**Activation of Kusama Validator**

**Source:** https://docs.senseinode.com/run-a-validator-or-polkadot-6a8f08aef96447ad91136ae2f51d54f6

**Initial Set-up**

**Requirements**

The most common way for a beginner to run a validator is on a cloud server running Linux. You may choose whatever VPS provider that your prefer, and whatever operating system you are comfortable with. For this guide we will be using Ubuntu 18.04, but the instructions should be similar for other platforms.

The transactions weights in Polkadot were benchmarked on standard hardware. It is recommended that validators run at least the standard hardware in order to ensure they are able to process all blocks in time. The following are not minimum requirements but if you decide to run with less than this beware that you might have performance issue.

**Standard Hardware**

For the full details of the standard hardware please see here.

* **CPU** - Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz
* **Storage** - An NVMe solid state drive of 1 TB (As it should be reasonably sized to deal with blockchain growth).
* **Memory** - 64GB ECC.

The specs posted above are by no means the minimum specs that you could use when running a validator, however you should be aware that if you are using less you may need to toggle some extra optimizations in order to be equal to other validators that are running the standard.

**Step 1:** Clone the Polkadot Repository

git clone https://github.com/paritytech/polkadot-sdk.git

cd polkadot-sdk

**Step 2: Build** the Polkadot Binary

cargo build –release –package polkadot

**Step 3:** Make a directory

mkdir kusama

**Step 4:** Generate Node Key

./target/release/polkadot key generate-node-key –file kusama-file

**Step 5:** Create a script

sudo nano kusama.sh

Type this command

#!/bin/bash

./target/release/polkadot \

--validator \

--chain=kusama \

--base-path kusama \

--node-key-file kusama-file \

--rpc-port 4545 \

--unsafe-rpc-external \

--name “Node Name”

Save the file.

Make the script executable

sudo chmod +x kusama.sh

**Step 6:** Run the script

./kusama.sh

**Step 7:** Bond Dot

It is highly recommended that you make your controller and stash accounts be two separate accounts. For this, you will create two accounts and make sure each of them have at least enough funds to pay the fees for making transactions. Keep most of your funds in the stash account since it is meant to be the custodian of your staking funds.

Make sure not to bond all your KSM balance since you will be unable to pay transaction fees from your bonded balance.

It is now time to set up our validator. We will do the following:

* Bond the KSM of the Stash account. These KSM will be put at stake for the security of the network and can be slashed.
* Select the Controller. This is the account that will decide when to start or stop validating.

Go to Network > Staking > Account > then + Stash

* **Stash account** - Select your Stash account. In this example, we will bond 1 KSM, where the minimum bonding amount is 1. Make sure that your Stash account contains at least this much. You can, of course, stake more than this.
* **Controller account** - Select the Controller account created earlier. This account will also need a small amount of KSM in order to start and stop validating.
* **Value bonde**d - How much KSM from the Stash account you want to bond/stake. Note that you do not need to bond all of the KSM in that account. Also note that you can always bond more KSM later. However, withdrawing any bonded amount requires the duration of the unbonding period. On Kusama, the unbonding period is 7 days. On Polkadot, the planned unbonding period is 28 days.
* **Payment destination** - The account where the rewards from validating are sent. More info here. Starting with runtime version v23 natively included in client version 0.9.3, payouts can go to any custom address. If you'd like to redirect payments to an account that is neither the controller nor the stash account, set one up. Note that it is extremely unsafe to set an exchange address as the recipient of the staking rewards.

Once everything is filled in properly, click Bond and sign the transaction with your Stash account.

After a few seconds, you should see an ExtrinsicSuccess message.

Your bonded account will available under Stashes. You should now see a new card with all your accounts (note: you may need to refresh the screen). The bonded amount on the right corresponds to the funds bonded by the Stash account.

**Step 8:** Generate Session Keys

Go to Developer > RPC Calls > author > rotateKeys then Submit RPC calls

The result is your session keys.

**Step 9:** Submitting the setKeys Transcation

Submitting the setKeys Transaction

You need to tell the chain your Session keys by signing and submitting an extrinsic. This is what associates your validator with your Controller account.

Go to Staking > Account Actions, and click "Set Session Key" on the bonding account you generated earlier. Enter the output from author\_rotateKeys in the field and click "Set Session Key".

Submit this extrinsic and you are now ready to start validating.

**Step 10:** Setup via Validator Tab

Go to Network > Staking > Accounts then click +Validator.

Here you will need to input the Keys from rotateKeys, which is the Hex output from author\_rotateKeys. The keys will show as pending until applied at the start of a new session.

The "reward commission percentage" is the commission percentage that you can declare against your validator's rewards. This is the rate that your validator will be commissioned with.

* **Payment preferences** - You can specify the percentage of the rewards that will get paid to you. The remaining will be split among your nominators.

**SETTING A COMMISSION RATE OF 100% SUGGESTS THAT YOU DO NOT WANT YOUR VALIDATOR TO RECEIVE NOMINATIONS**

You can also determine if you would like to receive nominations with the "allows new nominations" option.

Click "Bond & Validate".

If you go to the "Staking" tab, you will see a list of active validators currently running on the network. At the top of the page, it shows the number of validator slots that are available as well as the number of nodes that have signaled their intention to be a validator. You can go to the "Waiting" tab to double check to see whether your node is listed there.

The validator set is refreshed every era. In the next era, if there is a slot available and your node is selected to join the validator set, your node will become an active validator. Until then, it will remain in the waiting queue. If your validator is not selected to become part of the validator set, it will remain in the waiting queue until it is. There is no need to re-start if you are not selected for the validator set in a particular era. However, it may be necessary to increase the number of KSM staked or seek out nominators for your validator in order to join the validator set.

**Requirements:**

**Source:** https://www.reddit.com/r/Kusama/comments/mdciiv/how\_much\_ksm\_do\_i\_need\_for\_a\_validator\_node/?utm\_source=share&utm\_medium=ios\_app&utm\_name=iossmf

To be a validator you need 3 KSM to keep your identity on-chain

The minimum you need to stake is pretty much 0 KSM, but that's not going to make your node look attractive. You'll want to write your identity on chain and get it judged by a registrar (that costs .3 KSM and you need to lock up ~2 KSM to keep your identity on-chain). That way you'll get that sexy checkmark! You'll also want to show nominators that you have skin in the game by self-staking a non-trivial amount. I think a good amount of KSM would be a decent show of good faith.