWQWM6r6hCm21AvA2H3DkwIDAQABo4IBfTCCAXkwEgYDVR0T\nAQH/BAgwBgEB/wIB
ADAOBgNVHQ8BAf8EBAMCAYYwfwYIKwYBBQUHAQEEczBxMDIG\nCCsGAQUFBzABhiZodHRwOi8vaX
NyZy5OcnVzdGlkLm9jc3AuaWRlbnRydXN0LmNv\nbTA7BggrBgEFBQcwAoYvaHR0cDovL2FwcHMu
aWRlbnRydXN0LmNvbS9yb290cy9k\nc3Ryb290Y2F4My5wN2MwHwYDVR0jBBgwFoAUxKexpHsscf
rb4UuQdf/EFWCFiRAw\nVAYDVR0gBE0wSzAIBgZngQwBAgEwPwYLKwYBBAGC3xMBAQEwMDAuBggr
BgEFBQcC\nARYiaHR0cDovL2Nwcy5yb290LXgxLmxldHNlbmNyeXB0Lm9yZzA8BgNVHR8ENTAz\n
MDGgL6AthitodHRwOi8vY3JsLmlkZW50cnVzdC5jb20vRFNUUk9PVENBWDNDUkwu\nY3JsMB0GA1
UdDgQWBBSoSmpjBH3duubRObemRWXv86jsoTANBgkqhkiG9w0BAQsF\nAAOCAQEA3TPXEfNjWDjd
GBX7CVW+dla5cEilaUcne8IkCJLxWh9KEik3JHRRHGJo\nuM2VcGfl96S8TihRzZvoroed6ti6Wq
EBmtzw3Wodatg+VyOeph4EYpr/1wXKtx8/\nwApIvJSwtmVi4MFU5aMqrSDE6ea73Mj2tcMyo5jM
d6jmeWUHK8so/joWUoHOUgwu\nX4Po1QYz+3dszkDqMp4fklxBwXRsW10KXzPMTZ+sOPAveyxind
mjkW8lGy+QsRlG\nPfZ+G6Z6h7mjem0Y+iWlkYcV4PIWL1iwBi8saCbGS5jN2p8M+X+Q7UNKEkRO
b3N6\nKOqkqm57TH2H3eDJAkSnh6/DNFu0Qg==\n-----END
CERTIFICATE-----\n-----BEGIN
CERTIFICATE-----\nMIIDSjCCAjKgAwIBAgIQRK+wgNajJ7qJMDmGLvhAazANBgkqhkiG9w0BAQ
UFADA/\nMSQwIgYDVQQKExtEaWdpdGFsIFNpZ25hdHVyZSBUcnVzdCBDby4xFzAVBgNVBAMT\nDk
RTVCBSb290IENBIFgzMB4XDTAwMDkzMDIxMTIxOVoXDTIxMDkzMDE0MDExNVow\nPzEkMCIGA1UE
ChMbRGlnaXRhbCBTaWduYXR1cmUgVHJ1c3QgQ28uMRcwFQYDVQQD\nEw5EU1QgUm9vdCBDQSBYMz
CCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB\nAN+v6ZdQCINXtMxiZfaQguzH0yxrMMpb
7NnDfcdAwRgUi+DoM3ZJKuM/IUmTrE4O\nrz5Iy2Xu/NMhD2XSKtkyj4zl93ewEnu1lcCJo6m67X
MuegwGMoOifooUMM0RoOEq\nOLl5CjH9UL2AZd+3UWODyOKIYepLYYHsUmu5ouJLGiifSKOeDNoJ
jj4XLh7dIN9b\nxiqKqy69cK3FCxolkHRyxXtqqzTWMIn/5WgTe1QLyNau7Fqckh49ZLOMxt+/yU
Fw\n7BZy1SbsOFU5Q9D8/RhcQPGX69Wam40dutolucbY38EVAjqr2m7xPi71XAicPNaD\naeQQmx
kqtilX4+U9m5/wAl0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNV\nHQ8BAf8EBAMCAQYw
HQYDVR0OBBYEFMSnsaR7LHH62+FLkHX/xBVghYkQMA0GCSqG\nSIb3DQEBBQUAA4IBAQCjGiybFw
BcqR7uKGY3Or+Dxz9LwwmglSBd49lZRNI+DT69\nikugdB/OEIKcdBodfpga3csTS7MgROSR6cz8
faXbauX+5v3gTt23ADq1cEmv8uXr\nAvHRAosZy5Q6XkjEGB5YGV8eAlrwDPGxrancWYaLbumR9Y
bK+rlmM6pZW87ipxZz\nR8srzJmwN0jP41ZL9c8PDHIyh8bwRLtTcm1D9SZImlJnt1ir/md2cXjb
DaJWFBM5\nJDGFoqgCWjBH4d1QB7wCCZAA62RjYJsWvIjJEubSfZGL+T0yjWW06XyxV3bqxbYo\n
Ob8VZRzI9neWagqNdwvYkQsEjgfbKbYK7p2CNTUQ\n-----END CERTIFICATE-----? |
base64 $FLAG
PyOtLSOtQkVHSU4gQOVSVElGSUNBVEUtLSOtLW5NSUlGYWpDQOJGS2dBdOlCQWdJUOEOelJkdWJa
Q2MvYjhCN2R4akdERkUwSk1BMEdDU3FHUOliMORRRUJDd1VBbk1Fb3hDekFKQmdOVkJBWVRBbFZU
TVJZdOZBWURWUVFLRXcxTVpYUW5jeUJGYm1OeWVYQjBNUO13SVFZRFZRUURuRXhwTVpYUW5jeUJG
Ym1OeWVYQjBJRUYxZEdodmNtbDBlUOJZTXpBZUZ3MHhPREV5TURJeU1USTBOREZhRncweG5PVEF6
```

```
TURJeU1USTBOREZhTUNReElqQWdCZO5WQkFNTUdTb3VkWE13Tnk1aWJHOWphMk5vWVdsdUxtbGli
UzVqbmIyMHdnZOVpTUEwRONTcUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBbOlCQVFEaUt6b1JM
bWcyM3BIa2xGalhuSkllKOowVHFvdDBVTTM3enJXR2djYkJOU1Izc1hiTFRONXYObU9BSFBOUjJB
ZOdFVy9CRXQ1N2Y5dHFVUlM4Nm4OdlV1OHMyQU54QlhFcG8vSHNZR2Z4ZWlGNGd1NlQ4amlOMOxL
dHRVRVk4K2g1WkoxdUl6VTdJMzcxLOdTTO01bnhlYlo2UFNMUUUrZmJ4TzhyeDZnT3NRK1R4SWJ1
VGtZL3dVUE54aVZkRWhwOW4rTS9pN2IxY1h3ZUtOT1RTYWNud1pnQjdOUOJWRHRjbll5VXp4Tnhj
RVBneHJ4eVl3Y2N5QzlKR2c5UDkvbitJZnFKeFVBRjJvRU5mWjM2a2xPS25XZmhUYk9ycENNiRW1o
ZXhLc3h5TVN2SE13VTBLcFdzVkluWjFVQzlQVOo4c1VIOE5qUnVJeG5rK3NhSkNOWVR4blJDeW5B
ZO1CQUFHamdnSnVNSUlDYWpBTOJnTlZIUThCQWY4RUJBTUNCYUF3SFFZRFZSMGxCQll3RkFZSUt3
WUJuQlFVSEF3RUdDQ3NHQVFVRkJ3TUNNQXdHQTFVZEV3RUIvd1FDTUFBdOhRWURWUjBPQkJZRUZD
NjVkQlFZOGtxY241ZkIxZTNLSEhsUnN1aOJXTUI4ROExVWRJd1FZTUJhQUZLaEthbU1FZmQyNjVO
RTVONlpGWmUvenFPeWhNRzhHbkNDcOdBUVVGQndFQkJHTXdZVEF1QmdnckJnRUZCUWN3QVlZaWFI
UjBjRG92TDI5amMzQXVhVzUwTFhnekxteGxuZEhObGJtTnllWEIwTGO5eVp6QXZCZ2dyQmdFRkJR
Y3dBb1lqYUhSMGNEb3ZMMk5sY25RdWFXNTBMWGd6TG14bG5kSE5sYm1OeWVYQjBMbTl5Wnk4dOpB
WURWUjBSQkIwdOcOSVpLaTUxY3pBMOxtSnNiMk5yWTJoaGFXNHVhVOpObkxtTnZiVEJNQmdOVkhT
QUVSVEJETUFnROJtZUJEQUVDQVRBMOJnc3JCZOVFQVlMZkV3RUJBVEFvTUNZRONDcOduQVFVRkJ3
SUJGaHBvZEhSdO9pOHZZMOJ6TG14bGRITmxibU55ZVhCMExtOXlaekNDQVFRRONpcOdBUVFCMW5r
Q25CQUlFZ2ZVRWdmSUE4QUIyQU9KcFM2NGO2T2xBQ2VpROc3WTdnOVErNS81MGlQdWtqeWlUQVoz
ZDhkditBQUFCblozRUdEek1BQUFRREFFY3dSUUloQUxIRDdlbUxDMTZsVnZZamxQMlpkRHhlUmQw
UnhwTHVPd1V5UFROa1p3QnduQWlBVUdORitJWWluWmdlL0xaemdFbm9JY2R1ZTZEQlFSQ0hOM1Zz
ZHc3dVVFUUIyQUdQeTI4M29POHdzendOeW5oQ2RYYXpPa2pXRjNqNzExcGppeHgyaFVTOWlOQUFB
QlozRUdEWlFBQUFRREFFY3dSUUlnREpjVlZRcDZvbUk2blcyZDh1ZFVCSHhUamNaZEMwVFJLMjNV
eDlNUlZ1R2tDSVFEQ3JFYOVGcHhsTk42elphNlpCbDhBL2loYnlJYXZuY24ybklQOVh2MWFOT2pB
TkJna3Foa2lHOXcwQkFRcOZBQU9DQVFFQUpMNDRmVHBVYTZxcDhDc3JnMExBSGVFd24vTFNPTlZY
bWJzV3hPOE5aUGtxYUUwNVFwdm5OeCtlbmJ2ci9aUnFtNzZvbOdRSnFnTnVZa1JKVWNwc1dNQkVT
bmhPWWpZQiswaldORjdQdW1vaEhjQU5PM3JaVUdvQTBOYSt2Y3BnbitvSzJ1Yi9TYUdubzJBallJJ
ZHRnZ2VqYkZucmpYWGFQYWdLY3ZhTHREWjRRd2lMa2O1SGJmYWhWeTFNL29YQXNyMnVScOwvSlU3
QzV6M2RCRW1OYU5HaG9pQW5jYlhvZEx4eDhjSlNwUWVEcnQ4UFNXbONDUG9NT1FyTONWSjB4VXMv
cVFKTmdWWm1RLO8xRTNFb3lYYOdGZ2FibnIzSytmVkxqTGxWcDJOZUFUNVdUb3oyTktJdWhJVEFC
UzMOKOpaVk9VWityTkpHWDdPaEcxZUJZQkhvV1RnPT1uLSOtLS1FTkQgQOVSVElGSUNBVEUtLSOt
LW4tLSOtLUJFROlOIENFUlRJRklDQVRFLSOtLS1uTUlJRWtqQONBM3FnQXdJQkFnSVFDZOZCUWdB
QUFWTOZjMm9MaGV5bkNEQU5CZ2txaGtpRzl3MEJBUXNGQURBL25NU1F3SWdZRFZRUUtFeHRFYVdk
cGRHRnNJRk5wWjIlaGRIVnlaUOJVY25WemRDQkRieTR4RnpBVkJnTlZCQU1UbkRrUlRWQOJTYjI5
MElFTkJJRmd6TUIOWERURTJNRE14TnpFMk5EQTBObG9YRFRJeE1ETXhOekUyTkRBME5sb3duU2pF
TE1BaOdBMVVFQmhNQ1ZWTXhGakFVQmdOVkJBb1REVXhsZENNkeklFVnVZMOo1YOhReEl6QWhCZO5W
QkFNVG5Ha3hsZENNkeklFVnVZMOo1YOhRZ1FYVjBhRzl5YVhSNUlGZ3pNSUlCSWpBTkJna3Foa2lH
OXcwQkFRRUZBQU9DbkFROEFNSUlCQ2dLQOFRRUFuTk1NOEZybExrZTNNjbDAzZzdOb1l6RHExelVt
R1NYaHZiNDE4WENTTDdlNFMwRUZucTZtZU5RaFk3TEVxeEdpSEM2UGpkZVRtODZkaWNicDVnVOFm
MTVHYW4vUFFlR2R4eUdrT2xaSFAvdWFNldBOG5TTXgreWsxMOVpU2RSSeHRhNjducOhqYOFISnlz
ZTZjRjZzNUs2NzFCNVRhWXVjdjliVHlXYU44aktrS1FFSVowbHo4aC9wWnEOVW1FVUV6OWw2WUtI
eTl2NkRsYjJob256aFQrWGhxK3czQnJ2YXcyVkZuMOVLNkJsc3BrRU5uVOFuYTZ4Szh4dVFTWGd2
b3BaUEtpQWxLUVRHZE1EUU1jMlBNVGlWRnJxb0O3aEQ4YkVmd3pCL29ua3hFejBOTnZqam4vUEl6
YXJrNU1jV3Z4STBOSFdRVOO2cjZoQ2OyMUF2QTJIMORrdOlEQVFBQm8OSUJmVENDQVhrdOVnWURW
UjBUbkFRSC9CQWd3QmdFQi93SUJBREFPQmdOVkhROEJBZjhFQkFNQOFZWXdmd1lJS3dZQkJRVUhB
UUVFY3pCeE1ESUduQONzROFRVUZCekFCaGlab2RIUndPaTh2YVhOeVp5NTBjblZzZEdsaOxtOWpj
MOF1YVdSbGJuUnlkWE4wTG1Odm5iVEE3QmdnckJnRUZCUWN3QW9ZdmFIUjBjRG92TDJGd2NITXVh
V1JsYm5SeWRYTjBMbU52YlM5eWIyOTBjeTlrYmMzUnliaMjkwWTJGNE15NXdOMk13SHdZRFZSMGpC
Qmd3Rm9BVXhhLZXhwSHNzY2ZyYjRVdVFkZi9FRldDRmlSQXduVkFZRFZSMGdCRTB3U3pBSUJnWm5n
UXdCQWdFd1B3WUxLd1lCQkFHQzN4TUJBUUV3TURBdUJnZ3JCZ0VGQlFjQ0FSWWlhSFIwY0RvdkwyTm
Mk53Y3k1eWIyOTBMWGd4TG14bGRITmxibU55ZVhCMExtOXlaekE4QmdOVkhSOEVOVEF6bk1ER2dM
NkFOaGlOb2RIUndPaTh2WTNKcOxtbGtaVzUwY25WemRDNWpiMjB2UkZOVVVVrOVBWRU5CVORORFVr
d3VuWTNKcO1CMEdBMVVkRGdRV0JCU29TbXBqQkgzZHV1YlJPYmVtUldYdjg2anNvVEFOQmdrcWhr
aUc5dzBCQVFzRm5BQU9DQVFFQTNUUFhFZk5qV0RqQ1dDdDVlcrZGxhNWNFaWxhVWNuZThJaONK
THhXaDlLRWlrMOpIUlJIROpvbnVNMlZjR2ZsOTZTOFRpaFJ6WnZvcm9lZDZOaTZXcUVCbXR6dzNX
b2RhdGcrVnlPZXBoNEVZcHIvMXdYS3R4OC9udOFwSXZKU3dObVZpNE1GVTVhTXFyUORFNmVhNzNN
```

```
ajJOY015bzVqTWQ2am1lV1VISzhzby9qb1dVbOhPVWd3dW5YNFBvMVFZeiszZHN6aORxTXAOZmts
eEJ3WFJzVzEwS1h6UE1UWitzT1BBdmV5eGluZG1qa1c4bEd5K1FzUmxHblBmWitHNlo2aDdtamVt
MFkraVdsa1ljVjRQSVdMMWl3Qmk4c2FDYkdTNWpOMnA4TStYK1E3VU5LRWtST2IzTjZuSO9xa3Ft
NTdUSDJIM2VESkFrU25oNi9ETkZ1MFFnPT1uLSOtLS1FTkQgQOVSVElGSUNBVEUtLSOtLW4tLSOt
LUJFROlOIENFUlRJRklDQVRFLSOtLS1uTUlJRFNqQONBaktnQXdJQkFnSVFSSyt3ZO5hako3cUpN
RG1HTHZoQWF6QU5CZ2txaGtpRzl3MEJBUVVGQURBL25NU1F3SWdZRFZRUUtFeHRFYVdkcGRHRnNJ
Rk5wWjI1aGRIVnlaUOJVY25WemRDQkRieTR4RnpBVkJnTlZCQU1UbkRrUlRWQOJTYjI5MElFTkJJ
Rmd6TUIOWERUQXdNRGt6TURJeE1USXhPVm9YRFRJeE1Ea3pNREUwTURFeE5Wb3duUHpFaO1DSUdB
MVVFQ2hNYlJHbG5hWFJoYkNCVGFXZHVZWFIxY21VZ1ZISjFjM1FnUTI4dU1SY3dGUVlEVlFRRG5F
dzVFVTFRZ1VtOXZkQOJEUVNCWU16QONBUOl3RFFZSktvWklodmNOQVFFQkJRQURnZOVQQURDQOFR
bONnZOVCbkFOK3Y2WmRRQOlOWHRNeGlaZmFRZ3V6SDB5eHJNTXBiNO5uRGZjZEF3UmdVaStEbOOz
WkpLdUOvSVVtVHJFNE9ucno1SXkyWHUvTk1oRDJYUOtOa3lqNHpsOTNldOVudTFsYONKbzZtNjdY
TXVlZ3dHTW9PaWZvb1VNTTBSbO9FcW5PTGw1Q2pIOVVMMkFaZCszVVdPRHlPSOlZZXBMWVlIc1Vt
dTVvdUpMR2lpZlNLT2VETm9KamoOWExoN2RJTjlibnhpcUtxeTY5YOszRkN4b2xrSFJ5eFhOcXF6
VFdNSW4vNVdnVGUxUUx5TmF1NOZxY2toNDlaTE9NeHQrL3lVRnduNOJaeTFTYnNPRlU1UTlEOC9S
aGNRUEdYNjlXYWOOMGR1dG9sdWNiWTM4RVZBanFyMmO3eFBpNzFYQWljUE5hRG5hZVFRbXhrcXRp
bFgOK1U5bTUvdOFsMENBdOVBQWFOQO1FQXdEd1lEVlIwVEFRSC9CQVV3QXdFQi96QU9CZO5WbkhR
OEJBZjhFQkFNQOFRWXdIUVlEVlIwTOJCWUVGTVNuc2FSNOxISDYyKOZMaOhYL3hCVmdoWWtRTUEw
RONTcUduUOliMORRRUJCUVVBQTRJQkFRQ2pHaXliRndCY3FSN3VLR1kzT3IrRHh6OUx3d21nbFNC
ZDQ5bFpSTkkrRFQ2OW5pa3VnZEIvTOVJS2NkQm9kZnBnYTNjc1RTNO1nUk9TUjZjejhmYVhiYXVY
KzV2M2dUdDIzQURxMWNFbXY4dVhybkF2SFJBb3NaeTVRNlhrakVHQjVZR1Y4ZUFscndEUEd4cmFu
Y1dZYUxidW1SOVliSytybG1NNnBaVzg3aXB4WnpuUjhzcnpKbXdOMGpQNDFaTDljOFBESEl5aDhi
d1JMdFRjbTFEOVNaSW1sSm5OMWlyL21kMmNYamJEYUpXRkJNNW5KREdGb3FnQ1dqQkgOZDFRQjd3
QONaQUE2MlJqWUpzV3ZJakpFdWJTZlpHTCtUMHlqV1cwNlh5eFYzYnF4YllvbK9iOFZaUnpJOW5l
V2FncU5kd3ZZa1FzRWpnZmJLYllLN3AyQO5UVVFuLSOtLS1FTkQgQOVSVElGSUNBVEUtLSOtLT8K
```

The bolded part is the new Base64 certificate to insert into the JSON connection profile.

The JSON connection profile should now look like the following example:

