

Sui Full Node Configuration

These instructions are for advanced users. If you just need a local development environment, you should instead follow the instructions in [Create a Local Sui Network](#) to create a local Full node, validators, and faucet.

Sui Full nodes validate blockchain activities, including transactions, checkpoints, and epoch changes. Each Full node stores and services the queries for the blockchain state and history.

This role enables validators to focus on servicing and processing transactions. When a validator commits a new set of transactions (or a block of transactions), the validator pushes that block to all connected Full nodes that then service the queries from clients.

Sui Full nodes:

Sui Full nodes sync with validators to receive new transactions on the network.

A transaction requires a few round trips to $2f+1$ validators to form a transaction certificate (TxCert).

This synchronization process includes:

This synchronization process requires listening to at a minimum $2f+1$ validators to ensure that a Full node has properly processed all new transactions. Sui will improve the synchronization process with the introduction of checkpoints and the ability to synchronize with other Full nodes.

A Sui Full node is essentially a read-only view of the network state. Unlike validator nodes, Full nodes cannot sign transactions, although they can validate the integrity of the chain by re-executing transactions that a quorum of validators previously committed.

Today, a Sui Full node maintains the full history of the chain.

Validator nodes store only the latest transactions on the frontier of the object graph (for example, transactions with >0 unspent output objects).

Follow the instructions here to run your own Sui Full node. Sui Full nodes run using the sui-node binary.

Suggested minimum hardware to run a Sui Full node:

Sui recommends running Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can run a Sui Full node on macOS, but this is only recommended for development and not for production use.

Make sure to [update Rust](#).

Use the following command to install additional Linux dependencies.

Instructions for building, installing, or downloading the sui-node binary are available at [Sui Install](#). These install instructions are specific to the sui cli, but apply to the sui-node binary as well.

There are many ways to run a Sui Full node (bare metal, virtual machine, Kubernetes statefulset, and so on), and the solution that you choose depends on your specific needs as well as the infrastructure that you have available.

There are some specific considerations to keep in mind when running a Sui Full node that apply to all environments:

There's a guide in the Sui repository on running a Full node via [Docker Compose](#). This alone is not suitable for a production environment, but can be used to get a Full node up and running quickly on a virtual machine or local machine for development purposes. Refer to [Running a Full node](#) for instructions relevant to production use cases.

When you are ready to run sui-node in your production environment, you can set up your Full node by completing the following steps:

Make a copy of the [Full node YAML template](#):

Download the genesis blob for the network to use:

For Testnet or Mainnet: Edit the fullnode.yaml file to include peer nodes for state synchronization. Append the following to the end of the current configuration:

Optional: Set up the [Archival Fallback](#), which allows you to sync checkpoints if you fall behind the network's seed-peers.

Optional: Skip this step to accept the default paths to resources. Edit the `fullnode.yaml` file to use custom paths.

Update the `db-path` field with the path to the Full node database. `db-path: "/db-files/sui-fullnode"`

Update the `genesis-file-location` with the path to `genesis.blob`.

You should not start syncing your Full node from the start of the genesis. This will take a very long time and consume a lot of resources (including likely filling up your disk).

Instead, start your Full node from a recent snapshot. You can find details on how to obtain a snapshot from the [Sui Snapshots guide](#).

Now that you have your Full node config file set up, and you've obtained a snapshot, you can start your Full node by running the `sui-node` binary with your `fullnode.yaml` configuration file:

It's a good idea to use something like `systemd` to manage your Full node in a production environment.

If you receive a `cannot find -lpq` error, you are missing the `libpq` library. Use `sudo apt-get install libpq-dev` to install on Linux, or `brew install libpq` on MacOS. After you install on MacOS, create a Homebrew link using `brew link --force libpq`. For further context, reference the [issue on Stack Overflow](#).

If you receive the following error:

Then update the metrics address in your `fullnode.yaml` file to use port 9180.

Features

Sui Full nodes:

Sui Full nodes sync with validators to receive new transactions on the network.

A transaction requires a few round trips to $2f+1$ validators to form a transaction certificate (TxCert).

This synchronization process includes:

This synchronization process requires listening to at a minimum $2f+1$ validators to ensure that a Full node has properly processed all new transactions. Sui will improve the synchronization process with the introduction of checkpoints and the ability to synchronize with other Full nodes.

A Sui Full node is essentially a read-only view of the network state. Unlike validator nodes, Full nodes cannot sign transactions, although they can validate the integrity of the chain by re-executing transactions that a quorum of validators previously committed.

Today, a Sui Full node maintains the full history of the chain.

