

Name: Nicole Staehle

Project Name: Quorum Blockchain Implementation

Abstract

Problem Statement

With the advent of blockchain's private, permissioned networks, developers in various industries are tasked with implementing a daunting set of options to ensure transactional and institutional security.

Developed by J.P. Morgan Chase to address this task, Quorum supports transactional volumes per institution, with privacy at the transaction level. Quorum is a blockchain consortium (private network), with support for smart contract business logic applications, and a proprietary dashboard interface called Cakeshop for managing the network and contracts.

In this demo, we will:

- Discuss the security options of the Quorum Single Member Blockchain Network
- Deploy one, exploring PowerShell options for displaying and creating resources

High-Level Overview of steps:

Install or download from Azure Portal:

- Quorum Single Member Blockchain Network
- Configure:
- JSON template variables and parameters (if desired)
 - PowerShell script for Automation Account

File Information:

- Format of code files: JSON (47 kb), .PS1 (4 kb)
- Hardware: Quorum Consortium blockchain VMs (2), Standard, LRS storage; Standard D1 v2 size
- Software: Ubuntu Server 16.04 (guest); Windows 7 (host)

Sources:

- [Ethereum white paper](#)
- [Quorum white paper](#)
- [Quorum documentation link from Azure portal](#)

Phase I: Setting Up Your Blockchain

Quorum's security requirements are explicit in this process, within the permissions settings. Let's walk through step by step, as many times you won't know the complete requirements until after entering an unsuitable value.

Portal Blades: Quorum Single Member Blockchain Network

Blade Tab: Step 1: Basics

Create Quorum Single Member Blockchain Network ×

Basics ×

1 Basics >
Configure basic settings

2 Network size and performance >
Define the number and size an...

3 Quorum Specific Settings >
Configure the Quorum

4 Summary >
Quorum Single Member Block...

5 Buy >

* Resource prefix ⓘ
qdemo ✓

VM user name ⓘ
gethadmin

* Authentication type
Password SSH public key

* Password ⓘ
..... ✓

* Confirm password
..... ✓

Restrict access by IP address ⓘ
No ▾

Subscription
NikiAzure ▾

During this process, you *must* hover over the gray ‘I’ icons (as will be explained in the demo video) to get the requirements for each line item (if there are any). Following is an explanation of each element’s role in the resource you’re creating, along with the requirements from the icons:

Resource prefix:

- Used to categorize resources
- Must begin with a lowercase letter, be between 2-6 characters, and can contain only numbers and lowercase letters

Password:

Must be between 12-72 characters long and contain 3 of the following:

- 1 lowercase letter
- 1 uppercase letter
- 1 number
- 1 special character

Blade tab: Step 2: Network Size and Performance

Create Quorum Single Member... ×

Network Size and Performance

1 Basics Done ✓

2 Network size and performance Define the number and size an... >

3 Quorum Specific Settings Configure the Quorum >

4 Summary Quorum Single Member Block... >

5 Buy >

Consortium Member Id ⓘ
0 ▾

Number of VMs
Number of block makers ⓘ
1 ▾

Number of voters ⓘ
1 ▾

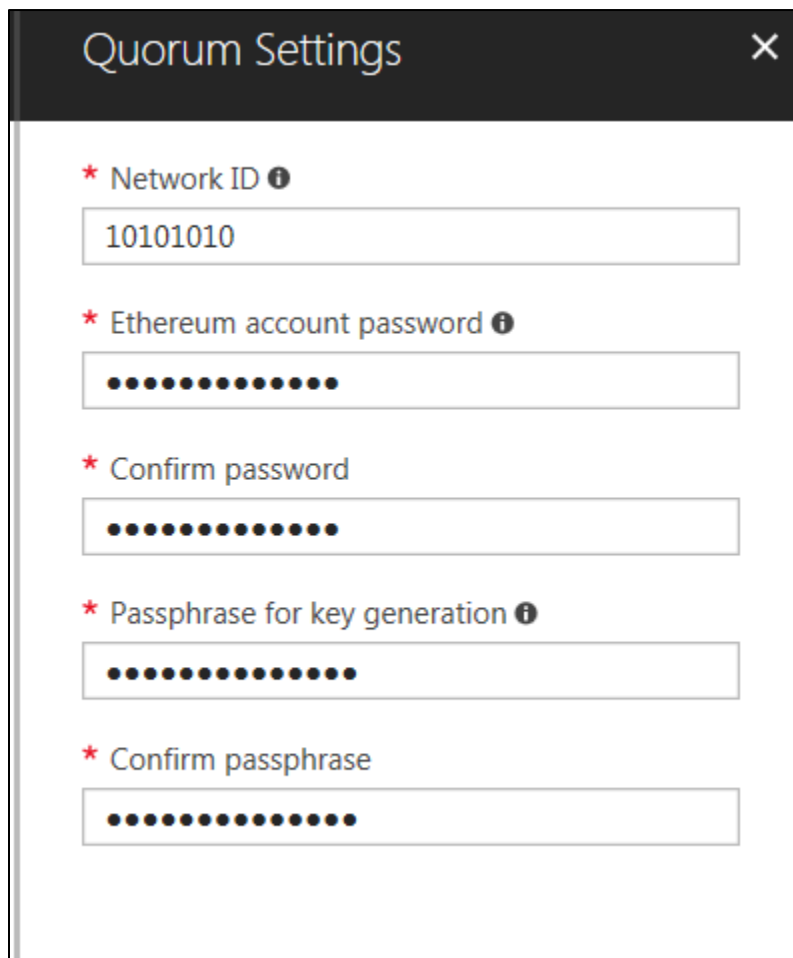
Number of observers ⓘ
0 ▾

Infrastructure options
Storage performance ⓘ
Standard Premium

* Virtual machine size
2x Standard D1 v2 >

- Consortium member id: In a multi-member setup, each member should have a unique value to ensure that they can connect.
- Number of block makers (1).
Block makers create and propose blocks to the network (currently limited to 1 for this offer type).
- Number of Voters (1)
Voters vote on blocks.
- Number of observers (0)
Observers passively listen to traffic on the network like transactions and blocks.
- Infrastructure options: Choose standard (HDD); premium is SSD and is more expensive
- Virtual Machine Size
2x standard D1 v2

Blade tab: Step 3: Quorum Settings



You'll need to set two different types of passwords. The first one, *Ethereum account password*, is for the default account that will be created once you click the 'Create' button in the portal.

The *passphrase* is used to generate a private key. This lets Quorum identify and verify users whose private key matches their passphrase.

- Network ID: Only nodes that share the same ID can peer with each other.
- Ethereum account password: Secures default account that is created.

Requirements:

- 12+ characters
- 1 lower case
- 1 upper case
- 1 number
- No double quotes or forward slashes

- Passphrase for key generation:
 - Same requirements as Ethereum account password
 - Needs to be random enough to generate a strong private key

Blade tab: Step 4: Summary

Below is a summary of our choices from the previous blade tabs:

Summary

Validation passed

Basics

Subscription	NikiAzure
Resource group	Quorum
Location	East US
Resource prefix	qdemo
VM user name	gethadmin
Password	*****
Restrict access by IP address	No

Network Size and Performance

Consortium Member Id	0
Number of block makers	1
Number of voters	1
Number of observers	0
Storage performance	Standard
Virtual machine size	Standard D1 v2

Quorum Settings

Network ID	10101010
Ethereum account password	*****
Passphrase for key generation	*****

We can click ‘Ok’ (button not pictured) to move on to step 5, but we really need to view the template first to see what we’re getting (see next step). With most users having no or limited knowledge of blockchain, it’s important to know everything that Azure will create for us (please also note that the charges will differ, since this is a third-party and not Microsoft service).

What’s in the template? You can click through the sidebar to view the code sections for the different resources that Azure will create. Azure gives you the option to export the template in your language of choice (in addition to JSON): .NET, Powershell, Bash, and Ruby.

Viewing the template in the portal

```
1 {
2   "$schema":
3     "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
4   "contentVersion": "1.0.0.0",
5   "parameters": {
6     "namePrefix": {
7       "type": "string",
8       "metadata": {
9         "description": "String used as a base for naming resources (6
10 alphanumeric characters or less). A unique hash is prepended to the string for some
11 resources, while resource-specific information is appended."
12       },
13       "maxLength": 6
14     },
15     "authType": {
16       "type": "string",
```


Viewing another part of the template in Visual Studio 2017. This part includes storage and disk information for the block maker (first screenshot) and voter (second screenshot) virtual machines. As chosen, they have standard, locally-redundant storage, and are built on the Ubuntu Server 16.04 image.