```
{
    "enrollment": {
        "component": {
            "cahost": "9.60.87.24",
            "caport": "30722",
            "caname": "OrdererCA",
            "catls": {
                "cacert":
"PyOtLSOtQkVHSU4gQOVSVElGSUNBVEUtLSOtLW5NSUlGYWpDQOJGS2dBdOlCQWdJUOEOelJkdWJa
Q2MvYjhCN2R4akdERkUwSk1BMEdDU3FHUOliMORRRUJDd1VBbk1Fb3hDekFKQmdOVkJBWVRBbFZU
TVJZdOZBWURWUVFLRXcxTVpYUW5jeUJGYm1OeWVYQjBNUO13SVFZRFZRUURuRXhwTVpYUW5jeUJG
Ym1OeWVYQjBJRUYxZEdodmNtbDBlUOJZTXpBZUZ3MHhPREV5TURJeU1USTBOREZhRncweG5PVEF6
TURJeU1USTBOREZhTUNReElqQWdCZO5WQkFNTUdTb3VkWE13Tnk1aWJHOWphMk5vWVdsdUxtbGli
UzVqbmIyMHdnZOVpTUEwRONTcUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBbOlCQVFEaUt6b1JM
bWcyM3BIa2xGalhuSkllKOowVHFvdDBVTTM3enJXR2djYkJOU1Izc1hiTFRONXYObU9BSFBOUjJB
ZOdFVy9CRXQ1N2Y5dHFVUlM4Nm4OdlV1OHMyQU54QlhFcG8vSHNZR2Z4ZWlGNGd1NlQ4amlOMOxL
dHRVRVk4K2g1WkoxdUl6VTdJMzcxLOdTTOO1bnhlYlo2UFNMUUUrZmJ4TzhyeDZnT3NRK1R4SWJ1
VGtZL3dVUE54aVZkRWhwOW4rTS9pN2IxY1h3ZUtOT1RTYWNud1pnQjdOUOJWRHRjbll5VXp4Tnhj
RVBneHJ4eVl3Y2N5QzlKR2c5UDkvbitJZnFKeFVBRjJvRU5mWjM2a2xPS25XZmhUYk9ycENNiRW1o
ZXhLc3h5TVN2SE13VTBLcFdzVkluWjFVQzlQVOo4c1VIOE5qUnVJeG5rK3NhSkNOWVR4blJDeW5B
ZO1CQUFHamdnSnVNSUlDYWpBTOJnTlZIUThCQWY4RUJBTUNCYUF3SFFZRFZSMGxCQll3RkFZSUt3
WUJuQlFVSEF3RUdDQ3NHQVFVRkJ3TUNNQXdHQTFVZEV3RUIvd1FDTUFBdOhRWURWUjBPQkJZRUZD
NjVkQlFZOGtxY241ZkIxZTNLSEhsUnN1aOJXTUI4ROExVWRJd1FZTUJhQUZLaEthbU1FZmQyNjVO
RTVONlpGWmUvenFPeWhNRzhHbkNDcOdBUVVGQndFQkJHTXdZVEF1QmdnckJnRUZCUWN3QVlZaWFI
UjBjRG92TDI5amMzQXVhVzUwTFhnekxteGuZEhObGJtTnllWEIwTG05eVp6QXZCZ2dyQmdFRkJR
Y3dBb1lqYUhSMGNEb3ZMMk5sY25RdWFXNTBMWGd6TG14bG5kSE5sYm1OeWVYQjBMbTl5Wnk4dOpB
WURWUjBSQkIwdOcOSVpLaTUxY3pBMOxtSnNiMk5yWTJoaGFXNHVhVOpObkxtTnZiVEJNQmdOVkhT
```

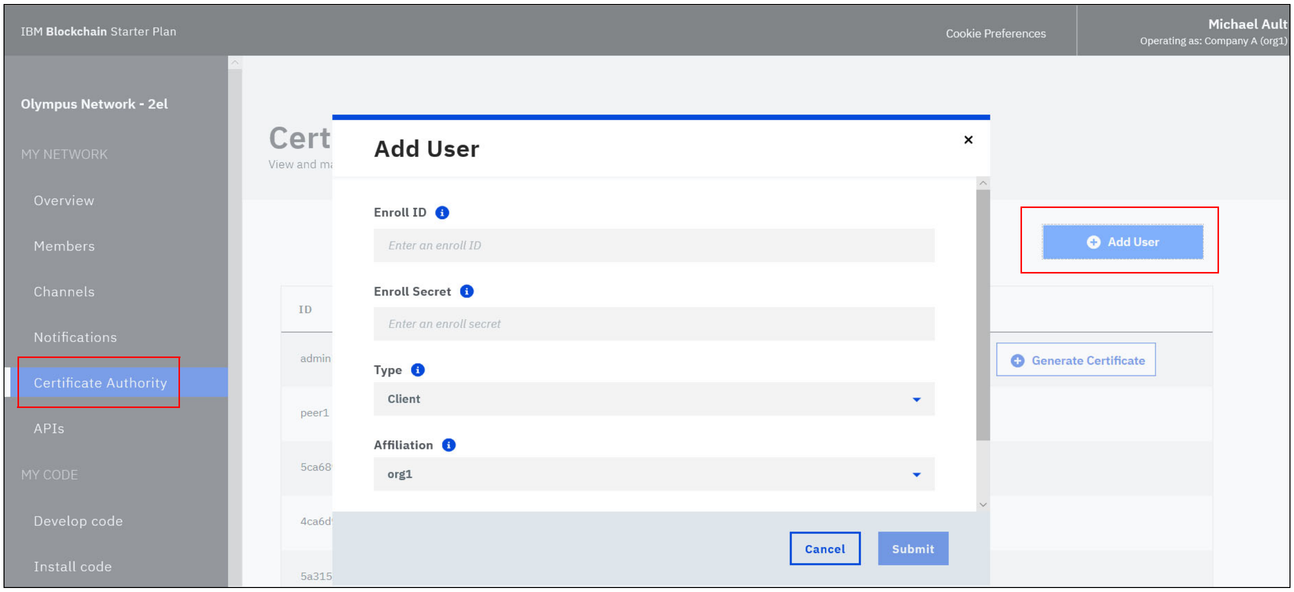
```
QUVSVEJETUFnR0JtZUJEQUVDQVRBM0Jnc3JCZ0VFQVlMZkV3RUJBVEFvTUNZR0NDc0duQVFVRkJ3
SUJGaHBvZEhSd09pOHZZM0J6TG14bGRITmxibU55ZVhCMExtOXlaekNDQVFRR0NpcOdBUVFCMW5r
Q25CQUlFZ2ZVRWdmSUE4QUIyQU9KcFM2NGO2T2xBQ2VpROc3WTdnOVErNS81MGlQdWtqeWlUQVoz
ZDhkditBQUFCblozRUdEek1BQUFRREFFY3dSUUloQUxIRDdlbUxDMTZsVnZZamxQMlpkRHhlUmQw
UnhwTHVPd1V5UFROa1p3QnduQWlBVUdORitJWWluWmdlL0xaemdFbm9JY2R1ZTZEQlFQSGhOM1Zz
ZHc3dVVFUUIyQUdQeTI4M29POHdzendOeW5oQ2RYYXpPa2pXRjNqNzExcGppeHgyaFVTOWlOQUFB
QlozRUdEWlFBQUFRREFFY3dSUUlnREpjVlZRcDZvbUk2blcyZDh1ZFVCSHhUamNaZEMwVFJLMjNV
eDlNUlZ1R2tDSVFEQ3JFY0VGcHhsTk42elphNlpCbDhBL2loYnlJYXZuY24ybklQOVh2MWFOT2pB
TkJna3Foa2lHOXcwQkFRc0ZBQU9DQVFFQUpMNDRmVHBBVYTZxcDhDc3JnMExxBSGVFd24vTFNPTlZY
bWJzV3hPOE5aUGtxYUUwNVFwdm5OeCtlbmJ2ci9aUnFtNzZvb0dRSnFnTnVZa1JKVWNwc1dNQkVT
bmhPWWpZQiswaldORjdQdW1vaEhjQU5PM0JaVUdvQTBOYSt2Y3BnbitvSzJ1Yi9TYUdubzJBa1lJ
ZHRnZ2VqYkZucmpYWGFQYWdLY3ZhTHREWjRRd2lMa2O1SGJmYWhWeTFNL29YQXNyMnVScOwvSlU3
QzV6M2RCRW1OYU5HaG9pQW5jYlhvZEx4eEhjSlNwUWVEcnQ4UFNXbONDUG9NT1FyTONWSjB4VXMv
cVFKTmdWWm1RLO8xRTNFb3lYYOdGZ2FibnIzSytmVkxqTGxWcDJOZUFUNVdUb3oyTktJdWhJVEFC
UzMOKOpaVk9VWityTkpHWDdPaEcxZUJZQkhvV1RnPT1uLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0t
LW4tLS0tLUJFR0lOIENFUlRJRklDQVRFLS0tLS1uTUlJRWtqQ0NBM3FnQXdJQkFnSVFDZOZCUWdB
QUFWTOZjMm9MaGV5bkNEQU5CZ2txaGtpRzl3MEJBUXNGQURBL25NU1F3SWdZRFZRUUtFeHRFYVdk
cGRHRnNJRk5wWjI1aGRIVnlaUOJVY25WemRDQkRieTR4RnpBVkJnTlZCQU1UbkRrUlRWQ0JTYjI5
MElFTkJJRmd6TUlOWERURTJNRE14TnpFMk5EQTBObG9YRFRJeE1ETXhOekUyTkRBME5sb3duU2pF
TE1BaOdBMVVFQmhNQ1ZWTXhGakFVQmdOVkJBb1REVXhsZENkekJFVnVZM0oo1YOhReEl6QWhCZO5W
QkFNVG5Ha3hsZENkekJFVnVZM0oo1YOhRZ1FYVjBhRzl5YVhSNUlGZ3pNSUlCSWpBTkJna3Foa2lH
OXcwQkFRRUZBQU9DbkFROEFNSUlCQ2dLQ0FRRUFuTk1NOEZybExrZTNNjbDAzZzdOb1l6RHExelVt
R1NYaHZiNDE4WENTTDdlNFMwRUZucTZtU05TZU5raFk3TEVxeEdpSEM2UGpkZVRtODZkaWNDVnVOOFm
MTVHYW4vUFFlR2R4eUdrT2xaSFAvdWFNldBOG5TTXgreWsxMOVpU2RSSeHRhNjducUhOY0FIS
ZTZjRjZzNUs2NzFCNVRhWXVjdjliVHlXYU44aktrS1FESVowbloo4aC9wWnEOVW1FVUV6OWw2WUtI
eTl2NkRsYjJob256aFQrWGhxK3czczQnJ2YXcyVkZuMOVLNkJsc3BrRU5uVOFuYTZ4Szh4dVFTWGd2
b3BaUEtpQWxLUVRHZE1EUU1jMlBNVGlWRnJxb0O3aEQ4YkVmd3pCL29ua3hFejBOTnZqam4vUEl6
YXJrNU1jV3Z4STBOSFdRVO2cjZoQ2OyMUF2QTJIMORrdO1EQVFBQm8OSUJmVENDQVhrd0VnWURW
UjBUbkFRSC9CQWd3QmdFQi93SUJBREFPQmdOVkhROEJBZjhFQkFNQ0FZWXdmd1lJS3dZQkJRVUhB
UUVFY3pCeE1ESUduQ0NzRVFRJBDsBvtNTTOZdP8DDPgeNWSpxafnXZIEDKM2gLnzNOtJVrMzQzAbYL
MOF1YVdSbGJuUnlkWE4wTG1Odm5iVEE3QmdnckJnRUZCUWN3QW9ZdmFIUjBjRG92TDJGd2NITXVh
V1JsYm5SeWRYTjBMbU52YlM5eWIyOTBjeTlrYmMzUnliMjkwWTJFek15NXdOMk13SHdZRFZSMGpC
Qmd3Rm9BVXhLZXhwSHNzY2ZyYjRVdVFkZi9FRldDRmlSQXdWQVlEVlIwZ0JFMHdTekFJQmdabmdR
UXdCQWdDMB3WUxLd1lCQkFHQzN4TUJBUUV3TURBdUJnZ3JCZ0VGQlFjQ0FSWWlhSFIwY0RvdkwzQnBZ
Mk53Y3k1eWIyOTBMWGd4TG14bGRITmxibU55ZVhCMExtOXlaekE4QmdOVkhSOEVOVEF6bk1ER2dM
NkF0aGlOb2RIUndPaTh2WTNKc0xtbGtaVzUwY25WemRDNWpiMjB2UkZOVVVrOVBWRU5CV1JU5CVOROFVr
d3VuWTNKc01CMEdBMVVkRGdRV0JCU29TbXBqQkgzZHV1YlJPYmVtUldYdjg2anNvVEFOQmdrcWho
aUc5dzBCQVFzRm5BQU9DQVFFQTNUUFhFZk5qV0RqQ0Vlckd4aFVWNFaWxhVWNuZThJaONK
THhXaDlLRWlrMOpIUlJIROpvbnVNMlZjR2ZsOTZTOFRpaFJ6WnZvcm9lZDZOaTZXcUVCbXR6dzNX
b2RhdGcrVnlPZXBoNEVZcHIvMXdYS3R4OC9udOFwSXZKU3dObVZpNE1GVTVhTXFyUORFNmVhNzNN
ajJOYO15bzVqTWQ2am1lV1VISzhzby9qb1dVbOhPVWd3dW5YNFBvMVFZeiszZHN6aORxTXAOZmts
eEJ3WFJzVzEwS1h6UE1UWitzT1BBdmV5eGluZG1qa1c4bEd5K1FzUmxHblBmWitHNlo2aDdtamVt
MFkraVdsa1ljVjRQSVdMMWl3Qmk4c2FDYkdTNWpOMnA4TStYK1E3VU5LRWtST2IzTjZuS09xa3Ft
NTdUSDJIM2VESkFrU25oNi9ETkZ1MFFnPT1uLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLW4tLS0t
LUJFR0lOIENFUlRJRklDQVRFLS0tLS1uTUlJRFNqQ0NBaktnQXdJQkFnSVFSSyt3Z05haako3cUpN
RG1HTHZZoQWF6QU5CZ2txaGtpRzl3MEJBUVVGQURBL25NU1F3SWdZRFZRUUtFeHRFYVdkcGRHRnNJ
Rk5wWjI1aGRIVnlaUOJVY25WemRDQkRieTR4RnpBVkJnTlZCQU1UbkRrUlRWQ0JTYjI5MElFTkJJ
Rmd6TUlOWERUQXdNRGt6TURJeE1USXhPVm9YRFRJeE1Ea3pNREUwTURFeE5Wb3duUHpFa01DSUdB
MVVFQ2hNYlJHbG5hWFJoYkNCVGFXZHVZWFIxY21VZ1ZISjFjM1FnUTI4dU1SY3dGUVlEVlFRRG5F
dzVFVTFRZ1VtOXZkQ0JEUVNCWU16Q0NBU0l3RFFZSktvWklodmNOQVFFQkJRQURnZ0VQQURDQ0FR
b0NnZ0VCQU4rdjZaZHFRQ0lObWRRQ0lPWHRNeGlaZmFZZ3V6SDB5eEhJNTXBiN05uRGZjZEF3UmdVaStEb0z
WkpLdFJOVVNkVVtPY3V5WHVUWWpUUyFOSGJzTmhTSSbzOVMRsQdeFEXpBPnVsMhHNEdpSVMsZWtUTE5KbzJ
TXZlZ3dHTW9PaWZvb1VNTTBSaGRPSFlVPSZZXBBMWVlIc1Vt
dTVvdUpqZlpTNlpCV2lOQjc5TjA2WThVdEtaZnRKZWdQQUgkVmtVNVF5d2pGeFdpSmF1SFZVNTFlYzlS
VFdNSW4vNVdnVGUxUUx5TmF1NOZxY2toNDlaVEQrL3lVRnduuNOJaeTFTTWNPR3RMnPRlU1UTlEOC9S
```

```
aGNRUEdYNjlXYWOOMGR1dG9sdWNiWTM4RVZBanFyMmO3eFBpNzFYQWljUE5hRG5hZVFRbXhrcXRp
bFgOK1U5bTUvdOFsMENBdOVBQWFOQO1FQXdEd1lEVlIwVEFRSC9CQVV3QXdFQi96QU9CZO5WbkhR
OEJBZjhFQkFNQOFRWXdIUVlEVlIwTOJCWUVGTVNuc2FSNOxISDYyKOZMaOhYL3hCVmdoWWtRTUEw
RONTcUduUOliMORRRUJCUVVBQTRJQkFRQ2pHaXliRndCY3FSN3VLR1kzT3IrRHh6OUx3d21nbFNC
ZDQ5bFpSTkkrRFQ2OW5pa3VnZEIvTOVJS2NkQm9kZnBnYTNjc1RTNO1nUk9TUjZjejhmYVhiYXVY
KzV2M2dUdDIzQURxMWNFbXY4dVhybkF2SFJBb3NaeTVRNlhrakVHQjVZR1Y4ZUFscndEUEd4cmFu
Y1dZYUxidW1SOVliSytybG1NNnBaVzg3aXB4WnpuUjhzcnpKbXdOMGpQNDFaTDljOFBESEl5aDhi
d1JMdFRjbTFEOVNaSW1sSm5OMWlyL21kMmNYamJEYUpXRkJNNW5KREdGb3FnQ1dqQkgOZDFRQjd3
QONaQUE2MlJqWUpzV3ZJakpFdWJTZlpHTCtUMHlqV1cwNlh5eFYzYnF4YllvbK9iOFZaUnpJOW5l
V2FncU5kd3ZZa1FzRWpnZmJLYllLN3AyQO5UVVFuLSOtLS1FTkQgQOVSVElGSUNBVEUtLSOtLT8K"
            },
            "enrollid": "",
            "enrollsecret": "",
            "admincerts": [""]
        },
        "tls": {
            "cahost": "",
            "caport": "",
            "caname": "",
            "catls": {
                "cacert": ""
            },
            "enrollid": "",
            "enrollsecret": "",
            "csr": {
                "hosts": [""]
            }
        }
    }
}
```

3. Register your peer by completing the following steps:

   a. Go to the Cloud console for your starter or enterprise account to the dashboard.

   b. From the left-side menu, select **Certificate Authority**. On the CA, click **Add Use**, as shown in Figure 15 on page 32.



Add User

Enroll ID: Enter an enroll ID

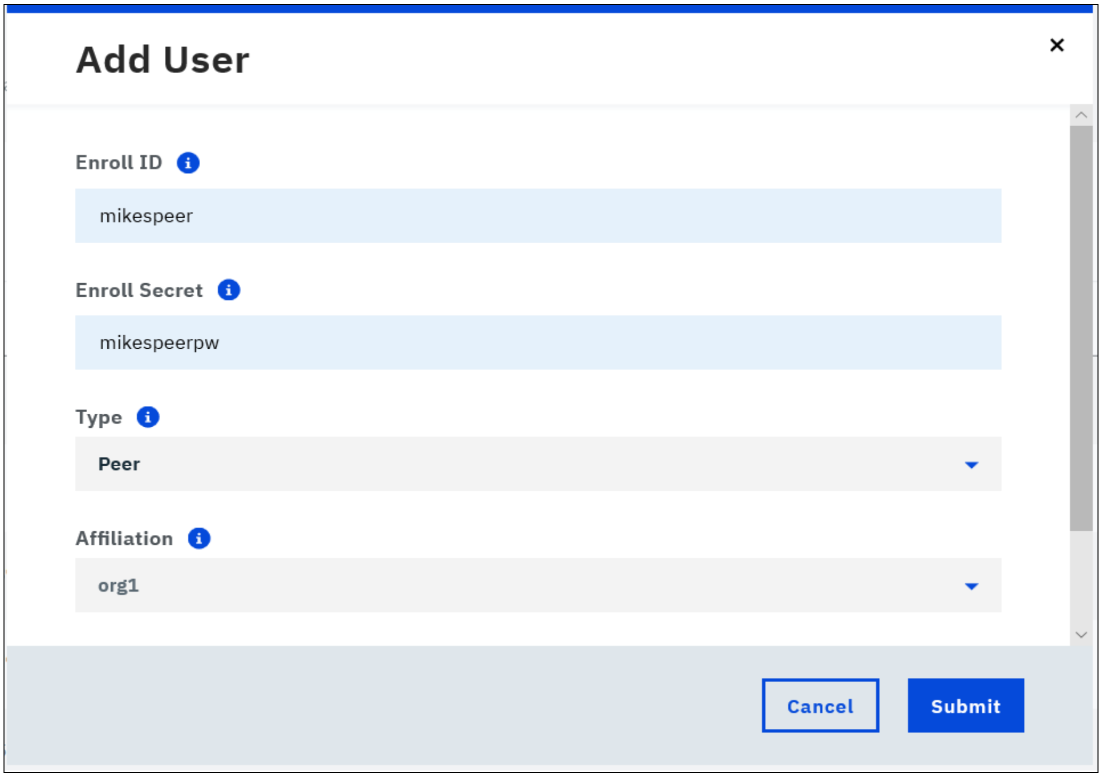
Enroll Secret: Enter an enroll secret

Type: Client

Affiliation: org1

Figure 15 Registering your peer

c. Complete the values for your new peer (in this case, mikespeer, mikespeerpw, and the type should be set to peer, as shown in Figure 16). Ensure that all values are correct because after you submit the user, you cannot delete or change it. Click **Submit** to register your peer.



Add User

Enroll ID: mikespeer

Enroll Secret: mikespeerpw

Type: Peer

Affiliation: org1

Figure 16 Values for new peer

4. The same values that are used for the peer user should be added to the JSON connection profile. The file section for the peer now looks like the following example:

```
"enrollid": "mikespeer",
"enrollsecret": "mikespeerpw",
"admincerts": [""]
```

5. Create a place to store information and certs for the peer and reset your `$HOME` logical, as shown in the following example:

```
[root@ibpzms03 ~]# cd $HOME/fabric-ca-client
[root@ibpzms03 fabric-ca-client]# mkdir peer-admin
[root@ibpzms03 fabric-ca-client]# mkdir tls-ibp
[root@ibpzms03 fabric-ca-client]# export
FABRIC_CA_CLIENT_HOME=$HOME/fabric-ca-client/peer-admin
```

6. Download your root cert for your Starter or Enterprise Plan and copy it into your `tls-ibp` directory, as shown in the following example:

```
[root@ibpzms03 ~]# cp us07.blockchain.ibm.com.cert
$HOME/fabric-ca-client/tls-ibp/tls.pem
```

> **Note:** Download the TLS certs from IBM Cloud depending on the service plan, location, and cluster that you use. You can find your cluster based on the domain name of your certificate authority URL as stored in the JSON file from the IBP console connection profile JSON file; for example: `us01.blockchain.ibm.com:31011` or `us02.blockchain.ibm.com:31011`.
>
> Then, the root cert is at `loc.blockchain.ibm.com.cert`.
>
> Where loc is the location code; for example, `us01` - `us08`. For our example, the value is `us07.blockchain.ibm.com:31001`.
>
> However, the address is different based on whether yours is a starter or enterprise plan or your system is on an enterprise cluster or local node.

7. Generate certificates for our Peer Admin that we registered by using the following commands for fabric-ca-client enroll:

```
-u https://<admin peer name>:<admin peer secret>@<CA URL with Port>
--caname <CA Name in Connection Profile>
--tls.certfiles <path to tls-ibp/tls.pem>


[root@ibpzms03 fabric-ca-client]# cd fabric-binaries

[root@ibpzms03 fabric-binaries]# fabric-ca-client enroll
-u https://mikespeer:mikespeerpw@9.60.87.24:30722
--caname OrdererCA
--tls.certfiles $HOME/fabric-ca-client/tls-ibp/tls.pem

2019/03/05 21:30:38 [INFO] Created a default configuration file at
/root/fabric-ca-client/peer-admin/fabric-ca-client-config.yaml
2019/03/05 21:30:38 [INFO] TLS Enabled
2019/03/05 21:30:38 [INFO] generating key: &{A:ecdsa S:256}
2019/03/05 21:30:38 [INFO] encoded CSR
2019/03/05 21:30:43 [INFO] Stored client certificate at
/root/fabric-ca-client/peer-admin/msp/signcerts/cert.pem
```

```
2019/03/05 21:30:43 [INFO] Stored root CA certificate at
/root/fabric-ca-client/peer-admin/msp/cacerts/fft-zbc01c-4-blockchain-ibm-com-2
0260-mikespeerCA.pem
2019/03/05 21:30:43 [INFO] Stored intermediate CA certificates at
/root/fabric-ca-client/peer-admin/msp/intermediatecerts/fft-zbc01c-4-secure-blo
ckchain-ibm-com-20260-PeerOrg2CA.pem
```

8. Convert the peer-admin's cert into a BASE64 cert so that we can continue to complete our configuration. The following basic command is used:

   ```
   cat $HOME/fabric-ca-client/peer-admin/msp/signcerts/cert.pem | base64 $FLAG
   ```

   After the command is run, take the resulting certificate and place it in the admincert portion of our configuration file:

   ```
      "enrollid": "mikespeer",
      "enrollsecret": "mikespeerpw",
      "admincerts":
      ["LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5ekNDQW5LZ0F3SUJBZ0lVQkJXU3RQSD
      ZN
   ```

   We continue to complete our configuration file with more information. All of this information falls under the tls section, as shown in the following example:

   ```
      cahost": 9.60.87.24 # Your CA URL without its port
      caport": 30722 # Your CA port
      caname": OrdererCA # Your CA name from your CA deployment
   ```

9. Get our certificate by issuing the following command (with your variables):

   ```
   [root@ibpzms03 fabric-ca-client]# cat $HOME/fabric-ca-client/catls/tls.pem |
   base64 $FLAG
   ```

   Place the output into the tls section of the configuration file:

   ```
   "tls": {
      "cahost": "9.60.87.24",
      "caport": "30727",
      "caname": "OrdererCA",
      "catls": {
         "cacert":
         "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUNGakNDQWIyZ0F3SUJBZ0lVR0xkeFc5
         eU
   ```

10. Create a directory and export its logical:

    ```
    [root@ibpzms03 ~]# cd $HOME/fabric-ca-client
    [root@ibpzms03 fabric-ca-client]# mkdir tlsca-admin
    [root@ibpzms03 fabric-ca-client]# export
    FABRIC_CA_CLIENT_HOME=$HOME/fabric-ca-client/tlsca-admin
    ```

11. Generate the certificates of our TLS CA Admin by using the following syntax:

    ```
    fabric-ca-client enroll
    -u https://<Username for CA secret>:<Password for CA secret@<Your CA Deployment
    with URL>
    --caname <Your Deployed CA Name>
    --tls.certfiles >Path to catls/tls.pem file>

    [root@ibpzms03 fabric-ca-client]# fabric-ca-client enroll -u
    https://ord-ca-admin:secure_password@9.60.87.24:30722

       --caname OrdererCA
    ```

```
   --tls.certfiles $ibp4icp_install_dir/catls/tls.pem

2019/03/05  21:35:40 [INFO] Created a default configuration file at
/Users/Austin/fabric-ca-client/tlsca-admin/fabric-ca-client-config.yaml
2019/03/05 21:35:40 [INFO] TLS Enabled
2019/03/05 21:35:40 [INFO] generating key: &{A:ecdsa S:256}
2019/03/05 21:35:40 [INFO] encoded CSR
2019/03/05 21:35:49 [INFO] Stored client certificate at
/Users/Austin/fabric-ca-client/tlsca-admin/msp/signcerts/cert.pem
2019/03/05 21:35:49 [INFO] Stored root CA certificate at
/Users/Austin/fabric-ca-client/tlsca-admin/msp/cacerts/5-3-19-115-31216-tlsca.p
em
```

12. Determine what your affiliation by using the following syntax:

```
fabric-ca-client affiliation list
--caname <CA caname>
--tls.certfiles <Path to /catls/tls.pem file>

[root@ibpzms03 fabric-ca-client]# fabric-ca-client affiliation list --caname
OrdererCA
--tls.certfiles $HOME/fabric-ca-client/catls/tls.pem
affiliation: .
affiliation: org2
  affiliation: org2.department1
affiliation: org1
affiliation: org1.department1
  affiliation: org1.department2
```

13. Register our peer by using the following syntax:

```
fabric-ca-client register --caname <Your CA Deployed CA name>
--id.affiliation <Your affiliation>
--id.name <Peer name>
--id.secret <Peer secret>
--id.type peer
--tls.certfiles <Path to /catls/tls.pem file>
```

For example:

```
[root@ibpzms03 fabric-ca-client]# fabric-ca-client register
--caname OrdererCA
--id.affiliation org1.department1
--id.name mikestlspeer
--id.secret mikestlspeerpw
--id.type peer
--tls.certfiles /root/fabric-ca-client/catls/tls.pem
2019/03/05 21:38:33 [INFO] Configuration file location:
/root/fabric-ca-client/tlsca-admin/fabric-ca-client-config.yaml
2019/03/05 21:38:33 [INFO] TLS Enabled
2019/03/05 21:38:33 [INFO] TLS Enabled Password: mikestlspeerpw
Fillout more of your JSON file:
"enrollid": "mikestlspeer",
"enrollsecret": "mikestlspeerpw"
```

14. For the CSR section of the configuration file, add your proxy node IP address and then what you are going to call your peer helm chart, as shown in the following example:

```
"csr": {
"hosts": [ "9.60.87.24",
"mikespeer" ] }
```

The configuration process for JSON is completed. You can now create your configuration file (`secret.json`).

Encode your `secret.json` file into base64 format to put it in IBM Cloud Private, as shown in the following example:

```
[root@ibpzms03 fabric-ca-client]# cat secret.json | base64 $FLAG
```

Optionally, you can encode your CouchDB information that is used later, as shown in the following example:

```
[root@ibpzms03 fabric-ca-client]# echo -n 'admin' | base64 $FLAG
```

15. Log on to IBM Cloud Private and create your Peer's secret. Also, create the secret that is required to enable CouchDB as your state database. Select **Configuration →Secrets**. Then, click **Create Secret**, as shown in Figure 17.

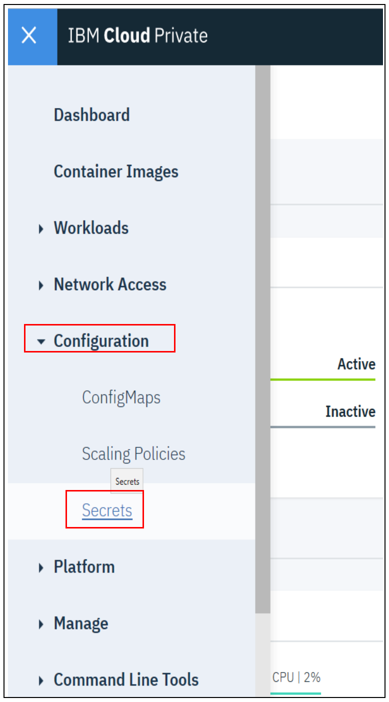
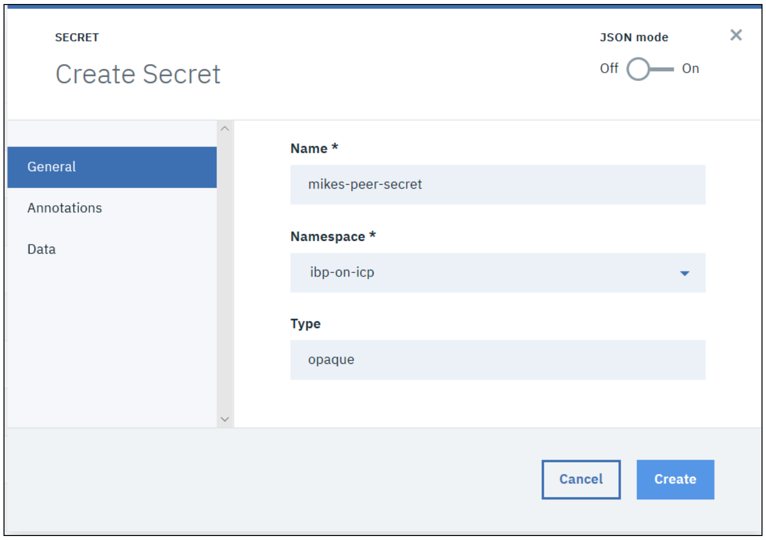


Figure 17 Creating Secrets

The Create Secret pop-up window opens, as shown in Figure 18. Enter the secret name, select the correct namespace for your peer, and enter the type of secret as `opaque`. Click the **Data menu** item.



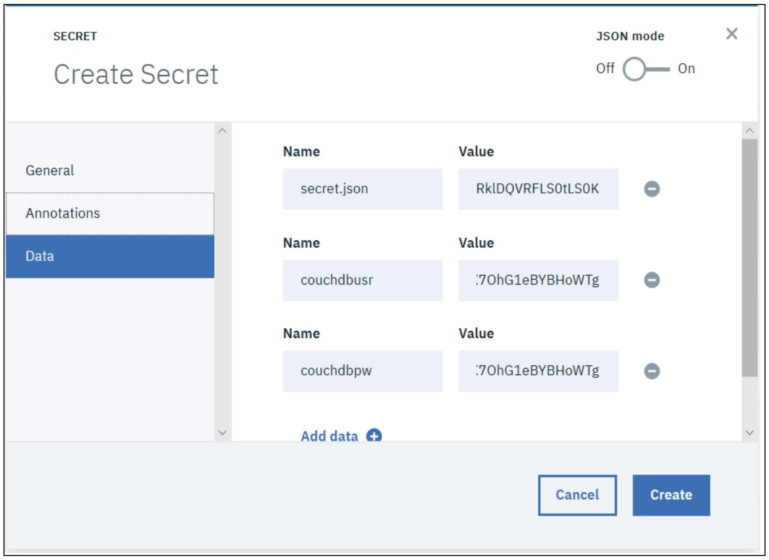
Name *
mikes-peer-secret

Namespace *
ibp-on-icp

Type
opaque

Figure 18 Creating a secret name

In the data area, we add three data items: one for the peer and two for the CouchDB. The peer value is the `secret.json` file and points to its contents. The CouchDB has a user and password that point to the CouchDB base64 value that was created in the optional section of step 13 on page 35. After you complete copying in the certifications, click **Create** to create the secret (see Figure 19).



Name	Value
secret.json	RklDQVRFLS0tLS0K
couchdbusr	:7OhG1eBYBHoWTg
couchdbpw	:7OhG1eBYBHoWTg

Figure 19 Adding Data item secrets

After the peer and CouchDB secrets are established, you have all of the required data to complete the configuration of the peer and install it. Return to the Cloud Private Console, click **Catalog,** choose **Blockchain**.

In the helm configuration chart, complete the peer section with the values that are listed in Table 6.

Table 6 Values for Peer section

Parameter	Description	Value
Install Peer	Select to install a peer.	Selected
Peer worker node architecture	Select your cloud platform architecture (AMD64 or S390x).	S390x
Peer image repository	Location of the Peer Helm Chart. This field is autofilled to the installed path.	ibmcom/ibp-fabric-peer
Peer Docker image tag	Autofilled to the version of the Peer image.	1.2.1
Peer configuration	You can customize the configuration of the peer. This information overwrite the content in the peer configuration file; that is, `core.yaml`.	None
Peer configuration secret (Required)	Name of the Peer configuration secret you created in IBM Cloud Private.	mikes_peer_secret
Organization MSP (Required)	This value can be found in Network Monitor (IBP UI) by clicking **Remote Peer Configuration** in the Overview window. If you are not connecting to an IBP network, you can create an Organization MSP value, such as “org1” or specify an Organization MSP of which the peer will be a part.	Org1
Peer service type	Used to specify whether external ports should be exposed on the peer. Select **NodePort** to expose the ports externally (recommended), and **ClusterIP** to not expose the ports. LoadBalancer and ExternalName are not supported in this release.	NodePort
State database	The state database that is used to store your channel ledger. The peer must use the same database as your blockchain network.	couchdb
CouchDB image repository	Applies only if CouchDB is selected as the ledger database. This field is autofilled to the installed path.	ibmcom/ibp-couchdb

Parameter	Description	Value
CouchDB Docker image tag	Applies only if CouchDB is selected as the ledger database. Autofilled to the version of the CouchDB image.	0.4.10
Peer Data persistence enabled	Enable the ability to persist data after cluster restarts or fails. For more information, see storage in Kubernetes. **Note:** If not selected, all data is lost if a failover or pod restart occurs.	Selected
Peer use dynamic provisioning	Select to enable dynamic provisioning for storage volumes.	Not selected
Peer persistent volume claim	For new claim only. Enter a name for your new Persistent Volume Claim (PVC) to be created.	Blockchain_PV06
Peer storage class name	Specify a storage class name for the peer.	Blank if you want to create a new PVC; otherwise, specify the storage class that is associated with the PVC.
Peer existing volume claim	Specify the name of a Volume Claim and leave all other fields blank.	New claim name
Peer selector label	Specify the Selector label for your PVC.	Default
Peer selector value	Specify the Selector value for your PVC.	Default
Peer storage access mode	Specify the storage access mode for the PVC.	ReadWriteMany
Peer volume claim size	Size of the Volume Claim. This value must be larger than 2 Gi.	8 Gi
State database persistent volume claim	For new claim only. Enter a name for your new PVC to be created.	Blockchain_PV07
State database storage class name	Specify a storage class name for state database.	None
State database that is in volume claim	Specify the name of an existing Volume Claim and leave all other fields blank.	None
State database selector label	Specify the Selector label for your PVC.	None
State database selector value	Specify the Selector value for your PVC.	None

Parameter	Description	Value
State database storage access mode	Specify the storage access mode for the PVC.	ReadWriteMany
State database volume claim size	Choose the size of disk to use.	8 Gi
CouchDB - Data persistence enabled	For CouchDB container, ledger data will be available when the container restarts. If cleared, all data is lost if a failover or pod restart occurs.	Selected
CouchDB - Use dynamic provisioning	For CouchDB container use Kubernetes dynamic storage.	Not selected
Peer CPU request	Minimum number of CPUs to allocate to the peer.	
Peer CPU limit	Maximum number of CPUs to allocate to the peer.	
Peer Memory request	Minimum amount of memory to allocate to the peer.	1 Gi
Peer Memory limit	Maximum amount of memory to allocate to the peer.	4 Gi
CouchDB CPU request	Minimum number of CPUs to allocate to CouchDB.	
CouchDB CPU limit	Maximum number of CPUs to allocate to CouchDB.	
CouchDB Memory request	Minimum amount of memory to allocate to CouchDB.	1 Gi
CouchDB Memory limit	Maximum amount of memory to allocate to CouchDB.	4 Gi

After the values are entered, click **Install**.

16. Confirm that your Peer is working by reviewing the logs of the `init` container, by using the following syntax:

```
kubectl logs <Your Peer's Pod> -c init | grep EXIT
```

Example:

```
[root@ibpzms03 fabric-ca-client]# kubectl logs mikespeer-74b89b485f-bmfs9 -c
init | grep EXIT
EXIT WITH RC=0 #
```

An RC=0 is a normal entry that indicates that the peer is working normally.

IBM Cloud Object Storage installation

With over 600 technology patents, IBM Cloud Object Storage is a software-defined storage platform that stores massive amounts of data with efficiency, reliability, simplicity, and cloud native accessibility to transform the enterprise for multiple use cases. IBM Cloud Object Storage breaks down barriers for storing massive amounts of data by using an Information Dispersal Algorithm (IDA) and flexible configurations to spread data across multiple nodes by using IBM's patented technologies. Our proven solutions can turn storage challenges into business advantages.

The on-premises IBM Cloud Object Storage System is a breakthrough platform for storing large amounts of unstructured data. It provides scalability, availability, security with simplicity, and lower total cost of ownership (TCO). It is available as an integrated storage system or as a software-only solution. In addition, IBM Cloud Object Storage is available as a public cloud service in the IBM Cloud. IBM Cloud Object Storage is ideal for use cases, such as remote file collaboration, backup or archive repository, and as a content repository for images, video, and voice.

IBM Cloud Object Storage can integrate with analytics workloads and now offers a new metadata management and insight software with IBM Spectrum Discover. This feature makes it an ideal candidate for blockchain applications. One of the advantages for customers with the IBM Cloud Object Storage architecture is that as more use cases are put on the system, more benefits can be realized.

Clients can start with as few as three commodity x86 server nodes or as little as 72 TB and grow to exabytes of usable storage without ever losing access to the data. By combining a single copy of protected data and the ability to lock down data by using policy-based WORM storage, IBM Cloud Object Storage is quickly becoming the choice for many industries, such as finance, healthcare, and government, that have compliance or other data retention requirements.

For more information about IBM Cloud Object Storage see the following website:

https://www.ibm.com/cloud/object-storage

You must obtain from your Cloud Object Storage administrator what is known as a bucket (a place to store objects). By using the bucket identifier, access, secret keys, user, and password data, we connect the BDS instance that we create to the Cloud Object Storage bucket for off-chain storage. The Bucket data resembles the information that is listed in Table 7.

Table 7 Vault (Bucket) properties, authentication, and access

Vault (Bucket) properties	
Vault (Bucket) Name	Vault name
Endpoint	Endpoint URL
Vault (Bucket) Description	For IBM Blockchain
Vault (Bucket) User	email/id
Name Index Enabled	True
Recovery Listing Enabled	False
SecureSlice Enabled	True
Versioning	False
Compliance Enabled	False

Expiration Date	Expiration Date
Secret Key Authentication	
Access Key ID	Access Key ID
Secret Access Key	Secret Key
Virtual Host Access	
Can be an accessible URL	
Path-Style Access	
Can be an accessible URL	

IBM Blockchain Document Store

The following prerequisites must be met:

- Have an IBM ID and IBM Cloud account.
- Have a starter or enterprise level blockchain account with blockchain installed, an organization, and channel created.
- Be on the Blockchain Document Store (BDS) whitelist.

After you complete the prerequisites, you can continue the installation process.

Note: To install this platform on a remote peer, you need IBM support to add the remote peer connection certification to the IBP in the cloud’s connection JSON.

To install BDS Utilize, see the following website (log in required):

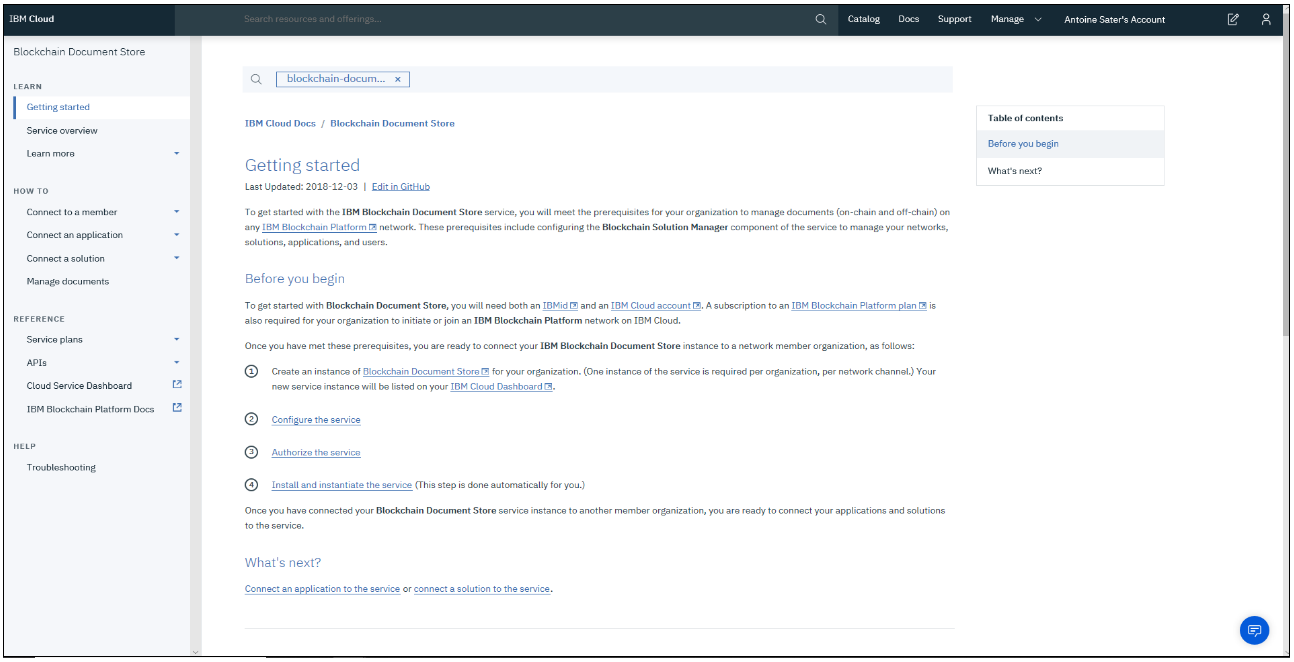
https://console.test.cloud.ibm.com/docs/services/blockchain-document-store/getting-started.html#getting-started

Note: The Blockchain Document Store is a whitelist product, which means that you cannot access it unless you are on the whitelist for BDS. After you are on the Whitelist the link (https://console.bluemix.net/catalog/services/blockchain-document-store), you can access the BDS.

Complete the following steps:

1. Browse to the following ULR to see the window that is shown in Figure 20:

   `https://console.bluemix.net/catalog/services/blockchain-document-store`



Getting started

Last Updated: 2018-12-03 | Edit in GitHub

To get started with the **IBM Blockchain Document Store** service, you will meet the prerequisites for your organization to manage documents (on-chain and off-chain) on any IBM Blockchain Platform network. These prerequisites include configuring the **Blockchain Solution Manager** component of the service to manage your networks, solutions, applications, and users.

Before you begin

To get started with **Blockchain Document Store,** you will need both an IBMid and an IBM Cloud account. A subscription to an IBM Blockchain Platform plan is also required for your organization to initiate or join an **IBM Blockchain Platform** network on IBM Cloud.

Once you have met these prerequisites, you are ready to connect your **IBM Blockchain Document Store** instance to a network member organization, as follows:

① Create an instance of Blockchain Document Store for your organization. (One instance of the service is required per organization, per network channel.) Your new service instance will be listed on your IBM Cloud Dashboard.

② Configure the service

③ Authorize the service

④ Install and instantiate the service (This step is done automatically for you.)

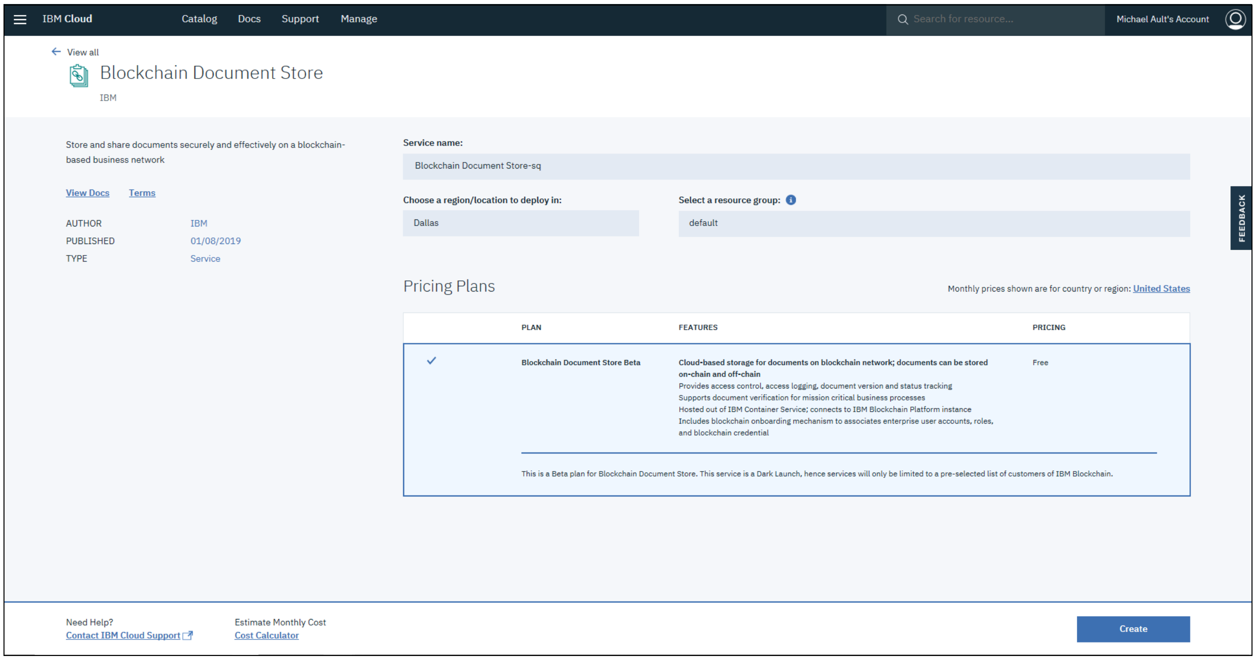
Once you have connected your **Blockchain Document Store** service instance to another member organization, you are ready to connect your applications and solutions to the service.

What's next?

Connect an application to the service or connect a solution to the service.

Figure 20   Link to the Blockchain Document Store

2. From this window, select the link to the catalog. You are taken to the Blockchain Document Store installation package. If you are not taken to the package, you are not on the whitelist. You must be on the whitelist to proceed. If the connection is unsuccessful, you see the standard catalog. If the connection is it is successful, you see the menu that is shown in Figure 21.



Store and share documents securely and effectively on a blockchain-based business network

AUTHOR IBM
PUBLISHED 01/08/2019
TYPE Service

Service name: Blockchain Document Store-sq

Choose a region/location to deploy in: Dallas

Select a resource group: default

Pricing Plans

Monthly prices shown are for country or region: United States

	PLAN	FEATURES	PRICING
✓	Blockchain Document Store Beta	**Cloud-based storage for documents on blockchain network; documents can be stored on-chain and off-chain** Provides access control, access logging, document version and status tracking Supports document verification for mission critical business processes Hosted out of IBM Container Service; connects to IBM Blockchain Platform instance Includes blockchain onboarding mechanism to associates enterprise user accounts, roles, and blockchain credential	Free

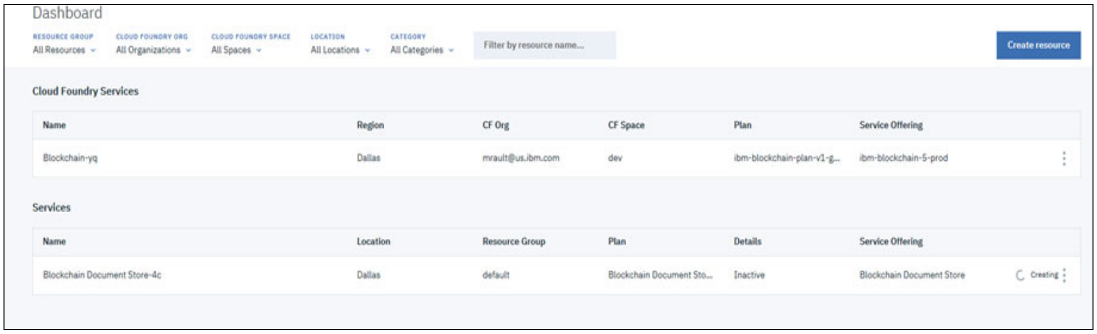
This is a Beta plan for Blockchain Document Store. This service is a Dark Launch, hence services will only be limited to a pre-selected list of customers of IBM Blockchain.

Need Help?
Contact IBM Cloud Support

Estimate Monthly Cost
Cost Calculator

Figure 21   Blockchain Document store

3. Click **Create** to create an instance of BDS in your blockchain.
4. After the instance is created, you must configure it. Open your blockchain services dashboard. The window that is shown in Figure 22 is displayed.



Cloud Foundry Services

Name	Region	CF Org	CF Space	Plan	Service Offering
Blockchain-yq	Dallas	mrault@us.ibm.com	dev	ibm-blockchain-plan-v1-g...	ibm-blockchain-5-prod

Services

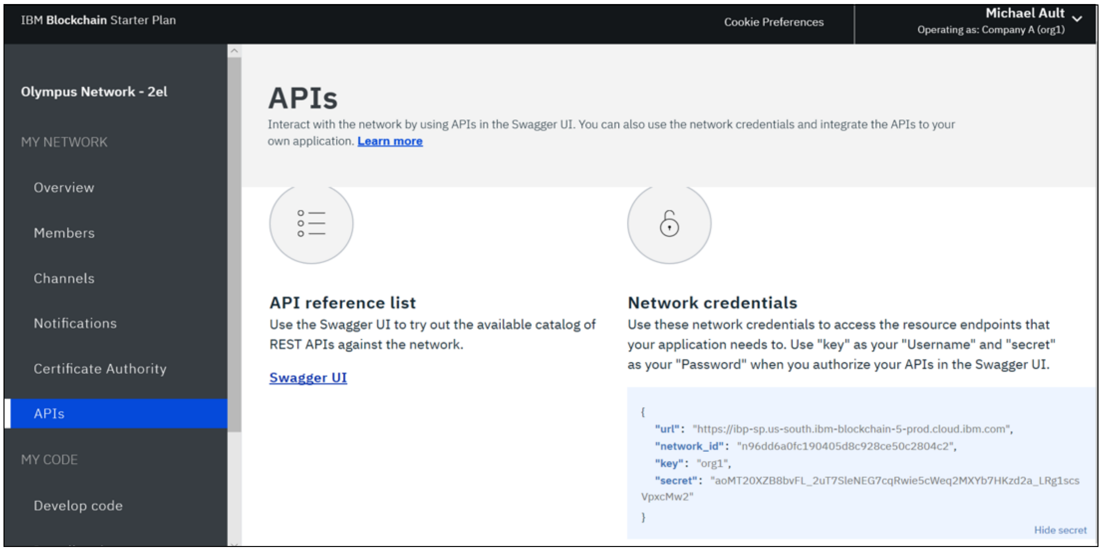
Name	Location	Resource Group	Plan	Details	Service Offering	
Blockchain Document Store-4c	Dallas	default	Blockchain Document Sto...	Inactive	Blockchain Document Store	Creating

Figure 22 Blockchain service dashboard

5. From the listed services, select the BDS service and then, click **Create Instance**.

   After the instance is created, you must configure it so it can connect to your peer node network, which requires a JSON network credential.

6. Select the Blockchain service and then, select **Monitor** from the window. Select the **APIs** tab, as shown in Figure 23.



APIs

Interact with the network by using APIs in the Swagger UI. You can also use the network credentials and integrate the APIs to your own application. Learn more

API reference list

Use the Swagger UI to try out the available catalog of REST APIs against the network.

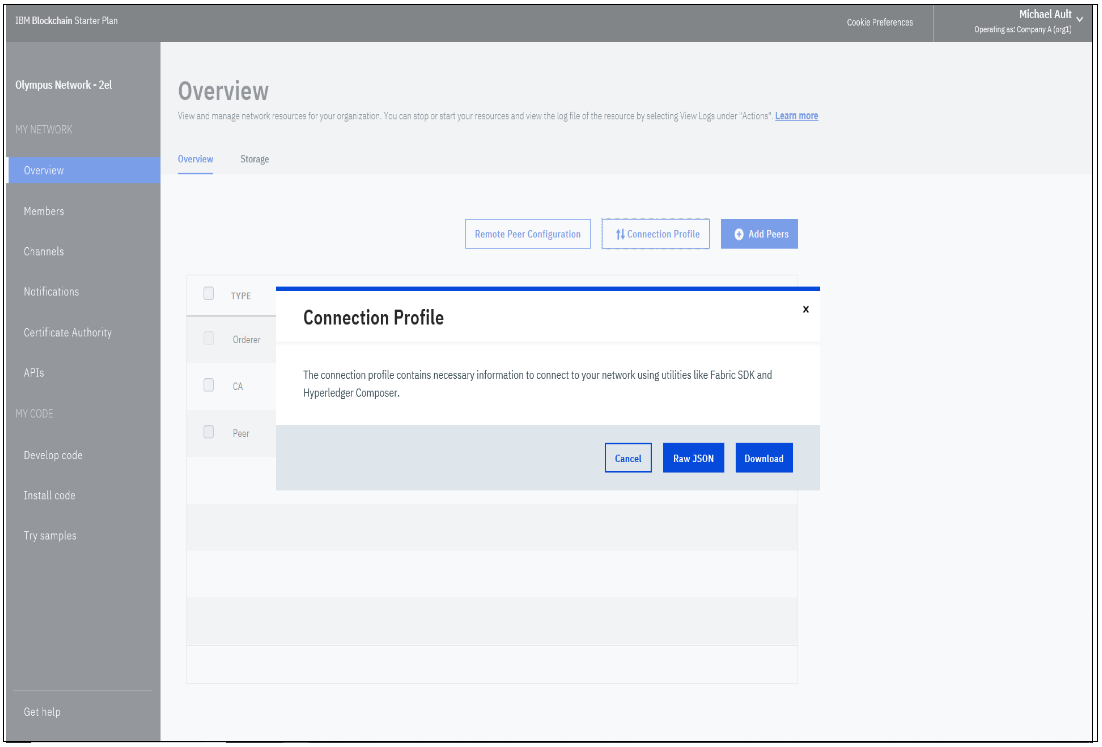
Network credentials

Use these network credentials to access the resource endpoints that your application needs to. Use "key" as your "Username" and "secret" as your "Password" when you authorize your APIs in the Swagger UI.

