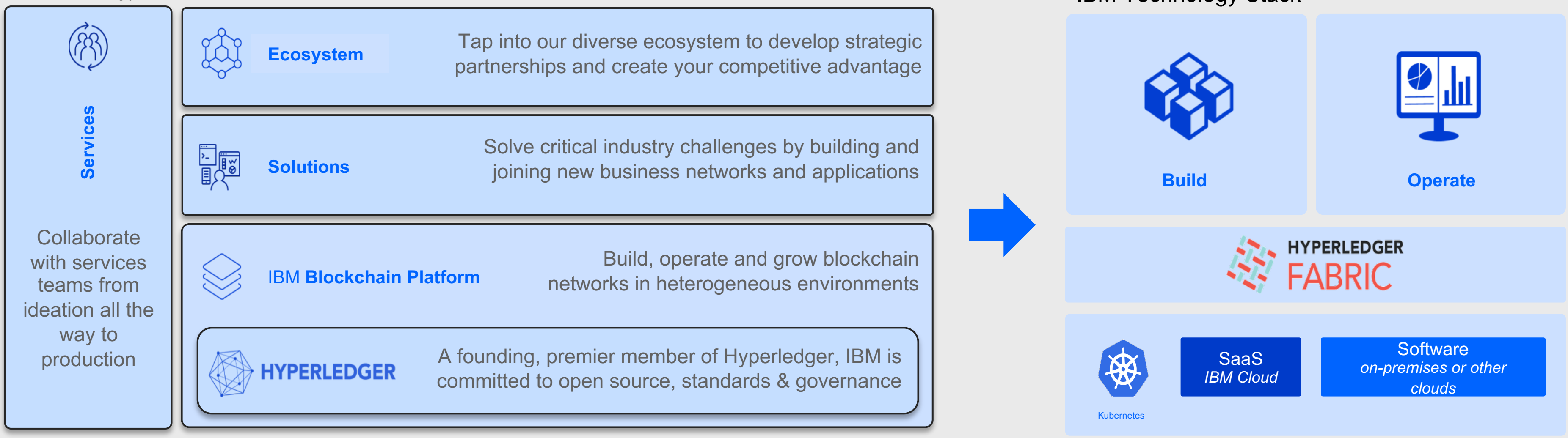
Hyperledger Fabric 2.0 : What’s new?

Maciej Jędrzejczyk
CEE Blockchain Technical Leader, IBM
http://ibm.biz/maciej-jedrzejczyk

Quick Recap of IBM Blockchain



IBM Strategy

Services

Collaborate with services teams from ideation all the way to production

Ecosystem

Tap into our diverse ecosystem to develop strategic partnerships and create your competitive advantage

Solutions

Solve critical industry challenges by building and joining new business networks and applications

IBM **Blockchain Platform**

Build, operate and grow blockchain networks in heterogeneous environments

HYPERLEDGER

A founding, premier member of Hyperledger, IBM is committed to open source, standards & governance