Validator nodes store only the latest transactions on the frontier of the object graph (for example, transactions with >0 unspent output objects).

Follow the instructions here to run your own Sui Full node. Sui Full nodes run using the `sui-node` binary.

Suggested minimum hardware to run a Sui Full node:

Sui recommends running Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can run a Sui Full node on macOS, but this is only recommended for development and not for production use.

Make sure to [update Rust](#).

Use the following command to install additional Linux dependencies.

Instructions for building, installing, or downloading the `sui-node` binary are available at [Sui Install](#). These install instructions are specific to the `sui cli`, but apply to the `sui-node` binary as well.

There are many ways to run a Sui Full node (bare metal, virtual machine, Kubernetes statefulset, and so on), and the solution that you choose depends on your specific needs as well as the infrastructure that you have available.

There are some specific considerations to keep in mind when running a Sui Full node that apply to all environments:

There's a guide in the Sui repository on running a Full node via [Docker Compose](#) . This alone is not suitable for a production environment, but can be used to get a Full node up and running quickly on a virtual machine or local machine for development purposes. Refer to [Running a Full node](#) for instructions relevant to production use cases.

When you are ready to run sui-node in your production environment, you can set up your Full node by completing the following steps:

Make a copy of the [Full node YAML template](#) :

Download the genesis blob for the network to use:

For Testnet or Mainnet: Edit the fullnode.yaml file to include peer nodes for state synchronization. Append the following to the end of the current configuration:

Optional: Set up the [Archival Fallback](#) , which allows you to sync checkpoints if you fall behind the network's seed-peers .

Optional: Skip this step to accept the default paths to resources. Edit the fullnode.yaml file to use custom paths.

Update the db-path field with the path to the Full node database. db-path: "/db-files/sui-fullnode"

Update the genesis-file-location with the path to genesis.blob.

You should not start syncing your Full node from the start of the genesis. This will take a very long time and consume a lot of resources (including likely filling up your disk).

Instead, start your Full node from a recent snapshot. You can find details on how to obtain a snapshot from the [Sui Snapshots guide](#) .

Now that you have your Full node config file set up, and you've obtained a snapshot, you can start your Full node by running the sui-node binary with your fullnode.yaml configuration file:

It's a good idea to use something like systemd to manage your Full node in a production environment.

If you receive a cannot find -lpq error, you are missing the libpq library. Use sudo apt-get install libpq-dev to install on Linux, or brew install libpq on MacOS. After you install on MacOS, create a Homebrew link using brew link --force libpq . For further context, reference the [issue on Stack Overflow](#) .

If you receive the following error:

Then update the metrics address in your fullnode.yaml file to use port 9180 .

State synchronization

Sui Full nodes sync with validators to receive new transactions on the network.

A transaction requires a few round trips to $2f+1$ validators to form a transaction certificate (TxCert).

This synchronization process includes:

This synchronization process requires listening to at a minimum $2f+1$ validators to ensure that a Full node has properly processed all new transactions. Sui will improve the synchronization process with the introduction of checkpoints and the ability to synchronize with other Full nodes.

A Sui Full node is essentially a read-only view of the network state. Unlike validator nodes, Full nodes cannot sign transactions, although they can validate the integrity of the chain by re-executing transactions that a quorum of validators previously committed.

Today, a Sui Full node maintains the full history of the chain.

Validator nodes store only the latest transactions on the frontier of the object graph (for example, transactions with >0 unspent output objects).

Follow the instructions here to run your own Sui Full node. Sui Full nodes run using the sui-node binary.

Suggested minimum hardware to run a Sui Full node:

Sui recommends running Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can run a Sui Full node on macOS, but this is only recommended for development and not for production use.

Make sure to [update Rust](#) .

Use the following command to install additional Linux dependencies.

Instructions for building, installing, or downloading the sui-node binary are available at [Sui Install](#) . These install instructions are specific to the sui cli, but apply to the sui-node binary as well.

There are many ways to run a Sui Full node (bare metal, virtual machine, Kubernetes statefulset, and so on), and the solution that you choose depends on your specific needs as well as the infrastructure that you have available.

There are some specific considerations to keep in mind when running a Sui Full node that apply to all environments:

There's a guide in the Sui repository on running a Full node via [Docker Compose](#) . This alone is not suitable for a production environment, but can be used to get a Full node up and running quickly on a virtual machine or local machine for development purposes. Refer to [Running a Full node](#) for instructions relevant to production use cases.