Block Maker

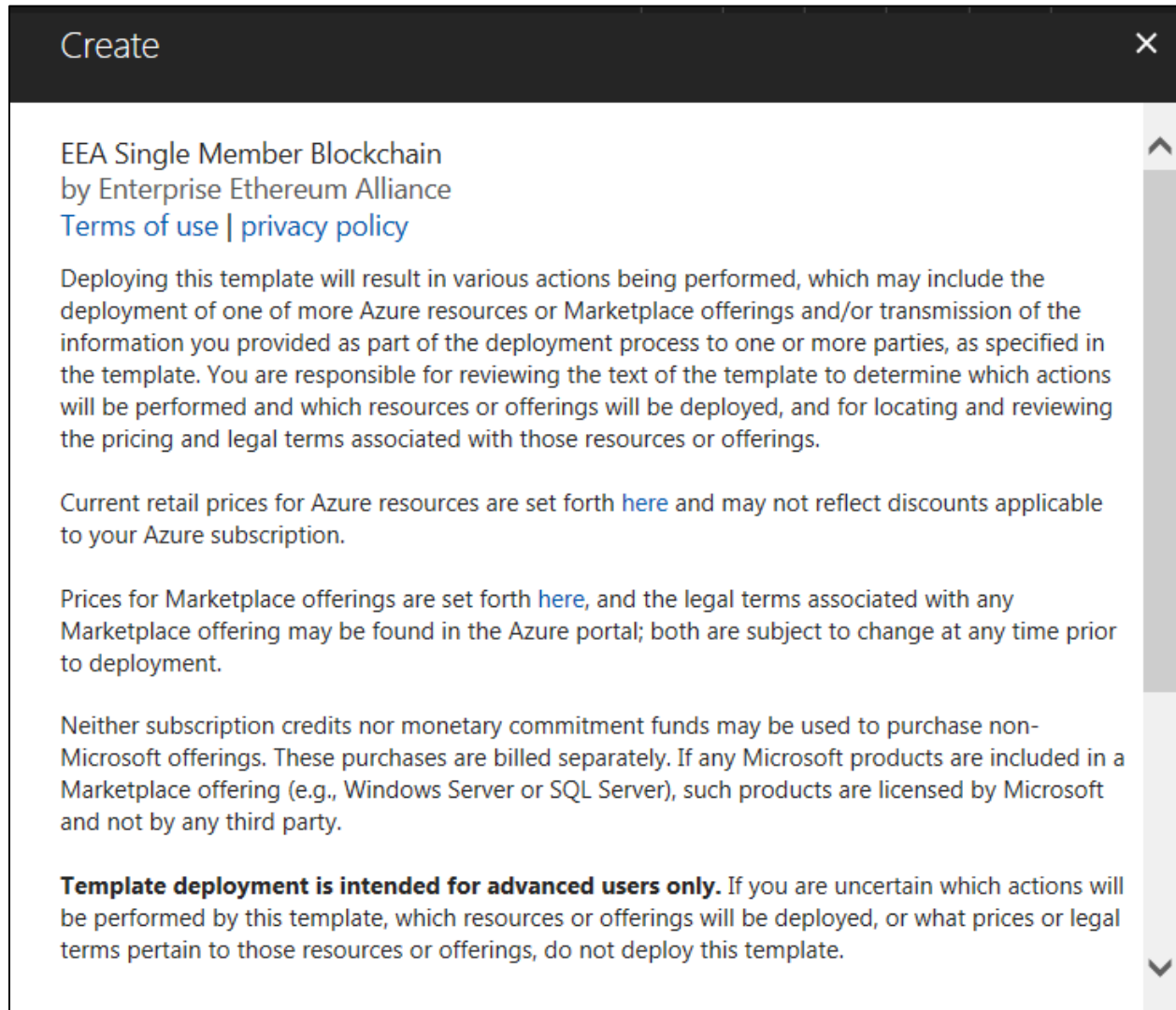
```
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "Canonical",
        "offer": "UbuntuServer",
        "sku": "16.04.0-LTS",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Linux",
        "name": "[concat(parameters('virtualMachines_qdemo4b3g_bm0_name'),
        'osdisk')]",
        "createOption": "FromImage",
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS",
          "id": "[parameters('virtualMachines_qdemo4b3g_bm0_id')]"
        },
        "diskSizeGB": 30
      }
    },
  },
}
```

```
},
"storageProfile": {
  "imageReference": {
    "publisher": "Canonical",
    "offer": "UbuntuServer",
    "sku": "16.04.0-LTS",
    "version": "latest"
  },
  "osDisk": {
    "osType": "Linux",
    "name": "[concat(parameters('virtualMachines_qdemo4b3g_vtr0_name'))",
    "createOption": "FromImage",
    "caching": "ReadWrite",
    "managedDisk": {
      "storageAccountType": "Standard_LRS",
      "id": "[parameters('virtualMachines_qdemo4b3g_vtr0_id')]"
    },
    "diskSizeGB": 30
  },
  "dataDisks": []
}
```

