
Blockchain Innovation Program Tutorial Framework



POWERED BY



Blockchain Innovation Program

Tutorial Framework

The Blockchain Innovation Program is designed as an intensive 10-week programme which will provide students with educational and entrepreneurial support for them to develop a comprehensive understanding of what's involved in Bitcoin Application Development. The programme will see the students complete the three bitcoin primitive courses Hash Functions, Merkle Trees, and Digital Signatures before they complete the newly refactored Introduction to Bitcoin Development.

Educators from the Bitcoin SV Academy team will prescribe weekly resource and question packs to stimulate the students to develop a deeper consideration for what is involved in creating a scalable Bitcoin application. The students will attend fortnightly tutorials where their answers to the question pack will be evaluated and discussed in greater detail.

Live session #1

Hash Functions	<ul style="list-style-type: none">• Hash Functions and Hash Tables• Content Addressed Distributed Data Structures• Efficient Provable Data Possession for Hybrid Clouds
----------------	---

Course pre-requisite: Hash Functions (primitives)

Worksheet to complete prior to the live session: Week 1 – Hash Functions

Live session #2

Merkle Trees	<ul style="list-style-type: none">• GitHub MerkleDAG• ForkBase: Immutable, Tamper-evident Storage• Substrate for Branchable Applications• Merkle-CRDTs - MerkleDAGs meet CRDTs• Merkle²: A Low Latency Transparency Log System
Digital Signatures	<ul style="list-style-type: none">• Digital Signatures• Legitimizing Technologies: Digital Signatures Case Study.• Segwit, Mixing and Law• SigHash Flags

Course pre-requisite: Merkle Trees and Digital Signatures (Primitives)

Worksheet to complete prior to the live session: Week 2 Merkle Trees and Week 3 Digital Signatures.

Live session #3

Data and Databases	<ul style="list-style-type: none">• What is DBaaS?• SQL vs NoSQL• What is Cloud Storage?• What is Object Storage?• Block vs File Storage• What is a Load Balancer?• Kubernetes vs Docker
API led Event-Driven & Microservices Architectures	<ul style="list-style-type: none">• API vs SDK• What is API Management?• What is a REST API?• What is an API Gateway?• What is Event Driven Architecture?• What are Microservices?• Architecting a Cloud Native API Solution.• Blockchain Enabled Trustless API Marketplace• Unofficial API and Browser Extension Development for Augmenting Student Resources

Course pre-requisite: Bitcoin Development Chapter 1

Worksheet to complete prior to the live session: Week 4 Data and Databases and API led Event-Driven & Week 5 Microservices Architectures

Live session #4

Debunking the Blockchain Trilemma, CAP Theorem & Application Scalability	<ul style="list-style-type: none">• Myths of Decentralisation• On Decentralisation• The Wizard of Blockchain• Cost Performance Trade-Off Evaluation in Microservices impacted by the CAP Theorem Limitations
Working Blockchain & Overlay Networks	<ul style="list-style-type: none">• A Survey and Comparison of P2P Overlay Network Schemes.• Virtual Networking Explained• What is a Content Delivery Network• Mandala Network• SPV• Working Blockchain

Course pre-requisite: Bitcoin Development Chapter 2&3

Worksheet to complete prior to the live session: Week 6 Debunking the Blockchain Trilemma, CAP Theorem & Week 7 Application Scalability and Working Blockchain & Overlay Networks.

Live session #5

Intro to Git and Github	<ul style="list-style-type: none">• Git and GitHub for Beginners• Getting Started With OpenSSH Key Management.• Setting up an Nx monorepo with Angular• Setting up CI/CD with Github Actions and Vercel
Constructing Transactions & Script	<ul style="list-style-type: none">• Introduction to Bitcoin Transactions• MintBlue API, SDK and Integrations

Course pre-requisite: Bitcoin Development Chapter 4-5

Worksheet to complete prior to the live session: Week 8 Intro to Git and Github and Week 9 Constructing Transactions and Bitcoin Script.