IBM Technology Stack

Build

Operate

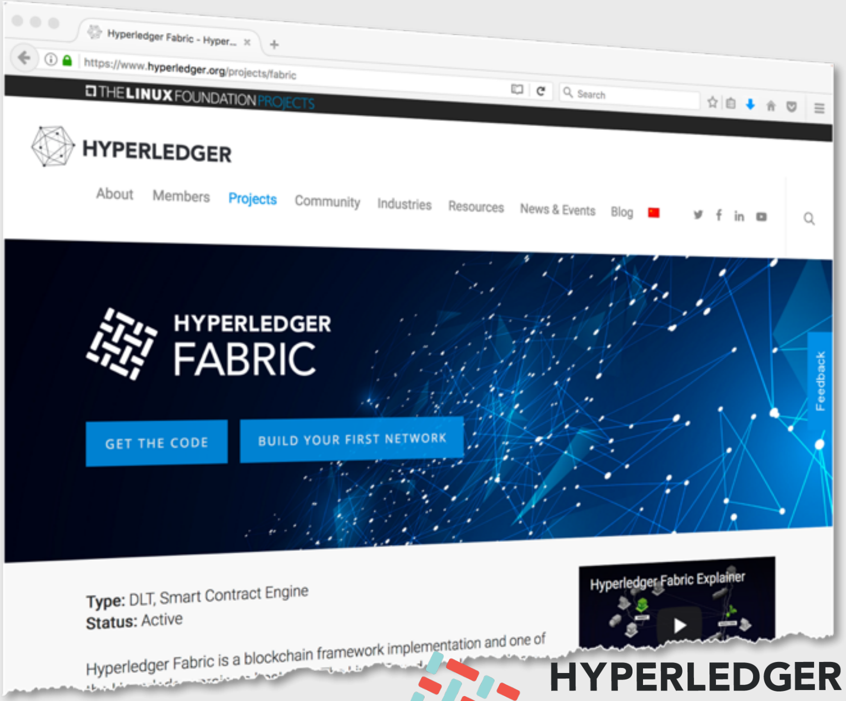
HYPERLEDGER FABRIC

Kubernetes

SaaS
IBM Cloud

Software
on-premises or other clouds

What is Hyperledger Fabric?



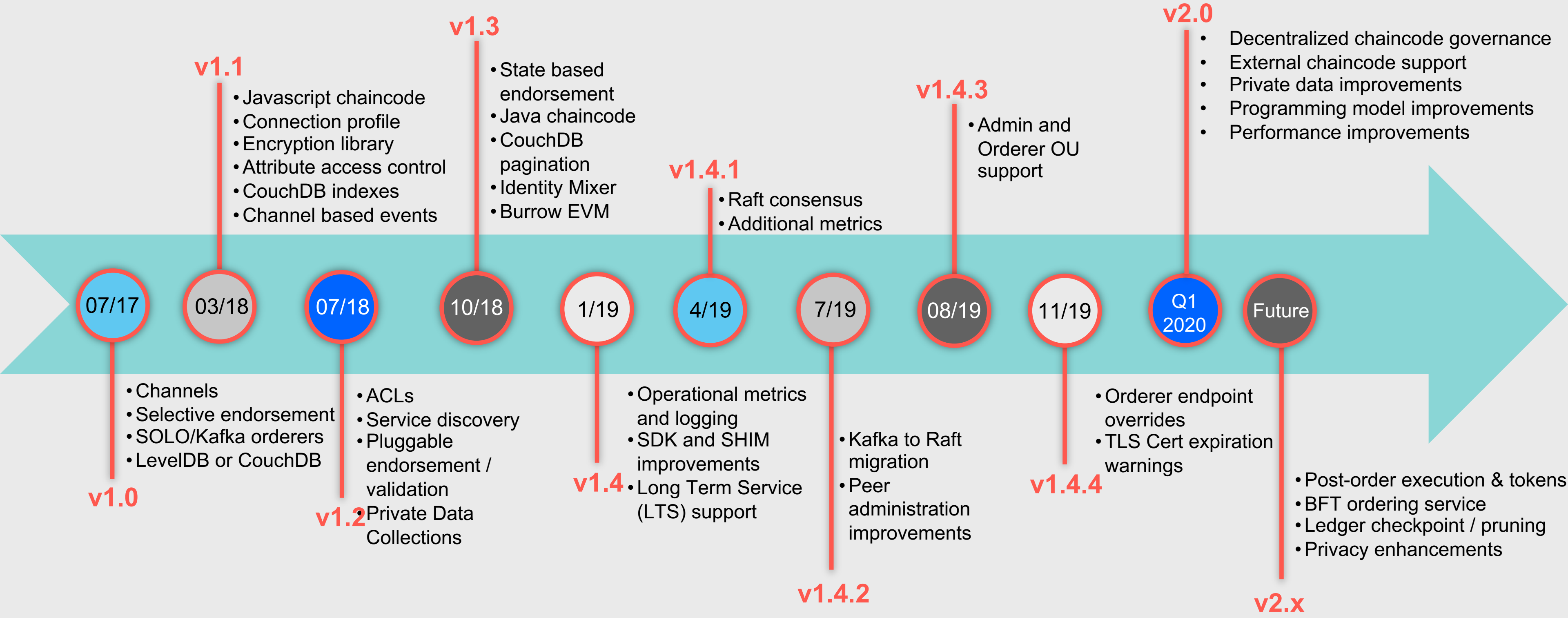
Type: DLT, Smart Contract Engine
Status: Active

Hyperledger Fabric is a blockchain framework implementation and one of

HYPERLEDGER
FABRIC

- An implementation of blockchain technology that is a foundation for developing blockchain applications for business
- Emphasis on ledger, smart contracts, consensus, confidentiality, resiliency and scalability.
- V2.0.1 released February 2020
- V1.4 Long Term Service release with emphasis on production operational and serviceability enhancements; new programming model abstractions for ease of development
- IBM is one of the many contributing organizations

