

Algorand Development

Algorand Development

F. Appendix

1. [Additional Tools & Resources](#)
2. [Development Versus Production Considerations](#)
3. [Additional Topics for Further Exploration](#)

G. Appendix

1. Additional Tools & Resources

The Algorand community offers a growing [ecosystem of tools, projects, and resources](#). The following sets of tools may be particularly useful for developers:

- [API Services](#)
- [Block Explorers](#)
- [Frameworks](#)
- [IDEs](#)
- [SDKs](#)

The following resources may also be helpful as you learn more about Algorand development:

- [Official Algorand Documentation](#)
- [PyTeal documentation](#)
- [Reach documentation](#)
 - [Reach Self-Study Projects](#)
- [Algorand Developer Community](#)
 - [Algorand Developer Portal](#)
 - [Algorand Developer Forums](#)
 - [Algorand Developer Discord](#)
 - [Algorand Developer Blog](#)
- [Ecosystem Tools & Projects](#)
- [Curated List of Algorand Projects](#)

2. Development Versus Production Considerations

Sandbox makes it easy to create private local networks, as well as remove networks and reset network states. Connecting to a "real" network (and "catching up" to the latest round) is also an option. However, this may add additional considerations, depending on the stage of your development process.

The normal catchup procedure involves cryptographic validation of the *entire* history of the blockchain (i.e. by processing every single block in the chain). Full catchup is time-consuming but necessary to run trusted nodes. [Fast catchup](#) enables node networks to be rapidly synced and updated using catchpoint snapshots. This feature is available within Sandbox and the Algorand CLI through the `goal node catchup` command. Sandbox makes it possible to rapidly implement fast catchup, or skip catchup altogether for immediate bootstrapping capabilities.

Before deploying an application on MainNet, it is prudent to transition to a production-ready environment with full catchup by:

- Using a third-party service which provides access to various developer tools.
- [Setting up your own node](#) to leverage *all* developer tools. (*Recommended*)

A detailed comparison of these options is provided in the [Algorand Developer Documentation](#):

	Use a third-party service	Use Docker Sandbox	Run your own node
Time	Seconds - Just signup	Minutes - Same as running a node with no catchup	Days - need to wait for node to catchup
Trust	1 party	1 party	Yourself
Cost	Usually free for development; pay based on rate limits in production	Variable (with free option) - see node types	Variable (with free option) - see node types
Private Networks	✗	✓	✓
goal , algokey , kmd	✗	✓	✓
Platform	Varies	MacOS; Linux	MacOS; Linux
Production Ready	✓	✗	✓

3. Additional Topics for Further Exploration

- Rekeying
- REST APIs and Endpoints
- Running nodes - various ways
- Debugging
- Nodes: Installation, Common Operations, Consensus participation, References
- Encoding and Decoding
- Algorand consensus
- Supporting Algos or Algorand Standard Assets (ASAs) in your applications
- Algorand implementations in various languages
- Indexer
 - Search & filter on-chain data
- Cross-chain applications & compatibility