Besides these 2 VMs, we can scroll through the rest of the template (please see template code in .zip and on GitHub) to view the rest of the VM resources that will be created. This includes load balancers, NICs, NSGs, and more.

Blade tab: Step 5: Buy

You need to read this carefully. Going over the provided template is a prerequisite for understanding what it is that Azure is creating. Since we've already done that, we can click 'Create'. [How long does it take to deploy? Is portal or C# faster?]



Create ×

EEA Single Member Blockchain
by Enterprise Ethereum Alliance
[Terms of use](#) | [privacy policy](#)

Deploying this template will result in various actions being performed, which may include the deployment of one or more Azure resources or Marketplace offerings and/or transmission of the information you provided as part of the deployment process to one or more parties, as specified in the template. You are responsible for reviewing the text of the template to determine which actions will be performed and which resources or offerings will be deployed, and for locating and reviewing the pricing and legal terms associated with those resources or offerings.

Current retail prices for Azure resources are set forth [here](#) and may not reflect discounts applicable to your Azure subscription.

Prices for Marketplace offerings are set forth [here](#), and the legal terms associated with any Marketplace offering may be found in the Azure portal; both are subject to change at any time prior to deployment.

Neither subscription credits nor monetary commitment funds may be used to purchase non-Microsoft offerings. These purchases are billed separately. If any Microsoft products are included in a Marketplace offering (e.g., Windows Server or SQL Server), such products are licensed by Microsoft and not by any third party.











Template deployment is intended for advanced users only. If you are uncertain which actions will be performed by this template, which resources or offerings will be deployed, or what prices or legal terms pertain to those resources or offerings, do not deploy this template.











Summary of What We've Learned So Far

- Quorum, as a private blockchain, has different layers of security than the Azure Linux VMs. Multiple password types, in addition to stringent requirements for each, make this a viable option for the enterprise.
- You need to review Azure template that you generate based on your summary (blade tab 4), to ensure you know how your resource group will be populated, and how you will be charged.


Phase II: Admin Portal

Once created, go to your resource group to verify the resources you saw in the template were created:

NAME 	TYPE 	LOCATION 
 nic-bm0	Network interface	East US
 nic-voter0	Network interface	East US
 qdemo4b3g-bm0	Virtual machine	East US
 qdemo4b3g-bm0_OsDisk_1_fb4b2fee3ff04e3c90ddd6d85...	Disk	East US
 qdemo4b3g-LB	Load balancer	East US
 qdemo4b3gnsng	Network security group	East US
 qdemo4b3g-publicip	Public IP address	East US

NAME 	TYPE 	LOCATION 
 qdemo4b3g-LB	Load balancer	East US
 qdemo4b3gnsng	Network security group	East US
 qdemo4b3g-publicip	Public IP address	East US
 qdemo4b3gquorumAvSet	Availability set	East US
 qdemo4b3gvnet	Virtual network	East US
 qdemo4b3g-vtr0	Virtual machine	East US
 qdemo4b3g-vtr0_OsDisk_1_3a46a4ec59ba4584b483dbe9...	Disk	East US

We're going to verify the template output and get the URL for Quorum's admin portal and the accompanying dashboard, Cakeshop.