Live session #6

Metanet	<ul style="list-style-type: none">• Metanet Overlay• Dagda• The Birth of Ontology & the DAG• Tutorial on directed Acyclic Graphs• A.N.N.E preview.
End of programme	<ul style="list-style-type: none">• Feedback on project completed by students• Wrap up of the programme

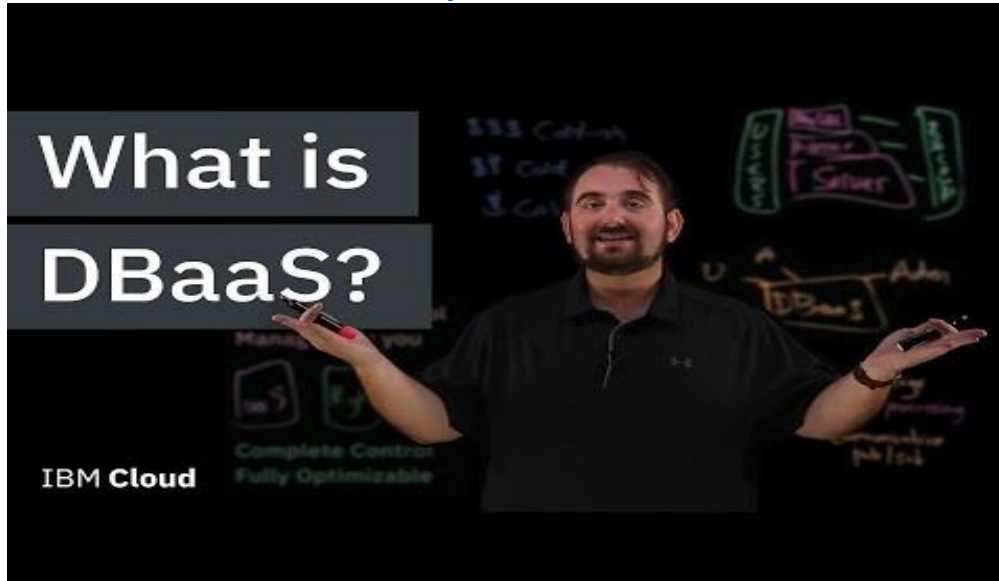
Course pre-requisite: Bitcoin Development Chapter 6

Worksheet to complete prior to the live session: Week 10 Metanet

Blockchain Innovation Program

Worksheets

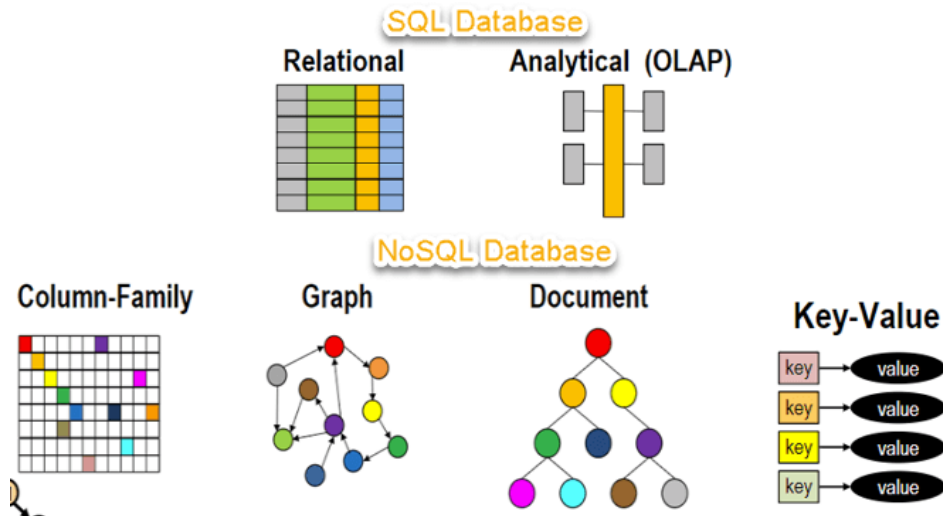
1. [What is DBaaS? Should you use it?](#)



- How could the blockchain be thought of as a DBaaS?
- What are some limitation to using the blockchain as a DBaaS?
- How could the blockchain enhance legacy DBaaS?

