

How to Build an AVS

Step 1: Grasp EigenLayer fundamentals

Basic Understanding

- Understand the basics of EigenLayer* Understand [how EigenLayer works](#).
- - - Read our [GitHub](#)
- - - on how EigenLayer contract architecture works.
- - - Understand the type of trust you would need with [programmable trust](#)
- - - .
- - - Play with our [Incredible Squaring AVS demo](#)
- - - - a dummy AVS we built to showcase the on-chain contracts and off-chain node software components of an AVS.1. Incredible Squaring [Overall 5 min Walk-Through](#)
- - - - 1. Incredible Squaring [TaskManager 5 min Walk-Through](#)
- - - Once you have an idea of what you want to build on EigenLayer, submit an [AVS Questionnaire](#)
- - - and get in touch with us.

Assessing

- Once we review your AVS Questionnaire response, we will reach out and hop on a call with you to address any questions you have about the technical architecture of your system, the process of becoming an AVS, and EigenLayer's value proposition.
- Look into our docs more and determine whether building on EigenLayer makes sense and is feasible. You can ask any questions in our shared Telegram group, and our researchers and engineers will answer any specific questions.

Step 2: From idea to testnet:

Engineering

- AVS must write on-chain contracts, which contain the AVS registry contract and slashing contract, and test out the integration to EigenLayer by studying our Incredible Squaring demo.* AVS can use [Incredible Squaring](#)
- - to write on-chain code.
- - The Incredible Squaring contracts consist of shadow deployment of Eigenlayer core contracts that AVSs can use for understanding how off-chain and on-chain code for any AVS can interact with Eigenlayer contracts. It contains the Strategy Manager (used to deposit and withdraw stake and stake accounting), Delegation Manager (used for operator registration and stakers delegating to operators), and experimental_slasher.sol (used for AVS syncing with Eigenlayer state) contracts.
- - We are providing a set of registry contracts that AVS teams can use as their own registry contracts, which will help them save development effort. Of course, AVS teams are free to write their own registry contracts instead.
- AVSs must write their off-chain node software for operators to run.* AVS can use our [EigenSDK](#)
- - for writing off-chain code; the EigenSDK is being openly developed and is a work in progress.

- - The EigenSDK has libraries and templates to help you write your off-chain node software code. It consists of modules for signature aggregation, interacting with EigenLayer contracts, networking, cryptography, and clients for events monitoring. Currently, many of these libraries are still work in progress.

Step 3: From testnet to mainnet:

Testing

We can invite our partners to test out your AVS test network

Auditing

Audit with at least 2-3 reputable audit firms

Go live with us

Launch on mainnet after having a sufficient period of risk assessment, testing, and simulation![Previous AVS Developer Guide](#) [Next First Steps Towards Designing an AVS](#)