```
{
   "url":  "https://ibp-sp.us-south.ibm-blockchain-5-prod.cloud.ibm.com",
   "network_id":  "n96dd6a0fc190405d8c928ce50c2804c2",
   "key":  "org1",
   "secret":  "aoMT20XZB8bvFL_2uT7SleNEG7cqRwie5cWeq2MXYb7HKzd2a_LRg1scs
VpxcMw2"
}
```

Figure 23 APIs

7. Return to the overview window and select **C** to see the required JSON script (see Figure 24). Select **Raw JSON** and copy the JSON script, or download it to a file so that you can upload it later.



Overview

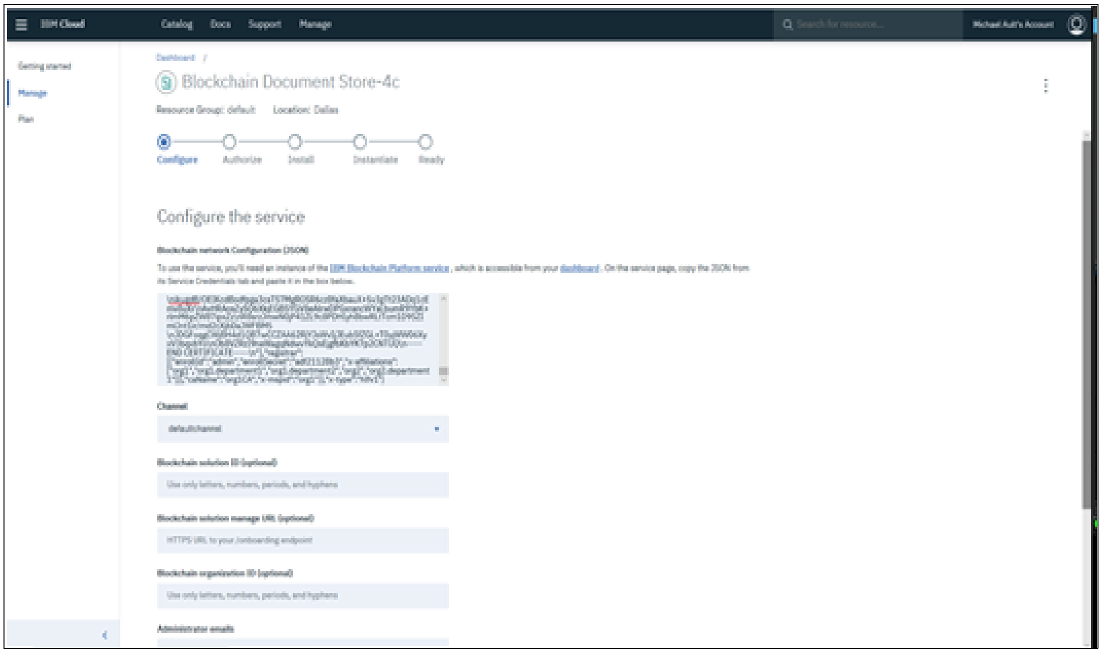
View and manage network resources for your organization. You can stop or start your resources and view the log file of the resource by selecting View Logs under "Actions". Learn more

Connection Profile

The connection profile contains necessary information to connect to your network using utilities like Fabric SDK and Hyperledger Composer.

Figure 24 Obtaining the JSON script

8. Return to the services and select the **BDS** service. Select **Manage** and the configuration window that is shown in Figure 25 opens.



Blockchain Document Store-4c

Resource Group: default Location: Dallas

Configure the service

Blockchain network Configuration (JSON)

To use the service, you'll need an instance of the IBM Blockchain Platform service, which is accessible from your dashboard. On the service page, copy the JSON from its Service Credentials tab and paste it in the box below.

[illegible]

Channel

defaultchannel

Blockchain solution ID (optional)

Use only letters, numbers, periods, and hyphens

Blockchain solution manage URL (optional)

HTTPS URL to your /onboarding endpoint

Blockchain organization ID (optional)

Use only letters, numbers, periods, and hyphens

Administrator emails

Figure 25 Configure the services

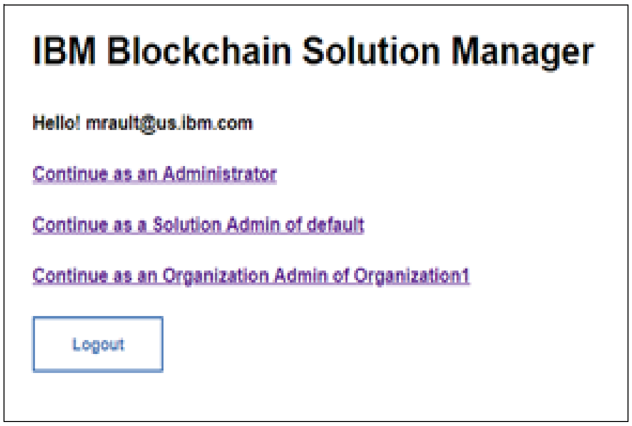
9. Using the information that was gathered in the previous steps, complete the necessary values.

10. Complete the following steps to authorize the service:

    a. You need the administrator certificate from the IBM Blockchain Solutions Manager. Browse to the following URL to get to the service:

    `https://pbsa-prod.us-south.containers.mybluemix.net/1e627852-f65e-48e6-98b0-002de67ff533/onboarding/v1/logins`

    This URL is the base URL for the BDS service plus the path to the onboarding service, as shown in Figure 26.



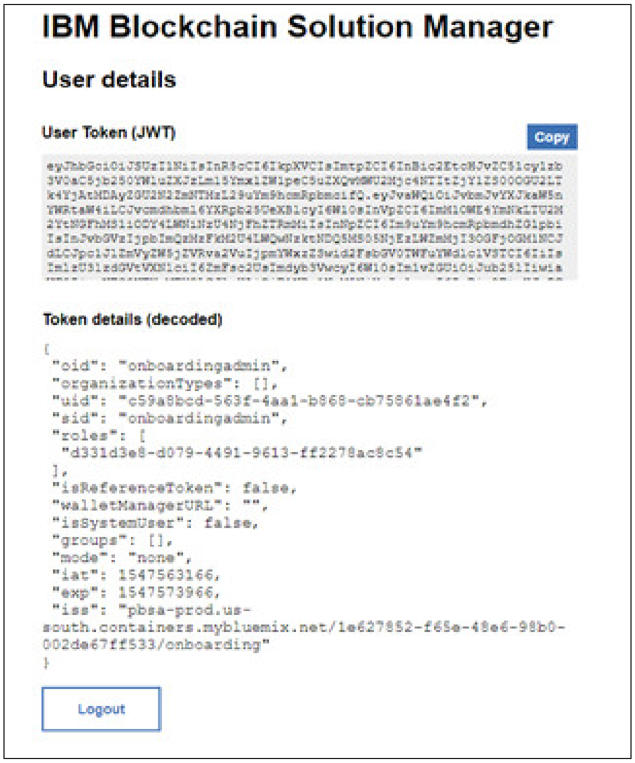
IBM Blockchain Solution Manager

Hello! mrault@us.ibm.com

Continue as an Administrator
Continue as a Solution Admin of default
Continue as an Organization Admin of Organization1
Logout

Figure 26 Top-level IBM Blockchain Solution Manager window

    b. From this window, select **Continue as Solution Admin of default**, as shown in Figure 27.



IBM Blockchain Solution Manager

User details

User Token (JWT)

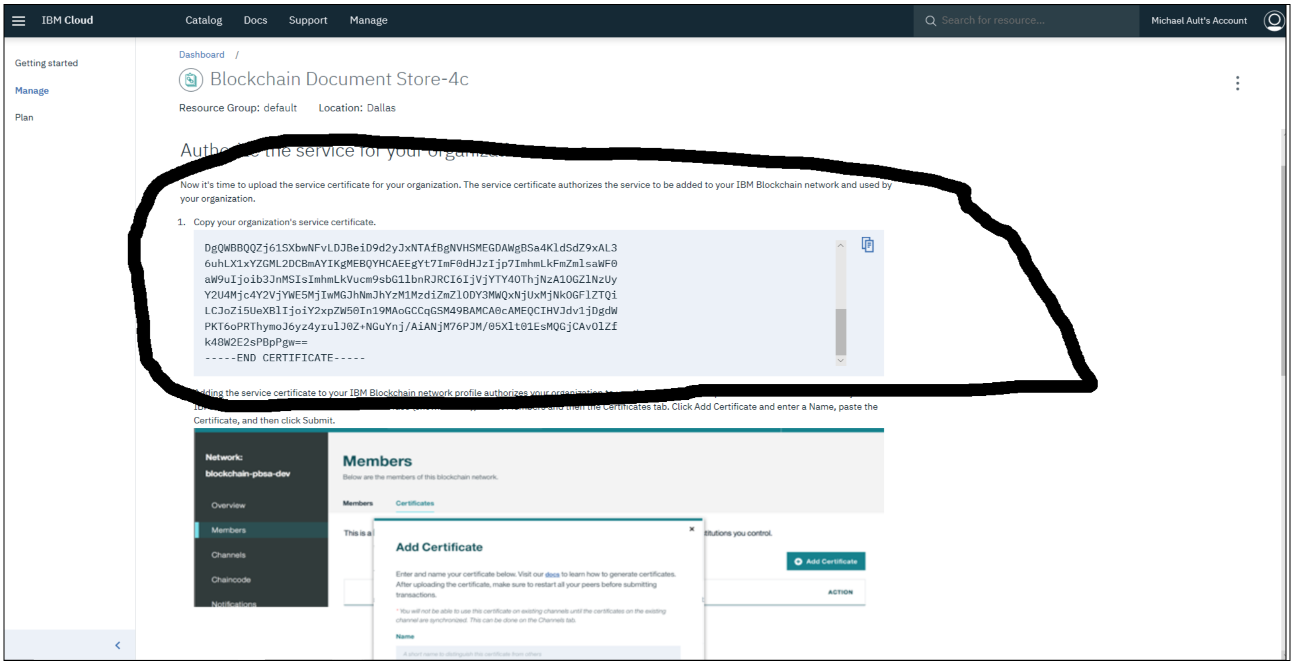
```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InBic2EtcHJvZC51cy1zb
3V0aC5jb250YWluZXJzLm15Ymx1ZW1peC5uZXQvMWU2Mjc4NTItZjY1ZS00OGU2LT
k4YjAtMDAyZGU2N2ZmNTMzL29uYm9hcmRpbmcifQ.eyJvaWQiOiJvbmJvYXJkaW5n
YWRtaW4iLCJvcmdhbml6YXRpb25UeXBlcyI6W10sInVpZCI6ImM1OWE4YmNkLTU2M
2YtNGFhMS1iODY4LWNiNzU4NjFhZTRmMiIsInNpZCI6Im9uYm9hcmRpbmdhZG1pbi
IsInJvbGVzIjpbImQzMzFkM2U4LWQwNzktNDQ5MS05NjEzLWZmMjI3OGFjOGM1NCJ
dLCJpc1JlZmVyZW5jZVRva2VuIjpmYWxzZSwid2FsbGV0TWFuYWdlclVSTCI6IiIs
ImlzU3lzdGVtVXNlciI6ZmFsc2UsImdyb3VwcyI6W10sIm1vZGUiOiJub25lIiwia
```

Token details (decoded)

```
{
 "oid": "onboardingadmin",
 "organizationTypes": [],
 "uid": "c59a8bcd-563f-4aa1-b868-cb75861ae4f2",
 "sid": "onboardingadmin",
 "roles": [
  "d331d3e8-d079-4491-9613-ff2278ac8c54"
 ],
 "isReferenceToken": false,
 "walletManagerURL": "",
 "isSystemUser": false,
 "groups": [],
 "mode": "none",
 "iat": 1547563166,
 "exp": 1547573966,
 "iss": "pbsa-prod.us-
south.containers.mybluemix.net/1e627852-f65e-48e6-98b0-
002de67ff533/onboarding"
}
```

Figure 27 User Details

c. Select **Copy**. Return to the Authorize window and paste the certificate into the field that is circled in Figure 28.



Blockchain Document Store-4c

Resource Group: default   Location: Dallas

Auth[illegible] the service for your organiz[illegible]

Now it's time to upload the service certificate for your organization. The service certificate authorizes the service to be added to your IBM Blockchain network and used by your organization.

1. Copy your organization's service certificate.