 <i>Search for deployments by name...</i>		
DEPLOYMENT NAME	↑↓	STATUS
enterprise-ethereum-allian...		✓ Succeeded
vmExtensionLinkedTemplate		✓ Succeeded
bmVMLinkedTemplate		✓ Succeeded
voterVMLinkedTemplate		✓ Succeeded
vmExtensionLinkedTemplat...		✓ Succeeded
observerVMLinkedTemplate		✓ Succeeded
loadBalancerLinkedTemplate		✓ Succeeded

Go to one of the VM resources and copy-paste the URL under ‘DNS’.

Resource group (change) Quorum	Computer name qdemo4b3g-bm0
Status Running	Operating system Linux
Location East US	Size Standard D1 v2 (1 vcpu, 3.5 GB memory)
Subscription (change) NikiAzure	Public IP address <input type="text"/>
Subscription ID <input type="text"/>	Virtual network/subnet qdemo4b3gvnet/befficslsz55i
	DNS name qdemo4b3g.eastus.cloudapp.azure.com

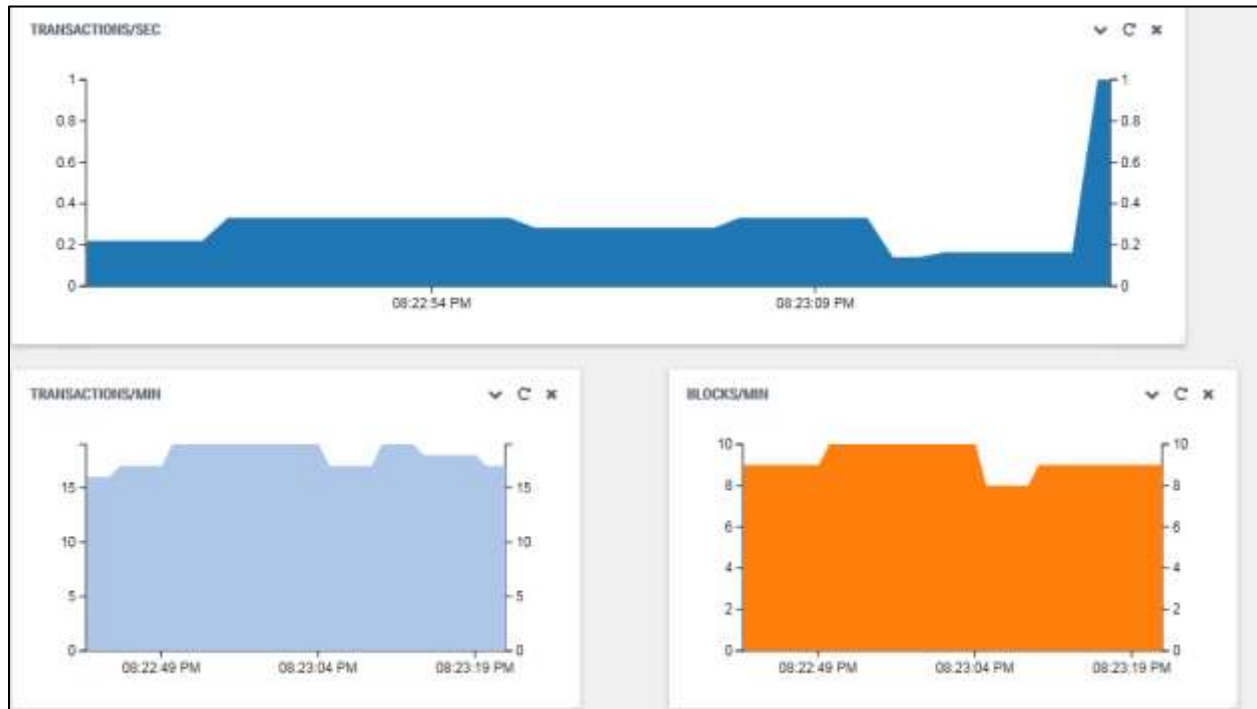
This takes me to the admin portal landing page, which shows my node status. I have two nodes (remember that this network creates 2 VMs, as seen in step 2 of the blade tab): one block maker (top, suffix bm0) and one voter (bottom, suffix vtr0).

Ethereum Node Status

Node Hostname	Cakeshop	Peer Count	Latest Block Number
qdemo4b3g-bm0	qdemo4b3g-bm0	1	125
qdemo4b3g-vtr0	qdemo4b3g-vtr0	1	125

By clicking on either of the values in the Cakeshop column, I can go through to the dashboard visualization and smart contracts tool, Cakeshop. The following screenshots are for the block maker as it's mining blocks and creating transactions.