Hyperledger Fabric Roadmap



07/17 – v1.0
- Channels
- Selective endorsement
- SOLO/Kafka orderers
- LevelDB or CouchDB

03/18 – v1.1
- Javascript chaincode
- Connection profile
- Encryption library
- Attribute access control
- CouchDB indexes
- Channel based events

07/18 – v1.2
- ACLs
- Service discovery
- Pluggable endorsement / validation
- Private Data Collections

10/18 – v1.3
- State based endorsement
- Java chaincode
- CouchDB pagination
- Identity Mixer
- Burrow EVM

1/19 – v1.4
- Operational metrics and logging
- SDK and SHIM improvements
- Long Term Service (LTS) support

4/19 – v1.4.1
- Raft consensus
- Additional metrics

7/19 – v1.4.2
- Kafka to Raft migration
- Peer administration improvements

08/19 – v1.4.3
- Admin and Orderer OU support

11/19 – v1.4.4
- Orderer endpoint overrides
- TLS Cert expiration warnings

Q1 2020 – v2.0
- Decentralized chaincode governance
- External chaincode support
- Private data improvements
- Programming model improvements
- Performance improvements

Future – v2.x
- Post-order execution & tokens
- BFT ordering service
- Ledger checkpoint / pruning
- Privacy enhancements

Based on https://wiki.hyperledger.org/display/fabric/Hyperledger+Fabric+Roadmap - Dates determined by the Hyperledger community - (*) Subject to change

What’s New in Hyperledger Fabric 2.0?

In this release, Hyperledger Fabric developers **focused on**:

- Chaincode lifecycle
- Private Data Collections enhancements
- Security and performance improvements

What this release **does not** introduce:

- Token management model
- BFT consensus
- Upgrades to Fabric-CA and zero-knowledge proofs

Warning! Hyperledger Fabric 2.0 **is not** an LTS release! LTS tag has not been dropped from 1.4.x yet.

What’s new in chaincode

What’s new in chaincode

Decentralized governance brings new patterns for collaboration and consensus

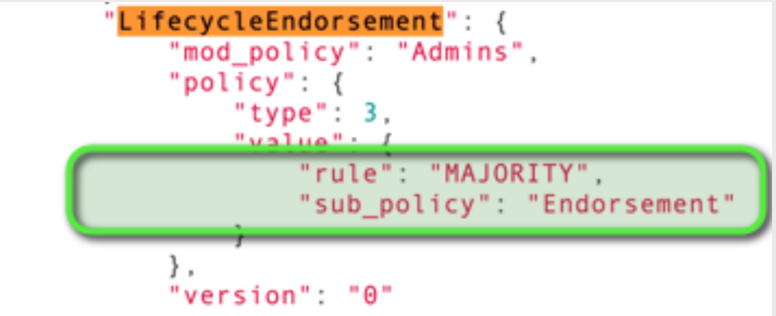
Multiple organizations must agree to the parameters of a chaincode

New channel policy (LifecycleEndorsement) which specifies the set of organizations required to approve a chaincode definition on a channel.

Committing a channel definition requires endorsement to fulfill LifecycleEndorsement policy

Number of organizations needed to approve definition is governed by
the Channel/Application/LifecycleEndorsement policy

Chaincode definition includes: Endorsement policy, Private data collection configurations, etc



```
"LifecycleEndorsement": {
    "mod_policy": "Admins",
    "policy": {
        "type": 3,
        "value": {
            "rule": "MAJORITY",
            "sub_policy": "Endorsement"
        }
    },
    "version": "0"
```

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/enable_cc_lifecycle.html#create-enable-lifecycle-json

New Chaincode Lifecycle Flow - Installation

Introduces a more deliberate chaincode installation and upgrade process

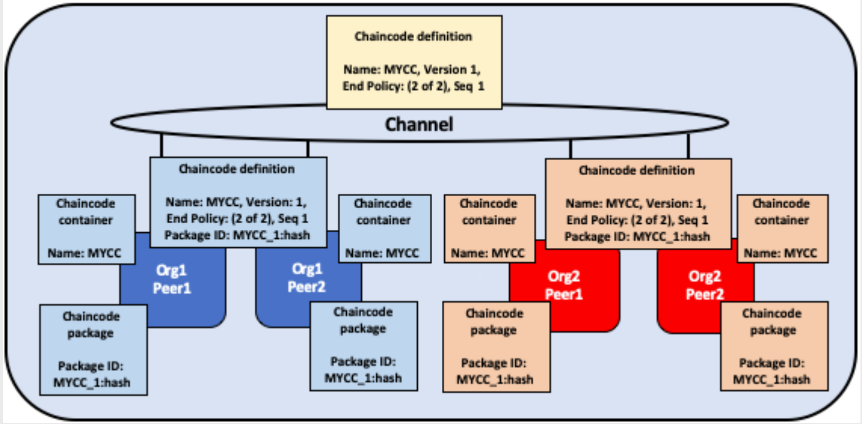
Each organization packages/installs chaincode to their peers