When you are ready to run sui-node in your production environment, you can set up your Full node by completing the following steps:

Make a copy of the [Full node YAML template](#) :

Download the genesis blob for the network to use:

For Testnet or Mainnet: Edit the fullnode.yaml file to include peer nodes for state synchronization. Append the following to the end of the current configuration:

Optional: Set up the [Archival Fallback](#) , which allows you to sync checkpoints if you fall behind the network's seed-peers .

Optional: Skip this step to accept the default paths to resources. Edit the fullnode.yaml file to use custom paths.

Update the db-path field with the path to the Full node database. db-path: "/db-files/sui-fullnode"

Update the genesis-file-location with the path to genesis.blob.

You should not start syncing your Full node from the start of the genesis. This will take a very long time and consume a lot of resources (including likely filling up your disk).

Instead, start your Full node from a recent snapshot. You can find details on how to obtain a snapshot from the [Sui Snapshots guide](#) .

Now that you have your Full node config file set up, and you've obtained a snapshot, you can start your Full node by running the sui-node binary with your fullnode.yaml configuration file:

It's a good idea to use something like systemd to manage your Full node in a production environment.

If you receive a cannot find -lpq error, you are missing the libpq library. Use sudo apt-get install libpq-dev to install on Linux, or brew install libpq on MacOS. After you install on MacOS, create a Homebrew link using brew link --force libpq . For further context, reference the [issue on Stack Overflow](#) .

If you receive the following error:

Then update the metrics address in your fullnode.yaml file to use port 9180 .

Architecture

A Sui Full node is essentially a read-only view of the network state. Unlike validator nodes, Full nodes cannot sign transactions, although they can validate the integrity of the chain by re-executing transactions that a quorum of validators previously committed.

Today, a Sui Full node maintains the full history of the chain.

Validator nodes store only the latest transactions on the frontier of the object graph (for example, transactions with >0 unspent output objects).

Follow the instructions here to run your own Sui Full node. Sui Full nodes run using the sui-node binary.

Suggested minimum hardware to run a Sui Full node:

Sui recommends running Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can run a Sui Full node on macOS, but this is only recommended for development and not for production use.

Make sure to [update Rust](#) .

Use the following command to install additional Linux dependencies.

Instructions for building, installing, or downloading the sui-node binary are available at [Sui Install](#) . These install instructions are specific to the sui cli, but apply to the sui-node binary as well.

There are many ways to run a Sui Full node (bare metal, virtual machine, Kubernetes statefulset, and so on), and the solution that you choose depends on your specific needs as well as the infrastructure that you have available.

There are some specific considerations to keep in mind when running a Sui Full node that apply to all environments:

There's a guide in the Sui repository on running a Full node via [Docker Compose](#) . This alone is not suitable for a production environment, but can be used to get a Full node up and running quickly on a virtual machine or local machine for development purposes. Refer to [Running a Full node](#) for instructions relevant to production use cases.

When you are ready to run sui-node in your production environment, you can set up your Full node by completing the following steps:

Make a copy of the [Full node YAML template](#) :

Download the genesis blob for the network to use:

For Testnet or Mainnet: Edit the fullnode.yaml file to include peer nodes for state synchronization. Append the following to the end of the current configuration:

Optional: Set up the [Archival Fallback](#) , which allows you to sync checkpoints if you fall behind the network's seed-peers .

Optional: Skip this step to accept the default paths to resources. Edit the fullnode.yaml file to use custom paths.

Update the db-path field with the path to the Full node database. db-path: "/db-files/sui-fullnode"

Update the genesis-file-location with the path to genesis.blob.

You should not start syncing your Full node from the start of the genesis. This will take a very long time and consume a lot of resources (including likely filling up your disk).

Instead, start your Full node from a recent snapshot. You can find details on how to obtain a snapshot from the [Sui Snapshots guide](#) .

Now that you have your Full node config file set up, and you've obtained a snapshot, you can start your Full node by running the sui-node binary with your fullnode.yaml configuration file:

It's a good idea to use something like systemd to manage your Full node in a production environment.

If you receive a cannot find -lpq error, you are missing the libpq library. Use sudo apt-get install libpq-dev to install on Linux, or brew install libpq on MacOS. After you install on MacOS, create a Homebrew link using brew link --force libpq . For further context, reference the [issue on Stack Overflow](#) .

If you receive the following error:

Then update the metrics address in your fullnode.yaml file to use port 9180 .

Full node setup

Follow the instructions here to run your own Sui Full node. Sui Full nodes run using the sui-node binary.

Suggested minimum hardware to run a Sui Full node:

Sui recommends running Sui Full nodes on Linux. Sui supports the Ubuntu and Debian distributions. You can run a Sui Full node on macOS, but this is only recommended for development and not for production use.