Transactions per second and minute; blocks per minute



Dashboard with node status, blocks, transactions and more

CAKESHOP // DASHBOARD

CONSOLE

CONTRACTS

SAMBOX

CHAIN EXPLORER

WALLET

PEERS

API

HELP

Cakeshop 0.9.1
Build 6c7a88ee

Console

NODE STATUS
Running

PEERS
1

BLOCKS
177

QUEUED TXNS
3

NODE INFO

ID	1578a5d3a307390a7e829503e0bc041cf23247cb...
Node URL	enode://1578a5d3a307390a7e829503e0bc041cf...
Rpc URL	http://10.0.0.5:8545
Node Name	Geth/qdemo4b3g-bm0/v1.5.0-unstable-6d0b810...
Node IP	172.18.0.2
Latest Block	177

NODE CONTROL

Restart Node

Stop Node

Start Node

Create New Chain

NODE SETTINGS

Committing Transactions
Yes

Network ID
1056

Identity
cakeshop

Sample contract code in smart contracts editor (Sandbox)

```
contract Owned {
    address owner;

    function Owned() { owner = msg.sender; }

    // This contract only defines a modifier but does not use it - it will
    // be used in derived contracts.
    // The function body is inserted where the special symbol "_" in the
    // definition of a modifier appears.
    modifier only_contract_owner { if (msg.sender == owner) _ }
}

contract Bank is Owned {






    bool enabled;


    struct Record {
        address owner;
        bytes32 id;
        uint value;
    }


    bytes32[] vault_ids;
    uint num_vault_ids;
    mapping (bytes32 => Record) vault;

    // only account owner or Bank owner is allowed
    modifier only_account_owner(bytes32 _id) { if (vault[_id].owner != 0 &&
(vault[_id].owner == msg.sender || msg.sender == owner)) _ }
```

Sandbox options for smart contracts



#182

Choose Contract

From Deployed Contracts:


▼


Or Deploy From Editor:


▼


Or Enter Address:

address


Transact



Accounts

0x1932adf7b7e7d3530395...100000

0x4d66ee04f588100ee09a...100000

< 00011151 1011000 10000 >

Contract State

Paper Tape

18 | Nicole Staehle | Blockchain

Phase III: Coding Resources

You can use the .PS1 template provided by Azure when you create your resources in the portal to do other things, such as creating a similar resource group. An automation account (with elevated permissions) will allow you to create runbooks to accomplish various tasks.

```
63 #*****
64 # Script body
65 # Execution begins here
66 #*****
67 # $ErrorActionPreference = "Stop"
68
69 # sign in
70 Write-Host "Logging in...";
71 Login-AzureRmAccount;
72
73 # select subscription
74 Write-Host "Selecting subscription '$subscriptionId';
75 Select-AzureRmSubscription -SubscriptionID $subscriptionId;
76
77 # Register RPs
78 $resourceProviders = @("microsoft.compute","microsoft.network");
79 if($resourceProviders.length) {
80     Write-Host "Registering resource providers"
```

Then you can add in the script, or in the Cloud Shell, PS commands to verify your resources were deployed (or view an existing resource group).

```
PowerShell ▾ | 🔌 ? ⚙️
Azure:\
PS Azure:\> Get-AzureRMResourceGroup -name quorum

ResourceGroupName : Quorum
Location           : eastus
ProvisioningState  : Succeeded
Tags               :
ResourceId         : /subscriptions/9d34afe4-302e-4
                    591-8c18-a08408735c57/resource
                    Groups/Quorum
```

Lessons Learned

- Blockchain is still a new, open-source technology, and SDKs are still being updated.

Pros

- Fast deployment
- High security is available, including at the transaction level and around permissions
- Ease of use with admin portal
- One language for smart contract writing
- Immutable ledger
- Smart contracts catch suspicious behavior

Cons

- Solidity not supported everywhere, limited IDEs; VS extension and Ethereum Studio deprecated

URLs:

- Short Video: <https://www.youtube.com/watch?v=k5czvZfoHYY>
- Long Video: <https://www.youtube.com/watch?v=5g-Xy82pfKw>
- GitHub Repository with all artifacts:
<https://www.github.com/healthdatachick/FinalProject>