```
DgQWBBQQZj61SXbwNFvLDJBeiD9d2yJxNTAfBgNVHSMEGDAWgBSa4KldSdZ9xAL3
6uhLX1xYZGML2DCBmAYIKgMEBQYHCAEEgYt7ImF0dHJzIjp7ImhmLkFmZmlsaWF0
aW9uIjoib3JnMSIsImhmLkVucm9sbG1lbnRJRCI6IjVjYTY4OThjNzA1OGZlNzUy
Y2U4Mjc4Y2VjYWE5MjIwMGJhNmJhYzM1MzdiZmZlODY3MWQxNjUxMjNkOGFlZTQi
LCJoZi5UeXBlIjoiY2xpZW50In19MAoGCCqGSM49BAMCA0cAMEQCIHVJdv1jDgdW
PKT6oPRThymoJ6yz4yrulJ0Z+NGuYnj/AiANjM76PJM/05Xlt01EsMQGjCAv0lZf
k48W2E2sPBpPgw==
-----END CERTIFICATE-----
```

[illegible]dding the service certificate to your IBM Blockchain network profile authorizes your organiza[illegible] IB[illegible] [illegible]embers and then the Certificates tab. Click Add Certificate and enter a Name, paste the Certificate, and then click Submit.

Add Certificate

Enter and name your certificate below. Visit our docs to learn how to generate certificates. After uploading the certificate, make sure to restart all your peers before submitting transactions.

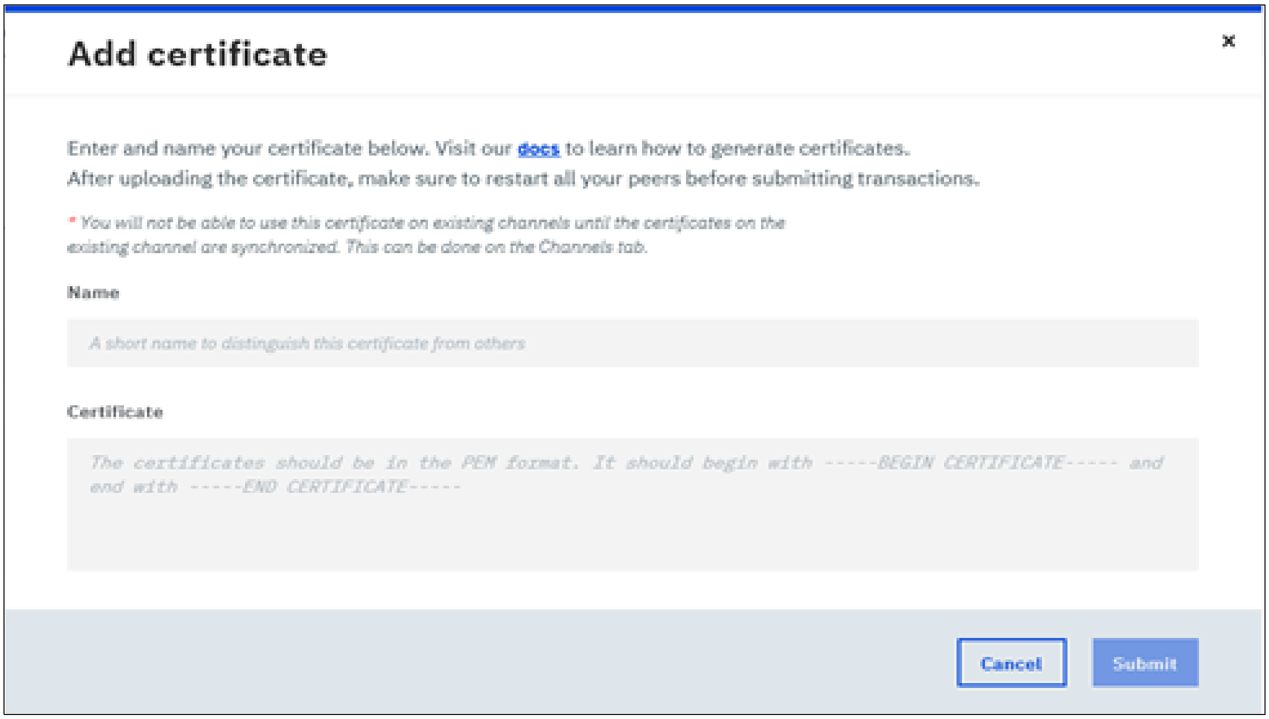
* You will not be able to use this certificate on existing channels until the certificates on the existing channel are synchronized. This can be done on the Channels tab.

Name

A short name to distinguish this certificate from others

Figure 28 Token Details

d. Click **Submit** to submit the new certificate (see Figure 29).



Add certificate

Enter and name your certificate below. Visit our docs to learn how to generate certificates.
After uploading the certificate, make sure to restart all your peers before submitting transactions.

* You will not be able to use this certificate on existing channels until the certificates on the existing channel are synchronized. This can be done on the Channels tab.

Name

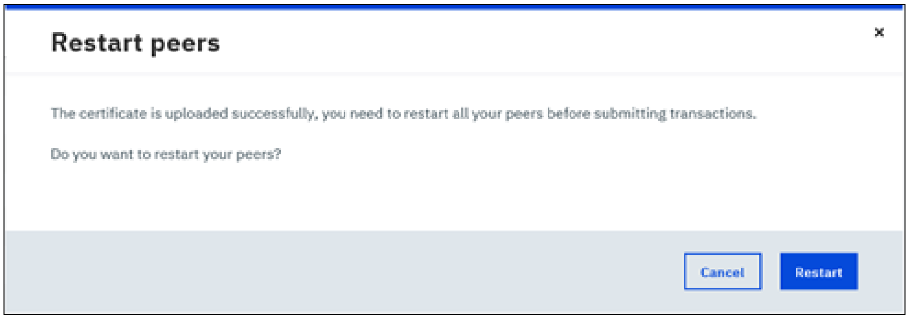
A short name to distinguish this certificate from others

Certificate

The certificates should be in the PEM format. It should begin with -----BEGIN CERTIFICATE----- and end with -----END CERTIFICATE-----

Figure 29 Adding a new certificate

e. Restart the peers so that they recognize the new certificate by clicking **Restart** (see Figure 30).



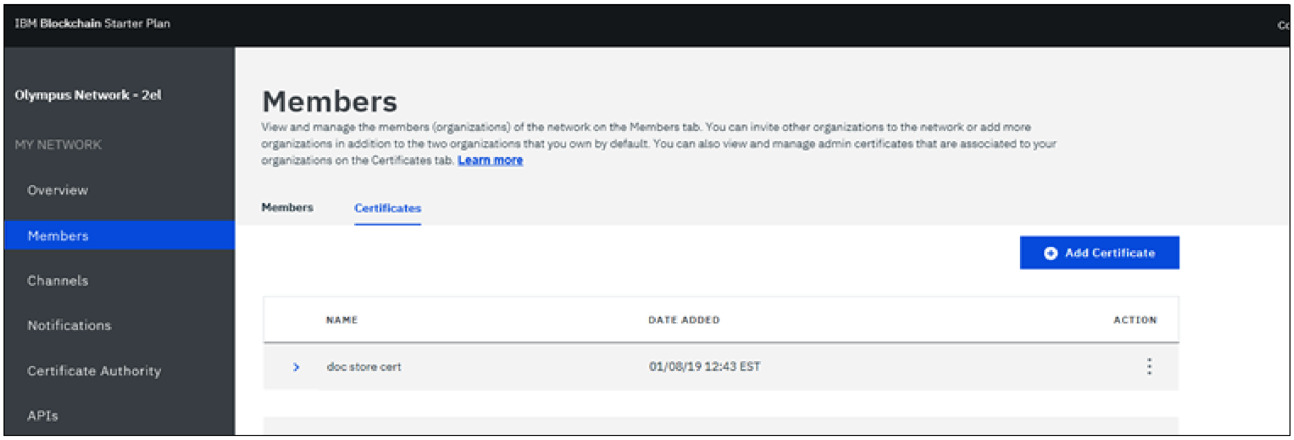
Restart peers

The certificate is uploaded successfully, you need to restart all your peers before submitting transactions.

Do you want to restart your peers?

Figure 30 Restart peers window

11. Specify the network channels to which the service is connected by clicking the **Members** menu on the IBP console. Verify that the certificate was added and that the Members are correct, as shown in Figure 31.



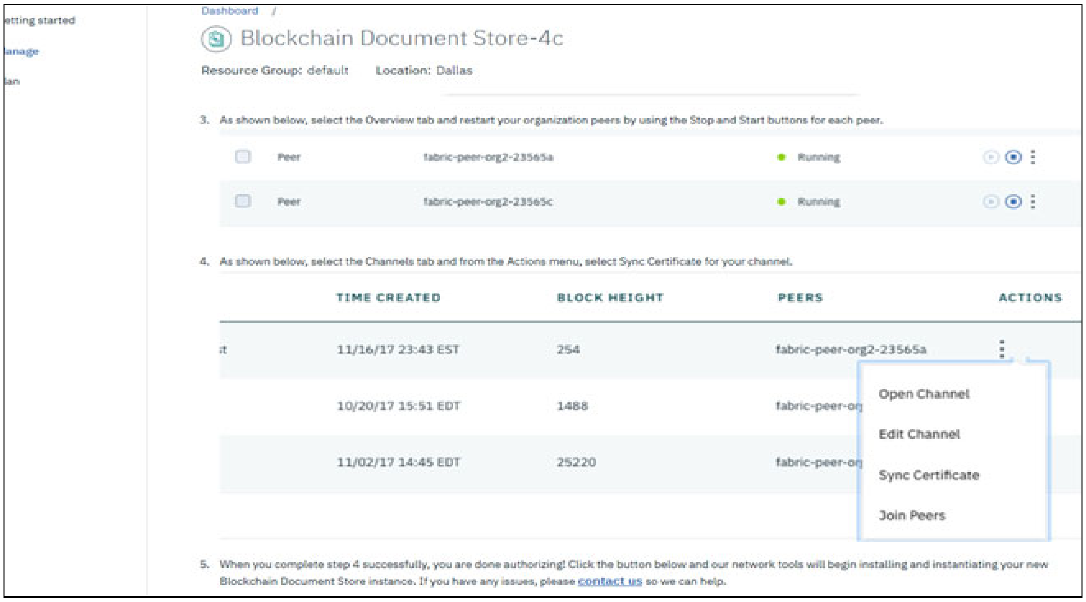
Members

View and manage the members (organizations) of the network on the Members tab. You can invite other organizations to the network or add more organizations in addition to the two organizations that you own by default. You can also view and manage admin certificates that are associated to your organizations on the Certificates tab. Learn more

NAME	DATE ADDED	ACTION
doc store cert	01/08/19 12:43 EST	⋮

Figure 31 Members menu

12. From the Blockchain console, use the right-side menu in the channel section to open the channel area of the Blockchain console (see Figure 32).



Blockchain Document Store-4c

Resource Group: default Location: Dallas

3. As shown below, select the Overview tab and restart your organization peers by using the Stop and Start buttons for each peer.

Peer	fabric-peer-org2-23565a	Running	
Peer	fabric-peer-org2-23565c	Running	

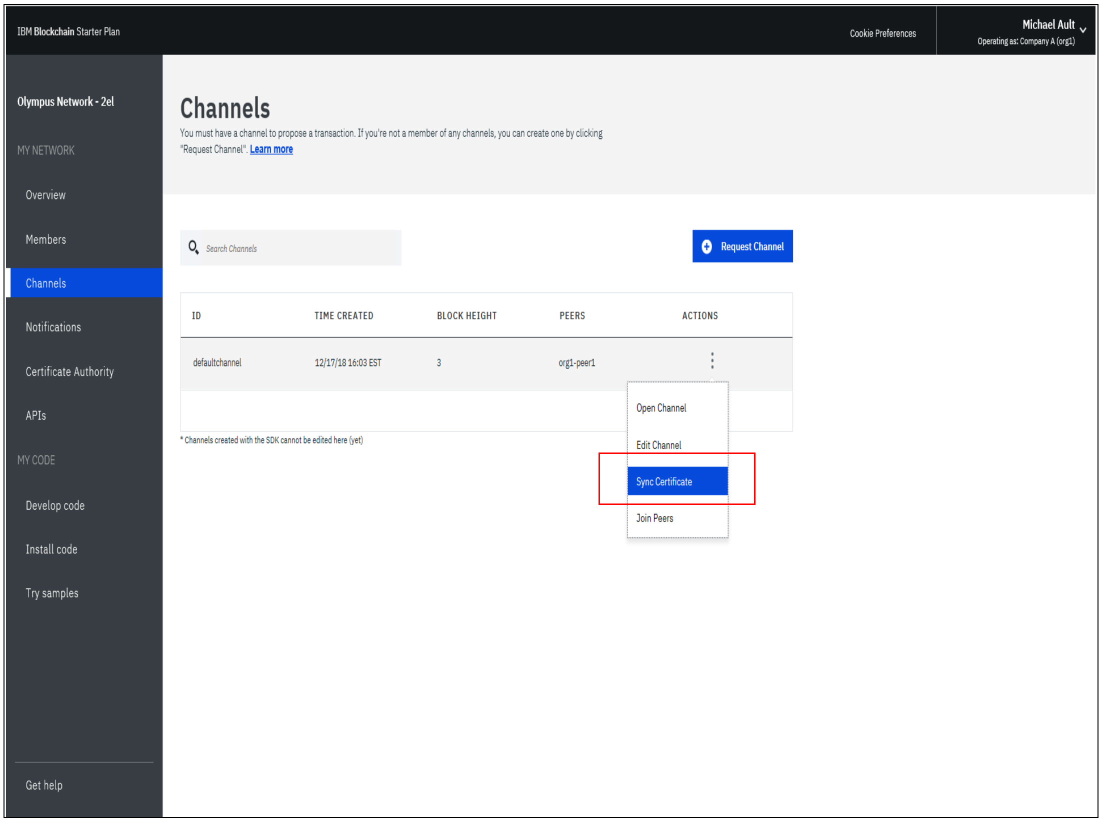
4. As shown below, select the Channels tab and from the Actions menu, select Sync Certificate for your channel.

	TIME CREATED	BLOCK HEIGHT	PEERS	ACTIONS
t	11/16/17 23:43 EST	254	fabric-peer-org2-23565a	⋮
	10/20/17 15:51 EDT	1488	fabric-peer-or	
	11/02/17 14:45 EDT	25220	fabric-peer-or	

5. When you complete step 4 successfully, you are done authorizing! Click the button below and our network tools will begin installing and instantiating your new Blockchain Document Store instance. If you have any issues, please contact us so we can help.

Figure 32 Blockchain console, channel area

13. Select the correct channel and use the right-side menu link to select **Sync Certificate** (see Figure 33).



Channels

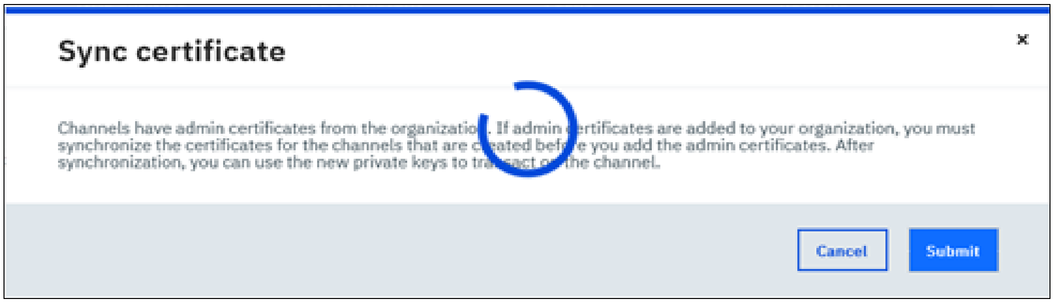
You must have a channel to propose a transaction. If you're not a member of any channels, you can create one by clicking "Request Channel". Learn more

ID	TIME CREATED	BLOCK HEIGHT	PEERS	ACTIONS
defaultchannel	12/17/18 16:03 EST	3	org1-peer1	⋮

* Channels created with the SDK cannot be edited here (yet)

Figure 33 Defining the Channels

The certificate sync occurs automatically, as shown in Figure 34.

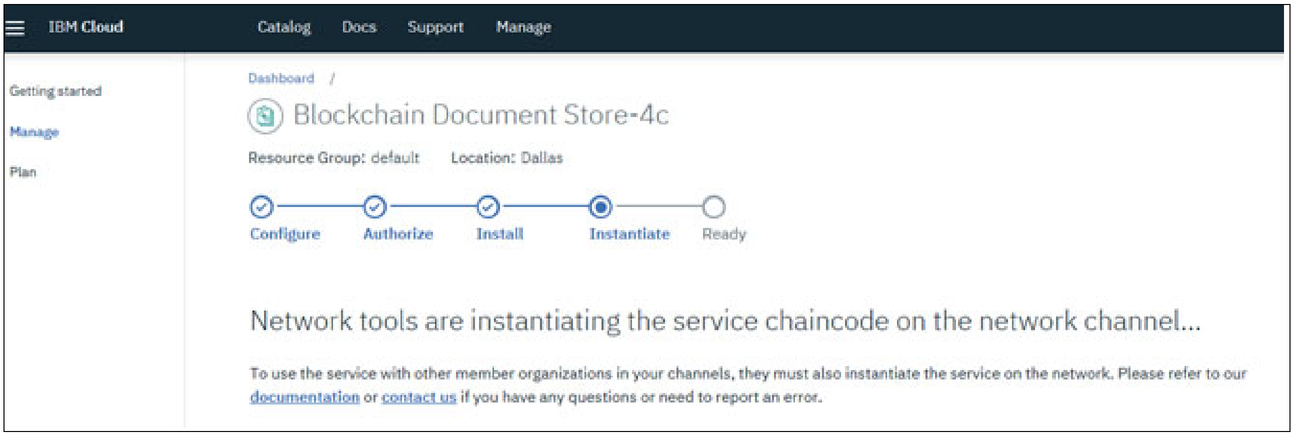


Sync certificate

Channels have admin certificates from the organization. If admin certificates are added to your organization, you must synchronize the certificates for the channels that are created before you add the admin certificates. After synchronization, you can use the new private keys to transact on the channel.

Figure 34 Sync the certificate

14. Instantiate the chain code automatically, as shown by Figure 35. This process can take several seconds to complete.



Blockchain Document Store-4c

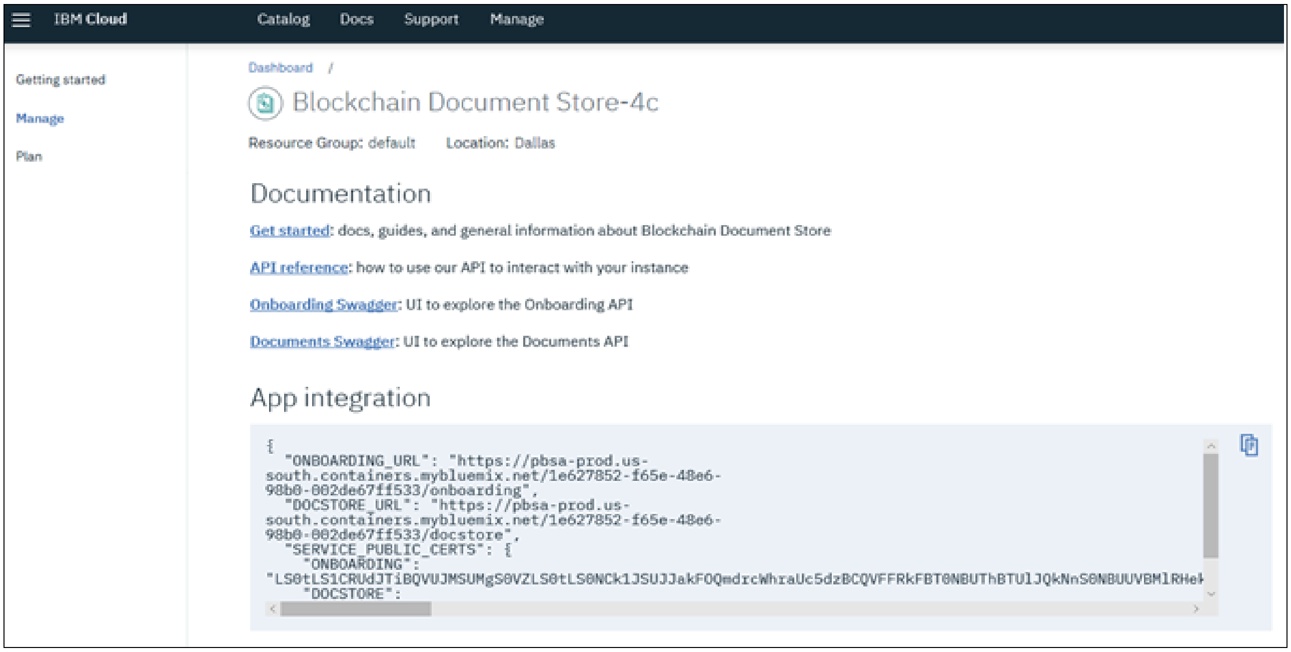
Resource Group: default Location: Dallas

Network tools are instantiating the service chaincode on the network channel...

To use the service with other member organizations in your channels, they must also instantiate the service on the network. Please refer to our documentation or contact us if you have any questions or need to report an error.

Figure 35 Instantiate the chain code

15. After the code is instantiated, the system displays the needed API links to use when an application is attached to the API (see Figure 36).



Blockchain Document Store-4c

Resource Group: default   Location: Dallas

Documentation

Get started: docs, guides, and general information about Blockchain Document Store

API reference: how to use our API to interact with your instance

Onboarding Swagger: UI to explore the Onboarding API

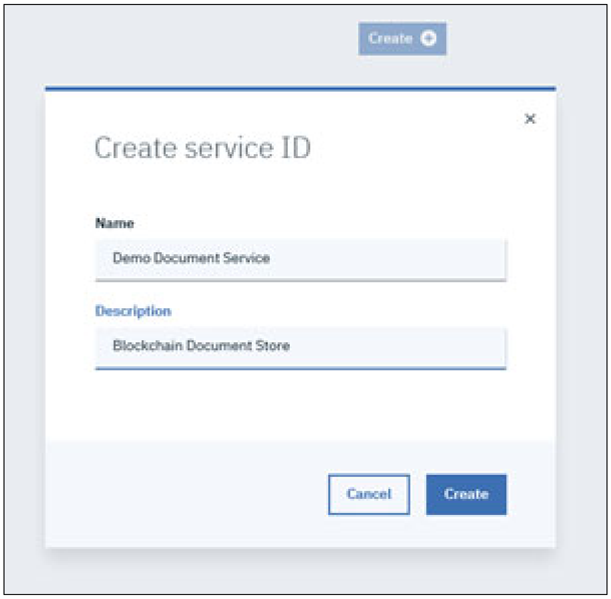
Documents Swagger: UI to explore the Documents API

App integration

```
{
  "ONBOARDING_URL": "https://pbsa-prod.us-
south.containers.mybluemix.net/1e627852-f65e-48e6-
98b0-002de67ff533/onboarding",
  "DOCSTORE_URL": "https://pbsa-prod.us-
south.containers.mybluemix.net/1e627852-f65e-48e6-
98b0-002de67ff533/docstore",
  "SERVICE_PUBLIC_CERTS": {
    "ONBOARDING":
"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0NCk1JSUJJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0NBUUVBMlRHeł
    "DOCSTORE":
```

Figure 36   API links

16. To use the BDS, an application must be interfaced. IBM provides the Swagger API to provide a basic interface to the BDS. To instantiate the Swagger API, you must obtain a service ID. A service ID is created by click the Cloud Control Console **Identity & Access** option in the Service ID submenu. Selecting that option and then **Create** opens to window that is shown in Figure 37.



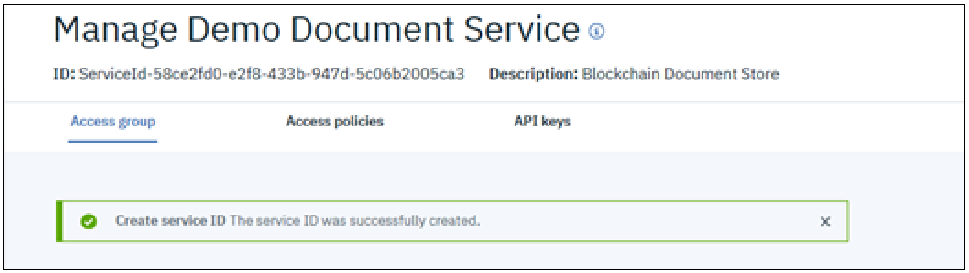
Create service ID

Name: Demo Document Service

Description: Blockchain Document Store

Figure 37   Entering values or creating a service ID

17. Enter the correct values and select **Create**. Successful service ID creation results are displayed in the window that is shown Figure 38.



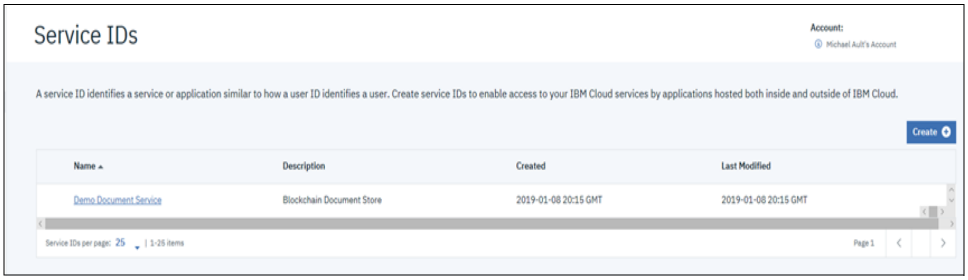
Manage Demo Document Service

ID: ServiceId-58ce2fd0-e2f8-433b-947d-5c06b2005ca3 **Description:** Blockchain Document Store

Create service ID The service ID was successfully created.

Figure 38 Creating the service ID

The Service ID option now shows the created ID. Clicking the ID returns to the display that is shown in Figure 39.



Service IDs

Account: Michael Ault's Account

A service ID identifies a service or application similar to how a user ID identifies a user. Create service IDs to enable access to your IBM Cloud services by applications hosted both inside and outside of IBM Cloud.

Name ▴	Description	Created	Last Modified
Demo Document Service	Blockchain Document Store	2019-01-08 20:15 GMT	2019-01-08 20:15 GMT

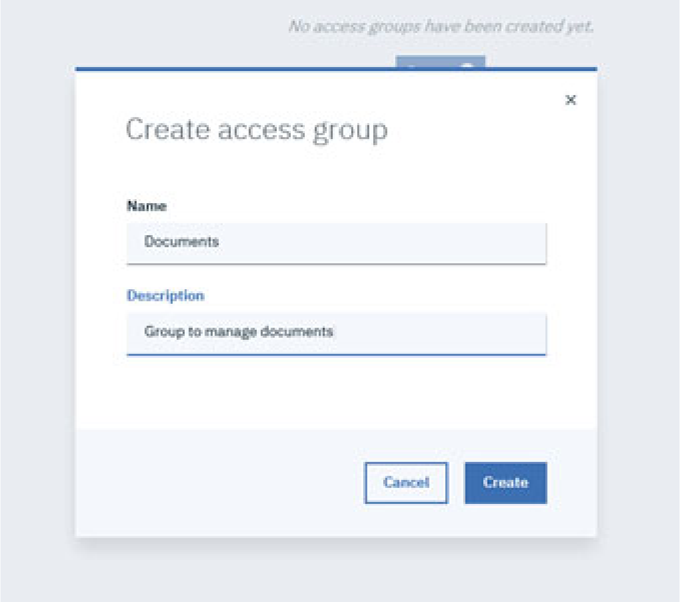
Service IDs per page: 25 | 1-25 items

Figure 39 Service ID display

From the display of the service ID, you need the service ID, as shown in the following example:

```
ServiceId-58ce2fd0-e2f8-433b-947d-5c06b2005ca3
```

18. Assign the service ID to an access group after an appropriate group is created if such a group does not exist. To create a group, browse to the **Access Groups** menu in the **Identity & Access** menu and click **Create** (see Figure 40).



No access groups have been created yet.

Create access group

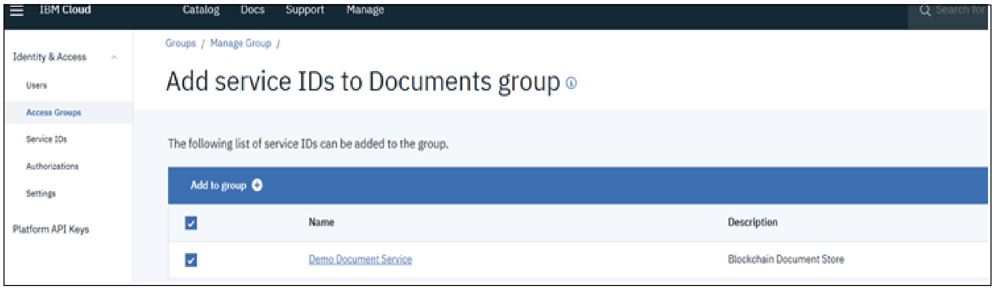
Name: Documents

Description: Group to manage documents

Figure 40 Creating an access group

19. Enter the necessary information for your group and select **Create**.

20. Add the service ID to the group you created (see Figure 41).



Add service IDs to Documents group

The following list of service IDs can be added to the group.

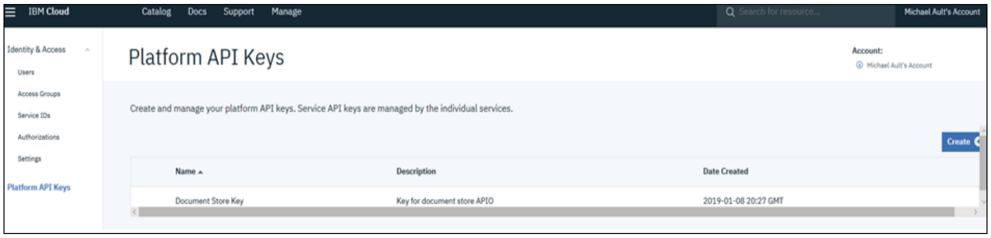
☑	Name	Description
☑	Demo Document Service	Blockchain Document Store

Figure 41 Adding service IDS to Documents Groups

21. Create the platform API key by selecting **Platform API Key** and clicking **Create**.

22. Select the created API Key to copy its value. In our example, the following key was created (see Figure 42):

    MkSR3Q8jaHDOjJg7hT9I7ta75w9eqHrnQ9GTA0-KSHvI



Platform API Keys

Create and manage your platform API keys. Service API keys are managed by the individual services.

Name ▲	Description	Date Created
Document Store Key	Key for document store APIO	2019-01-08 20:27 GMT

Figure 42 API key value

23. By using the keys that were generated, you can quickly authorize other applications to access your BDS by using the on and off boarding JSON that is provided in step 16 page 51.

    Use the following URL to use you are when accessing the BDS:

    `https://console.bluemix.net/dashboard/apps`

24. Log in by using your ID. The BDS instance is displayed. Click the instance to access the needed swagger URL.

25. To install this on a remote peer, you need Support to add the remote peer connection certification to the IBP in the cloud's connection JSON.

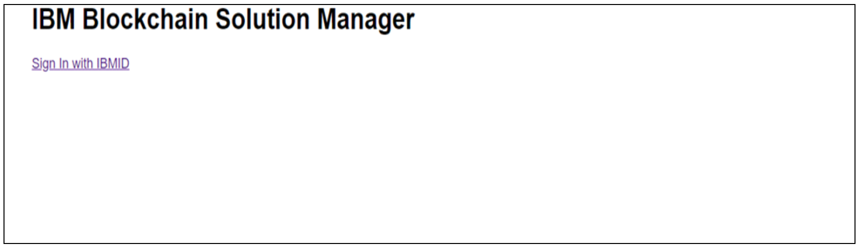
    The BDS is installed in each peer that requires access to the BDS.

Changing from Cloud COS Storage to Local, Onsite COS Storage for off-chain data

Many clients might not want to use IBM or other cloud storage. Instead, they want to use their own onsite storage. Blockchain Document Store uses IBM Cloud COS as a default. In this section, we review this feature. To begin, go to your BDS instance's solution manager website to get an organization admin token for the organization that had BDS installed (login required):