Make sure to [update Rust](#) .

Use the following command to install additional Linux dependencies.

Instructions for building, installing, or downloading the sui-node binary are available at [Sui Install](#) . These install instructions are specific to the sui cli, but apply to the sui-node binary as well.

There are many ways to run a Sui Full node (bare metal, virtual machine, Kubernetes statefulset, and so on), and the solution that you choose depends on your specific needs as well as the infrastructure that you have available.

There are some specific considerations to keep in mind when running a Sui Full node that apply to all environments:

There's a guide in the Sui repository on running a Full node via [Docker Compose](#) . This alone is not suitable for a production environment, but can be used to get a Full node up and running quickly on a virtual machine or local machine for development purposes. Refer to [Running a Full node](#) for instructions relevant to production use cases.

When you are ready to run sui-node in your production environment, you can set up your Full node by completing the following steps:

Make a copy of the [Full node YAML template](#) :

Download the genesis blob for the network to use:

For Testnet or Mainnet: Edit the fullnode.yaml file to include peer nodes for state synchronization. Append the following to the end of the current configuration:

Optional: Set up the [Archival Fallback](#) , which allows you to sync checkpoints if you fall behind the network's seed-peers .

Optional: Skip this step to accept the default paths to resources. Edit the fullnode.yaml file to use custom paths.

Update the db-path field with the path to the Full node database. db-path: "/db-files/sui-fullnode"

Update the genesis-file-location with the path to genesis.blob.

You should not start syncing your Full node from the start of the genesis. This will take a very long time and consume a lot of resources (including likely filling up your disk).

Instead, start your Full node from a recent snapshot. You can find details on how to obtain a snapshot from the [Sui Snapshots guide](#) .

Now that you have your Full node config file set up, and you've obtained a snapshot, you can start your Full node by running the sui-node binary with your fullnode.yaml configuration file:

It's a good idea to use something like systemd to manage your Full node in a production environment.

If you receive a cannot find -lpq error, you are missing the libpq library. Use sudo apt-get install libpq-dev to install on Linux, or brew install libpq on MacOS. After you install on MacOS, create a Homebrew link using brew link --force libpq . For further context, reference the [issue on Stack Overflow](#) .

If you receive the following error:

Then update the metrics address in your fullnode.yaml file to use port 9180 .

Running a Full node

Instructions for building, installing, or downloading the sui-node binary are available at [Sui Install](#) . These install instructions are specific to the sui cli, but apply to the sui-node binary as well.

There are many ways to run a Sui Full node (bare metal, virtual machine, Kubernetes statefulset, and so on), and the solution that you choose depends on your specific needs as well as the infrastructure that you have available.

There are some specific considerations to keep in mind when running a Sui Full node that apply to all environments:

There's a guide in the Sui repository on running a Full node via [Docker Compose](#) . This alone is not suitable for a production environment, but can be used to get a Full node up and running quickly on a virtual machine or local machine for development purposes. Refer to [Running a Full node](#) for instructions relevant to production use cases.

When you are ready to run sui-node in your production environment, you can set up your Full node by completing the following steps:

Make a copy of the [Full node YAML template](#) :

Download the genesis blob for the network to use:

For Testnet or Mainnet: Edit the fullnode.yaml file to include peer nodes for state synchronization. Append the following to the end of the current configuration:

Optional: Set up the [Archival Fallback](#) , which allows you to sync checkpoints if you fall behind the network's seed-peers .

Optional: Skip this step to accept the default paths to resources. Edit the fullnode.yaml file to use custom paths.

Update the db-path field with the path to the Full node database. db-path: "/db-files/sui-fullnode"

Update the genesis-file-location with the path to genesis.blob.

You should not start syncing your Full node from the start of the genesis. This will take a very long time and consume a lot of resources (including likely filling up your disk).

Instead, start your Full node from a recent snapshot. You can find details on how to obtain a snapshot from the [Sui Snapshots guide](#) .

Now that you have your Full node config file set up, and you've obtained a snapshot, you can start your Full node by running the sui-node binary with your fullnode.yaml configuration file:

It's a good idea to use something like systemd to manage your Full node in a production environment.

If you receive a cannot find -lpq error, you are missing the libpq library. Use sudo apt-get install libpq-dev to install on Linux, or brew install libpq on MacOS. After you install on MacOS, create a Homebrew link using brew link --force libpq . For further context, reference the [issue on Stack Overflow](#) .

If you receive the following error:

Then update the metrics address in your fullnode.yaml file to use port 9180 .