Individual organizations each call lifecycle system chaincode to approve a chaincode definition

```
$ peer lifecycle chaincode approveformyorg --init-required ...
```

When sufficient number of organizations have approved, one organization calls lifecycle system chaincode to commit the chaincode definition to the channel

```
$ peer lifecycle chaincode commit ...
```



More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode_lifecycle.html#upgrade-a-chaincode

What’s new in chaincode

New Chaincode Lifecycle Flow - Upgrade

Chaincode can be upgraded using the same process as used to install for the first time

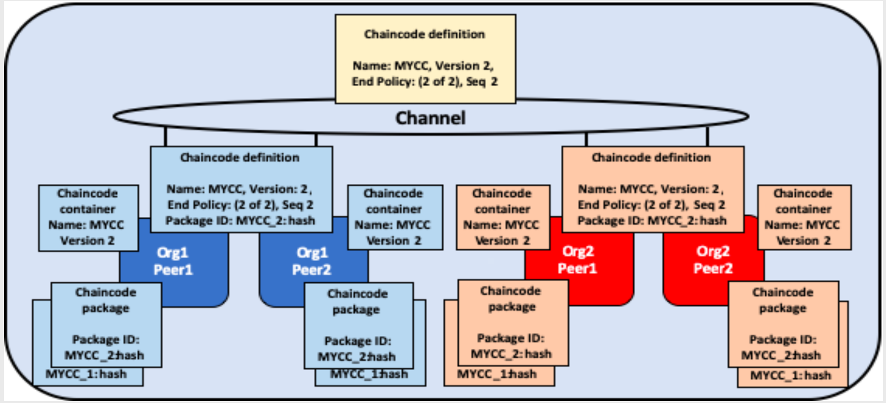
1. Repackage the chaincode (Only if chaincode updated)

2. Install the new chaincode package on peers (Only if chaincode updated)

3. Approve a new chaincode definition

4. Commit the definition to the channel

5. Upgrade the chaincode container (Automatic - Only if chaincode updated)



More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode_lifecycle.html#upgrade-a-chaincode

What’s new in chaincode

New Chaincode Lifecycle Flow – Deployment Scenarios

Joining a channel

- New organization can start using chaincode after joining an existing channel
- No need to be commit definition again
- When using default LifecycleEndorsement policy, number of required approvals/endorsements will change automatically.

Updating an endorsement policy

- Can be done without repackaging/re-installing chaincode
- Will take effect after the new definition is committed
- No need to restart the chaincode container

Approving definition without installed chaincode

- Organizations can approve a chaincode definition without installing the chaincode package
- No need to provide a packageID as part of chaincode definition

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode_lifecycle.html#upgrade-a-chaincode

New Chaincode Lifecycle Flow – Conflict Scenarios

Organization disagrees on chaincode definition

- Organization cannot use chaincode unless definition is approved
- Organizations with different chaincode definition will not be able to execute the chaincode on their peers
- Other organizations in the channel can use chaincode for approved definition

Channel does not agree on a definition

- Happens when organizations approve definitions that do not match
- Definition cannot be committed to the channel
- None of the channel members will be able to use the chaincode

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode_lifecycle.html#upgrade-a-chaincode

What’s new in chaincode

Chaincode packages are inspectable

Package in a tar file, ending with a .tar.gz file extension

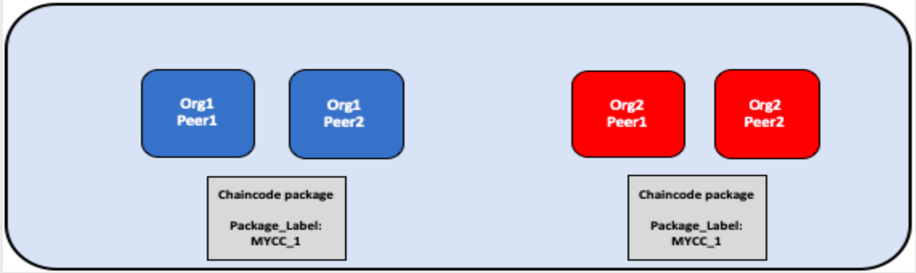
Two files: A metadata file “Chaincode-Package-Metadata.json” and tar containing the chaincode files