2. [SQL vs. NoSQL: What's the difference?](#)





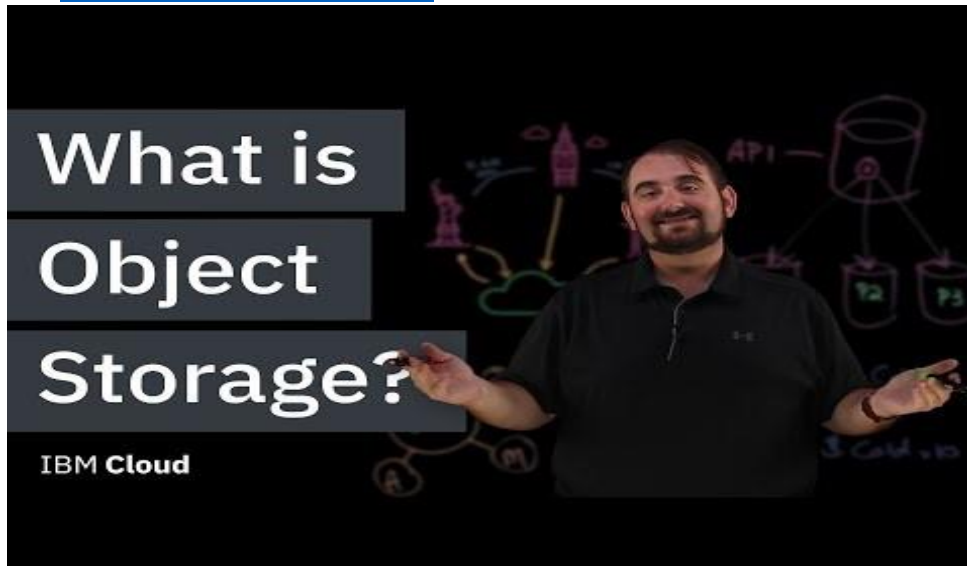
- Does the blockchain more closely resemble a SQL or NoSQL database? How?
- Discuss the feasibility of a SQL and NoSQL hybrid.

3. What is Cloud Storage?



- How could an overreliance upon OP_RETURN outputs expose you to the vulnerability of ephemeral storage?
- Why does the host of blockchain storage need to be highly performant?
- As data pushed to the blockchain may be pruned, which would be the cheapest type of cloud storage for maintaining blockchain data as a service?

4. [What is Object Storage?](#)



5. [Block vs. File Storage](#)



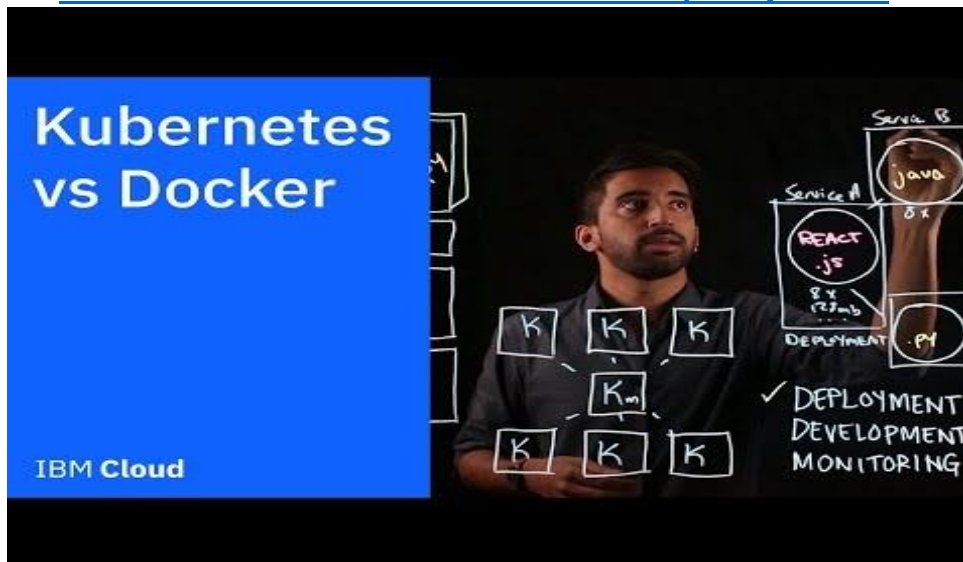
- What are the main benefits and drawbacks of block, object and file storage?
- Which type of storage does bitcoin natively support?

6. [What is a Load Balancer?](#)



- How is a load balancer critical for achieving horizontal scaling of your application server?
- How can a load balancer help to optimise running costs to service your application?
- How is a load balancer a key component for cloud native architecture?
- How does a load balancer decide how to direct the load?

7. [Kubernetes vs. Docker: It's Not an Either/Or Question](#)



- How is Kubernetes a logical progression for achieving scale with a dockerised application?
- What three advantages does Kubernetes orchestration offer?