```
https://pbsa-prod.us-south.containers.mybluemix.net/703c9b46-3443-4a77-8cbf-7d5b55
5dbf37/onboarding/v1/logins?responseMode=undefined
```

After you enter in the URL for your instance, you see the main login window for the solution manager, as shown in Figure 43.



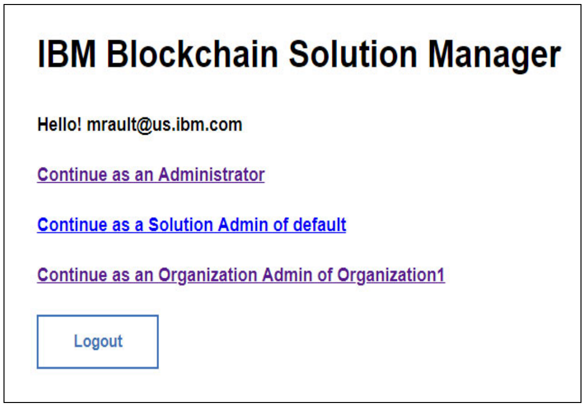
IBM Blockchain Solution Manager

Figure 43 Example login window for IBM Blockchain Solution Manager

From the main login window, complete the following steps:

1. Select **Sign in with IBMID**. The standard IBM login page opens in which you enter your IBM login ID and your password.

   After you are authorized, the IBM Blockchain Solution Manager menu is displayed, as shown in Figure 44.



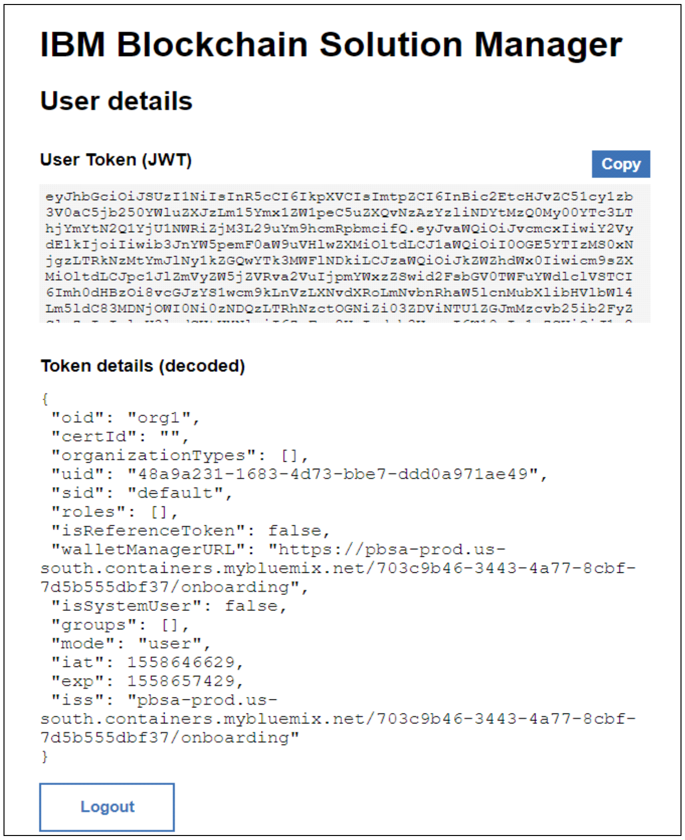
IBM Blockchain Solution Manager

Hello! mrault@us.ibm.com

Continue as an Administrator
Continue as a Solution Admin of default
Continue as an Organization Admin of Organization1
Logout

Figure 44 IBM Blockchain Solution Manager menu

2. For BDS use, select **Continue as an Organization Admin of Organization1**. The user token that is needed for the Swagger application that controls the low-level access to the BDS APIs is displayed, as shown in Figure 45.



IBM Blockchain Solution Manager

User details

User Token (JWT)

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InBic2EtcHJvZC51cy1zb
3V0aC5jb250YWluZXJzLm15Ymx1ZW1peC5uZXQvNzAzYzliNDYtMzQ0My00YTc3LT
hjYmYtN2Q1YjU1NWRiZjM3L29uYm9hcmRpbmcifQ.eyJvaWQiOiJvcmcxIiwiY2Vy
dElkIjoiIiwib3JnYW5pemF0aW9uVHlwZXMiOltdLCJ1aWQiOiI0OGE5YTIzMS0xN
jgzLTRkNzMtYmJlNy1kZGQwYTk3MWFlNDkiLCJzaWQiOiJkZWZhdWx0Iiwicm9sZX
MiOltdLCJpc1JlZmVyZW5jZVRva2VuIjpmYWxzZSwid2FsbGV0TWFuYWdlclVSTCI
6Imh0dHBzOi8vcGJzYS1wcm9kLnVzLXNvdXRoLmNvbnRhaW5lcnMubXlibHVlbWl4
Lm5ldC83MDNjOWI0Ni0zNDQzLTRhNzctOGNiZi03ZDViNTU1ZGJmMzcvb25ib2FyZ
```

Token details (decoded)

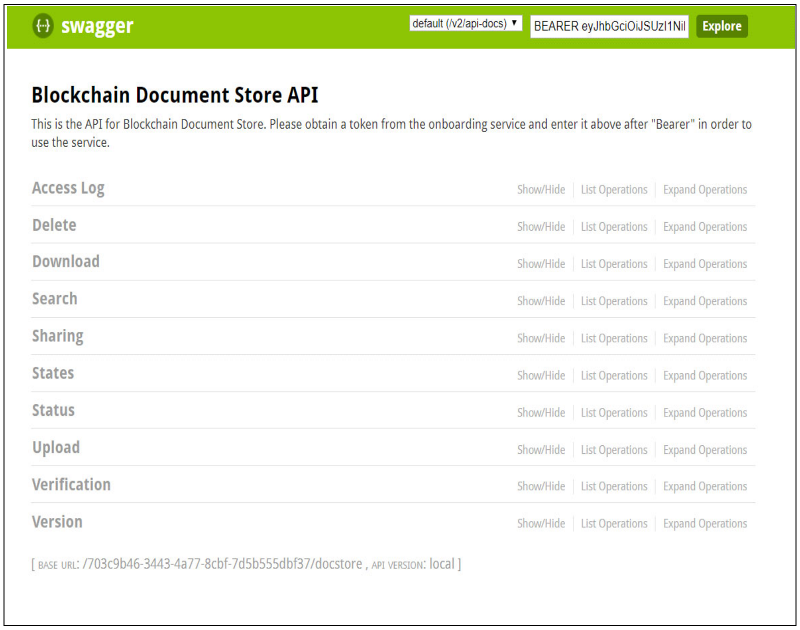
```
{
 "oid": "org1",
 "certId": "",
 "organizationTypes": [],
 "uid": "48a9a231-1683-4d73-bbe7-ddd0a971ae49",
 "sid": "default",
 "roles": [],
 "isReferenceToken": false,
 "walletManagerURL": "https://pbsa-prod.us-
south.containers.mybluemix.net/703c9b46-3443-4a77-8cbf-
7d5b555dbf37/onboarding",
 "isSystemUser": false,
 "groups": [],
 "mode": "user",
 "iat": 1558646629,
 "exp": 1558657429,
 "iss": "pbsa-prod.us-
south.containers.mybluemix.net/703c9b46-3443-4a77-8cbf-
7d5b555dbf37/onboarding"
}
```

Figure 45   User Token window

3. Select **Copy** from the User Token menu to copy the JWT token. Start your BDS instance Swagger application by using the provided URL. The following URL was used in our example:

   `https://pbsa-prod.us-south.containers.mybluemix.net/703c9b46-3443-4a77-8cbf-7d5b555dbf37/docstore/swagger-ui.html#/`

Successful startup of the Swagger application is shown in Figure 46.



Blockchain Document Store API

This is the API for Blockchain Document Store. Please obtain a token from the onboarding service and enter it above after "Bearer" in order to use the service.

Access Log Show/Hide | List Operations | Expand Operations

Delete Show/Hide | List Operations | Expand Operations

Download Show/Hide | List Operations | Expand Operations

Search Show/Hide | List Operations | Expand Operations

Sharing Show/Hide | List Operations | Expand Operations

States Show/Hide | List Operations | Expand Operations

Status Show/Hide | List Operations | Expand Operations

Upload Show/Hide | List Operations | Expand Operations

Verification Show/Hide | List Operations | Expand Operations

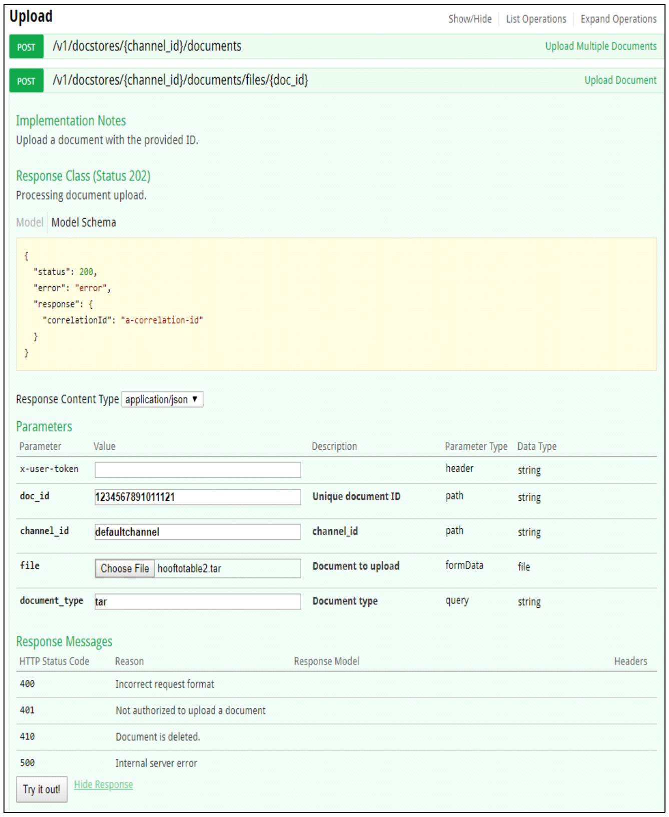
Version Show/Hide | List Operations | Expand Operations

[BASE URL: /703c9b46-3443-4a77-8cbf-7d5b555dbf37/docstore , API VERSION: local]

Figure 46 Swagger application for BDS APIs

4. Copy the JWT token into the space after the word “Bearer”. Do not press Enter, or you must copy the token again.

5. To test if we are attached to the Cloud COS, we upload a file. Select the **Upload** option. Then, select the option for Uploading a single document. A window opens, as shown in the example, in Figure 47.



POST /v1/docstores/{channel_id}/documents

POST /v1/docstores/{channel_id}/documents/files/{doc_id}

Implementation Notes

Upload a document with the provided ID.

Response Class (Status 202)

Processing document upload.

```
{
  "status": 200,
  "error": "error",
  "response": {
    "correlationId": "a-correlation-id"
  }
}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-user-token			header	string
doc_id	1234567891011121	Unique document ID	path	string
channel_id	defaultchannel	channel_id	path	string
file	Choose File hooftotable2.tar	Document to upload	formData	file
document_type	tar	Document type	query	string

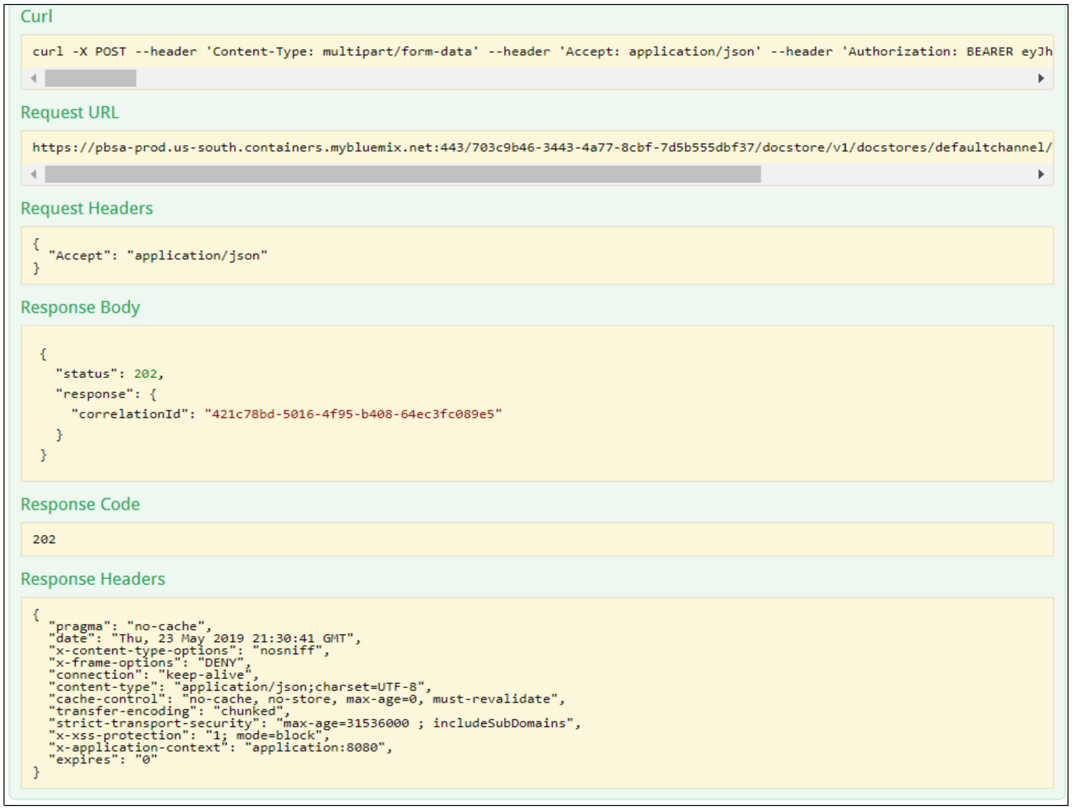
Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Incorrect request format		
401	Not authorized to upload a document		
410	Document is deleted.		
500	Internal server error		

Figure 47 Swagger BDS Upload display

6. To upload a file, you need a unique document identifier (because this document is the first document, any identifier is unique), the document type, the channel ID, and knowledge where the document is stored. The entries should not have quotes around them.

Select **Try it out** after entering the correct information. If you are successful, you receive a 200 series response, as shown in Figure 48.



Curl

```
curl -X POST --header 'Content-Type: multipart/form-data' --header 'Accept: application/json' --header 'Authorization: BEARER eyJh
```

Request URL

```
https://pbsa-prod.us-south.containers.mybluemix.net:443/703c9b46-3443-4a77-8cbf-7d5b555dbf37/docstore/v1/docstores/defaultchannel/
```

Request Headers

```
{
  "Accept": "application/json"
}
```

Response Body

```
{
  "status": 202,
  "response": {
    "correlationId": "421c78bd-5016-4f95-b408-64ec3fc089e5"
  }
}
```

Response Code

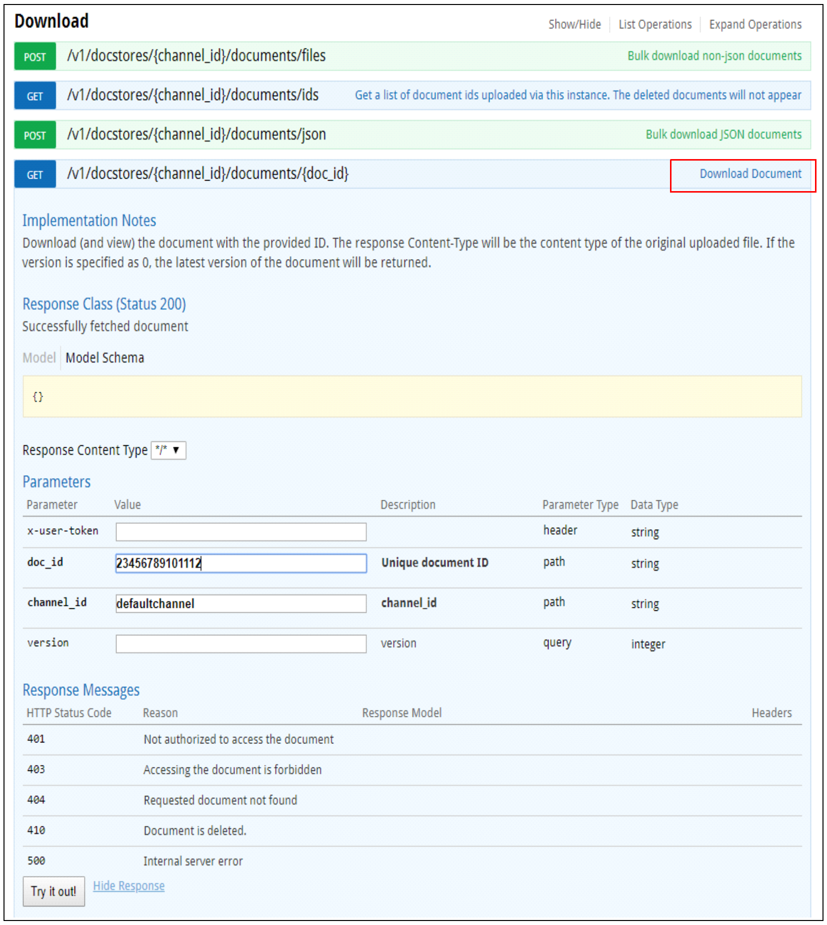
```
202
```

Response Headers

```
{
  "pragma": "no-cache",
  "date": "Thu, 23 May 2019 21:30:41 GMT",
  "x-content-type-options": "nosniff",
  "x-frame-options": "DENY",
  "connection": "keep-alive",
  "content-type": "application/json;charset=UTF-8",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "transfer-encoding": "chunked",
  "strict-transport-security": "max-age=31536000 ; includeSubDomains",
  "x-xss-protection": "1; mode=block",
  "x-application-context": "application:8080",
  "expires": "0"
}
```

Figure 48 Successful file upload

The easiest way to verify the document is in the Cloud COS is to use the **Download Document** function, shown in Figure 49.



Download

POST /v1/docstores/{channel_id}/documents/files — Bulk download non-json documents

GET /v1/docstores/{channel_id}/documents/ids — Get a list of document ids uploaded via this instance. The deleted documents will not appear

POST /v1/docstores/{channel_id}/documents/json — Bulk download JSON documents

GET /v1/docstores/{channel_id}/documents/{doc_id} — Download Document

Implementation Notes

Download (and view) the document with the provided ID. The response Content-Type will be the content type of the original uploaded file. If the version is specified as 0, the latest version of the document will be returned.

Response Class (Status 200)

Successfully fetched document

Model | Model Schema

```
{}
```

Response Content Type */* ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-user-token			header	string
doc_id	23456789101112	Unique document ID	path	string
channel_id	defaultchannel	channel_id	path	string
version		version	query	integer

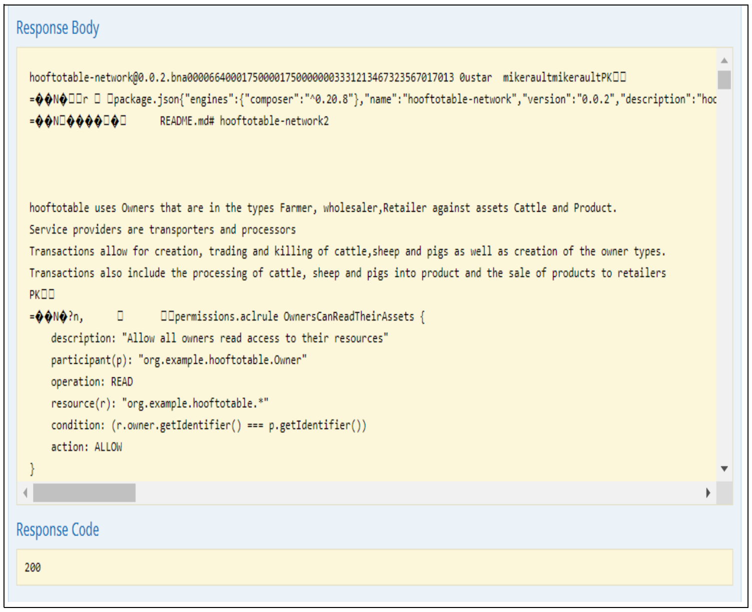
Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Not authorized to access the document		
403	Accessing the document is forbidden		
404	Requested document not found		
410	Document is deleted.		
500	Internal server error		

Figure 49 Download single document

7. Enter the document ID that you assigned and the channel name. Then, select **Try it out**.

   If the document is successfully uploaded, you receive a successful download message, as shown in Figure 50.