Metadata specifies the chaincode language, code path, and package label

It is not necessary for organizations to use the same package label

```
$ peer lifecycle chaincode package --label ...
```

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/commands/peerchaincode.html#peer-chaincode-package



What’s new in chaincode

Chaincode logic on peer and chaincode definition on channel are decoupled

The previous lifecycle defined each chaincode on the channel using a name and version that was specified when the chaincode package was installed. Chaincodes had to have identical logic and policies.

Multiple definitions can be used to run different instances of chaincode. Same chaincode can have multiple endorsement policies.

Ability to update chaincode definition (endorsement policy, collection configuration, etc) on the channel, without having to install new chaincode on each peer

Chaincode package and chaincode language installed on peer can now be different for each organization, so long as the ultimate chaincode execution results (RW sets) are the same on the required endorsers

Use Case #1: Each organization can now roll out minor fixes on their own schedules without requiring the entire network to proceed in lock-step.

Use Case #2: You can now use a single chaincode package and deploy it multiple times with different names on the same channel or on different channels to track different types of assets in their own chaincode namespace.

What’s new in chaincode

External chaincode support

Eliminate Docker daemon dependency – Previously peers were required to have access to a Docker daemon to build and launch chaincode, requiring the peer to have process privileges

Users can choose between running chaincode in a docker container (default) or using the external chaincode launcher feature to build and launch chaincode with the technology of their choice.

Chaincodes can now run as an external service (e.g. Kubernetes pod) which the peer connects to

Alternatives to containers – Chaincode no longer required to run in Docker containers, and instead can be run in the user’s choice of environment (including containers)

External builder executables – A user provides a set of external builder executables to override how the peer prepares and launches chaincode

Location of external builder executables are configured in peer core.yaml.

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/cc_launcher.html

What's new in chaincode

Other enhancements

Support for Node.js 12 and Java 11 for smart contracts
Moved to new Long Term Support release allowing smart contract developers to take advantage of the latest language features and performance enhancements
Investigating support for Node.js and Java SDKs

High-level programming model for Go smart contracts
Consistent experience across all supported languages, for smart contract developers who wish to use Go, including updated samples, tutorials, and documentation

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/whatsnew.html

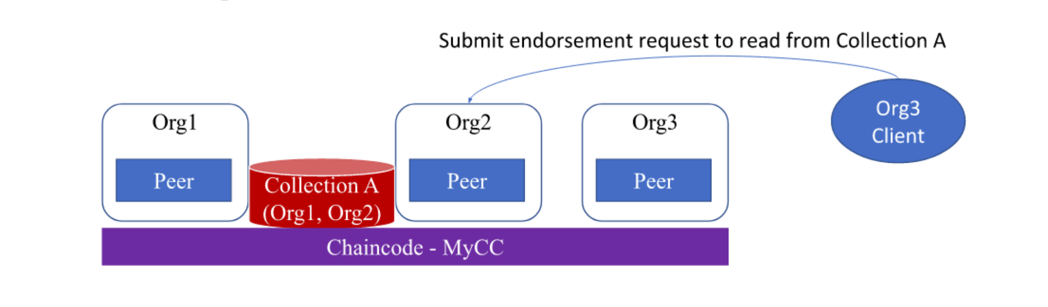
What’s new in private data collections

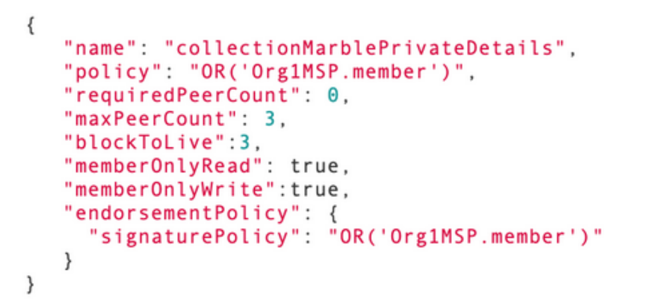
What’s new in private data collections

Read-Write access control on collections

Previously, any channel member could read-write to a collection if the chaincode did not have an adequate ACL

Now, the collection has memberonlyread and memberonlywrite attributes that can control that behaviour





```
{
    "name": "collectionMarblePrivateDetails",
    "policy": "OR('Org1MSP.member')",
    "requiredPeerCount": 0,
    "maxPeerCount": 3,
    "blockToLive":3,
    "memberOnlyRead": true,
    "memberOnlyWrite":true,
    "endorsementPolicy": {
      "signaturePolicy": "OR('Org1MSP.member')"
    }
}
```

More: http://www.bchainledger.com/2020/03/whats-new-in-hyperledger-fabric-v20.html

What’s new in private data collections

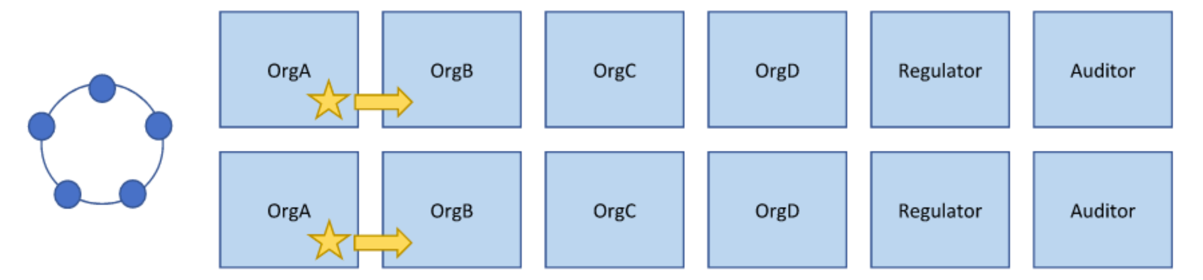
Sharing and verifying private data

Utilize implicit built-in private data collections per organization, no need to define

Each per-organization collection has an endorsement policy matching the organization
e.g. PutPrivateData(“_implicit_org_<MSPID>”, key, value)

Implicit per-organization private data collections allow for sharing private data at the key level on a need-to-know basis, e.g. when transferring an asset from one organization to another

Share and verify private data with new GetPrivateDataHash() chaincode API to verify on-chain hashes or when private data moves from one organization to another.



More: http://www.bchainledger.com/2020/03/whats-new-in-hyperledger-fabric-v20.html

What’s new in private data collections

Collection level endorsement policy

Previously, any member could write to a collection if the transaction has adequate endorsement

Now it is possible to specify an endorsement policy for a private data collection

Overrides chaincode level endorsement policy for keys in the collection

Collections which organizations can write to a collection Organizations can safely interact with smart contracts - e.g. bid, approve, transfer workflows with data privacy and nonrepudiation

More: https://hyperledger-fabric.readthedocs.io/en/release-2.0/endorsement-policies.html#setting-collection-level-endorsement-policies

Security and performance enhancements

State database cache for improved performance

When using external CouchDB state database, read delays during endorsement and validation phases has historically been a performance bottleneck.

New peer cache replaces many of these expensive lookups with fast local cache reads.

The cache size can be configured by using the core.yaml property cacheSize.

Ordering Service Network performance improvements

Transaction validation is parallelized in the commit phase. The ordering node message processing flow has been optimized to remove redundant checks, and the write block processing is now asynchronous.

Ordering service nodes can now choose which channels it will serve instead of the legacy requirement of serving all channels.

Other Enhancements

Other enhancements

Changes to the SDK, Fabric components and chaincode shim

Simplified wallet management for Node.js and Java SDKs provides a persistent data format that is more portable across different SDK language implementations, and pluggable persistent storage to allow users to make use of storage mechanisms appropriate to their deployment environment

The shim package and dependencies for go chaincode are no longer included in the chaincode build environment.

Chaincode that used the shim's NewLogger() will need to shift to a new logging mechanism.

The 'Solo’ and 'Kafka' consensus type is deprecated.

Support for specifying orderer endpoints at the global level in channel configuration is deprecated.

Hyperledger Fabric Docker images will use Alpine Linux, a security-oriented, lightweight Linux distribution.