```
hooftotable-network@0.0.2.bna00006640001750000175000000003331213467323567017013 0ustar  mikeraultmikeraultPK□□
=��N�□□r □ □package.json{"engines":{"composer":"^0.20.8"},"name":"hooftotable-network","version":"0.0.2","description":"hoc
=��N□�����□�□      README.md# hooftotable-network2




hooftotable uses Owners that are in the types Farmer, wholesaler,Retailer against assets Cattle and Product.
Service providers are transporters and processors
Transactions allow for creation, trading and killing of cattle,sheep and pigs as well as creation of the owner types.
Transactions also include the processing of cattle, sheep and pigs into product and the sale of products to retailers
PK□□
=��N�?n,      □      □□permissions.aclrule OwnersCanReadTheirAssets {
    description: "Allow all owners read access to their resources"
    participant(p): "org.example.hooftotable.Owner"
    operation: READ
    resource(r): "org.example.hooftotable.*"
    condition: (r.owner.getIdentifier() === p.getIdentifier())
    action: ALLOW
}
```

```
200
```

Figure 50   Successful download from cloud

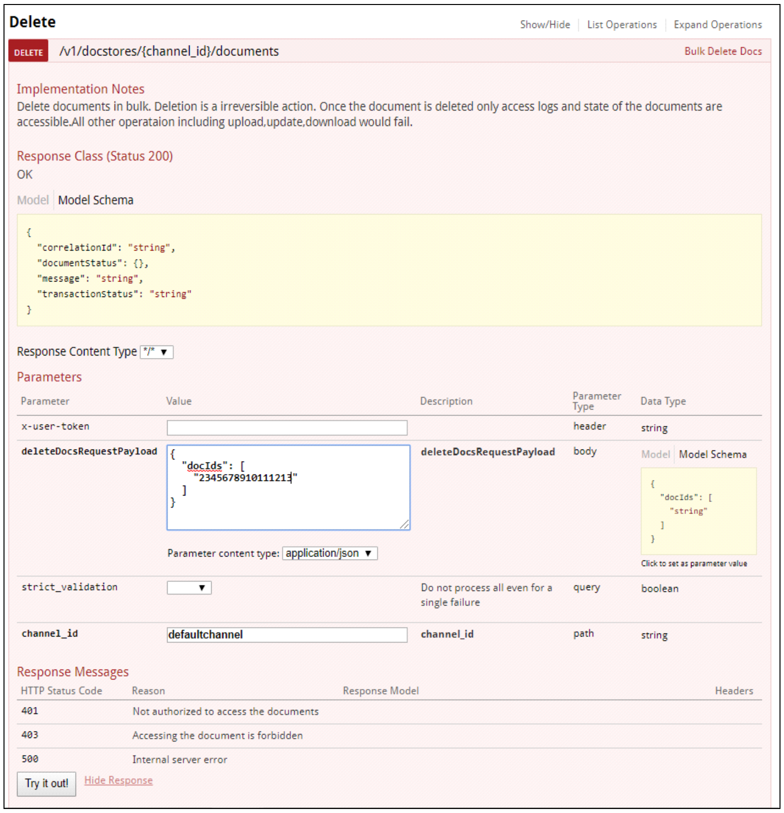
This message demonstrates that Cloud-based COS for BDS is working.

But, what about local?

We must edit the underlying json document that provides the storage location information for BDS. This process is done by BDS Support through a Support ticket. The new vault is in an onsite COS environment and uses the vault information that is provided by the COS administrator. That information is sent to the BDS support team.

So as not to leave orphan records, we delete the record we created. Because this is blockchain, the path of creation, listing, and deletion is still recorded in the blockchain; however, the off-chain record is deleted and the blockchain is updated to reflect its deleted status.

To delete a document, use the Delete set of APIs (the single document deletion), as shown in Figure 51.



Delete

Show/Hide | List Operations | Expand Operations

DELETE /v1/docstores/{channel_id}/documents — Bulk Delete Docs

Implementation Notes

Delete documents in bulk. Deletion is a irreversible action. Once the document is deleted only access logs and state of the documents are accessible.All other operataion including upload,update,download would fail.

Response Class (Status 200)

OK

Model | Model Schema

```
{
  "correlationId": "string",
  "documentStatus": {},
  "message": "string",
  "transactionStatus": "string"
}
```

Response Content Type: */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
x-user-token			header	string

deleteDocsRequestPayload	{
	"docIds": [
	"2345678910111213"
	]
	}
	Parameter content type: application/json
		deleteDocsRequestPayload	body	Model | Model Schema
				{
				"docIds": [
				"string"
				]
				}
				Click to set as parameter value

strict_validation		Do not process all even for a single failure	query	boolean
channel_id	defaultchannel	channel_id	path	string

Response Messages

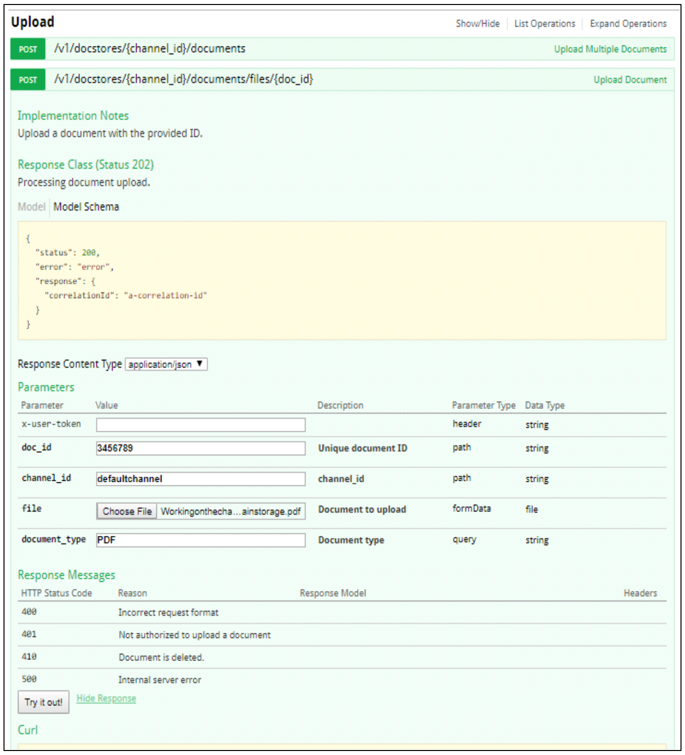
HTTP Status Code	Reason	Response Model	Headers
401	Not authorized to access the documents		
403	Accessing the document is forbidden		
500	Internal server error		

Figure 51   Delete Document API

After a document is deleted, any further requests result in a 410 status that says the document or file is deleted. The status check still indicates a 200 status because a record of the file is still available, but the file no longer exists in the BDS.

To test the capability to use a new onsite COS linkage, a COS vault was established on a remote COS appliance and the login information was sent to the BDS team. The BDS team re-pointed the internal credentials and address of the vault being used by the BDS instance in our Hyperledger blockchain.

What we expect to see is that it does not find the document we tested with the cloud-based COS, but does allow us to load a new document. Figure 52 shows our request to upload a new document.



Upload

POST /v1/docstores/{channel_id}/documents — Upload Multiple Documents

POST /v1/docstores/{channel_id}/documents/files/{doc_id} — Upload Document

Implementation Notes

Upload a document with the provided ID.

Response Class (Status 202)

Processing document upload.

```
{
  "status": 200,
  "error": "error",
  "response": {
    "correlationId": "a-correlation-id"
  }
}
```

Response Content Type application/json ▼

Parameters

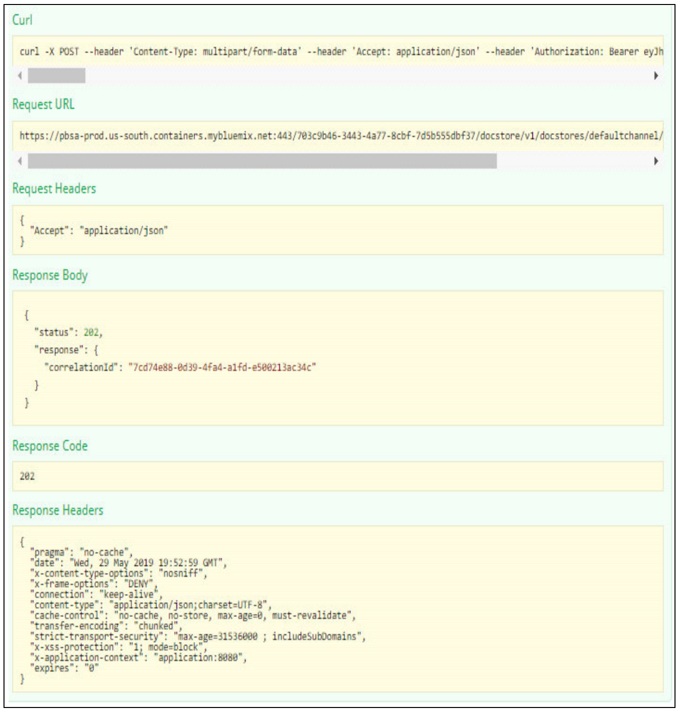
Parameter	Value	Description	Parameter Type	Data Type
x-user-token			header	string
doc_id	3456789	Unique document ID	path	string
channel_id	defaultchannel	channel_id	path	string
file	Choose File Workingonthecha…ainstorage.pdf	Document to upload	formData	file
document_type	PDF	Document type	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Incorrect request format		
401	Not authorized to upload a document		
410	Document is deleted.		
500	Internal server error		

Figure 52 Attempt to upload into the new COS onsite storage

The successful upload results are shown in Figure 53.



Curl

```
curl -X POST --header 'Content-Type: multipart/form-data' --header 'Accept: application/json' --header 'Authorization: Bearer eyJh
```

Request URL

```
https://pbsa-prod.us-south.containers.mybluemix.net:443/703c9b46-3443-4a77-8cbf-7d5b555dbf37/docstore/v1/docstores/defaultchannel/
```

Request Headers

```
{
  "Accept": "application/json"
}
```

Response Body

```
{
  "status": 202,
  "response": {
    "correlationId": "7cd74e88-0d39-4fa4-a1fd-e500213ac34c"
  }
}
```

Response Code

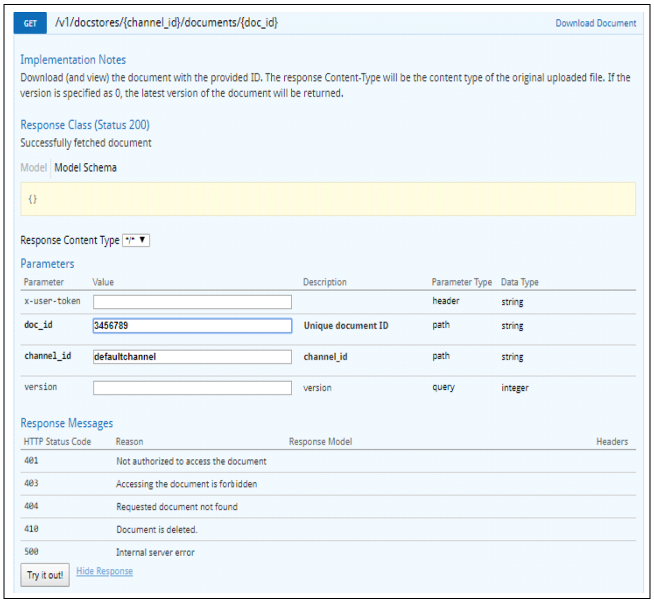
```
202
```

Response Headers

```
{
  "pragma": "no-cache",
  "date": "Wed, 29 May 2019 19:52:59 GMT",
  "x-content-type-options": "nosniff",
  "x-frame-options": "DENY",
  "connection": "keep-alive",
  "content-type": "application/json;charset=UTF-8",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "transfer-encoding": "chunked",
  "strict-transport-security": "max-age=31536000 ; includeSubDomains",
  "x-xss-protection": "1; mode=block",
  "x-application-context": "application:8080",
  "expires": "0"
}
```

Figure 53 Successful document upload

As you can see by the 200 series Response Code, the file was successfully uploaded to the new onsite location, just as it is if we were pointed to the cloud COS instead. To verify it was loaded, we download it. Figure 54 shows the download request.



GET /v1/docstores/{channel_id}/documents/{doc_id}

Implementation Notes

Download (and view) the document with the provided ID. The response Content-Type will be the content type of the original uploaded file. If the version is specified as 0, the latest version of the document will be returned.

Response Class (Status 200)

Successfully fetched document

```
{}
```

Response Content Type */*

Parameters

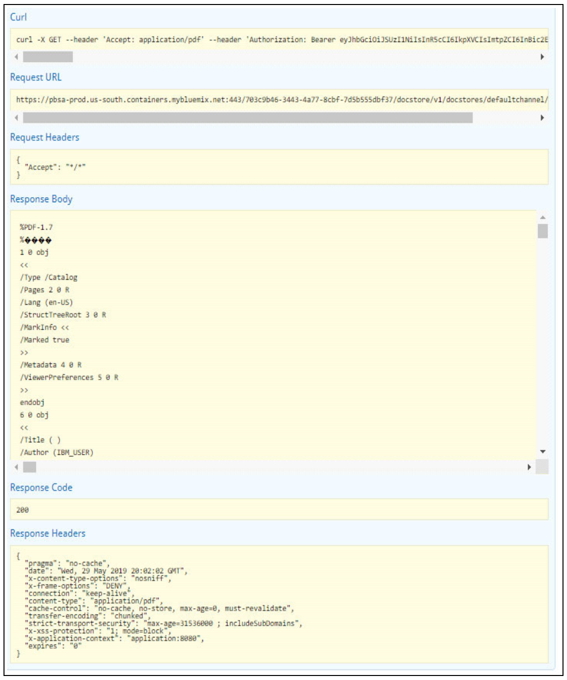
Parameter	Value	Description	Parameter Type	Data Type
x-user-token			header	string
doc_id	3456789	Unique document ID	path	string
channel_id	defaultchannel	channel_id	path	string
version		version	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Not authorized to access the document		
403	Accessing the document is forbidden		
404	Requested document not found		
410	Document is deleted.		
500	Internal server error		

Figure 54 Download of single document or file

The download results are shown in Figure 55.



Curl

```
curl -X GET --header 'Accept: application/pdf' --header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InBic2E
```

Request URL

```
https://pbsa-prod.us-south.containers.mybluemix.net:443/703c9b46-3443-4a77-8cbf-7d5b555dbf37/docstore/v1/docstores/defaultchannel/
```

Request Headers

```
{
  "Accept": "*/*"
}
```

Response Body

```
%PDF-1.7
%����
1 0 obj
<<
/Type /Catalog
/Pages 2 0 R
/Lang (en-US)
/StructTreeRoot 3 0 R
/MarkInfo <<
/Marked true
>>
/Metadata 4 0 R
/ViewerPreferences 5 0 R
>>
endobj
6 0 obj
<<
/Title ( )
/Author (IBM_USER)
```

Response Code

```
200
```

Response Headers

```
{
  "pragma": "no-cache",
  "date": "Wed, 29 May 2019 20:02:02 GMT",
  "x-content-type-options": "nosniff",
  "x-frame-options": "DENY",
  "connection": "keep-alive",
  "content-type": "application/pdf",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "transfer-encoding": "chunked",
  "strict-transport-security": "max-age=31536000 ; includeSubDomains",
  "x-xss-protection": "1; mode=block",
  "x-application-context": "application:8080",
  "expires": "0"
}
```

Figure 55 Successful download of the uploaded document

As you can see in Figure 55, the first few lines of our PDF file are shown in the Response Body and the Response Code shows a 200 which shows success.

In this section, we showed that Hyperledger blockchain can be used successfully with off-chain storage in the cloud or with local onsite storage. As with any type of blockchain on or off-chain storage, you must be sure that it is globally accessible to all members of the blockchain network that might need to see the data that is stored there.

Summary

This Blueprint delivers an end-to-end blockchain infrastructure that is ready for any blockchain implementation.

Clients are not locked into one version of an application stack because the solution uses open industry standards. Instead, clients can pick and choose the open source Hyperledger-based solution that is best for their environment.

IBM Blockchain Protocol on IBM Cloud Private with the IBM Blockchain Document Store provides clients with an enterprise-grade on-premises cloud stack that is enabled by IBM compute and storage infrastructure. With this IBM Storage Solution for Blockchain, clients can rest easy knowing that their data is within their control and that their solution allows them to use blockchain related services and manage operational expenses within the confines of their environment.

For more information

For more information, see the following resources:

- How to get the benefits of cloud behind your firewall: IBM Cloud Private:

  https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=KUW12527USEN
- IBM FlashSystem 9100:

  https://www.ibm.com/us-en/marketplace/flashsystem-9100
- IBM Redbooks: Implementing the IBM Storwize V7000 and IBM Spectrum Virtualize V7.8:

  https://www.redbooks.ibm.com/redbooks/pdfs/sg247938.pdf
- IBM Redbooks: VersaStack Solution for File Storage Using IBM Storwize V5030 and Windows Server 2016:

  http://www.redbooks.ibm.com/redpapers/pdfs/redp5442.pdf
- IBM Spectrum Connect:

  https://www.ibm.com/support/knowledgecenter/en/SS6JWS/landing/IBM_Spectrum_Connect_welcome_page.html
- IBM Blockchain:

  https://www.ibm.com/blockchain
- IBM Cloud Object Storage:

  https://www.ibm.com/cloud/object-storage
- IBM DS8880 and IBM Z Synergy:

  http://www.redbooks.ibm.com/redpieces/abstracts/redp5186.html?Open

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

June 2019

ISBN 0738457817