Other enhancements

Rolling update enhancements

The upgrade documentation has been significantly expanded and reworked, and now have a standalone home in the documentation:

Upgrading to the latest release: https://hyperledger-fabric.readthedocs.io/en/release-2.0/upgrade.html

Upgrade considerations: https://hyperledger-fabric.readthedocs.io/en/release-2.0/upgrade_to_newest_version.html

Enabling the new chaincode lifecycle: https://hyperledger-fabric.readthedocs.io/en/release-2.0/enable_cc_lifecycle.html

Conclusions and next steps

Conclusion and next steps

Fabric Roadmap for 2.x in 2020

Major impact on the following personas:

- Stability & Resilience in Releases
- Tokens
- BFT Consensus
- Ledger Checkpointing
- Usability Updates

Follow JIRA for up to date information: https://jira.hyperledger.org/projects/FAB/issues/FAB-16971?filter=allopenissues

Conclusion and next steps

What do these changes mean for the business?

- An organization can add its own chaincode flavour and validations to better protect the organization’s specific interests before chaincode is deployed or executed on a channel.

- Organizations can privately share data with other organizations in the channel on a need-to-know basis, eliminating the need to define channels or private data collections for many combinations of members

Conclusion and next steps

What do these changes mean for developers and operators?

- New topology models for Raft OSN to address performance and privacy issues.
- Changing approach to chaincode lifecycle and private data collections may increase flexibility and reduce operational overhead if automated checks are implemented.
- Implicit per-organization private data collections combined with collection level endorsement policies allow for a confidential transfer of assets without a need to implement ZKPs oraz encryption.
- Alpine-based images mean the only viable way to execute transactions on Peers and Orderers are through fabric-tools image.
- Using external chaincode launcher is tempting, especially when we consider a potential deployment on Kubernetes clusters.
- Operators can build and launch chaincode with the technology of their choice, removing the requirement to give the peer access to a Docker daemon.

Conclusion and next steps

Further Reading

Evolution of Fabric and the significance of the new release:
https://www.hyperledger.org/blog/2020/01/30/welcome-hyperledger-fabric-2-0:-enterprise-dlt-for-production

As always, you can find additional details of release content in the What's New documentation:
https://hyperledger-fabric.readthedocs.io/en/release-2.0/whatsnew.html

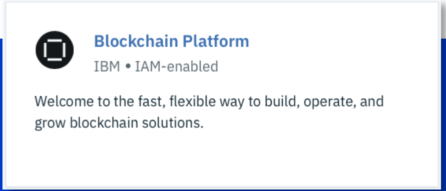
Information about Fabric changes and upgrade considerations can be found in the release notes:
https://github.com/hyperledger/fabric/releases/tag/v2.0.0

Download Hyperledger Fabric v2.0 today and give the new test network in the fabric-samples repository a try:
https://hyperledger-fabric.readthedocs.io/en/release-2.0/install.html

Great article going in-depth regarding Hyperledger Fabric 2.0:
http://www.bchainledger.com/2020/03/whats-new-in-hyperledger-fabric-v20.html

Fabric development roadmap:
https://jira.hyperledger.org/browse/FAB-13754?jql=project%20%3D%20FAB%20AND%20issuetype%20%3D%20Epic%20AND%20fixVersion%20%3D%20v2.1.0

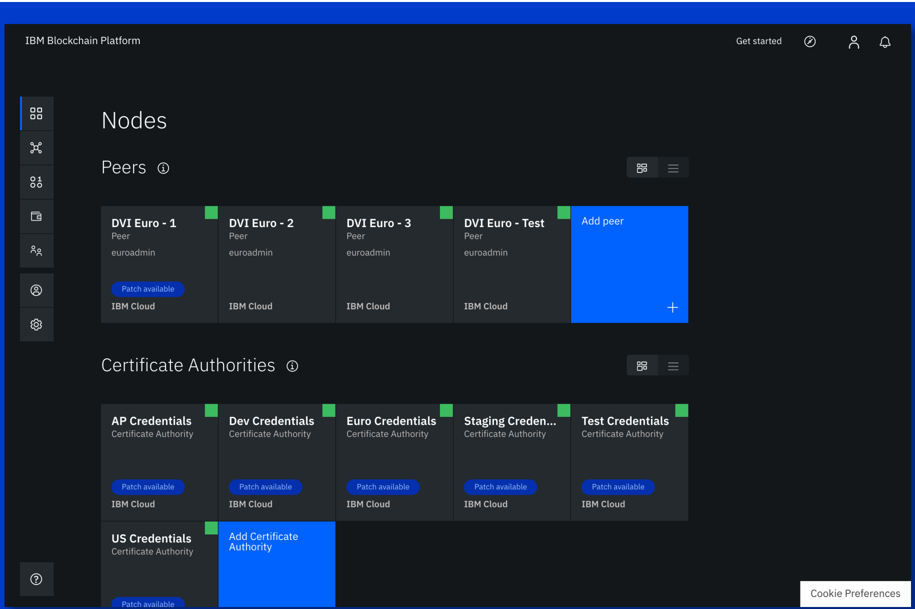
Next Webinar: Deploying IBM Blockchain Platform on IBM Cloud



Blockchain Platform
IBM • IAM-enabled

Welcome to the fast, flexible way to build, operate, and grow blockchain solutions.

1. Go to the IBM Cloud console (cloud.ibm.com) and search the catalog for blockchain



IBM Blockchain Platform

Nodes

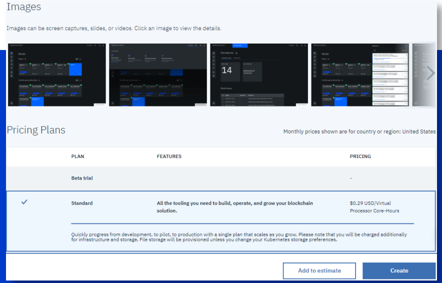
Peers

DVI Euro - 1	DVI Euro - 2	DVI Euro - 3	DVI Euro - Test	Add peer
Peer	Peer	Peer	Peer	
euroadmin	euroadmin	euroadmin	euroadmin	
Patch available				
IBM Cloud	IBM Cloud	IBM Cloud	IBM Cloud	+

Certificate Authorities

AP Credentials	Dev Credentials	Euro Credentials	Staging Creden...	Test Credentials
Certificate Authority	Certificate Authority	Certificate Authority	Certificate Authority	Certificate Authority
Patch available	Patch available	Patch available	Patch available	Patch available
IBM Cloud	IBM Cloud	IBM Cloud	IBM Cloud	IBM Cloud

US Credentials	Add Certificate Authority
Certificate Authority	
Patch available	



Images

Images can be screen captures, slides, or videos. Click an image to view the details.

Pricing Plans

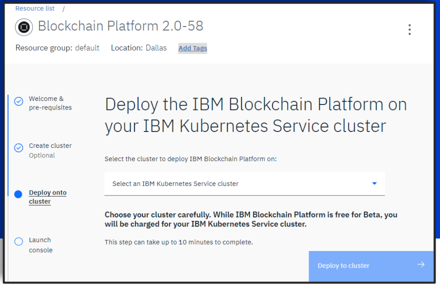
Monthly prices shown are for country or region: United States

	PLAN	FEATURES	PRICING
	Beta trial		-
✓	Standard	All the tooling you need to build, operate, and grow your blockchain solution.	$0.29 USD/Virtual Processor Core-Hours

Quickly progress from development, to pilot, to production with a single plan that scales as you grow. Please note that you will be charged additionally for infrastructure and storage. File storage will be provisioned unless you change your Kubernetes storage preferences.

2. Create your blockchain service instance

3. Create and attach an IBM Kubernetes Service cluster



Resource list /

Blockchain Platform 2.0-58

Resource group: default Location: Dallas Add Tags

- ☑ Welcome & pre-requisites
- ☑ Create cluster Optional
- ● Deploy onto cluster
- ○ Launch console

Deploy the IBM Blockchain Platform on your IBM Kubernetes Service cluster

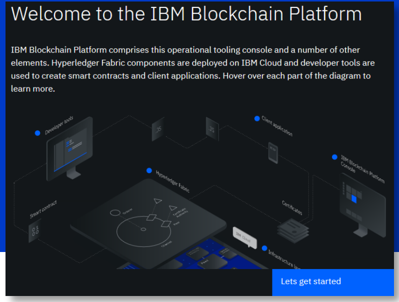
Select the cluster to deploy IBM Blockchain Platform on:

Select an IBM Kubernetes Service cluster

Choose your cluster carefully. While IBM Blockchain Platform is free for Beta, you will be charged for your IBM Kubernetes Service cluster.

This step can take up to 10 minutes to complete.

4. Launch blockchain console



Welcome to the IBM Blockchain Platform

IBM Blockchain Platform comprises this operational tooling console and a number of other elements. Hyperledger Fabric components are deployed on IBM Cloud and developer tools are used to create smart contracts and client applications. Hover over each part of the diagram to learn more.

5. Success! Create or join networks, scale, customize, add HA…

…and you're on your way!

Q&A